Profiling Fake News: Learning the Semantics and Characterisation of Misinformation

Swati Agarwal[0000−0001−9586−2794] and Adithya Samavedhi

BITS Pilani, Goa Campus, India
{swatia, f20170071}@goa.bits-pilani.ac.in

Abstract. Research shows that much recent information spreading on social media is dubious and untrue, intended to mislead audiences. In the last few years, the fake news spreading community has emerged on a large-scale, which intentionally promotes incorrect information, becoming challenging to detect even with human annotation. This paper aims to profile fake news and investigate the characteristics that make the news untrue and fake. We validate our features set on four open-source datasets popularly used in the community. To evaluate the proposed features set's performance and importance, we employ various machine learning, artificial, and recurrent neural network models. Our results show that proposed features are generalized and not specific to a domain. Our findings reveal that the model outperforms on raw content-based datasets (ISOT and Fake News Master) than interpretation-based datasets (Liar-Liar and Kaggle Emergent).

Keywords: Misinformation · Features Extraction · Supervised Learning · Context Analysis · Applied Computational Linguistics

1 Introduction

The world wide web (WWW) comes with a plethora of features that facilitate real-time information dissemination to a wide range of audiences worldwide. Due to the simplicity of navigation, low-cost publication, and wide reachability, online social media (OSM) has become a powerful platform for social interactions and information diffusion [26,27]. Recently, WWW specially OSM have become an ideal platform for spreading fake news compared to mainstream or print media [5,12]. Nowadays, phoney articles also tend to be obtrusive and diverse, including fake reviews, advertisements, misleading and targeted content, rumour, satire, political sentiments, counterfeit opinions. [33] describes fake news as *all kinds of false stories or information that are mainly published and distributed on the Internet, to purposely mislead, befool or lure readers for financial, political or other gains.* The issue of fake news gained researchers and practitioners' attention after 2016 U.S.A. presidential elections being accused of misleading political polarisation, conflicts, and influencing voters by making false claims [4,7,20,24].

Research shows that over the past few years, fake news has been sprawling over the Internet, further leading to a harmful and overpowering impact on individuals, internet communities, and society [9,21]. Due to the exponential

growth in fake news content, various online fact-checking systems and organisations identify and verify fake news. Majority of these systems (FactCheck, Hoax-Slayer, PolitiFact, Snopes, and TruthOrFiction) use manual detection approaches by experts and fact-checkers while Haoxy uses other fact-checker websites for verification. Furthermore, these systems are specific to political news and not applicable to a wide range of fake news [33]. The manual identification of fake news is overwhelmingly impractical due to large volume, real-time velocity, and uncertainty in fake news topics, eventually delaying the annotation process [2]. Due to online media's dynamic and heterogeneous nature, detecting credible news and mitigating misinformation is a technically challenging problem. Classify and Factmata systems use artificial intelligence (AI)-based approaches for identifying misinformation posts. However, the lack of high-quality data on fake news is an additional challenge for developing automated models [25]. [33] describes fake news to be comprised of physical and non-physical content. Physical content includes the title, body, and multimedia content, whereas non-physical attributes include the topic, sentiment, and aim that define the narrative's intention. Thus, a piece of fake news can be identified using linguistic, contextual, knowledge-based, and stylometric-based attributes [34]. References [2], [3] and [10] show that n-gram and bag-of-words are the most commonly used models for fake news detection. Recently, with the advancement of deep learning in natural language processing (NLP), deep syntax analysis, word2vec, LSTM (long short-term memory) neural network, sequence-2-sequence based deep neural network architectures are heavily employed for fake news classification [23,30]. [10] and [13] describe that semantic level attributes of documents can reveal the writing patterns and creators' intent and background to validate the credibility of the news. [26] discovers that while AI and knowledge-base analysis approaches are useful for fake news detection, they still require human intervention to validate a large size of inputs.

Due to large-scale vocabulary, diverse writing patterns and purposes, maintaining high classification accuracy is challenging for AI-based models as they need to depend on human knowledge. Furthermore, these models require a high-quality annotated dataset for learning, one of the primary technical challenges in the fake news detection research community. Additionally, existing datasets are incomplete, unstructured, and noisy posing challenges in automatic detector tools or applications. In-terms of non-physical content, real news depict emotions and opinion towards certain object or content whereas, fake news are written to deceive people and obfuscate intentionally [1]. Physical content analysis may utilise N-grams, external URLs, hashtags, @username mentions, and e-shouting features [2,15]. Similarly, the non-physical content analysis may include grammar, typographical errors, word and language pattern, focused topic (social, political) as important characteristics [1,6]. Recent studies analyse the content creators' network to capture anomalous information using social context analysis. Graph analysis based methods such as the user network analysis and distribution pattern analysis allow researchers to identify like-minded people [23] and capture the epidemic of misinformation spread across social networking websites. Among various different relations on social networking portals, *like*, *share*, *flag*,

and *comment* are the most commonly used features in identifying the credibility of the user [16]. Distribution pattern analysis of interactions between users reveals the insights on the exposure of misinformation spread and reciprocity of users [34]. [14] proposed vector-based language models which use a deep, bidirectional LSTM architecture to identify fake news from genuine ones. Due to the heterogeneous and dynamic nature of interactions in social networking websites, finding suspicious or anomaly patterns in the news is a technically challenging problem [22]. On a different note, [29] proposes to use a multimodal approach combining text and visual content for fake news classification. They also reveal the visual features that are discriminatory between fake and real news articles.

Prior literature reveals that despite the existing models and systems for fake news identification, the problem persists. The linguistic, interaction or contextual metadata-based features are not sufficient for the classification since they are dependent on the platform. For example, text length, chat slangs, hashtags, likes, reblogs, shares, comments, flags/reports, and so on. In this paper, we aim to profile fake news to investigate their underlying patterns and identify features that are strong indicators of misinformation independent of the carrier platform. We aim to employ these features across various classification models and evaluate their performance to check the validity of proposed features.

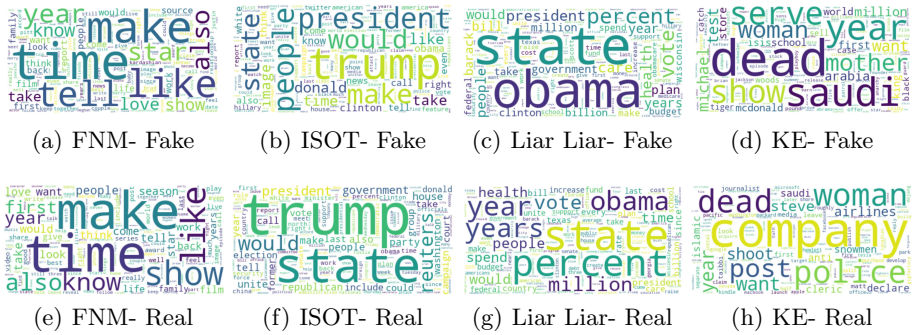
2 Experimental Dataset

We conduct our experiments on various public datasets downloaded from different sources (existing literature, data challenges, and public repositories). We select these datasets as they are popularly used in the research community [24] and hence used for comparing and benchmarking our results. We downloaded four open-source datasets; Liar-Liar, Kaggle Emergent, ISOT, and Fake News Master. Unlike many other popular datasets, which are originally labelled as a rumour or satire, these datasets are labelled as Fake, Real, and Unverified. Liar-Liar dataset is a benchmarking dataset on fake news published by [32]. It contains a total of $11,552$ short statements ($10,269$ training and 1283 testing) manually labelled into six categories: barely true, false, half true, mostly true, true, and pants on fire. The labeling was performed by a professional editor from PolitiFact. ISOT fake news dataset was published by Information Security and Object Technology Lab at University of Victoria [2,3]. The dataset contains links to $21,417$ real news collected from Reuters News Agency and $23,481$ fake news flagged by PolitiFact and Wikipedia. Similarly, Fake News Master (FNM) dataset contains four dataset files consisting of hyperlinks to real and fake articles from Gossip Cop and PolitiFact records [25,26]. Gossip Cop data files contain links to 5324 fake and $16,817$ real articles, whereas PolitiFact data files contain 434 fake and 655 real articles' links. Due to data size and labelling consistency, we merge Gossip-Cop and PolitiFact data and referred it to as Fake News Master (FNM) dataset. For both ISOT and FNM, we used *Newspaper3k* API to fetch the article content from URLs. Kaggle Emergent (KE) dataset was downloaded from Kaggle website. KE is a collection of all web pages cited in Emergent.info, an online real-time rumour tracker. This dataset contains 2146

claims reported on different news media sources, and the tracker identifies them as false, true, or unverified.

2.1 Data Pre-processing

We collect our experimental datasets from heterogeneous sources; therefore, we perform pre-processing to remove any possible biases from the datasets. We first remove all duplicate records from the dataset to avoid overfitting in our model. After removing duplicates in the Liar-Liar dataset, we had 8178 training and 2045 testing records. Similarly, in both ISOT and FNM datasets, we identified several duplicates along with the URLs that were deleted and returned empty news article during data collection. In ISOT, we had 38,514 unique articles, whereas the final FNM dataset consisted of 833 fake and 2754 real news articles. As discussed above, Liar-Liar dataset contains six classes identified in manual annotation by experts. We merge these classes into two groups as fake (barely true, false, and pants on fire) and real (half true, mostly true, and true) based on their intensity towards the truthness of news to reduce the class imbalance for minority classes. During pre-processing, we observe a major limitation in the Kaggle Emergent dataset. Out of 2146 records, only 107 were unique, while the rest were duplicate having the same link and the content. Furthermore, both KE and Liar-Liar datasets contain claims reported online about fake and real news. Therefore, these datasets do not include the original news/article but their interpretation reported by different users. We further removed the instances with blank text (due to formatting or unavailability of the record). We used English as a base language for our experiments. Therefore, as a part of additional pre-processing, all non-English texts (partial or complete text) were translated into the English language using Yandex Translation API.



(a) FNM- Fake (b) ISOT- Fake (c) Liar Liar- Fake (d) KE- Fake

(e) FNM- Real (f) ISOT- Real (g) Liar Liar- Real (h) KE- Real

Fig. 1. Word Cloud of Terms Present in Fake and Real News in Each Dataset

We removed all English stopwords from the text and converted the sentences into lowercase to ensure uniformity and reduce the sparsity in term-document matrix. Figure 1 shows the word clouds of the terms present in fake (top) and real (bottom) news in each dataset. These word clouds reveal that there are several overlapping terms not only across different datasets but fake and real news within a dataset as well. The overlapping words and similar frequency poses challenges in learning the underlying patterns, extracting distinct features, or relying on

Table 1. Summary of the Experimental Datasets. Dup = Duplicate, Size= Number of Original Instances, Final = Number of Instances After Pre-processing.

Source	Data	Public	Type	Dup	URL	Size	Final
Kaggle	Kaggle Emergent (KE)	✓	Claim	✓	✓	2146	107
SotA	Fake News Master (FNM)	✓	Article	✓	✗	23230	3587
SotA	Liar-Liar (Liar)	✓	Claim	✓	✓	11552	10223
SotA	ISOT	✓	Article	✓	✓	44898	38514

keyword-based lookup approaches. Table 1 shows a summary of our experimental datasets. As illustrated in Table 1, we downloaded Kaggle Emergent (KE) data from Kaggle, whereas Liar-Liar (Liar), Fake News Master (FNM) and ISOT datasets are acquired from the state-of-the-art (SotA) literature. All datasets are public and available for download without any restrictions. KE and Liar datasets contain claims, while FNM and ISOT datasets have original articles of fake and real news. Table 1 reveals that all datasets have duplicate records. Except for FNM, all datasets have external URLs present in the dataset. As discussed above, Table 1 also illustrates the total size of the original and final dataset after pre-processing.

3 Proposed Solution Approach

In this section, we discuss the proposed methodology for the automated classification of fake news. We identified specific underlying characteristics of misinformation, which are crucial to differentiate them from genuine and authentic articles. The aim is to identify features independent of a data corpus and are reliable indicators of fake news. We use several natural language processing (NLP) based models to extract these features and validate them on different datasets by employing machine learning and deep learning models. The extracted features are not limited by the training data, unlike Bag-of-words models. The aim is to use features that extract linguistic and semantic attributes from the text. For example, out-of-context terms make use of proximity information of token sequences in the text. In contrast, unigram, bi-gram term frequencies make use of the linguistic metadata of the document. The frequent occurrences of particular n-grams in an instance (input article) can also indicate or characterise fake news. We also aim to look for both prevalent and unusual patterns in a news article but varying in real and fake news to identify low-level features for classification.

3.1 Features Extraction and Selection

While existing studies give high priority to the contextual metadata, we extract features based on fake news's linguistic and semantic characteristics. For example, the number of likes, shares, and reports/flags are subjective and vary based on the platform. Instead the proposed features are based on the content and context of the news and not only their contextual metadata. Table 2 shows the list of all features. Based on their nature, the proposed features are grouped into eight broad categories; i.e., F1: statistical, F2: syntactic, F3: semantic, F4: temporal (time instance), F5: writing cues, F6: emotion, F7: topical and context, and finally F8: named entities. The pre-processed text is void of stop-words

and case insensitive. While pre-processing, we store the number of stop-words in every instance; this makes up the number of Stop-words features. We use the sklearn CountVectorizer module to extract n-gram term frequencies, i.e. uni and bigrams. While extracting term frequencies, we stored the average normalised scores of every instance (news post) using $\hat{X} = \frac{1+\log_{10} X_i}{1+\log_{10} \sum_{i \in I}(X_i)}$ where X_i represent the term frequency of i^{th} n-gram. The unigram and bi-gram term frequency feature indicates the number of unigram and bi-grams in an instance which have a normalised score greater than a threshold (median of the normalised scores of every input sample). This feature captures the unigrams and bi-grams that appear more often than usual. We use LIWC (Linguistic Inquiry Word Count) API to identify features that reveal that pattern of the writing style of fake news publishers or source. These features primarily include syntactic properties of text, writing cues (mention of social, cognitive, and perceptual attributes), emotional range, and time instance (past, present, and future) mention [19]. These features are essential to explore and investigate as they further reveal the intention and similarities in the agenda of fake news [18]. We also identify the pragmatic features of the text in faux and genuine articles. The out-of-context terms feature captures the number of semantically unfitted token pairs in the text frequently present in fake news targeting clickbait, misinformation spread, manipulation, etc. We use WordNet as a knowledge graph to compute the distance (path length) between every token (non-stop word) pair in the input text. In our experimental dataset D, given three terms t_1, t_2, and t_3, if $d(t_1, t_2) \ggg d(t_2, t_3)$ then $\{t_1, t_2\}$ are less likely to semantically occur together than $\{t_2, t_3\}$. We compute the average path length l_D of all unique pairs in the experimental dataset and use l_D as a threshold value to identify out-of-context terms in each text instance. A token t_i with path length $d(t_i, t_j) > l_D \mid t_i, t_j \in D, t_j \neq t_i$ is marked as out-of-context term. We tokenise the clean text at two granularity levels: sentence and token. We compare the token pairs' path similarity in the same sentence using the wordnet module from the NLTK library. We use the median of average path similarities of all instances as the threshold. The word pairs in an instance with lesser path similarity than the threshold are considered to be out-of-context. [15] reveals that fake news consists of more focus and mention of past tense compared to the present tense. They significantly use past news to support their claims. Therefore, we extract the additional feature "temporal" as a strong indicator that identifies past, present, and timestamps mention from the input text. Existing studies reveal that fake news may target an individual, group of people, and organisations [28]; therefore, inspired by these studies, we extract named entities present in the text posts. We use Spacy library to extract `person`, `organization`, `time`, and `geo political entity`.

[8] and [31] found that in addition to misinformation dissemination, fake news is also posted for target audiences and hence written focusing on a specific topic could be sensitive or trending. Therefore, we perform topic modelling on our experimental dataset to identify each news's unique topics. We use TextRazor API to extract the list of categories and topics along with their confidence score in each topic. The median of the average confidence scores of categories and topics

Table 2. The List of Features Set used for Classification

Code	Type	Feature	Description
F1	Statistical	F1.1 Unigram TF	Term Frequency of unigram
		F1.2 Bi-gram TF	Term Frequency of bigram
		F1.3 # Stopwords	#Common Words
		F1.4 Sixltr	words >= six letters
F2	Syntactic	F2.1 Conjunction	syntax and structure
		F2.2 Interrogation	
		F2.3 Number	
F3	Semantic	F3.1 out-of-context	keywords in context (KWICs)
F4	Temporal	F4.1 focuspast	time-based features
		F4.2 focuspresent	
		F4.3 timestamp	
F5	Writing Cues	F5.1 social	societal mention
		F5.2 certain	cognitive
		F5.3 perceptual	see, hear, feel
F6	Emotion	F6.1 negative emotion	psychological
F7	Topical and Context	F7.1 arts and entertainment	topics and taxonomy
		F7.2 conspiracy	
		F7.3 business and industrial	
		F7.4 economy and finance	
		F7.5 law, govt & politics	
		F7.6 science	
		F7.7 religion and belief	
		F7.8 education	
		F7.9 IT and technology	
F8	Named Entities	F8.1 PERSON	an individual or group
		F8.2 ORG	organization
		F8.3 GPE	geo political entity
		F8.4 TIME	time as an entity

are selected as a threshold. Finally, the topics and categories with confidence scores greater than the threshold are selected; this procedure reduces the features set's dimensionality. The category tagging feature of TextRazor provides fixed standardised taxonomies of categories, whereas the topic tagging feature identifies topics at different levels of abstraction. The primary source for topic modelling is from the TextRazor categories. In the absence of categories for a given small size instance, we select topics identified by TextRazor. We group these categories and topics into generic groups based on the taxonomy defined by IBM Watson Natural Language Classifier Taxonomy. F7 feature set in Table 2 shows the nine classes of taxonomies. The generalisation of these topics or categories curbs the dimensionality of the feature set. The category and topic confidence scores are then placed into respective taxonomy, which will be used for topic modelling. Table 2 shows the list of topics and categories grouped into high-level features. [11] reveals that keywords in context (KWICs) are important features of linguistics since they reveal the structure and language of the text. We use GrammarBot API to identify the grammatical errors in the posts.

The identification of grammatical errors further compliments the out-of-context terms in the news.

3.2 Classification Models

The steps discussed in Section 2.1 reveal that the removal of duplicate and empty records affects the class imbalance in the training and testing dataset. Therefore, we merge the original training and testing dataset respective to each source. To avoid any biases and class imbalance, we applied stratified sampling [17] on each dataset and divided the dataset into distinct groups called strata/stratum. The groups are equal to the number of classes adequately represented within the whole dataset, making it mutually exclusive and collectively exhaustive. Later, a simple random sample (or probability sample) is drawn from each stratum to create a test dataset with balanced classes. All datasets were split into 20% testing and 80% training and standardised using the StandardScalar module from sklearn.preprocessing. We employ several classification models that cover a wide range of simple machine learning (ML) algorithms, artificial neural networks (ANNs), and context-based deep learning algorithms to validate our features.

Machine Learning Models: We use a variety of ML models that cover a wide range of probabilistic, tree-based, ensemble, and kernel-based models suitable for small and large size of training datasets. We employ Gaussian Naive Bayes (GNB), Random Forest (RF), Decision Tree (DT), K-Nearest Neighbour (KNN), XGBoost (XGB), and Support Vector Machine (SVM) algorithms on each dataset. The judgement metric was the K-cross validation score with a 10 fold setting number on each of these classifiers. We also apply a Grid search to these classifiers for hyper-parameter tuning.

Artificial Neural Network Models: Recent advancements in NLP has shown that deep learning-based approaches give outstanding performances for binary classification. Therefore, we also employ an ANN models to validate the proposed features set for fake news classification. For all datasets, we use a neural network with a total of five layers where the first layer is the input layer of features acquired in Section 3.1. We use three hidden layers with 32, 32, and 8 neurons on the first, second, and third hidden layer, respectively. The number of neurons on the output layer depends on the type of labels available in our dataset. For KE data, the output layer has three neurons with a softmax activation layer to generate class probabilities. For FNM, Liar, and ISOT datasets, the output layer has only 1 neuron and a sigmoid activation layer. We use binary cross-entropy loss to report the loss in classification in all datasets except for the KE dataset, which is tested over categorical cross-entropy loss (multi-class dataset). We designed four different ANN architectures and experimented across all four datasets. The ANNs have been tested on the numerical features extracted from the text in contrast to the recurrent neural networks (RNN), which work solely with the text. We designed models varying with 2 to 3 hidden layers and $TanH$ and rectified linear unit ($ReLU$) activation functions. We also experimented with combinations of Adam and stochastic gradient descent (SGD) optimisers. The ANN takes the numerical features extracted as input and passes them through hidden layers with activation and dropout layers between

and produces the output. The dropout layers have been added to give equal importance to all neurons while training. Table 3 shows the detailed configuration of ANN models used for the experiments.

Table 3. ANN Models Variants. Learning rate= 0.005, Dropout Probability= 0.4

Model	**ANN-1**	**ANN-2**	**ANN-3**	**ANN-4**
Hidden Layers	(32: 32:8)	(32:8)	(32: 32:8)	(32:8)
Activation	ReLU	TanH	ReLU	TanH
Optimiser	Adam	Adam	SGD	SGD

Recurrent Neural Network Models: RNNs are widely used neural network architectures in NLP and set a performance benchmark in the areas where the sequence has more value than the individual words. In this paper, we experiment with RNNs on the direct input text. We employ RNN models to compare them with ML and ANN models, which use numerical features extracted from the text. Across the four datasets, we employed variants of RNNs such as Simple RNN, long short-term memory (LSTM), gated recurrent unit (GRU), Bi-directional LSTM to classify fake news. We designed a fixed architecture involving an embedding layer followed by an RNN layer. The RNN layer outputs are passed through a fully connected dense layer to produce the classification outputs. In between the dense layers and the RNN layers, we designed a ReLU activation layer and a dropout layer to give all neurons weightage while training. The GRU, LSTM, RNN models differ by the type of cell used in the RNN layer; they are GRU, LSTM, RNN cells, respectively, that make up this layer. The RNN cell is conceptually simple, consisting of a hidden state from the previous time-step, which influences the current time-step decision. The hidden state acts as a memory unit. The LSTM cell has an added advantage over the RNN cell since it consists of long-term and short-term memory, which proves to be useful while working with longer texts as encountered in the ISOT and FNM datasets. In these models, only the output of the last time step is considered in decision making. The Bi-LSTM model differs from the rest of the models. Here, we have designed the bidirectional layer to return the intermediate time-step outputs, which are then passed through a time distributed layer and flattened before going through the fully connected dense layer to produce the result. Unlike the other models, before making a decision, the Bi-LSTM model considers the text before and after a particular word.

These four model architectures are employed on all datasets while varying the optimiser. We have experimented with RMSprop (Root Mean Square Propagation), Adam, and SGD optimisers, hence giving us three variants in each architecture referred to as LSTM-1, LSTM-2, LSTM-3, RNN-1 and so on. The SGD optimiser is conceptually and behaviorally simple, moving along the cost surface in the direction of lower loss. This optimiser has been considered due to the shallow architectures used in this paper. However, this method may get stuck in local minima in the curve. The RMSprop optimiser is similar to the Adadelta optimiser available in Keras, with the distinction of having an exponential factor reducing the learning rate. We use RMSprop optimiser for its adaptive learning rate optimisation feature. The Adam optimiser, in addition to the RMSprop,

provides an adaptive momentum estimation. The optimisers are used with their default parameters to carry out the experiments. The batch sizes are updated based on the average length of the document in each dataset. The batch size used for ISOT, FNM, Liar-Liar, and KE is 64, 64, 32, and 8, respectively, each with 10 epochs, early stopping on validation loss.

4 Experimental Results

This section discusses the performance evaluation metrics carried out to test and validate our proposed features and models. We use standard binary and multiclass classifier accuracy, macro precision, macro recall, and area under the curve (AUC) of ROC to evaluate our models. Additionally, we employ each ML model with 10-fold cross-validation to address any possible overfitting in the results. Table 4 shows the accuracy results of all models categorised into ML, ANN, and context-based RNN models. The highlighted results in the table demonstrate the variants of models that perform significantly better than other variants employed on the same dataset.

Table 4. Performance Results of Classifiers Employed on All Experimental Datasets. P= Precision, R= Recall, and A= Accuracy

	FN Master			**ISOT**			**Liar-Liar**			**KE**		
	P	R	A	P	R	A	P	R	A	P	R	A
GNB	0.78	0.87	0.83	0.75	0.59	0.63	0.49	0.50	0.54	0.67	0.48	0.55
RF	0.98	0.96	0.98	0.92	0.92	0.92	0.56	0.56	0.58	0.63	0.49	0.50
DT	0.95	0.95	0.96	0.85	0.85	0.86	0.52	0.52	0.53	0.52	0.51	0.50
KNN	0.90	0.88	0.92	0.89	0.88	0.88	0.55	0.54	0.57	0.61	0.56	0.59
XGB	0.97	0.96	0.98	0.92	0.92	0.92	0.56	0.55	0.58	0.57	0.55	0.55
SVM	0.94	0.93	0.96	0.93	0.93	0.93	0.56	0.54	0.57	0.52	0.50	0.50
ANN-1	0.94	0.92	0.95	0.92	0.91	0.92	0.28	0.50	0.56	0.36	0.40	0.45
ANN-2	0.95	0.91	0.95	0.92	0.92	0.92	0.56	0.53	0.57	0.58	0.52	0.55
ANN-3	0.90	0.92	0.94	0.89	0.88	0.89	0.57	0.52	0.56	0.20	0.33	0.27
ANN-4	0.93	0.94	0.95	0.90	0.89	0.89	0.57	0.51	0.56	0.25	0.29	0.32
LSTM-1	0.56	0.52	0.74	0.96	0.96	0.96	0.59	0.57	0.60	0.23	0.28	0.32
LSTM-2	0.55	0.52	0.75	0.96	0.96	0.96	0.58	0.58	0.57	0.14	0.33	0.41
LSTM-3	0.38	0.50	0.77	0.27	0.50	0.55	0.29	0.50	0.57	0.28	0.33	0.36
RNN-1	0.38	0.50	0.77	0.47	0.48	0.51	0.55	0.55	0.57	0.20	0.25	0.27
RNN-2	0.59	0.50	0.77	0.75	0.57	0.61	0.54	0.54	0.56	0.23	0.31	0.36
RNN-3	0.38	0.50	0.77	0.27	0.50	0.55	0.54	0.53	0.57	0.32	0.42	0.50
GRU-1	0.54	0.52	0.74	0.94	0.93	0.93	0.57	0.57	0.54	0.14	0.33	0.41
GRU-2	0.54	0.51	0.75	0.94	0.93	0.93	0.58	0.50	0.57	0.14	0.33	0.41
GRU-3	0.38	0.50	0.77	0.27	0.50	0.55	0.29	0.50	0.57	0.28	0.33	0.36
BiLSTM-1	0.85	0.71	0.85	0.99	0.99	0.99	0.57	0.57	0.58	0.12	0.33	0.36
BiLSTM-2	0.67	0.72	0.73	0.99	0.99	0.99	0.59	0.59	0.60	0.30	0.34	0.41
BiLSTM-3	0.38	0.50	0.77	0.80	0.60	0.64	0.29	0.50	0.57	0.48	0.38	0.45

Table 4 shows that the instances FNM and ISOT datasets got classified with very high accuracy (mostly >90%). Whereas, KE and Liar datasets got abysmal accuracy and sometimes even lower than the baseline. Furthermore, our results reveal that among all models, tree-based models (random forest and decision trees), XGBoost, and support vector machine outperform other models. Interestingly, while ANN give significantly higher performance for ISOT and FNM including all its variants, it does poorly on both Liar and KE datasets. The best performance of ANN in these datasets is still <60%. Table 4 shows that almost all ML models and ANN variants gave above 92% accuracy on FNM dataset except GNB which provides 84% accuracy. Based on our manual inspection, we find that experts and journalists created KE and Liar datasets. The original

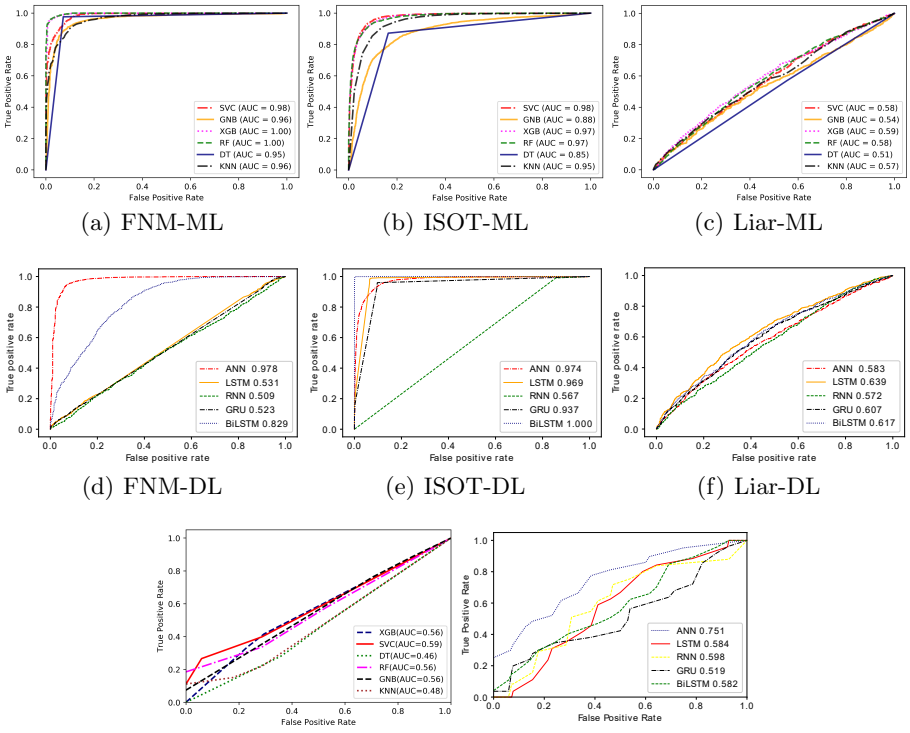
news is interpreted and explained in these datasets and hence lose the linguistic and deep semantic features of fake news. Therefore, the model is not able to find features in these datasets and results a poor accuracy. On contrast, ISOT and FNM datasets contain original news articles acquired from various sources. Hence they vary in writing style, topics, and other aspects. Across classification model, the statistics in Table 4 unveil that any of random forest, decision trees, XGBoost, support vectors or ANN can be used with the identified set of features. Table 4 further reveals that within the text based RNN variants, the Bidirectional LSTM model performs better than the other RNN models in all datasets. The Bidirectional LSTM model with RMSProp optimiser outperforms for both ISOT and FNM with an accuracy of 85% and 99%. Especially for the ISOT dataset, due to the large text length, the model extracts more value from the text; hence the LSTM with Adam optimiser and Bi-LSTM with RMSProp models show a higher performance than the models trained on our extracted features. Across other datasets, the models trained on extracted features perform better than the benchmark RNN models. For example, on Liar and KE datasets, the performance declines significantly especially affecting the macro precision for KE dataset. The microposts (very small text) datasets such as Liar and KE show lower accuracy in all models trained due to the lack of sizeable text to extract value from.

Table 5. Confusion Matrix for Outperforming ML, ANN, and RNN Models Employed on ISOT, FNM, and Liar Data. Each field represents a quadruple [TP, FN, FP, TN].

Model	ISOT	FNM	Liar
GNB	[676, 2789, 80, 4158]	[797, 36, 577, 2177]	[117, 779, 157, 992]
RF	[3084, 381, 244, 3994]	[770, 63, 20, 2734]	[338, 558, 303, 846]
DT	[2898, 567, 542, 3696]	[772, 61, 65, 2689]	[417, 479, 489, 660]
KNN	[2807, 658, 242, 3996]	[671, 162, 115, 2639]	[251, 645, 230, 919]
XGB	[3070, 395, 220, 4018]	[775, 58, 30, 2724]	[290, 606, 263, 886]
SVM	[3136, 329, 217, 4021]	[740, 93, 65, 2689]	[274, 622, 252, 897]
ANN	[3071, 400, 225, 4007]	[739, 79, 89, 2680]	[139, 769, 116, 1021]
LSTM	[3250, 249, 49, 4155]	[78, 755, 158, 2596]	[262, 615, 193, 975]
RNN	[502, 2997, 40, 4164]	[8, 825, 11, 2743]	[332, 545, 332, 836]
GRU	[3154, 345, 168, 4036]	[66, 767, 143, 2611]	[639, 238, 693, 475]
BiLSTM	[3492, 7, 10, 4194]	[378, 455, 66, 2688]	[490, 387, 441, 727]

Table 5 shows the detailed result on classification in form of a confusion matrix (true positive TP, false positive FP, true negative TN, and false negative FN) where positive and negative classes correspond to fake and real news categories, respectively. Since KE dataset has the worst performance among all datasets, we only report the confusion matrix of other datasets with all models with their outperforming variants. Table 5 reveals that GNB trained model on only one class and learnt very few about the second class, hence a large number of FNs. Results show that random forest increases the number of TPs by a huge margin in comparison to GNB and reduces the false alarms. The table also reveals that the number of TNs does not change significantly in GNB, while other models learn on positive class and hence increases the number of TPs by reducing FNs. While ANN increases TPs by a huge percentage, XGBoost increases the number of TNs without compromising with TP rate, unlike GNB. Confusion matrix statistics of Liar data reveals that neither of the models can segregate fake news

from real effectively. If XGBoost classifies real news well, then it decreases the recall on fake news (learns from negative class only). Another outperforming model ANN increases the recall of positive class (fake) but reduces the recall of negative class by a huge margin. This happens due to lack of variation in real or fake news in the dataset as they are the interpretations and not the original news. Among context-aware models, both LSTM and Bi-LSTM learn well on both positive and negative classes, reducing the number of FNs and FPs. However, simple RNN and GRU do not learn the fake class well in FNM dataset. Within Liar dataset, all context-aware models do not learn enough from either of the classes and hence give poor TP and TN resulting into poor accuracy.



(a) FNM-ML (b) ISOT-ML (c) Liar-ML

(d) FNM-DL (e) ISOT-DL (f) Liar-DL

(g) KE-ML (h) KE-DL

Fig. 2. ROC Curves of Various Models Employed on Experimental Datasets

In addition to the accuracy metrics, we also plot the area under the ROC curve (AUC) for each model and dataset pair. Figures 2(a), 2(b), and 2(c) shows the AUC for all ML algorithms employed on FNM, ISOT, and Liar datasets. The graphs reveal that all models perform outstandingly on FNM and give a minimum AUC of 0.95 while XGB and RF gives a perfect AUC. These statistics reveal that for any randomly selected input, XGB and RF will predict the classes correctly. Interestingly, all these models perform similarly on ISOT data as well. Since FNM is a more descriptive dataset and has a large number of records, we find a slightly better performance of the models in FNM compared to ISOT.

Furthermore, for the ISOT dataset, SVM outperforms XGB and RF. In the KE and Liar dataset, we find that while the dataset's size is one of the primary issues of misclassification, the nature of the dataset (interpretation of actual news articles) is the major setback for the model. Figures 2(d) and 2(e) reveal that ANN gives high performance for both FNM and ISOT datasets. While, Bi-LSTM gives 100% AUC for ISOT, it gives a reasonably good performance for FNM data which aligns with our explanation of results from Tables 4 and 5. However, the simple RNN model does not do well for any of the datasets. Despite the poor performance on Liar (Figures 2(c) and 2(f)) and KE (Figures 2(g) and 2(h)) dataset, we include these graphs because they reveal the insights on datasets. For ISOT and FNM, the results support our features vectors and show that they are reliable indicators of a document to be fake news. Hence they work significantly well with each model with a little difference in the accuracy. Moreover, since the features set is robust, this work proposes a performance-effective solution and an inexpensive solution. Furthermore, the proposed features are generic and cover a variety of topics under fake news detection. Hence, the work presented in this paper can be trained and validated for unseen documents of different domains. We believe that our results and discussed features set are useful for the research community because they do not rely on crowdsourcing or network information but only the fake news content. This paper presents the profiling of misinformation and trains the model based on these profiles or characteristics.

4.1 Reproducibility of Results

We plan to make our source code and results public on Google Cloud and Kaggle upon acceptance. Currently, we share an anonymous DropBox link (`https://bit.ly/30bJ1kc`) to the source codes (Jupyter Notebooks) and the obtained results. Since the datasets used in this research are open-source , we do not share the database files. We confirm that the results presented in this paper are reproducible using the files provided in the DropBox.

5 Conclusion and Future Work

In this paper, we proposed various features set to profile fake news. These features include statistical, syntactic, semantic, temporal, writing cues, emotion, topical, and named entities based attributes. We employ several ML algorithms, ANN, and RNN variants to test and validate our proposed features. Our results reveal that ensemble-based classifiers outperform singleton classifier models. In contrast to FNM and ISOT datasets, KE and Liar datasets give poor performance. We observe that these datasets are not the original news but their interpretation. Hence, all instances (genuine and fake information) share the same writing style, grammar structure, and emotional range. We conclude that with the proposed features set, XGBoost, SVM, and tree-based algorithms give at least 90% accuracy in the classification. Similar to ML models, ANN models and Bi-LSTM also outperform for ISOT and FNM datasets. Due to the small text and interpretation in KE and Liar datasets, Bi-LSTM does not learn more value from the text and performs poor than the models employed on extracted features. Future work includes mining the multimedia metadata associated with

news articles. We also plan to have domain-specific concept extraction for specialised fake news detection. For example, fake news specific to COVID-19 pandemic and health beliefs.

References

1. Afroz, S., Brennan, M., Greenstadt, R.: Detecting hoaxes, frauds, and deception in writing style online. In: Symposium on Security and Privacy. pp. 461–475. IEEE (2012)
2. Ahmed, H., Traore, I., Saad, S.: 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. pp. 127–138. Springer (2017)
3. Ahmed, H., Traore, I., Saad, S.: Detecting opinion spams and fake news using text classification. Security and Privacy **1**(1), e9 (2018)
4. Allcott, H., Gentzkow, M.: Social media and fake news in the 2016 election. Journal of economic perspectives **31**(2), 211–36 (2017)
5. Balmas, M.: When fake news becomes real: Combined exposure to multiple news sources and political attitudes of inefficacy, alienation, and cynicism. Communication research **41**(3), 430–454 (2014)
6. Banerjee, R., Feng, S., Kang, J.S., Choi, Y.: Keystroke patterns as prosody in digital writings: A case study with deceptive reviews and essays. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1469–1473 (2014)
7. Bovet, A., Makse, H.A.: Influence of fake news in twitter during the 2016 us presidential election. Nature communications **10**(1), 1–14 (2019)
8. Brennen, B.: Making sense of lies, deceptive propaganda, and fake news. Journal of Media Ethics **32**(3), 179–181 (2017)
9. Chandra, Y.U., et al.: Higher education student behaviors in spreading fake news on social media: A case of line group. In: 2017 International Conference on Information Management and Technology (ICIMTech). pp. 54–59. IEEE (2017)
10. Conroy, N.K., Rubin, V.L., Chen, Y.: Automatic deception detection: Methods for finding fake news. Proceedings of the Association for Information Science and Technology **52**(1), 1–4 (2015)
11. Cunha, E., Magno, G., Caetano, J., Teixeira, D., Almeida, V.: Fake news as we feel it: perception and conceptualization of the term "fake news" in the media. In: International Conference on Social Informatics. pp. 151–166. Springer (2018)
12. Dale, R.: NLP in a post-truth world. Natural Language Engineering **23**(2), 319–324 (2017)
13. Feng, V.W., Hirst, G.: Detecting deceptive opinions with profile compatibility. In: 6th International Joint Conference on Natural Language Processing. pp. 338–346. Asian Federation of Natural Language Processing / ACL (2013)
14. Ghosh, S., Shah, C.: Towards automatic fake news classification. Proceedings of the Association for Information Science and Technology **55**(1), 805–807 (2018)
15. Horne, B.D., Adali, S.: This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In: 11th International AAAI Conference on Web and Social Media (2017)
16. Kumar, S., West, R., Leskovec, J.: Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes. In: Proceedings of the 25th international conference on World Wide Web. pp. 591–602 (2016)

17. Liberty, E., Lang, K., Shmakov, K.: Stratified sampling meets machine learning. In: International conference on machine learning. pp. 2320–2329 (2016)
18. Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., Stein, B.: A stylometric inquiry into hyperpartisan and fake news. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 231–240 (2018)
19. Ray, A., George, J.: Online disinformation and the psychological bases of prejudice and political conservatism. In: Proceedings of the 52nd Hawaii International Conference on System Sciences. pp. 1–11. ScholarSpace (2019)
20. Riedel, B., Augenstein, I., Spithourakis, G.P., Riedel, S.: A simple but tough-to-beat baseline for the fake news challenge stance detection task. arXiv preprint arXiv:1707.03264 (2017)
21. Roets, A., et al.: 'fake news': Incorrect, but hard to correct. the role of cognitive ability on the impact of false information on social impressions. Intelligence **65**, 107–110 (2017)
22. Romero, D.M., Meeder, B., Kleinberg, J.: Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In: 20th international conference on World wide web. pp. 695–704 (2011)
23. Ruchansky, N., Seo, S., Liu, Y.: CSI: A hybrid deep model for fake news detection. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 797–806. ACM (2017)
24. Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M., Liu, Y.: Combating fake news: A survey on identification and mitigation techniques. ACM Transactions on Intelligent Systems and Technology (TIST) **10**(3), 1–42 (2019)
25. Shu, K., Mahudeswaran, D., Wang, S., Lee, D., Liu, H.: Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. Big Data **8**(3), 171–188 (2020)
26. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter **19**(1), 22–36 (2017)
27. Shu, K., Wang, S., Liu, H.: Exploiting tri-relationship for fake news detection. arXiv preprint arXiv:1712.07709 **8** (2017)
28. Cardoso Durier da Silva, F., Vieira, R., Garcia, A.C.: Can machines learn to detect fake news? a survey focused on social media. In: Proceedings of the 52nd Hawaii International Conference on System Sciences (2019)
29. Singh, V.K., Ghosh, I., Sonagara, D.: Detecting fake news stories via multimodal analysis. Journal of the Association for Information Science and Technology **72**(1), 3–17 (2021)
30. Singhania, S., Fernandez, N., Rao, S.: 3han: A deep neural network for fake news detection. In: Neural Information Processing - 24th International Conference. vol. 10635, pp. 572–581. Springer (2017)
31. Tandoc Jr, E.C., Lim, Z.W., Ling, R.: Defining "fake news" a typology of scholarly definitions. Digital journalism **6**(2), 137–153 (2018)
32. Wang, W.Y.: "liar, liar pants on fire": A new benchmark dataset for fake news detection. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 422–426. ACL (2017)
33. Zhang, X., Ghorbani, A.A.: An overview of online fake news: Characterization, detection, and discussion. Information Processing & Management **57**(2) (2020)
34. Zhao, J., Cao, N., Wen, Z., Song, Y., Lin, Y.R., Collins, C.: # fluxflow: Visual analysis of anomalous information spreading on social media. IEEE transactions on visualization and computer graphics **20**(12), 1773–1782 (2014)