# ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет Санкт-Петербургская школа физико-математических и компьютерных наук

Гащенко Екатерина Александровна

# Исследование применимости методов машинного обучения для детектирования преамбул и синхронизации в процедуре начального доступа в сетях сотовой связи 4G-5G

Выпускная квалификационная работа

по направлению подготовки 01.04.02 Прикладная математика и информатика образовательная программа «Машинное обучение и анализ данных»

Рецензент д-р техн. наук, проф. <u>А.М.Тюрликов</u> Научный руководитель к. техн. наук, ст. преп. <u>А.А.Шпильман</u> Консультант С.В.Торгашов

Санкт-Петербург 2023

# Содержание

1	Абб	бревиатуры и сокращения	10
2	Обі	цее описание процедуры случайного доступа	11
3	Пос	становка задачи и принятые ограничения	15
4	Оп	исание принципа генерации преамбулы	17
	4.1	Описание преамбулы	17
	4.2	Генерация преамбулы во временной области	18
	4.3	Генерация преамбулы в частотной области	18
<b>5</b>	Оп	исание принципа работы корреляционного обнаружителя	19
	5.1	Общее описание корреляционного обнаружителя	19
	5.2	Корреляционный обнаружитель во временной области	20
	5.3	Корреляционный обнаружитель в частотной области	20
	5.4	Порог обнаружителя	20
	5.5	Определение задержки	22
6	Mo	делирование канала связи с шумом	23
	6.1	Отношение сигнал-шум (SNR)	23
	6.2	Модель канала с аддитивным шумом	23
	6.3	Данные	25
7	Обн	наружение сигнала в канале без временной задержки	27
	7.1	Корреляционный обнаружитель	27

	7.2	Линейный метод	27
	7.3	Многослойный перцептрон	28
	7.4	Сверточная нейронная сеть	29
	7.5	Автоэнкодер	30
	7.6	Сравнение подходов	31
8	Bpe	еменная задержка и циклический префикс	33
	8.1	Временная задержка	33
	8.2	Циклический префикс	34
9	Обн	аружение сигнала в канале связи с временной задержкой	37
	9.1	Многослойный перцептрон	37
	9.2	Сверточная нейронная сеть	40
		9.2.1 Эксперименты с разными моделями	40
		9.2.2 Эксперименты с разными параметрами отношения сигнал-	
		шум и задержки	46
	9.3	Автоэнкодер	48
	9.4	Сравнение подходов	51
10	Оце	енка сложности аппаратной реализации	53

### Аннотация

В данной работе рассматривается задача детектирования комплексной последовательности Задова-Чу и определения ее временной задержки в канале с аддитивным гауссовским шумом. Задача решается с помощью применения нескольких видов моделей нейронных сетей, таких как многослойный перцептрон, сверточная нейронная сеть, автоэнкодер. В работе проводится сравнительный анализ результатов работы этих моделей, а также оценка сложности их аппаратной реализации. Один из существующих подходов для решения данной задачи – алгоритм корреляционного обнаружителя. Мотивация использования нейронных сетей для решения поставленной задачи в том, что нейронные сети имеют ряд преимуществ в аппаратной реализации перед алгоритмом корреляционного обнаружителя. А именно: нейронные сети легче адаптируются к новым условиям среды, а также аппаратные решения на базе нейронных сетей можно адаптировать для решения различных задач. Работа разбита на описание построения данных на основе модели канала с шумом, описание работы корреляционного обнаружителя, сравнение результатов моделей для решения задачи без временной задержки и с временной задержкой.

# Annotation

In this paper, we consider the problem of detecting a complex Zadov-Chu sequence and determining it is time delay in a channel with additive Gaussian noise. The problem is solved using several types of neural network models, such as a multilayer perceptron, a convolutional neural network, an autoencoder. The paper presents a comparative analysis of the results of these models and an assessment of the complexity of their hardware implementation. The correlation detector algorithm is often used to solve this problem. The motivation for using neural networks to solve the problem is that neural networks are easier to adapt to new environmental conditions than the classical algorithm of the correlation detector. Hardware solutions for neural networks can be reused for the needs of various tasks. The work is divided into a description of data construction based on a noisy channel model, a description of a correlation detector, a comparison of the results of models for solving a problem without a time delay and with a time delay.

# Список ключевых слов

Сети сотовой связи, 4G, 5G, процедура случайного доступа, преамбула, временная задержка, последовательность Задова-Чу, нейронные сети, многослойный перцептрон, сверточная нейронная сеть, автоэнкодер.

## Введение

Данная работа посвящена исследованию применимости нейронных сетей в телекоммуникационном оборудовании, а именно в процедуре случайного множественного доступа мобильных терминалов (UE) в сети. Устройства, с которыми взаимодействуют мобильные терминалы и которые обеспечивают покрытие мобильными сервисами, называются базовыми станциями (eNB). Прежде чем осуществлять передачу или прием данных, мобильному терминалу необходимо пройти первичную регистрацию и синхронизироваться во времени с базовой станцией. Процедуру регистрации и синхронизации необходимо повторять каждый раз, когда мобильный терминал выходит из состояния сна: приходит входящий звонок, сообщение, абонент решил воспользоваться телефоном и т.п. Также регистрация и синхронизация требуются, если мобильный терминал перемещается в другую соту (зона покрытия, обслуживаемая одной базовой станцией, часто иллюстрируется шестиугольником).

Для этих целей существует процедура случайного доступа. Одним из начальных этапов этой процедуры является этап детектирования преамбулы и синхронизации. Для этого мобильный терминал осуществляет передачу специальной последовательности (преамбулы) на базовую станцию. Базовая станция ожидает получить одну из заданных конечным множеством последовательностей. Если принятая последовательность попадает в нужное множество, осуществляется синхронизация. Для этого вычисляется временная задержка на основе принятого сигнала. Сложность такой задачи заключается в том, что на практике в канале связи, по которому передается преамбула, присутствуют различные шумы (аддитивные, мультипликативные), присутствует затухание сигнала и помехи от соседних сот. Необходимо правильно детектировать до 64-х различных преамбул. При этом нужно избежать детектирования и регистрации преамбул, которые не входят в ожидаемое множество, так как это создает излишнюю нагрузку на базовую станцию.

Чаще всего такую задачу решают с помощью алгоритма корреляционного обнаружителя (подробнее о нем в главе 5). В идеальных условиях (а именно при известном уровне мощности шума относительно мощности исходного сигнала и самом известном исходном сигнале) невозможно получить решение лучше, чем решение корреляционного обнаружителя [8] для канала с аддитивным гауссовским шумом. Но на практике мы сталкиваемся с негауссовскими или с неаддитивными шумами, мы не знаем значение мощности шума относительно мощности исходной преамбулы, и конечно же не знаем искомого сигнала. Для работы с алгоритмом при неизвестной мощности шума относительно мощности исходной преамбулы, используются различные адаптивные алгоритмы, использующие оценки значения SNR (характеристика, показывающая насколько сильно мощность шума больше мощности полезного сигнала, подробнее о ней в главе 6.1). Но оценить значение SNR точно – довольно сложная математически и алгоритмически задача [6] [7].

В этой работе мы рассмотрим иной подход к решению задачи детектирования преамбулы и временной синхронизации, а именно с помощью нейронных сетей. Мы рассмотрим несколько моделей нейронных сетей для решения этой задачи и сравним их результаты. Мотивация использования нейронных сетей в том, что нейронные сети обладают гибкой структурой: при смене условий среды, достаточно переобучить нейронную сеть на новых данных, и загрузить новые веса. Классический алгоритм корреляционного обнаружителя сложнее

8

адаптировать, особенно если реализация поддержана на аппаратном уровне. Также имея несколько решений различных задач на базе нейронных сетей можно переиспользовать аппаратное решение одной задачи под нужды другой задачи, что дает возможность сэкономить ресурсы. Это достигается за счет того, что все нейронные сети используют только ограниченный ряд математических операций внутри себя. На аппаратном уровне достаточно реализовать этот набор, после чего можно перестраивать одно аппаратное решение под другое. Классические алгоритмы для решения разных задач так перестроить друг под друга не всегда возможно. Тем не менее перенос нейронных сетей на аппаратный уровень также является нетривиальной задачей. В главе 10 приведена краткая оценка сложности полученных результатов с точки зрения аппаратной реализации.

# 1 Аббревиатуры и сокращения

В данной работе используются некоторые общепринятые английские аббревиатуры. В таблице 1 приведена их расшифровка на русском языке.

UE	Мобильный терминал
eNB	Базовая станция
LTE	Стандарт беспроводной высокоскоростной передачи данных
RA	Случайный доступ
RAP	Преамбула процедуры случайного доступа
PRACH	Физический канал передачи запросов случайного доступа
PUSCH	Физический канал передачи пользовательского трафика
MLP	Многослойный перцептрон
CNN Сверточная нейронная сеть	
АЕ Автоэнкодер	
МАЕ Среднее абсолютное отклонение	
MCE	Средняя точность
ROC AUC	Площадь под кривой ошибок
FFT	Быстрое преобразование Фурье
IFFT	Обратное быстрое преобразование Фурье
ВСЕ Бинарная кросс-энтропия	
SNR	Отношение сигнал-шум
db	Децибел

Таблица 1: Сокращения

# 2 Общее описание процедуры случайного доступа

В 4G/5G синхронизация в нисходящем канале (передатчик = eNB, получатель = UE) происходит с помощью специального канала синхронизации. В этом случае сигнал синхронизации транслируется всем и передается постоянно с определенным интервалом. Однако в нашем случае во восходящем канале для осуществления регистрации мобильного терминала в сети такой подход неэффективен: если мобильный терминал будет транслировать такой сигнал, он потратит на это много энергии и может создать большие помехи для других UE. Поэтому требуется такая процедура, при которой мобильному терминалу достаточно отправить сравнительно короткую сигнализацию на базовую станцию для прохождения регистрации и синхронизации. Мобильный терминал передаёт такой сигнал в сторону базовой станции только при осуществлении передачи.

Таким образом, необходимо осуществлять регистрацию и синхронизацию максимально эффективно: на стороне eNB необходимо распознавать запросы с минимально возможным количеством ошибок, чтобы UE не тратила ресурсы на повторные запросы. При этом также нужно минимизировать количество ложных обнаружений сигналов от UE, чтобы также не тратить ресурсы eNB. Кроме того, весь процесс должен происходить достаточно быстро, стандарт 3GPP определяет окно обнаружения и ответа в пределах 10 миллисекунд [3]. Процедура случайного доступа (RA, Random Access) включает в себя процесс регистрации и временной синхронизации. Синхронизация требуется для дальнейшей передачи сообщений от мобильного терминала на базовую

11

станцию. После получения информации о временной задержке, мобильный терминал может скорректировать время начала передачи дальнейших сообщений, чтобы попасть в нужное временное окно, в которое ожидает прием сообщения базовая станция.

Перед началом процедуры случайного доступа, мобильный терминал получает информацию о первичной синхронизации от базовой станции. Для этого мобильный терминал сканирует радио эфир в поисках сот с максимальным сигналом. Базовая станция в каждой соте транслирует первичный сигнал синхронизации (PSS), который позволяет синхронизироваться по TTI, слотам и OFDM-символам. PSS представляет собой взаимно-ортогональные последовательности Задова-Чу. Вторичный синхросигнал (SSS) позволяет мобильному терминалу определить идентификатор соты (в качестве SSS используются М-последовательности). После установления первичной синхронизации мобильный терминал может прочитать системную информацию, относящуюся к выбранной соте, и определить частотно-временное окно, в котором может выполнять передачу преамбулы (преамбула – часть последующей процедуры случайного доступа). При этом мобильный терминал не знает своё реальное расстояния до базовой станции и не может оценить запаздывание сигнала — эта задача решается базовой станцией в процессе приёма и детектирования преамбулы во время процедуры случайного доступа, поскольку, одна из целей процедуры случайного доступа – определение базовой станцией запаздывания сигнала от мобильного терминала относительно синхросигналов. Процедура состоит из нескольких сообщений.

Message 1 (сообщение 1) – Random Access Preamble (RAP) – преамбула, которая является комплексной последовательностью специального вида, облада-

12

ющая особыми корреляционными свойствами. При успешном детектировании этой последовательности на стороне базовой станции принимается решение о начале процедуры регистрации мобильного терминала. Одновременно с этим определяется временная задержка для установления временной синхронизации. Далее происходит выделение ресурсов и создание ответного сообщения. Message 2 (сообщение 2) – Random Access Response (RAR) – ответ eNB на преамбулу, который посылается согласно полученным данным из message 1. Содержит необходимые параметры для message 3.

RA используется для получения радио ресурса для передачи message 3 (сообщения 3, см. рис. 1). Существует несколько различных типов message 3 в зависимости от ситуации.

Наше исследование будет сосредоточено вокруг выделенного красным на рис.1: детектирование преамбулы и оценка временной задержки.



Рис. 1: Диаграмма процедуры Random Access

# 3 Постановка задачи и принятые

ограничения

В нашей работе мы будем рассматривать преамбулы формата 4 согласно стандарту LTE: последовательность имеет длину 139. Преамбулы аналогичного формата используются в стандарте 5G.

Считаем, что осуществляется однократная передача преамбулы от одного мобильного терминала в канале с аддитивным шумом (не рассматриваем отражения сигнала и мультипликативные шумы). Предполагаем отсутствие разделения частного ресурса между PRACH и PUSCH. Модель канала будем строить с одной передающей и одной принимающей (SISO – Single Input Single Output) антеннами.

Сигнал будет моделироваться в комплексном виде. Полезным сигналом считаем один цикл последовательности.

Таким образом, мы работаем с комплексными последовательностями Задова-Чу с временным сдвигом и циклическим префиксом в канале с аддитивным гауссовским шумом без затухания сигнала. Временной сдвиг и циклический префикс описаны в разделе 8.

Необходимо детектировать сигнал в канале, который содержит одну из 64-х заданных последовательностей. Параметры (а именно индекс и циклический сдвиг, подробно описанные в разделе 4) этих 64-х последовательностей выбираются согласно стандарту 3GPP [1]. Эти параметры в стандарте заданы так, чтобы максимизировать дистанцию во времени между соседними последовательностями, а также иметь возможность задавать разные наборы преамбул для соседних сот. Также на вход может быть подана последовательность с

15

иными параметрами или последовательность, в которой присутствует только шум. Такие последовательности не должны детектироваться. Помимо детектирования сигнала, мы хотим определять значение временной задержки сигнала.

# 4 Описание принципа генерации преамбулы

## 4.1 Описание преамбулы

Преамбула генерируется с помощью последовательностей Задова-Чу [13]. Последовательность Задова-Чу – комплексная последовательность, заданная следующим соотношением:

$$x_u(n) = \exp(-j\frac{\pi u n(n+c_f+2q)}{N_{ZC}}),$$
 (1)

где  $N_{ZC}$  – длина последовательности,

$$q \in \mathbf{Z},$$

 $c_f = N_{ZC} \mod 2$ ,

u – индекс последовательности,  $0 < u < N_{ZC}$ ,  $gcd(N_{ZC}, u) = 1$ ,

 $0 \leq n < N_{ZC}$ .

Для генерации преамбулы используются последовательности с q = 0 [1], длина последовательности – простое число из некоторого множества. Таким образом, мы используем последовательности вида

$$x_u(n) = \exp(-j\frac{\pi u n(n+1)}{N_{ZC}})$$
(2)

Такая последовательность выбрана из-за ее полезных свойств:

- 1. Последовательность имеет постоянную амплитуду во временной и частотных областях.
- Коэффициент корреляции последовательности с любым ее ненулевым циклическим сдвигом равен 0. При нулевом (по модулю длины последовательности) циклическом сдвиге автокорреляция равна 1.

3. Коэффициент корреляции двух последовательностей с разными индексами  $u_1$  и  $u_2$  равен  $\frac{1}{N_{ZC}}$ , если  $N_{ZC}$  не кратен  $|u_1 - u_2|$ . Поэтому принято брать  $N_{ZC}$  простым числом.

### 4.2 Генерация преамбулы во временной области

Во временной области преамбула имеет вид

$$x_{u,\nu}(n) = x_u((n+C_{\nu}) \mod N_{ZC}),$$
 (3)

где  $C_{\nu}$  – циклический сдвиг [1]. Пары  $(u, C_{\nu})$  заданы стандартом 3GPP специальным образом, чтобы избежать пересечения последовательностей.

### 4.3 Генерация преамбулы в частотной области

В частотной области преамбулу можно вычислить с помощью применения дискретного преобразования Фурье к преамбуле во временной области:

$$y_{u,\nu}(n) = \sum_{m=0}^{N_{ZC}-1} x_{u,\nu}(m) \cdot \exp(-j\frac{2\pi mn}{N_{ZC}})$$
(4)

# 5 Описание принципа работыкорреляционного обнаружителя

### 5.1 Общее описание корреляционного обнаружителя

На вход обнаружитель получает зашумленный сигнал  $h(t) = w_1(t)u(t) + w_2(t)$ , где  $w_1(t)$  – мультипликативный шум,  $w_2(t)$  – аддитивный шум. Мультипликативные шумы могут быть вызваны отражением сигнала от поверхностей (в нашем случае полагаем  $w_1(t) = 1$ ), аддитивные шумы могут быть вызваны, например, сигналами от других UE в канале. Цель обнаружителя определить наличие сигнала u(t) в канале связи (его может не быть вовсе, либо может быть сигнал с неподходящими параметрами) и оценить его запаздывание.

Для этого генерируются эталонные сигналы  $u_{x_i}(t)$ , где  $x_i$  некоторые параметры, которые определяют сигнал, взятые из конечного множества. В нашем случае это длина  $N_{ZC}$ , циклический сдвиг  $C_{\nu}$ , индекс последовательности u; подробные таблицы с допустимыми значениями этих параметров можно найти в 3GPP[1]. Для каждой рассматриваемой области, в которой может находиться UE и передавать сигнал, имеется не более 64-х различных последовательностей. Таким образом, обнаружитель ищет только ограниченное множество последовательностей.

Для каждой пары  $(u_{x_i}(t), h(t))$  вычисляется корреляция. Если для данного  $x_i$  результат корреляции превышает определенный порог, то принимается решение, что передавался сигнал  $u_{x_i}(t)$ .

# 5.2 Корреляционный обнаружитель во временной области

Корреляция c(k) двух комплексных последовательностей u(t), v(t) длины N вычисляется по формуле

$$c(k) = (u \star v)(k) = \sum_{t=0}^{N-1} \overline{u(t)} v((t+k) \mod N),$$
(5)

где  $\overline{v}(t)$  – комплексное сопряженное для v(t), k – сдвиг последовательностей относительно друг друга,  $k \in [0, 2N - 1]$  [5, 9].

# 5.3 Корреляционный обнаружитель в частотной области

Корреляцию двух последовательностей в частотной области  $U = F_D(u), V = F_D(v)$  можно вычислить с помощью теоремы о корреляции [10]:

$$(u \star v)(k) = F_D^{-1}(\overline{U}V)_k \tag{6}$$

Где  $F_D(x)$  – дискретное преобразование Фурье,  $F_D^{-1}(x)$  – обратное дискретное преобразование Фурье.

Тогда:

$$c(k) = (u(t) \star v(t))(k) = F_D^{-1}(\overline{U(t)}V(t))_k = \frac{1}{N} \sum_{t=0}^{N-1} \frac{\overline{U(t)}V(t)}{N} \cdot \exp(j\frac{2\pi tk}{N})$$
(7)

### 5.4 Порог обнаружителя

Порог обнаружителя M, при превышении которого последовательность принимается, вычисляется по алгоритму. На вход подается зашумленная последовательность  $x = \{x\}_{i=1}^{N}$ , вычисляется

$$\eta = 2\sqrt{\mathbf{E}(x) * \mathbf{D}(x)},\tag{8}$$

где  $\mathbf{E}(x) = \sum_{i=1}^{N} \frac{x_i}{N}$  – среднее значение,  $\mathbf{D}(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mathbf{E}(x))^2$  – дисперсия.

Из последовательности x удаляются все пики – значения, которые превышают порог  $\eta$ . Получается новая последовательность y. Если она пуста, то положим y = x. Тогда  $M = C \cdot \mathbf{E}(y)$ , где C подбирается (также может периодически корректироваться неким адаптивным алгоритмом). Подобный подход также реализован в srsRAN [12].

На рисунке 2 пример вычисленной корреляции между двумя сигналами (зашумленным и предполагаемым) в частотной области (синий график) и значение порога (оранжевая линия). По оси абсцисс отложен индекс корреляции, по оси ординат – ненормированное значение корреляции (арифметический результат свёртки).



Рис. 2: Корреляция сигналов и порог обнаружителя в частотной области

На рисунке 3 пример работы обнаружителя при достаточно низком значении SNR. Обнаружитель не справляется обнаружить пик, превышающий порог. При этом в сигнале была последовательность и корреляция вычисляется именно с исходной последовательностью, поэтому в этом случае обнаружитель ошибается, выдавая отрицательный ответ. Зеленой линией на рисунке отображен результат возможного адаптивного алгоритма. При использовании адаптивного алгоритма пороговое значение может корректироваться в зависимости от оценки SNR в канале связи.



Рис. 3: Корреляция сигналов и порог в частотной области, отрицательный результат обнаружителя

### 5.5 Определение задержки

Значение задержки определяется как индекс максимального пика в отрезке значений корреляции от 0 до MAX DELAY, где MAX DELAY – максимально возможное значение задержки.

### 6 Моделирование канала связи с шумом

### 6.1 Отношение сигнал-шум (SNR)

Отношение сигнал-шум (SNR, Signal to Noise Ratio) – безразмерная величина, равная отношению мощности полезного сигнала к мощности шума. В децибелах выражается как:

$$SNR(db) = 10 \log_{10}(\frac{P_{signal}}{P_{noise}})$$
(9)

Где  $P_{signal}$  – средняя мощность полезного сигнала,  $P_{noise}$  – средняя мощность шума.

#### 6.2 Модель канала с аддитивным шумом

Мы создали свою модель канала связи с шумом: на вход подается комплексная последовательность Задова-Чу во временной или частотной области, значение SNR в децибелах, временная задержка, длина циклического префикса. Сигнал сдвигается согласно значению временной задержки, и добавляется циклический префикс, если он был задан.

Вычисляется мощность  $P_s$  входного сигнала x(t):

В частотной области:

$$P_s = \frac{1}{N_{ZC}} \left( \sum_{t=1}^{N_{ZC}} |X(f)|^2 \right)$$
(10)

Во временной:

$$P_s = \sum_{t=1}^{N_{ZC}} |x(t)|^2 \tag{11}$$

На основе SNR и полученной мощности сигнала вычисляется мощность  $P_n$  шума:

$$SNR = 10 \log_{10}(P_s) - 10 \log_{10}(P_n)$$
$$P_n = ((10 * \log_{10}(P_s) - SNR)/10)^{10}$$
(12)

Положим, что мнимая и вещественная части шума распределены одинаково как нормальные величины со средним значением ноль и отклонением  $\sigma$ . Вычислим отклонение  $\sigma$  для шума в частотной области. Мощность шума w(f) в частотной области:

$$P_n = \frac{1}{N} \sum_{t=1}^{N} |w(f)|^2 \tag{13}$$

Среднеквадратичное отклонение  $\Sigma$  комплексной последовательности w(f) равно:

$$\Sigma = \sqrt{\frac{1}{N} \sum_{t=1}^{N} |w(f)|^2},$$
(14)

т.к. математическое ожидание считаем равным 0. Распишем среднеквадратичное отклонение комплексной последовательности через среднеквадратичное отклонение мнимой  $\sigma_{img}$  и вещественной  $\sigma_{real}$  частей. В нашем предположении они равны, поэтому обозначим их через  $\sigma$ :

$$\Sigma = \sqrt{\sigma_{img}^2 + \sigma_{real}^2} = \sqrt{2\sigma^2}$$

$$\sigma = \sqrt{\frac{1}{2N} \sum_{t=1}^N |w(f)|^2}$$

$$\sigma = \sqrt{P_n/2}$$
(15)

На основе этого генерируется вещественная и мнимая часть аддитивного шума в частотной области. Далее полученная последовательность складывается с входным сигналом, если тот был в частотной области, либо последовательность переводится во временную область с помощью обратного преобразования Фурье и уже там складывается с входным сигналом, если тот был изначально во временной области.

### 6.3 Данные

На основе полученной модели канала мы сгенерировали несколько наборов данных для обучения и тестирования моделей. Для генерации задаются значение диапазона SNR, размер данных, область (частотная или временная). Длина циклического префикса и максимальное значение временной задержки фиксированы, подробнее об этом в разделе 8.

Фиксируются 64 различных набора параметров (индекс и сдвиг) для генерации последовательностей Задова-Чу, согласно 3GPP. Эти последовательности считаются верными и помечаются индексом 1. Берутся еще другие последовательности Задова-Чу и помечаются как ложные последовательности (индекс 0) – это сделано для имитации помех от UE из других областей, которые мы не хотим детектировать. Также генерируются "пустые"последовательности, которые содержат только шум. Они тоже помечаются как ложные последовательности. При генерации очередной последовательности равномерной выбирается, будет она иметь индекс 1 или 0. Если последовательность не пуста, то ее индекс и сдвиг также выбираются равномерно из соответствующего множества.

К последовательностям добавляется циклический префикс (если он задан) и

временной сдвиг (если он задан). Временной сдвиг выбирается случайно из заданного диапазона от нуля до некоторого максимального значения сдвига (подробнее описано в главе 8), является равномерно распределенной величиной.

Полученные последовательности проходят через модель канала (пункт 5.2), где на них накладывается шум, согласно случайно выбранному значению SNR из заданного диапазона. Значение SNR равномерно распределено на всем заданном промежутке.

Если были заданы значения циклического префикса и временного сдвига, зашумленные последовательности обрезаются до длины равной сумме длины циклического префикса и длины исходной последовательности. В данной работе мы не удаляем циклический префикс.

# Обнаружение сигнала в канале без временной задержки

Для начала мы рассмотрим случай с нулевой временной задержкой. Задача состоит в том, чтобы определить, содержит ли принятый сигнал преамбулу с заданными параметрами длины, индекса и сдвига. Длина фиксирована и равна 139, что соответствует преамбуле формата 4 [11] в стандарте LTE. Параметры индекса и сдвига берутся из конечного множества, которое определяется по таблицам из 3GPP [1]. Сигнал содержит аддитивный гауссовский шум.

### 7.1 Корреляционный обнаружитель

Корреляционный обнаружитель реализован согласно разделу 5: вычисляется корреляция и порог. Если в корреляции присутствует пик, который превышает порог, то обнаружитель выдает, что в канале была нужная последовательность. Это простейший вариант корреляционного обнаружителя, который выбран в качестве базового решения. Метрика качества: ROC AUC.

### 7.2 Линейный метод

Как самая простая модель была выбрана линейная модель. На вход подается тензор размера  $2^*139$  – сконкатинированная реальная и мнимая часть комплексного сигнала. Обучается вектор весов  $\{w\}_0^{N=278}$ , предсказание формируется как

$$f(x) = \begin{cases} 1, w \cdot x + w_0 > 0\\ 0, иначе \end{cases}$$
(16)

Где "1"означает присутствие последовательности, а "0"отсутствие. Обучение длится 1000 эпох, использована L2 регуляризация. В качестве функции ошибки выбрана MCE (Mean Consequential Error). Для сравнения с другими методами оценивается метрика ROC AUC.

### 7.3 Многослойный перцептрон

Трехслойный перцептрон для бинарной классификации. На вход подается тензор размера 2\*139 – сконкатинированная реальная и мнимая часть комплексного сигнала. Для обучения используется оптимизатор Adam [16], обучение длится 100 эпох. В качестве функции ошибки выбрана бинарная кроссэнтропия. Для сравнения с другими методами оценивается метрика ROC AUC.



Рис. 4: Диаграмма перцептрона

### 7.4 Сверточная нейронная сеть



Рис. 5: Диаграмма сверточной нейронной сети

На рисунке 5 представлена схема сверточной нейронной сети. На вход поступает тензор размерности (2,  $N_{ZC}$ ) – комплексная последовательность сигнала, разделенная на мнимую (Imaginary Part) и вещественную (Real Part) части. Далее следует два сверточных слоя с MaxPooling-ом, Dropout-ом и функцией активации ReLU. В конце три линейных слоя с функцией активации ReLU и последний слой с одним выходом и функцией активации Sigmoid.

Для обучения используется оптимизатор Adam, обучение длится 40 эпох. В качестве функции ошибки выбрана бинарная кросс-энтропия. Для сравнения

с другими методами оценивается метрика ROC AUC.

### 7.5 Автоэнкодер



Рис. 6: Диаграмма автоэнкодера

На рисунке 6 представлена схема автоэнкодера. На вход поступает тензор размерности  $(2, N_{ZC})$  – комплексная последовательность сигнала, разделенная на мнимую (Imaginary Part) и вещественную (Real Part) части. Автоэнкодер состоит из кодера и декодера. Кодер состоит из двух сверточных слоев с батч-нормализацией и линейного слоя. Декодер соответственно состоит из линейного слоя и двух обратных сверточных слоев с батч-нормализацией. Входная последовательность проходит через кодер, декодер и линейный слой. На выходе из декодера тензор, который имеет длину как входной тензор, пропущенный через слой Flatten. Функция активации на последнем слое – Sigmoid. В кодере и декодере используется функция активации ReLU. Для обучения используется оптимизатор Adam, обучение длится 40 эпох. В качестве функции ошибки выбрана бинарная кросс-энтропия. Для сравнения с другими методами оценивается метрика ROC AUC.

### 7.6 Сравнение подходов

Для сравнения подходов выбрана метрика ROC AUC (площадь под кривой ошибок). Сгенерирован набор сигналов в частотной области с SNR в промежутке [-30, -5], без временной задержки и циклического префикса. В данных присутствуют 64-и различные последовательности с шумом, они считаются нужными последовательностями и имеют метку "1". Также имеются другие 64-и последовательности, они все помечены как "0". Ко всем последовательностям добавлен шум, согласно модели канала с аддитивным шумом (пункт 5.2). Добавлены сигналы с нулевой последовательностью, то есть в них присутствует только шум. Такие сигналы также помечены как "0". Для каждого эксперимента вычисляется метрика ROC AUC с помощью кросс валидации (5 итераций). Результаты приведены в таблице 2

Модель	ROC AUC метрика	
Корреляционный обнаружитель	0.99	
Линейная модель	0.81	
Многослойный перцептрон	0.99	
Сверточная нейронная сеть	0.97	
Автоэнкодер	0.94	

Таблица 2: Результаты

Лучше всего себя показывает перцептрон: его результаты ближе всего к результатам корреляционного обнаружителя. Это может быть связано с простотой задачи детектирования последовательности без временных задержек. Модели с более сложной архитектурой могут переобучаться, хотя мы старались этого избежать.

# 8 Временная задержка и циклический префикс

### 8.1 Временная задержка

Временная задержка (TO, Timing Offset) соответствует времени, которое требуется сигналу для достижения базовой станции с мобильного устройства. Требуется для установления синхронизации между восходящими (отправка сообщений) и нисходящими (принятие сообщений) каналами на стороне базовой станции. Это необходимо, например, для совместного доступа несколькими пользователями (UE) к одной частоте: базовая станция распределяет временные промежутки, в которые ожидает услышать сообщение от пользователей. На основе временной задержки мобильный терминал может заранее отправить сообщение, чтобы попасть во временной промежуток, в который его ожидает услышать базовая станция.

Таким образом, значение временной задержки равно задержке распространения сигнала между мобильным терминалом (UE) и базовой станцией (eNB) в предположении, что задержка одинакова в восходящем и нисходящем канале.

Начальное значение временной задержки определяется базовой станцией с помощью преамбулы, передаваемой мобильным терминалом в процессе процедуры случайного доступа. Вычисленное значение временной задержки в виде команды Timing Advance передаётся на мобильный терминал в сообщении RA Response, которое базовая станция отсылает мобильному устройству в качестве ответа на переданную преамбулу.

33

Временная задержка измеряется в единицах  $T_S$ :

$$T_S = \frac{1 \text{ second}}{15000 \cdot 2048}$$
 (17)

Команда Timing Advance, посылаемая на мобильный терминал (UE), содержит целочисленное значение сдвига восходящей передачи кратное  $16T_S$ . Возможные значения команды  $N_{TA}$  в сообщении RA Response – целые числа в диапазоне [0, 1282] (11 бит).

### 8.2 Циклический префикс

Для работы с сигналами, которые имеют временную задержку, к исходной последовательности добавляется циклический префикс. Циклический префикс – это добавляемый в начало сигнала кусок, взятый с конца передаваемого сигнала. Согласно 3GPP для преамбул формата 4 длина циклического префикса составляет  $448T_S$ , а длина временной задержки составляет не более длины циклического префикса, то есть не более  $448T_S$ . [2], [11]



Рис. 7: Циклический префикс

В реальном канале после генерации последовательности Задова-Чу длины 139 (для последовательностей формата 4), последовательность распределяется на 139 поднесущих, шаг между которыми составляет 7,5КГц. Таким образом, длина последовательности будет 1/7500 = 133,  $(3)\mu s = 4096T_S$ . На стороне базовой станции после приема сигнала происходит обратная процедура, где из поднесущих извлекается последовательность.



Рис. 8: Схема распределения сигнала на поднесущие

В нашей работе нет смысла моделировать распределение на поднесущие и обратную этому процедуру, поэтому мы рассмотрим задержку и циклический префикс без перевода в единицы  $T_S$ . Длина последовательности Задова-Чу равна 139, поэтому длина циклического префикса и максимально возможная задержка составляют  $139 * 448/4096 \approx 15$ .

# 9 Обнаружение сигнала в канале связи с временной задержкой

Рассмотрим случай, когда в канале присутствует временная задержка. Считаем, что время передачи сигнала линейно зависит от расстояния между приёмником и передатчиком. Теперь к последовательностям в данных будет добавлять циклический префикс. Длина префикса и максимальное значение временной задержки описаны в разделе 8. Сигнал также содержит аддитивный гауссовский шум. Эксперименты будут проводиться при значении SNR в промежутке [-30, -5]. Значение SNR равномерно распределено. В канале присутствует только аддитивный гауссовский шум. Длина циклического префикса и максимальное значение задержки равно 15. Значение временной задержки равномерно распределено от 0 до 15. Длина последовательности Задова-Чу равна 139. В качестве искомых правильных последовательностей используются 64 различные последовательности Задова-Чу. В качестве неправильных последовательностей, которые нельзя детектировать, взяты другие 64 последовательности Задова-Чу. А также пустые последовательности – сигналы, в которых присутствует только шум.

### 9.1 Многослойный перцептрон

Для начала рассмотрим задачу, где все последовательности верные (имеют индекс 1). Задача состоит в определении значения временной задержки. Для этого построим многослойный перцептрон на три слоя, на финальном слое котором находится N + 1 выходов, где N – максимальное значение временной задержки. На вход подается тензор размера 2\*139 – сконкатинированная

реальная и мнимая часть комплексного сигнала. Перцептрон осуществляет классификацию на N классов – значение задержки. Обучение проходит в 100 эпох, функция ошибки – кросс-энтропия. Метрика оценки качества – среднее абсолютное отклонение (MAE). Для этого у выходного тензора выбирается индекс, который имеет максимальное значение, – это будет искомое значение задержки. Здесь и далее используется MAE, а не другая оценка точности классификатора, так как нам интереснее оценить отклонение от истинного значения задержки.

На рисунке 9 представлена схема перцептрона для классификации задержки сигнала.



### Рис. 9: Перцептрон для классификации задержки

В таблице 3 представлены результаты экспериментов.

Модель	MAE
Перцептрон для классификации задержки	3.2
Таблица 3: Результаты	

Рассмотрим теперь отдельно задачу детектирования последовательности

на данных с временными задержками. Для этого используем трехслойный перцептрон для бинарной классификации. На вход также подается тензор размера 2\*139 – сконкатинированная реальная и мнимая часть комплексного сигнала. Функция ошибки – бинарная кросс-энтропия. Обучение длится 100 эпох, используется оптимизатор Adam. Для оценки качества используется оценка точности предсказания.

На рисунке 10 представлена схема перцептрона для детектирования последовательности.



Рис. 10: Перцептрон-детектор

В таблице 4 представлены результаты экспериментов.

Модель	Точность
Перцептрон-детектор	0.82

Таблица 4: Результаты

Из таблиц видно, что перцептрон плохо справляется с поставленной задачей.

### 9.2 Сверточная нейронная сеть

#### 9.2.1 Эксперименты с разными моделями

Аналогично сначала рассмотрим задачу классификации на N + 1 класс для определения временной задержки в предположении, что данные содержат только верные последовательности. На рисунке 11 представлена схема работы классификатора. Функция ошибки – кросс-энтропия, оптимизатор Adam.



Рис. 11: Сверточная нейронная сеть для классификации

Далее рассмотрим задачу регрессии для определения временной задержки в предположении, что данные содержат только верные последовательности. На рисунке 12 представлена схема работы сверточной нейронной сети с регрессией. Функция ошибки – среднеквадратичная ошибка, оптимизатор Adam.



Рис. 12: Сверточная нейронная сеть для регрессии

На вход в обоих случаях поступает тензор размерности  $(2, N_{ZC})$  – комплексная последовательность сигнала, разделенная на мнимую (Imaginary Part) и вещественную (Real Part) части. Для сравнения результатов используется метрика среднего абсолютного отклонения (MAE).

Модель	MAE
CNN классификация	1.9
СNN регрессия	2.1

Таблица 5: Результаты

В таблице 5 представлены результаты экспериментов.

Регрессия показывает результаты хуже классификации. Это связано с тем, что мы предсказываем дискретные величины.

Для решения общей задачи детектирования и определения задержки, одним из возможных решений будет создать связку из двух нейронных сетей: первая детектирует (например, это будет MLP), и если результат положительный, передает последовательность на вторую сеть (например, классификация CNN), которая определяет задержку. Тогда вторую сеть достаточно обучать только на верных последовательностях. Для оценки качества работы первой сети (MLP) используется метрика точности (ассигасу). Для оценки работы второй сети (классификация CNN) используется метрика среднего абсолютного значения (MAE). Для этого у выходного тензора выбирается индекс, который имеет максимальное значение, – это будет искомое значение задержки. На рисунке 13 представлена схема работы двух нейронных сетей. На вход двум нейронным сетям поступает тензор размерности (2,  $N_{ZC}$ ) – комплексная последовательность сигнала, разделенная на мнимую (Imaginary Part) и вещественную (Real Part) части. Вторая нейронная сеть осуществляет работу только в том случае, когда первая сеть детектирует последовательность.



Рис. 13: Связка перцептрона и сверточной нейронной сети

Модель	Точность	MAE
Связка MLP и CNN	0.82	1.9

Таблица 6: Результаты

В таблице 6 представлены результаты такой последовательной работы двух сетей.

Теперь попробуем создать единую сеть на основе сверточных нейронных сетей, которая бы решала обе задачи.

Для этого пусть в сети будут несколько общих сверточных слоев. Выход с этих слоев подается параллельно на два линейных слоя: первый имеет вы-

ход размера 1, а второй N + 1. На рисунке 14 представлена схема нейронной сети с двумя выходами. Первый выход отвечает за детектирование последовательности, а второй за определение задержки. На вход поступает тензор размерности (2,  $N_{ZC}$ ) – комплексная последовательность сигнала, разделенная на мнимую (Imaginary Part) и вещественную (Real Part) части.



Рис. 14: Сверточная нейронная сеть с двумя выходами

Такая сеть тренируется сначала на данных, которые содержат только правильные последовательности: последовательности, помеченные лейблом "1". Функция ошибки учитывает только выход второй головы, функция ошибки – кросс энтропия:

$$Loss = CrossEntropy\_Loss(o[1 : N + 2], t)$$
(18)

о – тензор, который возвращает нейронная сеть.

*t* – тензор, который содержит данные из датасета, на котором происходит обучение. Он содержит только значение временной задержки.

Затем нейронная сеть доучивается на полном датасете: данные содержат последовательности с метками "1"и "0". Функция ошибки представляет собой взвешенную сумму ошибки ВСЕ для "первой головы"и кросс энтропии для второй головы:

$$Loss = BCE\_Loss(o[0], t[0]) + C * t[0] * CrossEntropy\_Loss(o[1:N+2], t[1])$$
(19)

o – тензор, который возвращает нейронная сеть. o[0] – выход первой головы, который отвечает за детектирование сигнала. o[1:N+2] – выход второй головы, который отвечает за временную задержку.

t – тензор, который содержит данные датасета, на котором происходит обучение. t[0] – принимает значение 0 или 1, определяет метку последовательности. Если t[0] = 1, то в сигнале присутствовала правильная последовательность, если t[0] = 0, то в сигнале нет нужной последовательности и нет смысла вычислять временную задержку. t[1] – принимает значение от 0 до N, значение временной задержки.

С – регулирующий множитель.

Таким образом обучение происходит в два этапа: на данных, где присутствуют только последовательности с метками "1 и на полных данных. Первый этап длится 20 эпох, второй 15 эпох.

Для оценки качества результатов выхода, который детектирует последовательность, используется метрика точности (accuracy). Для оценки выхода, который определяет задержку, используется метрика среднего абсолютного значения (MAE). Для этого у выходного тензора выбирается индекс, который

45

имеет максимальное значение, – это будет искомое значение задержки.

Модель	Точность	MAE
CNN с двумя выходами	0.86	2.0

Таблица 7: Результаты

В таблице 7 представлены результаты такой сети.

# 9.2.2 Эксперименты с разными параметрами отношения сигналшум и задержки

Ранее все эксперименты были проведены при уровне SNR [-30, -5]. Теперь построим те же эксперименты, но при других уровнях мощности шума. В качестве модели взята модель сверточной нейронной сети с двумя выходами. Используются те же метрики: точность для детектирования и среднее абсолютное отклонение для значения задержки. Метрики вычислены на кросс валидации. В таблице 8 представлены результаты.

Промежуток SNR	Точность	MAE
[-30, -5]	0.86	2.0
[-20, -5]	0.92	1.34
[-15, -5]	0.95	0.69

Таблица 8: Результаты при разных значениях SNR

Видно, что при увеличении значений SNR результаты качественно улучшаются. Отдельно рассмотрим случай при SNR = -6.9db. Такое значение предлагает стандарт 3GPP [4]. для оценки качества работы построенных алгоритмов.

SNR	Точность	MAE
-6.9	0.97	0.61

Таблица 9: Результаты при SNR = -6.9 db

Рассмотрим изменение качества предсказаний модели при изменении длины максимальной задержки. SNR будет в промежутке [-30, -5]. В таблице 10 представлены результаты экспериментов.

Максимальная задержка	Точность	MAE
8	0.89	0.52
10	0.87	0.85
13	0.84	1.59
15	0.86	2.0

Таблица 10: Результаты при разном значении максимальной задержки

### 9.3 Автоэнкодер



Рис. 15: Автоэнкодер

На рисунке 15 представлена схема автоэнкодера. На вход поступает тензор размерности  $(2, N_{ZC})$  — комплексная последовательность сигнала, разделенная на мнимую (Imaginary Part) и вещественную (Real Part) части. Автоэнкодер состоит из кодера и декодера. Кодер состоит из двух сверточных слоев с батч-нормализацией и линейного слоя. Декодер соответственно состоит из линейного слоя и двух обратных сверточных слоев с батч-нормализацией. Сначала обучается автоэнкодер. Он учится очищать входную последователь-

ность от шума. Для этого используются данные, в которых представлена исходная зашумленная последовательность длины 139 и такая же последовательность без наложенного шума как эталонная. Чтобы получить последовательности длины 139, после симуляции работы канала (добавления циклического префикса, сдвига и шума) последовательность обрезается до первых 139 символов. Аналогично поступаем с эталонной последовательностью, но без наложения шума. В качестве функции ошибки используется среднеквадратичная ошибка. Оптимизатор Adam.

После обучения автоэнкодера к нему добавляются дополнительные полносвязные слои для предсказания задержки и детектирования последовательности. На рисунке 16 представлена общая схема.



Рис. 16: Автоэнкодер с двумя полносвязными выходами

Входная последовательность проходит через автоэнкодер. Далее после-

довательность пропускается параллельно через две последовательности линейных слоев, образуя два выхода. Первый выход имеет размерность 1, он детектирует последовательность. Второй выход имеет размерность N + 1, он классифицирует значение задержки.

Теперь необходимо дообучить такую сеть. Для обучения используется оптимизатор Adam, функция ошибки:

 $Loss = BCE\_Loss(o[0], t[0]) + C * t[0] * CrossEntropy\_Loss(o[1:N+2], t[1]) (20)$ 

о – тензор, который возвращает нейронная сеть. o[0] – выход первой головы,
 который отвечает за детектирование сигнала. o[1 : N + 2] – выход второй
 головы, который отвечает за временную задержку.

t – тензор, который содержит данные датасета, на котором происходит обучение. t[0] – принимает значение 0 или 1, определяет метку последовательности. Если t[0] = 1, то в сигнале присутствовала правильная последовательность, если t[0] = 0, то в сигнале нет нужной последовательности и нет смысла вычислять временную задержку. t[1] – принимает значение от 0 до N, значение временной задержки.

С – регулирующий множитель.

Для оценки полученных результатов используется оценка точности детектирования и среднее абсолютное отклонение для оценки предсказания задержки. В таблице 12 представлены результаты экспериментов для автоэнкодера, максимальная задержка равна 15, SNR в промежутке [-30, -5].

Модель	Точность	MAE	
Автоэнкодер	0.94	1.6	

Таблица 11: Результаты

Проверим результаты автоэнкодера при SNR = -6.9db, как делали для сверточной нейронной сети:

SNR	Точность	MAE
-6.9	0.99	0.55

Таблица 12: Результаты автоэнкодера при SNR = -6.9

### 9.4 Сравнение подходов

В таблице 14 собраны результаты полученных моделей при SNR в промежутке [-30, -5] и максимальной задержке равной 15. В таблицу первой строкой добавлены результаты корреляционного обнаружителя при условии, что алгоритму известны параметры искомой последовательности и значение SNR.

I

1

Модель	Точность	MAE
Корреляционный обнаружитель с известными параметрами	0.99	0.0
Корреляционный обнаружитель	0.97	1.0
Перцептрон для определения задержки	_	3.2
Перцептрон для детектирования	0.82	_
Перцептрон и сверточная нейронная сеть	0.82	1.9
Сверточная нейронная сеть с двумя выходами	0.86	2.0
Автоэнкодер с двумя полносвязными выходами	0.94	1.6

Таблица 13: Результаты всех моделей

Из всех моделей машинного обучения автоэнкодер показывает самые лучшие результаты: результаты наиболее приближенные к результатам корреляционного обнаружителя. Так как корреляционный обнаружитель (с известными параметрами искомого сигнала и значения SNR) теоретически лучший алгоритм для решения нашей задачи, приближение результатов модели к его результатам является хорошим показателем для оценки качества модели.

# 10 Оценка сложности аппаратной реализации

Для реализации решения для телекоммуникационного оборудования для сетей сотовой связи интересна именно аппаратная реализация алгоритмов обработки сигнала на физическом уровне (L1) в силу требований к производительности. Один из самых простых подходов для исследования этого направления – прототипирование таких устройств на ПЛИС (Программируемая Логическая Интегральная Схема). Реализация нейронных сетей на ПЛИС позволяет значительно увеличить производительность за счет параллелизации вычислений.

Для переноса необходимо обучить нейронную сеть стандартными способами. После вычисления весов модели, можно построить аналогичную структуру сети на ПЛИС.

Оценим сложность требуемых ПЛИС для работы с построенными нейронными сетями. Для оценки посчитаем требуемое количество умножителей и сумматоров на самом большом слое нейронной сети. Рассмотрим нейронные сети описанные в главе 9.

Модель	Умножители	Сумматоры	Beca	
Перцептрон для определения задержки	8896	8832	8896	
Перцептрон для детектирования	8896	8832	8896	
Сверточная нейронная сеть с	276480	276352	276480	
двумя выходами	210400			
Автоэнкодер с двумя	27800	27700	27800	
полносвязными выходами	21000	21100	21000	

Таблица 14: Сложность моделей для ПЛИС

В таблице 14 приведено количество умножителей, сумматоров и весов для различных моделей для их самых больших слоев. Обеспечение параллельной работы такого количества умножителей и сумматоров в числах с плавающей точкой является нетривиальной задачей и в общем случае без дополнительной декомпозиции может не иметь практически пригодного решения. Приведенные выше результаты только для сравнительной оценки. Перенос архитектуры нейронной сети на ПЛИС является отдельной трудоемкой задачей [14], [15], которую в рамках данной работы мы не рассматриваем.

## Заключение

В работе проведен сравнительный анализ нескольких моделей нейронных сетей для решения задачи детектирования преамбулы и определения временной задержки в канале с аддитивным гауссовским шумом и равномерно распределенной временной задержкой. По результатам проделанной работы можно видеть, что модель на основе автоэнкодера представляется наиболее перспективной и показывает результаты наиболее приближенные к теоретическим результатам корреляционного обнаружителя. Из этого следует, что можно продолжать работу над нейронными сетями для решения данной задачи, в частности над моделью с автоэнкодером. В будущем можно поставить эксперименты над этой нейронной сетью с другими моделями канала (модели каналов Райса и Релея, затухание сигнала), а также рассмотреть переход от вычислений с плавающей точкой к вычислениям с фиксированной точкой для экономии вычислительных ресурсов. Можно начать разработку архитектуры модели нейронной сети для переноса на ПЛИС.

Данная работа может служить хорошим стартом для разработки решения поставленной задачи в реальных условиях и разработки готового решения для телекоммуникационного оборудования. Исследование подтверждает эффективность применения нейронных сетей в задаче обнаружения преамбулы и определения временной задержки, а также открывает путь для будущих связанных исследований.

## Список литературы

- [1] Specification 38.211, 6.3.3.1 3GPP
- [2] Specification 36.211, 5.7.1 3GPP
- [3] Specification 36.321, 5.4.1 3GPP
- [4] Specification 36.141, 8.4.1.1 3GPP
- [5] A. F. Kohn (2006). Autocorrelation and Cross-Correlation Methods. In: Wiley Encyclopedia of Biomedical Engineering, p. 271
- [6] Zadoff-Chu Sequence Based Time-Domain SNR Estimation for OFDM Systems, Milan Zivkovic, Rudolf Mathar, 2014
- [7] Adaptive Modulation With CAZAC Preamble-Based Signal-to-Noise-Ratio Estimator in OFDM Cooperative Communication System, Shahid Manzoor, Noor Shamsiah Othman, 2022
- [8] Статистическая радиотехника. В.И.Тихонов, издательство Советское Радио, Москва, 1966
- [9] correlate, NumPy документация
- [10] Cross-Correlation Theorem
- [11] 14.1 PRACH, Long Term Evolution in Bullets, Chris Johnson
- [12] srsRAN prach, github.com

- [13] 9.2 Zadoff Chu (ZC) Sequences, LTE for 4G Mobile Broadband: Air Interface Technologies and Performance, Farooq Khan
- [14] Разработка на ПЛИС нейронной сети, реализующей распознавание объектовинтереса на изображениях, Холод Николай Григорьевич, 2019
- [15] Разработка аппаратной реализации нейронных сетей на ПЛИС, Американов Александр Александрович, 2017
- [16] Оптимизатор Adam, документация Pytorch