

## Article

# Harnessing Large Language Models and Deep Neural Networks for Fake News Detection

Eleftheria Papageorgiou, Iraklis Varlamis \*  and Christos Chronis 

Department of Informatics and Telematics, Harokopio University of Athens, GR-17778 Athens, Greece; csi23310@hua.gr (E.P.); chronis@hua.gr (C.C.)

\* Correspondence: varlamis@hua.gr

**Abstract:** The spread of fake news threatens trust in both traditional and digital media. Early detection methods, based on linguistic patterns and handcrafted features, struggle to identify more sophisticated misinformation. Large language models (LLMs) offer promising solutions by capturing complex text patterns, but challenges remain in ensuring their accuracy and generalizability. This study evaluates LLM-based feature extraction for fake news detection across multiple datasets. We compare BERT-based text representations, introduce a method for extracting factual segments from news articles, and create two new datasets with fact-based features. Additionally, we explore graph-based text representations using LLMs to capture relationships within news content. By integrating these approaches, we improve fake news detection, making it more accurate and interpretable. Our findings provide insights into how LLMs and graph-based techniques can enhance misinformation detection.

**Keywords:** fake news; large language models (LLMs); text classification



Academic Editor: Nirmalya Thakur

Received: 13 March 2025

Revised: 2 April 2025

Accepted: 5 April 2025

Published: 8 April 2025

**Citation:** Papageorgiou, E.; Varlamis, I.; Chronis, C. Harnessing Large Language Models and Deep Neural Networks for Fake News Detection. *Information* **2025**, *16*, 297. <https://doi.org/10.3390/info16040297>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The dissemination of deceptive information as factual in an attempt to influence public opinion, which is prevalent in fake news, poses several challenges to social trust towards traditional media and social media. Traditional fake news detection approaches, which initially were sufficient for the task, nowadays face many difficulties in detecting news that uses sophisticated language and fabricated evidence to appear credible, and they rely on more complex representation models and deep neural network architectures to capture the intrinsic characteristics of fake news. The advances in natural language processing (NLP), particularly through large language models (LLMs), offer promising solutions by capturing complicated text patterns and improving detection accuracy [1]. However, several challenges still remain unsolved, and these refer to the generalizability and interpretability of the results of such solutions.

Despite the recent advancements in NLP and deep neural network technology, existing approaches often face limitations in feature extraction and representation, which hinder their effectiveness and generalizability. The methods that rely on shallow linguistic patterns or handcrafted features demonstrate low performance since they fail to capture the context-dependent nature of fake news. The approaches that rely on machine learning and deep learning techniques still struggle to fully capture the rich semantics and syntactic structures of news content [2]. To tackle these issues, more complex neural network structures, such as graph neural networks, have been recommended for capturing the differences between genuine and fake news in content as well as in the way they spread in social networks [3]. In

addition, pre-trained large language models are used by researchers as tools for extracting complex language patterns and text features, generating context-aware embeddings, and capturing the meaning of text at a deeper level.

While LLMs and graph neural networks have shown promise, there is a notable gap in the literature in terms of the evaluation of their performance across diverse datasets, which is critical for understanding their generalizability and robustness. The potential of combining LLMs with graph-based representations and factual segment extraction remains largely unexplored. Graph-based approaches can model the structural relationships within text, while factual segment extraction can isolate the most relevant and verifiable parts of news articles, potentially enhancing detection accuracy and advancing the state of the art in fake news detection.

The primary objective of this work is to evaluate the performance of LLM-based feature extraction techniques in fake news detection. For this purpose, we evaluate different approaches that rely on pre-trained LLMs to extract features, across multiple datasets, and examine the generalizability of the approach. We also use LLMs to extract useful representations of news texts and employ deep learning methods to distinguish between fake and legitimate news, thus providing more insights into the intrinsic patterns that are used to mislead the audience. The contributions of the proposed work can be summarized in the following:

- It conducts a comprehensive comparison of BERT-based text representations in various popular datasets, to examine the generalization of the approach;
- It proposes a method for extracting factual segments from texts using LLMs, isolating the most verifiable and information-rich parts of news articles, and representing these segments using BERT-based features;
- It contributes two new datasets that are based on the ISOT fake news dataset [4], a dataset that was collected from real-world sources (truthful sources from Reuters.com and unreliable websites that were flagged by Politifact) but contains factual information extracted from its articles using LLMs;
- It explores the representation of texts as graphs using large language models (LLMs) to capture structural relationships within the content, enabling a deeper understanding of the connections between entities and concepts.

By integrating these innovative techniques, this work aims to push the boundaries of fake news detection, offering new methodologies that enhance both accuracy and interoperability.

Section 2 that follows provides the necessary theoretical background and key concepts relevant to fake news detection and the role of large language models (LLMs) in this task. Section 3 illustrates the three alternative approaches that we examine and compare. Section 4 details the experimental evaluation process, while Section 5 presents and analyzes the results. Section 6 discusses the limitations of the study and the future challenges in the field, and Section 7 concludes the paper by summarizing key findings and outlining future research directions.

## 2. Related Work

### 2.1. Fake News Detection Using LLMs

Fake news detection was initially implemented by using traditional machine learning models like logistic regression, Naive Bayes, Support Vector Machines (SVMs), and decision trees [5]. These methods depended on specific features and had to adapt to the constantly evolving patterns of misinformation [6]. Contextual understanding through Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) [7] was introduced with the advent of deep learning by using dense word vectors from pre-trained

models such as Word2Vec and GloVe. Transformer models like Bidirectional Encoder Representations from Transformers (BERT) [8] and RoBERTa, which captured deeper semantic relationships, came along later as further improvements. BERT is a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based upon their connection. A new era began with the introduction of large language models (LLMs, like GPT-4, LLaMa2, and PaLM), where pre-trained models on large text datasets were able not only to produce coherent human-like textual outputs but also to easily adapt to a variety of detection tasks with minimal supervision, thus making them capable of tackling complex tasks like fake news identification. As the field advances, LLMs often integrate with graph neural networks (GNNs) in order to represent news content and its intrinsic associations more efficiently, including the extraction and representation of entities and topics.

Fact extraction from articles is crucial for detecting camouflaged fake news, as misinformation often mimics the style of genuine sources. Actually, LLMs can easily rewrite fake news to resemble the tone and the structure of the reliable real ones. As a result, LLM-generated fake news significantly reduces the effectiveness of state-of-the-art text-based detectors, with their performance dropping by up to 38% in F1-Score (i.e., the harmonic mean of precision and recall) due to the stylistic manipulations [9]. Thus, style-related features can also be exploited for style-based attacks. This demonstrates why the process of determining fake news should focus on the content rather than the writing style. Large language models (LLMs) have become an essential tool for enhancing textual features in various text-related tasks, outperforming conventional lexical-based techniques [10]. The conventional methods for fact-based extraction are inadequate, as fake news often mimics the language style of real news and can introduce misleading relationships between real-world entities and topics, which traditional models fail to detect. LLMs can extract the semantic relationships and the factual errors by utilizing deep contextual understanding and in-context learning, detecting also camouflaged misinformation. The reliability of fake news detection systems is strengthened by their ability to detect subtle differences beyond surface-level writing style significantly.

A promising alternative that can be used in extremely low-resource scenarios is known as Few-Shot Fake News Detection (FS-FND), which aims to distinguish inaccurate news from real ones by using a small number of training samples [11]. Usually, conventional supervised methods rely on extensive, annotated datasets, while FS-FND utilizes large language models (LLMs) and in-context learning in order to generalize from a small number of examples, making it more adaptable to constantly evolving patterns of misinformation.

From all the above, it is clear that pre-trained large language models (LLMs) can process vast amounts of text data and adapt across different tasks with minimal supervision, making them capable of tackling complex tasks like fake news identification or fake news generation with a specific style. These models outperform traditional machine learning approaches, as they leverage extensive pre-training on diverse datasets to enhance both the accuracy and efficiency of downstream tasks. The following section explores the evolution of LLMs, highlighting their key advancements.

## 2.2. Evolution of Large Language Models (LLMs)

The evolution of large language models began with BERT (Bidirectional Encoder Representations from Transformers), introduced by Google researchers in 2018. BERT is a language representation model that learns the language in a bidirectional manner. It has two variants: BERT-Base (12 layers, 110 M parameters) and BERT-LARGE (24 layers, 340 M parameters). BERT was trained on the BooksCorpus (800 million words) and English Wikipedia (2.5 billion words), excluding lists, tables, and headers [8]. Building on BERT, M-

BERT (Multilingual BERT) was released in 2019 as a single language model pre-trained on Wikipedia pages of 104 languages. It has a 12-layer transformer architecture and comes in several variants: BERT-Base Multilingual Cased (104 languages, 110 M parameters), BERT-Base Multilingual Uncased (102 languages, 110 M parameters), and BERT-Base Chinese (110 M parameters) [12].

Another significant model is T5 (Text-to-Text Transfer Transformer), released in 2019, which is an encoder–decoder model that converts each task into a text-to-text format. It was trained on the Colossal Clean Crawled Corpus (C4) dataset, a 750 GB dataset created from Common Crawl’s web-extracted text. The model architecture is similar to the original Transformer, with both encoder and decoder consisting of 12 blocks, and it has 220 million parameters [13]. Following T5, RoBERTa is an enhanced version of BERT with more data, dynamic masking, and byte-pair encoding. RoBERTa improves performance by training longer, using larger batches, removing the next sentence prediction objective, and dynamically changing the masking pattern. RoBERTa was trained on a large dataset called CC-NEWS [14].

In the realm of models that are fine-tuned for specific tasks, FN-BERT stands out as a BERT-based model fine-tuned on a fake news classification dataset in 2023. The model is based on distilBERT and was trained using a specific fake news dataset [15]. Meanwhile, Grover, developed by the Allen Institute for AI, is a GAN-based model designed to generate and detect fake news. Grover has three sizes: Grover-Base (117 M parameters), Grover-Large (345 M parameters), and Grover-Mega (1.5 B parameters). It was trained on the RealNews dataset, created from Common Crawl news articles [16].

Recent advancements include Llama, a collection of pre-trained and fine-tuned LLMs from Meta with varying parameter sizes. It includes Llama Chat, optimized for dialogue, and Code Llama, specialized for code generation. Llama2 was trained on 2 trillion tokens, with increased context length and grouped-query attention [17]. The recent Llama 3 series includes models with up to 405 billion parameters and offers enhanced capabilities such as multilingual support and extended context lengths.

Microsoft’s Phi-3 [18] also presents significant advancements in performance and scalability. It emphasizes improvements in computational efficiency and model adaptability, making it a strong contender in the evolving landscape of large language models. Phi-3 comes in different versions ranging from 3.8 B (Phi-3 mini) to 14 B (Phi-3 medium) parameters, which offer two context lengths, 128 K and 4 K, and outperform bigger models. They have been trained on 3.3 million tokens.

OpenAI’s GPT-3 demonstrated remarkable abilities in translation, question answering, and content generation, comparable to human-written texts. Furthermore, ChatGPT-3.5, also developed by OpenAI, is widely used for its conversational abilities. It was trained using Reinforcement Learning from Human Feedback (RLHF) and fine-tuned from a model in the GPT-3.5 series. ChatGPT can answer follow-up questions, admit mistakes, and reject inappropriate requests [19]. Continuing this progress, OpenAI’s GPT-4, released in March 2023 [20], opened new horizons by enabling the understanding and answering of questions from both text and image inputs, fostering a more detailed and comprehensive understanding of information. This progress in the GPT family has raised expectations and broadened the range of their applications.

Alibaba’s Qwen2 [21], another advanced model, focuses on enhancing accuracy and efficiency in natural language processing. Mistral AI’s Mistral models [22] and Google’s Gemini models [23] also contribute to expanding the capabilities of open standards by providing advanced solutions for complex learning problems.

Table 1 provides a summary of the most popular LLMs and their characteristics.

**Table 1.** Overview of key large language models (LLMs).

Model	Tokens	Parameters	Training Corpus Size	Architecture	Reference
BERT	3.3 B	BERT-BASE: 110 M, BERT-LARGE: 340 M	BooksCorpus (800 M words), English Wikipedia (2.5 B words)	Transformer (12/24 layers)	[8]
M-BERT	300 M	110 M	Wikipedia pages of 104 languages	Transformer (12 layers)	[12]
T5	500 M	220 M	C4 (750 GB)	Encoder–Decoder (12 blocks)	[13]
RoBERTa	500 M	355 M	CC-NEWS, Web texts (160 GB)	Enhanced BERT	[14]
FN-BERT	300 M	66 M	Fake news dataset	Fine-tuned DistilBERT	[15]
Grover	400 M	Grover-Base: 117 M, Grover-Large: 345 M, Grover-Mega: 1.5 B	RealNews dataset (120 GB)	GAN-based Transformer	[16]
Llama2	2 T	Llama2-7B, Llama2-13B, Llama2-70B	Various datasets	Transformer	[17]
Llama3	15 T	8 B, 70 B	Massive dataset (incl. Common Crawl, code, data, books)	Improved Transformer	-
Phi3	3.3 M	Small 3.8 B, Medium 14 B	Large-scale dataset	Improved Transformer	[18]
GPT-3	500 B	175 B	Diverse datasets (web, books, Wikipedia)	Multimodal Transformer	[20]
GPT-4	10 T	1.7 T	Diverse datasets (web, books, Wikipedia)	Multimodal Transformer	[20]
Qwen2	500 B	200 B	Massive dataset	Transformer	[21]
Mistral	500 B	175 B	Large-scale dataset	Transformer	[22]
Gemini	1 T	1.4 T	Massive dataset	Advanced Transformer	[23]

Despite the advancements in LLM-based fake news detection, critical research gaps persist. First, while existing methods leverage LLMs for semantic analysis, their generalizability across diverse datasets remains underexplored, particularly when handling adversarial stylistic manipulations that reduce detector effectiveness. Second, many approaches still rely on stylistic features or full-text analysis rather than isolating verifiable factual segments, leaving them vulnerable to sophisticated misinformation that blends credible writing styles with factual inaccuracies. Third, the lack of standardized datasets focusing on fact-rich representations limits progress in developing robust, content-centric detection frameworks. Finally, structural relationships between entities and concepts in news articles—key to identifying misleading claims—are often overlooked in favor of surface-level text analysis. Our work addresses these gaps through four key contributions: (1) a comprehensive evaluation of BERT-based representations across multiple datasets to assess generalizability, (2) a novel LLM-driven method for extracting and encoding factual segments to bypass stylistic biases, (3) two new datasets derived from ISOT, enriched with LLM-extracted factual information to enable content-focused analysis, and (4) graph-based representations of textual structure to capture entity relationships for deeper semantic verification. By integrating these innovations, our approach advances fake news detection by prioritizing factual integrity and structural coherence over superficial stylistic patterns, as detailed in the following sections.

### 3. Proposed Approaches

The use of large language models (LLMs) for fake news detection can be applied in several ways. Different LLMs may either be used to extract valuable information from

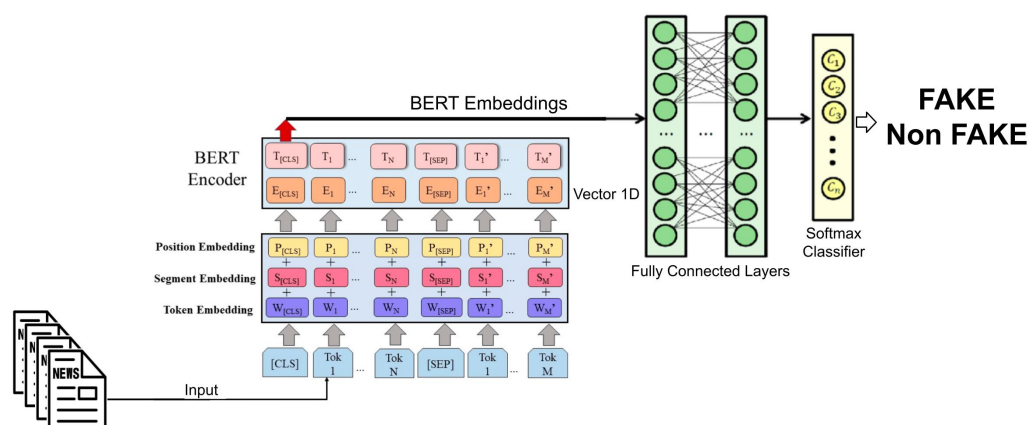


news texts or to directly convert these texts into embeddings for analysis. In this section, we describe the three approaches that we evaluate in this study: (1) using BERT embeddings to train a classifier, (2) extracting factual sentences from news articles using an LLM, and (3) generating knowledge graphs from the news content with an LLM.

### 3.1. Text Representation Using BERT Embeddings

In the first approach, we leverage the power of BERT to convert news texts into dense, contextualized embeddings that capture the nuanced semantics of the content. BERT is pre-trained on vast corpora of text and can generate rich representations that encode both the meaning and context of the input sentences. By feeding each news article through BERT, we obtain a fixed-size embedding vector that encapsulates the textual information in a form that is well suited for further analysis.

These embeddings are then used as input features for a classifier designed to distinguish between genuine and fake news. Specifically, we employ a deep neural network (DNN) to process the BERT embeddings. The DNN, with its series of fully connected layers and non-linear activation functions, is trained to learn complex patterns in the feature space that are indicative of deceptive content. The overall process is depicted in Figure 1. Through this approach, the classifier benefits from both the advanced language representations provided by BERT and the robust pattern recognition capabilities of deep neural networks, ultimately improving the accuracy of fake news detection.



**Figure 1.** The flow of the baseline method that uses BERT embeddings and a DNN classifier.

### 3.2. Extraction of Factual Data Using LLMs

The extraction of facts and claims from news text is a critical task against the rapid dissemination of misinformation. Fact and claim extraction enables the identification of verifiable statements within news articles, facilitating efficient fact checking. Recent advancements in the field, such as claim normalization [24] and abstraction summarization [25], condense the fundamental assertion from a text and facilitate verification by fact checkers.

The method that we implemented employs a pre-trained LLM to extract factual sentences from the news text. It employs a very simple prompt (see Listing 1) that is sent to the LLM asking for the sentences to be returned in a JSON format.

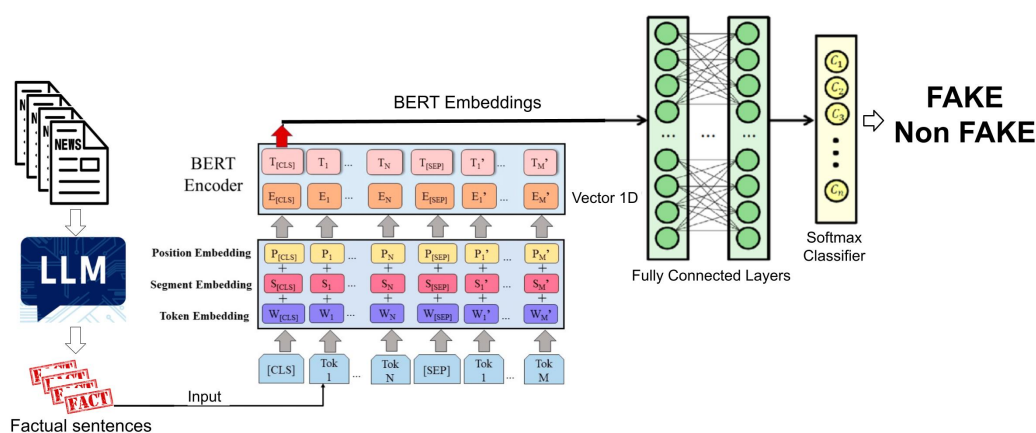
**Listing 1.** The prompt for extracting factual sentences in JSON format.

```

prompt:> ‘‘Extract all the factual sentences from this text: [text].
Return only text segments and do not generate additional text.
Respond in a JSON object that contains sentences as a list
of strings.
The following is an acceptable JSON string:
{{‘‘sentences’’: [‘‘sentence 1’’, ‘‘sentence 2’’, etc.]}}’’

```

The resulting sentences are converted to embeddings using BERT and the aggregated embeddings of the text are fed to the DNN for classification. The overall process is depicted in Figure 2.

**Figure 2.** The flow of the method that uses the LLM to extract factual sentences from the news before using BERT embeddings and the DNN classifier.

### 3.3. Graph Extraction Using LLMs and Graph Classification

In the third approach, we combine the power of LLMs in extracting useful information from raw text and the ability of deep neural networks, and more specifically, graph convolutional networks, to capture more complex relations hidden in the text.

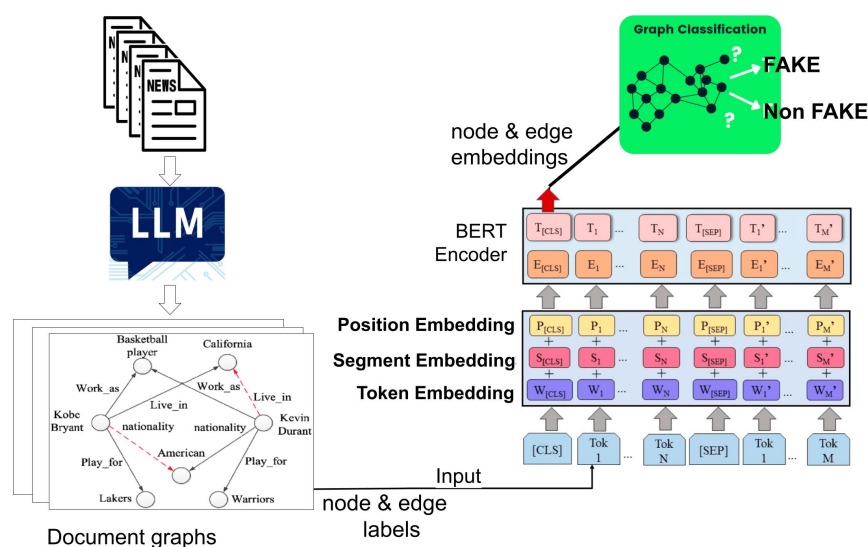
The first step of the approach, which is depicted in Figure 3, is to feed the news text to an LLM and obtain the main entities mentioned in the text and their relations. For the entities we obtain a label and a type and for the relations we obtain the source and target entity, the type of the relation, and a short description. It is possible that self-referring relations are extracted for an entity, as additional descriptions that do not connect two different entities. In this case, the source and target entity of the relation will be the same. The prompt that is used in this case is depicted in Listing 2.

**Listing 2.** The prompt for extracting a graph from text in JSON format.

```

prompt:> ‘‘Extract a graph (nodes & edges) from this text: [text].
Use JSON notation. Nodes must have attributes id (a number),
label (the text that describes the node), and type (a string
such as person, location, organization, etc). Edges must have
attributes source, target, relation, and description. Source
and target must use the node IDs as numbers. Avoid having null
values for source and target. When you are not sure, use the
same value for both source and target. Return only the JSON
code that contains the list of nodes and the list of edges.
The following is an acceptable JSON string:
{‘‘nodes’’: [{‘‘id’’: 1,
‘‘label’’: ‘‘a text value’’, ‘‘type’’: ‘‘a short text value’’},
{‘‘id’’: 2, ‘‘label’’: ‘‘a text value’’, ‘‘type’’: ‘‘a short text
value’’}],
edges’’: [{‘‘source’’: 1, ‘‘target’’: 2, ‘‘relation’’: ‘‘a short
text value’’, ‘‘description’’: ‘‘a text value’’}]}

```

**Figure 3.** The flow of the method that uses the LLM to extract graphs from text, BERT to generate node and edge embeddings from the graph labels, and a GCN as a classifier.

The output of this process is a JSON file for each news item that in essence represents a labeled graph, with labels in the nodes and edges. The next step is to represent the news item as a graph with the labeled entities as nodes and the labeled relations as edges, and for this purpose, we employ the NetworkX Python (<https://networkx.org/>, accessed on 13 March 2025) library. The resulting document graphs are parsed with BERT to extract node and edge embeddings from the respective labels.

The last step of the process is to train a graph convolutional network classifier with the resulting graphs. The GCN learns to classify the whole graph as a fake or non-fake news graph.

#### 4. Experimental Evaluation

In order to experimentally evaluate the feasibility of the proposed alternatives, we perform a series of experiments in which four popular fake news datasets that vary in size, structure, and labeling schemes are employed. In the first set of experiments, we apply the



simple approach which uses the whole news text to generate BERT embeddings and train the classifier. This allows us to test the performance of a popular approach across various datasets and evaluate its ability to generalize. The second set of experiments compares the three proposed approaches in the same dataset in an attempt to examine whether fact or named entity and relation extraction using LLMs, and deeper neural networks such as GCNs, can improve fake news detection performance.

#### 4.1. Datasets

The dataset used in the experiments comprises four popular and publicly available datasets in the fake news domain: the LIAR dataset, the FakeNewsNet dataset, the Politifact fact-check data dataset, and the ISOT Fake News Dataset. The latter has also been used in the second set of experiments, and for this dataset, we created the JSON files that contain the factual sentences extracted by the LLM and the JSON files that contain the document graph as it was extracted by the LLM. We make these two new datasets that have been created using ISOT as a basis publicly available for researchers who work in this field and are interested to test more methods on these refined datasets. In the preparation of these datasets, we used Llama 3.1 and DeepSeek-R1 as our LLMs and we accessed them through the OLLAMA service.

Table 2 summarizes the datasets used in our experiments.

**Table 2.** Fake news datasets’ overview.

Dataset	Source	Number of Total Articles	Labels	Source	URL
LIAR benchmark	Articles	12,800	pants-fire, false, barely true, half-true, mostly true, and true	Politifact.com	<a href="https://paperswithcode.com/dataset/liar">https://paperswithcode.com/dataset/liar</a> (accessed on 12 March 2025)
Politifact fact-check data	fact-Claims	19,423	full-flop, half-flip, no-flip, true, mostly true, half-true, barely true, false, pants-fire	Politifact.com	<a href="https://www.kaggle.com/datasets/rmisra/politifact-fact-check-dataset/data">https://www.kaggle.com/datasets/rmisra/politifact-fact-check-dataset/data</a> (accessed on 12 March 2025)
FakeNewsNet (GossipCop, Politifact)	Social Media Posts	23,202	fake, real	Twitter	<a href="https://github.com/KaiDMML/FakeNewsNet">https://github.com/KaiDMML/FakeNewsNet</a> (accessed on 12 March 2025)
ISOT Fake News Dataset	Articles	44,898	fake, real	real-world sources, Reuters.com	<a href="https://www.kaggle.com/datasets/emineyetm/fake-news-detection-datasets">https://www.kaggle.com/datasets/emineyetm/fake-news-detection-datasets</a> (accessed on 12 March 2025)

##### 4.1.1. LIAR Dataset

LIAR is a publicly available dataset for fake news detection [26] and contains 12.8 K manually labeled short statements collected from politifact.com. Each statement is evaluated by a politifact.com editor for its truthfulness. The LIAR dataset contains three TSV files containing the train, test, and validation data for training, testing, and validating the model, respectively [27], see Table 3. This 80%-10%-10% split is typical in machine learning and has been provided by the creators of the dataset.

**Table 3.** LIAR dataset statistics.

Dataset Statistics	
Training set size	10,269
Validation set size	1284
Testing set size	1283

#### 4.1.2. PolitiFact Fact-Check Data Dataset

The PolitiFact fact-check data dataset [28] is collected from the PolitiFact website. It contains the claims made by individuals in terms of blog posts and the opinions of the PolitiFact curators [29]. It contains several fields for each blog post as described below:

- verdict: The verdict of fact check in one of six categories: true, mostly true, half-true, mostly false, false, and pants-fire.
- statement originator: The person who made the statement being fact checked.
- statement: The statement being fact checked.
- statement date: The date when the statement was made.
- statement source: The source where the statement was made, categorized as: speech, television, news, blog, social media, advertisement, campaign, meeting, radio, email, testimony, statement, or other.
- factchecker: The name of the person who fact checked the claim.
- factcheck date: The date when the fact-checked article was published.
- factcheck analysis link: The link to the fact-checked analysis article.

#### 4.1.3. FakeNewsNet Dataset

The FakeNewsNet dataset contains the GossipCop and the PolitiFact dataset; each of them consists of samples related to fake news collected from GossipCop and PolitiFact respectively [30–32]. Each dataset contains two CSV files with fake and real news, which are being used in our experiments [33]. Table 4 summarizes the number of fake and real news articles that GossipCop and Politifact contain.

**Table 4.** Number of articles per subdataset.

Dataset	Number of Articles	Number of Fake News Articles	Number of Real News Articles
Gossipcop	22,144	5325	16,819
Politifact	1058	433	625

Each of the above CSV files is a comma-separated file and has the following columns:

- id—Unique identifier for each news;
- url—Url of the article from web that published that news;
- title—Title of the news article;
- tweet\_ids—Tweet ids of tweets sharing the news. This field is a list of tweet ids separated by tab.

#### 4.1.4. ISOT Fake News Dataset

The ISOT Fake News Dataset contains two types of articles: fake and real ones. The real articles were obtained by crawling articles from Reuters.com (News website), while the fake ones came from different sources from unreliable websites that were flagged by Politifact (a fact-checking organization in the USA) and Wikipedia. The dataset articles deal with different topics; however, the majority of articles focus on political and world news topics. Each article contains the following information: article title, text, type, and the date the article was published on [34]. Table 5 summarizes the information regarding the articles contained in the dataset.

**Table 5.** ISOT articles per type and domain.

News Type	Size (Number of Articles)	Subjects	
		Domain	#articles
Real news	21,417	World	10,145
		Politics	11,272
Fake news	23,481	Domain	#articles
		Government	1570
		Middle East	778
		US	783
		Left news	4459
		Politics	6841
		General news	9050

#### 4.2. ISOT Facts Datasets

As mentioned earlier, with this work, we contribute two new datasets to the domain that are generated from the articles in the ISOT dataset. We processed all articles using the respective prompts and two LLMs which were accessed through the OLLAMA [35] REST API.

In the case of factual sentence extraction, Llama 3.1 with 8 billion parameters has been used. For each article, a corresponding JSON file is generated, that contains only the extracted sentences, as shown in Listing 3.

**Listing 3.** JSON format of the extracting sentences.

```
{ 'sentences': ['sentence 1', 'sentence 2', ...]}
```

In the case of document graph extraction, DeepSeek-R1 with 1.5 billion parameters has been used. For each article, a corresponding JSON file is generated, which contains the extracted entities as nodes and their relations as edges. Each node contains an id, a label, and a type, while each edge contains the source, the target, the relation between the source and target, and a description regarding the connection. The structure of each JSON file is shown in Listing 4.

Both datasets are available for download at: <https://github.com/Eleftheria-99/isot-facts-datasets> (accessed on 12 March 2025).

#### 4.3. Performance Metrics

There is a large variety of evaluation metrics used in the task of fake news or fake profiles detection. The metrics consider fake news or profile detection as a binary classification task. Table 6 summarizes the most widely used evaluation metrics.

**Listing 4.** JSON format of the extracted graph.

```

{
  'nodes': [
    { 'id': number, 'label': 'string', 'type': 'string'},
    ...,
    { 'id': number, 'label': 'string', 'type': 'string'}
  ],
  'edges': [
    { 'source': number, 'target': number,
      'relation': 'string', 'description': 'string'},
    ...,
    { 'source': number, 'target': number,
      'relation': 'string', 'description': 'string'}
  ]
}

```

**Table 6.** Evaluation metrics for fake news classification [36].

Metric	Description
Accuracy	The proportion of correctly classified statements, providing a general measure of the models' performance.
Precision	The ratio of true positive predictions to the total positive predictions, measuring the models' ability to correctly identify true statements without generating false positives.
Recall	The ratio of true positive predictions to the total actual positives, assessing the models' ability to detect all true statements, capturing their sensitivity to true information.
F1-Score	The harmonic mean of precision and recall, offering a balanced measure of the models' performance, considering both false positives and false negatives.

#### 4.4. Implementation Details

##### 4.4.1. Generating BERT Embeddings

In the first experiment, the code implements a binary fake news classification system using BERT (Bidirectional Encoder Representations from Transformers). It utilizes PyTorch (<https://pytorch.org/>, accessed on 13 March 2025) and Hugging Face's Transformers library to fine-tune a pre-trained BERT-base-uncased model on a dataset containing both real and fake news articles. The dataset is loaded into Pandas dataframes. In the experiments with FakeNewsNet or LIAR, the classification is binary with the label 0 for fake news articles and 1 for real news. In the case of the Politifact check dataset, the classification is multiclass with labels: true, mostly true, half-true, barely true, false, or pants-fire. The dataset is split into training (60%), validation (20%), and test (20%) sets using a stratified train test split. The BertTokenizerFast is used for text tokenization while truncating/padding sequences to a fixed length, appropriate in each case. For text tokenization, the BertTokenizerFast is used, utilizing the bertbaseuncased tokenizer. The tokenized sequences (input ids, attention mask) are converted into PyTorch tensors for model training.

Regarding the model architecture, a class named BERT Arch is implemented, which also extends the nn.Module. In each experiment, dense layers, normalization layers, dropout layers, or LogSoftmax activation functions are used. The dense layers contain a GELU activation function for classification. The LayerNorm layer stabilizes training, and the dropout layer is used to prevent overfitting. Usually, the final layer is a LogSoftmax activation function to output probabilities for the two classes. All layers of the BERT model are frozen initially to prevent them from being updated during training. However, in each

experiment, the last  $N$  layers are unfrozen, allowing for fine-tuning of those layers during training. This is carried out to reduce the number of parameters updated, speeding up the training.

The mechanism of early stopping is also used. This means that in case there is no improvement in validation loss after a specific number of epochs, the training stops. This mechanism prevents overfitting, saves computation time, and ensures the best model is used, as it saves the best model whenever validation loss improves.

#### 4.4.2. The Deep Neural Network Used for the ISOT Dataset

The DNNClassifier is in this case a deep neural network designed for classification tasks, featuring two fully connected layers with ReLU activation and dropout for regularization, followed by sentence-level aggregation via mean pooling and a final classification layer. The network processes multi-sentence inputs by reshaping and aggregating embeddings, making it suitable for tasks like document classification or sentiment analysis.

#### 4.4.3. The Implementation of GCN Variations

Using the ISOT Fake News dataset, we test three different variations of the GCN-based approach, the BERT embedding approach using the full text, and the BERT embedding approach using only the factual sentences. In this set of experiments we kept a random 10% of the dataset for testing (held-out set) and the remaining 90% was further split into training and validation using an 80%-20% stratified ratio. The goal of these experiments was to measure the effectiveness of LLMs in extracting useful information from texts, and how this information affects the GCN classification performance. The texts in these datasets are assigned with labels 0 and 1 for fake and true news articles, respectively.

In the first experiment, each article of the ISOT Fake News Dataset is given to the LLM for graph extraction through appropriate prompting. The graph representation of each article is saved in JSON format, which contains nodes to represent key elements and edges to capture inter-article relationships. Then, it is converted into a NetworkX graph, and after it has been preprocessed using BERT-based tokenization, it is saved as a PyTorch Geometric graph, where text embeddings are generated using a pre-trained BERT model. The nodes found in each JSON file are leveraged for the generation of embeddings. The saving of the graphs in PyTorch's .pt format ensures efficient usage of storage and easier retrieval for model training and evaluation. The model utilizes a graph convolutional network (GCN) to capture relational dependencies between news articles. For the classification pipeline, a two-layer GCN is employed, where the first layer captures neighborhood information and the second layer refines node embeddings, followed by global mean pooling to aggregate the final node embeddings into a single representation, making it useful for classification tasks, like fake news detection. A RELU function is also being used between the two GCN layers. The model is trained in a supervised setting, leveraging a labeled dataset.

In the second experiment, the transformation of the dataset's article into PyTorch Geometric graphs, the usage of the GCN model, the training, and the evaluation of the model follow the same process. The only difference with the first experiment is that the BERT embeddings are created from both nodes and edges contained in the JSON file. The incorporation of both node and edge-level semantics makes the model more robust in learning complex graph structures. As a result, the contextual learning from graph-structured data improves.

The third experiment follows the same process as the second one, but with some optimizations in order to achieve better results. Basically, the structure of the GCNClassifier class has changed. Instead of the two GCN layers, a RELU, and a global mean pooling function, the architecture consists of two GINEConv layers, a residual connection, a RELU,

and the global mean pooling step. The first layer transforms the input node features into a 32-dimensional representation. The second layer maps the 32-dimensional features to the output space. The model also incorporates a residual connection, which guarantees that the original node features are retained and incorporated to the final embeddings. This way improves stability.

In all cases, the `train_test_split` function is used to split the dataset into training, validation, and test sets. Model evaluation is performed on the test set.

## 5. Experimental Results

All experiments were performed on real-world text corpora from the datasets listed in Table 2, avoiding synthetic or simulated data. This ensures that our findings reflect practical challenges in fake news detection. This section and its subsections present the results of the experimental evaluation process described in the previous section.

### 5.1. State of the Art Performance

Before reporting our results, we choose to report on the best results achieved on the four datasets according to the related literature. Table 7 provides a summary of the best approaches per dataset. The feature size is reported when the authors of the respective publication give it.

**Table 7.** The best performance in each of the four datasets as reported in the literature.

Dataset	Model	Feature Extraction	Accuracy	F1	Feature Size	Citation
LIAR	Hybrid CNN Models (Text + All Metadata)	Randomly initialized embeddings for metadata, concatenated the max-pooled text representations	0.27	-	N/A	[26]
PolitiFact fact check data	Multilayer Perceptron	BoW for Text and Metadata	0.65	0.59	18,000	[37]
Fake News Net	RoBERTa-MWSS (Clean + Weak)	weak social supervision signals	0.80	0.80	N/A	[30]
ISOT Fake News	LSVM	TF-IDF feature extraction	0.92	-	50,000	[4]

Researchers have used the PolitiFact fact-check data dataset to develop and evaluate automated fact-checking systems that aim to enhance the robustness of fact checking by detecting and filtering out leaked and unreliable evidence [38], or to evaluate the performance of ChatGPT prompts in directly classifying news headlines in one of the six verdict classes [39]. Although the performance of zero-shot classification using ChatGPT ranged between 49% and 62%, there exist some code paradigms on the Kaggle page of the dataset that combine Bag-of-Words models for text and metadata (like who made the statement, where it came from, dates, named entities, emotions, and sentiment) with Multi-Layer-Perceptrons and achieve an accuracy of 65%.

The highest accuracy score for LIAR Dataset belongs to the Hybrid CNN model, incorporating text and all metadata [26]. The accuracy score achieved on the test set was 27.4%. This model outperformed traditional text-only classifiers such as SVMs, Logistic Regression, Bi-LSTMs, and standard CNNs, which performed lower in comparison. The integration of speaker-related metadata features contributed significantly to improving classification accuracy. This shows how crucial contextual information beyond just textual characteristics is in detecting false information or misleading claims.

The RoBERTa-MWSS (Clean + Weak) model achieved the highest performance for fake news detection using the FakeNewsNet Dataset [30], which contains the GossipCop and the PolitiFact datasets. The accuracy score of the RoBERTa-MWSS (Clean + Weak) model



was 80%, outperforming other machine learning and deep learning approaches. The data used in the experiment are manually labeled (clean) and weekly supervised/labeled (weak). This model integrates RoBERTa, a pre-trained transformer-based language model, with a Multi-Source Weak Social Supervision (MWSS) framework [40], an approach that integrates multiple weak social supervision signals (labels) from social engagements of users.

The highest accuracy score for ISOT Fake News Dataset belongs to linear SVM, where a 92% score was achieved. With a feature size of 5000, TF-IDF (Term Frequency-Inverse Document Frequency) was used to achieve this outcome. The feature size used in the experiment represents the number of n-gram features used in the text classification models and can range from 1000 up to 50,000 as mentioned in the study [4]. The larger the feature sizes are, the more contextual and lexical information they capture, usually improving classification performance. In this case, the score indicates that linear models benefit from a large feature collection. The dataset includes both real and fake news articles, making it a benchmark for fake news detection.

The effectiveness of TF-IDF feature extraction in this dataset emphasizes the importance of term frequency weighting in order to distinguish between real and misleading content. For the implementation of TF-IDF, Scikit-learn was utilized. TfidfVectorizer converts the documents into a matrix of TF-IDF features, while TfidfTransformer is used to normalize the TF-IDF representation. These steps belong to the preprocessing process of the data, and these are essential in optimizing the SVM model's performance.

## 5.2. BERT-Embedding Generalization

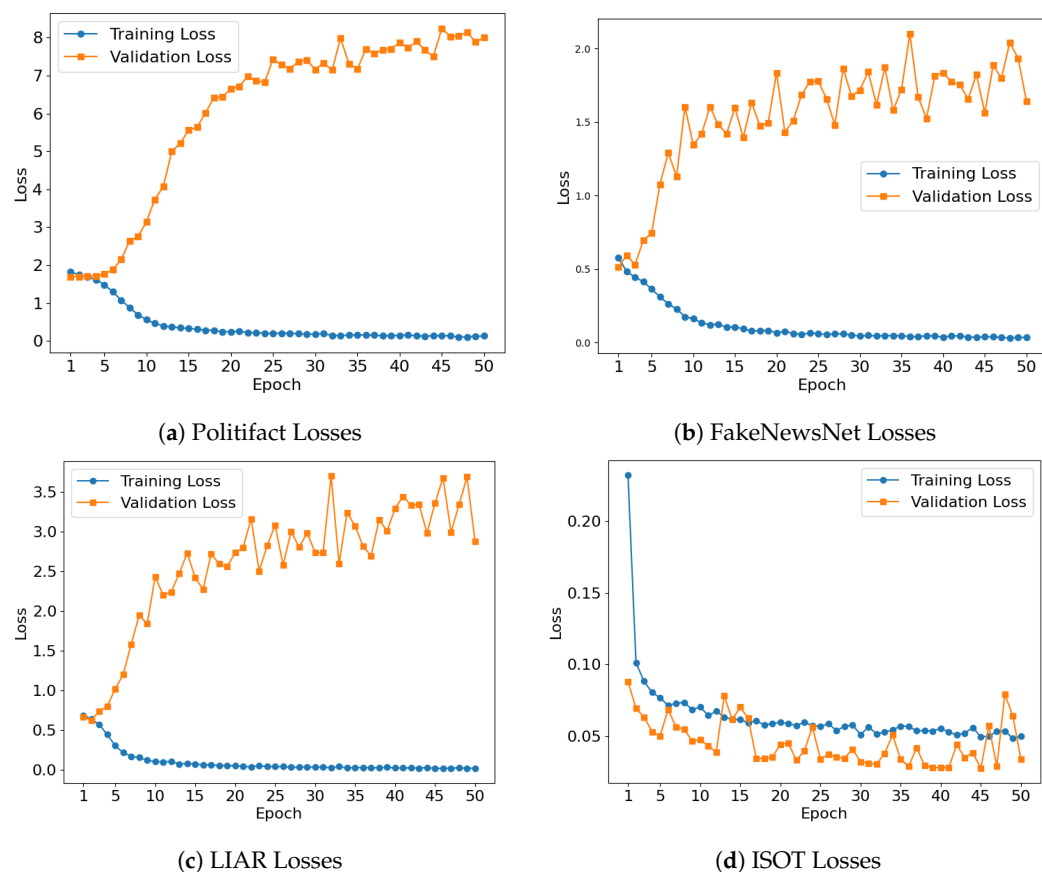
For the first set of experiments that tests the generalization of the simple BERT-embeddings-based approach, across different datasets, all predictions are made on the test set of each dataset. Various evaluation metrics such as precision, recall, F1-score, and Matthews Correlation Coefficient (MCC) are calculated. In all cases, the `train_test_split` function is used to split the dataset into training, validation, and test sets. The test set is 20% as mentioned in the previous section, on which the model evaluation is also performed. Each experiment is executed for 50 epochs. To visualize the performance, the loss curves on the training and validation subsets of each dataset are depicted in Figure 4.

The results show that the performance of the classifier that uses BERT embeddings of the articles' texts varies across datasets. The training loss quickly drops after a few training epochs. However, the validation loss is still high and drops slower.

In Table 8, we see similar behavior on the test subset of each dataset. The accuracy on the datasets with two classes (i.e., FakeNewsNet and ISOT) is high, reaching 99% in the case of ISOT Fake News. The respective scores in Politifact and LIAR are much lower since the classifier has to learn a multi-class task. These results have been achieved using the pre-trained BERT embeddings and a deep neural network.

**Table 8.** Performance of the BERT-embeddings-based classifier on the respective test subsets of each dataset.

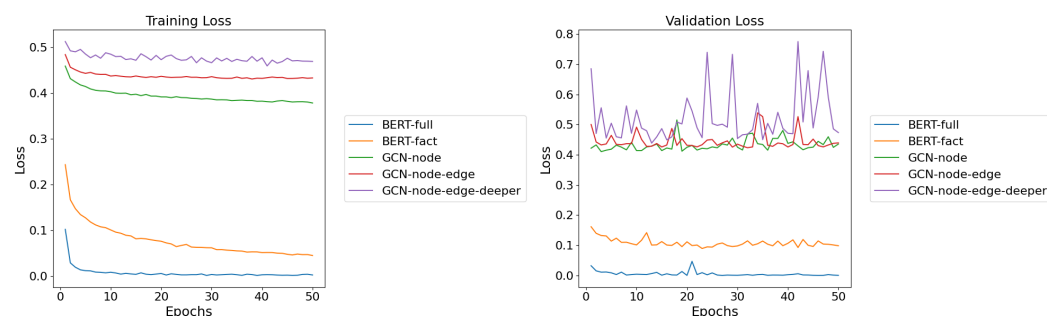
Dataset	Matthews Corr. Coefficient	Precision	Recall	F1-Score	Accuracy
LIAR	0.28	0.48	0.69	0.64	0.63
Politifact fact check data (overall metrics)	0.14	0.31	0.28	0.28	0.28
FakeNewsNet	0.50	0.91	0.80	0.80	0.79
ISOT Fake News	0.98	0.99	0.99	0.99	0.99



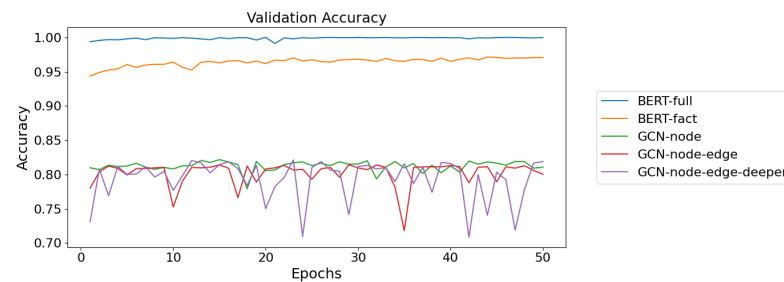
**Figure 4.** Comparison of losses for different datasets—BERT.

### 5.3. Combining LLMs and Deep NNs on the ISOT Dataset

The methods that we evaluate on the ISOT dataset are: (i) the simple approach (*BERT-full*) that takes the BERT embeddings from the full text and trains a deep neural network, (ii) the fact-based approach (*BERT-fact*) that generates BERT embeddings only for factual sentences extracted by the LLM and trains the same DNN, (iii) the graph extraction approach that uses node information only (*GCN-node*) as extracted by the LLM and trains a GCN classifier, (iv) the graph extraction approach that uses node and edge information (*GCN-node-edge*) as extracted by the LLM and trains the same GCN classifier, and (v) the graph extraction approach that uses node and edge information (*GCN-node-edge-deeper*) as extracted by the LLM and trains a more complex GCN classifier. The results that we obtain using the 80–20 training test split on the 90% of the ISOT dataset are presented in Table 9 and Figures 5 and 6.



**Figure 5.** The training and validation accuracy of the LLM-based methods on the ISOT dataset.



**Figure 6.** The validation accuracy of the LLM-based methods on the ISOT dataset.

As shown in the two figures, the training and validation loss improve for the first few epochs for all models but they stabilize after epoch 20. From the table, it is obvious that the performance of full-text BERT embeddings is better than all other techniques, which denotes that using the full content of the article is better than extracting specific information from the text. Among the three GCN-based approaches, the use of both node and edge information is preferred and the use of deeper NN architectures with more layers slightly improves the overall performance.

**Table 9.** A comparison of the LLM-based approaches on the test subset of the ISOT dataset.

Method	Accuracy	Precision	Recall	F-Measure	Math. Corr. Coef.
BERT-full	0.9993	0.9993	0.9993	0.9993	0.9987
BERT-fact	0.9708	0.9709	0.9708	0.9708	0.9415
GCN-node	0.8174	0.818	0.8174	0.8168	0.6332
GCN-node-edge	0.8082	0.8086	0.8082	0.8077	0.6147
GCN-node-edge-deeper	0.816	0.8166	0.816	0.8155	0.6305

## 6. Challenges and Limitations

Our study revealed that pre-trained large language models (LLMs) such as BERT or Llama and deep neural networks present significant advancements in fake news detection. However, they also have several challenges to address to further improve the robustness and generalizability of the proposed methods.

The first challenge is the generalization of any approach across datasets. Our study has shown that the performance of the BERT-embeddings-based classifier varies significantly across different datasets, which is an indication that the approaches struggle to generalize across datasets with different characteristics. Our primary focus on binary classification tasks may not fully capture the complexity of real-world fake news detection, where multi-class classification is often required. It is among our next objectives to examine more robust feature extraction and representation techniques that can adapt to varying data structures and labeling schemes.

The second challenge lies in the dependence on full-text content for optimal performance. While full-text BERT embeddings yield high accuracy, they may lack interpretability and efficiency compared to methods focusing on specific factual segments or structural relationships. Additionally, relying on full-text content increases computational complexity, making it less feasible for real-time applications. The limited effectiveness of graph-based representations and factual segment extraction may be due to the use of generic LLMs, which are not fine-tuned for the task. Although these methods provide deeper insights into structural relationships and verifiable information, they may not fully encapsulate the nuanced semantics required for accurate detection. In this direction, we aim to explore information extraction and representation methods that balance accuracy with interpretability and computational efficiency. We will also focus on refining information extraction techniques that use LLMs to enhance their performance and reliability.

Another challenge relates to the significant computational resource requirements associated with using large language models (LLMs) and deep neural networks, particularly for full-text embeddings and graph-based representations. Training and inference with these models demand substantial computational power, which can be prohibitive for real-time applications or resource-constrained environments. While the study demonstrates the effectiveness of these approaches, it does not extensively address their computational costs. Moving forward, we aim to explore optimization techniques, such as model distillation or quantization, to reduce resource demands while maintaining performance, making these methods more accessible and practical for broader deployment.

Finally, various ethical and social concerns arise from the deployment of LLM-based fake news detection systems, including biases in the training data and the risk of false positives/negatives. Although these concerns are not explicitly addressed in the current study, it is among our future plans to consider the broader ethical and social implications of automated fake news detection, and engage with stakeholders to ensure that the technology is used responsibly.

## 7. Conclusions

This study explores the potential of large language models (LLMs) and deep neural networks to advance fake news detection, addressing the limitations of traditional methods that struggle with sophisticated misinformation. By evaluating BERT-based text representations, introducing factual segment extraction, and exploring graph-based text representations, we demonstrate significant improvements in detection accuracy and interpretability. The integration of LLMs with deep learning techniques, such as deep neural networks (DNNs) and graph convolutional networks (GCNs), provides a robust framework for capturing complex linguistic patterns and structural relationships within news content. Our findings highlight the effectiveness of full-text BERT embeddings while also revealing the challenges of generalizing across diverse datasets and balancing accuracy with computational efficiency.

Despite these advancements, several challenges remain, including the need for more robust generalization across datasets, improved interpretability of model decisions, and reduced computational resource requirements. Future research will focus on refining feature extraction techniques, optimizing model architectures for efficiency, and exploring ethical implications to ensure responsible deployment. By addressing these challenges, we aim to push the boundaries of fake news detection, offering more reliable and scalable solutions to combat the growing threat of misinformation in both traditional and digital media.

**Author Contributions:** Conceptualization, I.V.; data curation, C.C.; funding acquisition, I.V.; investigation, E.P. and C.C.; experiment execution, C.C. and E.P.; project administration, I.V.; resources, E.P.; validation, I.V.; visualization, C.C.; writing—original draft, E.P.; writing—review and editing, I.V. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Both datasets are available for download at: <https://github.com/Eleftheria-99/isot-facts-datasets> (accessed on 12 March 2025).

**Acknowledgments:** The authors would like to thank the MPhil program in Computer Science and Informatics of Harokopio University of Athens <https://mphil.dit.hua.gr/en/home/> for supporting this work.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Papageorgiou, E.; Chronis, C.; Varlamis, I.; Himeur, Y. A survey on the use of large language models (LLMs) in fake news. *Future Internet* **2024**, *16*, 298. [CrossRef]
- Mallick, C.; Mishra, S.; Senapati, M.R. A cooperative deep learning model for fake news detection in online social networks. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 4451–4460. [CrossRef] [PubMed]
- Phan, H.T.; Nguyen, N.T.; Hwang, D. Fake news detection: A survey of graph neural network methods. *Appl. Soft Comput.* **2023**, *139*, 110235. [PubMed]
- Ahmed, H.; Traore, I.; Saad, S. Detecting opinion spams and fake news using text classification. *Secur. Priv.* **2018**, *1*, e9.
- Khanam, Z.; Alwasel, B.; Sirafi, H.; Rashid, M. Fake news detection using machine learning approaches. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Sanya, China, 12–14 November 2021; IOP Publishing: Bristol, UK, 2021; Volume 1099, p. 012040.
- Yi, J.; Xu, Z.; Huang, T.; Yu, P. Challenges and Innovations in LLM-Powered Fake News Detection: A Synthesis of Approaches and Future Directions. *arXiv* **2025**, arXiv:cs.CL/2502.00339.
- Jain, P.; Sharma, S.; Aggarwal, P.K. Classifying fake news detection using SVM, Naive Bayes and LSTM. In Proceedings of the 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Virtual, 27–28 January 2022; IEEE: New York, NY, USA, 2022; pp. 460–464.
- Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In 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), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
- Wu, J.; Guo, J.; Hooi, B. Fake News in Sheep’s Clothing: Robust Fake News Detection Against LLM-Empowered Style Attacks. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Barcelona, Spain, 25–29 August 2024; ACM: New York, NY, USA, 2024; pp. 3367–3378.
- Ma, X.; Zhang, Y.; Ding, K.; Yang, J.; Wu, J.; Fan, H. On Fake News Detection with LLM Enhanced Semantics Mining. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024; pp. 508–521.
- Liu, Y.; Zhu, J.; Liu, X.; Tang, H.; Zhang, Y.; Zhang, K.; Zhou, X.; Chen, E. Detect, Investigate, Judge and Determine: A Knowledge-guided Framework for Few-shot Fake News Detection. *arXiv* **2025**, arXiv:cs.CL/2407.08952.
- Pires, T.; Schlinger, E.; Garrette, D. How Multilingual is Multilingual BERT? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4996–5001.
- Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:cs.CL/1907.11692.
- Fn-Bert. 2023. Available online: [https://huggingface.co/ungjus/Fake\\_News\\_BERT\\_Classifier](https://huggingface.co/ungjus/Fake_News_BERT_Classifier) (accessed on 13 March 2025).
- Zellers, R.; Holtzman, A.; Rashkin, H.; Bisk, Y.; Farhadi, A.; Roesner, F.; Choi, Y. Defending against neural fake news. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 9054–9065.
- Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv* **2023**, arXiv:cs.CL/2307.09288.
- Abdin, M.; Aneja, J.; Awadalla, H.; Awadallah, A.; Awan, A.A.; Bach, N.; Bahree, A.; Bakhtiari, A.; Bao, J.; Behl, H.; et al. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. *arXiv* **2024**, arXiv:cs.CL/2404.14219.
- OpenAI. Chatgpt 3.5. Available online: <https://chat.openai.com/chat> (accessed on 13 March 2025).
- OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; et al. GPT-4 Technical Report. *arXiv* **2024**, arXiv:cs.CL/2303.08774.
- Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; et al. Qwen2 Technical Report. *arXiv* **2024**, arXiv:cs.CL/2407.10671.
- Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. *arXiv* **2023**, arXiv:cs.CL/2310.06825.
- Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A.M.; Hauth, A.; Millican, K.; et al. Gemini: A Family of Highly Capable Multimodal Models. *arXiv* **2024**, arXiv:cs.CL/2312.11805.
- Sundriyal, M.; Chakraborty, T.; Nakov, P. From Chaos to Clarity: Claim Normalization to Empower Fact-Checking. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 6–10 December 2023; pp. 6594–6609.
- Bhatnagar, V.; Kanojia, D.; Chebrolu, K. Harnessing Abstractive Summarization for Fact-Checked Claim Detection. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 2934–2945.

26. Wang, W.Y. “Liar, Liar Pants on Fire”: A New Benchmark Dataset for Fake News Detection. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 422–426.
27. LIAR Dataset. Available online: <https://paperswithcode.com/dataset/liar> (accessed on 2 March 2025).
28. Ganesh, S.K. Politifact Factcheck Data Dataset. Available online: <https://www.kaggle.com/datasets/shivkumarganesh/politifact-factcheck-data/data> (accessed on 2 March 2025).
29. Misra, R.; Grover, J. Do not ‘fake it till you make it!’ synopsis of trending fake news detection methodologies using deep learning. In *Deep Learning for Social Media Data Analytics*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 213–235.
30. Shu, K.; Mahudeswaran, D.; Wang, S.; Lee, D.; Liu, H. FakeNewsNet: A Data Repository with News Content, Social Context and Spatiotemporal Information for Studying Fake News on Social Media. *J. Big Data* **2020**, *8*, 171–188. [CrossRef] [PubMed]
31. Shu, K.; Sliva, A.; Wang, S.; Tang, J.; Liu, H. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explor. Newsl.* **2017**, *19*, 22–36.
32. Shu, K.; Wang, S.; Liu, H. Beyond News Contents: The Role of Social Context for Fake News Detection. *arXiv* **2018**, arXiv:cs.SI/1712.07709.
33. FakeNewsNet. Available online: <https://github.com/KaiDMML/FakeNewsNet> (accessed on 2 March 2025).
34. Bozkus, E. ISOT Fake News Dataset. 2024. Available online: <https://www.kaggle.com/datasets/emineyetm/fake-news-detection-datasets/data/> (accessed on 2 March 2025).
35. Ollama. Available online: <https://ollama.com/> (accessed on 2 March 2025).
36. Boissonneault, D.; Hensen, E. Fake News Detection with Large Language Models on the LIAR Dataset. *Res. Sq.* **2024**. [CrossRef]
37. Sophie, B. Predicting PolitiFact’s Verdict. 2023. Available online: <https://www.kaggle.com/code/sophieb/predicting-politifact-s-verdict> (accessed on 2 March 2025).
38. Chrysidis, Z.; Papadopoulos, S.I.; Papadopoulos, S.; Petrantonakis, P. Credible, unreliable or leaked?: Evidence verification for enhanced automated fact-checking. In Proceedings of the 3rd ACM International Workshop on Multimedia AI against Disinformation, Phuket, Thailand, 10 June 2024; pp. 73–81.
39. Li, Y.; Zhai, C. An exploration of large language models for verification of news headlines. In Proceedings of the 2023 IEEE International Conference on Data Mining Workshops (ICDMW), Shanghai, China, 1–4 December 2023; IEEE: Shanghai, China, 2023; pp. 197–206.
40. Shu, K.; Zheng, G.; Li, Y.; Mukherjee, S.; Awadallah, A.H.; Ruston, S.; Liu, H. Leveraging Multi-Source Weak Social Supervision for Early Detection of Fake News. *arXiv* **2020**, arXiv:cs.LG/2004.01732.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.