Poster: AUTOLYCUS: Exploiting Explainable AI (XAI) for Model Extraction Attacks against Decision Tree Models

Abdullah Caglar Oksuz
Case Western Reserve University
abdullahcaglar.oksuz@case.edu

Anisa Halimi
IBM Research
anisa.halimi@ibm.com

Erman Ayday
Case Western Reserve University
erman.ayday@case.edu

***Abstract*—Model extraction attack is one of the most prominent adversarial techniques to target machine learning models along with membership inference attack and model inversion attack. On the other hand, Explainable Artificial Intelligence (XAI) is a set of techniques and procedures to explain the decision making process behind AI. XAI is a great tool to understand the reasoning behind AI models but the data provided for such revelation creates security and privacy vulnerabilities. In this poster, we propose AUTOLYCUS, a model extraction attack that exploits the explanations provided by LIME to infer the decision boundaries of decision tree models and create extracted surrogate models that behave similar to a target model.**

I. Introduction

Model inversion [1] and model extraction (stealing) attacks [2] are advanced state-of-the-art adversarial techniques that compromise the security and privacy of machine learning (ML) models. These attacks extract sensitive information from the models such as training data and hyper-parameters. Increasing number of ML-as-a-service systems which provide ML models trained with confidential data and these models' publicly accessible query interfaces are investigated in the literature for growing security and privacy concerns.

Investigation of the interpretability of black-box AI models gave rise to a new research topic Explainable Artificial Intelligence (XAI) [3]. XAI aims to describe the inner workings of AI models -that are challenging to be understood by humans- such that the model accuracy, fairness, transparency, and outcomes can be characterized [4]. One such method is local interpretable model agnostic explanations (LIME) [5] that identifies the most important features (that influences the decision-making of AI) of a given sample by generating locally surrogate dataset around that sample.

Capabilities of adversarial ML attacks can be enhanced by combining the additional vulnerabilities XAI produces. Hence, we propose a novel model extraction attack called AUTOLYCUS that employs model explanations from LIME to generate less queries than the state-of-the-art attacks for revealing the decision boundaries of decision tree models and extract those boundaries to locally trained surrogate models. AUTOLYCUS also can draw on information from the samples obtained by other resources. These resources may include previously sampled queries and other datasets. We demonstrate the effectiveness of our method in different datasets by using accuracy, R_{test} and # of queries metrics.

II. Related Work

There exist known privacy attacks to machine learning models, such as membership inference attacks to determine whether a target data sample was part of the training dataset or not, attribute inference attacks to infer sensitive information based on the observed ones, model inversion attacks [1] to reconstruct data samples from a target ML model, and model extraction attacks [2] to extract model parameters/hyper-parameters from a target ML model. With the introductions of model explanations [3] such as LIME, SHAP, LRP etc., it was shown that some of the aforementioned attacks (e.g., membership inference attacks and model inversion attacks) can be improved by exploiting explanations. In this work, we investigate how to exploit the explanations provided by LIME [5] to reconstruct a target decision tree model.

III. Methodology

In most ML-as-a-services, non-adversary users such as Alice can send a query $Q(s_i)$ per sample s_i to a model M which is stored on cloud or locally. In order to reflect under what circumstances can AUTOLYCUS be utilized, we set some preliminaries.

For simplicity, we assume that the architecture of M and the general constraints of features in samples are known by Alice. Unknown values in s_i are not permitted. Hence, missing features in s_i should be filled with averages or permitted random values, for M to respond. M responds to $Q(s_i)$ by providing only a prediction label and explanations using features such that the reasoning behind the particular prediction is explained.

An adversary user Bob can send queries $Q(s_i)$ to the model M just like Alice does. However, Bob exploits explanations to conduct a model extraction attack on M using targeted queries. Targeted queries are the new samples $s_i^1, s_i^2, ...$ to be queried, guided by the explanations. These new samples differ from s_i in only single feature such that the decision boundaries of M in prediction are extracted. Recursive procedure of sending queries and creating new samples from the responses of those queries is referred as *TRAV-A* shortened for traversal algorithm in this paper. *TRAV-A* can be initialized by a single sample

filled with valid random values. But for efficiency, having at least a single sample for each prediction label (or node in decision trees) is recommended. Bob can also utilize multiple samples from other resources as auxiliary information for less queries. *TRAV-A* explores and generates new queries until it collects a pre-determined amount of samples from each prediction class. Bob uses these samples as a surrogate dataset to train a new model $M^{'}$ with the same architecture as M.

function TRAV-A(D_{smp}, *Exp*, M_{trg}, c_n, f_n, L_b, U_b)
  samples, v_samples, preds $\leftarrow$ $D_{smp}, D_{smp}[\mathbf{0}], \{\}$
  n_visits $\leftarrow$ $\{0, 0, ...0\}$ ▷ (c_n times)
  query $\leftarrow$ 0
  while samples $\neq \emptyset$ **and** ANY(nvisits) $\leq L_b$ **do**
    query++
    curr $\leftarrow$ samples.POP()
    pred $\leftarrow$ model.PREDICT(curr)
    if n_visits[**pred**] $< U_b$ **then**
      n_visits[**pred**]++
      preds.ADD(pred)
      v_samples.ADD(curr)
      e $\leftarrow$ **PARSE_EXP**(*curr, Exp,* M_{trg}, c_n, f_n)
      cand_samples $\leftarrow$ **GENERATE_SMP**(e)
      for i **in** cand_samples **do**
        if i $\notin$ samples, v_samples **then**
          samples.PUSH(i)
  return samples, preds, query

IV. Experiments and Discussion

A. Datasets & Models & Metrics

For the experiments, we have used Iris, Crop and German Credit (numeric) datasets. The details about the datasets and the decision tree models are provided in Table I. We measured

Table I: Dataset and Model Properties

Dataset	# Samples	# Features	# Classes	Depth	Leaves
Iris	150	4	3	2	3
Crop	2200	7	22	10	26
German Credit	1000	24	2	11	80

the success of our extraction using; model accuracy, R_{test} [2] (classification similarity against a neutral dataset) and the number of queries as our metrics.

B. Results

We evaluated our approach with the datasets mentioned in Section IV-A. Iris dataset experiments yielded comparable accuracy to the target model even if the surrogate dataset is very minimal. That is due to Iris dataset's low tree depth, small node size and small number of features. Experiments on Crop and German Credit datasets were more informative to test the effectiveness of our method. For Crop Dataset, the target model is trained with 800 samples. The model has 97.5% accuracy and a comparable model with 90% R_{test} is achieved with only 50% of the data. German credit dataset consists of both continuous and categorical variables and extracted surrogate samples followed the constraints the datasets have, making them realistic for utilization. In Figure 1, the target model is a decision tree classifier trained with 400 samples

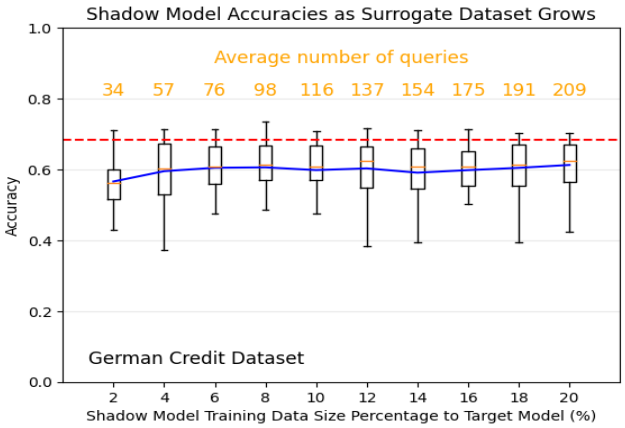

Fig. 1. Impact of surrogate dataset size to the accuracy of the model resulting from model extraction attack

from the German Credit dataset and its accuracy is 68.5%. Bob has access to 50 samples from the German Credit dataset, which consists of 2% of the target's model training dataset. When Bob sends queries of 20% target's data size, with an average of only 209 queries, the surrogate model's accuracy raised from 58% to 63% and $R_{test} = 70\%$ is achieved. In [2] 100% reconstruction in German Credit dataset was achieved with 1722 queries. However, AUTOLYCUS partially reconstructs 70% of the model with 8 times less queries. Hence, it can be utilized as a budget option with softer requirements against platforms that provide AI explanations.

V. Conclusion

In this poster, we demonstrated a new model extraction attack AUTOLYCUS against decision-tree models. AUTOLYCUS exploits XAI to enhance the capabilities of state-of-the-art methods by softening the requirements needed for reconstruction like decreasing the number of queries and utilizing existing samples as auxiliary information.

The work was partly supported by the National Science Foundation (NSF) under grant number OAC-2112606.

References

[1] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1322–1333. [Online]. Available: https://doi.org/10.1145/2810103.2813677

[2] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *Proceedings of the 25th USENIX Conference on Security Symposium*, ser. SEC'16. USA: USENIX Association, 2016, p. 601–618.

[3] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103

[4] "Explainable artificial intelligence (xai)." [Online]. Available: https://www.ibm.com/watson/explainable-ai

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

POSTER: AUTOLYCUS: Exploiting Explainable AI (XAI) for Model Extraction Attacks against Decision Tree Models

Abdullah Caglar Oksuz
axo223@case.edu
Case Western Reserve University

Anisa Halimi
anisa.halimi@ibm.com
IBM Research Lab in Dublin

Erman Ayday
exa208@case.edu
Case Western Reserve University

Abstract

Model extraction attack is one of the most prominent adversarial techniques to target machine learning models along with membership inference attack and model inversion attack. On the other hand, Explainable Artificial Intelligence (XAI) is a set of techniques and procedures to explain the decision-making process behind AI. XAI is a great tool to understand the reasoning behind AI models, but the data provided for such revelation creates security and privacy vulnerabilities. In this poster, we propose **AUTOLYCUS**, a model extraction attack that exploits the explanations provided by LIME to infer the decision boundaries of decision tree models and create extracted surrogate models that behave like a target model.

Preliminaries

- Explainable AI (XAI)
  - XAI aims to describe the inner workings of AI models -that are challenging to be understood by humans- such that the model accuracy, fairness, transparency, and outcomes can be characterized.[4]
  - One such method is LIME! [5]
    - LIME identifies the most important features (that influences the decision-making of AI) of a given sample by generating locally surrogate dataset around that sample.

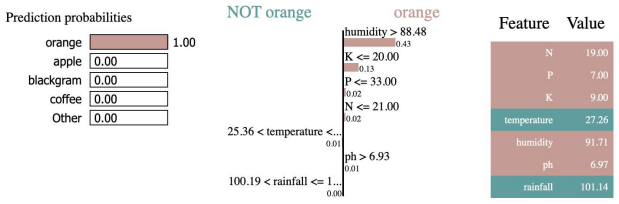

Fig. 1: Example LIME explanation

- Model Extraction Attacks
  - State-of-the-art version for both white-box and black-box models proposed by Tramer et al.[2]
  - Similar attacks have been worked previously as *adversarial learning*.
  - Aims to reconstruct (preferably) identical new model through stealing model parameters or model decision boundaries.

Methodology

- Our **system model** (Fig. 2) is defined as a user Alice can send a query **Q(s)** per sample **s** to an ML-as-a-service platform's model **M** which is stored on cloud or locally.
  - (For experiments and accurate query sending, The metadata about the model like, the training data size of **M**, the architecture of **M** as a decision tree, which features and classes **M** have etc., are known.
  - N/A values are not permitted in **s.**
  - **Q(s)** only returns the predicted class and its explanation; no decision paths or node ids are returned.

- Our **threat model** (Fig. 2) is defined as an adversary user Bob having the similar constraints Alice has. However, Bob exploits explanations to conduct a model extraction attack on **M** using targeted queries.
  - Bob creates new samples s'_1, s'_2, ... to be queried, guided by the explanations. These new samples differ from **s** in only single feature such that the decision boundaries of **M** in prediction are explored.
  - Bob recursively sends queries and creates new samples from the response of those queries by utilizing a traversal algorithm which is referred as **TRAV-A (Fig. 3)**.
  - Bob can utilize multiple samples from other resources as auxiliary information for sending less queries.
  - Bob uses the samples collected by **TRAV-A** as a surrogate dataset to train a new model **M'** with the same architecture as **M**.

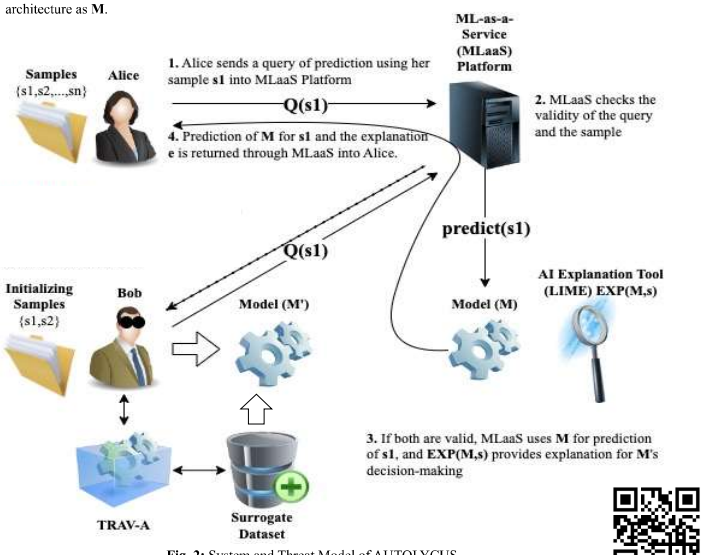

Fig. 2: System and Threat Model of AUTOLYCUS

Evaluation

function TRAV-A(D_{smp}, Exp, M_{trg}, c_n, f_n, L_b, U_b)
  samples, v_samples, preds ← D_{smp}, D_{smp}[**0**], {}
  n_visits ← {0, 0, ...0} ▷ (c_n times)
  query ← 0
  while samples ≠ ∅ **and** ANY(nvisits) ≤ L_b **do**
    query++
    curr ← samples.POP()
    pred ← model.PREDICT(curr)
    if n_visits[**pred**] < U_b **then**
      n_visits[**pred**]++
      preds.ADD(pred)
      v_samples.ADD(curr)
      e ← **PARSE_EXP**(*curr, Exp, M_{trg}, c_n, f_n*)
      cand_samples ← **GENERATE_SMP**(e)
      for i **in** cand_samples **do**
        if i ∉ samples, v_samples **then**
          samples.PUSH(i)
  return samples, preds, query

Fig. 3: Traversal Algorithm Pseudocode

Table I: Dataset and Model Properties

Dataset	# Samples	# Features	# Classes	Depth	Leaves
Iris	150	4	3	2	3
Crop	2200	7	22	10	26
German Credit	1000	24	2	11	80

- For the experiments we used three datasets; Iris, Crop and German Credit (numeric)
- We measured the success of our attack using; model accuracy, R_{test} [2] (classification similarity against a neutral dataset) and the number of queries as our metrics.

- **Iris dataset** yielded comparable accuracy to the target model even if the surrogate dataset that trains the surrogate model is very minimal. That is due to Iris dataset's:
  - Low tree depth,
  - Small node size,
  - Small number of features.

- Experiments on **Crop and German Credit datasets** were more informative to test the effectiveness of our method.
  - In **Crop dataset**, the target model is trained with 800 samples. In Fig. 4, the target model has 97.5% accuracy and a comparable surrogate model with 90% R_{test} is achieved with only 50% of the data.

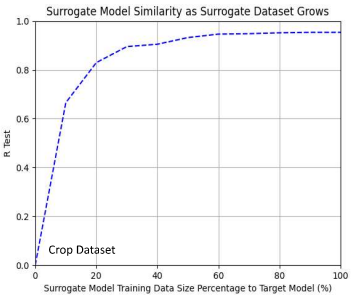

Fig. 4: Plot of R_{test} in Crop dataset

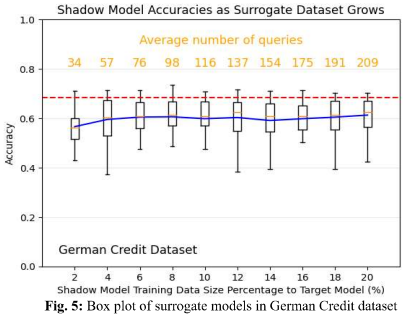

Fig. 5: Box plot of surrogate models in German Credit dataset

- In **German Credit dataset** which consists of both continuous and categorical variables and extracted surrogate samples followed the constraints the datasets have, making them realistic for utilization.
  - In Fig. 5, the target model is a decision tree classifier trained with 400 samples from the German Credit dataset and its accuracy is 68.5%.
  - Bob has access to 50 samples from the German Credit dataset, which consists of 2% of the target's model training dataset.
  - When Bob sends queries of 20% target's data size, with an average of only 209 queries, the surrogate model's accuracy raised from 58% to 63% and R_{test} =70% is achieved.

- For balancing the dataset and preventing the bias on features, not all the queried samples are used in training the surrogate model.

- In [2], 100% reconstruction in German Credit dataset was achieved with **1722** queries. However, AUTOLYCUS partially reconstructs 70% of the model with 8 times less queries.

70% with 200 Q vs. **100% with 1700 Q**

Conclusion

In this poster, we demonstrated a new model extraction attack AUTOLYCUS against decision-tree models. AUTOLYCUS exploits XAI to enhance the capabilities of state-of-the-art methods by softening the requirements needed for reconstruction like decreasing the number of queries and utilizing existing samples as auxiliary information.

The work was partly supported by the National Science Foundation (NSF) under grant number OAC-2112606.

References

[1] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ser. CCS '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1322–1333. [Online]. Available: https://doi.org/10.1145/2810103.2813677

[2] F. Tramer, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in Proceedings of the 25th USENIX Conference on Security Symposium, ser. SEC'16. USA: USENIX Association, 2016, p. 601–618.

[3] A. Barredo Arrieta, N. Diaz-Rodriguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," Information Fusion, vol. 58, pp. 82–115, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103

[4] "Explainable artificial intelligence (xai)." [Online]. Available: https://www.ibm.com/watson/explainable-ai

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778