

УДК 004.8

doi: 10.15622/rcai.2025.095

## АЛГОРИТМЫ ИДЕНТИФИКАЦИИ КЛАССА В ЗАДАЧЕ РАСПОЗНАВАНИЯ КЛАВИАТУРНОГО ПОЧЕРКА С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ

Д.Н. Кобзаренко (*kobzarenko\_dm@mail.ru*)<sup>A,B</sup>

А.Г. Мустафаев (*arslan\_mustafaev@mail.ru*)<sup>A,B</sup>

<sup>A</sup> Дагестанский государственный институт народного хозяйства,  
Махачкала

<sup>B</sup> МИРЭА – Российский технологический университет, Москва

В рамках научных исследований по идентификации личности на основе клавиатурного почерка с использованием современных методов искусственного интеллекта, получены экспериментальные результаты. На основе новых разработанных алгоритмов получен максимальный результат точности распознавания классов на тестовых выборках. Результаты показывают высокую эффективность выбранных моделей, методов обработки исходных и алгоритмов идентификации класса в задаче распознавания личности по клавиатурному почерку.

**Ключевые слова:** клавиатурный почерк, машинное обучение, задача классификации, CatBoost, Python.

### Введение

Отечественные работы относительно формулировки задачи идентификации личности по клавиатурному почерку в подавляющем большинстве носят теоретико-гипотетический характер без каких-либо описаний конкретных технических решений и проблем, возникающих в контексте рассматриваемой задачи. Авторами современных публикаций предполагается, что задача идентификации клавиатурного почерка может быть полезной в различных областях, таких как кибербезопасность, криминалистика, психология и др. Коротко остановимся на некоторых работах.

В работе [Сатыбалдиева, 2024] приведены общие сведения о том какие методы в основном используются для подобных задач в процентном соотношении и методы машинного обучения с нейронными сетями не на первом месте, что неудивительно, поскольку обзор соответствует 2000-м

годам согласно приведенному списку литературы. В современных же исследованиях подобного рода (классификация и регрессия) машинное обучение и/или нейронные сети вытеснили старые методы и алгоритмы, и являются инструментами номер 1.

Имеется готовое техническое решение подобной задачи в виде свидетельства программы для ЭВМ [Лунев, 2024], в описании которого программа позиционируется как средство, позволяющее собрать статистику и провести аутентификацию пользователя по поведенческим паттернам, свойственным каждому человеку. Оцениваются параметры, скорость и динамика набора текста, время перехода между клавишами и опечатки. Более конкретной информации (какие используются методы, алгоритмы, точность прогноза и т.д.), кроме того, что основа программы написана на Python по данному свидетельству нет, как отсутствуют и публикации, связанные с ним. Поэтому сравнивать наши результаты с приведенным программным обеспечением невозможно отсутствует.

В работе [Варламова, 2023] используется и анализируется (без машинного обучения) так называемый оценочный параметр времени удержания клавиш для определения пользователя и утверждается, что время удержания клавиш способно идентифицировать пользователей в 85% случаев. Причем в конце статьи поясняется, что суть исследования состоит в идентификации по типу «свой»-«чужой», что соответствует более простому варианту прогнозной модели – бинарной классификации.

Разработка абстрактной модели искусственной иммунной сети и ее использование для распознавания образов клавиатурного почерка описана в [Сулавко, 2020]. Здесь подробно рассматриваются математические и технические аспекты построения данной модели, ключевыми особенностями которой являются то, что система также выполняет классификацию по принципу «свой»-«чужой», и основана на однородных исходных данных (обучающие и тестовые данные вводятся в виде одной и той же небольшой фразы многократным повторением  $N$  раз, кроме того, данная фраза должна быть введена безошибочно, иначе пример не заходит в базу).

Мировой опыт решения подобной задачи состоит из подробного рассмотрения и сопоставления авторами различных вариантов моделей как классического машинного обучения наряду с нейронными сетями [Sharma et al., 2023], так и исключительно нейронных сетей различных модификаций и функций активации [Acien et al., 2021]. А с учетом того, что в задаче распознавания клавиатурного почерка в целом не удастся приблизиться к 100% результату, в последнее время решили совместить клавиатурный почерк с движением мыши [Wang et al., 2023], что несомненно дает свои преимущества с точки зрения качества обучения модели, поскольку число независимых признаков возрастает.

Исходя из опыта наших предыдущих исследований, мы установили, что перебор архитектур нейронных сетей (DNN, Conv1D, LSTM), по сравнению с другими эффективными моделями машинного обучения (XBoost, CatBoost) не дает какого-либо прироста в точности прогноза и даже порой уступает последним. Поэтому повышение эффективности в решении задачи нам представляется не в поиске оптимальной архитектуры модели и параметров (что конечно тоже важно, но не дает хорошего прироста), а в работе с исходными данными до их поступления в модель и в постобработке результатов выхода модели с целью принятия решения о прогнозе. Для этого нами разработаны рассматриваемые далее оригинальные алгоритмы принятия решения по принадлежности пула примеров, сгенерированных на основе набора на клавиатуре нескольких предложений, тому или иному объекту.

Основные характеристики задачи идентификации клавиатурного почерка в нашей концепции ее реализации заключаются в следующем:

- задача позиционируется как мультиклассовая классификация (на выходе нужно идентифицировать владельца клавиатурного почерка);
- текст, вводимый субъектом для генерации обучающего и тестового набора примеров неоднородный, не повторяющийся и не пересекающийся;
- при вводе текста субъекту не ставятся никаких ограничений (допускается свободное редактирование, исправления ошибок и т.д.), за исключением разумного лимита времени достаточного для ввода текста даже для неопытного пользователя ПК;
- признаки классификации формируются только на основании локальной динамики набора текста (именно этот параметр является существенным, что установлено предыдущими нашими экспериментами);
- задача решается на основе моделей машинного обучения, для идентификации класса используется алгоритм, который принимает решение на основе пула тестовых примеров (сгенерированных на основе набранного фрагмента текста), пропущенных через модель.

От алгоритма идентификации класса на основе пула примеров, пропущенных через модель, зависит точность распознавания. Данная работа посвящена экспериментальному изучению этого вопроса с разработкой таких алгоритмов и оценкой их точности.

## **1. Постановка задачи идентификации клавиатурного почерка на основе машинного обучения**

Участников испытаний (число классов) 35 человек. Испытуемым предложено пройти процедуру снятия клавиатурного почерка в комфортных условиях, на любой удобной клавиатуре (мембранного типа), в том числе и на ноутбуке.

Испытуемый вводит осмысленную текстовую информацию, разделяемую на небольшие порции (по 4-6 строк), называемые *эксперимент 1, эксперимент 2, ..., эксперимент N*. Допускается перерыв любой длительности между экспериментами (чтобы не накапливалась усталость), но во время набора текста (тайпинга) в рамках одного эксперимента, испытуемый должен полностью концентрировать свое внимание на данной процедуре. В среднем на 1 эксперимент тайпинга уходит 1-2 минуты времени.

Задача испытуемого состоит в точном повторении шаблона текста, высвечиваемого на экране. При этом он может ошибаться, стирать символы, но для удачного завершения очередного эксперимента необходимо в точности повторить шаблон текста. Для защиты от случайной или специальной затяжки времени устанавливается лимит прерывающий эксперимент, который в этом случае необходимо начать заново. Работа считается законченной тогда, когда выполнены все N экспериментов.

Количество экспериментов (этапов) принято равным  $N = 40$  (содержимое этапов одинаковое для всех испытуемых). Результаты экспериментов заносятся в таблицу с полями: тип операции (d – нажатие клавиши / u – отжатие клавиши), код клавиши, время операции (до миллисекунды). Количество исходных записей для одного испытуемого на основе 40 этапов набора текста оценочно составляет 15000 (плюс-минус 3000) записей, в зависимости от того насколько точно он повторяет шаблонный текст без исправлений.

Приложение для снятия клавиатурного почерка разработано нами ранее и его работа описана в [Мустафаев, 2023].

В качестве модели данных для формирования окончательной таблицы примеров (датасета) принята модель на основе трех признаков – временных разностей  $t_1, t_2, t_3$  (в миллисекундах) между четырьмя ближайшими событиями из исходной базы данных. География клавиш (на основе кодов и их расположения) не учитывается, поскольку она не улучшает качества прогноза, что установлено в предыдущих экспериментах. Условия для принятия примера в датасет следующие:

1. Все четыре события должны быть исключительно от 32 клавиш кириллицы (кроме «ё») либо от клавиши точки «.», поскольку именно последовательность буквенных клавиш составляет основу уникальности тайпинга. Можно было бы добавить клавишу «пробела», которая часто используется, но предыдущие наши эксперименты показали, что такой вариант не улучшает прогнозные свойства.

2. Временной интервал между четырьмя событиями не должен превышать 2 секунды. Суть этого условия в том, чтобы исключить цепочки действий, в которых пользователь мог отвлекаться. Другие возможные значения этого интервала пока не исследованы.

3. В [Мустафаев, 2024] нами установлен факт того, что вся уникальная информация о динамике набора текста для субъекта (для четырех событий) содержится в последовательности событий вида: d(нажатие)-u(отжатие)-d(нажатие)-u(отжатие). Остальных 15 вариантов последовательностей событий сравнительно немного, и они отбрасываются, как неинформативные.

Последний параметр результирующего датасета с примерами – sf отвечает за тип выборки: 0 – обучающая, 1 – тестовая № 1, 2 – тестовая № 2. Обучающая выборка примеров генерируется на основе экспериментов с 1 по 38, 39-й эксперимент – основа для тестовой выборки №1 и 40-й эксперимент – для тестовой выборки №2.

В качестве модели машинного обучения для решения задачи на данном этапе исследований принята модель градиентного бустинга, реализуемая с помощью библиотеки CatBoost (<https://catboost.ai/>). Для табличных данных (всего с тремя признаками) такая модель вполне достаточна и эффективна.

Входными данными являются массив  $X$  векторов с 3-мя признаками (нормализованными в интервале  $[0, 1]$ ) и массив  $Y$  указаний учителя – номера классов. При прогнозировании класса через обученную модель пропускается пул тестовых примеров  $X$  в количестве  $n$ , полученных на основе набора текста одного из субъектов. На выходе снимается матрица  $R$  вероятностей принадлежности примеров одному из классов размерностью  $(n, 35)$ . Далее на основе матрицы  $R$  требуется определить номер класса субъекта. Для этого и необходим алгоритм идентификации.

Два алгоритма идентификации приведены нами в [Мустафаев, 2024], но, как показала практика, они недостаточно эффективны. В данной работе разработаны и исследованы три дополнительных алгоритма. Для исследования точности алгоритмов в среде Google Colaboratory разработан ноутбук с настройкой опций тестирования и визуализирующий результаты.

## **2. Алгоритмы идентификации субъекта клавиатурного почерка**

Приведем пять алгоритмов идентификации, обозначим их латинскими буквами «A», «B», «C», «D», «E».

Примем следующие обозначения:

- На вход подается матрица вероятностей от выходов модели  $R[n, m]$ , где  $n$  – количество примеров от испытуемого,  $m$  – число классов.
- Требуется идентифицировать испытуемого – найти номер класса  $S$ .

### **2.1 Алгоритм «A»**

Искомый класс идентифицируется по максимальному количеству примеров, которые могут принадлежать классу исходя из максимальной вероятности. В таком алгоритме классов-победителей может быть несколько, поскольку оценка целочисленная.

Для каждого  $i$ -го из  $n$  примеров, находим столбец  $j$  (номер класса) соответствующий максимальному значению вероятности и записываем номера классов в вектор  $P_{\max}[n]$ . На основе полученного вектора  $P_{\max}$  рассчитываем вектор  $L[m]$ , значение  $i$ -го элемента которого соответствует количеству найденных элементов со значением  $i$  в векторе  $P_{\max}$ . Результирующий номер класса  $S$  вычисляется как позиция  $i$  максимального элемента вектора  $L$ .

## 2.2. Алгоритм «В»

Находятся суммы вероятностей для всех примеров по классам, побеждает класс с максимальной суммой вероятности. Здесь критерий вещественный, поэтому шансы на то, что будет больше одного победителя с одной и той же суммарной вероятностью практически равны нулю.

Для входной матрицы  $R[n, m]$  находятся суммы столбцов и результат заносится в вектор  $P_{\text{sum}}[m]$ . Результирующий номер класса  $S$  вычисляется как позиция  $i$  максимального элемента вектора  $P_{\text{sum}}$ .

## 2.3. Алгоритм «С»

Начиная с этого алгоритма и далее вычисления проводятся в два этапа, поэтому можно их условно назвать двухэтапными. Здесь и далее вводится целочисленная переменная, которая может принимать значение от 2 и теоретически до  $m$  (количество классов), обозначим ее через  $w$ .

На первом этапе находится вектор  $P_{\text{sum}}$  как в алгоритме «В». Далее на основе вектора  $P_{\text{sum}}$  определяется список номеров классов  $F[w]$  с наибольшими вероятностями в количестве  $w$ . Назовем эти классы пулом победителей.

На втором этапе во входной матрице  $R[n, m]$  обнуляются столбцы классов, номера, которых отсутствуют в  $F[w]$ . Оставшиеся столбцы определяют победителя (или победителей при одинаковых количествах побед в примерах) по алгоритму «А», соревнуясь только в рамках пула победителя.

## 2.4. Алгоритм «D»

Первый этап аналогичен первому этапу в алгоритме «С» и определяет пул победителей  $F[w]$ .

На втором этапе на основе пула классов победителей генерируются соревновательные пары классов каждый с каждым. Например, если список  $F$  содержит классы  $[2, 5, 10]$ , то имеем три соревновательные пары (2 против 5, 2 против 10, 5 против 10). Для каждой соревновательной пары из входной матрицы  $R[n, m]$  выделяются два соответствующих столбца вероятностей и по алгоритму «А» выясняется победитель в этой конкретной паре. В копилку класса-победителя пары заносится 2 балла, проигравшему 0 баллов, если ничья (количество побед в примерах поровну), то оба класса получают по 1 баллу.

В конце алгоритма подсчитываются баллы, набранные классами на втором этапе в парных соревнованиях и определяется победитель по наибольшим баллам (или победители).

### **2.5. Алгоритм «Е»**

Этот алгоритм аналогичен алгоритму «D» за исключением подсчета баллов на втором этапе в парных соревнованиях. Подсчет баллов следующий: в копилку класса добавляются баллы по количеству примеров, в которых класс победил и отнимаются баллы по количеству примеров, в которых класс проиграл.


## **3. Оценка эффективности алгоритмов идентификации**


На основе разработанных алгоритмов реализованы программные коды, а также функции отчета результатов идентификации на тестовых примерах.

Имеется два набора тестовых примеров «тест-1» и «тест-2», в каждом наборе представлено по одному пулу примеров для каждого класса. Таким образом, в двух тестах содержатся 70 пулов примеров, по 2 на каждый класс. Точность прогноза определяется как частное от деления количества правильно идентифицированных классов на 70. Если при идентификации победителей больше одного, то считаем, что класс не распознан.

Кроме основного обучающего массива данных (True\_Train) с количеством примеров 82073, с помощью библиотеки `tabgan` (<https://pypi.org/project/tabgan/>), сгенерированы массивы синтетических обучающих данных на основе генератора `OriginalGenerator` (`OriginalGenerator_Train`) с количеством примеров 245075 и генератора `GANGenerator` (`GANGenerator_Train`) с количеством примеров 117787, которые также участвуют в эксперименте. На основе трех массивов обучены три модели типа `CatBoostClassifier`. Обращаем внимание, что обучение необходимо выполнить в режиме CPU (без видеоускорителя), поскольку установлено, что в режиме GPU (с видеоускорителем) результаты ухудшаются. Данный факт не связан с особенностями решаемой задачи, причина его пока не ясна, у других разработчиков происходит тоже самое, что подтверждается форумом (<https://github.com/catboost/catboost/issues/1408>).

Для выбора опций тестирования разработан набор виджетов, в котором выбирается модель, алгоритм идентификации, тестовый набор и количество победителей первого этапа (параметр `w` для алгоритмов «C», «D», «E») (рис. 1).


Показать код



Модель:

True\_Train

▼

Алгоритм:

E

▼

Тест:

тест-2

▼

Кол-во поб.:

3

▼

Выполнить тест

Рис. 1. Набор опций виджетов для тестирования в блокноте Google Colaboratory

В табл. 1 сведены результаты тестирования всех моделей, алгоритмов и тестовых наборов при  $w = 3$ . Такое значение  $w$  выбрано исходя из того, что отчет о тестировании по сумме вероятностей на алгоритме «В» для одного из нераспознанных классов показал, что по значению суммы вероятностей этот класс вошел в тройку, а остальные нераспознанные классы вошли в двойку лучших. Таким образом, чтобы не потерять искомый класс в алгоритмах «С», «D», «Е» минимальное значение  $w$  для нашего случая требуется взять равным 3.

Таблица 1

Результаты тестирования алгоритмов (при  $w = 3$ )

Алгоритм / тест	Модель True_Train		Модель OriginalGenerator		Модель GANGenerator	
	Нераспоз. классы	Точн.	Нераспоз. классы	Точн.	Нераспоз. классы	Точн.
«А» тест-1	2, 11, 14, 28, 29, 34	0,814	2, 14, 28, 29, 34	0,871	2, 9, 11, 14, 28, 29, 34	0,757
«А» тест-2	1, 4, 9, 25, 29, 33, 34		4, 9, 33, 34		1, 2, 4, 9, 10, 11, 18, 29, 33, 34	
«В» тест-1	2	0,938	-	0,943	14	0,914
«В» тест-2	4, 9, 33, 34		4, 9, 33, 34		2, 9, 11, 33, 34	
«С» тест-1	-	0,943	-	0,971	2, 29	0,900
«С» тест-2	9, 31, 33, 34		9, 33		1, 2, 9, 10, 33	
«D» тест-1	-	0,971	-	0,985	11, 29	0,943
«D» тест-2	9, 34		9		10, 22	
«Е» тест-1	-	0,985	-	0,985	29	0,943
«Е» тест-2	9		9		10, 22, 29	



Результаты тестирования при  $w=3$  показывают, что алгоритм «Е» при всех вариантах обучения показывает лучшие результаты, которые состоят в том, что из 70 пулов примеров нераспознанным остается только один. Также видно, что модель, обученная на датасете GANGenerator существенно уступает другим двум моделям, из чего следует, что данный генератор не совсем подходит к решению текущей задачи.

В табл. 2 представлены результаты экспериментов для  $w$  меняющегося от 3 до 7 для алгоритма «Е» (который признан лучшим из табл. 1). Они показывают, что при  $4 \leq w \leq 7$  модель, обученная на датасете OriginalGenerator дает 100% результат в распознавании классов.

Таблица 2

Результаты тестирования алгоритма «Е» (при  $w = [3 \dots 8]$ )

w / тест	Модель True_Train		Модель OriginalGenerator		Модель GANGenerator	
	Нераспоз. классы	Точн.	Нераспоз. классы	Точн.	Нераспоз. классы	Точн.
3 / тест-1	-	0,985	-	0,985	29	0,943
3 / тест-2	9		9		10, 22, 29	
4 / тест-1	-	0,985	-	1,000	29	0,957
4 / тест-2	31		-		22, 29	
5 / тест-1	-	0,985	-	1,000	29	0,957
5 / тест-2	31		-		22, 29	
6 / тест-1	-	0,971	-	1,000	29	0,957
6 / тест-2	22, 31		-		22, 29	
7 / тест-1	-	0,971	-	1,000	29	0,943
7 / тест-2	22, 31		-		3, 22, 29	
8 / тест-1	-	0,971	-	0,985	29	0,914
8 / тест-2	22, 31		22		1, 3, 22, 29, 31	

## Заключение

В итоге, можно сделать вывод о том, что максимальная эффективность в решении задачи идентификации клавиатурного почерка машинным обучением достигается применением алгоритма «Е» и генерации обучающих примеров модели с помощью генератора OriginalGenerator библиотеки tabgan. При этом требует дополнительного изучения вопрос как подбирать оптимальный параметр  $w$  и как он зависит от количества классов в системе, что планируется сделать в дальнейшем.

На данный момент экспериментальная база насчитывает 35 субъектов исследования (база пополняется и далее будет расширена). Это не много, чтобы быть полностью уверенными в надежности разработанного подхода с точки зрения практического использования, но и не мало, поскольку в других подобных исследованиях их в среднем в интервале 50-150. Кроме

того, следует отметить, что даже при 35 классах, без рассмотренных оригинальных алгоритмов точность распознавания на тестовой выборке (без синтетического обогащения) составляет 93,8%, а с алгоритмом «Е» 98,5%, эта разница очень существенна и при переборе моделей ИИ с гиперпараметрами такую разницу не получить.

При практическом применении всех текущих наработок для построения реальной системы идентификации личности по клавиатурному почерку нужно учитывать факт того, что данный вид биометрии основан на приобретенной координации движений пальцев, которая со временем может меняться, поэтому в реальной системе необходимо выполнять переобучение модели на основе новых актуальных данных тайпинга. Возможно также внедрить систему механизм согласно которому система каждый раз при идентификации личности будет давать оценку на сколько легко распознается тайпинг субъекта и давать рекомендации по переобучению, если его почерк становится все больше не похож на оригинал.

Влияние на клавиатурный почерк психофизического состояния оценить сложно, возникает вопросы как оценить сиюминутного состояние человека при тайпинге или как искусственно ввести его в как-либо состояние для сбора исходного материала исследования? В данном случае мы исходим из того, что клавиатурный почерк это приобретенный навык, связанный с координацией, который выполняется на бессознательном уровне подобно управлению автомобилем, а также из того, что в процессе идентификации пользователь просто отложит процедуру в случаях нестандартных ситуаций, влияющих на его психофизическое состояние.

### Список литературы

- [Варламова, 2023] Варламова С.А., Вавилина Е.А. Идентификация пользователя на основе клавиатурного почерка // Инновационное приборостроение. – 2023. – Т. 2, № 3. – С. 67-71. – doi: 10.31799/2949-0693-2023-3-67-71.
- [Лунев, 2024] Свидетельство о государственной регистрации программы для ЭВМ № 2024684582 Российская Федерация. Программа анализа клавиатурного почерка KBoardTrack: № 2024683137: заявл. 08.10.2024 : опубл. 18.10.2024 / А.А. Уксусова, Д.Л. Перчаткин, А.К. Хамитов; заявитель ООО "Системы нейробезопасности".
- [Мустафаев, 2023] Мустафаев А.Г., Кобзаренко Д.Н., Паштаев Б.Д. Нейросетевая модель идентификации клавиатурного почерка: датасет, архитектура, метрика // Промышленные АСУ и контроллеры. – 2023. – № 10. – С. 34-41. – doi: 10.25791/asu.10.2023.1466.
- [Мустафаев, 2024] Мустафаев А.Г., Кобзаренко Д.Н., Шихсаидов Б.И. Способы повышения точности в задаче идентификации клавиатурного почерка нейросетевой моделью // Промышленные АСУ и контроллеры. – 2024. – № 5. – С. 3-13.
- [Сатыбалдиева, 2024] Сатыбалдиева М.М. Исследование систем для идентификации пользователя на основе анализа клавиатурного почерка // Научный аспект. – 2024. – Т. 14, № 5. – С. 1897-1903.

- [Сулавко, 2020] Сулавко А.Е. Абстрактная модель искусственной иммунной сети на основе комитета классификаторов и ее использование для распознавания образов клавиатурного почерка // Компьютерная оптика. – 2020. – Т. 44, № 5. – С. 830-842. – doi: 10.18287/2412-6179-CO-717.
- [Acien et.al., 2021] Acien A., Morales A., Monaco J.V., Vera-Rodriguez R. TypeNet: Deep Learning Keystroke Biometrics. JOURNAL OF LATEX CLASS FILES. – 2021. – Vol. 14, No. 8. – <https://doi.org/10.48550/arXiv.2101.05570>.
- [Sharma et al., 2023] Sharma A., Jureček M., Stamp M. Keystroke Dynamics for User Identification // Computer Science (Machine Learning). – 2023. – <https://doi.org/10.48550/arXiv.2307.05529> .
- [Wang et al., 2023] Wang X., Shi Y., Zheng K., Zhang Y., Hong W., Cao S. User Authentication Method Based on Keystroke Dynamics and Mouse Dynamics with Scene-Irrelated Features in Hybrid Scenes // Sensors. – 2022, – Vol. 22(16), 6627. – <https://doi.org/10.3390/s22176627>.