mathematics

MDPI

Article

Improving Recommender Systems for Fake News Detection in Social Networks with Knowledge Graphs and Graph Attention Networks

Aleksei Golovin [1,*], Nataly Zhukova [2], Radhakrishnan Delhibabu [3,*] and Alexey Subbotin [1]

[1] Department of Software Engineering and Computer Applications, Saint Petersburg Electrotechnical University 'LETI', 197376 St. Petersburg, Russia; ansubbotin@stud.etu.ru
[2] St. Petersburg Federal Research Centre of the Russian Academy of Sciences (SPCRAS), 199178 St. Petersburg, Russia; zhukova.n@iias.spb.su
[3] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, India
* Correspondence: aagolovin@stud.etu.ru (A.G.); rdelhibabu@vit.ac.in (R.D.)

Abstract: This paper addresses the pervasive problem of fake news propagation in social networks. Traditional text-based detection models often suffer from performance degradation over time due to their reliance on evolving textual features. To overcome this limitation, we propose a novel recommender system that leverages the power of knowledge graphs and graph attention networks (GATs). This approach captures both the semantic relationships within the news content and the underlying social network structure, enabling more accurate and robust fake news detection. The GAT model, by assigning different weights to neighboring nodes, effectively captures the importance of various users in disseminating information. We conducted a comprehensive evaluation of our system using the FakeNewsNet dataset, comparing its performance against classical machine learning models and the DistilBERT language model. Our results demonstrate that the proposed graph-based system achieves state-of-the-art performance, with an F1-score of 95%, significantly outperforming other models. Moreover, it maintains its effectiveness over time, unlike text-based approaches that are susceptible to concept drift. This research underscores the potential of knowledge graphs and GATs in combating fake news and provides a robust framework for building more resilient and accurate detection systems.

Keywords: fake news detection; social networks; knowledge graphs; graph attention networks; recommender system; machine learning

MSC: 68T07

check for updates

Academic Editors: Silvia N. Schiaffino, Marcelo Gabriel Armentano and Ariel Monteserin

Received: 31 January 2025
Revised: 17 March 2025
Accepted: 17 March 2025
Published: 20 March 2025

Citation: Golovin, A.; Zhukova, N.; Delhibabu, R.; Subbotin, A. Improving Recommender Systems for Fake News Detection in Social Networks with Knowledge Graphs and Graph Attention Networks. *Mathematics* **2025**, *13*, 1011. https://doi.org/10.3390/math13061011

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 development of the Internet, as well as social networks, has made it possible to quickly and efficiently exchange information. Nowadays, people are more likely to get news from social networks rather than from traditional media [1]. This has led to the widespread propagation of fake news [2]. Fake news can be either low-quality or incomplete information, or deliberately false politically motivated news.

The spread of fake news has an extremely negative impact on people and society. The problem is especially acute because it is easy to mislead people into believing false information, which can lead to a fatal outcome [3]. Therefore, detecting and predicting the spread of fake news are relevant and important areas of research [4]. By quickly detecting

and predicting the spread of fake news, it is possible to promptly apply measures to prevent its further advancement [5].

Existing solutions are mainly based on systems that use a text-processing approach. The crucial problem of the systems is that they become outdated over time. Text features in news titles change as new trends appear in the world. If a fundamentally new event occurs, a system may not process it correctly. In contrast, the knowledge graph approach leverages the structural relationships within the data, ensuring greater resilience and adaptability to these changes. Furthermore, knowledge graphs can capture complex relationships and contextual information that text-based methods may overlook, thus enabling a more comprehensive understanding of information dissemination.

In recent years, graph attention networks (GATs) have emerged as a powerful tool for various applications, including fake news detection. While several graph-based models exist, such as graph convolutional networks (GCNs), GATs have unique advantages. Unlike GCNs, which treat all neighboring nodes with equal significance, GATs utilize an attention mechanism to assign different weights to nodes based on their relevance. This allows the model to prioritize influential users and adaptively capture the intricacies of information propagation in social networks. Additionally, GAT's ability to dynamically adjust these weights means that they remain effective even as new data are introduced, making them particularly suitable for the ever-evolving landscape of fake news dissemination. By leveraging GAT's capabilities, it is possible to enhance the accuracy and robustness of the fake news detection system, making them a superior choice for our specific research objectives.

In this work, a recommender system for binary classification of news articles into fake and real for social networks was developed and implemented. The developed system is built on the basis of a graph model with attention (GAT), which, unlike other models, takes into account the importance of nodes in the graph. In contrast to traditional natural language processing models, the use of this model allows the value of accuracy and F1-score to be maintained over time and improves the effectiveness of detection of fake news.

A special technique was used to evaluate the effectiveness of the graph model, providing a correct comparison with other models. The evaluation was conducted using the same dataset presented in two formats: tabular and knowledge graph, constructed by converting the dataset using RDF Mapping Language (RML) [6]. This approach allowed us to correctly analyze the capabilities of various models for solving the problem of fake news detection and evaluate the effectiveness and prospects of the graph approach in comparison with others.

An open-source software package with the implementation of the proposed model was developed (https://github.com/A1gord/FakeNewsDetection accessed on 16 March 2025) and its practical usage showed a significant improvement in the results of fake news detection compared to analogs. The resulting F1-score value is 95% which indicates the high effectiveness of the developed model. The software package, along with the implementation of the proposed graph-based model, also contains the implementation of the natural language processing model.

The developed software package can be used as a comprehensive solution to solve the problem of detecting fake news in various user scenarios, taking into account the advantages of different approaches.

The main contributions of this research are as follows:

- A recommender system for detecting fake news in social networks is proposed, based on the use of knowledge graphs, which maintains its performance over time compared to systems based only on text processing.
- A comprehensive open-source software package for detecting fake news in social networks was developed and implemented.

In the first part of this paper, an overview of existing solutions for fake news detection problems is provided. Next, the theoretical background and the proposed solution are given. At the end, the description of the evaluation technique is given and the results of the developed model evaluation are presented.

2. Overview of Existing Solutions

In recent scientific research [7,8], solution methods based on classical machine learning models, such as the Naive Bayes classifier, logistic regression, and SVM, are widely used as a baseline [9,10]. However, these models are outdated and generally show poor results. The Naive Bayes classifier also has the unrealistic assumption of feature independence, which can lead to poor generalization ability on complex data.

Currently, the BERT neural network language model is widely used to solve various natural language processing problems. One such task is determining the reliability of news, where BERT is also successfully used [11]. In addition to the original BERT transformer, there is a family of modified versions of this model. These models have also gained widespread popularity, leading to the emergence of a research field known as BERTology, which looks at the internal workings of models and their performance on various tasks [12]. Recent studies [13–15] have actively used models based on BERT transformer architecture.

The main advantage of BERT models is that they are bidirectional and take into account context on both sides of a word when predicting its meaning and role. This allows BERT to better understand the meaning and the structure of the text. The crucial disadvantage of BERT and other methods based only on natural language processing is the loss of efficiency over time [16]. The textual features of headlines change as new trends emerge in the world.

Knowledge graph technology plays an important role in improving the results of systems that use BERT [17], significantly improving the efficiency of fake news classification. The combination of the BERT language model and knowledge graph technology finds its application in the KG-BERT algorithm, which is used to restore missing nodes and links in a graph [18].

However, KG-BERT technology also has a significant problem called knowledge noise (KN), where the addition of too much knowledge causes the sentence to deviate from its correct meaning [19].

Today, systems that use only graph approaches (for example, GNN) are actively used to solve the problem of detecting fake news [20–22]. In [20], the authors propose a model based on user activity analysis, using the architecture of graph convolutional neural networks (GCNs), which allows the detection of fake news without reference to text features. The work in [21] uses external sources, such as Wikipedia, in addition to the main dataset, to classify fake news. The authors used the BiLSTM neural network to extract features from the text. The paper [22] proposes a new approach to detecting fake news based not on the news content data, but on information about its propagation in the social network Twitter. It is proposed to use BiMGCL architecture and the news propagation structure in the form of a graph as features. Studies [23,24] consider the use of knowledge graphs together with reinforcement learning and language models for multimodal fake news detection. In [25], the authors propose to use tensor factorization with sparse and graph regularization for the fake news detection problem on social networks. However, this method fails to explicitly account for the significance of individual nodes and relationships within the knowledge graph. These factors are often critical in evaluating the influence of various users and assessing the reliability of information sources.

Nevertheless, these solutions do not take into account the importance of relations and nodes, which can cause poor results due to the large number of elements in the knowledge graph. A summary of existing solutions is presented in Table 1.

Table 1. Summary of related work for fake news detection in social networks.

Year [Ref]	Algorithms	Features	Best Performance
2023 [7]	Classical Machine Learning	Word, character and functional n-grams	F1 (89%)
2023 [8]	Classical Machine Learning	Word, character and functional n-grams	F1 (88%)
2021 [11]	FakeBERT	GloVe, TF-IDF, word2Vec	Accuracy (98%)
2022 [13]	VGCN-BERT	Title and social context of news	F1 (74%)
2023 [14]	BERT-320	FastText	F1 (83%)
2022 [15]	AraBERT-COV19	TF-IDF	F1 (81%)
2019 [20]	GCN	User activity	F1 (84%)
2021 [21]	CompareNet	BiLSTM	F1 (72%)
2023 [22]	BiMGCL	Propagation structure of news	F1 (86%)
2024 [23]	AKA-Fake	Multimodal	F1 (93%)
2024 [24]	Language model + KG	Multimodal	Accuracy (91%)

Thus, considering the prospects of using knowledge graphs to solve fake news detection problems and the limitations of existing solutions, we propose a graph-based system that considers the importance of the elements in the knowledge graph.

3. Theoretical Background

Graph attention networks (GATs) are one of the most powerful types of graph neural networks [26]. They have wide applicability due to a number of reasons. In regular graph convolutional networks (GCNs), all neighboring nodes have the same importance, which is not always the right approach since each node can have its own unique significance and influence the outcome of the network. Because in the context of fake news detection, users and information sources often vary significantly in terms of their influence and credibility. To solve this problem, graph networks with attention mechanisms (GATs) have been developed. They take into account the importance of each neighboring node by assigning a weight to each connection. Thereby, the network can more accurately determine which nodes to consider when performing a task. This enables the model to prioritize the most influential users and information sources during prediction, enhancing both accuracy and robustness. Furthermore, GAT's ability to dynamically adjust these weights makes it particularly well-suited for the rapidly changing landscape of fake news, where new users and sources continuously emerge.

The attention mechanism is an important tool in various fields that require working with data graphs. It helps increase the efficiency and accuracy of neural networks, which makes it an indispensable tool for developers [27].

For the GAT architecture, a fundamentally new approach is introduced, based on the use of "self-attention" technology. This technology allows us to model the importance of nodes in graphs, which is a key aspect of solving many machine learning problems.

The building blocks used in the GAT neural network are Graph Attentional Layers [28]. The input data for the attention layer is a set of node features, $h = \left\{\vec{h}_1, \vec{h}_2, \cdots, \vec{h}_N\right\}$, $\vec{h}_i \in \mathbb{R}^F$, where N—number of nodes, and F—number of features in each node. The output layer creates a new set of node features, $h\prime = \left\{\vec{h}'_1, \vec{h}'_2, \cdots, \vec{h}'_N\right\}$, $\vec{h}'_i \in \mathbb{R}^{F'}$.

To obtain sufficient expressive power to transform input features into higher-level features, at least one trainable linear transformation is required. To this end, as an initial step, a general linear transformation is applied to each node, parameterized by a weight

matrix, $W \in \mathbb{R}^{F' \times F}$. General mechanism of attention $a : \mathbb{R}^{F'} \times \mathbb{R}^{F} \to \mathbb{R}$ is calculating attention coefficients (1):

$$e_{ij} = a(W\vec{h}_i, W\vec{h}_j). \tag{1}$$

To determine the significance of each relation, pairs of hidden vectors are required, which can be obtained by concatenating the vectors from both nodes. Only after this, it is possible to apply a new linear transformation using the weight matrix W_{att}^T (2):

$$e_{ij} = W_{att}^T[W\vec{h}_i||W\vec{h}_j], \tag{2}$$

where T—transposition operation, and $||$—concatenation operation.

The attention coefficient e_{ij} indicates the importance of features of node j for node i. In its most general formulation, the model allows each node to attend to every other node, discarding all structural information. The graph structure is introduced into the mechanism by performing masked attention. That is, e_{ij} are calculated only for nodes $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ is some neighborhood of node i in the graph. It means the first-order neighbors of node i (including i). In order for the coefficients to be easily comparable for different nodes, they are normalized for all variants of j using the $softmax$ function (3):

$$\alpha_{ij} = softmax(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} exp(e_{ik})}. \tag{3}$$

The nonlinearity of *LeakyReLU* [29] is used as the activation function. The fully expanded coefficients calculated by the attention mechanism (Figure 1) can be expressed as (4):

$$\alpha_{ij} = \frac{exp\left(LeakyReLU\left(W_{att}^T[W\vec{h}_i||W\vec{h}_j]\right)\right)}{\sum_{k \in \mathcal{N}_i} exp\left(LeakyReLU\left(W_{att}^T[W\vec{h}_i||W\vec{h}_k]\right)\right)}. \tag{4}$$

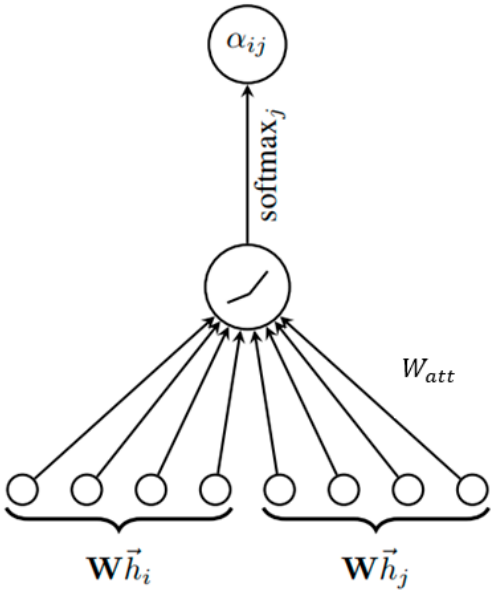


Figure 1. Illustration of the attention mechanism.

Once obtained, the normalized attention coefficients are used to compute a linear combination of their corresponding features, which serve as the final output features for each node (after possibly applying nonlinearity σ) (5):

$$\vec{h}_i' = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W\vec{h}_j\right). \tag{5}$$

In order to stabilize the process of training "self-attention", the technology of multi-headed attention is used [30]. Specifically, K independent attention mechanisms perform the transformation (5) and then their features are combined, resulting in the following output feature representation (6):

$$\vec{h}'_i = ||_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k W^k \vec{h}_j\right), \tag{6}$$

where α_{ij}^k—normalized attention coefficients computed by the k-th attention engine (a_k), and W^k is the corresponding input linear transformation weight matrix.

When applying multi-headed attention to the last (predictive) layer of the network, concatenation no longer makes sense. Instead, averaging (7) is used, and only then is the final nonlinearity applied:

$$\vec{h}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K} \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k W^k \vec{h}_j\right). \tag{7}$$

The aggregation process of the multi-head graph attention layer is illustrated in Figure 2.

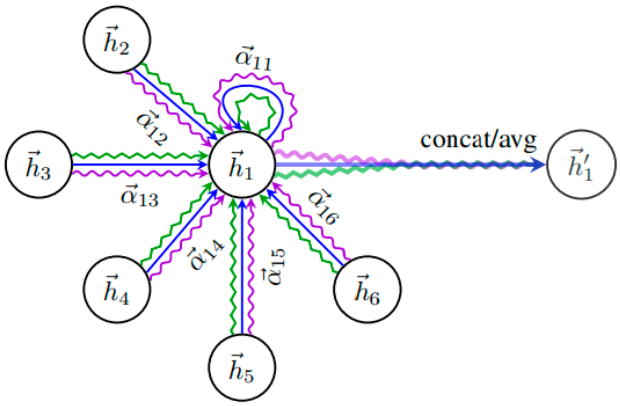


Figure 2. Illustration of node 1's multi-headed attention (K = 3) to its neighborhood.

4. Proposed Solution

4.1. Proposed System

The proposed system is based on graph neural network (GNN) technology. The GAT attention mechanism with three GATv2Conv convolutional layers was used as the graph neural network architecture (Figure 3).

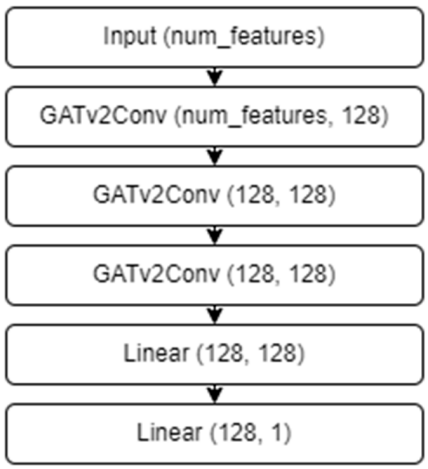


Figure 3. GAT neural network architecture.

The attention mechanism effectively assigns and adjusts weights to neighboring nodes based on their importance in the context of information propagation. The attention weights are computed through a self-attention mechanism that evaluates the relevance of each node while considering the features of its neighbors. Through this process, the attention mechanism in GAT architecture not only assigns weights based on the structural relationships in the graph but also dynamically adjusts these weights as new data come in, enhancing the system's responsiveness and accuracy in detecting fake news.

In the classic version of the GATConv convolutional layer, attention coefficients are calculated as follows (8):

$$e_{ij} = LeakyReLU\left(W_{att}^{T}[W\vec{h}_i || W\vec{h}_j]\right). \tag{8}$$

In the new modified version of the GATv2Conv, the weight matrix W is applied after the concatenation, and the attention weight matrix W_{att} is applied after the $LeakyReLU$ function. The formula for attention coefficients for GATv2Conv (9) is as follows:

$$e_{ij} = W_{att}^{T} LeakyReLU\left(W[\vec{h}_i || \vec{h}_j]\right). \tag{9}$$

GATv2Conv is considered more efficient than the classic version (GATConv) and allows for better results in benchmarks [31]. The workflow diagram of the proposed solution is presented in Figure 4.

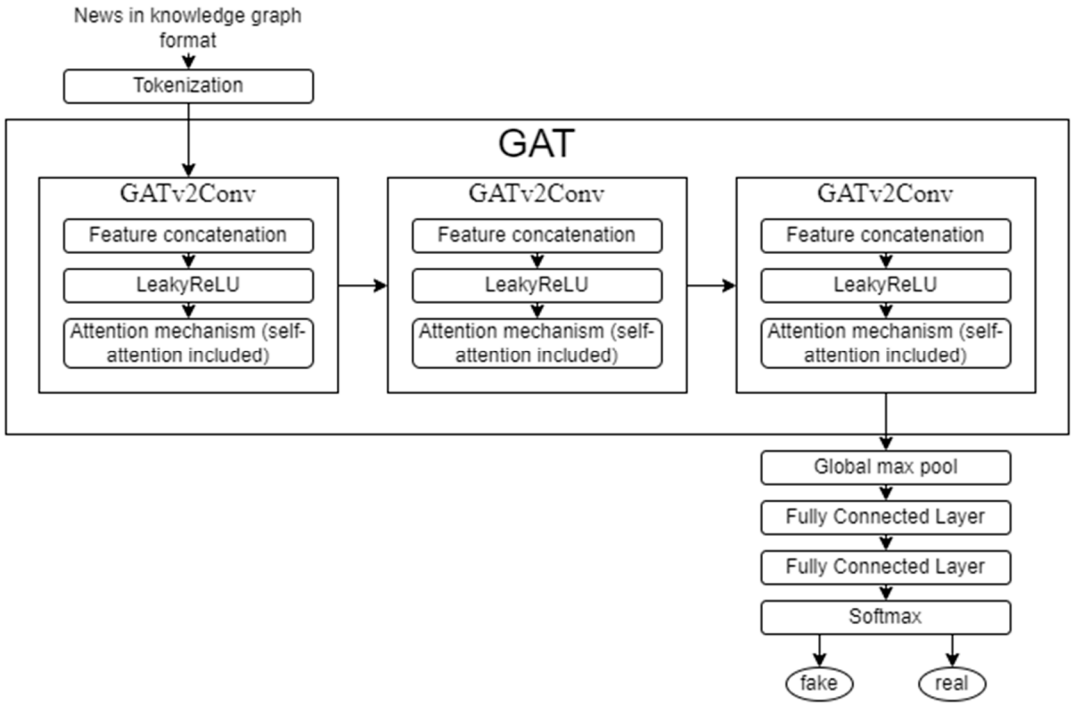


Figure 4. Graphical illustration of proposed solution.

Graph attention layers are used in two variations. In the first and second layers, multi-head attention is used, and in the third, there is only one head, in which the final result is calculated before nonlinearity.

The specific adjustments made to the model's parameters depend on the nature of the feedback. For example, if a user flags a news article as fake, the model may reduce the weight assigned to that article or its source in future predictions. Conversely, if a user provides evidence supporting the accuracy of an article, the model may increase its weight. The model also employs continuous learning techniques, allowing it to incrementally

update its knowledge graph and parameters as new data and feedback become available. This ensures that the model remains current and adapts to evolving trends and patterns in fake news propagation. Additionally, the graph attention network (GAT) architecture plays a key role in incorporating user feedback. By dynamically weighing the importance of nodes and their connections within the knowledge graph, the GAT can effectively integrate user feedback into its decision-making process.

The model's ability to continuously learn and adapt to new data helps it to identify and adjust to evolving attack strategies, potentially mitigating the impact of adversarial manipulations.

The proposed solution is included as a part of a comprehensive recommendation system for detecting fake news that assumes using both the graph-based model and existing models that are based on natural language processing. The model is selected depending on the conditions of fake news detection and considering the advantages of the graph and natural language processing approaches. There are situations when the model based on GAT is not applicable, but the model based on using text processing is able to cope with solving the problem. With GAT, the model focuses on the distribution structure of a news article. However, if the news being processed has just been published and has not yet been distributed among users, or if the recommendation system does not have access to it, then it is necessary to use the model based on the text processing approach.

Thus, we suggest using the model based on text processing in cases where the news article has just been published, or if for some reason the system cannot access the users who shared the news. In other cases, it is proposed to use the model based on the GAT neural network architecture.

4.2. Software Implementation

The open-source software package for detecting fake news in social networks, which implements classical machine learning models, the DistilBERT language model, and the proposed model, was developed. The block diagrams of the algorithms for fake news detection based on different models are presented in Figures 5–7.

The software was developed using the interactive development environment Jupyter Notebook (Python 3.10). The source code is publicly available on the GitHub platform [32]. To implement the software, the high-level interpreted programming language Python 3.10 was used. This choice is explained by the great popularity of this language in the modern scientific research community. The Python language has a huge number of ready-made and well-implemented third-party libraries. The key libraries used in the developed software are NumPy [33], Pandas [34], Matplotlib [35], PyTorch [36], and PyG [37].

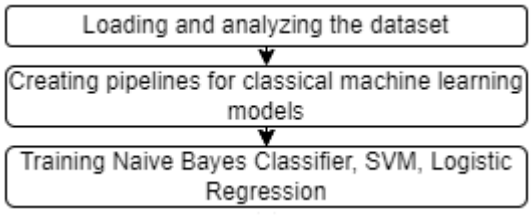


Figure 5. The block diagram of the algorithm for fake news detection based on machine learning models.

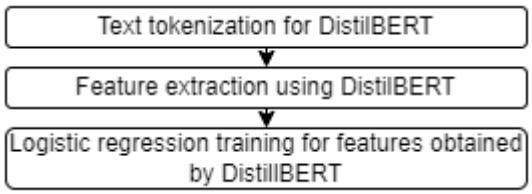


Figure 6. The block diagram of the algorithm for fake news detection based on DistilBERT language model.

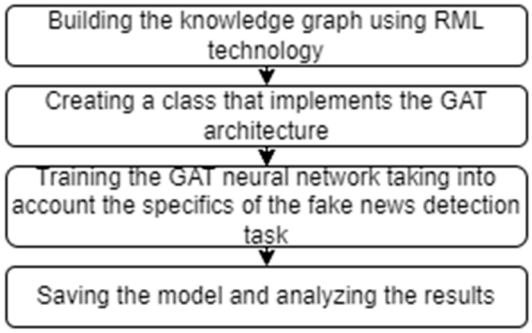


Figure 7. The block diagram of the algorithm for fake news detection based on the proposed model.

To speed up and effectively vectorize, training took place on the GPU. The Adaptive Moment Estimation (Adam) algorithm based on gradient descent was used as an optimizer. It has an adaptive learning rate, which means that the learning rate changes depending on the gradient. This allows us to ensure faster convergence to the optimal solution and avoid gradient decay. Binary Cross Entropy Loss (BCELoss) was chosen as the error function, which is one of the most common loss functions in binary classification problems. It is used to evaluate the performance of the model, which must determine whether the input object belongs to one of the two classes. BCELoss calculates the error between the model's predicted values and the actual class labels. It applies to binary data, where each object can be assigned to only one of the two classes. The BCELoss function is differentiable, allowing the use of gradient descent-based optimizers for model training.

Also, the proposed model incorporates several adaptability mechanisms that allow it to learn from new data without requiring substantial retraining.

Firstly, it leverages continuous learning techniques, enabling the model to update its knowledge base incrementally as new data become available. This allows for real-time adjustments to the model's parameters, ensuring it remains current with the latest information while minimizing the need for complete retraining.

Additionally, the graph attention network (GAT) architecture facilitates this adaptability by dynamically weighing the importance of nodes and their connections within the knowledge graph. This means that as new nodes (representing recent news articles or user interactions) are introduced, the model can effectively evaluate their significance and integrate them into its existing framework, enhancing its performance and accuracy.

Finally, the system employs techniques such as transfer learning, which enables it to utilize knowledge gained from previous training on similar datasets, thereby reducing the amount of new data required for training and accelerating the learning process.

In summary, the model's adaptability mechanisms, including continuous learning, dynamic node weighting through GAT, and transfer learning, significantly contribute to its ability to stay effective over time without extensive retraining.

5. Evaluation

5.1. Dataset

The "FakeNewsNet" dataset [38] was chosen as a dataset to solve the problem. "FakeNewsNet" was created to study the spread of fake news on social networks [39]. The dataset contains data on real and fake news, including sources, headlines, and mentions on the social network Twitter (Table 2). "FakeNewsNet" was formed on the basis of information about news publications marked by journalists from the GossipCop and PolitiFact organizations, which are authorities in the field of exposing fake news [40,41]. The "FakeNewsNet" dataset is a valuable tool for researching the phenomenon of fake news [42].

Table 2. Statistics of the "FakeNewsNet" dataset.

	Category	Features	PolitiFact		GossipCop	
			Fake	**Real**	**Fake**	**Real**
News Content	Linguistic	News articles	432	624	5323	16,817
		News articles with text	420	528	4947	16,694
	Visual	News articles with images	336	447	1650	16,767
Social Content	User	User posting tweets	95,553	249,887	265,155	80,137
		User involved in likes	113,473	401,363	348,852	145,078
		User involved in retweets	106,195	346,459	239,483	118,894
		User involved in replies	40,585	186,675	106,325	50,799
	Post	Tweets posting news	164,892	399,237	519,581	876,967
	Response	Tweets with replies	11,975	41,852	39,717	11,912
		Tweets with likes	31,692	93,839	96,906	41,889
		Tweets with retweets	23,489	67,035	56,552	24,955
	Network	Followers	405,509,460	1,012,218,640	630,231,413	293,001,487
		Followees	449,463,557	1,071,492,603	619,207,586	308,428,225
		Average followers	1299.98	982.67	1020.99	933.64
		Average followees	1440.89	1040.21	1003.14	982.80

The dataset consists of several parts. The first part contains information about news articles, such as headlines, text, and links to sources. The second part contains information about the users who published articles, as well as those who made reposts or comments.

For each news article in the dataset, its status is also indicated—real news or fake. This allows the dataset to be used to train machine learning models to detect fake news.

The data were pre-processed, cleaned, and published on the Kaggle platform [43]. Pre-processing includes combining parts of the dataset into a single one. We removed any duplicate entries and filtered out articles with incomplete data. Missing values were handled through imputation or exclusion based on their significance to the overall dataset. Common stopwords (e.g., "and", "the", "is") were eliminated from the data, as they do not provide meaningful insights for classification tasks. The resulting dataset is the CSV file in which each element is a news article publication.

The dataset was converted to a knowledge graph for further use of graph approaches [44]. In this representation, each news is a separate graph, in which the root node is the news itself, and subsequent nodes are Twitter users who shared this news with each other. In addition, the user nodes are the parent of nodes containing textual information about their social network profiles (pre-processed using BERT).

The construction of the knowledge graph involves several key steps and criteria for selecting nodes and edges. Firstly, nodes are selected based on the relevance and significance of the entities they represent within the specific context of fake news detection. These entities can include news articles, users, topics, and even external sources like social media platforms. The criteria for selecting nodes often include factors such as their influence on information dissemination, frequency of interactions, and historical data on their previous contributions to news propagation.

For edges, which depict the relationships between nodes, the selection process focuses on the nature and strength of these connections. Various types of relationships are considered, such as user interactions (retweets, shares, comments), temporal relationships

(when a news item was published, how quickly it spread), and semantic relationships (the thematic similarity between articles). The strength of these connections can be quantified through metrics like engagement rates or the frequency of interactions, thereby creating a more nuanced understanding of the network dynamics.

Moreover, advanced techniques, such as natural language processing and machine learning, are utilized to optimize the enrichment of the graphs. For instance, text data from news articles can be analyzed to identify keywords and context, facilitating the dynamic addition of nodes and edges that encapsulate new trending topics or emerging influencers.

The use of one dataset in two formats, tabular and graph, allows for a correct assessment of the graph model in comparison with traditional text processing approaches.

5.2. Technique for Evaluation of the Effectiveness of the Developed Model

An experiment was conducted to study the properties of the proposed solution for detecting fake news on social networks using knowledge graphs. Two different formats of input data were utilized. The classic tabular format of the "FakeNewsNet" dataset was used to study the effectiveness of classical machine learning methods and natural language processing models. To train and evaluate the classification effectiveness of the developed system, a version of the dataset converted into a knowledge graph format using RML technology was used [6].

This technique made it possible to correctly analyze various models for solving the problem and evaluate the effectiveness and prospects of the graph approach in comparison with analogous ones.

F1-score was chosen as the main metric for assessing and comparing the effectiveness of the models. The choice is explained by the fact that accuracy is not suitable for assessing the results of the problem being solved, as it can lead to confusion due to class imbalance. Meanwhile, the F1-score is a balanced metric and represents a compromise between precision and recall.

The data were split into training, validation, and test sets. To prevent overfitting, the early stop was implemented that interrupts a training cycle if the value of the error function on the validation set does not decrease within 10 epochs. The best model is preserved through serialization. The training was carried out using the Google Colab platform on GPU. Graphs of the dependence of the value of the error function and F1-score on the epoch number are presented in Figure 8 and Figure 9, respectively.

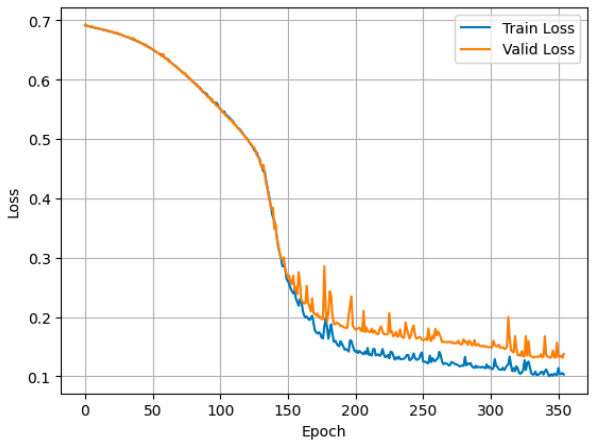


Figure 8. Graph of the dependence of the value of the error function on the epoch number for training and validation samples.

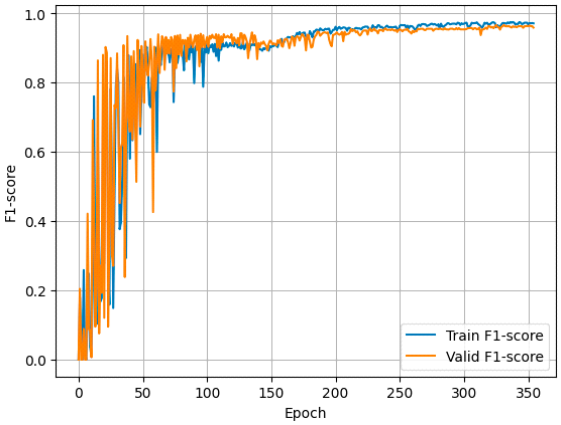


Figure 9. Graph of the dependence of the F1-score value on the epoch number for training and validation samples.

Having studied the graphs, we can conclude that the model was not overfitted and reached a local optimum.

On test data, the model achieved accuracy of 96.8% and F1-score of 95.3%, which are extremely high results [38].

5.3. Comparison with Analogous Models

To assess the results obtained using the developed system, the effectiveness of the classical machine learning models, such as naive Bayes classifier, logistic regression, and support vector machine, was also evaluated. The effectiveness of the DistilBERT language model, which is one of the models of the BERT family of neural network architectures, was also considered.

In the process of evaluation of the models based on classical machine learning methods, a pipeline was constructed that implements a sequence of stages of data transformation (Figure 10). This is necessary due to the fact that in order to work with text in natural language, the data must be represented in a numerical form perceived by the models.

The first step of the pipeline is the CountVectorizer method, which converts the input text into a matrix of the number of occurrences of words in the text. Next, the TfidfTransformer method is used, which allows us to transform numeric vectors obtained using CountVectorizer into vectors that take into account the importance of words in the text. Tfidf stands for Term Frequency-Inverse Document Frequency and is one of the most popular methods for vectorizing text data. TfidfTransformer takes into account both the frequency of occurrence of a word in a document (term frequency) and the inverse document frequency of a word in all documents. This allows to highlight keywords that are most characteristic of a given document and can help in its classification.

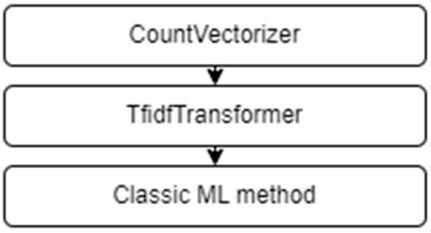


Figure 10. Data transformation pipeline for classic ML models.

The F1-score results for classic machine learning models trained on the "FakeNewsNet" dataset in the tabular format were approximately 60% (Table 3).

Table 3. Evaluation metric results for proposed and existing solutions.

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	85%	63%	65%	64%
SVM	85%	67%	63%	65%
Logistic regression	83%	59%	53%	56%
DistilBERT	84%	79%	86%	82%
GAT	96%	96%	95%	95%

Next, the DistilBERT language model was evaluated (Figure 11). DistilBERT is based on the idea of distillation [45]. DistilBERT language model optimizes the training process by reducing the size and increasing the speed of BERT while maintaining up to 97% performance.

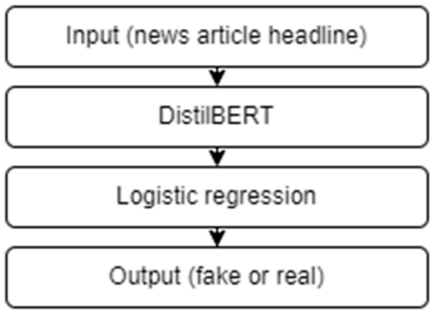


Figure 11. Pipeline for DistilBERT model.

The DistilBERT model was used to obtain features from news headlines for further processing using logistic regression algorithms for binary classification of news into fake and real.

DistilBERT received news article headlines as input, tokenized them, and then extracted features on the GPU and fed them into the next model for processing. The advantage of using BERT family models is in two-way context analysis, which makes it easy to get fairly good results.

For the task of detecting fake news, the model based on using the DistilBERT and logistic regression showed good results and achieved an F1-score of 82% on the test sample.

The values of accuracy and F1-score for the implemented models are presented in Table 3.

Thus, it can be concluded that the use of graph approaches can significantly improve the effectiveness of binary classification of news articles into fake and real. At the same time, the graph neural network with attention (GAT) has additional advantages due to the use of a graph approach and data structure [46,47].

A comparison was also made with the results obtained in a number of recent researches. Models based on the BERT transformer architecture received F1-score results of 75–85% [13–15]. The graph-based model using the BiMGCL architecture achieved an F1-score of 86% [22]. The results of the proposed model are superior to existing solutions (Table 3).

Additionally, a study was conducted to analyze the changes in model performance over time. The proposed model, which is based on the GAT architecture, was compared with classical machine learning models and the DistilBERT language model. These models were trained only on data from the dataset obtained before the end of 2015. Next, F1-score values were calculated for test data from subsequent years after 2015 (Figure 12).

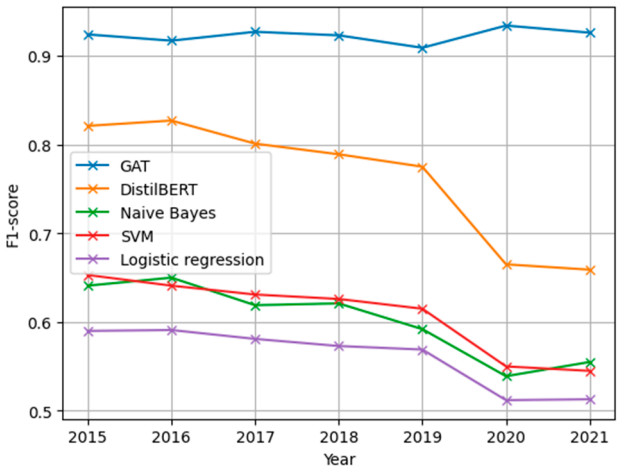


Figure 12. Changes in model performance over time.

Therefore, the GAT model maintains its performance over time compared to models based only on text processing.

6. Conclusions

This work addressed the limitations of existing fake news detection methods, particularly the performance decline of text-based models over time. We demonstrated the power of combining knowledge graphs with a graph attention network (GAT) to create a robust and accurate fake news detection system. Our proposed system achieved state-of-the-art results, significantly outperforming classical machine learning models, DistilBERT, and other graph-based approaches without attention mechanisms. The GAT's ability to weigh the importance of different nodes in the knowledge graph proved crucial in capturing the nuances of news propagation and user influence.

The rigorous evaluation using the FakeNewsNet dataset in both tabular and knowledge graph formats validated the effectiveness and reliability of our approach. Furthermore, the system's consistent performance over time highlights its resilience against evolving textual features and emerging trends. This research contributes a novel and practical solution to the fight against fake news, offering a promising direction for future research in this critical area.

By making our software package open-source, we aim to empower researchers and developers to build upon our findings and create even more effective tools for combating fake news in social networks. Future work could explore incorporating additional contextual information, such as user profiles and temporal dynamics, to further enhance the system's accuracy and adaptability.

Author Contributions: A.G.: Investigation (Lead), Methodology (Lead), Writing—Original Draft (Lead); N.Z.: Conceptualization (Lead), Supervision (Lead); R.D.: Data Analysis (Supporting), Writing—Review and Editing (Supporting); and A.S.: Resources (Supporting), Methodology (Supporting). All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the state budget, project No. FFZF-2025-0008.

Data Availability Statement: The data underlying this study are not publicly available at this time due to ongoing research. However, requests for access to the data will be considered by the authors. Please contact the corresponding authors for data requests.

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

References

1. Shu, K.; Wang, S.; Liu, H. Exploiting Tri-Relationship for Fake News Detection. *arXiv* **2017**, arXiv:1712.07709.
2. Ahmad, I.; Yousaf, M.; Yousaf, S.; Ahmad, M.O. Fake News Detection Using Machine Learning Ensemble Methods. *Complexity* **2020**, *2020*, 8885861. [CrossRef]
3. Wani, A.; Joshi, I.; Khandve, S.; Wagh, V.; Joshi, R. Evaluating Deep Learning Approaches for Covid19 Fake News Detection. *Commun. Comput. Inf. Sci.* **2021**, *1402*, 153–163. [CrossRef]
4. Shu, K.; Sliva, A.; Wang, S.; Tang, J.; Liu, H. Fake News Detection on Social Media. *ACM SIGKDD Explor. Newsl.* **2017**, *19*, 22–36. [CrossRef]
5. Tian, L.; Zhang, X.; Peng, M. FakeFinder: Twitter Fake News Detection on Mobile. In Proceedings of the Companion Proceedings of the Web Conference, Taipei, Taiwan, 20–24 April 2020; pp. 79–80. [CrossRef]
6. Dimou, A.; Vander Sande, M.; Colpaert, P.; Verborgh, R.; Mannens, E.; Van de Walle, R. RML: A generic language for integrated RDF mappings of heterogeneous data. *Ldow* **2014**, *1184*, 1–5.
7. Farooq, M.S.; Naseem, A.; Rustam, F.; Ashraf, I. Fake news detection in Urdu language using machine learning. *PeerJ Comput. Sci.* **2023**, *9*, e1353. [CrossRef]
8. Wasim, M.; Cheema, S.M.; Pires, I.M. Normalized effect size (NES): A novel feature selection model for Urdu fake news classification. *PeerJ Comput. Sci.* **2023**, *9*, e1612. [CrossRef]
9. Ala'raj, M.; Majdalawieh, M.; Abbod, M.F. Improving binary classification using filtering based on k-NN proximity graphs. *J. Big Data* **2020**, *7*, 15.
10. Alkhateem, Y.N.S.; Mejri, M. Auto Encoder Fixed-Target Training Features Extraction Approach for Binary Classification Problems. *Asian J. Res. Comput. Sci.* **2023**, *15*, 32–43. [CrossRef]
11. Kaliyar, R.K.; Goswami, A.; Narang, P. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimed. Tools Appl.* **2021**, *80*, 11765–11788. [CrossRef]
12. BERTology. Hugging Face—The AI Community Building the Future. Available online: https://huggingface.co/docs/transformers/bertology (accessed on 24 December 2024).
13. Raza, S.; Ding, C. Fake news detection based on news content and social contexts: A transformer-based approach. *Int. J. Data Sci. Anal.* **2022**, *13*, 335–362. [CrossRef]
14. Malik, M.S.I.; Imran, T.; Mona Mamdouh, J. How to detect propaganda from social media? Exploitation of semantic and fine-tuned language models. *PeerJ Comput. Sci.* **2023**, *9*, e1248. [CrossRef]
15. Obeidat, R.; Gharaibeh, M.; Abdullah, M.; Alharahsheh, Y. Multi-label multi-class COVID-19 Arabic Twitter dataset with fine-grained misinformation and situational information annotations. *PeerJ Comput. Sci.* **2022**, *8*, e1151. [CrossRef]
16. Faldu, K.; Sheth, A.; Kikani, P.; Akbari, H. KI-BERT: Infusing Knowledge Context for Better Language and Domain Understanding. *arXiv* **2021**, arXiv:2104.08145. [CrossRef]
17. Ostendorff, M.; Bourgonje, P.; Berger, M.; Moreno-Schneider, J.; Rehm, G.; Gipp, B. Enriching BERT with Knowledge Graph Embeddings for Document Classification. In Proceedings of the 15th Conference on Natural Language Processing, Erlangen, Germany, 9–11 October 2019; pp. 1–8.
18. Yao, L.; Mao, C.; Luo, Y. KG-BERT: BERT for knowledge graph completion. *arXiv* **2019**, arXiv:1909.03193.
19. Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; Wang, P. K-BERT: Enabling Language Representation with Knowledge Graph. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), New York, NY, USA, 7–12 February 2020; pp. 2901–2908.
20. Monti, F.; Frasca, F.; Eynard, D.; Mannion, D.; Bronstein, M.M. Fake news detection on social media using geometric deep learning. *arXiv* **2019**, arXiv:1902.06673.
21. Hu, L.; Yang, T.; Zhang, L.; Zhong, W.; Tang, D.; Shi, C.; Duan, N.; Zhou, M. Compare to the knowledge: Graph neural fake news detection with external knowledge. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Virtual Event, 1–6 August 2021; pp. 754–763. [CrossRef]
22. Feng, W.; Li, Y.; Li, B.; Jia, Z.; Chu, Z. BiMGCL: Rumor detection via bi-directional multi-level graph contrastive learning. *PeerJ Comput. Sci.* **2023**, *9*, e1659. [CrossRef]
23. Zhang, L.; Zhang, X.; Zhou, Z.; Huang, F.; Li, C. Reinforced adaptive knowledge learning for multimodal fake news detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; Volume 38. [CrossRef]
24. Gao, X.; Wang, X.; Chen, Z.; Zhou, W.; Hoi, S.C.H. Knowledge enhanced vision and language model for multi-modal fake news detection. *IEEE Trans. Multimed.* **2024**, *26*, 8312–8322. [CrossRef]
25. Che, H.; Pan, B.; Leung, M.-F.; Cao, Y.; Yan, Z. Tensor Factorization with Sparse and Graph Regularization for Fake News Detection on Social Networks. *IEEE Trans. Comput. Soc. Syst.* **2024**, *11*, 4888–4898. [CrossRef]

26. Chatzianastasis, M.; Lutzeyer, J.F.; Dasoulas, G.; Vazirgiannis, M. Graph Ordering Attention Networks. In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 7006–7014.
27. Zhou, Y.; Zheng, H.; Huang, X.; Hao, S.; Li, D.; Zhao, J. Graph Neural Networks: Taxonomy, Advances, and Trends. *ACM Trans. Intell. Syst. Technol.* **2022**, *13*, 1–54. [CrossRef]
28. Knyazev, B.; Tailor, G.W.; Amer, M.R. Understanding Attention and Generalization in Graph Neural Networks. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
29. Xu, J.; Li, Z.; Du, B.; Zhang, M.; Liu, J. Reluplex made more practical: Leaky ReLU. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–7.
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762v5.
31. Brody, S.; Alon, U.; Yahav, E. How Attentive are Graph Attention Networks? *arXiv* **2022**, arXiv:2105.14491v3.
32. A1gord/FakeNewsDetection. Available online: https://github.com/A1gord/FakeNewsDetection (accessed on 15 October 2024).
33. NumPy v1.24 Manual. Available online: https://numpy.org/doc/stable/ (accessed on 15 October 2024).
34. pandas—Python Data Analysis Library. Available online: https://pandas.pydata.org/ (accessed on 15 October 2024).
35. Matplotlib 3.7.1 Documentation. Available online: https://matplotlib.org/stable/contents.html (accessed on 15 October 2024).
36. PyTorch Documentation. Available online: https://pytorch.org/docs/stable/index.html (accessed on 15 October 2024).
37. PyG Documentation. Available online: https://pytorch-geometric.readthedocs.io/en/latest/ (accessed on 15 October 2024).
38. Shu, K. FakeNewsNet: A Data Repository with News Content, Social Context and Spatiotemporal Information for Studying Fake News on Social Media. *Big Data* **2020**, *8*, 171–188.
39. Li, Q.; Zhou, W. Connecting the Dots Between Fact Verification and Fake News Detection. *arXiv* **2022**, arXiv:2010.05202v1. [CrossRef]
40. Gossip Cop. Available online: http://gossipcop.com/ (accessed on 24 September 2024).
41. PolitiFact. Available online: https://www.politifact.com/ (accessed on 24 September 2024).
42. D'Ulizia, A.; Caschera, M.C.; Ferri, F.; Grifoni, P. Fake news detection: A survey of evaluation datasets. *PeerJ Comput. Sci.* **2021**, *7*, e518. [CrossRef]
43. Fake News. Kaggle. Available online: https://www.kaggle.com/datasets/algord/fake-news (accessed on 24 September 2024).
44. Dou, Y. User Preference-aware Fake News Detection. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 11–15 July 2021.
45. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108. [CrossRef]
46. Song, C.; Shu, K.; Wu, B. Temporally evolving graph neural network for fake news detection. *Inf. Process. Manag.* **2021**, *58*, 102712.
47. Matsumoto, H.; Yoshida, S.; Muneyasu, M. Propagation-Based Fake News Detection Using Graph Neural Networks with Transformer. In Proceedings of the 2021 IEEE 10th Global Conference on Consumer Electronics, Kyoto, Japan, 12–15 October 2021; pp. 19–20.

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.