Некоммерческое акционерное общество «Казахский национальный исследовательский технический университет имени К.И.Сатпаева»

УДК 621.391.8: 004.932

На правах рукописи

НУРЛАНКЫЗЫ АЙГУЛЬ

Разработка интеллектуального метода детектирования речевого сигнала при низком отношении сигнал/шум

6D071900 – Радиотехника, электроника и телекоммуникации

Диссертация на соискание ученой степени
доктора философии (PhD)

Научные консультанты:
PhD, ассоц. проф. Евразийского национального университета имени Л.М. Гумилева
Медетов Б.Ж.

Зарубежный научный консультант:
д.э.н., к.т.н., профессор МГТУ имени Н.Э. Баумана, РФ
Тихвинский В.О.

Республика Казахстан
Алматы, 2025

СОДЕРЖАНИЕ

НОРМАТИВНЫЕ ССЫЛКИ

В данной диссертационной работе были использованы следующие нормативные документы:

Инструкция по оформлению диссертации и автореферата, утвержденная приказом Председателя ВАК МОН РК от 28 сентября 2004 г. № 377-3ж.

Положение о диссертационном совете НАО «КазНИТУ имени К.И.Сатпаева». П.029-03-04-02.2.02 – 2024;

ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание. Общие требования и правила оформления

ГОСТ 7.32-2017. Отчет о научно-исследовательской работе. Структура и правила оформления

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

VAD - детектор голосовой активности
CNN - сверточная нейронная сеть
RNN - рекуррентная нейронная сеть
MLP - многослойный перцептрон
TDNN - нейронная сеть с временной задержкой
LSTM - долговременная краткосрочная память
BiLSTM - двунаправленная долговременная краткосрочная память
GRU - рекуррентная нейронная сеть с управляемым блоком
BiGRU - двунаправленная рекуррентная нейронная сеть с управляемым блоком
GMM - смесь гауссовских моделей
HMM - скрытая марковская модель
SVM - метод опорных векторов
КРК – корпус казахского языка
ISSAI – Институт умных систем и искусственного интеллекта
FFT - Быстрое преобразование Фурье

ВВЕДЕНИЕ

Актуальность темы и научная проблема исследования. В современном мире все больше возрастает потребность в развитии технологий, способных обрабатывать и анализировать аудиосигналы в реальном времени. Поэтому в настоящее время обработка и детектирование речевого сигнала являются важным направлением применения технологий искусственного интеллекта и машинного обучения (Machine learning). Так, в соответствии со Стратегическим планом развития Республики Казахстан до 2025 года, утвержденным Указом Президента Республики Казахстан от 15 февраля 2018 года № 636, среди приоритетных задач стоит расширение покрытия сетей связи и развитие информационно-коммуникационной инфраструктуры. Это имеет огромное значение для обеспечения доступа к качественным телекоммуникационным услугам населению, поддержания конкурентоспособности страны в цифровой сфере способствует улучшению качества жизни граждан.

Также в соответствии с Национальным проектом «Технологический рывок за счет цифровизации, науки и инноваций» на 2021 – 2025 г., утвержденной Постановлением Правительства Республики Казахстан от 12 октября 2021 г. № 727, одним из основных направлений развития является цифровая трансформация. Главным фактором цифровой трансформации является применение технологий и сетей Интернет вещей (Internet of Things, IoT). Реализация сетей IoT предусматривает возможность поддержки различного рода датчиков, обеспечивающих сбор разнообразной информации для управления, выполнения информационных или производственно-технологических процессов современных отраслей экономики Республики Казахстан.

Кроме этого, создание «умных» городов также является одним из основных направлений развития. Умные города представляют собой интегрированные городские агломерационные системы, которые используют передовые телекоммуникационные технологии для оптимизации городской инфраструктуры. Тем самым повышая безопасность населения, улучшая экологическую среду и увеличивая эффективность городского управления. Развитие умных городов является важным шагом в направлении создания современной, инновационной и устойчивой городской среды.

Для успешной реализации концепции «умных» городов будет также необходимо активно развивать умные речевые приложения, которые будут использоваться жителями и городскими службами для повышения комфорта и эффективности жизни в городе. Особое внимание следует уделить разработке новых методов детектирования речевого сигнала при низком отношении сигнал/шум (С/Ш). Технологии распознавания речевого сигнала способствуют удобству взаимодействия с устройствами и приложениями, что является важным элементом в создании умных городов и современной информационно-коммуникационной инфраструктуры. Развитие и совершенствование таких технологий будет способствовать более эффективному функционированию умных городов, обеспечивая комфорт для населения.

Тема данной диссертации имеет огромное значение для решения задач, связанных с применением речевых сигналов в различных телекоммуникационных приложениях и сетях с их последующей акустической обработкой. В условиях современного информационного общества миллионы людей ежедневно полагаются на голосовые коммуникации посредством мобильных сетей и интернета. Технологии передачи и обработки речевой информации такие как Voice over IP (VoIP), Voice Over LTE (VoLTE), Voice Over Wi-Fi (VoWi-Fi) и IMS-based VoiceServices (VoIMS), работающие на платформах IMS, играют особую роль в обеспечении непрерывной и качественной голосовой связи и различных приложений. Также системы передачи голоса по пакетам или голоса по IP (VoIP) должны обеспечивать, чтобы качество передачи голоса существенно не снижало скорость передачи данных из-за таких сетевых условий, как потеря пакетов и задержки.

Тем не менее, существует множество факторов, которые могут усложнить процесс обмена информацией через речевые каналы. Одним из основных препятствий является низкое отношение С/Ш. Этот параметр часто становится критическим, особенно в условиях нестабильного соединения, больших городских агломерациях с массивным числом пользователей или в удаленных районах с ограниченной инфраструктурой.

Детектирование, распознавание и обработка речевого сигнала имеют широкий спектр применений, включая различные системы телекоммуникации, включая управление умными абонентскими устройствами, управление перегрузками в сети и многое другое. Однако при приеме и записи аудиосигналов в реальных условиях для распознавания речевого сигнала часто возникает проблема, связанная с наличием различного рода помех, которые складываются с существующим уровнем шумовых помех и воздействуют на входной сигнал, что в свою очередь приводит к снижению отношения С/Ш. Низкие значения отношения С/Ш является распространенной проблемой, которая может серьезно затруднить точное детектирование и распознавание сигналов речевой активности в сложных условиях. Фоновые шумы и помехи могут значительно повлиять на качество работы системы VAD (Voice Activity Detection). Это делает необходимым разработку усовершенствованных методов и алгоритмов, способных обеспечить надежное и точное определение моментов наступления активности речевого сигнала при неблагоприятных условиях, вызванных воздействием суммарного уровня помех Ш=Швн+Шпом.

В рамках данной диссертационной работы для эффективного детектирования и обработки речевых сигналов необходимо было разработать интеллектуальный метод детектирования речевого сигнала, использующий комбинацию нейронных сетей таких как CNN (Convolutional neural network) и RNN (Recurrent neural network), который будет работать эффективно в неблагоприятных условиях, определяемых высоким суммарным уровнем шума и помех, следовательно при низком отношении С/Ш. Поэтому для решения задачи детектирования потока данных речевого сигнала является наиболее подходящей система VAD, которая позволяет эффективно детектировать сигналы на фоне высокого уровня суммарного шума и помех. Данная система

предназначена для определение наличия или отсутствия речевого сигнала в детектируемом потоке данных. Однако при детектировании могут возникнуть серьезные проблемы, связанные с фоновым шумом, акустическими искажениями и тд. Это все в дальнейшем может привести к снижению отношения С/Ш, что значительно может усложнить работу системы VAD.

Также стоит отметить, что система VAD является важным компонентом системы распознавания речи ASR (Automatic Speech Recognition), поскольку позволяет определять и выделять в потоке сигналов данные содержащие речевые фрагменты. Правильная и безошибочная работа системы ASR является важным фактором для улучшения качества и точности распознавания и детектирования речевого сигнала в реальных условиях работы.

Актуальность разработки интеллектуального метода детектирования VAD, которая может работать даже в неблагоприятных условиях связана со следующим рядом факторов:

- использование речевых технологий становится все более распространенным в различных устройствах и сетях, таких как мобильные устройства, умные дома и другие;
- в реальных условиях часто возникают проблемы, связанные с точностью распознавания голосовых команд из-за наличия фоновых шумов и помех;
- разнообразие языков, акцентов и стилей речи требует адаптивных методов, способных эффективно работать с различными вариантами речевого сигнала;
- важно учитывать пользователей с ограниченными возможностями, такими как слабослышащие и люди с особенностями речи, чтобы технологии были доступными для всех;
- в некоторых случаях (например, в банковских системах или системах безопасности) требуется надежное распознавание голосовых команд и аутентификация пользователей.

Таким образом, разработка интеллектуального метода детектирования речевого сигнала при низком отношении С/Ш приобретает особую значимость, поскольку позволяет эффективно выявлять моменты активности речевого сигнала в сигнальном потоке, игнорируя фоновые шумы и помехи. Такой подход не только повышает эффективность голосового управления, но и способствует улучшению качества распознавания и детектирования речевого сигнала, обеспечивая более надежное взаимодействие с голосовыми технологиями. Это особенно актуально в условиях современной мобильности и повсеместного использования голосовых устройств управления, где низкое отношение С/Ш может значительно затруднять коммуникацию и обработку речевых данных.

Несмотря на значительный уровень проработанности данной научной проблемы, все еще существует необходимость в дальнейших исследованиях и разработках, направленных на повышение эффективности и точности детектирования речевых сигналов в неблагоприятных условиях. В современных условиях развитие интеллектуальных методов детектирования речевого сигнала, основанных не технологиях искусственного интеллекта, могут существенно

способствовать решению этой проблемы, делая такие системы более надежными и эффективными.

Научная проблема, решаемая в данной диссертационной работе, заключается в необходимости повышения устойчивости и точности детектирования речевого сигнала в условиях низкого отношения С/Ш. Существующие VAD-модели зачастую демонстрируют ограниченную обобщающую способность при изменении уровня шума, особенно если обучение проводилось на фиксированном значении С/Ш. Поэтому требуется исследование и разработка архитектур нейронных сетей, способных сохранять высокую точность распознавания в широком диапазоне акустических условий. В частности, необходимо определить, какие комбинации CNN, RNN (GRU, LSTM, BiGRU, BiLSTM) и TDNN обеспечивают наилучшее качество распознавания речевого сигнала.

Результаты исследования, представленные в данной диссертации, обладают высокой значимостью по нескольким причинам. Во-первых, установлено, что обучение нейросетевых моделей на аудиоданных с широким диапазоном значений отношения С/Ш способствует существенному повышению точности и устойчивости систем детектирования речевой активности в условиях переменной зашумленности, характерной для реальной акустической среды. Во-вторых, показана высокая эффективность гибридных архитектур, таких как CNN+BiGRU и CNN+BiLSTM, способных обеспечивать стабильную работу и высокую обобщающую способность даже при неблагоприятных уровнях отношения С/Ш. В-третьих, разработанная в рамках диссертации виртуальная лабораторно-исследовательская работа позволяет интегрировать полученные научные результаты в образовательный процесс, предоставляя студентам возможность практического освоения современных методов глубокого обучения и анализа речевых сигналов.

Также необходимо подчеркнуть, что разработка интеллектуального метода детектирования голосовых команд на разных языках с использованием нейронных сетей, обученных на казахском языке с использованием ограниченного числа дикторов, является актуальной и важной задачей с научной точки зрения. Вопрос межъязыкового детектирования и распознавания речевого сигнала с помощью нейронных сетей, обученных исключительно на казахском языке, представляет собой относительно новую и малоисследованную научную проблему. В этой области достижения пока ограничены, а исследования, посвященные детектированию и распознаванию речевого сигнала с применением нейронных сетей, все еще недостаточно распространены и недостаточно комплексны.

Перспективы и потенциал данной области весьма значительны, однако для полноценного раскрытия возможностей этого направления требуются дальнейшие научные исследования и разработки. В настоящее время существующие исследования и технологии в области межъязыкового детектирования и распознавания речевого сигнала преимущественно основываются на классических методах обработки речевого сигнала. И хотя нейронные сети широко применяются для распознавания и детектирования

речевого сигнала на других языках, их использование для казахского языка остается вызовом и открытой научной проблемой.

Дальнейшее развитие данной области требует углубленных исследований, с учетом специфики казахского языка и его особенностей. Таким образом, несмотря на наличие определенных исследований и примеров применения нейронных сетей в области межъязыкового распознавания речевого сигнала, степень разработанности этой научной проблемы остается относительно низкой и требует дальнейшего изучения. В условиях глобализации и возрастающей необходимости в мультиязычных технологиях важно подчеркнуть значимость создания таких систем с поддержкой нескольких языков. Эффективное детектирование и распознавание речевого сигнала на различных языках существенно упрощает доступ пользователей и повышает удобство взаимодействия с различными техническими устройствами и программным обеспечением. Обучение нейронных сетей на ограниченном количестве дикторов, и их адаптация для межъязыкового распознавания и детектирования представляет собой важную сторону разработки универсальных методов и алгоритмов для работы с различными языками. Полученные результаты могут способствовать разработке точных и надежных систем распознавания голосовых команд на многоязычной основе. Это улучшить современные технологий и повысить качество межкультурного взаимодействия в современном информационном обществе.

Степень разработанности научной проблемы. Вопросами разработки интеллектуального метода детектирования и распознавания речевого сигнала занимались такие ученые как: Rabiner L., Jelinek F., Baker J., Srinivasan S., Young S., Allen J., Nakanishi H., Жданов А., Бондаренко И., Баянов А., Козырев А., Кудрявцева Т., Никулин С., Козлов М., Смирнова Е., Петров Д., Иванова О., Соколов В., Федорова Н., Кузнецов И., Калимолдаев М., Амиргалиев Е., Мусаходжаева С., Мамырбаев О. и другие

Тем не менее, несмотря на то что в ряде научных работ рассматриваются различные интеллектуальные подходы к распознаванию и детектированию речевого сигнала, в том числе на казахском языке, большинство из них ограничиваются фрагментарным анализом и не затрагивают комплексную оценку производительности современных гибридных нейросетевых архитектур в условиях сильной акустической зашумленности. В частности, остается открытым вопрос, какие именно комбинации сверточных и рекуррентных слоев, таких как CNN+BiGRU, CNN+BiLSTM, CNN+TDNN и другие, демонстрируют наилучшую устойчивость к низкому отношению С/Ш при решении задачи детектирования речевого сигнала. Это подчеркивает актуальность дальнейших исследований, направленных на анализ, сравнение и оптимизацию гибридных моделей, способных эффективно обрабатывать зашумленные речевые сигналы в реальных акустических средах.

Таким образом, научная проблема детектирования и распознавания речевой активности при низком отношении С/Ш представляет собой перспективное направление исследований, требующее дальнейшего развития и

углубленного изучения для создания эффективных методов детектирования и распознавания речевого сигнала в условиях повышенного шума и помех.

Объектом исследования является процесс детектирования речевого сигнала в акустически зашумленных условиях с использованием гибридных нейросетевых архитектур, сочетающих сверточные и рекуррентные слои, обученных на казахском речевом корпусе.

Предметом исследования является архитектурные особенности, алгоритмические подходы и параметры обучения гибридных нейросетевых моделей, обеспечивающих устойчивое и точное детектирование речевого сигнала при различных уровнях отношения С/Ш и ее имитационная модель для выполнения лабораторно-исследовательских работ.

Цель и задачи диссертации.

Целью диссертации является повышение точности и устойчивости детектирования речевого сигнала в условиях различного уровня акустического шума за счет разработки, обучения и сравнительного анализа гибридных нейросетевых моделей, обученных на казахском речевом корпусе, с последующей оценкой их эффективности при распознавании речевого сигнала на различных языках.

Для достижения поставленной цели требуется решить следующие задачи:

- проанализировать влияние современных методов VAD, основанных на использовании нейросетевых подходов, на точность и устойчивость обнаружения речевого сигнала в условиях различных значений отношения сигнал/шум, проведя серию экспериментов по их тестированию;
- исследовать эффективность гибридных нейросетевых архитектур, обученных на казахском языке, в задаче детектирования речевого сигнала при низком отношении С/Ш, с последующей проверкой их межъязыкового применения;
- разработать и реализовать алгоритм детектирования речевого сигнала, учитывающий разные уровни отношения С/Ш, с использованием и сравнением различных типов нейросетей, включая их гибридные модификации;
- создать виртуальную лабораторно-исследовательскую работу, ориентированную на визуализацию, тестирование и сравнительный анализ различных архитектур нейросетей в задаче распознавания речевого сигнала.

Методология и методы исследования. Для решения поставленных задач в данной диссертации использовались аналитические методы, методы математического и компьютерного моделирования.

Методология и методы исследования включали в себя, изучение существующих методов распознавания и детектирования речевого сигнала на различных языках с учетом особенностей казахского языка. Нейронные сети были обучены на казахском языке для распознавания речевого сигнала, после чего было проведено тестирование на высказываниях на других языках с целью оценки их эффективности. Для сравнения результатов были проведены аналитические исследования, а также использованы математическое и компьютерное моделирование для сравнения работы нейронных сетей.

Научная новизна диссертации заключается в разработке и экспериментальной оценке интеллектуального метода детектирования речевого сигнала при низком отношении С/Ш, основанного на использовании гибридных нейросетевых архитектур, обученных на казахском языке, а также разработка ее имитационной модели для выполнения лабораторно-исследовательских работ.

Практическая значимость работы подтверждается:

- актом внедрения в учебный процесс кафедры «Радиотехника, электроника и телекоммуникации» Международного университета информационных технологий по дисциплине «Системы мобильной связи».

Основные положения, выносимые на защиту. На защиту выносятся следующие положения:

- обоснована эффективность гибридных нейросетевых архитектур для задачи детектирования речевого сигнала, обеспечивающих высокую точность даже в условиях низкого отношения С/Ш;
- разработан интеллектуальный метод VAD, адаптированный для работы в широком диапазоне низких значений отношения С/Ш, что подтверждено результатами экспериментальной оценки с использованием гибридных нейросетевых архитектур;
- подтверждена возможность межъязыкового применения моделей, обученных на казахском языке, при распознавании речевого сигнала на других языках;
- разработана виртуальная модель для лабораторных исследований моделей нейронных сетей, применяемых в задаче распознавания речевого сигнала.

Апробация работы. Основные идеи и результаты данной диссертационной работы были представлены и обсуждены на научных семинарах, где происходило обсуждение основных элементов и выводов исследования, а также получена обратная связь касательно содержания диссертации от профессорско-преподавательского состава следующих кафедр:

- кафедра «Электроника, телекоммуникация и космические технологии» КазНИТУ им. К. И. Сатпаева (2018 – 2021 гг.);
- кафедра «Космическая инженерия» Алматинского университета энергетики и связи им. Г. Даукеева (2022 – 2023 гг.).

Публикации по теме диссертации. В рамках данной диссертации были опубликованы 2 статьи в международном журнале Eastern-European Journal of Enterprise Technologies, который имеет показатель процентиля CiteScore -46 и проиндексирован в базе данных Scopus.

1. Medetov, B., Zhetpisbayeva, A., Akhmediyarova, A., Nurlankyzy, A., Namazbayev, T., Kulakayeva, A., Albanbay, N., Turdalyuly, M., Yskak, A., & Uristimbek, G. (2025). Evaluating the effectiveness of a voice activity detector based on various neural networks. *Eastern-European Journal of Enterprise Technologies*, *1*(5 (133), 19–28. https://doi.org/10.15587/1729-4061.2025.321659

2. Nurlankyzy A., Akhmediyarova A., Zhetpisbayeva A., Namazbayev T., Yskak A., Yerzhan N., Medetov B. The dependence of the effectiveness of neural networks for recognizing human voice on language (2024) Eastern-European Journal

of Enterprise Technologies, 1 (9(127)), pp. 72 - 81, https://doi.org/10.15587/1729-4061.2024.298687

Кроме того, также опубликовано 4 статьи в журналах, рекомендуемых Комитетом по обеспечению качества в сфере образования и науки Министерства образования и науки Республики Казахстан.

3. Бекболат Медетов, Айгуль Нурланкызы, Айгуль Кулакаева, Айнур Жетписбаева, Тимур Намазбаев. Оценка влияния языка на точность распознавания человеческого голоса с помощью искусственных нейронных сетей. Том 131 № 2 (2024): Вестник КазАТК, 456-466

4. А.Т. Ахмедиярова, А. Нурланкызы, А.Е. Кулакаева, Б.Ж. Медетов. Анализ эффективности нейронных сетей по распознаванию человеческого голоса. Том 1 № 64 (2024): Вестник АУЭС, 37-46

5. Медетов, Б., Нурланкызы, А., Ахмедиярова, А., Жетписбаева, А. и Жексебай, Д. 2024. Сравнительный анализ эффективности нейронных сетей при низком значении отношения С/Ш. *Известия НАН РК. Серия физико-математическая*. 4 (дек. 2024), 163–173. DOI:https://doi.org/10.32014/2024.2518-1726.315.

6. Kulakayeva, A., Tikhvinskiy, V., Nurlankyzy, A., & Namazbayev, T. (2024). Comparative analysis of the effectiveness of neural networks at different values of the SNR ratio. *Scientific Journal of Astana IT University*, *20*, 18–30. https://doi.org/10.37943/20TTRV6747

Также получено свидетельство о внесении сведений в государственный реестр прав на объекты, охраняемые авторским правом № 48570 от 24 июля 2024 года. Название объекта «Виртуальная лабораторно-исследовательская работа «Оценка производительности нейронных сетей в задаче распознавания речевого сигнала». Вид объекта авторского права: программа для ЭВМ

Личный вклад автора. Данная диссертация является самостоятельной работой Нурланкызы А. Все исследования, связанные с разработкой интеллектуального метода детектирования речевого сигнала при низком отношении С/Ш разработана и проведена самостоятельно докторантом. Вместе с тем, формулирование задач исследований, выбор методов решения, а также анализ результатов исследований осуществлялись при активном сотрудничестве с отечественным научным руководителем и зарубежным научным консультантом. Совместная работа способствовала глубокому анализу проблемы, а также выбору оптимального подхода для разработки интеллектуального метода, позволяющего детектировать речевой сигнал в условиях различных шумов и помех.

Нурланкызы А. является по данному направлению исследования обладателем гранта «Жас Ғалым», научным руководителем проекта № AP22684173 на тему «Разработка высокоэффективного нейросетевого метода обнаружения голосовой активности при низком уровне отношения сигнал/шум», финансируемого МНВО РК.

Кроме того, Нурланкызы А является исполнителем проекта грантового финансирования КН МНВО РК на 2023-2025 гг. AP19678995 на тему «Разработка метода распознавания дикторов с применением глубоких

нейронных сетей при ультракороткой продолжительности чистой речи» (договор № 344 от 03.08.2023 г.)

Структура и объем диссертации. Диссертация включает введение, четыре главы, заключение, список использованных источников с полными наименованиями. Общий объем работы составляет 134 страниц, включая 78 рисунков, 9 таблиц и 4 приложений.

1 АНАЛИЗ МЕТОДОВ ДЕТЕКТИРОВАНИЯ ГОЛОСОВОЙ АКТИВНОСТИ ДЛЯ СИСТЕМ РАСПОЗНАВАНИЯ РЕЧЕВОГО СИГНАЛА

1.1 Обоснование необходимости нейросетевого VAD в системе распознания речевого сигнала

Разговорный язык представляет собой одно из основных средств коммуникации для людей, позволяя передавать высокоуровневые концепции и идеи. Речь, являясь конкретным проявлением устной коммуникации, физически воспроизводится голосовыми органами человека. Она содержит множество сложных факторов, включая передачу сообщения, идентификацию личности говорящего, а также его физическое и эмоциональное состояние. Исследования в области автоматической обработки сложной информации, представленной в виде речи, стали актуальной темой в научных кругах [1].

Современные технологии активно развиваются в направлении обработки и анализа речи, позволяя создавать разнообразные системы, выполняющие специфические задачи. Одной из таких систем является автоматическое распознавание речи (ASR, Automatic Speech Recognition), которое предназначено для преобразования устной речи в текст. Существуют также системы идентификации говорящих (SR, Speaker Recognition), которые позволяют идентифицировать личность по голосу, используя уникальные акустические характеристики. Кроме этого, ведутся разработки систем распознавания эмоциональных состояний (SER, Speech Emotion Recognition), которые могут определить настроение, эмоции или уровень стресса собеседника [2].

В последние годы наблюдается значительное ускорение развитии автоматического распознавания речи, что связано с увеличением вычислительных мощностей и достижениями в области искусственного интеллекта. Данные инновации постепенно изменяют взаимодействие между человеком и машиной, переходя от прикосновений к голосовому управлению, что значительно упрощает жизнь людей, подчеркивая важность речевых технологий [3].

Современные системы ASR достигли значительных успехов за последние десятилетия. Несмотря на широкое распространение и разнообразие используемых систем, полное распознавание речи остается нерешенной задачей [4]. Данная проблема обусловлено множеством факторов, таких как шумы, искажения при передаче, акценты, а также различия в темпе и манере речи. Данные факторы могут значительно влиять на точность и надежность систем распознавания речи, создавая существенные вызовы для разработчиков и исследователей в данной области.

В последние годы исследования в области распознавания речи значительно продвинулись благодаря достижениям в области вычислительных технологий и искусственного интеллекта. Существует обширная научная литература, посвященная различным элементам и методам данной темы. На рисунке 1.1 представлена общая структура современных систем ASR, которая включает

несколько главных компонентов. К ним относятся выделение векторов признаков из входной речи, акустическое моделирование, языковое моделирование и декодирование последовательности слов.

Процесс распознавания речи начинается с выделения векторов признаков из аудиосигнала. На следующем этапе осуществляется акустическое моделирование, которое определяет, какие звуки были произнесены. Далее, с помощью языкового моделирования, проверяется соответствие этих звуков вероятным последовательностям слов. На заключительном этапе происходит декодирование, то есть определение точной последовательности слов, произнесенных человеком.

Процесс современного метода распознавания речевого сигнала включает последовательные этапы:

1. Выделение признаков (Feature Extraction). На данном этапе входной аудиосигнал преобразуется в набор признаков, которые представляют акустические характеристики речи. Это может включать спектрограммы, мел-кепстральные коэффициенты (MFCC) и другие аудиофункции.

2. Акустическое моделирование (Acoustic Modeling). Данный компонент преобразует признаки в вероятности фонем или других элементарных звуков речи. Акустическая модель обучается на базе данных, содержащей аудиозаписи и их текстовые транскрипции.

3. Языковое моделирование (Language Modeling). Языковая модель оценивает частоту слов и их сочетаний, создавая вероятностные предсказания.

4. Декодирование (Decoding). Данный компонент объединяет данные от акустической и языковой моделей, чтобы определить наиболее вероятную последовательность слов.

5. Постобработка (Post-processing). Включает в себя такие задачи, как корректировка грамматических ошибок, пунктуация и улучшение текстового вывода для конечного пользователя.

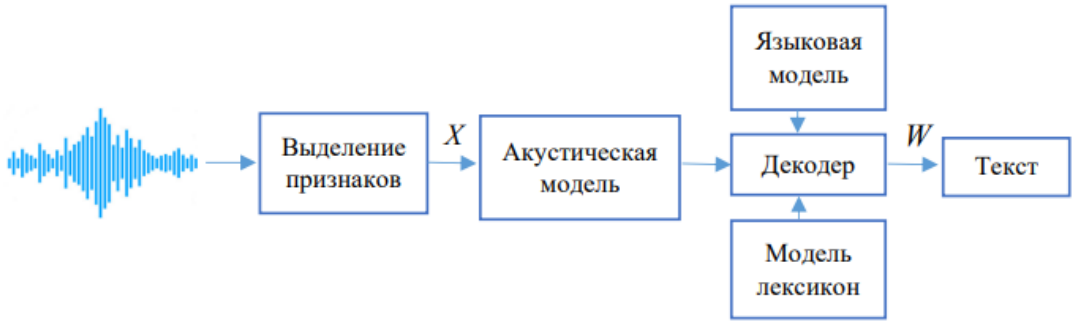


Рисунок 1.1 – Схема современной системы распознавания речевого сигнала [5, с. 16]

Речевые сигналы часто сопровождаются неречевыми звуками, вызванными внешними источниками шума. Искажения могут возникать по различным причинам, включая акустические особенности помещения, характеристики голоса говорящего, свойства используемых устройств и условия

записи. Данные факторы, а также фильтрация и передача сигнала через каналы связи, усложняют обработку и снижают точность автоматических систем распознавания речевого сигнала.

Посторонние звуки и искажения, которые называются шумами, могут существенно повлиять на качество распознавания речи. Наличие, типы и уровень шумов напрямую воздействуют на точность работы систем. Например, системы, хорошо справляющиеся с чистыми или слабо зашумленными данными, могут оказаться неэффективными в условиях значительного шума. Более того, системы, обученные на одном типе шумов, например, характерных для офисной среды, могут сталкиваться с трудностями при распознавании речи в иных условиях, таких как шумная обстановка в автомобиле [6].

При создании и проверке систем распознавания речевого сигнала важно учитывать присутствие, типы и уровень шумов в аудиозаписях. Данные факторы могут значительно влиять на точность распознавания, и их игнорирование может существенно снизить эффективность системы.

Одним из основных этапов, обеспечивающих успешное распознавание речи, является процесс извлечения признаков. Этот этап является критически важным, так как он позволяет выделить информативные элементы из речевого сигнала, несмотря на присутствие шумов, и подготовить данные для дальнейшего анализа и обработки системой распознавания [7].

Для уменьшения влияния неречевых фрагментов на системы обработки речевого сигнала предполагается, что входной аудиосигнал содержит только человеческую речь. Для этого используется VAD (Voice activity detection), также известное как детектирование речевой активности или распознавание речи. VAD представляет собой процесс идентификации присутствия или отсутствия человеческой речи в аудиосигнале. Этот этап является важным для отделения речевых сегментов от шумовых и неречевых фрагментов, что помогает улучшить качество обработки и распознавания речевого сигнала. На рисунке 1.2 представлена схема, показывающая работу VAD.

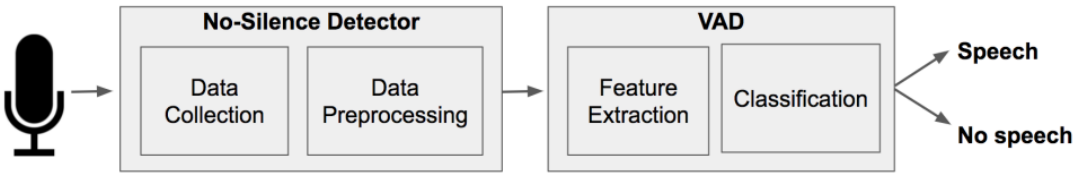


Рисунок 1.2 – Принцип работы системы VAD [8, с. 2]

Обнаружение речевой активности VAD обычно используется как механизм предварительной обработки, позволяющий активировать или деактивировать интерфейс обработки сигналов в нужные моменты. Использование системы обнаружения активности голоса VAD позволяет значительно повысить эффективность обработки речевых сигналов. Такая технология предотвращает ненужное кодирование и передачу пакетов с моментами молчания, что экономит вычислительные ресурсы и пропускную способность сети. VAD используется в

автоматическом распознавании речи, проверке личности по голосу и улучшении качества звука [9].

В системах голосовой связи применяется технология VAD, которая позволяет более эффективно использовать каналы передачи данных. Разделяя настоящую речь от фонового шума и тишины. Системы VAD позволяют передавать исключительно речевые данные, что значительно уменьшает объем информации, необходимый для передачи. В соответствии со стандартами для систем мобильной связи третьего поколения, голосовая активность занимает лишь около 40% от общей продолжительности вызова. Это означает, что значительная часть времени связи тратится на передачу неголосовых данных, таких как моменты тишины и фоновый шум. Сокращение количества закодированных битов в этих неголосовых сегментах благодаря VAD позволяет значительно снизить нагрузку на ресурсы беспроводной связи. Это, в свою очередь, ведет к более эффективному использованию доступной пропускной способности и экономии вычислительных ресурсов, что особенно важно в условиях ограниченного спектра и высокой нагрузки на сети [10].

Современные мобильные сети используют технологию VAD для идентификации неголосовых сегментов и передачи в них только комфортного шума. Этот режим, известный как прерывистая передача (DTX), способствует увеличению пропускной способности каналов. В алгоритме кодирования речи G.729, разработанном Международным союзом электросвязи, предусмотрено объединение VAD с DTX. По теоретическим расчетам, использование этого метода может повысить пропускную способность системы в 1,5 раза и увеличить число пользователей мобильной сети на 60% [3].

Извлеченные характеристики из аудиосигнала сравниваются с установленным пороговым значением, которое обычно определяется на основе анализа периодов с только шумом во входном сигнале. На основе этого сравнения принимается решение VAD. Проблему VAD можно описать следующим образом:

$$X \rightarrow Y \quad \text{где} \quad X \in R^d \quad \text{и} \quad Y \in \{0, 1\}, \tag{1.1}$$

где d - размер аудиокадра или его функциональное представление [11].

Базовый алгоритм обнаружения речевой активности VAD работает на основе извлечения признаков из входящего аудиосигнала. Этот сигнал разбивается на кадры длительностью от 5 до 40 мс для дальнейшего анализа.

Если характеристика входного кадра превышает рассчитанное пороговое значение, принимается решение VAD (Y=1), указывающее на наличие речевого сигнала. В противном случае принимается решение VAD (Y=0), означающее отсутствие речевого сигнала в данном кадре. Блок-схема базового алгоритма VAD представлена на рисунке 1.3.

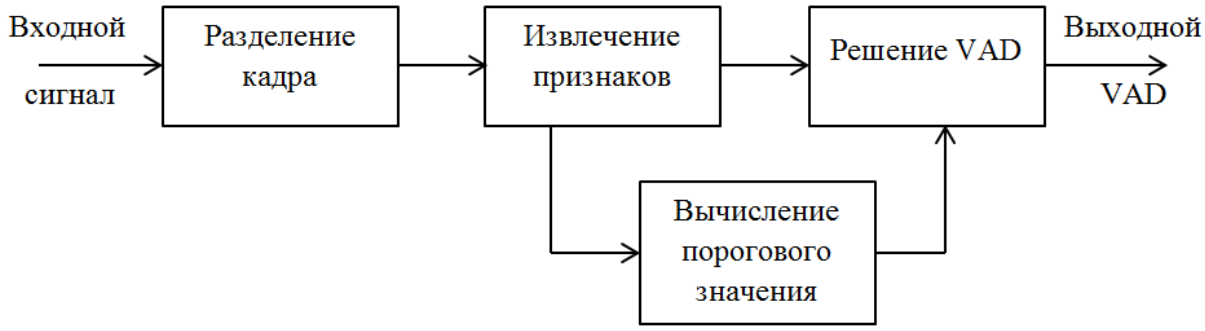


Рисунок 1.3 – Структурная схема VAD [12, с. 8]

Длительность каждого кадра, обозначаемая как длина кадра, представлена на рисунке 1.4. Перекрывающаяся область между двумя последовательными кадрами называется сдвигом кадров.

Для формирования кадра сигнала используется оконная функция $w(n)$, которая временно изменяется и сдвигается, таким образом, чтобы ее передний край соответствовал желаемому моменту времени $w(m-n)$. В результате кадр $f(n;m)$ сигнала $s(n)$, завершающийся в момент времени m, математически выражается следующим образом:

$$f_s(n;m) = s(n)w(m-n), \tag{1.2}$$

Таким образом, кадр сигнала $f_s(n;m)$ формируется согласно уравнению (1.2). Длина кадра и сдвиг кадра показаны на рисунке 1.4.

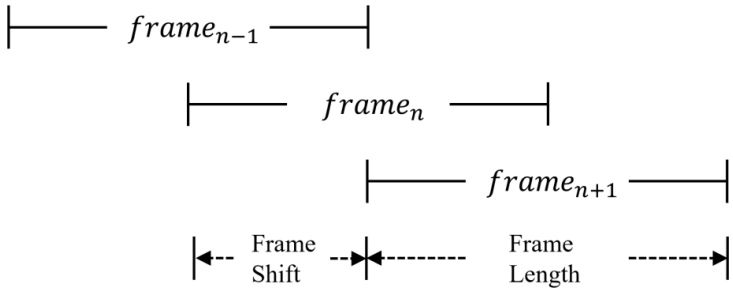


Рисунок 1.4 – Длина кадра и сдвиг кадра [3, с. 16]

В системах автоматического распознавания речи ASR точное обнаружение речевой активности VAD является основным элементом, который существенно влияет на эффективность и точность работы системы.

Правильная идентификация речевых и неречевых фрагментов позволяет не только сократить объем вычислений, но и уменьшить время обработки, исключая ненужные неречевые данные. Более того, VAD способствует повышению точности распознавания, устраняя шумовые компоненты, которые могут искажать результаты.

Результаты исследования в работе [13] демонстрируют, что задержка в обнаружении начальной точки речи всего на пять кадров приводит к снижению точности распознавания на 4,3%. Если задержка увеличивается до десяти кадров, точность падает уже на 10,8%. Аналогично, если конечная точка речи откладывается на пять кадров, точность снижается на 1,5%, а задержка на десять кадров приводит к снижению точности на 6,3%.

Важно отметить, что неправильное определение речи как шума VAD может значительно ухудшить результаты распознавания. В случаях, когда начальная точка определяется преждевременно или конечная точка обнаруживается с задержкой, вычислительная нагрузка на систему возрастает, что, в свою очередь, увеличивает время отклика. Такие неточности подчеркивают необходимость оптимизации и точной настройки VAD для обеспечения высокоэффективной и надежной работы систем ASR.

При разработке моделей для обнаружения речевой активности VAD следует учитывать два основных факторов: производительность и эффективность.

Производительность оценивает точность модели в предсказании речевых сегментов в аудиопотоке, тогда как эффективность указывает на объем вычислительных ресурсов, необходимых для выполнения таких задач.

С момента внедрения метода VAD было разработано множество различных подходов, основанных на анализе акустических характеристик. Такие методы используют параметры, такие как кратковременная энергия, частота пересечения нуля, двойной порог, кепстральные коэффициенты и спектральная энтропия, чтобы определить наличие речи в аудиосигнале. Они показывают высокую точность в тишине, но их производительность может заметно снижаться при высоком уровне шума и низком отношении С/Ш.

В таких ситуациях сложность заключается в том, что шум искажается и смешивается с речевым сигналом, что затрудняет точное определение моментов активности голоса. Несмотря на достигнутый прогресс, проблемы, связанные с низким отношением С/Ш и различными типами шумов, остаются актуальными, и требуют дальнейшего развития и совершенствования методов VAD для повышения их устойчивости и точности в разнообразных условиях окружающей среды.

В последнее время получили распространение модели VAD, основанные на методах машинного и глубокого обучения. Они демонстрируют высокую точность обнаружения даже в шумных средах. Тем не менее, высокая вычислительная нагрузка и большой размер таких моделей часто делают их непригодными для использования на устройствах с ограниченными вычислительными ресурсами [3].

На сегодняшний день не существует идеального метода VAD, который одновременно обеспечивал бы высокую точность обнаружения в условиях низкого уровня шума и минимальные вычислительные требования. Поэтому разработка эффективного метода VAD, способного достичь оптимального баланса между производительностью и вычислительной эффективностью, остается актуальной задачей для будущих исследований.

Основываясь на проведенном анализе, можно определить особые характеристики, которым должен соответствовать идеальный алгоритм VAD:

1. Для обеспечения эффективности работы в режиме реального времени важно, чтобы алгоритм VAD обладал низкой вычислительной сложностью и, соответственно, минимальными требованиями к потреблению энергии. Это особенно критично для устройств с ограниченными ресурсами.

2. Надежность алгоритма существенно возрастает, если он способен адаптироваться к изменяющимся условиям фонового шума, особенно в нестационарных акустических средах. Такая адаптивность позволяет VAD поддерживать высокую точность работы в разнообразных сценариях.

3. Для определения наличия или отсутствия речи алгоритм должен использовать физические характеристики входного аудиосигнала. Это включает анализ свойств кадров, что позволяет алгоритму выносить последовательные и точные решения при классификации, что является критически важным для обеспечения высококачественного распознавания речи.

Данные основные характеристики должны быть учтены при разработке эффективных и универсальных алгоритмов VAD, способных работать в различных условиях и на различных платформах [14].

1.2 Анализ существующих методов детектирования голосовой активности

Традиционные методы VAD основываются на анализе количественных звуковых характеристик, что позволяет отделять речевые сегменты от фонового шума. В таких методах звук, содержащий зашумленную речь, рассматривается как комбинация чистой речи и добавочного шума. Исходя из этого, цель VAD заключается в идентификации аудиосегментов, содержащих допустимую речь. Предполагается, что известны параметры зашумленного звука и аддитивного шума, что позволяет выделить речевые компоненты без значительных затруднений [15].

Например, при наличии информации об энергии речи и шума можно использовать метод вычитания энергии для определения сегментов, содержащих речь. Для различения речи и шума были разработаны различные звуковые характеристики. Большинство традиционных методов принимают решения, сравнивая значения этих характеристик с заранее установленными пороговыми значениями. Данные пороговые значения обычно определяются эмпирически. Данный подход обладает преимуществами, поскольку обеспечивает возможность обучения и оценки без необходимости использования сложных моделей [12].

В настоящее время представлено множество методов, разработанных для реализации VAD, которые можно классифицировать по следующим категориям:

1. Метод пересечения нуля (основан на анализе количества переходов сигнала через нулевую амплитуду).

2. Метод, основанный на использовании энергии (включает анализ энергии аудиосигнала для выявления речевых сегментов).

3. Метод линейного прогнозирования (использует модели линейного прогноза для выделения речевых участков).

4. Метод одночастотной фильтрации (осуществляет фильтрацию сигнала на определенной частоте для отделения речи от шума).

5. Метод нейронных сетей (применяет архитектуры искусственных нейронных сетей для обучения и распознавания речевых сегментов) [14].

1.2.1 Метод пересечения нуля

В 1975 году лаборатория Bell представила первый метод VAD, который опирался на использование кратковременной энергии и скорости пересечения нуля (ZCR, zero-crossing rate) как основных звуковых характеристик [16]. Данный метод основывался на предположении, что первые 100 мс аудиосигнала являются чистым фоновым шумом, что позволяло вычислить кратковременную энергию шума. Полученные данные использовались для установки пороговых значений энергии, необходимых для дальнейшего анализа. Кроме того, ZCR был использован для отслеживания того, как часто аудиосигнал пересекает нулевую линию, поскольку для беззвучных фрагментов речи характерно более высокое значение ZCR по сравнению с озвученными сегментами. Данная характеристика позволила повысить точность метода VAD за счет выявления низкоэнергетических неречевых деталей.

Метод включал двухуровневый механизм принятия решений, основанный на кратковременной энергии и ZCR, что способствовало более точному определению речевых сегментов. Однако, несмотря на его удовлетворительную работу в условиях минимального шума (при SNR 30 дБ и выше), реализация метода на практике сталкивается с рядом ограничений. Во-первых, метод предполагает, что аддитивный шум является стационарным, а первые 100 мс сигнала действительно содержат только шум. Во-вторых, считается, что речевые сегменты должны иметь более высокую кратковременную энергию, чем фрагменты, содержащие только шум. Такие предположения делают настройки пороговых значений неподходящими для всех условий прослушивания, что ограничивает универсальность метода.

Частота пересечения нуля – это количество случаев, когда последовательные значения речевого сигнала меняют знак или когда амплитуда сигнала проходит через ноль. Скорость пересечения нуля, обозначаемая как, Z_n рассчитывается следующим образом:

$$Z_n = \sum_{m=-\infty}^{\infty} |sgn[x(m)]| - |sgn[x(m-1)]| w(n-m), \quad (1.3)$$

$$sgn[x(m)] = \begin{Bmatrix} 1 & x(m) \geq 0 \\ -1 & x(m) < 0 \end{Bmatrix}, \quad (1.4)$$

$$w(n) = \begin{cases} \dfrac{1}{2N}, 0 \leq n \leq N-1 \\ 0 \ , \ \text{в остальных случаях} \end{cases}, \quad (1.5)$$

где N - продолжительность окна, используемого в методе [17].

Таким образом, скорость пересечения нуля Z_n рассчитывается в соответствии с уравнением (1.3). Также функция знака $sgn[x(m)]$ определяется уравнением (1.4), а весовая функция $w(n)$ задается уравнением (1.5).

Частота пересечения нуля является важным показателем для определения наличия речи в аудиосигнале. Данный параметр указывает, сколько раз последовательные отсчеты в сигнале изменяют свой алгебраический знак или пересекают нулевую амплитуду. Высокая частота пересечения нулей, как правило, свидетельствует о том, что кадр не содержит вокализованной речи, тогда как низкая частота указывает на наличие озвученного кадра.

Метод определения скорости пересечения нуля (ZCR) использует частотные свойства сигнала для формирования правила принятия решений в VAD. Данный метод базируется на предположении, что голосовые компоненты сигнала преимущественно находятся в низкочастотной области, в то время как шумовые компоненты располагаются в высокочастотной. Следовательно, если количество пересечений нуля $Z(n)$ низкое, сегмент классифицируется как содержащий голосовые данные. В противном случае, при высоком значении $Z(n)$, сегмент классифицируется как шумовой.

1.2.2 Методы и подходы на основе энергетических расчетов

Методы VAD, основанные на расчете энергии, являются одними из самых популярных и широко используемых в области обработки речи. Они привлекают внимание благодаря своей простоте реализации и низкой вычислительной сложности, что делает их особенно подходящими для различных практических приложений, включая мобильные устройства и системы с ограниченными ресурсами [18-20]. Основной принцип таких методов заключается в анализе уровней энергии речевого сигнала: при наличии речи уровень энергии, как правило, выше, чем в моменты тишины или присутствия неречевых шумов. Это позволяет эффективно различать активные речевые сегменты и периоды молчания, что делает такие методы весьма полезными для широкого спектра задач, таких как сжатие данных, улучшение качества передачи речи и другие. Несмотря на свои достоинства, методы, основанные на расчете энергии, могут сталкиваться с трудностями в условиях высоких уровней шума, что требует дополнительного использования адаптивных фильтров и других техник для повышения точности обнаружения речевой активности.

Существует множество способов представления энергии сигнала, каждый из которых может быть применен для анализа и классификации различных аудиосегментов. Данные представления играют основную роль в определении и различении речевых и неречевых компонентов сигнала.

$$E_n = \sum_{k=1}^{K} \log[x^2(k,n)]\,, \qquad (1.6)$$

$$E_n = \sum_{k=1}^{K} x^2(k,n)\,, \tag{1.7}$$

$$E_n = \sum_{k=1}^{K} |x^2(k,n)|\,, \tag{1.8}$$

где K обозначает общее количество отсчетов в кадре, n представляет текущий кадр, а $x(\cdot)$ – символизирует аудиосигнал.

Уравнение (1.6) характеризует логарифмическую кратковременную энергию, уравнение (1.7) – это кратковременная энергия, возведенная в квадрат, а уравнение (1.8) представляет собой абсолютное значение кратковременной энергии. Все эти величины являются примерами энергии, охватывающей полный диапазон частот. В некоторых случаях для уменьшения влияния нежелательных частотных компонентов может применяться техника управления окнами [21].

$$\sum_{m=-\infty}^{\infty} [x(m)h(n-m)]^2\,, \tag{1.9}$$

$$h(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{N-1}\right), 0 \le n \le N-1 \\ 0, \qquad\qquad otherwise \end{cases}, \tag{1.10}$$

где, в данном методе используется окно Хэмминга $h(\cdot)$.

Следовательно, вычисление кратковременной энергии с учетом функции задается уравнением (1.9). Оконная функция Хэмминга $h(n)$ определяется уравнением (1.10). Обновление порогового значения энергии выполняется согласно формуле (1.11), которое постоянно обновляется рассчитывается в соответствии с [19]:

$$E_{th_{new}} = E_{th_{old}} \cdot (1-\alpha) + E_{new} \cdot \alpha, \tag{1.11}$$

где E_{th_new} – это обновленное пороговое значение, E_{th_old} – предыдущее пороговое значение, E_{new} – энергия текущего кадра, а α коэффициент, принимающий значения от 0 до 1.

Алгоритм работы системы обнаружения VAD, основанный на анализе скорости пересечения нуля и кратковременной энергии, включает следующие основные этапы:

1. Инициализация пороговых значений. На начальном этапе устанавливаются пороговые значения для кратковременной энергии и частоты пересечения нуля. Данные пороги определяют, какие кадры будут считаться содержащими речь, а какие – нет. Пороговые значения могут быть выбраны

экспериментально или адаптированы на основе характеристик окружающей среды.

2. Обработка входного аудиосигнала. Входной аудиосигнал делится на небольшие временные отрезки, называемые кадрами. Каждый кадр обычно имеет продолжительность от 20 до 30 миллисекунд, что позволяет обеспечить достаточно подробное представление о временной структуре сигнала.

3. Расчет характеристик. Для каждого кадра рассчитываются основные характеристики: расчет кратковременной энергии и частоты пересечения нуля.

4. Обновление пороговых значений. Обновляются пороговые значения на основе текущих характеристик кадра.

5. Классификация. На основе сравнения расчетных характеристик с обновленными пороговыми значениями принимается решение о наличии или отсутствии речи в кадре.

6. Вывод. Система решает, является ли текущий кадр речевым или неречевым, и соответствующим образом обрабатывает сигнал.

Процесс классификации входного аудиосигнала на озвученные и невокализованные сегменты показан на блок-схеме в рисунке 1.5. Метод начинается с определения конечных точек, что включает в себя выявление начальных и конечных точек речевого высказывания. После идентификации конечных точек из интервала молчания перед началом речи берется небольшая выборка, на основе которой вычисляются кратковременная энергия и частота пересечения нуля. Такие значения устанавливаются в качестве пороговых для дальнейшего анализа.

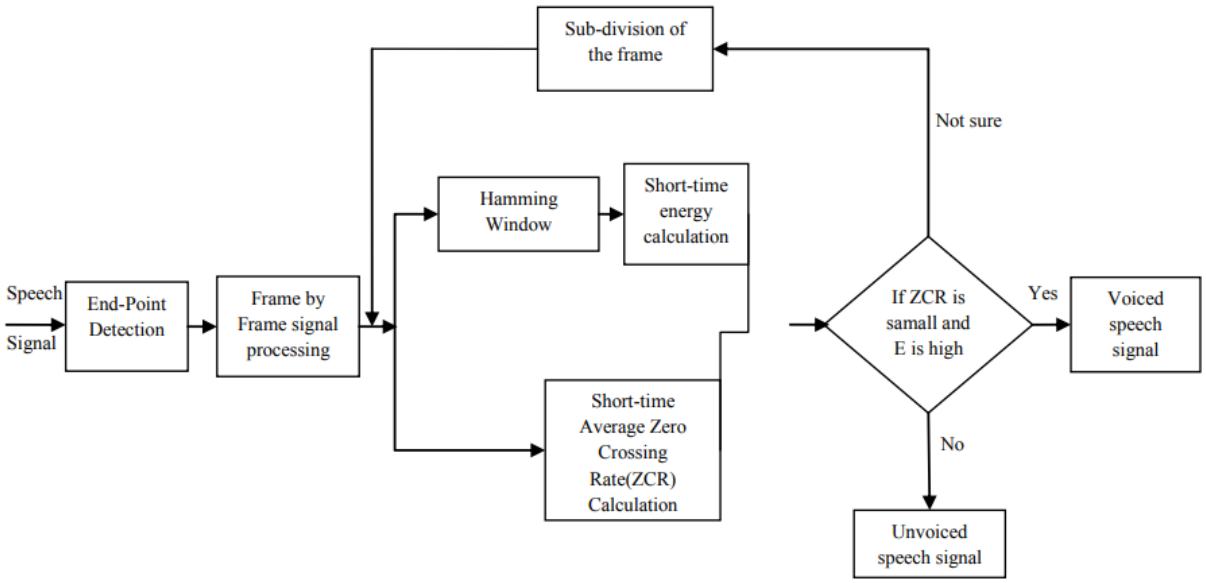


Рисунок 1.5 – Структурная схема VAD, основанная на измерениях скорости пересечения нуля и кратковременной энергии [22, с. 2]

На этапе обработки сигнала «кадр за кадром» речевой сигнал разделяется на непересекающиеся кадры, каждый из которых состоит из 400 отсчетов при частоте дискретизации 8 кГц, что эквивалентно 50 мс. Для каждого кадра

рассчитываются показатели кратковременной энергии и средней частоты пересечения нуля, которые затем сравниваются с установленными пороговыми значениями. Если кратковременная энергия кадра превышает порог, а средняя частота пересечения нуля ниже порогового значения, кадр классифицируется как озвученный сегмент. В противном случае кадр определяется как невокализованный сегмент.

В случаях, когда классификация не является однозначной, кадр делится на два подкадра, каждый из которых содержит 200 отсчетов (что эквивалентно 25 мс). Для этих подкадров снова вычисляются значения энергии и частоты пересечения нуля, и они сравниваются с пороговыми значениями, чтобы определить, относится ли подкадр к озвученным или невокализованным сегментам. Процесс продолжается до тех пор, пока не будут точно классифицированы все кадры.

Таким образом, для классификации речевого сигнала используются кратковременная энергия и средняя частота пересечения нуля, которые анализируются в кадрах длиной 50 мс. Кадры классифицируются как озвученные, если их кратковременная энергия превышает установленный порог, а частота пересечения нуля ниже порогового значения. В случаях, когда классификация не является однозначной, кадры делятся на более мелкие подкадры до тех пор, пока не удастся точно определить их принадлежность к озвученным или неозвученным сегментам. Такой подход обеспечивает высокую точность классификации [23].

1.2.3 Метод линейного прогнозирования

Алгоритмы линейного прогнозирования (LP) [24-26] активно используются в различных областях, таких как распознавание речи, идентификация говорящих, кодирование и синтез речи, а также в системах проверки личности по голосу. Данный метод получил свое название благодаря своей способности предсказывать текущее значение сигнала $x(n)$ на основе линейной комбинации предыдущих значений. Линейное прогнозирование позволяет эффективно моделировать речевые сигналы, что делает его незаменимым инструментом в технологиях обработки речи.

Суть метода LP заключается в анализе временных последовательностей сигнала и прогнозировании его дальнейшего поведения, что особенно важно для задач, связанных с речевыми данными. Применение LP в распознавании речи помогает улучшить точность систем, позволяя более точно интерпретировать акустические сигналы. В идентификации говорящих LP помогает выделять уникальные особенности голоса, что способствует более надежному распознаванию личности. В кодировании и синтезе речи метод LP позволяет сжимать и воспроизводить речевые сигналы с минимальными потерями качества.

Особое внимание в таких ситуациях заслуживает система VAD, которая часто интегрируется с LP для повышения точности обработки речи. VAD помогает отличать активные речевые сегменты от фоновых шумов и тишины,

что позволяет избежать передачи и кодирования ненужных данных. В сочетании с алгоритмами LP, VAD обеспечивает более эффективную и точную обработку речевых сигналов, делая системы более устойчивыми к внешним помехам и шумам. Предсказанное значение сигнала $\tilde{x}(n)$ вычисляется по уравнению (1.12):

$$\tilde{x}(n) = \sum_{k=1}^{K} \beta_k \cdot x(n-k)\,, \tag{1.12}$$

где K – общее количество выборок в кадре. Общее количество выборок в кадре определяется числом дискретных значений сигнала, полученных за период времени, составляющий длительность одного кадра. Количество выборок зависит от частоты дискретизации сигнала и длительности кадра.

Например, если частота дискретизации составляет 16 кГц, а длительность кадра – 50 мс, то общее количество выборок в кадре будет равно 800 (16,000 выборок/сек · 0,050 сек). n - текущий кадр. β_k - коэффициенты предсказателя, полученные методом минимизации среднеквадратичной погрешности предсказания. Данный коэффициент представляют собой набор параметров, определяющих линейную комбинацию предыдущих значений сигнала для наилучшего предсказания текущего значения. Для определения коэффициентов используется метод наименьших квадратов (МНК), который позволяет найти наилучшие оценки коэффициентов путем минимизации суммы квадратов ошибок предсказания. Ошибка предсказания определяется как разница между текущим значением сигнала и его предсказанным значением. В результате оптимизации получаются коэффициенты, которые минимизируют среднеквадратичную ошибку, что приводит к более точному предсказанию и моделированию речевого сигнала. Эти коэффициенты затем используются для восстановления сигнала или для других задач обработки речи, таких как кодирование или распознавание. Ошибка прогнозирования $e(n)$ определяется разностью фактического и предсказанного значений (1.13):

$$e(n) = x(n) - \tilde{x}(n), \tag{1.13}$$

Функции когерентности, определяется как:

$$C_{xe}(n,f) = \frac{P_{xe}(n,f)}{\sqrt{P_{xe}(n,f) \cdot P_{ee}(n,f)}}, \tag{1.14}$$

где $P_{ee}(n,f)$ – спектральная плотность $e(n)$, а $P_{xe}(n,f)$ – спектральная плотность взаимодействия сигналов между $x(n)$ и $e(n)$.

Функция когерентности $C_{xe}(n,f)$ задается уравнением (1.14). На рисунке 1.6 представлена структурная схема системы VAD, использующей усредненную функцию когерентности. Выходная амплитуда данной функции может принимать значения, близкие к единице, когда сигнал состоит преимущественно

из шума, и приближаться к нулю, если в сигнале присутствует речь. Для невокализованных сигналов амплитуда функции обычно принимает средние значения. В конечном итоге, для классификации сигналов используется определенное правило порогового значения, которое позволяет точно различать типы сигналов.

$$|C_{xe}(n,f)| \geq th \Longrightarrow noise$$
$$|C_{xe}(n,f)| < th \Longrightarrow speech, \quad (1.15)$$

где th значение порога.

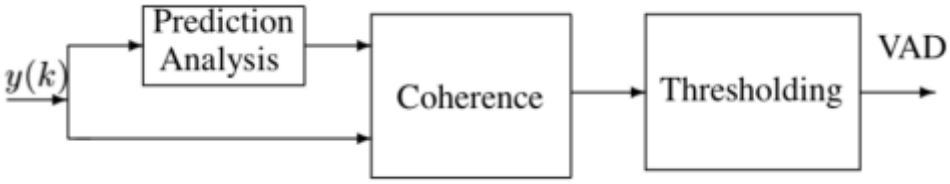


Рисунок 1.6 – VAD, основанный на функции когерентности [10, с. 6]

Таким образом, линейное прогнозирование представляет собой эффективный метод обработки речевых сигналов, позволяющий с высокой точностью оценивать текущее значение сигнала на основе его предыдущих значений. Применение функции когерентности способствует различению речевых и неречевых сигналов. Правило классификации «речь/шум» на основе модуля C_{xe} описано в уравнении (1.15). В итоге, комбинация алгоритмов линейного прогнозирования и функции когерентности предоставляет мощные инструменты для повышения качества и точности систем обработки речевых сигналов.

1.2.4 Метод одночастотной фильтрации

Метод одночастотной фильтрации SFF (Single Frequency Filtering) [27] основывается на предположении, что энергия шума равномерно распределена по всему частотному спектру, в то время как энергия речи распределена неравномерно. Значение отношения С/Ш (SNR) можно вычислить как:

$$SNR_a = \int_{f_0}^{f_l} \frac{S^2(f)}{N^2(f)} df, \quad (1.16)$$

Общая энергетическая метрика задается уравнением (1.16). Суммарное отношение С/Ш по L интервалам рассчитывается как:

$$SNR_b = \sum_{i=0}^{L-1} \frac{\int_{f_i}^{f_{i+1}} S^2(f) df}{\int_{f_i}^{f_{i+1}} N^2(f) df}, \quad (1.17)$$

Из формулы (1.17) видно, что при увеличении числа интервалов L метрика SNR_b становится более точной, но растет вычислительная сложность.

Отношение полной энергии определяется:

$$SNR_c = \frac{\int_{f_0}^{f_L} S^2(f)df}{\int_0^{f_L} N^2(f)df}, \tag{1.18}$$

где $D(f)$ где амплитуда сигнала, $N(f)$ – амплитуда шума на частоте f, (f_i, f_{i-1}) – это L непересекающихся частотных интервалов.

Так, согласно формуле (1.18), отношение SNR_c определяется как отношение полной энергии сигнала к полной шумовой энергии.

Следовательно, выполняется следующее неравенство:

$$SNR_a \geq SNR_b \geq SNR_c, \tag{1.19}$$

Как видно из уравнения (1.19), между метриками SNR_a, SNR_b и SNR_c всегда справедливо такое неравенство.

В работе [28] была представлена система обнаружения активности голоса VAD, основанная на методе одночастотной фильтрации (SFF). В данной системе входной сигнал сначала дискретизируется с заданной частотой дискретизации f_s. После этого сигнал проходит через дифференцирующий блок, где его дискретная временная версия x(n) определяется как разность между текущим значением сигнала $s(n)$ и предыдущим значением $s(n-1)$. Далее, результат дифференцирования умножается на комплексную синусоиду. Математически это записывается как:

$$x_k(n) = x(n) \cdot e^{j\overline{w}_{kn}},$$

$$\overline{w}_k = \frac{2\pi f_k}{f_s}, \tag{1.20}$$

где $\overline{w}_k$ – нормализованная частота.

Как следует из уравнения (1.20), для каждого канала сигнал $x(n)$ умножается на комплексную синусоиду с нормализованной частотой $\overline{w}_k$.

Данный процесс позволяет выделить частотные компоненты сигнала и улучшить разделение речевых и неречевых сегментов. Умножение на комплексную синусоиду помогает сместить спектр сигнала в определенную частотную область, что упрощает последующую фильтрацию и анализ. SFF в данной системе VAD используется для повышения точности обнаружения активности голоса, особенно в условиях низкого отношения С/Ш, где традиционные методы могут показывать менее стабильные результаты.

После преобразования разностного сигнала путем умножения на комплексную синусоиду с нормализованной частотой, сигнал подвергается

обработке с использованием однополюсного фильтра. Нормализованная частота служит для спектрального сдвига сигнала, что облегчает последующую обработку. Введение однополюсного фильтра на данном этапе выполняет важную функцию сглаживания сигнала и устранения высокочастотных шумов.

Однополюсный фильтр способствует выделению полезных сигналов, что является критически важным для систем VAD. Данный процесс улучшает способность системы распознавать активную речь, снижая влияние ненужных шумов и высокочастотных компонентов.

$$H(z) = \frac{1}{1 + rz^{-1}}, \tag{1.21}$$

Мощность шума обычно распределена равномерно по частотному спектру, но при нестационарном шуме это может изменяться. Для компенсации используется метод взвешивания сигнала на каждой частоте, основанный на минимальных значениях. Предполагается, что в начале каждого кадра речи нет, поэтому средняя амплитуда этой части служит для расчета нормализованного веса w_k на каждой частоте f_k.

$$w_k = \frac{\frac{1}{\mu_k}}{\sum_{p=1}^{N} \frac{1}{\mu_p}}, \tag{1.22}$$

Следовательно, передаточная функция однополосного фильтра задается в уравнении (1.21). Весовой коэффициент нормализации w_k определяется формулой (1.22).

Энергия сигнала может быть аппроксимирована через среднее значение μ(n) квадрата взвешенных частотных компонентов. Это значение обычно выше для речевого сигнала по сравнению с шумом, когда в сигнале присутствует речь.

Другая величина, которая также демонстрирует аналогичное поведение, которое представляет собой стандартное отклонение $\sigma(n)$ квадрата огибающих взвешенных частотных компонент. Чтобы лучше выделить контраст между речевыми и неречевыми участками, можно использовать комбинацию значений $\mu(n)$ и $\sigma(n)$, в следующей форме:

$$\delta(n) = \sqrt[M]{|\sigma^2(n) - \mu^2(n)|}, \tag{1.23}$$

Принятие решения о присутствии или отсутствии речи основывается на сравнении порогового значения $\theta(n)$ и сглаженного во времени показателя $\delta(n)$. Пороговое значение $\theta(n)$ рассчитывается следующим образом:

$$\theta = \mu_\theta + 3\sigma_\theta, \tag{1.24}$$

Пороговое значение $\theta(n)$ обновляется при каждом произнесении, чтобы учитывать изменения фонового шума. Для определения размера окна сглаживания l_m рассчитывается динамический диапазон ρ, основанный на энергии сигнала, для каждого кадра m длительностью 300 мс:

$$\rho = 10 \cdot \log_{10}\left[\frac{\max_m(E_m)}{\min_m(E_m)}\right] \Longrightarrow \begin{cases} L_w = 400\,ms & \rho < 30, \\ L_w = 300\,ms & 300 \le \rho \le 40, \\ L_w = 200\,ms & \rho > 40. \end{cases} \quad (1.25)$$

как только размер окна получен, можно определить усредненное значение $\delta(n)$ и сравнить его с пороговым значением:

$$d(n) = 1, for\ \overline{\delta}(n) > \theta(\text{наличие речи})$$
$$d(n) = 0, for\ \overline{\delta}(n) \le \theta(\text{отсутствие речи}), \quad (1.26)$$

Таким образом, контрастная характеристика $\delta(n)$ вычисляется по формуле (1.23) как М-й корень из модуля разницы $\sigma^2(n)$ и $\mu^2(n)$, порог $\theta(n)$ для классификации речи задается суммой фонового среднего и трех его стандартных отклонений согласно уравнению (1.24), динамический диапазон ρ и соответствующая длительность окна L_w определяются через отношение максимальной и минимальной энергии по уравнению (1.25), а собственно решение о наличии или отсутствии речи $d(n)$ формализовано правилом из уравнения (1.25).

Для сглаживания данного алгоритма принятия решения можно подсчитывать, сколько раз значение $d(n)$ равно единице в каждом кадре. При сравнении данного метода с методами адаптивной многоскоростной обработки (adaptive multi-rate, AMR) следует уменьшить длительность анализируемых кадров до 10 мс. Это сокращение позволяет достичь более точной и быстрой обработки, что критично для систем, требующих высокой чувствительности к изменениям в речевой активности.

1.3 Обзор и анализ существующих детекторов голосовой активности на основе искусственных нейронных сетей

В настоящее время существует множество способов реализации VAD, и решения, основанные на применении методов машинного обучения, приобретают все большую популярность. Возможность обучения технических средств принимать решения, основанные на предыдущем опыте, дают возможность создавать системы обработки информации, обладающие большим быстродействием и не уступающим в точности классическим алгоритмам. Одним из наиболее быстро развивающихся направлений в машинном обучении является использование нейронных сетей и глубокого обучения, что позволяет решать широкий круг задач. Глубокое обучение включает несколько уровней представлений, что имеет особое значение для систем машинного обучения. В

отличие от традиционных нейронных сетей, скрытые слои в глубоких сетях могут выявлять более абстрактные признаки из входных данных [29-34].

Модель на основе импульсных нейронных сетей SNN (Spiking Neural Networks) представлена в работе [29] для улучшения производительности VAD при минимальном энергопотреблении. Предлагаемая VAD-модель на основе SNN состоит из аудиокодера, предназначенного для выделения признаков, и классификатора для классификации на уровне кадров (представлен на рисунке 1.7).

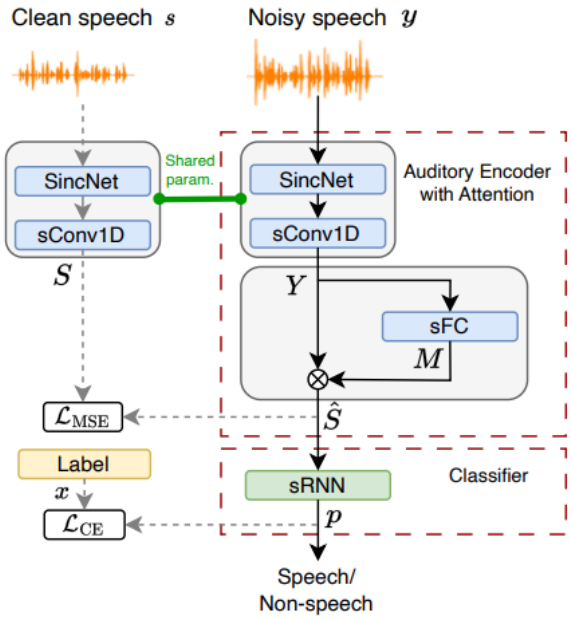


Рисунок 1.7 – VAD-модель на основе SNN [29, с. 2]

Чтобы подтвердить эффективность предлагаемого аудиокодера, в работе провели исследование в условиях высокого уровня шума. По сравнению с базовым VAD предложенная модель на основе импульсных нейронных сетей привела к значительному увеличению эффективности на 4,1%. Несмотря на то, что VAD демонстрирует значительную точность в условиях минимального фона, ее производительность существенно ухудшается в ситуациях с низким отношением С/Ш. Главная проблема VAD заключается в том, что модели VAD часто сталкиваются с серьезными трудностями при эффективном извлечении речевых сигналов в условиях высокого уровня шума.

В работе оценивают предложенную модель VAD на основе набора данных QUT-NOISE- TIMIT [30], состоящего из 600 часов зашумленной речи. Данный набор данных объединяет чистые записи речи из набора данных TIMIT с реальными сценариями шума, такими как кафе, автомобиль, дом, улица и реверберационная среда. Набор данных подразделяется на следующие уровня шума SNR +15 дБ, +10 дБ, +5 дБ, 0 дБ.

Большой объем зашумленной речи, доступный в QUTNOISE-TIMIT в широком диапазоне сценариев, позволил выбрать относительно надежные пороговые значения сегментации для этих систем по сравнению со встроенными пороговыми значениями в системах, основанных на стандартах. В частности, способность системы GMM-MFCC VAD изучать характеристики речевых и

неречевых сигналов в разных сценариях обеспечила наилучшую производительность при всех уровнях шума.

Исследование производительности работы VAD было проведено в работе [31] с использованием базы данных TIMIT. Предложенный алгоритм, основанный на пересечении уровней выборки для различения речевой и неречевой частей аудиосигнала представлен на рисунке 1.8.

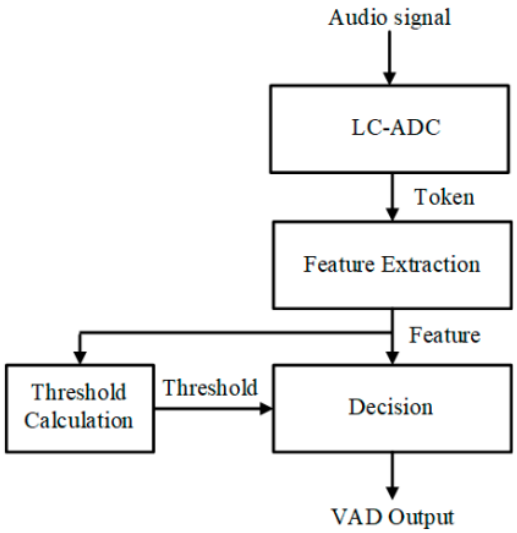


Рисунок 1.8 – Блок-схема предлагаемого алгоритма VAD [31, с. 6]

Предлагаемый алгоритм обеспечивает приемлемую точность для сигналов с различными типами шума и уровнями SNR. Однако данный метод намного сложнее, чем другие алгоритмы, что делает его подходящим вариантом только для аппаратной реализации VAD. Также производительность предлагаемого алгоритма может быть улучшена за счет применения таких методов, как адаптивное разрешение, но за счет увеличения энергопотребления.

Как показывают статистические наблюдения, типичный разговор на 60% состоит из молчания и на 40% - из речи [32]. Поскольку база данных, используемая в данной работе, содержит 90% речи и 10% тишины, в работе случайным образом добавили общее значение четырех секунд тишины в начале и конце каждого входного аудио, как это было сделано во многих других исследованиях [33]. Предлагаемая система обеспечивает в среднем 91,02%-ную частоту распознавания речи и 82,64%-ную частоту распознавания неречевых сигналов. Однако, в работе также отмечается, что производительность предложенного алгоритма можно повысить за счет применения таких методов, как адаптивное разрешение, фильтрация, но также за счет увеличения энергопотребления.

В работе [34] описывается метод обнаружение голосовой активности HiVAD на смартфонах в режиме реального времени с помощью сверточных нейронных сетей. Алгоритм данного метода представлен на рисунке 1.9. По результатам тестирования, HiVAD показал среднюю точность SHR 8,67%, 16,29%, 17,63% при уровне 5 дБ и 1,35%, 7,72%, 5,14% на уровне 10 дБ.

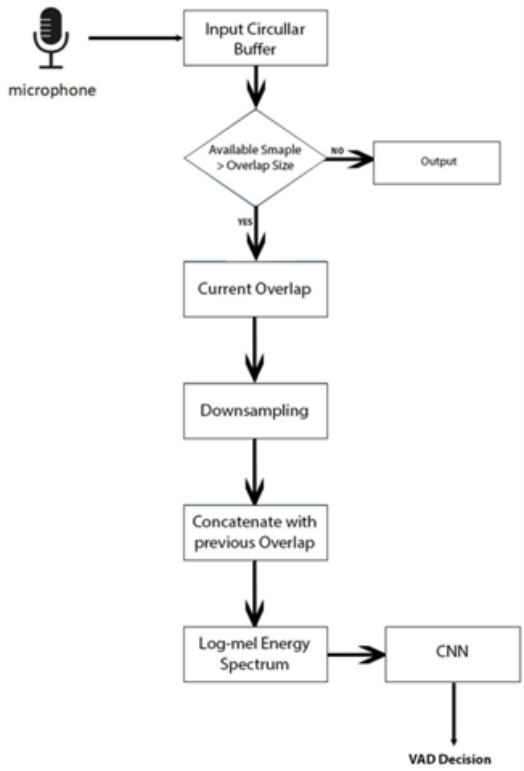


Рисунок 1.9 – Блок-схема метода HiVAD [34, с. 6]

Наборы данных FSDD (Free Spoken Digit Dataset) активно используются для тренировки и тестирования моделей CNN VAD [35]. Данный набор включает 3000 аудиофайлов, созданных 6 разными дикторами. Кроме того, в качестве набора данных о шуме использовался набор данных [36], который содержит 15 различных фоновых шумовых состояний, таких как пляж, бар, кафе /ресторан, автомобиль, центр города, лесную тропинку, продуктовый магазин, дом, библиотеку, станцию метро, офис, парк, жилой район, поезд и трамвай.

В исследовании [37] был предложен метод автоматического обнаружения речи в аудиосигналах, основанный на использовании глубоких нейронных сетей (DNN). Хорошие результаты были достигнуты с использованием нейронной сети CNN (см. рисунок 1.10).

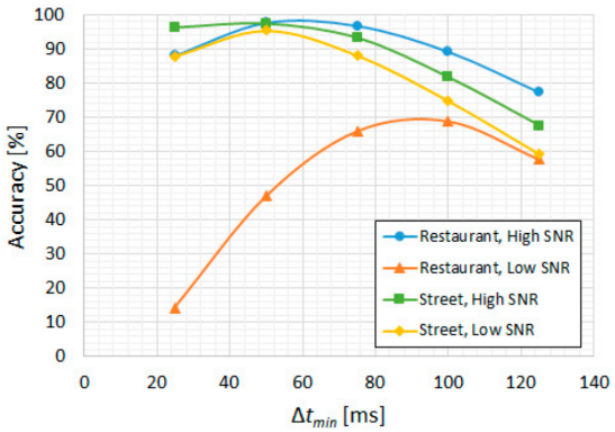


Рисунок 1.10 – Результаты тестирования VAD на основе CNN, подмножество CENSREC-1-C [37]

Для обучения и тестирования систем использовались несколько подмножеств данных из базы данных CENSREC-1-C, с различными моделируемыми условиями фонового шума. Дополнительное тестирование проводилось на другом подмножестве данных CENSREC-1-C, включающем реальные записи с фоновым шумом. Для данного набора данных была получена точность до 99,13%. Значение высокого ОСШ группирует уровни 10–20 дБ, а значение низкого ОСШ равно 5 дБ. Однако авторы работ утверждают, что необходимо изучить другие модели глубокого обучения для задачи VAD, а также включить этап улучшения речи в конвейер обработки, чтобы получить более высокие отношения С/Ш и уменьшить влияние окружающего шума.

В работе [38] было предложено максимизировать площадь под кривой ROC (MaxAUC) с помощью DNN, что может максимизировать эффективность VAD с точки зрения всей кривой ROC. Метод VAD на основе кривой ROC и AUC представлен на рисунке 1.11.

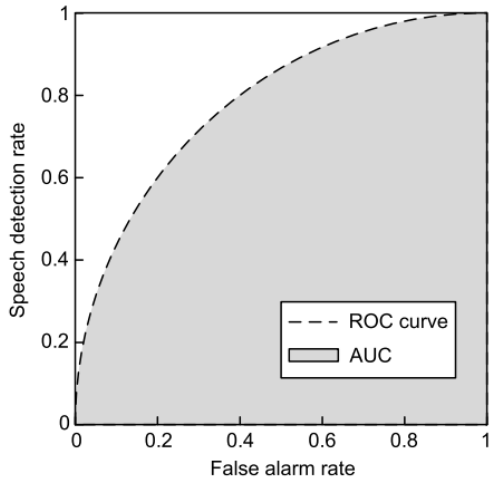


Рисунок 1.11 – VAD на основе кривой ROC и AUC [38, с. 4]

Для обучения и тестирования данного метода была использована база данных LibriSpeech ASR [39], включающая 1000 часов прочитанной английской речи, а также набор данных CHiME-4, содержащий 7138 высказываний на английском языке. Также использовали масштабную библиотеку звуковых эффектов, содержащий более 20 000 звуковых эффектов и базу данных NOISEX-92, содержащая 9 шумовых сценариев, длительность каждого из которых составляет около 5 минут. Используя эти данные сгенерировали более 20 часов шумной английской речи в широком диапазоне сценариев тестирования с учетом различных моделей DNN, акустических характеристик, обучающих наборов, а также сценариев несоответствия шума и языка.

В работе [40] представлена гетерогенная сверточная рекуррентная нейронная сеть (HCRNN) с механизмом внимания и агрегацией признаков для обнаружения речевой активности. Данный подход эффективно объединяет преимущества различных сетей, стремясь достичь превосходной производительности при обнаружении голосовой активности. Эксперименты проводились с синтетическими наборами данных VAD, наборами данных kaggle

VAD и наборами данных AVA-speech. Кривые средней точности mAP (mean Average Precision) и рабочих характеристик приемника ROC (receiver operating characteristic) демонстрируют эффективность предложенного метода. Данный метод показал эффективность распознавания речи 93% при значении ОСШ 10дБ

Для обучения моделей было использовано 6 эпох для синтетических наборов данных VAD, в то время как для наборов данных Kaggle VAD и AVA-speech было выделено 100 эпох. Модель обучается с помощью Adam optimizer в [41] со скоростью обучения 0,00005, используя эффективный размер пакета, равный 16. Однако авторы работ утверждают, что необходимо внедрить новый механизм attention и методы агрегации функций в большее количество сетевых архитектур на базе CRNN для повышения производительности VAD.

В работе [42] предлагается детектор голосовой активности на основе глубокой нейронной сети, который обнаруживает короткие паузы в высказывании для сокращения задержки ответа, пока пользователь произносит длинные предложения. В экспериментах по распознаванию корейской речи число одновременных клиентов увеличивается с 22 до 44 с использованием предлагаемого вычисления акустической оценки. В сочетании с методом пропуска кадров число увеличивается до 59 клиентов с небольшим ухудшением точности. Более того, средняя воспринимаемая пользователем задержка сокращается с 11,71 с до 3,09–5,41 секунд с использованием предлагаемого детектора голосовой активности на основе глубокой нейронной сети. Гистограмма воспринимаемой пользователем задержки для всех слов в тестовом наборе представлена на рисунке 1.12.

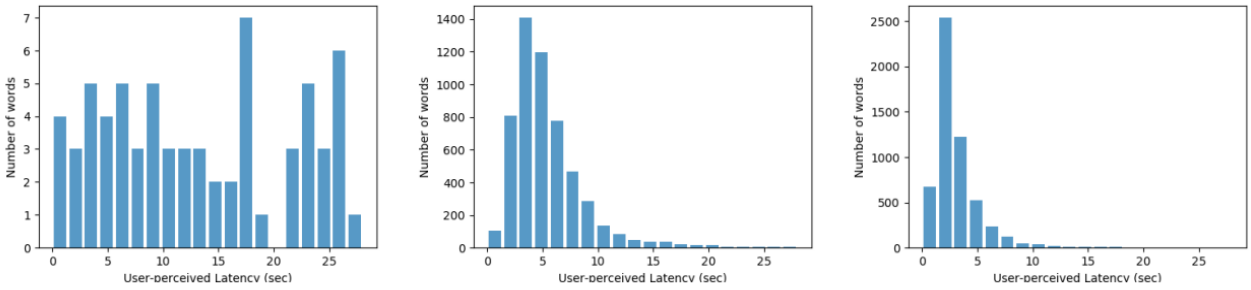


Рисунок 1.12 – Гистограмма воспринимаемой пользователем задержки для всех слов в тестовом наборе. (a) Без VAD (b) С VAD 200 кадров (c) С VAD, 100 кадров [42, стр 16]

Тем не менее, авторами работ [42] также отмечается, что необходимы дальнейшие исследования для преодоления сложных проблем, таких как повышение эффективности VAD при неблагоприятных условиях, а также обеспечение низкого энергопотребления при постоянной работе в режиме реального времени алгоритма VAD.

В работе [43] провели эксперименты с несколькими классификаторами нейронных сетей с использованием детектора пробуждающих слов, например, путем их совместного обучения, значительно улучшает производительность в

самых шумных условиях. Кроме того, в работе представили новую общедоступную базу данных речи, записанную для голосового помощника Telefónica, Aura. Пример спектрограмм улучшения речи показан на рисунке 1.13.

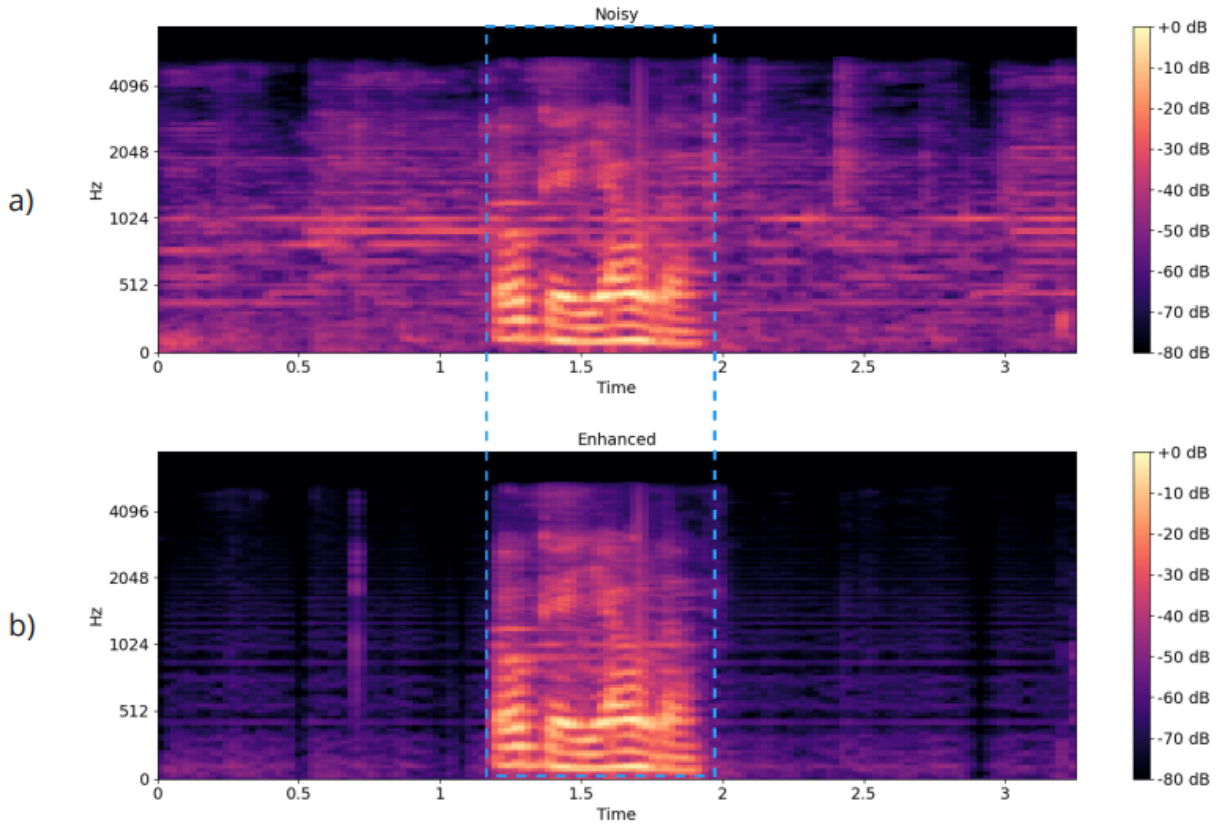


Рисунок – 1.13 – Пример спектрограмм улучшения речи. На каждом рисунке показана (a) шумная логарифмическая спектрограмма и (b) улучшенная логарифмическая спектрограмма. Синий прямоугольник показывает, где размещено ключевое слово «OK Aura» [43, с. 9].

Набор данных слов для пробуждения «OK Aura» содержит обширные метаданные, такие как демографические данные говорящего или условия в помещении, а также содержит жесткие отрицательные примеры, которые были тщательно отобраны для представления различных уровней фонетического сходства по отношению к словам-триггерам «OK Aura».

В работе [44] представлена нейронную сеть для обнаружения речевой активности Marble Net, способная обеспечить производительность в условиях ограниченных вычислительных возможностей, таких как мобильные и носимые устройства. Marble Net основан на архитектуре Quartz Net. Он включает в себя B блоков, каждый из которых имеет R подблоков. Все подблоки в каждом блоке имеют одинаковые выходные каналы C (представлен на рисунке 1.14).

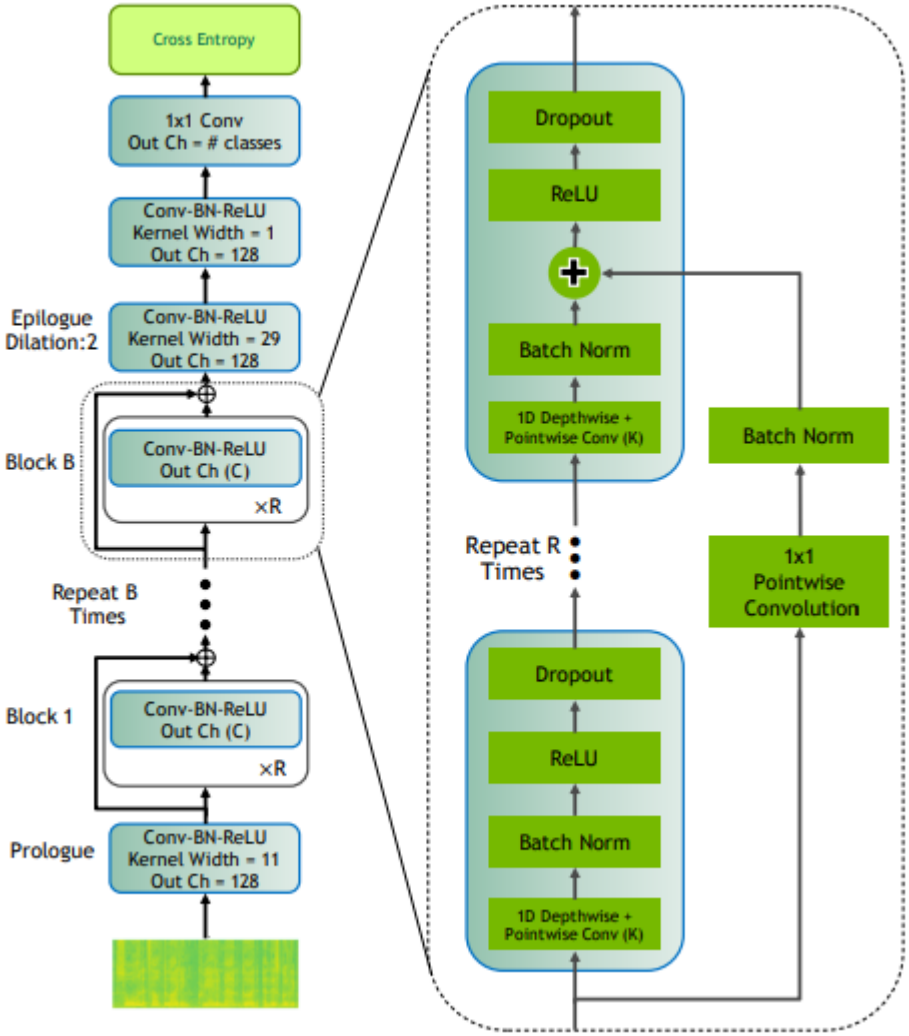


Рисунок 1.14 – Модель Marble Net, B – количество блоков, R – количество подблоков, C – количество каналов [44, с. 2]

В качестве речевых данных используют Google Speech Commands Dataset версии 2 [45]. Данный набор данных содержит 105 000 высказываний, каждое длиной около 1 секунды, относящихся к одному из 35 классов распространенных слов, таких как «Да» и «Вперед». В качестве неречевых данных использовались около 2700 выборок переменной длины из 35 фоновых категорий, таких как «дорожный шум», «внутри, в маленькой комнате» из freesound.org [46].

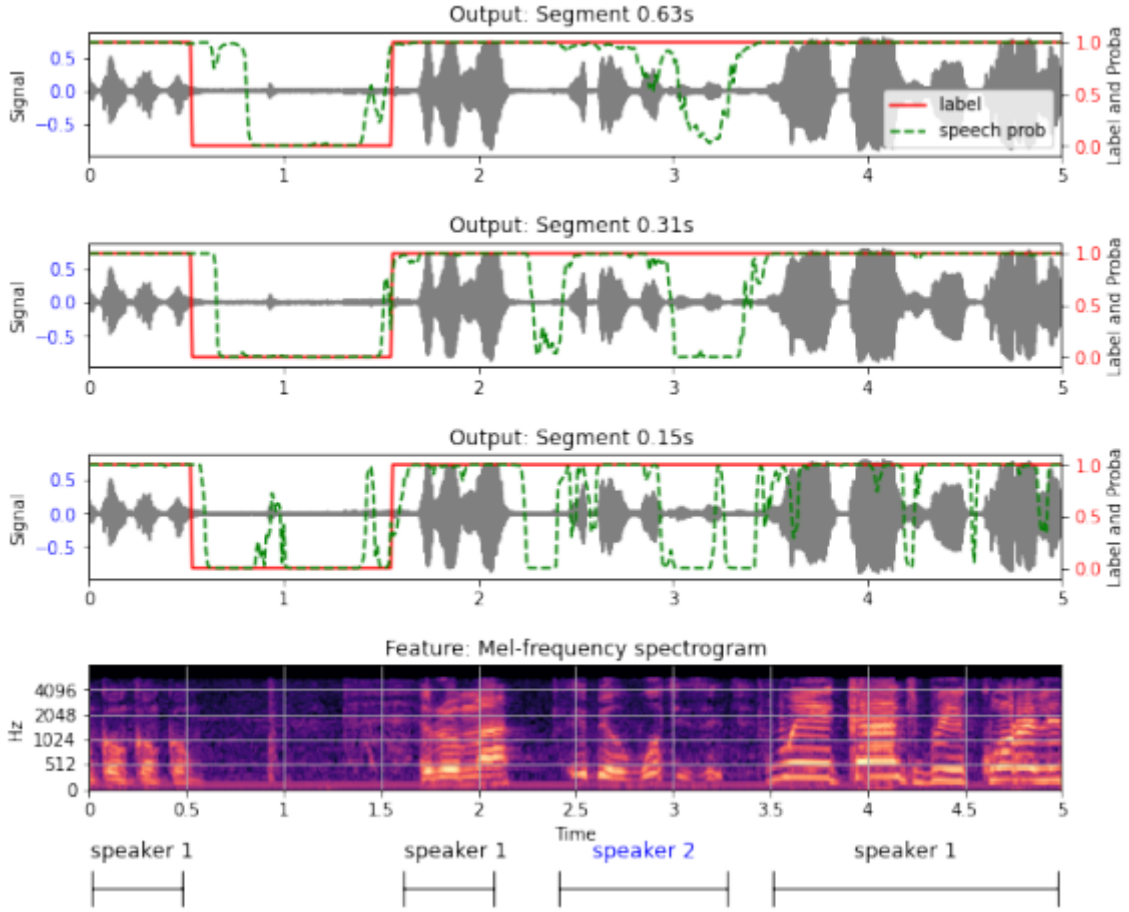


Рисунок 1.15 – Пример изменения длины сегмента ввода [44, с. 4]

Продолжительность каждой выборки составляет от 0,63 до 100 секунд. На рисунке 1.15 показан пример изменения длины сегмента ввода. Более короткие сегменты ввода действительно позволяют обнаруживать больше пауз (зеленые линии), однако метки истинности (красные линии) обозначают всю последнюю часть примера как «речь».

Хотя существует множество предложенных алгоритмов VAD, они неизбежно сталкиваются с типичными проблемами, обусловленными сложностью структуры речевого сигнала, что затрудняет его точное математическое описание. Кроме того, различные внешние факторы могут существенно влиять на процесс записи и передачи голоса. В большинстве случаев алгоритмы VAD должны обнаруживать присутствие речи в сигналах, искаженных шумом. Такие трудности приводят к основным недостаткам современных алгоритмов VAD, таким как недостаточная точность определения границ речевых сегментов и значительное ухудшение производительности в шумных условиях.

Выводы по главе:

1. Подтверждено, что в современных системах распознавания речевого сигнала, особенно при низком отношении С/Ш, традиционные методы VAD часто оказываются недостаточно эффективными. Нейросетевые методы VAD

демонстрируют значительное улучшение в точности и надежности детектирования, благодаря их способности обучаться на больших объемах данных и учитывать сложные нелинейные зависимости в речевом сигнале. Это делает их незаменимыми в условиях низкого отношения С/Ш.

2. Установлено, что традиционные методы VAD, такие как метод пересечения нуля, энергетические методы, методы линейного прогнозирования и одночастотной фильтрации, имеют свои преимущества и недостатки. Они могут быть эффективны в условиях высокого отношения С/Ш, но их производительность значительно снижается в шумных условиях. Методы на основе глубокого обучения, напротив, показывают высокую устойчивость к шуму и способны адаптироваться к различным акустическим условиям, что делает их более предпочтительными для современных систем распознавания речевого сигнала.

3. Выявлено, что современные VAD, основанные на искусственных нейронных сетях, демонстрируют высокую производительность и точность в различных условиях. Они способны учитывать широкий спектр признаков и адаптироваться к изменяющимся условиям окружающей среды. Анализ существующих нейросетевых VAD показал, что они превосходят традиционные методы по многим параметрам, включая устойчивость к шуму, точность детектирования и т.д. Это подтверждает целесообразность их использования в системах распознавания речевого сигнала.

2 ПРИМЕНЕНИЕ АРХИТЕКТУР ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ДЕТЕКТИРОВАНИЯ РЕЧЕВОГО СИГНАЛА

2.1 Методы глубокого обучения в задаче детектирования речевого сигнала

Технологии распознавания речи играют все более важную роль в повседневной жизни людей. Одним из важных этапов в распознавании речи является выделение речевых фрагментов из общего потока данных, что достигается с помощью системы VAD. С увеличением производительности компьютеров и развитием методов машинного обучения, VAD стал более точным и эффективным инструментом для обработки речевых данных. В настоящее время методы машинного обучения можно условно разделить на три главные категории: контролируемое обучение, неконтролируемое обучение и обучение с подкреплением. В рамках неконтролируемого обучения модели анализируют различные возможности и обучаются на основе больших массивов данных без предварительно заданных меток. Уменьшение размерности и кластеризация - два распространенных метода обучения без контроля, которые играют основную роль в задачах анализа данных и обработки информации [47].

Кластеризация же позволяет группировать данные на основе их сходства, что помогает выделить скрытые закономерности и структуры, улучшая понимание данных и обеспечивая новые выводы. Оба метода эффективно применяются в области обучения без учителя для автоматического извлечения информации из данных и выявления важных закономерностей. Методы уменьшения размерности позволяют извлекать надежные низкоразмерные признаки из многомерных данных, которые затем используются классификаторами. К таким методам относятся анализ главных компонент PCA (principal component analysis) [48], неотрицательная матричная факторизация NMF (Non-negative Matrix Factorization) [49] и спектральное разложение матрицы Лапласиана [50]. Алгоритмы кластеризации представляют собой методы машинного обучения, которые предназначены для группировки данных в категории на основе их заметных характеристик. Среди наиболее известных методов можно выделить K-средние и модели гауссовых смесей (GMM). Такие техники помогают выявлять структуры и связи в данных, что значительно облегчает их анализ и интерпретацию [51, 52]. Поскольку методы, основанные на неконтролируемом машинном обучении, не требуют больших объемов помеченных данных, получить обучающие данные легко. Однако данные методы плохо работают при низком уровне отношения С/Ш.

Другим направлением исследований в области VAD, основанного на машинном обучении, являются методы контролируемого обучения. В этой категории VAD рассматривается как задача бинарной классификации, где аудиофреймы необходимо разделить на речевые и неречевые. Хотя контролируемые методы требуют значительных объемов размеченных данных и вычислительных ресурсов для обучения и тестирования моделей, современная инфраструктура данных и вычислительная мощность компьютерных систем

позволяют успешно решать данные задачи. Наиболее распространенные подходы, основанные на контролируемом машинном обучении и применяемые для VAD, включают методы опорных векторов SVM (Support Vector Machines) [53], разреженное кодирование [49] и скрытые марковские модели HMM (Hidden Markov Models) [54].

В последние десятилетия глубокие нейронные сети DNN стали важным компонентом машинного обучения, показывая выдающиеся результаты в различных областях. Такие сети имеют уникальную способность автоматически извлекать сложные иерархические признаки из данных, что позволяет им эффективно обучаться на больших объемах информации и справляться с множеством разных задач. Эти современные методы машинного обучения с использованием DNN продолжают активно развиваться и находят применение во многих сферах жизни, способствуя развитию искусственного интеллекта. Их беспрецедентная эффективность также была признана во многих областях, таких как компьютерное зрение и обработка естественного языка. Многие исследователи предложили методы VAD на основе DNN, которые обладают исключительной эффективностью в плане точности обнаружения в условиях сильных внешних помех и шумов. Глубокие нейронные сети позволяют моделям автоматически выявлять особенности и шаблоны шума в аудиосигналах, что делает их намного эффективнее в сравнении с классическими подходами к обработке звука. Благодаря использованию DNN в методах VAD удается достичь высокой степени точности и надежности при определении наличия речи в условиях сильного шума. Данные подходы позволяют выявлять скрытые закономерности путем неявного моделирования обширных помеченных данных, которые не требуют предварительных знаний и предположений о явной модели. Такая характеристика объясняется способностью методов DNN к нелинейному преобразованию, которая позволяет им улавливать изменения в речи.

2.1.1 Сверточные нейронные сети в задаче детектирования речевого сигнала

Сеть CNN является разновидностью глубоких нейронных сетей DNN, преимущественно применяемых для классификации изображений и задач распознавания образов. В этой модели все изображение проходит процесс сканирования с использованием фильтров. В большинстве случаев в литературе упоминаются фильтры размеров 1x1, 3x3 и 5x5. Архитектуры сверточных нейронных сетей CNN обычно состоят из разнообразных типов слоев, включая сверточные слои, слои подвыборки (максимальное или среднее объединение) и полностью связанные слои. Важнейшими компонентами архитектуры CNN являются сверточные слои, которые выполняют сверточные операции для извлечения признаков. На рисунке 2.1 представлена обобщенная структура сверточной нейронной сети, включающая в себя несколько категорий уровней: сверточные слои, слои выборки или объединения и полносвязные слои.

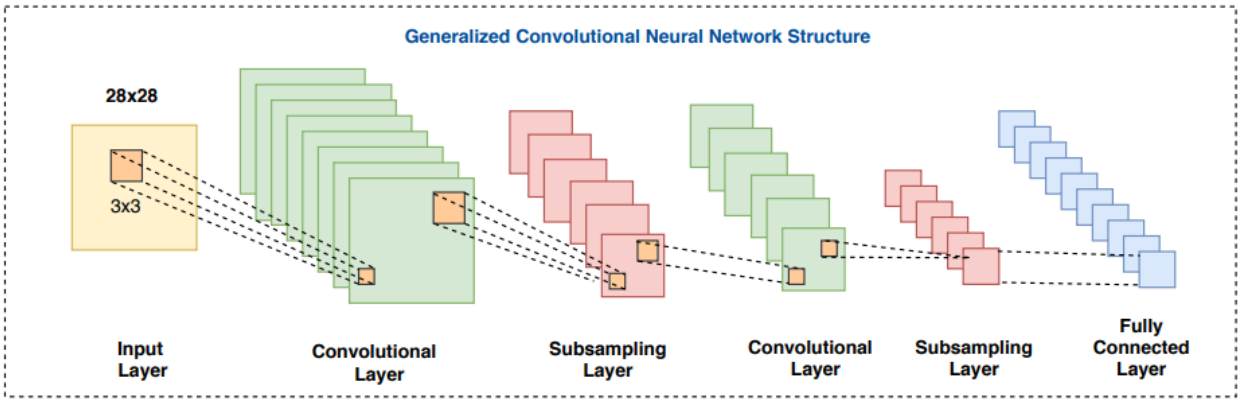


Рисунок 2.1 – Архитектура обобщенной сверточной нейронной сети [55, с. 5]

Сверточный слой является первым компонентом в архитектуре нейронной сети, предназначенным для извлечения различных признаков из входных изображений. Основная концепция применения сверточного слоя заключается в использовании математической операции свертки (или свертки) для обработки изображения [56]. Свертка представляет собой двумерную матрицу весов и имеет ряд преимуществ при использовании фильтров. Одно из главных достоинств это увеличение величины выходного сигнала в зависимости от степени соответствия элемента изображения фильтру. В результате сверточной операции создается выходное изображение, где каждый пиксель показывает уровень совпадения определенной области исходного изображения с заданным фильтром. При этом входной сигнал изображения подается на нейрон, охватывающий ограниченную область, обычно квадратной формы, такой как 3x3 пикселя. Эта область затем смещается вправо на заданное значение, например, на 1 пиксель, и входные данные подаются на следующий нейрон. Так продолжается процесс сканирования всего изображения. При этом весовые коэффициенты для всех нейронов в данной группе остаются неизменными.

Сверточный слой состоит из одного или нескольких каналов, каждый из которых содержит определенное количество узлов. Каждый канал на предыдущем уровне полностью соединен с каждым каналом на следующем уровне (см. рисунок 2.2 (a)). Связь между конкретным входным каналом (на уровне L) и определенным выходным каналом (в слое $L + 1$) представлена на рисунке 2.2 (b). Свертка, выполняемая между набором узлов, находящихся в заданном входном канале (обозначенном черным квадратом), и ядром свертки, состоящим из весов ($k_1, \dots k_9$), приводит к получению одного узла в выходном канале. Вычисления для этого узла в выходном канале осуществляются с использованием суммы, где $\sum A$ обозначает суммирование по всем входным каналам A (смотрите рисунок 2.2 (c)).

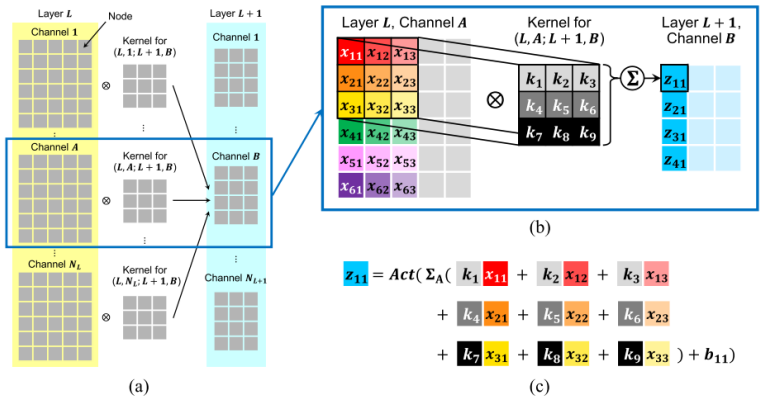


Рисунок 2.2 – Структура и вычисление сверточного слоя в нейронной сети [57, с. 8]

Значение $Z_{c,d}^{L+1,B}$ в координатах (c,d) в канале Bв слое $L+1$ определяется как:

$$Z_{c,d}^{L+1,B}(z^{L,1}, z^{L,2}, \dots z^{L,N_L};\ k^{L,1;L+1,B}, k^{L,2;L+1,B}, \dots k^{L,N_L;L+1,B}) = \text{Act}\left(\sum_{A=1}^{N_L} v_{c,d}^{L+1,B}(z^{L,A}; k^{L,A;L+1,B}) + b_{c,d}^{L+1,B}\right), \quad (2.1)$$

$$v_{c,d}^{L+1,B}(z^{L,A};\ k^{L,A;L+1,B}) = \sum_{a=1}^{M}\sum_{\beta=1}^{M}\left(k_{a,\beta}^{L,A;L+1,B} \cdot z_{c-\left[\frac{M}{2}\right]+a,d-\left[\frac{M}{2}\right]+\beta}^{L,A}\right), \quad (2.2)$$

где Act() – функция активации, $b_{c,d}^{L+1,B} \in R$ – смещение, связанное с выходным узлом, $z^{L,A}$ обозначает канал A в предыдущем слое L, N_L представляет собой количество каналов в слое L, $k^{L,A;L+1,B}$ относится к ядру свертки размером $M = M$, определенному для соединения между каналом A в слое L и каналом B в слое $L+1$.

После сверточного слоя часто используют слой субдискретизации, который помогает уменьшить размер карты признаков и сократить вычислительные затраты. Этот слой снижает число связей между слоями, работая независимо для каждой карты признаков. В результате уменьшается объем данных, а также выявляются более абстрактные и устойчивые признаки, что повышает общую эффективность модели.

Для уменьшения размерности карт признаков используются различные операции, как показано на рисунке 2.3. К таким операциям относятся:

- «MaxPooling» – выбирает наибольшие значения из заданной области выборки;
- «MinPooling» – выбирает наименьшие значения из заданной области выборки;
- «AveragePooling» – рассчитывает средние значения в рамках области выборки.

Линейный субдискретизирующий слой (S_k) уменьшает размерность карт признаков, выполняя усреднение соседних значений. Сначала карта признаков делится на равные непересекающиеся участки, затем для каждого участка вычисляется среднее, которое используется в обновленной карте признаков. Преобразование карты Y_j^{k-1} в Y_j^k описывается следующей формулой:

$$Y_j^k(x,y) = \varphi_j\left(\omega_{ij}^k \times \sum_{(u_x,u_y)} y_i^{k-1}(sx+u_x, sy+u_y)\right), \quad (2.3)$$

где k – коэффициент децимации, $U = (1 \dots s)x(1 \dots s)$. В процессе моделирования субдискретизирующего слоя часто применяются следующие параметры:

$$s = 2,\ \omega_{ij}^k = \frac{1}{s^2} = const, \quad (2.4)$$

Улучшение работы нейронной сети может быть достигнуто за счет применения нелинейной децимации. Один из самых популярных и эффективных методов субдискретизации – это слой max-pooling, который часто используется в больших сверточных сетях. В этом случае формула для отображения карт признаков имеет следующий вид:

$$y_j^k(x,y) = \max_{(u_x,u_y)\in U}\left(y_i^{k-1}(sx+u_x, sy+u_y)\right), \quad (2.5)$$

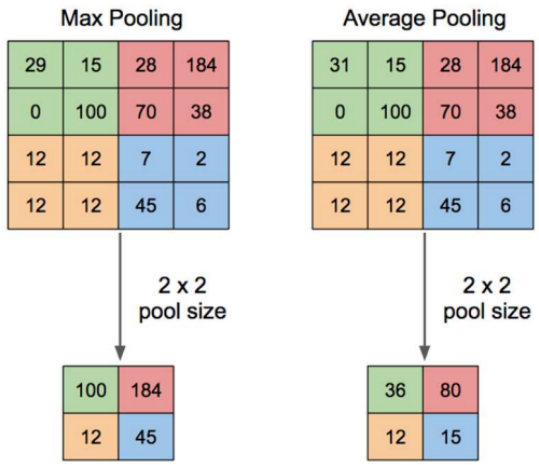


Рисунок 2.3 – Субдискретизирующий слой сверточной нейронной сети [58, с. 4]

Субдискретизирующий слой обычно служит мостом между сверточным слоем и полносвязным слоем. Данный слой CNN обобщает функции, извлеченные слоем свертки, и помогает сетям распознавать данные функции независимо. Благодаря этому сокращеются вычисления в сети.

В полностью подключенном слое на рисунке 2.4 (слева) каждый элемент подключен ко всем элементам предыдущих слоев. В сверточном слое (справа) каждый элемент подключен к постоянному числу элементов в локальной области предыдущего слоя. Кроме того, в сверточном слое все единицы измерения имеют общие веса для этих соединений, на что указывают общие типы линий.

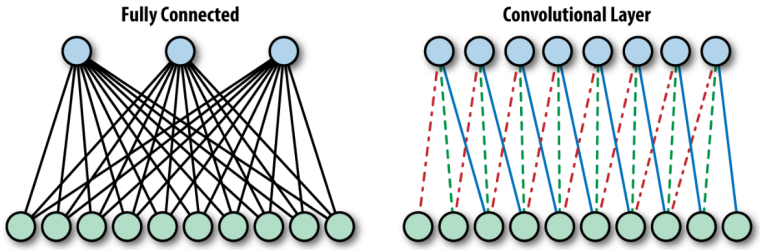


Рисунок 2.4 – Архитектура нейронной сети с полностью подключенным и сверточным слоем [59, с. 64]

В сетях CNN полносвязные слои чаще всего размещаются в завершающей части архитектуры, когда входные данные (изображение) уже были преобразованы в набор высокоуровневых характеристик. Типичная структура CNN включает несколько чередующихся сверточных и пуллинговых слоев, за которыми следуют один или несколько полносвязных слоев. Выходной сигнал от полносвязного слоя обычно генерирует предсказание, используя функцию активации, специально выбранную для конкретной задачи.

2.1.2 Реккурентные нейронные сети в задаче детектирования речевого сигнала

Сеть RNN [60] находят широкое применение в анализе последовательных данных, включая текст, речь, видео и временные ряды, где значения в текущий момент определяются предшествующими данными. Такие сети состоят из последовательных вычислительных блоков, которые осуществляют обработку информации. В отличие от прямосвязных нейронных сетей, RNN обладают внутренней памятью, что позволяет им хранить и учитывать его при анализе новых входящих данных.

Существуют разные архитектуры RNN: одно к многим, многие к одному, многие к многим. Обычно RNN обрабатывает входные данные поэтапно, принимая каждую последовательность отдельно. Блоки в скрытых слоях сохраняют информацию об истории входных данных в так называемом «векторе состояния» [61]. Обучение RNN осуществляется с помощью метода обратного распространения ошибки во времени (BPTT). Этот метод подразумевает, что градиенты потерь в любой момент времени t учитывают веса сети на предыдущих временных шагах. Процесс обучения RNN более сложен по сравнению с нейронными сетями с прямой связью (FFNN), и требует больше времени для завершения. На рисунке 2.5 представлен поток информации в скрытом слое RNN, разделенный на дискретные временные интервалы.

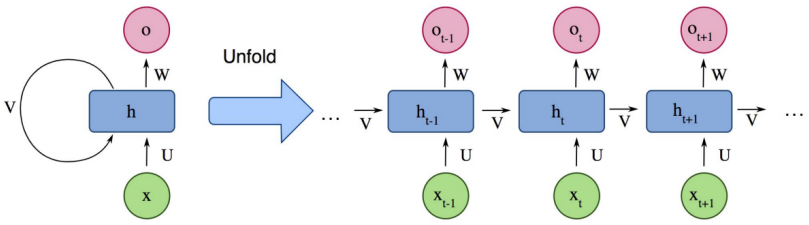


Рисунок 2.5 – Архитектура простой рекуррентной нейронной сети [62, с. 2]

Состояние узла h в разные моменты времени t обозначается как h, входное значение x в различные моменты времени представляется как x_t, а выходное значение o для каждого временного шага обозначается как o_t. Параметры $t(U, W, V)$ применяются последовательно на каждом шаге.

Состояние узла h в разные моменты времени t отображается как h, входное значение x в разные моменты времени равно x_t, а выходное значение o в разные моменты времени отображается как o_t. Значения параметров $t(U, W, V)$ всегда используются на одном и том же шаге. На каждой временной метке модель собирает входные данные из текущего времени x_i и скрытого состояния из предыдущего шага h_{i-1} и выводит целевое значение и новое скрытое состояние.

$$f_t = \sigma\left(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}\right), \tag{2.6}$$

$$i_t = \sigma\left(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}\right), \tag{2.7}$$

$$o_t = \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}\right), \tag{2.8}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\left(W^{(c)}x_t + U^{(c)}h_{t-1} + b^{(c)}\right), \tag{2.9}$$

$$h_t = o_t \odot \tanh(c_t), \tag{2.10}$$

где $x_t \in R^d$ – входной сигнал на временном шаге t, d – размер объекта для каждого слова, σ – сигмовидная функция, которая применяется к элементам (для нормализации значений в диапазоне [0, 1]), $\odot$ – поэлементное произведение. Символ c_t указывает на ячейку памяти, которая помогает уменьшить риск исчезновения или резкого увеличения градиента, что, в свою очередь, позволяет моделям изучать зависимости на более длительных временных интервалах, что особенно эффективно в традиционных RNN.

Элемент забывания f предназначен для сброса ячейки памяти. i_t и o_t обозначают элементы ввода и вывода, и соответственно управляют вводом и выводом ячейки памяти.

Таким образом, формулы (2.6) – (2.10) последовательно описывают основные этапы функционирования ячейки LSTM: формирование забывающего элемента, отвечающего за сохранение информации (2.6); входного элемента,

регулирующего поступление новых данных (2.7); выходного элемента, определяющего финальный вектор состояния (2.8); обновление содержимого памяти с учетом текущих и предыдущих значений (2.9); а также вычисление выходного состояния на основе обновленной ячейки памяти (2.10).

Одним из основных минусов RNN сетей является длительность процесса их обучения, который требует временных затрат из-за последовательной обработки входящих данных. Также главной проблемой остается способность RNN улавливать долгосрочные зависимости в последовательностях. Это связано с затруднениями в обработке информации, которая удалена во времени, что снижает эффективность таких моделей [63]. Однако тип RNNS, называемый долговременной кратковременной памятью (LSTM) [64], предназначен для того, чтобы избежать таких проблем.

2.1.2.1 Долговременная краткосрочная память LSTM

Одной из наиболее устойчивых к исчезающему градиенту разновидностей RNN является сеть с долгой краткосрочной памятью (Long Short-Term Memory, LSTM). Архитектура LSTM разработана для эффективного хранения и передачи информации на длинных участках последовательности за счет механизма внимания к значимым временным зависимостям, реализуемого через входной, выходной и логический элементы. Архитектура LSTM показана на рисунке 2.6, включает в себя три логических элемента (логический элемент ввода, логический элемент вывода, логический элемент забывания), которые регулируют поток информации в ячейку памяти и из нее, которая хранит значения в течение произвольных интервалов времени.

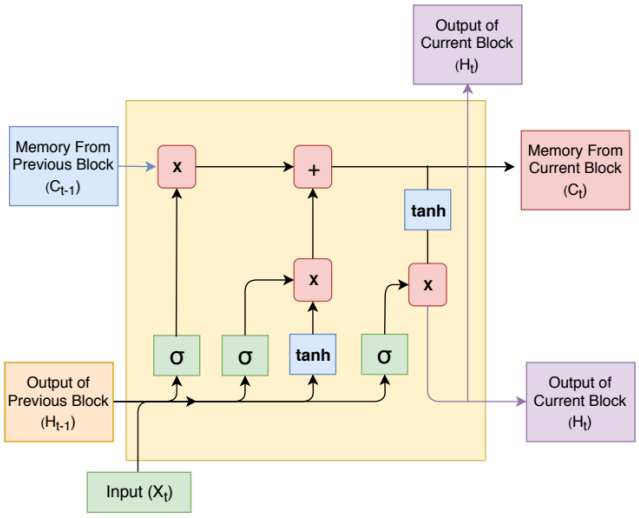


Рисунок 2.6 – Базовый модуль LSTM [55, с. 6]

В задаче детектирования речевого сигнала использование LSTM позволяет модели учитывать контекст между фреймами аудио, что особенно важно при классификации речевых сегментов, не имеющих ярко выраженных спектральных признаков. Благодаря способности LSTM моделировать долгосрочные

зависимости, она эффективно различает протяженные паузы и переходы между речью и шумом, а также улавливает временную структуру речи.

Однако при практическом применении LSTM в задаче детектирования речи выявляется ряд ограничений, снижающих ее эффективность. Основной недостаток LSTM заключается в односторонней направленности обработки последовательности. Поскольку модель анализирует данные строго в хронологическом порядке, от начала к концу сигнала, предсказания в конкретный момент времени основываются исключительно на прошлых фреймах, игнорируя будущий контекст. Для устранения указанных недостатков целесообразно использовать двунаправленные LSTM (BiLSTM), которые анализируют входной сигнал в двух направлениях: как вперед, так и назад. Такая архитектура позволяет формировать выходное представление для каждого временного фрейма с учетом как предшествующих, так и последующих фрагментов сигнала, что существенно повышает точность локализации границ между речью и шумом.

2.1.2.2 Двунаправленная долговременная краткосрочная память BiLSTM

BiLSTM (Bidirectional Long Short-Term Memory) – это расширение стандартной LSTM, при котором последовательность обрабатывается в двух направлениях одновременно: от начала к концу (forward) и от конца к началу (backward). Архитектура BiLSTM разработана для эффективного анализа последовательной информации в обоих временных направлениях, что позволяет учитывать как предшествующие, так и последующие элементы входной последовательности при обработке каждого временного шага. Архитектура BiLSTM, представлена на рисунке 2.7, включает в себя два параллельных слоя LSTM, обрабатывающих входную последовательность в противоположных направлениях.

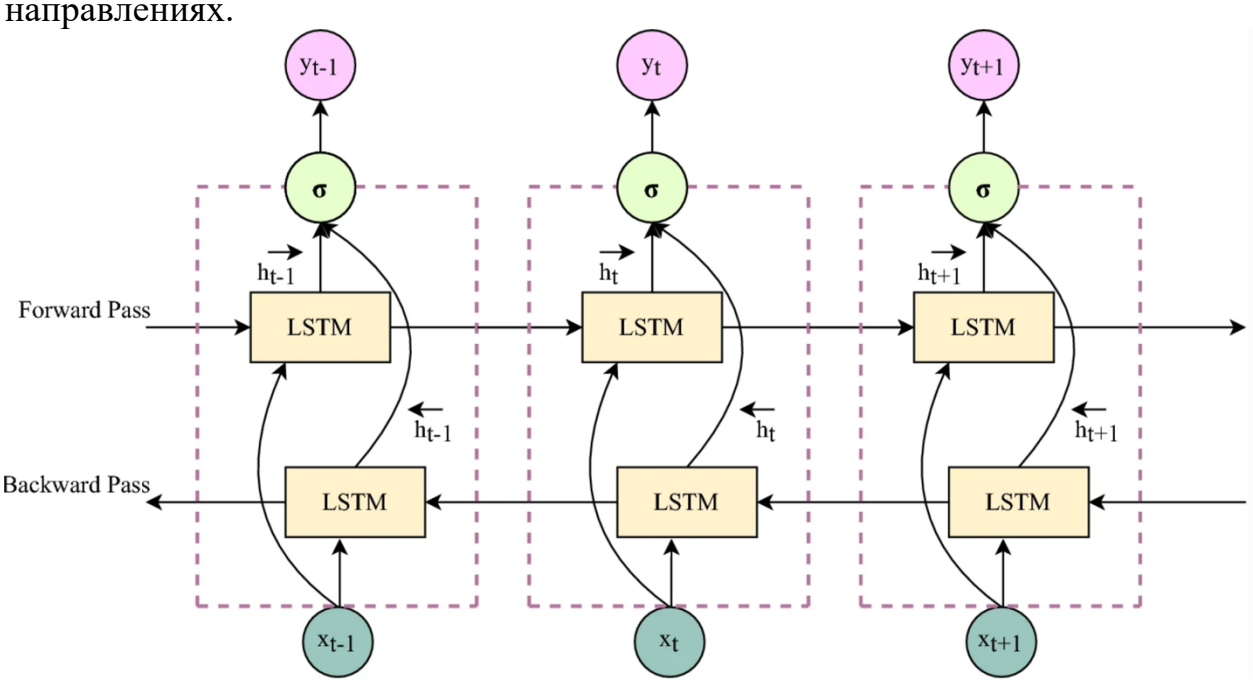


Рисунок 2.7 – Архитектура BiLSTM [65, с. 13]

Верхний слой осуществляет прямой проход (forward pass), от начала последовательности к ее завершению, в то время как нижний слой выполняет

обратный проход (backward pass), от конца к началу. На каждом временном шаге **t** входной вектор x_t подается одновременно в оба слоя. Каждая из LSTM-ячеек формирует собственное скрытое состояние h_t, отражающее контекст либо из прошлого, либо из будущего. Затем оба скрытых состояния конкатенируются и передаются на слой активации σ который формирует окончательный выход y_t для текущего фрейма [65].

2.1.2.3 Рекуррентные нейронные сети с управляемым блоком GRU

Gated Recurrent Unit (GRU) представляет собой одну из разновидностей рекуррентных нейронных сетей RNN, разработанную с целью более эффективной обработки последовательных данных и устранения проблем, характерных для классических RNN, таких как затухающие или взрывающиеся градиенты. В отличие от стандартных RNN, GRU включает в себя блоки, которые регулируют поток информации внутри ячейки, позволяя модели запоминать или забывать определенные элементы входной последовательности.

На рисунке 2.8 представлена внутренняя структура ячейки GRU, предназначенной для обработки последовательных данных с сохранением временных зависимостей.

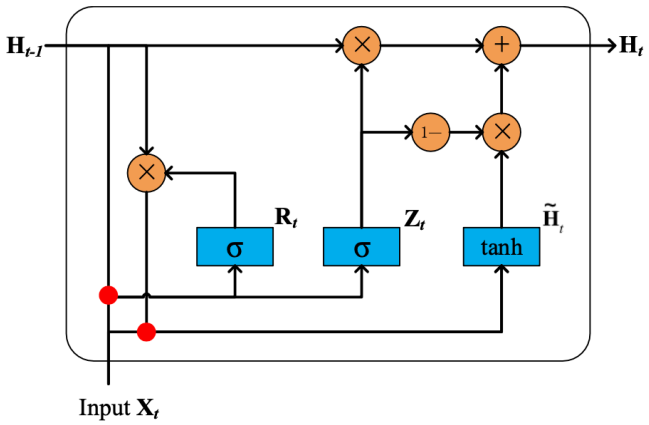


Рисунок 2.8 – Архитектура GRU [66, с. 6]

Входными параметрами являются вектор признаков на текущем временном шаге X_t и скрытое состояние с предыдущего шага H_{t-1}. Архитектура GRU использует два основных управляющих механизма: вентиль обновления Z_t и вентиль сброса R_t, оба из которых формируются с использованием сигмоидной активации σ.

Вентиль обновления:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \quad (2.12)$$

Вентиль сброса:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \quad (2.13)$$

Кандидатное скрытое состояние:

$$\widetilde{h_t} = \tanh(W_r x_t + U_h(r_t \odot h_{t-1}) + b_h)\,, \qquad (2.14)$$

Обновленное скрытое состояние:

$$h_t = (1 - z_t) \odot \widetilde{h_t} + z_t \odot h_{t-1}, \qquad (2.15)$$

где x_t – входной вектор в момент времени t; h_{t-1} – скрытое состояние на предыдущем шаге; $\odot$ –поэлементное умножение; σ – сигмоида; *tanh* – гиперболический тангенс; W_*, U_*, b_* –матрицы весов и смещений.

Хотя классическая архитектура GRU эффективно справляется с задачей моделирования временных зависимостей в последовательностях, она делает это в одном направлении, от предыдущих элементов к последующим. Однако в ряде задач, таких как VAD, критически важно учитывать как предшествующий, так и последующий контекст. Поскольку признаки речевого сигнала на текущем временном шаге могут быть недостаточно информативными для однозначного определения наличия речи, использование информации как из прошлого, так и из будущего позволяет модели более точно идентифицировать границы речевых сегментов, особенно в условиях шума.

Именно в таких условиях двунаправленная архитектура, такая как Bidirectional GRU (BiGRU), демонстрирует особую эффективность. За счет обработки сигнала одновременно в двух направлениях BiGRU позволяет учитывать полный временной контекст при анализе каждого момента аудиосигнала.

2.1.2.4 Двунаправленная рекуррентная нейронная сеть с управляемым блоком

Bidirectional GRU представляет собой расширение классической архитектуры GRU, предназначенное для более глубокого анализа последовательностей за счет учета как предыдущего, так и последующего контекста. На рисунке 2.9 представлена архитектура двунаправленной рекуррентной нейронной сети на основе GRU. Слева показан макроуровень взаимодействия слоев, справа, структура отдельной GRU-ячейки.

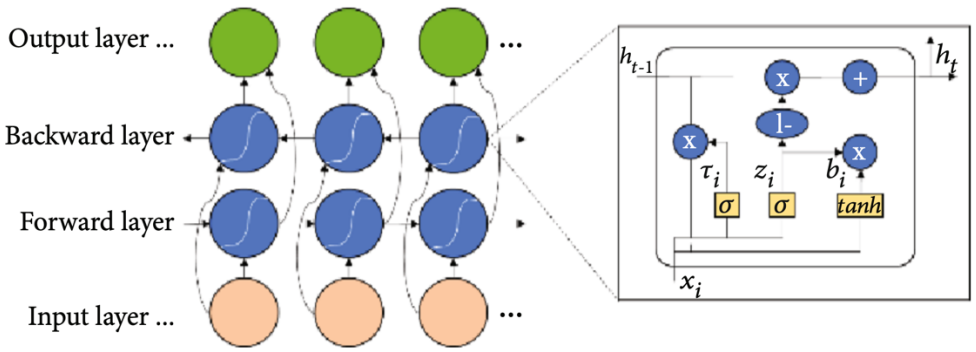


Рисунок 2.9 – Архитектура BiGRU [67, с. 7]

BiGRU использует две GRU-ячейки: прямую и обратную. Их выходы объединяются следующим образом:

$$\vec{h}_t = GRU_{forward}\left(x_t, \vec{h}_{t-1}\right), \quad (2.16)$$

$$\overleftarrow{h}_t = GRU_{backward}\left(x_t, \overleftarrow{h}_{t+1}\right), \quad (2.17)$$

$$h_t = concat\left(\vec{h}_t, \overleftarrow{h}_t\right), \quad (2.18)$$

Формулы (2.16) – (2.18) описывают механизм работы BiGRU: (2.16) – вычисление скрытого состояния в прямом направлении, (2.17) – в обратном, (2.15) – объединение результатов для учета контекста с обеих сторон.

Такая архитектура позволяет модели одновременно учитывать как прошлый, так и будущий контекст для каждого элемента входной последовательности. Это особенно ценно в задачах обнаружения речевой активности (VAD), где признаки могут быть неоднозначны без информации о ближайших соседних фреймах. BiGRU эффективно решает эту проблему, обеспечивая более точную классификацию сегментов речи и пауз.

2.1.3 Нейронная сеть с временной задержкой TDNN

В рамках данной диссертационной работы также было принято решение о проведении серии экспериментов с использованием архитектуры TDNN для решения задачи детектирования голосовой активности. В отличие от RNN-базированных моделей, TDNN оперирует сверточными слоями с временными смещениями, что позволяет ей захватывать широкий временной контекст без необходимости последовательной обработки входных данных. Это делает TDNN более устойчивой к колебаниям длины входного сигнала и обеспечивает высокую скорость вычислений за счет параллельной обработки фреймов. С математической точки зрения, каждый слой TDNN применяет операцию свертки по заданным временным окнам, охватывая как предшествующие, так и последующие временные шаги, что особенно актуально при распознавании коротких речевых событий на фоне шумов.

На рисунке 2.10 изображена архитектура TDNN, в которой входной сигнал u(t) и его временные задержки $u(t-\tau), u(t-2\tau), \ldots, u(t-N_d\tau)$ подаются на первый слой нейронной сети. Далее данные проходят через несколько полносвязных слоев с фиксированной архитектурой.

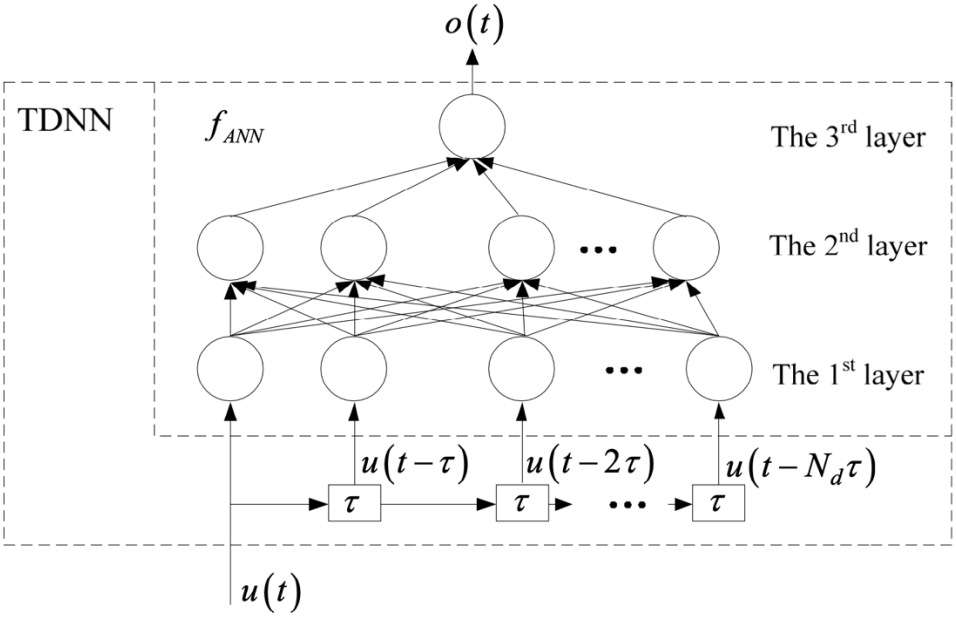


Рисунок 2.10 – Архитектура TDNN [68, с. 3]

Обобщенная формула TDNN представлена на данной формуле, формирование выходного сигнала *o(t)* на основе текущего и предыдущих значений входного сигнала.

$$o(t) = f_{ANN}(u(t), u(t-\tau), \dots u(t-N_d\tau), w), \quad (2.19)$$

где $o(t)$ – выход нейросети в момент времени t; $u(t)$ – входной сигнал в момент t; τ – шаг временной задержки; N_d – количество задержек, глубина временного окна; w – совокупность весов нейросети; f_{ANN} – функция, реализуемая многослойной нейросетью

Таким образом, были рассмотрены и проанализированы различные типы нейронных сетей, применяемых для обработки последовательных данных и акустических признаков. В частности, были изучены CNN, GRU, BiGRU, LSTM, BiLSTM, а также TDNN. Каждая из перечисленных архитектур обладает своими особенностями в плане извлечения пространственно-временных зависимостей, устойчивости к шуму и вычислительной сложности.

На основе проведенного анализа было принято решение о разработке и сравнительном исследовании гибридных нейросетевых архитектур, сочетающих преимущества различных подходов. Также планируется проведение экспериментов по задачам детектирования голосовой активности с использованием следующих моделей таких как, CNN+BiGRU, CNN + GRU, CNN + BiLSTM, CNN+LSTM и CNN+TDNN

2.2 Особенности разработки алгоритма распознавания речи при низком отношении С/Ш

Распознавание речи играет основную роль в развитии исследований в области искусственного интеллекта, поскольку позволяет существенно улучшить взаимодействие между человеком и компьютером. Однако, низкое отношение

С/Ш ставит перед данным процессом определенные сложности, которые усложняют задачу распознавания речи. Достижение высокой точности распознавания при низком С/Ш является актуальной и вызывающей интерес проблемой, поскольку это имеет прямое практическое применение в различных областях. Исследования направлены на разработку методов, которые улучшают распознавание речи при низком соотношении сигнал/шум. Ученые по всему миру работают над подходами, такими как нейронные сети, чтобы преодолеть вызовы и повысить точность распознавания в данных условиях.

Для преодоления сложностей распознавания речи при низком отношении сигнал/шум успешно могут применяться различные типы нейронных сетей, таких как сверточные нейронные сети CNN, рекуррентные нейронные сети RNN и многослойные персептроны MLP. Каждый из этих типов нейронных сетей обладает своими уникальными особенностями и возможностями, которые могут быть эффективно использованы для решения задачи распознавания речи при низком отношении сигнал/шум.

Выбор оптимального типа нейронной сети зависит от конкретной задачи распознавания речи при низком отношении сигнал/шум и требует тщательного анализа его особенностей и специфики. В сочетании с соответствующими методами обучения такие типы нейронных сетей могут быть эффективно применены для успешного решения данной проблемы.

Основное внимание при исследовании уделено на определение функциональной зависимости точности распознавания человеческого голоса нейронными сетями от количества дикторов, использованных при создании обучающих данных.

Точность работы нейронных сетей зависит от множества факторов, таких как количество дикторов (a), соотношение сигнал/шум (b), количество слоев сети (c) и другие параметры. В общем виде это можно описать функцией:

$$y = f(a, b, c, \dots), \tag{2.20}$$

Если во время обучения и тестирования нейронных сетей сделать все параметры константами, кроме количества дикторов, то мы будем рассматривать одномерную функцию. Таким образом, функция, указанная в (1) принимает следующий вид:

$$y = f(a), \tag{2.21}$$

Для того, чтобы добиться зависимости точности распознавания человеческого голоса нейронными сетями только от количества дикторов, данные для обучения нейронной сети были формированы следующим образов:

- создавался длинный аудиофайл, состоящий только из разных *шумов;*
- создавался аудиофайл, состоящий из *K* количества голосов казахского языка. Количество дикторов К варьируется от 2 до 40 с шагом 2, включая 1 мужской и 1 женский голос на каждом шаге;

- Если аудиофайл с *K* голосами был короче файла шума, голоса дублировались, чтобы выровнять длину файлов;

- затем два аудиофайла смешивались, устанавливая SNR на уровне 20 дБ;

- из смешанного аудиофайла рассчитываются 36 коэффициентов *MFCC* (с учетом delta и delta-delta*),* которые служат в качестве набора обучающих входных данных.

Сформированные файлы, состоящие из *MFCC* коэффициентов, были использованы для обучение нейронных сетей. В зависимости от типа нейронной сети, *MFCC* были скомпонованы по-разному. Для нейронной сети MLP обучаемые данные подавались в виде вектора длиной 36. А именно использовались все 36 коэффициентов *MFCC.*

При работе с нейронными сетями CNN, RNN использовались только первые 12 коэффициентов *MFCC*. А входные данные нейронных сетей принимала матрицу в виде 12x3, где были последовательно скомбинированы коэффициенты *MFCC*. Количество параметров нейронных сетей и обучаемых данных представлено в таблице 2.1

Таблица 2.1 – Количество параметров нейронных сетей и обучаемых данных

Нейронная сеть	Общее количество обучаемых данных	Количество параметров нейронной сети
CNN	1 393 100	98 657
MLP	1 507 403	97 985
RNN	1 393 103	98 981

Для обучения всех нейронных сетей использовался оптимизатор Adam и функция Binary Crossentropy для расчета потерь. Для нейронной сети RNN была выбрана функция потерь Mean Squared Error. В процессе обучения были применены методы ранней остановки (EarlyStopping) и сохранения лучших моделей (ModelCheckpoint). Механизм EarlyStopping автоматически завершал обучение, если значение «val_accuracy» не улучшалось в течение 10 эпох. С другой стороны, ModelCheckpoint сохранял модель с максимальным значением «val_accuracy» на каждой эпохе обучения. Такие стратегии обучения позволили оптимизировать процесс обучения, предотвратить переобучение и сохранить наилучшие результаты обучения в каждый момент времени. Структуры нейронных сетей CNN, RNN и MLP показаны на рисунках 2.11 – 2.13.

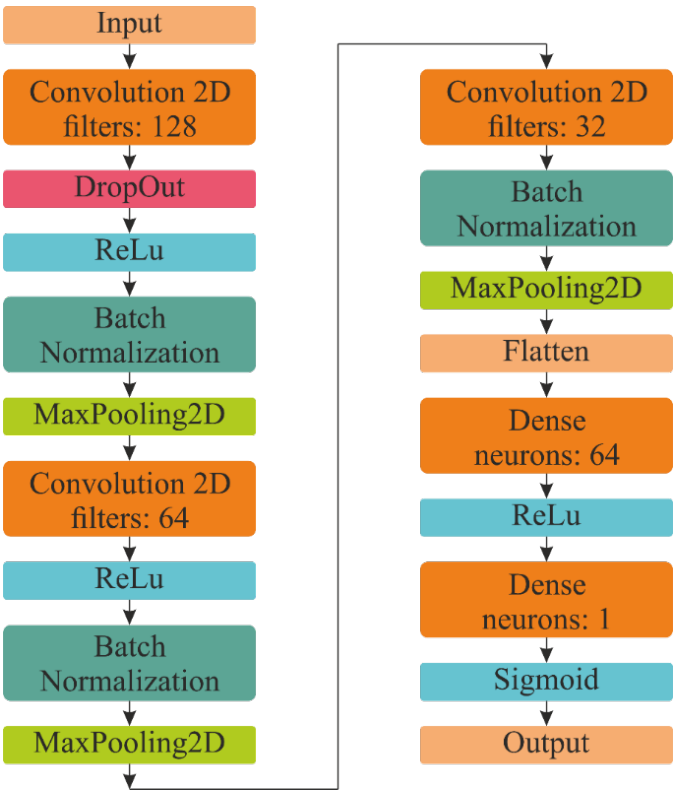


Рисунок 2.11 – Структура сетей CNN

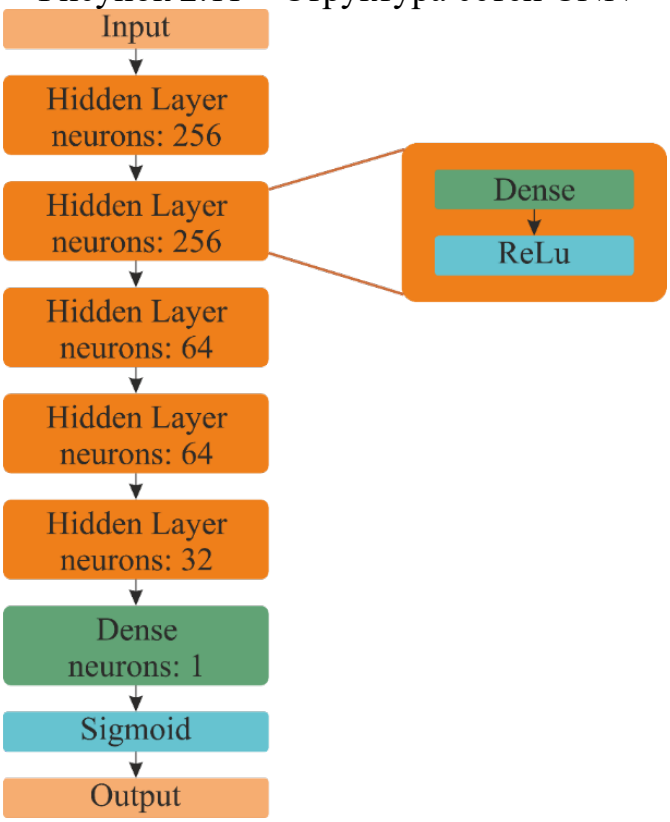


Рисунок 2.12 – Структура сетей MLP

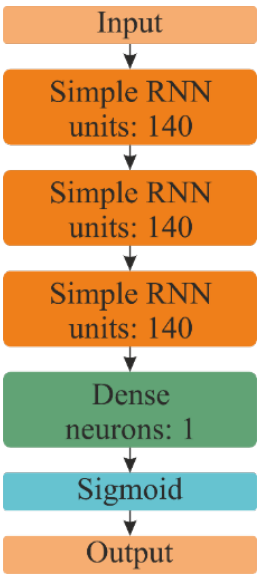


Рисунок 2.13 – Структура сетей RNN

В связи с вышеизложенным, данная диссертационная работа позволяет более глубоко понять важность и перспективы использования нейронных сетей CNN, RNN и MLP в задаче распознавания речи при низком уровне С/Ш. Результаты исследования подтверждают значимость выбора оптимального типа нейронной сети для успешного решения данной задачи, а также показывают, что применение современных методов обучения, таких как оптимизатор Adam, функция Binary Crossentropy и механизмы EarlyStopping и ModelCheckpoint, способствует повышению эффективности и точности системы распознавания речи. На основе данных результатов можно рекомендовать использование определенного типа нейронной сети в зависимости от конкретных условий задачи для достижения оптимальных результатов. Дальнейшие исследования в этом направлении могут способствовать развитию технологий распознавания речи и улучшению взаимодействия между человеком и компьютером.

2.3 Результаты анализа эффективности нейронных сетей по распознаванию речевого сигнала

В условиях современного мира, где технологии искусственного интеллекта и машинного обучения активно развиваются, вопрос распознавания речи приобретает все большую важность и становится весьма востребованным. Одним из самых эффективных подходов к решению данной задачи является применение искусственных нейронных сетей [69]. Эти сети способны адаптироваться к различным языкам, диалектам, акцентам и даже условиям записи, что делает их мощным инструментом для распознавания и обнаружения речевого сигнала на множестве языков [70].

Создание метода для распознавания речевого сигнала с помощью искусственных нейронных сетей является сложным и многокомпонентным процессом, который требует глубокого понимания принципов работы нейронных сетей, обработки сигналов, глубокого и машинного обучения [71]. Процесс включает несколько важных этапов, таких как подготовка и аннотирование

данных, а также предобработка аудиофайлов . Важным шагом является выбор наиболее подходящей архитектуры нейронной сети. Обучение модели будет проводиться на небольшой группе дикторов, что обеспечит уникальность данных. После завершения обучения предусмотрено тестирование, оценка эффективности и анализ полученных результатов [72, 73].

Изучение и создание методов обнаружения речевых сигналов с использованием искусственных нейронных сетей, таких как сверточные нейронные сети CNN, рекуррентные нейронные сети RNN и многослойные перцептроны MLP, является важной задачей для повышения эффективности и качества распознавания речевых сигналов. Данные методы в будущем могут найти применение в различных областях, включая автоматизацию обработки аудиосигналов, разные приложения-ассистенты, которые работаю с помощью голосовых команд, систем безопасности и другие сферы [74].

В рамках исследования данной диссертационной работы проведен комплексный анализ эффективности нейронных сетей различных архитектур – CNN, RNN и MLP – в задаче распознавания речевых сигналов. Основное внимание уделено оценке влияния количества дикторов в обучающем наборе данных на точность работы моделей. Кроме того, рассматривалась способность нейросетей, обученных исключительно на данных казахского языка, распознавать высказывания на других языках. Оценка качества моделей проводилась с использованием стандартных метрик. При этом следует учитывать, что обобщенная ошибка распознавания состоит из двух компонентов:

1. Ошибочное распознавание участка, не содержащего речевые данные, как человеческий голос (False positive).

2. Ошибочное распознавание участка, содержащего речевые данные, как не человеческий голос (False negative).

На рисунке 2.14 представлен результат тестирования обобщенной точности распознавания человеческого голоса нейронной сетью типа CNN в зависимости от количества дикторов.

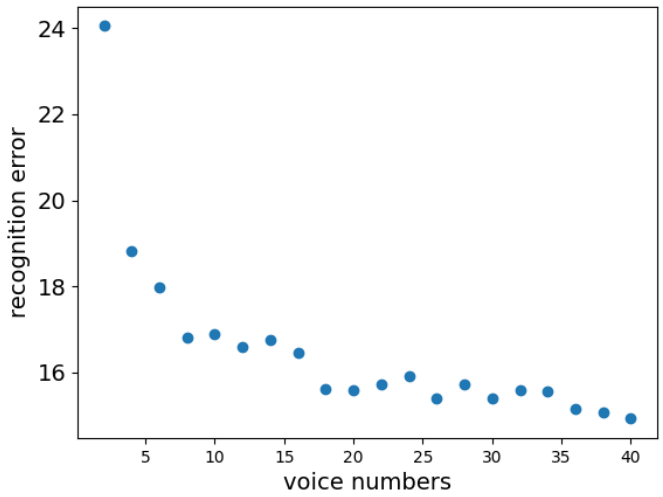


Рисунок 2.14 – Экспериментально измеренная зависимость общей ошибки распознавания речевого сигнала с помощью CNN от количества дикторов

Из рисунка 2.14 четко видно, что общая ошибка распознавания речевого сигнала с помощью CNN в зависимости от количества использованных дикторов при формировании обучающих данных, все время уменьшается. Также можно заметить, что данная зависимость явно нелинейная. В таком случае, для описания функционального вида этой зависимости, приведенной на рисунке 2.14, можем рассмотреть различные убивающиеся нелинейные функции следующих типов:

$$y = C \cdot x^n, \tag{2.22}$$

$$y = C \cdot n^x, \tag{2.23}$$

$$y = C \cdot e^{n \cdot x}, \tag{2.24}$$

$$y = a + b \cdot \log(x), \tag{2.25}$$

где y – ошибка распознавания, x – количество дикторов, a,b,C and n – некие эмпирические вещественные числа, которые определяются с помощью регрессионного анализа экспериментальных данных.

Таким образом, используя формулы (2.23) – (2.25), мы можем определить функцию, наиболее точно характеризующую зависимость, представленной на рисунке 2.14. Наиболее подходящая функция выбирается на основе вычисления ошибки определения значимых коэффициентов аппроксимирующих функций по экспериментальным данным, например, для функций (2.23) – (2.24) значимым параметром является n, а для функции (2.25) параметр b. В нижеследующей таблице 2.2 приведены ошибки определения значимых параметров аппроксимирующих функций для трех типов нейронных сетей.

Таблица 2.2 – Значение ошибки определения значимых параметров аппроксимирующих функций

Сеть, функция	$C \cdot x^n$	$C \cdot n^x$	$C \cdot e^{n \cdot x}$	$a + b \cdot \log(x)$
CNN	16%	33%	34%	19%
RNN	15%	34%	34%	19%
MLP	14%	31%	31%	15%

Как показывают данные из таблицы 2.2, самой лучшей функцией, наиболее точно описывающей зависимость ошибки распознавания речевого сигнала от количества дикторов по всем видам рассмотренных нейронных сетей, является степенная функция вида (2.22).

На рисунке 2.15 показан пример графика аппроксимирующей функции (3) для сети CNN, обученной только на казахском языке, и протестированной на высказываниях также на казахском языке.

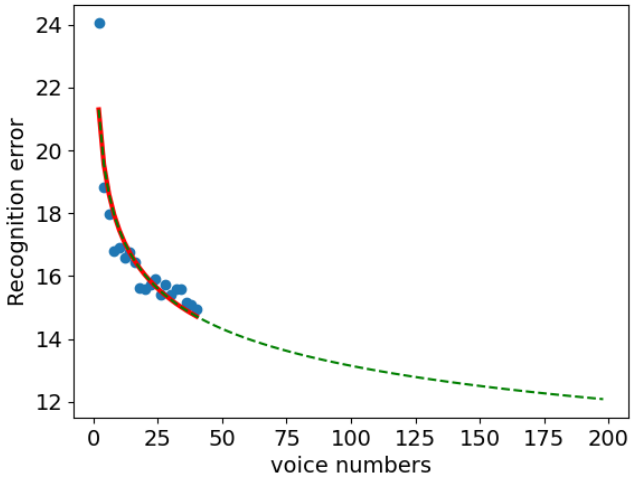


Рисунок 2.15 – Зависимость общей ошибки распознавания сети CNN речевого сигнала от количества дикторов

На этом рисунке 2.15 точками обозначены экспериментальные данные, красная линия показывает участок функции аппроксимации, соответствующего диапазону экспериментальных данных (от 2 до 40 дикторов), пунктирная зеленая часть соответствует прогнозным значениям функции аппроксимации. А в таблице 2.3, в качестве примера приведены все найденные функции аппроксимации для всех трех типов нейронных сетей при тестировании на высказываниях на казахском языке.

Таблица 2.3 – Вид аппроксипирующей функции зависимости точности распознавания речевого сигнала от количества дикторов для разных типов нейронных сетей

№	Нейронная сеть	Аппроксимирующая функция
1	CNN	$y = 23{,}2 \cdot x^{-0{,}123}$
2	RNN	$y = 24{,}36 \cdot x^{-0{,}144}$
3	MLP	$y = 39{,}05 \cdot x^{-0{,}096}$

На рисунках 2.16 и 2.17 приведены графики аппроксимирующих функций для нейронных сетей RNN и MLP, построенные согласно данным таблицы 2.3.

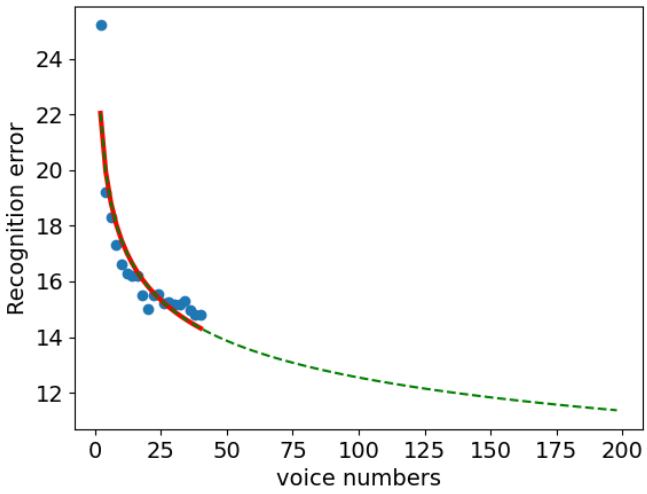


Рисунок 2.16 – Зависимость общей ошибки распознавания сети RNN речевого сигнала от количества дикторов

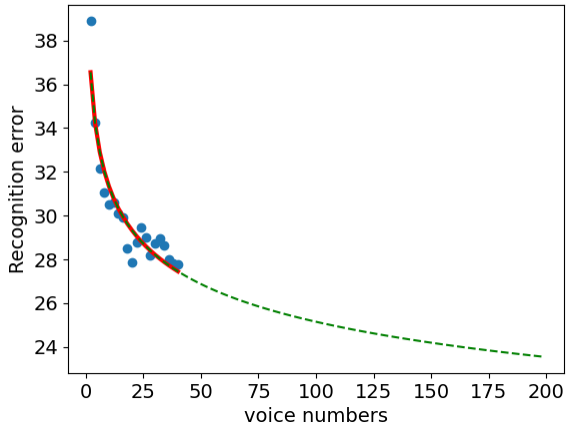


Рисунок 2.17 – Зависимость общей ошибки распознавания сети MLP речевого сигнала от количества дикторов

А на рисунке 2.18 приведены все три графика, показанные на рисунках 2.15, 2.16 и 2.17, на одной диаграмме для сравнения их между собой.

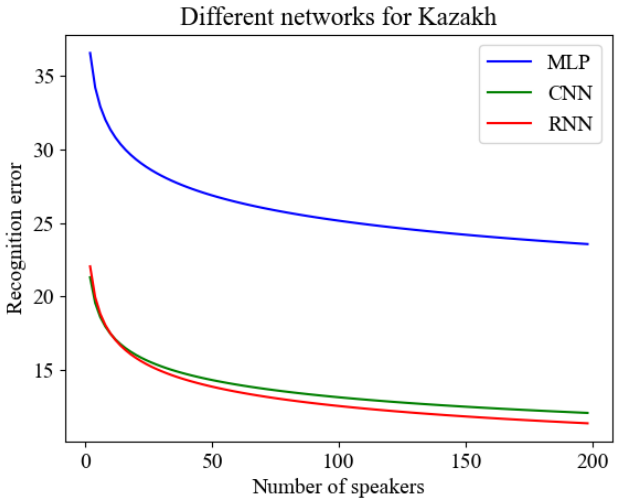


Рисунок 2.18 – Зависимость общей ошибки распознавания речевого сигнала от количества дикторов

Рисунок 2.18 дает ответ на вопрос, какой из нейронных сетей CNN, RNN или MLP дает наилучшие результаты при распознавании речевого сигнала. Как видим из этого рисунка, наилучшие результаты предоставляет нейронная сеть типа RNN. Также, используя виды аппроксимирующих функций из таблицы 2.3, можем приблизительно рассчитать, сколько различных дикторов надо использовать при обучении нейронной сети для достижения определенной точности распознавания человеческого голоса. Для этого можем использовать следующую формулу:

$$x = e^{\frac{\ln(y)-\ln(c)}{n}}, \tag{2.26}$$

где x – требуемое количество дикторов, y – необходимая точность распознавания сети, c и n – параметры аппроксимирующей функции.

Например, для точности работы сети 95% соответствует значение $y = 5$. А значение параметров c и n берем из таблицы 2.4. Тогда для сети RNN если требуется точность распознавания человеческого голоса не менее 95%, то согласно расчету по формуле (2.26) требуется использовать порядка 59 668 различных дикторов при обучении данной сети. Также расчеты показывают, что при использовании такого же количества дикторов при обучении, равному 59 668, нейронная сеть CNN дает точность распознавания 94%, т.е. чуть похуже, чем сеть RNN.

После того, как выяснили, что при прочих равных условиях речевой сигнал наилучшим образом распознается сетью типа RNN, рассмотрели другой вопрос. Этот вопрос звучит следующим образом: нейросеть, обученная на данных одного языка, сможет ли распознавать человеческий голос по данным на другом языке с той же эффективностью. Чтобы получить ответ на данный вопрос, было проведено тестирование нейронных сетей, обученных только на данных на казахском языке, с помощью данных на других языках. Результат данного тестирования на сети типа RNN представлен на рисунке 2.19.

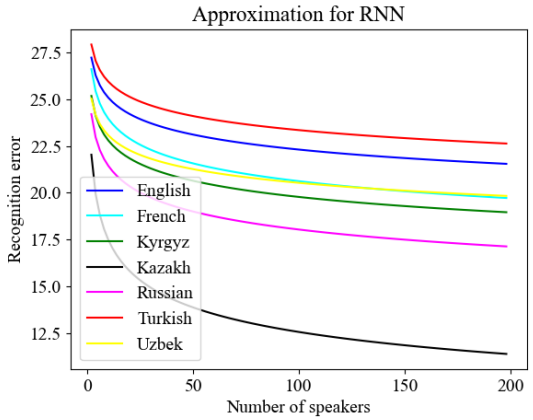


Рисунок 2.19 – Зависимость точности распознавания речевого сигнала сетью типа RNN от количества дикторов, при обучении сети данными на казахском языке, но при распознавании речевого сигнала на других языках

Необходимо отметить, что зависимость ошибки распознавания речевого сигнала на других языках также, как в случае казахского языка, имеет степенную форму. К примеру, данная зависимость для русского языка имеет следующий вид:

$$y = 25{,}53 \cdot x^{-0{,}075}, \tag{2.27}$$

Как видим из рисунка 2.19, нейросеть, обученная только на данных на казахском языке, довольно плохо распознает речевой сигнал на других языках. Например, точность распознавания речевого сигнала на русском языке для нейронной сети RNN, обученной с использованием 59 668 дикторов, согласно формуле (2.27), имело бы значение, равное 89%. Это по сравнению с 95% точностью для казахского языка довольно хуже, но не сильно. Значения точности распознавания речевого сигнала нейронной сетью типа RNN для всех остальных языков представлены в таблице 2.4.

Таблица 2.4 – Предполагаемая точность распознавания речевого сигнала на разных языках сетью типа RNN, обученной на казахском языке для 59 668 дикторов

№	Язык	Точность распознавания
1	Kazakh	95,0%
2	Russian	88,85%
3	Kyrgyz	86,68%
4	French	86,41%
5	Uzbek	85,15%
6	English	83,89%
7	Turkish	82,57%

Как видим из таблицы 2.4, точность распознавания речевого сигнала все же ощутимо зависит от языка, если сеть обучать на одном языке, при этом ее тестировать на данных другого языка.

Однако здесь появляется одна большая проблема. Она связана с тем, что для того, чтобы получить точность свыше 95% распознавания речевого сигнала с помощью сети типа RNN, требуется использовать образцы голосов почти 60-ти тысяч дикторов. На наш взгляд это будет составлять слишком большой набор обучающих данных, подготовка которых требует огромных ресурсов. Но эту проблему можно довольно сильно облегчить, если рассмотреть не общую ошибку сети, а только одну ее часть, связанной с ошибочным распознаванием участка, содержащего речевые данные, как не человеческий голос (False Negative).

Действительно, для VAD-систем достаточным и необходимым условием является не потерять речевые данные при анализе звуковых данных. Тогда нейронную сеть можно обучать так, чтобы минимизировалась только ошибка False Negative. На рисунке 2.20 приведена зависимость ошибки False Negative сети типа RNN от количества дикторов при тестировании на данных казахского языка.

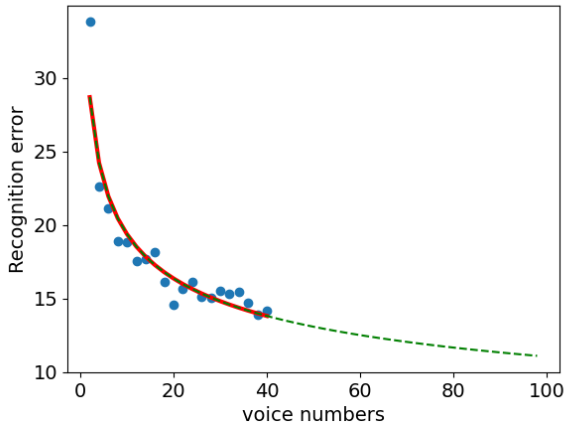


Рисунок 2.20 – Зависимость типа ошибки распознавания False Negative сети RNN от количества дикторов при тестировании на данных казахского языка

Зависимость на рисунке 2.20 также, как в предыдущих случаях имеет степенной характер, определяемый формулой:

$$y = 34 \cdot x^{-0,244}, \tag{2.28}$$

В таком случае, для того чтобы получить точность 95% правильного распознавания только речевых участков требуется всего 2582 образцов человеческого голоса. Как видите, это гораздо меньшее количество, чем требуется при попытке обеспечить 95% точности по общей ошибке. На рисунке 2.21 на одной диаграмме приведены графики зависимости ошибок распознавания для общей ошибки и ошибки типа False Negative сети RNN от количества дикторов при тестировании на данных казахского языка.

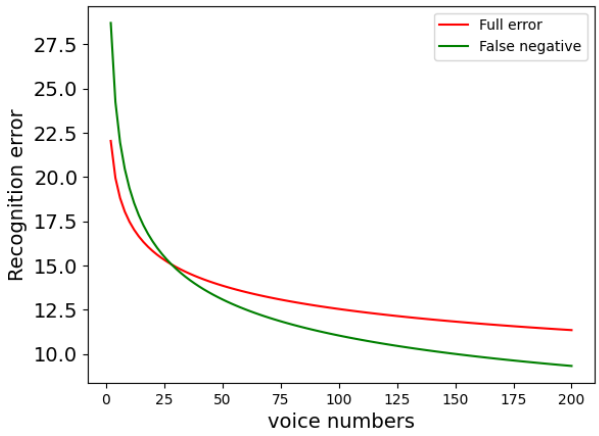


Рисунок 2.21 – Зависимости общей ошибки и типа ошибки False Negative сети RNN от количества дикторов (тестовые данные на казахском языке)

Графики зависимостей на рисунке 2.21 построены с помощью соответствующих функций аппроксимации. Как видим, из рисунка 2.21, функция ошибки типа False Negative быстрее убывает с ростом количества дикторов, использованных при формировании обучающих данных. Оно и понятно, так как с ростом количества дикторов участки звукового сигнала, содержащие человеческую речь, будут распознаваться точнее. А вот увеличение количества дикторов очень слабо влияет на точность распознавания не речевых участков. По этой причине общая ошибка, содержащая оба типа ошибок, более медленно убывает с ростом количества используемых дикторов при обучении.

Необходимо отметить, что нейронная сеть, обученная так, чтобы минимизировалась общая ошибка, может работать как самостоятельная VAD-система, так как она учитывает все типы ошибок. А сеть, ориентированная на минимизацию только ошибки False Negative, не может самостоятельно выполнять в полном объеме функции VAD-систем. Это из-за того, что она фактически не контролирует ошибок, связанных с распознаванием неголосовых участков как голосовые. Вообще нейронную сеть, ориентированную на минимизацию ошибки False Negative, можно использовать в комплекте с традиционными VAD-системами, работающими на основе анализа энергетических и спектральных характеристик сигнала. Схема возможной работы такой гибридной системы представлена на рисунке 2.22.

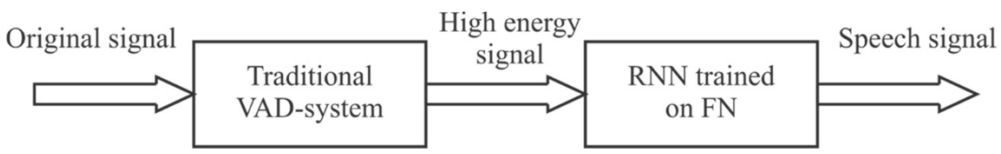


Рисунок 2.22 – Схема возможной работы комбинированной VAD системы

Согласно схеме, показанной на рисунке 2.22, нейронная сеть типа RNN, обученная на минимизации одного только типа ошибки False Error, используется

как дополнительный элемент, повышающий качество работы традиционных VAD-систем.

Выводы по главе:

1. Установлено, что среди негибридных архитектур наибольшую эффективность в задаче детектирования речевого сигнала демонстрируют CNN, благодаря способности извлекать локальные спектральные признаки и устойчивости к шуму, обеспеченной применением операций субдискретизации. Рекуррентные архитектуры, такие как LSTM и GRU, обеспечивают моделирование длительных временных зависимостей за счет механизма памяти, что позволяет эффективно обрабатывать непрерывные аудиопотоки. Использование двунаправленных структур BiLSTM и BiGRU способствует учету как предшествующего, так и последующего контекста, улучшая идентификацию границ речи. Кроме того, TDNN-сети, основанные на применении временных окон с фиксированными смещениями, обеспечивают высокую скорость обработки и устойчивость к изменяющейся длительности входного сигнала.

2. Установлено, что применение гибридных нейросетевых архитектур, сочетающих сверточные и рекуррентные слои позволяет достичь высокой точности распознавания речевого сигнала даже при низком отношении С/Ш. Такие модели эффективно объединяют преимущества пространственного анализа, характерного для CNN, с возможностями моделирования временных зависимостей, реализуемыми через рекуррентные компоненты. Архитектуры CNN+BiGRU и CNN+BiLSTM способны учитывать двунаправленный контекст, что улучшает идентификацию границ речевых сегментов.

3. Установлено, что эффективность распознавания речевого сигнала нейронными сетями зависит от количества дикторов, используемых при обучении. Также эффективность нейронных сетей в распознавании речи зависит от языка, на котором происходило обучение сети. Исследование подтверждает, что комбинирование традиционных методов с нейросетевыми решениями является оптимальным подходом для создания эффективных VAD-систем. Подход, включающий использование нейронной сети для минимизации ошибок False Negative, позволяет значительно сократить объем необходимых обучающих данных и достичь высокой точности распознавания без чрезмерного использования ресурсов.

3 РАЗРАБОТКА И ЭКСПЕРИМЕНТАЛЬНЫЙ АНАЛИЗ ИНТЕЛЛЕКТУАЛЬНЫХ МОДЕЛЕЙ ДЛЯ ДЕТЕКТИРОВАНИЯ РЕЧЕВОГО СИГНАЛА

3.1 Методологические основы подготовки и извлечения признаков MFCC для распознавания речевого сигнала

В рамках данной диссертационной работы для обучения и тестирования нейронных сетей, таких как CNN + BiGRU, CNN + GRU, CNN + BiLSTM, CNN + LSTM, CNN + TDNN, были использованы наборы данных Института умных систем и искусственного интеллекта (Institute of Smart Systems and Artificial Intelligence, ISSAI) Назарбаев Университета, а именно, был использован речевой корпус казахского языка (КРК) [75]. Данный корпус содержит 332 часа расшифрованных аудиозаписей с более чем 153 000 высказываний различных участников из разных регионов, который представлен на рисунке 3.1, а также на рисунке 3.2 представлено соотношение мужчин и женщин, принявших участие в записи аудиоданных, использованных в рамках данного исследования. Качество данных тщательно проверено носителями казахского языка.

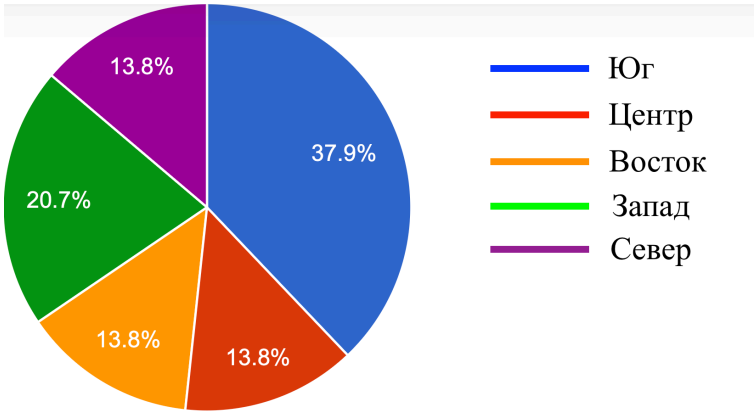


Рисунок 3.1 – Распределение по регионам Республики Казахстан

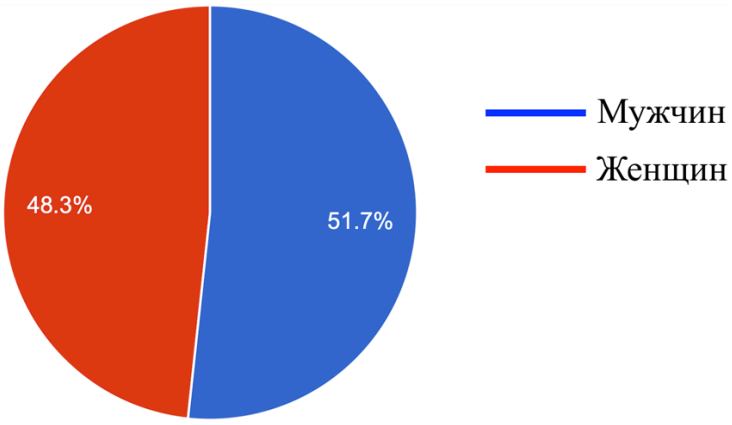


Рисунок 3.2 – Соотношение мужчин и женщин

КРК является крупнейшей открытой базой данных, созданной для развития приложений по обработке казахской речи. В рамках исследований, представленных в работах [76, 77, 78], проведенных нашими учеными, были осуществлены успешные эксперименты по распознаванию речевого сигнала. Результаты показали высокую точность, с уровнем ошибок, составляющим 2,8% на уровне символов и 8,7% на уровне слов в тестовом наборе.

Из данного речевого корпуса были выбраны записи 169 дикторов, по 75 записей на каждого диктора. Для обучения были выбраны записи первых 30 дикторов, в общей сложности 2250 аудиофайлов.

На этапе предварительной обработки аудиоданных была выполнена ручная сегментация речевых сигналов на отдельные фрагменты, соответствующие произнесенным словам, с применением программного обеспечения Audacity версии 3.7.3. Каждый аудиофайл был внимательно обработан, где сегменты с аудиосигналом были помечены как «1», а области пауз как «0». Также хотелось бы отметить, что приведенный на рисунке 3.3 пример ручной разметки аудиофайла играл важную роль в подготовке обучающих данных для использования в моделях нейронных сетей. Разметка данных играет важную роль в подготовке наборов данных для обучения нейронных сетей. Точность разметки аудиофайлов не только улучшает определение местоположения речи, но и значительно повышает эффективность обучения моделей в задачах распознавания и синтеза речевого сигнала. Такой подход способствует повышению качества обучения моделей и отражается на качестве их работы в дальнейшем. Правильная разметка данных является основой для создания надежных и точных моделей распознавания речевого сигнала, способных обрабатывать различные языки и акценты с высокой степенью точности.

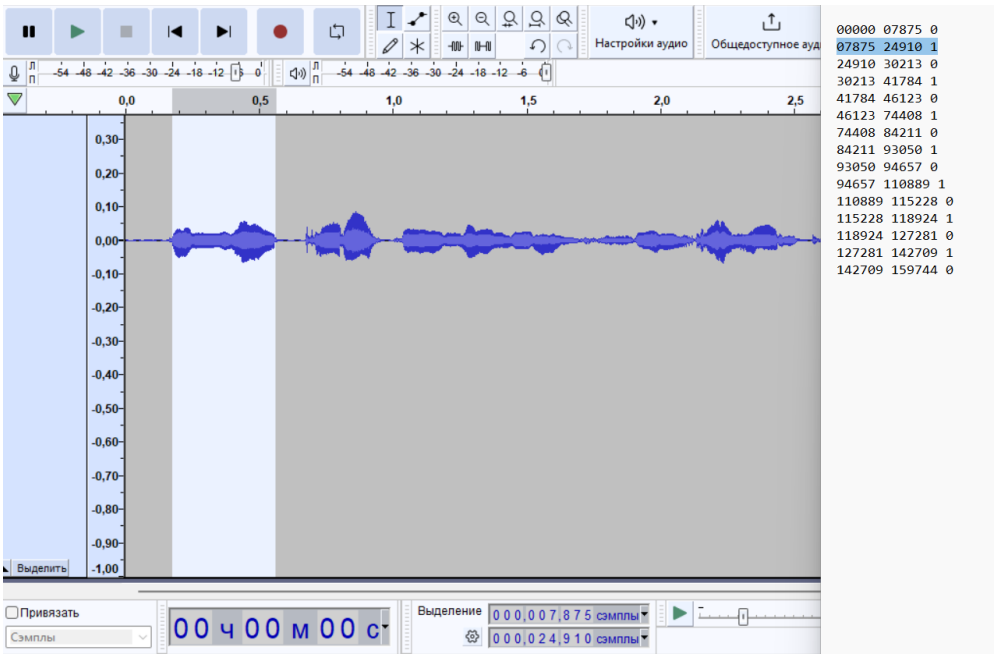


Рисунок 3.3 – Ручная разметка аудиоданных на блоки

Каждый сегмент сохранялся как отдельный аудиофайл, формируя датасет. Все 2250 аудиозаписей были рандомно распределены на 13 подмножеств, каждое из которых были размещены в отдельные папки. К аудиозаписям, содержащимся в каждой из 13 папок, был добавлен белый шум с уровнями от -18 дБ до 18 дБ с шагом 3 дБ, что позволило смоделировать различные условия акустических помех.

Методы извлечения признаков и алгоритмы распознавания являются основными компонентами системы автоматического распознавания речевого сигнала. Извлечение признаков представляет собой процесс, который фокусируется на получении небольшого объема данных, значимых для решения поставленной задачи [5].

Наиболее распространенным и доминантным методом, используемым для выделения спектральных признаков, является вычисление мел-кепстральных коэффициентов MFCC (Mel Frequency Cepstral Coefficients). MFCC - один из самых популярных методов выделения признаков, используемых при распознавании речевого сигнала на основе частотной области с использованием шкалы Mel, которая основана на шкале человеческого слуха. MFCC, рассматриваемые как характеристики в частотной области, гораздо более точны, чем характеристики во временной области [79]. Низкочастотные кепстральные коэффициенты получаются через быстрое преобразование Фурье (FFT) и устойчивы к изменениям дикторов и условий записи [80].

Метод MFCC использует линейные и логарифмические фильтры для извлечения основных речевых характеристик. Шкала Mel основана на восприятии высоты звука: частоты до 1000 Гц отображаются линейно, а выше 1000 Гц – логарифмически. Данная зависимость не является строго линейной и описывается следующей формулой:

$$Mel = 2595 \log_{10}\left(1 + \frac{f}{700}\right), \tag{3.1}$$

где $mel(f)$ – шкала частот mel, f – линейная частота.

Следовательно, зависимость между линейной частотой и шкалой Mel, лежащая в основе расчета MFCC-признаков, определяется согласно формуле (3.1). Данная формула отражает психоакустическое восприятие частоты человеком, обеспечивая переход к шкале, приближенной к восприятию слуха.

Как показано на рисунке 3.4 MFCC состоит из нескольких вычислительных этапов. Каждый этап имеет свою собственную функцию и математические подходы, которые кратко обсуждаются ниже.

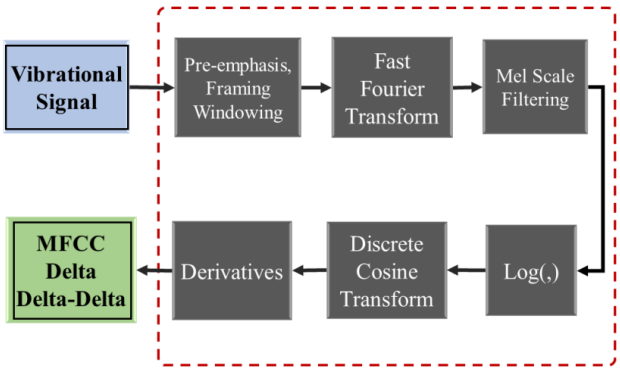


Рисунок 3.4 – Этапы расчета MFCC [81, с. 5]

Этап 1: Предварительное выделение (Pre–emphasis).

На данном этапе сигнал фильтруется таким образом, чтобы выделить высокие частоты в спектре. Это достигается за счет сравнения амплитуд высоких и низких частот, что улучшает общий уровень отношения СШ. В результате усиливается энергия сигнала на высоких частотах [82].

$$y(n) = x(n) - (x(n-1) \times \delta), \tag{3.2}$$

где $x(n)$ – значение входного сигнала перед процессом предварительного выделения, $y(n)$ – результаты предварительного выделения выходного сигнала, δ – коэффициент фильтрации, по умолчанию значение равно 0,9 до 1 [83].

Сигнал до и после предварительного выделения показан на рисунке 3.5, где видим, что огибающая исходного сигнала, представляющая низкую частоту, была удалена.

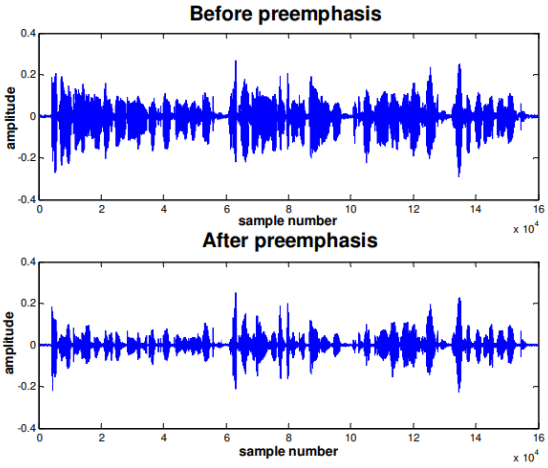


Рисунок 3.5 – До и после предварительного выделения [84, с. 45]

Этап 2: Создание фрейма (Framing).

На следующем этапе осуществляется разбиение сигнала на фреймы (рисунок 3.6). Длительность каждого кадра обычно находится в пределах от 20

до 40 мс, так как в этом интервале времени речевой сигнал остается устойчивым [85]. Речевой сигнал представляется следующим образом:

$$x(n), 0 \leq n < N, \tag{3.3}$$

где N размер фрейма или длина окна, $x_j(n)$ – j-ый фрейм.

Процесс продолжается до полного распределения всех звуковых сигналов, включая речевые, по фреймам. Обычно перекрытие между фреймами составляет от 30% до 50% от их длины. Далее для каждого фрейма выполняются последующие операции.

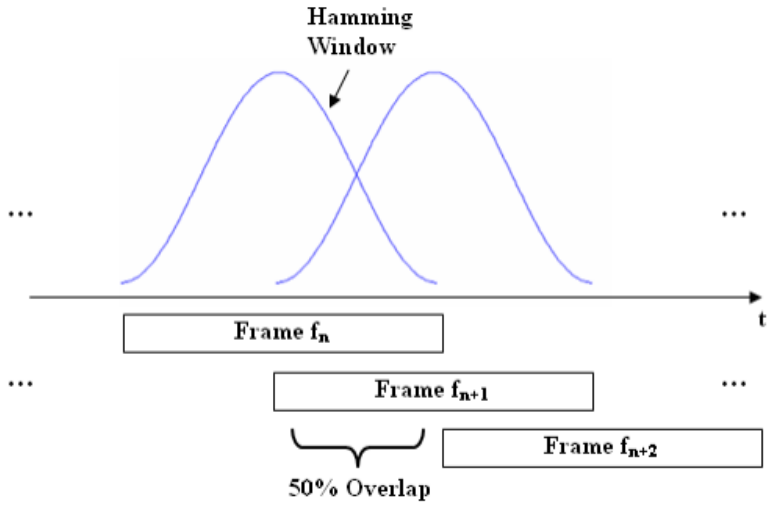


Рисунок 3.6 – Перекрытие кадров в алгоритме извлечения MFCC [86, с. 11].

Этап 3: Управление окнами Хэмминга (Hamming windowing).

Каждый отдельный кадр управляется таким образом, чтобы минимизировать разрывы сигнала в начале и конце каждого кадра. В качестве окна используется окно Хэмминга, которое задается следующим образом. Если окно определено как:

$$W\ (n), \quad 0\ \leq\ n\ \leq\ N-1, \tag{3.4}$$

где N – количество отсчетов в каждом кадре, число дискретных значений, составляющих один фрагмент сигнала после его разбиения. Это значение зависит от частоты дискретизации и длины кадра и может варьироваться от 256 до 2048 отсчетов, в зависимости от задачи и требований обработки. $Y(n)$ – выходной сигнал, это результат обработки данных, представленный в виде цифрового сигнала, предназначенный для дальнейшего использования или анализа. $X(n)$ – входной сигнал, исходные данные, поступающие в систему для последующей обработки или анализа. $W(n)$ – Окно Хэмминга, тогда результат отображения сигнала в окне показан ниже:

$$Y(n) = X(n) \cdot W(n);$$

$$W(n) = 0{,}54 - 0{,}46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1, \tag{3.5}$$

Этап 4: Быстрое преобразование Фурье.

На данном этапе каждый кадр будет переведен из временной области в частотную. Этот процесс позволит получить частотный спектр, который используется для упрощения вычислений и анализа [87].

$$X_j(k) = \sum_{n=0}^{N-1} x_j(n) W(n) e^{\frac{-j2\pi km}{N_m}}, \quad 0 \le k < N, \tag{3.6}$$

где j – номер фрейма.

Быстрое преобразование Фурье позволяет преобразовать сигнал из временной области в частотную, что дает возможность получить спектральное представление аудиосигнала, как это описано в выражении (3.6). Именно данная операция лежит в основе последующего анализа спектральных характеристик речевого сигнала.

Затем вычисляем периодограмму для каждого фрейма:

$$P_j(k) = \frac{\left|X_j(k)\right|^2}{N}, \tag{3.7}$$

Этап 5: Вычисление блок мел-фильтров.

Диапазон частот, получаемый при быстром преобразовании Фурье (БПФ), слишком широк и не соответствует линейной шкале речевого сигнала [88, 89, 90]. Для корректировки этого используются от 20 до 40 треугольных фильтров, применяемых к периодограмме и суммируемых для получения энергии каждого фильтра (рисунок 3.7). Форму треугольного фильтра описывает следующая функция:

$$H_m(k) = \begin{cases} 0, & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)}, & f(m-1) \le k < f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)}, & f(m) \le k \le f(m+1) \\ 0, & k > f(m+1) \end{cases}, \tag{3.8}$$

где m – это число фильтров, которое мы хотим получить.

Для получения энергии каждого фильтра используется треугольная амплитудно-частотная характеристика, математически описанная функцией $H_m(k)$ в формуле (3.8). Данная функция определяет форму фильтра, максимизируя чувствительность к соответствующему частотному диапазону.

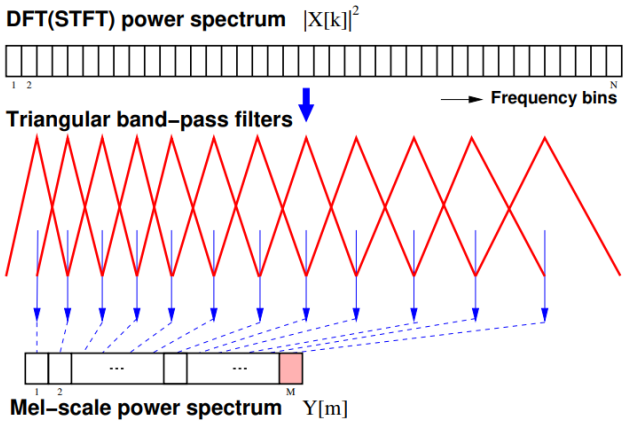


Рисунок 3.7 – Фильтры собираются в области низких частот, обеспечивая более высокое «разрешение» там, где это необходимо для распознавания [91, с. 26]

На рисунке представлен набор треугольных фильтров, которые применяются для вычисления взвешенной суммы спектральных компонентов, чтобы приблизить результат к шкале Mel. Каждый фильтр имеет амплитудно-частотную характеристику треугольной формы: она достигает единицы на центральной частоте и линейно спадает до нуля на центральных частотах соседних фильтров. Выход фильтра – это сумма отфильтрованных спектральных компонентов.

Так как человеческий слух не воспринимает громкость линейно, рассчитанные энергии логарифмируются. Чтобы удвоить воспринимаемую громкость, требуется в 8 раз больше энергии. Это означает, что значительное увеличение энергии не всегда приводит к сильному изменению восприятия звука, если он изначально был громким. Логарифмическое сжатие помогает приблизить звуковые характеристики к тому, как их воспринимает человек.

$$S_j(m) = \ln \sum_{k=0}^{N-1} P_j(k) H_m(k), \quad 0 \le m < M\,, \tag{3.9}$$

Для приведения распределения энергии к восприятию человеческого слуха применяется логарифмическое сжатие, расчет которого осуществляется согласно выражению (3.9). Это позволяет адаптировать характеристики сигнала к особенностям слухового восприятия.

Этап 6: Применение дискретного косинусного преобразования.

На этом этапе логарифмический спектр Mel преобразуется в временную область с использованием метода дискретного косинусного преобразования (DCT). Результат преобразования называется мел-кепстральными коэффициентами (MFCC):

$$C_j(n) = \sum_{m=0}^{M-1} S_j(m) \cos\left(\frac{\pi n\left(m+\frac{1}{2}\right)}{M}\right), \quad 0 \le n < M\,, \qquad (3.10)$$

где $C_j(n)$ мел-кепстральные коэффициенты.

Заключительным этапом является применение дискретного косинусного преобразования, результатом которого становятся MFCC коэффициенты, рассчитываемые по формуле (3.10). Данные коэффициенты отражают наиболее информативные параметры спектра и используются в дальнейшем в качестве признаков для классификации речевых и неречевых фрагментов.

3.2 Разработка нейросетевых архитектур с анализом их эффективности в задаче детектирования речевого сигнала

На этапе предварительной обработки аудиоданных была выполнена сегментация речевых сигналов на отдельные фрагменты, соответствующие произнесенным словам. Для этого использовались данные из файлов разметки формата .wrd, содержащих временные границы слов в аудиозаписях. Данные метки содержат начало и конец звучания каждого слова в отсчетах. На основе этих данных исходные аудиозаписи автоматически разрезаются на отдельные звуковые фрагменты, соответствующие словам. Каждый сегмент сохраняется как отдельный аудиофайл, формируя датасет, пригодный для анализа речи на уровне отдельных слов. На рисунке 3.8 показано количество аудиоданных с уровнями шума от -18 дБ до 18 дБ с шагом 3 дБ. Количество слов в каждом случае распределено почти равномерно.

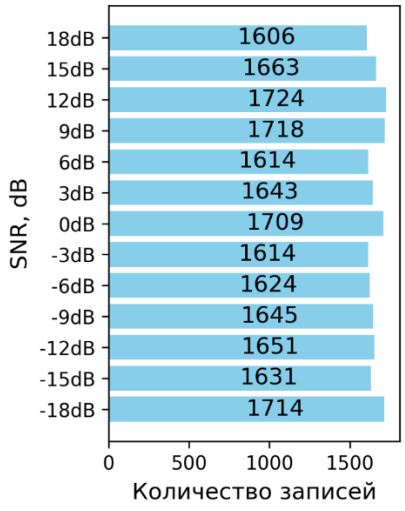


Рисунок 3.8 – Количество записей для каждого уровня данных (дБ)

Для подготовки обучающих данных был реализован программный модуль на языке Python с использованием библиотеки Librosa, предназначенный для извлечения признаков из аудиосигналов. Аудиофайлы были распределены по папкам, каждая из которых соответствовала определенному уровню шума в диапазоне от -18 дБ до 18 дБ с шагом 3 дБ.

Из каждого аудиофайла извлекались коэффициенты MFCC, являющиеся важными признаками для задач автоматической обработки речи. Всего вычислялось 25 коэффициентов, однако в дальнейшем использовались лишь 24, так как первый коэффициент, характеризующий энергию сигнала, был исключен из рассмотрения. Вычисление MFCC осуществлялось с использованием окна анализа длительностью 20 мс (1сек – 16000, 20мс – 320, то есть размер длины окна БПФ) и шагом 5 мс (5мс – 80 это hop_length (количество сэмплов между последовательными кадрами), что позволило отразить временно-частотные характеристики сигнала с высокой детализацией.

Для каждого аудиофайла формировалась матрица MFCC-признаков, и, при условии достаточной длины сигнала, по ней с применением скользящего окна формировались фрагменты фиксированной длины, содержащие по 24 временных шага. Каждый такой фрагмент представлял собой трехмерный массив размерности (24, 24), где одна ось соответствовала коэффициентам MFCC, а другая – временным интервалам. Полученные фрагменты объединялись в единый массив признаков, который впоследствии сохранялся в отдельный .npy файл для каждого уровня шума.

Таким образом, для всех уровней шумовых искажений были сформированы обучающие выборки, каждая из которых содержит нормализованные и структурированные признаки аудиосигналов, пригодные для последующего использования в задачах классификации или распознавания речи.

На рисунке 3.9 представлена диаграмма, иллюстрирующая количество сформированных обучающих фрагментов MFCC-признаков для различных уровней шума от –18 дБ до +18 дБ с шагом 3 дБ. Как видно из диаграммы, количество обучающих примеров варьируется в зависимости от уровня шума, что связано с особенностями продолжительности аудиофайлов и количеством файлов в каждой папке.

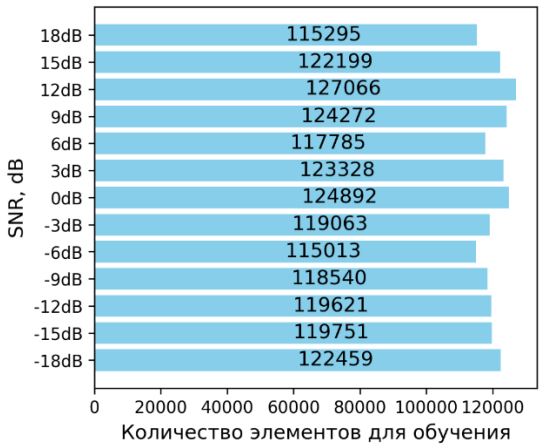


Рисунок 3.9 – Количество обучающих фрагментов MFCC для различных уровней шума

Для обучения и тестирования данных из данного набора были использованы следующие пропорции: 80% данных было случайным образом отобрано для обучения, а оставшиеся 20% – для тестирования. Затем из

полученных данных для обучения и тестирования были извлечены коэффициенты MFCC. Общее количество сформированных обучающих фрагментов составило 2 493 440, а для тестирования – 623 360. Далее в таблице 3.1 представлены основные гиперпараметры, использованные при расчете MFCC, включая частоту дискретизации, количество коэффициентов, параметры оконного анализа и временную структуру входных признаков, подаваемых на вход нейронной сети.

Таблица 3.1 – Описания гиперпараметров MFCC

№	Гиперпараметры	Значение
1	Частота дискретизации	16 000 Гц
2	Количество MFCC	25
3	Количество генерируемых полос Mel	25
4	Длина окна БПФ	320
5	Количество сэмплов между последовательными кадрами	80
6	MFCC, количество временных окон	24*24

Таким образом, подобранные значения гиперпараметров MFCC позволили получить представление речевых сигналов с высокой временной и спектральной разрешающей способностью, что обеспечило устойчивость моделей к шумовым искажениям и повысило точность классификации речевых сегментов.

Приведенный на рисунке 3.10 код демонстрирует процесс извлечения MFCC с учетом заданных параметров, обеспечивая необходимую точность и стабильность признакового представления речевых данных.

```
n_mfcc = 25
n_mels = 25
n_fft = 16*20          # Размер FFT (20 мс)
hop_length = 16*5      # Шаг окна (5 мс)

y, sr = librosa.load(file, sr=None)
mfcc = librosa.feature.mfcc(y=y,
                            sr=sr,
                            n_mfcc=n_mfcc,
                            n_mels=n_mels,
                            n_fft=n_fft,
                            hop_length=hop_length,
                            center=False)
```

Рисунок 3.10 – Программа подачи гиперпараметров для вычисления MFCC

Для обучения классификации речевых сигналов и шумов были использованы следующие архитектуры: CNN + BiGRU, CNN + GRU, CNN + BiLSTM, CNN + LSTM и CNN + TDNN.

На рисунке 3.11 представлена структура нейронной сети архитектуры CNN + BiGRU, включающая последовательность слоев с указанием их типов, выходных форм и количества обучаемых параметров. Данная модель

предназначена для решения задачи детектирования речевого сигнала, где сверточные слои (Conv2D и MaxPooling2D) используются для извлечения локальных спектральных признаков из MFCC-фрагментов, а двунаправленные рекуррентные слои BiGRU позволяют учитывать как предыдущий, так и будущий контекст аудиосигнала, повышая точность распознавания речевых и неречевых фрагментов.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 reshape (Reshape)           (None, 24, 24, 1)         0

 conv2d (Conv2D)             (None, 24, 24, 16)        160

 max_pooling2d (MaxPooling2D  (None, 12, 12, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 12, 12, 16)        2320

 max_pooling2d_1 (MaxPooling  (None, 6, 6, 16)         0
 2D)

 reshape_1 (Reshape)         (None, 36, 16)            0

 bidirectional (Bidirectiona  (None, 36, 32)           3264
 l)

 bidirectional_1 (Bidirectio  (None, 32)               4800
 nal)

 dense (Dense)               (None, 16)                528

 dense_1 (Dense)             (None, 2)                 34

=================================================================
Total params: 11,106
Trainable params: 11,106
Non-trainable params: 0
_________________________________________________________________
```

Рисунок 3.11 – Параметры сети CNN + BiGRU

Кроме того, на рисунке 3.12 представлены графики обучения модели CNN-BiGRU: слева – изменение значения функции потерь (loss и val_loss), справа – изменение точности классификации (acc и val_acc) на обучающей и валидационной выборках соответственно. Графики демонстрируют устойчивое снижение ошибки и рост точности на протяжении всех эпох, что свидетельствует о стабильной сходимости модели и отсутствии переобучения. Уже к 6–7 эпохе модель достигает высокой точности (свыше 96%), при этом значения функции потерь стабилизируются, что подтверждает эффективность выбранной архитектуры для задачи детектирования речевого сигнала.

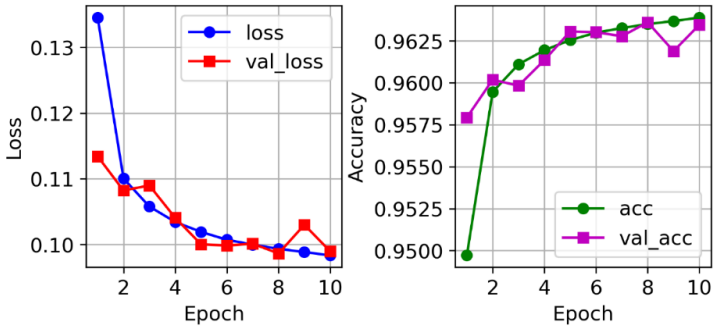


Рисунок 3.12 – Обучение модели CNN-BiGRU

На рисунке 3.13 представлены параметры архитектуры нейросетевой модели CNN + BiLSTM, используемой для детектирования речевого сигнала. Архитектура включает два сверточных слоя (Conv2D) с операциями субдискретизации (MaxPooling2D), преобразование размерности (Reshape), два двунаправленных слоя LSTM (Bidirectional), а также два полносвязных слоя (Dense). Общая обучаемая параметризация модели составляет 13 538 весов. Такая конфигурация обеспечивает извлечение пространственно-временных признаков и их последовательную обработку, что особенно важно при работе с акустическими сигналами в условиях шумов.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 reshape (Reshape)           (None, 24, 24, 1)         0

 conv2d (Conv2D)             (None, 24, 24, 16)        160

 max_pooling2d (MaxPooling2D  (None, 12, 12, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 12, 12, 16)        2320

 max_pooling2d_1 (MaxPooling  (None, 6, 6, 16)         0
 2D)

 reshape_1 (Reshape)         (None, 36, 16)            0

 bidirectional (Bidirectiona  (None, 36, 32)           4224
 l)

 bidirectional_1 (Bidirectio  (None, 32)               6272
 nal)

 dense (Dense)               (None, 16)                528

 dense_1 (Dense)             (None, 2)                 34

=================================================================
Total params: 13,538
Trainable params: 13,538
Non-trainable params: 0
_________________________________________________________________
```

Рисунок 3.13 – Параметры сети CNN + BiLSTM

На рисунке 3.14 представлены графики обучения модели CNN-BiLSTM, показывающие изменение функции потерь (Loss) и точности (Accuracy) как на обучающем наборе данных, так и на валидационном. Снижение значения ошибки и рост точности с каждой эпохой свидетельствуют о стабильной сходимости

модели и об отсутствии признаков переобучения. Это указывает на высокую эффективность архитектуры CNN-BiLSTM при решении задачи детектирования речевого сигнала в условиях зашумленной акустической среды.

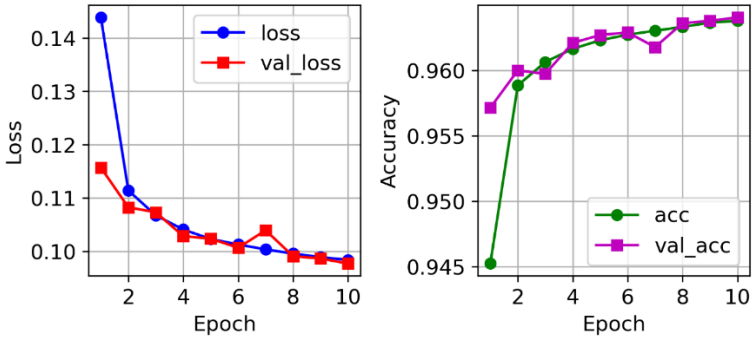


Рисунок 3.14 – Обучение модели CNN-BiLSTM

На рисунке 3.15 представлены параметры архитектуры нейронной сети CNN + GRU, реализованной в последовательной модели (sequential). Данная модель объединяет сверточные слои (Conv2D) и двухслойный GRU-модуль, что позволяет эффективно извлекать как пространственные, так и временные характеристики из входных MFCC-признаков. Всего в модели 6 050 обучаемых параметров.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 reshape (Reshape)           (None, 24, 24, 1)         0

 conv2d (Conv2D)             (None, 24, 24, 16)        160

 max_pooling2d (MaxPooling2D  (None, 12, 12, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 12, 12, 16)        2320

 max_pooling2d_1 (MaxPooling  (None, 6, 6, 16)         0
 2D)

 reshape_1 (Reshape)         (None, 36, 16)            0

 gru (GRU)                   (None, 36, 16)            1632

 gru_1 (GRU)                 (None, 16)                1632

 dense (Dense)               (None, 16)                272

 dense_1 (Dense)             (None, 2)                 34

=================================================================
Total params: 6,050
Trainable params: 6,050
Non-trainable params: 0
_________________________________________________________________
```

Рисунок 3.15 – Параметры сети CNN + GRU

На рисунке 3.16 представлена динамика обучения модели CNN-GRU, выраженная через графики функции потерь (loss) и точности (accuracy) на обучающей и валидационной выборках. Снижение значения потерь и рост

точности на протяжении всех эпох указывают на стабильную сходимость модели и отсутствие признаков переобучения. Это подтверждает эффективность выбранной архитектуры в задаче детектирования речевого сигнала.

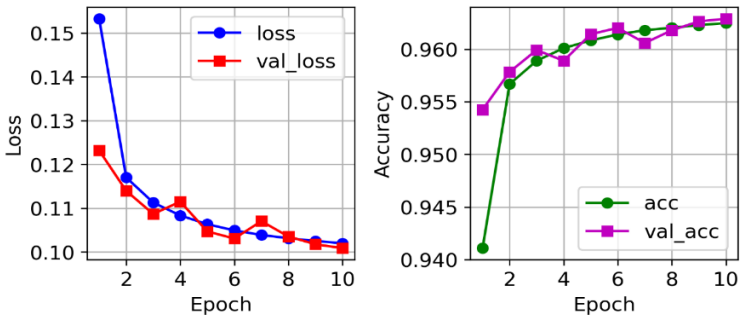


Рисунок 3.16 – Обучение модели CNN-GRU

На рисунке 3.17 представлены параметры модели нейронной сети с архитектурой CNN + LSTM, включающей в себя сверточные и пулинговые слои для извлечения пространственных признаков, а также два последовательно соединенных слоя LSTM, обеспечивающих обработку временных зависимостей. Каждый слой снабжен указанием выходной размерности тензора (Output Shape) и числом обучаемых параметров (Param #). Такая архитектура позволяет эффективно обрабатывать аудиосигналы с временной структурой.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 reshape (Reshape)           (None, 24, 24, 1)         0

 conv2d (Conv2D)             (None, 24, 24, 16)        160

 max_pooling2d (MaxPooling2D  (None, 12, 12, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 12, 12, 16)        2320

 max_pooling2d_1 (MaxPooling  (None, 6, 6, 16)         0
 2D)

 reshape_1 (Reshape)         (None, 36, 16)            0

 gru (GRU)                   (None, 36, 16)            1632

 gru_1 (GRU)                 (None, 16)                1632

 dense (Dense)               (None, 16)                272

 dense_1 (Dense)             (None, 2)                 34

=================================================================
Total params: 6,050
Trainable params: 6,050
Non-trainable params: 0
_________________________________________________________________
```

Рисунок 3.17 – Параметры сети CNN + LSTM

На рисунке 3.18 представлена динамика процесса обучения модели с архитектурой CNN + LSTM по метрикам потерь (loss) и точности (accuracy) на обучающей и валидационной выборках. Левый график отражает постепенное

снижение функции потерь как на обучении (синим), так и на валидации (красным), что свидетельствует о хорошей сходимости модели. Правый график демонстрирует устойчивый рост точности, стабилизирующийся к 10-й эпохе на уровне выше 96%, что подтверждает высокую способность модели к обобщению и ее эффективность при решении задачи детектирования речевого сигнала.

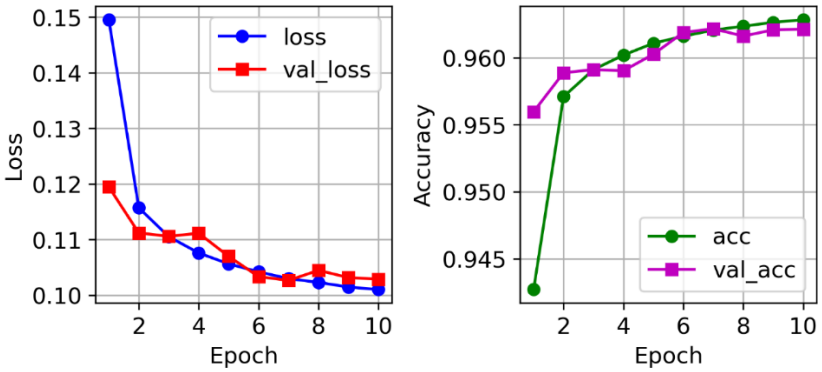


Рисунок 3.18 – Обучение модели CNN-LSTM

На рисунке 3.19 представлены параметры модели нейронной сети с архитектурой CNN + TDNN, используемой для детектирования речевого сигнала.

Model: "sequential"

Layer (type)	Output Shape	Param #
reshape (Reshape)	(None, 24, 24, 1)	0
conv2d (Conv2D)	(None, 24, 24, 16)	160
max_pooling2d (MaxPooling2D)	(None, 12, 12, 16)	0
conv2d_1 (Conv2D)	(None, 12, 12, 16)	2320
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 16)	0
reshape_1 (Reshape)	(None, 36, 16)	0
conv1d (Conv1D)	(None, 36, 16)	1296
conv1d_1 (Conv1D)	(None, 36, 16)	784
conv1d_2 (Conv1D)	(None, 36, 16)	784
global_average_pooling1d (GlobalAveragePooling1D)	(None, 16)	0
dense (Dense)	(None, 16)	272
dense_1 (Dense)	(None, 2)	34

Total params: 5,650
Trainable params: 5,650
Non-trainable params: 0

Рисунок 3.19 – Параметры сети CNN + TDNN

Сеть включает в себя начальные сверточные и подвыборочные слои (Conv2D и MaxPooling2D), переход к одноосевым сверточным слоям TDNN (Conv1D), а также слой глобального усреднения (GlobalAveragePooling1D), завершающийся двумя полносвязными слоями. Общая структура модели обеспечивает извлечение как пространственных, так и временных признаков, а общее количество обучаемых параметров составляет 5650. Такая архитектура отличается вычислительной эффективностью и подходит для задач классификации в условиях ограниченных ресурсов.

На рисунке 3.20 представлена динамика обучения модели с архитектурой CNN + TDNN, отображающая изменение значений функции потерь (loss и val_loss) и точности классификации (accuracy и val_accuracy) на обучающей и валидационной выборках в течение 10 эпох. Видно, что модель демонстрирует уверенное снижение ошибки и рост точности, достигая стабильных значений без признаков переобучения, что указывает на хорошую обобщающую способность данной архитектуры в задаче детектирования речевого сигнала.

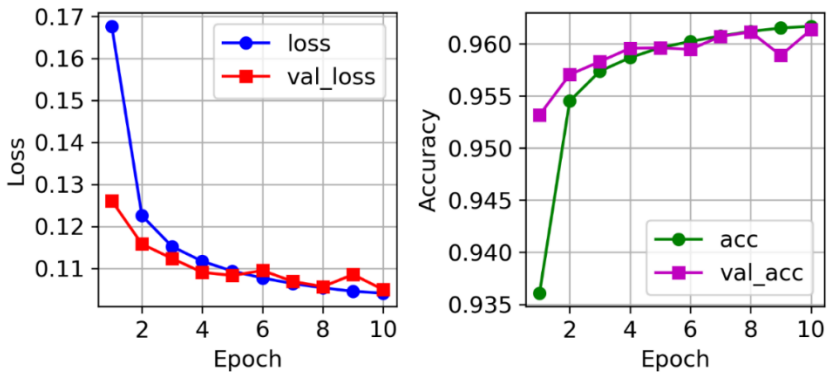


Рисунок 3.20 – Обучение модели CNN-TDNN

Сравнительный анализ временных затрат на обучение различных архитектур нейронных сетей представлен на рисунке 3.21 Для каждой модели были зафиксированы полные значения времени, затраченного на прохождение всех эпох обучения при идентичных условиях эксперимента, включая размер обучающей выборки, параметры оптимизации и вычислительную среду.

Как видно из графика, наибольшее время обучения продемонстрировали модели CNN+BiLSTM (558 секунд) и CNN+BiGRU (525 секунд), что объясняется высокой вычислительной сложностью рекуррентных компонентов с двунаправленной структурой. Модели CNN+GRU и CNN+LSTM показали более умеренные значения – 300 и 318 секунд соответственно. Наиболее быстрой оказалась модель CNN+TDNN, с временем обучения всего 164 секунды, что указывает на ее высокую вычислительную эффективность и потенциальную пригодность для задач, требующих минимальных затрат времени на обучение.

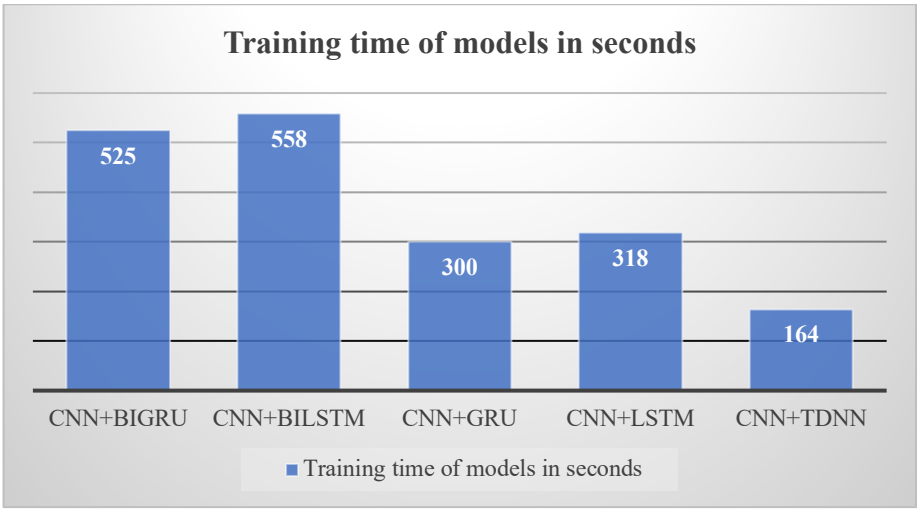


Рисунок 3.21 – Время обучения моделей различных архитектур нейронных сетей

Таким образом, были разработаны архитектуры нейросетевых моделей, основанные на комбинировании сверточных слоев (CNN) с BiGRU, GRU, BiLSTM, LSTM и TDNN для задачи детектирования голосовой активности. Проведен анализ структуры моделей, включающий описание всех входных и выходных параметров, формы тензоров, а также количества обучаемых весов на каждом слое. Архитектуры моделей получились компактными и вычислительно эффективными.

Результаты обучения моделей на экспериментальной выборке с различными уровнями шума показали стабильную сходимость и отсутствие признаков переобучения. На протяжении всех эпох наблюдалось устойчивое снижение функции потерь на обучающей и валидационной выборках, а также рост точности классификации, которая стабилизировалась на уровне выше 96% уже к 10-й эпохе.

3.3 Влияние параметров на ошибку тестирования

После определения структуры нейронной сети возникла необходимость рассмотреть другой аспект исследования, а именно влияние ряда параметров на ошибку распознавания. К основным параметрам, подлежащим анализу, относятся:

- тип нейронной сети (n);
- отношение С/Ш (SNR);
- количества дикторов (d);
- языки обучения и тестирования (l);
- пороговое значения на выходе нейронной сети (th).

Чтобы проверить влияния данных параметров, необходимо рассмотреть это как функцию зависящею от несколько параметров.

$$Err = f(n, SNR, d, l, th, et), \tag{3.11}$$

где: n – нейронные сети типа CNN и RNN; SNR – изменяется от 3дБ до 21 дБ с шагом 3; d – изменяться от 2 до 40 дикторов с шагом 2; l – языки английский (en), французский(fr), киргизский(kr), казахский(kz), русский(ru), турецкий(tu), узбекский(uz); th - пороговое значение менялось от 0,05 до 0,95 с шагом 0,05; et – тип ошибки FN и FP.

Чтобы проверить влияния каждого параметра, сперва было рассмотрено изменение одного параметра, а остальные параметры были константами. Далее рассмотрим изменения параметров на примере нейронной сети типа *RNN* и типа ошибки FP. Будут рассмотрены следующие состояния:

- изменения d, при этом $SNR = 21,\ l = kz, th = 0{,}9$.
- изменения SNR, при этом $d = 40,\ l = kz, th = 0{,}5$.
- изменения th, при этом $SNR = 21, d = 40,\ l = kz$.

Сначала методом аппроксимации находим подходящий тип функции, который описывает изменения. Для этого проведем аппроксимацию данных с использованием 4 различных функций и рассчитаем среднюю абсолютную процентную ошибку (MAPE).

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \times 100\,, \tag{3.12}$$

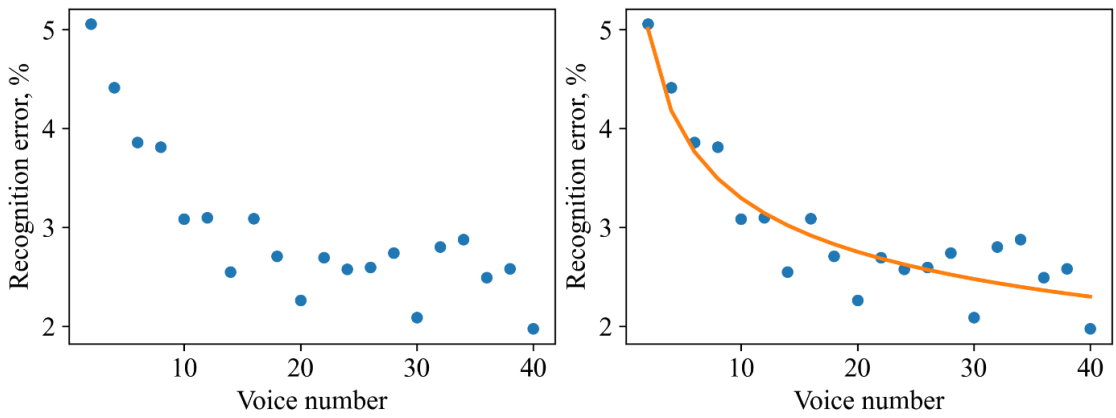
где, A_t – фактическое значение, F_t – прогнозируемое значение, n – количество наблюдений.

На основе полученных значений можно определить, к какому типу функции относятся изменения ошибки относительно количества дикторов. В таблице 3.2 представлены результаты ошибки аппроксимации, которые помогут выбрать соответствующий тип функции.

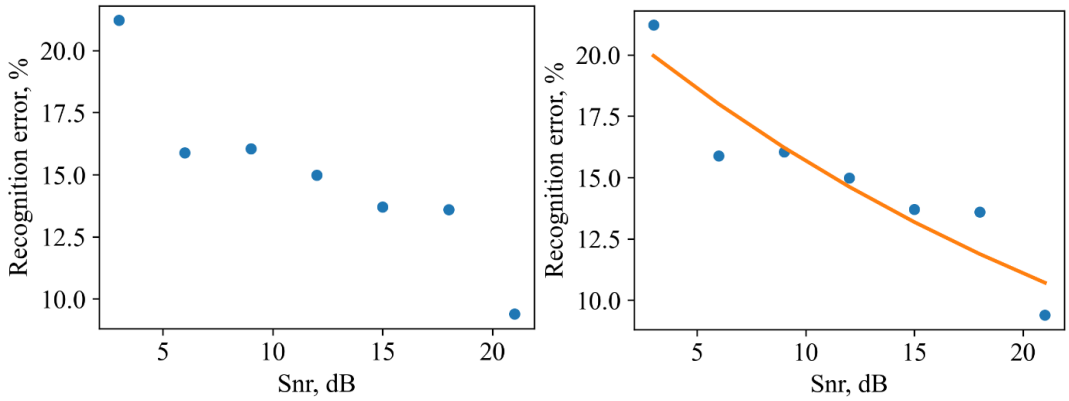
Таблица 3.2 – Расчетные показатели ошибки аппроксимации, полученные по формуле (1) для нескольких типов функций.

Variable	Function			
	$C \cdot x^n$	$C \cdot n^x$	$C \cdot e^{n \cdot x}$	$a + b \cdot \ln(x)$
d	8,33 %	11,93 %	11,99 %	9,13%
snr	8,96 %	7,57 %	7,60 %	8,16 %
th	38,76 %	16,54 %	16,55 %	11,43 %

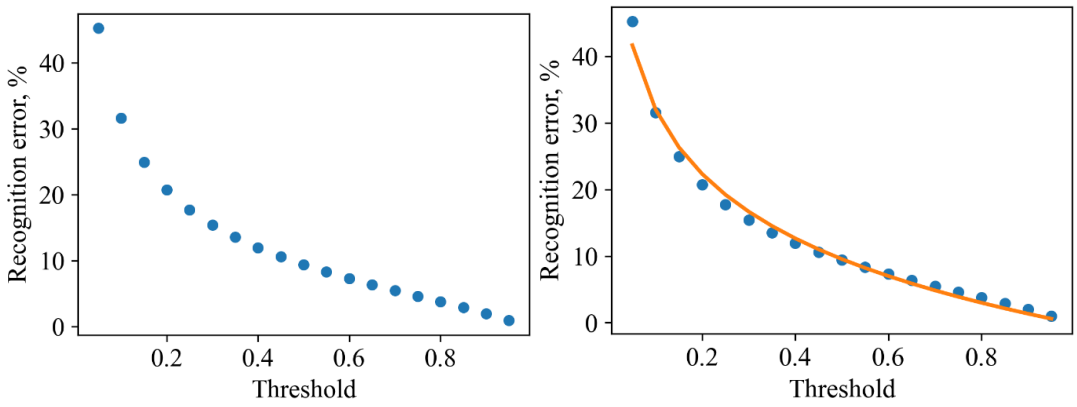
На основании данных из таблицы 3.2 были выбраны соответствующие типы аппроксимационных функций. Данные функции использовались для построения графиков, отображающих изменения ошибки (Рисунок 3.22). Графики на рисунке 3.22 показывают, как ошибка варьируется в зависимости от выбранного типа аппроксимационной функции, что позволяет визуально оценить точность каждой из функций. Таким образом, анализ этих графиков помогает определить, какая из аппроксимационных функций наиболее точно описывает зависимости в данных.



а) изменения количества дикторов d



б) изменения значения отношения С/Ш snr



в) изменения значения порогового значения th

Рисунок 3.22 – Изменение ошибки распознавания в зависимости от нескольких параметров и соответствующих типов функций

Следующим этапом является расчет скорости изменения каждой из аппроксимационных функций, что позволит нам сравнить полученные результаты. Этот анализ поможет более явно определить влияние каждого параметра на ошибку распознавания. В таблице 3.3 представлены аппроксимационные функции зависимостей для каждого изученного параметра, а также их производные.

Изучение производных функций даст нам возможность увидеть, как быстро изменяется ошибка в зависимости от каждого параметра, что, в свою очередь, позволит более точно оценить значимость этих параметров в процессе распознавания.

Таблица 3.3 – Виды аппроксимационных функции для соответствующих параметров

Variable	The approximation function	Function derivative
d	$y = 5{,}996 \cdot x^{-0{,}26}$	$\dot{y} = -1{,}559 \cdot x^{-1{,}26}$-
snr	$y = 22{,}145 \cdot 0{,}966^{x}$	$\dot{y} = -0{,}766 \cdot 0{,}966^{x}$
th	$y = -0{,}138 - 13{,}953 \cdot \ln(x)$	$\dot{y} = -\frac{13{,}953}{x}$

На основе полученных результатов расчета скорости изменения функций был построен нормированный график, показывающий сравнение влияния каждого параметра на ошибку распознавания (Рисунок 3.23).

Данный график позволяет визуально оценить, насколько сильно каждый параметр влияет на ошибку. Нормирование значений обеспечивает более наглядное представление и позволяет легко сопоставить различные параметры друг с другом.

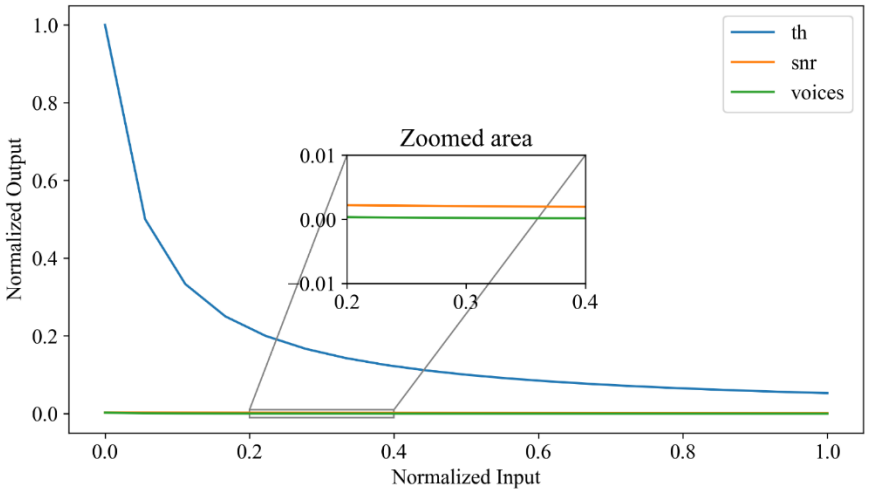


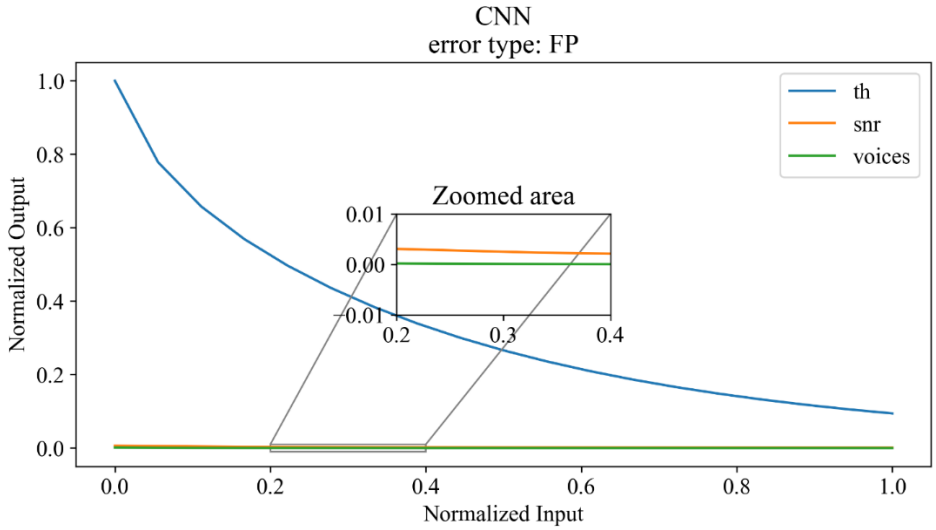
Рисунок 3.23 – Нормированный график сравнения влияния параметров на ошибку

Как видно из графика, на ошибку распознавания значительное влияние оказывает пороговое значение на выходе нейронной сети. В то же время изменения отношения С/Ш и количества дикторов оказывают почти одинаковый эффект на скорость изменения ошибки.

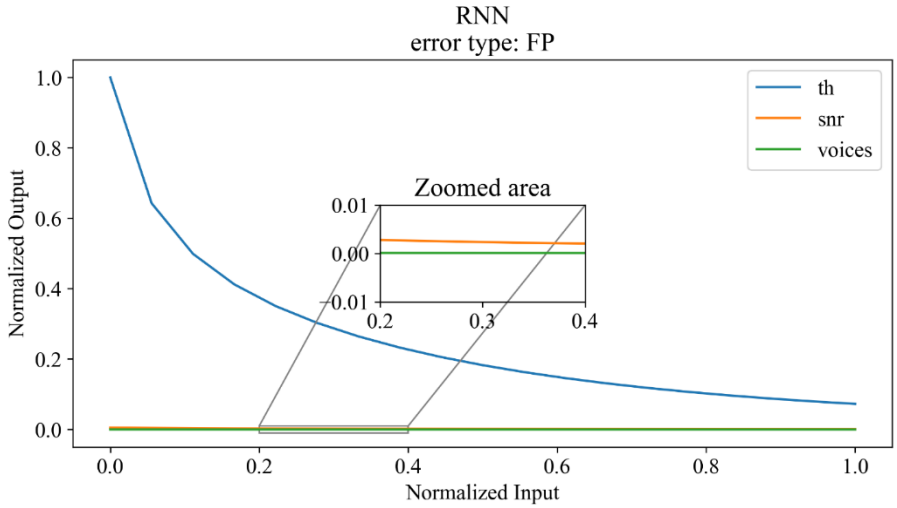
Данные результаты представляют собой лишь один из вариантов влияния параметров на ошибку. Чтобы получить более полное представление, в рамках данного исследования были проведены аналогичные расчеты для всех возможных комбинаций параметров получили обобщенные результаты влияния

параметров на скорость изменения ошибки. Результаты были усреднены с использованием метода среднего арифметического для того, чтобы отразить общее изменение. Такой подход позволяет получить более точное представление о тенденциях в данных. Усредненные значения дают возможность выявить общие закономерности и более явно показывает влияние параметров на скорость изменения ошибки.

В результате проведенного анализа, на рисунке 3.24 представлены результаты скорости изменения ошибки False Positive, а на рисунке 3.25 – скорость изменения ошибки False Negative.

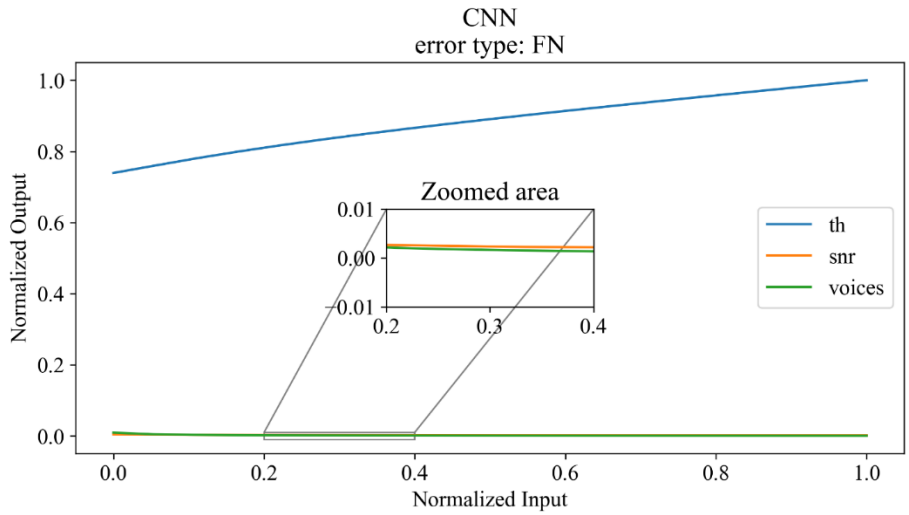


а) при нейронной сети CNN

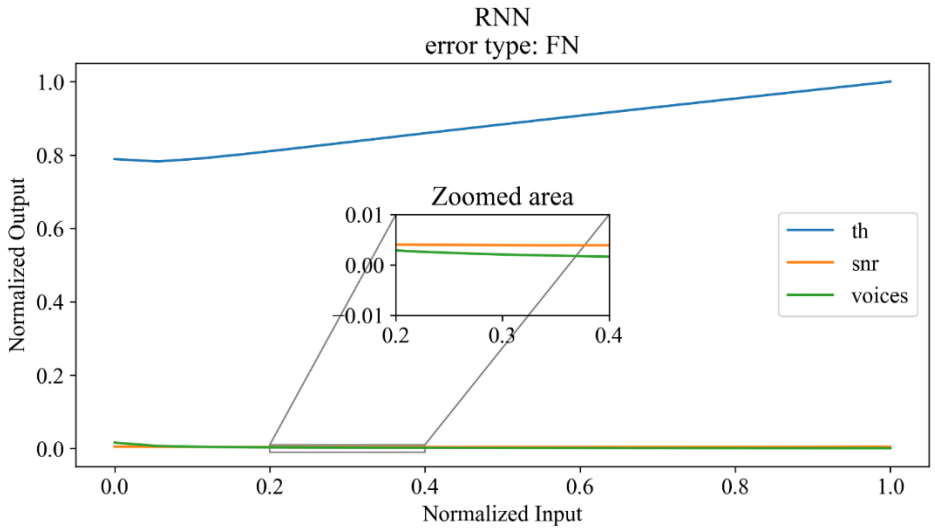


б) при нейронной сети RNN

Рисунок 3.24 – Результаты влияния параметров на скорость изменения ошибки False Positive



а) при нейронной сети CNN



б) при нейронной сети RNN

Рисунок 3.25 – Результаты влияния параметров на скорость изменения ошибки False Negative

Каждая линия из графиков является результатом усреднении большого количества данных. Количество использованных данных для усреднения изменения каждого параметра является следующими:

- при изменения порогового значения (*th*) – 980 (7 уровней отношения С/Ш, 7 разных языков и 20 вариантов количества дикторов);
- при изменении отношения С/Ш (*SNR*) – 2660 (19 вариантов порогового значения, 7 разных языков и 20 вариантов количества дикторов);
- при изменении количества дикторов (*voices*) – 931 (19 вариантов порогового значения, 7 разных языков и 7 уровней отношения С/Ш).

Как показывают результаты, изменения порогового значения оказывают значительное влияние на скорость роста ошибки, независимо от типа ошибки и используемой нейронной сети. В то время как отношения С/Ш и количество дикторов оказывают более слабое влияние на рост ошибок. Можно отметить, что

отношение С/Ш немного сильнее влияет на ошибки, чем изменения количества дикторов. Однако при сравнении с изменением порогового значения становится очевидным, что как отношение С/Ш, так и количество дикторов практически не влияют на увеличение ошибок.

Исходя из полученных результатов, можно утверждать, что изменения порогового значения существенно влияют на скорость изменения ошибок обоих типов. Это позволяет контролировать уровень ошибок, подстраивая пороговое значение в зависимости от конкретной задачи распознавания, что может способствовать уменьшению необходимого типа ошибки. Таким образом, настройка порогового значения становится важным инструментом для оптимизации производительности модели в различных сценариях.

Как показали результаты исследования, для того чтобы создать нейронную сеть, применимую для обоих классов задач по обработке речевых сигналов, задачу классификации необходимо перевести в задачу регрессии. Таким образом, получаем очень действенный инструмент в виде возможности оперативного изменения порога принятия решения на выходе единственного нейрона. Порог принятия решения может меняться от 0 до 1. При более высоких значениях порога принятия решения ошибка распознавания речи как не речь также растет, а вот ошибка распознавания не речевого участка как речевой, наоборот падает. Причем влияние выбора значения данного порога оказывает наибольшее влияние на точность работы нейронной сети. Таким образом, во время обработки речевых сигналов, можем легко менять точность того или иного типа ошибки сети без ее переобучения, только лишь изменив один параметр. Надо отметить, что минимум суммарной ошибки приходится на значение порога 0,5.

Вторым значимым фактором оказался язык данных обучающей выборки и язык данных тестирования нейронной сети. Если оба этих языка совпадают, то ошибки будут самыми минимальными. А если нейронная сеть обучалась на данных на одном языке, а тестирование проводить на данных на другом языке, то ошибки распознавания будут расти по мере уменьшения фонетического сходства между языками. Что интересно, фонетически наиболее близким к казахскому языку оказался русский язык, хотя эти языки относятся к разным семействам языков. Известно, что турецкий и казахские языки относятся к тюркской группе языков. Но несмотря на то, что эти два языка относятся к одной и той же группе языков, наша нейронная сеть вычисляет так, что эти языки оказываются наиболее далекими друг от друга. Это может означать только одно, что несмотря на наличие большого количества общих и схожих слов, казахский и турецкий языки фонетически звучат заметно разным образом. Таким образом, такой подход, т.е. когда нейронная сеть обучается на одном языке, а распознавание проводится на данных на другом языке, можно использовать при исследовании фонетического сходства между разными языками.

Третьим значимым фактором, влияющим на точность распознавания речевых участков в звуковом сигнала, оказалось соотношение С/Ш. Нейронная сеть, обученная без добавления шума или со слабым шумом, достаточно чутко реагирует на соотношение С/Ш в данных, используемых при тестировании сети. Возможно, что если нейронная сеть будет обучаться на данных с добавлением

разного уровня шума, то влияние данного фактора будет минимальным при тестировании.

Четвертым по значимости фактором оказался количество дикторов, используемых при обучении нейронной сети. С ростом количества дикторов, растет разнообразность голосов и соответственно растет вероятность правильного распознавания голосов других людей.

Выводы по главе:

1. Выявлено, что для эффективного детектирования речевого сигнала целесообразно использовать метод MFCC, обеспечивающий точное представление спектральных характеристик с учетом особенностей слухового восприятия человека. В качестве базы для обучения были использованы тщательно размеченные данные казахского речевого корпуса, дополненные искусственным шумом с различными уровнями отношения С/Ш, что позволило сформировать устойчивые выборки для обучения. Применение поэтапной обработки аудиосигнала, включающей предварительное выделение, разбиение на фреймы, оконное сглаживание, БПФ, применение мел-фильтров, логарифмическое сжатие и дискретное косинусное преобразование, позволило получить компактные и информативные признаки. Установленные этапы извлечения MFCC-признаков доказали свою пригодность для повышения точности классификации речевых и неречевых фрагментов в условиях зашумленности.

2. Установлено, что использование гибридных нейросетевых архитектур на основе сверточных и рекуррентных слоев (CNN+BiGRU, CNN+GRU, CNN+BiLSTM, CNN+LSTM, CNN+TDNN) обеспечивает высокую точность детектирования речевого сигнала. Модели продемонстрировали устойчивую сходимость, стабильный рост точности и отсутствие переобучения. Особенно эффективно проявили себя BiGRU и BiLSTM за счет учета двустороннего временного контекста, а TDNN показала минимальное время обучения, что делает ее подходящей для систем с ограниченными ресурсами. Наивысший результат F1-score был получен моделью CNN+BiGRU, что подтверждает ее преимущество среди всех протестированных. Установлено, что грамотный выбор архитектуры обеспечивает баланс между точностью, устойчивостью к шуму и эффективностью.

3. Обнаружено, что на точность распознавания речевого сигнала нейронной сетью значительное влияние оказывает выбор порогового значения на выходе модели: изменение данного параметра позволяет управлять ошибками первого и второго рода без переобучения сети. Совпадение языка обучающей и тестовой выборок существенно снижает ошибку, а различия между языками приводят к ее увеличению в зависимости от степени их фонетического сходства. Влияние отношения С/Ш выражается в росте ошибки при ухудшении акустических условий, однако обучение на зашумленных данных может повысить устойчивость сети. Доказано, что увеличение числа дикторов в обучающем наборе улучшает обобщающую способность модели.

4 ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ

4.1 Экспериментальный анализ чувствительности нейросетевых моделей к уровню шумов при обучении на фиксированных значениях отношения С/Ш

В рамках данной диссертационной работы была проведена серия экспериментов, направленных на оценку устойчивости различных нейросетевых архитектур (CNN+BiGRU, CNN+BiLSTM, CNN+GRU, CNN+LSTM и CNN+TDNN) при обнаружении речевой активности в условиях переменной зашумленности. Эксперименты были организованы таким образом, чтобы выявить особенности поведения моделей при обучении и тестировании на фиксированных уровнях отношения С/Ш.

На первом этапе была реализована стратегия, при которой обучение и тестирование проводились на одном и том же уровне отношения С/Ш. Это позволило оценить, насколько эффективно каждая модель способна распознавать речевые фрагменты при отсутствии вариативности в шумовой обстановке. Были выбраны следующие значения отношения С/Ш: –10 дБ, 0 дБ, 10 дБ, 20 дБ и 30 дБ. Для каждого из этих уровней обучения и тестирование проводились по отдельности, то есть модель обучалась исключительно на данных с заданным уровнем шума и тестировалась на аналогичных по шуму записях.

Такая постановка эксперимента позволяет оценить не только точность классификации в рамках конкретного шумового сценария, но и выявить, насколько разные архитектуры чувствительны к уровню зашумленности, если условия обучения и применения совпадают. Результаты каждого из этих экспериментов представлены ниже и сопровождаются соответствующими графиками зависимости точности от уровня отношения С/Ш для всех рассматриваемых моделей.

На рисунке 4.1 представлены результаты работы пяти нейросетевых архитектур, обученных и протестированных при уровне шума -10 дБ. Такой уровень отношения С/Ш отражает крайне неблагоприятные условия для распознавания речи, при которых фоновой шум доминирует над речевым сигналом, создавая высокую степень акустической маскировки.

Как показывает график, все модели достигают высокой точности вблизи обучающего уровня (от –10 до –5 дБ), где точность большинства архитектур находится в диапазоне от 90% до 93%. Однако при увеличении уровня отношения С/Ш от 0 до 15 дБ наблюдается резкое снижение точности практически у всех моделей до минимума в районе 10–15 дБ. В этом диапазоне точность падает до 25–40%, что указывает на потерю устойчивости моделей при отклонении от обучающих условий.

Наименьшее значение точности наблюдается именно при средних значениях отношения С/Ш, где речевой сигнал и шум находятся в относительном балансе. Это создает наибольшую неоднозначность для моделей, обученных исключительно на зашумленных данных. Лишь при дальнейшем увеличении отношения С/Ш, начиная с 15 дБ и выше, точность постепенно

восстанавливается. Наиболее уверенный рост точности после спада демонстрирует модель CNN-LSTM, достигая почти 95% при отношении С/Ш = 30 дБ. Архитектуры CNN-GRU и CNN-BiGRU также показывают заметное восстановление, однако с более медленной динамикой.

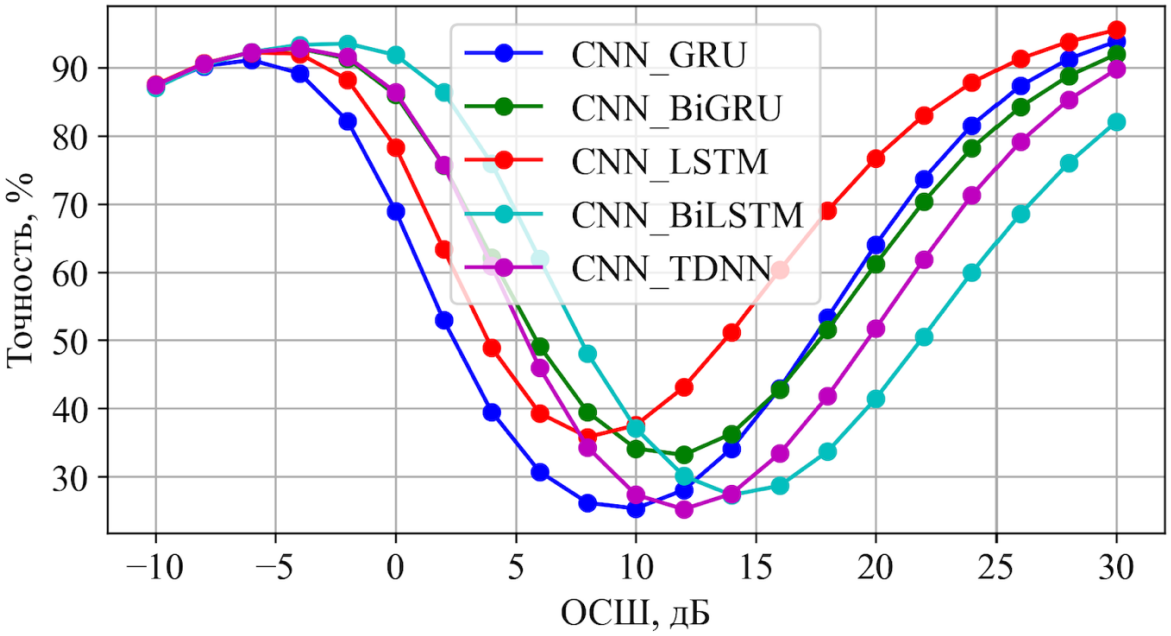


Рисунок 4.1 – Зависимость точности нейросетевых моделей от уровня отношения С/Ш при обучении на -10дБ

По результатам эксперимента можно заметить, что обучение на одном уровне сильной зашумленности приводит к переадаптации модели к шуму, снижая ее способность обрабатывать более чистые сигналы.

На рисунке 4.2 представлены результаты моделей, обученных и протестированных на данных с фиксированным уровнем отношения С/Ш, равным 0 дБ. Этот уровень шума представляет собой граничное состояние, при котором мощность речевого сигнала и шумовой компоненты являются примерно равными, что характерно для многих реальных акустических сред, таких как улица, общественный транспорт или помещения с фоновыми разговорами.

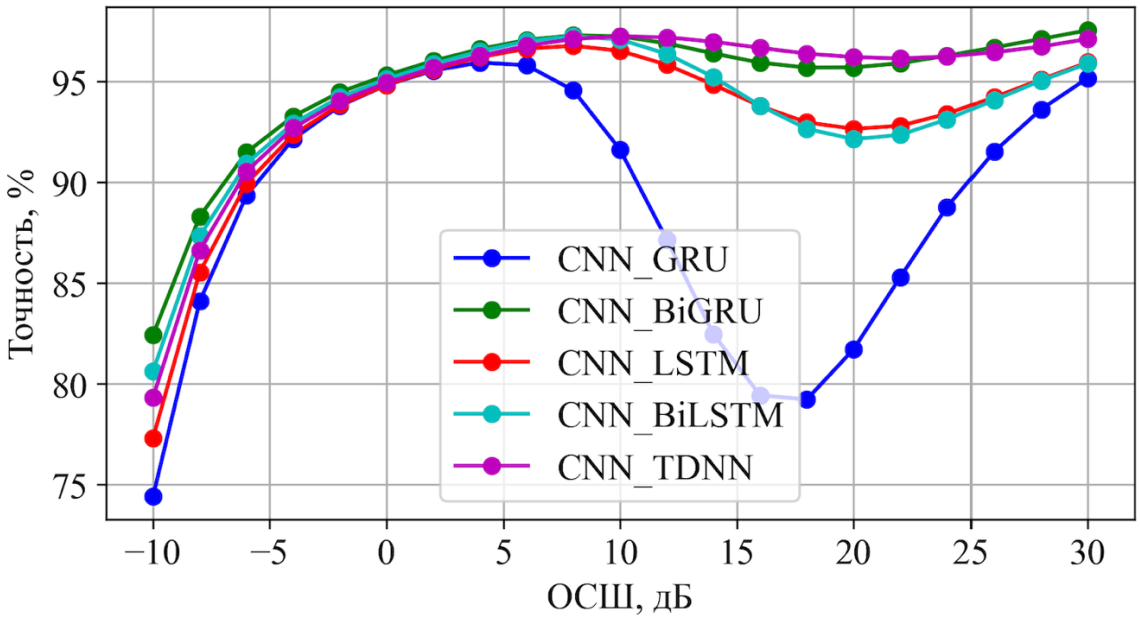


Рисунок 4.2 – Зависимость точности нейросетевых моделей от уровня отношения С/Ш при обучении на 0дБ

По полученным экспериментальным данным, все модели демонстрируют высокую точность вблизи уровня 0 дБ, что подтверждает эффективность обучения. На этом уровне почти все архитектуры достигают точности выше 94%. Однако при отклонении от обучающего уровня наблюдаются различные степени устойчивости моделей.

Наиболее заметное снижение точности фиксируется у модели CNN-GRU, начиная с 5 дБ и особенно выраженное в интервале от 10 до 20 дБ, где точность падает ниже 85% и достигает минимума в районе 78% при отношении С/Ш около 15 дБ. Это подтверждает ограниченную способность данной архитектуры к обобщению в более благоприятных акустических условиях.

С другой стороны, модели CNN-BiGRU, CNN-BiLSTM и CNN-TDNN демонстрируют более стабильное поведение. Несмотря на небольшое снижение точности в диапазоне от 10 до 20 дБ, их значения остаются выше 94–95%, что говорит о высокой адаптивности и устойчивости этих архитектур. Особенно выделяется CNN-TDNN, показывая наименьшую амплитуду колебаний точности и максимальную стабильность по всему диапазону отношения С/Ш.

Модель CNN-LSTM также демонстрирует умеренное снижение точности в средней области (10–20 дБ), но ее результаты остаются выше 93%, а точность быстро восстанавливается при переходе к более высоким значениям отношения С/Ш.

На рисунке 4.3 представлены результаты работы пяти нейросетевых архитектур, обученных и протестированных на аудиоданных с фиксированным уровнем отношения С/Ш, равным 10 дБ. Данный уровень зашумленности соответствует умеренным акустическим условиям, характерным для офисных помещений, улиц с умеренным трафиком и других сред, где речевой сигнал уже доминирует над шумом, но все еще может подвергаться искажениям.

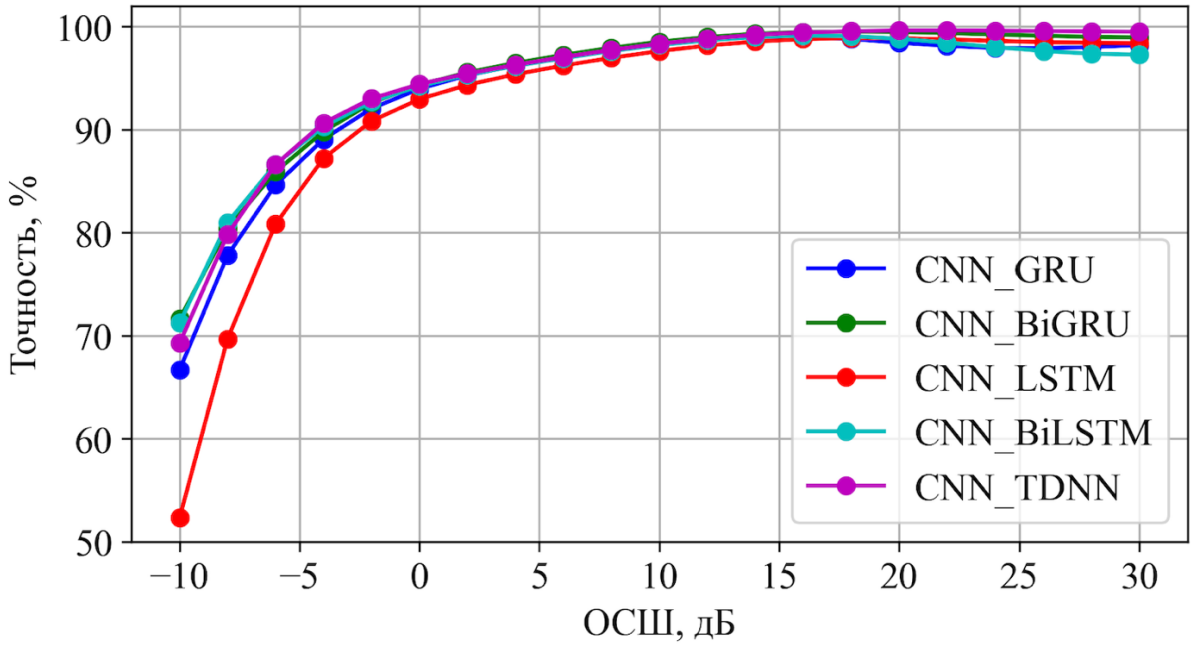


Рисунок 4.3 – Зависимость точности нейросетевых моделей от уровня отношения С/Ш при обучении на 10дБ

В ходе эксперимента модели обучались исключительно на аудиофайлах с отношением С/Ш = 10 дБ, после чего проходили тестирование на полном диапазоне отношения С/Ш от –10 до +30 дБ. Такой подход позволяет оценить способность моделей, обученных в условиях умеренного шума, адаптироваться как к более чистым, так и к более зашумленным сигналам, отличающимся от обучающей выборки.

Анализ графика показывает, что при низких значениях отношения С/Ш (от –10 до 0 дБ) модели демонстрируют значительные различия в точности. Наиболее устойчивой к шуму оказывается архитектура CNN-BiGRU, обеспечивая наилучшие показатели точности при всех уровнях шума, особенно в условиях низкого отношения С/Ш. Модель CNN-BiLSTM также показывает высокие результаты, особенно при значениях отношения С/Ш от 0 дБ и выше. Модели CNN-LSTM и CNN-GRU демонстрируют схожую динамику и уступают по точности BiGRU и BiLSTM на низких уровнях шума. Архитектура CNN-TDNN показывает наименьшую устойчивость при отношении С/Ш ниже 0 дБ, но быстро достигает высоких значений точности при повышении отношения С/Ш и стабилизируется на уровне, близком к 99%, начиная с 15 дБ.

На рисунке 4.4 представлены результаты эксперимента, в котором пять нейросетевых архитектур, которые были обучены и протестированы на аудиоданных с фиксированным уровнем отношения С/Ш, равным 20 дБ. Уровень 20 дБ соответствует условиям практически свободной от шумов речевой среды, характерной, например, для звукозаписывающих студий или хорошо изолированных помещений. Такой уровень шума отражает почти идеальные акустические условия, в которых речевой сигнал значительно преобладает над фоновыми помехами и практически не искажается.

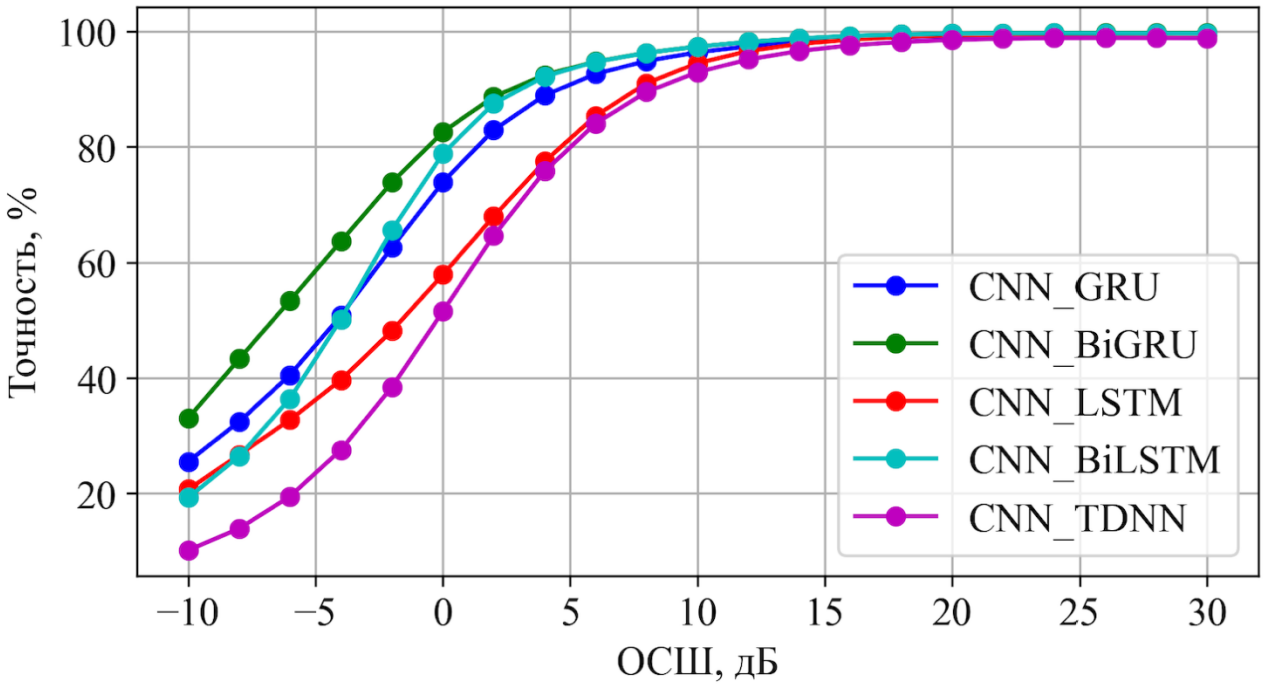


Рисунок 4.4 – Зависимость точности нейросетевых моделей от уровня отношения С/Ш при обучении на 20дБ

Как видно из графика, при приближении к уровню 20 дБ и выше все модели демонстрируют почти идеальную точность 98%, что свидетельствует о высокой обучаемости в условиях чистого сигнала. Однако по мере снижения отношения С/Ш ниже обучающего уровня точность всех моделей начинает уменьшаться. Особенно резкое падение наблюдается у моделей CNN-TDNN и CNN-LSTM при переходе к диапазону от 0 до –10 дБ, что может быть связано с их ограниченной способностью к адаптации и обобщению в условиях сильной зашумленности. Наилучшую устойчивость продемонстрировала архитектура CNN-BiGRU, которая сохраняет относительно высокую точность даже при отношении С/Ш, равном –10 дБ. Это объясняется наличием двунаправленных рекуррентных связей, способных учитывать как локальные, так и глобальные временные зависимости, что критически важно при деградации акустического сигнала.

На рисунке 4.5 представлены результаты моделей, обученных и протестированных на аудиоданных с высоким уровнем отношения сигнал/шум 30 дБ, что соответствует условиям практически полной слышимости речи и минимального влияния фоновых шумов.

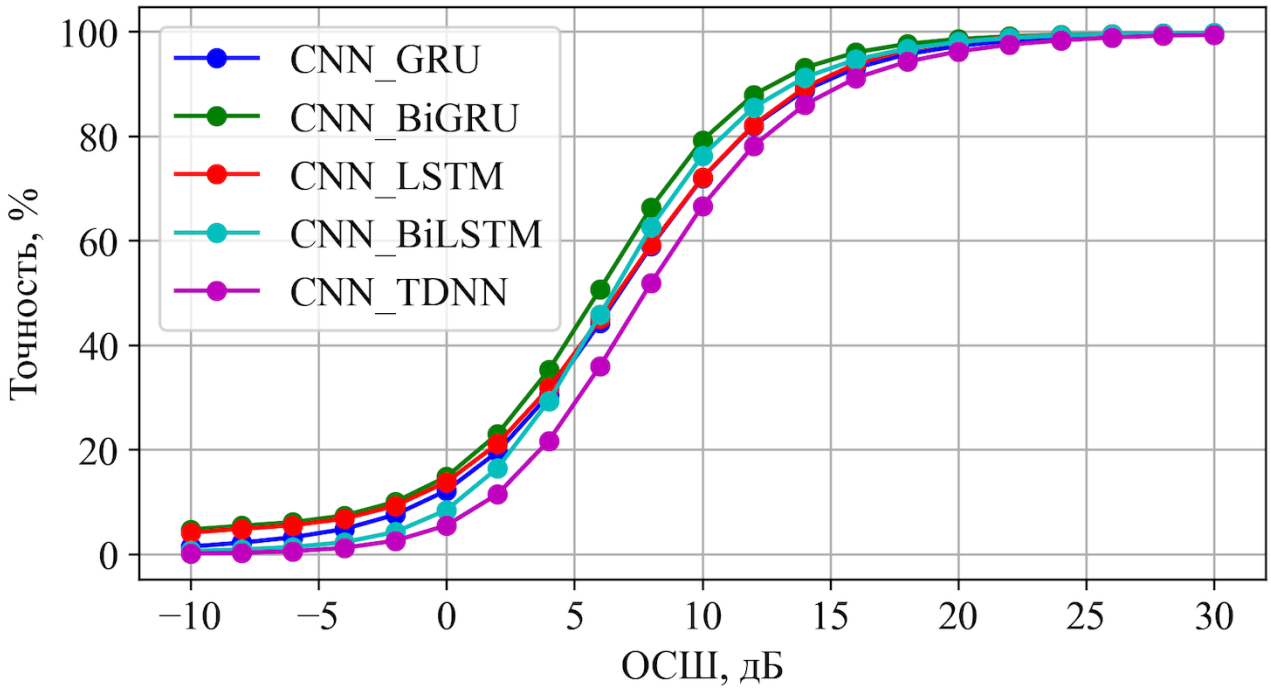


Рисунок 4.5 – Зависимость точности нейросетевых моделей от уровня отношения С/Ш при обучении на 30дБ

Анализ графика показывает, что при высоких значениях отношения С/Ш от 20 до 30 дБ все модели достигают почти максимальной точности 99%, что свидетельствует о полной адаптации к чистому речевому сигналу. Однако при снижении уровня отношения С/Ш модели начинают демонстрировать резкое падение точности, особенно в области от -10 до 0 дБ. Это связано с тем, что при обучении на чистом сигнале модель не получает достаточного опыта взаимодействия с зашумленными паттернами, и, как следствие, теряет способность к обобщению в измененных акустических условиях.

Таким образом, была проведена серия последовательных экспериментов по обучению и тестированию нейросетевых моделей для задачи детектирования голосовой активности при различных фиксированных уровнях отношения сигнал/шум. На основании проведенных экспериментов можно заключить, что обучение моделей на одном уровне шума ограничивает их обобщающую способность. В связи с этим возникает необходимость в следующем этапе исследования, предполагающем обучение на данных в диапазоне значений с переменными значениями отношения С/Ш.

4.2 Оценка устойчивости и точности нейросетевых моделей детектирования речевого сигнала при различных уровнях отношения С/Ш

В рамках данной диссертационной работы была реализована стратегия обучения моделей на аудиоданных, охватывающих широкий диапазон уровней отношения сигнал/шум от -18 дБ до +30 дБ. Такой подход позволяет приблизить условия обучения к реальным сценариям эксплуатации систем детектирования голосовой активности, где уровень шумовой нагрузки может существенно

колебаться в зависимости от акустической среды, от тихих помещений до шумных уличных пространств.

Результаты эксперимента, в котором модели обучались на аудиоданных, охватывающих широкий диапазон уровней отношения сигнал/шум представлен на рисунке 4.6 Такой подход был направлен на имитацию реальных условий работы систем голосовой активности, в которых уровень шума варьируется в зависимости от окружающей среды.

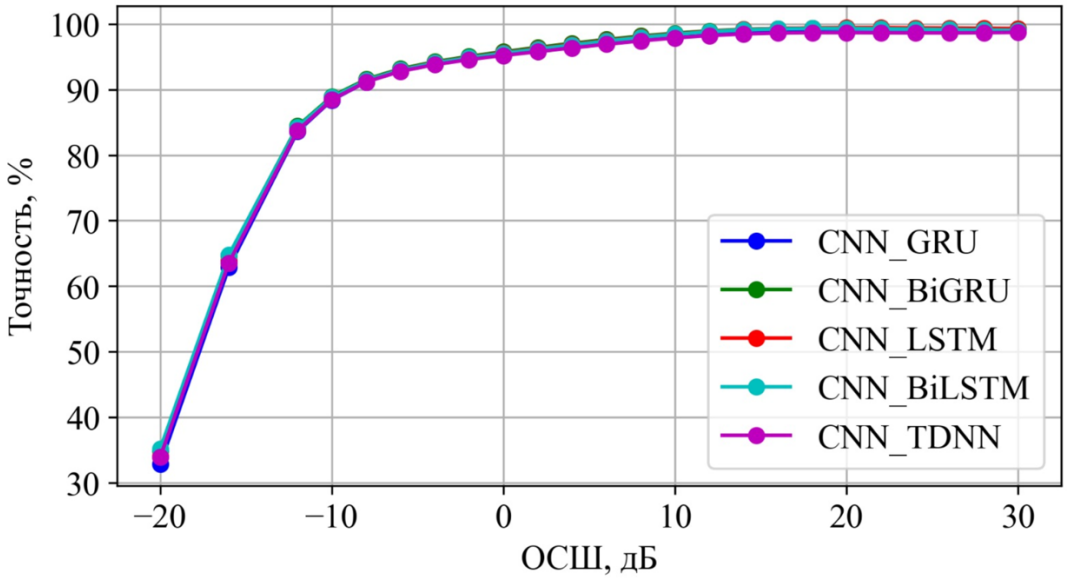


Рисунок 4.6 – Зависимость точности нейросетевых моделей от уровня отношения сигнал/шум при смешанном обучении

Анализ графика показывает, что все модели демонстрируют стабильную и высокую точность распознавания на всем диапазоне отношения С/Ш. Даже при значениях -10 дБ, где условия наиболее неблагоприятны, точность классификации не опускается ниже 88%, а при уровнях от 15 дБ и выше достигает значений свыше 98%. Это подтверждает, что обучение на множества значениях отношения С/Ш позволяет моделям формировать инвариантные к шуму признаки речи, обеспечивая устойчивость и высокую адаптивность в разнообразных акустических условиях. Полученные результаты подтверждают, что обучение на переменных уровнях шума значительно повышает надежность VAD-систем, обеспечивая высокое качество распознавания независимо от внешних акустических условий.

Соответствующие сводные значения метрик Precision, Recall и F1-score для моделей CNN+BiGRU, CNN+BiLSTM, CNN+GRU, CNN+LSTM и CNN+TDNN при различных уровнях отношения С/Ш приведены в таблице 4.1.

Таблица 4.1 – Сводные значения метрик (Precision, Recall, F1-score) для всех моделей при различных уровнях отношения С/Ш

Модель	С/Ш, дБ	Precision, %	Recall, %	F1-score, %
CNN+BiGRU	-20	30,0	99,4	46,1
	-16	62,1	99,7	76,5
	-12	83,4	99,8	90,8
	-10	88,2	99,8	93,7
	-8	91,1	99,8	95,2
	-4	94,0	99,8	96,8
	0	95,6	99,8	97,6
	4	97,0	99,8	98,4
	8	98,2	99,8	99,0
	12	99,0	99,8	99,4
	16	99,4	99,8	99,6
	20	99,5	99,8	99,7
	24	99,5	99,8	99,6
	28	99,4	99,8	99,6
CNN+BiLSTM	-20	30,3	99,4	46,4
	-16	62,2	99,7	76,6
	-12	83,2	99,8	90,8
	-10	88,1	99,8	93,6
	-8	90,9	99,8	95,2
	-4	93,8	99,8	96,7
	0	95,4	99,8	97,5
	4	96,7	99,8	98,3
	8	98,0	99,8	98,9
	12	98,9	99,8	99,4
	16	99,4	99,8	99,6
	20	99,4	99,8	99,6
	24	99,3	99,8	99,6
	28	99,2	99,8	99,5
CNN+GRU	-20	27,7	99,5	43,4
	-16	60,1	99,8	75,0
	-12	82,4	99,8	90,3
	-10	87,6	99,8	93,3
	-8	90,6	99,9	95,0
	-4	93,6	99,9	96,6
	0	95,2	99,9	97,5
	4	96,5	99,9	98,2
	8	97,7	99,9	98,8
	12	98,7	99,9	99,3
	16	99,2	99,9	99,5
	20	99,3	99,9	99,6

	24	99,3	99,9	99,6
	28	99,3	99,9	99,6
CNN+LSTM	-20	29,0	99,3	45,0
	-16	61,3	99,7	75,9
	-12	83,0	99,8	90,6
	-10	88,0	99,8	93,5
	-8	90,9	99,8	95,1
	-4	93,8	99,8	96,7
	0	95,5	99,8	97,6
	4	96,9	99,8	98,3
	8	98,1	99,8	98,9
	12	99,0	99,8	99,4
	16	99,5	99,8	99,6
	20	99,6	99,8	99,7
	24	99,6	99,8	99,7
	28	99,5	99,8	99,6
CNN+TDNN	-20	28,9	99,5	44,8
	-16	60,8	99,8	75,6
	-12	82,6	99,8	90,4
	-10	87,7	99,8	93,4
	-8	90,6	99,9	95,0
	-4	93,5	99,9	96,6
	0	95,0	99,9	97,4
	4	96,2	99,9	98,0
	8	97,4	99,9	98,6
	12	98,2	99,9	99,0
	16	98,7	99,9	99,3
	20	98,7	99,9	99,3
	24	98,7	99,9	99,3
	28	98,7	99,9	99,3

Проведенный анализ результатов, представленных в таблице 4.1, показал, что все исследуемые архитектуры нейронных сетей обеспечивают высокий уровень точности распознавания речевого сигнала при увеличении отношения С/Ш. Особенно эффективными оказались модели CNN+BiGRU и CNN+BiLSTM, продемонстрировавшие наивысшие значения F1-метрики при различных уровнях шума. Наименьшие значения F1-score наблюдались при крайне низком уровне отношения С/Ш (–20 дБ), однако уже начиная с –10 дБ все модели показали стабильный рост показателей. Это подтверждает важность использования тренировочных данных с широким диапазоном значений отношения С/Ш для повышения устойчивости и обобщающей способности VAD-моделей.

В таблице 4.2 представлены усредненные значения метрики F1-score для всех рассмотренных архитектур нейронных сетей, полученные по результатам

тестирования. F1-score был выбран в качестве основного критерия оценки, поскольку он учитывает как точность Precision, так и полноту Recall, что особенно важно в задаче детектирования голосовой активности, где важно не только обнаруживать речь, но и избегать ложных срабатываний на шум.

Таблица 4.2 – Среднее значение F1-score для всех моделей

Модель	Средний F1-score
CNN+BiGRU	97,71%
CNN+BiLSTM	97,65%
CNN+LSTM	96,98%
CNN+TDNN	96,25%
CNN+GRU	95,33%

Сводные значения F1-метрики, представленные в таблице 4.2, демонстрируют, что все рассматриваемые архитектуры нейронных сетей обладают высокой точностью при решении задачи детектирования речевого сигнала. Наивысший средний результат показала модель CNN+BiGRU – 97,71%, что подтверждает ее лучшую способность к обобщению и устойчивость к шумовым искажениям. Незначительно уступает ей модель CNN+BiLSTM с показателем 97,65%. Модели CNN+LSTM, CNN+TDNN и CNN+GRU продемонстрировали среднюю эффективность, но также сохранили значения F1-score выше 95%, что говорит об их пригодности для применения в условиях переменной зашумленности.

На рисунке 4.7 представлена тепловая карта значений F1-метрики для модели CNN+BiGRU, отражающая зависимость качества детектирования речевого сигнала от уровня отношения С/Ш.

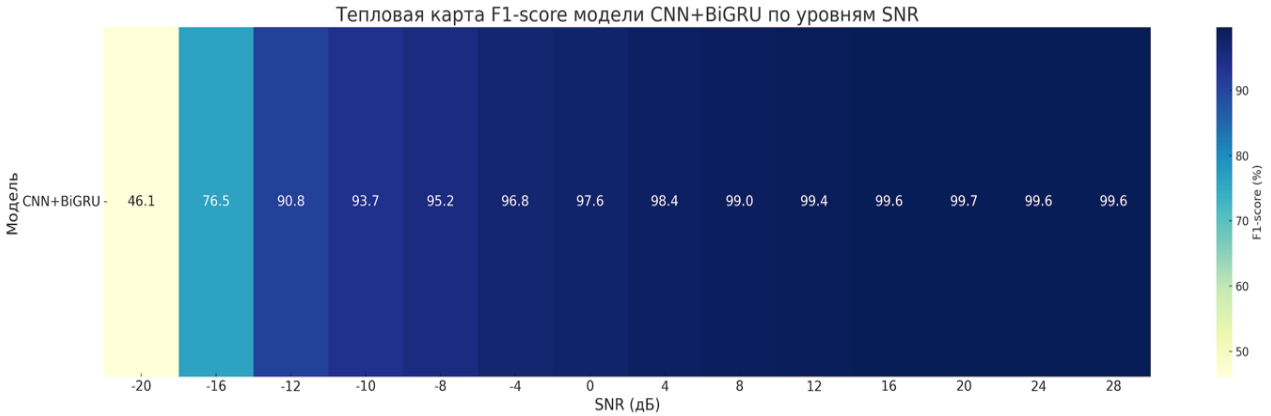


Рисунок 4.7 – Влияние уровня отношения С/Ш на значение F1-метрики модели CNN+BiGRU

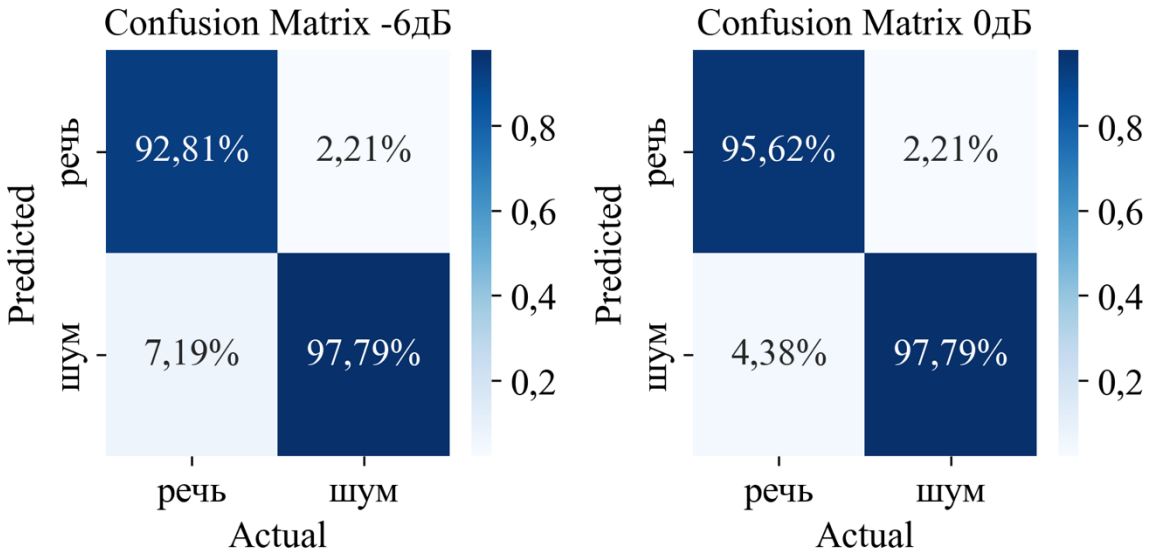
Анализ тепловой карты показывает, что при крайне низком уровне сигнала (С/Ш = –20 дБ) значение F1-score составляет 46,1%, что указывает на значительное затруднение в точной классификации речевых и неречевых фрагментов. Однако с увеличением отношения С/Ш наблюдается

стремительный рост метрики. Начиная с отношения С/Ш = –10 дБ значение F1-score превышает 90%, а уже при –4 дБ оно достигает 96,8%.

При отношении С/Ш ≥ 8 дБ модель демонстрирует почти идеальные результаты, где F1-score стабилизируется в пределах 99,0–99,6%. Это свидетельствует о высокой чувствительности архитектуры CNN+BiGRU к акустическим условиям, а также о ее способности сохранять максимальную точность детектирования в условиях умеренного и высокого отношения С/Ш.

Наивысшее среднее значение F1-score продемонстрировала модель CNN+BiGRU, что указывает на ее лучшую обобщающую способность и устойчивость к различным условиям зашумленности. Модель CNN+BiLSTM также показала высокую эффективность. Архитектуры CNN+GRU и CNN+TDNN, несмотря на меньшую вычислительную сложность, продемонстрировали немного более низкие показатели, особенно в условиях сильного шума. Тем не менее, все модели показали средние значения F1-score выше 95%, что свидетельствует об их общей пригодности к использованию в системах VAD.

На рисунке 4.8 представлены нормализованные матрицы ошибок модели CNN+BiGRU, полученные при тестировании на аудиоданных с уровнями отношения С/Ш от -6 дБ до +24 дБ. Каждая матрица отражает долю верных и ошибочных классификаций между двумя классами: речь и шум.



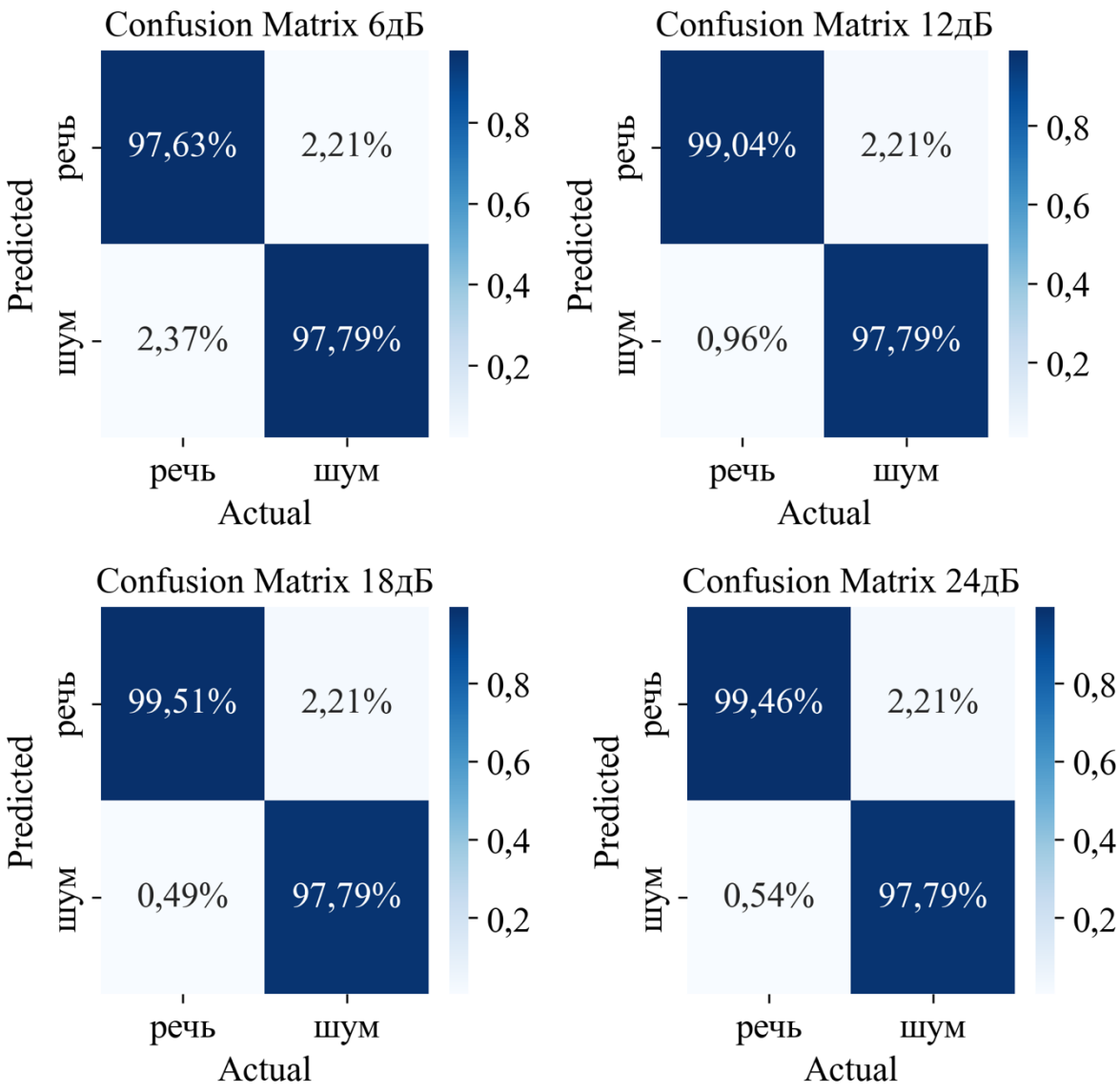


Рисунок 4.8 – Матрица ошибок CNN-BiGRU

Как видно из результатов, при низком уровне отношения С/Ш (–6 дБ) модель правильно классифицирует речевые сегменты с точностью 92,81%, а неречевые – с точностью 97,79%. По мере увеличения отношения С/Ш точность распознавания речи существенно возрастает, достигая 99,51% при 18 дБ и 99,46% при 24 дБ. При этом точность классификации шума остается стабильной на уровне 97,79% для всех тестируемых значений отношения С/Ш, что свидетельствует о высокой устойчивости модели к шумовым искажениям.

Ошибки первого рода (FN) уменьшаются с ростом отношения С/Ш. Если при –6 дБ вероятность ошибки составляет 7,19%, то при 18–24 дБ менее 0,54%. Ошибки второго рода (FP) во всех случаях держатся на уровне 2,21%.

На рисунке 4.9 представлены ROC-кривые (Receiver Operating Characteristic) и соответствующие значения площади под кривой (AUC) для модели CNN+BiGRU**,** обученной на задачу детектирования речевого сигнала при различных уровнях отношения С/Ш.

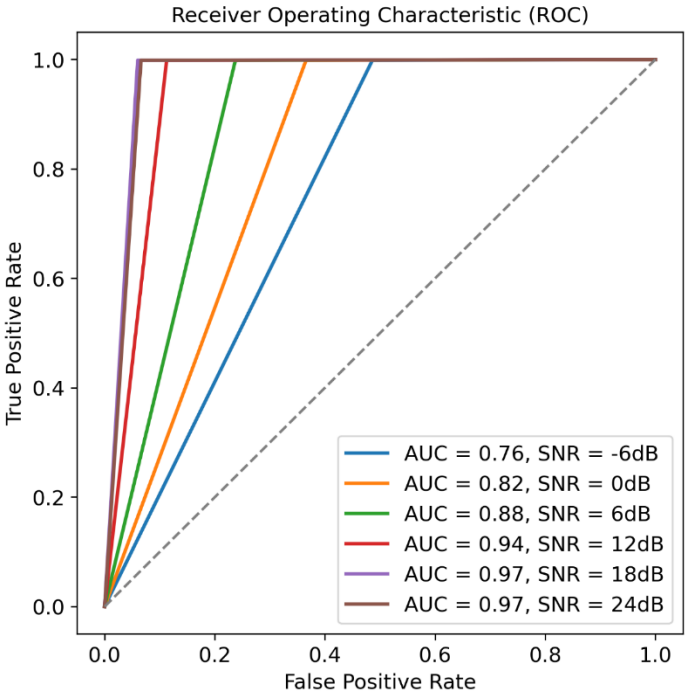


Рисунок 4.9 – Кривые ROC и значения AUC для модели CNN+BiGRU при различных уровнях отношения С/Ш

Как видно из графика, увеличение уровня отношения С/Ш приводит к улучшению качества модели. При низком уровне сигнала (С/Ш = –6 дБ) AUC составляет 0,76, что указывает на умеренное качество классификации. Однако по мере увеличения отношения С/Ш значение AUC стремительно возрастает и достигает 0,97 при С/Ш = 18 дБ и 24 дБ, что соответствует практически идеальной работе классификатора.

Таким образом, ROC-анализ подтверждает высокую чувствительность модели к качеству входного сигнала: чем выше уровень отношения С/Ш, тем выше достоверность детектирования речевого сигнала, что необходимо учитывать при внедрении системы в условиях реальной акустической среды.

4.3 Разработка программного приложения для оценки производительности нейронных сетей

В условиях современного информационного общества образовательные системы переживают значительные трансформации, обусловленные активным внедрением цифровых технологий. Такие изменения неизбежно приводят к переосмыслению традиционной роли преподавателей, а также к пересмотру подходов к передаче знаний и восприятию информации студентами. В настоящее время цифровые инструменты становятся неотъемлемой частью образовательного процесса, и их интеграция в учебные цели приобретает критическое значение для повышения качества образования. Предоставление учебного материала через интерактивные платформы, включая виртуальные лаборатории и практические занятия, создает для студентов более удобные и доступные условия обучения. Применение цифровых технологий в образовательной среде, безусловно, способствует повышению эффективности

учебного процесса. Более того, оно активно способствует развитию у студентов цифровых навыков, которые они смогут успешно применять в своей будущей профессиональной деятельности.

В современном образовательном процессе сочетание теоретических знаний с практическими занятиями в лабораторных условиях приобретает все большее значение как важный элемент формирования профессиональных компетенций у студентов технических специальностей. Лабораторные работы предоставляют уникальную возможность студентам экспериментировать, осваивать различные методики и совершенствовать свои навыки в условиях, максимально приближенных к реальным. Такие занятия играют главную роль в том, чтобы теоретические знания могли быть применены на практике, позволяя студентам решать конкретные задачи и развивать навыки, необходимые для успешного преодоления возникающих трудностей. В результате, студенты не только укрепляют уверенность в своих профессиональных возможностях, но и получают всестороннюю подготовку, необходимую для успешного выполнения профессиональных обязанностей в будущем.

Такие методы обучения также играют главную роль в развитии самостоятельности, критического мышления и творческого подхода к выполнению задач. Участвуя в лабораторных работах, студенты приобретают ценный опыт, который существенно способствует более глубокому усвоению изучаемого материала и значительно укрепляет их способность самостоятельно анализировать и эффективно решать возникающие проблемы [92-95].

В современных условиях многие технические вузы Республики Каазхстан используют аппаратные платформы для организации и проведения разнообразных экспериментов и практических занятий. Одной из таких платформ является LabVIEW, разработанная компанией National Instruments [96].

Однако, такие лабораторные установки могут быстро морально устаревать из-за изменений в технологиях и электронной базе. Быстрые темпы развития в области техники и науки приводят к тому, что оборудование и установки, которые сегодня используются для проведения лабораторных и практических занятий, могут оказаться устаревшими уже через несколько лет. Электроника, используемая в лабораторных установках, также подвержена устареванию из-за постоянного появления новых моделей и более современных технологий. Это может привести к тому, что студенты обучаются на устаревшем оборудовании, что не соответствует требованиям современной общества и может ограничить в будущем их профессиональные возможности.

Для решения проблемы быстрого морального устаревания лабораторного оборудования и электронной базы, эффективным решением может быть использование программных приложений, которые могут заменить реальные эксперименты и постоянно улучшаться. Такие виртуальные лаборатории позволяют студентам проводить практические работы и эксперименты, используя компьютеры или мобильные устройства, без необходимости доступа к дорогостоящему оборудованию.

Программные приложения для виртуальных лабораторий могут быть созданы специально для обучения определенному курсу или предмету, и могут содержать различные симуляции, интерактивные упражнения и учебные материалы. Кроме того, приложения могут предоставлять студентам широкие возможности для применения полученных знаний на практике, развивать навыки экспериментирования и помогать им лучше понимать концепции и принципы изучаемого материала [97-98].

Таким образом, программные приложения могут быть разработаны специально для определенных учебных курсов или дисциплин, включая в себя разнообразные симуляции, интерактивные упражнения и учебные материалы. Виртуальные лаборатории не только расширяют возможности для применения теоретических знаний на практике, но и способствуют развитию навыков экспериментировать, что помогает студентам лучше усваивать сложные учебные материалы. К тому же благодаря постоянным обновлениям и улучшениям, такие приложения позволяют поддерживать актуальность учебного процесса, обеспечивая студентам доступ к самым современным методам обучения.

Виртуальные лабораторные занятия имеют ряд значительных преимуществ перед традиционными методами обучения. Они обеспечивают более наглядное представление изучаемых процессов, что является более экономичным и безопасным. Программное обеспечение для виртуальных лабораторных работ может быть разработано на различных языках программирования с использованием разнообразных инструментов, открываядополнительные возможности для обучения студентов.

Исследование, проведенное в Университете Мумбаи, на которое ссылается работа [99], продемонстрировало положительное восприятие студентами использования виртуальных лабораторий для изучения мобильной связи. Более 90% опрошенных студентов выразили удовлетворение от работы с виртуальными лабораториями и отметили, что такой подход значительно помогает им в понимании учебного материала. Кроме того, свыше 80% студентов указали, что проведение экспериментов способствует более эффективному усвоению теоретических знаний. Результаты исследования также показали, что студенты, использовавшие виртуальные лаборатории, продемонстрировали более высокие результаты в обучении по сравнению с теми, кто не имел такой возможности, что свидетельствует о повышении успеваемости благодаря этому методу.

Исследование, проведенное в Институте технологии и науки Бирлы в Пилани, в котором приняли участие 270 студентов [100], представило убедительные данные о влиянии имитационных лабораторий на понимание теоретического материала и академическую успеваемость. Участников разделили на две группы: первая группа (G1) работала с учебной программой, включающей элементы имитации, в то время как вторая группа (G2) использовала программу без таких элементов. Результаты показали значительное улучшение успеваемости среди студентов группы G1, в то время как показатели группы G2 остались неизменными.

Таким образом, использование виртуальных и имитационных лабораторий существенно способствует более глубокому усвоению теоретического материала и повышению академических достижений студентов [101-102].

В рамках данной диссертационной работы была разработана виртуальная лаборатория для исследования производительности нейронных сетей в задаче распознавания речевых сигналов, который представлен на рисунке 4.10. Данная виртуальная лабораторно-исследовательская программа позволяет студентам изучать основы работы нейронных сетей и их применение в данной области. Разработанная виртуальная лабораторно-исследовательская программа предоставляет возможность проводить эксперименты с различными параметрами и архитектурами нейронных сетей.

Следовательно, разработанная виртуальная лабораторно-исследовательская программа представляет собой инструмент, который помогает студентам визуализировать и понять принципы работы нейронных сетей, а также применять их на практике. Оно обеспечивает удобный и доступный способ изучения сложных концепций и методов работы нейронных сетей в условиях распознавания речевого сигнала.

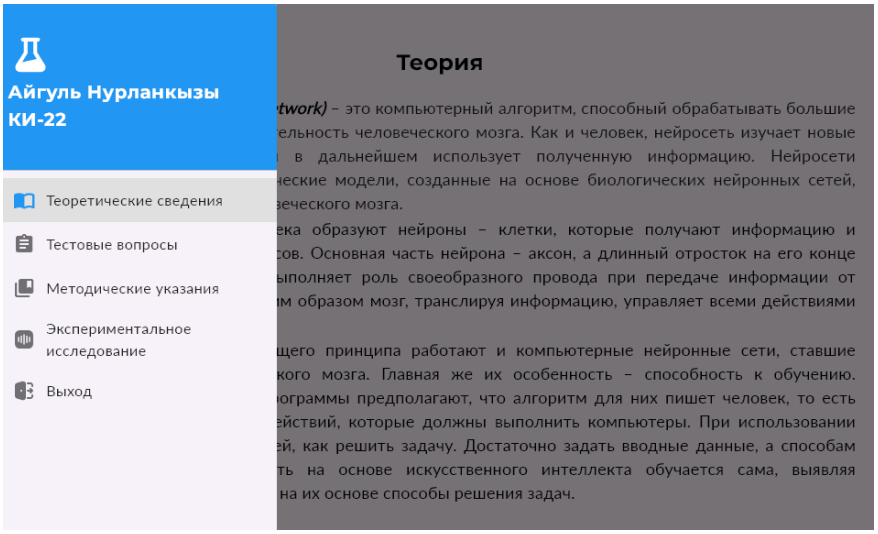


Рисунок 4.10 — Общий вид разработанной виртуальной лабораторно-исследовательской программы

Для успешного освоения курса студентам важно в первую очередь уделить внимание тщательному изучению теоретического материала, который представлен на рисунке 4.11. Глубокое понимание основ и основных концепций создаст прочную базу для дальнейшего практического применения знаний.

Рисунок 4.11 – Теоретический материал программы

С целью успешного выполнения лабораторной работы студентам необходимо пройти тестирование по изученному теоретическому материалу о нейронных сетях. Тест состоит из 20 вопросов. Каждый студент должен дать ответы на все вопросы. Для получения положительной оценки студентам необходимо набрать не менее 50% правильных ответов. Оценка результатов тестовых заданий представлена на рисунке 4.12.

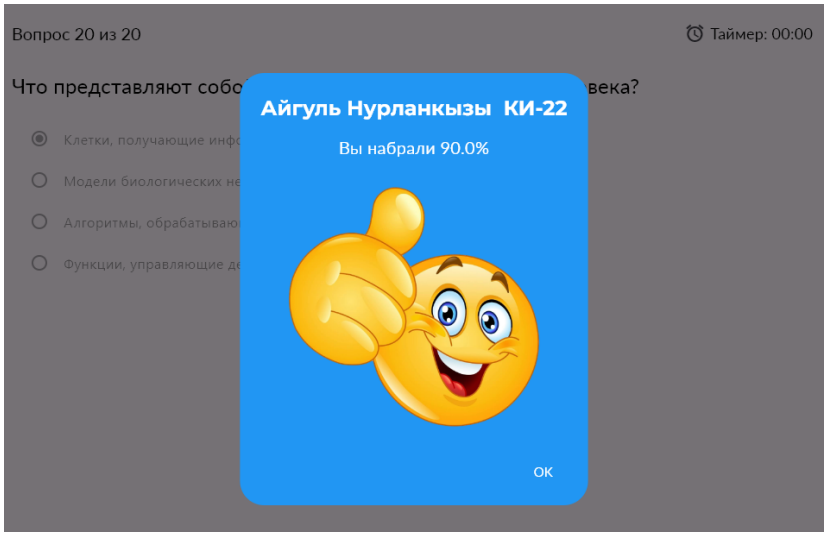


Рисунок 4.12 – Оценка результатов тестовых заданий

Таким образом, тестирование является важной частью обучения, которое позволяет студентам проверить свои знания, также помогает преподавателю оценить уровень усвоения материала студентами. Прохождение теста по нейронным сетям даст студентам возможность закрепить полученные знания, позволяя лучше подготовиться к дальнейшей работе. Такие требования необходимы для того, чтобы удостовериться, что студент действительно усвоил

теоретический материал. Также хорошее усвоение теоретического материала даст возможность успешно применять полученные знания на практике. Такой подход обеспечит глубокое понимание темы и развивает навыки студентам, необходимые для решения реальных задач.

В ходе выполнения данной лабораторной работы необходимо определить, какая из нейронных сетей демонстрирует наилучшие результаты. Следовательно, также необходимо провести детальный сравнительный анализ их эффективности. Также важно проанализировать, как языковые особенности могут влиять на производительность нейронных сетей в задаче распознавания речевого сигнала.

Таким образом, в рамках разработанной виртуальной лабораторно-исследовательской работы студенты могут проводить сравнительный анализ производительности трех типов нейронных сетей, таких как: CNN, RNN и MLP в задаче распознавания речевого сигнала на различных языках. Такое экспериментальное исследование позволит выявить значимые различия в производительности данных сетей в задаче распознавания речевого сигнала на разных языках. Методические указания к выполнению данной лабораторной работы представлены на рисунке 4.13.

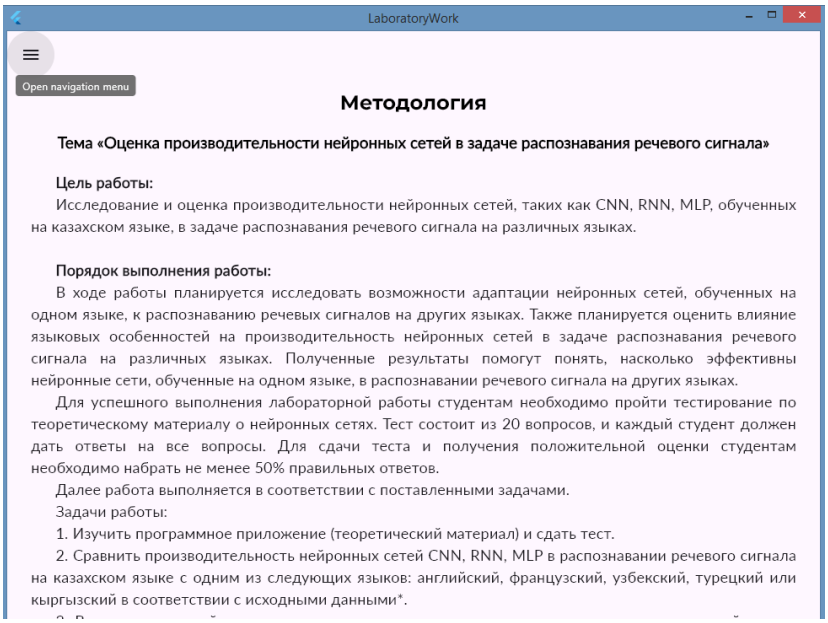


Рисунок 4.13 – Методические указания по выполнению лабораторной работы

Далее по необходимости надо добавлять различные уровни шума к исходным данным, оценивая его влияния на точность распознавания сигналов каждой из нейронных сетей (CNN, RNN и MLP) на обоих выбранных языках, как показано на рисунках 4.14-4.16. В конце необходимо написать и оформить отчет, с формулируя выводы о том, какая нейронная сеть является наиболее эффективной, основываясь на проведенных экспериментах и полученных результатах.

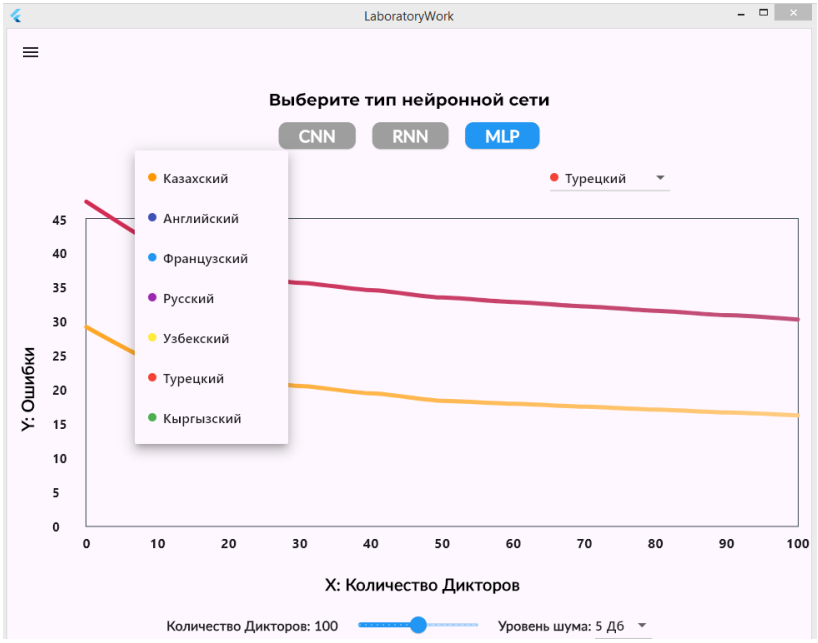


Рисунок 4.14 – График зависимости ошибок от количества дикторов для нейронной сети MLP

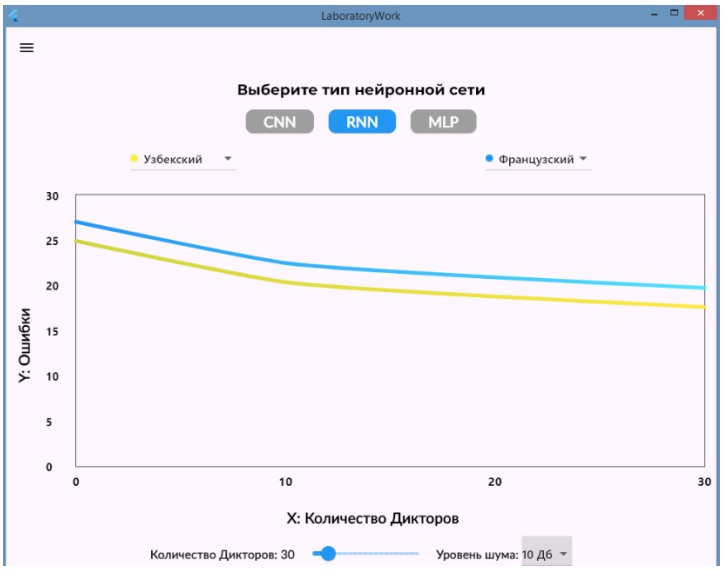


Рисунок 4.15 – График зависимости ошибок от количества дикторов для нейронной сети RNN

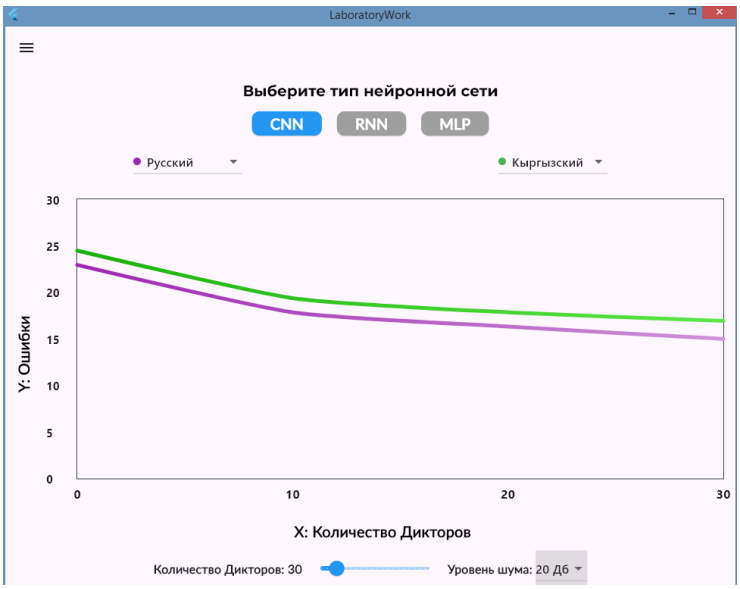


Рисунок 4.16 – График зависимости ошибок от количества дикторов для нейронной сети CNN

Кроме этого, необходимо отметить, что анализ результатов исследования позволит определить, как количество дикторов влияет на точность и надежность работы нейронных сетей в задаче распознавания речевого сигнала. Также в рамках данного исследования можно добавлять разные уровни шума в процесс обучения и тестирования нейронных сетей. Это сделано для того, чтобы оценить влияния шума на точность распознавания речевого сигнала. Такой подход позволяет провести более глубокий анализ и понять, как шум может повлиять на производительность каждого типа нейронной сети в задаче распознавания речевого сигнала.

Кроме того, результаты данного исследования позволят лучше понять, как шум влияет на точность распознавания речевого сигнала нейронными сетями и определить оптимальные стратегии работы с шумом при разработке систем распознавания речи на разных языках. Полученные данные могут быть полезны при дальнейшем совершенствовании алгоритмов и моделей нейронных сетей для задач распознавания речи в условиях реального мира.

Таким образом, программа по изучению нейронных сетей для распознавания речи предоставляет студентам ценный опыт и практические навыки в области машинного обучения и искусственного интеллекта. Она поможет им лучше понять особенности работы нейронных сетей, подготовиться к будущей карьере в сфере технологий и сделать вклад в развитие инноваций искусственного интеллекта. Такие программы значимы для студентов, интересующихся передовыми областями информационных технологий и ищущих возможности для применения теоретических знаний на практике и в реальных проектах.

Выводы по главе:

1. Подтверждено, что архитектуры CNN-BiGRU и CNN-BiLSTM демонстрируют наивысшую устойчивость и точность на широком диапазоне

уровней зашумленности, особенно при неблагоприятных условиях, благодаря способности учитывать временной контекст в обеих временных направлениях. Подтверждено, что модели, обученные при умеренном уровне С/Ш (10–20 дБ), показывают лучший баланс между точностью и обобщающей способностью. Таким образом, обучение на фиксированном уровне шума ограничивает адаптивность, а оптимальным решением является использование обучающих данных с разнообразными значениями С/Ш.

2. Установлено, что все архитектуры достигли высокой точности классификации уже при умеренном отношении С/Ш. Архитектура CNN+BiGRU показала наилучший результат среди всех моделей, продемонстрировав максимальное значение F1-score (97,71%) и высокую стабильность при различных акустических условиях. При этом модели CNN+BiLSTM и CNN+LSTM также продемонстрировали высокую эффективность, в то время как CNN+GRU и CNN+TDNN показали меньшую устойчивость в условиях сильного шума, но сохранили средние значения F1-score выше 95%. Анализ матриц ошибок и ROC-кривых подтвердил, что точность модели CNN+BiGRU значительно возрастает с увеличением С/Ш, достигая почти идеальных значений, что делает ее оптимальной для использования в системах VAD в реальной шумовой среде.

3. В соответствии с основными результатами, полученными в рамках данной диссертационной работы, была разработана виртуальная лабораторно-исследовательская работа «Оценка производительности нейронных сетей в задаче распознавания речевого сигнала», предназначенная для использования в учебных их целях. Данная разработка позволяет не только наглядно наблюдать и изучать работу различных архитектур нейронных сетей, но и проводить сравнительный анализ их производительности, устойчивости к шуму и способности адаптироваться к различным языковым условиям.

ЗАКЛЮЧЕНИЕ

В современных системах распознавания речи, особенно при низком уровне отношения С/Ш, традиционные методы VAD часто не обеспечивают требуемой точности. Методы на основе нейросетей значительно повышают надежность детектирования за счет обучения на больших объемах данных и способности учитывать сложные нелинейные зависимости, что делает их особенно полезными в шумных условиях. Традиционные подходы, такие как пересечение нуля, энергетический анализ, линейное прогнозирование и одночастотная фильтрация, эффективны при высоком отношении С/Ш, но теряют результативность в сложной акустической обстановке. Напротив, методы глубокого обучения демонстрируют устойчивую работу и хорошую адаптацию к изменениям шума. Современные VAD на базе нейронных сетей превосходят классические алгоритмы по точности, устойчивости и адаптивности, что делает их предпочтительными для практического применения.

Среди негибридных архитектур нейронных сетей наилучшие результаты в задаче детектирования речевого сигнала показали модели типа CNN, благодаря способности эффективно извлекать локальные спектральные признаки и устойчивости к шуму за счет операций субдискретизации. Рекуррентные архитектуры LSTM и GRU оказались полезными для обработки длительных временных зависимостей, а их двунаправленные версии BiLSTM и BiGRU продемонстрировали более точную идентификацию границ речевых сегментов за счет учета как предыдущего, так и последующего контекста. TDNN-модели обеспечили быструю обработку сигнала благодаря использованию фиксированных временных смещений. Применение гибридных архитектур, объединяющих сверточные и рекуррентные компоненты (например, CNN+BiGRU и CNN+BiLSTM), позволило значительно повысить точность распознавания речи даже при низком уровне отношения С/Ш, эффективно совмещая пространственный и временной анализ признаков. При этом архитектура CNN+TDNN выделилась минимальным временем обучения при сохранении высокой точности. Дополнительно было показано, что точность работы моделей зависит от количества дикторов, участвующих в обучении, а также от соответствия языка обучения и тестирования. Использование нейронных сетей для управления ошибками первого рода позволяет оптимизировать точность при минимальных объемах обучающих данных, делая такие решения особенно актуальными в системах VAD.

В рамках данной диссертационной работы для детектирования речевого сигнала эффективно применялся метод MFCC, который точно отражает спектральные характеристики с учетом особенностей слуха человека. В работе использовались размеченные данные казахского речевого корпуса с добавленным искусственным шумом на разных уровнях отношения С/Ш, что позволило сформировать устойчивые обучающие выборки. Последовательная обработка сигнала от фреймирования до дискретного косинусного преобразования обеспечила извлечение компактных признаков, пригодных для классификации в шумных условиях.

Также, в ходе эксперимента в рамках данной диссертационной работы гибридные архитектуры, сочетающие сверточные и рекуррентные слои (CNN+BiGRU, GRU, BiLSTM, LSTM, TDNN), продемонстрировали высокую точность, стабильную сходимость и отсутствие переобучения. BiGRU и BiLSTM оказались наиболее эффективными благодаря учету двустороннего контекста, а TDNN обеспечила минимальное время обучения при сохранении высокой точности. Лучшую метрику F1-score (97,71%) показала модель CNN+BiGRU, что указывает на ее превосходство среди протестированных решений. Оптимальный выбор архитектуры позволяет достичь баланса между точностью, устойчивостью к шуму и вычислительной эффективностью. На точность также влияет пороговое значение на выходе модели: его изменение позволяет управлять типами ошибок без переобучения. Совпадение языка обучающей и тестовой выборок снижает ошибку, а фонетические различия между языками увеличивают ее. Снижение отношения С/Ш ухудшает результаты, но обучение на зашумленных данных повышает устойчивость. Увеличение числа дикторов в обучении улучшает обобщающую способность модели.

Наибольшую устойчивость и точность в условиях различной зашумленности продемонстрировали архитектуры CNN-BiGRU и CNN-BiLSTM, особенно эффективные при низком отношении С/Ш за счет учета временного контекста в обоих направлениях. Наилучшие результаты достигаются при обучении на умеренном уровне шума (10–20 дБ), что обеспечивает оптимальный баланс между точностью и способностью к обобщению. Обучение на фиксированном уровне шума ограничивает гибкость модели, тогда как использование обучающих данных с различными значениями отношения С/Ш повышает ее адаптивность.

Среди всех протестированных моделей CNN+BiGRU показала максимальный F1-score (97,71%) и высокую стабильность при разных уровнях шума, в том числе в неблагоприятных акустических условиях. Модели CNN+BiLSTM и CNN+LSTM также обеспечили высокую точность, а CNN+GRU и CNN+TDNN показали чуть меньшую устойчивость к шуму, но сохранили F1-score выше 95%. Анализ матриц ошибок и ROC-кривых подтвердил значительное улучшение точности модели CNN+BiGRU с ростом отношения С/Ш, что делает ее перспективной для использования в системах VAD.

Дополнительно также в рамках данной диссертационной работы была создана виртуальная лабораторно-исследовательская работа, направленная на оценку производительности нейронных сетей в задаче распознавания речи. Программа ориентирована на образовательное применение и позволяет студентам исследовать архитектуры нейросетей, анализировать их устойчивость к шуму и языковую адаптивность, а также проводить сравнительный анализ эффективности моделей в практических условиях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Rabiah, Sitti. (2018). Language as a Tool for Communication and Cultural Reality Discloser. 10.31227/osf.io/nw94m.

2 Jakubec, M., Lieskovska, E., Jarina, R., Spisiak, M., & Kasak, P. (2024). Speech Emotion Recognition Using Transfer Learning: Integration of Advanced Speaker Embeddings and Image Recognition Models. Applied Sciences, 14(21), 9981. https://doi.org/10.3390/app14219981

3 Wang, Runze (2022) Voice Activity Detection Based on Deep Neural Networks. Masters thesis, Concordia University.

4 Dhouib, Amira & Othman, Achraf & Elghoul, Oussama & Khribi, Mohamed Koutheair & Sinani, Aisha. (2022). Arabic Automatic Speech Recognition: A Systematic Literature Review. Applied Sciences. 12. 8898.10.3390/app12178898.

5 Оралбекова Д. О. Разработка системы автоматического распознавания речи на основе интегрального подхода. Диссертация на соискание ученой степени доктора философии (PhD), Алматы, 2022

6 Sadjadi, S. O., Ko, T., Seltzer, M. L., & Liao, H. (2016). Environmental noise embeddings for robust speech recognition. arXiv preprint arXiv:1601.02553. https://arxiv.org/abs/1601.02553

7 Есенбаев Ж. А. Распознавание казахской речи по определенной словарной базе в условиях шумов. Диссертация на соискание ученой степени доктора философии (PhD), Астана, 2014

8 George Boateng, Prabhakaran Santhanam, Janina Lüscher, Urte Scholz, and Tobias Kowatsch. 2019. VADLite: an open-source lightweight system for real-time voice activity detection on smartwatches. In Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Association for Computing Machinery, New York, NY, USA, 902–906. https://doi.org/10.1145/3341162.3346274

9 Han I., Om C.-N., Kim U.-I. A gated recurrent unit based robust voice activity detector (2024) Multimedia Tools and Applications, 83 (14), pp. 41939 – 41949. DOI: 10.1007/s11042-023-17123-w

10 3GPP. (2011). Performance characterization of the Adaptive Multi-Rate Wideband (AMR-WB) speech codec (3GPP TR 26.976 V10.0.0). European Telecommunications Standards Institute (ETSI).

11 Graf, S., Herbig, T., Buck, M. et al. Features for voice activity detection: a comparative analysis. EURASIP J. Adv. Signal Process. 2015, 91 (2015). https://doi.org/10.1186/s13634-015-0277-z

12 Seshashyama S.M., Rufus A. A Survey and Evaluation of Voice Activity Detection Algorithms. Karlskrona, June 2011

13 Z. Chen and B. Xu, “Optimization of speech endpoint detection base on sub-band energy feature,” Acta Acustica, pp. 171–176, 2005

14 Marco C., ANALOG VOICE ACTIVITY DETECTION. 2018

15 Aalto University. (n.d.). Voice Activity Detection - Introduction to Speech Processing. Retrieved April 20, 2025, from https://speechprocessingbook.aalto.fi/Recognition/Voice_activity_detection.html

16 R. Le Bouquin-Jeannès and G. Faucon, "Study of a voice activity detector and its influence on a noise reduction system," Speech Commun., vol. 16, pp. 245-254, 4, 1995.

17 Парамонов П.А. Методы, алгоритмы и устройства распознания речи. Диссертация на соискание ученой степени кандидата технических наук, Москва, 2015, стр 19

18 A. Davis and S. Nordholm, "A low complexity statistical voice activity detector with performance comparisons to ITU-T/ETSI voice activity detectors," in Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on, 2003, pp. 119-123 Vol.1.

19 S. Haykin, Communication Systems, 3rd ed. New York: Wiley, 1994.

20 D. K. Freeman, G. Cosier, C. B. Southcott and I. Boyd, "The voice activity detector for the pan-european digital cellular mobile telephone service," in Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on, 1989, pp. 369-372 vol.1.

21 Hu, Y. and Loizou, P. (2007). “Subjective evaluation and comparison of speech enhancement algorithms,” Speech Communication, 49, 588-601.

22 R. G. Bachu, S. Kopparthi, B. Adapa and B. D. Barkana, "Voiced/Unvoiced Decision for Speech Signals Based on Zero-Crossing Rate and Energy," 2010.

23 Abdullah I. Al-Shoshan, Speech and Music Classification and Separation: A Review, Journal of King Saud University - Engineering Sciences, Volume 19, Issue 1, 2006, 95-132, https://doi.org/10.1016/S1018-3639(18)30850-X.

24 S. B. /jebara, “Multi-band coherence features for voiced-voiceless-silence speech classi- fication,” in 2nd International Conference on Information & Communication Tech- nologies. IEEE, October 2006, pp. 1248–1253.

25 S. R. M. P. Sarfaraz Jelil, Rohan Kumar Das and R. Sinha, “Role of voice activity detection methods for the speakers in the wild challenge,” in NCC, October 2017, pp. 1–6.

26 S. A. S.A. Soleimani, “Voice activity detection based on combination of multiple features using linear/kernel discriminant analyses,” in 3rd International Conference on Information and Communication. IEEE, May 2008, pp. 1–5.

27 D. R. B. Tejus Adiga M, “Improving single frequency filtering based voice activity detec- tion (vad) using spectral subtraction based noise cancellation,” in International confer- ence on Signal Processing, Communication, Power and Embedded System (SCOPES). IEEE, June 2016, pp. 18–23.

28 G. Aneeja and B. Yegnanarayana, “Single frequency filtering approach for discriminat- ing speech and nonspeech,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 4, pp. 705–717, February 2015.

29 Qu Yang, Qianhui Liu, Nan Li, Meng Ge, Zeyang Song, Haizhou Li. SVAD: A ROBUST, LOW-POWER, AND LIGHT-WEIGHT VOICE ACTIVITY DETECTION WITH SPIKING NEURAL NETWORKS. https://doi.org/10.48550/arXiv.2403.05772

30 D. Dean, S. Sridharan, R. Vogt, and M. Mason, “The qut-noise-timit corpus for evaluation of voice activity detection algorithms,” in Interspeech, 2010, pp. 3110–3113.

31 Faghani, M.; Rezaee-Dehsorkh, H.; Ravanshad, N.; Aminzadeh, H. Ultra-Low-Power Voice Activity Detection System Using Level-Crossing Sampling. Electronics 2023, 12, 795. https://doi.org/10.3390/electronics12040795

32 Ma, Y. Efficient Voice Activity Detection and Speech Enhancement Algorithms Based on Spectral Features. Ph.D. Thesis, Tokyo Institute of Technology, Tokyo, Japan, 2014.

33 Kim, J.; Hahn, M. Voice Activity Detection Using an Adaptive Context Attention Model. IEEE Signal Process. Lett. 2018, 25, 1181–1185.

34 Muhammad Hilmi Faridh, Ulil Surtia Zulpratita. HiVAD : A Voice Activity Detection Application Based on Deep Learning. ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika. Vol. 9 | No. 4 | Halaman 856 - 866 Oktober 2021

35 Zohar, J., César, S., Jason, F., Yuxin, P., Hereman, N., & Adhish, T. (2018). Free Spoken Digit Dataset (FSDD). Retrieved from https://www.kaggle.com/joserzapata/free-spokendigit-dataset-fsdd.

36 Mesaros, Annamaria, Heittola, Toni, & Virtanen, T. (2017). TUT Acoustic scenes 2017. Zenodo. Retrieved from https://zenodo.org/record/400515#.YI0uhbUzbIU

37 Mihalache, S.; Burileanu, D. Using Voice Activity Detection and Deep Neural Networks with Hybrid Speech Feature Extraction for Deceptive Speech Detection. Sensors 2022, 22, 1228. https://doi.org/10.3390/s22031228

38 Zhang, Xiao-Lei & Xu, Menglong. (2022). AUC optimization for deep learning-based voice activity detection. EURASIP Journal on Audio, Speech, and Music Processing. 2022. 10.1186/s13636-022-00260-9.

39 V. Panayotov, G. Chen, D. Povey and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 2015, pp. 5206-5210, doi: 10.1109/ICASSP.2015.7178964.

40 YingWei Tan, and XueFeng Ding. Heterogeneous Convolutional Recurrent Neural Network with Attention Mechanism and Feature Aggregation for Voice Activity Detection. APSIPA Transactions on Signal and Information Processing, 2024, 13

41 D. P. Kingma and J. Ba, “ Adam: A method for stochastic optimization”, arXiv preprint arXiv:1412.6980.

42 Oh, Y.R.; Park, K.; Park, J.G. Online Speech Recognition Using Multichannel Parallel Acoustic Score Computation and Deep Neural Network (DNN)- Based Voice-Activity Detector. Appl. Sci. 2020, 10, 4091. https://doi.org/10.3390/app10124091

43 Cámbara, G.; López, F.; Bonet, D.; Gómez, P.; Segura, C.; Farrús, M.; Luque, J. TASE: Task-Aware Speech Enhancement for Wake-Up Word Detection in Voice Assistants . Appl. Sci. 2022, 12, 1974. https://doi.org/10.3390/app12041974

44 F. Jia, S. Majumdar and B. Ginsburg, "MarbleNet: Deep 1D Time-Channel Separable Convolutional Neural Network for Voice Activity Detection," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, 2021, pp. 6818-6822, doi: 10.1109/ICASSP39728.2021.9414470.

45 P. Warden, “Speech commands: A dataset for limitedvocabulary speech recognition,” arXiv:1804.03209, 2018.

46 F. Font, G. Roma, and X. Serra, “Freesound technical demo,” in ACM International Conference on Multimedia (MM’13), 2013.

47 Shum, Stephen & Dehak, Najim & Dehak, R. & Glass, James. (2013). Unsupervised Methods for Speaker Diarization: An Integrated and Iterative Approach. Audio, Speech, and Language Processing, IEEE Transactions on. 21. 2015-2028. 10.1109/TASL.2013.2264673

48 S. O. Sadjadi and J. H. Hansen, “Unsupervised speech activity detection using voicing mea- sures and perceptual spectral fux,” IEEE signal processing letters, vol. 20, no. 3, pp. 197–200, 2013

49 P. Teng and Y. Jia, “Voice activity detection via noise reducing using non-negative sparse coding,” IEEE Signal Processing Letters, vol. 20, no. 5, pp. 475–478, 2013.

50 S. Mousazadeh and I. Cohen, “Voice activity detection in presence of transient noise using spectral clustering,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 6, pp. 1261–1271, 2013.

51 B. Luo, Z. Pei, L. Xu, and D. L. Hu, “A new method based on hmms and k-means algo- rithms for noise-robust voice activity detector,” in Applied Mechanics and Materials, vol. 128, pp. 461–464, Trans Tech Publ, 2012.

52 T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Vesely, and P. Mat ` ejka, ˇ “Developing a speech activity detection system for the darpa rats program,” in Thirteenth annual conference of the international speech communication association, 2012

53 J. Wu and X.-L. Zhang, “Effcient multiple kernel support vector machine based voice activity detection,” IEEE Signal Processing Letters, vol. 18, no. 8, pp. 466–469, 2011.

54. H. Veisi and H. Sameti, “Hidden-markov-model-based voice activity detector with high speech detection rate for speech enhancement,” IET signal processing, vol. 6, no. 1, pp. 54–63, 2012.

55 Ozbayoglu, Murat & Gudelek, Ugur & Sezer, Omer. (2020). Deep learning for financial applications : A survey. Applied Soft Computing. 106384. 10.1016/j.asoc.2020.106384

56. Liu, Y., Xu, D., Wang, C., & Wu, J. (2024). A comprehensive study of lightweight speech recognition models. arXiv. https://arxiv.org/abs/2402.15490

57. Cheng, Qixiang & Kwon, Jihye & Glick, Madeleine & Bahadori, Meisam & Carloni, Luca & Bergman, Keren. (2020). Silicon Photonics Codesign for Deep Learning. Proceedings of the IEEE. PP. 1-22. 10.1109/JPROC.2020.2968184

58. Yani, Muhamad & Irawan, S, & Setianingsih, Casi. (2019). Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry’s Nail. Journal of Physics: Conference Series. 1201. 012052. 10.1088/1742-6596/1201/1/012052

59 Learning Tensorflow: A Guide to Building Deep Learning Systems 1st Edition by Tom Hope (Author), Yehezkel Resheff (Author), Itay Lieder (Author)

60 D. E. Rumelhart, G. E. Hinton, R. J. Williams et al., “Learning representations by back-propagating errors,” Cognitive modeling, vol. 5, no. 3, p. 1, 1988.

61 Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, 2015.

62 Minaee, Shervin & Boykov, Yuri & Porikli, Fatih & Plaza, Antonio & Kehtarnavaz, Nasser & Terzopoulos, Demetri. (2020). Image SegmentationUsing Deep Learning: A Survey

63 Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning. 1310–1318.

64 S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

65 Naik, D., Jaidhar, C.D. A novel Multi-Layer Attention Framework for visual description prediction using bidirectional LSTM. J Big Data 9, 104 (2022). https://doi.org/10.1186/s40537-022-00664-6

66 Fang,Y.;Yang,S.;Zhao,B.; Huang, C. Cyberbullying Detection in Social Networks Using Bi-GRU with Self-Attention Mechanism. Information2021,12,171. https:// doi.org/10.3390/info12040171

67 Zhang, Huibing & Dong, Junchao. (2020). Prediction of Repeat Customers on E-Commerce Platform Based on Blockchain. Wireless Communications and Mobile Computing. 2020. 1-15. 10.1155/2020/8841437.

68 Liu, Wenyuan & Zhu, Lin & Feng, Feng & Zhang, Wei & Zhang, Qi-Jun & Lin, Qian & Liu, Gaohua. (2020). A Time Delay Neural Network Based Technique for Nonlinear Microwave Device Modeling. Micromachines. 11. 831. 10.3390/mi11090831.

69 Nurlankyzy A., Akhmediyarova A., Zhetpisbayeva A., Namazbayev T., Yskak A., Yerzhan N., Medetov B. The dependence of the effectiveness of neural networks for recognizing human voice on language (2024) Eastern-European Journal of Enterprise Technologies, 1 (9(127)), pp. 72 - 81, DOI: 10.15587/1729-4061.2024.298687

70 Bekbolat M., Nurlankyzy A., Namazbayev T., Kulakayeva A., Akhmediyarova A., Zhetpisbayeva A., Albanbay N., Turdalyuly M., Yskak A., Uristimbek G. EVALUATION OF THE EFFECTIVENESS OF THE VOICE ACTIVITY DETECTOR BASED ON VARIOUS NEURAL NETWORKS (2025) Eastern-European Journal of Enterprise Technologies, 1 (133).2025 (процентиль 45).

71 Kulakayeva, A., Tikhvinskiy, V., Nurlankyzy, A., & Namazbayev, T. (2024). Comparative analysis of the effectiveness of neural networks at different values of the SNR ratio. Scientific Journal of Astana IT University, 20, 18–30. DOI: 10.37943/20TTRV6747

72 Медетов, Б., Нурланкызы, А., Ахмедиярова, А., Жетписбаева, А. и Жексебай, Д. 2024. Сравнительный анализ эффективности нейронных сетей при низком значении отношения С/Ш. Известия НАН РК. Серия физико-математическая. 4 (дек. 2024), 163–173. DOI: 10.32014/2024.2518-1726.315

73 Медетов, Б., Нурланкызы, А., Кулакаева, А. ., Жетписбаева, А., & Намазбаев, Т. (2024). Оценка влияния языка на точность распознавания человеческого голоса с помощью искусственных нейронных сетей. Вестник КазАТК, 131(2), 456–466. https://doi.org/10.52167/1609-1817-2024-131-2-456-466

74 А.Т.Ахмедиярова, А. Нурланкызы, А.Е.Кулакаева, Б.Ж. Медетов. (2024) Анализ эффективности нейронных сетей по распознаванию человеческого голоса. Вестник АУЭС, 1(64), 37–46. https://doi.org/10.51775/2790-0886-2024-64-1-37

75 Mussakhojayeva, S., Khassanov, Y. , Varol, H.A.: KSC2: An Industrial-Scale Open-Source Kazakh Speech Corpus. In: Proceedings of the 23rd INTERSPEECH Conference: pp. 1367-1371. 2022

76 Mamyrbayev O.Z., Oralbekova D.O., Alimhan K., Nuranbayeva B.M. Hybrid end-to-end model for Kazakh speech recognition (2023) International Journal of Speech Technology, 26 (2), pp. 261 – 270 DOI: 10.1007/s10772-022-09983-8

77 Mussakhojayeva S., Dauletbek K., Yeshpanov R., Varol H.A. Multilingual Speech Recognition for Turkic Languages (2023) Information (Switzerland), 14 (2), art. no. 74. DOI: 10.3390/info14020074

78 Yeshpanov R., Khassanov Y., Varol H.A. KazNERD: Kazakh Named Entity Recognition Dataset (2022) Language Resources and Evaluation Conference, LREC 2022, pp. 417 – 426

79 Lei Xie, Zhi-Qiang Liu, “A Comparative Study of Audio Features For Audio to Vis ual Cobvers ion in MPEG-4 Compliant Facial Animation,” Proc. of ICMLC, Dalian, 13-16 Aug-2006.

80 Namrata Dave. Feature Extraction Methods LPC, PLP and MFCC In Speech Recognition. INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY. Volume 1, Issue VI, July 2013

81 Akpudo, U.E.; Hur, J.-W. A Cost-Efficient MFCC-Based Fault Detection and Isolation Technology for Electromagnetic Pumps. Electronics 2021, 10, 439.https://doi.org/10.3390/electronics10040439

82 Rupali S Chavan et al, International Journal of Computer Science and Mobile Computing Vol.2 Issue. 6, June- 2013, pg. 233-238

83 Abdulloh Salahul Haq. Speech Recognition Implementation using MFCC and DTW Algorithm for Home Automation. Proceeding of the Electrical Engineering Computer Science and Informatics Vol. 7, October 2020

84 Ling Feng. Speaker Recognition Ling Feng Kgs. Lyngby 2004 IMM-THESIS-2004-73

85 Cong-Thanh Do. End-to-End Speech Recognition with High-Frame-Rate Features Extraction. https://doi.org/10.48550/arXiv.1907.01957

86 Jason FILOS. Time-Domain Alignment of Non-Stationary Signals. Imperial College London, England, June 2007

87 Dixit A, Vidwans A, Sharma P. Improved MFCC and LPC algorithm for bundelkhandi isolated digit speech recognition. International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016. Chennai. 2016: 3755–3759.

88 Awad A, Omar H, Ahmed Y, Farghaly Y. Speech Recognition System Using MFCC and DTW. 2016; (December 2016): 4.

89 Ittichaichareon C, Suksri S, Yingthawornsuk T. Speech recognition using MFCC. PSRC - Planet Sci Res Cent Proceeding. Pattaya. 2012; (July 2012): 135–138.

90Muhammad HZ, Nasrun M, Setianingsih C, Murti MA. Speech Recogniti on for English to Indonesian Translator Using Hidden Markov Model. International Conference on Signals and Systems. Bali. 2018: 255–260.

91 Hiroshi Shimodaira and Steve Renals, Automatic Speech Recognition— ASR Lectures 2&3 17,21 January 2019, Speech Signal Analysis

92. Ржеуцкая С. Ю., Харина М. В. Междисциплинарное взаимодействие в интегрированной информационной среде обучения технического вуза //Открытое образование. – 2017. – Т.21. – №2. – С.21-28.

93. Венцель В. Д., Цорина О. А., Янчий С. В. Организация обучения и контроля знаний студентов с использованием информационных технологий: на примере технического вуза //Азимут научных исследований: педагогика и психология. – 2018. – Т7. – №1 (22). – С.50-54

94. Ciuclea C. et al. Management Structures in Distance Education in Technical Universities //Managing Innovation and Diversity in Knowledge Society Through Turbulent Time: Proceedings of the MakeLearn and TIIM Joint International Conference 2016. – ToKnowPress, 2016. – С. 847-851.

95. Kovalenko O., Konoplianyk L. Implementing blended learning at technical university: advantages and challenges //Молодий вчений. – 2019. – №4 (1). – С.61-65.

96 National Instruments [Электронный ресурс]. - 2020. - URL: https://ru.wikipedia.org/wiki/National_Instruments (дата обращения 22.10.2020).

97. Дайнеко Е.А, Ипалакова М.Т., Болатов Ж.Ж. Разработка архитектуры виртуальной физической лаборатории // Вестник Казахстанско-Британского технического университета. – 2018. – Т.16.- №. 2. – С. 14-21.

98. Minda A. A., Gillich G. R. The Role of Virtual Laboratories in Improving Students Learning Performance //Analele Universitatii'Eftimie Murgu'. – 2018. – Т. 25. – №. 1. С.43-51

99. Hernández-de-Menéndez M., Guevara A. V., Morales-Menendez R. Virtual reality laboratories: a review of experiences //International Journal on Interactive Design and Manufacturing (IJIDeM). – 2019. – Т. 13. – №. 3. – С. 947-966.

100 Jasti N. V. K., Kota S., Venkataraman P. B. An impact of simulation labs on engineering students' academic performance: a critical Investigation //Journal of Engineering, Design and Technology. – 2020.

101 Hernández-de-Menéndez M., Guevara A. V., Morales-Menendez R. Virtual reality laboratories: a review of experiences //International Journal on Interactive Design and Manufacturing (IJIDeM). – 2019. – Т. 13. – №. 3. – С. 947-966.

102 Jasti N. V. K., Kota S., Venkataraman P. B. An impact of simulation labs on engineering students' academic performance: a critical Investigation //Journal of Engineering, Design and Technology. – 2020.

Приложение А

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

СВИДЕТЕЛЬСТВО

О ВНЕСЕНИИ СВЕДЕНИЙ В ГОСУДАРСТВЕННЫЙ РЕЕСТР ПРАВ НА ОБЪЕКТЫ, ОХРАНЯЕМЫЕ АВТОРСКИМ ПРАВОМ

№ 48570 от «24» июля 2024 года

Фамилия, имя, отчество, (если оно указано в документе, удостоверяющем личность) автора (ов):
Кулакаева Айгуль Ергалиевна, Нурланкызы Айгуль, Медетов Бекболат Жаксылыкович

Вид объекта авторского права: **программа для ЭВМ**

Название объекта: **Виртуальная лабораторно-исследовательская работа "Оценка производительности нейронных сетей в задаче распознавания речевого сигнала"**

Дата создания объекта: **22.07.2024**

Құжат тұпнұсқалығын http://www.kazpatent.kz/ru сайтының "Авторлық құқық" бөлімінде тексеруге болады https://copyright.kazpatent.kz

Подлинность документа возможно проверить на сайте kazpatent.kz в разделе «Авторское право» https://copyright.kazpatent.kz

Подписано ЭЦП

Е. Оспанов

Приложение Б

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ
МИНИСТРЛІГІ
«Қ. И. СӘТБАЕВ АТЫНДАҒЫ
ҚАЗАҚ ҰЛТТЫҚ ТЕХНИКАЛЫҚ ЗЕРТТЕУ
УНИВЕРСИТЕТІ» КОММЕРЦИЯЛЫҚ ЕМЕС
АКЦИОНЕРЛІК ҚОҒАМЫ

СӘТБАЕВ
УНИВЕРСИТЕТІ

МИНИСТЕРСТВО НАУКИ И
ВЫСШЕГО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ КАЗАХСТАН
НЕКОММЕРЧЕСКОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО
«КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ К.И. САТПАЕВА»

050013, Алматы қ., Сәтбаев к-сі, 22 үй,
Тел.: 8(727) 320-40-01, факс: 8(727) 292-60-25
e-mail: info@satbayev.university

050013. г. Алматы. ул. Сатпаева. 22
Тел.: 8(727) 320-40-01,
факс: 8(727) 292-60-25
e-mail: info@satbayev.university

№ 03-04-01-18/1697
03.04.2025

АНЫҚТАМА

Нурланкызы Айгуль "Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті" КЕАҚ 217 «Ғылымды дамыту» бюджеттік бағдарламасы, 102 "ғылыми және/немесе ғылыми-техникалық қызмет субъектілерін гранттық қаржыландыру" қосалқы бағдарламасы аясындағы:

- ЖТН AP22684173 «Төмен сигнал/шу қатынасында дауыс белсенділігін анықтау үшін жоғары тиімді нейрондық желі әдісін әзірлеу» жобасының ғылыми жетекшісі лауазымында 2024 жылғы 20 маусымнан бастап қазіргі уақытқа дейін жұмыс істейтінін растайды.

Анықтама талап етілетін орынға ұсыну үшін берілді.

Басқарма мүшесі – Ғылым және корпоративтік даму жөніндегі проректор **Е. Көлдеев**

Издатель ЭЦП - ҰЛТТЫҚ КУӘЛАНДЫРУШЫ ОРТАЛЫҚ (GOST) 2022, КУЛЬДЕЕВ ЕРЖАН, Некоммерческое акционерное общество "Казахский национальный исследовательский технический университет имени К.И. Сатпаева", BIN150140008602

Приложение В

"Халықаралық ақпараттық технологиялар университеті" АҚ

АО "Международный университет информационных технологий"

JSC "International Information Technology University"

050040, Алматы к., Манас к-сі, 34/1
Тел.: +7-727 3200001, факс: +7-727 2445121
E-mail: reception@iitu.edu.kz

34/1 Manas Str., Almaty, 050040
Tel.: +7-727 3200001, Fax: +7-727 2445121
E-mail: reception@iitu.edu.kz

050040, г. Алматы, ул. Манаса, 34/1
Тел.: +7-727 3200001, факс: +7-727 2445121
E-mail: reception@iitu.edu.kz

Исх № 1379
24.12.2024

УТВЕРЖДАЮ
Проректор по академической деятельности
А.К. Мустафина
«24» 12 2024г.

АКТ
об использовании результатов,
полученных при выполнении диссертации PhD
Нурланкызы Айгуль
докторанта Казахского национального исследовательского технического университета им. К.И. Сатпаева по специальности
«6D071900 – Радиотехника, электроника и телекоммуникации»

Комиссия в составе:
Председатель: зав. кафедрой «Радиотехника, электроника и телекоммуникации» к.т.н., ассоц. проф. Бахтиярова Е.А.
Члены комиссии:
Декан факультета «Компьютерные технологии и кибербезопасность» к.т.н., ассоц. профессор Сейлова Н.А. и ППС кафедры составили настоящий акт о том, что в 2024-2025 учебном году на кафедре «Радиотехника, электроника и телекоммуникации» внедрены основные положения диссертационного исследования докторанта Нурланкызы Айгуль на тему «Оценка производительности нейронных сетей в задаче распознавания речевого сигнала». Результаты диссертационной работы используются при проведении лабораторных работ по дисицплине «Системы мобильной связи» в рамках образовательной программы «6B06201 – Телекоммуникационные системы и сети» по направлению подготовки 6B062 – Телекоммуникации.

Председатель комиссии: Е.А. Бахтиярова
Члены комиссии: Н.А. Сейлова
А.З. Айтмагамбетов
Л.Б. Илипбаева
А.Е. Кулакаева
А.К. Оразымбетова
Б.А. Кожахметова

001415

Приложение Г

Листинг программы для извлечения MFCC признаков

```
import librosa        # Библиотека для анализа и обработки аудио
import os             # Для работы с файловой системой
import numpy as np    # Для работы с массивами и численными вычислениями

#Настройки извлечения признаков
n_mfcc = 25           # Количество коэффициентов MFCC
n_mels = 25           # Количество фильтров в мел-шкале
n_fft = 16 * 20       # Размер окна FFT (20 мс при 16 кГц = 320 сэмплов)
hop_length = 16 * 5   # Шаг окна (5 мс при 16 кГц = 80 сэмплов)
n_t = 24              # Количество временных шагов в каждом окне

# Обработка аудиофайлов с разными уровнями шума от -10 дБ до +30 дБ с шагом 2 дБ
for j in range(-10, 32, 2):
    folder_path = './wrd_data_' + str(j) + 'dB'  # Путь к папке с аудиофайлами

    # Список всех .wav файлов в папке
    file_list = [f for f in os.listdir(folder_path) if f.endswith(".wav")]

    mfcc_list = []  # Список для хранения всех окон MFCC

    # Обрабатываем каждый аудиофайл
    for file in file_list:
        file = folder_path + '/' + file  # Полный путь к файлу

        # Загружаем аудиофайл
        y, sr = librosa.load(file, sr=None)  # y — сигнал, sr — частота дискретизации

        # Извлекаем MFCC
        mfcc = librosa.feature.mfcc(
            y=y,
            sr=sr,
            n_mfcc=n_mfcc,
            n_mels=n_mels,
            n_fft=n_fft,
            hop_length=hop_length,
            center=False
        )

        # Удаляем 0-й коэффициент MFCC
```

```
        mfcc = mfcc[1:]  # Теперь форма (24, T), где T — количество временных кадров

        # Проверка, что аудиофайл достаточно длинный для формирования окон
        if mfcc.shape[1] >= n_t:
            # Используем скользящее окно по временной оси
            windows = np.lib.stride_tricks.sliding_window_view(mfcc, (n_t,), axis=1)
            windows = windows.transpose(1, 0, 2)  # Перестановка осей → (число окон, 24 MFCC, 24 временных шага)
            mfcc_list.append(windows)  # Добавляем окна из текущего файла в общий список

    # Объединяем MFCC из всех файлов в один массив
    mfcc_combined = np.concatenate(mfcc_list, axis=0)  # Финальная форма: (кол-во окон, 24, 24)

    # Сохраняем итоговый массив в файл формата NumPy
    np.save("mfcc_data_" + str(j) + "dB.npy", mfcc_combined)

    # Выводим размерность массива для контроля
    print(str(j) + "dB MFCC:", mfcc_combined.shape)
```

Листинг программы для обучение нейронной сети CNN + BiGRU

```
# Импорт необходимых библиотек
import tensorflow as tf
from tensorflow.keras.utils import to_categorical
from tensorflow.keras import layers, models
from tensorflow.keras.optimizers import Adam
import numpy as np
from sklearn.model_selection import train_test_split
import gc

# Загрузка MFCC-признаков и соответствующих меток
noise_array = np.load("./big_data/mfcc_data_0.npy")          # Класс 0: только шум
noise_label = np.zeros(noise_array.shape[0])                 # Метки для шума: 0

signal_array = np.load("./big_data/mfcc_data_1.npy")         # Класс 1: сигнал без шума
signal_label = np.ones(signal_array.shape[0])                # Метки для сигнала: 1

# Загрузка сигналов с разными уровнями SNR (дБ)
signal_array_1 = np.load("./big_data/mfcc_data_-18dB.npy")
signal_label_1 = np.ones(signal_array_1.shape[0])
```

```
signal_array_2 = np.load("./big_data/mfcc_data_-15dB.npy")
signal_label_2 = np.ones(signal_array_2.shape[0])

signal_array_3 = np.load("./big_data/mfcc_data_-12dB.npy")
signal_label_3 = np.ones(signal_array_3.shape[0])

signal_array_4 = np.load("./big_data/mfcc_data_-9dB.npy")
signal_label_4 = np.ones(signal_array_4.shape[0])

signal_array_5 = np.load("./big_data/mfcc_data_-6dB.npy")
signal_label_5 = np.ones(signal_array_5.shape[0])

signal_array_6 = np.load("./big_data/mfcc_data_-3dB.npy")
signal_label_6 = np.ones(signal_array_6.shape[0])

signal_array_7 = np.load("./big_data/mfcc_data_0dB.npy")
signal_label_7 = np.ones(signal_array_7.shape[0])

signal_array_8 = np.load("./big_data/mfcc_data_-3dB.npy")
signal_label_8 = np.ones(signal_array_8.shape[0])

signal_array_9 = np.load("./big_data/mfcc_data_6dB.npy")
signal_label_9 = np.ones(signal_array_9.shape[0])

signal_array_10 = np.load("./big_data/mfcc_data_9dB.npy")
signal_label_10 = np.ones(signal_array_10.shape[0])

signal_array_11 = np.load("./big_data/mfcc_data_12dB.npy")
signal_label_11 = np.ones(signal_array_11.shape[0])

signal_array_12 = np.load("./big_data/mfcc_data_15dB.npy")
signal_label_12 = np.ones(signal_array_12.shape[0])

signal_array_13 = np.load("./big_data/mfcc_data_18dB.npy")
signal_label_13 = np.ones(signal_array_13.shape[0])

# Объединение всех массивов и меток в один тренировочный датасет
X = np.concatenate([
    noise_array, signal_array, signal_array_1, signal_array_2,
    signal_array_3, signal_array_4, signal_array_5, signal_array_6,
    signal_array_7, signal_array_8, signal_array_9, signal_array_10,
    signal_array_11, signal_array_12, signal_array_13
], axis=0)

y = np.concatenate([
```

```
    noise_label, signal_label, signal_label_1, signal_label_2,
    signal_label_3, signal_label_4, signal_label_5, signal_label_6,
    signal_label_7, signal_label_8, signal_label_9, signal_label_10,
    signal_label_11, signal_label_12, signal_label_13
], axis=0)

#  Очистка памяти: удаление ненужных переменных
del noise_array, noise_label, signal_array, signal_label
del signal_array_1, signal_label_1, signal_array_2, signal_label_2
del signal_array_3, signal_label_3, signal_array_4, signal_label_4
del signal_array_5, signal_label_5, signal_array_6, signal_label_6
del signal_array_7, signal_label_7, signal_array_8, signal_label_8
del signal_array_9, signal_label_9, signal_array_10, signal_label_10
del signal_array_11, signal_label_11, signal_array_12, signal_label_12
del signal_array_13, signal_label_13
gc.collect()

# Преобразование меток в one-hot формат
y = to_categorical(y, num_classes=2)  # 0 -> [1, 0], 1 -> [0, 1]

# Перемешивание данных
indices = np.random.permutation(X.shape[0])
X_shuffled = X[indices]
y_shuffled = y[indices]

# Очистка
del X, y
gc.collect()

# Разделение на обучающую и тестовую выборки (80/20)
X_train, X_test, y_train, y_test = train_test_split(
    X_shuffled, y_shuffled, test_size=0.2, random_state=42
)

# Очистка
del X_shuffled, y_shuffled
gc.collect()

# Проверка размеров
print("X_train:", X_train.shape)
print("X_test:", X_test.shape)
print("y_train:", y_train.shape)
print("y_test:", y_test.shape)

# Построение модели CNN + BiGRU
```

```
def build_model(input_shape):
    model = models.Sequential()

    # Вход и разворот в 4D (для Conv2D)
    model.add(layers.Reshape((input_shape[0], input_shape[1], 1),
input_shape=input_shape))

    # Сверточные слои (CNN)
    model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'))
    model.add(layers.MaxPooling2D((2, 2)))

    # Преобразование перед BiGRU
    model.add(layers.Reshape((-1, 16)))  # (timesteps, features)

    # Bi-GRU слои (двунаправленные)
    model.add(layers.Bidirectional(layers.GRU(16, return_sequences=True)))
    model.add(layers.Bidirectional(layers.GRU(16)))

    # Полносвязные выходные слои
    model.add(layers.Dense(16, activation='relu'))
    model.add(layers.Dense(2, activation='softmax'))  # Два класса: шум / сигнал

    # Компиляция модели
    model.compile(
        loss='categorical_crossentropy',
        optimizer=Adam(learning_rate=0.001),
        metrics=['accuracy']
    )

    return model

# Задание формы входа (например, 24 MFCC на 24 фрейма)
input_shape = (24, 24)
model = build_model(input_shape)
model.summary()

# Обучение модели
history = model.fit(
    X_train, y_train,
    epochs=10,
    batch_size=1024,
    validation_data=(X_test, y_test),
```

```
    verbose=2
)

# Сохранение модели
model.save("./cnn_bigru/cnn_bigru_m10.h5")
```

Листинг программы построения модели CNN + GRU

```
# Построение модели CNN + GRU

def build_model(input_shape):
    model = models.Sequential()

    # Вход и разворот в 4D (для Conv2D)
    model.add(layers.Reshape((input_shape[0], input_shape[1], 1), input_shape=input_shape))

    # Сверточные слои (CNN)
    model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'))
    model.add(layers.MaxPooling2D((2, 2)))

    # Преобразование перед GRU
    model.add(layers.Reshape((-1, 16)))  # (timesteps, features)

    # GRU слои
    model.add(layers.GRU(16, return_sequences=True))
    model.add(layers.GRU(16))

    # Полносвязные выходные слои
    model.add(layers.Dense(16, activation='relu'))
    model.add(layers.Dense(2, activation='softmax'))  # Два класса: шум / сигнал

    # Компиляция модели
    model.compile(
        loss='categorical_crossentropy',
        optimizer=Adam(learning_rate=0.001),
        metrics=['accuracy']
    )

    return model
```

Листинг программы построения модели CNN + BiLSTM

```
# Построение модели CNN + BiLSTM

def build_model(input_shape):
    model = models.Sequential()

    # Вход и разворот в 4D (для Conv2D)
    model.add(layers.Reshape((input_shape[0], input_shape[1], 1), input_shape=input_shape))

    # Сверточные слои (CNN)
    model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'))
    model.add(layers.MaxPooling2D((2, 2)))

    # Преобразование перед BiLSTM
    model.add(layers.Reshape((-1, 16)))  # (timesteps, features)

    # Bi-LSTM слои (двунаправленные)
    model.add(layers.Bidirectional(layers.LSTM(16, return_sequences=True)))
    model.add(layers.Bidirectional(layers.LSTM(16)))

    # Полносвязные выходные слои
    model.add(layers.Dense(16, activation='relu'))
    model.add(layers.Dense(2, activation='softmax'))  # Два класса: шум / сигнал

    # Компиляция модели
    model.compile(
        loss='categorical_crossentropy',
        optimizer=Adam(learning_rate=0.001),
        metrics=['accuracy']
    )

    return model
```

Листинг программы построения модели CNN + LSTM

```
# Построение модели CNN + LSTM

def build_model(input_shape):
    model = models.Sequential()

    # Вход и разворот в 4D (для Conv2D)
```

```
    model.add(layers.Reshape((input_shape[0], input_shape[1], 1), input_shape=input_shape))

    # Сверточные слои (CNN)
    model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'))
    model.add(layers.MaxPooling2D((2, 2)))

    # Преобразование перед LSTM
    model.add(layers.Reshape((-1, 16)))  # (timesteps, features)

    # LSTM слои
    model.add(layers.LSTM(16, return_sequences=True))
    model.add(layers.LSTM(16))

    # Полносвязные выходные слои
    model.add(layers.Dense(16, activation='relu'))
    model.add(layers.Dense(2, activation='softmax'))  # Два класса: шум / сигнал

    # Компиляция модели
    model.compile(
        loss='categorical_crossentropy',
        optimizer=Adam(learning_rate=0.001),
        metrics=['accuracy']
    )

    return model
```

Листинг программы построения модели CNN + TDNN

```
# Построение модели CNN + TDNN
def build_model(input_shape):
    model = models.Sequential()

    # Вход и разворот в 4D (для Conv2D)
    model.add(layers.Reshape((input_shape[0], input_shape[1], 1), input_shape=input_shape))

    # Сверточные слои (CNN)
    model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'))
    model.add(layers.MaxPooling2D((2, 2)))
```

```
    # Преобразование перед TDNN
    model.add(layers.Reshape((-1, 16)))  # Формат (timesteps, features)

    # TDNN = Conv1D с разными ядрами
    model.add(layers.Conv1D(16, kernel_size=5, dilation_rate=1, activation='relu',
padding='same'))
    model.add(layers.Conv1D(16, kernel_size=3, dilation_rate=2, activation='relu',
padding='same'))
    model.add(layers.Conv1D(16, kernel_size=3, dilation_rate=3, activation='relu',
padding='same'))

    # Глобальный усредненный пулинг (объединение информации)
    model.add(layers.GlobalAveragePooling1D())

    # Выходной слой (2 класса)
    model.add(layers.Dense(16, activation='relu'))
    model.add(layers.Dense(2, activation='softmax'))  # 2 класса

    # Компиляция модели
    model.compile(loss='categorical_crossentropy',
optimizer=Adam(learning_rate=0.001), metrics=['accuracy'])

    return model
```

Листинг программы для тестирования нейронных сетей

```
# Импорт необходимых библиотек
import numpy as np
import tensorflow as tf
from sklearn.metrics import accuracy_score, confusion_matrix
import os
import gc

# Функция для сохранения точности при каждом SNR
def save_snr_accuracy(filename, snr, accuracy):
    mode = "a" if os.path.exists(filename) else "w"
    with open(filename, mode) as file:
        file.write(f"{snr} {accuracy:.4f}\n")

# Функция для сохранения матрицы ошибок
def save_confusion_matrix(filename, snr, matrix):
    mode = "a" if os.path.exists(filename) else "w"
    with open(filename, mode) as file:
        file.write(f"SNR: {snr} dB\n")
        file.write("Confusion Matrix:\n")
        for row in matrix:
```

```
            file.write(" ".join(str(x) for x in row) + "\n")
        file.write("\n")

# Список моделей
model_names = ['cnn_bilstm', 'cnn_bigru', 'cnn_lstm', 'cnn_gru', 'cnn_tdnn']

# Уровни SNR для тестирования
snr_values = range(-10, 32, 2)

# Основной цикл
for name in model_names:
    print(f"=== Testing model: {name} ===")

    for snr in snr_values:
        # Загрузка данных
        if snr < 0:
            signal_path = f"mfcc_data_m{abs(snr)}dB.npy"
        else:
            signal_path = f"mfcc_data_{snr}dB.npy"

        try:
            signal_array = np.load(signal_path)
            noise_array = np.load("mfcc_data_00.npy")
        except FileNotFoundError as e:
            print(f"File not found: {e}")
            continue

        # Метки: 0 - шум, 1 - сигнал
        noise_label = np.zeros(noise_array.shape[0])
        signal_label = np.ones(signal_array.shape[0])

        # Объединение данных и меток
        X_test = np.concatenate([noise_array, signal_array], axis=0)
        y_test = np.concatenate([noise_label, signal_label], axis=0)

        # Добавляем размерность канала
        X_test = np.expand_dims(X_test, axis=-1)

        # Загрузка модели
        model_path = f"./{name}/{name}.h5"
        if not os.path.exists(model_path):
            print(f"Model file not found: {model_path}")
            continue

        model = tf.keras.models.load_model(model_path)
```

```
# Предсказание
y_pred_proba = model.predict(X_test, batch_size=1024)
y_pred = np.argmax(y_pred_proba, axis=1)

# Точность
accuracy = accuracy_score(y_test, y_pred)
print(f"SNR {snr} dB - Accuracy: {accuracy:.4f}")

# Матрица ошибок
cm = confusion_matrix(y_test, y_pred)

# Сохранение результатов
accuracy_file = f"./{name}.txt"
cm_file = f"./{name}_confusion_matrix.txt"

save_snr_accuracy(accuracy_file, snr, accuracy)
save_confusion_matrix(cm_file, snr, cm)

# Очистка памяти
del X_test, y_test, y_pred, y_pred_proba, model
gc.collect()
```