2022 IEEE 4th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA) | 978-1-6654-7408-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/TPS-ISA56441.2022.00016

Developing Self-evolving Deepfake Detectors Against AI Attacks

Ian Miller
Department of Computer Science
Vanderbilt University
Nashville, TN, USA
ian.miller@vanderbilt.edu

Dan Lin
Department of Computer Science
Vanderbilt University
Nashville, TN, USA
dan.lin@vanderbilt.edu

***Abstract*—As deep-learning based image and video manipulation technology advances, the future of truth and information looks bleak. In particular, Deepfakes, wherein a person's face can be transferred onto the face of someone else, pose a serious threat for potential spread of convincing misinformation that is drastic and ubiquitous enough to have catastrophic real-world consequences. To prevent this, an effective detection tool for manipulated media is needed. However, the detector cannot just be good, it has to evolve with the technology to keep pace with or even outpace the enemy. At the same time, it must defend against different attack types to which deep learning systems are vulnerable. To that end, in this paper, we review various methods of both attack and defense on AI systems, as well as modes of evolution for such a system. Then, we put forward a potential system that combines the latest technologies in multiple areas as well as several novel ideas to create a detection algorithm that is robust against many attacks and can learn over time with unprecedented effectiveness and efficiency.**

I. Introduction

Historically, recorded media has been regarded as a firm, reliable source of information. Whatever the content may be, one could generally trust that the things said in the audio, or the actions performed in the video, truly did happen. These sources are so trusted they are used to keep records and even as evidence in criminal proceedings. In recent years, photo and audio manipulation has grown more advanced, and can even go completely unnoticed by humans who neglect to look very hard. However, such manipulations can still typically be detected by algorithms. The advent of artificial intelligence puts even this level of security under threat. AI-based media manipulation has the potential for unparalleled levels of stealth and subtlety. This danger grows ever stronger as Deep Neural Networks (DNNs), Generative Adversarial Networks (DNNs), and basic computing power grow in size, complexity, and ubiquity.

The possibility for anyone with sufficient hardware and time to create extremely convincing fake videos of nearly anybody could be considered a threat even to core principles of the modern world, like democracy and security. It is no secret that cleverly manufactured, inflammatory misinformation can spread like wildfire on the internet, affecting the opinions of millions in mere days, while the truth of the matter lags behind, spreading slower because it is less interesting, and affecting still fewer minds simply because the misinformation reached that mind first. This is undeniably already the case with flimsy pieces of evidence such as out-of-context or completely fabricated quotes. If even video, a format widely thought to be hard evidence, is corrupted by malintent, it calls into question the safety of everyone.

Deepfake, perhaps the most well-known AI-based video manipulation tool, can be used to swap a victim's face into arbitrary scenes. This could be exploited to, among countless other things, create fake news regarding political figures, cause chaos in the financial market, incite public discontent and violence, or even inflame political and religious tensions between nations. Anybody could be made to say anything.

In such a future, one would hope that it is at least still possible to detect even the most advanced Deepfakes. However, this is insufficient to assuage the aforementioned concerns if the capability only lies with those who have extensive computing resources. It is necessary in such a world that the public themselves have easy access to software that can detect all sorts manipulated media. Even still, being an AI-based technology, Deepfake algorithms can improve in quality automatically to fool existing detectors. As such, it is also necessary that the detection technology is able to not only keep up with this rapid evolution with its own AI technology while maintaining its defenses against older algorithms, but also outpace the attackers, getting ahead of existing Deepfake algorithms with predictive learning.

The detectors must also be robust against common attacks on AI systems. Attacks on machine learning systems can be broadly classified into three categories: (1) adversarial input attacks; (2) data poisoning and backdoor attacks; and (3) model stealing attacks. In a model stealing attack, an attacker seeks to extract some or all of a trained model to use for themselves. This sort of threat is hardly germane to the topic at hand, though, since a Deepfake detection algorithm being in the hands of as many people as possible is a public good. The challenges in the area of defense against attackers for Deepfake detection algorithms largely relate to the first two types.

All of the aforementioned open problems need to be pondered to preserve the integrity of images and videos shared online.

The rest of the paper is organized as follows. Section

978-1-6654-7408-5/22/$31.00 ©2022 IEEE
DOI 10.1109/TPS-ISA56441.2022.00016

(a) Original Image (b) Noise Mask (Exaggerated for visibility) (c) Final Product

Fig. 1: A demonstration of how a noise mask can be subtly applied to an image with little visible change, but massively alter it in the eyes of a computer. Photo by Artistic Frames on Unsplash.

II discusses the existing adversarial input attacks and how Deepfake detectors may defend against such attacks. Section III discusses the data poisoning attacks and some possible defense mechanisms that Deepfake detectors could adopt. Section IV presents the challenges faced by Deepfake detectors to achieve self-learning and self-evolving down the road. Section V introduces an envisioned comprehensive Deepfake detection system that integrates the desired capabilities in the previous sections. Finally, Section VI concludes the paper.

II. Deepfake Detectors Face Adversarial Input Attacks

Adversarial Inputs would logically be the most common method of attack on a DeepFake detection system, since an attacker's goal is to camouflage a fake sample as legitimate and this is the most direct way to do so. Because such a detector would rely on a large dataset of existing DeepFakes, it would be vulnerable to perturbed fake images that mask the signature features of a certain type of DeepFake [1]. As noted by Szegedy et al. [2] in their initial study of such attacks, a subtle, cleverly designed noise mask injected into the image can hide these features from the detector while remaining nearly unnoticeable to the human eye. This presents the most immediate threat to the system, so defenses against this type of attack are especially needed.

A. Background about Adversarial Input Attacks

Adversarial inputs are produced through the process of adversarial training. The earliest such method was realized by Goodfellow et al. [3] and dubbed the Fast Gradient Sign Method (FGSM). This method exploits the then newly discovered linearity of deep learning models in higher dimensional space, which makes them vulnerable to simple perturbations. Later, Kurakin et al. [4] analyzed FGSM on the ImageNet dataset and found that it was successful at changing the classifier's top pick 63-69% of the time. In the few years since then, numerous other techniques of creating adversarial samples have emerged.

While the basic one-step method creates perturbations by taking a single large step down in the direction with the steepest gradient, a reasonable adjustment would be to take several smaller steps, adjusting the direction between each one, to ensure the perturbation continues to follow the steepest gradient descent direction. This Basic Iterative Method (BIM) is notably more effective but also more costly in terms of computation [5].

In some cases, perturbation of even a single pixel, which could be dismissed as a small image error by the common consumer, can be sufficient to cause misclassification [6]. Further, the method to create these adversarial samples utilizes differential evolution, meaning it does not rely on the target model's parameters to function. This method, fittingly dubbed the One Pixel Attack achieves up to 70% success at inducing misclassification.

Rather than attempting to modify a minimal number of pixels (l_0), one can also try to minimize the total difference (l_1), total square difference (l_2), or maximum single pixel difference (l_∞) between the original and perturbed image. By restricting the (l_0), (l_2), and (l_∞) norms, Carlini and Wagner [7] created another method of generating perturbations that aims to make the changes as imperceptible as possible while still accomplishing the desired effect on classification.

Yet another algorithm, known as DeepFool [8], expands on the work of Carlini and Wagner and is able to perturbations that are notably smaller than those produced by FGSM, while maintaining a similarly high fooling rate. It is even possible to produce perturbations that are effective regardless of the image they are applied to, as is the case with the Universal

Perturbations for Steering to Exact Targets (UPSET) algorithm [9]. These perturbations are a universal and generic way to steer any image toward a target class, so one need not even waste the time considering the content of the input image when producing adversarial samples. UPSET is able to produce a perturbation for each possible class, and when applied to an image not belonging to that class, the perturbation will reliably cause it to be misclassified as that target class regardless of image content or model.

B. Defending Against Adversarial Input Attacks

These types of attacks present the most immediate threat to the detection system, so defending against this sort of attack is critical. Since methods like FGSM rely on knowledge of the model's gradient to construct their perturbations, it could be effective to obscure this knowledge. In particular, one could make a model that is non-differentiable, such as a decision tree or nearest neighbor classifier. Unfortunately, this defense is rather simple to thwart, as the attacker can train a surrogate model that does have a gradient based the decisions of the target model, then craft their perturbations toward the surrogate to form a reasonably effective attack on the target.

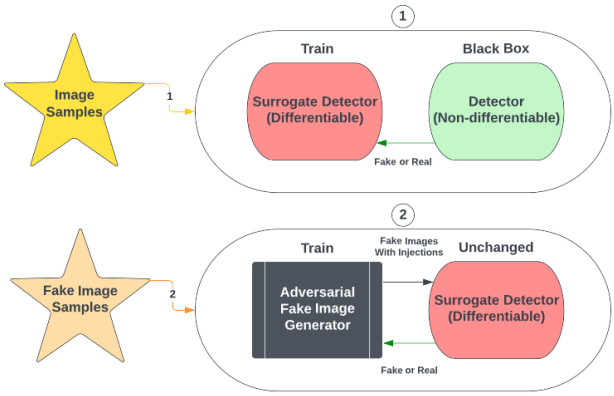

Fig. 2: The Training Process for an adversarial perturbation generator against a target model with hidden or nonexistent gradient

One of the most rudimentary defenses is to apply a generic transformation to all input samples that aims to mitigate potential adversarial perturbations before running them through the detector. This transformation could be compression, randomized resizing, or padding, to name a few [13]. In fact, many social media and video hosting sites already employ these sorts of transformations anyway, just to save space. All of these have been shown to be effective at reversing the accuracy drop from the perturbations to some extent, but are insufficient to provide robust protection. For example, Dziugate et al. [14] found that JPG compression, while effective at reducing the adversarial nature of small perturbations, quickly became less effective as the perturbations grew larger. It was also found that JPG compression has the very low impact on clean accuracy compared to other input transformations.

(a) Original Image (b) JPEG Compressed

Fig. 3: A comparison between an original and a JPEG compressed image. Because the two are nearly identical, it follows that JPEG compression has little impact on clean accuracy, and this is indeed what the data supports [13].

Total Variance Minimization (TVM) is another type of transform that can help mitigate perturbations. A compressed sensing approach combining pixel dropout with TVM works by randomly selecting a small set of pixels, then reconstructing a simpler image that is consistent with those pixels. Because perturbations tend to be small and localized, the reconstructed image is usually free of them. The resultant image also tends to be blurry/blocky, similar to compression. Other than that, this method is also reasonably effective for small and/or localized perturbations. TVM has the most impact on clean accuracy of the input transformations discussed here.

Image Quilting is one of the more complicated transformations [16]. It utilizes a large database of image patches as a reference. When an image is given to the image quilting algorithm, it is first split into a predefined grid. Then, for each grid point, the algorithm finds a patch in the database that closely matches that part of the image, usually by finding the K nearest neighbors and selecting one at random. Because the patch database is under the control of the defending party, it is free of any adversarial perturbations. This method has proven the most effective of the three in a black-box setting with relatively large perturbations. Image Quilting has more impact on clean accuracy than JPG compression, but less than TVM [13].

While input transformations are a reasonably effective first line of defense, as they force the adversarial perturbations into a more visible range, they are not solutions on their own for two reasons. For one, they are not capable of actually detecting the perturbations, just blindly attempting to remove them. Second, there is a massive trade-off between their effectiveness as a defense and the model's clean accuracy, since these methods necessarily alter every image, even unperturbed ones. It seems that more complex and adaptive strategies are needed.

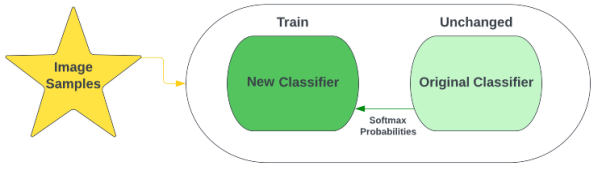

Fig. 4: A diagram depicting the process of defensive distillation wherein a classifier is trained from the softmax probability outputs of another classifier.

Rather than modifying the input data in an attempt to sterilize it, the next logical step would be to fortify an existing model. From this arises the idea of defensive distillation [17]. Suppose an existing model classifies a dataset X into target classes Y. The model's final layer is a softmax which produces a probability spread over Y based on the input's predicted labels. Now, suppose one trains a new model on dataset X, but rather than using the target class labels of the dataset - in this case, legitimate or illegitimate - one instead uses the probability spread output by the original model to train the new one. These new labels encode more information about X's membership to each target class compared to the simple one-hot labels with which the original model was trained. The motivation behind all of this is to smooth the model, converting hard binary class labels to soft targets. Afterward, one can replace the original model with the smoothed one, which has the added benefit of possibly throwing off adversarial perturbations that were trained for the original model [18, 19]. Unfortunately, modern adversarial perturbation creation strategies produce attacks that are highly transferable, meaning the same perturbation can fool several different models. The smoothed models produced by defensive distillation are simply too similar to the original to evade modern adversarial attacks [20]. This sort of universality is the next hurdle to overcome.

Even two separate models with different architectures and trained on entirely disjoint datasets can often be fooled by the exact same adversarial perturbation. This is the cause of failure for a large number of potential solutions, so the question becomes how to diminish transferability. Hosseini et al. [21] demonstrate the effectiveness of a so-called NULL Labeling Method. The idea here is to allow the classifier the freedom the label an input as NULL rather than classifying it normally. The model is then trained with both clean samples, labeled as they normally would be, as well as adversarial samples, labeled as NULL. However, clean data can also be perturbed, usually through digital compression. If the algorithm assigns all perturbed data to NULL, it will severely impact its clean accuracy. To alleviate this, the algorithm is designed in such a way that the NULL probability assigned to each input represents how strongly the classifier believes it to be adversarial. NULL labelling is among the most effective known strategies for defending against adversarial inputs.

Hussain et al. [27] have shown that state-of-the-art Deepfake detectors are vulnerable to adversarial attacks in the case where the attacker has full or even partial knowledge of the detector. However, little research has been conducted into the effectiveness and practicality of the various methods of defense against said attacks for this specific task. Chen et al. [28] proposed an interesting solution called MagDR (mask-guided detection and reconstruction) which involves iterating inputs through a process of detecting perturbations, removing them, and then reconstructing the image. MagDR shows promising results against modern adversarial inputs. However, the authors also admit that their method could become irrelevant quickly, with the increasing ability of attackers to generate more 'natural' perturbations.

III. Deepfake Detectors Face Data Poisoning and Backdoor Attacks

In general, data poisoning attacks aim to contaminate the model's training dataset with samples designed to impact the trained model's accuracy. Within the subfield of data poisoning attacks, one can further classify them into targeted and non-targeted varieties.

The non-targeted attacks aim to decrease the overall accuracy of the model by causing general confusion during training. These are relatively simple to detect because the overall accuracy drop can be spotted and its cause pinpointed and discarded.

Targeted attacks, however, are far stealthier. They aim to affect the classification of just a specific type of sample, in this case, to camouflage a specific type of Deepfake into being perceived as genuine. To be specific, one could accomplish this by flooding the training data with genuine images that have been edited to include a certain human-imperceptible feature that the algorithm can easily identify and associate with the real images, then use the same feature in a Deepfake generation algorithm to sneak falsified images through the detector. Such attacks are far harder to detect because they ideally only lower detection accuracy toward the attacker's manufactured images and nothing else, making them virtually undetectable in advance of the attack. As it stands, no generally applicable defense against all data poisoning attacks has been developed.

Backdoor attacks work on a similar principle to data poisoning attacks in that they aim to modify the model's behavior, but the exact way they do so is slightly different. While data poisoning attacks seek to lower a model's overall accuracy or its accuracy to a specific type of input, backdoor attacks don't aim to affect the model's clean accuracy at all. In fact, it is in their best interest to do so as little as possible to remain undetectable. Instead, these sorts of attacks aim to inject a spurious association between a certain label and a symbol/perturbation (trigger) that can be placed into a sample. Ideally, the model's clean accuracy is unaffected, but any samples with the trigger will be misidentified as the chosen label. Fortunately, it has been shown that the computational time needed to defend against backdoor attacks largely depends on the number of labels the model can choose.

Attackers against a theoretical Deepfake detector would really only have incentive to try to pass their samples as legitimate, meaning that is the only class that needs such robust defenses, so defending against backdoor attacks will present little issue.

IV. Deepfake Detectors Face Growing Pains

Identifying the optimal mechanism of evolution for a long running detection algorithm is critically important to its long term effectiveness. For reference, the AV-TEST institute estimates that the number of new malicious programs identified daily could be as high as 350,000 [19]. If even a tiny fraction of those contribute to production of fake visual content, that is a daunting rate of change to tackle. Since Deepfake technology is capable of evolving and changing over time, so too should a good detector be capable of adapting to that change.

Retraining the detector each time a new type of Deepfake is identified, however, is not only cumbersome and inefficient, but also ineffective, since it is likely many Deepfakes of a new type will slip by before they are identified. It would be much more desirable to have a detector that can be modified with new identifying capabilities over time. However, this is not as simple a task as it may seem, as any attempt at an adaptive AI system must overcome two major hurdles: catastrophic forgetting and data hunger.

A. Catastrophic Forgetting

When catastrophic forgetting occurs, it means that new information added to a system has caused previously learned information to collapse while trying to accommodate it. As such, over time, such a detector may keep up with the newest Deepfakes, but it will leave itself vulnerable to outdated types that it was at one point capable of spotting. There have been several interesting attempts at preventing this issue [22-24].

Perhaps the most intuitive way to assuage forgetting would be to do as humans do, rehearse. Rehearsal-based methods involve keeping a subset of stored samples to pepper into the new training material, that way the model learns to keep up with both the old and new material. This is not without its downsides, though. Cycling the same stored samples can lead to overfitting in the old material, which in practice is just about as bad as forgetting it entirely. The larger the number of stored samples, the less of a threat this poses, but at the same time, the more time is spent distracted from training on new material. Another potential avenue relies on regularization. By adding a new term to the loss function, these methods aim to consolidate previous knowledge while learning with new data. However, this too has limitations, as it can impact learning rate and detection accuracy.

In an ideal environment where architecture size is of no concern, parameter isolation methods become feasible. In this environment, certain branches or parts of a model can be masked in or out for different tasks. The parts of the model that retain old information can be frozen during training to prevent forgetting. Taking this idea to its highest degree, one could even dedicate an entire copy of a model to separate tasks, with a task oracle at the head of the system designating which copy will process what data.

B. Data Hunger

To make matters worse, the problem of data hunger - the tendency for an adaptive system to require a very large number of unique samples before it can learn new information - is a major problem for a Deepfake detector. If one relies on waiting for a Deepfake type to become widespread for the system to be able to recognize it, massive amounts of misinformation could be spread in the intervening period. Detectors need to be able to learn with as few novel samples of a new type as possible. It would seem that this would require a drastic increase in the learning rate, which would only exacerbate the previous problem of catastrophic forgetting. In that sense, there is a tradeoff between stability and adaptability of an algorithm.

The balance between these two values is a difficult one to maintain, but there are some emerging methods that may allow it to be done effectively. Few-shot learning is one such method; it aims to exhibit humanlike perception of an image. Even a small child can learn what, say, a cow looks like from a single image and from just that be capable of telling cows from non-cows in the real world with high rates of success. Few-shot learning has shown some early promise, achieving above 98% accuracy at classifying 4800 classes of handwritten characters in the 1-shot case, and above 99.5% accuracy in the 5-shot case. However, this is with 28 x 28 grayscale images. The results for even 84 x 84 color images were significantly lower, at roughly 50% for 1-shot and 70% for 5-shot [25]. There has yet to be any study on this method's effectiveness in Deepfake detection. Clearly, this technology does have major limitations, but as algorithms and computational power improve, learning to identify new types of high-quality Deepfakes with just a few examples could be feasible.

V. An Envisioned Comprehensive Deepfake Detector - DeepDetect

As should be clear by this point, no perfect solution to the major problems of deep learning systems is known. To be more precise, no implementable, computerized solution is known. The distinction is important because there is still one more system of Deepfake detection that outperforms all other current algorithms in training speed, retention of learned information, and consistency. That system is the human brain [26]. Korshunov and Marcel found that humans still beat out algorithms in identifying Deepfakes, with an AUC score of about 87% compared to top algorithms which averaged around 72%. The human brain is capable of several things that machine learning systems are not as good at. Namely, humans can integrate newly learned information into their existing knowledge base with incredible efficiency. We can also imagine possible future scenarios to better prepare ourselves for what may come. Lastly, we can draw shockingly accurate conclusions from a comparatively minuscule amount of data. It seems, then, that it could be a promising avenue of research to explore development of a system that mimics the

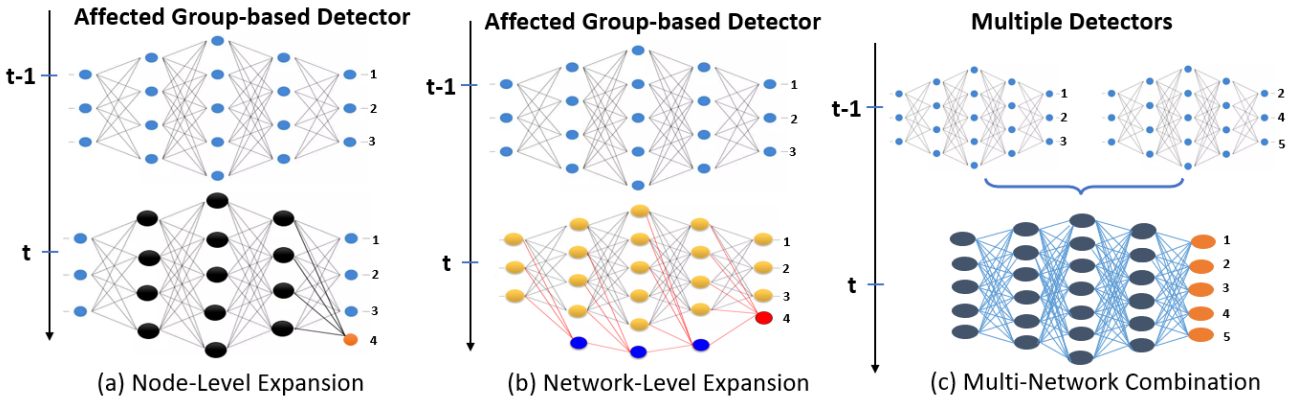

Fig. 5: An illustration demonstrating the three different modes of evolution of DeepDetect

way humans, learn, remember, and think. Following this idea, we envision the following new Deepfake Detection system, namely DeepDetect, which incorporates various techniques so as to remember the knowledge learned in the past, incrementally updated with present knowledge, and predict potential threats in the future.

A. *Remembering the Past*

To compensate for its imperfect recall abilities, the human brain has developed a keen eye for relations and patterns. It can identify objects and experiences that are similar to what it already knows and smoothly integrate the new knowledge into its memory bank. To translate this sense for relationships into a deep learning system is clearly no trivial matter, but it could prove highly beneficial to the efficiency of a long-running system. This budding subfield of research is known as group-based multi-task learning.

One open issue in multi-task learning is how to identify interrelated learning tasks automatically. In terms of DeepDetect, the challenge is to identify the related types of Deepfakes in order to leverage the combination of their collected samples for joint training. An intuitive starting point would be to develop a group identifier which partitions existing Deepfake types into groups that demonstrate similar principles and are generated by the same type of GAN or synthesizer. The latter is simple enough to implement, but from there, further study is required on both real and synthetic examples (discussed in subsection C) to identify discriminating features between groups. Some common identifiable features of modern Deepfakes include face wobble/distortion, waviness in a person's movements, inconsistencies between speech and mouth movements, erratic light and shadow, unnatural eye direction, etc. Once these features are extracted, one can define metrics to quantify contents of and inter-type similarity between groups. Doing so allows comparison between new images and existing groups, so incoming samples can not only be funneled down the optimal path through the system (discussed next), but also be integrated seamlessly into the model's perception of that group.

B. *Learning in the Present*

The next question one may ask of the above system is what happens when an incoming sample is nothing like any of the existing groups. At that point, because of how the system is structured, it is simple to extend the group identifier to allow for a new group. The average features of that group are not final, since at the start only one sample exists therein, but if and when more similar images arrive, they can be added to the group. Few-shot learning could be utilized to extract the features of these new groups as quickly and accurately as possible with the limited data.

This group-based similarity score system can be used to construct a system with several methods of evolution, hopefully allowing it to learn as effectively and efficiently as possible. These methods (illustrated in Figure 5) are: (i) Node-level expansion which absorbs new knowledge into existing nodes; (ii) Network-level expansion which increases the number of nodes in a single network; (iii) Multi-network combination which integrates deep neural networks with different specialties. In this process, dubbed DeepMixture for its expected performance at integrating knowledge, each mode of expansion serves and different purpose, and is applied in a different scenario.

In the previously discussed event that a new type of Deepfake is identified that is similar in principles to existing and known types, as measured by the group identifier, node-level expansion will be applied. The best matching group will be fine-tuned with samples from the new type. Once that is done, a new output for this group can be added to indicate the newly identified type.

If the newly identified type is instead significantly different from existing types, the process is somewhat more involved. The envisioned steps are the following: First, compare the new type T with all groups currently in the network to find the most similar one, denoted G. Though the similarity between G and T may not be significant enough to allow G to absorb T directly, few-shot learning will still be able to take advantage of their minor similarities to speed up the learning process for the new

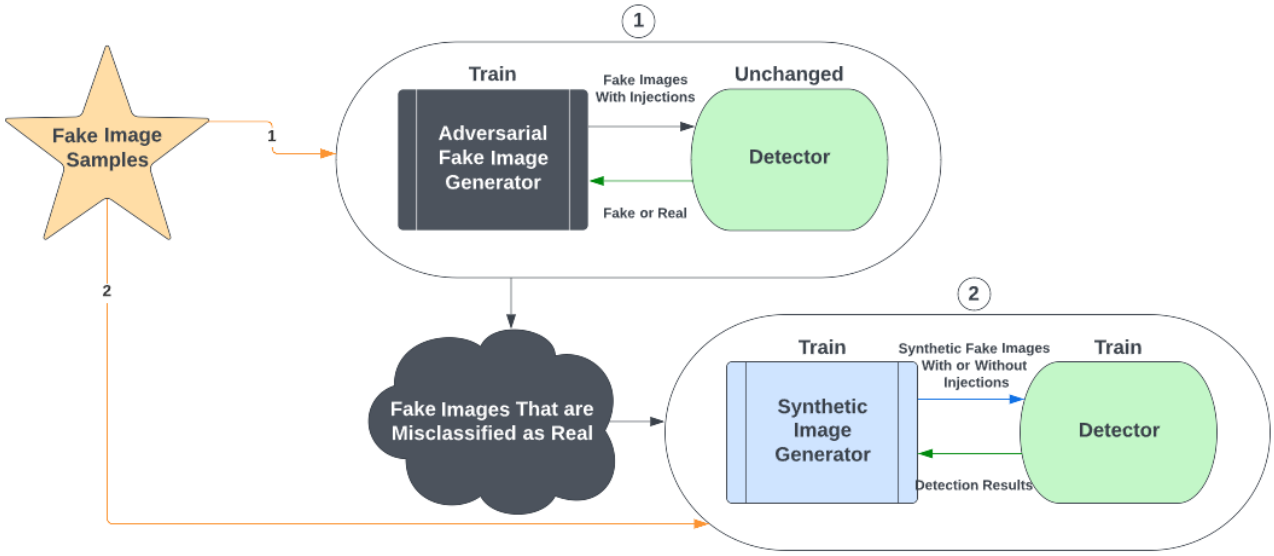

Fig. 6: Training DeepDetect to predict emerging fake image types and defend against attacks

type. Specifically, the initialization from G is used to initialize the learning of the detector for the new type T. This new detector will then be integrated into the current DeepDetect network.

The final and most aggressive way for the DeepDetect to grow quickly and expand its capabilities is to absorb other detection networks with different specialties in Multi-Network Combination. The idea is that as time passes, various other Deepfake detectors will be developed by many different groups of researchers. These detectors may, intentionally or not, specialize in detecting certain types of Deepfakes while also maintaining the ability to detect common types. If said detector is more effective in one area than DeepDetect, it could be beneficial to learn a mixture network, the outputs of which are a union of all detectors to be integrated. This integration process is not trivial as it will involve the design of algorithms for comparing networks, removing overlapping nodes and outputs, and adjusting node weights while maintaining detection accuracy. The first concrete step is to add a special output labeled "not-in-network" to each individual network. This output is used when an image appears that does not belong to any known malicious types in the current detector. Second, an algorithm must developed that can compare the network structures of the candidate detectors and identify dominant nodes for the common outputs. A similarity function will quantify the network similarity in terms of the amount of common nodes and outputs. Provided the similarity score is above a certain threshold, a network merger attempt will occur. The process will only be finalized if the resultant detector can demonstrate similar or better accuracy to before the merger.

C. Imagining the Future

One potential avenue to defend and improve the envisioned DeepDetect is to try to predict potential Deepfakes and adversarial injections in order to find the detector's possible vulnerabilities. This could be done with a dual generator system to create candidate adversarial inputs, Deepfakes with injected noise. The idea is illustrated in Figure 6. In one part of the system, the detector is used to train a generator that creates adversarially perturbed fake images with tiny changes in perturbations from the collected existing types of Deepfake images. The goal is to generate noises that will keep the sample from exhibiting any of the features of known Deepfake types. The generator is graded on its effectiveness at camouflaging fake images to be perceived as genuine. At this point, no changes are made to the detector yet. Those images that are misclassified as real are fed as input to the second generator, which would generate fake images both with and without injections for the detector to try to identify. This second generator will alternate training with the detector in a GAN environment, with the generator trying to fool the detector, and the detector working to identify both unperturbed and perturbed Deepfake images.

In the event that an adversarial Deepfake gets through the system's defenses and becomes part of the training dataset, it is imperative that it does not affect the detector too strongly. Generally, it can be assumed that such images would make up only a very small portion of the dataset, so it would stand to reason that each one would be designed to have maximum impact on the detector's training. In that case, one effective countermeasure would be to employ a network smoothing algorithm to reduce the variations of gradients from both original and adversarial Deepfakes. This makes the detector less sensitive to the occasional straggler that may taint the data. Another strategy could be to define a fingerprint vector for a training image in terms of the variations of nodes induced by the image. Then, one can explore if there is any outlier

fingerprint and conduct unlearning to remove its impact.

VI. Conclusion

In this paper we examined the potential challenges one might encounter when trying to develop a long-lasting Deepfake detection algorithm. We explored different types of adversarial attacks and how to defend against them, as well as more general hurdles faced by any system intended remain relevant in the face of an evolving adversary. We then put forward the framework for a new system that utilizes the latest advancements in multiple fields as well as novel strategies to integrate them practically. In future work, we intend to develop said algorithm and demonstrate its effectiveness not only as a defense against modern Deepfakes, but as an intelligent system capable of evolving and outpacing the opposing technology.

References

[1] N. Akhtar, A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey", IEEE Access, vol.6, pp.14410–14430, 2018.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, "Intriguing properties of neural networks", arXiv:1312.6199, 2014.

[3] I. J. Goodfellow, J. Shlens, C. Szegedy, "Explaining and Harnessing Adversarial Examples", arXiv:1412.6572, 2015.

[4] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial Machine Learning at Scale, arXiv preprint arXiv:1611.01236, 2017.

[5] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world, arXiv preprint arXiv:1607.02533, 2016.

[6] J. Su, D. V. Vargas, S. Kouichi, One pixel attack for fooling deep neural networks, arXiv preprint arXiv:1710.08864, 2017.

[7] N. Carlini, D. Wagner, Towards Evaluating the Robustness of Neural Networks, arXiv preprint arXiv:1608.04644, 2016.

[8] S. Moosavi-Dezfooli, A. Fawzi, P. Frossard, "DeepFool: a simple and accurate method to fool deep neural networks", IEEE CVPR, 2016.

[9] S. Sarkar, A. Bansal, U. Mahbub, and R. Chellappa, UPSET and ANGRI: Breaking High Performance Image Classifiers, arXiv preprint arXiv:1707.01159, 2017.

[10] M. Alzantot, B. Balaji, M. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition", arXiv:1801.00554, 2018.

[11] A. Athalye, L. Engstrom, A. Ilyas, K. Kwok, "Synthesizing robust adversarial examples", arXiv:1707.07397, 2017.

[12] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, D. Mukhopadhyay, "Adversarial attacks and defences: A survey", arXiv:1810.00069, 2018.

[13] C. Guo, M. Rana, M. Cisse, L. van der Maaten, Countering Adversarial Images using Inout Transformations, arXiv:1711.00117, 2017

[14] G. Karolina Dziugaite, Z. Ghahramani, D. Roy, A Study of the effect of JPG compression on adversarial images, CoRR, abs/1608.00853, 2016.

[15] L. Rudin, S. Osher, and E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D, 60:259–268, 1992.

[16] A. Efros and W. Freeman, Image quilting for texture synthesis and transfer, In Proc. SIGGRAPH, pp. 341–346, 2001.

[17] G. Hinton, O. Vinyals, and J. Dean, Distilling the Knowledge in a Neural Network. CoRR abs/1503.02531 (2015). arXiv:1503.02531 http://arxiv.org/abs/1503.02531, 2015

[18] N. Papernot and P. McDaniel, Extending Defensive Distillation. CoRR abs/1705.05264 (2017). http://arxiv.org/abs/1705.05264

[19] N. Papernot, P. McDaniel, Xi Wu, S. Jha, and A. Swami. 2016. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016. 582–597. https://doi.org/10.1109/SP.2016.41

[20] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical Black-Box Attacks against Machine Learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017. 506–519. https://doi.org/10.1145/3052973.3053009

[21] Hossein Hosseini, Yize Chen, Sreeram Kannan, Baosen Zhang, and Radha Poovendran, Blocking Transferability of Adversarial Examples in Black-Box Learning Systems. CoRR abs/1703.04318 (2017). arXiv:1703.04318 http://arxiv.org/abs/1703.04318, 2017

[22] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, "Continual learning: A comparative study on how to defy forgetting in classification tasks", arXiv:1909.08383, 2019.

[23] A. Mallya, D. Davis, S. Lazebnik, "Piggyback: Adapting a Single Network to Multiple Tasks by Learning to Mask Weights", ECCV, 2018.

[24] A. Mallya, S. Lazebnik, "PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning", IEEE CVPR, 2018.

[25] J. Snell, K. Swersky, R. Zemel, "Prototypical networks for few-shot learning", NIPS, pp. 4077–4087. 2017.

[26] P. Korshunov, S. Marcel, "Deepfake Detection: Humans vs. Machines", https://arxiv.org/abs/2009.03155, 2020

[27] Shehzeen Hussain, Paarth Neekhara, Malhar Jere, Farinaz Koushanfar, and Julian McAuley, "Adversarial Deepfakes: Evaluating Vulnerability of Deepfake Detectors to Adversarial Examples", Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2021, pp. 3348-3357

[28] Zhikai Chen, Lingxi Xie, Shanmin Pang, Yong He, and Bo Zhang, "MagDR: Mask-Guided Detection and Reconstruction for Defending Deepfakes" Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 9014-9023