Nahuel González

Generalización del modelado de cadencias de tecleo con contextos finitos para su utilización en ataques de presentación y canal lateral

Tesis Doctoral en Ciencias Informáticas
PREMIO DR. RAÚL GALLARD | Año 2023

edulp
EDITORIAL DE LA UNLP

Generalización del modelado de cadencias de tecleo con contextos finitos para su utilización en ataques de presentación y canal lateral

Nahuel González

Universidad Nacional de La Plata

Facultad de Informática

Generalización del modelado de cadencias de tecleo con contextos finitos para su utilización en ataques de presentación y canal lateral

Nahuel González

Director: Jorge S. Ierache
Codirector: Waldo Hasperué
Asesor científico: Enrique P. Calot

La Plata, Argentina
Junio de 2023

EDITORIAL DE LA UNLP

González, Nahuel
Generalización del modelado de cadencias de tecleo con contextos finitos para su utilización en ataques de presentación y canal lateral / Nahuel González. - 1a ed - La Plata: EDULP, 2024.
Libro digital, PDF

Archivo Digital: descarga y online
ISBN 978-631-6568-19-9

1. Ingeniería Informática. 2. Biometría. 3. Seguridad Informática. I. Título.
CDD 005

Generalización del modelado de cadencias de tecleo con contextos finitos para su utilización en ataques de presentación y canal lateral

Nahuel González

EDITORIAL DE LA UNIVERSIDAD NACIONAL DE LA PLATA (EDULP)
48 Nº 551-599 4º Piso/ La Plata B1900AMX / Buenos Aires, Argentina
+54 221 44-7150
edulp.editorial@gmail.com
www.editorial.unlp.edu.ar

Edulp integra la Red de Editoriales de las Universidades Nacionales (REUN)

ISBN 978-631-6568-19-9
Queda hecho el depósito que marca la Ley 11.723

Impreso en Argentina

Dedico esta producción a todos los poetas, especialmente aquellos que se disimulan entre científicos e ingenieros y que esconden su producción entre artículos académicos.

Soy un término más en la intrincada
ecuación que describe el Universo
y rige el devenir. Todo mi esfuerzo,
infinitesimal o vasto, es nada.
La ecuación permanece balanceada
por un término igual aunque contrario
y un cierto operador, que es unitario,
me impide diverger. Junto a mi
anverso estoy irreparablemente
inmerso en esta sucesión.
Es necesario.

Abstract

Keystroke dynamics is a soft biometric trait that can be used as a transparent second factor authentication method. Every computer system, and especially those security related, can be expected to be under constant attack and should be designed under this consideration. In particular, any authentication system based on keystroke dynamics is liable to presentation attacks with synthesized samples; also, the keystroke timings leaked as the result of a side-channel attack can be leveraged to identify the typed text or to reduce the complexity of a later brute force attack. In spite of this, most methods proposed in the literature of the topic have been evaluated under a zero-effort attack model, which underestimates or plainly ignores the risk posed by the aforementioned. In this thesis we propose a liveness detection scheme that employs a family of keystroke dynamics synthesizers, used as adversaries in a classification system augmented with distances based on the empirical histograms of the user profile, to mitigate the risks of a presentation attack. Additionally, a modification of this scheme allows the identification of typed text using only keystroke timings, in longer texts and with larger candidate lists than the current state-of-the-art methods can reach.

Resumen

Las cadencias de tecleo configuran un atributo biométrico comportamental que puede ser utilizado como segundo factor de autenticación para la verificación transparente de la identidad del usuario. Todos los sistemas informáticos, y sobre todo aquellos relacionados con la seguridad, se encuentran bajo ataque permanente y deben ser diseñados con foco en esta consideración. En particular, un sistema de autenticación basado en cadencias de tecleo puede ser sometido a ataques de presentación con muestras sintetizadas; también, los atributos temporales de la escritura filtrados en el transcurso de un ataque por canal lateral pueden ser utilizados para identificar el texto ingresado o potenciar un ulterior ataque por fuerza bruta. Sin embargo, la mayoría de las veces los métodos propuestos en la literatura han sido evaluados bajo un modelo de esfuerzo cero, también llamado de impostores no entrenados, que subestima o ignora el riesgo de los anteriores. En este escrito se propone un sistema de detección de vida que emplea una familia de estrategias de síntesis de muestras artificiales, utilizadas como adversarios en un sistema de clasificación aumentado con distancias basadas en los histogramas empíricos del perfil intrausuario, para mitigar los riesgos de un ataque de presentación. Adicionalmente, una modificación del esquema propuesto para la defensa permite abordar el problema de identificación del texto ingresado utilizando solo atributos temporales, en textos más largos y con listas de candidatos más numerosas que en el estado del arte.

Agradecimientos formales

Al Universo, o al Multiverso si debemos creerle a Hugh Everett III, por existir. Esta producción no hubiera sido posible si, como teme Martin Heidegger en su Introducción a la Metafísica, no fuese el ser sino más bien la nada.

A mi país, en donde todavía hay educación pública, laica, y gratuita, y en la que he podido participar como alumno de todos los niveles académicos. Deseo fervientemente que la idiocracia y el oscurantismo no lo derroten, pero soy pesimista al respecto. Ha querido la fortuna que pueda devolverle algo con mi tarea docente, interrumpida y que hoy extraño en demasía.

A la Universidad Nacional de La Plata, por mantener un elevado nivel académico, y al Laboratorio de Sistemas de Información Avanzados de la Universidad de Buenos Aires, por alojar mis anhelos de investigación.

A Jorge, director de este proyecto. Si no hubiera contado contigo, mi doctorado no sería del orden del ser, sino más bien de la nada. Ambos sabemos que esta afirmación no es de carácter genérico, como las que suelen abundar en los agradecimientos.

A Quique, asesor científico y causa primera, en estricto orden cronológico, de esta producción. Si no me hubieras pedido ayuda con uno de los artículos para la tuya, jamás me hubiera decidido a empezar.

A Waldo, codirector de este escrito. Si la especulación nietszcheana sobre el eterno retorno resulta ser cierta, prometo en la próxima iteración de estos mismos acontecimientos someter mis artículos a tu consideración menos cerca de la fecha de cierre de los congresos.

Agradecimientos informales

A Oriana, sin cuyo oportuno (no oportunista, sino oportuno, acertado, preciso, afortunado) cariño y amor no hubiera podido recuperar aquel proyecto de doctorarme, que parecía haberse extraviado tiempo atrás en el páramo yermo de las pretensiones ilusas ya olvidadas; es una lástima que no haya podido ser.

Al farsante, lobo con piel de cordero, que con palabras melifluas descartó una amistad de veinticinco años para estafarme un monto pecuniario insignificante -como l-, pues me enseñó que no se puede confiar en nadie.

Al Dr. Jorge S. Ierache, en quien sí se puede confiar porque *la función de un docente es ser el guardián de los sueños del alumno, no hacer que su vida sea una pesadilla* [Ierache, marzo 2021, comunicación privada]. Doy fe de la consistencia de tus ideales con tus palabras.

A G. W. F. Hegel, cuya extensa y a veces tediosa filosofía dialéctica me instruyó en el curioso hecho de que no confiar en nadie y a la vez poder confiar en Jorge no son hechos contradictorios, sino momentos necesarios, como todas las antinomias, en el arduo y recóndito camino a la verdad absoluta; destino, cabe aclarar, que no alcanzaremos jamás.

A G. W. Leibnitz (¿qué fracción de los conspicuos filósofos alemanes comparte las iniciales *G. W.*?), cuya doctrina me reveló que vivimos en el mejor de los mundos posibles.

A Philip K. Dick y J. G. Ballard, que me hicieron contemplar este hecho con consternación, horror, y esa enigmática emoción denominada *awe* por los anglosajones para la cual nosotros, meros hispanoparlantes, carecemos de un significante que la denote. Quiera el destino que mis fantasías académicas no ayuden a hacer realidad sus pavorosos, aunque aparentemente proféticos, sueños literarios sobre el devenir implacable de la tecnología.

A mis viejos, que me presentaron la letra de los aquí mencionados desde Hegel (inclusive) en adelante.

Haber compartido un terso trecho de este curso con el resto es responsabilidad, o quizás culpa, exclusivamente mía.

Insistimos todavía en el método: se trata de obstinarse. En cierto punto de su camino, todo hombre es solicitado. La historia no carece de religiones ni de profetas, inclusive sin dioses. Se le pide que salte. Todo lo que puede responder es que no comprende bien, que eso no es evidente.

Albert Camus, El Mito de Sísifo

Índice general

Índice de figuras

Índice de cuadros

Índice de código

Capítulo 1

Introducción

Parte I

El problema

Capítulo 2

Antecedentes, definiciones, y el estado del arte

> Para hacer una tarta de manzanas desde el principio, debemos comenzar por crear el universo
>
> Carl Sagan

2.1. Introducción

El epígrafe ilustra la motivación de este capítulo, cuyo objetivo es definir los conceptos más importantes de la disciplina de análisis de cadencias de tecleo para luego reseñar exhaustivamente el estado del arte. Las tendencias del campo, las áreas inexploradas, y las oportunidades de mejora que encontremos aquí y que se sintetizan en la sección 2.6 permitirán delinear posteriormente el problema central de esta tesis y el alcance de la solución, en el capítulo 3, además de motivar tanto los métodos propuestos en el capítulo 4 como el marco experimental de evaluación y el diseño de los experimentos en el capítulo 5.

Con el objetivo de agrupar en un único capítulo todo el contenido teórico necesario para el resto de la tesis y de esa manera simplificar la exposición, se ha preferido relegar a los apéndices algunos temas que el lector puede saltearse en una primera lectura. Los fundamentos algebraicos sobre distancias y métricas se refrescan en el apéndice A; allí se incluyen también las definiciones y derivaciones de los atributos calculados de uso más común en el análisis de cadencias de tecleo en textos libres para ahorrar al lector que no sea experto en el tema la engorrosa tarea de consultar una bibliografía dispersa. El análisis bibliométrico de la literatura del tema y los fundamentos de la revisión estructurada de la literatura pueden consultarse en el apéndice B.

El resto del capítulo está organizado como se describe a continuación. La sección 2.2 define y explica los conceptos que serán utilizados en el resto del libro. La sección 2.3 presenta la perspectiva histórica y los principales hitos de la disciplina, desde el primer estudio exploratorio a principios de la década de los 80s hasta hoy. La sección 2.4 reseña detalladamente la literatura del tópico, centrándose en la autenticación de usuarios (sección 2.4.2) pero sin dejar de mencionar otros usos de las técnicas y sus consideraciones

relacionadas (secciones 2.4.3 y 2.4.4). Se incluyen aquí tres secciones con una revisión detallada de los ejes que motivarán los métodos propuestos: distribuciones subyacentes en la sección 2.4.5, falsificaciones sintéticas y ataques de presentación en la sección 2.2.6.2, y ataques por canal lateral e identificación del texto ingresado en la sección 2.2.6.3. La sección 2.5 está dedicada a introducir la técnica de modelado con contextos finitos, sobre la cual se edificarán los restantes métodos.

Finalmente, la sección 2.6 sintetiza el estado del arte y motiva la investigación ulterior.

2.2. Conceptos y definiciones

En esta sección se definen los conceptos más importantes de la disciplina, con el objeto de facilitar la lectura de este y los restantes capítulos. En primer lugar, diferenciamos las tareas de autenticación, verificación, e identificación y los tipos de factores utilizados, a lo largo de la sección 2.2.1. Las métricas de eficiencia para evaluar cuantitativamente el rendimiento de un método de autenticación, que serán utilizadas para reportar los resultados de los experimentos, se describen en la sección 2.2.2. Los tiempos de retención y latencia, y otros atributos biométricos de la cadencia de tecleo que se consideraran en este estudio se definen en la sección 2.2.3, y sus agrupaciones en di gramas, trigramas, y *n*-gramas se justifica en la sección 2.2.4. Finalmente, los tipos de perfiles de usuarios y los ataques a los que son pasibles se explican respectivamente en las secciones 2.2.5 y 2.2.6.

2.2.1. Autenticación, verificación, e identificación

Se denomina *autenticación* al proceso necesario para demostrar que la identidad del usuario se corresponde con la requerida para acceder a los privilegios solicitados [1]. De acuerdo al medio utilizado, estos pueden dividirse en tres tipos fundamentales:

- *Basado en el conocimiento*. El factor de autenticación es una pieza de conocimiento que el usuario legítimo posee y se supone secreta, salvo tal vez para el sistema que autentica. La verificación de la identidad de usuario por medio de claves o *passwords* es la instancia más característica y más extendida de este tipo de autenticación. Otro ejemplo lo constituyen los patrones de puntos para desbloqueo de teléfonos celulares.

- *Basado en la posesión*. El factor de autenticación es un objeto físico único o con una cantidad de copias limitada, que se supone difícil o costoso de replicar. Las llaves son un ejemplo común; otro más actual son los *tokens* criptográficos para firma digital o electrónica.

- *Biométrico*. El factor de autenticación es una característica o una combinación de características fisiológicas y/o comportamentales medibles, que identifican con suficiente precisión al usuario legítimo. Como ejemplo de las primeras, tenemos las

huellas dactilares, el patrón de vasos sanguíneos de la retina, o su contorno facial. Como ejemplo de las segundas, tenemos la modulación de la voz y la cadencia de tecleo.

La búsqueda de niveles de seguridad más elevados y a la vez más sencillos en su empleo ha llevado a la experimentación con esquemas de autenticación mixta que mezclen seguridad basada en el conocimiento, en la posesión y por caracteres biométricos. Cuando se utilizan dos técnicas combinadas, se habla de un *segundo factor de autenticación*, en inglés *two-factor authentication*, o de su acrónimo *2-FA*. En el caso de que se apliquen más de dos factores, hablamos de *autenticacion multifactor*, en inglés *multi-factor authentication*, o de su acrónimo *MFA*.

El proceso de autenticación de usuarios puede realizarse en uno o varios pasos, y según donde intervenga el sistema en consideración se puede hablar de un proceso de *identificación* o de *verificación* [1]. Se denomina *identificación* al proceso de determinación de la identidad de un usuario dentro de un conjunto previamente registrado en base a la información dada al momento de la autenticación. Para la *verificación* se considera que el usuario ya ha sido identificado y el objetivo del proceso es determinar si este es efectivamente quien se corresponde con las credenciales dadas o si se trata de un impostor.

En forma equivalente, la identificación puede ser considerada un problema de clasificación de uno entre muchos, posiblemente con detección de valores anómalos, mientras que la verificación es un problema de clasificación con solo dos clases: usuario legítimo e impostores.

Los esquemas de autenticación más sencillos, como por ejemplo el control de acceso a computadoras portátiles por medio de huellas dactilares, combinan ambos procesos en un único paso. El tamaño reducido del conjunto de usuarios, la elevada precisión de los métodos de verificación y los bajos requerimientos de seguridad hacen aceptable y practico este ultimo esquema. Sin embargo, conjuntos de usuarios más grandes, métodos menos precisos, o mayores requerimientos de seguridad exigen uno o más pasos tanto de identificación como de verificación, como es fácilmente observable en los sistemas bancarios y otros que manejen información sensible o valiosa.

A pesar de que es posible aplicarlo a la identificación de usuarios, el análisis de cadencias de tecleo se aplica fundamentalmente a la verificación, teniendo como función comprobar que la identidad del usuario legítimo no está siendo usurpada o que el mismo no ha cedido su sesión a un usuario no autorizado o de menores privilegios. El motivo es simple; la eficiencia en la clasificación, a pesar de ser aceptable para complementar otros esquemas mejorando el nivel de seguridad total, no es suficiente por sí mismo para cumplir ambas funciones.

2.2.1.1.Autenticación continua y desafío/respuesta

La verificación de usuarios legítimos por su cadencia de tecleo característica puede insertarse en diferentes etapas de la sesión de usuario o en forma previa al inicio de

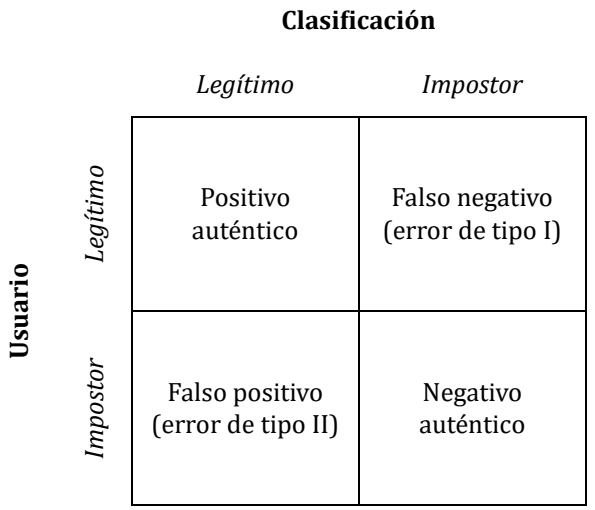

Figura 2.1: Matriz de confusión

la misma. Esta última posibilidad es generalmente utilizada cuando se restringe la verificación a una clave estática que el usuario emplea para iniciar su sesión.

Cuando una única verificación inicial no es suficiente para alcanzar el nivel de seguridad requerido, bien porque un impostor podría reemplazar al usuario legítimo autenticado una vez iniciada la sesión o porque el usuario legítimo podría ceder voluntariamente la misma, se requiere la implementación de un esquema de *autenticación continua* [2], también llamada de textos libres.

Una alternativa más laxa a la autenticación continua es la *autenticación por desafío-respuesta* [3], donde se solicita periódicamente al usuario el reingreso de alguno de los textos de entrenamiento, posiblemente modificado. El desafío-respuesta puede ser utilizado también en forma previa al inicio de sesión, cuando el nivel de seguridad biométrica requerido supera el alcanzable con un texto corto como la clave del usuario.

No menos importancia tiene la investigación forense [4], que puede ser requerida una vez finalizada la sesión, pero que demanda el registro persistente y detallado de todos los eventos de presión y liberación capturados durante el proceso.

2.2.2. Métricas de eficiencia

2.2.2.1. La matriz de confusión

Un algoritmo de clasificación binaria puede, para ambas clases - en este caso usuarios legítimos e impostores-, realizar una identificación correcta o incorrecta. Los cuatro resultados posibles suelen graficarse bajo la forma denominada *matriz de confusión* (ver figura 2.1) en donde las filas representan las clases a las que pueden pertenecer los objetos a clasificar y las columnas representan la clase asignada por el algoritmo. Cuando la clase de pertenencia y la clase asignada coinciden, indicando una clasificación correcta, se habla de *auténticos positivos* y *auténticos negativos*; los errores de clasificación de usuarios legítimos e impostores se denominan respectivamente *falsos negativos* y *falsos positivos* o, en la terminología usual de la estadística, errores de tipo I y II.

2.2.2.2. FAR y FRR

Las métricas generalmente utilizadas para determinar la calidad de la clasificación en los esquemas biométricos de autenticación, incluyendo el modelado de cadencias de tecleo son dos: FAR (*False Acceptance Rate* o *tasa de falsos positivos*) y FRR (*False Rejection Rate* o *tasa de falsos negativos*). La primera de ellas indica la proporción de intentos de autenticación realizados por un impostor que son equivocadamente clasificados como intentos legítimos mientras que la segunda expresa la correspondiente tasa de intentos rechazados a pesar de provenir de un usuario legítimo.

$$FAR = \frac{\#Intentos\ ilegítimos\ aceptados}{\#Total\ de\ intentos\ de\ autenticación}\ [\%] \quad (2.1)$$

$$FRR = \frac{\#Intentos\ genuinos\ rechazados}{\#Total\ de\ intentos\ de\ autenticación}\ [\%] \quad (2.2)$$

Debe entenderse que ambos valores expresan cocientes asintóticos o, en otras palabras, la esperanza matemática para secuencias largas de intentos de autenticación. La eficiencia en la clasificación para secuencias cortas o con pocos usuarios puede variar significativamente.

Bajo las denominaciones del apartado anterior, el FAR y el FRR corresponden a las tasas de errores de tipo I y de tipo II, respectivamente. Considerando como hipótesis nula la ausencia de correlación entre el modelo de la cadencia de tecleo del usuario y las características biométricas del intento realizado, el rechazo de la hipótesis es equivalente a la aceptación de un impostor e, inversamente, su aceptación equivale al rechazo de un usuario legítimo.

Para los esquemas de verificación de claves estáticas es común que la magnitud de ambas métricas sea muy parecida; valores del 5% o menos son usualmente reportados en la literatura [5]. En los esquemas de autenticación continua se suele exigir un FRR más bajo –generalmente logrado a costa de elevar significativamente el FAR–, ya que la mayor complejidad de los modelos, la mayor desviación en los parámetros de tecleo durante una sesión de usuario o entre sesiones pero sobre todo la cantidad de evaluaciones realizadas durante una sesión hacen que valores de FRR tolerables en la verificación de claves estáticas vuelvan imposible prolongar una sesión con autenticación continua.

A modo de ejemplo de esta última afirmación consideremos la experiencia de usuario de un sistema de verificación estática de claves con FAR y FRR del orden de 5%. En promedio, uno de cada veinte intentos de un usuario legítimo es rechazado por el sistema, mientras que un impostor que haya logrado conocimiento de una clave valida y no sepa imitar la cadencia de tecleo del usuario legítimo será rechazado veinte veces –cantidad suficiente para activar otros mecanismos de protección– antes de ser aceptado como falso positivo. La situación descripta es deseable y muestra

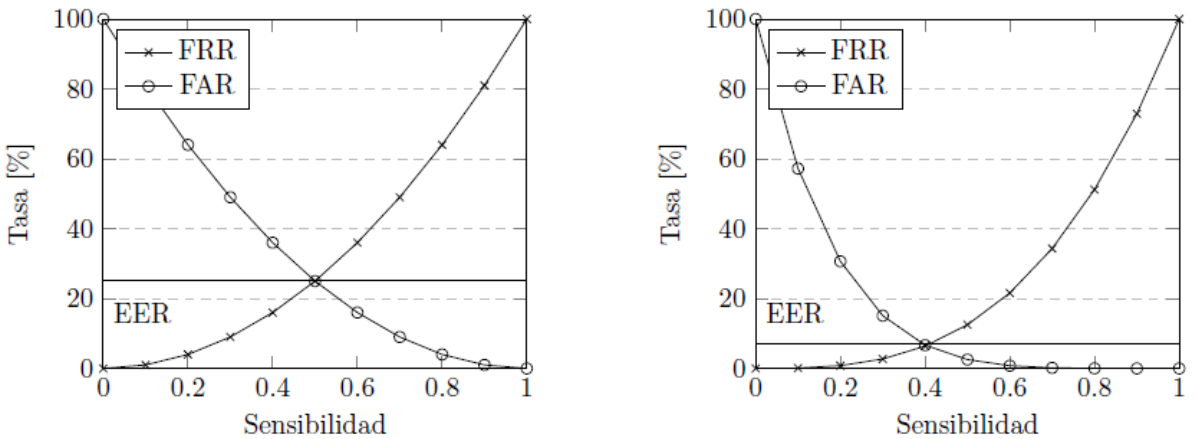

Figura 2.2: Curvas FAR y FRR en dos sistemas con distinto ERR

una clara mejora sobre la autenticación basada exclusivamente en el conocimiento. Por el contrario, el mismo FRR en un sistema de autenticación continua puntuada por oraciones generaría un cierre de sesión o un alerta de seguridad cada veinte oraciones, tornándolo inutilizable.

2.2.2.3. ERR

Es común que un sistema de identificación o verificación biométrica cuente con un conjunto de parámetros, no necesariamente accesibles a los usuarios finales, que permiten ajustar la respuesta del mismo o sus capacidades de clasificación. Los motivos son diversos; puede ser deseable canjear precisión por simplicidad, evitar falsos negativos o cualquier otro. Haciendo la simplificación de agrupar todos estos parámetros bajo una única variable, a nuestros propósitos sin unidades, que será denominada *sensibilidad* tenemos que el FAR y el FRR (y por lo tanto el costo de error) pueden expresarse en función de la misma. Nótese que no se está utilizando la significación usual del término en estadística, donde equivale a la tasa de positivos auténticos.

Es razonable esperar que a menor sensibilidad del sistema sea más probable con fundir al impostor con el usuario legítimo e, inversamente, que a mayor sensibilidad el usuario legítimo sea rechazado con más frecuencia. Salvando el detalle de la forma de las curvas, que no necesariamente son convexas ni simétricas, aunque si monótonas; la situación es similar a la graficada en la figura 2.2.

El nivel en el cual ambas curvas se cortan se denomina ERR (*Equal Error Rate*, *tasa de errores idénticos* o *cruce de errores*, aunque la forma castellana no es utilizada casi nunca) y es un indicador de la calidad del sistema biométrico. Un valor más bajo de ERR se debe necesariamente a curvas de FAR y FRR más aplastadas o, lo que es lo mismo, a tasas más bajas de falsos positivos y negativos para una sensibilidad dada, como se observa en la figura de la derecha.

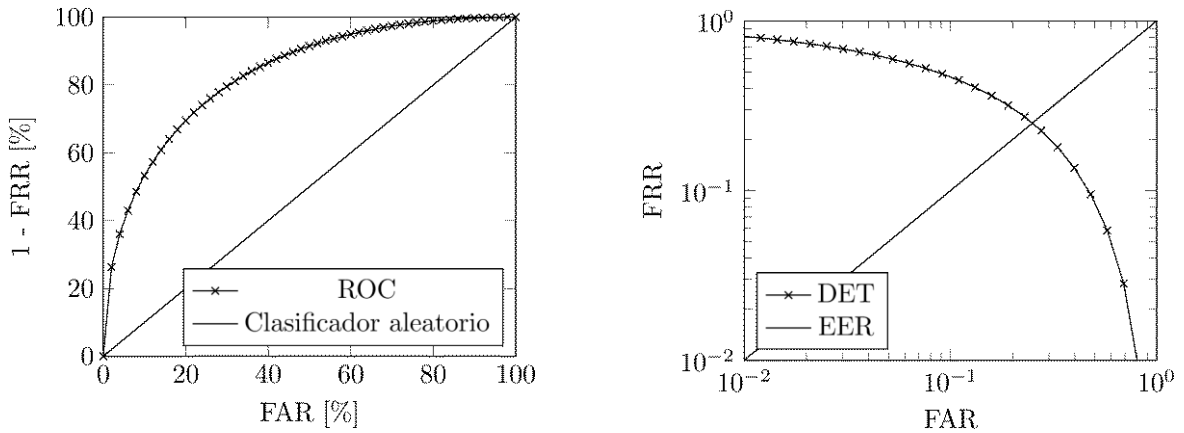

Figura 2.3: Ejemplos de curvas ROC y DET

2.2.2.4. Curvas ROC y DET

Para evitar toda referencia al parámetro de sensibilidad, que no siempre puede ser explicitado, pero sin perder la información dada por ambas curvas de FAR y FRR, se puede trazar la curva de una de ellas en función de la otra. Al graficar FRR en función de FAR obtenemos la curva DET (del inglés *Detection Error Tradeoff*), que suele graficarse con escalas logarítmicas, mientras que la curva ROC (del inglés *Receiver Operating Characteristic*) surge de graficar la tasa de positivos auténticos (que se puede calcular como $1 - FRR$) también en función del FAR. Las curvas ROC y DET para el sistema de la izquierda de la figura 2.2 pueden verse en la figura 2.3.

Como una asignación de clases enteramente aleatoria produce una tasa de falsos positivos igual a la tasa de positivos auténticos, la distancia a la diagonal del gráfico de la curva ROC es una medida de la calidad del clasificador para una sensibilidad o tasa de error dada. Para aclarar la primera afirmación, consideremos un clasificador que asigna el rotulo de usuario legítimo con probabilidad p e independientemente de la clase del usuario; dicho algoritmo tiene tasas tanto de falsos positivos como de positivos auténticos igual a p. De aquí que para cada p, el punto (p, p) de la diagonal corresponde a un tal clasificador. Hacia arriba de la diagonal se puede considerar que el algoritmo utilizado es mejor que el azar, con un límite en el extremo superior izquierdo del grafico donde se ubica el algoritmo perfecto capaz de clasificar correctamente a todos los usuarios legítimos sin autorizar a ningún impostor. Es por esto que el área bajo la curva ROC, donde valores mayores implican mejor calidad, permite evaluar un clasificador a largo de todo su rango de sensibilidad.

Un punto por debajo de la diagonal implica que el algoritmo de clasificación es peor que una elección aleatoria. En la práctica no se observan tales valores ya que, de existir, con solo invertir la clase asignada por el mismo obtendríamos un clasificador mejor que el aleatorio.

El concepto de la curva DET es similar, pero tiene la ventaja de que la sección de importancia de la curva ocupa un área mayor y por lo tanto permite una visualización más clara. En este caso, la diagonal corresponde a iguales tasas de falsos positivos y´ negativos, por lo que corresponde al concepto del ERR descrito más arriba.

2.2.2.5. Otras métricas menos utilizadas

Cuando la simplicidad en el uso de un sistema biométrico es un requisito de mayor peso que la seguridad del mismo o cuando los falsos negativos son fuertemente inconvenientes, el *punto de FRR cero* indica la tasa de falsos positivos al configurar el sistema para evitar el rechazo de usuario legítimos. En la práctica este último requisito no es alcanzable, por lo que se suele adoptar un FRR aceptablemente bajo para el tipo de uso esperado. Sin embargo, algunos estudios antiguos reportan resultados utilizando esta métrica.

La utilización de múltiples valores como el FAR y el FRR puede complicar la comparación de la capacidad de las distintas combinaciones de algoritmos y parámetros biométricos para realizar una clasificación eficaz. Es por lo tanto deseable una métrica que combine ambos en un único número. Para tal fin se han propuesto [6] las métricas denominadas *costo de error*, ponderado y no ponderado, cuyas expresiones son:

$$C_u = FAR\ [\%] + FRR\ [\%] \quad (2.3)$$

$$C_W = C_{FAR}\ x\ FAR\ [\%] + C_{FRR}\ x\ FRR[\%] \quad (2.4)$$

Sumando el FAR y el FRR, expresados en porcentaje, se obtiene el *costo de error no ponderado* C_U, que surge de considerar que un falso positivo y un falso negativo revisten la misma gravedad. Este no suele ser así en situaciones reales, pero puede suceder que no se cuente con información apriorística detallada que permita dar un peso adecuado a ambos errores; para este caso, el costo de error no ponderado es a la vez sencillo de calcular y carece del problema mencionado más arriba.

Para la evaluación del *costo de error ponderado* C_W, se dan las valoraciones relativas C_{FAR} y C_{FRR} al FAR y el FRR, lo cual refleja en forma más precisa tanto los requerimientos de los sistemas de autenticación como los marcos normativos en los que estos se insertan. Usualmente se exigen valores más bajos al FAR que al FRR en los sistemas de autenticación biométrica de factor único.

A modo de ejemplo, consideremos un clasificador con FAR de 2 % y FRR del 5 %. Realizando una valoración de gravedad de 10 para un falso positivo y de 1 para un falso negativo, tenemos que el costo de error ponderado es de 25 y el no ponderado de 7.

2.2.3. Tiempos de retención y latencia

Cada modalidad biométrica utiliza un conjunto de atributos para caracterizar la fisiología o el comportamiento del usuario. El análisis de cadencias de tecleo emplea, casi siempre, dos vectores de atributos temporales, los *tiempos de retención* y las *latencias*. Para definir ambos, consideramos que al escribir en un teclado se generan dos tipos de eventos, de *presión* y de *liberación*, para cada tecla ingresada.

- **Presión**, en inglés *keydown*. Este evento se dispara cuando el usuario presiona una tecla que no se encuentra presionada.

- **Liberación**, en inglés *keyup*. Este evento se dispara cuando el usuario suelta una tecla que estaba siendo presionada.

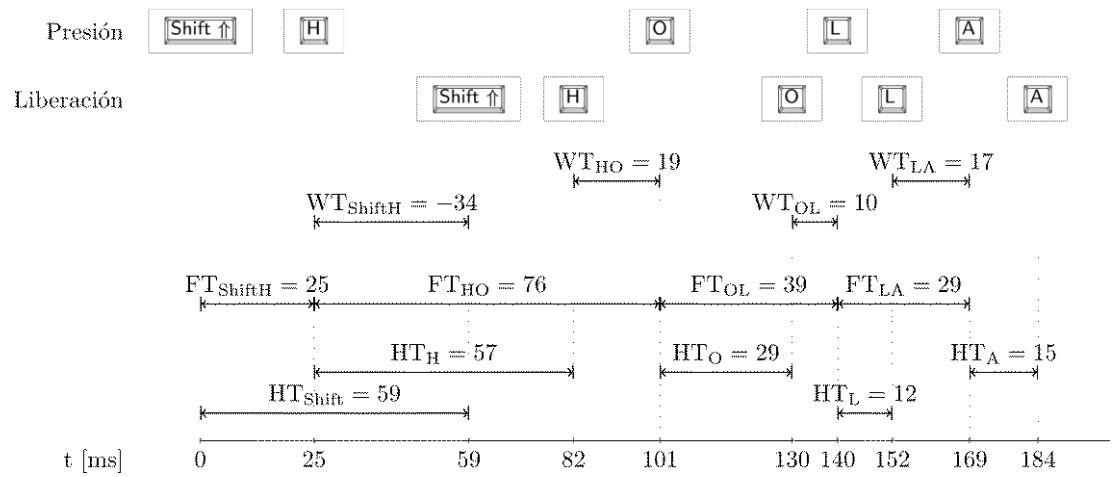

Figura 2.4: Eventos de presión y liberación de tecla, tiempos de retención y latencia

Como se observa en la figura 2.4, la escritura produce una secuencia de tiempos varios entre eventos. Nos interesan dos de ellos.

- **Tiempo de retención**, denotado en la figura como HT, del inglés *hold time*. Representa el tiempo entre el evento de presión de una tecla y el evento de liberación de la misma. Siempre es positivo.

- **Latencia**, denotado en la figura como FT, del inglés *flight time*. Representa el tiempo entre el evento de presión de una tecla y el evento de presión de la tecla´ siguiente. Siempre es positivo.

También se ha utilizado, aunque con menos frecuencia, el denominado *tiempo de espera* o *wait time* en inglés, denotado en la figura como WT. Este representa el intervalo entre el evento de liberación de una tecla y el evento de presión de la tecla siguiente. Nótese que, a diferencia de los dos anteriores, puede ser negativo si ocurre una *inversión de teclas*; es decir, si el usuario presiona la tecla siguiente antes de soltar la actualmente presionada. Una situación tal se observa en la figura 2.4 entre las teclas SHIFT y H. Para ingresar la letra H como mayúscula, primero se presiona SHIFT, luego la tecla H, y luego se sueltan ambas. De esta forma, el tiempo de espera WT_{ShiftH} termina siendo igual a -34 mseg. en el ejemplo.

La muestra de escritura de la figura determina dos vectores de tiempos de largo cinco, para retención y latencia, con una componente para cada tecla.

$$\mathbf{t}_{HT} = \{59, 57, 29, 12, 15\}$$

$$\mathbf{t}_{FT} = \{\emptyset, 25, 76, 39, 29\}$$

No hay tiempo de latencia para la primer tecla de la muestra, en este caso SHIFT, pues no tiene tecla precedente. Preferimos conservar ambos vectores con el mismo tamaño y utilizar Ø como valor nulo.

El acto de escribir no solo determina vectores de tiempos. Si se cuenta con los sensores adecuados, como por ejemplo en un dispositivo móvil, puede agregarse información muy variada sobre el comportamiento característico. Un conjunto de acelerómetros puede proveer valores de aceleración lineal y angular. Una pantalla táctil o un teclado de propósito específico pueden proveer valores de presión. Nos limitaremos en el marco de esta tesis a considerar solo atributos temporales.

2.2.3.1. Otros atributos derivados

Algunos patrones generales de comportamiento durante la escritura no son directamente medibles pero sí derivables de los tiempos de retención y latencia, o de los nombres de teclas. Entre ellos se cuentan:

- *Velocidad general de tecleo*. Es un resultado directo del nivel de instrucción y experiencia dactilográfica del usuario y por lo tanto varía lentamente con el paso del tiempo.
- *Probabilidad de error*. Se mide con la frecuencia de uso de las teclas *backspace* y *delete*. Al igual que en el caso anterior, es un resultado directo del nivel de instrucción y experiencia dactilográfica del usuario, pero puede presentar variaciones significativas entre sesiones o dentro de la misma sesión de usuario.
- *Hábitos de sectorización*. Las aplicaciones que utiliza el usuario durante su sesión pueden producir hábitos de sectorización, como por ejemplo diferentes frecuencias relativas en el uso de números y de caracteres alfabéticos o especiales. Un programador abundara en estos últimos, mientras que un contador probablemente presente una mayor frecuencia de los primeros.
- *Orden de liberación de teclas*. Al utilizar las teclas de mayúsculas, control, alteración o *host*, entre otras, el orden de liberación de la tecla afectada y la tecla afectante no produce cambios visibles en el texto ingresado. Dicho orden suele respetarse para las mismas teclas bajo los mismos contextos en un usuario dado, por lo que agrega información para la clasificación.

Ninguno de los mencionados es suficientemente significativo para ser considerado en forma aislada, pero todos ellos permiten mejorar la eficiencia de la clasificación, ya que agregan información de comportamiento correlacionada con la identidad del usuario.

2.2.4. Digramas, trigramas, y n-gramas

Los intervalos de tiempo descritos en la sección anterior se miden entre teclas consecutivas, combinación denominada *digrama* o a veces *digrafo* en la literatura del tópico.

Tanto para textos fijos como para textos libres [7, 8, 9] se ha explorado el análisis de los tiempos entre eventos correspondientes a grupos de teclas consecutivas de mayor orden. Al referirse a grupos de tres teclas sucesivas, se utiliza el nombre *trigramas* y, en general, *n-gramas* cuando hablamos de un grupo de tamaño *n*.

Con esta libertad podemos formar diversas combinaciones de eventos de presión y liberación de teclas, de los cuales los correspondientes a digramas son un subconjunto pequeño. Tiempo atrás, estas consideraciones han permitido mejorar, si bien en forma marginal, la eficiencia en la clasificación [7]. Sin embargo, cuando se comenzaron a utilizar técnicas de aprendizaje automático y clasificadores más poderosos la consideración de los mismos dejó de ser fructífera. En el mejor de los casos, las técnicas más modernas extraen esta información en forma implícita. En el peor, al estar muchos de estos atributos de *n*-gramas fuertemente correlacionados, se perjudica la precisión del método.

Debemos, sin embargo, reconocer en esta idea un antecedente de la utilización de contextos finitos para el modelado de cadencias de tecleo, puesto que explota, en forma parcial, la observación de que los parámetros correspondientes a una tecla son condicionados por las anteriormente ingresadas.

2.2.5. Perfiles intrausuario e interusuario

El *perfil biométrico* de un usuario consiste en un conjunto de *muestras*, que son observaciones fechadas de su comportamiento al momento de la autenticación. Para el caso que nos compete, una muestra consiste en dos vectores de tiempos, para retención y latencia, junto con la secuencia de teclas que los origina. En la mayoría de los casos se conservan en el perfil solo aquellas muestras que han sido clasificadas como *legítimas* -es decir originadas por el usuario y no por un impostor-, para no contaminar los modelos con información biométrica de otros usuarios. A veces pueden conservarse también las muestras clasificadas como impostores para fines de análisis forense, o para incluir en los modelos cierta variación esperada en la cadencia de tecleo del usuario legítimo.

Cuando todas las muestras de un perfil biométrico corresponden a un cierto usuario, hablamos de un *perfil intrausuario*. Todo sistema de autenticación posee necesariamente un perfil intrausuario para cada usuario legítimo que es verificado. Un atacante también puede tener acceso a un perfil intrausuario si ha conseguido robar información del sistema de autenticación o si ha logrado observar indirectamente el comportamiento del usuario por medio de un ataque de canal lateral (ver sección 2.2.6.3). Según si el atacante cuenta con la misma información que el sistema de autenticación, o solo una parte de ella, hablamos de un perfil intrausuario *total* o *parcial*.

Hablamos de un *perfil interusuario* cuando este ha sido construido con muestras de varios usuarios, o con las de un usuario que no es el usuario legítimo. Un perfil interusuario es útil para estudiar el comportamiento general de la población o para construir muestras sintéticas cuando se lleva a cabo un ataque con *malware* (ver sección 2.2.6.1). Puede ser utilizado también como adversario por el sistema de autenticación para entrenarse contra falsificaciones sintéticas.

2.2.6. Ataques

Hoy en día todos los sistemas se encuentran bajo ataque y deben ser diseñados priorizando esta consideración. Los sistemas de autenticación basados en cadencias de tecleo no están exentos de la regla. La sección 2.2.6.1 describe los modelos de ataque a los que puede ser sometidos los antedichos. En resumen, pueden ser atacados por impostores humanos, preparados o no, además de por *malware*.

Los tipos de ataque más relevantes para un sistema biométrico son los *ataques de presentación* y los *ataques por canal lateral*. En los primeros, que se describen en la sección 2.2.6.2, se presenta al sistema biométrico bajo ataque una muestra artificial que ha sido preparada para imitar al usuario legítimo. En los segundos, que se describen en la sección 2.2.6.3, se utilizan vulnerabilidades del sistema de autenticación, o de otros sistemas que conviven con él, para extraer información sobre el comportamiento del usuario legítimo que luego permitirá mejorar las posibilidades de éxito de un ataque de presentación.

2.2.6.1. Modelos de ataque

El significado de una tasa de error de clasificación reportada, ya sea FAR o FRR, depende del modelo de ataque que se está presuponiendo y con el cual se ha evaluado el modelo de autenticación que se propone. Si se supone que los usuarios impostores no tienen conocimiento de la existencia de una capa de autenticación biométrica oculta basada en la cadencia de tecleo además de las formas explícitas y expuestas a la vista como pueden ser la exigencia de una clave u algún otro identificador biométrico, es esperable que estos no intenten disimular su forma de escribir; hablamos en este caso del modelo de ataque *con impostores no entrenados* o *de esfuerzo cero*. Contrariamente, si el impostor es consciente de la existencia de un esquema tal, es razonable que busque imitar la cadencia del usuario legítimo y nos encontramos ante un modelo de ataque con *impostores entrenados*.

Es intuitivamente aprensible que la detección de impostores entrenados es, por lejos, el problema más difícil y deben por lo tanto interpretarse los valores de FAR en forma más benigna que con impostores no entrenados. En otras palabras, es esperable un FAR más elevado en el primer caso aunque no se ha realizado un estudio sistemático que permita cuantificar la influencia de este factor en términos de rendimiento. En forma indirecta el FRR también se ve influido, ya que la búsqueda de niveles más bajos de FAR obliga a ajustar la sensibilidad del sistema de autenticación hacia niveles más estrictos, lo que a su vez produce un aumento de la primera tasa.

Un modelo de ataque *con malware* no supone que los impostores sean humanos, sino que en este caso se trata de programas con la capacidad de sintetizar cadencias de tecleo artificiales. Dependiendo de la capacidad del atacante de observar la cadencia de tecleo del usuario objetivo o de extraer su perfil biométrico, hablaremos de un *malware* con perfil intrausuario o interusuario. Suponemos también que el atacante cuenta con la posibilidad de inyectar las cadencias sintetizadas en el sistema objetivo, remotamente o a través de un dispositivo físico conectado a la terminal objetivo. Es esperable que un ataque con *malware* sofisticado sea capaz de imitar la cadencia de tecleo del usuario legítimo con mayor precisión que un impostor entrenado.

2.2.6.2. Ataques de presentación

En el contexto de los sistemas biométricos, el estándar ISO/IEC 30107 [10] define un *ataque de presentación* como *"la acción de presentarse ante un sistema de captura biométrica con el objetivo de interferir su operación"*. La clase de ataques de presentación puede dividirse en aquellos de *suplantación de identidad* y *ofuscación*.

El objetivo de una suplantación de identidad es obtener acceso a una cuenta y sus privilegios asociados imitando el comportamiento o las características fisiológicas del usuario legítimo. Requiere la construcción de una muestra sintetizada (para el caso de *malware*) o entrenar a un impostor para imitar al usuario legítimo. Los métodos de síntesis existentes serán reseñados en la sección 2.4.6. Luego, esta muestra artificial debe ser inyectada en el sistema objetivo, ya sea remotamente o por medio de un dispositivo físico conectado a la terminal que se desea explotar. Para mejorar la calidad de las imitaciones sintéticas, es posible que un atacante utilice un perfil intrausuario, parcial o total, adquirido mediante una filtración de datos, un ataque exitoso contra el sistema de autenticación, o por medio de un ataque de canal lateral contra este último o contra otros programas en el entorno del usuario.

Una ofuscación busca evadir la identificación o incrementar la tasa de error del sistema de autenticación por encima de márgenes aceptables. Por ejemplo, en [11] y [12] se proponen sendos esquemas para interceptar y modificar la cadencia de tecleo de un usuario utilizando un perfil interusuario. De esta forma, las características comportamentales distintivas son borroneadas con el objetivo de que todos los usuarios compartan una cadencia promedio para la población.

2.2.6.3. Ataques por canal lateral

Se denomina *ataque por canal lateral* a cualquier tipo de ataque que permite extraer información explotando la implementación de un sistema y no el algoritmo en sí. De esta forma quedan excluidos, por ejemplo, errores en el sistema y debilidades inherentes a los algoritmos de cifrado. En particular, cualquier intento de violar la seguridad del sistema por medio de ingeniería social o coerción de los usuarios legítimos no se considera un ataque por canal lateral.

Las fuentes de información que pueden aprovecharse y que han sido aprovechadas para un ataque por canal lateral son diversas. Para extraer información de teclas presionadas se ha utilizado el sonido emanado al escribir [13], la información de acelerómetros y giróscopos [14], e infinidad de otros caminos [15].

En el marco de esta producción, los ataques por canal lateral que nos interesan son aquellos que permiten revelar los tiempos de retención y latencia que refleja la escritura del usuario legítimo. No es este nunca el objetivo final, sino una táctica intermedia hacia el resultado obtenido. Como se reseñará en la sección 2.4.6, los tiempos de retención y latencia filtrados por un ataque de canal lateral pueden ser utilizados para construir un perfil intrausuario parcial que mejore la calidad de síntesis de una muestra artificial; luego, esta muestra puede ser inyectada al sistema de autenticación por medio de un ataque de presentación para ganar los privilegios del usuario legítimo.

Puede que un ataque por canal lateral revele solamente los tiempos de retención y latencia, pero no las teclas asociadas. Veremos en la sección 2.4.7 que otro empleo para los tiempos filtrados consiste en reconstruir el texto que se desconoce y que los origina.

2.3. Perspectiva histórica

Nuestra identidad se revela en la forma en que escribimos. Hace cuarenta años, Gaines *et al.* [7] descubrieron que el análisis de cadencias de tecleo puede conducir a la verificación de la identidad del usuario, dando origen al campo que nos compete. Sin necesidad de restringirse a la autenticación, los ritmos de escritura también pueden revelar ciertas características fisiológicas o discapacidades clínicas [16], y también proporcionar datos suficientes como para identificar estados emocionales [17]. Hablaremos brevemente de la educación de emociones y otras inferencias en la sección 2.4.3; por lo demás, nos enfocaremos en los procesos de autenticación de usuarios y las técnicas de ataque para vulnerarlos [18].

Existe un curioso antecedente histórico del análisis de cadencias de tecleo. Ya a finales del siglo XIX Bryan y Harter, interesados en las características fisiológicas y psicológicas del aprendizaje de la telegrafía, notan que las cadencias de tipeo en código Morse de los operadores telegráficos tienen estilos suficientemente distintivos como para ser atribuidos unívocamente a un cierto telegrafista [19]. Contrariamente a la posibilidad de que fueran el resultado de un entrenamiento premeditado, ellos consideraron estas variaciones como inherentes al sujeto; es decir, productos automáticos e inconscientes de las personalidades de los operadores y la manera en que ellas se expresan. Los autores observaron diferencias tan pronunciadas que, incluso con los medios rudimentarios de análisis y clasificación con los que contaban, podían identificar operadores contando únicamente con la temporización exacta de unas pocas palabras transmitidas. Esta observación fue utilizada con éxito durante la Segunda Guerra Mundial por los Aliados. Los operadores de intercepción de radio británicos podían identificar a los radiotelegrafistas alemanes por su estilo personal de tipeo [20] y de esta manera seguir sus movimientos.

El desarrollo histórico de las técnicas de modelado de la cadencia de tecleo ha seguido, como es esperable, el curso natural de lo simple a lo complejo y de lo central a lo accesorio. Comenzando en los años ochenta por el reconocimiento de la existencia de un factor biométrico hasta ese momento desestimado como la variación entre usuarios de las latencias al teclear, se evaluaron todos los clasificadores de propósito general conocidos y se sugirieron algunos de propósito específico intentando aprovechar las características particulares de los patrones comportamentales estudiados. Incluso décadas después de la introducción de la metodología básica se ha seguido probando la misma con escasas modificaciones excepto la utilización de clasificadores de última generación recientemente desarrollados, como las máquinas de vectores de soporte y los bosques aleatorios. Hablaremos de muchos de ellos en la sección 2.4.2.2.

El análisis de cadencias de tecleo ya ha superado su infancia. Diez años atrás, un sinfín de problemas metodológicos minaban los estudios sobre el tema [21]. La reproducción y

comparación de resultados era difícil cuando no imposible, debido a la falta de conjuntos de datos públicamente disponibles, al uso inconsistente de métricas de error, y a la no utilización de intervalos de confianza e inferencia estadística al informar los resultados de los experimentos. Para la verificación de contraseñas, los métodos simples basados en distancias lograban EERs de alrededor del 10 % [22], mientras que la verificación de texto libre requería muchas muestras grandes, de alrededor de 800 caracteres, para alcanzar una precisión aceptable [8].

Hoy en día disponemos de grandes conjuntos de datos, que serán reseñados en 2.4.1, para entrenar y evaluar modelos de aprendizaje automático de vanguardia. Uno de ellos comprende más de 136 millones de pulsaciones de teclas para casi 200.000 usuarios [23]. En lugar de métodos elementales, contamos por ejemplo con una sofisticada red neuronal recurrente que alcanza alrededor de 5% de EER con un entrenamiento de solo 250 caracteres, incluso después de escalar el sistema para superar los 100.000 usuarios [24].

Sin embargo, a medida que las técnicas maduran y alcanzan una precisión competitiva sin requerir más que una cantidad limitada de datos de entrenamiento, salen a la luz nuevos desafíos más complejos que replican el desarrollo histórico de la biometría tradicional. Esperamos que todo sistema de información, y más aún aquellos relacionados con la seguridad, estén bajo constante ataque [25]; los sistemas de autenticación basados en cadencias de tecleo no son la excepción. Es grave entonces que se haya denunciado recientemente que los métodos más promisorios, es decir aquellos que muestran una precisión más alta para identificar usuarios, generalmente han sido evaluados bajo un modelo de ataque sin esfuerzo [26]. Bajo esta modalidad de evaluación tan benigna, solo se utilizan como adversarios a impostores no preparados. Es decir, se prueban contra el sistema de clasificación muestras generadas por otros usuarios sin ningún conocimiento de los hábitos de escritura del usuario legítimo, y que no hacen ningún esfuerzo por hacerse pasar por él. Esta suposición es ingenua. Un atacante preparado con acceso, al menos parcial, a información intrausuario está en condiciones de forzar el sistema de autenticación con mucha mayor probabilidad de la que las tasas de error evaluadas inocentemente nos harían suponer, como muestra el experimento [26] donde las tasas de falsos positivos aumentaron casi diez veces, hasta un espeluznante 87%, al cambiar el modelo de ataque de esfuerzo nulo a muestras falsificadas con un sencillo modelo gaussiano.

Desde el punto de vista de un atacante, el enfoque habitual para la síntesis de cadencias de tecleo artificiales ha sido el uso de modelos estadísticos de pares de teclas, con simples distribuciones uniformes o gaussianas [27], u otras que proporcionen un mejor ajuste a las muestras empíricas [28]. Por el lado de la defensa, un primer paso hacia la detección de vida actualmente ubicua en biométricos tradicionales ha sido la detección de falsificaciones sintéticas, pero solo para textos cortos y fijos como contraseñas. Bajo un modelo de ataque basado en *malware* y con estrategias básicas de suplantación de identidad, se demostró que el problema era factible de ser abordado [29]. Descubriremos que otros métodos para la creación de falsificaciones sintéticas pueden plantear un desafío inesperado y demandan un enfoque más sólido.

2.4. El estado del arte

A continuación, se reseñará el estado del arte en el tópico de análisis de cadencias de tecleo y aplicaciones. La primer parte de la reseña se realiza desde un punto de vista general, mientras que la segunda se concentra en los tópicos de relevancia para esta tesis. Con el objetivo de lograr la profundidad necesaria, se priorizarán los métodos y aplicaciones de texto libre; aquellos para claves y textos fijos se mencionarán cuando lo exija el orden de exposición, pero no debe considerarse que esta reseña los agote.

El punto de vista general incluye las siguientes subsecciones. La sección 2.4.1 describe los conjuntos de datos públicamente disponibles que han sido utilizados en estudios previos sobre cadencias de tecleo en textos libres. La sección 2.4.2 presenta los métodos de autenticación históricos y actuales, comenzando por aquellos basados en distancias y otros esquemas sencillos en 2.4.2.1, siguiendo por lo más modernos que utilizan aprendizaje automático en 2.4.2.2, y culminando con una perspectiva sobre la fusión de esquemas en 2.4.2.3. La sección 2.4.4 resume estudios sobre consideraciones relacionadas, como la limpieza de los datos en 2.4.4.1, la adaptación gradual de los modelos en 2.4.4.2, y consideraciones sobre seguridad y privacidad en 2.4.4.3.

Los tópicos de relevancia para esta tesis han recibido un tratamiento más exhaustivo. La sección 2.4.5 detalla los estudios previos sobre las distribuciones subyacentes para modelar los tiempos de retención y latencia. La sección 2.4.6 especifica los métodos propuestos para la síntesis de vectores de tiempos y su aplicación en ataques de presentación. Finalmente, la sección 2.4.7 describe los pocos estudios que se encuentran en la literatura del tópico sobre el uso de ataques de canal lateral para la identificación del texto ingresado.

2.4.1. Conjuntos de datos

En los albores de la disciplina, cada experimento de autenticación y cada método eran evaluados con un conjunto de datos privado, propio de los autores, y distinto para cada estudio. Las restricciones impuestas por esta metodología son diversas y negativas. Por ejemplo, es imposible realizar replicaciones de los experimentos originales y es difícil determinar la generalizabilidad de las conclusiones pues estas pueden aplicar exclusivamente al entorno de captura [21]. Un problema más sutil es que los conjuntos de datos pequeños pueden no permitir extraer conclusiones con significancia estadística.

Hoy en día existen suficientes conjuntos de datos sobre cadencias de tecleo, públicamente accesibles y de gran tamaño. A continuación, reseñamos algunos de ellos, que han sido ampliamente utilizados en experimentos anteriores y algunos de los cuales utilizaremos en el transcurso de esta tesis. Una descripción mucho más detallada de estos últimos junto con los criterios de selección puede leerse en la sección 5.1. Los tipos de tarea de escritura se describen en 5.1.2. A lo largo de esta tesis nos concentraremos en el texto libre, por lo que los conjuntos de datos de claves o textos fijos no se incluirán en esta sección; tampoco aquellos que no estén disponibles públicamente.

136M. Un conjunto de datos de texto libre muy reciente y probablemente el más extenso a disposición del público, aunque con pocas sesiones (y muy cortas) por usuario. Contiene

aproximadamente 136 millones de caracteres tecleados por unos 200.000 usuarios. Fue utilizado en [23] para realizar análisis estadístico del comportamiento promedio durante la escritura en la población general, y en [30] para evaluar la escalabilidad masiva de los sistemas de autenticación basados en cadencias de tecleo.

GP. El primer conjunto de datos públicos de texto libre, utilizado por los autores de [31, 8] para evaluar las métricas R y A, que se describen en la sección A.8. Es pequeño en contraste con los posteriores, e incluye muchos usuarios con una única sesión, para ser utilizados como impostores.

Nombre	Estudio	Año	C	T	Caracteres	Usuarios	Período
GP	[8]	2005	✓		≈ 400K	40+165	6 meses
KM	[40]	2012	✓	✓	≈ 100K	20	1 semana
VURAL	[39]	2012	✓	✓	≈ 840K	39	2 días
PROSODY	[37]	2014	✓	✓	≈ 1,6M	400	1 mes
LSIA	[33]	2015	✓		≈ 10M	200	4 años
SUN	[38]	2016	✓	✓	≈ 2,2M	148	1 a 3 días
MURPHY	[36]	2017	✓		≈ 13M	100	2,5 años
136M	[23]	2018	✓		≈ 136M	200K	1 día

Cuadro 2.1: Conjuntos de datos de texto libre públicamente accesibles

KM. Utilizado en [32] para evaluar si las tareas de composición o transcripción producirían perfiles lo suficientemente similares como para ser usados indistintamente con el propósito de simplificar futuras adquisiciones de datos. Se encontró que los voluntarios eran más propensos a contribuir con textos transcritos que componiendo originales. Contiene dos sesiones de composición y dos sesiones de transcripción para 20 usuarios.

LSIA. Contiene sesiones de escritura en texto libre registradas durante el trabajo diario, en un entorno sanitario, durante más de un año. Los usuarios, en su mayoría médicos, trabajaban en turnos rotativos y de guardia. Fue utilizado en [33] y [34] para evaluar el rendimiento de los métodos de autenticación en condiciones realistas, y en [35] para estimar el parámetro p óptimo para la distancia de Minkowski.

MURPHY. Sin ser tan extenso como 136M, este conjunto de datos tiene muchos menos usuarios, pero con muchas más sesiones para cada uno. Incluye unos 13 millones de caracteres, con un promedio de 125.000 caracteres por usuario. Fue utilizado por [36] para replicar el experimento de Gunetti-Picardi [8].

PROSODY. Utilizado en [37] para estudiar indicios de intención engañosa reflejados en la cadencia de tecleo. Contiene texto compuesto con intención de veracidad o de engaño, y texto transcrito donde el usuario está de acuerdo o en desacuerdo con el contenido. Incluye temas controvertidos como el matrimonio homosexual y el control de armas para inducir fuertes respuestas emocionales en los usuarios.

SUN. El objetivo de este conjunto de datos es permitir la exploración de los efectos que distintas configuraciones de teclado producen sobre las cadencias de tecleo. Fue capturado por los autores de [38].

VURAL. Notable por contener video del movimiento de manos y de las expresiones faciales de los usuarios al escribir, junto con texto fijo y libre [39]. Es también el que incluye las sesiones de escritura más prolongadas, muchas veces excediendo los 10.000 caracteres, aunque con solo 39 usuarios.

El cuadro 2.1 resume las características de los conjuntos de datos anteriores; C y T corresponden a las tareas de composición y transcripción, mientras que la columna período indica durante cuánto tiempo fueron seguidos los usuarios. Una lista exhaustiva y actualizada de los conjuntos de datos disponibles hoy en día, tanto de texto libre como de texto fijo, puede consultarse en [41].

2.4.2. Autenticación de usuarios

2.4.2.1. Distancias y métodos simples

A pesar de ser muy sencillas de implementar, las distancias normalizadas han arrojado excelentes resultados en multitud de experimentos ([42], [5] y [43], entre otros). Las diversas metodologías de reporte de rendimiento hacen complicado compararlos, pero en general presentan un ERR entre el 10% y el 15%, con valores de FAR y FRR más bajos si se compromete uno de ellos en beneficio del otro. La distancia de Manhattan suele producir un error de clasificación marginalmente más bajo. Su simplicidad de implementación las hace ideales para emplearlas como caso base contra el cual comparar clasificadores más avanzados.

En [44], Araujo *et al*. describen el uso de una variante de la distancia de Manhattan que brinda excelentes resultados según sus propias pruebas. El rendimiento ha sido confirmado independientemente por Killourhy y Maxion [5], bajo cuyo conjunto de entrenamiento y evaluación este método obtiene el menor EER y un punto de FRR cero entre los mejores de todos los clasificadores considerados. También Bleha *et al*. proponen una modificación [42], esta vez tanto a la distancia euclídea como de Mahalanobis, cuyo fin sería permitir que se acomoden en forma transparente a la variación en el largo de las claves. La misma consiste en escalar la distancia original por el producto de las normas de la nueva observación y de la media correspondiente al usuario. El propósito no se habría realizado, con la penalización adicional de que el efecto de la reforma sobre la clasificación es deletéreo; al replicar el experimento [22] tanto el EER como el punto de FRR cero serían peores que los de su contraparte original.

Una manera natural de considerar las correlaciones entre mediciones sucesivas de parámetros característicos es utilizar la distancia de Mahalanobis [45], que mide las desviaciones sobre las direcciones de los ejes principales dadas por los autovectores de la matriz de covarianza. El mérito de proponer esta métrica para la clasificación de usuarios en base a su cadencia de tecleo corresponde a Bleha *et al*. [42] quienes reportan un FAR de 2,8% y un FRR de 8,1% sobre más de 500 clasificaciones con 39 usuarios, divididos en 14 legítimos, 13 impostores entrenados, y 12 impostores no entrenados. Contrariamente a lo esperado, la consideración de las correlaciones no siempre es provechosa. Calot *et al*. [43] han comparado

el rendimiento de la clasificación con la distancia euclídea normalizada y con la distancia de Mahalanobis utilizando el área bajo la curva de ROC, concluyendo que los resultados con ambas son virtualmente idénticos para una cantidad de hasta veinte eventos –registran una variación del 0,24% para este último valor–, utilizando el mismo juego de datos de entrenamiento y prueba que en [5]. Los autores hacen notar que desde cualquier otro punto de vista la distancia euclídea normalizada resulta más conveniente, ya que sus requerimientos de almacenamiento y de potencia de cómputo son menores al no requerir persistencia ni cálculo de la matriz de covarianza.

El clasificador k-NN, de vecinos más cercanos, fue utilizado por Cho *et al*. [46] como caso base para comparar contra un clasificador implementado por medio de una red neuronal autoasociativa de múltiples capas. El mejor resultado se obtuvo para $k = 1$. Al replicar el experimento, k-NN resulto ser el mejor clasificador en base al punto de FRR cero y el segundo por un escaso margen de 0,4% en base al EER [5]; la red neuronal, en cambio, resulto consistentemente peor. En [8] este método es revisitado y los autores logran, en concordancia con los anteriores, un FAR significativamente bajo en el orden del 0,045%, aunque a costa de un tiempo de autenticación inaceptable, que excede los 20 segundos con solamente cinco usuarios y se eleva hasta casi 40 con 15 usuarios. Hu y Gingrich [47] mediante diversas consideraciones logran reducir los tiempos de [8] a entre 16 y 19 segundos para la misma cantidad de perfiles. Si bien estos números deben entenderse bajo la consideración de que aplican a un *hardware* no especificado del año 2008, no dejan de ser sorprendentemente elevados.

El algoritmo de *k-means* no es estrictamente un clasificador sino un algoritmo de *clustering* que ha sido utilizado en varias ocasiones [48, 46] como caso base para generar un clasificador sencillo y que permite comparar contra otros de mayor complejidad. Obaidat y Sadoun [48] reportan un FAR y FRR de aproximadamente 10% y 15%, muy cercanos a los obtenidos en sus pruebas con distancia euclídea.

El conteo de valores atípicos es una puntuación muy simple que indica la cantidad de valores de los parámetros característicos que difieren de la media en más de una cierta cota, medida en desvíos estándar (ver apéndice A.7). A pesar de su simplicidad permite obtener resultados similares a muchos clasificadores más complejos [5, 49].

No todos los clasificadores idóneos para claves estáticas funcionan bien con textos libres y en muchos casos ni siquiera es evidente como adaptarlos para esta situación. Por ejemplo, Monrose y Rubin [50] han evaluado extraer los promedios de tiempos de latencia y retención para digrafos, contrastándolos la muestra y la plantilla del usuario utilizando la distancia euclídea. Los resultados son decepcionantes; al pasar de texto fijo a texto libre la tasa de error se hunde, desde el 10 % para el primer caso hasta 77% para el segundo. Entre los clasificadores de propósito general, los basados en espacios métricos han dado buenos resultados [51, 3].

Sin embargo, dos nuevas técnicas de clasificación de propósito específico denominados *métrica R* (ver apéndice A.8) y *métrica A*, inicialmente propuestos por Bergadano, Gunetti y Picardi en [31] y [8], se llevan las palmas y han sido utilizadas por otros autores [52] luego de optimizarlas para su ejecución en tiempo real, ya que el costo computacional de estas técnicas no es irrelevante. Las tasas de error reportadas, en el orden del 7%, superaron ampliamente

a todos sus competidores al momento de la publicación. A la luz de la evolución de los algoritmos que se describirán en la sección siguiente, cabe criticar que el tamaño requerido de las muestras para alcanzar dichas tasas de error era muy grande, de entre 700 y 900 caracteres, y que se requerían muchas muestras para entrenar el perfil del usuario. Intentando determinar la capacidad de escalar el método y con otro conjunto de datos de evaluación, Murphy *et al.* [36] replicaron el experimento de Bergadano, Gunetti, y Picardi, obteniendo tasas de error en el orden del 10%.

2.4.2.2. Aprendizaje automático

Yu y Cho [53] inauguraron el uso de máquinas de vectores de soporte (SVM) para la clasificación de cadencias de tecleo. Utilizaron un conjunto de datos de entrenamiento que comprende entre 150 y 400 ingresos de claves con largos de entre seis y diez caracteres por parte de 21 participantes; adicionalmente, se incluyeron cinco repeticiones de la clave de cada usuario, ingresadas por quince impostores con entrenamiento. Los resultados se reportan bajo la forma del FRR en el punto de FAR cero, obteniéndose valores que oscilan entre 1,25% y 4,68% con una tendencia a obtener valores inferiores para claves más largas. Aunque el resultado no parezca sorprendente en vista del rendimiento esperado por parte de un clasificador SVM, debe tenerse en cuenta que la utilización de impostores con entrenamiento como desafío es mucho menos benigna que la practica usual de emplear ataques de esfuerzo cero. Para alcanzar estos valores no se utilizó la totalidad de la muestra temporal; los autores proponen un ingenioso esquema que por medio de algoritmos genéticos permite extraer los componentes relevantes para mejorar la eficacia en la clasificación. El empleo ingenuo de SVM con la muestra completa eleva el FRR en el punto de FAR cero a un promedio de 15 % con picos de hasta 20 %.

Posteriormente, Sung y Cho [54] han propuesto ciertas mejoras al esquema de selección de componentes que permiten disminuir los errores en la clasificación. La comparación con los resultados anteriores no es inmediata ya que en esta ocasión se reporta un FAR promedio de 3,85% y un FRR de 13,10% en lugar del FRR para el punto de FAR cero. Con valores respectivos para el anterior de 13,13% y 7,25 %, el nuevo método lograría una mejora en el FAR a costa de una degradación del FRR. Otra instancia del uso de SVM para la clasificación de patrones de tecleo puede encontrarse en [55]. Con secuencias de unos 500 caracteres, los autores reportan tasas de error cercanas a cero, pero en un conjunto de datos con solo 34 usuarios.

Cho *et al.* [46] han ensayado un clasificador implementado en base a una red neuronal autoasociativa de tres capas, en donde todas ellas tienen tantas neuronas como largo tiene la muestra de tiempos a analizar. Para un usuario dado, se forma una red cuyos pesos se inicializan en forma aleatoria y se ingresan a la misma los vectores de entrenamiento utilizando el algoritmo de *backpropagation* [56] para el aprendizaje. El mismo vector que se ingresa a la entrada se utiliza como salida esperada, forzando a la red a codificar las características salientes del usuario en la capa oculta. De esta manera, cuando el entrenamiento ha finalizado, al presentarse en la entrada un vector de similares características a las del usuario legítimo se obtendrá a la salida un vector muy parecido; inversamente, un vector generado por el tecleo de un impostor, al no corresponderse en las características fundamentales con el patrón codificado en la capa oculta, debería producir una

salida que difiera sustancialmente de la entrada. Para cuantificar la diferencia entre ambos vectores se puede utilizar cualquiera de las distancias mencionadas anteriormente; los autores no aclaran la utilizada en sus pruebas.

Una aproximación más convencional al problema de la clasificación de usuarios en base a su cadencia de tecleo mediante el uso de un perceptrón de tres capas fue realizada por Abbas *et al.* [49]. Debido a su simplicidad, el perceptrón fue una de las primeras redes neuronales artificiales en ser estudiadas [57] e implementadas y se cuenta entre las que mayor escrutinio han recibido, tanto en su forma original con dos capas como en las extensiones que agregan una o varias capas ocultas y otras variantes. Con un modelo más sencillo que el de la red autoasociativa, la última capa del perceptrón contiene una única neurona cuya intensidad de salida refleja la cercanía del vector utilizado como entrada al modelo inferido por la red neuronal durante el entrenamiento. Una vez más, se requiere una red neuronal para cada usuario que se pretende enrolar. Los autores proponen, sin justificación alguna, la utilización de dos nodos en la capa oculta por cada tres nodos en la capa de entrada y aprendizaje por *backtracking*.

Con valores de FAR y FRR de 22% y 20% respectivamente para dos intentos sucesivos en el ingreso de la clave y un FRR del 41% (FAR no reportado) para un único intento, los resultados logrados con el perceptrón de tres capas no resultan prometedores. Inversamente, Cho *et al.* reportan "autenticación perfecta" utilizando como criterio el punto de FRR cero para aproximadamente la mitad de los usuarios y para el resto una tasa de error promedio del 1% con un máximo del 4 %. Los resultados de Killourhy y Maxion [5] replican el desempeño pobre del perceptrón de tres capas, que resulta consistentemente el peor de los clasificadores estudiados, pero contradicen las afirmaciones de Cho *et al.* respecto del desempeño de la red autoasociativa.

Obaidat y Sadoun [48] han comparado múltiples tipos adicionales de redes con una cantidad variable de capas ocultas y estrategias de aprendizaje tan diversas como *backpropagation*, *counterpropagation* y LVQ (entre otras) reportando errores de clasificación muy cercanos a cero e incluso valores nulos de FAR para algunas como RBFM y ARTMAP, aunque no todas resultaran tan promisorias e incluso algunas puntuaran por debajo de otros clasificadores utilizados como caso base.

La autenticación de claves estáticas utilizando arboles de decisión ha sido explorada por Sheng *et al.* [58], obteniendo resultados promisorios. Para la clasificación se particionan los vectores característicos y se generan hasta ocho arboles de decisión basados en subconjuntos de los parámetros; el proceso acepta al usuario como legítimo si al menos tres de los árboles así lo convalidan. Un FAR del 0,88% con una tasa de rechazos del orden del 10% pueden obtenerse con frases de entre treinta y cuarenta caracteres. Esta debe considerarse una longitud mínima, ya que la eficiencia en la clasificación decae fuertemente al disminuir dicho largo. Los autores notan que no es requisito para una implementación practica exigir una clave de tal extensión, sino que basta con considerar nombre de usuario, clave, apellido y nombre del sujeto registrado para llegar a la cantidad de caracteres necesaria. Es destacable que solo nueve vectores de entrenamiento por usuario sirven para alcanzar el rendimiento mencionado. Las técnicas utilizadas por los autores para lograr una exigencia tan baja son dos. En primer lugar, la generación de nuevos vectores característicos en forma aleatoria utilizando la distribución implicada por el conjunto de entrenamiento permite, contrariamente a lo que

sucedería en los métodos que utilizan espacios métricos, evitar el sobreentrenamiento del árbol. En segundo lugar, la representación del vector característico en una base ortogonal obtenida por medio de *wavelets* o de una transformada discreta de Fourier agrega un grado de variedad en la representación de la entrada que hace posible variar el FAR y el FRR con mayor flexibilidad y así alcanzar un compromiso razonable.

Killourhy y Maxion [6] aplican el algoritmo de bosques aleatorios al problema de clasificación de cadencias de tecleo en un ambiente poco convencional. En lugar de utilizar un teclado estándar, estudian la posibilidad de reconocer usuarios en base a claves cortas tipo PIN ingresadas en un teclado numérico, como los que pueden encontrarse en cajeros automáticos o terminales de autenticación para tarjetas de crédito y débito. Este ambiente presenta en forma simultánea diversos problemas metodológicos, ya que además de utilizarse claves cortas (once números), la variedad de las mismas está restringida al mínimo por el tipo de teclado y la disposición geométrica de las teclas hace que la mayoría de las veces los usuarios utilicen un único dedo para el ingreso de información, lo que hace esperar mayor similaridad en los patrones de tecleo. A pesar de las complicaciones, los resultados son sorprendentemente robustos; un FAR de 0,46% *versus* un FRR de 12,5% posicionan a la clasificación con bosques aleatorios entre las más destacadas de todos los algoritmos aquí reseñados. Adicionalmente, el EER de 1,45% alcanzado es uno de los mejores reportados en la literatura. Es importante notar que para alcanzar un tal FAR se utilizaron cien repeticiones de entrenamiento, lo cual constituye un numero exageradamente alto para una implementación práctica, pero que a pesar de ser mayor no dista demasiado del numero usualmente empleado para evaluar otros clasificadores.

El exponente más actual de aprendizaje automático aplicado a la autenticación de usuarios utilizando cadencias de tecleo es el de Ancien *et al*. [24, 30]. Este estudio destaca no solo por el método y su rendimiento sino también por la dificultad del protocolo de evaluación, en el que el clasificador propuesto sorprende con bajas tasas de error. Los autores proponen la utilización de una red neuronal recurrente, del tipo siamesa, con dos capas LSTM de 128 neuronas; una implementación bajo Keras - Tensorflow [59] fue puesta a disposición de los lectores. Lo más sorprendente es la escasa cantidad de información por usuario utilizada para entrenar la red neuronal; hay solo 15 sesiones por usuario en el conjunto de datos de evaluación elegido, que entre ellas suman no mucho más de 250 caracteres. El conjunto de datos cuenta con uno 200.000 usuarios y aproximadamente 136 millones de caracteres en total, lo que lo hace óptimo para evaluar la posibilidad de escalar a tamaño masivo los sistemas de autenticación por medio de cadencias de tecleo. Los autores reportan un EER de 4,8% para mil usuarios, con una única sesión de evaluación por usuario, de aproximadamente 50 caracteres. Al incrementar la cantidad de usuarios por encima de 100.000, el rendimiento decrece un 5% en términos relativos.

Un antecedente del anterior es [60], que utiliza la misma arquitectura de red neural para la fusión de ocho modalidades biométricas. Las métricas de error utilizadas son diferentes; los autores reportan una tasa de falsos positivos del 0,1%, pero una tasa de falsos negativos del orden del 20%. Considerando que se emplea una ventana de observación de solo tres segundos, este ultimo valor no puede ser juzgado como malo. Sin embargo, como el conjunto de datos de evaluación cuenta únicamente con 37 usuarios es difícil determinar la generalizabilidad de los resultados.

2.4.2.3. Fusión de esquemas

De la misma forma que en los métodos de ensamble como los bosques aleatorios se reduce el error de clasificación mezclando los resultados de múltiples clasificadores relacionados, el principio se puede aplicar a distintos algoritmos que no tengan relación entre sí. Haider *et al.* [49] reportan una mejora consistente en el FAR y FRR al utilizar en conjunto métodos tan disímiles como la lógica difusa, las redes neuronales y el conteo de valores atípicos, a pesar de que algunos de ellos den por separado resultados poco alentadores. Son similares las conclusiones en [61], donde la combinación de distancias normalizadas y un clasificador de propósito específico supera a estos métodos por separado. En este último se aplica una suma ponderada de los resultados de ambos clasificadores variando los pesos para cada usuario.

2.4.3. Educción de emociones y otras inferencias

Un breve apartado de [50] puede considerarse el primer intento en la disciplina de inferir información adicional a la identidad del usuario. Los autores notan que entre los participantes del experimento se puede distinguir a los zurdos de los diestros ordenando las latencias de los digrafos, ya que en los primeros todas aquellas que corresponden a presiones sucesivas de teclas del lado izquierdo del teclado tienen en conjunto un promedio de duración menor que las del lado derecho e inversamente en los segundos. No se han llevado adelante investigaciones adicionales para elucidar qué otras características físicas del usuario pueden inferirse a partir de patrones de gran escala que emerjan de un conjunto de datos de entrenamiento.

La influencia de factores emocionales ha sido estudiada con mayor profundidad. Vizer *et al.* [62], analizando el efecto del estrés físico y cognitivo sobre los patrones de tecleo, concluyen que pueden inferirse las variaciones graduales o repentinas del estado del usuario en esta dimensión emocional con la suficiente confiabilidad como para utilizarlo en aplicaciones médicas en las que este factor pueda ser de interés, aunque no crítico. Mas general es el enfoque de [17], donde se intentan clasificar una docena de dimensiones emocionales entre las que se incluyen aburrimiento, frustración, felicidad, ansiedad y otras utilizando una metodología de muestreo en la cual se interrumpe aleatoriamente la tarea cotidiana del usuario para solicitarle que reporte su estado emocional en el momento para luego asociarlo con el registro de tecleo. Los resultados son promisorios, ya que a pesar de las dificultades metodológicas y de utilizar un modelo muy sencillo se logra una eficacia en la clasificación cercana al 80%. Una reseña actual de los primeros métodos y resultados obtenidos en la educción de emociones utilizando cadencias de tecleo puede consultarse en [63].

Mas recientemente, Calot *et al.* [64, 65] han demostrado que los sistemas de autenticación basados en cadencias de tecleo son robustos ante las variaciones emocionales de los usuarios. En un experimento en el que capturaron muestras de tecleo bajo distintos estados emocionales demostraron que la distancia euclídea se ve escasamente afectada, pero que otras métricas sufren consecuencias negativas. El estado emocional fue inferido por medio de técnicas encefalografías y un cuestionario de autoreporte.

2.4.4. Consideraciones relacionadas

Cuando el fin es la autenticación de usuarios en base a su cadencia de tecleo, existen diversas consideraciones relacionadas a tener en cuenta más allá de la precisión del algoritmo. Como nos enfrentamos con datos ruidosos, será necesario primero limpiarlos como se reseña en la sección 2.4.4.1, y como el comportamiento del usuario varía en forma suave pero constante a lo largo del tiempo, los modelos deben incluir adaptación gradual como se reseña en la sección 2.4.4.2. Finalmente, el almacenamiento de las muestras y los perfiles biométricos requiere consideraciones de seguridad y privacidad, como se reseña en la sección 2.4.4.3.

2.4.4.1. Limpieza de los datos

A diferencia de las características biológicas como las huellas dactilares o la estructura facial, que salvo en caso de accidentes o traumas de gravedad suelen conservarse estables a lo largo del tiempo, los patrones de comportamiento son intrínsecamente susceptibles de ser modulados por toda clase de factores ambientales y emocionales, además de presentar variaciones significativas que no pueden ser atribuidas más que al carácter ruidoso de las variables que resultan inevitablemente del comportamiento que se está midiendo o que lo originan. Durante la acción de tecleo, adicionalmente a las causas genéricas de deriva temporal que se han descrito, se registran en forma continua todo tipo de fenómenos que alteran la cadencia natural de un usuario. Pueden presentarse pausas al teclear por causas externas como interrupciones e internas para elaborar lo que se está escribiendo o leer lo que se pretende copiar, cualquier otro tipo de distracción, alteraciones del ritmo por mal funcionamiento de alguna tecla suelta o trabada, e infinidad de otros motivos que producen dispersión de las latencias entre teclas o el tiempo de presión de cualquiera de ellas.

Para evitar la contaminación de los modelos con valores de entrenamiento que no reflejan el estilo de tecleo del usuario es necesaria una estrategia de *limpieza de los datos*. Este postulado puede parecer evidente pero no ha sido suficientemente enfatizado en la literatura; solo unos pocos artículos señalan explícitamente la estrategia utilizada si acaso lo es alguna, o analizan el efecto detrimental de su ausencia. Sin embargo, es posible que las características de muchos de los clasificadores utilizados mitiguen naturalmente parte o la totalidad del efecto deletéreo de estos valores atípicos.

Una estrategia muy sencilla que presupone que las componentes de la muestra siguen una distribución normal es la adoptada en [66], donde se calcula la varianza y la media muestral para luego descartar todos los valores que se alejan más de tres desvíos estándar de la media, y se repite el proceso hasta que no haya valores para descartar. Es similar la estrategia de [17] con la excepción de que, por tratarse de textos libres, se han utilizado doce desvíos estándar como umbral de detección de valores atípicos. La cantidad de muestras descartadas asciende a 0,85% y 0,07% respectivamente por lo que, a pesar de que no se discute el efecto final sobre la tasa de errores del clasificador, la limpieza de los datos no parece impactar negativamente en la cantidad de reentrenamiento requerido y es esperable que tenga una influencia positiva. Los autores de [66] notan que los valores descartados suelen aparecer en forma aislada en los vectores característicos, reforzando la idea de que

probablemente se deban a efectos locales o espúreos y no a un arco comportamental que obligue a descartar la muestra entera.

2.4.4.2. Adaptación gradual

Así como las variaciones espúreas en la cadencia de tecleo por causas externas a los factores biométricos motivan el filtrado del conjunto de entrenamiento, debe también tenerse en cuenta que esta evoluciona en forma gradual, pero de manera suficientemente significativa como para validar el interrogante por la necesidad de modelos adaptativos. Algunas propuestas, como la de Monrose *et al*. [67], han sido diseñadas desde el principio con este requerimiento en mente y otras pueden ser fácilmente modificadas para brindar esta última característica. Por ejemplo, en todos los clasificadores elementales basados en espacios métricos no es necesario fijar el´ conjunto inicial de entrenamiento si se utilizan medias móviles [68] y el ingreso de cada nuevo vector característico puede reajustar el modelo, al menos en parte, al

12	12
34	34
56	56
78	78

12	13
34	35
57	56
78	79

12	14
34	36
58	56
78	80

Cuadro 2.2: Ejemplos de clave endurecida: registración, primer intento, segundo intento. En gris las partes utilizadas para reconstruir *hpwd*.

patrón de comportamiento actual del usuario [69].

Con la notoria excepción de la tesis doctoral de Killourhy [40], en la literatura suele presuponerse que el complejo de patrones motores e intelectuales que originan la cadencia de tecleo no sufre modificación a lo largo del tiempo o a lo sumo lo hace dentro de márgenes limitados. La extensión temporal de los experimentos, que nunca excede más que algunos meses, aunque generalmente es mucho más reducida, invita a convalidar implícitamente esta hipótesis ya que es imposible comprobar la influencia de un proceso de mediano plazo. Sin embargo, algunos de estos como el progreso esperable de un usuario con habilidad rudimentaria o la cristalización y optimización gradual en el tecleo de secuencias repetidas en el curso de la tarea profesional saltan a la vista y evocan la existencia de muchos otros procesos desconocidos o que no han sido analizados aún.

Tentativamente, puede afirmarse en base al repaso de los experimentos de mayor duración que la cadencia de tecleo para claves estáticas evoluciona en el sentido de la minimización de la varianza en los componentes del vector característico hasta un cierto umbral propio del usuario por la cristalización de los procesos motores debido al entrenamiento y hacia la estabilización de la media en busca de latencias más bajas que la inicial por el progreso de la habilidad mecanográfica en dicha secuencia específica. Esto no

deja de ser intuitivo y, positivamente, juega a favor de la eficacia de los clasificadores siempre que el conjunto de entrenamiento sea lo suficientemente grande como para capturar muchas observaciones una vez que la forma de teclear se ha estabilizado. Adicionalmente, señala que un modelo adaptativo no puede sino verse beneficiado por la estabilización ya que todas las nuevas observaciones contendrán menos ruido inicial y por lo tanto tendrán menos semejanza con las de otros usuarios o impostores. No es tan claro el efecto de los procesos de mediano plazo en el caso de autenticación de textos libres [8], lo que puede deberse a que la forma actual de modelarlos no refleja adecuadamente las características significativas de los mismos [40].

2.4.4.3. Seguridad y privacidad

Poca o ninguna atención se ha prestado a los requerimientos necesarios para salvaguardar las muestras recabadas durante el entrenamiento y evitar la extracción de datos significativos o valiosos de los modelos generados, entre ellos los propios patrones característicos del usuario para evitar que un impostor pueda entrenarse para replicarlos. Al igual que en el caso anterior, un contraejemplo es la propuesta de clave endurecida de [67, 70] que logra a la vez adaptabilidad y seguridad a costa de reducir la exigencia sobre las tasas de error en la clasificación. El ingenioso esquema reduce la información aportada por cada parámetro característico ϕ_i (en la notación de los autores) a un valor binario, que indica si este supera o no un cierto umbral t_i que puede ser un parámetro fijo del sistema o recalculado en base al entrenamiento.

Se denominan parámetros característicos distintivos a todos aquellos cuyas mediciones para un usuario legítimo se encuentran consistentemente, bajo una definición basada en el conteo de desvíos estándar, por encima o por debajo del umbral que corresponde. La letra m se utiliza para denotar la cantidad de ellos, independientemente del largo de la clave; los restantes parámetros se descartan y no tienen influencia adicional en este esquema de autenticación. Cuando un usuario se registra una clave secreta $hpwd \in \mathbb{Z}_q$, donde q es un primo suficientemente grande para propósitos criptográficos, se genera al azar y la misma se fracciona en $2m$ partes divididas en m conjuntos s_1, ..., s_m de dos partes, llamémoslas izquierda y derecha, de forma tal que con una parte de cada conjunto se pueda reconstruir el numero original. El conjunto total se cifra con la clave que el usuario eligió para la registración. Tanto en el momento del entrenamiento como de la autenticación se intenta reconstruir *hpwd* tomando, para cada parámetro característico distintivo ϕ_i, la parte izquierda del conjunto s_i si el valor medido es inferior al umbral t_i o la derecha en caso contrario. Si la autenticación es exitosa, se altera el otro valor de manera reversible. Un ejemplo simplificado de este proceso donde *hpwd* = 12345678 se muestra en el cuadro 2.2. Los autores proponen tres métodos sobre el mismo esquema para generar las particiones y manipularlas sin necesidad de desencriptar los conjuntos que conforman la clave endurecida, utilizando polinomios, exponenciación en un cuerpo finito y espacios vectoriales.

Claramente, si la cadencia con la que se ha ingresado la clave no se corresponde con la del usuario legítimo, las partes extraídas de los conjuntos s_1, ..., s_m no permitirán reconstruir *hpwd* y sí lo harón en caso contrario. Lo que, es más, si un atacante logra acceso al conjunto de partes encriptado, esta información no le basta para deducir *hpwd* ni cuales son los parámetros característicos distintivos. A un ataque por fuerza bruta para adivinar la clave de

registración se le agrega un factor multiplicativo de 2^m ya que se debe tambien considerar una selección de m partes de los $2m$ conjuntos para poder reconstruir *hpwd*. Es importante notar que la clave endurecida *hpwd* se conserva estable aun ante cambios en la cadencia de tecleo del usuario legítimo, por lo que puede ser utilizada para propósitos de largo plazo como encriptación de archivos. Los autores no analizan el resultado en función de las métricas de eficiencia usuales, sino que en vista de los requerimientos adicionales de seguridad consideran la entropía agregada a la clave, que en las condiciones del experimento alcanzan un poco más de seis bits para un promedio de reintentos por falso negativo menor a dos.

2.4.5. Distribuciones subyacentes

Para cada tecla en un texto fijo y para cada tecla precedida por un cierto contexto en texto libre, el conjunto de todas las observaciones de sus parámetros temporales que se encuentran almacenadas en el perfil biométrico del usuario conforma una cierta distribución empírica. Esta distribución empírica y discreta es un reflejo de una distribución ideal, que asumimos continua, y que refleja las características fisiológicas y comportamentales del usuario, como el tamaño de sus manos, el largo de sus dedos, y su pericia en la escritura mecanográfica. Podemos pensarla como una distribución estacionaria, que refleja un promedio de todas sus condiciones emocionales posibles, o podemos intentar aproximarla en base al contexto emocional u de otro tipo. No es disparatado suponer que la distribución de tiempos se modifica a lo largo del día con el cansancio del usuario, o con su nivel de cafeína en sangre si gusta del café.

Determinar la forma de las distribuciones de parámetros temporales subyacentes es un elemento crucial para el modelado preciso de las cadencias de tecleo. Aun así, este problema no ha sido abordado explícitamente, con la excepción de unos pocos autores y siempre en casos particulares. Cuando tratamos con texto libre, campo al que pertenecen todos los problemas explorados en esta tesis, el interrogante sigue abierto en su forma general.

El enfoque temprano en la disciplina ha sido presuponer que las variables subyacentes son normales, sin cuestionar la doctrina común de la estadística que afirma que casi todas las variables, si cumplen ciertos escasos requisitos, lo son. En vista de que muchos de los métodos de verificación reseñados en la sección 2.4.2.1 parecen funcionar bastante bien usando solo dos parámetros (media y varianza) y presuponiendo normalidad para los modelos de tiempo, resulta evidente que este postulado fundacional no se encuentra demasiado lejos de la realidad. Por ejemplo, Stefan *et al.* [27, 29] exploran la síntesis de cadencias de tecleo utilizando ruido uniforme y ruido gaussiano, al igual que [26, 71], y no con poco éxito.

Quizás sea esperable que, al verificar contraseñas o textos cortos y fijos, que se escriben consistentemente con una cadencia suficientemente estable, nos encontremos con perfiles de tiempo distribuidos en forma normal. Pero al tratarse de texto libre, debemos considerar también pausas y vacilaciones de todo tipo. Pensar, mirar el teclado, descansar, atender interrupciones externas, etc., son distracciones que ocurren invariablemente por corto que sea el intervalo de escritura. Estas distracciones sesgan las distribuciones de tiempos observados hacia la derecha, cambiando su forma y agregando colas largas. Si tenemos en cuenta que la mayoría de las distancias y los métodos de clasificación son sensibles a las discrepancias entre el modelo asumido y los datos empíricos, es desconcertante que un

estudio sistemático de las formas de los histogramas no fuera un paso temprano en la disciplina.

Probablemente el primer intento de considerar las particularidades del texto libre sea el de Montalvão *et al.* [72, 73], quienes proponen la ecualización del histograma empírico de tiempos para obtener una distribución log-normal, que brinda un buen ajuste para los tiempos de latencia en el conjunto de datos evaluado. Como el principal objetivo del estudio fue mejorar el rendimiento de los algoritmos que no incorporan ecualización de distribución de intervalos, los autores no intentaron justificar la elección de la distribución log-normal más allá del ajuste empírico y la reducción de la tasa de error, que resulto ser significativa. Contra un 15 % de error alcanzado por los métodos básicos este fue reducido a aproximadamente un tercio, en el orden del 5%, luego de normalizar los histogramas presuponiendo que responden a una distribución log-normal.

Año	Autores	N	L	E	Otras distribuciones	Estudio comp.	Texto libre
<2010	Muchos	✓				✗	✗ / ✓
2006	Montalvão & Freire [72]		✓			✗	✓
2006	Montalvão *et al.* [73]		✓			✗	✓
2010	Stefan & Yao [27]	✓			Uniforme	✗	✗
2011	Rahman *et al.* [26]	✓				✗	✗
2012	Stefan *et al.* [29]	✓			Uniforme	✗	✗
2013	Rahman *et al.* [71]	✓				✗	✗
2014	Chukharev-H. [74]			✓		✗	✓
2015	Iorliam [75]		✓		Benford/Zipf Exponencial	✓	✗
2015	Monaco & Tappert [28]		✓			✗	✓
2015	Monaco [76]		✓		Benford/Zipf Exponencial	✓	✓
2016	Monaco *et al.* [77]		✓			✗	✓
2018	Migdal & R. [78, 79]	✓	✓	✓	Gumbel Otras 15	✓	✗

Cuadro 2.3: Resumen de las distribuciones consideradas explícitamente en la literatura

Usando la misma distribución, Monaco *et al.* han mostrado tasas de error más bajas en comparación con otros detectores de anomalías al evaluar un modelo de Markov oculto parcialmente observable [77]. La distribución log-normal también resultó útil como modelo subyacente para un ataque de suplantación de identidad con información parcial en otro artículo de los mismos autores [28]. Barabási [80] ofrece una explicación plausible de la semejanza entre la distribución log-normal y la forma empírica de las distribuciones resultantes de diversos comportamientos humanos, sin restringirse a las tareas de escritura

en un teclado. En términos generales, esta puede ser deducida al suponer que los sujetos humanos ejecutan tareas utilizando un proceso de encolado basado en decisiones priorizadas.

Chukharev-Hudilainen [74] ha utilizado el parámetro de escala de la distribución exgaussiana, que se ajusta a los tiempos de latencia entre teclas, para detectar pausas y vacilaciones lingüísticas, y arrojar luz sobre los procesos psicolingüísticos subyacentes a la tarea de escritura. Como señala, existe en el campo de la psicología una literatura rica y bien establecida sobre el uso de la distribución exgaussiana para ajustar los tiempos de respuesta, que son muy similares a las latencias entre teclas tanto en sus distribuciones empíricas como en su modelo teórico de ocurrencia. Como ejemplo de esta última afirmación, puede consultarse [81] o [82], donde los valores de los parámetros estimados a partir de los tiempos de respuesta medidos se utilizaron´ para inferir conflictos entre tareas.

No sobran en la literatura sobra cadencias de tecleo las comparaciones sistemáticas entre distintas distribuciones. Como se ha visto, la mayoría de las veces se ignora el problema y se emplea una única distribución. Dos contraejemplos son [76] y [75], donde la distribución log-normal se compara con las leyes de potencia de Benford y Zipf, junto con la distribución exponencial. Intentando superar las limitaciones en los conjuntos de datos existentes, Migdal y Rosenberg [78, 79] han llevado a cabo una comparación detallada de casi veinte distribuciones candidatas para la generación de conjuntos de datos sintéticos utilizando modelos estadísticos; la distribución de Gumbel proporcionó el mejor ajuste general. Estos estudios, que utilizan los conjuntos de datos GREYC [83, 84], se restringen a textos cortos como nombres de usuario y contraseñas que el usuario ha escrito repetidamente, sin evaluar la generalizabilidad a textos libres.

Las técnicas utilizadas para determinar la calidad del ajuste de las distribuciones propuestas han sido X^2 en [78, 79], bondad de ajuste con un método de Monte Carlo en [76], y máxima verosimilitud en [75].

El cuadro 2.3 resume los estudios sobre cadencias de tecleo que han sido reseñados en esta sección por haber realizado consideraciones explícitas de las distribuciones subyacentes para los tiempos de retención y latencia. Las columnas N, L, y E refieren respectivamente a la distribución normal, log-normal, y exgaussiana. Se observa el gradual abandono, durante la última década, de la presuposición de normalidad y su intento de reemplazarla por otras distribuciones. Solo existen tres estudios comparativos en los últimos cinco años, uno solo de los cuales (Monaco [76]) considera texto libre pero que se limita a evaluar tres distribuciones.

Se deduce de aquí la necesidad de formular un estudio comparativo focalizado en texto libre, con mayor cantidad de distribuciones candidatas. No existe un consenso sobre la mejor técnica estadística para la evaluación de los méritos relativos de distintas distribuciones, por lo que deberían utilizarse múltiples criterios a los fines de determinar cuál de ellas provee mejor ajuste.

2.4.6. Falsificaciones sintéticas y ataques de presentación

El primer intento de construir una implementación práctica de un esquema de detección de vida para sistemas de autenticación basados en cadencias de tecleo ha sido el de Stefan y Yao [27]. Ellos han propuesto dos sencillas estrategias, denominadas GaussianBot y NoiseBot, y construido sendos programas para inyectar eventos de teclado generados estadísticamente

en una maquina remota, con el objetivo de eludir un sistema de verificación de cadencias de tecleo. Las estrategias emplean respectivamente distribuciones gaussianas y uniformes, infiriendo los parámetros en base a estadísticas de la población en general, y agregan un modelo de Markov de primer orden para forjar imitaciones sintéticas de una muestra temporal. Los autores pusieron a disposición del público una implementación práctica, que ejecuta bajo el sistema X Windows, de un protocolo para distinguir al usuario humano legítimo de los adversarios antedichos. Para la tarea utilizaron un clasificador SVM y análisis de componentes principales, reportando tasas de falsos positivos entre 2% y 3.5% para GaussianBot y alrededor de 1 % para NoiseBot, con aproximadamente 5% de rechazos para el usuario legítimo. Se informaron resultados similares en la versión extendida [29]. En [26, 71] también se emplea una estrategia idéntica a GaussianBot, pero agregando un filtro para las latencias de digrafos que exceden los 300ms.

Monaco *et al.* [28] han propuesto un modelo generativo más avanzado que puede aprovechar los tiempos de latencia entre teclas filtrados por un ataque de canal lateral, incluso si no se cuenta con los nombres de tecla correspondientes. Las distribuciones log-normales reemplazan aquí a las gaussianas para lograr un mejor ajuste, y se usa un modelo de Markov oculto de dos estados para estimar si el usuario se encuentra escribiendo activamente o si está decidiendo que escribir a continuación. En el primer caso, los tiempos de latencia entre teclas son aproximados midiendo la distancia entre las mismas en un teclado físico, e ingresando en un modelo comportamental un conjunto de parámetros estimados a partir de los datos de la población general y los tiempos observados del usuario que se desea imitar. Con solo 50 observaciones de latencias y sin los nombres de las teclas correspondientes, el EER de un clasificador [85] utilizado como base en la comparación sube hasta 60%. Posteriormente emplearemos este modelo, junto con GaussianBot, NoiseBot y las estrategias que propondremos, para un estudio comparativo. Siguiendo la terminología de los autores originales lo llamaremos *LBMC*, acrónimo del inglés *linguistic buffer and motor control* que se traduce como *almacenamiento lingüístico y control motor*. Cabe mencionar una limitación que los autores de LBMC reconocen, y es que este método permite sintetizar únicamente los tiempos de latencia y no los de retención.

En un estudio grato y curioso, que puede fácilmente pasarse por alto en la literatura del tópico, Ness [86] propone un robot mecánico que imita patrones de escritura en el transcurso de una exploración de alternativas de *hardware* para ataques de presentación contra sistemas de verificación de cadencias de tecleo. A pesar de la originalidad del dispositivo, su manera de enfocar la generación de muestras sintéticas es casi idéntica a la de GaussianBot. El autor nota un fenómeno parecido al que se describirá en la sección 5.3.2.4, donde algunas falsificaciones sintéticas son, en cierto sentido que se precisara más adelante, demasiado impolutas como para pertenecer realmente al usuario legítimo. Sin embargo, no se propone ninguna estrategia para aprovechar esta anomalía.

La utilidad de las muestras sintéticas no se restringe a su empleo en ataques de presentación. El tema ha sido también abordado con el objetivo de generar grandes conjuntos de datos sintéticos para suplir la ausencia de información en ciertos casos. Migdal y Rosenberger [78, 79] han evaluado una docena de distribuciones candidatas para una tarea de usurpación (en términos de los autores, es decir, pasar una falsificación por el usuario legítimo) y para la estimación apriorística del EER asintótico de un usuario al verificar texto

fijo. Aparentemente, en estos casos la distribución de Gumbel es la más adecuada. Veremos más adelante, al discutir los resultados del experimento sobre distribuciones subyacentes en la sección 6.1, que no es el caso al tratarse de texto libre.

Un ataque de repetición, del inglés *replay attack*, es una forma de ataque de presentación que busca explotar a un sistema de autenticación de cadencias de tecleo aprovechando filtraciones de la información biométrica del usuario legítimo. Consiste en inyectar en el sistema objetivo los mismos eventos temporales observados anteriormente, sin modificarlos. Como la fuente de la muestra original de tiempos es el propio usuario legítimo, una contramedida de defensa que detecte falsificaciones sintéticas no puede proteger contra ataques de repetición y se requerir a otro tipo de estrategia contra estos que contra aquellas. Hazan *et al.* [87] proponen un protocolo en el que solo se intercambia un subconjunto de los tiempos de retención y latencia al transmitir la cadencia de tecleo entre un cliente y un servidor, rellenado los huecos con datos falsos. De esta forma, cualquier filtración de estos datos resulta inútil a un atacante, que no podrá distinguir el relleno de la cadencia legítima.

Otra estrategia de defensa, que merece la mención pero que queda fuera del alcance de esta tesis, consiste en potenciar la información provista por los tiempos de retención y latencia con un esquema multimodal de sensores adicionales. Particular

Año	Autores	Síntesis	Defensa	Texto libre
2010	Stefan & Yao [27]	normal/uniforme	SVM	✗
2012	Stefan *et al.* [29]	normal/uniforme	SVM	✗
2013	Rahman *et al.* [71]	normal/uniforme		✗
2015	Monaco *et al.* [28]	log-normal/HMM/LBMC		✓
2016	Stanciu *et al.* [89]		Otros sensores	✓
2017	Ness [86]	normal		✓
2018	Migdal y R. [78]	Modelo generativo		✗
2019	Migdal y R. [79]	Modelo generativo		✗
2019	Hazan *et al.* [87]		Sólo replay	✓

Cuadro 2.4: Resumen de los métodos de síntesis de muestras temporales y de contramedidas de detección

mente en dispositivos móviles, se cuenta con acelerómetros, giróscopos, y sensores de presión, que proporcionan mayor cantidad de información biométrica del comportamiento [88]. Lo que es más, esta resulta ser más difícil de observar y de falsificar para un atacante que las muestras temporales [89].

En el cuadro 2.4 se muestra un resumen de los estudios reseñados en esta sección, cuyo objetivo haya sido la generación de muestras sintéticas para ataques de presentación u otros fines y la detección de las mismas como contramedida de defensa. Se observa que, con excepción de [28], los esquemas de síntesis son muy sencillos y es esperable que admitan

mejoras significativas. Lo que es más, no se han propuesto esquemas de detección de vida o filtrado de falsificaciones sintéticas en texto libre que utilicen exclusivamente parámetros temporales sin aumentar la potencia de los mismos con información comportamental extraída de otros sensores.

2.4.7. Ataques por canal lateral e identificación del texto ingresado

El objetivo último de un ataque de canal lateral para extracción de textos (*keylogging*) es reconstruir completamente el texto original que el usuario ha tipeado, pero la factibilidad de este objetivo depende tanto de la cantidad de información filtrada por el sistema interferido como de aquella disponible al atacante en forma previa. A modo de ejemplo, [90] reporta una precisión superior al 65% al intentar identificar un PIN de cuatro dígitos entre los tres candidatos principales, combinando emanaciones acústicas y movimientos de mano que fueran capturados con dispositivos portátiles. Al restringir la información disponible para el clasificador a emanaciones acústicas exclusivamente, la tasa de éxito se reduce al 40% para cada carácter individual y se degrada aun más al considerar el texto conjunto [91].

Queda fuera del alcance de esta tesis tratar ataques de canal lateral basados en múltiples fuentes de información biométrica, y por eso nos concentraremos en aquellos que se restringen en forma exclusiva a información de tiempos, de retención y latencia. Para el tratamiento de estos ataques casi se ha abandonado la ambición de reconstruir completamente un texto arbitrario en favor de dos enfoques más humildes: ya sea detectar si el texto escrito se encuentra en una lista (bastante corta) de candidatos predeterminados [92], o a lo sumo generar posibles candidatos en orden de probabilidad decreciente [93] para reducir la complejidad de un ataque de fuerza bruta posterior. Trataremos estos en mayor detalle a continuación.

El primer ataque con utilidad práctica que logró aprovechar la información de tiempos de latencia filtrados a través de un canal lateral para reducir la complejidad de las contraseñas de fuerza bruta fue propuesto por Song *et al.* [93]. Los autores descubrieron que SSH y otros protocolos de cifrado de tráfico interactivo envían cada tecla individual inmediatamente, al momento de ser presionada, a la maquina remota en un paquete IP de tamaño reducido. Esto permite que un intruso averigüe la longitud de la contraseña y recopile, sin mucho esfuerzo, los tiempos de latencia entre teclas. Utilizando estos últimos, la identidad del usuario puede ser revelada empleando las técnicas de autenticación que se han reseñado en la sección 2.4.2. Es más grave que, como se verá, la entropía de una contraseña, incluso cuando ha sido elegida al azar, se pueda reducir en aproximadamente un bit por carácter.

El método de Song *et al.* permite reducir el espacio de búsqueda al adivinar contraseñas utilizando ajuste gaussiano para las distribuciones de tiempo, un modelo oculto de Markov para estimar la secuencia de teclas, y el algoritmo de decodificación de Viterbi para inferir los candidatos más probables en orden de probabilidad decreciente. En promedio, los autores lograron reducir casi 50 veces la complejidad al intentar romper por fuerza bruta varias contraseñas armadas con un conjunto de caracteres reducido, siempre y cuando se contara con suficientes datos de entrenamiento intrausuario. Este valor empeoro al utilizar datos de entrenamiento interusuario. Al actualizar el método agregando decodificación *m*-*n*-Viterbi, modelos de Markov de orden superior, y una estimación apriorística de la frecuencia de pares

de caracteres en el idioma inglés, Zhang *et al.* [94] lograron mejorar la precisión del método en un orden de magnitud.

Hoy en día se prefiere el aprendizaje automático y los clasificadores de última generación, y la tarea de fuerza bruta se complementa con la detección de posibles coincidencias dentro una lista de textos candidatos predeterminados. Por ejemplo, Lipp *et al.* [95] han logrado una tasa de éxito entre 67% y 96% al identificar una URL escrita en la barra del navegador, siempre que la misma se encontrara en una lista corta de diez sitios web muy visitados. El enfoque consistió en utilizar un ataque por canal lateral que capitalizaba las demoras en las interrupciones de Javascript para extraer, desde un script que se ejecutaba en otra pestaña, las latencias entre teclas; luego, un clasificador k-NN ajustado para la tarea identifica la URL dentro de la lista de candidatos. Bajo el ingenioso nombre SILK-TV, Balagani *et al.* [96] reconstruyen claves y PINs en base a videos o capturas del cuadro de texto enmascarado donde se ingresan, utilizando redes neuronales y bosques aleatorios. Siempre que se cuente con varias observaciones para la misma clave, esta estrategia reduce la cantidad de intentos en un ataque por fuerza bruta entre un 25% y un 38,5% para claves, pero no resulta demasiado exitosa para secuencias exclusivamente numéricas.

El problema de reconstruir un texto en base a su vector de tiempos de tecla es difícil y se ha demostrado que su factibilidad depende fuertemente del usuario objetivo [97]. Es interesante el enfoque de Liu *et al.* [98], quienes proponen un modelo comportamental interusuario para la reconstrucción de PINs cortos. Con hasta seis dígitos y para ciertas combinaciones vulnerables, el método alcanza un 10% de éxito en menos de 10 intentos. Lamentablemente, este abordaje de la cuestión no es generalizable

Año	Autores	Método	Tamaño del texto	Tamaño de lista
2001	Song *et al.* [93]	HMM/Viterbi	7/8	N/A
2009	Zhang *et al.* [94]	HMM/m-n-Viterbi/frecuencias	8	N/A
2017	Lipp [95]	k-NN	6-12	10
2018	Balagani *et al.* [96]	ANN/bosques aleatorios	8	N/A
2019	Liu *et al.* [98]	Modelo comportamental	6	10
2019	Monaco *et al.* [92]	RNN/frecuencias	1-20 (palabras)	N/A

Cuadro 2.5: Resumen de los métodos de reconstrucción/identificación de textos utilizando parámetros temporales exclusivamente

a textos libres pues el modelo comportamental depende de la geometría de un teclado numérico.

Monaco [92] abordo el problema más complejo de reconstruir la consulta hecha a un motor de búsqueda. En primer lugar, las palabras individuales fueron delineadas utilizando los tamaños de paquete, para luego alimentar una red neuronal recurrente de tres capas que clasifica cada letra de acuerdo a su probabilidad. Un análisis posterior permitió la

reconstrucción de las palabras en orden de probabilidad y, combinando estas con un modelo lingüístico de frecuencias relativas de palabras, la consulta original. El esquema de clasificación alcanzó una precisión en torno al 15% en varios navegadores y motores de búsqueda, bajo el criterio de que la consulta original debía encontrarse entre las 50 determinadas como más probables. Esta cifra no debe interpretarse como baja considerando la enorme dificultad de la tarea.

Una revisión reciente y exhaustiva [15] sobre *keylogging* muestra que, a pesar de que los estudios dedicados a extracción de tiempos de escritura por medio de un canal lateral no son escasos, sí lo son aquellos que proponen técnicas para reconstruir o identificar el texto original cuya secuencia de teclas se desconoce. Casi todos los artículos allí reseñados proponen originales e ingeniosas estrategias, que muchas veces ni siquiera requieren acceso local, para detectar que el usuario se encuentra escribiendo y para filtrar, total o parcialmente, los tiempos entre eventos de teclas. Sin embargo, casi siempre refieren a los pocos artículos que se han discutido aquí para el proceso ulterior de recuperación o identificación del texto original. La tendencia común para ambas categorías de métodos consiste en que el texto original es de tamaño reducido, al igual que la lista de candidatos.

Un resumen de los estudios reseñados en esta sección puede verse en el cuadro 2.5. Se deduce de esta reseña que el problema de reconstrucción/identificación de textos utilizando parámetros temporales exclusivamente es un campo escasamente explorado y que se ha restringido, con excepción de [92], a atacar claves, PINs, y textos muy cortos, utilizando listas de candidatos de tamaño muy restringido. Un método que permita identificar textos de mayor longitud o dentro de listas de candidatos mayores, como el que se propondrá en la sección 4.5, constituye necesariamente una novedad.

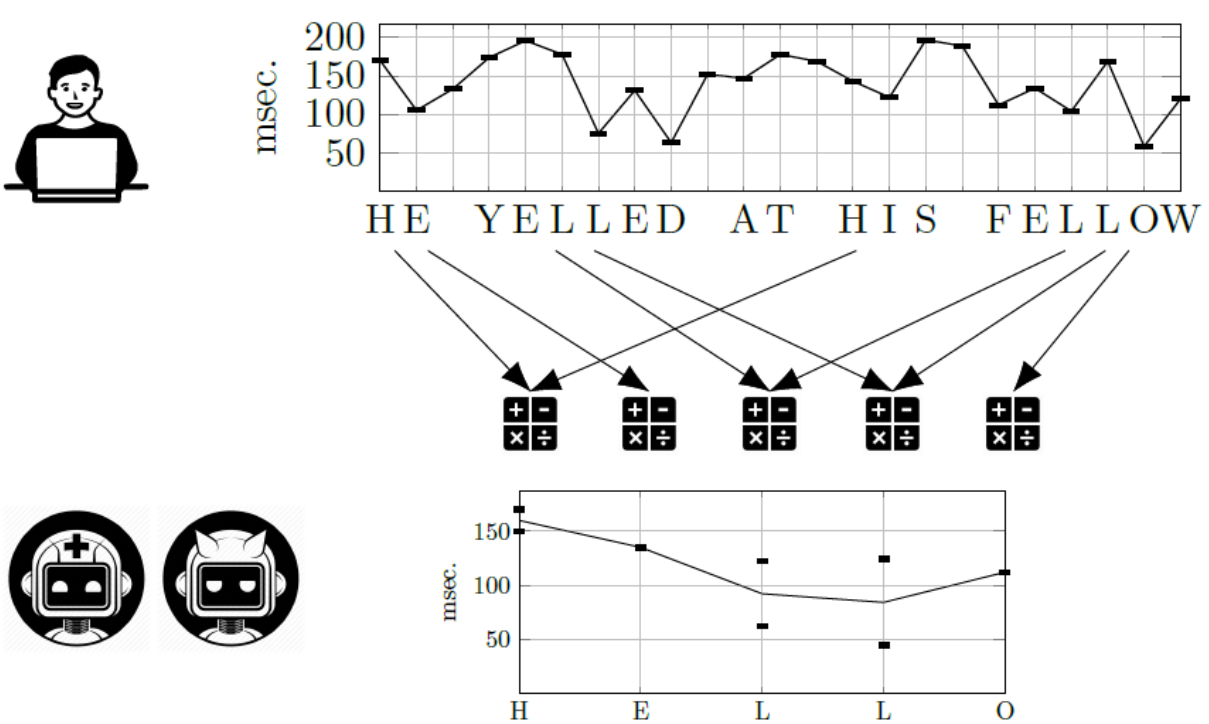

Figura 2.5: Modelado por contextos finitos de la palabra HELLO

2.5. Modelado por contextos finitos

El modelado por contextos finitos es una técnica que ha sido propuesta en [33] por el autor de esta tesis, inspirada en los métodos de predicción por coincidencias parciales para la compresión de datos [99, 100, 101, 102]. En resumen, para cada tecla k_i, el modelado por contextos finitos recopila en un conjunto S_i todas las observaciones pasadas de los atributos de tiempo en el perfil de usuario que están precedidas por el contexto de mejor coincidencia $k_{i-m} \ldots k_{i-1}$ de orden m. A modo de ejemplo, la figura 2.5 muestra a robot bueno y robot malo (que serán presentados con la pompa y circunstancia que merecen en 4.3), utilizando una muestra del texto *HE YELLED AT HIS FELLOW* para recrear los tiempos de la palabra inglesa *HELLO*.

El proceso para las letras E y la O es inmediato, pues existe una única observación de ellas en la muestra de entrenamiento; los conjuntos S_E y S_O tienen un solo elemento. En cambio, la H y ambas L cuentan con más de una observación pasada. En la figura se muestra a robot bueno (o a robot malo) promediando las observaciones de S_H, S_{L1}, y S_{L2} para sintetizar una muestra artificial. Veremos tanto en esta sección como en 4.2 que podemos hacer bastante más con los conjuntos S_i que solamente promediar sus elementos, y que de esta forma se logra mejorar la verificación de la identidad del usuario, incrementar la tasa de éxito de los ataques a sistemas de autenticación basados en cadencias de tecleo, posibilitar el diseño de contramedidas eficaces, e incluso descubrir que texto escribió el usuario aunque contemos únicamente con los tiempos y no con los nombres de tecla correspondientes.

Debido al alcance de esta tesis, aquí nos concentraremos es las últimas tres aplicaciones del modelado por contextos finitos. Un tratamiento detallado de la verificación de identidad en esquemas de autenticación mixtos puede encontrarse en [33], mientras que el tratamiento de sus dificultades computacionales, que aquí serán pasadas por alto, puede encontrarse en [103].

2.5.1. Notación y definiciones

Para un usuario dado, su *perfil* P es un conjunto de *muestras* M_i. Cada muestra de largo n se compone de un *vector de teclas* $k_{i,1} \ldots k_{i,n}$ y de un *vector de tiempos* $t_{i,1} \ldots t_{i,n}$. Nada impide tratar más de un vector de tiempos, como pueden ser retención y latencia, o vectores de otro tipo de atributos, como presión o aceleración; se considerarán, sin embargo, como muestras independientes.

Un *texto objetivo* de largo n consiste en una secuencia de teclas $k_1 \ldots k_n$ cuyos tiempos quieren reconstruirse en base al perfil del usuario.

2.5.2. Agrupamiento por contexto

Dado un texto objetivo $k_1 \ldots k_n$, el proceso de *agrupamiento por contexto* consiste en, para cada tecla k_j y para cada orden m desde cero hasta una cierta cota, recorrer el perfil P del usuario extrayendo de cada muestra M_i las observaciones de tiempo $t_{i,x}$ tales que

$$k_{i,x-m} \dots k_{i,x} = k_{j-m} \dots k_j \tag{2.5}$$

para formar los conjuntos S_i^m, que denominamos *conjuntos de observaciones contextualizadas*. Formalmente

$$S_{mj} = \{t_{i,x} \in P | k_{i,x-m} \dots k_{i,x} = k_{j-m} \dots k_j\} \tag{2.6}$$

El pseudocódigo del proceso de agrupamiento por contexto para una cierta tecla k_j del texto objetivo y cierto orden m puede verse en el listado 2.1.

El máximo orden m a computar se elige balanceando la potencia computacional disponible con la precisión esperada en la reconstrucción. Como se verá en la sección 6.2.4, a mayor orden de contexto mayor precisión en la reconstrucción, pero también mayor costo computacional. En general este último es linealmente proporcional al orden máximo.

Recorrer todas las muestras del perfil del usuario, como se refleja en el pseudocódigo de más arriba, es un enfoque trivial y costoso. Existen algoritmos y estructuras de datos, como los trie o árboles de prefijos, que permiten intercambiar tiempo de ejecución por uso de memoria al realizar el agrupamiento por contexto. Estas consideraciones, de importancia para una implementación práctica, escapan al alcance del uso que se le dará aquí al método. El lector interesado puede consultar la referencia [103].

Listado 2.1: Seudocódigo de agrupamiento por contexto

```
S_j^m = ∅
PARA CADA M_i EN P
    n <- |M_i|

    PARA CADA x EN (m+1) ... n
        cumple <- SI

        PARA CADA y EN 0 ... m
            SI k_{i,x-y} != k_{j-y}
                cumple <- NO

        SI cumple
            S_j^m <- S_j^m ∪ {t_i}

DEVOLVER S_j^m
```

2.5.3. Filtrado de los conjuntos de observaciones contextualizadas

El proceso de agrupamiento por contexto da como resultado un grupo de conjuntos de observaciones contextualizadas para la tecla k_j del texto objetivo. No todas las observaciones

en un $S^i_j \in G_j$ tienen necesariamente valores correctos o aceptables. Para evitar el efecto de pausas prolongadas o errores en la captura de datos, se aplicará un filtro f_S a cada S^i_j, para obtener

$$G_j = \{f(S_j^0), f(S_j^1), \dots f(S_j^M)\} \tag{2.7}$$

en donde

$$f_S(S_j^i) \subseteq S_j^i \tag{2.8}$$

Un ejemplo trivial de filtro consiste en remover todos aquellos valores que exceden los 1500 mseg. para descartar tiempos debidos a pausas y no al flujo normal de escritura [104, 105]. Por medio de estrategias de filtrado más sofisticadas, utilizando medias móviles sobre las muestras en el perfil del usuario, se obtienen mejores resultados; un detalle de estas puede leerse en [103]. En ocasiones es útil eliminar también los valores atípicos, es decir aquellos cuya diferencia con la media muestral excede una cierta cantidad de desvíos estándar.

No todos los S^m_j son de utilidad en etapas posteriores del método de modelado o en algoritmos derivados. Algunos de ellos pueden ser descartados. Por ejemplo, si el objetivo es utilizar los S_j para estimar los parámetros de un modelo gaussiano, debemos contar con al menos 20 observaciones [106] para lograr un error aceptable. Aplicamos entonces un nuevo filtro f_G, pero ahora sobre G_j, para eliminar los S^m_j que no cumplen los requisitos, obteniendo

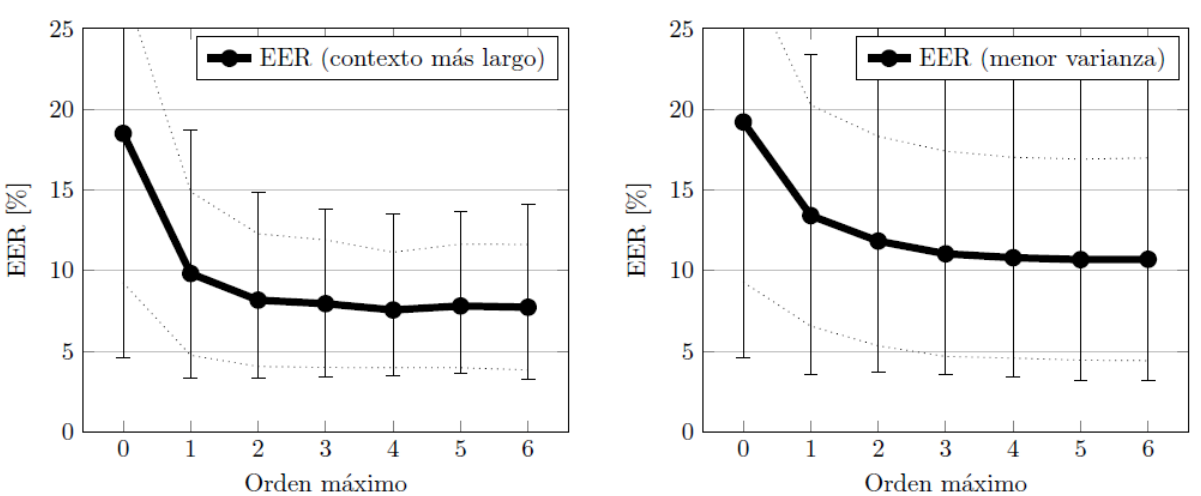

Figura 2.6: Comparación de estrategias de selección del contexto de mejor coincidencia

$$G'_j = f_G(G_j) \subseteq G_j \tag{2.9}$$

Otros criterios de filtrado posibles incluyen no exceder un cierto umbral para el desvío estándar, que las observaciones no sean demasiado antiguas, y muchas otras. Un tratamiento exhaustivo puede encontrarse en [103].

2.5.4. Selección del contexto de mejor coincidencia.

Aunque algunos métodos pueden utilizar simultáneamente contextos de distintos órdenes [107, 108], en la mayoría de las ocasiones es preferible seleccionar entre los miembros de G_j un único S^i_j para modelar los tiempos de la tecla k_j. En este caso, denominamos al S^i_j elegido simplemente como S_j, descartando el superíndice que indica el orden del contexto, y lo llamamos *contexto de mejor coincidencia*.

En general, a mayor orden de contexto se verifica una mejor coincidencia con los tiempos a modelar, como se discutirá en la sección 6.2.4. De esta observación se deduce el criterio de selección más sencillo: tomar el $S^i_j \in G_j$ con mayor i, es decir con mayor orden de contexto. No siempre obtendremos así los mejores resultados. Aunque exista mejor coincidencia con los tiempos a modelar en contextos de mayor orden, también se verifica en estos que la cantidad de observaciones disponibles es menor. Es esperable que, por ejemplo, el tiempo de latencia de la E en la palabra FALANGE se vea mejor representado por todos aquellos de la tecla E antecedidos por ANG que por todas las observaciones de la tecla E precedida por G; es esperable además que contemos con muchas más de las últimas que de las primeras en el perfil del usuario.

Otras estrategias de selección posible incluyen quedarse con el contexto cuyas observaciones presentan menor varianza, o pesar el largo del contexto contra la cantidad de observaciones. Una discusión de todas ellas y los resultados cuantitativos puede consultarse en [103]. La figura 2.6, gentileza de [103], compara los EERs de un sistema de autenticación empleando ambas estrategias para distintos órdenes máximos de contexto; allí seleccionar el contexto más largo provee mejor rendimiento que seleccionar el contexto de menor varianza.

La selección del contexto de mejor coincidencia para cada *kj* de un texto objetivo de largo *n* tiene como resultado una secuencia de conjuntos de observaciones contextualizadas

$$S = \{S_1, S_2, \dots, S_n\} \tag{2.10}$$

Puede ocurrir que algunos de ellos sean conjuntos vacíos, si no hay muestras suficientes que sobrevivan al filtro f_S, o si ningún conjunto de observaciones contextualizadas para la tecla k_j sobrevive al filtro f_G.

2.5.5. ¿Y después?

¿Qué hacer ahora con la secuencia *S* de contextos de mejor coincidencia? Lo que sigue ya no es parte del modelado por contexto finitos, sino que depende de la aplicación que se quiera hacer de este método.

Si nuestro objetivo es la verificación de la identidad, los S_i se utilizaran para estimar cuan cerca se encuentra la muestra del presunto usuario [33, 103]. Para sintetizar una cadencia de

tecleo artificial que será utilizada en un ataque de presentación, los S_i servirán para estimar los parámetros de las distribuciones correspondientes (ver sección 4.2). Cuando queramos defender un sistema de autenticación frente a tales ataques, los S_i proveerán una descripción empírica del comportamiento del usuario que será contrastada con la muestra a verificar (ver sección 4.3). Pero eso es otra historia que será contada más adelante.

2.5.6. Un ejemplo integrador

Retornemos a la figura 2.5 y supongamos que el perfil P del usuario legítimo cuenta con una única muestra M_1 de tiempos de retención, que corresponde al texto que se muestra.

$$P = \{M_1\}$$

El largo de la muestra M_1 es de 23 teclas y la secuencia es

$$k_{1,1} = H$$
$$k_{1,2} = E$$
$$k_{1,3} = espacio$$
$$k_{1,4} = Y$$
$$\ldots$$
$$k_{1,23} = W$$

Queremos reconstruir el texto objetivo HELLO, de largo cinco. La secuencia de teclas correspondiente es

$$k_1 = H$$
$$k_2 = E$$
$$k_3 = L$$
$$k_4 = L$$
$$k_5 = O$$

Veamos ahora el proceso de agrupamiento de contextos. Para la tecla $k_1 = H$, sin contexto, tenemos dos observaciones: una al inicio de la muestra M_1 (la H de HE) y otra al inicia de la palabra HIS (en la posición 14). Por lo tanto,

$$S_1^0 = \{t_{1,1};\ t_{1,14}\}$$

No hay en M_1 contextos de mayor orden que cero que contengan la H, porque es esta la tecla inicial de la muestra que queremos sintetizar. Entonces, S_1 contendrá sólo al conjunto anterior.

$$S_1 = \{S_1^0\}$$

La tecla $k_2 = E$ cuenta con cuatro observaciones en M_1: una dentro de HE, dos dentro de YELLED, y una dentro de FELLOW. Sin considerar el contexto, es decir en orden cero, tenemos que

$$S_2^0 = \{t_{1,2};\ t_{1,5};\ t_{1,8};\ t_{1,19}\}$$

El contexto de orden uno para $k_2 = E$ es $k_1k_2 = HE$. La muestra M_1 cuenta con una única observación de este, en la palabra HE al inicio de la frase. Entonces

$$S_2^1 = \{t_{1,2}\}$$

y los contextos de orden superior están vacíos, porque la secuencia HEL no aparece en la muestra M_1. Finalmente

$$S_2 = \{S_2^0, S_2^1\}$$

Para la tecla $k_3 = L$, en orden cero, existen cuatro observaciones; dos en YELLED y dos en FELLOW.

$$S_3^0 = \{t_{1,6};\ t_{1,7};\ t_{1,20};\ t_{1,21}\}$$

El contexto de orden uno para $k_3 = L$ es $k_2k_3 = EL$. Hay una observación en YELLED y otra en FELLOW.

Nombre	Secciones	Obs./Sat.	Actual
Desafío	2.3, 2.4.2, 2.4.3		**Autenticación** Educción de emociones Características fisiológicas
Tarea de escritura	2.3, 2.4.2	Claves Texto fijo	**Texto libre**
Técnicas	2.4.2.1, 2.4.2.2	Simples Distancias Técnicas *ad hoc*	**Aprendizaje automático**
Conjunto de datos	2.4.1	Privados	**Abiertos públicos**
Modelo de ataque	2.3, 2.4.2, 2.4.6	Esfuerzo cero	**Impostores entrenados** **Ataques de presentación** Malware

Cuadro 2.6: Tendencias en la disciplina de análisis de cadencias de tecleo

$$S_3^1 = \{t_{1,6};\ t_{1,20}\}$$

y los contextos de orden superior están vacíos pues HEL no aparece en M_1. Obtenemos entonces que

$$S_3 = \{S_3^0, S_3^1\}$$

El proceso es idéntico para k_4 = *L* y k_5 = *O*, con contextos de orden hasta dos y cero respectivamente. Como se observa en la figura, todos los tiempos de retención se encuentran en un intervalo razonable, por lo que un filtro f_S que elimine muestras con valor superior a 1500 mseg. dejará todos los S_i^j intactos. En este ejemplo reducido, consideremos también que f_G es la función identidad. Entonces

$$G_j = S_j$$

Para la selección de los contextos de mejor coincidencia, utilicemos el contexto más largo. Así tendremos finalmente que

$$S = \{S_1^0, S_2^1, S_3^1, S_4^2, S_5^0\}$$

Así culmina el proceso de modelado por contextos finitos para el texto objetivo HELLO en base al perfil P. Ahora, dándole una aplicación práctica, estos conjuntos de muestras del contexto de mejor coincidencia pueden utilizarse para sintetizar una cadencia de tecleo. La sencilla estrategia Average de la sección 4.3 promediara los valores de cada S_i, para devolver los tiempos de retención

$$t_1 t_2 t_3 t_4 t_5 = \mu\,(S_{10})\ \mu\,(S_{21})\ \mu\,(S_{31})\ \mu\,(S_{42})\ \mu\,(S_{50})$$

que deberían parecerse a los del usuario legítimo.

2.6. Síntesis del estado del arte

Como se ha visto en las secciones anteriores, la literatura relevante del tópico es extensa y las técnicas exploradas son diversas. Resulta, por tanto, imposible sintetizarla en un cuadro sinóptico que considere los estudios individuales, cuya extensión lo volvería incomprensible. Debemos entonces en primer lugar identificar tendencias generales que nos permitan estrechar la perspectiva y, hallando las áreas que demandan atención ulterior, definir el alcance de la investigación. Consideraremos que cada sección anterior de este capítulo revela una tendencia en la disciplina y/o expone una oportunidad de investigación en un tópico poco explorado. El cuadro 2.6 expone las primeras, mostrando cuales tópicos se encuentran saturados o se han vuelto obsoletos y cuales son objeto de investigación actual, junto con las referencias a las secciones donde son tratadas. El cuadro 2.7 resume las oportunidades de investigación, extrayendo un conjunto de tópicos en los que existen enfoques vacantes que

no han sido explorados aun y que resultan promisorios; allí se indica la referencia a las secciones en las que son tratadas, y los enfoques que ya han sido estudiados sobremanera, aquellos que se han utilizado recientemente, y los que permanecen inexplorados según surge de la reseña de este capítulo. A lo largo de esta sección, los conceptos resaltados en negrita corresponden a los que se utilizaran en el capítulo 3 para la definición del problema.

Surge de la sección 2.4.2 que, al restringirnos al proceso de **autenticación**, el **texto libre** ha recibido mucha menos atención que el texto fijo y las claves; entre otros motivos, porque es un problema de mayor complejidad que demanda cantidades mucho más grande de datos para el entrenamiento de modelos y para su evaluación. Sin embargo, descubrimos en la reseña de la sección 2.4.1 que en los últimos años se han puesto a disposición del público en forma abierta múltiples **conjuntos de datos** aptos para la tarea, los que nos habilita a encarar un estudio del tema sin demandar la recolección previa de los datos necesarios. También, como se ha mostrado en 2.4.2.2, la tendencia actual en el área es utilizar métodos de **aprendizaje automático** para la tarea, o al menos incluirlos como parte del esquema de autenticación.

Hemos visto al recorrer la historia de la disciplina en 2.3 que actualmente es necesario considerar un **modelo de ataque con impostores entrenados**, en contraste con la ingenua evaluación de esfuerzo cero del pasado, y la posibilidad de que un sistema de autenticación basado en cadencias de tecleo puede ser vulnerado por un **ataque de presentación** suficientemente sofisticado. Sin embargo, como nos ha hecho notar la sección 2.4.6, con una única excepción especializada para otro fin las estrategias de **síntesis de cadencias de tecleo artificiales** que se han propuesto en la literatura son todavía muy básicas. En particular, se suele utilizar modelos gaussianos pero las verdaderas **distribuciones subyacentes** no han sido **comparadas exhaustivamente** para texto libre, como demuestra la reseña de la sección 2.4.5, ni se han utilizado las **distribuciones empíricas** del perfil intrausuario.

El uso de las cadencias de tecleo para potenciar **ataques de canal lateral** ha sido escasamente estudiado; en la sección 2.4.7 observamos ya que la mayoría de los estudios refieren a dos estudios destacados que utilizan el algoritmo de Viterbi. El método de **modelado por contextos finitos** promete un tratamiento unificado de los problemas de autenticación, síntesis de muestras artificiales para ataques de presentación,

Nombre	Secciones	Saturado	Reciente	Vacancia
Fundamentos de la disciplina				
Distribuciones	2.4.5	Texto fijo	**Texto libre**	
- Texto libre			Benford/Zipf Exponencial Log-normal	**Comp. exhaustiva**
Ataque contra el sistema de autenticación				
Modelo de ataque	2.3, 2.4.6	Esfuerzo cero	**Imp. entr.** Malware	
Técnica de síntesis	2.4.6	Normal Uniforme	Log-normal HMM/LBMC Generativo	**Distr. empíricas**
Canal lateral	2.4.7		HMM Viterbi k-NN/RNN	**Contextos finitos**
- Texto			≈ 10 car.	> 10
- Lista			≈ 10 items	> 10
Contramedidas de defensa				
Detección	2.4.6		SVM Multisensor Replay	**Contextos finitos**

Cuadro 2.7: Oportunidades de investigación

e identificación del texto ingresado luego de un ataque de canal lateral.

Capítulo 3

Definición del problema

> Un problema que no tiene solución no es un problema
>
> Apócrifo

El problema que guiará los experimentos de esta tesis es la creación de contramedidas de defensa eficaces ante ataques de presentación que utilicen muestras sintetizadas artificialmente, en base a un perfil intrausuario total o parcial. En el camino hacia la solución se estudiarán las distribuciones temporales subyacentes y se propondrán estrategias de síntesis de muestras capaces de engañar a los sistemas de detección actuales con mayor frecuencia que los empleados usualmente para la evaluación. Posteriormente, descubriremos que una modificación de los mismos métodos permite potenciar los ataques por canal lateral para identificación del texto ingresado.

El resto del capítulo está organizado como se describe a continuación. La sección 3.1 define el problema que guiara los experimentos de esta tesis. La sección 3.2 enumera los objetivos concretos que se buscan. La sección 3.3 detalla la hipótesis a demostrar. Finalmente, la sección 3.4 define los límites y el alcance de la propuesta de solución.

3.1. Definición del problema

Todos los sistemas informáticos están bajo ataque permanente y deben diseñarse con esta consideración en mente. La primera línea de defensa fundamental en todo sistema informático es la autenticación de sus usuarios, tanto al inicio de la sesión como en forma continua durante su transcurso. La cadencia de tecleo es una característica biométrica comportamental que puede ser utilizada como segundo factor de autenticación y que tiene la ventaja de ser transparente, en el sentido de que no requiere acciones ulteriores del usuario para llevar a cabo el proceso de verificación de identidad. Los sistemas de autenticación basados en cadencias de tecleo han sido extensamente estudiados desde hace cuarenta años y el tópico ha alcanzado hoy en día elevados niveles de seguridad, eficiencia, y escalabilidad en condiciones realistas.

Sin embargo, en los últimos años se ha cuestionado la metodología usual de evaluación, denominada peyorativamente de esfuerzo cero, y se ha propuesto desafiar este tipo de sistemas bajo un modelo de ataque con impostores entrenados. Lo que es más, no sólo el sistema que el esquema de autenticación protege, sino el esquema de autenticación en sí será permanentemente sometido a intentos malintencionados, y por este motivo es pertinente comprender las modalidades de ataque presentes y anticipar las futuras. En particular, hemos visto que las familias de ataques más relevantes contra los sistemas de autenticación basados en cadencias de tecleo son los ataques de presentación y los ataques por canal lateral. Las técnicas actuales de síntesis de muestras artificiales para su uso en estos tipos de ataques, salvo contadas excepciones, se basan en sencillos modelos gaussianos de bajo orden.

Para mejorar la seguridad de los sistemas de autenticación basados en cadencias de tecleo frente a ataques de presentación y de canal lateral, y subsiguientemente plantear contramedidas de defensa eficaces, se requiere en primer lugar perfeccionar las técnicas de modelado de cadencias artificiales. Pero para este fin se requiere previamente entender en profundidad las distribuciones subyacentes y el modelo de comportamiento que genera la cadencia de tecleo de un usuario legítimo y sus variaciones, un tema que no ha sido objeto de estudio sistemático aún.

3.2. Objetivos

En la sección 2.6, que sintetiza los hallazgos surgidos al reseñar el estado del arte, se detectó el problema pendiente de *identificar las distribuciones subyacentes de los atributos temporales en texto libre*, cuya solución permitiría *proponer estrategias de síntesis de muestras artificiales* que superen a las actuales, y que sirvan de base para un *esquema de detección de vida* y para *explorar técnicas de identificación del texto ingresado* que aprovechen los tiempos filtrados por un ataque de canal lateral. Se necesitará, para cumplir estos objetivos, *crear una herramienta* acorde.

Los objetivos buscados en el presente estudio, que obedecen a las vacancias de investigación resumidas en el cuadro 2.7 y que se han enumerado en el párrafo anterior, se describen a continuación en mayor detalle.

- *Identificar las distribuciones subyacentes* y los patrones de comportamiento que generan la cadencia de tecleo en texto libre de un usuario legítimo por medio de una comparación sistemática de histogramas empíricos de tiempos y diversos métodos de evaluación, como bondad de ajuste y criterio de información de Akaike. Este objetivo intenta remediar la laguna detectada en la literatura del tema, mencionada dentro de la categoría *fundamentos de la disciplina* del cuadro 2.7, pues no existe actualmente una comparación sistemática en texto libre.

- *Proponer estrategias de síntesis de muestras artificiales* para su uso en ataques de presentación que, capitalizando el conocimiento recabado durante el estudio de las anteriores, logren engañar a los actuales sistemas de autenticación basados en cadencias de tecleo con frecuencia suficiente como para constituir una amenaza. Este

objetivo surge de la tendencia detectada en la disciplina, indicada en el cuadro 2.6 bajo la sección *modelo de ataque*, de alejarse del esquema de evaluación de esfuerzo cero para emplear modelos de impostores entrenados o ataques de presentación. Se llevará a cabo empleando técnicas de síntesis basadas en distribuciones empíricas, un camino aun no explorado como se indica en el ítem *técnica de síntesis* de la categoría *ataque contra el sistema de autenticación* dentro del cuadro 2.7.

- *Evaluar un sistema de detección de vida* que sirva como contramedida de defensa ante las anteriores y otras estrategias de síntesis de muestras artificiales que constituyan el estado del arte. Este objetivo se llevará a cabo utilizando la técnica de modelado por contextos finitos, que no ha sido explorada aun para la tarea como se indica en la categoría *contramedidas de defensa* del cuadro 2.7.

- *Explorar las técnicas derivadas de los anteriores para su uso en ataques de canal lateral* pues el incremento en la capacidad para generar muestras sintéticas indistinguibles de las del usuario legítimo puede ser aprovechado en la identificación del texto ingresado. Al igual que el anterior, este objetivo se llevará a cabo utilizando la técnica de modelado por contextos finitos, que no ha sido explorada aun para la tarea como se indica en el ítem *técnica de síntesis* de la categoría *ataque contra el sistema de autenticación* dentro del cuadro 2.7.

- *Crear una herramienta* que permita integrar los resultados de las anteriores para síntesis de muestras artificiales en base a un perfil intrausuario, detección de vida, e identificación del texto ingresado con las muestras temporales resultante de un ataque por canal lateral.

3.3. Hipótesis

La principal contribución de esta tesis ser a un conjunto de estrategias de síntesis de muestras artificiales para su uso en ataques de presentación, y un esquema de detección de vida derivado de estas estrategias que sirva como contramedida de defensa al verificar con suficiente precisión que la muestra autenticada ha sido escrita por el usuario humano.

La hipótesis central que fundamenta las técnicas y métodos propuestos puede resumirse en la siguiente forma:

Ninguna distribución suave es adecuada para modelar en general todos los atributos temporales que caracterizan la cadencia de tecleo de un usuario; solo las distribuciones empíricas del perfil intrausuario son capaces de capturar con precisión su comportamiento característico, y este fenómeno puede ser capitalizado tanto para generar muestras sintéticas que logren engañar a los actuales sistemas de autenticación basados en cadencias de tecleo como para construir medidas de defensa eficaces contra ataques de presentación que distingan la escritura del usuario humano legítimo de una muestra construida artificialmente.

Categoría	Alcance	Excluye
Aplica al proceso de	Verificación	Identificación
Modelo de autenticación	Continua	Al inicio de la sesión
Tarea de escritura	Texto libre	Claves Texto fijo
Tamaño de muestra	Largo (≥ 150 caracteres)	Corto
Atributos	Parámetros temporales	Presión Velocidad Aceleración Rotación Otros
Resultado	Herramienta	Producto Biblioteca de software Resultado teórico
Capacidades	Síntesis de muestras Detección de vida Determinación de textos	
Defensa contra	Ataques de presentación	Ataques por canal lateral

Cuadro 3.1: Resumen del alcance

3.4. Alcance

El resumen del alcance puede verse en el cuadro 3.1. Los términos marcados a continuación en letra negrita corresponden a entradas del cuadro.

Los métodos propuestos son aplicables al proceso de **verificación** de identidad, como se ha definido en la sección 2.2.1. La integración en un sistema de identificación no se considerará.

Los experimentos planteados y los métodos propuestos a lo largo de esta tesis se restringirán a cadencias de tecleo en **textos libres** para la **autenticación continua**, como se definen en las secciones 5.1.2 y 2.2.1.1. El estudio de textos libres, en contraste con claves y textos fijos, ha sido señalado como una tendencia actual de la disciplina en el cuadro 2.6; en particular, se estudiarán **textos largos**, en contraste con, por ejemplo, las claves de usuario o las muestras de tamaño reducido. Esta restricción puede justificarse en la cantidad de caracteres que demandan los métodos que se exploran para alcanzar tasas de error aceptables. Se presumirá que las muestras cuentan con al menos 150 caracteres. Esto

permitirá también suplir una ausencia en la literatura sobre ataques por canal lateral, como se indicó en el cuadro 2.7, pues los métodos actuales se limitan a buscar textos cortos.

Los atributos que se utilizaran para la detección de vida se restringirán a los **parámetros temporales** que se han definido en la sección 2.2.3. No se utilizaran otros parámetros que pueden ser capturados con teclados especiales, como la presión realizada o la velocidad de contacto con la tecla, pues su uso no se encuentra extendido. Tampoco se utilizarán otros parámetros biométricos disponibles con dispositivos móviles, tales como aceleración, rotación, etc. De esta forma, el problema se plantea para el caso de mayor dificultad, que es cuando se cuenta con la mínima información biométrica posible.

Se desarrollará una **herramienta** que, dado un perfil intrausuario, será capaz de **sintetizar muestras artificiales** de atributos temporales para su uso en ataques de presentación y de realizar **detección de vida**; es decir, determinar si una muestra alegadamente perteneciente al usuario legítimo ha sido generada en forma artificial. Adicionalmente, la herramienta también permitir a **determinar el texto ingresado** en base a una lista de candidatos y una muestra temporal del usuario legítimo sin nombres de teclas, que se haya originado en un ataque de canal lateral.

La detección de vida se plantea como un mecanismo de **defensa contra ataques de presentación** exclusivamente. Por este camino no es factible defenderse contra otros tipos de ataques y, en particular, no es factible defenderse así contra ataques por canal lateral. La factibilidad de estos últimos depende fuertemente de la implementación y del resto de los módulos integrados en el sistema. Se excluye, por tanto, la consideración de medidas de defensa contra ataques por canal lateral del alcance´ de este estudio.

Parte II

La solución

Capítulo 4

Métodos propuestos

Es mejor encender una vela que
maldecir las tinieblas

Rabindranath Tagore

4.1. Introducción

Hemos notado, al reseñar el estado del arte y los modelos de ataque en la sección 2.2.6.1, que los sistemas de autenticación de usuarios por medio de cadencias de tecleo presentan vulnerabilidades frente al avance reciente en las técnicas de ataque. Al definir el problema en la sección 3.1 se ha establecido la necesidad de proponer e implementar mecanismos de defensa contra ataques de presentación, y se ha dejado fuera del alcance de esta indagación la defensa contra ataques de canal lateral pues los mismos suelen involucrar componentes ajenos al sistema de autenticación en sí mismo.

Es fútil proteger contra las amenazas de ayer sin prever las advertencias del mañana. No nos limitaremos en esta sección a intentar evitar los embates de un agresor armado con las técnicas del estado del arte. El camino será, en primer lugar, mostrar la factibilidad de defender eficazmente un sistema de autenticación de usuarios basado en cadencias de tecleo contra estas últimas. Luego, proponer estrategias de ataque que las superen demostrablemente, utilizando las tasas de error del sistema de defensa anterior como testigo de la mejora. Finalmente, mostrar que el sistema de protección puede fortalecerse incluso contra estas técnicas novedosas y en el caso más pesimista: cuando el agresor cuenta con la totalidad de la información en el perfil biométrico del usuario.

Para seguir este camino comenzaremos proponiendo una familia de distancias basadas en las distribuciones empíricas de los parámetros temporales, que serán cruciales durante la detección de falsificaciones sintéticas pero que también proveerán inspiración para la novedosa estrategia de ataque que obtendrá mejores resultados.

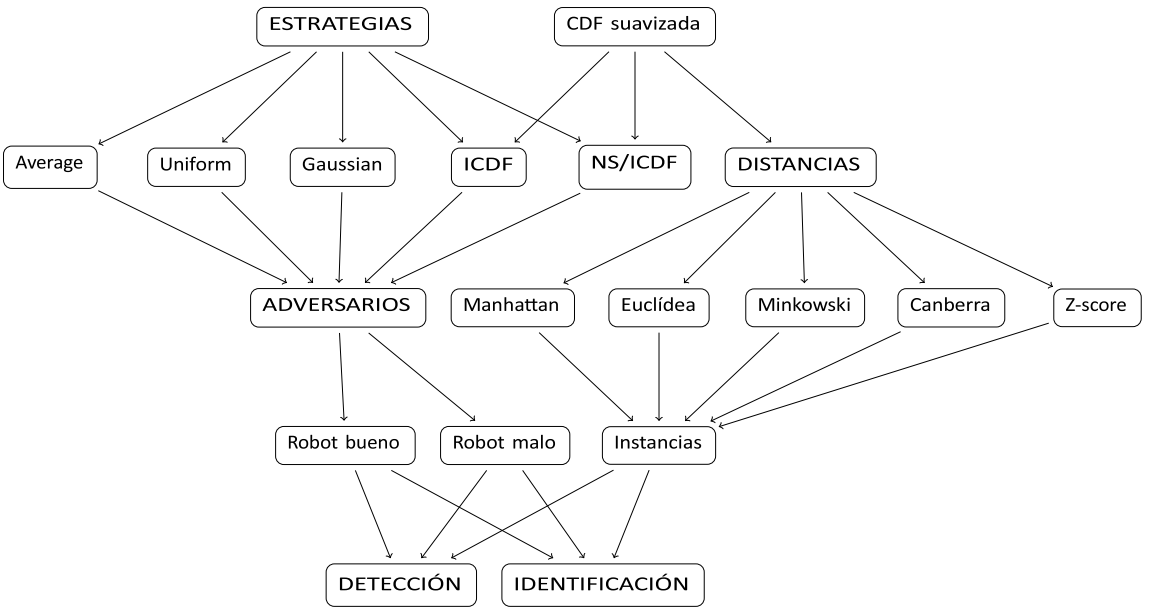

Figura 4.1: Diagrama de dependencias para los métodos propuestos

Luego se propondrá una familia de estrategias de síntesis que servirán como adversarios, pero que también cumplirán un rol fundamental durante el entrenamiento del clasificador que detectara falsificaciones sintéticas. Armados con ambas técnicas, podremos ya definir el esquema de defensa para fortalecer los sistemas de autenticación de usuarios por medio de cadencias de tecleo en textos libres. Finalmente, descubriremos que una modificación del esquema anterior permite, capitalizando las técnicas anteriores, empujar las fronteras de los ataques por canal lateral al permitir descubrir si una secuencia observada de tiempos de escritura, de latencia y/o de retención y desconociendo la secuencia de teclas que la originan, pertenece a alguno de los textos que se puede encontrar en una lista de candidatos que no es de tamaño reducido.

El resto de este capítulo está organizado de la siguiente manera. Como aquí se presentan diversos métodos y a los fines de facilitar la lectura, la subsección 4.1.1 describe las dependencias entre ellos mientras que la subsección 4.1.2 establece la notación general que se utilizará a lo largo del capítulo. La sección 4.2 propone nuevas distancias basadas en las distribuciones empíricas de los parámetros temporales. La sección 4.3 propone una familia de estrategias de síntesis de muestras temporales. La sección 4.4 detalla el esquema de detección de falsificaciones sintéticas propuesto. Finalmente, la sección 4.5 propone modificaciones al anterior para la identificación del texto escrito cuando no se cuenta con la secuencia de teclas, sino solamente los tiempos.

4.1.1. Dependencias entre los métodos y conceptos

La figura 4.1 muestra las dependencias entre los distintos conceptos y métodos propuestos en este capítulo. En la cima de la pirámide se encuentra la función CDF suavizada, que se obtiene a partir de la distribución de tiempos empírica y discreta

para una cierta tecla bajo un cierto contexto. Su descripción abarca la primer parte de la sección 4.2 y a partir de ella se derivan las distancias basadas en CDF, que abarcan el resto de la sección antedicha. Estas son, en cierta forma, equivalentes individualizados por un cierto usuario de las tradicionales distancias de Manhattan, euclídea, de Minkowski, Canberra, y Z-score. Las formas tradicionales de estas cinco son explicadas en el apéndice A.

Tres de las estrategias de síntesis (Average, Uniform, y Gaussian) no utilizan conceptos previos exceptuando el modelado por contextos finitos, que ha sido descrito en la sección 2.5. Tanto ICDF como NS/ICDF emplean como parte integral la función inversa de la CDF suavizada. Todas las estrategias de síntesis están descritas en la sección 4.3.

Los atributos resultantes de la comparación entre muestras con las distancias basadas en CDF son utilizados para formar instancias de entrenamiento/evaluación tanto para la detección de falsificaciones sintéticas, descrita en la sección 4.4, como para la identificación de textos, descrita en la sección 4.5. La selección de atributos, entre los cuales se cuentan aquellos basados en CDF pero también muchos otros, será descrita en las secciones 5.3.2.5 y 5.4.5. Ambos métodos, detección e identificación, utilizan internamente a los adversarios robot bueno y robot malo, que son presentados en la sección 4.3.

4.1.2. Notación general

Dada una secuencia de n teclas $k_1 \ldots k_n$, nuestro objetivo es doble. Al tomar el punto de vista de un atacante, intentamos generar un vector $\mathbf{t} = t_1 \ldots t_n$ de tiempos que imite a nuestro objetivo, el usuario humano. Pretendemos que el resultado sea lo suficientemente bueno como para engañar a un sistema de verificación de identidad. E inversamente, al tomar un punto de vista defensivo, queremos ser capaces de discernir tales falsificaciones sintéticas del comportamiento auténtico del usuario legítimo.

Ocurrirá también que, luego de un ataque de canal lateral exitoso, contemos con el vector de tiempos $\mathbf{t} = t_1 \ldots t_n$ pero no hayamos sido capaces de obtener la secuencia de teclas $k_1 \ldots k_n$ correspondiente, o que la hayamos obtenido solo en forma parcial.

Todos los métodos que se describen a continuación utilizarán, en alguna de sus etapas, el modelado de cadencias de tecleo por medio de contextos finitos como se ha descrito en 2.5. El subíndice i se utilizará para iterar sobre las distintas componentes de un vector de tiempos o sus teclas correspondientes. Para cada tecla k_i de la secuencia $k_1 \ldots k_n$, se agrupan en un conjunto S_i todas las observaciones pasadas de las características temporales que se encuentran disponibles en el perfil de usuario y que están precedidas por el contexto de mejor coincidencia $k_{i-m} \ldots k_{i-1}$, de orden m. Puede ocurrir que m sea cero. El tamaño del conjunto S_i se denotara, en forma´ estándar, como $|S_i|$.

Distancia	Tradicional	CDF
Manhattan (L_1)	$\frac{1}{n}\sum_{i=1}^{n}\left\|\frac{r_i-s_i}{\sigma_{S_i}}\right\|$	$\frac{1}{n}\sum_{i=1}^{n}\left\|F_{S_i}(r_i)-F_{S_i}(s_i)\right\|$
Euclídea (L_2)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{r_i-s_i}{\sigma_{S_i}}\right)^2}$	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(F_{S_i}(r_i)-F_{S_i}(s_i)\right)^2}$
Minkowski (p = 0, 4)	$\left(\frac{1}{n}\sum_{i=1}^{n}\left\|\frac{r_i-s_i}{\sigma_{S_i}}\right\|^{0,4}\right)^{\frac{1}{0,4}}$	$\left(\frac{1}{n}\sum_{i=1}^{n}\left\|F_{S_i}(r_i)-F_{S_i}(s_i)\right\|^{0,4}\right)^{\frac{1}{0,4}}$
Canberra	$\frac{1}{n}\sum_{i=1}^{n}\frac{\|r_i-s_i\|}{\|r_i\|+\|s_i\|}$	$\frac{1}{n}\sum_{i=1}^{n}\frac{\left\|F_{S_i}(r_i)-F_{S_i}(s_i)\right\|}{\left\|F_{S_i}(r_i)\right\|+\left\|F_{S_i}(s_i)\right\|}$
Z-score	$\frac{1}{n}\sum_{i=1}^{n}u(\|r_i-\mu_{S_i}\|-\sigma)$	$\frac{1}{n}\sum_{i=1}^{n}u\left(\left\|F_{S_i}(r_i)-0{,}5\right\|-\sigma\right)$
Distancia resultante	$\|Z(\mathbf{r})-Z(\mathbf{s})\|$	

Cuadro 4.1: Vista comparativa de las distancias tradicionales y basadas en CDF

4.2. Distancias basadas en las distribuciones empíricas

Ya se ha atisbado en la sección 2.4.5, y se confirmara en la sección 6.1.2 de resultados del experimento correspondiente, que las distribuciones empíricas de los parámetros temporales distan mucho de ser ideales. Nuestro objetivo es capitalizar tanto las irregularidades generales como las particularidades a nivel de usuario para potenciar los métodos de generación de imitaciones sintéticas y de defensa contra los mismos.

Veremos que la utilización de distribuciones empíricas para generar imitaciones sintéticas es un proceso directo. No es el caso al intentar utilizarlas para la verificación, pues nos enfrentamos a un problema de doble filo. Por un lado, aunque estas distribuciones tienden a ser sesgadas a la derecha y de cola larga, no siempre lo son y, al mostrar irregularidades y picos inconsistentes, pueden diferir significativamente de sus aproximaciones suaves y que decrecen monótonamente hacia la derecha. No solo son las distribuciones empíricas raramente gaussianas, sino que incluso fallan una prueba de hipótesis para el mejor ajuste contra una log-normal entre el 10% y el 20% de las veces, como se mostrara en la sección 6.1 [109]. Por otro lado, no esperamos que las muestras individuales bajo evaluación contengan suficientes observaciones de tiempo para cada tecla y contexto como para construir un histograma lo suficientemente detallado que permita hacer un test de hipótesis. Lo que sí podemos esperar es tener datos de sobra en el perfil de usuario, a partir de los cuales se pueda construir un histograma empírico y compararlo con los tiempos de la muestra.

4.2. Distancias basadas en las distribuciones empíricas

Para este propósito, supongamos que disponemos del conjunto S_i de observaciones temporales previas t_j para la tecla k_i, que sigue al contexto de mejor coincidencia $k_{i-m} \ldots k_{i-1}$ de orden m. Supongamos además que los t_j están ordenados por magnitud, de forma tal que $t_1 \leq t_2 \leq \cdots \leq t_r$, donde $r = |S_i|$. En este caso, la función discreta de distribución acumulada para S_i será

$$CDF_{S_i}(t) = \frac{max\{j; t_j \leq t\}}{|S_i|} \tag{4.1}$$

Esta es una función constante por partes, con saltos en algunos de los t_j. No todos los saltos tienen una magnitud de $|S_i|^{-1}$, ya que los valores t_j pueden repetirse; por ejemplo, como los temporizadores de Windows tienen una resolución aproximada de 15,6 ms, se puede esperar que muchas observaciones de tiempo tengan los mismos valores exactos como 15,6 ms, 31,2 ms, etc. En vista de estas repeticiones, un salto de

$$CDF_{S_i}(t_j) - CDF_{S_i}(t_j - \epsilon) = \frac{\#\{m, t_j = t_m\}}{|S_i|} \tag{4.2}$$

se observará en cada $t_j \in S_i$. O no se observará, si $t_j = t_{j-1}$. Por ser una función de distribución acumulada, CDF_{S_i} cumple las propiedades

$$CDF_{S_i}(t) = 0 \qquad \forall t \leq 0 \tag{4.3}$$

$$CDF_{S_i}(t) = 1 \qquad \forall t \geq t_{|S_i|} \tag{4.4}$$

Para nuestros propósitos necesitaremos convertir $CDF_{S_i}(t)$ en una función continua, por lo que definimos $F_{S_i}(t)$ como la función lineal por partes que concuerda con $CDF_{S_i}(t)$ en 0,1, y todos los t_j, en donde estos están unidos por segmentos lineales. Formalmente

$$F_{S_i}(t) = CDF_{S_i}(t) = 0 \qquad \forall t \leq 0 \tag{4.5}$$

$$F_{S_i}(t) = CDF_{S_i}(t) = 1 \qquad \forall t \geq t_{|S_i|} \tag{4.6}$$

$$F_{S_i}(t) = CDF_{S_i}(t_{j-1}) + \frac{CDF_{S_i}(t_j) - CDF_{S_i}(t_{j-1})}{t_j - t_{j-1}} \cdot (t_j - t_{j-1}) \qquad t_{j-1} \leq t \leq t_j \tag{4.7}$$

Nótese que la cuarta condición implica que

$$F_{S_i}(t_j) = CDF_{S_i}(t_j) \tag{4.8}$$

Ahora, F_{S_i} es monótona creciente y tiene por rango todo el intervalo cerrado [0,1], así que su función inversa está bien definida dentro de este dominio completo. Luego, para la síntesis de cadencias de tecleo podemos muestrearla con una variable uniforme en [0, 1], obteniendo el tiempo t_i de la tecla k_i, de la cual hemos extraído el conjunto S_i, en la forma

$$t_i = F_{S_i}^{-1}(U[0,1]) \tag{4.9}$$

Este proceso, que ahora solo se menciona al pasar, será descrito en detalle más adelante en las secciones 4.3.3 y 4.3.4. He aquí un pequeño abuso de notación, pues t_i en la ecuación 4.9 refiere a un tiempo sintetizado y no a uno de los valores $t_j \in S_i$ referenciados en las ecuaciones anteriores. Confiamos en que esta aclaración bastará para desambiguar el uso, exclusivo de la ecuación anterior.

De la definición de F_{S_i}, es inmediato que

$$min\ \{F_{S_i}(t)\} = F_{S_i}(0) = 0 \tag{4.10}$$

$$máx.\ \{F_{S_i}(t)\} = F_{S_i}(t_{|S_i|}) = 1 \tag{4.11}$$

Por lo que podemos calcular, utilizando el parámetro p, un equivalente de la familia de distancias de Minkowski entre vectores de tiempo $\mathbf{r} = r_1 \ldots r_n$ y $\mathbf{s} = s_1 \ldots s_n$, en la forma

$$d_{ICDF}(r,s) = \left(\frac{1}{n}\sum_{i=1}^{n}\left|F_{S_i}(r_i) - F_{S_i}(s_i)\right|^p\right)^{\frac{1}{p}} \tag{4.12}$$

Denominamos CDF a esta familia por el acrónimo inglés de *cumulative distribution function*, es decir *función de distribución acumulada*. Eligiendo $p = 1$ obtenemos una distancia análoga a la de Manhattan, pero que considera la forma de las distribuciones empíricas en lugar de solo la media y varianza extraídas de los conjuntos S_i. Idénticamente, $p = 2$ nos devuelve un equivalente de la distancia euclídea y $p = 0{,}4$ la distancia de Minkowski optimizada de [35]. Nótese que ninguna de las tres anteriores, ni las distancias que se describirán a continuación, asumen simetría ni monotonía a izquierda y derecha en las distribuciones subyacentes. Yendo más lejos, una analogía CDF de la distancia de Canberra escalada se puede escribir como

$$d(r,s) = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|F_{S_i}(r_i) - F_{S_i}(s_i)\right|}{\left|F_{S_i}(r_i)\right| + \left|F_{S_i}(s_i)\right|} \tag{4.13}$$

4.2. Distancias basadas en las distribuciones empíricas

También podemos recuperar una versión basada en CDF del tradicional Z-score o conteo de valores atípicos descrito en el apéndice A.7, que no es una distancia *per se* sino un atributo derivado de un único vector. La ecuación es la misma que para el Z-score gaussiano, pero como por definición

$$\mathbb{E}\left[F_{S_i}\right] = 0{,}5 \tag{4.14}$$

la media muestral puede reemplazarse por el valor fijo 0,5. Si $u(x)$ es la función escalón, que es cero para $t < 0$ y uno para $t \geq 0$, y utilizando un valor límite σ para detectar los valores atípicos, tenemos que

$$\mathcal{Z}(r) = \frac{1}{n}\sum_{i=1}^{n} u\left(\left|F_{S_i}(r_i) - 0{,}5\right| - \sigma\right) \tag{4.15}$$

devuelve la proporción de valores que se encuentran a más de σ del valor esperado de $F_{Si}(r_i)$, pero considerando las particularidades de la distribución empírica

Estrategia	Devuelve	Se asemeja a
Average	μ_i	
Uniform Noise	$\mathrm{U}(\mu_i, \sigma_u)$	NoiseBot [27, 29]
Gaussian Noise	$\mathrm{N}(\mu_i,\sigma_u)$	GaussianBot [27, 29], [26], [71]
LBMC/HMM	complejo, ver [28]	ver [28]
ICDF	$F_{S_i}^{-1}(U[0,1])$	
NS/ICDF	$\eta_i + F_{S_i}^{-1}(U[0,1])$	

Cuadro 4.2: Estrategias de síntesis en orden de complejidad creciente

registrada por S_i. Eligiendo un valor de $\sigma = 0,45$ detectamos valores de cola a ambos lados con ≤ 5 % de probabilidad acumulada.

Como esperamos que el mismo usuario legítimo produzca una cantidad estable de valores atípicos a lo largo de sus sesiones, los conteos de valores atípicos basados en CDF pueden convertirse en una distancia tomando el valor absoluto de la resta entre los correspondientes Z de los vectores comparados, en la forma

$$d\,(r,s) = \left|\mathcal{Z}\,(r) - \mathcal{Z}\,(s)\right| \tag{4.16}$$

Si bien es factible, en la misma forma que en la ecuación 4.12, utilizar un parámetro p para penalizar mayores (o menores desviaciones) y así generar una familia derivada de métricas, esta posibilidad no será considerada aquí sino pospuesta hacia futuras líneas de investigación.

De estas cinco maneras, análogas a las distancias de Mahattan, euclídea, de Minkowski optimizada, de Canberra, y Z-score, hemos intentado capturar globalmente que tan bien los componentes de los vectores de tiempo corresponden a las distribuciones empíricas de tiempos, como han quedado registradas en el perfil de usuario, para los contextos elegidos a través del modelado por contextos finitos. A diferencia de las distancias de Manhattan, euclídea y otras, las ecuaciones 4.12 y 4.13 no presuponen una determinada distribución teórica de los valores, no presuponen simetría, ni presuponen monotonía hacia la izquierda o hacia la derecha de la media. Esto nos permite distinguir el comportamiento ruidoso del usuario legítimo de intentos de suplantación de identidad mediante distribuciones más suaves, con mejor comportamiento. Un resumen comparativo puede verse en la tabla 4.1. Para una explicación de las métricas tradicionales y sus fundamentos algebraicos, debe consultarse el apéndice A.

Una pequeña ventaja de la que gozan las distancias basadas en CDF, de cara a su empleo ulterior para la clasificación, es que siempre se encuentran en el rango entre cero y uno. Por este motivo, no necesitan ser escaladas al momento de entrenar y verificar los vectores resultantes.

4.3. Estrategias de síntesis

Comenzaremos introduciendo, en orden de complejidad creciente, una serie de estrategias para generar vectores de tiempos sintéticos. Estas se encuentran resumidas en la tabla 4.2. Con excepción de LBCM/HMM, todas ellas se basan, directa o indirectamente, en el modelado por contextos finitos.

Dada una secuencia de n teclas $k_1 \ldots k_n$, la síntesis del correspondiente vector de tiempos $\mathbf{t} = t_1 \ldots t_n$ se realiza en forma iterativa, tecla a tecla. Para cada k_i, el modelador por contextos finitos recopila el conjunto S_i de observaciones pasadas de las características temporales representativas, seleccionadas con el contexto de mejor coincidencia. En lo que sigue, podemos ignorar el orden y las teclas del contexto; para nuestros fines alcanza tratar el modelado por contextos finitos como una caja negra y suponer que contamos con S_i.

Haciendo un pequeño abuso de notación, pero sin riesgo de confundir al lector, pues el contexto determinara claramente el uso, se emplearán las letras U y N no solo para denotar las distribuciones de las variables aleatorias uniforme y gaussiana, sino también para denotar una muestra en particular de ellas. De esta forma, $t_i = \mathrm{U}[0,1]$ no significa que t_i es la distribución uniforme en el intervalo $[0,1]$ sino un escalar que se obtuvo muestreando $\mathrm{U}(0,1)$.

Se opta por conservar los nombres en inglés para las estrategias de síntesis, resaltando la similitud con aquellas propuestas previamente en la literatura del tema. No se describirá en esta sección la estrategia LBMC/HMM, que se ha reseñado en 2.4.6, pues difiere sustancialmente de los métodos aquí empleados. Uniform y Gaussian, aunque no son originales, caben dentro del tratamiento unificado de esta sección.

4.3.1. Average

Entre todas las estrategias para generar el tiempo falsificado t_i en base a S_i, la opción más simple y que se muestra en la figura 2.5, es promediar todas las observaciones pasadas seleccionadas por contexto, eligiendo

$$t_i = \mu_i \tag{4.17}$$

en donde

$$\mu_i = \frac{1}{|S_i|} \sum_{t \, \epsilon \, S_i} t \tag{4.18}$$

Por trivial que pueda parecer esta estrategia, al discutir los resultados descubriremos que requiere un tratamiento especial, ya que siempre representa un desafío incluso para los clasificadores entrenados con adversarios sofisticados.

4.3.2. Uniform y Gaussian

Las dos estrategias siguientes, Uniform y Gaussian, introducen algo de ruido aditivo, pero se hallan aun en los primeros peldaños de la escalera de la complejidad. El desvío estándar del conjunto S_i puede calcularse como

$$\sigma_i = \sqrt{\frac{1}{|S_i|} \sum_{t \, \epsilon \, S_i} (t - \mu_i)^2} \tag{4.19}$$

Lo que nos permite introducir ruido aditivo muestreando una variable uniforme de media μ_i y desvío estándar σ_i, directamente en la forma

$$t_i = U(\mu_i, \sigma_i) \tag{4.20}$$

Idénticamente, muestreando una variable gaussiana de media μ_i y desvío estándar σ_i, tenemos

$$t_i = N(\mu_i, \sigma_i) \tag{4.21}$$

Al fijar el orden máximo del contexto a uno, estas opciones corresponden respectivamente a los programas llamados NoiseBot y GaussianBot en [27] y [29], y esta última a la estrategia de síntesis utilizada en [26] y [71] para generar impostores.

4.3.3. ICDF

La cantidad de distribuciones posibles para reemplazar a la uniforme y la gaussiana, incluso luego de filtrarlas bajo los criterios que se describirán en la sección 5.3.1.2, resulta innumerable. Es legítimo preguntar hasta qué punto pueden obtenerse mejoras de esta forma, pero en lugar de evaluar distribuciones más complejas preferimos saltar a un enfoque radicalmente distinto: capitalizar las distribuciones empíricas que se reflejan en cada conjunto S_i. De esta forma, el sesgo a la derecha y las colas largas de los valores de tiempo que suelen caracterizar las distribuciones temporales en los perfiles de cadencia de tecleo se tienen en cuenta sin complicaciones adicionales, y se salvan sin esfuerzo todas aquellas excepciones a la regla. Lo que es más, cuando existen tales excepciones estas son utilizadas para reforzar la precisión del método.

Se ha indicado más arriba, en la sección 4.2, como extraer del conjunto S_i una función de distribución acumulada F_{S_i}, continua y que admite una función inversa $F_{S^{-}i}{}^{1}$ con dominio de definición en el intervalo $[0, 1]$. Al muestrear esta última con una variable aleatoria uniforme en el mismo intervalo,

$$t_i = F_{S_i}^{-1}(U[0,1]) \tag{4.22}$$

obtenemos la estrategia de síntesis ICDF, acrónimo del inglés *inverse of the cumulative distribution function*; es decir, *función inversa de la distribución acumulada*.

Se verá en la sección 6.2 de resultados que esta estrategia produce, en promedio y a la larga, distribuciones locales para los tiempos sintéticos que son indistinguibles

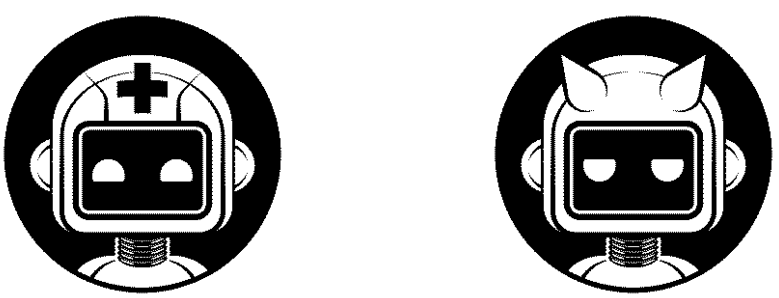

Figura 4.2: Robot bueno y robot malo quieren saludan al lector antes de sintetizar muestras

de los del usuario real. Lo que es más importante es que esto ocurre tanto si sus parámetros temporales obedecen a una de las funciones de probabilidad suaves y de buen comportamiento que han sido utilizadas en estudios anteriores, p.ej. log-normal, como si estos siguen un patrón característico y posiblemente irregular. Intentamos de esta forma restringir al clasificador de un sistema defensivo, quitándole la posibilidad de utilizar la forma empírica de las distribuciones de tiempo para discriminar entre una muestra real y una sintética, falsificada.

4.3.4. NS/ICDF

Finalmente, relajamos la estacionariedad de la estrategia anterior con una idea tomada de [28]. En lugar de confiar en un modelo de Markov oculto, que parece ser la mejor opción cuando ignoramos las teclas que corresponden a los tiempos, aprovechamos nuevamente el marco de modelado con contextos finitos, pero a nivel de fronteras entre palabras y oraciones.

Al comenzar el proceso de síntesis, procesamos todas las muestras del usuario legítimo separándolas por palabras y calculando el valor promedio, intra-palabra, de sus parámetros temporales. Pongamos que su desvío estándar es σ.

Para generar los tiempos falsificados, la estrategia NS/ICDF devuelve

$$t_i = \eta_i + F_{S_i}^{-1}(U[0,1]) \tag{4.23}$$

que es un valor idéntico al de ICDF con la excepción del término de compensación η_i. Este último no varía a nivel de tecla sino de palabra, por lo que

$$\eta_i = \eta_i - 1 \tag{4.24}$$

siempre que k_i no sea una tecla que marque el fin de una palabra, como puede ser puntuación, la tecla de espacio, o teclas especiales. Pero si se da este caso,

$$\eta_i = N(0, \sigma) \tag{4.24}$$

En esta forma se intenta forzar la secuencia temporal a ser no estacionaria, para simular parcialmente el efecto de la alternación de distintos estados mentales durante el proceso de escritura [110]

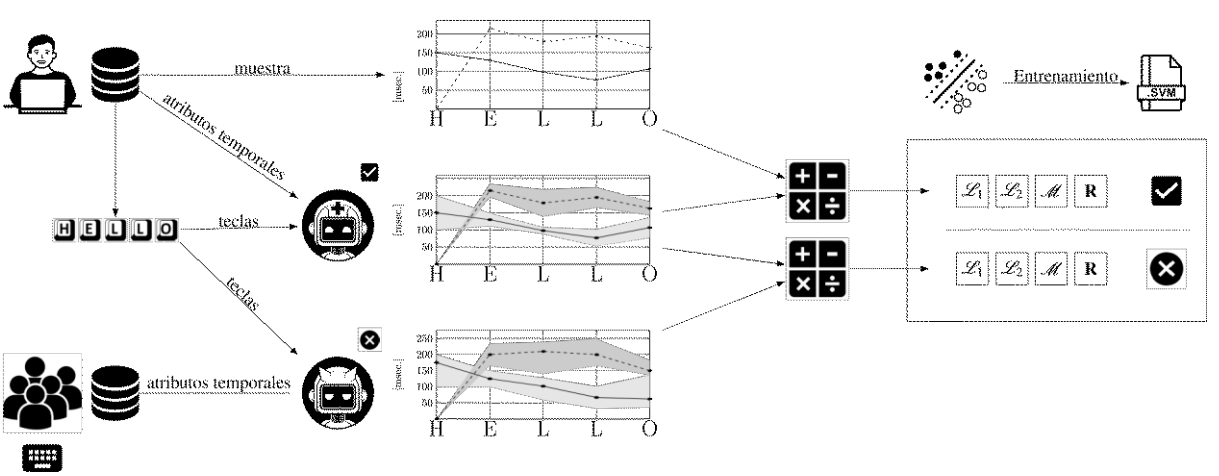

Figura 4.3: Esquema de entrenamiento para la detección de falsificaciones sintéticas

Wu y Liu han mostrado la existencia de variaciones en el factor de escala y el valor promedio de las distribuciones con la complejidad de la palabra o su frecuencia de uso [111].

La mejora de este modelo para utilizar tales criterios lingüísticos, que promete ser fructífera, queda fuera del alcance de esta tesis y se pospone a las futuras líneas de investigación.

4.3.5.Estrategia de rescate ante ausencia de datos temporales

Puede ocurrir que el conjunto S_i de observaciones temporales pasadas se encuentre vacío. Al modelar con contextos finitos se intenta encontrar el contexto de mejor coincidencia, y se acepta reducir progresivamente el orden del contexto si no hay observaciones suficientes. Sin embargo, incluso un contexto de orden cero puede devolver un S_i vacío, o de tamaño menor al necesario para, por ejemplo, calcular su desvío estándar.

Es esperable que las secuencias de teclas más comunes, entre ellas las alfanuméricas, tengan suficientes observaciones incluso con pocas muestras de texto. Si la tecla en cuestión no aparece suficientes veces como para formar un conjunto S_i del tamaño mínimo necesario, nos encontramos ante una tecla especial y un evento de poca frecuencia. Por este motivo utilizar un modelo simplificado para esta excepción no debería afectar en demasía la síntesis de un texto.

En estos casos, que constituyen un porcentaje despreciable del total, basta con muestrear una variable gaussiana, con el tiempo promedio del usuario para todas las teclas y el desvío muestral obtenido sobre el mismo conjunto.

4.4. Detección de falsificaciones sintéticas

Para defendernos de las falsificaciones sintéticas, haremos que estas mismas estrategias de suplantación de identidad se contrarresten entre sí entrenando un clasificador binario que las utilice como adversarios. *Robot bueno* y *robot malo*, quienes

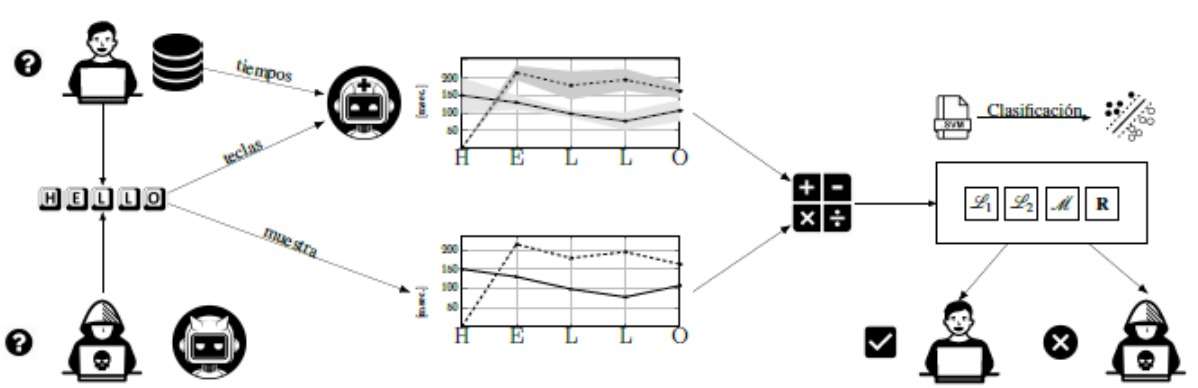

Figura 4.4: Esquema de evaluación para la detección de falsificaciones sintéticas.

saludan al lector desde la figura 4.2, serán los programas que utilicen una o más de las estrategias descritas en la sección anterior para generar falsificaciones sintéticas de muestras de un usuario humano objetivo. Robot bueno intentara imitar al usuario legítimo lo mejor posible, para determinar cómo escribiría un texto que aún no hemos observado y confrontarlo

con muestras a verificar. Robot malo hará las veces de atacante y servirá para mostrarle al clasificador las características de una muestra falsificada.

Utilizaremos muestras legítimas del usuario humano junto con muestras falsificadas que comparten la misma secuencia de teclas. Al ensenarle al clasificador el comportamiento de las distintas estrategias de síntesis, esperamos que la capacidad de generalización de los métodos modernos de aprendizaje automático le permitan luego detectar muestras falsificadas con otros perfiles de usuarios, con estadísticas de la población general, o incluso con perfiles parciales o completos del usuario legítimo.

Se asumirá de aquí en más que un potencial atacante tiene acceso a cantidades masivas de datos de la población general, en forma de conjuntos de datos que se encuentran a disposición del público. Se han reseñado muchos de ellos en la sección 2.4.1. De esta forma, no imaginamos una restricción arbitraria al inventar falsificaciones interusuario y preferimos, para cada conjunto de datos de evaluación, poner la totalidad de las sesiones que no pertenecen al usuario objetivo a disposición de los algoritmos de síntesis.

Supondremos también que un atacante sofisticado puede disponer de muestras intrausuario, robadas del perfil del usuario objetivo o capturadas por medio de un ataque de canal lateral. Sin embargo, dada la diferencia en dificultad en el acceso a ambos tipos de muestras, interusuario e interusuario, se evaluarán los dos casos separadamente.

Por supuesto que un defensor cuenta con todos los datos del perfil del usuario legítimo, en todos los casos. No hay, por tanto, razón alguna para no aprovechar esta información cuando se entrena al clasificador con estrategias adversarias.

El esquema de entrenamiento para la detección de falsificaciones sintéticas se muestra en la figura 4.3. Para cada muestra de texto en el perfil de usuario, se extrae la secuencia de teclas presionadas y se envían como entrada a ambos. Robot bueno emplea esa secuencia y los datos del perfil del usuario legítimo para falsificar los tiempos correspondientes, mientras que robot malo hace lo mismo, pero con datos

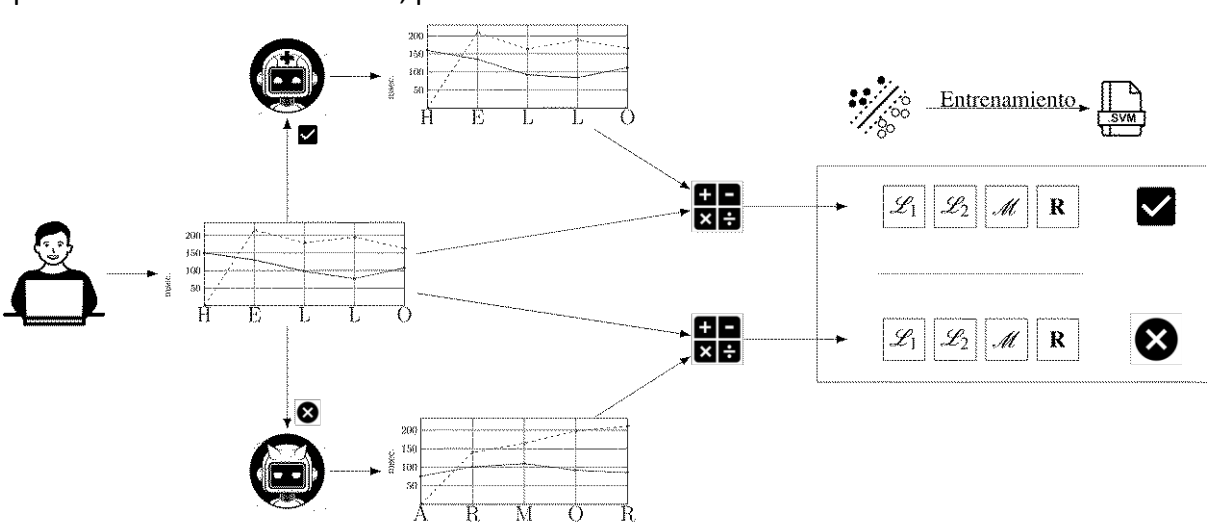

Figura 4.5: Esquema de entrenamiento para la identificación de textos

de la población general. Nótese que este proceso asegura naturalmente el balance de clases durante el entrenamiento, pues exactamente la mitad de las instancias presentadas corresponden a robot bueno y la otra mitad a robot malo.

Mas tarde, evaluaremos el sistema bajo una condición más severa: robot malo también podrá tener acceso a datos intrausuario, sin cambiar el resto de la configuración. Para cada muestra del usuario legítimo, se generan dos vectores de atributos derivados: uno que compara la muestra real con el intento de falsificación de robot bueno, marcada como legítima, y otro que compara los intentos de ambos robots, marcada como impostor. Estos vectores se utilizan para entrenar un clasificador binario.

En contraste con la dificultad habitual al verificar cadencias de tecleo en textos libres, donde las muestras de entrenamiento casi nunca comparten secuencias de teclas suficientemente largas con la muestra que se intenta verificar, aquí la comparación es sencilla pues los textos de la muestra del usuario y ambos intentos de falsificación son idénticos, tecla a tecla. Esta particularidad de los problemas que nos competen, que en cierto sentido los simplifica en contraste con el problema de verificación de texto libre en general, ha sido discutida en el apéndice A. Permite, por ejemplo, definir la distancia R_{all} como se explica en A.8.1.

La configuración de evaluación se muestra en la figura 4.4. Cuando se enfrenta a una muestra de escritura desconocida, el sistema solicita a robot bueno que sintetice una muestra falsificada utilizando la misma secuencia de teclas y el perfil del supuesto usuario. La muestra original y la muestra sintetizada son comparadas para generar las mismas características derivadas que se calcularon durante el entrenamiento, y el clasificador determina si estamos frente al usuario humano o una muestra sintética inyectada por un atacante.

4.5. Identificación de textos

Nuestro objetivo en esta sección es descubrir si una secuencia observada de tiempos de escritura, de latencia y/o de retención y de la cuál desconocemos la secuencia

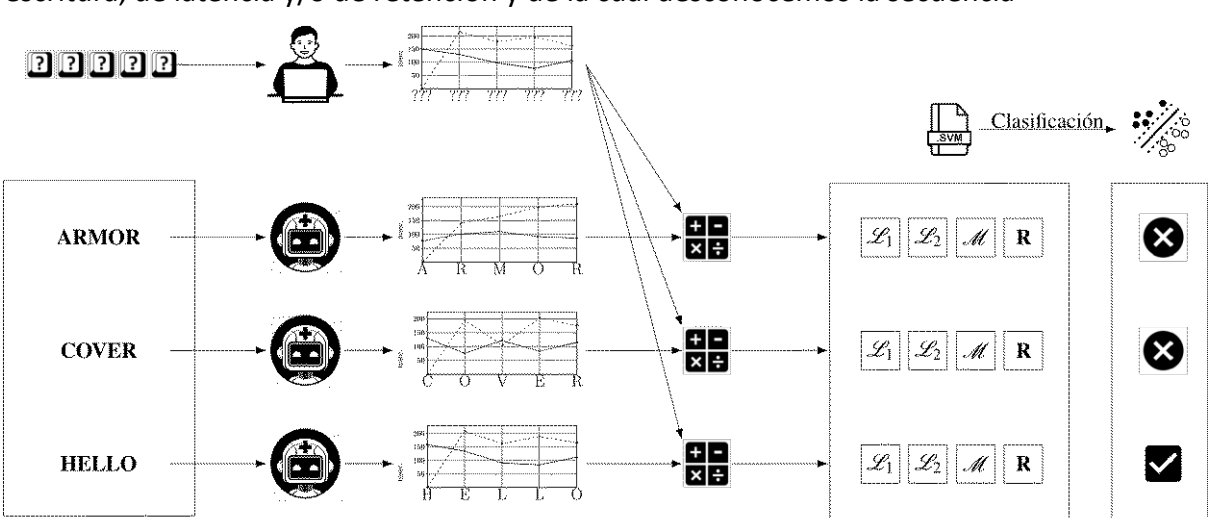

Figura 4.6: Esquema de evaluación para la identificación de textos

de teclas que la origina, pertenece a alguno de los textos que se puede encontrar en una lista de candidatos. Asumimos que la lista de candidatos no es pequeña, pudiendo incluir algunos cientos de textos posibles. Sin embargo, como no restringimos los candidatos a textos breves o palabras sueltas, no podemos esperar tener suficientes muestras pasadas (si acaso las haya) de cada una de ellas en el historial de usuario registrado como para entrenar el modelo. Como suele ocurrir al analizar cadencias de tecleo en textos libres, en escasísimas ocasiones dos muestras contendrán exactamente el mismo texto. De hecho, sería raro tener siquiera una instancia de la misma secuencia de teclas que esperamos identificar. Al igual que para la síntesis de muestras, utilizaremos el modelado por contextos finitos para reconstruir la forma en que el usuario teclearía idealmente los textos candidatos, y de esta manera se compensa la ausencia de muestras con el mismo texto.

Trabajaremos otra vez con robot malo y robot bueno, que han sido presentados en la sección anterior, brindando a ambos la posibilidad de contar no sólo con información interusuario sino también con la totalidad de la información intrausuario, en dos experimentos distintos para evaluar la precisión en ambos casos. A diferencia del esquema de detección de falsificaciones sintéticas de la sección anterior, aquí robot bueno y robot malo intentarán sintetizar textos distintos. Robot bueno siempre seguirá la misma secuencia de teclas de la muestra considerada que el usuario legítimo ha ingresado ya en el sistema, intentando reproducir su cadencia de tecleo de la mejor manera posible, pero sin contar con los parámetros temporales originales de la muestra. Robot malo hará un intento similar, pero no contará con la secuencia de teclas de la muestra original sino solo con su largo total. Un esquema de este proceso puede observarse en la figura 4.5, donde dada una muestra en donde el usuario objetivo ha escrito la palabra HELLO, robot bueno forja una imitación muy cercana también para HELLO mientras que robot malo, desconociendo que ha escrito el usuario, forja una imitación para la palabra ARMOR, del mismo largo.

Contrariamente a los métodos de la sección anterior, que nos demandarán evaluar todas las combinaciones de estrategias de síntesis para robot malo y robot bueno, aquí la elección es más sencilla. Queremos que robot bueno represente de la mejor manera posible la cadencia de tecleo del usuario, en promedio. No estamos interesados en agregar ruido o perfeccionar la replicación del comportamiento en las colas de las distribuciones, ni en descartar posibles muestras por eventuales comportamientos atípicos, sino todo lo contrario; buscamos representar de la manera más pura posible como el usuario legítimo habría escrito cada texto en la lista de candidatos. Por estos motivos, es esperable que la mejor elección de estrategia tanto para robot bueno cómo para robot malo sea Average, que a diferencia de todas las otras carecen de todo ruido aditivo. Esta intuición se ver a confirmada en la sección 6.3 al discutir los resultados.

La comparación de ambas muestras sintéticas con la muestra original del usuario legítimo produce dos vectores que son utilizados para entrenar un clasificador binario. De esta forma, el clasificador aprende a distinguir en base a los atributos derivados si una secuencia de parámetros temporales corresponde a un texto determinado o no. Nótese que este proceso asegura naturalmente el balance de clases durante el entrenamiento, pues exactamente la mitad de las instancias presentadas corresponden al mismo texto y la otra mitad no lo hacen. El conjunto de atributos utilizados para comparar las muestras es el mismo que se utiliza para la detección de falsificaciones sintéticas, y será detallado en la sección 5.3.2.3.

4.5. Identificación de textos

La figura 4.6 muestra el proceso para la identificación del texto tipeado entre aquellos que se encuentran en una lista de candidatos, una vez que hemos entrenado un modelo del usuario como se ha descrito. Podemos observar en la esquina superior izquierda de la figura que contamos con una secuencia de parámetros temporales originados por el usuario legítimo, pero no los nombres de las teclas asociadas. Dada una lista de candidatos del mismo largo, que aquí son las palabras ARMOR, COVER, y HELLO, se solicita a robot bueno que, utilizando el mismo perfil (intrausuario o interusuario) y la misma estrategia de síntesis empleados durante el entrenamiento, forje una muestra de tiempos para cada texto en la lista de candidatos. Luego comparamos cada una con la muestra original sin nombres de teclas utilizando los mismos atributos derivados que durante el entrenamiento. El clasificador, previamente entrenado, hará el resto de la tarea y nos indicar a con cuáles textos es esperable que coincida la secuencia de tiempos observada.

Capítulo 5

Marco experimental

> Los principios fundamentales se esconden, casi invisibles, tras una plétora de detalles técnicos
>
> Hermann Weyl

La *metodología* es el marco conceptual dentro del cual se lleva a cabo la investigación, que guía las decisiones del investigador en base a un esquema lógico de eficacia demostrada [112]. Tres asuntos fundamentales que trata el estudio de los marcos metodológicos son la elección o recolección de los datos, el diseño del experimento, y las técnicas de validación de los resultados.

Justificar la importancia de los anteriores no es difícil. Un experimento perfectamente diseñado pero llevado a cabo con datos erróneos o no relevantes es incapaz de entregar conclusiones generalizables al mundo real por más promisorios que sean sus resultados. Configura un mero pleonasmo afirmar que todas las teorías son ciertas en todos aquellos casos en las que se cumplen; se acerca más a la verdad enfatizar, con ironía, que solo en aquellos casos y en ningún otro. Idénticamente, un experimento que padece de un mal diseño es, como el reloj detenido que da bien la hora dos veces por día, una máquina de convertir datos válidos en conclusiones arbitrarias.

Es más sutil la consideración sobre la validación de los resultados de los experimentos. Argumentar la calidad de los datos, su relevancia para el problema, lo adecuado del entorno y las condiciones de captura, no deja de ser un problema sencillo excepto en casos extremos. Demostrar la validez de los procedimientos experimentales es más complejo, pero no deja de ser un problema estructurado. Lo que es más, tanto el acceso abierto de las publicaciones como el código libre que predomina hoy en día en las ciencias de la computación mitigan notoriamente el riesgo de que un error de diseño o una mala implementación pasen desapercibidos.

Demostrar la validez de los resultados: he aquí el problema crucial que demanda todo el ingenio del investigador. Se utilizarán fundamentalmente tres técnicas para asegurar que los resultados reportados en la sección siguiente no obedecen a artefactos, a interpretaciones subjetivas, o a equívocos. En primer lugar, la inferencia estadística, utilizando pruebas de hipótesis e intervalos de confianza para el reporte de los resultados. En segundo lugar, la

evaluación comparativa con referencias, denominada *benchmarking*, para contrastar el rendimiento de los métodos propuestos contra los existentes en el estado del arte. Finalmente, la estructuración de los experimentos en la forma de estudios prospectivos del tipo caso/control utilizando tres conjuntos de datos públicamente accesibles capturados en condiciones realistas.

El resto del capítulo está organizado como se describe a continuación. Los conjuntos de datos utilizados y el criterio de selección se describen en la sección 5.1, seguidos de los criterios metodológicos generales en la sección 5.2. El diseño de los tres experimentos que componen este estudio se detalla en la sección 5.3. Finalmente, las técnicas que hemos mencionado más arriba y que serán utilizadas para la validación de los experimentos se exponen en la sección 5.4.

5.1. Conjuntos de datos de evaluación

En esta subsección se describen en detalle los conjuntos de datos utilizados para los experimentos de esta tesis, los criterios individuales y grupales con los cuáles estos fueron seleccionados, y los tipos de tarea de escritura que se incluyen en los mismos.

5.1.1. Criterios de selección

Para asegurar que las conclusiones aplican de la manera más general posible y no solo en condiciones restringidas, se utilizaron varios conjuntos de datos diferentes para todos los experimentos que se describen en esta sección. Los criterios de selección individual para incluir un dataset en los experimentos han sido:

- *Encontrarse disponible en forma pública y gratuita*. Muchos de los conjuntos de datos son de acceso limitado, o bajo convenios de confidencialidad, lo que reduce la posibilidad de replicación ulterior del experimento por parte de otros grupos de investigación, a la vez que dificulta la construcción de experimentos comparativos.

- *Ser lo suficientemente extenso*. La cantidad de sesiones u oraciones, o el total de teclas por usuario debe exceder el mínimo requerido para entrenar los diversos métodos que se evaluarán. La cantidad de usuarios incluidos debe ser suficiente para permitir establecer los resultados con un margen de error aceptable al utilizar criterios estadísticos.

- *Haber sido ya utilizado en diversos estudios previos sobre cadencias de tecleo*. En todas las ocasiones en las que resulto posible, se intentó construir experimentos comparativos replicando resultados anteriores y contrastándolos con los métodos propuestos. Al utilizar los mismos conjuntos de datos originales, se verifica que la implementación de la réplica sea correcta y que la comparación sea justa.

Entre los conjuntos de datos reseñados 2.4.1 existen varios que cumplen con los requisitos. Es, sin embargo, imposible utilizarlos a todos, pues la extensión de los resultados

reportados crecería con su número. Buscando balancear la diversidad y la extensión, se utilizaron los siguientes criterios para seleccionar un grupo en particular:

- *Adquisición en entornos disímiles*. Las tasas de error de un método pueden variar significativamente en función del entorno de captura de los datos [34], ya que este impone una cierta cantidad de distracciones involuntarias y otras dificultades. A los fines de evitar conclusiones demasiado optimistas o pesimistas, las evaluaciones deben por tanto realizarse con varios conjuntos de datos adquiridos en entornos disímiles.

- *Variedad de tareas de escritura representadas*. Diferentes tareas de escritura producen patrones de tecleo con variaciones suficientemente significativas como para determinar con cuál de ellas se trata [32]. Para asegurar la robustez de los métodos propuestos, es necesario evaluar al menos las descritas en 5.1.2.

- *Adquisición por parte de autores o grupos de autores que no hayan colaborado mutuamente*. Complementando ambos criterios anteriores, de esta forma se introduce mayor variedad y se evitan posibles sesgos malintencionados o inconscientes en los protocolos de adquisición.

- *Diversidad de idiomas*. Se ha demostrado que el idioma no tiene gran influencia en la precisión del análisis de cadencias de tecleo [113] y que incluso es factible utilizar entrenamiento en un idioma para verificar plantillas en otro [114]. Sin embargo, estos resultados podrían no extrapolar a los métodos propuestos. Una vez más con el objetivo de asegurar la generalizabilidad de los resultados, se incluyó más de un idioma en la evaluación.

Si bien la consideración de los rendimientos de los métodos propuesto por grupo etario, por sexo, o por otras agrupaciones no deja de plantear interrogantes de interés en la investigación, los conjuntos de datos existentes que cumplen con los criterios antedichos suelen carecer de una clasificación fina de los usuarios. Por este motivo, en esta etapa se prefirió considerar los usuarios como cajas negras, sin utilizar información ulterior sobre ellos en la clasificación más que las muestras temporales de su escritura.

5.1.2. Tareas de escritura

Cada tarea de escritura representa una combinación de procesos motrices y de decisión, cuyas características inducen cambios en los parámetros temporales resultantes [32]. Denominamos en forma genérica *texto libre* a la escritura sin restricciones específicas que refleja en términos generales la utilización laboral cotidiana de una computadora. Es esperable en este caso que sean utilizadas con regularidad todas las teclas alfanuméricas, de puntuación, y especiales.

Una tarea de *composición* consiste en la invención y transcripción de un texto relacionado con un tema específico. Esperamos encontrar un mayor predominio de

Dataset		Tarea	Usuario	Orac.	Orac./ Usuario	Teclas/ oración	Perfiles N≥20	Perfiles N≥40
LSIA		Texto libre	158	38264	281	81	5167	2777
KM		Composición	20	1522	76	72	1644	854
		Transcripción					1551	840
PROSODY	GAY	Transcripción (en contra)	400	7871	20	114	13188	8911
		Transcripción (a favor)					12909	8772
		Composición (en contra)					13697	9233
		Composición (a favor)					15362	9959
	GUN	Transcripción (en contra)	400	11537	29	119	13966	9380
		Transcripción (a favor)					13408	9148
		Composición (en contra)					13991	9374
		Composición (a favor)					16026	10380
	REVIEW	Transcripción (en contra)	500	13326	27	97	13477	9850
		Transcripción (a favor)					13638	9902
		Composición (en contra)					14890	10483
		Composición (a favor)					15937	10909

Cuadro 5.1: Principales características de los conjuntos de datos seleccionados

caracteres alfabéticos y de puntuación, y pausas más prolongadas en los intervalos en los que el usuario elabora el texto siguiente o se encuentra indeciso. Durante la composición es probable que aparezcan secuencias largas de borrado, utilizando las teclas *backspace* y *delete*, para corregir errores o reconstruir fragmentos del texto.

En la *transcripción* o *copia*, donde el usuario reproduce un texto predeterminado que se muestra en otra ventana de aplicación o en una fuente externa en papel, es esperable una frecuencia similar de teclas alfabéticas y de puntuación. Sin embargo, nos encontraremos con pausas más cortas, pues el usuario no elaborara el texto mientras escribe, y con más errores locales, pero no secuencias largas de borrado.

Otro criterio para dividir las tareas de escritura, relacionado con la influencia del estado emocional del escribiente, utiliza la opinión del mismo al respecto del texto que se compone o se transcribe. Banerjee *et al.* [37] han demostrado que es factible, tanto en tareas de composición como de transcripción, determinar si el texto representa una opinión semejante o contraria a la del usuario.

5.1.3. Conjuntos de datos seleccionados

Siguiendo los criterios individuales y grupales detallados en la sección 5.1.1, fueron seleccionados tres conjuntos de datos. De esta forma se logra un balance adecuado entre la generalizabilidad producto de la diversidad en las fuentes de datos, y la extensión y claridad en el reporte de los resultados. Menos de tres conjuntos de datos no hubieran representado adecuadamente la variedad de idiomas, entornos de adquisición, y tareas de escritura

necesarias, mientras que una mayor cantidad haría incomprensibles las figuras y tablas de resultados.

A continuación, se describen los tres conjuntos de datos seleccionados: LSIA, KM, y PROSODY. En la tabla 5.1 se detallan sus principales características, incluyendo las tareas representadas, la cantidad de usuarios, oraciones, oraciones por usuario, promedio de teclas por oración, y cantidad de perfiles de *n*-gramas con más de 20 y de 40 observaciones. Estos últimos serán empleados para el experimento sobre distribuciones subyacentes que se describe en la sección 5.3.1, mientras que las oraciones se utilizarán en los experimentos sobre síntesis de muestras artificiales e identificación del texto ingresado, descritos en las secciones 5.3.2 y 5.3.3. Dos idiomas se encuentran representados en la selección, castellano en LSIA e inglés en KM y PROSODY.

5.1.3.1. LSIA

El conjunto de datos LSIA [115] integra las distintas tareas de adquisición de los miembros del Laboratorio de Sistema de Información Avanzados de la Facultad de Ingeniería de la Universidad de Buenos Aires. Ha sido utilizado en artículos previos para evaluar las tasas de error del modelado con contextos finitos para la autenticación [33], para replicar dos conocidos experimentos de verificación de textos libres [34], y para optimizar el parámetro *p* de la distancia de Minkowski con el objetivo de minimizar la tasa de error al autenticar usuarios [35].

La versión más actualizada del conjunto de datos LSIA incluye una gran cantidad de sesiones de escritura en texto libre, capturadas con teclados convencionales, que han sido ingresadas durante el curso del trabajo diario de dos centenas de usuarios, en el idioma español. La secuencia de teclas y parámetros de tiempo que componen cada sesión se han registrado junto con la identidad del usuario. Tanto los tiempos de retención como de latencia fueron registrados con precisión de milisegundos, aunque a veces se redondearon al múltiplo más cercano de ocho milisegundos, o posiblemente otros valores, debido a limitaciones en la herramienta de adquisición. Como el software utilizado para capturar el texto se ejecuta en una página web, el navegador (tanto como la plataforma de ejecución) ha restringido la precisión con la que se pueden temporizar los eventos de teclado.

Para generar el conjunto de datos LSIA, se capturaron sesiones de texto libre de 158 usuarios, tanto hombres como mujeres, durante un intervalo de tiempo que abarcó varios años. Hasta donde sabemos, nuestro conjunto de datos es el único en el que se pueden observar los efectos de los cambios a largo plazo en la cadencia de tecleo. Los usuarios, de entre 28 y 60 años de edad, presentan habilidades mecanográficas que varían desde la de un experto bien entrenado hasta la de un lego que tipea con un único dedo. Por supuesto, su precisión (medida como porcentaje de teclas de borrado a teclas totales) también varía mucho y, sorprendentemente, no de una manera correlacionada con la velocidad de escritura. La identidad de los usuarios se verifico antes de que comenzara cada sesión utilizando una frase de contraseña y a veces un segundo factor de autenticación adicional, reduciendo la posibilidad de que hayan sido etiquetados equívocamente. Algunas sesiones de escritura incluyen sólo 50 teclas, pero otras se extienden por encima de las mil con bastante frecuencia. El promedio se encuentra alrededor de 250 teclas por sesión.

En comparación con otros conjuntos de datos, adquiridos en entornos de laboratorio calmos, silenciosos, y predecibles, LSIA fue capturado en un entorno muy exigente del mundo real. Los usuarios no contaban con una estación de trabajo fijo, por lo que no necesariamente utilizaban el mismo teclado para cada sesión, y se encontraban sujetos a interrupciones, distracciones, y el estrés diario de un entorno demandante.

5.1.3.2. KM

El conjunto de datos KM se utilizó para comparar algoritmos de detección de anomalías para la autenticación de usuarios por medio de la cadencia de tecleo [5] y se puso a disposición del público en forma libre y gratuita. Contiene sesiones diferenciadas de texto libre de veinte usuarios, tanto de composición como de transcripción. El propósito de separarlas fue evaluar si diferentes tareas de escritura producen perfiles intercambiables para su uso posterior en el entrenamiento de un clasificador. Dado que a los voluntarios de un experimento les resulta más fácil transcribir un texto que crear uno propio, y están más dispuestos a participar en una tarea de transcripción que en una de composición, el uso del texto transcripto en su lugar puede ser fructífero.

Los autores de [5] descubrieron que los tiempos de retención y latencia resultan ser de dos a tres milisegundos más lentos, en promedio, durante una sesión de transcripción en contraste con una de composición. Esta variación, aparentemente pequeña, es suficiente para ser estadísticamente significativa y permite determinar la tarea que se está realizando. A pesar de esta diferencia, agregar sesiones de transcripción a la capacitación del clasificador no cambió el rendimiento durante la evaluación.

Por este motivo, se agruparon las sesiones de composición y transcripción en un solo conjunto de datos para algunos de los experimentos de esta tesis, siempre y cuando el objetivo fuese mostrar la robustez de los métodos propuestos ante distintas tareas de escritura, y no los distintos desafíos que estas plantean.

5.1.3.3. PROSODY

El conjunto de datos PROSODY, puesto a disposición del público en forma libre y gratuita por los autores de [37], se utilizó para explorar la posibilidad de detectar señales de intención maliciosa mediante el análisis de variaciones en los patrones de escritura. Al igual que en el conjunto de datos KM, también encontramos texto compuesto y transcripto, pero las tareas fueron subdivididas una vez más con el objetivo de agregar una dimensión emocional. La temática de los textos incluidos abarca tres tópicos controvertidos (GAY, matrimonio homosexual, GUN, control de armas, y REVIEW, reseñas de restaurantes). Para su adquisición, se solicitó a un conjunto de 400 usuarios que escribieran dos ensayos cortos, uno en contra y otro a favor de la idea propuesta, y que copiaran otros dos ensayos cortos con la misma polaridad ideológica.

Idénticamente al caso del conjunto de datos KM, los autores encontraron aquí variaciones en la cadencia de tecleo que son estadísticamente significativas al considerar diferentes tareas de escritura, pero también encontraron diferencias basadas en la reacción subjetiva del usuario al contenido que se está componiendo o transcribiendo. Al igual que con el conjunto

de datos KM, para algunos experimentos fueron agrupados las sesiones de diferentes tareas de escritura.

5.2. Criterios metodológicos generales

Varios problemas comunes han proliferado en los estudios de cadencias de tecleo, y aquellos relacionados con biometría en general, hasta tal punto que han motivado cierta literatura especializada que intenta abordarlos y tratar de prevenirlos. Killourhy y Maxion [21] enfatizan la importancia de realizar experimentos comparativos en contraste con las evaluaciones de una única vez, en donde una nueva tecnología y un nuevo conjunto de datos se evalúan juntos. Señalan también la necesidad de fortalecer las conclusiones con inferencia estadística. Jain *et al.* han propuesto un conjunto de directrices [116] con el objeto de garantizar buenas prácticas en la investigación de métodos biométricos; a pesar de que están dirigidos a la evaluación de sistemas clásicos basados en caracteres fisiológicos y, por lo tanto, no se extrapolan directamente a este experimento, seguimos las reglas que son aplicables.

Mas explícitamente, para asegurarnos de que los experimentos de esta tesis proporcionan valor a la comunidad de investigadores, nos aseguramos que, de ser posible, los mismos sean comparativos, replicables, generalizables, y fundamentados en inferencia estadística.

5.2.1. Experimentos comparativos

Un experimento comparativo es aquel que plantea una pregunta de investigación o hipótesis en la cual dos o más estrategias de resolución del problema afectan algún tipo de respuesta [117]. Acotando la definición general a las condiciones de este estudio, los experimentos comparativos evaluarán un conjunto de circunstancias o estrategias fijando el resto de las condiciones de evaluación.

En todos los casos se comparará el rendimiento de los métodos propuestos sobre tres conjuntos de datos, disponibles públicamente y que han sido utilizados previamente en otros estudios sobre cadencias de tecleo; estos son descritos junto con los criterios de selección en 5.1. A su vez, para cada conjunto de datos se compararán distintas estrategias propuestas y también aquellas del estado del arte.

En particular, el experimento sobre distribuciones subyacentes incluye entre las distribuciones candidatas a ser evaluadas varios casos previamente utilizados en la literatura y vistos en la reseña de 2.4.5, como ser las distribuciones normal y lognormal. El experimento sobre síntesis de cadencias de tecleo y contramedidas de defensa incluye las estrategias de síntesis de mejor rendimiento que han sido propuestas por otros autores y que han sido reseñadas en 2.4.6: NoiseBot, GaussianBot, y LBMC. Lamentablemente, no fue factible montar un experimento comparativo para el método de identificación del texto ingresado pues este intenta llenar una laguna en la literatura del tema y, al trabajar sobre textos de mayor extensión sobre los cuales otros métodos resultan inaplicables, no puede ser evaluado en idénticas condiciones.

5.2.2. Replicabilidad

La replicabilidad es uno de los principios fundamentales del método científico. Se entiende por replicabilidad la propiedad del experimento que permite que sus resultados sean alcanzados en forma idéntica, dentro de los márgenes de error intrínsecos al proceso, por otro equipo de investigadores que siga la misma metodología y los mismos procesos. Dentro de las ciencias de la computación, se admite una definición más acotada, según la cual un experimento sería replicable si los datos de entrada se encuentran disponibles y asimismo el código o una documentación suficientemente detallada como para reconstruirlo [118].

Adoptamos ambas definiciones. En el sentido estrecho, los conjuntos de datos de entrada y de resultados listados en la sección**??**, junto con las herramientas de base que se listan en las tres secciones sobre materiales y herramientas 5.3.1.4, 5.3.2.6, y 5.3.3.4, se encuentran disponibles de forma abierta y gratuita. No se necesitan datos o herramientas adicionales para replicar nuestros resultados. En el sentido amplio, las consideraciones de la sección siguiente sobre generalizabilidad reducen la posibilidad de que la evaluación sobre otros conjuntos de datos resulte en rendimientos que difieran significativamente de los aquí reportados.

5.2.3. Generalizabilidad

La generalizabilidad, o validez externa, refiere a la posibilidad de aplicar las conclusiones de un estudio fuera de las condiciones específicas de evaluación, y suele referirse a la posibilidad de aplicarlas a las condiciones del mundo real [119]. La importancia de este criterio metodológico es ilustrada por [34], del autor de esta tesis, donde un experimento clásico sobre identificación de usuarios por medio de cadencias de tecleo elevo las tasas de error en casi seis veces, hasta el 30%, al ser evaluado bajo condiciones más arduas que las originales, pero consistentes con un ambiente de trabajo productivo.

Los datos crudos que se han utilizado en los experimentos de esta tesis provienen de conjuntos de datos utilizados en diferentes estudios, que fueron capturados en diferentes circunstancias y entornos, con diferentes usuarios, por diferentes autores, y con el objetivo de resolver distintos interrogantes. Como se detalla en 5.1.3, los conjuntos de datos son extensos, incluyen más de un idioma, y al menos el nuestro (LSIA) corresponde a datos reales capturados durante el transcurso de la labor cotidiana de los usuarios. Se han realizado los esfuerzos necesarios para garantizar, en la medida de lo posible, que los conjuntos de evaluación seleccionados (ver 5.1.1) representen una variedad suficiente de condiciones de evaluación realistas y no un caso particularmente favorable o específico. En la sección 2.4.4 se han reseñado otras consideraciones adicionales que pueden afectar el rendimiento de los métodos, y la selección de conjuntos de datos y condiciones de evaluación se ha realizado teniéndolas en cuenta.

5.2.4. Inferencia estadística

Killourhy [40] ha mostrado que las tasas de error de los métodos biométricos no deben ser considerados valores puntuales sino variables aleatorias. Siguiendo a dicho autor, en todos

aquellos casos en los que resultara posible, las conclusiones se han fundamentado en inferencia estadística. En particular, los resultados numéricos se expresan con intervalos de

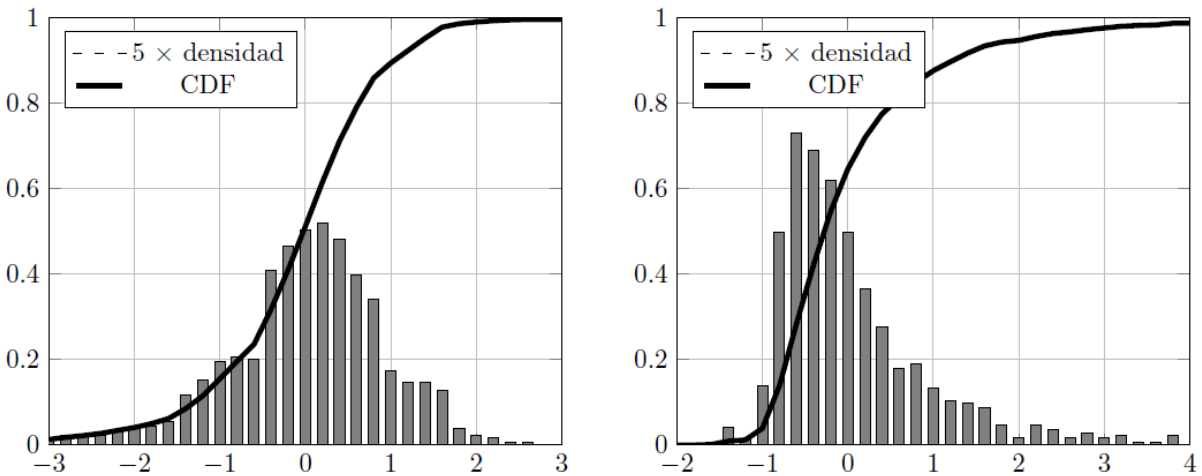

Figura 5.1: Ejemplo de histogramas típicos para tiempos de retención y latencia

estadística de Anderson-Darling y el criterio de información de Akaike, son utilizadas para mostrar significatividad y descartar artefactos muestrales.

5.3. Diseño de los experimentos

5.3.1. Experimento sobre distribuciones subyacentes

El experimento sobre distribuciones subyacentes tiene como fin determinar, dentro de un conjunto de candidatas, cual o cuáles distribuciones cumplen mejor la tarea de ajuste de los histogramas empíricos de atributos temporales. El planteo detallado del problema se realiza en la sección 5.3.1.1. Las distribuciones candidatas que han sido consideradas para la tarea y el criterio de selección se describen en la sección 5.3.1.2. El procedimiento de evaluación experimental se explica en la sección 5.3.1.3. Los materiales y herramientas utilizados para el experimento se enumeran en la sección 5.3.1.4. Finalmente, la disponibilidad y contenido de los conjuntos de datos de evaluación se indica en la sección 5.3.1.5.

El criterio de información de Akaike es una herramienta teórica que se utiliza en este experimento para determinar el contenido de información residual luego de ajustar los histogramas empíricos con las distribuciones candidatas. Como no es de uso tan común como la máxima verosimilitud y los tests de hipótesis, no se presupone que el lector lo conozca previamente y se explica en la sección 5.4.3.

5.3.1.1. Planteo del problema

El objetivo principal de este experimento es comparar varias distribuciones candidatas, intentando clasificarlas de acuerdo con su mérito para ajustar histogramas empíricos de atributos temporales de digramas en perfiles intrausuario. El problema es diferente y más

complejo que simplemente ajustar un único conjunto de observaciones a una distribución, porque se debe modelar el histograma para cada tecla alfanumérica de un teclado estándar. Por lo tanto, estamos solicitando una distribución que proporcione el mejor ajuste con más frecuencia, o que se ajuste mejor en promedio, o sea rechazada con menos frecuencia, en la colección de todos los histogramas de atributos temporales de todos los perfiles.

Además, el mérito no debe abordarse solo como la minimización de una cierta medida de la diferencia entre el modelo y los datos empíricos. Varias consideraciones adicionales que no tienen que ver con el problema puramente estadístico del ajuste son relevantes cuando los modelos están destinados a ser utilizados en sistemas biométricos del mundo real. Para empezar, el número de parámetros en las distribuciones candidatas debe ser lo más pequeño posible. No solo para evitar las molestias del sobreajuste, que obstaculiza la generalización, sino también porque la estimación de parámetros adicionales requiere más observaciones para ser confiable. Por lo tanto, al elegir como modelos ciertas distribuciones con muchos parámetros incrementamos la cantidad de entrenamiento requerido para lograr una tasa de error baja. Esta es una consideración práctica para la verificación continua de la cadencia de tecleo y para los ataques de suplantación de identidad como [28], que se vuelven más eficaces cuando se necesitan menos pulsaciones de teclas para asegurar que el perfil es confiable.

Idealmente, un modelo estadístico debería proporcionar a la vez poder predictivo y explicativo [120]. El primero refiere aquí a la bondad de ajuste de la distribución candidata respecto de los datos empíricos. El segundo, aunque a primera vista no es relevante para una implementación práctica de la verificación biométrica, puede encontrar aplicaciones en el campo extendido del análisis de la cadencia de tecleo. Después de todo, el histograma de tiempo observado es el resultado de procesos neuronales y motores que, si se modelan correctamente, pueden arrojar luz sobre el estado físico y emocional del usuario. De la misma forma en que el tiempo medio de vuelo está altamente correlacionado con la habilidad de escritura del usuario, y su variación puede ayudar a clasificar el estado emocional del usuario [121] o detectar deterioro cognitivo [122], los parámetros de las distribuciones que mejor ajustan deberían ofrecer algún tipo de información sobre los procesos subyacentes.

5.3.1.2. Distribuciones candidatas

Los tiempos de retención y los tiempos de latencia muestran diferentes formas de histograma; los primeros son bastante similares a una variable normal mientras que los segundo presentan colas más largas y sesgo positivo. La figura 5.1 muestra la densidad de probabilidad y la distribución acumulada de ambos tiempos, para la tecla espacio del usuario *s019* en el conjunto de datos KM; las observaciones se han agrupado en intervalos de 0,2 desvíos estándar alrededor de la media, para ejemplificar estas afirmaciones. Una inspección visual de los histogramas de diferentes usuarios y conjuntos de datos es suficiente para convencerse de que este comportamiento es típico.

Como candidatas para la evaluación, se eligieron de [123] siete distribuciones bien conocidas con dos parámetros y siete más con tres parámetros. Los requisitos fueron tres: que tuvieran sesgo positivo, cola larga, y soporte infinito, al menos en los números reales positivos. Como criterio subjetivo adicional, se buscó que tuvieran una semejanza general de la

envolvente con los perfiles observados. Se intento representar diferentes familias de distribuciones tanto como fuera posible, prefiriendo aquellas

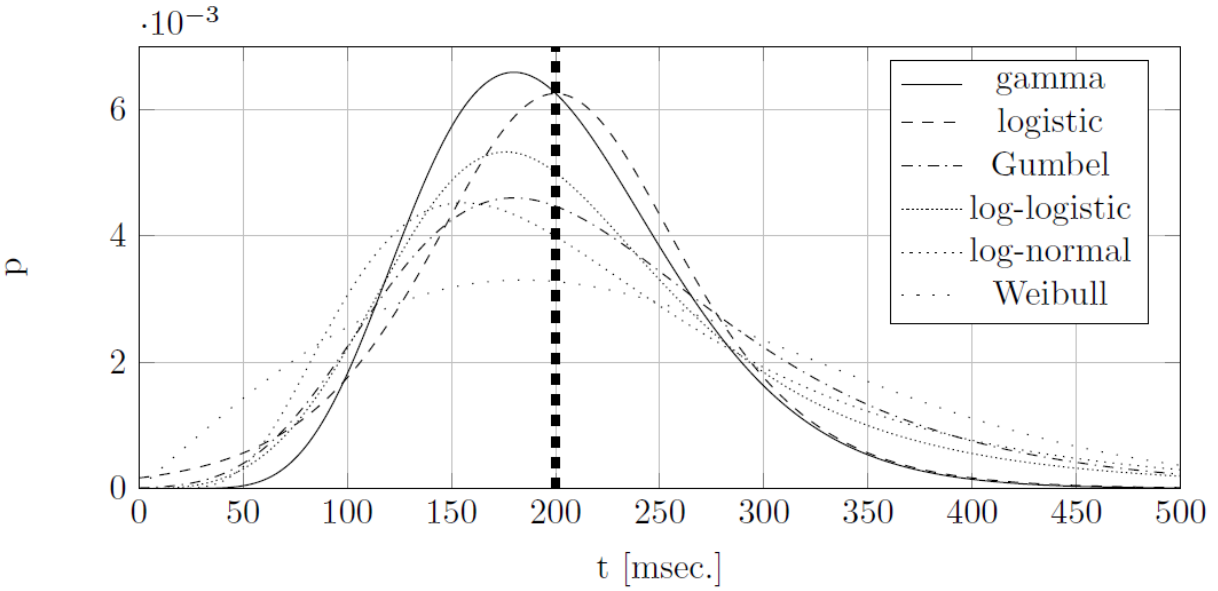

Figura 5.2: Distribuciones candidatas con dos parámetros

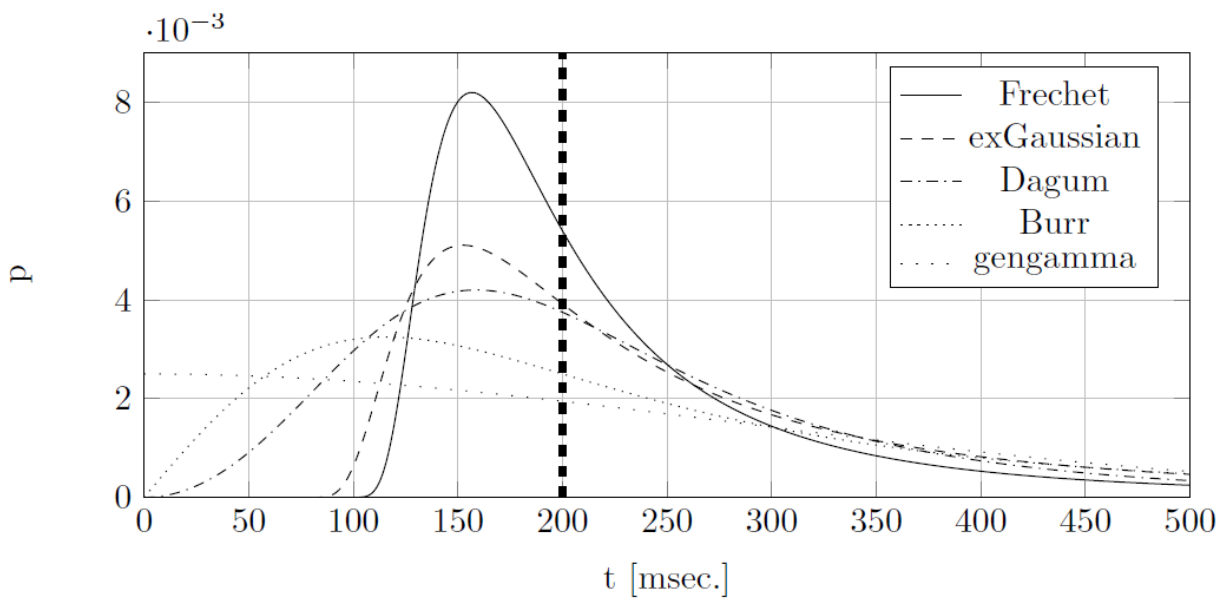

Figura 5.3: Distribuciones candidatas con tres parámetros

con parámetros de forma que controlaran la asimetría, para tener en cuenta tanto los tiempos de retención como los de latencia. Finalmente, las consideraciones prácticas de la implementación nos obligaron a elegir aquellas que tuvieran una implementación en R, compatible con el paquete fitdistrplus [124].

Las distribuciones candidatas con dos parámetros que fueron elegidas resultaron ser lognormal (lnorm), logística (log), log-logística (llog), gamma, Weibull y Gumbel; también se incluyó una distribución normal como caso base para la comparación. Los términos entre paréntesis muestran las abreviaturas que se utilizarán para las tablas de resultados, que coinciden con la función R correspondiente. La figura 5.2 ejemplifica las primeros seis, que han sido ajustadas para tener su media en 200 ms. y una varianza consistente con ejemplos empíricos de histogramas de atributos de tiempo como los disponibles en el conjunto de datos de resultados.

Las distribuciones candidatas de tres parámetros que fueron elegidas resultaron ser exgaussiana (exGAUS), log-normal desplazada (lnorm3), log-logística desplazada (llog3), Burr, Frechet, gamma generalizada (gg), y Dagum. Una vez más, los términos entre paréntesis muestran las abreviaturas que se utilizarán para las tablas de resultados, que coinciden con la función R correspondiente. La figura 5.3 ejemplifica cinco de ellas, en las mismas condiciones que en la anterior. No se muestran la log-normal ni la log-logística desplazada, ya que tienen la misma forma que sus homólogos de dos parámetros, pero incluyen un parámetro adicional que puede desplazarlas hacia la izquierda o hacia la derecha.

5.3.1.3. Procedimiento de evaluación experimental

Cada conjunto de datos considerado contiene varias sesiones de escrituras para cada usuario. Cada una de ellas consta de una secuencia de teclas junto con sus tiempos de retención latencia, además de otra información relevante que no fue utilizada en este experimento. Se conservo la separación de las sesiones por tarea para determinar si diferentes tareas influyen en el rendimiento de las distribuciones candidatas. Las observaciones de atributos temporales se agruparon por usuario, empaquetándolas todas juntas en independientemente de la sesión de origen. Así, se construyó un perfil para cada conjunto de datos, tarea, usuario, tecla, y atributo, que consta de un conjunto de observaciones temporales. No se consideró la evolución de las cadencias individuales, sino que se formó un único histograma para cada grupo de los anteriores.

Luego, todas las distribuciones candidatas se evaluaron frente a cada perfil alfanumérico con suficientes observaciones (20 para dos parámetros y 40 para tres), truncándolos a 100 muestras como máximo por consideraciones de rendimiento. Los parámetros de la distribución candidatas se estimaron utilizando máxima verosimilitud y la probabilidad logarítmica resultante se corrigió utilizando tanto los términos adicionales del criterio de información de Akaike [125] para el recuento de parámetros como el sesgo de muestra pequeña, de ser necesario. Luego se aplicó una verificación de hipótesis utilizando la prueba de bondad de ajuste de Anderson-Darling. Esta última se adapta mejor a la tarea en cuestión porque es más sensible a las colas que a los picos de las distribuciones, por ejemplo, en comparación con la prueba de Kolmogoroff-Smirnoff. Una ventaja adicional es que la implementación disponible en R compensa las repeticiones en el conjunto de muestras, un artefacto engorroso que resulta al registrar los tiempos de pulsación de teclas con un reloj discreto.

5.3.1.4. Materiales y herramientas

La herramienta de software estadístico R [126] se utilizó para la mayoría de los cálculos complejos de este experimento. El paquete fitdistrplus [124] proporcionó la funcionalidad principal para el ajuste y ADGofTest [127] la implementación de la prueba de bondad de ajuste de Anderson-Darling, mientras que los paquetes actuar
[128], brms [129], distr [130], FAdist [131], gamlss.dist [132] y qualityTools [133] proporcionaron las distribuciones candidatas. Los módulos adicionales para el análisis de archivos de conjuntos de datos, la creación de tablas, y el encolado general de módulos se implementaron en C#.

5.3.1.5. Disponibilidad de los conjuntos de datos

El conjunto de datos de evaluación contiene archivos CSV con la lista de atributos temporales (tiempos de retención y latencia) para cada tecla en los tres conjuntos de datos de origen. Estos han sido agrupados por conjunto de datos, usuario, tarea de escritura, código de tecla, y atributo.

Se encuentra disponible en forma abierta, publica, y gratuita en sendos repositorios de IEEE DataPort [134] y Mendeley Data [135].

5.3.2. Experimento sobre síntesis de muestras artificiales y contramedidas de defensa

El experimento sobre síntesis de muestras artificiales tiene como fin evaluar el rendimiento de las estrategias basadas en histogramas empíricos y del esquema de defensa propuesto. El planteo detallado del problema se realiza en la sección 5.3.2.1. El preprocesamiento y la limpieza de los datos se describe en la sección 5.3.2.2. La lista de atributos derivados que serán utilizados para entrenar los clasificadores y evaluar el método se enumera en la sección 5.3.2.3. Los procedimientos de entrenamiento y evaluación se describen en las secciones 5.3.2.4 y 5.3.2.5. Los materiales y herramientas utilizados para el experimento se enumeran en la sección 5.3.2.6. Finalmente, la disponibilidad y contenido de los conjuntos de datos de evaluación y resultados se indica en la sección 5.3.2.7.

5.3.2.1. Planteo del problema

El objetivo principal de este experimento es doble: determinar la eficacia de las estrategias de síntesis propuestas para la generación de muestras artificiales que puedan ser utilizadas en un ataque de presentación, y determinar la eficacia de las distancias basadas en histogramas empíricos y el método de defensa contra falsificaciones sintéticas que ha sido propuesto para detectar las anteriores.

Para tal fin se enfrentarán las estrategias contra el sistema de defensa, incluyendo aquellas que representen intentos anteriores de otros autores a los fines de montar

Tipo	Atributo
Histogramas empíricos	CDF con p = 0,4 (símil Minkowski) CDF con p = 1 (símil Manhattan) CDF con p = 2 (Símil euclídea) CDF/Canberra
Distancias	Manhattan Euclídea Camberra Minkowski (p = 0,4)
Grado de desorden	R Índice de direccionalidad
Valores atípicos	Z-score

Cuadro 5.2: Atributos derivados que se utilizan en el experimento de síntesis de muestras artificiales

un experimento comparativo. Finalmente, se utilizará un método de selección de atributos para demostrar que el uso de las distancias basadas en histogramas empíricos explica el rendimiento del método de defensa propuesto.

5.3.2.2. Preprocesamiento y limpieza de los datos

Las muestras de cada conjunto de datos de entrada se dividieron en los límites entre oraciones, utilizando el punto y seguido o el punto y aparte. Se eliminaron todos los valores de tiempos de retención y latencias superiores a 1500 mseg., pues las pausas y las vacilaciones no son representativas de la cadencia de escritura natural. Se consideraron vectores de tiempos para estos dos atributos temporales, y ningún otro.

5.3.2.3. Atributos derivados

Uno de los objetivos de este experimento consiste en evaluar el rendimiento de las distancias basadas en las distribuciones empíricas que se han desarrollado en 4.2. Sin embargo, estas pueden no ser suficientes para la tarea en cuestión y deben complementarse con otros atributos derivadas. Aunque la lista de aquellos utilizados en estudios previos de la literatura sobre cadencia de tecleo es extensa [18], estos pueden agruparse en tres grandes categorías: atributos basados en distancias, medidas del grado de desorden, y umbrales de valores atípicos.

Del primer grupo seleccionamos las distancias normalizadas y escaladas de Manhattan, euclídea, de Minkowski con p = 0,4 [35], y de Canberra; el apéndice A brinda las definiciones de todas las anteriores. Del segundo grupo seleccionamos la métrica R [8] y el índice de direccionalidad, que han sido descritas en A.8 y A.9. Se incluyo también en representación del tercer grupo al conteo de valores atípicos, que se describe en A.7. El cuadro 5.2 lista los atributos elegidos.

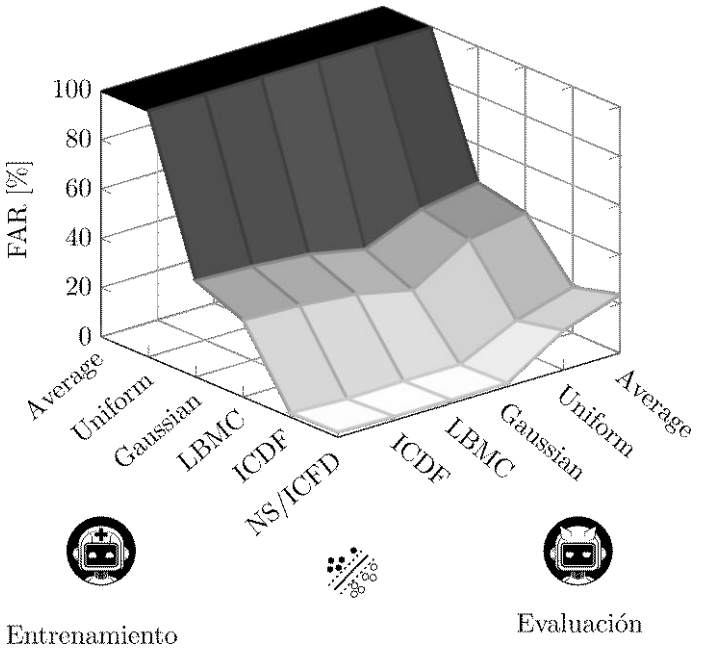

Figura 5.4: Tasas de falsos positivos para todas las combinaciones de estrategias de robot bueno y robot malo, utilizando entrenamiento intrausuario

Esta selección de atributos derivados intenta representar a todos los candidatos que han sido extensamente evaluados y que brindan el mejor desempeño en cada categoría. Junto con cuatro distancias basadas en histogramas empíricos (para p = 0,4, 1, 2 y símil Canberra), comprenden once atributos que se derivan de los tiempos de retención latencias, totalizando 22 atributos derivados para cada instancia de entrenamiento.

Observemos aquí una peculiaridad del problema tratado en este experimento, que simplifica la comparación de las muestras, ya sean estas legítimas o falsificadas, del usuario humano o sintetizadas por robot bueno y robot malo, durante el entrenamiento o durante la evaluación. Si bien el problema general de la verificación de la cadencia de tecleo en texto libre padece la dificultad de tener que comparar muestras con diferentes secuencias de teclas, demandando la extracción y el promediado de *n*-gramas y mucho ingenio, en el caso que tratamos ahora todas las comparaciones se simplifican. Como se ha mostrado en las figuras 4.3 y 4.4 del capítulo 4, las muestras y las falsificaciones sintéticas siempre comparten la misma secuencia exacta de teclas.

Ergo, el cálculo de cada atributo derivado se reduce al caso de texto fijo. Además, la métrica R se ve potenciada ya que no se necesita considerar solo los digramas compartidos entre las muestras comparadas; todos los digramas son compartidos.

5.3.2.4. Entrenamiento

La configuración general del entrenamiento se muestra en la figura 4.3 de la sección 4.4, en donde se ha descrito el proceso de defensa contra falsificaciones sintéticas. Todas las estrategias de síntesis de muestras se pueden probar contra el sistema una vez implementadas, pero en el momento del entrenamiento debemos decidir cuáles serán utilizadas por robot bueno y robot malo. La figura 5.4 muestra el rendimiento

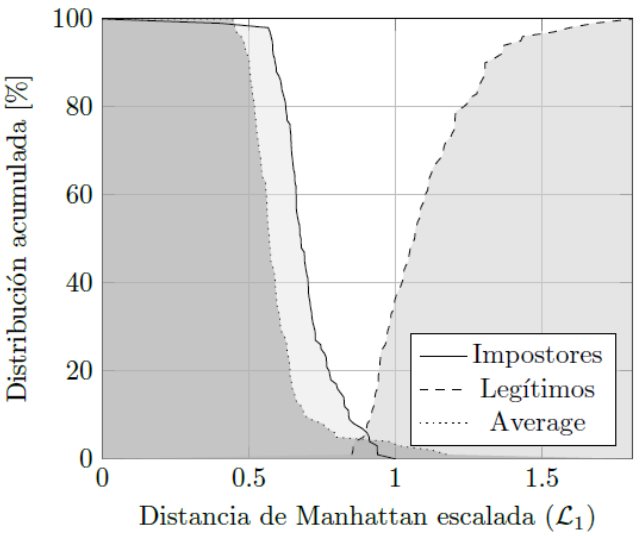

Figura 5.5: Distribuciones acumuladas de valores de la distancia de Manhattan escalada y normalizada, para un usuario legítimo, impostores no entrenados, y la estrategia Average con entrenamiento interusuario

de cada par de estrategias al ser enfrentadas entre sí. Como el objetivo de las estrategias de síntesis es lograr falsos positivos, la tasa de estos mide su rendimiento; valores más altos implican una mayor probabilidad de engañar al sistema. Inversamente, valores más bajos implican una mejor defensa, y es esperable que las estrategias más sofisticadas defiendan mejor contra las falsificaciones sintéticas de todo tipo. Por este motivo sorprende a primera vista que las estrategias triviales de síntesis como Average y Uniform logren vencer a todas las demás en manera consistente. Este fenómeno demanda una explicación.

El clasificador emplea un vector de atributos derivados, descritos en la sección anterior, para decidir si la muestra pertenece al usuario legítimo o es una falsificación artificial. La mayoría de estos corresponden a distancias y se espera que alcancen valores más altos cuanto más difiera la muestra del comportamiento de escritura promedio del usuario. El clasificador utiliza este patrón como una forma de discriminar entre ambos. Sin embargo, muchos usuarios tienen un ritmo de escritura característico que está muy cerca de la media de la población y una falsificación construida con Average, y en menor grado con Uniform, presenta una suerte de versión idealizada de la cadencia de tecleo del usuario legítimo. Las distancias basadas en histogramas empíricos que han sido propuestas en la sección 4.2 son capaces de detectar diferencias sutiles en el ruido, pero aquí no hay suficiente ruido para detectar ninguna diferencia.

Es fácil remediar este escollo. Sólo necesitamos permitir a robot bueno emplear más de una estrategia para generar instancias de entrenamiento de impostores. De esta forma, el clasificador sabrá identificar las falsificaciones Promedio y Uniforme, que son demasiado buenas para ser verdad. Para los experimentos, optamos por agregar un 20% de instancias de entrenamiento generadas con Average, un 20 % con Uniform, y el 60 % restante con ICDF, que la figura 5.4 muestra como de mejor rendimiento. Los porcentajes han sido resumidos en el cuadro 5.3.

La figura 5.5 ejemplifica el problema que estamos tratando de resolver, mostrando

Estrategia	Porcentaje
Averange	20%
Uniform	20%
ICDF	60%

Cuadro 5.3: Porcentajes de instancias de entrenamiento para cada estrategia de síntesis

las distribuciones acumulativas de valores de la distancia de Manhattan escalada y normalizada para un ejemplo de usuario legítimo, impostores sin entrenar, y la estrategia Average con entrenamiento intrausuario. Obsérvese que los valores que entrega esta última son demasiado bajos para ser auténticos; ni el usuario legítimo presenta un comportamiento tan puro. Necesitamos, sin embargo, presentar ejemplos explícitos al clasificador para que aprenda que esto es así.

5.3.2.5. Evaluación

El sistema de detección de falsificaciones sintéticas que ha sido propuesto en la sección 4.4 se evaluó contra a la familia de estrategias descrita en la sección 4.3. Se realizaron dos experimentos idénticos, con robot malo utilizando perfiles interusuario de la población general y el perfil intrausuario del usuario objetivo. Con el fin de confirmar la relevancia de las distancias basadas en histogramas empíricos para detectar falsificaciones sintéticas, se llevó a cabo una segunda etapa de selección de atributos basada en correlación (*correlation-based atribute selection* [136]) luego de los experimentos. Este método, ampliamente utilizado y estudiado en la literatura de aprendizaje automático, intenta encontrar un subconjunto de características que estén fuertemente correlacionadas con el resultado de la clasificación, pero no correlacionadas entre sí.

5.3.2.6. Materiales y herramientas

El preprocesamiento y limpieza de los datos, y el cálculo de los atributos derivados se realizó con la herramienta para el análisis de cadencias de tecleo desarrollada para esta tesis. La clasificación se realizó con la implementación SVM de WEKA´ 3.8.4 [137], utilizando optimización secuencial mínima (SMO) [138] para el entrenamiento, un núcleo polinomial y un calibrador logístico, todos ellos provistos en forma integrada por la misma herramienta de clasificación WEKA.

5.3.2.7. Disponibilidad de los conjuntos de datos

En el conjunto de datos de evaluación, se proporcionan el texto y los tiempos de retención y latencias de las oraciones en las que ha sido dividido. Adicionalmente, para cada oración se incluyen dos archivos ARFF con los conjuntos de entrenamiento y de prueba que contienen los respectivos valores de los atributos derivados para cada instancia.

En el conjunto de datos de resultados, se resume en un archivo en formato CSV para cada usuario los resultados para cada combinación de estrategias de robot bueno y robot malo.

Ambos se encuentran disponibles en forma abierta, publica, y gratuita en sendos repositorios de IEEE DataPort [139] y Mendeley Data [140].

5.3.3. Experimento sobre identificación del texto ingresado utilizando atributos temporales

El experimento sobre identificación del texto ingresado tiene como fin evaluar el rendimiento del método propuesto para tal fin. El planteo detallado del problema se realiza en la sección 5.3.3.1. El preprocesamiento y la limpieza de los datos se describe en la sección 5.3.3.2. El procedimiento de evaluación se describe en la sección 5.3.3.3. Los materiales y herramientas utilizados para el experimento se enumeran en la sección 5.3.3.4. Finalmente, la disponibilidad y contenido de los conjuntos de datos de evaluación y resultados se indica en la sección 5.3.3.5. La descripción de este experimento es más somera pues comparte muchos elementos, como los atributos derivados, con el anterior.

5.3.3.1. Planteo del problema

Hemos concluido a lo largo de la sección 2.6 que existe una laguna en la literatura de utilización de atributos temporales extraídos durante un ataque de canal lateral para la identificación del texto ingresado: los métodos existentes se limitan a texto cortos y listas de candidatos pequeñas. El método propuesto para tal fin en la sección 4.5, que es un método derivado de aquel para detección de falsificaciones sintéticas de la sección 4.4 y adaptado a este propósito, promete atacar el problema eficientemente´ con textos y listas de candidatos más extensos.

El objetivo principal de este experimento consiste en evaluar el rendimiento del método propuesto para la identificación del texto ingresado, utilizando sólo atributos temporales.

5.3.3.2. Preprocesamiento y limpieza de los datos

Las muestras de cada conjunto de datos de entrada se dividieron en los límites entre oraciones, utilizando el punto y seguido o el punto y aparte. Se eliminaron todos los valores de tiempos de retención y latencias superiores a 1500 mseg., pues las pausas y las vacilaciones no son representativas de la cadencia de escritura natural. Se consideraron vectores de tiempos para estos dos atributos temporales, y ningún otro.

5.3.3.3. Evaluación

El método evaluado ha sido descrito en la sección 4.5. Los datos de entrenamiento utilizados para reconstruir los textos candidatos con modelado por contextos finitos consistieron en, para cada muestra en consideración, todo el resto de las muestras en el perfil del usuario. Se utilizaron solamente los vectores de tiempos de retención y latencia, excluyendo todos los datos adicionales que se hubieran registrado en las muestras.

Por obvios motivos, al procesar cada oración el texto y los atributos temporales de la misma fueron excluidos del entrenamiento. Para la evaluación intrausuario, este último consistió en todo el resto de las oraciones disponibles en el perfil del usuario, etiquetadas como legítimas, y una cantidad idéntica de oraciones, pero con los valores temporales reemplazados por aquellos de otro fragmento de texto tomado al azar del perfil del usuario, etiquetadas como impostores. Para la evaluación interusuario, se emplearon cien oraciones elegidas al azar entre todos los usuarios manteniendo sus correspondientes atributos temporales, etiquetadas como legítimas, y otras cien oraciones elegidas al azar entre todos los usuarios, pero con los valores temporales reemplazados por aquellos de otro fragmento de texto, etiquetadas como impostores.

Aunque la inclusión de tiempos de retención mejora la precisión del método, solo las latencias se consideraron en este experimento, ya que los últimos siempre están disponibles después de ataques de canal lateral exitosos, pero los primeros rara vez lo están.

5.3.3.4. Materiales y herramientas

Los materiales y herramientas utilizados en este experimento son los mismos que en el anterior, detallados ya en la sección 5.3.2.6.

5.3.3.5. Disponibilidad de los conjuntos de datos

En el conjunto de datos de entrenamiento, se provee para cada oración extraída del perfil de cada usuario, tanto el texto como los tiempos de retención y latencia. Adicionalmente se proporcionan los archivos ARFF con los valores de atributos derivados que fueron utilizados para entrenar cada modelo.

El conjunto de datos de resultados incluye un archivo CSV de resumen por usuario con una lista de oraciones evaluadas y sus longitudes, las tasas de falsos positivos, y una etiqueta de falso negativo en caso de que el texto original no sea identificada como correspondiente a sus atributos temporales.

Ambos se encuentran disponibles en forma abierta, publica, y gratuita en sendos repositorios de IEEE DataPort [141] y Mendeley Data [142].

5.4. Validación de los experimentos

Un buen diseño del experimento requiere también un conjunto de técnicas de validación que permitan asegurar que los resultados obtenidos son válidos, y que el éxito puede ser causalmente atribuido a los métodos propuestos y no a otros factores externos o errores en el montaje del experimento. Como ejemplo extremo, hipotético, podemos mencionar un clasificador trivial evaluado sobre un conjunto de datos con una única clase; su tasa de error será del 0%, pero difícilmente podemos jactarnos de logro alguno en este caso. Las técnicas de validación de los resultados de los experimentos que serán utilizadas se describen en las

siguientes secciones, mientras que aquí planteamos los interrogantes de validación y mencionamos las técnicas utilizadas para responderlos.

En los tres experimentos planteados, es necesario validar si *estos resultados sólo aplican a estos conjuntos de datos o son generalizables a otros entornos o situaciones*. Para contestar esta pregunta, utilizamos una metodología de caso/control prospectivo, que se describe en la sección 5.4.1 para los conjuntos de datos seleccionados.

El experimento sobre distribuciones subyacentes requiere uno o más pruebas estadísticas de hipótesis para *determinar si la distribución candidata realmente ajusta a la muestra empírica de observaciones temporales*. Para tal fin emplearemos inferencia estadística en la forma de la prueba de Anderson-Darling, que se describe en la sección 5.4.2, y el criterio de información de Akaike, explicado en la sección 5.4.3.

Tanto el experimento de síntesis de muestras artificiales y contramedidas de defensa como la identificación del texto ingresado utilizando atributos temporales emplearán la técnica de evaluación comparativa con referencias (*benchmarking*), que se describe en la sección 5.4.4, para determinar si *el rendimiento de estos métodos es superior a sus antecesores en la literatura bajo similares condiciones de evaluación*.

Finalmente, estableceremos si *las distancias basadas en histogramas empíricos que han sido propuestas en esta tesis explican el mejor rendimiento de los métodos anteriores* utilizando el método de selección de atributos basado en correlación (*correlation-based atribute selection* [136]), que se describe en la sección 5.4.5.

5.4.1. Conjuntos de datos de evaluación y control

El diseño de experimentos dentro del tópico de la seguridad informática ha recibido fuertes críticas en el pasado. Aunque en ocasiones estas se han referido a la metodología experimental, la utilidad de muchos conjuntos de datos y la generalizabilidad de los resultados obtenidos a entornos o situaciones que no están representados en ellos es lo primero que debe revisarse [143]. Se han detallado en la sección 5.1 tanto los conjuntos de datos utilizados para los experimentos como los criterios de selección, y hemos tratado estas consideraciones en la sección 5.2.3.

Nos interesa tratar ahora la utilización de los conjuntos de datos para montar experimentos del tipo caso/control. Se denomina de esta forma a los estudios observacionales en el cual se identifican dos grupos para los cuáles los resultados experimentales difieren, y que se comparan en busca de atributos causales que expliquen las discrepancias. Por ejemplo, en [144] se utiliza esta metodología para comparar la gravedad de las vulnerabilidades y ataques contra distintos sistemas de seguridad.

En particular, los estudios prospectivos buscan detectar ciertos resultados con el objetivo de relacionarlos con los factores causales que se sospechan, utilizando grupos de evaluación y de control tomados de la misma población, y seleccionados en forma no relacionada con el resultado esperado [145]. Es importante que el resultado esperado sea idéntico en ambos grupos. Los estudios prospectivos presentan menores sesgos y confusión que los estudios retrospectivos.

5.4.2. Prueba estadística de Anderson-Darling

La prueba de Anderson-Darling es una prueba no paramétrica que intenta determinar si los datos de una muestra han sido originados por un proceso que obedece a una cierta distribución dada, que fue propuesta por Theodore Wilbur Anderson y Donald A. Darling en 1953 [146]. Los parámetros de la distribución deben ser estimados por otros medios.

La prueba de Anderson-Darling pertenece a la familia de criterios cuadráticos basados en distribuciones empíricas. El objetivo es medir una distancia entre la distribución empírica y la distribución candidata utilizando la expresión

$$A^2 = n \int_{-\infty}^{\infty} \frac{\big(F_n(x) - F(x)\big)^2}{F(x)\,\big(1 - F(X)\big)} dF(x) \tag{5.1}$$

en donde $F(x)$ es la función de probabilidad acumulada de la distribución candidata, $F_n(x)$ es la función de probabilidad acumulada empírica de la muestra, y n es la cantidad de observaciones en la muestra. El término

$$\big(F_n(x) - F(x)\big)^2 \tag{5.2}$$

del numerador no es más que la distancia cuadrática entre ambas distribuciones mientras que el término

$$F\,(x)\big(1 - \,F\,(x)\big) \tag{5.3}$$

del denominador corresponde a una ponderación que, en contraste con otras, asigna más relevancia a las colas de la distribución. Por ejemplo, el criterio de Cramérvon Mises, que utiliza la expresión

$$A^2 = n \int_{-\infty}^{\infty} \big(F_n(x) - F(x)\big)^2 \, dF(x) \tag{5.4}$$

emplea idéntica distancia cuadrática en el numerador, pero un denominador constante e igual a uno, ponderando por igual todos los intervalos.

Suponiendo que los datos de la muestra analizada han sido originados por un proceso que obedece a una cierta distribución dada, los puntos en la diferencia entre la distribución acumulada empírica y la ideal deberían estar distribuidos uniformemente. Suponiendo que los valores de la muestra son $t_1 \ldots t_n$, la fórmula para el test de hipótesis es

$$A^2 = -\,n - \sum_{i=1}^{n} \frac{2i-1}{n} \left[\log F\,(t_i) + log\big(1 - F(t_i)\big)\right] \tag{5.5}$$

A continuación, este estimador puede compararse contra los valores críticos de la distribución hipotetizada para determinar si el ajuste es verosímil o se rechaza la hipótesis.

Contrariamente a otras pruebas estadísticas, al utilizar Anderson-Darling se deben utilizar los valores críticos para cada familia particular de distribuciones candidatas, que deben estar disponibles con anticipación. Afortunadamente para nosotros, estos valores críticos ya han sido calculados y se encuentran disponibles en las implementaciones en R, enumeradas en la sección 5.3.1.4, de las distribuciones que se han utilizado a lo largo de esta tesis.

5.4.3. El criterio de información de Akaike

La distribución que mejor ajusta la mayor parte del tiempo no necesariamente es la que mejor ajusta en promedio. Como estos casos deben distinguirse, para determinar cuantitativamente este último necesitamos una medida de la calidad del ajuste y no solo una respuesta binaria (da el mejor ajuste o no) para cada perfil y distribución. Con este objetivo en mente, comenzamos con la valoración de Akaike [125] para un modelo.

$$AIC_C = 2k - 2\,ln(\hat{L}) \tag{5.6}$$

donde k es la cantidad de parámetros de la distribución candidata (que en este experimento puede ser dos o tres) y $\hat{L}$ es el valor máximo de la función de verosimilitud para el modelo. Del conjunto de candidatos, se debe preferir la distribución que produzca el valor mas bajo de AICc. El objetivo del término $2k$ es penalizar la introducción de más parámetros de los necesarios.

Akaike explica en [125] que el promedio del logaritmo de la verosimilitud es un estimador (con el signo invertido) de la entropía cruzada entre la distribución evaluada f con el conjunto de parámetros ϑ y la "verdadera" distribución subyacente g. Así, la primera tiende a la segunda con probabilidad uno a medida que N, el número de muestras, se incrementa indefinidamente.

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \log f\,(x_i|\vartheta) = \int g(x)\,\log f\,(x|\vartheta)\,dx \tag{5.7}$$

para casi todas las g, f, y conjuntos de puntos. Usando AICc, se puede construir un mejor estimador para propósitos prácticos gracias al término adicional $2k$ que corrige el sesgo introducido por el número de parámetros estimados y otros términos que pueden dar cuenta de pequeños tamaños de muestra [147]. Sin embargo, debido a la constante multiplicativa de AICc, debemos dividirla por dos para recuperar la entropía cruzada. Formalmente

$$\lim_{N\to\infty} \frac{1}{2\,N}\, AIC_C = -\int g(x)\,\log f\,(x|\theta)\,dx \tag{5.8}$$

$$\lim_{N\to\infty} \frac{1}{2\,N}\, AIC_C = \mathbb{E}_g[-\log f_\theta] \tag{5.9}$$

Este es el valor cuyo promedio para cada conjunto de datos y tarea de escritura se reporta en el segundo conjunto de tablas del apéndice C. Puede ser interpretado como el promedio de la entropía cruzada entre la distribución candidata y la distribución estadística verdadera generada por el proceso de escritura. Por lo tanto, cuanto menor es este valor mejor se ajusta la distribución candidata a la fuente legítima.

Siguiendo la interpretación que hace la teoría de la información, al ponderar estos números con la frecuencia de la tecla se estaría representando el contenido de información promedio en nats del atributo temporal bajo cada hipótesis o distribución candidata. Esta observación puede ser útil para la compresión de la secuencia de atributos temporales, pero esta línea de investigación no se seguirá en este estudio.

5.4.4. Evaluación comparativa con referencias (benchmarking)

Los experimentos en ciencias de la computación gozan del beneficio de ser replicables con precisión (siempre que los datos de entrada, el código fuente y/o las herramientas utilizadas se encuentren disponibles) pero, por otro lado, sufren el inconveniente de que, habitualmente, los resultados con distintos conjuntos de datos de entrada o con distintas herramientas, así se trate del mismo método, son inconmensurables [148]. A modo de ejemplo, Killourhy y Maxion han advertido que, dentro del ámbito de la autenticación con cadencias de tecleo, comparar las tasas de error de dos métodos distintos evaluados con distintos conjuntos de datos es síntoma de un mal diseño del experimento [21]. Una comparación justa entre varios métodos demanda ser llevada a cabo sobre el mismo conjunto de datos, y una evaluación justa de un método demanda ser llevada a cabo sobre diversos conjuntos de datos.

Como una manera de superar estos últimos inconvenientes se ha propuesto la metodología de evaluación comparativa con referencias, en inglés *benchmarking* [149]. La manera más común de llevar a cabo esta propuesta es utilizar bancos de prueba estandarizados. Por ejemplo, una herramienta de *benchmark* para procesadores utilizaría el mismo conjunto de rutinas que evalúen las distintas capacidades y componentes del sistema. Pero esto no siempre es factible, o puede que no existan bancos de prueba estandarizados para la tarea. En esos casos, la recomendación es utilizar métodos con implementación pública como casos de base para la comparación [150].

5.4.5. Selección de atributos basada en correlación

En el campo del aprendizaje automático, se denomina *selección de atributos* al proceso de elección de un subconjunto de variables, valores predictores, o atributos derivados, para la construcción de un modelo. El objetivo de la selección de atributos puede ser reducir la dimensionalidad o la complejidad del modelo, acotar el tiempo de entrenamiento y evaluación, mejorar la compatibilidad de los datos con el clasificador utilizado, o establecer que subconjunto de los atributos es más relevante para determinar la clase de las instancias.

Al realizar la selección de atributos presuponemos que no todos ellos cargan el mismo poder predictivo, e intentamos determinar con cuales quedarnos y cuales descartar. La

hipótesis fundamental de la selección de atributos basada en correlación (CFS) es que los mejores subconjuntos son aquellos compuestos de atributos fuertemente correlacionados con la clasificación y poco correlacionados entre ellos [136].

Consideremos que tenemos un conjunto A de atributos, y que *S* es un subconjunto de ellos. El método CFS resuelve el problema de hallar el subconjunto S que maximiza el mérito

$$M(S) = \frac{k \,.\, r_C}{\sqrt{k + \,(k+1) \,.\, r_f}} \tag{5.10}$$

en donde $k = |S|$ es la cantidad de atributos en S, r_c es el promedio de las correlaciones de los miembros de S con la clasificación, y r_f es el promedio de las correlaciones entre miembros de S.

Capítulo 6

Resultados y discusión

> Un científico es una máquina que convierte interrogantes existentes en conocimiento nuevo, y un filósofo es una máquina que convierte conocimiento existente en nuevos interrogantes
>
> Apócrifo

En este capítulo se detallan los resultados de los tres experimentos que han sido descritos en la sección anterior. Con el objetivo de evitar la navegación errática entre secciones, se ha intentado ilustrar los resultados completos de cada intento, agrupados por conjunto de datos, en el orden de exposición. La única excepción son las tablas detalladas de resultados del experimento sobre distribuciones subyacentes, que han sido relegadas al apéndice C; si bien se incluyen en este volumen por completitud, no son necesarias para comprender las conclusiones y basta para ello con recorrer las observaciones de la sección 6.1.2.

Ofrecer en este medio un detalle más fino que el agrupado por conjunto de datos resulta imposible. Hablamos aquí de miles de usuarios, con decenas de perfiles de teclas, centenares de sesiones, a veces miles de oraciones, y aún más palabras individuales para cada uno de ellos. El tamaño de los conjuntos de datos de resultados, enumerados en la sección**??** y que han sido puestos a disposición de la comunidad de investigadores, lo atestigua con algunos GBs de información descargable para cada uno de ellos. Los comportamientos individuales conforman un zoológico de idiosincrasias [151] que quizás valga la pena discriminar en un estudio ulterior, pero que por ahora queda relegado a las futuras líneas de investigación. El lector motivado siempre puede utilizar los conjuntos de datos, públicamente accesibles en forma gratuita, para llevar adelante un estudio más exhaustivo que el alcance de esta tesis nos habilita.

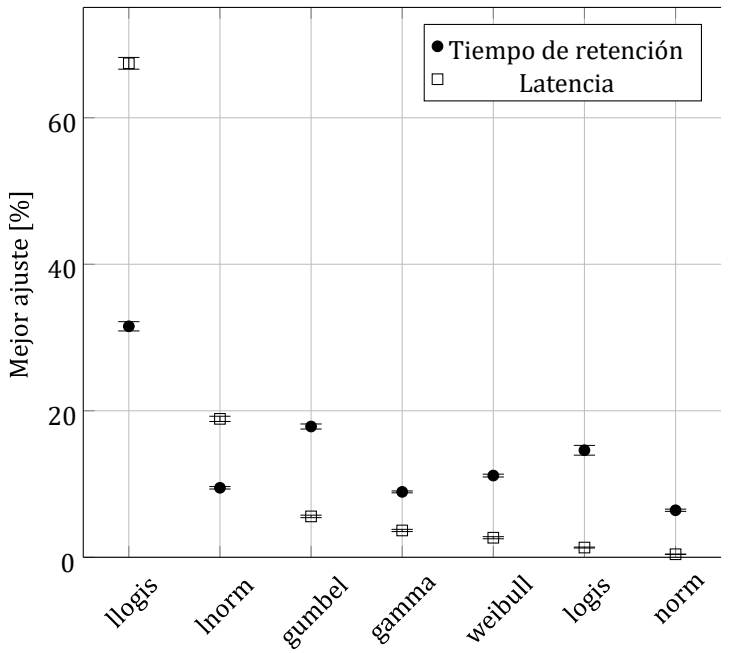

Figura 6.1: Merito relativo para distribuciones de dos parámetros

El resto del capítulo está organizado como sigue. La sección 6.1 discute los resultados del experimento sobre distribuciones subyacentes. La sección 6.2 discute los resultados del experimento sobre síntesis de muestras artificiales y contramedidas de defensa. La sección 6.3 discute los resultados del experimento sobre identificación del texto ingresado utilizando atributos temporales. La sección de cada experimento incluye una conclusión propia, restringida a su alcance; todas ellas serán sintetizadas en una única narrativa global en el capítulo siguiente. Podemos anticipar, para facilitar la lectura, que los resultados del experimento sobre distribuciones subyacentes motivan los métodos propuestos y evaluados en el segundo experimento; el tercero es un *bonus* fortuito, un grato resultado adicional que, con mínimas modificaciones del segundo experimento, nos devuelve una ganancia inesperada y valiosa. *Serendipitously*, como reza un intraducible adjetivo inglés.

6.1. Distribuciones subyacentes

Las tablas de resultados detallados para este experimento pueden consultarse en el apéndice C. El primer conjunto de tablas, que abarca desde la tabla C.1 hasta la tabla C.4, fue creado contando cuantas veces cada distribución candidata proporciona el mejor ajuste y clasificándolas en orden de éxito. Las tablas C.1 y C.2 muestran resultados detallados para tiempos de retención y latencia respectivamente, para distribuciones de dos parámetros; ambos están representados, junto con sus intervalos de confianza, en la figura 6.1. En forma similar, las tablas C.3 y C.4 muestran resultados detallados para tiempos de retención y latencia para distribuciones de tres parámetros, mientras que la figura 6.2 los muestra gráficamente y con intervalos de confianza.

El segundo conjunto de tablas, que abarca desde la tabla C.5 hasta la tabla C.8,

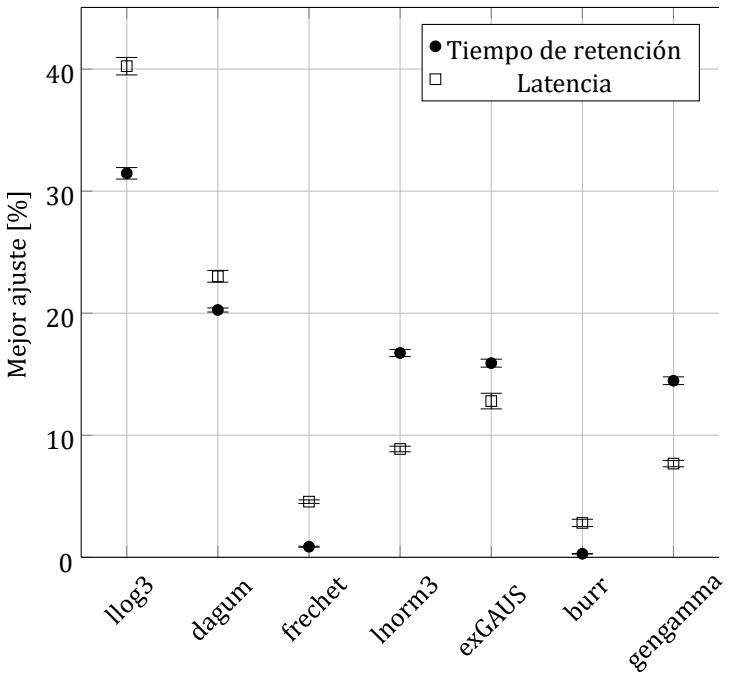

Figura 6.2: Merito relativo para distribuciones de tres parámetros

muestra el promedio de la mitad de la métrica AICc de cada perfil en cada conjunto de datos/tipo de tarea de escritura. Cuanto menor es este valor, mejor es el ajuste promedio. El diseño de tablas y figuras, para tiempos de espera y tiempos de vuelo, y para distribuciones de dos y tres parámetros, sigue el del primer conjunto de tablas. Las figuras 6.3 y 6.4 resumen las cuatro tablas con intervalos de confianza para los valores.

Finalmente, el tercer conjunto de tablas, que abarca desde la tabla C.9 hasta la tabla C.12, clasifica las distribuciones candidatas de acuerdo con la frecuencia de rechazo de hipótesis. Cuanto menor es este valor, mejor es el ajuste promedio. Una vez más, se ha seguido un diseño similar. Las figuras 6.5 y 6.6 resumen las cuatro tablas con intervalos de confianza para los valores.

El sombreado gris en cada celda de cada tabla del apéndice está destinado a transmitir, de un vistazo, el mérito relativo de cada distribución en su fila, que corresponde a un conjunto de datos y una tarea. Un sombreado más claro significa un mejor rendimiento. Por ejemplo, en la tabla C.1, Gumbel es la distribución con el recuento de coincidencias más alto y, por lo tanto, representa el mejor ajuste, mientras que la distribución normal tiene el peor rendimiento y, por lo tanto, se muestra como el más oscuro.

El cuadro 6.1 resume la distribución de mejor rendimiento para cada línea de cada tabla. Una línea del cuadro 6.1 incluye las distribuciones ganadoras en las doce categorías anteriores para el mismo conjunto de datos y tarea. Se ha utilizado la información de las tablas que van desde C.3 a C.12. Como llogis/llogis y llog3/llog3 son las entradas más comunes para dos y tres parámetros, las entradas excepcionales en donde aparecen otras distribuciones se muestran en letra negrita para que sean más fáciles de detectar.

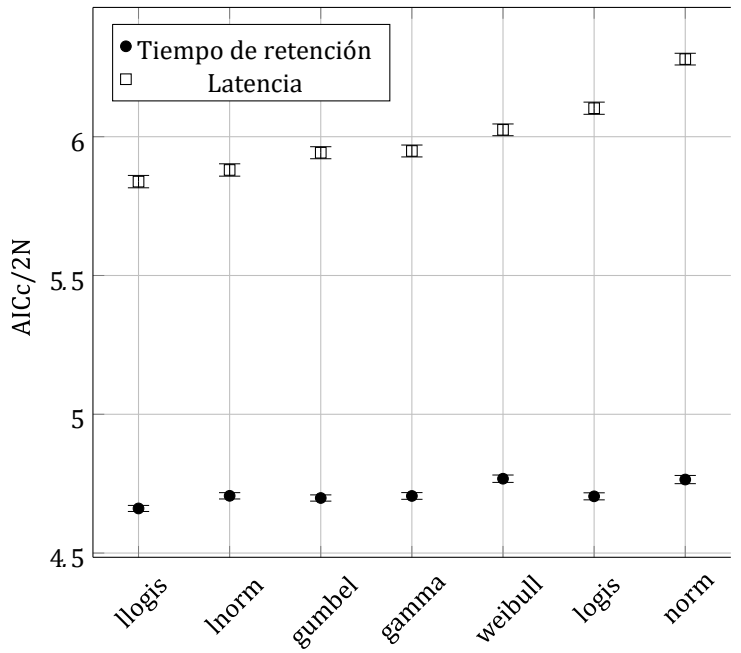

Figura 6.3: Valores promedio de un medio de AICc para distribuciones de dos parámetros

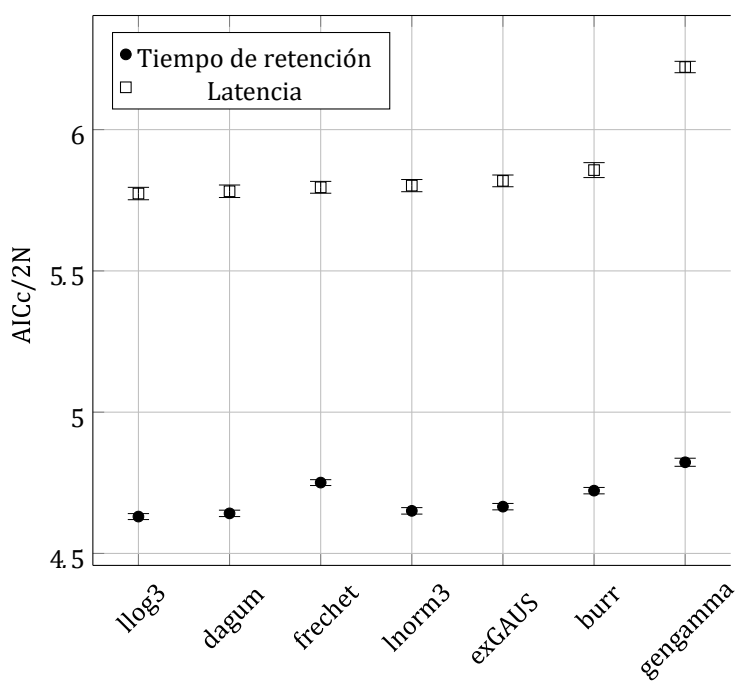

Figura 6.4: Valores promedio de un medio de AICc para distribuciones de tres parámetros

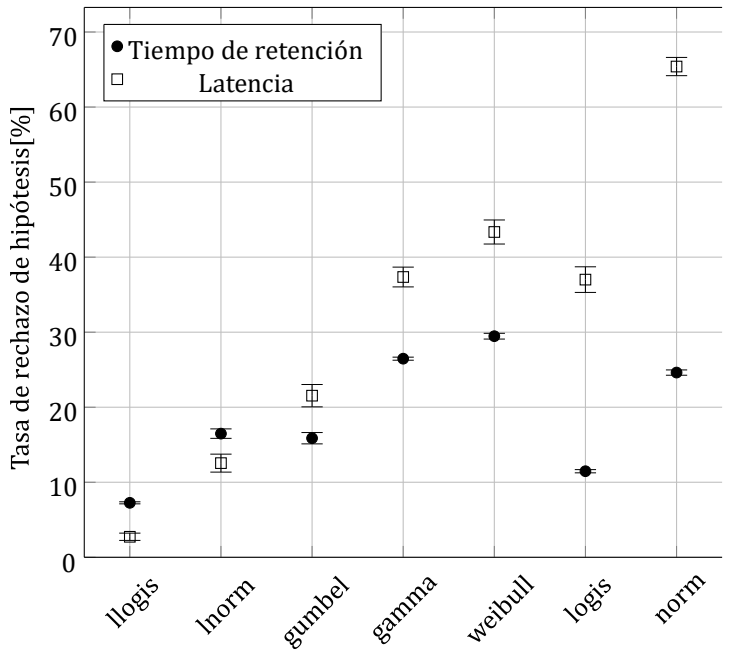

Figura 6.5: Tasa de rechazo de hipótesis para distribuciones de dos parámetros

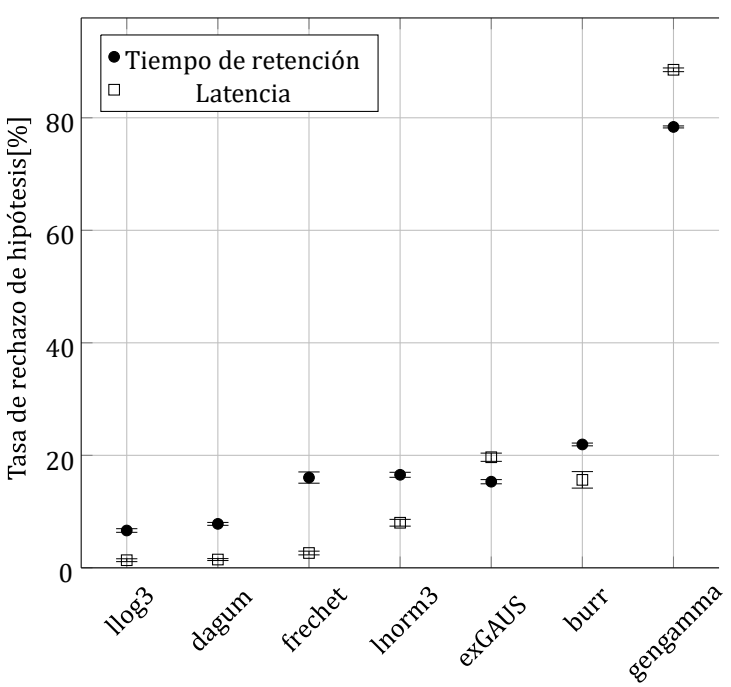

Figura 6.6: Tasa de rechazo de hipótesis para distribuciones de tres parámetros

		Dos parámetros			Tres parámetros		
Dataset	**Tarea**	**Mejor** HT/FT	**Promedio** HT/FT	**Menos rech.** HT/FT	**Mejor** HT/FT	**Promedio** HT/FT	**Menos rech.** HT/FT
LSIA	Free text	**gumbel**/llogis	llogis/llogis	llogis/llogis	**lnorm3**/llog3	**dagum**/llog3	$llog^3$/$llog^3$
KM	Free text Transcription	logis/llogis **gumbel**/llogis	logis/llogis	logis/llogis logis/**lnorm**	$llog^3$/$llog^3$ **lnorm3**/llog3	$llog^3$/$llog^3$ **dagum**/llog3	**lnorm3**/**dagum** **dagum**/**lnorm3**
PROSODY GAY	Copy [1] Copy [2] Fake Essay True Essay	logis/llogis	logis/llogis	logis/llogis	$llog^3$/$llog^3$	$llog^3$/$llog^3$	$llog^3$/$llog^3$ llog3/**dagum**
PROSODY GUN	Copy [1] Copy [2] Fake Essay True Essay	logis/llogis	logis/llogis	logis/llogis	$llog^3$/$llog^3$	$llog^3$/$llog^3$	$llog^3$/$llog^3$
PROSODY REVIEW	Copy [1] Copy [2] Fake Essay True Essay	logis/llogis	logis/llogis	logis/llogis	$llog^3$/$llog^3$	$llog^3$/$llog^3$	$llog^3$/$llog^3$

Cuadro 6.1: Resultados para cada conjunto de datos del experimento sobre distribuciones subyacentes

6.1.1. Validación

Varios fenómenos bien conocidos se pueden utilizar como casos de prueba para validar la corrección general de los resultados anteriores. Una de ellas es que los tiempos de retención, al ser el resultado de un proceso puramente motor, contienen necesariamente menos información que la latencia, pues estas últimas no solo incluyen retrasos en las decisiones sobre qué y cómo escribir, sino también pausas e interrupciones externas. Este último fenómeno es fácil de observar; los usuarios rara vez, o nunca, responden a las interrupciones manteniendo presionada una tecla. El segundo conjunto de cuadros muestra valores de AICc alrededor de 4,5 para tiempos de retención y alrededor de 6 para latencias; el significado de estos números se ha descripto en la sección 5.4.3. Incluso luego de considerar los intervalos de confianza ambos valores se mantienen disjuntos, proporcionando evidencia a favor y una estimación aproximada del contenido de información de los tiempos de retención y latencia.

Otro fenómeno evidente que se puede utilizar para validar el experimento es que las distribuciones con tres parámetros proporcionan un mejor ajuste que aquellas con dos. Una vez más, el segundo conjunto de tablas demuestra esta afirmación. En el tercer conjunto de tablas, se muestra que la distribución normal es rechazada con mucha frecuencia tanto para los tiempos de retención como para latencias. Los histogramas de estas últimas, al tener colas más pesadas, presentan un mayor porcentaje de rechazos. Es más, el hecho de que los histogramas de tiempos de retención presentan formas más estables se puede leer en que los porcentajes de distintas distribuciones no difieren tanto entre sí en comparación con los tiempos de vuelo.

Aunque no es un claro ganador, la distribución log-normal ha demostrado ser útil tanto con dos como con tres parámetros. En casi todos los cuadros, aunque no sea una de las de mejor desempeño tampoco es una de las peores. Observamos una curiosa excepción en la tabla C.1 para los tiempos de retención del conjunto de datos KM, donde incluso la distribución normal se elige con más frecuencia que la log-normal como la que mejor ajusta.

Las observaciones anteriores sobre la forma de las distribuciones de tiempo en los perfiles de cadencias de tecleo, que repiten las ya establecidas por otros autores como se indica en la sección 2.4.5, fueron confirmadas por este estudio. De esta forma se valida en términos generales la metodología e implementación de este experimento. Varios fenómenos nuevos e interesantes surgen también de los resultados numéricos, que trataremos a continuación.

6.1.2. Observaciones

A continuación, enumeramos un conjunto de observaciones relevantes que surgen de la inspección de los resultados.

- **La distribución log-logística, tanto con dos como con tres parámetros, es una clara ganadora entre todos los candidatos**. Ajustando los tiempos de latencia con dos parámetros, su recuento de mejores coincidencias para todos los conjuntos de datos es de alrededor del 50% en LSIA y por encima del 65% en el resto, mientras que el siguiente candidato, la distribución log-normal de dos parámetros, alcanza a lo sumo el 28% en LSIA y menos del 20% en el resto. En cuanto a tiempos de retención, supera siempre a las otras distribuciones por un margen de más del 10%, y usualmente alrededor del

20%, con la excepción de las dos tareas de KM. Con tres parámetros, la distribución log-logística sigue superando al resto en más de un 10%, con un margen más amplio para las latencias. Alcanza consistentemente el mínimo — o a lo sumo el segundo mínimo — en la tasa de rechazos de hipótesis para todos los conjuntos de datos, mientras que proporciona siempre el menor contenido de información (seguido de cerca por Dagum y log-normal) tanto para los tiempos de retención como para los de latencia. Constituye una grata sorpresa que, hasta donde alcanza nuestro conocimiento, esta distribución no haya sido mencionada previamente´ en la literatura sobre análisis de cadencias de tecleo.

- **La distribución que mejor ajusta no depende demasiado del conjunto de datos que se utilice para la evaluación**. Siempre que la distribución log-logística (tanto con dos como con tres parámetros) no alcanza el primer lugar, ocupa el segundo o tercero y por un margen insignificante. Por lo tanto, las condiciones ambientales del entorno de captura de los datos de evaluación no parecen tener una gran influencia en la forma general de los histogramas de tiempo.

- **Los méritos relativos de las distribuciones de tres parámetros no son tan claros**. Las tablas de contenido de información promedio y tasa de rechazo de hipótesis no muestran una distinción tan clara entre la distribución de mejor ajuste y las siguientes, como sí lo hacen las tablas para distribuciones de dos parámetros. La mayoría de las veces, tres o incluso cuatro de ellas presentan valores muy similares e intervalos de confianza casi superpuestos. Los valores de Dagum casi siempre están cerca de los de la distribución log-logística. El contenido de información promedio de la distribución log-logística, Dagum, Frechet y log-normal es casi idéntico.

- **La distribución log-normal es la segunda mejor opción entre los candidatos de dos parámetros**, aunque es un poco peor que Gumbel para los tiempos de retención. Sigue siendo una elección buena cuando se prefiere la versión de tres parámetros, aunque su rendimiento se encuentra detrás del de las distribuciones Dagum y exgaussiana. No es sorprendente, teniendo en cuenta la atención que ha recibido en el pasado en la literatura del tema. Al observar las tablas también se puede ver que, junto con la distribución log-logística, la versión de dos parámetros puede competir con distribuciones de tres parámetros en contenido de información promedio y tasa de rechazos de hipótesis, lo que la confirma como un excelente candidato.

- **El buen rendimiento de la distribución Dagum es inesperado**, ya que nunca antes había sido considerado en la literatura sobre análisis de cadencias de tecleo en texto libre. Sin embargo, su desempeño no está lejos del de la distribución log-logística.

- **La distribución exgaussiana no es una muy buena elección para modelar histogramas de latencias**. A pesar de la motivación presentada en la sección 2.4.5, la distribución exgaussiana no suele proporcionar el mejor ajuste y su tasa de rechazo de hipótesis es alto, especialmente en el conjunto de datos PROSODY. Sin embargo, a pesar de que el contenido de información promedio es relativamente uno de los peores en el cuadro C.8, tampoco está lejos de los mejores en términos absolutos. Las interrupciones externas aumentan el ruido en los histogramas y la distribución exgaussiana sufre sus

efectos más que los otros candidatos que pueden absorberlas sin problemas, lo que explica la elevada tasa de rechazo de hipótesis y el pobre recuento de mejor ajuste.

- **Las diferentes tareas de escritura y temas en el conjunto de datos de PROSODY no cambian significativamente la distribución que mejor se ajusta**. Los tres conjuntos de tablas muestran rangos de mérito bastante similares para cada tarea y tema en el conjunto de datos de PROSODY. Las formas generales de los histogramas de tiempos no parecen tampoco estar correlacionadas con el contexto emocional del usuario que escribe, excepto por la forma en que este último influye en los parámetros de la distribución subyacente. Esta es una observación interesante, porque confirmarla descartaría la distribución de mejor ajuste como criterio para detectar si el usuario está mintiendo o no. Desafortunadamente, al ser el único conjunto de datos con una distinción tan fina, no podemos saber si esta observación es generalizable o sólo se aplica al conjunto de datos PROSODY.

6.1.3. Conclusión del experimento

El resultado principal de este experimento es que *la distribución log-logística es una clara ganadora entre todos los candidatos para ajustar los histogramas de tiempo, tanto de retención como de latencia, producto de las cadencias de teclo, pero las tasas de rechazo de hipótesis y los méritos relativos de esta y otras distribuciones muestran que un enfoque que considere los histogramas empíricos en su individualidad es promisorio para la autenticación de usuarios*.

La conclusión anterior motiva y justifica tanto las distancias como los métodos de síntesis basados en histogramas empíricos que han sido propuestos en las secciones 4.2 y 4.3.

6.1.4. Preguntas abiertas

Una pregunta que surge inmediatamente de la conclusión anterior es si considerar la distribución log-logística, en lugar de la log-normal u otras que se han reseñado en 2.4.5, puede reducir las tasas de error de los algoritmos existentes más allá de las reportadas actualmente, y en cuanto. Otro interrogante, más complejo, involucra explicar el proceso motor y el proceso de decisión que, conjuntamente, dan como resultado tal distribución para los tiempos de retención y latencia. Ambas quedan fuera del alcance de este estudio, y se relegan a las futuras líneas de investigación.

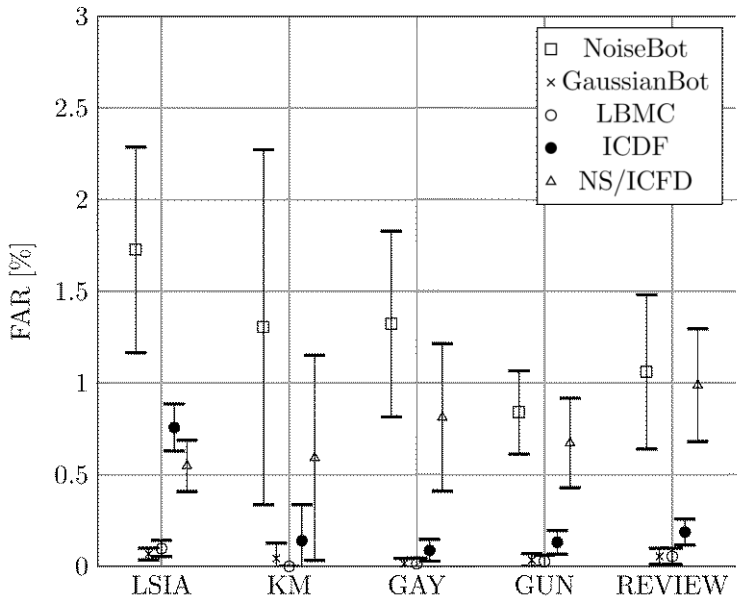

Figura 6.7: Tasas de falsos positivos para todos los conjuntos de datos, con entrenamiento interusuario

6.2. Síntesis de muestras artificiales y contramedidas de defensa

6.2.1. Rendimiento de las estrategias de síntesis

El éxito de una estrategia de suplantación de identidad se puede medir por la tasa de falsos positivos que puede lograr frente a un sistema de verificación. Las tasas de falsos positivos para todas las estrategias y todos los conjuntos de datos, junto con sus intervalos de confianza del 95%, se muestran en las figuras 6.7 y 6.8; la primera muestra los resultados cuando robot malo solo tuvo acceso a un perfil de entrenamiento interusuario, y la segunda cuando robot malo contaba con acceso pleno al perfil del usuario objetivo. Las tasas de falsos positivos para la estrategia de mejor rendimiento y sus intervalos de confianza del 95% se detallan para todos los conjuntos de datos en la tabla 6.2.

Cuando el atacante dispone solo de perfiles interusuario, las mejores estrategias resultan ser NoiseBot y NS/ICDF. El primero alcanza tasas de falsos positivos entre 1% y 2%. Las diferencias entre el ganador y el segundo no son tan notables como en el caso de entrenamiento intrausuario y, teniendo en cuenta los intervalos de confianza superpuestos, no se puede atribuir ninguna significación estadística a aquellos excepto para LSIA, donde NoiseBot es un claro ganador.

Observamos en la figura 6.8 que ICDF es la estrategia de síntesis más exitosa contra el sistema de defensa propuesto cuando tiene acceso al perfil intrausuario. Alcanza tasas de falsos positivos entre el 15% y el 20%, casi triplicando el rendimiento de la que la sigue en todos los casos. Esto es notable, considerando que en el 60 % de las instancias utilizadas para entrenar al clasificador, robot malo emplea ICDF para falsificar muestras sintéticas. Concluimos de aquí que este fenómeno no puede descartarse

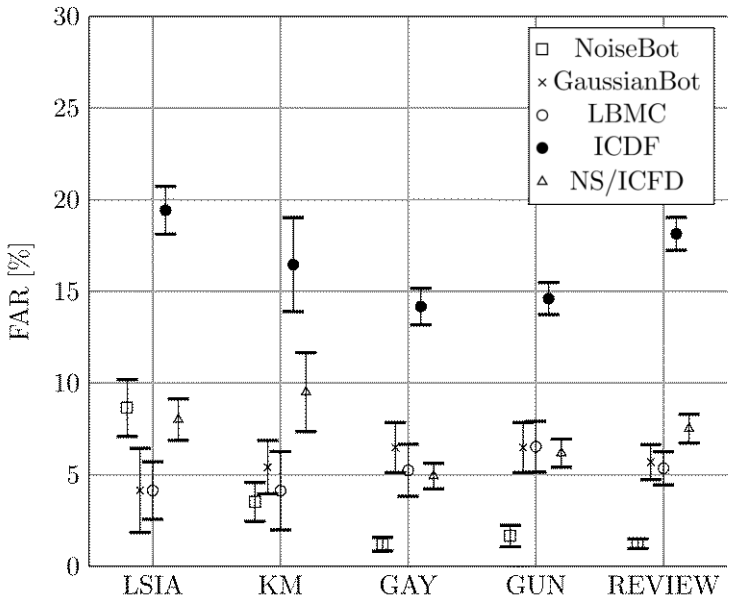

Figura 6.8: Tasas de falsos positivos para todos los conjuntos de datos, con entrenamiento intrausuario

como un efecto del clasificador careciendo de información suficiente para distinguir las falsificaciones hechas con ICDF del comportamiento del usuario legítimo. Interpretamos que ambos, muestra legítima y muestra sintetizada con ICDF, son lo suficientemente similares como para ser indistinguibles para este método en una de cada cinco a seis muestras a verificar.

6.2.2. Rendimiento del método de defensa

A la inversa del caso de las estrategias de suplantación de identidad, el éxito del sistema de detección se puede cuantificar por la baja tasa de falsos positivos que puede lograr frente a las estrategias de síntesis de muestras artificiales, ya sean las utilizadas para entrenar a su clasificador o las que sean verificadas. Como hemos enfrentado al sistema de detección propuesto contra las estrategias propuestas, y también algunas utilizadas en estudios anteriores, las figuras y tablas que muestran los resultados son las mismas que antes.

		Entrenamiento interusuario		Entrenamiento intrausuario	
Dataset		**FRR [%]**	**FAR [%]**	**Mejores estrategias**	**FAR [%]**	**Mejores estrategias**
LSIA		2.01 (± 0.16)	1.72 (± 0.56)	NoiseBot **ICDF**	19.42 (± 1.30)	**ICDF** NoiseBot
KM		1.86 (± 0.59)	1.30 (± 0.97)	NoiseBot **NS/ICDF**	16.46 (± 2.56)	**ICDF NS/ICDF**
PROSODY	GAY	0.99 (± 0.17)	1.32 (± 0.51)	NoiseBot **NS/ICDF**	14.17 (± 0.99)	**ICDF** GaussianBot
	GUN	1.01 (± 0.16)	0.84 (± 0.23)	NoiseBot **NS/ICDF**	14.61 (± 0.87)	**ICDF** LBMC
	REVIEW	1.14 (± 0.14)	1.06 (± 0.42)	NoiseBot **NS/ICDF**	18.14 (± 0.89)	**ICDF NS/ICDF**

Cuadro 6.2: Tasas de falsos positivos y negativos de la mejor estrategia de síntesis, para cada conjunto de datos

Cuando se evalúa el método de defensa contra la estrategia de síntesis de mejor rendimiento, las tasas de falsos positivos varían entre 1% y 2% para NoiseBot utilizando perfiles interusuario, pero aumentan a entre 15% y 20% si se utiliza ICDF con perfil intrausuario. Por lo tanto, cuando un atacante no tiene acceso a información privilegiada del usuario objetivo, el sistema de detección es muy efectivo. En el peor de los casos, frente a un atacante sofisticado que aprovecha una filtración de datos que haya revelado todas las muestras de los usuarios objetivo, aun podemos esperar del sistema de detección una protección significativa, aunque no optima, contra falsificaciones sintéticas. A lo sumo una de cada cinco o seis muestras falsificadas podrá engañar al sistema.

Como se espera que el sistema de detección se enfrente al usuario legítimo con más frecuencia que a un atacante, también necesitamos conocer sus tasas de falsos negativos. Estas varían entre 1% y 2%, como se muestra, para cada conjunto de datos y junto con sus intervalos de confianza del 95%, en la tabla 6.2. Aquí no hay distinción para los casos intrausuario e interusuario; como se explicó en la sección 4.4, el sistema de detección siempre tiene acceso a los datos del sujeto.

6.2.3. Relevancia de las distancias basadas en histogramas empíricos

La aplicación de la técnica de selección de atributos basada en correlación (*correlationbased atribute selection* [136]) sobre los conjuntos de datos de entrenamiento del sistema de defensa entrega subconjuntos de entre seis y siete características. Consistentemente en todos los conjuntos de datos, los atributos de distancias basadas en histogramas empíricos aparecen en cada uno de los subconjuntos, y comprenden alrededor del 60% de los atributos seleccionados.

Lo que, es más, proporcionan casi siempre la mayor ganancia de información, con valores alrededor de 0,25, para los tiempos de latencia y la mejor o la segunda mejor, con valores alrededor de 0,45, para los tiempos de retención. Los resultados detallados para cada atributo y usuario se proporcionan en los conjuntos de datos de resultados que se han hecho públicamente accesibles [140, 139].

6.2.4. Resultados comparados

Los resultados de la sección 6.2.1 sirven como experimento comparativo para evaluar el rendimiento de las estrategias propuestas frente a enfoques anteriores como NoiseBot, GaussianBot y LBMC. Se descubrió que, con acceso completo al perfil intrausuario del objetivo, la estrategia ICDF supera a cualquier otro método por un amplio margen, duplicando e incluso triplicando sus tasas de falsos positivos y alcanzando un promedio para todos los usuarios de todos los conjuntos de datos en el orden del 15%.

La importancia de incrementar el orden de los contextos para lograr el resultado que se reporta es ilustrada en la figura 6.9, que muestra el crecimiento de la tasa de falsos positivos de la estrategia de síntesis ICDF con el orden de contexto del modelado por contextos finitos. Las mejoras obtenidas al exceder un valor de siete para el orden del contexto no son significativas; la cantidad de observaciones de

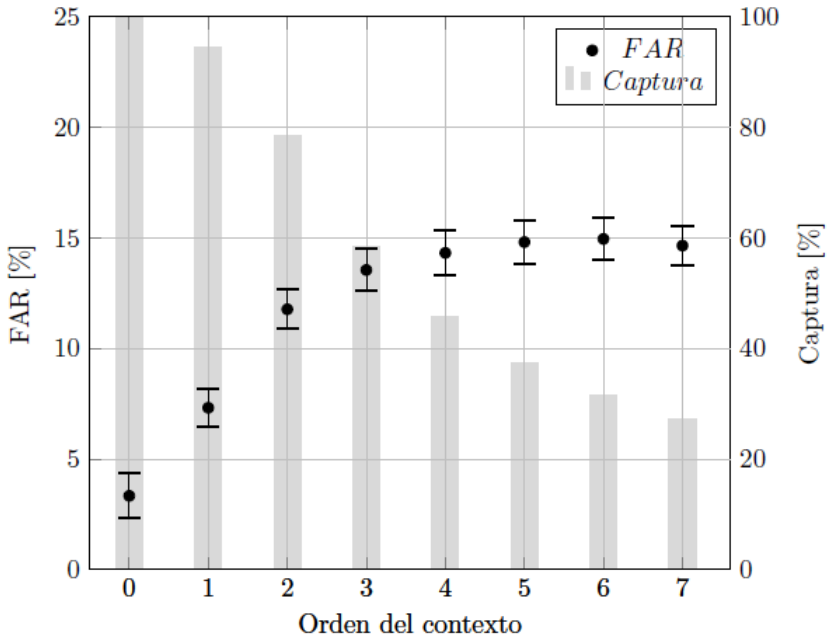

Figura 6.9: Porcentaje de captura en función del orden del contexto, y su efecto sobre la tasa de falsos positivos alcanzada por la estrategia de síntesis

atributos temporales capturadas por contextos de orden alto se reduce rápidamente, como muestra la misma imagen.

Aunque sus resultados parecen no impresionar a primera vista, LBMC demuestra ser un método robusto. Debe considerarse que los valores que alcanza se logran sin conocer los nombres de las teclas en el texto de entrenamiento, en contraste con las otras estrategias que si lo hacen. Finalmente, se puede notar que el fenómeno mencionado en la sección 5.3.2.4 todavía se hace evidente. NoiseBot supera ligeramente a todas las demás estrategias cuando se utilizan perfiles interusuario, incluso después de agregar un 20% de instancias de entrenamiento correspondientes al clasificador.

Las tasas de falsos positivos de NoiseBot se reducen a la mitad con el sistema de detección propuesto, en contraste con valores entre 1% y 3,5% alcanzados en [27, 29]. Lo que, es más, el método propuesto adolece de una penalización menor en la tasa de falsos negativos. La efectividad de GaussianBot frente a nuestro sistema se ve reducida a alrededor del 0,1 %. Incluso proporcionando a los atacantes el perfil intrausuario completo del objetivo, superamos a [26] por un margen muy amplio.

6.2.5. Conclusión del experimento

Las conclusiones principales de este experimento son dos. En primer lugar, *las estrategias de síntesis basadas en histogramas empíricos alcanzan un mejor rendimiento que aquellas basadas en distribuciones suaves al intentar engañar a un sistema de autenticación basado en cadencias de tecleo*. En segundo lugar, *las distancias basadas en histogramas empíricos alcanzan un mejor rendimiento que las tradicionales y otros métodos de clasificación al intentar detectar muestras sintetizadas*. Ambos resultados establecen la

importancia de utilizar histogramas empíricos en el análisis de cadencias de tecleo. A continuación,

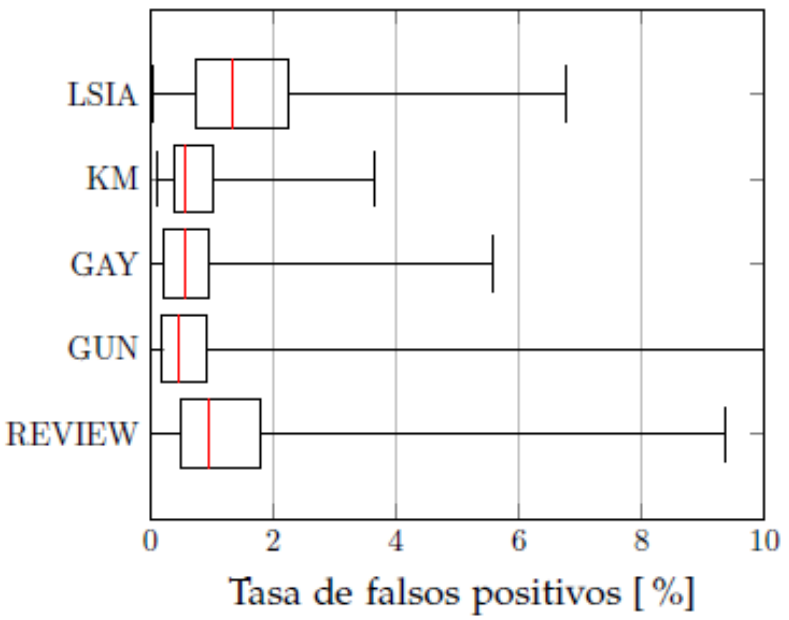

Figura 6.10: Tasas de falsos positivos por conjunto de datos para el experimento sobre identificación del texto ingresado, con entrenamiento intrausuario.

sintetizamos el resto de los resultados particulares.

Se evaluaron las estrategias de síntesis propuestas, que aprovechan contextos de orden alto y distribuciones empíricas para generar muestras artificiales de cadencias de tecleo, junto con un sistema de detección de vida que las emplea internamente como adversarios. Además, se determinó la utilidad de la familia de distancias basada en los histogramas empíricos de los atributos temporales para ayudar al clasificador a detectar falsificaciones que son demasiado suaves o demasiado buenas para ser auténticas.

Se demostró que una de las estrategias propuestas, ICDF, supera por un amplio margen a otros métodos previamente evaluados en la literatura, duplicando e incluso triplicando sus tasas de falsos positivos. Esta alcanzo, para un promedio de todos los usuarios y todos los conjuntos de datos, un valor de alrededor del 15% cuando tenía acceso al perfil completo del usuario objetivo. Si solo los datos de la población general se encuentran disponibles para el atacante, el sistema de detección de vida logra tasas de falsos positivos y negativos entre 1% y 2%.

La técnica de selección de atributos basada en la correlación demostró la relevancia de las distancias propuestas para lograr el rendimiento antedicho.

Cabe aclarar que el esquema de detección propuesto debe ser implementado como una segunda etapa en un sistema de autenticación, destinada a descartar falsificaciones sintéticas, pero no a identificar al usuario. No estamos en condiciones de afirmar nada sobre su precisión cuando se presenta una muestra de otro usuario humano; tal es la tarea

de un sistema clásico de autenticación por cadencias de tecleo. Debe notarse también que, por la propia naturaleza del método, este no defiende contra ataques de repetición, que requieren un enfoque como el de [87], que ha sido discutido en la sección 2.4.7.

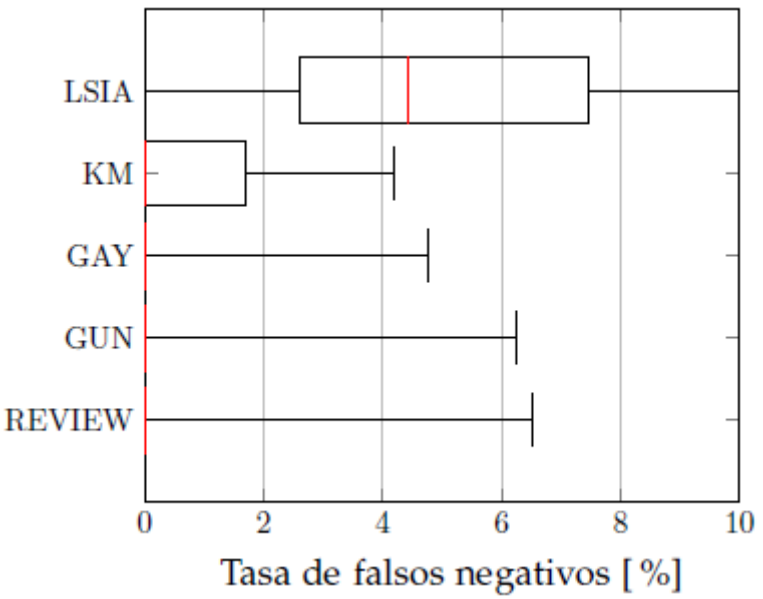

Figura 6.11: Tasas de falsos negativos por conjunto de datos para el experimento sobre identificación del texto ingresado, con entrenamiento intrausuario.6.3. Identificación del texto ingresado utilizando

atributos temporales

6.3.1. Rendimiento del método

El cuadro 6.3 resume las tasas de falsos positivos y falsos negativos para cada conjunto de datos, tanto para el entrenamiento intrausuario como interusuario. Se incluyen a continuación de todos los valores sus correspondientes intervalos de confianza del 95% para permitir compararlos con significatividad estadística. Para comprender mejor la distribución de las tasas de error entre los usuarios, se muestran diagramas de caja para las tasas de falsos positivos y falsos negativos para el entrenamiento intrausuario, en las figuras 6.10 y 6.11, y para el entrenamiento interusuario, en las figuras 6.12 y 6.13. Debe tenerse en cuenta que el grafico de tasas de falsos negativos en la última figura tiene una escala diferente en el eje de las *x*.

El método propuesto logra bajas tasas de error con oraciones y listas más largas que aquellas consideradas previamente en la literatura. Todos los conjuntos de datos puntúan por debajo del 1% en tasas de falsos positivos y negativos cuando se utiliza entrenamiento intrausuario, con la excepción de LSIA donde la primera es ligeramente superior y la última se eleva a alrededor del 5%. La consistencia de los valores para mantenerse en el mismo rango es notable, si consideramos cuan diferentes son las condiciones ambientales y tareas de escrituras que están representadas en los tres conjuntos de datos de evaluación. Por muy bajas que sean las tasas de error promedio, se puede observar en los diagramas de caja que las distribuciones son de cola larga. Algunos de los usuarios con peor desempeño que se

encuentran en el último cuartil presentan tasas de error más de un orden de magnitud más altas. Esto es más notorio para las tasas de falsos negativos.

Cuando se utilizan datos de la población general para el entrenamiento en lugar del perfil intrausuario esperamos un aumento en las tasas de error, pero la cantidad

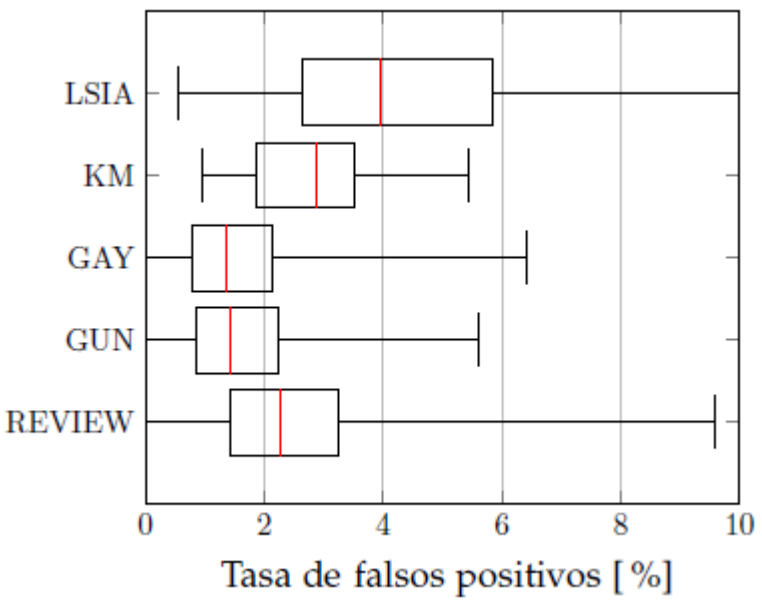

Figura 6.12: Tasas de falsos positivos por conjunto de datos para el experimento sobre identificación del texto ingresado, con entrenamiento interusuario

virtualmente ilimitada de datos disponibles debería ayudar a mitigar un poco este aumento. Empíricamente, las tasas de falsos positivos aumentan entre dos y tres veces, manteniéndose en torno al 2% con la excepción del conjunto de datos LSIA, mientras que las tasas de falsos negativos se disparan hasta el 20%. Las colas a la derecha de la distribución de usuarios con peores rendimientos también se alargan.

6.3.2. Discusión de los resultados y comparación con el estado del arte

Las tasas de error que se mostraron en la sección anterior podrían ser consideradas elevadas para un sistema de verificación biométrica, pero son más que competitivas si se las comparan con los métodos actuales para la identificación del texto ingresado utilizando atributos temporales que han sido reseñados en la sección 2.4.7.

Se ha mostrado que el método puede ser utilizado de manera efectiva, incluso si no se cuenta con muestras específicas del usuario objetivo para el entrenamiento pues los datos de la población general en cantidad suficiente pueden suplirlos en forma parcial. En comparación, [92] alcanza alrededor del 15% de verdaderos positivos; esto equivale a un 85% de falsos positivos, un valor mucho peor que el logrado por el método propuesto en esta tesis. Sin embargo, no sólo las tasas de error sino los méritos relativos deben ser tenidos en consideración. El problema atacado por [92] es más difícil que el aquí tratado pues pretende adivinar el texto carácter a carácter, y no con una lista de candidatos.

Hacer una comparación con [95] es más justo, pues tanto el método propuesto como el citado utilizan listas de candidatos. Aquí se han logrado tasas de error más bajas, y sometiendo la evaluación a las dificultades adicionales de clasificar textos más largos, con una lista más larga de candidatos potenciales, y sin muestras previas del texto correcto para entrenar el modelo. Como se ha visto en la sección 2.4.7, no hay más métodos para comparar los resultados con el aquí propuestos. De los

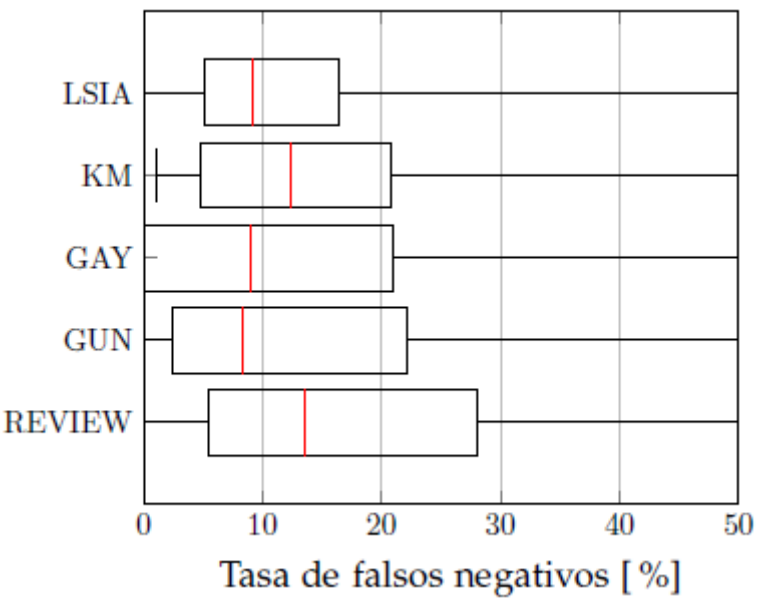

Figura 6.13: Tasas de falsos negativos por conjunto de datos para el experimento sobre identificación del texto ingresado, con entrenamiento interusuario

escasos seis artículos disponibles sobre el tema, solo los dos aquí discutidos admiten ser comparados con el método propuesto.

6.3.3. Conclusiones del experimento

La conclusión principal de este experimento es que *utilizando solo atributos temporales es posible identificar el texto ingresado dentro de una lista de candidatos de tamaño mediano utilizando el método propuesto, alcanzando tasas de error muy bajas y competitivas con el estado del arte, aunque tratemos con textos y listas de candidatos más largas*. De esta forma es factible potenciar la segunda etapa de un ataque por canal lateral que logre filtrar los tiempos entre eventos de teclas mientras el usuario escribe, aunque solo se puedan recuperar las latencias.

					Entrenamiento intrausuario		Entrenamiento interusuario	
Dataset	**Usuarios**	**Oraciones**	**Prom. Orac. por Usuario**	**Largo Prom. por Orac.**	**FAR [%]**	**FRR [%]**	**FAR [%]**	**FRR [%]**
LSIA	136	38264	281	81	1.63 (±0,21)	5.35 (±0,82)	4,32 (±0,4)	14,53 (±2,63)
KM	20	1522	76	72	0.81 (±0,36)	0.94 (±0,60)	2,79 (±0,51)	15,58 (±6,29)
GAY	400	7871	20	114	0.75 (±0,09)	0.12 (±0,07)	1,64 (±0,15)	16,99 (±2,59)
GUN	400	11537	29	119	0.85 (±0,14)	0.17 (±0,08)	1,63 (±0,11)	16,93 (±2,22)
REVIEW	500	13326	27	97	1.31 (±0,11)	0.09 (±0,05)	2,5 (±0,13)	20,15 (±1,79)

Cuadro 6.3: Resultados por conjunto de datos para el experimento sobre identificación de textos

Capítulo 7

Conclusiones

Cuando despertó, el dinosaurio todavía estaba allí

Augusto Monterroso

Hemos llegado al fin de esta tesis. Solo queda recapitular el camino seguido para comprender a vista de pájaro el arco global de este estudio, repasar las conclusiones, y recordar que preguntas quedaron sin respuesta.

El resto del capítulo está organizado como se describe a continuación. La sección 7.1 resume la narrativa global de esta tesis. La sección 7.2 sintetiza la conclusión general y las de los experimentos particulares. La sección 7.3 resume los resultados cuantitativos. La sección 7.4 enumera los aportes, que consisten en cuatro métodos, una herramienta integrada, y diversos conjuntos de datos. Finalmente, las secciones 7.5 y 7.6 retornan a las preguntas que este estudio ha dejado sin respuesta y plantean las futuras líneas de investigación y trabajo.

7.1. RECAPITULANDO EL CAMINO SEGUIDO

Comenzamos este camino notando, en la síntesis del estado del arte de la sección 2.6, que la tendencia actual de la disciplina, ya madura, es a tratar con las particularidades del caso ciertos problemas que los métodos biométricos tradicionales han debido enfrentar en el pasado. Enfatizamos la necesidad de diseñar todo sistema con la consideración de que se encontrará bajo ataque permanente, pero hallamos una laguna en la literatura al reseñar ataques de presentación en la sección 2.4.6: los métodos existentes de síntesis de cadencias de tecleo artificiales y las contramedidas de defensa resultantes se encuentran aún en la infancia. Sirva como ejemplo de este aserto que, con la excepción de [76], todavía se utilizan modelos gaussianos y contextos de orden uno. Hemos visto en la reseña sobre distribuciones subyacentes de la sección 2.4.5 que esta suposición ha sido abandonada ya para textos fijos, pero que aún falta una comparación sistemática y exhaustiva para textos libres.

Intentando compensar esta falta, decidimos encarar el experimento comparativo que se ha descrito en la sección 5.3.1 y cuyo objetivo es evaluar el mérito de distintas distribuciones de dos y tres parámetros para ajustar perfiles de tiempos de retención y latencia en textos libres. Con los resultados de la sección 6.1 a mano, confirmamos la utilidad de la distribución

log-normal estudiada por Montalvão [72] y descubrimos con alegría que la log-logística, nunca evaluada para la tarea, resulta una clara ganadora. Pero la conclusión cualitativa de mayor interés resultó ser que, cualesquiera las distribuciones que elijamos, el ajuste con leyes suaves deja fuera una cierta cantidad de información que no es ruido aleatorio, sino una expresión del comportamiento característico del usuario.

Este descubrimiento motivo la propuesta de las distancias basadas en histogramas empíricos de la sección 4.2, diseñadas para capturar la diferencia entre los idiosincráticos, característicos perfiles de cada usuario y una muestra artificial sintetizada utilizando distribuciones suaves. Sin embargo, la forma específica de los histogramas empíricos no solo ofrecía la posibilidad de mejorar la detección de ataques de presentación, sino también de mejorar las estrategias de síntesis existentes. Con este fin se propusieron los métodos de la sección 4.3. No solo para mejorar las posibilidades de que un ataque de presentación tenga éxito; al ser utilizadas como adversarios en el método de defensa propuesta en la sección 4.4, logramos protegernos contra ellas y contra el estado del arte previo superando el rendimiento de otros esquemas de defensa anteriores. El experimento que así lo establece ha sido descrito en la sección 5.3.2 y sus resultados reportados en la sección 6.2.

En ocasiones un método entrega más que aquello para lo cual ha sido diseñado. Tal es el caso del anterior que, con algunas modificaciones que se describen en la sección 4.5, ha demostrado servir también para identificar el texto ingresado utilizando solo los atributos temporales y no los nombres de teclas, como resulta de un ataque por canal lateral. Lo que es más, este resultó servir dentro de un alcance que la reseña de la sección 2.4.7 mostró como inatacado aún; mientras los métodos del estado del arte operan sobre textos cortos o listas de candidatos pequeñas, nuestro enfoque resulto apto para atacar textos largos y listas de candidatos con mayor cantidad de elementos. Para determinar el rendimiento del método, se diseñó el experimento descrito en la sección 5.3.3, cuyos resultados se reportan en la sección 6.3.

7.2. Síntesis de las conclusiones

La conclusión principal de esta tesis consiste en haber establecido la validez de la hipótesis de trabajo que se adoptó en la sección 3.3 sobre la definición del problema y que repetimos aquí.

Ninguna distribución suave es adecuada para modelar en general todos los atributos temporales que caracterizan la cadencia de tecleo de un usuario; sólo las distribuciones empíricas del perfil intrausuario son capaces de capturar con precisión su comportamiento característico, y este fenómeno puede ser capitalizado tanto para generar muestras sintéticas que logren engañar a los actuales sistemas de autenticación basados en cadencias de tecleo como para construir medidas de defensa eficaces contra ataques de presentación que distingan la escritura del usuario humano legítimo de una muestra construida artificialmente.

Adicionalmente, cada experimento arrojo una conclusión particular dentro de su alcance, que si bien cumple una función determinada en la narrativa general tiene también un valor

propio como resultado y como punto de partida para ulteriores investigaciones. El experimento sobre distribuciones subyacentes nos enseñó que

> *La distribución log-logística es una clara ganadora entre todos los candidatos para ajustar los histogramas de tiempo, tanto de retención como de latencia, producto de las cadencias de tecleo, pero las tasas de rechazo de hipótesis y los méritos relativos de esta y otras distribuciones muestran que un enfoque que considere los histogramas empíricos en su individualidad es promisorio para la autenticación de usuarios*.

De esta forma se motivaron y justificaron tanto las distancias como los métodos de síntesis basados en histogramas empíricos. Al evaluar su rendimiento en un ataque de presentación, se estableció que

> *Las estrategias de síntesis basadas en histogramas empíricos alcanzan un mejor rendimiento que aquellas basadas en distribuciones suaves al intentar engañar a un sistema de autenticación basado en cadencias de tecleo.*

Pero sobre todo se estableció que, al utilizarlas como adversarios en un esquema de contramedidas de defensa, resulta que

> *Las distancias basadas en histogramas empíricos alcanzan un mejor rendimiento que las tradicionales y otros métodos de clasificación al intentar detectar muestras sintetizadas.*

Finalmente, al evaluar una extensión del método anterior descubrimos que

> *Utilizando solo atributos temporales es posible identificar el texto ingresado dentro de una lista de candidatos de tamaño mediano, alcanzando tasas de error muy bajas y competitivas con el estado del arte, aunque tratemos con textos y listas de candidatos más largas.*

7.3. Síntesis de los resultados cuantitativos

Los resultados cuantitativos para los tres experimentos se sintetizan en el cuadro 7.1 y en las secciones siguientes.

Experimento	Resultado	Valor
Distribuciones	Mejor distribución	log-logística
	Mejores coincidencias	50 %-65 %
	Mejores coincidencias	50 %-65 %
Síntesis	Mejores estrategias (interusuario)	NoiseBot y NS/ICDF
	FAR	1 %-2 %
	Mejor estrategia (intrausuario)	ICDF
	FAR	15 %-20 %
Defensa	FAR (interusuario)	1 % - 2 %
	FAR (intrausuario)	15 % - 20 %
	FRR	1 % - 2 %
	Subconjunto seleccionado	60 % dist. hist. emp.
	InfoGain (retención)	0,45
	InfoGain (latencia)	0,25
Identificación	FAR (intrausuario)	< 1 %
	FRR (intrausuario)	< 1 %
	FAR (interusuario)	2 %
	FRR (interusuario)	≈20 %

Cuadro 7.1: Síntesis de los resultados cuantitativos

7.3.1. Experimento sobre distribuciones subyacentes

La distribución log-logística, tanto con dos como con tres parámetros, es una clara ganadora entre todos los candidatos. Ajustando los tiempos de latencia con dos parámetros, su recuento de mejores coincidencias para todos los conjuntos de datos es de alrededor del 50% en LSIA y por encima del 65% en el resto, mientras que el siguiente candidato, la distribución log-normal de dos parámetros, alcanza a lo sumo el 28% en LSIA y menos del 20% en el resto. En cuanto a tiempos de retención, supera siempre a las otras distribuciones por un margen de más del 10%, y usualmente alrededor del 20%, con la excepción de las dos tareas de KM. Con tres parámetros, la distribución log-logística sigue superando al resto en más de un 10%, con un margen más amplio para las latencias.

7.3.2. Experimento sobre síntesis de muestras artificiales y contramedidas de defensa

Cuando el atacante dispone sólo de perfiles interusuario, las mejores estrategias resultan ser NoiseBot y NS/ICDF. El primero alcanza tasas de falsos positivos entre 1% y 2%. Las diferencias entre el ganador y el segundo no son tan notables como en el caso de entrenamiento intrausuario y, teniendo en cuenta los intervalos de confianza superpuestos, no se puede atribuir ninguna significación estadística a aquellos excepto para LSIA, donde

NoiseBot es un claro ganador. ICDF es la estrategia de síntesis más exitosa contra el sistema de defensa propuesto cuando tiene acceso al perfil intrausuario. Alcanza tasas de falsos positivos entre el 15% y el 20%, casi triplicando el rendimiento de la que la sigue en todos los casos.

Cuando se evalúa el método de defensa contra la estrategia de síntesis de mejor rendimiento, las tasas de falsos positivos varían entre 1% y 2% para NoiseBot utilizando perfiles interusuario, pero aumentan entre 15% y 20% si se utiliza ICDF con perfil intrausuario. Por lo tanto, cuando un atacante no tiene acceso a información privilegiada del usuario objetivo, el sistema de detección es muy efectivo. En el peor de los casos, frente a un atacante sofisticado que aprovecha una filtración de datos que haya revelado todas las muestras de los usuarios objetivo, aún podemos esperar del sistema de detección una protección significativa, aunque no optima, contra falsificaciones sintéticas. Las tasas de falsos negativos varían entre 1% y 2%.

Consistentemente en todos los conjuntos de datos, los atributos de distancias basadas en histogramas empíricos aparecen en cada uno de los subconjuntos, y comprenden alrededor del 60% de los atributos seleccionados. Lo que es más, proporcionan casi siempre la mayor ganancia de información, con valores alrededor de 0,25, para los tiempos de latencia y la mejor o la segunda mejor, con valores alrededor de 0,45 para los tiempos de retención.

7.3.3. Experimento sobre identificación del texto ingresado utilizando atributos temporales

El método propuesto logra bajas tasas de error con oraciones y listas más largas que aquellos consideradas previamente en la literatura. Todos los conjuntos de datos puntúan por debajo del 1% en tasas de falsos positivos y negativos cuando se utiliza entrenamiento intrausuario, con la excepción de LSIA donde la primera es ligeramente superior y la última se eleva a alrededor del 5%. Las distribuciones son, sin embargo, de cola larga; algunos de los usuarios con peor desempeño que se encuentran en el último cuartil presentan tasas de error más de un orden de magnitud más altas.

Cuando se utilizan datos de la población general para el entrenamiento en lugar del perfil intrausuario las tasas de falsos positivos aumentan entre dos y tres veces, manteniéndose en torno al 2% con la excepción del conjunto de datos LSIA, mientras que las tasas de falsos negativos se disparan hasta el 20%. Las colas a la derecha de la distribución de usuarios con peores rendimientos también se alargan.

7.4. Aportes

Además de la producción científica derivada de esta tesis, que ha sido enumerada en la sección**??**, los aportes de esta tesis abarcan cuatro métodos, la herramienta integrada de evaluación, y los conjuntos de datos de evaluación y de resultados para los experimentos principales y secundarios.

7.4.1. Métodos

A lo largo de esta tesis se propusieron cuatro métodos como parte de la resolución del problema planteado en el capítulo 3.

- *Distancias basadas en histogramas empíricos*. En lugar de presuponer una cierta distribución suave para los tiempos de retención y latencia y estimar sus parámetros en base al perfil intrausuario, se propuso utilizar la totalidad de las muestras en los histogramas empíricos y calcular las distancias correspondientes en base a sus formas particulares para cada usuario y cada digrama. Las distancias basadas en histogramas empíricos han sido descritas en la sección A.6, y han sido utilizadas en los artículos [109], [152], y [153].
- *Estrategias de síntesis basadas en histogramas empíricos*. Mientras que los métodos del estado del arte presuponen y muestrean distribuciones suaves para generar muestras artificiales de la cadencia de tecleo de un usuario objetivo, se propuso muestrear los histogramas empíricos, generados en base a perfiles intrausuario e interusuario y seleccionados en base a contextos de mayor orden. Las estrategias de síntesis basadas en histogramas empíricos han sido descritas en la sección 4.3 y forman parte del artículo [152].
- *Contramedidas de defensa contra ataques de presentación basadas en estrategias adversarias que utilizan histogramas empíricos*. Las estrategias de síntesis propuestas pueden ser utilizadas como adversarios para entrenar un clasificador que, utilizando las distancias basadas en histogramas empíricos, pueden mitigar un ataque de presentación con una muestra artificial que imita la cadencia de tecleo del usuario objetivo. Las contramedidas de defensa han sido descritas en la sección 4.4 y constituyen el tema principal del artículo [152].
- *Esquema de identificación del texto ingresado basada en estrategias adversarias que utilizan histogramas empíricos*. El método de defensa contra ataques de presentación con muestras artificiales que imita la cadencia de tecleo del usuario objetivo puede ser adaptado, utilizando muestras intrausuario y adversarios con diferentes textos, para identificar el texto ingresado si se cuenta solamente con los tiempos de retención y latencia, y no la secuencia de teclas presionada, como se obtendría de un ataque por canal lateral. El esquema de identificación del texto ingresado ha sido descrito en la sección 4.5 y constituye el tema principal del artículo [153].

7.4.2. Herramientas

Se creo una herramienta integrada que implementa los métodos propuestos en esta tesis. El uso y configuración de la misma se detalla en el apéndice D. La herramienta y el código fuente se pone a disposición del público en el repositorio vinculado al Laboratorio de Sistemas de Información Avanzados, en la dirección

https://github.com/lsia/herramientaGonzalez2021

Por medio de esta herramienta es posible sintetizar una muestra artificial de atributos temporales al seleccionar un perfil de usuario y un texto objetivo. Cualquiera de los conjuntos de datos utilizados para los experimentos de esta tesis (LSIA, KM, y PROSODY, con las

subdivisiones GAY, GUN, y REVIEW) puede ser utilizado como fuente de perfiles intrausuario para la síntesis. Gracias a la arquitectura extensible de la herramienta, es muy sencillo agregar otros conjuntos de datos si estos no son suficientes. La salida del programa consiste en dos vectores de atributos temporales en formato CSV, uno para tiempos de retención y otro para tiempos de latencia.

Dado una muestra de escritura en texto libre, también en formato CSV, la herramienta permite verificar si la misma es una falsificación sintética utilizando los métodos propuestos en esta tesis. Para tal fin deberá especificarse, una vez más, el conjunto de datos al que pertenece el perfil del usuario correspondiente y el nombre del usuario.

Si además de la muestra, que no necesita contener nombres de teclas, se provee una lista de textos candidatos, la herramienta permite identificar si la muestra corresponde a alguno de estos últimos utilizando los métodos propuestos en esta tesis.

La herramienta fue desarrollada en C# y la compilación por defecto es compatible con el sistema operativo Microsoft Windows. Utilizando .NET Core es posible ejecutarla también en sistemas operativos Linux u otros compatibles.

7.4.3. Conjuntos de datos

Los tres experimentos principales de esta tesis produjeron conjuntos de datos de evaluación y de resultados, que se ponen a disponibilidad de la comunidad de investigadores en forma pública y gratuita, tanto en los repositorios de IEEE DataPort como de Mendeley Data.

Los conjuntos de datos para el experimento de distribuciones subyacentes pueden ser descargados de

- Nahuel González. *Dataset of Timing distributions in free text keystroke dynamics profiles*. Ver. 1. Mendeley Data. doi: 10.17632/sjk7kz35nh.1. url: https://data.mendeley.com/datasets/sjk7kz35nh/1 (visitado 04-03-2021)
- Nahuel González. *Dataset of Timing distributions in free text keystroke dynamics profiles*. Ver. 1. IEEE DataPort. doi: 10.21227/ngv9-fa18. url: https://ieeedataport.org/documents/timing-distributions-free-text-keystrokedynamics-profiles (visitado 07-03-2021)

Los conjuntos de datos para el experimento de síntesis de muestras artificiales y contramedidas de defensa pueden ser descargados de

- Nahuel González. *Dataset for Towards Liveness Detection in Keystroke Dynamics: Revealing Synthetic Forgeries*. Ver. 1. Mendeley Data. doi: 10.17632/xvg5j5z29p. 1. url: https://data.mendeley.com/datasets/xvg5j5z29p/1 (visitado 19-05-2021)
- Nahuel González. *Dataset for Towards Liveness Detection in Keystroke Dynamics: Revealing Synthetic Forgeries*. Ver. 1. IEEE DataPort. doi: 10.21227/1ka3-er49. url:

https://ieee-dataport.org/documents/dataset-towards-livenessdetection-keystroke-dynamics-revealing-synthetic-forgeries (visitado 19-05-2021)

Los conjuntos de datos para el experimento de identificación del texto ingresado utilizando parámetros temporales pueden ser descargados de

- Nahuel González. *Dataset for The Reverse Problem of Keystroke Dynamics: Guessing Typed Text with Keystroke Timings*. Ver. 1. Mendeley Data. doi: 10.17632/94dwkbxf2d.1. url: https://data.mendeley.com/datasets/94dwkbxf2d/1 (visitado 22-04-2021)

- Nahuel González. *Dataset for The Reverse Problem of Keystroke Dynamics: Guessing Typed Text with Keystroke Timings*. Ver. 1. IEEE DataPort. doi: 10.21227/7616-7964. url: https://ieee-dataport.org/documents/dataset-reverseproblem-keystroke-dynamics-guessing-typed-text-keystroke-timings (visitado 22-04-2021)

También son aportes de esta tesis los conjuntos de datos que se listan a continuación, que acompañan a los artículos [104], [105], y [154].

- Nahuel González. *Dataset for Exploring internal correlations in timing features of keystroke dynamics at word boundaries and their usage for authentication and identification*. Ver. 1. Mendeley Data. doi: 10.17632/vx83444p8n.1. url: https://data.mendeley.com/datasets/vx83444p8n/1 (visitado 22-02-2021)

- Nahuel González. *Dataset for An Ensemble Method for Keystroke Dynamics Authentication in Free-Text Using Word Boundaries*. Ver. 1. 2021. doi: 10.17632/xvg5j5z29p. 1. url: https://data.mendeley.com/datasets/xvg5j5z29p/1 (visitado 26-07-2021)

- Nahuel González. *Dataset for An Ensemble Method for Keystroke Dynamics Authentication in Free-Text Using Word Boundaries*. Ver. 1. 2021. doi: 10.21227/jdzh-4m97. url: https://ieee-dataport.org/documents/dataset-ensemblemethod-keystroke-dynamics-authentication-free-text-using-wordboundaries (visitado 26-07-2021)

7.5. Futuras líneas de investigación

Se ha determinado en forma cuantitativa en la sección 6.1.2 que la distribución log-logística es la que provee el mejor ajuste. Sin embargo, el motivo que hace de esta distribución la clara ganadora no queda claro, y la literatura del tópico no hace mención a ella. El objetivo de entender los procesos neuromusculares que hacen que los valores de tiempos al escribir se distribuyan como una log-logística puede abrir una nueva línea de

investigación, que la interfaz cerebro-máquina que fue utilizada por miembros del Laboratorio de Sistemas de Información Avanzados [65] puede ayudar a esclarecer.

Hemos visto, al reportar los resultados del experimento sobre síntesis de muestras artificiales en la sección 6.2, que la estrategia NS/ICDF resulta peor que la más sencilla ICDF con entrenamiento intrausuario. Interpretamos este hecho como que la estrategia planteada no captura adecuadamente la no estacionariedad de las series temporales de eventos entre teclas. Los resultados de [105] y [154] apuntan a que estas deben ser tenidas en cuenta a nivel de palabra. Investigar que otra estrategia alternativa puede mejorar la síntesis utilizando variaciones a nivel de palabras es una línea de investigación que se abre.

Atisbamos al inicio del capítulo 6 que los comportamientos individuales conforman un zoológico de idiosincrasias [151] que quizás valga la pena discriminar en un estudio ulterior. La eficacia de las distancias basadas en histogramas empíricos para la tarea incentiva al análisis ulterior de que otros atributos pueden extraerse de la cadencia de tecleo, que caractericen no al proceso motor general sino las especificidades de cada usuario.

7.6. Futuras líneas de trabajo

En la sección 4.2 se ha descrito una adaptación del conteo de valores atípicos de A.7 para utilizar histogramas empíricos. Utilizamos allí un valor fijo de $\sigma = 0,45$ imitando los parámetros de cola para distribuciones gaussianas. Sin embargo, es interesante explorar el efecto que distintos valores para σ pueden tener sobre la calidad del atributo derivado, y la utilidad de presentar varias instancias de conteo de valores atípicos al clasificador.

Entre los atributos derivados basados en medidas de desorden, se ha descrito el índice de direccionalidad en la sección A.9. Hemos notado en la sección 6.2.4 que el incremento del orden de los contextos ayuda a explicar el mejor rendimiento del método de síntesis. Abre una línea de trabajo la pregunta sobre el efecto que puede tener sobre su rendimiento incrementar el orden del índice de direccionalidad.

Para el experimento sobre distribuciones subyacentes, se ha utilizado el criterio de información de Akaike que se describe en la sección 5.4.3. Notamos allí que una interpretación basada en la teoría de la información permite estimar el contenido de información residual de los valores empíricos luego de ajustar en base a la distribución candidata. Llevar adelante un estudio al respecto permitiría explorar los límites de la compresión de datos temporales de cadencias de tecleo en textos libres.

Parte III

Bibliografía y apéndice

Apéndice A

Distancias, métricas, y atributos derivados

A.1. Introducción

En este apéndice se detallan las métricas, distancias, y otros atributos utilizados para la clasificación en los métodos propuestos a lo largo de esta tesis. Aunque sus equivalentes para ICDF han sido motivadas, deducidas, y explicadas en detalle en la sección 4.2, aquí se repiten algunas de las ecuaciones a los fines de ejemplificar el paralelo entre las distancias tradicionales y aquellas basadas en CDF.

La notación general es la misma que se ha definido en 4.1.2 y que ha sido utilizada en 4.3.3 y 4.3.4. En todos los casos se calcularán distancias entre los vectores de tiempo $\mathbf{r} = r_1 \ldots r_n$ y $\mathbf{s} = s_1 \ldots s_n$, para no abusar de la letra t. El subíndice i se utilizará para iterar sobre las distintas componentes de un vector de tiempos o sus teclas correspondientes. S_i será en todos los casos el conjunto devuelto por el modelado con contextos finitos para la tecla k_i, μ_i su media muestral, y σ_i su desvío estándar.

El resto del apéndice está organizado de la siguiente manera. La sección A.2 describe los conceptos algebraicos básicos que se requerirán a continuación, entre ellos normas, distancias, escalado, y normalización. Las secciones A.3, A.4, y A.5 introducen diversas distancias que son utilizadas en los métodos evaluados en esta tesis. La sección A.6 compara en términos generales las anteriores con las distancias basadas en CDF, que han sido propuestas en 4.2. La sección A.7 describe el conteo de valores atípicos, clásico y para CDF. La sección A.8 introduce la distancia R de Bergadano, Gunetti, y Picardi [31], estrella del análisis de cadencias de tecleo en textos libres, y una mejora propuesta para el caso particular en que los vectores temporales a comparar comparten idéntico texto. Finalmente, la sección A.9 fundamenta y describe el índice de direccionalidad.

A.2. Conceptos algebraicos básicos

En esta sección se enumeran sumariamente los conceptos de norma y distancia, junto con sus versiones escaladas y normalizadas, que serán utilizados en las siguientes secciones para definir distancias particulares de interés. El lector interesado en justificar y profundizar estos conceptos puede consultar cualquier libro de álgebra lineal.

A.2. Conceptos algebraicos básicos

A.2.1. Normas

Dado un espacio vectorial *V* sobre un subcuerpo *K* de los números complejos, una *norma* en *V* es una función

$$\rho: V \to R \tag{A.1}$$

que cumple las siguientes tres propiedades para todo $k \in K$ y **u**, **v** $\in V$:

1. *Escalabilidad absoluta*

$$\rho\,(kv) = |k|\,\rho\,(v) \tag{A.2}$$

2. *Desigualdad triangular*

$$\rho\,(u+v) \leq\ \rho\,(u) + \rho\,(v) \tag{A.3}$$

3. *Norma cero del vector nulo*

$$\rho\,(v) =\ 0\ \leftrightarrow v\ 0_v \tag{A.4}$$

Un espacio vectorial conjuntamente con una cierta norma se denomina *espacio normado*. Puede existir más de una norma para un espacio vectorial dado. Si se sobreentiende el espacio normado al que nos estamos refiriendo, la norma de un vector **v** se denota usualmente con dobles barras en la forma ‖**v**‖.

A.2.2. Distancias

Denominamos *distancia* sobre un conjunto *G* a una función

$$d: G\ x\ G \to R \tag{A.5}$$

que cumple las siguientes propiedades para todo **x**, **y**, **z** $\in G$

1. *Positividad*

$$d(x,y) \geq 0, x = y\ \leftrightarrow d(x,y) = 0 \tag{A.6}$$

2. *Simetría*

$$d(x,y) = d(y,x) \tag{A.7}$$

3. *Desigualdad triangular*

$$d(x,z) \leq d(x,y) + d(y,z) \tag{A.8}$$

Un conjunto G y una función de distancia *d* dados constituyen un espacio métrico. En todo espacio normado *V* la norma $\| \|_V$ induce una función de distancia

$$d\ (x, y) = \|x - y\|_V$$

que cumple las tres condiciones indicadas más arriba (la demostración es trivial).

A.2.3. Distancias normalizadas y escaladas

Sea *V* un espacio normado unidimensional con norma $\| \|_V$, y *d* la correspondiente distancia inducida. Sea $W = V^n$ un espacio vectorial *n*-dimensional, y $\mathbf{x} = x_1 \ldots x_n$ e $\mathbf{y} = y_1 \ldots y_n$ dos vectores de W. La *distancia escalada* en *W* inducida por *V* es una función

$$d: W\ x\ W \rightarrow R \tag{A.9}$$

de la forma

$$d\ (x, y) = f\left(\frac{1}{n}\sum_{i=1}^{n}\|x_i - y_i\|_V\right)$$

y que cumple las condiciones A.6, A.7, y A.8.

Esta definición está motivada por el intento de eliminar la dependencia de la magnitud esperada de una distancia respecto de la dimensión del espacio vectorial subyacente, con el objetivo de volver conmensurables los resultados obtenidos con vectores de distinta cantidad de componentes. Por ejemplo, podemos afirmar que cualesquiera dos vectores de muestras de una variable aleatoria de varianza uno y cuya distancia L_1 normalizada se encuentra en torno a uno son suficientemente parecidos, independientemente de su longitud.

Cuando las componentes de los vectores provienen del muestreo de variables aleatorias, es usual normalizar los sumandos con, por ejemplo, el desvío estándar. En general, si A_i es el factor de normalización de la *i*-esima componente, una distancia normalizada y escalada es una función de la forma

$$d\ (x, y) = f\left(\frac{1}{n}\sum_{i=1}^{n}\frac{\|x_i - y_i\|_V}{A_i}\right)$$

y que una vez más cumple las condiciones A.6, A.7, y A.8.

Así como al escalar intentamos eliminar la dependencia respecto de la dimensión del espacio vectorial subyacente, al normalizar intentamos eliminar la dependencia respecto de las distintas magnitudes de los componentes.

A.3. Distancias de Minkowski, Manhattan, y euclídea

La distancia de Minkowski [158] de orden *p* tiene la forma

$$d(r,s) = \left(\sum_{i=1}^{n} |r_i - s_i|^p \right)^{\frac{1}{p}} \tag{A.10}$$

cumple las condiciones A.6, A.7, y A.8 siempre y cuando $p \geq 1$. Cuando $p < 1$ falla la desigualdad triangular, por lo que no es estrictamente una distancia. Sin embargo, como esta propiedad no es crucial para los resultados que se presentan, sigue siendo útil para cuantificar con un escalar cuanto difieren dos vectores de tiempos. En particular, $p = 0,4$ minimiza el EER al ser utilizada para la verificación de usuarios [35]; a costa de una ligera imprecisión que habilita un tratamiento uniforme, la denominaremos distancia, en cualquier caso.

Mediante dos sencillas modificaciones a la ecuación de arriba obtenemos la distancia escalada y normalizada que se ha utilizado a lo largo de esta tesis, en la forma

$$d(r,s) = \left(\sum_{i=1}^{n} \left| \frac{r_i - s_i}{\sigma_i} \right|^p \right)^{\frac{1}{p}} \tag{A.11}$$

En donde los factores de escala σ_i son los desvíos estándar de los conjuntos $|S_i|$. Especializando $p = 1$ obtenemos la versión escalada y normalizada de la distancia de Manhattan [159].

$$\mathcal{L}_1(r,s) = \sum_{i=1}^{n} \left| \frac{r_i - s_i}{\sigma_i} \right| \tag{A.12}$$

Especializando $p = 2$ obtenemos la versión escalada y normalizada de la distancia euclídea [160].

$$\mathcal{L}_2(r,s) = \sqrt{\sum_{i=1}^{n} \left(\frac{r_i - s_i}{\sigma_i} \right)^2} \tag{A.13}$$

Otros valores de *p* producen distancias con variadas utilidades. Para la verificación de usuarios, $p = 0,4$ resulta ser el valor óptimo de la distancia de Minkowski [35].

$$M(r,s) = \left(\frac{1}{n} \sum_{i=1}^{n} \left| \frac{r_i - s_i}{\sigma_i} \right|^{0,4} \right)^{\frac{1}{0,4}} \tag{A.14}$$

Valores de *p* mayores que uno penalizan más las desviaciones más grandes, mientras que valores menores potencian el efecto de las desviaciones pequeñas. La figura A.1 muestra gráficamente las circunferencias unitarias de acuerdo a las distancias para distintos valores de *p*.

A.4. Distancia de Canberra

Utilizando factores de normalización iguales a

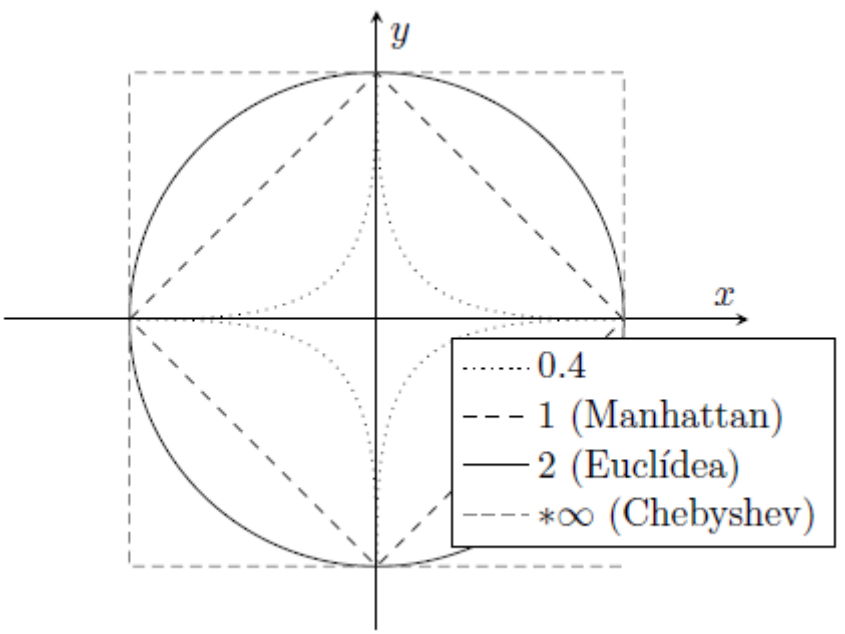

Figura A.1: Circunferencias unitarias según distintas normas. Gentileza de [64]

$$A_i = |r_i| + |s_i| \tag{A.15}$$

y un valor de *p* = 1 en la ecuación A.10, obtenemos la expresión de la distancia de Canberra [161, 162], en la forma

$$C(r,s) = \frac{1}{n}\sum_{i=1}^{n}\frac{|r_i - s_i|}{|r_i| + |s_i|} \tag{A.16}$$

Se observa que corresponde a la distancia (escalada y normalizada) de Manhattan, en donde el *i*-ésimo factor de normalización es igual a la suma de las magnitudes de las *i*-ésimas componentes. Fue propuesta por Lance y Williams [162].

A.5. Otras distancias

Sin salir de la forma general de A.10, podemos mencionar la distancia de Chebyshev [163] que se obtiene al hacer tender *p* hacia el infinito positivo. Cada valor de *p* en A.10 corresponde a otra distancia, y muchísimas otras que no siguen esa forma general han sido propuestas, estudiadas, evaluadas, y empleadas con mayor o menor éxito en la literatura de cadencias de tecleo, como la métrica A [31], o de otros tópicos. Las aquí reseñadas y utilizadas en el transcurso de esta tesis para alimentar los clasificadores cumplen con el requisito de ser bien conocidas y haber demostrado su utilidad en estudios previos del tema. Una multitud adicional de candidatas debe, necesariamente, quedar fuera del alcance de este estudio.

A.6. Distancias basadas en CDF

Como $0 \leq F_{S_i}(r_i) \leq 1$ (ver sección 4.2), no se requiere normalización al utilizar las distancias anteriores con CFD. Las expresiones resultan ser idénticas a las de L1, L2, M, y Canberra, salvo que el i-ésimo componente de cada vector siendo comparado es primero evaluado por su correspondiente FSi. Así, la versión CDF de la distancia de Manhattan

$$d(r,s) = \frac{1}{n}\sum_{i=1}^{n}\left|F_{S_i}(r_i) - (s_i)\right| \tag{A.17}$$

es igual a la de la ecuación A.12, con todos los factores de normalización igual a uno y las componentes r_i y s_i reemplazadas por $F_{Si}(r_i)$ y $F_{Si}(s_i)$. El patrón es idéntico en todas las demás distancias basadas en CFD y no se repetirá innecesariamente.

A.7. Conteo de valores atípicos o Z-score

El conteo de valores atípicos es una puntuación muy simple que indica la cantidad de valores de los parámetros temporales que difieren de la media en más de una cierta cota. Cuando asumimos una variable subyacente gaussiana esta cota se mide en desvíos estándar, pero cuando operamos con otras distribuciones o implementamos una versión de Z-score basada en CDF el significado de este parámetro no es tan inmediato.

Formalmente, dada una cota σ y un vector $\mathbf{t} = t_1 \ldots t_n$, el conteo de valores atípicos resulta ser

$$Z(t) = \frac{1}{n}\sum_{i=1}^{n} u\left(|t_i - \mu_i| - \sigma\right) \tag{A.18}$$

En donde $u(x)$ es la función escalón, que es cero para $t < 0$ y uno para $t \geq 0$, y μ_i es el valor esperado del conjunto S_i. A pesar de su simplicidad permite obtener resultados similares a muchos clasificadores más complejos [5, 49].

Como se ha descrito en 4.2, al tratarse de Z-score basado en CDF el parámetro σ no representa desvíos estándar sino probabilidad de cola. También se simplifica la expresión pues

$$\mathbb{E}\left[F_{S_i}\right] = 0{,}5 \tag{A.19}$$

de donde la expresión para Z-score basado en CDF resulta

$$Z(t) = \frac{1}{n}\sum_{i=1}^{n} u\left(\left|F_{S_i}(t_i) - 0{,}5\right| - \sigma\right) \tag{A.20}$$

Los conteos de valores atípicos son métricas para un único vector, pero pueden convertirse en una distancia tomando el valor absoluto de la resta entre los correspondientes Z de los vectores comparados, en la forma

$$d(r,s) = |Z(r) - Z(s)| \tag{A.21}$$

La expresión es idéntica para Z-score tradicional y basado en CDF.

A.8. R

La distancia R fue inicialmente propuesta por Bergadano, Gunetti y Picardi en [31] y [8] para su utilización en la verificación de usuarios por medio de cadencias de tecleo en textos libres. Al igual que el caso de las distancias de Minkowski con $p < 1$, R no cumple la desigualdad triangular por lo que, una vez más, no es estrictamente una distancia en el sentido formal, pero es practico tratarla como si lo fuera. Ha debido ser optimizada para su ejecución en tiempo real [52], ya que su costo computacional no es irrelevante.

En contraste con otras distancias, la distancia R no utiliza los valores temporales sino el orden relativo de estos en todos los digramas simultáneamente. Así, si el usuario varía su velocidad promedio de escritura sin alterar significativamente las relaciones temporales entre teclas, el valor de la distancia R presentara escasa variación. Lo que, es más, el puntaje otorgado por la distancia R es global, en el sentido de que pequeñas variaciones locales pueden propagarse y generar grandes diferencias en la valuación final.

Dado un vector de tiempos $\mathbf{t} = t_1 \ldots t_n$, se generan todos los pares ordenados distintos de digramas (k_{i-1}, k_i), en donde $k_{i-1}, k_i \in \mathbf{k} = k_1 \ldots k_n$. Para cada par ordenado distinto (k_{i-1}, k_i) se genera el conjunto $S_t(k_{i-1}, k_i)$ de todos los tiempos $t_j \in \mathbf{t} = t_1 \ldots t_n$ tales que $k_j = k_i$ y $k_{j-1} = k_{i-1}$. Dichos valores se promedian, para obtener

$$\mu_t(k_{i-1}, k_i) = \frac{1}{n} \sum_{t_i \in S_t(k_{i-1}, k_i)} t_j \tag{A.22}$$

Para calcular una distancia entre vectores **r** y **s**, se extraen los digramas (k_{i-1}, k_i) y se agrupan en respectivos vectores. Luego, se calculan sus escalares μ_r y μ_s y los vectores se ordenan en base a los mismos. Finalmente, se eliminan todos aquellos digramas que no sean comunes a ambos. Obtenemos

$$R = (k_{i_1-1}, k_{i_1}) \dots (k_{i_{m-1}}, k_{i_m}) \tag{A.23}$$

$$S = (k_{j_1-1}, k_{j_1}) \dots (k_{j_{m-1}}, k_{j_m}) \tag{A.24}$$

en donde *m* es la cantidad de digramas (k_{i-1}, k_i) comunes a ambos vectores, y para todos ellos

$$\mu_r \left(k_{i_j-1}, k_{i_j}\right) < \mu_r \left(k_{i_{j+1}-1}, k_{i_{j+1}}\right) \tag{A.25}$$

$$\mu_s \left(k_{i_j-1}, k_{i_j}\right) < \mu_s \left(k_{i_{j+1}-1}, k_{i_{j+1}}\right) \tag{A.26}$$

Para medir el *grado de desorden normalizado* entre **R** y **S**, llamemos *γ* a la permutación que lleva los elementos de uno al otro, de manera tal que

$$R = v_1 \dots v_n \tag{A.27}$$

$$S = v_{\gamma(1)} \dots v_{\gamma(n)} \tag{A.28}$$

Patrón	
cl	202ms
la	211ms
av	187ms
ve	153ms

Observación	
cl	192ms
la	224ms
av	177ms
ve	193ms

Patrón ordenado	
la	211ms
cl	202ms
av	187ms
ve	153ms

Observación ordenada	
la	224ms
ve	193ms
cl	192ms
av	177ms

Cuadro A.1: Ejemplo de grado de desorden

El grado de desorden del vector **R** en referencia al vector **S** se define como la suma de las distancias absolutas en las posiciones de todos los elementos; formalmente

$$d(\boldsymbol{R}, \boldsymbol{S}) = \sum_{i=1}^{n} |\gamma\ (i) - i| \tag{A.29}$$

Claramente, *d* es simétrica y si **R** = **S** tenemos que

$$d(\boldsymbol{R}, \boldsymbol{S}) = 0 \tag{A.30}$$

Se puede demostrar que *d* alcanza el valor máximo para vectores de *n* dimensiones cuando ambos son imágenes especulares o, lo que es lo mismo, cuando

$$\sigma(i) = n + 1 - i \tag{A.31}$$

Denominando a este valor D_{max}^{n} , se define la distancia R, que resulta ser un valor entre cero y uno, como

$$R(\boldsymbol{r}, \boldsymbol{s}) = \frac{d\,(\boldsymbol{R}, \boldsymbol{S})}{D_{max}^{n}} \tag{A.32}$$

Las permutaciones aleatorias se concentran en torno a valores de R entre 0,5 y 0,9, de manera más pronunciada a medida que crece*n*.

La evaluación empírica muestra que distintos textos ingresados por el mismo usuario producen valores del grado de desorden relativo consistentemente pequeños en comparación con los de otro usuario, aun descontando factores de diversidad como el largo de las particiones, el texto ingresado o el idioma. Los resultados de la clasificación no solamente son excelentes, con un FAR del orden del 5% a pesar de las dificultades adicionales, sino que además se logran con conjuntos de entrenamiento pequeños [8] en comparación con otros métodos.

Un ejemplo de evaluación del grado de desorden utilizando los digramas de dos ingresos de la palabra *clave* puede verse en el cuadro A.1. Allí, *patrón* y *observación* corresponden a los vectores de tiempos entre teclas **r** y **s** a ser comparados. Los cuatro digramas de la palabra *clave* son **cl**, **la**, **av**, y **ve**; como ninguno de ellos se repite, no es necesario promediarlos para calcular los μ_r y μ_s, por lo que tenemos

$$\boldsymbol{R} = (l, a)\ (c, l)\ (a, v)\ (v, e) \tag{A.33}$$

$$\boldsymbol{S} = (l, a)\ (v, e)\ (c, l)\ (a, v) \tag{A.34}$$

y el grado de desorden resulta ser

$$d(\boldsymbol{R}, \boldsymbol{S}) = |1 - 1| + |3 - 2| + |4 - 3| + |2 - 4| = 4 \tag{A.35}$$

El valor máximo de *d* que puede alcanzarse con cuatro componentes corresponde a invertir el vector, por lo que

$$D^4_{max} = |4-1| + |3-2| + |2-3| + |1-4| = 7 \tag{A.36}$$

y finalmente

$$R(\boldsymbol{r}, \boldsymbol{s}) = \frac{d(\boldsymbol{R}, \boldsymbol{S})}{D^4_{max}} = \frac{4}{7} \tag{A.37}$$

Este valor, superior a 0,5, nos indica que estamos probablemente frente a un impostor, pues se encuentra en el rango correspondiente a una permutación aleatoria.

A.8.1. R_1 y R_{all}

La distancia R permite comparar vectores de tiempos que corresponden a muestras temporales con distinto texto. Esta versatilidad es deseable al verificar la identidad de usuarios, pues no podemos esperar que el texto libre presente muestras idénticas con mucha frecuencia.

Sin embargo, las particularidades de los métodos que se exploran en esta tesis hacen que, por uno u otro motivo, siempre nos encontremos en una situación en la que queremos comparar instancias de idéntico texto. Ya sea porque estamos utilizando a robot malo y robot bueno como adversarios, porque queremos comprobar sus muestras sintetizadas a las del usuario legítimo, o porque presuponemos un texto candidato luego de un ataque por canal lateral, podremos asumir que **r** y **s** contienen exactamente los mismos digramas y en el mismo orden.

Nada nos impide calcular sobre ellos la distancia R como ha sido descrita, pero esta situación nos permite ir más lejos y lograr más precisión. En lugar de promediar las múltiples observaciones temporales de un digrama cuando este se repite en la secuencia **k** = $k_1 \ldots k_n$, como indica la ecuación A.22, podemos armar los vectores **R** y **S** con todos los digramas originales. En general, esto no es factible cuando los textos de **r** y **s** no se corresponden.

A modo de ejemplo, consideremos que tanto **r** como **s** son vectores de tiempos entre teclas para la palabra *barbaridad*. Utilizando el método de la sección anterior, que agrupa digramas, tendremos que tanto **R** como **S** serán permutaciones de

$$(b,a)\ (a,r)\ (r,b)\ (r,i)\ (i,d)\ (d,a)\ (a,d) \tag{A.38}$$

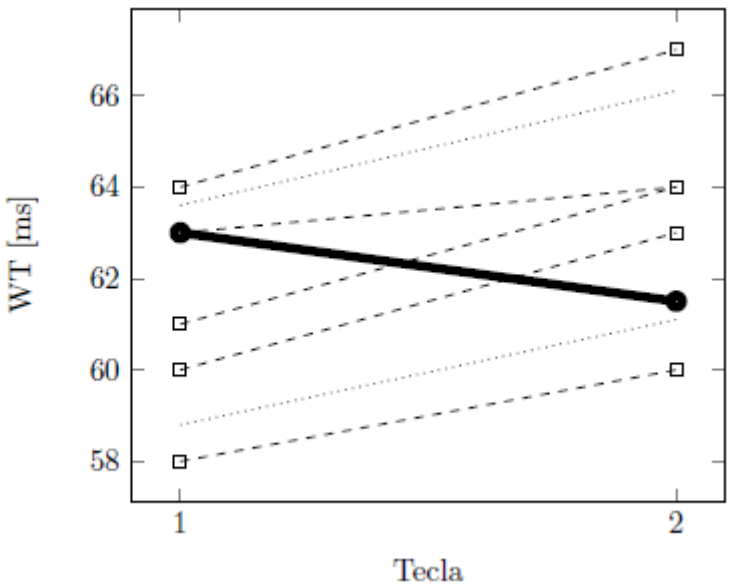

Figura A.2: Ejemplo de entrenamiento con índice de direccionalidad

y que el valor de μ_r o μ_s cada uno de los primeros tres digramas será el promedio de dos muestras, la de la primera y la de la segunda sílaba repetida *bar*. Sin promediar, debemos distinguir ambas para que nos queden dos permutaciones de

$$(b_1, a_1)\ (a_1, r_1)\ (r_1, b_1)\ (b_2, a_2)\ (a_2, r_2)\ (r, i)\ (i, d), (d, a)\ (a, d) \tag{A.39}$$

Denominaremos R_1 a la distancia resultante de promediar valores en digramas repetidos, que pueden calcularse entre vectores temporales de dos textos cualesquiera, y R_{all} a la distancia resultante de considerar todos los digramas, que solo puede calcularse cuando ambos vectores temporales comparten idéntico texto.

A.9. Índice de direccionalidad

En la figura A.2 se muestran, en línea rayada, cinco observaciones de entrenamiento para un digrama que unen los valores capturados de latencia entre teclas sucesivas, señalados con un cuadrado; la línea punteada indica el rango de un desvío estándar respecto del promedio. Considérese la nueva observación a analizar, que se muestra en trazo grueso, cuyos valores de latencia medidos son 63ms y 61,5ms. Para cualquier método de clasificación basado en distancias, la misma se encontraría íntegramente dentro del rango de aceptación, ya que se encuentra a menos de un desvío estándar del tiempo promedio para ambas teclas; lo que, es más, se encuentra más cerca que las dos observaciones de entrenamiento más lejanas.

Sin embargo, existe una característica compartida por todas las observaciones de entrenamiento que esta última no comparte: en cada observación la latencia de la primer tecla es estrictamente mayor que la de la segunda. La particularidad de una cierta tecla de

presentar un valor dado de cierto parámetro característico consistentemente mayor o menor que el de su antecesora bajo un cierto contexto, que será denominada *direccionalidad*, es una de los parámetros distintivos del usuario que se observan en la cadencia de tecleo y que permiten inferir su identidad. La consideración única de la distancia de una observación a un vector patrón no la tiene en cuenta, por lo que se requieren otras técnicas para incluir la información aportada en el análisis de la identidad del usuario.

Es claro que no todas las teclas, y no en todos los contextos, presentaran una direccionalidad claramente definida y que dicha tendencia a presentar valores inferiores o superiores a su antecedente nunca será determinista sino a lo sumo de alta probabilidad. Sin embargo, el rasgo de direccionalidad suele ser bastante estable para un usuario dado, y tanto más para contextos de orden más elevado.

El cálculo del índice de direccionalidad demanda un poco más del modelado por contextos finitos que todas las distancias y métricas anteriores. Para este propósito, asumiremos que el método en lugar de entregar un conjunto S_i de valores temporales t_j, extrae el conjunto P_i de pares de valores temporales (t'_j, t_j) tales que, para cada aparición de la tecla k_i en el perfil del usuario precedida por el contexto de mejor coincidencia $k_{i-m} \ldots k_{i-1}$ de orden m, t_j es el parámetro temporal correspondiente a k_i y t'_j es el parámetro temporal correspondiente a k_{i-1}. En simples palabras, no solo extraemos los tiempos observados para una tecla, sino también los de la tecla anterior; siempre atendiendo al contexto, claro está. Nótese que esta definición obliga a que el orden mínimo del contexto sea uno, pues un contexto de orden cero no es capaz de entregar valores temporales para la tecla anterior.

Ahora, en base a los conjuntos P_i podemos cuantificar la direccionalidad esperada en la tecla k_i contando cuantas veces en el perfil del usuario su tiempo excede al de la tecla anterior y cuantas veces ocurre lo contrario. Definamos el peso direccional $w(i)$ esperado para la tecla k_i en la forma

$$w(i) = \frac{1}{|P_i|} \sum_{(t'_j, t_j) \in P_i} u(t_j - t'_j) - u(t'_j - t_j) \tag{A.40}$$

en donde u(x) es la función escalón. El primer término de la sumatoria agrega uno toda vez que

$$t_j - t'_j > 0 \tag{A.41}$$

mientras que el segundo agrega uno siempre que

$$t'_j - t_j > 0 \tag{A.42}$$

lo que permite escribir una forma simplificada de la ecuación A.40 en la forma

A.9. Índice de direccionalidad

$$w(i) = \frac{\#\{(t'_j, t_j) | t_j > t'_j\} - \#\{(t'_j, t_j) | t'_j > t_j\}}{|P_i|} \quad \text{(A.43)}$$

Por definición, los valores de $w(i)$ se encuentran entre uno y menos uno, dependiendo de la proporción de ocasiones en las que t_j excede a t'_j. Utilizando esta ponderación, podemos definir una métrica direccional, escalada y normalizada, para´ el vector **t** = $t_1 \ldots t_n$ en la forma

$$D(i) = \frac{1}{n} \sum_{i=2}^{n} w(i).[u(\Delta t_i) - u(-\Delta t_i)] \quad \text{(A.44)}$$

en donde

$$\Delta t_i = t(i) - t(i-1) \quad \text{(A.45)}$$

La sumatoria de la ecuación A.44 agrega $w(i)$ si $t_i > t_{i-1}$ y resta $w(i)$ en caso contrario. Denominamos a $D(i)$ como el *índice de direccionalidad* del vector **t**.

Al igual que con la métrica Z-score, podemos convertir los índices de direccionalidad de dos vectores en una distancia al tomar el valor absoluto de la diferencia.

$$d(\boldsymbol{r}, \boldsymbol{s}) = |D(\boldsymbol{r}) - D(\boldsymbol{s})| \quad \text{(A.46)}$$

En contraste con las distancias de Manhattan, euclídea, etc., el índice de direccionalidad representa una métrica relativa, que al igual que R puede mantenerse inalterada, aunque el usuario modifique su velocidad promedio de escritura. A diferencia de R, el índice de direccionalidad es una métrica local, ya que considera sólo el comportamiento de teclas sucesivas.

La demora unitaria del operador Δ en la ecuación A.45 puede incrementarse para producir una familia de índices direccionales de distinto orden. No se perseguirá la posibilidad aquí, quedando esta relegada a las futuras líneas de investigación.

Apéndice B

Bibliometría

La bibliometría es la ciencia que utiliza métodos estadísticos para analizar publicaciones, y hablamos de cienciometría cuando nos restringimos a artículos científicos. En particular, el análisis de citas es el método de uso más extendido de esta última.

En este apéndice se describen los métodos y resultados bibliométricos obtenidos durante la revisión sistemática de la literatura sobre cadencias de tecleo en la que se basa la sección 2.4 sobre el estado del arte. El resto del capítulo está organizado como se describe a continuación. La sección B.1 describe las bases de datos académicas consultadas y la herramienta de consulta. La sección B.2 describe la metodología de consulta. La sección B.3 proporciona un análisis cuantitativo de las publicaciones anuales. Finalmente, la sección B.4 destaca los autores y publicaciones más influyentes, todas los cuales han sido tratadas en distintos apartados de la sección sobre el estado del arte.

B.1. Materiales y herramientas

La herramienta de software *Publish or Perish 7* [164] permite realizar e integrar búsquedas sobre distintas bases de datos académicas en un único conjunto de resultados. Una vez exportada la salida a archivos CSV, cualquier herramienta de análisis puede ser utilizada para agrupar los resultados y extraer la información relevante.

Publish or Perish 7 se encuentra disponible en forma pública y gratuita, y cuenta con versiones para Windows, Linux, y MacOS. Para este estudio se ha empleado la versión para Windows.

B.1.1. Bases de datos consultadas

Entre aquellas que ofrece la herramienta Publish or Perish, para este trabajo se consideraron las siguientes bases de datos:

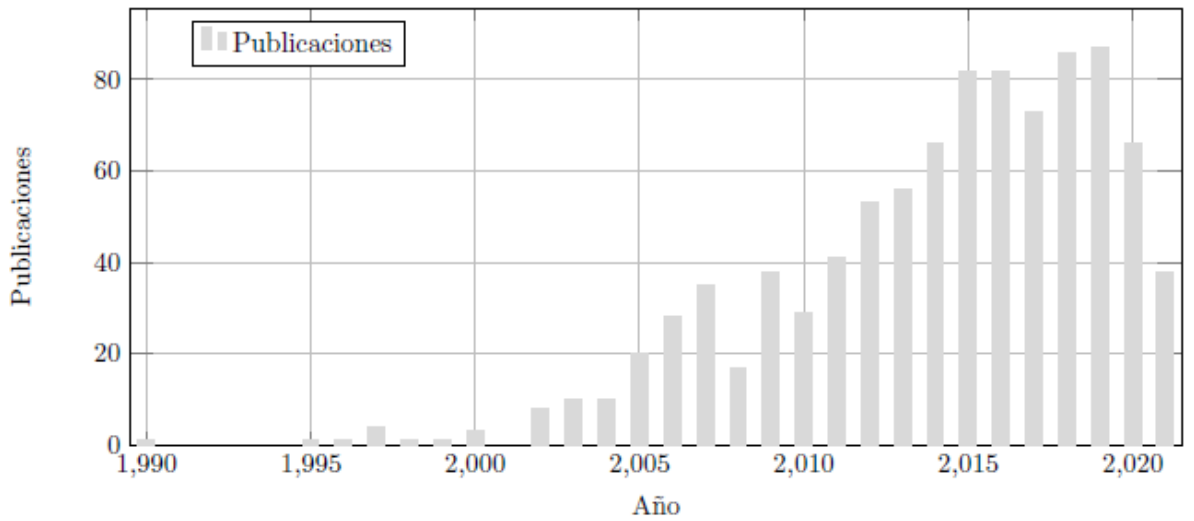

Figura B.1: Conteo de publicaciones por año

- **Google Scholar** es una especialización de los servicios de búsqueda de Google a publicaciones académicas y de otros tipos, como patentes y resoluciones judiciales. De acceso libre y gratuito a través de la web, provee también acceso a consultas estructuradas a través de una API. Los métodos para la determinación de la relevancia de los artículos pueden considerarse los más robustos entre las bases de datos aquí enumeradas.

- **Microsoft Academic** es la competencia a Google Scholar lanzada por Microsoft Research. En contraste con el anterior, no se limita a palabras claves, sino que emplea también tecnologías de análisis semántico. Al año 2021, indexa más de 260 millones de publicaciones, de las cuales 88 millones son artículos científicos y permite consultas a través de una API REST. Microsoft ha anunciado que los servicios serán discontinuados a fines del 2021.

- **Scopus** es una base de datos académica propietaria de Elsevier, que incluye resúmenes y citas de unos 35.000 jornales con revisión de pares.

Se opto por no utilizar PubMed ya que el área de interés de esta tesis no se intercepta con el alcance de esta base de datos. Asimismo, Google Scholar Profiles no se utilizó para las búsquedas pues el análisis se concentró en las publicaciones y no en los autores.

B.2. Metodología de consulta

En los albores de la disciplina del análisis de cadencias de tecleo, la denominación utilizada para referirse a ella no disponía de un término estándar. Se han utilizado denominaciones como *keystroke authentication*, *typing dynamics*, *keyboard biometrics*, y muchas otras variaciones. En el año 1990, el artículo de Bleha *et al*. [42] establece la denominación

keystroke dynamics, que aproximadamente a partir del año 2000 se vuelve estándar en detrimento de las alternativas.

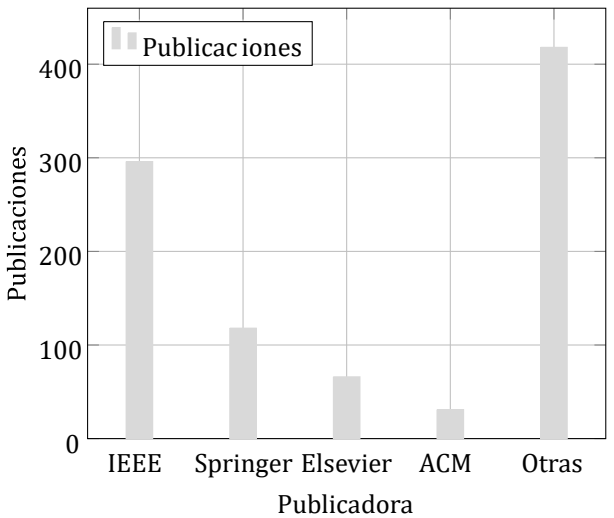

Figura B.2: Conteo de publicaciones para las principales publicadoras

Utilizamos esta denominación en forma exclusiva para consultar las bases de datos, aunque corramos el riesgo de ignorar algunos artículos. De esta forma se simplifica notoriamente la exposición y no perdemos de vista una porción significativa de la literatura del tema, pues previamente al año 2000 existen muy pocos artículos que traten sobre el análisis de cadencias de tecleo. La mayoría de ellos han sido reseñados individualmente en la sección 2.3 sobre la perspectiva histórica de la disciplina.

La pregunta sobre como extraer conocimiento relevante en la era de las vastas, inabarcables, cantidades de información siempre se encuentra presente. Aquí nos hemos concentrado en los mil artículos más relevantes según una consolidación de las tres bases de datos consultadas y no en el cuerpo completo de la literatura del tema. Llevar a cabo esto último sería imposible.

B.3. Publicaciones por año

En la figura B.1 se representa la cantidad de publicaciones anuales dentro del tópico tratado. Se observa una tendencia linealmente creciente desde el año 2000 en adelante hasta aproximadamente el año 2015, en donde se alcanza una meseta en torno a las 80 publicaciones anuales. Aunque aparente existir una disminución posterior en el interés del tópico, el gráfico es ligeramente engañoso. La consulta fue realizada en julio de 2021, por lo que la cantidad de artículos representada para ese año es cerca de la mitad que para los otros.

B.4. Publicaciones y autores más influyentes

En la figura B.2 se muestra la cantidad de publicaciones dentro del conjunto de mayor relevancia, agrupadas por publicadora. En primer lugar, IEEE cuenta con casi el 30% de las publicaciones del tema, seguida por Springer y Elsevier con poco más y menos del 10%, y finalmente ACM con un 5%. Todas las otras publicadoras de

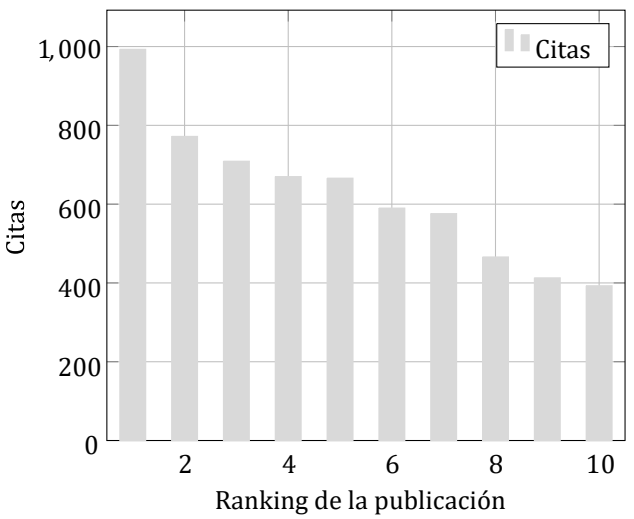

Figura B.3: Conteo de citas para las diez publicaciones más relevantes

menor importancia agrupa el 45% restante.

Las diez publicaciones de mayor influencia, medida en cantidad de citas, son las siguientes. La cantidad de citas que acumulan puede verse en la figura B.3.

- Fabian Monrose y Aviel D Rubin. Keystroke dynamics as a biometric for authentication. En: *Future Generation computer systems* 16.4 (2000), págs. 351-359 [165]
- Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic y Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. En: *IEEE transactions on information forensics and security* 8.1 (2012), págs. 136-148 [88]
- Fabian Monrose, Michael K Reiter y Susanne Wetzel. Password hardening based on keystroke dynamics. En: *International journal of Information security* 1.2 (2002), págs. 69-83 [70]
- Francesco Bergadano, Daniele Gunetti y Claudia Picardi. User authentication through keystroke dynamics. En: *ACM Transactions on Information and System Security (TISSEC)* 5.4 (2002), págs. 367-397 [31]
- Fabian Monrose y Aviel Rubin. Authentication via keystroke dynamics. En: *Proceedings of the 4th ACM conference on Computer and communications security*. ACM. 1997, págs. 48-56 [50]

B.4. Publicaciones y autores más influyentes

- Kevin S Killourhy y Roy A Maxion. Comparing anomaly-detection algorithms for keystroke dynamics*)*. En: *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*. IEEE. 2009, págs. 125-134 [22]
- Daniele Gunetti y Claudia Picardi. Keystroke analysis of free text. En: *ACM Transactions on Information and System Security (TISSEC)* 8.3 (2005), págs. 312-347 [8]
- Charles Slivinsky y Bassam Hussien Saleh Bleha. Computer-access security systems using keystroke dynamics. en. En: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 12.12 (1990), págs. 1217-1222. url: http://lsia.fi.uba.ar/papers/bleha90.pdf [42]
- Balqies Obaidat Mohammad S y Sadoun. Verification of computer users using keystroke dynamics. En: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 27.2 (1997), págs. 261-269 [48]
- Salil P Banerjee y Damon L Woodard. Biometric authentication and identification using keystroke dynamics: A survey. En: *Journal of Pattern Recognition Research* 7.1 (2012), págs. 116-139 [18]

Es interesante notar que tres de ellas corresponden a Fabian Monrose y colaboradores. Obsérvese también que, de estas diez, ocho de ellas tratan de la verificación de claves y solo dos del análisis de textos libres. Se confirma así por medios bibliométricos la laguna en la literatura del tema que es señalada en la síntesis del estado del arte de la sección 2.6.

Apéndice C

Tablas de resultados detallados para el experimento sobre distribuciones subyacentes

A continuación, se listan las tablas con los resultados detallados del experimento sobre distribuciones subyacentes que ha sido descrito en 6.1. Debido a su extensión horizontal, las mismas se encuentran en formato apaisado.

El resto del espacio en blanco en esta página es intencional.

B.4. Publicaciones y autores más influyentes

Dataset	Tarea	llogis	lnorm	gumbel	gamma	weibull	logis	norm
LSIA	Free text	22.51 %	12.07 %	26.85 %	11.36 %	8.52 %	12.29 %	6.39 %
KM	Free text	24.59 %	7.03 %	14.05 %	8.67 %	7.49 %	29.74 %	8.43 % 10 %
	Transcription	19.76 %	6.9 %	16.43 %	9.29 %	12.86 %	24.76 %	
PROSODY GAY	Copy 1	35.65 %	10.02 %	18.26 %	7.01 % 7.77	11.49 %	11.63 %	5.94 %
	Copy 2	35.3 %	10.18 %	17.92 %	% 8.33 %	10.45 %	12.55 %	5.83 %
	Fake Essay	33.98 %	10.67 %	17.4 %	9.08 %	10.76 %	12.74 %	6.12 %
	True Essay	35.81 %	9.36 %	17.04 %		10.19 %	12.91 %	5.63 %
PROSODY GAY	Copy 1	33.72 %	9.08 % 8.91	17.53 %	8.25 % 8.69	11.91 %	13.27 %	6.23 %
	Copy 2	33.44 % 34	%	18.24 %	%	11.55 %	13.36 %	5.82 %
	Fake Essay	%	9.23 %	17.31 %	8.72 %	11.66 %	13.57 %	5.51 %
	True Essay	34.54 %	8.78 %	15.97 %	8.99 %	11.09 %	14.11 %	6.52 %
PROSODY GUN	Copy 1	33.04 %	9.52 %	18.6 %	8.49 % 9.43	12.38 %	11.79 %	6.18 %
	Copy 2	32.08 %	10.16 %	18.44 %	%	12.78 %	11.04 %	6.07 %
PROSODY REVIEW	Fake Review	31.97 %	10.4 %	16.61 %	9.91 %	12.27 %	13.01 %	5.83 %
	True Review	32.62 %	10.01 %	17.17 %	9.97 %	11.98 %	12.4 %	5.85 %

Cuadro C.1: Merito de distribuciones candidatas para tiempos de retención ordenada por conteo de mejor ajuste, para distribuciones de dos parámetros

Dataset	Tarea	llogis	lnorm	gumbel	gamma	weibull	logis	norm
LSIA	Free text	5.62 %	16.22 %	21.84 %	31.27 %	38.5 %	36.52 %	62.02 %
KM	Free text	15.69 %	43.09 %	60.66 %	68.77 %	79.86 %	77.05 %	91.57 %
	Transcription	4.29 %	23.1 %	25.71 %	47.38 %	64.76 %	53.81 %	79.05 %
PROSODY GAY	Copy 1	0.74 % 0.98 % 1.07 %	8.39 % 8.38 %	14.73 %	31.85 %	37.46 %	28.87 %	59.02 %
	Copy 2	1.78 %	9.05 %	14.84 %	31.5 %	36.68 %	28.37 %	59.48 %
	Fake Essay		11.89 %	22.02 %	38.72 %	42.84 %	37.67 %	66.81 %
	True Essay			25.09 %	42.85 %	46.54 %	41.75 %	69.13 %
	Copy 1	1.5 %	9.23 % 9.26 %	15.05 %	31.77 %	37.23 %	28.85 %	58.92 %
	Copy 2	1.45 % 1.4 %		14.39 %	30.19 %	36.16 %	27.78 %	58.58 %
	Fake Essay		9.06 %	20.43 %	36.47 %	39.71 %	35.82 %	64.56 %
	True Essay	2.62 %	12.33 %	24.41 %	42.03 %	45.53 %	40.59 %	70.01 %
GUN	Copy 1	0.9 %	6.86 % 6.34 %	12.45 %	28.06 %	32.95 %	24.58 %	54.86 %
	Copy 2	0.97 %		11.36 %	26.81 %	32.94 %	23.54 %	55.36 %
REVIEW	Fake Review	0.94 %	7.19 %	19.3 %	35.61 %	38.5 %	33.6 %	65.94 %
	True Review	1.07 %	7.85 %	20.8 %	37.03 %	40.62 %	36.28 %	65.71 %

Cuadro C.2: Merito de distribuciones candidatas para latencias ordenada por conteo de mejor ajuste, para distribuciones de dos parámetros

Dataset	Tarea	llog3	dagum	frechet	lnorm3	exGAUS	burr	gengamma
LSIA	Free text	19.26 %	16.63 %	0.33 %	24.03 %	21.65 %	NONE	18.11 %
KM	Free text	34.34 %	18.69 %	0.76 %	17.93 %	19.95 %	0.25 %	8.08 %
	Transcription	29.49 %	20.79 %	1.12 %	18.54 %	20.22 %	NONE	9.83 %
PROSODY GAY	Copy 1	31.64 %	19.38 %	0.89 %	16.82 %	16.64 %	0.19 %	14.45 %
	Copy 2	33.88 %	21.03 %	0.43 %	15.88 %	14.24 %	0.34 %	14.2 %
	Fake Essay	30.19 %	19.88 %	1.33 %	17.93 %	14.87 %	0.4 %	15.41 %
	True Essay	31.82 %	20.67 %	1.36 %	15.1 %	15.32 %	0.62 %	15.1 %
PROSODY GUN	Copy 1	34.22 %	20.84 %	0.69 %	14.98 %	14.29 %	0.43 %	14.55 %
	Copy 2	34.19 %	20.83 %	0.61 %	15.88 %	14.15 %	0.47 %	13.87 %
	Fake Essay	33.35 %	21.06 %	0.81 %	15.71 %	13.7 %	0.43 %	14.94 %
	True Essay	33.36 %	22.05 %	1.09 %	14.99 %	13.59 %	0.32 %	14.61 %
PROSODY REVIEW	Copy 1	32.75 %	19.79 %	1.05 %	15.21 %	14.33 %	0.06 %	16.81 %
	Copy 2	31.81 %	20.34 %	1.06 %	15.93 %	14.98 %	0.37 %	15.51 %
	Fake Review	30.23 %	20.33 %	0.4 %	16.23 %	15.87 %	0.27 %	16.67 %
	True Review	31.4 %	21.65 %	1.1 %	15.94 %	14.84 %	0.2 %	14.88 %

Cuadro C.3: Merito de distribuciones candidatas para tiempos de retención ordenada por conteo de mejor ajuste, para distribuciones de tres parámetros

Dataset	Tarea	llog3	dagum	frechet	lnorm3	exGAUS	burr	gengamma
LSIA	Free text	29.45 %	12.43 %	5.02 %	14.72 %	28.77 %	1.62 %	8 %
KM	Free text	26.65 %	27.92 %	7.36 %	7.87 %	10.66 %	11.17 %	8.38 %
	Transcription	32.68 %	17.18 %	3.94 %	9.3 %	19.72 %	3.38 %	13.8 %
PROSODY GAY	Copy 1	42.23 %	24.45 %	3.52 %	9.57 %	10.46 %	1.41 %	8.35 % 7.38
	Copy 2	41.19 %	24.19 %	3.93 %	9.86 %	11.56 %	1.89 %	% 6.49 %
	Fake Essay	42.54 %	24.72 %	4.81 %	8.25 %	10.84 %	2.36 %	5.35 %
	True Essay	43.7 %	26.5 %	4.79 %	7.25 %	9.35 %	3.07 %	
PROSODY GUN	Copy 1	43.39 %	21.76 %	3.14 %	8.29 %	12.65 %	2.09 %	8.68 %
	Copy 2	43.04 %	23.22 %	3.4 %	8.87 %	12.27 %	1.93 %	7.27 % 6.18
	Fake Essay	42.66 %	23.67 %	5 %	7.72 %	12.49 %	2.28 %	%
	True Essay	42.78 %	25.73 %	5.68 %	8.22 %	9.22 %	2.57 %	5.79 %
PROSODY REVIEW	Copy 1	42.06 %	21.48 %	3.65 %	9.37 %	12.06 %	2.41 %	8.97 % 8.31
	Copy 2	42.9 %	22.88 %	3.45 %	8.85 %	11.87 %	1.73 %	%
	Fake Review	43.9 %	24.35 %	5.68 %	7.76 %	10.49 %	2.17 %	5.64 %
	True Review	44.43 %	24.89 %	5.03 %	7.24 %	9.61 %	2.21 %	6.59 %

Cuadro C.4: Merito de distribuciones candidatas para latencias ordenada por conteo de mejor ajuste, para distribuciones de tres parámetros

Dataset	Tarea	llogis	lnorm	gumbel	gamma	weibull	logis	norm
LSIA	Free text	4.393 (0.019)	4.409 (0.019)	4.416 (0.02)	4.413 (0.019)	4.478 (0.021)	4.425 (0.02)	4.454 (0.021)
KM	Free text	4.579 (0.044)	4.656 (0.039)	4.62 (0.037)	4.607 (0.04)	4.627 (0.039)	4.57 (0.04)	4.595 (0.038)
	Transcription	4.583 (0.044)	4.645 (0.04)	4.624 (0.04)	4.605 (0.041)	4.621 (0.04)	4.575 (0.04)	4.596 (0.039)
PROSODY GAY	Copy 1	4.67 (0.014)	4.713 (0.014)	4.71 (0.014)	4.724 (0.014)	4.799 (0.015)	4.729 (0.015)	4.8 (0.016)
	Copy 2	4.68 (0.014)	4.723 (0.013)	4.719 (0.014)	4.733 (0.014)	4.81 (0.016)	4.737 (0.014)	4.808 (0.015)
	Fake Essay	4.693 (0.013)	4.736 (0.013)	4.733 (0.014)	4.744 (0.014)	4.814 (0.015)	4.75 (0.014)	4.818 (0.016)
	True Essay	4.694 (0.013)	4.738 (0.013)	4.733 (0.013)	4.747 (0.013)	4.821 (0.014)	4.75 (0.013)	4.822 (0.015)
PROSODY GUN	Copy 1	4.67 (0.014)	4.717 (0.014)	4.71 (0.015)	4.722 (0.015)	4.791 (0.015)	4.721 (0.014)	4.793 (0.016)
	Copy 2	4.689 (0.015)	4.734 (0.014)	4.726 (0.015)	4.738 (0.015)	4.808 (0.015)	4.738 (0.015)	4.808 (0.016)
	Fake Essay	4.698 (0.015)	4.746 (0.014)	4.738 (0.015)	4.75 (0.015)	4.816 (0.015)	4.751 (0.015)	4.824 (0.016)
	True Essay	4.693 (0.014)	4.742 (0.013)	4.735 (0.014)	4.745 (0.014)	4.813 (0.014)	4.745 (0.014)	4.82 (0.015)
PROSODY REVIEW	Copy 1	4.699 (0.013)	4.741 (0.013)	4.735 (0.013)	4.749 (0.013)	4.817 (0.014)	4.753 (0.014)	4.82 (0.015)
	Copy 2	4.696 (0.013)	4.736 (0.013)	4.731 (0.014)	4.743 (0.013)	4.811 (0.014)	4.748 (0.014)	4.813 (0.015)
	Fake Review	4.739 (0.013)	4.782 (0.012)	4.773 (0.013)	4.784 (0.013)	4.846 (0.013)	4.786 (0.013)	4.85 (0.014)
	True Review	4.734 (0.012)	4.775 (0.012)	4.771 (0.013)	4.781 (0.012)	4.845 (0.013)	4.787 (0.013)	4.851 (0.014)

Cuadro C.5: Promedio de la verosimilitud logarítmica para tiempos de retención, para distribuciones de dos parámetros

Dataset	Tarea	llogis	lnorm	gumbel	gamma	weibull	logis	norm
LSIA	Free text	6.395 (0.025)	6.432 (0.024)	6.459 (0.024)	6.463 (0.022)	6.524 (0.021)	6.603 (0.024)	6.714 (0.022)
KM	Free text	6.04 (0.034)	6.086 (0.033)	6.152 (0.034)	6.162 (0.032)	6.247 (0.031)	6.334 (0.036)	6.531 (0.033)
	Transcription	5.859 (0.037)	5.904 (0.037)	5.916 (0.038)	5.949 (0.037)	6.045 (0.036)	6.054 (0.04)	6.213 (0.041)
PROSODY GAY	Copy 1	5.707 (0.012)	5.748 (0.012)	5.8 (0.013)	5.813 (0.012)	5.891 (0.012)	5.952 (0.014)	6.125 (0.016)
	Copy 2	5.725 (0.011)	5.766 (0.012)	5.821 (0.013)	5.832 (0.012)	5.908 (0.012)	5.976 (0.014)	6.148 (0.015)
	Fake Essay	5.814 (0.011)	5.857 (0.011)	5.943 (0.011)	5.942 (0.012)	6.018 (0.011)	6.116 (0.013)	6.311 (0.015)
	True Essay	5.824 (0.01)	5.866 (0.01)	5.954 (0.01)	5.952 (0.011)	6.027 (0.01)	6.126 (0.012)	6.327 (0.014)
PROSODY GUN	Copy 1	5.73 (0.011)	5.772 (0.011)	5.825 (0.012)	5.835 (0.012)	5.911 (0.012)	5.977 (0.013)	6.152 (0.015)
	Copy 2	5.746 (0.011)	5.788 (0.011)	5.838 (0.012)	5.848 (0.011)	5.923 (0.012)	5.989 (0.013)	6.159 (0.015)
	Fake Essay	5.823 (0.011)	5.864 (0.011)	5.947 (0.011)	5.942 (0.012)	6.014 (0.011)	6.115 (0.013)	6.302 (0.014)
	True Essay	5.845 (0.011)	5.889 (0.011)	5.976 (0.011)	5.97 (0.011)	6.043 (0.011)	6.148 (0.012)	6.348 (0.013)
PROSODY REVIEW	Copy 1	5.706 (0.011)	5.747 (0.011)	5.798 (0.012)	5.81 (0.012)	5.887 (0.012)	5.95 (0.013)	6.116 (0.015)
	Copy 2	5.709 (0.011)	5.749 (0.011)	5.801 (0.012)	5.812 (0.011)	5.888 (0.011)	5.952 (0.013)	6.118 (0.015)
	Fake Review	5.83 (0.01)	5.873 (0.01)	5.957 (0.01)	5.954 (0.011)	6.027 (0.01)	6.128 (0.012)	6.318 (0.014)
	True Review	5.821 (0.01)	5.862 (0.01)	5.948 (0.01)	5.945 (0.011)	6.019 (0.01)	6.12 (0.012)	6.312 (0.013)

Cuadro C.6: Promedio de la verosimilitud logarítmica para latencias, para distribuciones de dos parámetros

Dataset	Tarea	llog3	dagum	frechet	lnorm3	exGAUS	burr	gengamma
LSIA	Free text	4.373 (0.034)	4.369 (0.021)	4.478 (0.071)	4.372 (0.021)	4.4 (0.022)	4.461 (0.021)	4.43 (0.022)
KM	Free text	4.551 (0.04)	4.559 (0.04)	4.812 (0.04)	4.565 (0.071)	4.571 (0.039)	4.623 (0.04)	4.875 (0.077)
	Transcription	4.567 (0.041)	4.57 (0.042)	4.826 (0.1)	4.578 (0.075)	4.583 (0.041)	4.629 (0.043)	4.803 (0.042)
PROSODY GAY	Copy 1	4.632 (0.017)	4.64 (0.017)	4.734 (0.017)	4.652 (0.029)	4.667 (0.017)	4.725 (0.018)	4.858 (0.055)
	Copy 2	4.615 (0.018)	4.62 (0.018)	4.705 (0.018)	4.638 (0.03)	4.65 (0.018)	4.71 (0.017)	4.833 (0.058)
	Fake Essay	4.645 (0.017)	4.658 (0.017)	4.753 (0.016)	4.667 (0.03)	4.679 (0.017)	4.738 (0.017)	4.814 (0.058)
	True Essay	4.655 (0.016)	4.662 (0.016)	4.768 (0.015)	4.678 (0.026)	4.688 (0.016)	4.745 (0.016)	4.842 (0.052)
PROSODY GUN	Copy 1	4.647 (0.019)	4.669 (0.018)	4.747 (0.019)	4.674 (0.031)	4.687 (0.019)	4.748 (0.018)	4.85 (0.058)
	Copy 2	4.668 (0.02)	4.686 (0.019)	4.787 (0.019)	4.694 (0.032)	4.714 (0.02)	4.767 (0.019)	4.911 (0.055)
	Fake Essay	4.683 (0.019)	4.704 (0.018)	4.767 (0.019)	4.706 (0.033)	4.723 (0.02)	4.784 (0.018)	4.845 (0.06)
	True Essay	4.678 (0.018)	4.694 (0.017)	4.78 (0.017)	4.705 (0.03)	4.718 (0.017)	4.774 (0.017)	4.9 (0.05)
PROSODY REVIEW	Copy 1	4.661 (0.02)	4.673 (0.019)	4.744 (0.019)	4.684 (0.033)	4.701 (0.019)	4.761 (0.019)	4.831 (0.065)
	Copy 2	4.665 (0.021)	4.682 (0.019)	4.757 (0.02)	4.688 (0.032)	4.704 (0.02)	4.766 (0.019)	4.809 (0.066)
	Fake Review	4.705 (0.018)	4.716 (0.018)	4.794 (0.018)	4.728 (0.031)	4.747 (0.018)	4.799 (0.017)	4.848 (0.059)
	True Review	4.712 (0.017)	4.725 (0.016)	4.809 (0.016)	4.732 (0.025)	4.752 (0.017)	4.803 (0.016)	4.891 (0.053)

Cuadro C.7: Promedio de la verosimilitud logarítmica para tiempos de retención, para distribuciones de tres parámetros

Dataset	Tarea	llog3	dagum	frechet	lnorm3	exGAUS	burr	gengamma
LSIA	Free text	6.311 (0.026)	6.325 (0.025)	6.313 (0.026)	6.325 (0.025)	6.324 (0.025)	6.538 (0.034)	6.566 (0.083)
KM	Free text	5.994 (0.034)	5.998 (0.034)	6 (0.034)	6.022 (0.033)	6.022 (0.033)	6.06 (0.039)	6.447 (0.147)
	Transcription	5.818 (0.039)	5.825 (0.039)	5.843 (0.038)	5.844 (0.038)	5.835 (0.038)	5.911 (0.046)	5.979 (0.203)
PROSODY GAY	Copy 1	5.637 (0.015)	5.645 (0.015)	5.67 (0.015)	5.665 (0.015)	5.684 (0.015)	5.708 (0.015)	6.1 (0.059)
	Copy 2	5.66 (0.015)	5.666 (0.014)	5.691 (0.015)	5.688 (0.014)	5.704 (0.014)	5.729 (0.015)	6.126 (0.058)
	Fake Essay	5.742 (0.013)	5.748 (0.013)	5.759 (0.013)	5.772 (0.013)	5.796 (0.013)	5.814 (0.014)	6.273 (0.054)
	True Essay	5.75 (0.012)	5.755 (0.012)	5.769 (0.012)	5.782 (0.012)	5.807 (0.012)	5.818 (0.013)	6.331 (0.045)
PROSODY GUN	Copy 1	5.655 (0.014)	5.663 (0.014)	5.686 (0.014)	5.683 (0.014)	5.702 (0.014)	5.733 (0.015)	6.07 (0.059)
	Copy 2	5.669 (0.014)	5.677 (0.014)	5.697 (0.014)	5.698 (0.014)	5.714 (0.014)	5.744 (0.015)	6.137 (0.056)
	Fake Essay	5.748 (0.014)	5.76 (0.013)	5.766 (0.014)	5.777 (0.014)	5.802 (0.013)	5.827 (0.014)	6.23 (0.054)
	True Essay	5.749 (0.012)	5.757 (0.012)	5.767 (0.012)	5.78 (0.012)	5.805 (0.012)	5.826 (0.013)	6.283 (0.047)
PROSODY REVIEW	Copy 1	5.654 (0.016)	5.662 (0.016)	5.686 (0.016)	5.682 (0.016)	5.701 (0.016)	5.719 (0.016)	6.061 (0.069)
	Copy 2	5.672 (0.015)	5.679 (0.015)	5.7 (0.015)	5.7 (0.015)	5.718 (0.015)	5.737 (0.015)	6.109 (0.063)
	Fake Review	5.786 (0.014)	5.796 (0.014)	5.807 (0.014)	5.816 (0.014)	5.843 (0.014)	5.856 (0.014)	6.342 (0.05)
	True Review	5.762 (0.013)	5.772 (0.013)	5.784 (0.013)	5.794 (0.013)	5.822 (0.013)	5.832 (0.013)	6.27 (0.053)

Cuadro C.8: Promedio de la verosimilitud logarítmica para latencias, para distribuciones de tres parámetros

Dataset		Tarea	llogis	lnorm	gumbel	gamma	weibull	logis	norm
LSIA		Free text	5.61 %	10.09 %	11.72 %	27.41 %	24.72 %	8.88 %	18.32 %
KM		Free text	9.37 %	31.62 %	33.49 %	27.71 %	37.94 %	11.01 %	25.06 %
		Transcription	5.48 %	23.57 %	26.67 %	22.44 %	30.48 %	7.38 %	18.81 %
PROSODY	GAY	Copy 1	7.19 %	14.41 %	14.07 %	25.13 %	30.25 %	12.95 %	26.35 %
		Copy 2	7.24 %	14.85 %	13.6 %	26.11 %	29.63 %	12.41 %	26.37 %
		Fake Essay	6.98 %	14.84 %	13.9 %	25.4 %	28.44 %	12.95 %	26.01 %
		True Essay	7.62 %	15.58 %	15.18 %	26.84 %	31.66 %	13.64 %	28.47 %
	GUN	Copy 1	6.66 %	15.4 %	13.34 %	28.06 %	30.08 %	10.91 %	25.36 %
		Copy 2	6.27 %	14.99 %	12.88 %	25.22 %	28.19 %	10.59 %	24.23 %
		Fake Essay	6.41 %	15.12 %	13.26 %	27.44 %	28.04 %	10.39 %	24.26 %
		True Essay	7.82 %	17.27 %	15.49 %	29.09 %	32.28 %	12.55 %	27.63 %
	REVIEW	Copy 1	7.82 %	14.02 %	12.91 %	26.18 %	27.25 %	11.22 %	24.37 %
		Copy 2	7.6 %	14.23 %	12.86 %	25.45 %	27.21 %	11.91 %	23.7 %
		Fake Review	8.11 %	15.32 %	13.88 %	27.13 %	26.97 %	12.27 %	24.13 %
		True Review	8.71 %	15.9 %	14.82 %	27.43 %	28.86 %	13.02 %	26.2 %

Cuadro C.9: Porcentaje de rechazo para tiempos de retención, para distribuciones de dos parámetros

Dataset	Tarea	llogis	lnorm	gumbel	gamma	weibull	logis	norm
LSIA	Free text	5.62 %	16.22 %	21.84 %	31.27 %	38.5 %	36.52 %	62.02 %
KM	Free text	15.69 %	43.09 %	60.66 %	68.77 %	79.86 %	77.05 %	91.57 %
	Transcription	4.29 %	23.1 %	25.71 %	47.38 %	64.76 %	53.81 %	79.05 %
PROSODY GAY	Copy 1	0.74 % 0.98 % 1.07 %	8.39 % 8.38 %	14.73 %	31.85 %	37.46 %	28.87 %	59.02 %
	Copy 2			14.84 %	31.5 %	36.68 %	28.37 %	59.48 %
	Fake Essay	1.78 %	9.05 %	22.02 %	38.72 %	42.84 %	37.67 %	66.81 %
	True Essay		11.89 %	25.09 %	42.85 %	46.54 %	41.75 %	69.13 %
	Copy 1	1.5 %	9.23 % 9.26 %	15.05 %	31.77 %	37.23 %	28.85 %	58.92 %
	Copy 2	1.45 % 1.4 %		14.39 %	30.19 %	36.16 %	27.78 %	58.58 %
	Fake Essay		9.06 %	20.43 %	36.47 %	39.71 %	35.82 %	64.56 %
	True Essay	2.62 %	12.33 %	24.41 %	42.03 %	45.53 %	40.59 %	70.01 %
PROSODY GUN	Copy 1	0.9 %	6.86 % 6.34 %	12.45 %	28.06 %	32.95 %	24.58 %	54.86 %
	Copy 2	0.97 %		11.36 %	26.81 %	32.94 %	23.54 %	55.36 %
PROSODY REVIEW	Fake Review	0.94 %	7.19 %	19.3 %	35.61 %	38.5 %	33.6 %	65.94 %
	True Review	1.07 %	7.85 %	20.8 %	37.03 %	40.62 %	36.28 %	65.71 %

Cuadro C.10: Porcentaje de rechazo para latencias, para distribuciones de dos parámetros

Dataset	Tarea	llog3	dagum	frechet	lnorm3	exGAUS	burr	gengamma
LSIA	Free text	5.12 %	5.14 %	5.64 %	18.52 %	15.39 %	17.12 %	78.15 %
KM	Free text	1.77 %	4.81 %	40.78 %	12.12 %	10.1 %	24.49 %	77.46 %
	Transcription	0.84 %	3.37 %	25.58 %	6.18 %	7.58 %	18.82 %	74.05 %
PROSODY GAY	Copy 1	7.3 %	8 %	14.61 %	15.62 %	15.8 %	21.95 %	78.28 % 79.56
	Copy 2	6.7 %	7.9 %	12.52 %	16.23 %	14.82 %	22.52 %	% 77.68 %
	Fake Essay	7.05 %	7.72 %	13.07 %	14.96 %	14.12 %	21.82 %	78.56 %
	True Essay	7.79 %	8.33 %	14.18 %	16.64 %	15.73 %	23.39 %	
PROSODY GUN	Copy 1	6.09 %	7.74 % 7.4	14.37 %	18.4 %	15.11 %	21.95 %	79.69 % 78.19
	Copy 2	6.36 %	%	13.9 %	16.68 %	16.3 %	21.03 % 22	% 78.55 %
	Fake Essay	5.79 %	7.52 %	13.86 %	16.49 %	15.5 %	%	78.37 %
	True Essay	7.85 %	9.19 %	17.55 %	19.33 %	17.24 %	24.96 %	
PROSODY REVIEW	Copy 1	8.8 %	9.65 %	13.88 %	19.68 %	18.08 %	21.68 %	79.63 % 79.76
	Copy 2	9.37 %	10.3 %	14.03 %	18.8 %	17.43 %	22.97 %	%
	Fake Review	9.52 %	10.54 %	12.17 %	18.64 %	18.1 %	22.43 %	78.45 %
	True Review	9.17 %	9.62 %	14.56 %	19.78 %	18.38 %	21.88 %	79.24 %

Cuadro C.11: Porcentaje de rechazo para tiempos de retención, para distribuciones de tres parámetros

t	Tarea	llog3	dagum	frechet	lnorm3	exGAUS	burr	gengamma
LSIA	Free text	1.72 %	2.92 %	2.8 %	6.81 %	16.6 %	46.89 %	85.5 %
KM	Free text	8.38 %	5.6 %	10.46 %	23.86 %	36.55 %	32.23 %	91.12 %
	Transcription	2.26 %	2.83 %	6.94 %	11.27 %	12.39 %	31.27 %	82.23 %
PROSODY GAY	Copy 1	0.52 %	0.71 %	1.4 %	6.24 % 6.22	16.94 %	9.39 %	87.52 % 88.27
	Copy 2	0.44 %	0.54 %	1.59 %	% 5.71 %	16.46 %	9.81 %	% 90.59 %
	Fake Essay	0.23 %	0.27 %	1.17 %	9.31 %	22.09 %	11.47 %	91.46 %
	True Essay	0.86 %	0.83 %	2.2 %		24.56 %	12.93 %	
	Copy 1	1 %	1.27 %	2.4 %	7.37 % 6.14	18.05 %	11.34 %	87.43 %
	Copy 2	0.71 %	1.19 %	1.8 %	%	16.23 % 21	10.9 %	87.9 %
	Fake Essay	0.79 %	1.01 %	1.66 %	7.54 %	%	11.18 %	90.4 %
	True Essay	1.33 %	1.66 %	2.75 %	9.4 %	23.55 %	15.01 %	90.71 %
GUN	Copy 1	0.51 %	0.9 %	1.31 % 0.68	4.77 % 4.21	16.49 %	6.56 % 6.96	86.63 % 86.83
	Copy 2	0.49 %	0.54 %	%	%	13.92 %	%	%
REVIEW	Fake Review	0.46 %	0.88 %	1.13 %	4.99 %	19.41 %	9.15 %	91.34 %
	True Review	0.65 %	0.93 %	1.24 %	6.39 %	20.67 %	9.49 %	90.03 %

Cuadro C.12: Porcentaje de rechazo para latencias, para distribuciones de tres parámetros

Apéndice D

La herramienta desarrollada

En este apéndice se describe el funcionamiento de la herramienta de análisis de cadencias de tecleo que constituye uno de los aportes de esta tesis, junto con la forma de configurarla y extenderla.

El resto del capítulo está organizado como sigue. La sección D.1 provee las direcciones de descarga del código fuente y la compilación binaria lista para su ejecución. La sección D.2 enumera los conjuntos de datos que se incluyen con los anteriores. La sección D.3 lista los prerequisitos y dependencias para la compilación y ejecución de la herramienta. La sección D.4 explica la ejecución por línea de comandos. La sección D.5 explica la integración de la herramienta como biblioteca de software. La sección D.6 ejemplifica la salida estándar para distintos experimentos. La sección D.7 indica las rutas y estructura de los archivos ARFF de entrenamiento para los distintos métodos. La sección D.8 muestra como modificar la configuración de los experimentos. Finalmente, la sección D.9 detalla como extender la herramienta para incluir nuevos conjuntos de datos, atributos derivados, y estrategias de síntesis.

D.1. Binarios y código fuente

El código fuente de la herramienta de análisis de cadencias de tecleo utilizada para los experimentos de esta tesis se pone a disponibilidad en forma libre y gratuita en el repositorio público del Laboratorio de Sistemas de Información Avanzados. La dirección de descarga es

https://github.com/lsia/herramientaGonzalez2021

Adicionalmente, se provee una compilación binaria lista para su ejecución en plataformas Windows 7 o superior. Esta se encuentra también en el repositorio público del Laboratorio de Sistemas de Información Avanzados, aunque comprimido en volúmenes de 50MB debido a las restricciones del repositorio. Las direcciones de descarga son

https://github.com/lsia/herramientaGonzalez2021/blob/main/BINARY.part1.rar
https://github.com/lsia/herramientaGonzalez2021/blob/main/BINARY.part2.rar
https://github.com/lsia/herramientaGonzalez2021/blob/main/BINARY.part3.rar
https://github.com/lsia/herramientaGonzalez2021/blob/main/BINARY.part4.rar

El software se provee bajo licencia Creative Commons CC BY-SA 4.0, que puede ser consultada en línea siguiendo el enlace [166].

D.2. Conjuntos de datos incluidos

Los conjuntos de datos incluidos en la compilación binaria son LSIA, KM, GAY, GUN, y REVIEW, que corresponden a los descritos en la sección 5.1.3. Todos ellos se encuentran en el directorio Datasets.

Utilizando los archivos de configuración como se describe en las secciones D.8 y D.9 es muy sencillo extender la herramienta para que soporte nuevos conjuntos de datos.

D.3. Prerequisitos y dependencias

El código fuente requiere ser compilado bajo .NET Core 3.1, que se encuentra disponible para múltiples plataformas. El entorno alternativo de desarrollo Mono también puede ser utilizado para la tarea.

La compilación binaria requiere un sistema operativo Windows 7 o superior y .NET Framework 4.7.2 o .NET Core 3.1 previamente instalado.

La ejecución de algunos de los experimentos requiere que la herramienta de aprendizaje automático WEKA 3.8.1 o superior [137] se encuentre previamente instalada.

D.4. Ejecución por línea de comandos

La herramienta de análisis de cadencias de tecleo es una aplicación de consola desarrollada en C# bajo .NET Core 3.1. La distribución binaria que se pone a disponibilidad en forma libre y gratuita ejecuta bajo el sistema operativo Windows, pero es factible compilar el código fuente utilizando versiones de .NET Core para otras plataformas como Linux o MacOS. Los parámetros de línea de comandos no se modifican en estos casos.

El patrón general de invocación tiene la forma

KSDExperiments.exe {EXPERIMENTO} [[parámetros]]

en donde EXPERIMENTO puede ser *SYNTHESIZE*, *VERIFY*, o *IDENTIFY*. Estos corresponden a los principales experimentos de la tesis y sus parámetros específicos se describen a continuación. Cuando se omiten los parámetros, la herramienta muestra los parámetros esperados, como se muestra en la figura D.1

D.4. Ejecución por línea de comandos

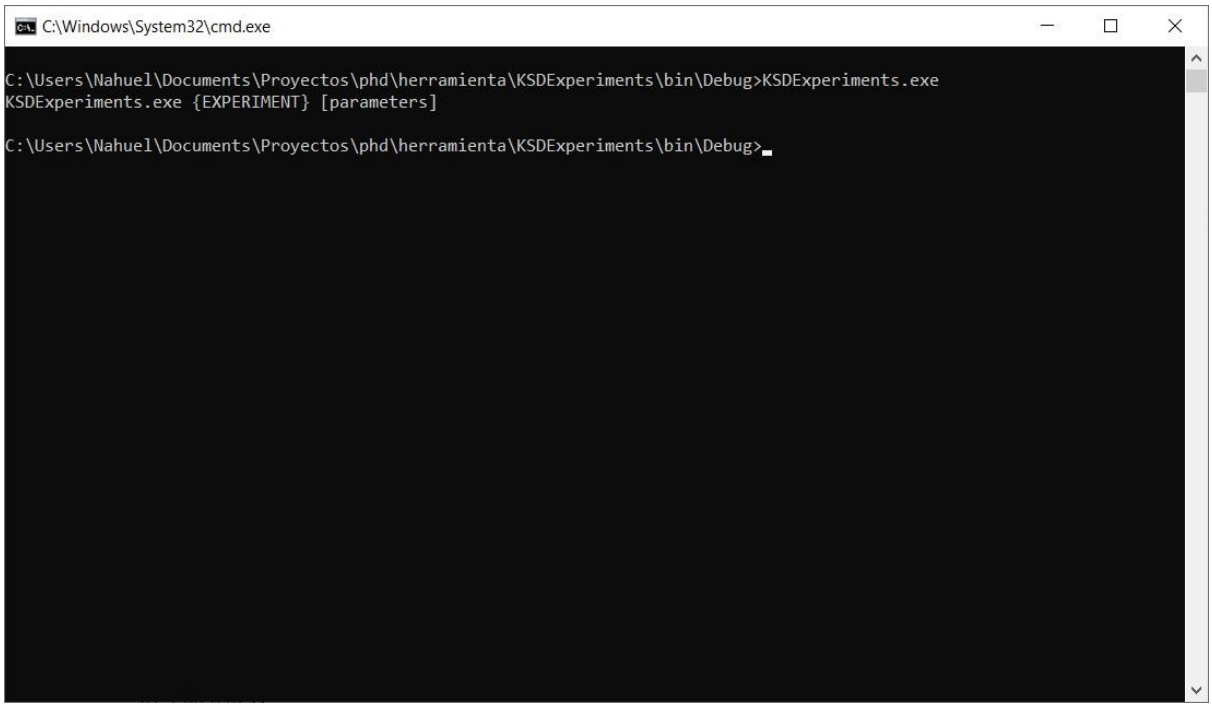

Figura D.1: Ejecución sin parámetros de la herramienta por línea de comandos

D.4.1. SYNTHESIZE

El experimento SYNTHESIZE permite sintetizar la cadencia de tecleo de un usuario dado en alguno de los conjuntos de datos cargados. Requiere cuatro parámetros adicionales, que son

- **dataset**. Nombre del conjunto de datos en el cual se encuentra el usuario cuyo perfil se utilizará para sintetizar la muestra artificial de su cadencia de tecleo.
- **usuario**. Nombre del usuario cuyo perfil se utilizará para sintetizar la muestra artificial de su cadencia de tecleo.
- **method**. Método utilizado para sintetizar la muestra artificial de su cadencia de tecleo.
- **text**. Texto de la muestra artificial cuyos atributos temporales serán sintetizados.

Los métodos de síntesis soportados corresponden con los descritos en la sección 4.3 y son los siguientes

- **AverageSynthesizer**. Sintetizador con estrategia Average. Utiliza para cada atributo temporal el promedio de los valores en cada conjunto S_i.
- **UniformSynthesizer**. Sintetizador con estrategia Uniform. Utiliza para cada atributo temporal una muestra de una variable uniforme con el promedio y el desvío estándar de los valores en cada conjunto S_i. Al utilizarse con contextos de orden uno corresponde a NoiseBot.**GaussianSynthesizer**. Sintetizador con estrategia Gaussian. Utiliza para cada atributo temporal una muestra de una variable normal con el promedio y el desvío estándar de los

valores en cada conjunto S_i. Al utilizarse con contextos de orden uno corresponde a GaussianeBot.

- **LCBMSynthesizer**. Sintetizador con la estrategia LBMC de [76]. Para los tiempos de retención utiliza un modelo gaussiano, pues la estrategia original no contempla los mismos.

- **HistogramSynthesizer**. Sintetizados basado en histogramas empíricos del tipo ICDF. Utiliza para cada atributo temporal un muestreo uniforme de la distribución inversa suavizada de cada S_i.

- **NonStationaryHistogramSynthesizer**. Sintetizados basado en histogramas empíricos del tipo NS/ICDF. Utiliza para cada atributo temporal un muestreo uniforme de la distribución inversa suavizada de cada S_i más un término de offset no estacionario a nivel de palabras.

- **NonStationaryHistogramSynthesizerReverse**. Sintetizados basado en histogramas empíricos del tipo NS/ICDF. Utiliza para cada atributo temporal un muestreo uniforme de la distribución inversa suavizada de cada S_i más un término de offset no estacionario a nivel de palabras, con una estrategia alternativa a la anterior.

Por ejemplo, para sintetizar una muestra del texto *¡Hola, mundo!* como la escribiría el usuario *s019* del conjunto de datos KM, utilizando la estrategia ICDF debemos escribir la siguiente línea de comandos

KSDExperiments.exe SYNTHESIZE KM s019 HistogramSynthesizer ‘‘¡Hola, mundo!’’

La cadencia sintetizada se encontrará en el archivo SYNTHETIC.csv del directorio Output/SYNTHESIZE. Es este un archivo de valores separados por comas con tres columnas (código virtual de tecla, tiempo de retención, y latencia), con encabezados y una fila para cada tecla del texto objetivo. Los valores se encuentran en milisegundos.

El cuadro D.1 ejemplifica la salida de la línea de comandos anterior. El valor 2147483648 (int.MinValue) indica ausencia de datos. Ese es el valor de latencia de la primer tecla pues no hay tecla anterior, y por lo tanto no hay tiempo de latencia entre teclas.

D.4.2. VERIFY

El experimento VERIFY permite evaluar las tasas de error del método de defensa.
Requiere tres parámetros, que son

- **dataset**. Nombre del conjunto de datos en el cual se encuentra el usuario cuyo perfil se utilizará para evaluar las tasas de error del método de defensa.

- **usuario**. Nombre del usuario cuyo perfil se utilizará para evaluar las tasas de error del método de defensa.

VK	HT (Retención)	FT (latencia)
72	66	-2147483648
79	91	78
76	63	205
65	125	192
32	98	197
32	82	149
77	79	205
85	83	216
78	30	170
68	22	93
79	57	197
32	97	118

Cuadro D.1: Ejemplo de salida de SYNTHESIZE

- **profile**. Tipo de perfil que se utilizara en el entrenamiento del método de defensa.

El tipo de perfil puede ser cualquiera de los dos siguientes

- **within-user**. Perfil intrausuario, con todas las muestras del usuario objetivo.

- **between-user**. Perfil interusuario de la población general, con cien muestras tomadas aleatoriamente entre todas las muestras de los otros usuarios del conjunto de datos considerado.

Por ejemplo, para evaluar las tasas de error del método de defensa para el usuario *s019* del conjunto de datos KM, entrenado con un perfil intrausuario, debemos escribir la siguiente línea de comandos

KSDExperiments.exe VERIFY KM s019 within-user

Además de contar con los resultados en la salida estándar de la consola, se almacenará el archivo ARFF de entrenamiento del modelo en el directorio de salida Output/VERIFY. La salida se describe en detalle en la sección D.6.

D.4.3. IDENTIFY

El experimento IDENTIFY permite evaluar el rendimiento del método de identificación del texto ingresado utilizando los atributos temporales sin nombres de teclas para un usuario dado en alguno de los conjuntos de datos cargados. Requiere dos parámetros adicionales, que son

- **dataset**. Nombre del conjunto de datos en el cual se encuentra el usuario cuyo perfil se utilizará para evaluar el rendimiento del método de identificación.

- **usuario**. Nombre del usuario cuyo perfil se utilizará para evaluar el rendimiento del método de identificación.

Por ejemplo, para evaluar el rendimiento del método de identificación para el usuario *s019* del conjunto de datos KM, debemos escribir la siguiente línea de comandos

KSDExperiments.exe IDENTIFY KM s019

Además de contar con los resultados en la salida estándar de la consola, se almacenará el archivo ARFF de entrenamiento del modelo en el directorio de salida Output/IDENTIFY para cada oración en el perfil del usuario. La salida se describe en detalle en la sección D.6.

D.5. Integración como biblioteca de software

El resultado de la compilación del código fuente de la herramienta es un *assembly* de .NET, con el manifiesto exportado. Las clases utilizadas para los experimentos han sido declaradas como públicas.

De esta forma, el mismo ejecutable puede ser referenciado como una biblioteca de software, y los métodos y clases que contiene pueden ser utilizados en obras derivadas sin necesidad de reutilizar el código fuente.

D.6. Salida estándar

La ejecución de todos los experimentos es precedida por una etapa de inicialización que se muestra a continuación.

Listado D.1: Inicialización de la herramienta

```
2021/07/25 11:21:11.222 I          Pipeline                          Initializing pipeline SYNTHESIZE...
2021/07/25 11:21:11.222 I          Pipeline                          Stage loadLSIA (LoadDatasetStage)
2021/07/25 11:21:11.222 I          LoadDatasetStage                      TYPE LoadDatasetStage
2021/07/25 11:21:11.222 I          LoadDatasetStage                        Creating instance...
2021/07/25 11:21:11.222 I          LoadDatasetStage                      Ready.
2021/07/25 11:21:11.281 I          Pipeline                          Stage loadKM (LoadDatasetStage)
2021/07/25 11:21:11.281 I          LoadDatasetStage                      TYPE LoadDatasetStage
2021/07/25 11:21:11.282 I          LoadDatasetStage                        Creating instance...
2021/07/25 11:21:11.282 I          LoadDatasetStage                      Ready.
2021/07/25 11:21:11.282 I          Pipeline                          Stage loadGAY (LoadDatasetStage)
2021/07/25 11:21:11.282 I          LoadDatasetStage                      TYPE LoadDatasetStage
2021/07/25 11:21:11.282 I          LoadDatasetStage                        Creating instance...
2021/07/25 11:21:11.282 I          LoadDatasetStage                      Ready.
2021/07/25 11:21:11.282 I          Pipeline                          Stage loadGUN (LoadDatasetStage)
2021/07/25 11:21:11.282 I          LoadDatasetStage                      TYPE LoadDatasetStage
2021/07/25 11:21:11.282 I          LoadDatasetStage                        Creating instance...
2021/07/25 11:21:11.282 I          LoadDatasetStage                      Ready.
2021/07/25 11:21:11.282 I          Pipeline                                Stage loadREVIEW (LoadDatasetStage)
2021/07/25 11:21:11.282 I          LoadDatasetStage                      TYPE LoadDatasetStage
2021/07/25 11:21:11.282 I          LoadDatasetStage                        Creating instance...
2021/07/25 11:21:11.282 I          LoadDatasetStage                      Ready.
2021/07/25 11:21:11.282 I          Pipeline                          Stage split (ThresholdPartitioner)
2021/07/25 11:21:11.282 I          Pipeline                          Stage fts (CleanFTs)
2021/07/25 11:21:11.282 I          Pipeline                          Stage run (RunDefaultExperiment)
2021/07/25 11:21:11.282 I              FiniteContextsConfiguration       Initialing model storage...
2021/07/25 11:21:11.282 I              FiniteContextsConfiguration         TD
2021/07/25 11:21:11.282 I              FiniteContextsConfiguration         HI
2021/07/25 11:21:11.282 I              FiniteContextsConfiguration         DIE
2021/07/25 11:21:11.282 I              FiniteContextsConfiguration         DIL
2021/07/25 11:21:11.282 I              FiniteContextsConfiguration       Initialing features...
2021/07/25 11:21:11.282 I              FiniteContextsConfiguration         AvgStdev
2021/07/25 11:21:11.282 I              FiniteContextsConfiguration         Histogram
2021/07/25 11:21:11.282 I              FiniteContextsConfiguration         DIE
```

D.6. Salida estándar

```
2021/07/25 11:21:11.282 I FiniteContextsConfiguration DIL
2021/07/25 11:21:11.282 I FiniteContextsConfiguration RMix
2021/07/25 11:21:11.282 I FiniteContextsConfiguration R1
2021/07/25 11:21:11.298 I FiniteContextsConfiguration Initialing attributes...
2021/07/25 11:21:11.298 I FiniteContextsConfiguration FTDM

2021/07/25 11:21:11.298 I FiniteContextsConfiguration FTDE
2021/07/25 11:21:11.298 I FiniteContextsConfiguration FTDMin
2021/07/25 11:21:11.298 I FiniteContextsConfiguration FTDC 2021/07/25 11:21:11.298 I FiniteContextsConfiguration
FTZ
2021/07/25 11:21:11.298 I FiniteContextsConfiguration FTHM
2021/07/25 11:21:11.298 I FiniteContextsConfiguration FTHE
2021/07/25 11:21:11.298 I FiniteContextsConfiguration FTHMin
2021/07/25 11:21:11.298 I FiniteContextsConfiguration FTHC 2021/07/25 11:21:11.298 I FiniteContextsConfiguration
FTHZ
2021/07/25 11:21:11.298 I FiniteContextsConfiguration FTDIE 2021/07/25 11:21:11.298 I FiniteContextsConfiguration
FTDIL
2021/07/25 11:21:11.298 I FiniteContextsConfiguration FTRAll 2021/07/25 11:21:11.298 I FiniteContextsConfiguration
FTRMix
2021/07/25 11:21:11.298 I FiniteContextsConfiguration FTR1
2021/07/25 11:21:11.298 I FiniteContextsConfiguration HTDM
2021/07/25 11:21:11.315 I FiniteContextsConfiguration HTDE
2021/07/25 11:21:11.315 I FiniteContextsConfiguration HTDMin
2021/07/25 11:21:11.315 I FiniteContextsConfiguration HTDC 2021/07/25 11:21:11.335 I FiniteContextsConfiguration
HTZ
2021/07/25 11:21:11.335 I FiniteContextsConfiguration HTHM
2021/07/25 11:21:11.348 I FiniteContextsConfiguration HTHE
2021/07/25 11:21:11.363 I FiniteContextsConfiguration HTHMin 2021/07/25 11:21:11.363 I FiniteContextsConfiguration
HTHC 2021/07/25 11:21:11.363 I FiniteContextsConfiguration HTHZ
2021/07/25 11:21:11.363 I FiniteContextsConfiguration HTDIE
2021/07/25 11:21:11.386 I FiniteContextsConfiguration HTDIL
2021/07/25 11:21:11.386 I FiniteContextsConfiguration HTRAll 2021/07/25 11:21:11.386 I FiniteContextsConfiguration
HTRMix
2021/07/25 11:21:11.412 I FiniteContextsConfiguration HTR1
2021/07/25 11:21:11.412 I LoadDatasetStage Loading dataset...
2021/07/25 11:21:11.412 I BinaryDatasetReader Reading binary dataset LSIA...
2021/07/25 11:21:16.731 I BinaryDatasetReader Ready.
2021/07/25 11:21:16.745 I LoadDatasetStage Ready.
2021/07/25 11:21:16.745 I LoadDatasetStage Loading dataset...
2021/07/25 11:21:16.745 I BinaryDatasetReader Reading binary dataset KM...
2021/07/25 11:21:16.763 I BinaryDatasetReader Ready.
2021/07/25 11:21:16.767 I LoadDatasetStage Ready.
2021/07/25 11:21:16.767 I LoadDatasetStage Loading dataset...
2021/07/25 11:21:16.767 I BinaryDatasetReader Reading binary dataset GAY...
2021/07/25 11:21:16.911 I BinaryDatasetReader Ready.
2021/07/25 11:21:16.911 I LoadDatasetStage Ready.
2021/07/25 11:21:16.911 I LoadDatasetStage Loading dataset...
2021/07/25 11:21:16.911 I BinaryDatasetReader Reading binary dataset GUN...
2021/07/25 11:21:17.070 I BinaryDatasetReader Ready.
2021/07/25 11:21:17.070 I LoadDatasetStage Ready.
2021/07/25 11:21:17.070 I LoadDatasetStage Loading dataset...
2021/07/25 11:21:17.070 I BinaryDatasetReader Reading binary dataset REVIEW...
2021/07/25 11:21:17.221 I BinaryDatasetReader Ready.
2021/07/25 11:21:17.221 I LoadDatasetStage Ready.
2021/07/25 11:21:17.221 I ThresholdPartitioner Partitioning sessions...
2021/07/25 11:21:17.221 I ExperimentParallelization DATASET LSIA
2021/07/25 11:21:18.276 I ExperimentParallelization DATASET KM
2021/07/25 11:21:18.276 I ExperimentParallelization DATASET GAY
2021/07/25 11:21:18.309 I ExperimentParallelization DATASET GUN
2021/07/25 11:21:18.330 I ExperimentParallelization DATASET REVIEW
2021/07/25 11:21:18.380 I CleanFTs Cleaning FTs after partitions...
2021/07/25 11:21:18.380 I ExperimentParallelization DATASET LSIA
2021/07/25 11:21:21.363 I ExperimentParallelization DATASET KM
2021/07/25 11:21:21.363 I ExperimentParallelization DATASET GAY
```

D.6. Salida estándar

```
2021/07/25 11:21:21.436 I ExperimentParallelization DATASET GUN
2021/07/25 11:21:21.499 I ExperimentParallelization DATASET REVIEW
```

Aquí se observa como la herramienta inicializa el pipeline de procesamiento y los generadores de atributos derivados, para luego cargar los conjuntos de datos y realizar un preprocesamiento inicial y limpieza de datos.

Cada experimento produce su salida específica luego de esta inicialización general. El experimento SYNTHESIZE entrena el perfil del usuario, carga la estrategia de síntesis, genera la salida esperada, y guarda los resultados en el directorio Output.

Listado D.2: Salida del experimento SYNTHESIZE

```
2021/07/25 11:27:17.873 I RunDefaultExperiment Running default experiment SYNTHESIZE . . .
2021/07/25 11:27:17.878 I PipelineStage DATASET KM
2021/07/25 11:27:17.878 I PipelineStage USER s019
2021/07/25 11:27:17.878 I PipelineStage Training user profile . . .
2021/07/25 11:27:17.989 I PipelineStage Loading synthesizer HistogramSynthesizer . . .
2021/07/25 11:27:17.989 I PipelineStage Generating virtual keys sequence . . .
2021/07/25 11:27:17.989 I PipelineStage Saving results . . .
2021/07/25 11:27:17.989 I PipelineStage 2021/07/25 11:27:18.008 I Program READY.
```

El experimento VERIFY entrena el perfil de robot malo, cuyas falsificaciones serán utilizadas como adversarios, para luego dividir las muestras del perfil del usuario en oraciones y generando instancias de entrenamiento para cada una de ellas como se ha descrito en la sección 4.4. Se explicita en la salida la cantidad de oraciones que conforman el entrenamiento del modelo, aquí 90.

Listado D.3: Salida del experimento VERIFY

```
2021/07/25 12:37:41.736 I PipelineStage DATASET KM
2021/07/25 12:37:41.736 I PipelineStage USER s019
2021/07/25 12:37:41.736 I PipelineStage Training WITHIN-USER adversaries profile . . .
2021/07/25 12:37:41.851 I PipelineStage Setting up evil bot synthesizers . . .
2021/07/25 12:37:41.884 I PipelineStage Splitting legitimate samples into sentences . . .
2021/07/25 12:37:41.884 I PipelineStage Synthesizing evil bot forgeries . . .
2021/07/25 12:37:41.930 I PipelineStage The training set contains 90 sentences of 30 characters or more .
```

A continuación, se repite la salida del clasificador de WEKA, incluyendo la matriz de confusión y las tasas de error.

Listado D.4: Salida de WEKA para el experimento VERIFY

D.6. Salida estándar

−0·7133 * (normalized) FT DM

*

+ −0·4103 (normalized) FT DE − − + −0.9754 * (normalized) FTDMin

− + −0.8604 *

(normalized)

FTDC + −0.2662 * (normalized) FTZ − −

+ 0.3617 * (normalized) FTHM + 0.485 * (normalized) FTHE

+ 0.1118 * (normalized) FTHMin

+ −2.9971 * (normalized) FTHC + − 2.0497 *

(normalized) FTHZ −

+ 0.7817 * (normalized) FTDIE

+ 0.9154 * (normalized) FTDIL

+ 0.4341 * (normalized) FTRAll + 0.0656 * (normalized) FTRMix

* + 0.0992 (normalized) FTR1

+ −0.4461 * (normalized) HT DM

+ −0.0605 * (normalized) HT DE

+ −0.7036 * (normalized) HT DMin

+ 0·341 * (normalized) HT DC

+ 1·3451 * (normalized) HT Z *

+ −0.3274 * (normalized) HTHM − + 0.9685 * (normalized) HTDIL

+ −0.3423 * (normalized) HT HE + 0.1264 * (

+ −0.4104 * (normalized) HT HMin normalized)

+ −1.3701 * (normalized) HT HC HTRAll + 0.1149 *

+ 1·644 * (normalized) HT HZ (normalized)

HTRMix +

+ 0·6054 (normalized) HT DIE −0.0541 * (

normalized) HTR1

+ 2.168

Number of kernel evaluations : 2675 (80.335 % cached)

Time taken to build model : 0.15 seconds

Time taken to test model on training data : 0.01 seconds

=== Error on training data ===

Correctly Classified Instances 167 99.4048 %

Incorrectly Classified Instances 1 0.5952 %

Kappa s t a t i s t i c 0.9881

Mean absolute error 0.006 Root mean squared error 0.0772 Relative absolute error 1.1905 %

Root relative squared error 15.4303 % Total Number of Instances 168

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F−Measure	MCC	ROC Area	PRC Area	Class
	0.988	0.000	1.000	0.988	0.994	0.988	0.994	0.994	legitimate
	1.000	0.012	0.988	1.000	0.994	0.988	0.994	0.988	impostor
Weighted Avg.	0.994	0.006	0.994	0.994	0.994	0.988	0.994	0.991	

=== Confusion Matrix ===

```
 a   b <−− classified as
83 1 |  a = legitimate
 0 84 | b = impostor
```

Time taken to perform cross −validation : 0.06 seconds

=== S t r a ti f i e d cross −validation ===

Correctly Classified Instances 167 99.4048 %

Incorrectly Classified Instances 1 0.5952 %

Kappa s t a t i s t i c 0.9881

Mean absolute error 0.006 Root mean squared error 0.0772

Relative absolute error 1.1901 %

D.6. Salida estándar

```
Root relative squared error 15.4253 %

Total Number of Instances 168

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   MCC      ROC Area  PRC Area  Class
               0.988     0.000     1.000       0.988   0.994       0.988    0.994     0.994     legitimate
               1.000     0.012     0.988       1.000   0.994       0.988    0.994     0.988     impostor
Weighted Avg.  0.994     0.006     0.994       0.994   0.994       0.988    0.994     0.991

=== Confusion Matrix ===

   a   b <-- classified as
 83  1 |  a = legitimate
  0 84 |  b = impostor
```

Finalmente, la herramienta reporta los EERs individuales para cada atributo derivado que se utilizó durante la evaluación, como se muestra a continuación.

Listado D.5: EERs individuales para el experimento VERIFY

```
2021/07/25 12:37:43.334 I PipelineStage BETWEEN-USER C7 MIXED FAR: 0% FRR: 1.19 %
2021/07/25 12:37:43.334 I PipelineStage FT DM 63.1 %/0.66

2021/07/25 12:37:43.334 I PipelineStage FTDE 62.5 %/0.96
2021/07/25 12:37:43.334 I PipelineStage FTDMin 73.81 %/0.48
2021/07/25 12:37:43.334 I PipelineStage FTDC 75 %/0.18
2021/07/25 12:37:43.334 I PipelineStage FTZ 61.9 %/1.78
2021/07/25 12:37:43.334 I PipelineStage FT HM 58.33 %/0.24
2021/07/25 12:37:43.334 I PipelineStage FTHE 57.14 %/0.28
2021/07/25 12:37:43.334 I PipelineStage FTHMin 61.31 %/0.2
2021/07/25 12:37:43.334 I PipelineStage FTHC 95.24 %/0.25
2021/07/25 12:37:43.334 I PipelineStage FTHZ 45.24 %/0.04
2021/07/25 12:37:43.334 I PipelineStage FTDIE 46.43 %/0.19
2021/07/25 12:37:43.349 I PipelineStage FTDIL 46.43 %/0.19
2021/07/25 12:37:43.349 I PipelineStage FTRAll 55.95 %/0.38
2021/07/25 12:37:43.349 I PipelineStage FTRMix 55.95 %/0.36
2021/07/25 12:37:43.349 I PipelineStage FTR1 58.33 %/0.33
2021/07/25 12:37:43.349 I PipelineStage HTDM 52.38 %/0.77
2021/07/25 12:37:43.349 I PipelineStage HTDE 50%/1
2021/07/25 12:37:43.349 I PipelineStage HTDMin 59.52 %/0.61
2021/07/25 12:37:43.349 I PipelineStage HTDC 54.76 %/0.09 2021/07/25 12:37:43.349 I PipelineStage HTZ 47.62 %/0.61
2021/07/25 12:37:43.349 I PipelineStage HTHM 59.52 %/0.25
2021/07/25 12:37:43.349 I PipelineStage HTHE 57.14 %/0.29
2021/07/25 12:37:43.349 I PipelineStage HTHMin 58.93 %/0.21
2021/07/25 12:37:43.349 I PipelineStage HTHC 80.36 %/0.27
2021/07/25 12:37:43.349 I PipelineStage HTHZ 45.24 %/0.04
2021/07/25 12:37:43.349 I PipelineStage HTDIE 45.24 %/0.18
2021/07/25 12:37:43.349 I PipelineStage HTDIL 43.45 %/0.18
2021/07/25 12:37:43.349 I PipelineStage HTRAll 55.95 %/0.35
2021/07/25 12:37:43.361 I PipelineStage HTRMix 58.33 %/0.35 2021/07/25 12:37:43.361 I PipelineStage HTR1 58.33 %/0.32 2021/07/25
12:37:43.361 I Program READY.
```

El experimento IDENTIFY comienza por particionar las muestras del usuario en oraciones, y quedarse con aquellas de largo suficiente y que tengan al menos 70% de caracteres alfanuméricos. Luego, para cada una de ellas genera desafíos con muestras de texto cambiadas, y utiliza a robot bueno y robot malo para generar el conjunto de entrenamiento, como se ha descrito en la sección 4.5. Luego de la evaluación se muestra si la oración fue identificada correctamente, y la lista de los falsos positivos detectados durante la evaluación.

Listado D.6: Salida para una oración del experimento IDENTIFY

```
2021/07/25 11:42:11.604 I PipelineStage DATASET LSIA
2021/07/25 11:42:11.612 I PipelineStage USER 2250574
2021/07/25 11:42:11.612 I PipelineStage Splitting legitimate samples into sentences . . .
2021/07/25 11:42:26.195 I PipelineStage SENTENCE 0.21 , SE AS[BACK]CUDE HABITACION , SE ENCUENTRA PACIENTE
EN REGULAR ESTADO GENERAL, SE REALIZA ELEVACION DE MIEMBROS INFERIORES ,
SE ADMINSITRA EXTRAPLAN 500 CC S[BACK]DE .SSN. 0.9[ LSHIFT]5 , SE REALIZA RX DET[BACK] [BACK] [BACK] [BACK] [BACK] [BACK] [BACK] [BACK] [BACK] [BACK] [BACK] [BACK] [BACK] [BACK] [BACK] [BACK] [BACK]
[BACK]SE INDICA OXIGENO POR BID[BACK]B[BACK]GOTERA
1.5 LIT7 [BACK]/MIN, SE REALIZA RX DE TORX[BACK]AX MALA
TECNICA CON BORA[BACK]RAMIENTO DE SENOS COSTODIAFRAGMATICOS.
2021/07/25 11:42:26.195 I PipelineStage Generating text challenges . . .
2021/07/25 11:42:26.211 I PipelineStage Sampling training sentences . . .
2021/07/25 11:42:26.211 I PipelineStage Generating training instances with GOOD BOT and EVIL BOT . . .
2021/07/25 11:42:26.211 I PipelineStage Training user profile but without this sentence . . .
2021/07/25 11:42:27.083 I PipelineStage Saving training . . .
2021/07/25 11:42:27.083 I PipelineStage Evaluating sentence and challenges . . .
2021/07/25 11:42:27.669 I PipelineStage *** FAILED TO AUTHENTICATE LEGITIMATE SENTENCE *** 2021/07/25 11:42:27.669 I PipelineStage One false negative will be added . 2021/07/25
11:42:27.669 I PipelineStage 0 false positives between challenges .
```

D.7. Archivos de entrenamiento

Al finalizar los experimentos VERIFY y IDENTIFY, en el directorio Output se almacenan los archivos ARFF de entrenamiento y evaluación de los modelos.

Estos contienen un atributo para cada atributo derivado y una cantidad balanceada de instancias de entrenamiento, en donde la mitad han sido etiquetadas como legítimas y la otra mitad como impostores. Según los atributos derivados que se haya elegido calcular en los archivos de configuración, la estructura de encabezados de los archivos ARFF puede variar. Por defecto, tal es la que se muestra a continuación.

Listado D.7: Encabezados de los archivos ARFF de entrenamiento

```
@RELATION ksd
@ATTRIBUTE FT DM NUMERIC @ATTRIBUTE FTDE NUMERIC
@ATTRIBUTE FTDMin NUMERIC
@ATTRIBUTE FTDC NUMERIC @ATTRIBUTE FTZ NUMERIC
@ATTRIBUTE FTHM NUMERIC @ATTRIBUTE FTHE NUMERIC
@ATTRIBUTE FTHMin NUMERIC
@ATTRIBUTE FTHC NUMERIC @ATTRIBUTE FTHZ NUMERIC
@ATTRIBUTE FTDIE NUMERIC
@ATTRIBUTE FTDIL NUMERIC
@ATTRIBUTE FTRAll NUMERIC @ATTRIBUTE FTRMix NUMERIC
@ATTRIBUTE FTR1 NUMERIC
@ATTRIBUTE HT DM NUMERIC @ATTRIBUTE HT DE NUMERIC
@ATTRIBUTE HT DMin NUMERIC
@ATTRIBUTE HT DC NUMERIC @ATTRIBUTE HT Z NUMERIC
@ATTRIBUTE HTHM NUMERIC @ATTRIBUTE HT HE NUMERIC
@ATTRIBUTE HT HMin NUMERIC
@ATTRIBUTE HT HC NUMERIC @ATTRIBUTE HT HZ NUMERIC
@ATTRIBUTE HTDIE NUMERIC
@ATTRIBUTE HTDIL NUMERIC
@ATTRIBUTE HTRAll NUMERIC @ATTRIBUTE HTRMix NUMERIC
@ATTRIBUTE HTR1 NUMERIC
@ATTRIBUTE RESULT {legitimate , impostor}
```

El atributo de clase se denomina RESULT y es el último de la lista anterior.

D.8. Archivos de configuración

Los archivos de configuración de los experimentos se encuentran en el directorio

D.8. Archivos de configuración

Experiments/CommandLine y permiten modificar distintas características de la ejecución, como los atributos derivados a utilizar, el orden de los contextos, y la limpieza de datos, sin cambiar el algoritmo. Los mismos se encuentran escritos en formato XML y siguen el estándar para archivos de configuración de .NET Framework 4.7.2 y .NET Core 3.1.

Comienzan con una sección que declara las secciones de configuración y asignan tipos internos a las mismas.

Listado D.8: Encabezado del archivo de configuración

```
<configSections>
        <section name="biometricParameters"          ...
      <section name="modelsAll"          ...
       <section name="featuresAll "          ...
       <section name=" attributesAll "          ...
         <section name="finiteContextsExperiment"          ...

         <section name="pipelineEmpiricalDistances "          ...
</configSections>
```

La sección *appSettings* incluye parámetros generales, como el máximo valor aceptable en una componente de las distancias, y si mostrar por salida estándar la línea de comandos con la que se invoca a WEKA.

Listado D.9: Sección appSettings del archivo de configuración

```
<appSettings>
        <add key="distance . maxComponentValue"          value="12" />
   <add key="verbose .WEKAcommands" value=" false " />
</appSettings>
```

La sección *biometricParameters* declara que atributos se utilizarán. Aquí hemos utilizado tiempos de retención y latencia.

Listado D.10: Sección biometric Parameters del archivo de configuración

```
<biometricParameters name="INDEPENDENT TIMING PARAMETERS">
   <biometricParameter name="HT" />
   <biometricParameter name="FT" />
</biometricParameters>
```

La sección *modelsAll* declara los tipos de modelos subyacentes. Aquí utilizamos cuatro de ellos: un modelo que calcula valores promedio y desvío estándar de los S_i (AvgStdevModelLinear), otro que emplea el histograma emprico (HistogramModel), un tercero para el índice de direccionalidad con medias exponenciales (SimpleExponentialDirectionalityModel), y el ultimo para el índice de direccionalidad con medias móviles lineales (SimpleLinearDirectionalityModel).

Listado D.11: Sección models All del archivo de configuración

```
<modelsAll name="ALL MODELS">
          <modelSet name="TD" type="AvgStdevModelLinear"          ...
        <modelSet name="HI" type="HistogramModel"          ...
       <modelSet name="DIE" type="SimpleExponentialDirectionalityModel"       ... <modelSet name="DIL"   ...
</modelsAll>
```

La sección *featuresAll* declara las familias de atributos derivados que se calcularan para las instancias de entrenamiento.

Listado D.12: Sección features All del archivo de configuración

```
<featuresAll name="ALL FEATURES">
          <feature name="AvgStdev" type="AvgStdevFeatures"          pattern="HT TD; FT TD" />
```

D.8. Archivos de configuración

```
<feature name="Histogram" type="HistogramFeatures" pattern="HT HI ; FT HI" />
<feature name="DIE" type="DIFeatures" pattern="HT DIE ; FT DIE" />
<feature name="DIL" type="DIFeatures" pattern="HT DIL ; FT DIL" />
<feature name="RMix" type="RFeature" pattern="HT TD; FT TD" parameters="hash" />
<feature name="R1" type="RFeature" pattern="HT TD; FT TD" parameters="ngram" />
</featuresAll>
```

La sección *attributes All* declara los atributos que se calcularán para cada familia de las anteriores.

Listado D.13: Sección attributes All del archivo de configuración

```
<attributesAll name="ALL ATTRIBUTES">
<attribute name="FT DM" type="ManhattanDistance" source="FT TD tms ; FT TD avg ; FT TD std" />
<attribute name="FTDE" type="EuclideanDistance" source="FT TD tms ; FT TD avg ; FT TD std" />
<attribute name="FTDMin" type="MinkowskiDistance" source="FT TD tms ; FT TD avg ; FT TD std" parameters ="0.4" />
<attribute name="FTDC" type="CanberraDistance" source="FT TD tms ; FT TD avg" />
<attribute name="FTZ" type="Z" source="FT TD tms ; FT TD avg ; FT TD std" />
<attribute name="FTHM" type="ManhattanDistance" source="FT HI tms ; FT HI avg ; FT HI std" />
<attribute name="FTHE" type="EuclideanDistance" source="FT HI tms ; FT HI avg ; FT HI std" />
<attribute name="FTHMin" type="MinkowskiDistance" source="FT HI tms ; FT HI avg ; FT HI std" parameters ="0.4" />
<attribute name="FTHC" type="CanberraDistance" source="FT HI tms ; FT HI avg" />
<attribute name="FTHZ" type="HZ" source="FT HI avg" parameters ="0.025" /> <attribute name="FTDIE" type="DI" source="FT TD tms ;
FT DIE dir" />
<attribute name="FTDIL" type="DI" source="FT TD tms ; FT DIL dir" />
<attribute name="FTRAll" type="R" source="FT TD tms ; FT TD avg" />
<attribute name="FTRMix" type="CopyNumericFeature" source="FT TD RMix" />
<attribute name="FTR1" type="CopyNumericFeature" source="FT TD R1" />

<attribute name="HT DM" type="ManhattanDistance" source="HT TD tms ; HT TD avg ; HT TD std" />
<attribute name="HT DE" type="EuclideanDistance" source="HT TD tms ; HT TD avg ; HT TD std" />
<attribute name="HT DMin" type="MinkowskiDistance" source="HT TD tms ; HT TD avg ; HT TD std" parameters ="0.4" />
<attribute name="HT DC" type="CanberraDistance" source="HT TD tms ; HT TD avg" />
<attribute name="HT Z" type="Z" source="HT TD tms ; HT TD avg ; HT TD std" />
<attribute name="HTHM" type="ManhattanDistance" source="HT HI tms ; HT HI avg ; HT HI std" />
<attribute name="HT HE" type="EuclideanDistance" source="HT HI tms ; HT HI avg ; HT HI std" />
<attribute name="HT HMin" type="MinkowskiDistance" source="HT HI tms ; HT HI avg ; HT HI std" parameters ="0.4" />
<attribute name="HT HC" type="CanberraDistance" source="HT HI tms ; HT HI avg" />
<attribute name="HT HZ" type="HZ" source="HT HI avg" parameters ="0.025" /> <attribute name="HTDIE" type="DI" source="HT TD tms
; HT DIE dir" />
<attribute name="HTDIL" type="DI" source="HT TD tms ; HT DIL dir" />
<attribute name="HTRAll" type="R" source="HT TD tms ; HT TD avg" />
<attribute name="HTRMix" type="CopyNumericFeature" source="HT TD RMix" />
<attribute name="HTR1" type="CopyNumericFeature" source="HT TD R1" />
</attributesAll>
```

La sección *finite Contexts Experiment* parametriza el modelado por contextos finitos que subyace a todos los métodos.

Listado D.14: Sección finite Contexts Experiment del archivo de configuración

```
<finiteContextsExperiment name="basic " initialTrainingSessions ="50" retrain ="true"
calculateEERs="true" biometricParameters="biometricParameters"
models="modelsAll"
features="featuresAll " attributes =" attributesAll "

profileBehavior="NotTheEvaluatedSession" minTrainingSessions="3"
minUserSessions="4" />
```

Finalmente, la sección *pipeline Empirical Distances* determina las etapas del *pipeline* de procesamiento y los conjuntos de datos que se cargan.

Listado D.15: Sección *pipeline* Empirical Distances del archivo de configuración

```
<pipelineEmpiricalDistances parallel =" false " name="IDENTIFY">
<stage name="loadLSIA" type="LoadDatasetStage" action="BinaryDatasetReader" parameters="LSIA" f i l e ="DATASETS/LSIA . bin" />
<stage name="loadKM" type="LoadDatasetStage" f i l e action="BinaryDatasetReader" parameters="KM"
="DATASETS/KM. bin" />
<stage name="loadGAY" type="LoadDatasetStage" f i l e action="BinaryDatasetReader" parameters="GAY"
="DATASETS/PROSODY GAY. bin" />
<stage name="loadGUN" type="LoadDatasetStage" f i l e action="BinaryDatasetReader" parameters="GUN"
="DATASETS/PROSODY GUN. bin" />
<stage name="loadREVIEW" type="LoadDatasetStage" f i l e action="BinaryDatasetReader" parameters="REVIEW"
="DATASETS/PROSODY REVIEW. bin" />
```

```
<stage name=" s p l i t " type="ThresholdPartitioner " />
   <stage name=" f t s "              type="CleanFTs" />
<stage name="run"  type="RunDefaultExperiment" /> </pipelineEmpiricalDistances>
```

D.9. Extendiendo la herramienta

Además de modificar los parámetros de los experimentos utilizando los archivos de configuración, es simple realizar extensiones y modificaciones al resto del comportamiento. En este caso es necesario modificar el código fuente. A continuación, se describen algunas de las modificaciones que se espera sean las más comunes.

D.9.1. Carga de nuevos conjuntos de datos

La carga de conjuntos de datos se realiza por medio de la clase *BinaryDatasetReader*. El conjunto de datos debe ser migrado al formato binario que deserializa esta clase antes de poder ser utilizado en la herramienta. Luego, debe agregarse al archivo de configuración, dentro de la sección *pipeline*, la línea

Listado D.16: Agregando un nuevo conjunto de datos en formato binario

```
<stage name="loadDATASET" type="LoadDatasetStage"
      action="BinaryDatasetReader" parameters="DATASET"
      f i l e ="DATASETS/DATASET. bin" />
```

reemplazando DATASET por el nombre del conjunto de datos y el atributo file por la ruta del archivo correspondiente.

Si no se desea migrar el formato del conjunto de datos, debe implementarse una clase que implemente la interfaz IDatasetReader para leerlo.

Listado D.17: Agregando un nuevo conjunto de datos en otro formato

```
<stage name="loadDATASET" type="LoadDatasetStage"
      action="MiClaseDerivadaDeIDatasetReader" parameters="DATASET"
      f i l e ="DATASETS/DATASET. bin" />
```

reemplazando DATASET por el nombre del conjunto de datos, el atributo file por la ruta del archivo correspondiente, y el nombre MiClaseDerivadaDeIDatasetReader por el nombre de la clase que implementa la deserialización.

D.9.2. Creación de nuevas estrategias de síntesis

Las estrategias de síntesis utilizadas en el experimento VERIFY se encuentran implementadas en las clases de sufijo *Synthesizer*. Todas ellas derivan de la clase *LocalForwardSynthesizer*, que devuelve un valor de atributo temporal sucesivo por cada invocación a sus métodos.

Estos son dos: *OnNullModel*, que es invocado cuando no se cuenta con un conjunto S_i que cumpla las condiciones de filtro, y *OnModelFound*, que se invoca con el conjunto S_i completo.

En caso de que sea necesario implementar estrategias no locales de síntesis, se puede implementar una clase derivada de *KeystrokeDynamicsSynthesizer*. En este caso, el método *SynthesizeFeature* debe devolver el vector completo para el atributo temporal solicitado.

D.9.3. Creación de nuevos atributos derivados

La herramienta admite ser extendida agregando nuevos atributos derivados. Por ejemplo, las distancias de Manhattan, euclídea, de Canberra, y de Minkowski son todas clases que implementan el método *GetValue* de la interfaz *INumericAttribute*, recibiendo los vectores de atributos temporales. Idéntico es el caso de la métrica R, que esta implementada en la clase homónima, y el índice de direccionalidad DI.

Para crear un nuevo atributo derivado e incluirlo en los experimentos, se debe crear una nueva clase que implemente la interfaz *INumericAttribute* e implementar el método *GetValue*. Luego, agregar en la sección attributesAll del archivo de configuración la línea

Listado D.18: Agregando un nuevo atributo derivado

```
<attribute name=”ATRIBUTO”
    type=”ClaseImplementaINumericAttribute” source=”FEATURES”
    />
```

en donde ATRIBUTO es el nombre del atributo como aparecerá en los archivos ARFF de salida, *ClaseImplementaINumericAttribute* es el nombre de la clase que lo implementa, y *FEATURES* son las familias de atributos de las cuales toma datos.

D.9.4. Creación de nuevos experimentos

Las clases SYNTHESIZE, VERIFY, y IDENTIFY, que implementan los tres experimentos soportados por la herramienta, derivan de la clase Experiment. Para crear un nuevo experimento, basta con crear una nueva clase derivada de Experiment e implementar algunos de sus métodos virtuales.

Los métodos OnStageStart y OnStageEnd se ejecutan al inicio y al final de la etapa del experimento. ForEachDataset, ForEachUser, y ForEachSession son invocado respectivamente para cada conjunto de datos cargado, para cada usuario, y para cada sesión de usuario.

Bibliografía

[1] Michael G. Solomon et. al. David Kim. *Fundamentals of information systems security*. Jones & Bartlett Publishers, 2013.

[2] Sergio Roberto de Lima e Silva, Mauro Roisenberg et al. Continuous authentication by keystroke dynamics using committee machines. En: *International Conference on Intelligence and Security Informatics*. Springer. 2006, págs. 686-687.

[3] Eric Flior y Kazimierz Kowalski. Continuous biometric user authentication in online examinations. En: *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*. IEEE. 2010, págs. 488-492.

[4] Martha Mohlala, Adeyemi R Ikuesan y Hein S Venter. User attribution based on keystroke dynamics in digital forensic readiness process. En: *2017 IEEE Conference on Application, Information and Network Security (AINS)*. IEEE. 2017, págs. 124-129.

[5] Kevin S. Killourhy y Roy A. Maxion. Comparing Anomaly-Detection Algorithms for Keystroke Dynamics. en. En: *International Conference on Dependable Systems & Networks (DSN-09)*. EEE Computer Society Press, Los Alamitos, California, 2009, págs. 125-134.

[6] Kevin S Maxion Roy A y Killourhy. Keystroke biometrics with number-pad input. En: *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*. IEEE. 2010, págs. 201-210.

[7] R Stockton Gaines, William Lisowski, S James Press y Norman Shapiro. *Authentication by keystroke timing: Some preliminary results*. Inf. téc. Rand Corp Santa Monica CA, 1980.

[8] Daniele Gunetti y Claudia Picardi. Keystroke analysis of free text. En: *ACM Transactions on Information and System Security (TISSEC)* 8.3 (2005), págs. 312-347.

[9] Dwijen Rudrapal y Smita Das. Continuous Authentication Based on Language Redundancy Analysis and Keystrokes Dynamics. En: (2013).

[10] *Information technology — Biometric presentation attack detection*. Standard. Geneva, CH: International Organization for Standardization, 2017.

[11] John V Monaco y Charles C Tappert. Obfuscating keystroke time intervals to avoid identification and impersonation. En: *arXiv preprint arXiv:1609.07612* (2016).

[12] Itay Hazan, Oded Margalit y Lior Rokach. Keystroke dynamics obfuscation using key grouping. En: *Expert Systems with Applications* 143 (2020), pág. 113091.

[13] David Slater, Scott Novotney, Jessica Moore, Sean Morgan y Scott Tenaglia. Robust keystroke transcription from the acoustic side-channel. En: *Proceedings of the 35th Annual Computer Security Applications Conference*. 2019, págs. 776-787.

[14] Muzammil Hussain, Ahmed Al-Haiqi, AA Zaidan, BB Zaidan, ML Mat Kiah, Nor Badrul Anuar y Mohamed Abdulnabi. The rise of keyloggers on smartphones: A survey and insight into motion-based tap inference attacks. En: *Pervasive and Mobile Computing* 25 (2016), págs. 1-25.

[15] J. V. Monaco. SoK: Keylogging Side Channels. En: *2018 IEEE Symposium on Security and Privacy (SP)*. 2018, págs. 211-228. doi: 10.1109/SP.2018.00026.

[16] Antony Milne, Katayoun Farrahi y Mihalis A Nicolaou. Less is more: Univariate modelling to detect early Parkinson's disease from keystroke dynamics. En: *International Conference on Discovery Science*. Springer. 2018, págs. 435-446.

[17] Clayton Epp, Michael Lippold y Regan L Mandryk. Identifying emotional states using keystroke dynamics. En: *Proceedings of the sigchi conference on human factors in computing systems*. 2011, págs. 715-724.

[18] Salil P Banerjee y Damon L Woodard. Biometric authentication and identification using keystroke dynamics: A survey. En: *Journal of Pattern Recognition Research* 7.1 (2012), págs. 116-139.

[19] William Lowe Bryan y Noble Harter. Studies in the physiology and psychology of the telegraphic language. En: *Psychological Review* 4.1 (1897), pág. 27.

[20] Malcolm Gladwell y Madelon E. Ruiter. Blink: The power of thinking without thinking. En: *Gedragstherapie* 41.2 (2008), pág. 199.

[21] Kevin S Killourhy y Roy A Maxion. Should security researchers experiment more and draw more inferences? En: *CSET*. 2011.

[22] Kevin S Killourhy y Roy A Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. En: *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*. IEEE. 2009, págs. 125-134.

[23] Vivek Dhakal, Anna Feit, Per Ola Kristensson y Antti Oulasvirta. Observations on Typing from 136 Million Keystrokes. En: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, 2018. doi: https://doi.org/10.1145/3173574.3174220.

[24] Alejandro Acien, Aythami Morales, Ruben Vera-Rodriguez, Julian Fierrez y John V Monaco. Typenet: Scaling up keystroke biometrics. En: *2020 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE. 2020, págs. 1-7.

[25] Ioannis Stellios, Panayiotis Kotzanikolaou, Mihalis Psarakis, Cristina Alcaraz y Javier Lopez. A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services. En: *IEEE Communications Surveys & Tutorials* 20.4 (2018), págs. 3453-3495.

[26] Khandaker A Rahman, Kiran S Balagani y Vir V Phoha. Making impostor pass rates meaningless: A case of snoop-forge-replay attack on continuous cyber-behavioral verification with keystrokes. En: *CVPR 2011 workshops*. IEEE. 2011, págs. 31-38.

[27] Deian Stefan y Danfeng Yao. Keystroke-dynamics authentication against synthetic forgeries. En: *6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2010)*. IEEE. 2010, págs. 1-8.

[28] John V Monaco, Md Liakat Ali y Charles C Tappert. Spoofing key-press latencies with a generative keystroke dynamics model. En: *2015 IEEE 7th international conference on biometrics theory, applications and systems (BTAS)*. IEEE. 2015, págs. 1-8.

[29] Deian Stefan, Xiaokui Shu y Danfeng Daphne Yao. Robustness of keystrokedynamics based biometrics against synthetic forgeries. En: *computers & security* 31.1 (2012), págs. 109-121.

[30] Alejandro Acien, Aythami Morales, John V Monaco, Ruben Vera-Rodriguez y Julian Fierrez. TypeNet: Deep Learning Keystroke Biometrics. En: *arXiv preprint arXiv:2101.05570* (2021).

[31] Francesco Bergadano, Daniele Gunetti y Claudia Picardi. User authentication through keystroke dynamics. En: *ACM Transactions on Information and System Security (TISSEC)* 5.4 (2002), págs. 367-397.

[32] Kevin S Killourhy y Roy A Maxion. Free vs. transcribed text for keystrokedynamics evaluations. En: *Proceedings of the 2012 Workshop on Learning from Authoritative Security Experiment Results*. 2012, págs. 1-8.

[33] Nahuel González y Enrique P. Calot. Finite Context Modeling of Keystroke Dynamics in Free Text. en. En: *Biometrics Special Interest Group (BIOSIG), 2015 International Conference of the*. Sep. de 2015, págs. 1-5. isbn: 978-3-88579-639-8. doi: 10.1109/BIOSIG.2015.7314606. url: https://ieeexplore.ieee.org/ document/7314606.

[34] Nahuel González, Enrique P Calot y Jorge S Ierache. A replication of two free text keystroke dynamics experiments under harsher conditions. En: *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE. 2016, págs. 1-6. isbn: 978-1-50900-780-6. doi: 10.1109/BIOSIG.2016.7736905. url: https://ieeexplore.ieee.org/document/7736905.

[35] Enrique P Calot, Jorge S Ierache y Waldo Hasperué. Document typist identification by classification metrics applying keystroke dynamics under unidealised conditions. En: *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*. Vol. 8. IEEE. 2019, págs. 19-24.

[36] Christopher Murphy, Jiaju Huang, Daqing Hou y Stephanie Schuckers. Shared dataset on natural human-computer interaction to support continuous authentication research. En: *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE. 2017, págs. 525-530.

[37] Ritwik Banerjee, Song Feng, Jun Seok Kang y Yejin Choi. Keystroke patterns as prosody in digital writings: A case study with deceptive reviews and essays. En: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, págs. 1469-1473.

[38] Yan Sun, Hayreddin Ceker y Shambhu Upadhyaya. Shared keystroke dataset for continuous authentication. En: *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE. 2016, págs. 1-6.

[39] Esra Vural, Jiaju Huang, Daqing Hou y Stephanie Schuckers. Shared research dataset to support development of keystroke authentication. En: *IEEE International joint conference on biometrics*. IEEE. 2014, págs. 1-8.

[40] Kevin S. Killourhy. *A Scientific Understanding of Keystroke Dynamics*. en. Inf. téc. DTIC Document, 2012. url: http://lsia.fi.uba.ar/papers/killourhy12. pdf.

[41] Vincent Monaco. *Keystroke Dynamics Datasets*. url: https://vmonaco.com/ datasets/.

[42] Charles Slivinsky y Bassam Hussien Saleh Bleha. Computer-access security systems using keystroke dynamics. en. En: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 12.12 (1990), págs. 1217-1222. url: http://lsia.fi.uba.ar/papers/bleha90.pdf.

[43] Juan Manuel Rodriguez y Jorge Salvador Ierache Enrique P. Calot. Improving versatility in keystroke dynamic systems. en. En: *XIX Congreso Argentino de Ciencias de la Computación*. 5606. 2013. isbn: 978-987-23963-1-2. url: http://ir.cs.uns.edu.ar/downloads/cacic_2013/11wsi.pdf.

[44] Livia C. F. Araujo et. al. User authentication through typing biometrics features. en. En: *Signal Processing, IEEE Transactions on* 53.2 (2005), págs. 851-855. url: http://lsia.fi.uba.ar/papers/araujo05.pdf.

[45] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. En: National Institute of Science of India. 1936.

[46] Dae Hee Han y Hyung-Il Kim Sungzoon Cho Chigeun Han. Web based Keystroke Dynamics Identity Verification using Neural Network. en. En: *Journal of Organizational Computing and Electronic Commerce* 10.4 (2000), págs. 295-307. url: http://lsia.fi.uba.ar/papers/cho00.pdf.

[47] Don Gingrich y Andy Sentosa Jiankun Hu. A k-nearest neighbor approach for user authentication through biometric keystroke dynamics. En: *Communications, 2008. ICC'08. IEEE International Conference on*. IEEE. 2008, págs. 1556-1560.

[48] Balqies Obaidat Mohammad S y Sadoun. Verification of computer users using keystroke dynamics. En: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 27.2 (1997), págs. 261-269.

[49] Ahmed Abbas y Abbas K. Zaidi Sajjad Haider. A multi-technique approach for user identification through keystroke dynamics. En: *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*. Vol. 2. IEEE. 2000, págs. 1336-1341.

[50] Fabian Monrose y Aviel Rubin. Authentication via keystroke dynamics. En: *Proceedings of the 4th ACM conference on Computer and communications security*. ACM. 1997, págs. 48-56.

[51] John Leggett y Glen Williams. Verifying identity via keystroke characterstics. En: *International Journal of Man-Machine Studies* 28.1 (1988), págs. 67-76.

[52] Seyit Ahmet Camtepe y Sahin Albayrak Arik Messerman Tarik Mustafic. Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics. En: *Biometrics (IJCB), 2011 International Joint Conference on*. IEEE. 2011, págs. 1-8.

[53] Enzhe Yu y Sungzoon Cho. GA-SVM wrapper approach for feature subset selection in keystroke dynamics identity verification. En: *Neural Networks, 2003. Proceedings of the International Joint Conference on*. Vol. 3. IEEE. 2003, págs. 2253-2257.

[54] Ki-seok Sung y Sungzoon Cho. GA SVM wrapper ensemble for keystroke dynamics authentication. En: *Advances in Biometrics*. Springer, 2005, págs. 654-660.

[55] Hayreddin C¸ eker y Shambhu Upadhyaya. User authentication with keystroke dynamics in long-text data. En: *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE. 2016, págs. 1-6.

[56] G. E. Hinton y R. J. Williams D. E. Rumelhart. *Learning Internal Representations by Error Propagation, Parallel Distributed Processing, Explorations in the Microstructure of Cognition, ed. DE Rumelhart and J. McClelland. Vol. 1. 1986*. 1986.

[57] JA Anderson. *An introduction to neural networks. A bradford book*. 1995.

[58] Vir Phoha y Steven M. Rovnyak Yong Sheng. A parallel decision tree-based method for user authentication based on keystroke patterns. En: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 35.4 (2005), págs. 826-833.

[59] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah,´ Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu y Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. url: https://www.tensorflow.org/.

[60] Debayan Deb, Arun Ross, Anil K Jain, Kwaku Prakah-Asante y K Venkatesh Prasad. Actions speak louder than (pass) words: Passive authentication of smartphone* users via deep temporal features. En: *2019 International Conference on Biometrics (ICB)*. IEEE. 2019, págs. 1-8.

[61] Thian Song Ong y Han Foon Neo Pin Shen Teh A. Teoh. Statistical fusion approach on keystroke dynamics. En: *Signal-Image Technologies and InternetBased System, 2007. SITIS'07. Third International IEEE Conference on*. IEEE. 2007, págs. 918-923.

[62] Lina Zhou y Andrew Sears Lisa M. Vizer. Automated stress detection using keystroke and linguistic features: An exploratory study. En: *International Journal of Human-Computer Studies* 67.10 (2009), págs. 870-886.

[63] A Kolakowska. A review of emotion recognition methods based on keystroke dynamics and mouse movements. En: *Human System Interaction (HSI), 2013 The 6th International Conference on*. IEEE. 2013, págs. 548-555.

[64] Enrique P Calot. Robustez de las métricas de clasificación de cadencia de tecleo frente a variaciones emocionales. Tesis doct. Universidad Nacional de La Plata, 2019.

[65] Enrique P Calot, Jorge S Ierache y Waldo Hasperué. Robustness of keystroke dynamics identification algorithms against brain-wave variations associated with emotional variations. En: *Proceedings of SAI Intelligent Systems Conference*. Springer. 2019, págs. 194-211.

[66] Rick Joyce y Gopal Gupta. Identity authentication based on keystroke latencies. en. En: *Commun. ACM* 33.2 (feb. de 1990), pags.´ 168-176. issn: 0001-0782. doi: 10.1145/75577.75582. url: http://doi.acm.org/10.1145/75577.75582.

[67] Michael K. Reiter y Susanne Wetzel Fabian Monrose. Password Hardening Based on Keystroke Dynamics. en. En: *Proceedings of the 6th ACM Conference on Computer and Communications Security*. CCS '99. Kent Ridge Digital Labs, Singapore: ACM, 1999, págs. 73-82. isbn: 1-58113-148-8. doi: 10.1145/319709. 319720. url: http://doi.acm.org/10.1145/319709.319720.

[68] Gene H. Golub y Randall J. LeVeque Tony F. Chan. Updating formulae and a pairwise algorithm for computing sample variances. en. En: *COMPSTAT 1982 5th Symposium held at Toulouse 1982*. Springer. 1982, págs. 30-41. url: http://lsia.fi.uba.ar/papers/chan82.pdf.

[69] Seong-seob Hwang y Sungzoon Cho Pilsung Kang. Continual retraining of keystroke dynamics based authenticator. En: *Advances in Biometrics*. Springer, 2007, págs. 1203-1211.

[70] Fabian Monrose, Michael K Reiter y Susanne Wetzel. Password hardening based on keystroke dynamics. En: *International journal of Information security* 1.2 (2002), págs. 69-83.

[71] Khandaker A Rahman, Kiran S Balagani y Vir V Phoha. Snoop-forge-replay attacks on continuous verification with keystrokes. En: *IEEE Transactions on information forensics and security* 8.3 (2013), págs. 528-541.

[72] Jugurta R Montalvão Filho y Eduardo O Freire. On the equalization of keystroke timing histograms. En: *Pattern Recognition Letters* 27.13 (2006), págs. 1440-1446.

[73] Jugurta Montalvão, Carlos Augusto S Almeida y Eduardo O Freire. Equalization of keystroke timing histograms for improved identification performance. En: *2006 International telecommunications symposium*. IEEE. 2006, págs. 560-565.

[74] Evgeny Chukharev-Hudilainen. Pauses in spontaneous written communication: A keystroke logging study. En: *Journal of Writing Research* 6.1 (2014), págs. 61-84.

[75] Aamo Iorliam, Anthony TS Ho, Norman Poh, Santosh Tirunagari y Patrick Bours. Data forensic techniques using Benford's law and Zipf's law for keystroke dynamics. En: *Biometrics and Forensics (IWBF), 2015 International Workshop on*. IEEE. 2015, págs. 1-6.

[76] John Vincent Monaco. Time intervals as a Behavioral Biometric. Tesis doct. PACE UNIVERSITY, 2015.

[77] John V Monaco y Charles C Tappert. The Partially Observable Hidden Markov Model with Application to Keystroke Biometrics. En: *arXiv preprint arXiv:1607.03854* (2016).

[78] Denis Migdal y Christophe Rosenberger. Analysis of keystroke dynamics for the generation of synthetic datasets. En: *2018 International Conference on Cyberworlds (CW)*. IEEE. 2018, págs. 339-344.

[79] Denis Migdal y Christophe Rosenberger. Statistical modeling of keystroke dynamics samples for the generation of synthetic datasets. En: *Future Generation Computer Systems* 100 (2019), págs. 907-920.

[80] Albert-Laszlo Barabasi. The origin of bursts and heavy tails in human dynamics. En: *Nature* 435.7039 (2005), págs. 207-211.

[81] Andrew Heathcote, Stephen J Popiel y DJ Mewhort. Analysis of response time distributions: An example using the Stroop task. En: *Psychological Bulletin* 109.2 (1991), pág. 340.

[82] Marco Steinhauser y Ronald Hubner.¨ Distinguishing response conflict and task conflict in the Stroop task: evidence from ex-Gaussian distribution analysis. En: *Journal of Experimental Psychology: Human Perception and Performance* 35.5 (2009), pág. 1398.

[83] Romain Giot, Mohamad El-Abed y Christophe Rosenberger. Greyc keystroke: a benchmark for keystroke dynamics biometric systems. En: *2009 IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems*. IEEE. 2009, págs.1-6.

[84] Romain Giot, Mohamad El-Abed y Christophe Rosenberger. Web-based benchmark for keystroke dynamics biometric systems: A statistical analysis. En: *2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE. 2012, págs. 11-15.

[85] John V Monaco, Ned Bakelman, Sung-Hyuk Cha y Charles C Tappert. Recent advances in the development of a long-text-input keystroke biometric authentication system for arbitrary text input. En: *2013 European Intelligence and Security Informatics Conference*. IEEE. 2013, págs. 60-66.

[86] Jonathan Ness. Presentation Attack and Detection in Keystroke Dynamics. Tesis de mtría. NTNU, 2017.

[87] Itay Hazan, Oded Margalit y Lior Rokach. Securing keystroke dynamics from replay attacks. En: *Applied Soft Computing* 85 (2019), pág. 105798.

[88] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic y Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. En: *IEEE transactions on information forensics and security* 8.1 (2012), págs. 136-148.

[89] Valeriu-Daniel Stanciu, Riccardo Spolaor, Mauro Conti y Cristiano Giuffrida. On the effectiveness of sensor-enhanced keystroke dynamics against statistical attacks. En: *proceedings of the sixth ACM conference on data and application security and privacy*. 2016, págs. 105-112.

[90] Xiangyu Liu, Zhe Zhou, Wenrui Diao, Zhou Li y Kehuan Zhang. When good becomes evil: Keystroke inference with smartwatch. En: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 2015, págs. 1273-1285.

[91] Tzipora Halevi y Nitesh Saxena. Keyboard acoustic side channel attacks: exploring realistic and security-sensitive scenarios. En: *International Journal of Information Security* 14.5 (2015), págs. 443-456.

[92] John V Monaco. What are you searching for? a remote keylogging attack on search engine autocomplete. En: *28th {USENIX} Security Symposium ({USENIX} Security 19)*.2019, págs. 959-976.

[93] Dawn Xiaodong Song, David A Wagner y Xuqing Tian. Timing analysis of keystrokes and timing attacks on ssh. En: *USENIX Security Symposium*. Vol. 2001. 2001.

[94] Kehuan Zhang y XiaoFeng Wang. Peeping Tom in the Neighborhood: Keystroke Eavesdropping on Multi-User Systems. En: *USENIX Security Symposium*. Vol. 20. 2009, pág. 23.

[95] Moritz Lipp, Daniel Gruss, Michael Schwarz, David Bidner, Clémentine Maurice y Stefan Mangard. Practical keystroke timing attacks in sandboxed javascript. En: *European Symposium on Research in Computer Security*. Springer. 2017, págs. 191-209.

[96] Kiran S Balagani, Mauro Conti, Paolo Gasti, Martin Georgiev, Tristan Gurtler, Daniele Lain, Charissa Miller, Kendall Molas, Nikita Samarin, Eugen Saraci *et al*. Silk-tv: Secret information leakage from keystroke timing videos. En: *European Symposium on Research in Computer Security*. Springer. 2018, págs. 263-280.

[97] John V Monaco. Poster: The side channel menagerie. En: *Proc. IEEE Symp. on Security & Privacy (SP). IEEE*. 2018.

[98] Ximing Liu, Yingjiu Li, Robert H Deng, Bing Chang y Shujun Li. When Human cognitive modeling meets PINs: User-independent inter-keystroke timing attacks. En: *Computers & Security* 80 (2019), págs. 90-107.

[99] John Cleary y Ian Witten. Data compression using adaptive coding and partial string matching. En: *IEEE transactions on Communications* 32.4 (1984), págs. 396-402.

[100] Timothy Bell, Ian H Witten y John G Cleary. Modeling for text compression. En: *ACM Computing Surveys (CSUR)* 21.4 (1989), págs. 557-591.

[101] Alistair Moffat. Implementing the PPM data compression scheme. En: *IEEE Transactions on communications* 38.11 (1990), págs. 1917-1921.

[102] John G Cleary y William J Teahan. Unbounded length contexts for PPM. En: *The Computer Journal* 40.2 and 3 (1997), págs. 67-75.

[103] Enrique P Calot. Utilización de contextos finitos para el modelado de cadencias de tecleo en esquemas de autenticación mixta. Tesis doct. Universidad de Buenos Aires, 2016.

[104] Nahuel González, Germán Concilio, Jorge S. Ierache, Enrique P. Calot y Waldo Hasperué. Exploracion de correlaciones internas de los parámetros temporales generados en dinámicas de tecleo. En: *XXVI Congreso Argentino de Ciencias de la Computación (CACIC)*. 2020, págs. 726-735. isbn: 978-987-4417-90-9. url: http://sedici.unlp.edu.ar/handle/10915/113243.

[105] Nahuel González, Germán Concilio, Enrique P. Calot, Jorge S. Ierache y Waldo Hasperué. Exploring Internal Correlations in Timing Features of Keystroke Dynamics at Word Boundaries and Their Usage for Authentication and Identification. En: *Computer Science–CACIC 2020: 26th Argentine Congress, CACIC 2020, San Justo, Buenos Aires, Argentina, October 5–9, 2020, Revised Selected Papers*. Vol. 1. Communications in Computer and Information Science. Springer Nature, págs. 321-332. isbn: 978-3-030-75835-6. doi: 10.1007/978-3-03075836-3_22. url: https://www.springerprofessional.de/en/exploringinternal-correlations-in-timing-features-of-keystroke-/19132746.

[106] John F Finch, Stephen G West y David P MacKinnon. Effects of sample size and nonnormality on the estimation of mediated effects in latent variable models. En: *Structural Equation Modeling: A Multidisciplinary Journal* 4.2 (1997), págs. 87-107.

[107] Frans MJ Willems, Yuri M Shtarkov y Tjalling J Tjalkens. The context-tree weighting method: Basic properties. En: *IEEE transactions on information theory* 41.3 (1995), pags. 653-664.

[108] Frans MJ Willems. The context-tree weighting method: Extensions. En: *IEEE Transactions on Information Theory* 44.2 (1998), págs. 792-798.

[109] Nahuel González, Jorge S. Ierache, Waldo Hasperué y Enrique P. Calot. On the shape of timing distributions in free-text keystroke dynamics profiles. En: *Heliyon* 7.11 (2021). Elsevier. issn: 2405-8440. doi: https://doi.org/10.1016/ j.heliyon.2021.e08413. url: https://www.cell.com/heliyon/fulltext/ S2405-8440(21)02516-0.

[110] Donald R Gentner. Keystroke timing in transcription typing. En: *Cognitive aspects of skilled typewriting*. Springer, 1983, págs. 95-120.

[111] Changxu Wu y Yili Liu. Queuing network modeling of transcription typing. En: *ACM Transactions on Computer-Human Interaction (TOCHI)* 15.1 (2008), págs. 1-45.

[112] Salvatore T March y Gerald F Smith. Design and natural science research on information technology. En: *Decisión support systems* 15.4 (1995), págs. 251-266.

[113] Daniele Gunetti, Claudia Picardi y Giancarlo Ruffo. Dealing with different languages and old profiles in keystroke analysis of free text. En: *Congress of the Italian Association for Artificial Intelligence*. Springer. 2005, págs. 347-358.

[114] Daniele Gunetti, Claudia Picardi y Giancarlo Ruffo. Keystroke analysis of different languages: a case study. En: *International Symposium on Intelligent Data Analysis*. Springer. 2005, págs. 133-144.

[115] Enrique P. Calot. *Keystroke Dynamics keypress latency dataset*. en. Database. Ene. de 2015. url: http://lsia.fi.uba.ar/pub/papers/kd-dataset/.

[116] Anil Jain, Brendan Klare y Arun Ross. Guidelines for best practices in biometrics research. En: *Biometrics (ICB), 2015 International Conference on*. IEEE. 2015, págs. 541-545.

[117] Rosemary A Bailey. *Design of comparative experiments*. Vol. 25. Cambridge University Press, 2008.

[118] Victoria Stodden. The scientific method in practice: Reproducibility in the computational sciences. En: (2010).

[119] Mark L Mitchell y Janina M Jolley. *Research design explained*. 2010.

[120] Galit Shmueli *et al*. To explain or to predict? En: *Statistical science* 25.3 (2010), págs. 289-310.

[121] Preeti Khanna y M Sasikumar. Recognising emotions from keyboard stroke pattern. En: *International journal of computer applications* 11.9 (2010), págs. 1-5.

[122] Ann Gledson, Dommy Asfiandy, Joseph Mellor, Thamer Omer Faraj Ba-Dhfari, Gemma Stringer, Samuel Couth, Alistair Burns, Iracema Leroi, Xiaojun Zeng, John Keane *et al*. Combining mouse and keyboard events with higher level desktop actions to detect mild cognitive impairment. En: *Healthcare Informatics (ICHI), 2016 IEEE International Conference on*. IEEE. 2016, págs. 139-145.

[123] Christian Walck. *Handbook on statistical distributions for experimentalists*. Vol. 10. University of Stockholm, 2007.

[124] Marie Laure Delignette-Muller y Christophe Dutang. fitdistrplus: An R Package for Fitting Distributions. En: *Journal of Statistical Software* 64.4 (2015), págs. 1-34. url: http://www.jstatsoft.org/v64/i04/.

[125] Hirotugu Akaike. A new look at the statistical model identification. En: *IEEE transactions on automatic control* 19.6 (1974), págs. 716-723.

[126] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2020. url: https://www.Rproject.org/.

[127] Carlos J. Gil Bellosta. *ADGofTest: Anderson-Darling GoF test*. R package version 0.3. 2011. url: https://CRAN.R-project.org/package=ADGofTest.

[128] Christophe Dutang, Vincent Goulet y Mathieu Pigeon. actuar: An R Package for Actuarial Science. En: *Journal of Statistical Software* 25.7 (2008), pág. 38. url: http://www.jstatsoft.org/v25/i07.

[129] Paul-Christian Bürkner. brms: An R Package for Bayesian Multilevel Models Using Stan. En: *Journal of Statistical Software* 80.1 (2017), págs. 1-28. doi: 10. 18637/jss.v080.i01.

[130] P. Ruckdeschel, M. Kohl, T. Stabla y F. Camphausen. S4 Classes for Distributions. English. En: *R News* 6.2 (mayo de 2006), págs. 2-6.

[131] Francois Aucoin. *FAdist: Distributions that are Sometimes Used in Hydrology*. R package version 2.2. 2015. url: https://CRAN.R-project.org/package= FAdist.

[132] Mikis Stasinopoulos y Robert Rigby. *gamlss.dist: Distributions for Generalized Additive Models for Location Scale and Shape*. R package version 5.1-1. 2018. url: https://CRAN.R-project.org/package=gamlss.dist.

[133] Thomas Roth. *qualityTools: Statistics in Quality Science.* R package version 1.55 http://www.r-qualitytools.org. 2016. url: http://www.r-qualitytools.org.

[134] Nahuel González. *Dataset of Timing distributions in free text keystroke dynamics profiles*. Ver. 1. IEEE DataPort. doi: 10.21227/ngv9-fa18. url: https://ieeedataport.org/documents/timing-distributions-free-text keystrokedynamics-profiles (visitado 07-03-2021).

[135] Nahuel González. *Dataset of Timing distributions in free text keystroke dynamics profiles*. Ver. 1. Mendeley Data. doi: 10.17632/sjk7kz35nh.1. url: https: //data.mendeley.com/datasets/sjk7kz35nh/1 (visitado 04-03-2021).

[136] Mark Andrew Hall. Correlation-based feature selection for machine learning. En: (1999).

[137] Ian H Witten, Eibe Frank, Mark A Hall, CJ Pal y MINING DATA. Practical machine learning tools and techniques. En: *DATA MINING*. Vol. 2. 2005, pág. 4.

[138] John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. En: (1998).

[139] Nahuel González. *Dataset for Towards Liveness Detection in Keystroke Dynamics: Revealing Synthetic Forgeries*. Ver. 1. IEEE DataPort. doi: 10.21227/1ka3-er49. url: https://ieee-dataport.org/documents/dataset-towards-livenessdetection-keystroke-dynamics-revealing-synthetic-forgeries (visitado 19-05-2021).

[140] Nahuel González. *Dataset for Towards Liveness Detection in Keystroke Dynamics: Revealing Synthetic Forgeries*. Ver. 1. Mendeley Data. doi: 10.17632/xvg5j5z29p. 1. url: https://data.mendeley.com/datasets/xvg5j5z29p/1 (visitado 19-05-2021).

[141] Nahuel González. *Dataset for The Reverse Problem of Keystroke Dynamics: Guessing Typed Text with Keystroke Timings*. Ver. 1. IEEE DataPort. doi: 10.21227/ 7616-7964. url: https://ieee-dataport.org/documents/dataset-reverseproblem-keystroke-dynamics-guessing-typed-text-keystroke-timings (visitado 22-04-2021).

[142] Nahuel González. *Dataset for The Reverse Problem of Keystroke Dynamics: Guessing Typed Text with Keystroke Timings*. Ver. 1. Mendeley Data. doi: 10.17632/94dwkbxf2d.1. url: https://data.mendeley.com/datasets/94dwkbxf2d/1 (visitado 22-04-2021).

[143] Sean Peisert y Matt Bishop. How to design computer security experiments. En: *IFIP World Conference on Information Security Education*. Springer. 2007, págs. 141-148.

[144] Luca Allodi y Fabio Massacci. Comparing vulnerability severity and exploits using case-control studies. En: *ACM Transactions on Information and System Security (TISSEC)* 17.1 (2014), págs. 1-20.

[145] CJ Mann. Observational research methods. Research design II: cohort, cross sectional, and case-control studies. En: *Emergency medicine journal* 20.1 (2003), págs. 54-60.

[146] Theodore W Anderson y Donald A Darling. Asymptotic theory of certain"goodness of fitc¸riteria based on stochastic processes. En: *The annals of mathematical statistics* (1952), págs. 193-212.

[147] Clifford M Hurvich y Chih-Ling Tsai. Regression and time series model selection in small samples. En: *Biometrika* (1989), págs. 297-307.

[148] Thomas J Santner, Brian J Williams, William I Notz y Brain J Williams. *The design and analysis of computer experiments*. Vol. 1. Springer, 2003.

[149] Walter F Tichy. Should computer scientists experiment more? En: *Computer* 31.5 (1998), págs. 32-40.

[150] Frédéric Desprez, Geoffrey Fox, Emmanuel Jeannot, Kate Keahey, Michael Kozuch, David Margery, Pierre Neyron, Lucas Nussbaum, Christian Perez, Olivier Richard et al. Supporting experimental computer science. Tesis doct. INRIA, 2012.

[151] Kevin O'Connor y Stephen Elliott. Biometric Zoo Menagerie. En: *Encyclopedia of Biometrics*. Ed. por Stan Z. Li y Anil K. Jain. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, págs. 1-4. isbn: 978-3-642-27733-7. doi: 10.1007/978-3-642-27733-7_9146-2. url: https://doi.org/10.1007/978-3-642-277337_9146-2.

[152] Nahuel González, Enrique P. Calot, Jorge S. Ierache y Waldo Hasperué. Towards liveness detection in keystroke dynamics: Revealing synthetic forgeries. En: *Systems and Soft Computing* (2022). Elsevier. issn: 2772-9419. doi: https://doi.org/10.1016/j.sasc.2022.200037. url: https://www.sciencedirect.com/science/article/pii/S2772941922000047.

[153] Nahuel González, Enrique P. Calot, Jorge S. Ierache y Waldo Hasperué. The Reverse Problem of Keystroke Dynamics: Guessing Typed Text with Keystroke Timings Only. En: *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*. IEEE, 2021, págs. 1-6. isbn: 978-1-6654-4231-2. doi: 10.1109/ICECET52533.2021.9698782. url: https://ieeexplore.ieee. org/document/9698782.

[154] Nahuel González, Jorge S. Ierache, Enrique P. Calot y Waldo Hasperué. Un método de ensamble basado en subsecuencias a nivel de palabras para la autenticación de usuarios con cadencias de tecleo en textos libres. En: *XXVII Congreso Argentino de*

Ciencias de la Computación (CACIC). 2021, págs. 685-694. isbn: 978-987-633-574-4. url: http://sedici.unlp.edu.ar/handle/10915/ 130532.

[155] Nahuel González. *Dataset for Exploring internal correlations in timing features of keystroke dynamics at word boundaries and their usage for authentication and identification*. Ver. 1. Mendeley Data. doi: 10.17632/vx83444p8n.1. url: https://data.mendeley.com/datasets/vx83444p8n/1 (visitado 22-02-2021).

[156] Nahuel González. *Dataset for An Ensemble Method for Keystroke Dynamics Authentication in Free-Text Using Word Boundaries*. Ver. 1. 2021. doi: 10.17632/xvg5j5z29p. 1. url: https://data.mendeley.com/datasets/xvg5j5z29p/1 (visitado 26-07-2021).

[157] Nahuel González. *Dataset for An Ensemble Method for Keystroke Dynamics Authentication in Free-Text Using Word Boundaries*. Ver. 1. 2021. doi: 10.21227/jdzh-4m97. url: https://ieee-dataport.org/documents/dataset-ensemblemethod-keystrok-edynamics-authentication-free-text-using-wordboundaries (visitado 26-07-2021).

[158] Hermann Minkowski. *Geometrie der Zahlen, Leipzig and Berlin: RG Teubner, JFM 41.0239. 03, MR 0249269*. Inf. téc. retrieved 2016-02-28, 1910.

[159] Martin Gardner. Taxicab geometry. En: *The Last Recreations*. Springer, 1997, págs. 159-175.

[160] Thomas Little Heath *et al*. *The thirteen books of Euclid's Elements*. Courier Corporation, 1956.

[161] Godfrey N Lance y William T Williams. Computer programs for hierarchical polythetic classification ("similarity analyses"*)*. En: *The Computer Journal* 9.1 (1966), págs. 60-64.

[162] Godfrey N Lance y William T Williams. Mixed-Data Classificatory Programs I - Agglomerative Systems. En: *Australian Computer Journal* 1.1 (1967), págs. 15-20.

[163] Cyrus D Cantrell. *Modern mathematical methods for physicists and engineers*. Cambridge University Press, 2000.

[164] A. W. Harzing. *Publish or Perish 7*. 2007. url: https://harzing.com/resources/ publish-or-perish.

[165] Fabian Monrose y Aviel D Rubin. Keystroke dynamics as a biometric for authentication. En: *Future Generation computer systems* 16.4 (2000), págs. 351-359.

[166] Creative Commons Corporation. *Creative Commons license CC BY-SA 4.0*. url: https://creativecommons.org/licenses/by/4.0/legalcode.

Esta red se constituyó formalmente en noviembre de 1996 y actualmente 51 universidades argentinas son miembros activos. Sus objetivos son:
"Coordinar actividades académicas relacionadas con el perfeccionamiento docente, la actualización curricular y la utilización de recursos compartidos en el apoyo al desarrollo de las carreras de Ciencia de la Computación y/o Informática en Argentina".
"Establecer un marco de colaboración para el desarrollo de las actividades de posgrado en Ciencia de la Computación y/o Informática de modo de optimizar la asignación y el aprovechamiento de recursos".

Las cadencias de tecleo configuran un atributo biométrico comportamental que puede ser utilizado como segundo factor de autenticación para la verificación transparente de la identidad del usuario. Todos los sistemas informáticos, y sobre todo aquellos relacionados con la seguridad, se encuentran bajo ataque permanente y deben ser diseñados con foco en esta consideración. En particular, un sistema de autenticación basado en cadencias de tecleo puede ser sometido a ataques de presentación con muestras sintetizadas; también, los atributos temporales de la escritura filtrados en el transcurso de un ataque por canal lateral pueden ser utilizados para identificar el texto ingresado o potenciar un ulterior ataque por fuerza bruta. Sin embargo, la mayoría de las veces los métodos propuestos en la literatura han sido evaluados bajo un modelo de esfuerzo cero, también llamado de impostores no entrenados, que subestima o ignora el riesgo de los anteriores. En este escrito se propone un sistema de detección de vida que emplea una familia de estrategias de síntesis de muestras artificiales, utilizadas como adversarios en un sistema de clasificación aumentado con distancias basadas en los histogramas empíricos del perfil intrausuario, para mitigar los riesgos de un ataque de presentación. Adicionalmente, una modificación del esquema propuesto para la defensa permite abordar el problema de identificación del texto ingresado utilizando solo atributos temporales, en textos más largos y con listas de candidatos más numerosas que en el estado del arte.

edulp
EDITORIAL DE LA UNLP

ISBN 978-631-6568-19-9