

Técnicas de aprendizaje automático aplicadas a electroencefalogramas

Omar Santa-Cruz, Lorena del Mar Ramírez, Felipe Trujillo-Romero

Universidad Tecnológica de la Mixteca,
Huajuapán de León, Oaxaca,
Mexico

iccsantacruz@gmail.com, lorenadelmar.rmz@gmail.com,
ftrujillo@mixteco.utm.mx

Resumen. Los electroencefalogramas (EEG) almacenan la actividad cerebral en un instante de tiempo determinado. Actualmente los EEG son utilizados en disciplinas médicas, científicas y tecnológicas. Debido a esto, lograr interpretar y clasificar de forma precisa estas señales es un problema abierto en el área de machine learning. En esta investigación se trabaja con un conjunto de datos que representan señales de EEG, en las cuales se encuentran codificados los estados de los ojos (abiertos o cerrados). Este conjunto ha sido analizado por diversos autores, lo que se propone en esta investigación es tratar de mejorar el desempeño de los métodos ya explorados. Por lo anterior, se explora mediante una búsqueda greedy los valores de los parámetros que ofrezcan un mayor rendimiento al utilizar SVM y RNA. También se aplican técnicas de reducción mediante selección de prototipos y extracción de características al conjunto de señales. El objetivo es tratar de mejorar la estructura de los datos mediante una representación más compacta, para posteriormente comprobar si se logra mejorar el proceso de discriminación.

Palabras clave: EEG, SVM, redes neuronales artificiales, extracción de características, selección de prototipos, optimización, predicción del los estados de los ojos.

1. Introducción

Las nuevas tecnologías en el ámbito médico y su fusión con áreas de las ciencias de la computación han dado origen a la llamada informática médica. La implementación de software que colabore en el diagnóstico médico es un problema de investigación con gran demanda en esta área[1]. Bioinformática y química computacional son dos claros ejemplos de la relación de la computación con áreas en donde la cantidad de datos que se procesa durante una investigación rebasa la capacidad de proceso humano. Un ejemplo de diagnóstico médico asistido por computadora son los sistemas CADx (Computer-Aided Diagnosis) y CAD (Computer Aided Detection)[2,3], los cuales se utilizan para ayudar a los radiólogos en diagnóstico y detección de anomalías en imágenes de

mamografías. Existen otros tipos de sistemas automáticos de reconocimiento de patrones que auxilian en el diagnóstico diferencial de casos médicos, los cuales involucran electrocardiogramas, electroencefalogramas, radiografías, tomografías, entre otros [1,4,6]. El diagnóstico médico objetivo y más acertado depende de la experiencia que el especialista ha adquirido a través de los años al tratar cientos de casos. Sin embargo, debido al número de variables que se deben manejar (edad, sexo, el tiempo de evolución, síntomas específicos) en ciertos casos clínicos, la probabilidad de error del diagnóstico diferencial aumenta. La hipótesis médica puede verse afectada por el sesgo de confirmación debido a muchos factores como son: estado de ánimo, falta de experiencia, influencia médica externa, etc. La incursión del aprendizaje automático en el área médica puede ayudar a la consolidación de un sistema de diagnóstico más eficiente [1,5]. El uso de un software que apoye al especialista médico a tomar una decisión debe empezar a considerarse como una necesidad básica debido al manejo de datos en grandes cantidades y con relaciones por lo general muy complejas entre ellos. Un programa de diagnóstico médico puede igualar la capacidad de un experto humano, además, no sólo es capaz de reconocer un patrón clínico y dar un diferencial, sino también puede facilitar el trabajo del médico, además de reforzar sus conocimientos, y establecer un pronóstico de riesgo [1,6]. En este documento se explora un problema de predicción de los estados de los ojos (abiertos o cerrados) mediante información extraída por medio de electroencefalogramas (EEG). Los EEG son ampliamente utilizados en muchos problemas de medicina para diagnosticar diferentes enfermedades.

2. Electroencefalogramas

Un EEG es un conjunto de señales que proviene de la captura de actividad cerebral por medio de sensores específicamente diseñados para esto. Un electroencefalograma registra las diferencias de potencial eléctrico producidas en el cerebro cuando se realiza cierta actividad[8]. Este potencia eléctrico en forma de impulsos contiene información sobre la comunicación neuronal en el cerebro. En la forma de onda de la señal se encuentra almacenada la información sobre las intenciones del usuario en un momento determinado. Por medio del análisis y búsqueda de patrones con técnicas de machine learning es posible obtener un modelo que logre predecir ciertos estados de la mente[9]. Algunos ejemplos de aplicaciones en las que es útil la información de un electroencefalograma son: reconocimiento de emociones[9,33], realidad virtual[14,15,16], videojuegos[17], interfaces cerebro-computadora[10,11], en el ámbito militar[13,32] y en software de control de objetos para personas con algún tipo de discapacidad [11,12]. También es posible predecir enfermedades que causan degeneración de células cerebrales como lo hace el Alzheimer's [18].

En este contexto, reconocer estímulos con suficiente precisión es de vital importancia en los problemas antes mencionados. Detectar el estado de los ojos (abiertos o cerrados) puede ser de utilidad en sistemas de prevención de accidentes [19], identificación de trastornos de sueño infantil[20], detección de convul-

siones epilépticas[21], clasificación de desorden bipolar y detección de pacientes con desorden de déficit de atención [22], identificación de las características de estrés [23] y detección de parpadeo en el ojo humano[24]. El problema de clasificación de los estados de los ojos se ha atacado con diferentes métodos de aprendizaje automático. Por ejemplo, Fukuda et al.[25], utilizan una red neuronal artificial para la clasificación del estado del los ojos. Yeo et al.[19] se enfocan en la detección de somnolencia durante la conducción de vehículos por medio del estado de los ojos, para ello, utiliza una máquina de soporte vectorial. Polat y Gne [21] utilizan un árbol de decisión en la detección de crisis epiléptica . Sulaiman et al. [23] utilizan k -NN para la identificación de características de estrés . Además, Rslar y Suendermann [26] generan un corpus con cerca de 15,000 muestras y someten los datos a 42 clasificadores auxiliándose de la plataforma WEKA[29]. [27,28,30,31] utilizan el mismo corpus([26]) para aplicar otros enfoques de clasificación a los ya reportados.

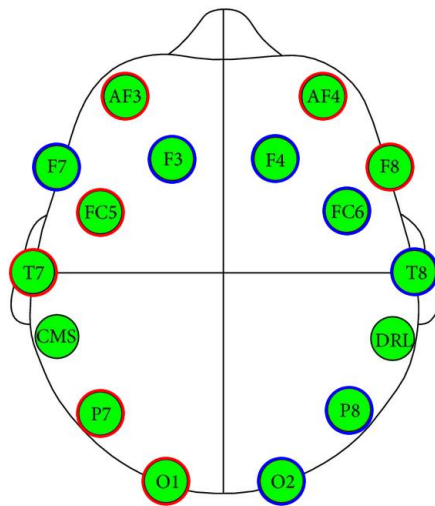


Fig. 1. Esquema general de la posición de los sensores. Los sensores marcados con color azul toman valores máximos, mientras que los sensores marcados en color rojo toman los valores mínimos cuando los ojos están abiertos [29].

Es este trabajo se analiza el mismo corpus, se utiliza SVM y una RNA con una búsqueda greedy que logre la obtención de los mejores parámetros del clasificador. Además se utiliza un algoritmo de selección de instancias y de extracción de características para reducir el número de muestras y el número de variables en el conjunto de datos respectivamente. Como prueba final, los conjuntos reducidos se someten a los clasificadores antes mencionados.

3. Algoritmos de aprendizaje automático

3.1. Prototype selection by clustering (PSC)

En un conjunto de entrenamiento T , los prototipos de borde, son relevantes para encontrar la frontera de las distintas clases, estos prototipos que se encuentran en la frontera de las clases son siempre críticos para el proceso de clasificación. Las instancias localizadas en el interior de estas regiones puede ser considerados como superfluas ya que su eliminación no conduce a ninguna pérdida en la capacidad de encontrar el discriminante de clase [34].

PSC utiliza C -means para dividir el conjunto de datos en C -clusters. Después de realizar el agrupamiento se analiza y se identifica cual de ellos es un cluster homogéneo y cual no lo es. Para los clusters no homogéneos se detectan los prototipos de borde de las diferentes clases y se elimina el resto. Para las regiones homogéneas PSC conservar sólo al prototipo que mejor represente a la región [36], esto es, la instancia más cercana a la media del grupo.

La forma en que PSC decide que prototipos son de borde, es subdividiendo los cluster no homogéneos de acuerdo a su clase de pertenencia. El subgrupo con mayor número de elementos es marcado como el subgrupo de clase mayoritaria [37]. Una vez que el subconjunto de clase mayoritaria ha sido encontrado, los prototipos de borde de los diferentes subgrupos de cada clusters serán los vecinos más cercanos(1-NN) a algún prototipo que pertenezca a la clase mayoritaria. De forma análoga, PSC encuentra los prototipos de borde del subgrupo de clase mayoritaria buscando a los prototipos que sean los vecinos más cercanos del algún prototipo de clase mayoritaria, pero que no pertenezca a dicha clase[38]. De forma análoga se encuentran los prototipos de borde de los subgrupos restantes.

3.2. Máquinas de soporte vectorial(SVM)

Si T representa un conjuntos de m datos de dimensión n en el que cada ejemplo x_i esta asociado a una etiqueta $y_i \in [-1, 1]$. El hiperplano que separa a ambas clases es un vector que satisface la siguiente ecuación :

$$w \cdot x - b = 0 \quad (1)$$

donde w representa la norma del vector al hiperplano y $\frac{b}{\|w\|}$ la distancia perpendicular al origen. SVM puede encontrar los valores óptimos que corresponda al hiperplano de margen máximo minimizando $\|w\|^2$ y garantizando que se cumpla la siguiente expresión: $y_i(x_i * w + b) - 1 \geq 0, \forall i$, donde y_i es la etiqueta asociada a la instancias x_i [42]. Para resolver este problema de optimización con restricción se puede utilizar los multiplicadores de Lagrange en donde la función objetivo L_p (lagrangiano primal) puede ser minimizada con respecto a w y b :

$$L_p(w, b, \alpha) = \frac{1}{1} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (x_i * w + b) - 1) \quad (2)$$

Esto quiere decir que se puede resolver el problema dual asociado al problema primal, que equivale a replantear la solución de la siguiente forma:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i * x_j \quad (3)$$

donde w , b desaparecen y los vectores del conjunto de entrenamiento que proporcionan un multiplicador $\alpha_i > 0$ son denominados vectores de soporte. El algoritmo SVM fue modificado en [46], introduciendo el concepto de margen suave, el cual permite que el margen no sea tan estricto, agregando una constante C al error de clasificación ϵ . SVM puede ser extendido a problema de separación no lineal aplicando el llamado truco kernel. Una particularidad de las funciones kernel es que el producto punto de las imágenes en $\Phi(X)$, el cual representa un mapeo no lineal, es equivalente a evaluar los datos del conjunto original en la función kernel, esto es:

$$\langle \Phi(x_i), \Phi(x_j) \rangle = k(x_i, x_j) \quad (4)$$

donde \langle, \rangle representa el producto punto.

3.3. Redes neuronales artificiales (RNA)

Una red neuronal recibe como entrada un conjuntos de $i = 1, \dots, n$ patrones en forma de vectores de dimensión p , cada vector de entrada es procesado a través de las neuronas de las I capas ocultas de acuerdo a las conexiones entre los nodos de estas. Cada nodo contiene un función de activación f , la cual mediante una suma ponderada de las entradas de los nodos y un valor de sesgo adicional obtiene el valor de salida del nodo[40].

Existen diferentes tipos de arquitectura, interconexiones y algoritmos de aprendizaje para la creación de una red neuronal. La estructura de una red neuronal de tipo feedforward recibe ese nombre debido a que las conexiones de sus neuronas fluyen unidireccionalmente desde la capa de entrada hasta la capa de salida sin que en el proceso haya ciclos de retroalimentación entre las neuronas. La regla más utilizada para que la red aprenda es el algoritmo backpropagation. El algoritmo de backpropagation utiliza la técnica del gradiente descendente para buscar el mínimo de la función de error en el espacio de pesos de los nodos. Así, mediante la obtención del gradiente con respecto a los pesos de la red se calcula la combinación de pesos que minimice dicha función[42].

Suponiendo que la integral de la función de activación es sólo la suma de las entradas del nodo, y debido a que la salida de los nodos en cada capa se pueden ver como una aplicación sucesiva (función composición) de la función de salida de los nodos de la capa anterior; es posible la obtención del gradiente de la función mediante cualquier función de transferencia diferenciable f .

La ecuación del error general de la red es:

$$E = \frac{1}{2} \sum_{i=1}^m \|o_i - t_i\|^2 \quad (5)$$

Conociendo el error E se puede calcular el gradiente con respecto de los pesos asociados a cada nodo y posteriormente reajustar los pesos y los bias de la red. El proceso de optimización termina cuando el error general de la red converge o bien cuando el error alcanza un valor mínimo deseado.

Gradiente descendente con momentum Cuando el mínimo de la función de error en el espacio de pesos se encuentra en una 'superficie' r -dimensional estrecha y la tasa γ no es la adecuada, esto provoca que la convergencia del error oscile, lo cual genera un retardo significativo de la convergencia y en el proceso de aprendizaje. Si utilizamos una constante de momentum β , que funcione como un filtro pasa bajas podemos evitar oscilaciones excesivas llegando a la convergencia de la red en un menor número de iteraciones. No existe una regla general para el ajuste de los dos parámetros de aprendizaje que acelere al algoritmo en la convergencia, normalmente este ajuste se realiza mediante ensayo y error o mediante una búsqueda greedy.

Gradiente descendente con tasa de aprendizaje variable Este algoritmo de aprendizaje trata de mantener la constante de aprendizaje γ lo más alta posible, haciendo que γ varíe de forma proporcional a la complejidad de la superficie de error. Al acoplarse a la forma de la superficie estabiliza el aprendizaje durante todo el proceso de entrenamiento. Existen diferentes técnicas para lograr que la tasa de aprendizaje varíe de acuerdo a rendimiento de la red para cada iteración. De forma general la lógica a seguir es la siguiente: Si el error disminuye, la dirección del gradiente es la correcta y se puede acelerar el proceso de convergencia incrementando el tamaño de γ . Por el contrario, si el error se incrementa el algoritmo actúa decrementando la tasa de aprendizaje [41,42]. El algoritmo Backpropagation con momentum y aprendizaje variable garantizan rápida convergencia para algunos problemas, sin embargo el valor adecuado de todos los parámetros debe ser asignado mediante ensayo y error, no garantizando la convergencia para problemas muy complejos.

3.4. Análisis de componentes principales (ACP)

Dado un conjunto de datos con p variable, ACP transforma dicho conjunto en uno nuevo con r variables, siendo $r \ll p$. Este nuevo conjunto debe cumplir ciertas restricciones: las nuevas variables, las cuales reciben el nombre de componentes principales, deben ser una combinación lineal de las variables de partida, además, cada nueva variable debe representar una parte de la variabilidad de los datos originales. Cuanto mayor sea la varianza de las componentes, mayor es la información que almacenan las nuevas variables. Siguiendo esta lógica, la varianza debe ir decreciendo según se vaya obteniendo cada nueva variable. Cabe mencionar que la suma total de estas varianzas es igual a la suma de las varianzas del conjunto original [44]. Otra condición que debe cumplir el nuevo conjunto es que las nuevas variables no estén correladas, ya que esta correlación impide apreciar de forma precisa el rol que juega cada variable en el problema bajo investigación [43].

4. El conjunto de datos

El conjunto de datos que se emplea en los experimentos es obtenido del repositorio de la Universidad de California (UCI) [39]. El conjunto de datos consta de 14980 ejemplos, que contiene 14 características descriptivas, donde cada característica representa la información que recolecta cada uno de los 14 sensores como muestra la figura 2. Cada medición dura 117 segundos, las señales que emiten las ondas cerebrales son retenidas por los sensores y mediante un proceso de transducción son mapeados a valores reales. El estado de los ojos asociado a cada secuencia se añadió de forma manual, después de analizar los fotogramas de vídeo que captó una cámara durante todo el proceso de obtención de la información. Cada ejemplo del conjunto de muestras tiene asociada la clase 1 cuando el ojo se encuentra abierto y la clase 0 de forma contraria. En [26] se analizan los rangos máximos y mínimos de las variables para los dos estados de los ojos. En este análisis se concluye que existe una diferencia significativa entre los dos estados (abiertos y cerrados) que permite realizar una discriminación mediante algoritmos de aprendizaje automático.

5. Método propuesto

Lo que se propone en esta investigación es tratar de mejorar la precisión de clasificación de algunas técnicas reportadas en el estado del arte, específicamente se trata de mejorar la precisión de SVM en [27] y del perceptrón multi-capas utilizado en [26,31]. Además se pretende verificar la precisión de k -NN debido a que en [26,27,28,31] se reporta una precisión de clasificación variable para 1-NN al utilizar WEKA. Por otra parte se someterá al conjunto de datos a un ACP, la cual es una técnica de extracción de características basada en la varianza, después de aplicar ACP se pretende verificar si k -NN, SVM y RNA logran mejorar la precisión reportada en la literatura utilizando el conjunto de datos original. De forma análoga, se aplica PSC, la cual es una técnica de selección de instancias basada en k -NN y C -means. Se utiliza ACP debido a que en las gráficas de dispersión por pares se observa correlación entre algunas variables. Desde otra perspectiva, PSC fue seleccionado debido a que k -NN es uno de los algoritmos que genera mayor rendimiento en el conjunto de datos original. La figura 2 muestra el diagrama de flujo del proceso planteado.

6. Preliminares

Para la etapa de experimentos se utilizó MATLAB R2013a, la plataforma WEKA 3,6 y S.O. Ubuntu 14,04 a 64 bits. El hardware utilizado fue una máquina de escritorio con procesador Phenom X4 a 2,5 GHz, 8 Gb de RAM. Para las pruebas de precisión se utilizó validación cruzada estratificada con 10 particiones. Para la búsqueda greedy de los diversos parámetros de SVM y RNA se seleccionó de forma aleatoria el 50 % del conjunto de datos. Para obtener el mejor modelo de PSC se utilizó validación cruzada estratificada. Para cada iteración se aplicó PSC

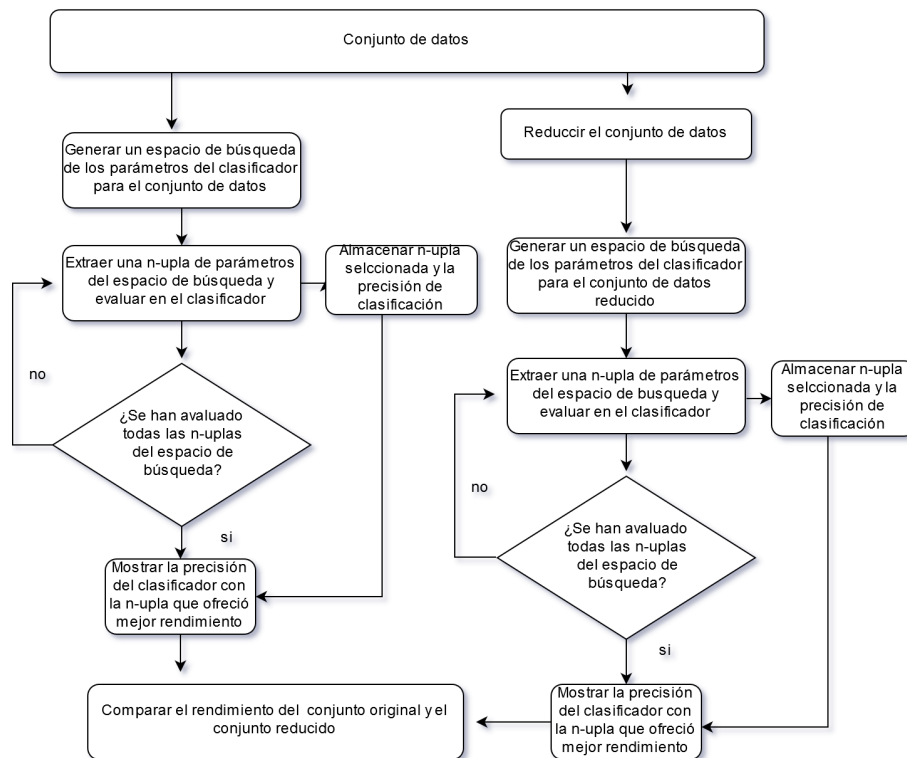


Fig. 2. Diagrama de flujo del método propuesto.

al conjunto de entrenamiento y se validó con el conjunto de prueba con 1-NN. El subconjunto con el que se obtuvo el mejor rendimiento para las 10 iteraciones se seleccionó y se descartaron las instancias restantes, tanto del conjunto de prueba como del conjunto de entrenamiento. Para las pruebas de clasificación sólo se ocupó el subconjunto seleccionado.

La primera interacción con los datos fue durante el análisis exploratorio, donde se observó su distribución, la posible correlación entre variables y se eliminaron 4 datos atípicos. El conjunto de datos final cuenta con 6,722 casos de los ojos cerrados y 8,254 casos de ojos abiertos. En [26,27,28,31] se reporta que la mejor precisión de clasificación es obtenida con k -NN y KStar, ambos clasificadores basados en instancias. [27] aborda el problema mediante máquinas de soporte vectorial, no obteniendo los resultados deseados, en [31] utilizan k -NN y un perceptrón multicapa para clasificar el conjunto, k -NN obtiene los mejores resultados. Las primeras pruebas se realizan con k -NN y KStar, obteniendo resultados similares, 1-NN obtuvo una precisión de clasificación de 97,42, mientras que para KStar su precisión es de 96,66.

7. Experimentos y resultados

El siguiente clasificador elegido para las pruebas es SVM, mediante una búsqueda greedy, se optimizaron los parámetros $C = 2^{-5}, \dots, 2^{10}$ y $gamma = 2^5, \dots, 2^{-10}$ para los 4 kernels que incluye la librería LIBSVM [45] de MATLAB. Los mejores resultados fueron para el kernel RBF y kernel polinomial con una precisión de clasificación de 98,865 y 93,095 respectivamente. Los parámetros utilizados son: $C = 13,4543$ y $gamma = 0,0017074$ para el kernel RBF y para el kernel polinomial $C = 0,42045$ y $gamma = 0,005309$. Las pruebas de clasificación utilizando una red neuronal feedforward y el algoritmo de backpropagation con momentum y tasa de aprendizaje variable se realizaron en MATLAB. Primero se realizó una búsqueda greedy para obtener los parámetros iniciales más adecuados para la constante de momentum, el número de capas ocultas, el número de neuronas por capa, la tasa de aprendizaje, el incremento de la tasa de aprendizaje y el decremento de la misma. La tabla 1 muestra el mejor resultado de la búsqueda.

Tabla 1. Valores de los parámetros de la red neuronal.

Parámetro	Mejor valor
Tasa de aprendizaje	0,005
Incremento del aprendizaje	0,0675
Decremento del aprendizaje	0,255
Momentum	0,63
Núm. épocas	50000
Configuración de la red	[14, 7, 4, 2, 1]
Precisión	88,67

Debido a que el número de muestras del conjunto de datos se acerca a 15,000 instancias con 14 características por muestras, los experimentos siguientes se enfocan a reducir el número de instancias y en número de características del conjunto de datos para observar el comportamiento de la precisión de clasificación con los algoritmos utilizados anteriormente.

La mejor precisión de clasificación reportada en el estado del arte se obtiene con k -NN y KStar, por este motivo se utiliza el algoritmo PSC. Este algoritmo reduce el conjunto de entrenamiento mediante clusters y k -NN. El parámetro c que recibe PSC es el número de clusters a crear. En [38] se menciona que el número de clusters recomendado es $c = f * l$, con $f = 10$, donde l representa en número de clases y f una constante de intensidad. Para las pruebas de reducción con PSC se hizo variar el parámetro $f \in [1, 8000]$. Para la obtención del modelo que mejor generalize el proceso de discriminación de nuevas muestras se utilizó k -NN. El mejor modelo creado por PSC obtuvo un rendimiento de 96,79 para 1-NN. Los detalles de este modelo son los siguientes:

Tabla 2. Valores de los parámetros para el mejor modelo obtenido por PSC.

Parámetro	valor
Núm. de clusters	6000
Umbral de k -meas	1
Núm. de instancias finales	6,275
% de retención	41,90
Rendimiento en el entrenamiento	97,80
Parámetro k	1

Para KStar el rendimiento con este subconjunto es de 94,659, para el mejor modelo de la red neuronal el rendimiento es 88,74 y para SVM con RBF se obtiene una precisión de 98,67.

Por último, se sometió al conjunto de datos original a un análisis de componentes principales, del cual se tomaron los primeros 7 componentes. Para los 7 primeros componentes principales, la retención de la varianza es de 91,03%. Al someter la nueva representación de los datos a los clasificadores el rendimiento se decremento de forma considerable comparado con los resultados anteriores. Estos resultados no son los esperados debido a que en las gráficas de dispersión por pares se observaba una aparente correlación entre algunos pares de variables. La siguiente tabla(ver tabla 3) resume el rendimiento obtenido para los 4 algoritmos de clasificación aplicado a este subconjunto.

Tabla 3. Tabla de precisión de clasificación para el subconjunto de datos obtenido al aplicar ACP.

Clasificador	Precisión
1-NN	86,879
KStar	86,99
RNA ([7, 4, 2, 1])	72,96
SVM (RBF, C = 1, gamma = 1,992)	89,95

8. Conclusiones y trabajo a futuro

En esta investigación se logra comprobar que las SVM ofrece resultados superiores a los reportados en el estado del arte cuando los parámetros del algoritmo son los adecuados. También podemos constatar que k -NN y KStar son una muy buena opción para la discriminación de los estados de los ojos. Otro punto a favor es que ofrecen resultados de rendimiento similares cuando el conjunto es reducido por PSC. Es válido mencionar que PSC es una buena elección para reducir un conjunto de datos en el que los mejores resultados de precisión los obtienen clasificadores basados en distancias. Las redes neuronales no ofrecieron un rendimiento similar al de los demás clasificadores utilizados,

pero si son superiores a los reportados en [31] con el perceptrón multicapa. Para la RNA es posible mejorar su precisión de clasificación si los parámetros de la red logran ser optimizados mediante otro enfoque, ya que una búsqueda greedy no es la mejor opción. Por otro lado, ACP no ofreció los resultados esperados, pero se puede notar que el rendimiento de SVM sigue siendo superior, seguido por k -NN y KStar.

Como trabajo a futuro se plantea mejorar el desempeño de la red optimizando sus parámetros y topología mediante un algoritmo genético. Los algoritmos genéticos pueden ser capaces de encontrar una mejor configuración para el número de capas, número de nodos por capa, conexiones entre los nodos, conexiones entre las capas, pesos y funciones de activación de los nodos. También se plantea experimentar con otros algoritmos de selección de instancias y de extracción de características esperando obtener una mejor estructura de los datos en una forma más compacta.

Referencias

1. Lugo-Reyes, S.O., Maldonado-Colín, G., Murata, C.: Inteligencia artificial para asistir el diagnóstico clínico en medicina. *Revista Alergia*, 61:110–120 (2014)
2. Bozek, J., Mustra, M., Delac, K., Grgic, M.: A survey of image processing algorithms in digital mammography. *Recent Advances in Multimedia Signal Processing and Communications*, 231, pp. 631–657 (2009)
3. Cheng, H. D., Shi, X., Min, R., Hu, L., Cai, X., Du, H. N.: Approaches for automated detection and classification of masses in mammograms. *Pattern Recognition*, Vol. 39, No. 4, pp. 646–668 (2006)
4. Murata, C., Ramírez, A.B., Ramírez, G., Cruz, A. et al.: Análisis discriminante para predecir el diagnóstico clínico de inmunodeficiencias primarias: reporte preliminar. *Revista Alergia*, 62:125–133 (2015)
5. Khattree, R.: *Computational methods in biomedical research*. Baton Rouge, LA: Chapman & Hall (2007)
6. Cleophas, T.J., Zwinderman, A.H.: *Machine Learning in Medicine*. Springer (2013)
7. Wang, T., Guan, S.-U., Man, K.L., Ting, T.O.: EEG Eye State Identification Using Incremental Attribute Learning with Time-Series Classification. *Mathematical Problems in Engineering*, Article ID 365101, 9 pages (2014)
8. Pour, P., Gulrez, T., Al Zoubi, O., Gargiulo, G., Calvo, R.: Brain Computer Interface: Next Generation Thought Controlled Distributed Video Game Development Platform. *Proc of the CIG, Perth, Australia* (2008)
9. Zheng, W.L., Dong, B.N., Lu, B.L.: Multimodal emotion recognition using EEG and eye tracking data. *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pp. 5040–5043 (2015)
10. Taylor, D.M., Tillery, S.I.H., Schwartz, A.B.: Direct Cortical Control of 3D Neuroprosthetic Devices. *Science* 296, pp. 1829–1832 (2002)
11. Lebedev, M.A., Nicolelis, M.A.L.: Brain-Machine Interfaces: Past, Present and Future. *Trends in Neurosciences*, Vol. 29, No. 9, pp. 536–546 (2006)
12. Wang, Y. & Makeig, S.: Predicting Intended Movement Direction Using EEG from Human Posterior Parietal Cortex. Dylan Schmorrow; Ivy V. Estabrooke & Marc Grootjen, eds., 'HCI (16)', Springer, pp. 437–446 (2009)

13. Muzik, J., Hana, K.: Real-time BSPM processing system, 4th European Conference of the International Federation for Medical and Biological Engineering, Vol. 22, pp. 377–380 (2009)
14. Doud, A.J., Lucas, J.P., Pisansky, M.T., He, B.: Continuous three-dimensional control of a virtual helicopter using a motor imagery based brain-computer interface. *PLoS One* 6(10) (2011)
15. McFarland, D.J., Sarnacki, W.A., Wolpaw, J.R.: Electroencephalographic (EEG) control of three-dimensional movement. *J Neural Eng* 7:036007 (2010)
16. Royer, A.S., Doud, A.J., Rose, M.L., He, B.: EEG control of a virtual helicopter in 3-dimensional space using intelligent control strategies. *IEEE Trans Neural Syst Rehabil Eng*, Vol. 18, No.6, pp. 581589 (2010)
17. Navalyal, G.U., Gavas, R.D.: A dynamic attention assessment and enhancement tool using computer graphics. *Navalyal and GavasHuman-centric Computing and Information Sciences*, Vol. 4, No. 11, Springer (2014)
18. Dauwels, J., Vialatte, F., Cichocki, A.: Diagnosis of Alzheimer’s disease from EEG signals: where are we standing? Vol. 7, No. 6, pp. 487–505 (2010)
19. Yeo, M.V.M., Li, X., Shen, K., Wilder-Smith, E.P.V.: Can SVM be used for automatic EEG detection of drowsiness during car driving? *Safety Science*, Vol. 47, No. 1, pp. 115–124 (2009)
20. Estévez, P.A., Held, C.M., Holzmann, C.A., Perez, C.A., Pérez, C.A., Heiss, J., Garrido, M., Peirano, P.: Polysomnographic pattern recognition for automated classification of sleep-waking states in infants. *Medical and Biological Engineering and Computing*, Vol. 40, No. 1, pp. 105–113 (2001)
21. Polat, K., Gne, S.: Classification of epileptic form EEG using a hybrid system based on decision tree classifier and fast Fourier transform. *Applied Mathematics and Computation*, Vol. 187, No. 2, pp. 1017–1026 (2007)
22. Sadatnezhad, K., Boostani, R., Ghanizadeh, A.: Classification of BMD and ADHD patients using their EEG signals. *Expert Systems with Applications*, Vol. 38, No. 3, pp. 1956–1963 (2011)
23. Sulaiman, N., Taib, M.N., Lias, S., Murat, Z.H., Aris, S.A.M., Hamid, N.H.A.: Novel methods for stress features identification using EEG signals. *International Journal of Simulation: Systems, Science and Technology*, Vol. 12, No. 1, pp. 27–33 (2011)
24. Nguyen, T., Nguyen, T.H., Truong, K.Q.D., van Vo, T.: A mean threshold algorithm for human eye blinking detection using EEG. *Proceedings of the 4th International Conference on the Development of Biomedical Engineering in Vietnam*, pp. 275–279, Ho Chi Minh City, Vietnam (2013)
25. Fukuda, O., Tsuji, T., Kaneko, M.: Pattern classification of EEG signals using a log-linearized Gaussian mixture neural network. *Proceedings of the IEEE International Conference on Neural Networks. Part 1 (of 6)*, pp. 2479–2484, Perth, Australia (1995)
26. Rösler, O., Suendermann, D.: First step towards eye state prediction using EEG. *Proceedings of the International Conference on Applied Informatics for Health and Life Sciences (AIHLS ’13)*, Istanbul, Turkey (2013)
27. Jain, N., Bhargava, S., Shivani, S., Goyal, D.: Eye state prediction using EEG by supervised learning. *International Journal of Science, Engineering and Technology* (2015)
28. Sahu, M., Nagwani, N.K., Verma, S., Shirke, S.: Performance Evaluation of Different Classifier for Eye State Prediction Using EEG Signal. *International Journal of Knowledge Engineering*, Vol. 1, No. 2 (2015)

29. Hall, M. Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, Vol. 11, pp. 10–18 (2009)
30. Wang, T., Guan, S.-U., Man, K.L., Ting, T.O.: EEG Eye State Identification Using Incremental Attribute Learning with Time-Series Classification. *Mathematical Problems in Engineering*.
31. Sabanc, K., Koklu, M.: The Classification of Eye State by Using k-NN and MLP Classification Models According to the EEG Signals. *International Journal of Intelligent Systems and Applications in Engineering*, pp.127–130 (2015)
32. van Erp, J., Reschke, S., Grootjen, M., Brouwer, A.-M.: Brain Performance Enhancement for Military Operators. *HFM Symposium on Human Performance Enhancement for NATO Military Operations (Science and Technology and Ethics)* (2009)
33. Pham, T., Tran, D.: Emotion recognition using the emotiv epoc device. *Lecture Notes in Computer Science*, Vol. 7667 (2012)
34. Kuncheva, L.I., Bezdek, J.C.: Nearest prototype classification, clustering, genetic algorithms, or random search? *IEEE Trans Syst Man Cybern C*28(1), pp. 160–164 (1998)
35. Bezdek, J.C., Kuncheva, L.I.: Nearest prototype classifier designs: an experimental study. *Int J Intell Syst*, Vol. 16, No. 12, pp. 1445–1473 (2001)
36. Wilson, D.R., Martínez, T.R.: Reduction techniques for instance-based learning algorithms. *Mach Learn*, Vol. 38, pp. 257–286 (2000)
37. Brighton, H., Mellish, C.: Advances in instance selection for instance-based learning algorithms. *Data Min Knowl Disc*, Vol. 6, No. 2, pp. 153–172 (2002)
38. Olvera-López, J.A., Carrasco-Ochoa, J.A., Martínez-Trinidad: A new Fast Prototype Selection Method based on Clustering. *Pattern Analysis and Applications*. *Pattern Analysis and Applications*, Vol. 13, No. 2, pp. 131–141 (2010)
39. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
40. Rojas, R.: *Neural Networks: A Systematic Introduction*. Springer-Verlag New York, Inc., New York, NY, USA. (1996)
41. Moreira, M., Fiesler, E.: *Neural Networks with Adaptive Learning Rate and Momentum Terms*. Technical report, Institut Dalle Molle d'intelligence artificielle perceptive (1995)
42. Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Prentice Hall (1998)
43. Lee, J.A., Verleysen, M.: *Nonlinear Dimensionality Reduction*. (1st ed.) Springer Publishing Company, Incorporated (2007)
44. Venna, J., Peltonen, J., Nybo, K., Aidos, H., Kaski, S.: Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, Vol. 11, pp. 451–490 (2010)
45. Chang, C., Lin, C.: LIBSVM: A library for Support Vector Machines. *ACM Trans Intell Syst Technol*, Vol. 2, No. 3, p. 27 (2011)
46. Cortes, C., Vapnik, V.N.: Support vector networks. *Mach Learn*, Vol. 20, No. 3, pp. 273–97 (1995)