Aprendizaje de Máquina para Autenticación Biométrica en Dinámica de Pulsaciones al Digitar una Clave

Sidney Paola Aguirre Castro
Departamento de Ingeniería de Sistemas
Universidad de Antioquia, Colombia
sidney.aguirre@udea.edu.co

John Alexander Galeano Ospina
Departamento de Ingeniería de Sistema
Universidad de Antioquia, Colombia
johna.galeano@udea.edu.co

***Abstract*—Mobile devices are an important personal tool for many different purposes, where sensitive information is stored, therefore, increasing authentication security is transcendent. We develop classifiers for biometric authentication through keystroke dynamics in order to detect fraud. The problem is approached using machine learning techniques, implementing Naive Bayes, K-Nearest Neighbors, Artificial Neural Network, Random Forest and Support Vector Machine models. In order to know the model that better classifies the genuine user from impostor, after simulating in each model, we compare results and select the best three models, in terms of accuracy and performance.**

Index Terms*—*Aprendizaje de máquina, dinámica de pulsaciones, biometría.

I. Descripción del Problema

El aumento en el uso de dispositivos móviles en conjunto con el tipo de información que almacenan, que en la mayoría de los casos corresponde a información personal y sensible; como técnica de protección de la información, el método más usado es la contraseña o el PIN; sin embargo, en muchos casos es posible descifrar dicha contraseña sin mayores complicaciones, esto genera la necesidad de fortalecer la seguridad en la autenticación con PIN o contraseña para evitar el fraude. En este estudio, se pretende evaluar, mediante información biométrica de usuarios, la dinámica de pulsaciones al digitar una clave en común '.tie5Roanl' en teclado digital, usando técnicas de aprendizaje de máquina.

Para abordar el problema, se usa un modelo de clasificación biclase para verificar si quien se autentica en un dispositivo móvil es o no el usuario real.

II. Variables del Problema

Nombre	Descripción	Número de características	Tipo de Codificación
Key hold time (H)	Tiempo que la persona mantiene la tecla presionada	14	cuantitativa
Down-down time (DD)	Espacio de tiempo entre teclas presionadas de manera consecutiva	13	
Up-down time (UD)	Tiempo entre la liberación de una tecla y la presión de otra tecla	13	
Key hold pressure (P)	Presión ejercida al presionar una tecla	14	
Finger area (FA)	Área ocupada por el dedo al presionar una tecla	14	
Average hold time (AH)	Promedio de los tiempos de mantener las teclas presionadas	1	
Average finger area (AFA)	Promedio del área ocupada por el dedo al presionar las teclas	1	
Average pressure (AP)	Promedio de la presión ejercida al presionar las teclas	1	
Total	-	71	-

Fig. 1: Variables del problema

Nota: El Número de características descrito en la tabla, corresponde a la cantidad de evaluaciones realizadas para la característica, es decir que la característica se tiene en cuenta

para cada una de las teclas involucradas en la escritura de la contraseña.

III. ESTADO DEL ARTE

A. Keystroke dynamics on Android platform

El artículo [1] presenta mediciones de identificación de usuarios por medio de la herramienta WEKA versión 3.6.11, una herramienta de software de machine learning. Las diferencias significativas en los resultados de este artículo se presentaron con prueba de emparejado t-test y con un nivel de significancia de 0.05. También se mejoraron algunos parámetros con los métodos de WEKA's search.

Los algoritmos utilizados son: Naive Bayes, este clasificador asume la independencia de las características de una instancia.

Vecinos cercanos (K-vecinos o IBk en Weka) algoritmo de clasificación donde una nueva etiqueta de la instancia es decidida por los vecinos k closet(Utilizando k=1, arrojó el mejor resultado).

Árboles de decisión, en este apartado utilizaron Weka's J48 del algoritmo C4.5 (factor de confianza de 0.2, mínimo de 4 instancias por hoja). Random forest classifier con 100 árboles.

Máquinas de soporte vectorial con implementación de LibSVM mediante WEKA con un kernel de base radial. Los parámetros C y ɣ optimizados por un algoritmo de búsqueda de cuadrícula distintivamente para los dos conjuntos de datos(C=7.46 y ɣ=0.25 para las 71 características) y normalización de las características en el rango de [0-1].

MLP con algoritmo de Backpropagation y el número de capas ocultas por medio de la configuración por defecto de WEKA [número de atributos + número de clases] / 2 .

Classifier	Accuracy using time based features H+DD+UD+AH (41 features)	Accuracy using time and touchscreen based features H+DD+UD+P+FA+AH+AP+AFA (71 features)
Naïve Bayes	50.15% (2.86)	78.93% (2.63)
Bayesian Networks	75.95% (2.65)	91.94% (1.73)
C4.5(J48)	54.79% (3.84)	69.02% (3.32)
k-NN (IBk)	41.07% (2.83)	72.98% (2.25)
SVM(LibSVM)	61.71% (3.22)	88.33% (1.87)
Random forest	82.53% (2.53)	93.04% (1.65)
MLP	53.01% (3.39)	86.26% (2.19)

Fig. 2: Resultados del conjunto de validación

B. Supervised learning methods for biometric authentication on mobile devices [3].

En este artículo, se aborda el problema de autenticación por dinámica de pulsaciones, mediante un comparativo entre modelos logísticos y redes neuronales profundas. Se incluye una variación propuesta por los autores denominada red neuronal profunda triangular con tres capas ocultas, cuya configuración consiste en que una capa oculta contiene la mitad de neuronas que la capa oculta anterior (100-50-25).

Los autores aplican técnicas de sobremuestreo (Synthetic Minority Oversampling Technique - SMOTE) y de submuestreo aleatorio (Random Undersampling - RUS).

La tabla siguiente muestra los resultados obtenidos en el estudio:

Model	Loss	Accuracy	Recall	Precision	False Negative Rate
Logistic	6.661	58.12%	23.88%	78.18%	76.12%
DNN-1	0.690	69.10%	49.05%	**83.31%**	50.95%
DNN-2	7.905	50.41%	**99.78%**	49.76%	**0.22%**
DNN-3	0.536	73.73%	70.37%	76.04%	29.63%
DNN-4	0.695	49.73%	0.21%	59.89%	99.79%
DNN-5	0.531	73.45%	72.52%	74.30%	27.48%
DNN-6	0.409	81.70%	91.18%	77.40%	8.82%
DNN-7	0.484	76.88%	97.16%	69.15%	2.84%
DNN-8	0.527	72.93%	80.23%	70.79%	19.77%
DNN-9	0.425	80.29%	81.69%	80.19%	18.31%
DNN-10	0.450	79.78%	77.46%	81.37%	22.54%
Triangle	0.380	**84.28%**	89.76%	81.14%	10.24%

Table 2. Model results on validation set. False Negative Rate is calculated as 100% - Recall.

Fig. 3: Resultados del conjunto de validación

El estudio concluye con que la técnica más acertada es la red neuronal triangular, sin embargo se destaca que la sensibilidad es muy alta cuando se usa una red neuronal profunda con dos capas ocultas.

C. An evaluation of one-class and two-class classification algorithms for keystroke dynamics authentication on mobile devices [4].

En esta conferencia, la base contaba con 42 usuarios con 51 registros cada uno. el problema se aborda de dos formas y se comparan resultados: como una sola clase y como biclase, usando validación cruzada con 10 folds. Para la clasificación con dos clases, se entrenaron los modelos usando 45-46 muestras positivas (usuario genuino) del 90% del total de muestras, y 82 muestras negativas (impostor). Los conjuntos de validación estuvieron compuestos por 5-6 muestras positivas (10% del total de muestras positivas) y 82 muestras negativas. Los modelos usados fueron: Random Forest con 100 árboles, Naive Bayes con parámetros por defecto en Weka, y K-vecinos más cercanos con k=3. Los resultados obtenidos fueron: tasa de muestras clasificadas como positivas, y tasa de muestras clasificadas como negativas. Las medidas fueron Tasa de Error en los Positivos, Tasa de Error en los Negativos, y Tasa de Igual Error (Equal Error Rate - EER); siendo ésta última la medida más importante. Se tomaron 3 conjuntos del dataset: 71, 17 y 3 características.

El mejor EER fue obtenido con Random Forest (3.1%) con un total de 71 características. Seguido por Bayes (4.3% EER). y KNN demostró ser el más débil (8.3% EER). Para un total de características de 17, todos los modelos produjeron EER entre 6.6-10.9%. El mejor desempeño se presentó con 3 características (7.1-9.8%), resaltando que son independientes de la contraseña y que reflejan el comportamiento propio del individuo (Average hold time, Average pressure and Average finger area).

IV. EXPERIMENTOS

Usamos la base de datos MEU-Mobile KSD (Keystroke Dynamics) de UCI Machine Learning Repository[2]. Esta base de datos cuenta con información de 56 sujetos con 51 registros cada uno, para un total de 2856 registros. Se tienen en cuenta para el análisis 71 variables, incluyendo Hold time, Down-down time, Up-down time, Pressure, finger area, averages of hold, finger area and pressure. (ver sección 2, Variables del Problema). Es importante especificar que la base de datos no contiene datos faltantes, y puesto que cada sujeto es entendido como una clase, no se presenta desbalance en los datos.

El problema se abordó como un modelo de clasificación biclase, usuario auténtico e impostor. Además, de los 56 sujetos registrados en la base de datos, escogimos 10 sujetos para los experimentos, utilizando una validación cruzada con 5 folds. esta decisión fue tomada debido a que los cinco modelos se debían entrenar y validar para cada uno de los 56 usuarios, y por fines académicos, para simplificar el problema, sólo se trabajó con 10 usuarios en los cinco modelos usados.

La clase impostor se genera usando 51 registros de otros sujetos en la misma base de datos, escogidos de manera aleatoria. Así se obtiene el conjunto de datos compuestos de 51 registros del usuario genuino, y 51 registros tomados aleatoriamente de los 9 usuarios restantes para la clase impostor; un total de 102 registros para cada set de datos. Esto se hace ya que cada usuario es un modelo independiente de los demás.

A. *Naive Bayes*

Para este modelo, se siguió la recomendación del artículo[1] con parámetros por defecto. Por lo tanto tomamos el método de sklearn y dejamos los parámetros por defecto. Se resalta que en el artículo[1] se usa Weka, y en nuestro caso, se usa sklearn.

B. *K vecinos más cercanos - KNN*

Para el modelo KNN, se usaron los parámetros sugeridos en el artículo [1], usando k = 1, k = 5 y k= 10.

C. *Redes neuronales artificiales - ANN*

Se establecieron 4 modelos de redes neuronales artificiales las cuales son: primero una MLP con parámetros por defecto de la librería sklearn con 200 épocas, 100 neuronas, 1 capa oculta y función de activación ReLu, segundo modelo con 25 épocas, 1 neuronas , 1 capa oculta y tercer modelo con 25 épocas, 10 neuronas, 10 capas ocultas y función de activación ReLu, y el cuarto modelo se le asignó una forma triangular con 20 épocas, una capa con 100 neuronas, otra capa con 50 neuronas y la última capa de 25 neuronas con función de activación ReLu, según los parámetros sugeridos en el artículo[3].

Incluímos la MLP con parámetros por defecto de sklearn, para analizar el comportamiento con más épocas y neuronas.

D. *Random Forest*

Se establecieron 4 modelos de Random Forest, con variando la cantidad de árboles en 10, 20, 50 y 100 sugerido en el artículo [1][2]

E. *Máquinas de Soporte Vectorial - SVM*

Se configuraron dos modelos para este el primero con parámetros por defecto con C = 1, gamma cercano a 0 y kernel lineal, y el segundo modelo con parámetros C = 7.46, gamma = 0.25 y kernel RBF

C: Restricción de caja. Este parámetro, limita el número de vectores de soporte.

Modelo	Tiempo
SVM - Kernel lineal con parámetros C = 1.0 y gamma = auto (tiende a cero)	0.008397
Naive Bayes	0.009597
Random Forest con 10 árboles	0.040186
Random Forest con 20 arboles	0.071776
MLP Triangular	0.101967
Random Forest con 50 árboles	0.164747
MLP con parámetros épocas = 200, neuronas = 100 y 1 capa oculta	0.288106
Random Forest con 100 árboles	0.370885

Fig 4: Modelos con eficiencia 1.0 y su desempeño

Como se observa en la tabla, los mejores modelos son:

- SVM - Kernel lineal
- Naive Bayes
- Random Forest con 10 árboles

Las simulaciones fueron en total 14 modelos con 5 folds para cada uno de los 10 sujetos (700 simulaciones en total). Para seleccionar los mejores modelos para la solución del problema de autenticación con dinámica de pulsaciones, se filtró por la mejor medida de eficiencia (Total de muestras clasificadas correctamente, tanto positivas como negativas), donde observamos que varios modelos tuvieron una eficiencia del 100%, de entre los cuales luego decidimos filtrar por el mejor tiempo de eficiencia computacional, obteniendo los tres modelos nombrados anteriormente, como los de mejor desempeño y solución para el problema.

VII. Análisis de características

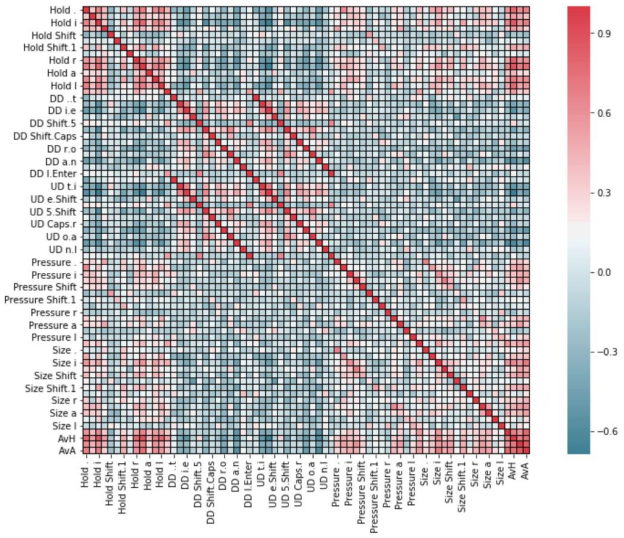

Fig 5: Matriz de correlación de las 71 variables

Características candidatas a ser eliminadas: DD i e,DD t,UD t i,UD e Shift,UD 5 Shift,UD caps r,UD o a,UD n l,DD r.o,DD a n, size l, size i,size Shift y size Shift 1

VIII. Selección de características

Como estrategia de búsqueda se utilizó Selección secuencia flotante descendente SFBS, y criterio de selección se usó Wrapper, ya que se acomoda muy bien a la esencia del problema en cuestión, puesto que provee mayor exactitud y debido a que están ajustados para reducir el error de validación, generaliza mejor comparado a un criterio tipo filtro. Como técnica de selección de características se recurrió a hacer análisis del índice de fisher, Este nos da como evidencia que el número de características que mejor discriminan las clases teniendo en cuenta 0.44 como valor base donde se observó una ganancia de información significativa. Las variables son:

Ganancia de información	Descripción de la característica
0.563	average hold time
0.543	average finger area
0.512	average pressure
0.482	fingerarea6
0.469	pressure9
0.466	holdtime12
0.458	pressure6
0.455	fingerarea2
0.452	pressure3
0.45	updown12
0.449	holdtime10
0.448	downdown3

Fig 6: Variables discriminativas - índice de Fisher

Como se puede observar, con el índice de Fisher se obtiene que las variables discriminativas son

- Average hold time (AH)
- Average finger area (AFA)
- Average pressure (AP)

Las cuales aportan información sobre el comportamiento propio del usuario, y son independientes de la contraseña en sí misma. Este resultado es similar al obtenido en el artículo de conferencia[4].

Los resultados al aplicar el criterio y estrategia de búsqueda descritos, son los siguientes en orden de eficiencia:

Modelo	Eficiencia	Precisión	Especificidad	Sensibilidad
SVM - Kernel lineal con parametros C =0 y gamma = 0	0.805+-0.083	0.983+-0.03	0.739+-0.110	0.739+-0.110
Random Forest con parametros numero de arboles = 10	0.510+-0.07	0.800+-0.400	NaN	0.409+-0.216
Naive Bayes	0.500 +-0.04	0.000+-0.000	NaN	0.000+-0.000

Fig 7: resultados al aplicar el criterio y estrategia de selección de características en la validación

IX. Extracción de características con PCA.

Para determinar el análisis mediante PCA primero se determina la varianza acumulada para considerar la mejor cantidad de componentes que describa nuestro conjunto de datos. Con el objetivo de mantener la mayor cantidad de información posible y un número pequeño de variables (entre 12 y 20) determinamos que una varianza acumulada igual o superior del 99.8% es adecuada para un conjunto de 71 características, dicho valor de varianza acumulada se logra con 15 componentes. Obteniendo los siguientes resultados con criterio de orden el mejor valor de Eficiencia.

Modelo	Eficiencia	Precisión	Especificidad	Sensibilidad
SVM - Kernel lineal con parametros C =0 y gamma = 0	0.980 +-0.02	0.981 +-0.03	0.977+-0.04	0.977+-0.04
Random Forest con parametros numero de arboles = 10	0.970+-0.02	1.000+-0.00	0.941+-0.05	0.941+-0.051
Naive Bayes	0.950+-0.04	0.983+-0.03	0.926+-0.926	0.926+-0.090

Fig 8: resultados al aplicar extracción de características en la validación.

X. Discusión

Una dificultad encontrada en el proceso, fue la cantidad de simulaciones y datos a analizar, en especial en el apartado de selección de características donde se implementó Wrapper lo cual elevó el costo computacional de los experimentos.

Inicialmente se toman como opciones a ser mejor modelo KNN con 1, 5 y 10, MLP, Random Forest con 10 y 20 árboles, y SVM con kernel lineal ya que presentaron en las simulaciones una eficiencia de 100%, sin embargo, dado que lo que se busca es una equilibrio entre desempeño y capacidad de predicción, se optó por considerar el tiempo de ejecución, por lo cual se hizo un filtro entre los mejores modelos.

Podemos observar que los modelos se sobre ajustaron debido a que los set de datos son pequeños (102 muestras) y presentan un problema de dimensionalidad con 72 características, por lo cual se abordó la problemática con una reducción agresiva de llevar ese espacios de características a uno mucho menor. También se debe resaltar que en un problema de fraude los modelos de cada usuario deben ser calibrados en los parámetros de cada modelo para garantizar un desempeño adecuado.

No usamos técnicas de sobremuestreo ni submuestreo como en el artículo [3], pues trabajamos con el conjunto de datos del usuario genuino y generamos aleatoriamente un conjunto de datos para la clase impostor tomando registros de otros usuarios.

La red neuronal triangular del artículo [3] ofreció un buen resultado en la eficiencia, con un 84.28. En nuestras simulaciones, esta configuración no apareció como una de las más significativas.

Naive Bayes en el artículo [1] presentó una eficiencia alta en sus resultados finales en comparativa a los experimentos realizados en este artículo, esto se presentarse ya que en el artículo [1] no realizaron selección y extracción de características.

References

[1] Antal, M. Szabó, L. Z, László, I. Keystroke dynamics on Android platform. 2014

[2] N. Al-Obaidi. MEU-Mobile KSD Data Set. UCI Machine Learning Repository. 2016

[3] et al. Supervised learning methods for biometric authentication on mobile devices. (S.F)

[4] Antal, M. Szabó, L. Z, An evaluation of one-class and two-class classification algorithms for keystroke dynamics authentication on mobile devices. 2015