VRoC: Variational Autoencoder-aided Multi-task Rumor Classifier Based on Text

Mingxi Cheng
University of Southern California
mingxic@usc.edu

Shahin Nazarian
University of Southern California
shahin.nazarian@usc.edu

Paul Bogdan
University of Southern California
pbogdan@usc.edu

ABSTRACT

Social media became popular and percolated almost all aspects of our daily lives. While online posting proves very convenient for individual users, it also fosters fast-spreading of various rumors. The rapid and wide percolation of rumors can cause persistent adverse or detrimental impacts. Therefore, researchers invest great efforts on reducing the negative impacts of rumors. Towards this end, the rumor classification system aims to to detect, track, and verify rumors in social media. Such systems typically include four components: (i) a rumor detector, (ii) a rumor tracker, (iii) a stance classifier, and (iv) a veracity classifier. In order to improve the state-of-the-art in rumor detection, tracking, and verification, we propose VRoC, a tweet-level variational autoencoder-based rumor classification system. VRoC consists of a co-train engine that trains variational autoencoders (VAEs) and rumor classification components. The co-train engine helps the VAEs to tune their latent representations to be classifier-friendly. We also show that VRoC is able to classify unseen rumors with high levels of accuracy. For the PHEME dataset, VRoC consistently outperforms several state-of-the-art techniques, on both observed and unobserved rumors, by up to 26.9%, in terms of macro-F1 scores.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; • **Information systems** → *Social networks*; • **Data mining**; • **Natural language processing**;

KEYWORDS

Variational Autoencoder, Rumor Detection, Rumor Tracking, Veracity Classification, Stance Classification, False Rumors, Fake News, Misinformation, Text Mining, LSTM.

ACM Reference Format:
Mingxi Cheng, Shahin Nazarian, and Paul Bogdan. 2020. VRoC: Variational Autoencoder-aided Multi-task Rumor Classifier Based on Text. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366423.3380054

1 INTRODUCTION

Social media and micro-blogging have gained popularity [10, 19] as tools for gathering and propagating information promptly. About two-thirds of Americans (68%) obtain news on social media [23], while enjoying its convenient and user-friendly interfaces for learning, teaching, shopping, etc. Journalists use social media as convenient yet powerful tools and ordinary citizens post and propagate information via social media easily [44]. Despite the success and popularity of online media, the suitability and rapidly-spreading nature of micro-blogs fosters the emergence of various rumors [1, 3]. Individuals encountering rumors on social media may turn to other sources to evaluate, expose, or reinforce rumors [6, 28]. The rapid and wide spread of rumors can cause various far-reaching consequences, for example, during the 2016 U.S. presidential election, 529 different rumors about Donald Trump and Hillary Clinton spread on social media [21] and reached millions of voters swiftly. Hence, these rumors could potentially influence the election [3]. More recently, the rapid spread of rumors about 2019 novel coronavirus [7, 34, 35] (some of which are verified to be very dangerous false claims [20], e.g., those that suggest drinking bleach cures the illness [40]) has made social media companies such as Facebook to find more effective solutions [43]. If not identified timely, sensational and scandalous rumors could provoke social panic during emergency events, e.g., coronavirus [22], threaten the internet credibility and trustworthiness [12], with serious implications [11].

Social media rumors are therefore a major concern. Commercial giants, government authorities, and academic researchers heavily invest in diminishing the adverse impacts of rumors [3]. The literature defines a rumor as "an item of circulating information whose veracity status is yet to be verified at the time of posting" [44]. On a related note, if the veracity status is confirmed to be false, the rumor can then be considered as fake news. Rumor handling research efforts cast four main elements: rumor detection, rumor tracking, rumor stance classification, and rumor veracity classification [44]. A typical rumor classification system includes all the four elements.

As shown in Fig. 1, the first step in rumor classification is rumor detection. Identifying rumors and non-rumors has been usually formulated into a binary classification problem. Among the numerous approaches, there are three major categories: hand-crafted features-based approaches, propagation-based approaches, and neural network approaches [3]. Traditional methods mostly utilize hand-crafted features extracted from textural and/or visual contents of rumors. Having applied these features to describe the distribution of rumors, classifiers are trained to detect rumors [4, 27]. The approaches based on the structure of social network use message propagation information and evaluate the credibility of the network [15], but ignore the textual features of rumors. Social bot detection and tracking built on social network structure and user information can be utilized to detect bot-generated rumors. Recent deep neural network (DNN)-based methods extract and learn features automatically and achieve significantly high accuracies on rumor detection [5]. Generative models and adversarial training techniques have also been used to improve the performance of rumor detectors

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380054

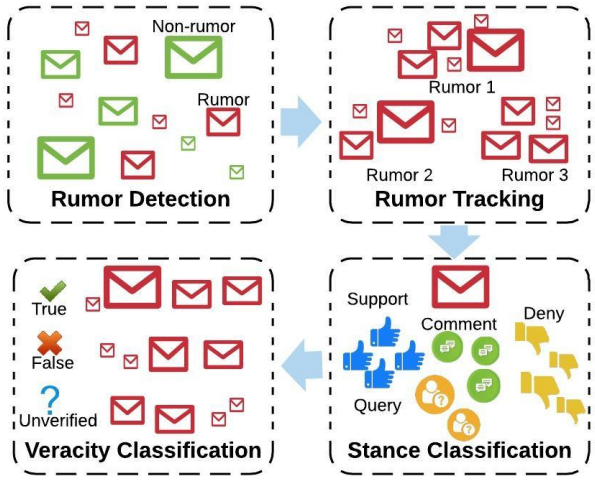

Figure 1: Rumor classification system consists of four components: rumor detection, rumor tracking, rumor stance classification, and rumor veracity classification.

[33]. After a rumor is identified, all the related posts or sentences discussing this rumor should be clustered together for later processing, and other unrelated posts should be filtered out. This rumor tracking task can be formulated into a binary classification problem, which classifies posts as related or unrelated to a rumor. Unlike other popular components in rumor classification system, research in rumor tracking has been scarce. The state-of-the-art work uses tweet latent vector to overcome the limited length of a tweet [16].

Once the rumor tracking component clusters posts related to a rumor, the stance classification component labels each individual post by its orientation toward rumor's veracity. For example, a post or reply can be labeled as support, deny, comment, or query [14]. Usually rumor stance classification can be realized as a two to four-way classification problem. Recurrent neural networks (RNNs) with long short-term memory (LSTM) [18] cells have been used to predict stance in social media conversations. The authors in [26] proposed to use convolution units in Tree LSTMs to realize a four-way rumor stance classification. Variational autoencoders (VAEs) [24] have been used to boost the performance of stance classification. The authors in [39] utilize LSTM-based VAEs to capture the hidden meanings of rumors containing both text and images.

The final component in rumor classification is veracity classification, which determines the truth value of a rumor, i.e., a rumor can be true, false, or unverified. Some works have limited the veracity classification to binary classification, i.e., a rumor can either be true or false [42]. The initiated research in this direction does not tackle the veracity of rumors directly, but rather their credibility perceptions [4]. Later works in this area dealing with veracity classification take advantage of temporal features, i.e., how rumors spread over time, and linguistic features. More recently, LSTM-based DNNs are frequently used to do veracity classification [25, 26]. Similarly, fake news detection is often formulated into a classification problem [37]. Without establishing a verified database, rumor veracity classification and fake news classification perform a similar task.

Instead of tackling each component in rumor classification system individually, multi-task classifiers are proposed to accomplish two or more functions. Tree LSTM models proposed in [26] employ stance and rumor detection and propagate the useful stance signal up in the tree for followup rumor detection. DNNs are trained jointly in [32] to unify stance classification, rumor detection, and veracity classification tasks. Rumor detection and veracity classification sometimes are executed together since they can be formulated into a four-way classification problem as in [16, 32]. A post can be labeled as a non-rumor, true rumor, false rumor, or unverified rumor. The authors of [25] proposed an LSTM-based multi-task learning approach that allows joint training of the veracity classification and auxiliary tasks, rumor detection and stance classification.

Previous works in scientific literature have accomplished one or a few tasks in rumor classification, but none of them provides a complete high performance rumor classification system to account for all four components. In this work, we propose VRoC to realize all four tasks. The contributions of this work are as follows:

- We propose VRoC, a tweet-level text-based novel rumor classification system based on variational autoencoders. VRoC realizes all four tasks in the rumor classification system and achieves high performance compared to state-of-the-art works.
- We propose a co-train engine to jointly train the VAEs and rumor classification components. This engine pressurizes the VAEs to tune their latent representations to be classifier-friendly. Therefore, higher accuracies are achieved compared to other VAE-based rumor detection approach.
- We show that the proposed VRoC has the ability to classify previously seen or unseen rumors. Due to the generative nature of VAEs, VRoC outperforms baselines under both training policies introduced in Section 3.

The rest of this paper is organized as follows. In Section 2, we introduce the proposed VRoC, including both VAE and the rumor classification system. In Section 3, we describe the dataset and baselines we used in our experiments. In Section 4, we show our experimental results. Section 5 concludes the paper.

2 VRoC FRAMEWORK

In this section we first present the problem statement and then describe the details of our VRoC framework. Fig. 2 illustrates our VRoC, a VAE-aided multi-task rumor classifier that consists of rumor detector, rumor tracker, stance classifier, and veracity classifier. VAE in this work is an LSTM-based variational autoencoder model to extract latent representations of tweet-level text. For each rumor classification component, a VAE is jointly trained to extract meaningful latent representations that not only is information rich, but also is friendly and suitable for each component.

2.1 Problem Statement

Rumor classification consists of four components: rumor detection (D), rumor tracking (T), stance (S) classification, and veracity (V) classification, each of which can be formulated into a classification problem. Given a tweet-level text x, VRoC can provide four outputs $\{y_D, y_T, y_S, y_V\}$, where $y_D \in \{Rumor, Nonrumor\}$, $y_T \in \{Related, Unrelated\}$, $y_S \in \{Support, Deny, Comment, Query\}$, $y_V \in \{True, False, Unverified\}$. Four components are realized independently in this work, but they can also be implemented as one general classifier that jointly produces four types of outputs, or two to three classifiers that each realizes one or more tasks.

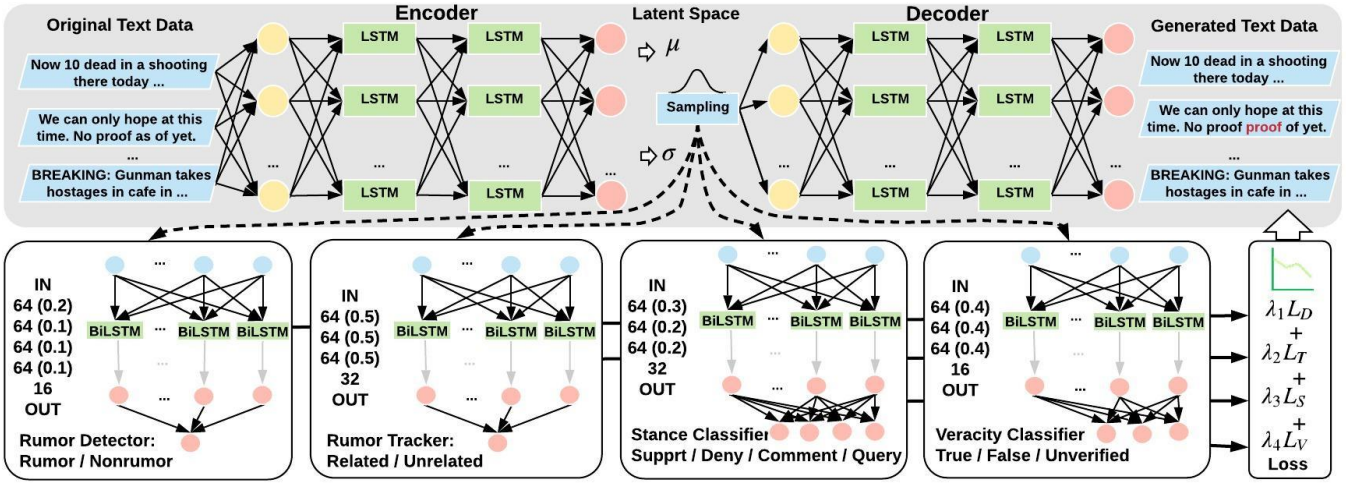

Figure 2: VRoC: The proposed VAE-aided multi-task rumor classification system. The top half illustrates the VAE structure, and the bottom half shows the four components in the rumor classification system. IN and OUT represent input layer and output layer, respectively. Numbers in parenthesis indicate the dropout rates. Note that the generated text could be different from the original text, if the VAE is not perfect.

2.2 LSTM-based Variational Autoencoder

The VAE model in this work consists of an encoder and a decoder, both of which are LSTM networks because of the sequential nature of language. To extract latent representation from tweet-level data, in this work, we take advantage of VAEs and utilize the encoder-extracted latent representations to compress and represent information in textural data. The decoder decodes the latent representations generated by the encoder into texts and ensures the latent representations are accurate and meaningful. Rumor classifier components are co-trained with VAEs to help the encoders tune their outputs to be more classifier-friendly.

Let us consider a set of tweets $X = \{x_1, x_2, ..., x_N\}$, each of which is generated by some random process $p_\theta(x|z)$ with an unobserved variable z. This z is generated by a prior distribution $p_\theta(z)$ that is hidden from us. In order to classify or utilize these tweets, we have to infer the marginal likelihood $p_\theta(x)$, however θ is unfortunately also unknown to us. In VAEs, z is drawn from a normal distribution, and we attempt to find a function $p_\theta(x|z)$ that can map z to x by optimizing θ, such that x looks like what we have in data X. From coding theory, z is a latent representation and we can use a recognition function $q_\phi(z|x)$ as the encoder [24]. $q_\phi(z|x)$ takes a value x and provides the distribution over z that is likely to produce x. $q_\phi(z|x)$ is also an approximation to the intractable true posterior $p_\theta(z|x)$. Kullback-Leibler (KL) divergence measures how close the $p_\theta(z|x)$ and $q_\phi(z|x)$ are:

$$D_{KL}[q_\phi(z|x)||p_\theta(z|x)] = E_{z\sim q_\phi(z|x)}[logq_\phi(z|x) - logp_\theta(z|x)]. \quad (1)$$

After applying Bayesian theorem, the above equation reads:

$$logp_\theta(x) = D_{KL}[q_\phi(z|x)||p_\theta(z|x)] + L(\theta, \phi; x), \quad (2)$$

$$L(\theta, \phi; x) = E_{z\sim q_\phi(z|x)}[-logq_\phi(z|x) + logp_\theta(x|z) + logp_\theta(z)]. \quad (3)$$

We can see the autoencoder from Eq. 3 already: q_ϕ encodes x into z and p_θ decodes z back to x. Since KL divergence is non-negative, $L(\theta, \phi; x)$ is called the evidence lower bound (ELBO) of $logp_\theta(x)$:

$$logp_\theta(x) \geq L(\theta, \phi; x). \quad (4)$$

In VAEs, $q_\phi(z|x)$ is a Gaussian: $N(z; \mu, \sigma^2 I)$, where μ and σ are outputs of the encoder. The reparameterization trick [24] is used to express $z \sim q_\phi(z|x)$ as a random variable $z = g_\phi(\epsilon, x) = \mu + \sigma \cdot \epsilon$, where the auxiliary variable ϵ is drawn from a standard normal distribution. A Monte Carlo estimation is then formed for ELBO as follows:

$$L^{MC}(\theta, \phi; x) = \frac{1}{N}\sum_{n=1}^{N} -logq_\phi(z_n|x) + logp_\theta(x|z_n) + logp_\theta(z_n), \quad (5)$$

where $z_n = g_\phi(\epsilon_n, x)$, $\epsilon_n \sim N(0, I)$.

2.2.1 Encoder. The encoder $q_\phi(z|x)$ is realized by an RNN with LSTM cells. The input to the encoder is a sequence of words $x = [w_1, w_2, ..., w_T]$, e.g., a tweet with length T. The hidden state h_t, $t \in [1, T]$ in LSTM is updated as follows:

$$h_t = o_t * tanh(C_t), \quad (6)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, w_t] + b_o), \quad (7)$$

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t, \quad (8)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, w_t] + b_f), \quad (9)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, w_t] + b_i), \quad (10)$$

$$\bar{C}_t = tanh(W_C[h_{t-1}, w_t] + b_C). \quad (11)$$

σ is a sigmoid function. o_t, f_t, i_t are output, forget, and input gate, respectively. C_t, $\bar{C}_t$ are new state and candidate cell state, respectively. W_o, W_f, W_i, W_C, b_o, b_f, b_i, b_C are parameter metrics. The outputs of the encoder are divided into μ and σ.

2.2.2 Decoder. The decoder $p_\phi(x|z)$ is realized by an RNN with LSTM cells which has a matching architecture as that of the encoder. It takes a z as input and outputs the probabilities of all words. The probabilities are then sampled and decoded into a sequence of words.

2.2.3 Training. In this work, we propose a *co-train engine* for VAEs. Each component in rumor classifier is trained jointly with a VAE, i.e., there are four sets of VAEs and components. Each set is trained individually. To co-train each set, we modify the loss function of VAE by adding a classification penalty, which is the loss of the rumor classification component. By backpropagating the loss of each component to the corresponding VAE, the VAE learns to tune its parameters to provide classifier-friendly latent representations. In addition, this operation introduces randomness into the training process of VAEs, hence the robustness and generalization ability of VRoC are improved. The loss function of VRoC then reads:

$$L_{VRoC} = L^{MC}(\theta, \phi; x) + \lambda_1 L_D + \lambda_2 L_T + \lambda_3 L_S + \lambda_4 L_V, \quad (12)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are balancing parameters for each rumor classification component. L_D, L_T, L_S, L_V are loss functions of rumor detector, rumor tracker, stance classifier, and veracity classifier, respectively. Training VAEs and the components jointly improves the performance of the rumor classifier, compared to the VAE-based rumor classifiers without our co-train engine. We confirm the improvement by conducting a set of comparison experiments between VRoC and VAE-LSTM in Section 4.

2.3 Rumor Classifier

In this section, we introduce all four components in VRoC's rumor classification system and their loss functions. The loss functions are consistent with free-energy minimization principle [13].

2.3.1 Rumor Detector. Given a set of social media posts $X = \{x_1, x_2, ..., x_N\}$, rumor detector determines which ones are rumors and which ones are not. The classified posts can then be tracked and verified in later steps. In this work, rumor detection task is formulated as a binary classification problem. Rumor detector D is realized by an RNN with Bidirectional-LSTM (BiLSTM) cells. It takes a z as input and provides a probability y_D of the corresponding post x being a rumor. The loss function L_D is defined as follows:

$$L_D = -E_{y \sim Y_D}[ylog(y_D) + (1-y)log(1-y_D)], \quad (13)$$

where $Y_D \in \{Rumor, Nonrumor\}$ is a set of ground truth labels.

2.3.2 Rumor Tracker. Rumor tracker T is activated once a rumor is identified. It takes a set of posts X as input, and determines whether each post is related or unrelated to the given rumor. Rumor tracking task is formulated into a classification problem in this work, and it is fulfilled by an RNN with BiLSTM cells. Given a z, T generates a probability y_T indicating whether the corresponding post is related to the identified rumor. The loss function L_T reads:

$$L_T = -E_{y \sim Y_T}[ylog(y_T) + (1-y)log(1-y_T)], \quad (14)$$

where $Y_T \in \{Related, Unrelated\}$.

2.3.3 Stance Classifier. Given a collection of rumors $R = \{r_1, r_2, ..., r_N\}$, where each rumor r_n ($n \in [1, N]$) consists of a set of posts discussing it, the stance classifier S determines whether each post is supporting, denying, commenting, or querying the related rumor r_n. In VRoC, we utilize an RNN with BiLSTM cells to perform a four-way rumor stance classification. The loss function L_S is defined as follows:

$$L_S = -E_{y \sim Y_S}[\sum ylog(y_S)], \quad (15)$$

where $Y_S \in \{Support, Deny, Comment, Query\}$.

2.3.4 Veracity Classifier. Once rumors are identified, their truth values are determined by rumor veracity classifier V. Instead of being true or false, some rumors could in reality remain unverified for a period of time. Hence, in this work, we provide a three-way veracity classification using an RNN with BiLSTM cells. The loss function L_V is defined as follows:

$$L_V = -E_{y \sim Y_V}[\sum ylog(y_V)], \quad (16)$$

where $Y_V \in \{True, False, Unverified\}$. This veracity classifier can also be used for fake news detection since it performs a similar task.

3 EXPERIMENTAL SETTINGS

In this section, we first describe the datasets and evaluation metrics, then introduce the baselines from state-of-the-art works.

3.1 Datasets and Evaluation Metrics

We evaluate our VRoC on PHEME dataset [25]. PHEME has two versions. PHEME5 contains 5792 tweets related to five news, 1972 of them are rumors and 3820 of them are non-rumors. PHEME9 is extended from PHEME5 and contains veracity labels. RumourEval dataset [14] is derived from PHEME, and its stance labels are used for rumor stance classification task. We use PHEME5 for rumor detection and tracking task, PHEME5 with stance labels from RumourEval for rumor stance classification task, and PHEME5 with veracity labels from PHEME9 for the rumor veracity classification task. Due to the imbalance of classes in dataset, accuracy alone as evaluation metric is not sufficient. Hence, we use precision, recall, and macro-F1 scores [30, 41] together with accuracy as evaluation metrics. Since baselines are trained under different principles, we carry out two types of trainings to guarantee fairness. To compare with baselines that are trained under the leave-one-out (L) principle, i.e., train on four news and test on another news, we train our models under the same principle. L principle evaluates the ability of generalization and constructs a test environment close to real world scenarios. To compare with baselines that do not use this principle, we hold out 10% of the data for model tuning.

3.2 Models

In **VRoC**, we co-train each set of VAE and classification component after pre-training the VAEs. To show the effectiveness of our designed co-train engine, we developed a baseline **VAE-LSTM** that trains VAE first, then uses its latent representations as input to train the LSTM models. VAE-LSTM's architecture is the same as VRoC, but without the proposed co-train engine. The VAEs used in VRoC and VAE-LSTM are pre-trained with the same procedure. The encoder architecture in VAE is EM-LSTM32-LSTM32-D32, where EM, LSTM32, D32 represent an embedding layer, an LSTM layer with 32 neurons, and a dense layer with 32 neurons, respectively. The decoder's architecture is IN-LSTM32-LSTM32-Dvs, where IN and Dvs represent input layer and dense layer with vocabulary size neurons. We chose our VAE architecture by an expert-guided random search (RS) under the consideration of data size. Compared to other computationally expensive neural architecture search methods, such as evolutionary search and reinforcement learning-based approaches, RS is confirmed to achieve competitive performance [9, 31, 38]. Early stopping strategy is used in training. Other state-of-the-art baselines used for comparison are described as follows.

3.2.1 Rumor Detection. **CRF** [46] is a content and social feature-based rumor detector that based on linear-chain conditional random fields learns the dynamics of information during breaking news. **GAN-GRU**, **GAN-BOW**, and **GAN-CNN** [33] are generative adversarial network-based rumor detectors. A generator is designed to create uncertainty and the complicated sequences force the discriminator to learn stronger rumor representations. **DataAUG** [17] is a contextual embedding model with data augmentation. It exploits the semantic relations between labeled and unlabeled data. **DMRF** [36] formulates rumor detection task as an inference problem in a Markov Random Field (MRF). It unfolds the mean-field algorithm into neural network and builds a deep MRF model. **DTSL** [8] is a deep semi-supervised learning model containing three CNNs.

3.2.2 Rumor Tracking. As we mentioned before, rumor tracking receives few attention in scientific literature. Hence, we train two baselines to compare with our proposed VRoC. **CNN** is a convolutional neural network-based baseline. **LSTM** is an RNN with LSTM cells. We input all five news in PHEME to all models, and perform a five-way classification, i.e., determine the input post is related to which one in five news. Five news are: Sydney siege (S), Germanwings (G), Ferguson (F), Charlie Hebdo (C), and Ottawa shooting (O). L principle is not applicable under this setup.

3.2.3 Stance Classification. **LinearCRF** and **TreeCRF** are two different models proposed in [45] for capturing the sequential structure of conversational threads. They analyse tweets by mining the context from conversational threads. The authors in [32] proposed a unified multi-task (rumor detection and stance classification) model based on multi-layer RNNs, where a shared layer and a task-specific layer are employed to accommodate different types of representations of the tasks and their corresponding parameters. **MT-US** and **MT-ES** are multi-task models with the uniform shared-layer and enhanced shared-layer architecture.

3.2.4 Veracity Classification. **MT-UA** [29] is a multi-task rumor detector that utilizes the user credibility information and attention mechanism. In [25], a joint multi-task deep learning model with hard parameters sharing is presented. It outperforms the sequential models by combining multiple tasks together. We choose two best performing models from [25]: **MTL2** that combines veracity and stance classification, and **MTL3** that combines detection, stance, and veracity classification. **TreeLSTM** [26] is a tree LSTM model that uses convolution and max-pooling unit.

4 EXPERIMENTAL RESULTS

The comparison between VRoC and baselines in all four rumor classification tasks are presented in Section 4.1 to 4.4. In all tables, * indicates the best results from the work that proposed the corresponding model. We finally describe the comparison results of VRoC and VAE-LSTM in Section 4.5.

4.1 Rumor Detection

Comparison results between VRoC and baselines on the rumor detection task are shown in Table 1. Compared to baselines, VRoC achieves significantly higher accuracy levels and macro-F1 scores, and VAE-LSTM stands as the second best. VRoC outperforms CRF and GAN-GRU by 26.9% and 9.5% in terms of macro-F1. Under L principle, on average, VRoC and VAE-LSTM outperforms baselines by 13.2% and 14.9% in terms of macro-F1. VAE's compact latent representations contribute to these results the most. Compared to VAE-LSTM, the proposed co-train engine boosts the performance of VRoC one step further.

Table 1: Comparison between VRoC and baselines on the rumor detection task.

	Accuracy	Precision	Recall	Macro-F1
CRF*	-	0.667	0.556	0.607
GAN-BOW*	0.781	0.782	0.781	0.781
GAN-CNN*	0.736	0.738	0.736	0.736
GAN-GRU*	0.688	0.689	0.688	0.687
VAE-LSTM	0.833	0.834	0.834	0.833
VRoC	**0.876**	**0.877**	**0.876**	**0.876**
DataAUG* (L)	0.707	0.580	0.497	0.535
DMFN* (L)	0.703	0.667	0.670	0.657
DTSL* (L)	-	0.560	**0.794**	0.615
VAE-LSTM (L)	0.736	0.746	0.736	0.735
VRoC (L)	**0.752**	**0.755**	0.752	**0.752**

4.2 Rumor Tracking

Comparison results between VRoC and baselines on the rumor tracking task are shown in Table 2. VRoC achieves the highest macro-F1, but it does not outperform baselines by a large percentage. In rumor tracking, raw data might be a preferable data source since they contain keywords and hashtags that can be used to directly track the topic. For long posts, rumor tracking can be effortlessly accomplished by retrieving the hashtags. Compared to models that use raw data, VRoC has advantages when dealing with imbalanced and unseen data since it can extract compact information from a few posts and generalize to a broader range of data.

Table 2: Comparison between VRoC and baselines on the rumor tracking task.

			S	G	F	C	O
	Accuracy	Macro-F1	F1	F1	F1	F1	F1
CNN	0.570	0.574	0.589	**0.534**	0.777	0.571	0.400
LSTM	0.585	0.585	0.607	0.352	**0.804**	**0.711**	0.453
VAE-LSTM	0.609	0.612	**0.666**	0.515	0.641	0.694	0.545
VRoC	**0.644**	**0.632**	0.611	0.520	0.640	0.685	**0.703**

4.3 Stance Classification

Comparison results between VRoC and baselines on the rumor stance classification task are shown in Table 3. Stance classification is the hardest component in rumor classification. The reason is two-fold: four-way classification problems are naturally more difficult than binary classification problems, and imbalanced data exaggerate the difficulty. In addition, the stance classification is not as easy as the tracking task. Stance classifier has to extract detailed patterns from a sentence and consider the whole sentence. In the tracking task, posts can be classified together by filtering out the obvious keywords, but stance is related to the semantic meaning of the whole sentence and hence is more complicated.

Baselines in the comparison all suffer from the extremely low F1 score on Deny class, which is caused by the small size of the Deny instances in the dataset. Feature extraction from raw data results in severely imbalanced performance among different classes. VRoC's and VAE-LSTM's classifiers are trained under latent representations. Although data imbalance affects the performance of VRoC and VAE-LSTM, the impact is not as drastic as in other baselines. Compared to state-of-the-art baselines that concentrate on stance classification, VRoC's stance classification component provides the highest macro-F1 scores under both training principles and VAE-LSTM follows as the second best.

Table 3: Comparison between VRoC and baselines on the rumor stance classification task.

	Accuracy	Macro-F1	Support F1	Deny F1	Comment F1	Query F1
MT-US*	-	0.400	0.355	0.116	**0.776**	0.337
MT-ES*	-	0.430	0.314	0.158	0.739	**0.531**
VAE-LSTM	0.464	0.461	0.447	0.395	0.588	0.416
VRoC	**0.533**	**0.522**	**0.452**	**0.415**	0.712	0.511
TreeCRF* (L)	-	0.440	0.462	0.088	**0.773**	0.435
LinearCRF* (L)	-	0.433	0.454	0.105	0.767	**0.495**
VAE-LSTM (L)	0.467	0.459	**0.463**	0.423	0.567	0.384
VRoC (L)	**0.480**	**0.473**	0.452	**0.429**	0.596	0.416

4.4 Veracity Classification

Comparison results between VRoC and baselines on the rumor veracity classification task are shown in Table 4. VRoC and VAE-LSTM achieve the highest macro-F1 and accuracy compared to baselines. On average, VRoC outperforms MT-UA and other baselines under L principle by 24.9% and 11.9% in terms of macro-F1, respectively. Rumor veracity classification under L principle is particularly difficult since there is no previously established verified news database. An unseen news on a previously unobserved event has to be classified without any knowledge related to this event. For example, you observed some news on A event, and you need to verify whether a news from an unrelated B event is true or not. Without a verified news database, the abstracted textural patterns of sentences are utilized to classify unobserved news. Latent representations extracted by VAEs are hence very helpful to generalize in veracity classification. The outperformance of VRoC and VAE-LSTM over baselines under L principle demonstrates the outstanding generalization ability of VAEs. In addition, VRoC beats VAE-LSTM in terms of both accuracy and macro-F1 in all cases. These results further demonstrate the power of the proposed co-train engine.

4.5 VRoC and VAE-LSTM

As shown in Tables 1-4, VRoC outperforms all the baselines in terms of macro-F1 and accuracy in all four rumor classification tasks, while VAE-LSTM stands as the second best. On average, in all four tasks, VRoC and VAE-LSTM surpass the baselines by 10.94% and 7.64% in terms of macro-F1 scores. The ability of VAE-based rumor classifier is confirmed by these results. The advantage of latent representations over raw tweet data is demonstrated as well. VRoC achieves higher performance than VAE-LSTM because of the designed co-train engine. VRoC's latent representations are more suitable and friendly to the rumor classification components. Furthermore, the co-train engine introduces randomness into the training process of VRoC, hence the robustness and generalization abilities of VRoC are improved. Dimensionality reduction [2] is also realized by the VAEs to further aid the generalization. Semantically and syntactically related samples are placed near each other in latent space. Although future news are unobserved, they may contain similar semantic and/or syntactic features to those observed news. Thus VRoC could generalize and place the new latent representations close to the old ones and classify them without the need of retrain. VRoC and VAE-LSTM are efficient since all four tasks can be performed in parallel. Assume the serial runtime is T_s, the parallel runtime is $T_p = \frac{T_s}{p}$ (if all four tasks are parallelized and $p = 4$ is number of processors used in parallel), then the efficiency $E = \frac{Speedup}{p} = \frac{T_s/T_p}{p} = 1$.

Table 4: Comparison between VRoC and baselines on the rumor veracity classification task. Lc represents that the news related to Charlie Hebdo is left out while trained under L principle.

	Accuracy	Macro-F1	True F1	False F1	Unverified F1
MT-UA*	0.483	0.418	-	-	-
VAE-LSTM	0.628	0.627	0.691	0.576	0.615
VRoC	**0.667**	**0.667**	**0.745**	**0.632**	**0.624**
MTL2* (Lc)	0.441	0.376	-	-	-
MTL3* (Lc)	0.492	0.396	**0.681**	0.232	0.351
VAE-LSTM (Lc)	0.507	0.503	0.545	**0.449**	**0.515**
VRoC (Lc)	**0.531**	**0.513**	0.564	0.434	0.480
TreeLSTM* (L)	0.500	0.379	0.396	**0.563**	0.506
VAE-LSTM (L)	0.494	0.475	0.429	0.472	**0.523**
VRoC (L)	**0.521**	**0.484**	**0.480**	0.504	0.465

5 CONCLUSIONS

Various rumors on social media during emergency events threaten the internet credibility and provoke social panic, and may lead to long-term negative consequences. Recent rumors on the 2019 novel coronavirus stand as a shocking example. To mitigate such issues provoked by rumors, we propose VRoC, a variational autoencoder-aided rumor classification system consisting of four components: rumor detection, rumor tracking, stance classification, and veracity classification. The novel architecture of VRoC, including its suitable classification techniques in the tasks associate with rumor handling and the designed co-train engine contribute to the high performance and generalization abilities of VRoC. Facing previously observed or unobserved rumors, VRoC outperforms state-of-the-art works by 10.94% in terms of macro-F1 scores on average. VRoC is efficient not only in the sense of parallel computing, but also in terms of speed of rumor detection. The high accuracy shortens the time of detection and hence helps with reducing the chance of rumor resurfacing. As part of our future work, we would like to investigate the cooperation of four components to improve the overall performance. We will also extend our work by including visual contents such as images, and explore the influence between text and visual features in rumors.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support by the National Science Foundation under the Career Award CPS/CNS-1453860, CCF-1837131, MCB-1936775, CNS-1932620, and the DARPA Young Faculty Award, under grant number N66001-17-1-4044. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied by the Defense Advanced Research Projects Agency, the Department of Defense or the National Science Foundation.

REFERENCES

[1] Hunt Allcott, Matthew Gentzkow, and Chuan Yu. 2019. Trends in the diffusion of misinformation on social media. *Research & Politics* 6, 2 (2019), 205316801984855 4.
[2] Jinwon An and Sungzoon Cho. 2015. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE* 2, 1 (2015).
[3] Juan Cao, Junbo Guo, Xirong Li, Zhiwei Jin, Han Guo, and Jintao Li. 2018. Automatic rumor detection on microblogs: A survey. *arXiv preprint arXiv:1807.03505* (2018).
[4] 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.
[5] Tong Chen, Xue Li, Hongzhi Yin, and Jun Zhang. 2018. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 40–52.
[6] Xinran Chen, Sei-Ching Joanna Sin, Yin-Leng Theng, and Chei Sian Lee. 2015. Why students share misinformation on social media: Motivation, gender, and study-level differences. *The Journal of Academic Librarianship* 41, 5 (2015), 583–592.
[7] Jon Cohen and Dennis Normile. 2020. New SARS-like virus in China triggers alarm.
[8] Xishuang Dong, Uboho Victor, Shanta Chowdhury, and Lijun Qian. 2019. Deep Two-path Semi-supervised Learning for Fake News Detection. *arXiv preprint arXiv:1906.05659* (2019).
[9] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. 2018. Neural architecture search: A survey. *arXiv preprint arXiv:1808.05377* (2018).
[10] Ortiz-Ospina Esteban. 2019. The rise of social media. https://ourworldindata.org/rise-of-social-media.
[11] FactCheck. 2020. Coronavirus Misinformation Spreads Like a Virus. https://www.factcheck.org/2020/01/coronavirus-misinformation-spreads-like-a-virus/.
[12] Adrien Friggeri, Lada Adamic, Dean Eckles, and Justin Cheng. 2014. Rumor cascades. In *Eighth International AAAI Conference on Weblogs and Social Media*.
[13] Karl Friston. 2009. The free-energy principle: a rough guide to the brain? *Trends in cognitive sciences* 13, 7 (2009), 293–301.
[14] Genevieve Gorrell, Kalina Bontcheva, Leon Derczynski, Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. RumourEval 2019: Determining Rumour Veracity and Support for Rumours. *arXiv preprint arXiv:1809.06683* (2018).
[15] Manish Gupta, Peixiang Zhao, and Jiawei Han. 2012. Evaluating event credibility on twitter. In *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 153–164.
[16] Sardar Hamidian and Mona T Diab. 2015. Rumor detection and classification for twitter data. In *Proceedings of the Fifth International Conference on Social Media Technologies, Communication, and Informatics (SOTICS)*. 71–77.
[17] Sooji Han, Jie Gao, and Fabio Ciravegna. 2019. Data Augmentation for Rumor Detection Using Context-Sensitive Neural Language Model With Large-Scale Credibility Corpus. (2019).
[18] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[19] Lowther Jenn. 2009. Microblogging is one of the top four trends in social media. https://www.straight.com/article-200494/microblogging-one-top-four-trends-social-media.
[20] McDonald Jessica. 2020. Q&A on the Wuhan Coronavirus. https://www.factcheck.org/2020/01/qa-on-the-wuhan-coronavirus/.
[21] Zhiwei Jin, Juan Cao, Han Guo, Yongdong Zhang, Yu Wang, and Jiebo Luo. 2017. Detection and analysis of 2016 us presidential election related rumors on twitter. In *International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation*. Springer, 14–24.
[22] Wernau Julie. 2020. Virus Sparks Chinese Panic Buying, Travel Cancellations and Social-Media Misinformation. https://www.wsj.com/articles/coronavirus-sparks-chinese-panic-buying-travel-cancellations-and-social-media-misinformation-11579698948.
[23] Matsa Katerina and Shearer Elisa. 2018. News Use Across Social Media Platforms 2018. Retrieved September 9, 2019 from https://www.journalism.org/2018/09/10/news-use-across-social-media-platforms-2018/
[24] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
[25] Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. All-in-one: Multi-task learning for rumour verification. *arXiv preprint arXiv:1806.03713* (2018).
[26] Sumeet Kumar and Kathleen M Carley. 2019. Tree LSTMs with Convolution Units to Predict Stance and Rumor Veracity in Social Media Conversations. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*. 5047–5058.
[27] Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *2013 IEEE 13th International Conference on Data Mining*. IEEE, 1103–1108.
[28] Stephan Lewandowsky, Ullrich KH Ecker, Colleen M Seifert, Norbert Schwarz, and John Cook. 2012. Misinformation and its correction: Continued influence and successful debiasing. *Psychological Science in the Public Interest* 13, 3 (2012), 106–131.
[29] Quanzhi Li, Qiong Zhang, and Luo Si. 2019. Rumor Detection by Exploiting User Credibility Information, Attention and Multi-task Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1173–1179.
[30] Yanling Li, Guoshe Sun, and Yehang Zhu. 2010. Data imbalance problem in text classification. In *2010 Third International Symposium on Information Processing*. IEEE, 301–305.
[31] Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. 2017. Hierarchical representations for efficient architecture search. *arXiv preprint arXiv:1711.00436* (2017).
[32] Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Detect rumor and stance jointly by neural multi-task learning. In *Companion Proceedings of the The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 585–593.
[33] Jing Ma, Wei Gao, and Kam-Fai Wong. 2019. Detect Rumors on Twitter by Promoting Information Campaigns with Generative Adversarial Learning. In *The World Wide Web Conference*. ACM, 3049–3055.
[34] Field Matt. 2020. Fake news epidemic: Coronavirus breeds hate and disinformation in India and beyond. https://thebulletin.org/2020/01/fake-news-epidemic-coronavirus-breeds-hate-and-disinformation-in-india-and-beyond/.
[35] Livingston Mercey. 2020. Coronavirus fact check: How to spot fake reports about the mysterious disease. https://www.cnet.com/how-to/false-information-about-coronavirus-here-are-the-top-rumors-spreading-about-it/.
[36] Duc Minh Nguyen, Tien Huu Do, Robert Calderbank, and Nikos Deligiannis. 2019. Fake News Detection using Deep Markov Random Fields. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 1391–1400.
[37] Ray Oshikawa, Jing Qian, and William Yang Wang. 2018. A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770* (2018).
[38] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. 2019. Regularized evolution for image classifier architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4780–4789.
[39] Saad Sadiq, Nicolas Wagner, Mei-Ling Shyu, and Daniel Feaster. 2019. High Dimensional Latent Space Variational AutoEncoders for Fake News Detection. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 437–442.
[40] Romm Tony. 2020. Facebook will remove misinformation about coronavirus. https://www.washingtonpost.com/technology/2020/01/30/facebook-coronavirus-fakes/.
[41] Vincent Van Asch. 2013. Macro-and micro-averaged evaluation measures [[basic draft]]. *Belgium: CLiPS* 49 (2013).
[42] Qiang Zhang, Aldo Lipani, Shangsong Liang, and Emine Yilmaz. 2019. Reply-Aided Detection of Misinformation via Bayesian Deep Learning. In *The World Wide Web Conference*. ACM, 2333–2343.
[43] Thomas Zoe. 2020. Coronavirus: How Facebook, TikTok and other apps tackle fake claims. https://www.bbc.com/news/technology-51337357.
[44] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)* 51, 2 (2018), 32.
[45] Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, and Michal Lukasik. 2016. Stance classification in rumours as a sequential task exploiting the tree structure of social media conversations. *arXiv preprint arXiv:1609.09028* (2016).
[46] Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2017. Exploiting context for rumour detection in social media. In *International Conference on Social Informatics*. Springer, 109–123.