UNIVERSIDAD NACIONAL HERMILIO VALDIZÁN

FACULTAD: INGENIERÍA INDUSTRIAL Y DE SISTEMAS

ESCUELA ACADÉMICO PROFESIONAL DE: INGENIERÍA DE SISTEMAS

AUTENTICACIÓN CONTINUA E IMPLÍCITA EN DISPOSITIVOS MÓVILES ANDROID UTILIZANDO DATOS DE USO DE LA PANTALLA TÁCTIL

TESIS PARA OPTAR EL TITULO EN INGENIERÍA DE SISTEMAS

BACHILLERES: **JUAN MANUEL ROJAS RONQUILLO**

ASESOR: **FERMÍN ROLANDO MONTESINOS CHAVEZ**

HUÁNUCO – PERÚ

2017

AGRADECIMIENTOS

Me gustaría agradecer a mi asesor de tesis, Msc. Fermín Montesinos por su guía. También me gustaría agradecer al Ing. Alcides Bernardo por sugerir el tema de investigación.

Finalmente, me gustaría agradecer a mis padres hermanos y hermanas por su apoyo a lo largo de mi carrera académica. También esta tesis no sería posible sin la ayuda de mis compañeros en Rhem Solutions y en la Secretaria Técnica.

DEDICATORIA

Esto está dedicado a mis padres.

RESUMEN

En esta investigación se revisa el estado del arte y se evalúa la efectividad de la Autenticación Continua e Implícita también llamado Autenticación Continua y Transparente en dispositivos móviles con sistema operativo Android. La Autenticación Continua e Implícita es una propuesta que busca mejorar la seguridad de los sistemas software verificando la identidad del usuario continuamente mientras usa el sistema y previniendo de este modo que un usuario no autorizado que ya accedió al sistema continúe manipulando los datos. En lugar de desarrollar un Sistema de Autenticación Continua e Implícita desde cero esta investigación utiliza proyectos open source existentes para recolectar datos y evaluar la efectividad de la autenticación continua e implícita utilizando métricas comunes de la literatura.

Palabras Clave

Seguridad, Autenticación, Biometría, Inicio de Sesión, Log In, Touch Events, Cybersecutiry, Novelty Detection, LIBSVM, SVM, Machine Learning, Support Vector Machines, One-Class Classification.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABLAS

1. INTRODUCCIÓN

Los dispositivos móviles han evolucionado de ser simples herramientas de comunicación a completos entornos computarizados. Las mejoras en velocidad de procesador, memoria y conexión a internet permiten el uso de aplicaciones y servicios para uso personal y de la empresa. Durante el uso de los dispositivos móviles se almacenan información personal, privada y corporativa. Por lo tanto existe un alto riesgo de mal uso de estos dispositivos por parte de usuarios malintencionados que puedan comprometer la privacidad y confidencialidad de la información.

Autenticación de usuario es la primera línea de defensa como un control preventivo para detener el acceso no deseado al dispositivo móvil. Este control funciona efectivamente cuando el esquema de autenticación es seguro y usable. Tradicionalmente se usa un Personal Identification Number (PIN) para autenticar usuarios en dispositivos móviles. Este PIN bloquea el acceso al dispositivo móvil pero no el acceso a las aplicaciones y esto conlleva un riesgo de mal uso mientras el dispositivo esta desbloqueado. Es posible instalar un aplicativo móvil para bloquear el acceso a las aplicaciones, pero ingresar un PIN a cada momento para usar una aplicación puede resultar muy inconveniente para el usuario haciendo que este desactive la seguridad en favor de la usabilidad.

1.1 Antecedentes del Problema

La autenticación en dispositivos móviles (smartphones, tables, wearables, etc) comparte los mismos problemas que la autenticación en aplicaciones web y aplicaciones de escritorio. Sin embargo debido al contexto en el que son usados, la información en los dispositivos móviles está más propenso y vulnerable a ser comprometido. Por ejemplo los servidores web están ubicados en un ambiente físico y están protegidos por la empresa responsable, las computadoras personales se encuentran protegidos por el ambiente empresarial o el hogar de los usuarios. Además las empresas suelen contar con políticas de seguridad y hardware de seguridad de red. En contraste los dispositivos móviles personales se encuentran físicamente con el usuario en todo momento y son susceptibles a pérdida y robo comprometiendo toda la información privada del usuario y la información empresarial por ejemplo si el usuario tiene abierto su correo empresarial en el dispositivo.

El Problema del Password se refiere a la tendencia de los usuarios a utilizar simples passwords compartirlos, reutilizarlos y anotarlos. Algunos sistemas automatizados miden la fortaleza del password en términos de la longitud de este, si contienen mayúsculas, números y caracteres especiales y no permiten el registro del usuario si este no crea un password fuerte. El problema se complica cuando el usuario tiene que utilizar diferentes

password fuertes para las múltiples cuentas que el usuario posee en social media y los sistemas empresariales a los cuales tiene acceso.

Políticas de Autenticación: Para contrarrestar la creación de password débiles, algunas organizaciones implementan políticas de autenticación para sus empleados. Estas políticas definen por ejemplo, la longitud del password, inclusión de caracteres especiales, y tiempo de expiración, de esta forma el empleado estaría obligado a crear un password diferente cada cierto tiempo.

Modernos dispositivos android incluyen Face Unlock una característica que se utiliza para bloquear el acceso al dispositivo. El argumento en contra del uso de software de reconocimiento facial es que es fácil de engañar al dispositivo en hacerle creer que eres el propietario.[1] En caso de que el dispositivo no pueda reconocer el rostro ya sea por la posición e iluminación ambiental, este provee una autenticación alternativa utilizando PIN o Pattern Unlock.

La autenticación biométrica está basado en las características únicas del usuario, estas pueden ser fisiológicas o de comportamiento y pueden ofrecer una mayor confiabilidad ya que estas características no pueden ser compartidas, olvidadas o perdidas y eliminan la inconveniencia de los passwords. Sin embargo luego de la verificación inicial no hay otras medidas

[1] http://www.androidcentral.com/how-set-face-unlock-your-htc-one-x-or-evo-4g-lte

adicionales que aseguren de que es el usuario quien está usando el dispositivo.

1.2 Enseñar a las computadoras a conocer su propietario

Según la Dr. Crawford (2012) Una solución al problema de autenticación en dispositivos móviles debería tener las siguientes características:

1. Requiera menos esfuerzo que los actuales métodos de autenticación,
2. Que vaya más allá de la verificación inicial para proteger los datos y la funcionalidad.
3. Autenticar a los usuarios continuamente para mantener la confianza en su identidad.
4. Proveer un método de seguridad que es aceptable y se considere confiable por los propietarios del dispositivo.
5. Respete las necesidades del entorno del dispositivo móvil en términos de su corto periodo de uso y sus limitaciones en cuanto a su velocidad de procesador y memoria.

1.3 Autenticación Continua e Implícita

Los modernos dispositivos móviles cuentan con múltiples sensores capaces de recolectar información del usuario como es la ubicación del usuario, el uso de la pantalla táctil o touch input data y la forma de caminar

del usuario o gait pattern, también el dispositivo puede recolectar el patrón de tipeo o keystroke pattern.

Para verificar la identidad del usuario de forma implícita se recolecta información de éste mientras el dispositivo está en uso. A esta fase de recolección de información se le llama training, una vez que se ha recolectado suficiente información el siguiente paso es autenticar al usuario y para esto es utilizan algoritmos de machine learning [4]. Los nuevos datos recolectados pasan un proceso de evaluación utilizando algoritmos de clasificación o classifiers, el resultado decidirá si el usuario actual es reconocido como el propietario del dispositivo.

El resultado del algoritmo de clasificación no es binario como en el caso de los passwords sino una aproximación numérica y para este se define un intervalo de error o máximo threshold. Si el resultado numérico de la clasificación es menor al threshold se considera al usuario autenticado en caso contrario se considera al usuario un intruso y se procede a bloquear el dispositivo solicitando al usuario ingrese un PIN o Pattern Unlock.

Al utilizar técnicas biométricas el resultado de la autenticación no es 100% confiable, se dan casos en que se rechaza al propietario del dispositivo, esto se mide con False Rejection Rate y otros casos en que se acepta un usuario intruso, esto se mide utilizando False Aceptance Rate.

1.4 Formulación del Problema

1.4.1 Problema general

¿Con la autenticación continua e implícita se puede lograr una mejor seguridad en dispositivos móviles?

1.4.2 Problemas específicos

- ✓ ¿Qué algoritmo de clasificación sería apropiado para su uso en dispositivos móviles respetando sus limitaciones de hardware?
- ✓ ¿Los datos uso de la pantalla táctil del dispositivo móvil presentan características distintivas para identificar al usuario?
- ✓ ¿Cuál es la efectividad de la autenticación continua e implícita en dispositivos móviles utilizando eventos touch de la pantalla táctil?

1.5 Objetivo de la Investigación

1.5.1 Objetivo General

Evaluar la efectividad de la autenticación continua e implícita en dispositivos móviles utilizando proyectos existentes open source[2].

1.5.2 Objetivos Específicos

- ✓ Utilizar un proyecto open source [3]para recolectar datos de uso de la pantalla táctil del dispositivo móvil de usuarios de prueba.
- ✓ Seleccionar un algoritmo de clasificación adecuado para su uso en dispositivos móviles.
- ✓ Utilizar un proyecto open source que implementa el algoritmo seleccionado en lenguaje de programación Java.
- ✓ Utilizar los datos recolectados para evaluar la efectividad de la autenticación continua e implícita.

1.6 Variables

1.6.1 Variable Dependiente

Efectividad de la autenticación implícita en dispositivos móviles.

1.6.2 Variables Independientes

- Algoritmo de clasificación utilizado
- Datos de uso de la pantalla táctil del dispositivo móvil

[2] https://en.wikipedia.org/wiki/List_of_free_and_open-source_Android_applications
[3] https://en.wikipedia.org/wiki/List_of_Android_launchers

2. MARCO TEÓRICO

2.1 Antecedentes

Nathan Clarke

2011 Transparent User Authentication Biometrics, RFID and Behavioral Profiling. Book, Springer Publisher

Uso actual de autenticación de usuario

Autenticación de Usuario es un componente esencial en todo sistema seguro. Sin este, es imposible mantener la confidencialidad, integridad y disponibilidad de sistemas. A diferencia de firewalls, antivirus y encriptación, autenticación es uno de los pocos controles de seguridad con el que todos los usuarios deben interactuar. Los enfoques basados en conocimiento secreto (secret-knowledge) y token dependen del usuario para mantener la seguridad del sistema. Un token perdido o robado o password compartido comprometerá el sistema. Técnicas Biométricas proveen un nivel adicional de seguridad, pero no están necesariamente libres de ser comprometidos. Por lo tanto se puede decir que los actuales enfoques de autenticación fallan en cumplir las necesidades o expectativas de usuarios u organizaciones.

El entorno tecnológico cambiante

El rápidamente cambiante entorno tecnológico y de amenazas coloca una carga real y considerable sobre los individuos. Mientras iniciativas de concientización de seguridad pueden buscar mejorar el entendimiento de las

personas sobre los problemas asociados al usar la tecnología y en particular el Internet, la velocidad de cambio en las amenazas dejara un vacío entre el conocimiento de los usuarios y las amenazas que buscan explotar usuarios. Las amenazas tecnológicas continuarán atacando sistemas y nuevos desarrollos en malware encontrarán formas de traspasar las medidas de seguridad. Al final, las medidas de seguridad tecnológicas se enfocarán en mitigar y reducir el riesgo de ataque. Los atacantes por lo tanto continuaran en enfocarse en los usuarios finales tomando ventaja de su falta de conciencia de seguridad. Mejor concientización, mejor diseño de interfaces de seguridad y funcionalidad permitirá a los usuarios estar en una mejor posición para tomar la decisión correcta.

Enfoques de autenticación intrusiva

Autenticación es clave para mantener la seguridad y un amplio rango de enfoques han sido desarrollados para cumplir las necesidades únicas de los sistemas que han sido creados. En su forma más pura, autenticación es acerca de algo secreto y comparar una muestra provista contra ese algo secreto. Passwords son por lo tanto teoréticamente la forma más sólida de autenticación y pueden proveer más que protección suficiente – asumiendo que no dependen en las personas para recordarlos. Una evolución natural ha ocurrido en la identificación de debilidades en passwords, desarrollo de tokens para quitar la presión cognitiva en las personas, reconocer que tokens sólo autentica la presencia del token – no a la persona en posesión

de este – y finalmente biometrics, una credencial de autenticación atado a la persona, removiendo la dependencia en ellos en recordarlos.

Biometrics, sin embargo, no son la panacea del problema de autenticación, éstos introducen su propio conjunto de incidencias, siendo por ejemplo debilidad en enfoques individuales, falta de universalidad (todos los usuarios presentan la característica) haciendo que los administradores deban deployar y administrar múltiples sistemas, pobre permanencia (la característica de la persona cambia con el tiempo), baja aceptabilidad, la necesidad de complejos sistemas para recolectar y procesar muestras o asuntos relacionados a privacidad. Sin embargo Biometrics ofrece una capacidad para desconectar la carga cognitiva del usuario y lo asegura en un contenedor que provee una seguridad más efectiva que un token. Pero esto en sí mismo no será suficiente.

Técnicas transparentes

Autenticación transparente no es un simple objetivo a lograr. Mientras un número de enfoques podrían ser utilizados, la misma naturaleza de transparencia introduce una variabilidad en el proceso en el cual biometrics muestran tradicionalmente un bajo performance. Sin embargo, la investigación ha ido mejorando y la necesidad existe en deployar enfoques biométricos a lo largo de una variedad de escenarios de aplicación, proveedores de tecnología están desarrollando algoritmos más robustos y

flexibles que pueden soportar la variabilidad en esos sistemas. Mejoras en rotación de imagen, escala, normalización y segmentación todo esto permitirá mejoras en la usabilidad y aceptabilidad de los enfoques.

También es evidente que las características de comportamiento son más aplicables a la Transparencia que las características fisiológicas en general. Además se debe considerar la disponibilidad del hardware necesario. Técnicas que utilizan hardware existente tienen una obvia ventaja en costo beneficio.

Multibiometrics

Multibiometrics ofrece una oportunidad para superar muchas de las debilidades de individual uni-modal biometrics. Ofrece varias ventajas:

Resuelve incidencias en los cuales los usuarios no tienen o no pueden presentar una característica biométrica en particular.

- La habilidad de hacer decisiones de autenticación en un amplio conjunto de información, incluyendo múltiples muestras de la misma característica biométrica, múltiples algoritmos o muestras de diferentes características biométricas.
- Modos de procesamiento síncrono y asíncrono, minimizando la carga computacional y mejorando la aceptación de usuario.
- Proveyendo una mejora significativa en el performance total del sistema biométrico.

Estándares biométricos

Estándares son clave en asegurar interoperabilidad dentro y entre sistemas. Al proveer a los clientes con elección y para los proveedores de tecnología una oportunidad para competir en igualdad. Aún más importante, el desarrollo de estándares también demuestra un nivel de madurez en el area.

La estandarización de muestras de datos biométricos, encapsulación de datos y un framework API que permitan una completa interoperabilidad de componentes biométricos es esencial para proveer una autenticación compuesta más robusta.

Heather Anne Crawford

2012 A Framework for Continuous, Transparent Authentication on Mobile Devices. PhD thesis.

Conclusiones

Esta investigación provee una descripción de un framework de autenticación continua y transparente en dispositivos móviles. Para proveer soporte para el uso de técnicas biométricas de comportamiento (behavioral biometrics) dentro del framework, se realizaron estudios de factibilidad en cuanto a keystroke dynamics (patrón de tipeo), speaker verification (reconocimiento de voz) y una combinación multimodal de estas técnicas biométricas. Para

responder preguntas sobre la utilidad y necesidad de los usuarios que podrían usar el framework, se implementó un estudio de percepción de usuario sobre la autenticación transparente.

Futuras investigaciones incluyen crear simulaciones de los procesos y estructuras de datos que comprende el framework, y una implementación de prueba de concepto en una plataforma de dispositivo móvil que soporte ejecución en segundo plano (background processing). Para soportar estas futuras investigaciones, un gran estudio con más participantes y más datos sobre la utilidad de keystroke dynamics y speaker verification están justificados por los estudios de factibilidad de esta investigación.

Resumen

Esta investigación se centra en la elección de algoritmos de Machine Learning apropiados para dispositivos móviles. Se crean aplicaciones para el sistema operativo IOS para la recolección de patrones de tipeo y de voz de una muestra pequeña de 8 a 12 usuarios durante un tiempo de una a dos semanas. Luego de recolectar la muestra de datos la investigadora hace una pequeña simulación utilizando MatLab para la ejecución de los algoritmos de Machine Learning y luego realiza una comparación de resultados. También la investigadora realiza un aplicativo IOS que simula el funcionamiento del Framework de Autenticación Continua para su estudio de percepción de usuario.

Sevasti Karatzouni

2014 Non-Intrusive Continuous User Authentication for Mobile Devices. PhD thesis.

Resumen

Esta tesis presenta una aplicación NICA Non-Intrusive Continuous Authentication una aplicación cliente-servidor desarrollado en .Net para el Sistema Operativo Windows que utiliza el patrón de tipeo y reconocimiento de voz para autenticar al usuario. Se consideró que los algoritmos de Machine Learning requerían de mucho recurso de hardware para su ejecución por lo que utilizaron un servidor para la ejecución de los algoritmos y la aplicación cliente recolecta la información biométrica. Luego evalúan el performance de NICA y la percepción de usuario en un estudio de laboratorio donde los participantes realizan un conjunto determinado de tareas facilitados por los investigadores. Luego la investigadora se centra en la simulación de los algoritmos de Machine Learning utilizados en NICA utilizando el dataset Reality Commons[4] del MIT y Matlab, donde los resultados demuestran que el bajo performance de NICA se debió en gran medida a una mala programación de los algoritmos de clasificación utilizados y no a las técnicas biométricas utilizadas. Luego la investigadora propone algunas mejoras al funcionamiento de NICA resultando en el Framework

[4] http://realitycommons.media.mit.edu/realitymining4.html

Casper. Luego realiza una simulación de Casper y compara los resultados con los de NICA. Es importante recordar que el dataset Reality Commons del MIT no se recolectó teniendo en cuenta la autenticación transparente por lo que la investigadora realiza un pre-procesamiento de la data para que se ajuste a los requerimientos necesarios para una simulación de autenticación continua y transparente.

Hassan Khan, Aaron Atwater, and Urs Hengartner

2014 Itus: An Implicit Authentication Framework for Android. Research paper

Resumen

Esta investigación se centra en la implementación de autenticación implícita en dispositivos android y la evaluación del impacto en el consumo de memoria, uso de CPU y consumo de batería. El resultado es la librería Itus[5] disponible como código abierto (open source) en github[6]. A diferencia de las anteriores investigaciones sobre autenticación transparente en el cual el diseño del framework consiste en interceptar eventos y recolectar datos de múltiples aplicaciones, Itus está desarrollado para interceptar y recolectar datos de una sola aplicación basándose en la premisa de que cada aplicación tiene un modo distinto de uso y los eventos capturados en una

[5] https://crysp.uwaterloo.ca/software/itus/
[6] https://github.com/hassan-khan/itus

aplicación son diferentes a los de otra aplicación. Es importante resaltar que el principal aporte de esta investigación consiste en proveer de una arquitectura base en el cual Itus se encarga de gran parte de los detalles de implementación y el desarrollador puede concentrarse en el desarrollo de los algoritmos de clasificación. Sin embargo Itus es una implementación en desarrollo y no se considera un proyecto listo para su uso en entornos de producción. Esto se debe a que algunas funcionalidades descritas en el paper no están presentes en el código fuente. Adicionalmente el código fuente en el repositorio de Itus está desarrollado en Eclipse IDE y el nuevo entorno integrado de desarrollo de la plataforma Android es Android Studio basado en IntelliJ IDEA y Gradle. Sin embargo una versión de Itus para Gradle[7] se puede encontrar en el repositorio del proyecto de una Tesis Master [6] FireLock[8].

[7] https://en.wikipedia.org/wiki/Gradle
[8] https://github.com/lalitagarwal/FireLock/tree/master/demo/TextSecure-master

2.2 Bases Teóricas

2.2.1 Passwords, PINs y Pregunta Secreta

Son los mecanismos de autenticación más comúnmente deployados y conocidos por los usuarios. Típicamente, mientras más difícil sea el password de adivinar por un atacante, puede resultar más difícil de recordar por el usuario legítimo. Los PINs al estar limitados a usar sólo números pueden ser más fáciles de recordar y más fáciles de adivinar por un atacante. La Pregunta Secreta se basa en utilizar un número de preguntas sobre tu historial personal, gustos/disgustos, opiniones y actitudes.

2.2.2 Passwords Gráficos

Se basa en la habilidad de los usuarios de reconocer y recordar imágenes en lugar de cadenas de caracteres largos y complejos. Passwords Gráficos se clasifican en tres categorías [7].

Recognition – requiere que los usuarios memoricen un conjunto de imágenes durante la creación del password y luego reconocer sus imágenes de un superconjunto.

Recall – requiere que los usuarios recuerden y reproduzcan una imagen que ha sido previamente dibujado en un canvas o en un grid

Cued-recall – requiere que los usuarios recuerden e identifiquen ubicaciones específicas dentro de una imagen.

2.2.3 Ataques a Passwords

Los ataques a passwords se pueden clasificar en técnicas y no técnicas [1]. Lo enfoques técnicos incluyen:

- Brute-force attack – un enfoque sistemático que consiste en probar todas las permutaciones de passwords.
- Phishing – Un ataque basado en email que pretende haber venido de un origen legítimo (ej. una banco) preguntando que confirmes tus datos de acceso.
- Dispositivo de entrada manipulado – capturar la información a través de dispositivos de entrada falsos o dispositivos que interceptan la información.
- Troyano – Son aplicaciones que se disfrazan de auténticas aplicaciones con el propósito de robar información, espiar tu ubicación y/o espiar tus conversaciones[9].
- Network sniffer – captura todo el tráfico en una red y lo analiza en busca de passwords.
- Smudge attack – utiliza las manchas aceitosas de la pantalla táctil para inferir el password[10].

[9] http://pocketnow.com/2013/04/03/android-trojans
[10] https://en.wikipedia.org/wiki/Smudge_attack

- Ihavebeenpwned[11] – Sucede cuando alguna de las redes de social media en la cual estas registrado sufre un data breach (es decir el sitio web ha sido comprometido y todas las cuentas de usuario y contraseñas son expuestos).

Ataques no técnicos incluyen:

- Guesswork – es posible que el atacante pueda adivinar el password utilizando información del usuario.
- Shoulder surfing – observar el password ingresado desde una ubicación cercana.
- Social engineering – manipular al usuario en realizar ciertas acciones o divulgar información confidencial[12].

2.2.4 Biometrics

Biometrics se define como la "ciencia de reconocer un individuo basándose en sus características fisiológicas o de comportamiento". Biometric fisiológico (physiological biometric) es uno que se mide del cuerpo humano. Ejemplos incluyen escaneo de iris y de retina, huella digital y reconocimiento facial. Biometrics de comportamiento (behavioral biometrics) dependen del comportamiento único de la persona, es decir, cómo realizan tareas

[11] https://haveibeenpwned.com/
[12] https://en.wikipedia.org/wiki/Social_engineering_(security)

particulares. Behavioral Biometrics son por ejemplo firmas, gait (forma de caminar), reconocimiento de voz y patrón de tipeo (keystroke dynamics).

Características Biométricas están atados a la persona quien los provee y no pueden ser compartidos y es poco probable que sean robados, a diferencia de passswords y PINs. Biometrics presenta una alternativa a la autenticación tradicional basado en un conocimiento y propiedad (es decir, algo que sabes y algo que posees, respectivamente). Sin embargo, no son universales en el sentido que no toda la población de usuarios puede presentar la característica fisiológica. Por ejemplo el Reniec[13] utiliza la huella dactillar sin embargo es posible que haya personas que no tengan o no puedan presentar su huella dactillar.

Biometrics son usados para dos propósitos:

Verificación (alias autenticación): la persona que provee la característica biométrica se identifica como un usuario del sistema, luego la característica biométrica es comparado a la característica almacenada en el sistema del usuario registrado. Si coinciden, se le concede acceso a la persona al recurso protegido, de lo contrario se le niega el acceso. Verificación es llamado comparación 1:1. Una persona se presenta ante un sistema utilizando un nombre de usuario o email, o usando una tarjeta de identificación con o sin chip que contiene información adicional. Tarjetas que

[13] http://portales.reniec.gob.pe/web/seminario_biometrico/biometria

contienen chips son llamados smartcards. Por ejemplo el DNI electrónico DNIe[14] contiene la huella dactilar dentro del chip entre otros datos adicionales[15]. Otro ejemplo es el servicio web de verificación biométrica del Reniec [8] para el cual se debe proporcionar el número del DNI y la huella dactilar disponible para instituciones públicas y privadas[16].

Identificación: reconocer a una persona determinada vía una comparación entre la característica provista por este y todas las características en una base de datos de personas registradas. La característica biométrica es comparada a cada patrón en la base de datos hasta que encuentre una coincidencia. Si no existe, la persona no se encuentra registrado. Identificación es llamado comparación 1:N. Esto es un proceso que tiene un mayor consumo de tiempo y dificultad comparado a la verificación, pero requiere menos información de la persona. Un ejemplo es el módulo de consultas 1:N del servicio biométrico del Reniec. Este módulo sólo se encuentra habilitado para la Policía Nacional y organizaciones que administran Justicia [8].

2.2.5 Multimodal Biometrics

El objetivo de Multimodal Biometrics [3] es minimizar las debilidades de biometrics individuales al proveer mayor información sobre el cual se pueden

[14] https://www.youtube.com/embed/_XgcSzINNE0
[15] http://portales.reniec.gob.pe/web/dni/uso
[16] http://www.reniec.gob.pe/portal/masServiciosLinea.htm#

hacer decisiones biométricas. Algunos métodos para combinar biométrics son los siguientes:

1. Medir la misma muestra con múltiples sensores:
    a. Una sola muestra, múltiple sensores (es decir, una huella dactilar, múltiples scanners). Este método de combinación obtiene la misma muestra biométrica con una serie de scanners.
    b. Una sola muestra, múltiple classifiers (es decir, presentar una sola huella dactilar a más de un algoritmo de clasificación)
    c. Una sola muestra, múltiples versiones. En este método de combinación dos o más muestras son tomadas del mismo tipo de biometric, pero el origen es diferente. Por ejemplo la huella dactilar del dedo índice y del dedo pulgar.
2. Medir más de un distinto identificador biométrico y combinar los resultados de los algoritmos de clasificación.
    a. Feature Fusion: combina los feature vectors[17] (datos extraídos) de dos o más características biométricas concatenando los datos extraídos en un solo vector que se presenta al algoritmo de clasificación.
    b. Match Score Fusion; dos o más biométrics son presentados al algoritmo de clasificación y son asignados un score (que identifica

[17] https://en.wikipedia.org/wiki/Feature_vector

la aproximación de la muestra con la plantilla). Los scores son combinados y se toma la decisión.

c. Decision Score Fusion: multiple feature vectors son presentados a los algoritmos de clasificación y de acuerdo al resultado son divididos en dos grupos: aceptados y rechazados. Los scores son combinados utilizando un factor de peso.

2.2.6 Physiological Biometrics

La mayoría de las técnicas biométricas comerciales son fisiológicas y tienden a estar basados en una tecnología más madura y probada. Adicionalmente, medidas biométricas fisiológicas presentan una mayor discriminación y menor variabilidad, por lo tanto se utilizan comúnmente en sistemas de verificación e identificación [1]. Algunos ejemplos serían:

- Ear geometry
- Facial recognition
- Fingerprint recognition
- Hand geometry
- Iris recognition
- Retina recognition

2.2.7 Behavioral Biometrics

Behavioral biometrics clasifican a una persona de acuerdo a alguna clase de comportamiento único. Sin embargo, los comportamientos tienden a cambiar

con el tiempo, las características discriminativas usadas en el reconocimiento también cambian. En general, behavioral biometrics tienden a ser más transparentes y convenientes para el usuario, sin embargo, con el costo de un menor performance de autenticación [1]. Algunos ejemplos serían:

- Behavioral profiling
- Gait recognition
- Keystroke analysis
- Speaker recognition

2.2.8 Classifiers y Machine Learning

La comparación o pattern matching en mayoría de sistemas biométricos utilizan algoritmos de clasificación estándares [3].

Varios algoritmos han sido inventados para machine learning. Estos usan resultados de los campos de inteligencia artificial, probabilidad y estadística, teoría de información, neurobiología y otros [9].

El proceso inicia con la selección de uno o más classifiers para los datos disponibles. Luego los classifiers son entrenados (trained) con un subconjunto de la data obtenida para crear el modelo. Una vez el training está completo, se prueba el modelo para saber su precisión utilizando datos de prueba o test data que no ha sido usado en la etapa del training [3].

El siguiente paso es simplificar el modelo al identificar las medidas que proveen la información más discriminatoria y eliminar las que proveen mínima información [3].

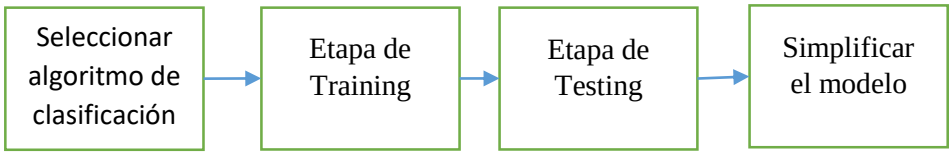

Figura 1 Workflow del problema de clasificación adaptado de [3]

Los siguientes son algunos de los algoritmos disponibles en Itus [2] (k- NN y SVM).

k-Nearest Neighbor (k-NN): Este algoritmo coloca los datos o training data en un gráfico de n dimensiones como si fueran puntos. Cuando se desea saber a qué clase pertenece el nuevo dato éste se coloca en el mismo gráfico, y luego se le asigna la clase de sus vecinos mayoritarios o neighbors, donde k representa el número de neighbors.

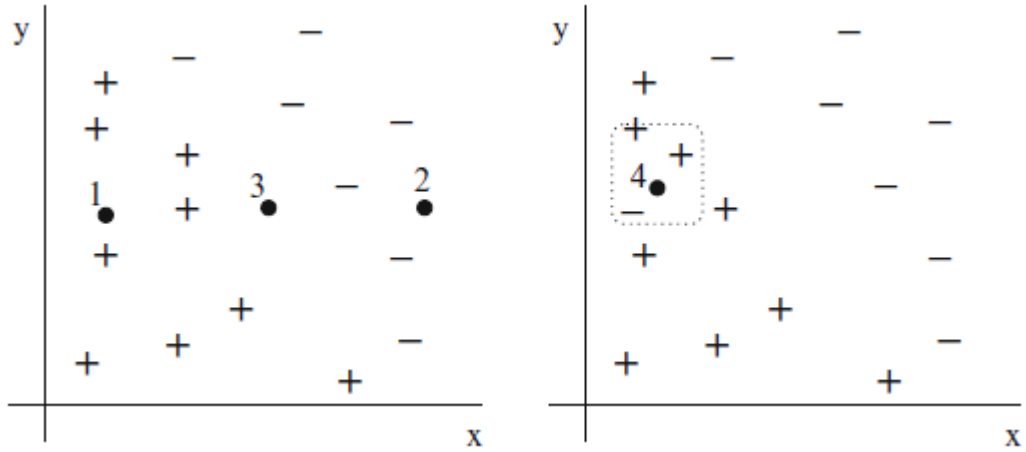

Figura 2- El objeto 3 es dificil de clasificar de acuerdo a sus vecinos. El 1-NN classifier clasificará incorrectamente al objeto 4, pero el error se corrige si se utiliza un clasificador 3-NN. Copiado de [4]

Support Vector Machine (SVM): Este clasificador se usa comúnmente en problemas de dos clases. El modelo representa datos como puntos en el espacio divididos por un hiperplano; una de las clases está en cada lado del hiperplano. Si se desea clasificar un nuevo dato éste se ubica en el mismo espacio y se predice su clase basándose en cuál lado del hiperplano se ubica.

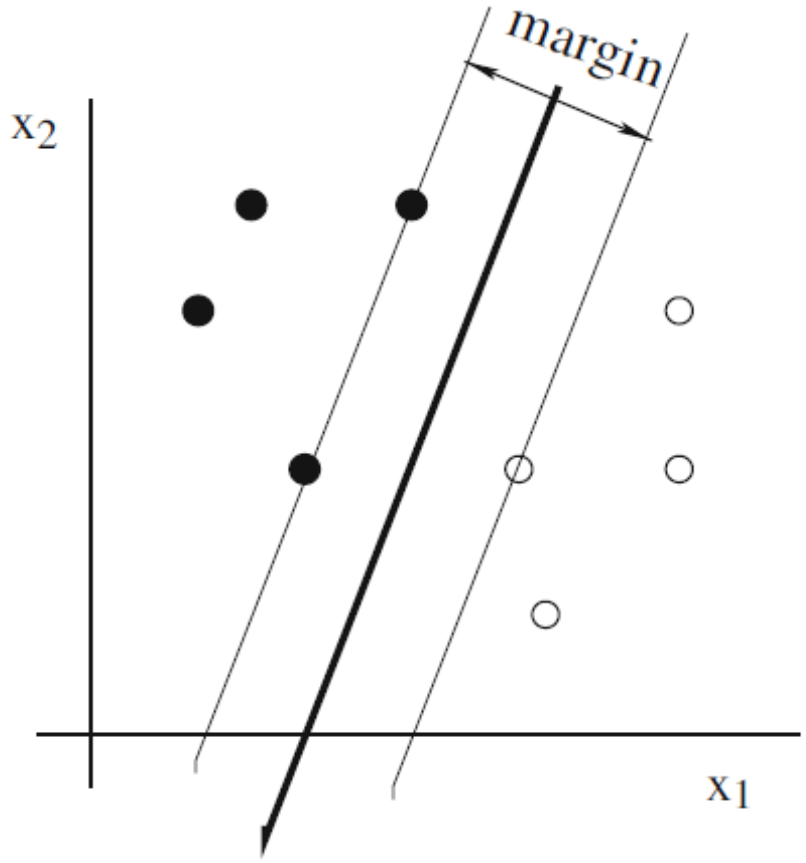

Figura 3- La técnica de support vector machine busca un hiperplano de separación que tiene el máximo margen. Copiado de [4]

Los puntos que están más cerca de la frontera de datos positivos y negativos se les conoce como support vectors y son los que influyen en el cálculo del hiperplano.

2.2.9 Novelty Detection

Novelty detection es la identificación de nueva o data desconocida en el cual un sistema de machine learning no ha sido entrenado o ha sido previamente desconocido con la ayuda de enfoques estadísticos o de machine learning[18].

[18] https://en.wikipedia.org/wiki/Novelty_detection

También se le conoce Anomaly Detection en minería de datos y Outlier Detection en estadística.

Novelty detection se puede definir como la tarea de reconocer que el test data difiere en alguna medida de la data disponible durante el training. Su importancia práctica y naturaleza retadora ha dado lugar a varios enfoques propuestos. Estos métodos son típicamente aplicados en datasets en el cual un largo número de ejemplos de la condición “normal” (también conocido como positive examples) está disponible y donde no hay datos suficientes que describan “anormalidades” (también conocido como negative examples).

Novelty detection se ha utilizado en el diagnóstico de problemas médicos, detección de fallas en complejos sistemas industriales, detección de intrusos en sistemas de seguridad electrónicos, video vigilancia, y otros [10].

El reconocimiento de patrones convencional se enfoca típicamente en la clasificación de dos o más clases. Problemas generales de multi-class clasificación frecuentemente se descomponen en múltiples problemas de clasificación binaria (two-class) [10].

El problema de novelty detection se enfoca como un problema de clasificación de una sola clase (one-class classification) en el cual esta clase (condición normal, clase positiva) debe distinguirse de todas las otras posibilidades. Usualmente se asume que hay una buena cantidad de datos de la clase positiva, mientras que datos de la otra clase(s) existen muy

pocos o son muy difíciles de conseguir. Por ejemplo, en un sistema de monitoreo de máquinas, se requiere que se encienda una alarma siempre que una máquina muestra un comportamiento “anormal”. Medidas de la máquina de su estado de operación normal no son costosos y son fáciles de obtener. Sin embargo, es difícil, sino imposible, obtener una buena muestra representativa de la clase negativa [10].

2.2.10 Métricas de Performance Biométricas

La salida del algoritmo de clasificación es sensible a muchos factores, incluyendo la elección del algoritmo, la cantidad de training data, las features[19] elegidas para el feature vector, y la variación de los datos de un participante. Esta salida puede clasificar erróneamente como impostor a un usuario autorizado y como un usuario autorizado a un impostor.

Tabla 1- Cantidades básicas usados en las definiciones de los criterios de performance. Por ejemplo NFP es el número de falsos positivos: labels negativos clasificados por el algoritmo como positivos, traducido de [4]

		Labels retornados por el algoritmo de clasificación	
		pos	neg
Labels verdaderos	pos	N_{TP}	N_{FN}
	neg	N_{FP}	N_{TN}

[19] https://en.wikipedia.org/wiki/Feature_%28machine_learning%29

False acceptance rate (FAR), o la tasa al cual un impostor es aceptado por el sistema. Digamos FA es el número de false accepts (falsos positivos) y NI el número de muestras del impostor ($N_{FP} + N_{TN}$).

$$FAR = \frac{FA}{NI}$$

False rejection rate (FRR), o tasa al cual el usuario autorizado es rechazado del sistema. Digamos FR representa el número de false rejects (falsos negativos) y NA el número de muestras del usuario autorizado ($N_{TP} + N_{FN}$).

$$FRR = \frac{FR}{NA}$$

Estas tasas de error o error rates comparten una relación mutuamente exclusiva – mientras un error rate incrementa el otro disminuye.

True acceptance rate (TAR), o la tasa al cual un usuario autorizado es aceptado por el sistema. Digamos TA es el número de true accepts (true positives) y NA el número de muestras del usuario autorizado ($N_{TP} + N_{FN}$).

$$TAR = \frac{TA}{NA}$$

True rejection rate (TRR), o tasa al cual un impostor es rechazado del sistema. Digamos TR representa el número de true rejects (true negatives) y NI el número de muestras del impostor ($N_{FP} + N_{TN}$).

$$TRR = \frac{TR}{NI}$$

Equal error rate (EER), es el punto en el cual FAR = FRR y se usa para comparar el performance de diferentes técnicas biométricas.

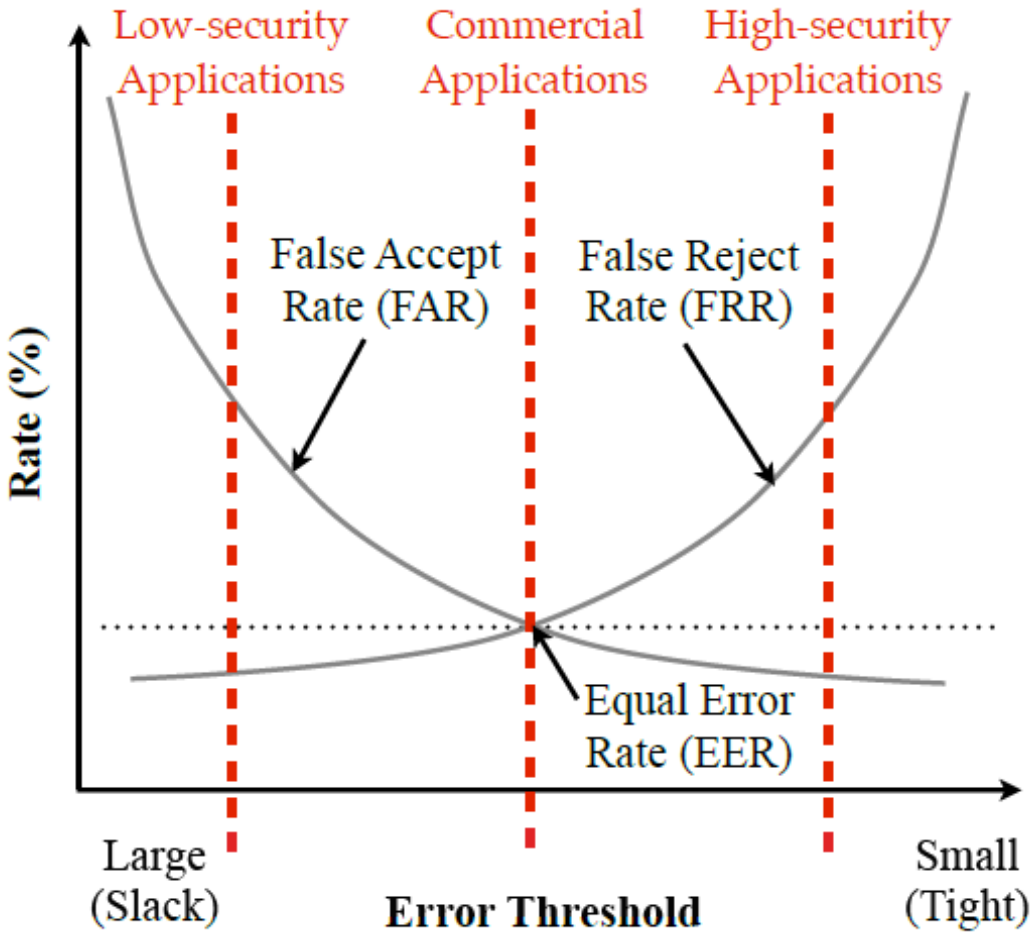

Figura 4- Métricas de performance de clasificación. Copiado de [3]

ROC Curve: A Receiver Operating Caracteristic Curve (ROC) curve, como se muestra en la figura 2.2, muestra la relación entre FAR y True Accept Rate (TAR), que es la tasa de datos que se han reconocido del usuario legítimo. La curva ROC muestra la total utilidad del algoritmo de clasificación. Mientras la curva esté más cerca de la posición superior izquierda, mejor es el algoritmo de clasificación en identificar o verificar correctamente al usuario.

Area Under Curve (AUC): AUC es una medida del area bajo la curva ROC para un algoritmo de clasificación y un usuario. Representa la probabilidad de una respuesta correcta (positiva o negativa) al hacer la clasificación – un algoritmo de clasificación elegido al azar tendrá un valor de AUC de 0.5 (50%) y un ideal algoritmo de clasificación tendrá un AUC de 1.0 (100%). AUC intenta resumir la curva ROC en un solo valor.

European Standard for Access Control Systems (EN 50133-1) establece que para que un Sistema de autenticación biométrica sea usado en producción debe tener un False Accept Rate (FAR) de menos de 0.001% y un False Rejection Rate de menos de 1%. Sin embarbo estas tasas de error sugeridas no son específicas a las técnicas biométricas de comportamiento (behavioral biometrics), que son conocidos por ser menos distintivos que las técnicas biométricas fisiológicas (physiological biometrics) [3].

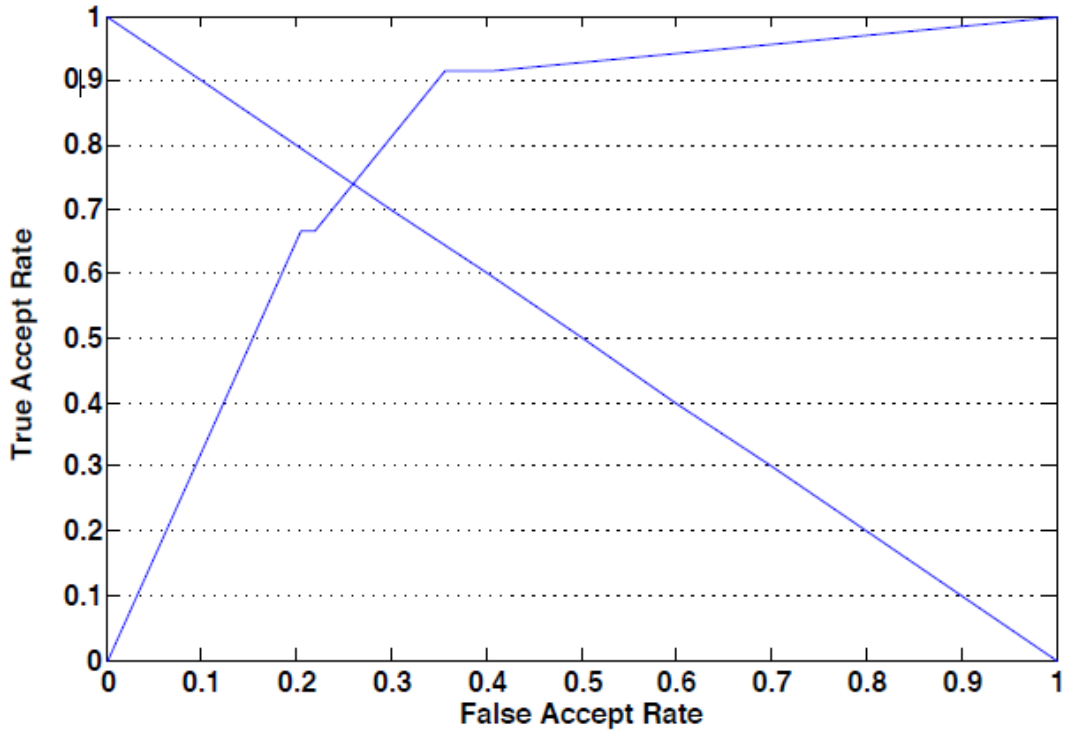

Figura 5- Un ejemplo de curva ROC. El AUC para esta curva es del 80.13%. El EER es 25.99% es el punto donde las dos lineas se cruzan. Copiado de [3]

Error Rate and Classification Accuracy: A classifier's error rate, E, es la frecuencia de errors hechos por el clasificador sobre un conjunto de datos. Se calcula dividiendo el número de errores entre el número total de datos.

$$E = \frac{N_{FP} + N_{FN}}{N_{FP} + N_{FN} + N_{TP} + N_{TN}}$$

Accuracy es la frecuencia de clasificaciones correctas hechos por un clasificador dado un conjunto de datos y se calcula dividiendo el número de correctas clasificaciones entre el número total de datos. Debe notarse que *Acc = 1 – E.*

$$Acc = \frac{N_{TP} + N_{TN}}{N_{FP} + N_{FN} + N_{TP} + N_{TN}}$$

Precision: Representa la probabilidad de que tan correcto es el clasificador cuando clasifica (o etiqueta) un dato como positivo:

$$Pr = \frac{N_{TP}}{N_{FP} + N_{TP}}$$

Recall: Representa la probabilidad de que un dato positivo será correctamente reconocido como tal por el clasificador

$$Re = \frac{N_{TP}}{N_{FN} + N_{TP}}$$

2.2.11 N-fold crossvalidation

N-fold crossvalidation es una forma de organizar el dataset para calcular las medidas de performance del algoritmo de clasificación [4]. El principio se muestra en la figura 6. Para empezar, el conjunto de datos pre-clasificados se divide en N partes iguales (o casi iguales) a estos subconjuntos se le conoce como folds en la jerga de machine-learning.

N-fold crossvalidation luego ejecuta los N experimentos. En cada uno de los experimentos uno de los N datasets es removido para ser utilizado sólo como testing (esto garantiza que, en cada ejecución, un diferente testing set es usado). Luego se realiza el training en la unión de los N-1 datasets

restantes. Al final, los resultados son promediados y se calcula la desviación estándar.

Los datos de cada subset pueden ser elegidos al azar (Random subsampling) o cada subset puede contener un porcentaje de cada clase (Stratified approach).

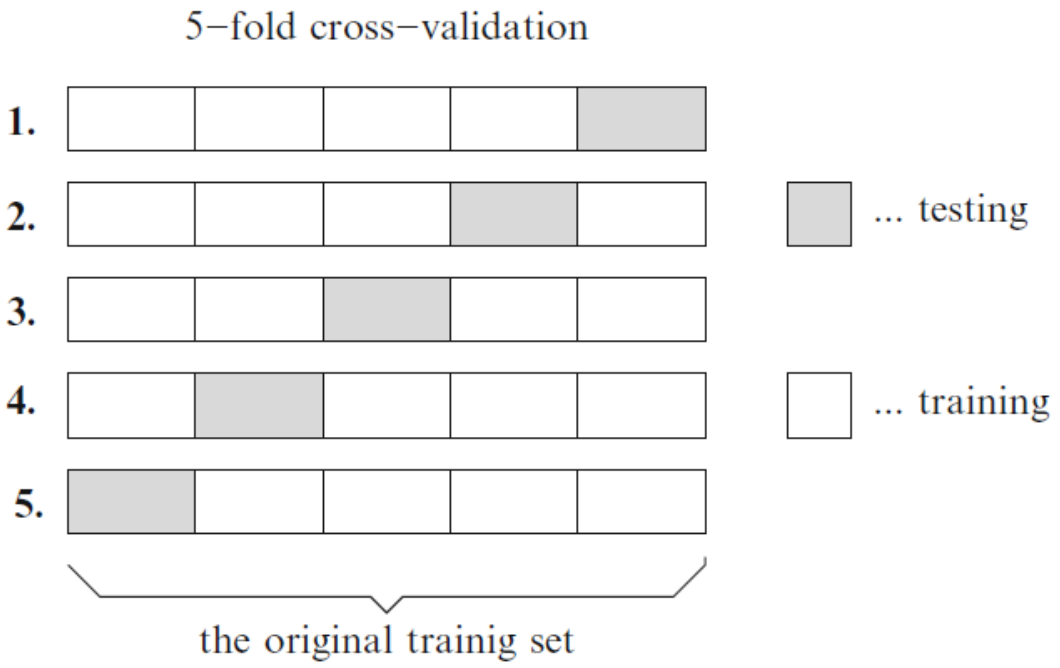

Figura 6 N-fold cross-validation divide el training set en N subsets iguales. En cada uno de las ejecuciones experimentales, utiliza un diferente subconjunto para testing, entrenando el clasificador en la unión de los N-1 sets restantes. Copiado de [4]

One-Class classification: En el caso de problemas que presentan datos de una sola clase (o un solo label) el procedimiento para hacer el N-fold cross validation no está establecido o hay muy poca información al respecto. La principal diferencia con el normal k-fold cross-validation es que, los outliers si están disponibles estos también se dividen en N sets, pero no se usan en el

training [12]. Aunque en [12] se realiza el testing en la unión de un set de la clase original y un subset del outlier, la desventaja de este procedimiento es que no permite calcular las medidas TAR, FRR, TRR, FAR, la ventaja es que permite el cálculo de medidas únicas de performance como Accuracy (no confundir con Precision [4]) y AUC. En cambio en [13] se realizan dos testings separados: uno con el set de la clase original y otro con un subset de los outliers. La ventaja de este último procedimiento es que permite calcular las medidas de performance de interés (TAR, FRR, TRR, FAR). La Figura 7 muestra el procedimiento de cross-validation a realizar en esta investigación.

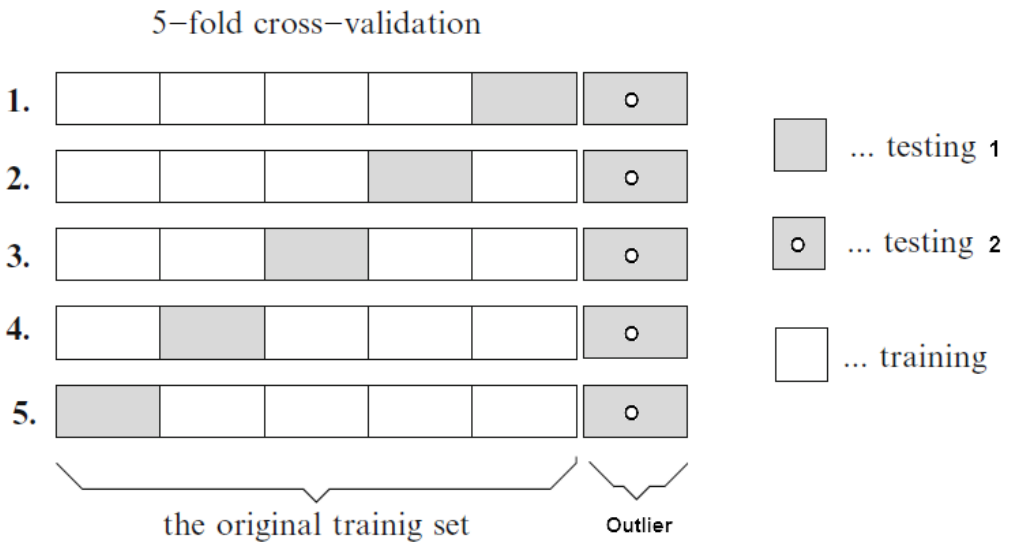

Figura 7 5-fold cross-validation a utilizar para el problema de one-class classification [13].

2.2.12 The no-free-lunch teorem

No se puede esperar que una técnica de machine learning sea superior a otra o que sea la solución para cualquier tipo de problema. El performance de cada técnica de machine learning depende mucho del contexto. Cada paradigma tiene sus ventajas que le permiten tener éxito en determinados dominios y desventajas que hacen que su performance sea muy bajo en otros dominios. Sólo con experimentos sistemáticos se puede determinar el algoritmo de clasificación más apropiado para el problema a tratar.

Matemáticos han probado la validez de este enunciado mediante una prueba rigurosa. El resultado es conocido con el nombre de “no-free-lunch teorem”

2.2.13 LIBSVM: A Library for Support Vector Machines

Support Vector Machines (SVMs) son un popular método para clasificación, regresión, y otras tareas en machine learning. Desde el año 2000, los autores han ido desarrollando el paquete software LIBSVM como una librería para support vector machines. La dirección web del paquete es http://www.csie.ntu.edu.tw/~cjlin/libsvm. LIBSVM es actualmente uno de los SVM software más ampliamente utilizados [11].

LIBSVM soporta las siguientes tareas:

1. SVC: support vector classification (two-class and multi-class).
2. SVR: support vector regression.

3. One-class SVM.

Para poder realizar el training y testing LIBSVM requiere la data en el siguiente formato:

```
<label> <index1>:<value1> <index2>:<value2> ...
```

Cada línea contiene una instancia y es terminado por el carácter retorno de carro '\n', <label> es un entero indicando el class label (soporta multi-clase). Para one-class SVM <label> no es usado entonces puede ser cualquier número[20].

El parámetro "-n nu" corresponde al límite en fracción de puntos que se volverán support vectors (SVs), por lo tanto limita la complejidad del modelo (mientras más pequeño es el número de SVs, más simple el modelo), esto correspondería directamente a[21]:

"Un límite superior en la fracción de outliers"

"Un límite inferior en la fracción de SVs"

Básicamente "-n 0.01" significa 1% de la data puede ser rechazado y marcado como outlier[22]. No existe una forma establecida de calcular el óptimo 'nu' así que solo se prueban diferentes valores.

[20] https://github.com/cjlin1/libsvm/blob/master/README
[21] http://stats.stackexchange.com/a/72518
[22] http://stackoverflow.com/a/22207480/403999

3. MARCO METODOLÓGICO

3.1 Tipo de Investigación

Es una **Investigación Aplicada** porque trata de resolver el problema de Autenticación Continua e Implícita en dispositivos móviles utilizando conceptos y técnicas de Machine Learning.

3.2 Diseño de la Investigación

Es un **Diseño No Experimental – Transversal Descriptivo**.

Es No Experimental porque no se manipula la variable independiente, en este caso se utilizará un solo algoritmo de clasificación y durante la recolección de datos de uso del dispositivo móvil de los usuarios el investigador se limitó a observar y no ha intervenido en la generación de estos datos. **Es Transversal** porque para esta investigación es suficiente la recolección de los datos una sola vez durante dos semanas. **Es Transversal Descriptivo** porque se analiza que tan discriminativo son los datos de uso del dispositivo como para diferenciar al usuario propietario de un intruso.

3.3 Población y Muestra

3.3.1 Población

La población lo constituyen los usuarios que poseen un dispositivo móvil Android.

3.3.2 Muestra

Ya que es imposible recolectar datos de todos los propietarios de un dispositivo móvil Androide se realiza un **Muestreo no Probabilístico** por conveniencia. Se recolectan datos de colaboradores voluntarios que poseen un dispositivo móvil Android.

3.4 Fuentes y Técnicas de Recojo de Datos

3.4.1 Fuentes

Los datos se obtienen directamente de los usuarios que participan en el estudio. Al finalizar un periodo de 2 semanas los usuarios enviarán el archivo con los datos de uso de la pantalla táctil vía correo electrónico o vía bluetooth al investigador.

3.4.2 Técnicas

Se utiliza un aplicativo Android[23] para recolectar datos automáticamente en un archivo mientras el usuario utiliza su dispositivo móvil para realizar sus tareas cotidianas.

[23] https://github.com/poseidonjm/TurboLauncher-Itus

3.4.3 Instrumentos de Recolección de Datos

Los datos a recolectar corresponden a eventos de sistema generados por la interacción del usuario con la pantalla táctil.

Tabla 2– Datos de uso de pantalla táctil del dispositivo móvil a recolectar

Características Touch	**Definición**	**Número de medidas**
Posición	Inicio X/Y Fin X/Y del evento touch	4
Características Temporales	Duración del evento touch, tiempo entre touchs consecutivos	2
Longitud	Distancia directa inicio final, media, longitud de trayectoria	3
Velocidad Lineal	Porcentaje de velocidad en pares, velocidad promedio, mediana de los últimos 3 puntos	5
Aceleración Lineal	Porcentaje de aceleración en pares, mediana de los primeros 5 puntos	4

Presión	Presión durante el trazo touch	1
Angulo	Dirección de la línea, promedio, mayor desviación, porcentaje de desviación, orientación del teléfono	8
Área	Área del primer touch, área de cobertura de línea	2

El aplicativo móvil Android que se utiliza para la recolección automática de datos utiliza dos proyectos open source TurboLauncher[24] e Itus[25] para guardar la información de uso de la pantalla táctil del usuario mientras éste utiliza la aplicación. Este aplicativo tiene un peso de 1.98MB.

Los datos de eventos touch se guardan en el sistema de archivos de Android en la memoria del dispositivo utilizando Itus en Modo Config.

[24] https://github.com/Phonemetra/TurboLauncher
[25] https://crysp.uwaterloo.ca/software/itus/

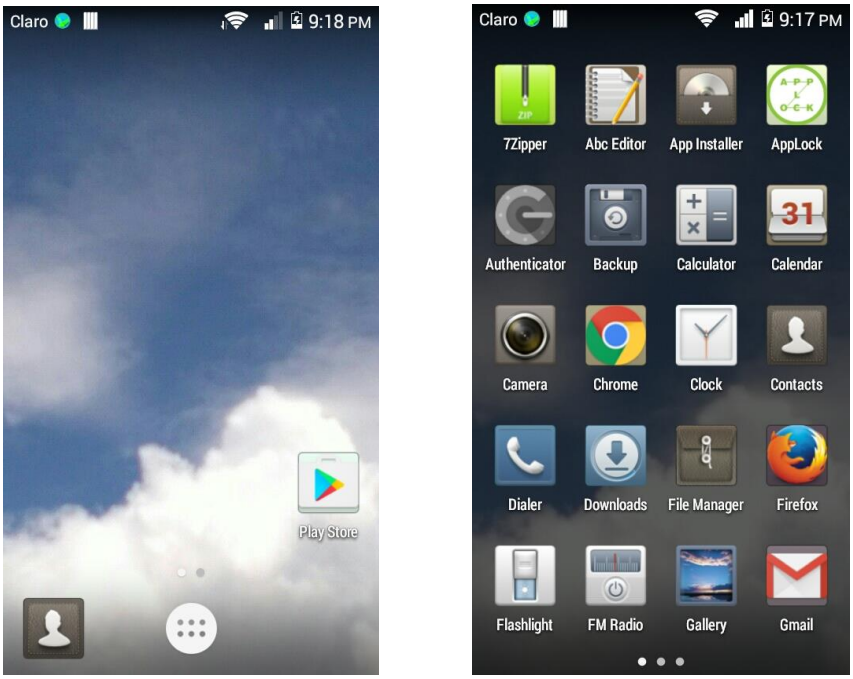

Figura 8- Aplicativo TurboLauncher

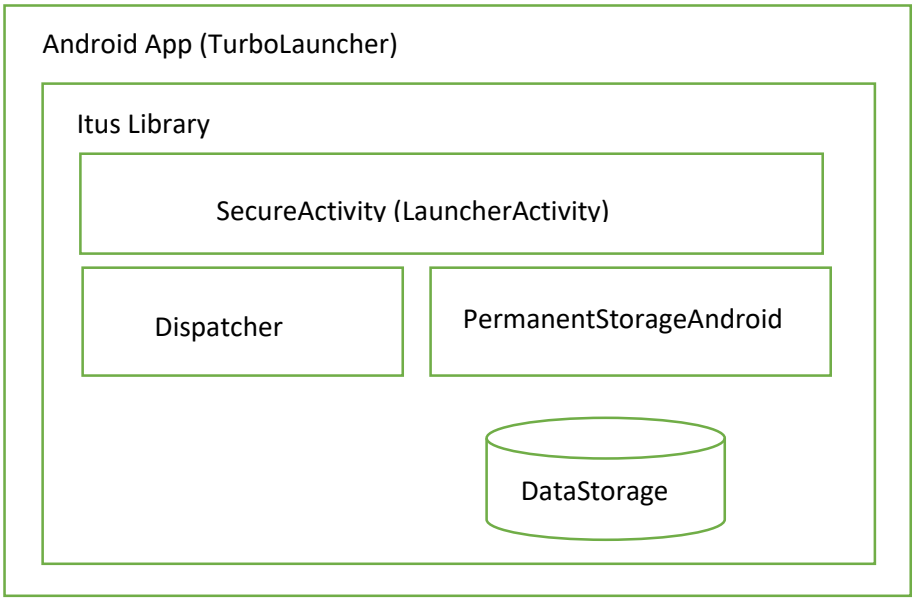

Figura 9- Recolección de datos en TurboLauncher utilizando Itus en Config Mode

Cada usuario instaló el aplicativo móvil en su propio dispositivo con Sistema Operativo Android.

Tabla 3- Dispositivos utilizados para la recolección de datos

Model Number	**CPU**	**RAM**	**Phone storage**	**Android version**
HUAWEI Y550	Quad core 1.2 GHz	1.0GB	163 MB free, 4.00 GB total	4.4.4
HUAWEI CUN-L03	Quad core 1.3 GHz	1.0GB	2.9 GB free, 8.00 GB total	5.1
Samsung Galaxy S4 Mini Black Edition	dual-core 1.7Ghz	1.0GB	8.00 GB total	4.2
SM-G357M	dual-core 1.2GHz	512MB	1.23GB free, 8.00 GB Total	4.4.2
LG-D625	Quad core 1.7 GHz	1.0GB	1.03 GB free, 4.41 GB total	4.4.2
ZTE Blade V6	Quad core 1.3 GHz	2GB	16 GB total	5.0.2
LG-H635C	Quad core 1.2 GHz	1.0GB	3.4 GB total	5.0.2
Lenovo A2010I36	Quad core 1.0 GHz	1.0GB	8.00 GB Total	5.1

SM-J500M	Quad core 1.2 GHz	1.5GB	386 MB free, 8.00 GB total	5.1.1
SM-J200M	Quad core 1.3 GHz	1.0GB	8.00 GB total	5.1.1
ALE-L23	Octa core 1.2 GHz	2GB	2.49 GB free, 16.00 GB total	5.0.1
SM-J105B	Quad core 1.2 GHz	764 MB	8.00 GB total	5.1.1
MOTO G	Quad core 1.2 GHz	1.0GB	16.00 GB total	4.4.4

Luego de instalar el aplicativo móvil el usuario debe realizar un paso adicional.

Para configurar TurboLauncher como el aplicativo Launcher por defecto en Android se debe ir a *Configuración o Ajustes (Settings) ->Dispositivo (Device) –Inicio(Home) y Seleccionar Turbo Launcher*. En el caso del modelo HUAWEI Y5 II es un poco diferente similar al HUAWEI HONOR[26]: *Settings -> Manage Apps->Default App Settings -> Launcher*. Si no se encuentra la opción Home o Inicio se recomienda cambiar el idioma del dispositivo a Inglés temporalmente y buscar la configuración correcta utilizando tu buscador preferido

[26] http://www.talkandroid.com/guides/misc/how-to-change-launchers-on-your-huawei-or-honor-device/

(Por ejemplo: Google) y buscar “How to change launchers on your Huawei” indicando el modelo de tu dispositivo.

Con Los datos extraídos de los eventos touch la librería Itus forma el Feature Vector:

$$f_{user} = (Start\ X, Start\ Y, End\ X, End\ Y, \dots, Phone\ Orientation, First\ touch\ area) \quad (3.1)$$

Cada uno de estos Feature Vectors se forma por cada evento touch que el usuario genera al interactuar con la pantalla táctil.

El resultado final de la recolección de datos son 13 distintos grupos de todo el dataset, cada uno de los cuales contiene un dataset del propietario y un dataset del resto. El aplicativo no logro recolectar la data de 2 participantes con Smartphone Moto G 2nd Generation y Huawei P9, no se investigó las razones por la cual el app no pudo crear el archivo en la memoria del dispositivo.

4. PRESENTACIÓN DE DATOS, RESULTADOS DE LA INVESTIGACIÓN

En total se recolectaron 19832 eventos touch de trece participantes, en la tabla 3 se muestran los datos de los eventos touch capturados por cada uno de los dispositivos. La mayor cantidad de datos 1952 KB se obtuvo del celular de modelo MOTO G, con un total de 2952 eventos touch. La menor cantidad de datos 58 KB se obtuvo del celular de modelo ZTE Blade V6, con un total de 92 eventos touch, pero esto se debe a que el usuario desinstalo el aplicativo por un supuesto problema con el WIFI.

Tabla 4- Número de eventos touch recolectados

Model Number	**# de Eventos Touch**	**Data Storage Size (KB)**
HUAWEI Y550	1174	709 KB
HUAWEI CUN-L03	1566	861 KB
Samsung Galaxy S4 Mini Black Edition	232	139 KB
SM-G357M	1703	1031 KB
LG-D625	2798	1508 KB
ZTE Blade V6	92	58 KB
LG-H635C	2041	1340 KB
Lenovo A2010I36	1581	946 KB
SM-J500M	1447	833 KB

SM-J200M	1593	945 KB
ALE-L23	2112	1271 KB
SM-J105B	541	333 KB
MOTO G	2952	1952 KB
Total	**19832**	**11926 KB**

El tamaño pequeño de la data recolectada muestra que no sería un problema para el dispositivo almacenar los datos necesarios para la autenticación implícita ya que los dispositivos Android actuales poseen capacidades de almacenamiento de 4GB, 8GB o 32GB.

4.1 Métricas de performance

Antes de realizar el training es necesario escalar o normalizar la data para que cada medida tenga la misma influencia en el performance. LIBSVM incluye una herramienta svm-scale que facilita la normalización de la data en rangos [0,1] o [-1,1]. Por simplicidad se utilizará la herramienta que viene incluido en LIBSVM.

Sin embargo para mejorar el performance se podría utilizar otras formas de normalización para escalar la data[27] o utilizar un enfoque de medidas como el propuesto por [13] o un enfoque propio.

El valor del parámetro de configuración “-n nu” se determinó probando varios valores de éste realizando los Testing 1 y 2 con los datos del

[27] https://en.wikipedia.org/wiki/Feature_scaling

dispositivo HUAWEI CUN-L03 (owner) y un subconjunto de los datos de los otros dispositivos (outlier) respectivamente.

La guía LIBSVM [14] también recomienda calcular dos parámetros adicionales de configuración '-c' costo y '-g' gamma para mejorar el performance. Al realizar el training y testing indicando que se calculen los mejores parámetros '-c' costo y '-g' gamma se obtiene una disminución de 0.18612 % del FAR y aporta 0.0641% en el incremento del TAR. Además el incremento del tiempo es bastante considerable (41.027s) si se calculan los parámetros '-c' costo y '-g' gamma comparado a si no se calculan estos parámetros. Por lo tanto ya que el aporte en el performance es muy poco y la penalidad de tiempo es bastante, no se calcularán los parámetros mencionados y se utilizarán los valores por defecto de –c y –g (gamma)[28].

También se ha realizado el training y testing utilizando 5-fold cross validation, se hizo pruebas con 10-fold cross validation pero no se obtuvo un incremento considerable del performance.

[28] https://github.com/cjlin1/libsvm

Tabla 5 – Resultados de clasificación promediados obtenidos utilizando 5-fold cross-validation con nu = 0.1. Desviaciones estándares en paréntesis

Propietario (owner)	**TAR % (Testing 1)**	**FAR % (Testing 2)**
HUAWEI Y550	87.23118 (6.27720614)	30.995452 (19.85797509)
LG-D625	89.5621 (2.71619798)	39.8319 (15.09629385)
Samsung Galaxy S4 Mini	88.46016 (7.39091635)	63.63778 (16.45383656)
Samsung SM-G357M	90.19548 (3.64679905)	25.412088 (14.33209018)
HUAWEI CUN-L03	89.01732 (3.53009308)	83.62596 (7.93932903)
ZTE Blade V6	82.77776 (15.5158175)	70.08612 (10.16758917)
LG-H635C	89.36826 (4.34423126)	83.82418 (9.47331323)
Lenovo A2010I36	87.7263 (8.87351827)	41.197418 (34.15706)
SM-J500M	89.49096 (2.57740134)	95.21892 (3.15772682)
SM-J200M	89.01192 (2.79673829)	50.9287 (30.9870702)
ALE-L23	89.34366 (3.96865679)	59.67268 (18.08883491)
SM-J105B	89.47162 (2.44393645)	34.363592 (24.58168583)
MOTO G	87.34792 (14.62065255)	79.60306 (20.32204438)
Mean (Standard Deviation)	88.384972 (1.915583)	58.338296 (23.034385)

Tabla 5 muestra que algunos propietarios tienen un patrón táctil muy distintivo (por ejemplo; los 2 modelos Samsung SM-G357M y SM-J105B presentan la tasa más baja de aceptación de intruso FAR) pero los patrones táctiles de los propietarios de los otros modelos son similares al del resto (por ejemplo; los modelos SM-J500M y HUAWEI CUN-L03 presentan una tasa alta de aceptación de intruso FAR).

Tabla 5 también muestra resultados alentadores al presentar una tasa alta de aceptación de propietario TAR del 88 % esto significa que la pantalla de login se le presentará al propietario sólo un 12 % (FRR) de las veces.

A continuación se intenta obtener una única medida de performance del algoritmo de clasificación.

Tabla 6 - Resultados de clasificación promediados obtenidos utilizando 5-fold cross-validation con nu = 0.1. Desviaciones estándares en paréntesis.

Propietario (owner)	**Testing 1 (owner)**		**Testing 2 (outlier)**	
	N_{TP}	N_{FN}	N_{FP}	N_{TN}
HUAWEI Y550	204.8 (14.44645285)	30 (14.7478812)	1156.6 (740.85781362)	2575 (741.01855577)
LG-D625	501.2 (15.6588633)	58.4 (15.14265499)	1357 (514.20472577)	2049.8 (514.14365697)
Samsung Galaxy S4 Mini	41 (2.82842712)	5.4 (3.57770876)	2494.6 (644.99015496)	1425.4 (644.99015496)
Samsung SM-G357M	307.2 (12.27599283)	33.4 (12.42175511)	921.4 (519.55731156)	2704.4 (519.48801719)
HUAWEI CUN-L03	278.8 (10.98635517)	34.4 (11.05893304)	3055 (289.75506898)	598.2 (290.12790283)
ZTE Blade V6	15.2 (2.68328157)	3.2 (2.86356421)	2767 (401.41686561)	1181 (401.41686561)
LG-H635C	364.8 (17.69745744)	43.4 (17.72850812)	2982.6 (336.77781993)	575.6 (337.19178519)
Lenovo A2010l36	277.4 (28.16558183)	38.8 (28.0303407)	1503.8 (1246.75326348)	2146.4 (1246.71019086)
SM-J500M	259 (8.06225775)	30.4 (7.40270221)	3501.2 (116.10856988)	175.8 (116.10856988)
SM-J200M	283.6 (9.28977933)	35 (8.86002257)	1857.6 (1129.89304804)	1790.2 (1130.61076414)
ALE-L23	377.4 (17.12600362)	45 (16.7182535)	2114.8 (641.06840509)	1429.2 (641.06840509)
SM-J105B	96.8 (2.28035085)	11.4 (2.70185122)	1325.8 (948.33944345)	2532.4 (948.4235868)
MOTO G	515.6 (85.72222582)	74.8 (86.5950345)	2687.4 (686.07200788)	688.6 (686.07200788)
Mean (Standard Deviation)	270.984615 (154.843286)	34.123077 (20.000090)	2132.676923 (838.207033)	1528.615385 (849.397396)

Con los datos de la tabla 6 se pueden calcular las siguientes medidas de performance:

Accuracy:

$$Acc = \frac{N_{TP} + N_{TN}}{N_{FP} + N_{FN} + N_{TP} + N_{TN}}$$

$$Acc = 0.45371117386042764$$

Accuracy indica que el algoritmo clasificara correctamente el 45% de los datos. Este resultado está influenciado debido a que hay más datos del usuario intruso.

Error Rate:

$$E = 1 - Acc$$

$$E = 1 - 0.45371117386042764$$

$$E = 0.5462888261395724$$

Error Rate indica que el algoritmo clasificará incorrectamente el 55% de los datos. Este resultado está influenciado debido a que hay más datos del usuario intruso.

Precision:

$$Pr = \frac{N_{TP}}{N_{FP} + N_{TP}}$$

$$Pr = 0.11273825815987244$$

Precision indica que el algoritmo tiene una probabilidad de 0.1 de estar correcto al clasificar un dato como positivo. Esta baja probabilidad se

debe a que no se obtuvo un buen performance en la detección de outliers como consecuencia el número de falsos positivos es mayor.

Recall (owner):

$$Re = \frac{N_{TP}}{N_{FN} + N_{TP}}$$

$$Re = 0.8881605482434051$$

Recall (owner) indica que un dato positivo será correctamente reconocido como tal por el clasificador con una probabilidad de 0.89.

Recall (outliers):

$$Re = \frac{N_{TN}}{N_{FP} + N_{TN}}$$

$$Re = 0.4175070593680662$$

Recall (outlier) indica que un dato negativo será correctamente reconocido como tal por el clasificador con una probabilidad de 0.42.

Recall sólo se basa en la clase de interés (positivo o negativo) así que representa una medida de performance adecuada para el presente estudio en el que se realizaron testings separados en datos positivos (owner) y negativos (outlier).

4.2 Limitaciones del estudio

Hay varias limitaciones potenciales en este estudio. Los resultados se deben evaluar teniéndolos en cuenta.

Dispositivos: Los dispositivos usados en el estudio fueron provistos por los usuarios que participaron en el estudio. Esto significa que el experimentador no tenía control sobre la recolección de datos, es posible que el propietario presto su Smartphone a otra persona por un corto periodo de tiempo generando ruido en la data recolectada. Pero se consideró que el Smartphone es un dispositivo muy personal y que es poco probable que otro usuario lo use por un largo periodo de tiempo. También a los participantes se les indicó que no desinstalen el aplicativo durante el periodo de prueba.

Diferencias en la pantalla: Los Smartphones utilizados en el estudio tienen diferentes resoluciones y tamaños de pantalla el cual puede afectar como usan la pantalla táctil.

Número de participantes: En el presente estudio sólo se utilizaron 13 participantes en comparación con el estudio [10] donde los autores utilizaron una muestra de 51 participantes recolectados durante una sesión. Sin embargo el presente estudio es similar al estudio [3] realizado por Crawford donde utiliza 8 participantes y recolecta la data por un periodo de 2 semanas utilizando un aplicativo para IOS.

4.3 Contribuciones de la investigación

El problema planteado al inicio de la investigación es el siguiente

¿Con la autenticación continua e implícita se puede lograr una mejor seguridad en dispositivos móviles?

Los resultados de la Tabla 5 muestran que sí es posible mejorar la seguridad de dispositivos móviles utilizando autenticación continua e implícita.

Se plantearon los siguientes problemas específicos:

- ✓ ¿Qué algoritmo de clasificación sería apropiado para su uso en dispositivos móviles respetando sus limitaciones de hardware?

Se eligió el algoritmo Support Vector Machines ya que el estudio [2] utiliza la librería LIBSVM[29] y muestra que este tiene un bajo consumo de memoria lo que lo hace apropiado para su uso en dispositivos móviles. Otra consideración que se tomó en cuenta es que la librería soporta One-Class classification que lo hacen adecuado para problemas de Novelty Detection y en especial para el problema de seguridad en dispositivos móviles donde se cuenta con una gran cantidad de datos del usuario propietario y no existen o no se cuenta con datos del usuario impostor. Otra razón para el uso de SVM es que no hay mucha información sobre Novelty Detection en la actualidad y no está establecido o no hay un

[29] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

consenso sobre que modificaciones se deben hacer a las técnicas existentes para poder resolver problemas en los que existe una sola clase (la de la condición normal) [10].

- ✓ ¿Los datos uso de la pantalla táctil del dispositivo móvil presentan características distintivas para identificar al usuario?

Los resultados de la Tabla 5 muestran que los datos de uso de la pantalla táctil sí son distintivas con un performance por encima del 80% para identificar al usuario propietario.

- ✓ Utilizar un proyecto open source que implementa el algoritmo seleccionado en lenguaje de programación Java.

El proyecto open source utilizado es LIBSVM[30] por razones ya mencionados anteriormente.

- ✓ ¿Cuál es la efectividad de la autenticación continua e implícita en dispositivos móviles utilizando eventos touch de la pantalla táctil?

Con los resultados de la Tabla 6 se ha obtenido un 88.81% de efectividad para reconocer al propietario y un 41.75% para reconocer al impostor. Utilizando “Recall" como medida de performance.

4.3.1 Principales contribuciones

La presente investigación presenta algunas contribuciones que pueden ayudar a resolver problemas de Novelty Detection utilizando LIBSVM.

[30] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

1. Se muestra un código de ejemplo sobre el uso de un aplicativo Android para recolectar datos de eventos touch[31].
2. Se pone a disponibilidad del público el dataset recolectado para que otros usuarios puedan realizar pruebas de clasificación[32].
3. Se propone un método para realizar N-fold cross-validation para problemas de una sola clase (One-Class) en el cual se realizan 2 testings por cada fold (Figura 7).
4. Se ha desarrollado un programa en lenguaje de programación python easy-one-class.py[33] que ayuda a calcular medidas de performance de problemas de una sola clase (One-Class) utilizando LIBSVM.

4.4 Trabajos Futuros

En el desarrollo de esta tesis se busca como mejorar la seguridad de dispositivos móviles haciendo uso de la autenticación continua e implícita. Sin embargo al tratar de buscar la solución se encontró que esta es muy amplia y compleja para ser resuelta en una sola Tesis. Es por eso que se dejan varios temas pendientes para investigaciones futuras.

1. Usar una muestra más grande para recolectar datos de uso de pantalla táctil del dispositivo móvil utilizando el aplicativo Itus y

[31] https://github.com/poseidonjm/TurboLauncher-Itus
[32] https://github.com/poseidonjm/TurboLauncher-Itus/tree/master/dataset
[33] https://github.com/poseidonjm/TurboLauncher-Itus/tree/master/tools

mejorar el performance de SVM one-class classification con LIBSVM[34] en la detección de intrusos. Adicionalmente se puede comparar con otra técnica de clasificación.

2. Extender Itus[35] e incluir el soporte de SVM one-class ya que el disponible utiliza two-class classification y éste no se considera adecuado ya que en un entorno real sólo existe una clase de datos el del propietario.
3. Actualmente Itus guarda una pequeña cantidad de datos del propietario para realizar el training esto se podría modificar para que guarde una cantidad mayor. Además Itus utiliza un Thread para realizar la autenticación continua y cada cierto tiempo la aplicación deja de funcionar durante la etapa de clasificación. Esto se podría mejorar iniciando el Thread dentro de un Android Service. Otra desventaja de Itus es que soporta un solo Activity. No hay ejemplos sobre cómo realizar el training/classification cuando se usan múltiples Activities.
4. Itus está pensado para soportar Biometric Fusion pero según el modelo de clases sólo soportaría un tipo Fusión Biométrica: Feature Fusion que es la fusión de los feature vectors. Pero no hay ejemplos de cómo se podría utilizar esta funcionalidad. Existen otras formas de

[34] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[35] https://crysp.uwaterloo.ca/software/itus/

Fusión Biométricas (ver sección 2.2.5 Multimodal Biométrics) que se podrían incluir en Itus.

5. De acuerdo al paper de Itus éste soportaría “re-training” es decir volver a entrenar el algoritmo con nueva data del usuario legítimo que el algoritmo no pudo reconocer como tal. Sin embargo en el código fuente de Itus no existe evidencia de esta funcionalidad. En un futuro proyecto se puede incluir “re-training” en Itus y se podría evaluar cómo ésta funcionalidad afectaría el performance de clasificación y cuál sería la penalidad en el consumo de recursos hardware al hacer uso de esta funcionalidad.
6. Una vez que los anteriores ítems se hayan investigado/desarrollado se puede incluir Itus en el desarrollo de un aplicativo Android para evaluar cómo se mejora la seguridad. De esta forma se lograría construir un prototipo funcional que implementaría autenticación continua e implícita. No se recomienda incluir Itus en un proyecto existente para realizar la autenticación continua ya que éstos proyectos si es que son open source presentan una gran complejidad como para construir versiones seguras de éstos. Según Gary McGraw especialista en seguridad de software: “El desarrollador de software también debe ser especialista en seguridad de software, seguridad se construye desde cero junto con el software, seguridad no es una funcionalidad que se agrega al final si es que se dispone de tiempo”.

4.5 Conclusiones

Esta investigación al iniciar se enfocó desde el punto de vista de desarrollo de software con la esperanza de desarrollar un prototipo de prueba que implemente autenticación continua y transparente (implícita). Sin embargo la creación de este tipo de software presenta una gran complejidad pues prácticamente no existen implementaciones conocidas excepto Itus. Es posible que esto se deba a que la autenticación continúa e implícita es aún un tema en desarrollo y que aún le falta mucho por madurar. Además no existe mucha información sobre Novelty Detection para la autenticación. Muchos trabajos sobre autenticación continua tratan el problema de la autenticación como un problema de 2 clases en el cual se dispone de muestras de datos de cada una de las clases adecuadamente balanceados por igual sobre el cual se realiza en training y testing lo cual es muy poco realista. Es por eso que esta investigación enfoca el problema de la autenticación desde el punto de vista de Novelty Detection ya que es el enfoque que se aproxima más a la realidad.

En esta Tesis parece que se critica mucho a Itus, Sin embargo Itus es uno de los pocos intentos (si es que no es el único) de tratar de implementar la autenticación continua en dispositivos móviles. El objetivo de Itus se centra en la implementación y no en la precisión o performance de clasificación. Es por eso que el performance de clasificación no es un

motivo válido para criticar a Itus pues en primer lugar no es éste su objetivo sino evaluar el impacto de la autenticación continua en dispositivos móviles en cuanto al consumo de recursos hardware y ha demostrado que si es viable y posible el uso de la autenticación continua en un dispositivo con recursos limitados como es el dispositivo móvil Android. Itus puede no ser un proyecto terminado y completamente funcional pero es un buen comienzo el cual otros investigadores pueden continuar y mejorar.

5. BIBLIOGRAFIA

1. Clarke, Nathan (2011) Transparent User Authentication Biometrics, RFID and Behavioral Profiling, DOI: 10.1007/978-0-85729-805-8 Publisher: Springer
2. Hassan Khan, Aaron Atwater and Urs Hengartner (2014) Itus: An Implicit Authentication Framework for Android. Available at http://www.cs.uwaterloo.ca/~uhengart/publications/mobicom14.pdf DOI: http://dx.doi.org/10.1145/2639108.2639141
3. Crawford, Heather Anne (2012) “A Framework for Continuous, Transparent Authentication on Mobile Devices”. PhD thesis, University of Glasgow. Available at http://theses.gla.ac.uk/4046/

4. Miroslav Kubat (2015) An Introduction to Machine Learning, DOI: 10.1007/978-3-319-20010-1 Publisher: Springer
5. Karatzouni, Sevasti (2014): “Non-Intrusive Continuous User Authentication for Mobile Devices”. PhD Thesis, University of Plymouth, 2014, available at https://www.cscan.org/?page=studentprofile&id=147
6. Lalit Agarwal (2016): Evaluating Re-authentication Strategies for Smartphones. Master Thesis, University of Waterloo, available at http://hdl.handle.net/10012/10611
7. Hassan Khan (2016): Evaluating the Efficacy of Implicit Authentication Under Realistic Operating Scenarios. PhD Thesis, University of Waterloo, available at http://hdl.handle.net/10012/10621
8. Catálogo de Servicios Biométricos del RENIEC, disponible en https://serviciosbiometricos.reniec.gob.pe/portal/resources-1.2.3/bio/CatalogoServiciosBiometricos_v1.0.2.pdf
9. Juraj Figura (2012): Machine Learning For Google Android. Bachelor Thesis, Charles University in Prague, available at http://www1.cuni.cz/~obo/vyuka/projekty/figura-ml-for-android.pdf
10. Marco A.F. Pimentel, David A. Clifton, Lei Clifton, Lionel Tarassenko (2014): A review of novelty detection. Paper, Institute of Biomedical Engineering, Department of Engineering Science,

University of Oxford UK, DOI: 10.1016/j.sigpro.2013.12.026, available at https://www.researchgate.net/publication/260030075_A_review_of_novelty_detection

11. Chih-Chung Chang and Chih-Jen Lin (2014): LIBSVM: A Library for Support Vector Machines. Paper, Department of Computer Science, National Taiwan University, Taipei, Taiwan. DOI: 10.1145/1961189.1961199, available at http://www.csie.ntu.edu.tw/%7Ecjlin/papers/libsvm.pdf
12. Frédéric Ratle, Mikhail Kanevski, Anne-Laure Terrettaz-Zufferey, Pierre, Esseiva and Olivier Ribaux (2007): A Comparison of One-Class Classifiers for Novelty Detection in Forensic Case Data. Paper, Institute of Geomatics and Risk Analysis, Faculty of Earth and Environmental Sciences, University of Lausanne, Switzerland and School of Criminal Sciences, Faculty of Law, University of Lausanne, Switzerland. DOI: 10.1007/978-3-540-77226-2_8, available at https://www.researchgate.net/publication/221252985_A_Comparison_of_One-Class_Classifiers_for_Novelty_Detection_in_Forensic_Case_Data
13. Chao Shen, Yong Zhang, Zhongmin Cai, Tianwen Yu, Xiaohong Guan (2015): Touch-Interaction Behavior for Continuous User

Authentication on Smartphones. Article, MOE KLINNS Lab, Xi'an Jiaotong University, Xi'an, China and Center for Intelligent and Networked Systems and TNLIST Lab, Tsinghua University, Beijing, China. DOI: 10.1109/ICB.2015.7139046

14. Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin (2016): A Practical Guide to Support Vector Classification. Paper, Department of Computer Science, National Taiwan University, Taipei 106, Taiwan, available at http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf