Universidad
Rey Juan Carlos

Tesis Doctoral

Mejora de los sistemas de gestión de identidades federados mediante técnicas de análisis de comportamientos

Autor:

Alejandro García de Marina Martín

Directores:

Marta Beltrán Pardo

Isaac Martín de Diego

Programa de Doctorado en Tecnologías de la Información y las Comunicaciones

Escuela Internacional de Doctorado

Mayo de 2022

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN, ARQUITECTURA DE COMPUTADORES, LENGUAJES Y SISTEMAS INFORMÁTICOS Y ESTADÍSTICA E INVESTIGACIÓN OPERATIVA

Mejora de los sistemas de gestión de identidades federados mediante técnicas de análisis de comportamientos

Tesis Doctoral

Autor: Alejandro García de Marina Martín
Ingeniero del Software

Directores: Marta Beltrán Pardo
Doctora en Informática
Isaac Martín de Diego
Doctor en Matemáticas

Mayo de 2022

No aprendas nada, y el próximo mundo será igual que éste,
con las mismas limitaciones y pesos de plomo que superar.
Richard Bach

La ciencia se compone de errores,
que a su vez son pasos hacia la verdad
Julio Verne

Agradecimientos

En primer lugar, me gustaría agradecer a mis directores de tesis, Marta e Isaac, sin vuestra confianza este trabajo no hubiese sido posible. Habéis sido fuentes de inspiración y me habéis guiado de forma increíble. Durante este tiempo he aprendido muchísimo a vuestro lado, por eso entre otras tantas cosas, os estoy muy agradecido.

Me gustaría agradecer a mi familia, en especial a mis padres Juan y Margarita y a mi hermano Ricardo. Gracias a vosotros he tenido la motivación necesaria para seguir adelante y poder terminar este proceso. Ha habido momentos de frustración y momentos de alegrías pero siempre me he sentido arropado y apoyado de forma incondicional por vosotros. Habéis conseguido que de otro paso en el camino.

Agradecer a Raquel, la motivación, ánimo y apoyo sincero que me has dado siempre, y en especial durante esta tesis, es incalculable. Te estoy muy agradecido por acompañarme a lo largo de este proceso. Me has sabido entender y me has ayudado a gestionar las emociones en los momentos difíciles. Por todo esto, y mucho más, te doy las gracias.

Agradecer también a todos mis compañeros del Data Science Laboratory (DSLAB). Me gustaría mencionar en este punto a Alberto por colaborar en las publicaciones científicas asociadas a esta tesis y a Rubén por su gran ayuda.

Por último, me gustaría agradecer a todas esas personas que han estado ahí, incluso en ocasiones dándome ánimos y fuerzas sin ser conscientes de ello. Familia, amigos y compañeros muchísimas gracias.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Contexto de la investigación

La necesidad de proteger los activos y recursos frente a accesos indebidos de usuario no autorizados siempre ha existido. Del mismo modo que en un entorno físico como un aeropuerto, se necesita controlar que personas están autorizadas a salir o entrar de un país, en un sistema de información existe esta misma necesidad de controlar el acceso a sus recursos.

En el ámbito digital, la disciplina que se encarga de gestionar las diferentes identidades de los usuarios y la información relacionada con las mismas, con el objetivo de controlar el acceso a los diferentes recursos, servicios o aplicaciones, se denomina gestión de identidades y accesos [1]. Esta disciplina trata de garantizar la seguridad de los usuarios y de los recursos, servicios o aplicaciones, ya sea de manera individual o aislada, o al mismo tiempo en multitud de ellos alojados en diversos dominios.

Los primeros sistemas de información estaban pensados para ser utilizados por un único usuario. De esta forma, el usuario encargado de utilizar el sistema debía autenticarse con el objetivo de validar sus credenciales y, por lo tanto, pasar a comprobar los permisos y privilegios disponibles para interactuar con dicho sistema. En otras palabras, dicho usuario debía autenticarse en el sistema con el objetivo de poder autorizar las acciones que quería realizar en dicho sistema de información. Con el paso del tiempo, se empezaron a implantar los sistemas

multiusuario, en los que diversos usuarios interactúan simultáneamente con el mismo sistema y pueden estar autorizados a realizar diferentes operaciones dependiendo de su nivel de privilegios. De esta forma, los recursos computacionales han de repartirse en tiempo y espacio entre todos los usuarios de forma totalmente transparente para ellos. Así, los sistemas de gestión de identidades tuvieron que evolucionar para dar respuesta a los requisitos de los nuevos casos de uso.

Desde la llegada de Internet la gestión de identidades y accesos se fue adaptando para considerar que un número elevado de usuarios pueden estar solicitando acceso a un mismo servicio, aplicación o recurso alojados en servidores externos, heterogéneos y completamente distribuidos, haciendo que su gestión sea más complicada. Esto cambió el paradigma, pues la gestión de identidades se estaba volviendo demasiado compleja debido al gran número de identidades digitales que cada usuario tenía que manejar de forma independiente, ya que para cada servicio o aplicación se tenía que generar una nueva identidad (con su cuenta y credenciales asociadas).

Los últimos mecanismos desarrollados e implantados para solventar la gestión de identidades y accesos se basan en el concepto de federación. Estos mecanismos se basan en estándares y protocolos que permiten que un mismo usuario pueda utilizar una misma identidad digital para realizar cualquier flujo de Identificación, Autenticación, Autorización y Auditoria (IAAA) a lo largo de multitud de servicios, recursos o aplicaciones, alojados en diversos dominios, utilizando para ello el mismo proveedor de identidades de confianza. Esto se consigue porque los proveedores de servicios y aplicaciones delegan los procesos de IAAA en el proveedor de identidades. Sin embargo, esto no quiere decir que los recursos, servicios o aplicaciones deleguen totalmente la seguridad de los usuarios en manos de los proveedores de identidades y, por lo tanto, también han de asumir su responsabilidad a la hora de proteger a los usuarios.

¿De qué se debe proteger a los usuarios en este contexto? Principalmente de sufrir una suplantación de identidad, es decir, de situaciones en las que un tercero malicioso pueda actuar en su nombre (suplantando su identidad) para actuar con un recurso, servicio o aplicación. Este tipo de suplantaciones suelen producirse cuando las credenciales del usuario se ven comprometidas (por ejemplo, por un ataque de *phishing* o por una brecha de datos) o cuando su sesión se secuestra (por ejemplo, por un ataque de tipo *TCP hijacking* o por uno de *Cross Site Scripting* o *Cross Site Request Foegery*).

Para detectar o incluso evitar este tipo de ataques, esta tesis doctoral pretende explorar la aplicación de técnicas de análisis de comportamientos. Los fundamentos del análisis de comportamientos (en inglés, *behavioral analysis*) fueron introducidos por primera vez en 1953 [2]. En aquel entonces esta rama estaba muy ligada al campo de la psicología y se basaba principalmente en el análisis del comportamiento humano con el objetivo de poder aportar un beneficio concreto. Sus fundamentos principales fueron determinados posteriormente en 1960 [3].

Hoy en día, el análisis de comportamiento se denomina analítica del comportamiento (en inglés, *behavioral analytics*). La analítica del comportamiento se define como la rama científica que se centra en modelar el comportamiento de una entidad (usuario, animales, sensores, etc) para poder analizar y entender las interacciones que realiza en un sistema de información, con el objetivo de obtener un beneficio de negocio [4]. De esta forma, se hace uso del aprendizaje máquina para modelar y predecir los comportamientos pasados, presentes y futuros de las entidades.

Las técnicas del análisis de comportamiento se pueden categorizar en función del tipo de entidad que se considera como objeto de estudio. De esta forma, existe el análisis de comportamiento animal [5], análisis de comportamiento de sensores [6] y el análisis de comportamiento humano [7]. El análisis de comportamiento animal se basa en utilizar sensores acoplados a animales para comprender, monitorizar y predecir determinadas actividades de interés, por ejemplo, en una granja animal se utilizan collares inteligentes para monitorizar la actividad del ganado y predecir comportamientos que puedan ser indicadores de que el ganado está enfermo [8]. El análisis de comportamiento de sensores se basa en modelar las medidas captadas por ciertos sensores con el objetivo de detectar anomalías [9] para diversos fines como, por ejemplo, realizar procesos de autenticación [10]. Hoy en día, estas ramas del análisis de comportamiento están evolucionando constantemente gracias a la aparición del *Internet of Things* (IoT) y al desarrollo y aumento del número de sensores de bajo coste que se pueden acoplar en multitud de dispositivos del entorno físico de forma sencilla.

Por otro lado, el análisis de comportamiento humano se basa en modelar el comportamiento de personas o usuarios, tanto a nivel individual [11], como en colectivos o grupos sociales [12]. Cuando el dominio de aplicación es un sistema de información específico, el área científica se denomina análisis de comportamiento de usuarios.

Y es en este ámbito en el que se centra la presente tesis doctoral, ya que las técnicas de análisis de comportamiento deberían proporcionar una gran ventaja frente a los modelos convencionales a la hora de detectar intrusiones, ataques o fraudes.

1.2. Hipótesis de partida

Las especificaciones de gestión de identidades y accesos federada están siendo adoptadas muy rápidamente y a gran escala para solventar los procesos de IAAA. Algunos ejemplos claros se pueden ver en productos como Google Connect [13], Facebook Connect [14] o en el sector bancario con el desarrollo de *Financial-grade API* (FAPI) [15]. Sin embargo, estas especificaciones y sus implementaciones presentan vulnerabilidades de seguridad que se pueden explotar causando daños irreversibles en los sistemas de información [16]. Las principales líneas de investigación y soluciones planteadas hasta el momento para mitigar todas estas amenazas suelen estar orientadas al enriquecimiento de las peticiones y/o *tokens* empleados en los estándares, a realizar una mejora en la gestión de las sesiones de usuario, a la mejora de las *Application Programming Interface*s (APIs) y *Software Development Kit*s (SDKs) ofrecidas, o al uso de criptografía a diferentes niveles [17].

Los modelos de análisis de comportamiento se presentan como una solución efectiva y viable para solventar problemas relacionados con los procesos de IAAA [18]. Estas soluciones, no sólo permiten automatizar ciertos procesos, sino que también se posicionan como una de las soluciones más eficaces a la hora de detectar brechas de seguridad. Aun siendo una buena solución, todavía existe margen de mejora en los modelos existentes, especialmente en la combinación de información de múltiples fuentes de información heterogéneas.

Teniendo en cuenta estas dos líneas, surge como principal motivación de esta investigación tratar de integrar los modelos de análisis de comportamientos dentro de los principales estándares de gestión de identidades federada. Esto ayudaría a aumentar los niveles de seguridad proporcionados en dichos estándares, además de evitar que los recursos, servicios o aplicaciones deleguen totalmente la seguridad en los proveedores de identidades. Por otro lado, se pueden mejorar los modelos de análisis de comportamientos existentes, con el objetivo de que sean más eficaces y puedan escalar de forma sencilla para considerar simultáneamente multitud

de fuentes de información. Por todo esto, la hipótesis de partida de la presente tesis doctoral es:

Es posible mejorar los niveles de seguridad proporcionados actualmente por los principales esquemas de gestión de identidades federada integrando métodos de análisis de comportamientos en los flujos de información. Además, es posible mejorar la eficacia de los métodos de análisis de comportamiento actuales generando modelos que puedan combinar información de múltiples fuentes de datos heterogéneas de forma efectiva.

1.3. Objetivos

Los objetivos que la presente tesis doctoral pretende conseguir surgen de su hipótesis de partida. Estos objetivos generales se resumen en:

- Diseñar, implementar y evaluar un flujo de trabajo que defina y marque las pautas a seguir para integrar métodos de análisis de comportamientos en los flujos de información de los esquemas de gestión de identidades federada.
- Diseñar, implementar y evaluar un método de análisis de comportamientos que permita combinar información de fuentes de datos heterogéneas, mejorando la eficacia de los modelos del estado del arte.

Para conseguir el primer objetivo general se han definido los siguientes objetivos específicos:

1. Definir y esquematizar las principales tareas que ha de seguir una aplicación o servicio para integrar una solución de análisis de comportamientos en los estándares federados.
2. Definir los procesos de obtención y generación de huellas digitales que mejor representen el comportamiento de los usuarios de un sistema de información.
3. Definir y categorizar los posibles modelos de aprendizaje máquina y las métricas de evaluación de interés en el ámbito del análisis de comportamiento.
4. Analizar los posibles métodos de acción para integrar la información recogida de los modelos de análisis de comportamientos en los flujos de información de los estándares federados.

5. Validar y evaluar el flujo de trabajo propuesto en un entorno real.

Del mismo modo, para el segundo objetivo general, se definen los siguientes objetivos específicos:

1. Generar un conjunto de datos nuevo para aliviar la escasez y falta de datos en el ámbito del análisis de comportamiento.
2. Definir e implementar una técnica para representar la información de comportamientos que permita combinarla a nivel de características.
3. Definir e implementar una técnica que permita mejorar la eficiencia de los modelos de análisis de comportamientos.
4. Proponer un modelo de riesgos que, considerando las técnicas anteriores, sea capaz de detectar anomalías de comportamiento de forma efectiva.
5. Validar y evaluar el método basado en estas técnicas y modelo en un entorno real.

1.4. Metodología

La metodología utilizada con el fin de cumplir los objetivos planteados y poder demostrar la hipótesis de partida se define como sigue:

- Análisis y comprensión los flujos de información definidos por los esquemas de gestión de identidades federados.
- Análisis de los trabajos previos en el ámbito del análisis de comportamiento con el objetivo de encontrar limitaciones en los mismos cuando se aplican al dominio de interés en esta tesis.
- Análisis de los tipos de huellas digitales que se utilizan en la literatura para seleccionar los atributos más representativos y singulares.
- Análisis de las peculiaridades de los datos de dinámicas de comportamientos para seleccionar las métricas de evaluación que mejor se adapten y caractericen la problemática definida.

- Análisis y comprensión de las técnicas de combinación de la información en el ámbito del aprendizaje máquina.
- Creación de un entorno de trabajo real. Este entorno ha de implementar un sistema de gestión de identidades federado así como, un servicio o aplicación que haga uso de él.
- Recolección de los datos que contienen dinámicas de comportamientos.
- Evaluación y validación del flujo de trabajo propuesto utilizando el entorno de trabajo. Repercusión y evaluación de la seguridad añadida que se proporciona, así como de las posibles latencias añadidas que supone su implantación.
- Evaluación y validación del método de combinación de la información utilizando los datos recolectados.
- Comparativa del método propuesto con otros trabajos y algoritmos del estado del arte.

1.5. Estructura del documento

El presente documento de tesis doctoral se compone de un total de seis capítulos. En este primer capítulo se introducen los primeros conceptos relacionados con las temáticas relacionadas con la tesis, es decir, la gestión de identidades y accesos y el análisis de comportamientos. Además, se ha definido la hipótesis de partida, los objetivos tanto generales como específicos y la metodología a seguir para poder demostrar la hipótesis de partida. De aquí en adelante, los cinco capítulos restantes se resumen en:

- En el **Capítulo 2** se exponen y analizan los trabajos del estado del arte relacionados con las dos líneas de investigación de interés. En primer lugar, se abordan los conceptos fundamentales del campo del control de accesos y se recorre la evolución de los sistemas de gestión de identidades hasta llegar a los esquemas federados. Posteriormente se profundiza en el análisis de comportamiento, partiendo de los principales dominios de aplicación hasta llegar al ámbito de la ciberseguridad. A continuación, se presentan los trabajos relacionados con el análisis de comportamientos para solventar problemas específicos de control de accesos, mayoritariamente problemas de autenticación utilizando fuentes de

datos como el teclado y el ratón. Finalmente, se detallan las limitaciones encontradas en la literatura.

- En el **Capítulo 3** se propone un flujo de trabajo que permite integrar los métodos de análisis de comportamiento en los principales estándares de gestión de identidades federados. En primer lugar, se exponen los casos de uso en los que el flujo de trabajo puede mejorar los niveles de seguridad proporcionados actualmente. Posteriormente, se detallan todas las tareas que componen dicho flujo de trabajo y que marcan las directrices, consideraciones y recomendaciones que se deben seguir para una correcta implantación.

- En el **Capítulo 4** se propone un método para combinar información de múltiples fuentes de datos heterogéneas en el ámbito del análisis de comportamiento. Este método permite detectar anomalías de comportamiento y, por lo tanto, detectar brechas de seguridad y mejorar los sistemas de control de accesos. De esta forma, se definen y detallan todos los algoritmos y técnicas de aprendizaje máquina empleadas con el objetivo de generar un modelo robusto, con una eficacia elevada y capaz de generalizar de forma correcta.

- En el **Capítulo 5** se exponen los diferentes experimentos llevados a cabo para poder evaluar y validar las propuestas definidas en los Capítulos 3 y 4. En primer lugar, se detalla la creación de un entorno de trabajo. Este entorno permite la creación de un conjunto de datos que contiene dinámicas de comportamiento. Posteriormente, se procede a evaluar por separado tanto el flujo de trabajo, como el método de combinación de la información.

- En el **Capítulo 6** se exponen las conclusiones extraídas en la realización del presente trabajo. De esta forma, se analizan las principales lecciones aprendidas, el cumplimiento de los objetivos planteados y la validación de la hipótesis de partida. Además, se proponen futuras líneas de investigación relacionadas.

Por último, en el **apéndice A** se encuentra el glosario de acrónimos utilizados en el presente documento de tesis doctoral.

Capítulo 2

Estado del Arte

A lo largo de este capítulo se abordan los conceptos fundamentales y los trabajos del estado del arte relacionados con la temática del presente trabajo de tesis doctoral. De esta forma, se exponen los conceptos clave abordados con el objetivo de sentar los cimientos de la compresión de los capítulos posteriores en el ámbito de la gestión de identidades y del análisis de comportamiento. También, se analizan los trabajos y la literatura más relevantes en dichos ámbitos con el objetivo de situar el presente trabajo en un contexto y por consiguiente, poder establecer las carencias actualmente existentes y así dar sentido al trabajo de investigación aquí realizado.

En primer lugar, se aborda la gestión de identidades. Para ello, se analiza la evolución de la gestión de identidades, desde los primeros sistemas desarrollados, pasando por los sistemas centralizados, hasta llegar a los sistemas federados. Además, se profundiza en los estándares de gestión de identidades federados más populares hoy en día, OpenID [19], OAuth [20] y *OpenId Connect* (OIDC) [21]. Posteriormente, se exponen los principales trabajos que analizan las amenazas de estos estándares y las soluciones propuestas. En segundo lugar, se analiza la rama del aprendizaje máquina denominada análisis de comportamiento. En esta sección se explican las bases del análisis de comportamiento, los dominios de aplicación, así como los trabajos más relevantes y relacionados con el presente trabajo de investigación. Por último se exponen las limitaciones encontradas en la literatura.

2.1. Gestión de identidades y accesos

La gestión de identidades y accesos es la rama de las ciencias de computación que se encarga de gestionar el ciclo de vida de una identidad en un sistema de información, desde que se registra en dicho sistema, durante la interacción con el mismo y hasta que finalmente es eliminada [22]. Está compuesto principalmente por dos ramas fundamentales: el control de accesos y la gestión de identidades. En las siguientes secciones se profundiza en dichas ramas.

2.1.1. Contexto y conceptos básicos

Toda persona posee una identidad física que utiliza para identificarse frente a los controles de accesos existentes en el mundo físico como, por ejemplo, una aduana en un aeropuerto, o para autorizar una transacción económica en una entidad bancaria, entre otros. Esta identidad física está compuesta por el conjunto de características que identifican a una persona concreta como, por ejemplo, el nombre y apellidos (características legales), la huella dactilar (características físicas), u otros atributos menos identificativos como el documento nacional de identidad o el conjunto de propiedades a su nombre, etc. En el mundo tecnológico y de los sistemas de información estas identidades físicas son reemplazadas por identidades digitales. Del mismo modo, las identidades digitales se pueden definir como el conjunto de características, preferencias y reputación que identifican a una entidad o usuario concreto dentro de un sistema de información [23]. El concepto de entidad en el ámbito digital no solo hace referencia a los usuarios, sino que también incluye a toda entidad que puede interactuar con dicho sistema como, por ejemplo, empresas, dispositivos o agentes software (servicios web, clientes web, etc), entre otros. De este modo, una entidad física puede corresponderse con una o muchas identidades digitales, pero una identidad digital solo puede corresponderse con una o ninguna identidad física (p. ej. un agente software no se corresponde con ninguna identidad física).

Los sistemas de información alojan un conjunto de activos con los cuales las entidades interactúan. Estos activos son denominados recursos, y al igual que en el mundo físico, han de estar protegidos con el objetivo de garantizar que no se realicen acciones no autorizadas sobre ellos, como accesos indebidos o modificaciones que puedan corromperlos. Estos sistemas que alojan los recursos o que proveen a una entidad o usuario de algún tipo de servicio, son

comúnmente denominados, por los estándares de gestión de identidades, como proveedores de servicio (en inglés, *Service Provider* [SP]). Además, hay que hacer mención al agente encargado de gestionar el ciclo de vida de las identidades propiamente dichos, el proveedor de identidades (en inglés, *Identity Provider* [IdP]).

Las identidades digitales son la base de cualquier sistema de información en el que interactúan múltiples entidades o usuarios, pues permiten que se cumplan los tres principios básicos de la seguridad sobre los recursos alojados. Estos principios son: la confidencialidad, la integridad y la disponibilidad [24]. La confidencialidad garantiza que los datos alojados en el sistema de información solo pueden ser accedidos por aquellas entidades autorizadas a ello. La integridad asegura que los recursos no han sido manipulados o corrompidos por un tercero. La disponibilidad hace referencia a que los recursos se encuentren accesibles cuando son requeridos por una entidad. Además, gracias a las identidades digitales también se puede hacer cumplir un cuarto principio de la seguridad, el del no repudio, el cual garantiza que una entidad concreta no puede negar haber realizado una acción concreta en caso de que verdaderamente se haya llevado a cabo.

El éxito de garantizar que los principios básicos de seguridad se cumplan recae sobre los procesos de IAAA. Estos procesos, analizados a continuación, se pueden ver esquematizados en la Figura 2.1.

La identificación es el proceso mediante el cual se establece un vinculo de relación entre una entidad y su identidad, dentro del sistema de información, en base a los atributos proporcionas por la misma al sistema. Esta muy relacionado con el proceso de verificación, el cual se encarga de comprobar que una entidad previamente identificada en el sistema se corresponde con otra identidad, por ejemplo, una identidad física. Supóngase que un usuario se da de alta en una aplicación de la agencia tributaria para poder realizar unas gestiones particulares. En este momento, el usuario al darse de alta se ha identificado en el sistema. Sin embargo, es posible que, para realizar ciertas gestiones restringidas, la agencia tributaria deba corroborar que ese usuario es quien dice ser y, por consiguiente, ha de verificar que la identidad digital con la que se ha dado de alta se corresponde con una identidad física que posee los privilegios para poder realizar la gestión restringida. De este modo, puede solicitar al usuario que se presente en las instalaciones físicas de hacienda y presente su DNI, quedando así verificada su identidad digital

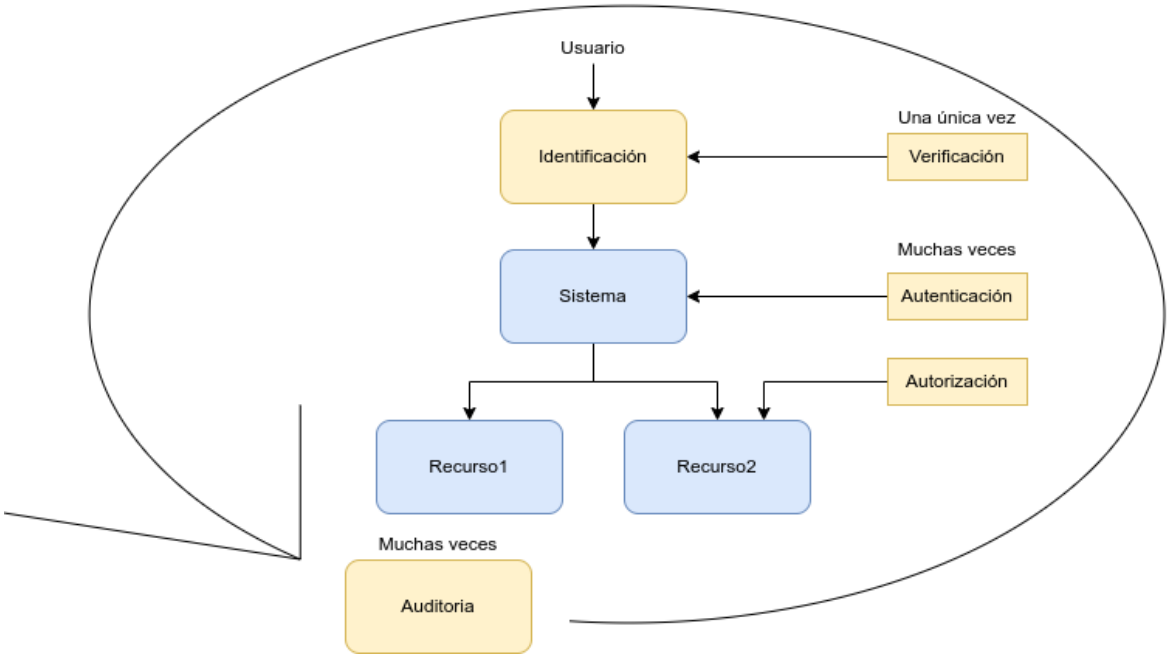


Figura 2.1: Esquema de los procesos de IAAA.

de manera segura.

La autenticación es el proceso por el cual un sistema de información es capaz de verificar, en un momento determinado, que una entidad es quien dice ser. Los mecanismos de autenticación se basan en proponer distintos retos que solo una entidad o usuario debería ser capaz de responder. Estos retos se pueden estructurar en tres categorías principales de acorde a su naturaleza [23]: algo que se sabe, algo que se tiene, o algo que eres. Para algo que se sabe normalmente se suelen utilizar contraseñas o un código personal (PIN), para algo que se tiene se suelen utilizar tarjetas inteligentes, *tokens* o un teléfono inteligente y para algo que eres se suelen utilizar rasgos biométricos como la huella dactilar. Estos retos se pueden combinar entre sí, aumentando así la seguridad del sistema que lo incorpora. A este proceso se le denomina Autenticación de Múltiples Factores (AMF) [25]. Cabe destacar que estos retos se suelen evaluar en un único instante, al comienzo de la interacción entre el usuario y el sistema. Con el objetivo de mejorar la seguridad, estos retos se pueden extender a lo largo del tiempo, es decir, a lo largo de toda una sesión de interacciones, pudiendo así verificar la identidad de una entidad tantas veces como sea necesario mientras una sesión esté activa. El proceso de extender la autenticación a lo largo de toda una sesión se denomina autenticación continua [26].

La autorización consiste en permitir o denegar el acceso a un recurso mediante la comprobación de los privilegios y permisos que posee la entidad interesada en un sistema de información. La identificación ocurre una única vez en un sistema, al comienzo de las interacciones, sin embargo, tanto la autenticación como la autorización son procesos que se van a repetir múltiples veces a lo largo del ciclo de vida de una identidad digital.

La auditoría es el proceso que se encarga de registrar todas las interacciones de las entidades con un sistema de información. Gracias a este proceso, todas las interacciones realizadas por las diversas entidades, quedan almacenadas y por lo tanto se pueden utilizar para analizar y verificar si los procesos de identificación, autenticación y autorización, anteriormente descritos, se han realizado de forma satisfactoria.

Finalmente, cabe destacar que, todo sistema de gestión de identidades y accesos ha de garantizar en la medida de lo posible, además de los tres principios básicos de seguridad, los siguientes requisitos de privacidad: revocación, anonimato, libre elección, verificación, antifraude y mínima información. La revocación hace referencia a que un usuario ha de poder solicitar que el sistema deje de almacenar la información asociada a su identidad digital. El anonimato hace referencia a que las identidades, tanto físicas como digitales, no han de poderse asociar con otras identidades para obtener información adicional. La libre elección hace referencia a que cada entidad ha de poder elegir entre múltiples proveedores de identidades. La verificación hace referencia a que un usuario ha de poder contrastar y consultar la información que un proveedor de identidades posee sobre su identidad. El antifraude se refiere a la imposibilidad de realizar determinadas acciones por un agente externo que ha suplantado una identidad del sistema. La mínima información hace referencia a que el sistema no debe almacenar más información sobre la entidad que la estrictamente necesaria.

La Tabla 2.1 recopila, a modo resumen, los conceptos fundamentales abordados en esta sección con el objetivo de mejorar la comprensión global de este documento.

2.1.2. Control de accesos

El control de accesos se encarga de garantizar o restringir el acceso de las entidades y usuarios (bajo una identidad digital) a un servicio o recurso del sistema. Esto garantiza que solo

Concepto	Definición
Entidad o usuario	Ente que interactúa con un sistema (usuario, empresa, agente software, etc)
Recurso	Activo de un sistema de información
Identidad digital	Conjunto de atributos, preferencias y reputación que identifican a una entidad o usuario concreto dentro de un sistema de información
Service Provider (SP)	Proveedor de servicio, recursos o aplicación.
Identity Provider (IdP)	Proveedor de identidades. Encargado de gestionar el ciclo de vida de una identidad digital.
Identificación	Proceso mediante el cual se corrobora la identidad digital de una entidad de forma exclusiva en un sistema de información con el objetivo de poder validarla en las siguientes interacciones.
Autenticación	Proceso de validación una identidad digital, garantizando así que una entidad es quien dice ser en un momento determinado.
Autenticación continua	Extensión a lo largo de toda una sesión del proceso de autenticación.
Autorización	Proceso de restringir el acceso de una entidad o usuario a un recurso.
Auditoría	Proceso de registro de todas las interacciones de las entidades o usuarios con un sistema de información.

Tabla 2.1: Resumen de los conceptos básicos relacionados con la gestión de identidades y accesos.

	Archivo 1	Archivo 2	Ejecutable 1	....	Objeto M
Usuario 1	Acción1...AcciónK	Acción1...AcciónK	Acción1...AcciónK		
Usuario 2	Acción1...AcciónK	Acción1...AcciónK	Acción1...AcciónK		
....					
Usuario N					

Figura 2.2: Almacenamiento en directorios. Modelo HRU.

los usuarios legítimos puedan acceder a los recursos o servicios determinados para ello bajo unas condiciones seguras y predefinidas, negándole dicho acceso a los usuarios no autorizados.

Los primeros sistemas informáticos estaban pensados para ser utilizados por un único operador o usuario. Este único usuario tenía plena disponibilidad sobre los recursos del sistema. Sin embargo, debido a las necesidades intrínsecas de estos sistemas, evolucionaron hasta convertirse en sistemas multiusuario.

En los sistemas multiusuario, los usuarios compiten por los recursos disponibles tanto en tiempo como en espacio. Debido a esta competencia, surgió la necesidad de elaborar sistemas de control de accesos que restringiesen el acceso de los diferentes usuarios a los recursos, servicios o aplicaciones alojados en el sistema. De esta forma, surgieron las políticas de acceso, las cuales recopilan un conjunto de reglas que determina los privilegios de las entidades y usuarios sobre los recursos del sistema. Por tanto, el control de accesos está muy relacionado con el concepto de autorización.

Existen dos formas muy extendidas, entre otras, de almacenar estas políticas de accesos [27]: directorios y las listas de control de accesos (en inglés, *Access Control Lists* [ACL]). El modelo de directorios, también conocido como modelo Harrison, Ruzzo y Ullman (HRU), viene definido como una matriz en la que cada fila se corresponde un un usuario del sistema y cada columna se corresponde con un recurso del sistema (ver Figura 2.2). Como se puede observar, cada celda posee los privilegios de lectura, escritura, ejecución y propiedad de un usuario para un recurso concreto. Por otro lado, las listas de control de accesos definen un conjunto de listas en el que cada elemento representa un usuario o un recurso (ver Figura 2.3). De este modo, las conexiones entre dos elementos denotan que el primer elemento posee los privilegios indicados sobre el segundo elemento.

Sea cual sea la forma de almacenar las políticas de acceso, existen multitud de formas de

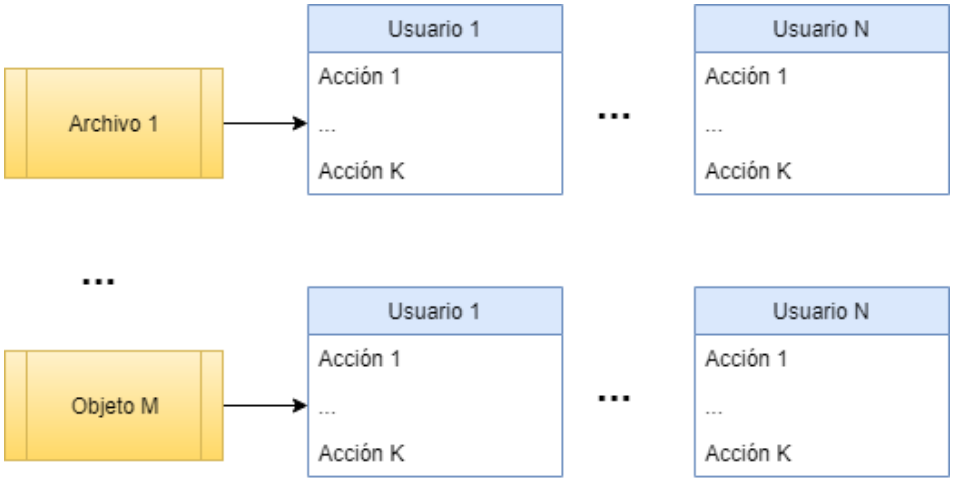


Figura 2.3: Almacenamiento en listas de control de accesos.

gestionar los accesos. Estos mecanismos de gestión son denominados modelos de control de acceso. Hoy en día, existen cuatro modelos fundamentales de control de accesos:

1. Modelos de control de accesos obligatorio (en inglés, *Mandatory Access Control* [MAC]): el administrador del sistema es el encargado de definir los privilegios y permisos de los recursos y de los propios usuarios con el objetivo de determinar si un acceso es legítimo. Un claro ejemplo de este sistema de control de accesos es la clasificación de archivos de la Agencia de Seguridad Nacional de Estados Unidos. En este modelo, cada recurso del sistema se categoriza de acorde a su sensibilidad (p. ej. público, privado, confidencial, secreto, etc) y a su categoría (p. ej. departamento, proyecto, rango de mando, etc). De esta forma, solo los usuarios que posean el nivel requerido de sensibilidad y categoría pueden acceder a los recursos de los dichos niveles. Este tipo de sistemas son centralizados, pues es un administrador el que define las políticas de acceso del sistema globalmente.

2. Modelos de control de accesos discrecionales (en inglés, *Discretionary Access Control* [DAC]): definido por *Trusted Computer System Evaluation Criteria*. En este tipo de sistemas, son los propios usuarios los que definen el conjunto de reglas (permisos y privilegios) sobre los recursos de los cuales son propietarios. De esta forma, un usuario puede decidir que recursos, siendo propietario del recurso, son accesibles y de qué forma por terceros. Un claro ejemplo de la implementación de este modelo son los sistemas GNU/Linux. Este tipo de sistemas son descentralizados, pues cada usuario toma decisio-

nes sobre sus recursos.

3. Modelos de control de acceso basados en roles (en inglés, *Role-Based Access Control* [RBAC]): se basa en asignar roles a cada usuario y recurso del sistema. Cada rol recopila los privilegios y permisos asignados sobre los recursos del sistema. De esta forma, la gestión de las identidades dentro de un sistema se convierte en una tarea más sencilla que en los modelos anteriores, pues se pueden manejar los grupos de permisos de forma conjunta. Por ejemplo, supóngase un rol Director que posee todos los permisos sobre los recursos del sistema, mientras que el rol Programador solo posee acceso a un número limitado de recursos. En caso de que se quiera registrar un nuevo usuario Programador en el sistema, bastará con asignarle el rol ya existente, en vez de tener que modificar los permisos de cada recurso del sistema para otorgarle los accesos correspondientes.

4. Modelos de control de accesos basados en atributos (en inglés, *Attribute-Based Access Control* [ABAC]): se basa en establecer la política de accesos en base a atributos de usuario, recurso y entorno. Estos atributos no son más que características que definen a cada uno de los agentes mencionados anteriormente. De esta forma, el control de accesos se realiza comprobando que cada uno de los atributos necesarios de una petición se cumplen en las reglas recopiladas en la política de accesos. Por ejemplo, supóngase un usuario que posee el atributo Director y que solicita acceso a un recurso con un atributo Confidencial y posee un atributo de entorno 11:00 pm. La política define que Director puede acceder a los recursos Confidencial, sin embargo, se tiene que cumplir una tercera regla para el atributo entorno que sea menor de 7:00pm. En este caso, se denegará el acceso, pues los atributos evaluados no se corresponden al completo con las reglas de la política de accesos. *eXtensible Access Control Markup Language* (XACML) es el estándar más extendido para implementar este tipo de modelo.

Cabe destacar que los modelos aquí expuestos, además de otros existentes, no son excluyentes entre sí y, por consiguiente, un mismo sistema puede implementar varios de ellos simultáneamente. Esta combinación de múltiples modelos resultará, con una buena implementación, en un sistema más seguro. Sin embargo, también aumenta la complejidad del modelo, creando reglas muy complejas que pueden afectar al rendimiento y a la mantenibilidad del mismo. Es por esto que un buen sistema de control de accesos ha de buscar un equilibrio entre seguridad

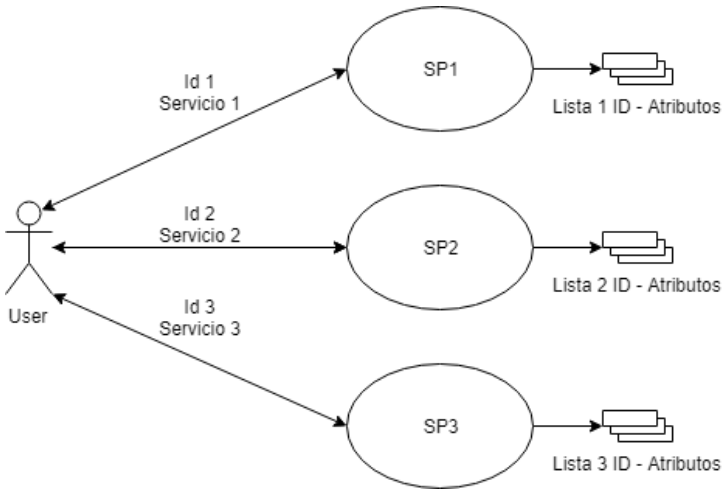


Figura 2.4: Modelo Silo para la gestión de identidades.

y el resto de los requisitos, buscando siempre obtener un modelo lo más seguro posible siendo lo más sencillo posible. Un ejemplo de esta combinación de modelos, es el módulo de seguridad SELinux [28], el cual permite extender el control de acceso DAC de los sistemas Linux, implementando políticas MAC y RBAC.

2.1.3. Modelos para la gestión de identidades

La gestión de identidades hace referencia a los sistemas y arquitecturas que se encargan de almacenar y administrar las identidades digitales dentro de un sistema de información.

El modelo Silo es la primera arquitectura que surgió para gestionar las identidades digitales [29]. En esta arquitectura, el sistema de gestión de identidades actúa tanto de SP como de IdP. En primer lugar, el sistema actúa de SP, ya que provee a los usuarios de un servicio, aplicación o recurso. En segundo lugar, actúa de IdP, ya que tiene como responsabilidad almacenar todas las identidades digitales del sistema, junto a sus credenciales y validar su autenticidad con el objetivo de poder gestionar y completar cualquier flujo de IAAA. De esta forma, en este tipo de sistemas, la figura del IdP y del SP se solapan en un único agente que se encarga de realizar ambas funcionalidades. Su funcionamiento se ve reflejado en la Figura 2.4.

Esta arquitectura para la gestión de identidades es la más antigua. Se implementó cuando el número de servicios o aplicaciones, a los que accedía un mismo usuario, era razonable y limitado y, por lo tanto, el propio usuario podía gestionar todas las identidades que manejaba a

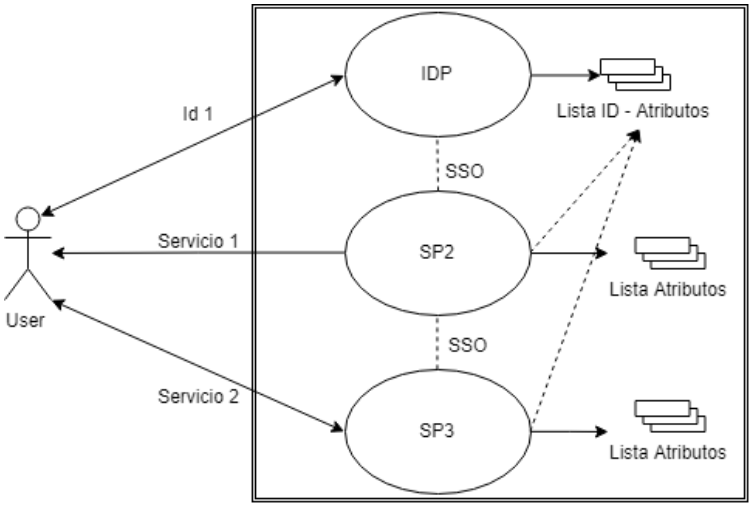


Figura 2.5: Modelo centralizado para la gestión de identidades.

lo largo de todos estos servicios o aplicaciones. A medida que el número de servicios o aplicaciones ofertados fue aumentando, la usabilidad real de estos sistemas fue decreciendo. Esto dio paso a los sistemas de gestión de identidades centralizados.

Los sistemas de gestión de identidades centralizados surgieron con el objetivo de suplir las carencias del modelo Silo. De esta forma, tratan de unificar y centralizar en un único punto la gestión de identidades a lo largo de múltiples servicios dentro de un mismo dominio [29]. En esta ocasión, las figuras del IdP y del SP se desvinculan formando agentes totalmente distintos. De este modo, el IdP se encarga de almacenar y gestionar toda la información referente a las identidades digitales. En esta arquitectura, el SP delega todo el flujo de IAAA sobre el IdP. El funcionamiento de este tipo de arquitecturas se puede observar en la Figura 2.5. Como se puede observar, cuando un usuario inicia cualquier flujo de IAAA, lo inicia sobre el IdP, y es este último quien verifica y le otorga acceso a todos los SPs bajo los que opera. Esto permite que un mismo usuario pueda realizar procesos de IAAA sobre múltiples servicios, en el mismo dominio, utilizando únicamente una identidad digital. Este tipo de arquitecturas solventa los problemas encontrados para el modelo Silo. Sin embargo, este tipo de sistemas centralizan toda la seguridad en un único punto y, por consiguiente, si un atacante logra sobrepasar dicha barrera, obtendrá acceso no solo a un SP, sino a todos los SP en los que el usuario atacado tiene acceso.

Los sistemas centralizados permiten implementar el proceso de *Single Sign On* (SSO). El SSO unifica todos los puntos de acceso a múltiples SP en un único punto. Esto se consigue

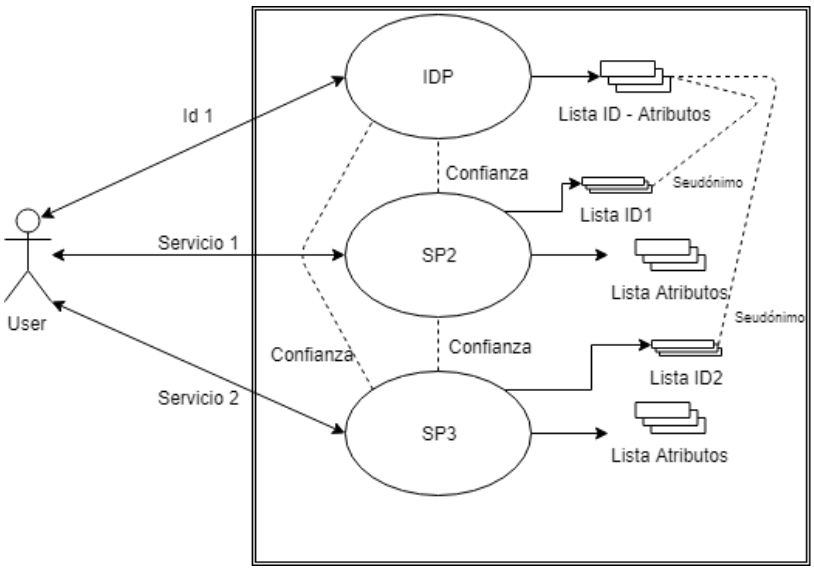


Figura 2.6: Modelo federado para la gestión de identidades.

ya que el IdPs implementa una base de datos centralizada que contiene toda la información referente a las identidades digitales, incluidas sus credenciales, que recogen todos los SPs.

Existen multitud de estándares, protocolos y sistemas de gestión de identidades centralizados. Los más extendidos y utilizados hoy en día son *Lightweight Directory Access Protocol* (LDAP) [30], Kerberos [31] y *Remote authentication dial in user service* (RADIUS) [32].

La federación de identidades es el conjunto de arquitecturas y estándares que permiten distribuir de forma dinámica las identidades y su información a lo largo de múltiples dominios seguros [33]. Estas arquitecturas extienden a los sistemas centralizados, permitiendo que un usuario pueda realizar procesos de IAAA sobre múltiples SP alojados en diferentes dominios utilizando un IdP externo y común, en el que se establece una relación de confianza. De esta forma, un SP crea un vínculo de confianza con un IdP externo y delega los procesos de IAAA sobre él. Estos vínculos de confianza forman círculos de confianza (en inglés, *Circle of Trust* [COT]). Una vez creados los COT, el IdP crea seudónimos para identificar y vincular a la identidades digitales que almacena y sus SPs correspondientes. En la Figura 2.6 se puede observar la arquitectura de este tipo de modelos.

Este tipo de modelos vienen siendo utilizados desde el año 2003, con la creación de *Security Assertion Markup Language* (SAML) [34]. Hoy en día, se han extendido y adoptado otros estándares como OpenID [19], OAuth [20] y OIDC [21]. De aquí en adelante, se analizan los

estándares federados más utilizados hoy en día, haciendo especial hincapié en sus conceptos fundamentales y sus flujos de información.

2.1.4. Estándares federados para la gestión de identidades

OpenID

OpenID se creó en el año 2005 y se define como un *framework* abierto, descentralizado y gratis para la gestión de identidades [19]. Su desarrollo ha sido apoyado por grandes corporaciones como Google, Microsoft y IBM. Está orientado a solventar la autenticación en escenarios web.

OpenID define tres roles fundamentales que interactúan entre sí para lograr realizar el proceso de autenticación:

- *Relying Party (RP)*: es la parte en la que los otros roles confían. Su función principal es hacer de cliente, es decir, es la aplicación o servicio con la que el usuario final interactúa y por lo tanto con la que necesita autenticarse. El rol de la RP es el de SP.

- *OpenID Provider (OP)*: es el encargado de gestionar el ciclo de vida de una identidad digital. El rol del OP es el de IdP.

- *End User (EU)*: es el usuario final, es decir, es la entidad que posee una identidad digital dentro del OP e interactúa con la RP para realizar la operativa deseada.

El flujo de autenticación que utiliza OpenID se resume a continuación:

(A) El EU accede a la RP e inicia el proceso de autenticación presentando el identificador, previamente registrado, a la RP por medio del agente de usuario (p. ej. un navegador web).

(B) La RP recibe y normaliza el identificador recibido. De esta forma, extrae e identifica el OP que necesita el EU para lograr autenticarse.

(C) La RP y el OP establecen un código secreto compartido que es almacenado por la RP. Este código secreto se utiliza para verificar que el intercambio de mensajes entre ambas partes es correcto.

(D) La RP redirige al agente de usuario del EU al OP, previamente identificado, realizando una petición de autenticación.

(E) El OP valida las credenciales proporcionadas por el EU.

(F) El OP redirige el agente de usuario del EU de vuelta a la RP comunicándole si el proceso de autenticación ha quedado completado o si por el contrario ha fallado.

(G) RP La RP verifica que la información recibida es correcta y por tanto el proceso de autenticación ha quedado finalmente completado.

OAuth

OAuth es un *framework* de código libre para la gestión de identidades de forma federada. Su desarrollo empezó en el año 2006, logrando su primera versión estable OAuth 1.0 en el año 2010, publicado como RFC 5849 [35]. Actualmente se encuentra en su versión OAuth 2.0 publicado como RFC 6749 [20].

OAuth 2.0 provee todos los mecanismos necesarios para poder realizar autorización en aplicaciones web, aplicaciones de escritorio y dispositivos inteligentes de forma federada. Cuando un usuario realiza una petición de acceso a un recurso protegido alojado en un SP, este le redirige al IdP externo, en el que confían tanto el usuario, como el SP. En este instante, el IdP verifica la identidad del usuario y procede a evaluar la petición, categorizándola como legítima en caso de que el usuario introduzca unas credenciales válidas y por consiguiente otorgándole acceso al recurso solicitado, o como no legítima, en caso contrario y, por consiguiente, negándole dicho acceso. Este proceso se utiliza exclusivamente para garantizar un proceso de autorización y, por lo tanto, OAuth 2.0 asume que el proceso de autenticación se realiza por otra vía, ya sea de forma local, con un modelo centralizado o de forma federada con otro estándar que lo permita. De aquí en adelante se va a detallar el funcionamiento de OAuth 2.0.

OAuth 2.0 define cuatro roles fundamentales que interactúan en cualquier flujo de autorización:

- *Resource Owner*: entidad propietaria de los recursos protegidos. Esta entidad puede ser el propio EU.
- *Resource Server*: es el servidor que almacena los recursos protegidos. Su funcionalidad es recibir y responder correctamente a las peticiones de acceso a los recursos protegidos.
- *Client*: es el cliente, es decir, es una aplicación que solicita acceso a los recursos protegidos. Por ejemplo, un navegador web que utiliza un usuario para acceder a un recurso. Dependiendo de su capacidad para mantener la confidencialidad y las credenciales del usuario pueden ser confidenciales o públicos.
- *Authorization Server*: servidor encargado de verificar los privilegios y roles de un *Client* con el objetivo de garantizar que solo los usuarios legítimos accedan a los recursos protegidos.

El modo en el que estos roles interactúan y se comunican entre sí, para lograr el proceso de autorización es por medio del uso de *tokens*. En OAuth 2.0 se distinguen fundamentalmente dos tipos de *tokens*:

- *Access token*: es el *token* de acceso. Lo utiliza el *Client* para acceder a los recursos protegidos alojados en el *Resource Server*. En otras palabras, este *token* sustituye a las credenciales del usuario en un proceso convencional de autorización. Este *token* es generado por el *Authorization Server*.
- *Refresh token*: es el *token* de refresco. Lo utiliza el *Client* para solicitar un nuevo *access token* cuando el vigente ha sido invalidado.

El flujo de autorización de OAuth 2.0, que se lleva a cabo entre los diferentes roles para lograr el proceso de autorización, se puede ver ilustrado en la Figura 2.7.

(A) El *Client* solicita autorización al propietario del recurso protegido (*Resource Owner*).

Figura 2.7: Flujo de autorización de OAuth.
Fuente: https://tools.ietf.org/html/rfc6749

(B) El *Client* recibe la concesión de autorización (en inglés, *Authorization Grant*) que le otorga permisos para poder acceder al recurso protegido.

(C) El *Client* utiliza esta concesión para solicitar al *Authorization Server* el *access token*.

(D) El *Authorization Server* comprueba que la concesión es válida, y en caso afirmativo procede a devolverle el *access token*.

(E) El *Client* utiliza el *access token* para solicitar acceder al recurso protegido alojado en el *Resource Server*.

(F) El *Resource Server* valida el *access token* y devuelve al *Client* el recurso solicitado.

Las concesiones de autorización (paso B y C de la Figura 2.7) definen las tareas e interacciones que se han de realizar, de forma secuencial, entre los diferentes roles, con el objetivo de garantizar que se logre un proceso de autorización. De este modo, dependiendo de las diferentes casuísticas y escenarios posibles, los diferentes roles han de seguir una de las posibles concesiones de autorización disponibles. Dentro del estándar de OAuth, se definen, cuatro tipos de concesiones: *Authorization Code*, *Implicit*, *Resource Owner Password Credentials* y *Client Credentials*.

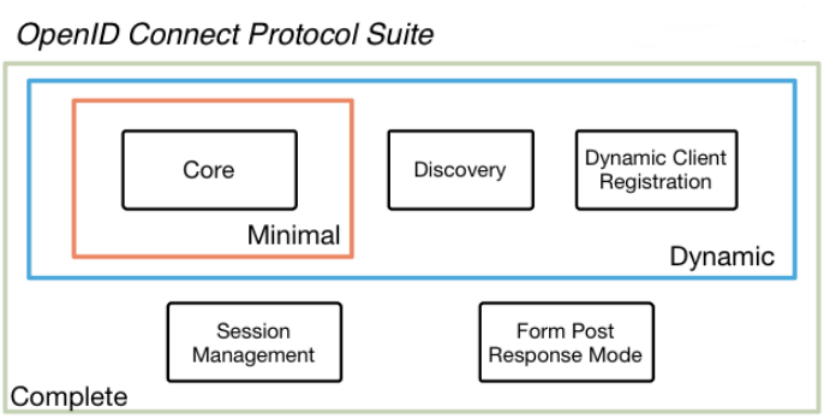


Figura 2.8: Módulos de OpenID Connect.

Fuente: https://openid.net/connect

OpenID Connect

OIDC es una especificación para manejar la gestión de identidades federadas creada en el año 2014 por la OpenID Foundation [21]. Este estándar permite realizar procesos tanto de autenticación como de autorización. Se posiciona como una simbiosis entre OpenID y OAuth 2.0. Toma como punto de partida OAuth 2.0, permitiendo así, realizar procesos de autorización y auditoría. Por otro lado, OIDC integra y extiende a OpenID, con el objetivo de dotar al estándar resultante de los procesos de autenticación. Todos estos procesos pueden ser implementados y accesibles por medio de una APIs, lo cual hace que su uso e implantación sea muy amigable para los desarrolladores.

OIDC se presenta como un protocolo ligero que permite que todo tipo de clientes (p. ej. aplicaciones web y aplicaciones móviles), puedan verificar y consultar información de las identidades digitales. Además, es flexible y extensible, permitiendo incorporar características adicionales como servicios de encriptación de la información, servicios de gestión de sesiones o el descubrimiento de proveedores de OpenID Connect de forma automática. OIDC esta compuesto por tres módulos funcionales que recogen todas estas funcionalidades (ver Figura 2.8):

- *Core*: define la funcionalidad básica y mínima para resolver los procesos de IAAA. Utiliza tres puntos de acceso: *Authorization Server Endpoint*, *token Endpoint* y *UserInfo Endpoint*.

- *Dynamic*: extiende el Core para incluir el servicio de descubrimiento dinámico de registro de clientes (*Discovery Dynamic Client Registration*). Utiliza dos nuevos punto de acceso: *Discovery Endpoint* y *Client Registration Endpoint*.

- *Complete*: añade los servicios de gestión de sesión y el *From Post Responde Mode* para codificar los parámetros de respuesta de autorización como valores de formularios HTML.

En OIDC se utilizan los tres mismos roles que se definen para OpenID. Estos son: EU (el usuario final), RP (la parte confiable o SP) y OP (OpenID Provider o IdP). Hace uso de los *tokens* definidos en OAuth 2.0, esto es, el *access* token, y el *refresh token*. Además, define un tercer *token* llamado *ID token*. El *ID token* contiene los atributos específicos asociados a una entidad (en inglés, *claims*) y que son necesarios durante el proceso de autenticación.

Estos roles interactúen entre sí por medio de peticiones y respuestas a dichas peticiones. Se distinguen seis tipos de ellas:

- Petición de autorización/autenticación: la realiza la RP al *Authroization Server Endpoint*, que se encuentra en el OP. El objetivo es solicitar el *Authorization Code*, *access token* y/o el *ID token* dependiendo del flujo de información a realizar. Se realiza mediante HTTP GET/POST.

- Respuesta de autorización/autenticación: Se envía desde el *Authorization Server Endpoint* a la URI de redirección indicada por la RP en la petición de autorización. Incluye el *Authorization Code* y los *tokens* solicitados, así como el estado y opcionalmente la caducidad de los *tokens*.

- Petición de tokens: Este realizada por la RP al *token Endpoint*, que se encuentra en el OP, para solicitar los *tokens*. Se realiza por medio de HTTP POST.

- Respuesta de Tokens: el *token Endpoint* realiza esta respuesta a la RP devolviendo los *tokens* solicitados. Estos *tokens* son el *access* token, el *ID token* y el *refresh token*.

- Petición de información del usuario: la RP realiza esta petición al *UserInfo Endpoint*, que se encuentra en el OP, para pedir información del EU. Se codifica mediante una petición HTTP GET/POST, e incluye el *access token* del EU con el objetivo de poder identificarlo.

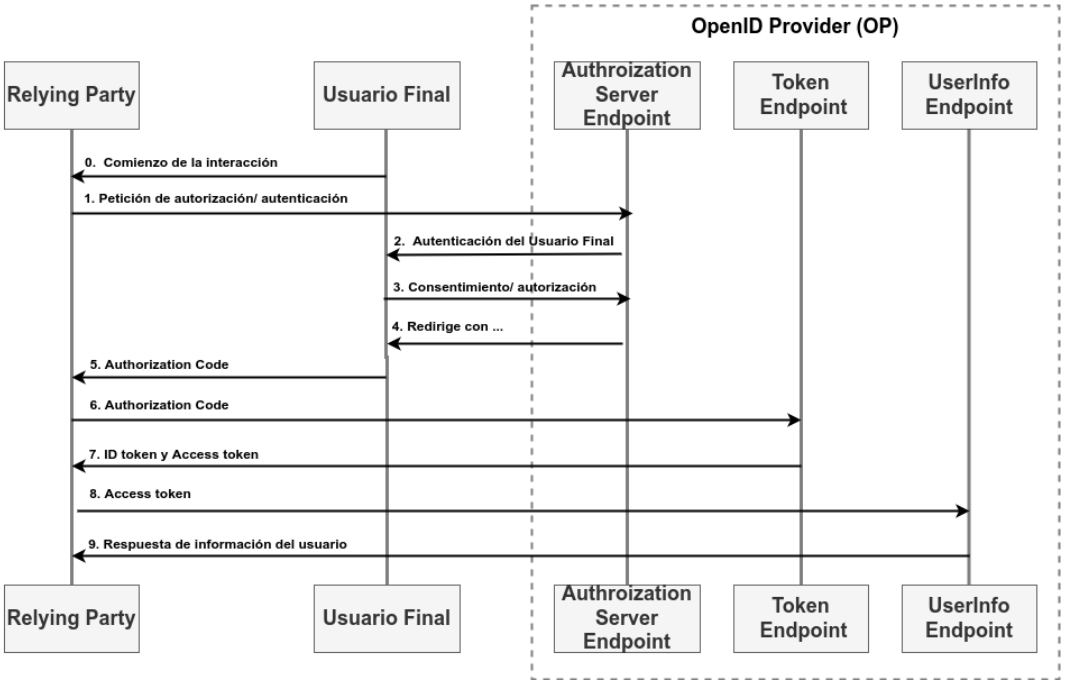


Figura 2.9: Flujo *Authorization Code* de OpenID Connect.

- Respuesta de información del usuario: el *UserInfo Endpoint* devuelve en formato JSON la información solicitada referente al EU identificado por el acess *token* recibido.

OIDC contempla tres flujos de información, muy parecidos a los proporcionados por OAuth 2.0. Estos flujos son: *Authorization Code, Implicit y Hybrid*. En primer lugar, el flujo *Authorization Code* es el análogo al de OAuth 2.0. Este flujo está representado en la Figura 2.9 y consta de los siguientes pasos:

0. El EU comienza a interaccionar con la RP iniciando el flujo de autorización/autenticación.
1. La RP envía una petición de autorización/autenticación al *Authorization Server Endpoint*.
2. El *Authorization Server Endpoint* se comunica con el EU para autenticarle.
3. El EU proporciona las credenciales al *Authorization Server Endpoint* o le da el consentimiento necesario.
4. El *Authorization Server Endpoint* valida las credenciales y en caso satisfactorio continua con el flujo y redirige al EU a la RP.

5. El *Authorization Server Endpoint* envía la respuesta de autorización/autenticación proporcionando el *Authorization Code* a la RP.

6. La RP utiliza el *Authorization Code* para enviárselo al *token Endpoint* del OP por medio de una petición de tokens.

7. El *token Endpoint* del OP valida el *Authorization Code* y en caso satisfactorio proporciona a la RP el *ID token* y el *access token* por medio de la respuesta de tokens.

8. De forma opcional, la RP utiliza el *access token* para enviar una petición de información del usuario.

9. El *UserInfo Endpoint* devuelve la información solicitada.

Por otro lado, el flujo *Implicit* se puede ver detallado en la Figura 2.10. La única diferencia con el flujo anterior viene en el paso 4 y 5, pues la RP recibe directamente el *ID token* y el *access token* ya que el *Authorization Server* puede determinar que dicho usuario ya estaba autenticado o autorizado previamente. Este flujo se consigue gracias a que el *Authorization Server* es capaz de verificar que el usuario esta previamente autenticado o autorizado por el uso de algún lenguaje de *Scripting* (p ej. *Javascript*). Este proceso se realiza, normalmente, recuperando una *cookie* por medio del agente de usuario. Este flujo es menos seguro que el *Authorization Code* pues un atacante podría obtener el *ID token* y el *access token* del usuario simplemente secuestrando una sesión activa del usuario, sin embargo, permite agilizar mucho el proceso cuando el *Authorization Server* y el EU ya confían en el cliente.

Finalmente, el flujo *Hybrid* es una combinación de los dos flujos anteriores, es decir, del *Authorization Code* y del *Implicit*. En este caso, se proporciona flexibilidad y, por lo tanto, dependiendo del tipo de conexión que se quiera utilizar y de la configuración del IdP, los *tokens* pueden ser proporcionado por el OP o por el *token Endpoint*.

En la Tabla 2.2 se puede observar las principales similitudes y diferencias entre los tres flujos.

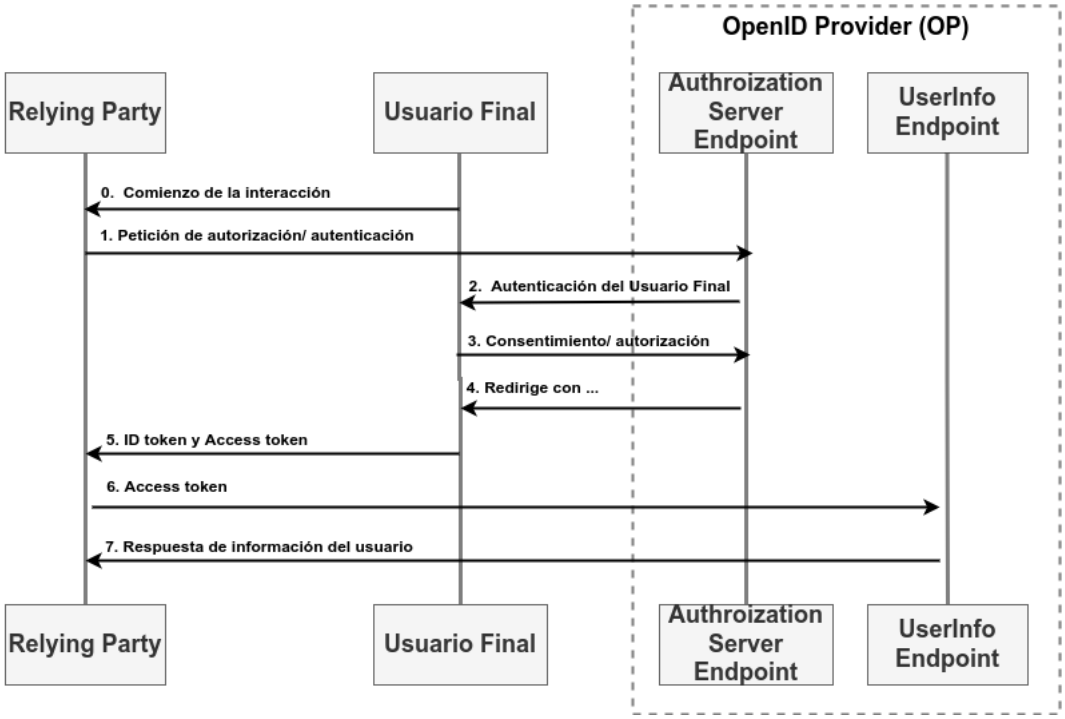


Figura 2.10: Flujo *Implicit* de OpenID Connect.

	Authorization Code	*Implicit*	*Hybrid*
Authorization Endpoint devuelve *tokens*	No	Si	Si
Token Endpoint devuelve *tokens*	Si	No	Si
Los *tokens* pasan por el agente de usuario	No	Si	Si
El cliente se puede autenticar	Si	No	Si
Puede usar *refresh tokens*	Si	No	Si

Tabla 2.2: Comparativa de flujos de información en OIDC.

2.1.5. Amenazas y soluciones actuales en la federación de identidades

Las especificaciones federadas son muy utilizadas hoy en día, sin embargo, como cualquier especificación o tecnología, existen amenazas asociadas tanto a la propia especificación como a las implementaciones de las mismas [36], [37]. Además, existen multitud de vulnerabilidades conocidas que pueden ser explotadas por cualquier atacante.

Las principales líneas de investigación y soluciones planteadas hasta el momento para tratar de solventar estas amenazas suelen estar orientadas al enriquecimiento de las peticiones y de

los *tokens* empleados, a realizar una mejora en la gestión de las sesiones de usuario, a la mejora de la propia implementación por medio de las APIs y SDKs ofrecidas, a la mejora de los flujos de información, al uso de criptografía a diferentes niveles o a la creación de políticas o sistemas de reputación [17].

Por ejemplo, profundizando en la especificación de OAuth, en [38] tratan de identificar todas las ambigüedades o aspectos que no quedan claros dentro de la propia especificación. Posteriormente, analizan implementaciones específicas para ver como se han solventado dichos aspectos, llegando a la conclusión de que casi un 60 % de las implementaciones no han sido implementadas correctamente y por lo tanto son vulnerables. En [39] demuestran que, por motivos de diseño, OAuth es vulnerable a sufrir suplantación a nivel de aplicación debido a los propios flujos de autorización y tipos de tokens de los que se disponen. En [40] proponen un modelo adaptativo que permite detectar a gran escala vulnerabilidades existentes y nuevas en las implementaciones de OAuth. Además proponen mitigar algunas de las nuevas vulnerabilidades encontradas que permiten materializar ataques como *Cross Site Request Forgery* (CSRF) y ataques de suplantación de identidad por medio de la mejora de los SDKs y de los propios tokens de la especificación. En [41] se propone modificar el estándar de OAuth para unificar todos los clientes externos y los flujos de información en uno común, con el objetivo de simplificar la configuración necesaria para su implementación. Además, se propone utilizar AMF para autenticar clientes externos, y firmas digitales y criptografía para mitigar posibles riesgos y vulnerabilidades asociados al mal uso de los *tokens*. En [42] se analizan las principales APIs proporcionadas por los mayores IdPs y los flujos de autorización de OAuth para analizar las posibles implicaciones a la privacidad de los usuarios finales.

En cuanto a la especificación de OIDC, en la propuesta [36] descubren que multitud de ataques ya existentes para otros protocolos de SSO son igualmente aplicables realizando pequeñas modificaciones de los mismos. Además, proponen dos nuevos ataques para materializar nuevas vulnerabilidades asociadas a los propios flujos de la especificación. Finalmente, proponen soluciones para mitigar estos ataques basadas en la mejora del propio estándar, las cuales han sido actualmente incluidas en el propio estándar, y soluciones de menos impacto basadas en la mejora de los propios *tokens* y en la mejora de las implementaciones. En [16] realizan un análisis formal de la seguridad de la especificación y proponen métodos para solventar las

vulnerabilidades encontradas basándose en la mejora de las propias implementaciones modificando los flujos de información y los *tokens*. En [43] y [44] tratan de mejorar la privacidad de OIDC por medio del uso de técnicas criptográficas y la modificación de los propios flujos de información. En [45] proponen una serie de políticas y técnicas criptográficas en las distintas comunicaciones de los agentes para mejorar la privacidad de los usuarios. En [17] analizan amenazas tanto de la seguridad como de la privacidad y proponen diversas técnicas para mitigarlas basadas en la mejora de los tokens, en la mejora de la implementación, en la mejora de los flujos de información, en aspectos de criptografía y en la creación de políticas y sistemas de reputación. En [46] se propone un método para mejorar la privacidad de los usuarios basado en realizar pequeñas modificaciones a los flujos de información ya existentes. En [47] proponen un servicio novedoso de *tokens* para mantener los *access tokens* durante un mayor periodo de tiempo aumentando la seguridad de los tokens de larga duración (usualmente menos seguro que los que tienen un ciclo de vida corto). En [48] se proponen una serie de buenas prácticas para mejorar la seguridad de OAuth y OIDC en aplicaciones nativas para clientes Android. Además demuestran que la gran mayoría de las implementaciones actuales son vulnerables a todo tipo de ataques como la suplantación de identidad debido a una mala implementación de las buenas prácticas propuestas.

En la Tabla 2.3 se pueden observar las características principales de los trabajos analizados anteriormente.

Por último, cabe destacar que la rama del análisis de comportamiento se ha posicionado como una línea de investigación muy recomendable y adoptada para mejorar los sistemas de control de accesos y gestión de identidades [4]. Sin embargo, tal y como se ha podido ver hasta ahora, los trabajos que tratan de mejorar los niveles de seguridad de los estándares de gestión de identidades federados no integran o implementan estas técnicas. Además, estas técnicas son de especial interés para lograr implementar autenticación continua, un recurso poco utilizado hasta ahora en la federación de identidades.

Trabajo	Especificación	Ámbito	*Token*	Criptografía	Implementación	Políticas/ Reputación	Flujo
[37]	OAuth	S	-	-	-	-	-
[38]	OAuth	S	-	-	-	-	-
[39]	OAuth	S	-	-	-	-	-
[40]	OAuth	S	✓	-	✓	-	-
[41]	OAuth	S	✓	✓	-	-	✓
[42]	OAuth	P	-	-	-	✓	-
[16]	OIDC	S	✓	-	✓	-	✓
[36]	OIDC	S	✓	-	✓	-	-
[43]	OIDC	P	-	✓	-	-	✓
[44]	OIDC	P	-	✓	-	-	✓
[45]	OIDC	P	-	✓	-	✓	-
[17]	OIDC	S/P	✓	✓	✓	✓	✓
[46]	OIDC	P	-	-	-	-	✓
[47]	OIDC	S	✓	✓	✓	-	✓
[48]	OAuth /OIDC	S	-	-	-	-	-

Tabla 2.3: Trabajos de la literatura sobre amenazas y mejoras de los esquemas de gestión de identidades federados. La columna Ámbito representa si el trabajo se centra en mejorar la seguridad (*S*) o la privacidad (*P*). Las columnas Token, Criptografía, Implementación, Políticas/Reputación y Flujo representan los elementos donde se centran las soluciones propuestas de los trabajos analizados.

2.2. Análisis de comportamientos

En el ámbito tecnológico, el análisis de comportamientos se centra en analizar, modelar y predecir los comportamientos pasados, presentes y futuros de los usuarios o entes que interactúan con un sistema de información, con el objetivo de obtener un beneficio de negocio [4]. Es comúnmente conocido por sus siglas en inglés, *User and Entity Behavior Analysis* (UEBA). El uso de técnicas de análisis de comportamientos aporta grandes ventajas para conseguir multitud de objetivos, a lo largo de una gran variedad de dominios de aplicación. En la Figura 2.11, se pueden observar los cuatro grandes dominios de aplicación de este área hoy en día, y sus áreas específicas de aplicación. Estos dominios son: redes, seguridad y salud, mejora de un servicio y ciberseguridad.

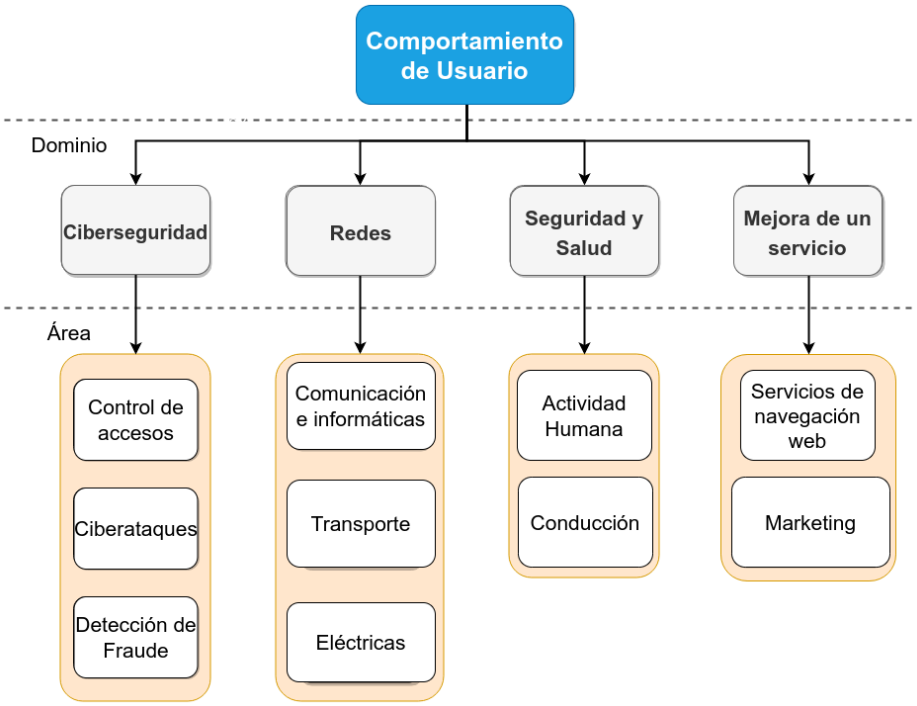


Figura 2.11: Dominios y áreas específicas de aplicación de los trabajos de análisis de comportamiento.

En el dominio de las redes, el análisis de comportamientos es una herramienta muy efectiva a la hora de mejorar las redes de comunicación, de transporte y redes eléctricas. En este ámbito, el objetivo suele ser mejorar la eficiencia de las distintas dichas redes, por ejemplo, extrayendo patrones de uso que permiten establecer distintos perfiles de usuario. Estos patrones se pueden utilizar para detectar posibles congestiones en la red o cuellos de botella, permitiendo desviar el tráfico de forma óptima dependiendo del perfil de usuario, tanto en redes informáticas [49], como en redes de transporte público o privado [50], [51]. Otra aplicación, es la predicción del precio del mercado eléctrico con el objetivo de que los productores eléctricos puedan ganar las subasta energética a sus competidores y obtener un mayor beneficio de ello [52]. En este área, con la incorporación de los contadores inteligentes, también se utilizan estas técnicas para mejorar la eficiencia de la propia red y por lo tanto poder ofrecer precios más competitivos a los consumidores [53].

En el dominio de la seguridad y salud, el análisis de comportamientos es de gran utilidad para la detección temprana de ciertos factores de interés que pueden suponer un riesgo para la salud o la seguridad de un individuo. Por ejemplo, con la incorporación de sensores en

una vivienda se puede detectar si una persona ha sufrido un accidente o una caída en el caso de personas mayores [54], [55], o se puede detectar si una persona está perdiendo capacidad cognitiva debido al cambio de su comportamiento [56]. Más específicamente relacionado con la seguridad, hoy en día multitud de vehículos disponen de sistemas basados en el análisis de comportamiento para detectar situaciones adversas, como un posible accidente [57] o fatiga y cansancio [58].

En el dominio de la mejora de un servicio, el análisis de comportamientos permite generar perfiles de usuario que son utilizados para entender las carencias del servicio y, por consiguiente, poder mejorarlo u obtener un beneficio por ello. Algunos ejemplos son, los sistemas de recomendación que permiten aumentar las ventas de un comercio tanto electrónico como físico [59]. La fidelización de clientes o el lanzamiento de campañas específicas de publicidad son algunos otros ejemplos en los que las técnicas de análisis de comportamientos son efectivas en este dominio [60].

En el dominio de la ciberseguridad, el análisis de comportamientos se utiliza para elaborar sistemas de control de accesos, para detectar ciberataques y para la detección de fraude. Los sistemas de control de accesos basados en técnicas de análisis de comportamientos se fundamentan en analizar los patrones de comportamiento de un usuario o entidad específica con el objetivo de poder realizar los procesos de IAAA frente a un sistema de información [61], [62]. En el campo de la detección de ciberataques, el análisis de comportamientos se utiliza para detectar patrones anómalos, por ejemplo, durante la ejecución de un software que pueden causar daños en el sistema [63] o en las comunicaciones de una red [64]. Por último, el análisis de comportamiento es útil en la detección de fraude. En este ámbito, el análisis de comportamiento en el uso de tarjetas de crédito puede ayudar a prevenir la realización de transacciones fraudulentas [65].

De aquí en adelante, se analizan los trabajos relacionados con el análisis de comportamiento en el dominio de la ciberseguridad y específicamente en el área del control de accesos.

2.2.1. Análisis de comportamientos para el control de accesos

En esta sección se analizan los trabajos del estado del arte que utilizan técnicas de análisis de comportamiento para solventar el control de accesos. Los trabajos, aquí recopilados, se centran mayoritariamente en la autenticación, tanto progresiva como autenticación continua. El objetivo de estas propuestas es tratar de modelar la información de comportamiento para extraer y detectar patrones intrínsecos a cada usuario. Estos patrones se utilizan posteriormente para evaluar nuevas muestras, pudiendo determinar si concuerdan, es decir, pertenecen a un usuario legítimo, o por el contrario si son diferentes, es decir, son anomalías de comportamiento y por lo tanto es probable que no pertenezcan al usuario legítimo y puedan suponer una amenaza para la seguridad. En este último caso, el sistema debe evaluar los riesgos y proceder a realizar las contramedidas necesarias, como solicitar al usuario nuevamente sus credenciales, o utilizar un segundo factor de autenticación con el objetivo de corroborar que el usuario es legítimo.

Los trabajos del estado del arte en este ámbito se dividen principalmente en dos categorías dependiendo del dispositivo donde se aplican los modelos de análisis de comportamientos: teléfonos inteligentes y ordenador. Los modelos de aprendizaje máquina que se aplican en ambas categorías suelen ser de la misma naturaleza y por lo tanto muy parecidos. Sin embargo, los procesos de recopilación de información, las fuentes de información utilizadas (distintas dinámicas de comportamiento) y su integración final (debido a los recursos limitados de los teléfonos inteligentes) son más dispares.

En el caso de los teléfonos inteligentes, las fuentes de información son principalmente los sensores que están habitualmente integrados en dichos dispositivos. Estos sensores se pueden categorizar en cuatro grandes grupos [62]: movimiento (p. ej. acelerómetro y giroscopio), entorno (p. ej. luz ambiente y temperatura), posición (p. ej. GPS y brújula) y de pantalla (p. ej. presión y capacitivo). El objetivo de los trabajos que se engloban en este ámbito suele ser detectar anomalías en el comportamiento del usuario a partir de la información recogida por estos sensores. Estas anomalías se deben principalmente a la suplantación de identidad, ya sea por un robo de credenciales (robo de los autenticadores que pertenecen al usuario legítimo) o por un secuestro de sesión (obtención de una sesión activa perteneciente a un usuario legítimo para lograr un acceso no autorizado). Además, los trabajos categorizados aquí, se pueden dividir a su vez en los que se centran en la autenticación progresiva y los que se centran en la autenticación

continua.

En la autenticación progresiva, las propuestas suelen estar centradas en evaluar una determinada petición de acceso a una aplicación o recurso restringido, y categorizarla en función de los valores recopilados por los sensores. En este ámbito, normalmente se suelen utilizar los sensores de posición como el GPS, y los sensores de pantalla. Un ejemplo de la utilización de sensores de posición se encuentra en [66], donde los usuarios se agrupan de acuerdo a un algoritmo de *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) basándose en sus posiciones. Posteriormente, los grupos y las transiciones entre grupos se modelan como un proceso de Markov, permitiendo utilizar un *Hidden Markov Model* (HMM) para determinar si una petición es legítima o no. Por otro lado, en [67] utilizan las dinámicas de pulsación de teclado recopiladas por los sensores de pantalla. Esta información longitudinal se agrupa en ventanas para poder comparar las dinámicas de comportamiento entre sí. De este modo, las nuevas dinámicas que se quieren evaluar se comparan con el histórico del usuario, pudiendo así obtener la distancia entre ellas y clasificarlas en legítimas o impostoras en función de un umbral.

En la autenticación continua, existe más variabilidad respecto a las fuentes de información. En este caso, los sensores de movimiento, entorno y los sensores de la pantalla suelen ser los más utilizados. Independientemente de la fuente, cabe destacar que la naturaleza de los datos recogidos es la misma, es decir, son datos longitudinales. En primer lugar, se va a considerar los trabajos que utilizan información recogida de los sensores de pantalla. Normalmente, el tratamiento de estos datos comienza con la preparación y limpieza de los mismos, esto es, con los procesos de normalización y estandarización [61], generación de datos artificiales [68] y extracción de descriptores y medidas de centralidad y de dispersión (p. ej. media y desviación típica). Además, existen algunos trabajos que también utilizan la lógica difusa [69] para extraer conocimiento [70]. Normalmente, los trabajos en este ámbito suelen representar la información en forma de secuencias. El siguiente paso que siguen es modelar la información limpia y procesada. Para ello se utilizan múltiples algoritmos de aprendizaje máquina. Cabe destacar el uso de *K-Nearest Neighbors* (KNN) y de *Support Vector Machine*s (SVMs) cuando las secuencias generadas son lo suficientemente largas [61]. Por otro lado, *Naïve Bayes* (NB), *Bayesian Network* (BN) y *Neural Network* (NN) suelen funcionar correctamente para conjuntos de datos pequeños

[71].

En el caso de los trabajos que utilizan información recogida del acelerómetro y del giroscopio, los algoritmos de clasificación de clase única son los más utilizados. Por ejemplo, en [68] utilizan técnicas de aumento de datos, extracción exhaustiva de características y el algoritmo de clasificación *One-Class SVM* (OC-SVM) para definir los patrones de uso de cada usuario. Estos patrones extraídos, permiten distinguir entre las dinámicas de comportamiento que se consideran normales para un usuario, y las muestras atípicas, las cuales se clasifican como muestras no pertenecientes al usuario legítimo. Además, existen otros trabajos que tratan de detectar estos mismos patrones utilizando un HMM de clase única en múltiples escenarios como, sostener el dispositivo, coger el dispositivo desde una mesa y sostener el dispositivo mientras se camina [62]. Cabe destacar la propuesta [72], donde modelan información de estos sensores combinados con técnicas de criptografía con el objetivo de garantizar la privacidad de los usuarios, no viéndose afectado significativamente el rendimiento en cuanto a eficacia y eficiencia de los modelos propuestos.

Para los sensores de ambiente, normalmente se utilizan algoritmos como KNN, NB y *Hoeffding Adaptive Trees* (HAT). Estos sensores, además, se suelen utilizar en combinación con otro tipo de sensores e indicadores como, por ejemplo, el uso de la batería con el objetivo de mejorar los modelos previamente desarrollados. Estas propuestas suelen ser las menos invasivas para la privacidad de los usuarios [73].

Por otro lado, en los trabajos que se centran en el dispositivo del ordenador personal, también se analiza tanto la autenticación progresiva, como la autenticación continua. Para el primer caso, las principales fuentes de información son los registros de navegación y la información recogida de los sistemas de gestión de identidades, es decir, registros de peticiones de accesos a recursos. En el caso de la autenticación continua, se suelen utilizar las dinámicas de comportamiento recogidas desde el teclado y las dinámicas de comportamiento recogidas del ratón.

La mayoría de los trabajos que utilizan el ordenador personal, se centran en agrupar las interacciones de los usuarios legítimos, con el objetivo de definir los comportamientos esperados para cada usuario y poder así evaluarlos frente a las nuevas muestras. Por ejemplo, para el caso de los registros de navegación, estos se suelen agrupar en sesiones. Cada una de estas sesiones está formada por una secuencia de acciones, que recoge las dinámicas de comportamiento de

cada usuario durante un intervalo de tiempo. De esta forma, por ejemplo, se utilizan medidas simples de similitud [74], SVM [75] y el modelo de Markov [76] para poder compararlas y clasificarlas.

En el caso de las propuestas en las cuales los datos provienen de sistemas de gestión de identidades, los datos normalmente consisten en secuencias de peticiones de acceso a aplicaciones, servicios o recursos. El principal objetivo de estos trabajos es determinar la legitimidad de una petición teniendo en cuenta los atributos de la misma, y poder establecer el riesgo asociado. Al igual que en el caso anterior, las aproximaciones en este ámbito suelen utilizar la información legítima para alimentar un algoritmo de aprendizaje máquina. Posteriormente, este algoritmo se utiliza para comparar las nuevas muestras, y categorizarlas en legítimas o no legítimas acorde a un umbral de decisión. De esta forma, cuando una nueva muestra sobrepasa el umbral, el sistema lanza una alerta que puede ser utilizada para avisar a los administradores del mismo, o directamente proceder a tomar una contramedida de forma automática como, por ejemplo, rechazar la petición. Algunos ejemplos representativos de algoritmos que se utilizan en este ámbito son SVM [77], NB [78], técnicas de reducción de la dimensionalidad [79] y OC-SVM [80].

Como se ha podido observar hasta ahora, uno de los factores que más diferencian a las propuestas analizadas es la fuente de información que se utiliza para diseñar el modelo de análisis de comportamiento. Es por esto que es de gran importancia analizar en detalle los trabajos que encajan en cada una de estas fuentes de información específicas.

2.2.2. Fuentes de información específicas y su combinación

Los trabajos aquí presentados se han categorizado en función de la fuente de información utilizada para modelar el comportamiento de los usuarios. De este modo, se presentan tres categorías de acorde a los intereses de la presente investigación. Estas fuentes de información son: teclado, ratón y combinación de información.

En cuanto a la categoría del teclado, los primeros trabajos en el área empezaron a surgir en 1980 con el primer análisis de las dinámicas de comportamiento asociadas a dicha fuente de información [81]. Posteriormente, los trabajos en este área empezaron a utilizar clasificadores

bayesianos [82], NNs [83] o técnicas *clustering* [84] para modelar el comportamiento de los usuarios y poder detectar así anomalías en el comportamiento de los usuarios. Estas anomalías entonces podían ser categorizadas como posibles brechas de seguridad. Estos trabajos, aún siguen siendo de gran utilidad y permiten hoy en día obtener resultados satisfactorios a la hora de detectar comportamientos sospechosos.

Con el paso del tiempo han ido surgiendo nuevos trabajos que han ido utilizando otras técnicas o han probado más clasificadores para solventar el mismo problema. Por ejemplo, en [85] se utilizan 14 clasificadores, incluyendo algunos basados en la distancia de Mahalanobis, la distancia de Manhattan, KNN, NNs, K-means, lógica difusa y SVMs, para comparar y analizar la eficacia de dichos clasificadores a la hora detectar impostores cuando ingresan una cierta contraseña predefinida. En [86] se utiliza una *Ant Colony* (AC) para realizar un primer paso de selección de características, de esta forma, se pueden ordenar por importancia y seleccionar las más representativas, aunque no sean las más convencionales en la literatura. En el trabajo [87] se presenta un método novedoso para seleccionar las características más representativas, de forma independiente para cada usuario en particular, y posteriormente generar un modelo independiente para cada uno de ellos. Esta propuesta se basa en utilizar un modelo de estimación de densidades gausiano, Parzen window density estimation, OC-SVM, KNN y K-means. En [88] se demuestra que, para ciertos casos de uso, las variables no agregadas son más discriminatorias a la hora de comparar dinámicas de comportamiento recogidas del teclado. Utilizando este tipo de variables, se entrenan NB, *Tree Augmented Naïve Bayes* (TANB), KNN y modelos de regresión logística para detectar usuarios impostores. Cabe destacar que los mejores resultados los obtienen cuando combinan vectores de variables tanto agregadas como variables no agregadas de forma simultánea. En [89], se propone una técnica novedosa para transformar los vectores de características en espectrogramas de frecuencias, consiguiendo así transformar los datos longitudinales (es decir, señales) en una imagen. Posteriormente, utilizan una NN basada en la optimización de Gauss-Newton para clasificar estas imágenes y poder determinar así, si el vector de características original es de un impostor o por el contrario pertenece a un usuario genuino. En la propuesta [90], se utilizan NN convolucionales y recurrentes de forma combinada para generar un modelo efectivo para el conjunto de datos, bien conocido en la literatura, de Buffalo [91]. En [92] se considera utilizar *Kernel Density Estimation* (KDE) para compararse contra algoritmos populares en la literatura en este ámbito, en multitud de conjuntos de

datos como el de Clarkson, Torino y Buffalo. Por último, en [93], proponen utilizar una métrica basada en *Instance-based Tail Area Density* (ITAD) para reducir el número de interacciones necesarias para autenticar a un usuario, de esta forma, mejoran la eficiencia de los modelos y reducen latencias, no afectando a la eficacia.

El análisis de las dinámicas de ratón con el objetivo de autenticar usuarios comenzó en los años 2000 [94]. Posteriormente, en el trabajo [95], se empiezan a considerar variables relevantes como la velocidad de movimiento, la dirección, el tipo de acción que se realiza (movimiento, clic, desplazamiento de la rueda), la distancia recorrida y el tiempo transcurrido entre acciones. Este proceso de extracción de características se utiliza para generar un vector de variable que alimenta una NN obteniendo resultados satisfactorios. En el trabajo [96], también utilizan NN, en este caso convolucionales, recurrentes y un modelo híbrido que considera ambos tipos, basándose en la propagación de la relevancia por capas. Estos algoritmos son evaluados utilizando múltiples conjuntos de datos bien conocidos de la literatura. En [97], las variables comúnmente utilizadas en la literatura se categorizan en holísticas y procedimentales. Posteriormente, comparan que tipo de características son más discriminatorias a la hora de autenticar usuarios utilizando una OC-SVM. La investigación propuesta en [98] utiliza el algoritmo de *Progress-Adjusted Dynamic Time Wraping* (PADTW) combinado con un algoritmo de segmentación para transformar las variables originales y alimentar así un clasificador SVM. En [99], convierten las variables temporales en imágenes utilizando una función de mapeo que permite aumentar la dimensión de los datos. A continuación, utilizan una NN convolucional para comparar las imágenes entre sí y poder determinar si los vectores originales de características pertenecen a un usuario genuino o a un usuario impostor.

Finalmente, está surgiendo una nueva línea de investigación que trata de combinar información recogida de múltiples fuentes de datos heterogéneas. Esto es, generar modelos de aprendizaje máquina que permiten utilizar información del teclado y del ratón simultáneamente. Esto se traduce en que los clasificadores generados aumentan su eficacia con respecto a solo los que utilizan una única fuente de información. De forma general, existen dos formas de conseguir combinar la información: a nivel de decisión y a nivel de características. Ambas formas de combinar la información presentan multitud de ventajas y mejoran de forma notable la eficacia en los sistemas de autenticación, aunque añaden cierta complejidad a los modelos finales. Las

bases de la combinación de información en el ámbito de las dinámicas de comportamiento se fijaron en [100]. En este trabajo, se aborda la combinación de información, tanto a nivel decisión como a nivel características para el reconocimiento facial, el reconocimiento de huella dactilar y reconocimiento de las texturas y formas de la mano, mejorando considerablemente los resultados de los trabajos de la literatura existentes.

La combinación a nivel de decisión se basa en generar un modelo de aprendizaje máquina de forma independiente para cada una de las fuentes de información. De esta manera, cada modelo se entrena para con un tipo de datos específicos. A la hora de dar una predicción en un instante concreto de tiempo, cada modelo de aprendizaje máquina procesa la información de una fuente de información específica, dando una probabilidad de pertenecer a la clase genuina o impostora hasta ese instante. Finalmente, se combinan las probabilidades de todos los modelos, obteniendo una probabilidad final y por lo tanto permitiendo categorizar muestras de múltiples fuentes de información.

En [101] se implementa un Modelo de Confianza (MC) basado en ajustar de forma dinámica los pesos de los modelos independientes generados para el teclado y el ratón, utilizando algoritmos genéticos. Los modelos propuestos en este trabajo son NNs y Counter-Propagation Artificial NNs y utilizan una SVM para combinar ambos modelos. Además, proponen otro método en el que prueban diferentes métricas de distancias como paso previo al entrenamiento de los modelos, con el objetivo de no utilizar datos de impostores en esta fase. En [102], utilizan una combinación basada en utilizar NB para representar cada fuente de información a el mismo espacio de decisión. Posteriormente, utilizan una SVM para clasificar las muestras. En [103], evalúan los modelos de *Random Forest* (RF), SVM, *Decision Tree*s (DTs) y BN con el mismo objetivo, es decir, combinar la información a nivel de decisión. En [104] se propone combinar información de contexto de la sesión con información de comportamiento recogida del teclado y del ratón para mejorar la eficacia de los sistemas de autenticación continua. Para evaluar su propuesta, utilizan el conjunto de datos *The Wolf of SUTD* (TWOS) [105]. En primer lugar, generan un modelo que utiliza única y exclusivamente la información de contexto. Por otro lado, implementan otro modelo que combina tanto la información de teclado como la de ratón, durante una sesión. La combinación de ambos modelos se evalúa por medio de tres modelos: *Parametric Linear Combination* (PLC), un clasificador de RF y un clasificador de SVM.

De esta manera, para cada sesión se evalúa la información proveniente de las tres fuentes de información obteniendo una única predicción.

La combinación a nivel de características se basa en generar un único modelo de aprendizaje máquina que permita evaluar de forma simultánea las características provenientes de múltiples fuentes de información. De esta manera, este modelo puede clasificar una muestra que contenga información de una o múltiples fuentes de información sin necesidad de tener que entrenar un modelo para cada una de las mismas. Algunos ejemplos de trabajos en este ámbito se analizan a continuación.

En [106] se utiliza un *Multi-kernel Learning Method* (MKL) para combinar las características de las fuentes de información de teclado y de ratón. Posteriormente, se evalúan los modelos de DT, RF, NB, OC-SVM y SVM entrenados con el kernel obtenido, obteniendo resultados prometedores. En [107] se compara tanto la combinación a nivel de decisión como la combinación a nivel de características. En primer lugar, para combinar a nivel de decisión se genera un modelo de NB para las dinámicas de teclado y una SVM para las dinámicas de ratón. Posteriormente, ensamblan ambos modelos utilizando un J48 DT para combinar las predicciones obtenidas para cada modelo de forma independiente. A nivel de características, se utiliza *Principal Component Analysis* (PCA) para realizar la combinación y posteriormente evaluar los modelos BN, J48 y SVM de forma independiente. Cabe destacar, que también utilizan una tercera fuente de información proveniente de las interacciones que realizan los usuarios con la interfaz gráfica. Por último, en [108] proponen utilizar estas técnicas de combinación de la información a nivel de características utilizando fuentes de información provenientes de los sensores de los teléfonos inteligentes. Nuevamente, se confirma que la combinación de información supone una gran ventaja a la hora de aumentar la precisión y eficacia para detectar comportamientos sospechosos, frente a las propuestas que no lo utilizan.

La Tabla 2.4 muestra los trabajos relacionados con el método propuesto. Las propuestas que únicamente utilizan una fuente de información (es decir, única y exclusivamente teclado o ratón), han sido reducidas y seleccionadas de acorde a la relación con la presente propuesta, debido al gran número de trabajos en este ámbito.

Trabajo	Dinámica de Comportamiento	Método	Nivel de combinación	Datos	Interacción Libre
[85]	T	14 clasificadores	-	.tie5Roanl	No
[86]	T	DT + SVM + AC	-	Propio	Si
[87]	T	Gauss + Parzen + OC-SVM + k-NN + K-means	-	Propio	Si
[88]	T	NB + TANB + KNN	-	Propio	No
[89]	T	NNs	-	Propio	No
[90]	T	NNs	-	Buffalo	Si
[92]	T	KDE	-	Buffalo + Clarkson + Torino	Si
[93]	T	ITAD	-	Buffalo	Si
[95]	R	NNs	-	Propio	Si
[96]	R	NNs	-	Balabit + TWOS	Si
[97]	R	OC-SVM	-	Propio	No
[98]	R	PADTW	-	Propio+ [97]	No
[99]	R	NNs	-	Balabit	Si
[101]	T + R	MC +NNs +SVM	Decisión	Propio	Si
[102]	T + R	NB + SVM	Decisión	Propio	Si
[109]	T + R	BN + BFS	Decisión	Propio	Si
[103]	T + R	RF+ SVM+ DT+ BN	Decisión	Propio	Si
[104]	T + R+ C	RF+SVM+PLC	Decisión	TWOS	Si
[106]	T + R	MKL + DT + RF + NB+ OC-SVM + SVM	Características	Propio	Si
[107]	T + R	BN + J48 + SVM	Decisión y Características	Propio	Si

Tabla 2.4: Comparación de trabajos previos relacionados con el análisis de comportamiento. *T* y *R* representan las dinámicas de teclado y de ratón respectivamente. *C* se refiere a información de contexto. La columna datos representa el conjunto de datos utilizado el trabajo específico. La columna interacción libre representa si el conjunto de datos contiene dinámicas de comportamiento recogidas en un entorno en el que el usuario es libre de interactuar con el sistema sin realizar una tarea predeterminada.

2.3. Limitaciones de los trabajos previos

Después de analizar en profundidad el estado del arte, se han encontrado las siguientes limitaciones:

- Las principales líneas de investigación y soluciones para mejorar los niveles de seguridad están orientadas al enriquecimiento de las peticiones y/o de los *tokens*, a realizar una mejora en la gestión de las sesiones de usuario, a la mejora de la propia implementación

por medio de las APIs y SDKs ofrecidas, a la mejora de los flujos de información, al uso de criptografía a diferentes niveles o a la creación de políticas o sistemas de reputación. No se conocen metodologías o flujos de trabajo para integrar el análisis de comportamiento en los sistemas de gestión de identidades federados con el objetivo de mejorar la seguridad de los mismos.

- Los trabajos previos que proponen utilizar técnicas de análisis de comportamientos para solventar los procesos de IAAA se suelen centrar principalmente en la autenticación adaptativa, ya sea centralizada o distribuida. En casi ningún caso, estas soluciones se integran dentro de un protocolo, *framework* o estándar de gestión de identidades, y mucho menos en los estándares federados en los que se tiene en cuenta a un IdP externo.

- Los trabajos previos para detectar anomalías de comportamientos se suelen centrar en un dominio específico de aplicación. Esto hace que los modelos puedan estar sesgados y no sean capaces de generalizar correctamente.

- El número de trabajos de análisis de comportamiento que combinan información es muy reducido, siendo más notable en el caso de la combinación a nivel de características. Sin embargo, estos trabajos son los que suelen obtener mejores resultados en cuanto a eficacia a la hora de detectar comportamientos anómalos.

- Existen una falta de conjuntos de datos públicamente disponibles que contengan dinámicas de comportamiento. Esto se ve reflejado en que la mayoría de propuestas utilizan conjuntos de datos propios recogidos normalmente en un entorno de laboratorio.

De aquí en adelante se trata de superar todas estas limitaciones encontradas. Para ello se propone un flujo de trabajo para integrar las técnicas de análisis de comportamiento dentro de las especificaciones de gestión de identidades federadas. Posteriormente se propone un modelo novedoso de análisis de comportamientos que permite combinar información a nivel de características. Finalmente se evalúan dichas propuestas y se detalla la creación de un conjunto de datos específico que contiene dinámicas de comportamiento.

Capítulo 3

Flujo de trabajo para la integración en los estándares federados

En este capítulo se propone un flujo de trabajo, que cualquier RP puede implementar, para integrar las técnicas de análisis de comportamientos en los esquemas de gestión de identidades federados (p. ej. OAuth y OIDC) [110]. Tal y como se ha podido ver en el capítulo del estado del arte (Sección 2.1), en las especificaciones federadas, las credenciales de los usuarios son almacenadas por el IdP. Cuando un usuario trata de acceder a un recurso, servicio o aplicación (es decir, SP o en el entorno federado RP), la RP confía en el IdP para solventar los procesos de IAAA. De este modo, el usuario final se autentica de forma externa a la RP, es decir, en el IdP, obteniendo una acreditación en forma de *token*. Finalmente, el usuario final utiliza este *token* para comunicarse con la RP, y poder así interactuar con ella.

La solución propuesta en esta tesis doctoral para mejorar la seguridad de estos esquemas federados se basa en utilizar los modelos de análisis de comportamientos. Cuando una RP utiliza un flujo federado para solventar los procesos de IAAA, está delegando parte de la seguridad en el IdP. Sin embargo, delegar los procesos de IAAA en un agente externo no tiene por qué significar delegar todos los aspectos de la seguridad como, por ejemplo, desde el punto de vista de la prevención o de la detección.

La RP tiene un rol muy significativo a la hora de proteger a sus usuarios de recibir un ciber-

ataque como, por ejemplo, ataques de suplantación de identidad, ya que los usuarios interactúan la mayor parte del tiempo con ella. Cuando se utiliza un esquema de gestión de identidades federado, la mayoría de las partes implicadas asumen que la seguridad pasa a ser responsabilidad del IdP en exclusiva, siendo esta una suposición errónea. Tanto IdP como RP deben estar concienciados en asumir su responsabilidad en cuanto a la seguridad, para poder proteger a los usuarios de las posibles amenazas que existen.

De aquí en adelante, se detalla el flujo de trabajo propuesto, que puede utilizar cualquier RP con el objetivo de mejorar los niveles de seguridad proporcionados al usuario final. Este flujo de trabajo se centra en la prevención y detección de ataques de suplantación de identidades por medio del robo de credenciales o secuestros de sesión. Se basa en utilizar técnicas de análisis de comportamiento, y en los detalles de integración con los principales estándares de gestión de identidades.

3.1. Arquitectura y premisas

A lo largo de este capítulo se asume una arquitectura federada en la que existen tres roles principales: el EU, el IdP y la RP.

Las RPs necesitan conocimiento acerca del comportamiento de sus usuarios. Este conocimiento se suele extraer de diferentes tecnologías, interfaces o APIs que son capaces de recopilar información relativa al software o hardware utilizado por el EU, aspectos de configuración (ya sea del sistema o del cliente web utilizado) y de su comportamiento. Toda esta información se utiliza para generar una huella digital, capaz de identificar única y exclusivamente a cada EU.

Una huella digital puede suponer una invasión para la privacidad de los usuarios de los que se recopila información. Estas huellas se pueden utilizar para fines como el perfilado de usuario o para la personalización de contenido publicitario, lo cual puede suponer un rechazo por parte de los usuarios y, por consiguiente, la negativa a su adopción [111]. Hoy en día, existen multitud de trabajos de investigación y de productos finales de empresas privadas que protegen a los clientes web y a las aplicaciones de los usuarios, bloqueando los códigos encargados de la recopilación de la información o cambiando los valores de los atributos que estos recopilan [112].

Por todo lo expuesto anteriormente, en este trabajo se asumen las siguientes premisas:

- Existe suficiente diversidad en el comportamiento de los usuarios como para poder generar una huella digital, única y identificativa para cada usuario.
- La RP no colabora con el IdP para generar esta huella, ni para analizarla y detectar posibles anomalías que suponen una brecha de seguridad para la RP. Esto quiere decir que, la RP es capaz de seguir el flujo de trabajo propuesto de forma independiente y utilizando única y exclusivamente sus propios recursos.
- La RP informa a los usuarios del flujo de trabajo que va a utilizar con el objetivo de mejorar los niveles de seguridad ofrecidos. Más concretamente, la RP informa a los usuarios que este flujo de trabajo puede prevenir y detectar ataques de robo de credenciales y de secuestro de sesión.
- La RP cumple con las garantías de privacidad y protección de datos (Reglamento General de Protección de Datos (RGPD) y Ley Orgánica de Protección de Datos de Carácter Personal (LOPD)). Además, sigue los principios de privacidad desde el diseño minimizando la recogida de información, obteniendo el necesario consentimiento explícito e informado del usuario final, y garantizando los niveles adecuados de transparencia.

3.2. Casos de uso

En general, el flujo de trabajo propuesto a continuación puede ser implementado por cualquier RP para mejorar los niveles de seguridad cuando se utiliza un estándar de gestión de identidades federado para realizar los procesos de IAAA. Más específicamente, este flujo de trabajo permite mejorar los niveles de seguridad añadiendo nuevos mecanismos para el control de accesos utilizando los sistemas basados en riesgos, la autenticación continua y los sistemas de alerta temprana. Se han seleccionado estos tres casos de uso representativos, en los que la implantación del flujo de trabajo es de utilidad:

- Caso de uso 1: una RP decide utilizar el nivel de garantía (en inglés, *Level of Assurance* [LoA]) en la petición de autenticación. De esta manera, utilizando el LoA, una RP puede

especificar el grado de confianza que requiere al IdP para poder autenticar a un usuario (uno o dos factores de autenticación, factores biométricos o criptografía). Normalmente, el LoA se define de forma estática, sin embargo, gracias al uso del flujo de trabajo propuesto, este valor se puede fijar de forma dinámica en función de los valores devueltos por los modelos de análisis de comportamientos. Por ejemplo, un comercio electrónico en el que un usuario realiza una compra desde un navegador totalmente distinto al que suele utilizar, y sus dinámicas de comportamiento del uso del teclado y ratón son anómalas. En este caso, la RP puede solicitar al IdP un LoA mayor, de tal manera que el EU debe mostrar más autenticadores para terminar satisfactoriamente el proceso de autenticación. Este comportamiento anómalo es un signo de una posible suplantación de identidad. En caso de que el usuario sea legítimo podrá demostrarlo por medio de más autenticadores, mientras que para un atacante será más difícil o imposible.

- Caso de uso 2: una RP decide implantar un mecanismo de autenticación continua durante las sesiones de sus usuarios. En el flujo de trabajo propuesto, estos mecanismos de autenticación continua son implementados mediante modelos de análisis de comportamientos. De esta manera, el EU deberá de presentar los autenticadores necesarios, cuando la RP lo considere a lo largo de una sesión, teniendo en cuenta los comportamientos anómalos detectados por los modelos de análisis de comportamientos. Este proceso le permitirá quedar autenticado recibiendo los *tokens* pertinentes. Nuevamente, estos comportamientos anómalos pueden estar vinculados a un evento que no sea necesariamente un incidente de ciberseguridad, o por el contrario pueden ser debidos a un secuestro de sesión, el cual quedaría bloqueado gracias a esta propuesta.

- Caso de uso 3: una RP decide almacenar el histórico de comportamiento de todos los EU. En este sentido, el análisis de toda esta información se prepara para ser procesado de forma periódica (p. ej. cada noche o cada semana). Los modelos de análisis de comportamientos determinan si se han detectado comportamientos anómalos, de tal manera que se levanten alertas en caso afirmativo. De esta forma, tanto los EU como las RPs pueden tomar las acciones necesarias teniendo en cuenta estas alertas. Este procesamiento no se realiza en tiempo real, como sucede en los casos de uso 1 y 2. La gran ventaja de este caso de uso es que la no tener el requisito de procesar la información en tiempo real, los mode-

los de análisis de comportamientos tienden a ser más precisos y eficaces. Por otro lado, la desventaja es que la brecha de seguridad ya habrá sucedido y por tanto no se podrá evitar, únicamente se podrá levantar una alerta para tener constancia de su ocurrencia.

Tal y como se ha comentado en el estado del arte, la gestión de identidades no solo es aplicable a usuarios, sino también a entidades (p. ej. sensores en entornos IoT). Cabe destacar, que los casos de uso aquí propuestos también son de utilidad para esta casuística. Por ejemplo, la fase de alta o *enrollment* en sensores en una ciudad inteligente puede ser muy costosa debido a la gran escala de este tipo de proyectos. El análisis de comportamiento es de gran utilidad para solventar este caso de uso, así como sus fases posteriores de autenticación y autorización [113]. Otro caso de uso en este ámbito es la detección de intrusiones analizando el tráfico de red de las comunicaciones de los sensores [114].

3.3. Flujo de trabajo

El flujo de trabajo propuesto en esta tesis doctoral, proporciona a las RPs las líneas generales para integrar soluciones basadas en el análisis de comportamiento dentro de los esquemas de gestión de identidades federados. Este flujo está pensado para poder ser utilizado en todos los casos de uso detallados en la Sección 3.2. Estos casos de uso pueden ser previos o posteriores a la autenticación, algunos son *offline* y otros son *online*. Además, los recursos de computación o los datos recopilados, así como los consentimientos necesarios por parte de los EU son muy diversos.

En la Figura 3.1 se muestra el flujo general que han de seguir todos los agentes que interactúan en un flujo federado de gestión de identidades. Es decir, esta figura extrapola el funcionamiento de cualquier estándar de gestión de identidades federado, y el lugar donde se integran las técnicas de análisis de comportamiento y cada caso de uso. Estos pasos no están enumerados, pues se pueden realizar de forma asíncrona, es decir, los agentes pueden ir avanzando y volviendo a pasos anteriores en función del estado en el que se encuentren.

En la Figura 3.2 se detallan los pasos precisos que ha de seguir cualquier RP para integrar las técnicas de análisis de comportamientos dentro del flujo visto en la Figura 3.1. Consta de cinco pasos fundamentales: la selección de la huella digital, la generación de la huella digital,

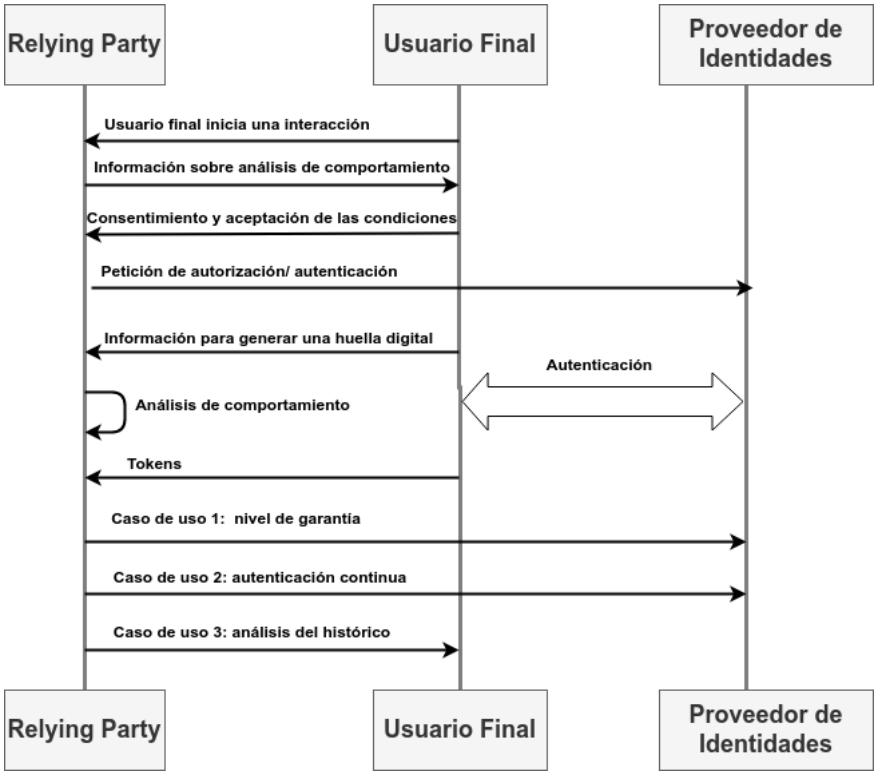


Figura 3.1: Visión global de la integración del análisis de comportamiento en un estándar federado.

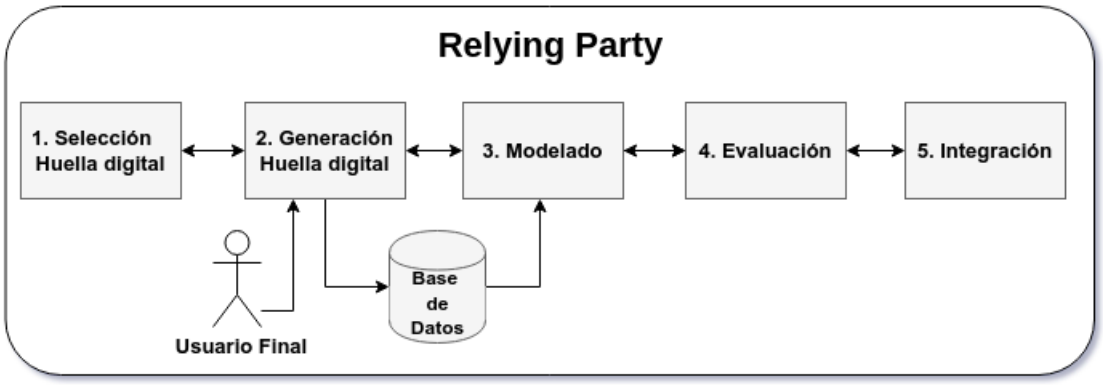


Figura 3.2: Flujo de trabajo para la integración del análisis de comportamiento en un estándar federado.

el modelado de las huellas digitales, la evaluación de los modelos generados y su integración en los flujos de información de los estándares federados. Cabe destacar, que una RP puede avanzar y retroceder en la realización de estos pasos en función de los resultados parciales obtenidos en los mismos con el objetivo de lograr una integración satisfactoria.

El primer paso es definir y seleccionar la huella digital. Para ello tiene que utilizar información de comportamiento y atributos lo suficientemente relevantes y descriptivos para que cada huella sea única. De esta forma, las huellas son distinguibles entre sí y por tanto se pueden detectar anomalías de comportamientos.

El segundo paso se centra en generar esta huella digital considerando la información y atributos previamente seleccionados. Para ello se debe utilizar la información recopilada y limpiar la información no relevante (ruido), realizar un preprocesamineto de los datos y transformarlos en variables descriptivas que puedan ser utilizados por los algoritmos de aprendizaje máquina. Posteriormente, el conocimiento extraído en forma de variables descriptivas ha de almacenarse de forma eficiente en una base de datos robusta y escalable. Además, se han de definir procesos para verificar que las huellas digitales son válidas a lo largo del tiempo, ya que estas dinámicas y pueden ir cambiando y poder así actualizarlas.

El tercer paso es modelar las huellas digitales. Para ello se va a hacer uso de los modelos de análisis de comportamiento (es decir, modelos de aprendizaje máquina) específicos que permiten realizar la detección de anomalías. Estos algoritmos se nutren del conocimiento extraído en el segundo paso, por lo que, si este segundo paso no se realiza de forma óptima, es de esperar que el resultado obtenido para el tercer paso no sea tampoco óptimo. Los modelos de análisis de comportamientos tienen que permitir comparar nuevos comportamientos con las huellas digitales generadas, con el objetivo de poder encontrar anomalías y por consiguiente detectar posibles brechas de seguridad. Para obtener buenos resultados, una RP ha de considerar una métrica de similitud o distancias entre huellas digitales precisa. Posteriormente, ha de ser capaz de fijar un umbral de decisión a partir del cual una muestra analizada se considere legítima o una anomalía. Finalmente, al igual que con la generación de huellas digitales, los modelos de aprendizaje máquina han de estar preparados para reentrenarse con el objetivo de considerar los cambios de comportamiento intrínsecos que surgen en los propios usuarios. Esto último, también se puede realizar considerando un modelo de entrenamiento online, es decir, un modelo que va entrenándose a medida que va realizando nuevas predicciones en el sistema.

El cuarto paso es evaluar los modelos de análisis de comportamientos, elaborados en el paso anterior. Este proceso permite detectar posibles errores de diseño que no se han considerado en un primer momento como, por ejemplo, que las huellas digitales seleccionadas y generadas en

los pasos previos, no son lo suficientemente descriptivas como para ser funcionales. Además, permite determinar si los modelos de aprendizaje máquina obtenidos se comportan de la manera esperada, es decir, si permiten detectar anomalías de comportamiento, son robustos y tienen capacidad de generalización. En caso de que no se comporten de la manera esperada, la RP ha de evaluar los motivos y solventarlos volviendo atrás a alguno de los pasos anteriores. Cabe destacar, que las métricas estándares de evaluación de modelos de aprendizaje máquina no son lo suficientemente descriptivas en el ámbito del análisis de comportamiento. Es por esto que se suelen utilizar métricas específicas para este dominio.

Finalmente, la quinta tarea es la integración del flujo de trabajo en los estándares de gestión de identidades federados. Este proceso depende específicamente del estándar donde se quiera integrar el flujo. En este trabajo se establecen la líneas y procedimientos que una RP ha de considerar como, por ejemplo, modificar lo menos posible los flujos de información ya determinados por los estándares y utilizar los propios mecanismos que estos estándares proveen para modificarlos en caso de ser necesario. Además, al realizar esta integración y teniendo en cuenta los aspectos relativos a la privacidad, la RP ha de informar a los usuarios de la recopilación y tratamiento de los datos que se van a realizar.

De aquí en adelante, se va a analizar cada una de estas tareas en detalle.

3.3.1. Selección de huella digital

En esta sección se proponen una serie de atributos que cualquier RP puede utilizar para generar una huella digital con el objetivo de poder completar el flujo de trabajo propuesto. Cabe destacar, que es imposible definir una huella digital general que sirva como solución a todas y cada una de las RPs. Esto se debe a que, existen multitud de RPs de naturaleza totalmente distinta (p. ej. aplicación web, aplicación móvil) y que poseen recursos totalmente diferentes y heterogéneos para contemplar multitud de casos de uso en distintos dominios. Es por esto que cada RP ha de decidir un mínimo de atributos que garantice los niveles de seguridad necesarios en función de sus necesidades.

Debe existir una solución equilibrada entre eficacia y privacidad. Esto se debe a que, cuanto mayor sea el número de atributos seleccionados, más eficaces serán los modelos de análisis de

comportamientos a la hora de detectar posibles brechas de seguridad. Sin embargo, a medida que el número de atributos aumenta, el proceso de recolección de la información será más costoso y además será más invasivo para la privacidad del usuario final. Un claro ejemplo de este equilibrio se puede encontrar en la diferencia de aumentar la seguridad de una aplicación financiera y en una red social. En el caso de la aplicación financiera, posiblemente un usuario prefiera sacrificar privacidad a cambio de obtener un nivel de seguridad más alto, no siendo así para proteger su perfil en una red social. Toda RP dispuesta a integrar este flujo de trabajo ha de asegurarse el cumplimiento de la normativa sobre privacidad y protección de datos vigente.

Todos los atributos propuestos se pueden ver categorizados en la Figura 3.3. En primer lugar, estos atributos se dividen en estáticos y dinámicos. Los atributos estáticos están compuestos mayoritariamente por información de contexto. Estos atributos se consideran estáticos, porque suelen cambiar nada o muy poco a lo largo del tiempo. Por otro lado, los atributos dinámicos representan las interacciones del usuario final con la RP. Estos atributos son muy cambiantes a lo largo del tiempo. Muchos de los atributos estáticos están definidos por un conjunto finito de posibles valores, por lo que, pueden existir muchos usuarios con valores similares o idénticos, mientras que los atributos dinámicos pueden tomar valores muy heterogéneos (p. ej. no existe un patrón único de comportamiento en las dinámicas de teclado que agrupe un conjunto amplio de usuarios). El grupo de atributos estáticos que garantiza la suficiente singularidad en las huellas digitales, es decir, sus valores presentan la suficiente diversidad como para que la huella digital sea única, son los siguientes:

- Codificación del contenido: normalmente un algoritmo de compresión que el navegador soporta.
- Idioma del contenido: preferencias de lenguaje seleccionado por el usuario en el navegador.
- Cabeceras HTTP: parámetros contenidos en las cabeceras de las peticiones HTTP.
- Lista de plugins: plugins instalados en el navegador. Por ejemplo, *Java, Flash, o Silverlight*.
- *Cookies* habilitadas: configuración de *cookies* en el navegador.

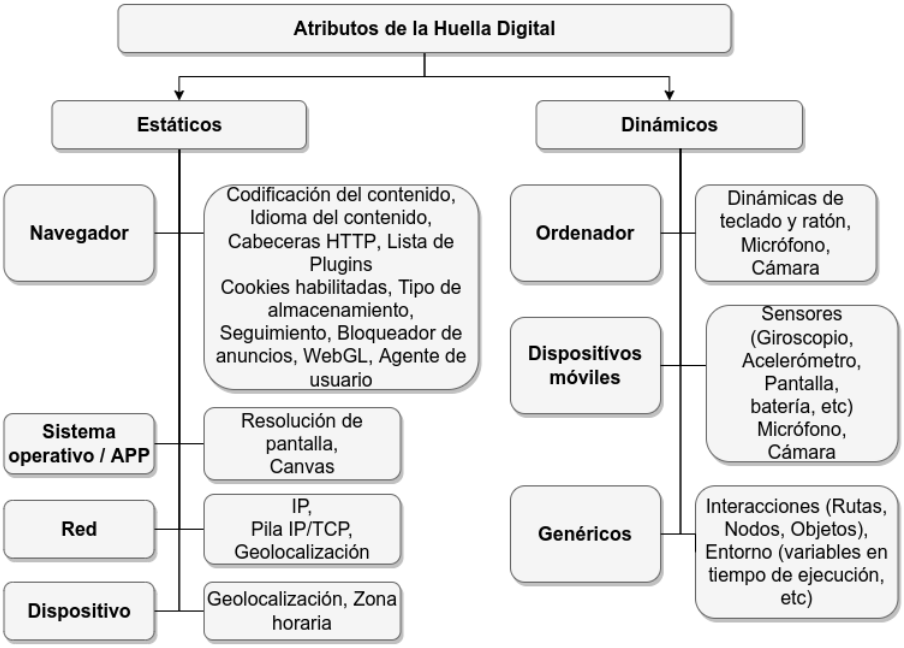


Figura 3.3: Tipos de atributos de la huella digital.

- Tipo de almacenamiento: mecanismos de almacenamiento soportados por el navegador (local, *indexedDB*, sesión, *Web-SQL*, etc.).

- Seguimiento: cabecera que se incluye cuando no se desea obtener seguimiento de ciertas páginas o aplicaciones web.

- Bloqueador de anuncios: software instalado que bloquea la publicidad agresiva o que afecta a la privacidad del usuario en el navegador.

- *WebGL*: *WebGL* es una API para renderizar los gráficos en el navegador. Esta API permite obtener diferentes atributos relacionados con el navegador instalado.

- Agente de usuario: información relacionada con el navegador y sistema operativo utilizado por el usuario.

- Resolución de pantalla, profundidad del color y densidad de píxeles: información relacionada con la tarjeta gráfica utilizada y los drivers instalados.

- Canvas: script que se utiliza para renderizar el texto y gráficos de HTML. Proporciona información del software y el hardware del usuario.

- Dirección IP: identifica la fuente de comunicación.
- TCP/IP: puertos abiertos, DNS utilizado, configuración NAT, etc.
- Geolocalización: obtenida por medio de inferencia por la red usada o por APIs específicas.
- Zona horaria: zona horaria seleccionada en la configuración del navegador.

La recomendación que se propone en esta tesis doctoral para seleccionar los atributos estáticos es la de seleccionar al menos un atributo de cada una de las categorías siempre y cuando sea posible.

Por otro lado, los atributos dinámicos están fuertemente ligados al dispositivo que se utiliza para acceder a la RP. Por ejemplo, las dinámicas de teclado, ratón o el uso de sensores de entrada como el micrófono o cámara, están muy ligados a dispositivos como un ordenador personal. Sin embargo, sensores más específicos como el giroscopio, acelerómetro o el sensor de una pantalla táctil están ligados a dispositivos como teléfonos móviles inteligente o tabletas. Cabe destacar que, algunos atributos dinámicos pueden ser generales, es decir, pueden ser recopilados independientemente del dispositivo utilizado, como las trazas de interacción con una interfaz gráfica, los nodos visitados y rutas seguidas para realizar una tarea concreta, etc.

A continuación, se definen una serie de perfiles para que las RPs puedan tener unas bases para seleccionar unos atributos u otros. Estos perfiles se basan en estudios del estado del arte que tratan de analizar la entropía o singularidad que proporcionan cada uno de estos atributos a la hora de generar una huella digital, así como la facilidad de recopilación de los mismos, y la posible aceptación que un usuario puede tener a la hora de permitir recopilarlos (debido a la invasión de la privacidad que puede suponer recopilar ciertos atributos). Para los atributos estáticos cabe destacar los trabajos [115], [116], mientras que para los atributos dinámicos el estudio realizado en [117], [118]. Estos perfiles distinguen, en primer lugar, por RPs accesibles desde web o desde un entorno móvil, y en segundo lugar se categorizan en tres niveles de seguridad (básico, medio y alto). Estos niveles de seguridad son incrementales, es decir, los niveles superiores incluyen los atributos seleccionados para los niveles inferiores. A continuación, se detallan los atributos específicos que se incluyen en cada uno de los niveles:

- Seguridad básica para una RP web: agente de usuario, lista de plugins, resolución de pantalla, zona horaria y habilitación de *cookies*.
- Seguridad básica para una RP en entorno móvil: Resolución de pantalla, zona horaria, utilización de batería y utilización del sistema.
- Seguridad media para una RP web: Canvas, WebGL y entorno.
- Seguridad media para una RP en entorno móvil: Canvas, giroscopio, acelerómetro y entorno.
- Seguridad alta para una RP web: dinámicas de ratón, dinámicas de teclado, interacciones con el entorno y geolocalización.
- Seguridad alta para una RP en entorno móvil: dinámicas de pulsación en pantalla, interacciones con el entorno y geolocalización.

Como se ha mencionado con anterioridad, estos perfiles son una recomendación, es decir, no son obligatorios y por consiguiente cualquier RP tiene la capacidad de partir de cualquiera de estos perfiles e ir añadiendo o eliminando atributos específicos para cumplir con sus requisitos y necesidades. Si una RP decide eliminar o no puede incluir alguno de los atributos específicos, para conseguir el mismo nivel de seguridad se recomienda seleccionar un atributo de un nivel de seguridad inmediatamente superior. Si esta casuística sucediese para el nivel más alto, la recomendación es seleccionar al menos dos atributos del nivel de seguridad inmediatamente inferior.

3.3.2. Generación de la huella digital

La información recopilada durante este paso ha de ser común a todos los usuarios y amplia en cuanto a volumen para cada usuario. Esto se debe a que, las huellas digitales han de poder ser comparadas entre los diferentes usuarios, por lo que, tienen que poseer las mismas características. Además, cuanta más información se recopile para cada usuario, se podrá determinar con mayor facilidad las fronteras de decisión y los patrones que definen a cada usuario de forma única. La tecnología utilizada ha de ser genérica y multiplataforma con el objetivo

de que sea adaptativa. Los procesos de recopilación y generación de la información han de ser eficientes, evitando introducir latencias innecesarias y un consumo de recursos excesivos. La recopilación de los datos se puede llevar a cabo por lotes, es decir, recibiendo la información de forma periódica, o en tiempo real y de forma continua en el tiempo, es decir, en *streaming*.

Una vez se ha recopilado la información, el siguiente paso es el proceso de almacenamiento de la información. Este proceso tiene que hacer uso de un sistema fácilmente escalable, con el objetivo de que pueda aumentar sus capacidades en el caso de que el número de usuarios o de información recopilada sea masivo.

El procesado de los datos también debe estar claramente definido. Para ello, se deben descartar valores atípicos, información con ruido o información obsoleta, la cual puede hacer que los modelos generados se vean lastrados.

Es indispensable determinar como la información recopilada y posteriormente almacenada se va a representar. Por ejemplo, las cadenas de Markov son de utilidad para transformar a estados cada comportamiento, acción o interacción del usuario. En este caso concreto, se analizarán las transiciones entre estados, determinando que, si el usuario se ha desplazado de un estado en el que es altamente probable que se encuentre, a un estado poco probable, el comportamiento realizado es atípico y por consiguiente se considera una anomalía y una posible brecha de seguridad. Otras soluciones tratan cada interacción del usuario como un conjunto de características que lo definen, por ejemplo, velocidades de movimiento o ángulo de desplazamiento en el caso de considerar dinámicas de movimientos de ratón. Estas características se agrupan temporalmente formando un vector que define el comportamiento de un usuario en un intervalo de tiempo.

En este trabajo se recomienda que una RP primero seleccione una solución simple que se adapte correctamente a sus necesidades. Esta solución puede basarse en generar secuencias de vectores, que se pueden agrupar posteriormente utilizando ventanas temporales deslizantes, de tal manera que los vectores consecutivos contengan también información de los vectores adyacentes. De esta forma, los vectores consideran una mayor cantidad de información de seguido. Por ejemplo, dado un vector que contenga cuatro elementos, estos se pueden separar en subvectores de longitud dos tal que, el primer subvector contiene los elementos uno y dos, el siguiente subvector los elementos dos y tres y así sucesivamente.

3.3.3. Modelado

En la parte de modelado, las técnicas de aprendizaje máquina basadas en detección de anomalías se posicionan como una de las mejores soluciones para detectar posibles brechas de seguridad utilizando información de comportamientos. Estas técnicas se denominan, en el ámbito del aprendizaje máquina, detección de atípicos. Además, estas técnicas permiten, entre otras cosas, que las RPs puedan manejar datos no etiquetados (aprendizaje no supervisado) y desequilibrados, los cuales son dos de los grandes problemáticas en este ámbito.

La detección de atípicos se basa fundamentalmente en establecer una medida de distancias o similitud (p. ej. la distancia Euclídea) entre objetos [85]. Para el caso concreto que aplica a esta tesis doctoral, estos objetos son huellas digitales de comportamiento. De esta forma, gracias a la medida de distancia o similitud, se pueden definir núcleos de comportamiento que se consideran normales o esperados para un usuario concreto teniendo en cuenta su huella digital, es decir, teniendo en cuenta su histórico de comportamientos. Posteriormente, las nuevas interacciones que llegan al sistema generan nuevas trazas de comportamientos que se comparan con el comportamiento esperado, es decir, con la huella digital. Definiendo un umbral, se puede determinar si estos nuevos comportamientos son normales, o si por el contrario son atípicos y por lo tanto pueden suponer una brecha de seguridad.

Los algoritmos de aprendizaje máquina que pueden ser útiles para una RP con el objetivo de poder seguir el flujo de trabajo propuesto se pueden clasificar en tres grupos [119]:

- *Forecasting*: se basan en el aprendizaje supervisado. En este grupo, los algoritmos se entrenan utilizando datos etiquetados. Estos algoritmos tratan de realizar predicciones basándose en el análisis de tendencia, de estacionalidad y ciclos de datos temporales. ARIMA [120], SVM, K-NN, NN y las redes bayesianas son algunos ejemplos de algoritmos que encajan en este ámbito.
- No supervisado: se basan mayoritariamente en agrupar los datos. De esta forma, se toma como premisa que los datos generados por la misma fuente (es decir, generados por el mismo usuario) pertenecerán a los mismos grupos, mientras que los datos atípicos se encontrarán fuera de estos grupos definidos como normales. OC-SVM, K-means e *isolation forest* [121] son algunos ejemplos de algoritmos que encajan en este grupo.

- Basados en densidades: este grupo es en realidad una subcategoría de los no supervisados. Se basa en determinar la densidad de datos que contienen las distintas regiones del espacio. De esta forma, las regiones de alta densidad se pueden definir como las regiones normales o esperadas. Por otro lado, si una muestra se encuentra en una región de baja densidad, será considerado como atípica y, por consiguiente, se determina que no pertenece al usuario legitimo. Algunos ejemplos de estos algoritmos son *Local Outlier Factor* [122] o DBSCAN [123].

Una RP específica que desee implementar el flujo de trabajo propuesto, ha de primero analizar los datos recopilados y almacenados para generar la huella digital. Esta huella digital, en función de la naturaleza de la RP específica, poseerá unas cualidades únicas. Por ejemplo, una RP que recopile información de la sesión del usuario, podrá tener datos etiquetados y por lo tanto utilizar modelos de *forecasting* para realizar las predicciones o para mejorar las predicciones realizadas por modelos no supervisados. Cabe destacar, que la información de las etiquetas también puede estar sesgada, es decir, si durante el proceso de recopilación de datos ya existía una brecha de seguridad, los datos recopilados pueden contener comportamientos atípicos. Esto significa, que estos datos atípicos serán considerados como normales o esperados si no se realiza un proceso de preprocesamiento y limpieza de datos adecuado. La recomendación que se propone en esta tesis doctoral es utilizar las etiquetas siempre y cuando sea posible, pero teniendo en cuenta el propio sesgo que puede existir en dichos datos y, por consiguiente, combinar modelos supervisados con modelos no supervisados o basados en densidades para obtener predicciones más precisas. Por ejemplo, una buena aproximación de combinación de varios tipos de algoritmos se encuentra en [124], donde primero se aplican técnicas de *clustering* para agrupar los diferentes comportamientos para posteriormente aplicar múltiples algoritmos específicos que modelen cada grupo de forma independiente, obteniendo resultados más precisos para cada uno de ellos.

Todos los tipos de algoritmos, arriba mencionados, se pueden implementar de dos formas diferentes, *offline* y *online*. Los algoritmos *offline* utilizan el histórico de información para generar un modelo estático que posteriormente se utiliza para realizar predicciones durante un periodo de tiempo. Estos modelos han de irse adaptando para poder considerar los cambios de la distribución de los datos de entrada, es decir, los usuarios no mantienen su comportamiento

estable a lo largo del tiempo. Para poder considerar estos cambios de comportamiento, estos modelos han de irse reentrenando. Para lograr que el reentrenamiento sea efectivo, se deben establecer unas políticas que contemplen indicadores para determinar cuándo las distribuciones iniciales de los datos han cambiado y por lo tanto el modelo ha quedado obsoleto y está perdiendo eficacia y precisión.

Por otro lado, los algoritmos *online* van entrenando a medida que van realizando nuevas predicciones. De esta forma, cuando llegan nuevas muestras para evaluar, el propio modelo se va actualizando y cambiando sus fronteras de decisión para considerar estos nuevos comportamientos. Así, a medida que el modelo obtiene nuevas muestras, su eficacia y precisión va aumentando. Sin embargo, también hay que considerar que estos modelos también han de ir descartando la información obsoleta que puede quedar cuando el modelo lleva funcionando durante un largo periodo de tiempo. Este tipo de modelos son de mucha utilidad cuando se combinan con una recopilación de la información en *streaming*.

Una de las problemáticas más comunes a las que se enfrentan las propuestas de este ámbito, es la de detectar anomalías para los primeros accesos, es decir, cuando apenas se posee información de comportamiento del usuario. Esta problemática se denomina arranque en frío. Para este caso concreto, el resultado suelen ser modelos de aprendizaje máquina muy pobres, que no poseen suficiente información para comparar las muestras a evaluar y, por consiguiente, categorizan multitud de muestras como anomalías, cuando no lo son. Esto se traduce, en modelos con una tasa de falsos positivos muy elevada, haciendo que los sistemas en los que se integran estos modelos no sean muy fiables. Una buena solución para este problema es utilizar un sistema de recomendación, capaz de detectar cuando un usuario está utilizando el sistema por primera vez, y alimentar una máquina de factorización para tratar de asemejar estos comportamientos de los de otros usuarios parecidos, mitigando así el problema [125]. Otras posibles soluciones se basan en utilizar modelos no paramétricos para la fase inicial en la que puede existir arranque en frío y otros modelos más precisos cuando ya se ha superado esta fase [126], o tomar como punto de partida un modelo previamente entrenado para otro usuario. De esta forma, aunque siga existiendo el arranque en frío, este durará un menor tiempo, pues el modelo no parte de cero y solo tendrá que irse adaptando, modificando sus fronteras de decisión, a los nuevos comportamientos que genere el nuevo usuario.

		Predicción	
		Positivo	Negativo
Valor Real	Positivo	VP	FP
	Negativo	FN	VN

Figura 3.4: Matriz de confusión.

3.3.4. Evaluación

El paso de evaluación permite cuantificar la eficacia de los modelos de aprendizaje máquina desarrollados en el paso anterior. De esta forma, se pueden encontrar errores de diseño, ya sea a la hora de seleccionar o generar la huella digital, como determinar los motivos que hacen que el modelo en sí no esté funcionando correctamente. Esta detección temprana permite que la RP pueda regresar a los pasos anteriores, con el objetivo de mejorar la eficacia del modelo final. Es por esto que este paso es fundamental, pues se posiciona como la capa intermedia entre el desarrollo del flujo de trabajo y su puesta en producción.

En primer lugar, se definen las clases que se quieren evaluar. Para el caso del análisis de comportamiento, típicamente se consideran dos clases, la positiva que representa al usuario genuino, y la clase negativa que representa al usuario impostor. Dadas estas dos clases, un modelo de aprendizaje máquina puede predecir de forma correcta o incorrecta cada una de estas clases, obteniendo así cuatro posibles valores. En primer lugar, el valor Verdaderos Positivos (VP) representa a los usuarios genuinos autenticados correctamente. Verdaderos Negativos (VN) representa a los impostores rechazados por el modelo de forma correcta. Falsos Positivos (FP) son los usuarios impostores que se han logrado autenticar como usuarios genuinos y no han sido detectados por el modelo. Falsos Negativos (FN) son los usuarios genuinos que se han detectado incorrectamente como impostores por el modelo. Estos cuatro valores, se agrupan formando una matriz de confusión tal y como se muestra a continuación (ver Figura 3.4):

Existen multitud de métricas para evaluar diferentes aspectos específicos de los resultados obtenidos para los algoritmos de aprendizaje máquina. Dada esta matriz de confusión, típicamente se suelen calcular tres métricas asociadas: la exactitud, la precisión y la sensibilidad. La exactitud es la tasa de acierto sobre el total de muestras, es decir, el porcentaje de acierto del

modelo de manera global. La precisión es el porcentaje de aciertos para la clase positiva. La sensibilidad es el porcentaje de muestras para la clase positiva que se aciertan de forma correcta sobre el total de posibles positivos. Estas métricas vienen definidas como sigue:

- Exactitud= (VP+VN)/(VP+FP+FN+VN)
- Precisión= VP/(VP+FP)
- Sensibilidad= VP/(VP+FN)

Sin embargo, en el ámbito del modelado de comportamiento, estas métricas no suelen ser suficiente, pues los problemas de este tipo poseen unas cualidades peculiares, como los datos desequilibrados. Por ejemplo, en el caso de considerar una clase mayoritaria que posee el 99 % de la información genuina, es decir, de información de clase positiva, un modelo trivial que clasifique cada muestra en esta clase obtendría un 99 % de tasa de acierto o exactitud. Si una RP únicamente tuviese en cuenta esta métrica, seguramente este modelo sería seleccionado, pues tiene un acierto muy alto. Sin embargo, el objetivo del problema es justo detectar la clase minoritaria, es decir, la clase que contiene los comportamientos realizados por usuarios impostores (clase negativa) y que pueden suponer una brecha de seguridad. En este caso, este modelo trivial tendría una tasa de acierto del 0 %, convirtiendo así a este modelo en inútil y por tanto habría que desecharlo.

Una métrica que ha demostrado ser más efectiva que las anteriores para lidiar con datos desequilibrados es el F1-score [127]. Esta métrica se puede definir como la media armónica entre la precisión y la sensibilidad. Nuevamente, esta métrica considera como clase de interés la clase genuina.

Todas las métricas definidas hasta ahora se basan fundamentalmente en evaluar la clase positiva, es decir, la de usuarios genuinos. Para el problema que se desea resolver en esta tesis, debería ser al contrario, es decir, se debe evaluar la clase de usuarios impostores, pues es la clase de interés.

Existen métricas específicas que detallan con más precisión las casuísticas específicas en el ámbito del análisis de comportamiento. Estas métricas son: tasa de aceptación falsa (en inglés, *False Acceptance Rate* [FAR]), tasa de rechazo falso (en inglés, *False Rejection Rate* [FRR])

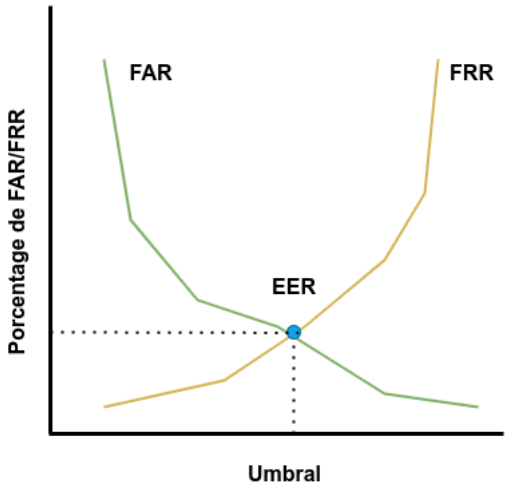


Figura 3.5: Representación del Equal Error rate (EER) en función del False Acceptance Rate (FAR) y el False Rejection Rate (FRR).

y la tasa de error igual (en inglés, *Equal Error Rate* [EER]) [128]. El FAR se define como la proporción de usuarios impostores que no son detectados correctamente por el modelo. El FRR es la proporción de usuarios legítimos que el modelo predice como usuarios impostores. Un mismo modelo puede ser más restrictivo o más permisivo en función del umbral definido. Esto es, un mismo modelo puede determinar la probabilidad de una muestra de pertenecer a la clase genuina o a la clase impostora, pero es gracias al umbral cuando finalmente se clasifica cada muestra. De esta forma, un mismo modelo con un umbral alto puede ser más restrictivo y por lo tanto tener más acierto para la clase impostora pero menos para la clase genuina (sistema más seguro), o puede ser más permisivo fijando un umbral bajo y por consiguiente tener más acierto en la clase genuina pero menos en la impostora (nunca se podría fijar un umbral que haga tener más acierto en ambas clases simultáneamente). De este modo, dependiendo del umbral se obtienen una pareja de valores FAR y FRR. El EER se puede definir, por tanto, como el punto óptimo entre estos dos valores, es decir, el umbral donde se obtiene un mejor valor para ambas métricas de forma simultánea (ver Figura 3.5).

Las métricas utilizadas en este documento para evaluar los modelos y que se recomienda que utilice cualquier RP que desee integrar el flujo de trabajo propuesto, se definen a continuación.

- Especificidad= VN/(VN+FP)
- FAR= FP/(FP + VN)

- FRR= FN/(FN + VP)
- EER= Valor de la intersección entre FAR y FRR cuando se consideran múltiples umbrales.
- Valor Predictivo Negativo (VPN)= VN/(VN+FN)
- $F1^{-} = 2 \cdot \frac{\text{VPN} \cdot \text{Especificidad}}{\text{VPN} + \text{Especificidad}}$

Donde la especificidad es la proporción de impostores correctamente identificados, esto es, la sensibilidad para el grupo de impostores (clase negativa). VPN es la eficacia del método a la hora de detectar impostores, esto es, la precisión para el grupo de los impostores. $F1^{-}$ es la media harmónica entre el VPN y la especificidad [129], es decir, el F1-Score para el grupo de los impostores.

3.3.5. Integración en los sistemas de gestión de identidades federados

La integración del flujo de trabajo en los esquemas de gestión de identidades federados acarrea multitud de retos.

En primer lugar, los usuarios finales seguramente no estén dispuestos a tener que instalar software externo, especialmente si este va a afectar a su privacidad. Este flujo de trabajo es claramente un ejemplo en el que los usuarios pueden sacrificar cierta privacidad (dependiendo de la huella digital seleccionada) con el objetivo de ganar seguridad. Por ejemplo, un usuario puede ser reacio a que una red social recopile información de su comportamiento, sin embargo, si su aplicación de banca electrónica se lo solicita puede optar por aceptar las condiciones.

El flujo de trabajo aquí expuesto no requiere que el usuario final tenga que instalar software externo, simplemente necesita que el usuario acepte las condiciones y por consiguiente consienta que la RP recopile información para generar su huella digital de comportamiento. Es por esto que un mismo usuario puede optar por permitir que ciertas RPs recopilen más información que otras, o incluso ninguna, con el objetivo de que el propio usuario final se vea beneficiado o no, de la implantación del flujo de trabajo propuesto. En cuanto a la privacidad, la RP ha de informar de forma transparente de todos y cada uno de los datos que va a recopilar, así como de cuál va a ser el uso que le va a dar. Cabe destacar, que esta información solo ha de ser accesible

por la propia RP, es decir, no se tiene que compartir con terceros, y ha de ser debidamente protegida, desde que se genera en el dispositivo del usuario, mientras se envía por algún canal de comunicación debidamente encriptado utilizando *Transport Layer Security* (TLS), hasta que se almacena en algún sistema de información.

Por otro lado, la aceptación de los usuarios también se va a ver afectada dependiendo de la eficiencia y la usabilidad. La solución finalmente implementada ha de estar basada en una huella digital lo suficientemente única y robusta como para poder detectar las brechas de seguridad, sin afectar altamente al consumo de recursos, evitando introducir latencias innecesarias y disminuyendo el número de falsos positivos que pueden hacer que el sistema se vuelva poco usable.

Los IdPs no se deben ver afectados por la implantación de este flujo de trabajo. Todos los aspectos relacionados con la integración (consumo de recursos, notificación a los EU, etc) han de recaer sobre la propia RP, la cual es la interesada en proporcionar a sus usuarios finales la capacidad de aumentar sus niveles de seguridad. Es por esto que la implantación de este flujo de trabajo solo dependerá de la comunicación entre la RP y el EU, no necesitando de la colaboración de terceros o del IdP.

Otro reto asociado a la integración del flujo de trabajo es la aceptación de las propias RPs. Esto se debe a que, si el flujo de trabajo requiere modificar los propios estándares de gestión de identidades existentes, probablemente se van a volver reacias a implementarlos. Esto se debe a la dificultad añadida de implantación que esto supondría. Es por esto que los flujos de información de los estándares federados no han de ser modificados, o en su defecto lo menos posible y, por lo tanto, las RPs deben de utilizar los propios mecanismos facilitados por dichos estándares para realizar su cometido. De este modo, se deben utilizar las propias peticiones y respuestas, los parámetros, *tokens* y cualquier otro aspecto ya existente en los flujos de información para realizar la implantación del flujo de trabajo.

3.3.6. Resumen del flujo de trabajo

A modo resumen, en la Tabla 3.1 se muestra de forma gráfica, todas las decisiones y consideraciones que ha de tomar una RP para implantar el flujo de trabajo propuesto. Estas decisiones

Tarea	Decisión	Alternativas	Criterio de decisión
1. Selección de huella digital	Huella digital que proporciona la suficiente singularidad y no es invasiva para el usuario	Atributos estáticos y dinámicos (Figure 3.3)	Caso de uso. Niveles de seguridad deseados.
2. Generación de la huella digital	Recopilar, almacenar, preprocesar y representar los datos: tecnologías y procedimientos	Por lotes o streaming; SQL o No-SQL; lenguajes de programación; Técnicas de representación de la información	Eficiencia y eficacia, Consumo de recursos, escalabilidad
3. Modelado	Detectar anomalías en las huellas digitales: técnicas	*Forecasting*, aprendizaje no supervisado, algoritmos basados en densidades; modelos *offline* o *online*	Huella digital seleccionada, caso de uso, recursos computacionales disponibles en la RP
4. Evaluación	Decidir si se necesitan más iteraciones en el flujo de trabajo: métricas de evaluación	Exactitud, especificidad, FAR FRR, EER, VPN y $F1^{-}$	Caso de uso
5. Integración	La RP debe integrar los modelos de análisis de comportamiento para realizar los procesos de IAAA: modificaciones	Cambios en el estándar (peticiones, *tokens*, flujos); Cambios en la implementación (comprobaciones adicionales, estructura de datos)	Caso de uso, coste permitido

Tabla 3.1: Resumen del flujo de trabajo propuesto

se corresponden con cada uno de los pasos fundamentales en los que se basa dicho flujo de trabajo.

Capítulo 4

Método de combinación de información de comportamientos

Hasta el momento se ha presentado un flujo de trabajo que permite incorporar técnicas de análisis de comportamientos dentro de los flujos de autenticación y autorización de los estándares de gestión de identidades federados. Este flujo de trabajo tiene como objetivo utilizar las técnicas de análisis de comportamientos para aumentar los niveles de seguridad proporcionados. Sin embargo, los modelos de análisis de comportamientos de la literatura poseen ciertas limitaciones.

En este capítulo se detalla el método propuesto para mejorar los modelos de análisis de comportamientos actuales. Este método se basa en combinar información de comportamientos a nivel de características. Tal y como se ha visto reflejado en el estado del arte (ver Capítulo 2.2.2), la combinación de información está posicionándose como una técnica novedosa que permite mejorar la eficacia de los mismos. Sin embargo, hoy en día, son pocos los trabajos que combinan la información de comportamiento, y muchos menos los que lo hacen a nivel de características.

El método propuesto se presenta como un conjunto de tareas novedosas que permiten combinar la información de comportamientos a nivel de características y mejorar, por tanto, la eficacia de los sistemas de autenticación basados en el análisis de comportamientos [130]. Ha sido espe-

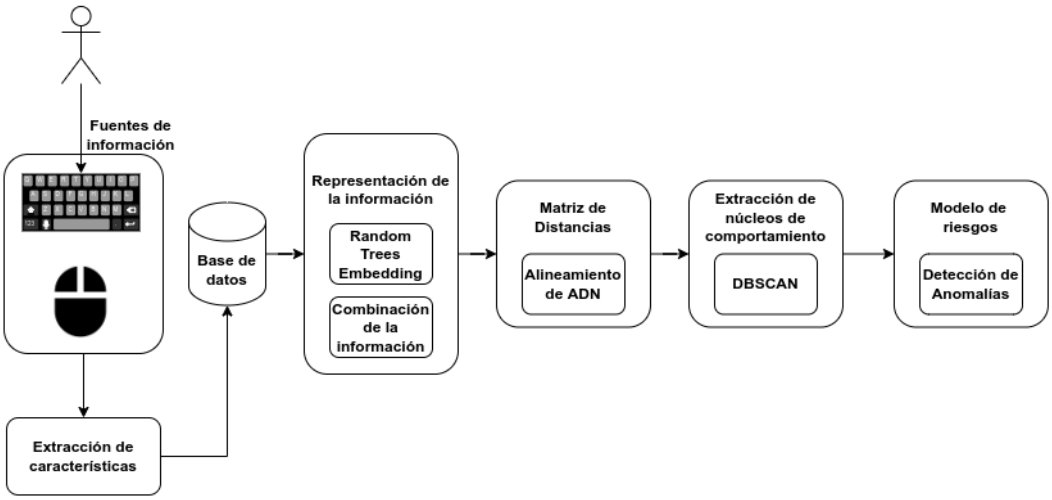


Figura 4.1: Método propuesto para combinar información de comportamientos.

cíficamente diseñado para combinar información temporal, recogida de fuentes de información heterogéneas. Además, no asume que los datos temporales siguen una distribución específica. Esto es, no asume que los datos temporales se recogen con una frecuencia o patrón específico.

El método se presenta esquematizado en la Figura 4.1. Está compuesto de cuatro tareas principales. La primera tarea se basa en representar la información recopilada de fuentes heterogéneas de datos en el mismo espacio. Para ello, se transforma en secuencias de n-gramas. Esta tarea se describe en la Sección 4.1 y se basa en la implementación de una técnica de *Symbolic Aggregate approximation* (SAX) multivariante [131] novedosa utilizando *Random Trees Embedding*s (RTEs) [132], [133]. La siguiente tarea se basa en construir una matriz de distancias comparando los n-gramas extraídos anteriormente utilizando técnicas de alineamiento de secuencias de ADN. Esta tarea se detalla en la Sección 4.2. Posteriormente, se describe el proceso de entrenamiento de un algoritmo de *clustering* basado en densidades con el objetivo de poder extraer los núcleos de comportamiento que caracterizan a cada usuario utilizando la matriz de distancias. Este proceso se contempla en la Sección 4.3. A continuación, se detalla la implementación de un modelo de riesgos, el cual utiliza los núcleos de comportamiento para categorizar las muestras, en la Sección 4.4 . Finalmente, en la Sección 4.5 se detalla cómo se pueden fijar los diferentes parámetros e hiperparámetros que utiliza el modelo con el objetivo de ser reproducible.

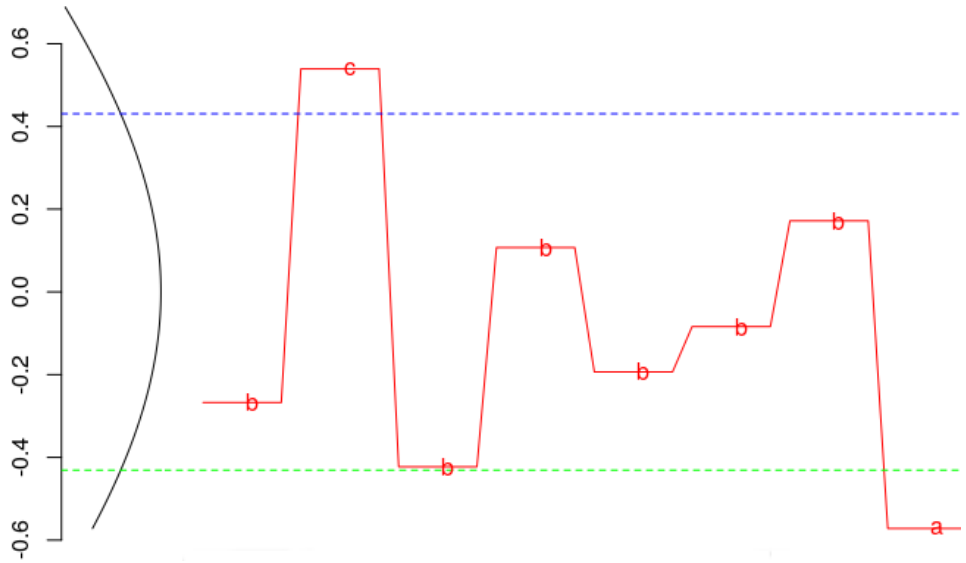


Figura 4.2: Discretización de una serie temporal usando SAX.

4.1. Representación de la información

En primer lugar, se presenta una técnica para transformar datos temporales procedentes de fuentes heterogéneas de datos (p. ej. el teclado y el ratón) en una representación tabular adecuada, que sirve para poder alimentar los modelos tradicionales de aprendizaje máquina. Este enfoque se basa en implementar una técnica novedosa de SAX multivariante [131]. Específicamente, esta técnica se implementa haciendo uso de los RTEs [132], [133].

La técnica de SAX tradicional se define como una técnica para discretizar series temporales y reducir la dimensionalidad de las mismas. Para lograr esto, se hace uso de *Piecewise Aggregate Approximation* (PAA) para normalizar y dividir las series temporales en secciones con un tamaño equidistante. Posteriormente, se calculan los valores medios de cada sección y se les asigna un nivel de probabilidad de la función gaussiana. Este proceso permite discretizar la serie. Finalmente, a cada uno de estos niveles se les asigna un símbolo concreto, logrando así transformar una serie temporal en una secuencia de símbolos (ver Figura 4.2).

Tomando como punto de partida y objetivos los mismos que la técnica de SAX tradicional, se propone extender esta implementación para considerar series temporales multivariantes. Para ello se propone utilizar el algoritmo de RTE, ya que las técnicas basadas en árboles han demostrado ser más eficaces a la hora de representar y agrupar series temporales multivariantes

[134].

El algoritmo de RTE se basa en generar una secuencia de árboles aleatorios de decisión, de tal manera que al clasificar una muestra se genera un vector con los valores de los nodos hojas que le han sido asignados. La longitud de esta secuencia de árboles viene definida por el parámetro *Número de árboles*, mientras que también se tiene que definir la profundidad de cada uno de estos árboles con el parámetro *Máxima profundidad*. Dicho de otra forma, los RTEs se basan en obtener un vector que indica, para cada árbol generado, el número de hoja a la que una muestra concreta pertenece. Esto es, la posición n del vector embedding resultante, indica el número de hoja (ordenando los árboles y, por consiguiente, las hojas de izquierda a derecha) donde la observación ha sido clasificada en el árbol de decisión n. Alternativamente, este vector embedding se suele representar de forma binaria, utilizando un vector cuya longitud es el número de hojas en el RTE y cuyo valor es 1 en caso de que la muestra haya caído en dicha hoja y 0 para el caso contrario.

Los embeddings obtenidos de los RTEs se mapean entonces a símbolos con el objetivo de habilitar la comparación entre ellos y poder procesarlos. El número de símbolos necesarios para poder mapear todos los embeddings crece de forma exponencial a medida que aumenta el *Número de árboles* y el número de hojas (en función del parámetro *Máxima profundidad*) del RTE. A pesar de esto, la mayoría de los símbolos aparecerán en muy rara ocasión, es decir, si el número de árboles generados y hojas es extremadamente alto, la mayoría de las muestras caerán en hojas distintas y, por consiguiente, se les asignara un símbolo diferente. Esta limitación se soluciona asignando los símbolos a una tabla de frecuencias de aparición. De esta manera, a las muestras mayoritarias que compartan las mismas hojas se les asigna un símbolo representativo, mientras que las muestras minoritarias se les asigna un símbolo arbitrario (algo parecido en el SAX tradicional al utilizar la función gaussiana). Por ejemplo, el embedding con más ocurrencias se le asigna el carácter *A*, al siguiente el carácter *B*, y así sucesivamente. De este modo, si dos muestras caen en las mismas hojas, son asignadas con el mismo símbolo. Cuando todos los símbolos son utilizados, los embeddings que quedan y que agrupan a las muestras menos representadas en el conjunto de datos se les asigna un símbolo arbitrario.

El proceso explicado anteriormente se ejecuta para cada fuente de información disponible. Esto es, cada fuente de información posee su propio RTE y su único conjunto de símbolos.

En este caso, la información extraída del teclado se representa utilizando símbolos de letras mayúsculas, mientras que para la información extraída del ratón se va a representar utilizando símbolos de letras minúsculas. Una vez que cada muestra de cada fuente de información se ha convertido a un símbolo, se utiliza la información temporal para poder ordenarlos. De este modo, estos símbolos se agrupan formando una secuencia ordenada que representa el comportamiento de un usuario específico para ambas fuentes de información. Finalmente, estas secuencias se dividen en n-gramas definidos por el parámetro *N-Gram Length (NGL)*. Estos n-gramas representan el comportamiento de un usuario durante un tiempo limitado de interacciones. Por ejemplo, el n-grama *AbB*, cuyo *NGL* viene definido por el valor 3, podría representar una pulsación lenta de teclado, seguido por un movimiento rápido de ratón y finalmente una pulsación rápida de teclado. Cuanto mayor sea el conjunto de símbolos, mayor precisión se obtendrá a la hora de representar el comportamiento de un usuario, sin embargo, si el número de símbolos es muy elevado, se obtiene un sobreajuste, pues cada símbolo representará una interacción muy específica y, por lo tanto, cada n-grama será diferente al resto de n-gramas haciendo así que la comparación entre ellos sea ineficaz. El proceso de combinación de la información de fuentes de datos heterogéneas se puede ver ilustrado en la Figura 4.3. Este proceso logra transformar la información de comportamientos recogida de fuentes de datos heterogéneas en el mismo espacio de representación (símbolos). De este modo, la representación de la información elegida permite combinar información a nivel de características.

4.2. Generación de la matriz de distancias

Una vez obtenida la representación de la información en n-gramas, la siguiente tarea es establecer un método para poder comparar estas secuencias entre sí. Existen multitud de métricas de distancias entre secuencias de símbolos en la literatura, siendo algunos ejemplos de las más conocidas, la distancia de Hamming, la distancia de Levenshtein, la distancia coseno y el coeficiente de Jaccard [135]. En el método propuesto, la distancia entre dos n-gramas se calcula utilizando técnicas de alineamiento de secuencias de ADN [136].

En el ámbito de la bioinformática, las secuencias de ADN se representan como secuencias de símbolos. Analizar la similitud o disimilitud entre las diferentes regiones del ADN permite detectar, entre otras cosas, multitud de enfermedades o variaciones genéticas que pueden ser

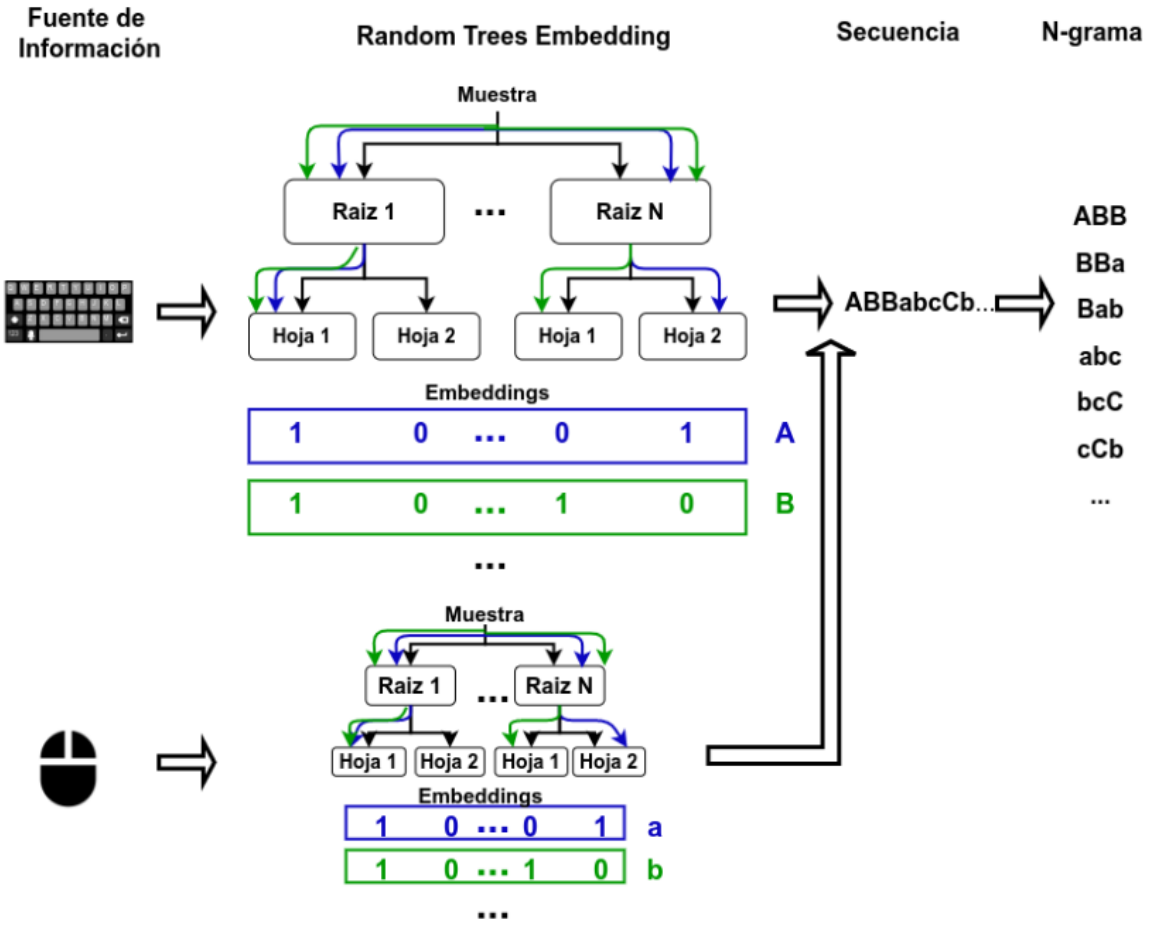


Figura 4.3: Proceso de SAX multivariante utilizando RTEs para multiples fuentes de información.

de interés para su análisis. Las técnicas de alineamiento de secuencias de ADN se posicionan como el método más utilizado para extraer patrones y comparar dichas secuencias [137]. Es por esta razón, por la que se ha decidido incorporar este tipo de técnicas en el método propuesto.

A nivel general, existen dos técnicas de alineamiento de ADN [138]: la global y la local (ver Figura 4.4).

El algoritmo de alineamiento global se basa en encontrar la mejor sincronía entre dos secuencias haciendo coincidir todos los caracteres de ambas secuencias. Es adecuado cuando las dos secuencias a comparar tienen una longitud igual o similar y se espera que compartan similitudes a lo largo de toda la secuencia. Por otro lado, el algoritmo de alineamiento local se basa en la obtención de la subcadena de la primera secuencia que más se acerca a otra subcadena de la segunda secuencia. Es adecuado para comparar secuencias de diferentes longitudes y

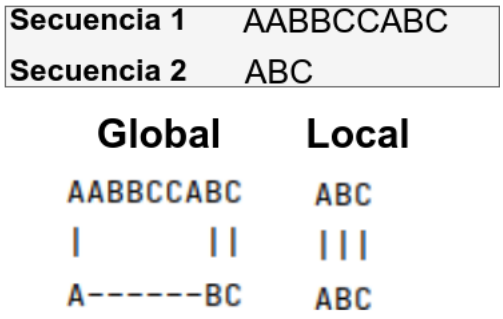


Figura 4.4: Alineamiento global y local de secuencias de ADN.

secuencias menos similares o relacionadas.

Cualquiera de estos algoritmos devuelve dos valores. En primer lugar, el alineamiento como tal, es decir, la cadena o subcadena de mayor coincidencia entre dos secuencias. En el caso de la presente investigación, este alineamiento puede ser utilizado para tratar de explicar los diferentes patrones de comportamiento que caracterizan a cada uno de los usuarios. El segundo resultado, es el valor de similitud entre ambas secuencias. Para obtener este resultado, normalmente se debe fijar el valor por el que se va a ponderar obtener una coincidencia, obtener un error y la penalización por extender el error a lo largo de una ventana. Cabe destacar que, la ponderación de coincidencia debe ser positiva para aumentar la puntuación de similitud final, mientras que las puntuaciones de error y ventana de error deben ser negativas para disminuir la similitud final entre las secuencias.

Atendiendo a lo expuesto con anterioridad, en el método propuesto se utiliza la técnica de alineamiento global [139]. De este modo, todos los n-gramas obtenidos para cada usuario se comparan en pares entre sí. El alineamiento devuelve el valor de similitud entre ambos n-gramas, considerando la longitud total de la secuencia objetivo, y los valores fijados para los parámetros de coincidencia, error y ventana de error. Posteriormente, este valor de similitud se transforma a un valor de distancia. Para ello, en primer lugar, se normaliza en el rango $[0, 1]$, siendo 1 el valor máximo de similitud (es decir, un n-grama consigo mismo) y 0 el mínimo. A continuación, para obtener el valor de distancias se aplica uno menos el valor de similitud normalizado. De esta forma, el valor obtenido está también en el rango $[0, 1]$, siendo 0 el valor de mínima distancia (un n-grama consigo mismo) y 1 el valor de máxima distancia entre n-gramas. Cabe destacar que, no pueden existir valores de distancia negativos. Estos valores de

distancias se apilan en una una matriz simétrica de tamaño NxN, donde N es el número de n-gramas para un usuario específico.

El resultado de este proceso es, por tanto, una matriz de distancias que representa la disimilitud entre los comportamientos generados por un mismo usuario. Por tanto, cada usuario obtendrá una matriz de distancias específica e independiente.

4.3. Extracción de los núcleos de comportamiento

La siguiente tarea se corresponde con utilizar la matriz de distancias, obtenida para cada usuario, para calcular los núcleos de comportamiento de los mismos. Identificar y definir correctamente estos núcleos de comportamiento es una de las tareas más críticas para poder generar un modelo de análisis de comportamiento preciso, pues es la fuente de conocimiento que se utilizará para comparar las nuevas muestras y, por lo tanto, la base que utiliza el modelo para detectar comportamientos anómalos que puedan suponer una brecha de seguridad. La mayoría de los sistemas de control de accesos del estado del arte utilizan toda la información disponible de los comportamientos para cada usuario. Sin embargo, en el propio comportamiento de los usuarios pueden existir valores atípicos que pueden afectar negativamente al rendimiento y precisión del clasificador. Además, utilizar todos los datos en lugar de un subconjunto de los mismos puede generar latencias añadidas, tanto a la hora de entrenar el clasificador, como a la hora de predecir y evaluar nuevas muestras. Esto se debe a que cada secuencia (n-grama) se debe comparar contra un número mayor de información. De este modo, lo ideal sería obtener núcleos de comportamiento tan pequeños como sea posible, siempre y cuando estos núcleos representen correctamente el comportamiento del usuario de tal manera que el clasificador pueda generalizar correctamente.

En la presente propuesta, se utiliza el algoritmo de *clustering* basado en densidades llamado DBSCAN [123] para obtener los núcleos de comportamiento. La idea bajo este algoritmo es dividir el espacio de decisión en áreas de densidad. Para entrenar el DBSCAN, hay que fijar dos hiperparámetros. En primer lugar, la distancia máxima entre dos muestras que se consideran vecinas. Este hiperparámetro se denomina EPS y es el parámetro más crítico. En segundo lugar, el número mínimo de puntos que debe tener una vecindad para ser considerada como un *cluster*.

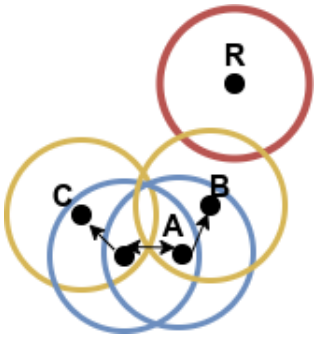


Figura 4.5: Ejemplo del funcionamiento del algoritmo de DBSCAN.

Este hiperparámetro se denomina $MINS$. Todos los vecinos dentro del radio EPS de un punto central se consideran parte del mismo *cluster*. Si alguno de estos vecinos es de nuevo un punto central, sus vecinos se incluyen de nuevo para su análisis. Los puntos no centrales de este conjunto se denominan puntos fronterizos. Los puntos que no son alcanzables desde ningún punto central se consideran ruido y no pertenecen a ningún cluster. DBSCAN no necesita información sobre el número de *clusters* deseados a priori para su entrenamiento. Esto es coherente con el hecho de que, es imposible saber cuántos núcleos de comportamiento (*clusters*) se obtendrán para un usuario específico.

El funcionamiento de DBSCAN se puede ver ilustrado en la Figura 4.5. En este ejemplo, el parámetro MIN es 4 y el radio EPS está representado por los círculos. A es un punto central. Los puntos B y C son puntos fronterizos. Las flechas indican la densidad. Los puntos B y C están conectados pues son alcanzables desde A tanto de forma directa, como por medio de otro punto central. R es un punto de ruido puesto que no es alcanzable desde A.

De este modo, la matriz de distancias (que solo contiene comportamiento generado por el propio usuario) alimenta el algoritmo DBSCAN. Así, las regiones de alta densidad obtenidas representan los núcleos de comportamiento, mientras que las regiones de baja densidad representan comportamientos atípicos del usuario. Estos comportamientos atípicos se descartan para que las muestras que se van a evaluar no puedan ser comparadas con ellos.

En la Figura 4.6, se puede observar un ejemplo, para un usuario específico, de los núcleos extraídos (puntos azules) y de los comportamientos atípicos (puntos rojos). Estos puntos se han representado utilizando las dos primeras componentes principales calculadas utilizando el algoritmo de escalado multidimensional [140] (solo la información más representativa está

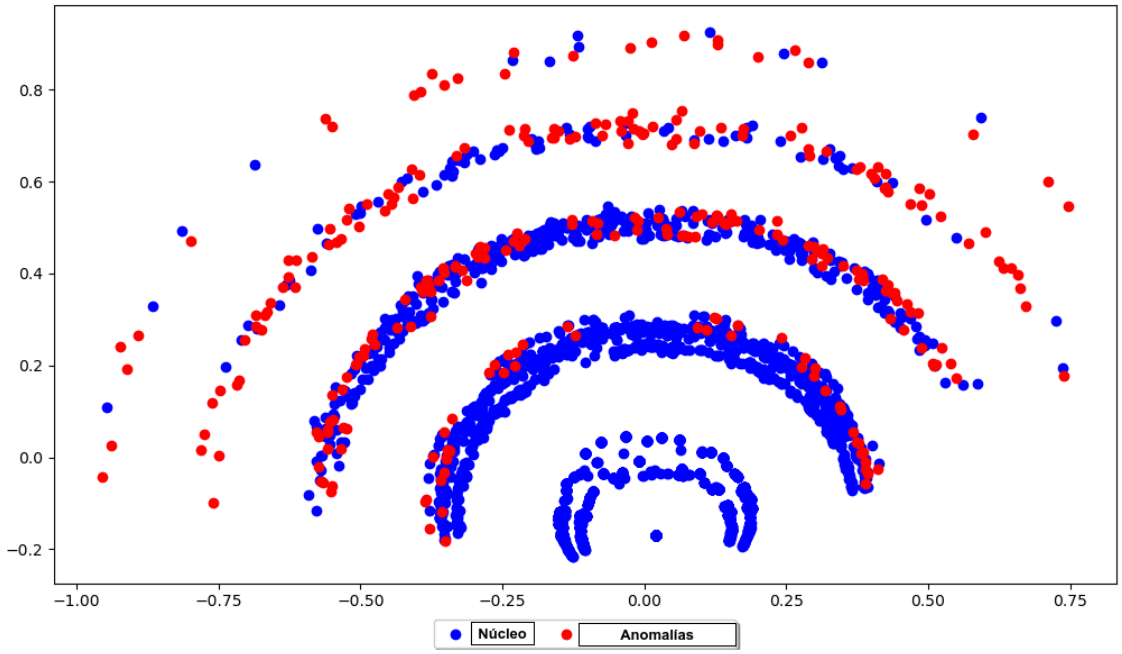


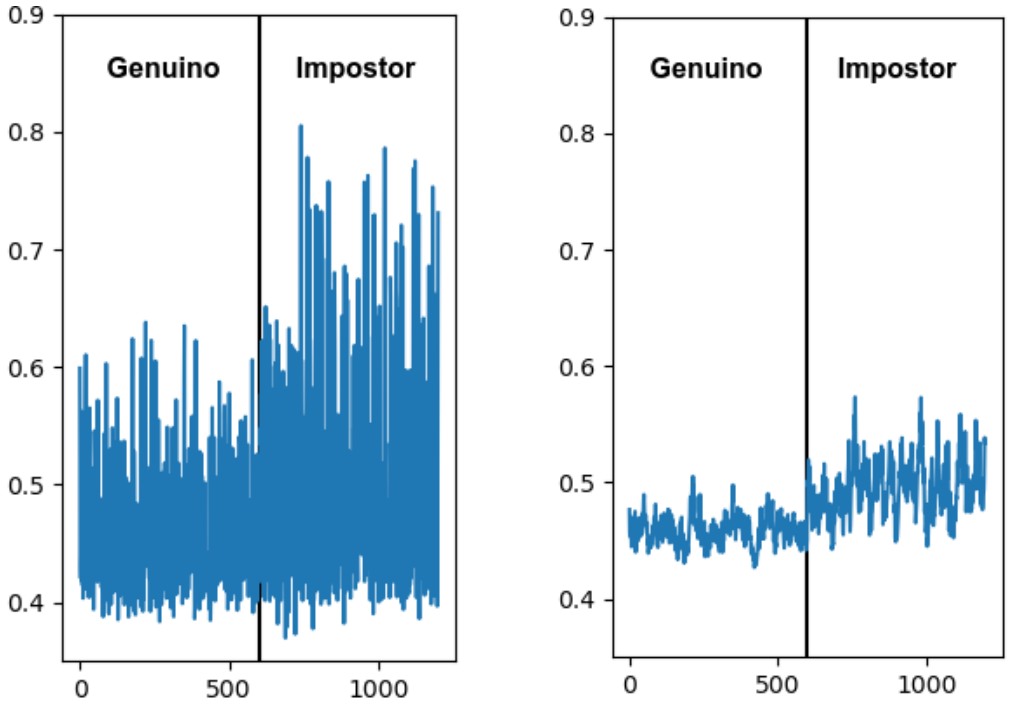
Figura 4.6: Representación de los núcleos de comportamiento de un usuario específico utilizando escalado multidimensional.

siendo visualizada en la figura). Como se puede observar, a medida que los puntos se alejan del núcleo principal (mostrado en la parte inferior de la figura), la distribución se va convirtiendo en más heterogénea, es decir, contiene un mayor número de comportamientos atípicos. Esto corrobora que DBSCAN está descartando correctamente los comportamientos que se desvían de la distribución de comportamientos esperada para este usuario concreto.

4.4. Modelo de riesgos

En esta sección, se va a explicar detalladamente la elaboración de un modelo de riesgos. Este modelo tiene el objetivo de clasificar las muestras en función de su similitud con los núcleos de comportamiento obtenidos previamente. Esto permite categorizar estas nuevas muestras en genuinas, o por el contrario detectar si son anomalías y por lo tanto pertenecen a un impostor.

En primer lugar, toda nueva muestra se procesa siguiendo los pasos explicados en las secciones anteriores. De este modo, se obtienen los n-gramas que representan el comportamiento de dichas muestras. Estos n-gramas se comparan utilizando las técnicas de alineamiento de ADN con todos los n-gramas de los núcleos de comportamiento. El resultado de esta operación es



(a) Valores de riesgo en bruto.

(b) Valores de riesgo aplicando MME.

Figura 4.7: Valores del *buffer* de riesgo para un usuario concreto.

un vector de distancias para cada nueva muestra, cuya longitud es el número de puntos de los núcleos de comportamiento. Este vector, por lo tanto, determina la distancia entre esta nueva muestra y el conocimiento extraído previamente para un usuario concreto.

La premisa de partida es que un mismo usuario va a generar, por norma general, vectores de distancias con valores pequeños (semejanzas altas), mientras que las dinámicas de comportamiento generadas por un usuario impostor resultarán en vectores de distancias con valores altos (poco semejantes con los núcleos de comportamiento).

El riesgo asociado a esta nueva dinámica de comportamiento se puede calcular como la media de los valores del vector de distancias. De este modo, un valor bajo de riesgo determina que la nueva dinámica (en forma de n-grama) se encuentra cerca de los núcleos de comportamiento y, por lo tanto, es altamente probable que pertenezca al mismo usuario objetivo. Por el contrario, un valor alto de riesgo representa que esta nueva dinámica se encuentra lejos de los núcleos de comportamiento y por lo tanto es probable que no pertenezca al usuario objetivo, sino que sea de un usuario impostor. Este proceso permite comparar cada nueva dinámica de comportamiento de forma individual. Sin embargo, en un entorno real se generan multitud de dinámicas

de comportamiento y además lo hacen de forma secuencial. De este modo, estos valores de riesgo se ordenan a lo largo del tiempo, generando así, un *buffer* de riesgo (ver Figura 4.7a). Tal y como se puede observar, los valores de riesgo obtenidos son muy dispares, es decir, son muy cambiantes a lo largo del tiempo. Esto se debe a que cada dinámica de comportamiento, de forma independiente, puede ser muy parecida o distinta a los núcleos de comportamiento calculados. Este resultado se traduce en que el sistema final obtiene multitud de falsos positivos y falsos negativos.

Tomando como base la premisa de partida, se procede a suavizar el *buffer* de riesgo obtenido. Para ello se aplica la Media Móvil Exponencial (MME), con el objetivo de reducir los altos cambios que se producen en los valores de riesgos, de la siguiente manera:

$$MME_t = \begin{cases} Y_1, & \text{si } t = 1 \\ \alpha Y_t + (1-\alpha) \cdot MME_{t-1}, & \text{si } t > 1 \end{cases}$$

donde MME_t es la media móvil exponencial en el instante t, Y_t es el valor de riesgo en un instante t y α es el coeficiente de suavizado en el rango $[0, 1]$. Un valor de α pequeño pondera más alto las observaciones más antiguas, mientras que un valor alto repercute en un modelo más olvidadizo que pondera más las observaciones últimas y cercanas al instante t.

El coeficiente α se suele calcular en función del número de observaciones pasadas a tener en cuenta [141]. En este caso, se calcula de acorde al parámetro $WinSize$ como sigue:

$$\alpha = 2/(WinSize + 1)$$

donde $WinSize$ es el número de observaciones a tener en cuenta de la secuencia de riesgos. Una vez se ha aplicado la MME, los valores pasan de ser muy cambiantes de lo largo del tiempo, a ser mucho más estables (ver Figura 4.7b). Esto se debe a que, en esta ocasión, no es un único valor el que determina el riesgo asociado para un instante, sino el conjunto de $WinSize$ observaciones el que lo hace, es decir, el histórico de los valores de riesgo. Es por esto que para establecer un nivel óptimo del parámetro $WinSize$ debe existir un equilibrio entre un mejor desempeño a la hora de realizar predicciones y la usabilidad del método en un entorno real. Un valor alto del parámetro $WinSize$, considera más información y por lo tanto suaviza mejor la curva obteniendo así más exactitud al categorizar las dinámicas de comportamiento. Sin embargo un valor bajo

del parámetro $WinSize$ genera un modelo de riesgos más usable, pues al considerar menos información se necesitan menos recursos computacionales para su funcionamiento y también se pueden empezar a realizar predicciones en un intervalo menor de tiempo.

Una vez calculado el *buffer* de riesgo, la siguiente tarea es determinar que comportamientos se consideran normales y cuales se consideran anómalos. Esto permite categorizar las observaciones en genuinas (pertenecen al mismo usuario objetivo) y en impostores (pertenecen a un usuario impostor). Para lograr esto se tiene que establecer una frontera de decisión en la curva de riesgo. Para ello se determina un umbral de tal manera que los valores de riesgo por debajo del umbral se consideran genuinos, y los valores por encima del umbral se consideran impostores (ver Figura 4.8).

La forma de determinar el umbral de forma óptima se fijándolo al valor que produce el menor EER. Tal y como se ha explicado en la Sección 3.3.4, este valor se calcula como la intersección entre la función de FAR y la función de FRR. A modo recordatorio, FAR se define como la probabilidad de que un usuario no autorizado (impostor) sea aceptado por el modelo. Por otro lado, FRR se define como la probabilidad de un usuario autorizado (genuino) sea rechazado injustamente por el modelo (sea considerado impostor). Tanto FAR como FRR son funciones, pues se pueden obtener múltiples valores para ellos dependiendo del umbral seleccionado. Fijar un valor extremo del umbral resulta en obtener un modelo restrictivo (menor FAR pero mayor FRR) o un modelo permisivo (menor FRR pero mayor FAR), pero nunca ambos valores menores simultáneamente. De esto modo, el EER se considera como el punto óptimo para fijar el umbral, pues es el valor que consigue obtener el mínimo valor para FAR y FRR simultáneamente.

Por último, cabe destacar que cuando se generan los n-gramas, la secuencia adyacente obtenida es la misma que la posterior excepto por el primer y último símbolo. Es por esto que el método propuesto en esta tesis puede realizar predicciones para cada interacción del usuario (es decir, para cada pulsación de teclado o movimiento de ratón) una vez que se han obtenido al menos $WinSize$ interacciones (ver Figura 4.9).

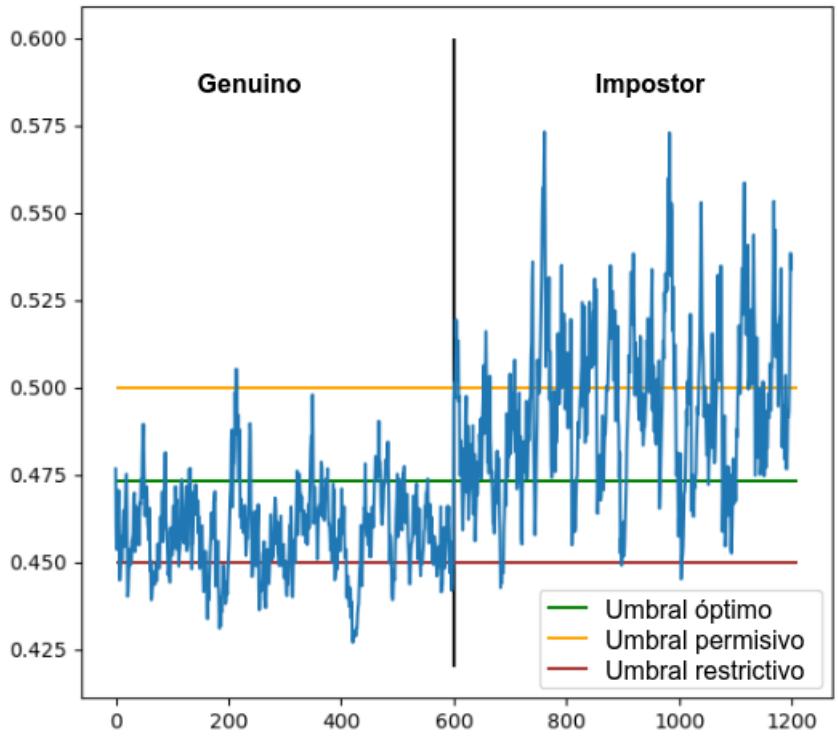


Figura 4.8: Tipos de umbrales aplicados sobre el *buffer* de riesgo utilizando MME.

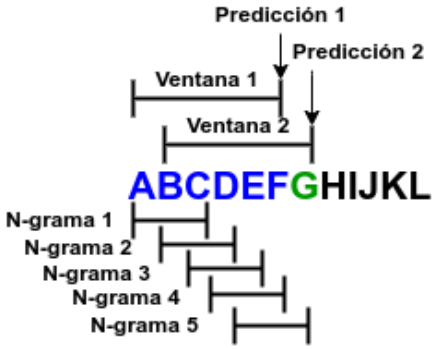


Figura 4.9: Secuencia de predicciones en función de los parámetros del método. *N-Gram Length (NGL)* = 3 y *WinSize*=4.

4.5. Selección de parámetros

El método de combinación de la información propuesto en esta investigación necesita definir un total de dos parámetros y cuatro hiperparametros (ver Tabla 4.1). Los resultados obtenidos están muy ligados a la selección correcta de estos parámetros. Es por esto que es necesario hacer especial atención en entender y definir correctamente cada uno de ellos. Los parámetros requeridos por el modelo son *NGL* y *WinSize*.

El valor de *NGL* determina el número de interacciones (teclado, ratón o ambas) que se agrupan para cada instante con el objetivo de obtener una predicción. Un valor extremo de este parámetro puede resultar en subajuste (para valores pequeños) y sobreajuste (para valores altos). La búsqueda de este parámetro se ha limitado a los valores $[5, 10, 20, 30]$ basándose en la experiencia empírica.

El valor de $WinSize$ define cuanta información histórica de comportamiento (en forma de secuencias de n-gramas) va a ser utilizada para realizar una predicción. El objetivo de este parámetro es suavizar el *buffer* de riesgo. La búsqueda de este parámetro de ha limitado a los valores $[5, 10, 20, 50, 100]$ basándose en la experiencia empírica.

El algoritmo de RTE necesita fijar dos hiperparámetros: el *Número de árboles* y la *Máxima profundidad* de cada árbol. Ambos hiperparámetros determinan la longitud del abecedario en el que cada muestra puede ser categorizada. De esta forma, un valor alto puede resultar en sobreajuste, pues se obtendrá un alfabeto muy amplio, en el que cada símbolo representa algo muy específico, haciendo que las secuencias obtenidas no se parezcan entre sí. Por otro lado, un valor pequeño de ambos resultara en un abecedario muy limitado, obteniendo así secuencias muy parecidas entre sí y por lo tanto no discriminatorias (subajuste). Para cada fuente de información, la búsqueda de estos hiperparámetros se ha limitado al rango $[2, 5]$ basándose en la experiencia empírica.

Para el algoritmo de DBSCAN se necesitan fijar también dos hiperparámetros: EPS y $MINS$. Estos hiperparámetros se han delimitado utilizando una búsqueda en cuadrículas (en inglés, *grid search*). Para hacer la búsqueda más eficiente, los valores posibles se han acotado. De esta forma, los valores de EPS están en el rango $[0 : f_{100}]$, donde f_{100} representa la distancia con posición cien de la matriz de distancias ordenada de menor a mayor y descartando los valores iguales a cero. El parámetro $MINS$ se ha limitado a los valores $[2, 10, 50, 100]$ basándose en la experiencia empírica.

Cabe destacar que fijar de forma óptima estos parámetros dependerá ampliamente de los requisitos particulares de los datos en donde se quiera aplicar el método. Sin embargo, se debe tener en cuenta algunas consideraciones. Fijar los parámetros *NGL* y $WinSize$ a valores altos repercute en obtener un modelo con más exactitud a la hora de clasificar dinámicas de comportamiento. Esto se debe a que, el aumento de cualquiera de ellos hace que se considere más

Nombre	**Descripción**	**Valores**
NGL	Longitud del N-grama	[5, 10, 20, 30]
WinSize	Información histórica considerada en forma de secuencias	[5, 10, 20, 50, 100]
Número de árboles	Número de árboles generados en el RTE	[2, 5]
Máxima profundidad	Máxima profundidad de los árboles del RTE	[2, 5]
EPS	Máxima distancia entre dos muestras para ser consideradas vecinas en DBSCAN	$[0 : f_{100}]$
MINS	Mínimo número de puntos que tiene que tener una vecindad para considerarse *cluster* en DBSCAN	[2, 10, 50, 100]

Tabla 4.1: Resumen de los parámetros e hiperparametros del método de combinación de la información de comportamientos. f_{100} representa la distancia con posición cien de la matriz de distancias ordenada de menor a mayor y descartando los valores iguales a cero.

información para realizar una predicción. Sin embargo, esto también repercute en la usabilidad.

Por otro lado, para fijar los hiperparametros de RTE y DBSCAN se recomienda utilizar una búsqueda en cuadrículas acotando los posibles valores para realizarla de forma eficiente. Hay que tener en cuenta las consideraciones explicadas anteriormente para evitar tanto el sobreajuste como el subajuste.

Capítulo 5

Evaluación y validación de la propuesta

En esta sección se detallan los experimentos realizados para poder evaluar y validar la propuesta. En primer lugar, se evalúa el flujo de trabajo propuesto en el Capítulo 3. Finalmente, se evalúa el método propuesto en el Capítulo 4 para la combinación de información de comportamientos.

5.1. Evaluación del flujo de trabajo

5.1.1. Desarrollo del entorno de trabajo y conjunto de datos UEBA

Con el objetivo de poder evaluar el flujo de trabajo propuesto, se ha desarrollado un entorno de trabajo totalmente funcional que implementa la gestión de identidades federada. Para ello, se ha desarrollado una aplicación de chat, cuyo rol es el de RP, y un IdP cuya función es la de gestionar las identidades digitales necesarias para realizar los experimentos.

El chat se ha desarrollado utilizando la aplicación de código abierto *Letschat* [142], el cual se encuentra bajo la licencia MIT. Esta aplicación se ha desarrollado en *Node.js* [143] y utiliza una base de datos *MongoDB* [144] para almacenar toda la información necesaria. La aplicación es compatible con la mayoría de navegadores web del mercado y es multidipositivo. En el presente trabajo de tesis, esta aplicación ha sido ejecutada en un ordenador portátil MSI GF62 8RD-256XES equipado con un procesador Intel Core i7 8750H (2.2 GHz, 9MB). Además, tiene

una memoria RAM de 16 GB DDRIV (2666 MHz).

Por otro lado, se ha desarrollado un componente que se integra dentro del chat con el objetivo de recopilar la información de comportamiento. Este componente, desarrollado en *Javascript* y basado en el trabajo [145], se integra en el cliente web donde se ejecuta la aplicación. Su funcionalidad es la de recoger la información relacionada con los eventos de pulsación de teclas en el teclado y de movimientos de ratón y almacenarla. Los eventos relacionados con el teclado son recopilados cada vez que suceden, sin embargo, los eventos de ratón se han de recoger por *pooling*, es decir, cada cierto tiempo. En este caso concreto los eventos de *pooling* se han recopilado cada 200 milisegundos debido a los recursos de memoria y computación disponibles.

Finalmente, el IdP se ha desarrollado utilizando el *framework* OpenAM [146]. De esta forma, se ha hecho uso del *Web application Resource* (WAR) para instalar la versión 13.5. Una vez instalado, y teniendo en cuenta los requisitos del presente trabajo, se ha implementado un cliente OAuth 2.0/OpenID Connect. Este cliente proporciona de forma sencilla todos los mecanismos para poder realizar los flujos de autorización y autenticación disponibles en dichas especificaciones.

Este desarrollo, ha dado lugar a el conjunto de datos denominado *Keystroke and Mouse Dynamics for UEBA Dataset*, el cual ha sido recopilado para el presente trabajo de tesis doctoral y está disponible públicamente para la comunidad científica [147]. Para recopilar el conjunto de datos, un grupo de once investigadores de la Universidad Rey Juan Carlos heterogéneos en cuanto a edad y género, han hecho uso de la aplicación de chat. Todos los participantes, mayores de edad, han recibido información sobre la finalidad del experimento y han participado en él de manera completamente voluntaria, sin recibir ningún tipo de presión y proporcionando para ello su consentimiento explícito. A todos se les ha informado de los mínimos riesgos que corrían, ya que solo se recogían datos relacionados con el uso explícito de la aplicación de chat, que no permiten identificarles y que además no se asociaban a sus datos personales. De este modo, se han recopilado dos bases de datos denominadas $EVTRACKINFO$ y $EVTRACKTRACK$. Estas bases de datos mantienen una relación 1:N entre ellas respectivamente. $EVTRACKINFO$ contiene información de sesión, es decir, almacena los atributos estáticos como, por ejemplo, el usuario autenticado, agente de usuario y la resolución de la pantalla asociado a cada sesión (ver Sección 3.3.1). Por otro lado, $EVTRACKTRACK$ contiene la información asociada a las

dinámicas de comportamiento de cada sesión. Esto es, los eventos derivados de las pulsaciones de teclado y de ratón.

En total, 347 y 142691 registros han sido recopilados para *EVTRACKINFO* y *EVTRACKTRACK* respectivamente. Del total de las dinámicas de comportamiento, 113471 pertenecen a dinámicas de teclado, mientras que 29220 pertenecen a dinámicas de ratón. Para cada usuario, se han obtenido un total de 28524 ± 18541 registros (media ± desviación típica).

El entorno de trabajo desarrollado y el conjunto de datos obtenido se han utilizado para evaluar y validar todas las tareas que componen el flujo de trabajo propuesto. De aquí en adelante, se detalla la implementación de cada una de estas tareas.

5.1.2. Selección de la huella digital en el conjunto de datos UEBA

En el caso del chat desarrollado, se ha seleccionado una RP con un perfil de seguridad medio (ver Sección 3.3.1). Hay que destacar, que de acuerdo con los bajos recursos computacionales del servidor donde se ejecuta la aplicación y a la gran cantidad de usuarios que pueden interactuar con la aplicación de forma simultánea, la huella digital debe ser lo más simple posible con el objetivo de no saturar el sistema, pero lo suficientemente discriminatoria para poder diferenciar entre los distintos comportamientos de los usuarios.

De esta forma, el agente de usuario y la resolución de pantalla se han seleccionado como atributos estáticos, mientras que las dinámicas de teclado y las dinámicas de ratón se han seleccionado como atributos dinámicos. Cabe destacar que, debido a la naturaleza de la aplicación de chat, se espera obtener una recopilación masiva de datos, ya que el teclado y el ratón van a ser utilizados asiduamente por los usuarios.

5.1.3. Generación de la huella digital en el conjunto de datos UEBA

El objetivo de esta tarea es generar una huella digital para cada usuario con la suficiente singularidad como para ser distintiva entre usuarios. De esta forma, los atributos estáticos y los atributos dinámicos son recopilados haciendo uso del componente específicamente desarrollado para ello. Esta información está recogida en bruto y, por tanto, se trata para poder extraer conocimiento y generar características significativas que puedan alimentar a los modelos de

aprendizaje máquina. En el caso de los atributos estáticos la información en bruto es transformada directamente a variables categóricas. En el caso de los atributos dinámicos se distinguen dos posibles casuísticas: las dinámicas de teclado y las dinámicas de ratón.

En el caso de las dinámicas de teclado, cada pulsación de una tecla, es decir, cada interacción, genera dos tipos de eventos principalmente (debido a la tecnología seleccionada para su implementación): *keydown* y *keyup*. El evento de *Keydown* contiene la marca de tiempo sobre cuándo se ha inicializado la pulsación, mientras que el evento *keyup* contiene la marca de tiempo de cuando la pulsación ha concluido, es decir, cuando el usuario ha dejado de mantener la tecla pulsada. Estas interacciones se agrupan en ventanas temporales de dos pulsaciones formando digrafos. De esta forma, cada digrafo contiene cuatro marcas de tiempo (para cada pulsación un evento de *keyupX* y otro de *KeydownX*, donde X representa el número de interacción). De esta forma, se calculan seis características o variables para cada digrafo [105]:

- H1 (*keyup1-Keydown1*): tiempo transcurrido para la primera interacción.
- H2 (*Keyup2-Keydown2*): tiempo transcurrido para la segunda interacción.
- HP (*Keydown2-Keyup1*): tiempo transcurrido desde que termina la primera interacción hasta que se inicia la segunda interacción.
- PP (*Keydown2-Keydown1*): tiempo transcurrido desde que se inicia la primera interacción hasta que se inicia la segunda interacción.
- HH (*Keyup2-Keyup1*): tiempo transcurrido desde que acaba la primera interacción hasta que acaba la segunda interacción.
- PH (*Keyup2-Keydown1*): tiempo transcurrido desde que se inicia la primera interacción hasta que acaba la segunda interacción.

Además, se almacena información sobre la propia tecla pulsada, esto es, se almacenan los caracteres específicos pulsados para cada digrafo. Estos caracteres se categorizan teniendo en cuenta una división física del teclado, seleccionando las teclas que se encuentran a la izquierda del teclado y aquellas a la derecha en otro grupo. También se incluye la posición en el eje

vertical, arriba y abajo para cada uno de los grupos anteriormente mencionados. De esta forma, esta categoría puede tomar cuatro valores distintos.

En el caso de las dinámicas del ratón, se han extraído cinco características. Al igual que en el caso del teclado, los eventos del ratón se han agrupado formando digrafos. Para cada digrafo se ha calculado: el ángulo, la distancia, la velocidad, el tiempo total transcurrido y el tipo de evento de *Javascript*.

Utilizando las características recopiladas, se genera una huella digital para cada usuario. Esta huella contiene tres componentes: el componente de los atributos estáticos, el componente de las dinámicas de teclado y el componente de las dinámicas de ratón. La información de cada uno de los componentes está almacenada en forma de vector.

5.1.4. Modelado y evaluación

Los procesos de modelado y evaluación tienen como objetivo construir los modelos de aprendizaje máquina, basados en UEBA, que clasifican las dinámicas de comportamiento. Estos procesos se han considerado conjuntamente en esta sección, pues se van a realizar de forma simultánea, avanzando y retrocediendo hasta llegar a unos modelos con alta capacidad de generalización, robustos y con eficacia alta.

Los experimentos aquí detallados se han centrado en desarrollar un modelo de aprendizaje máquina capaz de comparar y evaluar las huellas digitales generadas en los pasos anteriores. En primer lugar, los datos recopilados se han separado en tres conjuntos: conjunto de entrenamiento, test y validación. El conjunto de entrenamiento contiene el 70 % de los datos de cada usuario específico (muestras genuinas). Los conjuntos de test y de validación contienen, respectivamente, un 15 % de los datos restantes. Los conjuntos de test y validación son completados utilizando muestras atípicas para simular comportamientos anómalos, es decir, comportamientos que pueden suponer una brecha de seguridad y por tanto se quieren detectar (muestras de impostores). Para llevarlo a cabo, se seleccionan de forma aleatoria tantas muestras del resto de usuarios como tenga el propio usuario para cada uno de los conjuntos. En conclusión, el conjunto de entrenamiento solo contiene muestras genuinas y se utiliza para entrenar los modelos de detección de anomalías pertinentes. Por otro lado, los conjuntos de test y validación contienen

muestras tanto genuinas, como anómalas, y se utilizan para evaluar la eficacia de los modelos propuestos y determinar si son robustos y capaces de generalizar correctamente.

En cuanto a la comparación de huellas digitales, cada componente (atributos estáticos, dinámicas de teclado y dinámicas de ratón) se modela de forma independiente. De esta forma, los componentes pueden ser activados o desactivados para poder evaluar la comparación de huellas digitales de forma independiente para cada uno de ellos.

En primer lugar se ha implementado un NB para comparar los atributos estáticos. Este clasificador asume que cada variable es totalmente independiente y, por lo tanto, cada variable determina la probabilidad de pertenecer a la clase genuina o a la impostora [148]. Debido al bajo número de participantes en el conjunto de datos UEBA, los resultados obtenidos muestran una clasificación perfecta. Esto quiere decir que simplemente considerando el agente de usuario y la resolución de pantalla, este simple clasificador es capaz de determinar si un usuario es genuino o impostor de forma exacta. En el caso de que el número de usuarios aumentase, el número de atributos estáticos que habría que considerar sería mayor y muy probablemente no se conseguirá esta clasificación perfecta. Además, estos atributos estáticos son fáciles de manipular y falsear (para los atributos dinámicos la complejidad de manipulación o falseo es mayor o incluso imposible). Por otro lado, ciertos ataques pueden tomar el control de la máquina del usuario genuino y por lo tanto hacer que estos atributos sean idénticos a los esperados e imposibles de detectar por el modelo. Es por esto que en caso de que el número de usuarios aumente, se recomienda aumentar el nivel de seguridad de las RPs (actualmente se ha seleccionado un nivel medio de seguridad).

En el caso de las dinámicas de teclado, los digrafos generados se agrupan en n-gramas [149]. De esta forma se concatenan los digrafos formando vectores de longitud dos, tres y seis. Sobre estos vectores se utiliza la distancia de Manhattan [150] para comparar las dinámicas de comportamiento asociadas al teclado [85].

El conjunto de entrenamiento para cada usuario se utiliza para calcular la media de cada característica recopilada. Posteriormente se extraen los vectores de características para el conjunto de test. De esta forma se utiliza la distancia de Manhattan para comparar estos vectores con la media calculada en el conjunto de entrenamiento. De este modo se obtiene la similitud entre estas muestras y lo esperado. Con estas similitudes calculadas en el conjunto de test se puede

fijar un umbral de decisión utilizando el EER. En caso de que la similitud entre las muestras sea menor al umbral determinado se clasifica como genuina, mientras que en caso de que supere el umbral se clasifica como impostor.

Con el umbral ya fijado, se procede a evaluar el conjunto de validación. Los datos almacenados en este conjunto no han sido nunca antes vistos por el modelo y, por lo tanto, permiten evaluar el modelo de forma justa y analizar la capacidad de generalización del modelo obtenido. Los resultados obtenidos para este modelo son bastante pobres, obteniendo un EER de 0,511 ± 0,124 (media ± desviación típica). Estos resultados se obtienen porque los datos recopilados no tienen una frecuencia o distribución predeterminada, a diferencia de otros trabajos del estado del arte que sí asumen un patrón determinado (usuarios introduciendo la misma contraseña un número determinado de veces). Es por esto que se deben incluir ciertas mejoras en el modelo para obtener unos resultados óptimos.

Siguiendo esta línea se ha realizado un proceso de exploración de datos más exhaustivo, permitiendo determinar que los vectores que comienzan con la misma pulsación, tienden a ser más similares entre sí que los que comienzan con una pulsación distinta (p. ej. los vectores que comienzan por la pulsación del carácter 'a' son más similares entre sí, a los que comienzan por el carácter 'b' y viceversa). De esta forma, se han agrupado los vectores en función de su primera pulsación. En este caso, también se ha observado que el vector medias que se utiliza en el conjunto de entrenamiento no es lo suficientemente discriminatorio debido a la alta dispersión de los datos de cada usuario. Es por esto que se ha decidido sustituir este vector medias por un modelo de KNN utilizando la misma distancia de Manhattan. De esta forma, cada muestra no se va a comparar contra el vector media de características, sino contra la etiqueta conocida que posean los n-vecinos más cercanos. Al igual que para el modelo anterior, se utiliza el EER para fijar un umbral óptimo de clasificación.

Los resultados obtenidos se pueden observar en la Tabla 5.1. Estos resultados confirman que el modelo mejorado representa de forma óptima el comportamiento de los usuarios y por consiguiente clasifica correctamente un número elevado de muestras, es decir, obtiene valores de EER FAR y FRR menores.

El modelo obtenido compara cada vector de características de forma totalmente independiente. Sin embargo, en un entorno real, estos vectores de características se van generando a

	Test			Validación	
N-grama	**FAR**	**FRR**	**EER**	**FAR**	**FRR**
2	0,257	0,265	0,296 (0,099)	0,321	0,304
3	0,284	0,294	0,308 (0,099)	0,317	0,309
6	0,279	0,294	0,347 (0,141)	0,267	0,299

Tabla 5.1: Resultados del modelo de clasificación KNN utilizando la distancia de Manhattan para las dinámicas de teclado agrupando por tecla pulsada: FAR, FRR y EER (desviación típica).

lo largo del tiempo y de forma ordenada a medida que los usuarios van interaccionando con la aplicación. Esto se traduce, en que las distancias obtenidas pueden también ordenarse y analizarse a lo largo del tiempo y no compararse una a una como se ha hecho hasta ahora. Para realizar esto, se propone utilizar un *buffer* temporal. En este caso, las distancias entre vectores se utilizan para llenar el *buffer*. De esta manera, si la distancia obtenida para un vector concreto supera el umbral determinado, el *buffer* se llenará. En caso contrario, es decir, si la distancia es menor que el umbral, el *buffer* disminuirá. El valor a aumentar o disminuir en el *buffer* es la distancia a dicho umbral. De esta forma, se determina un nuevo umbral de clasificación para este *buffer* basándose, al igual que para los modelos previos, en el menor EER.

A continuación se utilizan los conjuntos de test y validación para evaluar el modelo. En esta ocasión, los datos pertenecientes a usuarios impostores se concatenan al final de las muestras genuinas. Los resultados para este experimento se encuentran en la Tabla 5.2. Tal y como se puede observar, incluir la información temporal hace que el modelo aumente considerablemente su eficacia a la hora de clasificar las muestras de test correctamente. Además, el modelo obtenido es robusto, pues también obtiene resultados semejantes y óptimos para el conjunto de validación. Sin embargo, este modelo contiene un sesgo pues se está asumiendo que los comportamientos anómalos suceden siempre al final de las muestras genuinas, algo que en un escenario real no es así necesariamente. En un escenario real las anomalías de comportamiento pueden suceder en cualquier instante temporal pues no se puede saber a priori cuando un usuario va a sufrir un ataque ni la duración del mismo. Es por esto que el modelo generado, a pesar de no ser una buena solución para un entorno real, sirve para corroborar que, si se considera la

	Test			Validación	
N-grama	**FAR**	**FRR**	**EER**	**FAR**	**FRR**
2	0,125	0,128	0,133 (0,107)	0,012	0,156
3	0,087	0,091	0,099 (0,126)	0,050	0,082
6	0,130	0,140	0,175 (0,156)	0,068	0,144

Tabla 5.2: Resultados del modelo de clasificación KNN utilizando la distancia de Manhattan y el *buffer* temporal para las dinámicas de teclado: FAR, FRR y EER (desviación típica). Modelo de referencia.

suficiente información de comportamientos, se pueden obtener resultados óptimos a la hora de clasificar las muestras. Es decir, sirve para generar un modelo de referencia cuyos resultados son los mejores que se pueden obtener siguiendo los pasos aquí descritos.

Con el objetivo de definir un experimento más realista y semejante a un entorno real, los vectores de características se agrupan en sesiones utilizando ventanas temporales. De esta forma, los vectores de características se agrupan en sesiones de longitud 5, 10 y 20. Por ejemplo, para una longitud de sesión de 10, 10 vectores genuinos se concatenan con 10 vectores impostores. Este vector resultante va a formar una sesión independiente. Los resultados obtenidos para esta situación más realista se pueden encontrar en la Tabla 5.3 y la Tabla 5.4. Los mejores resultados se obtienen utilizando n-gramas de longitud 2 y 3. Los resultados obtenidos son peores que los obtenidos para el modelo que se toma como referencia en el paso anterior, sin embargo, se llega a una clasificación mejor a medida que el tamaño de sesión aumenta. Esto corrobora que, cuanta más información es considerada, mejores resultados se obtienen.

La Figura 5.1 ilustra un ejemplo de los valores del *buffer* para un usuario concreto. La línea azul representa el valor del *buffer* a lo largo del tiempo. La línea negra horizontal representa el umbral de decisión. La línea verde representa que una predicción se ha realizado correctamente, tanto en el caso de clasificar una muestra genuina como en el de clasificar una muestra de un impostor. La línea roja representa una predicción errónea.

Finalmente, los mismos pasos se han seguido para el caso de las dinámicas de ratón. En primer lugar, se han utilizado n-gramas de longitud 1, 2, 3 y 6. Posteriormente, se ha considerado la distancia de Manhattan para comparar de forma independiente cada vector contra el vector

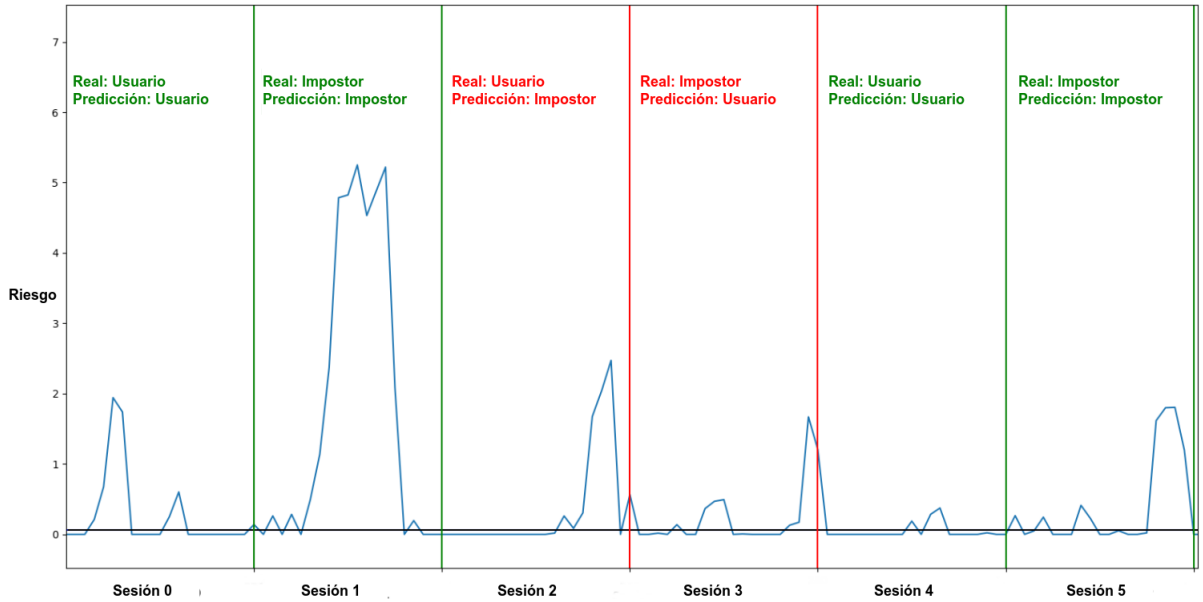


Figura 5.1: Valores del *buffer* utilizando las dinámicas de teclado para un usuario concreto.

Tamaño de sesión	**N-grama**	**FAR**	**FRR**	**EER**
5	2	0,221	0,213	0,238 (0,172)
5	3	0,254	0,242	0,205 (0,136)
5	6	0,235	0,197	0,247 (0,195)
10	2	0,169	0,162	0,279 (0,120)
10	3	0,185	0,194	0,188 (0,157)
10	6	0,221	0,228	0,206 (0,206)
20	2	0,141	0,162	0,128 (0,140)
20	3	0,172	0,140	0,145 (0,189)
20	6	0,153	0,172	0,174 (0,208)

Tabla 5.3: Resultados del modelo de clasificación KNN utilizando la distancia de Manhattan y el *buffer* temporal para las dinámicas de teclado divididas en sesiones utilizando en el conjunto de test: FAR, FRR y EER (desviación típica).

medias. Al igual que para el caso del teclado, los resultados obtenidos son bastante pobres, obteniendo un EER de $0{,}496 \pm 0{,}160$ para la mejor selección del parámetro (longitud de n-grama 6). En este caso concreto, para mejorar el modelo inicial, se ha optado por utilizar OC-SVM. Los resultados obtenidos se pueden observar en la Tabla 5.5. Estos resultados no se pueden

Tamaño de sesión	N-grama	FAR	FRR
5	2	0,199	0,220
5	3	0,213	0,213
5	6	0,262	0,211
10	2	0,170	0,216
10	3	0,170	0,170
10	6	0,155	0,196
20	2	0,126	0,188
20	3	0,161	0,110
20	6	0,155	0,180

Tabla 5.4: Resultados del modelo de clasificación KNN utilizando la distancia de Manhattan y el *buffer* temporal para las dinámicas de teclado divididas en sesiones utilizando en el conjunto de validación: FAR y FRR.

	Test			Validación	
N-grama	FAR	FRR	EER	FAR	FRR
1	0,447	0,448	0,455 (0,028)	0,440	0,441
2	0,442	0,443	0,427 (0,058)	0,444	0,447
3	0,440	0,440	0,432 (0,060)	0,442	0,445
6	0,447	0,419	0,407 (0,081)	0,476	0,468

Tabla 5.5: Resultados del modelo de clasificación OC-SVM para las dinámicas de ratón de forma independiente: FAR, FRR y EER (desviación típica).

comparar con los obtenidos para las dinámicas de teclado pues pertenecen a dos fuentes de datos totalmente distintas.

Siguiendo las líneas definidas para el caso de las dinámicas de teclado, se procede a utilizar un *buffer* temporal. En esta primera aproximación, para los conjuntos de test y de validación, se concatenan todas las muestras de impostores al final de las muestras genuinas. Los resultados obtenidos se pueden observar en la Tabla 5.6. Estos son los resultados pertenecientes al modelo referencia.

	Test			Validación	
N-grama	**FAR**	**FRR**	**EER**	**FAR**	**FRR**
1	0,072	0,108	0,151 (0,083)	0,115	0,093
2	0,205	0,214	0,249 (0,162)	0,168	0,088
3	0,399	0,399	0,409 (0,081)	0,306	0,173
6	0,415	0,447	0,399 (0,088)	0,460	0,479

Tabla 5.6: Resultados del modelo de clasificación OC-SVM utilizando el *buffer* temporal para las dinámicas de ratón: FAR, FRR y EER (desviación típica). Modelo de referencia.

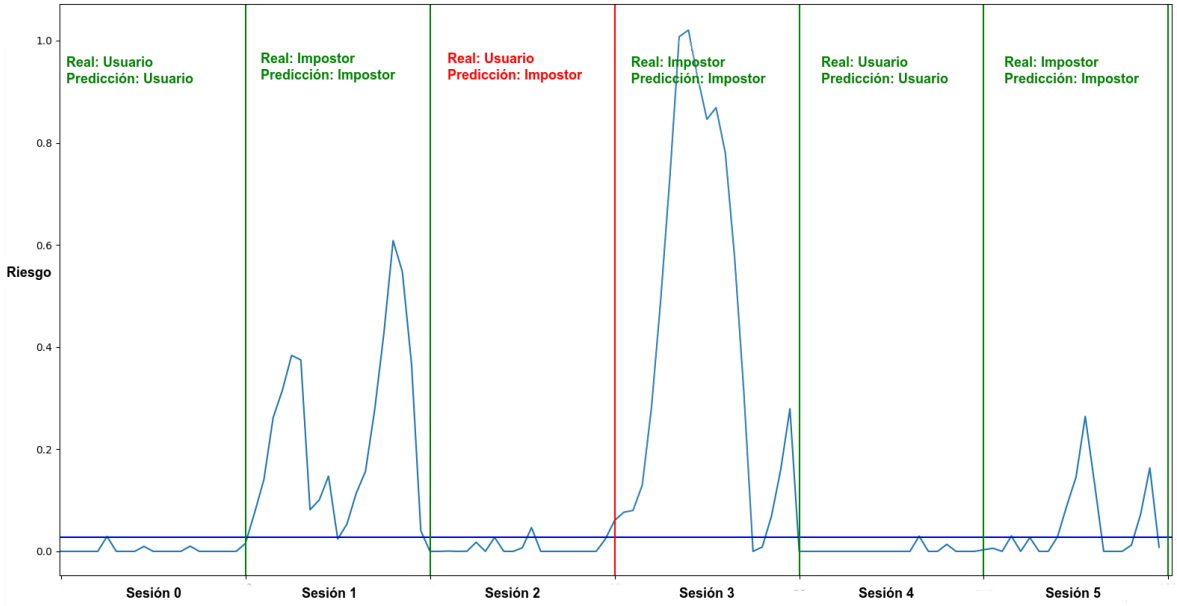


Figura 5.2: Valores del *buffer* utilizando las dinámicas de ratón para un usuario concreto.

Con el objetivo de contemplar un escenario real, se procede a agrupar la información en sesiones, al igual que en el caso del teclado. Los resultados se pueden observar en la Tabla 5.7 y la Tabla 5.8. En la Figura 5.2 se muestra un ejemplo de los valores del *buffer* para un usuario concreto. Centrando la atención en la sesión 2, se puede observar que el umbral óptimo definido en este caso es ligeramente restrictivo pues el modelo clasifica como impostor dicha sesión a pesar de los valores bajos del *buffer* en ese instante. Esto hace que el sistema sea más seguro a la hora de detectar impostores pero también repercute en que el sistema considere a un usuario genuino como impostor en más ocasiones.

En resumen, los mejores resultados para la RP específica propuesta en el caso de uso (chat)

Tamaño de sesión	N-grama	FAR	FRR	EER
5	1	0,352	0,357	0,373 (0,071)
5	2	0,376	0,388	0,337 (0,131)
5	3	0,376	0,360	0,381 (0,076)
5	6	0,366	0,386	0,296 (0,175)
10	1	0,237	0,225	0,242 (0,131)
10	2	0,292	0,306	0,279 (0,120)
10	3	0,329	0,325	0,327 (0,119)
10	6	0,285	0,381	0,278 (0,181)
20	1	0,151	0,151	0,143 (0,122)
20	2	0,232	0,232	0,220 (0,135)
20	3	0,274	0,246	0,240 (0,205)
20	6	0,258	0,366	0,248 (0,188)

Tabla 5.7: Resultados del modelo de clasificación OC-SVM utilizando el *buffer* temporal para las dinámicas de ratón divididas en sesiones utilizando en el conjunto de test: FAR, FRR y EER (desviación típica).

se obtienen modelando las dinámicas de teclado. En particular, los mejores resultados para este escenario se obtienen utilizando la distancia de Manhattan con un número de vecinos igual a 3 para el modelo de KNN y para sesiones de longitud 20. Por otro lado, para el caso de las dinámicas de ratón, los mejores resultados se obtienen utilizando una OC-SVM con n-gramas de longitud 1 y para sesiones de longitud 20. En ambos casos, tal y como ha quedado demostrado anteriormente, cuanto mayor es la duración de la sesión más información se considera y, por lo tanto, se obtienen mejores resultados.

5.1.5. Integración en los estándares de federación de identidades

La integración del flujo de trabajo propuesto depende del caso de uso en cuestión. Considerando los casos de uso vistos en la Sección 3.2, se precisan realizar modificaciones que se detallan a continuación y que se pueden ver esquematizadas en la Figura 5.3. El entorno de trabajo desarrollado implementa el estándar federado OIDC (ver Sección 5.1.1), sin embargo,

Tamaño de sesión	N-grama	FAR	FRR
5	1	0,353	0,355
5	2	0,381	0,410
5	3	0,430	0,426
5	6	0,425	0,451
10	1	0,307	0,329
10	2	0,341	0,372
10	3	0,370	0,380
10	6	0,386	0,462
20	1	0,256	0,277
20	2	0,295	0,356
20	3	0,355	0,385
20	6	0,363	0,368

Tabla 5.8: Resultados del modelo de clasificación OC-SVM utilizando el *buffer* temporal para las dinámicas de ratón divididas en sesiones utilizando en el conjunto de validación: FAR y FRR.

todos estos casos de uso son igualmente aplicables a cualquier estándar federado.

En el caso de uso 1 solo se requiere realizar una pequeña modificación en el flujo de autenticación. Esta modificación consiste en marcar como obligatorio el parámetro, actualmente considerado como opcional, *acr_values* de la *petición de autorización/autenticación*. De esta forma la RP debe utilizar este valor para especificar el LoA que requiere por parte del usuario final para completar el proceso de autenticación frente al IdP. Cuanto mayor sea el riesgo asociado a la petición de acceso, porque el modelo ha determinado una probabilidad alta de que sea un comportamiento anormal (impostor), mayor será el LoA requerido. De este mismo modo, el parámetro *acr_values* del *ID token* también se vuelve obligatorio. Así, el IdP debe informar a la RP del método utilizado para autenticar al usuario final cumpliendo con el LoA previamente determinado. Por la parte del IdP se han de contemplar diferentes métodos y procesos para autenticar a los usuarios. Todas estas consideraciones de implementación no afectan a la especificación federada, sino que han de solventarse a nivel de implementación de código por

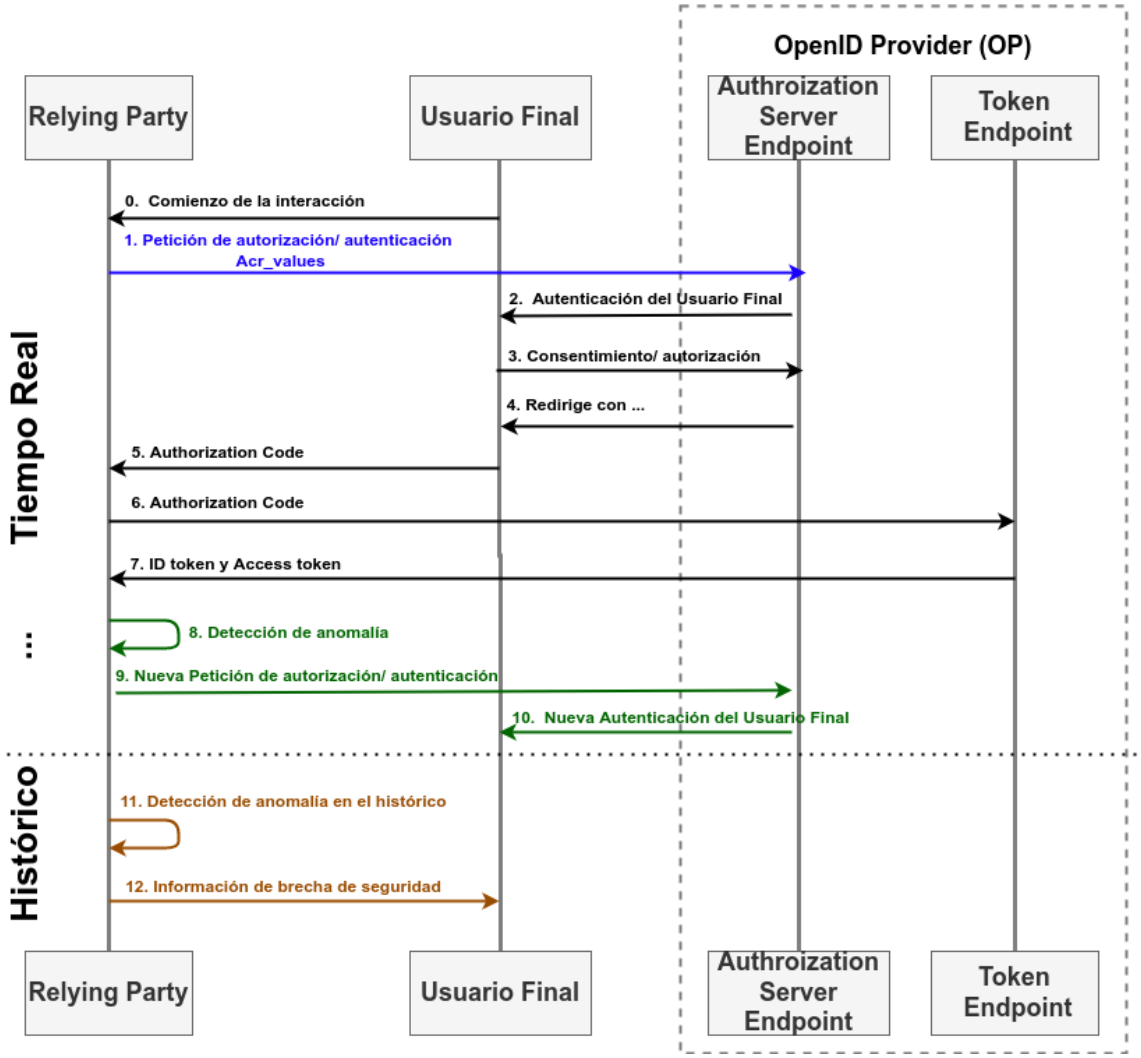


Figura 5.3: Modificaciones necesarias en el flujo de OpenId Connect para incorporar los casos de uso del flujo de trabajo propuesto. Caso de uso 1 en azul, caso de uso 2 en verde y caso de uso 3 en naranja.

parte del IdP y la RP.

En el caso de uso 2 la RPs debe en primer lugar invalidar los *tokens* asociados a la sesión activa. Para ello, debe enviar dichos *tokens* al *Token Revocation Endpoint* [151]. De esta forma, la sesión activa del usuario queda invalidada. Posteriormente, la RP debe iniciar nuevamente el proceso enviado una nueva petición de autorización/autenticación al *Authorization Server Endpoint*, el cual volverá nuevamente a solicitar al usuario los autenticadores pertinentes. Este caso de uso se puede combinar con el caso de uso 1, pues la RP al detectar la anomalía puede solicitar un LoA mayor aumentando así los niveles de seguridad. Para este caso de uso, no se

requiere ninguna modificación de los flujos de las especificaciones federadas. Se requiere una implementación a nivel de código por parte de la RP para forzar que se invaliden los *tokens* activos y se realice una nueva petición de ellos cada vez que los modelos de autenticación continua determinen que se ha producido una anomalía de comportamiento y, por consiguiente, pueda estar sucediendo un ataque (p. ej. un secuestro de sesión). Esto desencadena una nueva petición de autorización/autenticación inicializando nuevamente el proceso.

En el caso de uso 3 no se ven afectados los estándares tradicionales de gestión de identidades federados. Nuevamente, implica implementar nuevos procedimientos en la RP para poder gestionar las situaciones adversas y procedimientos para comunicárselo al resto de agentes implicados en caso de que suceda un ataque de suplantación de identidad.

En resumen, para poder integrar el flujo de trabajo propuesto en los principales esquemas de gestión de identidades, el único cambio que habría que realizar sobre ellos es marcar como obligatorios dos parámetros (uno en el *ID token* y otro en el *access token*) que actualmente se encuentran como opcionales. De esta forma, se puede corroborar que la integración del flujo de trabajo es muy simple y, por consiguiente, facilita de forma notoria la integración e implantación del mismo por parte de multitud de RPs.

5.1.6. Eficiencia y análisis de seguridad

En las secciones anteriores se ha mencionado que las técnicas de UEBA pueden ser utilizadas en múltiples casos de uso y en diversas RPs de diferente naturaleza y dominios, ejecutadas sobre plataformas heterogeneas. Es por esto que uno de los puntos clave a analizar para lograr una adopción del flujo de trabajo propuesto es la eficiencia del mismo. De este modo, el consumo de recursos computacionales ha de ser mínimo y las posibles latencias introducidas no deben afectar al funcionamiento normal de una RP (en el caso de uso 2), o a la experiencia de usuario (en el caso de uso 1).

En este sentido, el tiempo medio de ejecución a la hora de comprar dos vectores de comportamiento utilizando el KNN y la distancia de Manhattan es de $0{,}000416 \pm 0{,}000883$ nanosegundos ejecutando sobre la máquina descrita en la Sección 5.1.1. En el caso de la OC-SVM, el tiempo medio de ejecución al comparar dos vectores de comportamiento es de $0{,}000040 \pm 0{,}0000923$.

Tipo	**FAR**	**FRR**
Atributos estáticos	0	0
Dinámicas de teclado	0,104	0,082
Dinámicas de ratón	0,146	0,127

Tabla 5.9: Resultados para el ataque de robo de credenciales en el caso de uso 2.

Estos tiempos corroboran que las técnicas de UEBA propuestas no afectan para la implantación e integración del flujo de trabajo para ninguno de los casos de uso propuestos, ya sean para autenticación estática o autenticación continua.

El último grupo de experimentos realizados se centran en analizar la mejora en cuanto a seguridad que ofrecen las técnicas de UEBA en el caso de uso más complejo. Recordemos que el caso de uso 2 requiere autenticación continua. Para lograr esto, se han incluido dos tipos de ataques. En primer lugar una suplantación de identidad por medio de un robo de credenciales. Los participantes en el experimento tuvieron que hacer públicas sus credenciales, de tal manera que todos los usuarios podían autenticarse como cualquier otro usuario, suplantando así su identidad. En segundo lugar, un secuestro de sesión. Los participantes tenían acceso a los diferentes dispositivos del resto de usuarios y, por lo tanto, tenían vía libre para poder hacerse pasar por cualquiera de ellos, nuevamente suplantando su identidad. Para evaluar este experimento, se han utilizado los mejores modelos obtenidos en la Sección 5.1.4.

Los resultados para el primer ataque se pueden observar en la Tabla 5.9. Se ha necesitado una media de 12,200 ± 4,176 interacciones para detectar a un impostor utilizando las dinámicas de teclado. Por otro lado, una media de 18,800 ± 2,081 interacciones se han necesitado para detectar a un impostor utilizando las dinámicas de ratón. Tal y como se ha mencionado en la Sección 5.1.4, utilizar atributos estáticos implica una clasificación perfecta debido al bajo número de participantes.

Los resultados para el segundo ataque, es decir, para el secuestro de sesión utilizando el dispositivo de la víctima, se pueden observar en la Tabla 5.10. Para las dinámicas de teclado se han necesitado una media de 13,100 ± 4,223 interacciones para detectar a un impostor, mientras que para las dinámicas de ratón 18,100 ± 2,714. En este caso, al utilizar el dispositivo de la víctima, los atributos estáticos no son útiles para detectar a ningún impostor pues siempre van

Tipo	**FAR**	**FRR**
Atributos estáticos	1	1
Dinámicas de teclado	0,149	0,104
Dinámicas de ratón	0,175	0,168

Tabla 5.10: Resultados para el ataque de secuestro de sesión en el caso de uso 2.

a tener el mismo valor esperado.

Como se puede observar, los resultados obtenidos para el primer ataque son mejores que para el segundo, por lo que, se puede afirmar que los modelos propuestos son mejores a la hora de detectar ataques de robo de identidad que ataques de secuestro de sesión.

5.2. Evaluación del método de combinación de la información

Hasta el momento ha quedado demostrado que es posible implementar el flujo de trabajo propuesto y, por consiguiente, integrar modelos de análisis de comportamiento dentro de los principales estándares de gestión de identidades federados. Sin embargo, los modelos de análisis de comportamiento desarrollados hasta el momento, aun funcionales, poseen multitud de limitaciones, entre las que cabe destacar la eficacia de los modelos. De aquí en adelante se evalúa el modelo de combinación de información propuesto que supera estas limitaciones.

5.2.1. Evaluación del método de combinación en el conjunto de datos UEBA

En esta sección se va a analizar la evaluación del modelo de combinación de información propuesto en el conjunto de datos UEBA. En primer lugar, se realiza el proceso de extracción de características. Para ello, en el caso del teclado se utilizan las características H1, H2, HP, PP, HH y PH descritas en la Sección 5.1.3. En el caso de las dinámicas de ratón, cada digrafo se ha agrupado en ventanas temporales de 5 segundos. Se han considerado el número de interacciones contenidas en la ventana temporal, el tiempo total transcurrido, la distancia, velocidad, ángulo y velocidad angular. De este modo se han calculado las medidas de dispersión (mínimo, máximo, media y desviación típica) para cada característica considerada anteriormente (tiempo transcurrido, distancia, etc). De esta forma, el vector de comportamiento obtenido, para el caso del

ratón, contiene un total de 21 características (número de interacciones contenidas en la ventana temporal y cuatro medidas de dispersión de cinco variables). En resumen, para cada usuario se obtienen dos vectores (uno para cada fuente de información) que contiene la información de comportamiento.

Con el objetivo de entrenar y posteriormente poder evaluar el método, la información, ya procesada para cada usuario, se separa en tres conjuntos denominados: entrenamiento, test y validación. El conjunto de entrenamiento representa el 70 % de la información total del usuario objetivo, es decir, contiene únicamente muestras genuinas. Por otro lado, el conjunto de test está formado por muestras tanto genuinas como pertenecientes a usuarios impostores. Para el caso de las muestras genuinas, del 30 % restante (no seleccionado para entrenamiento) se selecciona un total del 60 % (es decir, un 18 % del total de muestras genuinas). Para el caso de las muestras de impostores, el mismo número de muestras que para el caso de las genuinas son aleatoriamente seleccionadas del resto de usuarios. Además, se asegura que esta información de los usuarios impostores contenga muestras de ambas fuentes de información (teclado y ratón). Esto significa que el resto de los usuarios actúan como impostores a la hora de evaluar a un usuario específico. Finalmente, el conjunto de validación está formado por el 12 % restante de muestras genuinas y por el mismo número de muestras de usuarios impostores siguiendo los mismos criterios que para el conjunto de test.

En primer lugar, para cada usuario, el conjunto de entrenamiento se utiliza para estandarizar el resto de los conjuntos de datos. Esto se logra calculando la media y desviación típica para cada una de las fuentes de información y aplicando la fórmula que sigue:

$$z_i = \frac{x_i - \bar{X}}{S_X}$$

donde z_i es el valor estandarizado para la muestra i, x_i es el valor original de la muestra i, $\bar{X}$ es la media del conjunto de muestras y S_X es la desviación típica.

Una vez estandarizados los datos, se utiliza el conjunto de entrenamiento para entrenar el algoritmo de RTE. El resultado de este proceso es la secuencia de símbolos que representa el comportamiento de un usuario. Esta secuencia, se separa en n-gramas utilizando el parámetro *NGL*, obteniendo múltiples secuencias que representan la información de comportamiento de un

usuario concreto. Estos n-gramas se utilizan para generar la matriz de distancias, comparándolas en pares utilizando el algoritmo de alineamiento de secuencias de ADN. A continuación, la matriz de distancias se utiliza para entrenar el algoritmo de DBSCAN. El resultado son los núcleos de comportamiento que representan el conocimiento adquirido para un usuario concreto y, por tanto, concluyendo el proceso de entrenamiento.

Posteriormente, las muestras de test (ya estandarizadas) se evalúan sobre el RTE previamente entrenado. De esta forma y al igual que en el caso anterior, se obtiene la secuencia de símbolos y se divide en n-gramas. Estos n-gramas se comparan uno a uno contra todos los n-gramas contenidos en los núcleos de comportamiento utilizando el algoritmo de alineamiento de secuencias de ADN. El resultado son múltiples vectores de distancias. Cada vector contiene la distancia de cada muestra a los núcleos de comportamiento. Estos vectores se utilizan para entrenar el modelo de riesgos, agrupándose en ventanas temporales en función del parámetro $WinSize$ y pudiendo fijar el umbral óptimo de decisión.

Por último, el conjunto de validación se utiliza para evaluar todo el proceso con muestras que el método propuesto no ha considerado nunca. Este conjunto de datos permite analizar la capacidad de generalización y la efectividad del método propuesto. Esto se debe a que, al evaluar este conjunto de datos, todos los modelos que se integran en el método, así como el umbral de decisión, han sido previamente entrenados y fijados.

Una vez obtenidos los tres conjuntos de datos, se evalúa el método teniendo en cuenta las tres casuísticas asociadas a las fuentes de información. Esto es, considerando únicamente el teclado, considerando únicamente el ratón y considerando la combinación de ambas fuentes de información. Los resultados se pueden observar en las Tablas 5.11, 5.12 y 5.13 respectivamente. Los resultados mostrados para cada métrica de evaluación se corresponden con la media obtenida para todos los usuarios.

Para todas las casuísticas de fuentes de información, los valores de EER, FAR y FRR disminuyen (mejoran los resultados) cuando los valores de los parámetros NGL o $WinSize$ aumentan. Esto se debe a que, cuanto mayor son los parámetros, más información se está considerando para realizar una predicción. De este modo, los valores máximos de EER para cada caso de uso (0,457, 0,438 y 0,397 respectivamente) se obtienen para los valores $NGL = 5$ y $WinSize = 5$. Por otro lado, se obtienen predicciones casi perfectas cuando se fijan los parámetros a los va-

NGL	WinSize	EER	FAR	FRR	FAR_val	FRR_val
5	5	0.457	0.457	0.457	0.442	0.397
5	10	0.423	0.423	0.423	0.439	0.392
5	20	0.356	0.356	0.366	0.420	0.287
5	50	0.268	0.239	0.282	0.323	0.207
5	100	**0.000**	**0.000**	0.048	0.250	0.062
10	5	0.384	0.381	0.386	0.303	0.453
10	10	0.343	0.333	0.343	0.189	0.414
10	20	0.276	0.276	0.286	0.122	0.250
10	50	0.059	0.059	0.074	0.015	0.167
10	100	**0.000**	**0.000**	0.047	**0.000**	0.186
20	5	0.304	0.301	0.305	0.297	0.141
20	10	0.226	0.228	0.226	0.230	0.030
20	20	0.145	0.145	0.147	0.163	0.019
20	50	0.050	0.051	0.050	0.043	0.027
20	100	**0.000**	**0.000**	0.011	0.003	0.026
30	5	0.267	0.267	0.270	0.236	0.254
30	10	0.177	0.175	0.178	0.142	0.097
30	20	0.091	0.090	0.091	0.038	0.038
30	50	0.006	**0.000**	**0.008**	**0.000**	**0.002**
30	100	**0.000**	**0.000**	0.012	**0.000**	0.009

Tabla 5.11: Resultados obtenidos para las dinámicas de teclado utilizando el método de combinación en el conjunto de datos UEBA. El sufijo _val se refiere a conjunto de validación. Los mejores resultados obtenidos para cada métrica se muestran en negrita.

lores $NGL = 30$ y $WinSize = 100$. Unicamente fijando el parámetro $WinsSize$ a 100, se obtienen buenos resultados independientemente del valor de *NGL*. Sin embargo, estos resultados se vuelven cada vez más robustos a medida que aumenta el parámetro *NGL* para el conjunto de validación.

Los mejores resultados para test y validación simultáneamente se obtienen para el caso de uso de combinación de la información. Esto corrobora que combinar información de múltiples fuentes ayuda a mejorar la eficacia de los sistemas de autenticación continua que solo consideran

NGL	WinSize	EER	FAR	FRR	FAR_val	FRR_val
5	5	0.438	0.438	0.438	0.308	0.390
5	10	0.392	0.392	0.396	0.234	0.359
5	20	0.347	0.343	0.351	0.146	0.280
5	50	0.265	0.265	0.265	0.031	0.076
5	100	0.034	0.024	0.065	0.506	0.513
10	5	0.354	0.354	0.358	0.329	0.306
10	10	0.306	0.302	0.306	0.261	0.176
10	20	0.202	0.198	0.202	0.129	0.129
10	50	0.090	0.090	0.094	8.000	**0.000**
10	100	**0.000**	**0.000**	0.031	0.500	0.500
20	5	0.147	0.139	0.151	0.133	0.223
20	10	0.077	0.073	0.081	0.112	0.050
20	20	0.034	0.030	0.034	0.053	**0.000**
20	50	0.000	**0.000**	0.016	**0.000**	**0.000**
20	100	**0.000**	**0.000**	0.026	**0.000**	**0.000**
30	5	0.090	0.090	0.090	0.080	0.049
30	10	0.025	0.025	0.025	0.013	0.006
30	20	0.004	0.000	0.012	**0.000**	0.011
30	50	**0.000**	**0.000**	**0.007**	**0.000**	0.041
30	100	**0.000**	**0.000**	0.011	**0.000**	0.092

Tabla 5.12: Resultados obtenidos para las dinámicas de ratón utilizando el método de combinación en el conjunto de datos UEBA. El sufijo _val se refiere a conjunto de validación. Los mejores resultados obtenidos para cada métrica se muestran en negrita.

una única fuente.

5.2.2. Evaluación del método de combinación en el conjunto de datos TWOS

En esta sección se evalúa el método de combinación de información propuesto sobre el conjunto de datos TWOS. El conjunto de datos TWOS [105] se recogió durante la competición organizada por la Universidad de Singapur de tecnología y diseño en marzo de 2017. Los datos provienen de seis fuentes de información: teclado, ratón, tráfico de red, registros *Simple Mail*

NGL	WinSize	EER	FAR	FRR	FAR_val	FRR_val
5	5	0.397	0.397	0.397	0.342	0.486
5	10	0.364	0.364	0.366	0.294	0.501
5	20	0.320	0.317	0.320	0.273	0.484
5	50	0.229	0.223	0.232	0.131	0.489
5	100	0.112	0.115	0.112	0.049	0.448
10	5	0.289	0.284	0.292	0.322	0.329
10	10	0.219	0.222	0.219	0.284	0.296
10	20	0.141	0.138	0.144	0.154	0.263
10	50	0.029	0.024	0.032	0.019	0.100
10	100	**0.000**	**0.000**	0.007	**0.000**	0.019
20	5	0.167	0.167	0.169	0.117	0.146
20	10	0.080	0.075	0.085	0.052	0.070
20	20	0.006	0.006	0.010	0.016	0.015
20	50	**0.000**	**0.000**	0.006	**0.000**	0.008
20	100	**0.000**	**0.000**	0.008	**0.000**	0.011
30	5	0.128	0.128	0.131	0.217	0.116
30	10	0.054	0.068	0.054	0.165	0.030
30	20	0.020	0.014	0.023	0.098	0.001
30	50	**0.000**	**0.000**	0.006	**0.000**	0.008
30	100	**0.000**	**0.000**	**0.004**	**0.000**	**0.000**

Tabla 5.13: Resultados obtenidos para la combinación de información en el conjunto de datos UEBA. El sufijo _val se refiere a conjunto de validación. Los mejores resultados obtenidos para cada métrica se muestran en negrita.

Transfer Protocol (SMTP), información de inicio de sesión e información de la máquina anfitriona. Además, posee información relacionada a un test psicológico de personalidad realizado a cada uno de los participantes. En total, veinticuatro usuarios participaron en la recogida de datos durante un periodo de cinco días.

Al igual que para el conjunto de datos UEBA, se ha seleccionado la información que proviene de las fuentes de información del teclado y del ratón. Esta información se ha procesado y dividido en los conjuntos de entrenamiento, test y validación siguiendo las mismas directrices

NGL	WinSize	EER	FAR	FRR	FAR_val	FRR_val
5	5	0.414	0.413	0.415	0.407	0.414
5	10	0.386	0.386	0.386	0.375	0.384
5	20	0.341	0.343	0.341	0.346	0.353
5	50	0.269	0.269	0.269	0.268	0.233
5	100	0.214	0.219	0.214	0.189	0.131
10	5	0.362	0.362	0.362	0.380	0.382
10	10	0.337	0.335	0.339	0.340	0.320
10	20	0.305	0.304	0.307	0.282	0.237
10	50	0.217	0.217	0.217	0.180	0.126
10	100	0.133	0.133	0.134	0.044	0.052
20	5	0.317	0.318	0.317	0.346	0.302
20	10	0.259	0.257	0.259	0.305	0.245
20	20	0.189	0.189	0.191	0.227	0.172
20	50	0.080	0.078	0.082	0.110	0.072
20	100	0.008	0.008	0.008	0.018	0.019
30	5	0.299	0.299	0.299	0.293	0.288
30	10	0.231	0.231	0.232	0.222	0.224
30	20	0.156	0.156	0.156	0.139	0.139
30	50	0.055	0.054	0.055	0.060	0.052
30	100	**0.007**	**0.007**	**0.007**	**0.000**	**0.015**

Tabla 5.14: Resultados obtenidos para las dinámicas de teclado utilizando el método de combinación en TWOS. El sufijo _val se refiere a conjunto de validación. Los mejores resultados obtenidos para cada métrica se muestran en negrita.

marcadas en el caso anterior. En definitiva, se han extraído los núcleos de comportamiento y se ha entrenado el modelo de riesgos con el objetivo de poder evaluar los tres casos de uso.

Los resultados se pueden observar en las Tablas 5.14, 5.15 y 5.16, respectivamente. Al igual que en la sección anterior, se han considerado todas las posibles combinaciones de los parámetros *NGL* y *WinSize*. Los resultados obtenidos para cada métrica se corresponden con la media para todos los usuarios.

Los resultados tanto para cada fuente de información por separado, como para la combi-

NGL	WinSize	EER	FAR	FRR	FAR_val	FRR_val
5	5	0.433	0.433	0.434	0.400	0.432
5	10	0.404	0.403	0.404	0.375	0.400
5	20	0.371	0.370	0.372	0.313	0.351
5	50	0.317	0.316	0.318	0.252	0.286
5	100	0.266	0.265	0.266	0.205	0.260
10	5	0.420	0.419	0.420	0.406	0.394
10	10	0.392	0.392	0.393	0.369	0.356
10	20	0.355	0.354	0.355	0.290	0.315
10	50	0.285	0.285	0.285	0.263	0.209
10	100	0.180	0.176	0.180	0.250	0.126
20	5	0.377	0.376	0.379	0.362	0.388
20	10	0.336	0.335	0.337	0.311	0.347
20	20	0.286	0.285	0.288	0.255	0.287
20	50	0.174	0.173	0.174	0.153	0.211
20	100	0.088	0.086	0.101	0.086	0.117
30	5	0.359	0.357	0.361	0.338	0.352
30	10	0.313	0.312	0.314	0.308	0.302
30	20	0.250	0.250	0.251	0.237	0.243
30	50	0.136	0.136	0.137	0.141	0.146
30	100	**0.051**	**0.049**	**0.089**	**0.047**	**0.073**

Tabla 5.15: Resultados obtenidos para las dinámicas de ratón utilizando el método de combinación en TWOS. El sufijo _val se refiere a conjunto de validación. Los mejores resultados obtenidos para cada métrica se muestran en negrita.

nación de ambas fuentes son óptimos. Al igual que en la sección anterior, los resultados del método propuesto mejoran cuando se considera más información, es decir, cuando los valores de los parámetros *NGL* y *WinSize* aumentan. En el caso de uso del teclado, los valores de EER van desde 0,414 hasta 0,007, obteniendo valores de FAR y FRR en validación de 0,407 y 0,414 respectivamente para la peor combinación de parámetros y, 0,000 y 0,015 para la mejor combinación de los mismos. Por otro lado, en el caso de uso del ratón, los resultados son ligeramente peores. Así, los valores de EER van desde 0,433 hasta 0,051, obteniendo valores de FAR y FRR para validación de 0,400 y 0,432 respectivamente para la peor combinación de parámetros y,

NGL	WinSize	EER	FAR	FRR	FAR_val	FRR_val
5	5	0.407	0.407	0.407	0.409	0.410
5	10	0.380	0.380	0.380	0.381	0.375
5	20	0.335	0.335	0.335	0.343	0.321
5	50	0.257	0.257	0.257	0.270	0.228
5	100	0.177	0.177	0.176	0.197	0.150
10	5	0.359	0.359	0.359	0.371	0.349
10	10	0.320	0.319	0.319	0.335	0.307
10	20	0.263	0.262	0.263	0.289	0.245
10	50	0.169	0.168	0.169	0.213	0.159
10	100	0.082	0.082	0.082	0.140	0.099
20	5	0.295	0.295	0.296	0.310	0.306
20	10	0.226	0.226	0.226	0.267	0.230
20	20	0.156	0.156	0.157	0.206	0.157
20	50	0.066	0.065	0.067	0.098	0.080
20	100	0.022	0.022	0.022	0.041	0.018
30	5	0.268	0.268	0.268	0.282	0.285
30	10	0.201	0.200	0.201	0.227	0.211
30	20	0.121	0.122	0.121	0.168	0.110
30	50	0.038	0.037	0.038	0.076	0.020
30	100	**0.006**	**0.006**	**0.006**	**0.014**	**0.004**

Tabla 5.16: Resultados obtenidos para la combinación de información en TWOS. El sufijo _val se refiere a conjunto de validación. Los mejores resultados obtenidos para cada métrica se muestran en negrita.

0,047 y 0,073 para la mejor combinación de los mismos. Los mejores resultados de forma global se obtienen para la combinación de ambas fuentes de información. En este caso, los valores de EER se encuentran en el rango 0,407 y 0,006 llegando a valores de FAR y FRR en el conjunto de validación de 0,409 y 0,410 respectivamente para la peor combinación de parámetros y de 0,0014 y 0,004 para la mejor combinación.

En el caso particular del teclado, cada símbolo de la secuencia (digrafo) tiene un tiempo medio de ejecución de 0,227 segundos. En el caso de las dinámicas de movimiento del ratón,

cada símbolo de la secuencia representa una ventana temporal de 5 segundos. Esto significa que, considerando un $NGL = 10$, los n-gramas contienen aproximadamente 2,27 segundos de información para las dinámicas de teclado y 50 segundos de información para las dinámicas de ratón. Del mismo modo, seleccionando un $WinSize = 100$ para realizar una predicción se considera 22,7 segundos de información histórica para el teclado y 500 segundos para el ratón. En el caso de la combinación de información, una secuencia promedio para todos los usuarios contiene un 73 % de símbolos pertenecientes al teclado, mientras que un 27 % pertenecen al ratón. Considerando un $NGL = 10$, el vector promedio contendrá por lo tanto 7 símbolos del teclado y 3 del ratón. Esto se traduce en que un vector promedio contiene de media 16,589 segundos de información. Tomando un $WinSize = 100$, se considerará de media un total de 151,571 segundos de información histórica para realizar una predicción. A pesar de que se consideré toda esta información histórica, las predicciones se realizan para cada interacción del usuario, es decir, para cada nuevo símbolo independientemente de si proviene del teclado o del ratón. De este modo, las predicciones se realizan de media cada 0,227 segundos para el teclado y 5 segundos para el ratón. Esto se debe a que, cuando se evalúan los n-gramas, la secuencia adyacente es exactamente igual a la secuencia anterior a excepción del primer y último símbolo (ver Figura 4.9).

Finalmente, los algoritmos de SVM y RF se han seleccionado como algoritmos representativos del estado del arte frente a los que comparar el método propuesto. De este modo, para entrenar estos algoritmos se ha realizado un preprocesamiento de los datos para que consideren los parámetros *NGL* y *Winsize*, tal como lo hace el método propuesto. En primer lugar, los datos en bruto se han agrupado en subgrupos acorde al parámetro *NGL*. Posteriormente, cada uno de estos modelos ha sido entrenado utilizando una búsqueda en cuadrículas para fijar los hiperparámetros. Cada modelo devuelve la probabilidad de pertenecer a la clase genuina y a la clase impostora. Para obtener el riesgo, se selecciona la probabilidad devuelta de pertenecer a la clase impostora. Estas probabilidades se agrupan en ventanas temporales utilizando el parámetro $WinSize$. Posteriormente, se aplica MME para suavizar la curva de riesgo siguiendo las directrices marcadas por el método propuesto.

En la Tabla 5.17 se comparan los resultados obtenidos para el método propuesto frente a los algoritmos de SVM y RF, así como con otras propuestas del estado del arte. Los resultados

mostrados para [96] son los obtenidos para el algoritmo de 2D-CNN. Los resultados mostrados para [104] son los obtenidos para el algoritmo de SVM.

Los resultados mostrados para el método propuesto son los obtenidos para el conjunto de parámetros tal que, los resultados en validación son comparables a los resultados del resto de propuestas o algoritmos. Esto es, los valores de NGL y $WinSize$ son (30,100), (20,100), (5,100) y (20,50), respectivamente. Estos mismos parámetros han sido utilizados para entrenar los algoritmos de SVM y RF. El primer bloque de resultados se corresponde con el uso de dinámicas de teclado, el segundo bloque se corresponde con el uso de dinámicas de ratón y el tercer bloque se corresponde con la combinación de la información de teclado y de ratón. Finalmente, en el cuarto bloque se compara la combinación de información de la propuesta [104] en la que se utiliza información de contexto. Puesto que esta información no está disponible públicamente y pertenece a un conjunto de datos externo y privado, en este bloque se demuestra que utilizando el método propuesto se pueden obtener resultados comprables e incluso mejores en algunos ámbitos, única y exclusivamente utilizando información del teclado y del ratón.

Comparando los resultados de [96] con los obtenidos en la Tabla 5.15, los valores de $NGL = 30$ y $WinSize = 50$ muestran resultados similares para el método propuesto. Sin embargo, si se utiliza más información ($NGL = 20$ y $WinSize = 100$ o $NGL = 30$ y $WinSize = 100$) los resultados obtenidos por el método mejoran considerablemente. Lo mismo sucede en la combinación de información para la propuesta de [104], donde los resultados pueden ser igualados fijando los valores de NGL y $WinSize$ a (5,100), (10,50), (20,20) o (30,20) respectivamente (ver Tabla 5.16). Cuando en esa propuesta se considera información externa de contexto, los resultados pueden ser igualados o incluso mejorados utilizando únicamente información de teclado y de ratón por el método aquí propuesto fijando los parámetros a (20,50), (20,100), (30,50) o (20,100). Por último, considerando el EER, los algoritmos SVM y RF obtienen resultados satisfactorios. Sin embargo, ambos métodos son superados por el método propuesto en igualdad de condiciones.

Trabajo	FI	EER	FAR_val	FRR_val	$F1^{-}$_val	Exac_val	VPN_val	Espec_val
SVM	T	0.136	0.158	0.151	0.850	0.845	0.858	0.842
RF	T	0.084	0.093	0.174	0.860	0.865	0.819	0.907
Our	T	**0.007**	**0.000**	**0.015**	**0.968**	**0.979**	**0.945**	**0.993**
SVM	R	0.171	0.180	**0.063**	0.882	0.877	**0.955**	0.820
RF	R	0.109	0.141	0.081	0.890	0.888	0.925	0.859
[96]	R	0.130	0.136	0.149	-	-	-	-
Our	R	**0.088**	**0.086**	0.117	**0.900**	**0.909**	0.888	**0.914**
RF+RF	T + R	0.180	0.228	0.169	0.814	0.800	0.861	0.772
[104]	T + R	-	-	-	0.806	0.751	**0.932**	0.710
Our	T + R	**0.177**	**0.197**	**0.150**	**0.828**	**0.826**	0.856	**0.803**
RF+RF	T + R	0.121	0.126	0.152	0.860	0.860	0.848	0.874
[104]	T + R + C	-	-	-	**0.914**	**0.915**	0.874	**0.912**
Our	T +R	**0.066**	**0.098**	**0.080**	0.912	**0.915**	**0.921**	0.902

Tabla 5.17: Comparación de los resultados obtenidos con otras propuestas y algoritmos del estado del arte. *T* y *M* representan las dinámicas de teclado y de ratón respectivamente. *C* se refiere a información de contexto recogida de un conjunto de datos externos. RF+RF representa un modelo de combinación de información en el que se utiliza el algoritmo de RF de forma independiente para cada fuente de datos. FI se refiere a fuente de información. Exac se refiere a exactitud. Espec representa la especificación. El sufijo _val se refiere a conjunto de validación. Los mejores resultados obtenidos para métrica en cada bloque de experimentos se muestran en negrita.

Capítulo 6

Conclusiones

En este capítulo, en primer lugar, se exponen las conclusiones generales extraídas tras la realización de esta tesis doctoral. En segundo lugar, se exponen las conclusiones específicas asociadas a cada uno de los objetivos que se detallaron en el primer capítulo. En tercer lugar, se presentan las posibles líneas de investigación futura que han surgido. Finalmente, se detallan las principales contribuciones y publicaciones científicas que han derivado de la elaboración de la presente tesis.

6.1. Conclusiones generales

El presente trabajo de tesis doctoral tiene fijados dos objetivos generales para validar la hipótesis de partida. El primer objetivo se centra en el diseño de un flujo de trabajo que permita la integración de los métodos análisis de comportamiento en los principales estándares de gestión de identidades federada. Este objetivo se alcanzó en el Capítulo 3 del presente documento. Por otro lado, el segundo objetivo se centra en el diseño de un método de análisis de comportamiento capaz de combinar información de múltiples fuentes de datos. Este objetivo también se ha podido alcanzar en el Capítulo 4 del presente documento. Cabe destacar, que la metodología fijada al comienzo de este documento ha sido fundamental y determinante en la consecución de dichos objetivos.

Por lo expuesto anteriormente, la principal conclusión del presente trabajo de tesis doctoral,

puesto que los dos objetivos generales se han podido alcanzar, es que la hipótesis de partida ha quedado demostrada. Esto significa, por lo tanto, que es posible mejorar los niveles de seguridad proporcionados por los estándares de gestión de identidades federados utilizando para ello técnicas de análisis de comportamiento de los usuarios. Y además, que es posible mejorar la eficacia de los modelos de análisis de comportamiento del estado del arte utilizando técnicas de combinación de la información.

6.2. Conclusiones específicas

6.2.1. Conclusiones del flujo de trabajo

Las principales conclusiones extraídas en relación con el flujo de trabajo propuesto se pueden resumir en:

- El flujo de trabajo propuesto aumenta los niveles de seguridad de los estándares actuales que siguen el modelo federado (en relación con los ataques de suplantación de identidad) y puede ser fácilmente implementado por cualquier RP siguiendo las directrices proporcionadas.

- La elección de los atributos que las RPs han de emplear para generar una huella digital dependerá, en gran medida, del propio dominio y caso de uso particular. Sin embargo, los perfiles de seguridad propuestos son una referencia útil como punto de partida.

- Las RPs deben tener un equilibrio entre eficacia y usabilidad ya que la introducción de técnicas de análisis de comportamiento en algunos casos de uso puede añadir latencias significativas a los flujos de identificación y autenticación tradicionales.

- El flujo de trabajo propuesto es un claro ejemplo de caso de uso en el que se aumentan los riesgos para la privacidad con el objetivo de mitigar los riesgos para la seguridad. Existen multitud de dominios en los que los usuarios seguramente estén dispuestos a asumir este coste, pero otros, considerados “banales” en los que probablemente no sea así. Por lo tanto las RPs deben informar del tratamiento de datos que van a realizar (siguiendo la regulación vigente y el principio de responsabilidad proactiva), así como de la información

que van a recopilar para generar la huella digital. Es el usuario final el que debe proporcionar su consentimiento explícito e informado, debe tener siempre el poder para decidir si quiere que se le apliquen o no este tipo de mecanismos de análisis de comportamiento.

- La integración del flujo de trabajo propuesto en los flujos actuales de identificación y autenticación puede llevarse a cabo utilizando los propios mecanismos especificados en los estándares actuales. Esto implica que la integración puede realizarse en las etapas de implementación y, por lo tanto, es relativamente sencilla de realizar ya que no exige ningún cambio o modificación en las especificaciones vigentes (y que ya están ampliamente extendidas en entornos de producción).

6.2.2. Conclusiones del método de combinación de información de comportamientos

Las principales conclusiones relacionadas con este objetivo de la investigación realizada se pueden resumir en:

- Los RTE han demostrado ser una técnica prometedora para representar información de comportamientos. Esto se debe a que la discretización de la información se logra con una pérdida de información mínima.
- La representación de la información de comportamientos en forma de secuencia de caracteres o símbolos permite realizar la combinación de información a nivel de características de forma efectiva.
- El uso de técnicas de alineamiento de ADN permite comparar de forma precisa dinámicas de comportamiento, lo que repercute de forma directa en la eficacia de los métodos de detección de anomalías de comportamiento.
- El uso de técnicas de *clustering* basadas en densidades permite acotar la información utilizada para entrenar los modelos de análisis de comportamientos y para su posterior fase de predicción. Esto consigue una reducción en las latencias añadidas a los flujos de identificación y autenticación que integren el método propuesto.
- El modelo basado en riesgos permite detectar anomalías de comportamiento a lo largo del tiempo. Este modelo puede adaptarse acorde a los requisitos de los usuarios para ser

más permisivo o restrictivo dependiendo del dominio de aplicación y del caso de uso.

- El método propuesto es más eficaz detectando anomalías de comportamiento que otras propuestas específicas encontradas en la literatura y que otros algoritmos del estado del arte como RF y SVM. Esto se ha demostrado al evaluarlo sobre los conjuntos de datos de UEBA y TWOS. En el caso del conjunto de datos de TWOS, el método propuesto siempre obtiene EER menores para todas las casuísticas de evaluación. Estas mejoras van desde el 48,5 % hasta el 1,6 % dependiendo del algoritmo o método comparado y las fuentes de información consideradas.

6.3. Líneas de investigación futuras

La elaboración de esta tesis doctoral ha permitido identificar un conjunto de líneas de investigación que en el futuro pueden complementar, mejorar o extender este trabajo de investigación. Estas son las más interesantes o prometedoras:

- Las técnicas de análisis de comportamientos pueden ser integradas e implementadas dentro de un estándar federado, no solo en la parte que concierne a la RP (como se ha expuesto en esta tesis), sino en la parte del IdP. De esta forma, existen multitud de casos de uso como:

    1. Autenticación continua: el propio IdP puede utilizar la información de las peticiones de autenticación y/o autorización para generar un método de análisis de comportamiento de tal manera que, analizando datos históricos, determine el riesgo en tiempo real de las interacciones que está realizando un usuario. De esta manera, el IdP puede anular, por ejemplo, la *cookie* de sesión si así lo sugiere el método de análisis de comportamientos.
    2. Detección de suplantación o fraude: el IdP puede utilizar el histórico de interacciones del usuario para determinar, a posteriori, si se ha producido una suplantación de identidad o fraude. La gran ventaja de los modelos de análisis de comportamientos que se utilizarían con este objetivo es que al poder dejar de lado la eficiencia (no importa que la latencia sea alta porque no se ejecuta el modelo en tiempo real), se

pueden generar modelos que primen la eficacia y, por lo tanto, detectar con mayor precisión las posibles brechas de seguridad.

3. Registro dinámico de dispositivos y usuarios: normalmente el registro de usuarios y dispositivos se realiza de forma manual y remota en el IdP. Sin embargo, hoy en día con el avance del IoT, existen multitud de sensores que se deben registrar en un IdP (fase de *enrollment* o de *on-boarding*). Los métodos de análisis de comportamientos pueden utilizarse para completar esta fase de registro de forma eficiente o automática teniendo en cuenta el comportamiento observado durante un periodo de tiempo, o incluso analizando el comportamiento de dispositivos vecinos o similares en conjunto.

- El método de combinación de información propuesto se basa en discretizar la información de comportamiento proveniente de múltiples fuentes de datos heterogéneas utilizando una técnica novedosa de SAX basada en el uso de RTEs. Esta técnica ha demostrado empíricamente ser efectiva para este propósito, sin embargo, el uso de técnicas de lógica difusa o NNs han demostrado ser bastante efectiva a la hora de generar *embeddings* en otros ámbitos. Es por esto que una línea de investigación futura es la evaluación de estos tipos de algoritmos, en cuanto a eficiencia y eficacia, en el ámbito del análisis de comportamientos.

- La comparación de secuencias de caracteres o símbolos se realiza en esta tesis por medio de técnicas de alineamiento de secuencias de ADN. Sin embargo, hoy en día existen multitud de métricas de similitud entre este tipo de secuencias. Una posible línea de investigación futura sería evaluar otras métricas o incluso, realizar una combinación de matrices de similitud obtenidas utilizando diferentes métricas de forma independiente. Esto último se puede materializar mediante una combinación de *kernels* con el objetivo de ponderar las diferentes virtudes o fallos de cada una de ellas por separado.

- El método de combinación propuesto en esta tesis es escalable, es decir, se podrían considerar nuevas fuentes de información de forma sencilla. Esto permite añadir nuevos datos a los modelos que permiten identificar las anomalías en el comportamiento de los usuarios y por lo tanto, potenciales brechas de seguridad. Sería conveniente evaluarlo en un entorno IoT, donde los recursos computacionales disponibles en los dispositivos son limi-

tados. De este modo, para escalar el método y combinar más fuentes de información ser debería recurrir al *edge computing* para distribuir el procesamiento adicional requerido.

6.4. Principales contribuciones y publicaciones derivadas de la tesis

Las principales contribuciones del presente trabajo de tesis doctoral se resumen en:

1. Análisis en profundidad de los principales trabajos en el ámbito del análisis de comportamiento.
2. Flujo de trabajo para la integración de los modelos de análisis de comportamiento en los estándares de gestión de identidades federados.
3. Modelo de análisis de comportamiento que combina información recogida de fuentes de datos heterogéneas.
4. Nuevo conjunto de datos que contiene dinámicas de comportamiento de usuarios.

Las publicaciones realizadas durante el desarrollo de esta tesis doctoral en relación con estas contribuciones son las siguientes:

1. A. G. Martín, I. M. de Diego, A. Fernández-Isabel, M. Beltrán y Fernández, R. R. "Combining user behavioural information at the feature level to enhance continuous authentication systems" . Knowledge-Based Systems, 108544, 2022
2. A. G. Martín, M. Beltrán, A. Fernández-Isabel e I. M. de Diego, "An approach to detect user behaviour anomalies within identity federations" Computers & Security, vol. 108, p. 102 356, 2021.
3. A. G. Martín, A. Fernández-Isabel, I. M. de Diego y M. Beltrán, "A survey for user behavior analysis based on machine learning techniques: current models and applications," Applied Intelligence, pp. 1-27, 2021.
4. A. G. Martín, M. Beltrán, A. Fernández-Isabel e I. M. de Diego, "Keystroke and Mouse Dynamics for UEBA Dataset", Mendeley Data, v2, 2020

5. A. G. Martín, M. Beltrán, “Mejora de la seguridad de esquemas de gestión de identidades federados mediante técnicas de User Behaviour Analytics”, V Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2019), pp. 159-166,2019

Cada una de las publicaciones aquí presentadas se corresponden con un capítulo del presente trabajo de tesis doctoral y con una o varias contribuciones del mismo. De este modo, el Capítulo 1 se corresponde con la publicación [18]. En él se presentan los principales objetivos e ideas que surgieron al comienzo de la realización de la presente tesis. En la publicación [4] se aborda el estado del arte presentado en el Capítulo 2. Esta publicación se corresponde con la primera contribución. El Capítulo 3 se corresponde con la publicación [110] y en él se presentan la segunda y tercera contribución, es decir, el flujo de trabajo para lograr la integración en los estándares federados y el conjunto de datos que incluye dinámicas de comportamiento. El Capítulo 4 se corresponde con la publicación [130], y en él se presenta la cuarta contribución, es decir, el modelo de análisis de comportamiento que combina información. Finalmente, los experimentos abordados en el Capítulo 5, y las conclusiones obtenidas en el Capítulo 6 se ven reflejadas en todas estas publicaciones de forma simultánea.

Apéndice A

Glosario de acrónimos

ABAC *Attribute-Based Access Control*.

AC *Ant Colony*.

ACL *Access Control Lists*.

AMF Autenticación de Múltiples Factores.

API *Application Programming Interface*.

BN *Bayesian Network*.

COT *Circle of Trust*.

CSRF *Cross Site Request Forgery*.

DAC *Discretionary Access Control*.

DBSCAN *Density-Based Spatial Clustering of Applications with Noise*.

DT *Decision Tree*.

EER *Equal Error Rate*.

EU *End User*.

FAPI *Financial-grade API*.

FAR *False Acceptance Rate*.

FN Falsos Negativos.

FP Falsos Positivos.

FRR *False Rejection Rate*.

HAT *Hoeffding Adaptive Trees*.

HMM *Hidden Markov Model*.

HRU Harrison, Ruzzo y Ullman.

IAAA Identificación, Autenticación, Autorización y Auditoria.

IdP *Identity Provider*.

IoT *Internet of Things*.

ITAD *Instance-based Tail Area Density*.

KDE *Kernel Density Estimation*.

KNN *K-Nearest Neighbors*.

LDAP *Lightweight Directory Access Protocol*.

LoA *Level of Assurance*.

LOPD Ley Orgánica de Protección de Datos de Carácter Personal.

MAC *Mandatory Access Control*.

MC Modelo de Confianza.

MKL *Multi-kernel Learning Method*.

MME Media Móvil Exponencial.

NB *Naïve Bayes*.

NGL *N-Gram Length*.

NN *Neural Network*.

OC-SVM *One-Class SVM*.

OIDC *OpenId Connect*.

OP *OpenID Provider*.

PAA *Piecewise Aggregate Approximation*.

PADTW *Progress-Adjusted Dynamic Time Wraping*.

PCA *Principal Component Analysis*.

PLC *Parametric Linear Combination*.

RADIUS *Remote authentication dial in user service*.

RBAC *Role-Based Access Control*.

RF *Random Forest*.

RGPD Reglamento General de Protección de Datos.

RP *Relying Party*.

RTE *Random Trees Embedding*.

SAML *Security Assertion Markup Language*.

SAX *Symbolic Aggregate approximation*.

SDK *Software Development Kit*.

SMTP *Simple Mail Transfer Protocol*.

SP *Service Provider*.

SSO *Single Sign On*.

SVM *Support Vector Machine*.

TANB *Tree Augmented Naïve Bayes*.

TLS *Transport Layer Security*.

TWOS *The Wolf of SUTD*.

UEBA *User and Entity Behavior Analysis*.

VN Verdaderos Negativos.

VP Verdaderos Positivos.

VPN Valor Predictivo Negativo.

WAR *Web application Resource*.

XACML *eXtensible Access Control Markup Language*.

Bibliografía

[1] J. Pato y O. C. Center, "Identity management: Setting context," *Hewlett-Packard, Cambridge, MA*, 2003.

[2] B. F. Skinner, *Science and human behavior*, 92904. Simon y Schuster, 1953.

[3] M. Sidman, *Tactics of scientific research*. Basic Books, Incorporated, Pub., 1960.

[4] A. G. Martín, A. Fernández-Isabel, I. M. de Diego y M. Beltrán, "A survey for user behavior analysis based on machine learning techniques: current models and applications," *Applied Intelligence*, pp. 1-27, 2021.

[5] E. Gurarie, C. Bracis, M. Delgado, T. D. Meckley, I. Kojola y C. M. Wagner, "What is the animal doing? Tools for exploring behavioural structure in animal movements," *Journal of Animal Ecology*, vol. 85, n.º 1, pp. 69-84, 2016.

[6] J. Pacheco y S. Hariri, "Anomaly behavior analysis for IoT sensors," *Transactions on Emerging Telecommunications Technologies*, vol. 29, n.º 4, pp. 1-15, 2018.

[7] M. Pantic, A. Pentland, A. Nijholt y T. S. Huang, "Human computing and machine understanding of human behavior: a survey," en *Artifical Intelligence for Human Computing*, Springer, 2007, pp. 47-71.

[8] J. Navarro, I. M. de Diego, P. C. Pérez y F. Ortega, "Outlier detection in animal multivariate trajectories," *Computers and Electronics in Agriculture*, vol. 190, pp. 1-6, 2021.

[9] M. Xie, S. Han, B. Tian y S. Parvin, "Anomaly detection in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 34, n.º 4, pp. 1302-1325, 2011.

[10] M. Bohge y W. Trappe, "An authentication framework for hierarchical ad hoc sensor networks," en *Proceedings of the 2nd ACM workshop on Wireless security*, ACM, 2003, pp. 79-87.

[11] R. A. LeVine, *Culture, behavior, and personality: An introduction to the comparative study of psychosocial adaptation*. Routledge, 2018.

[12] I. Carter, *Human behavior in the social environment: A social systems approach*. Routledge, 2017.

[13] W. Li y C. J. Mitchell, "Analysing the Security of Google's implementation of OpenID Connect," en *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer, 2016, pp. 357-376.

[14] M. Miculan y C. Urban, "Formal analysis of Facebook Connect single sign-on authentication protocol," en *SOFSEM*, Citeseer, vol. 11, 2011, pp. 22-28.

[15] *Financial-grade API (FAPI)*, https://openid.net/wg/fapi/, Visitado: 2022-05-04.

[16] D. Fett, R. Küsters y G. Schmitz, "The web sso standard openid connect: In-depth formal security analysis and security guidelines," en *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, IEEE, 2017, pp. 189-202.

[17] J. Navas y M. Beltrán, "Understanding and mitigating OpenID Connect threats," *Computers & Security*, vol. 84, pp. 1-16, 2019.

[18] A. G. Martín y M. Beltrán, "Mejora de la seguridad de esquemas de gestión de identidades federados mediante técnicas de User Behaviour Analytics," en *V Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2019)*, UEX, 2019, pp. 159-166.

[19] D. Recordon y D. Reed, "OpenID 2.0: a platform for user-centric identity management," en *Proceedings of the second ACM workshop on Digital identity management*, 2006, pp. 11-16.

[20] D. Hardt et al., *The OAuth 2.0 authorization framework*, 2012.

[21] N. Sakimura, J. Bradley, M. Jones, B. De Medeiros y C. Mortimore, "Openid connect core 1.0," *The OpenID Foundation*, pp. 1-85, 2014.

[22] E. Bertino y K. Takahashi, *Identity management: Concepts, technologies, and systems*. Artech House, 2010.

[23] D. Gollmann, "Computer security," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, n.º 5, pp. 544-554, 2010.

[24] S. Samonas y D. Coss, "The CIA strikes back: Redefining confidentiality, integrity and availability in security.," *Journal of Information System Security*, vol. 10, n.º 3, 2014.

[25] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen e Y. Koucheryavy, "Multi-factor authentication: A survey," *Cryptography*, vol. 2, n.º 1, pp. 1-31, 2018.

[26] S. Ayeswarya y J. Norman, "A survey on different continuous authentication systems," *International Journal of Biometrics*, vol. 11, n.º 1, pp. 67-99, 2019.

[27] G. Saunders, M. Hitchens y V. Varadharajan, "An analysis of access control models," en *Australasian Conference on Information Security and Privacy*, Springer, 1999, pp. 281-293.

[28] S. Smalley, C. Vance y W. Salamon, "Implementing SELinux as a Linux security module," *NAI Labs Report*, vol. 1, n.º 43, pp. 1-58, 2001.

[29] M. Laurent y S. Bouzefrane, *Digital identity management*. Elsevier, 2015.

[30] K. Zeilenga et al., "Lightweight directory access protocol (ldap): Technical specification road map," RFC 4510, June, inf. téc., 2006.

[31] S. P. Miller, B. C. Neuman, J. I. Schiller y J. H. Saltzer, "Kerberos authentication and authorization system," en *In Project Athena Technical Plan*, Citeseer, 1988.

[32] C. Rigney, S. Willens, A. Rubens y W. Simpson, *Remote authentication dial in user service (RADIUS)*, 2000.

[33] E. Maler y D. Reed, "The venn of identity: Options and issues in federated identity management," *IEEE security & privacy*, vol. 6, n.º 2, pp. 16-23, 2008.

[34] A. Anderson y H. Lockhart, "SAML 2.0 profile of XACML," *OASIS, September*, vol. 51, n.º 1.4, 2004.

[35] E. Hammer-Lahav, D. Recordon y D. Hardt, "The oauth 1.0 protocol," RFC 5849, April, inf. téc., 2010.

[36] C. Mainka, V. Mladenov, J. Schwenk y T. Wich, "SoK: single sign-on security—an evaluation of openID connect," en *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, 2017, pp. 251-266.

[37] F. Yang y S. Manoharan, "A security analysis of the OAuth protocol," en *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, IEEE, 2013, pp. 271-276.

[38] E. Y. Chen, Y. Pei, S. Chen, Y. Tian, R. Kotcher y P. Tague, "Oauth demystified for mobile application developers," en *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 892-903.

[39] P. Hu, R. Yang, Y. Li y W. C. Lau, "Application impersonation: problems of OAuth and API design in online social networks," en *Proceedings of the second ACM conference on Online social networks*, 2014, pp. 271-278.

[40] R. Yang, G. Li, W. C. Lau, K. Zhang y P. Hu, "Model-based security testing: An empirical study on oauth 2.0 implementations," en *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, 2016, pp. 651-662.

[41] J. Singh y N. K. Chaudhary, "OAuth 2.0: Architectural design augmentation for mitigation of common security vulnerabilities," *Journal of Information Security and Applications*, vol. 65, pp. 1-11, 2022.

[42] S. G. Morkonda, S. Chiasson y P. C. van Oorschot, "Empirical Analysis and Privacy Implications in OAuth-based Single Sign-On Systems," en *Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society*, 2021, pp. 195-208.

[43] H. Halpin, "NEXTLEAP: Decentralizing identity with privacy for secure messaging," en *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2017, pp. 1-10.

[44] R. Weingärtner y C. M. Westphall, "A design towards personally identifiable information control and awareness in OpenID Connect identity providers," en *2017 IEEE International Conference on Computer and Information Technology (CIT)*, IEEE, 2017, pp. 37-46.

[45] J. Werner y C. M. Westphall, "A model for identity management with privacy in the cloud," en *2016 IEEE Symposium on Computers and Communication (ISCC)*, IEEE, 2016, pp. 463-468.

[46] C. Villarán y M. Beltrán, "Protecting End User's Privacy When using Social Login through GDPR Compliance," 2021.

[47] G. Zachmann, "Mytoken-OpenID Connect Tokens for Long-term Authorization," Tesis doct., Karlsruher Institut für Technologie (KIT), 2021.

[48] A. Sharif, R. Carbone, G. Sciarretta y S. Ranise, "Best current practices for OAuth/OIDC Native Apps: A study of their adoption in popular providers and top-ranked Android clients," *Journal of Information Security and Applications*, vol. 65, pp. 1-18, 2022.

[49] Z. Cao, C. Chi, R. Hao e Y. Xiao, "User behavior modeling and traffic analysis of IMS presence servers," en *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, IEEE, 2008, pp. 1-5.

[50] X. Kong, M. Li, T. Tang, K. Tian, L. Moreira-Matias y F. Xia, "Shared subway shuttle bus route planning based on transport data analytics," *IEEE Transactions on Automation Science and Engineering*, vol. 15, n.º 4, pp. 1507-1520, 2018.

[51] N. Ding, Q. He, C. Wu y J. Fetzer, "Modeling traffic control agency decision behavior for multimodal manual signal control under event occurrences," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, n.º 5, pp. 2467-2478, 2015.

[52] R. Faria, J. Sousa, A. Martins y J. Lagarto, "Modeling the strategic behavior of the iberian electricity market producers using time series analysis," en *2013 10th International Conference on the European Energy Market (EEM)*, IEEE, 2013, pp. 1-5.

[53] Y. Wang, Q. Chen, C. Kang y Q. Xia, "Clustering of electricity consumption behavior dynamics toward big data applications," *IEEE transactions on smart grid*, vol. 7, n.º 5, pp. 2437-2447, 2016.

[54] H. Alemdar, C. Tunca y C. Ersoy, "Daily life behaviour monitoring for health assessment using machine learning: bridging the gap between domains," *Personal and Ubiquitous Computing*, vol. 19, n.º 2, pp. 303-315, 2015.

[55] M. Manca, P. Parvin, F. Paternò y C. Santoro, "Detecting anomalous elderly behaviour in ambient assisted living," en *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, 2017, pp. 63-68.

[56] A. Lotfi, C. Langensiepen, S. M. Mahmoud y M. J. Akhlaghinia, "Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour," *Journal of ambient intelligence and humanized computing*, vol. 3, n.º 3, pp. 205-218, 2012.

[57] N. Arbabzadeh y M. Jafari, "A data-driven approach for driving safety risk prediction using driver behavior and roadway information data," *IEEE transactions on intelligent transportation systems*, vol. 19, n.º 2, pp. 446-460, 2017.

[58] W. Zhang y Q. Fan, "Identification of abnormal driving state based on driver's model," en *ICCAS 2010*, IEEE, 2010, pp. 14-18.

[59] A. K. Sahu y P. Dwivedi, "User profile as a bridge in cross-domain recommender systems for sparsity reduction," *Applied Intelligence*, vol. 49, n.º 7, pp. 2461-2481, 2019.

[60] T. Bai, W. X. Zhao, Y. He, J.-Y. Nie y J.-R. Wen, "Characterizing and predicting early reviewers for effective product marketing on e-commerce websites," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, n.º 12, pp. 2271-2284, 2018.

[61] M. Frank, R. Biedert, E. Ma, I. Martinovic y D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE transactions on information forensics and security*, vol. 8, n.º 1, pp. 136-148, 2013.

[62] C. Shen, Y. Li, Y. Chen, X. Guan y R. A. Maxion, "Performance analysis of multi-motion sensor behavior for active smartphone authentication," *IEEE Transactions on Information Forensics and Security*, vol. 13, n.º 1, pp. 48-62, 2017.

[63] I. Firdausi, A. Erwin, A. S. Nugroho et al., "Analysis of machine learning techniques used in behavior-based malware detection," en *2010 second international conference on advances in computing, control, and telecommunication technologies*, IEEE, 2010, pp. 201-203.

[64] F. Pérez-Bueno, L. García, G. Maciá-Fernández y R. Molina, "Leveraging a Probabilistic PCA Model to Understand the Multivariate Statistical Network Monitoring Framework for Network Security Anomaly Detection," *IEEE/ACM Transactions on Networking*, 2022.

[65] P. Ravisankar, V. Ravi, G. R. Rao e I. Bose, "Detection of financial statement fraud and feature selection using data mining techniques," *Decision support systems*, vol. 50, n.º 2, pp. 491-500, 2011.

[66] U. Mahbub y R. Chellappa, "PATH: person authentication using trace histories," en *Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual*, IEEE, 2016, pp. 1-8.

[67] C. Giuffrida, K. Majdanik, M. Conti y H. Bos, "I sensed it was you: authenticating mobile users with sensor-enhanced keystroke dynamics," en *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer, 2014, pp. 92-111.

[68] Y. Li, H. Hu y G. Zhou, "Using data augmentation in continuous authentication on smartphones," *IEEE Internet of Things Journal*, vol. 6, n.º 1, pp. 628-640, 2018.

[69] H. T. Nguyen, C. L. Walker y E. A. Walker, *A first course in fuzzy logic*. CRC press, 2018.

[70] I. Brosso, A. La Neve, G. Bressan y W. V. Ruggiero, "A continuous authentication system based on user behavior analysis," en *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, IEEE, 2010, pp. 380-385.

[71] Y. Cai, H. Jiang, D. Chen y M.-C. Huang, "Online learning classifier based behavioral biometrie authentication," en *2018 IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, IEEE, 2018, pp. 62-65.

[72] L. Hernández-Álvarez, J. M. De Fuentes, L. González-Manzano y L. H. Encinas, "SmartCAMPP-Smartphone-based continuous authentication leveraging motion sensors with privacy preservation," *Pattern Recognition Letters*, vol. 147, pp. 189-196, 2021.

[73] J. M. de Fuentes, L. Gonzalez-Manzano y A. Ribagorda, "Secure and Usable User-in-a-Context Continuous Authentication in Smartphones Leveraging Non-Assisted Sensors," *Sensors*, vol. 18, n.º 4, p. 1219, 2018.

[74] C. Liu y J. He, "Access control to web pages based on user browsing behavior," en *Communication Software and Networks (ICCSN), 2017 IEEE 9th International Conference on*, IEEE, 2017, pp. 1016-1020.

[75] H. Gomi, S. Yamaguchi, K. Tsubouchi y N. Sasaya, "Continuous Authentication System Using Online Activities," en *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, IEEE, 2018, pp. 522-532.

[76] P. Zhao, C. Yan y C. Jiang, "Authenticating Web User's Identity through Browsing Sequences Modeling," en *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*, IEEE, 2016, pp. 335-342.

[77] I. Molloy, L. Dickens, C. Morisset, P.-C. Cheng, J. Lobo y A. Russo, "Risk-based security decisions under uncertainty," en *Proceedings of the second ACM conference on Data and Application Security and Privacy*, ACM, 2012, pp. 157-168.

[78] Z. Lu e Y. Sagduyu, "Risk assessment based access control with text and behavior analysis for document management," en *Military Communications Conference, MILCOM 2016-2016 IEEE*, IEEE, 2016, pp. 37-42.

[79] B. Rožac, R. Sernec, A. Košir y A. Kos, "User behavior analysis based on Identity management systems' log data," *Machine learning*, vol. 143, pp. 1-5, 2012.

[80] M. Misbahuddin, B. Bindhumadhava y B. Dheeptha, "Design of a risk based authentication system using machine learning techniques," en *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, IEEE, 2017, pp. 1-6.

[81] R. S. Gaines, W. Lisowski, S. J. Press y N. Shapiro, "Authentication by keystroke timing: Some preliminary results," Rand Corp Santa Monica CA, inf. téc., 1980.

[82] S. Bleha, C. Slivinsky y B. Hussien, "Computer-access security systems using keystroke dynamics," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, n.º 12, pp. 1217-1222, 1990.

[83] S. Cho, C. Han, D. H. Han y H.-I. Kim, "Web-based keystroke dynamics identity verification using neural network," *Journal of organizational computing and electronic commerce*, vol. 10, n.º 4, pp. 295-307, 2000.

[84] F. Monrose y A. Rubin, "Authentication via keystroke dynamics," en *Proceedings of the 4th ACM Conference on Computer and Communications Security*, 1997, pp. 48-56.

[85] K. S. Killourhy y R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," en *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, IEEE, 2009, pp. 125-134.

[86] A. Alsultan, K. Warwick y H. Wei, "Non-conventional keystroke dynamics for user authentication," *Pattern Recognition Letters*, vol. 89, pp. 53-59, 2017.

[87] J. Kim, H. Kim y P. Kang, "Keystroke dynamics-based user authentication using freely typed text based on user-adaptive feature extraction and novelty detection," *Applied Soft Computing*, vol. 62, pp. 1077-1087, 2018.

[88] K. S. Balagani, V. V. Phoha, A. Ray y S. Phoha, "On the discriminability of keystroke feature vectors used in fixed text keystroke authentication," *Pattern Recognition Letters*, vol. 32, n.º 7, pp. 1070-1080, 2011.

[89] O. Alpar, "Frequency spectrograms for biometric keystroke authentication using neural network based classifier," *Knowledge-Based Systems*, vol. 116, pp. 163-171, 2017.

[90] L. Xiaofeng, Z. Shengfei e Y. Shengwei, "Continuous authentication by free-text keystroke based on CNN plus RNN," *Procedia computer science*, vol. 147, pp. 314-318, 2019.

[91] Y. Sun, H. Ceker y S. Upadhyaya, "Shared keystroke dataset for continuous authentication," en *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, 2016, pp. 1-6.

[92] J. Huang, D. Hou, S. Schuckers, T. Law y A. Sherwin, "Benchmarking keystroke authentication algorithms," en *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, IEEE, 2017, pp. 1-6.

[93] B. Ayotte, M. Banavar, D. Hou y S. Schuckers, "Fast Free-text Authentication via Instance-based Keystroke Dynamics," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, n.º 4, pp. 377-387, 2020.

[94] R. A. Everitt y P. W. McOwan, "Java-based internet biometric authentication system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, n.º 9, pp. 1166-1172, 2003.

[95] A. A. E. Ahmed e I. Traore, "A new biometric technology based on mouse dynamics," *IEEE Transactions on dependable and secure computing*, vol. 4, n.º 3, pp. 165-179, 2007.

[96] P. Chong, Y. Elovici y A. Binder, "User authentication based on mouse dynamics using deep neural networks: A comprehensive study," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1086-1101, 2019.

[97] C. Shen, Z. Cai, X. Guan, Y. Du y R. A. Maxion, "User authentication through mouse dynamics," *IEEE Transactions on Information Forensics and Security*, vol. 8, n.º 1, pp. 16-30, 2012.

[98] D. Qin, S. Fu, G. Amariucai, D. Qiao e Y. Guan, "MAUSPAD: Mouse-based Authentication Using Segmentation-based, Progress-Adjusted DTW," en *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, 2020, pp. 425-433.

[99] T. Hu, W. Niu, X. Zhang, X. Liu, J. Lu e Y. Liu, "An insider threat detection approach based on mouse dynamics and deep learning," *Security and Communication Networks*, vol. 2019, 2019.

[100] A. Ross y A. Jain, "Information fusion in biometrics," *Pattern recognition letters*, vol. 24, n.º 13, pp. 2115-2125, 2003.

[101] S. Mondal y P. Bours, "A study on continuous authentication using a combination of keystroke and mouse biometrics," *Neurocomputing*, vol. 230, pp. 1-22, 2017.

[102] L. Fridman et al., "Multi-modal decision fusion for continuous authentication," *Computers & Electrical Engineering*, vol. 41, pp. 142-156, 2015.

[103] S. Salmeron-Majadas, R. S. Baker, O. C. Santos y J. G. Boticario, "A machine learning approach to leverage individual keyboard and mouse interaction behavior from multiple users in real-world learning scenarios," *IEEE Access*, vol. 6, pp. 39 154-39 179, 2018.

[104] J. Solano, L. Camacho, A. Correa, C. Deiro, J. Vargas y M. Ochoa, "Combining behavioral biometrics and session context analytics to enhance risk-based static authentication in web applications," *International Journal of Information Security*, vol. 20, n.º 2, pp. 181-197, 2021.

[105] A. Harilal et al., "The Wolf Of SUTD (TWOS): A Dataset of Malicious Insider Threat Behavior Based on a Gamified Competition.," *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, vol. 9, n.º 1, pp. 54-85, 2018.

[106] X. Wang, Q. Zheng, K. Zheng y T. Wu, "User Authentication Method Based on MKL for Keystroke and Mouse Behavioral Feature Fusion," *Security and Communication Networks*, vol. 2020, 2020.

[107] K. O. Bailey, J. S. Okolica y G. L. Peterson, "User identification and authentication using multi-modal behavioral biometrics," *Computers & Security*, vol. 43, pp. 77-89, 2014.

[108] Y. Li, B. Zou, S. Deng y G. Zhou, "Using feature fusion strategies in continuous authentication on smartphones," *IEEE Internet Computing*, vol. 24, n.º 2, pp. 49-56, 2020.

[109] I. Traore, I. Woungang, M. S. Obaidat, Y. Nakkabi e I. Lai, "Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments," en *2012 fourth international conference on digital home*, IEEE, 2012, pp. 138-145.

[110] A. G. Martín, M. Beltrán, A. Fernández-Isabel e I. M. de Diego, "An approach to detect user behaviour anomalies within identity federations," *Computers & Security*, vol. 1–18, p. 102 356, 2021.

[111] L. Hernández-Álvarez, J. M. de Fuentes, L. González-Manzano y L. Hernández Encinas, "Privacy-preserving sensor-based continuous authentication and user profiling: a review," *Sensors*, vol. 21, n.º 1, pp. 92-115, 2020.

[112] A. Vastel, P. Laperdrix, W. Rudametkin y R. Rouvoy, "Fp-scanner: The privacy implications of browser fingerprint inconsistencies," en *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 135-150.

[113] M. Beltrán, "Identifying, authenticating and authorizing smart objects and end users to cloud services in Internet of Things," *Computers & Security*, vol. 77, pp. 595-611, 2018.

[114] R. Magán-Carrión, J. Camacho, G. Maciá-Fernández y Á. Ruíz-Zafra, "Multivariate Statistical Network Monitoring–Sensor: An effective tool for real-time monitoring and anomaly detection in complex networks and systems," *International Journal of Distributed Sensor Networks*, vol. 16, n.º 5, pp. 1-14, 2020.

[115] A. Gómez-Boix, P. Laperdrix y B. Baudry, "Hiding in the crowd: an analysis of the effectiveness of browser fingerprinting at large scale," en *Proceedings of the 2018 world wide web conference*, 2018, pp. 309-318.

[116] P. Laperdrix, N. Bielova, B. Baudry y G. Avoine, "Browser fingerprinting: A survey," *ACM Transactions on the Web (TWEB)*, vol. 14, n.º 2, pp. 1-33, 2020.

[117] M. Abuhamad, A. Abusnaina, D. Nyang y D. Mohaisen, "Sensor-based Continuous Authentication of Smartphones' Users Using Behavioral Biometrics: A Contemporary Survey," *IEEE Internet of Things Journal*, vol. 8, n.º 1, pp. 65-84, 2020.

[118] M. Bhatnagar, R. K. Jain y N. S. Khairnar, "A Survey on Behavioral Biometric Techniques: Mouse vs Keyboard Dynamics," *Int. J. Comput. Appl*, vol. 975, pp. 1-5, 2013.

[119] C. Chio y D. Freeman, *Machine learning and security: Protecting systems with data and algorithms*. .º'Reilly Media, Inc.", 2018.

[120] V. Kozitsin, I. Katser y D. Lakontsev, "Online Forecasting and Anomaly Detection Based on the ARIMA Model," *Applied Sciences*, vol. 11, n.º 7, pp. 1-13, 2021.

[121] S. Hariri, M. C. Kind y R. J. Brunner, "Extended isolation forest," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, n.º 4, pp. 1479-1489, 2019.

[122] Z. Cheng, C. Zou y J. Dong, "Outlier detection using isolation forest and local outlier factor," en *Proceedings of the conference on research in adaptive and convergent systems*, 2019, pp. 161-168.

[123] E. Schubert, J. Sander, M. Ester, H. P. Kriegel y X. Xu, "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN," *ACM Transactions on Database Systems (TODS)*, vol. 42, n.º 3, pp. 1-21, 2017.

[124] T. Shimshon, R. Moskovitch, L. Rokach e Y. Elovici, "Clustering di-graphs for continuously verifying users according to their typing patterns," en *2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel*, IEEE, 2010, pp. 445-449.

[125] B. Tang, Q. Hu y D. Lin, "Reducing false positives of user-to-entity first-access alerts for user behavior analytics," en *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, 2017, pp. 804-811.

[126] J. Yan, Y. Qi, Q. Rao y S. Qi, "Towards a user-friendly and secure hand shaking authentication for smartphones," en *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, IEEE, 2018, pp. 1170-1179.

[127] Z. C. Lipton, C. Elkan y B. Narayanaswamy, "Thresholding classifiers to maximize F1 score," *Machine Learning and Knowledge Discovery in Databases*, vol. 8725, pp. 225-239, 2014.

[128] S. Eberz, K. B. Rasmussen, V. Lenders e I. Martinovic, "Evaluating behavioral biometrics for continuous authentication: Challenges and metrics," en *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2017, pp. 386-399.

[129] I. M. De Diego, A. R. Redondo, R. R. Fernández, J. Navarro y J. M. Moguerza, "General Performance Score for classification problems," *Applied Intelligence*, 2022.

[130] A. G. Martín, I. M. de Diego, A. Fernández-Isabel, M. Beltrán y R. R. Fernández, "Combining user behavioural information at the feature level to enhance continuous authentication systems," *Knowledge-Based Systems*, pp. 1-13, 2022.

[131] Y. Sun, J. Li, J. Liu, B. Sun y C. Chow, "An improvement of symbolic aggregate approximation distance measure for time series," *Neurocomputing*, vol. 138, pp. 189-198, 2014.

[132] P. Geurts, D. Ernst y L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, n.º 1, pp. 3-42, 2006.

[133] F. Moosmann, B. Triggs y F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," en *Twentieth Annual Conference on Neural Information Processing Systems (NIPS'06)*, MIT Press, 2006, pp. 985-992.

[134] M. G. Baydogan y G. Runger, "Learning a symbolic representation for multivariate time series classification," *Data Mining and Knowledge Discovery*, vol. 29, n.º 2, pp. 400-422, 2015.

[135] M. P. Van der Loo et al., "The stringdist package for approximate string matching.," *R J.*, vol. 6, n.º 1, pp. 1-13, 2014.

[136] H. Li y N. Homer, "A survey of sequence alignment algorithms for next-generation sequencing," *Briefings in bioinformatics*, vol. 11, n.º 5, pp. 473-483, 2010.

[137] C. Trapnell y M. C. Schatz, "Optimizing data intensive GPGPU computations for DNA sequence alignment," *Parallel computing*, vol. 35, n.º 8-9, pp. 429-440, 2009.

[138] J. Cheetham, F. Dehne, S. Pitre, A. Rau-Chaplin y P. J. Taillon, "Parallel clustal w for pc clusters," en *International Conference on Computational Science and Its Applications*, Springer, 2003, pp. 300-309.

[139] X. Huang y K.-M. Chao, "A generalized global alignment algorithm," *Bioinformatics*, vol. 19, n.º 2, pp. 228-233, 2003.

[140] H. Abdi, "Metric multidimensional scaling (MDS): analyzing distance matrices," *Encyclopedia of measurement and statistics*, pp. 1-13, 2007.

[141] F. Klinker, "Exponential moving average versus moving exponential average," *Mathematische Semesterberichte*, vol. 58, n.º 1, pp. 97-107, 2011.

[142] *let's chat*, https://sdelements.github.io/lets-chat, Visitado: 2022-05-04.

[143] M. Cantelon, M. Harter, T. Holowaychuk y N. Rajlich, *Node. js in Action*. Manning Greenwich, 2014.

[144] *Mongodb*, https://www.mongodb.com/, Visitado: 2022-05-04.

[145] L. A. Leiva y R. Vivó, "Web browsing behavior analysis and interactive hypervideo," *ACM Transactions on the Web (TWEB)*, vol. 7, n.º 4, pp. 1-28, 2013.

[146] *OpenAM*, https://backstage.forgerock.com/docs/openam/13.5/, Visitado: 2022-05-04.

[147] Martín, Alejandro G and Beltrán, Marta and Fernández-Isabel, Alberto and de Diego, Isaac Martín, "Keystroke and Mouse Dynamics for UEBA Dataset, Mendeley Data, v2," 2020.

[148] J. Ho y D.-K. Kang, "One-class Naïve Bayes with duration feature ranking for accurate user authentication using keystroke dynamics," *Applied Intelligence*, vol. 48, n.º 6, pp. 1547-1564, 2018.

[149] Y. Zhao, "Learning user keystroke patterns for authentication," *Proceedings of the world academy of science, engineering and technology*, vol. 14, pp. 65-70, 2006.

[150] M. Malkauthekar, "Analysis of Euclidean distance and Manhattan distance measure in Face recognition," en *Third International Conference on Computational Intelligence and Information Technology (CIIT 2013)*, IET, 2013, pp. 503-507.

[151] T. Lodderstedt, S. Dronia y M. Scurtescu, *OAuth 2.0 token revocation*, 2013.