Exploring Adversaries to Defend Audio CAPTCHA

Heemany Shekhar
Dept. of Computer Science
San Jose State University
San Jose, CA, USA
heemany.shekhar@gmail.com

Melody Moh
Dept. of Computer Science
San Jose State University
San Jose, CA, USA
melody.moh@sjsu.edu

Teng-Sheng Moh
Dept. of Computer Science
San Jose State University
San Jose, CA, USA
teng.moh@sjsu.edu

***Abstract*— CAPTCHA is a web-based authentication method used by websites to distinguish between humans (valid users) and bots (attackers). Audio captcha is an accessible captcha meant for the visually disabled section of users such as color-blind, blind, near-sighted users. Firstly, this paper analyzes how secure current audio captchas are from attacks using machine learning (ML) and deep learning (DL) models. Each audio captcha is made up of five, seven or ten random digits[0-9] spoken one after the other along with varying background noise throughout the length of the audio. If the ML or DL model is able to correctly identify all spoken digits and in the correct order of occurance in a single audio captcha, we consider that captcha to be broken and the attack to be successful. Throughout the paper, accuracy refers to the attack model's success at breaking audio captchas. The higher the attack accuracy, the more unsecure the audio captchas are. In our baseline experiments, we found that attack models could break audio captchas that had no background noise or medium background noise with any number of spoken digits with nearly 99% to 100% accuracy. Whereas, audio captchas with high background noise were relatively more secure with attack accuracy of 85%. Secondly, we propose that the concepts of adversarial examples algorithms can be used to create a new kind of audio captcha that is more resilient towards attacks. We found that even after retraining the models on the new adversarial audio data, the attack accuracy remained as low as 25% to 36% only. Lastly, we explore the benefits of creating adversarial audio captcha through different algorithms such as Basic Iterative Method (BIM) and deepFool. We found that as long as the attacker has less than 45% sample from each kinds of adversarial audio datasets, the defense will be successful at preventing attacks.**

Keywords— captcha, security, machine learning, deep learning, adversarial examples attack, basic iterative method, DeepFool

I. INTRODUCTION

CAPTCHA stands for Completely Automated Public Turing test to tell Computers and Humans Apart. It is a computer-generated challenge that should be easy to solve by humans but difficult to solve by current computing systems. CAPTCHA generators assume that humans have superior reading and hearing capabilities as compared to machines [1]. It is a web-based authentication method that authenticates a user as either a human(valid user) or a bot (attacker). A bot can be a malicious program that performs repeated tasks automatically over the internet and thus creating problems in the network.

The web is a universal medium for communication and sharing that should be accessible by every human alike. Deploying only visual captchas creates a considerable obstacle to a certain group of users, such as the visually impaired, color-blind, near sighted users [2]. This necessitates the need for more accessible captchas such as audio captcha. But, with the advancements in machine learning and deep learning it is now possible to train speech to text models that can attack audio captcha and solve them like humans. The attackers can train their own attack models or use publicly available models to create DoS, DDoS attacks that compromise the availability of the websites that use audio captchas. Hence, it is important to study and analyze the different models and techniques that can be used to break audio captcha and the ways in which these attacks can be prevented.

Adversarial Examples Algorithms are algorithms that create well crafted noise intended to fool a deep learning or machine learning model. Currently, these adversarial examples are used to attack models, hence is often referred to as Adversarial Examples Attack. These adversarial examples are crafted in a way such that the perturbations or changes to the image/audio is imperceptible to humans, but they can fool the models into classifying them incorrectly.

The main contributions of this paper are as follows:

- We created audio captchas of lengths five, seven and ten with varying background noise. Each audio captcha consists of five, seven or ten digits [0-9] spoken at random. Similar to most current audio captchas deployed at websites, each audio captcha has either no noise, medium noise or high noise in the background. This allowed us to analyze a variety of audio captchas and understand what features make an audio captcha more secure.
- We trained multiple machine learning and deep learning models to attack the audio captchas. We are able to show that generic or normal audio captchas are not secure and we use the results as a baseline for future experiments.
- Whereas normal audio captcha is generated using recorded human voice (reading out numbers or alphabets) and adding noise to it, in this paper we propose a novel method of creating audio captcha that uses adversarial examples algorithms. We call this new kind of captcha, *adversarial audio captcha*.
- We show that the newly generated adversarial audio captcha can defend against pre-trained attack models that were trained on normal audio datasets and have never

978-1-7281-4550-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMLA.2019.00192

encountered an adversarial examples before. We call this a **Level I** defense mechanism.

- Though Level I does not represent a real-world defense scenario, simulating it helps us prove that the newly generated data is highly different from the normal audio data.
- We also analyze a **Level II** defense. We re-train the attack models on the adversarial audio captcha and analyze the security provided by adversarial audio captcha.
- We are able to show that adversarial audio captcha are more resilinent to attacks as compared to normal audio captcha by simulating Level I and Level II defense mechanisms.

In Section II we briefly describe the background and previous work in the field of attacking audio captcha that helped us in getting started with the project. In section III, we describe our dataset and feature extraction process. In section IV we describe in detail the attack models and their architecture. In section V we review the concepts of adversarial examples. In section VI, we describe the proposed solution of this paper. In section VII we explain our experimental setup and results from attacking audio captcha. We also demonstrate the attack results on Level I and Level II defense mechanism.

II. Background and Related Work

CAPTCHA stands for Completely Automated Public Turing test to tell Computers and Humans Apart. CAPTCHA's idea of authenticating between a human and a bot comes from the idea of *The Turing Test* from the 1950s, hence the term, *Turing test to tell Computers and Humans Apart*. The difference was that instead of a human distinguishing between another human and a machine, in case of CAPTCHAs the machine would have to distinguish between a human and a bot hence the term *Completely Automated*. The word Public refers to the fact that the code and data should be publicly available [1].

Bots, if not identified can create huge numbers of fake accounts, spam customers, post fake or harmful comments to tarnish the image of competition, submit false online surveys or create DoS or DDoS attacks thereby harming genuine customers of the website. Hence, it is important for websites to authenticate its valid human users.

Visual captchas have evolved over time from simply distorted images to interactive images, for example many websites today ask users to select a specific object like cars, traffic signal or hills from a set of images. These images keep switching and changing as the user interacts with them. These features make attacks on visual captchas difficult. But, the audio captchas still remain in their primitive form where spoken digits [0-9] or alphabets[A-Z] are combined together with some noise. There is also the issue of understandability of the audio captcha being spoken due to difference in accents, hearing capacities and intensity of background noise in the audio captcha. This makes audio captchas a compelling target for bypassing the captcha defense. Several papers have researched upon methods of breaking audio captchas.

Tam et al. breaks the task of solving an entire captcha into sub tasks of first segmenting the audio file and then predicting spoken digits/alphabets. Segmentation and labelling are performed using manual effort for training dataset whereas a fixed size window is used for segmenting the test data audio. The project implements AdaBoost with decision stumps, SVM with RBF kernel and K Nearest Neighbors (K-NN) to predict the captchas. For an exact match condition SVM gives the highest attack accuracy of 67%. Whereas, for one mistake passing condition, SVM gives the highest accuracy of 92% among all models. K-NN gives the lowest accuracy for both one mistake and exact match passing conditions [3].

Bursztein et al. aimed to break eBay's audio captcha using their prototype Decaptcha. At the time of the experiments eBay used audio captcha with 6 digits between 0 to 9 spoken by multiple people with different genders, accents and background noise levels. Using Decaptcha the paper was able to break 75% of eBay's audio captchas [17].

Sano et al. investigated the security of Google's continuous audio captcha using Hidden Markov Models. The continuous audio captcha contained two kinds of distortion: challenge distortion and digit distortion. Challenge distortion was a static noise that was added for the entire duration of the audio, whereas digit distortion was a noise applied to each digit. The audio captcha contained nine spoken digits clustered in three groups of three digits each. The results were evaluated with off-by-one accuracy, strict accuracy (all digits labelled correctly), per-cluster accuracy, per-digit accuracy and N-segment accuracy. The off-by-one accuracy for new test data is 52% but the strict accuracy stands at 17%. The per-digit classifier does better at 76%. Sano et. al also experimented by providing the model with the ground truth number of digits in each cluster, this increased off-by-one accuracy for new test data to 58% and strict accuracy to 27% [6].

Solanki et al. proposes that off-the-shelf (OTS) speech recognition systems can be misused by attackers to break audio captchas with very high accuracy. The paper tested the speech recognition models on various popular captcha service providers such as Recaptcha v2.0, Recaptcha v2.1, Apple, etc. Google's Cloud Speech to text (US accent) performs the best with an accuracy of 98.3% on its own Recaptcha v2.0 and with an accuracy of 83.9% on Recaptcha v2.1. Recaptcha audio captcha have no noise in the background and are generally 10 digits long [2].

Through this paper we aim to analyze the security of audio captchas of varying lengths and with varying background noise. We also create a new kind of audio captcha and analyze its security as well. We are able to show that the current audio captchas are not secure whereas the newly generated adversarial audio captcha are more secure and resilient to attacks.

III. Dataset and Feature Extraction

A. Dataset

To completely analyze security of audio captcha we generated 9 different audio captcha datasets as shown in TABLE I. All audio captchas are made up of digits between 0 to 9. Broadly, the 9 datasets can be placed into 3 major buckets based on the kind of background noise in the dataset. These three buckets are : no background noise datasets, medium background noise datasets and high background noise datasets. All datasets

contain voices of 6 different speakers that were mixed together at random speeds and intensities.

No Noise Datasets: We created three datasets with no noise in the background. These three datasets contain audio captchas of length 5, 7 and 10 respectively. To created this dataset we randomly mixed spoken digits using python scripts and audio libraries.

Medium Noise Datasets: Another three datasets of length 5, 7 and 10 were created using medium background noise.To create this dataset, we used python scripts and libraries to randomly mix spoken characters with beeping sounds (high intensity noise). Also, static noise was added through the entire length of the audio captcha.

High Noise Datasets: Lastly, we created three datasets with high noise in the background and audio captchas of length 5, 7 and 10. To create this dataset, we used python scripts and libraries to randomly mix spoken characters with beeping sounds (high intensity noise). Random spoken digits were chosen and played in reverse as background noise through the entite duration of the audio. Also, static noise was added through the entire length of the audio captcha.

B. *Feature Extraction*

Mel Frequency Cepstral Coefficient (MFCC) is considered a standard method to extract features from audio and is widely used in automatic speech-to-text models [5, 6]. In this paper, we too use MFCC features from audio captcha to train our attack models.

TABLE I. AUDIO CAPTCHA DATASETS

Dataset_Names	Audio captcha datasets				
	Length	*Background Noise*	*Accents*	*Random Speed*	*Random Intensity*
NoNoise_5	5	None	True	True	True
NoNoise_7	7	None	True	True	True
NoNoise_10	10	None	True	True	True
MediumNoise_5	5	Medium	True	True	True
MediumNoise_7	7	Medium	True	True	True
MediumNoise_10	10	Medium	True	True	True
HighNoise_5	5	High	True	True	True
HighNoise_7	7	High	True	True	True
HighNoise_10	10	High	True	True	True

IV. ATTACK MODELS

A. *Machine Learning Attack Models*

We used four different machine learning models to attack audio captchas. These models were Support Vector Classifiers (SVC), Random Forest Classifier (RFC), AdaBoost with RFC as base classifier and Gradient Boosting algorithm.

Support Vector Classifier – SVC is a powerful supervised learning algorithm that is used for classification, regression and novelty detection. The goal of SVC is twofold, it first aims to find a hyperplane that linearly separates the training data. Then, it aims to maximize the minimum distance between the hyperplane and any data of the training set, that is , it tries to maximize the margin [7]. Often to achieve this goal, a kernel function is used that moves data to a higher dimension. SVCs have been used to differentiate between noise and target audio data earlier [8] implements SVC for audio stream recorded in vehicles during different driving conditions. We use SVC for multi-class classification of audio data.

Random Forest Classifier (RFC) – Random Forest Classifier algorithm is an ensemble method for classification or regression that uses decision tree as base classifier. [9] uses RFC for wildlife intruder detection by classifying audio consisting of birds, gunshots, chainsaw, tractors and human voice. We also apply RFC to our audio captcha dataset to classify digits from 0 to 9.

Adaptive Boosting Algorithm – Also referred to as adaBoost algorithm, it is used in conjunction with other machine learning models (often referred to as weak learners). AdaBoost is an adaptive learning model that uses weak learners to classify data points. The misclassified data points are assigned higher weights in the subsequent iterations till the algorithm learns to best classify all data points. [10] uses AdaBoost to classify audio data in complex audio environments. In this paper, we use AdaBoost in conjunction with RFC to classify audio captcha data.

Gradient Boosting Algorithm – Similar to RFC and AdaBoost , gradient boosting algorithm also is an ensemble method that uses weak base learners, generally decision tree with fixed size, to create stronger prediction models. This method works in stages to optimize the arbitrary loss functions that are differentiable. We experiment with gradient boost algorithms to classify audio captcha dataset.

B. *Deep Learning Models*

Convolutional Neural Networks (CNNs) have proven to perform extremely well in the field of image classification. Recent work in audio classification [11] shows promise in the field of audio classification as well. We use four different CNN models to attack audio captchas. We do not use Recurrent Neural Networks (RNNs) since RNNs are meant for time-series data where there is a relation between datapoints. In case of audio captchas, the digits are spoken randomly and have no relation among each other. Hence, we experiment with below four CNN models : cnn-1, cnn-2, cnn-3, vgg.

cnn-1 - [12] uses cnn models for small footprint keywork spotting task in audio. We implement the cnn architecture which is referred to as cnn-one-fpool and call it cnn-1. This architecture consists of one cnn layer and two dense layers followed by a softmax layer.

cnn-2 – We use a second cnn model again from [12] which consists of two convolution layers and two dense layer followed by a softmax layer. This model is similar to the model named cnn-trad-fpool3 except that we have added an additional dense layer to the architecture.

cnn-3 – We created our own cnn architecture which is an extension of cnn-2 model. cnn-3 model consists of three convolution layers and two dense layers followed by a softmax layer.

vgg – To experiment with larger cnn models we chose the VGG classifier from [11] which consists of 16 layers (13

Identify applicable funding agency here. If none, delete this text box.

convolution layers and 3 dense layers) followed by a softmax layer.

V. ADVERSARIAL EXAMPLES

Machine learning, and deep learning models are highly expressive models that learn from data. But studies [13] show that the models learn from the space rather than the individual units. Also, the input-output mapping is quite discontinuous which leaves room for introducing small imperceptible perturbation that can result in misclassification of data. Utilizing this understanding [14, 15] created adversarial examples attack algorithms that generate adversarial examples. Adversarial examples are created by adding well crafted noise/perturbations to original/clean data which are not interpretable by humans. But, these adversarial examples can fool machine learning and deep learning models to misclassify them.

In our use case, deep learning models or machine learning models are used to attack audio captcha, hence, it makes sense to explore the use of adversarial examples to defend audio captcha from attacks.

We create adversarial audio using two popular adversarial examples attack algorithms, Basic Iterative Method (BIM) and DeepFool method.

Basic Iterative Method (BIM): BIM method iteratively implements Fast Gradient Sign Method [15] to create adversarial audio. The original data is updated based on the gradient value of the loss.

$$audio_{adv} = audio_{original} + \in * sign(grad) \quad (1)$$

Deepfool Method: To create adversarial examples, deep fool method aims to minimize L2 distance but also switch classes. It iteratively finds the minimum perturbation that will result in class switch. In each iteration the method calculates L2 distance between the adversary and the original sample as shown in equation 2.

$$NextIter_{inp} = inp_{curr} + perturbation_{curr} \quad (2)$$

VI. PROPOSED SOLUTION

Through this paper we are able to prove that no noise and medium noise datasets are very easy to attack, whereas high noise audio captcha must be longer (for example of length 10) to ensure some security. Our findings show that we need more secure forms of audio captcha. To do this, we propose the following:

- We propose to create adversarial audio captcha using BIM and deepFool algorithms. These algorithms use the clean audio data (digits spoken by humans without noise) and add well crafted perturbations to them to create adversarial audio data. We then combine adversarial audio at random to create adversarial audio captchas of varying lengths (5, 7, 10)
- On these newly created adversarial audio captcha, we perform a Level I defense simulation as shown in Fig 1, where we use a pre-trained attack models that was trained on normal audio datasets and has never encountered adversarial examples before. Though the level I defense will not be sufficient in a real-world scenario, we still perform it to get a theoretical understanding and proof that the newly created audio captcha is highly different from a normal audio dataset.
- We further simulate a Level II defense against re-trained attack models as shown in Fig. 2 and 3. These attack models have retrained themselves on adversarial examples. This represents a real world scenario.

A. Level I Defense

We use Adversarial Examples algorithms such as BIM and Deep Fool to create new adversarial audio. We combine these adversarial audios at random to form audio captchas of varying lengths. Then, pre-trained models such as SVM, RFC, CNNs which have not been trained on adversarial examples are used to attack these newly created audio captchas. We hypothesize that when such pre-trained models attack the newly generated adversarial audio captcha, the attacks will fail.

B. Level II Defense

[16] shows that retraining DL/ML models on all possible adversarial examples is very costly and not completely possible. This is because huge number of adversarial examples can be created with extremely small perturbations to the original data. On the contrary, creating multiple adversarial audios using a fixed perturbation is not costly. Keeping these features in mind, we created 7 sets of adversary audio using BIM algorithm, such that each set was created using a different level of perturbation and another 7 set was created using deepFool algorithm. Using these 14 sets we created audio captchas of lengths five, seven or ten. In total, we created 30,000 adversarial audio captchas of different lengths. We propose that audio captcha created using this method will be more secure. We simulate two real world attack scenarios and study how secure adversarial audio captchas are.

Scenario 1: We set up our first experiments as shows in Fig 2, which shows the re-training phase for the attacker. The attacker can retrain their attack models on normal audio and a subset of adversarial audio (as it is costly to train on all adversarial audio sets). In this experiment, we choose two adversarial audio datasets at random for re-training along with normal datasets.

This retrained attack model is used to attack audio captcha. To create test audio captcha we again randomly select audio from any of the adversarial sets and join them to form audio captcha of length 5, 7 or 10. As shown in Fig 6, this method reduces the attack accuracy to 36% for captcha of length 5 and 25% for captcha of length 10.

Scenario 2: Next, we test the scenario where the attacker collects adversarial audio data of each type and retrains their attack models on these data. We experimented with sample sizes ranging from 5% to 80% of each adversarial audio dataset.

These samples along with normal audio dataset was used to retrain the attack models.

We then use remaining 20% audio data from each adversarial audio dataset for test. Audio is selected at random from any dataset to create captcha of length 5, 7 or 10. This method can give concrete proof of how secure the adversarial audio captcha is from attacks. This method will also help us determine worst case attack accuracies for the audio captcha.

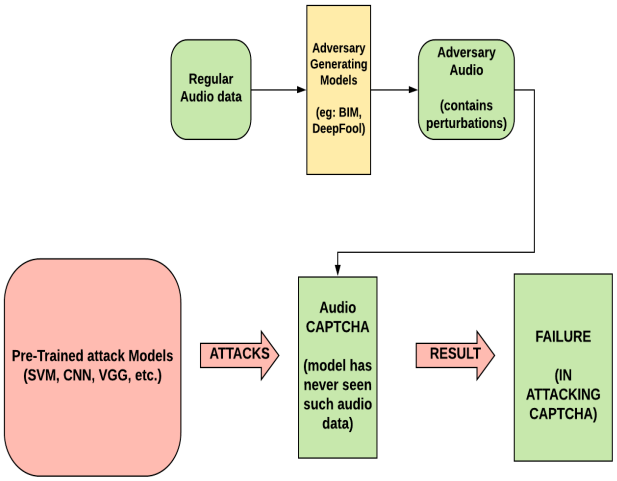

Fig. 1. Proposed Level I Defense

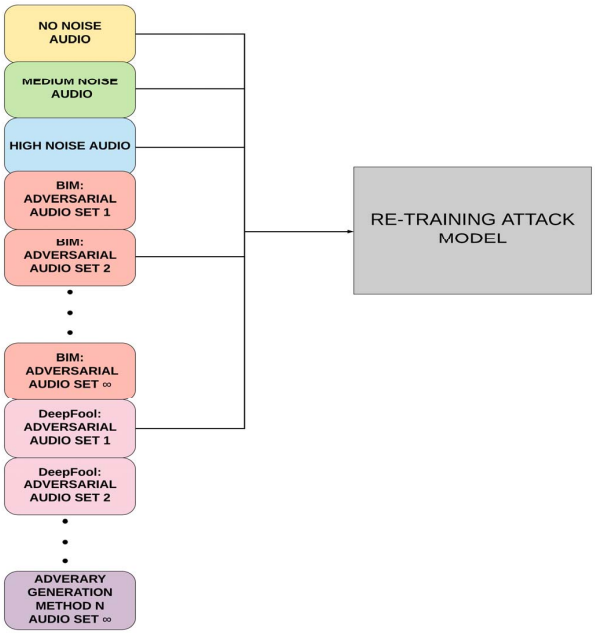

Fig. 2. Proposed Level II Defense (Re-training Phase)

VII. Performance Evaluation

In this section, we discuss our experiment results from attacking and defending audio captcha.

A. Experiment Setup

For attacking audio captchas we use SVM, RFC, AdaBoost, Gradient Boost, three versions of CNN models as discussed in section IV and VGG model. For creation of adversarial audio captcha we use adversarial examples generation methods BIM and DeepFool. The defense mechanism is shown in Fig 2. For our baseline tests on normal audio, we create 5000 audio captchas for each length of 5, 7, 10 and background noise. A total of 45000 different audio captchas were generated for baseline experiments. 14 different sets, each of around 500 to 1200 adversary audio were generated using BIM and deepFool. Audio from these 14 sets were joined at random to create 30,000 adversarial audio captchas. To run the experiments we used python, keras and pytorch for creating both attack and defense models. We used Intel® Core™ i7 with 2.80 GHz CPU and 16 GB RAM.

B. Attack Results

Fig. 3 displays the attacking accuracies of machine learning models on normal audio captchas. No noise audio captchas of any length can be attacked with 100% accuracy. Medium noise audio captcha can be attacked with 98% (length 5) to 95.64 % (length 10) accuracy. High Noise audio captcha can be attacked with highest accuracy of 62.61% (length 5) to 29.96 % (length 10).

Fig. 4 displays the attacking accuracies of deep learning models on normal audio captchas. No noise audio captchas of any length can be attacked with 100% accuracy. Medium noise audio captcha can be attacked with 98.40% (length 5) to 97 % (length 10) accuracy. High Noise audio captcha can be attacked with highest accuracy of 84.80% (length 5) to 75.25 % (length 10).

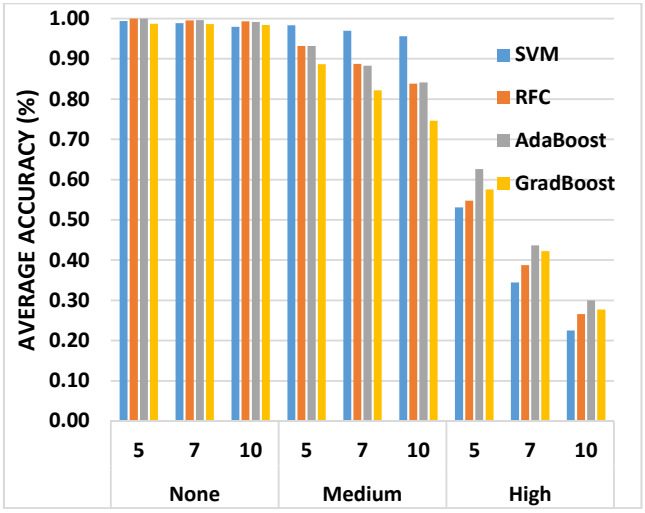

Fig. 3. Attack accuracy of ML models on normal (baseline) audio captcha

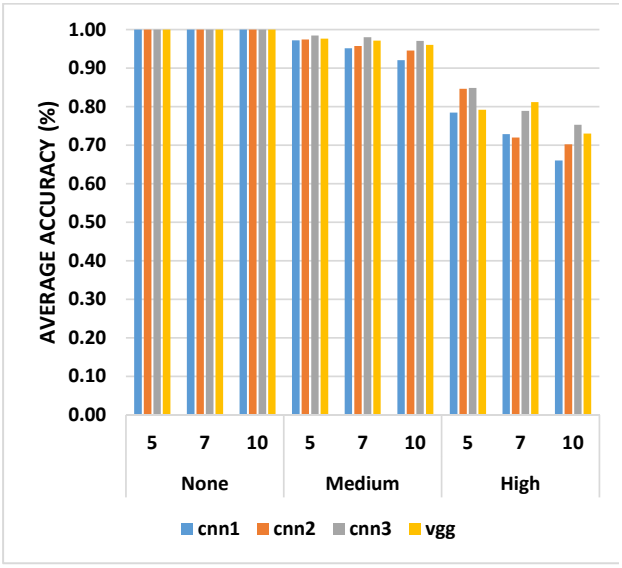

Fig. 4. Attack accuracy of DL models on normal (baseline) audio captcha

C. Defense Results

Below are the attack accuracies of ML and DL models on adversarial audio captcha We show the attack accuracies on both Level I and Level II defenses.

Fig 5. shows the attack results on Level I defense scenario, that is, pre-trained models have been used to attack audio captcha that were created using adversarial audio. All attack models are unsuccessful at attacking such audio captcha. The attack accuracy for all models is near 0%. This shows that the newly generated adversarial audio captcha is different from a normal audio captcha.

Fig 6. shows the attack accuracies on Level II – Scenario 1. For audio captcha of length 5 the attacking models could break the captcha with average accuracy of 36%. Audio captcha of length 7 could be broken with average accuracy of 31%. Whereas audio captcha of length 10 could be broken with average accuracy of 25%.

Fig 7 shows the attack accuracies for Level II – Scenario 2, with 5% sample from each type of adversarial audio the attacking accuracies are very low at 40.69% (captcha length 5) to 20.89% (captcha length 10).

An increase in attack accuracy is seen as we increase the sample size upto 55% where the attacking accuracy is 81.27% (captcha length 5) and 70.54% (captcha length 10). After this stage, adding more samples for retraining results in no major increase in attack accuracy. Even after using 80% of audio from each set for retraining the attack accuracy remains almost constant. The attack accuracy is always lower when compared to baseline datasets. This proves that Level II defense makes audio captcha more secure even if the attacker gets access to a huge sample of adversarial audio dataset and retrains their model on the same.

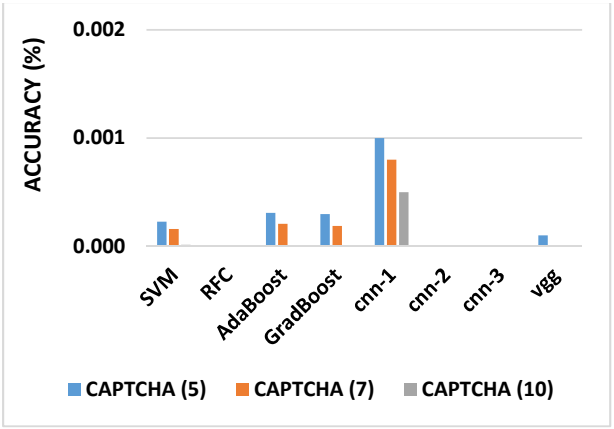

Fig. 5. Attack accuracy on Level I defense

D. User Study

Bigham et al. show that audio captchas in real world are difficult to understand by humans as well. Their surveys show that only 39% of users could correctly identify the audio captchas in the first try [18]. This shows that creating a tuff audio captcha that is not solvable by bots is not enough, these audio captchas should also be easy to understand by humans. To test this, we selected 15 people (6 female and 9 male) in the age range of 24 to 70 years and presented them with multiple audio captchas (generally 20 to 25 to each) that we had generated. We found no noise and medium noise datasets could be solved by 98% of the listeners. But, only 55% of the users could correctly identify the high noise audio captcha at their first try. Hence, though the high noise audio captcha were relatively difficult to solve by bots they were also unfriendly for the users. Lastly, we found that the adversarial audio captcha could be solved by 93% of the users at their first try. Hence, adversarial audio captcha are both user friendly and difficult for bots. We gave each user only one try to solve the audio captcha.

VIII. Conclusion

Through this work we show that audio captcha is not secure and recognize the need for more secure forms of audio captcha. We proposed a new architecture that uses adversarial audio to make audio captcha secure against attacks using deep learning and machine learning models. On performing level I and level II defense against attacks, we observed the following:

- L1 defense prevents nearly 99.9% attacks from pre-trained models.
- L2 defense prevents 64% attacks from re-trained models (on adversarial examples) for audio captcha of length 5.
- L2 defense prevents 69% attacks from re-trained models (on adversarial examples) for audio captcha of length 7.
- L2 defense prevents 75% attacks from re-trained models (on adversarial examples) for audio captcha of length 10.

Even if the attackers retrain their attack models on huge samples of adversarial datasets, their attack accuracy will still be lower as compared to their attack accuracies on baseline audio captchas. Through our experiments we prove that

adversarial audio captcha are safer as compared to normal audio captcha.

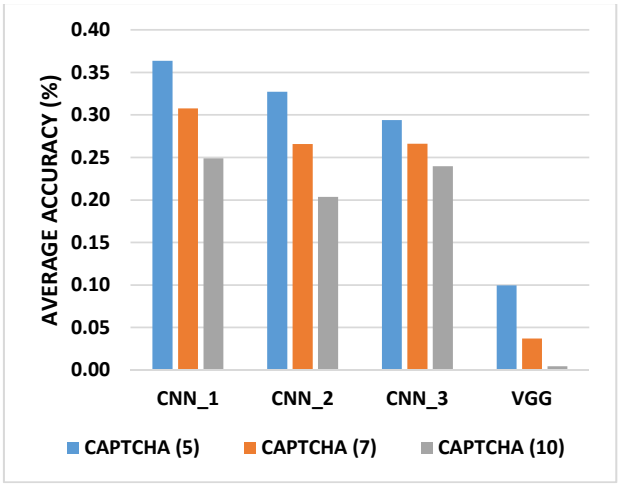

Fig. 6. Attack Accuracy on Level II Defense (Scenario 1)

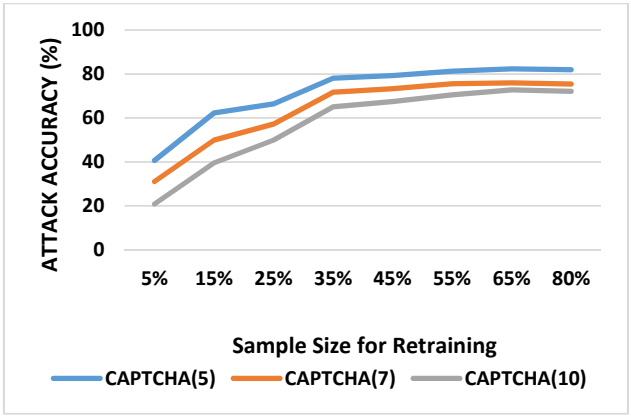

Fig. 7. Attack Accuracy on Level II Defense (Scenario-2) different sample sizes of each adversarial set.

IX. Future Work

Current audio captchas have not evolved like their visual counterparts. There is a need to create robust and secure audio captcha. In future, we would like to experiments with other adversarial example generation techniques. We would want to explore the idea of using alphanumeric audio captchas. We would also want to try ensemble attack models to further check the robustness of adversarial audio captchas. Lastly, we look forward to performing a larger survey where more users with diverse background are involved to get a better understanding of how effective the newly generated audio captchas are.

References

[1] L. Von Ahn, M. Blum and J. Langford, "Telling humans and computers apart automatically," Communications of the ACM, 47, 56-60 (2004).

[2] S. Solanki, G. Krishnan, V. Sampath and J. Polakis, "In (Cyber) Space Bots can Hear You Speak: Breaking Audio CAPTCHAs using OTS Speech Recognition," Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 2017, pp. 69-80.

[3] J. Tam, J. Simsa, S. Hyde and L.V. Ahn, "Breaking Audio Captchas," Advances in Neural Information Processing Systems, 2009, pp. 1625-1632.

[4] N. Mani and M. Moh, "Adversarial Attacks and Defense on Deep Learning Models for Big Data and IoT," in Handbook of Research on Cloud Computing and Big Data Applications in IoT, edited by Anonymous (IGI Global2019), pp. 39-66.

[5] A. Hagen, D.A. Connors and B.L. Pellom, "The Analysis and Design of Architecture Systems for Speech Recognition on Modern Handheld-Computing Devices," Proceedings of the 1st IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, 2003, pp. 65-70.

[6] S. Sano, T. Otsuka and H.G. Okuno, "Solving Google's Continuous Audio CAPTCHA with HMM-Based Automatic Speech Recognition," International Workshop on Security, 2013, pp. 36-52.

[7] K.P. Bennett and C. Campbell, "Support vector machines: Hype or hallelujah?" Acm Sigkdd Explorations Newsletter, 2, 1-13 (2000).

[8] M. Won, H. Alsaadan and Y. Eun, "Adaptive multi-class audio classification in noisy in-vehicle environment," arXiv Preprint arXiv:1703.07065, (2017).

[9] L. Grama and C. Rusu, "Audio Signal Classification using Linear Predictive Coding and Random Forests," 2017 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), 2017, pp. 1-9.

[10] D. Wu, "An Audio Classification Approach Based on Machine Learning," 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 2019, pp. 626-629.

[11] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss and K. Wilson, "CNN Architectures for Large-Scale Audio Classification," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 131-135.

[12] T. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," (2015).

[13] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow and R. Fergus, "Intriguing properties of neural networks," arXiv Preprint arXiv:1312.6199, (2013).

[14] A. Kurakin, I. Goodfellow and S. Bengio, "Adversarial examples in the physical world," arXiv Preprint arXiv:1607.02533, (2016).

[15] I.J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv Preprint arXiv:1412.6572, (2014).

[16] K. Pei, Y. Cao, J. Yang and S. Jana, "Deepxplore: Automated Whitebox Testing of Deep Learning Systems," Proceedings of the 26th Symposium on Operating Systems Principles, 2017, pp. 1-18.

[17] E. Bursztein and S. Bethard, "Decaptcha: Breaking 75% of eBay Audio CAPTCHAs," Proceedings of the 3rd USENIX Conference on Offensive Technologies, 2009, pp. 8.

[18] Bigham, Jeffrey P., and Anna C. Cavender. "Evaluating existing audio CAPTCHAs and an interface optimized for non-visual use." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2009