Reply-Aided Detection of Misinformation via Bayesian Deep Learning

Conference Paper · February 2019

DOI: 10.1145/3308558.3313718

4 authors, including:

Qiang Zhang
University College London
16 PUBLICATIONS **177** CITATIONS
SEE PROFILE

Aldo Lipani
University College London
69 PUBLICATIONS **299** CITATIONS
SEE PROFILE

Emine Yilmaz
University College London
102 PUBLICATIONS **2,084** CITATIONS
SEE PROFILE

Reply-Aided Detection of Misinformation via Bayesian Deep Learning

Qiang Zhang
University College London
London, United Kingdom
qiang.zhang.16@ucl.ac.uk

Aldo Lipani
University College London
London, United Kingdom
aldo.lipani@ucl.ac.uk

Shangsong Liang
Sun Yat-sen University
Guangzhou, China
liangshangsong@gmail.com

Emine Yilmaz
University College London
London, United Kingdom
emine.yilmaz@ucl.ac.uk

ABSTRACT

Social media platforms are a plethora of misinformation and its potential negative influence on the public is a growing concern. This concern has drawn the attention of the research community on developing mechanisms to detect misinformation. The task of misinformation detection consists of classifying whether a claim is *True* or *False*. Most research concentrates on developing machine learning models, such as neural networks, that outputs a single value in order to predict the veracity of a claim. One of the major problem faced by these models is the inability of representing the uncertainty of the prediction, which is due incomplete or finite available information about the claim being examined. We address this problem by proposing a Bayesian deep learning model. The Bayesian model outputs a distribution used to represent both the prediction and its uncertainty. In addition to the claim content, we also encode auxiliary information given by people's replies to the claim. First, the model encodes a claim to be verified, and generate a *prior belief* distribution from which we sample a latent variable. Second, the model encodes all the people's replies to the claim in a temporal order through a Long Short Term Memory network in order to summarize their content. This summary is then used to update the prior belief generating the *posterior belief*. Moreover, in order to train this model, we develop a Stochastic Gradient Variational Bayes algorithm to approximate the analytically intractable posterior distribution. Experiments conducted on two public datasets demonstrate that our model outperforms the state-of-the-art detection models.

CCS CONCEPTS

• **Information systems** → **Web mining**; • **Computing methodologies** → *Information extraction*.

KEYWORDS

misinformation detection, bayesian analysis, deep learning

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313718

ACM Reference Format:
Qiang Zhang, Aldo Lipani, Shangsong Liang, and Emine Yilmaz. 2019. Reply-Aided Detection of Misinformation via Bayesian Deep Learning. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313718

1 INTRODUCTION

Although the digital news consumption has increased in the last decade, the increasing amount of misinformation and fake news has not certainly proven its quality. Different from traditional media where news are published by reputable organizations, online news on social media platforms such as Facebook and Twitter are shared by individuals and/or organizations without a careful checking or with malicious intents. In Figure 1 we show a false claim posted on Twitter about an alleged shooting in Ottawa. While some users showed surprise and asked for further clarifications in their replies, other users believed the claim and re-tweeted it as if it was true. This misinformation, when done on a large scale can influence the public by depicting a false picture of reality. Hence, detecting misinformation effectively has become one of the biggest challenges faced by social media platforms [17, 27].

A valuable attempt at rectifying this epidemic of false claims has been tackled by some news websites, such as: Snopes[1], Polifact[2], and Emergent[3], which have employed professional journalists to manually check and verify every potential false news. However, such manual approach is very expensive and way too slow to be able to check all the daily generated claims appearing on the web. Thus, making automatic tools is in great need to speed up this verification process.

In this paper, we tackle the automatic misinformation detection task, which consists in classifying whether a claim is *True* or *False*. Most existing models employ feature engineering or deep learning to extract features from claims' content and auxiliary information such as people's replies. However, these models generate deterministic mappings to capture the difference between true or false claims. A major limitation of these models is their inability to represent uncertainty caused by incomplete or finite available data about the claim being examined.

[1] https://www.snopes.com/
[2] https://www.politifact.com/
[3] http://www.emergent.info/

Figure 1: An example of a false claim and people's replies to it. From the replies, 231 users chose to trust the claim and re-tweeted it as if it was true, while only 4 users asked for further clarifications.

We address this problem by proposing a Bayesian deep learning model, which incorporates stochastic factors to capture complex relationships between the latent distribution and the observed variables. The proposed model makes use of the claim content and replies content. First, to represent the claim content we employ a neural model to extract textual features from claims. To deal with the ambiguity of the language used in claims and obtain salient credibility information, the model generates a latent distribution based on the extracted linguistic features. Since no auxiliary information has been used so far, we interpret this latent distribution as a *prior belief* of the claim being true. Second, to extract auxiliary information from people's replies content, we rank all the replies of the claim in temporal order, and summarize them using a Long Short Term Memory neural network (LSTM). Finally, after updating the *prior belief* with the aid of the LSTM output, the model computes the veracity prediction and its uncertainty. This updated prior belief distribution is interpreted as the *posterior belief*.

In order to train the proposed Bayesian deep learning model, due to the analytical intractability of the posterior distribution, we develop a Stochastic Gradient Variational Bayes (SGVB) algorithm. A tractable Evidence Lower BOund (ELBO) objective function of our model is derived to approximate the intractable distribution. The model is optimized along the direction of maximizing the ELBO objective function.

Our model inherit two advantages: first of all, the model incorporates a latent distribution, which enables to represent uncertainty and promote robustness; second, the Bayesian model formulates all of its prior knowledge about a claim being examined in the form of a prior, which can be updated by more added auxiliary information generating more accurate detection results. To sum up, the proposed model advances state-of-the-art methods in four aspects:

(1) An effective representation of uncertainty due to incomplete/finite available data;
(2) A temporal order-based approach to extract auxiliary information from people's replies;
(3) A SGVB algorithm to infer latent distributions;
(4) A systematic experimentation of our model on two real-world datasets.

The remainder of the paper is organized as follows: § 2 summarizes the related work; § 3 defines the misinformation detection task; § 4 details the proposed Bayesian deep learning model; § 5 derives the Stochastic Gradient Variational Bayes optimization algorithm; § 6 describes the used datasets and experimental setup; § 7 is devoted to experimental results, and; § 8 concludes the paper.

2 RELATED WORK

Misinformation has been existing for centuries in different forms of media, such as printed newspaper and television. Recently, online social media platforms are also suffering from the same issues. Recent work on misinformation detection have tried to understand the differences between true and false claims in various aspects: claim content, information source, multimedia such as affiliated images and videos, and other users' engagement.

2.1 Textual Content

The text of a claim can provide linguistic features to help predict its veracity. Since misinformation and false claims are created for financial or political purposes rather than to report an objective event, they often contain opinionated or inflammatory language [6]. In order to reveal linguistic differences between true and false claims, lexical and syntactic features at character, word, sentence and document level have been exploited [1, 11, 33, 36]. Wawer et al. [43] compute psycholinguistic features using a bag-of-words paradigm. Rashkin et al. [34] compare the language of true claims with that of satire, hoaxes, and propaganda to find linguistic characteristics of untrustworthy text. Kakol et al. [21] construct a content credibility corpus and examine a list of language factors that might affect web content credibility based on which a predictive model is developed. Bountouridis et al. [3] compare heterogeneous articles of the same story and reveal that pieces of information cross-referenced are more likely to be credible. Derczynski et al. [9] extract features from claim tweets including bag-of-words, presence of URLs, and presence of hashtags. A Support Vector Machine (SVM) is then used to distinguish between true and false claims. Guacho et al. [14] leverages a tensor decomposition to derive concise claim embeddings that capture contextual information from each claim; and uses these embeddings to create a claim-by-claim graph on which the labels propagate. Textual content has been empirically proven to be a strong indicator of claim veracity, and thus can be used as a prior probability.

2.2 Source Credibility Analysis

The credibility analysis of the sources of a claim is an important auxiliary information. As misinformation is usually published by unbelievable individuals or automatic bots, credibility plays a crucial role in message communication [18, 32]. Accurate and timely discrimination of such accounts inhibits the proliferation of misinformation at an early stage. Tseng and Fogg [40] identify two components of source credibility, namely trustworthiness and expertise. Trustworthiness is generally taken to mean truthful, unbiased and well intentioned. Expertise instead is understood as knowledgeable, experienced and competent. Thus, features that can reveal the trustworthiness and expertise of information sources are strong indicators of source credibility. With the aid of information source Thomson et al. [39] examine the credibility of tweets related to the Fukushima Daiichi nuclear disaster in Japan. They found that tweets from highly credible institutions and individuals are mostly correct. Useful account features can be derived from the account demographics, such as integrity of personal information, the number of followers and followees [5]. Besides, aggregating a group of account features are indicative, since spreaders of true and false claims might come from different communities [44], such as the percentage of verified user accounts [28] and the average number of followers [26]. However, account demographics can easily be altered to decrease the similarity between credible and incredible sources.

2.3 Multimedia Features

Multimedia features have been shown to be an important manipulator for propaganda based on misinformation [4]. As we have characterized, online misinformation exploits the individual vulnerabilities of people and thus often relies on sensational or even fake images to provoke anger or other emotional response of consumers. Visual-based features are extracted from images and videos to capture the different characteristics of misinformation. Faking images are identified based on various user-level and tweet-level hand-crafted features [15]. Recently, various visual and statistical features have been extracted for news verification [20]. Yang et al. [45] develop a convolutional neural network to extract text and visual features simultaneously. Visual features include clarity score, coherence score, diversity score, and clustering score. Statistical features include count, image ratio, multi-image ratio, hot image ratio, long image ratio, etc. This approach suffers from the problem that some misinformation on social media does not contain multimedia content.

2.4 Social Engagement

The news spreading process over time on social media involves user-driven engagement. Auxiliary information can also be derived from such engagement to improve the claim veracity detection. Ma et al. [29] propose to learn discriminative features by following non-sequential propagation structure of tweets. A top-down and a bottom-up recursive neural networks are proposed to predict claim veracity. Glenski et al. [12] seek to better understand how users react to trusted and deceptive news sources across two popular, and very different, social media platforms. Significant differences have been observed in the speed and the type of reactions between trusted and deceptive news sources on Twitter, but far smaller differences on Reddit. People react to a piece of claim by expressing their stances or emotions in social media posts. Stances can be categorized as supportive, opposing, and neutral, which can be used to infer claim veracity [19, 46, 47]. Kochkina et al. [25] propose a neural multi-task model that leverages the relationship between veracity detection and stance detection in a joint learning setup. Another common post feature is the topic distribution that indicates the central point of relevant affairs, which is derived by topic models [2]. Post features are expanded in two ways: via aggregation with relevant posts for a specific affair, and via temporal evolution of post features. The first way relies on the "wisdom of crowds" to locate potential misinformation [5], while the second way captures the periodic fluctuations of shock cycles [26] or temporal pattern of user activities, such as the number of engaged users and time intervals between engagements [37]. Yet, semantic coherence and temporal changes between users' replies are not fully explored by existing methods.

3 PROBLEM STATEMENT

The task of misinformation detection is to predict the news' veracity of claims, given their content and their people's replies.

Let $C = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_N\}$ be a set of N claims. The claim $\boldsymbol{c}_i$ is commented by a set of M user replies $\mathcal{D}_i = \{\boldsymbol{d}_{i,1}, \boldsymbol{d}_{i,2}, \ldots, \boldsymbol{d}_{i,M}\}$. We use y_i to denote the binary veracity label of the claim $\boldsymbol{c}_i$, which could be either $y_i = 1$ for true or $y_i = 0$ for false. The tuple of a claim and people's replies, i.e., $\{\boldsymbol{c}_i, \mathcal{D}_i\}$, forms a data instance to predict the claim veracity y_i. For the sake of clarity, in the following, we will omit the subscripts i when describing a single instance: $\{\boldsymbol{c}, \mathcal{D}, y\}$.

4 BAYESIAN DEEP LEARNING

In this section, we present our proposed Bayesian deep learning model that effectively integrates claim and people's replies. We will first introduce how to encode claim content with deep learning and generate a latent distribution that is interpreted as a prior belief of claim veracity. We then describe the temporal-ordered approach to encode people's replies, which captures semantic variation along the time line. Finally, we correct the prior belief with the aid of people's replies, the result of which process is interpreted as the posterior belief of claim veracity. Figure 2 describes the proposed model.

4.1 Encoding a Claim

As content are strong indicators of claim veracity [42], we apply deep learning to extract linguistic features from the claim $\boldsymbol{c}$. To avoid the ambiguity of claims and obtain salient credibility information, we generate a latent distribution based on the extracted linguistic features. The output of this claim encoder is the prior belief of the veracity of the claim.

Let each claim $\boldsymbol{c}$ be a sequence of discrete words or tokens, i.e., $\boldsymbol{c} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_l]$, where $\boldsymbol{w}_l \in \mathbb{R}^d$ is a d-dimensional word embedding vector. Based on the sequence of word embeddings, textual features are extracted via a Bidirectional Long Short Term Memory (BiLSTM) neural network [13]. The BiLSTM captures long and short semantic dependencies both from previous time steps and future time steps via forward and backward states.The BiLSTM

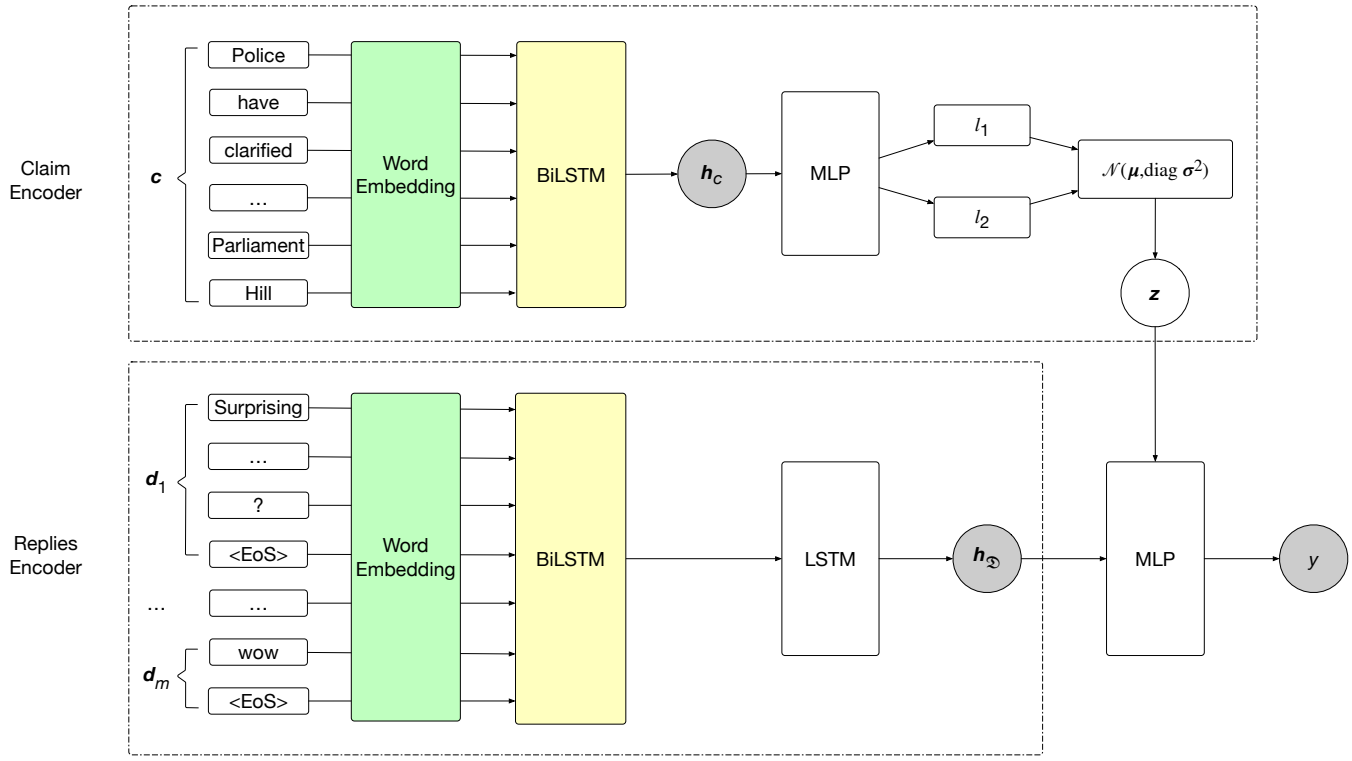


Figure 2: Framework of the Bayesian deep learning model. The framework consists of two parts, the claim encoder (§ 4.1) and the replies encoder (§ 4.2), the concatenation of which determines the *posterior belief* of claim veracity. Blocks and nodes represent computation modules and variables. Grey nodes are observed variables while blank nodes are latent variables (similarly with Figure 3). Note that blocks of the same color denote the same module.

takes as input $\boldsymbol{c}$, converts the sequence of word embeddings into a dense representation, and outputs the concatenation of two hidden states capturing past and future information:

$$\boldsymbol{h}_c = \text{BiLSTM}(\boldsymbol{c}), \tag{1}$$

where $\boldsymbol{h}_c$ denote the concatenated hidden states.

To avoid the ambiguity of claims, instead of a deterministic non-linear transformation, we generate a latent distribution, from which we sample a latent stochastic variable $\boldsymbol{z}$. To embed linguistic information into the latent variable, we set the latent variable to be conditional on $\boldsymbol{h}_c$:

$$\boldsymbol{z} \sim p_\theta(\boldsymbol{z}|\boldsymbol{h}_c), \tag{2}$$

where p is a latent distribution and θ denotes the non-linear transformation of $\boldsymbol{h}_c$ to generate the parameters of p. This non-linear transformation is essential to capture higher level representations of $\boldsymbol{h}_c$; we implement this non-linear transformation via a Multi-Layer Perceptron (MLP).

We assume that the latent variable $\boldsymbol{z}$ is continuous and follows a multivariate Gaussian distribution. The variable $\boldsymbol{z}$ is parameterized as follows:

$$p_\theta(\boldsymbol{z}|\boldsymbol{h}_c) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_\theta, \text{diag}(\boldsymbol{\sigma}_\theta^2)), \tag{3}$$

where $\boldsymbol{\mu}_\theta$ and $\text{diag}(\boldsymbol{\sigma}_\theta^2)$ are the mean and the covariance matrix of the multivariate Gaussian distribution. Since the variable $\boldsymbol{z}$ is conditional on the the claim hidden states $\boldsymbol{h}_c$, we derive these two parameters of the Gaussian distribution from $\boldsymbol{h}_c$ through a deep neural network:

$$\boldsymbol{\pi}_\theta = f_\theta(\boldsymbol{h}_c), \tag{4}$$

$$\boldsymbol{\mu}_\theta = l_1(\boldsymbol{\pi}_\theta) \quad \ln(\boldsymbol{\sigma}_\theta) = l_2(\boldsymbol{\pi}_\theta), \tag{5}$$

where f_θ denotes a MLP, l_1 and l_2 denote two Linear Transformations (LT). Since LT can generate negative values, to produce σ_θ we exponentiate the result of l_2.

In order to make μ_θ and σ_θ differentiable and backpropagate the loss through the latent distribution (p), the following reparameterization trick is used:

$$\boldsymbol{z} = \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma}_\theta + \boldsymbol{\mu}_\theta \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{6}$$

where $\mathbf{0}$ is a vector of zeros and $\mathbf{I}$ is the identity matrix. By making use of the latent variable ($\boldsymbol{z}$), our model is able to capture complex noisy patterns in the data.

4.2 Encoding People's Replies

We now present the people's replies encoder to obtain auxiliary information. This auxiliary information is claim-specific and is used to generate the posterior belief by correcting the prior belief of the claim veracity.

Replies on social media platforms are listed along the time line as shown in Figure 1, where the earliest reply appears at the top of the list. Truth about an event can be gradually manifest as more

evidence emerges, thus we assume that the latest replies tend to be more reliable and more important than the earlier replies. Based on this assumption, we design a two-layer recurrent neural network to encode replies: the first-layer applies a BiLSTM to summarize the semantic information of each reply and the second-layer applies a LSTM to capture the temporal semantic variation of the replies.

Given a claim $\boldsymbol{c}$ commented by a list of replies $\mathfrak{D} = \{\boldsymbol{d}_1, \ldots \ldots, \boldsymbol{d}_m, \ldots, \boldsymbol{d}_M\}$, these replies are ranked based on their temporal order. The content of a $\boldsymbol{d}_m$ consists of a sequence of words $\boldsymbol{d} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k]$. To project the claim and replies into the same semantic space, we use the same pre-trained word embeddings for both claims and replies. Hence, $\boldsymbol{w}_k \in \mathbb{R}^d$ is a d-dimensional vector such as the word embedding vector used to encoding the claim. For the sake of semantic coherence, we also employ the same BiLSTM to encode both the claim and its replies. Take the reply $\boldsymbol{d}_m \in \mathfrak{D}$ for example, the concatenation of hidden states from forward and backward directions is denoted as:

$$\boldsymbol{h}_{d_m} = \text{BiLSTM}(\boldsymbol{d}_m), \tag{7}$$

where $\boldsymbol{h}_{d_m}$ is the summary of the reply $\boldsymbol{d}_m$.

In order to capture the semantic information of all replies, we sequentially input the concatenated hidden states of each reply into a LSTM. We use the LSTM rather than a BiLSTM because the former gives high weights to recent input, which matches our assumption on the relative importance of the latest reply. Specifically, the LSTM takes the hidden states of each reply as input in a sequential way:

$$\boldsymbol{h}_{\mathfrak{D}} = \text{LSTM}(\boldsymbol{h}_d^M), \tag{8}$$

where $\boldsymbol{h}_d^M = \{\boldsymbol{h}_{d_1}, \ldots, \boldsymbol{h}_{d_m}, \ldots, \boldsymbol{h}_{d_M}\}$.

The last hidden state $\boldsymbol{h}_{\mathfrak{D}}$ contains both the linguistic information of each reply and the semantic changes between replies along the time line. This $\boldsymbol{h}_{\mathfrak{D}}$ is then used to generate the posterior belief by correcting the prior belief of the claim veracity.

4.3 Veracity Modeling

In § 4.1, we developed a prior belief of the claim veracity. In this section we show how to correct this prior belief by including its replies.

The posterior belief is generated by combining the claim and reply information via a MLP. The strong non-linearity of MLPs make them suitable to find complex relationships between the claim and its replies. Specifically, the MLP input is the latent claim variable $\boldsymbol{z}$ concatenated to the hidden state of replies $\boldsymbol{h}_{\mathfrak{D}}$:

$$y = \text{MLP}(\boldsymbol{z}, \boldsymbol{h}_{\mathfrak{D}}). \tag{9}$$

This is the final prediction of our Bayesian deep learning model for misinformation detection.

5 OPTIMIZATION

The stochastic variables of our model are non-linear and non-conjugate [41]. Hence, the posterior distribution cannot be derived analytically. To approximate the posterior distribution, we construct an inference model parameterized by ϕ to approximate the intractable true posterior $p_\theta(\boldsymbol{z}|\boldsymbol{h}_c)$; then we derive an objective function to measure how well $p_\theta(\boldsymbol{z}|\boldsymbol{h}_c)$ is approximated; finally we exploit the Stochastic Gradient Variational Bayes (SGVB) method [24, 35] to learn the inference model parameters ϕ together with the generative model parameters θ. Figure 3 shows the graphical representation of the generative model and the inference model.

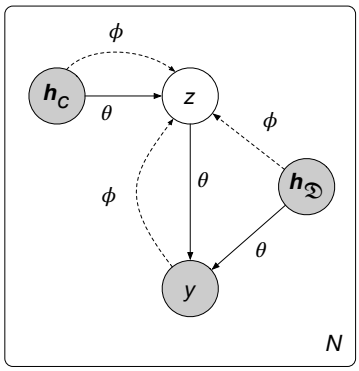


Figure 3: The directed graphical model. Grey nodes represent observed variables while blank nodes represent latent variables. Solid lines denote the generative model $p_\theta(z|\boldsymbol{h}_c)p(y|z, \boldsymbol{h}_{\mathfrak{D}})$, dashed lines denote the variational approximation $q_\phi(z|y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}})$ to the intractable posterior $p_\theta(z|\boldsymbol{h}_c)$. The variational parameters are learned together with the generative model parameters.

5.1 Inference Model

Following the neural variational inference approach [24], we construct an inference model (as in Fig. 3) parameterized by ϕ to compute an approximated posterior distribution, called variational distribution. Given the observed variables, we define a variational distribution $q_\phi(\boldsymbol{z}|y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}})$ to approximate the true posterior distribution $p_\theta(\boldsymbol{z}|\boldsymbol{h}_c)$. Like for the Variational Auto Encoder (VAE) [24], similarly to Eq. 3 for $p_\theta(\boldsymbol{z}|\boldsymbol{h}_c)$, the variational distribution is chosen to be a multivariate Gaussian distribution:

$$q_\phi(\boldsymbol{z}|y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_\phi, \text{diag}(\boldsymbol{\sigma}_\phi^2)), \tag{10}$$

where $\boldsymbol{\mu}_\phi$ and $\text{diag}(\boldsymbol{\sigma}_\phi^2)$ are the mean and the covariance matrix of the multivariate Gaussian distribution. We use a deep neural network to derive these two parameters from the observed variables:

$$\boldsymbol{\pi}_\phi = f_\phi(y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}}), \tag{11}$$

$$\boldsymbol{\mu}_\phi = l_3(\boldsymbol{\pi}_\phi), \quad \ln(\boldsymbol{\sigma}_\phi) = l_4(\boldsymbol{\pi}_\phi), \tag{12}$$

where f_ϕ denotes a MLP, and l_3 and l_4 denote two LTs. Note that in the inference model to compute $\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}$ we use $y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}}$ and not only $\boldsymbol{h}_c$ as in the generative model.

5.2 Objective Function

In the following we derive the objective function of our Bayesian deep learning model following the variational principle. To maximize the log-likelihood $\ln(p(y|\boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}}))$, we derive an Evidence Lower Bound (ELBO) objective function, which ensures a correct approximation of the true posterior. To simplify the notation of the derivation of the objective function we make the following substitutions: $p_\theta(y) = p_\theta(y|\boldsymbol{z}, \boldsymbol{h}_{\mathfrak{D}})$, $p_\theta(\boldsymbol{z}) = p_\theta(\boldsymbol{z}|\boldsymbol{h}_c)$, $q_\phi(\boldsymbol{z}) =$

Algorithm 1: Optimization of the proposed model.

input : Claim hidden state $\boldsymbol{h}_c$, replies hidden state $\boldsymbol{h}_{\mathfrak{D}}$, veracity label y, the number claims N and the learning rate η.

begin
  $\theta, \phi \leftarrow$ Initialize parameters;
  repeat
    Randomly draw a minibatch of B claims;
    for *i=1,...,B* **do**
      $\tilde{\boldsymbol{z}} \leftarrow$ randomly draw S samples from $\mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_\phi, \text{diag}(\boldsymbol{\sigma}^2_\phi))$;
      $\mathcal{L}(\theta, \phi|y_i, \boldsymbol{h}_{c_i}, \boldsymbol{h}_{\mathfrak{D}_i}) \leftarrow$ compute the loss for one claim according to Eq. 14;
    $\mathcal{L}(\theta, \phi|y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}}) \leftarrow$ compute the loss for the full dataset according to Eq. 15;
    $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathcal{L}(\theta, \phi|y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}})$;
    $\phi \leftarrow \phi - \eta \cdot \nabla_\phi \mathcal{L}(\theta, \phi|y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}})$;
  until θ, ϕ *converge*;
  return θ, ϕ;

$q_\phi(\boldsymbol{z}|y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}})$. The objective function is derived as follows:

$$\begin{aligned}\ln(p(y|\boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}})) &= \ln\left(\int p_\theta(y) p_\theta(\boldsymbol{z}) d\boldsymbol{z}\right) \\ &\geq \int q_\phi(\boldsymbol{z}) \ln\left(\frac{p_\theta(\boldsymbol{z})}{q_\phi(\boldsymbol{z})} p_\theta(y)\right) d\boldsymbol{z} \\ &= \mathbb{E}_{q_\phi(\boldsymbol{z})}[\ln(p_\theta(y))] - D_{KL}[q_\phi(\boldsymbol{z}))\|p_\theta(\boldsymbol{z})] \\ &= \mathcal{L}(\theta, \phi|y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}}), \end{aligned} \tag{13}$$

where $\mathcal{L}(\theta, \phi|y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}})$ is the ELBO objective function. The second line of the derivation is possible by using the Jensen's inequality [16]. Since the ELBO objective function is a lower bound of the log-likelihood $\ln(p(y|\boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}}))$, its maximization maximizes the log-likelihood.

5.3 Gradient Estimation

Large-scale inference needs minibatch optimization. Thus, we derive a minibatch-based SGVB estimator to differentiate and optimize the ELBO objective function ($\mathcal{L}(\theta, \phi|y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}})$) with respect to both the inference parameters (ϕ) and the generative parameters (θ).

Through Monte Carlo estimation we compute the expectation part of the ELBO objective function. Let the minibatch size to be B and, for each claim $\boldsymbol{c}_i$ with $i \in [1, B]$, S a sample drawn from the variational posterior distribution $\tilde{\boldsymbol{z}} \sim q_\phi$. Given a subset of claims, we can construct an estimator of ELBO objective function for the full dataset based on mini-batches as follows:

$$\begin{aligned}\mathcal{L}(\theta, \phi|y_i, \boldsymbol{h}_{c_i}, \boldsymbol{h}_{\mathfrak{D}_i}) = &\frac{1}{S}\sum_{s=1}^{S}\left[\log p_\theta(y_i|\tilde{\boldsymbol{z}}^{(s)}, \boldsymbol{h}_{\mathfrak{D}_i})\right] \\ &- D_{KL}[q_\phi(\boldsymbol{z}))\|p_\theta(\boldsymbol{z})], \end{aligned} \tag{14}$$

$$\mathcal{L}(\theta, \phi|y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}}) \approx \frac{N}{B}\sum_{i=1}^{B}\mathcal{L}(\theta, \phi|y_i, \boldsymbol{h}_{c_i}, \boldsymbol{h}_{\mathfrak{D}_i}), \tag{15}$$

where $\mathcal{L}(\theta, \phi|y_i, \boldsymbol{h}_{c_i}, \boldsymbol{h}_{\mathfrak{D}_i})$ denote the estimates based on the i claim and N is the total number of claims. Algorithm 1 shows the minibatch gradient descent optimization process for both the generative (θ) and inference (ϕ) parameters. Note that the gradient steps in Algorithm 1 can easily be alternated with a more powerful optimizer such as the Adam algorithm [23].

Although both $q_\phi(\boldsymbol{z}|y, \boldsymbol{h}_c, \boldsymbol{h}_{\mathfrak{D}})$ and $p_\theta(\boldsymbol{z}|\boldsymbol{h}_c)$ are modeled as parameterized Gaussian distributions. The former is an approximation of the latter that only functions during learning. The latter, instead, is the learned distribution from which samples are generated in order to classify claim veracity.

5.4 Prediction

After training, we compute the posterior distribution $p_\theta(\boldsymbol{z}|\boldsymbol{h}_c)$ through the generative network. The actual prediction of a claim veracity is given by taking the expectation of S samples:

$$\overline{y} = \frac{1}{S}\sum_{s=1}^{S}\text{MLP}(\boldsymbol{z}^s, \boldsymbol{h}_{\mathfrak{D}}), \tag{16}$$

where $\boldsymbol{z}^s$ denote the samples drawn from the true posterior distribution $p_\theta(\boldsymbol{z}|\boldsymbol{h}_c)$.

6 EXPERIMENT SETUP

In this section we start by introducing 4 research questions. We then present the methodology used to answer them. The software used to run the experiments in this paper is available on the website of the first author.

6.1 Research Questions

We seek to answer the following four research questions, which will be guide the remainder of the paper:

RQ1 Does our model outperform the state-of-the-art misinformation detection baselines?
RQ2 Does the incorporation of the latent distribution outperforms a deterministic counterpart?
RQ3 Does the auxiliary information from people's replies produce a more accurate *posterior belief* of claim veracity?
RQ4 Is the temporal order better than random when encoding replies?
RQ5 Is it beneficial to incorporate a latent variable to encode replies?
RQ6 How does the dimension of the latent variable $\boldsymbol{z}$ affect the model's performance?

6.2 Datasets

In order to compare the performance of our proposed model against the baselines, we experimented with two real-world benchmark datasets, the *RumourEval* [9] and the *Pheme* [48] datasets. Both datasets contain Twitter conversation threads about news (like the example shown in Figure 1). A conversation thread consists of a tweet making a true or false claim, and branches of people's replies expressing their opinion about it. A summary of the datasets statistics is available in Table 1.

The RumourEval dataset has been developed for the SemEval-2017 Task 8 competition. This dataset consists of 325 source tweets

Table 1: Statistics of the datasets.

Subset	Veracity	RumourEval		Pheme	
		#Claims	#Replies	#Claims	#Replies
Training	True	83	1,949	861	24,438
	False	70	1,504	625	17,676
	Total	153	3,453	1,468	42,114
Validation	True	10	101	95	1,154
	False	12	141	115	1,611
	Total	22	242	210	2,765
Testing	True	9	412	198	3077
	False	12	437	219	3,265
	Total	21	849	417	6,342

and 5,568 user reply tweets. The veracity of each tweet can be true (45%), false (23%) or unverified (32%). Since we aim to only distinguish true and false claims, we filter out the unverified tweets. We divide the filtered dataset into a training subset, a validation subset and a testing subset. The training subset contains 153 claims with 3,453 replies, the validation subset contains 22 claims with to 242 replies, and the test subset contains 21 claims with 849 replies.

The *Pheme* dataset is constructed to help understand how users treat online rumour before and after the news is detected to be true or false. Like the *RumourEval* dataset, we divide the *Pheme* dataset into a training subset, a validation subset and a testing subset. Specifically, 70% of the claims are randomly selected as training instances, 10% as validation instances and the rest as testing instances. Users' replies are divided according to the claims.

6.3 Evaluation Measures

The misinformation detection task is a binary classification task. Such tasks are commonly evaluated by the following evaluation measures: Accuracy, F_1, Precision, and Recall.

Accuracy is a common evaluation measure for classification tasks. However, it is less reliable when datasets suffer from class imbalance. The evaluation measures Precision, Recall and F_1 complement Accuracy because not suffering from this problem.

6.4 Hyperparameters Setting

The activation function of the three LSTMs is tanh. The activation function of the MLPs is *ReLu*.

The hyperparameters tuned on the validation subset are:

- the dimension of the hidden layer of all three LSTMs is 30;
- the dimension of the latent variables is 10;
- the minibatch size is 32;
- the number of samples used in Monte Carlo estimates is 20.

State-of-the-art techniques have been employed to optimize the objective function: Dropout [38] is applied to improve neural networks training, L2-norm regularization is imposed on the weights of the neural networks, Adam optimizer [23] is exploited for fast convergence, and stepwise exponential learning rate decay is adopted to anneal the variations of convergence.

6.5 Baselines

We test our Bayesian deep learning model against six state-of-the-art models. In order to have a fair comparison, only those models using the claim content and users' replies have been selected.

Support Vector Machine (SVM). This model evaluates the performance of manually extracted features. The extracted features from claim content include: bag-of-words representation, presence of URLs, presence of hashtags, proportion of supporting and denying response [9]. These features are then input to a linear Support Vector Machine classifier. This classifier achieves the highest misinformation detection performance in the SemEval-2017 Task 8[4];

Convolutional Neural Networks (CNN). This model evaluates the performance of CNNs on the veracity detection task. Apart from the sequential approach such as BiLSTM, the convolutional model is another powerful neural architecture for natural language understanding [7, 8, 10, 22, 45]. CNN takes as input pre-trained word embeddings generated with Word2Vec [30] trained on the Google News dataset. To capture features similar to n-grams, we apply different convolutional window sizes. A max poling layer is applied to compress the output information of the convolutional layers [7];

Tensor Embeddings (TE). This model leverages tensor decomposition to derive concise claim embeddings, which are used to create a claim-by-claim graph on which labels propagate [14];

Evidence-Aware Deep Learning (DeClarE). This model retrieves evidences from replies using claims as a queries [31]. Then both claims and retrieved replies are input into a deep neural network with attention mechanism. Claim veracity is then computed by aggregating over the prediction generated by every claim-retrieved reply pair;

Multitask Learning (Multitask). This model leverages the relationship between two tasks of the veracity detection pipeline [25], stance detection and veracity prediction tasks. The model is trained on both jointly. We apply the hard parameter sharing mechanism, where different tasks share the same hidden LSTM layers. Task-specific layers takes the shared hidden information and generate per-task predictions;

Tree-structured RNN (TRNN). This model learns discriminative features from replies content by following their non-sequential propagation structure. Among the proposed two structures, we select the top-down structure for tweet representation learning because marginally better than the bottom-up structure [29].

7 RESULTS AND DISCUSSION

This section answers the research questions proposed in § 6.

7.1 Performance Comparison (RQ1)

Table 2 summarizes the classification performance of the baselines and our Bayesian deep learning framework on the RumourEval and Pheme datasets. We can observe that: (1) In terms of *Accuracy*

[4] http://alt.qcri.org/semeval2017/task8/

Table 2: Performance comparison of the proposed Bayesian deep learning framework against the baselines.

Dataset	Measure	SVM	CNN	TE	DeClarE	Multitask	TRNN	Ours
RumourEval	Accuracy (%)	71.42	61.90	66.67	66.67	66.67	76.19	**80.95**
	Precision (%)	66.67	54.54	60.00	58.33	57.14	70.00	**77.78**
	Recall (%)	66.67	66.67	66.67	77.78	**88.89**	77.78	77.78
	F_1 (%)	66.67	59.88	63.15	66.67	69.57	73.68	**77.78**
Pheme	Accuracy (%)	72.18	59.23	65.22	67.87	74.94	78.65	**80.33**
	Precision (%)	**78.80**	56.14	63.05	64.68	68.77	77.11	78.29
	Recall (%)	75.75	64.64	64.64	71.21	**87.87**	78.28	79.29
	F_1 (%)	72.10	60.09	63.83	67.89	77.15	77.69	**78.78**

and F_1, most deep learning-based models, such as ours, TRNN and Multitask, outperform the feature engineering-based models, i.e., SVM. This demonstrates that deep neural networks indeed help to learn better hidden representation of claims and replies. (2) Methods exploring relationships between a claim and its replies, such as ours, TRNN and Multitask, achieve better performance than claim content-based methods like TE and CNN. This demonstrates the significance of utilizing people's replies in the misinformation detection task. (3) Our model achieves state-of-the-art performance on both measures and both datasets demonstrating the effectiveness of our model in the misinformation detection task. This is true despite, our system is not the best for precision and recall alone. This because precision and recall alone do not offer a clear picture about the performance of a model since one measure can be increased at the expense of the other and *vice versa*.

Specifically, our model achieves the highest accuracy (80.95%), precision (77.78%) and F_1 (77.78%) on the RumourEval test subset, and the highest Accuracy (80.33%) and F_1 (78.78%) on the Pheme test subset. The TRNN is the strongest baseline achieving the second highest accuracy and F_1 on both RumourEval (76.19% and 73.68%) and Pheme (78.65% and 77.69%) test subsets.

7.2 Ablation of the Latent Distribution (RQ2)

In this subsection, we evaluate the impact of using a latent distribution into the claim encoder on the misinformation detection task. To evaluate the impact of the latent distribution p, we ablate p in our model and compare its classification performance against the full model. Specifically, the ablation is done by taking the output of the BiLSTM hidden states, i.e., $\boldsymbol{h}_c$ and give this as input to the output MLP. The rest of the model remains unchanged. Since no latent distribution is involved, the ablated model is optimized in accordance with the conventional Softmax loss minimization.

In Figure 4(a) and 4(b) we show the classification performance of the ablated model against the full model on the *RumourEval* and *Pheme* test subsets. We observe that the full model outperforms the ablated one by at least 7.77% on every evaluation measure. This demonstrates the better representation quality achieved by the use of the latent distribution.

7.3 Ablation of People's Replies (RQ3)

We now evaluate the contribution people's replies in the misinformation detection task. In order to examine its contribution we compare our full model with and without replies. Specifically, we ablate the input coming from the replies to the final MLP, which now is used only to perform a non-linear transformation of the latent variable $\boldsymbol{z}$.

In Figure 5(a) and 5(b) we show the classification performance of the ablated model against the full model on the RumourEval and the Pheme test subsets. Here, we observe that the auxiliary information extracted from people's replies has a large impact to the final performance our model. In fact, every evaluation measure is increased by at least 10.11%.

7.4 Random vs. Temporal Ordered Replies (RQ4)

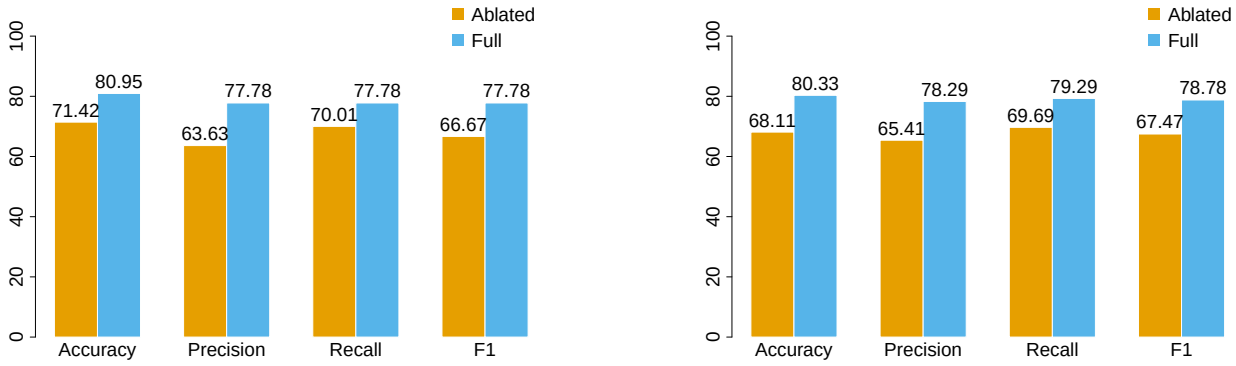
The proposed model rank people's replies based on the temporal order. In this subsection, we analyze the contribution of ranking the replies according to their temporal order. We compare this against a random order.Specifically, we randomize the $\boldsymbol{h}_d^M$ before it is input to the LSTM.

In Figure 6(a) and 6(b) we show the performance comparison of these two orders. We observe that the temporal ordered replies achieve better performance than the random ordered. Besides, the random ordered model is still worse TRNN yet better than Multitask. This is probably because TRNN takes the temporal structure of replies into the model while Multitask fail to involve temporal information.

7.5 A Latent Distribution for Replies (RQ5)

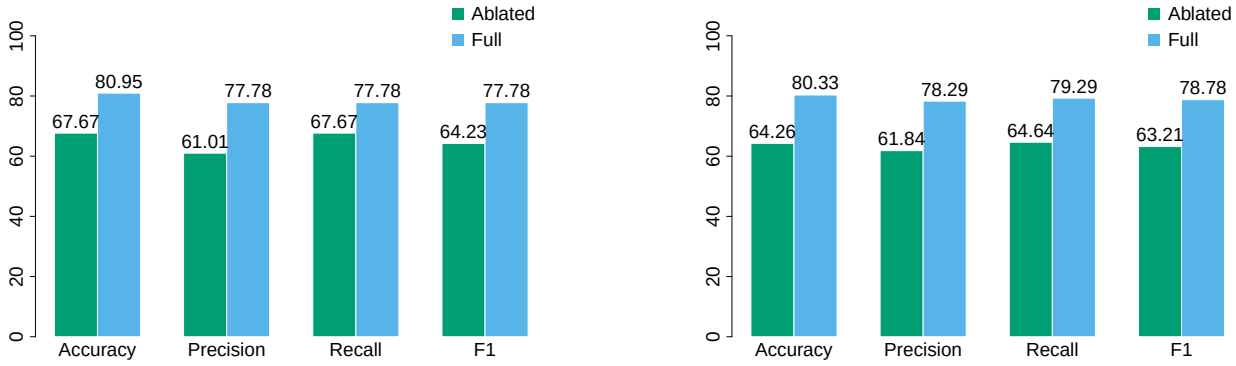
Considering the improved performance brought by the latent distribution for claims, in this subsection we answer whether it would be beneficial to incorporate a latent distribution also for replies. In order to answer this research question, we expand our model by adding a new latent distribution in the reply encoder. Similarly to what done for the claim encoder, the new latent distribution is designed as a multidimensional Gaussian distribution with mean and covariance matrix derived from the LSTM output $\boldsymbol{h}_{\mathcal{D}}$ (as in Eq. 3, 4 and 5). A new latent variable is sampled similarly as in Eq 6 and input to the MLP to predicting veracity of the claim being examined.

In Figure 8(a) and 8(b) we show the model performance comparison. We observe that the new latent distribution does not have an effect on the performance on the model for all the evaluation measures and dataset test subsets. Based on this analysis, we conclude that the incorporation of the additional latent distribution for replies does not provide any additional improvement in performance.



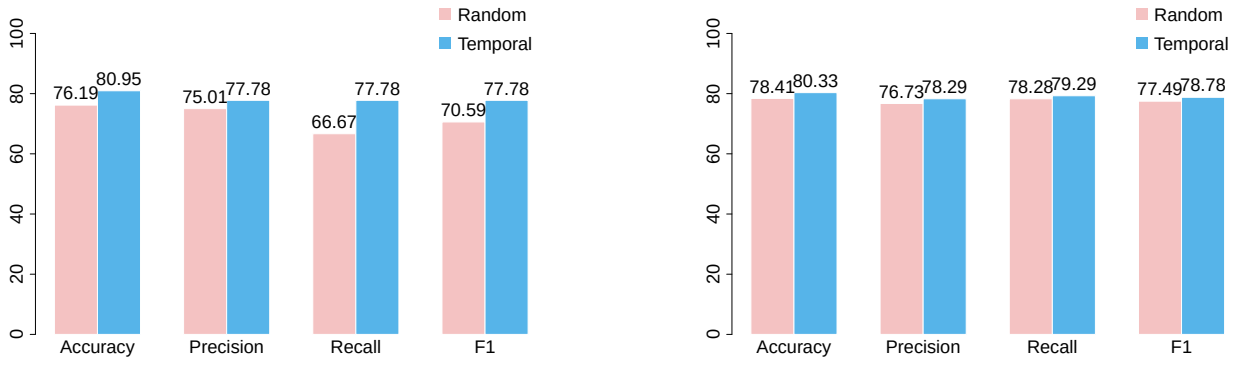
(a) RumourEval (b) Pheme

Figure 4: The impact of the latent distribution p on the model performance. In both figures we show the performance change on all the evaluation measures of the model with (Full) or without (Ablated) p. Figure (a) shows it for the RumourEval test subset and Figure (b) shows it for the Pheme test subset.



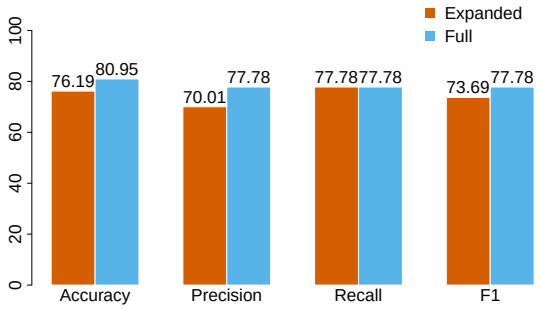
(a) RumourEval (b) Pheme

Figure 5: The impact of people's replies on the model performance. In both figures we show the performance change on all the evaluation measures of the model with (Full) or without (Ablated) replies information. Figure (a) shows it for the RumourEval test subset Figure (b) shows it for the Pheme test subset.

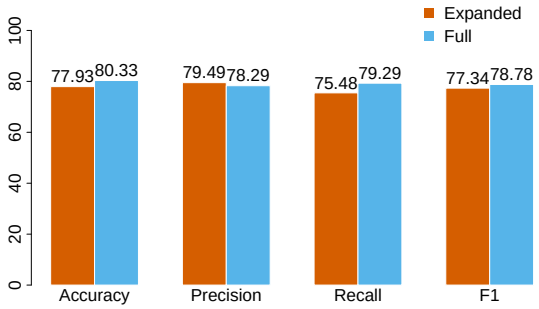


(a) RumourEval (b) Pheme

Figure 6: The effect of the temporal order of the reply encoder on model performance. In both figures we show the performance change on all the evaluation measures of the model with random and temporal ordered people's replies. Figure (a) shows it for the RumourEval test subset and Figure (b) shows it for the Pheme test subset.

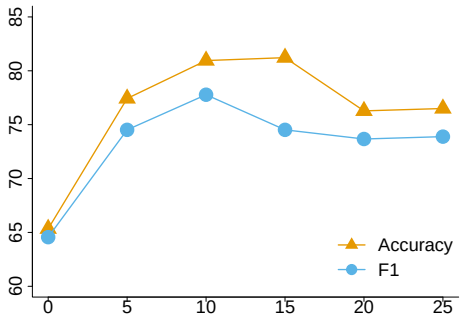


(a) RumourEval

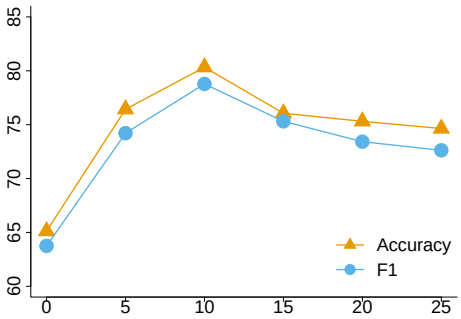


(b) Pheme

Figure 7: The effect of an additional latent distribution for people's replies on the model performance. In both figures we show the performance change on all the evaluation measures of the model with (Expanded) and without (Full) an additional latent variable for the people's replies. Figure (a) shows it for the RumourEval test subset and Figure (b) shows it for the Pheme test subset.



(a) RumourEval



(b) Pheme

Figure 8: The effect of the latent variable dimension on model performance. In both figures we show how the accuracy and F_1 scores change when varying the dimension of z. Figure (a) shows it for the RumourEval test subset and Figure (b) shows it for the Pheme test subset.

7.6 Sensitivity Analysis (RQ6)

In this subsection we evaluate the effect of the dimension of the latent variable z. To do this after setting a dimension for z we optimize the rest of the hyperparameters on the validation subset.

In Figure 8(a) and 8(b) we show the effect on performance of the dimension of z on both datasets. We observe that the results are similar for both evaluation measures, accuracy and F_1. Varying the dimension from 1 to 5 the model brings a larger performance improvement than when varying it from 5 to 25. When the dimension is 15 the model obtains the highest accuracy, 81.22%, on the RumourEval test subset while when the dimension is 10 the model obtains the highest F_1, 78.78%, on the RumourEval test subset and highest accuracy, 80.33% and F_1, 78.78%, on the Pheme test subset. These results also show that the increase in model capacity may not necessarily lead to an improvement in performance. The reason could be found on the limited size of the datasets, which might cause overfitting when the model is too complex.

8 CONCLUSIONS

In this paper, we study the problem of misinformation detection on social media platforms. One major problem faced by existing machine learning methods is the inability to represent uncertainty due to incomplete or finite available information. We address the problem by proposing a Bayesian deep learning model. When encoding claim content, we incorporate a latent distribution accounting for uncertainty and randomness caused by noisy patterns in the finite dataset. This latent distribution provides a prior belief of claim veracity. We also encode auxiliary information from people's replies in a temporal order through an LSTM. Such auxiliary information is then used to update the prior belief generating a posterior belief. In order to optimize the Bayes model, we derive a minibatch-based gradient estimation algorithm. Systematic experimentation has demonstrated the superiority of our approach against the state-of-the-art approaches in the misinformation detection task.

Despite encouraging experimental results, online misinformation detection is still a challenging problem with many open questions. In this paper, auxiliary information comes from people's replies alone, we argue that the proposed model can be enriched by utilizing other auxiliary information, such as source credibility. Also, the reply stances are a strong veracity indicator for a claim, since false claims are usually controversial and accompanied by opposite stances. We let for future work, the combination of features extract from credibility analysis and reply stances.

ACKNOWLEDGMENTS

This project was funded by the EPSRC Fellowship titled "Task Based Information Retrieval", grant reference number EP/P024289/1. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] Sadia Afroz, Michael Brennan, and Rachel Greenstadt. 2012. Detecting Hoaxes, Frauds, and Deception in Writing Style Online. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP '12)*. IEEE Computer Society, Washington, DC, USA, 461–475. https://doi.org/10.1109/SP.2012.34

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[3] Dimitrios Bountouridis, Mónica Marrero, Nava Tintarev, and Claudia Hauff. 2018. Explaining Credibility in News Articles using Cross-Referencing. In *SIGIR workshop on ExplainAble Recommendation and Search (EARS)*.

[4] Carlos Castillo, Mohammed El-Haddad, Jürgen Pfeffer, and Matt Stempeck. 2014. Characterizing the life cycle of online news stories using social media reactions. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 211–223.

[5] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*. ACM, 675–684.

[6] Yimin Chen, Niall J Conroy, and Victoria L Rubin. 2015. Misleading online content: Recognizing clickbait as false news. In *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*. ACM, 15–19.

[7] Yi-Chin Chen, Zhao-Yang Liu, and Hung-Yu Kao. 2017. IKM at SemEval-2017 Task 8: Convolutional neural networks for stance detection and rumor verification. In *Proceedings of the 11th International Workshop on Semantic Evaluation*. 465–469.

[8] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language Modeling with Gated Convolutional Networks. In *Proceedings of the 34th International Conference on Machine Learning*. 933–941.

[9] Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, 69–76. https://doi.org/10.18653/v1/S17-2006

[10] Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 69–78.

[11] Johannes Fürnkranz. 1998. A study using n-gram features for text categorization. *Austrian Research Institute for Artifical Intelligence* 3, 1998 (1998), 1–10.

[12] Maria Glenski, Tim Weninger, and Svitlana Volkova. 2018. Identifying and Understanding User Reactions to Deceptive and Trusted Social News Sources. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 176–181.

[13] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional LSTM networks for improved phoneme classification and recognition. In *International Conference on Artificial Neural Networks*. Springer, 799–804.

[14] Gisel Bastidas Guacho, Sara Abdali, Neil Shah, and Evangelos E. Papalexakis. 2018. Semi-supervised Content-Based Detection of Misinformation via Tensor Embeddings. In *IEEE/ACM 2018 International Conference on Advances in Social Networks Analysis and Mining*. 322–325. https://doi.org/10.1109/ASONAM.2018.8508241

[15] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. 2013. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 729–736.

[16] Frank Hansen and Gert K Pedersen. 2003. Jensen's operator inequality. *Bulletin of the London Mathematical Society* 35, 4 (2003), 553–564.

[17] Del Harvey and Yoel Roth. 2018. An Update On Our Elections Integrity Work. https://blog.twitter.com/official/en{_}us/topics/company/2018/an-update-on-our-elections-integrity-work.html

[18] Carl I Hovland and Walter Weiss. 1951. The influence of source credibility on communication effectiveness. *Public opinion quarterly* 15, 4 (1951), 635–650.

[19] Zhiwei Jin, Juan Cao, Yongdong Zhang, and Jiebo Luo. 2016. News verification by exploiting conflicting social viewpoints in microblogs. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[20] Zhiwei Jin, Juan Cao, Yongdong Zhang, Jianshe Zhou, and Qi Tian. 2017. Novel visual and statistical image features for microblogs news verification. *IEEE transactions on multimedia* 19, 3 (2017), 598–608.

[21] Michal Kakol, Radoslaw Nielek, and Adam Wierzbicki. 2017. Understanding and predicting Web content credibility using the Content Credibility Corpus. *Information Processing & Management* 53, 5 (2017), 1043–1061.

[22] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. 1746–1751.

[23] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).

[24] Diederik P. Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. *CoRR* abs/1312.6114 (2013). arXiv:1312.6114

[25] Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. All-in-one: Multi-task Learning for Rumour Verification. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 3402–3413. http://aclweb.org/anthology/C18-1288

[26] Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, et al. 2013. Prominent features of rumor propagation in online social media. In *International Conference on Data Mining*. IEEE.

[27] Tessa Lyons. 2018. Increasing Our Efforts to Fight False News | Facebook Newsroom. https://newsroom.fb.com/news/2018/06/increasing-our-efforts-to-fight-false-news/

[28] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 2015. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 1751–1754.

[29] Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor Detection on Twitter with Tree-structured Recursive Neural Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 1980–1989.

[30] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[31] Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 22–32.

[32] Chanthika Pornpitakpan. 2004. The persuasiveness of source credibility: A critical review of five decades' evidence. *Journal of applied social psychology* 34, 2 (2004), 243–281.

[33] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A Stylometric Inquiry into Hyperpartisan and Fake News. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 231–240.

[34] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2931–2937.

[35] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Eric P. Xing and Tony Jebara (Eds.), Vol. 32. PMLR, Bejing, China, 1278–1286.

[36] Victoria Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. 2016. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*. 7–17.

[37] Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 797–806.

[38] Nitish Srivastava. 2013. Improving neural networks with dropout. *University of Toronto* 182 (2013), 566.

[39] Robert Thomson, Naoya Ito, Hinako Suda, Fangyu Lin, Yafei Liu, Ryo Hayasaka, Ryuzo Isochi, and Zian Wang. 2012. Trusting tweets: The Fukushima disaster and information source credibility on Twitter. In *Proceedings of the 9th International ISCRAM Conference*. Vancouver: Simon Fraser University, 1–10.

[40] Shawn Tseng and BJ Fogg. 1999. Credibility and computing technology. *Commun. ACM* 42, 5 (1999), 39–44.

[41] Chong Wang and David M. Blei. 2013. Variational Inference in Nonconjugate Models. *J. Mach. Learn. Res.* 14, 1 (April 2013), 1005–1031.

[42] William Yang Wang. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 422–426. https://doi.org/10.18653/v1/P17-2067

[43] Aleksander Wawer, Radoslaw Nielek, and Adam Wierzbicki. 2014. Predicting webpage credibility using linguistic features. In *Proceedings of the 23rd international conference on world wide web*. ACM, 1135–1140.

[44] Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. 2012. Automatic detection of rumor on Sina Weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*. ACM, 13.

[45] Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S Yu. 2018. TI-CNN: Convolutional Neural Networks for Fake News Detection. *arXiv preprint arXiv:1806.00749* (2018).

[46] Qiang Zhang, Shangsong Liang, Aldo Lipani, Zhaochun Ren, and Emine Yilmaz. 2019. From Stances' Imbalance to Their Hierarchical Representation and Detection. In *Companion Proceedings of the The Web Conference 2019*. ACM Press.

[47] Qiang Zhang, Emine Yilmaz, and Shangsong Liang. 2018. Ranking-based Method for News Stance Detection. In *Companion Proceedings of the The Web Conference 2018*. ACM Press.

[48] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one* 11, 3 (2016), e0150989.