|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Фундаментальные науки

КАФЕДРА ФН11 «Вычислительная математика и математическая физика»

**ОТЧЕТ ПО ПРАКТИКЕ**

Студент Галкин Дмитрий Александрович

*фамилия, имя, отчество*

Группа ФН11-62Б

Тип практики производственная

Название предприятия ФГБУ Институт проблем управления им. В.А.Трапезникова РАН (ИПУ РАН)

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Галкин Д.А.

*подпись, дата фамилия, и.о.*

Руководитель практики **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Прозоровский А.А.

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2019 г.*

СОДЕРЖАНИЕ

с.

ВВЕДЕНИЕ……………………………………………………………………………………….....3

1 Отчёт о посещении ИПУ РАН им. В.А. Трапезникова………….…………….…...………......4

1.1 ИПУ РАН им. В.А. Трапезникова…………….………………..……………………………4

1.2 История создания…………………………………………………………………………….4

1.3 Лаборатория № 77………....…………………………………………………………………5

2 Процесс решения задачи классификации в машинном обучении ….......………….….…........6

3 Выполнение индивидуального практического задания ………………………………………..7

3.1 Теоретическая часть………………………………………………………………………….7

3.2 Практическая часть ...……………………………………………………….…………….....9

3.2.1 Постановка задачи …...…..………………………………………………………………9

3.2.2 Генерация данных ......………..……………………..………………………………….10

3.2.3 Обучение и классификация ……….………………..………………………………….11

3.2.4 Структура программного пакета …………..………..…………...…………………….13

3.2.5 Результаты работы программы ………………………………………………………..14

ЗАКЛЮЧЕНИЕ..…………………………………………………………..………………………19

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ………………...…………............................20

ВВЕДЕНИЕ

Производственная практика студентов МГТУ им. Н.Э. Баумана является обязательной частью основной образовательной программы высшего образования, одной из форм организации учебного процесса.

Практика – вид учебной работы, направленный на развитие практических навыков и умений, а также формирование компетенций обучающихся в процессе выполнения определенных видов работ, связанных с будущей профессиональной деятельностью.

Основными видами практики студентов Университета, обучающихся по основным образовательным программам высшего профессионального образования, являются:

- учебная;

- производственная.

Производственная практика направлена на формирование у студентов

практических профессиональных умений, приобретение практического опыта по основным видам производственной деятельности для последующего освоения ими общих и профессиональных компетенций по избранной специальности/направлению подготовки.

Целями производственной практики являются:

- ознакомление с реальным производственным процессом на предприятии и участие в нём;

- выполнение индивидуального задания.

Задачами производственной практики являются:

- решение задачи и проведение исследования результатов в лаборатории ИПУ РАН;

- изучение природы сигналов ЛЧМ, зашумления сигналов и оценки качества сигнала;

- изучение этапов исследовательской работы, по решению задачи классификации методами машинного и глубокого обучения;

- выполнение индивидуального задания, которое включает в себя генерацию данных, зашумление тестовых данных, разработку программных модулей для решения задачи многоклассовой классификации сигналов и для оценки качества моделей классификации;

- изучение ГОСТ 7.32-2001 «Отчёт о научно-исследовательской работе. Структура и правила оформления».

1 Отчёт о посещении ИПУ РАН им. В.А. Трапезникова

* 1. ИПУ РАН им. В.А. Трапезникова

Основные направления научной деятельности Института проблем управления им. В.А. Трапезникова РАН, утверждённые Президиумом Российской академии наук 18 марта 2008 г.:

- теория систем и общая теория управления;

- методы управления сложными техническими и человеко-машинными системами;

- теория управления в междисциплинарных моделях организационных, социальных, экономических, медико-биологических и экологических систем;

- научные основы технологий управления подвижными объектами и навигации;

-теория и методы разработки программно-аппаратных и технических средств управления и сложных информационно-управляющих систем;

Научные основы интегрированных систем управления и автоматизации технологических процессов и управление производством [1].

1.2 История создания [2]

Институт автоматики и телемеханики (ИАТ) АН СССР основан 16 июня 1939 года указом Совнаркома СССР. В 1969 году ИАТ переименован в Институт проблем управления (ИПУ) АН СССР, а в 1998 году Институту присвоено имя академика В.А. Трапезникова, бывшего директором Института с 1951 по 1987 годы.

В указе Совнаркома была обозначена цель «развернуть фундаментальные исследования в области теории автоматического регулирования и создания автоматических устройств». За почти 80-летний период область деятельности ИПУ РАН значительно расширилась и включает в настоящий момент все аспекты теории и практики управления, в том числе: теория систем и общая теория управления; методы управления сложными техническими и человеко-машинными системами; теория управления в междисциплинарных моделях организационных, социальных, экономических, медико-биологических и экологических систем; научные основы технологий управления подвижными объектами и навигации; теория и методы разработки программно-аппаратных и технических средств управления и сложных информационно-управляющих систем; научные основы интегрированных систем управления и автоматизации технологических процессов управления производством.

Наряду с фундаментальными исследованиями в области современной теории управления, Институт активно участвует в прикладных разработках по заказу Министерства обороны, Министерства внутренних дел, ОАО «РЖД», РКК «Энергия», Федерального НПЦ «Научно-исследовательский институт измерительных систем им. Ю.Е. Седакова», ОАО «Научно-исследовательский электромеханический институт», ВНИИ по эксплуатации атомных электростанций, ФГУП «ЦАГИ», ФГУП «ГосНИИАС», ОАО «Корпорация Иркут» и многих других организаций. Благодаря исследованиям ИПУ РАН в разные годы были решены важнейшие научно-технические задачи государственного масштаба.

Сегодня ИПУ РАН, в котором работают более 800 человек (в т.ч. более 100 докторов наук и более 200 кандидатов наук), – один из самых крупных и авторитетных научных центров мира в области теории управления, методов и средств автоматизации. Более 100 сотрудников ИПУ РАН в разные годы были отмечены Государственными премиями СССР и РФ, премиями Правительства РФ и Совета министров СССР, научными премиями и медалями Академии наук СССР и РАН, а также зарубежных академий и университетов.

Институт продолжает развиваться, расширять и углублять фундаментальные исследования в области теории управления и ее приложений. В последние годы на конкурсной основе организованы два десятка молодежных научных школ под руководством ведущих ученых Института; действуют базовые кафедры ведущих вузов; значителен ежегодный набор в аспирантуру; имеются лаборатории, в которых большинство составляют молодые сотрудники.

Ежегодно в стенах Института проводится ряд профильных международных и всероссийских научных и научно-практических конференций и семинаров. В их работе принимают участие сотни ведущих специалистов российской и мировой науки управления.

1.3 Лаборатория № 77

Лаборатория «Вычислительной кибернетики» № 77 была организована в стенах института проблем управления в апреле 2017 года под руководством к. т. н. Макаренко А. В. Название «вычислительная кибернетика» очень точно отражает суть научных подходов и методов, которые составляют основу деятельности лаборатории:

- кибернетика;

- машинное обучение;

- нелинейная динамика.

Круг задач, решаемых сотрудниками лаборатории очень широк. Среди них разработка алгоритмов для обработки, локализации, классификации сигналов (например, сигналов оптических рефлектометров, укладываемых рядом с магистральными нефтепроводами), исследования, связанные с глубоко адаптивными системами активного диалога (диалоговые системы в банках, страховых компаниях и медицинских учреждениях).

Стратегия функционирования лаборатории состоит в организации взаимодействия между «интерфейсными посредниками» [3] (специалистами, которые, с одной стороны хорошо знают предметную область, в которой они хотели бы решать задачу методами машинного обучения и кибернетики, а с другой стороны, концептуально понимают техническую сторону реализации решений, предоставляемых лабораторией) и, со стороны ИПУ РАН, специалистами для решения задач (математики, программисты) и специалистами по системному анализу (постановки цели и задач, увязка проекта, ранжирование ограничений).

Такое сотрудничество между теми, кто ставит задачу и теми, кто ее решает, позволяет работать с широким тематическим разбросом: от электротехники и экономики, до биофизики и космологии. Все эти задачи структурно, на уровне математики, идентичны и связаны с анализом временных рядов, идентификацией сложных нелинейных систем, с выявлением пространственно-временных паттернов.

2 Процесс решения задачи классификации в машинном обучении

Исторически задача классификации возникла из задачи машинного зрения, поэтому часто употребляемый синоним – распознавание образов. В классической задаче классификации обучающая выборка представляет собой набор отдельных объектов обозначаемых:

где – объект, представленный в виде:

где – вектор из k признаков объекта.

В качестве исхода объекта *x* фигурирует переменная *y*, принимающая конечное число значений из множества:

,

где Y – множество меток классов.

Требуется построить алгоритм/модель (классификатор), который по вектору признаков *x* вернул бы метку класса или вектор оценок принадлежности (апостериорных вероятностей) к каждому из классов.

Для определения качества построенной модели и для последующей тонкой настройки, улучшения предсказательной функции используются различные метрики качества, котороые, вообще говоря, могут показывать широкий разброс оценок, что связано с природой данных, процессов и постановкой задачи.

Дальнейшее улучшение показателей метрики достигается различными способами. Наиболее распространенные подходы – это изучение признаков объектов, выявление наиболее значимых и информативных, понижение признакового пространства объектов выборки, и побор параметров модели по сетке.

3 Выполнение индивидуального практического задания

Целью индивидуального практического задание было сгенерировать ЛЧМ-сигналы с разной скоростью изменения частот, зашумление данных и сравнение моделей классификации волн, с целью выявить наиболее подходящей для дальнейшей работы.

3.1 Теоретическая часть

ЛЧМ-сигнал – сигнал, частота которого изменяется во времени по линейному закону. Замечательное свойство ЛЧМ-сигналов, определяющее их практическую значимость, состоит в следующем. Предположим, что имеется некоторое физическое устройство, осуществляющее задержку сигналов, подаваемых ему на вход. Если время задержки зависит от частоты сигнала, причем с ростом частоты это время уменьшается, то при определенных условиях, подавая на вход такого устройства ЛЧМ-импульс большой длительности, можно добиться существенного «сжатия» его во времени. Этот эффект обусловлен тем, что на выходе устройства задержки одновременно будут появляться составляющие как более низкочастотные, относящиеся к началу импульса, так и более высокочастотные.

Если потери в устройстве сжатия малы, то амплитуда выходного сигнала может значительно превысить уровень шумов. Это повышает надежность обнаружения радиолокационным приемником слабых отраженных сигналов [4].

Отношение сигнал/шум (ОСШ; [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) signal-to-noise ratio, сокр. SNR) — [безразмерная величина](https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D0%B7%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B2%D0%B5%D0%BB%D0%B8%D1%87%D0%B8%D0%BD%D0%B0), равная отношению [мощности](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D1%89%D0%BD%D0%BE%D1%81%D1%82%D1%8C) полезного [сигнала](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB) к мощности [шума](https://ru.wikipedia.org/wiki/%D0%A8%D1%83%D0%BC). Рассчитывается по формуле:

где *P* — средняя мощность;

*A* — [среднеквадратичное](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B5%D0%B4%D0%BD%D0%B5%D0%B5_%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5) значение амплитуды;

Оба сигнала измеряются в полосе пропускания системы.Обычно отношение сигнал/шум выражается в [децибелах](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%86%D0%B8%D0%B1%D0%B5%D0%BB) (дБ). Чем больше это отношение, тем меньше шум влияет на характеристики системы. Формула в [дБ]:

Основные причины высокого уровня шума в сигнальных системах:

- рассогласованные линии передачи сигнала;

- [тепловой шум](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BF%D0%BB%D0%BE%D0%B2%D0%BE%D0%B9_%D1%88%D1%83%D0%BC) и [дробовой шум](https://ru.wikipedia.org/wiki/%D0%94%D1%80%D0%BE%D0%B1%D0%BE%D0%B2%D0%BE%D0%B9_%D1%88%D1%83%D0%BC) в компонентах системы;

- недостаточная разрядность [АЦП](https://ru.wikipedia.org/wiki/%D0%90%D0%A6%D0%9F), резонансные явления;

- паразитные связи ([паразитная ёмкость](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B7%D0%B8%D1%82%D0%BD%D0%B0%D1%8F_%D1%91%D0%BC%D0%BA%D0%BE%D1%81%D1%82%D1%8C));

- [самовозбуждение](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%BC%D0%BE%D0%B2%D0%BE%D0%B7%D0%B1%D1%83%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5) систем;

- нелинейность передаточных характеристик.

Классификация ЛЧМ-сигналов находит свое применение в задачах улучшения Тактико-технические характеристики (ТТХ) приемников ЛЧМ-сигналов: детектирование сигналов, выявление их свойств (в данной конкретной задаче: классификация по скорости изменения частот волн). Сами ЛЧМ-сигналы применяются в радиолокации и гидролокации, в радиолокаторах для исследования состояния ионосферы. Также использование ЛЧМ-сигнала позволяет упростить оцифровку радиолокационных данных, обработку изображений и радиотомограмм.

Типичный вид ЛЧМ-сигнала приведён на Рисунке 2.

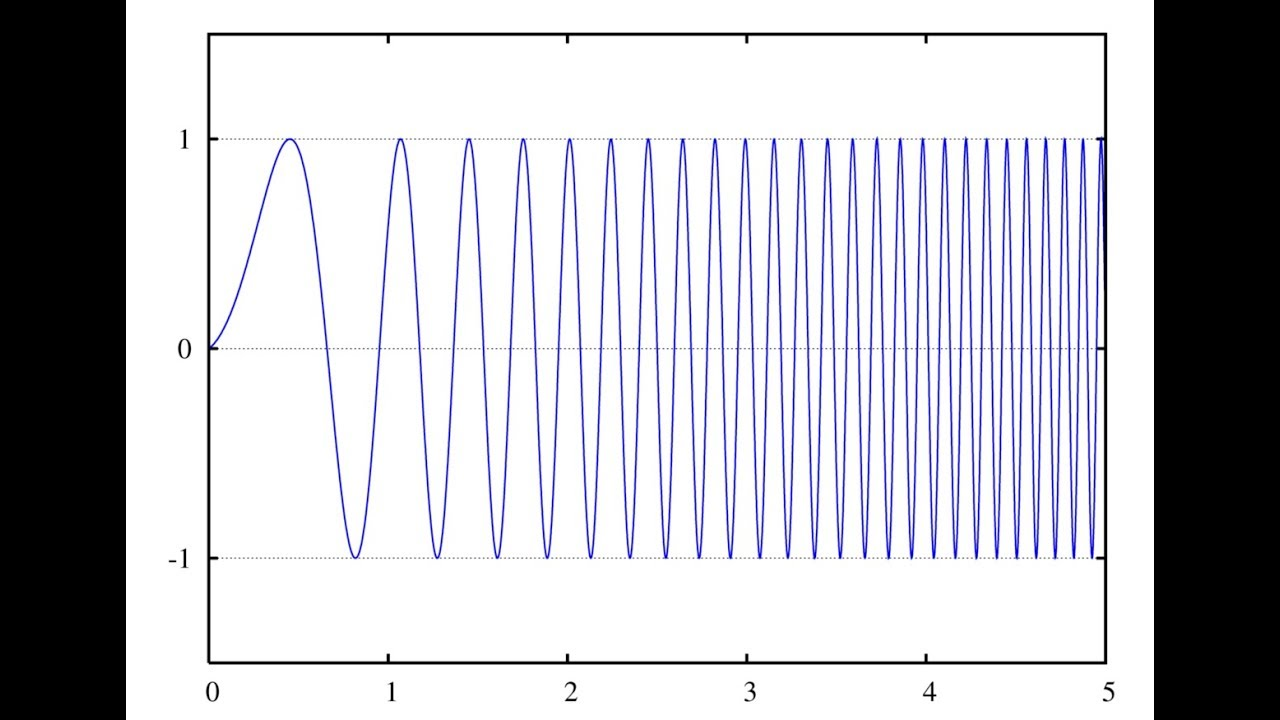


Рисунок 2– ЛЧМ-сигнал

3.2 Практическая часть

3.2.1 Постановка задачи

Написать программный пакет для численной генерации чистых сигналов ЛЧМ и их зашумления. Обучить на «чистых» данных классификаторы методами классического машинного обучения и глубокого обучения. Проанализировать качество работы классификаторов, реализованных различными подходами на тестовых данных разной степени зашумленности. Предусмотреть возможность изменения начальных параметров сигналов:

– скорости изменения частоты сигналов;

– начальной фазы сигналов;

– характеристик распределения шума в сигналах;

– период времени, в который рассматривается сигнал.

3.2.2 Генерация данных

При численной генерации данных, на которых далее будет проведено обучение алгоритмов и оценка их качества, воспользуемся общим уравнением сигнала следующего вида:

,

,

где – амплитуда сигнала;

*t* – время;

начальная фаза сигнала.

Таким образом, мы получаем формулу для генерации чистого сигнала с произвольно заданной скоростью изменения частоты. Для решения задачи классификации было решено сгенерировать данные для трех классов сигнала: 1-ый сигнал со скоростью изменения частоты , 2-ой сигнал с параметром , а 3-й класс представлял собой не ЛЧМ сигнал, а некий произвольно взятый сигнал любой природы (для конкретики был сгенерирован шум из нормального распределения).

Амплитуда сигнала для классов 1 и 2, начальная фаза сигнала генерируется из равномерного распределения от 0 до , а время наблюдения сигнала изменяется от 0 до 128 с С шагом 1 с . Т. о. сигнал протяженностью в 128 с задается вектором значений сигнала в момент времени *t*:

,

где S – сигнал протяженность в 128 с.

Для зашумления данных прибавляем к чистому сигналу шум из нормального распределения:

*,*

где – чистый сигнал;

*S –* чистый сигнал;

*N* – вектор шумов из нормального распределения , где – произвольно задается.

Так как исходные чистые сигналы имели значения в диапазоне *[-1, 1],* то после зашумления данных появляется необходимость в масштабировании полученных значений к исходному диапазону. Этот процесс выполняется по формуле:

*+a,*

где *p* – преобразованное значение;

*s* – исходное значение;

*[, ] –* диапазон изменения входной переменной;

*[a, b] –* допустимый диапазон входных переменных (в данной задаче *[-1, 1]*).

Параметр для зашумления данных будем менять в соответствие с показателем отношения сигнал/шум (ОСШ; [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) signal-to-noise ratio, сокр. SNR), измеряемым в  [децибелах](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%86%D0%B8%D0%B1%D0%B5%D0%BB) (дБ). В данном исследовании будем менять значение , меняя SNR в диапазоне *[5, -30]* с шагом 2,5 дБ. Таким образом мы получаем 15 папок с 30 наборами данных для каждой.

Наборы данных, состоят из тренировочной выборки (чистых сигналов) размером 1080 и тестовой выборки (зашумленных данных) размером 120. Объектов каждого класса в выборках поровну.

3.2.3 Обучение и классификация

Решение данной задачи сводится к использованию некоторых классических методов машинного обучения и построению классификатора на основе многослойной нейронной сети (англ. A multilayer perceptron сокр.MLP). Суть задачи сводится к сравнению качества алгоритмов классификации и к представлению сводной таблицы результатов. Для сравнения были выбраны следующие методы с параметрами, заданными по умолчанию:

- наивный байесовский классификатор (GaussianNB) — простой вероятностный [классификатор](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8), основанный на применении [теоремы Байеса](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D0%91%D0%B0%D0%B9%D0%B5%D1%81%D0%B0) со строгими (наивными) предположениями о [независимости](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B7%D0%B0%D0%B2%D0%B8%D1%81%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D1%8C);

- логистическая регрессия (LogisticRegression) — метод построения [линейного классификатора](http://www.machinelearning.ru/wiki/index.php?title=%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80), позволяющий оценивать апостериорные вероятности принадлежности объектов классам;

- RandomForest (с [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) — «случайный лес»)  — алгоритм машинного обучения, предложенный [Лео Брейманом](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B5%D0%B9%D0%BC%D0%B0%D0%BD,_%D0%9B%D0%B5%D0%BE) и [Адель Катлер](https://ru.wikipedia.org/w/index.php?title=%D0%90%D0%B4%D0%B5%D0%BB%D1%8C_%D0%9A%D0%B0%D1%82%D0%BB%D0%B5%D1%80&action=edit&redlink=1), заключающийся в использовании комитета (ансамбля) [решающих деревьев](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE_%D0%BF%D1%80%D0%B8%D0%BD%D1%8F%D1%82%D0%B8%D1%8F_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B9). Алгоритм сочетает в себе две основные идеи: метод [бэггинга](https://ru.wikipedia.org/wiki/%D0%91%D1%8D%D0%B3%D0%B3%D0%B8%D0%BD%D0%B3) Бреймана, и [метод случайных подпространств](https://ru.wikipedia.org/w/index.php?title=Random_subspace_method&action=edit&redlink=1), предложенный Tin Kam Ho. Алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля [решающих деревьев](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE_%D0%BF%D1%80%D0%B8%D0%BD%D1%8F%D1%82%D0%B8%D1%8F_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B9), каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим;

- метод k-ближайших соседей ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) k-nearest neighbors algorithm, k-NN) — метрический алгоритм для автоматической [классификации](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8) объектов или [регрессии](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)). В случае использования метода для [классификации](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8) объект присваивается тому классу который является наиболее распространённым среди k-соседей данного элемента, классы которых уже известны;

- Метод опорных векторов ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) SVM, support vector machine) — набор схожих алгоритмов [обучения с учителем](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81_%D1%83%D1%87%D0%B8%D1%82%D0%B5%D0%BB%D0%B5%D0%BC), использующихся для [задач классификации](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8) и [регрессионного анализа](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7). Принадлежит семейству [линейных классификаторов](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80) и может также рассматриваться как специальный случай [регуляризации по Тихонову](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%83%D0%BB%D1%8F%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%A2%D0%B8%D1%85%D0%BE%D0%BD%D0%BE%D0%B2%D0%B0) [5]. Особым свойством метода опорных векторов является непрерывное уменьшение эмпирической ошибки классификации и увеличение зазора, поэтому метод также известен как метод классификатора с максимальным зазором. Основная идея метода — перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Две параллельных гиперплоскости строятся по обеим сторонам гиперплоскости, разделяющей классы. Разделяющей гиперплоскостью будет гиперплоскость, максимизирующая расстояние до двух параллельных гиперплоскостей. Алгоритм работает в предположении, что чем больше разница или расстояние между этими параллельными гиперплоскостями, тем меньше будет средняя ошибка классификатора.

Также в задаче используется классификатор построенный на основе многослойного перцептрона. Создадим нейросеть следующей архитектуры: 128 входа под каждый признак, 50 нейронов в скрытом слое и функцией активации *relu*, слой Dropout с параметром 0,2, и три выхода — по количеству классов, с активационной функцией *sigmoid*. Вопрос архитектуры сетей до сих пор не является решенным, выбор конфигурации, как правило, основывается на опыте и оптимизируется с помощью экспериментов [6]. Зададим 200 эпох обучения с количеством батчей равным 10.

Обучение проводится следующим образом: в рамках 30 наборов данных из каждой папки, соответствующей определенному значению , каждый классификатор обучается на тренировочной выборке. Затем, на тестовой выборке зашумленных данных предсказываются классы, после чего проводится оценка классификатора. В качестве оценки берется *f1-*метрика, которая определяется следующим образом:

где - *f1-*метрика;

– точность;

*Recall* – полнота.

Точность (precision) и полнота (recall) являются метриками которые используются при оценке большей части алгоритмов извлечения информации. Иногда они используются сами по себе, иногда в качестве базиса для производных метрик, таких как F-мера или R-Precision. Суть точности и полноты очень проста.

Точность системы в пределах класса – это доля объектов, действительно принадлежащих данному классу относительно всех объектов, которые система отнесла к этому классу. Полнота системы – это доля объектов, найденных классификатором и принадлежащих классу относительно всех объектов этого класса в тестовой выборке [7].

В итоге для каждого классификатора мы получаем набор *f1*-метрик по 30 выборкам для каждого значения параметра , т. е. 450 значений метрики.

Для большей достоверности выводов о качестве работы методов классификации в зависимости от изменения шумовых характеристик строятся графики зависимости квантилей *f1*-метрик (в виде box-plot) от SNR.

3.2.4 Структура программного пакета

Программный пакет реализован таким образом, что его легко можно модифицировать. Для программной реализации всех необходимых методов и для генерации данных использован язык Python. В файле, в котором выполнено исследование, импортируются все необходимые функции из реализованных в ходе практики модулей, обрабатываются данные, производится обучение, оценка качества и построение графиков.

Пакет состоит из следующих частей:

– модуля analysis.py в котором прописаны метод считывания данных *read\_Data* из файлов формата .hdf5; конвейерный метод *Classifier* для классификаторов, позволяющий легко применять самые различные классификаторы, произвольно заданные пользователем. Конвейерный метод проводит обучение на тренировочных данных, предсказывает метки классов для тестовых данных, измеряет и возвращает метрику качества *f-1;* метод *Quantile* считает квантили для *f-1* метрик для каждой из 15 папок с тестовыми данными из разных распределений; метод *Q\_SNR\_plot,* который выводит графики зависимостей квантилей от SNR;

## – модуля chirpGen.py, в котором реализованы методы генерации вектора значений сигнала в зависимости от времени, генерации не ЛЧМ сигнала и метод зашумления данных;

– модуля DataGen.py, где происходит генерация тренировочной и тестовой выборки, с использованием методов из модуля chirpGen.py, а также вычисляется вектор значений , в зависимости от промежутка значения SNR и шага по этому промежутку. Все эти методы объединяются в конвейерный метод *DatasetGen*, в котором происходит запись наборов данных в соответствующие файлы формата .hdf5 .

Данные метрик по каждому классификатору записывается в файл quantile.txt и читаются графическим методом *Q\_SNR\_plot* с их последующим их выводом в виде графиков.

3.2.5 Результаты работы программы

Далее на графиках наглядно показаны квантили метрики качества для различных классификаторов в зависимости от изменения значения отношения сигнал/шум от 5 дБ

до -30 дБ.

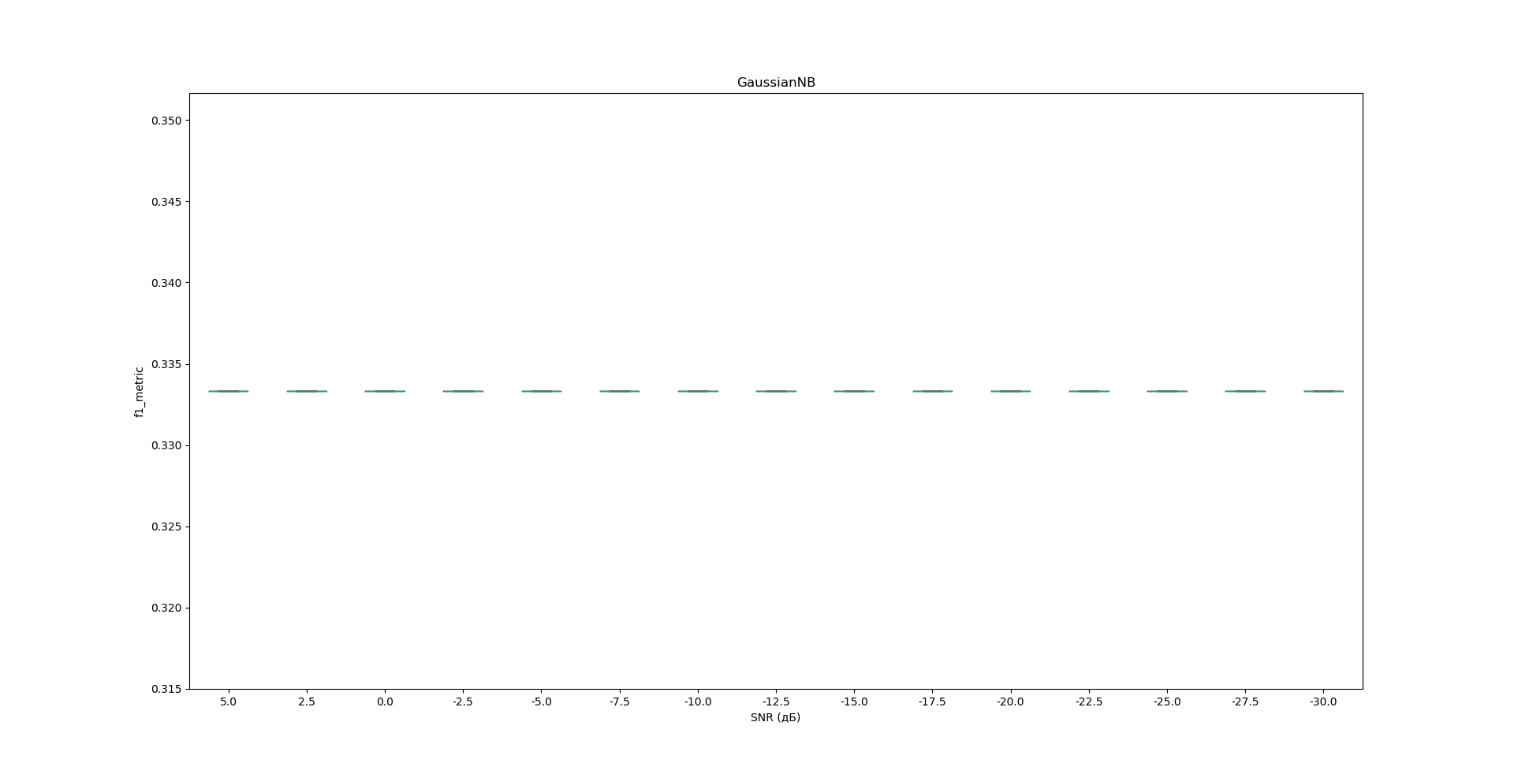


Рисунок 3.1 – Наивный Байес

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3.2 – Логистическая регрессия

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3.3 – Случайный лес

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3.4 – k-ближайшие соседи

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3.5 – Метод опорных векторов

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3.4 – Многослойный персептрон

На Рисунке 3.1 показатель метрики для классификатора никак не изменяется с изменением шума и соответствует величине, меньшей 0,5, из чего можем сделать вывод, что наивный Байесовский классификатор не подходит для работы с такого рода данными и очень плохо реагирует на зашумление данных.

На Рисунке 3.2 Показан результат классификатора на основе логистической регрессии. Классификатор показывает очень хорошие результаты при сравнительно малых шумах, но с увеличением шума быстро падает *f1-*метрика.

На Рисунке 3.3, Рисунке 3.4 и Рисунке 3.5наблюдается сравнительно хорошие показатели метрики при SNR = 5, дальше качество классификации SVM и k-ближайших соседей очень сильно падает, в то время как Случайный лес показывает большую устойчивость и на SNR = 2,5 все еще работает с метрикой чуть больше 0,5.

Лучше всех себя показывает классификатор на основе нейронной сети, что наблюдается на Рисунке 3.6. Метрика *f1* показывает почти 100% качество при малых шумах, как и Логистическая регрессия, но в отличие от регрессии нейронная сеть больше устойчива к шумам и при отношении сигнал/шум = 2,5 дает качество близкое к 0,7, после чего, как и все остальные модели, качество сильно падает.

Из наблюдаемых результатов можно прийти к выводу, что при дальнейшей настройке параметров для классификаторов на основе Случайного леса и MLP можно сильно улучшить показатели метрики, при том, что, в отличие от классических алгоритмов машинного обучения, нейронная сеть показала и хороший результат на малых шумах, и сравнительно хорошую устойчивость к большим шумам. Исходя из вышесказанного, представляется оптимальным работа над улучшением архитектуры нейронной сети.

ЗАКЛЮЧЕНИЕ

В результате прохождения производственной практики:

- ознакомлены с реальным производственным процессом на предприятии и участии в нём;

- выполнено индивидуальное задание.

Выполнены следующие задачи учебной практики:

- в лаборатории ИПУ РАН решена задача и проведена исследовательская работа по анализу полученных результатов;

- изучены понятия и природа ЛЧМ-сигналов, методы зашумления сигналов и оценки качества сигнала;

- изучены этапы исследовательской работы, по решению задачи классификации методами машинного и глубокого обучения;

- выполнено индивидуальное задания, которое включает в себя генерацию данных, зашумление тестовых данных, разработку программных модулей для решения задачи многоклассовой классификации сигналов и для оценки качества моделей классификации;

- изучен ГОСТ 7.32-2001 «Отчёт о научно-исследовательской работе. Структура и правила оформления».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 [https://ru.wikipedia.org/wiki/Институт проблем управления им. В.А.Трапезникова](https://ru.wikipedia.org/wiki/Институт%20проблем%20управления%20им.%20В.А.Трапезникова_) РАН

2 https://www.ipu.ru

3 https://www.ipu.ru/press-center/42869

4 Тараканов А.Н., Приоров А.Л. Сигналы в радиотехнических и телекоммуникационных системах: Учебное пособие. Л.: ЛРИД, 2001

5 Alexey Nefedov. [Support Vector Machines: A Simple Tutorial](https://svmtutorial.online/). — 2016.

6 Джулли А., Пал С. Библиотека Keras – инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlpw / пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2018.

7 Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron (2016). [Deep Learning](http://www.deeplearningbook.org/),