



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**SmartBeds
Minería de datos aplicada a
la detección de ataques
epilépticos mediante
colchones inteligentes.**



Presentado por Alicia Olivares Gil
en Universidad de Burgos — 13 de junio
de 2019

Tutores: Álgvar Arnaiz González y José
Francisco Díez Pastor



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Álvar Arnaiz González, profesor del departamento de Ingeniería Civil, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Alicia Olivares Gil, con DNI 71299943N, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado "SmartBeds - Minería de datos aplicada a la detección de ataques epilépticos mediante colchones inteligentes".

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 13 de junio de 2019

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. Álvar Arnáiz González

D. José Francisco Díez Pastor

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
1.1. Estructura de la memoria	2
1.2. Materiales adjuntos	2
Objetivos del proyecto	5
2.1. Objetivos generales	5
2.2. Objetivos técnicos	6
2.3. Objetivos personales	7
Conceptos teóricos	9
3.1. Secciones	9
3.2. Referencias	9
3.3. Imágenes	10
3.4. Listas de ítems	10
3.5. Tablas	11
Técnicas y herramientas	13
4.1. Técnicas metodológicas	13
4.2. Herramientas en la fase de investigación	14
4.3. Herramientas en la fase de diseño de la aplicación	16
4.4. Herramientas en la fase de desarrollo de la aplicación	17

4.5. Otras herramientas generales	18
Aspectos relevantes del desarrollo del proyecto	19
Trabajos relacionados	21
Conclusiones y Líneas de trabajo futuras	23
Bibliografía	25

Índice de figuras

3.1. Autómata para una expresión vacía	10
--	----

Índice de tablas

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	12
---	----

Introducción

La epilepsia es un trastorno neurológico provocado por la alteración de la actividad normal de una región cerebral, que desencadena crisis caracterizadas por convulsiones musculares reiteradas y, en ocasiones, pérdida de la consciencia. Se trata de una de las enfermedades neurológicas más habituales, y aunque las crisis epilépticas pueden experimentarse de forma aislada o durante periodos de tiempo limitados, una gran cantidad de la población las sufre de forma crónica. Según la Federación Española de Epilepsia [2] alrededor de 700 000 personas en España padecen o han padecido epilepsia a lo largo de su vida, y más de 200 000 la padecen de forma activa.

Para una persona con epilepsia crónica, la detección inmediata de una crisis es vital para permitir la aplicación de primeros auxilios que ayuden a evitar consecuencias permanentes. Actualmente, en la bibliografía se habla de varias técnicas para la detección automática de crisis [7, 8], la mayoría basadas en Electroencefalogramas (EEG) o en el uso de dispositivos portátiles (*wearables*) como pulseras inteligentes basadas en la monitorización de las constantes vitales del paciente.

La detección mediante EEG se usa principalmente para diagnósticos médicos, ya que los dispositivos que se necesitan son demasiado costosos o aparatosos como para ser usados en el día a día del paciente. Por otro lado, las pulseras inteligentes resultan más convenientes para este cometido, ya que son más baratas y cómodas de utilizar. Sin embargo, ambas técnicas requieren del uso consciente y continuado de los dispositivos de detección, lo que puede suponer un inconveniente para pacientes dependientes o con necesidades especiales. Por esta razón se propone el uso de colchones inteligentes para la detección automática de crisis nocturnas, mediante sensores de presión y biométricos incorporados en el interior del propio colchón.

Sea cual sea el dispositivo utilizado, la captación de los datos (actividad eléctrica del cerebro, constantes vitales, presiones, etc.) no basta para detectar una crisis. Es necesario un procesamiento adecuado para determinar si estos datos corresponden o no con una crisis epiléptica. Para ello, las técnicas de minería de datos permiten generar modelos de clasificación capaces de realizar esta tarea. Para este trabajo de fin de grado, el principal objetivo será encontrar un modelo de clasificación efectivo mediante la aplicación de este tipo de técnicas sobre los datos disponibles.

1.1. Estructura de la memoria

Esta memoria incluye los siguientes apartados:

- **Objetivos del proyecto:** se definen los objetivos generales, técnicos y personales que se persiguen con la realización de este trabajo.
- **Conceptos teóricos:** TODO
- **Técnicas y herramientas:** TODO
- **Aspectos relevantes del desarrollo del proyecto:** TODO
- **Trabajo relacionados:** TODO
- **Conclusiones y líneas de trabajo futuras:** TODO

1.2. Materiales adjuntos

- **Anexos:**
 - Plan de Proyecto Software
 - Especificación de Requisitos
 - Especificación de diseño
 - Documentación técnica de programación
 - Documentación de usuario
- **Cuaderno de investigación:** recoge las explicaciones de todas las técnicas probadas, los resultados y las comparativas de todos los experimentos realizados con el fin de encontrar el mejor modelo de clasificación para el problema.

- **Experimentos:** Conjunto de notebooks de jupyter que contienen todos los experimentos realizados.
- **App de Android:** Archivo .apk para la distribución e instalación de la aplicación para Android desarrollada con el fin de mostrar la aplicabilidad del modelo de clasificación.

A la memoria y a todos los demás materiales adjuntos se puede acceder a través del repositorio del proyecto en GitHub: <https://github.com/aog0036/TFG-SmartBeds>.

Objetivos del proyecto

En este apartado se exponen los objetivos perseguidos con la realización de este trabajo:

2.1. Objetivos generales

- Investigar sobre técnicas del estado del arte aplicadas a problemas similares.
- Aplicar técnicas de minería de datos siguiendo los pasos del Descubrimiento de Conocimiento en Bases de Datos (*KDD*) [5].
- Explorar, aplicar y comparar distintas formas de preprocesado de los datos (filtrado, normalización, transformación, etc.).
- Usar técnicas de proyección de datos a dos dimensiones para comprobar si los instancias de «crisis» son fácilmente separables de las instancias de «no crisis».
- Probar modelos de clasificación para conjuntos de datos con preprocesados basados en estadísticas simples.
- Probar modelos de clasificación para conjuntos de datos con preprocesados basados en características de series temporales.
- Probar modelos de clasificación mediante *ensembles* para conjuntos de datos desequilibrados.
- Probar detección de anomalías mediante un modelo *One-Class*.

- Comparar el rendimiento de los modelos obtenidos.
- Comparar distintas métricas usadas para evaluar el rendimiento los modelos obtenidos.
- Generar un modelo de clasificación capaz de detectar crisis epilépticas a partir de los datos disponibles.
- Desarrollar una app de Android para mostrar la aplicabilidad del modelo de clasificación generado.

2.2. Objetivos técnicos

- Aprender \LaTeX , en concreto su uso en herramientas de escritorio y online (tales como Overleaf).
- Realizar y exponer los resultados de los experimentos en notebooks de jupyter empleando código Python.
- Usar bibliotecas de minería de datos en Python, en concreto sklearn [6], para la proyección de los datos y para la obtención y evaluación de los modelos.
- Generar un conjunto de Transformadores compatibles con sklearn para poder aplicar las técnicas de preprocesado de forma directa.
- Utilizar bibliotecas de procesamiento y visualización de datos tales como Pandas y Matplotlib.
- Usar la biblioteca tsfresh [1] para extracción de características en series temporales.
- Usar la herramienta Weka [3] para probar modelos de clasificación mediante ensembles en conjuntos de datos desequilibrados.
- Desarrollar una app Android con soporte para API 23 y superiores.
- Usar peticiones HTTP desde la app para la comunicación con la API del servidor remoto.
- Usar Socket.IO desde la app para la obtención de datos en tiempo real del servidor remoto.
- Usar la plataforma GitHub junto con la extensión ZenHub para la gestión del proyecto.

- Aplicar la metodología Scrum adaptada a un proyecto con fines educativos.
- Realizar test TODO.

2.3. Objetivos personales

- Iniciarme en el campo de la investigación.
- Explorar técnicas y herramientas aplicadas a la minería de datos.
- Aprender a generar documentación con L^AT_EX.
- Iniciarme en el desarrollo de aplicaciones Android.

Conceptos teóricos

En aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

Algunos conceptos teóricos de L^AT_EX¹.

3.1. Secciones

Las secciones se incluyen con el comando `section`.

Subsecciones

Además de secciones tenemos subsecciones.

Subsubsecciones

Y subsecciones.

3.2. Referencias

Las referencias se incluyen en el texto usando `cite` [9]. Para citar webs, artículos o libros [4].

¹Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz

3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de \LaTeX , pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:



Figura 3.1: Autómata para una expresión vacía

3.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

▪

3.5. Tablas

Igualmente se pueden usar los comandos específicos de \LaTeX o bien usar alguno de los comandos de la plantilla.

Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket		X	X	X	X
MikTeX					X
TeXMaker					X
Astah					X
Balsamiq Mockups		X			
VersionOne		X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

Técnicas y herramientas

En este apartado se presentan las técnicas metodológicas y las herramientas que se han usado en las distintas fases del desarrollo del proyecto.

4.1. Técnicas metodológicas

Scrum

Como se explica más detenidamente en el apartado de Planificación temporal del Apéndice A, para la planificación y desarrollo del proyecto se ha utilizado la metodología ágil *Scrum*, manteniendo el enfoque incremental de los *sprints* pero adaptándola al contexto de un trabajo con fines educativos.

KDD

En la fase de investigación se ha seguido el proceso de Descubrimiento de Conocimiento en Bases de Datos o *KDD*, dedicado a encontrar un modelo válido y útil en la medida en la que sirva para describir los patrones subyacentes de los datos. El término *KDD* suele ser empleado a menudo como sinónimo de minería de datos, pero esta corresponde en realidad con uno solo de los pasos del proceso. Se suele hablar de las siguientes fases englobados en el proceso de *KDD*:

- **Limpieza de datos:** Consiste en eliminar los datos inconsistentes o con ruido.
- **Integración de datos:** Se integran los datos de todas las fuentes disponibles en un formato uniforme y adecuado para los pasos posteriores.

- **Selección de datos:** Se trata de seleccionar solo aquellos datos relevantes para la tarea de análisis.
- **Transformación de datos:** Se realizan transformaciones y cálculos a partir de los datos en bruto con el fin de aplicar las técnicas de minería a las formas más apropiadas de los mismos.
- **Minería de datos:** Aplicación de técnicas para encontrar patrones subyacentes.
- **Evaluación de patrones:** Se estudia hasta qué punto los patrones encontrados son interesantes.
- **Presentación del conocimiento:** Consiste en ofrecer una representación comprensible y útil de los patrones y del conocimiento extraído.

Estos pasos no se aplican necesariamente de forma secuencial, ya que en muchos casos conviene volver a pasos anteriores tras la evaluación de los resultados del paso actual.

4.2. Herramientas en la fase de investigación

Anaconda

Anaconda es un administrador de paquetes y de entornos considerado un estándar para el desarrollo de minería de datos en lenguajes como Python y R. Al instalar Anaconda se tienen automáticamente disponibles más de 200 paquetes, además de ofrecer la posibilidad de añadir nuevos de forma sencilla.

Licencia: *New BSD License*

Jupyter Notebook

Los experimentos se han desarrollado en código Python distribuido en múltiples jupyter notebooks, ya que ofrecen un formato de estructuración y documentación del código muy adecuado para la investigación. *Jupyter Notebook* se encuentra disponible al instalar Anaconda.

Licencia: *Modified BSD License*

Scikit-Learn

Es la principal biblioteca empleada en la fase de investigación. Incluye modelos de clasificación, predicción y clustering de todo tipo y herramientas para entrenarlos, explotarlos y evaluarlos de forma sencilla. Está especialmente diseñada para operar con las bibliotecas *NumPy* y *SciPy*, y es compatible con *Pandas*.

Licencia: *New BSD License*

Weka

En algunas partes de la investigación se han usado modelos de Weka, otra plataforma para aprendizaje automático y minería de datos escrita en Java. Fue utilizada sobre todo para el entrenamiento de *ensembles* aplicados a conjuntos de datos desequilibrados.

Licencia: *GNU General Public License*

tsfresh

Es una biblioteca de Python dedicada al cálculo de grandes cantidades de características de series temporales. Permite calcular 64 tipos distintos de características y dispone de herramientas para filtrarlas en función de su relevancia. Es compatible con *Pandas*.

Licencia: *MIT License*

DEAP

Es el framework de python para computación evolutiva más utilizado. Se ha empleado como una de las alternativas para intentar encontrar la mejor selección de características de series temporales mediante un algoritmo genético, esperando que al aplicar este conjunto en la fase de minería se obtuviesen los mejores resultados posibles.

Licencia: *GNU Lesser General Public License v3.0*

tmux

Se trata un multiplexador de terminales que permite abrir varias sesiones simultáneamente y dejarlas corriendo en segundo plano. Esta herramienta ha resultado especialmente útil para la ejecución de los experimentos más

costosos en cuanto a tiempo de ejecución. Debido a que las ejecuciones de los experimentos se han llevado a cabo sobre un equipo de cómputo del grupo de investigación de los tutores mediante una conexión ssh que a menudo se cerraba en medio de un trabajo, ha sido necesario abrir una sesión de tmux en el equipo para cada una de estas ejecuciones de forma que, aunque se perdiera la conexión, el proceso siguiera corriendo en segundo plano y pudiéramos acceder a los resultados reactivando la sesión cuando el trabajo finalizase.

Licencia: *BSD License*

Otras bibliotecas relevantes

- *NumPy* y *Pandas* para la gestión, modificación y presentación de los datos.
- *Matplotlib* para la presentación gráfica de los resultados.

4.3. Herramientas en la fase de diseño de la aplicación

StarUML

Software de edición de diagramas UML utilizado en la fase de modelado.

Licencia: Propietaria aunque dispone de una demo gratuita.

Material Design

Es una guía de estilo desarrollada por Google e integrada a partir de la versión *Lollipop* (5.0) de Android. Esta guía de estilos trata los elementos de la interfaz como elementos materiales, con unas dimensiones y una posición definida dentro del espacio (no solo en el plano, también en una tercera dimensión representada mediante el atributo *elevation*), propone una serie de dimensiones idóneas para cada tipo de elemento (textos, botones, tarjetas, etc.), y define la forma de generar el esquema de colores de la interfaz.

Pencil

Software de prototipado de interfaces gráficas. Permite incluir paquetes para incorporar elementos propios de *Material Design* a los prototipos, por

lo que proporciona una visión más próxima al aspecto final de las interfaces de la aplicación que otros tipos de prototipado.

Licencia: *GNU Public License version 2*

4.4. Herramientas en la fase de desarrollo de la aplicación

Android Studio

Es el entorno de desarrollo integrado oficial de Android, disponible para Windows, GNU/Linux y MacOS. Android Studio incluye, entre otras muchas cosas, un editor de código, un editor gráfico de *layouts*, emuladores para todas las versiones de Android existentes, y soporte para construcción automática con Gradle.

Licencia: *Apache License 2.0*

Gradle

Herramienta para la automatización de la construcción del software para proyectos Java. Es la herramienta soportada de forma oficial por Android.

Licencia: *Apache License 2.0*

Android Support Library

Es la biblioteca que gestiona la compatibilidad de funciones de versiones avanzadas con su equivalente en versiones anteriores de Android. Además, incluye *layouts*, elementos y utilidades que no están disponibles en el framework oficial. Aunque la biblioteca recomendada actualmente para este cometido es *AndroidX*, la cual incluye las mismas funcionalidades y algunas más, se ha escogido *Android Support Library* por su simplicidad y por su documentación clara y completa, lo que resulta de mucha utilidad cuando se desarrolla una aplicación Android por primera vez.

Licencia: *Apache License 2.0*

MPAndroidChart

Es una biblioteca para generación de gráficos en aplicaciones Android. En este caso, y aunque no está pensada para ello, se ha utilizado para

visualizar gráficas dinámicas cuyos datos se modifican en tiempo real. Para implementar esta funcionalidad se tuvo en cuenta también la biblioteca SciChart, al ser la biblioteca de referencia para generación de gráficos en tiempo real de Android, pero se descartó al tratarse de un software de pago.

Licencia: *Apache License 2.0*

Socket.IO-client Library

Es una biblioteca para gestión de comunicación mediante sockets para Java. El uso de esta biblioteca viene impuesto por la implementación de la API del servidor remoto, que gestiona el envío de datos en tiempo real de esta forma.

Licencia: *MIT License*

4.5. Otras herramientas generales

- **Git** como sistema de control de versiones distribuido.
- **GitHub** como plataforma para el *hosting* del repositorio del proyecto.
- **ZenHub** como extensión de GitHub para la gestión del proyecto basada en la metodología *Scrum*.
- **GitKraken** como cliente de Git mediante interfaz gráfica. (*GNU Public License*).
- **Overleaf** como editor colaborativo de \LaTeX online para la generación del cuaderno de investigación conjunto.
- **T_EXstudio** como editor de \LaTeX para la generación de la memoria y los anexos. (*GNU General Public License v2*).
- **FileZilla** como aplicación para transferencia de ficheros. Soporta los protocolos FTP, SFTP y FTPS. (*GNU General Public License v2*).

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros³, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

- [1] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W Kempa-Liehr. Time series feature extraction on basis of scalable hypothesis tests (tsfresh—a python package). *Neurocomputing*, 307:72–77, 2018.
- [2] Federación Española de Epilepsia. Qué es la epilepsia, 2013. [Internet; consultado 11-Junio-2019].
- [3] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
- [4] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [5] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 03 1996.
- [6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [7] Sriram Ramgopal, Sigride Thome-Souza, Michele Jackson, Navah Ester Kadish, Iván Sánchez Fernández, Jacquelyn Klehm, William Bosl, Claus Reinsberger, Steven Schachter, and Tobias Loddenkemper. Seizure detection, seizure prediction, and closed-loop warning systems in epilepsy. *Epilepsy & behavior*, 37:291–307, 2014.

- [8] Alexandros T Tzallas, Markos G Tsipouras, Dimitrios G Tsalikakis, Evangelos C Karvounis, Loukas Astrakas, Spiros Konitsiotis, and Margaret Tzaphlidou. Automated epileptic seizure detection methods: a review study. In *Epilepsy-histological, electroencephalographic and psychological aspects*. IntechOpen, 2012.
- [9] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2015. [Internet; descargado 30-septiembre-2015].