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

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: Snopes¹, Polifact², and Emergent³, 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.

¹https://www.snopes.com/
²https://www.polifact.com/
³http://www.emergent.info/
Police have clarified that there were two shootings in Ottawa today, not three: at the War Memorial and Parliament Hill.

11:26 AM - 22 Oct 2014

231 Retweets 72 Likes

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 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 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, 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-recursive 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 $C = \{c_1, c_2, \ldots, c_N\}$ be a set of $N$ claims. The claim $c_i$ is commented by a set of $M$ user replies $D_i = \{d_{i,1}, d_{i,2}, \ldots, d_{i,M}\}$. We use $y_i$ to denote the binary veracity label of the claim $c_i$, 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., $\{c_i, D_i\}$, 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: $(c, D, y)$.

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 $c$. 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 $c$ be a sequence of discrete words or tokens, i.e., $c = [w_1, w_2, \ldots, w_J]$, where $w_j \in \mathbb{R}^d$ 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
4.1 Claim Encoder

The claim encoder takes as input $c$, converts the sequence of word embeddings into a dense representation, and outputs the concatenation of two hidden states capturing past and future information:

$$h_c = \text{BiLSTM}(c),$$

where $h_c$ 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 $z$. To embed linguistic information into the latent variable, we set the latent variable to be conditional on $h_c$:

$$z \sim p_\theta(z|h_c).$$

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

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

$$p_\theta(z|h_c) = \mathcal{N}(z|\mu_\theta, \text{diag}(\sigma_\theta^2)),$$

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

$$\pi_\theta = f_\theta(h_c),$$

$$\mu_\theta = l_1(\pi_\theta), \quad \ln(\sigma_\theta) = l_2(\pi_\theta),$$

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

In order to make $\mu_\theta$ and $\sigma_\theta$ differentiable and backpropagate the loss through the latent distribution ($p$), the following reparameterization trick is used:

$$z = \epsilon \cdot \sigma_\theta + \mu_\theta \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, I),$$

where $\epsilon$ is a vector of zeros and $I$ is the identity matrix. By making use of the latent variable ($z$), 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 an LSTM to capture the temporal semantic variation of the replies.

Given a claim $c$ commented by a list of replies $\mathcal{D} = \{d_1, \ldots, d_m, \ldots, d_M\}$, these replies are ranked based on their temporal order. The content of a $d_m$ consists of a sequence of words $d = \{w_1, \ldots, w_k\}$. 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, $w_k \in \mathbb{R}^d$ 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 $d_m \in \mathcal{D}$ for example, the concatenation of hidden states from forward and backward directions is denoted as:

$$h_{d_m} = \text{BiLSTM}(d_m),$$

(7)

where $h_{d_m}$ is the summary of the reply $d_m$.

In order to capture the semantic information of all replies, we sequentially input the concatenated hidden states of each reply into an 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:

$$h_D = \text{LSTM}(h_{d_M}^{TM}),$$

(8)

where $h_{d_M}^{TM} = \{h_{d_1}, \ldots, h_{d_M}, \ldots, h_{d_M}\}$.

The last hidden state $h_D$ contains both the linguistic information of each reply and the semantic changes between replies along the time line. This $h_D$ 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 $z$ concatenated to the hidden state of replies $h_D$:

$$y = \text{MLP}(z, h_D).$$

(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 $\phi$ to approximate the intractable true posterior $p_\theta(z|h_c)$; then we derive an objective function to measure how well $p_\theta(z|h_c)$ is approximated; finally we exploit the Stochastic Gradient Variational Bayes (SGVB) method [24, 35] to learn the inference model parameters $\phi$ together with the generative model parameters $\theta$. Figure 3 shows the graphical representation of the generative model and the inference model.

5.1 Inference Model

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

$$q_\phi(z|y, h_c, h_D) = N(z|\mu_\phi, \text{diag}(\sigma^2_\phi)),$$

(10)

where $\mu_\phi$ and $\text{diag}(\sigma^2_\phi)$ 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:

$$\pi_\phi = f_\phi(y, h_c, h_D),$$

(11)

$$\mu_\phi = l_3(\pi_\phi), \quad \text{diag}(\sigma^2_\phi) = l_4(\pi_\phi),$$

(12)

where $f_\phi$ denotes a MLP, and $l_3$ and $l_4$ denote two LTs. Note that in the inference model to compute $\mu$ and $\log \sigma$ we use $y$, $h_c$, $h_D$ and not only $h_c$ 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 $\ln(p(y|h_c, h_D))$, 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:

$$p_\theta(y) = p_\theta(y|z, h_D), \quad p_\theta(z) = p_\theta(z|h_c), \quad q_\phi(z) = \cdots$$

Figure 3: The directed graphical model. Grey nodes represent observed variables while blank nodes represent latent variables. Solid lines denote the generative model $p_\theta(z|h_c)p(y|z, h_D)$, dashed lines denote the variational approximation $q_\phi(z|y, h_c, h_D)$ to the intractable posterior $p_\theta(z|h_c)$. The variational parameters are learned together with the generative model parameters.
We can construct an estimator of ELBO objective function for the variational posterior distribution full dataset based on mini-batches as follows:

\[
\mathcal{L}(\theta, \phi|y_i, h_{c_i}, h_{T_i}) = \frac{1}{B} \sum_{i=1}^{B} \mathcal{L}(\theta, \phi|y_i, h_{c_i}, h_{T_i}),
\]

where \(\mathcal{L}(\theta, \phi|y_i, h_{c_i}, h_{T_i})\) 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 (\(\theta\)) and inference (\(\phi\)) parameters. Note that the gradient steps in Algorithm 1 can be easily alternated with a more powerful optimizer such as the Adam algorithm [23].

Although both \(q_\phi(z|y_i, h_{c_i}, h_{T_i})\) and \(p_\theta(z|h_{c_i})\) 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 \(p_\theta(z|h_{c})\) through the generative network. The actual prediction of a claim veracity is given by taking the expectation of \(S\) samples:

\[
\bar{y} = \frac{1}{S} \sum_{s=1}^{S} \text{MLP}(z^s, h_{T}),
\]

where \(z^s\) denote the samples drawn from the true posterior distribution \(p_\theta(z|h_{c})\).

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 \(z\) 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...
We divide the filtered 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. Users’ replies are divided according to the claims.

### 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      |

6.3 Evaluation Measures

The misinformation detection task is a binary classification task. Such tasks are commonly evaluated by the following evaluation measures: Accuracy, F1, 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 F1 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;

**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 pooling 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 reply 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

---

http://alt.qcri.org/semeval2017/task8/
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 is 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., $h_c$ 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 $z$.

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 of 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 $h_M^p$ 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 $h_M$ (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.
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.

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.

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.
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.
