Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра информатики

ОТЧЕТ

по лабораторной работе №11

**Реализация криптографических атак с помощью машинного обучения на физически неклонируемые функции**

|  |  |  |
| --- | --- | --- |
| Студент |  | И.Ю. Фалько |
| Преподаватель |  | М. В. Стержанов |

Минск 2019

СОДЕРЖАНИЕ

[Условие лабораторной работы 3](#_Toc25788657)

[Ход выполнения 3](#_Toc25788658)

[Заключение 9](#_Toc25788659)

# Условие лабораторной работы

Физически неклонируемые функции (ФНФ) часто используются в качестве криптографических примитивов при реализации протоколов аутентификации.

Рассмотрим простейший из них, основанный на на запросах и ответах (challenge response). Схема данного типа протокола приведена на рис. 1.

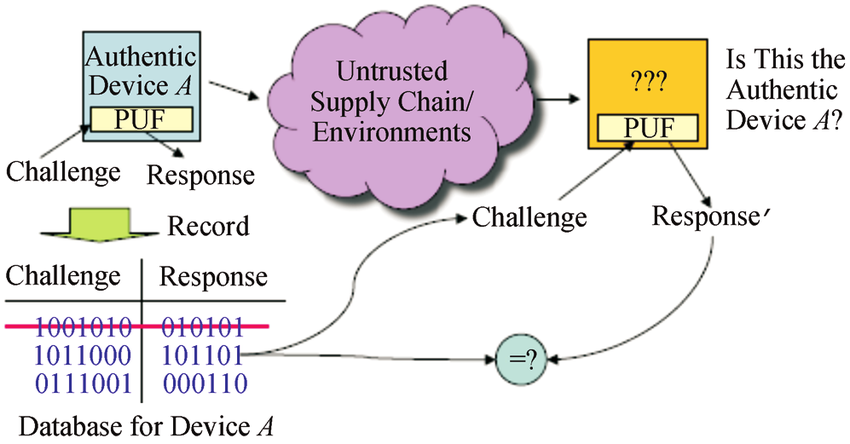


Рисунок 1. Протокол аутентификации, основанный на ФНФ.

В данном случае устройство А, содержащее реализацию ФНФ, может быть аутентифицировано с помощью набора запросов (challenge) и проверки ответов на них (response). При этом использованные пары запрос-ответ удаляются из базы данных устройства.

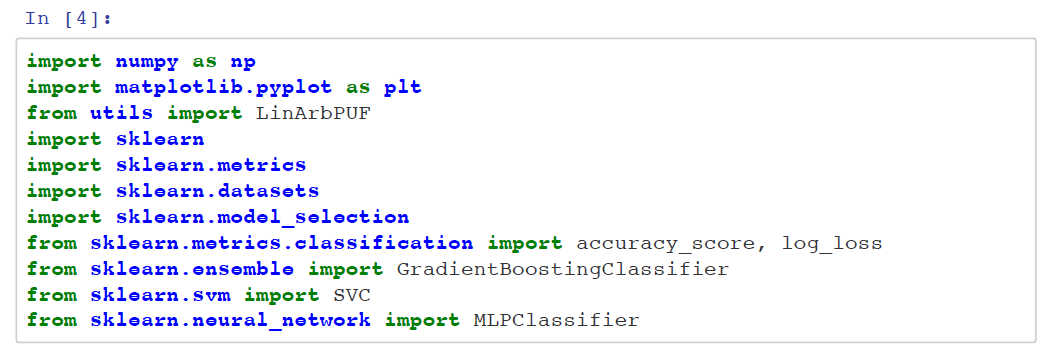
# Ход выполнения

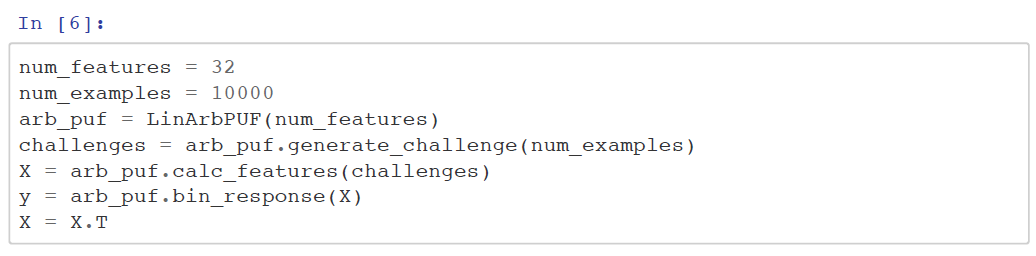
1. Сформулируйте задачу в терминах машинного обучения.

Поскольку ФНФ задается парами значений входных и соответствующим им выходных значений,которые называются парой запрос-ответ (Challenge-Response Pairs, CRP), то в простейшем случае ФНФ можно рассматривать как функцию, которая преобразует C(i) в ответы R(i): R(i) = PUF(C(i))

Т.е. задачей машинного обучения будет задача найти максимально точное приближение данной функции, если нам даны точные параметры запроса и другие пары запрос-ответ. Как входные данные используем набор пар запрос ответ (CRP), запрос представляет из себя массив бинарных значений (0 или 1) и соответствующий ему ответ бинарное значение 0 или 1. Т.е. получаем задачу классификации, где на вход получаем массив бинарных данных, где каждое значение это отдельный признак и 2 класса для классификации. Для начала рассмотрим набор простейших ФНФ - Arbiter. В физических устройствах они реализуются так, использую значения запроса строится 2 пути одинакой длины на схеме устройста, после чего идет сигнал по обоим путям, при установлении каждого бита есть аппартная задержка уникальная для устройста и в итоге выбирается путь, который завершился быстрее, т.е. на выход получаем 0 или 1 в зависимости от результата. Сгенерируем данные для данного вида ФНФ, массив задержек сгенерируем используя стандартное нормальное распределение, а набор запросов выберем из всех возможных запросов использую равномерное распределение. Будем использовать размер входного запроса 32 бит.

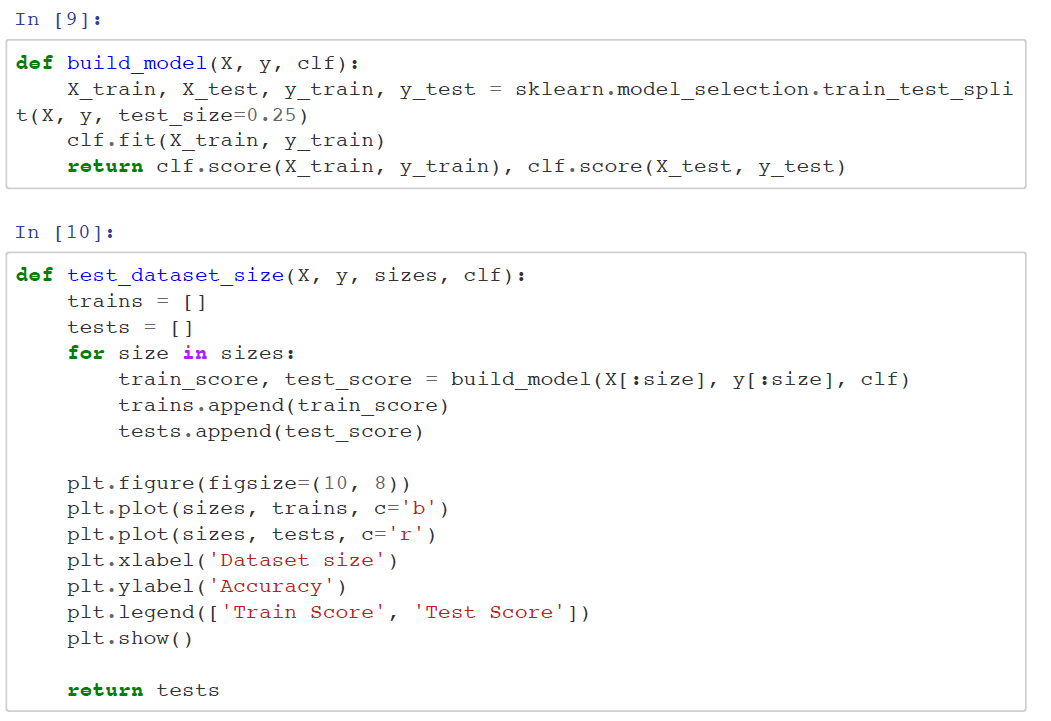
1. Обучите модель, которая могла бы предсказывать ответы по запросам, которых нет в обучающей выборке.





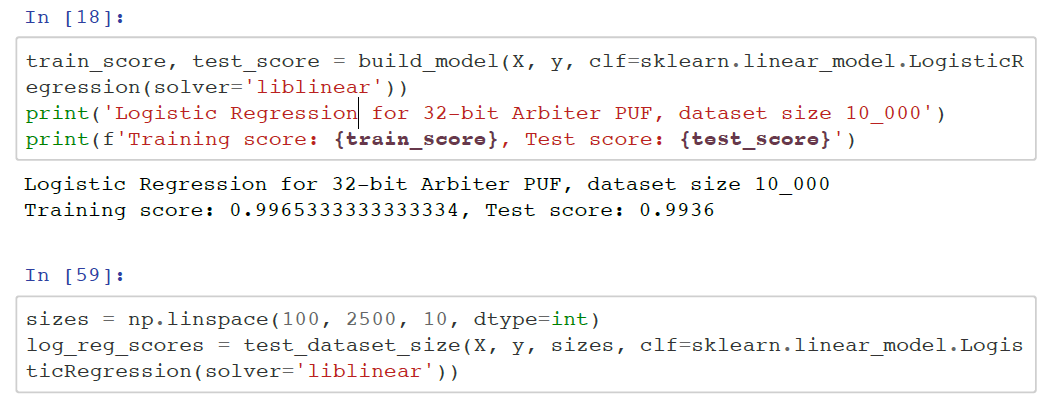
Для обучения модели попробуем использовать различные алгоритмы: Логистическую регрессию, градиентный бустинг, метод опорных векторов, нейронную сеть. Поскольку в данной задаче нам важно только правильно предсказать класс для входных данных и не важно какой именно это класс, то для оценки качества будем исползовать долю правильных ответов (Accuracy).

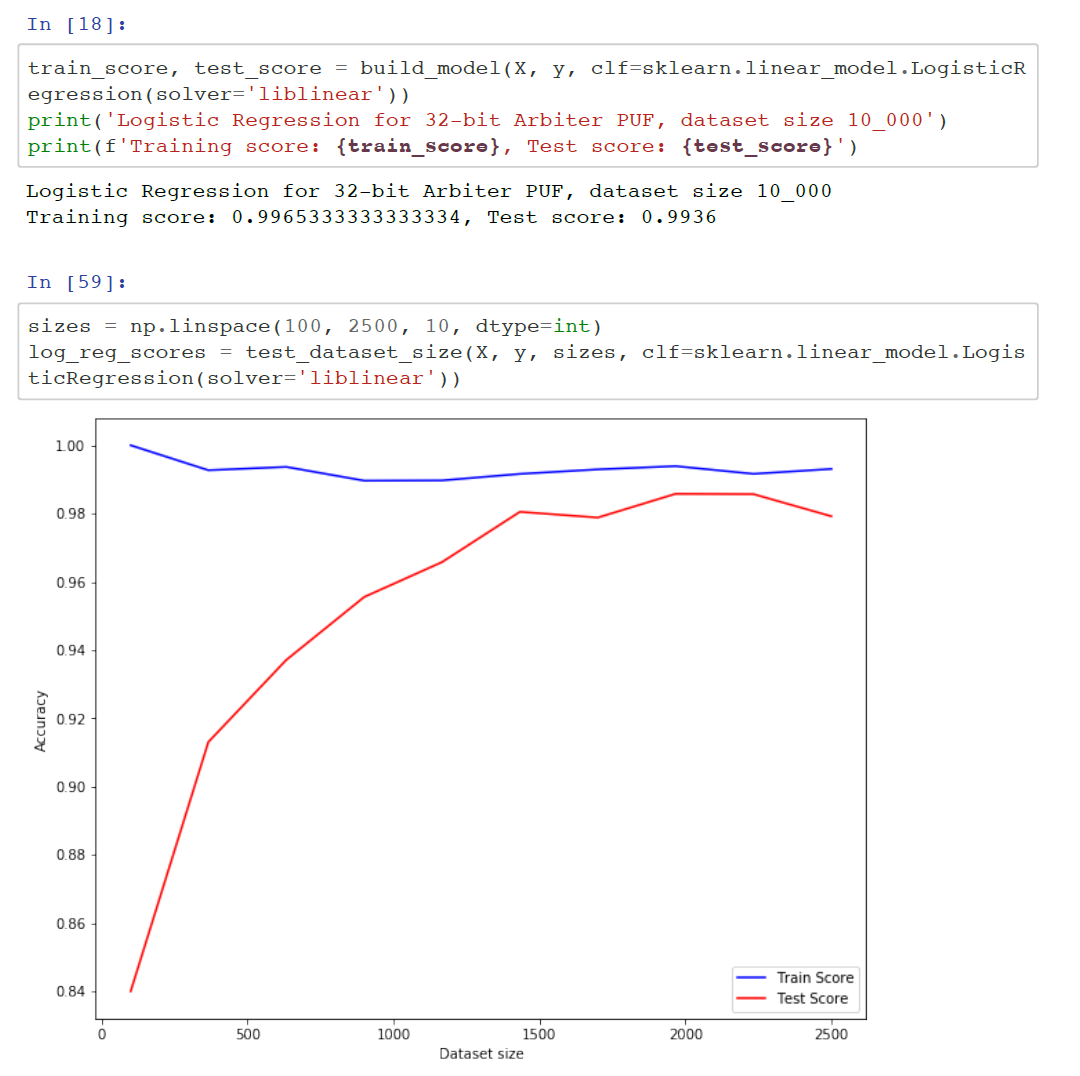
Реализуем функцию, которая будет по заданным параметрам и входным данным стоить одель и считать результаты



1. Применить как минимум 3 различных алгоритма (например, метод опорных векторов, логистическая регрессия и градиентный бустинг).

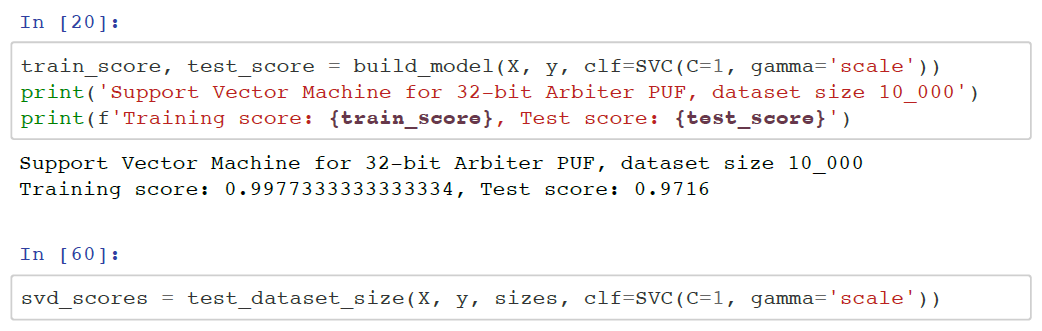
Проверим как справляется с задачей Логистическая регрессия:

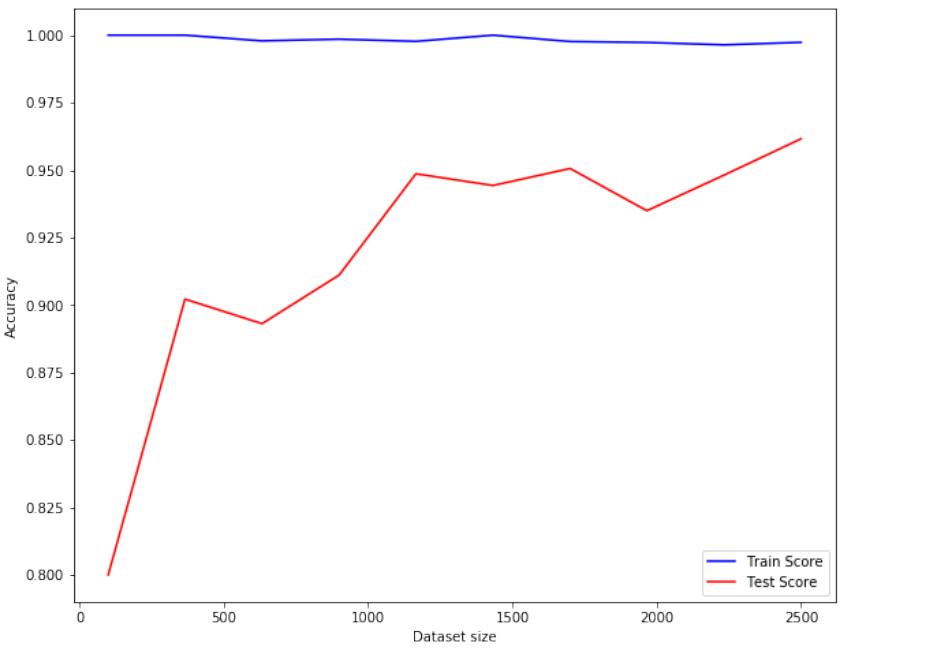




Как видно из результатов мы достигли довольно высокой точности работы на большой выборке, больше 99%. На графике видна зависимость точности предсказаний от размера данных. Точности в 95% получилось достичь когда размер набора данных приблизился к тысяче.

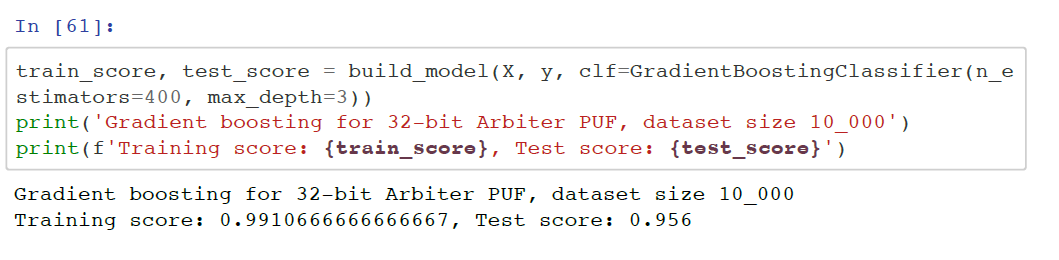
Проверим как справляется с задачей метод опорных векторов:

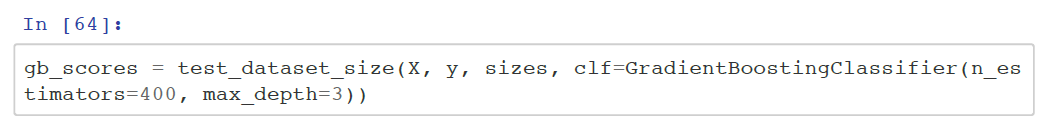




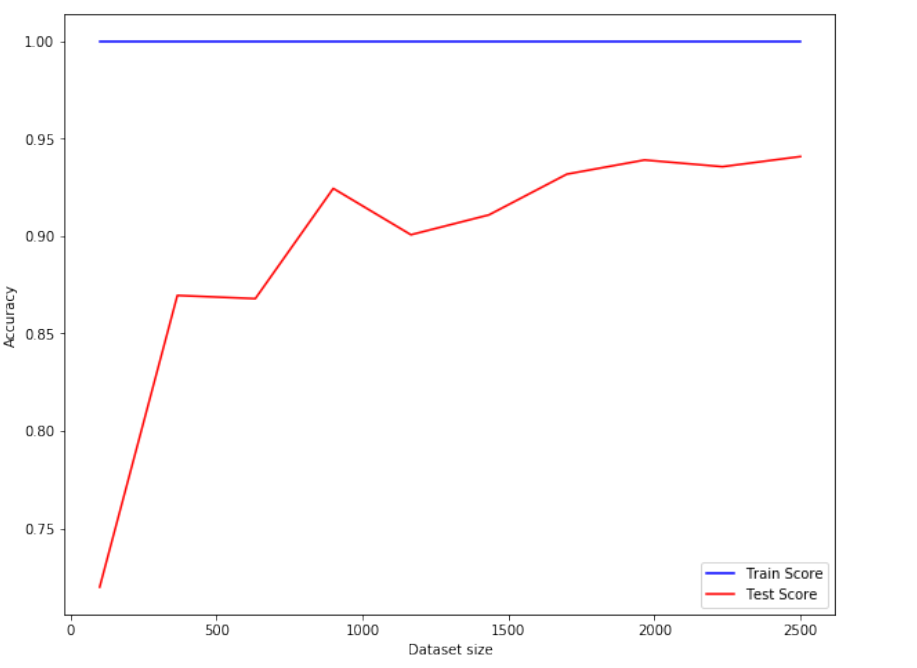
Как видим метод опорных векторов справился с задачей немного хуже чем логистическая регрессия, хотя 95% точности было достигнутся на таком же размере набора данных, около 1000.

Проверим как справляется с задачей метод градиентного бустинга:





Используя метод градиентного бустинга основанных на решающих деревьях, получилось добится неплохи результатов используя большое количество отдельных решаюших деревьев и их небольшую глубину.



1. Какая метрика наиболее подходит для оценки качества алгоритма?

Видно что для простых Arbiter ФНФ хорошо справляется логистическая регрессия, она работает стабильней, хотя стоит заметить судя по графику в итоге все кривые стресятся приблизительно в одну точку.

1. Какой наибольшей доли правильных ответов (Accuracy) удалось достичь?

99%

1. Какой размер обучающей выборки необходим, чтобы достигнуть доли правильных ответов минимум 0.95?

Около 1000.

1. Как зависит доля правильных ответов от *N*?

Чем больше N, тем более точные ответы

# ЗАКЛЮЧЕНИЕ

В данной работе было исследована возможность криптографических атак на физически неклонируемые функции, используя алгоритмы машинного обучения, в качестве алгоритмов были выбраны: логистическая регрессия, градиентный бустинг, метод опорных векторов. Приведено сравнения результатов работы алгоритмов на различные виды ФНФ различных размеров входных запросов.