THEME ARTICLE: CYBER SOCIAL HEALTH

Approaches for Fake Content Detection: Strengths and Weaknesses to Adversarial Attacks

Matthew Carter and Michail Tsikerdekis, *Western Washington University, Bellingham, WA, 98225, USA*

Sherali Zeadally, *University of Kentucky, Lexington, KY, 40506, USA*

In the last few years, we have witnessed an explosive growth of fake content on the Internet which has significantly affected the veracity of information on many social platforms. Much of this disruption has been caused by the proliferation of advanced machine and deep learning methods. In turn, social platforms have been using the same technological methods in order to detect fake content. However, there is understanding of the strengths and weaknesses of these detection methods. In this article, we describe examples of machine and deep learning approaches that can be used to detect different types of fake content. We also discuss the characteristics and the potential for adversarial attacks on these methods that could reduce the accuracy of fake content detection. Finally, we identify and discuss some future research challenges in this area.

Fake content is being created with the sole intention of misinforming, deceiving, and manipulating a targeted audience. It is multimodal and takes the form of videos, photos, news, reviews, and even Facebook likes. A more concrete example would be the forgery of a video that contains an important personality. The forgery could be minor such as changing their clothing or more substantial such as changing their face or speech. Advances in technology are making it easier to generate such fake content that is realistic in appearance. As such, there is a potential to have a major impact on business and society through influencing people's beliefs and decisions. A single fake news article could easily sway people's opinions on various topics.

One such case of fake content shaping a community's beliefs would be the mass distribution and consumption of fake tweets during Hurricane Sandy. One example of the fake content distributed during that time were pictures of sharks swimming in the residential areas hit by Hurricane Sandy. These drastically modified images caused even more panic throughout various communities and those affected by Hurricane Sandy.[14] It is easy to generate fake content on the Internet especially when using simpler forms of content such as news and reviews. During Hurricane Sandy, there was a deluge of fake photos on Twitter. An analysis of this event discovered that 10 215 users posted 10 350 unique tweets that contained some form of fake content. Thirty (30%) of users were responsible for the distribution of 90% of the content.[14]

Current fake content detection methods are capable of detecting trivial cases such as blatant fake posts but lack the capability to detect more sophisticated attacks. In recent years, detection techniques have shifted away from rule-based techniques to statistical approaches that use machine learning in order to become more effective in detecting fake content. However, with the proliferation of advanced computing technology and software, adversaries are becoming better at not only bypassing these detection methods, but also altering their intended function through the use of adversarial attacks.

In this article, we present machine learning fake content detection methods for different forms of content and we evaluate their strengths and weaknesses. In particular, we focus on adversarial attacks on these methods, and the challenges and opportunities for fake content detection. The remainder of this article is

Digital Object Identifier 10.1109/MIC.2020.3032323
Date of publication 20 October 2020; date of current version 16 April 2021.

organized follows. The “Fake Content Detection” section describes the common machine and deep learning methods that are used in detecting fake content for text, images, audio, and video. In the “Detection Method Characteristics” section, identify some key characteristics of these methods that influence their accuracy and computational overhead. Based on these characteristics, in the “Adversarial Attacks” section, we further identify how these characteristics minimize the risk of different types of adversarial attacks. Finally, we provide some recommendations in regards to the challenges and opportunities in fake content detection in the “Challenges and Opportunities” section.

FAKE CONTENT DETECTION

Fake content detection is the process of analyzing and classifying content as either real or fake. There are several techniques for identifying fake content and in recent years, the use of machine learning and deep learning approaches has become more common. These statistical detection techniques offer both better autonomy as well as higher detection rates. We describe the different techniques that are used for different types of content that are observed on social media which include text, image, audio, and video. These are not meant to be exhaustive but representative of the current trends in this research domain.

Supervised Machine Learning for Fake Content Detection

Machine learning is one of the most popular methods for detecting fake content due to its efficiency and minimal need of human assistance. Machine learning utilizes data analysis by “learning” from a set (or sets) of data provided, and then makes a decision through the identification of patterns. In the case of fake content, machine learning can be applied by training a classifier on a sample set of both fake and real content in order to identify the similarities and differences between the two and then have the classifier make a decision on new content to determine whether or not it is fake. Machine learning classifiers analyze features within each piece of content. These features include but are not limited to text-based features (spelling, punctuation, random characters), web-based features (domain name and links), and social-based features (number of likes, shares, comments, and friends/followers).

Fake Text:

Fake text refers to content that has been forged through natural language such as news articles, social media posts, and online reviews. This can be generated through humans or bots. Using machine learning, fake text can be detected through the analysis of the textual features. These features can be part of a social media post, or additional web metadata such as titles or keywords. Beyond these atomic features, additional information can be extracted based on the relations of terms within a text using natural language processing approaches such as tokenization and Term Frequency-Inverse Document Frequency (TF-IDF). In turn, the results from these processes are used to construct features that establish a model that defines a baseline between legitimate and deceptive texts.

One study that focused on news articles constructed a machine learning model that verified the relevancy of a headline with the body of an article based on word similarity patterns.[19] The aim was to establish a probability that the two are related and detect articles that intentionally deceive with their headlines in order to achieve better propagation through online social networks. A similar work[2] aimed to classify fake Yelp reviews that are generated with the purposes of promoting or suppressing a listing on Yelp. Typically, recommendation algorithms promote online shops with better reviews, and as such this method can have ramifications on a business’s viability. The study built supervised learning models that leveraged behavioral user analysis techniques used by the classifier in order to improve detection rates. The final model uses a combination of textual features found in each review as well as behavioral features that were extracted from a user’s account. The behavioral features included the frequency at which an account posts reviews, account age, and the number of positive reviews associated with an account. Using these classification features, the authors obtained an accuracy of 86.5% in detecting fake reviews.

Fake Images and Videos:

Fake content associated with graphics contain a variety of forged content such as fake images and fake videos. Machine learning can be used to detect images and videos that have been tampered with or faked altogether. Detecting this type of fake content requires a much deeper analysis of individual pixels, their intensities, and how they relate to past and future frames if the content is video. As such, training these models becomes computationally expensive. To mitigate the computational overhead, some methods apply data reduction solutions. One approach[8] for detecting fake or spoofed videos starts by removing all unnecessary data from each frame leaving only the key features and artifacts to be analyzed. Afterwards, the authors create a visual rhythm of the video which summarizes the entire video into a single frame. After these two pieces of information are created, machine learning is used to classify the patterns found in order to determine if the video is valid or spoofed. A similar study has demonstrated how it is possible to detect fake videos by looking for improper head position using support vector machines.[23]

Fake Audio:

Fake audio is an emerging type of fake content that focuses in spoofing specific audio features to produce a desired audio pattern. One main form of fake audio is synthetic audio which is the generation of audio patterns through text-to-speech and voice conversion software. Text-to-speech creates the base audio by generating audio patterns from text. In turn, voice conversion takes those created patterns and attempts to alter them to match a specified target. This process creates a nearly perfect audio sample for a target that did not produce that specific audio.[20]

One approach[21] for detecting fake audio focuses on detecting audio samples where the intonation has been intentionally faked. The intonations used in this study were normal, whisper, thick voice, thin voice, and nasal. This approach uses a neural network that does a subjective comparison of audio pairs in order to determine if the audio sample contained a faked intonation. This model took in pixels of wavelet coherence and outputted a binary value which represented if the current sample contained the same speaker or a different speaker. At the end of their experiments, they obtained an overall accuracy of 86.8% when classifying whether a speaker was the same or different.

Unsupervised Machine Learning for Fake Content Detection

Unsupervised machine learning allows for clustering of data without requiring an extensively annotated data set. This type of machine learning can be applied to fake content detection in order to cluster data such as news articles and tweets into groups that represent fake and real content. This can be beneficial because the approach has the potential of detecting previously unseen fake content. However, these techniques have not been extensively explored beyond the analysis of fake text.

Fake Text:

Unsupervised machine learning has been used to detect fake news on social media. The method[22] considers user credibility and truths of the news as latent random variables. A user opinion about a piece of news was proxied by measuring the amount of user engagement (such as tweet, liking, forwarding, or replying to a news tweet) with the news. Then, an analysis of these user opinions regarding the authenticity of the news results was used as an estimate of the authenticity of the news. The utilization of a probabilistic graphical model provides an unsupervised approach for calculating each latent random variable's probability. This algorithm uses Gibbs sampling after randomly initializing the latent random variables. Using several iterations of Gibbs sampling, estimations of the random variables are calculated and updated using a Bayesian update function. The final estimation of the authenticity of the news is calculated based on the average sampling values. Another approach has demonstrated fake news detection on Twitter using unsupervised clustering methods.[9]

Deep Learning for Fake Content Detection

Deep learning models attempt to imitate the decisions that humans make through an artificial neural network with numerous hidden layers. These layers allow deep learning models to not require the use of identified features but can use abstract data instead. In practice this means that the number of features and sample size tend to be substantially larger than in machine learning models. However, the proliferation of tools that use deep learning has also radically changed the abilities of adversaries to create fake content. As such, deep learning has become a mechanism for both creating fake content and detecting fake content.

Fake Images:

Deep learning can be used to detect artifacts in fake images and more advanced types of image forgery. An example of the latter is deepfake images, which are often the result of having facial features from one face mapped to another or having an entire face swapped with another using deep learning. For example, a study has shown that it is possible to detect fake images without using any of the image metadata as indicators of their fakeness[18]. Similar methods for detecting these types of forgeries compare the face and facial features with the background of the image and then evaluate the difference in quality. By looking at the difference between the two, the model[1] can determine if the qualities are vastly different in which case the image is most likely to have been forged. Such differences exist in forged images as a result of inserting an object (e.g., a face) from one image into another image. This processing, involves an encoder that reduces the dimension (size) of the face in the image resulting in a reduced number of features. The reduction of features when used in generating images leads to a lower quality and hence the image's "blurriness" is detectable.

Fake Videos:
Fake videos can also be detected using deep learning approaches. For example, we can use deep learning against other deep learning models used by attackers that can record facial expressions one of a person's face and then map them onto another person's face. Similar to deepfake images, deepfake videos often map facial features or entire faces from one person to another with the key difference being an added layer of complexity dealing with frames as opposed to a single image. Deepfake videos require more attention to detail as it is harder to make a convincing fake video than an image. Videos require that the facial features that are being mapped are continuously changing in order to match the position and orientation of the original face. A deep learning model described by Guera and Delp[13] detects these types of fake videos by looking at the features of each individual frame in the video. It then calculates a probability of how it fits within the profile of known fakes, and repeats this process for every frame in the video. The combined probabilities for each frame becomes the model which then makes the final decision on whether the video is fake. A shortcoming of many detection methods in this area is that they focus on detecting fake videos through inconsistencies that are produced by the fake video generation tools.[16] As such, these detection methods are likely to be inadequate as fake video generation tools improve.

DETECTION METHOD CHARACTERISTICS

Fake content detection methods vary in terms of the modeling algorithm used (supervised versus unsupervised) as well as the approach of data collection and the features used. As such, this variance creates variable detection rates. In fact, the detection accuracy produced through an experiment may yield unexpected results in real-world scenarios. As such, understanding the characteristics of method leads to a better understanding of the method's performance in real-world applications.

Here, we present the following detection method characteristics that we have identified from related works.

- Algorithmic complexity.
- Feature complexity.
- Data sanitization.
- Training sampling.

Algorithmic Complexity

Algorithmic complexity refers to the complexity of the detection system or classifier's detection technique. In other words, how difficult it will be for an adversary to reverse engineer the system. One such factor that contributes to this difficulty in reverse engineering is the incorporation of mechanisms that relate to the probability of having nondeterministic components.

Several effective types of attack against machine learning models focus on the exploitation of the model's complexity or decision boundary (the threshold beyond which the content is classified as fake). In these attacks, an adversary needs to gain knowledge about the model's training data or classification features in order to be able to determine the decision boundary. Once the decision boundary has been determined, the adversary can then send content to be incorrectly classified by the model.[4]

Feature Complexity

Some methods utilize easily obtainable features (e.g., keywords in a text) but others construct complex features from underlying simple variables. The result of the amount of postprocessing performed on gathered data can transform these variables in ways that are seemingly detached from the original dataset. For example, one can calculate the frequency of words and then apply the Gini coefficient (a statistical measure of distribution) to obtain a single number between 0 and 1 that determines how evenly distributed these word frequencies are.

A recent example[6] on the topic of adversarial stylometry combines various complex static and dynamic feature sets in order to determine the success rates of author stylometry recognition compared to adversarial stylometry. Stylometry is the statistical analysis of variations in literary style among various writers. The technique combines the use of a static feature set, which is independent of the documents being classified, and a dynamic feature set that is dependent on the documents being classified. Some of the observed results from using translation technologies (e.g., Bing Microsoft Translator, Google Translate) on a text passage reduce the precision of author recognition systems anywhere from 10% to 60% depending on the number of times the passage was translated.

Data Sanitization

Data sanitization is the process of removing outliers and unnecessary data from a dataset. This limits the amount of data that needs to be processed and removes unwanted data. A properly sanitized training dataset can lead to models that yield higher detection accuracy. However, improper sanitization can introduce biases in detection models. For example, an overly "clean" dataset can have experimental detection accuracy that is high but underperform in real-world problems. For example, Cretu *et al.*[7] describe an approach that utilizes data sanitization as a defense technique against adversarial attacks. The general approach aims to classify and remove outlying data points. The outliers are those classified as being too different from neighboring data points. This is done by using a score function that considers some data points and returns a real-value that represents how anomalous the point is with respect to predefined neighbors. If the value exceeds the defined threshold, the point is then removed from the set. Once this process is completed, the resulting dataset is then the sanitized set and can be used for training.

Training Sampling

Training sampling is the specific selection of a subset of data from the training dataset. This subset of data is then used to retrain the classifier instead of using the entire dataset. Due to the use of a smaller subset, training sampling allows faster training but with some tradeoff on the detection accuracy. For example, training sampling can mitigate imbalance in training sets.[3] This can occur due to some extreme outlier cases in the data that may otherwise skew the detection accuracy of the machine learning model. There are two main approaches for training sampling: under sampling and over sampling. Under sampling refers to the technique of removing examples from the majority class in order to balance the class distribution. Over sampling refers to the technique of adding copies of examples to the minor class in order to reduce skewness. Results from experiments conducted by Barandela *et al.*[3] found that under sampling techniques reduce the imbalance in the set and increase the overall performance of the classifier.

ADVERSARIAL ATTACKS

Adversarial attacks refer to a variety of data extraction and data manipulation techniques that are used against supervised and unsupervised models. In the context of fake content, adversarial attacks usually aim to exploit or circumvent classifiers in order to avoid detection. The effectiveness of such attacks lies in the assumptions that researchers make about their models that involve the model itself (e.g., algorithm used and parameters) as well as the data that is often derived from a closed set that is curated for laboratory experiments. Most adversarial attacks can be categorized into one of the following types.

- *Causative*: Attacks that mainly manipulate the training data in order to influence the training process.
  - *Training data poisoning*: Attacks that focus on polluting the training data in order to skew a model's classification of good and bad data.
  - *Testing data poisoning*: Attacks that abuse feedback systems attempting to manipulate a model's classification.

- *Exploratory*: Attacks that probe the system in order to learn the features the classifier uses.
  - *Black-box probing*: Attacks that attempt to reverse engineer the classifier's training data, features, and algorithm used for classification.
  - *Adversarial inputs*: Data that has been specifically crafted in order to be misclassified and avoid detection.

The aforementioned attack vectors (also shown in Figure 1) for machine learning model will yield different results based on the detection methods' characteristics associated with them. Put simply, some detection methods are more resilient against adversarial attacks. As such determining the resilience of a method can lead to more realistic expectations on its fake content detection accuracy. Next, we present an analysis that compares these attacks against the characteristics of the detection methods.

Table 1 summarizes the impact these attacks have on various detection methods' characteristics. In the following sections, we elaborate on adversarial attacks and their effect on the accuracy of detection methods.

Training Data Poisoning

Training data poisoning is the process of manipulating a classifier's training data in order to influence the

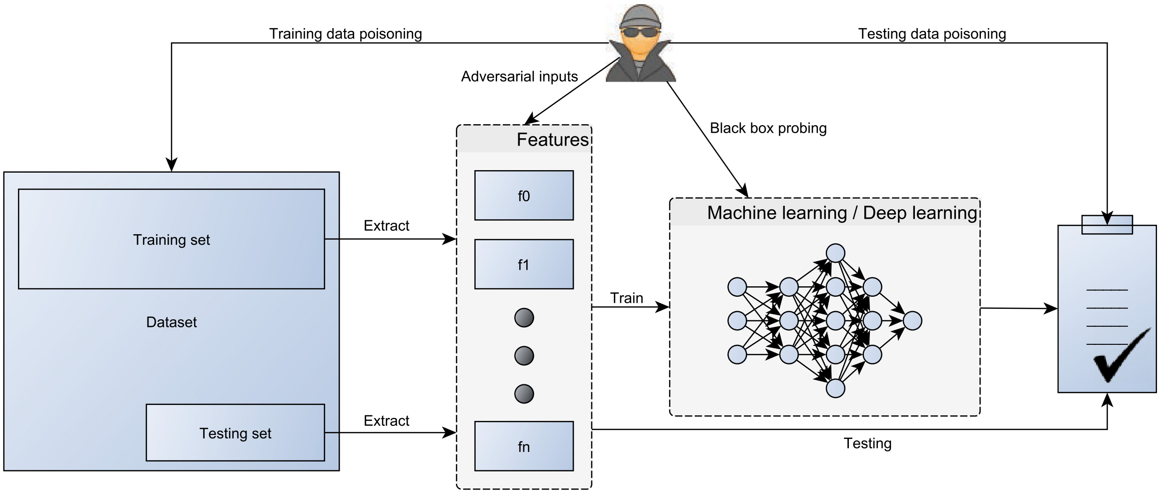

FIGURE 1. Overview of a typical machine learning workflow and how adversarial attacks impact different aspects of that system.

training process.[17] This is done through an adversary gaining control over some portion of the training set and then influencing the classifier for future exploitation. The training dataset is a crucial component in training a fake content detection model that utilizes some form of machine or deep learning as the base on which the model is being built.

The data poisoning process involves an attacker injecting carefully crafted data into a training set in order for the classifier to be retrained using this poisoned data. Once the classifier is retrained with the poisoned data, similarly crafted packets sent by the adversary could avoid detection.

For example, if the content contains a set of specific keywords (e.g., "real") is tagged as fake, an attacker can populate the system with legitimate content that uses these keywords. After some period of time, when the classifier is retrained by an online platform using more recent data, these keywords will not be flagged and associate the content as fake anymore.

Causative type attacks are effective against defensive techniques that focus on classification because they target the classifier's training data. If the training data can be manipulated, well-crafted data can then evade classification of these defensive techniques.

Algorithmic Complexity:

The degree of algorithmic complexity for a detection method does not influence the effectiveness of training data poisoning attacks. Put simply, no matter how complex a classifying algorithm is, if it is trained on poisoned data then the classifier will still learn incorrectly.

Feature Complexity:

Feature complexity can potentially mitigate the effect of training data poisoning but not the action of tampering with the training data itself. In turn the poisoned data will become part of the classifier but in an unpredictable way. The mitigating effect exists due to the information asymmetry between the attacker and the defender because feature complexity obfuscates the process through which fake content is detected. Put simply, if the original variables that make up a feature are substantially processed, then the poisoning of these original variables may not have a predictable effect on the classifier from the perspective of an attacker that lacks the knowledge about what the constructed features are.

TABLE 1. Risk reduction of adversarial attacks based on fake content detection methods' characteristics.

	Sub-category	Algorithmic complexity	Feature complexity	Data sanitization	Training sampling
Causative	Training data poisoning Testing data poisoning	Low Low	Medium Medium	High High	High Medium/High
Exploratory	Black-Box probing	High	High	Low	Low/Medium
Adversarial inputs	High	High	Medium	Low	

Data Sanitization:
Training data poisoning can have a lower impact on models where data sanitization is applied. Even if the training data is poisoned, data sanitization will likely remove the "tainted" data from the training data before retraining the classifier. Manual data sanitization is more effective than an automated solution. When the classifier is retrained, the risk of an adversary's influence on the classifier decreases.

Training Sampling:
Training data poisoning is not effective against machine learning models that use training sampling techniques. This is due to the fact that training sampling trains the classifier using only a portion of the training data and not the entire set. The process of training the classifier with a subset of the training data lowers the chances of the classifier being trained on enough poisoned data which can result in incorrect classifications.

Testing Data Poisoning

Similar to training data poisoning, testing data poisoning focuses on exploiting the system to gain some control over the testing dataset. An attacker has the potential to influence the testing data if such data has been collected from public sources (e.g., posts in a forum). The testing dataset is used to evaluate a model's validity and as such this attack aims to weaken this process.[10] As such, if an attacker is able to influence this set, the engineers of the machine learning model can overestimate a fake content detection model's accuracy. An example of such an attack would be for an attacker to generate easily detectable fake reviews at an electronic commerce website in order to influence the model's estimated accuracy. Defenders will assume that the detector's accuracy is high when in reality the cases that it detects are trivial.

Algorithmic Complexity:
The algorithm's complexity does not play a significant role in testing data poisoning attacks. That is, the model will still be evaluated on the poisoned testing dataset. For example, with supervised models, a decision tree and an ensemble model will be evaluated the same way based on the same poisoned testing data. The problem persists even for models that are not generated *a priori* such as unsupervised models that are still evaluated for their accuracy against a testing set.

Feature Complexity:
Complex features provide some mitigating effect on testing data poisoning similar to training data poisoning attacks. The effect is unknown for both attacker as well as defenders that build the detection model because feature construction processes the original data. In general, the more a feature has been processed the more unpredictable the effect of testing data poisoning becomes.

Data Sanitization:
Data sanitization is effective against testing data poisoning attacks. Manual implementation of such an approach on testing data is more feasible because the testing data tends to be smaller and intentionally including interesting outliers is advisable because it provides a better benchmark for the limits of the detection model.

Training Sampling:
Most of the literature described in section that used training sampling did not use any samples from the testing data. As such, poisoning attacks on testing data will be effective for such machine learning models. An exception exists for models that apply sampling on a dataset and then use the derived dataset in techniques such as k-fold cross-validation, where efficient training and testing of the data are derived from the same sampled set. In this case, since sampling occurred initially before any of the model testing, an attacker's poisoned data will have a lower impact because not all "tainted" data are guaranteed to be sampled.

Black-Box Probing

Black-box probing focuses on extracting the features or performance metrics used in classification by a detection system.[15] The process of crafting data to be incorrectly classified occurs after the attacker has probed the classifier (see "Adversarial Inputs" section). Once the attacker has found the classification features used when identifying data, the attacker can specifically craft his/her data to avoid detection on those features. Furthermore, an attacker who is given the ability to probe the system multiple times will be more likely to succeed in reverse engineering the fake content detection model based on the output and derive the features that are used in fake content detection.

Algorithmic Complexity:
Black-box probing attacks are not effective against complex machine learning algorithms because they

are more difficult to reverse engineer. Such systems are difficult to reverse engineer by an attacker who has limited interaction with the detection system. Typically, the use of ensemble methods and complex deep learning models are more effective in mitigating black box probing attempts.

Complex Features:
Black-box probing is not effective against systems that use complex features. This is a by-product of the amount of post processing that is performed on the extracted features. As a result, it is more difficult for an adversary to reverse engineer the features that have been originally extracted.

Data Sanitization:
Data sanitization is an approach that aims to protect training and potentially testing data for machine learning models. As such, it has no effect on an attacker probing a model.

Training Sampling:
Sampling over the training set will have no effect on the ability of an attacker to reverse engineer the machine learning model. However, frequent training of the machine learning model that uses sampling is likely to cause the adversary to probe the detection model frequently.

Adversarial Inputs

Adversarial inputs are an attack vector that aims to violate a machine learning model's policies or operational foundations. These types of attacks focus on evading the detection of a classifier by exploiting blind spots in what a classifier has learned. An attacker will carefully craft his/her data in such a way that it is incorrectly classified. The process of crafting data so that it is incorrectly classified usually occurs after the attacker has probed the classifier. As such, a protective measure that could determine the success of this attack relies on the number of attempts that an attacker has to probe a machine learning model. Once the attacker has found the classification features used to identify fake content, the attacker can specifically craft his/her data to avoid detection on those features. An example of this approach relates to image quality and processing with deep learning models to detect modification artifacts in fake images. High fidelity images provide the detection model with more information and as such the likelihood of detecting artifacts indicative of a fake image.[12] Thus, an attacker can identify the image quality bounds for which the deep learning model does not perform that well in detecting fake content. Another recent example described by Behzadan and Munir[5] has shown that such attacks are also viable for reinforcement learning models. Attackers can train an attack model at a fraction of a cost that it takes a defender to build a model and confuse the defender's model policies (behavior rules).

Algorithmic Complexity:
Algorithmic complexity can minimize the adversarial risk because black-box probing needs to be successful first. However, it is worth noting that complexity can also have an adverse effect and instead lead to security theater (i.e., providing a false sense of security). Seemingly complex models to humans can often be easily reverse engineered by machines.[11] As such, increasing algorithmic complexity decreases the chances of successful adversary attacks for models that have additional security measures such as limiting the ability of an attacker to probe a system.

Feature Complexity:
Complex features, in general, can reduce the likelihood for adversarial inputs to succeed. However, due to the added complexity there could be borderline cases that if discovered by an attacker, the model can be exploited. In other words, the added complexity obfuscates the degree of security for both the defender and the attacker of a system. As such, substantial testing on adversarial inputs is recommended.

Data Sanitization:
Data sanitization provides a minimal mitigating effect against adversarial inputs because the actions performed on a dataset will not reduce the ability of an attacker to reverse engineer a model's inputs. Furthermore, if an attack is able to extract the features used by the classifier, data sanitization will not help prevent the incorrect classification of data. There is however a benefit if data sanitization involves the intentional injection of examples of adversarial inputs into the model. In such a case, the outcomes could be a detection model that performs better against adversarial inputs or a model with a more realistic estimation of the real world accuracy in detecting fake content.

Training Sampling:
Similar to data sanitization, training sampling only minimally reduces the risk against adversarial input attacks. Training sampling is a technique that retrains the classifier with a subset of the full training dataset but will not prevent reverse engineering the classifier.

CHALLENGES AND OPPORTUNITIES

Based on the analysis that we have conducted in this article, we recommend that a combination of characteristics be used in order to lower the overall susceptibility of a model against adversarial attacks. For example, a model with high algorithmic complexity that utilizes data sanitization techniques will have a lower risk toward both causative and exploratory adversarial attacks. While algorithmic complexity and feature complexity are not inherent in all models, the inclusion of such characteristics reduces susceptibility to various types of exploratory attacks.

Beyond this, the above findings on the existing content detection methods that we have discussed lead to several challenges and opportunities that we have identified that can reduce the susceptibility of fake content detection models to adversarial attacks. These include the quasi-experimental evaluation of classifiers, the distribution and use of open datasets, and domain-specific fake content detection algorithms focusing on anomaly detection.

Quasi-experimental evaluation of classifiers: Maintaining best practices for testing and training sets is crucial because several types of adversarial attacks focus on poisoning both the testing and training data. For example, an improvement on fake content detection methods is to revise the updating process for the classifier so that both the testing and training data are re-evaluated against past and current trends. This evaluation of the evolution of these metrics in a quasi-experimental fashion should prove beneficial because it provides a higher degree of control and security over the contents of the testing and training datasets and will alert engineers in the event that such data has been altered substantially. The expectation is that user behavior can change due to various factors such as revisions of a platform's policies but these behavioral changes are not typically radical.

Open datasets: Furthermore, the creation and use of open datasets for evaluating fake content detection models would provide a better foundation for future work to be built on. In our survey, we have identified that many of the models and derived data are not released or cannot be released due to license and privacy restrictions. The lack of universal datasets that can serve as a benchmark for fake content detection leads to ambiguous accuracy for published models. Open datasets will further foster both academic and industry collaborations and lead to the development of better fake content detection models.

Domain-specific fake content detection algorithms: Finally, although we have identified studies that have used unsupervised modeling solutions, it is important to highlight that these are not mutually exclusive to supervised machine learning. Both unsupervised and supervised machine learning solutions are valid and they can be used together to secure an online platform much like in network security these two types of machine learning paradigms are used to detect known but also zero-day attacks. There is currently a large number of studies that focus on supervised machine learning solutions and as such the potential for adversarial attacks is higher. On the other hand, many unsupervised machine learning algorithms perform better in this context. However, they are not necessarily designed for nondeterministic data (human behavior often falls under this category). For example, deep belief networks is an unsupervised algorithm that is mainly designed to identify features based on an unlabeled input, and then uses the identified features as labels for a supervised learning model (e.g., neural network). Another unsupervised learning model such as DBSCAN is built for robustness toward outliers. However, in fake content detection, we often have a lot of legitimate content and the fake content appears as outliers, rendering these types of algorithms to be of limited use in this domain. As such we recommend further research into novel unsupervised machine learning algorithms that focus on anomaly detection especially for noisy data that can be multidimensional.

CONCLUSION

In this article, we reviewed fake content detection approaches that use machine learning techniques and we have highlighted the strengths and weaknesses of these techniques against adversarial attacks. We further demonstrated that the detection model's characteristics affect the susceptibility of the fake content detection model to adversarial attacks. Given that fake content creation techniques are becoming more advanced with various deep learning generation algorithms the challenges for ensuring an asymmetrical advantage for fake content detection models must be addressed. As such, not only further research is required in this domain but substantial progress in data and algorithm communication is needed among researchers involved in this field.

ACKNOWLEDGMENTS

We thank the guest editors and reviewers for their valuable comments.

REFERENCES

1. D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in 2018 *IEEE Int. Workshop Inf. Forensics Secur.*, pp. 1–7, 2018.
2. M. Arjun, V. Vivek, L. Bing, and G. Natalie, "Fake review detection: Classification and analysis of real and pseudo reviews," *Tech. Rep.*, vol. 80, no. 2, pp. 159–169, 2013.
3. R. Barandela, R. M. Valdovinos, J. S. Sánchez, and F. J. Ferri, "The imbalanced training sample problem: Under or over sampling?" *BT—Structural Syntactic Statist. Pattern Recognit.*, pp. 806–814, 2004.
4. M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. ACM Symp. Inform., Comput. Commun. Secur.*, vol. 2006, 2006, pp. 16–25.
5. V. Behzadan and A. Munir, *Vulnerability of Deep Reinforcement Learning to Policy Induction Attacks BT—Machine Learning and Data Mining in Pattern Recognition*. Cham, Switzerland: Springer, 2017, pp. 262–275.
6. M. Brennan, S. Afroz, and R. Greenstadt, "Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity," *ACM Trans. Inf. Syst. Secur.*, vol. 15, no. 3, 2012, Art. no. 12.
7. G. F. Cretu, A. Stavrou, M. E. Locasto, S. J. Stolfo, and A. D. Keromytis, "Casting out demons: Sanitizing training data for anomaly sensors," in *Proc. IEEE Symp. Secur. Privacy*, 2008, pp. 81–95.
8. A. Da Silva Pinto, H. Pedrini, W. Schwartz, and A. Rocha, "Video-based face spoofing detection through visual rhythm analysis," in *Proc. Brazilian Symp. Comput. Graph. Image Process.*, 2012, pp. 221–228.
9. K. Darwish, P. Stefanov, M. Aupetit, and P. Nakov, "Unsupervised user stance detection on Twitter," *Proc. Int. AAAI Conf. Web Social Media*, May 2020, vol. 14, pp. 141–152.
10. S. De Silva, J. Kim, and R. Raich, "Cost aware adversarial learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 3587–3591.
11. A. Gleave, M. Dennis, N. Kant, C. Wild, S. Levine, and S. Russell, "Adversarial policies: Attacking deep reinforcement learning," 2020, *arXiv:1905.10615*.
12. L. Guarnera, O. Giudice, and S. Battiato, "Deepfake detection by analyzing convolutional traces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2020, pp. 2841–2850.
13. D. Guera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal-Based Surveillance*, 2019, pp. 1–6.
14. A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, "Faking Sandy: Characterizing and identifying fake images on," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 729–736.
15. N. Narodytska and S. Kasiviswanathan, "Simple black-box adversarial attacks on deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 1310–1318.
16. T. T. Nguyen, C. M. Nguyen, D. T. Nguyen, D. T. Nguyen, and S. Nahavandi, "Deep learning for deepfakes creation and detection: A survey," 2019, *arXiv:1909.11573*.
17. J. Steinhardt, P. W. W. Koh, and P. S. Liang, "Certified defenses for data poisoning attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 3517–3529.
18. S. Tariq, S. Lee, H. Kim, Y. Shin, and S. S. Woo, "Detecting both machine and human created fake face images in the wild," in *Proc. 2nd Int. Workshop Multimedia Privacy Secur.*, 2018, pp. 81–87.
19. J. Thorne, M. Chen, G. Myrianthous, J. Pu, X. Wang, and A. Vlachos, "Fake news stance detection using stacked ensemble of classifiers," in *Proc. Workshop: Natural Lang. Process. Meets Journalism*, 2017, pp. 80–83.
20. M. Todisco *et al.*, "ASVSpoof 2019: Future horizons in spoofed and fake audio detection," *Proc. Annu. Conf. Int. Speech Commun. Assoc., INTERSPEECH*, 2019, pp. 1008–1012.
21. F. Tom, M. Jain, and P. Dey, "End-to-end audio replay attack detection using deep convolutional networks with attention," *Proc. Annu. Conf. Int. Speech Commun. Assoc., INTERSPEECH*, 2018, pp. 681–685.
22. S. Yang, K. Shu, S. Wang, R. Gu, F. Wu, and H. Liu, "Unsupervised fake news detection on social media: A generative approach," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 5644–5651, 2019.
23. X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2019, pp. 8261–8265.

MATTHEW CARTER is currently working toward the Graduate degree in the computer science program with Western Washington University, Bellingham, WA, USA. Contact him at carter36@wwu.edu.

MICHAIL TSIKERDEKIS is currently an Assistant Professor with the Computer Science Department, Western Washington University, Bellingham, WA, USA. His research interests include deception, data mining, cybersecurity, and social computing. He received the Ph.D. degree in informatics from Masaryk University, Brno, Czechia. He is a senior member of the Institute of Electrical and Electronics Engineers. He is the corresponding author of this article. Contact him at michael.tsikerdekis@wwu.edu.

SHERALI ZEADALLY is currently an Associate Professor with the College of Communication and Information, University of Kentucky, Lexington, KY, USA. His research interests include cybersecurity, privacy, Internet of Things, computer networks, and energy-efficient networking. He received the bachelor's degree in computer science from the University of Cambridge, Cambridge, U.K., and the doctoral degree in computer science from the University of Buckingham, Buckingham, U.K. He is a Fellow of the British Computer Society and the Institution of Engineering Technology, U.K. He is a Senior Member of the IEEE. Contact him at szeadally@uky.edu.