# Введение

Система распознавания диктора — это совокупность элементов, позволяющих идентифицировать личность человека только на основе его голоса. Основой этой системы являются выделение особенностей каждого диктора, создание на основе этих особенностей модели диктора и сопоставление полученной модели с поданным на вход речевым файлом.

Возможность распознавания диктора основывается на том, что каждый человек обладает уникальным голосом, зависящим от строения голосового тракта, манеры говорить и характерного этому человеку активного лексикона.

Распознавание диктора обладает большой областью применения. Примеры можно найти в таких сферах деятельности, как аутентификация, call-центры, разведка, банковская деятельность. Наличие такой системы позволит сократить количество работников, отвечающих за обработку звонков клиентов, которым необходимо, например, сбросить пароль. А в совокупности с системой распознавания речи можно полностью автоматизировать этот вид деятельности. Банки и их клиенты получат больше мобильности при управлении счётами, ведь теперь для подтверждения той или иной операции необходимо лишь позвонить. Государство и спецслужбы также извлекут выгоду от использования таких систем. Это позволит осуществлять автоматический поиск преступников по записям телефонных разговоров.

Кроме этого, идентификация по голосу является наиболее удобной. Биометрические методы верификации основанные на уникальности отпечатков пальцев или сетчатки глаза обладают высокой надёжностью, но для их применения необходимо непосредственное присутствие человека, а это снижает мобильность.

Вообще говоря, системы распознавания диктора могут выполнять задачи двух видов: идентификация и верификация. Идентификация — это процесс

поиска голоса текущего диктора в базе данных остальных голосов, тогда как верификация подтверждает личность говорившего.

Кроме того, существует деление на тексто-зависимые и тексто-независимые. В тексто-зависимых системах фраза, которую должен произнести диктор, уже известна, а в тексто-независимых модель диктора учитывает только характеристики его голоса.

Основной проблемой при построении и последующем использовании модели является изменчивость голоса одного и того же диктора в течение времени. Причинами этого могут являться изменение возраста или болезнь. Также на модель могут влиять такие внешние параметры, как шум окружающей среды и смена оборудования для записи голоса.


Рис. 1. Регистрация нового диктора


Рис. 2. Идентификация диктора

На рисунках изображены элементы системы распознавания диктора: регистрация нового диктора (рис. 1), идентификация диктора (рис. 2).

Эта работа посвящена разработке системы идентификации диктора по голосу на основе MFCC-коэффициентов и алгоритма машинного обучения Random Forest.

# 1. Обзор предыдущих исследований

Работы по созданию систем распознавания голоса начались примерно полстолетия назад. Были предложены “отпечатки голоса” (англ. voice prints), которые, однако, не оказались неэффективны, потому что были сильно подвержены изменчивости, и не могли быть использованы в роли признаков [1].

Последующие исследования были направлены на поиск таких признаков голоса, которые учитывали бы все особенности речеобразования. Среди них: частота основного тона, формантные частоты, барк-шкала и другие.

Наиболее популярными в последние годы были коэффициенты линейного предсказания (англ. *linear predictive cepstral coeffcients*, LPCC) и мел-кепстральные коэффициенты (MFCC).

Главная идея LPCC состоит в том, чтобы вычислить текущий сдвиг цифрового сигнала на основе данных о предыдущих сдвигах. Пусть $s(i)$ — сигнал. Оценка текущего сдвига сигнала $\bar{s}(i)$ определяется как линейная комбинация предыдущих сдвигов:

$$\bar{s}(i) = \sum_{k=1}^{N} a_k (i-k) .$$

Необходимо найти такой набор коэффициентов $\{a_k\}$, чтобы минимизировать среднеквадратичную ошибку:

$$< (s(i) - \bar{s}(i))^2 > \ \rightarrow \ min .$$

Величина $N$ называется порядком линейного предсказания [2].

Также в [3] было проведено сравнительное исследование различных признаков, построенное на основе шипящих звуков. В качестве признаков среди прочих использовались: нормированная автокорреляция, LPCC, коэффициенты отражения LPCC, нормированная автокорреляция LPCC, кепстр LPCC и MFCC. В таблице 1 приведены результаты полученных исследований.

Табл. 1. Результаты численного исследования систем признаков

| **Система признаков** | **Вероятность верификации** |
|---|---|
| Нормированная автокорреляция | 0,44 |
| Коэффициенты LPCC | 0,78 |
| Коэффициенты отражения LPCC | 0,96 |
| Нормированная автокорреляция LPCC | 0,78 |
| Кепстр LPCC | 0,68 |
| MFCC | 0,96 |

В последние годы преобладают кепстральные коэффициенты, так как они предоставляют высокий уровень распознавания. Поэтому большинство исследований направлено на методы построения моделей диктора и различных классификаторов. Классические методы моделирования дикторов могут быть разделены на шаблонные и стохастические.

Шаблонные методы напрямую сравнивают тестовые и тренировочные векторы особенностей, а величина искажений между ними представляет степень их сходства. К ним можно отнести квантование вектора (англ. *vector quantization*) [4], алгоритм динамической трансформации

временной шкалы (англ. *dynamic time wrapping*) [3], метод опорных векторов [5][6] и его модификации [7].

## 1.1 DTW

*Dynamic time warping (*DTW) — это метод, позволяющий измерить схожесть двух последовательностей, длина которых может различаться [8]. Пусть даны две последовательности $Q = \{q_1, ..., q_n\}$ и $C = \{c_1, ..., c_m\}$. Введём матрицу выравнивания последовательностей $d_{m \times n}$, в позиции $(i,j)$ которой содержится евклидово расстояние между элементами $c_i$ и $q_j$. Далее строится матрица трансформаций D:

$$D_{ij} = d_{ij} + min(D_{i-1\,j},\ D_{i-1\,j-1},\ D_{i\,j-1}).$$

После заполнения этой матрицы, строится оптимальный путь трансформации $W$ — набор смежных элементов матрицы, устанавливающий соответствие между исходными векторами. Вектор $W$ представляет собой путь, который минимизирует общее расстояние между $Q$ и $C$, и определяется как $w_k = (i,j)_k$. При этом должны выполняться следующие условия:

1. $w_1 = (1,1),\ w_K = (m,n),$ где $K$ — длина пути и $max(m,n) \le K < m + n$.
2. Если $w_k = (a,b)$ и $w_{k+1}(c,d)$, то должны выполняться неравенства $0 \le a - c \le 1, 0 \le b - d \le 1$.

Для нахождения оптимального пути используется формула:

$$DTW(Q,C) = min\left\{\sum_{k=1}^{K} d(w_k)\right\}\ [8].$$

## 1.2 Квантование вектора

Дана последовательность входных векторов $X = \{x_1, ..., x_T\}$ и кодовые вектора $R = \{r_1, ..., r_K\}$. Среднее искажение определятся формулой

$$D_Q(X, R) = \frac{1}{T} \sum_{t=1}^{T} min_k(d(x_t, r_k)), \quad (2)$$

где $d(,)$ — мера расстояния. Чем меньше (2), тем вероятнее принадлежность X и R одному диктору [2].

## 1.3 Метод опорных векторов

Метод опорных векторов (англ. support vector machine, SVM) представляет собой бинарный классификатор, который строит разделяющую гиперплоскость между двумя классами. В верификации диктора один класс содержит вектора зарегистрированного диктора, а другой класс — вектора “нарушителя”. Метод максимизирует расстояния ближайших точек классов до разделяющей гиперплоскости с помощью дискриминантной функции

$$f(x) = \sum_{i=1}^{N} a_i t_i K(x, x_i) + d,$$

где $t_i \in \{+1, -1\}$, $\sum_{i=1}^{N} a_i t_i = 0$ и $a_i > 0$. Опорные вектора $x_i$, их веса $a_i$ и их смещение $d$ вычисляются из набора тренировочных векторов [2].

В стохастических методах каждый диктор определяется некоторой фиксированной функцией плотности вероятности. Во время этапа обучения определяются параметры этой функции. Соответствие определяется оценкой максимального правдоподобия модели. Наиболее популярным стохастическим

методов является модель гауссовых смесей (англ. Gaussian mixture model, GMM). GMM может быть использована в комбинации с SVM [5].

## 1.4 GMM

GMM состоит из конечного числа взвешенных гауссовских компонент. GMM характеризуется функцией плотности вероятности:

$$p(x|\lambda) = \sum_{k=1}^{K} p_k N(x|\mu_k, \sum_k),$$

где $K$ — это количество компонент, $p_k$ — вес $k$-ой компоненты, а

$$N(x|\mu_k, \sum_k) = (2\pi)^{-\frac{2}{d}} |\sum_k|^{-\frac{1}{2}} exp\left\{ -\frac{1}{2}(x-\mu_k)^T \sum_k^{-1} (x-\mu_k) \right\}$$

является функцией плотности распределения с математическим ожиданием $\mu_k$ и матрицей ковариации $\sum_k$ [2].

Для автоматизации процесса распознавания были использованы и другие алгоритмы машинного обучения, помимо описанного метода опорных векторов. Например, искусственные нейронные сети [7].

# 2. Выделение особенностей

Задача предварительной обработки сигнала состоит в выделении особенностей голоса для их последующего анализа. Речевой сигнал содержит в себе множество различных особенностей, но не все из них подойдут для идентификации диктора. Чтобы обеспечить надёжную и быструю идентификацию, выделяемые признаки должны обладать следующим набором свойств [2]:

- сильно различаться для разных дикторов и слабо для разных записей одного диктора;
- быть надёжными относительно шума и искажений;
- быть легко выделяемыми из речевого сигнала;
- быть сложными для подражания/подделки;
- не реагировать на изменение здоровья диктора;
- общее количество особенностей должно быть относительно мало.

С физиологической точки зрения, все признаки можно разделить на 5 основных групп [2]:

1. краткосрочные спектральные особенности.
2. особенности источника голоса;
3. спектро-временные особенности;
4. просодические особенности;
5. высокоуровневые особенности.

К первым двум относятся спектр, спектро-временные особенности включают ритм, темп, продолжительность речи, а просодические и высокоуровневые особенности характеризуются акцентом и активным запасом слов.

Речевой сигнал несильно изменяется за короткие (5-10 мс) промежутки времени. Если же рассматривать большие периоды, то можно заметить, как сигнал меняется в зависимости от произнесённых звуков [2][9].

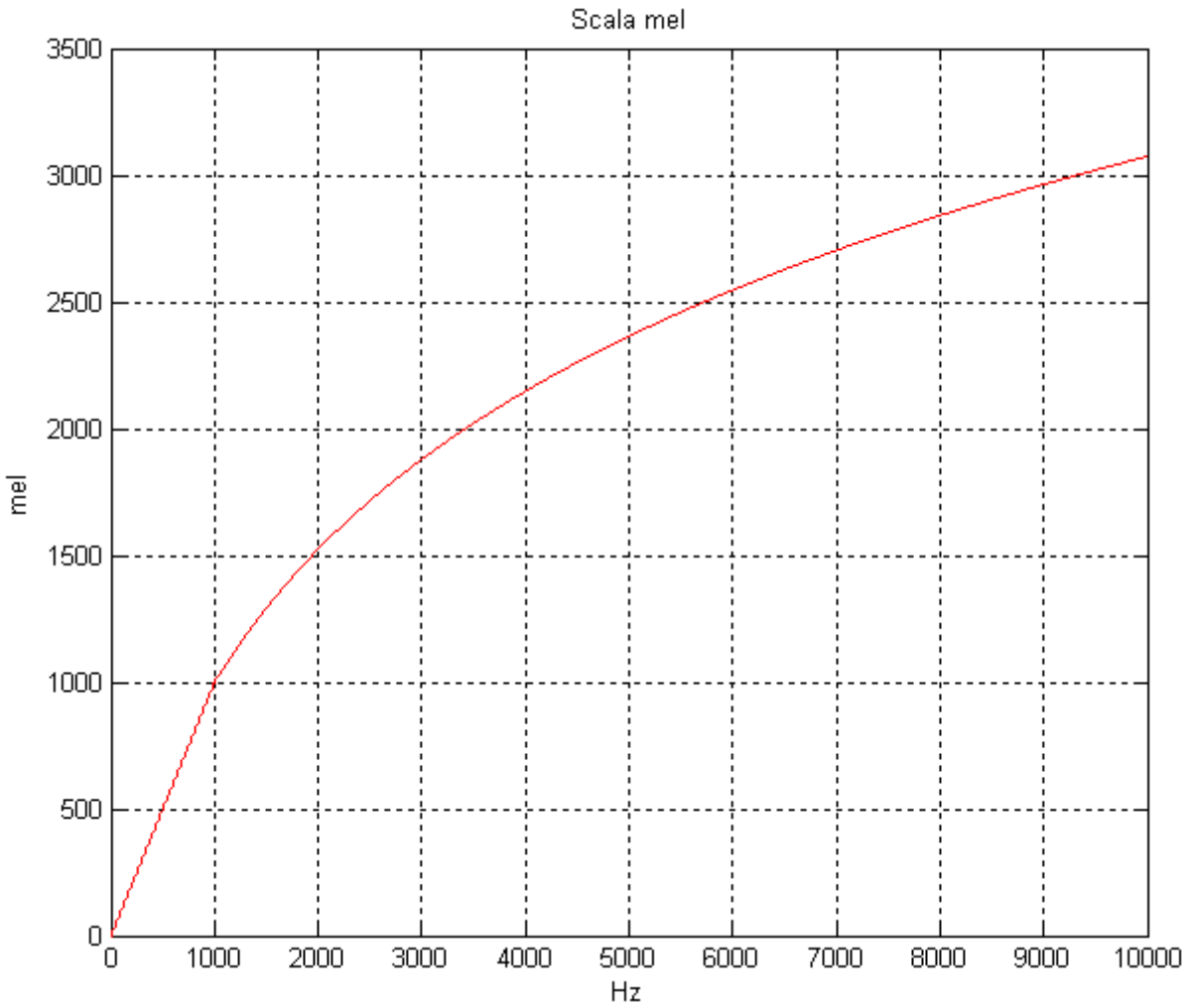


Рис. 3. Мел-шкала

Человеческое ухо воспринимает звук нелинейно от его частоты: чем ниже частота, тем выше чувствительность. Поэтому шкала частот преобразуется в мел-шкалу (рис. 3), в которой присутствуют так называемые критические полосы, такие что в пределах одной полосы частоты сигналов неразличимы [9][1]. Использование этой шкалы делает наши признаки (особенности) более приближёнными к тому, как слышат люди. Формула для преобразования частоты в мел-шкалу [9]:

$$M(f) = 1125 \bullet ln(1 + \frac{f}{700}). \qquad (1)$$

Формула для обратного преобразования [2]:

$$M^{-1}(m) = 700 \bullet (exp(\frac{m}{1125}) - 1).$$

Всё это приводит к выбору мел-кепстральных коэффициентов (MFCC) в качестве характеристик голоса. Изложим алгоритм вычисления MFCC так, как это было сделано в [9]:

1. Делим сигнал на отрезки по 20-40мс. Шаг обычно равен 10мс, что создаёт некоторое пересечение отрезков. Если речевой файл не делится на целое число отрезков, то можно дополнить его нулями. Обозначим наш основной сигнал как $s(n)$. А после разделения мы получаем $s_i(n)$, где $n$ изменяется в пределах выбранной длины отрезка, а $i$ — в пределах количества отрезков. Следующие шаги применяются к каждому отрезку, из каждого из которых извлекается набор MFCC коэффициентов.
2. Вычисляем спектрограмму каждого отрезка в отдельности. Для этих целей используем дискретное преобразования Фурье (ДПФ):

$$S_i(k) = \sum_{n=1}^{N} s_i(n) \bullet h(n) \bullet exp(\frac{-j \bullet 2 \bullet \pi \bullet k \bullet n}{N}),$$

где $1 \le k \le K$; $h(n)$ — оконная функция; $K$ — длина ДПФ. Тогда спектрограмма для отрезка $s_i(n)$ вычисляется по формуле:

$$P_i(k) = \frac{1}{N} \bullet |S_i(k)|^2.$$

После вычисления ДПФ мы получаем $S_i(k)$, где $i$ обозначает номер отрезка, соответствующий области времени сигнала. $P_i(k)$ тогда является мощностью спектра для $i$-го отрезка.

3. Полученную спектрограмму переводим в мел-шкалу с помощью формулы (1). После этого используются треугольные фильтры, суммирующие количество энергии на определённом диапазоне частот и позволяющие получить мел-коэффициент.
4. Вычисляем логарифм каждого числа, полученного на шаге 3.
5. Осуществляем дискретное косинусное преобразование и получаем кепстральные коэффициенты.

# 3. Супервекторы

После этапа выделения особенностей получаем из речевого файла матрицу размерности $N \times M$, где $M$ — это количество характеристик для каждого отрезка, $N$ — количество таких фрагментов, которое зависит от длины файла и от выбранного размера фрагментов. Задача состоит в том, как представить матрицы характеристик речевых файлов в общем для всех виде.

Для этого применяются супервекторы — способ представить речевой файл в виде вектора фиксированной размерности. Часто супервекторы являются векторами большей размерности, составленными из нескольких векторов меньшей размерности. В этой работе рассмотрены два способа представления супервекторов.

## 3.1 Среднее значение векторов

Так как необходимо получить обобщённую характеристику нескольких векторов, то на первый взгляд логично было бы использовать их усреднённое значение. Покажем на примере двух векторов размерности $n$: $X = \{x_1, x_2, ..., x_n\}$, $Y = \{y_1, y_2, ..., y_n\}$. Получим:

$$z_i = \tfrac{1}{2}(x_i + y_i),\ 1 \le i \le n.$$

## 3.2 Супервектор *k*-средних

Дан набор из $N$ векторов $(x_1, x_2, ..., x_N)$ размерности $d$. Метод $k$-средних делит это множество векторов на $k$ кластеров так, чтобы минимизировать суммарное квадратичное отклонение точек кластеров от центроидов этих кластеров:

$$dev = \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2,$$

где $S_i$ — полученные кластеры; $i = \overline{1,k}$; $\mu_i$ — средние значения векторов в *i*-м кластере.

Дан исходный набор из *k* средних значений векторов $m_1^{(1)}, ..., m_k^{(1)}$. Далее запускается итерационный процесс, каждая итерация которого состоит из двух этапов:

- Каждый вектор набора ставится в соответствие кластеру, расстояние до которого минимально.

$$S_i^{(t)} = \left\{ x_p : \|x_p - m_i^{(t)}\|^2 \le \|x_p - m_j^{(t)}\|^2 \ \forall j,\ 1 \le j \le k \right\}$$

- Находятся новые центры кластеров.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Алгоритм завершатся, когда центры кластеров перестают меняться [10].

# 4. Random Forest

В качестве метода классификации векторов особенностей был выбран алгоритм машинного обучения Random Forest, который является набором деревьев решений.

Дерево решений — это алгоритм машинного обучения, состоящий из слабых классификаторов, объединённых в дерево. Дерево решений имеет два типа узлов [11]:

- Внутренние — узлы, имеющие более одного потомка.
- Терминальные — узлы, не имеющие потомков.

Внутренний узел дерева представляет собой разделяющую функцию, которая определяет к какому потомку будет отнесён текущий элемент. Для бинарного разделения эта функция имеет вид:

$$h(\overline{v}, \theta) : R^d \times T \rightarrow \{0, 1\},$$

где $T$ — множество параметров разделения, а $\overline{v}$ — вектор особенностей.

Чтобы выбрать разделяющую функцию, нужно оценить результат разделения для каждой функции. Критерием для этого может быть прирост информации (англ. information gain) [11]:

$$I = H(S) - \sum_{i \in [L,R]} \frac{|S^i|}{|S|} H(S^i),$$

где $H(S)$ — энтропия, которая вычисляется так:

$$H(S) = -\sum_{c \in C} p(c) \bullet log(p(c)).$$

Здесь множество S делится на два подмножества $S^L$ и $S^R$. С учётом этого критерия параметры разделяющей функции вычисляются как $\theta = arg\ max\ I,\ \theta \in T$.

Чтобы определить, когда будет создан терминальный узел применяются различные критерии [12]:

- Достигнута максимальная глубина дерева.
- Энтропия меньше заданного уровня, то есть полученное разделение достаточно чистое.

# 5. Эксперименты

## 5.1 База данных

В работе была использована база дикторов CSTR VCTK Corpus. Она содержит образцы голоса 109 различных людей, родным языком которых является английский, и имеющих различные акценты.

## 5.2 Результаты

Далее приведены результаты экспериментов с разработанной системой идентификации. Для этого была проведена кросс-валидация по следующим параметрам модели:

- количество деревьев в Random Forest (trees);
- максимальная глубина деревьев (max_depth);
- критерий оценки чистоты разделения (measure_quality_of_split);
- метод для вычисления супервектора (supervector);
- количество кластеров для супервектора по методу k-средних;
- количество используемых мел-кепстральных коэффициентов (MFCCs).

Полученные модели будут сравниваться по точности идентификации на тестовой выборке. Измерялся также показатель на обучающей выборке, но в большинстве случаев он равнялся 100%, в этих случаях он не указывался.

В первую очередь был протестирован метод построения супервектора как среднее значение векторов. Как видно из рисунков (рис.4 — рис.7), этот метод не даёт высокий результат независимо от параметров модели.

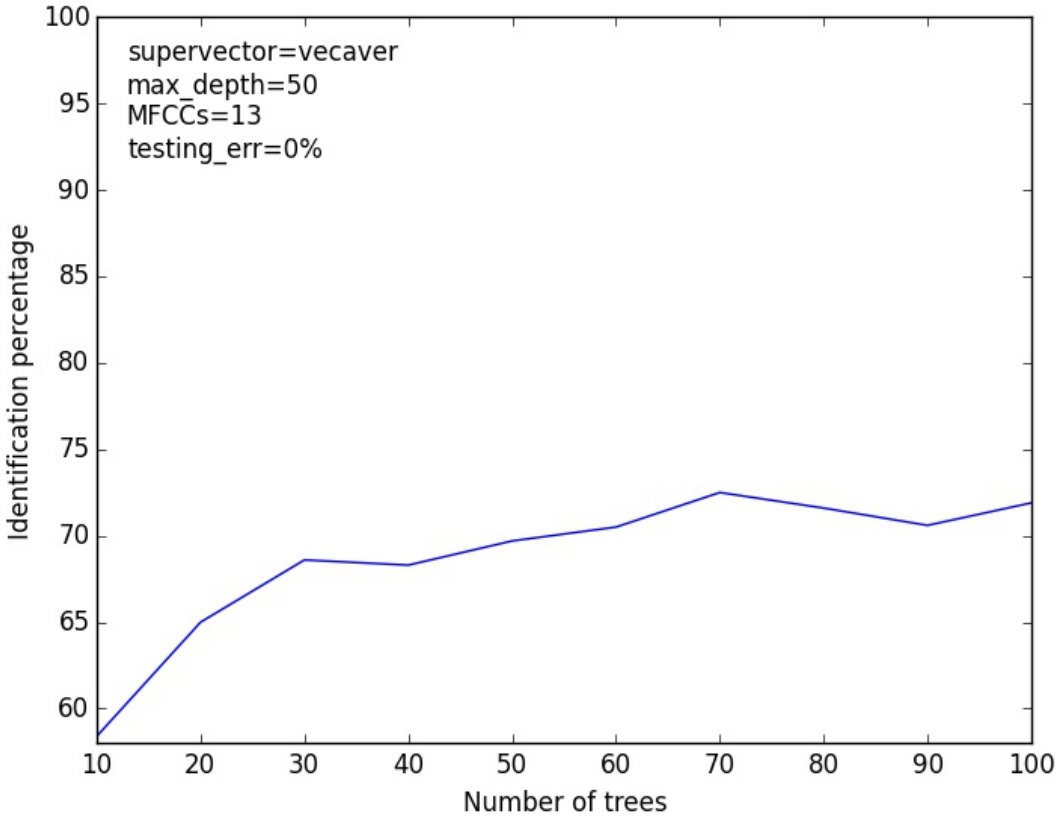


Рис.4. Зависимость от количества деревьев

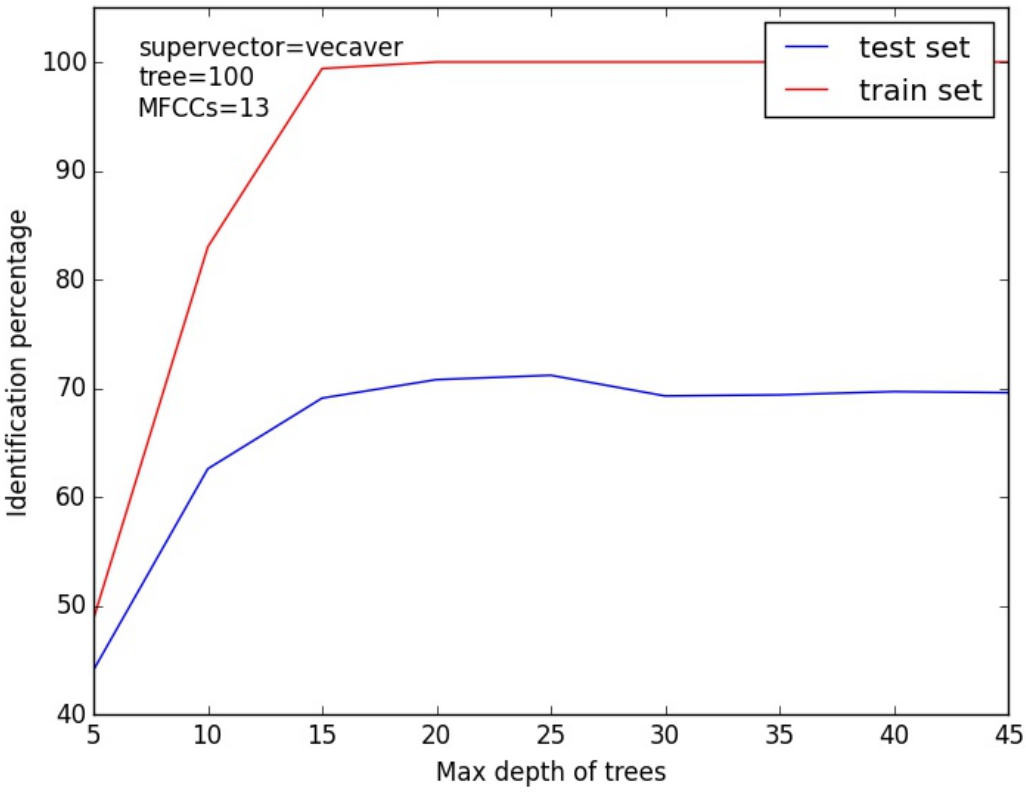


Рис. 5. Зависимость от максимальной глубины деревьев

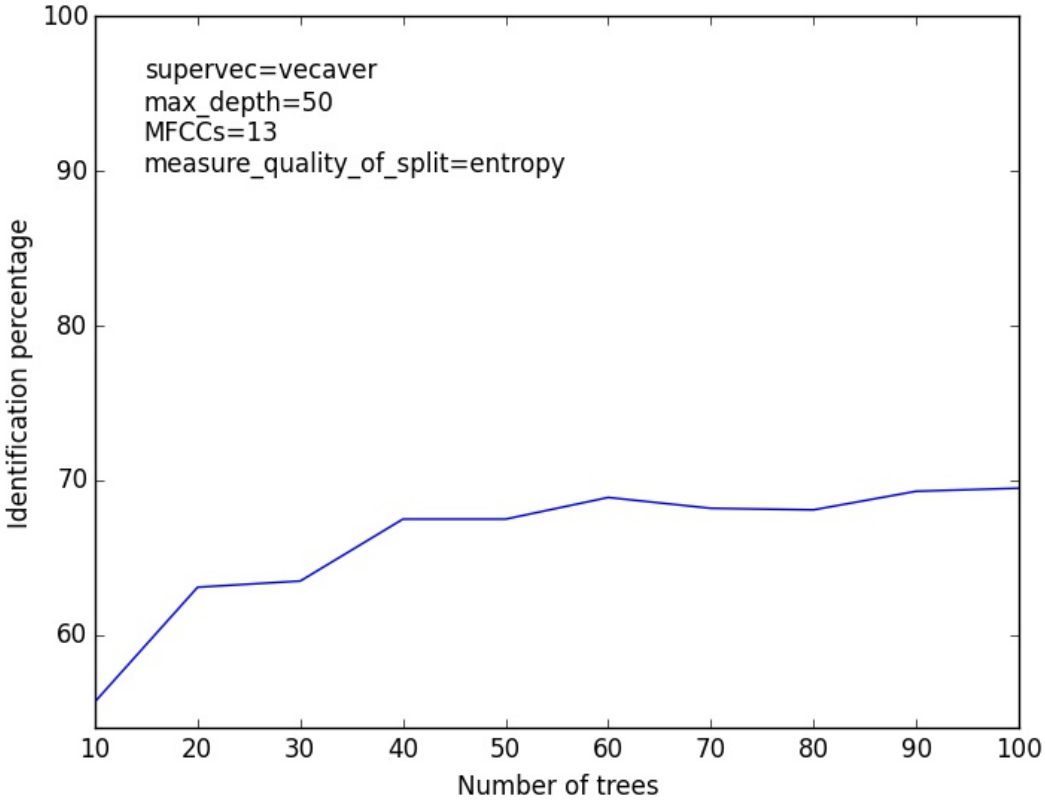


Рис. 6. Зависимость от количества деревьев (entropy)

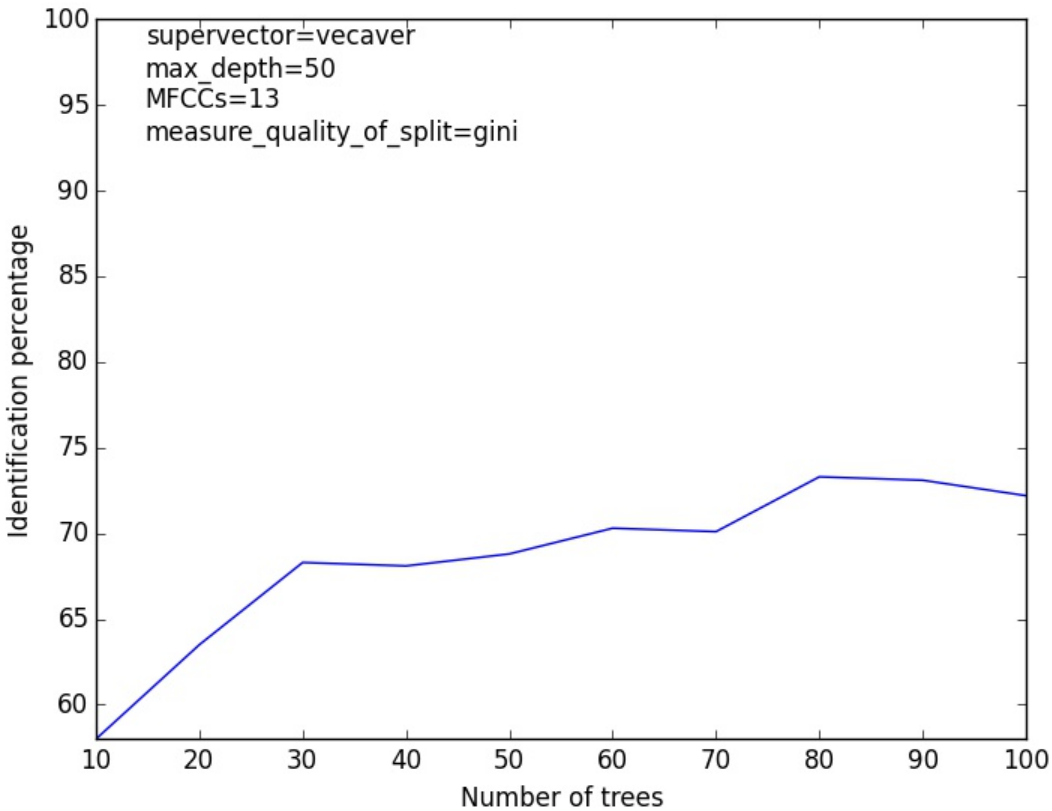


Рис. 7. Зависимость от количества деревьев (gini)

Далее был испытан метод k-средних для построения супервектора (рис.8, рис.9). Изначально параметр, отвечающий за количество кластеров, приравняли к двум. После серии экспериментов (рис.11) было решено, что это оптимальное значение данного параметра.

Кроме того, до 26 было увеличено количество признаков, выделяемых из речевого файла (рис.10).

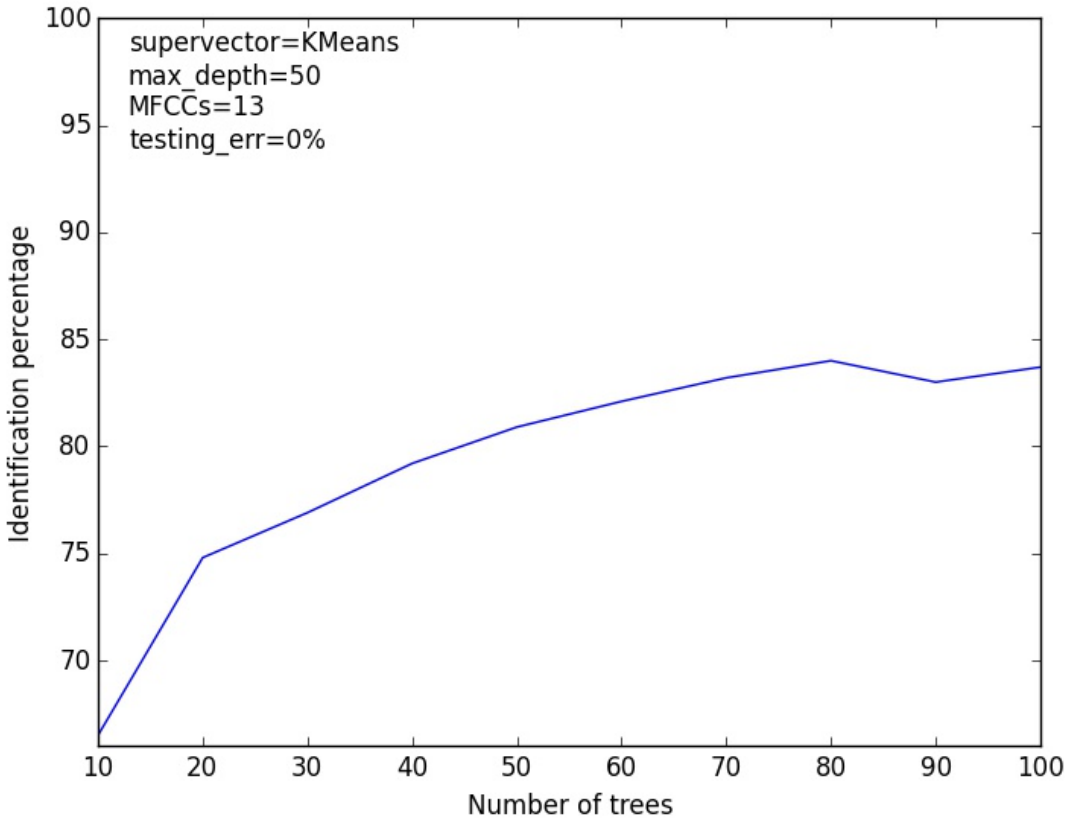


Рис. 8. Зависимость от количество деревьев

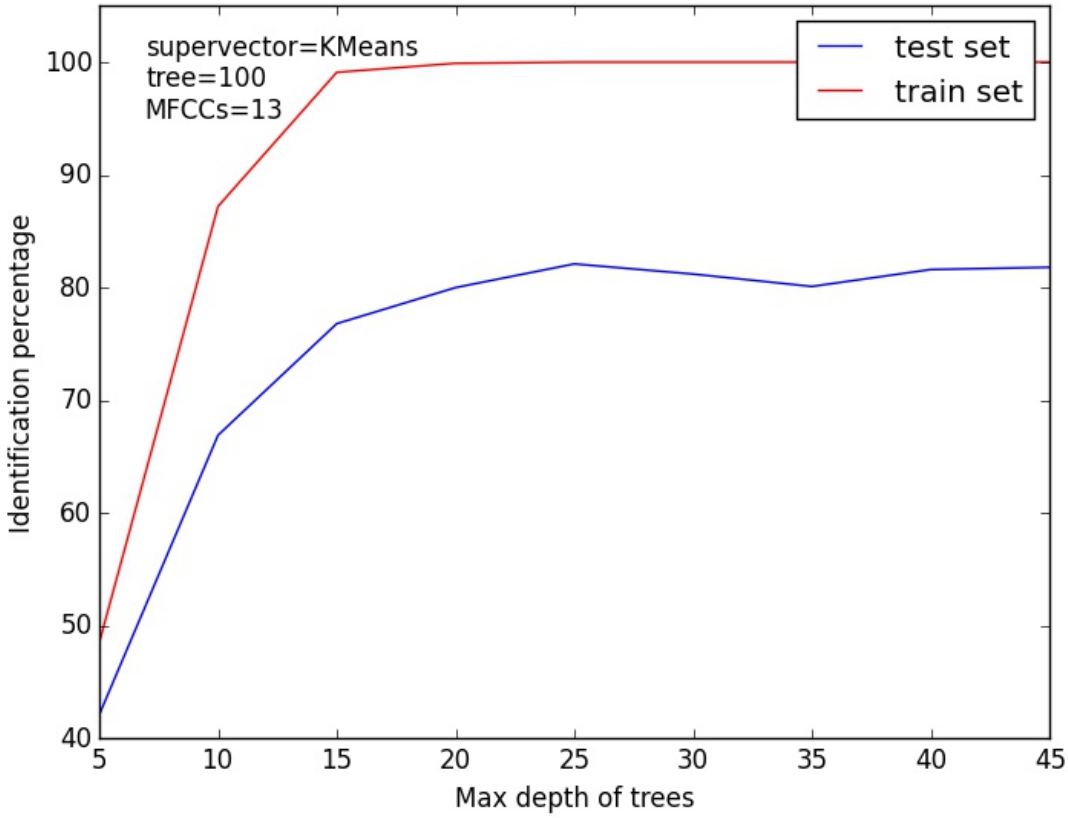


Рис. 9. Зависимость от максимальной глубины деревьев

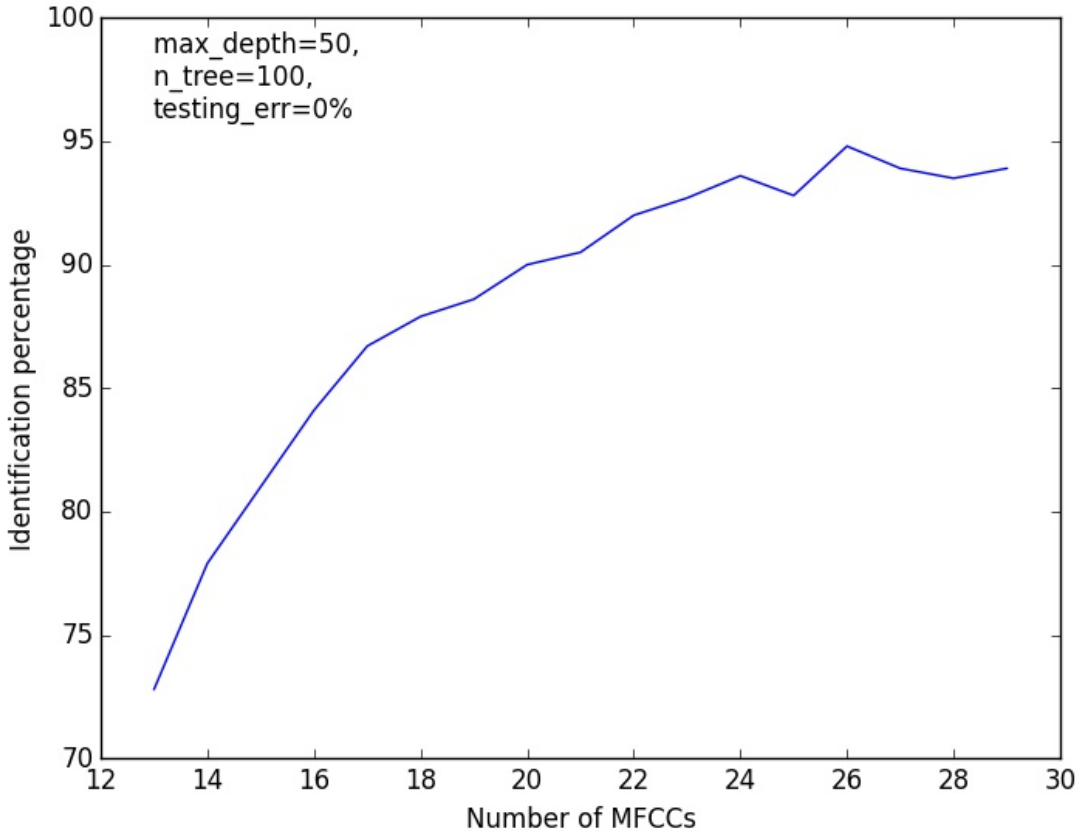


Рис. 10. Зависимость от количества признаков

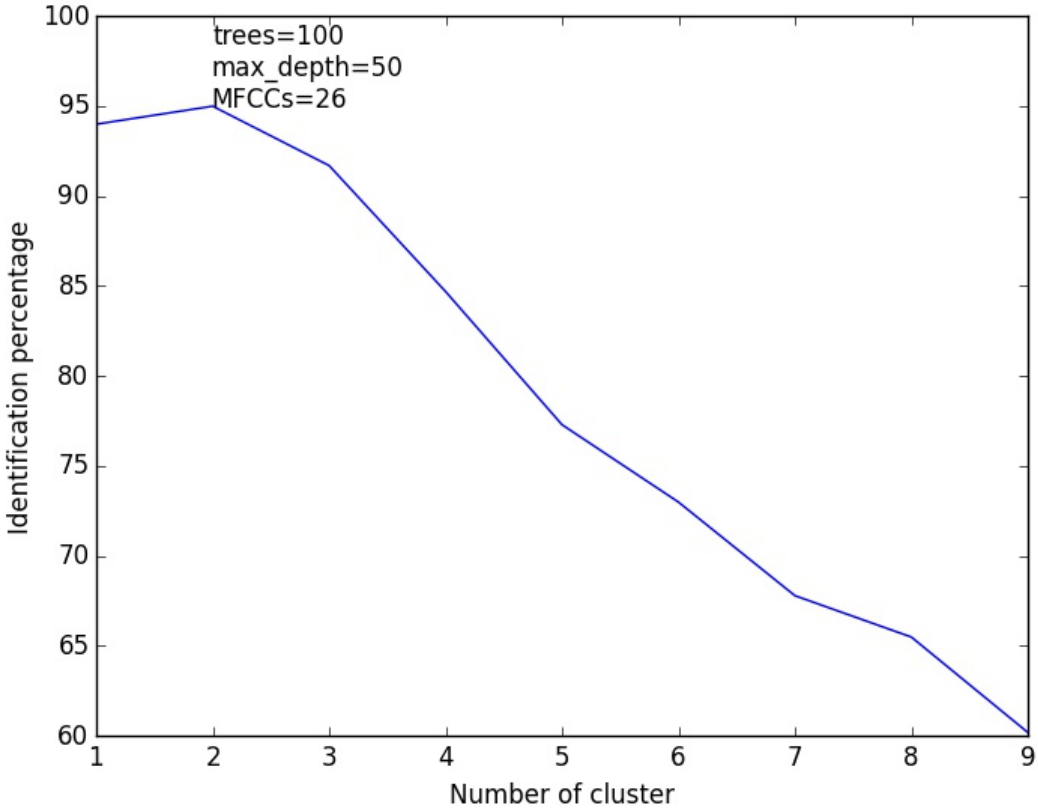


Рис. 11. Зависимость от количества кластеров

Для выбора оптимальных параметров была снова проведена серия экспериментов, определяющих зависимость точности идентификации от максимальной глубины деревьев. Установив этот параметр равным 25, получили следующие результаты для различного количества деревьев (рис.12).

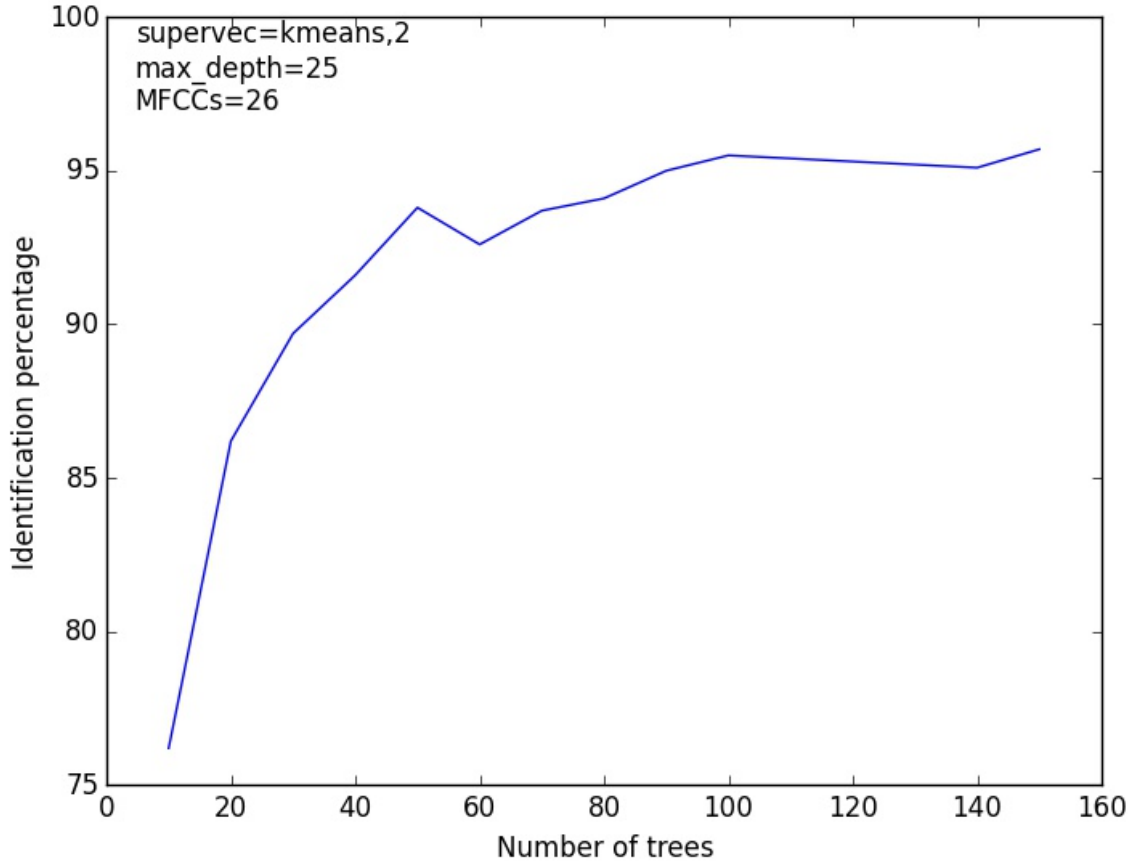


Рис. 12. Зависимость от количества деревьев.

В результате вышеперечисленных действий был получен показатель 95% успешной идентификации на тестовой выборке.

На основе оптимальных параметров были проведены эксперименты по определению следующих признаков дикторов: пол, возраст, акцент (табл. 2).

Табл. 2. Точность для различных признаков диктора

| **Признак диктора** | **Точность определения** |
|---|---|
| Пол | 98,7% |
| Возраст | 82,8% |
| Акцент | 85,2% |

# Заключение

В данной работе была разработан алгоритм идентификации диктора по голосу. В качестве признаков речевого файла были выбраны хорошо зарекомендовавшие себя мел-кепстральные коэффициенты. Супервектор был вычислен по алгоритму машинного обучения k-средних. Для построения модели диктора был использован другой алгоритм машинного обучения — Random Forest.

Разработанный алгоритм реализован в виде программной системы. Проведены экспериментальные исследования зависимости эффективности атаки от ее параметров, в результате которых был получен показатель в 95% точности идентификации диктора.