

AVANCES EN INFORMÁTICA Y AUTOMÁTICA

Décimotercer workshop



Avances en Informática y Automática

Décimotercer *workshop*

Avances en Informática y Automática

Decimotercer *workshop*

Editor
G. Hernández

Publicado en España por:

Departamento de Informática y Automática Facultad de Ciencias

Universidad de Salamanca

Plaza de los Caídos s/n

37008, Salamanca, España

Tel.: + 34 923294653

Fax: + 34 923294514

Web: <http://mastersi.usal.es>

Email: mastersi@usal.es

ISBN **978-84-09-23575-9**

Editor:

Guillermo Hernández González

Prólogo

El Máster Oficial en Sistemas Inteligentes de la Universidad de Salamanca tiene como principal objetivo promover la iniciación de los estudiantes en el ámbito de la investigación. El congreso organizado por el Departamento de Informática y Automática que se celebra dentro del Máster en Sistemas Inteligentes de la Universidad de Salamanca proporciona la oportunidad ideal para que sus estudiantes presenten los principales resultados de sus Trabajos de Fin de Máster y obtengan una realimentación del interés de los mismos.

La decimotercera edición del *workshop* «Avances en Informática y Automática», correspondiente al curso 2018-2019, ha sido un encuentro interdisciplinar donde se han presentado trabajos pertenecientes a un amplio abanico de líneas de investigación. Todos los trabajos han sido supervisados por investigadores de reconocido prestigio pertenecientes a la Universidad de Salamanca, proporcionando el marco idóneo para sentar las bases de una futura tesis doctoral. Entre los principales objetivos del congreso se encuentran:

- Ofrecer a los estudiantes un marco donde exponer sus primeros trabajos de investigación.
- Proporcionar a los participantes un foro donde discutir ideas y encontrar nuevas sugerencias de compañeros, investigadores y otros asistentes a la reunión.
- Permitir a cada estudiante una realimentación de los participantes sobre su trabajo y una orientación sobre las futuras direcciones de investigación.
- Contribuir al desarrollo del espíritu de colaboración en la investigación.

Organización

El *workshop* «Avances en Informática y Automática» está organizado por el Departamento de Informática y Automática de la Universidad de Salamanca.

Comité organizador

Dr. Vidal Moreno Rodilla
Dr. Francisco J. García Peñalvo
Dr. Roberto Therón Sánchez
Dr. José Rafael García-Bermejo Giner
Dra. Belén Pérez Lancho
Dr. Carlos García-Figuerola Paniagua

Índice general

| | |
|---|-----|
| Sistema de análisis de vídeo mediante la utilización del marco metodológico de los sistemas de razonamiento basados en casos y el uso de algoritmos de aprendizaje profundo | 1 |
| <i>Guillermo Hernández, Angélica González Arrieta y Sara Rodríguez</i> | |
| Sistema basado en blockchain para la realización de análisis en entornos laborales informatizados | 27 |
| <i>Francisco Lecumberri de Alba, Juan M. Corchado y Javier Prieto</i> | |
| Sistema de recomendación de música basado en etiquetado social | 47 |
| <i>Alberto Moro Calvo, Vivian López Batista y Gabriel Villarrubia González</i> | |
| Sistema de recomendación sensible al contexto basado en información biométrica | 69 |
| <i>Alejandra Ospina Bohórquez, María N. Moreno y Ana B. Gil González</i> | |
| Diseño de pipelines para el análisis y extracción de conocimiento a partir de exomas individuales | 87 |
| <i>Alejandro Rodríguez Díaz y Luis Antonio Miguel Quintales</i> | |
| Sistema de predicción de mercados de criptomonedas | 101 |
| <i>Diego Valdeolmillos Villaverde, Pablo Chamoso y Juan M. Corchado</i> | |
| Autores | 115 |

Sistema de análisis de vídeo mediante la utilización del marco metodológico de los sistemas de razonamiento basados en casos y el uso de algoritmos de aprendizaje profundo

Guillermo Hernández¹, Angélica González Arrieta¹, Sara Rodríguez¹

¹Departamento de informática y automática,
Facultad de Ciencias,
Universidad de Salamanca,
Plaza de los Caídos s/n 37008 Salamanca
{guillehg,angelica,srg}@usal.es

Resumen La detección de duplicados de vídeos es un campo de investigación activo, motivado por la protección de la propiedad intelectual, la lucha contra la piratería o el rastreo de la procedencia de segmentos de vídeo reutilizados.

En este trabajo se propone e implementa un método para la detección de vídeos duplicados haciendo uso de métodos de aprendizaje profundo y de las técnicas propias del campo de la recuperación de información. Dicho método se ha evaluado con un conjunto de datos habitualmente empleado en el campo, con el que se han obtenido precisiones medias elevadas, por encima del 85%. Se ha estudiado el efecto de las diferentes capas de la red neuronal convolucional empleada por el algoritmo, los mecanismos de agregación que se pueden emplear sobre estas, y la influencia del modelo de recuperación, encontrándose un juego de parámetros que optimiza la precisión global del sistema.

Keywords: Detección de vídeos duplicados, aprendizaje profundo

1. Introducción

La detección de vídeos con contenido duplicado es un campo de investigación activo, motivado por necesidades como la protección de la propiedad intelectual, la lucha contra la piratería o el rastreo de la procedencia de segmentos de vídeo reutilizados.

La cantidad de vídeos disponibles en Internet ha experimentado un importante aumento en los últimos años, relacionado en gran medida con el desarrollo de las redes sociales. Podemos hacernos una idea de la magnitud que suponen a través del dato de que se suben unas 100 horas de contenido audiovisual a la plataforma YouTube cada minuto [12]. Asimismo, la aparición de aplicaciones que facilitan el proceso de descarga, modificación y subida de vídeos ha contribuido a aumentar la cantidad de contenido duplicado [44], en línea con los intereses de los usuarios. Se estima actualmente que el contenido duplicado constituye un 27% frente al original [71].

Una posible definición de vídeo «duplicado» —para evitar limitarse a los duplicados exactos—, es la siguiente, tomada de Wu *et al.* [72]: «vídeos aproximadamente iguales a un original, con posibles diferencias en el formato de archivo, ajustes de la codificación, variaciones de naturaleza fotométrica (color, luminosidad), operaciones de edición (inclusión de etiquetas, logos, bordes, ...), ajustes en la duración u otras modificaciones (añadido o borrado de fotogramas)». La idea intuitiva

que subyace bajo esta es que son pares de vídeos que un usuario reconocería como una modificación del original, pese a las diferencias que existan entre ellos.

1.1. Objetivos

En este trabajo se han propuesto los siguientes objetivos:

- Estudiar los modelos existentes para la detección de duplicados de vídeos mediante un proceso de mapeo sistemático.
- Proponer e implementar un algoritmo que haga uso de técnicas de aprendizaje profundo para la detección de duplicados de vídeos que se pueda utilizar en el marco del sistema anterior.
- Validar la capacidad del sistema propuesto con un conjunto de datos adecuado.

2. Estado del Arte

En este apartado se presenta una revisión sistemática del estado del arte mediante la técnica del mapeo sistemático [19, 32].

El mapeo sistemático es una técnica que permite la extracción de conocimiento científico existente en una temática específica, identificando necesidades y vacíos en ella mediante la categorización de las publicaciones existentes, obtenidas mediante un proceso lo más formal, definido y repetible. Se pueden identificar las siguientes etapas en dicho proceso:

- Formulación de preguntas de investigación que el mapeo aspira a responder. Normalmente son preguntas amplias sobre el contenido de la literatura especializada [32].
- Definición de un sistema de búsqueda, a partir de una serie de palabras clave con las que se construye una consulta y mediante un conjunto de base de datos bibliográficas.
- Definición de criterios de inclusión/exclusión para la consideración de los trabajos en la revisión y, posiblemente, la selección mediante indicadores de calidad.
- Se realiza la búsqueda bibliográfica definida.
- Se seleccionan los trabajos que cumplen con los criterios impuestos.
- Se extrae de cada trabajo la información orientada a responder a las preguntas antes planteadas.

Este desarrollo no es estrictamente lineal, sino que la realización del propio estudio pueden sugerir reformularlo para su mejora, adaptándose este de forma iterativa. Ejemplos de esto pueden ser la inclusión de nuevas preguntas, la extracción de nuevos datos o la inclusión de bibliografía adicional relevante citada en los trabajos que escapara a los criterios de selección.

2.1. Planificación

Preguntas de Investigación. Las preguntas que se han propuesto para guiar el mapeo sistemático son las siguientes:

1. ¿Qué técnicas se usan para la detección de duplicados de vídeos?
2. ¿Cómo han evolucionado en el tiempo las tendencias en dicho campo?
3. ¿Qué modelos han considerado la detección de subsecuencias copiadas?

Si bien la última de estas preguntas no se corresponde directamente con los objetivos recogidos en la presente memoria, se ha considerado su inclusión, sugerida por el contenido de los trabajos analizados, por ser información potencialmente útil para un estudio continuador.

Palabras Clave. Las palabras clave en inglés consideradas para la búsqueda fueron *video* y *copy detection*. Una primera búsqueda se utilizó para detectar posibles sinónimos usuales del segundo de los términos, pero este se reveló como exhaustivo a este respecto. Una expresión de búsqueda simple sería, por lo tanto, ("video") AND ("copy detection").

Base de Datos. La base de datos considerada fue Scopus¹. La razón de limitarse a una única base de datos es para permitir el desarrollo del estudio en un tiempo razonable.

Criterios de Inclusión/Exclusión. Los criterios de inclusión considerados son los siguientes:

1. El trabajo debe ser un artículo de revista.
2. El trabajo debe estar escrito en inglés.
3. El trabajo debe presentar un sistema para la detección de duplicados de vídeos.
4. El trabajo debe estar accesible con los recursos bibliográficos de la Universidad de Salamanca.

Para detallar las posibles causas de fallo de la tercera de estas consideraciones se han definido los siguientes criterios de exclusión para complementar a la negación de los anteriores:

1. El trabajo presenta un sistema exclusivamente para vídeo 3D.
2. El trabajo es una revisión bibliográfica.

Formulario de Extracción de Datos. El formulario de extracción de datos construido se muestra en la Fig. 1. Los campos escogidos están orientados a la respuesta de las preguntas planteadas anteriormente. Respecto a los valores ofrecidos para el campo de algoritmos, se iteró a lo largo del estudio de la bibliografía para extender el conjunto de valores válidos para dar lugar al mostrado.

Las dos últimas preguntas en el formulario han venido sugeridas por la casuística encontrada durante el periodo de revisión. En primer lugar, algunos de los trabajos utilizan la información auditiva de los vídeos en lugar de limitarse a los fotogramas, para lo que se ha recogido esta posibilidad, que aporta información en cierta medida sobre las técnicas empleadas, aunque no sea exactamente un algoritmo. Finalmente, algunos de los artículos hacen referencia específica a la detección de segmentos de vídeo que sean duplicados, lo que motivó la tercera de las preguntas de investigación de esta revisión, como se indicó anteriormente.

¹ <http://www.scopus.com>.

| Data Extraction Form | | | |
|----------------------|----------------------|-------------------|--|
| | Description | Type | Values |
| ⌵ ⌶ | Year | Integer Field | n/a |
| ⌵ ⌶ | Algorithms | Select Many Field | <ul style="list-style-type: none"> • Clustering • CNN • Feature extraction • Frame selection • Hashing • Nearest-neighbor • Probabilistic model |
| ⌵ ⌶ | Audio treatment | Boolean Field | n/a |
| ⌵ ⌶ | Subsequence matching | Boolean Field | n/a |

Figura 1. Formulario de extracción de datos en la herramienta Parsifal.

2.2. Resultados

Selección de Trabajos. La búsqueda anteriormente planteada se refinó mediante la expresión, con la sintaxis específica de Scopus,

```
TITLE-ABS-KEY("video") AND TITLE-ABS-KEY("copy detection")
AND SRCTYPE(J)
AND LANGUAGE("English")
```

En dicha expresión se ha exigido que los términos empleados en la búsqueda aparezcan en el título, el resumen o la lista de palabras clave del artículo, para evitar resultados cuyo tema principal no sea la detección de duplicados. Tras este proceso, se aplicaron los criterios de exclusión restantes. La evolución del número de artículos considerados a lo largo de estas etapas se refleja en la Fig. 2. Esta recuperación está actualizada en mayo de 2019.

Finalmente han resultado seleccionados 57 artículos en la revisión [2-11, 13, 15-18, 20-25, 27-31, 35-43, 45, 46, 48-54, 57-60, 64, 65, 67-70, 73-75]. Las causas de exclusión de los trabajos que superaron el filtro de búsqueda se detallan en la Tab. 1.

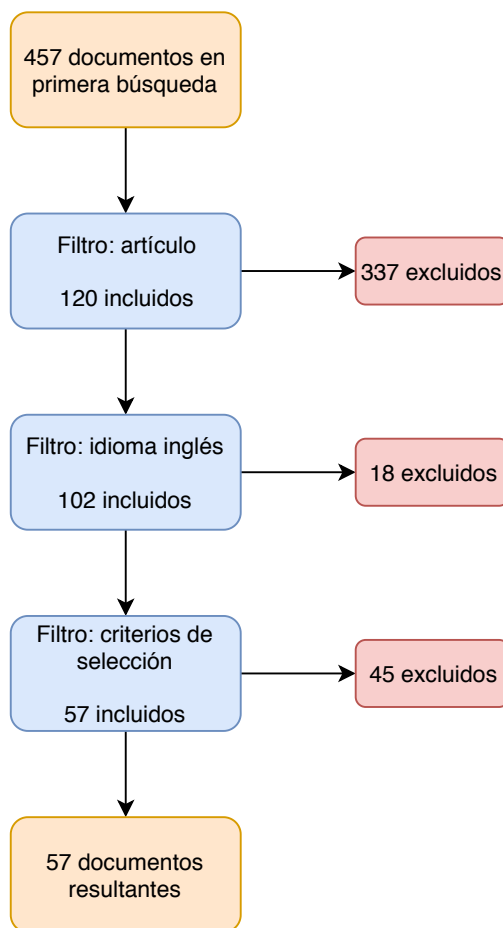


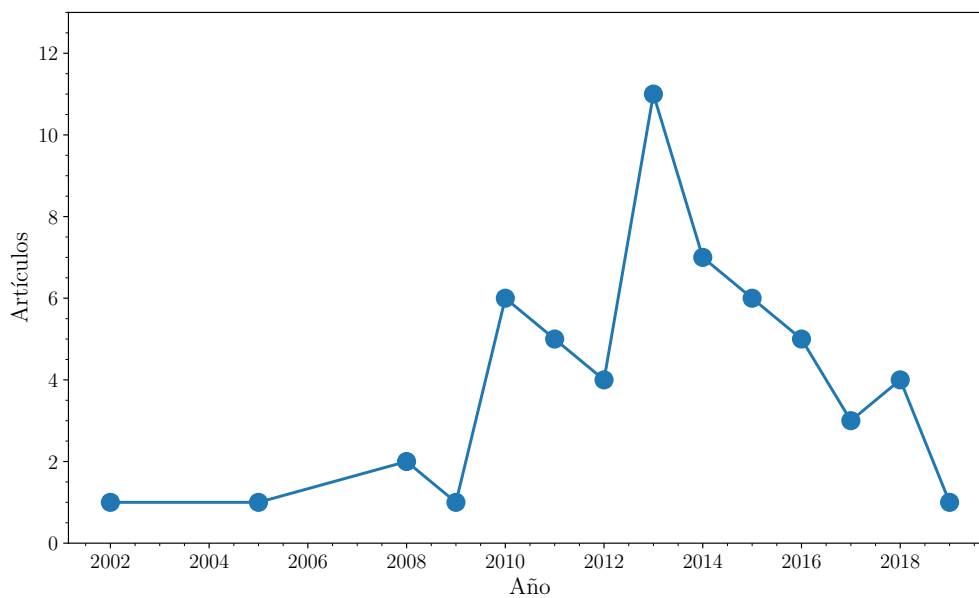
Figura 2. Flujo de selección de artículos en el mapeo sistemático.

Análisis de los datos La evolución de la cantidad de artículos por año se muestra en la Fig. 3. Se puede observar una tendencia con un máximo en torno al año 2013. Actualmente parece conservarse el interés en el campo, con una evolución hacia el declive.

El desglose por algoritmos empleados se puede estudiar en la Fig. 4. Se observa que la mayoría de artículos han encajado en la categoría de «extracción de características», con lo que queda patente que esta es demasiado generalista como para detallar información. Volveremos a esta cuestión en el apartado de conclusiones. Respecto a los algoritmos, según la selección considerada las técnicas más actuales son las redes convolucionales [28, 74], el *clustering* (que se emplea para acelerar las búsquedas) [36, 40, 43] y la inclusión de métodos específicos para la selección de fotogramas [3, 22, 28, 64].

Tabla 1. Causas de exclusión de los trabajos descartados.

| Causa | Ocurrencias |
|---|-------------|
| El trabajo no presenta de forma directa un algoritmo de detección de duplicados | 32 |
| El texto completo no está accesible con las herramientas disponibles | 9 |
| El trabajo es una revisión bibliográfica | 2 |
| El trabajo trata exclusivamente vídeo 3D | 2 |

**Figura 3.** Cantidad de artículos aceptados por año de publicación. Los datos del año 2019 se refieren a la fecha actual.

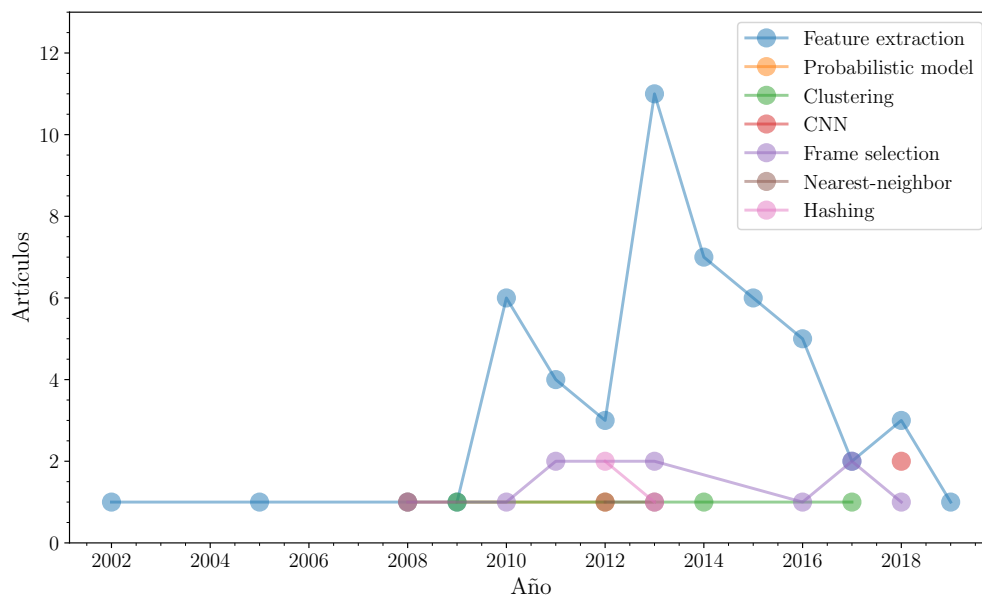


Figura 4. Algoritmos empleados en los trabajos seleccionados por año de publicación.

La cantidad de trabajos que toman en consideración el audio para la extracción de características se representa como la primera de las series en la Fig. 5, mientras que la cantidad de artículos que analizan la detección de segmentos de vídeo se detalla como la segunda. Como se puede observar, ambas cuestiones tienen un interés relativamente reciente.

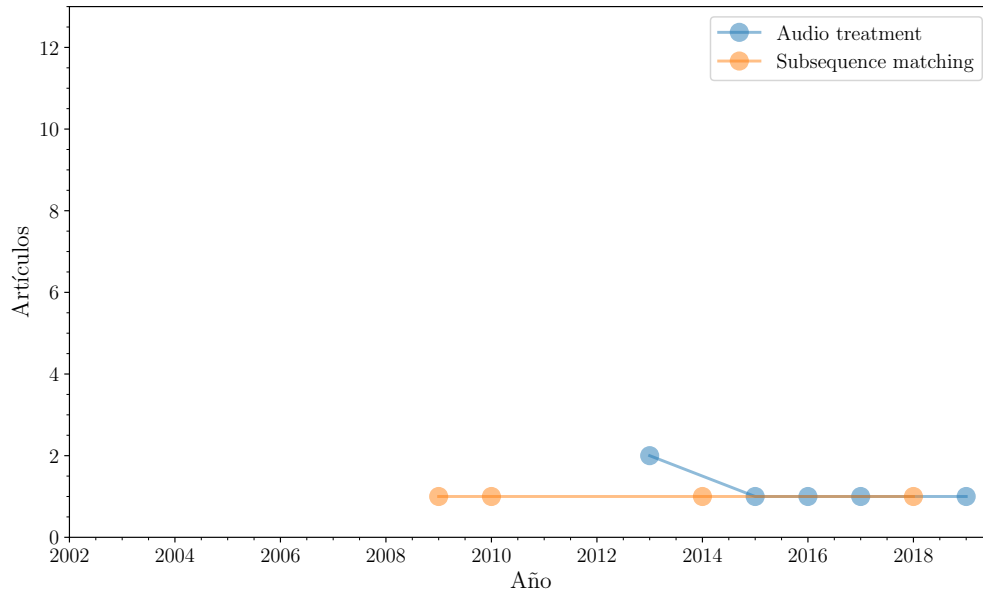


Figura 5. Información adicional empleada en los trabajos seleccionados por año de publicación. Los ejes mantienen la escala de las Figs. 3 y 4 para facilitar la comparación.

Datos completos La lista de referencias completa se recoge con sus datos, por completitud, en la Tab. 2.

Tabla 2: Información de los artículos en el mapeo sistemático de la literatura.

| Ref. | Año | Algoritmos | Audio | Subsecuencias |
|------|------|---|-------|---------------|
| [60] | 2015 | Feature extraction | No | No |
| [58] | 2016 | Feature extraction | No | No |
| [38] | 2012 | Feature extraction, Probabilistic model | No | No |
| [36] | 2009 | Clustering, Feature extraction | No | Sí |
| [9] | 2010 | Feature extraction | No | Sí |
| [69] | 2015 | Feature extraction | Sí | No |
| [53] | 2013 | Feature extraction | No | No |
| [30] | 2005 | Feature extraction | No | No |
| [49] | 2012 | Feature extraction | No | No |

| | | | |
|------|---|----|----|
| [8] | 2014 Feature extraction | No | Sí |
| [28] | 2018 CNN, Frame selection | No | No |
| [70] | 2011 Feature extraction | No | No |
| [39] | 2016 Feature extraction | Sí | No |
| [67] | 2014 Feature extraction | No | No |
| [50] | 2013 Feature extraction | Sí | No |
| [13] | 2014 Feature extraction | No | No |
| [2] | 2013 Feature extraction, Nearest-neighbor | No | No |
| [24] | 2015 Feature extraction | No | No |
| [48] | 2011 Feature extraction | No | No |
| [29] | 2014 Feature extraction | No | No |
| [43] | 2017 Clustering | No | No |
| [15] | 2010 Feature extraction, Frame selection | No | No |
| [40] | 2014 Clustering, Feature extraction | No | No |
| [42] | 2013 Feature extraction, Frame selection | No | No |
| [65] | 2013 Feature extraction | No | No |
| [73] | 2012 Feature extraction, Hashing | No | No |
| [17] | 2011 Feature extraction | No | No |
| [59] | 2013 Feature extraction | Sí | No |
| [27] | 2016 Feature extraction | No | No |
| [18] | 2012 Hashing, Nearest-neighbor | No | No |
| [54] | 2011 Feature extraction, Frame selection | No | No |
| [51] | 2015 Feature extraction | No | No |
| [25] | 2018 Feature extraction | No | No |
| [37] | 2010 Feature extraction | No | No |
| [4] | 2013 Feature extraction | No | No |
| [64] | 2017 Feature extraction, Frame selection | No | No |
| [7] | 2013 Feature extraction, Hashing | No | No |
| [16] | 2013 Feature extraction | No | No |
| [52] | 2013 Feature extraction, Frame selection | No | No |
| [46] | 2015 Feature extraction | No | No |
| [22] | 2017 Feature extraction, Frame selection | Sí | No |
| [3] | 2016 Feature extraction, Frame selection | No | No |
| [57] | 2010 Feature extraction | No | No |
| [75] | 2011 Frame selection | No | No |
| [35] | 2010 Feature extraction | No | No |
| [10] | 2010 Feature extraction | No | No |
| [11] | 2015 Feature extraction | No | No |
| [6] | 2008 Frame selection, Probabilistic model | No | No |
| [5] | 2008 Feature extraction | No | No |
| [23] | 2002 Feature extraction | No | No |
| [68] | 2013 Feature extraction | No | No |
| [31] | 2014 Feature extraction | No | No |
| [41] | 2014 Feature extraction | No | No |
| [45] | 2018 Feature extraction | No | No |
| [74] | 2018 CNN, Feature extraction | No | Sí |
| [20] | 2019 Feature extraction | Sí | No |
| [21] | 2016 Feature extraction | No | No |

2.3. Análisis

Respecto a la primera pregunta de investigación de qué técnicas servirían para realizar la detección de duplicados, la exploración de la literatura ha permitido encontrar que la extracción de características —de fotogramas, secuencias temporales de estos y audio— son, de forma generalista, lo medios empleados. Formalmente se sirven de herramientas matemáticas variadas —con una casuística demasiado compleja como para que resulte complicado establecer una clasificación de éstas—, así como de algoritmos entre los que predominan las técnicas probabilísticas (frecuentemente basadas en grafos) y las redes neuronales convolucionales.

De la cantidad de publicaciones en función de las técnicas que emplean se desprende que las más actuales son las redes convolucionales [28, 74], la inclusión de algoritmos de *clustering* como medio para acelerar las búsquedas [36, 40, 43] y la consideración de métodos específicos para la selección de fotogramas [3, 22, 28, 64].

Algunos trabajos han estudiado específicamente la cuestión de detección de segmentos copiados. Los trabajos detectados así han sido las Refs. [8, 9, 36, 74].

El estudio de revisión aquí realizado se ha acotado por exigencia tiempo disponible para su desarrollo. Las siguientes preguntas, sugeridas por el propio estudio, podrían servir para mejorar este primer mapeo sistemático:

- ¿Qué trabajos emplean un conjunto de datos sintético y cuáles uno real?
- ¿Qué bases de datos se han usado como partida?
- ¿Qué trabajos recaban información temporal y cuáles no?

También sería conveniente extender los criterios de selección con otra base de datos o con la relajación de los criterios para permitir actas de congreso a fin de ver mejor reflejada la evolución cronológica de las tendencias.

3. Sistema Propuesto

3.1. Sistema de Detección

El proceso general, basado en los flujos de trabajo de las Refs. [28, 33, 74], es el siguiente:

1. Se extraen los **fotogramas clave** (*keyframes*) del vídeo.
2. Se utilizan los fotogramas como entrada a una red neuronal convolucional. Las funciones de activación de sus capas intermedias se recogen mediante la técnica de *max pooling*, permitiendo así la **extracción de características** del fotograma.
3. Las características se tratan con el modelo de bolsa de palabras, comparándose con un **libro de códigos** para obtener la «palabra válida» (*code word*) más similar que se encuentre en este.
4. La lista de ocurrencias de cada característica del libro de códigos se convierte a un **modelo de vectores con peso**.
5. Se obtiene mediante una **medida de similitud** la lista ordenada de vídeos más similares.

En los siguientes subapartados se analizan en más detalle las diversas componentes del proceso. Adicionalmente, en §3.1 se presentan unos métodos de agregación que pueden servir para mejorar los resultados del esquema.

Extracción de fotogramas clave Un vídeo puede descomponerse de forma jerárquica en escenas —unidades narrativas con continuidad espacial o temporal—, planos —perspectivas de los elementos que capta el observador— y fotogramas —cada imagen que compone la animación del vídeo.

Los fotogramas clave se pueden definir como el subconjunto de fotogramas que puede representar de forma abreviada la componente visual de un vídeo [55], para los que existen varios métodos de extracción automática [56].

En el conjunto de datos empleado para la validación de este trabajo ya se proporciona un conjunto de fotogramas clave extraídos (*cf.* más adelante §3.2), por lo que las técnicas de extracción de fotogramas clave no se estudiarán en detalle.

Alternativamente, se podrían muestrear con otras técnicas fotogramas del vídeo, aunque esta posibilidad queda fuera del alcance de este trabajo.

Extracción de características La red neuronal empleada ha sido la denominada AlexNet [34], una red convolucional profunda entrenada con el conjunto de datos ImageNet [14], sobre cuyo problema de clasificación obtuvo resultados que mejoraban significativamente el estado del arte en el año 2012. Entre los puntos clave que permitieron su éxito se destacan el uso de la función lineal rectificadora $x : \max(0, x)$ frente a las alternativas basadas en la \tanh o en funciones sigmoidales, que son saturantes; el uso de GPU para aplicar de forma rápida las operaciones de convolución, distribuidas en dos tarjetas diferentes, y el uso de la técnica de *dropout* para evitar el sobreentrenamiento [26].

La Fig. 6 muestra de forma resumida el proceso de extracción de características que tiene lugar en las capas internas de la red, omitiendo pasos intermedios y sin detallar algunas cuestiones sobre la naturaleza de las operaciones. En particular se omiten unas últimas capas densas que estarían destinadas a resolver el problema de clasificación de ImageNet, y que no son útiles para la presente aplicación, motivo por el cual se han omitido. Para una descripción completa de la red puede consultarse la Ref. [34].

Los pesos e implementación en TensorFlow de la red se han tomado de la réplica preparada por E. Shelhamer, que se proporciona bajo una licencia sin restricciones a su uso².

Libro de códigos El libro de códigos, que transforma cada característica en una palabra válida dentro de su vocabulario, se puede construir mediante un algoritmo de *clustering*, en el que los centroides constituirían las palabras válidas.

Con vistas a la construcción progresiva del *cluster* en un sistema real, así como para permitir el entrenamiento con una gran cantidad de datos, es conveniente el uso de algoritmos por lotes. En este trabajo se ha considerado empleado el denominado «mini batch k -means» [62].

Modelo de vectores con peso Los vectores con la cuenta del número de ocurrencias de las palabras válidas se deben transformar en una representación que tenga en cuenta adecuadamente el efecto de su presencia tanto a nivel de vídeo como al nivel de la colección. La herramienta usual para ello es el denominado *tf-idf* (*Term Frequency–Inverse Document Frequency*), para el que existen varios esquemas disponibles [1, 61].

De forma general, la representación será un vector cuyas componentes vendrán dadas por el producto de dos funciones de los términos t , $\text{tf}(t)$ e $\text{idf}(t)$, sobre el que después se aplicará posiblemente

² https://github.com/BVLC/caffe/tree/master/models/bvlc_alexnet.

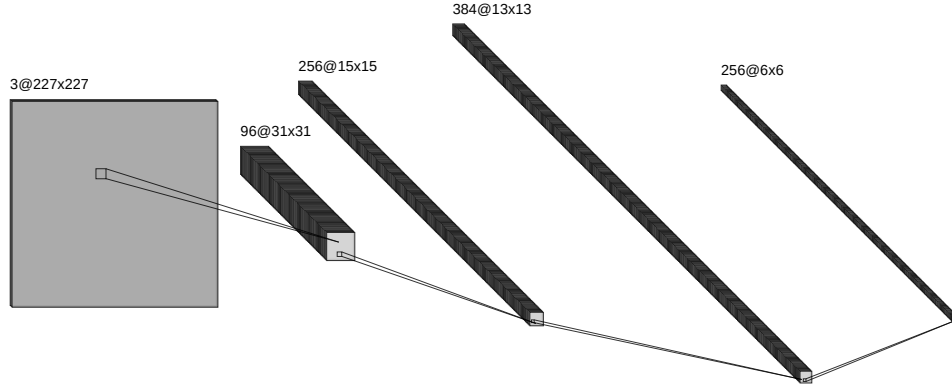


Figura 6. Esquema simplificado de las capas convolucionales de AlexNet. La notación $c@w \times h$ indica c canales de tamaño (w, h) . Los cuadrados representados contienen los mapas de características. Cada transición entre capas supone operaciones de convolución y de *max-pooling* que aquí no se detallan. Tras la última capa habría dos capas densas de neuronas que resolverían el problema de clasificación de ImageNet a partir de las características de alto nivel y que aquí no se representan. Para una descripción completa de la red puede consultarse la Ref. [34].

algún esquema de normalización, lo que se puede expresar como

$$\|(\text{tf}(t_1) \cdot \text{idf}(t_1), \dots, \text{tf}(t_n) \cdot \text{idf}(t_n))\|_{\text{norm}}. \quad (1)$$

De este modo, las tres funciones tf , idf y $\|\cdot\|_{\text{norm}}$ definen completamente el esquema empleado.

El modelo de recuperación considerado como base para este trabajo se ha escogido haciendo que la frecuencia de términos tf se corresponde directamente con la cantidad de ocurrencias del término, mientras que la frecuencia inversa de documento se ha modelado de forma logarítmica suavizada, dado por

$$\text{idf}(t) = \log \frac{1+n}{1+\text{df}(t)} + 1, \quad (2)$$

donde n es el número de documentos en el conjunto de recuperación y $\text{df}(t)$ es la cantidad de documentos con la palabra t . A la aparición de los dos sumandos 1 en la fracción de (2) se la denomina ocasionalmente suavizado, y obedece a garantizar que la función esté definida cuando el término no figure en la colección documental³. Finalmente, se ha normalizado cada vector en el sentido de la norma euclídea

$$\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}. \quad (3)$$

En la sección de resultados (§4) se analizará la capacidad de recuperación del sistema en función del modelo empleado, para lo que se considerarán las variaciones recogidas en la Tab. 3.

³ Si bien en nuestro caso esto no es posible debido al uso del libro de códigos.

Tabla 3. Variaciones sobre el modelo de recuperación. $f(t)$ denota la cantidad apariciones del término t , n la cantidad de documentos y $df(t)$ la cantidad de documentos con el término t .

| | tf | idf | normalización |
|----------------|------------------|--------------------------------|---------------|
| Estándar | $f(t)$ | $\log \frac{1+n}{1+df(t)} + 1$ | $\ \cdot\ _2$ |
| Sin idf | $f(t)$ | 1 | $\ \cdot\ _2$ |
| Sin suavizado | $f(t)$ | $\log \frac{n}{df(t)} + 1$ | $\ \cdot\ _2$ |
| Tf logarítmica | $1 + \log(f(t))$ | $\log \frac{1+n}{1+df(t)} + 1$ | $\ \cdot\ _2$ |

Medida de similitud En la última etapa del proceso, los vectores de pesos se comparan entre ellos para encontrar los más similares a uno dado. Existen varias nociones que se pueden aplicar [1], en este trabajo consideraremos la similitud coseno, definida como

$$\text{sim}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}, \quad (4)$$

Recordemos que, con el modelo normalizado como se ha descrito en §3.1, el denominador es igual a la unidad y la similitud coseno coincide con el producto escalar.

Mecanismos de agregación En el trabajo de Kordopatis-Zilos *et al.* se presentan dos propuestas que pueden servir para utilizar de forma combinada la información extraída de varias capas de la red.

La primera de ellas es la denominada «agregación por capas», en que se construye un libro de códigos para cada capa siguiendo el método propuesto. A continuación, el modelo vectorial utilizará las ocurrencias de cada palabra de los diferentes libros, dando lugar a un vocabulario cuyo tamaño se incrementa en un factor dado por el número de capas empleadas en la agregación. Este método se puede entender como una agregación realizada en el paso 3.

La alternativa es la denominada «agregación vectorial», en que los vectores de características extraídos en cada capa se concatenan, utilizando el vector resultante para generar un único libro de códigos. En este caso la agregación se puede entender como realizada en el paso 2.

3.2. Conjunto de Datos

El conjunto de datos considerado para la validación del sistema ha sido el denominado CC_WEB_VIDEO [71, 72]. Este conjunto de vídeos se recopiló en noviembre de 2006, a partir de una serie de consultas textuales destinadas a recuperar una serie de vídeos populares en ese momento. Para estas consultas se marcó como «original» el vídeo más frecuente, etiquetándose la relación de los demás con este con unas pocas categorías de duplicados (Duplicado exacto, vídeo similar, versión larga, ...).

El mismo conjunto de datos provee de una serie de fotogramas para cada vídeo, extraído empleando un algoritmo de detección de cortes. En total dispone de 398015 fotogramas clave, que son los empleados como entrada para el análisis presentado en §4.

4. Resultados

4.1. Evaluación del Sistema

El comportamiento de la recuperación de duplicados para el vídeo de referencia de cada una de las consultas del conjunto de datos puede resumirse a través de un diagrama precisión-exhaustividad, como los que se muestran en la Fig. 7. Conforme aumenta la cantidad de elementos recuperados como potenciales duplicados por el sistema, la exhaustividad (fracción de los auténticos duplicados respecto al total de vídeos) aumenta o permanece constante, lo que se refleja en la curva que avanza sólo hacia la derecha. Solo en caso de que los vídeos recuperados como duplicados realmente sean tales, la precisión (fracción de auténticos duplicados respecto a los vídeos recuperados) será elevada. Se han escogido tres casos característicos para explicar el comportamiento de la red.

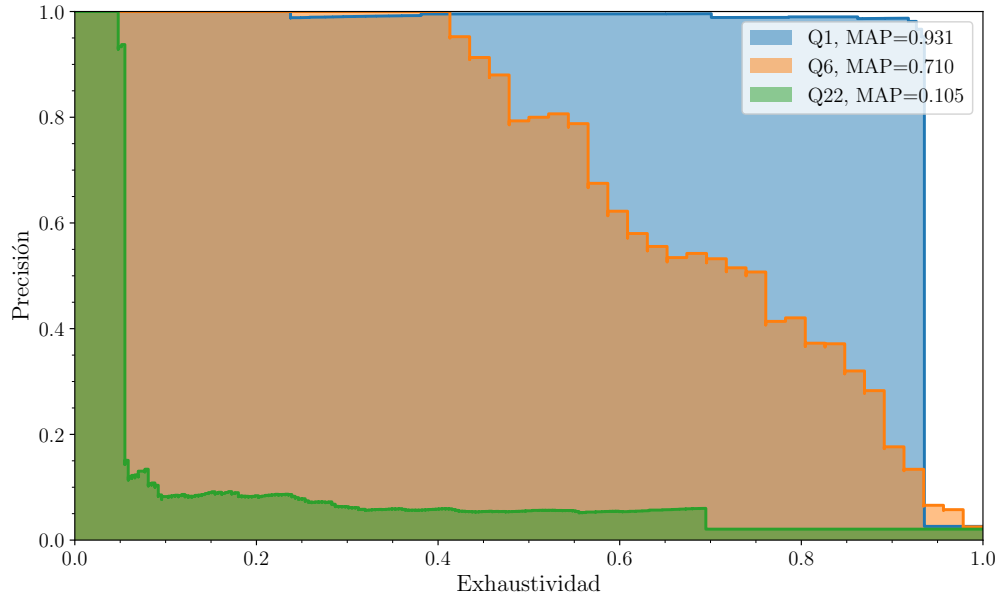


Figura 7. Diagrama precisión-exhaustividad para consultas con una precisión media elevada (Q1), media (Q6) y muy baja (Q22).

Las curvas presentan siempre un pedestal a la izquierda que se corresponde con los duplicados exactos. A continuación, si la recuperación ha sido buena la curva descenderá lentamente, mientras que la caída será más rápida en otro caso.

Para resumir con un único valor las curvas recurriremos a la precisión media, que se define como

$$AP = \sum_n (R_n - R_{n-1})P_n, \quad (5)$$

donde el índice n se refiere a cada punto de la curva de precisión-exhaustividad, cuyas coordenadas respectivas son P_n, R_n . Este valor es equivalente a una aproximación numérica del área bajo dicha curva.

El comportamiento del sistema al emplear las características de una u otra capa de la red neuronal se muestra en la Fig. 8. Como se puede observar, la tendencia general es que las capas más profundas identifican con más precisión los duplicados.

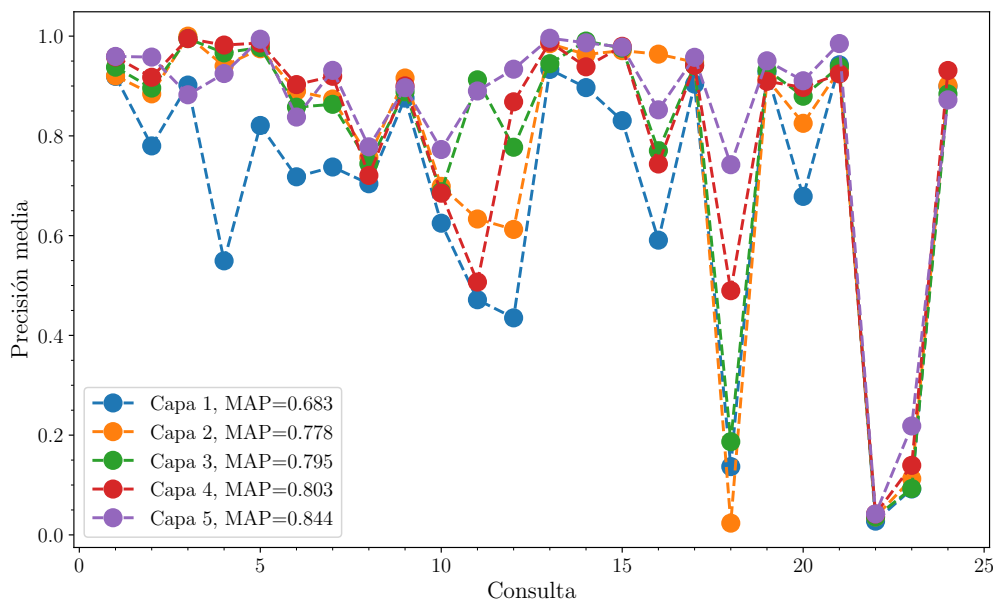


Figura 8. Precisión media (vertical) para cada consulta (horizontal). Cada serie muestra el empleo de las características de una profundidad diferente de la red (leyenda). Las líneas discontinuas sirven meramente como ayuda visual. Las consultas Q18 y Q22 están reconocidas con los casos más complicados del conjunto de datos (*cf.* Ref. [72]).

Debe notarse que esta evaluación se ha llevado a cabo frente al total de vídeos de la colección, y no sólo frente a aquellos pertinentes a cada consulta. Una evaluación con este último sistema daría lugar a resultados más optimistas. Se ha escogido la primera de las formas para reflejar mejor un escenario de uso realista.

4.2. Modelos de Recuperación

Para estudiar la dependencia con el modelo de recuperación empleado (*cf.* §3.1) se ensayó el proceso de recuperación de duplicados con las configuraciones resumidas en la Tab. 3. Los resultados se recogen de forma general en la Fig. 9.

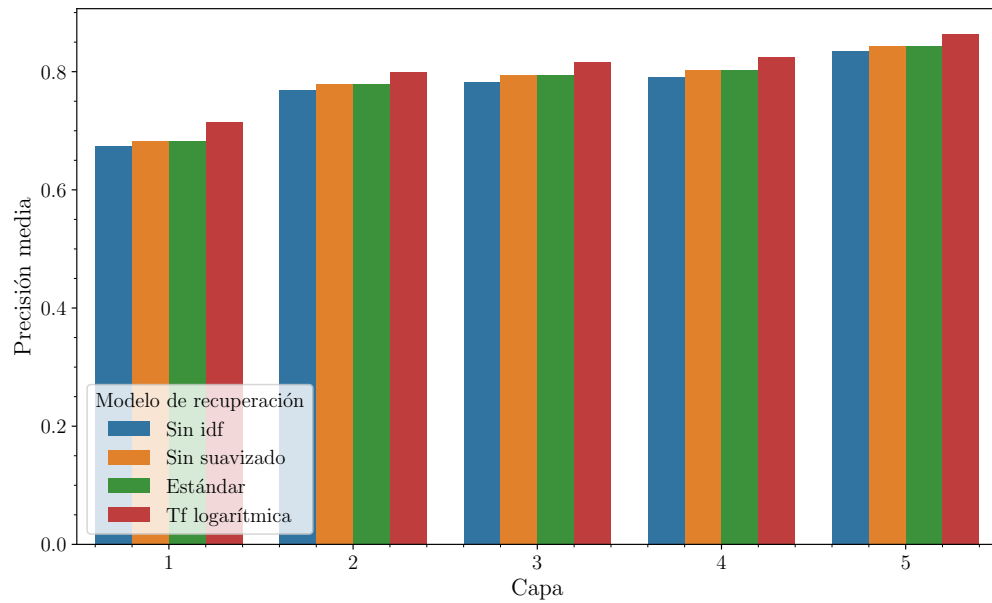


Figura 9. Precisión media (vertical) para cada capa de la red (eje horizontal). Cada serie (leyenda) muestra un modelo de recuperación diferente, cuyas descripciones formales se pueden consultar en la Tab. 3.

Como se puede observar, la inclusión del término de idf resulta beneficiosa para la recuperación, no habiendo una diferencia significativa en la consideración del suavizado (2). Por otro lado, el uso de la forma logarítmica para la tf mejora los resultados del modelo considerado.

4.3. Métodos de agregación

Los dos mecanismos de agregación presentados en §3.1 se ensayaron, produciendo los resultados que se recogen en la Fig. 10. Como se puede observar en esta, la agregación por capas proporciona una ligera mejora al sistema, mientras que la agregación vectorial empeora los resultados.

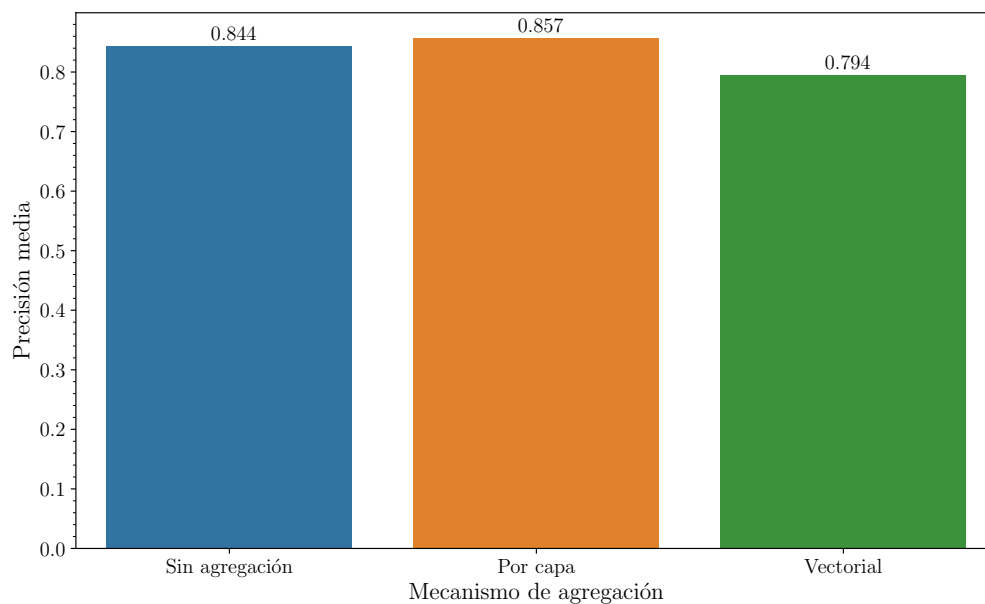


Figura 10. Precisión media (vertical) para cada método de agregación (eje horizontal). El valor se muestra en la etiqueta de cada barra.

Estas observaciones son coherentes con los resultados originales de la Ref. [33], donde la agregación por capas proporciona una ligera mejora en el caso de AlexNet y el vectorial resulta tener un peor desempeño que las mejores capas de forma individual.

4.4. Magnitud de las similitudes encontradas

La distribución de la medida de similitud para los diferentes tipos de duplicados que el etiquetado del conjunto de datos proporciona se muestra mediante el diagrama de cajas de la Fig. 11. Como muestra la línea de las medianas (mitad de la caja), las copias marcadas como simplemente como «similares» son las que presentan unos valores más elevados del coeficiente, mientras que los casos más complicados exhiben valores más reducidos en general, como cabía esperar. Existe algún caso en que la versión larga sí guarda un mayor parecido con el original, como muestran los valores atípicos, pero en realidad esto se debe a que las copias etiquetadas como versión «larga» suponen modificaciones muy diferentes. En el caso de la consulta 24, realmente es un vídeo de mayor duración con el mismo contenido, de ahí los valores con alto grado de similitud. Por otro lado, los de la consulta

1 incluyen tomas diferentes, similares a las originales pero recortadas o rotadas, que necesariamente tendrán menores coeficientes de similitud. Respecto a las «alteraciones considerables», en ocasiones resultan más similares en el sentido de la medida empleada (como muestran los numerosos valores atípicos). Como conclusión, pese a las diferentes etiquetas presentes en el conjunto de datos la distinción entre ellas no ayuda a mejorar mucho el análisis.

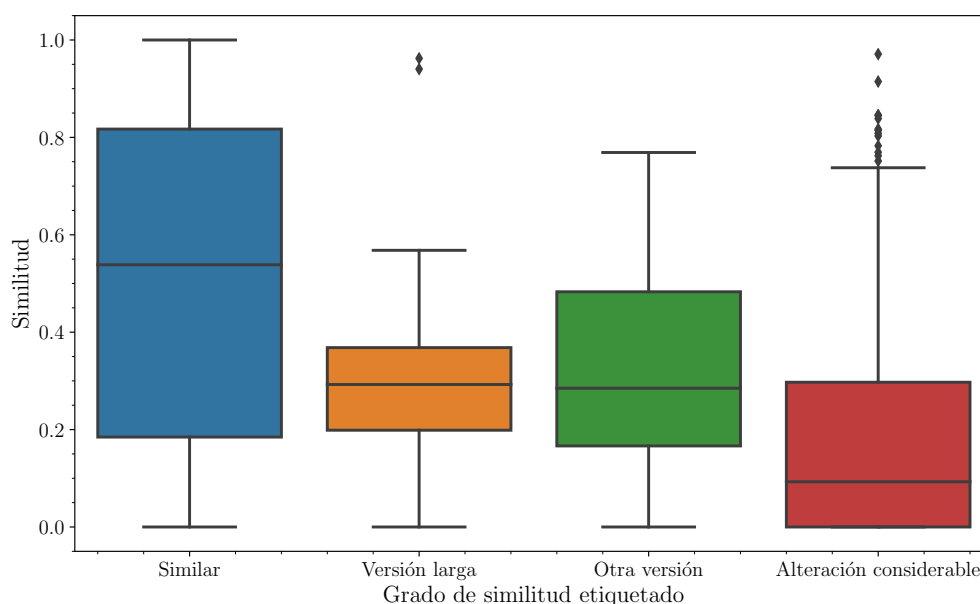


Figura 11. Distribución de medida de similitud de los vídeos en el conjunto de datos con el etiquetado de estos. Cada caja acota el primer y el tercer cuartil de la distribución, con la mediana marcada con la línea horizontal. Los bigotes (líneas semejantes a las barras de error) delimitan el 150 % del rango intercuartílico —sin exceder los máximos y mínimos de los datos—. Las observaciones fuera de dicho rango («valores atípicos») se muestran con un marcador.

Por otro lado, debe tenerse en cuenta que la magnitud depende mucho del vídeo considerado, como muestra la Fig. 12, empleando el mismo tipo de diagrama de cajas antes descrito. Esto hace difícil definir un umbral universal por el que puedan separarse duplicados de originales.

Para hacer esta afirmación un poco más rigurosa se puede recurrir a una comprobación estadística dada por la prueba de Kolmogórov-Smirnov [47]. Dicha prueba permite, de forma somera, estudiar si los datos procedentes de dos muestras diferentes pueden provenir de una misma distribución estadística. El resultado del test es un p -valor, un número entre 0 y 1 que indica la probabilidad de que los datos observados sean compatibles con la denominada hipótesis nula, que es la que se desea demostrar que es falsa. En este caso dicha hipótesis es que «la distribución del par de variables aleatorias es la misma».

Apenas hay unos pocos pares de consultas en que la prueba no sirva para concluir que las distribuciones sean diferentes. (Q6, Q8) es el caso más representativo, con p -valor 0.68. Los casos

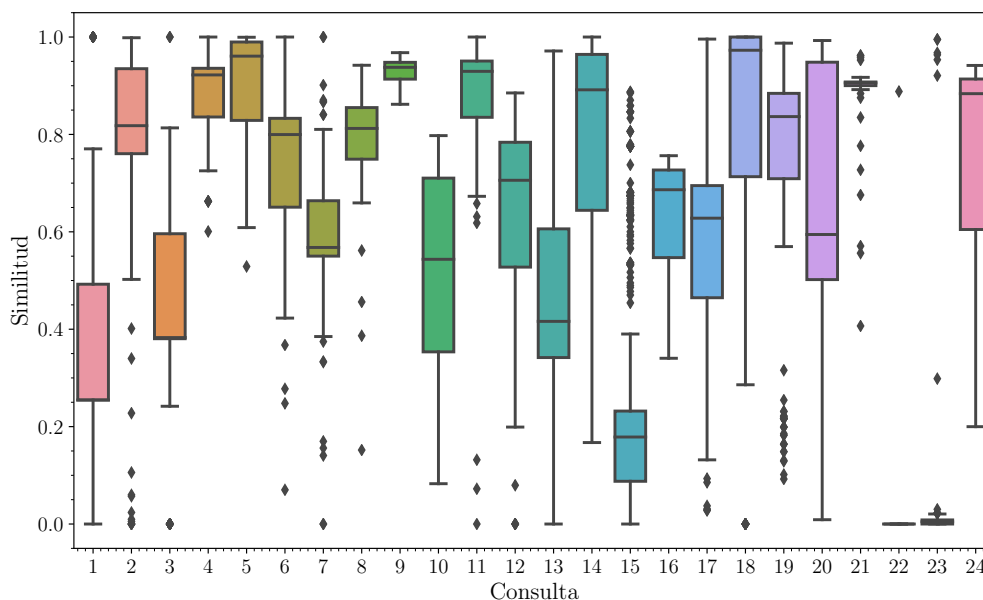


Figura 12. Distribución de medida de similitud de los vídeos marcados como copias similares en el conjunto de datos. Para la interpretación del diagrama de cajas ver el pie de la Fig. 11.

que involucran a Q24 presentan también p -valores elevados, pero cabe sospechar del número reducido de casos como causa de que la prueba no sea concluyente. En todo caso, debe recordarse que el resultado de la prueba no permite afirmar que las distribuciones sean diferentes, pero esto no implica que afirme que sean iguales.

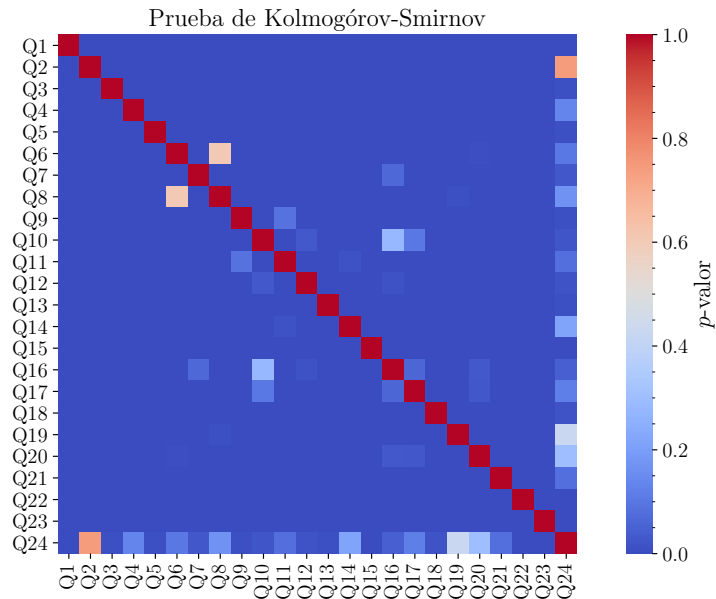


Figura 13. p -valores de la prueba de Kolmogórov-Smirnov para los pares de consultas. Los valores cercanos a cero (azul oscuro) indica que las distribuciones son significativamente diferentes, mientras que fuera de este rango no se alcanza dicha conclusión (lo que no demuestra que sean iguales).

5. Conclusiones y Trabajo Futuro

5.1. Conclusiones

De forma resumida, se han realizado las siguientes contribuciones en este trabajo:

- Se ha construido un sistema de recuperación de duplicados empleando técnicas de aprendizaje profundo.
- Se ha analizado el efecto de las diferentes capas de la red neuronal AlexNet, encontrando una mejora con la profundidad.
- Se han estudiado los mecanismos de agregación de capas propuestos en la Ref. [33]. La agregación por capas proporciona una ligera mejora de los resultados, mientras que la vectorial produce peores resultados que los individuales de la capa más profunda. Estos resultados son coherentes con los mostrados en dicho trabajo.
- Se han estudiado varios modelos vectoriales para la descripción de los términos del libro de códigos en los vídeos. El más ventajoso ha resultado ser el logarítmico, tanto para tf como para idf.
- Se ha estudiado la distribución de medidas de similitud en los duplicados del conjunto de datos. Las distribuciones muestran la dificultad de definir de forma universal un corte para la detección de vídeos duplicados. Se ha comprobado estadísticamente que las distribuciones son diferentes en cada consulta del conjunto de datos.

5.2. Trabajo futuro

A continuación, se recogen algunas ideas que pueden servir para extender este trabajo en el futuro:

- Revisar y extender la revisión de la literatura. En §2.3 se han identificado posibilidades para ello, a partir del propio estudio.
- Estudiar la influencia de los métodos de redimensionado de los fotogramas para su introducción en la red neuronal.
- Uso de otras redes neuronales preentrenadas, como pueden ser VGGNet [63] o GoogLeNet [66].

Referencias

- [1] R. Baeza-Yates y B. Ribeiro-Neto. *Modern information retrieval*. Addison-Wesley, 1999.
- [2] J. Barrios y B. Bustos. «Competitive content-based video copy detection using global descriptors». En: *Multimedia Tools and Applications* 62.1 (2013), págs. 75-110. DOI: 10.1007/s11042-011-0915-x.
- [3] A. Boukhari y A. Serir. «Weber Binarized Statistical Image Features (WBSIF) based video copy detection». En: *Journal of Visual Communication and Image Representation* 34 (2016), págs. 50-64. DOI: 10.1016/j.jvcir.2015.10.015.
- [4] D.-Y. Chen e Y.-M. Chiu. «Visual attention guided video copy detection based on feature points matching with geometric-constraint measurement». En: *Journal of Visual Communication and Image Representation* 24.5 (2013), págs. 544-551. DOI: 10.1016/j.jvcir.2013.04.005.
- [5] L. Chen y F. Stentiford. «Video sequence matching based on temporal ordinal measurement». En: *Pattern Recognition Letters* 29.13 (2008), págs. 1824-1831. DOI: 10.1016/j.patrec.2008.05.015.
- [6] C.-Y. Chiu, C.-S. Chen y L.-F. Chien. «A framework for handling spatiotemporal variations in video copy detection». En: *IEEE Transactions on Circuits and Systems for Video Technology* 18.3 (2008), págs. 412-417. DOI: 10.1109/TCSVT.2008.918447.
- [7] C.-Y. Chiu, T.-H. Tsai y C.-Y. Hsieh. «Efficient video segment matching for detecting temporal-based video copies». En: *Neurocomputing* 105 (2013), págs. 70-80. DOI: 10.1016/j.neucom.2012.04.036.
- [8] C.-Y. Chiu, T.-H. Tsai, Y.-C. Liou, G.-W. Han y H.-S. Chang. «Near-duplicate subsequence matching between the continuous stream and large video dataset». En: *IEEE Transactions on Multimedia* 16.7 (2014), págs. 1952-1962. DOI: 10.1109/TMM.2014.2342668.
- [9] C.-Y. Chiu y H.-M. Wang. «Time-series linear search for video copies based on compact signature manipulation and containment relation modeling». En: *IEEE Transactions on Circuits and Systems for Video Technology* 20.11 (2010), págs. 1603-1613. DOI: 10.1109/TCSVT.2010.2087471.
- [10] C.-Y. Chiu, H.-M. Wang y C.-S. Chen. «Fast min-hashing indexing and robust spatio-temporal matching for detecting video copies». En: *ACM Transactions on Multimedia Computing, Communications and Applications* 6.2 (2010). DOI: 10.1145/1671962.1671966.
- [11] C.-L. Chou, H.-T. Chen y S.-Y. Lee. «Pattern-based near-duplicate video retrieval and localization on web-scale videos». En: *IEEE Transactions on Multimedia* 17.3 (2015), págs. 382-395. DOI: 10.1109/TMM.2015.2391674.

- [12] C.-L. Chou, H.-T. Chen y S.-Y. Lee. «Pattern-based near-duplicate video retrieval and localization on web-scale videos». En: *IEEE Transactions on Multimedia* 17.3 (2015), págs. 382-395.
- [13] O. Cirakman, B. Günsel, N. Sengor y S. Kutluk. «Content-based copy detection by a subspace learning based video fingerprinting scheme». En: *Multimedia Tools and Applications* 71.3 (2014), págs. 1381-1409. DOI: 10.1007/s11042-012-1269-8.
- [14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li y L. Fei-Fei. «ImageNet: A large-scale hierarchical image database». En: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, págs. 248-255.
- [15] M. Douze, H. Jégou y C. Schmid. «An image-based approach to video copy detection with spatio-temporal post-filtering». En: *IEEE Transactions on Multimedia* 12.4 (2010), págs. 257-266. DOI: 10.1109/TMM.2010.2046265.
- [16] D. Dutta, S. Saha y B. Chanda. «An attack invariant scheme for content-based video copy detection». En: *Signal, Image and Video Processing* 7.4 (2013), págs. 665-677. DOI: 10.1007/s11760-013-0482-x.
- [17] M. Esmaili, M. Fatourehchi y R. Ward. «A robust and fast video copy detection system using content-based fingerprinting». En: *IEEE Transactions on Information Forensics and Security* 6.1 (2011), págs. 213-226. DOI: 10.1109/TIFS.2010.2097593.
- [18] M. Esmaili, R. Ward y M. Fatourehchi. «A fast approximate nearest neighbor search algorithm in the hamming space». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.12 (2012), págs. 2481-2488. DOI: 10.1109/TPAMI.2012.170.
- [19] F. Garca-Peñalvo. *Revisiones y mapeos sistemáticos de literatura*. 2019.
- [20] Z. Guzman-Zavaleta y C. Feregrino-Uribe. «Partial-copy detection of non-simulated videos using learning at decision level». En: *Multimedia Tools and Applications* 78.2 (2019), págs. 2427-2446. DOI: 10.1007/s11042-018-6345-2.
- [21] Z. Guzman-Zavaleta y C. Feregrino-Uribe. «Towards a video passive content fingerprinting method for partial-copy detection robust against non-simulated attacks». En: *PLoS ONE* 11.11 (2016). DOI: 10.1371/journal.pone.0166047.
- [22] Z. Guzman-Zavaleta, C. Feregrino-Uribe, M. Morales-Sandoval y A. Menendez-Ortiz. «A robust and low-cost video fingerprint extraction method for copy detection». En: *Multimedia Tools and Applications* 76.22 (2017), págs. 24143-24163. DOI: 10.1007/s11042-016-4168-6.
- [23] A. Hampapur, K.-H. Hyun y R. Bolle. «Comparison of sequence matching techniques for video copy detection». En: *Proceedings of SPIE - The International Society for Optical Engineering* 4676 (2002), págs. 194-201. DOI: 10.1117/12.451091.
- [24] M. Hefeeda, T. Elgamal, K. Calagari y A. Abdelsadek. «Cloud-based multimedia content protection system». En: *IEEE Transactions on Multimedia* 17.3 (2015), págs. 420-433. DOI: 10.1109/TMM.2015.2389628.
- [25] Y. Himeur y K. Sadi. «Robust video copy detection based on ring decomposition based binarized statistical image features and invariant color descriptor (RBSIF-ICD)». En: *Multimedia Tools and Applications* 77.13 (2018), págs. 17309-17331. DOI: 10.1007/s11042-017-5307-4.
- [26] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever y R. R. Salakhutdinov. «Improving neural networks by preventing co-adaptation of feature detectors». En: *arXiv preprint arXiv:1207.0580* (2012).
- [27] Y. Hou, X. Wang y S. Liu. «A multiple features video copy detection algorithm based on a SURF descriptor». En: *Journal of Information Processing Systems* 12.3 (2016), págs. 502-510. DOI: 10.3745/JIPS.02.0042.

- [28] Y. Hu y X. Lu. «Learning spatial-temporal features for video copy detection by the combination of CNN and RNN». En: *Journal of Visual Communication and Image Representation* 55 (2018), págs. 21-29. DOI: 10.1016/j.jvcir.2018.05.013.
- [29] Q.-G. Ji, Z.-F. Tan, Z.-M. Lu e Y. Zhang. «An improved video identification scheme based on video tomography». En: *IEICE Transactions on Information and Systems* E97-D.4 (2014), págs. 919-927. DOI: 10.1587/transinf.E97.D.919.
- [30] C. Kim y B. Vasudev. «Spatiotemporal sequence matching for efficient video copy detection». En: *IEEE Transactions on Circuits and Systems for Video Technology* 15.1 (2005), págs. 127-132. DOI: 10.1109/TCSVT.2004.836751.
- [31] S. Kim, J. Choi, S. Han e Y. Ro. «Adaptive weighted fusion with new spatial and temporal fingerprints for improved video copy detection». En: *Signal Processing: Image Communication* 29.7 (2014), págs. 788-806. DOI: 10.1016/j.image.2014.05.002.
- [32] B. Kitchenham y S. Charters. «Guidelines for performing systematic literature reviews in software engineering». En: (2007).
- [33] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras e Y. Kompatsiaris. «Near-duplicate video retrieval by aggregating intermediate cnn layers». En: *International conference on multimedia modeling*. Springer. 2017, págs. 251-263.
- [34] A. Krizhevsky, I. Sutskever y G. E. Hinton. «ImageNet classification with deep convolutional neural networks». En: *Advances in neural information processing systems*. 2012, págs. 1097-1105.
- [35] O. Küüktunç, M. Batan, U. Gülükbay y O. Ulusoy. «Video copy detection using multiple visual cues and MPEG-7 descriptors». En: *Journal of Visual Communication and Image Representation* 21.8 (2010), págs. 838-849. DOI: 10.1016/j.jvcir.2010.07.001.
- [36] J. Law-To, O. Buisson, V. Gouet-Brunet y N. Boujema. «ViCopT: A robust system for content-based video copy detection in large databases». En: *Multimedia Systems* 15.6 (2009), págs. 337-353. DOI: 10.1007/s00530-009-0164-2.
- [37] H. Lee y J. Kim. «Extended temporal ordinal measurement using spatially normalized mean for video copy detection». En: *ETRI Journal* 32.3 (2010), págs. 490-492. DOI: 10.4218/etrij.10.0209.0485.
- [38] J. Li, Y. Liang y B. Zhang. «Video copy detection based on spatiotemporal fusion model». En: *Tsinghua Science and Technology* 17.1 (2012), págs. 51-59. DOI: 10.1109/TST.2012.6151907.
- [39] T. Li, F. Nian, X. Wu, Q. Gao e Y. Lu. «Efficient video copy detection using multi-modality and dynamic path search». En: *Multimedia Systems* 22.1 (2016), págs. 29-39. DOI: 10.1007/s00530-014-0387-8.
- [40] K. Liao y G. Liu. «An efficient content based video copy detection using the sample based hierarchical adaptive k-means clustering». En: *Journal of Intelligent Information Systems* 44.1 (2014), págs. 133-158. DOI: 10.1007/s10844-014-0332-5.
- [41] K. Liao, F. Zhao y M. Zhang. «Parallel implementation strategy for content based video copy detection using a multi-core processor». En: *KSII Transactions on Internet and Information Systems* 8.10 (2014), págs. 3520-3537. DOI: 10.3837/tiis.2014.10.014.
- [42] H. Liu, H. Lu y X. Xue. «A Segmentation and graph-based video sequence matching method for video copy detection». En: *IEEE Transactions on Knowledge and Data Engineering* 25.8 (2013), págs. 1706-1718. DOI: 10.1109/TKDE.2012.92.
- [43] H. Liu, Q. Zhao, H. Wang, P. Lv e Y. Chen. «An image-based near-duplicate video retrieval and localization using improved Edit distance». En: *Multimedia Tools and Applications* 76.22 (2017), págs. 24435-24456. DOI: 10.1007/s11042-016-4176-6.

- [44] H. Liu, Q. Zhao, H. Wang, P. Lv e Y. Chen. «An image-based near-duplicate video retrieval and localization using improved edit distance». En: *Multimedia Tools and Applications* 76.22 (2017), págs. 24435-24456.
- [45] M. Liu, L.-M. Po, Y. Ur Rehman, X. Xu, Y. Li y L. Feng. «Video copy detection by conducting fast searching of inverted files». En: *Multimedia Tools and Applications* (2018). DOI: 10.1007/s11042-018-6639-4.
- [46] Z.-M. Lu, B. Li, Q.-G. Ji, Z.-F. Tan e Y. Zhang. «Robust video identification approach based on local non-negative matrix factorization». En: *AEU - International Journal of Electronics and Communications* 69.1 (2014), págs. 82-89. DOI: 10.1016/j.aeue.2014.07.021.
- [47] F. J. Massey Jr. «The Kolmogorov-Smirnov test for goodness of fit». En: *Journal of the American statistical Association* 46.253 (1951), págs. 68-78.
- [48] H.-S. Min, J. Choi, W. De Neve e Y. Ro. «Bimodal fusion of low-level visual features and high-level semantic features for near-duplicate video clip detection». En: *Signal Processing: Image Communication* 26.10 (2011), págs. 612-627. DOI: 10.1016/j.image.2011.04.001.
- [49] H.-S. Min, J. Choi, W. De Neve e Y. Ro. «Near-duplicate video clip detection using model-free semantic concept detection and adaptive semantic distance measurement». En: *IEEE Transactions on Circuits and Systems for Video Technology* 22.8 (2012), págs. 1174-1187. DOI: 10.1109/TCSVT.2012.2197080.
- [50] L. Mou, T. Huang, Y. Tian, M. Jiang y W. Gao. «Content-based copy detection through multimodal feature representation and temporal pyramid matching». En: *ACM Transactions on Multimedia Computing, Communications and Applications* 10.1 (2013). DOI: 10.1145/2542205.2542208.
- [51] N. Nan y G. Liu. «Video Copy Detection Based on Path Merging and Query Content Prediction». En: *IEEE Transactions on Circuits and Systems for Video Technology* 25.10 (2015), págs. 1682-1695. DOI: 10.1109/TCSVT.2015.2395771.
- [52] X. Nie, J. Sun, Z. Xing y X. Liu. «Video fingerprinting based on graph model». En: *Multimedia Tools and Applications* 69.2 (2014), págs. 429-442. DOI: 10.1007/s11042-012-1341-4.
- [53] X. Nie, W. Zeng, H. Yan, J. Sun, Z. Liu y Q. Wang. «Structural similarity-based video fingerprinting for video copy detection». En: *IET Image Processing* 8.11 (2014), págs. 655-661. DOI: 10.1049/iet-ipr.2013.0689.
- [54] S. Paisitkriangkrai, T. Mei, J. Zhang y X.-S. Hua. «Clip-based hierarchical representation for near-duplicate video detection». En: *International Journal of Computer Mathematics* 88.18 (2011), págs. 3817-3833. DOI: 10.1080/00207160.2011.592940.
- [55] C. Panagiotakis, A. Doulamis y G. Tziritas. «Equivalent key frames selection based on iso-content principles». En: *IEEE Transactions on circuits and systems for video technology* 19.3 (2009), págs. 447-451.
- [56] M. K. A. Paul, J. Kavitha y P. A. J. Rani. «Key-Frame Extraction Techniques: A Review». En: *Recent Patents on Computer Science* 11.1 (2018), págs. 3-16.
- [57] S. Poullot, O. Buisson y M. Crucianu. «Scaling content-based video copy detection to very large databases». En: *Multimedia Tools and Applications* 47.2 (2010), págs. 279-306. DOI: 10.1007/s11042-009-0323-7.
- [58] D. Reynolds y R. Messner. «Video copy detection utilizing the log-polar transformation». En: *International Journal of Computing* 15.1 (2016), págs. 8-13.
- [59] R. Roopalakshmi. «A novel framework for CBCD using integrated color and acoustic features». En: *International Journal of Multimedia Information Retrieval* 4.1 (2015), págs. 45-57. DOI: 10.1007/s13735-014-0062-z.

- [60] R. Roopalakshmi y G. Ram Mohana Reddy. «A novel spatio-temporal registration framework for video copy localization based on multimodal features». En: *Signal Processing* 93.8 (2013), págs. 2339-2351. DOI: 10.1016/j.sigpro.2012.06.004.
- [61] G. Salton y C. Buckley. «Term-weighting approaches in automatic text retrieval». En: *Information processing & management* 24.5 (1988), págs. 513-523.
- [62] D. Sculley. «Web-scale k-means clustering». En: *Proceedings of the 19th international conference on World wide web*. ACM. 2010, págs. 1177-1178.
- [63] K. Simonyan y A. Zisserman. «Very deep convolutional networks for large-scale image recognition». En: *arXiv preprint arXiv:1409.1556* (2014).
- [64] P.-C. Su y C.-S. Wu. «Efficient copy detection for compressed digital videos by spatial and temporal feature extraction». En: *Multimedia Tools and Applications* 76.1 (2017), págs. 1331-1353. DOI: 10.1007/s11042-015-3132-1.
- [65] Z. Sun, Y. Zhu, X. Liu y L. Zhang. «A robust video fingerprinting algorithm based on centroid of spatio-temporal gradient orientations». En: *KSII Transactions on Internet and Information Systems* 7.11 (2013), págs. 2754-2768. DOI: 10.3837/tiis.2013.11.012.
- [66] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke y A. Rabinovich. «Going deeper with convolutions». En: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, págs. 1-9.
- [67] K. Tadmir y A. Çetin. «Content-based video copy detection based on motion vectors estimated using a lower frame rate». En: *Signal, Image and Video Processing* 8.6 (2014), págs. 1049-1057. DOI: 10.1007/s11760-014-0627-6.
- [68] Y. Tian, T. Huang, M. Jiang y W. Gao. «Video copy-detection and localization with a scalable cascading framework». En: *IEEE Multimedia* 20.3 (2013), págs. 72-86. DOI: 10.1109/MMUL.2012.62.
- [69] Y. Tian, M. Qian y T. Huang. «TASC: A transformation-aware soft cascading approach for multimodal video copy detection». En: *ACM Transactions on Information Systems* 33.2 (2015), pág. 7. DOI: 10.1145/2699662.
- [70] S. Wei, Y. Zhao, C. Zhu, C. Xu y Z. Zhu. «Frame fusion for video copy detection». En: *IEEE Transactions on Circuits and Systems for Video Technology* 21.1 (2011), págs. 15-28. DOI: 10.1109/TCSVT.2011.2105554.
- [71] X. Wu, C.-W. Ngo, A. Hauptmann y H.-K. Tan. «Real-time near-duplicate elimination for web video search with content and context». En: *IEEE Transactions on Multimedia* 11.2 (2009), págs. 196-207. DOI: 10.1109/TMM.2008.2009673.
- [72] X. Wu, A. G. Hauptmann y C.-W. Ngo. «Practical elimination of near-duplicates from web video search». En: *Proceedings of the 15th ACM international conference on Multimedia*. ACM. 2007, págs. 218-227.
- [73] G. Yang, N. Chen y Q. Jiang. «A robust hashing algorithm based on SURF for video copy detection». En: *Computers and Security* 31.1 (2012), págs. 33-39. DOI: 10.1016/j.cose.2011.11.004.
- [74] X. Zhang, Y. Xie, X. Luan, J. He, L. Zhang y L. Wu. «Video Copy Detection Based on Deep CNN Features and Graph-Based Sequence Matching». En: *Wireless Personal Communications* 103.1 (2018), págs. 401-416. DOI: 10.1007/s11277-018-5450-x.
- [75] J. Zhu, S. Hoi, M. Lyu y S. Yan. «Near-duplicate keyframe retrieval by semi-supervised learning and Nonrigid image matching». En: *ACM Transactions on Multimedia Computing, Communications and Applications* 7.1 (2011). DOI: 10.1145/1870121.1870125.

Sistema basado en blockchain para la realización de análisis en entornos laborales informatizados

Francisco Lecumberri de Alba¹, Juan M. Corchado¹, Javier Prieto¹

¹Departamento de informática y automática,
Facultad de Ciencias,
Universidad de Salamanca,
Plaza de los Caídos s/n 37008 Salamanca
{fcolecumberri, corchado, javierp}@usal.es

Resumen En el presente trabajo se describe el desarrollo de un sistema de monitoreo pasivo para un ambiente de trabajo usando visión artificial sobre un sistema multi-agentes ROS y almacenando la información en un sistema *blockchain*. Los puntos importantes de este desarrollo son la detección de rostros, identificar la atención del usuario, almacenar la información de la *blockchain* y despegarlo en una aplicación web.

Keywords: Visión artificial, blockchain, ROS, redes neuronales, sistemas multi-agente

1. Introducción

1.1. Motivación y Justificación

La evaluación del desempeño de los trabajadores en un ambiente de oficina puede llegar a ser complicado ya que el empleador no puede estar supervisando toda la jornada laboral de sus empleados y revisando que todos cumplan en estas horas laborales las múltiples tareas asignadas. Esto se complica aún más si el empleador tiene un gran número de trabajadores. También hoy en día se tiene la nueva tendencia del *home-office* donde la supervisión del rendimiento del trabajador resulta aún más difícil.

Los problemas que pueden existir durante las horas laborales en un ambiente de oficina son que los empleados desvíen su atención del trabajo a realizar, pudiendo ser posibles distracciones como visitar sus redes sociales, jugando con el teléfono celular, hablar con sus compañeros o incluso estar durmiendo.

En consecuencia de lo anteriormente expuesto el presente trabajo propone un sistema de monitoreo pasivo de forma tal que no interfiera con las actividades laborales del empleado, que sea capaz de detectar actividades que considere y clasifique como distracciones en el horario laboral. Por esta razón se ha optado por utilizar las nuevas tecnologías de la visión artificial.

Debido a que la manipulación de información puede ser de interés para propósitos carentes de ética esta información será almacenada en un sistema que pueda asegurar su inmutabilidad posterior a la generación.

El *blockchain* posee como característica principal que una vez iniciada una aplicación distribuida en el *blockchain*, toda la información que sea almacenada no podrá ser alterada. Esto nos motiva al uso del *blockchain* para satisfacer la necesidad de seguridad.

1.2. Objetivos

El objetivo general del presente trabajo de fin de máster es desarrollar un sistema de visión artificial que permita la supervisión automatizada de trabajadores que utilicen un ordenador durante su jornada laboral almacenando la información en una aplicación distribuida para posteriormente poderla consultar.

A partir de este objetivo general se subdividen varios objetivos específicos, a continuación se describirán a mayor profundidad:

Detección de rostros.

Se pretende realizar la detección del rostro de un trabajador, la cual se conseguirá con un sistema que sea capaz de detectar la presencia de dicho rostro, midiendo parámetros como la posición y tamaño de este dentro de la imagen y a su vez los ojos dentro del mismo.

Identificar la atención del usuario.

Se identificará la atención o distracción del trabajador a partir de analizar la información de los parámetros del rostro adquiridos en el objetivo anterior.

Almacenarlo en una *blockchain*.

El almacenamiento en una *blockchain* es en si mismo un objetivo que debe dividirse en varios objetivos particulares que serán planteados a continuación.

Crear una aplicación distribuida.

Es necesario crear una aplicación distribuida sobre la tecnología *blockchain* que permita el almacenamiento de la información para poder posteriormente tener acceso a esta.

Es preciso que la aplicación distribuida sea lo mas ligera y eficiente posible pues se asume que cualquier costo en términos de procesamiento se debe costear por cada empleado y por cada imagen procesada.

Crear un cliente *blockchain* para el registro de información.

Una vez que la aplicación distribuida este funcionando. Es necesario crear el cliente de la aplicación que se encargara de emitir la información adquirida sobre el empleado para que esta se pueda almacenar en la *blockchain*.

Crear un cliente *blockchain* para el acceso a la información.

Una vez que la información se encuentre almacenada en el *blockchain*. Sera necesario crear un cliente que permita acceder a la información y poder obtener la información a detalle.

Desarrollo sobre un sistema de agentes.

Las partes descrita en las secciones 1.1, 1.2 y 1.3 son perfectamente separables en tareas independientes por lo que deberán ser programadas usando un sistema de agentes.

1.3. Análisis del Estado del Arte

En la presente sección se desarrollara el análisis del estado del arte comenzando con la visión artificial y el reconocimiento de rostros haciendo énfasis principal en los algoritmos Viola-Jones y las redes neuronales convolucionales. En la segunda parte de esta sección se hace una revisión sobre las tecnologías de programación orientada a agentes pasando por algunos lenguajes y algunas herramientas del tipo *middleware* como lo son JADE, GOAL, SARL, Jason y ROS. Finalmente esta revisión del estado del arte expondrá las tecnologías del *blockchain* mas prominentes como lo son Ethereum y Hyperledger.

Visión Artificial y Reconocimiento de Rostros. En el tema de visión artificial, existen principalmente dos algoritmos, el primer algoritmo es el algoritmo red neuronal artificial convolucional (CNN) y el segundo es el algoritmo de Viola-Jones.

Red neuronal artificial convolucional.

Los algoritmos CNN tienen sus orígenes al final de la década de los noventa [11] y desde entonces han evolucionado [5] en gran medida gracias al desarrollo de la tecnologías GPU.

Estos algoritmos empiezan por aplicar de forma iterativa una convolución con una matriz determinada a una imagen, debido a que cada iteración el tamaño de la imagen disminuye, llega a un punto en el que la imagen ocupa pocos bits. Este proceso se aplica con diversas matrices de convolución por lo que cada matriz reduce la imagen original a unos cuantos bits. Una vez realizado ese procesamiento de simplificación se entrena una red neuronal con la información simplificada. Esto se puede apreciar en la Fig. 1

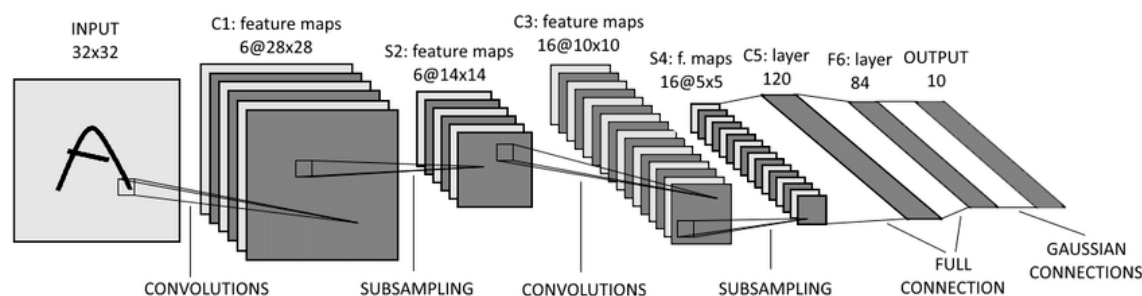


Figura 1. Proceso de simplificación por convoluciones del algoritmo CNN

El principal problema que tiene este algoritmo es que si bien logra reducir la complejidad del análisis lo suficiente como para que una red neuronal se encargue de resolver el problema, lamentablemente parte de la información que se pierde en la etapa de simplificación es la posición del elemento para el que se esta entrenando.

Debido a este problema, el algoritmo CNN queda descartado como una solución en esta aplicación.

Viola-Jones.

El algoritmo de Viola-Jones[29] es un algoritmo que esta basado en la extracción de características de tipo-Haar de una representación en escala de grises de la imagen para posteriormente hacer una clasificación mediante un árbol de decisiones entrenado con el método AdaBoost. El árbol de clasificación se realiza comparando secciones de la imagen con una serie de mascararas que se pueden apreciar en la imagen 2. La forma de comparar las imágenes se puede ver en la figura 3.

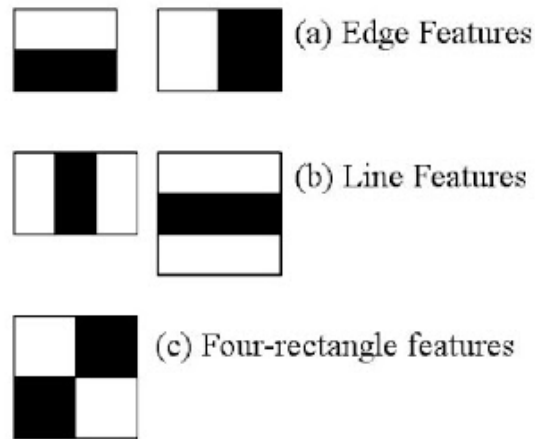


Figura 2. Características de tipo-Haar usadas en el algoritmo Viola-Jones

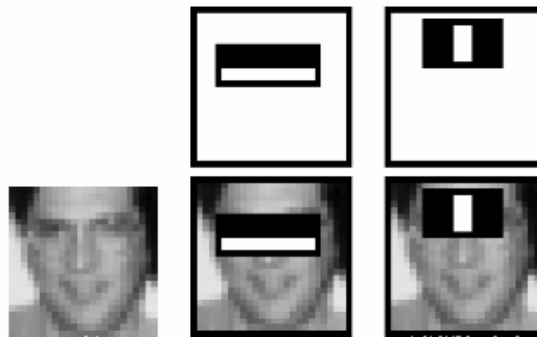


Figura 3. Forma de la comparación de las características tipo-Haaren el algoritmo Viola-Jones

Una de las ventajas mas significativas que demostró el algoritmo de Viola-Jones es la alta eficiencia computacional. El algoritmo logró reconocer en tiempo real rostros en una imagen de resolución 384x288 a 15 imágenes por segundo usando un procesador Intel Pentium III de 700 MHz lo que supuso una mejora importante en el tema.

Si bien es cierto que para este problema no tenemos una resolución definida, una frecuencia de muestreo definida ni tenemos un procesador definido y que se espera que la imagen posea una resolución mayor a 384x288 de la misma manera podemos asumir que el procesador sobre el que se ejecute el sistema sera mas potente que un Intel Pentium III de 700 MHz. Por lo que sabemos que en términos de capacidad computacional el algoritmo no supone un inconveniente.

Asimismo otra ventaja que supone este algoritmo es que se obtiene no solo la presencia o ausencia del elemento buscado, también se obtiene posición, anchura y altura, por lo que el algoritmo resulta muy conveniente para este problema.

Otra ventaja que ofrece es que una vez encontrado el rostro de la persona, se puede volver a aplicar el algoritmo en el espacio reducido del rostro para obtener la posición de los ojos.

Sistemas de Programación Orientada a Agentes. La programación orientada a agentes propuesta originalmente en la universidad de Stanford [25] es un paradigma de programación que permite separar un programa que debiera solucionar múltiples tareas en múltiples agentes que deben solucionar una tarea cada agente.

Existen varias herramientas para la implementación del concepto de programación orientada a agentes como lo son: Java Agent DEvelopment Framework (JADE) [8], Jason [9], GOAL [4], SARL [23] y Robot Operating System (ROS) [20].

JADE.

JADE es un *middleware* desarrollado para el lenguaje de programación Java y permite crear un desarrollo orientado a agentes sobre este lenguaje. Jade permite manejar a cada agente como un hilo independiente o entre distintas maquinas a través de la red. JADE presenta múltiples ventajas en el concepto del desarrollo orientado a agentes. El único problema que presenta JADE es el que no permite reutilizar código desarrollado por terceros mientras que esta característica esta muy presente en ROS por lo que al final se opto por usar ROS por encima de JADE. De no ser por la existencia de ROS JADE seria nuestra siguiente opción.

GOAL.

GOAL es un lenguaje de programación orientado a agentes para el desarrollo de un sistemas cognitivos. GOAL ofrece herramientas para programar los objetivos y creencias de cada agente así como su interacción. Debido al hecho de que este desarrollo requiere resolver problemas como captura de imagen de una cámara web, comunicación con aplicaciones distribuidas en *blockchain*, entre otras tareas, usar un lenguaje que no presenta soluciones para estos problemas resulta contraproducente por lo que no se usara GOAL en este desarrollo.

SARL.

SARL es un lenguaje de programación orientado a agentes que posee extensiones para cooperar con desarrollos en el lenguaje de programación Java. Este lenguaje posee la misma carencia que GOAL. Usar un lenguaje que en si mismo no trae integrado las herramientas para el acceso al hardware utilizado ni formas de comunicación con las aplicaciones distribuidas supone mas trabajo que el que solventa.

Jason.

Jason es otra herramienta de programación orientada a agentes la cual funciona como extensión del lenguaje AgentSpeak sobre el lenguaje de programación Java. Jason sufre de los problemas ya descritos aquí sobre las otras herramientas. Jason no ofrece la funcionalidad desarrollada por terceros y esto genera mas problemas.

ROS.

Ros es un sistema enfocado en la capacidad de reutilizar software de manera tal que permite que muchos de los agentes que otros desarrolladores han creado se puedan volver a utilizar sin problemas. Además, varios de los problemas descritos en la sección 1.2 ya poseen una solución integrada por parte del sistema ROS.

ROS propiamente no se define a si mismo como un sistema multi-agentes, sin embargo cabe destacar que tampoco es un sistema operativo a pesar del nombre y el hecho de que su funcionamiento se acopla perfectamente bien a la definición de sistemas multi-agentes.

El sistema ROS permite desarrollar agentes o nodos ¹ que se comunicaran entre ellos utilizando 2 sistemas: El sistema de Tópicos y el sistema de servicios.

Ambos sistemas usan un nombre para identificar el canal de comunicación y una estructura la cual es registrada en el momento en que un nodo hace uso de el canal. De esta forma se fuerza a que todos los nodos que pretendan usar ese canal de comunicación sean congruentes con la información que pasa a través del canal. En el caso de los tópicos, cualquier nodo puede publicar o suscribirse y cuando un nodo publica, todos los nodos suscritos reciben una copia del mensaje. En el caso de los servicios, un nodo toma el papel de servidor y cualquier nodo puede acceder al servicio. La estructura de los servicios difiere levemente con respecto a la de los tópicos en cuanto a que esta se subdivide en 2 subestructuras, una de mensaje y otra de respuesta.

Existen nodos ya implementados en ROS que permiten acceder a la cámara web[2] y utilizar el algoritmo de Viola-Jones [16] por lo que usar ROS ahorrara tiempo de desarrollo y permitirá que en el futuro cualquier modificación sea menos difícil pues estamos usando soluciones ya estandarizadas dentro de la plataforma.

Debido a esto se opto por usar ROS como herramienta de programación orientada a agentes.

Blockchain. *Blockchain*, en el más puro sentido del concepto, es una serie de datos almacenados en bloques de forma tal que cada bloque posee el resultado de una función de dispersión evaluada sobre los datos del bloque anterior. Esto ofrece una forma de asegurar que para hacer un mínimo cambio en un bloque de la cadena tenga un coste computacional inasumible por lo que le da inmutabilidad a la información. Esta inmutabilidad permite que la información se distribuya por la red sin riesgo a que existan alteraciones.

Una *blockchain* posee una estructura descentralizada sin jerarquías que esta formada por seis capas las cuales son datos, red, consenso, contrato, servicio y aplicación. La capa de datos almacena los datos, los metadatos e información de cifrado. Por otro lado, la capa de red abarca la conexión entre todos los nodos del sistema e incluyen las conexiones entre los nodos de la red. Luego las capas de consenso y contrato incluyen contratos inteligentes, protocolos de consenso y mecanismos de incentivos. mientras que las capas de servicio y aplicación implementan en la práctica actividades basadas en *blockchain*.

Para promover el intercambio de recursos y promover la comunicación colaborativa se suele crear mecanismo de incentivo. el mecanismo de incentivo podría construirse como una capa separada Como protocolo de promoción.

Con respecto a la práctica industrial, las plataformas de consorcios y *blockchain* privadas se establecen para un grupo específico, en lugar de para todos. Por lo tanto, los mecanismos de incentivo o las actividades mineras no son obligatorias. Este cambio es más propicio para las comunicaciones y transacciones entre los participantes en los sistemas de cadena de bloques más amplios. De acuerdo con la arquitectura, el *blockchain* incluye tres elementos centrales: una estructura de bloque de cadena basada en la marca de tiempo, un mecanismo de almacenamiento distribuido basado en una red P2P y un mecanismo de consenso basado en nodos descentralizados.

Por otra parte la tecnología *blockchain* posee las siguientes características [12]:

Descentralización.

La red consta de nodos que se comunican en un protocolo punto a punto (*peer to peer* o

¹ La documentación de ROS llama a los agentes con el nombre de **nodos** por lo que usaremos ambos términos de forma indistinta.

p2p). Esto permite que la información se comparta y distribuya hacia todos los nodos sin la intervención de un tercer elemento. En este sistema todos los participantes y nodos son libres de unirse a actividades y transacciones.

Desconfianza.

Debido a que la tecnología *blockchain* es implementada de forma descentralizada, la transferencia de información entre los nodos no requiere una confianza mutua entre los participantes. La tecnología *blockchain* utiliza protocolos de consenso y funciones de dispersión para resolver este problema.

Transparencia.

A lo largo del *blockchain* todos los participantes comparten los mismos registros en una estructura descentralizada. Cada participante puede consultar los registros en la *blockchain* para que la información en el sistema distribuido sea transparente y consistente. Cada nodo de la misma plataforma tiene los mismos permisos y obligaciones para acceder a la información autorizada y permite que otros nodos en la misma red accedan a esta información.

Trazable e inmutable.

La cadena de bloques utiliza marcas de tiempo para identificar y registrar cada transacción, mejorando así la dimensión temporal de los datos. Esto permite al nodo mantener el orden de las transacciones y hacer que los datos sean rastreables. La marca de tiempo no solo garantiza la originalidad de los datos, sino que también reduce el costo de la trazabilidad de la transacción. Al mismo tiempo, refuerza modificaciones irreversibles a los datos o información. Una vez que se valida una transacción y se agrega al bloque, no se manipulará. La mayoría de los nodos del sistema deben revisar las transacciones antes de poder registrarlas.

Anonimato.

El *blockchain* encripta los datos utilizando técnicas de cifrado asimétrica. Este cifrado asimétrico tiene dos usos en *blockchains*: cifrado de datos y firmas digitales. El cifrado de datos en la cadena de bloques garantiza la seguridad de los datos de transacción y reduce el riesgo de perder o falsificar los datos de transacción. Los datos de la transacción se transmiten a través de la red y se firman digitalmente para indicar la identidad del firmante y si se ha identificado la transacción. En el sistema *blockchain*, no es necesario revelar la verdadera identidad del nodo asociado con el participante.

Credibilidad.

El intercambio de datos de la cadena de bloques depende completamente del autocontrol. Se basa en cada nodo para formar un cálculo poderoso para defenderse contra ataques externos sin intervención humana. Los participantes pueden completar la transacción bajo condiciones de completo anonimato. Protege la privacidad de todas las partes involucradas y aumenta la seguridad y la credibilidad de la transacción. Además, cada nodo en la cadena de bloques almacena los datos completos.

Para poder desarrollar sobre un sistema *blockchain* es indispensable buscar un ecosistema de desarrollo adecuado.

La tecnología *blockchain* es aun bastante novedosa (los primeros bosquejos dados por Satoshi Nakamoto tienen aproximadamente una década de existir) por lo que están constantemente apareciendo tecnologías nuevas con enfoques distintos que se pueden considerar que se encuentran en estado experimental.

Debido a esto, aunque la cantidad de tecnologías este aumentando aceleradamente, hay pocas tecnologías que tengan un nivel de madurez lo suficientemente elevado para que podamos presumir

que no quedaran en desuso ante la creciente competencia. También es necesario que la documentación oficial y extraoficial sea lo suficientemente amplia para que no suponga un problema la falta de documentación. Todo lo anteriormente mencionado hace que la cantidad de opciones disminuya drásticamente.

A partir de aquí aparecen dos tecnologías con el nivel de madurez adecuado para este desarrollo. Estas tecnologías son Hyperledger y Ethereum. Ambas tecnologías tienen enfoques distintos que se discutirán a continuación.

Hyperledger.

Hyperledger es un sistema *blockchain* con objetivos empresariales que emplea un contrato inteligente para imponer la confianza entre los participantes. Hyperledger está diseñado para aplicaciones de *blockchain* de nivel empresarial e introduce servicios de administración de miembros. En diciembre de 2015, la Fundación Linux lanzó el Proyecto Hyperledger para desarrollar una plataforma de *blockchain* de negocios entre industrias. Hyperledger es un *blockchain*, pero no es una criptomoneda. En el sistema Hyperledger, no es necesario incrustar criptomoneda o actividad minera. El beneficio de este cambio en Hyperledger es el rendimiento de todo el sistema. En general, Hyperledger no aplica hardware específico, software adicional, infraestructura de red o módulos de seguridad. La plataforma Hyperledger es un sistema relativamente adecuado que puede satisfacer los diversos requisitos de las actividades comerciales.

Ethereum.

En diciembre de 2013, se propuso la plataforma de cadena de bloques Ethereum y un lenguaje de programación completo para redactar contratos inteligentes basados en transacciones de moneda digital Ethereum incorporadas. Ethereum tiene el potencial de crear una computadora mundial descentralizada que nunca se detiene, que no se censura y se da mantenimiento automáticamente. Ethereum es una plataforma de cadena de bloques de código abierto que utiliza contratos inteligentes para ofrecer servicios o aplicaciones. Ethereum está diseñado para emplear el lenguaje EVM (código de máquina virtual de Ethereum) para ejecutar el programa, y Solidity es el programa de compilación más utilizado.

Tras analizar ambas plataformas se determinó que el desarrollo podría realizarse en cualquiera de las dos plataformas de manera indistinta. La decisión final fue desarrollarla en Ethereum.

2. Desarrollo

En la presente sección tras haber analizado los objetivos y el análisis del estado del arte se optó por generar dos desarrollos simples de soporte y el desarrollo principal. Los desarrollos simples son el desarrollo de adquisición de información para el entrenamiento del sistema de reconocimiento de atención y el entrenamiento del sistema de atención.

2.1. Programa de adquisición de información para el entrenamiento del sistema de reconocimiento de atención.

El primer desarrollo es el programa de adquisición de información para el entrenamiento del sistema de reconocimiento de atención. Este desarrollo comienza con la adquisición de la información de la cámara, para esto se aprovechó la existencia de un nodo de ROS del paquete `cv_camera[2]` que permite adquirir la imagen de forma transparente.

Esta información es publicada en un tópico llamado `/cv_camera/image_raw` misma que es adquirida por el nodo `face_detection` del paquete `opencv_apps` [16] y se publica en el tópico `/face_detection/faces`.

El tópico `/cv_camera/image_raw` usa la estructura de mensajes `sensor_msgs/Image` y el tópico `/face_detection/faces` usa la estructura `opencv_apps/FaceArrayStamped`. Estas dos estructuras se describen en las tablas 1, 2, 3, 4 y 5.

| sensor_msgs/Image | |
|-------------------|--------------|
| tipo | campo |
| std_msgs/Header | header |
| uint32 | height |
| uint32 | width |
| string | encoding |
| uint8 | is_bigendian |
| uint32 | step |
| uint8[] | data |

Tabla 1. Estructura del mensaje `sensor_msgs/Image`[24].

| std_msgs/Header | |
|-----------------|----------|
| tipo | campo |
| uint32 | seq |
| time | stamp |
| string | frame_id |

Tabla 2. Estructura del mensaje `std_msgs/Header`[26].

| opencv_apps/FaceArrayStamped | |
|------------------------------|--------|
| tipo | campo |
| std_msgs/Header | header |
| opencv_apps/Face[] | faces |

Tabla 3. Estructura del mensaje `opencv_apps/FaceArrayStamped`[18].

| opencv_apps/Face | |
|--------------------|------------|
| tipo | campo |
| opencv_apps/Rect | face |
| opencv_apps/Rect[] | eyes |
| string | label |
| float64 | confidence |

Tabla 4. Estructura del mensaje opencv_apps/Face[17].

| opencv_apps/Rect | |
|------------------|--------|
| tipo | campo |
| float64 | x |
| float64 | y |
| float64 | width |
| float64 | height |

Tabla 5. Estructura del mensaje opencv_apps/Rect[19].

El nodo `face_detection` utiliza el algoritmo de `Cascade Classifier`[15] también conocido como `Haar-Cascade` el cual es simplemente la implementación de OpenCV del algoritmo de Viola-Jones[15].

La información de los rostros es después accedida por el nodo de desarrollo propio `training_data_maker.py` y la almacena en un archivo en formato `comma separated value` (CSV).

Este proceso puede ser apreciado en la Fig. 4. todas las figuras similares se hicieron usando el software `ROSGraphVizDoc`[10]. Sobre la estructura del archivo de información de rostros, esta se puede revisar en la Tab. 6.

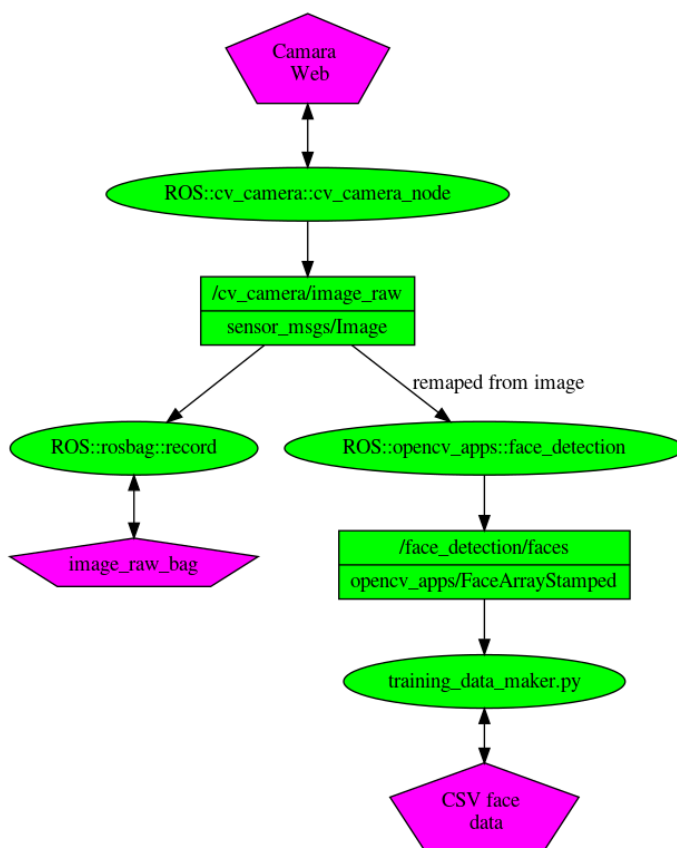


Figura 4. Diagrama del desarrollo de adquisición de información para el entrenamiento del sistema de reconocimiento de atención.

| Numero de campo | Contenido |
|-----------------|--|
| 1 | ¿Esta viendo a la pantalla? |
| 2 | Posición del centro del rostro en X |
| 3 | Posición del centro del rostro en Y |
| 4 | Anchura del rostro |
| 5 | Altura del rostro |
| 6 | Posición del centro del primer ojo en X |
| 7 | Posición del centro del primer ojo en Y |
| 8 | Anchura del primer ojo |
| 9 | Altura del primer ojo |
| 10 | Posición del centro del segundo ojo en X |
| 11 | Posición del centro del segundo ojo en Y |
| 12 | Anchura del segundo ojo |
| 13 | Altura del segundo ojo |

Tabla 6. Campos del archivo CSV (Todos los campos son enteros)

Como se puede notar, el campo 1 de la Tab. 6 es un campo que se adquiere como variable de entorno. Por esto una vez iniciado el proceso de adquisición de la información el usuario debe solo actuar como si prestara atención o solo actuar como si estuviera distraído hasta que termine el proceso de muestreo dependiendo de cual sea la muestra que se este realizando. Debido a esto, es necesario hacer dos muestreos.

La ventaja mas grande que hay al realizar los dos muestreos por separado es que se omite la necesidad de separar manualmente los casos donde un usuario se encuentra distraído o no.

Se puede observar en la Fig. 4 que a parte del proceso indicado, también se almaceno la información de `/cv_camera/image_raw` en un `bag` usando la herramienta `rosvbag` [21]. Esto permite que en caso de que en caso de algún inconveniente, se pueda replicar el proceso sin necesidad de repetir la medición.

2.2. Programa de entrenamiento del sistema de reconocimiento de atención.

La información adquirida como consecuencia del desarrollo descrito en la sección 2.1 resulto en una serie de archivos CSV que al concatenarse dieron lugar a un archivo CSV con 2236 muestras.

Una vez adquirida la información fue necesario desarrollar un sistema que fuera capaz de discernir si el usuario presta o no atención. Para este desarrollo se decidió utilizar una red neuronal. La red neuronal fue desarrollada utilizando la herramienta Keras[1] el cual utiliza como *backend* a TensorFlow[13].

La red construida cuenta con dos capas, la primera capa con cincuenta neuronas y la segunda con una. Ambas capas cuentan con una función de activación sigmoideal.

La red neuronal fue entrenada con 2000 muestras obtenidas de forma aleatoria lo que constituye el 89.45 % y se usaron 236 muestras para evaluar el entrenamiento. De las 236 muestras, la red neuronal logró acertar en 230 lo que representa un 97.46 % de confianza. Colocando esto en perspectiva, asumiendo que el sistema evalúa a un empleado que trabaja ocho horas, habría una incertidumbre de doce minutos con doce segundos.

Un ejemplo de como se ve una imagen donde el usuario se encuentra poniendo atención se puede observar en la Fig. 5. Mientras que, en contraste, en la Fig. 6 se muestra a un usuario no prestando atención al monitor.

Las figuras 5 y 6 son capturas de pantalla al software `rqt`[22] el cual es una de las múltiples herramientas de visualización de ROS desarrollada sobre `qt`[14]. En esta imágenes podemos observar que el sistema realizo varios marcajes siendo estos un ovalo magenta y dos óvalos azules. La marca en color magenta sitúa el centro del óvalo en el centro del rostro mientras que el alto y ancho del óvalo son congruentes con el alto y ancho del rostro. De la misma manera las marcas azules colocan el centro de los óvalos azules en el centro de cada ojo. Se puede apreciar también en la imagen de la figura 6 que el ojo que esta más lejos de la cámara es mas pequeño debido a la perspectiva lo que conlleva a que el óvalo sea sustancialmente mas pequeño mientras que en la imagen de la Fig. 5 ambos ojos se encuentran del mismo tamaño. Utilizando este tipo de información una red neuronal debiera ser capaz de cumplir con el objetivo de determinar si el usuario presta o no atención al entrono laboral.

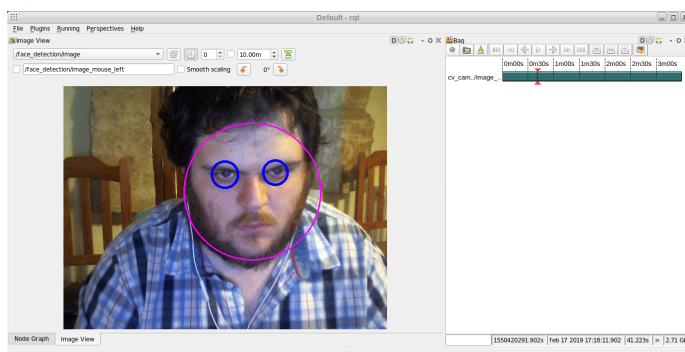


Figura 5. Usuario prestando atención al monitor

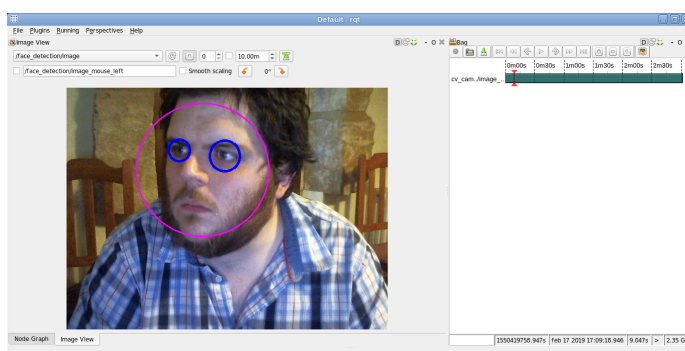


Figura 6. Usuario no prestando atención al monitor

2.3. Programa Principal

El programa principal se divide en tres subdesarrollos complementarios. El primero es el desarrollo por el lado del sistema de medición el cual se encarga de detectar el comportamiento del empleado para posteriormente enviar la información al *blockchain*, el segundo es la aplicación distribuida la cual se encarga de almacenar la información de los empleados y el tercero es el cliente de visualización del *blockchain* el cual se encarga de presentar los datos.

Desarrollo por el Lado del Sistema de Medición. Este desarrollo no difiere demasiado del desarrollo mostrado en la sección 2.1. Al comienzo del sistema tenemos el nodo `cv_camera_node` que adquiere la imagen de la cámara web y la transmite al tópico `/cv_camera/image_raw`. Posteriormente el nodo `face_detection` adquiere la información y tras aplicar el algoritmo `Cascade Classifier` brinda la información de los rostros sobre el tópico `/face_detection/faces`.

Posteriormente, a diferencia del desarrollo de la sección 2.1, la información de los rostros la adquiere el nodo `nnet_predictor.py` el cual utiliza el modelo de la red neuronal generado en la sección 2.2 y predice si la persona está o no prestando atención. Cabe destacar que el nodo `nnet_predictor.py` antes de aplicar la red neuronal revisa que al menos exista una cara con dos ojos detectados, de no ser así se asume que no hay persona frente al monitor. Después de establecer

que efectivamente hay un rostro y que este posee los dos ojos visibles, se procede a enviar un mensaje sobre el t3pico `is_looking` con la estructura `IsLooking`. La estructura `IsLooking` fue definida como parte de este desarrollo y la estructura se puede apreciar en la Tab. 7.

| IsLooking | |
|------------------------------|-------------------------|
| tipo | campo |
| <code>std_msgs/Header</code> | <code>header</code> |
| <code>bool</code> | <code>is_looking</code> |

Tabla 7. Estructura de mensaje `IsLooking`

Cabe recordar que la estructura `std_msgs/Header` es una estructura ya definida en ROS que posee metadatos de manera estandarizada. Esta estructura se encuentra presente desde la estructura `sensor_msgs/Image` y esta es transmitida por la estructura `opencv_apps/FaceArrayStamped` de igual forma. Lo importante de esta estructura es que contiene el timestamp del momento en que la imagen fue adquirida.

Una vez que la informaci3n del t3pico `/is_looking` llega al nodo `blockchain_client.py`, el nodo utiliza esa informaci3n para generar la petici3n que ser3 realizada a la aplicaci3n distribuida utilizando la librer3a `web3py`[6].

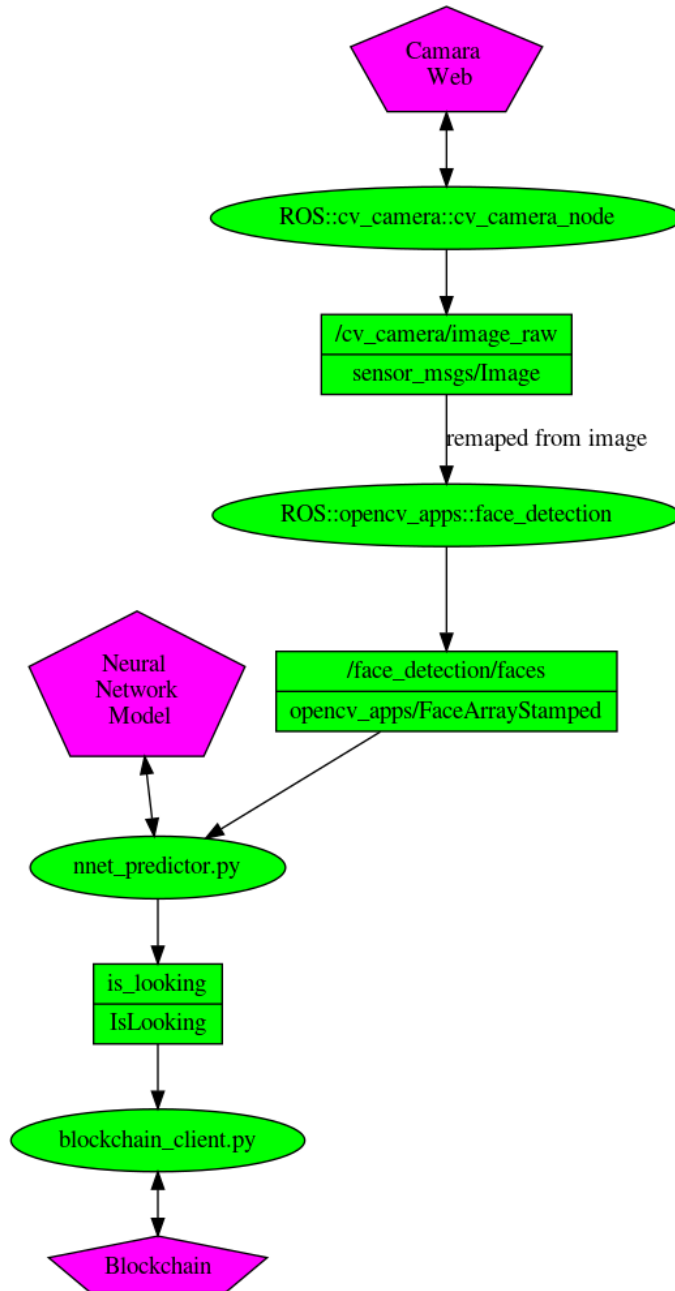


Figura 7. Diagrama del desarrollo general.

La aplicación distribuida en blockchain. Como se explicó en la sección 1.2 la aplicación distribuida debe ser desarrollada de la forma mas ligera posible. Por esta razón la información es almacenada en forma de eventos en lugar de arreglos o cualquier otra forma de almacenamiento.

La aplicación distribuida posee la función externa llamada `regist_activity()` la cual recibe tres parámetros los cuales son el identificador del empleado del tipo `string`, el momento del suceso evaluado nanosegundos desde el 1 de enero de 1970 en formato `uint64` y la evaluación booleana en formato `uint8` que describe si la persona estaba en su puesto de trabajo y prestando atención. La información que recibe esta función queda almacenada en el evento que esta emite.

Hay que recordar que aunque la *blockchain* posea un sistema intrínseco de marcar el tiempo en que un mensaje fue generado, es necesario usar el tiempo que proviene del mensaje porque es el tiempo en el que se adquirió la imagen. El tiempo después de procesar y emitir la imagen no es importante.

El desarrollo se realizo utilizando las herramientas `Truffle`[28] y `Ganache`[27], donde `Truffle` es una herramienta para compilación y despliegue de aplicaciones distribuidas y `Ganache` es un entorno virtual que simula ser la red `Ethereum`.

La interfaz gráfica del blockchain. La interfaz gráfica fue desarrollada desde una aplicación web utilizando las herramientas de la librería `web3js`[3] para poder acceder a la información almacenada en la *blockchain*. Una vez obtenida esta información, esta se agrupa por empleado y se genera un gráfica por cada empleado como se puede observar en la imagen 8 utilizando la librería `Chart.js`[7].

En cuanto a la gráfica que se muestra en la Fig. 8 se tiene que en el eje X representa el tiempo en el que fue adquirida la información del usuario mientras que en el eje Y se representa la presencia y la atención del usuario en el su puesto de trabajo de forma tal que cuando cuando el valor es 1 esta indica que el usuario se encuentra en su puesto de trabajo prestando atención. En contraste cuando el valor esta en 0 significa que el usuario no se encuentra prestando atención o simplemente se encuentra ausente. A la derecha de la gráfica aparece el nombre del usuario que esta siendo representado en la gráfica.

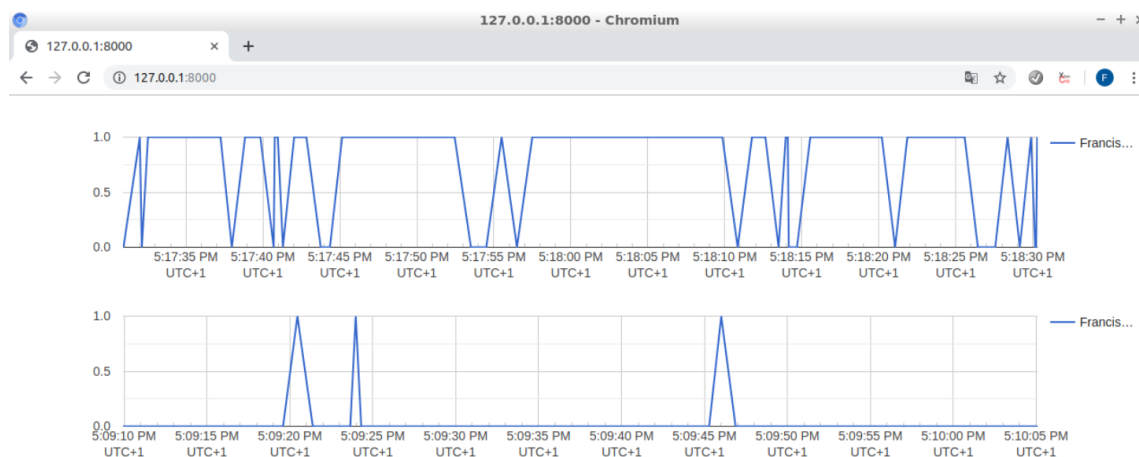


Figura 8. Interfaz web

3. Conclusiones y Resultados

En las siguientes sección se dará evidencia de que cada uno de los objetivos en la sección 1.2 fue cumplida exitosamente.

Detección de rostros.

El objetivo de la detección de rostros se puede observar que se cumplió con éxito ya que en las secciones 2.1 y 2.3 en ambas se hace uso del nodo que permite detectar rostros y en los dos se abstrae información que permitirá que una red neuronal determine si el usuario esta o no prestando atención.

Identificar la atención del usuario.

Por otra parte el objetivo de poder distinguir si el usuario esta o no prestando atención fue exitosamente completado pues como se puede observar a lo largo de toda la sección 2 y especialmente en la sección 2.2 donde se puede apreciar que la identificación de atención logro un 97.46 % de éxito. Debido a esto es evidente que el objetivo queda completo.

Almacenarlo en una blockchain.

En cuanto a el objetivo del desarrollo de la aplicación distribuida también resultado exitoso como se muestra en la sección 2.3 donde se detalla el desarrollo de la aplicación distribuida.

El objetivo de crear un cliente que registre la información proveniente de ROS en la *blockchain* se cumple gracias al nodo de ROS `blockchain_client.py` el cual registra en la *blockchain* a través de la aplicación distribuida la actividad del usuario.

El objetivo planteado sobre crear una interfaz para acceder a la información de la *blockchain* se cumple gracias a la aplicación web que lee la información de la *blockchain* para graficar la información del usuario planteada en la sección 2.3.

Desarrollo sobre un sistema de agentes.

El objetivo de desarrollarlo sobre un sistema de agentes también fue realizado pues con la excepción de la aplicación distribuida y la aplicación web de visualización todos los demás componentes del desarrollo son agentes.

También es evidente la virtud de haber utilizado un sistema de agentes ya que se puede observar con la excepción de la aplicación distribuida y la aplicación web de visualización todos y cada uno de los objetivos planteados en la sección 1.2 se pueden referir de manera individual a un agente y cada agente cumple con un objetivo en particular de forma tal que si hubiera cambios en un objetivo entonces eso se traduce directamente a hacer cambios en un solo nodo.

Referencias

- [1] F. Chollet *et al.* *Keras*. <https://keras.io>. 2015.
- [2] *cv_camera* - *ROS Wiki*. URL: http://wiki.ros.org/cv_camera.
- [3] *ethereum/web3.js: Ethereum JavaScript API*. URL: <https://github.com/ethereum/web3.js/>.
- [4] *GOAL - GOAL Confluence*. URL: <https://goalapl.atlassian.net/wiki/spaces/GOAL/overview?mode=global>.
- [5] M. A. Hmani y D. Petrovska-Delacretaz. «State-of-the-art face recognition performance using publicly available software and datasets». En: *2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. IEEE, mar. de 2018. DOI: 10.1109/atsip.2018.8364450.
- [6] <https://github.com/ethereum/web3.py>. URL: [ethereum/web3.py:%20A%20python%20interface%20for%20interacting%20with%20the%20Ethereum%20blockchain%20and%20ecosystem..](https://github.com/ethereum/web3.py)
- [7] <https://www.chartjs.org/>. URL: [Chart.js%20%7C%20open%20source%20HTML5%20Charts%20for%20your%20website.](https://www.chartjs.org/)
- [8] *Jade Site | Java Agent DEvelopment Framework*. URL: <https://jade.tilab.com>.
- [9] *jason download | SourceForge.net*. URL: <https://sourceforge.net/projects/jason/>.
- [10] F. Lecumberri. *Robot Operating System Graph Visualization Architecture Documenter*. URL: <https://github.com/fcolecumberri/ROSGraphvizDoc>.
- [11] Y. Lecun, L. Bottou, Y. Bengio y P. Haffner. «Gradient-based learning applied to document recognition». En: *Proceedings of the IEEE* 86.11 (1998), págs. 2278-2324. DOI: 10.1109/5.726791.
- [12] Y. Lu. «The blockchain: State-of-the-art and research challenges». En: *Journal of Industrial Information Integration* (abr. de 2019). DOI: 10.1016/j.jii.2019.04.002.
- [13] Martn Abadi *et al.* *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.
- [14] Nokia Corp. *Qt : cross-platform application and UI framework*. 2012. URL: <http://qt.nokia.com/>.
- [15] *OpenCV: Cascade Classifier*. URL: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html.
- [16] *opencv_apps* - *ROS Wiki*. URL: http://wiki.ros.org/opencv_apps.
- [17] *opencv_apps/Face Documentation*. URL: http://docs.ros.org/kinetic/api/opencv_apps/html/msg/Face.html.

- [18] *opencv_apps/FaceArrayStamped Documentation*. URL: http://docs.ros.org/kinetic/api/opencv_apps/html/msg/FaceArrayStamped.html.
- [19] *opencv_apps/Rect Documentation*. URL: http://docs.ros.org/kinetic/api/opencv_apps/html/msg/Rect.html.
- [20] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler y A. Ng. «ROS: an open-source Robot Operating System». En: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*. Kobe, Japan, mayo de 2009.
- [21] *rosvbag - ROS Wiki*. URL: <http://wiki.ros.org/rosvbag>.
- [22] *rqt - ROS Wiki*. URL: <http://wiki.ros.org/rqt>.
- [23] *SARL Agent-Oriented Programming Language*. URL: <http://www.sarl.io/>.
- [24] *sensor_msgs/Image Documentation*. URL: http://docs.ros.org/kinetic/api/sensor_msgs/html/msg/Image.html.
- [25] Y. Shoham. *Elsevier ARTINT 931 Agent-oriented programming*. 1991.
- [26] *std_msgs/Header Documentation*. URL: http://docs.ros.org/kinetic/api/std_msgs/html/msg/Header.html.
- [27] *trufflesuite/ganache: Personal blockchain for Ethereum development*. URL: <https://github.com/trufflesuite/ganache>.
- [28] *trufflesuite/truffle: A tool for developing smart contracts. Crafted with the finest cacaos*. URL: <https://github.com/trufflesuite/truffle>.
- [29] P. Viola y M. Jones. «Rapid object detection using a boosted cascade of simple features». En: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. IEEE Comput. Soc, 2001. DOI: 10.1109/cvpr.2001.990517.

Sistema de recomendación de música basado en etiquetado social

Alberto Moro Calvo, Vivian López Batista, Gabriel Villarrubia González

¹Departamento de Informática y Automoción, Universidad de Salamanca
Plaza Caídos, s/n, 37008, Salamanca, España
{alberto_moro, vivian, gvg}@usal.es

Resumen En la actualidad, los servicios de *streaming* están en auge, ya que permiten el acceso a la gran mayoría de música, a pesar de la enorme cantidad existente. Una parte fundamental de estos servicios son los sistemas de recomendación, ya que facilitan la experiencia a los usuarios, permitiendo proporcionar un conjunto de canciones o artistas que más les gustan de forma más rápida, a partir de un conjunto muy grande de información. Por otra parte, el uso de las redes sociales crece exponencialmente con los años y son una valiosa fuente de información, reuniendo gustos, opiniones y subjetividad de una gran cantidad de usuarios de todo el mundo. A través del procesamiento del lenguaje natural en redes sociales como Twitter, se pueden extraer características que identifican los gustos musicales de cada usuario, mediante el *hashtag* #nowplaying. Este trabajo propone la creación de un sistema de recomendación de música basado en etiquetado social, con el uso del algoritmo *Single Value Decomposition*. La adición de información adicional como etiquetas o *tags* extraídas del procesamiento del lenguaje natural mediante el Análisis Semántico Latente en Twitter, permite mejorar los resultados de un sistema de recomendación de filtrado colaborativo básico, generando una lista de artistas afines a los gustos de un usuario, permitiendo mayor libertad de expresión al usuario y obtener recomendaciones de artistas a partir de una red social como Twitter.

Keywords: Sistema de recomendación, etiquetado social, Filtrado Colaborativo, Procesamiento del Lenguaje Natural, Single Value Decomposition, Latent Semantic Analysis, Twitter

1. Introducción

A lo largo de los últimos años, el volumen de información generado en Internet ha crecido exponencialmente. La aparición de las redes sociales ha aumentado significativamente la información subjetiva de los usuarios con la creación de sitios donde opinar, colaborar, crear y compartir contenido. Como consecuencia de ello, tanto la web como las redes sociales se han convertido en un ámbito usual para la recuperación y el procesamiento de la información, con el fin de abstraer y recuperar solo los datos necesarios. Toda esta información es muy difícil de procesar, debido a que el conjunto de datos es muy elevado, por lo que lograr obtener palabras claves sobre un texto de tan grandes dimensiones es todo un desafío para el procesamiento del lenguaje natural.

Las redes sociales como Twitter¹ y su limitación de caracteres en sus publicaciones facilitan este procesamiento, estableciendo un límite a la información subjetiva de un usuario. Otras características de Twitter como la adición de *hashtags* permite filtrar contenidos o tópicos que nos interesan, excluyendo información irrelevante. Por ejemplo, el *hashtag* #nowplaying es uno de los cuatro *hashtags* más utilizados en Twitter. Este *hashtag* suele estar incluido en publicaciones relacionadas con

¹ Twitter: <https://twitter.com/>

la música, e incluso es añadido automáticamente al publicar un *tweet* de alguna red social como Last.fm² sobre alguna canción o artista escuchado.

Los sistemas de recomendación son una herramienta fundamental en el mundo actual para que el usuario tenga a disposición los productos o recursos más afines a sus gustos, sin la necesidad de analizar la inmensa cantidad de información que existe actualmente.

Uno de los sistemas de recomendación más innovadores son los de etiquetado social, ya que permiten a los usuarios identificar de manera libre y subjetiva los recursos mediante palabras claves o etiquetas. El poder del etiquetado reside en la posibilidad que los usuarios tienen de calificar o categorizar libremente un recurso, sin una cuantificación o jerarquía predeterminada. Las etiquetas crean una conexión entre los usuarios y los recursos, de forma que los usuarios que etiqueten los mismos artistas de manera similar, tendrán gustos similares, al igual que los recursos que sean etiquetados de manera parecida serán muy similares.

Este trabajo propone un sistema de recomendación de música basado en etiquetado social a partir de redes sociales. Con la ayuda de algunos servicios de *streaming*, la asignación de etiquetas y mediante el procesamiento del lenguaje natural en Twitter, se modelará una folksonomía que permitirá recomendar a un usuario de Twitter los artistas más afines a sus gustos.

1.1. Objetivos

En este trabajo se analizarán diferentes métodos para realizar recomendaciones de música basadas en etiquetado social. El objetivo principal será analizar los comentarios de los usuarios en redes sociales como Twitter, mediante el *hashtag* #nowplaying. Esta información será utilizada para realizar predicciones de artistas similares a sus gustos. El objetivo final será la elaboración de una lista de artistas afines a los gustos de un usuario.

A este objetivo principal se le añaden otros muy importantes, como:

- **Revisión de los sistemas de recomendación:** se definirán los sistemas de recomendación y se expondrán los diferentes tipos de sistemas junto a sus características, ventajas y desventajas.
- **Revisión de los sistemas de recomendación de etiquetado social:** se realizará una breve descripción de estos sistemas, destacando las ventajas que propone respecto a otros sistemas de recomendación.
- **Revisión del procesamiento del lenguaje natural:** se analizará el procesamiento del lenguaje que se llevará a cabo en los comentarios de los usuarios de Twitter, con el fin de obtener un vector de palabras, utilizando éstas como etiquetas en el sistema de recomendación de etiquetado social.
- **Aplicación del sistema de recomendación de etiquetado social a los comentarios de la red social Twitter:** la implementación del sistema de recomendación de música basado en etiquetado social se basará en la red social Twitter, obteniendo comentarios de dicha plataforma y generando una lista de artistas afines a los gustos de cada usuario que haya realizado comentarios en Twitter con el *hashtag* #nowplaying.
- **Análisis de algoritmos de recomendación:** se aplicarán diferentes algoritmos de recomendación, evaluando y comparando sus resultados.
- **Análisis de los resultados obtenidos:** se aplicarán métricas de evaluación de resultados al sistema propuesto, comparando estos resultados con otros sistemas de recomendación existentes.

² Last.fm: <https://www.last.fm/>

La memoria está estructurada de la siguiente forma: tras la introducción y la definición de los objetivos, en la sección 2 se elaborará una revisión del estado del arte sobre los sistemas de recomendación, centrándose en los basados en etiquetado social y sobre el procesamiento del lenguaje natural. En la sección 3 se describen las herramientas, la arquitectura y el prototipo del sistema implementado. En la sección 4, se describe el conjunto de datos utilizados y las métricas utilizadas en la evaluación del sistema, explicando los resultados obtenidos. Por último, en la sección 5 se sintetizan las conclusiones sobre el trabajo y se indicarán las posibles mejoras a incorporar al sistema propuesto como líneas de trabajo futuras.

2. Estado del Arte

2.1. Sistemas de Recomendación

Los sistemas de recomendación[21] [20] son herramientas que permiten proporcionar al usuario un conjunto de ítems o información de su interés, a partir de un conjunto muy grande de información.

En la actualidad, los sistemas de recomendación resultan muy útiles en la Web y son ampliamente utilizados. La información que existe en la actualidad es tan inmensa que se necesitan herramientas que permitan proporcionar al usuario los productos en los que puede estar interesado de manera personalizada. El proceso de elección del ítem a escoger cuando el usuario sabe lo que quiere buscar es rápido y sencillo. El problema se encuentra cuando el usuario no sabe realmente lo que quiere elegir y debe explorar toda la información, lo cual consume una gran cantidad de tiempo e incluso que el usuario acabe perdiendo el interés por encontrarlo. Es ahí donde los sistemas de recomendación forman una parte esencial del sistema.

2.2. Filtrado Colaborativo

En la vida cotidiana, la gente confía en las recomendaciones de otras personas por medio de palabras, cartas de referencia, informes de noticias, comentarios, o encuestas. Los sistemas de recomendación asisten y aumentan este proceso social con el fin de ayudar a las personas a encontrar los libros, artículos, películas, música, restaurantes o tiendas más interesantes para ellos.

La suposición fundamental del filtrado colaborativo [25] consiste en que si un usuario A y otro usuario B califican N ítems de manera similar o tienen comportamientos similares, en consecuencia, calificarán o actuarán de manera similar sobre otros ítems. Las técnicas de filtrado colaborativo utilizan una base de datos de preferencias para los ítems por usuarios para predecir temas o productos adicionales a un nuevo usuario que podría gustarle. En un escenario típico de filtrado colaborativo, hay una lista de usuarios y cada usuario tiene una lista de ítems que el usuario ha calificado.

Enfoques Basados en Memoria o Basados en Vecinos Más Cercanos. Los algoritmos de filtrado colaborativo basados en memoria[6] [10] [13] [8] utilizan la totalidad o una muestra de la base de datos de *user-ítem* para generar una predicción. Cada usuario es parte de un grupo de personas con intereses similares, por lo que se puede obtener una predicción de preferencias de un nuevo usuario a partir de sus "vecinos".

Las recomendaciones Top-N permiten recomendar un conjunto N de ítems de mayor interés para un determinado usuario. Los algoritmos de recomendación Top-N basados en usuarios[6] [5] primero identifican los vecinos más cercanos al usuario activo utilizando un algoritmo de cálculo de similitud, como la correlación de Pearson o la medida del coseno, por ejemplo. Después de haber

obtenido los k usuarios más similares, se agregan a la matriz *user-item* para identificar un conjunto de ítems adquiridos por el grupo junto a su frecuencia. Tras obtener este conjunto, las técnicas de filtrado colaborativo basadas en usuarios recomiendan los ítems más frecuentes en el conjunto que el usuario activo aún no ha adquirido. Los algoritmos de recomendación Top-N basados en usuarios tienen algunas limitaciones relacionadas con la escalabilidad y el rendimiento en tiempo real.

2.3. Filtrado Basado en Contenido

A los sistemas de recomendación basados en contenido [21] [15] [12] [18] [1] [19] también se les conoce como sistemas de recomendación no colaborativos. En ellos, el sistema recomienda productos similares a los productos que le ha gustado al usuario anteriormente. Para comparar la similitud entre ítems se buscan palabras claves del ítem que calificó el usuario. Si se recomiendan continuamente ítems similares a los vistos anteriormente se presenta un problema conocido como sobre-especialización. Por ejemplo, si en un sistema de recomendación de libros un usuario solo valora libros de género romántico, resultará imposible que el sistema le recomiende libros de comedia, de miedo o de historia.

2.4. Sistemas de Recomendación Híbridos

Los sistemas colaborativos y los basados en contenido tienen un carácter complementario, ya que cada tipo de sistema tiene sus propias debilidades o desventajas y éstas son resueltas por otros tipos de sistemas de recomendación. Los sistemas de recomendación híbridos [2] [4] pretenden unir las ventajas de cada tipo con el fin de realizar mejores recomendaciones.

En los sistemas de recomendación híbridos, los usuarios se mantienen según el análisis de los contenidos de los ítems, y directamente se comparan esos perfiles para determinar las similitudes entre usuarios para una recomendación colaborativa.

2.5. Sistemas de Recomendación de Etiquetado Social

Los sistemas de recomendación de etiquetado social (Social Tagging Systems) [16] [24] [28] toman al usuario común como un pilar fundamental, al que se le permite compartir metadatos ligeros en forma de palabras clave de libre elección, llamadas etiquetas, además de publicar y editar recursos. La exposición de los usuarios tanto a las etiquetas como a los recursos crea un detonante fundamental para la comunicación y el intercambio, reduciendo así las barreras a la cooperación y contribuyendo a la creación de estructuras de conocimiento ligeras y colaborativas conocidas como folksonomías, representadas como un tensor de tercer orden o como un hipergrafo.

Las etiquetas son una forma de agrupar el contenido por categorías para facilitar su visualización por temas y las folksonomías son las estructuras subyacentes de los sistemas de etiquetado social y resultado de la práctica de crear estas etiquetas de manera colaborativa y categorizar el contenido. Este es un enfoque para organizar y ayudar a los usuarios a encontrar el contenido que les interesa, por lo que, como se ha comentado anteriormente, se aumenta la relación binaria *usuario-ítem* y se convierte en una ternaria *usuario-ítem-etiqueta*.

Sin embargo, las etiquetas, al ser proporcionadas de manera gratuita y voluntaria, pueden provocar problemas por la falta de voluntad de etiquetado o por el vocabulario tan divergente de los usuarios. Estas etiquetas representan información adicional sobre los recursos que pueden aumentar el rendimiento del recurso del sistema de recomendación. A continuación, se definirán las folksonomías y sus estructuras de datos.

Folksonomías. Las folksonomías[16] [7] [9] son el resultado del etiquetado libre y personal de información y la principal estructura subyacente a los sistemas de recomendación basados en etiquetado social. A diferencia de las taxonomías, las folksonomías no están sometidas a un esquema predefinido de categorías, pudiendo permitir categorizaciones flexibles y evolutivas. Además, categorizan grandes espacios de información ya que se determinan a partir de la colaboración de comunidades de usuarios.

Una folksonomía[11] es una tupla $F := (U, T, R, Y)$ donde U , T y R son conjuntos finitos no vacíos cuyos elementos son usuarios, etiquetas y recursos respectivamente e Y es una relación ternaria entre ellos cuyos elementos se denominan asignaciones de etiquetas. Los usuarios normalmente se describen por su ID y las etiquetas por cadenas arbitrarias, mientras que los recursos dependerán del tipo de sistema (artistas o canciones en Spotify, productos en Amazon, películas en Netflix, URLs o referencias en BibSonomy, etc). Los datos de folksonomía pueden ser representados de diferentes maneras, y cada representación puede conducir a diferentes algoritmos de recomendación.

Sistema de Etiquetado Social: Filtrado Colaborativo. Como ya se ha explicado anteriormente en la Sección (2.2), el filtrado colaborativo encuentra personas con gustos similares al usuario y recomienda los ítems que dichas personas prefieren. El filtrado colaborativo trabaja con un tensor de segundo orden, por lo que es necesario reducir la dimensionalidad de la relación ternaria Y de la folksonomía para aplicar este algoritmo en un sistema de recomendación de etiquetado social. Para un filtrado colaborativo basado en usuarios, se consideran dos matrices, generadas a partir de las proyecciones de la folksonomía: $\pi_{U,R}Y \in \{0, 1\}^{|U| \times |R|}$ con $(\pi_{U,R}Y)_{u,r} := 1$ si existe $t \in T$ para $(u, t, r) \in Y$ y 0 en caso contrario, y $\pi_{U,T}Y \in \{0, 1\}^{|U| \times |T|}$ con $(\pi_{U,T}Y)_{u,t} := 1$ si existe $r \in R$ para $(u, t, r) \in Y$ y 0 en caso contrario (Fig. 1), siendo u un usuario, t una etiqueta y r un recurso. Estas proyecciones preservan la información del usuario, conduciendo al sistema de recomendación a la ocurrencia de ítems o etiquetas respecto a cada usuario. Se aplicará el algoritmo de vecinos más cercanos a un usuario considerando los ítems o las etiquetas en el filtrado colaborativo.

Tras el cálculo del vecindario, se extrae el conjunto $T(u, t)$ como el conjunto de ítems recomendados para un usuario u , una etiqueta t y un subconjunto de usuarios que denotan el vecindario, según la ecuación (Ec. 1):

$$T(u, t) := \arg \max_{r \in R} \sum_{v \in N_u^k} \text{sim}(x_u, x_v) \delta(v, t, r) \quad (1)$$

siendo $\delta(v, t, r) := 1$ si $(v, t, r) \in Y$ y 0 en caso contrario.

El filtrado colaborativo tiene problemas de escalabilidad, ya que toda la matriz necesita ser guardada en memoria. En los sistemas de recomendación de etiquetado social, puede que el problema incremente si se necesita más de una proyección de la folksonomía para realizar los cálculos. Este sistema se basa en tres pasos a la hora de realizar predicciones:

- Cálculo de proyecciones: es necesario crear las proyecciones que sean necesarias a partir de las relaciones ternarias Y de la folksonomía definida.
- Cálculo de vecindarios: será necesario calcular la similitud del usuario activo respecto a todos los usuarios del conjunto de datos y ordenarlos con el fin de determinar los k vecinos más cercanos.
- Calcular las recomendaciones: Para predecir los Top-N recursos más afines a los gustos del usuario activo es necesario contar el número de coincidencias de etiquetas/ítems con los vecinos más cercanos, calcular el peso de cada coincidencia según la similitud con el vecino y, por último, ordenar dichas etiquetas/ítems según sus pesos.

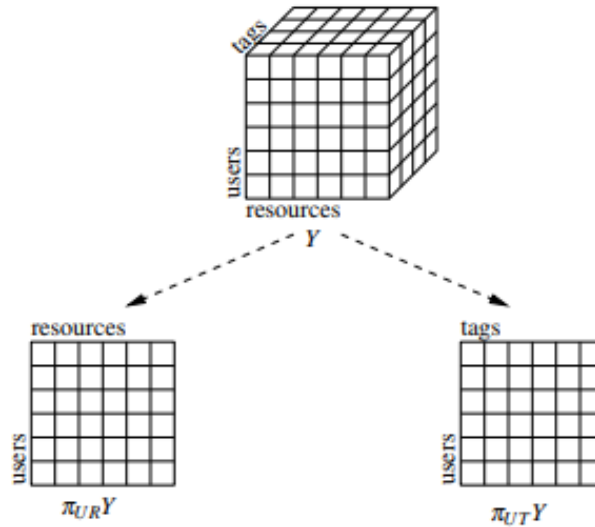


Figura 1. Proyecciones de la folksonomía en el espacio de los recursos y etiquetas del usuario [16].

2.6. Procesamiento del Lenguaje Natural

El avance de la tecnología y la cantidad de información generada en lenguaje natural ha crecido de manera exponencial, provocando un gran volumen de información. Para que toda esa información tenga sentido y se transforme en conocimiento, los textos y documentos tienen que ser procesados. A partir de este problema, se creó un nuevo campo de estudio denominado Procesamiento del Lenguaje Natural (PLN), que permite unir los campos de la inteligencia artificial y lingüística aplicada con el objetivo de solventar la comprensión y el procesamiento automático del lenguaje natural, haciendo posible la interacción entre personas y máquinas.

El primer paso para que el texto pueda ser procesado por los algoritmos de recomendación es el preprocesamiento. Las técnicas de preprocesamiento permiten la eliminación de atributos irrelevantes, redundantes o que no aportan información semántica. Entre ellas destacan:

- **Normalización:** la normalización permite una homogeneidad en el texto para lograr una mayor eficacia en el procesamiento del lenguaje, eliminando términos que no aportan ningún tipo de información. En este trabajo se realizarán algunas normalizaciones como la conversión a letra minúscula, eliminación de signos de puntuación, caracteres especiales, emoticonos y enlaces a páginas web o vídeos.
- **Tokenización:** creación de un array de palabras al separar un texto en pequeños fragmentos, denominados *tokens*.
- **Eliminación de *stopwords*:** los *stopwords* o palabras vacías es el nombre que reciben las palabras sin significado como artículos, pronombres, preposiciones, etc. que serán filtradas en el PLN.
- **Lematización:** la lematización es un proceso lingüístico que consiste en hallar el lema correspondiente a una palabra, facilitando la eficacia de la indexación y la consiguiente recuperación.

2.7. Extracción de Características

Las palabras o términos son almacenadas en un vector de palabras. En ocasiones, el tamaño de los vectores es enorme ya que tendrá tantos elementos como vocabulario tenga el conjunto de datos y, por lo tanto, puede provocar grandes problemas de escalabilidad.

Las características son términos relevantes que se extraen de un texto y nos permite encontrar patrones, con el fin de obtener palabras clave utilizadas en el proceso de recomendación del sistema. Un conjunto de documentos poseen palabras o características irrelevantes o redundantes a la hora del proceso de recomendación. La eliminación de características irrelevantes proporciona una reducción de dimensionalidad del espacio de datos, mejorando la escalabilidad y eficiencia de los algoritmos de recomendación. Este tipo de reducción de dimensionalidad minimiza el problema de *overfitting*, desechando características que no contribuyen al sistema de recomendación.

Para reducir la dimensionalidad de los vectores y la extracción de características, se utilizan técnicas como Single Value Decomposition (SVD).

Single Value Decomposition. LSI (Latent Semantic Indexing) [23] es una técnica de reducción de dimensionalidad para resolver problemas de sinonimia y polisemia. Dada una matriz de frecuencias término-documento, LSI se utiliza para construir dos matrices de dimensionalidad reducida. Estas matrices representan atributos latentes de los términos, reflejados por su aparición en los documentos, y de documentos, reflejados por los términos que aparecen en ellos. LSI utiliza SVD (Single Value Decomposition) como algoritmo de factorización de matrices. Las dimensiones ortogonales reducidas, resultantes de SVD son menos ruidosas que los datos originales y captura las asociaciones latentes entre términos y documentos.

SVD es una técnica bien conocida de factorización matricial que factoriza una matriz R $m \times n$ en tres matrices, según la ecuación (Ec. 2):

$$R = U \cdot S \cdot V' \quad (2)$$

donde U y V son matrices ortogonales de tamaño $m \times r$ y $n \times r$ respectivamente y r es el “rango” de la matriz R . S es una matriz diagonal de tamaño $r \times r$. Todas las entradas de la matriz S son positivas y ordenadas de mayor a menor. SVD permite obtener una aproximación de menor rango de la matriz R original, según la norma de Frobenius. Si se reduce la matriz S para obtener k valores, se obtiene la matriz S_k , $k < r$, por lo que, si se reducen de manera adecuada las matrices U y V , entonces se puede obtener la matriz $R_k = U_k \cdot S_k \cdot V_k'$, obteniendo una matriz con las k características más importantes, ya que están ordenadas de mayor a menor, y obviando el resto de ellas, obteniendo una matriz reducida.

Tf-Idf. Significa la frecuencia de ocurrencia del término en la colección de documentos [26]. La puntuación TF-IDF es una medida estadística utilizada para evaluar la importancia de una palabra para un documento en una colección. La importancia aumenta proporcionalmente al número de veces que una palabra aparece en el documento, e inversamente proporcional a la frecuencia de la palabra en la colección. La combinación de la técnica Tf-Idf junto a SVD se conoce como Análisis Semántico Latente (LSA).

Análisis Semántico Latente. El LSA [17] [14] o análisis semántico latente es una técnica de extracción de información y relaciones inferidas del uso de palabras contextual. Toma como entrada

sólo el texto en bruto, analizando las palabras que son separadas en oraciones. Es un tipo de análisis computacional que permite determinar y cuantificar la similitud de significado entre piezas textuales pertenecientes a un mismo dominio de conocimiento, basándose en un algoritmo matemático.

El primer paso es representar el texto como una matriz en la que cada línea representa una palabra única y cada columna representa un documento u oración. En el sistema implementado, cada “documento” estará definido por un *tweet*, lo cual permite un límite de caracteres. Cada celda contiene la frecuencia con la que la palabra de su fila aparece en el documento denotado por su columna. A continuación, se calcula un peso para cada celda en función de la importancia de la palabra en el documento y el grado de información que ésta aporta.

Posteriormente, LSA aplica SVD a la matriz para detectar patrones entre los términos de la colección. SVD descompone la matriz en el producto de otras tres matrices, como se ha explicado anteriormente en (Sec. 2.7) para reducir la dimensionalidad. El análisis semántico latente, además de aplicar SVD, calcula el valor TF-IDF de cada término para mejorar los resultados, de forma que los términos más característicos a los documentos consiguen un valor mayor. Tras obtener los valores de R_k , se calcula la similitud entre los documentos mediante cualquier medida. En este trabajo se ha utilizado la similitud coseno.

El LSA permite obtener una reducción de la dimensionalidad de la matriz documentos-términos y soluciona algunos problemas mediante el cálculo TF-IDF como la polisemia o la sinonimia en una colección. En este sistema, los documentos hacen referencia a los diferentes *tweets* publicados por los usuarios.

3. Sistema Propuesto

En este apartado se describirán las herramientas utilizadas, el sistema desarrollado de manera detallada, explicando brevemente cada uno de los métodos que se han ido desarrollando para obtener el sistema de recomendación basado en etiquetado social. Este sistema de recomendación obtendrá los artistas más afines a los gustos de un usuario de Twitter, extrayendo información de éste a través de sus *tweets* publicados con el *hashtag* #nowplaying.

3.1. Herramientas

El sistema de recomendación ha sido implementado en Python 3.7³, un lenguaje de programación interpretado muy utilizado en el *machine learning* y en *deep learning*.

Para mostrar los resultados de los artistas más afines a los gustos de un usuario, se ha creado una aplicación web, cuyo nombre es “ArTWists Recommender”, en la que el *frontend* está desarrollado en Angular 7.0⁴, mientras que el *backend* estará implementado en Python como se ha explicado anteriormente, usando Flask⁵ como *framework* entre ambos para la creación de una aplicación web.

3.2. Extracción de datos de Twitter

Para crear y obtener los perfiles de usuario, lo primero que se debe hacer es extraer los datos de la cuenta de Twitter del usuario. Para ello, es necesario haber creado una aplicación de Twitter

³ Python: <https://www.python.org/>

⁴ Angular: <https://angular.io/>

⁵ Flask: <http://flask.pocoo.org/>

y obtener las cuatro claves secretas que permiten obtener datos de *streaming* de Twitter o de un usuario en concreto. Para la extracción de datos, se hará uso de la API Tweepy⁶, que permite, a partir de las cuatro claves secretas, extraer los *tweets* en modo *streaming* o en modo histórico. En este caso, se extraerán todos los *tweets* de un usuario concreto, filtrados por el *hashtag* #nowplaying.

A partir del conjunto de *tweets* obtenidos del usuario, el siguiente paso a realizar será el preprocesamiento de cada *tweet*. El primer paso en el preprocesamiento será la tokenización del *tweet*, convirtiendo éste en un array de palabras. Las técnicas de preprocesamiento permiten la eliminación de atributos irrelevantes, por lo que el siguiente paso será eliminar los denominados *stopwords*. En este conjunto se han incluido URLs y números, aparte de los ya por defecto artículos, pronombres, preposiciones, etc. El último paso del preprocesamiento será la lematización de las palabras restantes. Este proceso facilita la recuperación de información y por lo tanto, la obtención de etiquetas para un usuario. Todo este preprocesamiento del texto se llevará a cabo mediante una librería denominada NLTK (Natural Language Toolkit)⁷, una herramienta para trabajar con el lenguaje humano y obtener información relevante a partir de él.

3.3. Sistema de recomendación basado en etiquetado social

El sistema de recomendación desarrollado está basado en etiquetado social. Este enfoque está basado en Twitter, el cual carece de servicio de *streaming* pero tiene un potente componente social. Este sistema utilizará los *tweets* como modo de extracción de características y atributos del usuario en forma de etiquetas, creando los perfiles de usuario. Estas etiquetas obtenidas de Twitter serán utilizadas con el fin de obtener una folksonomía sobre un segundo servicio de *streaming* como Last.fm. Por último, se realizarán predicciones de los n artistas más afines a los gustos del usuario activo mediante un algoritmo de recomendación.

Perfiles de Usuario. La creación de perfiles de usuario es una parte fundamental de los sistemas de recomendación, los cuales representan las preferencias e intereses de los usuarios. Los perfiles de usuario permiten obtener la información necesaria sobre un usuario a partir de diferentes conjuntos de datos.

En este caso, para la creación de perfiles de usuario, se necesitan dos conjuntos de entrada, los perfiles de ítems basados en etiquetas y los perfiles de usuario basados en valoraciones. A partir de dichos datos de entrada se obtienen los perfiles de usuario basados en etiqueta como datos de salida.

El conjunto de perfiles de ítems se puede extraer a través del análisis de *tweets* sobre los artistas o canciones o bien mediante APIs como Last.fm. Esto permite que los usuarios etiqueten las canciones o artistas libremente. Los perfiles de los ítems se obtienen mediante la proyección de la folksonomía en la dimensión de los usuarios, dando lugar a una matriz M^A , donde cada fila es el perfil del ítem P_i^A y donde cada valor $w_{a,t}$ de dicha matriz es el sumatorio de los valores para cada usuario. El conjunto de perfiles de usuario se extrae a través de cualquier plataforma *streaming*, como por ejemplo Last.fm, donde se puede obtener si un usuario ha escuchado una canción o artista. El conjunto de perfiles de usuarios se representa como una matriz M^U donde cada fila representa el perfil P^U de un usuario $u \in U$.

⁶ Tweepy: <https://www.tweepy.org/>

⁷ Natural Language Toolkit: <https://www.nltk.org/>

El objetivo es la obtención de unos perfiles de usuario basados en etiquetas. Para ello, se creará una nueva folksonomía a partir de las matrices M^A y M^U , donde cada relación ternaria $\langle u, a, t, y \rangle$ se calcula mediante la siguiente ecuación (Ec. 3):

$$y_{u,a,t} = w_{a,t} \cdot r_{u,a}, w_{a,t} \in M^A, r_{u,a} \in M^U \quad (3)$$

De esta folksonomía $F := (U, A, T, Y)$, donde U son los usuarios del sistema, A los artistas y T las etiquetas del etiquetado social, se pueden obtener los perfiles de usuarios basados en etiquetas (Fig. 2), mediante la proyección de la folksonomía sobre la dimensión de los artistas, según la ecuación (Ec. 4):

$$w_{u,t} = \sum_{a \in A^*} y_{u,a,t} \quad (4)$$

donde $a \in A^*$ son los artistas del conjunto de datos e $y_{u,a,t} \in Y^*$ es la relación entre el usuario u , el artista a y el *tag* t , cuyo valor será el número de veces que esa etiqueta t haya sido asignada al artista a por el usuario u . Este perfil de usuario final se representa como una matriz $M^{U,T}$ donde cada fila es el perfil P_u de un usuario $u \in U$.

Todas las matrices obtenidas han sido normalizadas, de manera que el valor de cada elemento de la matriz será proporcional a su frecuencia y la suma total de la fila P_x sea igual a 1. De este modo, los valores se corresponderán a la frecuencia en la que un usuario ha utilizado dicha etiqueta.

Algoritmo de Recomendación Basado en Etiquetado Social. Tras obtener todos los datos disponibles, se procederá a realizar la predicción de los Top-N artistas más afines a los gustos del usuario activo. Para ello, la metodología elegida es el filtrado colaborativo, explicado en (Sec. 2.5). El algoritmo kNN (*k Nearest Neighbours* o *k* vecinos más cercanos), utilizado en la implementación del sistema, encontrará los *k* usuarios más cercanos al usuario activo, es decir, los *k* usuarios que tengan unos gustos más similares al usuario, según las etiquetas utilizadas.

Se tendrán dos conjuntos de entrada, el vector de etiquetas del usuario activo y la matriz $M^{A,T}$, donde cada fila P_u será el vector de etiquetas del usuario u . El vector de etiquetas del usuario activo será el obtenido a partir de la extracción, análisis y procesamiento de sus *tweets* con el *hashtag* #nowplaying, explicado anteriormente. La matriz $M^{A,T}$ será la obtenida anteriormente en la creación de perfiles de usuario.

Reducción de dimensionalidad La mayoría de los usuarios de Twitter suelen publicar una gran cantidad de *tweets* y cada *tweet* suele contener más de una palabra clave que lo identifique. Es por ello, que el número de etiquetas que se le asocian a un usuario puede ser demasiado grande, y esto afectaría a la escalabilidad del sistema, ya que se crearía una matriz tan grande que empeorarían los resultados del algoritmo de recomendación. Por lo tanto, para reducir la dimensionalidad del vector de etiquetas que un usuario posee, se ha aplicado LSA, mediante la librería *Gensim*⁸ de Python, con el fin de obtener las palabras más características del conjunto de etiquetas del vector.

A partir de este vector de etiquetas del usuario activo, se calcula la similitud con cada perfil de usuario del conjunto de datos, comparando los vectores de etiquetas mediante la medida del coseno, ya que es la medida que mejores resultados ha tenido, obteniendo los *k*-vecinos más cercanos. Durante este proceso se llevará a cabo, mediante la librería de Python *Fuzzywuzzy*⁹, un proceso de

⁸ Gensim: <https://radimrehurek.com/gensim/>

⁹ FuzzyWuzzy: <https://github.com/seatgeek/fuzzywuzzy>

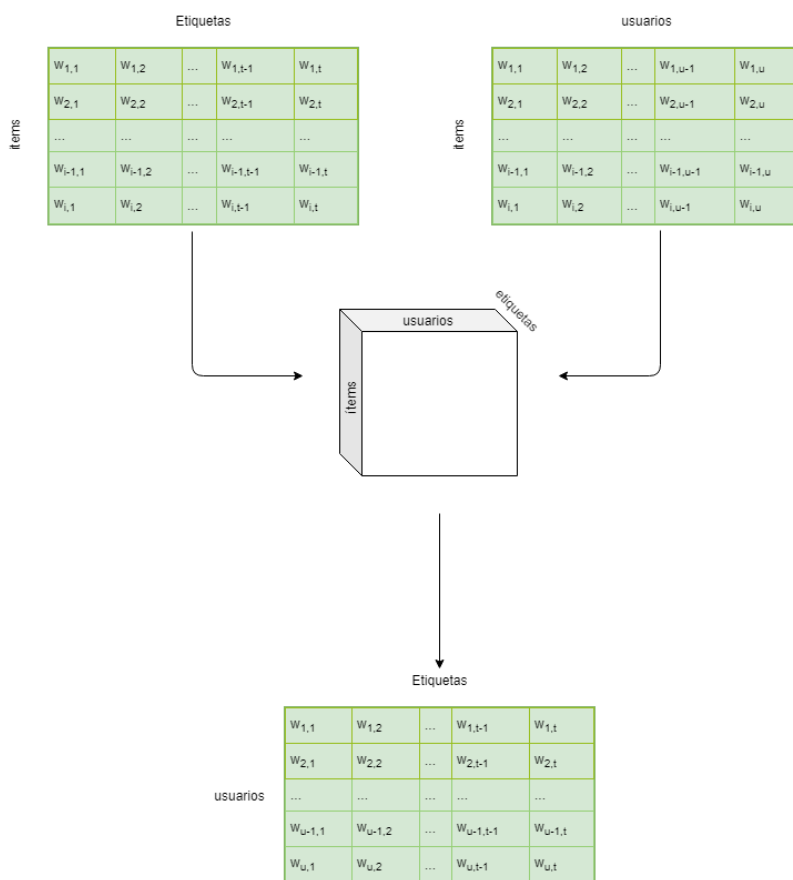


Figura 2. Proyección de la folksonomía sobre la dimensión de los ítems.

fuzzy matching [27], una técnica que permite mejorar el proceso de comparación de etiquetas en un conjunto de datos, permitiendo encontrar resultados de comparación entre dos etiquetas que no son completamente iguales pero que tienen un gran porcentaje de similitud.

Se formará un vecindario, donde los vecinos más cercanos serán los que tengan unos gustos más similares al usuario activo. A partir del cálculo de similitud entre los usuarios, se seleccionan los k usuarios con mayor coeficiente de similitud con el usuario activo. Por último, el conjunto de ítems recomendados (Fig.3), se obtiene a partir de una combinación ponderada de las calificaciones del subconjunto de vecinos seleccionado previamente, mediante la ecuación (Ec. 5):

$$P_{u,a} = \frac{\sum_{k \in K} sim_{u,k} \cdot r_{k,a}}{\sum_{k \in K} r_{k,a}} \quad (5)$$

donde $P_{u,a}$ es la predicción para el usuario u sobre el artista a , $sim_{u,k}$ es la similitud entre el usuario activo u y su vecino k , K es el subconjunto de k -usuarios más similares y $r_{k,a}$ es la valoración del usuario vecino k al artista a .

De esta forma, este sistema de recomendación basado en etiquetado social tiene una gran escalabilidad respecto a un sistemas de filtrado colaborativo básico, utilizando las etiquetas como herramienta de libertad y subjetividad, a la vez que permite una reducción de dimensionalidad de la matriz del conjunto a entrenar mediante LSA, obteniendo los ítems más afines a los gustos del usuario activo a partir de un subconjunto de usuarios U_k más cercanos del conjunto U de usuarios.

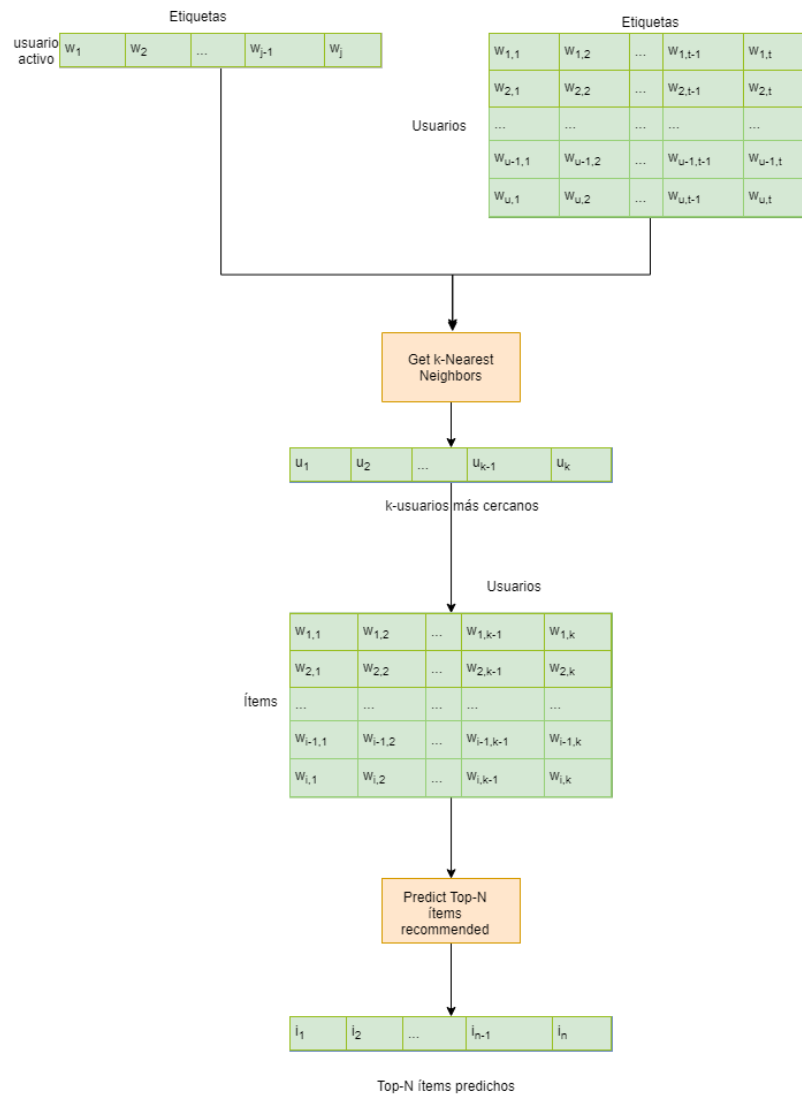


Figura 3. Recomendación Top-N artistas.

3.4. Prototipo del sistema

En esta sección se introduce un prototipo del funcionamiento del sistema de recomendación basado en etiquetado social, cuyo funcionamiento es muy sencillo. El usuario podrá acceder a la pantalla principal del sistema sin loggarse y deberá introducir el nombre de su cuenta de Twitter y el sistema de recomendación procesará la información de sus *tweets*, mostrando los artistas más afines a sus gustos por pantalla (Fig. 4). Para conseguir un diseño más bonito de la aplicación, se ha hecho uso de la API de Spotify¹⁰ para la obtención de las características de un artista, en concreto la imagen y el nombre, aunque podrían añadirse más atributos en un futuro.

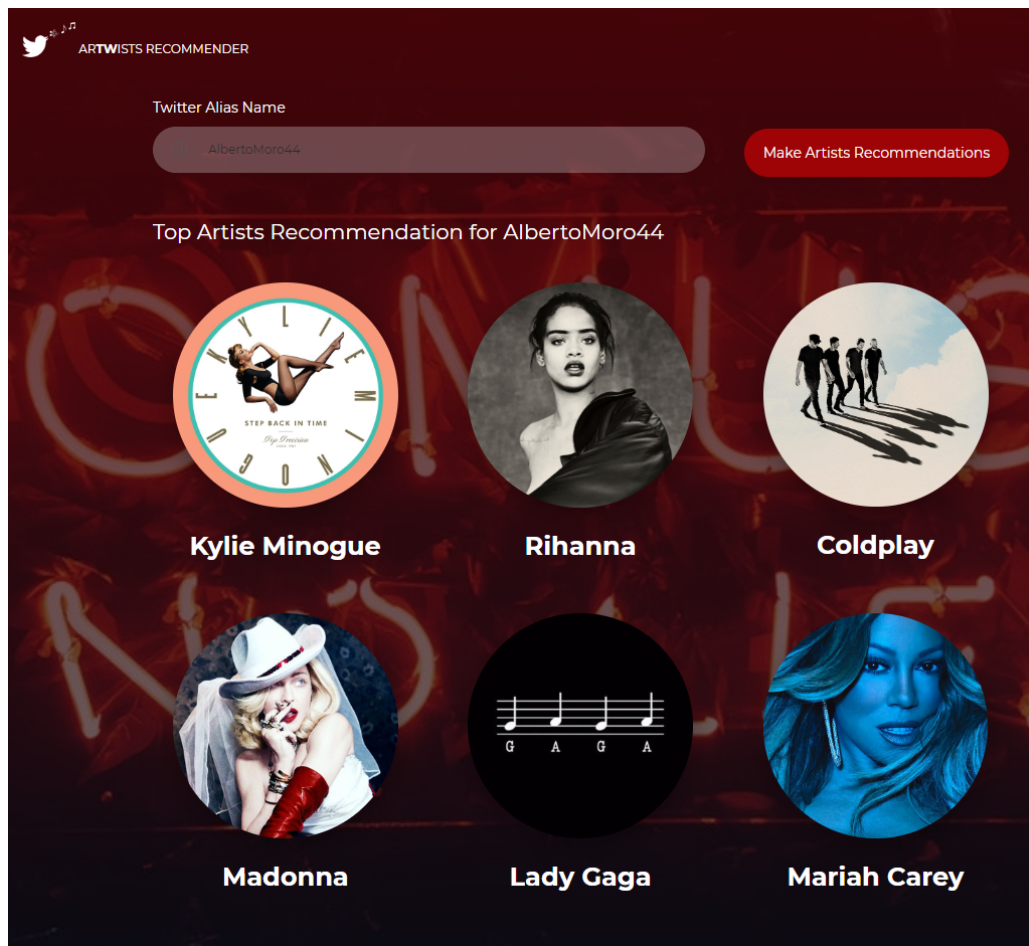


Figura 4. Obtención de artistas del sistema de recomendación propuesto.

¹⁰ Spotify API: <https://developer.spotify.com/>

4. Resultados de los Experimentos

Tras describir y analizar el sistema propuesto, en este apartado se obtendrán los resultados sobre los experimentos realizados con el sistema propuesto, en comparación con otros sistemas.

4.1. Conjunto de datos

Para realizar los experimentos se ha utilizado el conjunto de datos *hetrec2011-lastfm-2k*[3] publicado en el segundo taller internacional sobre Heterogeneidad de la Información y Fusión en Sistemas de Recomendación (HetRec 2011¹¹). Este conjunto de datos contiene información de casi 2.000 usuarios que forman parte de la red social Last.fm.

De este conjunto de datos se obtienen las dos entradas principales del sistema: los perfiles de artistas basados en etiquetas y los perfiles de usuario basados en valoraciones. Estos perfiles han sido obtenidos a partir de las tuplas ternarias de los datos $\langle userID, tagID, artistID \rangle$ que denota la etiqueta *tagID* que el usuario *userID* ha proporcionado al artista *artistID*.

Además de esta información, el conjunto de datos proporciona una tupla ternaria $\langle userID, artistID, plays \rangle$ que denota el número de veces que un usuario *userID* ha escuchado una canción del artista *artistID*. El *rating* que un usuario le da a un artista es vital para obtener los artistas más afines a los gustos de un usuario. Para ello, se ha utilizado una ecuación similar a la que se ofrece en [22], donde se calcula la frecuencia de reproducciones para un artista *i* dado y un usuario *j* según la ecuación (Ec. 6):

$$Freq_{i,j} = \frac{p_{i,j}}{\sum_{i'} p_{i',j}} \quad (6)$$

donde $p_{i,j}$ es el número de veces que el usuario *j* ha reproducido al artista *i*. Por otro lado, $Freq_k(j)$ denota la frecuencia del *k*-ésimo artista más escuchado por el usuario *j*. A partir de este valor, se puede obtener el *rating* de un artista con un *ranking* *k* mediante la ecuación (Ec. 7):

$$r_{i,j} = 4 \left(1 - \sum_{k'=1}^{k-1} Freq_{k'}(j) \right) + 1 \quad (7)$$

Esta ecuación permite obtener una valoración del 1 al 5, según el número de veces que el usuario *j* haya escuchado al artista *i*, en relación a la frecuencia del resto de artistas escuchados, de manera que el artista más escuchado tendrá valores próximos a 5, mientras que el menos escuchado tendrá una valoración de 1.

El conjunto de datos está formado por 1892 usuarios, 11946 etiquetas y un total de 17632 artistas distintos, de los cuales, 12523 han sido etiquetados alguna vez por al menos un usuario. El número de tuplas $\langle userID, tagID, artistID \rangle$ es de 186.479, que indica el número de veces que cualquier usuario ha etiquetado con una sola etiqueta a un artista, mientras que el número de tuplas $\langle userID, artistID, plays \rangle$ es de 92.834.

El conjunto de datos ha sido dividido en una proporción 75 %-25 % entre el conjunto de entrenamiento y el conjunto de pruebas respectivamente.

¹¹ HetRec 2011: <http://ir.ii.uam.es/hetrec2011>

4.2. Evaluación de los resultados

A continuación, se llevará a cabo la evaluación de los resultados obtenidos, con el fin de lograr sacar conclusiones finales y comparar el sistema de recomendación basado en etiquetado social con otros sistemas de recomendación.

Evaluación de algoritmos de recomendación Antes de implementar el sistema de recomendación basado en etiquetado social, se realizaron un conjunto de experimentos con el objetivo de hacer una comparación entre diferentes algoritmos de recomendación para el conjunto de datos. Estos fueron obtenidos mediante la librería Surprise¹² de Python, con el fin de obtener el que menor medida del error produzca y aplicarlo al sistema de recomendación propuesto. Para ello, los diferentes modelos implementados han sido sometidos a pruebas de error, mediante validación cruzada, que divide el conjunto de datos en k particiones distintas (*folds*) y cada una de ellas es usada como conjunto de entrenamiento y de prueba al menos una vez durante la ejecución de cada algoritmo. Se utilizaron las medidas de evaluación MAE (*Mean Absolute Error*) y RMSE (*Root Mean Squared Error*) para obtener una media del error en cada algoritmo. Estos modelos son kNN básico, kNN con técnica *Mean centering*, kNN con *Z-score* y SVD.

Para comparar los algoritmos de recomendación, a partir de los datos obtenidos, se representarán los resultados de las medidas MAE y RMSE en los gráficos Fig. 5 y Fig. 6 respectivamente.

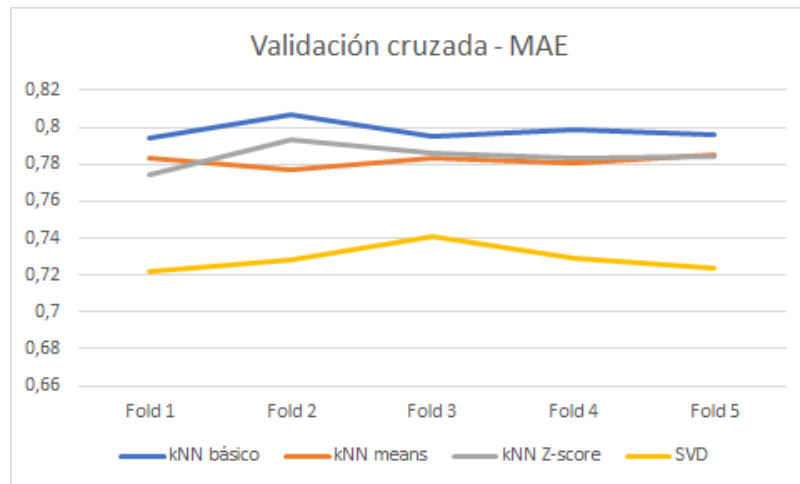


Figura 5. Comparación de algoritmos de recomendación mediante MAE.

¹² Surprise: <http://surpriselib.com/>

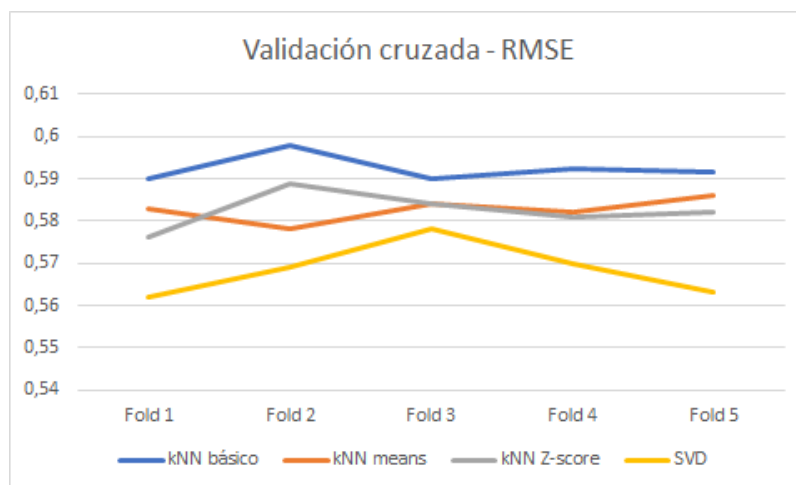


Figura 6. Comparación de algoritmos de recomendación mediante RMSE.

La Fig. 5 muestra una comparación de los diferentes cuatro algoritmos de recomendación probados en este trabajo. Como se puede comprobar, el algoritmo que mejores resultados obtiene es el algoritmo SVD, con una gran diferencia respecto al resto de algoritmos. El algoritmo kNN básico es el que presenta mayor error absoluto en todas las iteraciones. Al aplicar técnicas de normalización como el *mean-centering* o el *z-score*, el error disminuye. La diferencia más destacada entre estas dos técnicas se encuentra en el *fold 2*, donde el *kNN z-score* supera al algoritmo *kNN means*. En la Fig. 6, la comparación entre estos algoritmos es muy similar a la anterior, destacando un aumento del error RMSE del algoritmo SVD en el *fold 3*, que logra alcanzar el error mínimo de algoritmos como *kNN means* y *kNN z-score*.

En conclusión, entre todos los modelos de recomendación analizados, el enfoque basado en modelo SVD presenta unas medidas de MAE y RMSE menores al resto de enfoques, por lo que ha sido el algoritmo de recomendación elegido a la hora de realizar el sistema de recomendación basado en etiquetado social. Además, a pesar de requerir un tiempo de entrenamiento bastante mayor al resto de enfoques, posee un tiempo de prueba menor.

Evaluación del sistema implementado Una vez elegido el algoritmo de recomendación con el que se implementará el sistema de recomendación basado en etiquetado social, se llevará a cabo una comparación entre el sistema implementado y otros sistemas de recomendación: un sistema de recomendación aleatorio, un sistema de recomendación mediante popularidad del artista y un sistema de recomendación de filtrado colaborativo mediante el algoritmo SVD. El aspecto fundamental reside en los resultados entre el sistema de recomendación SVD y el sistema implementado, ya que su única diferencia es la inclusión del etiquetado.

El sistema de recomendación basado en etiquetado social da como resultado una lista Top-N de los n artistas más afines a sus gustos. Para evaluar este sistema con el resto de sistemas anteriormente mencionados, se compararán dichas listas Top-N con las reales en el conjunto de datos de prueba, mediante las medidas de precisión y *recall*. Estas medidas son las más utilizadas e idóneas para comparar resultados como una lista Top-N de ítems predichos, tomando como referencia

una clasificación binaria (si está en la lista Top-N real o si no está). Se ha utilizado la librería *Recmetrics*¹³ para la evaluación de los resultados.

La curva PR (*Precision-Recall curve*) indica de manera visual la relación entre la precisión y la sensibilidad del modelo. Las métricas de precisión y *recall* están relacionadas de manera que si entrenas tu clasificador para aumentar la precisión, disminuirá el *recall* y viceversa. La curva PR es el resultado de dibujar la relación gráfica entre el precisión y el *recall*. Esta gráfica permite ver a partir de qué *recall* tenemos una degradación de la precisión y viceversa. Lo ideal sería una curva que se acerque lo máximo posible a la esquina superior derecha y que el *Average Precision* sea lo más cercano a 1.

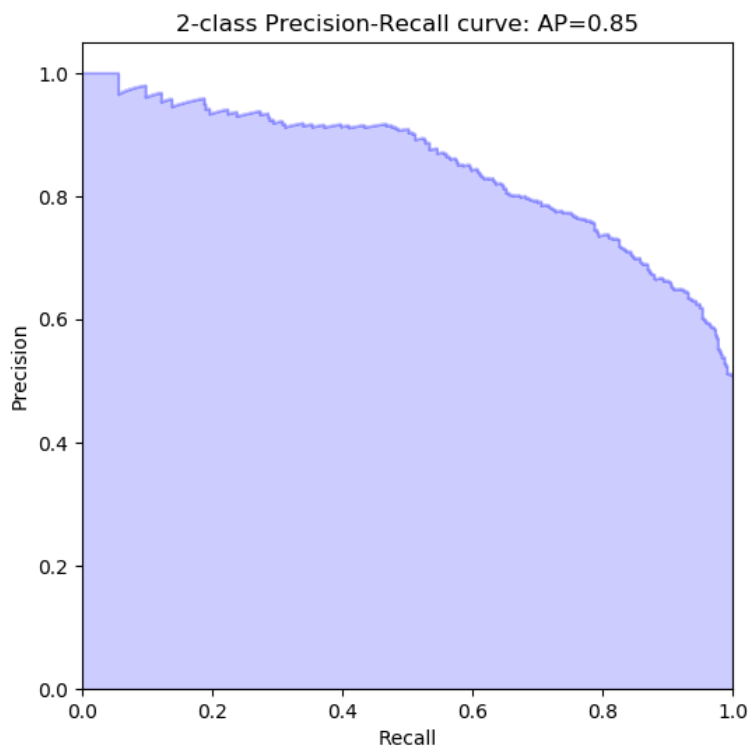


Figura 7. Gráfico de medida precisión-*recall*.

Como se puede observar en la Fig. 7, la relación entre la precisión y el *recall* en el sistema es bastante buena, con un 0.85 de *Average Precision*. Para un mayor detalle, se comparará el sistema implementado con el resto de sistemas mediante las medidas MAP (*Mean Average Precision*) y MAR (*Mean Average Recall*), según (Fig. 8) y (Fig. 9) respectivamente.

¹³ Recmetrics: <https://github.com/statisticianinstiletto/recmetrics>

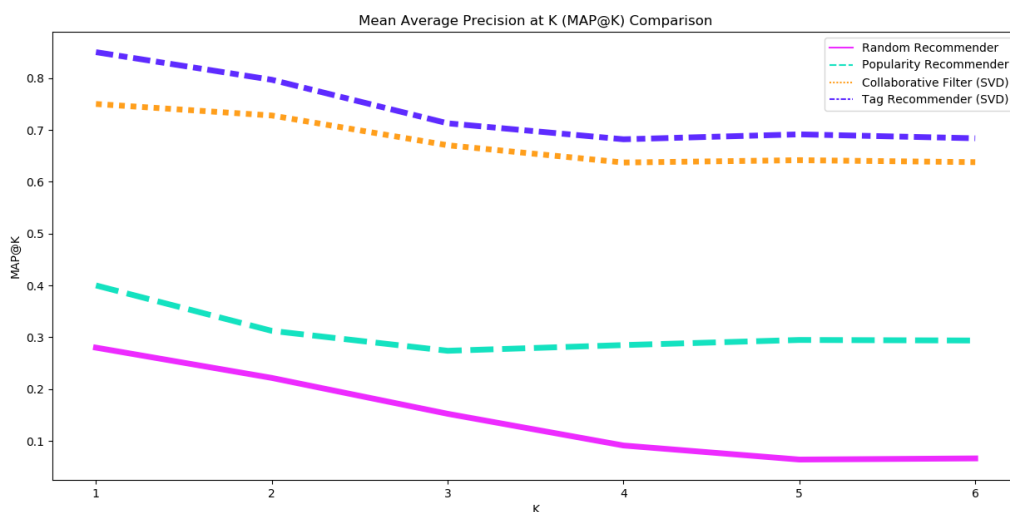


Figura 8. Comparación de los sistemas mediante la medida MAP@K.

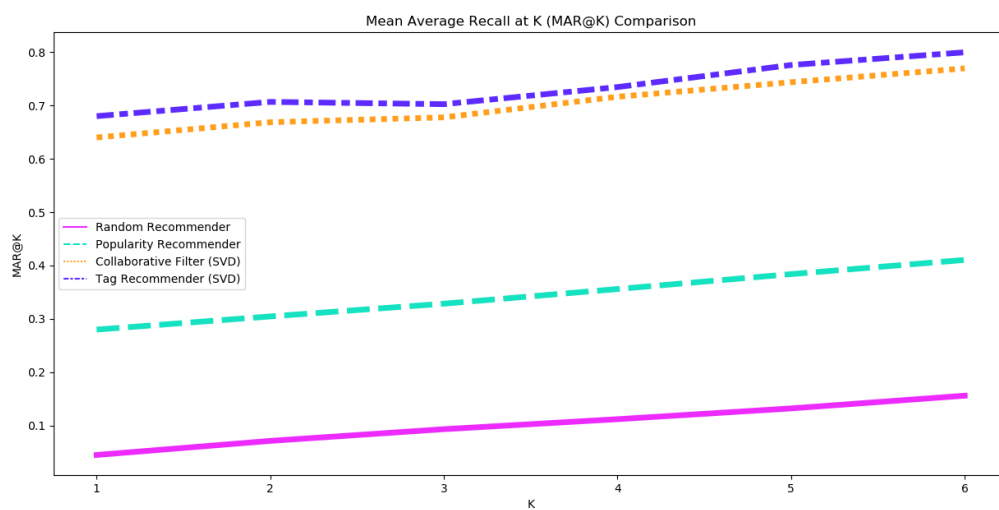


Figura 9. Comparación de los sistemas mediante la medida MAR@K.

Tanto en la Fig. 8 como en la Fig. 9, se observa el mismo comportamiento del sistema: una mejoría sobre el sistema de filtrado colaborativo sin etiquetado. Mientras que el sistema aleatorio sin duda obtiene los peores resultados, seguido del sistema basado en popularidad. Los sistemas de recomendación de filtrado colaborativo basados en el algoritmo SVD presentan unos resultados significativamente mejores a los otros. Sin embargo, la inclusión de una tercera dimensión a través

de etiquetas en el sistema de recomendación permite una mejoría en el sistema, proporcionando mejores resultados, independientemente del k , siendo k el número de ítems predichos a tener en cuenta, hasta un total de 6, que serán los que se tengan en cuenta en la aplicación web.

5. Conclusiones y Líneas de Trabajo Futuras

Este trabajo de investigación propone un sistema de recomendación de música basado en etiquetado social a partir de las redes sociales. Twitter es una red social enorme donde los usuarios expresan sus opiniones y gustos. La idea de utilizar una herramienta de información tan grande como Twitter en un sistema de recomendación, ha logrado mejorar los resultados en comparación con un sistema de recomendación de filtrado colaborativo básico.

Se han estudiado con detenimiento los diferentes tipos de sistemas de recomendación existentes, analizando sus ventajas y desventajas. Se han comparado mediante las medidas MAE y RMSE, siendo el algoritmo SVD el que menor error ha obtenido en el conjunto de datos proporcionado, por lo que ha sido elegido para la solución propuesta.

Además, se ha estudiado en profundidad el procesamiento del lenguaje natural, sus técnicas y herramientas como el Análisis Semántico Latente, con el fin de extraer las palabras claves, características o etiquetas más importantes de un conjunto de documentos, como son los *tweets* de un usuario.

La investigación se ha centrado en los sistemas de recomendación de etiquetado social. El uso de Twitter como herramienta de información, extrayendo de él etiquetas o palabras clave, añade información adicional del usuario sobre los ítems, generando una relación ternaria entre estos tres elementos. Mediante las medidas de precisión y exhaustividad, se ha comprobado que a la hora de generar una lista de los artistas más afines a los gustos de un usuario, la inclusión del etiquetado mejora los resultados obtenidos, prediciendo más artistas relevantes para el usuario.

El sistema permite una gran escalabilidad respecto a un sistemas de filtrado colaborativo básico, utilizando el etiquetado como herramienta de libertad y subjetividad, a la vez que alcanza una reducción de dimensionalidad de la matriz del conjunto de entrada mediante LSA, obteniendo los ítems más afines a los gustos del usuario activo.

En conclusión, el procesamiento del lenguaje natural ha permitido extraer etiquetas claves de los *tweets* de un usuario, logrando implementar un sistema de recomendación, que genera una lista Top-N de artistas afines a los gustos del usuario con mejores resultados que un sistema de recomendación sin etiquetado.

5.1. Líneas de trabajo futuras

Como líneas de trabajo futuras se plantea la creación de un conjunto de datos a partir de Twitter, mediante el *hashtag* *#nowplaying*, ya que actualmente sólo se han podido procesar los perfiles de usuario basados en etiquetas mediante Tweepy, pero no los perfiles de ítems. En un principio, se propuso crear un conjunto de datos mediante la API Tweepy en modo *streaming*. Por cada *tweet* obtenido se pueden obtener el usuario que lo tuiteó y el conjunto de etiquetas obtenidas tras el procesado del texto, pero ha resultado imposible obtener el artista sobre el que se habla en dicho *tweet*, con el fin de crear la folksonomía a partir de éstos. Uno de los posibles métodos que se podría aplicar sería la filtración de *tweets* mediante el *hashtag* de cada artista. Sin embargo, este proceso ha sido imposible de realizar debido al tiempo de dedicación que esto conllevaría. También se plantea la inclusión del *clustering* como método de reducción de dimensionalidad, permitiendo

mayor rapidez y eficiencia en la elección de los k vecinos más cercanos. Además, se estudiarán técnicas que permitan solventar problemas semánticos en el preprocesamiento de los *tweets* como la sinonimia, la polisemia o la ambigüedad semántica, permitiendo una mejora considerable en la extracción de características. Por último, se pretende mejorar la aplicación web implementada con más información de los artistas y dar más interactividad al usuario, además de la creación de una aplicación móvil para el sistema.

Referencias

- [1] M. Balabanovi e Y. Shoham. «Fab: content-based, collaborative recommendation». En: *Communications of the ACM* 40.3 (1997), págs. 66-72.
- [2] R. Burke. «Hybrid recommender systems: Survey and experiments». En: *User modeling and user-adapted interaction* 12.4 (2002), págs. 331-370.
- [3] I. Cantador, P. Brusilovsky y T. Kuflik. «2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011)». En: *Proceedings of the 5th ACM conference on Recommender systems*. RecSys 2011. Chicago, IL, USA: ACM, 2011.
- [4] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes y M. Sartin. «Combing content-based and collaborative filters in an online newspaper». En: (1999).
- [5] M. Deshpande y G. Karypis. «Item-based top-n recommendation algorithms». En: *ACM Transactions on Information Systems (TOIS)* 22.1 (2004), págs. 143-177.
- [6] C. Desrosiers y G. Karypis. «A comprehensive survey of neighborhood-based recommendation methods». En: *Recommender systems handbook*. Springer, 2011, págs. 107-144.
- [7] J. Diederich y T. Iofciu. «Finding communities of practice from user profiles based on folksonomies». En: *Innovative approaches for learning and knowledge sharing, ec-tel workshop proc.* 2006, págs. 288-297.
- [8] A. K. Ghosh. «On optimum choice of k in nearest neighbor classification». En: *Computational Statistics & Data Analysis* 50.11 (2006), págs. 3113-3123.
- [9] D. Godoy y A. Corbellini. «Folksonomy-Based Recommender Systems: A State-of-the-Art Review». En: *International Journal of Intelligent Systems* 31.4 (2016), págs. 314-346.
- [10] P. Hall, B. U. Park, R. J. Samworth *et al.* «Choice of neighbor order in nearest-neighbor classification». En: *The Annals of Statistics* 36.5 (2008), págs. 2135-2152.
- [11] A. Hotho, R. Jäschke, C. Schmitz y G. Stumme. «Information retrieval in folksonomies: Search and ranking». En: *European semantic web conference*. Springer. 2006, págs. 411-426.
- [12] L. Iaquinta, M. De Gemmis, P. Lops, G. Semeraro, M. Filannino y P. Molino. «Introducing serendipity in a content-based recommender system». En: *2008 Eighth International Conference on Hybrid Intelligent Systems*. IEEE. 2008, págs. 168-173.
- [13] S. B. Imandoust y M. Bolandraftar. «Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background». En: *International Journal of Engineering Research and Applications* 3.5 (2013), págs. 605-610.
- [14] T. K. Landauer, P. W. Foltz y D. Laham. «An introduction to latent semantic analysis». En: *Discourse processes* 25.2-3 (1998), págs. 259-284.
- [15] P. Lops, M. De Gemmis y G. Semeraro. «Content-based recommender systems: State of the art and trends». En: *Recommender systems handbook*. Springer, 2011, págs. 73-105.
- [16] L. B. Marinho, A. Nanopoulos, L. Schmidt-Thieme, R. Jäschke, A. Hotho, G. Stumme y P. Symeonidis. «Social tagging recommender systems». En: *Recommender systems handbook*. Springer, 2011, págs. 615-644.

- [17] I. Mogotsi. *Christopher d. manning, prabhakar raghavan, and hinrich schütze: Introduction to information retrieval*. 2010.
- [18] E. R. Núñez Valdéz. «Sistemas de recomendación de contenidos para libros inteligentes». En: (2012).
- [19] M. J. Pazzani y D. Billsus. «Content-based recommendation systems». En: *The adaptive web*. Springer, 2007, págs. 325-341.
- [20] I. Portugal, P. Alencar y D. Cowan. «The use of machine learning algorithms in recommender systems: A systematic review». En: *Expert Systems with Applications* 97 (2018), págs. 205-227.
- [21] F. Ricci, L. Rokach y B. Shapira. «Introduction to recommender systems handbook». En: *Recommender systems handbook*. Springer, 2011, págs. 1-35.
- [22] D. Sánchez-Moreno, A. B. G. González, M. D. M. Vicente, V. F. L. Batista y M. N. M. Garca. «A collaborative filtering method for music recommendation using playing coefficients for artists and users». En: *Expert Systems with Applications* 66 (2016), págs. 234-244.
- [23] B. Sarwar, G. Karypis, J. Konstan y J. Riedl. *Application of dimensionality reduction in recommender system-a case study*. Inf. téc. Minnesota Univ Minneapolis Dept of Computer Science, 2000.
- [24] A. Shepitsen, J. Gemmel, B. Mobasher y R. Burke. «Personalized recommendation in social tagging systems using hierarchical clustering». En: *Proceedings of the 2008 ACM conference on Recommender systems*. ACM. 2008, págs. 259-266.
- [25] X. Su y T. M. Khoshgoftaar. «A survey of collaborative filtering techniques». En: *Advances in artificial intelligence 2009* (2009).
- [26] Wikipedia. *Tf-idf — Wikipedia, La enciclopedia libre*. [Internet; descargado 21-mayo-2019]. 2018.
- [27] T. Yu. *Email analysis using fuzzy matching of text*. US Patent 7,644,127. Ene. de 2010.
- [28] N. Zheng y Q. Li. «A recommender system based on tag and time information for social tagging systems». En: *Expert Systems with Applications* 38.4 (2011), págs. 4575-4587.

Sistema de recomendación sensible al contexto basado en información biométrica

Alejandra Ospina Bohórquez, María N. Moreno, Ana B. Gil González

¹Universidad de Salamanca, Facultad de Ciencias,
Plaza de los Caídos. s/n, 37008, Salamanca, Salamanca
{ale.ospina15,mmg,abg}@usal.es

Resumen La gran cantidad de música a la que se puede acceder, ha llevado al desarrollo de sistemas de recomendación de música más fiables. Han surgido los sistemas de recomendación de música sensibles al contexto, capaces de sugerir música teniendo en cuenta información contextual. Hay estudios que demuestran que la música ayuda a mejorar el estado de ánimo, y puede cambiar el enfoque de atención de los usuarios durante la realización de alguna actividad, ayudando a hacer esta actividad más llevadera.

El fenómeno del *entrainment* [9] dice que dos procesos rítmicos tienden a sincronizarse, en el caso de los usuarios que escuchan música mientras realizan alguna actividad, se dice que estos intentan sincronizar su actividad con el tempo musical, mejorando la experiencia.

Se propone un sistema de recomendación de música que motive a los usuarios en sus actividades cotidianas, recomendando música basándose en la actividad que se encuentre haciendo el usuario en ese momento.

Se concluyó que este sistema es útil y satisfactorio cuando la actividad no requería una gran concentración. Durante las actividades que requerían movimiento, los usuarios, en su mayoría, indicaron que el esfuerzo percibido era menor usando el sistema de recomendación. Los usuarios indicaron que tras usar el sistema de recomendación su estado de ánimo había mejorado.

Keywords: Sistemas de recomendación sensibles al contexto, música, actividades, *entrainment*, estado de ánimo

1. Introducción

La gran cantidad de música a la que se puede acceder en la actualidad ha hecho que la tarea de escoger canciones adecuadas para los usuarios sea cada vez más difícil.

Hay estudios que demuestran que escuchar música mejora el rendimiento cuando se realiza algún tipo de ejercicio, además de motivar y distraer a los usuarios del cansancio. Asimismo, se ha probado que escuchar música durante alguna actividad o tarea, mejora el estado de ánimo de la mayoría de estos, lo que conduce a que los usuarios consideran esta actividad llevadera.

Si un usuario tiene acceso a miles de canciones, es difícil y poco práctico que escoja las que son adecuadas para escuchar. Organizaciones predeterminadas como *playlists* o sistemas de recomendación pueden facilitarle la tarea.

Los sistemas de recomendación surgen para ofrecer de forma personalizada productos a los usuarios. En este caso, el objetivo principal de un sistema de recomendación de música es minimizar el esfuerzo empleado por el usuario en la búsqueda de música.

Han surgido nuevos sistemas de recomendación que tienen en cuenta no solo los gustos del usuario sino también su entorno, los llamados sistemas de recomendación sensibles al contexto.

Se propone un sistema de recomendación de música sensible al contexto, en el que se tiene en cuenta la actividad que esté realizando el usuario con el fin de recomendarle música que pueda motivarle para continuar con la actividad.

Se ha desarrollado una aplicación móvil que permita detectar la actividad que el usuario esté haciendo y en base a ello realizar la elección adecuada de canciones para el usuario en cada momento.

1.1. Sistemas de Recomendación

Los sistemas de recomendación se han convertido en los últimos tiempos en una importante área de investigación.

Existen diferentes tipos: los que le recuerdan al usuario los ítems que les han mostrado anteriormente, lo que permite que el usuario explore para expandir sus preferencias y los sistemas híbridos que son una combinación de los dos anteriores.

Los sistemas de recomendación se clasifican en tres categorías según los métodos que utilicen: basados en el contenido, de filtrado colaborativo y basado en el conocimiento [10].

Sistemas de recomendación basados en el contenido

En este caso, el usuario recibe recomendaciones similares a aquellas por las que ha mostrado preferencia en el pasado.

Es necesario que el usuario valore un número suficiente de ítems para que el sistema de recomendación pueda entender sus preferencias y de esta manera pueda realizar recomendaciones apropiadas, por lo que si el usuario es nuevo en el sistema es muy difícil hacer una recomendación adecuada, lo que supone una limitación importante [8].

Estos sistemas solo recomiendan ítems similares a los de experiencias pasadas del usuario, por lo que puede ocurrir que un ítem similar a otro que el usuario no ha valorado antes, incluso estando dentro de sus potenciales preferencias, nunca le sea recomendado.

Sistemas de recomendación de filtrado colaborativo

Los sistemas de recomendación de filtrado colaborativo recomiendan a un usuario ítems que otros usuarios similares a él hayan valorado positivamente. El éxito de este tipo de sistemas recae en el número de usuarios que lo utilicen y en el número de valoraciones que asignen a los ítems.

Los usuarios deben indicar sus valoraciones (*ratings*) sobre diferentes productos a recomendar y a partir de estas valoraciones se calculan coeficientes de correlación que muestran similitudes entre usuarios [10].

Las valoraciones se pueden obtener de manera explícita, obteniéndolas directamente de los usuarios asignando alguna puntuación a los diferentes ítems.

También se pueden obtener de manera implícita, obteniéndola indirectamente de los usuarios analizando *timing logs*, registros de compras, registros de navegación, etc.

La técnica de filtrado colaborativo tiene limitaciones como el crecimiento del tiempo de computación según hay más ítems y usuarios, es prácticamente imposible realizar recomendaciones de productos nuevos o a clientes nuevos, además de que a veces hay usuarios con preferencias que no son compatibles con las de ningún grupo, por lo que las recomendaciones no son muy adecuadas.

Sistemas de recomendación basados en el conocimiento

En estos sistemas es necesario construir una base de datos que describa las propiedades de los ítems, además de obtener las preferencias del usuarios teniendo en cuenta los ítems que haya seleccionando anteriormente [10].

Se realiza un cálculo de las similitudes de otros ítems utilizando alguna medida de distancia. Finalmente se obtiene una lista de ítems que puede ser interesantes para el usuario y en base a esta se hace una recomendación.

Métodos híbridos

Los enfoques híbridos combinan técnicas de los diferentes enfoques. Los sistemas de recomendación híbridos pueden hacer recomendaciones en caso de que el usuario sea nuevo en el sistema, superando la limitación de el enfoque de filtrado colaborativo, y puede recomendar nuevos artículos superando las limitaciones del enfoque de los sistemas de recomendación basados en contenido [8].

Burke [2] propuso una clasificación en siete categorías de hibridación de sistemas de recomendación que se pueden apreciar en la Tab. 1.

| Método de hibridación | Descripción |
|--------------------------------|---|
| Ponderado | Las valoraciones de diferentes técnicas de recomendación son combinadas |
| Intercambio | El sistema cambia entre diferentes técnicas de recomendación dependiendo de la situación actual |
| Mezclado | Recomendaciones de diferentes sistemas de recomendación se presentan al mismo tiempo |
| Combinación de características | Características de diferentes fuentes de datos de recomendación se juntan en un solo algoritmo de recomendación |
| Aumento de características | La salida de una técnica se usa como característica de entrada de otra técnica |
| Cascada | Un sistema de recomendación afina las recomendaciones dadas por otro sistema |
| Meta-level | El modelo aprendido por un sistema de recomendación se usa como entrada de otro |

Tabla 1. Métodos para hibridación [1]

Recomendación

Durante el proceso de recomendación, primero se obtiene un conjunto de *ratings* de los usuarios, el sistema de recomendación estima la función R de las valoraciones y finalmente se recomienda los k ítems mejor valorados [10].

$$R : User \times Item \rightarrow Rating \quad (1)$$

Hay diferentes formas de realizar la recomendación al usuario entre ellas está hacer una predicción de *ratings*, en la que se intenta predecir las valoraciones que un usuario asignaría a un ítem, y teniendo en cuenta esta predicción se realiza la recomendación. Por otro lado también están la recomendaciones *top-N* que recomienda una lista ordenada de N ítems.

1.2. Sistemas de Recomendación Sensibles al Contexto

En muchos casos los sistemas de recomendación no solo analizan las preferencias de los usuarios, sino que también tienen en cuenta el contexto en el que se encuentran. Con la ayuda de los dispositivos móviles actuales, se pueden integrar distintos factores personales, sociales y contextuales al proceso de recomendación, de esta manera se puede dar una recomendación más adecuada a un usuario específico en el momento conveniente basado en su actividad actual o algún otro factor que se pueda deducir a partir de los datos obtenidos por el dispositivo.

En los últimos años se ha empezado a estudiar la importancia de incorporar información contextual a los sistemas de recomendación extendiendo el modelo bidimensional ($User \times Item$) y proponiendo una nueva función multidimensional de valoración [8]:

$$R : User \times Item \times Context \rightarrow rating \quad (2)$$

Se pueden clasificar en tres categorías según la forma de incorporar la información contextual: enfoques de prefiltrado contextual, enfoques de posfiltrado contextual y enfoque de modelado contextual [10].

Prefiltrado contextual

Con el enfoque de prefiltrado la información contextual es incorporada antes del cálculo de las recomendaciones eliminando los datos no correspondientes al contexto objeto de recomendación .

La información de contexto se usa durante la selección y construcción de los conjuntos de datos. Seguidamente se hace la predicción de las valoraciones usando los sistemas de recomendación bidimensionales para los datos seleccionados.

Posfiltrado contextual

Los enfoques de posfiltrado es en donde se ignora la información contextual cuando se está generando la recomendación y luego se ajusta la lista de recomendaciones o se descartan aquellas que se consideren irrelevantes.

Modelado contextual

El enfoque de modelado contextual es el que usa la información contextual directamente en la función multidimensional. Es decir, se usa dentro de la técnica de modelado como parte de la predicción de las valoraciones.

2. Estado del Arte

2.1. Sistemas de Recomendación de Música

Se han desarrollado sistemas de recomendación de música, basándose en los gustos de los usuarios, utilizando diferentes técnicas.

Fang et al. [3] proponen un sistema de recomendación que incorpore un perfil de usuario para poder dar al usuario un muy buen set de recomendaciones iniciales. Luego se utiliza un reforzamiento

del aprendizaje cada vez que una recomendación es aceptada o rechazada para asegurar que sea más probable que futuras recomendaciones sean aceptadas [3].

Asimismo, Yajie Hu [6] propuso un sistema de recomendación de música basado en el comportamiento del usuario y la clasificación por género. Los autores afirman que es posible evaluar la actitud de un usuario con respecto a una canción teniendo en cuenta si escucha la canción entera, y si no lo hace, teniendo en cuenta el tamaño del fragmento de canción que escuchó. Si el usuario salta la canción, se asume que fue una mala recomendación, sin importar la razón. Si la canción recomendada se reproduce completamente, se puede inferir que al usuario le gusta la canción y que es una recomendación satisfactoria. Usando esta idea se propuso un método que recomendara música automática, basándose en varios factores: novedad, género según patrones temporales, año y si al usuario le gusta.

Por otro lado Sánchez Moreno et al. [12] proponen un sistema de recomendación en lo que la información sobre *ratings* y contenidos no está disponible. La mejora en la recomendación la consiguen centrándose en los usuarios ovejitas negras (en inglés *grey sheeps*). A diferencia de otros sistemas de recomendación, aquí las ovejitas negras no se tratan diferente a los demás usuarios y no se separan en un grupo diferente. La estrategia consiste en determinar un coeficiente que indique el grado en que cada usuario se considera una ovejita negra.

2.2. Sistemas de Recomendación de Música Sensibles al Contexto

Con el aumento de las capacidades de los dispositivos móviles y la disponibilidad inmediata de diferentes tipos de información en la Web, las oportunidades de incluir información contextual adicional ha aumentado de manera continua [1].

Greenhalgh et al. [5] proponen la recomendación de canciones (*geotracks*) y lista de reproducción (*geolists*) alineadas y adaptadas a trayectos específicos, como el camino al trabajo. Aspectos de la reproducción de *geotracks* y la transición entre ellas puede verse influenciados por el progreso del usuario, su actividad y contexto. Por ejemplo, un subset de un *geotrack* puede ser reproducido de manera que la música alcance un final adecuado cuando el usuario termina de recorrer un parte específica del camino.

Los *geotracks* se escogen porque de alguna forma son apropiados para el lugar donde van a ser oídos. Por ejemplo, una pista musicalmente densa puede ajustarse mejor a un paseo por una calle muy transitada, mientras que una pista más tranquila puede ser más adecuada para un paseo a través de un parque.

Braunhofer et al. [7] propusieron un sistema de recomendación basado en la ubicación. La idea es recomendar música que encaje con lugares de interés. Para ello usaron *tags* sobre las emociones que los usuarios le asignaban tanto a las canciones como a los lugares de interés, además consideraron una serie de métricas de similitud para los *tags* y establecieron una forma de emparejamiento entre las canciones y los lugares de interés. Con este objetivo en mente, desarrollaron una aplicación móvil que sugiriera un itinerario y reprodujera la música recomendada para cada lugar de interés visitado. Los resultados demostraron que los usuarios juzgaron la música recomendada como correcta para los lugares de interés.

Por otro lado, Moens et al. [9] propusieron un sistema musical llamada *D-Jogger* que utiliza el movimiento del cuerpo para seleccionar música dinámicamente y adaptar su tempo al paso del usuario. Este sistema consta de diferentes módulos independientes, como el algoritmo de detección de pasos y playlists que cuenten con sus tempos, que le permiten alcanzar su objetivo. Realizaron un experimento piloto que sugirió que cuando el tempo musical y el paso del usuario son lo suficientemente cercanos, la mayoría de los usuarios sincronizan su caminar a la tempo musical, dando

un paso con cada beat. Además, un cuestionario para los usuarios indicó que los participantes experimentaron esto como estimulante y motivador.

Existe un fenómeno denominado en inglés *entrainment* [9], que habla de cómo dos o más procesos rítmicos independientes se sincronizan entre ellos. Este proceso se puede apreciar en situaciones cotidianas, por ejemplo, durante los bailes en los que las personas realizan movimientos rítmicos en sincronía con el pulso musical percibido. Este fenómeno también se puede apreciar cuando las personas caminan al ritmo de la música.

También, Oliver et al. [11] propusieron un sistema móvil y personal (hardware y software) que los usuarios pudieran utilizar durante la actividad física. El hardware de este sistema incluye un monitor de ritmo cardiaco y aceleración conectado de manera inalámbrica a un teléfono móvil que llevaría el usuario. El software permite que el usuario introduzca una rutina de ejercicios, para ayudarlo después a alcanzar los objetivos de la rutina monitorizando de manera constante su fisiología (ritmo cardiaco) y movimiento así como seleccionando y reproduciendo música con características específicas que lo guiarían a alcanzar los objetivos de la dicha rutina.

Sen et al. [13] proponen un sistema de recomendación de músico cuyo objetivo es recomendar música novedosa para los usuarios basándose en información contextual obtenida de sensores. En este no sería necesario que el sistema aprendiera sobre los gustos musicales de los usuarios. De esta manera, el usuario podría descubrir música nueva continuamente sin ningún tipo de impedimento, como el tener que interactuar frecuentemente con el sistema.

La idea de este sistema de recomendación es, en vez de asumir que la recomendación de música se trata sobre todo de la personalización, este sistema intenta integrar la personalización mínima necesaria. Para ello se basan en dos ideas fundamentales: la primera es que el contexto o las situaciones en la que un usuario se encuentra reflejan su estilo de vida y preferencias musicales para lugares y actividades. La segunda es que se le permite al usuario un grado de control mínimo por ejemplo a la hora de excluir canciones por su género.

A continuación se puede ver una tabla comparativa en donde se muestran algunas de las características que se consideran interesantes para un sistema de recomendación de este tipo, indicando a su vez que propuestas lo cumplen.

| | Greenhalgh et al. | Braunhofer et al. | Moens et al. | Oliver et al. | Sen et al. |
|--|-------------------|-------------------|--------------|---------------|------------|
| Interacción mínima del usuario | ✗ | ✗ | ✗ | ✗ | ✓ |
| <i>Entrainment (BPM)</i> | ✗ | ✓ | ✗ | ✗ | ✗ |
| Necesidad solo de un smartphone | ✓ | ✓ | ✓ | ✗ | ✓ |
| Recomendación durante el ejercicio físico | ✗ | ✗ | ✓ | ✓ | ✓ |
| Recomendación durante actividades cotidianas | ✓ | ✓ | ✗ | ✗ | ✓ |

Tabla 2. Comparación de sistemas de recomendación de música sensibles al contexto

2.3. Influencia de la Música en el Desempeño de Actividades

La música como una forma natural de expresión es fundamental para los seres humanos. Permite la construcción de identidades personales y colectivas [15].

En los últimos tiempos se ha propuestos un modelo central regulado para explicar el rendimiento durante la realización de algún ejercicio. En este modelo todas las señales exteriores se integran y se usan para generar el índice de esfuerzo percibido (RPE - rating of Perceived Exertion) ¹ [8]. En la Fig. 1 ² se ve la escala del RPE.

Se ha notado que algunas técnicas de manipulación psicológica alteran la respuesta RPE durante un ejercicio constante. Por ejemplo, se ha demostrado que escuchar música rápida, animada, clásica o preseleccionada para el usuario durante el ejercicio reduce el RPE comparado con la misma situación pero sin música. También se descubrió que escuchar música durante la realización de algún ejercicio reduce el RPE comparado con condiciones de control o privación sensoriales.

| Escala RPE | Percepción subjetiva del esfuerzo |
|------------|---|
| 10 | Actividad de esfuerzo máximo: Se siente casi imposible seguir con la actividad. Completamente sin aliento, incapaz de hablar |
| 9 | Actividad muy dura: Muy difícil de mantener la intensidad del ejercicio. Apenas puede respirar. Decir solo unas pocas palabras. |
| 8 | Actividad vigorosa: Falta de aliento. Puede decir una oración |
| 7 | Actividad moderada: Respiración pesada, puede mantener una conversación corta. Actividad algo cómoda, pero cada vez más desafiante |
| 6 | Actividad ligera: Se siente como si pudieras realizar la actividad durante horas. Fácil de respirar y mantener una conversación |
| 5 | Actividad muy ligera: Casi ningún esfuerzo, pero más que dormir y ver televisión |
| 4 | |
| 3 | |
| 2 | |
| 1 | |

Figura 1. Índice de esfuerzo percibido

El modelo de procesos paralelos sugiere que cuando se escucha música durante el ejercicio, existe una competencia de la información cognitiva, donde esta información procede de diferentes fuentes (las condiciones externas, como puede ser la música, y las condiciones internas del cuerpo, como la frecuencia respiratoria, la ventilación, entre otras) que compiten por la atención.

Según este modelo, el hecho de escuchar música ejerce más influencia en ejercicios de no mucha intensidad, ya que en estos casos la estimulación externa puede competir de manera más clara frente a señales internas más débiles, de forma que pueda conseguir una atención focalizada.

Anteriores trabajos e investigaciones sugieren que la música tiene efectos muy positivos cuando se realiza alguna actividad física. Esto se debe a que la música provee ventajas en cuanto al ritmo, distrae de la fatiga, afecta el estado anímico de manera positiva, aumenta la confianza y la autoestima y motiva a los usuarios a ejercitarse más. Finalmente diez diferentes estudios concuerdan en que la percepción del rendimiento y los niveles de esfuerzo percibidos se ven influenciados de manera positiva en condiciones donde se escucha música en comparación con condiciones en las que no se escucha [11].

Se ha demostrado que la música tiene un efecto motivacional que anima a las personas a ejercitarse más vigorosamente o por periodos de tiempo más largos [3].

Simpson y Karageorghis examinaron el efecto de la música en el rendimiento durante carreras de 400 metros de velocidad mientras controlaban el estado de ánimo antes de la carrera. Se demostró que aquellos corredores que escucharon música tuvieron mejor rendimiento durante la carrera. En

¹ El RPE mide la gama entera del esfuerzo que un individuo percibe al hacer ejercicio

² Imagen tomada de <https://thefittutor.com/rpe-scale/>

otros estudios, Styns observó que los participantes de su estudio caminaban más rápido con música que con ticks de un metrónomo. Por su parte, Bood mostró que el tiempo hasta el agotamiento era bastante más largo con estimulación acústica que sin ella cuando se les pedía a los participantes que corrieran sobre una cinta eléctrica. De este tipo de estudios se puede deducir que la música puede ser usada durante actividades físicas, como correr o caminar, y que con ella se podrán apreciar efectos considerablemente positivos [14].

Además del factor motivacional, se cree que los ejercicios que son de naturaleza repetitiva se benefician más de la música que se sincroniza con el ritmo de los movimientos del ejercicio, con ella la resistencia de los que se ejercitan puede aumentar y pueden ejercitarse con mayor intensidad cuando se mueven en sincronía con el estímulo musical. Se ha sugerido que el efecto de la utilización de música sincronizada durante ejercicio se debe a que tiene la capacidad de reducir el coste metabólico del ejercicio al mejorar la eficiencia neuromuscular o metabólica.

Es un hecho que la música está directamente asociada con las emociones. La música estimula centros cerebrales que manejan las emociones y permiten manifestar el sentimiento musical. A pesar de que el campo de las emociones es bastante subjetivo, diferentes investigaciones concluyen que la música influye en las reacciones emocionales de las personas [15], por lo que puede cambiar el estado de ánimo.

3. Propuesta

Se propone un sistema de recomendación de música sensible al contexto. La idea de este sistema es que tenga en cuenta las actividades cotidianas de los usuarios y en base a ello le haga una recomendación que motive al usuario a continuar con dicha actividad.

Existen diferentes estudios que indican que escuchar música mientras se realiza algún tipo de actividad puede ser beneficioso. Una de las novedades de este trabajo es la recomendación de música para usuarios durante sus actividades cotidianas, la mayoría de los estudios analizados hablan del uso de la música en ejercicios de alta intensidad pero casi ninguno habla de recomendación de música para usuarios durante sus actividades cotidianas.

Teniendo esto en cuenta, la propuesta es una aplicación móvil que implementa un sistema de recomendación de música que con ayuda del acelerómetro del dispositivo pueda predecir la actividad del usuario y en base a esta haga una recomendación, teniendo también en cuenta las valoraciones que tienen las canciones.

Como la idea es recomendar música mientras el usuario está realizando alguna actividad, sería útil que el usuario no tuviera que interactuar constantemente con el sistema, sino que simplemente pudiera escuchar canciones que se le recomienden sin que este realmente pudiera intervenir, de esta manera se le puede recomendar música nueva al usuario que este valoraría y luego estas valoraciones se podrían tener en cuenta para futuras recomendaciones.

La idea es utilizar un set de datos que relacione los datos devueltos por el acelerómetro del móvil con las actividades físicas que realiza un usuario en base a ellas. El siguiente paso sería aplicar un algoritmo de aprendizaje automático que permita obtener como salida la actividad que esté realizando un usuario a partir de los datos del acelerómetro.

Una vez obtenido el modelo, se utilizará dentro de una aplicación móvil en donde primeramente se predecirá la actividad física con ayuda de este modelo dándole como entrada los valores que devuelva el acelerómetro del móvil.

Por otro lado se tendrá un *dataset* de canciones de donde se puedan obtener características de las mismas (título, artista, BPM, imagen, previo, etc.). Ahora bien, el siguiente paso será escoger

la canción en base a la actividad física. Para ello, se tuvo en cuenta el fenómeno del *entrainment*, que como se dijo anteriormente, habla de cómo dos procesos rítmicos independientes se sincronizan entre ellos. Por tanto la idea es emparejar el BPM de las canciones con la actividad del usuario.

Si un usuario está haciendo alguna actividad más intensa, como correr será necesario que el BPM de la canción que escucha sea más alta, es decir su tempo musical será mayor. Sin embargo, como la idea es recomendar música no solo para actividades rítmicas (correr, caminar, etc) sino también para actividades tan tranquilas como es estar sentado o de pie trabajando, se recomendarán también canciones más tranquilas con tempos musicales menores. Durante estas últimas actividades el usuario prefiere estar tranquilo y concentrado, por lo que para este tipo de actividades se utilizarán canciones con BPM más bajos, canciones más tranquilas, con tempos musicales bajos.

Por tanto, se considera aplicar un método de prefiltrado contextual en el se divida el set de canciones en rangos del BPM (tantos rangos como actividades se predigan), en donde cada una de las divisiones del dataset estén relacionadas con una de las actividades.

Ahora bien, sería importante también recomendar canciones que además de tener un BPM apropiado que gusten al usuario. Por tanto, se usará además un set de datos que contenga valoraciones de las canciones y partir de el aplicar una técnica top-N en la que se encuentran canciones ordenadas por la valoración obtenida, de esta manera se recomendarán también canciones a las que el usuario le asignaría valoraciones altas.

Finalmente, se reproducirá la canción que se le recomiende al usuario, esto se puede hacer reproduciendo música a partir del repositorio de Spotify. Como primera prueba se pueden utilizar previos que Spotify permite utilizar de manera gratuita y sin estar registrado.

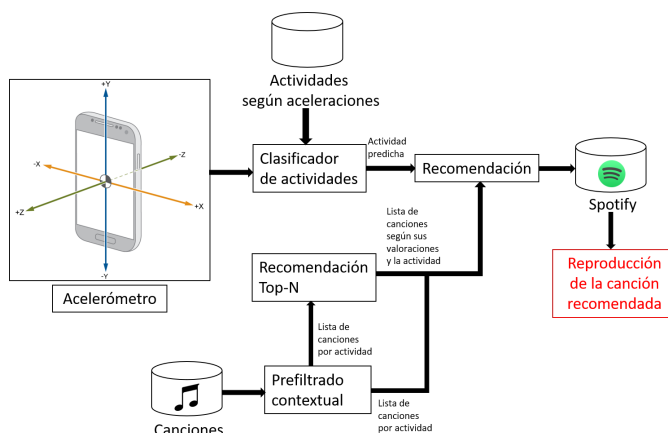


Figura 2. Arquitectura del sistema propuesto

Esta propuesta cuenta con características que se consideraron deseables en estudios anteriores como se puede apreciar en Tab. 3

| | Greenhalg et al. | Braunhofer et al. | Moens et al. | Oliver et al. | Sen et al. | Propuesta |
|--|------------------|-------------------|--------------|---------------|------------|-----------|
| Interacción mínima del usuario | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ |
| Entrainment (BPM) | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ |
| Necesidad solo de un smartphone | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ |
| Recomendación durante el ejercicio físico | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Recomendación durante actividades cotidianas | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ |

Tabla 3. Comparación de sistemas de recomendación de música sensibles al contexto y la propuesta

4. Estudio Experimental

Se desarrolló un aplicación móvil, a través de la cual se han recogido resultados experimentales de la propuesta. Para ello se exploraron diferentes opciones para la clasificación de actividades con algoritmos de aprendizaje automático, y se buscó la forma de trasladar el modelo obtenido a la aplicación.

Además, se aplicó la idea del *entrainment*, a partir de la cual se pretende sincronizar dos procesos rítmicos independientes (caminar, correr, etc. y las canciones), además de probar que las canciones con BPMs bajos permiten que los usuarios disfruten más de las actividades menos intensas (estar sentado o de pie). Entonces para relacionar las actividades con las diferentes canciones, se dividió el conjunto de canciones en tantos rangos como actividades que se pueden predecir, basándose en el BPM.

También se hizo tabla top-N de las canciones, basándose en valoraciones anteriores, de forma que, además de recomendar canciones apropiadas para las actividades, se recomendarán canciones que estén entre las lista de canciones mejor valoradas.

4.1. Clasificación de la Actividad Física

Hoy en día la mayoría de los *smartphones* cuentan con sensores, a través de los cuales se pueden obtener diferentes datos y de ellos inferir sobre la situación contextual del usuario de dicho dispositivo.

Para lograr el objetivo de este sistema, es necesario conocer la actividad que estás realizando un usuario. Esta actividad se puede inferir a partir del acelerómetro del móvil. Los móviles cuentan con un acelerómetro que mide la aceleración en las tres dimensiones espaciales

Para la realización de esta aplicación se usará un set de datos que proporciona el laboratorio de *Wireless Sensor Data Mining (WISDM)*.

Se eligió este *dataset*, debido a que las actividades que clasifica, son actividades cotidianas, actividades que cualquier usuario podría realizar en su día a día: caminar (*walking*), correr (*jogging*), subir escaleras (*upstairs*), bajar escaleras (*downstairs*), estar de sentado (*sitting*) y estar de pie (*standing*).

Este *dataset* sin valores desconocidos cuenta con 1098207 tuplas y 6 atributos: *user*, *activity*, *timestamp*, *x-axis*, *y-axis* y *z-axis*. El atributo clase en este caso sería la actividad que es aquello

que se quiere predecir. En el gráfico de la Fig. 3 se puede apreciar la distribución frecuencias de los valores de la clase dentro del *dataset*.

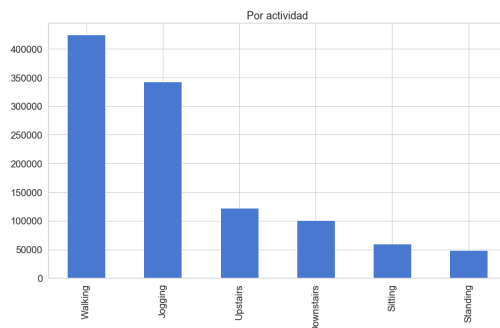


Figura 3. Distribución de frecuencias de las clases

El atributo *timestamp* indica la marca de tiempo de cuando se tomaron dichos datos, por otro lado *x-axis*, *y-axis* y *z-axis* son atributos que muestran las aceleraciones en cada uno de los ejes espaciales (*x*, *y* y *z*).

Algoritmo de Aprendizaje Automático. Para construir el clasificador de actividades se entrenó una red neuronal del tipo LSTM (Long Short Term Memory). Esta red tiene conexiones cíclicas entre los nodos, esto lleva a que puedan usar sus estados internos para procesar secuencias de entrada. Estas características hacen que este tipo de redes neuronales sean adecuadas a la hora de clasificar datos en series de tiempo, en esta caso, el *dataset* que se va a utilizar cuenta con el atributo *timestamp* que nos indica cuándo fueron tomados esos datos, por lo que se consideró interesante la implementación de este algoritmo.

Los resultados fueron buenos: una precisión de 0.97, un *recall* de 0.96 y una exactitud de 0.95. La librería *TensorFlow* se utilizó para la implementación de este algoritmo y para exportarlo a la aplicación Android.

A continuación en la Tab. 4 y en la Fig. 4 se puede ver el progreso durante el entrenamiento.

| Epoch | Precisión | Pérdida |
|-------|-----------|---------|
| 1 | 0,77 | 0,99 |
| 10 | 0,94 | 0,56 |
| 20 | 0,96 | 0,39 |
| 30 | 0,97 | 0,29 |
| 40 | 0,97 | 0,25 |

Tabla 4. Progreso durante el entrenamiento

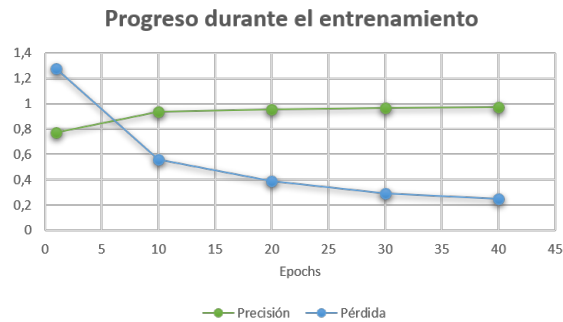


Figura 4. Progreso de la precisión y la pérdida

4.2. Clasificación de las Canciones según la Actividad

Para la clasificación de las canciones se usó un set de datos que relaciona las canciones con una serie de atributos. Los atributos que se utilizaron fueron:

- *N*: indica la numeración de las tuplas del *dataset*.
- *artistas*: se muestran los nombres de los artistas de cada canción. Esto se utilizará para desplegar los nombres de los artistas en en la interfaz.
- *bpm*: indica el BPM de la canciones, que como se ha dicho anteriormente, es la propiedad que se utilizará para escoger las canciones en base a la actividad que estén realizando los usuarios.
- *cancion_id*: indica el identificador de la canción en este *dataset*, luego este se utilizará para asociar este *dataset* con el dataset de valoraciones asociado en el que se incluye este atributo.
- *imagen*: este atributo hace referencia a la imagen que viene asociada con cada canción. Estas imágenes está sacadas de un repositorio de Spotify. Este se utilizará para mostrar esa imagen durante la reproducción de las canciones.
- *preview_url*: en este atributo están URLs a fragmentos de las canciones también sacados del repositorio de Spotify. Estos serán los que se reproducirán en la aplicación.

Según se indican en diversos estudios, el BPM puede verse asociado en con la intensidad y el ritmo en el que se realiza un actividad, por ende, se puede decir que para realizar actividades que requieran mayor esfuerzo será mejor que el usuario escuche música con un BPM mayor, de esta manera el usuario se verá más motivado para continuar con dicha actividad y su estado de ánimo mejorará.

Por otro lado, para que el usuario pueda relajarse o mantener la concentración necesaria en las actividades que puedan requerir estar sentado o de pie sería mejor que el usuario escuchara canciones con un BPM menor, canciones más tranquilas.

Para la incorporación de información contextual, se utiliza el enfoque de de prefiltrado contextual. Aquí se incorpora la información contextual antes de realizar la recomendación. Para ello, se hace una división del dataset de canciones en diferentes rangos según el BPM. Se hacen tantas divisiones como actividades se pueden predecir, de la siguiente manera:

- Estar sentado: 0-80 bpm
- Estar de pie: 80-100 bpm

- Caminar: 100-120 bpm
- Bajar escaleras: 120-140 bpm
- Subir escalers: 140-155 bpm
- Correr: más de 155 bpm

Esta división se hizo basándose en el fenómeno del *entrainment* que dice que dos procesos rítmicos independientes se sincronizan. Basándose en esto, la idea es reproducir canciones con un tempo musical más intenso (un BPM mayor) cuando el usuario esta realizando actividades que requieran movimiento (caminar, bajar y subir escaleras y correr), y será más alto según la intensidad de la actividad.

Además, para actividades en la que el usuario espera estar relajado o concentrado se reproducirán canciones con tempos musicales más lentos, que le permitan disfrutar o hacer más llevadera dicha actividad.

4.3. Recomendación al Usuario

Las valoraciones se consiguieron a partir de un set de datos proporcionado por Gomes et al. [4] asociado a la de las canciones. A este *dataset* le agregué dos atributos: count y mean. Los atributos que se utilizaron fueron:

- N
- *Canción_id*: atributo que permitió realizar la asociación entre este dataset y el dataset de las canciones.
- *Valoración*: atributo que indica la valoración dada por un usuario a una canción. Esto se utilizó para sacar la valoración media de cada una de las canciones.
- *Count*: atributo que indica el número de veces que una canción ha sido valorada. Se utilizó tanto para sacar la valoración media como para calcular la tabla top-N como se verá más adelante.
- *Mean*: atributo que indica la valoración media de cada canción. Este se usó para calcular la tabla top-N.

Debido a que las valoraciones no son suficientes, se hizo un sistema de recomendación relativamente sencillo, pero a futuro, cuando se tengan más valoraciones se debería utilizar uno que se adapte más a los usuarios.

Se hizo *valoración ponderada* que tomara en cuenta la media de las valoraciones y el número de votos que tengan las canciones. Para ello se utilizó la fórmula de *rating* ponderado (WR por su nombre en inglés Weighted rating) que se representa matemáticamente de la siguiente forma:

$$WR = \left(\frac{v}{v+m} \cdot R \right) + \left(\frac{m}{v+m} \cdot C \right) \quad (3)$$

donde:

- v es el número de votos de las canciones.
- m es el mínimo de votos requeridos para que sean incluidos en la tabla.
- R es la media de los *ratings* de las canciones.
- C es la media de todo los *ratings*.

Una vez aplicado el sistema se obtiene una lista tipo top-N de las canciones.

4.4. Aplicación Android

La aplicación obtiene datos del acelerómetro del *smartphone*, estos sirven como entrada a la red neuronal que predice la actividad del usuario (modelo que se exportó desde Python con ayuda de la librería de *TensorFlow*). Una vez el clasificador produce una salida, ésta se utiliza para la selección de la canción.

Se realiza la recomendación, eligiendo aquella canción que este dentro del top-N y encaje con la actividad que se haya predicho. Luego se reproduce el *preview* de la canción obteniéndola directamente de Internet a través de la url asociada a ella en el *dataset*.

Actualmente la lista top-N es un poco corta, por lo que además de recomendar canciones de esta lista, cada 5 canciones recomendadas del la lista del top-N se recomienda una que no lo esté, de forma que el usuario pueda valorarla y más adelante se pueda actualizar la lista incluyendo más canciones.

Como la idea era que el usuario tuviera la mínima interacción con el sistema para que no tuviera que interrumpir su actividad, la interfaz es muy sencilla (Fig. 5), es simplemente un reproductor de música típico en que se incluyen botones, la imagen y el nombre de la canción y los artistas que interpretan esta canción. Además incluye un forma de proporcionar valoraciones a las canciones (mediante las estrellas).

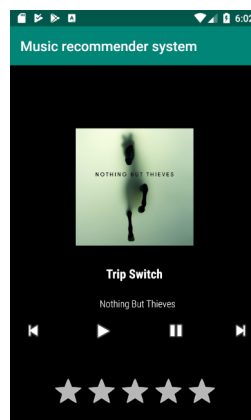


Figura 5. Interfaz de la aplicación

4.5. Resultados

Siete individuos probaron la aplicación para comprobar que tan útil es este sistema. Se hicieron diferentes pruebas en las que los usuarios realizaron las diferentes actividades usando el sistema de recomendación y sin usarlo. Luego se hizo una comparativa entre ambas experiencias.

Para ello se aplicó la idea del índice de esfuerzo percibido (RPE - *rating of Perceived Exertion*), a través de ésta se le pidió al usuario que indicara que el esfuerzo percibido según se indica en la imagen 1 durante las actividades que requerían movimiento (caminar, correr, subir y bajar escaleras).

Ahora bien, para las actividades que no requiere movimiento como estar sentado o de pie se le pidió al usuario primero que indicara qué estaba haciendo mientras estaba realizando dicha

actividad (relajarse o alguna actividad que requiriera concentración) y teniendo esto en cuenta se le preguntó si considera que logró concentrarse o relajarse adecuadamente es una escala de 1 al 5.

Finalmente, para todos los casos, se preguntó sobre su estado de ánimo tras hacer la actividad (mejoró, es igual o empeoró). A continuación en las figuras 7 y 6 se puede ver una comparativa de los resultados usando y sin usar el sistema de recomendación en actividades que requerían movimiento y actividades que no.

Se puede ver que, el índice de esfuerzo percibido (Fig. 7) disminuyó usando el sistema de recomendación en la mayoría de los casos, además de que el estado de ánimo mejora en casi todos los casos. Esto demuestra la utilidad de este sistema de recomendación a la hora de hacer este tipo de actividades.

| | Actividad | Tipo | Sin usar el sistema de recomendación | | Usando el sistema de recomendación | |
|-------------|---------------|------------------|--------------------------------------|-----------------|------------------------------------|-----------------|
| | | | Concentración | Estado de ánimo | Concentración | Estado de ánimo |
| Individuo 1 | Estar sentado | Relajante | 4 | Mejor | 5 | Mejor |
| | Estar de pie | De concentración | 4 | Igual | 2 | Peor |
| Individuo 2 | Estar sentado | Relajante | 5 | Igual | 5 | Mejor |
| | Estar de pie | Relajante | 4 | Igual | 4 | Mejor |
| Individuo 3 | Estar sentado | De concentración | 5 | Mejor | 3 | Peor |
| | Estar de pie | De concentración | 4 | Igual | 2 | Igual |
| Individuo 4 | Estar sentado | De concentración | 5 | Igual | 1 | Peor |
| | Estar de pie | De concentración | 4 | Mejor | 3 | Mejor |
| Individuo 5 | Estar sentado | Relajante | 3 | Igual | 5 | Mejor |
| | Estar de pie | De concentración | 3 | Igual | 3 | Mejor |
| Individuo 6 | Estar sentado | Relajante | 4 | Mejor | 4 | Mejor |
| | Estar de pie | Relajante | 4 | Igual | 5 | Mejor |
| Individuo 7 | Estar sentado | De concentración | 3 | Igual | 2 | Igual |
| | Estar de pie | De concentración | 2 | Mejor | 2 | Igual |

Figura 6. Comparación de resultados (actividades sin movimiento)

| | Actividad | Sin usar el sistema de recomendación | | Usando el sistema de recomendación | |
|-------------|-----------------|--------------------------------------|-----------------|------------------------------------|-----------------|
| | | RPE | Estado de ánimo | RPE | Estado de ánimo |
| Individuo 1 | Correr | 8 | Mejor | 6 | Mejor |
| | Subir escaleras | 7 | Peor | 7 | Igual |
| | Bajar escaleras | 4 | Igual | 4 | Igual |
| | Caminar | 3 | Igual | 2 | Mejor |
| Individuo 2 | Correr | 7 | Mejor | 5 | Mejor |
| | Subir escaleras | 5 | Igual | 4 | Mejor |
| | Bajar escaleras | 3 | Igual | 3 | Igual |
| | Caminar | 3 | Peor | 3 | Mejor |
| Individuo 3 | Correr | 5 | Igual | 5 | Mejor |
| | Subir escaleras | 6 | Igual | 7 | Igual |
| | Bajar escaleras | 4 | Igual | 3 | Mejor |
| | Caminar | 3 | Igual | 2 | Mejor |
| Individuo 4 | Correr | 7 | Peor | 7 | Mejor |
| | Subir escaleras | 5 | Mejor | 4 | Mejor |
| | Bajar escaleras | 3 | Igual | 3 | Igual |
| | Caminar | 2 | Igual | 1 | Mejor |
| Individuo 5 | Correr | 6 | Igual | 5 | Igual |
| | Subir escaleras | 7 | Igual | 7 | Mejor |
| | Bajar escaleras | 3 | Igual | 2 | Igual |
| | Caminar | 3 | Peor | 2 | Mejor |
| Individuo 6 | Correr | 6 | Mejor | 4 | Mejor |
| | Subir escaleras | 7 | Igual | 6 | Mejor |
| | Bajar escaleras | 2 | Igual | 2 | Mejor |
| | Caminar | 2 | Peor | 2 | Mejor |
| Individuo 7 | Correr | 7 | Igual | 5 | Mejor |
| | Subir escaleras | 5 | Mejor | 5 | Igual |
| | Bajar escaleras | 3 | Igual | 4 | Igual |
| | Caminar | 2 | Igual | 1 | Mejor |

Figura 7. Comparación de resultados (actividades con movimiento)

Sin embargo, en la caso de las actividades sin movimiento, y cuando la actividad realizada requería concentración los usuarios indicaban que conseguían mayor concentración sin música de ningún tipo. Por otro lado, para la actividades que tenía que ver con la relajación, la opinión de los usuarios está dividida la mitad dice que logra relajarse más con el sistema de recomendación y la otra mitad indica que es igual de relajante usar o no usar el sistema de recomendación. Pero todos aquellos que realizaban una actividad relajante mientras usaban el sistema de recomendación indicaron que mejoraba su estado de ánimo al terminar la actividad.

En base a esto se podría concluir que este sistema de recomendación es más indicado para las actividades que requieran movimiento o que sean relajantes.

5. Conclusiones y Líneas Futuras

La red neuronal de tipo LSTM demostró dar buenos resultados como clasificador de actividades, teniendo en cuenta tanto los resultados de sus métricas de calidad como las pruebas del estudio experimental.

Por otro lado, el estudio experimental permitió demostrar que el fenómeno del *entrainment* es muy útil para ligar las actividades de los usuarios con la música. Esto se puede apreciar en el hecho de que en general los usuarios indicaron que tendían a sincronizar su paso al tempo de la música, haciendo de la actividad más entretenida.

Teniendo en cuenta los resultados del estudio experimental, el sistema de recomendación resultó ser más útil a la hora de realizar actividades físicas o para ayudar al usuario a relajarse y mejorar su estado de ánimo. Sin embargo, para las actividades que requieran más concentración los usuarios en su mayoría preferían no escuchar nada.

Basándose en esto, a futuro se podría adaptar el sistema a los usuarios, realizando algún tipo de encuesta para realizar un perfil de usuario antes de que este puede hacer uso del reproductor. Aquí se podrían incluir preguntas sobre las actividades cotidianas y durante cuáles preferirían escuchar música. Este paso se incluiría la primera vez que el usuario acceda a la aplicación, de esta manera se puede seguir conservando la idea de la mínima interacción del usuario con el sistema, característica que se considera útil para que el usuario pueda realizar la actividad que quiera sin interrupciones.

Además, se tendría que continuar investigando para lograr implementar dentro de la aplicación Android el modelo que utilizaba la clasificación mediante un árbol de decisión de forma que la predicción de la actividad sea aún más fiable.

Por otro lado, se puede implementar un sistema que use las nuevas valoraciones de los usuarios y a partir de ellas se mejore la forma de recomendar. Se podrían tener en cuenta otras características para la valoración de canciones, como si una canción se salta rápidamente o se escucha entera.

A mayores, se podría aplicar alguna técnica de generación de *playlists* que permita, si el usuario quiere, escoger una *playlist* específica para alguna actividad.

Referencias

- [1] G. Bonnin y D. Jannach. «Automated generation of music playlists: Survey and experiments». En: *ACM Computing Surveys (CSUR)* 47.2 (2015), pág. 26.
- [2] R. Burke. «Hybrid recommender systems: Survey and experiments». En: *User modeling and user-adapted interaction* 12.4 (2002), págs. 331-370.
- [3] J. Fang, D. Grunberg, S. Luit e Y. Wang. «Development of a music recommendation system for motivating exercise». En: *2017 International Conference on Orange Technologies (ICOT)*. IEEE. 2017, págs. 83-86.
- [4] C. Gomes, M. M. García y A. G. González. «Sistema de recomendación de canciones basado en aspectos emocionales». En: *Trabajo Fin de Máster, Máster en Sistemas Inteligentes, Universidad de Salamanca*. 2018.
- [5] C. Greenhalgh, A. Hazzard, S. McGrath y S. Benford. «GeoTracks: Adaptive music for everyday journeys». En: *Proceedings of the 24th ACM international conference on Multimedia*. ACM. 2016, págs. 42-46.
- [6] Y. Hu. «A music recommendation system based on user behaviors and genre classification». En: (2012).
- [7] M. Kaminskis, F. Ricci y M. Schedl. «Location-aware music recommendation using auto-tagging and hybrid matching». En: *Proceedings of the 7th ACM conference on Recommender systems*. ACM. 2013, págs. 17-24.
- [8] J. P. Lopes-Silva, A. E. Lima-Silva, R. Bertuzzi y M. D. Silva-Cavalcante. «Influence of music on performance and psychophysiological responses during moderate-intensity exercise preceded by fatigue». En: *Physiology & behavior* 139 (2015), págs. 274-280.
- [9] B. Moens, L. van Noorden y M. Leman. «D-jogger: Syncing music with walking». En: *7th Sound and Music Computing Conference*. Universidad Pompeu Fabra. 2010, págs. 451-456.
- [10] M. N. Moreno-García. «Tema 3. Minería de uso web». 2018.
- [11] N. Oliver y L. Kreger-Stickles. «Enhancing exercise performance through real-time physiological monitoring and music: a user study». En: *2006 Pervasive Health Conference and Workshops*. IEEE. 2006, págs. 1-10.

- [12] D. Sánchez-Moreno, A. B. G. González, M. D. M. Vicente, V. F. L. Batista y M. N. M. Garca. «A collaborative filtering method for music recommendation using playing coefficients for artists and users». En: *Expert Systems with Applications* 66 (2016), págs. 234-244.
- [13] A. Sen y M. Larson. «From Sensors to Songs: A Learning-Free Novel Music Recommendation System using Contextual Sensor Data.» En: *LocalRec@ RecSys*. 2015, págs. 40-43.
- [14] E. Van Dyck, B. Moens, J. Buhmann, M. Demey, E. Coorevits, S. Dalla Bella y M. Leman. «Spontaneous entrainment of running cadence to music tempo». En: *Sports medicine-open* 1.1 (2015), pág. 15.
- [15] J. G. Vicente, A. B. Gil, A. de Luis Reboredo, D. Sánchez-Moreno y M. N. Moreno-Garca. «Moodsically. Personal music management tool with automatic classification of emotions». En: *International Symposium on Distributed Computing and Artificial Intelligence*. Springer. 2018, págs. 112-119.

Diseño de pipelines para el análisis y extracción de conocimiento a partir de exomas individuales

Alejandro Rodríguez Díaz¹, Luis Antonio Miguel Quintales²

¹VisUSAL (GRIAL), Universidad de Salamanca,
Paseo de Canalejas, 169 37008 Salamanca, España

²Departamento de informática y automática,
Facultad de Ciencias,
Universidad de Salamanca,
Plaza de los Caídos s/n 37008 Salamanca
{jancho,lamq}@usal.es

Resumen La secuenciación del genoma de un individuo y su posterior análisis permite detectar variaciones que puedan repercutir en el padecimiento de distintas enfermedades. El campo de la medicina de precisión se centra en el estudio de estas variaciones, haciendo uso de herramientas bioinformáticas con este fin. El incremento en las capacidades de secuenciación, la capacidad de procesamiento de los equipos actuales, y la disponibilidad de nuevas herramientas software para la manipulación de esta información ha hecho que cobre más interés compartir protocolos de análisis y automatizar los mismos. En este trabajo se ha diseñado e implementado la automatización de un flujo de trabajo que permite analizar de forma efectiva el análisis de variantes para experimentos de exoma completo: un tipo de análisis que únicamente utiliza las regiones del ADN que codifican proteínas.

Keywords: WES pipeline process automate

1. Introducción

La genómica es un campo dentro de la biología molecular que se centra en el estudio de la estructura de genomas completos¹ (el conjunto de genes para una especie). Dentro de este campo, el área de la medicina de precisión persigue la individualización del tratamiento: poder desarrollar o ajustar un tratamiento específico para las condiciones concretas del paciente cuya información genética se ha obtenido previamente a través de la secuenciación² y análisis de su genotipo o exoma³.

Un ejemplo de aplicación de la medicina de precisión es el caso de la hipercolesterolemia familiar, un desorden que incrementa el riesgo de sufrir ataques cardíacos. Se trata de un rasgo autosómico dominante; es decir, basta con heredar un gen para presentar la enfermedad. Sin embargo, la diagnosis y el tratamiento temprano es capaz de reducir en un 80% el riesgo de sufrir ataques cardíacos [16]. Una diagnosis en la que un análisis genómico es clave.

Este tipo de análisis, que compara el genotipo de un individuo con un genoma de referencia, se denomina análisis de variantes de línea germinal; un proceso que busca alteraciones —en adelante

¹ A diferencia de la genética, cuyo estudio se centra en los mecanismos de la herencia.

² La secuenciación consiste en obtener lecturas de los nucleótidos que posteriormente se mapearán contra un genoma de referencia de forma que se pueda saber qué valores tiene el individuo para diferentes posiciones conocidas.

³ Únicamente las regiones de los genes que codifican proteínas.

variantes— (sustituciones, inserciones y deleciones) en las secuencias de ADN del individuo que pueden afectar al funcionamiento de organismo o incluso a la cadena de aminoácidos de un gen, y por tanto en la proteína que produce.

En el caso de las variantes de línea germinal, éstas se transmiten de forma hereditaria. Razón por la que es posible la comparación con un genoma de consenso, el genoma de referencia.

Sin embargo no es necesario analizar el genoma completo para un estudio enfocado a las variantes que afectan a la expresión de los genes.

El exoma es el conjunto de fragmentos codificantes (exón) de cada gen del genoma, una cantidad de información genómica mucho menor que la presente en los estudios de genoma completo. En el caso del genoma humano, el exoma está formado por aproximadamente 180.000 exones; una cantidad que constituye menos del 2 % del genoma humano y que sin embargo, contiene aproximadamente un 85 % de las alteraciones de ADN relacionadas con enfermedades.

Esto se traduce en una reducción de los tamaños de fichero involucrados en el análisis (4-5 Gb frente a 90Gb de un genoma completo⁴), y el consecuente decremento del precio de secuenciación y coste computacional del análisis.

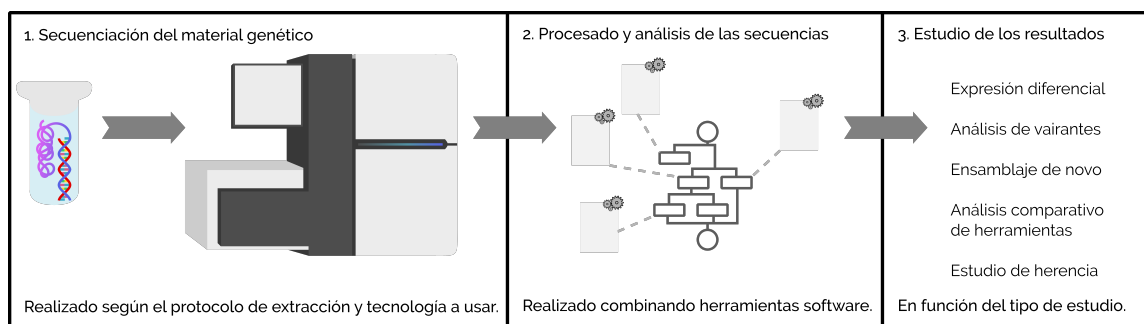


Figura 1. Proceso de análisis de variantes. *Elaboración propia.*

El análisis de variantes involucra tres etapas básicas⁵ (ver Fig. 1):

1. Secuenciación del genoma o exoma (en nuestro caso exoma) para obtener los ficheros FASTQ que contienen las lecturas de los nucleótidos.
2. Alineamiento de las secuencias contra el genoma de referencia para obtener los ficheros SAM o CRAM.
3. Identificar dónde difieren los genotipos. En un caso comparando la secuenciación con el genoma de referencia, y en otro caso entre dos tejidos del mismo organismo.

Las dos últimas etapas (el alineamiento y la detección de variantes) involucran una gran cantidad de pasos y herramientas informáticas que han crecido en número debido al auge de la medicina de

⁴ Según cálculos realizados por Illumina, referenciados en <https://www.illumina.com/techniques/sequencing/dna-sequencing/targeted-resequencing/exome-sequencing.html>.

⁵ <https://www.ebi.ac.uk/training/online/course/human-genetic-variation-i-introduction/variant-identification-and-analysis/what-variant>.

precisión, . La automatización de estos procesos facilita aplicar de forma consistente el mismo proceso de análisis, reduciendo el número de herramientas que es necesario conocer, y proporcionando otras ventajas como la paralelización del procesamiento y la monitorización de la ejecución.

Para la automatización existen multitud de alternativas como son el uso *scripts* de Unix⁶, usar sistemas basados en objetivos y reglas como *GNU Make*, usar herramientas externas para la automatización, o incluso desarrollar una plataforma propia.

El enfoque propuesto en este trabajo consiste en el uso de la librería de Python Luigi⁷ para la automatización de un proceso de análisis modular, fácil de extender, y que cubra la monitorización y escalabilidad de la ejecución.

2. Trabajo Previo

Una de las consecuencias del auge de la medicina de precisión ha sido el incremento en herramientas para el análisis de datos genómicos mencionado en la sección 1. Esto ha propiciado además, el desarrollo de herramientas que pretenden simplificar y/o automatizar el análisis, facilitando su uso.

Si embargo, la falta de estándares (aunque existen manuales de buena praxis) en el análisis se ha visto también reflejado en la falta de consenso en cuanto al enfoque para la automatización de experimentos: existen trabajos que comparten el protocolo usado mediante una descripción textual del proceso y la distribución de *scripts*[33], mientras otros trabajos que buscan la encapsulación del protocolo por medio de algún sistema informático.

Dentro de las soluciones informáticas tampoco hay una homogeneidad en las estrategias usadas; entre las alternativas más comunes están el desarrollo de plataformas propias [5, 7, 12], la implementación y distribución de las *pipelines* como programas [32], el desarrollo de módulos/*pipelines* para plataformas bioinformáticas existentes como proponen [8, 24, 31] para la plataforma Galaxy [10], o el uso de contenedores de Docker o máquinas virtuales para distribuir protocolos junto a un entorno de ejecución aislado y bien definido [2, 17, 22, 30].

Un denominador común es el uso de *pipelines*: diseñar el análisis o basar la ejecución del análisis en base a procesos en cadena definidos por secuencias de pasos. También existen diferencias la interfaz que proporcionan para la ejecución de la pipeline. Dos de los enfoques más comunes son el uso de interfaces gráficas [7, 12, 15, 18, 22, 27, 30, 32] para el manejo o monitorización de la pipeline, o el uso exclusivo de línea de comandos [1-3, 19].

Un enfoque intermedio se consigue con la distribución de las *pipelines* como código [11], que permite su uso por línea de comandos o su integración en *frameworks* que proporcionen interfaz gráfica como son Luigi, Apache Airflow, o Jenkins.

A pesar de la variedad de enfoques uno de los problemas identificados en los distintos enfoques revisados es el gran acoplamiento entre el la herramienta implementada y el experimento para el que se ha diseñado. Pocos trabajos enfocan el diseño e implementación de forma modular y con la intencionalidad de permitir su modificación o extensión; una decisión que impide aprovechar las ventajas del enfoque proporcionado en otros experimentos.

La falta de una solución que proporcione las ventajas de las interfaces web para la monitorización, la flexibilidad de la ejecución por línea de comandos, y la posibilidad de extenderla e incorporar nuevas estrategias como el uso de contenedores ha motivado la implementación de la herramienta que se presenta en este trabajo.

⁶ Ficheros de texto plano con órdenes que permiten interactuar con el sistema operativo.

⁷ <https://luigi.readthedocs.io/en/stable/>

3. Objetivos

Los objetivos del trabajo (listados a continuación) atienden al diseño e implementación de una estrategia que permita la automatización de un análisis de exomas completo, con un diseño modular para que un investigador pueda, en última instancia, adaptar (modificando, extendiendo, o usando parcialmente) el análisis a sus necesidades (ver Fig. 2).

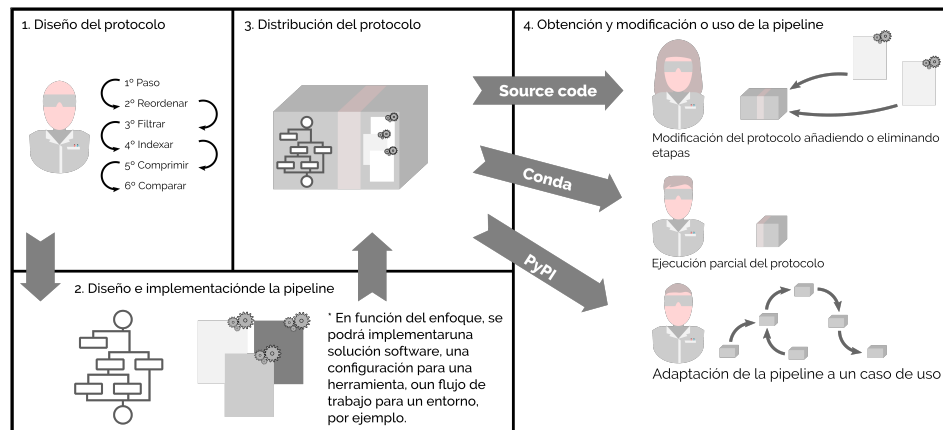


Figura 2. Proceso de diseño, implementación y distribución de análisis a cubrir por la solución propuesta.

Con esto en mente, se han planteado los siguientes objetivos:

- El diseño e implementación de una pipeline que realice el análisis de resultados de secuenciación de exoma completo.
- El diseño e implementación de una solución que cubra la automatización y monitorización del análisis.
- El diseño e implementación de una solución basada en la estrategia de *pipeline como software*, que exponga el código, sea modular, y fácilmente modificable.

4. Materiales y Métodos

4.1. Framework Usado

Para la implementación de la pipeline se ha elegido usar un *framework* que permita distribuir el análisis como un paquete instalable, que requiera el menor número de dependencias posibles, y que proporcione una interfaz para la monitorización.

Para este fin se ha elegido usar Luigi, un paquete de Python desarrollado por Spotify. Uno de los *frameworks* para la implementación y ejecución de *pipelines* más usados, y que se ha considerado una buena alternativa para automatizar procesos bioinformáticos en la revisión de herramientas realizada en [6]. Entre las ventajas que aporta el uso de Luigi están:

- Ejecución única de tareas. Luigi evita la ejecución de dos tareas que produzcan la misma salida (aún en *pipelines* separadas).

- Gestión del acceso a los recursos. Luigi proporciona mecanismos para simplificar el acceso a los recursos, y controlar el acceso concurrente.
- Histórico. Permite conectar con una base de datos relacional y registrar la ejecución de las tareas.
- Interfaz web. Proporciona una interfaz web (ver Fig. 4) para la monitorización de la ejecución de las tareas.
- Flexibilidad de implementación. Es posible crear tareas escritas en Python, usar herencia múltiple para añadir funcionalidad a tareas existentes sin cambiar su implementación, o usar programas externos.

Otra ventaja que ofrece es el uso del lenguaje Python, disponible por defecto en cualquier sistema Unix; evitando incorporar un nuevo lenguaje de descripción y especificación (SDL) como puede ser WDL⁸.

4.2. Requisitos y Dependencias

La implementación de la pipeline con Luigi requiere tener disponible el intérprete para Python versión 3.6, y los paquetes:

- luigi en la versión 2.8.7 o superior.
- python-daemon en la versión 2.2.0 o superior.

El resto de dependencias son las herramientas bioinformáticas usadas durante el proceso de análisis. Teniendo en cuenta que la ejecución de las tareas depende de qué etapa del análisis se elija, los parámetros de ejecución, y la existencia de resultados intermedios, es posible que no todas las herramientas se necesiten. En concreto, las herramientas usadas se muestran a continuación (ver Cuadro 1 para la relación de versiones):

| Herramienta | Versión | Forma de instalación |
|-------------|---------|---|
| BWA | 0.7.17 | Manual, disponible en Conda |
| DeepVariant | 0.8.0 | Automática. La imagen de Docker se descarga automáticamente |
| FastqC | 0.11.8 | Manual, disponible en Conda |
| Freebayes | 1.2.0 | Manual, disponible en Conda |
| GATK | 4.1.2.0 | Automática. La imagen de Docker se descarga automáticamente |
| Gunzip | 1.6 | Disponible en sistemas Unix |
| Piccard | 2.20.2 | Manual, disponible en Conda |
| Platypus | 0.8.1.2 | Manual, disponible en Conda |
| Samtools | 1.9 | Manual, disponible en Conda |
| Sra tools | 2.9.1_1 | Manual, disponible en Conda |
| TwoBit ToFa | | Automática. Se instala automáticamente en la pipeline |
| Varscan | 2.4.3 | Manual, disponible en Conda |
| VCFTools | 0.1.16 | Manual, disponible en Conda |

⁸ <https://software.broadinstitute.org/wdl/>

| | | |
|------|------------|-----------------------------|
| Vt | 2015.11.10 | Manual, disponible en Conda |
| Wget | 1.19.4 | Disponible en sistemas Unix |

Tabla 1: Correspondencia de versiones para las herramientas usadas.

5. Implementación

5.1. Análisis

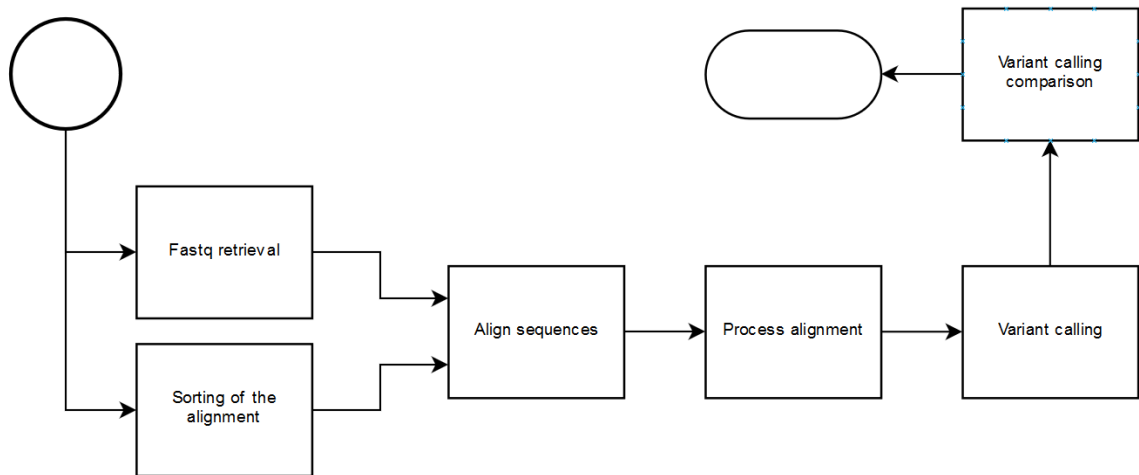


Figura 3. Diagrama de flujo del proceso de análisis de variantes para experimentos de exoma completo.

A partir de los pasos de análisis de la secuenciación (los dos últimos pasos) de la metodología mostrada en la sección 1 se han identificado 6 etapas que se muestran en la Fig. 3 : dos pasos de recuperación de los recursos para el análisis, dos pasos de alineamiento y procesado del mismo, y dos pasos de detección de variantes y su análisis.

1. **Obtención del genoma de referencia.** El genoma de referencia se utiliza en varias de las etapas del análisis. Éste sirve para alinear las lecturas obtenidas por el secuenciador, identificar variantes en el genotipo, y ordenar los ficheros de variantes. La ordenación se realiza con las utilidades *index* y *faidx* de Bwa [13] y Samtools [14] respectivamente. Y el fichero *dict* que se usará para la normalización de los ficheros de variantes usando la herramienta Picard [23].
2. **Obtención de las secuencias.** Se obtienen los ficheros FASTQ con la secuenciación del exoma a partir de una dirección de descarga, o el *accession number* del NCBI; esto es, un identificador único que se puede usar con las utilidades de descarga de SRA Toolkit [28] para obtener los FASTQ.

En función de si el experimento es *paired end*⁹ o no, se obtienen uno o dos ficheros.

⁹ Secuenciación que lee los fragmentos de ADN por ambos extremos

3. **Alineamiento de las secuencias.** La herramienta BWA [13] proporciona la utilidad *mem* que permite alinear el/los ficheros FASTQ contra el genoma de referencia. Es importante asegurarse de haber creado los ficheros de índice para el genoma de referencia para agilizar el procesamiento.
4. **Procesamiento del alineamiento.** Se obtiene un fichero BAM resultante de ordenar la alineación y eliminar los duplicados, y un fichero de índices BAI para agilizar el acceso a las diferentes posiciones genómicas. Para la ordenación y la creación de índices se usan las herramientas Samtools [14] y [13] respectivamente, y la utilidad *markduplicates* de Picard [23] para eliminar los duplicados.
5. **Detección de variantes.** La detección de variantes se puede realizar con un amplio abanico es un proceso delicado que involucra el genoma de referencia y los ficheros BAM y BAI con las secuencias.

Existen multitud de herramientas que permiten la detección de variantes, cada una con una estrategia distinta, que pueden llegar a obtener resultados distintos; bien por no identificar la misma variación, o por anotarla de forma distinta. En este trabajo se propone el uso de Varscan [21][20], Freebayes [9], GATK, DeepVariant[25], y Platypus [26].

6. **Análisis de las variantes detectadas.** Por último, el análisis de las variantes involucra la normalización de los ficheros de variantes, la obtención de información sobre la profundidad, y la comparación de variantes encontradas.

La normalización es un paso importante que permite obtener ficheros de variantes con el mismo ordenamiento de los cromosomas y la misma estrategia de anotación para las variantes¹⁰. Para esta tarea se usan las herramientas Picard [23] y Vt [29].

Para la comparativa entre los ficheros de variantes, se usa la herramienta VCFtools [4].

5.2. Implementación del Paquete Wespipeline

Una de las ventajas de los gestores de paquetes informáticos como Python Packaging Authority (PyPI) o Conda es la facilidad con la que se puede distribuir un proyecto usando sus repositorios. Por esta razón se ha implementado la pipeline como un paquete de Python siguiendo la guía propuesta por PyPI¹¹.

Tanto la lógica específica del dominio (la creación de comandos de ejecución para las herramientas) como el flujo de ejecución se codifican usando código en Python. De forma similar a las herramientas GNU Make y Snakemake, Luigi infiere las tareas a ejecutar y el orden de la ejecución a partir de las entradas y salidas de las distintas etapas. En el caso de Luigi las entradas y salidas, y la funcionalidad se encapsulan en clases denominadas tareas que utilizará para comprobar la ejecución de una tarea, obtener sus dependencias, y ejecutarla.

Siguiendo con esta estrategia, Luigi propone una infraestructura basada en tres elementos principales:

- **Tareas.** Cada tarea es una funcionalidad aislada, definida en una clase de Python, que define sus entradas, salidas, y la funcionalidad que proporciona de tal forma que se puede comprobar si se ha ejecutado correctamente, si es necesario y posible ejecutarla, y el código que se tendría que ejecutar.

¹⁰ La normalización involucra la simplificación de las anotaciones como por ejemplo transformar la anotación (AT ->TG) a dos anotaciones (A->T), (T->G).

¹¹ <https://packaging.python.org/tutorials/packaging-projects/>

```

class Etapa(utils.MetaOutputHandler, luigi.Task):

    param1 = luigi.Parameter(default="", description="")
    param2 = luigi.Parameter(default='', description="")

    def requires(self):
        dependencias = dict()

        dependencias.update({'a': Tarea1(param1=self.param1)})
        dependencias.update({'b': Tarea2(param2=self.param2)})

        return dependencias

    def run(self):
        yield TareaPostEjecucion(entrada=self.input()['a'])

```

Código 1: Ejemplo de implementación de una tarea de alto nivel. Ejemplo mostrando una tarea con varios parámetros y dependencias pre y post ejecución.

- Trabajador. Son instancias de Python usadas para ejecutar y notificar el estado de la ejecución de una tarea específica.
- Planificador. Es el módulo encargado de resolver las dependencias de las tareas lanzadas a ejecución, lanzar la ejecución de las mismas, y gestionar el correcto funcionamiento de la pipeline.

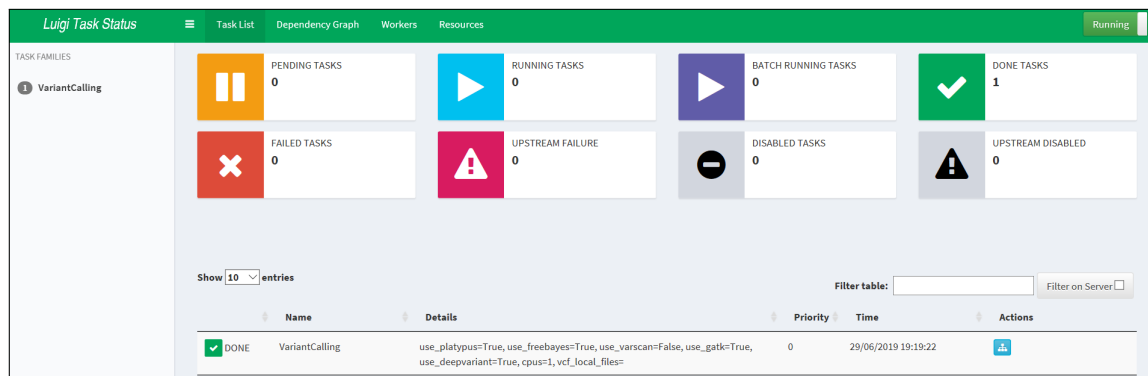


Figura 4. Panel de monitorización. Captura de pantalla de la herramienta Luigi.

La característica más distintiva del paquete implementado es su enfoque modular. Todos los módulos implementados tienen una tarea de alto nivel que expone los resultados de la etapa y permite añadir o quitar dependencias a la etapa de análisis de forma sencilla. Cada una de las tareas de alto nivel tienen una estructura similar a la mostrada en el código 1.

Todos los parámetros de ejecución de una etapa se recogen en esta clase, y las tareas involucradas en la etapa se especifican en los métodos *requires* y *run*, en función de si han de ejecutarse con la

etapa o tras su finalización respectivamente. En el código 1 se muestra cómo es posible acceder a las dependencias (entradas) usando el método *input* y la misma clave que se utilizó en el método *requires* para definir la dependencia.

La herencia múltiple mostrada en el código de ejemplo permite, a través de la clase *MetaOutputHandler*, permite propagar las entradas como salidas. De esta forma, Luigi evaluará si una etapa está completa en función de si todas las tareas de la etapa se han realizado, propagando los resultados para poder ser usados en otras tareas usando el nombre de la etapa y no el de las tareas específicas que proporcionan el resultado.

El objetivo de abstraer las etapas en vez de la pipeline completa como se propone en [1, 7, 8, 12, 32] y otros muchos trabajos, es implementar un protocolo de análisis que se pueda aplicar de forma parcial o completa. Para simplificar su uso, las tareas asociadas a una etapa toman el mayor número parámetros de la clase de alto nivel; esto permite configurar aspectos internos de una etapa sin conocer el nombre de sus tareas.

6. Resultados

El resultado de la implementación del análisis propuesto en la sección 5.1 es el paquete de Python *Wespipeline* cuyo código se encuentra disponible en el repositorio de Github <https://github.com/Janchorizo/wespipeline>, y accesible a través de los principales gestores de paquetes: está disponible en Pypi¹² y Conda¹³.

La instalación de la herramienta -que puede realizarse junto con el resto de las dependencias con un único comando (ver código 2)- permite lanzar la ejecución de *pipelines* (ver modelo de ejecución en la Fig. 5) especificando la tarea objetivo y los parámetros necesarios. Gracias a la implementación con Luigi, las dependencias¹⁴ se resolverán automáticamente y se paralelizará su ejecución en base al parámetro *workers* que se establezca.

| Módulo | Nombre de la tarea | Descripción |
|-----------|--------------------|--|
| reference | ReferenceGenome | Obtener fichero FASTA con el genoma de referencia. |
| fastq | GetFastq | Obtener ficheros FASTQ con la secuencias, |
| align | FastqAlign | Alinear las secuencias obtenidas contra el genoma de referencia. |

¹² <https://pypi.org/project/wespipeline>

¹³ <https://anaconda.org/jancho/wespipeline>

¹⁴ En este caso, las dependencias se refieren a los resultados previos necesarios para ejecutar una tarea.

```
conda install -y samtools bwa picard platypus-variant && \
conda install -y varscan freebayes vcftools && \
conda install -y gatk vt fastqc && \
conda install -c jancho wespipeline
```

Código 2: Ejemplo de comando para la instalación de la herramienta junto con el resto de dependencias.

| | | |
|--------------|------------------------|---|
| processalign | AlignProcessing | Procesar la alineación de secuencias. |
| vcf | VariantCalling | Identificar variantes en las secuencias usando varias herramientas. |
| vcfanalysis | VariantCallingAnalysis | Realizar estadísticas y comparativas entre ficheros de variantes. |

Tabla 2: Relación de módulos y tareas del paquete Wespipeline.

Cada una de las etapas propuestas en la sección 5.1 tiene un módulo y una tarea objetivo asociada (ver Cuadro 2) que se ejecuta, junto con los parámetros deseados, a través de la línea de comandos (ver ejemplo de comando en el código 3).

En la línea de comandos se pueden especificar parámetros para cualquier tarea involucrada. De forma que si se usan los mismos parámetros para una tarea, su ejecución será la misma tanto para su ejecución aislada (ver Código 4), como para su ejecución en otra pipeline (ver Código 5).

Con el planificador de Luigi funcionando, se puede acceder a la interfaz web de Luigi a través del puerto 8082 de la máquina, y comprobar el estado de la ejecución de nuestras tareas. La interfaz ofrece además un grafo mostrando la ejecución de la pipeline (ver ejemplo de detección de variantes en la Fig. 5).

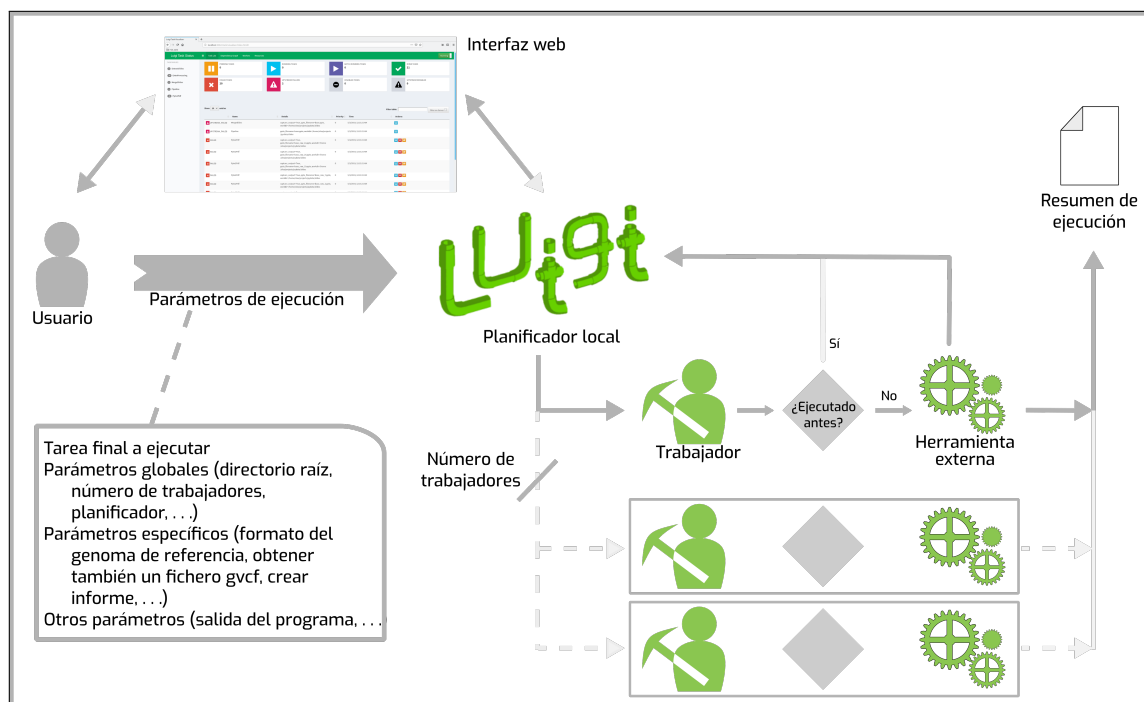


Figura 5. Modelo de ejecución de la pipeline bioinformática. Fuente: producción propia.

```
luigi --module wespipeline.<modulo> <tarea objetivo> \
--<tarea objetivo>-<nombre parametro 1> <valor parametro 1> \
--<tarea objetivo>-<nombre parametro 2> <valor parametro 2>
```

Código 3: Ejemplo de comando para la ejecución de una tarea de Wespipeline.

```
luigi --module wespipeline.fastq GetFastq \
--GlobalParams-base-dir . --GlobalParams-log-dir . --GlobalParams-exp-name hg19 \
--GetFastq-create-report True --GetFastq-paired-end False \
--GetFastq-fastq1-url ftp://example.domain.com/fastq \
--workers 3
```

Código 4: Ejemplo llamada para la ejecución de la tarea *GetFastq*.

A diferencia de otras herramientas que obligan a la ejecución completa o pre-instalan los paquetes , Wespipeline permite aplicar únicamente aquellas etapas necesarias para la tarea a realizar. Por tanto las dependencias para la ejecución varían en función de las tareas que se ejecutan finalmente en función de los parámetros especificados; si únicamente se va a realizar la alineación no es necesario tener VCFtools instalado.

Todas las dependencias se pueden instalar bien directamente junto con la herramienta si se usa Conda, o automáticamente en tiempo de ejecución si es una imagen de Docker. Por tanto, para el uso de la herramienta no es necesario tener mayores conocimientos informáticos que un uso básico de terminales Unix.

```
luigi --module wespipeline.reference FastqAlign \
--GlobalParams-base-dir . --GlobalParams-log-dir . --GlobalParams-exp-name hg19 \
--FastqAlign-cpus 6 \
--GetFastq-paired-end False \
--GetFastq-fastq1-urlftp://example.domain.com/fastq \
--ReferenceGenome-from2bit True \
--ReferenceGenome-ref-url ftp://example.domain.com/reference \
--workers 3
```

Código 5: Ejemplo de llamada para establecer los parámetros de la tarea intermedia *GetFastq*.

7. Conclusiones

En este trabajo se ha presentado el diseño e implementación de una solución novedosa para un problema actual en el ámbito de la medicina de precisión: la automatización y distribución de un protocolo de análisis de variantes usando *pipelines*.

La solución propuesta implementa un análisis de variantes completo fácilmente configurable. Además de múltiples parámetros para cada una de las etapas incluidas, el diseño modular y accesible (su modificación no requiere de compilación ni entorno de desarrollo) facilita su modificación y extensión; un aspecto que no es común en otras estrategias.

Una de las ventajas del uso de Wespipeline es la cantidad mínima de dependencias necesarias, y la facilidad de su instalación junto con el resto de herramientas al haberse distribuido en PyPI y Conda.

La mayor aportación del paquete Wespipeline es la estrategia de integración que incorpora. Una solución enfocada a añadir etapas y pasos con herramientas de distinta naturaleza haciendo que la ejecución se más sencilla a través de una interfaz de línea de comandos común, y la monitorización más cómoda con la interfaz web de Luigi.

Referencias

- [1] K. Bhuvaneshwar, L. Song, S. Madhavan e Y. Gusev. «viGEN: An Open Source Pipeline for the Detection and Quantification of Viral RNA in Human Tumors». En: *Frontiers in Microbiology* 9 (2018). DOI: 10.3389/fmicb.2018.01172.
- [2] J. L. Causey, C. Ashby, K. Walker, Z. P. Wang, M. Yang, Y. Guan, J. H. Moore y X. Huang. «DNAP: A Pipeline for DNA-seq Data Analysis». En: *Scientific Reports* 8 (2018). DOI: 10.1038/s41598-018-25022-6.
- [3] M. I. Cornwell, M. Vangala, L. Taing, Z. Herbert, J. Köster, B. Li, ... y H. W. Long. «VIPER: Visualization Pipeline for RNA-seq, a Snakemake workflow for efficient and complete RNA-seq analysis». En: *BMC Bioinformatics* 19.1 (2018). DOI: 10.1186/s12859-018-2139-9.
- [4] P. Danecek, A. Auton, G. Abecasis, C. A. Albers, E. Banks, M. A. DePristo, ... y R. Durbin. «The variant call format and tools». En: *Bioinformatics* 27.15 (2011), págs. 2156-2158. DOI: 10.1093/bioinformatics/btr330.
- [5] P. Di Tommaso, E. W. Floden, C. Magis, E. Palumbo y C. Notredame. «Nextflow, an efficient tool to improve computation numerical stability in genomic analysis ». En: *Biologie Aujourd'hui* 211.3 (2017), págs. 233-237. DOI: 10.1051/jbio/2017029.
- [6] M. P. Dolled-Filhart, M. Lee Jr., C.-w. Ou-yang, R. R. Haraksingh y J. C.-H. Lin. «Computational and Bioinformatics Frameworks for Next-Generation Whole Exome and Genome Sequencing». En: *Scientific World Journal* (2013). DOI: 10.1155/2013/730210.
- [7] Y. Dufresne, F. Lejzerowicz, L. A. Perret-Gentil, J. Pawlowski y T. Cordier. «SLIM: a flexible web application for the reproducible processing of environmental DNA metabarcoding data». En: *Bmc Bioinformatics* 20 (2019). DOI: 10.1186/s12859-019-2663-2.
- [8] C. Farkas, F. Fuentes-Villalobos, B. Rebolledo-Jaramillo, F. Benavides, A. F. Castro y R. Pincheira. «Streamlined computational pipeline for genetic background characterization of genetically engineered mice based on next generation sequencing data». En: *BMC Genomics* 20.1 (2019). DOI: 10.1186/s12864-019-5504-9.
- [9] E. Garrison y G. Marth. «Haplotype-based variant detection from short-read sequencing». En: (2012). arXiv: 1207.3907.

- [10] J. Goecks, A. Nekrutenko, J. Taylor y T. Galaxy. «Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences». En: *Genome Biology* 11.8 (2010). DOI: 10.1186/gb-2010-11-8-r86.
- [11] J. B. Goll, T. L. Jensen, M. Frasketi, K. Conway, L. Villarroel, H. Hill y K. Krampis. «RSEQREP: RNA-Seq Reports, an open-source cloud-enabled framework for reproducible RNA-Seq data processing, analysis, and result reporting [version 2; referees: 2 approved]». En: *F1000Research* 6 (2018). DOI: 10.12688/f1000research.13049.2.
- [12] B. A. Gruening, J. Fallmann, D. Yusuf, S. Will, A. Erxleben, F. Eggenhofer, ... y R. Backofen. «The RNA workbench: best practices for RNA and high-throughput sequencing bioinformatics in Galaxy». En: *Nucleic Acids Research* 45.W1 (2017), W560-W566. DOI: 10.1093/nar/gkx409.
- [13] L. H. y D. R. «Fast and accurate long-read alignment with Burrows-Wheeler transform». En: *Bioinformatics* (2010). DOI: <https://doi.org/10.1093/bioinformatics/btp698>.
- [14] B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin y 1000 Genome Project Data Processing Subgroup. «The Sequence Alignment/Map format and SAMtools.» En: *Bioinformatics (Oxford, England)* 25.16 (2009), págs. 2078-9. DOI: 10.1093/bioinformatics/btp352.
- [15] W. He, S. Zhao, C. Zhang, M. S. Vincent y B. Zhang. «QuickRNASeq: Guide for Pipeline Implementation and for Interactive Results Visualization». En: *Transcriptome Data Analysis: Methods and Protocols*. Ed. por Y. Wang y M. A. Sun. Vol. 1751. Methods in Molecular Biology. 2018, págs. 57-70. DOI: 10.1007/978-1-4939-7710-9_4.
- [16] M. J. Khoury, M. S. Bowen, M. Clyne, W. D. Dotson, M. L. Gwinn, R. F. Green, ... y W. Yu. «From public health genomics to precision public health: a 20-year journey». En: *Genetics In Medicine* 20 (dic. de 2017), pág. 574.
- [17] B. Kim, T. Ali, C. Lijeron, E. Afgan y K. Krampis. «Bio-Docklets: Virtualization containers for single-step execution of NGS pipelines». En: *GigaScience* 6.8 (2017). DOI: 10.1093/gigascience/gix048.
- [18] S. Kinjo, N. Monma, S. Misu, N. Kitamura, J. Imoto, K. Yoshitake, T. Gojobori y K. Ikeo. «Maser: one-stop platform for NGS big data from analysis to visualization». En: *Database-the Journal of Biological Databases and Curation* (2018). DOI: 10.1093/database/bay027.
- [19] M. Kluge y C. C. Friedel. «Watchdog - a workflow management system for the distributed analysis of large-scale experimental data». En: *BMC Bioinformatics* 19.1 (2018). DOI: 10.1186/s12859-018-2107-4.
- [20] D. C. Koboldt, Q. Zhang, D. E. Larson, D. Shen, M. D. McLellan, L. Lin, ... y R. K. Wilson. «VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing». En: *Genome Research* 22.3 (2012), págs. 568-576. DOI: 10.1101/gr.129684.111.
- [21] D. Koboldt, K. Chen, T. Wylie, D. Larson, M. McLellan, E. Mardis, ... y L. Ding. «VarScan: variant detection in massively parallel sequencing of individual and pooled samples». En: *Bioinformatics (Oxford, England)* 25.17 (2009), págs. 2283-2285.
- [22] C. Kwon, J. Kim y J. Ahn. «DockerBIO: Web application for efficient use of bioinformatics Docker images». En: *PeerJ* 2018.11 (2018). DOI: 10.7717/peerj.5954.
- [23] *Picard toolkit*. <http://broadinstitute.github.io/picard/>. 2019.
- [24] R. Pogorelnik, C. Vaury, P. Pouchin, S. Jensen y E. Brassat. «SRNAPipe: A Galaxy-based pipeline for bioinformatic in-depth exploration of small RNAseq data». En: *Mobile DNA* 9.1 (2018). DOI: 10.1186/s13100-018-0130-7.

- [25] R. Poplin, P.-C. Chang, D. Alexander, S. Schwartz, T. Colthurst, A. Ku, ... y M. A. DePristo. «A universal SNP and small-indel variant caller using deep neural networks». En: *Nature Biotechnology* 36.10 (sep. de 2018), pág. 983. DOI: 10.1038/nbt.4235.
- [26] A. Rimmer, H. Phan, I. Mathieson, Z. Iqbal, S. R. F. Twigg, W. Consortium, ... y G. Lunter. «Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications». En: *Nature Genetics* 46.8 (2014), págs. 912-918. DOI: 10.1038/ng.3036.
- [27] J. Singer, H.-J. Ruscheweyh, A. L. Hofmann, T. Thurnherr, F. Singer, N. C. Toussaint, ... y N. Beerenwinkel. «NGS-pipe: a flexible, easily extendable and highly configurable framework for NGS analysis». En: *Bioinformatics* 34.1 (2018), págs. 107-108. DOI: 10.1093/bioinformatics/btx540.
- [28] S. R. A. S. Staff. *Using the SRA Toolkit to convert .sra files into other formats*. <https://www.ncbi.nlm.nih.gov/books/NBK158900/>. [Online; accessed 22-June-2019]. Bethesda (MD): National Center for Biotechnology Information (US); 2011.
- [29] A. Tan, G. R. Abecasis y H. M. Kang. «Unified representation of genetic variants». En: *Bioinformatics* 31.13 (feb. de 2015), págs. 2202-2204. DOI: 10.1093/bioinformatics/btv112. eprint: <http://oup.prod.sis.lan/bioinformatics/article-pdf/31/13/2202/17122489/btv112.pdf>.
- [30] D. Torre, A. Lachmann y A. Ma'ayan. «BioJupies: Automated Generation of Interactive Notebooks for RNA-Seq Data Analysis in the Cloud». En: *Cell Systems* 7.5 (2018), 556-561.e3. DOI: 10.1016/j.cels.2018.10.007.
- [31] M. Wolfien, C. Rimbach, U. Schmitz, J. J. Jung, S. Krebs, G. Steinhoff, R. David y O. Wolkenhauer. «TRAPLINE: a standardized and automated pipeline for RNA sequencing data analysis, evaluation and annotation». En: *Bmc Bioinformatics* 17 (2016). DOI: 10.1186/s12859-015-0873-9.
- [32] R. Wurmus, B. Uyar, B. Osberg, V. Franke, A. Godschan, K. Wreczycka, J. Ronen y A. Akalin. «PiGx: Reproducible genomics analysis pipelines with GNU Guix». En: *GigaScience* 7.12 (2018). DOI: 10.1093/gigascience/giy123.
- [33] H. K. Yalamanchili, Y.-W. Wan y Z. Liu. «Data analysis pipeline for RNA-seq experiments: From differential expression to cryptic splicing». En: *Current Protocols in Bioinformatics* 2017 (2017), págs. 11.15.1-11.15.21. DOI: 10.1002/cpbi.33.

Sistema de predicción de mercados de criptomonedas

Diego Valdeolmillos¹, Pablo Chamoso¹, Juan M. Corchado¹

¹BISITE Digital Innovation Hub, Universidad de Salamanca. Edificio Multiusos I+D+i,37007, Salamanca, Spain,
{dval,chamoso,corchado}@usal.es

Resumen Tras el nacimiento de Bitcoin y el auge de la tecnología Blockchain ha aparecido un nuevo mercado de inversión con mayor volatilidad y mayores movimientos que los mercados tradicionales, es el mercado de las criptomonedas. El objetivo de este trabajo realizar un estudio de las técnicas tradicionales de predicción de mercado de valores en el mercado de criptomonedas, tanto en análisis técnico como fundamental aplicando algoritmos de inteligencia artificial como Random Forest, redes neuronales *long short-term memory* y análisis de sentimiento de textos, para alcanzar una aproximación de predicción del mercado.

Keywords: predicción de mercados, criptomonedas, análisis técnico, análisis fundamental, *long short-term memory*, *random forest*

1. Introducción

La predicción de precios de los mercados ha sido una tarea muy estudiada por investigadores y *traders*, su objetivo no es otro que obtener una buena inversión a través de herramientas automáticas que operen en las bolsas o bien como herramienta de apoyo que indique la dirección futura del mercado. Los estudios se centran en los mercados tradicionales pero las criptomonedas no se diferencian mucho de otros activos, y su precio se negocia libremente en el mercado, y se determina por la oferta y la demanda, pero, sin embargo, tienen demasiada volatilidad que produce que los intercambios en las casas de cambio de criptomonedas aumente su valor o disminuya drásticamente, esto es debido a que los tamaños de mercado son más pequeños en comparación con otros mercados de divisas tradicionales, por lo que pequeños movimientos pueden tener un efecto alcista o bajista muy grande, además la falta de capital institucional que si se invirtiese el suficiente dinero suavizaría la volatilidad.

Otro de los factores que afectan a la volatilidad es la percepción pública y depende mucho del sentimiento positivo o negativo de estas, y son afectadas por las noticias, las redes sociales, las regulaciones de los países, etc. Una cobertura positiva en los medios de comunicación llevará a Bitcoin a los precios más alto, puesto que conducen a una mejor comprensión de la criptomonedas por parte del público general, además con la gran utilización de las redes sociales, una noticia se extenderá rápidamente por las redes sociales. La falta de confianza en las economías de los países tiene un efecto de sentimiento más favorable a las criptomonedas sobre las monedas fiduciarias.

Bitcoin además tiene una gran influencia sobre el precio del resto de criptomonedas, debido a que la gran mayoría de casas de cambio ofrecen operar las criptomonedas alternativas (altcoins) por Bitcoin, en lugar de por FIAT (dinero cuyo valor se basa en su declaración por una entidad central) o criptomonedas estables como Tether (USDT).

En los mercados tradicionales existen los análisis de predicción técnicos y los fundamentales, los cuales ambos influyen en la dirección del mercado. Es por eso, que en este trabajo se van a estudiar estos dos tipos de análisis y cómo combinarlos con técnicas de inteligencia artificial, para intentar obtener buenos resultados para la predicción del mercado.

2. Estado del Arte

2.1. Blockchain

Blockchain es un sistema de almacenamiento de transacciones organizado en una secuencia de bloques, en la que cada bloque mantiene un puntero *hash* al bloque anterior. Bitcoin fue el primer desarrollo de Blockchain, y fueron surgiendo a partir de ella nuevos desarrollos para nuevos casos de uso.

Bitcoin. Es un software descentralizado formado por nodos en una red P2P que sirve como un sistema de pagos digital a nivel mundial, sin el control de ninguna autoridad central, para ello utiliza algoritmos criptográficos que son capaces de validar las transacciones, todos los participantes del software de Bitcoin son los que se encargan en dar confianza y seguridad a estas transacciones[21].

Un bloque en Bitcoin (Fig. 1) se mantiene inmutable debido a que es sellado en el tiempo y mantiene la información de las transacciones y la información del bloque previo[16].

Bitcoin utiliza para generar un nuevo bloque el algoritmo denominado Prueba de Trabajo (Proof of Work POW), donde los nodos que se dedican al minado generan bloques candidatos para unirse a la cadena de bloques, para minar un bloque es necesario resolver un problema criptográfico utilizando fuerza bruta cuya dificultad es directamente proporcional a la potencia de computo de la red P2P de Bitcoin, cuyo nivel de dificultad se va ajustando cada 2016 bloques (2 semanas) debido a que se busca que los bloques tarden 10 minutos en minarse[16]. El primer nodo que resuelve la prueba de trabajo difunde el nuevo bloque al resto de nodos de la red Bitcoin, estos lo añaden a su copia de la cadena de bloques local si es correcto y empiezan a resolver la prueba para el siguiente bloque[21].

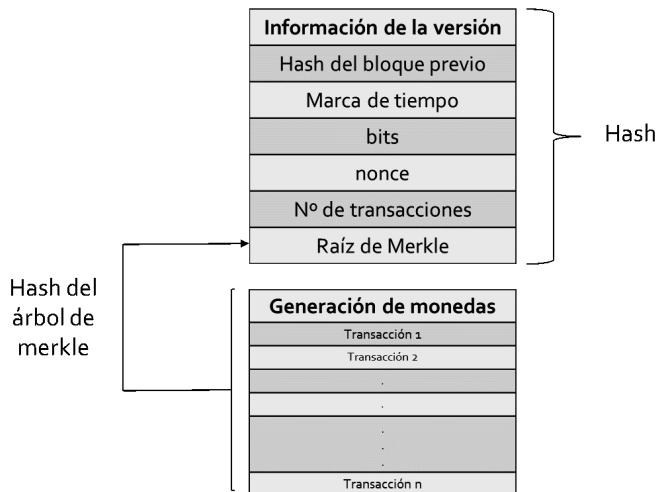


Figura 1. Estructura de un bloque de Bitcoin

Bitcoin fue diseñado para cumplir ciertas características deseables para ser un buen depósito de valor, algunas de estas características son [21] [19]:

- No es caduco ni fácil de destruir.
- Es fácil de almacenar y transportar.
- Es fácil de verificar.
- Es fácil de dividir.
- No es abundante ni fácil de producir.
- Bitcoin aún no está arraigado en el tiempo ni en la sociedad como depósito de valor.

Criptomonedas. En 2011 se lanza la criptomoneda Litecoin, imitando a su progenitora Bitcoin pero buscando transacciones más rápidas, desde ahí han ido surgiendo muchas más monedas llegando a 2479 criptomonedas [5], existiendo una capitalización bursátil a 27 de junio de 2019 de 275.803.293.070€ y un volumen de intercambio diario de 115.900.802.288 €.

2.2. Análisis Técnico

El análisis técnico [2], es un método en el que se utilizan los valores pasados de los mercados para intentar predecir el valor futuro, sin tener en cuenta ningún factor externo y se consideran únicamente como suposiciones las siguientes [7]:

- Los precios están determinados únicamente por la oferta y la demanda.
- Las tendencias influyen en las variaciones de precios.
- Las variaciones en oferta y demanda implican que las tendencias se inviertan.
- Los movimientos en la oferta y la demanda se pueden detectar en los gráficos.
- Los patrones que se observan en los gráficos tienden a repetirse.

2.3. Análisis Fundamental

Según otras teorías de marketing también los mercados bursátiles están afectados por factores macroeconómicos y ambientales, se denomina comúnmente análisis fundamental y utiliza el *framework* PESTEL para supervisar los factores del marketing externo que pueden afectar a una organización y por tanto afectar a su valor de mercado[22]. PESTEL significa [15]:

- P Político: Los factores políticos pueden intervenir en la economía, las políticas y restricciones comerciales que pueda imponer.
- E Económico: Factores macroeconómicos y microeconómicos.
- S Social: Los factores socioculturales influyen en los profesionales del marketing que han de ser capaces de explotarlos para obtener un beneficio.
- T Tecnológico: Estos factores influyen en la producción y distribución de bienes y servicios, así como en la comunicación hacia los mercados objetivo.
- E Medio Ambiental (*Environmental*): Los consumidores demandan que los productos que adquieren procedan de fuentes éticas y sostenibles.
- L Legal: Operar a nivel internacional significa adaptarse a las normativas de cada país.

2.4. Inteligencia Artificial Aplicada a la Predicción de Mercados

Existen diferentes estudios sobre estrategias para obtener beneficios de mercados bursátiles, entre ellas podemos encontrar algoritmos genéticos [11] [10], razonamiento basado en casos [5], clasificador de subespacio aleatorio [28], redes neuronales artificiales [29] o máquinas vectoriales [6]. Para este estudio se van analizar Random Forest y las redes neuronales profundas recurrentes Long Short-Term Memory.

Random Forest. La predicción de los retornos en un mercado de valores se presenta como un dilema, en que los precios se calculan y se predicen. Se busca clasificar los problemas prediciendo que a través de unos datos las cotizaciones bursátiles aumentarán o disminuirán respecto al precio anterior. Según la hipótesis de mercado eficiente [11], que establece que los precios del mercado se rigen por un proceso aleatorio y no pueden ser pronosticados como movimientos y comportamientos del mercado en momentos pasados, se puede ver como un problema de decisión y es posible la utilización de árboles de decisión.

Los bosques aleatorios o Random Forests son una serie de predictores de árboles de tal manera que cada uno de los arboles depende de los resultados de un vector aleatorio obtenido de manera independiente al resto [3].

Algunos investigadores [9] han utilizado este algoritmo junto con indicadores técnicos para la obtención de resultados binarios que indican si el precio del mercado debe aumentar o disminuir su valor, obteniendo para varios mercados siempre valores de exactitud de más del 84 %, llegando en algunos casos al 94 %, de esta manera predicen la dirección futura del movimiento de las acciones, confluendo a medida que se agregan más números de árboles aleatorios.

Otras aplicaciones [14] son la selección de las características a través de similitud difusa y la medida de entropía [10], se le aplicando al conjunto de datos indicadores técnicos, utilizando el algoritmo Random Forest para una clasificación cuaternaria obteniendo mejores resultados que con una binaria

Long Short-Term Memory. Una RNN de tipo Long Short-Term Memory, tiene una estructura que difiere de las RNN básicas, introduciendo la célula o bloque de memoria, una unidad de computación que reemplaza a las neuronas artificiales tradicionales en la capa oculta de la red, los bloques de memoria, que contienen compuertas de entrada y salida, y de una o más celdas de memoria, y utiliza como memoria a corto plazo (*short memory*) las conexiones de retroalimentación para almacenar las representaciones de las entradas anteriores. Se denomina Long Short-Term Memory debido a su capacidad de memoria a corto plazo, y a largo plazo dando diferentes pesos a cada uno y olvidando de la memoria lo que considera irrelevante para predecir el resultado [4], mejorando así el rendimiento ajustando los pesos y truncando el gradiente cuando la información no es necesaria. En la Fig. 2 se observa la capa sigmoidea, en la que se decide qué información es relevante y cual se excluye, se denomina comúnmente puerta del olvido. En la Fig. 3 se actualiza el estado de la

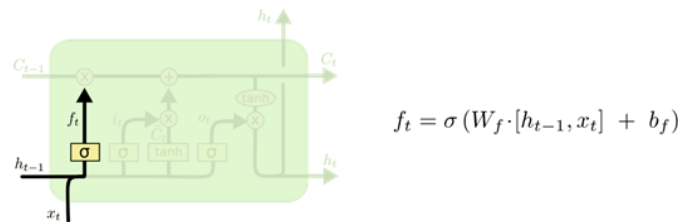


Figura 2. Capa sigmoidea de una celda LSTM

memoria, utilizando la entrada y el valor candidato, combinándose ambos.

En la Fig. 4 se obtienen los nuevos valores candidatos para la siguiente celda, multiplicando la parte que se olvida con la parte generada.

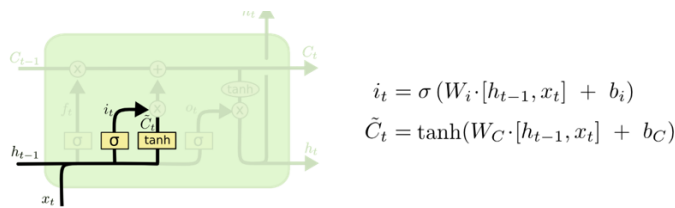


Figura 3. Combinación de la entrada y el valor candidato en una celda de LSTM

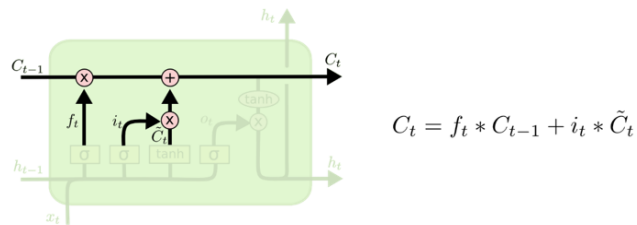


Figura 4. Nuevos valores candidatos en una celda de LSTM

En la Fig. 5 se obtiene el valor producido por la celda de la red LSTM, en la que se obtiene el valor filtrando el estado anterior y multiplicando por la puerta sigmoidea.

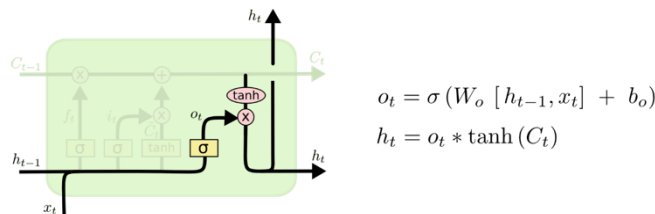


Figura 5. Salida de la celda LSTM

Se ha estudiado [17] la posibilidad de utilizar únicamente los indicadores técnicos, para ello como se ve en la Fig. 6 se han preprocesado los datos obtenidos de las velas japonesas para obtener los indicadores técnicos, estos han sido clasificados a través de una red LSTM, obteniendo como resultados al tratarse de un algoritmo de clasificación con LSTM la opción de comprar o bien no comprar.

Otros investigadores [20], hace uso de las redes LSTM para hacer un análisis de eventos que han sido publicadas en un pasado reciente de tal manera que el resultado obtenido será un valor binario, centrándose el estudio de con cuanta anticipación se puede prever un cambio en el mercado, para ello la capa de entrada tiene la misma dimensión que los datos de entrada, y siendo el análisis de datos utilizado un *framework* PESTEL (Fig. 6).

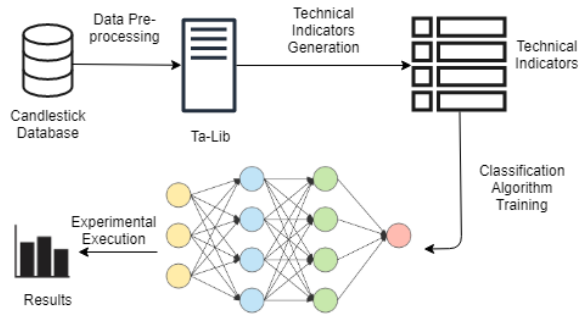


Figura 6. LSTM con indicadores técnicos

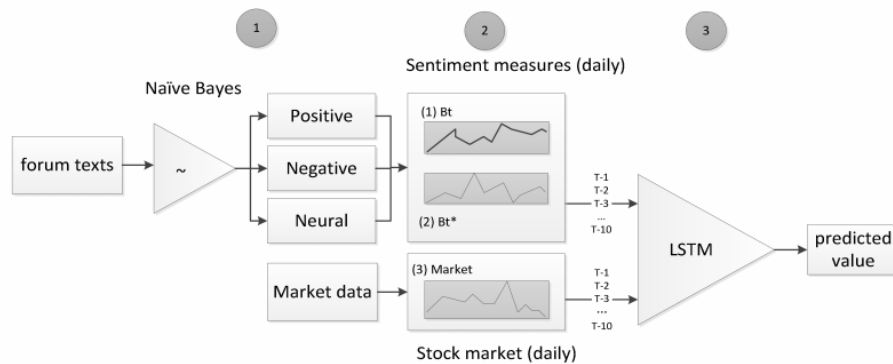


Figura 7. Metodología utilizada con análisis de sentimiento

Otras características que se han utilizado con series temporales es el análisis de sentimiento [12], el cual afecta a las decisiones de compras y ventas de activos en los mercados, de tal manera que los informes financieros que expresan un sentimiento negativo o positivo, afectan a los retornos de las acciones, y que según Antweiler y Frank [1] en las organizaciones, los mensajes publicados en foros de finanzas predecían rendimientos negativos según el sentimiento de los inversores, y por tanto estos influían en la volatilidad, volumen y precios de las operaciones del día siguiente. Se integra por tanto los datos de mercado como el sentimiento del supuesto inversor en un modelo de aprendizaje profundo prediciendo el valor futuro de las inversiones (Fig. 7).

Otros estudios [18] utilizan varias capas de redes neuronales LSTM junto con entradas de capas secuenciales, para ello utilizan las características básicas sin añadir indicadores técnicos. Para ello la arquitectura se basa en una entrada secuencial seguido de dos capas LSTM, y una capa densa con activación de la unidad lineal rectificadora (ReLU) [13].

3. Casos de Estudio

Como se ha estudiado en la sección anterior, existen diferentes tipos de análisis para la predicción del precio de los mercados tradicionales, se propone por tanto buscar diferentes soluciones para este estudio centrado en los mercados de criptomonedas, utilizando ambos enfoques existentes: el análisis técnico y en el análisis fundamental. En todos ellos se ha utilizado los datos obtenidos (Fig. 8) al consultar el *endpoint* de datos de velas japonesas de la casa de cambio Binance que incluye la hora de apertura, la hora de cierre, el precio de apertura, el precio de cierre, el precio más alto y más bajo, el volumen, el número de intercambios, el volumen del activo intercambiado, volumen de la divisa base y la divisa de cotización.

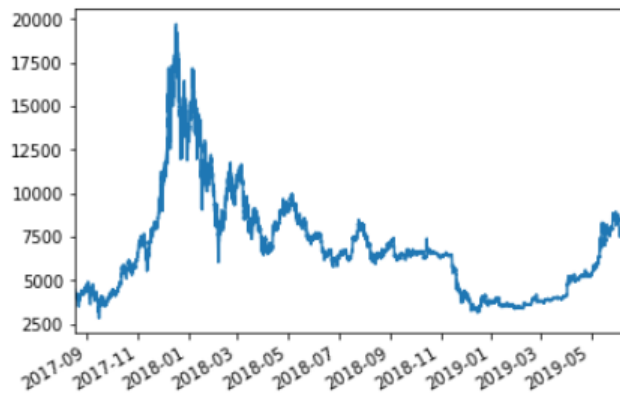


Figura 8. Precio de cierre de Bitcoin -USDT en Binance

3.1. Caso de Estudio 1

El primer caso de estudio, esquematizado en la Fig. 9 se utiliza Random Forest para clasificar y obtener valores binarios de compras o ventas. Para ello se realiza un suavizado exponencial como mecanismo de autocorrección suavizando algunos altibajos de grandes volúmenes de compra o venta, se calculan los indicadores técnicos: RSI,STOCH, ADL, ATR, MOM, MFI, ROC, OBV, CCI, EOM, TRIX, VI, MACD y EMA. A una muestra de este conjunto de datos ampliado se aplica el clasificador Random Forest.

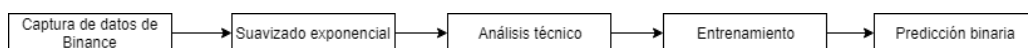


Figura 9. Esquema del primer caso de estudio

Solución. Los resultados obtenidos al dividir los datos sin mezclarlos se pueden observar que dan un rendimiento muy bajo en la evaluación, sobretodo en una tarea de clasificación.

| N= 153 | Invertir (Predicción) | No Invertir (Predicción) |
|---------------------------|------------------------------|---------------------------------|
| Invertir (Real) | 15 | 94 |
| No Invertir (Real) | 27 | 17 |

Tabla 1. Matriz de confusión de resultados

La matriz de confusión se puede observar en la Tabla 1, donde se refleja que la solución obtenida al dividir los datos sin mezclarlos, dan un rendimiento muy bajo en la evaluación, sobretodo para una tarea de clasificación (Tabla 2).

| | |
|----------------------|------|
| Precisión | 0.36 |
| Recuperación | 0.14 |
| Puntuación f1 | 0.20 |
| Exactitud | 0.21 |

Tabla 2. Evaluación de resultados

A partir de la matriz de confusión de la Tabla 3 se observa que si los datos de la división entre entrenamiento y prueba son aleatorios, se consiguen muy buenos resultados, obteniendo una exactitud del 86 % entre la observación correctamente predicha y las observaciones totales. La medida de precisión muestra que han existido un 90 % de las observaciones correctamente clasificadas entre el total de observaciones predichas (Tabla 4).

| N= 153 | Invertir (Predicción) | No Invertir (Predicción) |
|---------------------------|------------------------------|---------------------------------|
| Invertir (Real) | 69 | 14 |
| No Invertir (Real) | 8 | 62 |

Tabla 3. Matriz de confusión de resultados con datos mezclados

La matriz de confusión (Tabla 3) se interpreta que el algoritmo recomienda invertir o mantener en caso de poseer fondos, o vender en el caso de no invertir.

Además, gráficamente se puede observar como en la Fig. 10 y en la Fig. 11 como la predicción y la probabilidad de predicción respectivamente obtienen mejores resultados.

| | |
|----------------------|------|
| Precisión | 0.90 |
| Recuperación | 0.83 |
| Puntuación f1 | 0.86 |
| Exactitud | 0.86 |

Tabla 4. Evaluación de resultados con datos mezclados

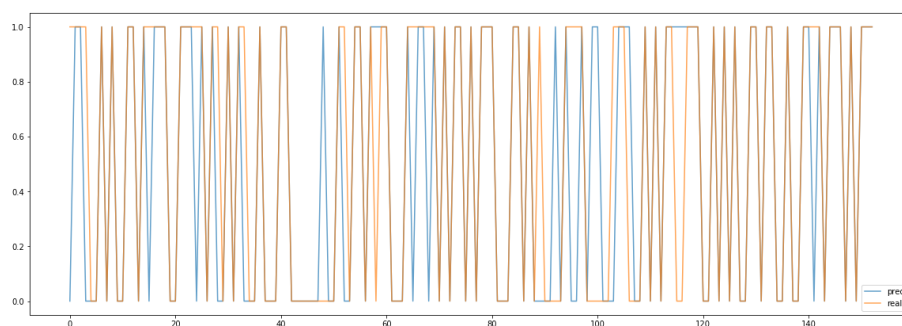


Figura 10. Predicción frente a realidad en el conjunto de datos de prueba - Datos mezclados

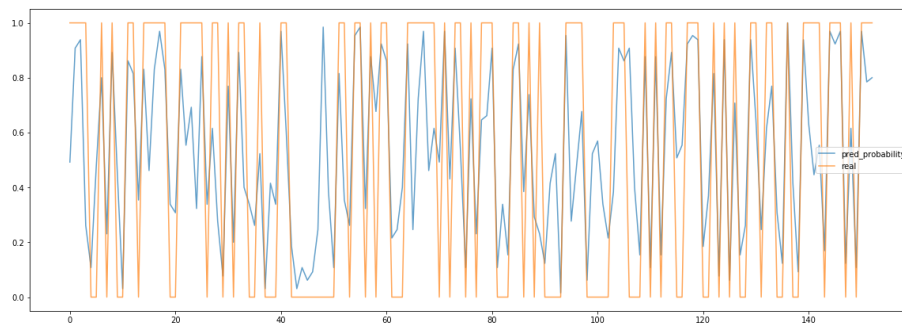


Figura 11. Probabilidad de predicción frente a realidad en el conjunto de datos de prueba - Datos mezclados

3.2. Caso de Estudio 2

El segundo caso de estudio utiliza el *framework* PESTEL, descrito anteriormente, centrándose en el factor social, para ello se va a tener en cuenta la red social de Twitter obteniendo los tuits sobre Bitcoin. Se va a seguir el esquema descrito en la Fig. 12

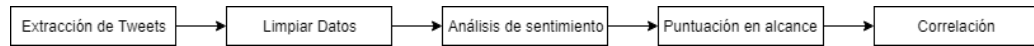


Figura 12. Esquema del segundo caso de estudio

Se ha calculado la puntuación para cada *tweet* utilizando análisis de sentimiento, obteniendo una puntuación para cada *tweet* entre -1 y 1 que indica si es negativo o positivo respectivamente.

Dado que no todos los *tweets* tienen el mismo alcance ni las mismas impresiones, la fórmula de la puntuación queda de la siguiente manera:

$$Puntuación = Puntuación \text{ análisis de sentimiento} * (N\acute{u}m \text{ de retuits} + 1) * (Followers \text{ de la cuenta} + 1) * (Likes \text{ del tuit} + 1)$$

Solución. Se hace un análisis de correlación entre las puntuaciones obtenidas de los *tweets* y la secuencia del par analizado, en este caso BTC/USDT. Para ello se utiliza el análisis de correlación de Pearson para cada columna de datos de tal manera que se pueda medir el grado de relación de dos variables cuantitativas y continuas. Se puede observar en la Fig. 13, que con *tweets* de mayor

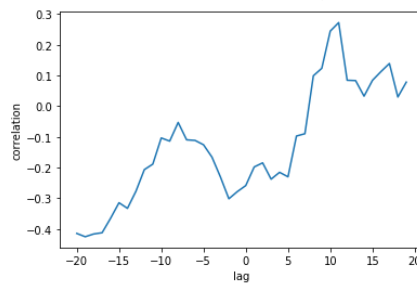


Figura 13. Coeficiente de Correlación de Pearson

puntuación existe una correlación positiva, y con los *tweets* de menor puntuación tenemos una correlación negativa respecto al precio del Bitcoin, por tanto las variables tienen algo de dependencia unas de otras.

3.3. Caso de Estudio 3

Se centra en la influencia de los precios de otras criptomonedas respecto a Bitcoin, se han utilizado los análisis técnicos, y las series temporales tomando únicamente los valores de tiempo en el que las dos monedas han tenido operaciones en el Bitcoin. Se ha seguido el esquema de la Fig. 14

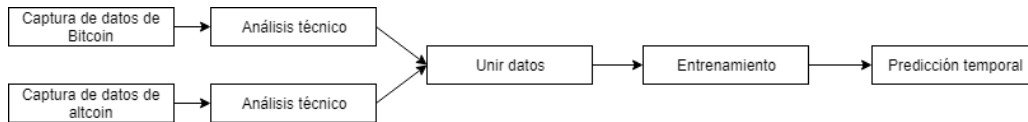


Figura 14. Esquema del tercer caso de estudio

Se ha obtenido los datos de Bitcoin y TRON (TRX) en rangos temporales de media hora, y se han añadido los siguientes indicadores técnicos a ambos conjuntos de datos: RSI, STOCH, ADL, ATR, MOM, MFI, ROC, OBV, CCI, EOM, TRIX, VI, MACD y EMA. Con estos conjuntos de datos se unen fijando el mismo rango de tiempo, de tal manera que ambos conjuntos tengan el mismo tamaño. Y se le aplica una arquitectura de LSTM apilados, que facilita la predicción de secuencias debido a que la profundidad de la red es más importante que el número de células de memoria [8].

Solución. Para ir disminuyendo el gradiente a través de las diferentes épocas que utiliza la red neuronal, para ello se utiliza el *mean squared error* (MSE) que mide la diferencia cuadrada media entre los valores medios y lo que se estima.

En la Fig. 15, se puede observar el resultado del precio actual del predicho utilizando los datos de prueba, donde se reflejan unos resultados muy robustos.

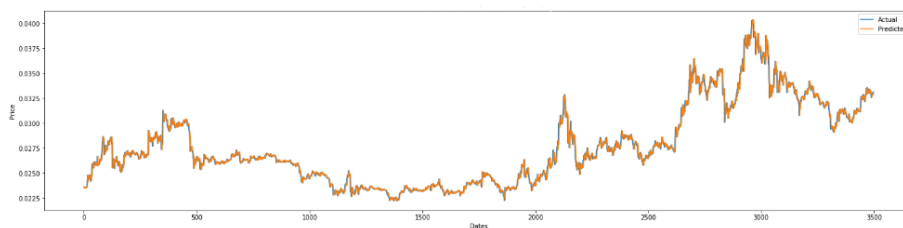


Figura 15. Secuencia de precios de TRX respecto al precio actual y el predicho

4. Conclusiones y Trabajo Futuro

En este trabajo se ha demostrado que en el mercado de criptomonedas tiene una gran importancia la utilización de indicadores propios del análisis técnico que aportan pautas de los movimientos

de los mercados, estos datos utilizados en un clasificador predictivo Random Forest han aportado un modelo robusto con solución binaria sobre la decisión sobre si se debe invertir o no invertir, indicando la dirección del mercado, con una precisión del 90 %. Además, se ha estudiado la influencia que tiene Bitcoin sobre las otras criptomonedas, y que es capaz de influir en el movimiento del resto, y que el uso de los datos de Bitcoin junto con una altcoin, y los indicadores técnicos de ambas puede dar unos resultados muy precisos sobre cómo va a variar los mercados de las altcoins. Se ha comprobado de una manera robusta que en ambos estudios que el análisis técnico está muy presente en los mercados de criptomonedas, y que seguramente los bots utilizados por los grandes *traders* utilizan estos analizadores para la obtención de beneficio y por tanto son los que mueven el mercado de criptomonedas, aun así se ha querido estudiar cómo afecta el análisis fundamental, el framework PESTEL y como los factores sociales y el sentimiento que la gente expresa en Twitter tiene una correlación con los precios de las criptomonedas.

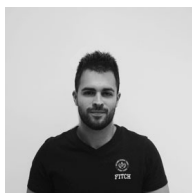
Como trabajo futuro se podrían estudiar todas las variables que puedan afectar al mercado indicadas por el framework PESTEL. Otro factor relevante que puede ayudar a la toma de decisiones es las *wallet* de las casas de cambio de las diferentes criptomonedas, y que se podría observar si existe un flujo entrante o saliente, y podría ayudar a tomar decisiones de entrada o salida al mercado. Además, un gran movimiento en una casa de cambio, afecta con un ligero retraso al resto, por tanto se podrían monitorizar todas las existentes de tal manera que se podría prever a pocos segundo el movimiento del mercado, ya que un movimiento en una afecta al resto.

Referencias

- [1] W. Antweiler y M. Z. Frank. «Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards». En: *The Journal of Finance* 59.3 (jun. de 2004), págs. 1259-1294. DOI: 10.1111/j.1540-6261.2004.00662.x.
- [2] L. Blume, D. Easley y M. O'hara. «Market statistics and technical analysis: The role of volume». En: *The Journal of Finance* 49.1 (1994), págs. 153-181.
- [3] L. Breiman. «Random Forests». En: *Machine Learning* 45.1 (2001), págs. 5-32. DOI: 10.1023/A:1010933404324.
- [4] K. Chen, Y. Zhou y F. Dai. «A LSTM-based method for stock returns prediction: A case study of China stock market». En: *2015 IEEE International Conference on Big Data (Big Data)*. IEEE. 2015, págs. 2823-2824.
- [5] S. Chun, Y. P. .-. E. S. with Applications y undefined 2005. «Dynamic adaptive ensemble case-based reasoning: application to stock market prediction». En: *Elsevier* ().
- [6] . on ... y undefined 2005. «Data preprocessing for stock market forecasting using random subspace classifier network». En: *ieeexplore.ieee.org* ().
- [7] R. D. Edwards, J. Magee y W. C. Bassetti. *Technical analysis of stock trends*. CRC press, 2018.
- [8] A. Graves, A. Mohamed, G. H. .-. 2. I. International y undefined 2013. «Speech recognition with deep recurrent neural networks». En: *ieeexplore.ieee.org* ().
- [9] L. Khaidem, S. Saha y S. R. Dey. «Predicting the direction of stock market prices using random forest». En: (abr. de 2016). arXiv: 1605.00003.
- [10] M. J. Kim, I. Han y K. C. Lee. «Hybrid knowledge integration using the fuzzy genetic algorithm: prediction of the Korea stock price index». En: *Intelligent Systems in Accounting, Finance & Management* 12.1 (ene. de 2004), págs. 43-60. DOI: 10.1002/isaf.240.

- [11] W. Leigh, R. Purvis, J. R. .-. . D. support Systems y undefined 2002. «Forecasting the NY-SE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision». En: *Elsevier* ().
- [12] J. Li, H. Bu y J. Wu. «Sentiment-aware stock market prediction: A deep learning method». En: *2017 International Conference on Service Systems and Service Management*. IEEE. 2017, págs. 1-6.
- [13] Y. Li, Y. Y. .-. . A. i. N. I. P. Systems y undefined 2017. «Convergence analysis of two-layer neural networks with relu activation». En: *papers.nips.cc* ().
- [14] C. Lohrmann y P. Luukka. «Classification of intraday S&P500 returns with a Random Forest». En: *International Journal of Forecasting* 35.1 (ene. de 2019), págs. 390-407. DOI: 10.1016/J.IJFORECAST.2018.08.004.
- [15] *Marketing Theories - PESTEL Analysis*. <https://www.professionalacademy.com/blogs-and-advice/marketing-theories---pestel-analysis>. Accedido: 08-Jul-2019.
- [16] S. Nakamoto *et al.* «Bitcoin: A peer-to-peer electronic cash system». En: (2008).
- [17] D. M. Nelson, A. C. Pereira y R. A. de Oliveira. «Stock market's price movement prediction with LSTM neural networks». En: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, págs. 1419-1426.
- [18] M. Roondiwala, H. Patel y S. Varma. «Predicting stock prices using LSTM». En: *International Journal of Science and Research (IJSR)* 6.4 (2017), págs. 1754-1756.
- [19] *The Bullish Case for Bitcoin - Vijay Boyapati - Medium*. <https://medium.com/@vijayboyapati/the-bullish-case-for-bitcoin-6ecc8bdecc1>. Accedido: 27-Jun-2019.
- [20] I. Verma, L. Dey, ... y undefined 2017. «Detecting, quantifying and accessing impact of news events on Indian stock indices». En: *dl.acm.org* ().
- [21] D. V. Villaverde y A. G. Briones. «Blockchain como revolución tecnológica del sector financiero: características y aspectos legales de las criptomonedas». En: *FODERTICS 7.0: estudios sobre derecho digital*. Comares. 2019, págs. 235-243.
- [22] I. Yüksel. «Developing a multi-criteria decision making model for PESTEL analysis». En: *International Journal of Business and Management* 7.24 (2012), pág. 52.

Autores



Pablo Chamoso es Profesor Asociado e Investigador en la Universidad de Salamanca. Es Doctor en Ingeniería Informática además de Ingeniero Técnico en Informática de Sistemas, Ingeniero en Informática (con premio i3 al mejor proyecto fin de carrera de Castilla y León), Máster en Comercio Electrónico y Máster en Sistemas Inteligentes.

Desde principios de 2011 es miembro del grupo de investigación BISITE, participando en el desarrollo técnico, análisis e investigación de múltiples proyectos internacionales y nacionales de áreas como Ciudades Inteligentes, Robótica, Análisis Visual, Aplicaciones Médicas e Inteligencia Artificial.

Juan M. Corchado es catedrático de Ciencia de la Computación e Inteligencia Artificial. Ha sido Vicerrector de Investigación y Transferencia de la Universidad de Salamanca y actualmente es Director del Grupo en Biotecnología, Sistemas Inteligentes y Tecnología Educativa (BISITE) de esta misma universidad. Doctor en Informática por la Universidad de Salamanca, España (1998) y doctor en Inteligencia Artificial por la University of the West of Scotland, Reino Unido (2000). Es Coordinador Académico del Instituto de Arte y Tecnología de la Animación. Anteriormente ha sido Subdirector de la Escuela Superior de Ingeniería Informática en la Universidad de Vigo (España, 1999-00) e investigador en la Universidad de Paisley (1995-98). Colabora, como investigador, con el Laboratorio Oceanográfico de Plymouth (Reino Unido) desde 1993. Ha dirigido más de 90 proyectos de Inteligencia Artificial con financiación tanto internacional, como nacional o autonómica. Ha dirigido más de 20 tesis doctorales y es co-autor de más de 800 libros, capítulos de libros, artículos en revistas científicas, etc. la mayoría de ellos presentan aspectos tanto prácticos como teóricos relacionados con los Sistemas Híbridos de Inteligencia Artificial, la Biomedicina, la Inteligencia Ambiental, los Sistemas Inalámbricos y la Oceanografía. Es autor de más de una veintena de propiedades intelectuales y patentes.



Ana B. Gil González es profesora Contratado Doctor en la Universidad de Salamanca en el área de Lenguajes y Sistemas Informáticos. Se Licenció en Ciencias Físicas por la Universidad de Salamanca. Es Doctora Europea en Informática y Automática con Premio Extraordinario de Doctorado por la Universidad de Salamanca. Ha sido investigadora en el LIP6 de la Universidad Pierre Marie Curie (París VI)-CNRS, realizando además estancias en distintos centros de Investigación (ESIEE de París, Universidad do Minho y Universidad de Aveiro). Sus principales áreas de interés en investigación están relacionadas con el desarrollo de marcos tecnológicos para la recuperación y personalización de contenido en diferentes ámbitos y la profundización en la representación y análisis de información, así como su aplicación en sistemas de recomendación. Es coautora de más de 80 publicaciones, incluidos artículos científicos, capítulos de libros y actas de congresos en diferentes áreas de investigación y editoriales. Ha participado en más de 25 proyectos competitivos de investigación de ámbito internacional, nacional y regional. Así mismo, ha participado en más de 10 contratos profesionales (Art. 83 LOU y afines), lo que ha permitido la transferencia conocimiento de la Universidad al tejido empresarial. Fruto de este trabajo se han re-

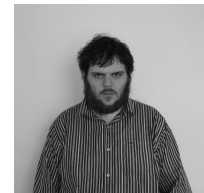
gistrado 10 propiedades intelectuales, 6 de ellas en explotación. Participa como comité científico en más de 20 congresos nacionales e internacionales así como en revistas científicas y en la organización de diferentes congresos.

Angélica González Arrieta es doctora en Informática por la Universidad de Salamanca. Cuenta con una amplia experiencia investigadora en el campo de sistemas conexionistas. Es Profesora Titular del Departamento de Informática y Automática de dicha Universidad, del área de Ciencia de la Computación e Inteligencia Artificial. Actualmente compatibiliza su labor docente e investigadora con la dirección de diversas actividades formativas sobre recursos tecnológicos, pericia y seguridad informática, colaborando activamente con las Fuerzas y Cuerpos de Seguridad.



Guillermo Hernández es doctor por el programa de Física Fundamental y Matemáticas de la Universidad de Salamanca, así como máster en Sistemas Inteligentes por la misma universidad. Actualmente trabaja en el grupo de investigación BISITE, también en la misma, donde desarrolla una investigación en análisis de datos en varios proyectos, incluyendo aspectos como el aprendizaje automático, el aprendizaje profundo, el procesamiento del lenguaje natural o la visualización de datos.

Francisco Lecumberri de Alba es maestro en Sistemas Inteligentes por parte de la Universidad de Salamanca (USAL) y es ingeniero en Cibernética y Sistemas Computacionales de la Universidad La Salle, México. Cuenta con experiencia en el desarrollo en robótica móvil y sistemas inteligentes en el laboratorio de robótica móvil y sistemas automatizados de la Universidad La Salle. Posteriormente formó parte del equipo de desarrollo para el robot comercial RoomieBot. Actualmente se encuentra en el grupo BISITE de la universidad de Salamanca como personal investigador.

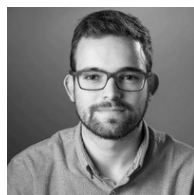


Vivian López Batista es profesora titular de la Universidad de Salamanca en el área de Ciencias de la Computación e Inteligencia Artificial. Doctorada en Informática por la Universidad de Valladolid en 1996. Miembro del Grupo de Minería de Datos. Ha realizado investigación en diferentes campos como procesamiento del lenguaje natural, redes neuronales y minería de datos. Tiene 80 artículos publicados en revistas de reconocido prestigio, talleres y actas de conferencias, 20 libros y capítulos de libros y 20 informes técnicos, la mayoría de ellos en estos temas. Miembro del comité organizador y científico de varios simposios internacionales. Fue directora del Máster en Sistema Inteligente y del Programa de Doctorado en Informática y Automática de la Universidad de Salamanca desde junio de 2010 hasta octubre de 2012.

María N. Moreno es Catedrática de Universidad del Departamento de Informática y Automática de la Universidad de Salamanca, directora del grupo de investigación en Minería de Datos y Coordinadora del Programa de Doctorado de Ingeniería Informática. Sus áreas de investigación de interés se centran en el



desarrollo y aplicación de algoritmos de minería de datos en diferentes dominios como minería web y de medios sociales o apoyo a las decisiones en medicina.



Alberto Moro Calvo estudió el Grado en Ingeniería Informática y el Máster en Sistemas Inteligentes en la Universidad de Salamanca. Es cofundador de BringLife S.L. y desarrollador *full stack* en CompuSoft Group.

Alejandra Ospina Bohórquez nació en Bogotá D.C., Colombia. Al finalizar sus estudios de bachillerato se mudó a España donde cursó el grado en Ingeniería Informática en Sistemas de Información en la Escuela Politécnica Superior de Zamora de la Universidad de Salamanca, en donde se le concedió el Premio Extraordinario de Fin de Carrera en reconocimiento a los méritos acreditados durante el grado. Una vez finalizado, cursó el Máster en Sistemas Inteligentes también en la Universidad de Salamanca, alcanzando unos resultados que la llevaron a obtener una de las becas concedidas por la Universidad de Salamanca y el Banco Santander a estudiantes latinoamericanos para realizar estudios de doctorado. Con ayuda de dicha beca, en la actualidad, se encuentra realizando el Doctorado en Ingeniería Informática de la USAL.



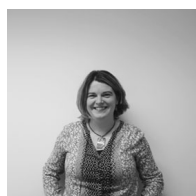
Javier Prieto Tejedor es Doctor en Tecnología de la Información y las Telecomunicaciones por la Universidad de Valladolid desde 2012 y Premio Extraordinario de Doctorado. En esa misma Universidad obtuvo sus títulos de Ingeniero de Telecomunicación (2008) y de Licenciado en Investigación y Técnicas de Mercado (2010). Desde 2007, Javier Prieto ha trabajado en diferentes centros públicos y privados de investigación, entre ellos la Universidad de Valladolid o el Massachusetts Institute of Technology. Actualmente, es miembro del grupo BISITE y

profesor asociado en la Universidad de Salamanca.

Ha publicado más de 50 artículos en revistas, libros y congresos internacionales, ha participado en más de 35 proyectos de investigación, y es autor de 2 patentes nacionales. Además, es editor de las revistas IEEE Communications Letters y Wireless Communications and Mobile Computing, editor invitado de la International Journal of Distributed Sensor Networks, miembro del comité científico de la Advances in Distributed Computing and Artificial Intelligence Journal, y revisor de numerosas revistas internacionales. También, es miembro del comité técnico de prestigiosos congresos internacionales (como IEEE GLOBECOM o IEEE ICC), y revisor en un número elevado de ellos. En el año 2013 recibió el premio “Jóvenes Emprendedores Castilla y León” del Instituto de la Juventud de Castilla y León.

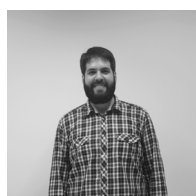
Sus intereses de investigación incluyen la computación social e inteligencia artificial, para crear ciudades más inteligentes y seguras, las tecnologías de localización y navegación, para guiar en entornos tanto de interior como de exterior, y las técnicas de inferencia Bayesiana, para la mejora del bienestar social y el desarrollo sostenible.

Luis Antonio Miguel Quintales es Doctor en Informática y Automática por la Universidad de Salamanca. Es profesor titular en el Departamento de Informática y Automática de la Universidad de Salamanca e investigador en el Instituto de Biología Funcional y Genómica (IBFG), centro mixto de investigación de titularidad compartida entre el Consejo Superior de Investigaciones Científicas (CSIC) y la Universidad de Salamanca. Las líneas de investigación en las que trabaja son la computación de altas prestaciones, la minería de datos y la genómica computacional. Como resultado de su trabajo de investigación ha publicado en algunas de las revistas más prestigiosas del ámbito de la Bioinformática como son Briefings in Bioinformatics, Bioinformatics y BMC Bioinformatics.



Sara Rodríguez es Profesora Contratada Doctor en el Departamento de Informática y Automática de la Universidad de Salamanca. Obtuvo su doctorado en Informática en la misma Universidad en 2010. Recibió el título de Ingeniería Técnica en Informática de Sistemas en 2004 y de Ingeniería en Informática en 2007. Por otra parte, ha llevado a cabo otros estudios de postgrado como el Máster en el desarrollo de sistemas de comercio electrónico y el Máster en animación de digital. Ha participado como co-autora en artículos publicados en revistas internacionales de reconocido prestigio. También ha participado como comité de programa en conferencias como PAAMS, IWANN, PACBB, ISAMI, HAIS o DCAI. En la actualidad, es miembro de la Asociación Española para la Inteligencia Artificial (AEPIA), del Instituto de Investigación de Arte y Tecnologías de la Animación de la Universidad de Salamanca y del grupo de investigación BISITE (Bioinformática, Sistemas Inteligentes y Tecnología) en el que sigue su labor investigadora.

Alejandro Rodríguez Díaz es un ingeniero informático graduado en la Universidad de Salamanca, donde también terminó sus estudios de máster sobre sistemas inteligentes. Actualmente forma parte del grupo de analítica visual VISUSAL (dentro del grupo de investigación reconocido GRIAL), donde trabaja en el diseño de nuevas técnicas de visualización para facilitar el entendimiento de conjuntos de datos complejos en el área de humanidades digitales. Otras áreas de su interés son la automatización de procesos informáticos y el diseño gráfico.



Diego Valdeolmillos Villaverde es Graduado en Ingeniería Informática por la Universidad de Salamanca y Técnico Superior en Desarrollo de Aplicaciones Informáticas, actualmente estudiante del Máster en Dirección de Sistemas de Información.

Su labor principal es la investigación de Big Data, Smart Cities y arquitecturas de computación distribuida.

Gabriel Villarubia González es Profesor Ayudante Doctor de la Universidad de Salamanca en el Departamento de Informática y Automática. Ingeniero Informático por la Universidad Pontificia de Salamanca (2011), habiendo estudiado el Máster en Sistemas Inteligentes en la Universidad de Salamanca (2012). Además ha realizado el Máster Universitario en Seguridad e Internet (2014) y el Máster en Dirección de Sistemas de Información (2015). A lo largo de su formación ha seguido una línea de investigación bien definida, centrada en la aplicación de los



sistemas multi-agente a los entornos de inteligencia ambiental, con especial atención a la definición de arquitecturas inteligentes y a la fusión de información. Ha participado en más de 30 proyectos de investigación, colaborando en la obtención de 15 propiedades intelectuales, habiendo realizado estancias internacionales. En el ámbito científico, es autor de más de 40 publicaciones científicas y codirector de 50 proyectos de fin de grado. Es miembro del comité científico de diferentes congresos, siendo miembro del comité organizador de (ISAMI, PACBB, MIS4TEL, PAAMS, DCAI). A nivel profesional fue nombrado Microsoft Student Partner en el año 2010. Ha ganado diferentes premios en el desarrollo de aplicativos móviles. Actualmente centra sus esfuerzos en diseñar una arquitectura multi-agente especialmente diseñada para la fusión dinámica de información en entornos inteligentes caracterizados por estar dotados de sensores heterogéneos que capturan información del entorno.
