Fake News Detection by Variational Autoencoder and Topic Modelling

Majid Feiz
Computer Science and Engineering
University of Connnecticut
Storrs, CT, US
majid.feiz@uconn.edu

Marjan Hosseini
Computer Science and Engineering
University of Connnecticut
Storrs, CT, US
marjan.hosseini@uconn.edu

***Abstract*—With the growth of social media and User Generated Content in the recent decade, people are more exposed to the news with unreliable sources and fake content. Validating the credibility of such information, however, is not a trivial task for the majority of the users, and spreading false information could potentially lead to losses and crimes. As a result, it is essential to develop accurate techniques for distinguishing fake and real news. Due to the absence of multimedia and information about the author and spread patterns in many real-world sources of news, we focus on extracting relevant features only from the textual content. Our contribution is proposing a multi-modal approach that aggregates the hidden representation of textual news using a variational autoencoder and topic-related features inferred from the Latent Dirichlet Allocation (LDA) mixture model. We achieve a more accurate and interpretable model by concatenating these two feature sets. Furthermore, we show that our model facilitates clustering fake and real news efficiently.**

***Index Terms*—Fake News, Topic Identification, Variational Autoencoder, Big Data, Natural Language Processing**

I. Introduction

In the recent decade, online social media are rapidly growing. Unlike traditional media such as newspapers and TV channels, in online platforms anyone can share and spread any type of information without providing the authenticity of it. On the other hand, the majority of the people rely on the information coming from these platforms. In the US, 62% of adults use social media as their source of daily news [1]. However, around 70% of people can successfully validate the authenticity of the news they read [2]. As a result, social media platforms have become a source of spreading fake news and information, which could cause many losses and crimes. Additionally, as social networks are being used as the source of news for most individuals, fake news now spreads at a faster pace and has a greater impact than ever before. This makes detection of fake news an extremely important challenge [3] and consequently, it has become vital to provide accurate tools for detecting fake news. However, not only the accuracy, but the interpretability is of high importance to understand the underlying causes of fake news. The contribution of this project is two-fold. We developed a framework and coupled LDA Bayesian model to the variational autoencoder to increase interpretability and also the accuracy of the model using only the textual content.

A. Organization

This report is organized as follows. In the remaining part of this section, we will explain the background that motivated us in this project. Then we briefly discuss existing directions for solving the problem of fake news detection in the literature and their drawbacks and advantages. Then we provide an overview of the proposed method, our contribution, and the potential challenges. In the next section, we review some of the most relevant papers to this project. Sections III and IV present a more detailed sketch of the components and flow of the information in the system and datasets that we have studied. Sect. V provides details in the experimental setting and configuration of the components. We report the results in Sect. VI and discuss the advantages and drawbacks of our model in Sect. VII. Then, we state some possible future work after this study in Sect. VIII. At the end, we conclude the report and bring final remarks in Sections IX and X.

B. Background

Fake news are the articles or piece of news that are intentionally false or misleading [4] or a story in online social media which cannot be authenticated [5]. Fake news detection usually refers to the task of classifying the news into fake and real when the *fake/real* labels are part of the input. More specifically, classification is the process in which a learner or classifier aims to find the underlying relationship between the input and the output data using a training set, and makes a predictive model of the distribution of the class labels for the features accordingly. Using these models, one can predict the label of any future test set based only on the values of the features in the input data. This process requires the classifier to know the class labels in advance, that is the reason it is called *supervised*. This section provides a brief background about the model's key components.

1) Latent Dirichlet Allocation: (LDA) [6] is a statistical admixture model for clustering the words into topics and making inference on the distribution of the topics in the text. Moreover, it provides the distribution of the words in each topic. These distributions can be estimated by the posterior probability of the trained parameters in the joint equation in the model. Fig. 1 is the plate model of the probabilistic graphical model with the corresponding parameters whose

posterior mean values will be estimated by training the model. The joint probability distribution of the model is computed in Eq. 1.

$$p(\theta, \beta, Z, W|\eta, \alpha) = p(\beta|\eta)p(\theta|\alpha)p(Z|\theta)p(W|\beta_{z_{nj}}) \quad (1)$$

Here, we are interested in inferring the parameters θ (distribu-

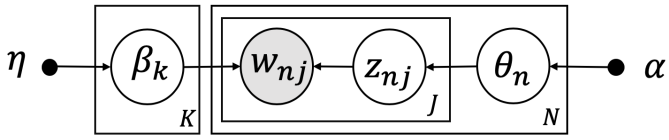

Fig. 1. Probabilistic Graphical model of LDA.

tion of the topics in the news) and β (distribution of the words in different topics) and also the frequent words that appeared in clusters (topics). Matrix Z is the topic assignments for the individual words and matrix W is the observed variable (news body). N, J, and K are the number of news, words in the news and clusters respectively, and η and α are hyper-parameters.

2) Autoencoder: is a combination of an encoder function, which converts the input data into a different representation, and a decoder function, which converts this new representation back into the original format. Autoencoders are trained to preserve as much information as possible when input is run through the encoder and then the decoder, but they are also trained to make the new representation have various specific properties. Different kinds of autoencoders aim to achieve different types of properties [7]. Autoencoders have been successfully applied to textual input by Salakhutdinov and Hinton [8], [9] for dimensionality reduction, and information retrieval tasks referred to "semantic hashing". Autoencoders have also been applied for anomaly detection, such as fraudulent transaction [10]. The autoencoder will have trouble reconstructing the fraudulent transaction, and hence the reconstruction error will be high, and the transaction then can be flagged based on a specified threshold error. Autoencoders has been used to learn latent visual and textual representations to aid fake news detection [11]. *Variational autoencoder (VAE)* draws from a probability distribution in the latent space rather than a point estimate in autoencoders. Given the input $\boldsymbol{x}$, the encoder constructs latent representation $\boldsymbol{z}$, and the decoder generates the input using latent space, and the objective is minimizing the KL divergence between $p(\boldsymbol{x}|\boldsymbol{z})$ (likelihood) and latent space probability distribution $q(\boldsymbol{z}|\boldsymbol{x})$.

3) Long short-term memory: (LSTM) [12] is a type of recurrent neural network (RNN). The main incentive for developing LSTM was the error flow in RNNs and addressing the problem of vanishing gradient and long time lags in methods that deal with sequential data such as traditional RNNs and HMMs. LSTM unit consists of input, output, and forget gates and solves that problem by remembering data with arbitrary gaps using its three gates. If the sequence of state h in LSTM are indexed by t, then we have

$$[f_t, i_t, o_t] = \sigma(Wh_{t-1} + Ux_t + b)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot (Wh_{t-1} + Ux_t + b)$$
$$h_t = o_t \odot tanh(c_t)$$

where σ is the sigmoid function, f_t, i_t, o_t and c_t are forget, input and output gates and memory state at t and x_t is the input. W and U are weight matrices and b is the bias vector and $\odot$ denotes element-wise matrix multiplication. *Bidirectional-LSTM* [13] is an improvement over traditional LSTMs in terms of model performance on sequence classification problems. The reason of this enhancement is that it combines two independent recurrent layers side-by-side, one of which receives the input sequence and the other one a reversed copy of it.

4) word2vec: Word2Vec [14] is a method for learning the word embeddings using a two-layer neural network. The input is the whole corpus and the output is the vectors corresponding to the words, such that the cosine similarity between the vectors associated with semantically similar words is closer to 1. In other words, the cosine similarity between words *embedding* is correlated to the similarity of corresponding words in terms of meaning according to the context.

5) Classification: Classification (supervised learning) is the process in which a learner or classifier aims to find the underlying relationship between the input and the output data using a training set (fitting), and according to the relationship makes a predictive model of the distribution of the class labels for the features. Using these models, it can predict the labels of any future test set based only on the values of the features in the input data. This process requires the classifier to know the class labels in advance, that is the reason it is called supervised. Here we briefly explain some of the classifiers that have been employed to evaluate the performance of the model in this project.

a) Support Vector Machine: (SVM) classifier [15] in the binary classification setting aims to classify the samples by solving an optimization problem in order to divide the data space (feature space) into two sub-spaces by fitting a hyper-plane that almost separates the samples from different classes, using a soft margin. This hyper-plane maximizes the margin to the training observations from two classes.

b) non-linear SVM: can be employed when there is not a linear boundary between two classes, because in this case, the performance of linear classifiers such as linear SVM would drop. In the above-mentioned conditions, non-linear SVM uses the same concepts to generate a nonlinear decision boundary between data points from two classes. This is performed by enlarging the feature space by applying functions of the features known as kernels, which are computationally efficient.

c) Naïve Bayes: is a simple linear probabilistic classifier based on the Bayes theorem, it constructs simple but efficient and well-performing models in many real-world problems. Naïve Bayes considers the mutual independence assumption among features for a given class label and that is the reason it

is called naïve. Having this assumption, it is easier to compute the probabilities, *i.e.* the probability that each sample belongs to a particular class. Unless this assumption (independence of variables) is strongly violated, this classifier is very efficient, accurate and sometimes performs even better than other powerful classifiers. It is also robust to noise and irrelevant attributes [16].

d) k-Nearest Neighbors: or k-NN is a non-parametric lazy classifier, *i.e.* it does not learn a particular model from the training dataset, but delay the predicting of class labels by memorizing the training data to use in the test stage. For prediction, it assigns the new observation a label by estimating the conditional probability that the new sample belongs to that class. It considers the k nearest data points in the training set to the test sample and decides about the label of a new sample according to those neighbors [15]. An advantage of k-NN is that it is non-parametric, so it does not make any prior assumption about the distribution of the data. This is quite helpful in many situations.

e) MLP: or feedforward neural network is a method of a deep artificial neural network classifier. It is composed of more than one perceptron with at least three layers of nodes, an input layer, an output layer that makes predictions about the input, and an arbitrary number of hidden layers. Every node in a hidden layer operates on activations from the preceding layer and transmits activations forward to nodes of the next layer. Training involves adjusting the parameters or the weights and biases of the model in order to minimize error.

f) Random Forest: is an ensemble method where each of the ensemble classifiers is forming a decision tree classifier. Following a bagging procedure to generate a new group of training sets, each group will be fed to a decision tree and the summation of all output will form the final output of the model. The individual decision trees are generated using a random selection of attributes at each node to determine the split. During classification, each tree votes and the most popular class is returned.

6) Feature Selection: is the task of selecting some features among all the feature set based on pre-defined criteria such that the most important features that have the most contribution for predicting the output would be selected with the higher priority [17], [18]. Advantages of feature selection include a reduction in the computational cost of training the classifier, simplification of the model structure, preventing overfitting by removing noise and redundant information and facilitating model interpretation and data understanding. One categorization of feature selection algorithms is univariate versus multivariate feature selection based on whether the importance of features is assessed individually or as a group of features. Univariate feature selection algorithms rank the features according to properties of the data obtained using evaluation metrics and the relevance of a single feature to the output, then the top features are selected [19], [20]. These algorithms ignore possible inherent interactions among the features so redundant features might appear close to each other in the ranking list and consequently they all will be selected [21]. However, they are very effective, fast, easy to implement and scalable on large datasets. Since feature selection is not the main purpose of this project, we have been using only univariate feature selection with the following evaluation metric for ranking the features.

a) Chi^2: correlation metric shows if there is a relationship between two random variables. It can also be used to test whether or not several events are occurring in equal frequencies, or according to a distribution. In other words, if we have two random variables, Chi^2 scores according to how well the frequency of events (measurements) deviates from the estimation (expected frequency) [22].

$$\mathcal{X}^2 = \frac{\text{(observed - expected)}^2}{\text{expected}}$$

b) Gini index: is an alternative impurity metric for information gain which measures the variance of a distribution associated with different classes in the dataset $\mathcal{D}$ [23]. This index has been used in classification and decision trees (CART) for building the decision tree. If P_i is the relative frequency of class i in $\mathcal{D}$ Gini index is computed as:

$$Gini(\mathcal{D}) = 1 - \sum_{i \in classes} P_i^2$$

7) Dimensionality Reduction: methods transform the data from a high-dimensional into a lower-dimensional space such that the new representation holds some meaningful properties of the original data. Like feature selection methods, they are desirable due to removing the dimensionality of the data, but the difference is that in feature selection algorithms the value of selected features does not change. They can be applied for better visualization of the data and facilitating unsupervised methods.

a) Principle Component Analysis: (PCA) is an orthogonal linear dimension reduction method that employs Eigenvalue decomposition to factorize data matrix in terms of its Eigenvectors and Eigenvalues and find the axes in which data has more variance. Then it projects the data points to these new axes such that the greatest variance by any projection of the data comes to lie on the first coordinate. In other words, it transforms many correlated variables into a smaller number of uncorrelated variables (principal components). The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. However, due to its linearity, it is often unable to capture the potential non-linear relationship between variables.

b) t-Distributed Stochastic Neighbor Embedding: (tSNE) is a non-linear probabilistic dimensionality reduction method, originally for visualizing high dimensional data, based on the idea that similar objects in the high dimensional space should be represented proportionally closer to each other than dissimilar objects in lower-dimensional space. Hence, it minimizes the KL divergence between a joint probability distribution, in the high dimensional space and a joint probability distribution

in the low dimensional space. tSNE uses a symmetrized version of the SNE cost function with simpler gradients and a Student t distribution to compute the similarity between two points in the low dimensional space. Unlike PCA which is a linear transformation, it can capture the potential non-linear relationship between random variables, with the cost of more computational complexity.

C. Existing Directions

Fake news detection techniques can be generally categorized into three types of approaches according to which type of information is extracted for detecting if the news is fake or not. Propagation-based, source-based and content-based. The motivation for propagation-based techniques is that fake news is spread differently than real news in social media in terms of the speed and the pattern of propagation [24]. Source-based methods benefit the information about the source of the news. This information includes the general behavior of the source in social media and is able to detect false news fast and efficient [25]. However, in many cases, the information about the spread of the news or the author is not provided and even if provided, it neglects that some real users can unintentionally spread some of the false news started from other sources. On the other hand, content-based techniques focus on the characteristics of the news to extract different types of features, for example linguistic (lexical or syntactic) characteristics. The assumption in this type of approach is that some features such as the language, topic, and style of the news body are discriminative attributes for validating its authenticity [26]–[28]. Because of the above-mentioned reasons, in this study, we focus on the content of the news rather than the source or propagation-based techniques. Content-based features can be categorized into three main types. Syntactic features include statistical information about the sentences, like the frequency of different parts of speech and specific patterns and quantifying the complexity of the sentences. Lexical features concern the usage of actual words in the texts such as bi-grams and tri-grams. Semantic features refer to sentimental characteristics of the content and they are usually extracted using advanced NLP and data mining techniques such as sentiment analysis, opinion, and emotion mining approach. Recently, extracting word embeddings [14] and topics of text [29] have been proposed as potentially useful information too. However, employing them in fake news detection methods gain relatively less attention compared to other characteristics. In this regard, we would like to include these types of features in our method. To extract the features, traditional machine learning methods handcraft the feature space manually which is a tedious task and subject to bias, while with the improvement of deep learning frameworks, hidden representations of the text can be obtained automatically and simpler [30]. To produce efficient and compact features, deep learning approaches concern is designing suitable layers and architecture of the network to capture all types of contents. Then for classifying the news either machine learning methods such as SVM, Random Forest, Decision Tree, Logistic Regression, CRM, and HMM are applied or deep learning methods such as RNNs and Convolutional Neural Networks (CNN) [31].

D. Proposal and contribution

Inspired by the idea proposed in [3], in this study we aim to capture the hidden representation of textual content employing word2vec [14] word embedding, and VAE with stacked bi-directional LSTM units. This method has been proven faster and more accurate than regular LSTM and is suitable for the settings in which all of the input sequences are available such as our case [13]. However, we intend to use only the textual data in our model because many news lacks the corresponding multimedia. We also believe that some higher-level information such as the topic of the text might not be properly captured using VAE, so our contribution in this project is incorporating LDA [6] in the model. We infer dataset-specific topics in the news and then annotate each news text according to the distribution of the topics in it. We then concatenate newly obtained features inferred from the LDA model to the VAE features. Then we train different classifiers using the combination of them and evaluate the performance of the model.

E. Challenges

One of the potential challenges of this project is the absence of an agreed benchmark dataset due to the unclear definition of fake news in the literature, and also finding relevant data and the process of authentication of the news. Besides, the abundance of real news articles are usually much more than the fake ones which leads to creating an imbalance in the collected data [32]. However, there are some publicly available datasets with *fake*/*real* labels, and the problem of imbalance can be alleviated using resampling methods. Another challenge for our model is that the majority of the available news articles are political. As a result, we need to focus on obtaining sub-topics in the news. In addition, LDA is a parametric method and the number of topics should be provided in advance which makes tuning this component challenging. Plus, tuning VAE is demanding due to the numerous hyper-parameters.

II. Related Work

In this section we review some the most relevant articles to this project. Perhaps the most relevant method is "Multi-modal Variational Autoencoder for Fake News Detection" (MVAE) [3], proposed by Khattar et. al in 2019. In this paper they offer a shared representation of the news for further classification of it as fake or real. The original encoder architecture consists of two encoders, one for extracting textual and visual information of the news, then they concatenate two modals of the news and output a general representation of the input. The textual encoder extract the content by recurrent networks (RNN). Since RNN is inherently sequential it would be a good choice in text especially. To overcome the exploding and vanishing gradient in RNN [33], they use a stacked bi-directional LSTM [12] cells. The contribution of the paper is proposing a multi-modal framework which employs textual

and visual content of the news for feature extraction. They apply the model on two datasets Twitter and Weibo, compared their model with six other models and showed their method outperforms them.

In another related study entitled "Detecting fake news over online social media via domain reputations and content understanding" [34], used Web sites and reputations of the publishers for classifying fake and real news based on their registration patterns, web site ages, domain rankings, domain popularity, and the probabilities of news disappearance from the Internet. LDA is used to analyze the similarity of the fake vs. real articles using the research data in BuzzFeed News.

The paper "Assessment of tweet credibility with LDA features" [29] use topic modeling to assess information credibility on Twitter. Two factors of *tweet topic* and *user topic* features derived from the LDA model confirming topical features is an effective way to assess tweet credibility. Their analysis is based on whether a tweet has an information source, how serious it is, and if the user of the tweet is reliable.

Another relevant work, "High Dimensional Latent Space Variational Autoencoders for Fake News Detection" [35] proposed a method that builds a latent representation of natural language to capture the underlying hidden meanings of human communication accurately to classify fake news. The pre-trained word2vec model from Google's News Corpus used for the datasets. Additional approach provided in the paper, "A Multi-modal Framework for Fake News Detection" [11] uses both the textual, and visual features of an article along with utilizing language models (like BERT) to learn text and image features for fake news detection. They used Twitter and Weibo datasets to perform their experiments.

The paper, "Detecting fake news stories via multi-modal analysis" [36] is using multi-modal approach by combining text, and visual content of online news to detect fake news. This work uses Fake News data set available on "Kaggle Fake News Dataset" (Kaggle, 2017). They compared and contrasted them with another data set that includes credible news stories from three news sources: The New York Times, Reuters, and Public Broadcasting Service (PBS). Implemented various machine learning algorithms such as Logistic regression, K nearest neighbors, Naïve Bayes etc., with Text only, Visual only, and Text + Visual. Their experimental results confirm other papers that a multi-modal where uses more than one resources approach outperforms single-modality approaches, allowing for better fake news detection.

III. System Design

This section explains the general overview of our model and information flow of the framework, individual components, and their motivations. The system is based on a multi-modal approach that we designed for the classification of news into fake and real. We consider our model multi-modal because we use two sources of information for feature extraction. A VAE for extracting textual content and LDA model for obtaining text topics features. Our motivation for discarding visual content and other information such as source and spread pattern of the news is that not all of the datasets include associated multimedia such as image or video or other types of attributes. So here, we focus on the latent clues in the textual content. Besides, since VAE is not originally designed to capture information such as topics, we add an LDA model which offers increased interpretability to the model. We hypothesis that concatenating latent representation produced by VAE and topic-related features together would enhance the accuracy of the model as well. Fig. 2 is an overview of the flow of the information in our method. Each box represents either a task/component or an important input/output.

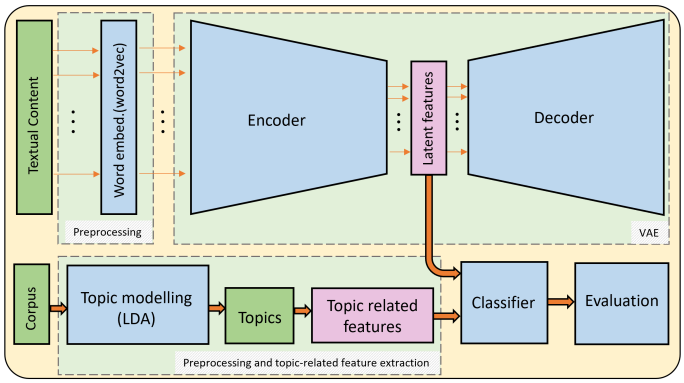

Fig. 2. General overview of the model. In the upper part, the data is pre-processed, transformed using the word2vec model, and enters the VAE unit, where the latent features are extracted. Meanwhile, in the lower part, the topics are obtained through training the LDA model on pre-processed data. Then the latent and topic-related features are concatenated and being used by the classifier.

A. Preprocessing

Before feeding the data into our model, we apply standard text pre-processing techniques such as removing stop words and then tokenize the text into words. Then only for VAE, we transform the words to w-dimensional vectors by applying the distributed word2vec pre-trained model [14]. As a result of this transformation, semantically similar words are mapped closer to each other in the space rather than very different words as it was explained in Sect. I-B. The set of vocabulary detected by word2vec model V changes according to the value of w.

B. Variational Autoencoder

Motivated by the architecture proposed by MVAE [3], one of the main components in our model, is VAE, which is composed of an encoder, a decoder, and a classifier. In the architecture of the encoder and decoder, we stacked layers of bi-directional LSTM and fully connected layers. When VAE includes only encoder and decoder, it is an unsupervised component that tries to reconstruct the input by optimizing a bound on the marginal likelihood and minimizing categorical cross-entropy loss. Here, we coupled a classifier along with encoder and decoder such that during training the parameters are optimized not only by reconstruction error of VAE ($\mathcal{L}_{rec}$),

but also by MSE error of the classifier ($\mathcal{L}_{MSE}$). So we will have the following loss functions:

$$\mathcal{L}_{CE} = -\mathbb{E}_{i\sim\mathcal{D}}\left[\sum_{j=1}^{l_i}\sum_{v\in V} 1_{v=t_i^{(j)}} \log t_i^{(j)}\right] \tag{2}$$

$$\mathcal{L}_{KL} = -\frac{1}{2}\sum_{f=1}^{n_f}(\mu_f^2 + \sigma_f^2 - \log(\sigma_f) - 1) \tag{3}$$

$$\mathcal{L}_{rec} = \mathcal{L}_{CE} + \mathcal{L}_{KL} \tag{4}$$

$$\tag{5}$$

where $\mathcal{L}_{CE}$ and $\mathcal{L}_{KL}$ are the cross entropy loss function and KL divergence respectively. $\mathcal{D}$ is the set of posts/news, which is indexed by i. Variable l_i denotes the length of post i, V is the set of vocabulary defined by word2vec model, and $t_i^{(j)}$ is the jth word in post i.

$$\mathcal{L}_{MSE} = -\mathbb{E}_{i\sim D}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \tag{6}$$

where y_i and $\hat{y}_i$ denote the post label and the probability that post i is fake detected by the classifier. The optimal parameters θ^* minimized the total loss functions.

$$\theta^* = \text{argmin}_\theta(\mathcal{L}_{rec} + \mathcal{L}_{MSE}) \tag{7}$$

The encoder is composed of two bidirectional LSTM (Bi-LSTM) and two fully connected layers. Our motivation for using Bi-LSTM is that sequential layers often suffer from a problem of vanishing gradients which might lead to learning inefficient dependencies between the words by the model. LSTM layers, however, overcome this problem by including the forget gate and hidden states. Furthermore, Bi-LSTM improves the accuracy over traditional LSTM, by utilizing two independent LSTM layers working side by side. The decoder has a similar architecture to the encoder but in opposite direction. The output of the decoder is the reconstructed posts. Together with the encoder and decoder, we train a 2-layered NN classifier. Our motivation for adding this part is obtaining more predictive feature space by including a supervised component to the model. Overall, in this project, the desired output from the VAE component is the extracted latent features.

C. Latent Dirichlet Allocation

As mentioned before, another component of our model is LDA, through which we infer the distribution of the words in hidden topics. Using sampling method (collapsed Gibbs Sampling [37]), we iteratively maximize the likelihood of the joint probability distribution (See Fig. 1 and Eq. 1). After training according to the model represented in Fig. 1, θ is a $N \times K$ matrix where N is the number of news and K is the number of topics and each row present the proportion of each topic in the text. For example θ_{ik} represents the proportion that news i has from topic k ($i \in 1, \ldots, N$ and $k = 1, \ldots, K$). We can directly concatenate K-dimensional vector corresponding to each news, to the n_f-dimensional latent features as it is illustrated in Fig. 2.

D. Classifier

Another main component of our model is a classifier that receives all the features as input and outputs the news labels. We intend to use an NN-based classifier (MLP) as well as traditional machine learning methods such as SVM, logistic regression, Naïve Bayes, Random Forest, and KNN. Our motivation to include different types of classifiers such as discriminative and generative classifiers is to inspect the sensitivity of these classifiers to the obtained features. Moreover, different classifiers have different performance and discriminative abilities depending on the number and the distribution of the features.

E. Evaluation

In this project, we evaluate the performance of the model compared to the baselines, which are the results of the experiment on VAE and LDA feature sets separately. In other words, the objective is to show the concatenation of these two feature sets can improve the performance criteria or reveal some hidden patterns more clearly. The performance criteria that we have implemented in this project are as follows.

Accuracy is the ratio of total correct labels to the size of the dataset as presented in Eq. 8.

$$\text{Accuracy} = \frac{\text{TP + TN}}{\text{TP + TN + FP + FN}} \tag{8}$$

Precision is the fraction of actual fake news among all the fake detected news.

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \tag{9}$$

Recall (*a.k.a.*, sensitivity) is the ratio of news truly detected as fake to all the fake news in the data.

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}} \tag{10}$$

We calculate F-score/F-measure combines precision (Eq. 9) and recall (Eq. 10) and is actually the harmonic mean of them (Eq. 11).

$$\text{F-score} = \frac{\text{2TP}}{\text{2TP + FP + FN}} \tag{11}$$

We are also interested in computing False Positive Rate (FPR) and False Negative Rate (FNR). FPR is the ratio of the real news that is falsely classified as fake (Eq. 12).

$$\text{FPR} = \frac{\text{FP}}{\text{TN + FP}} \tag{12}$$

FNR reveals the ratio of news that are wrongly classified as real (Eq. 13). This measure is more important in this setting because the consequence of this case is more than the other way around.

$$\text{FNR} = \frac{\text{FN}}{\text{FN + TP}} \tag{13}$$

After obtaining the feature sets from VAE and LDA components, we consider three types of experiments.

1) Dimensionality Reduction: We apply PCA and tSNE dimensionality reduction techniques on feature sets obtained by VAE, LDA, and their concatenation on training and test set. The concatenation of feature sets would be matrices of $|\mathcal{D}_{tr}| \times (n_f + K)$ and $|\mathcal{D}_{te}| \times (n_f + K)$ for $\mathcal{D}_{tr}$ and $\mathcal{D}_{te}$. The motivation for applying dimensionality reduction is to illustrate the concatenation of two feature sets can improve the separation of different classes. This is particularly useful in unsupervised settings or if the labels of posts are not available.

2) Metrics on classification outcome: We compute desired performance metrics after classifying posts/news using the entire feature sets obtained by VAE, LDA, and their concatenation.

3) Metrics on classification outcome after feature selection: This experiment evaluates the performance of the model for base features and their concatenation, after applying classifiers on selected features only. Here, we select features by univariate filter methods. They rank the features according to their Chi^2 and Gini indices and select top m features, where $m = \{1, 5x | 1 \leq 5x \leq \min\{n_f, K\}\}$. Since the results do not vary much, we present the only outcome of Chi^2 feature selection.

IV. Dataset

This section gives information about the datasets that we processed in this project. The first data is Twitter dataset [38] which was originally collected for MediaEval Workshop 2016 and is publicly available[1]. This dataset includes around 17000 posts on Twitter and covers 17 events. It comes with separate training and test set with *fake/real* labels and contains information about the post such as the source of the post, textual and visual content of the post. Our motivation to use this dataset is twofold; it previously gained attention in papers including MVAE [3], besides it contains tweets from different events and we expect the LDA model will find distinguishable topics after training.

Fig. 3. WordCloud representation of Twitter dataset. Left and right subplots correspond to training and test sets respectively.

However, the challenge regarding this dataset is that the posts include both textual and visual content, and the text parts of the posts tend to be shorter than actual news. For this reason, we intend to use another dataset. ISOT fake news dataset [39], [40] contains more than 25000 labeled news. The authentic articles and fake news are collected from Reuters.com and unreliable websites (flagged by Politifact) respectively. It does not provide separate training and test set. and the majority of the data are political news collected from 2016 to 2017. The number of the real and fake article are both 12600. This dataset is cleaned but the punctuations and mistakes in the fake news are kept and are accessible on the University of Victoria website[2].

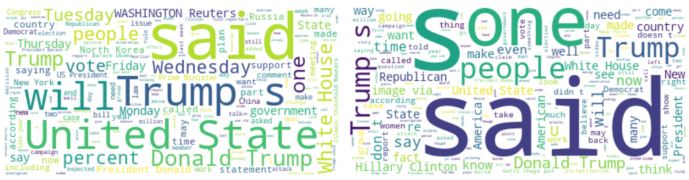

Fig. 4. WordCloud representation of ISOT dataset. Left and right subplots correspond to the text with real and fake labels respectively.

Figures 3 and 4 illustrate the WordCloud representation of the first and second datasets after removing words with non-English characters and punctuation marks.

We also applied LDA with the number of topics $= 10$ on these two datasets and plotted 20 most frequent words in each topic in Figures 5 and 6. As expected, the words of the topic in Twitter dataset seem to be more distinct rather than ISOT data.

V. Experimental Setting

In this section we provide some important technical details about the settings in the experiments. We implemented the model in Python 3.6 and run the experiments on a server with 1 CPU node (Xeon) and 10 GB of RAM. The code is available on Github[3] (still privately).

A. Notations

The main notations we are using in this section are the following:

- $\mathcal{D}$: dataset, $\mathcal{D}_{tr}$: training set, $\mathcal{D}_{te}$: test set
- N: Number of samples, indexed by i.
- V: The set of vocabulary detected by word2vec.
- n_f: Number of latent features obtained from encoder.
- w: word2vec dimension.
- $L = max\{l_i : i = 1, \ldots, N\}$
- l_i: Length of sample i (number of words).
- $t_i^{(j)}$: Word j in sample i.
- λ_1: Regularization parameter ($= 0.05$).
- λ_2: Regularization parameter ($= 0.3$).
- K: Number of topics.

B. Preprocessing

The first component of the system is preprocessing. In this part, we applied the relevant text-related processing suggested by [3]. In addition, We tokenize each post/news i to its words $t_i^{(1)}, \ldots, t_i^{(l_i)}$ in the lowercase and remove some of the punctuation marks and some words. We define l_i as the length of post i and:

$$L = \max\{l_i : i = 1, \ldots, N\}$$

[1]https://github.com/MKLab-ITI/image-verification-corpus

[2]https://www.uvic.ca/engineering/ece/isot/datasets/

[3]https://github.com/Marjan-Hosseini/Big-Data

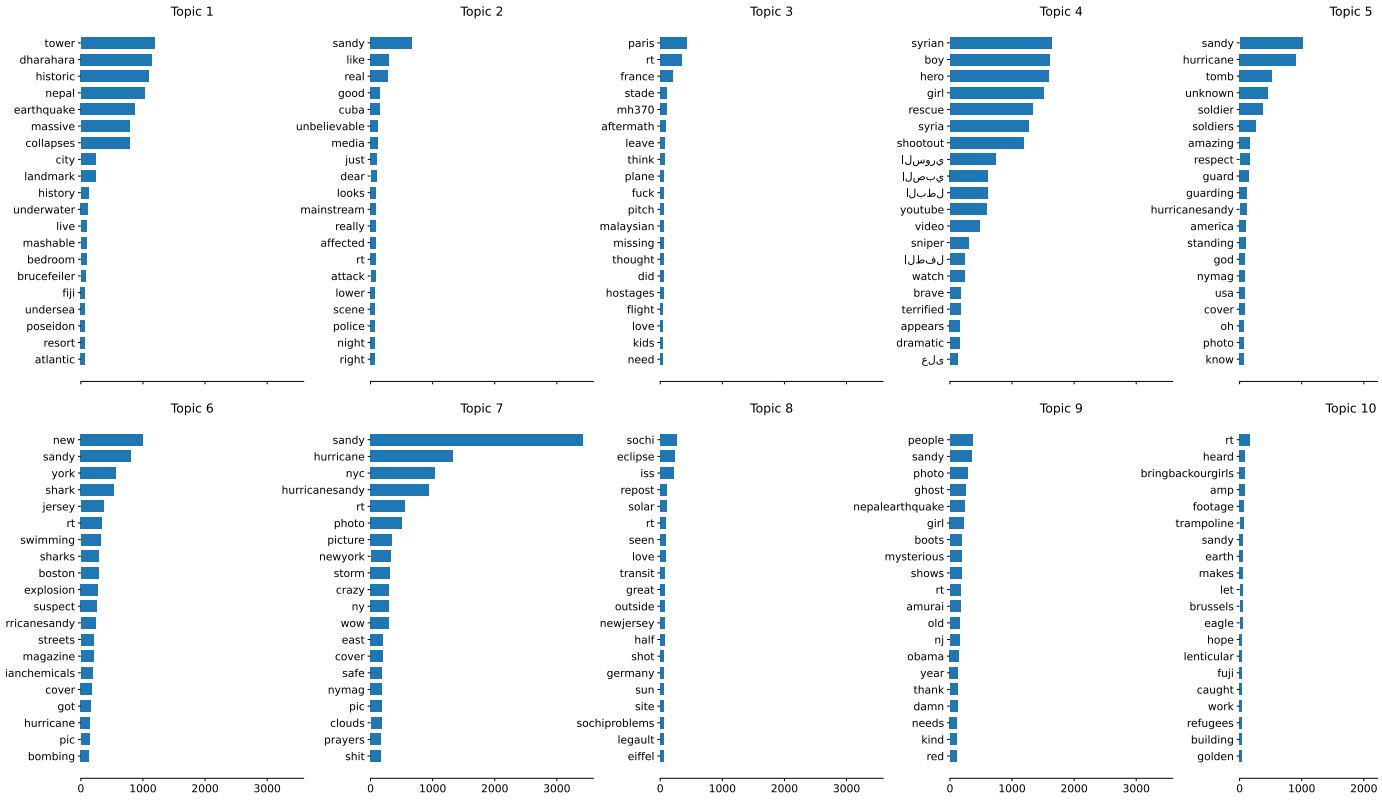

Fig. 5. Top 20 most frequent words in 10 topics detected by LDA model in Twitter dataset.

For ISOT $L = 45$ and in Twitter dataset $L = 28$. Then we filter out non-English posts using langdetect Python package. In the next step, we train word2vec model provided in gensim package (version 3.x or 4.x) on the dataset $\mathcal{D}$, and obtain a w-dimensional vector for each vocabulary $v \in V$ detected by word2vec. The result is a $|V| \times w$ embedding matrix, where $|V|$ is the size of vocabulary set in the model, varying depending on the value of w. Then we split ISOT dataset $\mathcal{D}$ into training and test sets ($\mathcal{D}_{tr}$ and $\mathcal{D}_{te}$), with 20% of the data as test and stratification of the news label. In this project, for different datasets, we have set $w \in \{8, 16, 32, 64, 160\}$. Twitter dataset has separated training and test sets.

TABLE I
THE SUMMARY OF THE ENCODER PART OF VAE

Layer	Output Shape	Param #	Other Setting
Input	[(None, L)]	0	
Embedding	(None, L, w)	$\|V\| \times w$	Non-trainable (word2vec weights)
Bi. LSTM	(None, L, $2n_f$)	$8n_f(w + n_f + 1)$	activation='tanh' $L_2 = \lambda_1$
Bi. LSTM	(None, $2n_f$)	$8n_f(3n_f + 1)$	activation='tanh' $L_2 = \lambda_1$
Dense	(None, n_f)	$n_f(2n_f + 1)$	activation='tanh' $L_2 = \lambda_1$
Dense (h)	(None, n_f)	$n_f(n_f + 1)$	activation='tanh' $L_2 = \lambda_1$
Sampling	(None, n_f)	0	$= \mu_h + \epsilon \exp(\sigma_h^2/2)$ $\epsilon \sim \mathcal{N}(0, 0.01)$

C. Variational Autoencoder

The input to autoencoder is the posts, each post is considered as a vector of L words $t_i^{(1)}, \ldots, t_i^{(l_i)}$. If $l_i \leq L$, we apply zero-padding to fix the size of the input. Then we train the autoencoder using only the training part of the data and minimize the overall objective function (see Eq. 7). The detail of the architectural configuration in the encoder, decoder and classifier part of this component are reported in Tables I, II and III respectively. We also computed the number of parameters being trained in each layer as a function of hyper-parameters. We implemented VAE in Python Keras package (version 2.4.0). Regularization hyper-parameters λ_1 and λ_2 are fixed to 0.05 and 0.3 as their default values suggested in [3]. Furthermore, to promote efficiency and prevent overfitting we include early stopping on the loss value while training. We set the maximum number of epochs $= 30$ and batch size $= 128$ (default value).

D. Latent Dirichlet Allocation

The LDA model is provided in the scikit learn Python package. Similar to word2vec model, LDA creates a set of vocabulary, but it assumes posts as a bag of words. We let the maximum number of the allowed vocabulary be greater than $|V|$ ($= 20000$), so that we prevent restriction for LDA. This setting is not considered a restriction for word2vec model

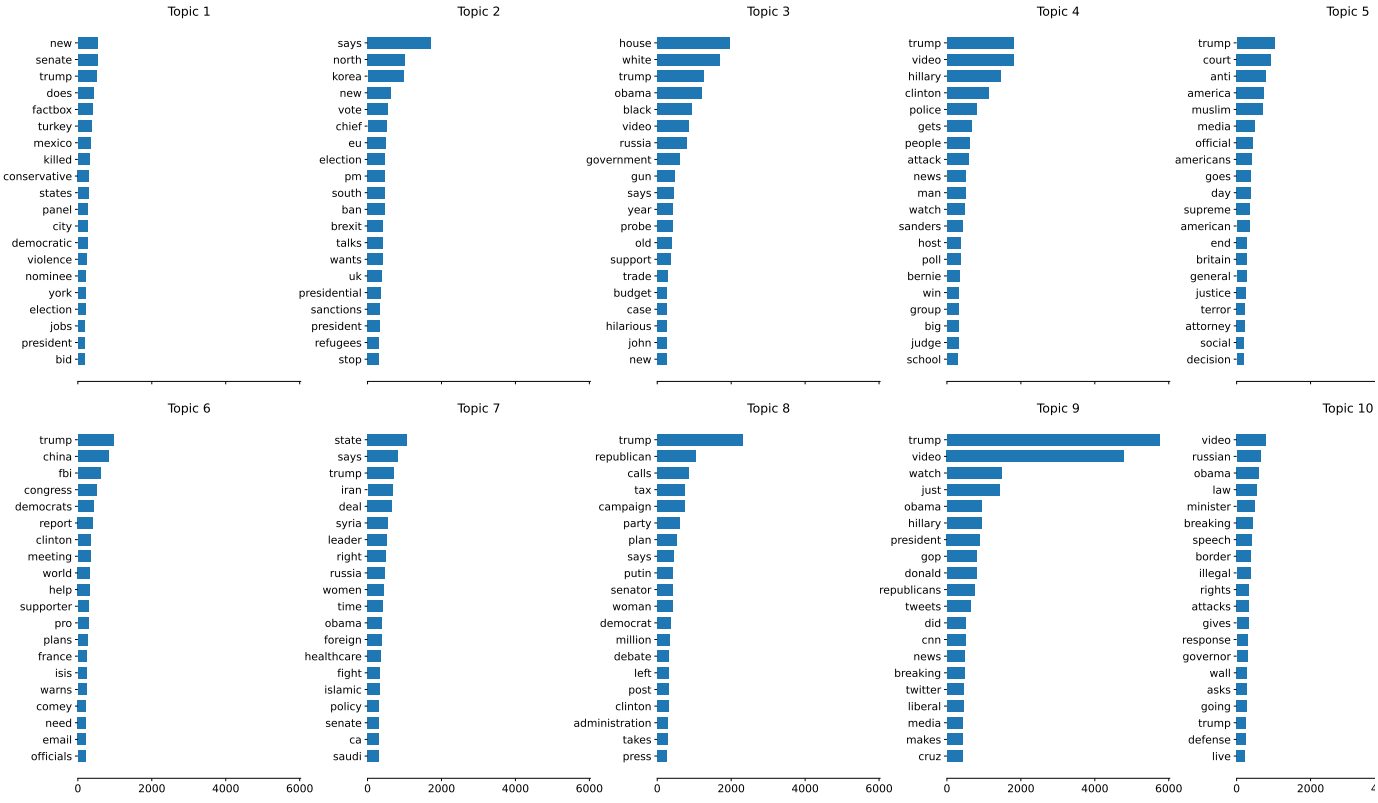

Fig. 6. Top 20 most frequent words in 10 topics detected by LDA model in ISOT dataset.

TABLE II
THE SUMMARY OF THE DECODER PART OF VAE

Layer	Output Shape	Param #	Other Setting
Input	[(None, n_f)]	0	
Dense	(None, n_f)	$n_f(n_f+1)$	activation='tanh' $L_2 = \lambda_1$
Repeat Vector	(None, L, n_f)	0	
LSTM	(None, L, n_f)	$4n_f(2n_f+1)$	activation='tanh' $L_2 = \lambda_1$
LSTM	(None, L, n_f)	$4n_f(2n_f+1)$	activation='tanh' $L_2 = \lambda_1$
Time Dist.	(None, L, $\|V\|$)	$\|V\|(n_f+1)$	activation='softmax'

TABLE III
VAE CLASSIFIER SUMMARY

Layer	Output Shape	Param #	Other Setting
Input	[(None, n_f)]	0	
Dense	(None, $2n_f$)	$2n_f(n_f+1)$	activation='tanh' $L_2 = \lambda_2$
Dense	(None, n_f)	$n_f(2n_f+1)$	activation='tanh' $L_2 = \lambda_2$
Output	(None, 1)	n_f+1	activation='sigmoid'

since the vocabulary set in word2vec is varying according to w parameter. Moreover, LDA is a parametric model and the number of topics (K) should be set in advance. We change $K \in \{8, 10, 16, 32, 64\}$. In addition, we chose the number of iterations in sampling $= 1000$.

E. Classifier

We have applied six widely used classifiers from scikit-learn Python package with the default setting. For MLP we set the number of iterations equal to 300. By default, Random Forest classifier has the number of estimators equal to 100. In SVM we use the linear kernel, and in KNN, we select k = 3. In logistic regression classifier, we use liblinear solver. Naïve Bayes has the default setting. Before applying all the classifiers, we normalize the features with z-Score normalization using StandardScaler in scikit-learn package.

F. Evaluation

We computed PCA and tSNE dimensionality reduction on features obtained by VAE, LDA, and their concatenation. In both methods, we selected the number of components $= 2$ for better visualization. In tSNE, we set perplexity $= 40$. For PCA, we normalize data with z-score normalization before transformation, to obtain eigenvalues from the features in the same scale. Then the classifiers are trained using the training set and the entire feature sets, and all the metrics provided in Sect. III-E are computed for $\mathcal{D}_{tr}$ and $\mathcal{D}_{te}$. In the next experiment, we first apply feature selection and after that, we repeat the classification.

VI. RESULTS

This section includes results and plots by running the code on both datasets. However, since Twitter data does not provide

labels for the test part, some of the results cannot be calculated for it. Accordingly, for a more meaningful performance evaluation of the model, most of the presented plots here are related to the ISOT dataset.

A. Convergence

After running the code, to make sure that the model parameters are converged, we provided the convergence of accuracy metric while training the model for two datasets.

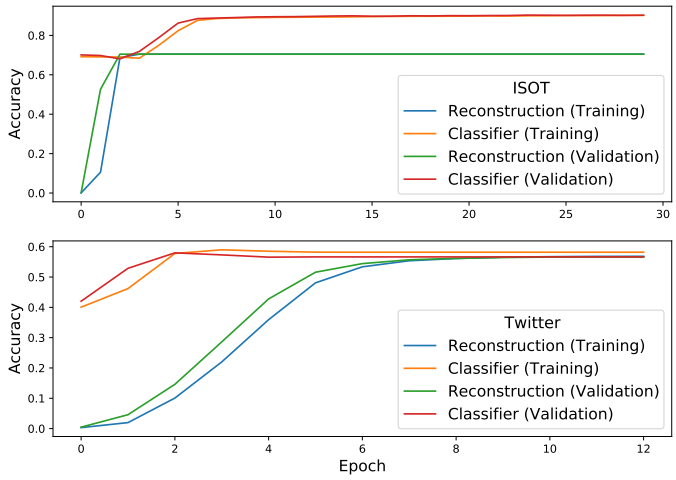

Fig. 7. Convergence Plot, the first plot shows the convergence of ISOT data, and the second is for Twitter dataset and in both plots $w = n_f = K = 32$.

Fig. 7 shows the accuracy metrics in the model history of ISOT and Twitter. In ISOT data, the reconstruction accuracy on training and validation set converged to a lower value than classifier accuracy.

B. Dimensionality Reduction

This section details the application of dimensionality reduction methods on obtained feature sets and their concatenation.

Fig. 8 shows the transformation of feature sets of ISOT data on 2 PCA dimensions. In this run we set $K = 10$ and $w = n_f = 32$. The feature sets obtained from VAE are drawn from a Gaussian, and the θ variables in LDA are Dirichlet distribution. The first two rows of this plot show the first two principal components obtained from these two distributions respectively. The combination of these two feature sets separates two classes more clearly compared to the first two rows.

Since PCA is a linear transformation and cannot capture the non-linear relationship between features, we provided the tSNE transformation for the same feature sets in Fig. 9 too.

C. Classification

In this experiment, we evaluate the classification results without performing any feature selection before it.

As it is depicted in Fig. 10, the performance metrics are slightly higher when two feature sets are concatenated. Here, as expected, LDA features do not show predictive results compared to VAE, since the setting for LDA is supervised,

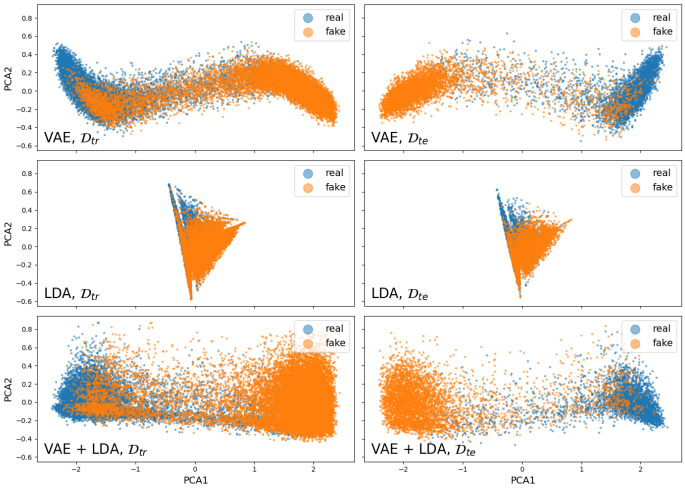

Fig. 8. PCA Plot for ISOT dataset, the first and second columns illustrate the application of PCA on training and test data. The first two rows correspond to the feature sets obtained by VAE and LDA separately, the third row shows the result on their concatenation. In all the plots $K = 10$ and $w = n_f = 32$.

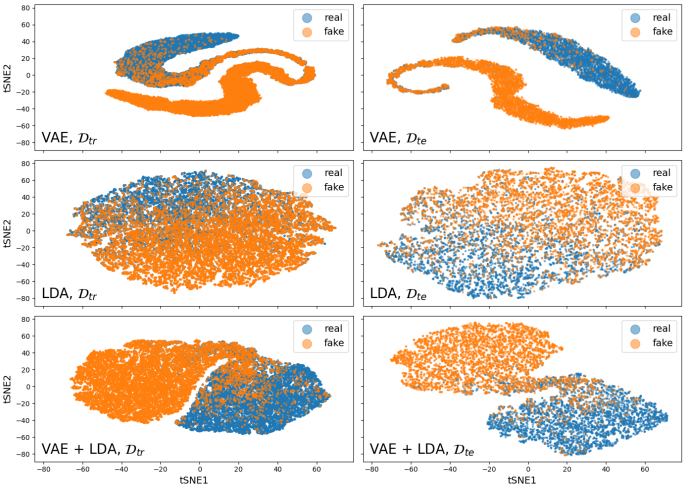

Fig. 9. tSNE Plot for ISOT dataset, the first and second columns illustrate the application of tSNE on training and test data. The first two rows correspond to the feature sets obtained by VAE and LDA separately, the third row shows the result on their concatenation. In all the plots $K = 10$ and $w = n_f = 32$.

while VAE is coupled with a classifier. However, including LDA features seems to improve the accuracy metrics in most of the classifiers.

Tab. IV reports the FPR and FNR values obtained after classifying different feature sets. In this experiment we set $K = w = n_f = 32$. This result suggests that by concatenating two feature sets, FNR and FPR are reduced in most of the classifiers. We have bold when the test results outperform previous results.

D. Feature Selection and Classification

In this experiment we first apply Chi^2 feature selection method and then classify the feature sets similar to the previous.

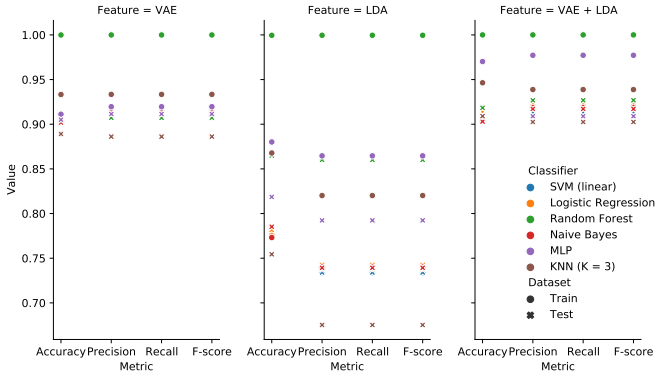

Fig. 10. Accuracy metrics after running the model on ISOT training and test dataset and classifying VAE, LDA feature set, and their concatenation. In this experiment $K = w = n_f = 32$.

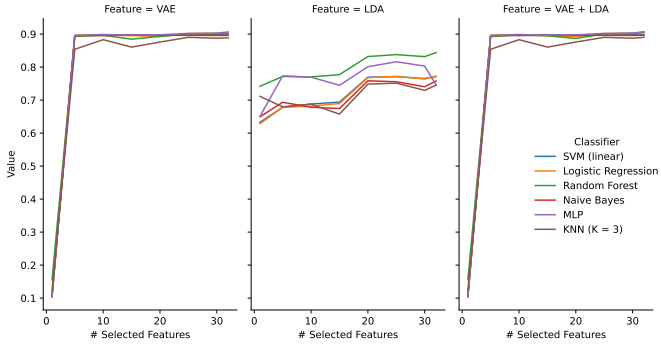

Fig. 11. Accuracy metrics after applying Chi^2 feature selection on ISOT training and test dataset and classifying VAE, LDA feature set and their concatenation. In this experiment $K = w = n_f = 32$.

TABLE IV
FPR AND FNR METRICS AFTER RUNNING THE MODEL ON ISOT TRAINING AND TEST DATASET AND CLASSIFYING VAE, LDA FEATURE SETS, AND THEIR CONCATENATION.

Classifier	Metric	VAE Train (Test)	LDA Train (Test)	VAE + LDA Train (Test)
SVM	FPR	0.1092 (0.1090)	0.1870 (0.1788)	0.0973 (**0.0984**)
(linear)	FNR	0.0842 (0.0876)	0.2661 (0.2659)	0.0822 (**0.0866**)
Logistic	FPR	0.0994 (0.0980)	0.1917 (0.1842)	0.0889 (**0.0896**)
Regression	FNR	0.0905 (0.0896)	0.2615 (0.2575)	0.0793 (**0.0824**)
Random	FPR	0.0001 (0.0963)	0.0005 (0.1308)	0.0000 (**0.0890**)
Forest	FNR	0.0001 (0.0928)	0.0004 (0.1399)	0.0000 (**0.0732**)
Naïve	FPR	0.1110 (0.1092)	0.1981 (0.1749)	0.1093 (0.1092)
Bayes	FNR	0.0821 (0.0853)	0.2600 (0.2607)	0.0812 (**0.0829**)
MLP	FPR	0.0961 (0.1006)	0.1064 (0.1586)	0.0359 (**0.0900**)
	FNR	0.0804 (0.0886)	0.1354 (0.2076)	0.0228 (0.0910)
KNN	FPR	0.0668 (0.1084)	0.0909 (0.1769)	0.0470 (**0.0853**)
(K = 3)	FNR	0.0666 (0.1139)	0.1798 (0.3247)	0.0612 (**0.0975**)

In Fig. 11 we do not see much improvement over the VAE results. This could be as a result of applying univariate feature selection over multivariate. In univariate feature selection, the importance of features are assessed individually towards the class label and possible correlations among features are ignored. Plus some features might not show discriminative property by themselves, however, if they are used with other features, the overall performance could be potentially improved. In this regard, a possible future work could be using multivariate feature selection techniques or other ranking indices.

VII. Discussion

In this section, some advantages and disadvantages of this work are discussed.

A. Advantages

One of the most important contributions of this work is increasing interpretability to the fake news detection model, which we achieved By adding LDA to our model. Moreover, even if we use only the textual part of the news, our model improves performance metrics criteria together with most of the classifiers. Another important result of this project is making a framework for clustering fake and real news in an unsupervised setting. We have shown that the concatenation of obtained features can separate these two classes more clearly, so we expect clustering methods can be applied more efficiently.

B. Disadvantages

The main disadvantage of our model is its time and computational complexity. Depending on the dataset, word2vec dimensions, and the number of iterations in LDA, each run of the algorithm takes roughly between three to 15 hours on a server with 10 GB of RAM. Only one epoch of VAE takes around 250 and 1200 seconds in Twitter and ISOT datasets respectively (without considering multiprocessing and GPU). Apart from mentioned variables, we empirically noticed that the run-time depends on variable L too. There are possible ways to overcome this issue, such as efficient use of GPUs while training VAE or changing batch sizes, or any other modification which leads to more efficient learning. We also noticed that results might change in different runs, which is a result of the probabilistic approach in VAE (sampling part) and LDA. Lastly, similar to many NLP models, the result is very dependent on text preprocessing.

VIII. Future Work

One of the possible ways to improve the results is adopting more efficient text preprocessing. In this project, we followed text processing by MVAE [3]. However, their approach might impose some limitations. For example, they do not use any stemming or lemmatization in the preprocessing. Another improvement can be achieved by applying more advanced text transformation methods such as BERT [41] instead of the word2vec model. Bert was introduced in 2018 by J. Devlin *et al.* It uses transformers in the architecture with attention mechanism described in a well-known paper entitled "Attention is all you need" [42] by Google in 2017. In addition, there are numerous variables in the model that can be tuned, such as regularization parameters λ_1 and λ_2 in the VAE layers and the number of vocabularies being used by LDA. We can apply supervised LDA [43] to obtain more predictive features from the LDA component. Regarding VAE, component modification

in the structure of the layers or objective functions can improve the overall performance, and in general, framework adopting multivariate feature selection methods can potentially select more discriminative features and increase the accuracy metrics.

IX. Conclusion

Here, based on the idea presented in [3], we implemented a VAE coupled with a classifier for feature extraction. We proposed to add LDA to the model, which is the Bayesian statistical method for topic modeling. Our motivation for adding the LDA component is to increase the interpretability of the model. Moreover, we evaluated our results by classification on individual feature sets as baselines and their concatenation and demonstrated that our model successfully improves the accuracy metrics in most of the classifiers. In another experiment, we performed linear and non-linear dimensionality reduction methods on the obtained features and showed that our model could facilitate clustering in unsupervised settings.

X. Responsible Content and Project Concluding Remarks

Concerning teamwork, we created a Github repository[4] from the beginning of the project to be synched more easily. We also had weekly meetings and have been constantly in touch up to now.

In this report, the introduction and background sections are written jointly by two authors. Related work is done by the first author and the other parts by the second author. In terms of individual tasks in the coding part, the second author finished implementing the preprocessing, VAE, and LDA components and visualizations. The first author implemented classification methods, normalization, and accuracy metrics. feature selection has been done jointly.

In general, throughout this course/project, we learned about important machine learning and data mining concepts. More specifically, in this project, we valued the importance of preprocessing in obtaining satisfactory results. In addition, we learned about deep learning and NLP techniques more in-depth, and we could successfully complete a project in these topics. Lastly, we learned various state-of-the-art methods and ideas during the literature review stage of the project, which can help us continue this work.

References

[1] Jeffrey Gottfried and Elisa Shearer. News use across social media platforms 2016. 2016.

[2] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*, 2017.

[3] Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, and Vasudeva Varma. Mvae: Multimodal variational autoencoder for fake news detection. In *The World Wide Web Conference*, pages 2915–2921, 2019.

[4] Wissam Antoun, Fady Baly, Rim Achour, Amir Hussein, and Hazem Hajj. State of the art models for fake news detection tasks. In *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pages 519–524. IEEE, 2020.

[4]https://github.com/Marjan-Hosseini/Big-Data

[5] Oluwaseun Ajao, Deepayan Bhowmik, and Shahrzad Zargari. Sentiment aware fake news detection on online social networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2507–2511. IEEE, 2019.

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

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[8] Ruslan Salakhutdinov and Geoffrey Hinton. Semantic hashing. *RBM*, 500(3):500, 2007.

[9] Ruslan Salakhutdinov and Geoffrey Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978, 2009.

[10] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

[11] Shivangi Singhal, Rajiv Ratn Shah, Tanmoy Chakraborty, Ponnurangam Kumaraguru, and Shin'ichi Satoh. Spotfake: A multi-modal framework for fake news detection. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, pages 39–47. IEEE, 2019.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013.

[15] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[16] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. Distributed feature selection: An application to microarray data classification. *Applied soft computing*, 30:136–150, 2015.

[17] Zena M Hira and Duncan F Gillies. A review of feature selection and feature extraction methods applied on microarray data. *Advances in bioinformatics*, 2015, 2015.

[18] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[19] Włodzisław Duch, Tomasz Winiarski, Jacek Biesiada, and Adam Kachel. Feature selection and ranking filters. In *International conference on artificial neural networks (ICANN) and International conference on neural information processing (ICONIP)*, volume 251, page 254, 2003.

[20] Alan Jović, Karla Brkić, and Nikola Bogunović. A review of feature selection methods with applications. In *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 1200–1205. Ieee, 2015.

[21] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of machine learning research*, 5(Oct):1205–1224, 2004.

[22] Huan Liu and Rudy Setiono. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, pages 388–391. IEEE, 1995.

[23] Robert Nisbet, John Elder, and Gary Miner. *Handbook of statistical analysis and data mining applications*. Academic Press, 2009.

[24] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[25] Ramy Baly, Georgi Karadzhov, Dimitar Alexandrov, James Glass, and Preslav Nakov. Predicting factuality of reporting and bias of news media sources. *arXiv preprint arXiv:1810.01765*, 2018.

[26] Sadia Afroz, Michael Brennan, and Rachel Greenstadt. Detecting hoaxes, frauds, and deception in writing style online. In *2012 IEEE Symposium on Security and Privacy*, pages 461–475. IEEE, 2012.

[27] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proc. of the 2017 conference on empirical methods in natural language processing*, pages 2931–2937, 2017.

[28] Victoria L Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the second workshop on computational approaches to deception detection*, pages 7–17, 2016.

[29] Jun Ito, Jing Song, Hiroyuki Toda, Yoshimasa Koike, and Satoshi Oyama. Assessment of tweet credibility with lda features. In *Proceedings of the 24th WWW*, pages 953–958, 2015.

[30] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. 2016.

[31] Alessandro Bondielli and Francesco Marcelloni. A survey on fake news and rumour detection techniques. *Information Sciences*, 497:38–55, 2019.

[32] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36, 2017.

[33] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR, 2013.

[34] Kuai Xu, Feng Wang, Haiyan Wang, and Bo Yang. Detecting fake news over online social media via domain reputations and content understanding. *Tsinghua Science and Technology*, 25(1):20–27, 2019.

[35] Saad Sadiq, Nicolas Wagner, Mei-Ling Shyu, and Daniel Feaster. High dimensional latent space variational autoencoders for fake news detection. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 437–442. IEEE, 2019.

[36] Vivek K Singh, Isha Ghosh, and Darshan Sonagara. Detecting fake news stories via multimodal analysis. *JASIST*, 72(1):3–17, 2021.

[37] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577, 2008.

[38] Detection and visualization of misleading content on Twitter. Boididou, christina and papadopoulos, symeon and zampoglou, markos and apostolidis, lazaros and papadopoulou, olga and kompatsiaris, yiannis. *IJMIR*, 7(1):71–86, 2018.

[39] Hadeer Ahmed, Issa Traore, and Sherif Saad. Detecting opinion spams and fake news using text classification. *Security and Privacy*, 1(1):e9, 2018.

[40] Hadeer Ahmed, Issa Traore, and Sherif Saad. Detection of online fake news using n-gram analysis and machine learning techniques. In *International conference on intelligent, secure, and dependable systems in distributed and cloud environments*, pages 127–138. Springer, 2017.

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[43] David M Blei and Jon D McAuliffe. Supervised topic models. *arXiv preprint arXiv:1003.0783*, 2010.