Министерство образования Республики Беларусь

Учреждение образования

“Белорусский государственный университет информатики и радиоэлектроники”

Факультет компьютерных систем и сетей

Специальность “Обработка больших объемов информации”

Лабораторная работа №11

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

Выполнил:

магистрант гр. 858641 Кальман В.А.

Проверил:

Стержанов М. В.

Минск 2019

**1. Задание**

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

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

3. Какая метрика наиболее подходит для оценки качества алгоритма?

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

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

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

7. Ответы на вопросы представьте в виде графиков.

8. Оформите отчет.

**2. Инструменты**

Для решения поставленной задачи использовался Python и библиотека для машинного обучения sklearn, а также matplotlib для отрисовки графиков.

**3. Алгоритмы**

Для решения поставленной задачи были выбраны 3 алгоритма логистическая регрессия, многослойный перцептрон и Ada-бустинг. Данные три алгоритма используются для классификации, выборки которой наша задача и является.

**4. Ход работы**

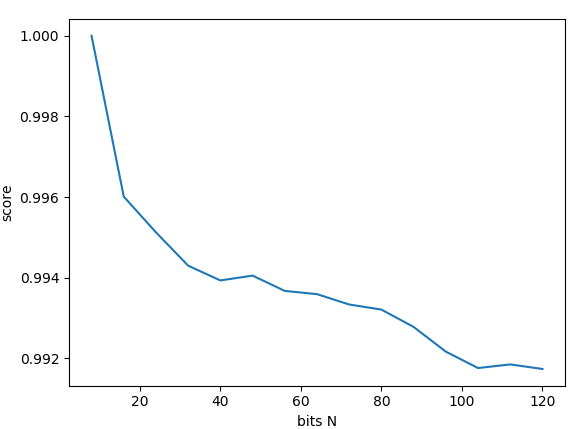
Для того чтобы решить задачу, сначала нужно разобраться, с исходными данными. И что в этих данных влияет на результат. Устройство пуфа типа арбитр разбиралось в первой лабораторной работе, было установлено что при 1 сигналы скрещиваются, а при 0 идут параллельно, именно это попробуем взять как за метрику, нам будет важно изменилось ли направление или нет.

Сырые данные для лабораторной работы были предварительно преобразованы в формат csv, для более удобного последующего использования.

Для обучения моделей и их проверок будет использоваться 2 выборки из одного датасета, одна для обучения другая для валидации.

1. Логистическая регрессия

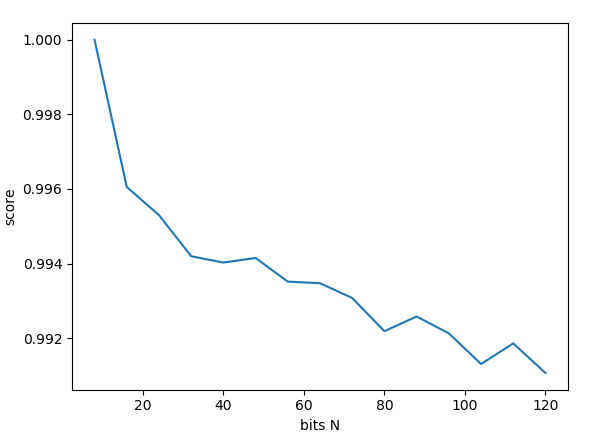
Основная идея логистической регрессии заключается в том, что пространство исходных значений может быть разделено линейной границей (т.е. прямой) на две соответствующих классам области. В случае двух измерений — это просто прямая линия без изгибов. В случае трех — плоскость, и так далее. В отличие от обычной регрессии, в методе логистической регрессии не производится предсказание значения числовой переменной исходя из выборки исходных значений. Вместо этого, значением функции является вероятность того, что данное исходное значение принадлежит к определенному классу.



Как видно из графика модель логистической регрессии очень хорошо обучилась на этих данных, с точностью от 1 до 0.992, что говорит о очень хорошем результате. При работе с 8 битами, получилась 1, это может означать то, что данных было более чем достаточно. Ну и объяснить это достаточно просто потому что данных для того чтобы обучить данную модель на 100% нужно 2^n данных где n количество бит, это такое количество перестановок возможно в данном пуфе, а данных на вход подавалось намного больше, что и помогло добиться данного результата, а также получается раз данных намного больше чем количество перестановок, то в тестовых данных было также больше количество данных из обучающей выборки

1. Многослойный перцептрон

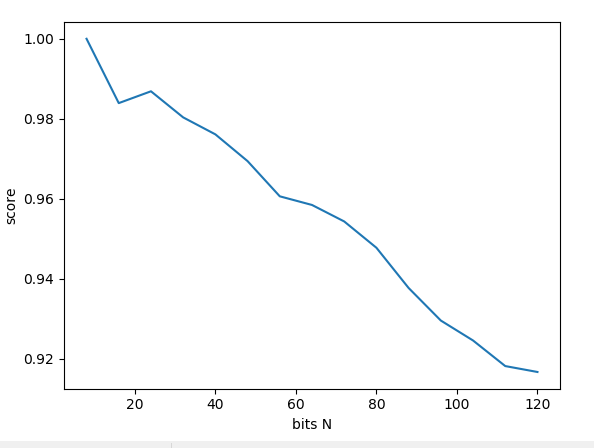
Многослойный перцептрон — частный случай перцептрона, в котором один алгоритм обратного распространения ошибки обучает все слои. Особенностью является наличие более чем одного обучаемого слоя (как правило — два или три). Используется, когда выходные параметры нелинейно зависят от входа.



Как видно из графика многослойный перцептрон достаточно хорошо обучился на основе входных данных. Однако при увеличении размерности ошибка увеличивается сильнее чем в методе логистической регрессии. Значит недостаточно данных для построения модели большой размерности.

1. Ada-бустинг

Общий подход заключался в жадном построении линейной комбинации простых моделей (базовых алгоритмов) путем перевзвешивания входных данных. Каждая последующая модель (дерево решений) строилась таким образом, чтобы придавать больший вес и предпочтение ранее некорректно предсказанным наблюдениям.



На основе графика можно сделать вывод, что Ada-бустинг сходится хуже всего по сравнению с остальными алгоритмами. Данная модель хорошо обучалась только на небольших наборах данных.

**Лучшая метрика**

Ошибка – отклонение ожидаемого результата от того, что отдала модель. Важно, так же, чтобы алгоритм после обучения умел работать с данными, не входящими в выборку на котором производилось обучение, в таком случае мы можем говорить о том, что модель поняла, как преобразовывать входные данные чтобы получить выходные данные. Скорость работы является одним из условий для выбора того или иного алгоритма в мо. Алгоритмы, которые работают быстрее могут давать большие ошибки.

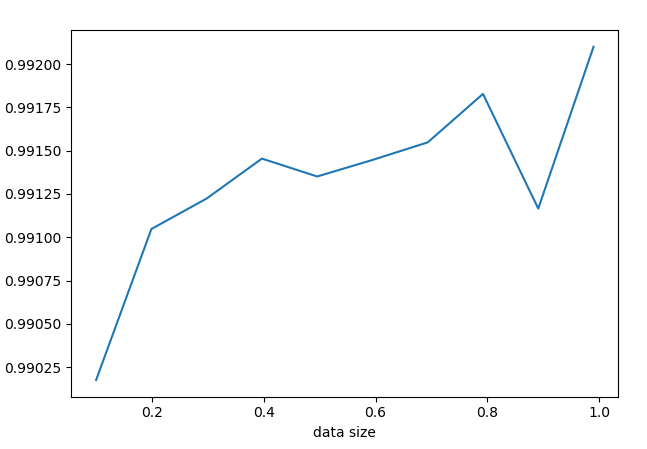
**Лучшая полученная точность**

На основе графиков можно сделать вывод, что чем больше данных подается для обучения модели, тем меньше ошибка, следовательно, модель более точно моделирует процесс, который решается ею. Точность варьируется и зависит от битности пуфа. Точность для 8-битного пуфа достигла 1, но для остальных нет, т.к. недостаточно данных для такого уровня точности.

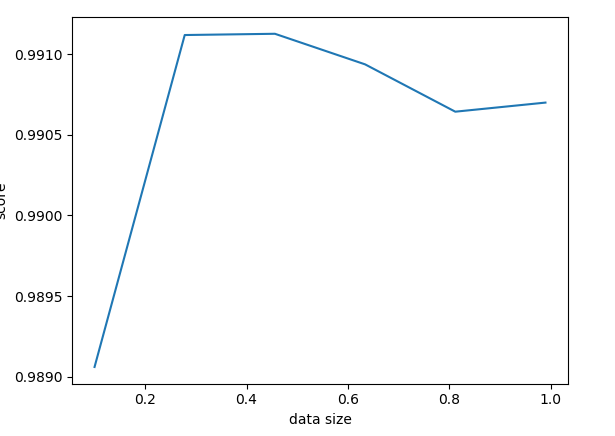
**Размер обучающей выборки для точности 0.95**

Размеры вычислялись на основе 128 битного датасета.

1. LR



1. MLP



1. Ada-boost

**Доля правильных ответов от N**

На основе графиков можно сделать вывод, что при меньшем количестве битов в пуфе модель обучалась лучше, так как меньше размерность и больше данных на вход, тем самым мы обучаем модель на очень большом множестве данных, которое либо полное или почти полное.

**Вывод**

Пуф типа арбитр - не надежен, и поддается построению математической модели на основе входных данных. Однако скорость ее построения и точность падает при увеличении разрядности.

Для линейной классификации ответов пуфов лучше всего подошел алгоритм линейной регрессии. Данный алгоритм хорошо себя показывает на разных объемах данных.

Выбор алгоритма и настройка зависит от типа решаемых задач.

Исходные данные перед обучением должны быть максимально упрощены.