ВЛИЯНИЕ ЭВОЛЮЦИИ ЦИФРОВЫХ ОТПЕЧАТКОВ УСТРОЙСТВ НА ДОСТОВЕРНОСТЬ ИДЕНТИФИКАЦИИ АНОНИМНЫХ ПОЛЬЗОВАТЕЛЕЙ

Шелухин О.И.[1], Ванюшина А.В.[2], Большаков А.С.[3], Желнов М.С.[4]

Цель исследования: *оценка эффективности программной идентификации анонимных пользователей в условиях эволюции цифровых отпечатков их устройств .*

Методы. *Технологии искусственного интеллекта включающие в себя обработку текста на естественных языках NLP(Natural Language Processing), методы латентно-семантического анализа LSA (Latent semantic analysis) , а также методы кластеризации и машинного обучения.*

Объектами исследования *являются теоретические и практические вопросы решения и визуализации задач информационной безопасности .*

Полученные результаты. *Для исследования влияния эволюции цифровых отпечатков анализируемых устройств, путем поочередного изменения анализируемых параметров оригинального фингерпринта (ФП) (fingerprint — цифровой отпечаток браузера или цифрового устройства) создана база модифицированных ФП. Предложена методика расчета и представлены численные результаты оценки вероятности правильной и ложной идентификации пользователя при эволюции атрибутов его цифровых отпечатков. Показана зависимость эффективности деанонимизации пользователя в зависимости от характеристик и свойств изменяемых атрибутов цифровых отпечатков его устройств.*

Область применения предложенного подхода – повышение эффективности систем идентификации анонимных пользователей на основе анализа цифровых отпечатков устройств.

Предлагаемая статья будет полезна как специалистам, разрабатывающим системы защиты информации, так и студентам, обучающимся по направлению подготовки «Информационная безопасность».

Ключевые слова: *фингерпринт, модифицированная база данных, набор данных, текстовые данные, категориальные данные, признаки, технологии искусственного интеллекта.*

DOI:10.21681/2311-3456-2022-2-72-86

Введение

Для совершения безнаказанного преступления в киберпространстве и сокрытия следов своих преступлений (несанкционированного доступа и кражи данных, подделки платежных реквизитов, нарушения авторских прав, атак, направленных на отказ в обслуживании и т.д.) нарушители активно используют методы анонимизации.

В связи с нарастающими проблемами организации преступных группировок и террористических актов в отечественное законодательство, нормативно-правовые акты были внесены различные поправки, касающиеся анонимности сетевых пользователей, имеющие непосредственное отношение к деанонимизации и информационной безопасности в Интернете [5,6].

Одним из возможных и вполне надежных способов деанонимизации является формирование цифровых отпечатков браузера – уникальных значений, отражающих настройки web-обозревателя пользователя [1,2].

5 Федеральный закон «О внесении изменений в Федеральный закон «О противодействии терроризму» и отдельные законодательные акты Российской Федерации в части установления дополнительных мер противодействия терроризму и обеспечения общественной безопасности» от 06.07.2016 N 374-ФЗ.

6 Федеральный закон «О внесении изменений в Федеральный закон “Об информации, информационных технологиях и о защите информации”» от 29.07.2017 N 276-ФЗ

1 Шелухин Олег Иванович, доктор технических наук, профессор, заведующий кафедрой «Информационная безопасность», Московский Технический Университет Связи и Информатики (МТУСИ), Москва, Россия. E-mail: sheluhin@mail.ru

2 Ванюшина Анна Вячеславовна, кандидат технических наук, доцент кафедры «Информационная безопасность», Московский Технический Университет Связи и Информатики (МТУСИ), Москва, Россия. E-mail: a.v.vaniushina@mtuci.ru

3 Большаков Александр Сергеевич, кандидат технических наук, доцент кафедры «Информационная безопасность», Московский Технический Университет Связи и Информатики (МТУСИ), Москва, Россия. E-mail: as.bolshakov57@mail.ru

4 Желнов Максим Сергеевич, студент магистратуры, Московский Технический Университет Связи и Информатики (МТУСИ), Москва, Россия. E-mail: max306211@yandex.ru

Существует большое количество сервисов, позволяющих осуществлять идентификацию пользователей на основании сведений, получаемых об их браузерах [3.4,5]. Базовая информация о веб-браузере уже давно собирается веб-аналитическими службами с целью точного измерения реального веб-трафика и фильтрации автоматически созданных запросов.

Некоторые из сервисов предназначены для формирования цифрового отпечатка браузера, представляемого в виде некоторого хеш-значения, на основании полученных данных о браузере [6,7,8]. Другие сервисы выводят собранные о браузере пользователя сведения, производят различные вычисления на основании уже имеющихся данных и предоставляют пользователю информацию об уникальности его браузера и, как следствие, возможности его идентификации при высоких показателях уникальности.

С помощью простого сценария, выполняемого внутри браузера, сервер может собирать широкий спектр информации из публичных интерфейсов, называемых интерфейсом прикладного программирования (API, application programming interface) и заголовков HTTP. Идентификация посетителей web-ресурсов является значимой и важной задачей для отслеживания злоумышленников. Самый распространенный механизм уникальной идентификации пользователей — это использование отправленных cookie-файлов web-сервером [3,4].

Цифровой отпечаток может полностью или частично идентифицировать отдельных пользователей или устройства, даже когда файлы cookie и другие данные для отслеживания отключены или недоступны. Цифровые отпечатки устройства весьма полезны, поскольку с их помощью проще обнаруживать и предотвращать кражи личных данных, различные виды мошенничества

Механизм определения цифрового отпечатка устройства подразумевает, что при изменении пользователем браузера, его также можно будет идентифицировать.

Фингерпринтинг веб-браузера (FingerprintJS — 2) –методика отслеживания пользователей при помощи браузера — обеспечивает сбор данных о браузере пользователя, его системе и устройстве [7,8,9].

Фингерпринтинг собирает такую информацию, как версия браузера, версия ОС, расширения, часовой пояс, GPU и CPU, разрешение монитора/-ов и размер окна браузера, шрифты, плагины, и прочее стороннее ПО [9,10]. Разработчики браузеров, ученые и органы стандартизации долгое время пытаются бороться с этой проблемой отслеживания пользователей, разрабатывая защитные средства от web-браузера ФП, которые работают точечно и не мешают работе пользователя с браузером. Главная цель атакующего при web-браузер фингерпринтинге – узнать, какую страницу посещает пользователь и какие действия он совершает.

Основная сложность деанонимизации пользователей с помощью цифровых отпечатков браузера, связана с тем, что цифровые отпечатки в следствие

обновлений системы, плагинов, браузеров, установки различных программ, а с ними и шрифтов со временем изменяются [11].

Этот процесс называется эволюцией цифровых отпечатков устройств. Говоря о постоянстве исследуемых характеристик браузера, следует отметить, что многие из них подвержены изменениям с разной частотой.

Некоторые параметры могут меняться довольно часто. Например, разрешение экрана при подключении дополнительного монитора. Какие-то параметры меняются реже, к примеру, версия браузера. Есть параметры, которые меняются очень редко или не меняются вовсе. К таким параметрам относятся те, что содержат информацию об аппаратной составляющей устройства, с которого запускается исследуемый web-обозреватель. Как следствие, при сборе информации необходимо учитывать, как часто для среднестатистического пользователя будет меняться тот или иной параметр, и придавать больший вес тем параметрам, которые дольше остаются неизменными

Целью работы является оценка эффективности программной идентификации анонимных пользователей в условиях эволюции цифровых отпечатков устройств .

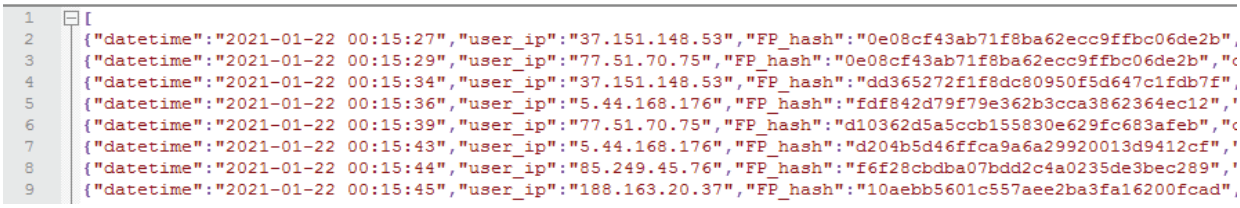
Формирование базы данных

Для выполнения задачи сбора информации (ФП устройств) будем использовать выделенный сервер с развернутым сайтом-одностраничником и внедренным в него скриптом сбора информации, а также базу данных MySQL, в которой будет храниться собираемая информация ФП устройств $\left\{\Phi\Pi_j\left(A_{i\,orig}\right), i=\overline{1,M};\ j=\overline{1,N}\right\}$ для последующего анализа. Здесь $j=\overline{1,N}$ объем экспериментально полученных ФП, каждый из которых характеризуется вектором атрибутов $A_{i,orig}=\left(A_{1\,orig}, A_{2\,orig}\ldots A_{i+1\,orig}\ldots A_{M\,orig}\right)$.

7 Fingerprintjs2 — modern flexible open-source browser fingerprinting library. http://valve.github.io/fingerprintjs2/.

8 Fingerprintjs2, modern and flexible browser fingerprinting library, a successor to the original fingerprintjs. https://github.com/Valve/fingerprintjs2

9 Security/Fingerprinting — Mozilla wiki. https://wiki. mozilla. org/Security/Fingerprinting. 2018



```
[
{"datetime":"2021-01-22 00:15:27","user_ip":"37.151.148.53","FP_hash":"0e08cf43ab71f8ba62ecc9ffbc06de2b",
{"datetime":"2021-01-22 00:15:29","user_ip":"77.51.70.75","FP_hash":"0e08cf43ab71f8ba62ecc9ffbc06de2b","
{"datetime":"2021-01-22 00:15:34","user_ip":"37.151.148.53","FP_hash":"dd365272f1f8dc80950f5d647c1fdb7f",
{"datetime":"2021-01-22 00:15:36","user_ip":"5.44.168.176","FP_hash":"fdf842d79f79e362b3cca3862364ec12","
{"datetime":"2021-01-22 00:15:39","user_ip":"77.51.70.75","FP_hash":"d10362d5a5ccb155830e629fc683afeb","
{"datetime":"2021-01-22 00:15:43","user_ip":"5.44.168.176","FP_hash":"d204b5d46ffca9a6a29920013d9412cf","
{"datetime":"2021-01-22 00:15:44","user_ip":"85.249.45.76","FP_hash":"f6f28cbdba07bdd2c4a0235de3bec289","
{"datetime":"2021-01-22 00:15:45","user_ip":"188.163.20.37","FP_hash":"10aebb5601c557aee2ba3fa16200fcad",
```


Рис. 1. Фрагмент экспортируемой БД формата JSON

За основу скрипта для сбора информации была взята модификация открытой программной библиотеки fingerprintjs2, представляющей собой открытую программную библиотеку JavaScript, которая может использоваться для извлечения уникальных характеристик браузера и технического устройства.

Универсальным решением для формирования базы данных (БД) является сочетание MySQL и языка программирования PHP.

Из соображений универсальности по отношению к обработчикам информации полученные данные из БД экспортировались в формат JSON (рис. 1).

Разработанный скрипт позволил зафиксировать 37 параметров, некоторые из которых разделяются на подпараметры. В результате был сформирован набор данных фингерпринтов $\left\{\Phi\Pi_j\left(A_{i\,orig}\right), i=\overline{1,M};\ j=\overline{1,N}\right\}$. Объемом записей об устройствах составлял N=6233, каждый из которых содержал M=70 атрибутов $A_{i,orig}=\left(A_{1\,orig}, A_{2\,orig}\ldots A_{i+1\,orig}\ldots A_{M\,orig}\right)$, как это показано в таблице 1.

На рис. 2 представлена гистограмма распределения анализируемых признаков по длине. По оси X показан номер признака в таблице 1, а по оси Y – его длина.

Как видно, наибольшей длиной в контексте рассматриваемых ФП являются параметры canvas, window_dump_types, style_dump, webgl.

Оценка важности параметров фингерпринта

Важность исходных атрибутов $\{A_{i,orig};\ ,i=\overline{1,M}\}$ фингерпринтов $\{\Phi\Pi_j\left(A_{i\,orig}\right);\ j=\overline{1,N}\}$ оценивалась с помощью библиотеки SelectKBest с применением функции chi2 [12], использующей статистические методы для отбора признаков. Процедура оценки важности chi2 параметров иллюстрируется на рис. 3.

Таблица 1

Исходный набор параметров фингерпринтов

i	Признак $A_{i,orig}$	Описание	i	Признак $A_{i,orig}$	Описание
1	user-agent	Характеристика клиентского приложения	37	document_dump_ functions	Список собственных функций и типов реализуемых глобально доступными объектами
2	language	Системный язык	38	document_ dump_types	
3	color_depth	Глубина цвета	39	style_dump	
4	device_memory	Объем ОЗУ	40	error_messages	Список ошибок браузера
5	hardware_concurrency	Многозадачность	41	silverlight_installed	Наличие SilverLight
6	resolution	Текущее разрешение экрана	42	silverlight_supported	Поддержка SilverLight
7	available_resolution	Доступное разрешение экрана	43	silverlight_versions	Версия SilverLight

i	Признак $A_{i,orig}$	Описание	i	Признак $A_{i,orig}$	Описание
8	timezone_offset	Сдвиг часового пояса	44	ActiveBorder	Фактическое значение RGB цветов CSS элементов, отображаемых экраном пользователя
9	session_storage	Наличие API HTML5 (хранение ключей/значений в браузере)	45	GrayText	
10	local_storage	Браузерное хранилище	46	ActiveCaption	
11	indexed_db	Наличие API HTML5 (постоянное хранение данных внутри браузера)	47	AppWorkspace	
12	open_database	Поддержка OpenDB	48	Background	
13	cpu_class	Разрядность ЦПУ	49	ButtonFace	
14	navigator_platform	Платформа устройства	50	ButtonHighlight	
15	regular_plugins	Список плагинов браузера	51	ButtonShadow	
16	canvas	Результат canvas изображения на странице	52	ButtonText	
17	webgl	Результат рендеринга 3D-графики	53	CaptionText	
18	webgl_vendor	Производитель/модель ГПУ	54	ThreeDShadow	
19	adblock	Наличие блокировщика рекламы	55	Highlight	
20	has_lied_languages	Факт ложного языка	56	HighlightText	
21	has_lied_resolution	Факт ложного разрешения экрана	57	InactiveBorder	
22	has_lied_os	Факт ложной ОС	58	InactiveCaption	
23	has_lied_browser	Факт ложного браузера	59	InactiveCaptionText	
24	touch_support	Поддержка тач-пада	60	InfoBackground	
25	js_fonts	Список установленных шрифтов JS	61	InfoText	
26	audio_fp	Результат обработки аудио	62	Menu	Фактическое значение RGB цветов CSS элементов, отображаемых экраном пользователя
27	activex_objects	Наличие ActiveXObject	63	MenuText	
28	ms_components	Наличие компонентов MS	64	Scrollbar	
29	navigator_dump_functions	Список собственных функций и типов реализуемых глобально доступными объектами	65	ThreeDDarkShadow	
30	navigator_dump_types		66	ThreeDFace	
31	toolbar_dump_functions		67	ThreeDHighlight	
32	toolbar_dump_types		68	ThreeDLightShadow	
33	crypto_dump_functions		69	Window	
34	crypto_dump_types		70	WindowFrame	
35	window_dump_functions		71	WindowText	
36	window_dump_types				

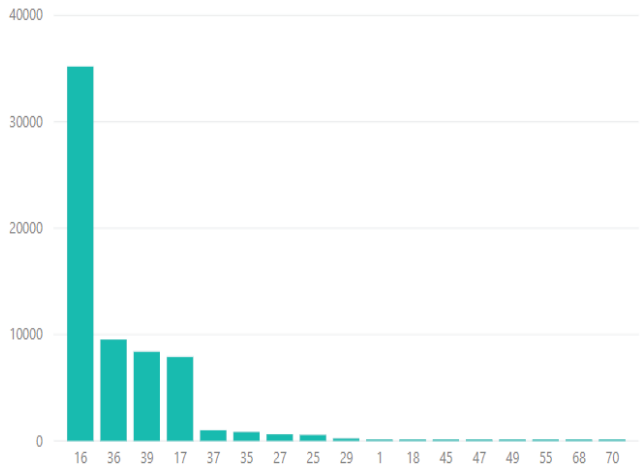


Рис.2. Гистограмма распределения признаков данных ФП по длине

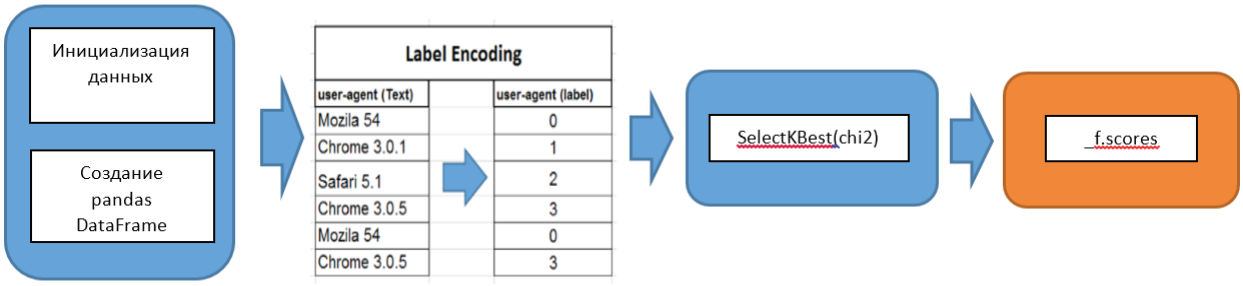


Рис. 3. Схема оценки важности chi2 параметров

Функция chi2 предусматривает вычисление характеристики хи-квадрат для каждого признака и показывает зависимость между значениями признака и классом и вычисляется по формуле:

$$x^2 = \sum_{c=1}^{C}\sum_{f=1}^{F}\frac{(O_{cf} - E_{cf})^2}{E_{cf}},$$

где C – количество классов, F – количество значений признака, O_{cf} и E_{cf} – наблюдаемая и ожидаемая частоты встречи признака f со значением в классе c. Ожидаемая частота вычисляется как вероятность двух независимых событий:

$E_{ij} = N * P(i = c \cap f) = N * (P(c) * P(f))$,

где N – количество всех записей в наборе данных, $P(c)$ и $P(f)$ – вероятности наличия записи с меткой класса или со значением признака f среди всех записей.

Набор анализируемых атрибутов может содержать как текстовые, так и категориальные значения (в основном не числовые), например такие параметры, как *user-agent, webgl, canvas* и т.д. Поэтому требуется предварительно произвести кодирование меток классов (LabelEncoding).

Результаты анализа важности атрибутов представлены в виде гистограммы на рис. 4.

Среди наиболее важных атрибутов можно выделить user-agent, webgl_vendor

language, webgl, adblock. Условно, по степени важности анализируемый набор может быть разделен на три группы:

1) высокая — параметры характеризуемые величиной chi2>100 ;
2) средняя — параметры значения chi2 которых лежат в интервале 100 >chi2>20;
3) низкая — параметры характеризуемые величиной chi2 <20.

Наиболее значимыми в контексте рассматриваемых ФП являются параметры со степенью важности “Высокая” – user-agent, webgl, language, canvas и т.д. (табл.1).

Эволюции цифровых отпечатков

Для исследования влияния эволюции цифровых отпечатков устройств на идентификацию анонимных пользователей из базы данных была случайным образом выбрана запись ФП. Путем незначительного по-

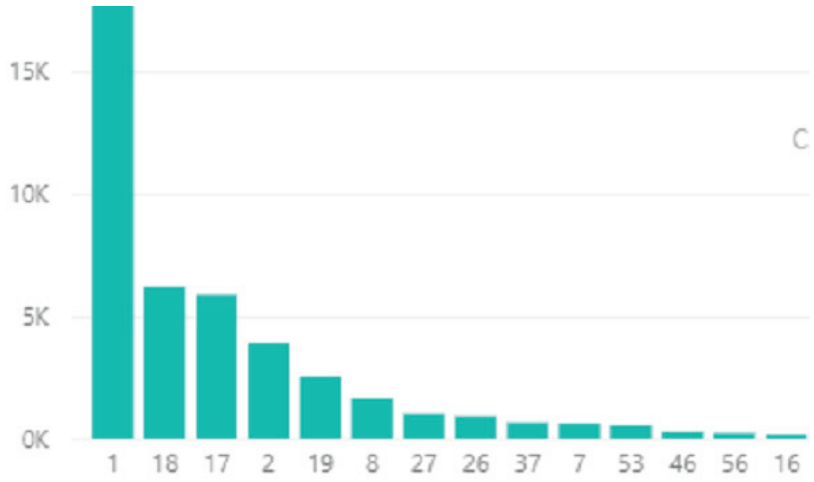


Рис. 4. Гистограмма распределения признаков по важности

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	user-agent	language	color_depth	device_memory	lware_concurr	resolution	available_resolution	timezone_offset	session_storage	local_storage	ndexed_d	en_databa	cpu_class	gator_platf
53	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	-180	1	1	1	1	unknown	Win32
54	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	-180	1	1	1	1	unknown	Win32
55	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	-180	1	1	1	1	unknown	Win32
56	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	-180	1	1	1	1	unknown	Win32
57	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	-180	1	1	1	1	unknown	Win32
58	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	-180	1	1	1	1	unknown	Win32
59	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	-180	1	1	1	1	unknown	Win326x3
60	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	-180	1	1	1	1	unknown	Win32
61	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	-180	1	1	1	5	unknown	Win32
62	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	-180	1	1	7	1	unknown	Win32
63	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	-180	1	5	1	1	unknown	Win32
64	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	-180	8	1	1	1	unknown	Win32
65	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	153	1	1	1	1	unknown	Win32
66	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768zYG	-180	1	1	1	1	unknown	Win32
67	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768@эМф1	1298|768	-180	1	1	1	1	unknown	Win32
68	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	6	1360|768	1298|768	-180	1	1	1	1	unknown	Win32
69	Mozilla/5.0 (Windows NT 6.	ru-RU	24	1	4	1360|768	1298|768	-180	1	1	1	1	unknown	Win32
70	Mozilla/5.0 (Windows NT 6.	ru-RU	54	8	4	1360|768	1298|768	-180	1	1	1	1	unknown	Win32
71	Mozilla/5.0 (Windows NT 6.	ru-RUьыяэ	24	8	4	1360|768	1298|768	-180	1	1	1	1	unknown	Win32
72	Mozilla/5.0 (Windows NT 6.	ru-RU	24	8	4	1360|768	1298|768	-180	1	1	1	1	unknown	Win32

Рис. 5 . Фрагмент тестового датафрейма модифицированного ФП

очередного изменения каждого параметра выбранного ФП сформирован тестовый набор модифицированных ФП и записан в базу данных.

На рис. 5 представлен фрагмент полученного тестового датафрейма с отмеченными измененными параметрами.

Процедура получения модифицированных фингепринтов заключается в замене каждого из 70 параметров исходного (оригинального) ФП$\{$ФПj $(A_{i\,orig})$, $i=\overline{1,M};\ j=\overline{1,N}\ \}$ на значение соответствующего параметра из базы данных. В результате для рассматриваемого j-го фингерпринта формируется модифицированная БД вида $\{$ФПj $(A^{*}_{i\,mod})$, ,$i=\overline{1,M};\ j=\overline{1,N}\}$, в которой значения j-го параметра заменяются (модифицируются) на отличные от исходного (оригинального) значения.

Здесь $A^{*}_{i\,mod}=\left(A_{1\,orig}, A_{2\,orig}\ldots\ \boldsymbol{A_{i\,mod}}\ \ldots A_{i+1\,orig}\ldots A_{M\,orig}\right)$.

Учитывая, что рассматриваемый набор данных может содержать текстовые или категориальные значения (в основном нечисловые значения), например такие параметры, как *user-agent, webgl, canvas* и т.д. полученные строки ФП рассматривались как некоторые тексты.

Для выявления характерных факторов (тематик), присущих всем документам и терминам встречающихся в ФП использован латентный семантический анализ LSA (англ. Latent semantic analysis) [15,16] анализирующий взаимосвязь между библиотекой документов и встречающимися в них терминами.

На этапе идентификации реализовывался алгоритм кластеризации, ориентированный на группировании данных по схожести, с помощью косинусного расстояния между выборками данных [13,14].

После создания 70 модифицированных ФП с помощью модели LSA были сформированы векторные представления модифицированных и оригинального ФП с различным количеством скрытых тем (КСТ). Использовались модели с КСТ = 70; 100; 125; 150;

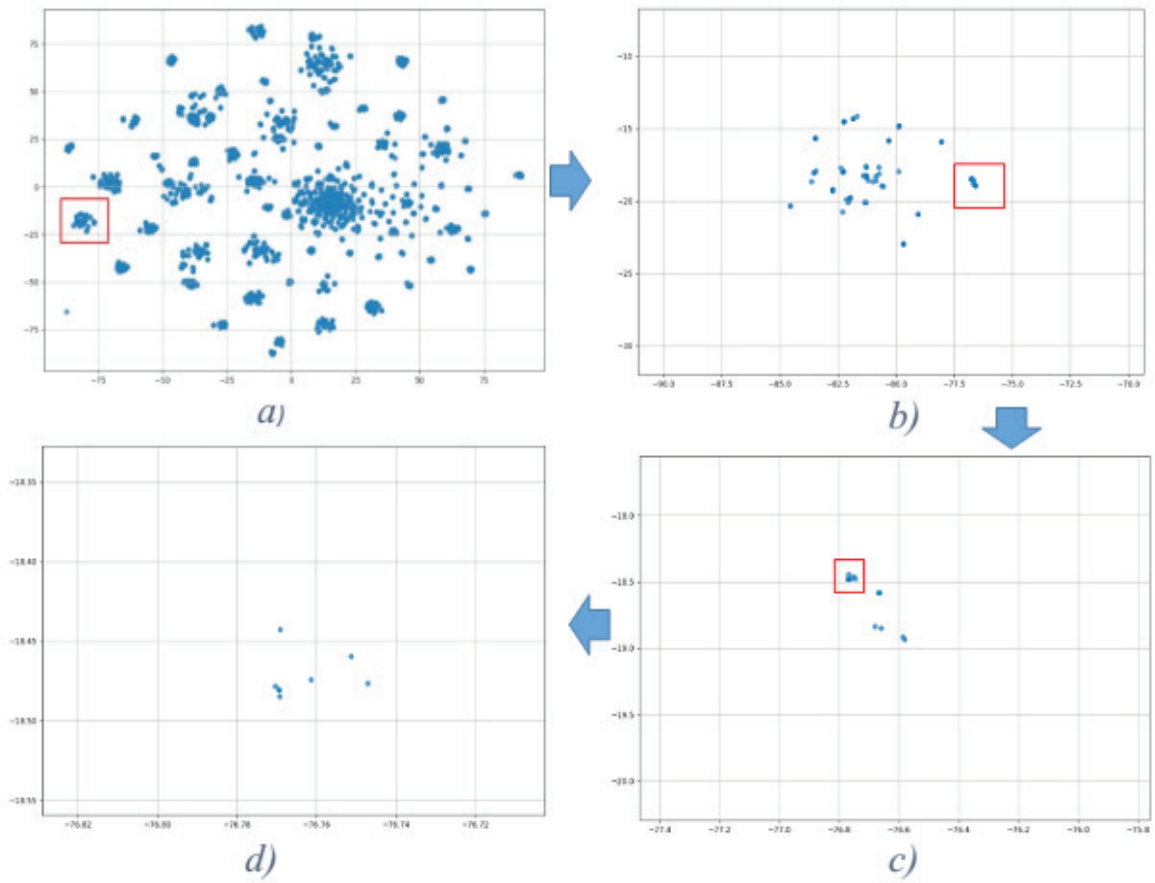


Рис. 6. Визуализация ФП согласно обученной модели LSA (КСТ = 70)

200; 500; 1000 и 2000. После векторизации модифицированных ФП выполнялся пересчет дистанций между оригинальным и модифицированными ФП с использованием библиотеки scipy функции distance. cosine [12,13], а полученные результаты фиксировались. На рис. 6 представлена визуализация ФП относительно скрытых семантических тем с использованием алгоритма стохастического вложения соседей с t-рас-пределением t-SNE, предназначенного для визуализации данных в пространство низкой размерности (двух- или трехмерного) [17,18,19].

В результате обработки каждая скрытая тема может быть представлена в виде самостоятельного кластера. На рис. 6а– красным квадратом выделен один из кластеров относящийся к n-теме, к которой относится группа ФП устройства. При масштабировании рассматриваемого кластера, видно скопление ФП (Рис. 6b). При дальнейшем масштабировании групп ФП внутри рассматриваемого кластера (Рис. 6c – 6d) можно заметить, что данные группы имеют сходство между собой относительно n-темы. Количество тем оценивалось идентификатором num_topics.

Для измерения схожести между текстами *будем использовать косинусную меру (дистанцию).* Основным преимуществом косинусного расстояния является то, что данная метрика хорошо работает на разреженных данных, поскольку реальные тексты ключевых фраз могут быть очень длинными и содержать значительные объемы служебной информации (минус-слова, стоп-слова и т. д.) Ключевым недостатком косинусного расстояния является его зависимость от форм слова.

При нахождении минимального косинусного расстояния использовался метод distance.cosine библиотеки sklearn (рис. 7).

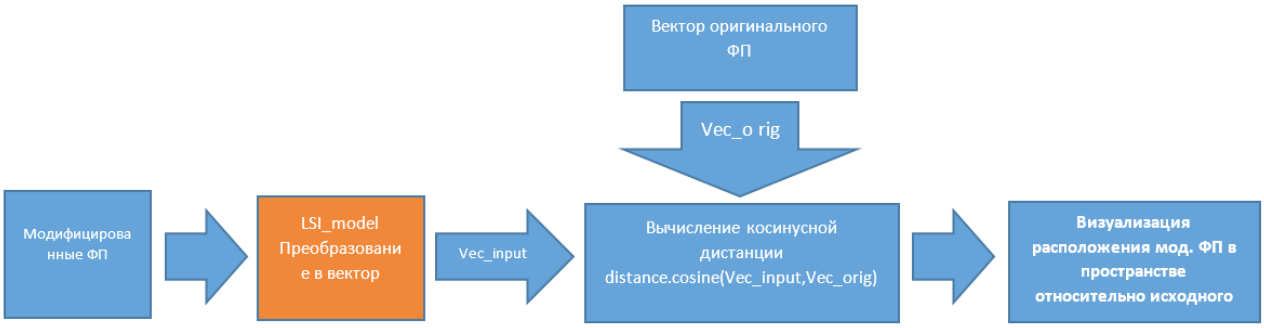


Рис. 7. Схема нахождения наиболее похожего фингерпринта и косинусного расстояния с использованием LSA

Результаты оценки достоверности идентификации пользователя при эволюции цифровых отпечатков пользователя

Для оценки точности идентификации ФП требуется оценить количество вхождений модифицированных $\left\{\Phi\Pi_j\left(A^*_{i\,mod}\right),\ i=\overline{1,70};\ j=1,N\right\}$ в пространственную область заданного порогового радиуса Rпор, как это показано на рис. 8. Вхождение в пороговую область оценивается косинусным расстоянии модифицированных $\left\{\Phi\Pi_j\left(A^*_{i\,mod}\right),\ i=\overline{1,70};\ j=\overline{1,N}\right\}$ от оригинального ФП $\left\{\Phi\Pi_j\left(A_{i\,orig}\right),\ j=\overline{1,N}\right\}$.

Искомая дистанция оценивается выражением [14,15]

$$dist\left(A_j,B_j\right)=1-cos\theta=1-\frac{A*B}{AB}=\frac{\sum_{i=1}^{M}A_{ji}B_{ji}}{\sqrt{\sum_{i=1}^{M}(A_{ji})^2}\sqrt{\sum_{i=1}^{M}(B_{ji})^2}};$$

где $A_{ji}=\left\{\Phi\Pi_j\left(A_{i\,orig}\right),i=\overline{1,M};\ j=\overline{1,N}\right\}$;

$B_{ji}=\left\{\Phi\Pi_j\left(A^*_{i\,mod}\right),i=\overline{1,M};\ j=\overline{1,N}\right\}$.

Поскольку значение $cos(\theta)$ находится в диапазоне [−1,1], то значение косинусной дистанции равное 1 указывает на отсутствие сходства, а 0 — на высокое сходство между векторами. Графическое представление оригинальных и модифицированных ФП, а также границы пороговых областей для различных КСТ приведены на рис. 9...11.

На рис. 9 изображено поэтапное уменьшение порогового уровня Рпф и зависящие от этого уровня вероятности правильной Рпф и ложной Рлф фиксации. На рис. 9с изображены модифицированные ФП с максимальным отклонением дистанции с указанием изменяемого в них атрибута, оказывающего наибольшее влияние на изменение косинусной дистанции. В данном случае это атрибуты: «canvas» — dist = 0,0031 ; «webgl» — dist = 0,0007; «activex_objects» — dist = 0,00034.

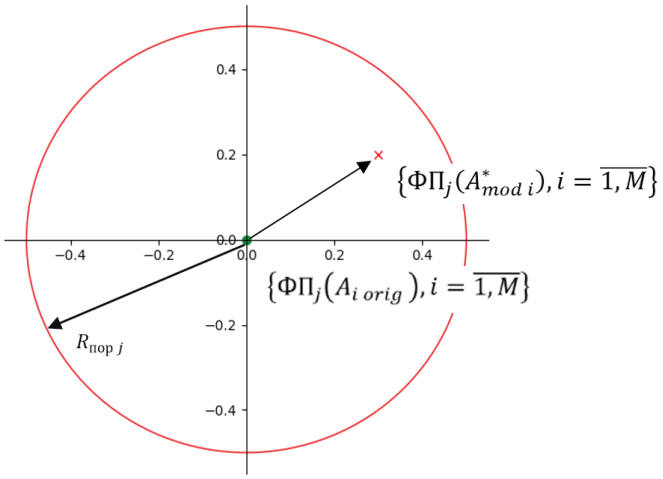


Рис. 8. Идентификация при модификации i-го атрибута ФП пользователя

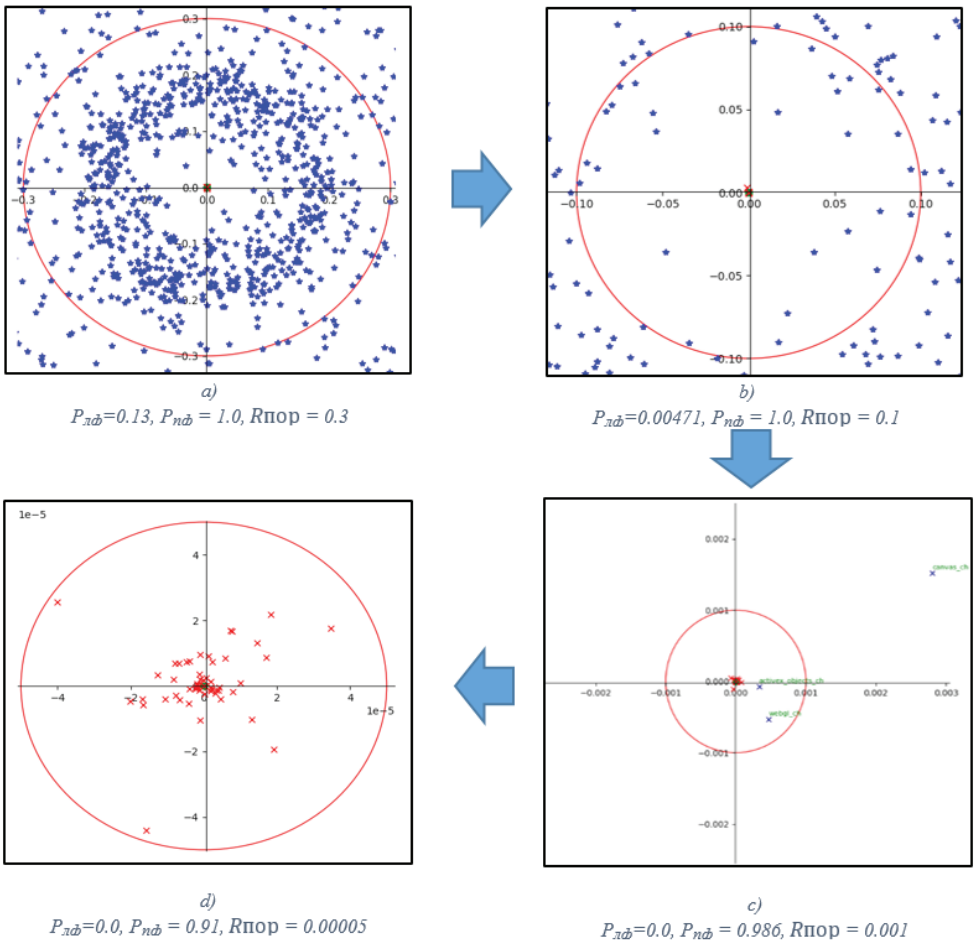


Рис. 9. Расположение оригинальных и модифицированных векторов ФП; границы порогов (КСТ =25), х – модифицированные; ФП, ● – центр оригинальный ФП

На рис. 10 изображено поэтапное уменьшение порогового уровня Рпф и зависящие от этого уровня вероятности правильной Рпф и ложной фиксации Рлф. На рис.10с изображены модифицированные ФП с максимальным отклонением дистанции с указанием изменяемых в них атрибутов, оказывающих наибольшее влияние на изменение косинусной дистанции. В данном случае это атрибуты: «canvas» — dist = 0,0075 ; «webgl» — dist = 0,0028; «activex_objects» — dist = 0,00031

На рис. 11 изображено поэтапное уменьшение порогового уровня *Рлф* и зависящие от этого уровня вероятности правильной Рпф и ложной фиксации *Рлф*. На рис.11с изображены модифицированные ФП с максимальным отклонением дистанции с указанием изменяемого в них атрибута, оказывающего наибольшее влияние на изменение косинусной дистанции. В данном случае это атрибуты «canvas» — dist = 0,127; «webgl» — dist = 0,012; « InactiveCaption» — dist = 0,009.

Анализ полученных результатов показывает, что при изменении наиболее важных параметров ФП — user-agent, webgl, language, canvas , либо атрибутов их наибольшую длину — canvas ; webgl; InactiveCaption, модифицированные ФП могут быть достаточно эффективно идентифицированы.

С учетом механизма идентификации пользователя при модификации отдельных атрибутов его ФП, вероятность правильной фиксации может быть оценена выражением

$$P_{\text{пф}} = \frac{1}{M}\sum_{i=1}^{M} Z_{ij},$$

где $Z_{ij} = ind\ dist\left(\Phi\Pi_j\left(A^*_{i\,mod}\right), \Phi\Pi_j\left(A_{i\,orig}\right)\right);\ j = \overline{1,N};\ i = \overline{1,M}$

$$ind\ \ (\cdot) = \begin{cases} 1 \text{ при } dist(\cdot) \le R_{\text{пор } j} \\ 0 \text{ при } dist(\cdot) > R_{\text{пор } j} \end{cases};$$

Соответственно вероятность ложной идентификации j-го пользователя при попадании в пороговую

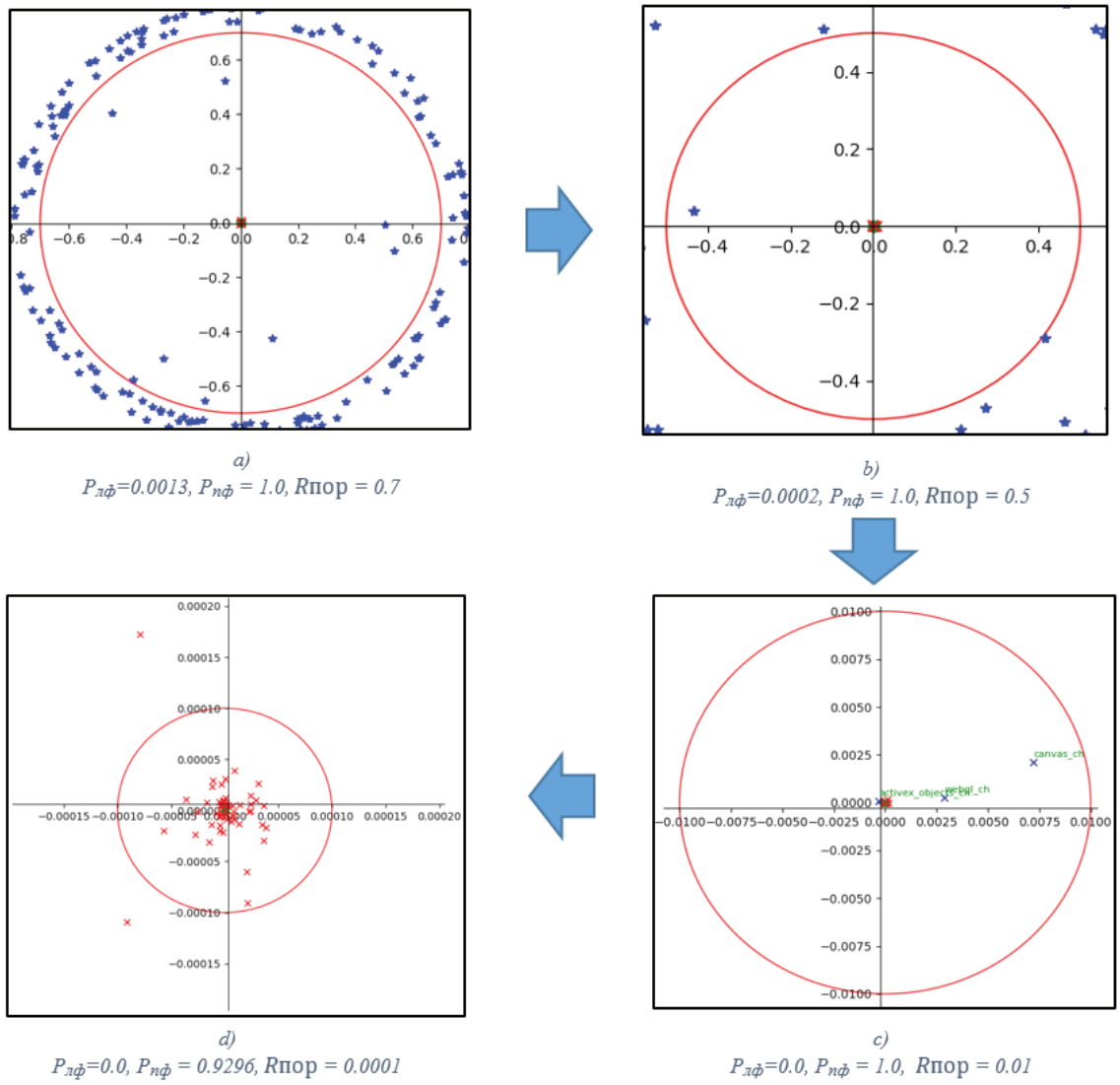


Рис.10. Расположение оригинальных и модифицированных векторов ФП; границы порогов (КСТ =100) где: х – модифицированные ФП, ● – центр оригинальный ФП

область других ФП $j=\overline{1,N};\ ;k\neq j$ пользователей можно оценить выражением

$$P_{\text{пф}}=\frac{1}{N}\sum_{\substack{i=1\\k\neq j}}^{N}K_{kj},$$

где $K_{kj}=ind\ dist\left(\Phi\Pi_j\left(A_{i\,orig}\right),\Phi\Pi_k\left(A_{i\,orig}\right)\right)$;
$j=\overline{1,N};\ i=\overline{1,M};k\neq j$

На рис. 12 и 13 показаны зависимости вероятностей правильной Рпф и ложной Рлф фиксации ФП в зависимости от КСТ. Из представленных зависимостей видно, что, например, при дистанции 0.0001 модель с КСТ 25 идентифицирует 70% искомых ФП, а при дистанции 0.1 идентифицируется уже 100% модифицированных ФП.

Из рис. 12 следует, что вероятность правильной фиксаций возрастает с уменьшением Rпор и уменьшением КСТ. Однако при этом возрастает вероятность неправильной идентификации — (1-Рпф) ,что иллюстрируется на рис. 9 с и 11 с.

Из рис.13 следует, что при значительном увеличении порогового уровня резко возрастает вероятность ложной идентификации модифицированного ФП из-за попадания в пороговую область «чужих» оригинальных ФП. Чем больше величина КСТ, тем значительнее рост таких событий.

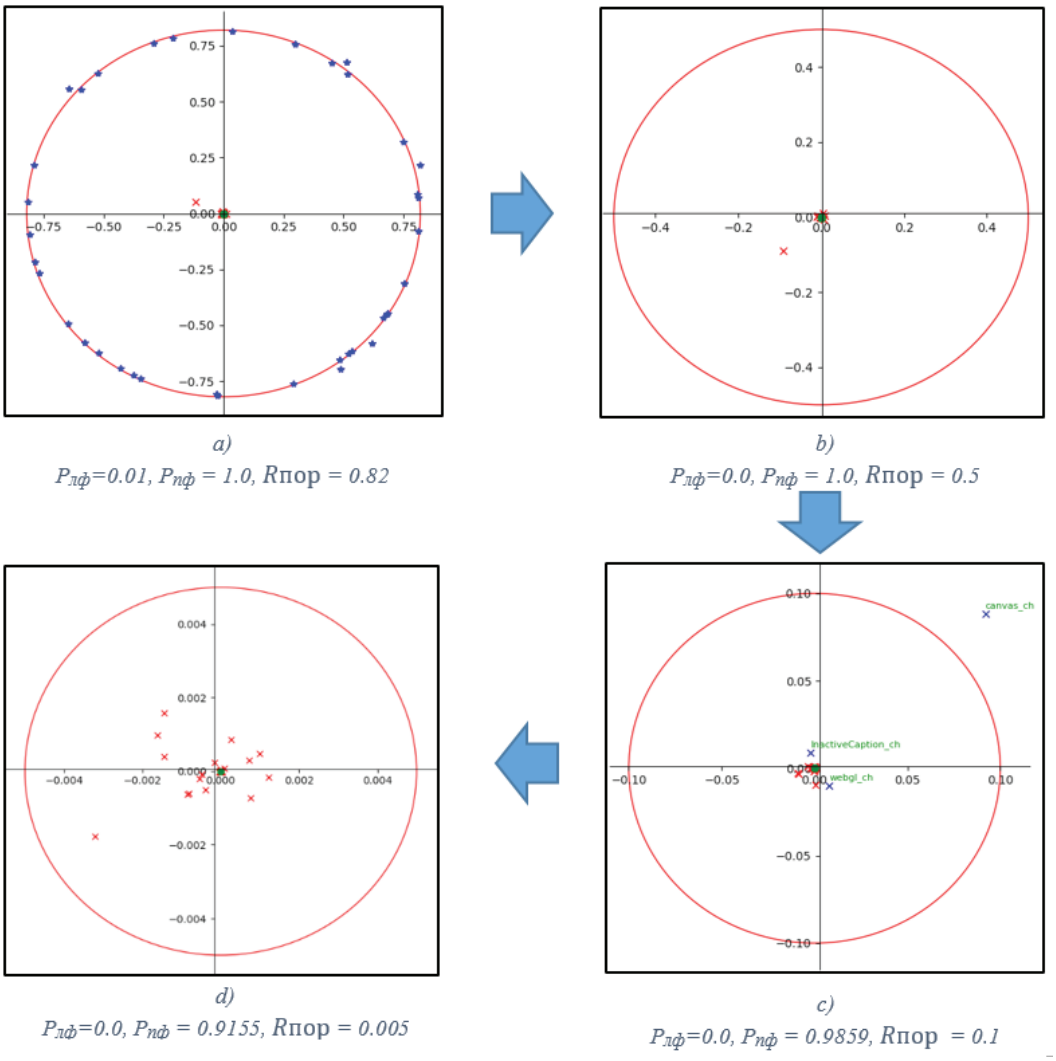


Рис.11. Расположение модифицированных векторов ФП, и границы порогов (КСТ =2000), где: x – модифицированные. ФП, ● – центр оригинальный ФП

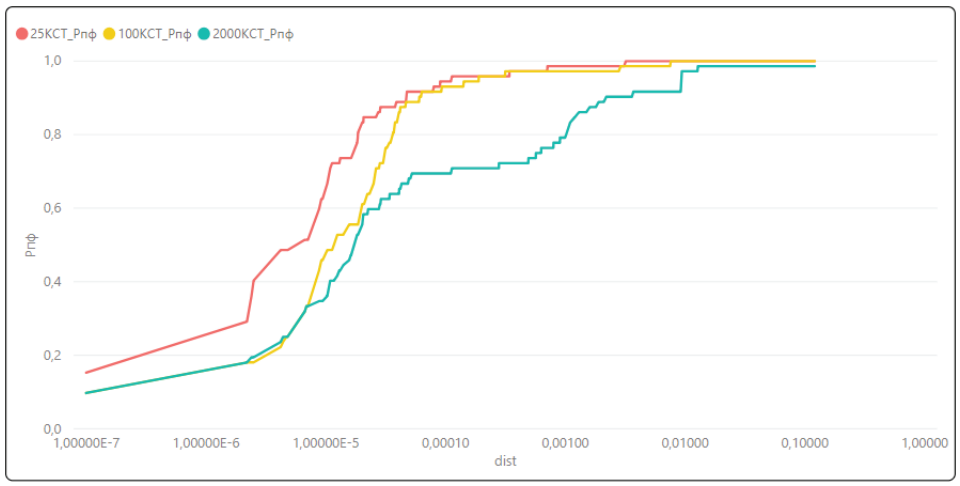


Рис. 12. Зависимости Рпф модифицированных ФП от косинусной дистанции

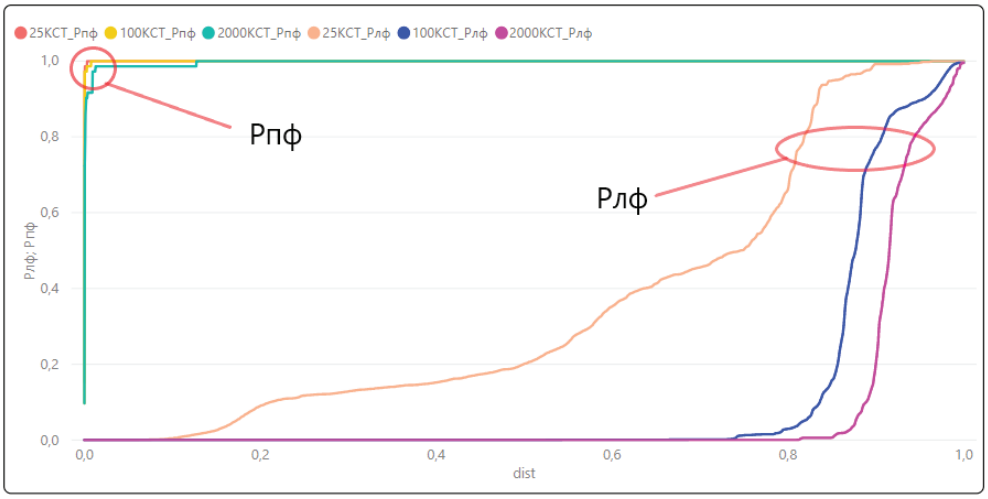


Рис. 13. Зависимости вероятностей правильной и ложной фиксации модифицированных ФП от косинусной дистанции

Выводы

Разработан метод, позволяющий идентифицировать пользователя путем отслеживания эволюции цифровых отпечатков цифровых устройств.

Показано, что при изменении наиболее важных параметров ФП (user-agent, webgl, language, canvas) либо атрибутов имеющих наибольшую длину (canvas, window_ dump_types, style_dump, webgl) модифицированные ФП могут быть эффективно идентифицированы. Показана возможность реализации кластеризации пользователей в рамках задачи идентификации пользователей путем визуализации ФП относительно скрытых семантических тем с помощью модели латентного семантического анализа LSA.

По итогам полученных результатов показано, что основными пунктами реализации предложенного алгоритма идентификации являются:

- выявление характерных факторов (тематик), присущих всем документам и терминам встречающимся в ФП с использованием латентного семантического анализа ;
- векторизация модифицированных ФП, путем пересчета дистанций между оригинальным и модифицированными цифровыми отпечатками с использованием библиотеки scipy функции distance.cosine библиотеки sklearn ;
- визуализация ФП относительно скрытых семантических тем с использованием алгоритма стохастического вложения соседей с t-распределением t-SNE;
- реализация алгоритма кластеризации, ориентированного на группировании данных по схожести , полученных после векторизации с помощью косинусного расстояния.

Достоверность результатов подтверждена вычислительными экспериментами с применением разработанного программного обеспечения, в ходе которых отмечена высокая эффективность предложенного подхода для решения задачи идентификации анонимных пользователей в условиях эволюции цифровых отпечатков их устройств.

Рецензент: Басараб Михаил Алексеевич, доктор физико-математических наук, профессор, заведующий кафедрой «Информационная безопасность» МГТУ им. Н.Э. Баумана. Москва, Россия. E-mail: bmic@mail.ru

Литература

1. Liu X., Liu Q., Wang X., and Jia Z. Fingerprinting web Browser for Tracing Anonymous Web Attackers. In IEEE First International Conference on Data Sciencein Cyberspace. DSC 2016. Changsha. China. IEEE Computer Society 2016. June 13-16. P. 222. DOI:10.1109/DSC.2016.78
2. Luangmaneerote S., Zaluska E., Carr L. Survey of existing Fingerprint countermeasures. In 2016 International Conference on Information Society (i-Society), IEEE Computer Society, October 2016. DOI:10.1109/I-SOCIETY.2016.7854198
3. Vastel A., Laperdrix P., Rudametkin W. Rouvoy R. FP-STALKER: Tracking Browser Fingerprint Evolutions // 39th IEEE Symposium on Security and Privacy (S&P 2018). San Fransisco, United States. DOI: 10.1109/SP.2018.00008
4. Roussev V. Data Fingerprinting with Similarity Digests. In Advances in Digital Forensics VI. Springer, 2010. https://doi.org/10.1007/978-3-642-15506-2_15
5. Bujlow T., Carela-Español V., Solé-Pareta J., Barlet-Ros P. A Survey on Web Tracking: Mechanisms, Implications, and Defenses. In Proceedings of the IEEE (2017). DOI:10.1.1109/JPROC.2016.2637878
6. Laperdrix P., Bielova N., Baudry B., Avoine G. Browser Fingerprinting: A survey. arXiv, Vol. 1, No. 1, Article . Publication date: May 2019.
7. Chen L., Wang G. An Efficient Piecewise Hashing Method for Computer Forensics. In IEEE WKDD, 2008. DOI:10.1109/WKDD.2008.80
8. Шелухин О.И., Желнов М. С. Идентификация анонимных пользователей ВЕБ-ресурса на основе нечетких хэш функций цифровых отпечатков устройств // REDS: Телекоммуникационные устройства и системы. 2021. №2. С. 57-63.
9. Laperdrix P., Baudry B., Mishra V. FPRandom: Randomizing core browser objects to break advanced device fingerprinting techniques. In 9th International Symposium on Engineering Secure Software and Systems (ESSoS Jul. 2017). Bonn, Germany. https://hal.inria.fr/hal-01527580
10. Alaca F. Oorschot P. C. V. Device Fingerprinting for Augmenting Web Authentication: Classification and Analysis of Methods // Annual Computer Security Applications Conference (ASAC'32), 2016. DOI: http://dx.doi.org/10.1145/2991079.2991091
11. Fifield D., Egelman S. Fingerprinting web users through font metrics. In Proceedings of the 19th international conference on Financial Cryptography and Data Security. Springer-Verlag. Berlin. Heidelberg. 2015. DOI:10.1007/978-3-662-47854-7_7
12. Bolón-Canedo V., Sánchez-Maroño N., Alonso-Betanzos A. A review of feature selection methods on synthetic data. Knowl. Inform. Syst., 34 (3) (2013), pp. 483-519
13. Доренская Е. А., Семенов Ю. А. Улучшенный алгоритм вычисления контекстного значения слов в тексте // Современные информационные технологии и ИТ-образование. 2019. Т. 15, № 4. С. 954-960. DOI: 10.25559/SITITO.15.201904.954-960
14. Carreira-Perpinan M.A. The elastic embedding algorithm for dimensionality reduction. In ´ Proceedings of the International Conference on Machine Learning, 2010. Pp. 167–174
15. Шелухин О. И., Осин А.В. Безопасность сетевых приложений / Под ред. О. И. Шелухина. М.: Горячая линия – Телеком, 2021. 224с. ISBN 978-5-9912-0911-3
16. Scott Deerwester et al. Indexing by Latent Semantic Analysis // Journal of the American society for information science. 41(6): pp. 391-407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391:AID-ASI1>3.0.CO;2-9
17. Liu S, Maljovec D, Wang B, Bremer PT, Pascucci V. Visualizing high-dimensional data: Advances in the past decade // IEEE Transactions on Visualization and Computer Graphics. 2017. Vol. 23(3). P.1249–1268. DOI: 10.1109/TVCG.2016.2640960
18. L.J.P. van der Maaten, Hinton G.E. Visualizing High-Dimensional Data Using t-SNE // Journal of Machine Learning Research. 2008, vol. 9. P.2579-2605,
19. L.J.P. van der Maaten. Accelerating t-SNE using Tree-Based Algorithms// Journal of Machine Learning Research 2014, vol.15. P.3221-3245.

THE IMPACT OF DIGITAL FINGERPRINT EVOLUTION ON THE AUTHENTICITY OF ANONYMOUS USER IDENTIFICATION

Sheluhin O.I.[10], Vanyushina A.V.[11], Bolshakov A.S.[12], Zhelnov M.S.[13]

Purpose of work *– is to evaluate the effectiveness of software identification of anonymous users in the context of the evolution of digital fingerprints on their devices.*

Research method. *Artificial intelligence technologies, including NLP (Natural Language Processing), methods of LSA (Latent semantic analysis), as well as methods of clustering and machine learning.*

Objects of study are *theoretical and practical issues of solving and visualizing information security problems.*

Results of the study. *To study the impact of the evolution of digital fingerprints of analyzed devices, by alternately changing the analyzed parameters of the original fingerprint (a digital fingerprint of a browser or digital device), a database of modified fingerprints was created. A calculation technique is proposed and numerical results are presented for estimating the probability of correct and false user identifications during the evolution of the attributes of digital fingerprints. The dependence of the effectiveness of user deanonymization depending on the characteristics and properties of the variable attributes of digital fingerprints of his devices is shown.*

Practical relevance *relevance. To improve the efficiency of anonymous user identification systems based on the analysis of device digital fingerprints.*

The proposed article will be useful both to specialists developing information security systems and to students studying "Information Security" course.

Keywords: *fingerprint, modified database, data set, text data, categorical data, features, artificial intelligence technologies.*

References

1. Liu X., Liu Q., Wang X., and Jia Z. Fingerprinting web Browser for Tracing Anonymous Web Attackers. In IEEE First International Conference on Data Sciencein Cyberspace. DSC 2016. Changsha. China. IEEE Computer Society 2016. June 13-16. P. 222. DOI:10.1109/DSC.2016.78
2. Luangmaneerote S., Zaluska E., Carr L. Survey of existing Fingerprint countermeasures. In 2016 International Conference on Information Society (i-Society), IEEE Computer Society, October 2016. DOI:10.1109/I-SOCIETY.2016.7854198
3. Vastel A., Laperdrix P., Rudametkin W. Rouvoy R. FP-STALKER: Tracking Browser Fingerprint Evolutions // 39th IEEE Symposium on Security and Privacy (S&P 2018). San Fransisco, United States. DOI: 10.1109/SP.2018.00008
4. Roussev V. Data Fingerprinting with Similarity Digests. In Advances in Digital Forensics VI. Springer, 2010. https://doi.org/10.1007/978-3-642-15506-2_15
5. Bujlow T., Carela-Español V., Solé-Pareta J., Barlet-Ros P. A Survey on Web Tracking: Mechanisms, Implications, and Defenses. In Proceedings of the IEEE (2017). DOI:10.1.1109/JPROC.2016.2637878
6. Laperdrix P., Bielova N., Baudry B., Avoine G. Browser Fingerprinting: A survey. arXiv, Vol. 1, No. 1, Article . Publication date: May 2019.
7. Chen L., Wang G. An Efficient Piecewise Hashing Method for Computer Forensics. In IEEE WKDD, 2008. DOI:10.1109/WKDD.2008.80
8. Sheluhin O.I., Zhelnov M. S. Identifikacija anonimnyh pol'zovatelej VEB-resursa na osnove nechetkih hjesh funkcij cifrovyh otpechatkov ustrojstv // REDS: Telekommunikacionnye ustrojstva i sistemy. 2021. №2. S. 57-63.

10 Oleg I. Sheluhin, Dr.Sc., Professor, Head of department of Information security , MTUCI, Moscow, Russia. E-mail: sheluhin@mail.ru

11 Anna V. Vanyushina, Ph.D., associate Professor at the Department of Information security, MTUCI, Moscow, Russia, E-mail: a.v.vaniushina@mtuci.ru

12 Alexander S. Bolshakov, Ph.D., associate Professor at the Department of Information security, MTUCI, Moscow, Russia, E-mail: as.bolshakov57@mail.ru

13 Maksim S. Zhelnov, student, MTUCI, Moscow, Russia, E-mail: max306211@yandex.ru

9. Laperdrix P., Baudry B., Mishra V. FPRandom: Randomizing core browser objects to break advanced device fingerprinting techniques. In 9th International Symposium on Engineering Secure Software and Systems (ESSoS Jul. 2017). Bonn, Germany. https://hal.inria.fr/hal-01527580
10. Alaca F. Oorschot P. C. V. Device Fingerprinting for Augmenting Web Authentication: Classification and Analysis of Methods // Annual Computer Security Applications Conference (ASAC'32), 2016. DOI: http://dx.doi.org/10.1145/2991079.2991091
11. Fifield D., Egelman S. Fingerprinting web users through font metrics. In Proceedings of the 19th international conference on Financial Cryptography and Data Security. Springer-Verlag. Berlin. Heidelberg. 2015. DOI:10.1007/978-3-662-47854-7_7
12. Bolón-Canedo V., Sánchez-Maroño N., Alonso-Betanzos A. A review of feature selection methods on synthetic data. Knowl. Inform. Syst., 34 (3) (2013), pp. 483-519
13. Dorenskaja E. A., Semenov Ju. A. Uluchshennyj algoritm vychislenija kontekstnogo znachenija slov v tekste // Sovremennye informacionnye tehnologii i IT-obrazovanie. 2019. T. 15, № 4. S. 954-960. DOI: 10.25559/SITITO.15.201904.954-960
14. Carreira-Perpinan M.A. The elastic embedding algorithm for dimensionality reduction. In ´ Proceedings of the International Conference on Machine Learning, 2010. Pp. 167–174
15. Sheluhin O. I., Osin A.V. Bezopasnost' setevyh prilozhenij / Pod red. O. I. Sheluhina. M.: Gorjachaja linija – Telekom, 2021. 224s. ISBN 978-5-9912-0911-3
16. Scott Deerwester et al. Indexing by Latent Semantic Analysis // Journal of the American society for information science. 41(6): pp. 391-407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391:AID-ASI1>3.0.CO;2-9
17. Liu S, Maljovec D, Wang B, Bremer PT, Pascucci V. Visualizing high-dimensional data: Advances in the past decade // IEEE Transactions on Visualization and Computer Graphics. 2017. Vol. 23(3). P.1249–1268. DOI: 10.1109/TVCG.2016.2640960
18. L.J.P. van der Maaten, Hinton G.E. Visualizing High-Dimensional Data Using t SNE // Journal of Machine Learning Research. 2008, vol. 9. P.2579-2605,
19. L.J.P. van der Maaten. Accelerating t-SNE using Tree-Based Algorithms// Journal of Machine Learning Research 2014, vol.15. P.3221-3245.