# Introduction

## Core essay

### Automatically evading classifiers

This essay aims to have automatically evading attacks to two state-of-art PDF malware classifiers named PDFrate and Hidost based on genetic algorithm without changing the basic functionality of the malware.

They use *deletion, replacement, and insertion* to body part of PDF files because body part of PDF files is like tree-structure which makes it easier for attackers to *insertion, deletion and replacement,* where *deletion* is usually preferred, however, it sometimes would be harmful to the functionality of the malware.

For PDFrate and Hidost these two malware classifiers, Hidost is trained with ability against evasion attacks, thus, more robust than PDFrate to evasion attacks.

Essay included in zip file.

Resources

* Malware dataset: <http://contagiodump.blogspot.de/2010/08/malicious-documents-archive-for.html>.
* PDFrate and Hidost (where PDFrate was modified to mimicus, another version of PDFrate, which will be used in my project): <https://github.com/srndic/>
* Claimed essay resource code: <https://evademl.org/>, and I think their codes should be here: <https://github.com/uvasrg/EvadeML>.
* PDF parser (modified strictless grammar checking version): <https://github.com/mzweilin/pdfrw>
* More resources can be accessed in that essay.

### Immune algorithm

The algorithm I’m going to use is immune algorithm. Compared to genetic algorithm, immune algorithm solves the undulation problem of genetic algorithm and speeds up convergence speed and thus improving also performance and accuracy. The author firstly introduces two immune operators after genetic algorithm, namely vaccination and immune selection.

**Important concepts:**

* Antigen: the problem (namely the codes classified by malware classifiers) we need to solve or get rid of.
* Vaccine: the basic information of features extracted from antigen.
* Antibody: dealing with the features information, namely vaccine, we can make a program solving this problem. Antibody is the set of solutions to that program. (e.g. insertion, deletion, balbala)
* **Anyway, we are aiming to make the malware not detected by classifiers by modifying the raw codes.**
* Global annealing algorithm: used in this algorithm, can be accessed by directly browsing the essay included.
* **NOTES:** immune algorithm & global annealing（退火算法） are written in Chinese. I’m answering any questions about these two essay if you cannot understand.

### On training robust pdf malware classifiers (optional)

Rather complicated… not suggested without time. If you can finish the immune algorithm aiming to PDFrate and Hidost and also have time in doing this, let me know.

Simply speaking, they created two new metrics named Estimated Robust Accuracy and Verified Robust Accuracy to evaluate the robustness of their models. They used ABCDE five properties as components building up their models, as well as monotonic model. They have testified that their models profoundly increased L0 distance by 3.6 times while makes attackers need 21 times larger mutation operations for the first essay I mentioned. Within E in their model, essay 1 cannot 100% succeed in generating evasion variant. Faced with monotonic model and model robust A+B+E, it takes two weeks for the evasion attack to generate variant. Even the simplest models against evasion attacks make the attacker 3 days on that.

Except for evasion attack, they have testified that their models are robust to many other attacks, which we’re not talking about.

**NOTES:** there are **no available resource codes** now. therefore I almost give up.

# Idea

My idea is rather simple: using improved immune algorithm instead of genetic algorithm to test if it has better performance and also accuracy. The evaluation might be based on these two indexes:

1. Speed
2. Accuracy

In fact the best way is to attack the latest state-of-art classifiers stated in the On training… essay because their models have decreased efficiency of the automatic evasion attacks, but since there are no codes available… the thing might be easier: compare the statistics including time consumption, success rate, success amount, balabala..

Other requirements

1. Must use python
2. Better use colab.

Deadline:

1. Presentation: April 23. This is not that serious, I can just simply talk about basic functions and algorithms without too many details. Optimization to adaptability function, fitness functions can be not finished by the date. But basic function must be finished, while the whole program must be successfully run.
2. Paper: May 3. Must clarify all details.