Doruk ÖZER
70192

Assignment 3

(a)

The one of the assumptions made in 2.3 states that according to Ronisnon's method presence of tokens of words in an email affect the mail's spam status independently. That means, words are independently treated from each other while the scores for the words are calculated. According to that formulation, score of a word depends on three things: the number of occurrences of words in the training data, the number of spam emails that includes the word and number of legitimate emails that includes the word. So if I add a different word to my email which is w that word wont increase the spam score that is produced for the word u since they are treated independently.

(b)

During training, since the tokens of words in an email affect the mail's spam status independently, the best strategy to use while attacking is to add additional words in attack e-mail to maximize expected spam score. So during test time, if a company sends a legitimate e-mail eventough the legitimate email doesn't includes the words added by attacker the mail can go to spam folder.

(c)

If the attacks based on attacker's knowledge about certain information about the distribution of the incoming mails, causes an increase on the spam score of both ham email and spam e-mail. So Dynamic Threshold proposes two thresholds which are dynamically adjusted to distinguish ham and spam mails after the distribution based attacks. By this adaptive threshold scheme, attacks which would shift all scores (the scores of both ham and spam email) will not be effective because rankings are changeless to such shifts.

(d)

The results of dictionary attack results shows that the attack emails makes small amount of number of messages inbox while the attack emails makes large percentage in terms of tokens. So token-based outlier detection can be used to defend against this paper's attack

(e)

This optimization is trying to achieve, to optimization of the of adversarial samples where the attacker only has black-box access to the model.

The optimization function F is the function of functionality-preserving manipulations that can be applied to the input program x,which is what we want to minimize. The process is based on the classification output on the manipulated program which is function of the manipulations added on the the malicious input to the program. There is also a penalty term that punishes the number of bytes that is injected and works as a regularizer which helps to the minimization process. The minimization also

applied according to the number of queries that potentially can be executed since the access to the black box model can be limited.

Inputs :
s -> functionality-preserving manipulations that can be applied to the input program x
S -> e a set of k distinct functionality-preserving manipulations that can be applied to the input program x
F(.)-> the objective function (it is the function of s)
f(.) -> the classification output on the manipulated program
x -> malicious input program
⊕ -> the function that applies the manipulations described by s to the input program x, preserving functionality, and returns the manipulated program.
ƛ -> an hyper parameter that tunes the tradeoff between number of bytes injected and reduction of the probability
C(.) -> penalty term that can be considered as regularizer
q -> number of queries
T -> upper bound of number of queries can be performed

(f)

Query efficient implies that the attacker shouldn't be making the transformations to the input malware randomly instead they should implement it querying semantically.

Functionality preserving implies that we should devoid of altering the execution traces as it relies on the injection of benign content that passes sandboxing.

(h)

The paper shows that their attacks can successfully transferable to the other commercial antivirus solutions as it evaded 12 commercial antivirus engines. In addition, by optimizing the the attack against those antivirus tools they expect to be more effective . Because the antivirus engines are black box models that can be evaded by injecting payload with query efficient way and functionality preserving way so that antivirus softwares are evaded.

Question 1

	DT	LR	SVC
0.05	1	1	1
0.1	1	0.979	0.979
0.2	0.959	0.979	0.979
0.4	0.918	0.959	0.979

It is observed that accuracy decreased as the percentage of the labels to be flipped increased for all of the models. However label flipping attack affected differently in terms of magnitude to different models. Based on the table the most vulnerable model to the label flipping attack is DT while SVC is the least impacted model (there is slight difference as the p=0.4). So SVC is the most robust model to the attack while DT is the most vulnerable model to the attack

Question 2

	0.99	0.98	0.96	0.8	0.7	0.5
Recall	0.43	0.52	0.59	0.81	0.93	1

Based on the experimental results (table) it is observed that as the threshold probability value decreased, recall value increased so they are negatively correlated.

Since we are working on training dataset, recall values makes sense. Because FN + TP remains the same since all of the samples are actually included in the training dataset so we can observe the selectivness of the threshold filter and reason about it

The relationship between threshold and recall can be explained trivally if we look at the case threshold equals to 0.5. That predicts all of the samples are included in the training dataset. As the threshold value increases number of samples that passes through the filter decreases, as a result number of true positives remains the same.

The recall means in this context is the ratio between the amount of sample that the attack correctly predicts that it is a member of the training dataset, and sum of the samples that model correctly predicts that is a member of training dataset and the amount of samples the attack incorrectly predicts that it is not a member of the training dataset.

Question 3

	DT	LR	SVC
0	0	0	0
1	1	0	0.6
3	1	0.2	0
5	1	0.4	0.2
10	1	1	0.2

I tried to inject the trigger pattern of having FWI value as 4 and make the model predict samples which have FWI value 4 as 1. In order to do the create backdoor attack, i take num_samples amount of unique samples which are normally predicted as 0. Because I want to make the model predict an instance actually labeled as 0 as if it was labeled as 1 with the trigger pattern. So I took those samples and made its FWI values as 4. Later I added those samples to the training dataset so that the model is trained with the benign data included with the backdoored data. Later I conducted the experiment by taking the 5 random samples normally predicted as 0 from training data because I want to predict the data containing the trigger pattern to be classified as 1 . From those 5 samples I added all of them the trigger pattern so I made all of those samples' FWI value to be 4. Later, I made my predictions by using the backdoored model. Latet I divided how many of those samples predicted as 1 with the number of backdoored samples which is 5 that defined my succes rate.

Question 4

```
Avg perturbation for evasion attack using DT : 1.432549999999949
Avg perturbation for evasion attack using LR : 1.8714499999999297
Avg perturbation for evasion attack using SVC : 2.0744499999999215
```

This screenshot show the avg. perturbations which are below the limits.

My attack strategy is as follows. I conducted a manuel experiment to find the features which are the most correlated with the labels of the samples. I found out that Ws, Rain, FFMC were the most correlated with the labeling of the data. Later I modified the adversarial example's Ws, Rain, FFMC features until the label of the data changes. I manipulated the features as follows; I decreased the Ws

value and the FFMC value, and increased the Rain value if the actual label is 1 I did the opposite if the actual label is 0. Because the Ws and FFMC features were positively correlated with labeling the sample as 1 while Rain feature was negatively correlated with labeling the sample as 1.

Question 5

My experiments showed that

```
The transferibility from  DT  to  LR  is  12.5
The transferibility from  DT  to  SVC  is  17.5
The transferibility from  LR  to  DT  is  50.0
The transferibility from  LR  to  SVC  is  32.5
The transferibility from  SVC  to  DT  is  45.0
The transferibility from  SVC  to  LR  is  70.0
```

The numerical results are in terms of percentage (12.5 means 12.5 percentage i.e)

Based on my results the evasion attack has low cross-model transferability. However, it is seen that from SVC to LR there is %70 transferability which may seem reasonable. What I mean is I might create a SVC model in my local server than if the remote server uses LR model then my evasion attack may be %70 efficient in the remote server that uses LR model. However, the transferability in other scenarios is not more than %50 so I can conclude my model has low cross-model transferability.

Question 6

	DT	LR	SVC
6	0.959	0.918	0.693
12	0.979	0.877	0.775
16	0.979	0.897	0.918
20	0.979	0.979	0.959
24	0.979	1	0.979

I observed that as the number of examples increased the accuracy of the models increased so they were positively correlated. Because the ML models tends to learn more and learn to generalize and not overfit as the number of data that is used in training increases. So more the data it is trained more the accurate the models become, and experiments also showed what we were expecting