Decidir cuándo autenticar en dispositivos móviles a partir de modelos de machine learning[1]

En los dispositivos móviles como tablets o teléfonos celulares se tiene la opción de implementar o no un sistemas de autenticación para asegurar la identidad del usuario al utilizar el dispositivo. Este mecanismo suele ser un PIN o un patrón. En estos dispositivos se suelen guardar contraseñas de correo electrónico, redes sociales y bancos al igual que información personal como fotografías entre otros. Ciertamente es importante proteger estos datos y quién tiene acceso a ellos, sin embargo bajo el mecanismo actual de autenticación muchos usuarios prefieren no tener un mecanismo de seguridad a tener que estarse autenticando todo el tiempo. Un estudio en el 2011 determino que al menos un 30% de los usuarios de teléfonos celulares no bloquean su teléfono con un PIN aunque utilizan aplicaciones bancarias[2].

Para solventar este problema se han estado intentando implementar mecanismos de autenticación automáticos como reconocimiento facial o de voz, sin embargo estos suelen tener un alto número de falsos positivos así como un uso alto de energía para generar la autenticación debido a la captura de datos y el procesamiento que deben realizar. Recientemente Apple ha introducido la autenticación mediante huella digital en los nuevos modelos de iphone.[3] Este tipo de autenticaciones tiene además el problema de la privacidad ya que con un simple cambio en la política de privacidad las compañías pueden generar una base de datos privada de huellas digitales de los usuarios.

Por estas razones se quiere presentar un nuevo modelo que intenta atacar el problema de la autenticación por otro flanco. Se estudia el problema de cuando autenticar y para que aplicaciones en vez de la autenticación en sí misma para poder reducir el número de veces que se tiene que introducir una contraseña y por lo tanto que mas usuarios lo utilicen sin comprometer su seguridad. Para esto se desarrollo un sistema que partir de datos tomados por el teléfono y que utilicen poca batería se genere un modelo con machine learning para predecir si se ha comprometido o no la identidad del usuario y por lo tanto si debe autenticarse.

---

1 Este articulo está basado en el artículo con esta referencia
Riva, O. et al, "*Progressive authentication: deciding when to authenticate on mobile phones*", 21st USENIX Security Symposium 2011, pag 301-316.

2 "*24% of mobile users bank from a phone. Yet most don't have security measures in place.*" http://htmltest.bullguard.com/news/latest-press-releases/press-release-archive/2011-06-21.aspx (03/11/13)

3 "*iPhone 5s: About Touch ID security*", http://support.apple.com/kb/HT5949 (03/11/13)

Para poder formar este modelo se asume que el dispositivo móvil tiene únicamente un usuario y que el equipo cuenta con varios sensores de bajo costo para medir la presencia e identidad del usuario mediante sensores ambientales. El objetivo principal del mecanismo será proteger la aplicaciones más importantes de uso no autorizado. También se requerirá que exista transferencia de permisos, por ejemplo cuando el dueño del teléfono se autentica y luego le da el teléfono a otra persona mientras él está a su lado.

El entrenamiento de algunos de los modelos de machine learning puede ser muy intenso computacionalmente por lo que se desea en la medida de los posible que casi todo el entrenamiento sea incorporado al teléfono desde la fabrica y ajustes menores de personalización al llegar al usuario, por ejemplo detección de voz o facial. De forma tal que el modelo deberá depender principalmente de variables no dependientes del usuario y se harán calibraciones menores una vez que el dispositivo este con su usuario final.

El mecanismo se basara en señales biométricas como reconocimiento de voz y facial, en señales de comportamiento como que el teléfono sea utilizado a una hora o en un lugar extraño, en señales de posesión como estar cerca de otro dispositivo del mismo usuario como una laptop o PC, o que el usuario tenga siempre el dispositivo en su mano y finalmente en autenticaciones usuales como PIN o contraseña. A partir de todas estas señales se generara un puntaje que decidirá si se requiere autenticar o no al usuario. Se consideran señales de bajo nivel los sensores ambientales mientras que de alto nivel son el reconocimiento facial y de voz.

Finalmente existirán tres niveles de autenticación para las aplicaciones: Publico, Privado y Confidencial. El primero no requiere un puntaje muy bajo de confianza, por ejemplo para utilizar la cámara, el segundo requiere un nivel más alto de seguridad como por ejemplo para enviar mensajes de texto o entrar al correo electrónico mientras que el ultimo será usualmente reservado para aplicaciones bancarias y requiere un puntaje muy alto y por lo tanto casi siempre requerirá un PIN o contraseña.

Además de manera general se crean tres estados posibles para la ubicación física del dispositivo: En la mano del usuario, en el bolsillo o maletín del usuario y finalmente en una mesa o superficie. Dependiendo del estado del dispositivo se calibrara el modelo usando estas mediciones de bajo nivel para decidir si se debe autenticar o no el usuario. Cada uno de estos estados se detectará mediante los sensores, por ejemplo el sensor de humedad junto con el de luz activado determinara que el dispositivo está en la mano del usuario, mientras que una falta total de luz indicara que está en el bolsillo o maletín.

En este experimento se utilizo un modelo de maquinas de soporte vectorial. Este es un modelo de clasificación de machine learning basado en un vector de soporte.

Esto quiere decir que a partir de los valores de los atributos en un momento especifico se clasifica en dos categorías; es necesario autenticarse o no es necesario autenticarse.

En un modelo lineal separable de maquinas de soporte vectorial (SVM) se quiere trazar una línea o margen de decisión que separe adecuadamente los puntos de un hiperespacio en las diferentes categorías. Esto se puede observar en la figura 1. Además de todas las líneas posibles que pueden separar se busca la que maximice el margen de separación "d" de la línea a los puntos más cercanos de cada categoría. Esto se puede observar en la figura 2. Los puntos en los bordes a una distancia d de la línea constituyen los vectores de soporte.

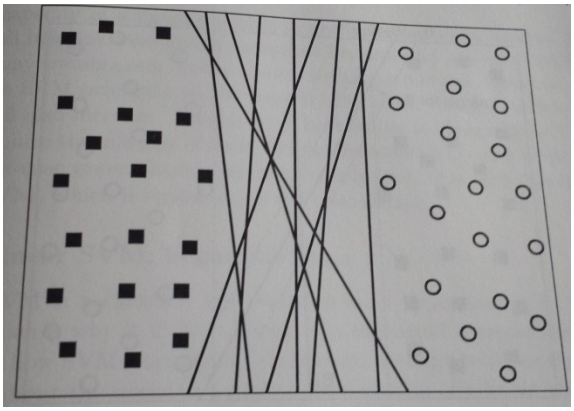

Figura 1. Posibles líneas de división entre puntos.[4]

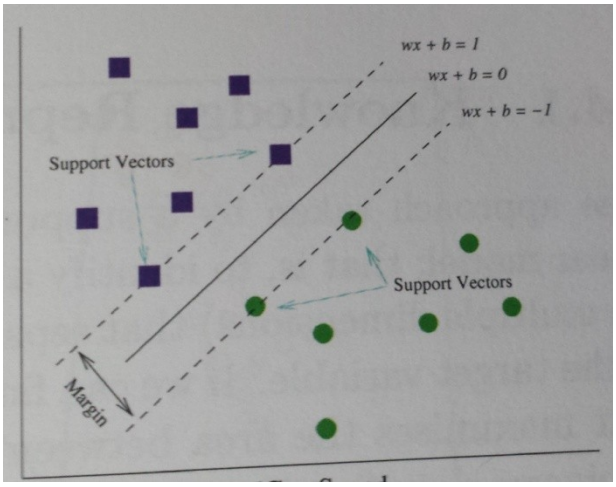

Figura 2. Vectores de soporte, Margen y líneas paralelas a la de decisión.[5]

4 Kumar, V., Steinbach, M., Tan, P., "*Introduction to Data Mining*", Addison Wesley, 1era ed., Boston, 2006, pag. 257

5 Williams, G., "*Data mining with Rattle and R*", Springer, 1era ed., New York, 2011, pag. 293.

Para el caso en que se deba clasificar únicamente en dos categorías como en nuestro caso. Si se tiene N arreglos de datos para entrenar el modelo y cada uno de estos se puede denotar por ($\acute{x}_i, y_i$) donde $\acute{x}_i$ es el vector de atributos para el ejemplo i-esimo y y_i es el resultado de la clasificación (requiere o no requiere autenticación) entonces la línea de decisión puede denotarse como:

$$\acute{w} \cdot \acute{x} + b = 0$$

donde w es el vector de pesos y b una constante, ambos parámetros del modelo.

Para cualquier cuadrado arriba de la línea de decisión, definiendo cuadrado =+1, circulo=-1 se puede mostrar que

$$\acute{w} \cdot \acute{x}_{cu} + b > 0$$

análogamente para los círculos

$$\acute{w} \cdot \acute{x}_{ci} + b < 0$$

además si $\acute{x}_a$ y $\acute{x}_b$ son dos vectores de soporte (están a una distancia d de la línea de separación) para el cuadrado y para el circulo respectivamente y se reescala $\acute{w}$ y b para definir +1 como cuadrado y -1 como circulo

$$\acute{w} \cdot \acute{x}_a + b = 1$$

$$\acute{w} \cdot \acute{x}_b + b = -1$$

$$\acute{w} \cdot \left(\acute{x}_a - \acute{x}_b \right) = 2$$

$$\| \acute{w} \| \cdot d = 2$$

$$d = \frac{2}{\| \acute{w} \|}$$

Entonces se puede definir el problema de SVM como encontrar la línea que minimiza $\acute{w}$ en

$$\min_{\acute{w}} \frac{\|\acute{w}\|}{2}$$

Sujeto a

$$y_i\left(\acute{w}\cdot\acute{x}_i+b\right)\geq 1$$

Este problema es convexo y por lo tanto se puede resolver mediante multiplicadores de Lagrange o mediante programación cuadrática.

En el caso que los datos no estén perfectamente separados en dos grupos es fácil modificar las ecuaciones para que se ajusten a esto sumando un factor de error a las ecuaciones para agrandar los márgenes y permitir flexibilidad lo además previene overfitting. Esto es que el modelo se aprende los datos en vez de aprender a distinguir entre ellos, por lo que tiene un excelente poder predictivo para los datos con los que se entrenó pero es muy malo para predecir cualquier dato nuevo.

Para el caso de datos que no puedan ser separados por una línea recta (caso no lineal) lo que se hace es cambiar el espacio de coordenadas original de los atributos en otro espacio en el cual los puntos si puedan ser separados por una línea como se observa en la figura 3.[6]

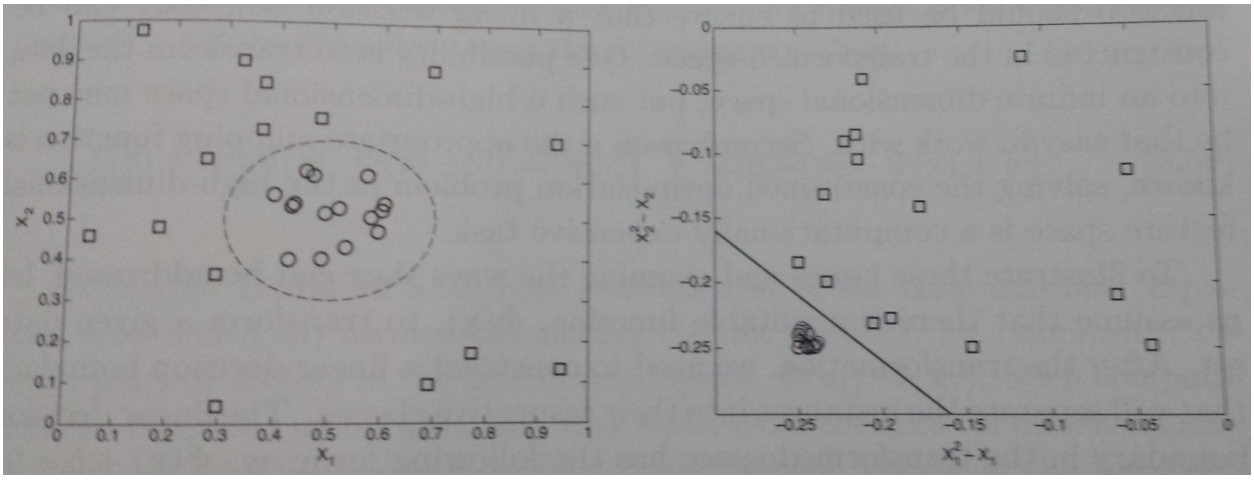

Figura 3. Transformación de espacio para poder hacer una separación lineal.

Otro modelo de machine learning es el árbol de decisión. Este sistema consiste en un árbol en el cual para bajar un nivel debe hacerse una pregunta cuyo objetivo es separar en grupos que son cada uno de los hijos hasta reducir el grupo lo suficiente como para poder clasificar a un objeto dentro de un grupo. Este procedimiento de bajar niveles se repite hasta que en las hojas se obtiene una respuesta que es la

6 Kumar, V., Steinbach, M., Tan, P., "*Introduction to data mining*", Addison Wesley, 1ed, 2006, pag. 271

clasificación del objeto. Cada hoja tiene un único nodo padre y los nodos internos tienen un único padre y dos o más hijos.

En la figura 4 se puede observar un ejemplo de un árbol de decisión para determinar si el un animal es mamífero o no.

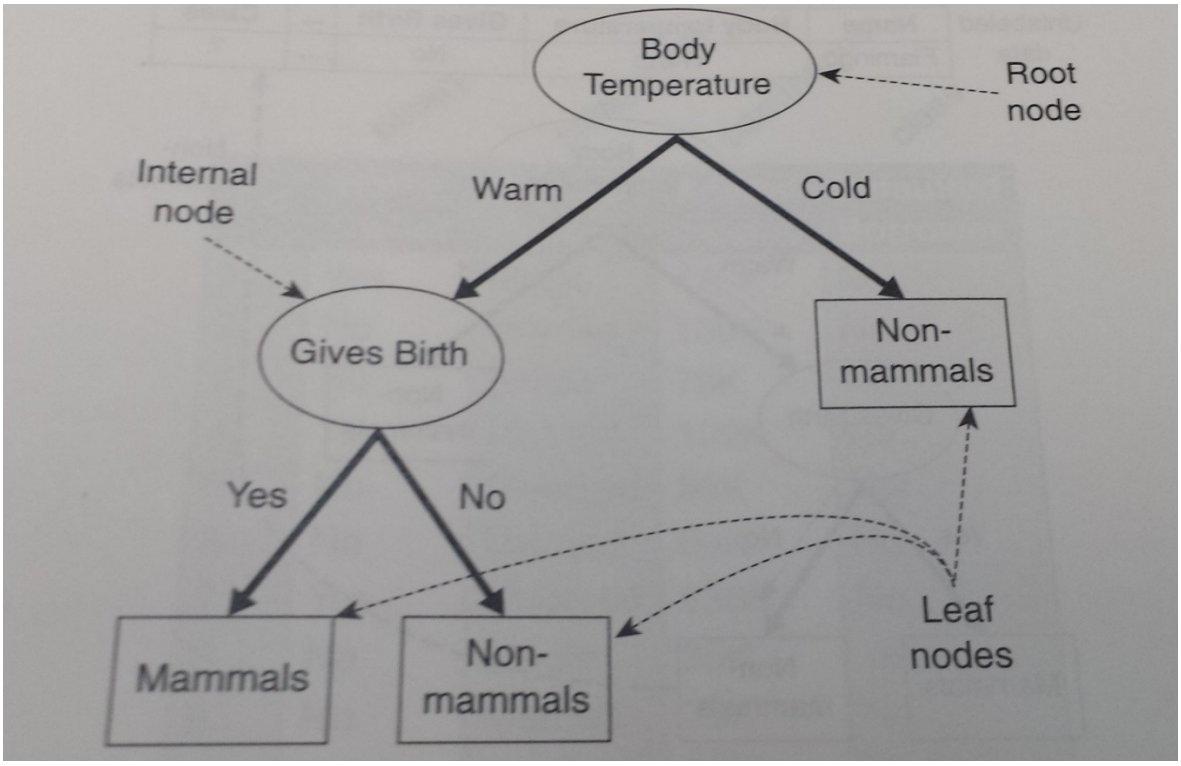

Figura 4. Árbol de clasificación para mamíferos.[7]

Para generar un árbol de decisión lo usual es utilizar el algoritmo de Hunt[8] que consiste en

Paso 1: Si todos los objetos en el conjunto de ejemplos pertenecen a la misma clase y_t entonces t es una hoja

Paso 2: Si en el conjunto de objetos hay algún objeto que pertenece a mas de una clase, se selecciona una condición de separación utilizando los atributos para particionar el conjunto en subconjuntos y por cada subconjunto se crea un hijo. Se repite recursivamente el algoritmo a cada nodo.

Ahora las decisiones que diferencian cada árbol son cómo se deben dividir los conjuntos en subconjuntos y cuándo debe detenerse el algoritmo. En el primer caso

7 Kumar, V., Steinbach, M., Tan, P., "*Introduction to data mining*", Addison Wesley, 1ed, 2006, pag. 151

8 Ibid

hay varios métodos de división, sin embargo el más popular es elegir la división que maximice la diferencia de impureza (grado de mezcla) entre el padre y los hijos. Esto se puede hacer utilizando como criterio de impureza la entropía ya que la diferencia entre entropías se conoce como ganancia de información. Para la segunda pregunta si el algoritmo sigue indefinidamente ocurre overfitting, con lo cual el algoritmo de clasificación deja de ser útil. Para evitar esto se realizan procedimientos de detiene la creación de hijos cuando el nivel de impureza llega un límite definido y también se poda el árbol una vez que está construido para reducir su tamaño.

Para el caso de autenticación al usar un dispositivo móvil se utilizo el siguiente vector de atributos de alto nivel para entrenar los modelos:

Tabla 1 Atributos de alto nivel elegidos para entrenar los modelos de clasificación[9].

Category	Features	Description
Cont.	Placement, PlacementDuration, PlacementConf	Current placement of the phone, how long it has lasted, and associated confidence
Cont.	LastPlacement, LastPlacementDuration	Last placement of the phone, and how long it lasted
Cont.	TimeSinceTable, TimeSinceHands, TimeSincePocket	Time elapsed since the last time the phone was on the table, in the user's hands, or pocket
Cont./Secrets	TimeSincePIN, TimeSinceTouch	Time since last login event and time since the phone's screen was last touched
Biom.	Speaker, SpeakerConf	Whether a human voice was identified (owner, other, no-voice) and associated confidence
Biom.	TimeSinceOwnerVoice, TimeSinceNonOwnerVoice	Time since (any) voice was identified
Biom.	TimeSinceSound	Time since any sound (either voice or noise) was detected
Poss./Biom.	ProxAuthDev, ProxAuthDevConf	Proximity of phone to a device where the user is logged in and active, and confidence
Poss./Biom.	TimeSinceProx	Time elapsed since the proximity status last changed

A partir de este modelo y con una base de datos generada por 20 usuarios distintos se logro entrenar un modelo de SVM en el cual las autenticaciones falsas son únicamente un 8% de todas las autenticaciones. Los otros modelos no dieron resultados tan buenos, como se verá a posteriormente.

Al incluir el procesamiento de bajo nivel como por ejemplo el estado del dispositivo (en una superficie, bolsillo o en la mano) y reconocimiento facial y de voz al modelo de SVM se observó que para determinar el estado del dispositivo los atributos que más contribuyen son el acelerómetro y un detector de temperatura y humedad y se determina adecuadamente el estado un 94% de las veces. Por otro lado se determinó que el reconocimiento facial no aporta mucho al modelo para determinar si debe autenticarse o no de nuevo el usuario mientras que el reconocimiento de voz aumenta considerablemente la confianza del dispositivo en que el usuario está cerca. Esto es una buena noticia ya que por ejemplo el nuevo sistema operativo de android Kitkat® implementara un sistema en el cual el dispositivo siempre está

9 Riva, O. et al, "*Progressive authentication: deciding when to authenticate on mobile phones*", 21st USENIX Security Symposium 2011, Pag 319.

escuchando y listo para hacer búsquedas en Google®[10], por lo tanto el sistema operativo estará optimizado para hacer este tipo de procesamiento sin consumir tanta batería. Los resultados para las variables se pueden observar en la importancia relativa de las variables para el modelo de SVM en la tabla 2

Tabla 2. Importancia relativa de cada atributo para el modelo de SVM.[11]

Feature rank	Feature name	Gain ratio
1	ProxAuthDev	0.16105
2	LastPlacementDuration	0.09785
3	TimeSincePIN	0.04879
4	ProxAuthDevConf	0.04584
5	TimeSinceOwnerVoice	0.04554
6	TimeSinceProx	0.03919
7	TimeSinceTouch	0.02802
8	TimeSinceSound	0.02618
9	LastPlacement	0.02529
10	TimeSinceTable	0.02264
11	Placement	0.01849
12	TimeSinceHands	0.0174
13	TimeSinceNonOwnerVoice	0.01505
14	TimeSincePocket	0.01456
15	Speaker	0.00983
16	SpeakerConf	0.00907
17	PlacementDuration	0.00884
18	PlacementConf	0.00000

Para evaluar la precisión del modelo de SVM comparado con otros modelos de machine learning se utilizó la precisión y recall. La precisión es la fracción de predicciones correctas de todos los ejemplos con el mismo resultado que la predicción (debe autenticarse o no) mientras que el recall es la fracción de predicciones correctas de todos los ejemplos en los que se determino adecuadamente el nivel de seguridad que se requería (público, privado, confidencial). Es decir si se tiene un alto recall, se determinará adecuadamente el nivel en el cual tiene que estar el puntaje de confianza para autorizar el uso, por lo

10 Gibbs, S., "Google Android 4.4 'Kitkat': seven things you need to know", http://www.theguardian.com/technology/2013/nov/01/google-android-44-kitkat-update (04/11/13)

11 Riva, O. et al, "*Progressive authentication: deciding when to authenticate on mobile phones*", 21st USENIX Security Symposium 2011, pag 312.

que van a haber muy pocas falsas autenticaciones a niveles altos de seguridad (se dio más permisos al usuario de los que se debía). Una precisión alta significa que el usuario rara vez tuvo que autenticarse en determinado nivel de seguridad ya que el sistema lo reconoció como el dueño del dispositivo mientras que una baja precisión significa que le pidió demasiadas veces autenticarse.

Se pueden observar los resultados de precisión y recall para varios modelos en la figura 5. Aquí se observa que el que tuvo mejores resultados en cuanto a garantizar la seguridad del usuario y mantener un equilibrio de nivel de autenticaciones fue el modelo de SVM. Este modelo obtuvo una precisión de 92.5% y un recall de 81%.

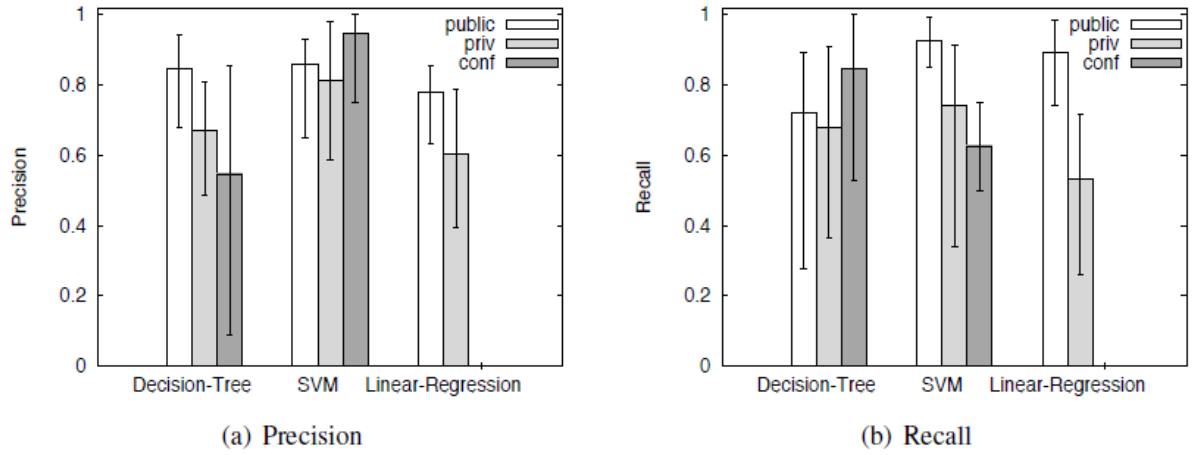

Figura 5. Precisión y recall para distintos modelos entrenados con 8 usuarios y validados con el noveno

En conclusión, al implementar este modelo de SVM para reconocer si el usuario debe o no autenticarse de nuevo se logró reducir el número de autenticaciones en un 42% y con solo 8% de accesos se tuvo más permisos de los que debía, sin embargo todos estos fueron para el dueño del equipo. Se hicieron pruebas de personas que intentaran tener acceso no autorizado al dispositivo y ninguno de ellos tuvo éxito por lo que el sistema es muy seguro. Además tomando en cuenta que la mayoría de las autenticaciones son para niveles bajos de seguridad, estos accesos no autorizados no representan un peligro real y este sistema es una excelente alternativa para usuarios que no tienen ningún mecanismo de seguridad o tienen uno de bajo nivel.

Bibliografía

Riva, O. et al, "*Progressive authentication: deciding when to authenticate on mobile phones*", 21st USENIX Security Symposium 2011, pag 301-316.

"*24% of mobile users bank from a phone. Yet most don't have security measures in place.*"
http://htmltest.bullguard.com/news/latest-press-releases/press-release-archive/2011-06-21.aspx (03/11/13)

"*iPhone 5s: About Touch ID security*", http://support.apple.com/kb/HT5949 (03/11/13)

Kumar, V., Steinbach, M., Tan, P., "*Introduction to Data Mining*", Addison Wesley, 1era ed., Boston, 2006, pags. 151, 257, 271

Williams, G., "*Data mining with Rattle and R*", Springer, 1era ed., New York, 2011, pag. 293.

Gibbs, S., "*Google Android 4.4 'Kitkat': seven things you need to know*",
http://www.theguardian.com/technology/2013/nov/01/google-android-44-kitkat-update (04/11/13)