ISBN 978–987–633–574–4

Un método de ensamble basado en subsecuencias a nivel de palabras para la autenticación de usuarios con cadencias de tecleo en textos libres

Nahuel González[1*], Jorge S. Ierache[1],
Enrique P. Calot[1], and Waldo Hasperué[2]

[1] Laboratorio de Sistemas de Información Avanzados,
Facultad de Ingeniería, Universidad de Buenos Aires,
Ciudad AutBuenos Aires, Argentina
{ngonzalez,jierache,ecalot}@lsia.fi.uba.ar

[2] Instituto de Investigación en Informática (III-LIDI)
Facultad de Informática Universidad Nacional de La Plata
Investigador Asociado – Comisión de Investigaciones Científicas (CIC)
whasperue@lidi.info.unlp.edu.ar

Resumen Utilizando sólo los tiempos entre eventos de presión y liberación de teclas, es posible construir un segundo factor de autenticación basado en la cadencia de tecleo tanto para endurecer claves de usuario como para verificar la identidad en forma continua dentro de una muestra de texto libre. Dentro de este último caso se propone un método de ensamble para la autenticación de usuarios que utiliza subsecuencias a nivel de palabras. En contraste con otros métodos del estado del arte, que alcanzan tasas de error cercanas al 5 % con entrenamientos muy reducidos del orden de 250 caracteres, nuestro método demanda grandes cantidades de información para el entrenamiento inicial, en el orden de 50.000 caracteres, pero alcanza un EER más bajo, cercano al 3,6 %, al ser evaluado con un conjunto de datos públicamente accesible y capturado en condiciones realistas.

Keywords: seguridad informática, biometría comportamental, cadencias de tecleo, texto libre, aprendizaje automático, métodos de ensamble

1. Introducción

Las sutiles variaciones en la forma en que distintas personas teclean son suficientes para revelar su identidad. Hace cuarenta años, Gaines et al. [1], pioneros del análisis de cadencias de tecleo, reconocieron la utilidad de este fenómeno para la autenticación de usuarios. Utilizando sólo los tiempos entre eventos de presión y liberación de teclas, es posible construir un segundo factor de autenticación para endurecer claves de usuario [2] y para verificar identidad en forma

* Autor para correspondencia: Nahuel González (ngonzalez@lsia.fi.uba.ar)

ISBN 978-987-633-574-4

continua con texto libre [3]. Más recientemente, el análisis de cadencias de tecleo también ha encontrado usos fuera del dominio de la seguridad informática; por ejemplo, descubrir ciertas características fisiológicas o impedimentos clínicos del usuario [4], e incluso determinar en forma aproximada las variaciones de su estado emocional mientras escribe, basándose en autoreporte [5] o aplicando una interfaz cerebro-máquina para etiquetar las muestras [6].

El análisis de cadencias de tecleo ha dejado atrás una infancia difícil. Hace más de diez años eran muchos los problemas metodológicos que aquejaban a la disciplina, como la ausencia de conjuntos de datos públicamente accesibles de tamaño suficiente y el uso inconsistente de métricas de error incompatibles al expresar los resultados [7]. La posibilidad de plantear experimentos comparativos reproducibles y generalizables a las condiciones del mundo real se encontraba muy limitada y los estudios más rigurosos se restringían a métodos sencillos para verificar claves estáticas, reportando tasas de error del orden del 10 % [8]. Verificar textos libres demandaba muchas muestras muy extensas, con más de 800 caracteres, para acercarse a una precisión aceptable [9].

El estado de situación actual es, inversamente, satisfactorio y alentador. Hoy contamos con muchos conjuntos de datos enormes y públicamente accesibles; por ejemplo, [10] abarca casi 200.000 usuarios y contiene más de 136 millones de caracteres de texto libre capturado en condiciones realistas. A la par con muchas otras disciplinas relacionadas, el análisis de cadencias de tecleo ha integrado los métodos generales de aprendizaje automatizado en detrimento de técnicas *ad hoc*. Al éxito de este abordaje lo ilustra un estudio reciente, que empleando una sofisticada red neuronal recurrente del tipo siamesa alcanza un EER del 5 % al autenticar texto libre, aún restringiendo el entrenamiento a sólo 250 caracteres e incluso luego de escalar el sistema a más de 100.000 usuarios [11].

La cuestión que aquí nos compete es el problema dual. Mientras los autores del anterior exploran cuánto puede reducirse el tamaño del conjunto de entrenamiento sin comprometer las tasas de error y la escalabilidad del método, nosotros nos preguntamos cuánto puede reducirse la tasa de error si permitimos crecer al conjunto de entrenamiento. Proponemos un método de ensamble que utiliza subsecuencias a nivel de palabras, derivado de un estudio exploratorio previo de los mismos autores [12] sobre las correlaciones internas de los tiempos entre eventos de tecleo dentro de fronteras semánticas. Esperamos motivar procedimientos híbridos, que al combinar métodos de autenticación de convergencia rápida con aquellos asintóticamente óptimos logren, a la vez, tasas de error aceptables con mínimo entrenamiento y tasas de error óptimas en el largo plazo, cuando la plantilla biométrica del usuario cuente con suficientes muestras.

Contribuciones. El objetivo de este estudio es proponer un método de ensamble para autenticación con cadencias de tecleo en textos libres y evaluar su rendimiento cuando se cuenta con grandes cantidades de información, en la forma de muestras de escritura, para cada usuario. Las principales contribuciones ofrecidas son:

- Proponemos un método de ensamble para la autenticación de usuarios basada en cadencias de tecleo en textos libres, que fragmenta las muestras a

ISBN 978-987-633-574-4

verificar en subsecuencias a nivel de palabras y utiliza clasificadores individuales para cada una de ellas.
- Evaluamos el método propuesto sobre un conjunto de datos públicamente accesible, de gran extensión, capturado en condiciones realistas, y que ha sido utilizado en estudios anteriores [13].
- Ofrecemos en forma abierta los conjuntos de datos de entrenamiento y de resultados [14, 15] para permitir la verificación independiente y para facilitar ulteriores exploraciones y mejoras de este tipo de métodos.

Organización. El resto del artículo está organizado como se describe a continuación. La sección 2 reseña brevemente algunos estudios previos sobre el tema. La sección 3 describe el método propuesto. La sección 4 detalla la metodología del experimento, incluyendo el conjuntos de datos utilizado, el preprocesamiento y la limpieza de los datos, el proceso de clasificación, y la disponibilidad de los conjuntos de datos y resultados. La sección 5 discute los resultados. Finalmente, la sección 6 resume las conclusiones.

2. Estudios previos

Si bien existen antecedentes como el de Monrose y Rubin [2] para la autenticación de usuarios con cadencias de tecleo en textos libres, se trata más bien de estudios exploratorios. Sólo a partir de los trabajos de Bergadano, Gunetti, y Picardi [16] con la métrica R se alcanzan tasas de error equiparables a la verificación con claves estáticas o textos fijos. Sin embargo, esta métrica requiere muestras muy grandes, de más de 800 caracteres [9], para alcanzar resultados óptimos. También se ha observado que, al replicar el experimento utilizando un conjunto de datos capturado en condiciones realistas (en contraste con condiciones de laboratorio), las tasas de error se elevan sobremanera, hasta cuatro o cinco veces los valores reportados originalmente [17]. El método de modelado con contextos finitos [18] logra sortear esta última dificultad, alcanzando tasas de error óptimas en torno a los 250 caracteres y sin que estas se degraden notoriamente con la dificultad del conjunto de datos de evaluación.

El empleo de técnicas de aprendizaje automático para la autenticación de usuario en base a su cadencia de tecleo tiene una larga historia. Entre otros, Yu y Chao [19] han utilizado atributos derivados y un clasificador SVM para la tarea, Obaidat [20] ha explorado diversos tipos de redes neuronales, y Killourhy y Maxion [21] han aplicado bosques aleatorios pero en el caso de PINs.

El exponente más actual de aprendizaje automático aplicado a la autenticación de usuarios utilizando cadencias de tecleo es el de Ancien et al. [11]. Este estudio destaca no sólo 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. Lo más sorprendente es la escasa cantidad de información por usuario utilizada para entrenar la red neuronal; hay sólo 15 muestras por usuario en el conjunto de datos

ISBN 978-987-633-574-4

de evaluación elegido, que entre ellas suman no mucho más de 250 caracteres. El conjunto de datos [10] cuenta con unos 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 muestra 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.

Los métodos de ensamble han sido utilizados extensivamente en tareas de aprendizaje automático y sus aplicaciones. Probablemente los bosques aleatorios, que no necesitan presentación ulterior, sean la implementación más reconocida. Hasta donde alcanza nuestro conocimiento de la literatura del tema, no se han ensayado métodos de ensamble para la autenticación de usuarios con cadencias de tecleo en textos libres como el que aquí se propone, excepto como clasificadores enlatados luego de un proceso de extracción de atributos [21].

3. Método propuesto

Supongamos que contamos con una muestra $M = \{K, R, L\}$ de texto libre, de largo m, y queremos verificar que pertenezca al usuario legítimo, utilizando su perfil biométrico que cuenta con muestras pasadas. Es este un problema de clasificación binaria, ya que las únicas respuestas posibles son sí o no. La secuencia de teclas de M es $K = k_1 \dots k_m$, sus tiempos de retención (intervalos entre el evento de presión y liberación de cada tecla) son $R = r_1 \dots r_m$ y sus tiempos de latencia (intervalos entre eventos de presión de teclas sucesivas) son $L = l_1 \dots l_m$.

Sea E un conjunto de caracteres que incluye la tecla espacio, teclas de puntuación, caracteres especiales, etc. Particionamos M en las posiciones de todos los caracteres que pertenecen a E y descartamos las subsecuencias vacías, para obtener un ensamble de palabras P_i, con sus subsecuencias de teclas, tiempos de retención, y de latencia. Por ejemplo, si $K =$ *hola, mundo, soy ng123.* y $E = \{\ ,.\}$, tenemos que $P_1 =$ *hola*, $P_2 =$ *mundo*, $P_3 =$ *soy*, y $P_4 =$ *ng123.* En particular y salvo que se indique lo contrario, presupondremos que E se compone de todos los caracteres no alfanuméricos y, por lo tanto, que las palabras resultantes de la partición son secuencias alfanuméricas ininterrumpidas.

Ahora queremos autenticar, independientemente, cada palabra P_i y luego combinar los resultados para responder si M pertenece a un usuario legítimo o a un impostor. Nuestro objetivo es entrenar un clasificador para cada P_i de la muestra M de este usuario utilizando observaciones de la misma palabra en otras muestras de este y otros usuarios, que serán tratados como impostores.

Particionamos las muestras existentes en el perfil del usuario legítimo y recolectamos todas las observaciones pasadas disponibles de cada palabra P_i, generando para cada una de ellas una instancia de entrenamiento de la forma $r_1 \dots r_{m_i} l_2 \dots l_{m_i}$, en donde m_i es el largo de P_i. Estas instancias contienen $2m_i - 1$ atributos, uno para cada tiempo de retención y uno para cada laten-

ISBN 978-987-633-574-4

cia, exceptuando la de la primera tecla. El primer tiempo de latencia se excluye pues corresponde al intervalo entre la tecla especial anterior (que puede ser cualquiera) y la primer tecla de P_i; no es representativa de la palabra en sí, y no presenta la misma consistencia entre observaciones [12] que las demás. Todas estas instancias de entrenamiento se rotulan con la clase *legítimo*. Para generar los instancias de impostores, empleamos una colección de muestras de otros usuarios, que una vez más particionamos en la misma forma a nivel de palabra para extraer tantas muestras de cada P_i como instancias de entrenamiento del usuario legítimo tengamos. De esta forma, mantenemos balanceadas las clases simplificando la tarea del clasificador. A estas otras instancias de entrenamiento las rotulamos con la clase *impostor*.

Finalmente, con el conjunto de instancias de legítimo e impostor resultantes de ambos procesos, entrenamos el ensamble de clasificadores y luego registramos los veredictos para las correspondientes P_i. Cada veredicto individual de cada P_i otorga un voto a la decisión global del ensamble para la muestra M, que se define por mayoría de votos. La evaluación de estrategias de ponderación de los votos se plantea como una futura línea de investigación en la sección 5.1.

Si no contamos con suficientes observaciones de alguna palabra en el perfil del usuario (o entre las muestras de impostores) para generar un conjunto de entrenamiento, se elimina la P_i correspondiente del ensamble. Hemos utilizado un umbral de diez observaciones requeridas como mínimo.

4. Evaluación experimental

4.1. El conjunto de datos

Para este trabajo se ha utilizado el conjunto de datos LSIA de [17, 18], actualizado para incluir muestras adicionales capturadas desde entonces. El mismo está compuesto de muestras de texto libre, ingresadas en un teclado convencional. Se han registrado las teclas presionadas y los tiempos de retención (intervalo entre evento de presión y evento de liberación de tecla) y latencia (intervalo entre eventos de presión de teclas sucesivas) con precisión de milisegundos, junto con la identidad del usuario correspondiente. Debido a ciertas restricciones de la plataforma de captura, en ocasiones los tiempos fueron redondeados a múltiplos de 8 o 16 milisegundos.

Las muestras fueron capturadas en un entorno realista durante más de cuatro años, con usuarios de ambos sexos en un rango de edad entre 28 y 60 años, y aptitud para la escritura con grandes variaciones. El texto corresponde a lenguaje natural compuesto durante el transcurso de la labor cotidiana de los usuarios. Luego del preprocesamiento y limpieza de los datos descrita en la sección siguiente, quedaron disponibles 7897 muestras de 79 usuarios para la evaluación del método aquí propuesto.

Disponibilidad pública. Tanto el conjunto de datos de entrenamiento como el de resultados se encuentran a disponibilidad del público en forma abierta y gratuita, en los repositorios de Mendeley Data [14] y IEEE DataPort [15].

ISBN 978-987-633-574-4

4.2. Preprocesamiento y limpieza de los datos

Las muestras del conjunto de datos fueron preprocesadas por medio de una herramienta propia para experimentos de cadencias de tecleo, con el objetivo de convertir a un formato abierto el esquema binario propietario en el cual se encuentra almacenada la información original. Para cada muestra de cada usuario, se utilizó el proceso de partición descrito en la sección 3, descartándose todas aquellas P_i con algún tiempo de retención o latencia que faltara, tuviera valores negativos, o superara los 3000 milisegundos; este último criterio es para eliminar las pausas que no se corresponden con el ritmo normal de escritura.

Para cada usuario, cada muestra, y cada P_i que cumpliera los criterios anteriores, se generó un archivo CSV con $2m - 1$ columnas, en donde m es el largo de la palabra, y una fila por cada instancia de entrenamiento, tanto para los rótulos legítimo como impostor. Los tiempos de retención y latencia del P_i en consideración, o de otras repeticiones de la misma palabra en la muestra, no se incluyeron en el conjunto de entrenamiento para no contaminar este con la instancia a clasificar, lo que sesgaría el sistema hacia una menor tasa de error que la alcanzable en un caso real. Así, aunque una palabra aparezca en varias muestras de un cierto usuario, para cada una de ellas el conjunto de entrenamiento difiere, pues no se incluye ninguna subsecuencia de la muestra actual; incluirlas sería hacer trampa. Todos los usuarios restantes fueron considerados como potenciales impostores, y sus muestras disponibles para extraer instancias de entrenamiento y evaluación con rótulo impostor. Como de esta forma es esperable que haya muchas más observaciones de cada palabra entre las muestras de impostores, se realizó un muestreo aleatorio de las mismas hasta recolectar tantas observaciones como del usuario legítimo haya disponibles.

Finalmente, se generó un archivo CSV para cada usuario y cada muestra, conteniendo la lista de palabras que no fueron rechazadas y sus tiempos de retención y latencia, para ser evaluadas por los clasificadores individuales.

4.3. Clasificación

El siguiente proceso se realizó para cada usuario y cada muestra, obteniendo una lista de clasificaciones para cada P_i, junto con el valor de exactitud obtenida al evaluar el modelo correspondiente. La salida de la etapa de clasificación para cada usuario y cada muestra es un archivo CSV, cuyas filas enumeran la palabra evaluada, la exactitud del modelo, y el rótulo asignado (legítimo o impostor).

Para cada P_i que haya sobrevivido a los filtros, se utilizó la implementación de bosques aleatorios `RandomForestClassifier`, versión 0.24.2, de la librería `scikit-learn` [22] para entrenar un clasificador con el conjunto de datos de entrenamiento generado en la etapa de preprocesamiento. El motivo de esta elección puede hallarse en un estudio anterior de los autores [12], donde se realiza una comparación de rendimiento para la tarea de verificación de palabras individuales con distintos clasificadores, en donde los bosques aleatorios obtienen una precisión similar a las redes neuronales con una fracción del costo computacional. No se realizó escalado y normalización de los atributos en las instancias

ISBN 978-987-633-574-4

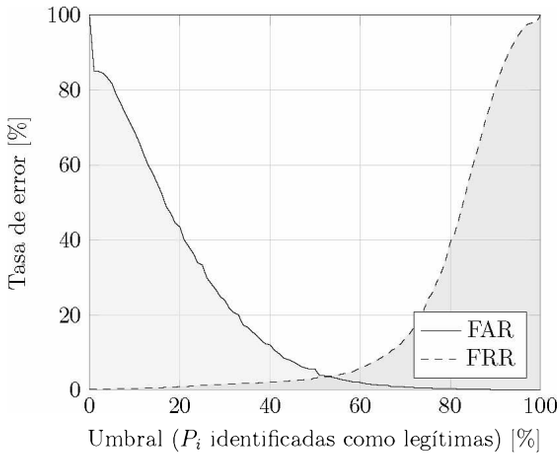

Figura 1: Distribución de falsos positivos y negativos para distintos umbrales

de entrenamiento, pues los bosques aleatorios no lo precisan [23], conservándose los valores originales en milisegundos.

La evaluación de exactitud de los modelos para palabras individuales fue realizada con el método `cross_val_score` de la librería `scikit-learn` [22], utilizando validación cruzada de cinco iteraciones. Las muestras de impostores utilizadas para evaluar la tasa de falsos positivos surgen, para cada usuario, de una selección aleatoria de las muestras de otros usuarios, del mismo tamaño que el conjunto de muestras del usuario legítimo.

5. Resultados y discusión

El proceso descrito en la sección anterior se llevó a cabo para cada muestra en el conjunto de datos de entrada, y se registró el porcentaje de P_i reconocidas como legítimas dentro de la muestra, tanto para los usuarios legítimos como para los impostores. En la figura 1 pueden observarse las tasas de falsos positivos (FAR) y falsos negativos (FRR) resultantes al fijar un umbral de aceptación, que es el porcentaje de votos positivos otorgados por las P_i de cada muestra requeridos para clasificarla como perteneciente al usuario legítimo.

Contrastemos estos resultados con aquellos de [11], citado más arriba, que puede considerarse el pináculo más reciente de la autenticación por medio de cadencias de tecleo en texto libre. Los autores reportan un EER de aproximadamente 5 % utilizando un entrenamiento de sólo 250 caracteres. Aquí alcanzamos un 3,6 % y entrenar cada clasificador por palabra requiere un mínimo de $10m$ caracteres, en donde m es el largo de la misma. Sin embargo, al tratarse de texto libre en donde muchas palabras son poco comunes, necesitamos una gran cantidad de muestras anteriores para conseguir suficientes observaciones de las P_i en consideración. Alcanzar esta tasa de error ha demandado suficientes muestras

ISBN 978–987–633–574–4

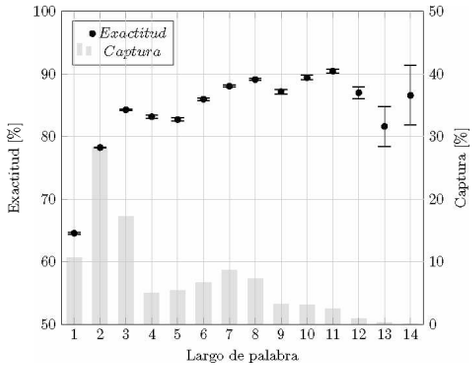

Figura 2: Captura y exactitud por largo de palabra para muestras legítimas

del usuario como para sumar aproximadamente 50.000 caracteres. La enorme mayoría del contenido de las muestras no se utiliza directamente en el entrenamiento de los clasificadores, pero esta ineficiencia es insalvable. No podemos elegir qué ha escrito el usuario y debemos utilizar el texto existente.

No es esperable que el método conserve su precisión al reducir este número. Lamentablemente, la implementación de [11] no se encuentra disponible en forma pública, y el dataset utilizado por los autores [10] no cuenta con suficientes caracteres por usuario para evaluar nuestro método. Para poner en perspectiva el tamaño del entrenamiento requerido, un usuario que hace un uso diario de la computadora intenso y prolongado teclea aproximadamente 15.000 caracteres por día, mientras que para el uso liviano el valor se reduce a aproximadamente un quinto [24]; convertido a días, los 50.000 caracteres requeridos para el entrenamiento oscilan entre cuatro y treinta. Una comparación de ambos se muestra en el cuadro 1. La contribución de palabras de distintos largos a la clasificación de

Característica	**Acien et al.**	**Método propuesto**
Clasificador	ANN siamesa recurrente	Ensamble de palabras + RF
Entrenamiento	250 caracteres	50.000 caracteres
EER	$\approx 5\,\%$	$\approx 3{,}6\,\%$

Cuadro 1: Comparación de principales características

muestras legítimas puede observarse en la figura 2. Se han incluido intervalos de confianza del 95 % para la exactitud promedio de los clasificadores individuales. Nótese que esta mejora con el largo de palabra hasta acercarse al 90 %, consistentemente con lo reportado en [12], pero que este efecto es difícil de aprovechar pues se utilizan con menor frecuencia. El intervalo de confianza crece con el largo de la palabra pues el tamaño de la población disminuye significativamente.

Es interesante notar una dificultad adicional. Al evaluar muestras de impostores, las palabras que pasan los filtros descritos en las secciones anteriores (sin

ISBN 978–987–633–574–4

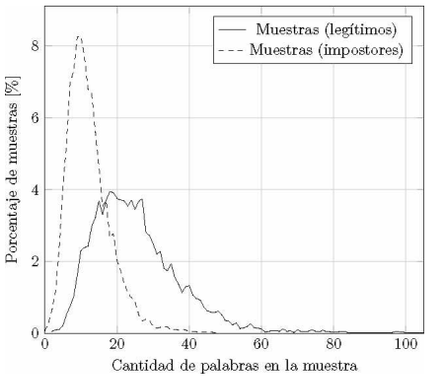

Figura 3: Porcentaje de muestras por cantidad de palabras utilizadas

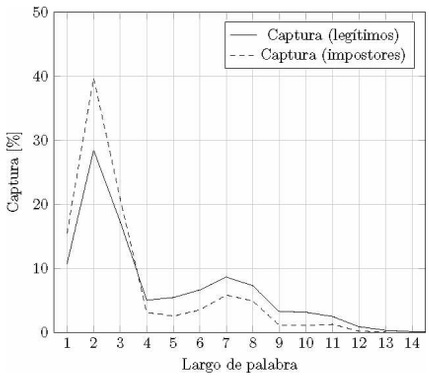

Figura 4: Captura por largo de palabra, para legítimos e impostores

atributos faltantes, ausencia de pausas, y suficientes muestras disponibles) tienden a ser menos y más cortas. Este efecto, que puede observarse en las figuras 3 y 4, es esperable ya que distintos usuarios tienden a utilizar distintas palabras con distintas frecuencias, y sólo aquellas que aparecen en común con suficiente frecuencia pueden ser utilizadas por este método.

5.1. Futuras líneas de investigación

La figura 2 muestra que con el incremento del largo de palabra la exactitud de los clasificadores individuales mejora, confirmando las conclusiones de [12]. En el método propuesto cada palabra brinda un voto al ensamble, pero el fenómeno antedicho apunta a la posibilidad de mejorar el rendimiento utilizando distintos pesos para distintas palabras, en base a su largo, la frecuencia de uso, y la exactitud de su modelo individual. La exploración de esta mejora se relega a futuras líneas de investigación.

6. Conclusión

En el presente estudio se propuso un método de ensamble para la autenticación de usuarios con cadencias de tecleo en textos libres, que utiliza subsecuencias a nivel de palabras. En contraste con otros métodos del estado del arte [11], que alcanzan tasas de error en el orden del 5 % con entrenamientos muy reducidos, nuestro método demanda grandes cantidades de información para el entrenamiento inicial pero alcanza un EER más bajo, del orden del 3,6 %. La combinación de los dos enfoques en un esquema mixto permitiría en principio lograr ambos objetivos, utilizando alguno de los primeros cuando se cuenta con poca información para alcanzar tasas aceptables rápidamente, y delegando al segundo cuando se hayan acumulado suficientes muestras. Los conjuntos de datos de entrenamiento y de resultados fueron puestos a disposición en forma pública y abierta en IEEE DataPort [15] y Mendeley Data [14].

Bibliografía

[1] R Stockton Gaines, William Lisowski, S James Press, and Norman Shapiro. Authentication by keystroke timing: Some preliminary results. Technical report, Rand Corp Santa Monica CA, 1980.

[2] Fabian Monrose and Aviel D Rubin. Keystroke dynamics as a biometric for authentication. *Future Generation computer systems*, 16(4):351–359, 2000.

[3] Patrick Bours and Hafez Barghouthi. Continuous authentication using biometric keystroke dynamics. In *The Norwegian Information Security Conference (NISK)*, volume 2009, 2009.

[4] Antony Milne, Katayoun Farrahi, and Mihalis A Nicolaou. Less is more: Univariate modelling to detect early parkinson's disease from keystroke dynamics. In *International Conference on Discovery Science*, pages 435–446. Springer, 2018.

[5] Clayton Epp, Michael Lippold, and Regan L Mandryk. Identifying emotional states using keystroke dynamics. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 715–724, 2011.

[6] Enrique P Calot, Jorge S Ierache, and Waldo Hasperué. Robustness of keystroke dynamics identification algorithms against brain-wave variations associated with emotional variations. In *Proceedings of SAI Intelligent Systems Conference*, pages 194–211. Springer, 2019.

[7] Kevin S Killourhy and Roy A Maxion. Should security researchers experiment more and draw more inferences? In *CSET*, 2011.

[8] Kevin S Killourhy and Roy A Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, pages 125–134. IEEE, 2009.

[9] Daniele Gunetti and Claudia Picardi. Keystroke analysis of free text. *ACM Transactions on Information and System Security (TISSEC)*, 8(3):312–347, 2005.

[10] Vivek Dhakal, Anna Feit, Per Ola Kristensson, and Antti Oulasvirta. Observations on Typing from 136 Million Keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, 2018. doi:https://doi.org/10.1145/3173574.3174220.

[11] Alejandro Acien, Aythami Morales, Ruben Vera-Rodriguez, Julian Fierrez, and John V Monaco. Typenet: Scaling up keystroke biometrics. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–7. IEEE, 2020.

[12] Nahuel González, Germán Concilio, Enrique P. Calot, Jorge S. Ierache, and Waldo Hasperué. Exploring internal correlations in timing features of keystroke dynamics at word boundaries and their usage for authentication and identification. In *Computer Science–CACIC 2020: 26th Argentine Congress, CACIC 2020, San Justo, Buenos Aires, Argentina, October 5–9, 2020, Revised Selected Papers*, volume 1, page 321. Springer Nature.

[13] Enrique P. Calot. Keystroke dynamics keypress latency dataset. Database, jan 2015. URL `http://lsia.fi.uba.ar/pub/papers/kd-dataset/`.

[14] Nahuel González. Dataset for an ensemble method for keystroke dynamics authentication in free-text using word boundaries, 2021. URL `https://data.mendeley.com/datasets/xvg5j5z29p/1`.

[15] Nahuel González. Dataset for an ensemble method for keystroke dynamics authentication in free-text using word boundaries, 2021. URL `https://ieee-dataport.org/documents/dataset-ensemble-method-keystroke-dynamics-authentication-free-text-using-word-boundaries`.

[16] Francesco Bergadano, Daniele Gunetti, and Claudia Picardi. User authentication through keystroke dynamics. *ACM Transactions on Information and System Security (TISSEC)*, 5(4): 367–397, 2002.

[17] Nahuel González, Enrique P Calot, and Jorge S Ierache. A replication of two free text keystroke dynamics experiments under harsher conditions. In *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6. IEEE, 2016.

[18] Nahuel González and Enrique P Calot. Finite context modeling of keystroke dynamics in free text. In *Biometrics Special Interest Group (BIOSIG), 2015 International Conference of the*, pages 1–5. IEEE, 2015.

[19] Enzhe Yu y Sungzoon Cho. Ga-svm wrapper approach for feature subset selection in keystroke dynamics identity verification. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 3, pages 2253–2257. IEEE, 2003.

[20] Balqies Obaidat, Mohammad S y Sadoun. Verification of computer users using keystroke dynamics. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 27(2): 261–269, 1997.

[21] Kevin S Maxion, Roy A y Killourhy. Keystroke biometrics with number-pad input. In *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, pages 201–210. IEEE, 2010.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] Yanjun Qi. Random forest for bioinformatics. In *Ensemble machine learning*, pages 307–323. Springer, 2012.

[24] Average keyboard use statistics per user. URL `https://whatpulse.org/stats/overall/numbers/#averages-per-user`.