Towards Automatic Annotation and Detection of Fake News

Mohammad Majid Akhtar
The University of New South Wales
Sydney, Australia
majid.akhtar@unsw.edu.au

Ishan Karunanayake
The University of New South Wales
Sydney, Australia
ishan.karunanayake@unsw.edu.au

Bibhas Sharma
The University of New South Wales
Sydney, Australia
bibhas.sharma@home-in.com.au

Rahat Masood
The University of New South Wales
Sydney, Australia
rahat.masood@unsw.edu.au

Muhammad Ikram
Macquarie University
Sydney, Australia
muhammad.ikram@mq.edu.au

Salil S. Kanhere
The University of New South Wales
Sydney, Australia
salil.kanhere@unsw.edu.au

***Abstract*—Automated accounts or bots on Online Social Networks (OSNs) play a significant role in disseminating information, including false news, which may instigate cyber propaganda. The existing research on fake news detection does not account for the existence of bots. Also, they only focus on identifying fake news in "the articles shared in posts" rather than the post's (textual) content and use manually labeled limited datasets. In this research, we overcome the challenge of data scarcity by proposing an automated approach for labeling data using verified fact-checked statements on OSNs such as Twitter. Moreover, we analyze the presence and impact of bots and show that bots change their behavior over time. Our experiments focus on COVID-19, collect 10.22 million COVID-19-related tweets, and use our annotation model to build an extensive ground truth dataset for classification purposes. We evaluated our automatic annotation model on two existing COVID-19-related misinformation datasets and achieved a $\sim$2% increase in precision compared to the existing annotation models. In addition, our best classification model achieves 83% precision, 96% recall, and a $\sim$4% false positive rate on our annotated dataset, outperforming existing techniques.**

***Index Terms*—COVID-19, Misinformation Detection, Automatic Annotation, Online Social Networks, Social Bot Detection**

I. Introduction

Social media is generally a rich source of (mis)information for its users. Of particular concern, misinformation often spreads deeper (as independent cascades) and reaches a wider user base than genuine information. Many users believe misinformation to be accurate due to various psychological phenomena like confirmation bias, naive realism, or homeostasis [1]. Automated accounts, also known as "bots", actively spread misinformation, posing a severe threat to genuine users of OSNs by hijacking public discussions and promoting their malicious goals [2]. Bots are often designed to mimic the behavior of genuine users [3], thus confounding genuine users to believe that conversations are organic rather than artificially promoted. An example of bot-fueled misinformation campaigns is the conspiracy theory that associated 5G technology with the COVID-19 outbreak in Wuhan in 2019 [4]. Similar misinformation was prevalent during the COVID-19 pandemic when many users turned to social media for updated information, where they were exposed to false remedies, practices, and other conspiracy theories [5], [6].

Recently, researchers have utilized knowledge bases and artificial intelligence to detect misinformation in social media [7], [8] or news articles [9], [10]. However, most existing works only detect false information in OSN-based content with no emphasis on fake accounts that disseminate the false information in the initial phase of propagation [11]. Moreover, the current research has limitations in collecting labeled misinformation data efficiently as most works resort to manual annotation (or labeling) of data [7], [8]. Our work differs from existing studies as we *automate the data labeling using machine learning and natural language processing techniques that help identify misinformation in social media posts (text in posts) with high accuracy*. Further, we analyze and detect the *presence and impact of OSN bots responsible for spreading misinformation*. Our work makes the following main contributions:

1) We *propose an automatic annotation model* to annotate the textual content of tweets using supporting statements (i.e., verified fact-checked statements from government officials or experts). Our model uses a relevance matching (between tweets and supporting statements) machine learning model [12] and a labeling algorithm to build an extensive ground truth dataset containing binary labels (*fake* or *real*) for information (or news) in the tweets. We evaluated our annotation model on two public datasets and achieved a $\sim$2% increase in precision, compared to the annotation model in [13].

2) We *design and develop an ensemble stack model* to detect fake information by combining and evaluating the applicability of various supervised learning classifiers. Stacking the individual classifiers overcomes overfitting and achieves better performance than any individual classifier used for classification. We validate and test our model over a dataset of 10.22 million COVID-19-related tweets. For every tweet, we use three types of features: tweet-level (such as no. of

URLs in a tweet), user-level (such as followers count), and textual (tweet's content). We use three different techniques to extract textual features, i.e., a standard BERT Transformer model, COVID-Twitter-BERT model, and Term Frequency-Inverted Document Frequency (TF-IDF). We show that the *ensemble-based machine learning classifier* consisting of Support Vector Machine (SVM), Random Forest, and Logistic Regression with TF-IDF performs best with 83% precision and 96% recall. We recreated four baseline misinformation classification models and compared their performance against our classification model using three public COVID-19-related misinformation datasets. Our classification model performs best against all three datasets.

3) We *investigate the impact of bots in our dataset* and find that bots generated approximately 10% of misinformation tweets attaining 0.5 million retweets. We utilize a metric, bot score, which helps determine the probability of a user account being operated by a bot. This allows us to compare the behaviors or actions of bot accounts across two distinct time periods: June-August 2021, which corresponds to the peak of the COVID pandemic, and August 2022, which is after the peak of the pandemic. During Jun-Aug 2021, 5,315 unique accounts were identified as bots, whereas the number was 3279 in Aug 2022. This shows that more bot accounts have been active during the peak COVID times, confirming that bot behavior changes over time and is most active during misinformation campaigns compared to other times to maximize the goals of the bot campaign.

The rest of the paper is structured as follows: SectionII discusses the related work, Section III describes our data collection method, and Section IV elaborates our proposed model for annotating misinformation. Our fake news detection methodology is presented in Section V. We leverage the annotated data to analyze and detect fake news and bots in Section VI and Section VII, respectively. Section VIII concludes the paper.

II. Related Work

This section reviews prior work on dataset annotation and classification of false information. When considering existing annotation models, Zhang et al. [14] presented a method for annotating news with content and context indicators, such as looking for click-bait titles, logical fallacies, and reputations of citations. Though these indicators are helpful, the proposed manual annotation approach is not scalable. Likewise, Wang et al. [15] suggested using user reports, which are tedious and not always available. Few other research, such as Bonet-Jover [16] use semi-automatic annotation models that rely heavily on human-in-the-loop for annotation. On the other hand, Perez-Rosas et al. [17] proposed an innovative way of increasing the annotated dataset by producing a fake version of a true version of news with the help of Amazon MTurk workers. All the aforementioned annotation models involve manual verification, which is an expensive and time-consuming process. At last, Paka et al. [13] proposed Cross-SEAN, an automatic annotation method based on the cosine similarity threshold between tweets and the fact-checked statements. However, their method only considers the first matched fact-checked statement, unlike our model which considers multiple supporting statements.

There exist several models that classify false information, and they usually rely on three main types of input features; tweet's textual features, user-level features, and tweet-level features. For example, Patwa et al. [18] used a simple model with Linear SVM applied on TF-IDF features, while Ahmed et al. [19] used N-gram with 50,000 features fed to the Linear SVM model. Wang et al. [20] used BERT with the BiLSTM model to identify fake news. All the above techniques use textual features. Meanwhile, Al-Rakhami et al. [21] proposed a stacked ensemble model using LinearSVM and Random forest as the base learner and C4.5 decision tree as the meta-learner. Their model incorporates both user-level and tweet-level features such as the number of friends, followers, URLs, and hashtags. Our model uses all three types of features to provide additional indicators for more robust classification.

Most previous works were either tested on small annotated datasets or focused detecting on fake news information in "news articles" rather than social media posts [22]. For instance, Wang et al., [9] used hybrid-CNN to detect fake news in short political statements extracted from Politifact. Similarly, Aldwairi and Alwahedi [23] used the syntactical structure of web links and words of the news titles to detect clickbait and fake news. Shu et al. [10] focused on identifying the partisanship of news publishers, and Wang et al. [15] used reinforcement learning to detect fake news in new articles based on user reports.

Our work is different from previous works as we identify misinformation in social media posts instead of fake news articles or URLs that link to external websites. Our work focuses on the post's text that is used for spreading misinformation since social media is a conversation-based platform. In summary, we provide annotation and classification models that leverage all three features to identify misinformation. Lastly, unlike existing studies, we are not limited to only fake news detection but extend our research to discuss the presence and impact of bot-generated misinformation tweets.

III. Data Collection Methodology

For validation of our work, we use the COVID-19 pandemic as a use case as the impact of misinformation was widespread containing false remedies, practices, and conspiracy theories [5]. In the following, we present our data collection and filtering methodology.

Data Collection. We begin by collecting tweets related to COVID-19 using a dataset from the Panacea Lab [24], which contains roughly 730 million COVID-19-related tweet IDs. We use these tweet IDs to extract information, such as the tweet's textual content and metadata (by using Python libraries like Twarc [25] and the Twitter API). Our data crawling process collected tweets from 1st January 2020 to 21st June 2021 (533 days). Note that the selection of the dataset during the aforementioned time periods is based on the fact that the prevalence of misinformation was widespread during the COVID

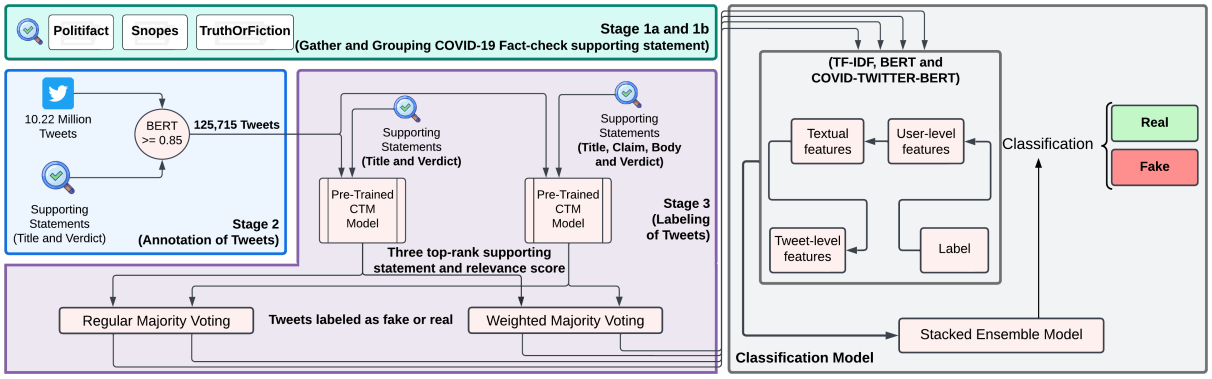

Fig. 1: Data collection, annotation, and classification pipeline for Fake News Detection.

pandemic and encompasses various instances of conspiracy theories and false remedies [4], [5].

Data Filtering. During the pandemic, some countries were exposed to more misinformation than others due to the significantly high Infection Rates (IRs), causing more panic among citizens [26]. Therefore, we selected three geolocations, India, the United States of America (US), and the United Kingdom (UK), based on their higher IRs and death counts. We also selected Australia due to its more restrictive strategy of imposing a nationwide lockdown and border closure for a prolonged period. Next, we excluded non-English tweets from the data of these selected countries. Our final dataset contains 10.22 Million tweets for further analysis.

Ethics Consideration. We obtained ethics approval for data collection from our organization's ethics board. We do not intend to use, track, or de-anonymize users during our data collection, abiding by the ethics guidelines in [27]. Also, the data collected was not released publicly, and no personally identifiable information besides tweets and other metadata was collected.

IV. Misinformation Annotation Model

This section provides an outline of the method used to annotate (or label) tweets in our dataset as *fake* or *real* to solve the data scarcity problem. Our proposed model involves three stages i.e., data gathering, filtering fake and real tweets (annotation), and labeling of tweets. Figure 1 illustrates the data collection, annotation, and classification pipeline that comprises distinct stages which are elaborated below.

Stage 1a. Gather COVID-19 Fact Checks: Fact-checking websites such as *Snopes*[1], *Politifact*[2] and *TruthOrFiction*[3] examine trending news across social media and use evaluations from journalists and experts, to give a verdict (i.e., *real* or *fake*). We collect these fact-checked statements as supporting statements and use it in our annotation model. We follow two data scraping approaches: *without body text* and *with body text*. In the first approach, we only extract the corresponding title and verdict, while in the second approach, the claim (statements tested for verifiability) and the body content (up to 350 words) are extracted in addition to the title and the verdict. Our intuition behind the second approach is to improve semantic matching using auxiliary information and avoid over-matching similarities of common words like COVID, vaccine, and virus. We restricted scrapping upto 350 words to cater to the memory requirements of the algorithms used in §IV (Stage 3). We collected 1,923 COVID-19 supporting statements for use in the next stage.

Stage 1b. Grouping verdicts of COVID-19 Fact Checks: Many supporting statements–labels have different verdict classes, such as *false, true, misattributed, pants-fire, etc.* As we are focusing on binary classification (i.e. *fake* or *real*), first, we need to group those different verdict classes into two. For example, we grouped together verdicts like pants-fire (from PolitiFact), misattributed (from Snopes), not true (from TruthOrFiction), and similar other untrue instances of information (full-flop, false, mostly-false) to the same category, *fake*. We perform a similar step for the *real* category and discarded supporting statements with complex verdicts like *unknown, unproven, research in progress*. This led to 1,655 supporting statements that were labeled with binary classes.

Stage 2. Filter real and fake tweets (Annotation): Herein, we follow a two-step process. First, we collect all tweets in the dataset that were posted by health organizations such as WHO, National Health Service (NHS), Governments' health portals or websites, etc. These tweets were annotated as *real* based on the assumption that the content posted by such health organizations is genuine. Next, we used the BERT transformer model to generate embeddings of the supporting statements (titles) and the tweets. We calculate the pairwise cosine distance between the supporting statement and the tweets with a defined threshold. To determine the threshold, we make use of a publicly available COVID-19 misinformation dataset [18] consisting of approximately 8,560 labeled tweets. In Table I, we observe that the threshold has an inverse relationship with the number of annotated tweets. For instance, with a threshold of 0.70, 6,528 tweets are annotated, and with a threshold of 0.90, 430 tweets are annotated.

In contrast, the *threshold value has a linear relationship with the accuracy of accurately predicting fake/real news*. With the same 0.70 threshold, we achieved 50% accuracy, while it increased to 98% with a threshold of 0.90. Considering both these factors, 0.85 was deemed an appropriate threshold for pairwise cosine distance. We used this threshold to annotate

[1] https://snopes.com/ [2] https://politifact.com/ [3] https://truthorfiction.com/

TABLE I: Accuracy and no. of annotated tweets by thresholds.

Cosine Similarity Threshold	Prediction Accuracy	# Tweets annotated (n = 8,560)
0.70	50.1%	**6,528**
0.80	71.9%	1,953
0.85	91.7%	668
0.90	98.6%	**430**

the total corpus of 125,715 tweets resulting in 17,289 *real* and 108,426 *fake* tweets. We named this annotated dataset as *labeled data 0*.

Stage 3. Labeling Tweets: We improve the accuracy of labeling by considering three supporting statements from three different fact-checking organizations. First, we used a pre-trained Contextual Text Matching (CTM) model [12], to measure the similarity between a tweet and supporting statements and determine if the supporting statement fact-checks the tweet. The CTM model [12] is pre-trained on 467 Politifact fact-check articles, as most statements are from Politifact. Next, we used this pre-trained CTM model for testing. When a tweet (t) is tested using the CTM model, it gives three top-rank statements (s) from the supporting statements corpus that fact-checks the tweet with a relevance score $f(t, s)$. The higher the relevance score, the higher the probability that the supporting statement fact-checks the tweet. Furthermore, we removed six tweets from the entire set of 125,715 tweets as a relevance score could not be generated for them. We tested the remaining 125,709 tweets with both forms of scrapped data (supporting statements), i.e. *without body text* and *with body text* using the CTM model and achieved three top-rank supporting statements for each tweet with their relevance score. We refer to the outcome of this process as the $tweet_{result}$. As we have only acquired verified fact-checked statements from three fact-checking websites, i.e., Snopes, Politifact, and TruthOrFiction, we have only used three supporting statements. Here, we assume that the same misinformation story is covered only once by a single fact-checking website and using three supporting statements allows us to consider the verdict from every website if they all cover the same misinformation story.

Next, we use two labeling algorithms to determine the final label for the tweet using its $tweet_{result}$. First, a majority vote is calculated for the three top-rank statements in $tweet_{result}$ using only the verdict of each statement. For example, if two of the three statement's verdicts in $tweet_{result}$ is fake, then the tweet is fake. We call this method as **regular majority voting**. Second, we introduce a weighting mechanism to give importance (weights) to supporting statements based on their cosine similarity and relevance scores in the $tweet_{result}$. For example, suppose rank 2 and 3 supporting statement's verdict is fake with a low relevance score and cosine similarity, but rank 1 verdict is *real* with a high relevance score and cosine similarity. In that case, the tweet is labeled as *real*. Next, we iterate through three top-ranked supporting statements for each tweet and calculate two scores, i.e., $score_{fake}$ and $score_{real}$ using the supporting statement's verdict, cosine similarity and relevance score. Finally, we compare both $score_{fake}$ and $score_{real}$ to label the respective tweet. The algorithm for **weighted majority voting** is described in Algorithm 1. In summary, we assign a label to tweets using regular and weighted majority voting, each with scrapped supporting statements (*without body text* and *with body text*). This results in four variants of the labeled data (1-4) as shown in Table II.

Algorithm 1 Algorithm for Weighted Majority Voting

for each tweet in $tweet_{result}$ **do**
  $score\_fake \leftarrow 0$; $score\_real \leftarrow 0$; $cosine\_sim\_sum \leftarrow 0$
  $cosine\_sim = [\,]$; $tweet\_embedding \leftarrow bert.encode(tweet)$
  for i in range(0,3) **do** ▷ for each supporting_statement (ss)
    $ss\_embedding \leftarrow bert.encode(ss[i])$
    $sim = cosine\_similarity(tweet\_embedding, ss\_embedding)$
    $cosine\_sim\_sum = cosine\_sim\_sum + sim$; $cosine\_sim.append(sim)$
  end for
  for i in range(0,3) **do** ▷ for each supporting_statement
    $verdict \leftarrow verdict\_of\_supporting\_statement\_at\_rank\_i$
    $relevance\_score \leftarrow relevance\_score\_of s\_statement\ at\ rank\ i$
    if verdict == 'fake' **then**
      $score\_fake \leftarrow score\_fake + (cosine\_sim[i]/cosine\_sim\_sum) * relevance\_score$
    else if verdict == 'real' **then**
      $score\_real \leftarrow score\_real + ((cosine\_sim[i]/cosine\_sim\_sum) * relevance\_score)$
    end if
  end for
  if score_fake $\geq$ score_real **then** tweet in $tweet_{result} \leftarrow fake$ **else** $tweet_{result} \leftarrow real$
end for

TABLE II: Train and Test Set. We named our four datasets (column *'Labeled as Data *'*).

Labeled Data Description		Labeled *	Total Tweets	Train (80%)		Test (20%)	
				Fake	Real	Fake	Real
cosine distance >= 0.85		Data 0	125,715	86,738	13,834	21,688	3455
Without body text	Regular Majority	Data 1	125,709	74,638	25,929	18,710	6,432
	Weighted Majority	Data 2	125,709	74,008	26,559	18,611	6,531
With body text	Regular Majority	Data 3	125,709	77,200	23,367	19,261	5,881
	Weighted Majority	Data 4	125,709	75,697	24,870	18,903	6,239

V. Proposed Fake News Detection Methodology

Feature Extraction. We extract three different types of features for each tweet: tweet-level†ato, user-level, and textual. For tweet-level features, we consider a range of different attributes including - *number of user mentions, number of hashtags, number of URLs, number of favourites, number of retweets, number of media, is a reply (0,1), number of special characters* and *tweet length*. For user-level features, we consider attributes such as *is verified user (0,1), number of followers, number of friends, number of favourites* and *number of statuses*. We apply TF-IDF and the BERT transformer model to extract features from the tweet text. Further, we have used two different BERT models for sentence embeddings, i.e., a standard (BERT) model and COVID-Twitter-BERT model [28] that is pre-trained on a corpus of 97 million Twitter data related to COVID-19. We use COVID-Twitter-BERT to evaluate our model with a domain-specific BERT transformer model for the downstream task (classification).

Stack Ensemble Model Architecture. Our proposed model architecture consists of an ensemble learning approach

wherein multiple single machine-learning models (i.e. base learners) are trained, and a final meta-learner combines the predictions of each base learner to produce a final prediction as shown in Figure 2. This increases the performance and enhances the model's ability to generalize a trend from the training samples.

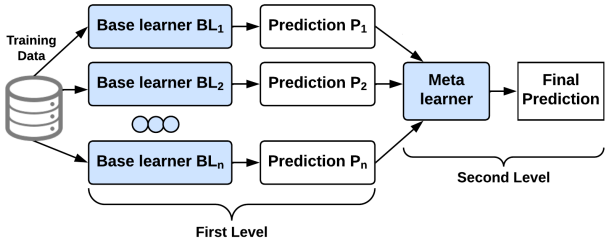

Fig. 2: Stacked Ensemble model.

VI. Experimental results

This section presents results on the COVID-19 misinformation classification and detection task with our proposed model.

A. Performance comparison of Single Base-learners

The first step is to determine the appropriate base learners. For this, we selected multiple machine learning classifiers and individually compared the performances of each of those models to train them with the (*labeled data 0*) dataset. We use precision and recall as metrics to evaluate the classifiers. As the dataset is imbalanced, accuracy is not deemed an ideal performance metric in this scenario. Table III shows the performance of the six base learners we used. We can see that Random Forest (RF) outperforms all the other models with a *precision* score of 92% and *recall* of 92% since RF avoids overfitting using multiple decision trees.

TABLE III: Performance of base-learners.

Base-learner	Weighted Precision	Weighted Recall	F1-score
KNN	82%	86%	83%
Decision Tree	87%	89%	87%
Random Forest	**92%**	**92%**	**91%**
Logistic Regression	79%	87%	81%
Support Vector Machine	80%	51%	59%

B. Determining the Appropriate Meta-learner

We consider four different machine learning classifiers and compared their performance to determine the best meta-learner. As base-learners, we selected KNN, Decision Tree, and RF at this experiment stage as they were the top three performing base-learners (individual models) in Table III. The performance of candidate meta-learners is illustrated in Table IV. Logistic Regression (LR) emerges as an appropriate choice for the meta-learner as it outperforms all the other models. This is because LR is a simple model that is less prone to over-fit base-learners predictions. It is worth noting that the SVM model takes significantly longer to run (2,821 seconds) compared to the LR model (1,183 seconds). This is the primary reason for not choosing the SVM model at this stage.

TABLE IV: Performance comparison of meta-learners.

Meta-learner	WP	WR	F1-score
Logistic Regression	**92%**	**93%**	**92%**
KNN	91%	92%	91%
Decision Tree	92%	92%	92%
Random Forest	92%	92%	92%
Support Vector Machine	92%	93%	92%

C. Determining the Appropriate Base-learner Combination

We selected three models to determine the appropriate combination of base learners with the Logistic Regression as the meta-learner. Table V shows that all combinations, except for KNN+Decision Tree, exhibit similar performance. This observation aligns with the fact that KNN involves distance calculation with each existing point, thus resulting in a decreased performance with a large dataset. To ensure simplicity, faster training, and testing, we selected Decision Tree and Random Forest as the base learners for the final model. Figure 3 depicts our final model for detecting misinformation.

TABLE V: Comparison of the combination of base-learners.

Base-learners	WP	WR	F1- score
KNN + Decision Tree	87%	89%	87%
KNN + Random Forest	92%	93%	92%
Decision Tree + Random Forest	**92%**	**93%**	**92%**
SVM + Random Forest	92%	93%	92%
All Models	92%	93%	92%

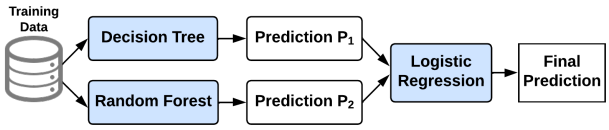

Fig. 3: Overview of our methodology for detecting Fake News.

D. Performance Evaluation of Our Annotation Model

To evaluate the annotation capability of our annotation model, we selected two of the COVID-related publicly available annotated datasets (D1 and D2) from Table VI. Firstly, we employed a pairwise cosine similarity function between the BERT embeddings of tweets in the two datasets and supporting statements. Secondly, we applied a threshold of 0.85 from our previous experiment (cf. §IV) to obtain a subset of the dataset for acquiring the final label from our annotation model. To compare our results with other state-of-the-art, we selected the Cross-SEAN model [13], which employs a similar approach of automatically annotating the tweets based on the BERT similarity (using a threshold $\geq$ 0.85 in this paper) between tweets and supporting statements. However, the Cross-SEAN model assigns the verdict with the first supporting statement that crosses the 0.85 threshold. We observe that our annotation model performs equivalent to the baseline model while improving the precision, true positives (TP), and false negatives (FN) as shown in Table VII.

TABLE VI: Details of the publicly available annotated datasets.

Dataset Name	Features Available	Total Tweets	After Hydrating	Train (80%)		Test (20%)	
				Fake	Real	Fake	Real
ANTi-Vax [29] (D1)	Tweet Content, User and Tweet-level	15,074	7,894	3,120	3,120	827	827
CMU-MisCov19 [30] (D2)	Tweet Content, User and Tweet-level	4573	1,726	689	689	174	174
COVID-19 Fake News Dataset [18] (D3)	Only Tweet Content	8,560	8,160	3,268	3,268	812	812

TABLE VII: Performance evaluation of our annotation model

Dataset and Model Name	Size	Correct Prediction		Wrong Prediction		Correctly Annotated	Precision	Recall	F1-score
	(≥ 0.85 threshold for BERT)	TP	TN	FP	FN				
(ANTi-Vax [29]) Our Model	586/15,074	**9**	460	22	**95**	80%	**83%**	95%	89%
(ANTi-Vax [29]) Cross-SEAN [13]	586/15,074	1	474	8	103	81%	82%	98%	90%
(COVID-19 [18]) Our Model	668/8,560	**33**	551	59	**25**	87%	**96%**	90%	93%
(COVID-19 [18]) Cross-SEAN [13]	668/8,560	17	596	14	41	91%	94%	98%	97%

TABLE VIII: Comparison of our Fake News detection model with baselines.

Model Type	Work	Method	Feature	**D1** [29] *ANTi-Vax*	**D2** [30] *CMU-MisCov*	**D3** [18] *COVID-19*
				Accuracy	**Accuracy**	**Accuracy**
Vectorizer -based	Patwa et al. [18]	TF-IDF with Linear SVM	Only Tweet Content	97%	84%	92%
	Ahmed et al. [19]	N-gram (unigram) with Linear SVM	Only Tweet Content	97%	84%	92%
	Al-Rakhami et al. [21]	Stacked (Linear SVM + RF + C4.5)	User-level, Tweet-level	74%	61%	N/A
	Our Model	TF-IDF with Stacked$_{DT}$	All three features	94%	75%	88%
		TF-IDF with Stacked$_{SVM}$	All three features	**97%**	**85%**	**92%**
Transformer -based	Wang et al. [20]	BERT with BiLSTM	Only Tweet Content	97%	82%	95%
	Our Model	BERT with Stacked$_{DT}$	All three features	91%	72%	85%
		BERT with Stacked$_{SVM}$	All three features	91%	72%	90%
	Our Model	COVID-Twitter-BERT with Stacked$_{DT}$	All three features	93%	83%	90%
		COVID-Twitter-BERT with Stacked$_{SVM}$	All three features	97%	82%	93%

E. Benchmarking Performance of our Approach

We assessed the effectiveness of our stack ensemble model for detecting fake news by using all three datasets from Table VI. In the next step, we employed four baselines in two categories to compare our approach with state-of-the-art techniques. Three of these methods utilize vectorizer-based feature extractors, while the remaining method leverage transformer-based techniques. The baseline methods range from simple machine learning models (such as LinearSVM) to advance neural network models (such as BiLSTM). Table VIII shows the accuracy obtained by all the models applied to each dataset. We observe that our Stacked$_{\text{DT}}$ model (decision tree + random forest + logit. reg.) only performed better against one of the baselines (Al-Rakhami et al. [21]). Among the $Stacked_{\text{DT}}$ models, the TF-IDF-based Stacked$_{\text{DT}}$ model performed better than the BERT-based model on datasets 1 and 2.

Moreover, from our previous experiment result shown in Table V, we note that we achieved a similar high-performing result using SVM + Random Forest + Logic. Reg., which we name as $Stacked_{\text{SVM}}$. In Table VIII, the TF-IDF-based Stacked$_{\text{SVM}}$ performed best across all three datasets. We conjecture that SVM performs better in fake news detection tasks, as even the top three baselines include the SVM model. However, the $Stacked_{\text{SVM}}$ model requires significantly more training time than $Stacked_{\text{DT}}$. Thus, we kept the $Stacked_{\text{DT}}$ model for the final evaluation of our labeled dataset.

F. Performance Evaluation of Our Fake News Detection Model on our labeled dataset

Before conducting experiments, we split each of our labeled datasets 1-4 into 80% training and 20% testing samples, respectively. Table II lists the train and test sets. Table IX shows the performance comparison of our stack ensemble model evaluated on our four datasets (cf. Table II).

The values in Table IX represent precision, recall, and F1-score for *fake* class, as our primary objective is to detect fake tweets accurately. We are also interested in a higher recall value, which means that few fake tweets are misclassified as real information. Overall, we obtain the best precision, recall, and F_1 score values for *labeled data 3* when text features were extracted using TF-IDF. In the case of TF-IDF-based Stacked$_{\text{SVM}}$ and TF-IDF-based Stacked$_{\text{DT}}$ models, the precision values were 83% and 82%, respectively, while the recall values were 96% for both models. Moreover, TF-IDF outperformed or performed similarly to both BERT models across all labeled datasets.

COVID-Twitter-BERT performed better than BERT in the recall in the stacked model for all data; however, it did not outperform the standard BERT model in precision and F1-score. Since our work is specific to COVID-related, it leads to fewer contextual differences that BERT and COVID-Twitter-BERT learn. Hence, the pre-trained model (COVID-Twitter-BERT) did not significantly affect misinformation classification, as also identified in [31]. We also compared the execution time of TF-IDF, BERT, and COVID-Twitter-BERT for labeled data 3. TF-IDF took 443 seconds to execute, whereas BERT and COVID-Twitter-BERT took 1889 seconds and 2199 seconds, respectively. Consequently, TF-IDF was ≈ 4 times faster than BERT and ≈ 5 times faster than COVID-Twitter-BERT. Furthermore, the TF-IDF-based Stacked$_{\text{DT}}$ model is ≈ 2 times faster than the TF-IDF-based Stacked$_{\text{SVM}}$ model. Our result depicts that data labeled using weighted majority voting

TABLE IX: Performance of our proposed *model*. Here P, R, F_1, and FPR represent precision, recall, F_1 score, and false positive rate. The evaluation is performed across different datasets: **Regular Majority Voting**: Majority out of three supporting statement verdicts; **Weighted Majority Voting**: Voting based on rank and relevance score of supporting statement; **Without body text**: Supporting statement's title and verdict; **With body text**: Supporting statement's title, claim, body (up to 350 words), and verdict. **COVID-Twitter-BERT (C-BERT)** refers to BERT sentence embedding model pre-trained on COVID-related corpus. **Stacked$_{\mathbf{DT}}$** refers to ensemble model with Decision tree + Random forest + Logistic Regression; **Stacked$_{\mathbf{SVM}}$** refers to ensemble model with SVM + Random forest + Logistic Regression.

Labeled Data Description		Without body text						With body text					
		Regular Majority *(Labeled Data 1)*			Weighted Majority *(Labeled Data 2)*			Regular Majority *(Labeled Data 3)*			Weighted Majority *(Labeled Data 4)*		
		TF-IDF	BERT	C-BERT	TF-IDF	BERT	C-BERT	TF-IDF	BERT	C-BERT	TF-IDF	BERT	C-BERT
Stacked$_{DT}$	**Precision**	77%	77%	76%	77%	77%	76%	82%	81%	80%	81%	80%	79%
	Recall	98%	98%	99%	97%	98%	97%	96%	97%	98%	96%	97%	97%
	F1-score	86%	86%	86%	86%	86%	86%	89%	89%	88%	88%	88%	87%
	FPR	∼2%	∼2%	<1%	∼3%	∼2%	∼1%	∼4%	∼3%	∼2%	∼4%	∼3%	∼3%
Stacked$_{SVM}$	**Precision**	78%	77%	76%	78%	76%	75%	83%	81%	80%	82%	80%	79%
	Recall	96%	99%	99%	96%	98%	100%	96%	96%	97%	96%	95%	96%
	F1-score	86%	86%	86%	86%	86%	86%	89%	88%	88%	88%	87%	87%
	FPR	∼4%	∼1%	∼1%	∼4%	∼2%	<1%	∼4%	∼3%	∼3%	∼4%	∼5%	∼4%

(*labeled data 2 and 4*) has slightly low or equal precision as compared to regular majority voting (*labeled data 1 and 3 respectively*) in all models. This is because the weighted majority method labels fewer fake instances than the regular majority, as shown in Table II, thus reducing the number of indicators for detecting fake news. Finally, our second method of data collection - *with body text* consisting of the supporting statement's title, claim, body content, and verdict helped in increasing the precision and F1-score compared to the datasets *without body text*. As adding more information adds more context for fact-checking.

Our model only depends on the verified fact-checked statements from the fact-checking websites and not particularly on COVID-time period for the fake news detection. Thus, if COVID-related misinformation tweets persist on OSN even during non-peak COVID times, our model would work on it. Lastly, the model can be fine-tuned or retrained with other labeled non-COVID tweet data using our annotation model presented in §IV of the paper.

Ablation Analysis. We conducted an ablation analysis to understand the importance of the three features, i.e., tweet-level, user-level and textual features. The analysis is conducted on labeled data 3 as it achieves the best precision and F1-score, as shown in Table IX. First, we use all features and then remove components (features) to investigate the performance of our model. As shown in Figure 4, precision and F1-score are highest with all features. Removing tweet-level features only or user-level features only from all features results in no change in values. In contrast, removing textual features deteriorates precision and F1-score but increases recall when tweet-level and user-level features are considered together. This is confirmed when user-level and tweet-level features are considered individually as recall remains higher than with 'all features combined'. Therefore, user-level and tweet-level features are important, along with textual features, for improving recall. However, tweet-level features add little to the performance since bots show signs of mimicking real users by posting the same number of URLs, mentions and hashtags.

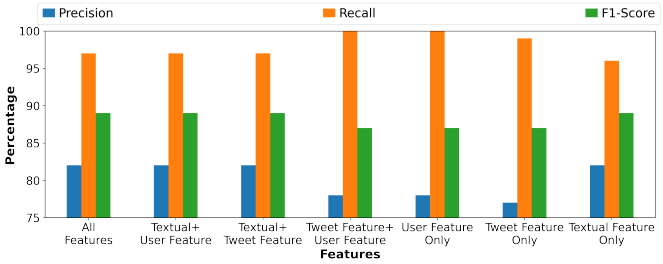

Fig. 4: Overview of Ablation analysis study.

VII. Bot Detection from Misinformation Tweets

False information is often intentionally disseminated by bad actors on OSN. Vosoughi et al. [32] point towards humans as being the source distributor of misinformation. However, it is evidenced by other research that misinformation is also disseminated by computational means via the usage of inauthentic profiles/fake accounts or bots working in a collaborative manner [11], [33], [34]. In this work, we do not examine the role of human profiles in misinformation dissemination and limit our scope to the role of bots. Therefore, this work aims to detect and analyze social bots spreading misinformation. To this end, we propose two hypotheses: **(h1)** bots have an active role in spreading misinformation, and **(h2)** bots are most active during misinformation campaigns (e.g., during peak COVID misinformation phase) compared to other times to maximize their goals. To analyze these two hypotheses, we investigate bots in OSNs using our COVID-19-related misinformation tweets.

TABLE X: Number of *bots'* generated tweets in four labeled.

Data	# Fake Tweets	# Bot-generated Tweets	%
Labeled Data 1	93,348	9,885	**∼11%**
Labeled Data 2	92,619	9,894	**∼11%**
Labeled Data 3	96,461	9,574	**∼10%**
Labeled Data 4	94,600	9,417	**∼10%**

Several studies [35], [36] on bot detection have used Botometer [37] (a state-of-the-art tool) to detect bots. Botometer takes over 1000 features (content, network, sentiment,

TABLE XI: Impact of *bots'* generated tweets in *labeled data 3*.

Data	Tweets	Retweet	Follower	Retweet/Follower	Likes	URLs	Avg. URLs	Mentions	Avg. Ment.	User	Verified
Bot Mis-Tweet	9,574	502,463	16,250,472	∼3%	44,686	6,756	0.67	6,060	0.63	4,474	24
Human Real-Tweet	25,618	1,615,275	422,529,672	<1%	380,810	17,229	0.70	16,337	0.63	17,085	1,031

TABLE XII: No. of *unique bot accounts* across different *bot score* threshold values.

Thresholds	# Uniq. Accounts	# Unique Accounts		
		Bot	Human	Deleted/Suspended
>= 0.5	**5,315**	**3,279 (62%)**	**1,941 (37%)**	**95 (∼2%)**
>= 0.6	2,680	1,796 (67%)	838 (31%)	46 (∼2%)
>= 0.7	1,316	936 (71%)	361 (27%)	19 (∼1%)
>= 0.8	526	336 (64%)	184 (35%)	6 (∼1%)
>= 0.9	103	54 (52%)	48 (47%)	1 (∼1%)
# Tweet Sample	125,709	122,314		
# Unique Users	69,665	68,377		

user features) to calculate a bot score probability between 0 (human) and 1 (automated). However, Botometer's bot score computation is time-consuming and not scalable for large datasets. Due to Twitter API rate limits, we use Botometer Lite [38], which is a variant of Botometer that takes in fewer features than Botometer while correlating strongly with Botometer bot scores. BotometerLite relies only on features extracted from user profile metadata contained in the tweet's JSON object when it was extracted.

Presence of Bots. To commence detecting bots in OSNs, we must establish the threshold value for the bot score. Prior bot detection studies have used various thresholds like 0.7 and 0.8 [39], while 0.5 is the more common choice [10], [11], [32], [40]. Table X shows the % of bots in the four labeled datasets with 0.5 as the threshold. In all four labeled datasets, bots generated approximately 10% misinformation tweets.

Impact of Bot-generated misinformation tweets. To assess the impact of bot-generated misinformation tweets (for threshold≥0.5 & verdict=fake), we used labeled data 3. As shown in Table XI, 4,474 unique bot accounts disseminate misinformation tweets to their 16 million followers and achieved more than 0.5 million retweets, thus displaying a high engagement rate (retweets/followers) of ∼3%. In contrast, human-generated real tweets (for threshold<0.5 & verdict=real) have an engagement rate of <1%. This result confirms that "bots are more reachable and visible to other users" [41]. Moreover, only <1% bot profiles were verified as compared to human profiles (users spreading real tweets) that have ∼6% verified profiles. This shows bot profiles are less likely to be verified. Furthermore, the average count of URLs and mentions in a misinformation tweet is similar to human users' tweets. This indicates that bot users mimic the behavior of human users. Therefore, all of these indicators provide evidence to confirm our hypothesis **(h1)** that ***bots play an active role in spreading misinformation in online social networks.***

Further, we assess the bot behavior from two periods (during and post-peak COVID misinformation phase). For this, we used 125,709 tweets that were crawled from Jun-Aug 2021 for labeling. These have collectively been tweeted by 69,665 unique users. As of Aug. 2022, 122,314 (∼97%) tweets from the original set are still available (3,395 (∼3%) tweets are either deleted by users or removed by Twitter) and have been tweeted by 68,377 unique users. Table XII compares bot statistics from Jun-Aug 2021 and Aug. 2022. With the threshold of 0.5 ($botscore >= 0.5$), 5,315 (out of 69,665) accounts were identified as bots during Jun-Aug 2021. By August 2022, out of the 5,315 bot accounts, 3,279 were still acting as bots, whereas, 1,941 were acting as genuine OSN accounts ($botscore < 0.5$), and 95 user accounts have been deleted or suspended. It is evident from Table XII that now one-third of bot accounts display genuine user (human) activity in the August 2022 data. We can use this result to show that bots have the maximum motive to spread misinformation during crisis periods like the peak of COVID-19 compared to normal times. This change in bot's behavior affirms our hypothesis **(h2)** that ***bots are most active during their misinformation campaigns compared to other times***. As confirmed from our hypotheses (**h1**) and (**h2**), bots have played a significant role in promoting misinformation during COVID-19.

Future Work: Filtering tweets with a claim worth fact-checking can improve the categorization of the misinformation dataset. Claim identification or stance detection can provide a basis for such extensive filtering. Our future work will focus on these areas and evaluate our model's applicability in other misinformation domains. Moreover, we plan to increase the verified fact-check statements corpus by incorporating Google search results and other verified sources instead of depending on a few fact-checking sources [42], [43].

VIII. Conclusion

In this paper, we proposed an *annotation model* for creating large datasets using COVID-19 as a case study and a machine learning-based classifier, called the *ensemble stack model*, to detect fake news. Our model achieved a *precision* score of 83%, *recall* score of 96%, and a false positive rate of ∼4% when utilizing TF-IDF for extracting the tweet's textual features. Additionally, we provided evidence that bots play an active role in disseminating misinformation i.e., bots generate approximately 10% misinformation tweets. We also showed that bot behavior changes over time, depicting that bots are most active during their misinformation campaigns compared to other times.

References

[1] Y. Dou, K. Shu, C. Xia, P. S. Yu, and L. Sun, "User preference-aware fake news detection," in *SIGIR*, 2021.

[2] J. Zhang, R. Zhang, Y. Zhang, and G. Yan, "The rise of social botnets: Attacks and countermeasures," *TDSC*, vol. 15, no. 6, 2018.

[3] M. Ikram and et al., "Measuring, characterizing, and detecting facebook like farms," *TOPS*, vol. 20, no. 4, 2017.

[4] E. Ferrara, "What types of covid-19 conspiracies are populated by twitter bots?" *First Monday*, vol. 25, no. 6, May 2020.

[5] S. Yang and et al., "Analysis and insights for myths circulating on twitter during the covid-19 pandemic," *OJCS*, vol. 1, 2020.

[6] B. Sharma, I. Karunanayake, R. Masood, and M. Ikram, "Don't be a victim during a pandemic! analysing security and privacy threats in twitter during covid-19," *IEEE Access*, vol. 11, 2023.

[7] T. L. Sutejo and D. P. Lestari, "Indonesia hate speech detection using deep learning," in *2018 International Conference on Asian Language Processing (IALP)*, 2018.

[8] U. Bhattacharjee, P. Srijith, and M. S. Desarkar, "Leveraging social media towards understanding anti-vaccination campaigns," in *COMSNETS*, 2019.

[9] W. Y. Wang, ""liar, liar pants on fire": A new benchmark dataset for fake news detection," in *ACL*, 2017.

[10] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media," *Big data*, 2020.

[11] C. Shao and et al., "The spread of low-credibility content by social bots," *Nature Comm.*, 2018.

[12] N. Vo and K. Lee, "Where are the facts? searching for fact-checked information to alleviate the spread of fake news," in *(EMNLP)*, 2020.

[13] W. S. Paka, R. Bansal, A. Kaushik, S. Sengupta, and T. Chakraborty, "Cross-sean: A cross-stitch semi-supervised neural attention model for covid-19 fake news detection," *Applied Soft Computing*, vol. 107, 2021.

[14] A. X. Zhang and et al., "A structured response to misinformation: Defining and annotating credibility indicators in news articles," ser. WWW, 2018.

[15] Y. Wang and et al., "Weak supervision for fake news detection via reinforcement learning," in *AAAI*, 2019.

[16] A. Bonet-Jover, "Semi-automatic annotation proposal for increasing a fake news dataset in spanish," 2021.

[17] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," in *ACL*, 2018.

[18] P. Patwa and et al., "Fighting an infodemic: Covid-19 fake news dataset," in *IWCOHP*, 2021.

[19] H. Ahmed, I. Traore, and S. Saad, "Detection of online fake news using n-gram analysis and machine learning techniques," in *ISDSDCE*, 2017.

[20] Y. Wang, Y. Zhang, X. Li, and X. Yu, "Covid-19 fake news detection using bidirectional encoder representations from transformers based models," *arXiv preprint arXiv:2109.14816*, 2021.

[21] M. S. Al-Rakhami and A. M. Al-Amri, "Lies kill, facts save: Detecting covid-19 misinformation in twitter," *IEEE Access*, vol. 8, 2020.

[22] A. Thakur, S. Shinde, T. Patil, B. Gaud, and V. Babanne, "Mythya: Fake news detector, real time news extractor and classifier," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, 2020, pp. 982–987.

[23] M. Aldwairi and A. Alwahedi, "Detecting fake news in social media networks," *PCS*, vol. 141, 2018.

[24] J. M. Banda and et al., "A large-scale covid-19 twitter chatter dataset for open scientific research-an international collaboration," *Epidemiologia*, 2021.

[25] twarc, "twarc," 2021. [Online]. Available: https://twarc-project.readthedocs.io/en/latest/

[26] A. Depoux, S. Martin, E. Karafillakis, R. Preet, A. Wilder-Smith, and H. Larson, "The pandemic of social media panic travels faster than the covid-19 outbreak," p. taaa031, 2020.

[27] C. M. Rivers and B. L. Lewis, "Ethical research standards in a world of big data," *F1000Research*, vol. 3, 2014.

[28] M. Müller, M. Salathé, and P. E. Kummervold, "Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter," *arXiv preprint arXiv:2005.07503*, 2020.

[29] K. Hayawi and et al., "Anti-vax: a novel twitter dataset for covid-19 vaccine misinformation detection," *Public Health*, vol. 203, 2022.

[30] S. A. Memon and K. M. Carley, "Characterizing COVID-19 misinformation communities using a novel twitter dataset," in *CIKM*, 2020.

[31] J. P. Wahle and et al., "Testing the generalization of neural language models for covid-19 misinformation detection," in *ICI*, 2022.

[32] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aap9559

[33] A. Arif, L. G. Stewart, and K. Starbird, "Acting the part: Examining information operations within# blacklivesmatter discourse," *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, no. CSCW, pp. 1–27, 2018.

[34] K. Starbird, A. Arif, and T. Wilson, "Disinformation as collaborative work: Surfacing the participatory nature of strategic information operations," *Proc. ACM Hum.-Comput. Interact.*, vol. 3, no. CSCW, nov 2019. [Online]. Available: https://doi.org/10.1145/3359229

[35] C. Grimme, D. Assenmacher, and L. Adam, "Changing perspectives: Is it sufficient to detect social bots?" in *ICSCSM*. Springer, 2018.

[36] D. Assenmacher, L. Clever, J. S. Pohl, H. Trautmann, and C. Grimme, "A two-phase framework for detecting manipulation campaigns in social media," in *ICHCI*, 2020.

[37] M. Sayyadiharikandeh, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, "Detection of novel social bots by ensembles of specialized classifiers," in *CIKM*, 2020.

[38] K.-C. Yang and et al., "Scalable and generalizable social bot detection through data selection," in *AAAI*, 2020.

[39] R. J. Schuchard and A. T. Crooks, "Insights into elections: An ensemble bot detection coverage framework applied to the 2018 u.s. midterm elections," *PLOS ONE*, 2021.

[40] K.-C. Yang, E. Ferrara, and F. Menczer, "Botometer 101: Social bot practicum for computational social scientists," *Journal of Computational Social Science*, pp. 1–18, 2022.

[41] J. Pastor-Galindo and et al., "Profiling users and bots in twitter through social media analysis," *IS*, vol. 613, 2022.

[42] N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, V. Sable, C. Li, and M. Tremayne, "Claimbuster: The first-ever end-to-end fact-checking system," *Proc. VLDB Endow.*, vol. 10, no. 12, p. 1945–1948, aug 2017. [Online]. Available: https://doi.org/10.14778/3137765.3137815

[43] H. Hammouchi and M. Ghogho, "Evidence-aware multilingual fake news detection," *IEEE Access*, vol. 10, pp. 116 808–116 818, 2022.