International Journal of Data Science and Analytics (2022) 13:335–362
https://doi.org/10.1007/s41060-021-00302-z

REGULAR PAPER

Fake news detection based on news content and social contexts: a transformer-based approach

Shaina Raza[1] · **Chen Ding**[1]

Received: 24 April 2021 / Accepted: 13 December 2021 / Published online: 30 January 2022

Abstract
Fake news is a real problem in today's world, and it has become more extensive and harder to identify. A major challenge in fake news detection is to detect it in the early phase. Another challenge in fake news detection is the unavailability or the shortage of labelled data for training the detection models. We propose a novel fake news detection framework that can address these challenges. Our proposed framework exploits the information from the news articles and the social contexts to detect fake news. The proposed model is based on a Transformer architecture, which has two parts: the encoder part to learn useful representations from the fake news data and the decoder part that predicts the future behaviour based on past observations. We also incorporate many features from the news content and social contexts into our model to help us classify the news better. In addition, we propose an effective labelling technique to address the label shortage problem. Experimental results on real-world data show that our model can detect fake news with higher accuracy within a few minutes after it propagates (early detection) than the baselines.

Keywords Fake news · Social contexts · Concept drift · Weak supervision · Transformer · User credibility · Zero shot learning

1 Introduction

Fake news detection is a subtask of text classification [1] and is often defined as the task of classifying news as real or fake. The term 'fake news' refers to the false or misleading information that appears as real news. It aims to deceive or mislead people. Fake news comes in many forms, such as clickbait (misleading headlines), disinformation (with malicious intention to mislead the public), misinformation (false information regardless of the motive behind), hoax, parody, satire, rumour, deceptive news and other forms as discussed in the literature [2].

Fake news is not a new topic; however, it has become a hot topic since the 2016 US election. Traditionally, people get news from trusted sources, media outlets and editors, usually following a strict code of practice. In the late twentieth century, the internet has provided a new way to consume, publish and share information with little or no editorial standards. Lately, social media has become a significant source of news for many people. According to a report by Statistica,[1] there are around 3.6 billion social media users (about half the population) in the world. There are obvious benefits of social media sites and networks in news dissemination, such as instantaneous access to information, free distribution, no time limit, and variety. However, these platforms are largely unregulated. Therefore, it is often difficult to tell whether some news is real or fake.

Recent studies [2–4] show that the speed at which fake news travels is unprecedented, and the outcome is its wide-scale proliferation. A clear example of this is the spread of anti-vaccination misinformation[2] and the rumour that incorrectly compared the number of registered voters in 2018 to the number of votes cast in US Elections 2020.[3] The implications of such news are seen during the anti-vaccine movements that prevented the global fight against COVID-19 or in post-

✉ Shaina Raza
Shaina.raza@ryerson.ca

Chen Ding
cding@ryerson.ca

[1] Ryerson University, Toronto, ON, Canada

[1] https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/.

[2] https://www.wrcbtv.com/story/43076383/first-doses-of-covid19-vaccines-administered-at-chattanooga-hospital-on-thursday.

[3] https://archive.is/OXJ60.

election unrest. Therefore, it is critically important to stop the spread of fake news at an early stage.

A significant research gap in the current state-of-the-art is that it focuses primarily on fake news detection rather than early fake news detection. The seminal works [4, 5] on early detection of fake news usually detect the fake news after at least 12 h of news propagation, which may be too late [6]. An effective model should be able to detect fake news early, which is the motivation of this research.

Another issue that we want to highlight here is the scarcity of labelled fake news data (news labelled as real or fake) in real-world scenarios. Existing state-of-the-art works [4, 7, 8] generally use fully labelled data to classify fake news. However, the real-world data is likely to be largely unlabelled [5]. Considering the practical constraints, such as unavailability of the domain experts for labelling, cost of manual labelling, and difficulty of choosing a proper label for each news item, we need to find an effective way to train a large-scale model. One alternative approach is to leverage noisy, limited, or imprecise sources to supervise labelling of large amounts of training data. The idea is that the training labels may be imprecise and partial but can be used to create a strong predictive model. This scheme of training labels is the weak supervision technique [9].

Usually, the fake news detection methods are trained on the current data (available during that time), which may not generalize to future events. Many of the labelled samples from the verified fake news get outdated soon with the newly developed events. For example, a model trained on fake news data before the COVID-19 may not classify fake news properly during COVID-19. The problem of dealing with a target concept (e.g. news as 'real' or 'fake') when the underlying relationship between the input data and target variable changes over time is called concept drift [10]. In this paper, we investigate whether concept drift affects the performance of our detection model, and if so, how we can mitigate them.

This paper addresses the challenges mentioned above (early fake news detection and scarcity of labelled data) to identify fake news. We propose a novel framework based on a deep neural network architecture for fake news detection. The existing works, in this regard, rely on the content of news [7, 11, 12], social contexts [1, 4, 5, 8, 13, 14], or both [4, 8, 15]. We include a broader set of news-related features and social context features compared to the previous works. We try to detect fake news early (i.e. after a few minutes of news propagation). We address the label shortage problem that happens in real-world scenarios. Furthermore, our model can combat concept drift.

Inspired by the bidirectional and autoregressive Transformer (BART) [16] model from Facebook that is successfully used in language modelling tasks, we propose to apply a deep bidirectional encoder and a left-to-right decoder under the hood of one unified model for the task of fake news detection. We choose to work with the BART model over the state-of-the-art BERT model [17], which has demonstrated its abilities in NLP (natural language processing) tasks (e.g. question answering and language inference), as well as the GPT-2 model [18], which has impressive autoregressive (time-series) properties. The main reason is that the BART model combines the unique features (bidirectional and autoregressive) of both text generation and temporal modelling, which we require to meet our goals.

Though we take inspiration from BART, our model is different from the original BART in the following aspects: (1) in comparison with the original BART, which takes a single sentence/document as input, we incorporate a rich set of features (from news content and social contexts) into the encoder part; (2) we use a decoder to get predictions not only from previous text sequences (in this case, news articles) as in the original BART but also from previous user behaviour (how users respond to those articles) sequences, and we detect fake news early by temporally modelling user behaviour; (3) on top of the original BART model, we add a single linear layer to classify news as fake or real.

Our contributions are summarized as follows:

1. We propose a novel framework that exploits news content and social contexts to learn useful representations for predicting fake news. Our model is based on a Transformer [19] architecture, which facilitates representation learning from fake news data and helps us detect fake news early. We also use the side information (metadata) from the news content and the social contexts to support our model to classify the truth better.
2. We present a systematic approach to investigate the relationship between the user profile and news veracity. We propose a novel Transformer-based model using zero-shot learning [20] to determine the credibility levels of the users. The advantage of our approach is that it can determine the credibility of both long-term and new users, and it can detect the malicious users who often change their tactics to come back to the system or vulnerable users who spread misinformation.
3. We propose a novel weak supervision model to label the news articles. The proposed model is an effective labelling technique that lessens the burden of extensive labelling tasks. With this approach, the labels can be extracted instantaneously from known sources and updated in real-time.

We evaluate our system by conducting experiments on real-world datasets: (i) NELA-GT-19 [21] that consists of news articles from multiple sources and (ii) Fakeddit [22] that is a multi-modal dataset containing text and images in posts on the social media website Reddit. While the social contexts used in this model are from Reddit, consisting of

upvotes, downvotes, and comments on posts, the same model can be generalized to fit other social media datasets. The same method is also generalizable for any other news dataset. The results show that our proposed model can detect fake news earlier and more accurately than baselines.

The rest of the paper is organized as follows. Section 2 is the related work. Section 3 discusses the proposed framework. Section 4 explains the details of our fake news detection model, Sect. 5 describes the experimental set-up, and Sect. 6 shows the results and analyses. Finally, Sect. 7 is about the limitations, and Sect. 8 gives the conclusion and lists the future directions.

2 Literature review

Fake news is information that is false or misleading and is presented as real news[23]. The term 'fake news' became mainstream during the 2016 presidential elections in United States. Following this, Google, Twitter, Facebook took steps to combat fake news. However, due to the exponential growth of information in online news portals and social media sites, distinguishing between real and fake news has become difficult.

In the state-of-the-art, the fake news detection methods are categorized into two types: (1) manual fact-checking; (2) automatic detection methods. Fact-checking websites, such as Reporterslab,[4] Politifact[5] and others [2], rely on human judgement to decide the truthfulness of some news. Crowdsourcing, e.g. Amazon's Mechanical Turk,[6] is also used for detecting fake news in online social networks. These fact-checking methods provide the ground truth (true/false labels) to determine the veracity of news. The manual fact-checking methods have some limitations: 1) it is time-consuming to detect and report every fake news produced on the internet; 2) it is challenging to scale well with the bulks of newly created news, especially on social media; 3) it is quite possible that the fact-checkers' biases (such as gender, race, prejudices) may affect the ground truth label.

The automatic detection methods are alternative to the manual fact-checking ones, which are widely used to detect the veracity of the news. In the previous research, the characteristics of fake news are usually extracted from the news-related features (e.g. news content) [21] or from the social contexts (social engagements of the users) [4, 22, 24] using automatic detection methods.

The content-based methods [25–28] use various types of information from the news, such as article content, news source, headline, image/video, to build fake news detection classifiers. Most content-based methods use stylometry features (e.g. sentence segmentation, tokenization, and POS tagging) and linguistic features (e.g. lexical features, bag-of-words, frequency of words, case schemes) of the news articles to capture deceptive cues or writing styles. For example, Horne and Adalı [29] extract stylometry and psychological features from the news titles to differentiate fake news from real. Przybyla et al. [26] develop a style-based text classifier, in which they use bidirectional Long short-term memory (LSTM) to capture style-based features from the news articles. Zellers et al. [12] develop a neural network model to determine the veracity of news from the news text. Some other works [27, 30] consider lexicons, bag-of-words, syntax, part-of-speech, context-free grammar, TFIDF, latent topics to extract the content-based features from news articles.

A general challenge of content-based methods is that fake news's style, platform, and topics keep changing. Models that are trained on one dataset may perform poorly on a new dataset with different content, style, or language. Furthermore, the target variables in fake news change over time, and some labels become obsolete, while others need to be re-labelled. Most content-based methods are not adaptable to these changes, which necessitates re-extracting news features and re-labelling data based on new features. These methods also require a large amount of training data to detect fake news. By the time these methods collect enough data, fake news has spread too far. Because the linguistic features used in content-based methods are mostly language-specific, their generality is also limited.

To address the shortcomings of content-based methods, a significant body of research has begun to focus on social contexts to detect fake news. The social context-based detection methods examine users' social interactions and extract relevant features representing the users' posts (review/post, comments, replies) and network aspects (followers–followee relationships) from social media. For example, Liu and Wu [5] propose a neural network classifier that uses social media tweets, retweet sequences, and Twitter user profiles to determine the veracity of the news.

The existing social contexts-based approaches are categorized into two types: (1) stance-based methods and (2) propagation-based methods. The stance-based approaches exploit the users' viewpoints from social media posts to determine the truth. The users express the stances either explicitly or implicitly. The explicit stances are the direct expressions of users' opinions usually available from their reactions on social media. Previous works [4, 5, 22] mostly use upvotes/downvotes, thumbs up/down to extract explicit stances. The implicit stance-based methods [5, 31], on the other hand, are usually based on extracting linguistic features from social media posts. To learn the latent stances from

[4] https://reporterslab.org/fact-checking/.

[5] https://www.politifact.com/.

[6] https://www.mturk.com/.

topics, some studies [11] use topic modelling. Other studies [13, 32] look at fake users' accounts and behaviours to see if they can detect implicit stances. A recent study also analyses users' views on COVID-19 by focusing on people who interact and share information on Twitter [33]. This study provides an opportunity to assess early information flows on social media. Other related studies [34, 35] examine users' feelings about fake news on social media and discover a link between sentiment analysis and fake news detection.

The propagation-based methods [36–39] utilize information related to fake news, e.g. how users spread it. In general, the input to a propagation-based method can be either a news cascade (direct representation of news propagation) or self-defined graph (indirect representation capturing information on news propagation) [2]. Hence, these methods use graphs and multi-dimensional points for fake news detection [36, 39]. The research in propagation-based methods is still in its early stages.

To conclude, social media contexts, malicious user profiles, and user activities can be used to identify fake news. However, these approaches pose additional challenges. Gathering social contexts, for example, is a broad subject. The data is not only big, but also incomplete, noisy, and unstructured, which may render existing detection algorithms ineffective.

Other than NLP methods, visual information is also used as a supplement to determine the veracity of the news. A few studies investigate the relationship between images and tweet credibility [40]. However, the visual information in this work [40] is hand-crafted, limiting its ability to extract complex visual information from the data. In capturing automatic visual information from data, Jin et al. [41] propose a deep neural network approach to combine high-level visual features with textual and social contexts automatically.

Recently, transfer learning has been applied to detect fake news [1, 7]. Although transfer learning has shown promising results in image processing and NLP tasks, its application in fake news detection is still under-explored. This is because fake news detection is a delicate task in which transfer learning must deal with semantics, hidden meanings, and contexts from fake news data. In this paper, we propose a transfer learning-based scheme, and we pay careful attention to the syntax, semantics and meanings in fake news data.

2.1 State-of-the-art fake news detection models

In one of earlier works, Karimi et al. [42] use convolutional neural network (CNN) and LSTM methods to combine various text-based features, such as those from statements (claims) related to news data. Liu et al. [39] also use RNN and CNN-based methods to build propagation paths for detecting fake news at the early stage of its propagation. Shu et al. [4] propose a matrix factorization method TriFN to model the relationships among the publishers, news stories and social media users for fake news detection.

Cui et al. [12] propose an explainable fake news detection system DEFEND based on LSTM networks. The DEFEND considers users' comments to explain if some news is real or fake. Nguyen et al. [15] propose a fake news detection method FANG that uses the graph learning framework to learn the representations of social contexts. These methods discussed above are regarded as benchmark standards in the field of fake news research.

In recent years, there has been a greater focus in NLP research on pre-trained models. BERT [17] and GPT-2 [43] are two state-of-the-art pre-trained language models. In the first stage, the language model (e.g. BERT or GPT-2) is pre-trained on the unlabelled text to absorb maximum amount of knowledge from data (unsupervised learning). In the second stage, the model is fine-tuned on specific tasks using a small-labelled dataset. It is a semi-supervised sequence learning task. These models are also used in fake news research.

BERT is used in some fake news detection models [1, 7, 44] to classify news as real or fake. BERT uses bidirectional representations to learn information and is generally more suitable for NLP tasks, such as text classification and translation. The GPT-2, on the other hand, uses the unidirectional representation to predict the future using left-to-right context and is better suited for autoregressive tasks, where timeliness is a crucial factor. In related work, Zellers et al. [12] propose a Grover framework for the task of fake news detection, which uses a language model close to the architecture of GPT-2 trained on millions of news articles. Despite these models' robust designs, there are a few research gaps. First, these models do not consider a broader set of features from news and social contexts. Second, these methods ignore the issue of label scarcity in real-world scenarios. Finally, the emphasis is not on early fake news detection.

The state-of-the-art focuses primarily on fake news detection methods rather than early fake news detection. A few works [4, 5] propose early detection of fake news. However, to detect fake news, these methods [4, 5] usually rely on a large amount of fake news data observed over a long period of time (depending upon the availability of the social contexts). The work in [4] detects fake news after at least 12 h of news propagation, as demonstrated in their experiments, which may be too late. According to research [6], the fake news spreads within minutes once planted. For example, the fake news that Elon Musk's Tesla team is inviting people to give them any amount (ranging from 0.1 to 20) of bitcoins in exchange for double the amount resulted in a loss of millions of dollars within the first few minutes.[7] Therefore, it is critical to detect fake news early on before it spreads.

[7] https://www.bbc.com/news/technology-56402378.

Our work is intended to address these issues (early fake news detection, labels scarcity) in fake news research. BERT and GPT-2 (or similar) have not been used to their full potential for representation learning and autoregression tasks in a single unifying model that we intend to work on going forward in our research. We propose a combination of Transformer architectures that can be applied to a wide range of scenarios, languages, and platforms.

3 Overview of the proposed framework

3.1 Problem definition

Given a multi-source news dataset and social contexts of news consumers (social media users), the task of fake news detection is to determine if a news item is fake or real. Formally, we define the problem of fake news detection as:

- Input: News items, social contexts and associated side information
- Output: One of two labels: 'fake' or 'real'.

3.2 Proposed architecture

Figure 1 shows an overview of our proposed framework. Initially, the news comes from the news ecosystem [45], which we refer to as the dataset (input) in this work. The news content and social contexts go into the respective components where the data is being preprocessed. The input to the embedding layer is the features from news content and social contexts. The output from the embedding layer is the vector representations of news content and social contexts. These vector representations are combined to produce a single representation that is passed as input to the Transformer block. The output from the Transformer is transferred to the classification layer, followed by the cross-entropy layer. We get a label (fake or real) for each news as the final output.

We utilize three types of embeddings in the embedding layer: (1) token embeddings: to transform words into vector representations; (2) segment embeddings: to distinguish different segments or sentences from the content; (3) positional embeddings: to show tokens' positions within sequences.

We create sequences from the news content and social contexts (user behaviours). In our work, we maintain a temporal order in the sequences through positional encodings. The intuition is that each word in the sequence is temporally arranged and is assigned to a timestep, which means that the first few words correspond to the timestep 0, timestep 1, and so on, till the last word corresponding to the last timestep. We use the sinusoidal property of position encodings in sequences [46], where the distance between the neighbouring timesteps is symmetrical and decays over time.

We discuss the Transformer block that consists of the encoder, decoder and attention mechanism in detail in Sect. 4 and Fig. 3.

3.3 The news ecosystem

The news ecosystem consists of three basic entities: publishers (news media or editorial companies that publish the news article), information (news content) and users [4, 45, 47]. As shown in Fig. 1, initially, the news comes from publishers. Then, it goes to different websites or online news platforms. The users get news from these sources, sharing news on different platforms (blogs, social media). The multiple friends' connections, followers–followees links, hashtags, and bots make up a social media network.

3.3.1 News content

The news content component takes news content from the news ecosystem. The news body (content) and the corresponding side information represent a news article. The news body is the main text that elaborates the news story; generally, the way a news story is written reflects an author's main argument and viewpoint. We include the following side information related to news:

- *Source* The source of news (e.g. CNN, BBC).
- *Headline* The title text that describes the main topic of the article. Usually, the headlines are designed to catch the attention of readers.
- *Author* The author of the news article.
- *Publication time* The time when news is published; it is an indicator of recency or lateness of news.
- *Partisan information* This information is the adherence of a news source to a particular party. For example, a news source with many articles favouring the right-wing reflects the source' and authors' partisan bias.

3.3.2 Social contexts

The social contexts component takes the social contexts on the news, such as posts, likes, shares, replies, followers–followees and their activities. When the features related to the content of news articles are not enough or available, social contexts can provide helpful information on fake news. Each social context is represented by a post (comment, review, reply) and the corresponding side information (metadata). The post is a social media object posted by a user; it contains useful information to understand a user's view on a news article. We include the following side information related to social contexts:

- *User* A person or bot that registers on social media.

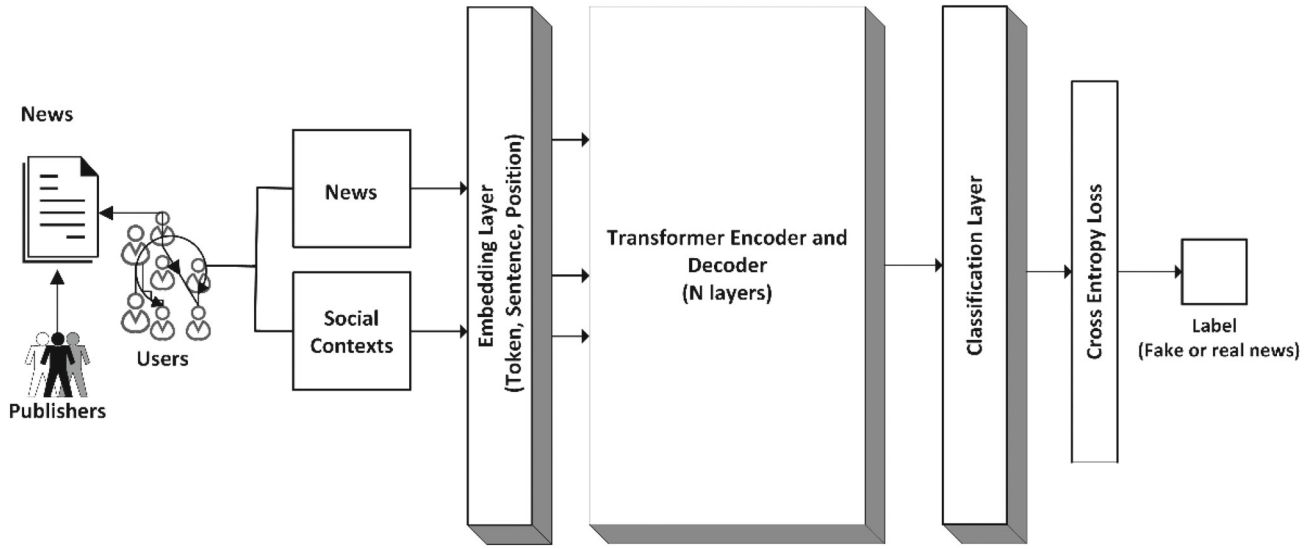

Fig. 1 Overview of the proposed framework

- *Title* The headline or short explanation of the post. The title of the post matches the news headline.
- *Score* A numeric score given to a post by another user; this feature determines whether another user approves or disapproves of the post.
- *Source* The source of news.
- *Number of comments* The count of comments on a post; this feature gives the popularity level of a post.
- *Upvote–Downvote ratio* An estimate of other users' approval/disapproval of a post.
- *Crowd (aggregate) response* We calculate the aggregate responses of all users on each news article. To calculate the aggregate response, we take all the scores on a post to determine a user's overall view of a news story. We assume that a news story or theme with a score less than 1 is not reliable and vice versa.
- *User credibility* We determine the credibility level of social media users as an additional social context feature. This feature is helpful to determine if a user tends to spread some fake news or not. For example, similar posts by a non-credible user on a news item is an indicator of a news being real or fake. We determine user credibility through a user credibility component, shown in Fig. 2 and discussed next.

3.4 User credibility module

The topic of determining the credibility of social media users is not new in the literature. Some previous works apply community detection [48], sentiment analysis [33] and profile ranking techniques [49]. However, there is not much work in the fake news detection that considers the user credibility of social media users. The seminal work [4], in this regard, is a simple clustering approach that assigns a credibility level to each user. We adopt a different approach to build the user credibility module, as shown in Fig. 2.

We use zero-shot learning (ZSL) [20] to determine the credibility of users. ZSL is a mechanism by which a computer program learns to recognize objects in an image or extract information from text without labelled training data. For example, a common approach to classifying news categories is training a model from scratch on task-specific data. However, ZSL enables this task to be completed without any previous task-specific training. ZSL can also detect and predict unknown classes that a model has never seen during training based on prior knowledge from the source domain [43, 51] or auxiliary information.

To determine the credibility level, we first group each user's engagements (comments, posts, replies) and then feed this information into our ZSL classifier. We build our ZSL classifier based on the Transformer architecture. We attach the pre-trained checkpoint[8] (weights of the model during the training phase) of a huge dataset: multi-genre natural language inference (MNLI) [50], with our classifier.

A multi-sourced collection is frequently used to collect information or opinions from large groups of people who submit their data through social media or the internet. We are using MNLI because it is a large-scale crowd-sourced dataset that covers a range of genres of spoken and written text. User credibility and crowdsourcing have been linked in previous research [55, 56]. Therefore, we anticipate that a large amount of crowdsourced data in MNLI could reveal a link between users' credibility and how they express their opinions. It would be expensive if we need to gather such

[8] https://dl.fbaipublicfiles.com/fairseq/models/bart.large.mnli.tar.gz.

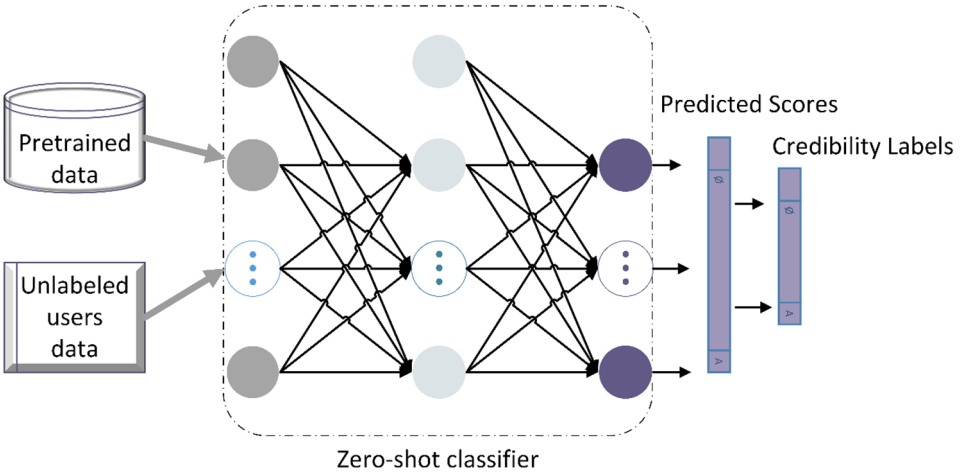

Fig. 2 The user credibility module

crowd-sourced opinions as well as direct user feedbacks ourselves. We gain the benefits of a pre-trained model in terms of size and training time and the benefit of accuracy by using MNLI.

Through ZSL, the checkpoint that is pre-trained can be fine-tuned for a specialized task, e.g. the user credibility task in our work. We could classify the users into different unseen classes (user credibility levels). In total, we define five credibility levels: 'New user', 'Very uncredible', 'Uncredible', 'Credible', 'Very credible'. We use the prior knowledge of a fine-tuned ZSL model and its checkpoint, and we also use the semantics of the auxiliary information to determine known user classes. Our model can also determine new unknown user classes. Later, we incorporate this information as the weak labels into our fake news detection model.

Another module in this framework is the weak supervision module that is related to our datasets and labelling scheme, so we will discuss it in the dataset section (Sect. 6.4).

4 Proposed method

4.1 Preliminaries

Let $N = \{n_1, n_2, \ldots, n_{|N|}\}$ be a set of news items, each of which is labelled as $y_i \in \{0, 1\}$, $y_i = 1$ is the fake news and $y_i = 0$ is the real news. The news item n_i is represented by its news body (content) and the side information (headline, body, source, etc.). When a news item n_i is posted on social media, it is usually responded to by a number of social media users $U = \{u_1, u_2, \ldots, u_{|U|}\}$. The social contexts include users' social engagements (interactions), such as comments on news, posts, replies, and upvotes/downvotes.

We define social contexts on a news item n_i as: $\mathrm{SC}(n_i) = \big((u_1, \mathrm{sc}_1, t_1), (u_2, \mathrm{sc}_2, t_2), \ldots, (u_{|\mathrm{sc}|}, \mathrm{sc}_{|\mathrm{sc}|}, t_{|\mathrm{sc}|})\big)$, where each tuple (u, sc, t) refers to a user u's social contexts sc on a news item n_i during time t. Here, a user may interact with a post multiple times, and each interaction is recorded with its timestamp.

The task of fake news detection is to find a model M that predicts a label $\hat{y}(n_i) \in \{0, 1\}$ for each news item based on its news content and the corresponding social contexts. Therefore, the task of fake news detection, in this paper, is defined as shown in Eq. (1):

$$\hat{y}(n_i) = M(C(n_i), \mathrm{SC}(n_i)) \tag{1}$$

where $C(n_i)$ refers to the content of news and $\mathrm{SC}(n_i)$ refers to the social contexts on the news. The notations used in this paper can be found in "Appendix A".

4.2 Proposed model: FND-NS

Here, we introduce our proposed classification model called FND-NS (Fake News Detection through News content and Social context), which adapts the bidirectional and autoregressive Transformers (BART) for a new task—fake news detection, as shown in Fig. 3. The original BART [16] is a denoising autoencoder that is trained in two steps. It first corrupts the text with an arbitrary noising function, and then it learns a model to reconstruct the original text. We use the BART as sequence-to-sequence Transformer with a bidirectional encoder (like BERT) and a left-to-right autoregressive decoder (like GPT-2).

Models such as BERT [17], which captures the text representations in both directions, and GPT-2 [18], which has autoregressive properties, are examples of self-supervised methods. Both are Transformer models with their strengths: BERT excels in discriminative tasks (identifying existing data to classify) but is limited in generative tasks. At the same time, GPT-2 is capable of generative tasks (learning the regularities in data to generate new samples) but not discriminative tasks due to its autoregressive properties. In comparison with these models, BART integrates text

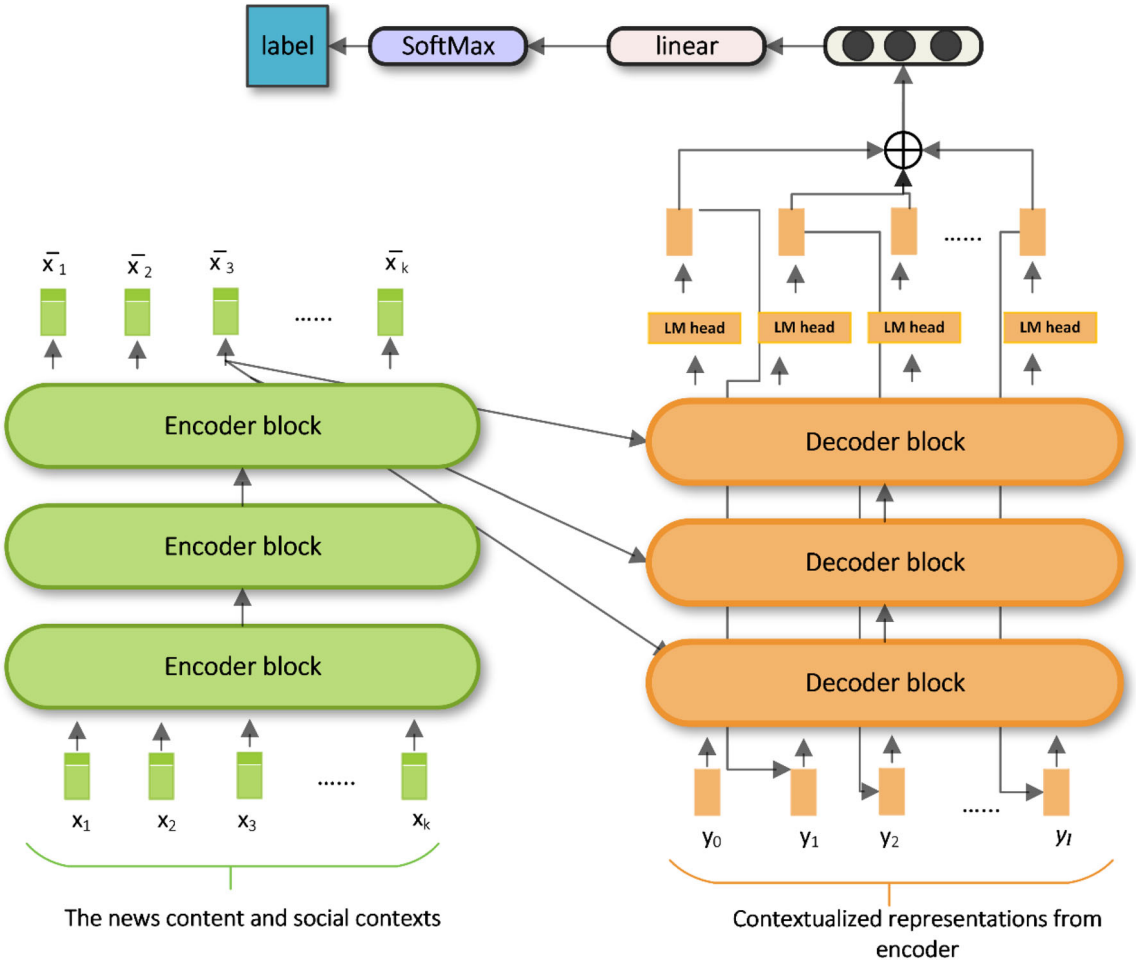

Fig. 3 The encoder and decoder blocks in FND-NS model

generation and comprehension using both bidirectional and autoregressive Transformers. Due to this reason, we choose to work on the BART architecture.

Though we get inspiration from BART, our network architecture is different from the original BART in the following manner. The first difference between our model and the original BART is the method of input. Original BART takes one piece of text as input in the encoder part. In contrast, we incorporate a rich set of features (from news content and social contexts) into the encoder part. We use multi-head attentions to weigh the importance of different pieces of information. For example, if the headline is more convincing in persuading readers to believe something, or if a post receives an exceptionally large number of interactions, we pay closer attention to such information. We have modified the data loader of the original BART to feed more information into our neural network architecture.

The second difference is the way the next token is predicted. By token, we mean the token-level tasks such as named entity recognition and question answering, where models are required to produce fine-grained output at the token level [52]. We randomly mask some tokens in the input sequence. We follow the default masking probability, i.e. masking 15% of tokens with (MASK), as in the original paper [16]. However, we predict the ids of those masked items based on the positions of missing inputs from the sequence. This way, we determine the next item in the sequence based on its temporal position. In our work, we use the decoder to make predictions based on the previous sequences of text (news articles) and the previous sequences of user behaviours (how users respond to those articles). Modelling user behaviours in such a temporal manner helps us detect fake news in the early stage.

Finally, different from the original BART, we add a linear transformation and SoftMax layer to output the final target label.

Next, we discuss our model (Fig. 3) and explain how we use it in fake news detection. Let N represents a set of news items. Each news item has a set of text and social context features. These features are merged to form a combined feature set, as shown in the flowchart in Fig. 4.

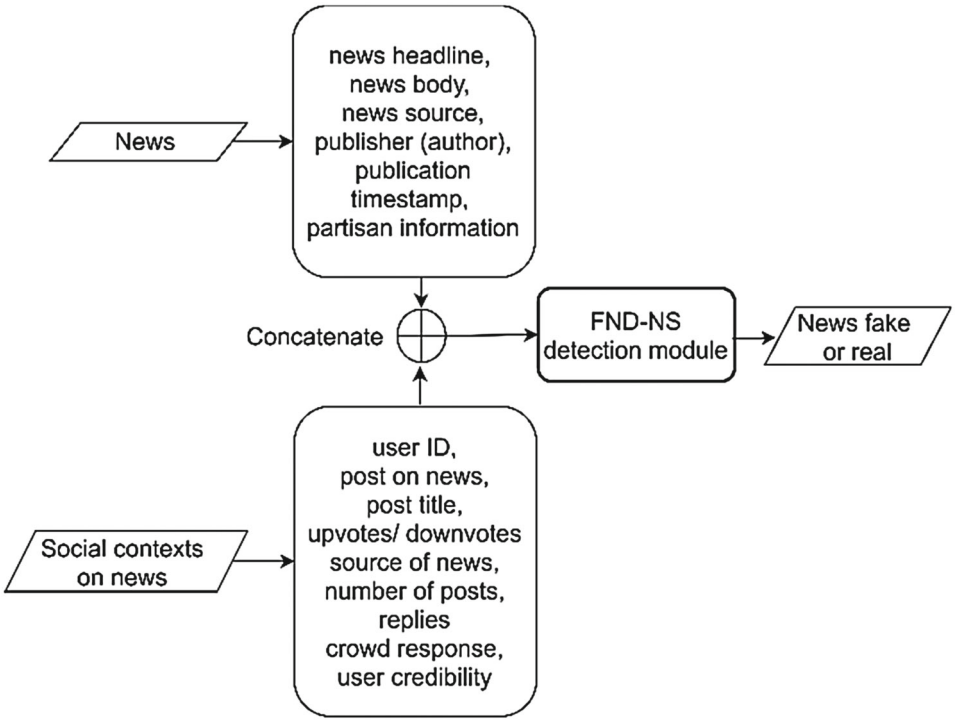

Fig. 4 Flowchart of proposed FND-NS model

These combined features are then encoded into a vector representation. Let X represent a sequence of k combined features for a news item, as shown in Eq. (2):

$$X = \{x_1, x_2, \ldots, x_k\}. \tag{2}$$

These features are given as input to the embedding layers. The embedding layer gives us a word embedding vector for each word (feature in our case) in the combined feature set. We also add a positional encoding vector with each word embedding vector. The word embedding vector gives us the (semantic) information for each word. The positional encoding describes the position of a token (word) in a sequence. Together they give us the semantic as well as temporal information for each word. We define this sequence of embedding vectors as $X' = \{x'_1, x'_2, \ldots, x'_k\}$.

In the sequence-to-sequence problem, we find a mapping f from an input sequence of k vectors $X'_{1:k}$ to a sequence of l target vectors $Y_{1:l}$. The number of target vectors is unknown a priori and depends on the input sequence. The f is shown in Eq. (3):

$$f : X'_{1:k} \rightarrow Y_{1:l}. \tag{3}$$

4.2.1 Encoder

The encoder is a stack of encoder blocks, as shown in green in Fig. 3. The encoder maps the input sequence to a contextualized encoding sequence. We use the bidirectional encoder to encode the input from both directions to get the contextualized information. The input to the encoder is the input sequence $X'_{1:k}$. The encoder maps the input sequence X' to a contextualized encoding sequence $\overline{X}$, as shown in Eq. (4):

$$f_{\theta_{\text{enc}}} : \left(X'_{1:k} \rightarrow \overline{X}_{1:k}\right). \tag{4}$$

The first encoder block transforms each context-independent input vector to a context-dependent vector representation. The next encoder blocks further refine the contextualized representation until the last encoder block outputs final contextualized encoding $\overline{X}_{1:k}$. Each encoder block consists of a bidirectional self-attention layer, followed by two feed-forward layers. We skip the details of feed-forward layers, which are the same as in [17]. We focus more on the bidirectional self-attention layer that we apply to the given inputs.

The bidirectional self-attention layer takes the vector representation $x'_i \in X'_{1:k}$ as the input. Each input vector x'_i in the encoder, block is projected to a key vector $\kappa_i \in \mathcal{K}_{1:k}$, value vector $v_i \in V_{1:k}$, and a query vector $q_i \in Q_{1:k}$, through three trainable weight matrices W_q, W_v, W_k, as shown in Eq. (5)

$$q_i = W_q x'_i;\ v_i = W_v x'_i;\ \kappa_i = W_\kappa x'_i \tag{5}$$

where $\forall i \in \{1, 2, \ldots, k\}$. The same weight matrices are applied to each input vector x'_i. After projecting each input vector x'_i to a query, key and value vector, each query vector is compared to all the key vectors. The intuition is that the higher the similarity between a key vector and a query

vector, the more important is the corresponding value for the output vector. The output from the self-attention layer is the output vector representation x_i'', which is a refined contextualized representation of x_i'. An output vector x'' is defined as the weighted sum of all value vectors V plus the input vector x'. The weights are proportional to the cosine similarity between the query vectors and respective key vectors, shown in Eq. (6):

$$X''_{1:k} = V_{1:k}\ \text{SoftMax}\left(Q^T_{1:k}\mathcal{K}_{1:k}\right) + X'_{1:k} \tag{6}$$

here X'' is the sequence of output vectors generated from the input X'. X'' is given to the last encoder block, and the output from the last encoder is a sequence of encoder hidden states $\overline{X}$. The final output from the encoder is the contextualized encoded sequence $\overline{X}_{1:k}$, which is passed to the decoder.

4.2.2 Decoder

The decoder only models on the leftward context, so it does not learn bidirectional interactions. Generally, the news (either real or fake) is shown or read in the order of publication timestamps. So, news reading is a left-to-right (backward-to-forward) process. Naturally, the timestamps of users' engagements also follow the order of the news. In our work, we model the left-to-right interdependencies in the sequences through the decoder part. The recurrent structure inside the decoder helps us use the predictions from a previous state to generate the next state. With autoregressive modelling, we can detect fake news in a timely manner, contributing to early detection.

The Transformer-based decoder is a stack of decoder blocks, as shown in orange in Fig. 3, and the dense layer language modelling (LM) head is on the top. The LM head is a linear layer with weights tied to the input embeddings. Each decoder block has a unidirectional self-attention layer, followed by a cross-attention layer and two feed-forward layers. The details about the feed-forward layers can be found in the paper [18]. Here, we focus more on the details of attention layers.

The input to the decoder is the contextualized encoding sequence $\overline{X}_{1:k}$ from the encoder part. The decoder models the conditional probability distribution of the target vector sequence $Y_{1:l}$, given the input $\overline{X}_{1:k}$, shown in Eq. (7):

$$p_{\theta_{\text{dec}}} : (Y_{1:l}|\overline{X}_{1:k}) \tag{7}$$

here l is the number of the target vectors and depends on the input sequence k. By Bayes' rule, this distribution can be factorized into conditional distributions of a target sequence $y_i \in Y_{1:l}$, as shown in Eq. (8):

$$p_{\theta_{\text{dec}}} : (Y_{1:l}|\overline{X}_{1:k} = \Pi_{i=1}^{l}\, p_{\theta_{\text{dec}}}(y_i|Y_{0:i-1}|\overline{X}_{1:k})) \tag{8}$$

where $\forall i \in \{1, 2, \ldots, l\}$. The LM head maps the encoded sequence of target vectors $\overline{Y}_{0:i-1}$ to a sequence of logit vectors $\mathcal{L}_{1:k} = \ell_1, \ldots, \ell_k$, where the dimensionality of each logit vector ℓ_i corresponds to the size of the input vocabulary $1 : k$. A probability distribution over the whole vocabulary is obtained by applying a SoftMax operation on ℓ_i, as shown in Eq. (9):

$$\begin{aligned} p_{\theta_{\text{dec}}}(y_i|\overline{X}_{1:k}, Y_{0:i-1}) &= \text{Softmax}\big(f_{\theta_{\text{dec}}}(\overline{X}_{1:k}, Y_{0:i-1})\big) \\ &= \text{Softmax}\Big(W^T_{\text{emb}}\overline{y}_{i-1}\Big) = \text{Softmax}\ (l_i) \end{aligned} \tag{9}$$

here W^T_{emb} is transpose of the word embedding matrix. We autoregressively generate output from the input sequences through probability distribution in $p_{\theta_{\text{dec}}}(y_i|\overline{X}_{1:k}, Y_{0:i-1})$.

The unidirectional attention takes the input vector y' (representation of y), and the output is the vector representation y''. Each query vector in the unidirectional self-attention layer is compared only to its respective key vector and previous ones to yield the respective attention weights. The attention weights are then multiplied by their respective value vectors and summed together, as in Eq. (10):

$$Y''_{1:l} = V_{1:l}\ \text{Softmax}\left(K^T_{1:l}Q_{1:l}\right) + Y'_{1:l}. \tag{10}$$

The cross-attention layer takes as input two vector sequences: (1) outputs of the unidirectional self-attention layer, i.e. $Y''_{0:l-1}$; (2) contextualized encoding vectors $\overline{X}_{1:k}$ from the encoder. The cross-attention layer puts each of its input vectors to condition the probability distribution of the next target vectors on the encoder's input. We summarize cross-attention in Eq. (11):

$$Y'''_{1:l} = V_{1:l}\ \text{SoftMax}\left(K^T_{1:l}Q_{1:l}\right) + Y''_{1:l}. \tag{11}$$

The index range of the key and value vectors is $1 : l$, which corresponds to the number of contextualized encoding vectors. Y''' is given to the last decoder block and the output from the decoder is a sequence of hidden states $\overline{Y}$.

4.2.3 Model training

In this work, we implement the transfer learning solution [19] for fake news detection. We leverage the previous learnings from a BART pre-trained checkpoint[9] and fine-tune the model on the downstream task of fake news detection. We perform the classification task for fake news detection. For the classification task, we input the same sequences into the encoder and decoder. The final hidden state of the final decoder token is fed into an output layer for classification.

[9] https://dl.fbaipublicfiles.com/fairseq/models/bart.large.mnli.tar.gz.

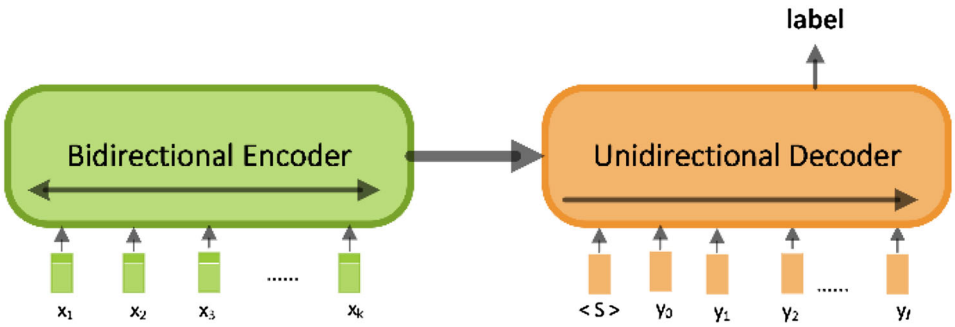

Fig. 5 The classification process; input fed into encoder goes into decoder; the output label

This approach is like the [CLS] representation (CLS for classification) in BERT that serves as the token for the output classification layer. The BERT has the CLS token returned by the encoder, but in BART, we need to add this additional token in the final decoder part. Therefore, we add the token < S > in the decoder to attend to other decoder states from the complete input. We show the classification process in Fig. 5 [16].

We represent the last hidden state $[\mathcal{S}]$ of the decoder as $h_{[\mathcal{S}]}$. The number of the classes is two (fake is 1, real is 0). A probability distribution $p \in [0, 1]^2$ is computed over the two classes using a fully connected layer with two output neurons on top of $h_{[\mathcal{S}]}$, which is followed by the SoftMax activation function, as shown in Eq. (12):

$$p = \text{SoftMax}\left(\mathcal{W} \cdot h_{[\mathcal{S}]} + b\right) \tag{12}$$

where $\mathcal{W}$ is the learnable projection matrix and b is the bias. We train our model for the sequence-pair classification task [17] to classify fake news. Unlike the typical sequence-pair classification task, we use the binary Cross-Entropy with logits loss function instead of the vanilla cross-entropy loss used for the multi-class classification. However, the same model can be adapted for the multi-class classification if there is a need. Through binary cross-entropy loss, our model can assign independent probabilities to the labels. The cross-entropy function H determines the distance between the true probability distribution and predicted probability distribution, as shown in Eq. (13):

$$H\left(y_j, \hat{y}_j\right) = -\sum_{j=1} y_j \log \hat{y}_j \tag{13}$$

where y_j is the ground truth for observation and $\hat{y}_j$ is the model prediction.

Based on the Transformer architecture, our model naturally takes the sequences of words as the input, which keeps flowing up the stacks from encoder to decoder, while the new sequences are coming in. We organize the news data according to the timestamps of users' engagements so that the temporal order is retained during the creation of the sequences. We use paddings to fill up the shorter readers' sequences, while the longer sequences are truncated.

5 Experimental set-up

5.1 Datasets

It was not a trivial task to find a suitable dataset to evaluate our proposed model because most of the standard datasets available for fake news detection are either too small, sparse, or void of temporal information.

A few state-of-the-art datasets, such as FakeNewsNet [23], are not available as the full version but can be found as the sample data. This is mainly because most of these datasets use Twitter data for social contexts and thus cannot be publicly accessible due to license policies. Other available datasets that consider the fake news content are outdated. Since fake news producers typically change their strategies over time, such datasets are not suitable to solve the issue of fake news data for the recent news data. After extensive research and careful consideration, we found that the NELA-GT-19 and Fakeddit are most suitable for our proposed problem regarding the number of news articles, temporal information, social contexts, and associated side information.

To evaluate the effectiveness of our proposed FND-NS model, we conducted comprehensive experiments on the data from the real-world datasets: NELA-GT-2019 [21] and Fakeddit [22]. Both datasets are in English, and we take the same timeline for the two datasets.

5.1.1 NELA-GT-2019

For our news component, we use the NELA-GT-2019 dataset [21], a large-scale, multi-source, multi-labelled benchmark dataset for news veracity research. This dataset can be accessed from here.[10] The dataset consists of 260 news sources with 1.12 million news articles. These news articles were published between January 1, 2019, and December 31, 2019. The actual news articles are not labelled. We get the ground truth labels (0—reliable, 1—mixed, 2—unreliable) at the source level and use the weak supervision (discussed in Sect. 5.2) to assign a label to each news article. We use the

[10] https://doi.org/10.7910/DVN/O7FWPO

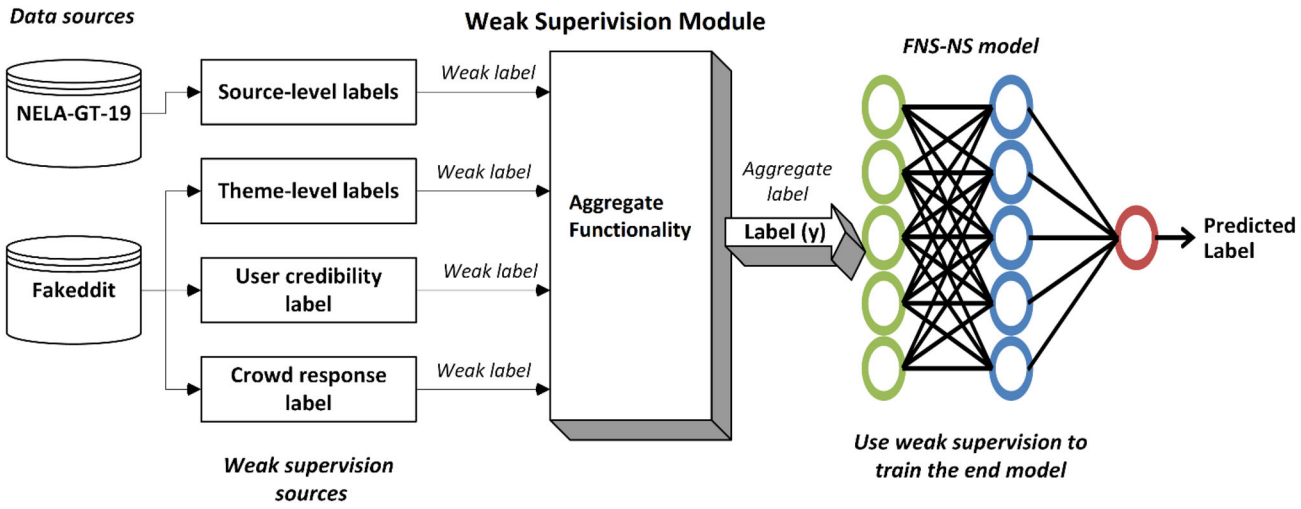

Fig. 6 Weak supervision module

article ID, publication timestamp, source of news, title, content (body), and article's author for the news features. We only use the 'reliable' and 'unreliable' source-level labels. For the 'mixed' labels, we change them to 'unreliable' if they are reported as 'mixed' by the majority of the assessment sites and omit the left-over 'mixed' sources. The statistics of the actual data can be found in the original paper [21].

5.1.2 Fakeddit

For the social contexts, we use the Fakeddit dataset [22], which is a large scale, multi-modal (text, image), multi-labelled dataset sourced from Reddit (social news and discussion website). This dataset can be accessed from here.[11] Fakeddit consists of over 1 million submissions from 22 different subreddit (users' community boards) and over 300,000 unique individual users. The data are collected from March 19, 2008, till October 24, 2019. We consider the data from January 01, 2019, till October 24, 2019, to match it with the timeline of the NELA-GT-19. According to previous work [3], this amount of data is considered sufficient for testing the concept drift. We use the features of the social context from this dataset: submission (the post on a news article), submission title (title of the post matching with the headline of the news story), users' comments on the submission, user IDs, subreddit (a forum dedicated to a specific topic on Reddit) source, news source, number of comments, up-vote to down-vote ratio, and timestamp of interaction. The statistics of the actual data can be found in the original paper [22].

[11] https://github.com/entitize/fakeddit.

5.2 Weak supervision

The weak supervision module is a part of our proposed framework as shown in Fig. 6. We utilize weak (distant) labelling to label news articles. Weak supervision (distant supervision) is an alternative approach to label creation, in which labels are created at the source level and can be used as proxies for the articles. One advantage of this method is that it reduces the labelling workload. Furthermore, the labels for articles from known sources are known instantly, allowing for real-time labelling, as well as parameter updates and news analysis. This method is also effective in the detection of misinformation [9, 53–55].

The intuition behind weak supervision is that the weak labels on the training data may be imprecise but can be used to make predictions using a strong model [54]. We overcome the scarcity issue of hand-labelled data by compiling a dataset like this, which can be done almost automatically and can yield good results as shown in Sect. 6.3.

In our work, we use the weak supervision to assign article-level labels for the NELA-GT-2019 dataset, where the source-level labels are provided by the dataset. This method is also suggested by the providers of NELA-GT-2019 dataset [21]. For the Fakeddit dataset, the ground truth labels are provided by the dataset itself, we only create two new labels for this dataset—'crowd response' and 'user credibility'. We use these labels provided by the datasets to create a new weighted aggregate label to be assigned to each news article.

From the NELA-GT-19, we get the ground truth labels associated with each source (e.g. NYTimes, CNN, BBC, theonion and many more). These labels are provided by seven different assessment sites: (1) Media Bias/Fact Check, (2) Pew Research Center, (3) Wikipedia, (4) OpenSources, (5) AllSides, (6) BuzzFeed News, and (7) Politifact, to each news

source. Based on these seven assessments, Gruppi et al. [21] created an aggregated 3-class label: unreliable, mixed and reliable, to assign to each source. We use the source-level labels as the proxies for the article-level labels. The assumption is that each news story belongs to a news source and the reliability of the news source has an impact on the news story. This approach is also suggested in the NELA-GT-18 [56] and NELA-GT-20 [57] papers and has shown promising results in the recent fake news detection work [3].

Once we get the label for each news article, we perform another step of processing over article-level labels. As mentioned earlier, the NELA-GT-19 provides the 3-class source-level labels: {'Unreliable', 'Mixed, 'Reliable'}. According to Gruppi et al. [21], the 'mixed' label means mixed factual reporting. We have not used the 'mixed' label in our work. We change the 'mixed' label to 'unreliable' if it is reported as 'mixed' by the majority of the assessment sites. For the remaining left-over mixed labels, we remove those sources to avoid ambiguity. This gives a final news dataset with 2-class labels: {'Reliable', 'Unreliable'}.

The other dataset used in this work is Fakeddit. Nakamura et al. [22] also use the weak supervision to create labels in the Fakeddit dataset. They use the Reddit themes to assign a label to each news story. More details about Reddit themes and the distant labelling process are available in their paper [22].

The dataset itself provides labels as 2-way, 3-way and 6-way labels. We use the 6-way label scheme, where the labels assigned to each sample are: 'True', 'Satire', 'Misleading Content', 'Imposter Content', 'False Connection', and 'Manipulated Content'. We assign two more weak labels in addition to 6-way labels, which are user credibility and crowd response labels that we compute using the social contexts. The user credibility level has five classes: 'New user', 'Very uncredible', 'Uncredible', 'Credible', 'Very credible'. The crowd response has two classes: 'Fake' and 'Real'.

We get the user credibility levels through our ZSL classifier (Fig. 2). For the crowd response, we simply take the scores of all the comments (posts) of users on a news story to determine the overall view of users on this news story. The goal is to make the label learning more accurate by adding more weak labels to the available labels. In a preliminary test, we find that using weak supervision with multiple weak labels (our case) achieves a better result than using Fakeddit theme-based weak labels alone [22] (they learned using their weak supervision model).

Based on this, we design a formula to assign the final label ('Real', 'Fake') to each sample in the aggregate functionality part. We assign a final label '*Fake*' to a new article if one of the following conditions is satisfied: (1) its 6-way label specified in Fakeddit is 'Satire', 'Misleading content', 'Imposter', 'False connection', or 'Manipulated content'; (2) its label specified in NELA-GT-19 is 'Unreliable'; (3) its label according to user credibility is 'Very uncredible' or 'Uncredible'; (4) its label according to crowd response is 'Fake'. We assign a label '*Real*' to the news if all of the following conditions are satisfied: (1) its label in Fakeddit is 'True'; (2) its label in NELA-GT-19 is 'Reliable'; (3) its label according to user credibility is 'New user', 'Credible', or 'Very credible'; (4) its label according to crowd response is 'Real'. We do not penalize a new user because we do not have sufficient information for a new user.

Table 1 Fake versus real samples used from the datasets

	Actual fake	Actual real
Predicted fake	2108	158
Predicted real	91	1643

Table 2 Confusion matrix

	Actual fake	Actual real
Predicted fake	TP	FP
Predicted real	FN	TN

Our FND-NS model implicitly assumes that these weak labels are precise and heuristically matches the labels against the corpus to generate the training data. The model predicts the final label: 'Real' or 'Fake' for the news.

To handle the data imbalance problem in both datasets, we use the under-sampling technique [58], in which the majority class is made closer to the minority class by removing records from the majority class. The number of fake and real news items from datasets used in this research is given in Table 1.

We temporally split the data for the model training. We use the last 15% of the chronologically sorted data as the test set, the second to last 10% of the data as the validation set and the initial 75% of the data as the train set. We also split the history of each user based on the interaction timestamp. We consider the last 15% of the interactions as the test set.

5.3 Evaluation metrics

In this paper, the fake news detection task is a binary decision problem, where the detection result is either fake or real news. To assess the performance of our proposed model, we use the accuracy ACC, precision Prec, recall Rec, F1-score F1, area under the curve AUC and average precision AP as the evaluation metrics. The confusion matrix determines the information about actual and predicted classifications, as shown in Table 2.

The variables TP, FP, TN and FN in the confusion matrix refer to the following:

- True Positive (TP): number of fake news that are identified as fake news.

- False Positive (FP): number of real news that are identified as fake news.
- True negative (TN): number of real news that are identified as real news.
- False negative (FN): number of fake news that are identified as real news.

For the Prec, Rec, F1 and ACC, we perform the specific calculation as:

$$\text{Prec} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

$$F1 = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FN} + \text{FP})} \tag{16}$$

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{17}$$

To calculate the AUC, we calculate the true positive rate (TPR) and the false positive rate (FPR). TPR is a synonym for the recall, whereas FPR is calculated as:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{18}$$

The receiver operating characteristic (ROC) plots the trade-offs between the TPR and FPR at different thresholds in a binary classifier. The AUC is an aggregate measure to evaluate the performance of the model across all those possible thresholds. Compared to the accuracy measure ACC, the AUC is better at ranking predictions. For example, if there are more fake news samples in the classification, the accuracy measure may favour the majority class. On the other hand, the AUC measure gives us the score order (ranking) along with the accuracy score. We also include the average precision AP that gives the average precision at all such possible thresholds, similar to the area under the precision–recall curve.

5.4 Hyperparameters

We implement our model with Pytorch on the GPUs provided by Google Colab Pro.[12] We use the pre-trained checkpoint of *bart-large-mnli*.[13] The MNLI is a crowd-sourced dataset that can be used for the tasks such as sentiment analysis, hate speech detection, detecting sarcastic tone, and textual entailment (conclude a particular use of a word, phrase, or sentence). The model is pre-trained on 12 encoder and 12 decoder layers, in total 24 layers with a dimensionality size of 1024. The model has 16-heads with around 1 million parameters. We add a 2-layer classification head fine-tuned on the MNLI. The model hyperparameters are shown in Table 3.

Table 3 Hyperparameters used for the proposed model

Hyperparameter	Value
Model	Bart Large MNLI
Vocabulary size	50,265 (vocabulary size defines the number of different tokens)
Dimensionality size	1024 (dimensionality of the layers and the pooling layer)
No. of encoder layers	12
No. of decoder layers	12
Attention heads	16 (number of attention heads for each attention layer in encoder), 16 (number of attention heads for each attention layer in decoder)
Feed-forward layer dimensionality	4096 (dimensionality of the feed-forward layer in encoder), 4096 (dimensionality of the feed-forward layer in decoder)
Activation function	Gelu (nonlinear activation function in the encoder and pooler)
Position embeddings	1024
Number of labels	2
Batch size	8 (tested 8, 16, 32)
Epochs	10
Sequence length	700 (other values used are 512, 1024, 2048 but 700 suits to current settings)
Learning rate	1e−4 (tested 1e−2, 1e−3, 1e−4)
Dropout	0.1 (dropout probability for all fully connected layers, tested in {0.0, 0.1, 0.2, …, 0.9})
Warm up steps	500 (tested 0, 100, 300, 500, 1000)
Optimizer	Adam
Loss function	Cross entropy
Output layer	SoftMax

Our model is trained using Adam optimizer [59]. In our experiments, the larger batch sizes did not work. So, we decrease the batch size from 32 (often used) to 8 until the memory issues get resolved. We keep the same batch size of 8 during the training and validation process. The number of train epochs is 10. The default sequence length supported by the BART is 1024. Through an initial analysis of our datasets, we find that the mean length of a news story is around 550 words, whereas a Reddit post is on average 50 words. The maximum sequence length of BERT and GPT-2 is 512, which

[12] https://colab.research.google.com/.

[13] https://dl.fbaipublicfiles.com/fairseq/models/bart.large.mnli.tar.gz.

is less than the mean length of a news story. So, we set the sequence length to 700 to include the average news length and the side information from the news and social contexts. The sequences are created based on the timestamps of the user's engagement. The longer sequences are truncated, while the shorter ones are padded to the maximum length.

5.5 Baseline approaches

We compare our model with the state-of-the-art fake news detection methods, including deep neural and traditional machine learning methods. We also consider other baselines, including a few recent Transformer models, a neural method (Text CNN) and a traditional baseline (XGboost).

A few state-of-the-art methods, such as a recent one by Liu and Wu [5], is not publicly accessible, so we have not included those in this experiment. Some of these baselines are by default using content features only (e.g. exBake, Grover, Transformer-based baselines, TextCNN), and some are using social contexts only (e.g. 2-Stage Tranf., SVM-L, SVM-G and LG group). A few baselines use the social contexts with content-based features (e.g. TriFN, FANG, Declare). Our model uses both the news content and the social contexts with the side information. For a fair comparison, we test the baselines using their default settings. In addition, we also test them by including both news content and social contexts. In this case, we create variants of baselines (default setting and setting with both news and social context).

To determine the optimal hyperparameter settings for the baselines, we primarily consult the original papers. However, there is little information provided on how the baselines are tuned in these papers. So, we optimize the hyperparameters of each baseline using our dataset. We also train all the models from scratch. We optimize the standard hyperparameters (epochs, batch size, dimensions, etc.) for each baseline. Some of the hyperparameters specific to individual models are reported below (along with the description of each method).

FANG [15]: it is a deep neural model to detect fake news using graph learning. We optimize the following losses simultaneously during the model training: 1) unsupervised proximity loss, 2) self-supervised stance loss, and 3) supervised fake news detection loss (details can be found in the paper [15]), whereas the implementation details are available here.[14] We feed both the news-related information and social contexts into the model.

2-Stage Tranf. [1]: it is a deep neural fake news detection model that focuses on fake news with short statements. The original model is based on BERT-base [17], and checkpoint is recommended to use, so we build this model using the same method. We feed the news-related information and social contexts into the model. We also represent another variant of this model where we remove the news body and news source, keeping only social contexts (as in the default model) and represent it as 2-Stage Tranf. (*nc*-).

exBAKE [7]: it is another fake news detection method based on deep neural networks. This model is also based on the BERT model and is designed for the content of news. Besides showing the original model's results, we also incorporate social contexts into the model by introducing another variant of this model. The model variants are exBAKE (with both news content and social contexts) and exBAKE (*sc*-) (default model, without social contexts).

Declare [8]: it is a deep neural network model that assesses the credibility of news claims. This model uses both the news content and social contexts by default, so we feed this information to the model. An implementation of the model can be found here.[15]

TriFN [4]: it is a matrix factorization based model that uses both news content and social contexts to detect fake news. We give both the news and social contexts to the model. The model implementation can be accessed here.[16]

Grover [12]: it is a deep neural network-based fake news detection model based on GPT-2 [18] architecture. The model takes news related information and can incorporate additional social contexts too. We give both the news content and social contexts to Grover. In addition, we remove the social contexts and keep the content information only (as in default Grover model), which we represent as Grover (*sc*-). We use the Grover-base implementation of the model and initialize the model using the GPT-2 checkpoint.[17] The model implementation is available here.[18]

SVM-L; SVM-G; LG [14]: it is a machine learning model based on similarity among the friends' networks to discover fake accounts in social networks and detect fake news. We use all the proposed variants: linear support vector machine (SVM), medium Gaussian SVM and logistic regression, and optimize them to their optimal settings.

BERT [17]: BERT (bidirectional encoder representations from Transformers) is a Google-developed Transformer-based model. We use both the cased (BERT-c) and uncased (BERT-u) version of the BERT, with 24-layer, 1024-hidden, 16-heads, and 336 M parameters. The model implementation can be found here.[19]

VGCN-BERT [60]: it is a deep neural network-based model that combines the capability of BERT with a vocab-

[14] https://github.com/nguyenvanhoang7398/FANG.

[15] https://github.com/atulkumarin/DeClare.

[16] https://github.com/KaiDMML/FakeNewsNet.

[17] https://openai.com/blog/tags/gpt-2/.

[18] https://github.com/rowanz/grover.

[19] https://github.com/google-research/bert.

ulary graph convolutional network (VGCN). The model implementation is available here.[20]

XLNET [61]: it is an extension of the Transformer-XL model, which was pre-trained with an autoregressive method to learn bidirectional contexts. We use the hyperparameters: 24-layer, 1024-hidden, 16-heads, 340 M parameters. The model implementation is available here.[21]

GPT-2 [18]: it is a causal (unidirectional) Transformer pre-trained using language modelling. We use the hyperparameters: 24-layer, 1024-hidden, 16-heads, 345 M parameters. The model implementation is available here.[22]

DistilBERT [62]: it is a BERT-based small, fast, cheap, and light Transformer model, which uses 40% fewer parameters than BERT-base, runs 60% faster, and keeps over 95% of BERT's results, as measured in the paper [62]. We only use the cased version for this model (based on the better performance of the BERT cased version, also shown in the later experiments). We use the hyperparameters: 6-layer, 768-hidden, 12-heads, 65 M parameters, and the model implementation is available here.[23]

Longformer [63]: it is a Transformer-based model that scales linearly with sequence length, making it easy to process documents of thousands of tokens or longer. We use the hyperparameters with 24-layer, 1024-hidden, 16-heads, ~435 M parameters and the model is initiated from the RoBERTa-large[24] checkpoint, trained on documents of max length 4,096. The model implementation is available here.[25]

We use both news content and social contexts to train the Transformer-based models (BERT, VGCN-BERT, XLNET, GPT-2, DistillBERT), which are built for taking textual information. But these models can also handle the social contexts, as evidenced in some preliminary tests where we first fed the news content, then news content with social contexts, and found a marginal difference in performance.

Text CNN [64]: it is a convolution neural network (CNN) with one layer of convolution on top of word vectors, where the vectors are pre-trained on a large number (~ 100 billion) of words from Google News.[26] The model implementation is available here.[27]

[20] https://github.com/Louis-udm/VGCN-BERT.

[21] https://github.com/zihangdai/xlnet.

[22] https://github.com/openai/gpt-2.

[23] https://huggingface.co/transformers/model_doc/distilbert.html.

[24] https://github.com/pytorch/fairseq/tree/master/examples/roberta.

[25] https://github.com/allenai/longformer.

[26] https://code.google.com/archive/p/word2vec/.

[27] https://github.com/dennybritz/cnn-text-classification-tf.

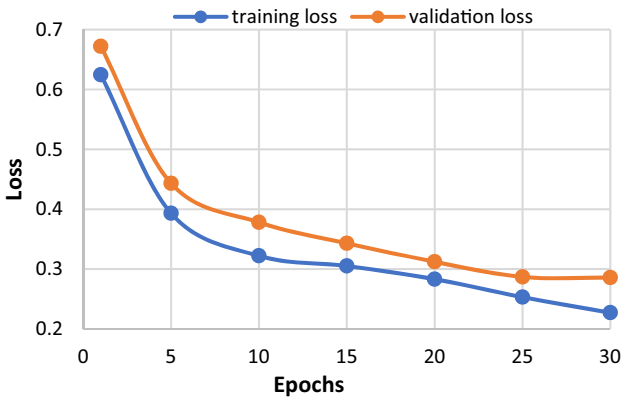

Fig. 7 Training versus validation loss

Table 4 Confusion matrix of the sample data

	Actual fake	Actual real
Predicted fake	35,104	12,997
Predicted real	9793	32,893

XGBoost [65]: it is an optimized distributed machine learning algorithm under the gradient boosting framework, with implementation from here.[28]

We report the results for each baseline based on best performing hyperparameters for each evaluation metric.

6 Results and analyses

In this section, we present the results and analyse them.

6.1 Model performance

We show the learning curve for training loss and validation loss during model training in Fig. 7.

In our model, the validation loss is quite close to the training loss. The validation loss is slightly higher than training loss, but overall, both values are converging (when plotting loss over time). Overall, it shows a good fit for model learning.

We also test the model's performance on the test data to show the confusion matrix in Table 4.

Based on the confusion matrix in Table 4, the model accuracy is 74.89%, which means more than 74% of the results are correct. We get the precision of 72.40%, which means that we have a few false positives (news is real but predicted as fake), and we can correctly predict a large portion of true positives (i.e. the news is fake and predicted as fake). We get a recall value of 77.68%, which shows we have many more

[28] https://xgboost.readthedocs.io/en/latest/.

Table 5 Overall performance comparison

Method	ACC	Prec	Rec	F1	AUC	AP
FND-NS	0.748	0.724	0.776	0.749	0.704	0.710
Fake news detection baselines						
FANG [15]	0.687	0.673	0.618	0.644	0.681	0.666
2-Stage Tranf. [1]	0.621	0.647	0.626	0.636	0.620	0.614
2-Stage Tranf. (*nc*-)	0.612	0.634	0.610	0.622	0.612	0.607
exBAKE [7]	0.665	0.625	0.614	0.619	0.640	0.601
exBAKE (*sc*-)	0.685	0.630	0.610	0.620	0.651	0.610
Declare [8]	0.621	0.610	0.607	0.608	0.578	0.590
TriFN [4]	0.615	0.601	0.614	0.607	0.601	0.596
Grover [12]	0.533	0.567	0.586	0.576	0.565	0.575
Grover (*sc*-)	0.578	0.601	0.617	0.609	0.582	0.612
SVM-L [14]	0.415	0.429	0.451	0.440	0.425	0.421
SVM-G [14]	0.434	0.450	0.455	0.452	0.428	0.430
LG [14]	0.425	0.468	0.457	0.462	0.431	0.420
Other baselines						
BERT-c [17]	0.640	0.610	0.620	0.615	0.622	0.639
BERT-u [17]	0.589	0.560	0.578	0.569	0.566	0.597
VGCN-BERT [60]	0.627	0.598	0.610	0.604	0.610	0.645
XLNET [61]	0.520	0.600	0.530	0.563	0.525	0.557
GPT-2 [18]	0.635	0.620	0.610	0.615	0.614	0.623
DistilBERT [62]	0.522	0.510	0.490	0.500	0.467	0.526
Longformer [63]	0.543	0.573	0.550	0.561	0.536	0.545
Text CNN [64]	0.520	0.480	0.501	0.490	0.530	0.522
XGBoost [65]	0.510	0.454	0.487	0.470	0.525	0.514

true positives than false negatives. Generally, a false negative (news is fake but predicted as real) is worse than a false positive in fake news detection. In our experiment, we get less false negatives than false positives. Our F1-score is 74.95%, which is also quite high.

6.2 Overall performance comparison

We show the best results of all baselines and our FND-NS model using all the evaluation metrics in Table 5. The results are based on data from both datasets, i.e. social contexts from Fakeddit on the NELA-GT-19 news. The input and hyperparameter optimization settings for each baseline model are given above (Sect. 5.5). The best scores are shown in bold.

Overall, we see that our proposed FND-NS model has the highest accuracy (74.8%), precision (72.4%), recall (77.6%), AUC (70.4%) and average precision (71%) among all the models. The superiority of our model is attributed to its advantages:

- Our model utilizes rich features from news content and social contexts to extract the specifics of fake news.
- We exploit the knowledge transition from large-scale pretrained models (e.g. MNLI) to the fake news detection problem with transfer learning. The right choice of the pretrained checkpoints on the specific corpus helps us make better predictions. During empirical testing, we check the performance of our model with and without including the pre-trained checkpoints. We find better results with the inclusion of the MNLI checkpoint.
- We model the timeliness in our model through an autoregressive model, which helps us detect fake news in a timely and early manner.
- We address the label shortage problem through the proposed weak supervision module, which helps us make better predictions on unforeseen news.

We have the following more findings from the results:

Among the fake news detection baselines, the overall performance of FANG is the best. The performance of FANG is the second best after our FND-NS model. FANG uses graph learning to detect fake news and focus on learning context representations from the data. The overall performance of exBAKE and 2-Stage Tranf. as indicated in most metrics is the next best. These models (exBAKE and 2-Stage Tranf.) are based on the BERT model and are suitable for representation learning. Our model outperforms these models, most likely because we focus on both autoregression and representation learning.

The 2-Stage-Tranf. uses the claim data from the social media. We also test this model with its default input setting as in 2-Stage Tranf. (*nc*-), omitting the news content (news body, headline, source) and allowing only social context features (such as post, title, score). With this change, we do not find much difference in the performance. We find the better performance (though marginal) of 2-Stage-Tranf. when we only keep the news-related features (not including social contexts). This is most likely due to the support of the 2-Stage-Tranf. model for auxiliary information. Our model performs better than 2-Stage-Tranf. with its support for side information. This is likely because our model can handle longer sequence lengths than the baselines, resulting in some loss of information and thus accuracy in those models.

Then comes the performance of Declare, TriFN and Grover models, all of which are considered the benchmark models in fake news research. Grover is a content-based neural fake news detection model. Declare is a neural network framework that detects fake news based on the claim data. TriFN is a non-negative matrix factorization algorithm that includes news content and social contexts to detect fake news.

We also test Grover (content-based model) without social contexts in Grover (*sc*-). We find some better performance of Grover (*sc*-) than Grover's (with both inputs). This result shows that a model built on rich content features (news body,

headline, publication date) with autoregressive properties (GPT-2 like architecture) can perform better even without social contexts.

The SVM and LG are also used for fake news detection. Due to their limited capabilities and the use of hand-crafted dataset features, the accuracies of SVM and LG are lower in these experiments. The results for SVM and LG do not generalize the performance of these models to all situations in this field.

In general, the performance of the Transformer-based methods is better than the traditional neural-based methods (Text CNN) and the linear models (SVM, LG, XGBoost). This is probably because the Transformer-based methods use the multi-head attention and positional embeddings, which are not by-default integrated with the CNNs (of text CNN) and the linear methods. With the default attention mechanisms and more encoding schemes (e.g. token, segment and position), the Transformers compute input and output representations better than the traditional neural methods. Our FND-NS model, however, performs better than these Transformer models. This is because our framework includes many add-ons, such as weak supervision, representation learning, autoregression, which (all of them together) are not present in the typical Transformer models.

The general performance of simple neural methods (e.g. Text CNN, Declare) that are not Transformer-based is better than the linear methods (SVM, LG, XGBoost). This is probably because the linear methods use manual feature engineering, which is not optimal. On the other hand, the neural-based methods can capture both the global and the local contexts in the news content and social contexts to detect the patterns of fake news.

Among the Transformers, the cased model (e.g. BERT-c), in general, performs better than its respective uncased version (e.g. BERT-u). Generally, fake or false news uses capital letters and emotion-bearing words to present something provoking. Horne and Adalı [29] also present several examples where fake titles use capitalized words excessively. This shows why the cased models can detect fake news better compared to the uncased versions.

The overall performance of the distilled (condensed) versions (Distill BERT) is slightly lower than their respective actual models (BERT). Based on the better performance of the BERT cased version over its uncased version, we use the cased version of Distill BERT. The Distill BERT does not use token-type embeddings and retains only half of the layers and parameters of the actual BERT, which probably results in the overall lower prediction accuracy. The distilled versions balance the computational complexity and accuracy. This result suggests that using the distilled version can achieve comparable results (to the original model) with better speed.

We also see that the general performance of the autoregressive models (XLNet and GPT-2) is better than the most autoencoding models (DistilBERT, Longformer, BERT-u). The exception is seen in BERT-c for some scores. The autoregressive Transformers usually model the data from left to the right and are suitable for time-series modelling. They predict the next token after reading all the previous ones. On the other hand, the autoencoding models usually build a bidirectional representation from the whole sentences and are suitable for natural language understanding tasks, such as GLUE (general language understanding evaluation), classification, and text categorization [17]. Our fake news detection problem implicitly involves data that vary over time. The autoregressive models show relatively better results. Our FND-NS model performs the best because it has both the autoencoding model and the autoregressive model.

We find VGCN-BERT as a competitive model. The VGCN is an extension of the CNN model combined with the graph-based method and the BERT model. The results in Table 5 show the good performance of CNN in the TextCNN method and that of the BERT model. The neural graph networks have recently demonstrated noticeable performance in the representative learning tasks by modelling the dependencies among the graph states [66]. That is why the performance of VGCN-BERT (using BERT-u) is better than TextCNN and BERT-u alone. This result also indicates that hybrid models are better than standalone models. FANG also uses a graph neural network with the supervised learning loss function and has shown promising results.

6.3 Effectiveness of weak supervision

In this experiment, we test the effectiveness of the weak supervision module on the validation data for the accuracy measure.

We show different settings for weak supervision. These settings are:

M1: Weak supervision on both datasets, NELA-GT-19 and Fakeddit with original labels + user credibility label + crowd response label;
M2: Weak supervision on both datasets, NELA-GT-19 and Fakeddit with original labels + user credibility label;
M3: Weak supervision on both datasets, NELA-GT-19 and Fakeddit with original labels + crowd response label;
M4: Weak supervision on both datasets, NELA-GT-19 and Fakeddit with original labels;
M5: Weak supervision on NELA-GT-19 only;
M6: Weak supervision on Fakeddit only with original labels;
M7: Weak supervision on Fakeddit with original + user credibility labels;
M8: Weak supervision on Fakeddit with original + crowd response labels;

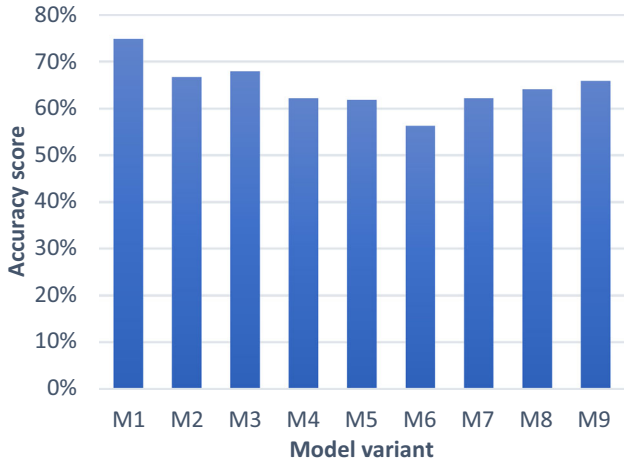

Fig. 8 Accuracy percentage on different settings of weak supervision for FND-NS model

M9: Weak supervision on Fakeddit with original + user credibility labels + crowd response labels.

The results of FND-NS on these settings are shown in Fig. 8. The results show that our model performs better when we include newly learned weak labels and worse when we omit any one of the weak labels. This is seen with the best performance of FND-NS in the M1 setting. The crowd response label proves to be more productive than the user credibility label. This is seen with > 2% loss in accuracy in M2 and M7 (without crowd response) compared to the M3 and M8 (with crowd response). This conclusion is also validated by another experiment in the ablation study (Sect. 6.4) discussed later.

We also see that the model performance improves when we include both datasets. This can be seen with the overall better performance of the model with both datasets. In general, all these results (in Fig. 8) indicate that the weak labels may be imprecise but can be used to provide accurate predictions. Some of these FND-NS results are also present in the Ablation study (Sect. 6.4) and are explored in more detail there.

In the Fakeddit paper [22], we see the performance of the original BERT model to be around 86%, which is understandable because the whole Fakeddit dataset (ranging from the year 2008 till 2019) is used in that work. In our paper, we use the Fakeddit data only for the year 2019. Usually, the models perform better with more data. In particular, deep neural networks (e.g. BERT) perform better with more training examples. Omitting many training examples could affect the performance of the model. This is the possible reason we see lower accuracy of our model in this experiment using the Fakeddit data. For the same reason, we see the performance of the original BERT a bit lower with the Fakeddit data in Table 5.

The results on the original NELA-GT-19 may also be different in our work. This is because we do not consider much of the mixed labels from the original dataset. Also, since we use under-sampling for data balancing for both of our datasets, the results may vary for the experiments in this paper versus the other papers using these datasets.

6.4 Ablation study

In the ablation study, we remove a key component from our model one a time and investigate its impact on the performance. The list of reduced variants of our model are listed below:

FND-NS: The original model with news and social contexts component;
FND-N: FND-NS with news component—removing social contexts component;
FND-N(h-): FND-N with headlines removed from the news component;
FND-N(b-): FND-N with news body removed from the news component;
FND-N(so-): FND-N with news source removed from the news component;
FND-N(h-)S: FND-NS with headlines removed from the news component;
FND-N(b-)S: FND-NS with news body removed from the news component;
FND-N(so-)S: FND-NS with news source removed from the news component;
FND-S: FND-NS with social context component—removing news component;
FND-S (uc-): FND-S with user credibility removed from the social contexts;
FND-S (cr-): FND-S with crowd responses removed from the social contexts;
FND-NS (uc-): FND-NS with user credibility removed from the social contexts;
FND-NS (cr-): FND-NS with crowd responses removed from the social contexts;
FND (en-)-NS: FND-NS with the encoder block removed—sequences from both the news and social contexts components are fed directly into the decoder;
FND (de-)-NS: FND-NS with the decoder block removed;
FND (12ly-)-NS: FND-NS with 12 layers removed (6 from encoder and 6 from decoder).

The results of the ablation study are shown in Table 6.
The findings from the results are summarized below:
When we remove the news component, the model accuracy drops. This is demonstrated by the lower scores of FND-S, compared to the original model FND-NS in Table 6. However, when we remove the social context component,

Table 6 Variants and the results

Variant	ACC	F1	AUC	AP
FND-NS	0.748	0.749	0.704	0.710
FND-N	0.618	0.609	0.572	0.589
FND-N(h-)	0.610	0.598	0.568	0.569
FND-N(b-)	0.569	0.574	0.545	0.547
FND-N(so-)	0.587	0.582	0.573	0.560
FND-N(h-)S	0.695	0.639	0.689	0.655
FND-N(b-)S	0.670	0.615	0.645	0.628
FND-N(so-)S	0.684	0.639	0.685	0.649
FND-S	0.659	0.621	0.619	0.607
FND-S (uc-)	0.641	0.614	0.587	0.591
FND-S (cr-)	0.622	0.597	0.572	0.582
FND-NS (uc-)	0.680	0.685	0.651	0.672
FND-NS (cr-)	0.667	0.660	0.635	0.622
FND (en-)-NS	0.575	0.567	0.551	0.580
FND (de-)-NS	0.527	0.526	0.520	0.534
FND (12ly-)-NS	0.515	0.510	0.468	0.520

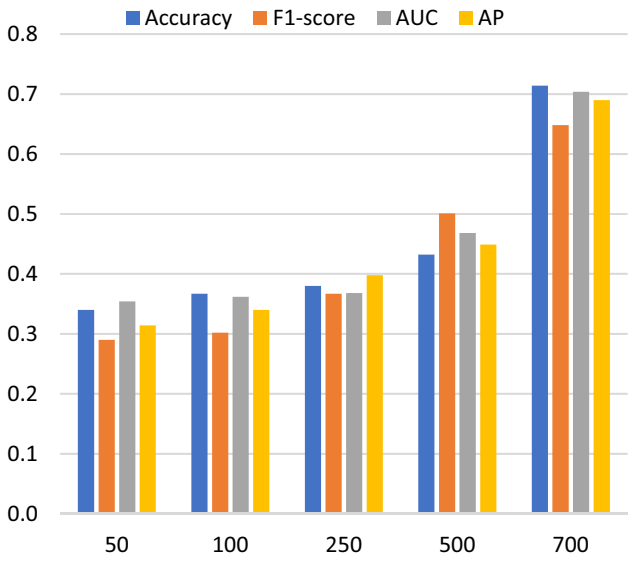

Fig. 9 The FND-NS with different sequence lengths

the model accuracy drops more. This is seen with the lower accuracy of FND-N (without social contexts) compared to the FND-S. This result indicates that both the news content and social contexts play an essential role in fake news detection, as indicated in the best performance of the FND-NS model.

The results also show that the performance of the FND-NS model is impacted more when we remove the news body than removing the headline or the source of the news. This is seen with relatively lower accuracy of FND-N(b-) compared to both the FND-N(h-) and FND-N(so-). The same results are seen in the lower accuracy of FND-N(b-)S compared to both the FND-N(h-)S and FND-N(so-)S. The result shows that the headline and source are important, but the news body alone carries more information about fake news. The source seems to carry more information than the headline; this is perhaps related to the partisan information.

From the social contexts, we find that when we remove the user credibility or the crowd responses, the model performance in terms of accuracy is decreased. Between the user credibility and crowd responses, the model performance is impacted more when we remove crowd responses. This is seen with the lower performance of FND-S(cr-) and FND-NS(cr-) compared to FND-S(uc-) and FND-NS(uc-). The same finding is also observed in Fig. 8 for the results of weak supervision. The probable reason for the crowd responses being more helpful for fake news detection could be that they provide users' overall score on a news article directly, whereas the user credibility only plays an indirect role in the prediction process. According to the concept drift theory, the credibility levels of the users may change over time. Some users leave the system permanently, some change their viewpoints, and new users keep coming into the system. Therefore, the user credibility may not be as informative as crowd responses and thus has less effect on the overall detection result.

The model performance is impacted when we remove the encoder from the FND-NS. The model performance is affected even more when we remove the decoder. This is seen with the lower scores of FND(de-)-NS, which is lower than FND(en-)-NS. In our work, the decoder is the autoregressive model, and the encoder is the autoencoding model. This result also validates our previous finding from the baselines (Table 5), where we find the better performance of the autoregressive model (e.g. GPT-2) compared to most autoencoding models (Longformer, DistillBERT, BERT-c).

Lastly, we find that removing layers from the model lowers the accuracy of the FND-NS model. We get better speed upon removing almost half the layers and the parameters, but this comes with the information loss and the lower accuracy. This also validates our baseline results in Table 5, where we see that the distilled models are faster in speed, but they do not perform as good as the original models.

We also test the sequence lengths in {50, 100, 250, 500, 700} in our model. It is important to mention that the large sequence length often causes memory issues and interrupts the model's working. However, we adjust the sequence length according to the batch size. This facility to include sequence length > 512 is provided by BART. Most models (e.g. BERT, GPT-2) do not support sequence length > 512. Our model performance with different sequence lengths is shown in Fig. 9.

The results in Fig. 9 show that our model performs the best when we use a sequence length of 700. Our datasets consist

of many features from the news content and social contexts over the span of close to one year. The news stories are on average 500 words or more, which carries important information about the news veracity. The associated side information is also essential.

The results clearly show that truncating the text could result in information loss. It is why we see the information loss with the smaller sequence lengths. With a larger sequence length, we could accurately include more news features and users' engagement data to accurately reflect the patterns in users' behaviours.

We also observe that the sequence length depends on the average sequence length of the dataset. Since our datasets are large and by default contain longer sequences, we get better performance with a larger sequence length. Due to the resource limitations, we could not test on further larger lengths, which we leave for future work.

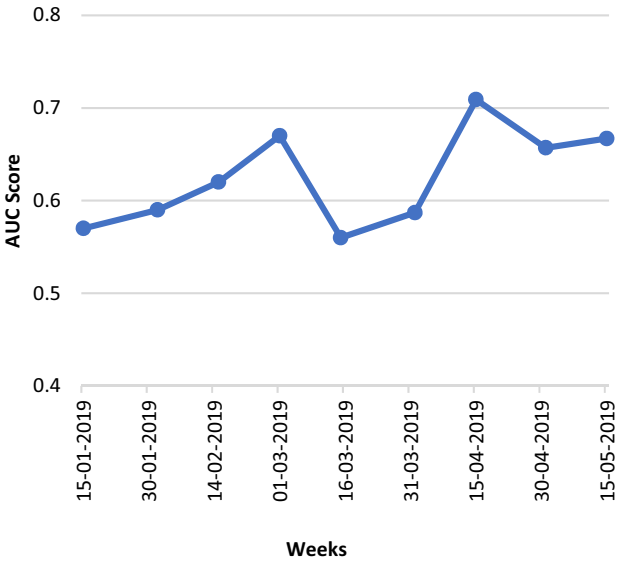

Fig. 10 AUC of FND-NS during different weeks

6.5 The impact of concept drift

The concept drift occurs when the interpretation of the data changes over time, even when the data may not have changed [10]. Concept drift is an issue that may lead to the predictions of trained classifiers becoming less accurate as time passes. For example, the news that is classified as real may become fake after some time. The news profiles and users' profiles classified as fake may also change over time (some profiles become obsolete and some are removed). Most importantly, the tactics of fake news change over time, as new ways are developed to produce fake news. These types of changes will result in the concept drift. A good model can combat the concept drift.

In this experiment, we train our model twice a month and then test on each week moving forward. At first, train on the first two weeks' data and test on the data from the third week. Next time, the model is trained on the data from the next two weeks plus the previous two weeks (e.g. week 1, 2, 3, 4) and tested on the next week (e.g. week 5) and this process (training on four weeks' data and testing on the following week) continues. We evaluate the performance of the model using AUC and report the results in Fig. 10. The reason we choose AUC here is that it is good at ranking predictions (compared to other metrics).

Overall, the concept drift impacts our FND-NS model's performance, but these changes happen slowly over time. As shown in Fig. 10, the model performance initially improves, then the performance is impacted by the concept drift in mid of March. This probably shows the arrival of unforeseen events during this time period. Once the model is trained on these events, we see a rise in performance. This is shown by a better and steady performance of the model in April. We then see a sudden rise in performance in mid-April. This is probably because, up to this point, the model has been trained on those events. After this point, the model performance becomes steady.

Overall, the results show that our model effectively deals with concept drift, as the model performance is not much impacted at a large scale during all the timesteps. In general, these results suggest that simply retraining the model every so often is enough to keep changes with fake news. We also observe that fake news's concept drift does not occur as often as in real news scenarios. The same kind of analysis is also seen in related work [3], where the authors performed extensive experiments on concept drift and concluded that the content of the fake news does not drift as abruptly as that of the real news. Though the fake news does not evolve so often as the real news, once planted, the fake news travels further, farther and broader than the real news. Therefore, it is important to detect fake news as early as possible.

6.6 The effectiveness of early fake news detection

In this experiment, we compare the performance of our model and baselines on early fake news detection. We follow the methodology of Liu and Wu [39] to define the propagation path for a news story, shown in Eq. (19):

$$\mathcal{P}(n_i, \mathcal{T}) = \prec \left(x_j, t < \mathcal{T}\right) \succ \tag{19}$$

where x_j is the observation sample and $\mathcal{T}$ is the detection deadline. The idea is that any observation data after the detection deadline $\mathcal{T}$ cannot be used for training. For example, a piece of news with timestep t means that the news is propagated t timesteps ago. Following [39] for choosing the unit for the detection deadlines, we also take the units in minutes. According to the research in fake news detection, fake news

usually takes less than an hour to spread. It is easy to detect fake news after 24 h, but earlier detection is a challenge, as discussed earlier.

In this experiment, we evaluate the performance of our model and the baselines on different detection deadlines or timesteps. To report the results, we take the observations under the detection deadlines: 15, 30, 60, 100 and 120 min, as shown in Fig. 11. For simplicity, we keep the best performing models among the available variants, e.g. among the Transformers, we keep only BERT-c, GPT-2 and VGCN-BERT based on better scores in the previous experiment (Table 5). Similarly, we keep the LG from its group [14]. Among the other fake news detection baselines, we include all (FANG, exBAKE, 2-Stage Tranf, Grover and Declare). We also keep the other baselines (TextCNN and XGBoost). We evaluate the performance of the models using the AUC measure.

The results show that our FND-NS model outperforms all the models for early fake news detection. FND-NS also shows a steady performance during all these detection deadlines. The autoregressive modelling (decoder) in FND-NS helps in modelling future values based on past observations. We have more observations listed below:

- The autoregressive models (GPT-2, Grover) perform better for early detection, probably because these models implicitly assume future values based on previous observations.
- The autoencoding models (BERT, exBAKE) show relatively lower performance than autoregressive models (GPT-2, Grover) in the early detection tasks. This is because these models are for representation learning tasks. These models perform well when more training data is fed into the models (as seen with the better performance in BERT-c in Table 5), but the deadline constraints have perhaps limited their capacity to do early detection.
- The FANG, exBAKE, 2-Stage Transf., TriFN, Declare, VGCN-BERT perform better during later time steps. This is understandable, as the model learns over time.
- The LG, TextCNN and XGBoost do not perform as good as the other baselines.

Overall, the results suggest that since linguistic features of the fake news and the social contexts on the fake news are less available during the early stage of the news, we see the lower performance of all the models during the early timesteps. Our model shows better accuracy than other models because we consider both the news and the social contexts. The news data and the social media posts contain sufficient linguistic features and are supplementary to each other, which helps us determine the fake news earlier than the other methods.

7 Limitations

Our data and approach have some limitations that we mention below:

7.1 Domain-level error analysis

The NELA-GT-19 comprises 260 news sources, which can only represent a limited amount of fake news detection analysis over a given period of time. As a result, the current results are based on the provided information. There may be other datasets that are more recent, covering different languages or target audiences, aligned with other fake news outlets (sources). They may have been missed in these results. In future, we would like to use other datasets such as NELA-GT-20 [57], or scrape more news sources from various websites and social media platforms.

Due to concept drift, the model trained on our datasets may have biases [67], causing some legitimate news sites to be incorrectly labelled. This may necessitate a re-labelling and re-evaluation process using more recent data.

According to recent research [21], the producers of disinformation change their tactics over time. We also want to see how these tactics evolve and incorporate these changes into our detection models.

At the moment, we evaluate our models on a binary classification problem. Our next step will be to consider multi-label classification, which will broaden the model's applicability to various levels of fake news detection.

7.2 Ground truth source-level labels for news articles

We have used the Media Bias Fact Check's source-level ground truth labels as proxies for the news articles. According to previous research, the choice of ground truth labels impacts downstream observations [68]. Our future research should evaluate models using different ground truth from fake and mainstream news sites. Furthermore, some sources consider more fine-grained fake news domains and more specific subcategories. Understanding whether existing models perform better in some subcategories than others can provide helpful information about model bias and weaknesses.

7.3 Weak supervision

Motivated by the success of weak supervision in similar previous works [9, 57, 69], we are currently using weak supervision to train deep neural network models effectively. In our specific scenario, applying this weak supervision scheme to the fake news classification problem also reduced the model development time from weeks to days. Moreover, despite noisy labels in weakly labelled training data, our

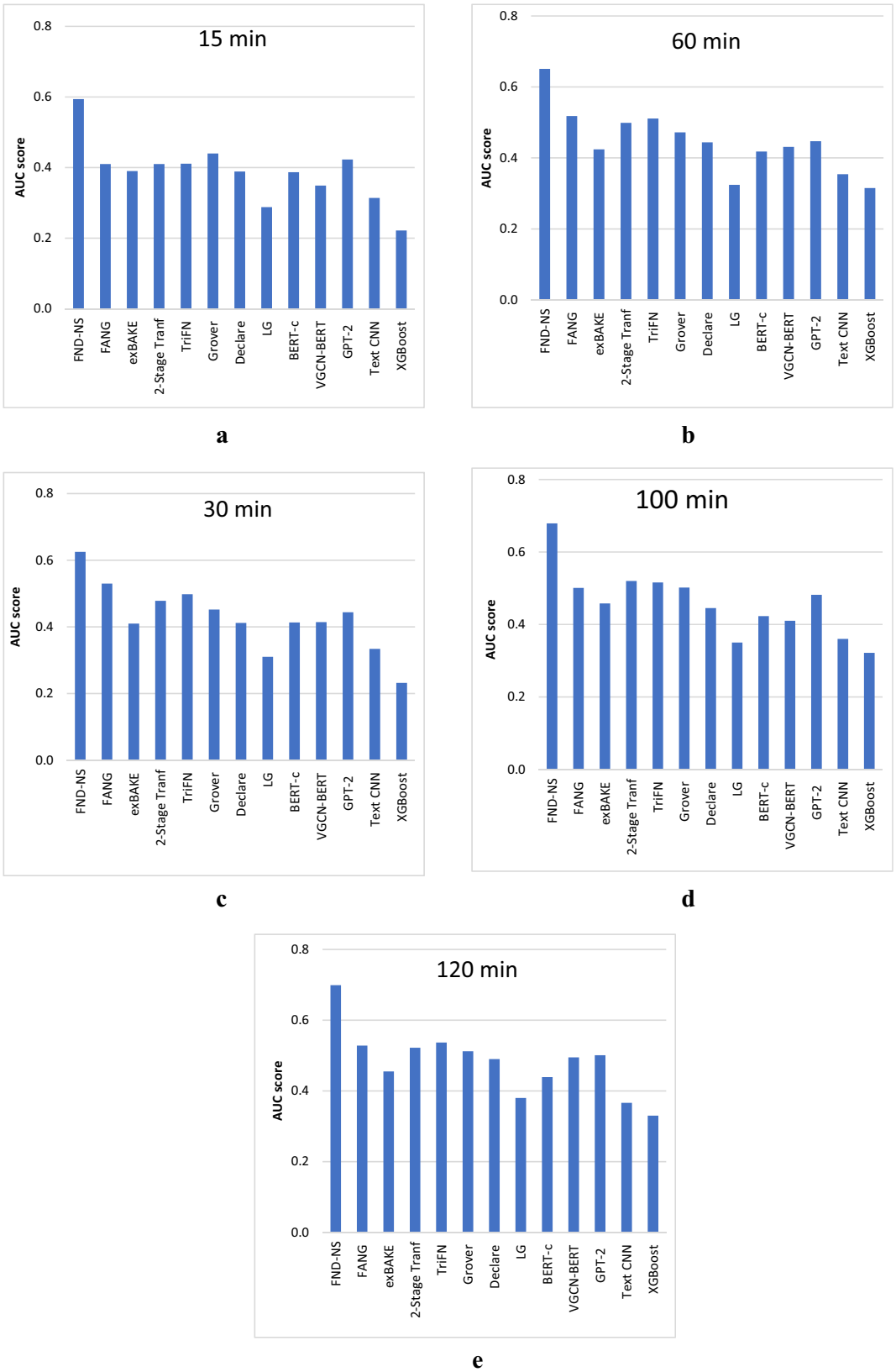

Fig. 11 **a** Fake news detection based on 15-min. deadline. **b** Fake news detection based on 30-min. deadline. **c** Fake news detection based on 30-min. deadline. **d** Fake news detection based on 100-min. deadline. **e** Fake news detection based on 120-min. deadline

results show that our proposed model performs well using weakly labelled data. However, we acknowledge that if we rely too much on weakly labelled data, the model may not generalize in all cases. This limitation can be overcome by considering manual article-level labelling, which has its own set of consequences (e.g. laborious and time-consuming process).

In future, we intend to use semi-supervised learning [70] techniques to leverage unlabelled data using structural assumptions automatically. We could also use the transfer learning technique [71] to pre-train the model only on fake news data. Furthermore, we plan to try knowledge-based weak supervision [54], which employs structured data to label a training corpus heuristically. The knowledge-based weak supervision also allows the automated learning of an indefinite number of relation extractors.

7.4 User profiles

Another limitation in this study is that we only use a small portion of users' profiles from the currently available dataset (i.e. Fakeddit). Though Fakeddit covers users' interactions over a long range of timestamps, we could only use a portion because we need to match users' interactions (social contexts) from the Fakeddit dataset with the timeline of news data from the NELA-GT-19 dataset. This limitation, however, only applies to our test scenarios. The preceding issue will not arise if a researcher or designer uses complete data to implement our model on their social media platform.

One future direction for our research is to expand the modelling of users' social contexts. First, we can include user connections in a social network in our model. User connections information can reveal the social group to which a user belongs and how the social network nurtures the spread of fake news. Second, we may incorporate user historical data to better estimate the user status, as a user's tendency to spread fake news may change over time.

Another approach is to crawl more real-world data from news sites and social media platforms (such as Twitter) to include more social contexts, which could help identify more fake news patterns. Crawling multi-modal data such as visual content and video information can also be useful for detecting fake news.

Our proposed fake news detection method can be applied to other domains, such as question-answering systems, news recommender systems [47, 72], to provide authentic news to readers.

7.5 Transfer learning

We have used transfer learning to match the tasks of fake news detection and user credibility classification. We have evidence that the MNLI can be useful for such tasks [73–75]. However, we must be cautious to avoid negative transfer, which is an open research problem.

We conducted preliminary research to understand the transferability between the source and target domains to avoid negative transfer learning. After that, we choose MNLI to extract knowledge based on appropriate transferability measures for learning fake news detection and user credibility. We understand that an entire domain (for example, from MNLI) cannot be used for transfer learning; however, for the time being, we rely on a portion of the source domain for useful learning in our target domain. The next step in this research will be to identify a more specific transfer learning domain.

7.6 User credibility

As previously stated, we transfer relevant knowledge from MNLI to user credibility, and we admit that the relatedness between the two tasks can be partial. In future, we plan to get user credibility scores through other measures such as FaceTrust [76], Alexa Rank [77], community detection algorithms [48], sentiment analysis [33] and profile ranking techniques [49].

7.7 Baselines

We include a variety of baseline methods in our experiment. While we choose algorithms with different behaviours and benchmarking schemes in mind, we must acknowledge that our baseline selection is small compared to what is available in the entire field. Our ultimate goal is to understand broad trends. We recognize that our research does not evaluate enough algorithms to make a broad statement about the whole fake news detection field.

7.8 Sequence length

We find that the difference in sequence length is the most critical factor contributing to FND-NS outperforming the benchmark models in our experiments. We acknowledge that most of the models used in this study do not support the sequence length larger than 512. We did not shorten the sequence lengths during the ablation study, but ablation of heavier features such as the news body or headline tends to reduce total sequences, which is why our model performed differently (worse than expected) during the ablation study. Nevertheless, we would like to draw the readers' attention to a trade-off between the model's predictive performance and computational cost. In our experiments, models that consider shorter sequences sacrifice some predictive performance for relatively shorter processing time. The predictive power of the classifiers usually improves by increasing the sequence length [19, 63] that we choose to work with.

7.9 Experimental set-up

Another limitation of this study is the availability of limited resources (like GPUs, memory, data storage, etc.), due to which we could not perform many experiments on other large-scale data sources. In future, we plan to expand our experiments using better infrastructure.

So far, our model is trained offline. To satisfy the real-time requirement, we just need to train and update the model periodically.

8 Conclusion

In this paper, we propose a novel deep neural framework for fake news detection. We identify and address two unique challenges related to fake news: (1) early fake news detection and (2) label shortage. The framework has three essential parts: (1) news module, (2) social contexts module and (3) detection module. We design a unique Transformer model for the detection part, which is inspired by the BART architecture. The encoder blocks in our model perform the task of representation learning. The decoder blocks predict the future behaviour based on past observations, which also helps us address the challenges of early fake news detection. The decoders depend on the working of the encoders. So, both modules are essential for fake news detection. To address the label shortage issue, we propose an effective weak supervision labelling scheme in our framework. To sum up, the inclusion of rich information from both the news and social contexts and weak labels proves helpful in building a strong fake news detection classifier.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Appendix A: Notations used in paper

Notation	Description
$N = \{n_1, n_2, \ldots, n_{\|N\|}\}$	Set of news items, $\|N\|$ is the size of the news dataset
$y_i \in \{0, 1\}$	Label $y_i = 1$ is fake news; $y_i = 0$ is real news
$U = \{u_1, u_2, \ldots, u_{\|U\|}\}$	Set of users; $\|U\|$ is the number of users
(u_0, sc_0, t_0)	A tuple, user u and social context sc during timestamp t
$C(n_i)$	Content of news
$SC(n_i) = ((u_0, sc_0, t_0), \ldots,)$	Sequence of a user' social contexts on a news item, |sc| is the size of SC
$\hat{y}(n_i) \in \{0, 1\}$	Predicted label for news item n_i
$\hat{y}(n_i) = M(n_i, SC(n_i))$	Model M predicts a label for news item based on its news features and social contexts
$X = \{x_1, x_2, \ldots, x_k\}$	Sequence of k input vector representations, k is the length
$X' = \{x'_1, x'_2, \ldots, x'_k\}$	Sequence of embedding vectors from X
$f : X'_{1:k} \rightarrow Y_{1:l}$	Mapping f from input sequence of k vectors to a sequence of l target vectors
x'', X''	Output vector representation from input x', and sequence of output vectors of x'
κ_i; v_i; q_i;$\mathbb{K}$	Key vector; value vector; query vector; set of key vectors
W_v, W_k,W_q, SoftMax	Trainable weight vectors of κ; v; q, activation function
$\overline{X}_{1:k}$; $Y_{1:l}$	Contextualized input sequence to decoder; the target vector sequence
$f_{\theta_{enc}}$; $p_{\theta_{dec}}$, $\mathcal{L}_{1:k} = \ell_1, \ldots, \ell_k$	Encoder function, decoder function; logit vector
y'; y''	Vector representation of y', and y''
<S>; $[\mathcal{S}]$;$h_{[\mathcal{S}]}$	Token in decoder; last state of the token; hien ste
$p \in [0, 1]^2$	Probability distribution over classes [0,1]
$\mathcal{W}$; b; h	Project matrix; bias term; cross-entropy function
X; X'; X'';$\overline{X}$	Input sequence to encoder; sequence generated from X; sequence generated from X'; output encoding sequence from X''
Y; Y'; Y''; Y'';$\overline{Y}$	Target sequence in decoder; sequence generated from Y; Y'; Y''; and Y''', respectively
y_j; $\hat{y}_j$	Ground truth label; model prediction

References

1. Liu, C., Wu, X., Yu, M., Li, G., Jiang, J., Huang, W., Lu, X.: A two-stage model based on BERT for short fake news detection. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 11776 LNAI, pp. 172–183 (2019). https://doi.org/10.1007/978-3-030-29563-9_17
2. Zhou, X., Zafarani, R.: A survey of fake news: fundamental theories, detection methods, and opportunities. ACM Comput. Surv. (2020). https://doi.org/10.1145/3395046
3. Horne, B.D., NØrregaard, J., Adali, S.: Robust fake news detection over time and attack. ACM Trans. Intell. Syst. Technol. (2019). https://doi.org/10.1145/3363818
4. Shu, K., Wang, S., Liu, H.: Beyond news contents: The role of social context for fake news detection. In: WSDM 2019—Proceedings of 12th ACM International Conference on Web Search Data Mining, vol. 9, pp. 312–320 (2019). https://doi.org/10.1145/3289600.3290994
5. Liu, Y., Wu, Y.F.B.: FNED: a deep network for fake news early detection on social media. ACM Trans. Inf. Syst. (2020). https://doi.org/10.1145/3386253
6. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. Science **359**, 1146–1151 (2018)
7. Jwa, H., Oh, D., Park, K., Kang, J.M., Lim, H.: exBAKE: automatic fake news detection model based on Bidirectional Encoder Representations from Transformers (BERT). Appl. Sci. **9**, 4062 (2019). https://doi.org/10.3390/app9194062
8. Popat, K., Mukherjee, S., Yates, A., Weikum, G.: Declare: debunking fake news and false claims using evidence-aware deep learning. arXiv Preprint. http://arxiv.org/abs/1809.06416. (2018)
9. Wang, Y., Yang, W., Ma, F., Xu, J., Zhong, B., Deng, Q., Gao, J.: Weak supervision for fake news detection via reinforcement learning. In: AAAI 2020—34th AAAI Conference on Artificial Intelligence, pp. 516–523 (2020)
10. Hoens, T.R., Polikar, R., Chawla, N.: V: Learning from streaming data with concept drift and imbalance: an overview. Prog. Artif. Intell. **1**, 89–101 (2012)
11. Kaliyar, R.K., Goswami, A., Narang, P., Sinha, S.: FNDNet—a deep convolutional neural network for fake news detection. Cogn. Syst. Res. **61**, 32–44 (2020). https://doi.org/10.1016/j.cogsys.2019.12.005
12. Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., Choi, Y.: Defending against neural fake news. Neurips (2020)
13. Yang, S., Shu, K., Wang, S., Gu, R., Wu, F., Liu, H.: Unsupervised fake news detection on social media: a generative approach. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 5644–5651 (2019)
14. Mohammadrezaei, M., Shiri, M.E., Rahmani, A.M.: Identifying fake accounts on social networks based on graph analysis and classification algorithms. Secur. Commun. Netw. (2018). https://doi.org/10.1155/2018/5923156
15. Nguyen, V.H., Sugiyama, K., Nakov, P., Kan, M.Y.: FANG: leveraging social context for fake news detection using graph representation. Int. Conf. Inf. Knowl. Manag. Proc. (2020). https://doi.org/10.1145/3340531.3412046
16. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv (2019). https://doi.org/10.18653/v1/2020.acl-main.703
17. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv Preprint. http://arxiv.org/abs/1810.04805. (2018)
18. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog. **1**, 9 (2019)
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
20. Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., Schiele, B.: Latent embeddings for zero-shot classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 69–77 (2016)
21. Gruppi, M., Horne, B.D., Adalı, S.: NELA-GT-2019: A large multi-labelled news dataset for the study of misinformation in news articles. arXiv Preprint. http://arxiv.org/abs/2003.08444v2 (2020)
22. Nakamura, K., Levy, S., Wang, W.Y.: r/fakeddit: a new multimodal benchmark dataset for fine-grained fake news detection. arXiv Preprint. http://arxiv.org/abs/1911.03854 (2019)
23. 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. Big Data **8**, 171–188 (2020). https://doi.org/10.1089/big.2020.0062
24. Pizarro, J.: Profiling bots and fake news spreaders at PAN'19 and PAN'20: bots and gender profiling 2019, profiling fake news spreaders on Twitter 2020. In: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), pp. 626–630 (2020)
25. Horne, B.D., Dron, W., Khedr, S., Adali, S.: Assessing the news landscape: a multi-module toolkit for evaluating the credibility of news. In: The Web Conference 2018—Companion of the World Wide Web Conference, WWW 2018, pp. 235–238 (2018)
26. Przybyla, P.: Capturing the style of fake news. Proc. AAAI Conf. Artif. Intell. **34**, 490–497 (2020). https://doi.org/10.1609/aaai.v34i01.5386
27. Silva, R.M., Santos, R.L.S., Almeida, T.A., Pardo, T.A.S.: Towards automatically filtering fake news in Portuguese. Expert Syst. Appl. **146**, 113199 (2020). https://doi.org/10.1016/j.eswa.2020.113199
28. Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., Stein, B.: A stylometric inquiry into hyperpartisan and fake news. arXiv Preprint. http://arxiv.org/abs/1702.05638. (2017)
29. Horne, B., Adali, S.: This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In: Proceedings of the International AAAI Conference on Web and Social Media (2017)
30. Zhou, X., Wu, J., Zafarani, R.: SAFE: similarity-aware multi-modal fake news detection. Adv. Knowl. Discov. Data Min. **12085**, 354 (2020)
31. De Maio, C., Fenza, G., Gallo, M., Loia, V., Volpe, A.: Cross-relating heterogeneous Text Streams for Credibility Assessment. In: IEEE Conference on Evolving and Adaptive Intelligent Systems, 2020-May, (2020). https://doi.org/10.1109/EAIS48028.2020.9122701
32. Wanda, P., Jie, H.J.: DeepProfile: finding fake profile in online social network using dynamic CNN. J. Inf. Secur. Appl. (2020). https://doi.org/10.1016/j.jisa.2020.102465
33. Naseem, U., Razzak, I., Khushi, M., Eklund, P.W., Kim, J.: Covidsenti: a large-scale benchmark Twitter data set for COVID-19 sentiment analysis. IEEE Trans. Comput. Soc. Syst. (2021)
34. Naseem, U., Razzak, I., Eklund, P.W.: A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. Multimed. Tools Appl. **80**, 1–28 (2020)
35. Naseem, U., Razzak, I., Hameed, I.A.: Deep context-aware embedding for abusive and hate speech detection on Twitter. Aust. J. Intell. Inf. Process. Syst. **15**, 69–76 (2019)
36. Huang, Q., Zhou, C., Wu, J., Liu, L., Wang, B.: Deep spatial–temporal structure learning for rumor detection on Twitter.

Neural Comput. Appl. (2020). https://doi.org/10.1007/s00521-020-05236-4
37. Jiang, S., Chen, X., Zhang, L., Chen, S., Liu, H.: User-characteristic enhanced model for fake news detection in social media. In: CCF International Conference on Natural Language Processing and Chinese Computing, pp. 634–646 (2019)
38. Qian, F., Gong, C., Sharma, K., Liu, Y.: Neural user response generator: fake news detection with collective user intelligence. In: IJCAI International Joint Conference on Artificial Intelligence, pp. 3834–3840 (2018)
39. Liu, Y., Wu, Y.F.B.: Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, pp. 354–361 (2018)
40. Cao, J., Qi, P., Sheng, Q., Yang, T., Guo, J., Li, J.: Exploring the role of visual content in fake news detection. Disinformation, Misinformation, Fake News Social Media, pp. 141–161 (2020)
41. Jin, Z., Cao, J., Guo, H., Zhang, Y., Luo, J.: Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In: Proceedings of the 25th ACM International Conference on Multimedia, pp. 795–816 (2017)
42. Karimi, H., Roy, P., Saba-Sadiya, S., Tang, J.: Multi-source multi-class fake news detection. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 1546–1557 (2018)
43. Wu, X., Lode, M.: Language models are unsupervised multitask learners (summarization). OpenAI Blog. **1**, 1–7 (2020)
44. Vijjali, R., Potluri, P., Kumar, S., Teki, S.: Two stage transformer model for covid-19 fake news detection and fact checking. arXiv Preprint. http://arxiv.org/abs/2011.13253. (2020)
45. Anderson, C.W.: News ecosystems. SAGE Handb. Digit. J. 410–423 (2016)
46. Wang, B., Shang, L., Lioma, C., Jiang, X., Yang, H., Liu, Q., Simonsen, J.G.: On position embeddings in BERT. In: International Conference on Learning Representations (2021)
47. Raza, S., Ding, C.: News recommender system: a review of recent progress, challenges, and opportunities. Artif. Intell. Rev. (2021). https://doi.org/10.1007/s10462-021-10043-x
48. Papadopoulos, S., Kompatsiaris, Y., Vakali, A., Spyridonos, P.: Community detection in social media. Data Min. Knowl. Discov. **24**, 515–554 (2012)
49. Abu-Salih, B., Wongthongtham, P., Chan, K.Y., Zhu, D.: CredSaT: credibility ranking of users in big social data incorporating semantic analysis and temporal factor. J. Inf. Sci. **45**, 259–280 (2019). https://doi.org/10.1177/0165551518790424
50. Williams, A., Nangia, N., Bowman, S.R.: A broad-coverage challenge corpus for sentence understanding through inference. arXiv Preprint. http://arxiv.org/abs/1704.05426 (2017)
51. Pushp, P.K., Srivastava, M.M.: Train once, test anywhere: zero-shot learning for text classification. arXiv Preprint. http://arxiv.org/abs/1712.05972 (2017)
52. Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., Hon, H.-W.: Unified language model pre-training for natural language understanding and generation. arXiv Preprint. http://arxiv.org/abs/1905.03197 (2019)
53. Helmstetter, S., Paulheim, H.: Weakly supervised learning for fake news detection on Twitter. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 274–277 (2018)
54. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 541–550 (2011)
55. Baly, R., Karadzhov, G., Alexandrov, D., Glass, J., Nakov, P.: Predicting factuality of reporting and bias of news media sources. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018, pp. 3528–3539 (2020)
56. Nørregaard, J., Horne, B.D., Adalı, S.: NELA-GT-2018: A large multi-labelled news dataset for the study of misinformation in news articles. In: Proceedings of 13th International Conference on Web and Social Media, ICWSM 2019, pp. 630–638 (2019). https://doi.org/10.7910/DVN/ULHLCB
57. Horne, Benjamin; Gruppi, M.: NELA-GT-2020: A Large Multi-Labelled News Dataset for The Study of Misinformation in News Articles. arXiv Preprint. http://arxiv.org/abs/2102.04567. (2021). https://doi.org/10.7910/DVN/CHMUYZ
58. Drummond, C., Holte, R.C., et al.: C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: Workshop on Learning from Imbalanced Datasets II, pp. 1–8 (2003)
59. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv Preprint. http://arxiv.org/abs/1711.05101 (2017)
60. Lu, Z., Du, P., Nie, J.Y.: VGCN-BERT: augmenting BERT with graph embedding for text classification. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 369–382 (2020)
61. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems, pp. 5753–5763 (2019)
62. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv Preprint. http://arxiv.org/abs/1910.01108 (2019)
63. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: the long-document transformer. arXiv Preprint. http://arxiv.org/abs/2004.05150 (2020)
64. Kim, Y.: Convolutional neural networks for sentence classification. arXiv Preprint. http://arxiv.org/abs/1408.5882 (2014)
65. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., et al.: Xgboost: extreme gradient boosting. R Package version 0.4-2.1, (2015)
66. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 7370–7377 (2019)
67. Raza, S., Ding, C.: News recommender system considering temporal dynamics and news taxonomy. In: Proceedings—2019 IEEE International Conference on Big Data, Big Data 2019, pp. 920–929. Institute of Electrical and Electronics Engineers Inc. (2019)
68. Bozarth, L., Saraf, A., Budak, C.: Higher ground? How groundtruth labeling impacts our understanding of fake news about the 2016 US presidential nominees. In: Proceedings of the International AAAI Conference on Web and Social Media, pp. 48–59 (2020)
69. Wang, Y., Sohn, S., Liu, S., Shen, F., Wang, L., Atkinson, E.J., Amin, S., Liu, H.: A clinical text classification paradigm using weak supervision and deep representation. BMC Med. Inform. Decis. Mak. **19**, 1–13 (2019). https://doi.org/10.1186/s12911-018-0723-6
70. Zhu, X., Goldberg, A.B.: Introduction to semi-supervised learning. Synth. Lect. Artif. Intell. Mach. Learn. **3**, 1–130 (2009)
71. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C.: A survey on deep transfer learning. In: International Conference on Artificial Neural Networks, pp. 270–279 (2018)
72. Raza, S., Ding, C.: A Regularized Model to Trade-off between Accuracy and Diversity in a News Recommender System. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 551–560 (2020)
73. Bhuiyan, M., Zhang, A., Sehat, C., Mitra, T.: Investigating "who" in the crowdsourcing of news credibility. In: Computational Journalism Symposium (2020)
74. Popat, K., Mukherjee, S., Strötgen, J., Weikum, G.: Credibility assessment of textual claims on the web. In: International Con-

ference on Information and Knowledge Managenent Proceedings, 24–28-October-2016, pp. 2173–2178 (2016). https://doi.org/10.1145/2983323.2983661
75. Yang, K.-C., Niven, T., Kao, H.-Y.: Fake News Detection as Natural Language Inference. arXiv Preprint. http://arxiv.org/abs/1907.07347 (2019)
76. Sirivianos, M., Kim, K., Yang, X.: FaceTrust: Assessing the credibility of online personas via social networks. In: Proceedings of 4th USENIX Conferences on Hot Topics in Security (2009)
77. Thakur, A., Sangal, A.L., Bindra, H.: Quantitative measurement and comparison of effects of various search engine optimization parameters on Alexa Traffic Rank. Int. J. Comput. Appl. **26**, 15–23 (2011)