МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАХЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский ядерный университет «МИФИ»»
(НИЯУ МИФИ)

ОТЧЁТ О ПОДГОТОВКЕ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Студент группы Б16-503: _____ / Смирнов А.С /

Руководитель: _____ / Максутов А.А /

Москва 2020 г.

СПИСОК ИСПОЛНИТЕЛЕЙ

Руководитель НИР,
Ассистент кафедры
«Компьютерные системы
и технологии» НИЯУ МИФИ _____ А. А. Максутов

подпись, дата

Исполнители:

Студент НИЯУ МИФИ _____ А. С. Смирнов

подпись дата

РЕФЕРАТ

Отчёт XX с., XX рис., XX табл., XX источн. DOLPHIN ATTACK, LIGHT COMMANDS, SPEECH RECOGNITION, PYTHON, HIDDEN VOICE COMMANDS, …

Объектом исследования является системы голосового управления.

Цель работы – создание способа защиты от нежелательных воздействий на систему голосового управления.

В процессе работы проводился анализ наиболее распространённых систем голосового управления. Были изучены основные атаки на данные технологию. Также, проводилась проектирование и реализация голосового управления. Был разработанный способ распознавания и фильтрация правильных инструкций из получаемых команд. С помощью синтезированных скрытых команд была протестирована атака и способ их распознавания.

В результате (Дополнить)

СОДЕРЖАНИЕ

Оглавление

ВСТУПЛЕНИЕ

В современном мире системы распознавания речи становятся всё более популярным методом взаимодействия человека с компьютером. Поэтому стали создаваться голосовые пользовательские интерфейсы, способные переводить речь в текст для понимания голосовых команд, обрабатывать естественный язык, переводить текст в речь для ответа на вопросы или показания работы системы, такие интерфейсы сейчас используют: в управлении автомобилем, системы домашней авторизации, компьютерные операционные системы, бытовые приборы, являются основным способом взаимодействия с виртуальным помощником на смартфонах или интеллектуальных колонках. Голосовые интерфейсы могут принимать и выполнять самые различные команды: поиск, по ключевым словам, управление различным устройством, голосовой набор, маршрутизация вызовов, авторизация, переводы денег, посещение различных сайтов.

С каждым годом в мире всё больше устройств использует данную технологию, выполняемые команды становятся всё сложнее и сложнее, очевидно, что в основе данной технологии не должно быть уязвимостей, иначе дальнейшая разработка будет не востребована.

1 ОБЗОРНАЯ ЧАСТЬ

1.1 Голосовое управление

Голосовое управление (или «голосовой интерфейс пользователя») – это система, позволяющая с помощью распознавания речи передавать команды управления на устройство с помощью голоса. Голосовые интерфейсы к компьютерным системам становятся всё популярнее, в основном из-за простоты в их использовании и благодаря уменьшению современных мобильных устройств, которые затрудняют физическое взаимодействие [1]. Многие мобильные устройства используют постоянно включённую модель, в которой идёт постоянно прослушивается возможный голосовой ввод. Хотя и голосовые команды обеспечивают повышенную доступность и потенциально более легкое взаимодействие человека и компьютера, они в тоже время подвержены внешним атакам, так как голос это широковещательный канал, открытый для любого злоумышленника, способного создавать звук в непосредственной близости от устройства. Это дает злоумышленникам возможность попытаться выдать не авторизованные голосовые команды этим устройствам.

Рассмотрим немного истории. В 2007 году такие компании как Google и Apple одни из первых задумались над созданием собственной технологией голосового управления. Прошли годы и с тех пор мир стал свидетелем множества подобных устройств. Google создала механизм распознавания речи под названием Pico TTS, а Apple создала и представила миру Siri. С тех пор каждый год на рынок выходит новое устройство способное считывать ваш голос и выполнять команды.

1.2 Распознавание речи

Распознавание речи – это подразделение компьютерной лингвистики, которое разрабатывает методологии и технологии, позволяющие

распознавать и переводить разговорный язык в текст с помощью компьютеров. Будем ориентироваться исключительно на независимые от говорящего системы распознавания речи, то есть системы для интерпретации речи любого говорящего. Это наиболее распространённые системы на смартфонах, планшетах, пк, а значит могут быть целью для злоумышленника.

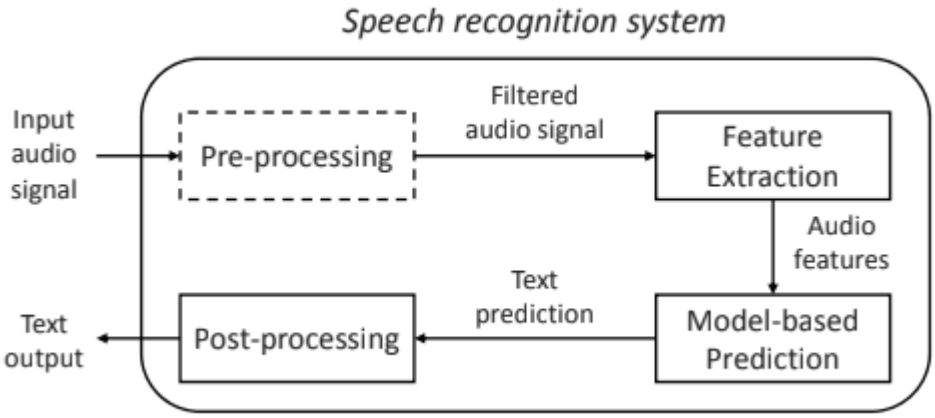

Рисунок X - принцип работы системы распознавания речи.

На рисунке X представлен процесс распознавания речи. Этот процесс состоит из четырёх этапов: предварительная обработка (pre-processing), извлечение признаков (Feature Extraction), прогнозирование (Model-based Prediction), пост-обработка (Post-processing).

На первом этапе выполняется начальная речевая/не речевая идентификация путём фильтрации частот, выходящих за пределы диапазона человеческого голоса, удаление фоновых шумов, устранение периодов времени, когда энергия падает ниже определённого порога. Этот шаг прост и выполняет элементарную фильтрацию, но именно на этом этапе, происходит отбор всех не речевых команд.

На втором шаге происходит извлечение признаков из входного сообщения. Сигнал разбивается на короткие кадры (обычно около 20мс) и извлекает объекты из каждого кадра. Алгоритм извлечения признаков, используемый в распознавании речи, почти всегда является преобразованием Мел-кепстральные коэффициенты (MFCC) [5, 6]. Но на высоком уровне его

можно рассматривать как преобразование, которое извлекает доминирующие частоты из входных данных.

Третий этап принимает извлеченные на предыдущем этапе объекты и сопоставляет их с существующей моделью, созданной в автономном режиме для создания текстовых прогнозов. Технология на этом этапе почти всегда отличается, кто-то использует скрытые марковские модели, но многие современные системы начали использовать рекуррентные нейронные сети (RNN).

Последний этап пост обработки ранжирует текст используя дополнительных источников информации. Обычно на этом этапе проверяется грамматика и значение всех слов.

1.3 Голосовой помощник

Одним из самых распространённых примеров голосового интерфейса служит виртуальный (или голосовой) помощник.
Голосовой помощник – сервис на основе искусственного интеллекта, распознающий человеческую речь и способный выполнить определенное действие в ответ на голосовую команду. Чаще всего голосовые помощники используются в смартфонах, умных колонках, веб-браузерах.
Функционал голосовых помощников достаточно разнообразен. Они умеют:

- Вести диалоги
- Предлагать быстрые ответы на вопросы пользователя
- Поиск нужной информации в интернете
- Вызывать такси
- Совершать звонки
- Прокладывать маршруты
- Делать заказы в интернет-магазине
- Взаимодействовать с техникой (дверьми, колонками, компьютером, электронным замком)

Также все голосовые помощники обладают искусственным интеллектом, который при общении с пользователем может учитывать и запоминать:

- Его местоположение
- Время суток и дни недели
- Историю поисковых запросов
- Предыдущие заказы в интернет-магазине

1.3.1 История развития голосовых помощников.

История голосовых ассистентов начинается с конца 1930-х годов, когда ученые начали предпринимать попытки распознать голос силами технологий. Тогда созданию качественного помощника мешали две большие проблемы:

- существование омонимов — слов с одинаковым звучанием, но с разным значением.
- постоянный шумовой фон, из которого система должна выбирать речь пользователя.

Сейчас для решения этих проблем разработчики используют машинное обучение. Оно учит нейронные сети самостоятельно анализировать контекст и эффективно определять основной источник звука. Однако пришли разработчики к этому не сразу — потребовалось как минимум 80 лет подготовительных работ:

1939 год. Советский физик Лев Мясников создал аппарат, способный распознавать человеческую речь — несколько гласных и согласных звуков. **1952 год.** Сотрудники лаборатории Bell разработали механизм, который распознавал продиктованные по телефону числа от 1 до 9. **1962 год.** Компания IBM представила собственную технологию распознавания речи — Shoebox. Машина распознавала 16 английских слов, 10 цифр и 6 арифметических команд. **1980 год.** Инженеры научились применять методы «Скрытой модели

Маркова». Со временем это позволило голосовым системам лучше распознавать речь. Они обрабатывают слово, учитывая несколько предыдущих и предсказывая, что может с ними сочетаться.
1987 год. В США компания Worlds of Wonder начала продавать говорящую куклу Джулию, которая училась распознавать речь ребенка во время игры. В куклу был встроен процессор, который позволял ей реагировать и генерировать речь. Джули воспринимала восемь высказываний: «Джули», «да», «нет», «хорошо», «притворяйся», «голодна», «пой» и «молчи».
1990-е годы. Появилась коммерческая программа Dragon Dictate, ориентированная на массовый рынок. Она распознавала речь и записывала надиктованный текст в файл.
1996 год. Появилось полноценное голосовое меню VAoL т BellSouth. Система обрабатывала телефонные справочные запросы и помогала покупателям в поиске нужной информации об интересующих товарах.
2001 год. Компания Microsoft добавила голосовой ввод текста в офисный пакет Office XP.
2002 год. Google запустил Voice Search — сервис для голосового поиска в интернете. Проект приостановили из-за неудобства использования — чтобы выполнить поиск, надо было позвонить на специальный номер. На Voice Search основан современный интерактивный помощник компании — Google Assistant.
2007 год. Центр исследования искусственного интеллекта SRI International начал разработку Siri. Siri стала первой голосовой помощницей — система умела не только искать информацию в интернете или работать как голосовое меню, но и вести с пользователем диалог.
2011-2014 годы. Google интегрировал функцию голосового поиска в браузер Chrome. Компания также запустила персонализированного ассистента Google Now с расширенными возможностями голосового поиска — сервис подбирал актуальную информацию с учетом местоположения пользователя, истории браузера и других поисковых запросов.

У Microsoft также появилась собственная виртуальная голосовая помощница - Cortana.

2014 год. Amazon представил первую в мире умную колонку Amazon Echo с голосовой ассистенткой Alexa.

2017 год. Alibaba представила умную колонку Tmall Genie с голосовым помощником AliGenie.

2018 год. Яндекс выпустил умную колонку Яндекс. Станция с голосовой помощницей Алисой.

2019 год. Банк «Тинькофф» запустил собственного голосового ассистента «Олега». Mail.Ru Group представила голосовую помощницу «Марусю».

1.3.2 Как устроены современные голосовые помощники.

Голосовые помощники пассивно считывают все звуковые сигналы, и для активной работы им необходима активация при помощи кодовой фразы. Например, “Окей, Google” или “Алиса”. В момент голосового запроса автоматическая система распознавания речи (ARS system) преобразует звуковой сигнал в текст. Это происходит в четыре этапа:

1) **Фильтрация.** Система убирает из звукового сигнала шумовой фон и помехи, возникающие при записи.
2) **Оцифровка.** Звуковые волны преобразуются в понятный для компьютера цифровой вид. Параметры получаемого кода в том числе определяют качество записи.
3) **Анализ.** В сигнале выделяются участки, содержащие речь. Система оценивает ее параметры — к какой части речи относится слово, в какой оно форме, насколько вероятна связь между двумя словами.
4) **Выявление шаблонов данных.** Полученную информацию система включает в словарь — собирает разные варианты произношения одного и того же слова. Чтобы точнее распознавать новые запросы, ассистенты сравнивают слова в них с шаблонами.

1.4 Обзор способов вредоносного воздействия.

Мы всё больше и больше доверяем умным колонкам нашу жизнь. Уже сегодня они могут тратить деньги с нашей карты на разные покупки в интернет-магазинах или управлять всей техникой в доме. Насколько безопасны эти разработки и всегда ли они слушаются только своего владельца? Голосовые помощники уже успели “засветиться” в нескольких примечательных историях:

- Студенты Калифорнийского университета в Беркли нашли способ запустить голосовые помощники Siri, Alexa и Google Assistant без ведома владельца. Для этого достаточно в музыку или видео добавить звуки, отдаленно напоминающие человеческую речь – программе будет этого достаточно, чтобы превратить их в слова и начать выполнять заданную команду.
- В Китае смогли активировать голосовых помощников с помощью звуковых частот, которые обычный человек не слышит.
- Burger King запустили рекламу с фразой «OK, Google, что такое Воппер?», на которую откликались голосовые ассистенты и начинали зачитывать строчки из Википедии.
- Amazon Echo заказал кукольный домик, услышав просьбу 6-летней девочки. А когда этот сюжет обсуждали в новостях, голосовые помощники, восприняв фразу о заказе домика как команду, стали повсеместно заказывать эти домики.
- Владельцы Amazon Echo жаловались, что устройство начинает самопроизвольно смеяться. Выяснилось, что устройство распознавало окружающие звуки как команду засмеяться и выполняло ее.

При этом Google и Amazon утверждают, что их помощники не включаются, если не слышат голос владельца. Apple говорит, что Siri никогда не выполнит команду, связанную с личными данными, если iPhone

или iPad заблокированы. Давайте разберём работу некоторых способов взлома наших помощников.

1.4.1 Hidden Voice Commands

Системы распознавания речи полагаются на акустические характеристики, которые он извлекает из входного аудио для генерации новой команды. Пока входное аудио сообщение содержит достаточно акустической информации система может распознать новую команду с достаточно высокой точностью.

Скрытая команда — это атака в которой злоумышленник изменяет входной аудио сигнал таким образом, что получившийся звук сохраняет достаточно акустических характеристик для системы распознавания речи, но является абсолютно непонятным для человека. На рисунке X показана схема работы этой атаки.

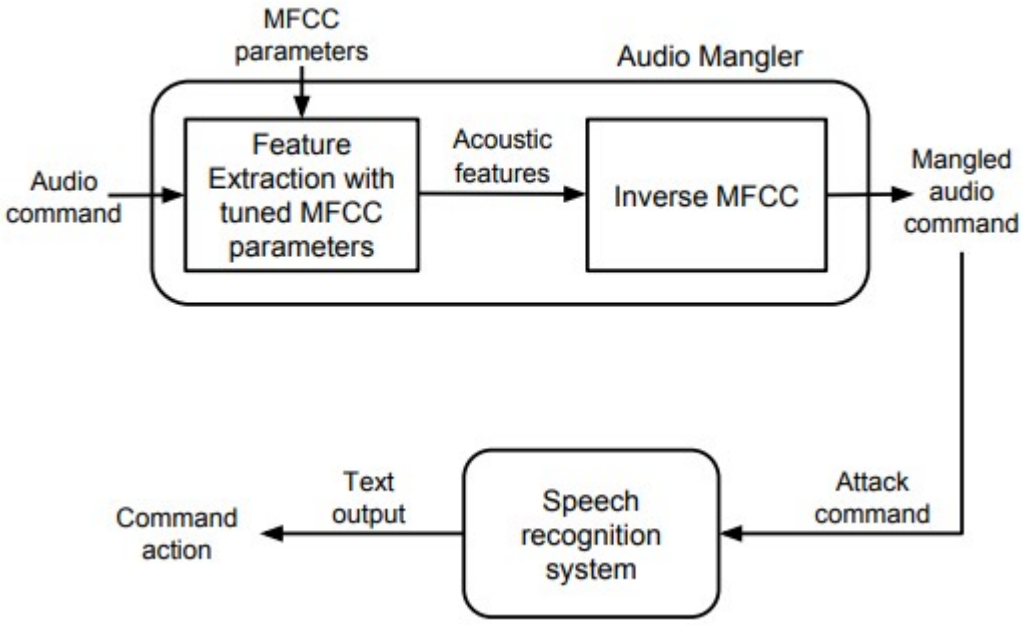

Рисунок X — схема работы атаки Hidden Voice Commands

В начале правильная голосовая команда подаётся в Audio Mangler, который создаёт изменённую версию исходной команды. Преобразованный звуковой сигнал сохраняет достаточные акустические характеристики исходной команды, но звучит совершенно иначе. После эта команда

воспроизводится рядом с устройством распознавания речи, который распознает и выполнит данную команду. Далее предлагаю подробно ознакомится с процессом искажения звука.

MFCC parameters. Расчёт MFCC требует указания различных параметров [4]. В данной атаке сфокусированы на 4 независимых параметрах: wintime, hoptime, numcep, nbands. Список изменённых MFCC параметров можно увидеть в таблице X.

Parameter	Description
wintime	time for which the signal is considered constant
hoptime	time step between adjacent windows
numcep	number of cepstral coefficients
nbands	no. of warped spectral bands for aggregating energy levels

Таблица X – список изменённых MFCC параметров.

Если правильно подобрать диапазон значений для этих параметров, то получившийся искажённый звук будет обладать минимальной, но достаточной акустической информацией для системы распознавания речи.

Извлечение функций с настроенными параметрами MFCC. После экспериментального определения параметров MFCC акустические характеристики извлекаются путём вычисления MFCC [4] входного сигнала с использованием этих параметров. Процесс считает, что сигнал является статистически постоянным в течение небольшого временного окна, а также агрегирует уровни энергии близко расположенных частот. Агрегация уровня энергии мотивируется слуховым механизмом человека, поскольку мы не может различать различия между близко расположенными частотами, и эффект становится более доминирующим при увеличении частот. Настроенные параметры MFCC, используемые для создания команды атаки, предназначены для дальнейшего увеличения этой потери информации способом, который является определяющим для человеческого понимания.

Обратный MFCC. На этом этапе мы из получившийся MFCC функции

воссоздаём новый аудио сигнал. Инверсия MFCC включает в себя добавление шума. На выходе имеем искажённый звук.

Конечно описанный процесс может занять много времени, но создание одного искажённого звука приведёт к успешной атаке.

1.4.2 Dolphin Attack

Dolphin Attack - способ контроля голосовых помощников с помощью частот, которые не различает человеческое ухо. Эту атаку представили шестеро исследователей из Чжэцзянского университета. Они хотели улучшить предыдущую атаку. По словам специалистов, преобразовать обычные голосовые команды в ультразвук не так уж трудно. В результате окружающие люди ничего не услышат, тогда как гаджеты распознают команды и начнут выполнять. На рисунке X представлена схема работы Dolphin Attack.

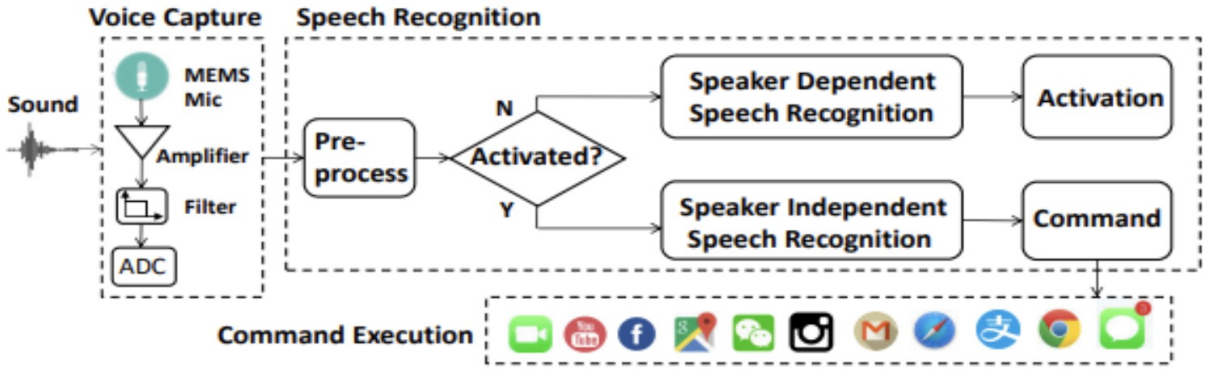

Рисунок X - схема работы Dolphin Attack

Специалисты протестировали свою идею на ассистентах Alexa, Cortana, Google Now, Huawei HiVoice, Samsung S Voice и Siri, а также 16 платформах и устройствах, использующих данное ПО (смартфоны, компьютеры, автомобили, «умные» дома). Тестирование включало в себя запуск Facetime на iPhone, проигрывание музыки на Amazon Echo и управление системой навигации в автомобилях Audi. Атаку тестировали на частоте от 25 до 39 кГц, и она показала стабильный результат на расстоянии 1,75 метра. Специалисты предупреждают, что на эффективность Dolphin Attack могут

влиять язык, на котором устройству передаются команды, а также уровень фоновых шумов. В таблице X приведена список результатов этих тестов.

Таблица X - общая таблица с результатами тестов

Manuf.	Model	OS/Ver.	SR System	Attacks		Modulation Parameters		Max Dist. (cm)	
				Recog.	Activ.	f_c (kHz) & [Prime f_c] ‡	Depth	Recog.	Activ.
Apple	iPhone 4s	iOS 9.3.5	Siri	√	√	20–42 [27.9]	≥ 9%	175	110
Apple	iPhone 5s	iOS 10.0.2	Siri	√	√	24.1 26.2 27 29.3 [24.1]	100%	7.5	10
Apple	iPhone SE	iOS 10.3.1	Siri	√	√	22–28 33 [22.6]	≥ 47%	30	25
			Chrome	√	N/A	22–26 28 [22.6]	≥ 37%	16	N/A
Apple	iPhone SE †	iOS 10.3.2	Siri	√	√	21–29 31 33 [22.4]	≥ 43%	21	24
Apple	iPhone 6s ⋆	iOS 10.2.1	Siri	√	√	26 [26]	100%	4	12
Apple	iPhone 6 Plus ⋆	iOS 10.3.1	Siri	×	√	– [24]	–	–	2
Apple	iPhone 7 Plus ⋆	iOS 10.3.1	Siri	√	√	21 24-29 [25.3]	≥ 50%	18	12
Apple	watch	watchOS 3.1	Siri	√	√	20–37 [22.3]	≥ 5%	111	164
Apple	iPad mini 4	iOS 10.2.1	Siri	√	√	22–40 [28.8]	≥ 25%	91.6	50.5
Apple	MacBook	macOS Sierra	Siri	√	N/A	20-22 24-25 27-37 39 [22.8]	≥ 76%	31	N/A
LG	Nexus 5X	Android 7.1.1	Google Now	√	√	30.7 [30.7]	100%	6	11
Asus	Nexus 7	Android 6.0.1	Google Now	√	√	24–39 [24.1]	≥ 5%	88	87
Samsung	Galaxy S6 edge	Android 6.0.1	S Voice	√	√	20–38 [28.4]	≥ 17%	36.1	56.2
Huawei	Honor 7	Android 6.0	HiVoice	√	√	29–37 [29.5]	≥ 17%	13	14
Lenovo	ThinkPad T440p	Windows 10	Cortana	√	√	23.4–29 [23.6]	≥ 35%	58	8
Amazon	Echo ⋆	5589	Alexa	√	√	20-21 23-31 33-34 [24]	≥ 20%	165	165
Audi	Q3	N/A	N/A	√	N/A	21–23 [22]	100%	10	N/A

‡ Prime f_c is the carrier wave frequency that exhibits highest baseband amplitude after demodulation. – No result
† Another iPhone SE with identical technical spec.
⋆ Experimented with the front/top microphones on devices.

При помощи Dolphin Attack можно осуществлять и более опасные действия, к примеру, заставить браузер пользователя посетить вредоносный сайт, установить на устройство вредоносное приложение, подписать жертву на платный сервис.

1.4.3 Light Commands

Light Commands - это уязвимость микрофонов MEMS, позволяющая злоумышленникам удаленно вводить неслышимые и невидимые команды в голосовые помощники. Обычный направленный лазерный луч можно использовать для передачи голосовых команд умным колонкам, к такому выводу пришли учёные из Мичиганского университета и Университета электросвязи в Токио. Как сообщают исследователи, направленное излучение

может воздействовать на электромеханические микрофоны внутри устройств.

Микрофоны преобразуют звук в электрические сигналы. Главное открытие за световыми командами состоит в том, что в дополнение к звуку микрофоны также реагируют на свет, направленный непосредственно на них. Таким образом, моделируя электрический сигнал по интенсивности светового луча, злоумышленники могут обманным путем заставить микрофоны генерировать электрические сигналы, как будто они получают подлинное аудио. На рисунке 2 представлен эксперимент управления голосовым помощником на расстоянии с помощью лазерной указки.

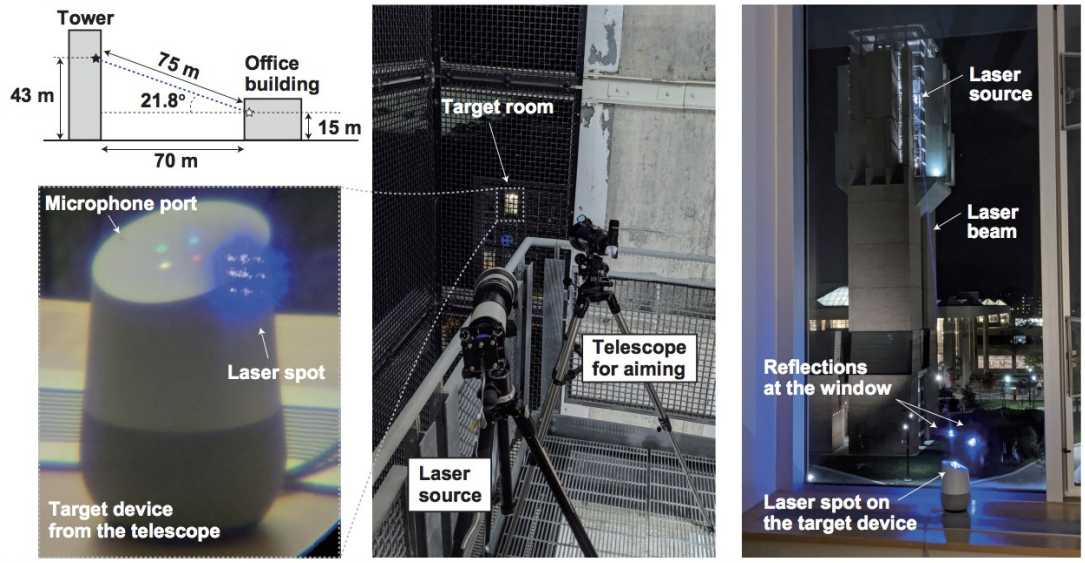

Fig. 10. Setup for the low-power cross-building attack: (Top left) Laser and target arrangement. (Bottom left) Picture of the target device as visible through the telescope, with the microphone ports and laser spot clearly visible. (Middle) Picture from the tower: laser on telephoto lens aiming down to the target. (Right) Picture from the office building: laser spot on the target device.

Рисунок 2 - Эксперимент по взлому умной колонки.

2 РАСЧЁТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ

2.1 Описание работы предлагаемого алгоритма защиты голосовых помощников.

Исходя из разобранных способов вредоносного воздействия на голосовые помощники, было решено создать систему способную противостоять атаке «hidden voice command». Эта атака нацелена на обман устройства, принимающего и обрабатывающего звуковой сигнал, на этапе прогнозирования (Model-based Prediction). На этом этапе рассматриваются полученные на этапе Feature Extraction акустические функции, нас интересуют mfcc коэффициенты, их сопоставляют с имеющимися шаблонами команд и выбрав подходящую передают эту команду на блок исполнения. Так как для правильного распознавания речи необходимо лишь первые 13 mfcc коэффициентов, в процессе генерации запутанных команд из правильной аудио записи извлекают первые 13 mfcc коэффициентов, а другие изменяют, добавляют шумы. В итоге команда может быть правильно распознана системой распознавания речи, а человек услышит лишь непонятные помехи.

Необходимо создать модель, которая проанализировав mfcc коэффициенты у скрытых команд и обычных команд, будет способна распознать попытку взлома и выдать системе распознавания речи предупреждение. Для того чтобы не замедлять работу системы распознавания речи, можно запускать эту модель параллельно с 3 этапом обработки звука, как показано на рисунке X. Пока голосовой помощник подбирает подходящий шаблон команды, блок Voice Command Classification успеет проанализировать поступившую аудио команду и выдаст вердикт, можно ли её выполнять.

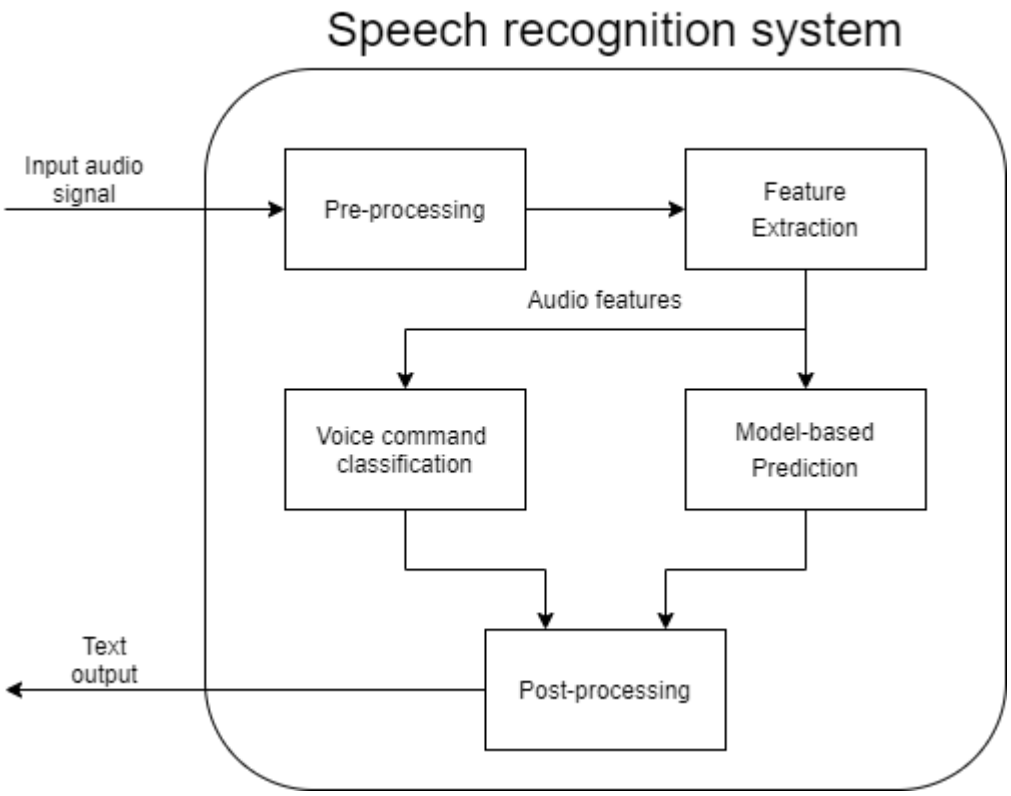

Рисунок X - Предлагаемое изменение системы распознавания голоса

2.2 Анализ работы со звуком.

Давайте разберём почему многие голосовые помощники используют Мел-кепстральные коэффициенты и как их получить. Процесс распознавания речи состоит из четырёх этапов: предварительная обработка (pre-processing), извлечение признаков (Feature Extraction), прогнозирование (Model-based Prediction), пост-обработка (Post-processing). Давайте поподробнее разберём подробнее второй этап.

На втором шаге происходит извлечение признаков из входного сообщения. Сигнал разбивается на короткие кадры (обычно около 20мс) и извлекает объекты из каждого кадра. Алгоритм извлечения признаков, используемый в распознавании речи, почти всегда является преобразованием Мел-кепстральных коэффициентов (MFCC) [5, 6]. Но на высоком уровне его можно рассматривать как преобразование, которое извлекает доминирующие частоты из входных данных.

Мел — это единица высоты звука, основанная на восприятии этого звука нашими органами слуха. Заметим что высота звука не совсем линейно зависит от частоты, что показано на рисунке X. Такую зависимость можно описать простой формулой m = 1125 * Ln(1 + f/700). Такие единицы измерения часто используют при решении задач распознавания, так как они позволяют приблизится к механизмам человеческого восприятия звука.

Теперь поговорим про второе слово в названии — кепсрт. Кепстр — один из видов гомоморфной обработки сигналов, функция обратного преобразования Фурье от логарифма спектра мощности сигнала. Кепстр можно записать следующим выражением:

$$C_s(q) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \ln |S(\omega)|^2 e^{i\omega q} \, d\omega$$

Где S(w) – спектр входного сигнала. В итоге получается набор не пересекающиеся характеристик исходного сигнала и фильтра.

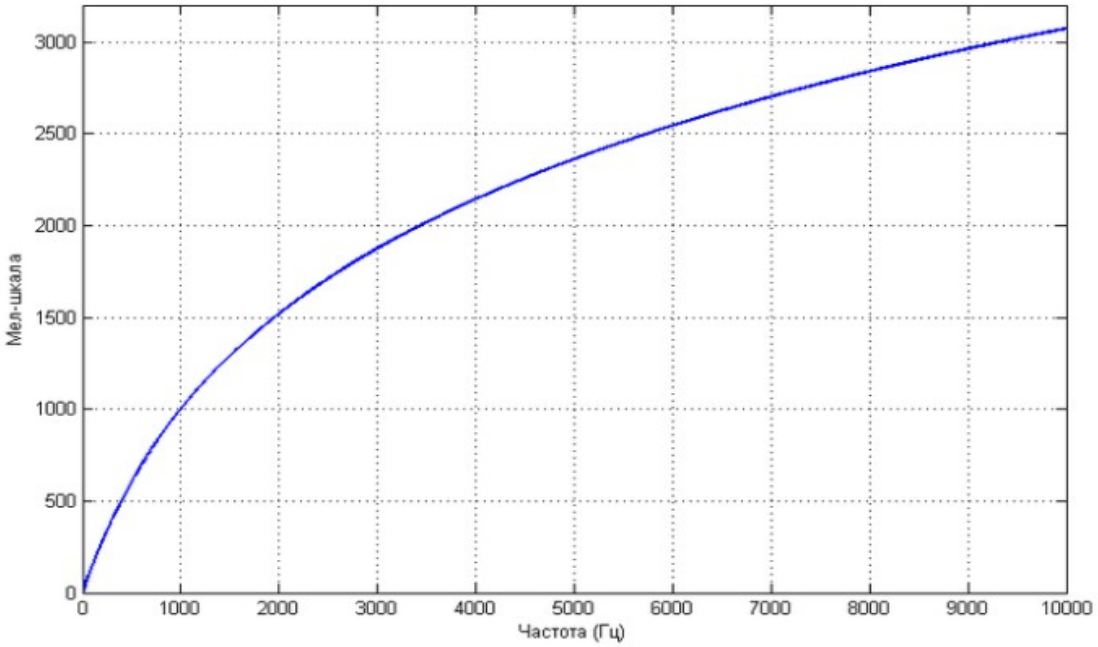

Рисунок X — Зависимость Мел коэффициентов от частоты звука.

2.2.1 Преобразование звука в набор коэффициентов MFCC.

Рассмотрим получение коэффициентов mfcc на тестовом примере. Возьмём временное представление числа «1» , которое изображено на рисунке X. Вначале, с помощью преобразования Фурье, получаем спектр исходного сигнала по всей временной оси. Теперь наш посчитанный спектр нужно разложить по мел-шкале. Для этого разбиваем спектр на маленькие окна, примерно по 20мс, равномерно расположенные на мел-оси. Переведя получившийся график в частотную шкалу увидим, что окна собираются в области низких частот, обеспечивая высокую точность распознавания речи. Чтобы найти энергию сигнала, которая попадает в каждое окно анализа, необходимо перемножить вектор спектра сигнала с оконными функциями. Так получим некий набор коэффициентов, возводим их в квадрат и логарифмируем. Нам осталось получить их них кепстральные коэффициенты, для этого можем ещё раз применить преобразование Фурье или использовать дискретное косинусное преобразование. В результате получаем последовательность показанную на рисунке X.

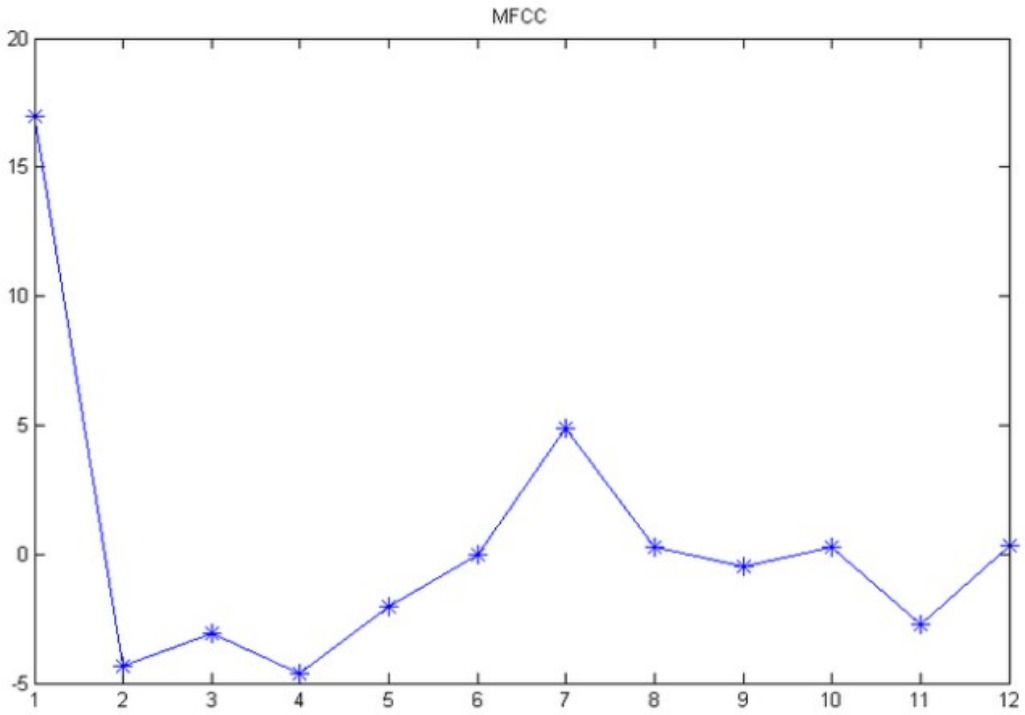

Рисунок X – последовательность коэффициентов mfcc.

Таким образом получается небольшое набор значений, который позволяют с большой точностью распознать речевой сигнал, высчитываются достаточно быстро и занимают меньше объёма, чем спектрограмма или временное представление сигнала. Проведя анализ я выявил, что для задачи распознавания слов достаточно первые 13 — 18 вычислительных коэффициентов.

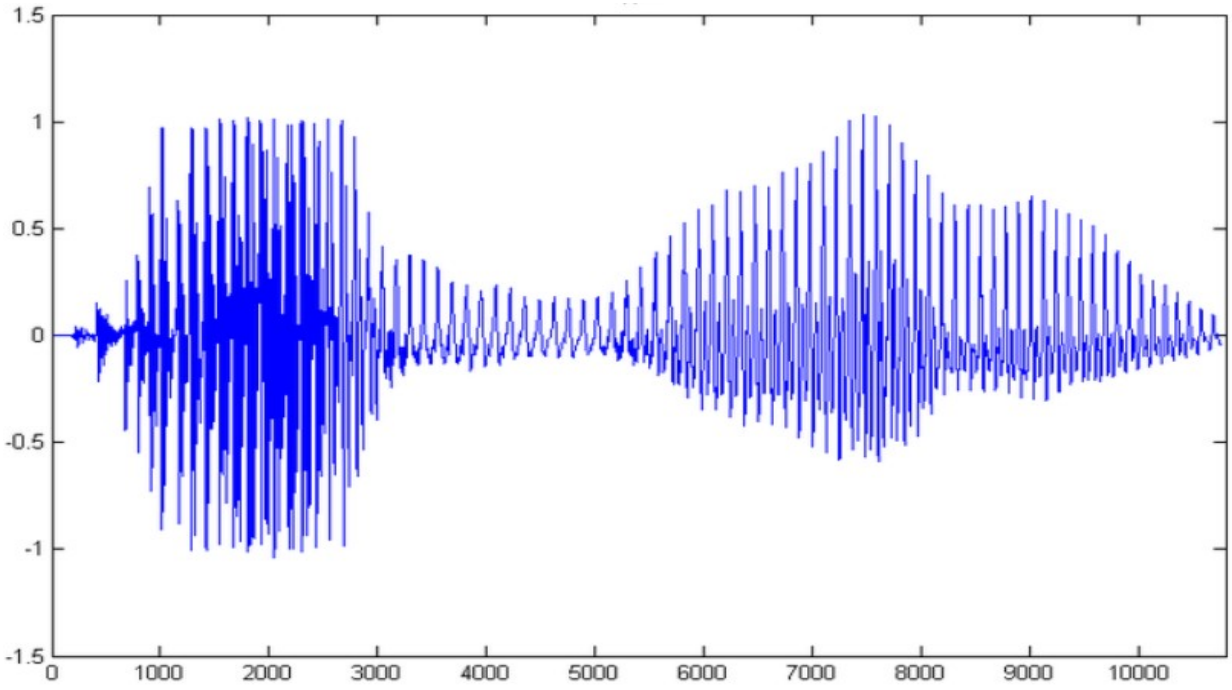

Рисунок X – временное представление цифры «1»

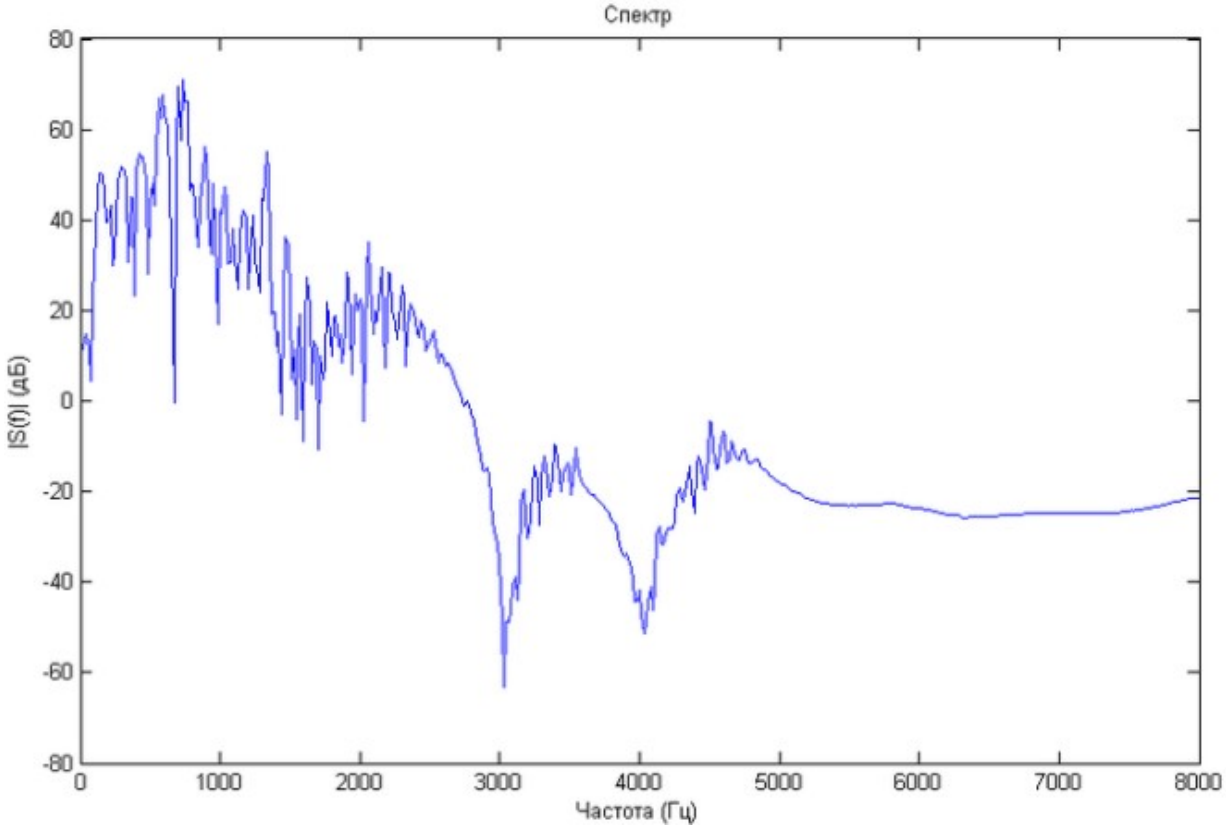

Рисунок X - спектр исходного сигнала.

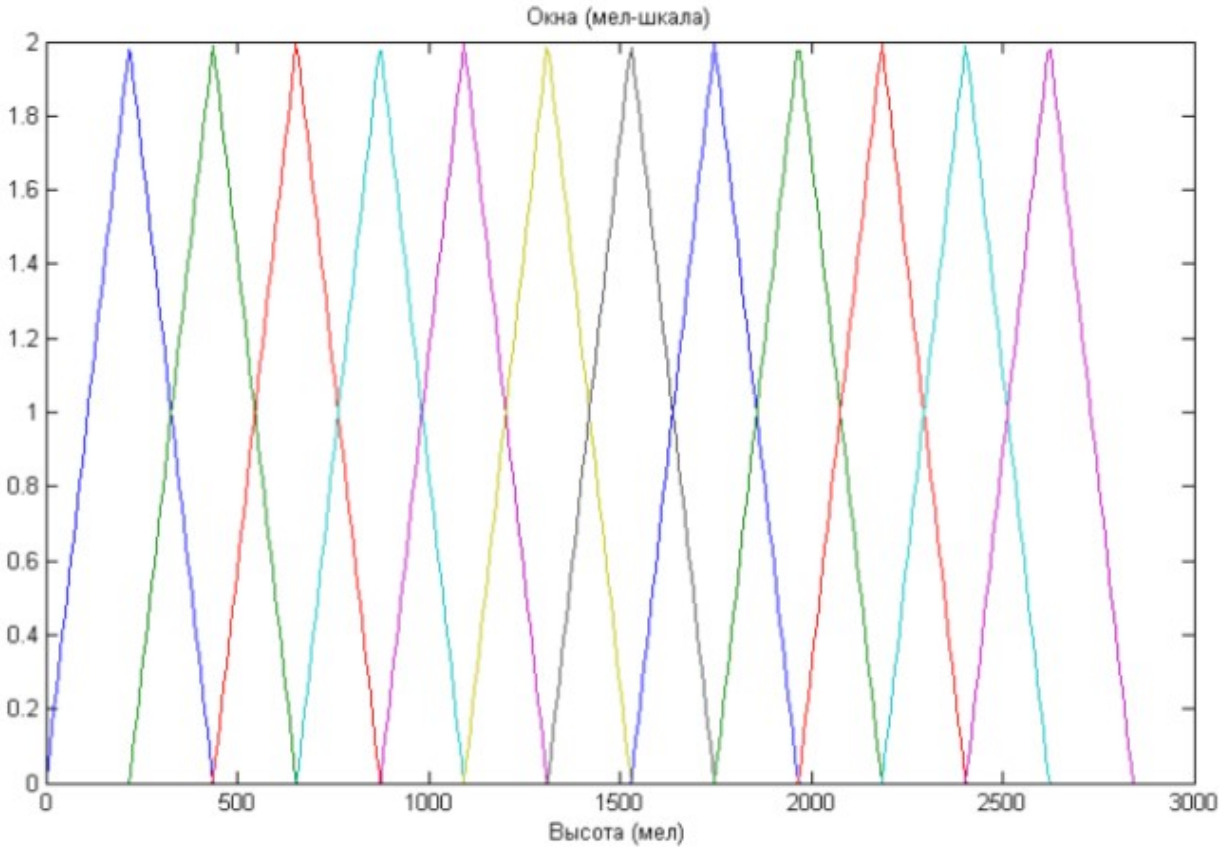

Рисунок X - Окна расположенные на мел-оси.

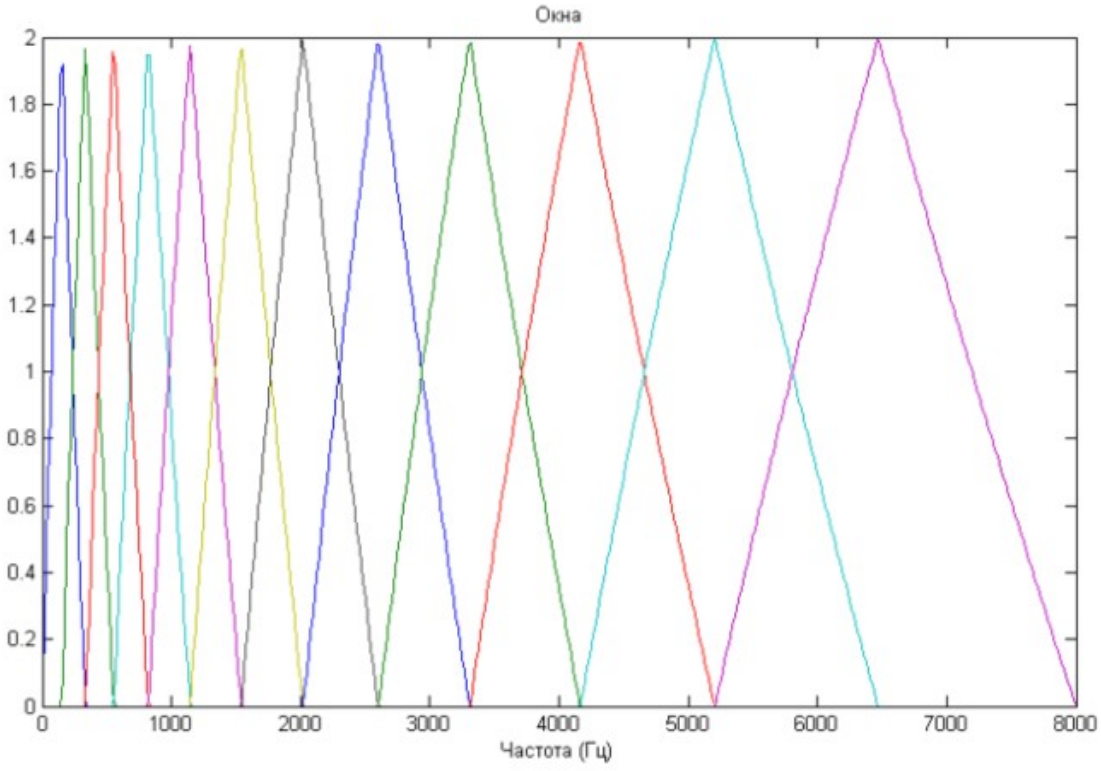

Рисунок X - мел-окна в частотной шкале.

2.3 Подготовка датасета для обучения модели.

Для обучения модели было создано множество аудио записей, как хороших, так и плохих. На таблице X показаны команды на которых будет обучаться наша модель.

Таблица X

1	OK Google.	6	Open the door please.
2	Call 911.	7	What’s my current location?
3	Open YouTube.com.	8	Open Bank of America.
4	Show facebook.com.	9	Play country music.
5	Turn on airplane mode.	10	What’s my schedule today?

Были выбраны десять самых популярных команд для взаимодействия с голосовыми помощниками. Среди них есть как безопасные, включение голосового ассистента от Google или включение музыки, так и потенциально опасные, переход на сайт или открытие двери, которое используется в системах умного дома.

Для создания хороших аудио команд я использовал сервис для преобразования текста в речь [7]. Этот сервис позволяет создать аудио записи с разной скоростью речи и с пятью разными голосами, три голоса женские и два - мужских.

Для создания множества запутанных команд был разработан программный продукт, способный преобразовывать хорошую команду в запутанную. Программа написана на языке Python. Архитектура работы программы показана на рисунке X. В качестве хороших команд используем аудио сигналы созданные выше. Для реализации шагов 1 и 2 используется

модуль Librosa [8]. Из аудио сигнала извлекаются первые последовательности мел-кепстральных коэффициентов. В момент преобразования акустических функций к записи добавляются шумы. Так получаются запутанные команды, которые с трудом могут быть распознаны человеком. На третьем шаге проверяется способен ли голосовой помощник правильно разобрать запутанную команду. Для этого используется модуль SpeechRecognition [9]. Если было установлено, что система распознавания голоса не распознала или неправильно распознала запутанную команду, то повторяем шаги 1 и 2 изменив количество извлекаемых последовательностей и время каждого кадра.

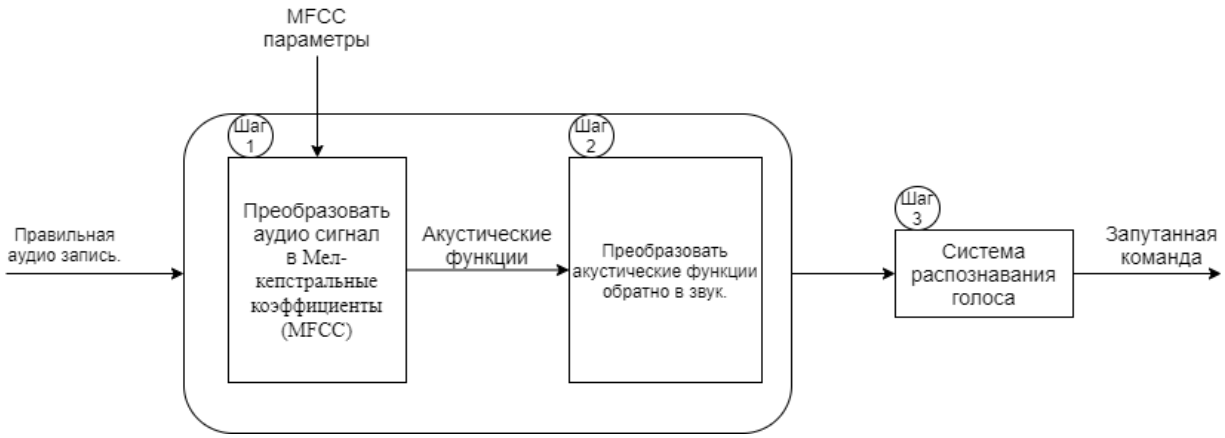

Рисунок X — схема создания запутанных команд.

2.3.1 Модуль Librosa.

Модуль Librosa – это пакет Python для анализа музыки и аудио. В программе он используется для преобразования аудио в мел-кепстральные коэффициенты, наложения шумов и преобразования обратно в аудио. На таблице X представлены используемые в этом проекте функции из модуля Librosa.

Таблица X

Librosa.core.load

<table>
<tr><td colspan="2">y, sr = librosa.core.load(path, sr=22050, mono=True, offset=0.0, duration=None, dtype=<class 'numpy.float32'>, res_type='kaiser_best')

Загрузка аудио в виде временного ряда с плавающей запятой.</td></tr>
<tr><td>Параметры:</td><td>path:string, int, pathlib.

Путь ко входному файлу(string). Может быть дескриптором открытого файла (int).</td></tr>
<tr><td></td><td>sr:number > 0 [scalar]

Частота дискретизации с которой будет автоматически передискредитировано до заданной скорости (по умолчанию sr = 22050).</td></tr>
<tr><td></td><td>Mono: bool

Преобразовать сигнал в моно.</td></tr>
<tr><td></td><td>Offset: float

Начать чтение после этого времени (в секундах).</td></tr>
<tr><td></td><td>Duration: float

Продолжительность в секундах загружаемого аудио</td></tr>
<tr><td></td><td>Dtype: numeric type

Тип данных y</td></tr>
<tr><td></td><td>res_type: str

Тип повторной выборки.</td></tr>
<tr><td colspan="2">Librosa.feature.mfcc</td></tr>
<tr><td colspan="2">librosa.feature.mfcc(y=None, sr=22050, S=None, n_mfcc=20, dct_type=2,</td></tr>
</table>

***norm='ortho'*, *lifter=0*, ***kwargs*)** Mel-кепстральные коэффициенты (MFCC)	
Параметры:	**y: np.ndarray** аудио временной ряд
	Sr: number > 0 [scalar] Частота дискретизации.
	n_mfcc: int > 0 [scalar] Количество mfcc для возврата.
	hop_length: int > 0 [scalar] Опциональный параметр. Устанавливает размер окна.
Librosa.feature.inverse.mfcc_to_audio	
librosa.feature.inverse.mfcc_to_audio(***mfcc*, *n_mels=128*, *dct_type=2*, *norm='ortho'*, *ref=1.0*, *lifter=0*, ***kwargs*)** Преобразовать кепстральные коэффициенты Mel-частоты в аудио**.**	
Параметры:	**Mfcc : np.ndarray [shape = (n_mfcc, n)]** Кепстральные коэффициенты Mel-частоты
	n_mels : int> 0 Количество частот Mel

2.3.2 Модуль SpeechRecognition.

Модуль SpeechRecognition – это обёртка над многими системами распознавания речи, с поддержкой многих движков и API, способна работать

онлайн и оффлайн. Этот модуль используется для тестирования запутанных команд, чтобы понять сможет ли голосовой помощник понять аудио запись.

2.4 Разработка классификатора.

2.4.2 Кластеризация. Алгоритм K-means.

Кластеризация — это разделение множества входных данных на группы по их схожести. Критерием для сравнения данных чаще всего является расстояние между этими данными. Евклидово расстояние — наиболее распространённая мера расстояния между двумя точками евклидова пространства, вычисляемое по теореме Пифагора. Для точек p = (p1, … , pn) и q = (q1, …, qn) евклидово расстояние вычисляется по формуле

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

K-means – это наиболее популярный метод кластеризации. Алгоритм k-средних выполняет разделение данных на заранее заданное количество кластеров в не маркированном многомерном наборе данных. Суть алгоритма такова, чтобы минимизировать среднеквадратичное отклонение на точках для каждого кластера. В виде формулы это можно представить как:

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (||x_i - \mu_j||^2)$$

Шаги алгоритма:

1. В начале берётся n случайных точек, как начальные центры для кластеров.
2. Для каждой точки вычисляется евклидово расстояние до всех центров.
3. Наш центр смещается в пространстве соответственно подсчётам. Каждая точка помещается в кластер того центра к которому он ближе.

4. Потом начинается всё заново, до тех пор, пока все точки не останутся в своих кластерах.

Главный недостаток этого алгоритма, это то, что необходимо заранее знать количество кластеров на которые мы будем разбивать данные. Для нашей программы, данные будут разбиваться на два кластера.

2.4.2 Выбор характеристик.

Для разработки классификатора необходимо выявить параметры которые с достаточной точностью помогут разделить данные на 2 кластера. Не все акустические характеристики достаточно уникальны, чтобы различить данные скрытых голосовых команд и обычных команд. Поэтому для того чтобы выявить подмножество достаточно уникальных функций из всех кандидатов, необходимо прибегнуть к статистическому анализу. Для чистоты эксперимента, было взято несколько команд в двух вариациях, потом разбив звук на акустические функции и нормализовав значения от 0 до 1, затем вычисляем оценку для каждого признака на основе уравнения X.

$$s = \frac{\bar{F}_{hid} - \bar{F}_{hum}}{max\left(\frac{\sqrt{\sum(F_{hid}(i)-\bar{F}_{hid})^2}}{n}, \frac{\sqrt{\sum(F_{hum}(j)-\bar{F}_{hum})^2}}{n}\right)}, \qquad (2)$$

Где Fhid(i) и Fhum(j) это значение каждой функции скрытой и обычной голосовой команды соответственно, а $\bar{F}$hid и $\bar{F}$hum – это средние значения функции. Рассчитанная оценка показывает насколько хорошо разделены функции двух типов голосовых команд. Используя небольшой набор скрытых и обычных голосовых команд, была рассчитана оценка для всех возможный кандидатов и были выбраны наиболее различимые функции.

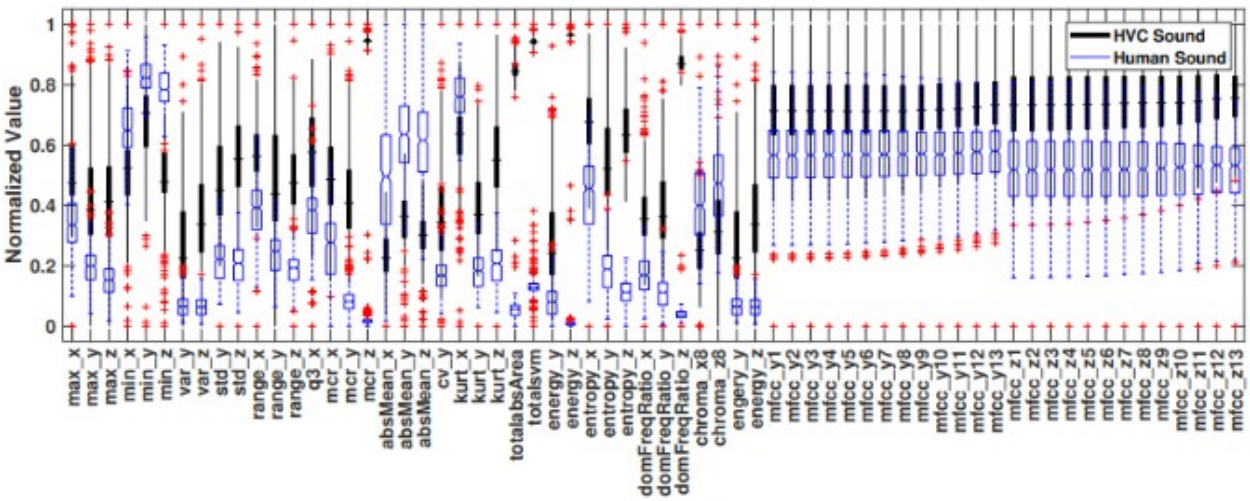

Рисунок X – распределение акустических функций обычных и скрытых команд.

На рисунке X показано распределение акустических функций запутанных и обычных команд. Было обнаружено, что некоторые акустические характеристики, такие как средняя скорость пересечения нуля, энтропия и MFCC, показывают достаточную разницу для построения классификатора.

3 ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

3.1 Результаты работы классификатора.

На рисунках X и X показаны координаты аудио записей относительно координат первого и второго mfcc коэффициента и третьего и четвертого. Зелёным показаны обычные команды, красным скрытые. Хотя многие аудиозаписи накладываются друг на друга, мы видим что преимущественно записи одной категории находятся близко к друг другу.

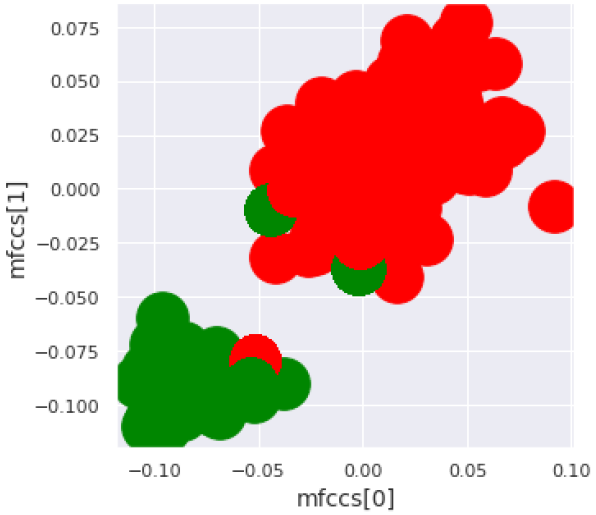

Рисунок X — Нахождение аудио записей относительно первого и второго mfcc коэффициента.

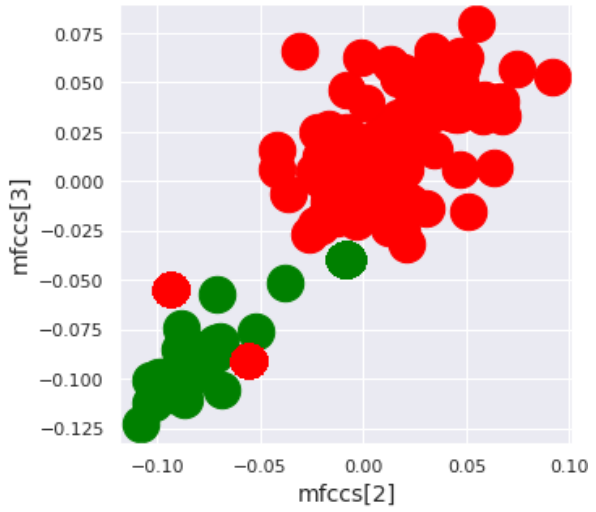

Рисунок X — Нахождение аудио записей относительно второго и третьего mfcc коэффициента.

На рисунках X и X показано нахождение кластеров относительно первого и второго коэффициента, третьего и четвёртого коэффициента mfcc. Большим прозрачным красным кругом показан центр каждого кластера. Желтым и фиолетовым помечены аудио записи входящие в один кластер.

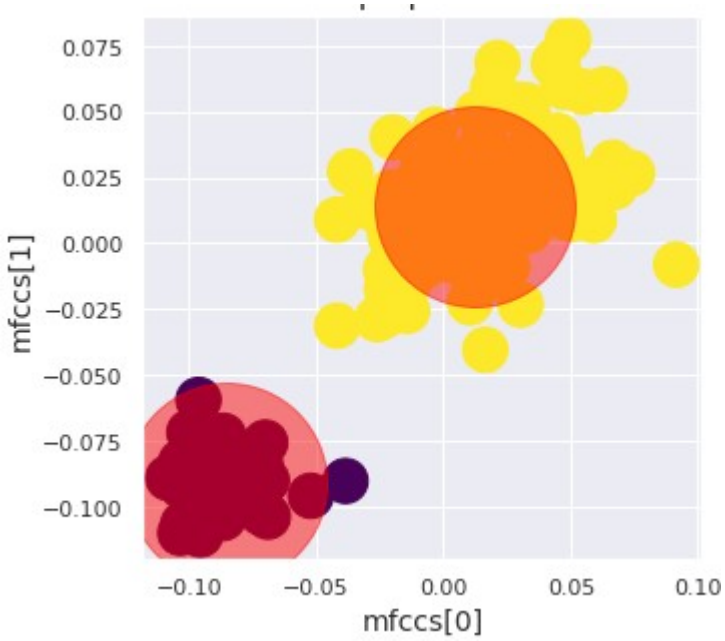

Рисунок X — центры кластеров относительно первого и второго коэффициента mfcc.

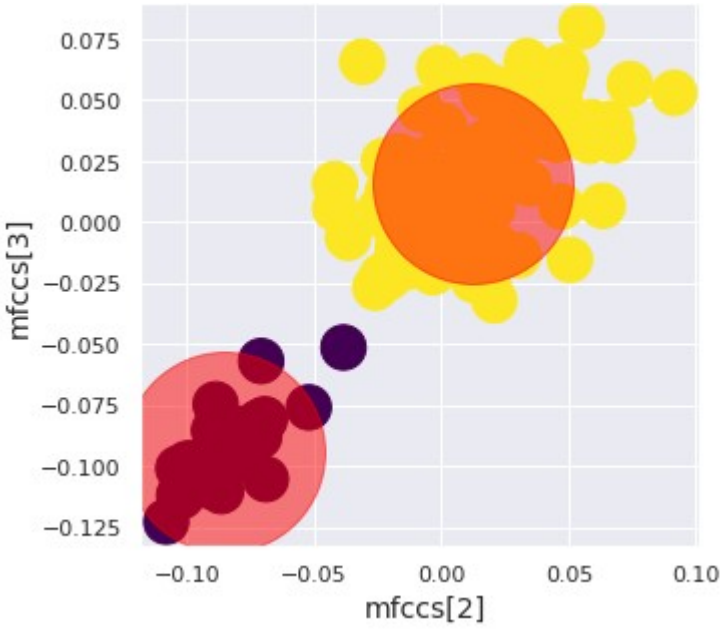

Рисунок X — центры кластеров относительно второго и третьего коэффициента mfcc.

Конечно рассмотрение задачи в двумерном пространстве не даёт полной картины, так как у каждой точки 20 измерений, но можно увидеть приблизительную работу классификатора. Зная какие записи обычные, а какие скрытые, была посчитана точность работы классификатора. Точность работы классификатора составляет около 94%.

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. 2016. Hidden voice commands. In Proceedings of the USENIX Security Symposium.
2. T. Vaidya, Y. Zhang, M. Sherr, and C. Shields. Cocaine Noodles: Exploiting the Gap between Human and Machine Speech Recognition. In USENIX Workshop on Offensive Technologies (WOOT), August 2015.
3. G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, “Dolphin Attack: Inaudible voice commands,” in ACM Conference on Computer and Communications Security. ACM, 2017, pp. 103–117.
4. D. P. W. Ellis. PLP and RASTA (and MFCC, and inversion) in Matlab. http://www.ee.columbia.edu/ ˜dpwe/resources/matlab/rastamat/, 2005
5. C. Ittichaichareon, S. Suksri, and T. Yingthawornsuk. Speech recognition using MFCC. In International Conference on Computer Graphics, Simulation and Modeling (ICGSM), 2012.
6. O. Viikki and K. Laurila. Cepstral Domain Segmental Feature Vector Normalization for Noise Robust Speech Recognition. Speech Communication, 25(13):133–147, 1998.
7. 2018. Text To Speech (TTS) service. http://www.fromtexttospeech. Com/.
8. Документация модуля Librosa – URL : https://librosa.github.io/librosa/index.html (дата обращения 23.05.2020)
9. Документация модуля speechrecognition – URL: https://github.com/Uberi/speech_recognition#readme (дата обращения 23.05.2020)
10.