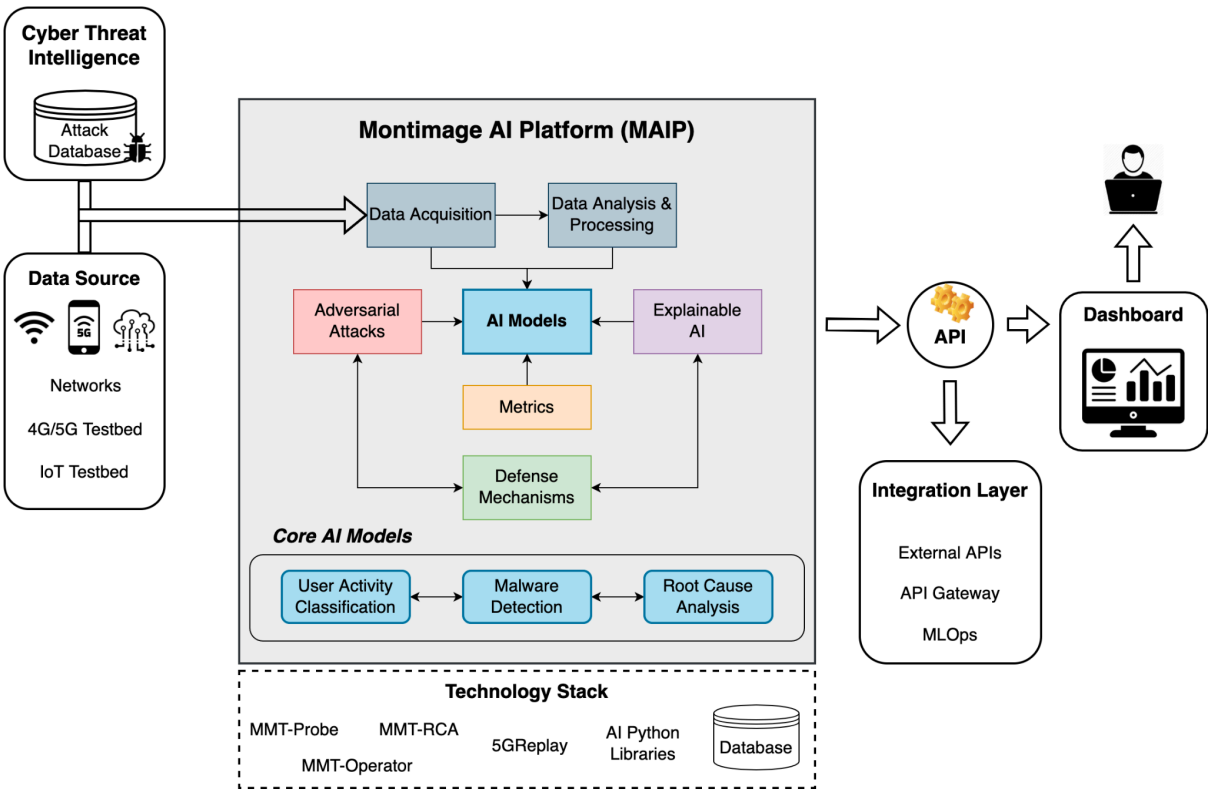
Architecture

Montimage AI Platform (MAIP) provides users with easy access to AI services developed by Montimage, through a friendly and intuitive interface for interacting with the APIs. It provides a range of ML services, including extract features, build or retrain the model, inject adversarial attacks, produce explanations and evaluate our model using different metrics. Each of these services is exposed through dedicated **APIs** that can be accessed through the server, making it easy to integrate with other applications and systems.



The above figure shows the architecture of our MAIP framework, that includes the following main components:

- **Data acquisition** module collects raw traffic data from networks or IoT testbed in either online or offline mode. It can also use Cyber Threat Intelligence (CTI) sources, e.g., deployed honeypots, to learn and continuously train our model using attack patterns and past malware information in the database.
- **Data analysis & processing** module employs our Montimage monitoring tool (MMT) to parse a wide range of network protocols (e.g., TCP, UDP, HTTP, and more than 700) and extract flow-based features. Then, the restructured and computed data is transformed into a numeric vector so that can be easily processed by our AI model.
- **AI models** module is responsible for creating and utilizing ML models able to classify the vectorized form of network traffic data for different purposes, such as user activity classification, malware detection in encrypted traffic or root cause analysis.
- **Adversarial attacks** module injects various evasion and poisoning adversarial attacks for robustness analysis of our system.

- **Explainable AI** module aims at producing post-hoc global and local explanations of predictions of our model.
- **Metrics** module allows to measure quantifiable metrics for its accountability and resilience.
- **Defense mechanisms** module provides countermeasures to prevent attacks against both AI and XAI models.

Design & Implementation

Implementation

Overall our framework is designed with a server written in ExpressJS, that employs the MMT tool written in C for feature extraction and leverages popular Python libraries for DL and XAI. The client is built in React and accessible via Swagger APIs, offering users an intuitive and user-friendly interface to interact with the DL services.

Current repository: https://gitlab.com/strongcourage/maip-app

Server side

We use Swagger APIs. So far **50+ APIs** have been implemented.

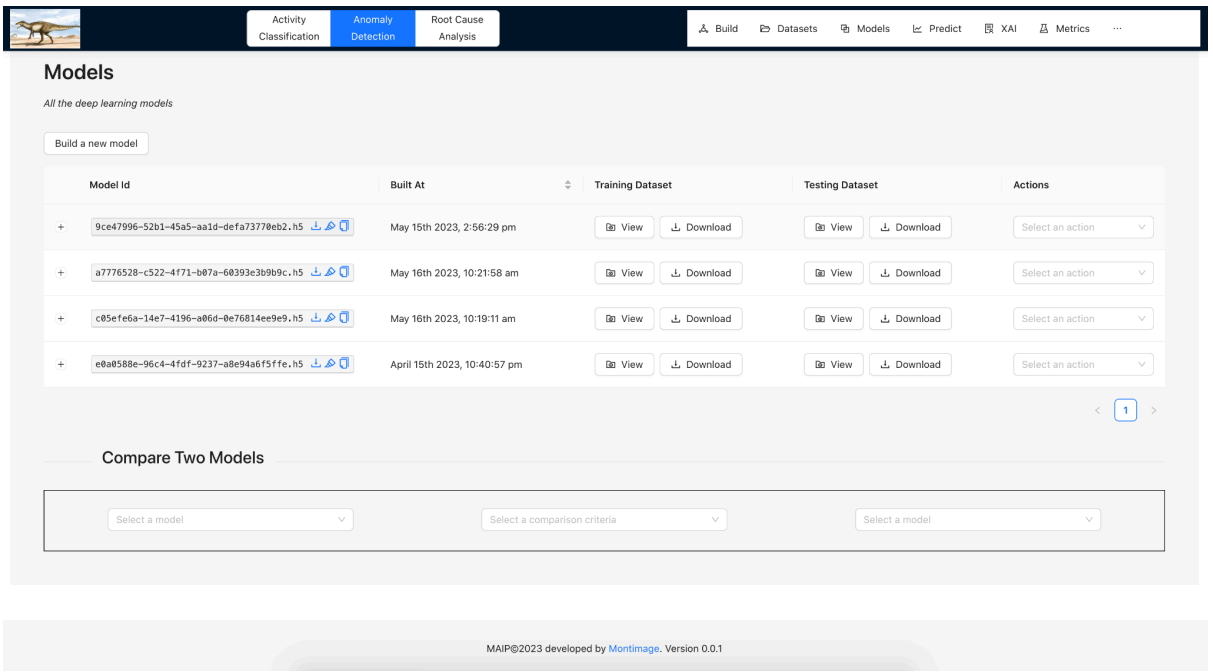
Client side

Overall

There are 3 modes corresponding to 3 existing AI-based applications:

- Anomaly Detection: the current implementation focuses on this mode
- Activity Classification: feasible, straight-forward (TODO)
- Root Cause Analysis: (TBD)

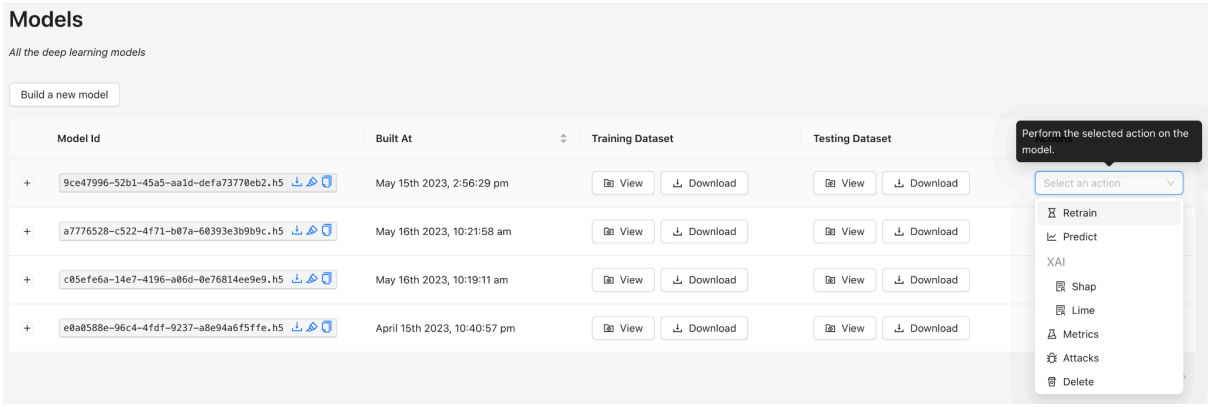
Ideally, users can select one or several application(s) they want to use. For instance, the combination of Anomaly Detection and RCA is useful to first detect malware attacks in IoT networks, then possibly to identify the root cause for quick remediation actions using MAIP.



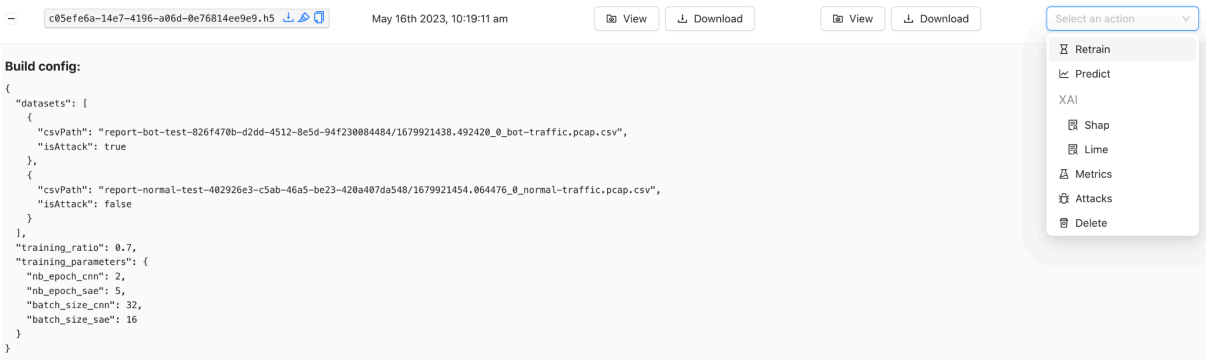
Models Page

This page provides an overview about all built models and allows users to compare performance of 2 selected models.

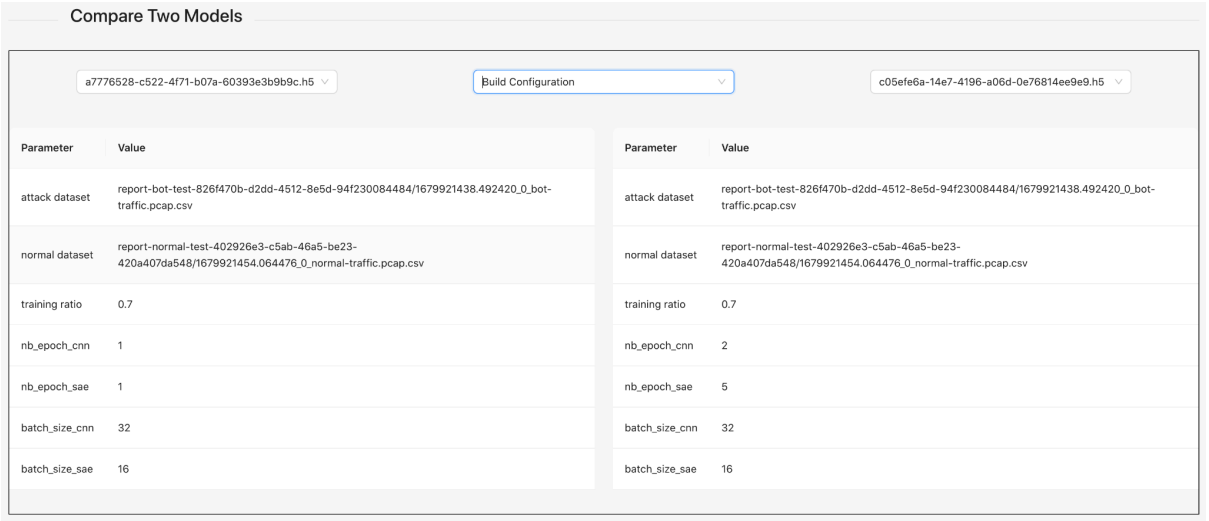
- The table shows a list of built DL models that can be sorted by the time we built them.



- For each model, users can download it, rename it or copy its name. Users can view or download the training/testing dataset. Users can see in detail the build configuration. Furthermore, users can perform some actions on the model.



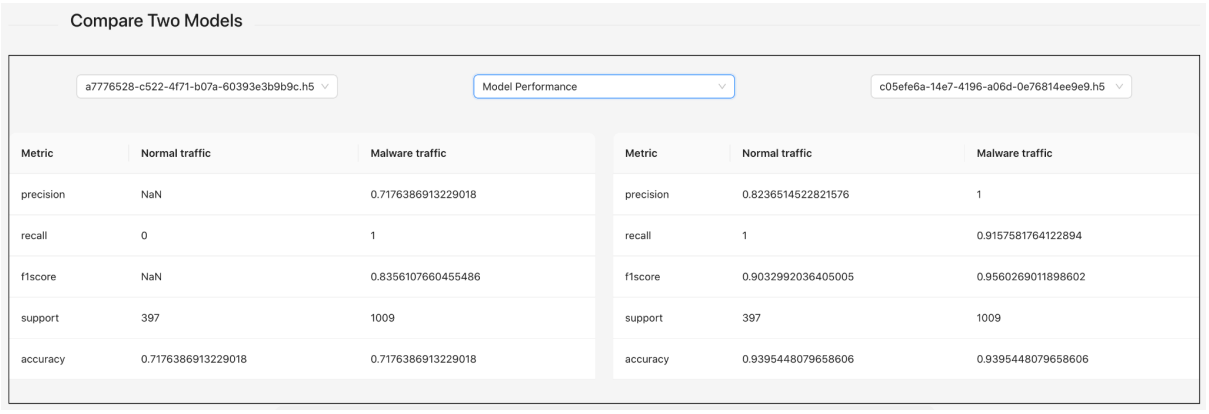
- Users can compare two DL models based on the build configuration, model performance and confusion matrix. For instance, 2 models below use the same build configuration, except 2 training parameters representing number of epochs for CNN and SAE.



Parameter	Value
attack dataset	report-bot-test-826f470b-d2dd-4512-8e5d-94f230084484/1679921438.492420_0_bot-traffic.pcap.csv
normal dataset	report-normal-test-402926e3-c5ab-46a5-be23-420a407da548/1679921454.064476_0_normal-traffic.pcap.csv
training ratio	0.7
nb_epoch_cnn	1
nb_epoch_sae	1
batch_size_cnn	32
batch_size_sae	16

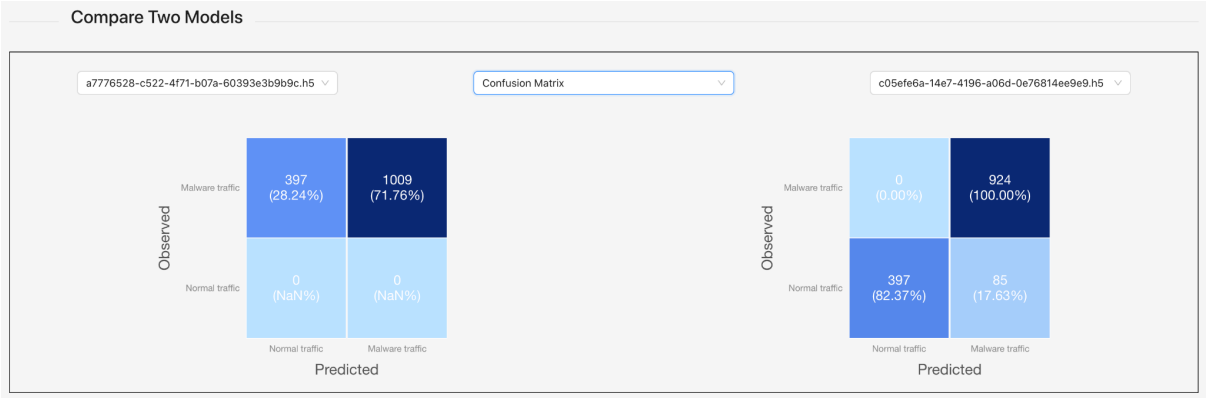
Parameter	Value
attack dataset	report-bot-test-826f470b-d2dd-4512-8e5d-94f230084484/1679921438.492420_0_bot-traffic.pcap.csv
normal dataset	report-normal-test-402926e3-c5ab-46a5-be23-420a407da548/1679921454.064476_0_normal-traffic.pcap.csv
training ratio	0.7
nb_epoch_cnn	2
nb_epoch_sae	5
batch_size_cnn	32
batch_size_sae	16

- The results show that the right model with accuracy 0.94 has a better performance than the left one with accuracy 0.72.



Metric	Normal traffic	Malware traffic
precision	NaN	0.7176386913229018
recall	0	1
f1score	NaN	0.8356107660455486
support	397	1009
accuracy	0.7176386913229018	0.7176386913229018

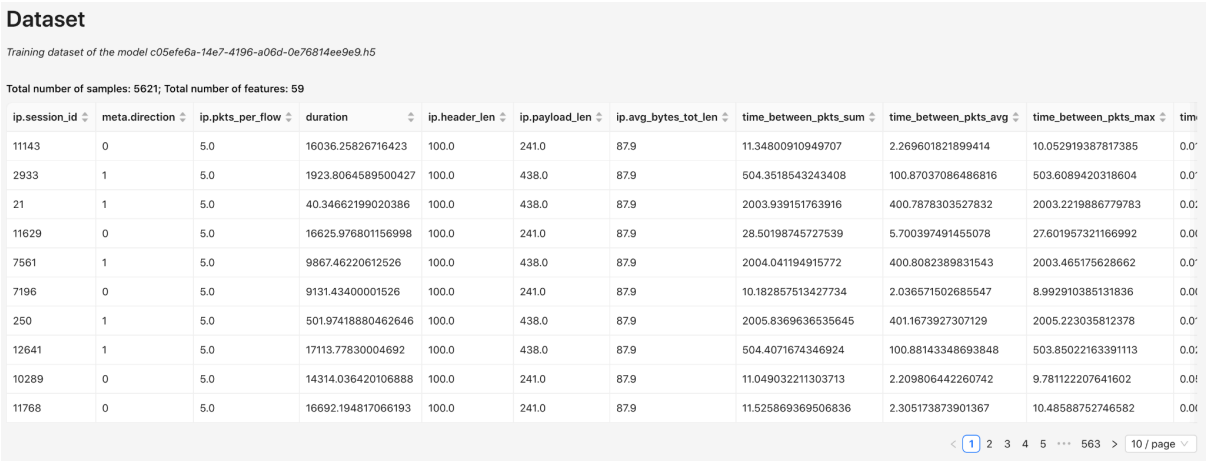
Metric	Normal traffic	Malware traffic
precision	0.8236514522821576	1
recall	1	0.9157581764122894
f1score	0.9032992036405005	0.9560269011898602
support	397	1009
accuracy	0.9395448079658606	0.9395448079658606



Dataset Page

This page provides insights of the training/testing dataset using different tables and plots.

- A quick glance of the dataset.



Dataset

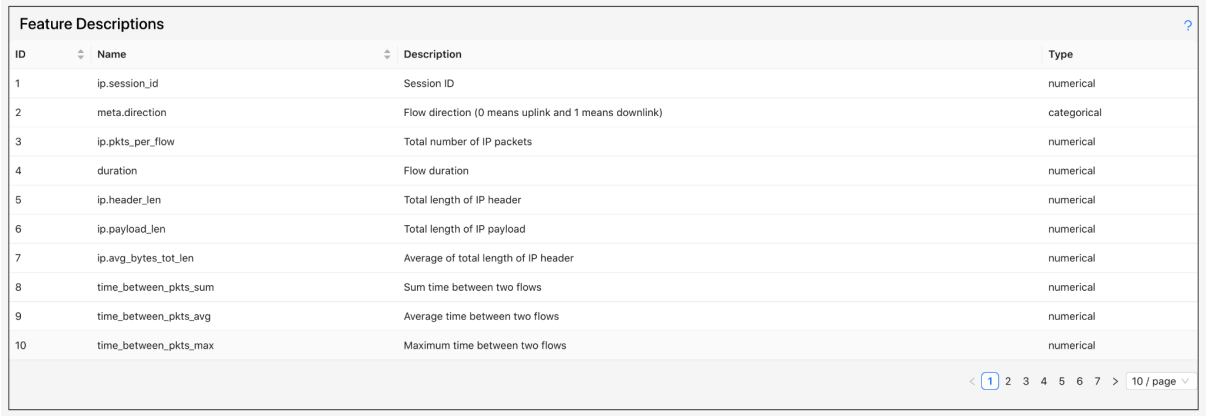
Training dataset of the model c05efe6a-14e7-4196-a06d-0e76814ee9e9.h5

Total number of samples: 5621; Total number of features: 59

ip.session_id	meta.direction	ip.pkts_per_flow	duration	ip.header_len	ip.payload_len	ip.avg_bytes_tot_len	time_between_pkts_sum	time_between_pkts_avg	time_between_pkts_max	tim
11143	0	5.0	16036.25826716423	100.0	241.0	87.9	11.34800910949707	2.269601821899414	10.052919387817385	0.0
2933	1	5.0	1923.8064589500427	100.0	438.0	87.9	504.3518543243408	100.87037086486816	503.6089420318604	0.0
21	1	5.0	40.34662199020386	100.0	438.0	87.9	2003.939151763916	400.7878303527832	2003.2219886779783	0.0
11629	0	5.0	16625.976801156998	100.0	241.0	87.9	28.50198745727539	5.700397491455078	27.601957321166992	0.0
7561	1	5.0	9867.46220612526	100.0	438.0	87.9	2004.041194915772	400.8082389831543	2003.465175628662	0.0
7196	0	5.0	9131.43400001526	100.0	241.0	87.9	10.182857513427734	2.036571502685547	8.992910385131836	0.0
250	1	5.0	501.97418880462646	100.0	438.0	87.9	2005.8369636535645	401.1673927307129	2005.223035812378	0.0
12641	1	5.0	17113.77830004692	100.0	438.0	87.9	504.4071674346924	100.88143348693848	503.85022163391113	0.0
10289	0	5.0	14314.036420106888	100.0	241.0	87.9	11.049032211303713	2.209806442260742	9.781122207641602	0.0
11768	0	5.0	16692.194817066193	100.0	241.0	87.9	11.525869369506836	2.305173873901367	10.48588752746582	0.0

< 1 2 3 4 5 ··· 563 > 10 / page

- This table shows detailed descriptions of features.

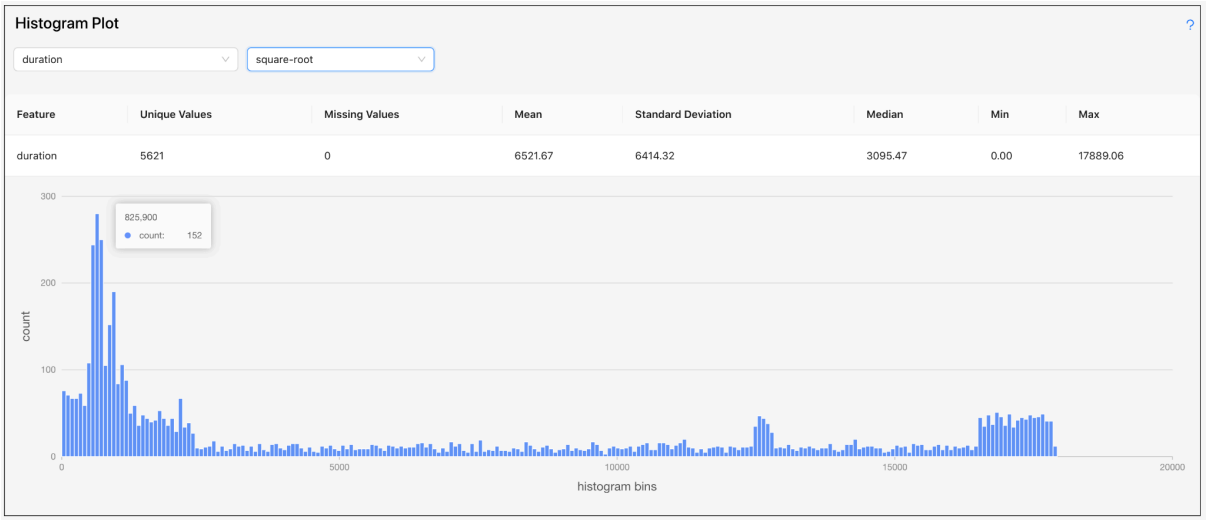


Feature Descriptions

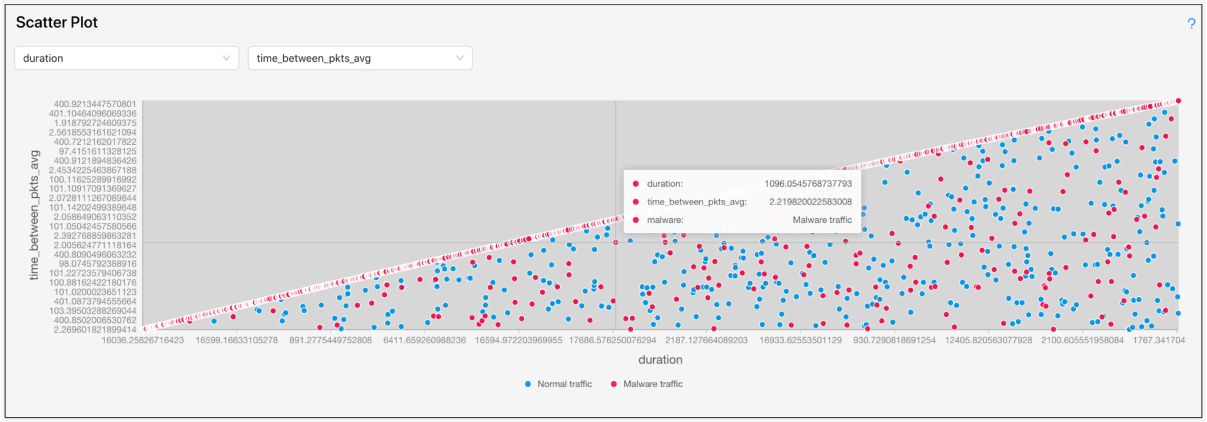
ID	Name	Description	Type
1	ip.session_id	Session ID	numerical
2	meta.direction	Flow direction (0 means uplink and 1 means downlink)	categorical
3	ip.pkts_per_flow	Total number of IP packets	numerical
4	duration	Flow duration	numerical
5	ip.header_len	Total length of IP header	numerical
6	ip.payload_len	Total length of IP payload	numerical
7	ip.avg_bytes_tot_len	Average of total length of IP header	numerical
8	time_between_pkts_sum	Sum time between two flows	numerical
9	time_between_pkts_avg	Average time between two flows	numerical
10	time_between_pkts_max	Maximum time between two flows	numerical

< 1 2 3 4 5 6 7 > 10 / page

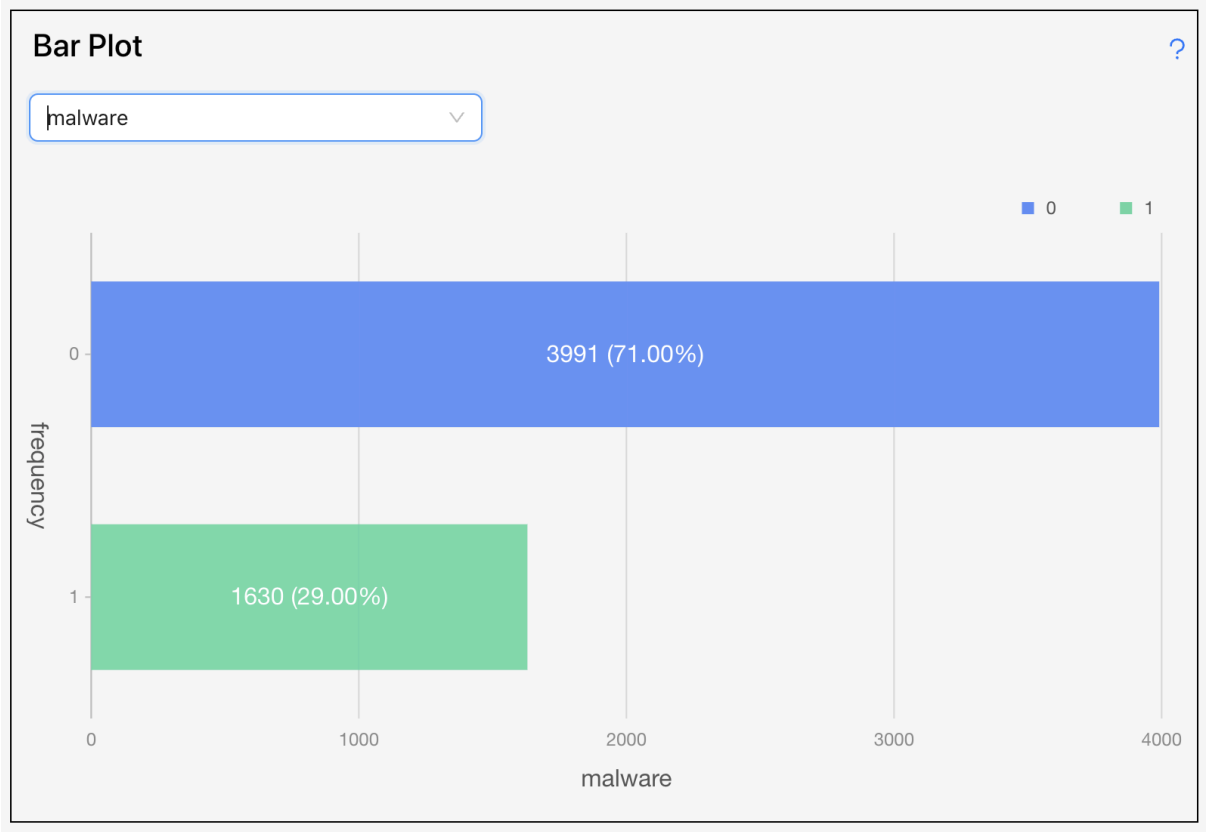
- A table contains different statistics of the feature, such as the number of unique values, number of missing values, mean, standard deviation, median, minimum, and maximum value. A histogram plot for each feature of the database shows the distribution of values in that feature.



- The scatter plot represents the relationship between two features of a dataset, each data point as a circle on a two-dimensional coordinate system. The color of each circle represents whether the traffic was Malware or Normal. Malware traffic is denoted with the color blue, while Normal traffic is denoted with the color red.



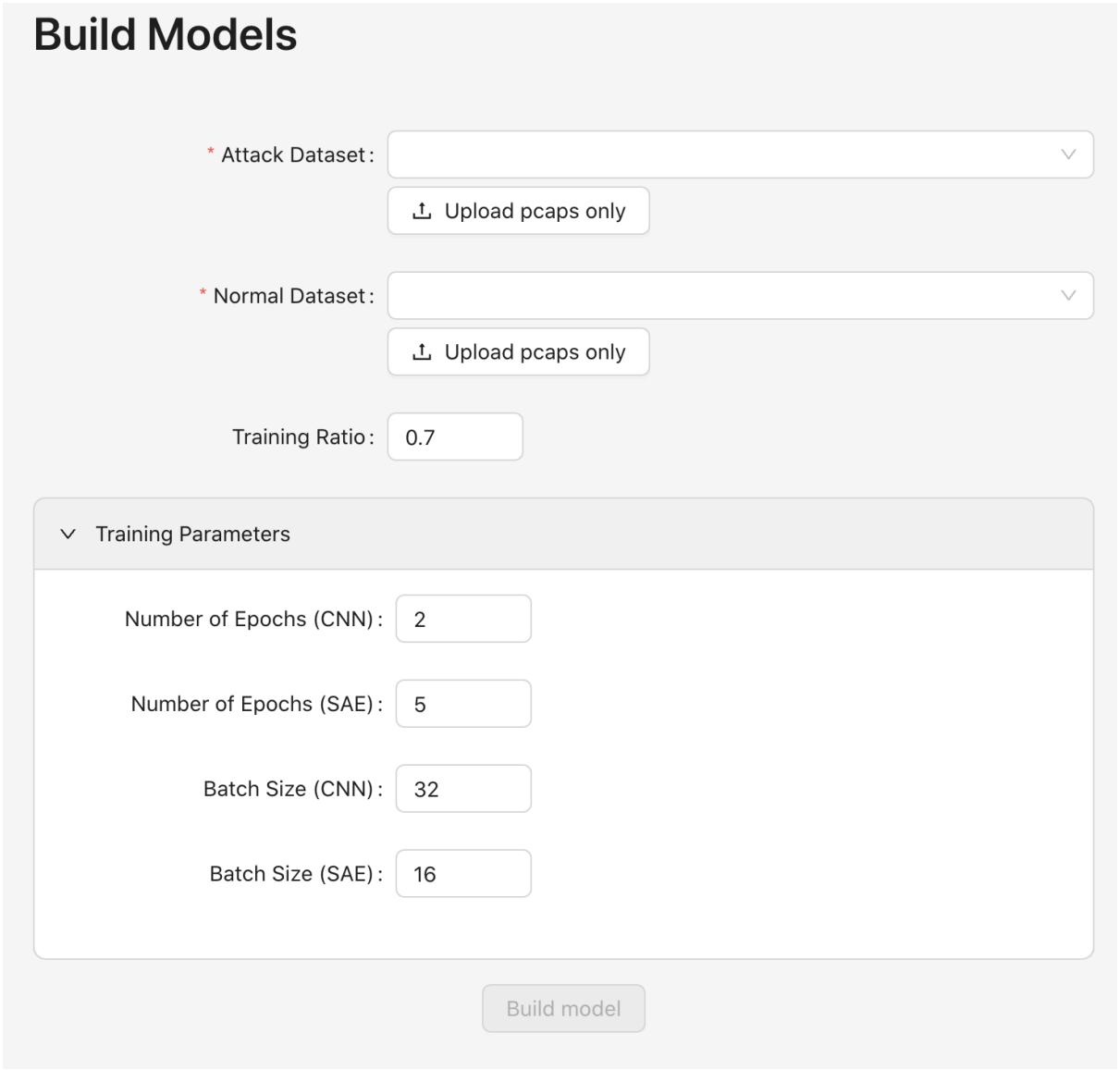
- The bar plot displays the frequency or proportion of a categorical feature.



Build Page

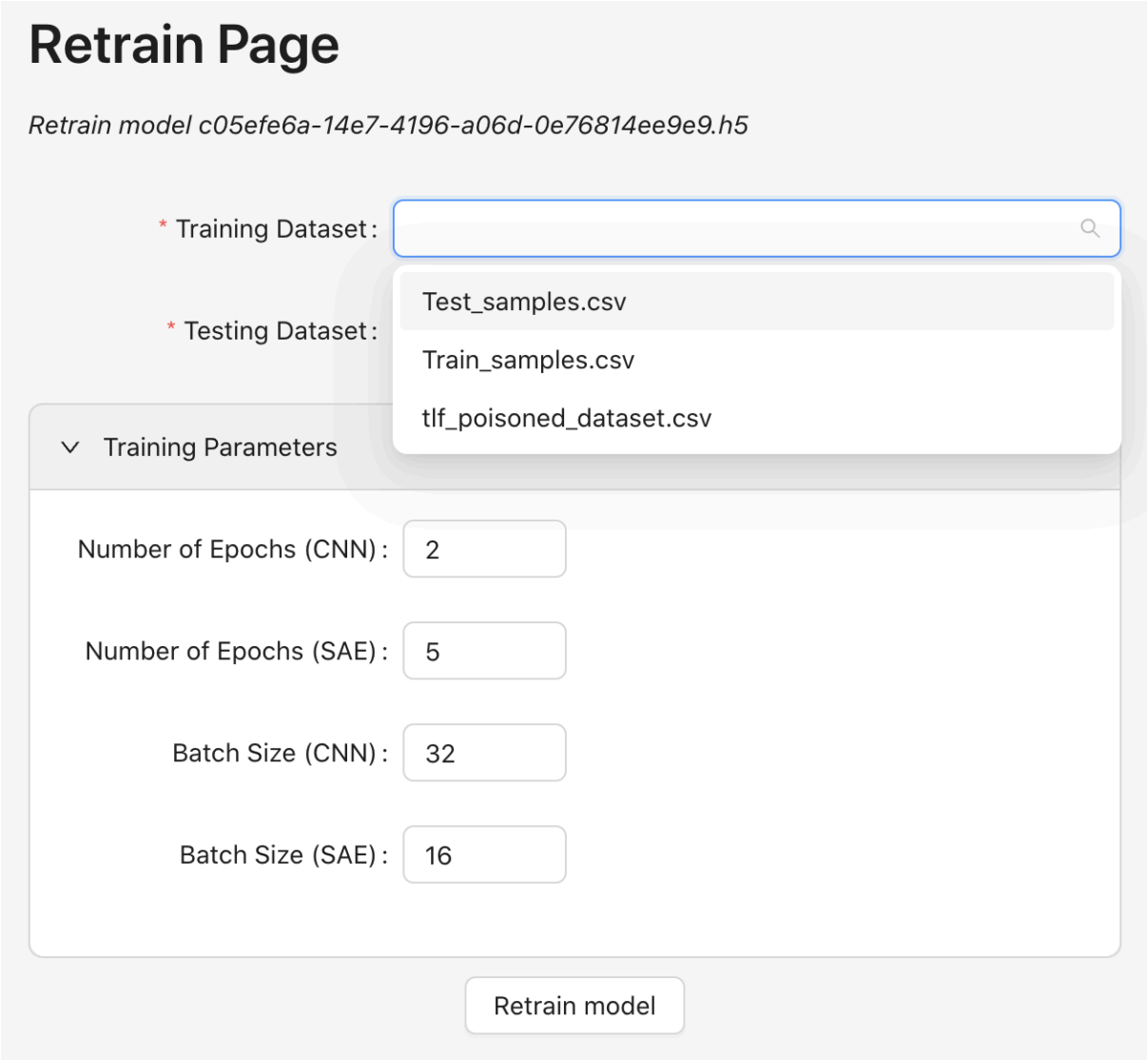
This page allows users to build a new DL model. For training/testing datasets, there are 2 options:

- Users can select existing analyzed reports generated by MMT monitoring tool
- Users can upload new pcap files (TODO)



Retrain Page

This page allows users to retrain a model by using different training/testing datasets or training parameters to find the optimal hyper parameters or measure impact of adversarial poisoning attacks.

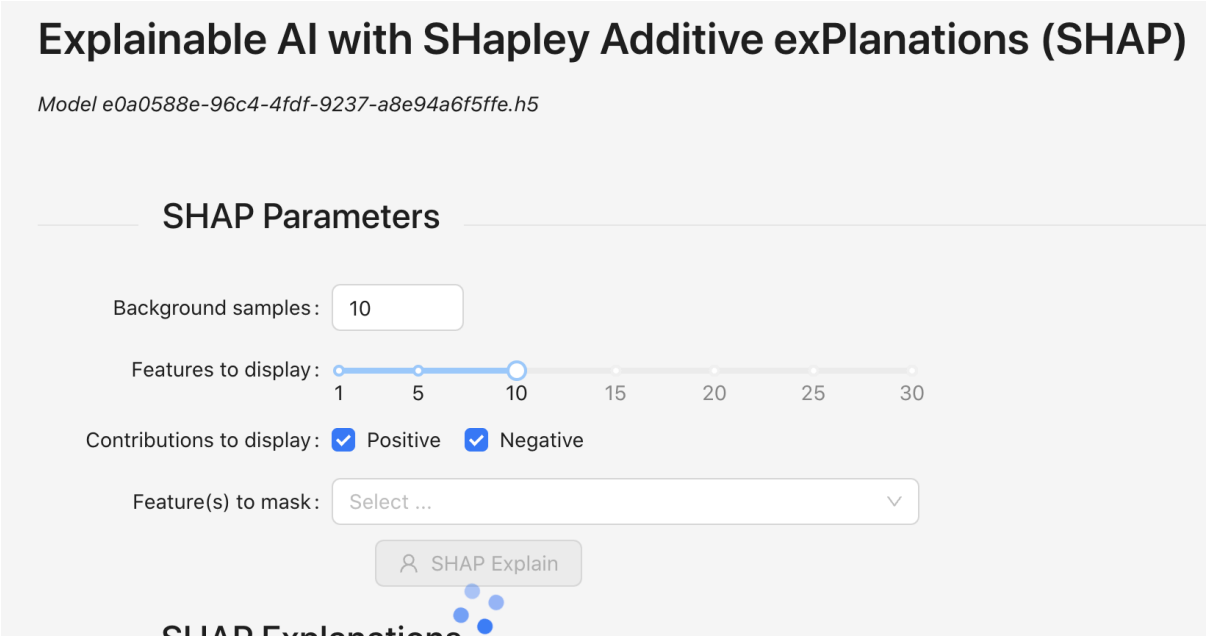


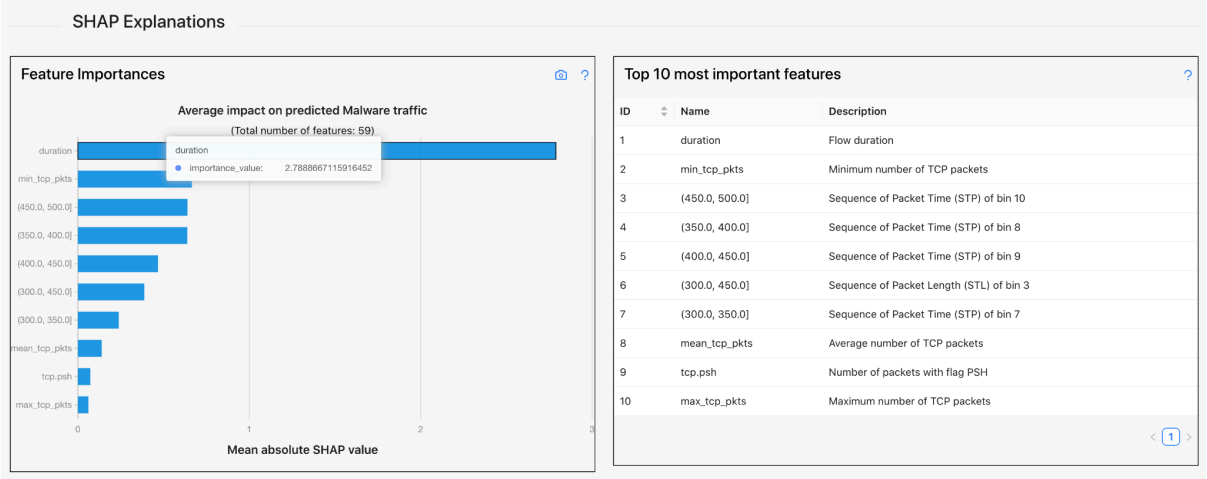
Predict Page (TODO)

This page allows users to predict a network traffic is benign or malware in both online and offline mode using existing models.

XAI SHAP Page

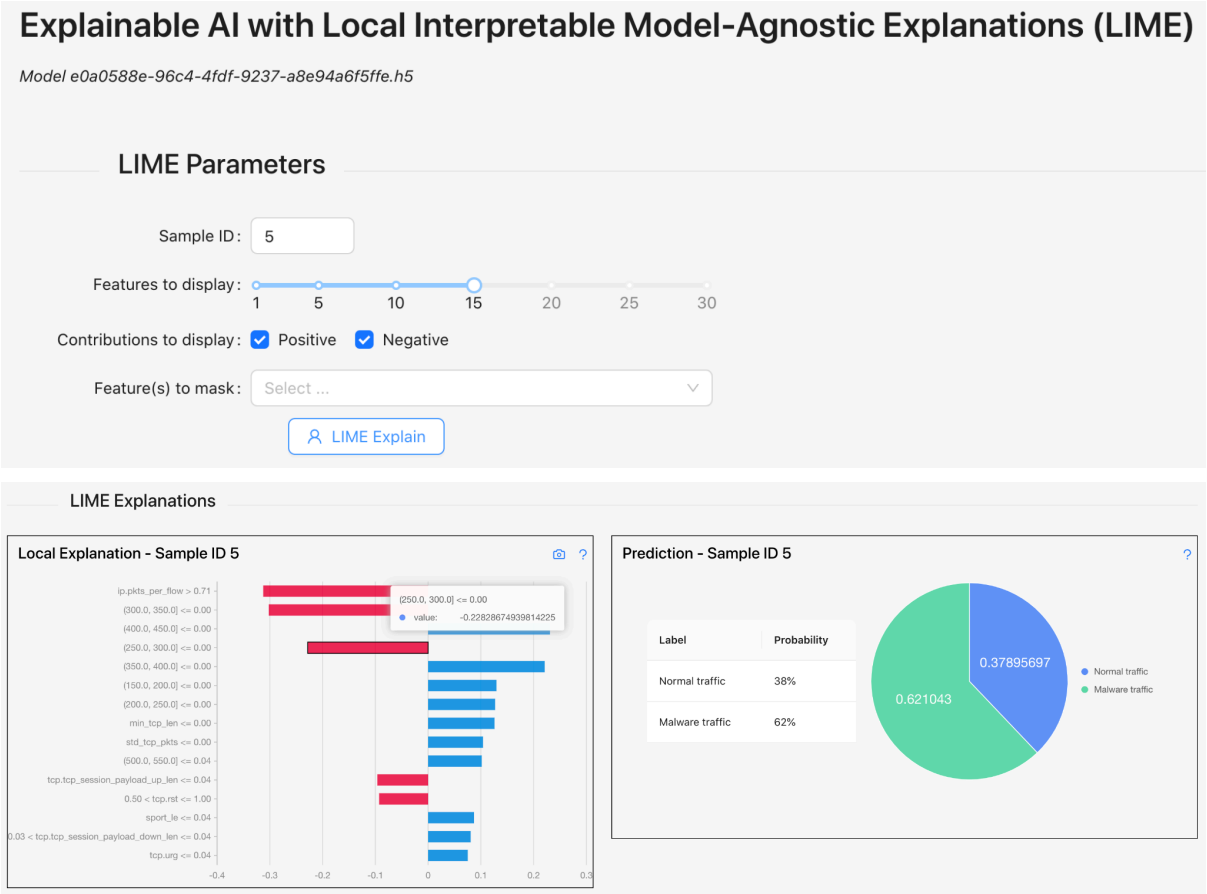
This page allows users to obtain SHAP explanations via SHAP feature importances plot which displays the sum of individual contributions, computed on the complete dataset.





XAI LIME Page

This page allows users to obtain LIME local explanations via local interpretability plot which displays each most important feature's contributions for this specific sample. The bar plot shows predicted probability for the selected sample.

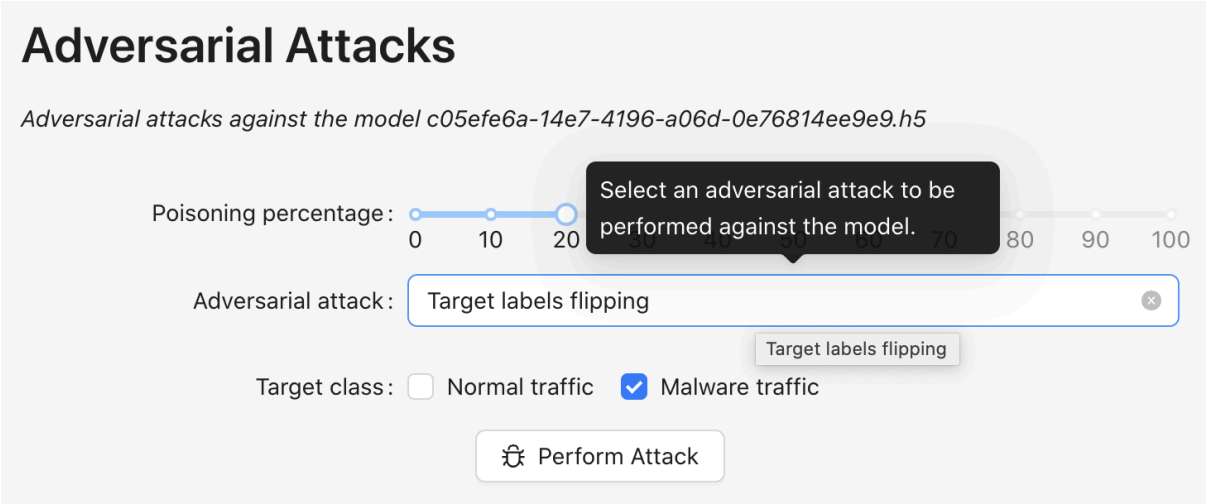


Attacks Page

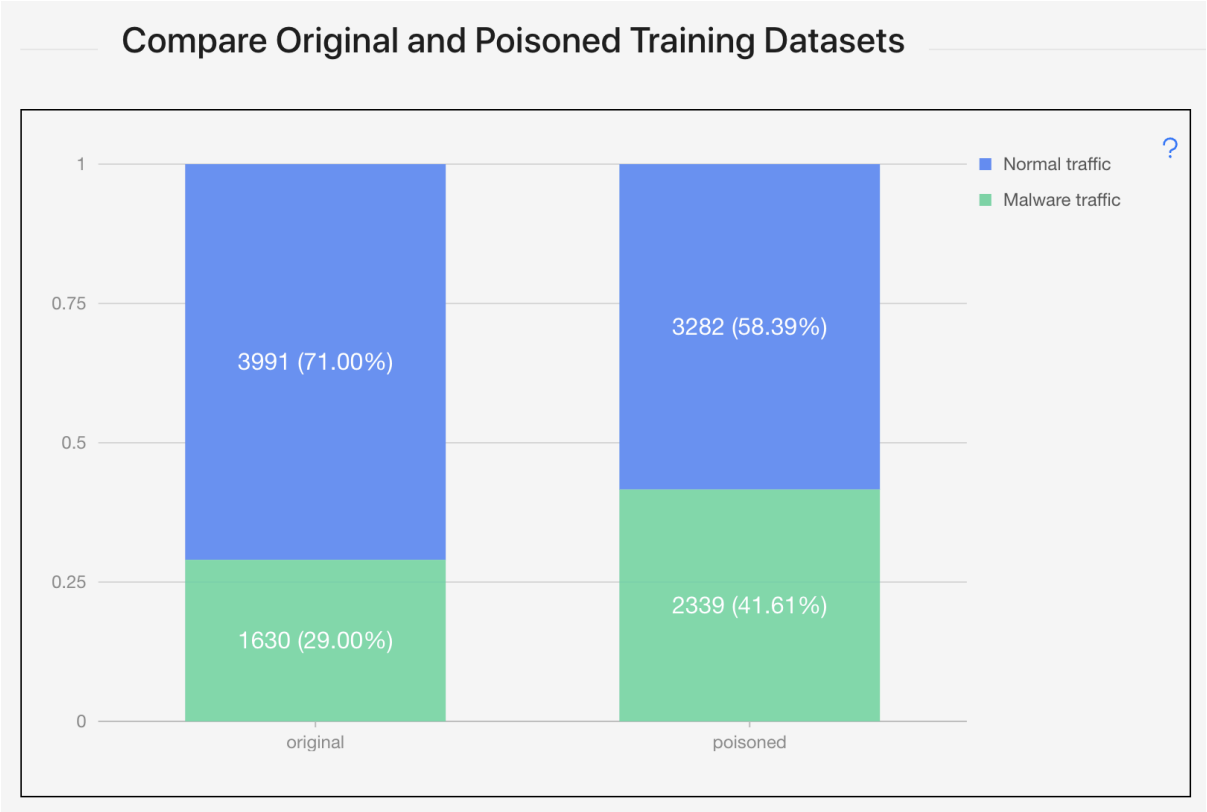
This page performs some adversarial attacks against the model. Currently we support 3 poisoning attacks:

- GAN-based poisoning attack
- Random swapping labels attack
- Target labels flipping attack

Here we perform the target labels flipping attack with the poisoning rate 20% and the target label “Malware traffic”.



- Clearly, the number of “Malware traffic” labels of the poisoning training dataset must be bigger than one of the original dataset.

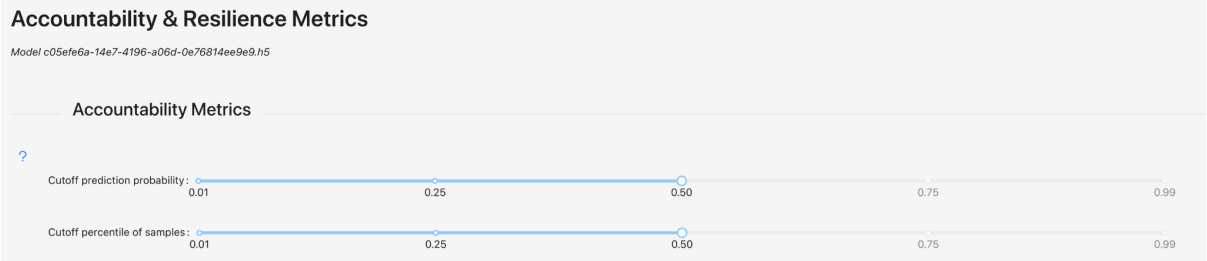


Metrics Page

This page provides different accountability and resilience metrics of a model (some metrics are proposed in WP2 of the SPATIAL project).

- Cutoff prediction probability is a fixed probability value above which the model will classify a sample as positive. For example, if the cutoff prediction probability is set to 0.5, the model will classify any sample with a predicted probability of belonging to the positive class greater than 0.5 as positive.

- Cutoff percentile is defined as the point on the predicted probability distribution above which the model will classify a sample as positive. For example, if the cutoff percentile is set to 90%, the model will classify any sample with a predicted probability of belonging to the positive class greater than the 90th percentile as positive.

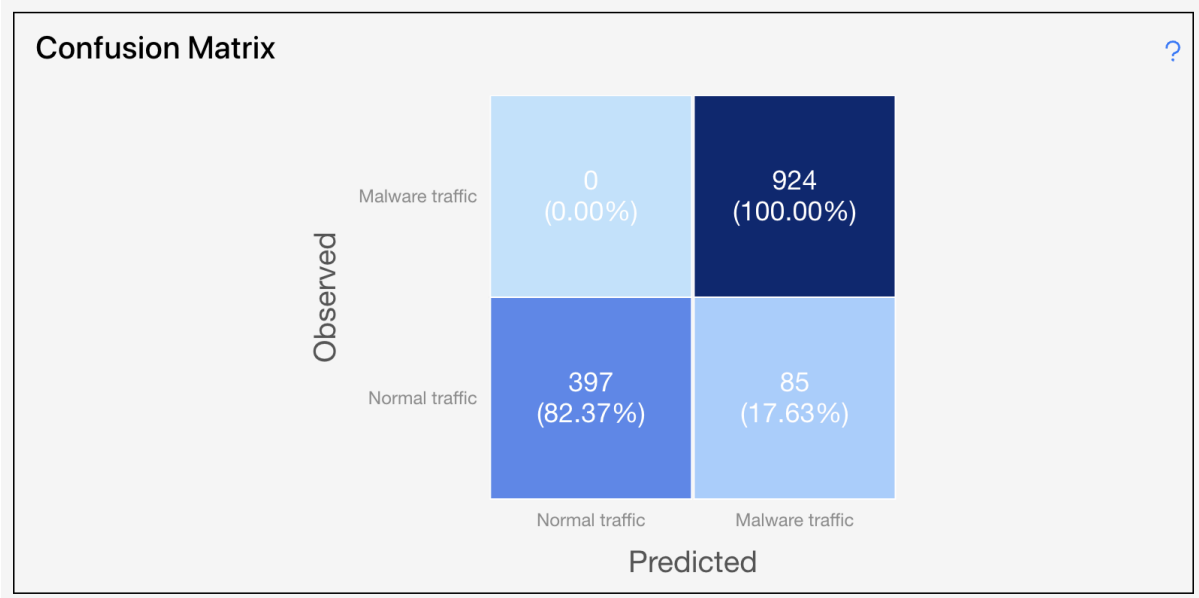


- This table shows a list of various model performance metrics for each class.

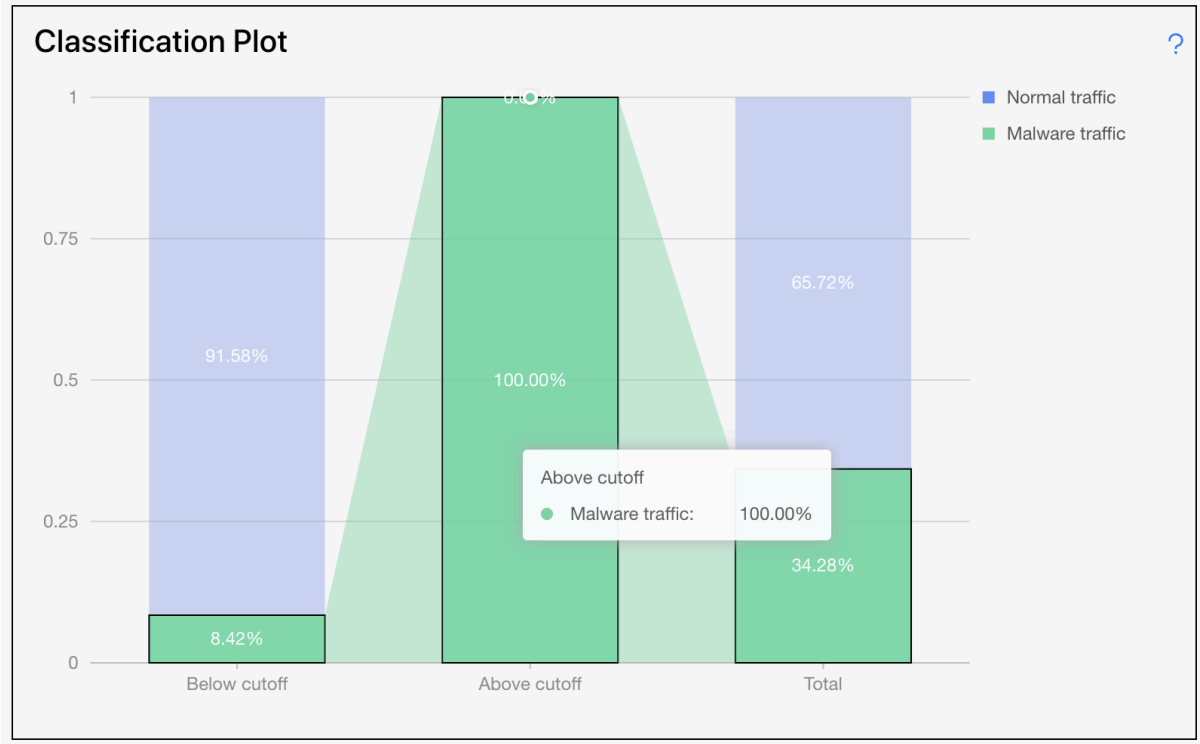
Model Performance

Metric	Normal traffic	Malware traffic
precision	0.8236514522821576	1
recall	1	0.9157581764122894
f1score	0.9032992036405005	0.9560269011898602
support	397	1009
accuracy	0.9395448079658606	0.9395448079658606

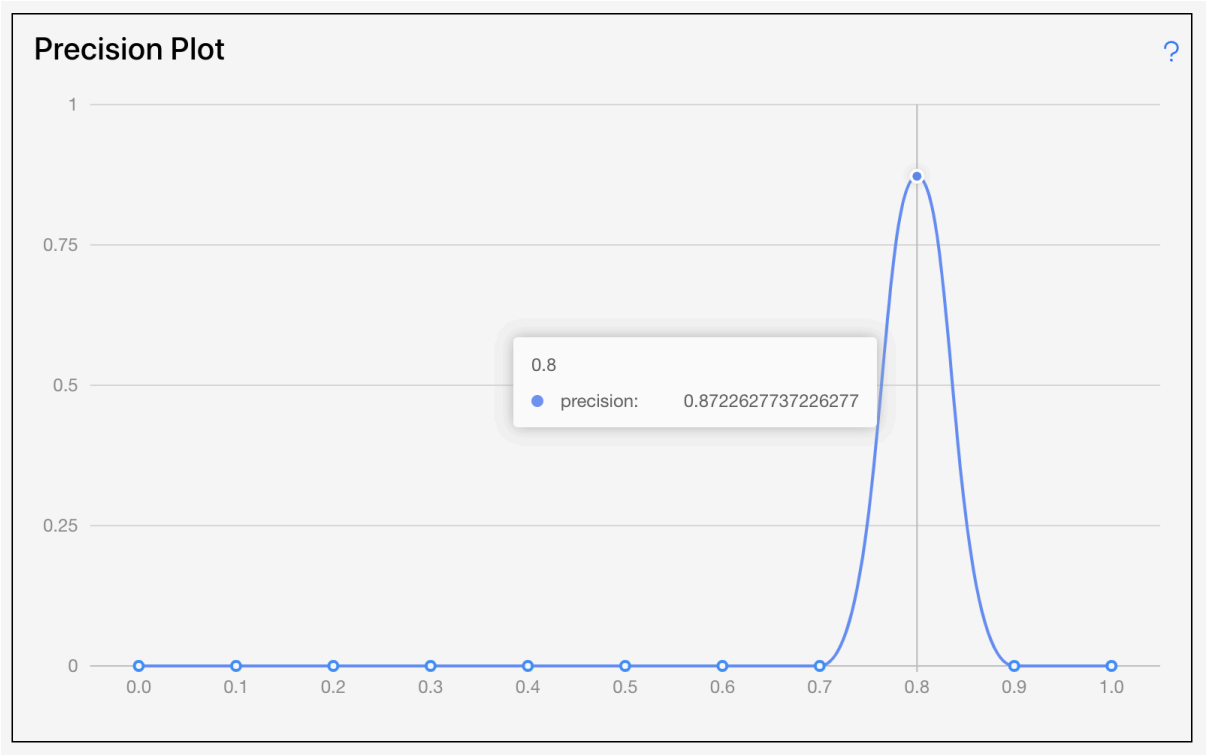
- The confusion matrix shows the number of True Negatives (predicted negative, observed negative), True Positives (predicted positive, observed positive), False Negatives (predicted negative, but observed positive) and False Positives (predicted positive, but observed negative). For different cutoff values, you will get a different number of False Positives and False Negatives. This plot allows you to find the optimal cutoff.



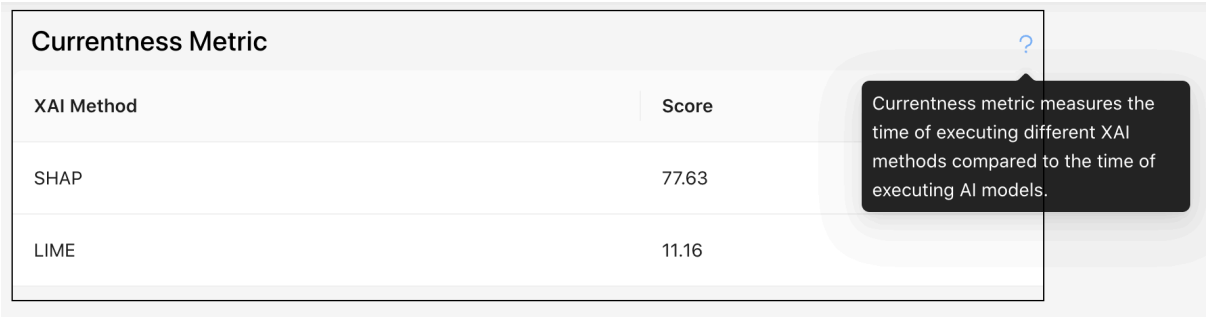
- This classification plot shows the fraction of each class above and below the cutoff.



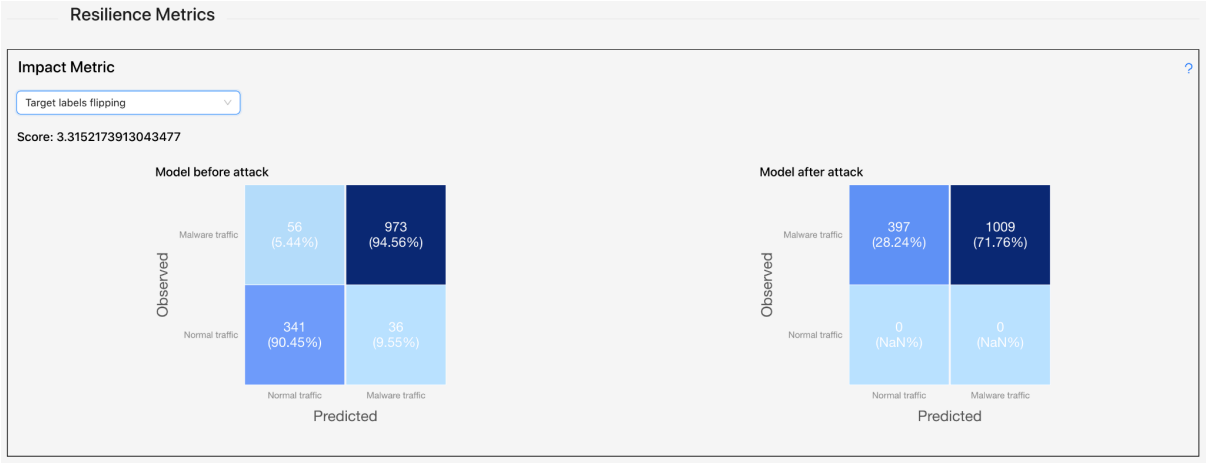
- The precision plot shows the precision values binned by equal prediction probabilities. It provides an overview of how precision changes as the prediction probability increases.



- Currentness metric measures the time of executing different XAI methods compared to the time of executing AI models. Obviously, SHAP's currentness score is much bigger than LIME's score as SHAP process is usually very slow.



- Impact metric shows difference between the original accuracy of a benign model compared to the accuracy of the compromised model after a successful poisoning attack.



Defense Page (TODO)

This page aims to apply some defense mechanisms to prevent attacks against both AI and XAI models and improve robustness of our models.

AI Reports Page (TODO)

This page aims to provide an AI report (in html, pdf, etc) to users using a predefined template. It should contain all information, insights, tables, plots, explanations and metrics that MAIP can produce for a specific model.