Original software publication

**secml**: Secure and explainable machine learning in Python

Maura Pintor\textsuperscript{a,b}, Luca Demetrio\textsuperscript{a,b}, Angelo Sotgiu\textsuperscript{a,b}, Marco Melis\textsuperscript{a}, Ambra Demontis\textsuperscript{a}, Battista Biggio\textsuperscript{a,b,*}

\textsuperscript{a} DIEE, University of Cagliari, Via Marengo, Cagliari, Italy
\textsuperscript{b} Pluribus One, Via Vincenzo Bellini, 9, Cagliari, Italy

**A B S T R A C T**

We present \texttt{secml}, an open-source Python library for secure and explainable machine learning. It implements the most popular attacks against machine learning, including test-time evasion attacks to generate adversarial examples against deep neural networks and training-time poisoning attacks against support vector machines and many other algorithms. These attacks enable evaluating the security of learning algorithms and the corresponding defenses under both white-box and black-box threat models. To this end, \texttt{secml} provides built-in functions to compute security evaluation curves, showing how quickly classification performance decreases against increasing adversarial perturbations of the input data. \texttt{secml} also includes explainability methods to help understand why adversarial attacks succeed against a given model, by visualizing the most influential features and training prototypes contributing to each decision. It is distributed under the Apache License 2.0 and hosted at https://github.com/pralab/secml. © 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

**1. Introduction**

Machine learning is vulnerable to well-crafted attacks. At test time, the attacker can stage evasion attacks (i.e., adversarial examples) [1–5], sponge examples [6], model stealing [7], and membership inference [8] attacks to violate system integrity, availability, or even its confidentiality. Similarly, at training time, the attacker may target either system integrity or availability via poisoning [9,10] and backdoor attacks [11]. The most studied attacks, namely evasion and poisoning, firstly explored by Biggio et al. [2,9], are formalized as constrained optimization problems, solved through gradient-based or gradient-free algorithms [10,12], depending on whether the attacker has white- or black-box access to the target system. Many libraries implement the former, however, they do not allow developers to assess machine learning models’ security easily. Hence, we present \texttt{secml}, an open-source Python library that serves as a complete tool for evaluating and assessing the performance and robustness of...
machine-learning models. To this end, secml implements: (i) a methodology for the empirical security evaluation of machine-learning algorithms under different evasion and poisoning attack scenarios; and (ii) explainable methods to help understand why and how these attacks are successful. With respect to other popular libraries that implement attacks almost solely against Deep Neural Networks (DNNs) [13–15], secml also implements training-time poisoning attacks and computationally-efficient test-time evasion attacks against many different algorithms, including support vector machines (SVMs) and random forests (RFs). It also incorporates both the feature-based and prototype-based explanation methods proposed by [16–18].

2. Software description

secml has a modular architecture oriented to code reuse. We have defined abstract interfaces for all components, including loss functions, regularizers, optimizers, classifiers, and attacks. By separating the definition of the optimization problem from the algorithm used to solve it, one can easily define novel attacks or classifiers (in terms of constrained optimization problems) and then use different optimizers to obtain a solution. This is a great advantage with respect to other libraries like CleverHans [13], Foolbox [14,19], and ART [15] as, we can switch from white-box to black-box attacks by just changing the optimizer (from a gradient-based to a gradient-free solver), without re-defining the entire optimization problem.

secml integrates different components via well-designed wrapper classes. We integrate several attack implementations from CleverHans and Foolbox, by extending them to also track the values of the loss function and the intermediate points optimized during the attack iterations, as well as the number of function and gradient evaluations. This is useful to debug and compare different attacks, e.g., by checking their convergence to a local optimum and properly tuning their hyperparameters (e.g., step size and number of iterations). secml supports DNNs via a dedicated PyTorch wrapper, which can be extended to include other popular deep-learning frameworks, like TensorFlow and Keras, and it natively supports scikit-learn classifiers as well, allowing the efficient execution of attacks on sparse data representations; figure implements some advanced plotting functions based on matplotlib (e.g., to visualize and debug attacks); and utils provides functionalities for logging and parallel code execution. We also provide a model zoo, available on GitHub as well, that contains pre-trained models to rapidly test newly implemented attacks and utilities.

We recap the main functionalities of secml in Table 1, where we also compare it with other relevant libraries. Notably, our library is the sole that provides attack loss inspection plots to choose the appropriate attacks’ hyperparameters, and security evaluation plots, to ease the complexity of assessing the robustness of machine learning models. Moreover, the features offered by secml are not only related to attacking machine learning models, but they are gathered to elevate secml as a complete tool for attacking, inspecting, and assessing the performances of machine learning models.

Testing and documentation. We have run extensive tests on macOS X, Ubuntu 16.04, Debian 9 and 10, through the GitHub Actions infrastructure, along with other relevant libraries. Notably, our library is the sole that provides attack loss inspection plots to choose the appropriate attacks’ hyperparameters, and security evaluation plots, to ease the complexity of assessing the robustness of machine learning models. Moreover, the features offered by secml are not only related to attacking machine learning models, but they are gathered to elevate secml as a complete tool for attacking, inspecting, and assessing the performances of machine learning models.

3. Impact

We now offer two examples extracted from secml to showcase its impact: evasion attacks against DNNs, and a poisoning attack against an SVM.

---

Table 1

Comparison of secml and the other adversarial machine learning libraries. We show whether the library offer full (✓), partial (∼) or no (✕) support of a particular feature.

<table>
<thead>
<tr>
<th></th>
<th>DL frameworks support</th>
<th>static template support</th>
<th>Built-in attack database</th>
<th>Wraps adversarial frameworks</th>
<th>Dense/Sparse data support</th>
<th>Security evaluation plots</th>
<th>Loss separated from Optimizer</th>
<th>Explainability</th>
<th>Model zoo</th>
<th>Comprehensiv tutorials</th>
</tr>
</thead>
<tbody>
<tr>
<td>secml</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Foolbox</td>
<td>✓</td>
<td>✕</td>
<td>✓</td>
<td>✕</td>
<td>✕</td>
<td>∼</td>
<td>X</td>
<td>✕</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>ART</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✕</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✕</td>
<td>✕</td>
<td>✕</td>
</tr>
<tr>
<td>CleverHans</td>
<td>✓</td>
<td>✕</td>
<td>✓</td>
<td>✕</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✕</td>
<td>✕</td>
<td>✕</td>
</tr>
</tbody>
</table>

Fig. 2. Adversarial images (CW, PGD, and PGD-patch) representing a race car misclassified as a tiger. For PGD-patch, we also report explanations via integrated gradients.

Fig. 3. Attack optimization. Left: loss minimization; Right: confidence of source class (race car, dashed lines) vs confidence of target class (tiger, solid lines), across iterations.

Evasion attacks. We show here how to use secml to run different evasion attacks against ResNet-18, a DNN pretrained on ImageNet, available from torchvision. This example demonstrates how secml enables running CleverHans attacks (implemented in TensorFlow) against PyTorch models. Our goal is perturbing the image of a race car to be misclassified as a tiger, using the $\ell_2$-norm targeted Carlini–Wagner (CW) attack (from CleverHans), the $\ell_2$ PGD attack implemented in secml, and PGD-patch, where the attacker can only change the pixels corresponding to the license plate [22].

We run all the attacks for 50 iterations, we set the confidence parameter of the CW attack $\kappa = 10^6$ to generate high-confidence misclassifications, and $c = 0.4$, yielding an $\ell_2$ perturbation size $\epsilon = 1.87$. We bound PGD to create an adversarial image with the same perturbation size. For PGD-patch, we do not bound the perturbation size for the pixels that can be modified.

The resulting adversarial images are shown in Fig. 2. For PGD-patch, we also highlight the most relevant pixels used by the DNN to classify this image as a tiger, using the integrated gradients explanation method. The most relevant pixels are found around the perturbed region containing the license plate, unveiling the presence of potential adversarial manipulations.

We also visualize the performances of the attack in Fig. 3. The leftmost plot shows how the attack losses (scaled linearly in $[0, 1]$ to enable comparison) iteration-wise, while the rightmost plot shows how the confidence assigned to class race car (dashed line) decreases in favor of the confidence assigned to class tiger (solid line) for each attack, across different iterations. By inspecting them, we can understand if these attacks have been correctly configured. For instance, by looking at the loss curves on the left, we can understand if the attacks reached convergence or not, thus facilitating tuning of either the step size or the number of iterations. Also, by looking at the plot on the right, it is clear that all the attacks are successful since the confidence of the target class exceeds the score of the original one.

Poisoning attacks. We also show the effect of a poisoning attack provided by secml applied to an SVM classifier implemented in scikit-learn. The experimental setting and code are available in one of our tutorials on GitHub.4

Results of the successful attack are represented in Fig. 4., highlighting the flexibility of secml in applying different strategies to third-party models as well, without the need of customizing them on a particular framework.


4 Conclusions and future work

The secml project was born more than five years ago and we open-sourced it in August 2019. Thanks to an emerging community of users and developers from our GitHub repository, we firmly believe that secml will soon become a reference
tool to evaluate the security of machine-learning algorithms. We are constantly working to enrich it with new functionalities, by adding novel defenses, wrappers for other third-party libraries, and more pretrained models to the secml zoo.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been partly supported by the PRIN 2017 project RexLearn, funded by the Italian Ministry of Education, University and Research (grant no. 2017TWNMH2); by the project ALOHA, under the EU’s H2020 programme (grant no. 780788); and by the project TESTABLE (grant no. 101019206), under the EU’s H2020 research and innovation programme.

References


Fig. 4. Poisoning attacks against an SVM implemented with scikit-learn. The poisoning data (denoted with • in the right plot) induce the model to learn a worse decision boundary.