

# Аннотация

Данная работа посвящена исследованию методов машинного обучения в задаче радиометрической идентификации.

Задача радиометрической идентификации заключается в распознавании устройств на основании априори известных принятых сигналов. Во всех версиях стандарта Wi-Fi 802.11 для обратной совместимости используются унаследованные преамбулы, определяемые стандартом. На основании этих преамбул можно составить признаковое описание, в котором каждое устройство будет иметь свой RF-fingerprint. Классификация устройств по этому отпечатку позволит обеспечить защиту от сетевых атак, например Man In The Middle, на физическом уровне. Ввиду того, что распознавание ведется на физическом уровне, практически невозможно подделать RF-fingerprint. Поэтому такие методы обеспечивают большую степень защиты, чем протоколы шифрования.

В рамках данной работы производится выбор эффективного признакового описания и алгоритма классификации в режиме обучения с учителем, а также разработка и анализ алгоритма онлайн-классификации.

В ходе решения этих задач было выявлено: наиболее эффективное признаковое описание получается, если преобразовать преамбулы, пользуясь взятием модуля комплексного числа, быстрым преобразованием Фурье и последовательным вычитанием частей спектра, а затем применить автоэнкодер; наилучший результат классификации показал MLP, который также удовлетворил всем поставленным требованиям. Был создан алгоритм, решающий задачу радиометрической идентификации в онлайн-режиме.

# Содержание

Аннотация	iii
1. Введение	1
2. Обзор предметной области	3
2.1. Обозначения и сокращения	3
2.2. Формальная запись проблемы	4
2.3. Постановка задачи	6
3. Выбор метода решения	7
3.1. Подходы к решению задачи	7
3.2. Формулировка требований к решению	8
3.3. Методы решения	9
4. Описание метода решения	11
4.1. Математический аппарат	11
4.2. Модель данных	16
5. Исследование решения	17
5.1. Описание эксперимента	17
5.2. Результаты	22
6. Заключение	
	25
Список литературы	27

# Глава 1

## Введение

### Актуальность проблемы

При столь быстром развитии технологий передачи сигналов появляется необходимость также и в развитии технологий детектирования устройств. Существуют методы, используемые для распознавания устройств на высших уровнях, но потребность в достаточном уровне защиты и точности все еще актуальна.

### Цель работы

Для обеспечения защиты информации необходимо с высокой точностью распознавать устройства.

### Задачи

- Выбор эффективного признакового описания
- Выбор алгоритмов классификации в режиме обучения с учителем
- Разработка и реализация алгоритма онлайн-классификации

То есть, разработка алгоритма, который по сигналу с передатчика сможет идентифицировать передающее устройство.

### Объект исследования

Задача радиометрической идентификации – выделение уникального набора шумов для использования их в качестве идентификатора устройства.

### Предмет исследования

RF fingerprint – особая форма высокочастотного сигнала, которая зависит от конкретного передающего устройства. Данный сигнал представляет из себя шум, обусловленный внутренним устройством передатчика. На практике это сигнал, полученный путем анализа преамбул принятого сигнала.

**Практическая значимость**

Радиометрическая идентификация, в отличие от других методов защиты, использует для распознавания именно самый низкий, физический уровень, поэтому обеспечивает большую безопасность, чем алгоритмы распознавания, которые осуществляются на более высоких уровнях стека сетевых протоколов OSI. RF fingerprint – уникальный для каждого устройства набор шумов, его практически невозможно подделать, а значит он может быть использован для предотвращения различных сетевых атак, например, MITM.

# Глава 2

## Обзор предметной области

### 2.1 Обозначения и сокращения

**RF** – Radio Frequency

**RF-fingerprint** – Radio Frequency Fingerprint

**MITM** – Man In The Middle

**OFDM** – Orthogonal Frequency-Division Multiplexing

**AE** – Autoencoder

**RAW** – Сырой, необработанный [сигнал]

**ML** – Machine Learning

**DT** – Decision Tree

**TSNE (t-SNE)** – t-distributed Stochastic Neighbor Embedding

**OSI** – The Open Systems Interconnection model

**AGS** – Automatic Gain Control

**DL** – Deep Learning

## 2.2 Формальная запись проблемы

Сформулировать проблему можно так – необходимо с высокой точностью распознавать на физическом уровне устройства, сигнал которых мы уже принимали.

Идентификация должна происходить вне зависимости от их конкретного расположения в пространстве, условий среды и зашумленности сигнала, а также с минимумом известных данных о конкретном устройстве.

Ввиду своих физических свойств, – в случае OFDM, из-за преобразователей, усилителей и фильтров в схеме, – каждое передающее устройство уникально искажает сигнал.

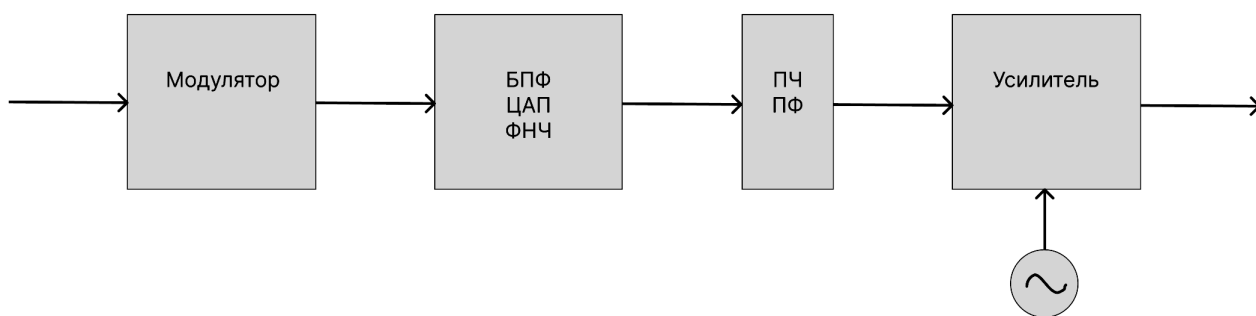


Рис. 1. Схема формирования OFDM-сигнала

Для того, чтобы обнаружить внесенные искажения, необходимо выделить постоянные составляющие сигнала – преамбулы, после чего обработать их и решить задачу классификации на полученных данных.

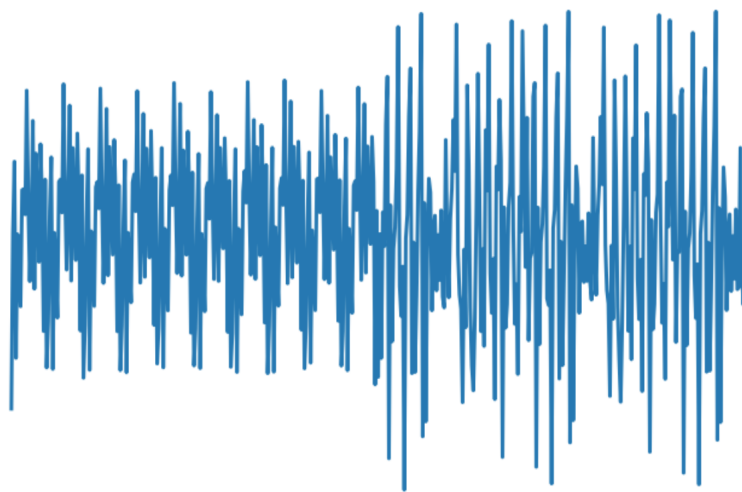


Рис. 2. Legacy преамбула

В сигнале на Рис. 2 можно увидеть два поля legacy преамбулы: первая состоит из 10 одинаковых частей, а вторая из 2,5 одинаковых частей, с циклически расположенной начальной частью.



Рис. 3. L-STF

L-STF – the Legacy short training field – это первое поле 802.11 OFDM PLCP legacy преамбулы.

Длительность L-STF зависит от полосы частот канала и составляет 8, 16 и 32  $\mu s$  для частот 20, 10 и 5 MHz соответственно.

L-STF обладает хорошими корреляционными свойствами, поэтому используется для детекции начала пакетов, грубой частотной коррекции и установки AGC.

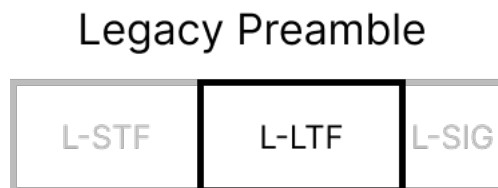


Рис. 4. L-LTF

L-LTF – the legacy long training field – второе поле 802.11 OFDM PLCP legacy преамбулы.

Длительность L-LTF также зависит от полосы частот канала и составляет 8, 16 и 32  $\mu s$  для частот 20, 10 и 5 MHz соответственно.

L-LTF используется для оценки канала, точной оценки частотного смещения и точной оценки смещения символьной скорости.

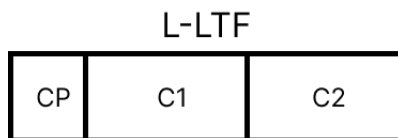


Рис. 5. Устройство L-LTF

L-LTF состоит из циклического префикса (CP) и двух следующих друг за другом идентичных символов C1 и C2. CP имеет длительность в 2 раза меньшую, чем C1 и C2 и в точности повторяет вторую половину последнего во временной области.

При обработке сигнала для распознавания конкретного устройства в задаче радиометрической идентификации необходимо не только выделить особенности преамбул, но и избавиться от зашумленности сигнала, внесенной условиями среды. Именно поэтому при обработке RAW сигнала необходимо отобрать значимые для классификации признаки.

## 2.3 Постановка задачи

По RAW сигналам необходимо, пользуясь методами машинного обучения, решить задачу классификации – распознавания каждого устройства, для которого ранее собиралась информация.



# Глава 3

## Выбор метода решения

### 3.1 Подходы к решению задачи

Для решения данной задачи предлагается действовать по данному алгоритму:

1. Предобработка данных
2. Отбор признаков
3. Решение задачи классификации
4. Адаптация классификатора

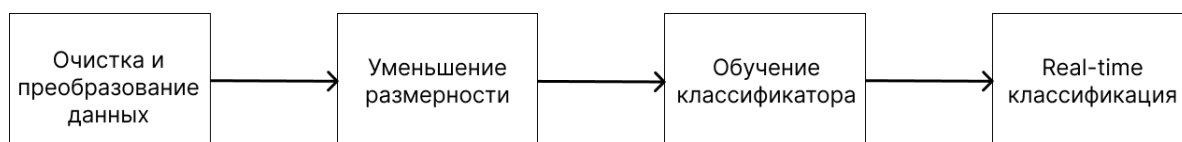


Рис. 6. Алгоритм решения

## 3.2 Формулировка требований к решению

### Необходимые свойства решения:

- Оптимальность по времени выполнения
- Реальные требования к вычислительной мощности обрабатывающего устройства
- Масштабируемость композиции алгоритмов
- Отсутствие вычислений для нового образца по всем данным
- Возможность объединения алгоритмов для потоковой обработки сигналов
- Возможность дообучения на неразмеченных данных
- Сохранение модели для дальнейшего использования
- Возможность классификации без хранения обработанных ранее данных

### 3.3 Методы решения

#### Выбор алгоритма для преобразования RAW сигнала

Есть несколько вариантов преобразований RAW сигнала, а также множество их комбинаций.

В данной работе рассмотрены:

- FFT – Fast Fourier transform – алгоритм ускоренного вычисления дискретного преобразования Фурье.  
Данный алгоритм, в отличие от прямого дискретного преобразования Фурье, выполняющегося за  $O(N^2)$ , имеет асимптотику  $O(N\log(N))$ .
- Разложение сигнала на модуль и аргумент.
- Разложение на действительную и мнимую части.

#### Выбор алгоритма для уменьшения размерности

Для данной задачи необходимо значительно уменьшить размерность признакового пространства, по минимуму потеряв важную информацию об RF-Fingerprint устройства.

Может понадобиться не просто убрать менее значимые признаки, а создавать композиции из имеющихся признаков для минимальных потерь. Этой способностью обладают автоэнкодеры на основе нейронных сетей.

В данной работе рассмотрены:

- Алгоритм проекции на 2-мерное признаковое пространство – TSNE
- Отбор признаков с помощью алгоритма Random Forest
- Встроенный автоэнкодер MATLAB
- Автоэнкодер, построенный с помощью библиотеки tensorflow.keras
- Permutation importance из библиотеки sklearn.inspection

### Выбор алгоритма для классификации

Для данной задачи необходимо достаточно точно классифицировать зашумленные данные, обладающие признаками со сложной внутренней зависимостью.

Алгоритмы классификации, которые подходят для решения данной задачи должны быть быстродействующими, нетривиальными и помехоустойчивыми.

В данной работе рассмотрены:

- Decision Tree – Решающее дерево – алгоритм, основанный на разделении по значению признаков для построения дерева
- Random Forest – Случайный лес – композиция решающих деревьев, более устойчивая к переобучению
- Multilayer Perceptron – полносвязная нейронная сеть построенная с помощью библиотеки `sklearn.sknn`
- CatBoostClassifier – open source классификатор от компании Yandex

# Глава 4

## Описание метода решения

### 4.1 Математический аппарат

#### Метрики

В работе используются следующие метрики:

1. MultiClass:

$$\frac{\sum_{i=1}^N w_i \log \left( \frac{e^{a_{it_i}}}{\sum_{j=0}^{M-1} e^{a_{ij}}} \right)}{\sum_{i=1}^N w_i}$$
$$t \in \{0, \dots, M-1\}$$

2. MultiClassOneVsAll:

$$\frac{\frac{1}{M} \sum_{i=1}^N w_i \sum_{j=0}^{M-1} [j = t_i] \log(p_{ij}) + [j \neq t_i] \log(1 - p_{ij})}{\sum_{i=1}^N w_i}$$
$$t \in \{0, \dots, M-1\}$$

3. Total F1 weighted:

$$\frac{\sum_{i=1}^M w_i F1_i}{\sum_{i=1}^M w_i}$$

4. MCC:

$$\frac{\sum_k \sum_l \sum_m C_{kk} C_{lm} - C_{kl} C_{mk}}{\sqrt{\sum_k (\sum_l C_{kl}) \left( \sum_{k' | k' \neq k} \sum_{l'} C_{k'l'} \right)}} \sqrt{\sum_k (\sum_l C_{lk}) \left( \sum_{k' | k' \neq k} \sum_{l'} C_{l'k'} \right)}$$

5. Accuracy:

$$\frac{\sum_{i=1}^N w_i [\operatorname{argmax}_{j=0,\dots,M-1} (a_{ij}) == t_i]}{\sum_{i=1}^N w_i}$$

$$t \in \{0, \dots, M-1\}$$

6. Hinge loss:

$$\ell(y) = \max(0, 1 - t \cdot y)$$

7. Hamming loss:

$$\sum_{i=1}^N w_i [\operatorname{argmax}_{j=0,\dots,M-1} (a_{ij}) \neq t_i]$$

$$\sum_{i=1}^N w_i$$

8. Zero One loss:

$$1 - \text{Accuracy}$$

9. Кappa:

$$1 - \frac{1 - \text{Accuracy}}{1 - \text{RAccuracy}}$$

$$\text{RAccuracy} = \frac{\sum_{k=0}^{M-1} n_{k_a} n_{k_t}}{\left(\sum_{i=1}^N w_i\right)^2}$$

10. WKappa:

$$\kappa_w = \frac{p_o - p_e}{1 - p_e}$$

где  $p_o = \sum_{i=1}^K \sum_{j=1}^K w_{ij} p_{ij}$  и  $p_e = \sum_{i=1}^K \sum_{j=1}^K w_{ij} p_{i,p} p_{j,p}$  при  $0 \leq w_{ij} \leq 1$  и  $w_{jj} = 1$  ( $i, j = 1, \dots, K$ ), или

$$\kappa_w = 1 - \frac{q_o}{q_e}$$

где  $q_o = \sum_{i=1}^K \sum_{j=1}^K v_{ij} p_{ij}$  и  $q_e = \sum_{i=1}^K \sum_{j=1}^K v_{ij} p_{i,p} p_{j,p}$  при  $0 \leq v_{ij} \leq 1$  и  $v_{jj} = 1$  ( $i, j = 1, \dots, K$ ).

11. AUC $_{\mu}$ :

$$\text{AUC} = \frac{1}{n_+ n_-} \sum_{\hat{p}^{(i)} \in D^+} \sum_{\hat{p}^{(j)} \in D^-} \tilde{I}(\hat{p}^{(i)} - \hat{p}^{(j)})$$

$$S(i, j) = \frac{1}{n_i n_j} \sum_{a \in D^i, b \in D^j} \tilde{I} \circ O(\mathbf{y}^{(a)}, \mathbf{y}^{(b)}, \hat{\mathbf{p}}^{(a)}, \hat{\mathbf{p}}^{(b)}, \mathbf{v}_{i,j})$$

$$\text{AUC}_{\mu} = \frac{2}{K(K-1)} \sum_{i < j} S(i, j)$$

## Метрический классификатор

Для произвольного объекта  $a \in X$  расположим элементы обучающей выборки  $x_1, \dots, x_\ell$  в порядке возрастания расстояний до  $a$ :

$$\rho(a, x_a^{(1)}) \leq \rho(a, x_a^{(2)}) \leq \dots \leq \rho(a, x_a^{(\ell)})$$

где через  $x_a^{(i)}$  обозначается  $i$ -й сосед объекта  $a$ . Соответственно, ответ на  $i$ -м соседе объекта  $a$  есть  $y_a^{(i)} = y^*(x_a^{(i)})$ . Таким образом, любой объект  $a \in X$  порождает свою перенумерацию выборки.

*Определение.* Метрический алгоритм классификации с обучающей выборкой  $X^\ell$  относит объект  $x$  к тому классу  $y \in Y$ , для которого суммарный вес ближайших обучающих объектов  $\Gamma_y(x, X^\ell)$  максимален:

$$a(x; X^\ell) = \arg \max_{y \in Y} \Gamma_y(x, X^\ell); \quad \Gamma_y(x, X^\ell) = \sum_{i=1}^{\ell} [y_x^{(i)} = y] w(i, x),$$

где функция  $w(i, x)$  оценивает степень важности  $i$ -го соседа для классификации объекта  $x$ .

Функция  $\Gamma_y(x, X^\ell)$  называется оценкой близости объекта  $x$  к классу  $y$ . [8]

## Линейный классификатор

Пусть  $X$  - пространство объектов;  $Y = \{-1, 1\}$  - множество допустимых ответов; объекты описываются  $n$  числовыми признаками  $f_j : X \rightarrow \mathbb{R}, j = 1, \dots, n$ . Вектор  $x = (x^1, \dots, x^n) \in \mathbb{R}^n$ , где  $x^j = f_j(x)$ , называется признаковым описанием объекта  $x$ . Если дискриминантная функция определяется как скалярное произведение вектора  $x$  и вектора параметров  $w \in \mathbb{R}^n$ , то получается линейный классификатор: [8]

$$a(x, w) = \text{sign}(\langle w, x \rangle - w_0) = \text{sign} \left( \sum_{j=1}^n w_j f_j(x) - w_0 \right)$$

## Multilayer perceptron

Персептрон – линейный классификатор. Это математическая модель нейрона – нервной клетки мозга.

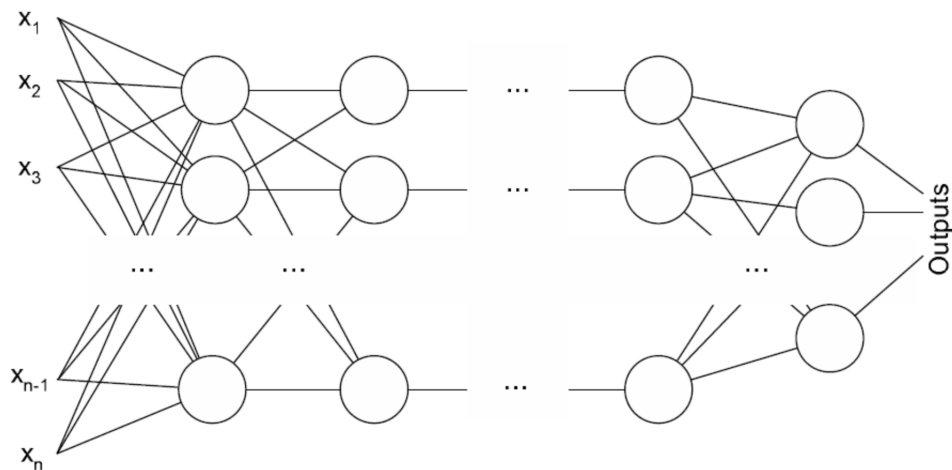
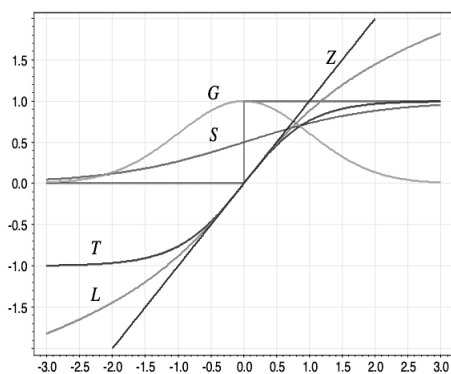


Рис. 7. Multilayer perceptron

Нейрон принимает каждый такт на свои  $n$  входов заряды величиной  $x_j = f_j(x)$ . Эти заряды умножаются на веса  $w_j$ , а затем берется их сумма. У возбуждающего синапса положительный вес, а у тормозящего – отрицательный.



$\theta(z) = [z \geq 0]$	пороговая функция Хевисайда;
$\sigma(z) = (1 + e^{-z})^{-1}$	сигмоидная функция (S);
$\text{th}(z) = 2\sigma(2z) - 1$	гиперболический тангенс (T);
$\ln(z + \sqrt{z^2 + 1})$	логарифмическая функция (L);
$\exp(-z^2/2)$	гауссовская функция (G);
$z$	линейная функция (Z);

Рис. 8. Активационные функции

Если суммарный заряд превышает порог активации  $w_0$ , то нейрон возбуждается и выдает 1, иначе -1.



## Autoencoder

Автоэнкодер – нейронная сеть, обучающаяся не на отображение в пространство классов, а на самоотображение.

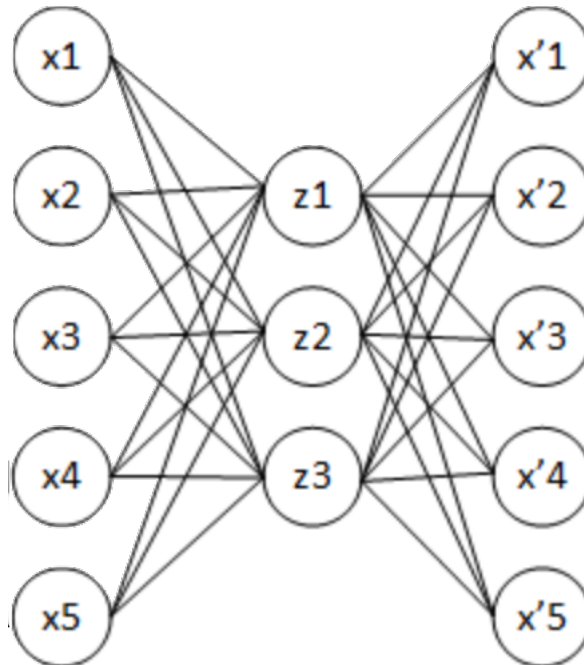


Рис. 9. Примитивный автоэнкодер

Обучение автоэнкодера происходит путем минимизации ошибки между входным и выходным слоем.

Промежуточные слои автоэнкодера, в отличие от персептрона, должны иметь меньшую размерность, чем входной и выходной слои, для того, чтобы исключить тривиальность решения.

## 4.2 Модель данных

Данные предоставлены в виде RAW сигналов преамбул с 5-ти различных устройств, расположенных в двух различных местах.

Эти устройства:

- Lenovo
- Moto 8
- Pixel
- Tab 4
- Zen

Хранятся данные в формате .mat файлов, в структуре данных dict.

Обращение к данным производится по полям "X" – для получения выборки, и "y" – для получения массива классов.

### Краткое описание датасета:

- Приемник – ADALM PLUTO
- Передатчики – смартфоны
- 5 несбалансированных классов
- 2 различных канала
- 17281 преамбула
- 480 признаков

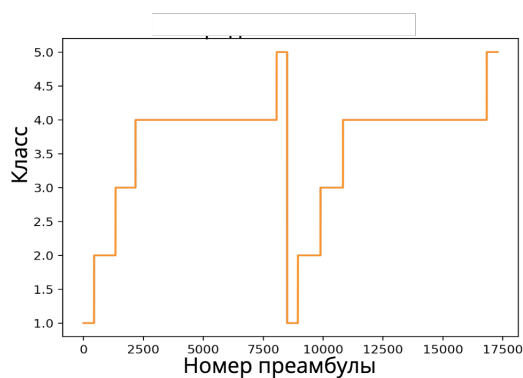


Рис. 10. График распределения классов

# Глава 5

## Исследование решения

### 5.1 Описание эксперимента

#### Предобработка данных

1. Выделение преамбул из RAW сигналов
2. Деление преамбул на поля
3. Параллельная обработка полей преамбул:
  - (a) FFT
  - (b) Модуль от преобразования Фурье
  - (c) Выделение значимых частей спектра
  - (d) Попарное вычитание частей спектра (последний элемент из первого и так далее)
4. Конкатенация полученных разностей в один датасет

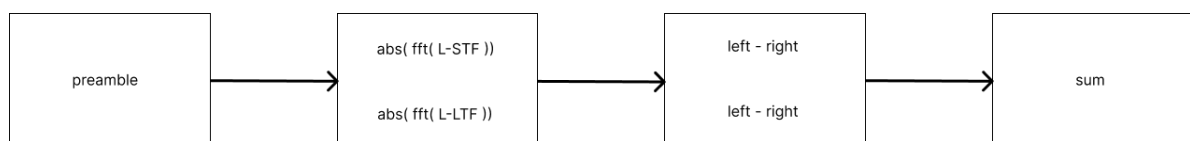


Рис. 11. Алгоритм предобработки данных

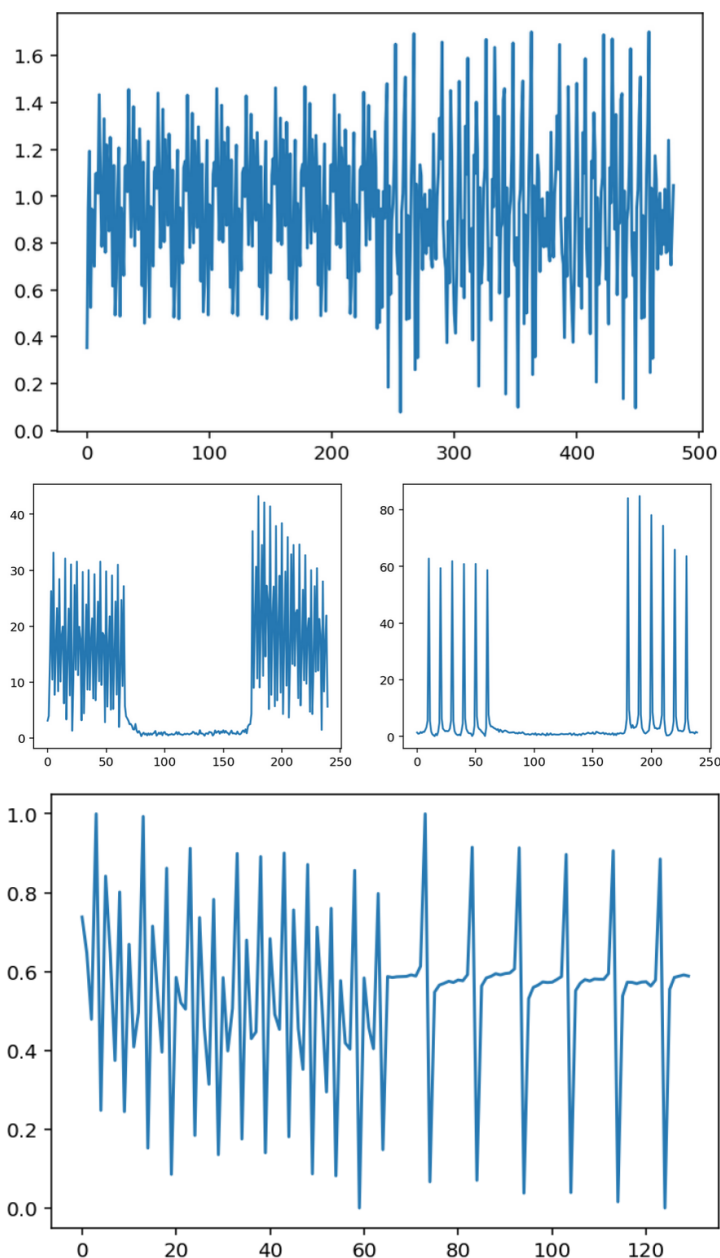


Рис. 12. Результаты обработки, соответствующие шагам 1, 2 и 4

### Кодирование/отбор признаков

#### 1. Отбор признаков с помощью Random Forest

- (a) Обучение Decision Tree, подбор гиперпараметров
- (b) Построение Random Forest на основе обученного дерева решений
- (c) Обучение Random Forest
- (d) Вызов метода `.feature_importances_`
- (e) Сортировка
- (f) Подбор оптимального количества признаков
- (g) Оценка результата

2. Уменьшение размерности признакового пространства с помощью встроенного в MATLAB автоэнкодера
3. Построение автоэнкодера на основе библиотеки Keras
  - (a) Выбор количества слоев и их размеров
  - (b) Выбор функций активации
  - (c) Подбор необходимой комбинации слоев и функций
  - (d) Оценка результата
4. TSNE — проекция выборки на двухмерное пространство для визуализации, а также получения новой информации о данных

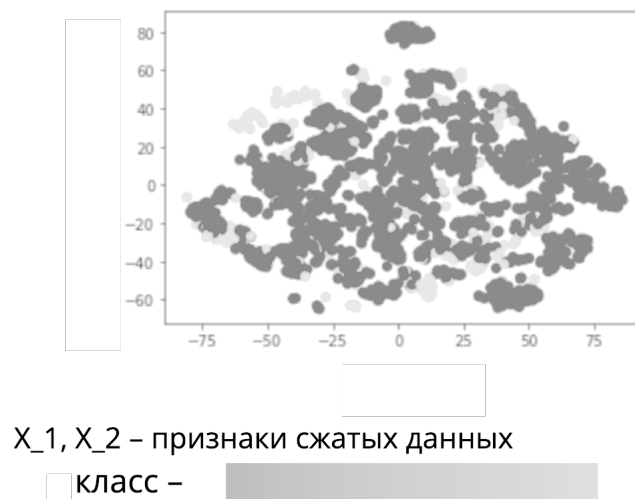


Рис. 13. TSNE

**Классификация**

1. Decision Tree
  - (a) Кросс-валидация
  - (b) Обучение
  - (c) Оценка результатов
2. Random Forest
  - (a) Кросс-валидация
  - (b) Обучение
  - (c) Оценка результатов
3. Multilayer Perceptron
  - (a) Кросс-валидация
  - (b) Обучение
  - (c) Оценка результатов
4. CatBoostClassifier
  - (a) Кросс-валидация
  - (b) Обучение
  - (c) Оценка результатов

**Адаптация алгоритма**

1. Алгоритм для адаптации – Multilayer Perceptron, как максимально подходящий требованиям к решению
2. Данные берутся по 300 отсчетов – batch
3. Обучение происходит постепенно – N epochs
4. После достижения локального взвешенного максимума подается только тестовая выборка

## Технические характеристики

- MacBook Pro 13 2015
- Процессор 2.7 GHz 2-ядерный Intel Core i5
- Память 8 ГБ 1867 MHz DDR3

## Программные средства

- Язык программирования Python 3.10
- Jupyter notebook
- Jupyter lab
- Atom + Terminal
- Graphix
- MATLAB
- Библиотеки:
  - Scikit-learn
  - Numpy
  - Pandas
  - Matplotlib
  - SciPy
  - TensorFlow
  - Keras
  - PyTorch
  - Seaborn
  - CatBoost
  - CatBoostClassifier

## 5.2 Результаты

### Выбор признакового описания

	Preamble					L-STF				L-LTF			
	real	imag	abs	angle	abs(fft)	real	imag	abs	abs(fft)	real	imag	abs	abs(fft)
Accuracy	0.732	0.733	0.842	0.702	0.828	0.734	0.731	0.838	0.785	0.725	0.728	0.836	0.822
F1-score	0.667	0.663	0.823	0.597	0.803	0.668	0.664	0.819	0.744	0.646	0.653	0.812	0.796

Таблица 1.

В Таблице 1 находятся результаты оценки Random Forest различных преобразований данных.

	L-STF + L-LTF	L-STF	L-LTF
Accuracy	0.85	0.83	0.84
F1-score	0.83	0.81	0.83

Таблица 2. Оценка для L-STF, L-LTF и их суммы

Таблица 2 – сравнительная таблица F1-score и Accuracy на классификаторе Random Forest для полей L-STF, L-LTF и их суммы, прошедших выбранную схему преобразования.

Очевидно, что L-LTF вносит больший вклад в решение задачи классификации, чем L-STF, но их сумма показывает большую точность.

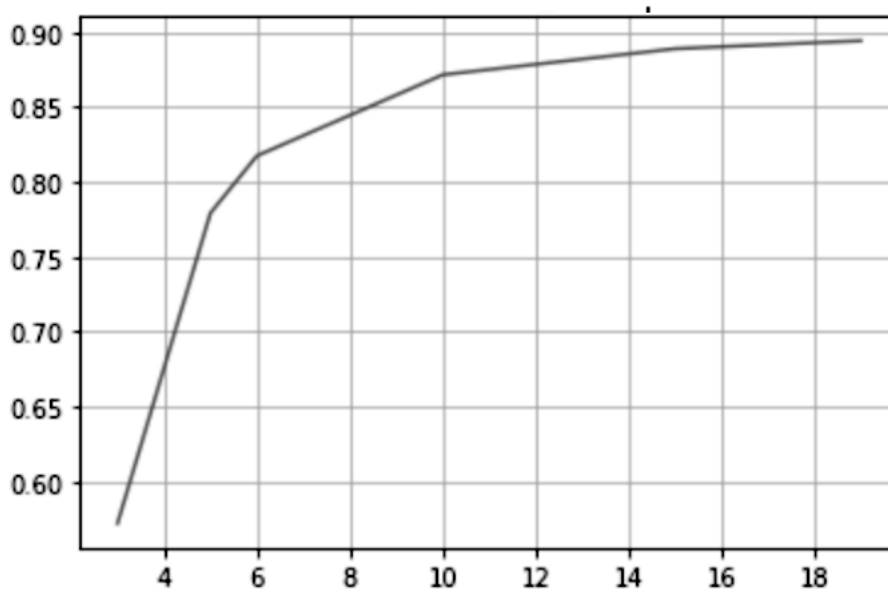


Рис. 14. Точность от количества признаков

На рис. 15 изображена зависимость Accuracy от количества признаков в выборке. Эти признаки отобраны по значимости, рассчитанной Random Forest. Можно заметить, что начиная с 10 признаков Accuracy меняется незначительно, то есть это значение оптимально.



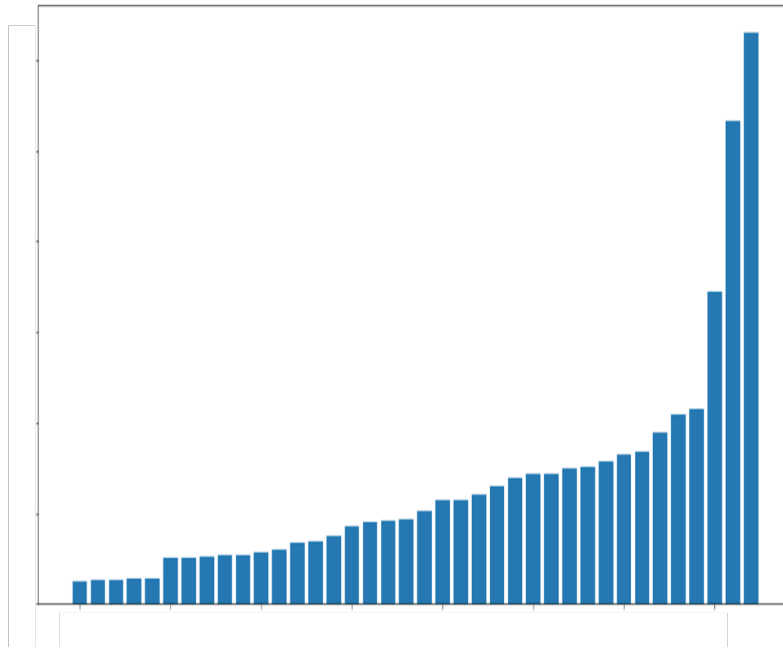


Рис. 15. Средний вклад в изменение энтропии от признака

На рис. 16 изображено распределение признаков выборки по их значимости, рассчитанной Random Forest.

Признаки расположены в порядке возрастания значимости – среднего вклада в изменение энтропии при построении дерева.

### Точность классификации

В таблице 3 указаны оценки классификаторов в различных конфигурациях.

	Accuracy	F1-score
Decision tree	0.81	0.73
Random forest	0.86	0.78
CatBoost Classifier	0.95	0.88
Autoencoder + Random forest	0.85	0.83
Autoencoder + MLP	0.92	0.91
Autoencoder + CatBoost	0.91	0.90

Таблица 3. Таблица классификаторов

Можно заметить, что самые примитивные классификаторы – DT и RF дают неплохой результат, но не полностью отвечают сформулированным требованиям. Лучшими комбинациями являются Autoencoder+MLP и Autoencoder+CatBoost.

## Сравнительный анализ

Рассмотрим подробнее классификаторы CatBoostClassifier и MLPClassifier.

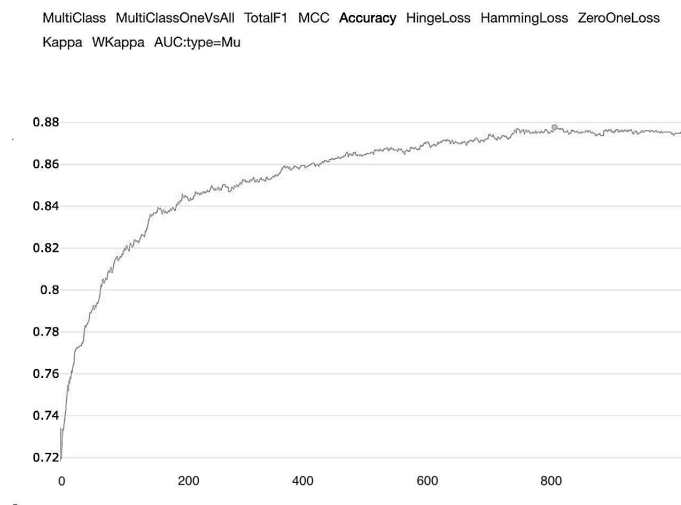


Рис. 16. CatBoostClassifier

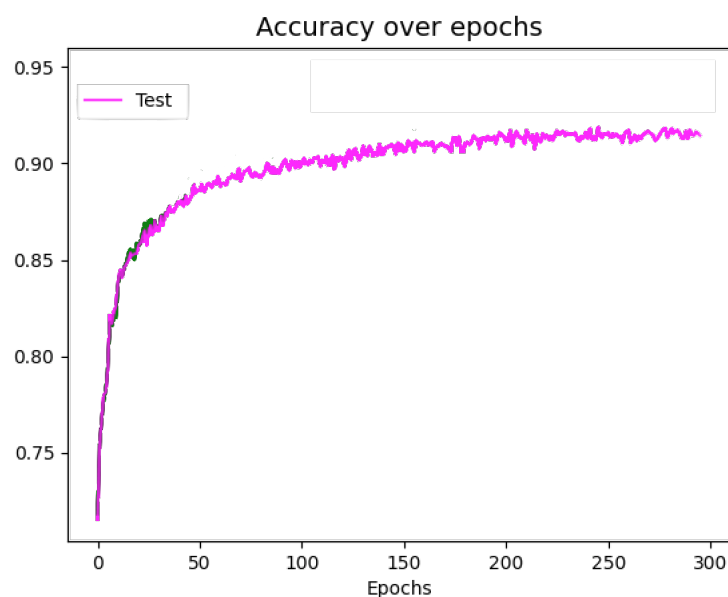


Рис. 17. Multilayer Perceptrone

Оба алгоритма сходятся, CatBoost – к 0.88, а MLP – к 0.91, при этом оба работают в режиме реального времени.

## Выводы

- Лучшее признаковое пространство получено преобразованиями+автоэнкодером
- Самую высокую точность показал многослойный персептрон
- Задача классификации решается композицией вышеназванных алгоритмов в онлайн-режиме со стабильной точностью

# Глава 6

## Заключение

### Цель

В ходе данной работы была достигнута поставленная цель – разработан алгоритм, позволяющий в реальном времени идентифицировать устройства с точностью  $>90\%$ .

### Задачи

Был произведен анализ предметной области, исследованы алгоритмы понижения размерности и классификации, а также различные способы адаптации решения к работе в реальном времени.

- Подобрано эффективное признаковое описание:
  - С помощью преобразований сигнала – последовательного вычитания друг из друга симметричных частей спектра, полученного после взятия модуля от быстрого преобразования Фурье от частей преамбулы и дальнейшего их объединения, – была уменьшена размерность с 480 признаков до 130 с потерей точности менее, чем 0.5%.
  - Смоделирован с помощью библиотеки Keras и обучен автоэнкодер, уменьшающий размерность с 130 признаков до 20 с потерей точности менее, чем 1.5%.
- Произведено сравнение и подбор классификаторов:
  - Модели, основанные на решающих деревьях дали точность 0.8-0.9, но показали себя немасштабируемыми: для того, чтобы давать хороший результат, им необходимо обучиться на большом объеме данных.
  - CatBoost от Yandex хорошо показала себя как модель, имеющая хорошую вариативность гиперпараметров и метрик, а также, даже без оптимизации параметров, CatBoost дает точность  $>0.85$ .
  - Модель MLP, схожая по устройству с автоэнкодером, показала лучший результат – Accuracy = 0.92.

- Разработан и реализован алгоритм онлайн-классификации:
  - В основу алгоритма легли Autoencoder и Multilayer Perceptrone
  - Обучение алгоритма происходит последовательно по эпохам на рандомизированных данных
  - Алгоритм имеет возможность сохранения модели для дальнейшего использования
  - Модель не требует большого количества памяти для сохранения

# Список литературы

- [1] Merchant, K., Revay, S., Stantchev, G., et al.:  
Deep learning for RF device fingerprinting in cognitive communication networks.  
*IEEE J. Sel. Top. Signal Process.*, 12(1), 160–167 (2018)
- [2] Danev, B., Capkun, S.:  
Transient-based identification of wireless sensor nodes.  
*International Conference on Information Processing in Sensor Networks IEEE 2009*,  
pp. 25–36. IEEE (2009)
- [3] Toonstra, J., Kinsner, W.:  
Transient analysis and genetic algorithms for classification.  
*Conference Proceedings of the IEEE Communications, Power, and Computing IEEE 1995*, WESCANEX 95, vol. 2, pp. 432–437. IEEE (1995)
- [4] Polak, A.C., Dolatshahi, S., Goeckel, D.L.:  
Identifying wireless users via transmitter imperfections.  
*IEEE J. Sel. Areas Commun.* 29(7), 1469–1479 (2011)
- [5] Novikoff A. B. J. :  
On convergence proofs on perceptrons  
*Proceedings of the Symposium on the Mathematical Theory of Automata*.  
Vol. 12. Polytechnic Institute of Brooklyn, 1962. Pp. 615–622
- [6] Nigmatullin R.R., Vorobev A.S.:  
The “Universal” Set of Quantitative Parameters for Reading of the Trendless  
Sequences  
*Fluctuation and Noise Letters*, 18(04), 2019
- [7] Li, Z., Xu, W., Miller, R., et al.:  
Securing wireless systems via lower layer enforcements.  
*ACM Workshop on Wireless Security ACM*, pp. 33–42. ACM (2006)
- [8] К. В. Воронцов  
Математические методы обучения по прецедентам (теория обучения машин),  
*В рамках курса лекций по машинному обучению*, 2013