SmartAttack: Open-source Attack Models for Enabling Security Research in Smart Homes

Keyang Yu, Dong Chen
School of Computing and Information Sciences
Florida International University

***Abstract*—The Internet of Things (IoT) has been erupting the world widely over the decade. Smart home and smart building owners are increasingly deploying IoT devices to monitor and control their environments due to the rapid decline in the price of IoT devices. The recent intensive research has shown that network traffic traces of IoT devices have significant cybersecurity and privacy issues. These security and privacy defending techniques have enabled sophisticated approaches to ensure security and preserve user privacy. However, due to the fact that different approaches are evaluated using their own datasets, their own developed security and privacy attack models, and their own evaluating metrics, it is being significantly difficult to make a fair and comprehensive comparisons among different IoT security strengthening and user privacy preserving research to better understand IoT security issues and end-user benefits. To address this problem, we present a deep learning-based adversarial attack model framework—*SmartAttack*, which enables a set of sophisticated adversarial attack models that can be leveraged by researchers and industrial users from IoT security community to better evaluate their work. In essence, we leverage the most widely used unsupervised machine learning and deep learning models to design and implement these attack models. *SmartAttack* also provides user options to select the detailed configuration for each attack model, such as kernel, dataset splitting, cross-validation states, and evaluating metrics. We also evaluate the performance of *SmartAttack* using two different datasets. In addition, we made the source codes and the related datasets of *SmartAttack* publicly-available on our research website such that researchers can use our SmartAttack to benchmark their security strengthening and privacy-preserving approaches.**

***Index Terms*—Deep Learning, IoT security, Adversarial Machine Learning, Attack Models, User Privacy.**

I. Introduction

The Internet of Things (IoT) has been widely deployed in smart homes especially in recent decades. The total installed base of IoT connected devices is projected to amount to 75.44 billion worldwide by 2025, a fivefold increase in ten years [1]. Many IoT device manufacturers have matured IoT ecosystem provided for smart homes or facilities for data collecting, appliance controlling, environmental data monitoring, etc. Applying IoT devices brings efficiency to many areas including smart home and industrial, yet the grooming IoT ecosystem shows vulnerability to cyberattacks.

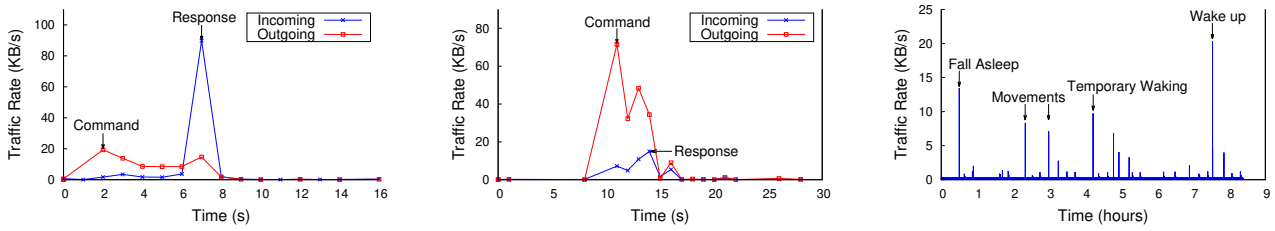
For instance, "Some popular home security cameras could allow would-be burglars to work out when you've left the building, according to a study published Monda. Researchers found they could tell if someone was in, and even what they were doing in the home, just by looking at data uploaded by the camera and without monitoring the video footage itself," reported on CNN Business [2]. In addition, Internet Service Providers (ISPs), such as Comcast, Time Warner, Sprint and Verizon, are collecting users' network traffic data and sharing them with third-parties for multiple purposes like generating monthly bills, detecting service outage, or providing customized promotions. In addition, ISPs are selling personal data like web browsing history without user's consent [3]. "Any service that provides Internet access can obviously see what resources users are accessing. And even with encryption, traffic patterns provide some information about activity." [4].

Although most IoT devices use encrypted network transmission and can apply these traffic reshaping approaches, there still exists side-channel information leakage to the on-path external adversaries, since user in-home activities highly correlates with simple time-series data statistical metrics, such as mean, variance, and range. Thus, the traffic data generated by IoT devices has significant privacy threats. Figure 1 shows the network traffic patterns of three widely deployed IoT devices. Prior work presented various methods that can be leveraged to extract a user's sensitive privacy information from IoT network traffic traces. For instance, applying data mining on smart meters recorded energy consumption data [5], using MAC addresses to distinguish IoT devices [6], or attacking on a specific communication protocol of IoT devices [7], etc. These attacks have been proved effective under different levels of hardware and software resource limitation.

Therefore, significant recent research [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20] are presented to mitigate or address these user security and privacy issues. Unfortunately, many prior approaches often uses different adversarial attack models, different datasets, and different evaluation metrics to benchmark their proposed techniques. And this has made it significantly difficult for researchers and end users to benchmark and comprehensively understand the benefits and limitations of different security and privacy defending approaches.

To address this problem, we design a new open-sourced adversarial attack model framework—*SmartAttack*, which enables a set of general sophisticated adversarial attack models that can be leveraged by researchers and industrial users of IoT security community to benchmark and evaluate their

978-1-6654-1552-1/20/$31.00 ©2020 IEEE



(a) Amazon Echo (b) Google Home (c) Withings Sleep Sensor

Fig. 1. IoT device traffic volume traces exposed to the on-path external adversaries.

work. Our hypothesis is an open-sourced adversarial attack toolkit, which is built using the most recent Machine Learning (ML) and Deep Learning (DL) models, can enable a fair and comprehensive evaluation of the state-of-the-art security strengthening and privacy preserving techniques in IoT security and smart home research. In essence, we leverages the most widely used ML and DL models to design and implement these attack models from the perspective of external on-path adversaries. *SmartAttack* also provides user options to select the detailed configuration for each attack model, such as kernel, dataset splitting, cross-validation states, and evaluating metrics. We also evaluate the performance of *SmartAttack* using two different datasets. In evaluating our hypothesis, this paper makes the following contributions.

Design Challenges. We highlight numerous challenges that we met when designing the new adversarial attack framework—*SmartAttack*. The performance of adversarial attack models are affected by numerous variables, including data pre-processing, kernel selection, hyper-parameter tuning, cross-validation, random states, etc. SmartAttack addresses these issues by enabling both empirical and user selective configurations to build adversarial attack models. In addition, SmartAttack has to handle large amount of time-series network traffic and supports both fast-speed and low-speed granularities data resampling.

SmartAttack Design. We present the design of SmartAttack, a new open-sourced adversarial ML-based and DL-based attack model framework that can enable a wide set of general sophisticated adversarial attack models for researchers and industrial users to benchmark and evaluate their security and privacy research work. In essence, SmartAttack enables users to select the detailed configuration for each attack model, such as kernel, datasetsplitting, cross-validation states, and evaluating metrics. SmartAttack also presents the most effective attack model per each user in-home activity threat.

Implementation and Evaluation. We implement the new SmartAttack toolkit using Python and also other widely available open-source frameworks. We evaluate SmartAttack using two different datasets—UNSW Sydney Smart Home Dataset [21], and our own "mock" smart home dataset that we deployed and collected in our lab space. The results show that SmartAttack is capable of detecting all 12 smart home user activities with considerable robustness (most of the activities can be reliably detected by more than one ML/DL models.)

Releasing Datasets and Code. We release all the datasets and the source code of SolarFinder on our website [22] such that other researchers may use SmartAttack to benchmark their future work.

II. Background

In this section, we first review attack models that the most recent research used to benchmark and evaluate their work. We then analyze the major issues of existing models that motivate our new open-source adversarial attack toolkit—SmartAttack.

A. Attacking Scenario

Below we present two typical attacking scenarios of the most recent work that defines an external adversary's capability. We introduce the models and evaluate their feasibility on real attacks.

Monitoring Network Traffic Traces. An attack scenario is, "a victim visits the attacker's website, which contains a malicious script that communicates with IoT devices on the local network that have open HTTP servers" [23]. Some prior research [6], [24] also assumes that external adversaries have device-level information. In this circumstance, attackers are able to capture packets generated by IoT devices, which means there are no obstacles on identifying device traffic and extracting user activity. However, in normal circumstances, as we described above, no device-level information should be presented in the network traffic traces that ISPs shared with third-parties.

Monitoring Network Traffic Volume Data. Prior work [11], [12] sets more limitations to external adversaries by only providing network traffic volume data. In this case, distinguishing devices can no longer rely on capturing packet headers. Furthermore, the overlapping of traffics generated by simultaneously awakened devices will hide important fingerprint information. Comparing with the former scenario, this one gives a better simulation since attackers are only capable of monitoring network traffic volume data. Unfortunately, the traffic volume data that ISPs provided to third-parties could be further processed by re-sampling, which may reveal user-in home activity at different granularities.

B. Evaluation Metrics

We next outline the most widely used evaluating metrics that are employed by prior work to evaluate their attacking protecting approaches. We analyze the accuracy and robustness to better understand the limitations of these evaluation metrics.

Adversary Confidence. Adversary Confidence is defined as the expected ratio of correct activity inferences to attempted activity inferences by an adversary with no prior knowledge

when traffic rate metadata is defended by a particular technique. The Adversary Confidence c is:

$$c = \frac{np}{np + n(1-p)q} = \left(1 + \frac{(1-p)q}{p}\right)^{-1} \quad (1)$$

where n is the given time periods, p is the probability that user activity occurs idependently, and q is the probability that the decision function chooses to start non-activity padding independently during any time period. [24].

Accuracy and Root Relative Squared Error (RRSE). Accuracy are applied to evaluate an IoT device classifier in the work [21]. The author calculated accuracy based on true positive rate and false positive rate, along with RRSE to monitor the error rate. Given TP as true positive, FP as false positive, TN as true negative, FN as false negative, P_{ij} as the value predicted by the individual model i for record j and T_j as the target value, the accuracy ACC and RRSE E_i can be calculated as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$E_i = \sqrt{\frac{\sum_{j=1}^{n}(P_{ij}) - T_j)^2}{\sum_{j=1}^{n}(T_j - \frac{1}{n}\sum_{j=1}^{n} T_j)^2}} \quad (3)$$

F1 Score. The F1 score is defined as a weighted average of the precision and recall, where an F1 score reaches its best value at 1, and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

$$F1 = \frac{2 * (precision * recall)}{(precision + recall)} \quad (4)$$

However, to evaluate user activities privacy preserving research in a smart home, we often have to handle a highly unbalanced data—positive and negative values are highly unequal. That says, true negative and false negative should also be taken into consideration, thus these metrics would not accurately present the performance of a ML or DL model.

C. Summary

As we can see from the prior work above, each prior work often user its own set of attack model, dataset, and metrics to evaluate its performance regarding to data security and privacy issues. It is difficult for other researchers to directly understand the benefits and limitations among the prior work. In addition, many attack models that are leveraged in the prior work did not provide details about how their classifiers are trained (e.g., supervised, re-trained, unsupervised), how their datasets are splitted or cross-validated for training or testing. These different setups and configuration may significantly affect the performance of prior work. These above insights have guided the design of our new open-sourced and configurable adversarial attack toolkit—SmartAttack.

III. Design Challenges

Our goal is to design a open-sourced deep learning-based adversarial attack model framework—*SmartAttack*, which enables a set of general sophisticated adversarial attack models that can be leveraged by IoT researchers and IoT industrial users to better evaluate their work. This toolkit should be capable of running across different IoT related platforms, while still keeping the same efficiency when attacking security strengthening and privacy preserving approaches. Also, to fit various research scenarios, the toolkit should provide user selective configuration options for customizing model training process. Unfortunately, none of these issues has been completely fulfilled in prior work. In this section, we highlight these challenges that we handled to design SmartAttack.

A. Platform Applicability and External Dependencies

To ensure our SmartAttack toolkit capable of deploying on different platforms, we investigated several widely used IoT applications, such as consumer, organisational, and industrial applications [25]. These scenarios typically have limitations such as highly depending on some specific operating systems, having special dependencies on certain program environment, or significant training time and thus significant energy consumption. Unfortunately, we observe that many prior work [23], [6], [24], [11], [12] have attacking scenarios that require significant energy consumption, which are difficult to deploy on many embedded systems or single-board computers (SBC). Furthermore, we have to guarantee the external dependencies to be open-sourced and lightweight, which would allow end user don't have to configure or retrain their ML or DL models to initialize the adversarial attacks using GPUs or remote servers. To address this challenge, we developed the attack model in $Python$, and we leveraged Scikit-Learn [26] for training ML/DL based attacking models.

B. Big Data Pre-processing

We study on the publicly-available IoT traffic traces from UNSW Sydney [21], which are 6.4 GB $pcap$ file that contains packet-level network traffic trace data of 22 IoT devices for 20 days. Training or testing directly on this significant amount of IoT network traffic traces is inefficient for attack model training and hype-parameter tuning. Furthermore, in order to simulate the various types of time-series data that are exposed to external adversaries, we need to pre-process the IoT network traffic traces into different formats as well as different granularities. We leveraged Pandas [27] and Scikit-Learn [26] for data preprocessing. In addition, the data pre-procssing module of our attack model toolkit—SmartAttack supports the most common data file formats, including $pcap$, $pcapng$, txt, csv, and we have successfully tested at different granularity as of per second, per minute and per hour level. To overcome the evaluation limitation using a single dataset, we deployed a new "mock" testbed in our lab space to provide SmartAttack users another complete IoT traffic dataset.

C. User Tunable Model Training

Many prior approaches assume the smart homes they are targeting at have static deployment of IoT devices, and thus typically these approaches developed a static attack model using certain special ML or DL models. In addition, many

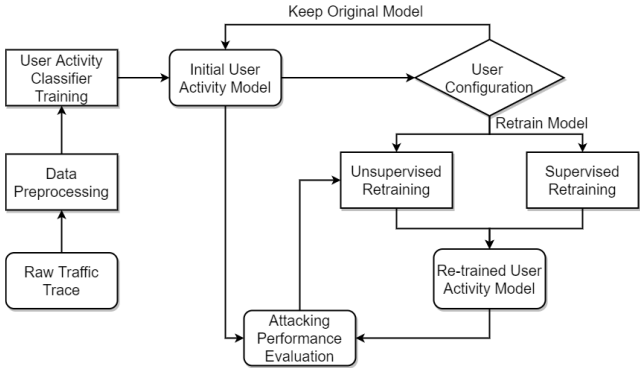


Fig. 2. The pipeline for SmartAttack toolkit.

existing work did not explain how they setup their experimental evaluations, and rather provided a "black-box" models for users to use. For instance, how to split the whole dataset into training, testing and validation portions may result in significant biased evaluation results. Thus, we have to provide configurable attacking models to fit different research scenarios. In addition, it is important to ensure the same data preprocessing processes are applied across the different ML or DL models that are being evaluated. To address this challenge, SmartAttack enables a wide range of user tunable "functions", such as dataset splitting, kennels, hypo-parameter tuning, grid searching methods and evaluating metrics, which allow users to perform a comprehensive evaluation using a whole set of ML or DL models with their own configuration preferences based on their detailed research problem.

IV. SmartAttack Design

In this section, we detailed how we design a new open-sourced adversarial attack model framework—SmartAttack, which enables a set of general sophisticated adversarial attack models that can be leveraged by researchers and industrial users of IoT security community to benchmark and evaluate their work. SmartAttack is capable of attacking on smart home IoT network traffic traces to learn user sensitive activities, and provides a fair attacking performance evaluation on different security strengthening and privacy preserving techniques.

A. System Design

Figure 2 shows the system design pipeline of our new adversarial machine learning toolkit—SmartAttack. ML and DL algorithms usually benefit from standardization of the datasets. First, we leverages the big data analytical approaches, including both standardization and normalization algorithms, to pre-process the input smart home network traces. Next, we build a wide set of adversarial attack models using ML and DL models. Note, rather than the prior work only provides user a "black-box" functions for this modeling traing process, SmartAttack fully enables the user selective options during this modeling process. For the general users that is not able to provide their own choices and prefer to receive the decent performance in a "unsupervised" manner, SmartAttack can recommend the most effective attacking model based on its large scale empirical observations per user in-home activity attack. Eventually, SmartAttack has integrated with many fair and the state-of-art evaluating metrics to assist users to fully benchmark their proposed research work.

B. Traffic Data Pre-processing

SmartAttack is designed and optimized to handle large scale IoT traffic data. In particular, SmartAttack can provide a wide of data pre-processing services, such as data cleaning, data standardization and data normalization. The goal of this data pre-processing is to remove marginal outliers in the input dataset. In addition, SmartAttack already provides the ready-to-use features that have been proven to be very effective at attacking smart home traffic data by many prior research work [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], including our own [28].

For instance, to pre-process UNSW dataset, which is packet-level network traffic traces of 22 IoT devices for 20 days, we first remove the most device-level information including packet headers and contents from the raw $pcap$ files. Then, we down-sampled the original dataset into 20 day's 1 minute-level dataset which has the size of 2.4MB csv file and contains 28000 rows. We leverage $Pandas$ to further process this training dataset. As we expected, the total training time has significantly reduced for attack model training and only requires less than 20 seconds on an Intel(R) Core(R) CPU i9-9900k @ 4.70 GHz. Similarly, we setup a "mock" smart home with 5 occupants and 16 IoT devices, and SmartAttack has been tested to be effective as well. We have successfully tested different formatting of time-series data including $pcap$, $pcapng$ and csv, as well as different time granularity including 1 second, 1 minute and 1 hour level, respectively. The pre-processing module of our SmartAttack stays efficiently operative when the duration of the input dataset varing from 7 to 30 days, which is sufficient to train regular attack models.

C. User Tunable Adversarial Machine Learning Modeling

Once the input dataset (and its principle features) has been pre-processed, SmartAttack then enables users to customize the training and testing processes of different attack models. The goal is to provide users the fully control over the attack model building process such that users can "tailor" this modeling based on their own needs to fit different research scenario. In essence, SmartAttack enable users to select the following principle "parameters" in the model training, testing and validation process.

Kernel Selection and Grid Search. Although most ML and DL models have been proved to be effective on extracting user sensitive in-home activities from network traffic traces, the different ML and DL models have shown different accuracy performance over the same dataset. In addition, when training the same Ml/DL model, the different configuration selections often result in significant different levels of attacking accuracy. In the current released SmartAttack, we implemented the following ML/DL models that perform well in extracting smart home user activities: Decision Tree, Logistic Regression, K-Nearest Neighbours (kNN), Naive Bayes, Random Forest, and

Support Vector Machines (SVMs) with different kernels. We also provide the a set of grid searching approaches. In addition, to reduce the grid search for the multidimensional parameter spaces, we also provide ready-to-use parameter spaces for users to start to train their models for best efficiency.

Cross Validation. Next, SmartAttack enables users to oversee another critical process—cross validation to train an accurate ML/DL smart home attacking model. The goal of this step is to address the potential over-fitting problem in the attacking model training process. Many ML/DL model learn their parameters for a specific prediction function and test them on the same dataset. However, this training process may result in a overfitting situation: the models can simply repeat the groundtruth labels of the training dataset that it has just seen would report a perfect accuracy score but would simply fail to predict any yet-unseen testing dataset. SmartAttack enables users to mitigate this issue by allowing users to specify a range of n—cross folds, m—random states, and grid search parameter spaces. In addition, user activity information is highly time-sensitive, which sets a limit to cross validation n—cross folds and m—random states. For instance, going back to a home typically would generate the network traffic of door sensors, TV, laptop and other regular home use IoT devices. Simply crossing over the dataset would remove the features that the potential ML/DL attacking model can leverage to attack IoT devices and their user privacy information. To address this problem, SmartAttack learns a proper sliding window size—s when training user activity attacking model. In SmartAttack, we also provide a pre-set n and m which have good prior cross performance in general smart home attacks.

Training and Testing Dataset Splitting. Many ML/DL bases system users do not set up or calibrate the sample ratio of training to testing of their models. Different splitting ratios have resulted in significant different accuracy level of smart home attack models. SmartAttack provides empirical ratios for each smart home network traffic attack. SmartAttack enables users to customize their dataset splittings such that they can train a reasonably accurate Ml/DL classifier in a timely matter. However, smart home users typically have their activity routines at the time scales of one day level and one month level. SmartAttack learns the suggested user "tunable" and default splitting ratios for different attack scenarios.

Pre-Trained vs. Re-Trained. In general, SmartAttack supports users to train and test their attack models in both pre-trained and re-trained modes. In real practice, SmartAttack supports users to pre-train their ML/DL based attacking models and then deploy these attacking models on IoT hubs, routers, or other hardware resource limited IoT devices to initialize these sophisticated adversarial attacks. However, SmartAttack can also support onsite re-trained models when the potential IoT hardware deployment can support these DL/ML classifier re-training. It is expected that re-trained adversarial attack models are having better performance than the pre-trained ones. In addition, SmartAttack periodically refreshes the detected list of IoT devices in a smart home, and provide users the recommendations proactively to select either pre-trained or re-trained mode for a given IoT device. Rather than the pure or native the existing attacking models proposed in prior work, SmartAttack leverages adversarial ML and DL techniques to design and implement a wide set of effective attack models that are aiming at learning user in-home activity without smart home users' authorizations. The (external) adversaries can apply any data mining or advanced inferring techniques to "hack" the IoT devices, by understanding the underlying fundamental relationship between user behaviors and the network traffic traces generated by these IoT devices.

D. Fair Performance Evaluating

As we had discussed in the background section, many prior work use their own evaluating metrics, rather than use general evaluating metrics that can enable users to clearly understand their approach's benefits and limitations straightforwardly. In addition, many IoT traffics are not reflecting continuous status. For instance, rather the significant and high frequency network traffic traces generated by a laptop or TV, the duration for many other IoT devices like door sensors, occupancy sensors are short, and it also has a *relatively* low frequency (a.k.a. inactive class devices in this paper). That says, in real practice, the IoT traffic dataset we used or collected, are highly unbalanced regarding certain IoT devices' appearance frequency in the whole traffic traces. For instance, in the UNSW dataset, we identified 12 user activities and most of them have a roughly 50:1 ratio between active and inactive classes. To address this issue, SmartAttack implements or integrates with many fair and robust metrics such that researchers can directly benchmark their work's performance against other related work. We outline two primarly metrics—Matthews Correlation Coefficient (MCC) and Cohen's Kappa (CK).

Matthews Correlation Coefficient (MCC). To quantify the accuracy of different user privacy enhancing approaches, we note that the standard evaluating metrics, e.g, accuracy, F1, would not work well on our highly imbalanced IoT traffic data. Based on the recommendation from prior work [29], [30], we use the MCC [31], a standard measure of a classifier's performance, where values are in the range -1.0 to 1.0, with 1.0 being perfect user activity detection, 0.0 being random user activity prediction, and -1.0 indicating user activity detection is always wrong. The expression for computing MCC is below, where TP is the fraction of true positives, FP is the fraction of false positives, TN is the fraction of true negatives, and FN is the fraction of false negatives, such that TP+FP+TN+FN$= 1$.

$$\frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (5)$$

Cohen's Kappa (CK). The Cohen's kappa [32] is a measure of the agreement between two classifiers who each classify N items into C mutually exclusive categories. The definition of Cohen's Kappa is as follows,

$$\kappa = 1 - \frac{1 - p_o}{1 - p_e} \quad (6)$$

where p_o is the relative observed agreement among classifiers, and p_e is the hypothetical probability of chance agreement,

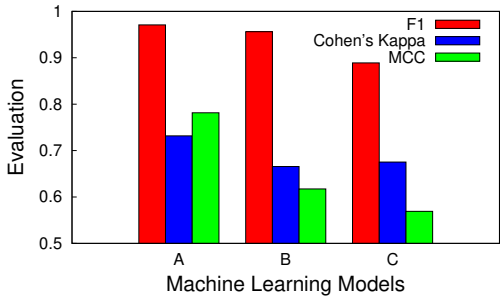


Fig. 3. The comparison results of 3 different attack models using F1, MCC, and Cohen's Kappa, respectively.

using the observed data to calculate the probabilities of each classifier randomly seeing each category. If the classifiers are in complete agreement then κ should be 1. If there is no agreement among the classifiers other than what would be expected by chance, $\kappa = 0$. SmartAttack employed the general and fair metrics—Matthews Correlation Coefficient (MCC) and Cohen's Kappa (CK) to evaluate the researchers' new security and privacy defending approaches.

$$F_1 = 2TP/(2TP + FP + FN) \tag{7}$$

F1 score is able to reflect the true accuracy of a binary ML/DL classifier when the input dataset is well-balanced. However, as shown in Figure 3, we evaluate three different ML-based attack models, and we find that different metrics show different results. This is mainly due to the fact that the input smart home network trace dataset is highly imbalanced, and F1 score is not considering the significant true negatives (TN) in testing dataset, as shown in the Equation IV-D. SmartAttack reports the accuacy for this kind of models using MCC an CK, which can evaluate the true accuracy for each model in fair and comperhensive manner.

V. Implementation

We implement the SmartAttack prototype in *Python* using multiple widely available open-sourced frameworks, including Pandas [27], and Scikit-learn [26]. SmartAttack takes IoT traffic data as input, applies data pre-processing and re-sampling, and leverages multiple Ml/DL methods to attack smart home user in-home activity information. For data pre-processing, we use Pandas for efficiently edit the *csv* files. The processing time for the 2.4MB, 28000 rows file can be limited to less than 10 minutes on a Raspberry Pi 3. We use the Scikit-learn ML library in *Python* to implement the attacking on user activities. We implement a wide set of ML models for SmartAttack, including Decision Tree, Logistic Regression, K-Nearest Neighbours, Naive Bayes, Random Forest, and Support Vector Machine (SVM). We also implemented different kernels for SVM classifiers, including linear, linear-passive-aggressive, linear ridge, polynomial with $1 \sim 10$ degrees, and radial basis function (RBF). We have already optimized the parameters for these models an provide multiple user "tunable" options that allow users to customize parameter setting (via grid search), dataset splitting and pre-train/re-train options. We

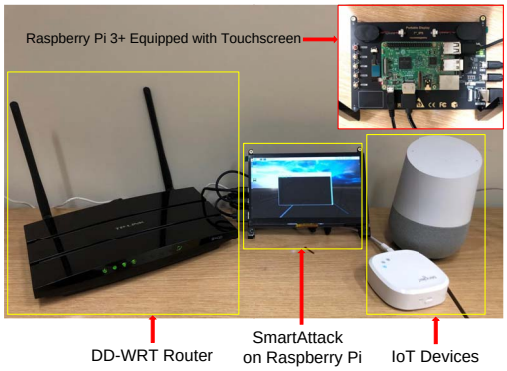


Fig. 4. The overview of SmartAttack prototype

also deploy a SmartAttack testbed (shown in Figure 4) in our lab space. The network traffic data is captured from the smart Wi-Fi router and periodically synchronized with Raspberry Pi 3B for processing. All of SmartAttack adversarial attack models are learned and evaluated on Raspberry Pi.

VI. Experimental Evaluation

In this section, we describe datasets, evaluating metrics, and evaluation results for our SmartAttack toolkit.

A. Datasets

We use traffic traces from 1 "mocked" smart home and 1 public dataset for the evaluation of SmartAttack. Both datasets are resampled to 3 different granularities— 1 second, 1 minute and 10 minutes level to test the robustness of our approach.
Dataset 1: UNSW: We downloaded publicly-available the IoT traffic traces from UNSW Sydney website [21] that contains packet level network traffic data of 22 IoT devices for 20 days. The raw traffic data was directly captured from the router as *pcap* format, which contains detailed packet information including header and body, as well as payloads. To evaluate our attacking approaches of SmartAttack and perform a proper simulation to an external adversary, we cleaned the data to drop packet level information, and resample the traffic payload as needed. Since this dataset didn't provide ground truth information for user activities, we write a *python* script to automatically label user activities based on IoT traffic patterns.
Dataset 2: SmartFIU: We set up a "mocked" smart home in our lab spaces which has 4 students staying in the room and operates 31 IoT devices daily. We first set a NETGEAR AC1750 smart Wi-Fi router flashed with DD-WRT—a Linux based open-source firmware. All the traffic data generated by the IoT devices was captured through *tcpdump* and stored to a 128GB USB drive connected to the router. Similar with dataset 1, our raw *pcap* files are cleaned and resampled. We recorded all of our interaction with IoT devices as well as occupancy status information as groundtruth for evaluation.

B. Evaluating Metrics

We use the metrics—Matthews Correlation Coefficient (MCC) to evaluate the performance of different SmartAttack approaches. The details of MCC and CK have been discussed in the design section.

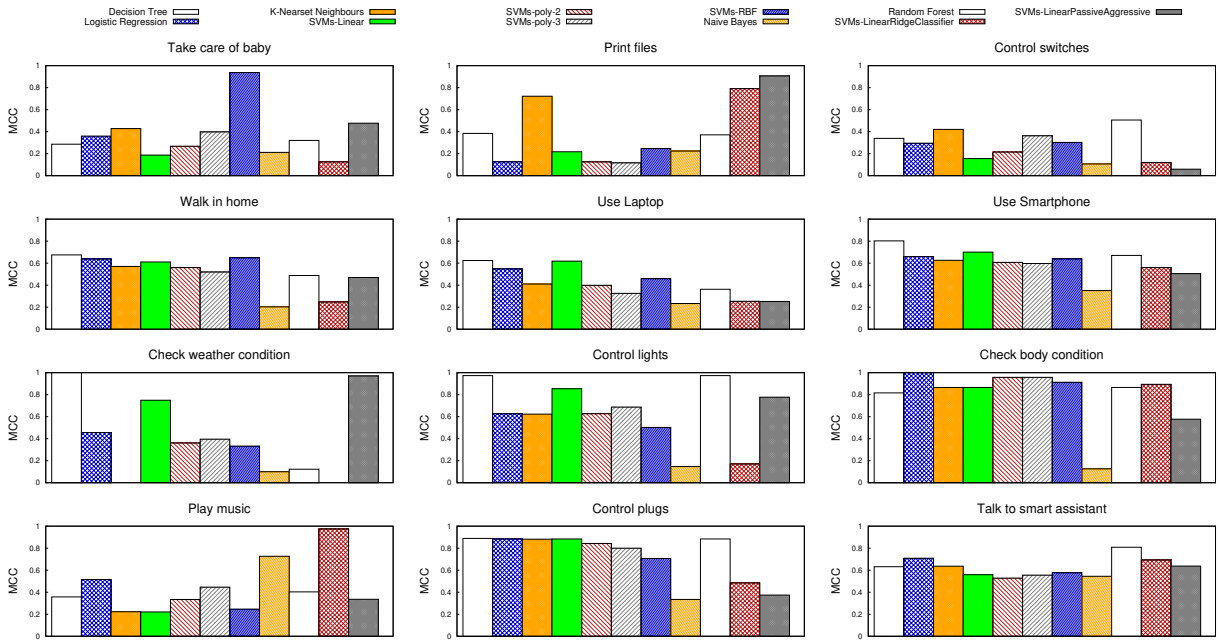


Fig. 5. SmartAttack performances on 13 user activities.

C. Experimental Results

We next evaluate the performance of SmartAttack in 2 folds. First, we performance all the ML/DL-based adversarial attack which are provided in SmartAttack as shown in Figure 5, and then report the best performing attack model for each user in-home activity as shown in Table I.

User "tunable" evaluation. As shown in Figure 5, SmartAttack enables users to select the 11 different ML/DL models as their adversarial attack models. Interestingly, even the same ML/DL-based adversarial attack model has significant different performance when attacking different user in-home activities. For instance, SVM (with Linear Ridge Classifier) reports the MCC as of 0.97 when attacking the activity (User is listening to music?), while only yields a MMC of attacking the user activity (Taking care of baby?). As we discussed in the design section, this is mainly due to the fact that how significant, how long and how often per each specific IoT device event. Similar results can be observed on other ML classifiers as well. In addition, as we can see from Figure 5, for the same SVM models but different kernels and grid search methods also yield significant different attacking performance. These results also reflect the necessity of our insight—enabling users the options to costimzie or tune their ML/DL classifiers are critical in evaluating security research work.

Results: *SmartAttack enables users the options to select different Ml/DL models to build their attack models. In doing so, SmartAttack can provide users a comprehensive understanding of benefits and limitations using different ML/DL models and their detailed configurations to attack smart home users.*

Fair and comprehensive evaluations. As shown in Table I, SmartAttack is able to identify the best performing adversarial attack model for each user activities. As expected, at least one adversarial attack model yields a MCC which is > 0.6) among 11 of 12 activities, and all of them also report high Cohen's Kappa. That says, SmartAttack can provide users the most effective attack model which can receive the agreement over all the classifiers. Regarding user activity (Control switches), SmartAttack achieves a highest MCC value as of 0.5056, which is the lowest result among all 12 different user activities. This is mainly due the fact that the network traffic pattern for smart switches is typically a short and sharp pulse signal, which is likely to be treated as noise/background traffic. In addition, Table I indicates that the input dataset is imbalanced sample dataset which has 0$\sim$5% TP and 95$\sim$99.99% TN. By employing MCC and CK, SmartAttack is still able to evaluate the different attack models in a fair manner.

Results: *SmartAttack yields the MCC values in a range of 0.62$\sim$1.00 among 11 of 12 user in-home activities. Thus, SmartAttack is capable of effectively and accurately detecting user activities from smart home IoT traffic traces.*

Limitation: For IoT devices that have traffic patterns as of short duration or low payload, SmartAttack achieves an MCC value as $\sim$0.5. This is mainly due the fact that the network traffic generated by these IoT devices (e.g., smart switches, wemo plugs) can be easily mislabelled as outliers in traffic traces. We plan to leverage generative adversarial networks (GAN) to address this issue in future work.

VII. Conclusion

This paper presents SmartAttack, a new adversarial attack model framework that can be leveraged by researchers and users from IoT security community to better evaluate their work. In essence, we leverages the most widely used machine learning and deep learning models to design and implement these attack models. SmartAttack also provides user options to select the detailed configuration for each attack model, such as kernel, dataset splitting, cross-validation states, and evaluating

User Activities	Model	TP	FN	TN	FP	MCC	Cohen's Kappa
Talk to smart assistant	Random Forest	1.34%	0.54%	98.01%	0.11%	0.8082	0.6964
Control switches	Random Forest	0.22%	0.25%	99.34%	0.19%	0.5056	0.7813
Print files	SVMs (LinearPassiveAggressive)	0.12%	0.02%	99.86%	0.00%	0.9128	0.9127
Take care of baby	SVMs (RBF)	0.13%	0.01%	99.86%	0.00%	0.9534	0.9539
Use smartphone	Decision Tree	2.99%	0.72%	95.60%	0.69%	0.8023	0.9970
Use laptop	Decision Tree	0.24%	0.13%	99.48%	0.15%	0.6238	0.8719
Walk in home	Decision Tree	3.86%	1.57%	92.72%	1.85%	0.6750	0.9413
Check body condition	SVMs (poly-2)	0.13%	0.01%	99.86%	0.00%	0.9574	0.9576
Control lights	Decision Tree	0.21%	0.00%	99.78%	0.01%	0.9732	0.9990
Check weather condition	Random Forest	0.19%	0.00%	99.81%	0.00%	1.0000	1.0000
Play music	SVMs (LinearRidgeClassifier)	0.18%	0%	99.79%	0.01%	0.9697	0.9706
Control plugs	Decision Tree	0.19%	0.02%	99.76%	0.02%	0.8887	0.9997
Other Activities	Logictic Regression	81.56%	2.83%	11.77%	3.84%	0.7405	0.7725

TABLE I
THE BEST PERFORMING ATTACK MODELS TO DETECT 13 DIFFERENT USER ACTIVITIES USING UNSW SYDNEY DATASET [21].

metrics. We evaluate our approach using 2 different smart home datasets. The results show that SmartAttack can accurately detect 12 user in-home activities, and thus can employed by IoT security researchers to evaluate their potential security strengthening and privacy preserving approaches.

Acknowledgements. This research is supported by Cyber Florida Collaborative Seed Program.

REFERENCES

[1] Statista, "Internet of Things Connected Devices Installed base Worldwide from 2015 to 2025 (in billions)," https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/, 2016.

[2] "Security Cameras Can Tell Burglars When You're Not Home, Study Shows," https://www.cnn.com/2020/07/06/tech/home-security-cameras-risks-scli-intl-scn/index.html.

[3] T. Brewster, "Now Those, Privacy Rules Are Gone, This Is How ISPs Will Actually Sell Your Personal Data," https://www.forbes.com/sites/thomasbrewster/2017/03/30/fcc-privacy-rules-how-isps-will-actually-sell-your-data/#2e30f75921d1, Mar 2017.

[4] Mirimir, "Collection of User Data by ISPs and Telecom Providers, and Sharing with Third Parties," https://www.ivpn.net/blog/collection-of-user-data-by-isps-and-telecom-providers-and-sharing-with-third-parties, March 15th 2018.

[5] A. Ukil, S. Bandyopadhyay, and A. Pal, "Iot-privacy: To be private or not to be private," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2014, pp. 123–124.

[6] N. J. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, and N. Feamster, "Spying on the smart home: Privacy attacks and defenses on encrypted iot traffic," *CoRR*, vol. abs/1708.05044, 2017. [Online]. Available: http://arxiv.org/abs/1708.05044

[7] S. Andy, B. Rahardjo, and B. Hanindhito, "Attack scenarios and security analysis of mqtt communication protocol in iot system," in *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2017, pp. 1–6.

[8] A. Lakhina, M. Crovella, and C. Diot, "Characterization of network-wide anomalies in traffic flows," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. ACM, 2004, pp. 201–206.

[9] G. A. Bergman and N. Liu, "Hvac controller with wireless network based occupancy detection and control," Oct. 25 2016, uS Patent 9,477,239.

[10] K. L. Barrett, S. Buckley, C. Gannon, and M. Minhazuddin, "Dynamic feedback from an internet service provider about network occupancy/availability," Sep. 6 2011, uS Patent 8,014,273.

[11] R. Melfi, B. Rosenblum, B. Nordman, and K. Christensen, "Measuring building occupancy using existing network infrastructure," in *2011 International Green Computing Conference and Workshops*. IEEE, 2011, pp. 1–8.

[12] S. Meyn, A. Surana, Y. Lin, S. M. Oggianu, S. Narayanan, and T. A. Frewen, "A sensor-utility-network method for estimation of occupancy in buildings," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, 2009, pp. 1494–1500.

[13] A. Fichou, C. Galand, and J.-F. Le Pennec, "Connections bandwidth right sizing based on network resources occupancy monitoring," Jul. 20 2004, uS Patent 6,765,873.

[14] T. A. Nguyen and M. Aiello, "Energy intelligent buildings based on user activity: A survey," *Energy and buildings*, vol. 56, pp. 244–257, 2013.

[15] R. P. Morris, "System and method for tracking user activity related to network resources using a browser," Dec. 8 2009, uS Patent 7,631,007.

[16] W. Ding and H. Hu, "On the safety of iot device physical interaction control," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 832–846.

[17] H. Yu, J. Lim, K. Kim, and S.-B. Lee, "Pinto: Enabling video privacy for commodity iot cameras," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 1089–1101.

[18] Q. Wang, W. U. Hassan, A. Bates, and C. Gunter, "Fear and logging in the internet of things," in *NDSS*, 2018.

[19] W. Zhang, Y. Meng, Y. Liu, X. Zhang, Y. Zhang, and H. Zhu, "Homonit: Monitoring smart home apps from encrypted traffic," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018.

[20] R. Schuster, V. Shmatikov, and E. Tromer, "Situational access control in the internet of things," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 1056–1073.

[21] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying iot devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, 2019.

[22] "SmartAttack," https://github.com/cyber-physical-systems/SmartAttack.

[23] G. Acar, D. Y. Huang, F. Li, A. Narayanan, and N. Feamster, "Web-based attacks to discover and control local iot devices," in *Proceedings of the 2018 Workshop on IoT Security and Privacy*. ACM, 2018.

[24] N. J. Apthorpe, D. Y. Huang, D. Reisman, A. Narayanan, and N. Feamster, "Keeping the smart home private with smart(er) iot traffic shaping," *CoRR*, vol. abs/1812.00955, 2018. [Online]. Available: http://arxiv.org/abs/1812.00955

[25] "Internet of things," https://en.wikipedia.org/wiki/Internet_of_things#Consumer_applications.

[26] "Scikit-learn Machine Learning in Python," https://scikit-learn.org/stable/.

[27] "Pandas," https://pandas.pydata.org/.

[28] L. Deng, Y. Feng, D. Chen, and N. Rishe, "Iotspot: Identifying the iot devices using their anonymous network traffic data," in *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*. IEEE, 2019, pp. 1–6.

[29] J. Akosa, "Predictive accuracy: a misleading performance measure for highly imbalanced data."

[30] S. Picek, A. Heuser, A. Jovic, S. Bhasin, and F. Regazzoni, "The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations," 2018.

[31] "Matthews Correlation Coefficient," https://en.wikipedia.org/wiki/Matthews_correlation_coefficient.

[32] "Cohen's Kappa," https://en.wikipedia.org/wiki/Cohen\%27s_kappa.