# DeSI: Deepfake Source Identifier for Social Media

Kartik Narayan*, Harsh Agarwal*, Surbhi Mittal, Kartik Thakral, Suman Kundu, Mayank Vatsa, Richa Singh
IIT Jodhpur, India
{narayan.2, agarwal.10, mittal.5, thakral.1, suman, mvatsa, richa}@iitj.ac.in

## Abstract

*Social media holds the power to influence a significant change in the population. Through social media, people all around the world can connect and share their views. However, this social space is now infected due to the infiltration of fraudulent, obscene, fake and possibly, influential media. According to a UNESCO report, prevalence of fake news and deepfake content possess the potential of spreading fake propaganda and can lead to political and social unrest. Trust on social media is an emerging problem and there is an urgent need to address the same. There has been some research around approaches that detect fake news and deepfakes, however, identification of the source of these deepfakes posted on social media platforms is an equally important but relatively unexplored challenge. This paper proposes a novel Deepfake Source Identification (DeSI) algorithm that identifies the sources of deepfakes posted on Twitter. The proposed DeSI algorithm allows for two input modalities - text and images. We rigorously test our algorithm in both constrained and unconstrained experimental setups and report the observed results. In the constrained setting, the algorithm correctly identifies all the deepfake tweets as well their sources. The complete framework is further encased in a web portal to facilitate intuitive use and analysis of the results.*

## 1. Introduction

The advent of smartphones and the rise of social media applications have made digital photographs and videos more prevalent in recent decades. According to myriad of reports [31], about three billion photos and 7,20,000 hours of video are shared on the internet every day. Hence, social media platforms have become a go-to source of daily information and news. One such platform with a significant active user base is Twitter. More specifically, Twitter has established its place as a notable source of news, typically spreading information in a relatively short time when compared to traditional media. Also, Twitter being a decentralised and dynamic platform - where anyone is free to share content results in information spreading quickly. This has resulted in a colossal rise of image forgery techniques to rapidly spread erroneous information.

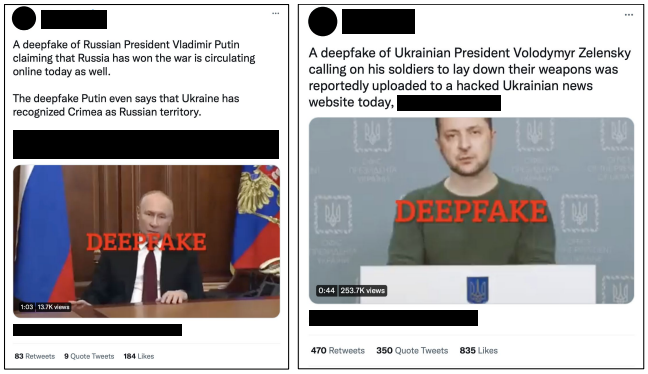


Figure 1. The prevalence of deepfakes on social media platforms like Twitter.

The generated fake content is fabricated in such a way that it appears to be a legitimate piece of information. Such content may focus on the trending topics so as to accelerate its spread and lure more folks into following false propaganda which is detrimental for the society. Fake news has the potential to sway public opinion and is widely disseminated for monetary and sociopolitical gain as described through Figure 1. The creation and dissemination of deepfake multimedia on social networking platforms has already led to a huge distress in politics and spread of obscene content. The term "Deepfake" is a careful combination of the words deep and fake, i.e. , it pertains to fake AI-generated multi-modal content such as videos, pictures and audios. If not contained timely, such multimedia content can spread and give rise to a narrative far from the truth. Despite the fact that rumours are typically started by a handful of individuals, it is difficult to determine who is the originating source of the posted media. This is an important problem because once the sources of rumours are identified, essential links for rumour propagation can be cut off thereby controlling the spread at an

*Equal contribution

early stage. This can help the law enforcement authorities to curtail the spread of misinformation on social media and keep vigil on notorious social media accounts.

In this work, we present a novel Deepfake Source Identification (DeSI) algorithm that helps in detecting the source of tweets containing deepfakes filtered by a particular query input. The DeSI algorithm supports two input modalities- image and text. This framework is embedded into a web portal where the user can input a query text or image alongwith a specified number of tweets which are fetched from Twitter. From the pool of the extracted tweets, we filter out the tweets containing deepfake media (image or video) corresponding to the queried text which are also similar to the input image queried by the user. In the absence of such an image, all the tweets containing deepfake media corresponding to the queried text are extracted. After filtering, the DeSI algorithm identifies the possible origin of the tweets. Further, we plot an interactive directed graph showing the network of tweets. It gives a temporal insight into the spread pattern and also identifies the volatile nodes in the network. The volatility of nodes is predicted through a *Retweet Proneness model* by estimating the possible retweet count per minute for every filtered tweet.

The paper is organised as follows- Section 2 summarizes the related work in the field of deepfake detection and its source identification along with retweet proneness. Section 3 describes the various components of the DeSI algorithm. In Section 4, we report the designed experimental setup (Section 4.1), the implementation details (Section 4.2), and finally, the performance of our algorithm (Section 4.3).

## 2. Related Work

Researchers have explored the problem of identifying the source of misinformation, estimating the rate of retweet, and detecting deepfakes individually. In this section, we discuss the existing research in these fields.

**Detecting Fakes on Social Media:** Recently, misinformation campaigns have resulted in widespread hysteria among masses [18]. It has been noted that such misinformation campaigns influenced the 2016 US presidential election [26]. Some of these fakes heavily disfigure the facts, and target and demonize celebrities on social media. Such fakes on social media can be propagated mainly through two media: Textual-fakes and Deepfakes.

**Textual media:** Fagni et al. [8] presented the first dataset of deepfake texts extracted from Twitter consisting of tweets from 23 bot accounts each mapped to one of the 17 human accounts they imitated. It contains machine made short texts from a variety of text generative models thus, enabling researchers to investigate the generalizability of their detection algorithm. They demonstrated that the RoBERTa [21] detector performs exceptionally well and is generalizable for all generative models.Kar et al. [11] proposed an architecture built over mBERT [6] for multiple Indian languages so as to identify fake tweets related to COVID-19 on Twitter. For extending their approach to other Indian languages as well, the authors propose a zero-shot learning model which achieved a state-of-the-art performance for Hindi and Bengali languages. Konkobo et al. [12] presented a semi-supervised learning approach by employing features such as users' replies, network, and their credibility which can handle unmarked label on social media. They focused on building a network that detects the spread of misinformation on social media at an early stage. Their model demonstrated promising results on Politifact and Gossipcop.

**Deepfakes:** One of the prime and most impactful source of misinformation are deepfakes. They are becoming easier to generate and share on social media platforms. Recently, a lot researchers are focusing on detecting deepfakes. Afchar et al. [1] introduced a dual CNN based approach for face forgery detection that demonstrated 98% and 95% detection rate on deepfake and Face2Face [30] videos, respectively. The first network (Meso-4) consists of four convolution and pooling layers followed by a dense hidden layer. The second network (MesoInception-4) is built on a variant of inception module which includes dilated convolutions. Li and Lyu [20] proposed a Deep NN for detecting deepfakes by observing the artifacts (around the facial region) introduced during face warping in deepfake generators. Their approach was tested on UADFV [34] and Deepfake TIMIT datasets [13] and outperformed the state-of-the-art methods for those datasets. In another work, Li et al. [19] proposed a method based on eye-blinking in humans that has a specific duration and frequency which is not present in deepfake videos. They built an architecture based on long term recurrent network which identified the temporal irregularity in eye blinking sequences. Nirkin et al. [24] hypothesized that deepfake generation approaches result in discrepancies between faces and their context (hair, eyes, etc.). They proposed a novel architecture based on dual XceptionNet based networks for face and context recognition. Their approach utilizes recognition signals from the aforementioned networks for detecting discrepancies. Kumar et al. [16] divided the facial region into patches which were then fed to parallel ResNet18 models for detecting face manipulated videos. Jain et al. [10] presented a novel framework based on a sequential three level approach that accurately differentiates real vs manipulated photos, retouches vs GAN generated photos and lastly the GAN approach employed. Agarwal et al. [2] devise a novel computationally efficient algorithm in order to identify digital presentation attacks which achieves lower error rates across various databases, attack types and generative models.

**Source Identification:** Researchers have explored and developed various algorithms to identify the source of a given fake news which forms the crux of rumor detection in so-

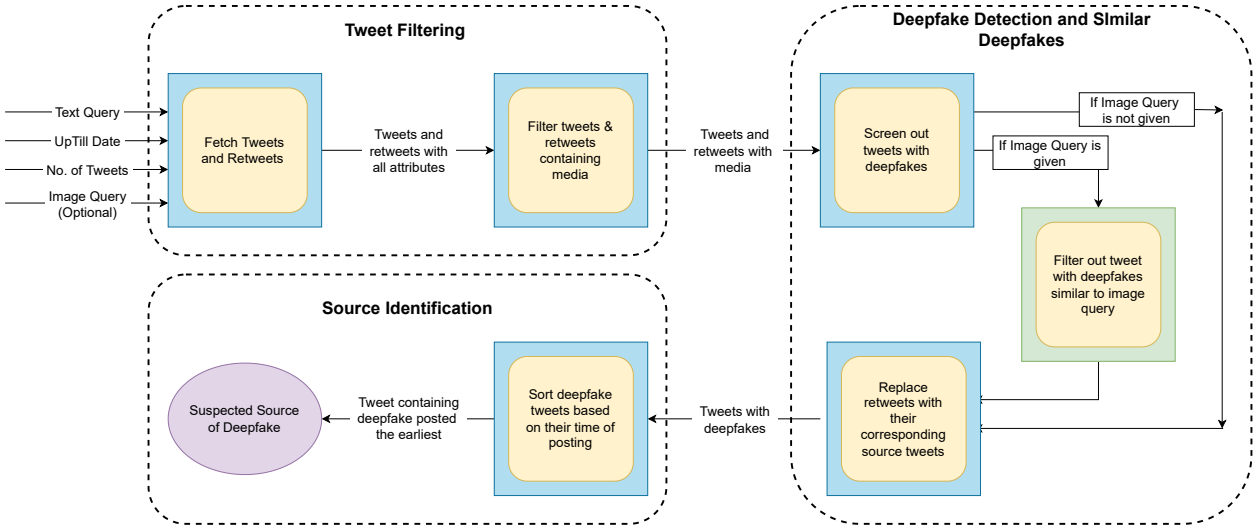


Figure 2. Step-by-step description of the proposed DeSI Algorithm.

cial networks. This is particularly helpful in limiting the harm caused by them by separating the sources from the network. Krol et al. [15] proposed a two-fold algorithm for detecting the initial possible sources in a social networking website. Their algorithm was able to identify potential rumourmongers within a large network with more ease as compared to a smaller and more compact network. Dong et al. [7] proposed a Graph Convolutional Network (GCN) based rumour source detection model that identifies numerous sources of incorrect information which requires no foreknowledge of the propagation model that underlies it. However, it not possible to establish single source identification problem on their proposed idea. Zhou et al. [36] observed the snapshots of graph topology and tracked the source by classifying nodes into Susceptible, Exposed, Infected and Recovered by employing the proposed SEIR model. They proposed a source probability estimator based on Optimal-infection Processes (OP) model and demonstrated better results than traditional centrality heuristic measures.

**Retweet Proneness:** When an existing tweet is re-shared by a user, it is termed as a retweet. The propagation effectiveness of a twitter post is directly proportional to number of times it has been shared, i.e., retweeted [25] . This has been a long-standing topic of interest for many researchers. Nesi et al. [23] utilized different attributes associated with a tweet and author profile to predict the probability of retweet and listed them in terms of their impact. They demonstrated that CART based decision tree model with recursive partitioning outperformed other traditional ensemble models like Random Forest and Stochastic Gradient Boosting. Kumar et al. [17] proposed a novel Forest Fire-based algorithm to model the information diffusion process in online social networks. They computed a similarity score and Topic Significance metric for user accounts as a weighted summation of user attributes. Wang et al. [32] showed that apart from text, multi-modal media also has an impact on the popularity of a post. They utilise a joint embedding model trained under a bidirectional ranking loss to integrate the tweet text and embedded images. Thereafter, a joint embedding space of the learnt features and user-based social features is computed which is then fed to the Poisson regression model. Kowalczyk et al. [14] highlighted that to elude frequent trade-offs between accuracy, scalability, and privacy, precise alignment of data collection, management, and analysis algorithms is required. They present a new approach for acquiring massive datasets, high-accuracy supervisory signals, and multi-language emotion estimation while satisfying all applicable privacy requests in their article. Further, a unique gradient boosting approach is proposed to achieve state-of-the-art outcomes in virality rating.

In the literature, a lot of approaches have been proposed that separately focus on identifying the source of a fake news, estimating the spread of a tweet or detecting if a video is deepfake or not. In this work, we explore the problem of identifying the source of a detected deepfake on Twitter and estimate its virality on Twitter.

## 3. Proposed DeSI Algorithm

In this section, we discuss the proposed DeSI algorithm for source identification of deepfakes on the Twitter platform. The DeSI algorithm, as shown in Figure 2, can be viewed as a filtering process where in each step we discard tweets which cannot be the sources of deepfake media. The steps of extracting tweets and discarding the tweets with solely textual content form a part of *Tweet Filtering* (Section 3.2). The next step of *Deepfake Detection and Similar Deepfakes*

Table 1. List of tweet attributes provided by the Twitter API which are used for extracting relevant tweets.

| | | |
|---|---|---|
| created at | retweet count | in reply to screen name |
| full text | favorite count | in reply to user id str |
| text | quoted status | in reply to status id str |
| user | quote count | retweeted status |
| lang | favorited | extended entities |
| id | retweeted | in reply to status id |
| source | reply count | quoted status id str |
| place | filter level | possibly sensitive |
| coordinates | is quote status | in reply to user id |
| entities | matching rules | quoted status id |
| id str | truncated | |

corresponds to the identification of deepfake media and filtering for similar deepfake content based on an input query (Section 3.1). The input query can be given in the form of text or an image. In case of the text modality, a text query is provided and all the deepfakes available with that hashtag are fetched[1]. In case of an image-based query, an additional input in the form of an image is given which is used to filter for suspected sources responsible for posting similar deepfakes. Lastly, reduction of the possible suspects by locating the source tweets of all retweets and identifying the source from the *Source Identification* (Section 3.2). The complete DeSI algorithm returns a list of tweets, suspected to be the source of deepfake with an estimation of its virality on Twitter. The approach to determine the possibility of a tweet to go viral by estimating its retweet proneness is described in Section 3.3.

### 3.1. Deepfake Detection and Similar Deepfakes

A deepfake detection algorithm is required in order to filter the tweets containing deepfake images/videos. For this, we employ a deep learning based detection model $f$ with parameters $\theta$. The model $f_\theta$ takes an image or extracted frame $x$ as input and provides a confidence score between zero and one as described below:

$$score = \arg\max[\sigma(f_\theta(x))] \tag{1}$$

where $\sigma$ denotes the softmax activation function. A score value closer to one implies that the model is highly confident in its prediction of the image to be fake. A value closer to zero implies high confidence for the image to be *not* fake (i.e. real). Based on a chosen confidence threshold, the input image $x$ is deemed to be real or a deepfake. The same model $f_\theta$ is employed in case of deepfake detection in videos. However, a fixed number of frames are extracted from the video, and model prediction over these frames is aggregated for the final decision.

In order to extract deepfakes from Twitter using an image, we employ another deep learning model $g$ with parameters $\phi$ that provide meaningful feature representations for matching. The deepfakes fetched from Twitter may be in the form of images or videos. If an input image $x_{in}$ is uploaded by the user on the web portal, all relevant tweets containing a deepfake image $x_{image}$ or video $x_{video}$ similar to $x_{in}$ should be filtered for source identification. The similarity $isim(,)$ between $x_{in}$ and $x_{image}$ is computed using cosine similarity.

$$isim(x_{in}, x_{image}) = cosine\_sim(x_{in}, x_{image}) \tag{2}$$

$$= \frac{g_\phi(x_{in}) \cdot g_\phi(x_{image})}{\|g_\phi(x_{in})\| \|g_\phi(x_{image}))\|} \tag{3}$$

In case of video, $n$ frames are extracted from $x_{video}$ and the similarity $vsim(,)$ is calculated as follows,

$$vsim(x_{in}, x_{video}) = \sum_{i=0}^{n} \frac{isim(x_{in}, x^i_{video})}{n} \tag{4}$$

where, $x^i_{video}$ represents the $i^{th}$ frame of the video. After calculating the similarity values, the relevant deepfake tweets are selected based on a similarity threshold value.

### 3.2. Tweet Filtering and Source Identification

The source identification algorithm begins with extracting tweets through the Twitter API based on two input modalities- *text* and *images*. For both the modalities, queries such as "date up till when the tweets need to be extracted", and "the number of tweets" are given as inputs. For *text*, a text query is given as an input and the algorithm screens for relevant tweets containing deepfakes. The text query may contain hashtags for better filtering of tweets. In case of *images*, an additional image query along with the text query is provided as input. In both cases, the API fetches the tweets along with a set of attributes. Table 1 tabulates the list of attributes fetched from the Twitter API. The next step involves screening of the tweets for media entity such as images or videos. All tweets having textual content only are discarded as the algorithm focuses on finding the source of tweets with deepfake images or videos. The media from the remaining tweets are extracted and stored. For both *text* and *images*, the deepfake model $f_\theta$ is used to filter out the deepfakes from the already screened tweets with media. For *images*, the algorithm screens out tweets with deepfakes similar to the queried image. In addition to $f_\theta$, we use an image similarity model $g_\phi$ for image-to-image and image-to-video matching. The working of both the models is explained in Section 3.1.

[1]The extracted tweets are limited by the restrictions on the Twitter Academic API. The restrictions are further described in *Implementation Details*.

**Algorithm 1:** Proposed DeSI algorithm

**Input:** Text query ($T$), Number of tweets to be extracted ($N$), Image query ($I$)
**Parameters:** $tweets\_list$, $final\_list$, $source\_list$, $sim\_thresh$
**Ensure:** $len(T) > 0$ and $N > 0$
$tweets\_list \leftarrow$ Extract $N$ tweets using the Twitter API using $T$.
**if** *$I$ is None* **then**
  **for** *tweet in $tweets\_list$* **do**
    **if** *the media in tweet is deepfake* **then**
      $final\_list \leftarrow tweet$
    **end**
  **end**
**else**
  **for** *tweet in $tweets\_list$* **do**
    **if** *the media in tweet is deepfake AND* $sim(media, I) > sim\_thresh$ **then**
      $final\_list \leftarrow tweet$
    **end**
  **end**
**end**
**for** *each tweet in $final\_list$* **do**
  **if** *tweet is a retweet* **then**
    Query the tweet using Twitter API for $source$.
    $source\_list \leftarrow source$
  **end**
**end**
Sort the $source\_list$ using timestamp of tweets.
**return** the first element of sorted $source\_list$

The list of all the tweets with relevant deepfakes is then used for further processing. The list which consists of tweets and retweets is converted into a list with only tweets. This is achieved by identifying the retweets and replacing them with their respective source tweets. The source tweets for the retweets are fetched by making requests to the Twitter API using the retweet ID. It may so happen that multiple users retweet the same tweet. In order to avoid repetition, we keep only the unique source tweets and discard the repeated entries. This provides a pool of tweets and their corresponding sources, which have deepfake media content. Finally, we sort these source tweets based on their time of posting. The tweet which is posted the earliest is the source tweet of that particular media content posted on the Twitter platform, and the user of the account from which it was posted is the suspected source. The deepfake source identification algorithm has been summarized in Algorithm 1.

### 3.3. Retweet Proneness Estimation

In this section, we describe the algorithm used for retweet proneness estimation of a particular tweet. This estimation acts as a proxy for virality of tweets. In order to train the regressor, we utilize features from each user profile. The profile of a user comprises of four incumbent attributes namely *friends count*, *followers count*, *favourited tweet count* and the *account age (in seconds)*. These attributes quantify the influence of the user on the social network and helps to compute the retweet proneness. The dataset $D$ with $m$ samples is defined as $D = \{(r_1, y_1), (r_2, y_2), ..., (r_m, y_m)\}$ where $(r_i, y_i)$ represents the input feature vector $r_i$ with its corresponding ground truth $y_i$. We employ a Random Forest Regressor [3] $\hat{h}_{rf}^{B}$ with $B$ random trees with "retweets per minute" as the target variable. The regressor is defined as,

$$\hat{h}_{rf}^{B} = \frac{1}{B} \sum_{b=1}^{B} T_b(r) \tag{5}$$

where, $T_b$ is a random tree in ensemble of trees $\{T_b\}_1^B$ and provides the predicted retweets per minute for a given input vector $r$. The model $\hat{h}_{rf}^{B}$ [3] is trained using a Mean-Squared Error (MSE) Loss for $m$ samples available in the training set as described below:

$$MSE = \frac{1}{m} \sum_{1}^{m} (\hat{h}_{rf}^{B}(r_i) - y_i)^2 \tag{6}$$

## 4. Experimental Design, Results and Key Observations

To validate the performance of the proposed DeSI algorithm, we design two experiments on the Twitter social media platform.

### 4.1. Experimental Setup for Source Identification of Deepfakes

The first experiment is performed in a **constrained setting**, where we create five anonymous Twitter users. We post a total of 70 tweets and 90 retweets through these user accounts, cumulatively. Out of the 70 tweets, 10 tweets are purely text, 30 tweets contain real untampered videos, and 30 tweets contain deepfake videos. The real and deepfake videos are taken from the test set of the FaceForensics++ dataset [28]. We use 5 real and 5 deepfake videos which are tweeted (and thereafter, retweeted) multiple times from different users leading to a total of 70 (and 90 retweets). By keeping track of the source while designing the experiment, the results of the DeSI algorithm are validated. All the tweets are posted with the hashtag "sourceidentification-experiment". We employ two protocols for evaluation with text and image query, respectively. In case of **text query**, all the deepfakes corresponding to the input text query are extracted, and the corresponding source is identified. For the **image query** setting, we employ 6 query images. These query images are used to match the media extracted through a particular text query. Among the similar deepfake media,

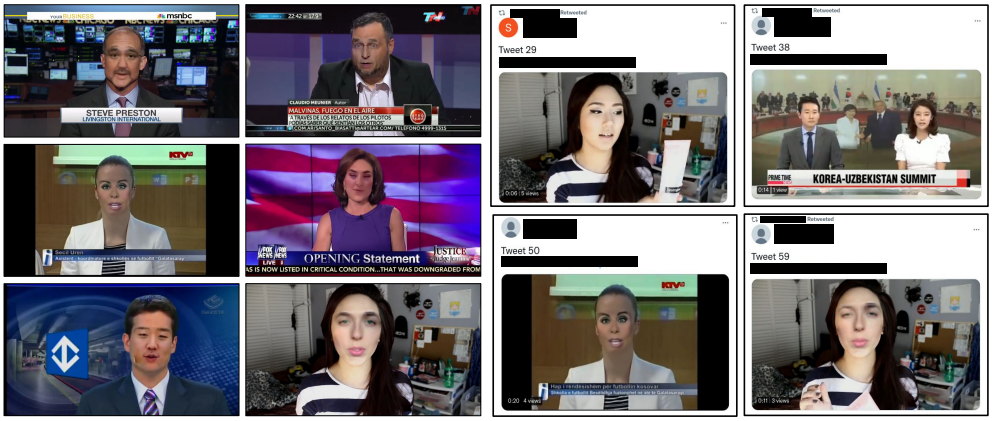


Figure 3. The experimental design for source identification. (a) Set of 6 deepfake query images, and (b) A sample of posted tweets and retweets. The first row contains real media, while the second row contains deepfake videos.

the source tweet is identified. The constrained experimental design is depicted in Figure 3.

We next evaluate the proposed DeSI algorithm in an **unconstrained setting** by fetching 10,000 tweets using the input text query "#russia #ukraine". However, in this scenario, the ground truth for source of deepfakes are not available. There is no database of queries/hashtags which have similar deepfakes which makes the validation infeasible. Therefore, we manually validate the extracted tweets and their sources to be deepfakes assisting the testing for real-time performance of the DeSI algorithm. Providing an additional image query enables us to filter out tweets containing deepfakes similar to the input query image. This can be extremely beneficial for law enforcement agencies to filter the source tweet of a deepfake considerably faster and with better accuracy.

**Retweet Proneness Estimation:** In order to estimate the possibility of retweet, we employ a Random Forest Regressor which computes the prediction on the basis of the user profile. The Random forest regressor is trained and evaluated using the user data of 20,000 tweets. These tweets are extracted by using 20 different hashtags with 1000 tweets per hashtag. These hashtags are: *christmas, winters, mumbai, india, newyear, omicron, love, nature, ViratKohli, ThursdayThoughts, Bollywood, Hollywood, Taiwan, sunset, niki, US, football, 2021*, and *travel*. The tweets extracted are processed and split into 70-30 ratio for training and testing sets, respectively. The Random Forest Regressor is trained on 14000 tweets and tested on 6000 tweets.

### 4.2. Implementation Details

The details of implementation for the different components of the proposed DeSI algorithm have been described below.

**Twitter API:** We have used the python library *Tweepy* [27] to access the Twitter API. Twitter Academic API is used to make requests to the Twitter Platform. It helps in collecting real-time and historical public data. The API has a limit of 50 requests per 15 minutes when using the search functionality and the relevant tweets are extracted based on the provided query. There is no guarantee that we obtain an exhaustive list of all the tweets as per the given query. However, the Twitter Premium API does not suffer from these limitations. We choose Twitter over other social media platforms as it is among the prime medium for news distribution and celebrities as well politicians are active users of this platform. Additionally, it has an extensive API support when compared to other social media platforms.

**Image Similarity:** To compute the similarity between the input image and extracted deepfake images/videos, we use a LightCNN-29 model [33] pre-trained on the MS-Celeb-1M dataset [9] for feature extraction. Before extracting features from the images (or, video frames), facial regions are cropped from the image using a face detection algorithm. For this, we use the MTCNN model [35]. To compute image to video similarity, we extract 10 frames from the video, and take the mean of similarity scores obtained using the 10 frames as described in the previous section. A similarity threshold of 0.4 is used for matching.

**Deepfake Detection:** We employ a deep learning network namely XceptionNet [4] for deepfake detection. The XceptionNet network has been shown to provide state-of-the-art results for the problem of deepfake detection [4]. This model is trained using high-resolution videos of the FaceForensics++ dataset [28] which is a popular large-scale deepfake dataset. The dataset consists of four manipulation techniques- Deepfakes [5], Face2face [30], FaceSwap [22], and Neural Textures [29]. For training, 10 frames are extracted from each video followed by face detection align-

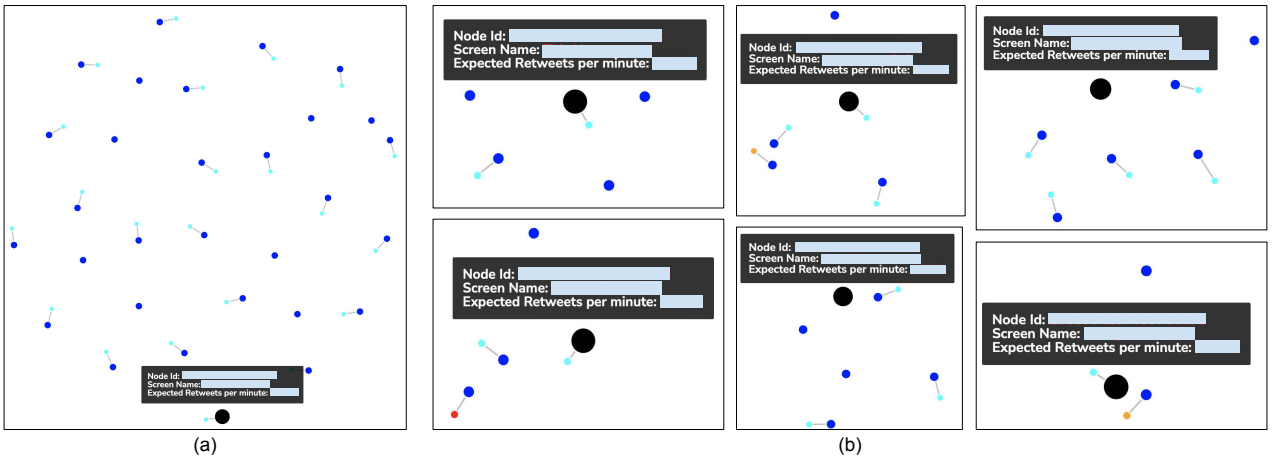


Figure 4. Results of the DeSI algorithm in the constrained setting (a) with text query. The network represents nodes(tweets) with deepfakes relevant to the text query, (b) with image query using 6 different images as queries. The nodes (tweets) in the network contains deepfakes similar to the given image query. The nodes provide additional information about the tweet like "Tweet ID", "User Screen Name" and "Expected Retweets per minute".

| Color | Range of Time(in seconds) of Retweet after Source tweet | Significance |
|---|---|---|
| Red | 2403. - 2432.4 | Volatile |
| Orange | 2432.4 - 2461.8 | Spreading |
| Yellow | 2461.8 - 2491.2 | Susceptible |
| Green | 2491.2 - 2520.6 | Calm |
| Cyan | 2520.6 - 2550. | Non-Volatile |

Figure 5. Color correspondence of the nodes in the network graphs (Best viewed in color).

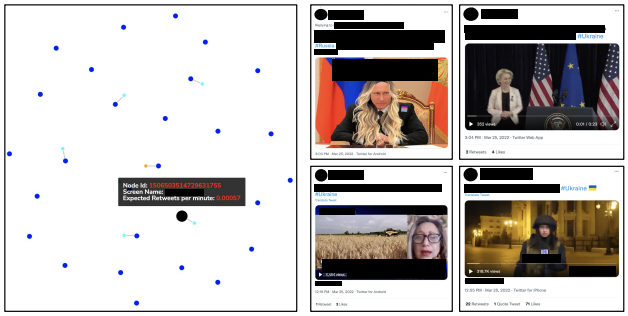


Figure 6. Results of DeSI algorithm in the unconstrained setting (a) The network graph for 10,000 tweets obtained with the query "#russia #ukraine". The nodes represent tweets with deepfakes along with additional information about the tweet like "Tweet ID", "User Screen Name" and "Expected Retweets per minute". (b) Deepfakes in some nodes (tweets) obtained in the network graph (Best viewed in color).

ment using MTCNN [35]. The model is trained using the Binary Cross-Entropy loss function for 30 epochs using the Adam optimizer. This model is trained on a Nvidia DGX station consisting of four V100 GPUs.

**Retweet Proneness:** For training the random forest regressor, we use the mean-squared error with 100 estimators and a min-sample split of 2.

### 4.3. Results and Analysis

In this section, we discuss the results obtained for the proposed DeSI along with the retweet proneness estimation based on the experimental setup described above.

**Results of Source Identification:** In case of source identification, the results are obtained in the form of possible sources of deepfake tweets. Additionally, we plot a network graph where the nodes correspond to a unique tweet. The suspected source of a tweet with deepfake media is reported and is highlighted in the network graph by enhancing its size and colorng it in black as shown in Figure 4(a) and (b). Other attributes such as "Node ID", "Screen Name" and "Expected Retweets per minute" are recorded for every node in the network graph. The nodes in the graph are color-coded as shown in Figure 5 describing the significance of a tweet depending on the range of time (in seconds) of retweet after source tweet. The red color signifies that retweet to a tweet happens in very short time relative to others after posting the source tweet. Therefore, the node is volatile and is more probable to go viral. In contrast, the cyan color shows that the node is non-volatile and there is very little chance of the tweet to go viral. Node colors like orange, yellow and green depict the probable virality state as spreading, susceptible and calm, respectively. The edge length between the nodes in the network corresponds to the min-max scaled version of the difference of time in seconds of retweets and

their respective tweets.

For the **constrained setting**, the results are presented in Figure 4. For queries with **text**, we search the Twitter platform for the presence of unique text query and run the source identification algorithm on the extracted tweets. We observe that the predicted Tweet ID of suspected source matches with the actual source Tweet ID which is recorded while designing the experiment. Hence, we conclude that the source identification algorithm delivers the source tweet with deepfake media when queried for a particular hashtag. Similarly, in case of queries with **images**, we observe that the network graph contains nodes (tweets) with deepfake media similar to the queried image. We use 6 query images and plot the network graph in all the 6 cases as shown in Figure 4. We keep track of sources of all the cases while designing the experiment and observe that the predicted suspected source matches with the actual source Tweet ID. The source identification algorithm successfully filters out tweets with deepfake media similar to the queried image and finds the source amongst them. For the **unconstrained setting**, we filter out the deepfake tweets and plot their network as shown in Figure 6(a). Among the extracted tweets, the source for the deepfake tweets is correctly identified. Due to the limitations of the API, it is infeasible to collect all possible tweets belonging to a particular query or determine the ground truth. On manually checking the tweets identified by the algorithm, we observe that the filtered images or videos can potentially be fake. For example, the first tweet in the first row of Figure 6(b) is clearly a manipulated face. While not as clear, the other samples may also be fake. For the objective of identifying suspect sources, it is imperative we consider all potential tweets that might be fake. This is especially useful when an image query is provided to the algorithm.

**Results of Retweet Proneness Estimation:** The retweet proneness algorithm is evaluated on the test set consisting of 6000 tweets as described in the previous section. We obtain a mean-squared error of 1.08e-3 which indicates that the model is able to estimate the retweet count extremely well based on the profile of the user. In Figure 7, we represent the effectiveness of the algorithm. The difference between the predicted and actual retweet count per minute is calculated. If this difference is within a certain error threshold, the prediction is considered a hit. If the error is greater than the threshold, the prediction is a hit. We use six error thresholds ranging from 1e-7 to 1e-1. From the stacked bar plot, we can see that even for low error thresholds, the results are extremely promising with 90% accuracy at a threshold of 1e-4. The retweet count per minute is calculated for each node in the network as shown in Figure 6(a) giving a proxy estimate for its virality. We also observe a correlation between high retweet proneness and the temporal edges between source tweet and retweet used in the previous section.

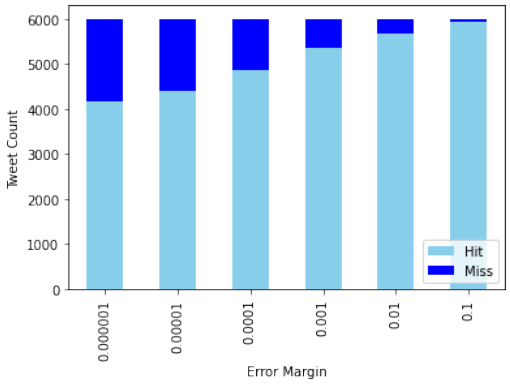


Figure 7. Tweet Count of "Hit" and "Miss" of predicted retweets/minute for different error margins (Best viewed in color)

## 5. Conclusion

This research presents a deepfake source identification algorithm, specifically for the Twitter platform. The sources are depicted through a network of deepfake tweets and retweets. We also estimate the retweet proneness and virality of deepfake media tweets. We encapsulate the entire framework into an easy-to-use web-based portal for better accessibility. In the paper, we use "#russia #ukraine" as the text query to showcase the scalability of the algorithm in practical applications. However, the proposed DeSI algorithm has numerous applications in real life scenarios. Deepfake content of public figures and celebrities can be used as a means for defamation or false propaganda. The DeSI algorithm can be used to track users indulging in the spread of deepfake content on social media platforms.

## 6. Acknowledgements

This research is supported through a grant from Ministry of Home Affairs, Government of India. S. Mittal is partially supported by the UGC-Net JRF Fellowship. K. Thakral is partly supported by the PMRF Fellowship. M. Vatsa is partially supported through the Swarnajayanti Fellowship.

## References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec 2018. 2

[2] Akshay Agarwal, Richa Singh, Mayank Vatsa, and Afzel Noore. Magnet: Detecting digital presentation attacks on face recognition. *Frontiers in Artificial Intelligence*, 4, 2021. 2

[3] Leo Breiman. Machine learning, volume 45, number 1 - springerlink. *Machine Learning*, 45:5–32, 10 2001. 5

[4] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017. 6

[5] DeepFakes. Deepfakes FaceSwap. https://tinyurl.com/y7gnkurs, 2017. 6

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. 2

[7] Ming Dong, Bolong Zheng, Nguyen Quoc Viet Hung, Han Su, and Guohui Li. Multiple rumor source detection with graph convolutional networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 569–578, New York, NY, USA, 2019. Association for Computing Machinery. 3

[8] Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. Tweepfake: about detecting deepfake tweets. *CoRR*, abs/2008.00036, 2020. 2

[9] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*, pages 87–102. Springer, 2016. 6

[10] Anubhav Jain, Puspita Majumdar, Richa Singh, and Mayank Vatsa. Detecting gans and retouching based digital alterations via dad-hcnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 2

[11] Debanjana Kar, Mohit Bhardwaj, Suranjana Samanta, and Amar Prakash Azad. No rumours please! a multi-indic-lingual approach for covid fake-tweet detection. In *2021 Grace Hopper Celebration India (GHCI)*, pages 1–5, 2021. 2

[12] Pakindessama M. Konkobo, Rui Zhang, Siyuan Huang, Toussida T. Minoungou, Jose A. Ouedraogo, and Lin Li. A deep learning model for early detection of fake news on social media*. In *2020 7th International Conference on Behavioural and Social Computing (BESC)*, pages 1–6, 2020. 2

[13] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018. 2

[14] Damian Kowalczyk and Jan Larsen. Scalable privacy-compliant virality prediction on twitter, 12 2018. 3

[15] Dariusz Krol and Karolina Wiśniewska. On rumor source detection and its experimental verification on twitter. pages 110–119, 02 2017. 3

[16] Prabhat Kumar, Mayank Vatsa, and Richa Singh. Detecting face2face facial reenactment in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020. 2

[17] Sanjay Kumar, Muskan Saini, Muskan Goel, and Bishwajit Panda. Modeling information diffusion in online social networks using a modified forest-fire model. *Journal of Intelligent Information Systems*, 56, 04 2021. 3

[18] David M. J. Lazer, Matthew A. Baum, Yochai Benkler, Adam J. Berinsky, Kelly M. Greenhill, Filippo Menczer, Miriam J. Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, Michael Schudson, Steven A. Sloman, Cass R. Sunstein, Emily A. Thorson, Duncan J. Watts, and Jonathan L. Zittrain. The science of fake news. *Science*, 359(6380):1094–1096, 2018. 2

[19] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing AI generated fake face videos by detecting eye blinking. *CoRR*, abs/1806.02877, 2018. 2

[20] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. *CoRR*, abs/1811.00656, 2018. 2

[21] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. 2

[22] MarekKowalski. FaceSwap. https://github.com/MarekKowalski/FaceSwap/, 2016. 6

[23] Paolo Nesi, Gianni Pantaleo, Irene Paoli, and Imad Zaza. Assessing the retweet proneness of tweets: Predictive models for retweeting. *Multimedia Tools Appl.*, 77(20):26371–26396, oct 2018. 3

[24] Yuval Nirkin, Lior Wolf, Yosi Keller, and Tal Hassner. Deepfake detection based on the discrepancy between the face and its context. *CoRR*, abs/2008.12262, 2020. 2

[25] Fabio Pezzoni, Jisun An, Andrea Passarella, Jon Crowcroft, and Marco Conti. Why do i retweet it? an information propagation model for microblogs. In *Proceedings of the 5th International Conference on Social Informatics - Volume 8238*, SocInfo 2013, page 360–369, Berlin, Heidelberg, 2013. Springer-Verlag. 3

[26] Gunther R, Bech PA, and Nisbet EC. Fake news may have contributed to trump's 2016 victory. 2018. 2

[27] Joshua Roesslein. Tweepy: Twitter for python! *URL: https://github.com/tweepy/tweepy*, 2020. 6

[28] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF ICCV*, October 2019. 5, 6

[29] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 6

[30] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016. 2, 6

[31] T.J. Thomson, Daniel Angus, and Paula Dootson. 3.2 billion images and 720,000 hours of video are shared online daily. can you sort real from fake? *The Conversation*, 2020. 1

[32] Ke Wang, Mohit Bansal, and Jan-Michael Frahm. Retweet wars: Tweet popularity prediction via dynamic multimodal regression. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1842–1851, 2018. 3

[33] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light cnn for deep face representation with noisy labels. *IEEE TIFS*, 13(11):2884–2896, 2018. 6

[34] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. *CoRR*, abs/1811.00661, 2018. 2

[35] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 6, 7

[36] Yousheng Zhou, Chujun Wu, Qingyi Zhu, Yong Xiang, and Seng W. Loke. Rumor source detection in networks based on the seir model. *IEEE Access*, 7:45240–45258, 2019. 3