بنام خدا

گزارش تمرین دوم

موضوع تمرین:

تحلیل داده مربوط به DDOS

استاد:

دکتر تاجبخش

سینا زینالی بایرامی

۱۴۰۴۶۹۹۱۱۱

زمستان ۱۴۰۴

فهرست مطالب

فصل اول

مقدمه

با گسترش اینترنت اشیاء، مشکلات امنیتی مرتبط با این تکنولوژی نیز افزایش یافت. از آنجا که دستگاه‌های IOT معمولا سطح امنیت پایین‌تری دارند، اهداف مناسبی برای اجرای حمله DDOS هستند. در این حملات با ارسال حجم زیادی از ترافیک بیهوده، موجب از کار افتادن سرویس هدف می‌شوند.

بررسی داده‌های شبکه با استفاده از ماشین لرنینگ، یکی از روش های شناسایی و classification حملات محسوب می شود.

هدف این پروژه تحلیل داده‌های مربوط به حمله DDOS در IOT با استفاده از دیتاست BOT-IOT و بررسی عملکرد سه مدل درختی مختلف است.

معرفی دیتاست

در این تمرین از دیتاست BOT-IOT استفاده شده است که شامل ترافیک شبکه شبیه‌سازی شده در محیط IOT است. طبق تعریف تمرین از فایل csv موجود در فولدر ۵٪ استفاده شده است. نام فایل نیز UNSW_2018_IoT_Botnet_Full5pc_2.csv بود. این داده ها شامل حدود یک میلیون نمونه و ۴۶ فیچر است که شامل اطلاعاتی مانند تعداد پکت‌ها، حجم داده، نرخ ارسال و ... بود. (شکل ۱-۱)

```
# Basic info
print(df.shape)
df.head()
```

```
C:\Users\szb20\AppData\Local\Temp\ipykernel_18300\2020635744.py:6: DtypeWarning: Columns (7,9) have mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('../data/UNSW_2018_IoT_Botnet_Full5pc_2.csv')
(1000000, 46)
```

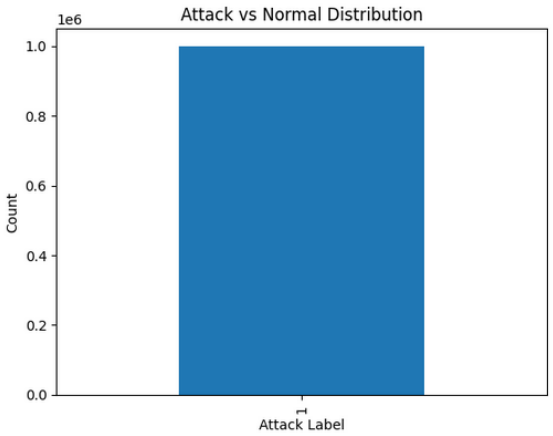
	pkSeqID	stime	flgs	flgs_number	proto	proto_number	saddr	sport	daddr	dport	...	AR_P_Proto_P_DstIP	N_IN_Conn_P_DstIP	N_IN_Conn_P_SrcIP	AR_P_Proto_P_Sport	AR_P_Proto_P_Dport	Pkts_P
0	1000001	1.528085e+09	e	1	udp	3	192.168.100.148	37153	192.168.100.6	80	...	0.319943	100	100	0.319979	0.319943	
1	1000002	1.528085e+09	e	1	udp	3	192.168.100.148	37154	192.168.100.6	80	...	0.319943	100	100	0.319979	0.319943	
2	1000003	1.528085e+09	e	1	udp	3	192.168.100.148	37155	192.168.100.6	80	...	0.319943	100	100	0.319979	0.319943	
3	1000004	1.528085e+09	e	1	udp	3	192.168.100.148	37156	192.168.100.6	80	...	0.319943	100	100	0.319979	0.319943	
4	1000005	1.528085e+09	e	1	udp	3	192.168.100.148	37157	192.168.100.6	80	...	0.319943	100	100	0.319979	0.319943	

(شکل ۱-۱) – اطلاعات داده ها

لیبل های موجود در این دیتا به صورت زیر بود:

- Attack: حمله یا عدم حمله
- Category: نوع حمله
- Subcategory: زیرنوع حمله

در بررسی اولیه داده ها مشخص شد که فایل انتخاب شده، صرفا شامل ترافیک حمله بود. بنابراین مدل ماشین لرنینگ مسئله classification دوتایی بین دو نوع حمله DOS و DDOS است. (شکل ۲-۱)



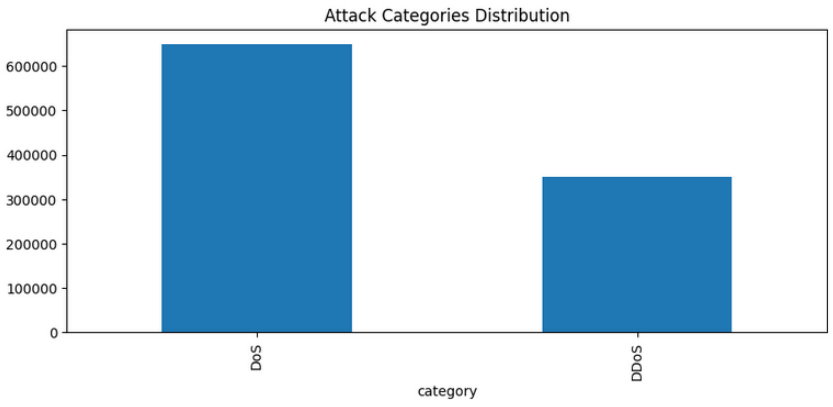
(شکل ۲-۱) – دسته بندی حمله یا عدم حمله

فصل دوم

یکی مراحل preprocessing، بررسی توزیع داده ها است. عدم توازن داده ها می‌تواند باعث جهت گیری مدل به یک کلاس خاص بشود.

Imbalance data

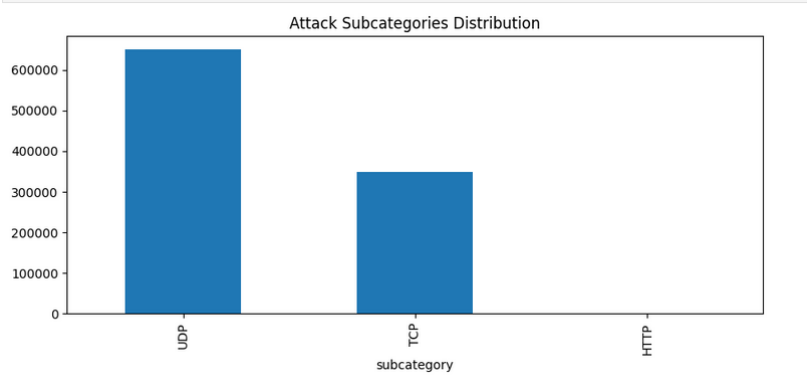
در ابتدا توازن داده ها بر اساس لیبل category بررسی شد. (شکل ۱-۲)



(شکل ۱-۲) دسته بندی نوع حمله

در نمودار مشاهده می‌شود که دیتا شامل دو کتگوری DOS و DDOS است که توزیع آن نیز کمی نامتوازن است.

در ادامه این توازن براساس لیبل subcategory مورد بررسی قرار گرفت. (شکل ۲-۲)

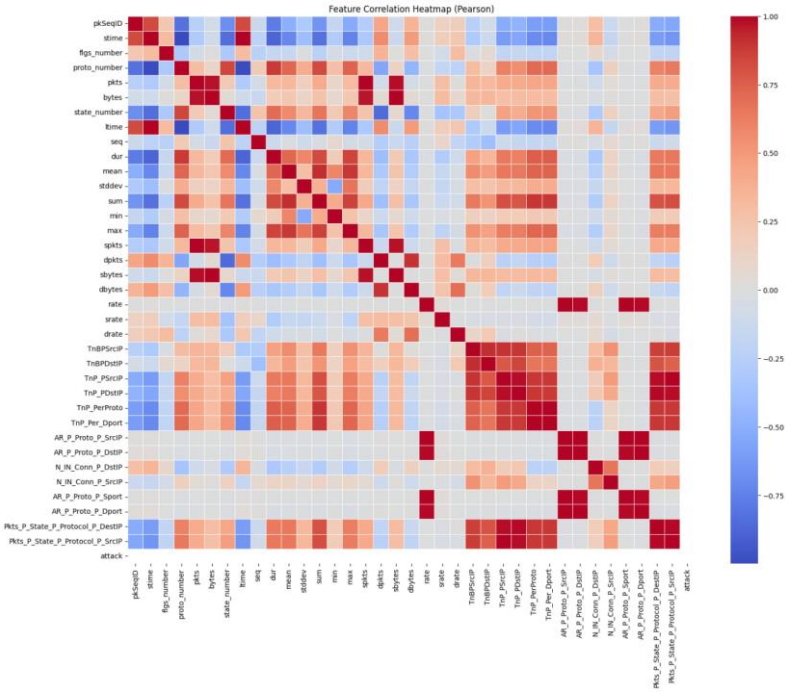


(شکل ۲-۲) دسته بندی subcategory

نتایج این نمودار نشان می‌دهد که subcategory ها شامل UDP، TCP و HHTP است که حملات با پروتکل UDP بیشترین فراوانی را دارند. همچنین subcategory حمله مبتنی بر HTTP سهم بسیار کوچکی از داده را شامل می‌شود. مدل های یادگیری ماشین در تشخیص این نوع حملات عملکرد ضعیف‌تری خواهند داشت.

Correlation analysis

در مرحله بعدی همبستگی بین ویژگی های عددی داده ها را با استفاده از ضریب همبستگی pearson محاسبه کزدیم. نتایج به صورت heatmap به صورت زیر است. (شکل ۳-۲)



(شکل ۳-۲) correlation heatmap –

فصل سوم

مدل های یادگیری

در این تمرین برای classification انواع حملات DDOS و DOS از سه مدل مبتنی بر درخت تصمیم استفاده شده است.

- Random Forest: مدلی مبتنی بر تجمیع چندین درخت تصمیم که با کاهش واریانس، عملکرد پایداری را در مسائل classification ارائه می‌دهد.
- XGBoost: یکی از قدرتمندترین الگوریتم های Gradian Boosting که با بهینه سازی تدریجی خطا، توانایی بالایی در یادگیری الگوهای پیچیده دارد.
- CatBoost: الگوریتم مبتنی بر Boosting که به ویژه در برخورد با ویژگی های دسته‌ای عملکرد مناسبی از خود نشان می‌دهد و نیاز کمتری به پیش‌پردازش داده ها دارد.

ارزیابی عملکرد مدل ها

نتایج به دست آمده نشان داد هر سه مدل عملکرد کاملا صحیح داشته‌اند و بررسی confusion matrix ها نشون میده مقدار خطا همه مدل صفر و یا فقط ۱ عدد بوده است. در ادامه مقادیر precision، recall و accuracy هر مدل را می‌بینیم. (شکل ۱-۳) (شکل ۲-۳) (شکل ۳-۳)

```
Random Forest Confusion Matrix:
 [[130052      0]
 [     0  69948]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    130052
           1       1.00      1.00      1.00     69948

    accuracy                           1.00    200000
   macro avg       1.00      1.00      1.00    200000
weighted avg       1.00      1.00      1.00    200000
```

(شکل ۱-۳) – مدل Random Forest

```
[[130051      1]
 [     0  69948]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    130052
           1       1.00      1.00      1.00     69948

    accuracy                           1.00    200000
   macro avg       1.00      1.00      1.00    200000
weighted avg       1.00      1.00      1.00    200000
```

(شکل ۳-۲) – مدل CatBoost

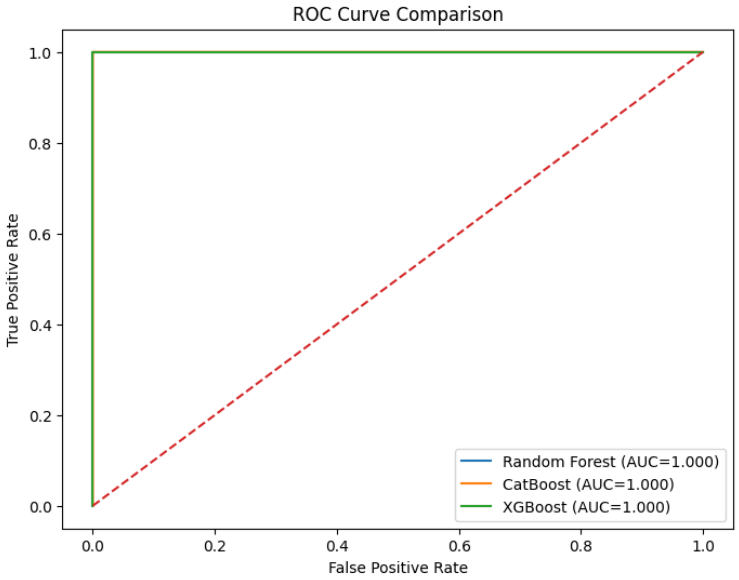
```
[[130052      0]
 [     0  69948]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    130052
           1       1.00      1.00      1.00     69948

    accuracy                           1.00    200000
   macro avg       1.00      1.00      1.00    200000
weighted avg       1.00      1.00      1.00    200000
```

(شکل ۳-۳) – مدل XGBoost

در ادامه نمودار ROC نیز رسم شد. (شکل ۳-۴)



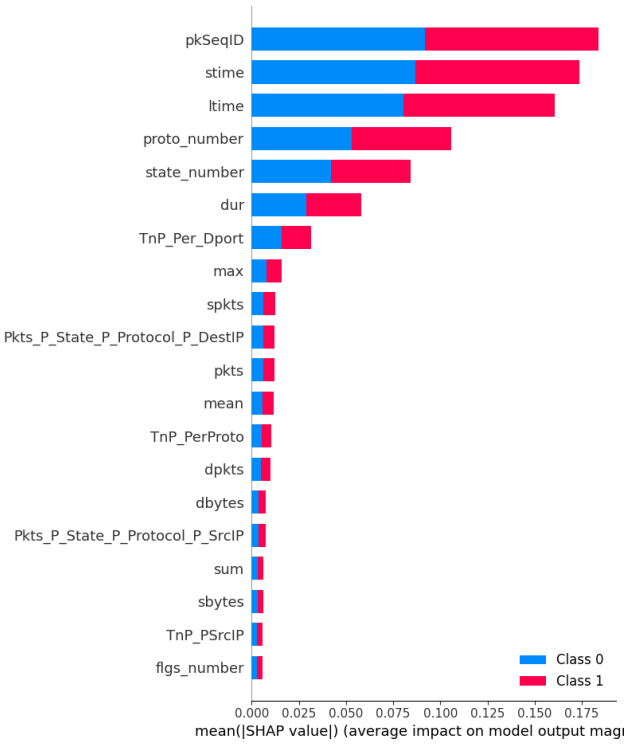
(شکل ۳-۴) نمودار ROC

اگرچه دقت ۱۰۰٪ کاملا درست به نظر می‌رسد، اما نتایج به‌دست‌آمده نیازمند تحلیل دقیق‌تری هستند. بررسی‌های انجام‌شده در بخش‌های قبلی نشان داد که مدل‌ها تمایل دارند به تعداد محدودی از ویژگی‌ها وابسته شوند. این مسئله می‌تواند نشان‌دهنده‌ی ساده بودن الگوی جداسازی کلاس‌ها در داده باشد. همچنین وجود ویژگی‌هایی باشد که به‌صورت مستقیم یا غیرمستقیم اطلاعات مربوط به لیبل را در خو دارند نیز موثر باشد.

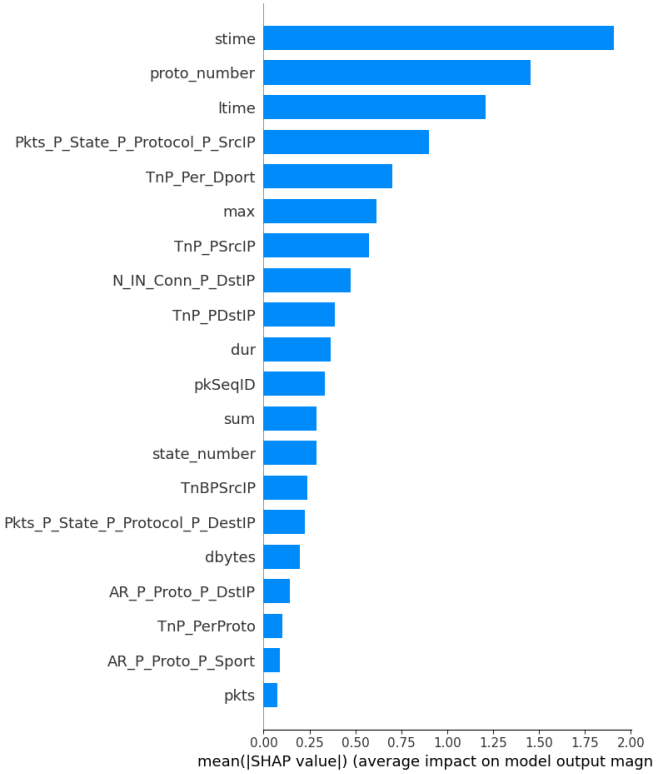
علاوه بر این، عدم توازن داده‌ها نیز می‌تواند بر نتایج ارزیابی تاثیرگذار باشد و باعث شود که برخی معیارها بیش از حد خوش‌بینانه گزارش شوند. بنابراین، نتایج این پروژه بیشتر از آن‌که نشان‌دهنده‌ی برتری یک مدل خاص باشند، بیانگر ویژگی‌های ذاتی مجموعه‌داده و ساختار حملات موجود در آن هستند.

Shap

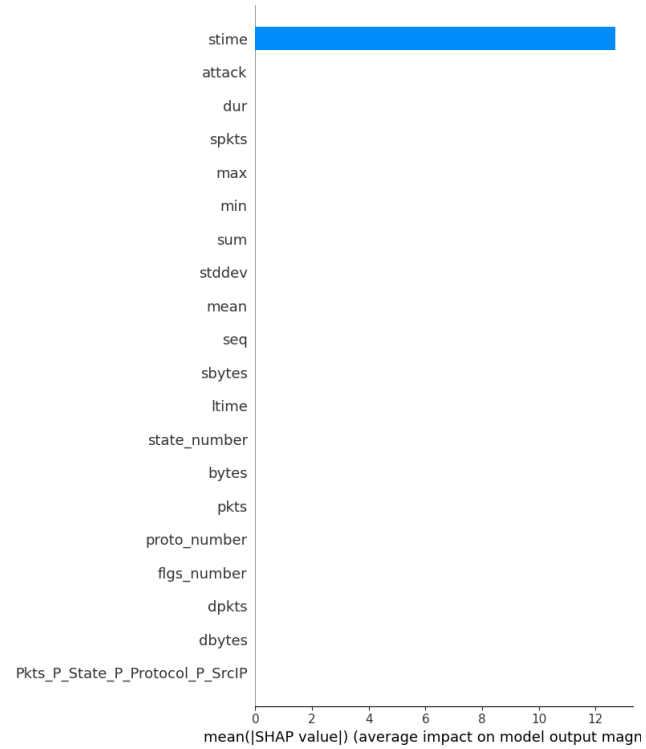
در مرحله بعدی برای هر سه مدل نمودار shap را رسم کردیم تا بررسی کنیم که هر یک از فیچر ها چقدر تاثیر دارند. (شکل ۵-۳) (شکل ۶-۳) (شکل ۷-۳)



(شکل ۵-۳) – نمودار shap مدل RF



(شکل ۶-۳) - نمودار shap مدل CatBoost

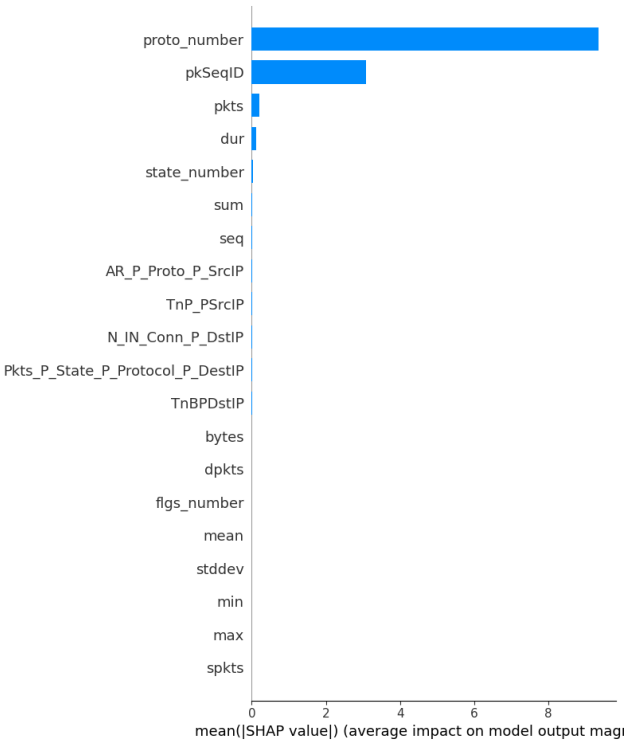


(شکل ۷-۳) - نمودار shap مدل XGBoost

نمودار shap مربوط به مدل XGBoost، نشون میده که این نمودار فقط بر اساس stime، نوع حمله را تشخیص می‌دهد و بقیه فیچر ها را نادیده می‌گیرد. این مثالی از data Leakage است.

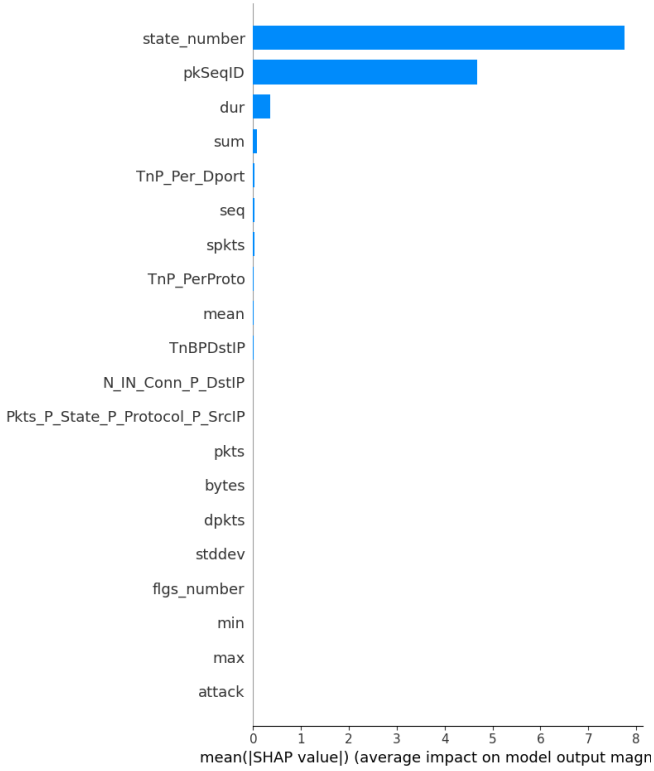
جلوگیری از Leakage

در بخش قبلی دیدیم که در مدل XGBoost با توجه به نمودار، data leakage اتفاق افتاده است و فقط بر اساس stime تصمیم گیری می‌کند. برای این که از این اتفاق جلوگیری کنیم، از داده ها ستون های stime و ltime را حذف کردیم و مدل ها را دوباره ترین کردیم. نتیجه accuracy، recall و precision همانند قبل بود و مقدار ۱۰۰ درصد بود. نمودار های shap دو مدل دیگر تغییر آنچنانی نداشتند اما نمودار مدل XGBoost به صورت تغییر یافت. (شکل ۸-۳)



(شکل ۸-۳) - نمودار shap مدل XGBoost بعد از حذف stime,ltime

مشاهده می‌کنیم که تا حدودی از data lekeage جلوگیری شده است. برای بهبود عملکرد مدل، در مرحله بعدی ستون proto_number را نیز از دیتا ست حذف کردیم و همچنان accuracy ۱۰۰٪ بود.(شکل ۹-۳)



(شکل ۹-۳) - نمودار shap مدل XGBoost بعد از حذف stime,ltime,proto_number

مشاهده می‌کنیم که این مدل این بار به فیچر های state_num و PakSeq_ID متکی هست و سایر فیچرها در یادگیری مدل تاثیری ندارند.

نتیجه گیری

در این پروژه، تحلیل داده‌های مربوط به حملات DoS و DDoS در شبکه‌های IOT با دیتاست مشخص انجام شد. در ابتدا، بررسی داده‌های نشان داد که داده‌ها از نظر کلاس‌های هدف و زیردسته‌های حمله دارای عدم توازن قابل‌توجهی هستند که این موضوع می‌تواند بر ارزیابی عملکرد مدل‌های یادگیری ماشین تأثیرگذار باشد.

در نهایت پس از آموزش مدل ها، با روش SHAP، اهمییت فیچر ها در تصمیم گیری مدل ها بررسی شد. نتایج نشان داد مدل ها، مخصوصا مدل XGBoost، تمایل دارند به فیچر های محدودی وابسته باشند و data Leaekage رخ داد. با حذف برخی ستون ها برای رفع این مشکل تلاش کردیم و هر بار ستون دیگری متکی شد. این نشان دهنده اسن هست که دیتاست موردنظر به طور ذاتی نوع حمله را encode کرده و حذف تدریجی ستون‌ها فقط باعث جابجایی منبع leakage می‌شود.

لینک گیت هاب

در ادامه لینک گیت هاب مربوط به کد های پروژه قرار داده شده است.

https://github.com/sinazb/DDOS-Analysis-BotIoT