Fooling A Deep-Learning Based Gait Behavioral Biometric System

Honghao Guo, Zuo Wang, Benfang Wang, Xiangyang Li
Johns Hopkins University Information Security Institute
Baltimore, USA
{hguo24, mango, bwang54, xyli}@jhu.edu

Devu M Shila
Unknot.ID Inc.
Orlando, USA
devums@unknot.id

***Abstract*—We leverage deep learning algorithms on various user behavioral information gathered from end-user devices to classify a subject of interest. In spite of the ability of these techniques to counter spoofing threats, they are vulnerable to adversarial learning attacks, where an attacker adds adversarial noise to the input samples to fool the classifier into false acceptance. Recently, a handful of mature techniques like Fast Gradient Sign Method (FGSM) have been proposed to aid white-box attacks, where an attacker has a complete knowledge of the machine learning model. On the contrary, we exploit a black-box attack to a behavioral biometric system based on gait patterns, by using FGSM and training a shadow model that mimics the target system. The attacker has limited knowledge on the target model and no knowledge of the real user being authenticated, but induces a false acceptance in authentication. Our goal is to understand the feasibility of a black-box attack and to what extent FGSM on shadow models would contribute to its success. Our results manifest that the performance of FGSM highly depends on the quality of the shadow model, which is in turn impacted by key factors including the number of queries allowed by the target system in order to train the shadow model. Our experimentation results have revealed strong relationships between the shadow model and FGSM performance, as well as the effect of the number of FGSM iterations used to create an attack instance. These insights also shed light on deep-learning algorithms' model shareability that can be exploited to launch a successful attack.**

Keywords—gait behavioral biometrics, deep-learning, FGSM, adversarial machine learning, LSTM, shadow model, black-box attack, authentication

I. INTRODUCTION

A. Motivation

State-of-the-art techniques for behavioral biometrics include keystroke dynamics, mouse dynamics, gait, touchscreens, call usage patterns, and location. Typically, behavioral biometrics systems exploit a wealth of measurements made by today's devices (accelerometer, gyroscope, GPS, etc.) and can use machine learning algorithms on these measurements to build secure authentication models that can continuously and unobtrusively authenticate user. As user behaviors are hard to spoof, these techniques resist against classic user spoofing attacks (e.g., password hacks, biometric spoofing etc.). Recently, much attention has been paid to adversarial machine learning and researchers have successfully fooled such systems, e.g., Deepfake. For instance, machine learning (deep neural networks) classifiers are vulnerable to adversarial examples that are carefully designed to mislead classifiers with a small perturbation added to original input samples. The noise can be chosen by gradient based search with access to the structure and parameters of a classifier, in so called white-box attacks. However, the performance of black-box attacks against deployed models, i.e., without knowledge of the model parameters, often degrades significantly.

This research explores the possibility of applying Fast Gradient Sign Method (FGSM), a common approach used for white-box attacks on convolution neural networks, to launch a black-box attack and construct attack samples to fool a behavioral biometric system built on a Long Short-Term Memory (LSTM) recurrent neural network. We exploit the model shareability feature of deep learning algorithms and train shadow models that mimic the targeted LSTM model for this purpose.

B. Our Contributions

The novelty of our work stems from the following perspectives:

- **No knowledge of user data used to train the target model** - In typical adversarial learning efforts, the adversary has access to data from all classes. For example, in a number recognition system attack, an attacker has access to images of all numbers from one through nine. However, in this study, an adversary has no knowledge of data samples of the real user and any other users used to build the classification model being targeted by the attack. It only has access to one class of input data, i.e., adversarial data samples of a different user.
- **A black-box attack against Recurrent Neural Networks (RNN)** - Traditional adversarial attacks focus on image classification systems, which usually processes stationary data using Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), etc. Nevertheless, this study exploits the RNN architecture trained on time-series data, reflecting user walking behaviors. A strong correlation among temporal features in its input brings multiple challenges including query sampling, constructing

DOI 10.1109/SPW50608.2020.00052

inputs to the target model, and building the shadow model.

One key finding of our study is that the performance of such a black-box attack relies on the quality of the shadow model being trained. And the parameter configurations in our approach, including the number of queries allowed to generate the training instances for the shadow model and FGSM settings, in turn affect the quality of the shadow models and the performance of FGSM. With such insights, we will provide a potential explanation to some interesting phenomena during the FGSM iterations.

C. Threat Model

We consider a gait-based behavioral biometric system that uses recurrent deep learning algorithms on tri-axial accelerometer and gyroscope data gathered on a smartphone to learn a gait-based user identification model. The authentication model resides in a cloud or on the end device and in realistic settings, a user with access to smartphone applications can access a service if the user's behavioral characteristics are consistent with the model stored. We assume that the user's smartphone can be compromised by an external attacker, e.g., by exploiting a vulnerability or installing a malware, or by an internal attacker. During the information gathering phase, the attacker sends multiple authentication requests with manipulated input, without getting locked out from the end system. Once enough knowledge about the end system is gathered, the attacker carefully constructs perturbed examples to evade the behavioral biometrics authentication classifier, i.e., the target model.

We assume that the attacker does not target the cloud or the end device that is secured, and instead perturbates the input of his own according to a strategy. Thus, the attacker can record the tri-axial movement sequences of a non-real user and make perturbations before sending it to the authentication system. We extend the threat model in a paper by Papernot et al. [1] by not allowing the attacker to query the target classifier without limit, a more realistic assumption. It is important to note that the recorded data is not a subset of the data used to train the authentication model. Moreover, the confidence (a probability) of the authentication decision is accessible to the attacker, maybe through its application programming interface (API).

II. Background

Adversarial machine learning studies have seen massive growth recently [2]. There are typically two types of attacks, white-box attack and black-box attack [3]. In a white-box attack, the attacker has access to the target model, including its architecture, parameters, and even training data. While in a black-box attack, the attacker is only able to interact with the target model to certain extent, as well as with some knowledge of its architecture at most. Fast Gradient Sign Method (FGSM), a common tool used to perturb an input for desired misclassification effect was introduced by Ian Goodfellow et al. [4]. The method works by finding how much each feature in the input contributes to a change in classification result in a trained model, and adding a perturbation accordingly. However, FGSM requires complete knowledge of the machine learning model.

Most of the adversarial machine learning attacks to date have been focused on image classifiers. For example, on the well-known MNIST database of handwritten digits, adversarial attacks with small changes to the input image were shown to fool some of the best image processing neural networks with around 90% success rates [5]. For black-box attacks, an attacker with no knowledge of either model internals or training data could successfully launch attacks to DNN models remotely hosted online [1]. The attack strategy consists of training a local model, called a shadow model, to substitute for the target model, using inputs synthetically generated by an adversary and labeled by the target DNN [1]. Using a shadow model that mimics the target model helps because an adversarial vulnerability is transferable among models with the same structure [6].

Adversarial attacks have not been well studied within the context of biometric authentication. One such example is the FakeBob attack on a voice authentication system [7]. As part of this effort, we select the gait-based behavioral biometric system designed with LSTM by Shila et al. [8] to be used on mobile devices for authentication applications as our target system, and construct a black-box attack system by applying FGSM on a shadow model.

III. Technical Approach

Our experiments were implemented in TensorFlow environment, where the target model is written with TensorFlow sessions and the shadow model with the Keras package in TensorFlow. We also utilized the FGSM library offered within TensorFlow, which was implemented based on the CleverHans library [9]. We used personal laptops without GPUs for the experiments, but the codes were processed on Google Collab environment utilizing resources from Google Cloud. The computational cost is mainly with the training of shadow models, which ranges from hours to days depending on the number of training instances.

A. Dataset and Features

The dataset contained three categories of data: *real user data* of the authenticated user, *non-real user data* of 10 other users, and *adversarial data* of another different user. The real user data and the non-real user data are used to train the target model. The dataset contains raw sensor streams, Google Play API to classify user activities, and other device and application parameters. Eight features are selected to train the model: tri-axial accelerometer with its magnitude, and tri-axial gyroscope with its magnitude, while the Google API is

TABLE I. Example of One Data Slice in One Input Instance

AX	AY	AZ	AM	GX	GY	GZ	GM
-14.41670	-3.769670	2.688689	15.14200	-0.2590100	0.4557050	-0.1246200	0.5387780

used to filter the dataset so that only data collected when the user is walking is used. The sensors work under 50Hz frequency, and the target model takes 200 continuous slices of data, which is 4 seconds worth of data. We extract data with the step set to 10, e.g. from 0 to 200, 10 to 210, 20 to 220 and so. Therefore, each input has the form of 200x8 dimensions. Table I shows a sample of one slice of data with 8 dimensions in one input instance of 200 slices.

B. Terminologies

Before going into the details of our experiment design, we define some terminologies that are going to be used when describing our experiments. First, use of "negative" and "positive" in classification can be ambiguous depending on the application context. Therefore, "acceptance" and "rejection" are used for the two classification classes of an authentication system. If a real user input instance is denied by the authentication system, it is a *false rejection* case. On the contrary, if a non-real user input instance is authenticated by the system, it is a *false acceptance* case. Our study is interested in false acceptance cases.

Second, the output of the target and shadow models is in the format of a confidence level between 0 and 1 of a class. In our experiments, FGSM tries to lower the confidence on being the non-real user class for an attack instance, which is the probability that authentication is to be denied. So, we use the term *denial probability* to refer to the output of models. The goal of the attacker is to lower this denial probability as much as possible.

Last, different types of data are used in several steps of each experiment:

- *Queries* are inputs to the target model to create training instances used for a shadow model, which are randomly sampled from the adversary data.
- *Training Instances* are the above queries together with their corresponding outputs (denial probabilities) generated from the target model. Training instances are used to train the shadow models.
- *Adversarial Instances* are inputs to the shadow model, also randomly sampled from the adversary data. Our goal is to perturb these instances to create attack instances.
- *Attack Instances* are created by FGSM working on the shadow model by modifying the adversarial instances. As inputs, attack instances are fed to the target system in a hope to trigger false acceptance.

Note that we generated training and adversarial instances by randomly sampling the adversarial data of an actual user. This improves the efficiency compared to sampling the whole input space that has large regions not representing any possible human users.

C. The Target LSTM Model

The target model is implemented using TensorFlow. It has two output classes: real user or non-real user, and 64 hidden units. This model chooses the Rectified Linear Unit (ReLU) as the activation function and stacks two LSTM layers with 64 hidden units. The output layer uses SoftMax function. This model chooses L2 loss as its loss function. L2 loss is used to optimize the regular term in the objective function to prevent parameters from being too complex and easy to overfit. Although this is not a very sophisticated model, it is suitable for this study that explores the feasibility of adversarial attacks.

In training, the model takes an input vector with 8 features for each perceptron, and recurrently takes in 200 slices to generate a classification based on their accumulated outputs. The step size between two consecutive classifications is 10 slices, which corresponds to 0.2 seconds of observation. We train and test the model with the data of the real-user and 10 non-real users. We used 80% of the dataset for training the model and the rest 20% of the dataset to test the model. We use batch gradient descent with a batch size of 100 and Adam optimizer with a learning rate of 0.0025 for optimization, and run 20 epochs to train our model. The continuous authentication threshold is set at 0.5 for an input, which means an alert will be triggered only if its denial probability is higher than 0.5. The overall classification accuracy of the trained model is higher than 99% on the testing data.

D. The Shadow Model

The attacker only has the knowledge that the target model is an LSTM model and can receive the denial probability of its query to train a shadow model. For this shadow model we choose a vanilla tf.keras.layers.LSTM model with 200x8 input form and 1x2 output form, and everything else is set to default. We also use SoftMax activation function to ensure that the output probability is between 0 and 1.

E. Experimentation Framework

As shown in Fig. 1, there are four control parameters in this framework that could affect the experiment result.

Step 1. At the beginning, a set of queries are generated and fed into the target model to get their corresponding denial probabilities, referred to as Denial Probability 1 (DP1).

Here we introduce the first parameter, the number of queries used. According to the threat model, a small number of queries needed means the attack is easier to be implemented, or the system is more vulnerable. On the contrary, a large number of queries required implies the system is more secure. For example, in practice, there is usually a security threshold configured for such kind of authentication systems, which is the maximum number of failed authentication attempts that can be tolerated. If the number of rejections exceeded this threshold, the system would be locked.

Intuitively, fewer queries to the target model and resulting training instance for the shadow model would result in less knowledge being revealed about the target model and transferred to the shadow model. More queries could better explore the input /output relationships, so that the shadow model would mimic the target model for a higher chance to launch a successful attack. However, the sampling strategy is also important. Counter-intuitively, simply increasing the training data without representative samples of the feature space actually runs the risk to result in a more biased shadow model. Apart from the considerations of the security threshold and the shadow model performance, a larger number of

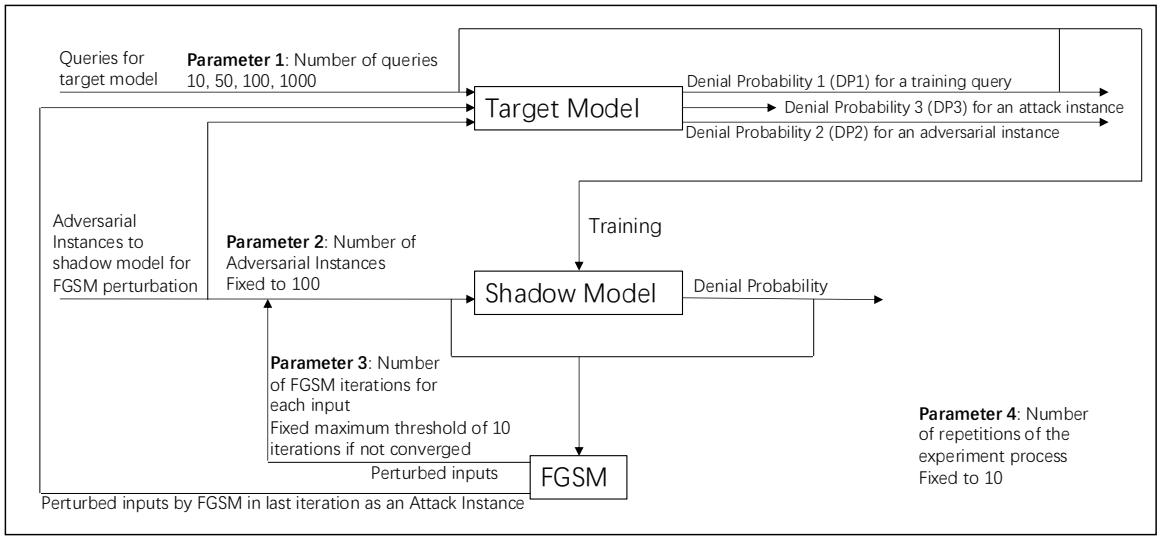

Figure 1. Experimentation Process and Steps.

queries also lead to a greater computational cost to run the experiment, mainly in training the shadow model. In our experimentation, we pick 10, 50, 100, and 1000 as the candidate values of this parameter.

Step 2. We use the training instances, i.e., queries and their corresponding denial probabilities calculated by the target model, to train the shadow model.

Step 3. In this step, we apply FGSM using the shadow model and a set of adversarial instances to create attack instances. FGSM iteratively derives the gradient of each feature and the desired perturbation to each adversarial instance.

Here we introduce the second and third parameters in the experimentation. The second parameter is the number of adversarial instances we feed into the shadow model. The purpose of using a number of adversarial instances is to increase the chance of success by creating multiple attack instances, since FGSM may not be equally effective on different adversarial instances. FGSM is applied to each adversarial instance one at a time. For simplicity, we fix this parameter to 100 in our experiment.

The third parameter is the maximum number of iterations that FGSM will be run for each adversarial instance. In each iteration, FGSM perturbs the current input slightly in the hope that the corresponding denial probability predicted by the shadow model would be reduced. However, we need to decide when to stop this loop so to output the resulting perturbed instance as an attack instance to test the target model. This parameter is decided in an empirical way. Going through a series of tests of FGSM processing randomly selected adversarial instances, we notice that in most cases, the denial probability on their perturbations converges within 5-6 iterations. And if that is not the case, it seems very likely that this denial probability does not converge at all. Based on this observation, we relax the computation cost a little and set this parameter to 10. So, for each adversarial instance given to the shadow model watched by FGSM, the iteration stops either when the denial probability predicted by the shadow model converges or if the number of iterations reaches 10. Then the perturbed input in the last iteration becomes the output of FGSM, i.e., an attack instance.

Step 4. We take the 100 initial adversarial instances and feed them into the target model and record their denial probabilities, referred to as Denial Probability 2 (DP2). It is used to compare to that of the corresponding attack instance (see below).

Step 5. Finally, we take the 100 attack instances generated from FGSM and feed them to the target model to get their corresponding denial probabilities. This probability is referred to as Denial Probability 3 (DP3).

F. Performance Measures

We use two performance measures in the result of every experiment:

- We record the lowest score in DP3 on the target model for the 100 attack instances, as the best performance that the shadow model and FGSM achieve.
- We calculate DP2-DP3 for each pair of adversarial instance and attack instance, denoted by *delta*, and take the average of these 100 *delta* values. This measure represents on average how much the shadow model and FGSM help in lowering the denial probability over adversarial instances. If the approach helps, we expect the average *delta* to be positive. Otherwise, the approach is largely not helping.

There is one last parameter used: the above process is repeated 10 times for each setting of the number of queries in order to train the shadow model. The outcome of one single experiment following this process may not be statistically meaningful. Thus, we need to repeat the whole process several times. So, to illustrate the performance, we calculate the average of the 10 lowest denial probabilities as well as the average of all the *delta* values of these 10 repetitions of experiment.

IV. Experiment Result and Analysis

A. Experiment Results

The experiment results for query sizes of 10, 50, 100 and 1000 are shown in TABLE II. We further plot how the average lowest DP3 and the average *delta* change with the number of queries in Fig. 2. And notice that we invert the Y-axis for the average *delta* to show the relationship between these two measures more intuitively. Among the four settings, query size 1000 achieves the lowest average lowest DP3 and the highest average delta. Actually, only its average *delta* is positive, just slightly above zero, while the other three settings yield negative values in average *delta*. This shows the challenge that FGSM faces to create attack instances that can perform better on the target model than original adversarial instances.

TABLE II. Experiment Results

Number of Queries	Average Lowest DP3	Average *Delta*
10	8.749E-01	-3.600E-03
50	8.400E-01	-7.266E-03
100	9.158E-01	-5.975E-03
1000	5.616E-01	1.434E-10

Based on Fig. 2, we suspect that more queries likely can improve the quality of the shadow model to be trained. Such a shadow model can possess relationships between input and output that are more similar to the target model. And if the shadow model has higher quality, more likely the FGSM method generates attack instances that can lower the probability being rejected for authentication on the target model, with improvement over the original adversarial instances. However, it is not abundantly obvious to draw a clear conclusion.

So, we take a closer look at individual shadow models in experiment repetitions. Fig. 3 and Fig. 4 use two Y-axis, which are the lowest DP3 and the average *delta*. X-axis is each shadow model trained in one experiment repetition. We only show the plots for query sizes 10 and 1000 since others show similar patterns.

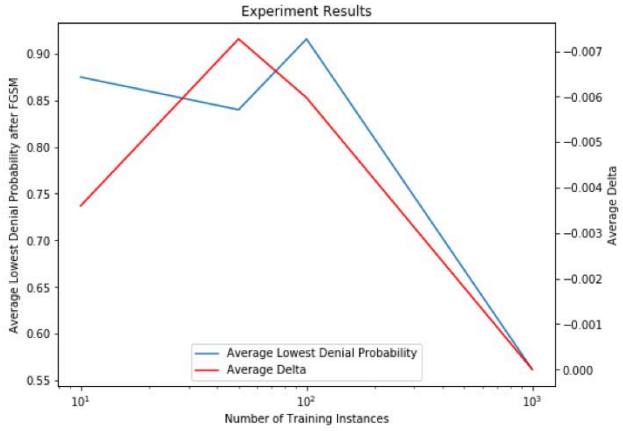

Figure 2. Average Lowest DP3 and Average *Delta* vs. Number of Training Queries.

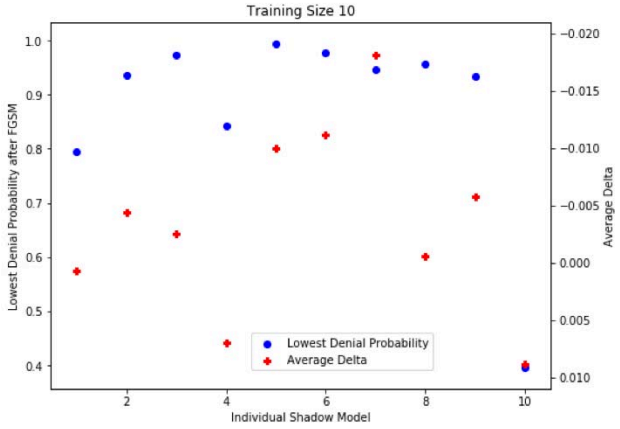

Figure 3. Performance of Individual Shadow Models for Query Size 10.

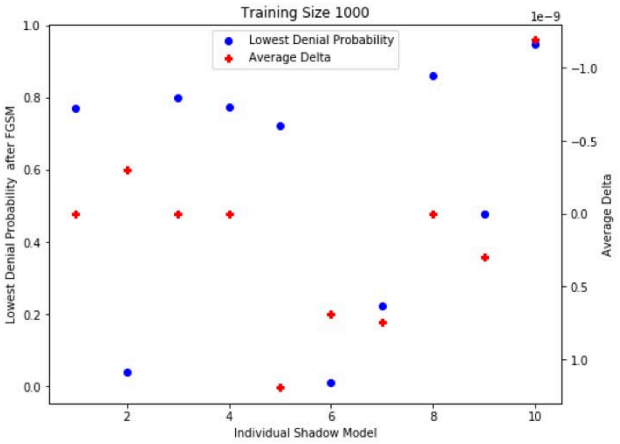

Figure 4. Performance of Individual Shadow Models for Query Size 1000.

As in Fig. 3, for most shadow models using only 10 training queries, when the average delta is higher, the lowest denial probability is lower, a clear correlation between these two measures. If authentication requires lower than 0.5 in the denial probability for continuous authentication, one shadow model (#10) succeeds in creating attack instances to fool this system.

Fig. 4 shows the same pattern for shadow models using 1000 training instances. Overall, more models achieve better performance with success than those in Fig. 3. For example, the lowest DP3 among all the shadow models is around 0.04.

To conclude, applying FGSM on a shadow model, we observe a strong relationship between the lowest denial probability that attack instances can achieve and the average *delta*. It is an indicator of how closely a shadow model "represents" the target model. Moreover, the risk of this type of black-box attack to succeed does exist, with a caveat that more training queries/instances allowed can increase its chance.

B. FGSM Performance during Iterations

We choose the experiments using 1000 training queries, seeming the best among all, to look into the FGSM process. We further choose arguably the best shadow model that has the highest average *delta* value. One adversarial instance goes through FGSM iterations, during which we keep adding perturbation to this instance. Intuitively, we select the adversarial instance which goes through 10 FGSM iterations and generates the lowest denial probability on the shadow model among all 100 adversarial instances.

As in Fig. 5, the denial probability of this instance on the shadow model rapidly decreases through the 10 FGSM perturbations. At the end, the denial probability output from the shadow model is close to 0.05. On the target model, its denial probability is around 0.95 at the beginning. However, through FGSM iterations, this probability falls at first but then goes up after five FGSM perturbations. The lowest denial probability ever achieved is around 0.64. Although only one adversarial instance is demonstrated here for ease of illustration, we can see that the FGSM method does work when the shadow model is similar to the target model.

We try to understand the counter-intuitive phenomenon that the denial probability on the target model falls firstly and then rises back. One possible reason is that too many iterations of FGSM applied can "over-perturb" the instance due to the fact that the shadow model is not the same as the target model. At first, since the shadow model is somewhat similar to the target model, FGSM processing an adversarial instance with the help of the shadow model can generate a perturbed instance that moves into a region belonging to the desired class in the target model. So, when we add small perturbations to the adversarial instances at the beginning, not only can we decrease the denial probability on the shadow model, but also the denial probability on the target model. After adding a few more perturbations (in this case, the number is 5), the denial probability on the shadow model is still decreasing, indicating the instance stays in the real-user class region in the shadow model. However, the denial probability on the target model starts to increase, since the perturbed instance starts to leave the region of real-user class in the target model. At the end of the day, the shadow model only mimics the target model to certain extent but not perfectly.

Fig. 6 shows the above scenario using a conceptual illustration of simplified behavioral biometric data (not actual input). The two models of a real-user are represented by two circle areas where their intersection is the similarity of these two models. Instances outside of the area of a model have high denial probability predicted by that model. Within the model's circle, the closer the instance to the center of the circle, the lower predicted denial probability this instance gets from this model.

In this figure, FGSM starts with a black dot at the bottom that is a randomly sampled adversary instance and fed to the shadow model. In each iteration of FGSM, this instance gets closer, in general, to the center of the shadow model. A new instance after perturbation is shown as another black dot along a path directed toward the center of the shadow model. Initially, a perturbed instance also gets closer to the center of

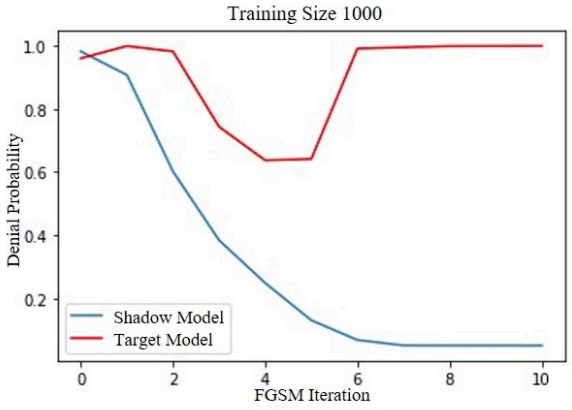

Figure 5. Denial Probability Change during FGSM Iterations on the Shadow and Target Models.

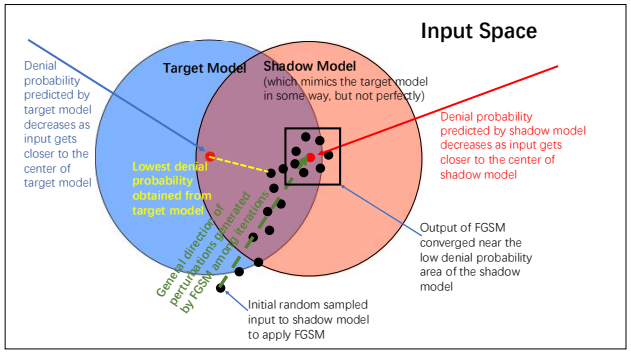

Figure 6. Venn Diagram of the Target and Shadow Models to illustrate the Denial Probability Change during FGSM Iterations.

the target model once it enters the intersection area of the shadow model and the target model. So, its denial probability predicted by the target model also decreases. Eventually the instance reaches one point that is closest to the center of the target model, which generates the lowest possible denial probability on this specific shadow model. But this is not yet the lowest possible denial probability on the shadow model, so the FGSM continues. However, now these new perturbed instances turn away from the center of the target model further and further, resulting in rising denial probabilities. Again, this is because the shadow model is not a perfect replica of the target model. If the instance is very close to the center of the shadow model, it cannot be that close to the center of the target model.

V. Conclusion

Our study has shown that AI security applications based on deep-learning algorithms (and classification models in general) are vulnerable to black-box attacks. However, such exploits highly depend on the quality of the trained shadow model in order to enable effective attack input generation through FGSM. As such, it is critical to install guards around core machine learning models, so an adversary is limited in

knowledge that he is able of gathering. Insights from this work provides guidance to future research in this subject.

ACKNOWLEDGMENT

The dataset and LSTM code for the target model used in this project are provided by Unknot.ID Inc.

REFERENCES

[1] N. Papernot *et al*, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security,* 2017.

[2] (Jun 15,). *A Complete List of All Adversarial Example Papers*. Available: https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html.

[3] L. Huang *et al*, "Adversarial machine learning," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence,* 2011.

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

[5] A. Madry *et al*, "Towards deep learning models resistant to adversarial attacks," *arXiv Preprint arXiv:1706.06083,* 2017.

[6] F. Tramèr *et al*, "The space of transferable adversarial examples," *arXiv Preprint arXiv:1704.03453,* 2017.

[7] R. Xie *et al*, "A Deep, Information-theoretic Framework for Robust Biometric Recognition," *arXiv Preprint arXiv:1902.08785,* 2019.

[8] D. M. Shila and E. Eyisi, "Adversarial gait detection on mobile devices using recurrent neural networks," in *2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE),* 2018.

[9] (Dec 16,). *CleverHans Blog*. Available: http://www.cleverhans.io/.