



UNIVERSIDAD
DE GRANADA

Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación

PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE LA
INFORMACIÓN Y LA COMUNICACIÓN

Métodos y herramientas para Ciencia de Datos aplicada: control eficiente de edificios y análisis de datos médicos

Roberto Morcillo Jiménez
Dra. María Amparo Vila Miranda
Dr. Juan Gómez Romero



Métodos y herramientas para Ciencia de Datos aplicada: control eficiente de edificios y análisis de datos médicos

Roberto Morcillo Jiménez

Editor: Universidad de Granada. Tesis Doctorales
Autor: Roberto Morcillo Jiménez
ISBN: 978-84-1195-175-3
URI: <https://hdl.handle.net/10481/89442>

Autor: Roberto Morcillo Jiménez

Directores: Dra. María Amparo Vila Miranda
Dr. Juan Gómez Romero

Programa: Programa de Doctorado en Tecnologías de la Información y la
Comunicación
Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación
Universidad de Granada

DECLARACIÓN DE ORIGINALIDAD

D. Roberto Morcillo Jiménez

El siguiente documento, titulado Métodos y herramientas para Ciencia de Datos aplicada: control eficiente de edificios y análisis de datos médicos, ha sido redactado por Roberto Morcillo Jiménez, como requisito para la obtención del título de Doctor en Ciencias de la Computación e Inteligencia Artificial por el Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada. El trabajo aquí presentado es original y ha sido realizado bajo la supervisión de Dra María Amparo Vila Miranda y Dr Juan Gómez Romero . Todas las fuentes utilizadas han sido debidamente citadas y se han respetado los derechos de autor correspondientes.

En Granada a 30 de octubre de 2023

Fdo: Roberto Morcillo Jiménez

Dedicado a toda mi familia, en especial, a mi hermano, sobrinos, cuñados y suegros por que siempre os llevo en mente y siempre sé de donde provengo. A mi mujer y a mis hijos Martín y Enrique por soportar mis altos y bajos en esta lucha incansable, para que se sientan orgullosos de mi y que sepan que todo esto lo he hecho por ellos. Y finalmente a dos personas que soy lo que soy gracias a ellos, MAMÁ, sin ti no hubiese conseguido nada, tú eres la responsable de mi éxito, tú eres la que ha conseguido esto, siempre te lo tendré agradecido. Y a ti PAPÁ, que allá donde quieras que estés espero que te sientas orgulloso de mi. Esta Tesis es para todos vosotros.

Índice general

Agradecimientos	VII
Resumen	IX
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	2
1.3 Organización Tesis Doctoral	2
2 Preliminares	5
2.1 Ciencia de Datos	5
2.1.1 Conceptos	5
2.1.2 Aplicaciones	6
2.1.3 Metodología	6
2.1.4 Técnicas y aplicaciones	7
2.2 Plataforma de análisis de datos	10
2.2.1 Concepto	10
2.2.2 Técnicas y aplicaciones	12
3 Resumen y Resultados	15
3.1 TSxtend: Una herramienta integral para el preprocesamiento y análisis de datos temporales de sensores en investigación.	15
3.2 Análisis de series temporales y predicción de consumo de energía en edificios utilizando técnicas de aprendizaje profundo	16
3.3 Extracción de conocimiento oculto en registros médicos mediante minería de datos y lógica difusa en un entorno distribuido.	18
3.4 AIMDP: Plataforma de datos basada en Big Data y IA para la gestión eficiente y análisis de información en entornos heterogéneos.	19
4 Publicaciones	21
4.1 TSxtend: Una herramienta integral para el preprocesamiento y análisis de datos temporales de sensores en investigación.	21
4.2 Análisis de series temporales y predicción de consumo de energía en edificios utilizando técnicas de aprendizaje profundo	52
4.3 Extracción de conocimiento oculto en registros médicos mediante minería de datos y lógica difusa en un entorno distribuido.	74
4.4 AIMDP: Plataforma de datos basada en Big Data y IA para la gestión eficiente y análisis de información en entornos heterogéneos.	89
5 Conclusiones y Trabajos Futuros	109
Bibliografía	111

Agradecimientos

Siempre he tenido muy presente el dicho que dice, “Es de buen nacido ser agradecido”, y después del largo camino que he recorrido no puedo dejar escapar la oportunidad de agradecer a todas aquellas personas que han sido participes en mayor y en menor medida de la realización de mi trabajo. En primer lugar agradecer a mis compañeros de grupo Karel, Carlos, Mariló, Andrea, Jose Ángel, Bart, Juan y Salva por esa ayuda que nos hemos prestado para conformar este gran grupo. Mención a parte para María José Martín Bautista por haber confiado en mi y ser miembro de su grupo de investigación pudiendo tener continuidad en el desarrollo de mi trabajo, gracias. También agradecer a Rafael Perez Gómez, María Dolores Martínez Aries, Antonio Ruiz Moya y Javier Medina Quero, ya que fueron mis primeros compañeros y los que me animaron a desarrollar el trabajo que hoy día he conseguido finalizar. Finalmente mencionar a tres personas que son el núcleo de mi trabajo. En primer lugar, Juan Gómez director de mi tesis, gracias por tu tiempo y dedicación, siempre lo tendré presente. Finalmente, aunque sé que no les gusta, agradecer a las dos personas que me dieron la oportunidad de poder entrar en la Universidad, a mis dos profesores y amigos, Amparo Vila y Miguel Delgado, gracias por haberme dado la oportunidad de trabajar con vosotros y de haber conocido a esta maravillosa familia. A todos los que os he nombrado y alguno más que se habrá escapado, muchas gracias de corazón y espero que sintáis que una parte de este trabajo también es vuestro.

Resumen

El creciente incremento en las capacidades computacionales para generación y almacenamiento de grandes volúmenes de datos en diversos campos científicos ha posibilitado el desarrollo de investigaciones novedosas basadas en el análisis de dichos datos. Para facilitar el trabajo de los investigadores, expertos en su campo pero no en Ciencia de Datos, se ha identificado la necesidad de desarrollar herramientas que permitan utilizar algoritmos avanzados sin requerir un conocimiento técnico elevado. En esta tesis doctoral se presenta una nueva plataforma para realizar tareas de Ciencia de Datos con fuentes heterogéneas, así como dos casos de uso en dominios de aplicación especializados: la eficiencia energética y la medicina. Se trata de una tesis por compendio, en la que se recogen tres artículos indexados con resultados de esta investigación y uno en estado de revisión por parte de la revista: 1) una biblioteca para análisis de series temporales y su utilización a predicción de demanda energética; 2) aplicación de la propia biblioteca en un caso de uso real; 3) un algoritmo para extracción de patrones en conjuntos de datos distribuidos, aplicado a información médica; 4) una plataforma para almacenamiento y procesamiento de datos masivos que integra diversas herramientas, incluidas las anteriores, destinada a especialistas en medicina.

1 Introducción

1.1. Motivación

En las últimas dos décadas se han producido grandes avances en las tecnologías de generación y almacenamiento masivo de datos, lo cual ha permitido realizar investigaciones más profundas en diversos dominios científicos a partir del análisis de estos datos. No obstante, para procesar e interpretar estos datos masivos, en general se requiere un alto conocimiento de técnicas computacionales, lo cual dificulta el trabajo de las personas expertas en su campo pero no en Ciencia de Datos. Si bien la confección de equipos interdisciplinarios y la existencia de herramientas que no requieren programación pueden aliviar esta problemática, en muchas ocasiones estos recursos no son suficientes o no están al alcance de los investigadores. Así, ocurre que muchas veces los datos se almacenan en *silos* y no son explotados adecuadamente.

El término silo de datos se refiere a un conjunto de datos, normalmente de un determinado dominio, que se encuentran almacenados de manera aislada y que no son fácilmente accesibles ni compartidos entre diferentes departamentos o sistemas dentro de una organización. Estos silos pueden existir por diversas razones: la diferencia de formatos es considerable, no existe un software capaz de integrarlos, los equipos no son capaces de combinar esa información e, incluso, la propia legislación no permite el procesamiento de ciertos subconjuntos de datos fuera de donde se encuentran almacenados.

En el ámbito del *soft computing* se han propuesto diversas plataformas para dotar a los investigadores de herramientas sencillas, amigables e intuitivas (en inglés, *user friendly*), que facilitan la utilización de algoritmos de procesamiento de datos masivos. Normalmente, estas herramientas están alineadas con una determinada metodología, donde se establecen los pasos a seguir para completar un proceso de análisis dentro de un marco conceptual coherente. Una característica deseable en estas plataformas es que puedan gestionar datos heterogéneos y, además, poco estructurados. Sin embargo, los estudios bibliográficos realizados han evidenciado que en la actualidad no existen plataformas que satisfagan las necesidades de flexibilidad, facilidad de uso y capacidad de procesamiento avanzado que se requieren en dominios especializados.

Esta tesis doctoral describe el diseño, implementación y aplicación de una plataforma de análisis de datos para expertos que no requiere de conocimientos en Ciencia de Datos. La plataforma se fundamenta en una arquitectura que abstrae la complejidad de la representación, integración y procesamiento de los datos en diversas capas de procesamiento. Asimismo, incluye diversos algoritmos para solventar varias tareas de Ciencia de Datos habituales, como son la búsqueda de asociaciones y el análisis de series temporales. Para ilustrar las capacidades de la plataforma y de sus principales componentes, se han abordado dos casos de uso, relacionados con los dos proyectos en los que se enmarca el trabajo de la tesis: predicción del consumo de energía en edificios (proyecto PROFICIENT) y clasificación para diagnóstico de enfermedades y comorbilidad (proyecto BIGDATAMED).

La tesis doctoral consiste en el agrupamiento de los cuatro trabajos de investigación publicados en revistas científicas donde se describen estas propuestas:

- **R. Morcillo-Jimenez**, K. Gutiérrez-Batista and J. Gómez-Romero, «Tsxtend: A tool for batch analysis of temporal sensor data», *Energies*, vol. 16, no. 4, 2023. <https://doi.org/10.3390/en16041581>.
- **R. Morcillo-Jimenez**, J. Mesa, J. Gómez-Romero, M.-A. Vila and M.J. Martin-Bautista, «Deep learning for prediction of energy consumption: an applied use case in an office building». (Bajo revisión en revista *Applied Intelligence*).
- C. Fernandez-Basso, K. Gutiérrez-Batista, **R. Morcillo-Jiménez**, M.-A. Vila and M.J. Martin-Bautista, «A fuzzy-based medical system for pattern mining in a distributed environment: Application to diagnostic and co-morbidity», *Applied Soft Computing*, vol. 122, p. 108 870, 2022. <https://doi.org/10.1016/j.asoc.2022.108870>.
- A. S. Ortega-Calvo, **R. Morcillo-Jimenez**, C. Fernandez-Basso, K. Gutiérrez-Batista, M.-A. Vila and M.J. Martin-Bautista, «Aimdp: An artificial intelligence modern data platform. use case for Spanish national health service data silo», *Future Generation Computer Systems*, vol. 143, pp. 248–264, 2023. <https://doi.org/10.1016/j.future.2023.02.002>.

1.2. Objetivos

De manera consecuente a la motivación anterior, el objetivo principal de esta tesis doctoral es el desarrollo de una plataforma software que permita a usuarios expertos en un dominio pero sin conocimientos en Ciencia de Datos realizar tareas avanzadas de análisis de datos. La plataforma permite aplicar, de manera sencilla y consistente, las etapas habituales de la metodología de Ciencia de Datos: exploración, preprocesamiento, análisis y validación.

Los objetivos específicos de la tesis son los siguientes:

1. Identificar las tareas habituales para la resolución de problemas en Ciencia de Datos, conociendo su papel, temporización y dependencias en el proceso de análisis de datos.
2. Caracterizar las funcionalidades que debe cubrir una plataforma de datos que dé soporte a las tareas anteriores, considerando diversos dominios, tipos de datos y necesidades de los usuarios.
3. Crear estructuras para la representación y el almacenamiento de datos heterogéneos dentro de la nueva plataforma.
4. Diseñar e implementar la plataforma de datos.
5. Ilustrar las capacidades de la plataforma en varios dominios de aplicación; energía y medicina, en este caso.

1.3. Organización Tesis Doctoral

Esta memoria de tesis está compuesta por cinco capítulos, en cumplimiento de la normativa de las tesis por compendio de publicaciones de la Universidad de Granada.

Capítulo 2 Preliminares. En este capítulo se presentan los principales conceptos que se utilizan en la tesis, así como algunos aspectos generales de la aproximación al problema realizada en relación con las soluciones existentes en la literatura.

Capítulo 3 Resumen y resultados. En este capítulo se resumen las publicaciones que apoyan las tesis y los principales resultados obtenidos durante la investigación.

Capítulo 4 Publicaciones. Este capítulo recoge las cuatro publicaciones realizadas en revistas de alto impacto.

Capítulo 5 Conclusiones y trabajo futuro. En este último capítulo se resumen las principales conclusiones de la investigación y se proponen varias líneas de investigación futuras.

2 Preliminares

En esta sección se recogen los principales conceptos utilizados en esta tesis doctoral. En primer lugar, en la Sección 2.1, se describirá el área de la Ciencia de Datos, así como las diferentes etapas que abarca una metodología en este ámbito. A continuación, se presentarán las diferentes técnicas que se han usado en esta tesis doctoral (en particular, las series temporales y las reglas de asociación) y cómo se han desarrollado una serie de herramientas (TSxtend [22] y algoritmos de reglas de asociación difusos [7]). En la Sección 2.2 se expondrá el concepto de plataforma de análisis de datos, una herramienta software para gestionar y aprovechar de manera eficiente los datos en diferentes dominios de aplicación. Se comentará la importancia de que este tipo de plataformas faciliten el trabajo de usuarios sin conocimientos de programación y la manera de estructurar la información de forma que pueda ser explotada de manera eficiente y segura. [24]

2.1. Ciencia de Datos

2.1.1. Conceptos

La Ciencia de Datos[13] es un campo interdisciplinario que se ocupa de extraer conocimientos y patrones significativos a partir de conjuntos de datos. Es por tanto un ámbito del conocimiento relacionado con la inteligencia artificial, entendida como el desarrollo de sistemas capaces de realizar tareas que requieren inteligencia humana, como el razonamiento, la percepción, el reconocimiento de voz y la toma de decisiones.

Una rama importante en la Ciencia de Datos es el **aprendizaje automático**[9]. Se enfoca en el desarrollo de algoritmos y modelos que permite aprender y mejorar automáticamente a partir de datos. Incluye una serie de técnicas entre las que destacan la clasificación, regresión, agrupamiento y procesamiento del lenguaje natural.

La **minería de datos**[29] es el proceso de descubrir patrones y relaciones interesantes en grandes conjuntos de datos. Utiliza técnicas estadísticas, de aprendizaje automático y de visualización para identificar información valiosa y conocimientos ocultos.

La representación gráfica de datos y patrones para facilitar la comprensión y comunicación de la información se denomina **visualización de datos**[30]. Las visualizaciones pueden incluir gráficos, mapas, diagramas y otras representaciones visuales. Es importante para que el usuario pueda entender los resultados de la ejecución de los diferentes algoritmos.

El **Big Data**[19], por su parte, se refiere a soluciones que trabajan con conjuntos de datos extremadamente grandes y complejos que exceden las capacidades de las herramientas de procesamiento de datos tradicionales y, por tanto, requieren tecnologías específicas para extraer información útil.

Un concepto que está recibiendo mucha atención en la actualidad es la **privacidad y la ética**[6] en el manejo y explotación de los datos. Debido a que la Ciencia de Datos utiliza grandes cantidades de datos, que pueden ser personales y confidenciales, es importante considerar la privacidad y la ética en el manejo de la información. Esto implica garantizar la protección de la privacidad de los individuos y tomar decisiones éticas al utilizar los datos

para evitar sesgos y discriminación. Estos aspectos se tienen en cuenta en este trabajo debido al manejo de datos sensibles como son los datos de historias clínicas.

2.1.2. Aplicaciones

La Ciencia de Datos se aplica en una amplia gama de ámbitos y sectores, incluyendo:

- **Empresas y comercio:** Se utiliza para analizar datos de ventas, marketing, operaciones y finanzas, con el objetivo de optimizar procesos, identificar oportunidades de crecimiento, mejorar la eficiencia y comprender mejor el comportamiento del cliente.
- **Ciencias sociales y comportamientos:** Ayuda a comprender y predecir patrones de comportamiento humano en áreas como la demografía, la psicología, la economía y la sociología. Puede utilizarse para estudiar el comportamiento del consumidor, la opinión pública y el análisis de redes sociales, entre otros.
- **Ciencias de la salud y medicina:** Contribuye a la investigación biomédica, el análisis de datos clínicos, el descubrimiento de medicamentos, el diagnóstico médico asistido por computadora, la genómica y la epidemiología. La ciencia de datos en salud también puede ayudar en la detección de brotes de enfermedades y en la implementación de intervenciones de salud pública.
- **Consumo y eficiencia energética:** Es fundamental para entender cómo se utiliza la energía en diferentes espacios y optimizar su consumo. Mediante el monitoreo en tiempo real y el análisis histórico de datos, se pueden identificar patrones de consumo y oportunidades de mejora. Además, el uso de algoritmos de aprendizaje automático permite predecir la demanda energética futura y optimizar sistemas de control para un uso más eficiente de la energía.
- **Gobierno y administración pública:** Permite analizar datos gubernamentales para mejorar la eficiencia en la prestación de servicios públicos, identificar patrones delictivos, realizar análisis de políticas y tomar decisiones informadas.
- **Investigación científica:** Se aplica en diversos campos científicos para analizar datos experimentales, realizar simulaciones, modelar fenómenos complejos y descubrir nuevos conocimientos. La ciencia de datos es particularmente útil en la astronomía, la física de partículas, la climatología y la biología.
- **Tecnología y empresas de internet:** Empresas como Google, Facebook y Amazon utilizan la ciencia de datos para mejorar la experiencia del usuario, personalizar recomendaciones, optimizar motores de búsqueda, detectar fraudes y realizar análisis de big data.

2.1.3. Metodología

En líneas generales, una metodología es un conjunto estructurado de pasos y procedimientos que se siguen para abordar un problema o tarea de manera eficiente y efectiva. Una metodología proporciona un marco de trabajo para planificar y realizar una investigación, lo que en análisis de datos incluye la selección de la muestra, la recolección y estudio de los datos, la interpretación de resultados y la presentación de conclusiones.

Existen diferentes metodologías para llevar a cabo un proyecto de Ciencia de Datos pero, en general, se pueden identificar las siguientes etapas comunes:

- Definición del problema, donde se identifica el problema o la pregunta de investigación que se desea abordar. Se define el alcance del proyecto, se identifican las fuentes de datos relevantes y se establecen los objetivos del proyecto.
- La recopilación de datos, que es la etapa en la que se almacenan y preparan los datos necesarios para su análisis. Esto incluye la identificación de las fuentes de datos, la selección de los datos relevantes y la limpieza y transformación de los datos para su posterior análisis.
- El análisis exploratorio de datos, que se aplica para comprender mejor su estructura, distribución y relaciones entre variables. Para ello, se utilizan técnicas de visualización de datos y estadísticas descriptivas.
- El modelado, que es la etapa donde se seleccionan y aplican algoritmos y técnicas de inteligencia artificial para analizar los datos y generar modelos matemáticos o computacionales que puedan utilizarse para hacer predicciones o tomar decisiones.
- La evaluación y validación, etapa en la que se evalúa el rendimiento del modelo y se valida su capacidad para hacer predicciones precisas y confiables en nuevos conjuntos de datos.
- La comunicación de resultados, cuando se presentan los resultados del análisis en un informe final que incluye las conclusiones, las recomendaciones y las implicaciones de los resultados, pudiéndose presentar los resultados de una manera visual para facilitar su comprensión.

2.1.4. Técnicas y aplicaciones

En el ámbito de la Ciencia de Datos se han propuesto gran cantidad de técnicas, resultando la mayoría de ellas complicadas para usuarios sin conocimientos avanzados en programación. Por tanto, en esta tesis doctoral se han creado un conjunto de herramientas que permiten el uso de técnicas avanzadas de análisis de datos sin requerir conocimientos técnicos de computación, y que se describen a continuación.

2.1.4.1. Análisis de series temporales y estimación de demanda energética

Las series temporales son un conjuntos de datos secuenciales que se recopilan y registran en intervalos de tiempo específicos. Pueden abarcar diversas áreas, desde finanzas y economía hasta meteorología, medicina y consumo energético. Estas series suelen tener una dependencia temporal, lo que significa que los valores anteriores influyen en los futuros. La comprensión y el análisis de las series temporales son fundamentales en la predicción de patrones, la detección de anomalías y la toma de decisiones basada en datos.

La visualización de series temporales es una herramienta esencial para comprender la evolución de los datos a lo largo del tiempo. A menudo se utilizan gráficos de líneas o gráficos de dispersión para representar estas series, donde el eje horizontal muestra el tiempo y el eje vertical representa el valor correspondiente. Al observar estos gráficos, es posible

identificar tendencias, estacionalidad, patrones cíclicos o fluctuaciones aleatorias en los datos, lo que ayuda a comprender mejor el comportamiento de la serie.

El análisis de series temporales implica técnicas estadísticas y matemáticas para modelar y predecir los datos futuros. Un enfoque común es utilizar modelos autoregresivos, como el modelo ARIMA (AutoRegressive Integrated Moving Average, Media Móvil Integrada Autoregresiva), que tiene en cuenta tanto las dependencias temporales como los componentes estacionales de la serie. Estos modelos permiten hacer predicciones a corto plazo y pueden ser útiles en la planificación y la toma de decisiones estratégicas. En la actualidad, las técnicas prevalentes son las basadas en redes neuronales.

Estas técnicas basadas en redes neuronales, como las redes neuronales recurrentes (RNN) y las redes neuronales de memoria a corto y largo plazo (LSTM), han demostrado ser especialmente efectivas para el análisis de series temporales debido a su capacidad para capturar patrones temporales complejos y hacer predicciones a largo plazo con precisión. Esto las hace relevantes en una variedad de aplicaciones, desde la predicción del clima y el análisis de datos financieros hasta la gestión de inventarios y la monitorización de la salud. La combinación de modelos autoregresivos tradicionales y técnicas basadas en redes neuronales proporciona un conjunto completo de herramientas para abordar una amplia gama de desafíos en el análisis de series temporales en la actualidad, lo que permite a las organizaciones tomar decisiones informadas y estratégicas en un mundo cada vez más impulsado por los datos.

En el ámbito de la eficiencia energética, el análisis de series temporales desempeña un papel crucial para comprender los patrones de consumo energético y encontrar oportunidades de mejora. Las series temporales permiten recopilar y analizar datos históricos que ayuda a revelar información valiosa sobre los patrones de consumo de energía.

Una parte del trabajo realizado en nuestra tesis doctoral se centró en el análisis de series temporales en el ámbito de la eficiencia energética. A través de la recopilación y análisis de datos históricos de consumo de energía, examinamos detenidamente los patrones y tendencias en el consumo energético de edificios de oficinas. Utilizando modelos de series temporales como ARIMA y técnicas de aprendizaje automático, desarrollamos procedimiento para predecir el consumo futuro de energía y detectar anomalías en el rendimiento energético.

Para ello, desarrollamos una biblioteca llamada TSxtend [23] compuesta por una serie de módulos con diversos algoritmos destinados a tareas específicas (Figura 2.1). En concreto, TSxtend se organiza en cinco módulos: almacenamiento de datos, configuración, procesamiento de datos, lanzamiento de algoritmos de aprendizaje automático y lanzamiento de algoritmos de aprendizaje profundo.

2.1.4.2. Reglas de asociación y lógica difusa para análisis de historiales clínicos

Las reglas de asociación [1] son una técnica comúnmente utilizada en la ciencia de datos para descubrir patrones y relaciones entre elementos de conjuntos de datos. Las reglas de asociación son una técnica comúnmente utilizada en la ciencia de datos para descubrir patrones y relaciones ocultas entre elementos dentro de conjuntos de datos. Estas reglas son especialmente útiles cuando se trabaja con grandes volúmenes de información, ya que pueden revelar conexiones significativas que podrían pasar desapercibidas mediante un análisis manual o tradicional. El concepto principal detrás de las reglas de asociación es la idea de que ciertos elementos o características tienden a aparecer juntos con una frecuencia mayor de la que se esperaría al azar.

El descubrimiento de reglas de asociación abrió nuevas perspectivas para la toma de

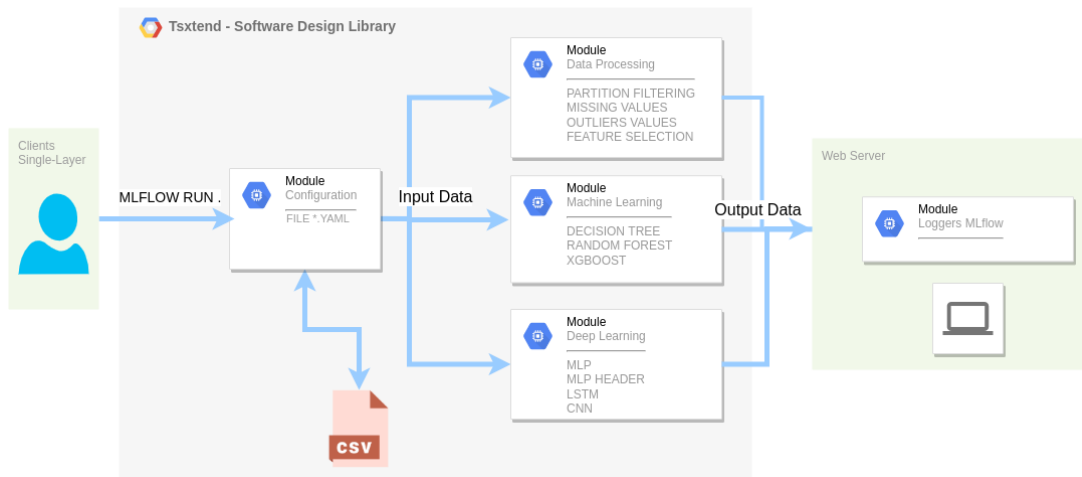


Figura 2.1: Diseño de TSxtend

decisiones basadas en datos, la segmentación de mercado, la optimización de procesos y la identificación de tendencias emergentes, lo que la convirtió en una herramienta esencial en el análisis de datos y la investigación.

En esencia, las reglas de asociación buscan responder preguntas como: "Si un cliente compra un cierto producto, ¿qué otros productos es más probable que compre también?", o "¿Qué síntomas suelen estar relacionados en pacientes con ciertas enfermedades?". Estas respuestas pueden tener aplicaciones en una amplia variedad de campos, desde la recomendación de productos en línea hasta el diagnóstico médico[8] y la toma de decisiones en la gestión empresarial.

En el contexto de la salud[11], las reglas de asociación pueden ser aplicadas para analizar datos clínicos, epidemiológicos y de investigación biomédica con el fin de obtener información relevante. A continuación, se presentan algunos ejemplos de cómo se pueden aplicar las reglas de asociación en el dominio de la salud:

- **Análisis de síntomas y diagnóstico:** Se pueden analizar datos clínicos de pacientes para descubrir patrones de síntomas que estén asociados con ciertas enfermedades. Por ejemplo, se podría descubrir que la presencia de fiebre alta, dolor de garganta y tos persistente se asocia fuertemente con la gripe. Esto puede ayudar en el diagnóstico temprano y la toma de decisiones clínicas.
- **Predicción de enfermedades:** Utilizando datos históricos de pacientes, como su historial médico, exámenes de laboratorio y características demográficas, se pueden descubrir reglas de asociación que permitan predecir la aparición de ciertas enfermedades en individuos similares. Esto puede ser útil para identificar grupos de riesgo y tomar medidas preventivas.
- **Medicina personalizada:** Al analizar datos genómicos, datos de tratamiento y respuestas de pacientes, se pueden descubrir asociaciones entre variantes genéticas y la respuesta a ciertos medicamentos. Esto puede permitir la personalización de tratamientos farmacológicos y minimizar los efectos secundarios.

- **Análisis de efectividad de tratamientos:** Al analizar datos de ensayos clínicos y registros médicos, se pueden descubrir reglas de asociación entre tratamientos específicos y resultados de salud. Esto puede ayudar a identificar qué tratamientos son más efectivos para diferentes grupos de pacientes y optimizar la elección de terapias.
- **Detección de factores de riesgo:** Al analizar datos epidemiológicos, como registros de salud pública y encuestas, se pueden descubrir reglas de asociación entre ciertos factores de riesgo y la aparición de enfermedades. Por ejemplo, se podría encontrar una asociación entre el consumo de tabaco y el cáncer de pulmón. Estas asociaciones pueden ayudar a diseñar estrategias de prevención y promover la salud pública.

Es importante destacar que la aplicación de reglas de asociación en el dominio de la salud requiere un manejo cuidadoso de los datos y consideraciones éticas y de privacidad para garantizar la confidencialidad de la información del paciente. Además, los resultados de las reglas de asociación deben ser interpretados con cuidado y validados en estudios adicionales antes de tomar decisiones clínicas o de salud pública.

Otra cuestión importante es cómo podemos mejorar nuestro conocimiento de forma que beneficie a nuestros algoritmos y mejore sus resultados en los experimentos. En este sentido, la lógica difusa desempeña un papel crucial al generar nuevos registros que son más fáciles de interpretar para diversos algoritmos, lo que resulta fundamental para enriquecer nuestros datos. Esto, a su vez, conduce a un enriquecimiento adicional del conocimiento almacenado en nuestro sistema a través de la incorporación de estos nuevos registros.

En el dominio de la medicina ha resultado ser muy útil sobre todo para interpretar estados de pacientes y poder conseguir clasificarlos en diferentes variables dependiendo del grado de pertenencia dictado por los algoritmos difusos.

Existen trabajos altamente escalables para aplicar algoritmos distribuidos [20]. Este estudio presenta datos biomédicos almacenado en la nube y demuestra como tales algoritmos son ideales para resolver grandes y problemas escalables.

En esta tesis doctoral hemos explotado los silos de datos, extrayendo la información y almacenándolos en bases de datos que posteriormente pueden ser usadas por el investigador. Se transforman los registros de diagnósticos médicos en datos difusos, a través de las relaciones de comorbilidad generadas a partir de ellos se puede obtener una clasificación sobre los pacientes de manera que, se pueda representar una trazabilidad sobre las diferentes enfermedades acontecidas a lo largo del historial del paciente.

2.2. Plataforma de análisis de datos

2.2.1. Concepto

El concepto plataforma se lleva estudiando desde 2015 [10], cuando se discutían las diferentes definiciones sobre este término. En la literatura existen numerosas definiciones de lo que es una plataforma de datos. Para [18] es una interfaz web para coleccionar, almacenar y administrar un conjunto de datos organizados de manera estructurada. Un concepto similar es desarrollado por [14], donde cuatro plataformas de datos diferentes son analizadas y comparadas. Este autor considera varias dimensiones para definir una plataforma: el acceso a los datos, el uso de los datos y la gobernanza de los datos (gestión de acceso).

En el libro [34], una plataforma de datos es capaz de proveer todas las operaciones necesarias para la implementación de un análisis de datos. Existen otros autores como [34, 21, 27],

que no incluye el proceso de análisis de datos dentro de la plataforma. En otras palabras, el propósito de la plataforma de datos no es obtener, almacenar, procesar y preparar los datos para su análisis, sino exclusivamente la ejecución de algoritmos usando una serie de datos como entrada. Sin embargo otros autores proponen la implementación de sus propias herramientas de preprocesamiento de los datos considerando este proceso como parte de la plataforma de datos [17, 5, 12].

Algunas funcionalidades relevantes de las plataformas de datos son las siguientes:

- El almacenamiento y tratamiento de diferentes tipos de datos en diferentes dominios.
- Ejecución de algoritmos y técnicas de Inteligencia Artificial para predecir, clasificar y obtener nuevas características sobre los conjuntos de datos.
- Elementos y técnicas que garantizan la seguridad de acceso a los datos dentro de las plataformas de datos que gestionan su disponibilidad, integridad e usabilidad
- Escalabilidad y flexibilidad del sistema siendo usado por diferentes propósitos con datos de diversa naturaleza y ámbitos.
- Computación en la nube mediante la integración dentro de IaaS (Infrastructure as a Service, Infraestructura como Servicio), como Google, Azure, AWS (Amazon Web Service, Servicios Web de Amazon) dando la posibilidad del uso de técnicas de Big Data con diferentes frameworks sobre los datos.

Hoy día, las diferentes arquitecturas de almacenamiento de datos y operaciones sobre ellos están experimentando un crecimiento notable [34], desde los almacenes de datos (*data warehouses*) a los repositorios masivos (*data lakes*) pasando por los silos de datos. Este aumento se debe a un incremento en la cantidad de datos semiestructurados y desestructurados, a la popularidad de las arquitecturas de microservicios que no tienen asociados ninguna base de datos central y a la necesidad de satisfacer las cinco Vs *variabilidad, volumen, velocidad, veracidad y valor*.

Los conceptos de data warehouse, data lake y silo de datos se suelen denominar común y conjuntamente como MDP (*Modern Data Platform, Plataforma de Datos Moderna*). En general, un MDP provee más características dirigidas a las necesidades de los usuarios. Algunas de las características destacables es que no necesita de un esquema rígido para manejar los datos ya que con la ayuda de marcos de trabajo como Spark [33] se realiza computación distribuida.

Al necesitar un amplio conocimiento sobre ciencia de datos y programación, los investigadores pueden llegar a perder la oportunidad de poder estudiar este tipo de datos en el estudio de sus dominios. Otro de los retos que intenta abordar las plataformas de datos son la integración de datos heterogéneos dentro ella.

Los datos estructurados poseen una serie de atributos y relaciones con otros dentro del conjunto o fuente de los datos. Ejemplos de este tipo de datos son documentos en XML (eXtensible Markup Language) y JSON (JavaScript Object Notation). Sin embargo, los datos desestructurados no poseen ninguna relación con otros dentro del propio conjunto. Muchos de estos datos son por ejemplo, imágenes, vídeos, audios. En nuestra tesis doctoral se han realizado dos tipos de integraciones dirigidas a la disponibilidad y transformación de los datos y a la cohesión entre diferentes estructuras de datos dentro de la propia plataforma.

Poseer diferentes tipos de datos en la plataforma debería requerir más esfuerzo en aspectos como son la administración de las bases de datos o la implementación de diferentes

algoritmos. Por tanto, a la hora de desarrollar una plataforma se debe tener en cuenta estos dos aspectos.

Otro aspecto a tener en cuenta es el paradigma de computación en la nube sobre las plataformas de datos. Algunas de las características que cita [34] son la elasticidad, modularidad (recursos propios en almacenaje y computación), disponibilidad (recursos disponibles en cualquier momento) y rapidez en el desarrollo de nuevos recursos (introducción rápida de nuevas características en el entorno de producción).

También pueden ser capaces de utilizar infraestructuras como un servicio (IaaS) y servicios de computación en la nube. Esta característica en una plataforma de datos es una gran ventaja debido a que permite escalabilidad y flexibilidad que no puede ser garantizada con un servidor estándar. La plataforma de datos propuesta en [5, 31] usa servicios a demanda, permitiendo administrar la plataforma para incrementar el almacenaje y las capacidades computacionales de sus sistemas basado en el tráfico y necesidades. En la plataforma desarrollada en esta tesis doctoral se implementa una solución para cada una de las características que define la infraestructura de computación en la nube

El concepto Big Data se aplica en salud, electrónica, biología, banca, meteorología y otros muchos campos que afectan a la vida diaria de las personas. Implementar arquitecturas de Big Data para organizaciones requiere obtener caras licencias de software, preparar una sofisticada infraestructura, y pagar a expertos cuyo conocimiento sea usar el sistema, organizar e integrar los datos generados para su análisis [2]. Aquí surge el término BDP (*Big Data Platform*, Plataforma aplicada al Big Data), un tipo de solución capaz de proveer los mismos servicios y características como plataforma de datos pero trabajando con conjuntos de datos masivos. Nuestra plataforma va dirigida al uso de este conjunto de datos masivos.

Esta plataforma nos ayudará a resolver algunos problemas relacionados con la explotación de grandes cantidades de datos almacenados en silos de datos[15]. Esto significa que la inmensa mayoría de las investigaciones realizadas por los propios investigadores fuera del ámbito de la Ciencia de Datos generan una inmensa cantidad de datos que no han podido ser explotados. Los problemas en entornos multidisciplinarios y la cantidad de silos de datos sin explotar son dos razones que motivaron el desarrollo de esta tesis doctoral.

Un caso de uso especialmente destacable fue durante y después del Covid-19. Para muchos investigadores en el campo médico fue crucial el uso de plataformas de este tipo para que sus investigaciones pudiesen salvar muchas vidas y administrar acciones de gobiernos y toma de decisiones sobre la sociedad [32]. Los autores de [25] y [16] muestran como en ámbitos multidisciplinarios se aplica en una situación real.

El problema que presentan la mayoría de las plataformas de datos hasta el día de hoy es que están muy dirigidas a un determinado ámbito. Esto provoca que surjan plataformas especializadas y no genéricas que puedan manejar cualquier conjunto de datos.

2.2.2. Técnicas y aplicaciones

En la plataforma de datos llevada a cabo en esta tesis doctoral se desarrolla enfocada a las características descritas en la sección anterior. En nuestro caso, la plataforma desarrollada puede trabajar con datos de diferente dominio, siendo los casos de uso incluidos en esta tesis doctoral de eficiencia energética y de medicina.

En uno de los trabajos realizado en esta tesis doctoral se ha implementado una plataforma encapsulando módulos dentro de la propia plataforma. En la figura 2.2 podemos observar la implementación de la plataforma.

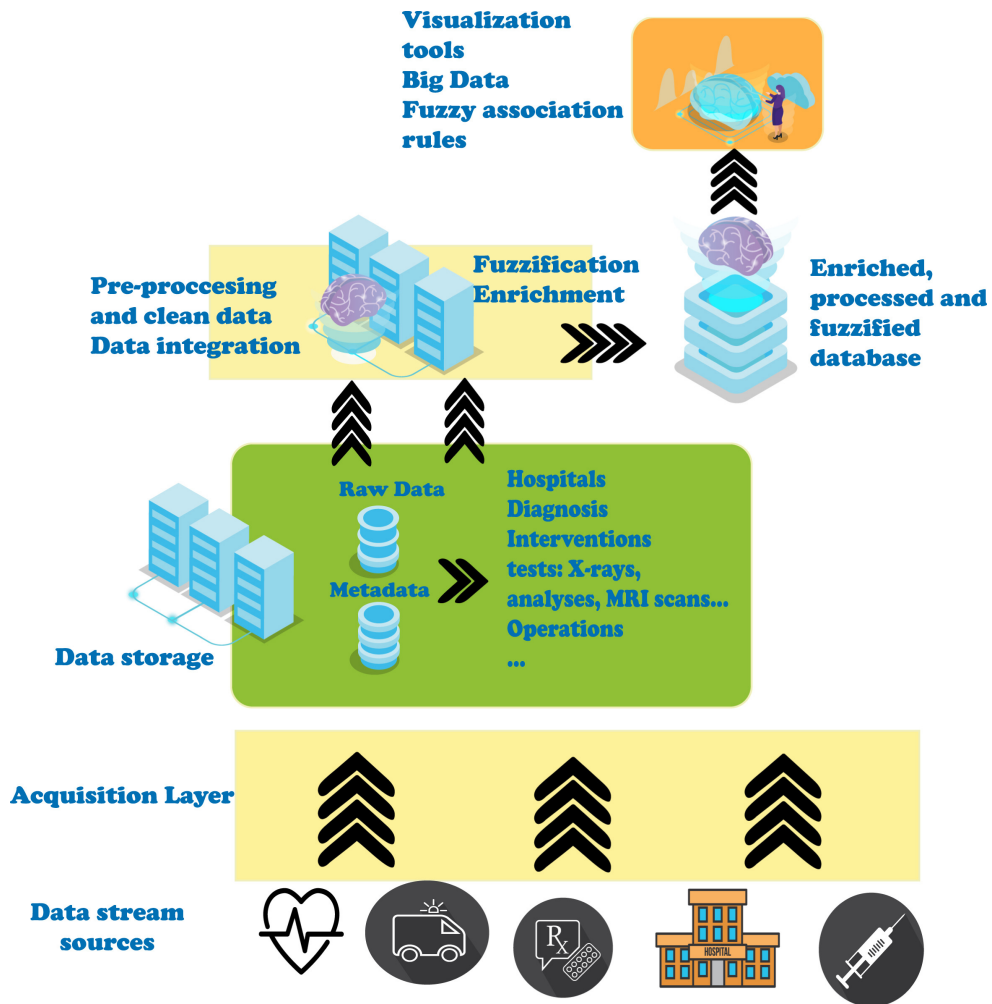


Figura 2.2: Plataforma moderna de datos

2 Preliminares

Como ya hemos dicho anteriormente una de las principales limitaciones de muchas plataformas es la dificultad de adaptación a cualquier dominio de investigación. Esto se debe a la complejidad de algunos conjuntos de datos, principalmente en el dominio de la salud, siendo demasiado complicado conseguir ensamblar estas estructuras de datos dentro de la propia plataforma.

Por tanto, en esta tesis doctoral se ha focalizado en trabajar sobre este punto, creando una plataforma compleja que consiga separar la funcionalidad de cada módulo en diferentes capas, agrupando cada una en sus diferentes funciones trabajando cada una de manera independiente. De esta manera la plataforma es capaz de adaptarse a cualquier problema independientemente del dominio sobre el que se quiera trabajar. También se evita la propagación de errores haciendo que la plataforma sea más robusta y no se propague por el resto de capas consiguiendo reducir los errores críticos en el sistema. En la siguiente figura 2.3 se muestra la plataforma construida en esta tesis doctoral.

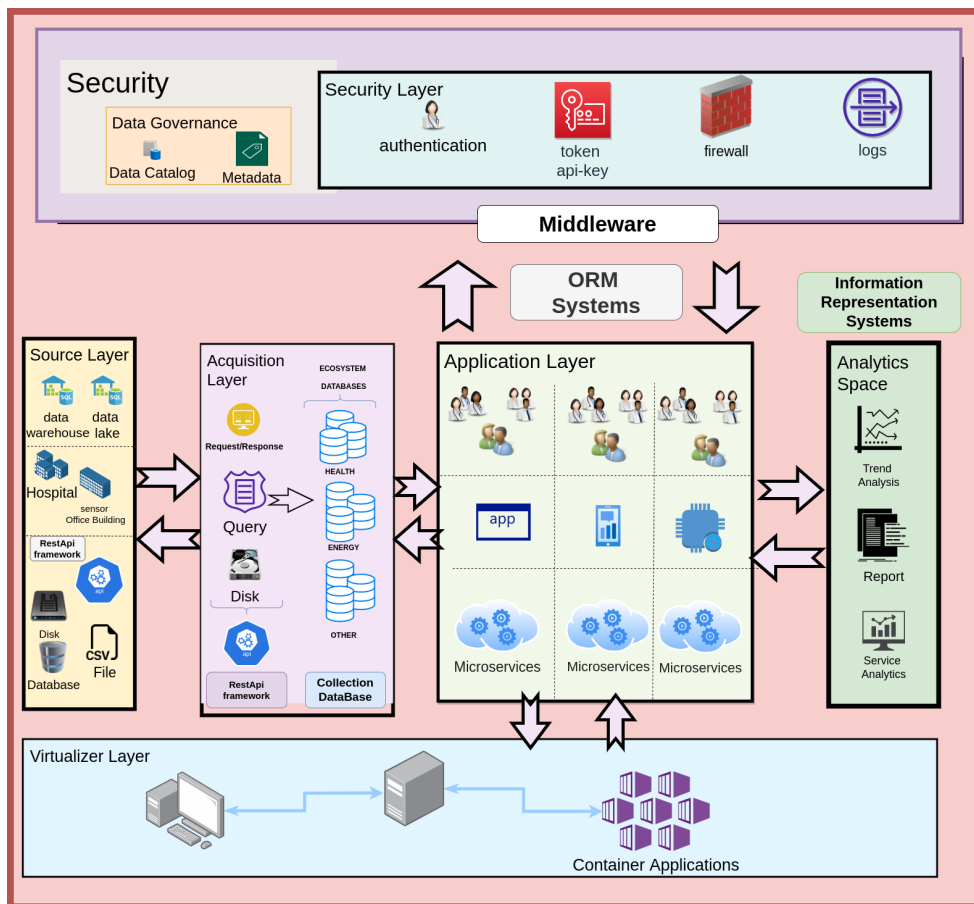


Figura 2.3: Plataforma de Datos aplicada sobre Inteligencia Artificial

3 Resumen y Resultados

En esta sección se presentan las publicaciones realizadas durante la realización de esta tesis doctoral. En cada una de las siguientes subsecciones se muestra un resumen que describe la idea general del trabajo. A continuación se detallan los trabajos mencionados anteriormente.

- R. Morcillo-Jimenez, K. Gutiérrez-Batista and J. Gómez-Romero, «Tsxtend: A tool for batch analysis of temporal sensor data», *Energies*, vol. 16, no. 4, 2023, issn: 1996-1073. en16041581
- R. Morcillo-Jimenez, Jesus Mesa, Juan Gomez-Romero, M.Amparo Vila and Maria J. Martin-Bautista, «Deep Learning for Prediction of Energy Consumption: An Applied Use Case in an Office Building». 324234234
- A. S. Ortega-Calvo, R. Morcillo-Jimenez, C. Fernandez-Basso, K. Gutiérrez-Batista, M.-A. Vila and M. J. Martin-Bautista, «Aimdp: An artificial intelligence modern data platform. use case for spanish national health service data silo», *Future Generation Computer Systems*, vol. 143, pp. 248–264, 2023, issn: 0167-739X. ORTEGACALVO2023248
- C. Fernandez-Basso, K. Gutiérrez-Batista, R. Morcillo-Jiménez, M.-A. Vila and M. J. Martin-Bautista, «A fuzzy-based medical system for pattern mining in a distributed environment: Application to diagnostic and co-morbidity», *Applied Soft Computing*, vol. 122, p. 108 870, 2022, issn: 1568-4946. FERNANDEZBASSO2022108870

El resto de la sección está organizado como sigue: En primer lugar, la Sección 3.1 muestra la herramienta implementada denominada TSxtend (Time Series Extend, Series Temporales Ampliadas) donde se resuelven problemas de series temporales agrupando los datos de diferentes tipos de edificios (ASHRAE) con datos preprocesados previamente. En la Sección 3.2, se emplea esta metodología en un conjunto de datos sin procesar de un edificio de oficinas (ICPE). Se ha incorporado el módulo de preprocesamiento en la herramienta TSxtend para realizar predicciones sobre el consumo energético del edificio. A continuación, en la Sección 3.3, se muestra cómo los algoritmos difusos mejoran la predicción de comorbilidad asociada a una enfermedad previa. Por último, en la Sección 3.4, se presenta una plataforma creada para tratar los datos de distintos silos de información de dos hospitales de Andalucía: el Hospital de Marbella (Málaga) y el Hospital Clínico San Cecilio (Granada). Esta plataforma permite extraer todo el conocimiento necesario.

3.1. TSxtend: Una herramienta integral para el preprocesamiento y análisis de datos temporales de sensores en investigación.

En este trabajo se introduce TSxtend [23], una herramienta que aborda los numerosos desafíos asociados al preprocesamiento y análisis de datos generados por sensores. TSxtend integra diversas técnicas ya existentes para la partición, limpieza e imputación de series

temporales, además de aprovechan algoritmos de aprendizaje automático para realizar predicciones con alta precisión.

Simplifica el proceso de transformación, limpieza y análisis de series temporales de sensores mediante el uso de un lenguaje declarativo para definir y ejecutar flujos de trabajo. Esta herramienta tiene como objetivo capacitar a los usuarios para que puedan explotar sus datos, lo que les permite tomar decisiones oportunas e informadas durante el proceso de investigación. Con TSxtend, los usuarios sin conocimientos de programación pueden analizar rápidamente sus datos, lo que les permite concentrarse en aplicar su investigación y en la comprensión de sus resultados.

Además, la herramienta ofrece funcionalidades para la definición y seguimiento de experimentos, y destaca por su arquitectura modular, que permite la fácil incorporación de métodos adicionales. Los ejemplos presentados en este artículo, utilizando el conjunto de datos de ASHRAE Great Energy Predictor, demuestran la eficacia de TSxtend en el análisis de series temporales con datos de energía procesados.

La aplicación de la Ciencia de Datos ha generado un gran avance en diferentes ámbitos de la investigación. El problema que plantea los algoritmos que trabajan sobre esta ciencia es la necesidad de poseer un amplio conocimiento de este tipo de técnicas que en la mayoría de los casos se sale del ámbito del investigador.

El hecho de resolver esta problemática provocó el comienzo del estudio de una metodología que fuese capaz de realizar las diferentes etapas que abarca la ejecución de este tipo de problemas planteados por esta ciencia. Es por ello que la principal motivación de este trabajo trata de desarrollar una herramienta que abarque todo el flujo de trabajo sobre un problema de ciencia de datos.

TSxtend se ha desarrollado siguiendo un enfoque amigable para el usuario, lo cual mejora significativamente la experiencia al ejecutar algoritmos y obtener resultados preliminares para el análisis de datos. Además, se logró diseñar un flujo de trabajo flexible en la herramienta que permitió la integración de técnicas adicionales sin afectar a las ya implementadas.

Los resultados obtenidos al aplicar TSxtend al conjunto de datos ASHRAE Great Energy Predictor demuestran su eficacia en el análisis de datos relacionados con la eficiencia energética, especialmente cuando los datos ya han sido procesados. Esta herramienta representa una valiosa contribución al campo del análisis de series temporales, ya que puede mejorar y aumentar la productividad de investigadores y profesionales en diversas áreas.

- **R. Morcillo-Jimenez**, K. Gutiérrez-Batista and J. Gómez-Romero, «Tsxtend: A tool for batch analysis of temporal sensor data», *Energies*, vol. 16, no. 4, 2023. <https://doi.org/10.3390/en16041581>.

3.2. Análisis de series temporales y predicción de consumo de energía en edificios utilizando técnicas de aprendizaje profundo

Los edificios no residenciales tienen un impacto significativo en el consumo global de energía, representando más de un tercio del mismo. Para identificar ineficiencias y optimizar las políticas de gestión energética, es fundamental estimar el consumo de energía en estos edificios. En este artículo, se presenta un estudio que se centra en técnicas de aprendizaje profundo aplicadas al análisis de series temporales para predecir el consumo energético en edificios no residenciales.

Se recopilaron datos de sensores provenientes de diversas fuentes en un edificio de oficinas en condiciones normales de funcionamiento. Estos datos fueron preprocesados y se llevó a cabo una evaluación exhaustiva de la precisión de las redes neuronales en la predicción del consumo de energía.

Después de confirmar la efectividad de la metodología implementada en la herramienta TSxtend en un conjunto de series temporales preprocesadas, se tomó la decisión de aplicar esta metodología a conjuntos de datos temporales más realistas, que aún no han sido sometidos a preprocesamiento. Para realizar este trabajo se tomaron los datos del edificio ICPE que consistía en un edificio de tres plantas divididas en diferentes áreas del que se tomaron mediciones en bruto de los propios sensores.

Los resultados obtenidos validaron la eficacia de una metodología basada en lograr resultados rápidos en conjuntos de datos sin procesar. En este estudio, se han aplicado diferentes etapas de un problema de ciencia de datos, utilizando técnicas de EDA (Exploration Data Analysis, Análisis Exploratorio de Datos), para filtrar y dividir los datos de consumo de energía en un edificio. Al finalizar el procesamiento de todo el conjunto de datos, se realizó una comparación entre varios algoritmos de aprendizaje ampliamente utilizados. Los resultados destacan que los algoritmos que no consideran el orden cronológico presentan un rendimiento inferior en comparación con aquellos que sí lo tienen en cuenta.

Los resultados obtenidos en este estudio indican que el modelo XGBoost(Extremme Gradient Boosting) muestra un rendimiento inferior debido a su incapacidad para capturar las dependencias entre las variables de entrada en la predicción del consumo de energía. Por otro lado, los modelos MLPs (Multi-Layer Perceptron, Preceptron Multi Capa) no ofrecen mejoras significativas en comparación con otros modelos, ya que no permiten tener en cuenta el orden cronológico en el conjunto de datos analizados. Por el contrario, al utilizar modelos RNN (Recurrent Neuronal Network, Redes Neuronales Recurrentes) en el análisis de datos, se logró reducir el error de la función NMAE (Normalized Mean Absolute Error, Error absoluto medio normalizado) a la mitad, ya que estos algoritmos pueden recordar la información al procesar los datos en la secuencia cronológica.

Adicionalmente, hemos aplicado el enfoque Seq2Seq (Sequences to Sequences, Secuencia a Secuencia) en nuestro estudio, lo cual nos ha permitido observar que, al igual que en las redes RNN, este método obtiene resultados favorables en general. Sin embargo, hemos identificado que tanto RNN como Seq2Seq presentan dificultades al lidiar con conjuntos de datos que contienen un alto número de valores perdidos, especialmente cuando estos conjuntos son de tamaño reducido. Por último, hemos encontrado que las redes CNN (Convolutional Neuronal Network, Red Neuronal Convolutacional) muestran un mejor desempeño que los algoritmos RNN y Seq2Seq en secciones del conjunto de datos caracterizados por una gran cantidad de valores perdidos y tendencias estables.

- **R. Morcillo-Jimenez**, J. Mesa, J. Gómez-Romero, M.-A. Vila and M.J. Martin-Bautista, «Deep learning for prediction of energy consumption: an applied use case in an office building». (Enviado a la revista Applied Intelligence).

3.3. Extracción de conocimiento oculto en registros médicos mediante minería de datos y lógica difusa en un entorno distribuido.

Un desafío significativo en este dominio es que aunque hay muchos estudios dedicados al análisis de datos de salud, muy pocos se centran en la comprensión, interpretación de los datos y los patrones ocultos presentes dentro de dichos datos. En este trabajo se aborda la extracción de conocimiento oculto de registros médicos utilizando técnicas de minería de datos como reglas de asociación en conjunto con lógica difusa en un entorno distribuido.

Un gran desafío en esta área es que muchos estudios de análisis de datos de salud se han centrado en la clasificación, la predicción o la extracción de conocimiento y los usuarios finales encuentran poca interpretabilidad o comprensión de los resultados. Esto se debe al uso de algoritmos de caja negra o porque la naturaleza de los datos no se representa correctamente. Es por eso que es necesario centrar el análisis no solo en la extracción de conocimiento sino también en la transformación y procesamiento de los datos para mejorar la modelización de la naturaleza de los datos. Técnicas como la extracción de reglas de asociación y la lógica difusa ayudan a mejorar la interpretabilidad de los datos y tratarlos con la incertidumbre inherente de los datos del mundo real.

Con este fin, se propone una plataforma de datos que automáticamente: a) preprocese la base de datos transformando y adaptando los datos para el proceso de minería de datos y enriqueciendo los datos para generar patrones más interesantes, b) que realice la difusión de la base de datos médica para representar y analizar datos médicos del mundo real con su incertidumbre inherente, c) descubrir interrelaciones y patrones entre diferentes características (diagnósticos, alta hospitalaria, etc.), y d) visualizar los resultados obtenidos de manera eficiente para facilitar el análisis y mejorar la interpretabilidad de la información extraída.

La plataforma de datos propuesta produce un aumento significativo en la comprensión y la interpretabilidad de los datos médicos para los usuarios finales, lo que les permite analizar los datos correctamente y tomar las decisiones correctas. Se presenta un caso práctico utilizando un conjunto de datos relacionados con la salud para demostrar la viabilidad de nuestra propuesta con datos reales.

El descubrimiento y explotación de información recopilada en hospitales han atraído la atención debido a su impacto económico y en la salud en la última década. El uso de lógica difusa puede mejorar la interpretabilidad de los datos recopilados, ofreciendo mejores resultados e interpretación a los usuarios finales.

El objetivo principal de este estudio ha sido extraer y analizar conocimientos ocultos presentes en los registros médicos. Para lograrlo, se ha aplicado una serie de algoritmos difusos sobre el diagnóstico de enfermedades de un paciente para obtener una predicción de su diagnóstico.

Nuestra propuesta fue validada a través de experimentos utilizando datos reales, demostrando así su viabilidad. Los resultados obtenidos revelan la capacidad de nuestra propuesta para descubrir reglas de interés, como por ejemplo, 'la presencia de alcoholismo y dependencia de drogas se relaciona con problemas digestivos en pacientes frecuentes', o 'el diagnóstico de diabetes concomitante a problemas cardíacos o respiratorios indica un diagnóstico complejo y pacientes en estado grave'. Estas reglas tienen aplicaciones en la predicción y prevención de enfermedades, así como en el análisis de la comorbilidad y las relaciones entre diferentes características relevantes.

- C. Fernandez-Basso, K. Gutiérrez-Batista, **R. Morcillo-Jiménez**, M.-A. Vila and M.J. Martin-Bautista, «A fuzzy-based medical system for pattern mining in a distributed environment: Application to diagnostic and co-morbidity», *Applied Soft Computing*, vol. 122, p. 108 870, 2022. <https://doi.org/10.1016/j.asoc.2022.108870>.

3.4. AIMDP: Plataforma de datos basada en Big Data y IA para la gestión eficiente y análisis de información en entornos heterogéneos.

La enorme cantidad de datos que se manejan hoy en día en cualquier entorno, como el energético, económico o sanitario, hace que los sistemas de gestión de datos sea clave para extraer información, analizar y crear procesos diarios más eficientes en estos entornos. Sin embargo, la incapacidad de los sistemas actuales para aprovechar los datos generados puede desperdiciar buenas oportunidades para analizar y extraer información de los datos.

Este trabajo presenta una plataforma de datos llamada AIMDP (Artificial Intelligence Modern Data Platform, Plataforma de Datos Basada en Big Data y IA) que aplica inteligencia artificial a la gestión y manejo eficiente de datos. Los diferentes componentes de AIMDP intervienen en la fase de adquisición de datos e implementan algoritmos capaces de analizar datos masivos recopilados de diferentes fuentes.

Además, la plataforma se centra en la gestión y aprovechamiento de datos, incorporando una capa de seguridad y gobernanza de datos que garantiza la integridad y privacidad de las bases de datos. Se ha diseñado la plataforma propuesta pensando en usuarios que no son expertos en ciencia de datos. Para este propósito, se ha implementado un marco de trabajo orientado al usuario que ha sido exitosamente aplicado en un caso de uso en dos hospitales andaluces Hospital Costa del Sol (Malaga) y Hospital Clinico San Cecilio (Granada). Este enfoque permitió extraer conocimiento de los datos históricos de los hospitales, los cuales estaban almacenados en silos de datos y nunca habían sido explorados por investigadores o médicos del hospital.

En este trabajo se ha demostrado la efectividad de contar con una plataforma que permita a usuarios no expertos extraer conocimientos ocultos haciendo uso de tecnologías innovadoras. Esta plataforma ya incluye funcionalidades para la importación, procesamiento y enriquecimiento de datos, así como técnicas de inteligencia artificial para extraer conocimientos a partir de grandes conjuntos de datos.

El resultado final de este trabajo es proporcionar a los usuarios herramientas novedosas que les permitan gestionar y procesar eficientemente los grandes volúmenes de datos generados por redes sociales, registros médicos, imágenes, sensores y otras fuentes externas. Todo esto se logra mediante el aprovechamiento de la computación distribuida y la inteligencia artificial, en un proceso guiado, transparente y de fácil aplicación por parte del usuario.


- A. S. Ortega-Calvo, **R. Morcillo-Jimenez**, C. Fernandez-Basso, K. Gutiérrez-Batista, M.-A. Vila and M.J. Martin-Bautista, «Aimdp: An artificial intelligence modern data platform. use case for Spanish national health service data silo», *Future Generation Computer Systems*, vol. 143, pp. 248–264, 2023. <https://doi.org/10.1016/j.future.2023.02.002>.

4 Publicaciones

4.1. TSxtend: Una herramienta integral para el preprocesamiento y análisis de datos temporales de sensores en investigación.

- Referencia:0360-5442
- Estado: Aceptado
- Factor de Impacto: Q1
- Categoría: Energy
- DOI: <https://doi.org/10.3390/en16041581>
- Revista/Editorial: Energy

TSxtend: A Tool for Batch Analysis of Temporal Sensor Data

Roberto Morcillo-Jimenez , Karel Gutiérrez-Batista  and Juan Gómez-Romero * 

Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain

* Correspondence: jgomez@decsai.ugr.es

Abstract: Pre-processing and analysis of sensor data present several challenges due to their increasingly complex structure and lack of consistency. In this paper, we present TSxtend, a software tool that allows non-programmers to transform, clean, and analyze temporal sensor data by defining and executing process workflows in a declarative language. TSxtend integrates several existing techniques for temporal data partitioning, cleaning, and imputation, along with state-of-the-art machine learning algorithms for prediction and tools for experiment definition and tracking. Moreover, the modular architecture of the tool facilitates the incorporation of additional methods. The examples presented in this paper using the ASHRAE Great Energy Predictor dataset show that TSxtend is particularly effective to analyze energy data.

Keywords: time series; pre-processing; prediction; machine learning; deep learning

1. Introduction

The development and growth of information and communication technologies have precipitated the daily generation of massive data. Today, much of the generated data come from sensor data. Temporal sensor data have become of great interest to the academic and private sectors, as studying this sort of data allows for studies of the evolution of data over time, providing end-users with robust algorithms and tools for decision-making.

There are several applications that use sensor data for various purposes [1,2], such as handling and managing measures such as temperature, humidity, pressure, gas, optical, and many others. Many companies in different industries are becoming increasingly aware of the great potential that the research and exploitation of temporal sensor data can offer [3,4].

There are many challenges in dealing with temporal sensor data. In the following, we mention the main challenges concerning this sort of data:

1. One of the main challenges is dealing with the large volumes of data that are generated by sensors each day. This can make storing, managing, and processing the data difficult, requiring specialized tools and techniques.
2. Another challenge is the variability and noise in the data, which makes it difficult to identify trends and patterns. This may require advanced data filtering and cleaning methods to remove irrelevant or inaccurate information.
3. Additionally, the lack of consistency and complex structure inherent in temporal sensor data make its processing and analysis even more difficult. Sophisticated algorithms and techniques may be required to analyze and interpret the data appropriately.
4. Furthermore, temporal sensor data are often harvested from heterogeneous sources. This can be challenging, requiring efficient integration approaches to handle the data in a timely manner.

All the aforementioned challenges hinder the definition of a workflow that allows for promising results to be obtained. Furthermore, selecting the most suitable algorithm to solve the problem constitutes another burdensome task. Solving this problem is hard work due to the nature and variability of each related problem.



Citation: Morcillo-Jimenez, R.; Gutiérrez-Batista, K.; Gómez-Romero, J. TSxtend: A Tool for Batch Analysis of Temporal Sensor Data. *Energies* **2023**, *16*, 1581. <https://doi.org/10.3390/en16041581>

Academic Editor: Fernando Morgado-Dias

Received: 24 December 2022

Revised: 23 January 2023

Accepted: 1 February 2023

Published: 4 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

One of the most significant challenges in prediction problems for time series and any other data type is the time needed to design the correct strategy for the experiment. Different studies focus on the application of specific algorithms to a data series using a static configuration [5–7]; in other words, without offering the possibility of parameterising the different experiments. This consumes a great deal of time for each experiment definition, thus slowing down the researcher’s main objective.

As stated before, time series data have gained the attention of the entire research community, as its analysis allows for an understanding of changes in the data over time. By analyzing the trends and patterns in the data, insights into how the data are evolving can be acquired, and predictions about future results can be made. This is particularly useful in fields where changes over time can significantly impact decision-making and strategy. Additionally, temporal sensor data analysis can help to identify potential issues or anomalies, allowing for timely interventions and corrective actions.

This paper proposes a tool called TSxtend for time series analysis. TSxtend presents a modular architecture and standardises the different stages of experimentation, creating a workflow through a simple configuration file. The proposed tool allows for the end-users to work on time series data without programming knowledge using the available techniques and focusing on developing the research. The tool enables data filtering and cleaning to remove incorrect or nonessential information. Finally, TSxtend has the ability to execute deep prediction and machine learning algorithms that are commonly utilized in various domains, such as energy [8], medicine [9], or geoscience [10,11]. This allows for a wide range of applications and flexibility in its usage. The results are presented through the visualization module, enabling end users to analyze the results at different workflow stages. A summary of the main contributions of this paper is as follows:

- The paper’s main contribution is the development of a tool called TSxtend for time series analysis, which has a modular architecture and standardizes different stages of experimentation.
- The tool allows for end-users to work on time series data without programming knowledge, simplifying the process of data filtering and cleaning to remove incorrect or nonessential information.
- TSxtend offers the possibility of executing prediction algorithms and visualizing the results through a visualisation module, enabling end-users to analyse the results at different workflow stages.

It is important to note that, to the best of the authors’ knowledge, there is no existing tool in the literature that has the same abilities as TSxtend. This tool is unique in its ability to standardize the experimentation process, simplify time series analysis for end-users without programming knowledge, and provide a visualisation module for a better analysis of results.

In this paper, we apply the proposed tool in an energy consumption problem. We obtained the data from the prediction features contest on the Kaggle platform called ASHRAE—Great Energy Predictor III [12]. The database comprised three years of hourly meter readings from more than a thousand buildings in different locations worldwide. It should be noted that the database belongs to ASHRAE, a large building technology association [13].

The rest of the paper is structured as follows: Section 2 reviews previous work on this topic. Section 3 presents the design and functionalities details of the presented tool (Tsxtend). Section 4 extends the description of the different modules that comprise the tool. In order to showcase the feasibility of the proposed tool, Section 5 presents a real-world use-case using TSxtend and discusses the obtained results. Finally, in Section 6, the conclusions and future research are presented.

2. Related Works

Many applications aim to make the work of researchers working with time series data easier [14–18]. This is driven by the need to abstract programming knowledge to a higher level, allowing for researchers who are not experts in data science to focus on studying the data rather than the technical details of the algorithm. Some tools that can be used

to achieve this abstraction, isolating the researcher from the technical complexities, are described in the following.

To make the literature review easier to understand and better highlight the novelty of this research, a comparative study of libraries was conducted. Various aspects of the libraries were analysed, such as their ability to collect and use heterogeneous data sources, the capacity of the processing tools available within the library, and the REST API abilities of the libraries. Furthermore, we also evaluated whether the libraries can implement Artificial Intelligence (AI) algorithms, if they have been used in real-world use-cases and if they are user-friendly. The ease of use is particularly important when the libraries are used by non-expert data-mining users. Through this analysis, we aim to provide a comprehensive comparison of the different libraries and their features, to aid in the understanding of the significance of this research.

One example of such a tool is Enlopy [19], which is an open-source tool developed in Python that offers a variety of methods for processing, analyzing, and plotting time series data. This tool has modules that are primarily focused on studying time series data, with capabilities such as analysis techniques, graphing, data augmentation, and feature extraction, mainly in the energy domain. However, it should be noted that this tool requires a significant amount of programming knowledge, making it inaccessible to researchers who are not experts in data science.

The TSSA tool [14] primarily focuses on the preprocessing stage, to obtain the correlation between the resistance variables of memories in different devices. This is not suitable for work with heterogeneous data and requires extensive knowledge in data science. Additionally, it does not have a REST API for data retrieval, which limits its functionality and accessibility for users.

Another tool that addresses the work with time series is tsfresh [20]. This tool enables the study of time series data by extracting features and training simple classification or regression models. However, it should be noted that this tool requires a significant amount of programming knowledge to be used to its full extent. Additionally, it does not offer a workflow feature to guide users through their work process. In [18], a Visual Warning Tool for Financial Time Series (VWSTFTS) is presented based on scaling analysis. The proposed method uses the time-dependent Generalised Hurst Exponent (GHE) method to analyse financial time series and identify temporal patterns in GHE profiles. By applying this methodology and using a visual tool, the researchers can analyse significant and peripheral stock market indices. The proposed method offers a new way of identifying patterns in financial time series data and can be used to provide early warning signals for market fluctuations. However, the tool is limited in its capabilities and does not support AI techniques. Additionally, it is primarily used in the financial domain and might not be suitable for other fields.

Darts [21] is one of the most comprehensive tools available, providing a wide range of algorithms for working with time series. Darts supports both univariate and multivariate time series and models. Acycle [17] is a specialized software for paleoclimate research and education that focuses on signal processing, particularly for cyclostratigraphy and astrochronology. It includes various models, such as sedimentation noise and sedimentation rate, which are specific to sedimentary research. Acycle's fully implemented graphical user interface makes it easy for users to operate and navigate the software, making it user-friendly for both researchers and educators. It should be noted that neither Darts nor Acycle includes a preprocessing phase, meaning that the data need to be cleaned, transformed and prepared before being used with these tools. This could mean that additional steps and resources are required for the effective use of these tools.

Another state-of-the-art tool for time series analysis is Kats [22]. This tool aims to make time series analysis more accessible to researchers with a solid background in data science. Kats offers a variety of forecasting algorithms, such as individual forecasting models, ensemble models, a self-supervised learning model (meta-learning), backtesting, hyperparameter fitting and empirical prediction intervals. This tool is a comprehensive and powerful option for researchers looking to perform advanced time series analysis.

TSFEL [23] is a tool that primarily focuses on the preprocessing stage of time series analysis, providing a range of options for exploratory analysis and feature extraction. One of its notable features is the inclusion of a set of unit tests to verify the runs. However, like many other tools, TSFEL can be challenging to use for researchers without strong programming skills.

SSTS [16] is a tool for searching patterns in time series data. It utilises a syntactic approach, analysing the structure and relationships within the data. SSTS allows for users to specify complex patterns using formal grammar and search for instances of a pattern. The tool can identify patterns that are difficult to detect using traditional methods such as statistical analysis or machine learning. Additionally, SSTS can extract features from time series data for further analysis or to train predictive models. It is a powerful tool for searching patterns in time series data that can be used in various fields, such as finance, healthcare, and transportation.

cleanTS [15] is an automated tool for cleaning univariate time series data. It utilises machine learning techniques to remove noise, outliers and missing values, helping to improve the accuracy and reliability of time series analysis. It also makes it easier to identify patterns and trends in the data. cleanTS can also be used to interpolate missing data and resample the data at different time scales. This tool can be applied in various domains, such as financial time series, sensor data and other time series data. It is a useful tool for the preprocessing of time series data, making it more reliable and accurate for further analysis.

Another library that allows for working with time series data is GreyKite [24]. This tool provides preprocessing capabilities as well as prediction models for heterogeneous data. One of its unique features is its own algorithm, which can be used for creating prediction models, called *Silverkite*. GreyKite is also aimed at data scientists. Another complete tool is AutoTS [25], which utilizes other libraries for the generation of prediction models with the help of a framework. As with other tools, it requires a good understanding of programming. Table 1 provides a comparison of the libraries mentioned above with the tool proposed in this paper.

Table 1. Comparison of the proposed tool with the tools mentioned earlier for analysing temporal sensor data. Legend: ✓—Feature supported, ✗—Feature not supported, ~—Feature not fully supported or with explicit limitations, ?—Unknown; information not available about the feature.

Name	Heterogeneous Data Collection	Preprocessing Data	REST API	AI Tools	Applications Wide Range of Areas	User Friendly
Enlopy	✓	✓	✗	✗	✗	✗
TSfresh	✗	✓	✗	✓	?	✗
Kats	✗	✗	✗	✓	✓	~
Darts	✓	✗	✗	✓	✓	✗
SSTS	?	✓	✗	✓	✓	✗
TSSA	✗	✓	✗	✗	?	✗
cleanTS	✗	✓	✗	✓	✓	✗
GreyKite	✓	✓	✗	✓	✓	✗
TSfel	✓	✓	✗	✗	✗	~
AutoTs	✓	✗	✗	✓	✓	✗
Acycle	✓	✗	✗	✓	✗	✓
VWSTFTS	✓	✓	✗	✗	✗	~
TSxtend	✓	✓	✓	✓	✗	✓

The proposed tool is designed to facilitate data processing for non-expert users with a simple, user-oriented interface. The tool is primarily aimed at the processing of time series data such as energy consumption, but can also process and analyse various other types of data. Additionally, the library includes commonly used data processing, cleaning, and transformation tools. Its design allows for the easy integration of new methods and algorithms. A wide range of data analysis algorithms is also available. One of the key strengths of TSxtend is its user-oriented design and communication through its REST API, which makes the tool dynamic and easy to use.

All the presented tools are comprehensive libraries for analysing temporal sensor data, helping researchers to study these types of data without the need for extensive knowledge of data science. However, almost all of these tools require programming knowledge, which can be a hindrance to researchers who want to focus solely on data analysis. In contrast, TSxtend not only provides a tool for working with temporal sensor data, but also includes an architecture that allows for researchers to create a workflow for their research by editing a configuration file. This enables researchers to conduct experiments step by step, analysing the data at each stage, without the need for programming knowledge.

3. Design and Functionalities of TSxtend

TSxtend comprises a series of modules for specific tasks. The modules that comprise this tool are grouped into different data science paradigms, each of which offers the possibility of executing different algorithms. In the following section, the software design of our tool will be introduced, and a brief explanation of its functionalities will be provided.

3.1. Design

The main idea behind designing this tool is to create a series of dynamic and flexible modules, allowing for a wide range of operations on the input data and providing a final result. One of the primary advantages of this design approach is that it provides a robust tool, where each component can be executed separately. This allows for each step of the pipeline to be run separately, without necessarily having to run the whole process every time the experiment is performed.

As shown in Figure 1, TSxtend is organised into five modules: one module to store the loggers of the tool, a configuration module, a module to process data, a module to execute machine learning algorithms and a module to execute deep learning algorithms. In the following, we briefly introduce each of these modules.

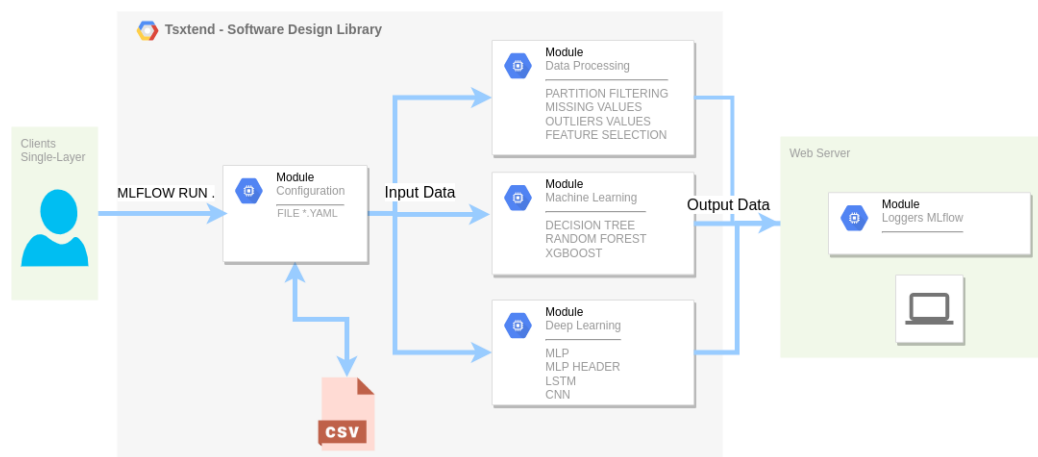


Figure 1. Design of the proposed tool.

3.2. Functionalities

Firstly, through a plain text file, the configuration module allows for the insertion of a series of parameters to configure the experimentation and automate the whole process without programming knowledge.

The data processing module includes a series of data preprocessing algorithms, such as eliminating missing values, eliminating out-of-range data in our input data, and a series of algorithms aiming to obtain the different relationships between the different characteristics of our input data. These algorithms were chosen because they cause one of the most common problems when working with temporal series data.

The machine learning module includes the standard algorithms used in this paradigm, such as XGBoost [26], Decision [27] and Regression Trees [28]. These algorithms were

chosen because they are among the most widely used in the state-of-the-art to solve prediction problems based on structured time series data.

In the deep learning module, the same selection policy for the integrated algorithms was followed as in the previous module. The most widely used algorithms for time series focused on the deep learning paradigm were selected, such as Convolutional Neural Network (CNN) [29], Long-Short Term Memory (LSTM) [30] and Multilayer Perceptron [31].

Finally, the module recorders represent the results and can track them. For this, we integrated Mlflow [32] because it is a tool that is used to keep track of the experiments performed by any user. This tool helps to show the status of the experiments with an accessible, user-friendly interface. In the following sections, we will explain the functionalities of each module in more detail.

4. Module Descriptions

The proposed tool presents a modular architecture that facilitates the effortless incorporation of future functionalities. In the subsequent sections, the characteristics and functionalities of each module will be described in detail.

4.1. Coordination Module

Our tool is mainly focused on carrying out exploratory data analyses in a fast and agile way. In this way, the end-users gain a better understanding of the data, and can identify potential issues or problems with the data and quickly develop more refined analyses. The tool offers a set of prediction algorithms that are mainly used to solve time series problems, such as neural networks. Although the algorithms provided by the tool can be applied to any data type, the proposed library specialises in temporal sensor data regarding energy.

Several works address energy-related problems using data science techniques to identify opportunities for energy-efficiency improvements [33–35]. However, only a few can automate the entire experimentation process. Furthermore, end-users require programming skills to use the library.

The configuration module is one of the most important. It allows for different executions of the defined experiment to be planned. The objective of this module is to configure a road map so that users can schedule the whole experimentation strategy. The main advantage of this approach is that it can endow end-users with a robust tool, which can be handled without programming knowledge.

The controller file, called `MLproject`, contains a definition of the algorithms that are to be used and the configuration parameters. The file reads the data in the YAML files and executes the user-defined workflow automatically.

The entire roadmap of the experimentation is configured through a series of YAML files [36], whose primary function is to define and serialize the different processes for all kinds of programming languages. Each file has a series of parameters that the researcher can edit. The parameters can be consulted in [37], where all the different options accepted by the tool have been detailed. The nomenclature used for this file is based on dictionaries (key:value pair).

Each module of the tool has one of these configuration files, and the controller file controls its execution. This allows for the configuration of some of the functions that are explicitly offered by the tool's sub-modules. It is possible to choose the algorithms that will run during the experimentation for each module, as well as the input and output data for the execution of these algorithms, and the number of rows in the dataset, which is of great help when simplifying the analysis of large datasets and reduce experimentation times. Once this module is configured, `TSxtent` can automatically execute each step configured in the file.

4.2. Data Pre-Processing Module

In this module, the researcher begins to prepare the data that will be used in the experimentation. The module is paramount for temporal sensor data (especially energy data) since the data are collected from heterogeneous data sources (sensors, databases, etc.) [38,39].

Generally, the raw data need to be pre-processed to feed the different learning algorithms, so the data require a cleaning and preparation process to improve the algorithms' performance. This module comprises different submodules, which are described below.

4.2.1. Partitioning

Data partition techniques transform the current datasets to generate new ones. These techniques are essential in time series on energy data because they generate new, hidden knowledge that can improve the results offered by learning algorithms.

Numerous works include this sort of technique [40–42]. The data-partitioning module takes care of partitioning, grouping, and extracting the necessary information so the user can experiment, using a given time interval. It helps to study the dataset defined in a specific time interval, which can be far smaller than the original dataset.

Another important feature is the possibility of grouping the dataset fields at different hierarchical levels, generating a new dataset for analysis in the experiment. For example, it is possible to group according to specific dataset field and obtain several subsets of data to analyse, generating new, hidden knowledge.

For this purpose, dynamic tree generation was implemented to create a new dataset following the parent–child–grandchild hierarchy. It should be noted that, deeper into the tree, the execution becomes less efficient. This permits the generation of new, specific datasets and focuses the experimentation on data with a more well-defined feature set. Finally, the module offers the possibility of defining where the generated dataset is stored and where the executions of the experiments are performed. The structure of this tree can be seen in Figure 2.

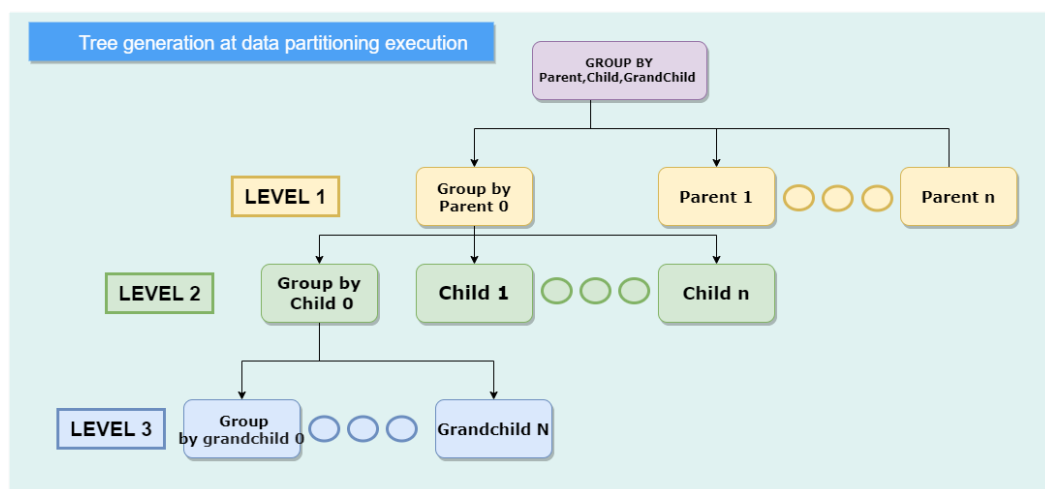


Figure 2. Example of tree generation.

4.2.2. Missing Values

Dealing with incomplete data is a common challenge when working with datasets, especially regarding energy efficiency [43,44]. Incomplete data can arise for various reasons, including faulty sensors, lost data during transmission, or missing data due to human error. This type of data can cause malfunctions in machine learning prediction algorithms, so it is essential to apply pre-processing techniques to deal with these errors and achieve better results. To address this issue, a submodule was created within the data processing module and the focus was placed on removing missing values from the dataset.

This submodule offers the option of treating the missing values by configuring the file and the possibility of indicating the path in which the dataset is located in order to apply the missing value techniques. The currently implemented algorithms are the *interpolation algorithm*, which performs an interpolation between the different rows of the dataset, and another, more aggressive one, which eliminates the rows with missing values.

The *interpolation algorithm* [45] allows for the generation of new data and enhances the quality of the original dataset. It involves estimating the value of a field at intermediate

points between known datapoints. The *row elimination algorithm* is more aggressive and, unlike the previous one, eliminates the row. This algorithm is necessary when the number of missing values within a row of the dataset is too high.

Once either algorithm is executed, the system generates a graphical resource to verify that the missing values have been correctly removed, showing the number of records in each field. The resources are stored in the log module so that the results are saved and can be consulted at any time. It should be highlighted that this process transforms the original dataset, thus improving the quality of the dataset and, therefore, the performance of the learning algorithms.

4.2.3. Outliers

Outliers are another common problem to be faced when solving data science problems [46–48]. These types of values are all too frequent in energy problems because the devices that are responsible for collecting the information and sending it to our source of knowledge are prone to manipulation by external sources from the environment in which they are installed.

There are cases in which the sensors are exposed to temperatures that do not correspond to the actual temperature of the environment that exists at that moment, either due to exposure to a foreign heat source or to a cold source. These actions lead to a series of outliers that are outside of the typical pattern of measurements.

To solve this type of problem, this preprocessing submodule was created, which focuses on the elimination of these outliers from the dataset with which the researcher is going to work. This includes the possibility of selecting the fields to be analyzed in the configuration file, as well as indicating the path on which the dataset to which we are going to apply the outlier preprocessing techniques is located.

In this first layer of our architecture, we included the “z score mean” algorithm. This algorithm detects the values that are within a range defined by us as missing values, and performs an average over its nearest neighbours to modify that value and obtain a prediction of what value should exist at that moment. The system generates a graphical resource to check that the outliers have been correctly eliminated by means of a simple box plot. The resources are stored in the loggers module, so the results are stored and can be queried later in a more effective way.

This process also transforms our dataset, i.e., the application of this algorithm modifies the dataset with which the research continues to more effectively apply our learning algorithms.

4.2.4. Feature Selection

Feature selection is a preprocessing technique that allows for us to obtain valuable information to learn what type of features are most relevant and how the different variables that make up our dataset relate to each other and the problem we are analysing. Like the previous ones, it is one of the most widely used techniques in data processing.

In problems such as those posed by energy efficiency in buildings [49,50], this is advantageous because, in most cases, there are a large number of variables, for which, depending on the problem we are trying to solve, we will need a series of variables or others.

The creation of this module manages to show the relationship between the different variables of the dataset on which we are working. For this to work correctly, it is advisable there are no missing values.

In our tool, it is possible to choose the fields that are to be analysed, as well as the route from which the data to be analysed are obtained. One of the algorithms that is implemented is a correlation algorithm, which generates an image-type resource, with a heat map displaying the correlation between the different selected fields. This algorithm shows the degree, between 0 and 1, of correspondence between one variable and another, with 0 being the lowest degree and 1 the highest degree of correspondence. Another of the implemented algorithms is a proprietary algorithm called FSMeasure that manages to

obtain the measures of mean, standard deviation, entropy, chi-square and dispersion of the data of the different variables of the dataset with which we are working.

This type of analysis helps us in the selection of the input fields of our time series, which we are going to insert into our learning algorithms. The resources generated by this type of algorithm are stored in the loggers module so that they can be consulted later in a convenient way.

4.3. Machine Learning Module

The machine learning paradigm has been widely applied to solve energy efficiency problems [51–54]. This type of paradigm allows for the input of a series of data and the execution of processes that result in an output. Depending on the problem being solved, this output can predict a result with a range of success or classify data into a specific category.

In the case of energy problems, most of these problems are regression prediction problems. In this library, which is based on time series analysis, the input dataset obtained from preprocessing and exploratory analysis is used as the input. A series of algorithms are run, which produce predictions about the building's consumption as output. This helps to optimize consumption and plan a series of energy-saving strategies for the end user.

This machine learning module contains several sub-modules, which include algorithms based on solving time series problems to make predictions using our datasets. As a time-series-focused tool, these algorithms can all solve regression problems.

These algorithms generate a series of resources that help the researcher to draw conclusions about the model generated in our experimentation. The models generated in our experimentation are stored in the logger module. Each of the algorithms integrated in the tool is explained below.

4.3.1. Random Forest Regression

Random Forest [55] averages many models with noise and impartial, reducing the variance. Trees are ideal candidates for bagging, as they can record complex interaction structures in the data.

For the prediction of a new element, a tree leaf is delved through the tree leaves. Then, it is assigned the label at the final node. This process is iteratively railed through all the trees to be assembled in the run and the node that obtains the highest coefficient is reported as the prediction.

The advantages of random forests is that they are highly adaptable to large amounts of data, run efficiently and handle a large number of features, and many energy-efficiency-related jobs are solved using this type of algorithm [56–58].

For these reasons, we implemented this submodule, which runs the random forest regression algorithm included in the sklearn library [59]. The main configuration allowed in this submodule includes the number of Kfolds to be performed within the algorithm, the measurement criteria for each selected Kfold, and the inputs and outputs for the model.

In addition, other parameters are required to further fine-tune the execution, such as the depth of the trees, the estimation and the minimum number of children. The submodule, as in the previous one, displays the scores of the different runs, as well as the mean and standard deviation of these results.

The resources generate a graph showing the tree generated with the various calculations performed by the algorithm, and a report with their respective scores, as well as the mean and standard deviation of these results. This includes the model generated by the algorithm. These resources can also be visualized in the logger module, and the models generated by the execution of these algorithms can be reused.

4.3.2. Decision Tree Regression

A decision tree [27] is a prediction model that has been used in countless fields, especially in the energy field [34,60,61], and is one of the most widely used algorithms in this field.

Given a set of data, a series of logical diagrams are created, similar to rule-based prediction systems, which represent and categorise a series of conditions that occur in succession to solve a problem.

Due to its widespread use in the energy field, the decision was made to implement this submodule, which executes this regression algorithm with the decision trees included in the sklearn library [59]. The main configuration submodule includes the number of Kfolds to be performed in the module, the measurement for each chosen Kfold, and the inputs and outputs for the model.

In addition, other parameters are included to fine-tune further the execution, such as the depth of the generated decision trees, the estimation, and the minimum number of children. In this submodule, as in the previous ones explained above, Random Forest Regression and XGBoost show the scores of the different runs and the mean and standard deviation of these results.

The generated resources consist of a graph showing the tree generated with the different calculations executed by the algorithm and a report with the scores generated in each Kfold, as well as the mean and the standard deviation of these results. These results are recorded in our logger's module and allow for the researcher to reuse them.

4.3.3. XGBoost

XgBoost [26] is quite efficient when running large amounts of data and is flexible regardless of the nature of the data. As a tree running in parallel, it can solve many problems quickly and accurately. Using this type of decision tree in our architecture is essential, as it has been shown in numerous works to work well [45,62–64].

XgBoost is one of the most widely used frameworks in energy efficiency for solving time series regression problems. This submodule runs an XgBoost regressor algorithm. The configuration mainly allowed for covers the number of Kfolds to be performed within the algorithm run, the "target" measure for each chosen Kfold, and the inputs and outputs for the model. The scores of the different runs are shown, as well as the mean and standard deviation of these results.

Once all these calculations have been performed, the generated resources are stored in the logger's module. The generated resources include a graph with the different "scores" of the chosen splits, reports with these "scores", the mean and standard deviation of these results, the parameters used in a run, and the model generated by the algorithm run.

The resulting models stored can be used at any time by the researcher to make predictions with other data, and even to analyse the different results between the runs performed in the experiment.

4.4. Deep Learning

The deep learning paradigm represented a different jump forward in the energy prediction [65,66]. Neural networks underwent a significant evolution, and can be used to solve the different problems that arise during optimisation in the field of energy consumption optimisation. Algorithms that simulate the neural behaviour of the brain, such as convolution networks or recursive neural networks, have been shown to give much better results than decision trees in problems dealing with time series.

For these reasons, it was proposed to create a module involving some of the most efficient neural networks for working with time series [67–69]. Initially, a multilayer perceptron network, a convolutional network and an extended short-term network were chosen. This module was configured through the configuration file, where the algorithms were indicated. Each of the implemented submodules will be explained in the following sections.

4.4.1. Multi-Layer Perceptron

A multilayer perceptron network [70] consists of an input layer, a hidden layer, and an output layer [71]. The most straightforward neural network uses a supervised learning technique for training and a non-linear activation function. This has been used in several

papers in the field of energy efficiency, but there is little work. Therefore, it is interesting to continue working with this type of network and to have our architecture ready to continue studying the behaviour of this algorithm in different problems [72–74].

For this reason, this algorithm was chosen to create a submodule within Deep Learning. A multilayer perceptron (MLP) neural network for univariate time series forecasting models was included. The configuration file allows for us to configure both the inputs and outputs of the model.

An exciting aspect of configuration is the n-steps parameter. This allows for the input/output sequence to be split into multiple patterns, which, being univariate, result in the prediction of an output. In addition, the configuration of the hidden layers and the number of epochs that the neural network executes are included. This neural network is composed of two density layers.

Another interesting aspect is the grouping of the data with the batch-size parameter. We can group these into more or fewer data. This sub-module generates a report with the obtained results and stores the model in the logger's module so that the researcher can make any prediction later and include the configuration of the hidden layers.

4.4.2. Convolutional Neuronal Network

A convolutional neural network [75] consists of an input layer, n-hidden layers and an output layer. In any feed-forward neural network, the intermediate layers are called hidden because the activation function and the final convolution mask their inputs and outputs. In a convolutional neural network, the hidden layers include layers that perform convolutions. Their activation function is usually ReLU. The convolution operation generates a feature map, which contributes to the next layer's input. This is followed by other layers, such as grouping, fully connected, and normalisation layers.

This was rarely used in the energy field [76–78] because it is a network that works very well with images. This is why it has been little tested in the energy domain, which tends to use time series. Thus, by carrying out a series of transformations, it can be adapted for use in time series, and we can study this type of network in problems in different domains. This neural network comprises a one-dimensional convolutional layer, another max-pooling subsampling layer, another flattening layer to reduce the matrix into a flat matrix, and two dense layers.

In the submodule in which we have included the Convolutional Network (CNN) [71], we can configure where the inputs and outputs of our model will be stored in the file. As in the previous submodule, n-steps are configured. We can configure the number of features considered for the model, the number of hidden layers, and the number of other parameters highlighted in this algorithm.

This submodule generates a report with the obtained results, and stores the module so that the researcher can make any prediction at a later date.

4.4.3. Long Short Term Memory

Short-term memory (LSTM) [30] is an artificial neural network in artificial intelligence and deep learning. Unlike standard neural networks, LSTM has feedback connections. This recurrent neural network (RNN) can process images and time series. In the energy field, it is currently one of the most widely used neural networks to date [39,79–82].

The submodule is configured in the file and allows for us to see both the inputs and outputs of the model. As in the previous modules, we can configure the n-steps. We can select a large number of features that are considered when generating the model and the number of hidden layers.

This module generates a report with the obtained results and stores the module so that the researcher can later make any type of prediction that can be consulted in the logger module. This neural network comprises an lstm layer and a dense layer.

4.5. Loggers

It is worth highlighting the module in which all the loggers of our tool are stored due to the importance given to the interpretability of data in machine learning problems. This type of solution, which is far from the realm of experts, can be understood and comprehended with little data science knowledge. This is one of the motivations for integrating MLflow into TSxtend.

MLFlow is a tool developed by [32], in which all the results produced during the execution of an experiment are recorded. With this tool, it is possible to keep track of all executions, as well as the creation of the different processes that are used to run our different modules. Another important reason that it has been integrated is that it shares the same language in its implementation. MLflow developed the same programming language that we developed in our tool, such as Python, so this integration is easier. As an open-source tool used by data engineers, we can follow the different executions of our work. At the same time, we can take advantage of the knowledge that is generated to adapt the tool to our needs.

Finally, this allows for us to visualize all the files generated by our tool when it finishes executing the experimentation we dictated in the configuration file. This way, all the logger generated by our tool is stored graphically and through a user-friendly interface. This will help us to better understand our results and to make decisions by planning a strategy to develop new experiments.

For installation and start-up, TSxtend automatically installs MLflow on the server on which the software is deployed, with the help of the Python package manager called Conda [83]. Once our library is executed, a web browser will be launched, on which each of the experiments carried out by the user can be accessed.

5. Use Case: Energy Data Analysis

In this section, a use case is presented to demonstrate the feasibility of the proposed tool. For this purpose, experimentation with real data is carried out. The workflow defined for the use case is depicted in Figure 3.

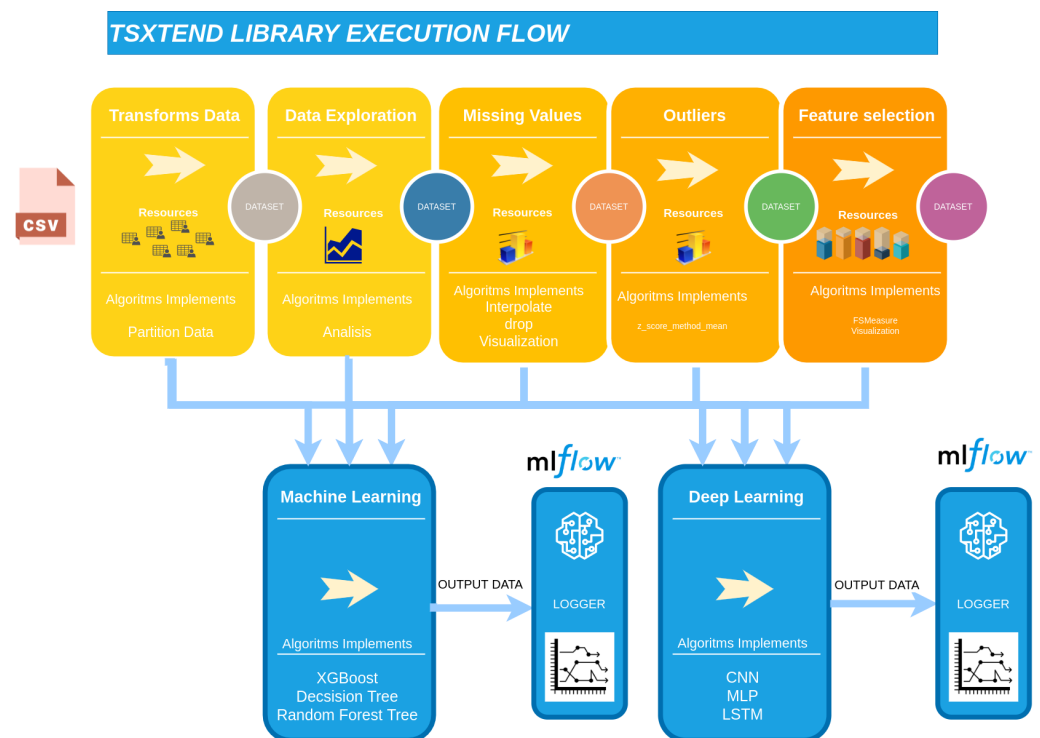


Figure 3. TSxtend execution flow.

5.1. Methodology

As previously outlined in Section 4, the proposed workflow includes stages for data mining, prediction using Machine Learning and deep learning techniques, and the storage and consultation of results on the server launched by the tool, which has been integrated with MLflow.

The data mining component enables the transformation of data through the execution of algorithms, partitioning it into smaller datasets for more efficient experimentation. The following stage, as shown in Figure 3, is data exploration which allows for the analysis of data through various visual representations such as tables, text files, and graphs.

Another stage of the methodology is the removal of missing values, followed by the detection and removal of values that fall outside the range of the data sampling and can be considered noise. The final stage of the data mining phase is feature selection, which allows for the identification of correlations between variables and the determination of the most important features within the dataset. These steps can be performed independently and in any order, allowing for any stage to be executed and prediction algorithms to be used at any point.

The final stages of the methodology allow for the implementation of machine learning and deep learning algorithms to make predictions on the specified output variable (as seen in the Appendices A.6 and A.7). The following sections demonstrate the application of this methodology within a specific use case in the energy domain.

5.2. Dataset Description

To conduct the experimentation, the dataset was obtained from the Kaggle Predictive Features Competition called ASHRAE - Great Energy Predictor III, which focuses on energy consumption analysis. To ease the experimentation, the original dataset was transformed and consolidated into a single training file. Additionally, the fields that were deemed most relevant for the analysis were selected. Table 2 showcases the selected fields. The dataset includes three years of hourly readings from counters of over a thousand buildings in different parts of the world, with 1,048,000 records. The data were taken from the contest posed on the Kaggle platform. For more details regarding the dataset, refer to [12].

Table 2. Dataset description.

Name Field	Descriptions
building_id	Foreign key for the building metadata.
meter	The meter ID code.
timestamp	When the measurement was taken
meter_reading	The target variable. Energy consumption in kWh (or equivalent).
site_id	Foreign key for the weather files.
air_temperature	Degrees Celsius
cloud_coverage	Portion of the sky covered in clouds, in oktas
dew_temperature	Degrees Celsius
precip_depth_1_hr	Millimeters
sea_level_pressure	Millibar/hectopascals
wind_direction	Compass direction (0–360)
wind_speed	Meters per second

5.3. Data Partitioning

As explained in Section 4.2.1, this phase aims to show how the tool works with energy data when pooling data from an original dataset. In this way, the disparate energy data are further disaggregated in the first capture phase. In the following sections, the use of the module with our tool is demonstrated.

In our use case, the data were grouped by *site_id* and *meter* to associate electricity consumption with specific areas. This process allows for the generation of new, hidden insights from the dataset. A specific period was selected, in this case, all of 2016, so that the algorithm could partition the energy data accordingly. The grouping and partitioning of energy data was carried out in an attempt to extract less biased information. An example of this configuration can be found in Appendix A.1. Once these parameters were selected, our tool generated more than thirty files containing energy consumption data for all the buildings that comprise ASHRAE. A diagram illustrating this process can be found in Figure 4.

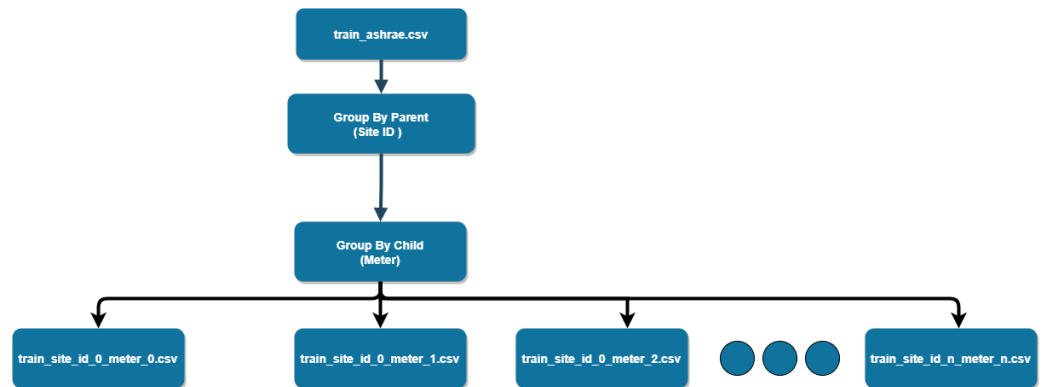


Figure 4. Generating CSVs through experimentation.

5.4. Data Exploration

The researcher typically examines the data using various statistical and visual techniques during the exploratory data analysis. TSxtend allows for this process to be defined through the *configuration file*. Appendix A.2 showcases an example of the *configuration file*. For the current use-case, we selected the fields *meter reading*, *air temperature*, *dew point temperature*, *precipitation depth of one h*, *pressure at sea level*, and *wind speed* because these variables are the most relevant for the use-case (energy consumption analysis). The exploratory data analysis generates a series of charts that can help to understand the data distribution, identify any outliers or anomalies, and identify trends and patterns.

In Figures 5 and 6, it can also be observed that *wind speed* is the variable that has the least correlation with the other ones. Finally, Figure 7 depicts a strong inverse correlation between the variables *air temperature* and *dew point temperature* and the buildings' energy consumption.

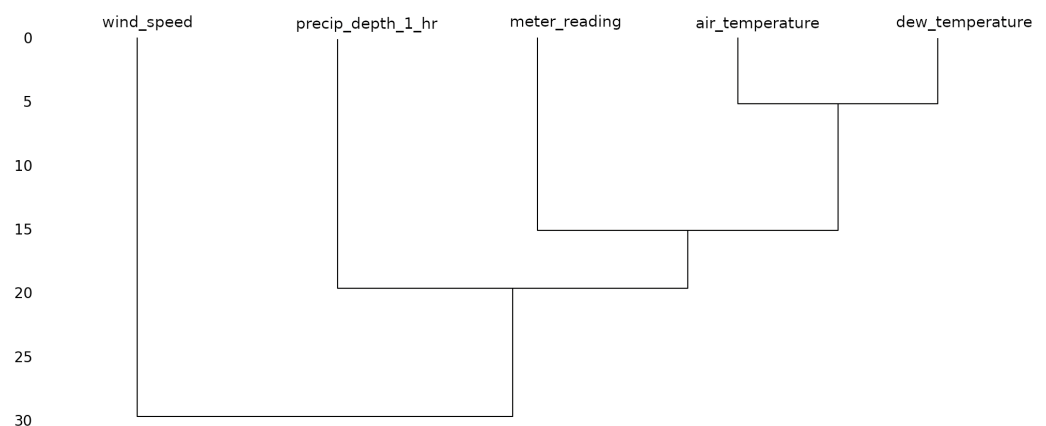


Figure 5. Dendrogram generated using selected variables.

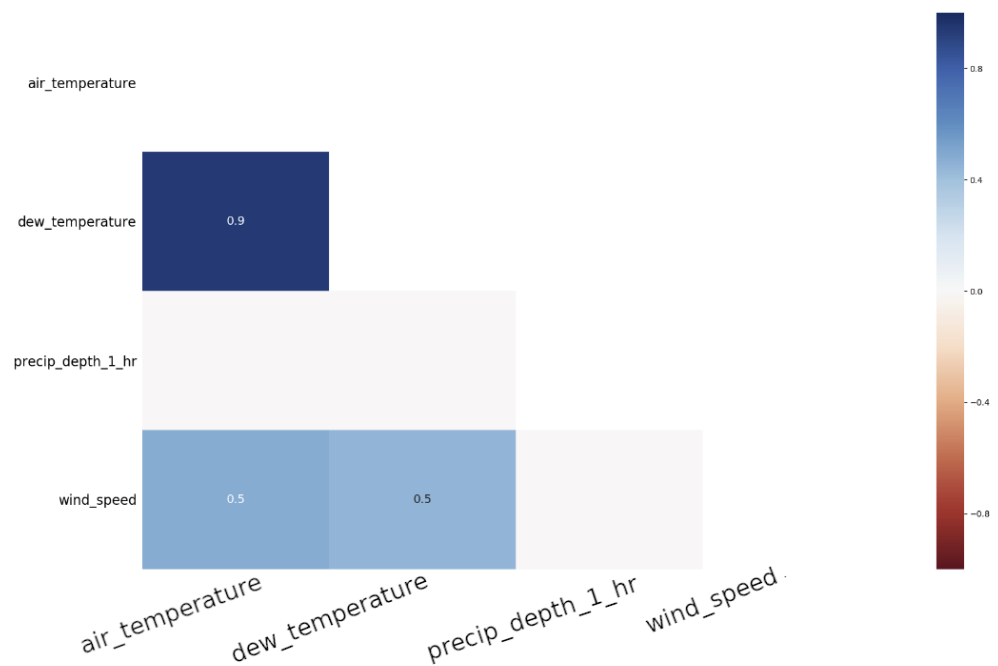


Figure 6. Correlation for selected variables.

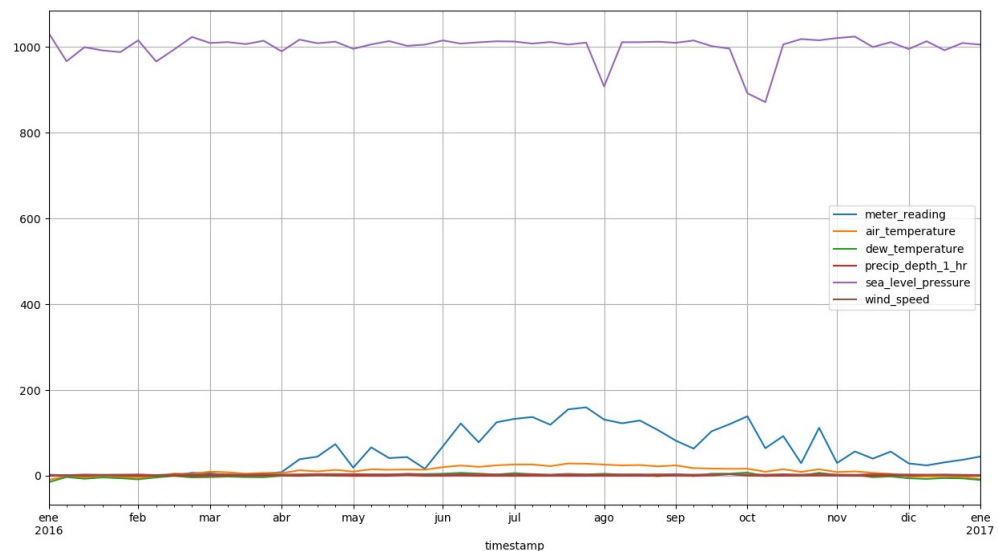


Figure 7. Evolution of time series over time for selected variables.

5.5. Removing Missing Values

During this experimentation phase, missing values were removed from the dataset. The removal of missing values is crucial to obtaining accurate, reliable, and valid results from any data analysis. The preceding section demonstrated the process of visualizing the data using TSxtend. From the charts, it can be concluded that missing values are present in all fields except for the *meter-reading field* variable.

Once the fields with missing values are detected, algorithms are applied to remove them. Only the *interpolation* and *omission* algorithms are implemented. For a more detailed explanation of the above algorithms, our repository can be consulted [37].

In the use-case developed in this paper, the *interpolation* algorithm based on its nearest neighbours was used. Missing values for fields such as *wind speed*, *precip depth*, *dew temperature* and *air temperature* were removed. It is important to note that this type of algorithm transforms the original dataset. The final results are shown in Figure 8, verifying that all the missing values were removed. Appendix A.3 illustrates how to remove missing values through the *alg_missing* parameter.

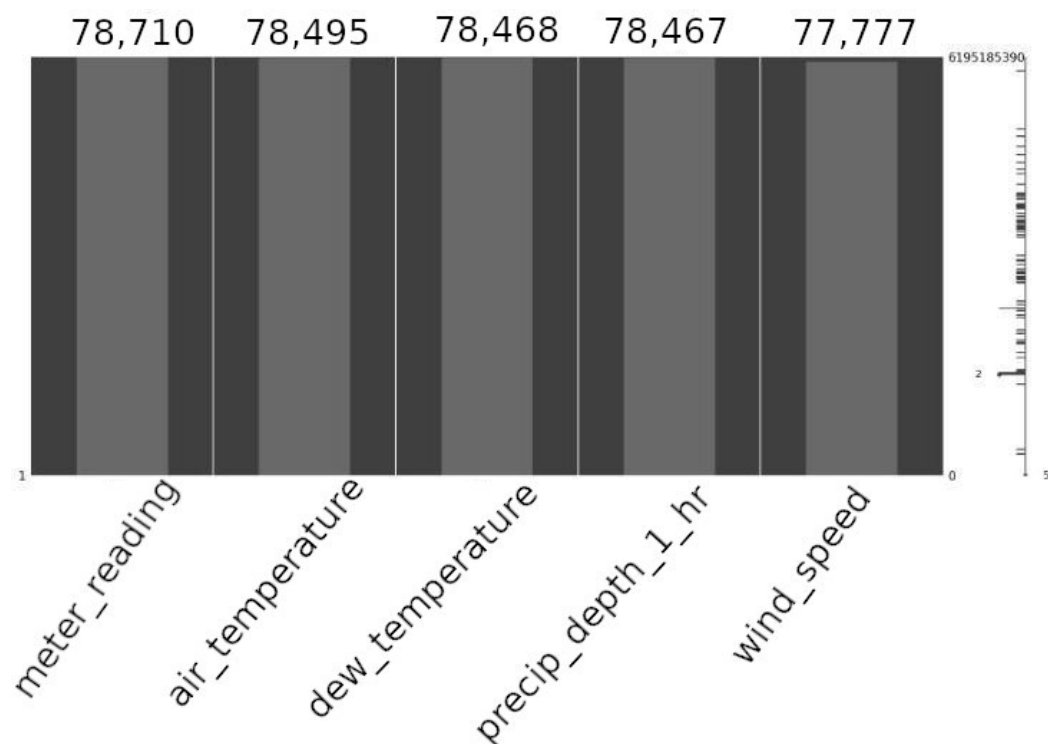


Figure 8. Removing missing values.

5.6. Noise Reduction

The next step comprises noise reduction. From the previous exploratory data analysis, we know there are outliers in our dataset, which is reasonable when dealing with temporal sensor data about energy. To achieve this, the *z-score-mean* algorithm will be applied (see Appendix A.4). This method identifies and removes outliers from a dataset by calculating the *z-score* for each datapoint and comparing it to a threshold. If the *z-score* of a datapoint is greater than the threshold, it is considered an outlier and removed from the dataset.

To use the *z-score-mean* algorithm, it is necessary to specify the threshold value, which specifies how many standard deviations a datapoint needs to be from the mean to be considered an outlier. A common threshold value is 3, meaning that a datapoint needs more than 3 standard deviations from the mean to be considered an outlier. However, the appropriate threshold value will depend on the specific characteristics of the data and the goals of the analysis. Figure 9 shows all the outliers for the use-case. It is important to note that the variables with the most variability in their values are *dew temperature*, *air temperature*, and *wind direction*, which are also the most influential variables in the energy dataset.

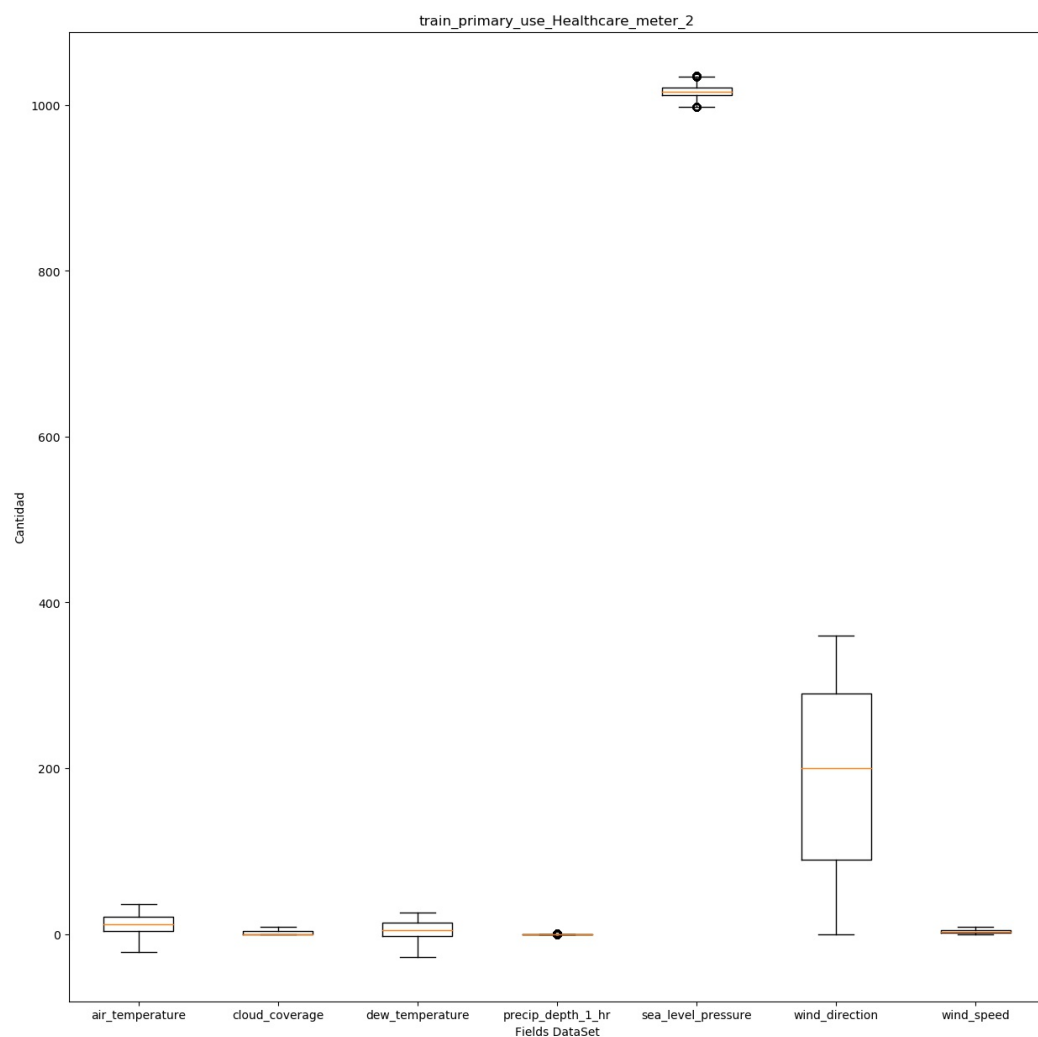


Figure 9. Outliers detection.

5.7. Feature Extraction

This process, one of the most critical in preprocessing, helps us to obtain the most relevant variables from our dataset. Our algorithms are shown in [37].

In this process, we ran a variable correlation algorithm that shows the relationship between our variables. Figures 10 and 11 showcases the correlation between variables *dew temperature* and *air temperature*. In this case, the variables are highly correlated, with a value of 0.9. Wind direction and wind speed have a correlation, but this is not very strong; the value we obtained was 0.5.

By executing the algorithm that measures the characteristics of the different variables in our dataset, the reliability of the different variables can be determined. In our use-case, it can be seen that the most reliable variables are *dew temperature*, *air temperature* and *wind speed*. Therefore, it can be verified that these are the most crucial variables for predicting the energy consumption of our buildings.

In this manner, the effectiveness of the algorithms was verified with the energy dataset and assisted us, together with the exploratory analysis, in conducting a comprehensive study of the different variables that comprise our experimentation.

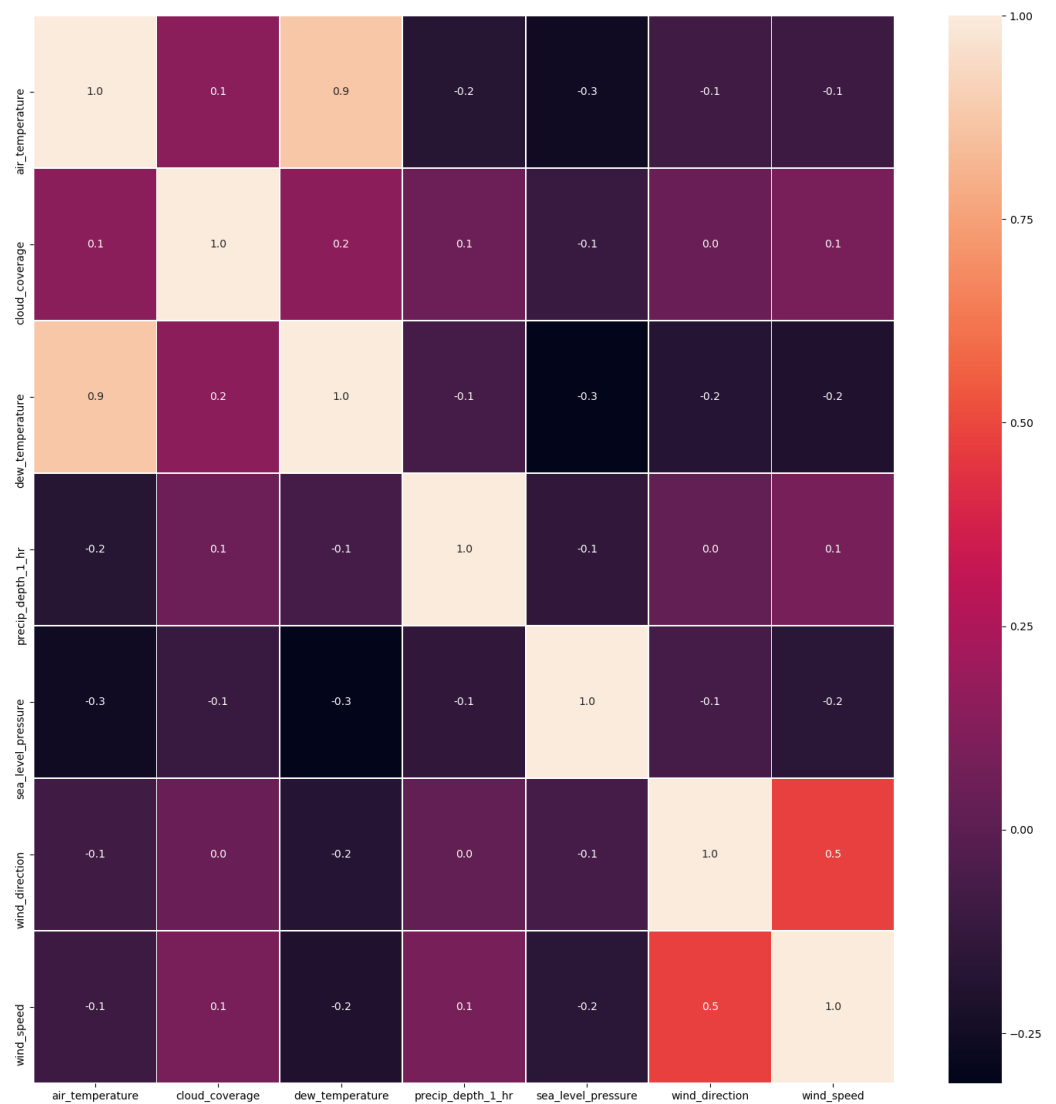


Figure 10. Correlation between selected variables.

	mean	Std.Dev	Var	entropy	chi	dispersion
meter_reading	226.25	376.54	141,788.92	6.44	5.67*10 ⁸	89.44
air_temperature	22.86	6.01	36.11	3.76	1.43*10 ⁶	1.12
dew_temperature	16.85	6.48	42.01	3.64	2.25*10 ⁶	1.81
precip_depth_1_hr	1.38	12.97	168.24	0.52	1.10*10 ⁸	2.29
sea_level_pressure	1,017.95	4.03	16.24	5.15	1.44*10 ⁴	1.03
wind_speed	3.37	2.15	4.64	2.59	1.24*10 ⁶	1.42

Figure 11. Statistical measures for selected variables.

5.8. Employing Machine Learning Techniques to Generate Predictions

One of the key points towards which energy-based information processing is directed is the execution of algorithms for predicting energy consumption. For this reason, the decision was made to implement these types of algorithms in TSxtend. The tool can make a prediction by running classical machine learning algorithms to see how previously

preprocessed data behave. In this use-case, the execution of a single algorithm, XGBoost, is demonstrated. The rest of the implementations can be viewed in [37].

This algorithm uses a K-fold number of models with XGBOOST Regressor algorithms. It makes predictions and stores each result in the array called scores to obtain the mean squared error of the energy consumption prediction. An important variable in the configuration is *n_splits* (see Appendix A.6), because it indicates the number of times the received dataset will be divided. Finally, metrics such as *mean squared error*, *standard deviation* and *number of partitions* of our data are computed.

As discussed in Section 4.5, the outputs will be stored in the logger module integrated by MLflow. These results are stored as graphs, plain text or files that can be consulted by users at any time, helping the user to interpret the experiment.

Tables 3 and 4 show the scores and statistical measures, respectively, for the Xgboost algorithm, while Figure 12 depicts the performance of the Xgboost algorithm. In our use-case, the three most crucial variables mentioned in previous sections, *dew temperature*, *air temperature*, and *wind speed*, were selected. Based on the selected variables that were used as input, the Xgboost algorithm was executed, and ten splits were made to generate the mean error when making the prediction.

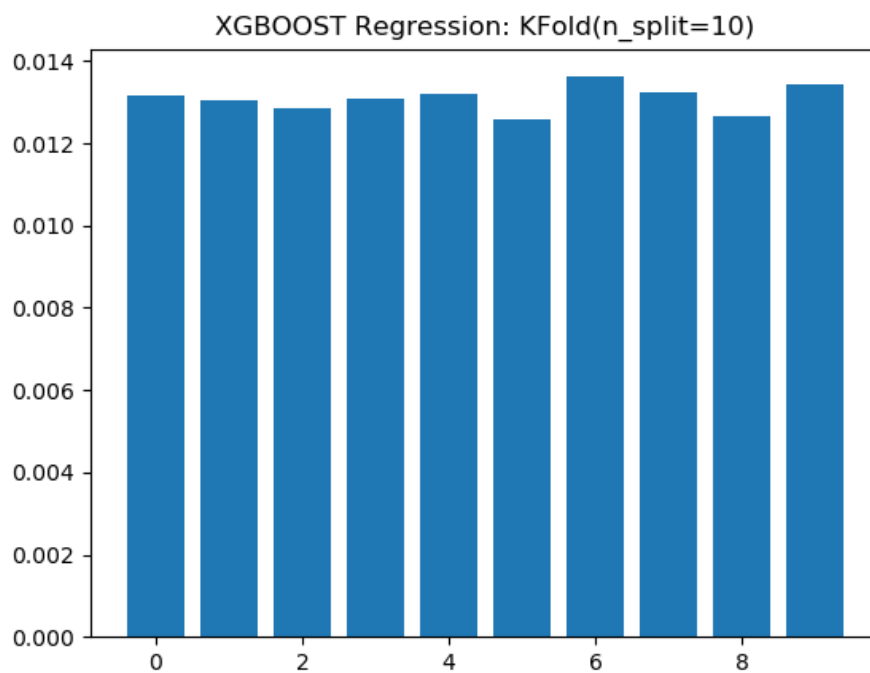


Figure 12. Performance of XGBoost algorithm.

Table 3. Scores for Xgboost algorithm.

Display Scores	Scores
0	0.0132
1	0.0130
2	0.0129
3	0.0131
4	0.0132
5	0.0126
6	0.0136
7	0.0132
8	0.0126
9	0.0134

Table 4. Mean, standard deviation, and number for the Xgboost algorithm.

MAE	STD	Display Scores
0.0131	0.001	10

The result shows us a squared error of 0.0131, which is quite low, with a standard deviation of 0.001, indicating that the execution with the passed input data is correct. This suggests that the input variables are well-correlated with the target variable (energy consumption), and that the algorithm is effectively learning the relationship between them. A test could be run to verify the results, but the goal of this work was to run the tool and see that it works correctly.

5.9. Using Deep Learning Algorithms to Make Predictions

As we mentioned in the previous module, the trend in the study of energy data is predicting consumption using machine learning algorithms.

With this, the tool can perform a prediction executed on classical neural network algorithms to see how the previously preprocessed data behave. In this use-case, the execution of a single algorithm, the long-term memory (LSTM) algorithm, is demonstrated. The rest of the implementations can be seen in [37].

Next, the data were sequenced, and the intervals were divided according to the n_steps (see Appendix A.7). Once the data were obtained, we created a model using, in this case, an LSTM network. The inputs were the same as those used in the XGboost algorithm executed in the previous section. The variables *dew temperature*, *air temperature*, and *wind speed* were used to predict the building's energy consumption.

Finally, we obtained the metrics we measured in our model (rmse, mae, mse). The LSTM model is stored in the system, and the evolution of the model over the execution of different periods is shown a graph. The results of the experimentation are presented in Table 5 and Figure 13.

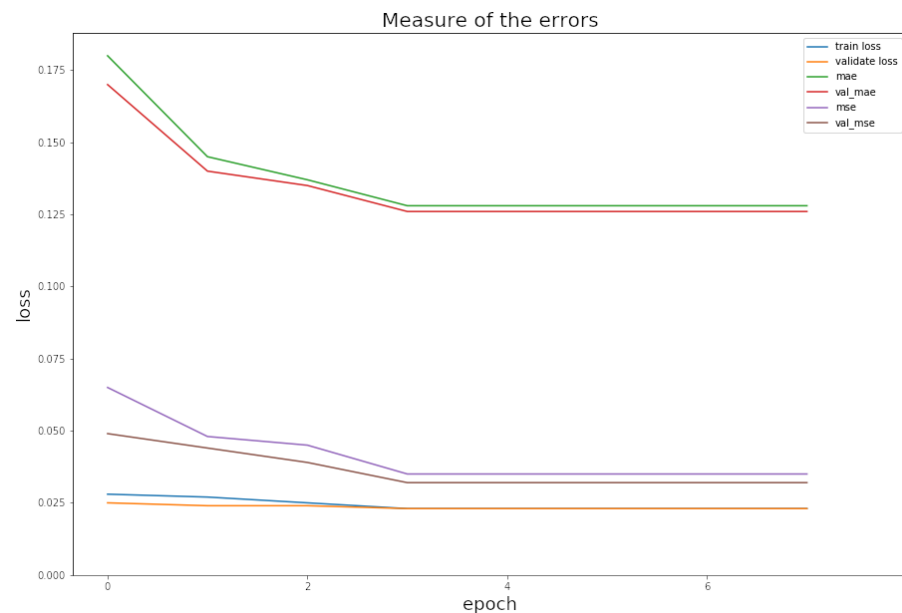
**Figure 13.** Graphical representation of the results of LSTM model execution.

Table 5. Evaluating the performance of an LSTM model in our case study.

Measure Error	Display Scores
rmse	0.1292
mse	0.1055
mae	0.0167

These results can be used by users, as mentioned in Section 4.5, the outputs generated by the different executions within our system will be stored in the loggers module, where, with the help of MLflow, they can be consulted at any time by the user. This will allow for them to perform an analysis of their experimentation and create reports, and also help when making a decision about the experimentation.

In this case, the RMSE, MSE, and MAE values were all relatively low, which suggests that the model is making relatively accurate predictions. The RMSE value of 0.1292 indicates that, on average, the model's predictions are off by about 0.1292 units. The MSE value of 0.1055 and the MAE value of 0.0167 are both lower than the RMSE value, which further supports the conclusion that the model is making accurate predictions.

6. Conclusions and Future Work

This paper introduces TSxtend, a new software tool designed to assist non-programming users in analysing temporal sensor data. TSxtend simplifies the process of transforming, cleaning, and analysing temporal sensor data through the use of a declarative language for defining and executing workflows. This tool aims to empower users to take control of their data, enabling them to make timely and well-informed decisions during the research process. With TSxtend, non-programming users can quickly analyse their data, allowing for them to focus on developing their research and understanding their results rather than struggling with programming.

The paper's main contribution is the development of the tool itself and the features that make it unique. The tool was implemented using a modular architecture, which allows for the standardisation of different stages of experimentation. TSxtend allows for data transformation, cleaning, and imputation, as well as the execution of prediction algorithms and the visualisation of results at different workflow stages. The tool also allows for the use of different techniques for time series analysis and makes experimentation more accessible for non-programmers. Finally, can obtain the first set of results for the analysed dataset in a fast and agile way. In this sense, this helps end-users to establish timely and proper strategies during the decision-making process. The main difference between the proposed tool and the others that were implemented is the possibility of defining a workflow quickly and readily, without prior programming knowledge.

TSxtend was implemented using an easy-to-use approach. This led to a considerable improvement in the user experience when executing algorithms and obtaining initial results, which could be used to analyse the data. Furthermore, the modular architecture of the tool enables the incorporation of additional techniques. The results obtained using the ASHRAE Great Energy Predictor dataset demonstrate the effectiveness of TSxtend in analysing energy data. This tool is a valuable addition to the field of time series analysis and can facilitate the work of researchers and practitioners in various domains.

This tool constitutes a starting point, opening up the possibility for future works. Next, we will implement and integrate more efficient algorithms to provide end-users with more powerful techniques. To make the user experience friendlier, we will develop a more intuitive user interface. Finally, it would be interesting to consider the problems with a multi-variable, as well as developing a complete tool in a distributed environment.

Author Contributions: R.M.-J.: Conceptualization, Software, Data curation, Validation, Writing Original Draft, Writing, Writing Review and Editing. K.G.-B.: Methodology, Writing Original Draft, Writing Review and Editing, Formal Analysis. J.G.-R.: Conceptualization, Writing Review and Editing, Supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by FEDER/Junta de Andalucía (DEEPSIM project, A-TIC-244-UGR20), Spanish Ministry of Science (SINERGY project, PID2021-125537NA-I00) and the NextGenerationEU funds (IA4TES project, MIA.2021.M04.0008).

Data Availability Statement: Data is publicly available at <https://www.kaggle.com/competitions/ashrae-energy-prediction/data> (accessed on 23 December 2022). See more details in [12].

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

Nomenclature and Formula

REST API
user-friendly
pipeline
end-user
roadmap
serializacion
dictionary
measures of mean
standard deviation
entropy
chi-square
heat map
bagging
fine-tune
RMSE
MSE
MAE

Description

Representational State Transfer (REST) *Application Programming Interface (API)*.
Describe applications, websites, and other digital products that are easy to use and understand.
Series of steps or stages that a set of data go through in order to be processed, analyzed, or otherwise transformed
The people who use the software that was developed by the programmers.
Plan or strategy that outlines the steps to be taken to achieve a specific research goal or set of goals.
Process of converting an object or data structure into a format that can be stored or transmitted over a network.
Data structure that is used to store a collection of key-value pairs.
Describe the average value of a dataset.
Measure of the spread or dispersion of a dataset.
Quantity that represents the amount of disorder or randomness in a system.
Statistical test used to determine whether there is a significant difference between an observed frequency distribution and a theoretical distribution.
Graphical representation of data, where individual values are represented as colors.
Technique used in ensemble learning to improve the stability and accuracy of predictive models.
Technique used in deep learning to adjust the parameters of a pre-trained model on a new dataset.
The square root of the mean of the squared differences between the predicted and actual values.
The mean of the absolute differences between the predicted and actual values.
The mean of the absolute differences between the predicted and actual values.

Appendix A

Appendix A.1. Partition Data

Listing A1: Config/main.yaml.

```
1 etl:      partition-data
2 deepl:   ""
3 mlearn:  ""
4 n_rows:  0.0
5 elements: ""
6 output_dir: Data/train_ashrae
```

Listing A2: Config/partition-data.yaml.

```
1 date_init: 2016-01-01 00:00:00
2 date_end:  2016-12-31 00:00:00
3 fields_include: None
4 path_data: train.csv
5 group_by_parent: site_id, meter
6 output_dir: Data/train_ashrae
```

Listing A3: header function.

```

1 def PartitionDF(date_init, date_end, path_data, n_rows, fields_include,
2                 group_by_parent, output_dir)
3
4     ## Parameters ##
5     date_init:[ (string) Correct Date ] . Transform in datatime date init, in
6     order to get data from dataset.
7     date_end: [ (string) Correct Date ]. Transform in data timedata end, in
8     order to get data from dataset.
9     path_data: [ (string) path ] Path origin dataset.
10    n_rows: [ (int) ] Number rows dataset.
11    fields_include: [ (list string) name field dataset ]. Filter by dataset
12    fields.
13    group_by_parent: [ (list string) name field dataset ]. Group by dataset
14    levels.
15    output_dir: [ ( string ) name directory ]. Ouput directory, to save data.

```

*Appendix A.2. Exploratory Analysis***Listing A4:** Config/main.yaml.

```

1 etl:      exploratory-analysis
2 deepl:   ""
3 mlearn:  ""
4 n_rows:  0.0
5 elements: ""
6 output_dir: Data/train_ashrae

```

Listing A5: Config/exploratory.yaml.

```

1 field_x:  timestamp
2 field_y:  site_id
3 graph:    line
4 _resample: W
5 measures: meter_reading,air_temperature,dew_temperature,precip_depth_1_hr
6           ,sea_level_pressure,wind_speed
7 input_dir: Data/test_icpe_v2

```

Listing A6: header function.

```

1 def Visualization(n_rows,field_x, field_y, graph, measures, _resample,
2                 input_dir,elements)
3
4     ## Parameters ##
5     n_rows: [ (int) ] Numbers rows DataSet. This params get from Config/main.
6     yaml
7     elements:[ (string) name elements ] Filter by elements. This params get
8     from Config/main.yaml
9     field_x: [ (string) name field] Field X graphs.Usually this timestamp.
10    field_y: [ (string) name field] Field Y graphs.
11    graph: [ (line | missing) ] Line Graphs time series or show missing values
12    graph.
13    measures: [ (list string) measures ] Determinate fields. Example: site_id,
14    meter
15    resample: [ (string) W, M, Y ] Resamples Week(W), Month(M), Y(Year). Only
16    show data graph.
17    input_dir: [ (string) name directory ] Input directory to get data.

```

*Appendix A.3. Elimination of Missing Values***Listing A7: Config/main.yaml.**

```

1 etl:      missing-values
2 deepl:   ""
3 mlearn:  ""
4 n_rows:  0.0
5 elements: ""
6 output_dir: Data/train_ashrae

```

Listing A8: Config/missing-values.yaml.

```

1 fields_include: None
2 input_dir: Data/train_ashrae
3 alg_missing: interpolate

```

Listing A9: header function.

```

1 def missing_values(n_rows, fields_include, input_dir, elements, alg_missing)
2
3
4 ## Parameters ##
5 n_rows: [ (int) ] Numbers rows DataSet. This params get from Config/main.
      yaml
6 elements: [ (string) name elements ] Filter by elements. This params get
      from main.yaml
7 field_include: [ (list string) name field DataSet ] Filter by DataSet
      fields.
8 input_dir: [ (string) name directory ] Input directory to get data.
9 alg_missing: [ (string) name algorithms] Name Algorithms missing values.
      [interpolate, drop]

```

*Appendix A.4. Elimination of Noise***Listing A10: Config/main.yaml.**

```

1 etl:      outliers
2 deepl:   ""
3 mlearn:  ""
4 n_rows:  0.0
5 elements: ""
6 output_dir: Data/train_ashrae

```

Listing A11: Config/outliers.

```

1 fields_include: None
2 input_dir: Data/train_ashrae
3 alg_outliers: z-score-mean

```

Listing A12: header function.

```

1 def outliers(input_dir, n_rows, q1, q3, fields_include, alg_outliers):
2
3 ## Parameters ##
4 n_rows: [int] number of rows to be extracted, 0 extracts all. This params
      get from main.yaml
5 field_include: [ (list string) name field DataSet ] Filter by DataSet
      fields.
6 input_dir: [ (string) name directory ] Input directory to get data.
7 alg_outliers: [ (string) name algorithms] Name Algorithms outliers. [
      z_score_method_mean]
8 q1: [ (int) ] \% outliers remove.
9 q3: [ (int) ] \% outliers remove.

```


*Appendix A.5. Feature Extraction***Listing A13:** Config/main.yaml.

```

1 etl:      feature-selection
2 deepl:   ""
3 mlearn:  ""
4 n_rows:  0.0
5 elements: ""
6 output_dir: Data/train_ashrae

```

Listing A14: Config/feature-selection.yaml.

```

1 fields_include: meter_reading, air_temperature, dew_temperature,
   precip_depth_1_hr, sea_level_pressure, wind_speed
2 input_dir: Data/train_ashrae
3 alg_fs: FSMeasures

```

Listing A15: header function.

```

1 def feature_selection( n_rows,fields_include,input_dir, elements,alg_fs)
2
3 ## Parameters ##
4 n_rows: [ (int) ] Numbers rows DataSet. This params get from Config/main.
   yaml
5 elements:[ (string) name elements ] Filter by elements. This params get
   from main.yaml
6 field_include: [ (list string) name field DataSet ] Filter by DataSet
   fields.
7 input_dir: [ (string) name directory ] Input directory to get data.
8 alg_fs: [ (string) name algorithms] Name Algorithms feature selection. [
   FSMeasure, correlation]

```

*Appendix A.6. Prediction by Running Machine Learning Algorithms***Listing A16:** Config/main.yaml.

```

1 etl:      ""
2 deepl:   ""
3 mlearn:  xgb
4 n_rows:  0.0
5 elements: ""
6 output_dir: Data/train_ashrae

```

Listing A17: Config/xgb.yaml.

```

1 model_input: air_temperature, cloud_coverage, dew_temperature,
   precip_depth_1_hr, sea_level_pressure, meter_reading
2 model_output: meter_reading
3 n_splits: 5
4 objective: reg:squarederror
5 input_dir: Data/train_ashrae
6

```

Listing A18: header function.

```

1 def xgboost(file_analysis,artifact_uri,experiment_id, run_id, input_dir,
      n_rows,model_input,model_output,n_splits, objective )
2
3 ## Parameters ##
4
5 file_analysis: File analyse. This param is generate from main.py
6 artifact_uri: URL artifact mlflow. This param is generate from main.py
7 experiment_id: Experiment id mlflow. This params is generate from main.py.
8 run_id: Run id mlflow. This param is generate from main.py
9 input_dir: [ (string) name\_directory ] Directory get Data.
10 n_rows: [ (int) ] Numbers rows DataSet. This params get from main.yaml
11 model_input: [ (list string) fields ] Fields input for run algorithms.
12 model_output: [ (list string) fields ] Fields output for run algorithms.
13 n_splits: [ (int) ] Number trees
14 objective: [ (string) ] Params algorithms XGBRegressor

```

*Appendix A.7. Prediction by Running Deep Learning Algorithms***Listing A19:** Config/main.yaml.

```

1 etl:      ""
2 deepl:   lstm
3 mlearn:  ""
4 n_rows:  0.0
5 elements: ""
6 output_dir: Data/train_ashrae

```

Listing A20: Config/lstm.yaml.

```

1 model_input: air_temperature, cloud_coverage, dew_temperature,
      precip_depth_1_hr,sea_level_pressure, meter_reading
2 model_output:      meter_reading
3 input_dir:         Data/train_ashrae
4 n_steps:           3
5 n_features:        3
6 conv_filters:      64
7 conv_kernel_size:  2
8 pool_size:         2
9 hidden_units:      50
10 epochs:            10
11 batch_size:        72
12 verbose:           1

```

Listing A21: header function.

```

1 def lstm(file_analysis,artifact_uri,experiment_id, run_id, input_dir,
      model_input,model_output,n_rows,n_steps,epochs,hidden_units,batch_size,
      verbose)
2
3 ## Parameters ##
4
5 file_analysis: File analyse. This param is generate from main.py
6 artifact_uri: URL artifact mlflow. This param is generate from main.py
7 experiment_id: Experiment id mlflow. This params is generate
8 from main.py.
9 run_id: Run id mlflow. This param is generate from main.py
10 input_dir: [ (string) name\_directory ] Directory get Data.
11 n_rows: [ (int) ] Numbers rows DataSet. This params get from main.yaml
12 model_input: [ (list string) fields ] Fields input for run algorithms.
13 model_output: [ (list string) fields ] Fields output for run algorithms.
14 n_splits: [ (int) ] Number trees
15 n_steps: [ (string) ] Params Split Sequences DataSet.
16 epochs: [ (string) ] Epochs Neuronal Network.
17 hidden_units: [ (string) ] Hidden Neuronals.
18 batch_size: [ (string) ] Batch Size every DataSet.
19 verbose: [ (string) ] Verbose algorithms.

```

References

1. Hang, L.; Kim, D.H. Design and implementation of an integrated iot blockchain platform for sensing data integrity. *Sensors* **2019**, *19*, 2228. [CrossRef] [PubMed]
2. Tushar, W.; Wijerathne, N.; Li, W.T.; Yuen, C.; Poor, H.V.; Saha, T.K.; Wood, K.L. Internet of things for green building management: Disruptive innovations through low-cost sensor technology and artificial intelligence. *IEEE Signal Process. Mag.* **2018**, *35*, 100–110. [CrossRef]
3. Kiran, M.S.; Özceylan, E.; Gündüz, M.; Paksoy, T. Swarm intelligence approaches to estimate electricity energy demand in Turkey. *Knowl.-Based Syst.* **2012**, *36*, 93–103. [CrossRef]
4. Nalcaci, G.; Özmen, A.; Weber, G.W. Long-term load forecasting: Models based on MARS, ANN and LR methods. *Cent. Eur. J. Oper. Res.* **2019**, *27*, 1033–1049. [CrossRef]
5. Salgotra, R.; Gandomi, M.; Gandomi, A.H. Time series analysis and forecast of the COVID-19 pandemic in India using genetic programming. *Chaos Solitons Fractals* **2020**, *138*, 109945. [CrossRef]
6. Tandon, H.; Ranjan, P.; Chakraborty, T.; Suhag, V. Coronavirus (COVID-19): ARIMA-based Time-series Analysis to Forecast near Future and the Effect of School Reopening in India. *J. Health Manag.* **2022**, *24*, 373–388. [CrossRef]
7. Chou, J.S.; Tran, D.S. Forecasting energy consumption time series using machine learning techniques based on usage patterns of residential householders. *Energy* **2018**, *165*, 709–726. [CrossRef]
8. Ruiz, M.D.; Gómez-Romero, J.; Fernandez-Basso, C.; Martin-Bautista, M.J. Big Data Architecture for Building Energy Management Systems. *IEEE Trans. Ind. Inform.* **2022**, *18*, 5738–5747. [CrossRef]
9. Fernandez-Basso, C.; Gutiérrez-Batista, K.; Morcillo-Jiménez, R.; Vila, M.A.; Martin-Bautista, M.J. A fuzzy-based medical system for pattern mining in a distributed environment: Application to diagnostic and co-morbidity. *Appl. Soft Comput.* **2022**, *122*, 108870. [CrossRef]
10. Zhang, W.; Li, H.; Li, Y.; Liu, H.; Chen, Y.; Ding, X. Application of deep learning algorithms in geotechnical engineering: A short critical review. *Artif. Intell. Rev.* **2021**, *54*, 5633–5673. [CrossRef]
11. Zhang, W.; Gu, X.; Tang, L.; Yin, Y.; Liu, D.; Zhang, Y. Application of machine learning, deep learning and optimization algorithms in geoenvironment and geoscience: Comprehensive review and future challenge. *Gondwana Res.* **2022**, *109*, 1–17. [CrossRef]
12. ASHRAE—Great Energy Predictor III. Available online: <https://www.kaggle.com/competitions/ashrae-energy-prediction/data> (accessed on 1 December 2022).
13. ASHRAE. Available online: <https://www.ashrae.org/> (accessed on 15 December 2022).
14. Roldán, J.; Alonso, F.; Aguilera, A.; Maldonado, D.; Lanza, M. Time series statistical analysis: A powerful tool to evaluate the variability of resistive switching memories. *J. Appl. Phys.* **2019**, *125*, 174504. [CrossRef]
15. Shende, M.K.; Feijoo-Lorenzo, A.E.; Bokde, N.D. cleanTS: Automated (AutoML) Tool to Clean Univariate Time Series at Microscales. *Neurocomputing* **2022**, *500*, 155–176. [CrossRef]
16. Rodrigues, J.; Folgado, D.; Belo, D.; Gamboa, H. SSTS: A syntactic tool for pattern search on time series. *Inf. Process. Manag.* **2019**, *56*, 61–76. [CrossRef]
17. Li, M.; Hinnov, L.; Kump, L. Acycle: Time-series analysis software for paleoclimate research and education. *Comput. Geosci.* **2019**, *127*, 12–22. [CrossRef]
18. Antoniadis, I.P.; Brandi, G.; Magafas, L.; Di Matteo, T. The use of scaling properties to detect relevant changes in financial time series: A new visual warning tool. *Phys. A Stat. Mech. Its Appl.* **2021**, *565*, 125561. [CrossRef]
19. Quoilin, S.; Kavvadias, K.; Mercier, A.; Pappone, I.; Zucker, A. Quantifying self-consumption linked to solar home battery systems: Statistical analysis and economic assessment. *Appl. Energy* **2016**, *182*, 58–67. [CrossRef]
20. Christ, M.; Braun, N.; Neuffer, J.; Kempa-Liehr, A.W. Time Series Feature Extraction on basis of Scalable Hypothesis tests (tsfresh—A Python package). *Neurocomputing* **2018**, *307*, 72–77. [CrossRef]
21. Herzen, J.; LÅssig, F.; Piazzetta, S.G.; Neuer, T.; Tafti, L.; Raille, G.; Pottelbergh, T.V.; Pasiaka, M.; Skrodzki, A.; Huguenin, N.; et al. Darts: User-Friendly Modern Machine Learning for Time Series. *J. Mach. Learn. Res.* **2022**, *23*, 1–6.
22. Jiang, X. KATS. Available online: <https://github.com/facebookresearch/Kats> (accessed on 1 December 2022).
23. Barandas, M.; Folgado, D.; Fernandes, L.; Santos, S.; Abreu, M.; Bota, P.; Liu, H.; Schultz, T.; Gamboa, H. TSFEL: Time Series Feature Extraction Library. *SoftwareX* **2020**, *11*, 100456. [CrossRef]
24. Hosseini, R.; Chen, A.; Yang, K.; Patra, S.; Su, Y.; Al Orjany, S.E.; Tang, S.; Ahammad, P. Greykite: Deploying Flexible Forecasting at Scale at LinkedIn. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), Washington, DC, USA, 14–18 August 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 3007–3017. [CrossRef]
25. Winedarksea, P. AutoTS. Available online: <https://github.com/winedarksea/AutoTS> (accessed on 1 December 2022).
26. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), San Francisco, CA, USA, 13–17 August 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 785–794. [CrossRef]
27. Quinlan, J. Induction of Decision Trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
28. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
29. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef]

30. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
31. Bradley, A.P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* **1997**, *30*, 1145–1159. [CrossRef]
32. Zaharia, M.; Chen, A.; Davidson, A.; Ghodsi, A.; Hong, S.A.; Konwinski, A.; Murching, S.; Nykodym, T.; Ogilvie, P.; Parkhe, M.; et al. Accelerating the machine learning lifecycle with MLflow. *IEEE Data Eng. Bull.* **2018**, *41*, 39–45.
33. Bavithra, K.; Siva Kumar, R. Energy Efficient and Reliable K Best Detection Approach with Hybrid Decomposition for WiMAX Applications. *Int. J. Commun. Syst.* **2022**, *35*, e5043. [CrossRef]
34. Tarek, Z.; Shams, M.Y.; Elshewey, A.M.; El-Kenawy, E.S.M.; Ibrahim, A.; Abdelhamid, A.A.; El-Dosuky, M.A. Wind Power Prediction Based on Machine Learning and Deep Learning Models. *Comput. Mater. Contin.* **2023**, *74*, 715–732. [CrossRef]
35. Jeong, S.; Kwon, Y. Energy Efficient Text Spotting Technique for Mobile Edge Computing. In Proceedings of the 2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS), Incheon, Republic of Korea, 13–15 June 2022; pp. 106–109. [CrossRef]
36. Ben-Kiki, O.; Evans, C.; Ingerson, B. Yaml ain't markup language (yaml™) version 1.1. *Work. Draft* **2009**, *5*, 11.
37. Roberto, M.J. TSxtend. 2022. Available online: <https://github.com/robermorjiUgr/Tsxtend> (accessed on 1 December 2022).
38. Ángeles Simarro, M.; García-Mollá, V.M.; Martínez-Zaldívar, F.; Gonzalez, A. Low-complexity soft ML detection for generalized spatial modulation. *Signal Process.* **2022**, *196*, 108509. [CrossRef]
39. Alsharekh, M.F.; Habib, S.; Dewi, D.A.; Albattah, W.; Islam, M.; Albahli, S. Improving the Efficiency of Multistep Short-Term Electricity Load Forecasting via R-CNN with ML-LSTM. *Sensors* **2022**, *22*, 6913. [CrossRef] [PubMed]
40. Tian, T.; Zhao, L.; Wang, X.; Wu, Q.; Yuan, W.; Jin, X. FP-GNN: Adaptive FPGA accelerator for Graph Neural Networks. *Future Gener. Comput. Syst.* **2022**, *136*, 294–310. [CrossRef]
41. Li, X.; Wang, S.; Zhu, G.; Zhou, Z.; Huang, K.; Gong, Y. Data Partition and Rate Control for Learning and Energy Efficient Edge Intelligence. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 9127–9142. [CrossRef]
42. Prasannababu, D.; Amgoth, T. Joint mobile wireless energy transmitter and data collector for rechargeable wireless sensor networks. *Wirel. Netw.* **2022**, *28*, 3563–3576. [CrossRef]
43. Jung, S.; Moon, J.; Park, S.; Rho, S.; Baik, S.W.; Hwang, E. Bagging ensemble of multilayer perceptrons for missing electricity consumption data imputation. *Sensors* **2020**, *20*, 1772. [CrossRef] [PubMed]
44. Pan, J.; Li, C.; Tang, Y.; Li, W.; Li, X. Energy Consumption Prediction of a CNC Machining Process with Incomplete Data. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 987–1000. [CrossRef]
45. Alachiotis, N.; Skrimponis, P.; Pissadakis, M.; Pnevmatikatos, D. Scalable Phylogeny Reconstruction with Disaggregated Near-memory Processing. *Acm Trans. Reconfig. Technol. Syst.* **2022**, *15*, 1–32. [CrossRef]
46. Rahmani, M.K.I.; Khan, F.; Muzaffar, A.W.; Jan, M.A. Internet of Things-Enabled Optimal Data Aggregation Approach for the Intelligent Surveillance Systems. *Mob. Inf. Syst.* **2022**, *2022*, 4681583. [CrossRef]
47. Soga, N.; Sato, S.; Nakahara, H. Energy-efficient ECG signals outlier detection hardware using a sparse robust deep autoencoder. *IEICE Trans. Inf. Syst.* **2021**, *104*, 1121–1129. [CrossRef]
48. Sanyal, S.; Zhang, P. Improving quality of data: IoT data aggregation using device to device communications. *IEEE Access* **2018**, *6*, 67830–67840. [CrossRef]
49. Reddy, K.H.K.; Luhach, A.K.; Kumar, V.V.; Pratihari, S.; Kumar, D.; Roy, D.S. Towards energy efficient Smart city services: A software defined resource management scheme for data centers. *Sustain. Comput. Inform. Syst.* **2022**, *35*, 100776. [CrossRef]
50. Feng, C.; Huang, Y.; Wu, Y.; Zhang, J. Feature-based optimization method integrating sequencing and cutting parameters for minimizing energy consumption of CNC machine tools. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 503–515. [CrossRef]
51. Li, X.; Zhong, K.; Feng, L. Machine learning-based metaheuristic optimization of an integrated biomass gasification cycle for fuel and cooling production. *Fuel* **2023**, *332*, 125969. [CrossRef]
52. Munawar, U.; Wang, Z. Coordinated integration of distributed energy resources in unit commitment. *Int. J. Electr. Power Energy Syst.* **2023**, *145*, 108671. [CrossRef]
53. Chi, L.; Su, H.; Zio, E.; Qadrdan, M.; Zhou, J.; Zhang, L.; Fan, L.; Yang, Z.; Xie, F.; Zuo, L.; et al. A systematic framework for the assessment of the reliability of energy supply in Integrated Energy Systems based on a quasi-steady-state model. *Energy* **2023**, *263*, 125740. [CrossRef]
54. Hai, T.; Hikmat Hama Aziz, K.; Zhou, J.; Dhahad, H.A.; Sharma, K.; Fahad Almojil, S.; Ibrahim Almohana, A.; Fahmi Alali, A.; Ismail Kh, T.; Mehrez, S.; et al. -Neural network-based optimization of hydrogen fuel production energy system with proton exchange electrolyzer supported nanomaterial. *Fuel* **2023**, *332*, 125827. [CrossRef]
55. Cutler, A.; Cutler, D.; Stevens, J. Random Forests. *Mach. Learn.* **2011**, *45*, 157–176. [CrossRef]
56. Zekić-Sušac, M.; Mitrović, S.; Has, A. Machine learning based system for managing energy efficiency of public sector as an approach towards smart cities. *Int. J. Inf. Manag.* **2021**, *58*, 102074. [CrossRef]
57. Pérez-Cutiño, M.; Rodríguez, F.; Pascual, L.; Díaz-Báñez, J. Ornithopter Trajectory Optimization with Neural Networks and Random Forest. *J. Intell. Robot. Syst. Theory Appl.* **2022**, *105*, 17. [CrossRef]
58. Senagi, K.; Jouandeau, N. Parallel construction of Random Forest on GPU. *J. Supercomput.* **2022**, *78*, 10480–10500. [CrossRef]
59. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

60. Aqdas, A.; Amin, R.; Ramzan, S.; Alshamrani, S.S.; Alshehri, A.; El-Kenawy, E.S.M. Detection Collision Flows in SDN Based 5G Using Machine Learning Algorithms. *Comput. Mater. Contin.* **2023**, *74*, 1413–1435. [[CrossRef](#)]
61. Ortiz, D.; Migueis, V.; Leal, V.; Knox-Hayes, J.; Chun, J. Analysis of Renewable Energy Policies through Decision Trees. *Sustainability* **2022**, *14*, 7720. [[CrossRef](#)]
62. Sakshi, T.; Singh, P. Short Term and Long term Building Electricity Consumption Prediction Using Extreme Gradient Boosting. *Recent Adv. Comput. Sci. Commun.* **2022**, *15*, 1082–1095. [[CrossRef](#)]
63. Sauer, J.; Mariani, V.C.; dos Santos Coelho, L.; Ribeiro, M.H.D.M.; Rampazzo, M. Extreme gradient boosting model based on improved Jaya optimizer applied to forecasting energy consumption in residential buildings. *Evol. Syst.* **2022**, *13*, 577–588. [[CrossRef](#)]
64. Nayakwadi, N.; Fatima, R. Automatic handover execution technique using machine learning algorithm for heterogeneous wireless networks. *Int. J. Inf. Technol.* **2021**, *13*, 1431–1439. [[CrossRef](#)]
65. Mariano-Hernández, D.; Hernández-Callejo, L.; Solís, M.; Zorita-Lamadrid, A.; Duque-Pérez, O.; Gonzalez-Morales, L.; García, F.S.; Jaramillo-Duque, A.; Ospino-Castro, A.; Alonso-Gómez, V.; et al. Analysis of the Integration of Drift Detection Methods in Learning Algorithms for Electrical Consumption Forecasting in Smart Buildings. *Sustainability* **2022**, *14*, 5857. [[CrossRef](#)]
66. Himeur, Y.; Alsalemi, A.; Bensaali, F.; Amira, A.; Al-Kababji, A. Recent trends of smart nonintrusive load monitoring in buildings: A review, open challenges, and future directions. *Int. J. Intell. Syst.* **2022**, *37*, 7124–7179. [[CrossRef](#)]
67. Zhou, M.; Shao, S.; Wang, X.; Zhu, Z.; Hu, F. Deep Learning-Based Non-Intrusive Commercial Load Monitoring. *Sensors* **2022**, *22*, 5250. [[CrossRef](#)]
68. Kalapothas, S.; Flamis, G.; Kitsos, P. Efficient Edge-AI Application Deployment for FPGAs. *Information* **2022**, *13*, 279. . [[CrossRef](#)]
69. Bouhamed, O.; Amayri, M.; Bouguila, N. Weakly Supervised Occupancy Prediction Using Training Data Collected via Interactive Learning. *Sensors* **2022**, *22*, 3186. . [[CrossRef](#)]
70. Hagan, M.T.; Menhaj, M.B. Training Feedforward Networks with the Marquardt Algorithm. *IEEE Trans. Neural Netw.* **1994**, *5*, 989–993. [[CrossRef](#)] [[PubMed](#)]
71. BrownLee, J. *Deep Learning for Time Series Forecasting*; Machine Learning Mastery: San Francisco, CA, USA, 2020.
72. Zhou, G.; Moayed, H.; Foong, L.K. Teaching–learning-based metaheuristic scheme for modifying neural computing in appraising energy performance of building. *Eng. Comput.* **2021**, *37*, 3037–3048. [[CrossRef](#)]
73. Irfan, M.; Ramlie, F.; Widiyanto, W.; Lestandy, M.; Faruq, A. Prediction of Residential Building Energy Efficiency Performance using Deep Neural Network. *IAENG Int. J. Comput. Sci.* **2021**, *48*, 731–737.
74. Ibrahim, D.M.; Almhafdy, A.; Al-Shargabi, A.A.; Alghieth, M.; Elragi, A.; Chiclana, F. The use of statistical and machine learning tools to accurately quantify the energy performance of residential buildings. *PeerJ Comput. Sci.* **2022**, *8*, e856. [[CrossRef](#)] [[PubMed](#)]
75. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2012**, *2*, 1097–1105. [[CrossRef](#)]
76. Mozafari, M.; Kheradpisheh, S.R.; Masquelier, T.; Nowzari-Dalini, A.; Ganjtabesh, M. First-spike-based visual categorization using reward-modulated STDP. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 6178–6190. . [[CrossRef](#)] [[PubMed](#)]
77. Wu, Z.; Zhang, H.; Lin, Y.; Li, G.; Wang, M.; Tang, Y. LIAF-Net: Leaky Integrate and Analog Fire Network for Lightweight and Efficient Spatiotemporal Information Processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6249–6262. . [[CrossRef](#)]
78. Chakraborty, I.; Roy, D.; Roy, K. Technology Aware Training in Memristive Neuromorphic Systems for Nonideal Synaptic Crossbars. *IEEE Trans. Emerg. Top. Comput. Intell.* **2018**, *2*, 335–344. . [[CrossRef](#)]
79. Xu, Y.H.; Sun, Q.M.; Zhou, W.; Yu, G. Resource allocation for UAV-aided energy harvesting-powered D2D communications: A reinforcement learning-based scheme. *Ad Hoc Netw.* **2022**, *136*, 102973. [[CrossRef](#)]
80. Jayanthi, E.; Vallikannu, R. Enhancing the performance of asymmetric architectures and workload characterization using LSTM learning algorithm. *Adv. Eng. Softw.* **2022**, *173*, 103266. [[CrossRef](#)]
81. Wu, B.; Wang, Z.; Chen, K.; Yan, C.; Liu, W. GBC: An Energy-Efficient LSTM Accelerator With Gating Units Level Balanced Compression Strategy. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2022**, *69*, 3655–3665. [[CrossRef](#)]
82. Zeng, J.; Ding, D.; Kang, K.; Xie, H.; Yin, Q. Adaptive DRL-Based Virtual Machine Consolidation in Energy-Efficient Cloud Data Center. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 2991–3002. [[CrossRef](#)]
83. Gressling, T. 11 Python standard libraries and Conda. In *Data Science in Chemistry*; De Gruyter: Berlin, Germany, 2020; pp. 45–54.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

4.2. Análisis de series temporales y predicción de consumo de energía en edificios utilizando técnicas de aprendizaje profundo

- Referencia:0924-669X
- Estado: Revisión
- Factor de Impacto:Q2
- Categoría: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE - SCIE
- DOI:
- Revista/Editorial: Applied Intelligence

Deep Learning for Prediction of Energy Consumption: An Applied Use Case in an Office Building

Roberto Morcillo-Jimenez^{1*}, Jesús Mesa, Juan Gómez-Romero, M. Amparo Vila and Maria J. Martin-Bautista

¹Department of Computer Science and Artificial Intelligence, Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación, Universidad de Granada, Granada, 18071 Spain.

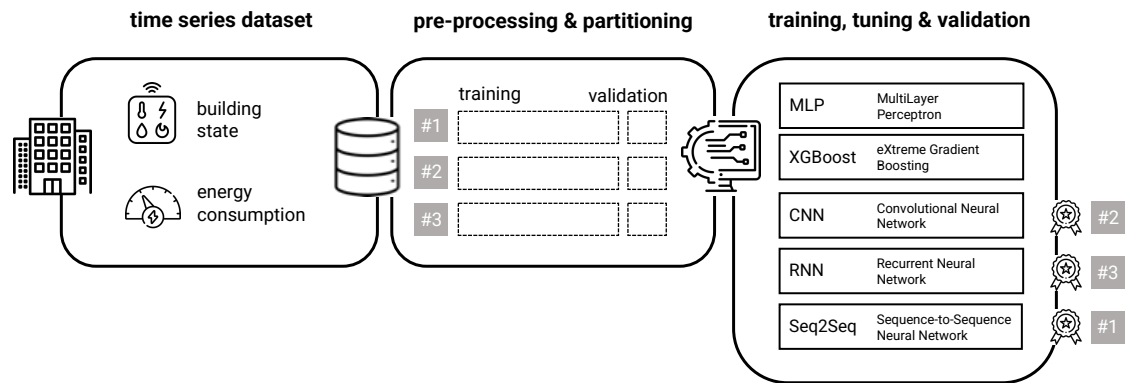
*Corresponding author(s). E-mail(s): robermorji@decsai.ugr.es;
Contributing authors: sulimesa@gmail.com;
jgomez@decsai.ugr.es; avila@decsai.es; mbautis@decsai.ugr.es;

Abstract

Non-residential buildings are responsible for more than a third of global energy consumption. Estimating building energy consumption is the first step towards identifying inefficiencies and optimizing energy management policies. This paper presents a study of Deep Learning techniques for time series analysis applied to building energy prediction with real environments. We collected multisource sensor data from an actual office building under normal operating conditions, pre-processed them, and performed a comprehensive evaluation of the accuracy of feed-forward and recurrent neural networks to predict energy consumption. The results show that memory-based architectures (LSTMs) perform better than stateless ones (MLPs) even without data aggregation (CNNs), although the lack of ample usable data in this type of problem avoids making the most of recent techniques such as sequence-to-sequence (Seq2Seq).

Keywords: buildings, energy consumption forecasting, time series, deep learning, XGBoost, multilayer perceptron, recurrent neural networks, convolutional neural networks, sequence to sequence networks

Graphical Abstract



1 Introduction

The current unstable situation in Europe has led to a decrease in the amount of energy supplied to the European continent, causing an exorbitant increase in prices across the continent. It is therefore necessary to reduce energy consumption in general, to prevent inflation from increasing further and the energy crisis from continuing to grow. Our work focuses on increasing energy efficiency in office buildings, as this is one of the most energy intensive areas due to the number of hours they are in operation. Reducing energy consumption by controlling consumption through algorithmic artificial intelligence techniques is one of the goals of our society in the coming years and our case study.

According to (Pérez-Lombard et al, 2008), residential, office, and industrial buildings account for 20-40% of global energy consumption. In 2010 in the USA, buildings energy consumption accounted for 41% of their total energy consumption, with 75% of this energy coming from fossil fuels (Marino et al, 2016). Meanwhile, in 2012 in the European Union, buildings consumed approximately 40% of the total energy used (Marino et al, 2016). More than two-thirds of the energy consumed by buildings goes to heating systems (37%), water heating (12%), air conditioning (10%) and lighting (9%). Several case studies have shown that the operational phase is the most energy-consuming stage of the buildings life cycle, accounting for 90% in conventional buildings and 50% in low-energy buildings (Schmidt and Åhlund, 2018).

In the current context in Europe, it is important to reduce energy consumption due to the high prices being achieved in the various energy markets. It is not surprising that interest in improving the energy efficiency of buildings has increased. According to (Chau et al, 2015) and (Benedetti, 2015), this interest is driven by three factors: rising energy prices, increasingly restrictive environmental regulations and increased environmental awareness among citizens. All over the world, public policies are being developed to increase this efficiency,

as reflected in the Unsustainable Development Goals and the European Green Deal.

One of the first steps towards improving building energy efficiency is studying how consumption happens and knowing the factors that significantly impact it. In particular, predicting energy consumption from historical data allows building operators and managers to anticipate peaks in energy demand, modify building uses to shift this demand and plan equipment operation appropriately. Likewise, accurate consumption prediction allows assessing the improvement in building performance when making improvements or implementing new energy policies by comparing the actual consumption with the estimated consumption (or baseline). Data Science has emerged as an effective tool to address these objectives ([Molina-Solana et al, 2017](#)).

In the literature, there can be found numerous proposals for estimating energy consumption in buildings throughout time series prediction. Traditionally, these approaches were based on numerical regression or moving average models, which have limitations regarding multivariate series and series with changing trends. In contrast, modern machine learning methods based on neural networks have shown more effective in those scenarios ([Torres, 2021](#)). However, the application of these techniques is often hampered by the noisy and incomplete nature of building energy data in real environments ([Chen et al, 2017](#)), which results in a gap between theoretical and practical works.

The rationale behind this work is to perform a comprehensive evaluation of neural network methods in a real-world scenario and to draw conclusions for practical application in similar contexts. More specifically, this paper studies the accuracy of several methods for heating consumption prediction, namely XGBoost, MLPs, RNNs, CNNs and Seq2Seq. The need and impact of pre-processing, which is applied to remove noisy and missing values and for data reduction, is also discussed.

The dataset was collected in the ICPE office building located in Bucharest, one of the pilot buildings considered in the Energy IN TIME¹. Our starting hypothesis is that modern neural network techniques improve the performance of other approaches, and within them, memory-based architectures (RNNs, Seq2Seq) are superior. The results show that the hypothesis holds, despite the risk of overfitting these techniques when applied to not very large datasets.

The remainder of this paper is structured as follows. We first provide a review of related works on prediction of building energy consumption (Section 2). Next, we describe the data used in the study (Section 3), the methodology (Section 4), and the experimentation (Section 5). At the end of the paper conclusions and directions for future research work (Section 6) will be exposed.

¹Energy IN TIME was an European project running in 2013-2017. The aim of the project was to implement a model-predictive control system to improve the energy efficiency of non-residential buildings ([Gómez-Romero et al, 2019](#)).

2 Related Work

Energy consumption forecasting models can be differentiated into categories based on their respective energy end-uses, such as cooling, heating, space heating, primary, natural gas, electricity, and steam load consumption (Wang and Srinivasan, 2016). Regarding the application of the models, (Tien et al, 2022) identified two primary categories: (1) model-based control, demand response, and optimization of energy consumption in buildings; and (2) design and modernization of building parameters, including energy planning and assessing the impact of buildings on climate change.

Also in (Tien et al, 2022), numerous factors impacting energy consumption were also identified, mainly the number and type of buildings under consideration. The temporal horizon of the prediction and the resolution of the sensor data are also relevant. Remarkably, natural time-based groupings (i.e., hours, days, months, and years) were proved superior in (Wang and Srinivasan, 2016) to the more generic short-, medium-, and long-term ranking schemes proposed in (Ahmad and Chen, 2018).

Many data-driven techniques have proved effective for estimating building energy consumption, ranging from classical statistical regression to modern deep learning architectures. Regarding the former, (Lago et al, 2021) evaluated four models incorporating exogenous inputs, specifically autoregressive moving average models with exogenous inputs (ARMAX). (Chou and Ngo, 2016) proposed a system utilizing a seasonal autoregressive integrated moving average model (SARIMA) and a least squares support vector regression model based on firefly metaheuristic algorithms (MetaFA-LSSVR). The prediction system yielded highly accurate and reliable day-ahead predictions of building energy consumption, with an overall error rate of 1.18%.

Another interesting study is (Mocanu et al, 2016a), which investigated two stochastic models for short-term time series prediction of energy consumption, namely the conditional constrained Boltzmann machine (CRBM) and the factored conditional constrained Boltzmann machine (FCRBM). In the comparison, the results showed that the FCRBM outperformed the artificial neural network, support vector machine, recurrent neural networks and CRBM. The work was extended to include a deep belief network with automated feature extraction for the short-term building energy modeling process (Mocanu et al, 2016b). Other relevant work applying classical learning techniques is (Pachauri and Ahn, 2022), which used a decision tree method (C4.5). In addition to obtaining accurate results, this algorithm was able to identify the factors contributing to building energy use. A sophisticated regression tree algorithm (Chi-Square Automatic Interaction Automatic Detector) was used in (Ezan et al, 2017) to predict short-term heating and cooling load.

(Raza and Khosravi, 2015) provided a comprehensive review of artificial intelligence-based load demand forecasting techniques for smart grids and buildings. The authors explored various machine learning algorithms used in load forecasting, including artificial neural networks, fuzzy logic, genetic algorithms, and support vector regression. (Arpanahi and Javadi, 2018) also

reviewed the applications of artificial neural networks (ANNs) and support vector machines (SVMs) for building electrical energy consumption forecasting. The authors compared the performance of ANNs and SVMs with traditional statistical methods used in energy forecasting. (Seyedzadeh et al, 2018) explored the different ML algorithms used in the field, including artificial neural networks, support vector regression, decision trees, and clustering methods. They also discussed the challenges associated with accurate data acquisition and modelling and the limitations of different ML algorithms. More recently, (Khalil et al, 2022) emphasized that applying machine learning and statistical analysis techniques can lead to significant energy savings and cost reductions. (Zhang et al, 2021) examined the advantages and limitations of different ML algorithms, including artificial neural networks, decision trees, and support vector machines, and explore their application in different building load prediction scenarios. (Rahman et al, 2018) proposed using deep recurrent neural networks (DRNNs) for predicting heating, ventilation, and air conditioning (HVAC) loads in commercial buildings. The authors describe the architecture and training process of the DRNN model, which included a combination of convolutional neural networks (CNNs) and long short-term memory (LSTM) layers.

Finally, other works showed the importance of preprocessing in building energy forecasting, e.g., data cleaning and feature selection. For instance, (Ahmad et al, 2018) reviewed the current development of machine learning (ML) techniques for predicting building energy consumption and discussed the challenges associated with data acquisition, feature selection, and model validation.

Table 1 summarises the related works mentioned in this section.

3 Data

Our data was collected from the ICPE building (Institute of Technologies for Sustainable Development) in Bucharest (Romania). This is a three-building, each one divided into areas. For the experiments, we defined a pilot zone through a transversely cut of the building covering three areas (D1, D2, D5/2) of the three floors (see figures 1,2).

The building is equipped with sensors to measure room temperatures and meters measuring electricity, heating, and water consumption in distinct zones of the building. We focused on predicting heating meter values, given the higher contribution of this subsystem to the total energy consumption. Due to the low temperatures in these countries, buildings are powered by a system called district heating, which is used to prevent pipes from freezing outside, and then energy is consumed inside the building to raise the temperature of the water before it is distributed to the radiators inside.

We considered as prediction targets the three heating meters that respectively cover the area D1, D2 and D5/2 from buildings. However, we found a problem with area D2 data was not available because of errors in sensors, we

6 *Deep Learning for Prediction of Energy Consumption*

Authors	Approach	Methods	Dataset
(Khalil et al, 2022)	Machine Learning Algorithms	Artificial neural networks, decision trees and SVR	Multiple datasets
(Pachauri and Ahn, 2022)	Machine Learning Algorithms	Decision tree	Japanese residential buildings
(Molina-Solana et al, 2017)	Building operation, Fraud detection Applications	MPC	Sanomatato Building
(Wang and Srinivasan, 2016)	Artificial Intelligence, Artificial Neuronal Networks	MLR, ANNs, SVR, ensemble models	Multiple datasets
(Zhang et al, 2021)	Machine Learning	MLR, ANNs, SVR	Multiple datasets
(Arpanahi and Javadi, 2018)	Machine Learning, forecasting methods	ANNs, SVR, GMDH, LSSVM	Multiple datasets
(Raza and Khosravi, 2015)	Short term, load forecasting	ANNs	Multiple datasets
(Lago et al, 2021)	Forecasting	SVR with chaotic gravitational search	Historical electric data from the Northern China
(Chou and Ngo, 2016)	Machine Learning	Time Series Analytics	Multiple datasets
(Mocanu et al, 2016a)	Artificial Neuronal Networks	CRBM, FCRBM	Multiple datasets
(Rahman et al, 2018)	Artificial Neuronal Networks	Multiple algorithms	Multiple datasets
(Seyedzadeh et al, 2018)	Machine Learning	ANNs, SVM, Gaussian-based regressions, clustering	Energy benchmarking
(Ahmad et al, 2018)	Building energy modeling Building optimal control	Agent-Based Model, System Identification	Building Energy model
(Ahmad and Chen, 2018)	Load forecasting, Data mining based approaches	TB, BoostedT, GPR, NN and BaggedT	Office building in Beijing, China
(Ezan et al, 2017)	Energy consumption: Pattern prediction, Time-series technique, Metaheuristic optimization, Machine learning	ARIMA, SARIMA, LSSVR, ML, SVR, ANNs, FA	Smart grid infrastructure
(Mocanu et al, 2016b)	Reinforcement learning, Deep Belief Networks, Machine learning	DBN, SARSA, Q-learning algorithm	Multiple datasets
(Tien et al, 2022)	Machine learning techniques such as supervised, unsupervised, and reinforcement learning	Linear regression, decision trees, random forests, and neural networks, clustering and anomaly detection, Q-learning	Multiple datasets

Table 1: Summary of related works

only selected zone D1 and zone D5/2 of our building. The variables that we select for the D1 zone of our building are H-F123-D1, where all the heating consumption of the D1 zone in the floor 1, 2, and 3 are grouped, and for the D5/2 zone are H-F123-D5/2W and H-F123-D5/2E, where the heating consumption are grouped as the previous one, both for the West zone and for the East zone. We have other external variables that are related to our building, these variables are the outdoor temperature and the building occupancy, which are crucial for our experiments.

We collected data in 2017 before the implementation of the new control system. We focused on heating, and only the cold season (January, February and March) was considered in the experimentation. We re-sampled all variables at 15-minute intervals and calculated the cumulative consumption values. As



Fig. 1: Aerial view

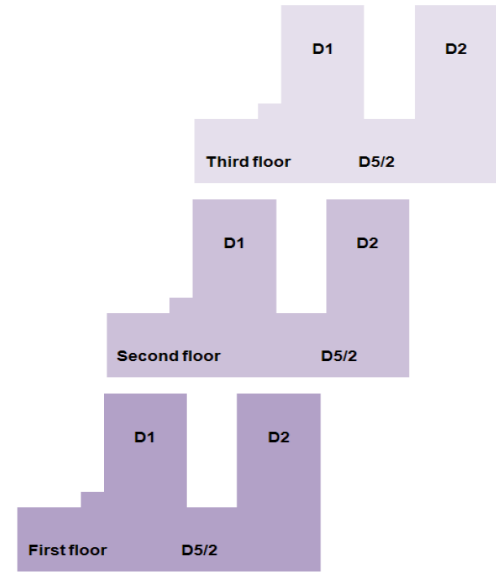


Fig. 2: Pilot zone

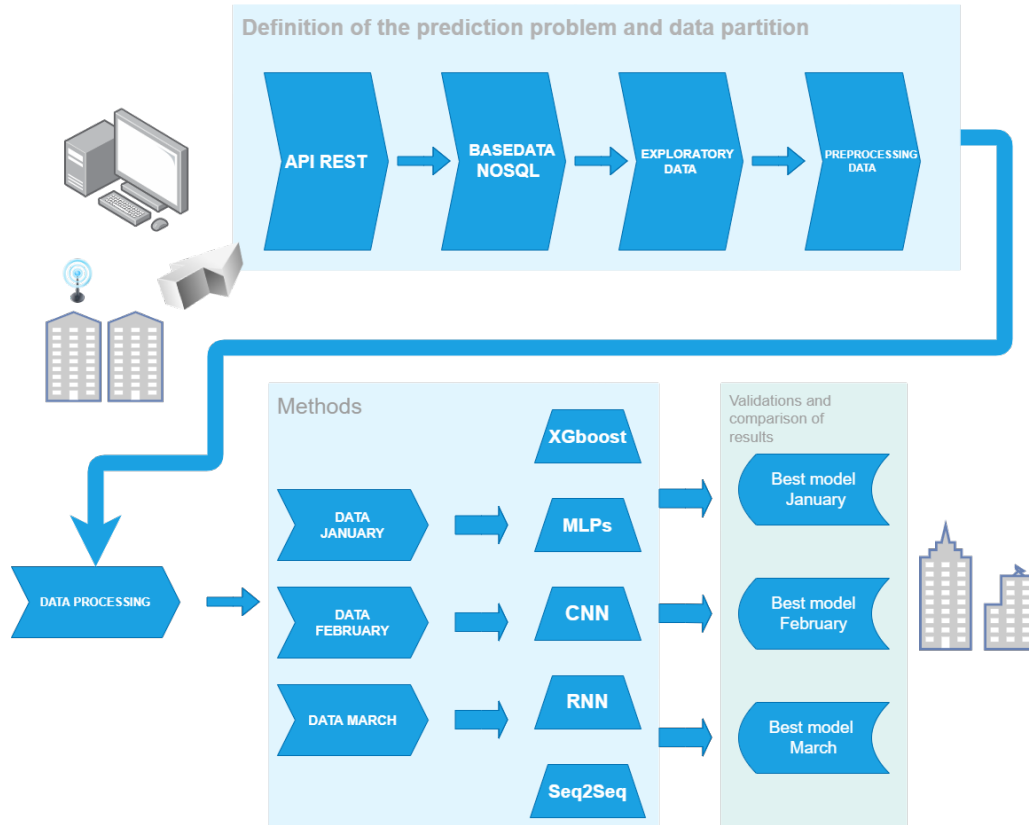
a result, we obtained a data set with 8640 samples. The dataset is not publicly available but an excerpt can be obtained on demand.

4 Methodology

The methodological approach of the experiments followed the workflow of the Data Science process applied to energy data presented in (Molina-Solana et al, 2017). As shown in figure 3, we retrieved the data through an API REST and carried out data preprocessing, including outliers removal, missing value imputation, normalization and feature selection. Afterwards, we continued splitting these data by months to apply the learning algorithms and get the best prediction models for each period. Next, we describe how data is partitioned, as well as the parameters used to train the models and the evaluation metrics.

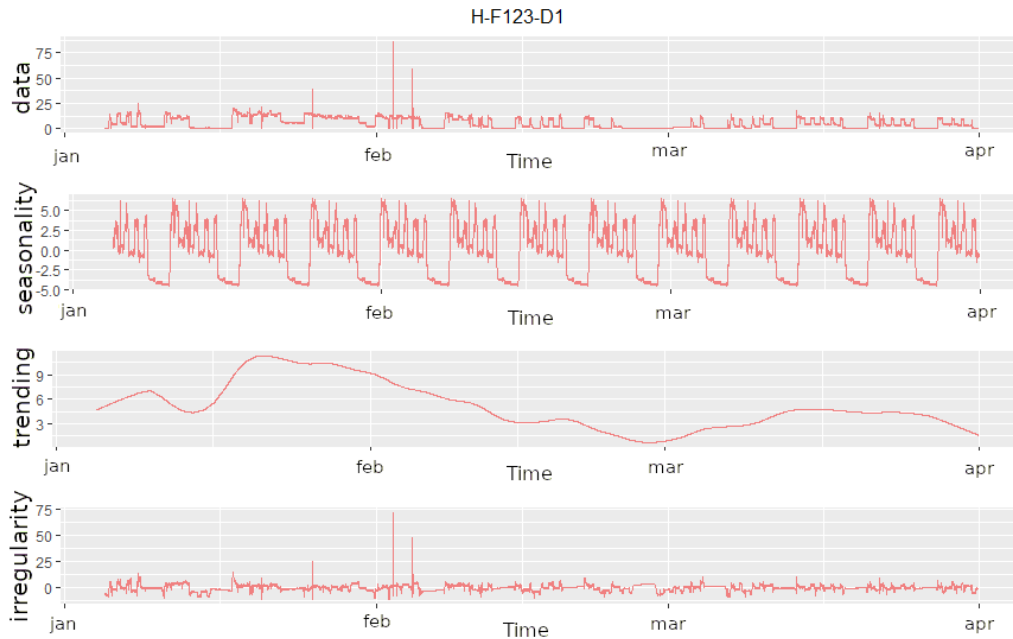
4.1 Definition of the Prediction Problem and Data Partition

The problem we aim to solve is to predict the energy usage of a building for the next 12 hours. We need to partition the original dataset into training and validation samples in chronological order to train our model. Given the different energy consumption patterns at the beginning and the end of the data collection period, we decided to split the full dataset into smaller chunks (namely, subproblems) and then make the training and validation partitions within each one. This implies that we have a different model trained specifically for each subproblem, which must be conveniently selected during the test phase to calculate the predictions depending on the date of the observations.

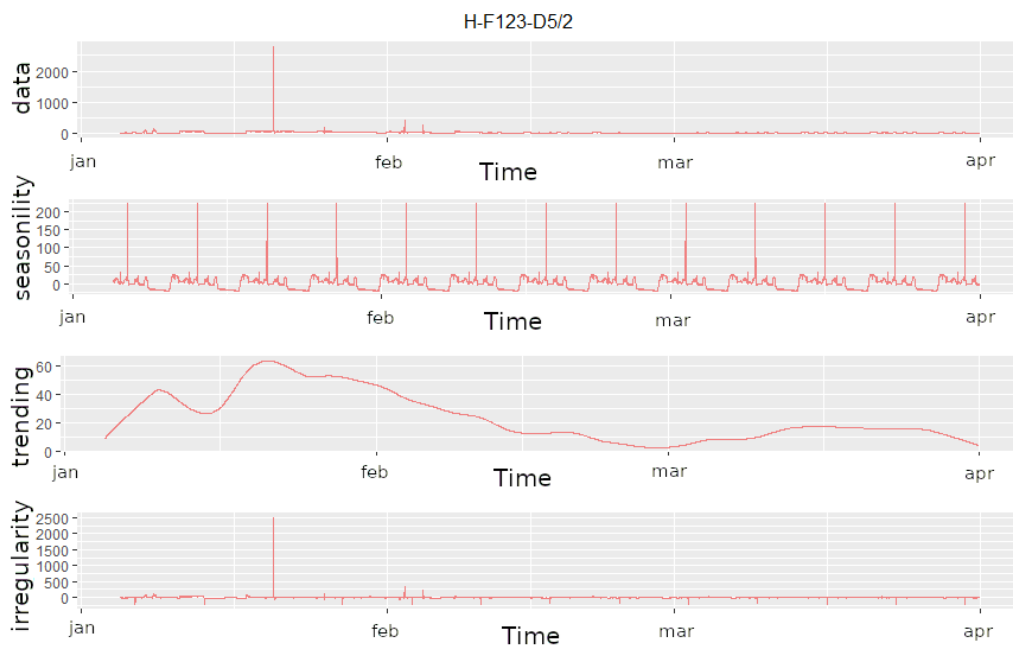
**Fig. 3:** Methodology

To perform the splits, we applied time series decomposition to analyze the evolution of the energy consumption variables and identify potential cut points, focusing on trend change. This study, which is explained below, showed that we could safely split the data in months since the time series have consistent behaviour in each of these periods. Furthermore, a preliminary analysis of the stability of the training, in terms of errors obtained with slightly different splits, showed that the differences were not significant. While this is a rough approximation of a more fine-grained splitting, it has the advantage of facilitating the selection of the model to be applied for prediction. It remains for future work to apply a more sophisticated time series splitting algorithm (Micheletti et al, 2020) and a more comprehensive analysis of the impact of the splitting.

The analysis of the trend change to assess the monthly splitting was based on the decomposition of the time series with STL technique (Seasonal-Trend decomposition using LOESS), which has been previously used for energy demand time series (Phinikarides et al, 2015). Figure 4 depicts the decomposition of the heating consumption variables into three components: trend, i.e., long-term evolution of values; seasonality, i.e., repetitions in fixed periods; and irregularity, i.e., random patterns that remain after removing the other two components. It can be seen that there is a clear pattern of increasingly high consumption in January, followed by a steady decline in consumption in February and a plateau in March. The irregularity component throughout the



(a) H-F123-D1.



(b) H-F123-D5/2.

Fig. 4: Decomposition of the heating consumption variables of plants 1, 2, and 3 in the pilot zone of the ICPE building.

whole series remains quite stable, meaning that the trend and season components capture quite well the changes in the series. Therefore, we proceeded to split the dataset in two parts, training and validation. Specifically, the selected training samples were 70% of each month's observations (e.g., from 2016-01-05 00:15:00 until 2016-01-21 23:45:00 in January), while the remaining values were used for validation.

Algorithms	Parameters	Range
XGBoost	Number of trees	{50,100,50}
	Maximum Depth	{51,100,50}
	Learning Rate	{0.05,0.001,0.1}
CNN	Number of convolutional layers	{1,2}
	Max pooling 1D	{3,5,7}
	Number of filters	{16,32,64,128}
MLP, RNN, Seq2Seq	Number of hidden layers	{1,2}
	Number of dropout layers	{0.15,0.3,0.5}
	Number of neurons	{16,32,64,128}

Table 2: Hyperparameter grids used to tune the learning algorithms.

4.2 Methods

The methods used to build the predictions models are XGBoost (Chen and Guestrin, 2016) and a selection of neural networks including CNNs (Chauhan et al, 2018), RNNs (Sherstinsky, 2020), MLPs (Taud and Mas, 2018) and Seq2Seq (Gong, 2019). The results with Seq2Seq suggested that other techniques, such as transformers, would not be very useful given the relatively small size of each partition, in line with the recent literature (Zeng et al, 2023).

Table 2 summarises the configuration of the models and the hyperparameters probed with each technique. For XGBoost, we reflect the number of trees, maximum depth, and learning rate. For CNN, we describe the number of convolutional layers, max pooling 1D, and the number of filters. For MLP, RNN and Seq2Seq, the table lists the number of hidden layers, dropout layers, and layer size. The best configurations obtained with the training data are highlighted in section 5.2.

4.3 Metrics

To validate and compare the different results of our experiments, we must define the error metrics to be minimized. Since we have a regression problem, we use MAE (Mean Absolute Error), which aggregates at m the absolute difference between the predictions and the actual values of *delay* data points:

$$MAE(m) = \frac{\sum_{i=0}^{delay-1} |y_i - \hat{y}_i|}{delay} \quad (1)$$

We also use the normalized MAE, namely NMAE, which averages the error for a batch of size N .

$$NMAE = \frac{\sum_{m=0}^{N-1} MAE(m)}{N} \quad (2)$$

5 Experiments and Results

This section shows the results of the experiments after data preprocessing and model training and validation with the algorithms mentioned above: decision trees, XGBoost, and neural networks (MLPs, CNNs, RNNs and Seq2Seq).

The implementation of the preprocessing and the prediction algorithms was developed with TSxtend ([Morcillo-Jimenez et al, 2023](#)), our open source library for batch analysis of sensor data.

5.1 Preprocessing and Data Preparation

In our experiment, we used processing data techniques such as aggregation, data modification and removal, data transformation, outliers detection, missing values detection, normalization and feature selection. We discarded heating consumption with null values to discard rows with no heating consumption. Then, we eliminated the variables with extreme values and finally joined the variables measuring the same consumption type on the same floor.

The variables in the collected dataset store aggregated values, e.g., summing up new values at each instant. The transformation of these variables was done with the differences between each instant of time, thus obtaining the actual consumption of each variable at each instant of time. These variables were used as inputs for the execution of our prediction algorithms.

In the subsequent step, we employed techniques to detect and remove outliers for each consumption variable. We calculated percentiles for each variable and searched for values outside this range. Such values were substituted with more conventional values (mean, median, etc...). This approach, however, leads to a loss of information. Fortunately, the number of outliers in our case is limited.

We had in our dataset missing values in February when the system did not gather data, for which we applied missing values imputation algorithms. Particularly, we used Interp to fill gaps by interpolation of known values. Interpolation techniques are utilized to fill gaps by interpolating known values, as mentioned in work by ([Montero-Manso and Hyndman, 2021](#)). Specifically, this approach involves the utilization of Interp, a commonly employed software for performing interpolation. This methodology aims to accurately estimate values within a given range based on the available data. The interpolation process involves estimating each data point's value by considering its neighbouring data points. Applying this technique makes the resulting dataset complete and can be analyzed more effectively for the intended purpose. [Figure 5](#) depicts data after imputation for three variables.

Then, we selected the most relevant variables for predicting energy consumption as Energy Zone D1 and Energy Zone D5/2. We calculated the cross-correlation between pairs of variables and grouped them by levels according to their correlation to the prediction target. [Figure 6](#) indicates few dependencies between the electrical sensors. There is a high correlation between the heating consumption variables, while water consumption does not correlate. In the case of the occupancy variables, there are more correlations, but it is not high because the values are estimated. We can also observe energy consumption when the building is empty.

Then we used XGBoost algorithm for the assessment of variable importance. In [figure 7](#), we can see that the most relevant variables for energy

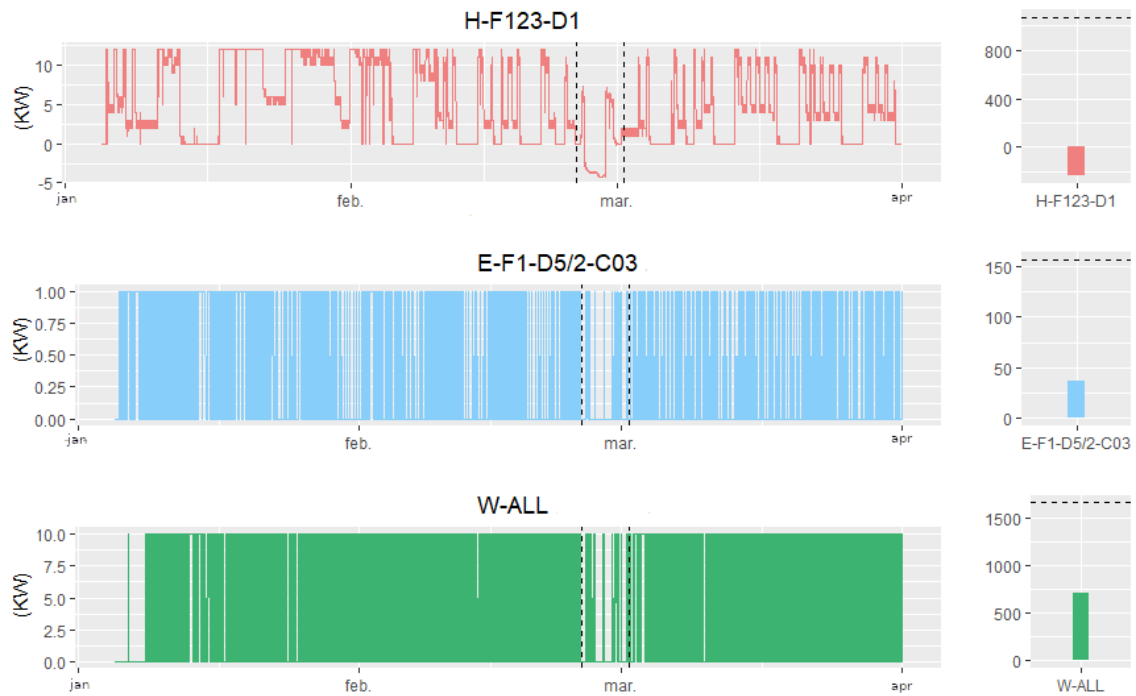


Fig. 5: Imputation of missing values ((end February) with Interpolate for variables Energy Zone D1, E-F1-D5/2-C03 and W-ALL.

consumption are occupancy, outside temperature and heating consumption (Energy Zone D1 and Energy Zone D5/2). With this information, we performed the selection of variables, being the most important ones (not surprisingly) Outside Temperature, Occupation, Energy Zone D5/2 and Energy Zone D1.

The last step in order to preprocess data is to normalize the dataset. To do this, we calculated the mean and standard deviation of each variable, and then transformed the values to the range $[0, 1]$.

5.2 Results and Discussion

We experimented with the January, February and March data using the selected algorithms. We applied grid search to obtain the best configurations and hyperparameter, yielding the values depicted in table 3. We can observe that the best configurations are similar for each subproblem.

The validation errors of the best models are shown in tables 4 and 5, while figures 8 to 13 depict the predicted vs the real values. In January, the algorithm that offered the best results was Seq2Seq, which achieved a validation NMAE of 0.21 for the prediction of consumption in zone D1 and 0.20 for zone D5/2, significantly improving the performance of the other models. The best model for February in both zones was CNNs. Overfitting was lower than that with other algorithms, at the cost of having worse validation metrics —the model tends to ignore or lessen the consumption peaks, as it can be seen in the chart. A second reason to explain these results is that we had many more missing

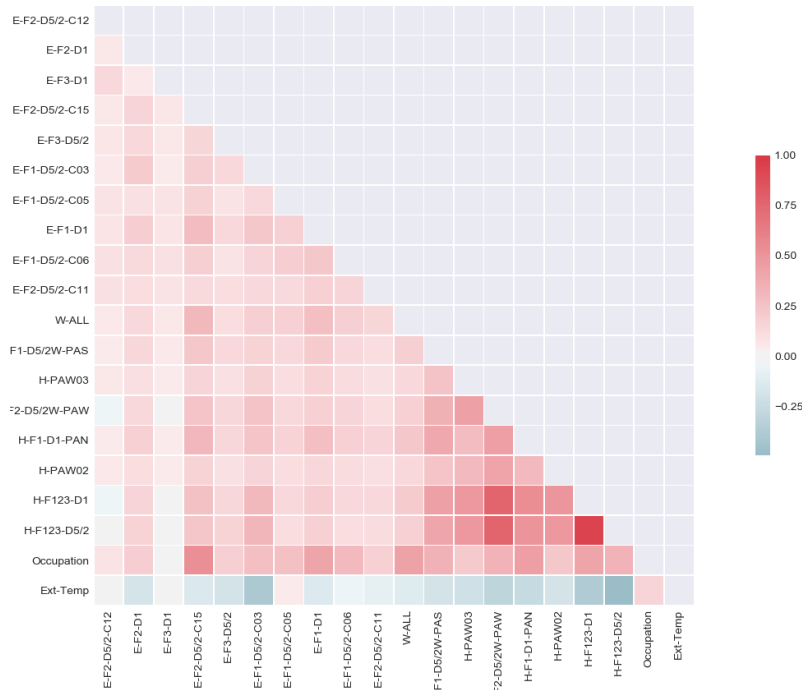


Fig. 6: Heatmap showing correlations between variables.

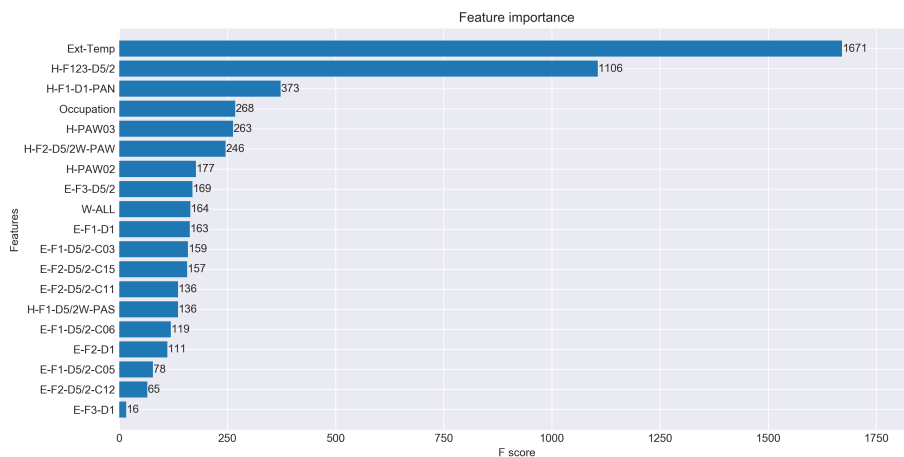


Fig. 7: Ranking of variables by importance with XGBoost.

values to impute in this period. The best method for March was RNNs, with an NMAE of 0.29 in both zones, almost twice better than MLP and XGBoost. Overall, we confirmed the ability of memory-based models (Seq2Seq and RNN) to extract characteristics from the input time series and learn the predictions in more diverse scenarios, while CNNs were slightly better when there was a uniform trend and fewer data. In all cases, the best models' MAE was around 5 kW, which is small enough for this type of applications.

Regarding the drawback of having limited data for training and validation, it resulted in that the more complex models did not learn the prediction of heating consumption as well as it could be expected. Hence, we suggest

Algorithms	Parameters	Month January	Month February	Month March
XGBoost	Number of trees	{50}	{100}	{50}
	Maximum Depth	{51}	{100}	{51}
	Learning rate	{0.05}	{0.001}	{0.005}
CNN	Number of convolutional layers	{1}	{1}	{1}
	Max pooling 1D	{3}	{3}	{3}
	Number of filters	{32}	{16}	{16}
MLP	Number of hidden layers	{1}	{1}	{1}
	Number of dropout layers	{0.3}	{0.15}	{0.15}
	Number of neurons	{128}	{128}	{128}
RNN	Number of hidden layers	{1}	{1}	{1}
	Number of dropout layers	{0.15}	{0.3}	{0.5}
	Number of neurons	{128}	{128}	{128}
Seq2Seq	Number of hidden layers	{2}	{2}	{2}
	Number of dropout layers	{0.3}	{0.3}	{0.3}
	Number of neurons	{64}	{64}	{64}

Table 3: Best configurations and hyperparameters for each algorithm and subproblem.

<i>Models</i>	<i>January</i>		<i>February</i>		<i>March</i>	
	<i>NMAE</i>	<i>MAE (kw)</i>	<i>NMAE</i>	<i>MAE (kw)</i>	<i>NMAE</i>	<i>MAE (kw)</i>
XGBoost	0.48	2.5	1.17	5.88	0.56	2.24
MLPs	0.38	1.97	0.82	4.12	0.53	2.12
RNNs	0.27	1.40	0.43	2.16	0.29	1.16
CNNs	0.34	1.77	0.43	2.16	0.34	1.36
Seq2Seq	0.21	1.09	0.43	2.16	0.39	1.56

Table 4: Performance of the best models for Energy Zone D1.

<i>Models</i>	<i>January</i>		<i>February</i>		<i>March</i>	
	<i>NMAE</i>	<i>MAE (kw)</i>	<i>NMAE</i>	<i>MAE (kw)</i>	<i>NMAE</i>	<i>MAE (kw)</i>
XGBoost	0.56	16.05	1.10	26.125	0.60	8.25
MLPs	0.31	8.88	0.61	14.48	0.64	8.8
RNNs	0.30	8.6	0.31	7.36	0.29	3.98
CNNs	0.36	10.32	0.30	7.12	0.33	4.53
Seq2Seq	0.20	5.73	0.36	8.55	0.37	5.08

Table 5: Performance of the best models for Energy Zone D5/2.

that more sophisticated techniques (Seq2Seq, but also transformer-based architectures) might not be necessary in this kind of problems or under similar circumstances. Instead, RNNs or CNNs with proper data pre-processing could be precise enough and less prone to overfitting.

6 Conclusions and Future Work

Concluding our discussion, we can see how the XGBoost model shows worse results, as XGBoost does not extract the dependencies between the variables it receives as input to perform the energy consumption prediction. With MLP

Graphical representation of best models for each subproblem and zone.

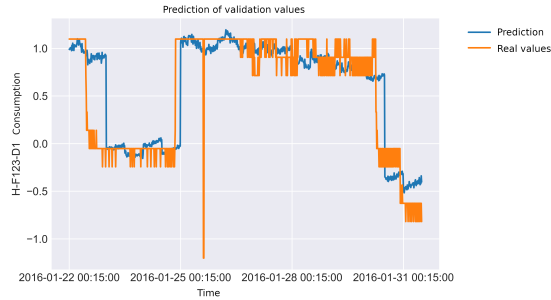


Fig. 8: Prediction of the Validation Values in January with Seq2Seq Energy Zone D1.

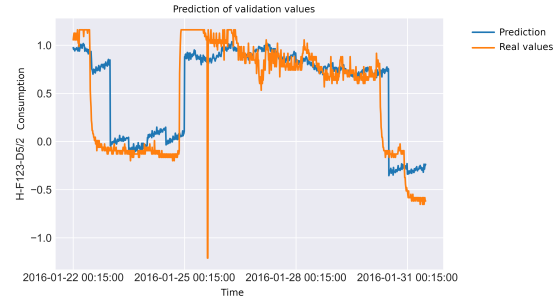


Fig. 9: Prediction of the Validation Values in January with Seq2Seq Energy Zone D5/2.

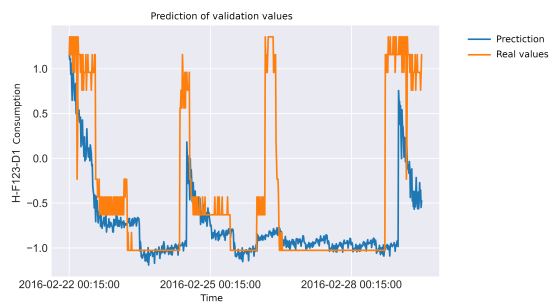


Fig. 10: Prediction of the Validation Values in February with CNNs Energy Zone D1.

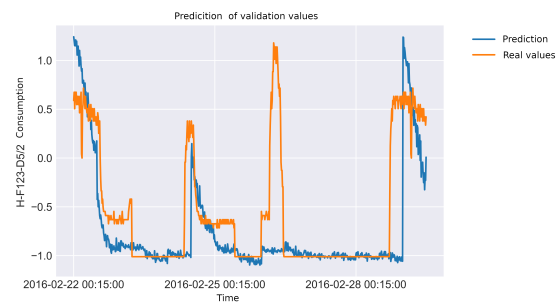


Fig. 11: Prediction of the Validation Values in February with CNNs Energy Zone D5/2.

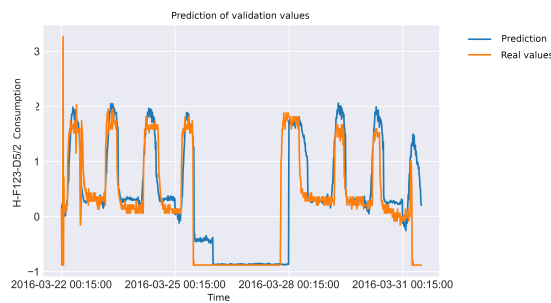


Fig. 12: Prediction of the Validation Values in March with RNNs Energy Zone D1.

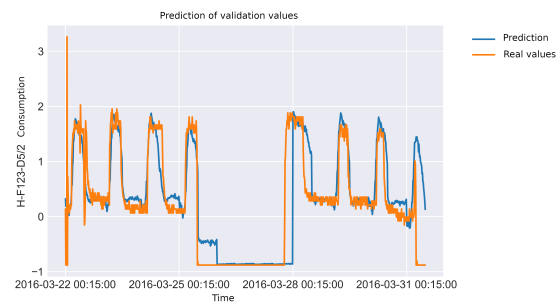


Fig. 13: Prediction of the Validation Values in March with RNNs Energy Zone D5/2.

models we have shown that they do not provide better results than other models because they do not allow to follow a chronological order in our dataset. With respect to the use of RNNs on our dataset, we have been able to reduce the NMAE error function by half, as these algorithms remember the information by processing the data in chronological order. Furthermore, we have applied Seq2Seq, which has allowed us to observe that, like RNNs, they obtain

good results in general but do not perform very well on datasets with too many missing values (i.e., smaller in size). Finally, convolution networks (CNNs) have been found to perform better than RNNs and Seq2Seq algorithms on sections of the dataset with a large number of missing values and steady trends.

One of the problems we have in this study has been the limited amount and quality of data, since we have only worked with the ICPE building sensors for three months including many missing values. In the future we can take the ICPE building sensors from other years to have a more extensive training and validation. Another option may be to generate artificial values with building simulation models. We can also use recurrent neural network models using attention mechanisms to improve synthetic data generation and missing values imputation (Bülte et al, 2023). Additionally, peak changes could be addressed with noise reduction techniques to smooth abrupt oscillations, e.g., as the filter proposed in (Ma et al, 2019).

7 Acknowledgments

The research reported in this paper was supported by the project IA4TES (MIA.2021.M04.0008, funded by the European Union – NextGenerationEU); the FEDER programme 2014-2020 (B-TIC-145-UGR18 and P18-RT-1765); the FEDER programme and the Andalusian Regional Government (A-TIC-244-UGR20); and the European Union (Energy IN TIME EeB.NMP.2013-4, No. 608981).

References

- Ahmad T, Chen H (2018) Short and medium-term forecasting of cooling and heating load demand in building environment with data-mining based approaches. *Energy and Buildings* 166:460–476. <https://doi.org/10.1016/J.ENBUILD.2018.01.066>
- Ahmad T, Chen H, Guo Y, et al (2018) A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand: A review. *Energy and Buildings* 165:301 – 320. <https://doi.org/10.1016/j.enbuild.2018.01.017>
- Arpanahi G, Javadi M (2018) A review on applications of artificial neural networks and support vector machines for building electrical energy consumption forecasting. *Renewable and Sustainable Energy Reviews* 82:1814–1832. <https://doi.org/10.1016/j.rser.2014.01.069>
- Benedetti M (2015) A proposal for energy services classification including a product service systems perspective. *Procedia CIRP* 30:251 – 256. <https://doi.org/10.1016/j.procir.2015.02.121>

- Bülte C, Kleinebrahm M, Yilmaz HÜ, et al (2023) Multivariate time series imputation for energy data using neural networks. *Energy and AI* 13:100,239. <https://doi.org/https://doi.org/10.1016/j.egyai.2023.100239>, URL <https://www.sciencedirect.com/science/article/pii/S2666546823000113>
- Chau C, Leung T, Ng W (2015) A review on life cycle assessment, life cycle energy assessment and life cycle carbon emissions assessment on buildings. *Applied Energy* 143:395 – 413. <https://doi.org/10.1016/j.apenergy.2015.01.023>
- Chauhan R, Ghanshala KK, Joshi RC (2018) Convolutional Neural Network (CNN) for Image Detection and Recognition. ICSCCC 2018 - 1st International Conference on Secure Cyber Computing and Communications pp 278–282. <https://doi.org/10.1109/ICSCCC.2018.8703316>
- Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system p 785–794. <https://doi.org/10.1145/2939672.2939785>
- Chen W, Zhou K, Yang S, et al (2017) Data quality of electricity consumption data in a smart grid environment. *Renewable and Sustainable Energy Reviews* 75:98–105. <https://doi.org/10.1016/j.rser.2016.10.054>
- Chou JS, Ngo NT (2016) Time series analytics using sliding window meta-heuristic optimization-based machine learning system for identifying building energy consumption patterns. *Applied Energy* 177:751–770. <https://doi.org/10.1016/J.APENERGY.2016.05.074>
- Ezan MA, Uçan ON, Kalfa M (2017) Predicting short-term building heating and cooling load using regression tree algorithm. *Journal of Building Performance Simulation* 10(5):487–502. <https://doi.org/10.1080/19401493.2016.1202888>
- Gong G (2019) Research on short-term load prediction based on Seq2Seq model. *Energies* 12:3199. <https://doi.org/10.3390/en12163199>
- Gómez-Romero J, Fernández-Basso CJ, Cambronero MV, et al (2019) A probabilistic algorithm for predictive control with full-complexity models in non-residential buildings. *IEEE Access* 7:38,748–38,765. <https://doi.org/10.1109/ACCESS.2019.2906311>
- Khalil M, McGough AS, Pourmirza Z, et al (2022) Machine learning, deep learning and statistical analysis for forecasting building energy consumption — a systematic review. *Engineering Applications of Artificial Intelligence* 115:105,287. <https://doi.org/https://doi.org/10.1016/j.engappai.2022.105287>

- Lago J, Marcjasz G, De Schutter B, et al (2021) Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy* 293:116,983. <https://doi.org/https://doi.org/10.1016/j.apenergy.2021.116983>
- Ma W, Wang W, Wu X, et al (2019) Control strategy of a hybrid energy storage system to smooth photovoltaic power fluctuations considering photovoltaic output power curtailment. *Sustainability* 11(5). <https://doi.org/10.3390/su11051324>, URL <https://www.mdpi.com/2071-1050/11/5/1324>
- Marino DL, Amarasinghe K, Manic M (2016) Building energy load forecasting using deep neural networks pp 7046–7051. <https://doi.org/10.1109/IECON.2016.7793413>
- Micheletti A, Aletti G, Ferrandi G, et al (2020) A weighted χ^2 test to detect the presence of a major change point in non-stationary markov chains. *Statistical Methods & Applications* 29(4):899–912. <https://doi.org/10.1007/s10260-020-00510-0>, URL <https://doi.org/10.1007/s10260-020-00510-0>
- Mocanu E, Nguyen PH, Gibescu M, et al (2016a) Deep learning for estimating building energy consumption. *Sustainable Energy, Grids and Networks* 6:91–99. <https://doi.org/10.1016/j.segan.2016.02.005>
- Mocanu E, Nguyen PH, Kling WL, et al (2016b) Unsupervised energy prediction in a smart grid context using reinforcement cross-building transfer learning. *Energy and Buildings* 116:646–655. <https://doi.org/10.1016/J.ENBUILD.2016.01.030>
- Molina-Solana M, Ros M, Ruiz MD, et al (2017) Data science for building energy management: A review. *Renewable and Sustainable Energy Reviews* 70:598–609. <https://doi.org/10.1016/j.rser.2016.11.132>
- Montero-Manso P, Hyndman RJ (2021) Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting* 37(4):1632 – 1653. <https://doi.org/10.1016/j.ijforecast.2021.03.004>
- Morcillo-Jimenez R, Gutiérrez-Batista K, Gómez-Romero J (2023) Tsxtend: A tool for batch analysis of temporal sensor data. *Energies* 16(4). <https://doi.org/10.3390/en16041581>, URL <https://www.mdpi.com/1996-1073/16/4/1581>
- Pachauri N, Ahn CW (2022) Regression tree ensemble learning-based prediction of the heating and cooling loads of residential buildings. *Building Simulation* 15(11):2003 – 2017. <https://doi.org/10.1007/s12273-022-0908-x>

- Phinikarides A, Makrides G, Zinsser B, et al (2015) Analysis of photovoltaic system performance time series: Seasonality and performance loss. *Renewable Energy* 77:51–63. <https://doi.org/https://doi.org/10.1016/j.renene.2014.11.091>, URL <https://www.sciencedirect.com/science/article/pii/S0960148114008222>
- Pérez-Lombard L, Ortiz J, Pout C (2008) A review on buildings energy consumption information. *Energy and Buildings* 40(3):394–398. <https://doi.org/https://doi.org/10.1016/j.enbuild.2007.03.007>
- Rahman A, Srikumar V, Smith AD (2018) Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Applied Energy* 212:372–385. <https://doi.org/https://doi.org/10.1016/j.apenergy.2017.12.051>, URL <https://www.sciencedirect.com/science/article/pii/S0306261917317658>
- Raza MQ, Khosravi A (2015) A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renewable and Sustainable Energy Reviews* 50:1352–1372. <https://doi.org/10.1016/j.rser.2015.04.065>
- Schmidt M, Åhlund C (2018) Smart buildings as cyber-physical systems: Data-driven predictive control strategies for energy efficiency. *Renewable and Sustainable Energy Reviews* 90:742 – 756. <https://doi.org/10.1016/j.rser.2018.04.013>
- Seyedzadeh S, Rahimian FP, Glesk I, et al (2018) Machine learning for estimation of building energy consumption and performance: a review. *Visualization in Engineering* 6:1–20. <https://doi.org/10.1186/s40327-018-0064-7>
- Sherstinsky A (2020) Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena* 404:132,306. <https://doi.org/10.1016/j.physd.2019.132306>
- Taud H, Mas J (2018) Multilayer perceptron (MLP) pp 451–455. https://doi.org/10.1007/978-3-319-60801-3_27
- Tien PW, Wei S, Darkwa J, et al (2022) Machine learning and deep learning methods for enhancing building energy efficiency and indoor environmental quality – a review. *Energy and AI* 10:100,198. <https://doi.org/https://doi.org/10.1016/j.egyai.2022.100198>, URL <https://www.sciencedirect.com/science/article/pii/S2666546822000441>
- Torres JF (2021) Deep learning for time series forecasting: A survey. *Big Data* 9:3–21. <https://doi.org/10.1089/big.2020.0159>

Wang Z, Srinivasan RS (2016) A review of artificial intelligence based building energy prediction with a focus on ensemble prediction models. p 3438 – 3448, <https://doi.org/10.1109/WSC.2015.7408504>

Zeng A, Chen M, Zhang L, et al (2023) Are transformers effective for time series forecasting?

Zhang L, Wen J, Li Y, et al (2021) A review of machine learning in building load prediction <https://doi.org/10.1016/j.apenergy.2021.116452>

4.3. Extracción de conocimiento oculto en registros médicos mediante minería de datos y lógica difusa en un entorno distribuido.

- Referencia:1568-4946
- Estado: Aceptado
- Factor de Impacto: Q1
- Categoría: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE - SCIE
- DOI:<https://doi.org/10.1016/j.asoc.2022.108870>
- Revista/Editorial: Applied Soft Computing



A fuzzy-based medical system for pattern mining in a distributed environment: Application to diagnostic and co-morbidity

Carlos Fernandez-Basso*, Karel Gutiérrez-Batista, Roberto Morcillo-Jiménez, Maria-Amparo Vila, Maria J. Martín-Bautista

Department of Computer Science and Artificial Intelligence, University of Granada, 18071, Granada, Spain



ARTICLE INFO

Article history:

Received 30 November 2021
Received in revised form 4 March 2022
Accepted 10 April 2022
Available online 26 April 2022

Keywords:

Association rules
Fuzzy logic
Data mining
Medical records

ABSTRACT

In this paper we have addressed the extraction of hidden knowledge from medical records using data mining techniques such as association rules in conjunction with fuzzy logic in a distributed environment. A significant challenge in this domain is that although there are a lot of studies devoted to analysing health data, very few focus on the understanding and interpretability of the data and the hidden patterns present within the data. A major challenge in this area is that many health data analysis studies have focussed on classification, prediction or knowledge extraction and end users find little interpretability or understanding of the results. This is due to the use of black-box algorithms or because the nature of the data is not represented correctly. This is why it is necessary to focus the analysis not only on knowledge extraction but also on the transformation and processing of the data to improve the modelling of the nature of the data. Techniques such as association rule mining and fuzzy logic help to improve the interpretability of the data and treat it with the inherent uncertainty of real-world data. To this end, we propose a system that automatically: a) pre-processes the database by transforming and adapting the data for the data mining process and enriching the data to generate more interesting patterns, b) performs the fuzzification of the medical database to represent and analyse real-world medical data with its inherent uncertainty, c) discovers interrelations and patterns amongst different features (diagnostic, hospital discharge, etc.), and d) visualizes the obtained results efficiently to facilitate the analysis and improve the interpretability of the information extracted. Our proposed system yields a significant increase in the compression and interpretability of medical data for end-users, allowing them to analyse the data correctly and make the right decisions. We present one practical case using two health-related datasets to demonstrate the feasibility of our proposal for real data.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Due to the growing necessity for analysing health data properly, the development of robust systems that enable knowledge discovery from this sort of data has become a subject of great interest for companies and researchers. The extraction of hidden knowledge from health-related data, diagnostic and co-morbidity [1] analysis to create valuable medical information and improve healthcare services have all become vital challenges for health management and medical decision support [2,3].

Another significant challenge in this domain is the interpretability of the data mining systems. Interpretability can be defined as the degree to which a human can understand the cause of a decision [4] or the degree to which a human can consistently predict the results of the model [5]. The greater

the interpretability of a system is, the easier it is for end-users to understand why certain decisions or predictions have been made.

In order to improve the interpretability and diagnostic and co-morbidity analysis of the medical data, we can use valuable techniques such as association rules extraction and fuzzy logic. The first is a popular unsupervised approach used to explore and interpret large transactional datasets to identify unique patterns and rules [6]. The latter was introduced in [7] and allows real-world data to be represented and analysed in a better way. In other words, the fuzzy set theory provides efficient mechanisms to manage incomplete or imprecise information.

Association rules have been widely used for pattern discovery in different domains [8–11]. Specifically, in medicine, we can find several studies that use association rules to extract hidden patterns which are useful for the diagnosis and subsequent possible treatment of a patient [12–14]. Taking into account the above and the fact that in medical databases there are incomplete and

* Corresponding author.

E-mail address: cjferba@decsai.ugr.es (C. Fernandez-Basso).

imprecise data (mostly continuous features), approaches such as fuzzy association rules would seem to be the most suitable for being an excellent solution to discretize numerical values softly and uncover and represent hidden relationships in an understandable way for end-users [15].

Another important aspect is that standard association rule mining (ARM) algorithms do not solve multitasking rule problems, because they ignore the correlation between tasks. There are algorithms such as multi-task association rule miner (MTARM) that works in a way that joins several rules to deal with this type of problem. It divides the rules into individual tasks and through a voting system generates new rules that produce global results called multitask rules [9]. In our study we approach the problem in a distributed way by dividing the problem, finally unifying the different solutions.

Let us consider the following example: we have a medical-related dataset consisting of the attributes *diagnosis*, *occupation*, and *age* (where each instance in the dataset corresponds to a patient's surveillance), and we want to know the number of patients with a specific diagnosis in a given range of ages.

We can solve the problem by means of a simple query. This approach is helpful if it is intended for performing a shallow analysis of the data. However, it does not allow a more detailed analysis, such as determining the number of *young* patients with a *complex* diagnosis. In this situation, technologies such as fuzzy association rules are revealed as the most appropriate technologies to help with these problems.

One of the main goals of medical databases is to store historical data about the patients. That is why when using this sort of database, we should keep in mind that the use of data mining methods runs into problems when they are used to analyse vast amounts of data and become less efficient at processing and analysis. To tackle this problem, we must implement the algorithm in a distributed environment.

In this study, we propose creating a fuzzy-based medical system for pattern mining in a distributed environment. The system allows end-users to discover and interpret hidden patterns from health-related data, thus facilitating diagnostic and co-morbidity analysis, subsequent treatment, and also the early prevention of potential diseases. Our proposal is based on tools such as association rules, fuzzy logic, and Big Data. The proposal is entirely automatic and unsupervised, allowing us to experiment with both labelled and unlabelled data. We now summarize the main contributions of this paper:

- **Data enrichment** - During this process, the original medical database is transformed and adapted (features engineering) to prepare the data for the data mining process to generate more interesting patterns.
- **Feature fuzzification** - Through this process, we perform the fuzzification of the features enriched in the above step, enabling us to treat imprecise data and discover and analyse relevant patterns and relationships.
- **Visualization** - Finally, we visualize the obtained results efficiently and in a user-friendly way to facilitate the analysis and improve the interpretability of the information extracted.

The rest of the paper is organized as follows: Section 2 summarizes the main work developed in this research area. Section 3 presents the proposed fuzzy-based system for pattern mining in Big Data. Section 4 presents one case from real-world medical data and we discuss the obtained results. Finally, in Section 5, the conclusions derived from the analysis are presented.

2. Related studies

As mentioned above, in this paper, we propose a fuzzy information-based system for discovering hidden patterns and relationships to address the problem of interpretability and diagnostic and co-morbidity analysis of medical-related data. One of the main problems of such studies is how the results are presented to end-users. Most studies show results without focusing on end-user understanding. Therefore, the main objective of this research is to improve the interpretability of obtained results in medical data centres using data mining techniques such as association rule discovery and fuzzy logic. This section presents research on data mining and fuzzy-based systems, mainly focused on the medical domain.

2.1. Data mining system

Data mining techniques have been widely applied in numerous fields of science. An instance where we can observe this is in energy [16–19]. In [16,17] the authors review how some traditional data mining techniques have been used to obtain construction-related information. In social science, we can observe numerous studies about data mining [20–22] where text pre-processing is applied. Other fields such as Physics and Astronomy apply pre-processing techniques to images [23–26]. As in our case, data mining techniques are also widely used in the medical field, as we can observe in [27–29]. In this field, we can classify the studies into two groups, those dealing with imaging data [30–33] and those dealing with medical records [34–37].

In a more specific health study [28], such as the analysis of brain signals, problems such as sample size and signals considered as noise are encountered. In this study, the authors present a plausible method to detect and distinguish directions from Electroencephalography (EEG) signals by using feature extraction techniques to perform brain signal processing.

In this other study [30], an automated system for the extraction and classification of tumours from magnetic resonance images has been developed. The proposed system consists of five main steps: tumour contrast, tumour extraction, multi-model feature extraction, feature extraction and classification. Other techniques used in some studies, such as [38–41], focus on unsupervised algorithms.

We can observe studies that pre-process the source data to improve data quality and improve results. Different types of data require different processing technologies. Most structured data usually require classical pre-processing technologies, such as data cleaning, data integration, data transformation and data reduction [34,42,43].

Big data technology has many areas of application in the healthcare sector [27,44–46], such as predictive modelling and clinical decision support, disease surveillance and research. Big data analytic often leverages analytical methods developed in data mining, such as classification, clustering and regression.

In our study we have proposed a series of techniques that differ from other studies in the way we apply the techniques and unify them so that at the end of the procedure we have a set of data ready for knowledge extraction and interpretation.

We have enriched our data by adding the different diagnoses that occur during the duration of the patient's stay in the medical action protocol to resolve the proposed diagnosis using external sources. We have used basic data pre-processing techniques, such as the detection and elimination of missing values within the medical data centre, as well as the elimination of outliers by selecting fields not relevant to our study.

The next step in our procedure is the transformation of our data by applying fuzzy techniques to finally create association rules with the different data in order to detect the existing relationships between the different fields that we are going to analyse in our study.

2.2. Fuzzy-based system

Most of the problems presented by the medical data centres are related to the way to structure the information within the database, as well as the way to represent it, so that it can be correctly interpreted by the end user. Therefore, it is necessary to use fuzzy techniques to transform imprecise data into accurate data capable of being interpreted by the end-user. In [47] the authors propose new measures of accuracy and usefulness for fuzzy association rules extraction from medical relational databases. The approach presented by the authors allows the significant reduction of the number of rules without information being lost.

Another interesting example found in the state of the art is the following study [48] aimed at the application of fuzzy techniques in the healthcare sector. It shows how wearable sensors can be used to create a system for recommending specific prescriptions for patients with diabetes.

There is also a theoretical study [49,50] on solving linearly posed problems and applying a fuzzy programming algorithm, where the results can be obtained in a way that eliminates uncertainty.

Such stacks can be made fully scalable by applying the algorithms in a distributive manner [51]. This study presents biomedical data stored in the cloud and demonstrates how such algorithms are ideal for solving large-scale problems.

In our study, we look at the fuzzification of diagnoses and how through the co-morbidity of some diagnoses we can give results on patients so that they are displayed through our application as simple, medium or complex, depending on the number of co-morbidity diagnoses they present, thus allowing us to represent a traceability of the patient's history.

2.3. Co-morbidity analysis

Another aspect that we address in the scope of this study is to take into account the co-morbidity of the different diagnoses. Co-morbidity is the occurrence of a disease as a function of having a previous disease in the same person [52]. At the same time these co-morbidity data are often reported statistically, mainly in the context of academic research to inform the health system and public health agencies in their decision-making.

There are very few studies that focus on the analysis of data directed to the initial diagnosis as well as, the diagnosis of some co-morbidity diseases, generated by the main diagnosis. This study [35] focuses on autism in children and on the different degrees according to the different co-morbidities they suffer from. In another study [53] we found in the state of the art is aimed at detecting unnecessary blood tests, giving as an example patients with upper gastrointestinal bleeding and patients with unspecified bleeding in the gastrointestinal tract in order to analyse the amount of calcium and haemocytes in the blood. Two experiments are performed, firstly, labelling the different tests that are promising and secondly grouping patients with co-morbidity.

Finally, a case study is shown on the disease called diabetic retinopathy, which is a co-morbidity generated by diabetes [54], and how through the application of classification techniques together with fuzzy techniques and balancers we can determine which patients are most at risk of suffering from this type of co-morbidity.

Part of our study has focused on the analysis of the co-morbidity of our records. Based on this analysis, we will be able to interpret patients as simple, standard and complex, by applying fuzzy logic.

3. A fuzzy-based system for pattern mining in Big Data

In this paper, we propose a complete system for managing and extracting information in health systems using big data architecture. It presents a process of integration, data processing with two novel phases of enrichment and treatment of uncertainty in this type of data using fuzzy techniques. The system also integrates an algorithm for extracting fuzzy association rules in Spark and tools for visualizing and interpreting the results by end-users.

3.1. System architecture and workflow

Fig. 1 depicts the complete system that follows our proposal. It can be divided into three big blocks. In the first, the data collection is carried out. Hospital data in Spain follow a standard format concerning basic patient data for each visit to a hospital, a diagnostic testing centre, surgical interventions or specialist consultations. This data can then be aggregated from external sources from other areas or services of the hospital depending on the hospital and its data management system. Therefore, depending on the type of source, be it a hospital, medical tests from one of the areas of the hospital or databases from each of the medical departments of the hospital, our system collects the data and merges it with the patient's historical data in order to have them all modelled according to the patient. In this way, for example, in the case of patients with recurrent visits to different services and departments of the hospital, we can see their traceability and diagnosis in each of these visits. In addition, this innovative system can merge hospital data into a single database with all the validations described above in this first phase of the data collection process.

The data is then stored in Big Data architecture that allows large data sets from heterogeneous sources to be used. For this purpose, NoSQL databases [55,56] have been used because they provide great flexibility for storing data from heterogeneous sources and large volumes. On the other hand, distributed processing tools such as Apache Spark [57] have been used to manage, process and analyse this massive data efficiently. Using this type of technology, the system processes this data through three modules: preprocessing, enrichment and fuzzification, which will be explained in Section 3.3.

In the last block, knowledge extraction from the processed database will be carried out. This block uses an algorithm of fuzzy association rules in Big Data [58] which will allow us to extract association rules using Spark. In addition, our user interface will implement some of the most widely used tools for visualizing association rules to facilitate the understanding and interpretability of the results to the end-users.

This whole process has been applied in the scope of the BigDataMED project to different hospitals in Spain. Here we will present the results obtained from 2016 to 2020 in two hospitals, one in Marbella and the other in Granada. However, the proposed system is general and can be applied to other hospitals and medical centres.

3.2. Data collection

The developed workflow has been used in two Spanish hospitals, the clinical hospital of Granada and the hospital of Marbella. The figure below depicts the different data sources collected and added to the database (see Fig. 2).

Every piece of data has been collected through different procedures. On the one hand, we have considered the data provided by the medical management systems, which, by means of user applications, collect the data from the different services such as consultations, emergencies, etc. On the other hand, we have

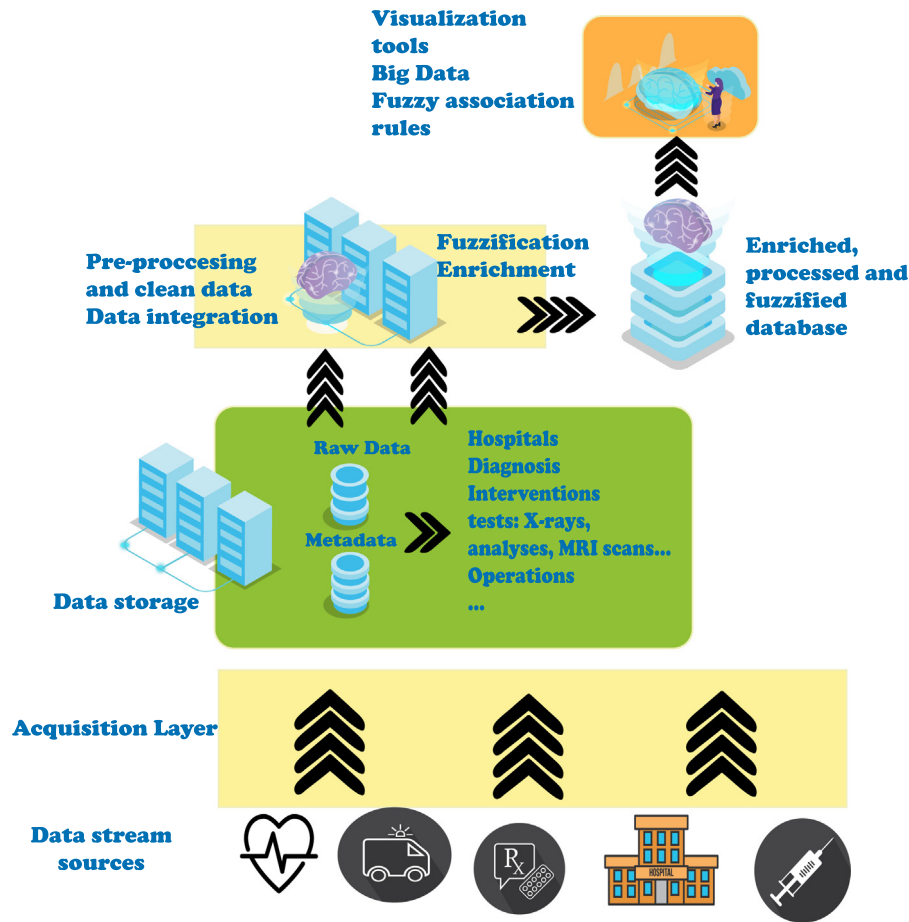


Fig. 1. General process of our proposal.

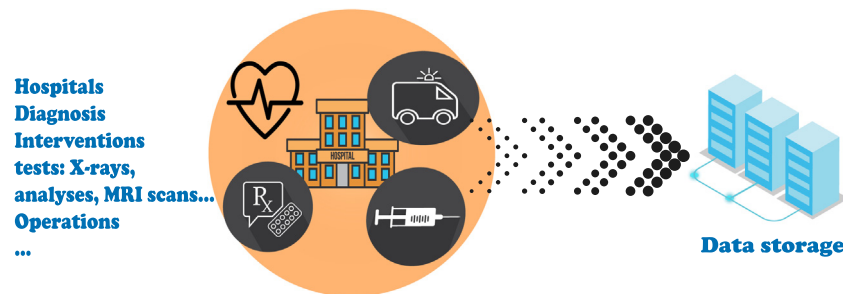


Fig. 2. Diagram of the type of data collected for the hospital system.

the data sources of the surgical interventions, diagnostic tests, etc. In particular, all these data have to be merged using the patient’s history and adapting the fields so that these data can be merged by modelling their relationships. For example, the patient has to be related to the diagnostic tests performed, the surgical interventions and other sources in the system.

3.3. Pre-processing, enrichment and fuzzification

One of the essential phases in the extraction of interesting knowledge from large datasets is the preparation of these datasets for the application of data mining techniques. This phase is called data pre-processing and encompasses different groups of data processing techniques that enable the creation of a dataset with better conditions for knowledge extraction.

In our proposal, several pre-processing modules are presented. In them, some variables have been transformed to better model

the information they contain. Moreover, in some cases, external data have been used to enrich the data contained in the dataset variables; finally, an uncertainty treatment process has been carried out using fuzzy logic. In this process, a novel fuzzification process has been used that either automatically or guided by expert knowledge can use fuzzy labels to better model the information and allow more interpretative results for the end-users.

3.3.1. Pre-processing data

In the first pre-processing module a classical preprocessing is carried out, eliminating codes from the database that do not contain useful information, normalizing some values and transforming some factor variables. Subsequently, processing which is more oriented to medical data has been carried out. Variables such as history codes, postcodes or dates have been processed into data that better represent the information. For example, postcodes have been transformed into data of municipalities,

Table 1
Example of the processing of temporary data from the hospital database.

Date admission	Date discharge	Birthdate	
13/5/2016 15:14:30	20/5/2016 15:14:30	11/2/1957	
10/10/2019 15:16:45	12/10/2019 15:16:45	11/2/2016	
↓			
Admission time	Season	Age	Patient type
7	Spring	64	Adult
2	Autumn	5	Child

Table 2
Element description.

Features	Codification	Method for enrichment
Diagnoses	International Classification of Diseases (ICD)	External API
Origin of patients	Spanish health system	Database
Reason for discharge	International Classification of Diseases (ICD)	External API
Reason for admission	International Classification of Diseases (ICD)	External API
Surgical procedures	International Classification of Procedures	External API
Diagnostic tests	International Classification of Test	External API
Services/ departments	Spanish health system (SNS)	Database
Other data ^a	Andalusian health system (SAS)	Database

^aRest of the variables related to the management and information of the Spanish hospital system.

cities and countries and dates by their month, year and day of the week, and whether the day is a holiday. In addition, the dates have been processed with the patient's age and data related to the patient's stay in hospital, such as the time of admission, stay in the Intensive Care Unit (ICU), etc. This process can be seen in the example in Table 1.

3.3.2. Enrichment

Some of the data collected within the database come from taxonomies, external coding dictionaries. This is the case for diagnoses, the origin of the patients, reasons for discharge or admission, surgical procedures and 22 other variables. It is for these types of variables that within our processing workflow, we have created a data enrichment module in which we have added information to the codes associated with each of these fields, or we have modelled their structure in order to add the information found in the external taxonomies and databases where the codes are decoded. Table 2 shows some of the characteristics that required the extraction of external information to improve their interpretability and meaning.

For this enrichment, a series of processes have been created to extract the information contained in the coding of the variable and this information has been modelled to add to the database. This occurs because the coded variables such as diagnoses have a tree-like coding that includes other levels. In addition to this, at the lower level, we find interesting information such as synonyms, alterations or problems related to the disease. Therefore our pre-processing for each diagnosis will generate up to 5 different levels of the disease. It will also generate information such as:

- Applicable to these diseases (list of diseases).
- Approximate synonyms.

- The disease is grouped within the Diagnostic Related Group (list of diseases).

An example of this process can be seen in Table 3, in which we can see how we process the diagnosis in question, as well as how the different new elements providing more information about it are grouped.

This process has been created for the different variables in the table. For each, the type of data and the type of information of interest were taken into account for their extraction.

3.3.3. Fuzzification

In the final part of the processing, we have developed a novel feature fuzzification methodology to improve data interpretability. This last phase is crucial as much of the data coming from the system is challenging to represent and interpret by end-users, as it is a continuous value with measurements that are often complex to interpret and understand. Fuzzification of these data can improve the results found by the mining algorithms, and at the same time, increase the interpretability of the obtained results.

We propose a fuzzification algorithm that allows an automatic treatment of data values according to their distribution, depending on the variables in the dataset or information provided by an expert (see the types of input provided to the algorithm). For this purpose, we have developed a distributed algorithm in Spark following the MapReduce philosophy. This allows us to process large amounts of data, such as in the case of the data stored by different hospitals in the Spanish health system.

The general process is described in Algorithm 1. For this, we use Spark for the distribution of data along the cluster. The algorithm has input a dataset, a python dictionary (hash) and an integer. The dictionary is used to store the ranges and labels of the variables that the experts have defined. On the other hand, the default number of labels is used for variables that the users have not defined and will be created automatically based on their distribution. If the values depend on dataset variables, we will pass a dictionary containing the labels and the validation function applied to the dataset variable to the algorithm. For example, if we depend on the number of diagnoses, we will pass the variable number of diagnoses and the function that returns a label with its membership value.

The whole process will be processed in a distributed way. It should be noted that Spark automatically divides the data into chunks for the distributed computation. We have specified this with the acronym DCS (distribute computing using Spark) and representing each chunk of data by S_i . In line 6, we can see that a global variable is used throughout the cluster, which is then used by the function that distributes the computation through MapReduce (line 8 of Algorithm 1).

Additionally, in line 8, the procedure calls to the *fuzzification* function described in Algorithm 2. This function is divided into different parts. Firstly, it checks if the name of the variable is found in the *Intervals* hash-list, if it is found in the python dictionary, the new fuzzified variables are created using the names of the labels specified by *Intervals* and its configuration (i.e. computation of membership degrees) attending to the specified interval (see lines 10–16 of Algorithm 2). If the variable is not found in the dictionary, an automatic procedure is used that divides the values of the variable into several intervals defined in *DefaultIntervals* according to the percentiles of the variable.

Fig. 3 shows an example with the value of *DefaultIntervals* = 3 where the y -axis represents the degree of membership and

Table 3
Example of the processing of a diagnosis C00.4.

Diagnosis					
C00.4					
Dig level1	Dig level2	Dig level3	Application to	Approximate synonyms	Group diagnosis
Neoplasms	Malignant neoplasms of lip, oral cavity and pharynx	Malignant neoplasm of lip	Cancer, lower lip, inner aspect	Malignant neoplasm of frenulum of lower lip ...	Tracheostomy for face, mouth and neck diagnoses or laryngectomy with mcc

Algorithm 1 Main Spark procedure for fuzzification preprocessing algorithm.

```

1: Input: Data: RDD transactions: {t1, ..., tn}
2: Input: DefaultIntervals: number of intervals automatically generated by the algorithm
3: Input: Intervals: Hash-list of intervals for each variable: {Variablei : [{Intervals}, {Labels}], ..., Variablep : [{Intervals}, {Labels}]}
4: Output: Fuzzy transactions containing fuzzified values

Start Algorithm

5: Features = Dataset.NameFeatures()
6: broadcast(Global_Features) #Create a broadcast variable for its use across the cluster
7: DCS in q chunks of Data: {S1, ..., Sq}
8: FuzzyDataSi ← Si.Map (Fuzzification(tk ∈ Si))
   # Map function computes independently each transaction in Si
9: FuzzyDatabase =
   = ReduceByKey(Aggregation(FuzzyDataS1, ..., FuzzyDataSq))
10: return FuzzyDatabase
    
```

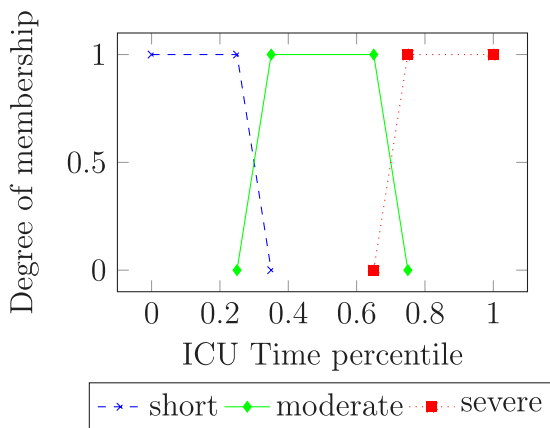


Fig. 3. Example of automatic execution with 3 default intervals.

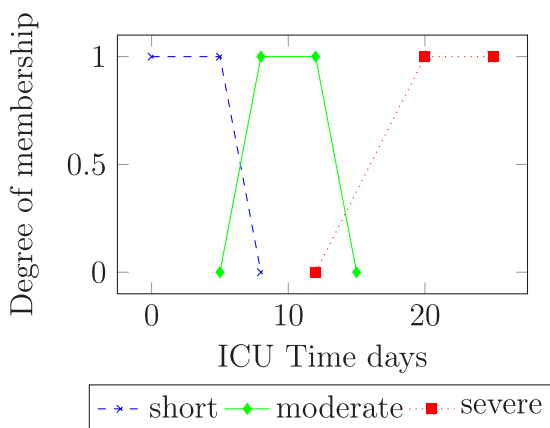


Fig. 4. Example of expert definition execution with 3 intervals.

x-axis the percentile of the variable. In this example, the percentiles used have been 25 and 37,5 for defining the trapezoidal form of the first label and left part of the second label, and 62,5 and 75 for defining the right part of second label and the third label. So, the *GenerateIntervals* function divides the set into *k* equidistributed fuzzy sets using the corresponding percentiles. For instance, for *k* = 4, the considered percentiles are computed as follows: $\{\frac{100}{k+1}, \frac{100}{k+1} + \frac{100}{(k+1)(k-1)}, \frac{2 \cdot 100}{k+1} + \frac{100}{(k+1)(k-1)}, \frac{2 \cdot 100}{k+1} + \frac{2 \cdot 100}{(k+1)(k-1)}, \frac{3 \cdot 100}{k+1} + \frac{2 \cdot 100}{(k+1)(k-1)}, \frac{3 \cdot 100}{k+1} + \frac{3 \cdot 100}{(k+1)(k-1)}\}$ which results in $\{p_{20}, p_{26.6}, p_{46.6}, p_{53.3}, p_{73.3}, p_{80}\}$. On the contrary, the *FuzzyDivision* function uses the defined intervals of the global variable *Intervals*.

This processing method (Algorithms 1 and 2) has a complexity $O(n/c)$ where *n* is the number of transactions and *c* is the number of computation units used in a distributed way. This complexity is due to the fact that the algorithm must go through the data set elements transforming them into the different fuzzy labels.

algorithm 2 Fuzzification function.

```

1: Input: Data: A transaction: tk = {item1, ..., itemm}
2: Global distributed variable: Intervals: Hash-list of intervals for each variable : {Variable1 : [{Intervals}, {Labels}], ..., Variablep : [{Intervals}, {Labels}]}
3: Input: DefaultIntervals: number of intervals automatically generated by the algorithm
4: Output: Fuzzy transaction
    
```

Start Algorithm

```

5: Features = Dataset.NameFeatures()
6: DistributeVariable(Features)
7: DCS in q chunks of Data: {S1, ..., Sq}
8: i=0
9: do
   # Check if the variable exists in the hash list
10: if Feature[i] ∈ Intervals then
11:   Interval=Intervals[Feature[i]][0]
12:   Labels=Intervals[Feature[i]][1]
13: else
14:   Interval = GenerateIntervals(DefaultIntervals,Data[Feature[i]])
15:   Labels = GenerateLabels(DefaultIntervals)
16: end if
17: for j = 0; j < |Labels|; j++ do
18:   FuzzyData[Label]=FuzzyDivision(Interval[j], Interval[j+1], type)
   # type ="linear", "exponential", "logarithmic"...
19:   j++
20: end for
21: while |Feature| > i
22: return FuzzyData
    
```

For the use case under study, the experts determined different intervals for generating the fuzzy labels. These depend on the nature of the variable, e.g. the ICU stay could be defined automatically as we have seen in Fig. 3 or defined by an expert as in Fig. 4.

3.4. Data mining: Fuzzy association rules in Big Data

After data pre-processing and fuzzification, data mining techniques were applied to the processed data. In particular, an algorithm for association rule mining was applied in Big Data

Table 4
Different data sources.

Dataset	Documents	Features
Marbella hospital	750 000	273
Granada hospital	220 000	273

(BDFARE Apriori-TID Big Data Fuzzy Association Rules Extraction [58,59]). This algorithm has also been implemented following the MapReduce paradigm under the Spark Framework. It enables the processing of huge sets of fuzzy transactions, finding frequent itemsets and fuzzy association rules exceeding the imposed thresholds for support and confidence, given a set of α -cuts.

4. Use case: medical records

The results obtained by applying our proposal should be analysed from two points of view. On the one hand, the efficiency and capacity of our distributed processing and management system that allows the processing of large datasets from different sources and with heterogeneous and massive data in a more efficient way. On the other hand, the fuzzification of the variables and their application to discover fuzzy association rules through this processing.

The aim is to extract patterns between diagnoses and patient characteristics to study co-morbidity and improve our knowledge about the information in our system. This can improve tasks such as obtaining a diagnosis or preventing a possible diagnosis related to patient factors such as obesity, other diagnoses, smoking, etc.

4.1. Data sources

In order to validate the whole process and the architecture presented above, a use case will be carried out by extracting fuzzy association rules from two hospitals in the south of Spain. For this purpose, the data used have been collected from the different systems of each of the hospitals (emergency, hospitalization, consultations, etc.). We can see the characteristics of each of them in Table 4. The table already shows the values of the records and the characteristics of the database before preprocessing.

All these data have been collected automatically from the different hospitals. They have also been stored in our system explained in Section 3.1. By having these data in our tool, we have been able to carry out the processing and knowledge extraction processes.

4.2. Data transformation and enrichment

Having all the raw data in our architecture, we will explain how the knowledge extraction process would be. This will be done using the data explained above and analysing the relationships that can be obtained from the dataset after pre-processing, cleaning and enrichment of the data. We have aimed to study the co-morbidity within the different hospital data and the complete dataset. For this purpose, it has been necessary to carry out the steps described in Section 3.3.

In the first step, we have obtained the data from the hospitals and transformed all time-related variables as explained in Section 3.3.1 and the example in Table 1. In addition, some other variables have been processed by normalizing or changing their formats to improve the performance of the algorithms.

In the next step, the enrichment phase has been applied, obtaining information from external sources such as those discussed in Table 2 and an example can be seen in Table 3. This is necessary because all databases come coded with codes from different external sources. Because of this, the characteristics of the datasets

Table 5
Features after processing phase.

Feature	Group	Origin
Age	User data	Pre-processing phase
Origin	User data	Enrichment phase
Health (public or private)	User data	Raw data
...	User data	17 features more
Name of hospital	Hospital data	Raw data
Department	Hospital data	Enrichment phase
Type of service	Hospital data	Enrichment phase
Diagnosis (main and 19 supplementary)	Hospital admission	Enrichment phase
Admission service	Hospital admission	Enrichment phase
Reason for discharge	Hospital admission	Enrichment phase
Type of diagnosis	Hospital admission	Enrichment phase
Type of diagnosis(1 per diagnosis)	Hospital admission	Enrichment phase
Disgnostic factors(3 per diagnosis)	Hospital admission	Enrichment phase
...	Hospital admission	56 features more

have to be extracted through external APIs, which has allowed us to add information related to diagnoses, interventions, elements such as hospitals, nationalities etc., thus improving the information contained in the system. An example can be seen with the diagnostic variable of the execution of this type of process in Table 3.

The dataset obtained from the set has various variables that can be grouped into three main groups. These would be the user's characteristics such as age, origin, history, etc. Others would be the information corresponding to the hospital and its services such as name, location, service, department, etc. Finally, we would have the information related to the entry of the patient in the system that would be the data related to the admission in the hospital or system. These are the ones related to the diagnosis, tests, etc., carried out in that admission or passage through the hospital. In Table 5 we can see some of the characteristics and the origin of the data, whether it has been data generated by processing, whether it has been data extracted or processed by the enrichment phase or whether it is as it appeared in the raw data.

As can be seen, most of these variables are not fully modelled and interpreted. This occurs because, on the one hand, we have many numerical variables whose nature is difficult to represent in the algorithms to be used (association rules). For example, when discretizing these variables to create rules, we must create intervals that do not represent the nature of the variable, so we lose part of the knowledge that we could extract from the data set. To solve the handling of these types of variables, we are going to use the algorithm described in Section 3.3.3, in which, depending on the variables, we can apply labels automatically, using expert knowledge or implementing a function that generates the distribution of the degree of membership. With this processing, we can model these variables more correctly, and we can also label them with fuzzy labels so that the results of our algorithms are more interpretative for end-users.

As an example of some of these variables, we can see in Fig. 5 how the patient's age variable has been transformed into fuzzy labels with different degrees of membership. This division has been made with respect to the segmentation presented in [60] and allows us to represent the nature of the variable and its final interpretation better, as can be seen in some of the results obtained.

Another of the fuzzified variables is the length of time spent in the hospital. This is expressed as the number of days a patient is admitted to a hospital ward. In this, we have used seven linguistic

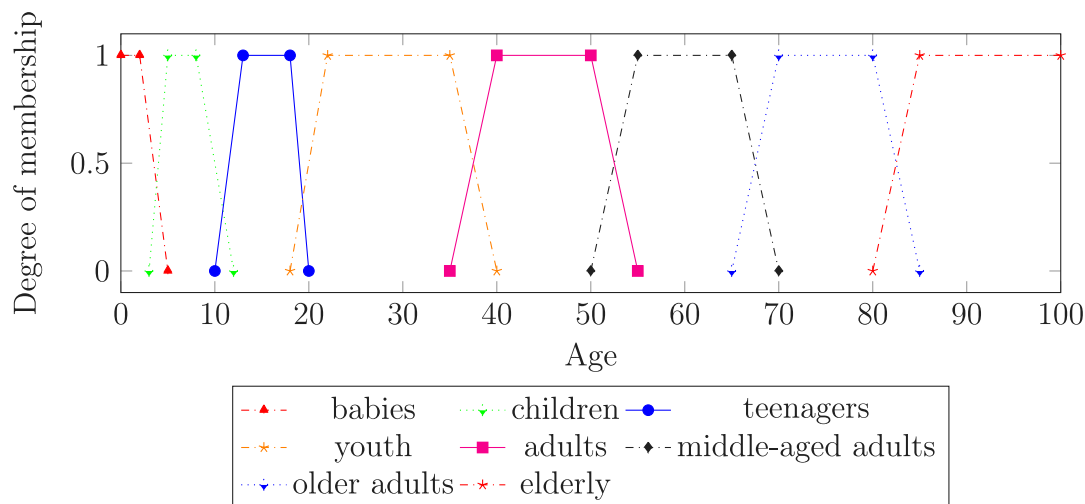


Fig. 5. Example of fuzzification of the Age feature.

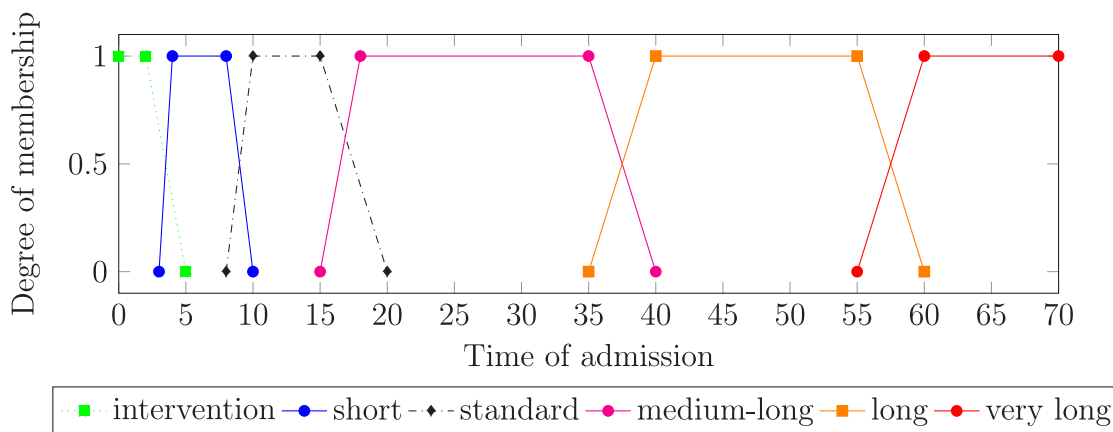


Fig. 6. Distribution of the fuzzy labels as a function of the variable time of admission.

labels to express what the stay is like; from the label intervention related to stays of 1 to 3 days, mostly after a simple surgical intervention, childbirth, etc., to very long stays which can last more than two and a half months. These labels can be seen in Fig. 6. If the patient is admitted to the ICU, the variable that stores this information is different, although it has also been fuzzified with the values shown in Fig. 4. The information stored in other variables enables us to extract more knowledge about the database. For example, from diagnoses such as drug dependence, we can infer that the person suffers from an addiction; from problems of nicotine dependence or smoking that the person is a smoker and from other diagnoses, we can infer obesity, alcoholism, etc.

On the other hand, we can obtain levels of complexity or the severity of the person. For example, to infer the level of complexity of a patient’s diagnosis, this can be obtained according to the number of diagnoses carried out on admission. A patient may have only one primary diagnosis or up to 20. Of these, we have several kinds, diagnoses obtained on admission (new diagnoses obtained because of their symptoms) or diagnoses that the patient already had. Thanks to this information, we can conclude that the number of diagnoses obtained when a patient is admitted is determined by the number of new diagnoses obtained in that admission. In this way, we have defined their fuzzy labels, as can be seen in Fig. 7. In addition to these examples, we have fuzzified more variables such as the severity of the patient according to the time of admission; we have also processed the ICU times,

if the patient is frequently admitted (who has frequent monthly admissions), the weight of the newborns, etc.

4.3. Efficiency analysis and comparison with the sequential approach

As previously mentioned, the proposal was implemented using Spark which enables MapReduce implementation in large data sets. We have carried out different tests in order to be able to analyse the improvement obtained by processing and fuzzifying these data using this framework. The experimental evaluation was carried out on a server cluster with 3 nodes with a total of 102 cores and 320 GB of RAM, running on an operating system with Ubuntu 20. The Spark version was 2.3 using a fully distributed mode with Ambari.

An analysis of the data fuzzification process has been carried out. Depending on the number of processors, different percentages of improvement can be achieved (regarding the computation time see Fig. 8). With the aim of analysing the speed up and the efficiency [61–63] according to the number of cores, we have used the known measure of speed up defined as [63,64]

$$S_n = T_1/T_n \tag{1}$$

where T_1 is the time of the sequential algorithm and T_n is the execution time of the parallel algorithm using several cores. The efficiency [61–63] can be defined in a similar way as

$$E_n = S_n/n = T_1/(n \cdot T_n) \tag{2}$$

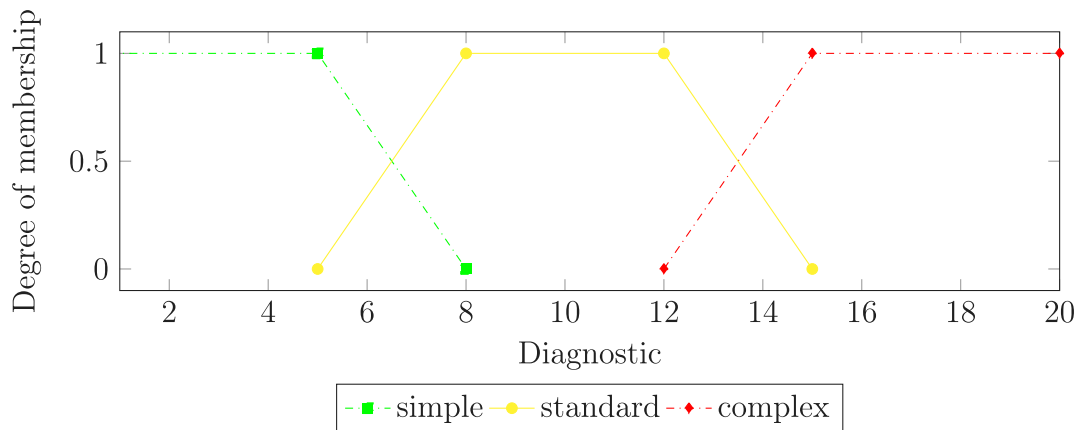


Fig. 7. Distribution of the fuzzy labels as a function of the variable *diagnostic*.

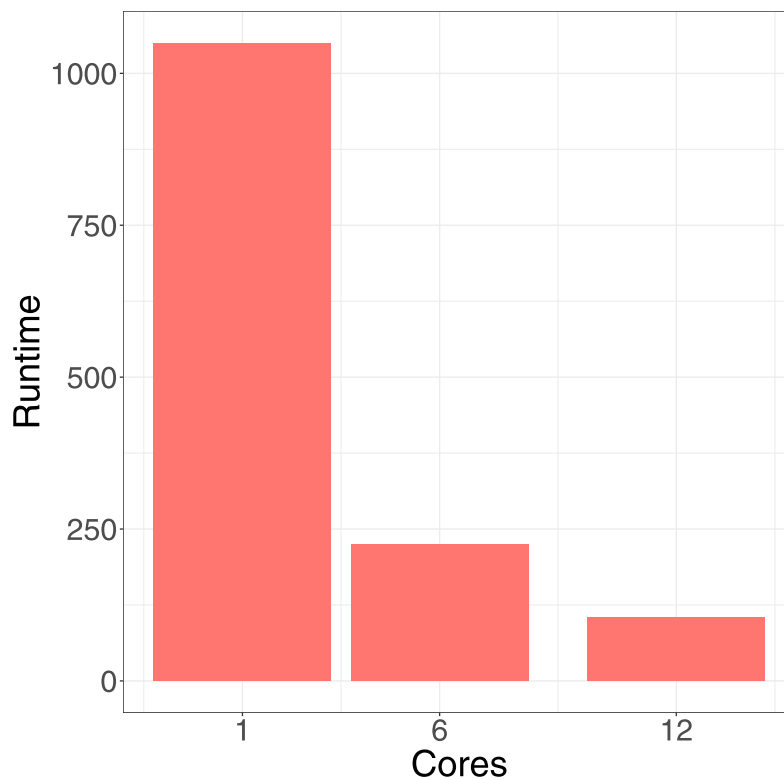


Fig. 8. Speed up of fuzzification process versus number of processing cores.

Figs. 9 and 10 show that the efficiency and speed up are improved as the numbers of cores increases, even when are not optimal. The decrease in efficiency is due to the cores workloads and the network congestion used for the communication amongst the cores.

Using more computational capacity does not represent an improvement from 12 cores because Spark does not divide the data set further due to its size. This implies that if we had more data this process would be able to maintain its efficiency.

In addition, Fig. 9 shows the speedup and evolution of the execution times consumed by the proposal. In this figure, the greatest reduction in calculation time is achieved when the number of processors is 12 is clearly observed.

On the other hand, the use of the fuzzy association rules algorithm has been found to improve the sequential version

significantly. In this case, due to the complexity of the process, the algorithm does use the full capacity of the cluster (102 cores).

It can be seen in Fig. 11 how the sequential algorithm of association rules has a worse efficiency and much higher execution times for the two datasets and the experiments with both datasets together. Furthermore, in Fig. 12 we can also see how the behaviour of the algorithm improves from 1 core (sequential) to using the distributed algorithm with more resources (from 24 to 102 cores).

This distributed processing using Big Data improves the processing and computational capacity of large massive data sets. Improving some state-of-the-art proposals such as [47,65], it can be applied to large datasets from hospitals, clinics etc. or even the

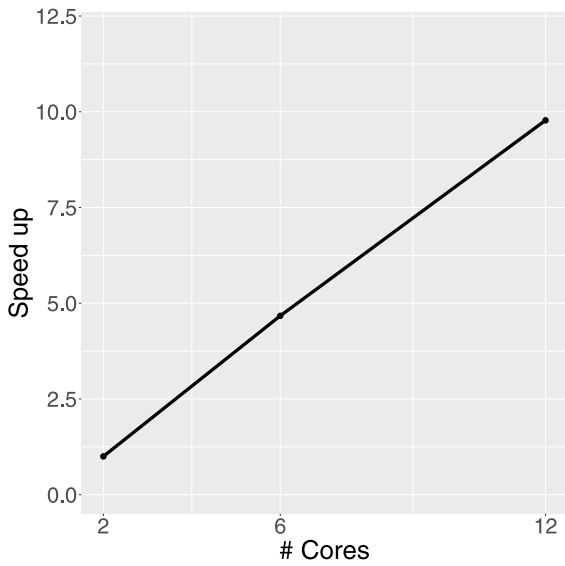


Fig. 9. Time in seconds with different core configurations of the fuzzification algorithm.

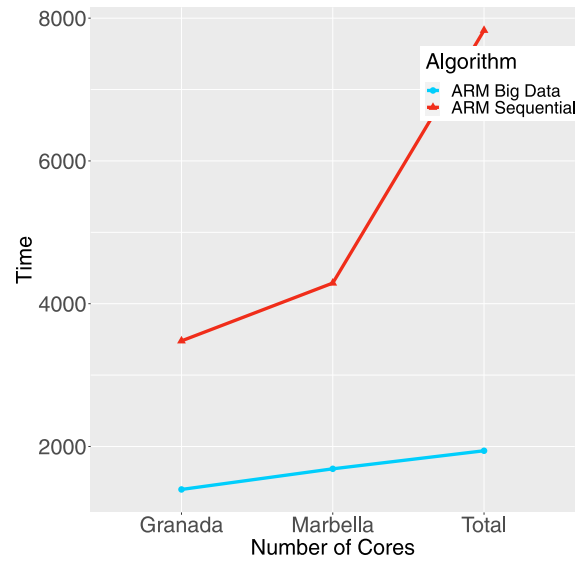


Fig. 11. Execution time of the association rule extraction algorithm with data from each hospital and with all compared to the sequential version (1 core) and in Big Data.

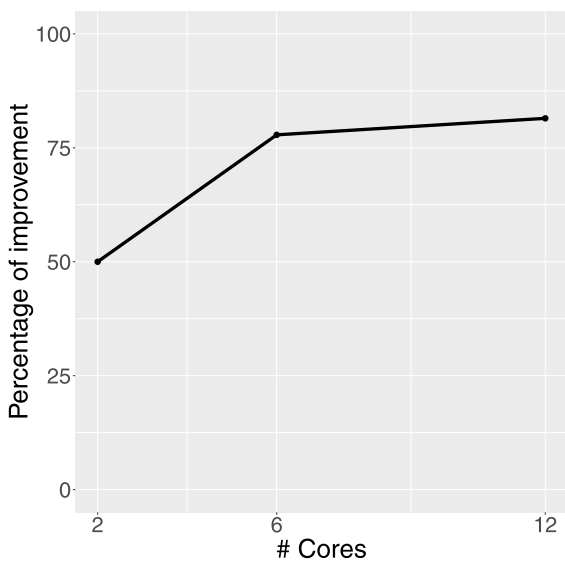


Fig. 10. Efficiency of fuzzification process versus number of processing cores of the rule mining algorithm.

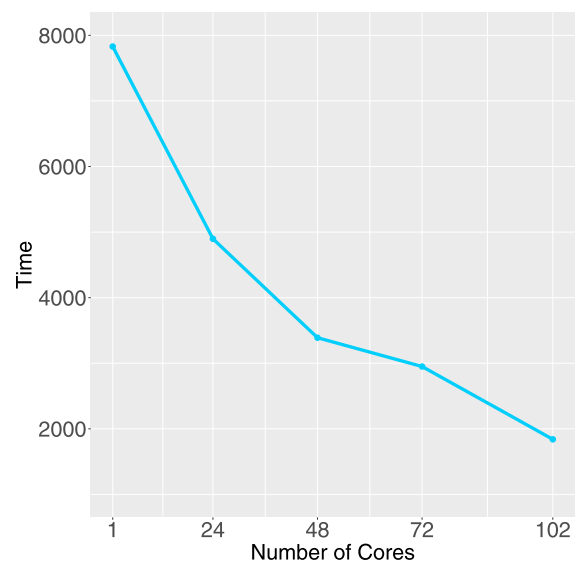


Fig. 12. Efficiency of association rule mining algorithm process versus number of processing cores (1 core → sequential).

combination of different hospital data sources as has been done in the experimentation.

4.4. Results and discussion

Association rule algorithms have been applied to this processed and enriched data. These algorithms are implemented in Spark to be able to process large data sets because, as we have seen, we have a large number of records with a very high number of features.

The experiments have been applied for different threshold configurations. In particular, we show here the results obtained when the minimum support was set to 0.1 and the minimum confidence to 0.8. We also considered a set of ten equidistributed α -cuts.

The support and confidence thresholds have been set higher than usual due to the high number of resulting rules obtained for

lower values. In Fig. 4, the relationship between the amount of support and the number of rules obtained is shown. As can be seen, the number of rules increases as the support increases (see Fig. 13).

The obtained set of rules has enabled us to discover hidden patterns in the relationship between the different diagnoses that appear in the patients and relate these to the patient's characteristics. This type of relationship allows us to study the co-morbidity of the data contained in the system.

Having a look at the discovered patterns, we can highlight different rules. For example:

$$\{Age = long, Time\ of\ admission = medium_long\} \rightarrow \{Reason_medical_discharge = death, Patientseverity = high\} \quad (3)$$

This rule shows how the relationship between elderly patients with a medium to long admission times and that these patients

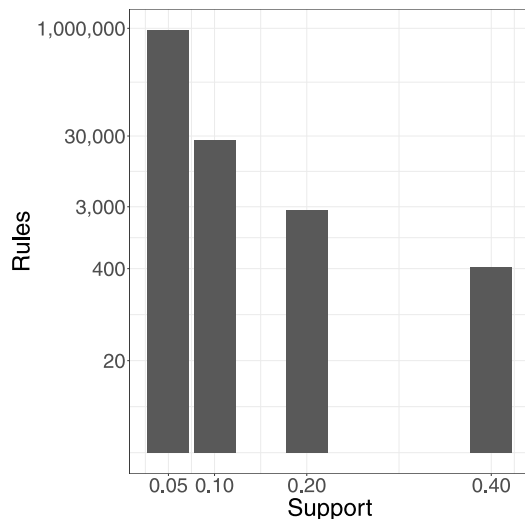


Fig. 13. Number of rules obtained with different parameters of the extraction algorithm.

are seriously ill or die is described. Therefore, we could determine that age and medium-long stays have a grave prognosis.

On the other hand, some rules have been selected, where we can see different behaviour depending on the hospital. For example, this rule obtained in Granada:

$$\{Age = Adult, season = winter, D1 = S83\} \rightarrow \{Diagnostic = simple, Time_Admission = intervention\} \quad (4)$$

where S83 is equivalent to *dislocation and sprain of joints and ligaments of knee*. This rule provides important information about the relationship between winter knee traumas in Granada; probably because winter sports such as snowboarding and skiing are practised, there are many knee injuries [66]. Moreover, thanks to the diffuse labels we can see how the time of admission of this type of lesions is very short and simple to diagnose because they are not usually associated with other derived diagnoses.

On the other hand, by analysing these resulting rules, we can study the co-morbidity of the different diagnoses. The following rules are obtained by unifying all the data from the two hospitals. In this case, we can see how some diseases such as diabetes generate more complex diagnoses or longer stays. In the first, we can see how the diagnosis of diabetes with cardiac or respiratory problems implies a complex diagnosis and severe patients.

$$\{diabetes, Diseasesofthecirculatorysystem\} \rightarrow \{Diagnostic = complex, Time_Admission = standard\}$$

The following association rule shows how the diagnosis of alcoholism and drug dependence is related to digestive problems and frequently admitted patients.

$$\{Drugdependence, Alcoholism\} \rightarrow \{Diseasesofthedigestivesystem, frequency = high\}$$

In addition, we have studied, in particular, the COVID19 pandemic-related diagnoses since we have 2020 data from both hospitals in the records of our set. We can find the following rule with a confidence of 0.83, which allows us to relate patients with respiratory diseases who had COVID, needed a ventilator.

$$\{COVID19, Diseasesoftherespiratorysystem\} \rightarrow \{Time_Admission = long, Time_Admission = verylong, Dependenceonrespirator, severity = high\}$$

Specifically, this type of patient is admitted for a long stay and has a complex prognosis due to the fact that when assisted ventilation is necessary, the after-effects and recovery are very complex.

5. Conclusions

The discovery and exploitation of information collected from hospitals have attracted attention due to their economic and health impact in the last decade. Big Data offers a suitable framework for the efficient implementation of analysis techniques capable of handling large amounts of data, especially those produced in healthcare systems. In addition, the use of fuzzy logic can improve the interpretability of collected data, offering improved results and interpretation to end-users.

This study has been aimed at the extraction of hidden knowledge from medical records and its analysis and interpretation. For such a purpose, we have implemented a data mining system using the Big Data framework, and we have applied it to different data sets collected from two hospitals in the south of Spain. In particular, we have enriched some features and applied a fuzzification algorithm to improve the performance of data mining techniques, such as association rules. The whole system has been deployed using the Spark platform to analyse such an amount of data generated by the different systems in the hospitals.

The experimentation was conducted with real data from two hospitals to demonstrate the feasibility of our proposal. Our results show the capability of our proposal, discovering interesting rules such as “*alcoholism and drug-dependence is related to digestive-problems and frequent-patients*”, and “*the diagnosis of diabetes with cardiac or respiratory-problems implies a complex-diagnosis and severe-patients*”, which can be used by end-users to predict and prevent possible diseases, discover relevant relationships between different features and analyse co-morbidity.

The presented system presents significant advantages for the process of analysing medical records in hospitals. This is possible thanks to the distributed computing capability and the innovative data processing process using expert knowledge. This can enable the system to use the data of a hospital or a set of hospitals to process and extract the data contained in the medical records of their patients. It has limitations in that it is adapted to the Spanish standard medical records system, as well as using expert knowledge from Spanish data sources due to the use of this standard.

This is why this research constitutes a starting point, opening up new research lines for the future. The following step consists of using external sources of knowledge such as SNOMED CT (Systematized Nomenclature of Medicine Clinical Terms). This integration will considerably improve the enrichment process of the feature. Another future enhancement concerns visualizing the results which should be more informative and complete for end-users.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The research reported in this paper was partially supported by the BIGDATAMED project, which has received funding from the Andalusian Government (Junta de Andalucía) under grant agreement No P18-RT-1765. In addition, this work has been partially supported by the Ministry of Universities through the EU-funded margarita salas programme NextGenerationEU.

References

- [1] A.R. Feinstein, The pre-therapeutic classification of co-morbidity in chronic disease, *J. Chronic Dis.* 23 (7) (1970) 455–468.
- [2] P. Fraccaro, D. O'Sullivan, P. Plastiras, H. O'Sullivan, C. Dentone, A. Di Biagio, P. Weller, Behind the screens: Clinical decision support methodologies – A review, *Health Policy Technol.* 4 (1) (2015) 29–38.
- [3] J. Chen, W. Wei, C. Guo, L. Tang, L. Sun, Textual analysis and visualization of research trends in data mining for electronic health records, *Health Policy Technol.* 6 (4) (2017) 389–400.
- [4] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, *Artificial Intelligence* 267 (2019) 1–38.
- [5] B. Kim, R. Khanna, O.O. Koyejo, Examples are not enough, learn to criticize! criticism for interpretability, *Adv. Neural Inf. Process. Syst.* 29 (2016).
- [6] A. Subasi, Chapter 3 - machine learning techniques, in: A. Subasi (Ed.), *Practical Machine Learning for Data Analysis using Python*, Academic Press, 2020, pp. 91–202.
- [7] L. Zadeh, Fuzzy sets, *Inf. Control* 8 (1965) 338–353.
- [8] C. Fernandez-Basso, M.D. Ruiz, M.J. Martín-Bautista, Extraction of association rules using big data technologies, *Int. J. Des. Nature Ecodyn.* 11 (3) (2016) 178–185.
- [9] P.Y. Taşer, K.U. Birant, D. Birant, Multitask-based association rule mining, *Turk. J. Electr. Eng. Comput. Sci.* 28 (2) (2020) 933–955.
- [10] H. Li, Y. Wang, D. Zhang, M. Zhang, E.Y. Chang, Pfp: parallel fp-growth for query recommendation, in: *Proceedings of the 2008 ACM Conference on Recommender Systems*, 2008, pp. 107–114.
- [11] K. Koperski, J. Han, Discovery of spatial association rules in geographic information databases, in: *International Symposium on Spatial Databases*, Springer, 1995, pp. 47–66.
- [12] D.d. Castro Rodrigues, V. Siqueira, F. Tavares, M. Lima, F. Oliveira, L. Osco, W. Junior, R. Costa, R. Barbosa, Discovering associative patterns in healthcare data, in: *Proceedings of Sixth International Congress on Information and Communication Technology*, Springer, 2022, pp. 371–379.
- [13] C. Ordonez, N. Ezquerra, C.A. Santana, Constraining and summarizing association rules in medical data, *Knowl. Inf. Syst.* 9 (3) (2006) 1–2.
- [14] I.R. Mewes, H. Jenzer, F. Einsele, A study about discovery of critical food consumption patterns linked with lifestyle diseases for swiss population using data mining methods., in: *HEALTHINF*, 2021, pp. 30–38.
- [15] M. Delgado, N. Marín, D. Sánchez, M. Vila, Fuzzy association rules: General model and applications, *IEEE Trans. Fuzzy Syst.* 11 (2) (2003) 214–225.
- [16] Z. Yu, B.C. Fung, F. Haghighat, Extracting knowledge from building-related data. a data mining framework, in: *Building Simulation*, Vol. 6, Springer, 2013, pp. 207–222.
- [17] Z.J. Yu, F. Haghighat, B.C. Fung, Advances and challenges in building engineering and data mining applications for energy-efficient communities, *Sustainable Cities Soc.* 25 (2016) 33–38.
- [18] M. Molina-Solana, M. Ros, M.D. Ruiz, J. Gómez-Romero, M. Martín-Bautista, Data science for building energy management: a review, *Renew. Sustain. Energy Rev.* 70 (2017) 598–609.
- [19] C. Fan, F. Xiao, Mining gradual patterns in big building operational data for building energy efficiency enhancement, *Energy Procedia* 143 (2017) 119–124.
- [20] J. Davis, A. Clark, Data preprocessing for anomaly based network intrusion detection: A review, *Comput. Security* 30 (6–7) (2011) 353–375.
- [21] A. Chandra Pandey, D. Singh Rajpoot, M. Saraswat, Twitter sentiment analysis using hybrid cuckoo search method, *Inf. Process. Manage.* 53 (4) (2017) 764–779.
- [22] J. Jeon, C. Lee, Y. Park, How to use patent information to search potential technology partners in open innovation, *J. Intellect. Property Rights* 16 (5) (2011) 385–393.
- [23] H. Amirkolaei, H. Arefi, Height estimation from single aerial images using a deep convolutional encoder-decoder network, *ISPRS J. Photogramm. Remote Sens.* 149 (2019) 50–66.
- [24] S. Zhang, Q. Shen, C. Nie, Y. Huang, J. Wang, Q. Hu, X. Ding, Y. Zhou, Y. Chen, Hyperspectral inversion of heavy metal content in reclaimed soil from a mining wasteland based on different spectral transformation and modeling methods, *Spectrochim. Acta - Part A: Mol. Biomol. Spectrosc.* 211 (2019) 393–400.
- [25] L. Sun, X. Zhang, J. Xu, S. Zhang, An attribute reduction method using neighborhood entropy measures in neighborhood rough sets, *Entropy* 21 (2) (2019).
- [26] L. Zhang, H. Sun, Z. Rao, H. Ji, Hyperspectral imaging technology combined with deep forest model to identify frost-damaged rice seeds, *Spectrochim. Acta - Part A: Mol. Biomol. Spectrosc.* 229 (2020).
- [27] A. Beam, I. Kohane, Big data and machine learning in health care, *JAMA - J. Amer. Med. Assoc.* 319 (13) (2018) 1317–1318.
- [28] C. Lee, H.-J. Yoon, Medical big data: Promise and challenges, *Kidney Res. Clin. Pract.* 36 (1) (2017) 3–11.
- [29] M. Sabzevari, E. Imani, Separation of movement direction concepts based on independent component analysis algorithm, linear discriminant analysis, deep belief network, artificial and fuzzy neural networks, *Biomed. Signal Process. Control* 62 (2020).
- [30] M. Khan, I. Lali, A. Rehman, M. Ishaq, M. Sharif, T. Saba, S. Zahoor, T. Akram, Brain tumor detection and classification: A framework of marker-based watershed algorithm and multilevel priority features selection, *Microsc. Res. Tech.* 82 (6) (2019) 909–922.
- [31] S. Vajda, A. Karargyris, S. Jaeger, K. Santosh, S. Candemir, Z. Xue, S. Antani, G. Thoma, Feature selection for automatic tuberculosis screening in frontal chest radiographs, *J. Med. Syst.* 42 (8) (2018).
- [32] C. Sakar, G. Serbes, A. Gunduz, H. Tunc, H. Nizam, B. Sakar, M. Tutuncu, T. Aydin, M. Isenkul, H. Apaydin, A comparative analysis of speech signal processing algorithms for parkinson's disease classification and the use of the tunable Q-factor wavelet transform, *Appl. Soft Comput.* 74 (2019) 255–263.
- [33] J. Chen, B. Hu, P. Moore, X. Zhang, X. Ma, Electroencephalogram-based emotion assessment system using ontology and data mining techniques, *Appl. Soft Comput.* 30 (2015) 663–674.
- [34] W. Sun, Z. Cai, Y. Li, F. Liu, S. Fang, G. Wang, Data processing and text mining technologies on electronic medical records: A review, *J. Healthcare Eng.* 2018 (2018).
- [35] S. Maitra, N. Akter, A. Zahan Mithila, T. Hossain, M. Shafiqul Alam, Apriori-backed fuzzy unification and statistical inference in feature reduction: An application in prognosis of autism in toddlers, *Adv. Intell. Syst. Comput.* 1299 AISC (2021) 233–254.
- [36] K. Majumdar, Human scalp EEG processing: Various soft computing approaches, *Appl. Soft Comput.* 11 (8) (2011) 4433–4447.
- [37] S. Lakshmanaprabu, S. Mohanty, S. S., S. Krishnamoorthy, J. Uthayakumar, K. Shankar, Online clinical decision support system using optimal deep neural networks, *Appl. Soft Comput.* 81 (2019).
- [38] A. Nikfarjam, A. Sarker, k. O'Connor, R. Ginn, G. Gonzalez, Pharmacovigilance from social media: Mining adverse drug reaction mentions using sequence labeling with word embedding cluster features, *J. Amer. Med. Inform. Assoc.* 22 (3) (2015) 671–681.
- [39] R. Bauder, T. Khoshgoftaar, N. Seliya, A survey on the state of healthcare upcoding fraud analysis and detection, *Health Services Outcomes Res. Methodol.* 17 (1) (2017) 31–55.
- [40] L. Silva, A. Santos, R. Bravo, A. Silva, D. Muchaluat-Saade, A. Conci, Hybrid analysis for indicating patients with breast cancer using temperature time series, *Comput. Methods Programs Biomed.* 130 (2016) 142–153.
- [41] S. Chakraborty, K. Mali, Fuzzy electromagnetism optimization (FEMO) and its application in biomedical image segmentation, *Appl. Soft Comput.* 97 (2020).
- [42] F. Gargiulo, S. Silvestri, M. Ciampi, G. De Pietro, Deep neural network for hierarchical extreme multi-label text classification, *Appl. Soft Comput.* 79 (2019) 125–138.
- [43] R. Catelli, F. Gargiulo, V. Casola, G. De Pietro, H. Fujita, M. Esposito, Crosslingual named entity recognition for clinical de-identification applied to a COVID-19 Italian data set, *Appl. Soft Comput.* 97 (2020).
- [44] A. Ponnmalar, V. Dhanakoti, An intrusion detection approach using ensemble support vector machine based chaos game optimization algorithm in big data platform, *Appl. Soft Comput.* 116 (2022).
- [45] W. Ai, K. Li, K. Li, An effective hot topic detection method for microblog on spark, *Appl. Soft Comput.* 70 (2018) 1010–1023.
- [46] S. Malla, A. P.J.A., COVID-19 outbreak: An ensemble pre-trained deep learning model for detecting informative tweets, *Appl. Soft Comput.* 107 (2021).
- [47] M. Delgado, D. Sánchez, M.-A. Vila, Acquisition of fuzzy association rules from medical data, in: S. Barro, R. Marín (Eds.), *Fuzzy Logic in Medicine*, Physica-Verlag HD, Heidelberg, 2002, pp. 286–310.
- [48] F. Ali, S. Islam, D. Kwak, P. Khan, N. Ullah, S.-J. Yoo, K. Kwak, Type-2 fuzzy ontology-aided recommendation systems for IoT-based healthcare, *Comput. Commun.* 119 (2018) 138–155.
- [49] F. Goodarziyan, H. Hosseini-Nasab, J. Muñuzuri, M.-B. Fakhrazad, A multi-objective pharmaceutical supply chain network based on a robust fuzzy model: A comparison of meta-heuristics, *Appl. Soft Comput.* 92 (2020).
- [50] C. Fernandez-Basso, M.D. Ruiz, M.J. Martín-Bautista, A fuzzy mining approach for energy efficiency in a big data framework, *IEEE Trans. Fuzzy Syst.* 28 (11) (2020) 2747–2758.
- [51] B. Malysiak-Mrozek, M. Stabla, D. Mrozek, Soft and declarative fishing of information in big data lake, *IEEE Trans. Fuzzy Syst.* 26 (5) (2018) 2732–2747.
- [52] J. Kooij, Adult ADHD: Diagnostic Assessment and Treatment, 2014, pp. 1–292.
- [53] G. Mahani, M.-R. Pajoohan, Predicting lab values for gastrointestinal bleeding patients in the intensive care unit: A comparative study on the impact of comorbidities and medications, *Artif. Intell. Med.* 94 (2019) 79–87.
- [54] E. Saleh, J. Błaszczyński, A. Moreno, A. Valls, P. Romero-Aroca, S. de la Riva-Fernández, R. Słowiński, Learning ensemble classifiers for diabetic retinopathy assessment, *Artif. Intell. Med.* 85 (2018) 50–63.

- [55] J. Yoon, D. Jeong, C.-H. Kang, S. Lee, Forensic investigation framework for the document store NoSQL DBMS: MongoDB as a case study, *Digital Invest.* 17 (2016) 53–65.
- [56] M.D. Ruiz, J. Gomez-Romero, C. Fernandez-Basso, M.J. Martin-Bautista, Big data architecture for building energy management systems, *IEEE Trans. Ind. Inf.* (2021).
- [57] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M.J. Franklin, S. Shenker, I. Stoica, Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing, in: *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, USENIX Association, 2012, p. 2.
- [58] C. Fernandez-Basso, M.D. Ruiz, M.J. Martin-Bautista, Spark solutions for discovering fuzzy association rules in Big Data, *Internat. J. Approx. Reason.* 137 (2021) 94–112.
- [59] C. Fernandez-Basso, M.D. Ruiz, M.J. Martin-Bautista, Fuzzy association rules mining using spark, in: *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Springer, 2018, pp. 15–25.
- [60] S.I. Vuik, E. Mayer, A. Darzi, A quantitative evidence base for population health: applying utilization-based cluster analysis to segment a patient population, *Popul. Health Metr.* 14 (1) (2016) 1–9.
- [61] V.P. Kumar, A. Gupta, Analyzing scalability of parallel algorithms and architectures, *J. Parallel Distrib. Comput.* 22 (3) (1994) 379–391.
- [62] A.Y. Grama, A. Gupta, V. Kumar, Isoefficiency: Measuring the scalability of parallel algorithms and architectures, *IEEE Parallel Distrib. Technol. Syst. Appl.* 1 (3) (1993) 12–21.
- [63] C. Barba-González, J. García-Nieto, A. Benítez-Hidalgo, A.J. Nebro, J.F. Aldana-Montes, Scalable inference of gene regulatory networks with the spark distributed computing platform, in: J. Del Ser, E. Osaba, M.N. Bilbao, J.J. Sanchez-Medina, M. Vecchio, X.-S. Yang (Eds.), *Intelligent Distributed Computing XII*, Springer International Publishing, Cham, 2018, pp. 61–70.
- [64] M.D. Ruiz, J. Gomez-Romero, C. Fernandez-Basso, M.J. Martin-Bautista, Big data architecture for building energy management systems, *IEEE Trans. Ind. Inf.* (2021).
- [65] J. Calero, G. Delgado, M. Sánchez-Marañón, D. Sánchez, M.A.V. Miranda, J. Serrano, An experience in management of imprecise soil databases by means of fuzzy association rules and fuzzy approximate dependencies, in: *ICEIS 2004, Proceedings of the 6th International Conference on Enterprise Information Systems*, Porto, Portugal, April 14–17, 2004, 2004, pp. 138–146.
- [66] M.J. Jordan, P. Aagaard, W. Herzog, Anterior cruciate ligament injury/reinjury in alpine ski racing: a narrative review, *Open Access J. Sports Med.* 8 (2017) 71.

4.4. AIMDP: Plataforma de datos basada en Big Data y IA para la gestión eficiente y análisis de información en entornos heterogéneos.

- Referencia:0167-739X
- Estado: Aceptado
- Factor de Impacto:Q1
- Categoría: COMPUTER SCIENCE, THEORY METHODS - SCIE
- DOI: <https://doi.org/10.1016/j.future.2023.02.002>
- Revista/Editorial:Future Generation Computer Systems



AIMDP: An Artificial Intelligence Modern Data Platform. Use case for Spanish national health service data silo

Alberto S. Ortega-Calvo^b, Roberto Morcillo-Jimenez^b, Carlos Fernandez-Basso^{a,b,*}, Karel Gutiérrez-Batista^b, Maria-Amparo Vila^b, Maria J. Martin-Bautista^b

^a Causal Cognition Lab, Division of Psychology and Language Sciences, University College London, London, United Kingdom

^b Department of Computer Science and Artificial Intelligence, University of Granada, 18071, Granada, Spain



ARTICLE INFO

Article history:

Received 14 July 2022

Received in revised form 30 January 2023

Accepted 2 February 2023

Available online 6 February 2023

Keywords:

Data platform

Artificial Intelligence

Intelligent data analysis

Big data

Medical informatics

Data silo

ABSTRACT

The huge amount of data being handled today in any environment, such as energy, economics or healthcare, makes data management systems key to extracting information, analysing and creating more efficient daily processes in these environments. However, the inability of current systems to take advantage of the data generated can waste good opportunities for analysing and extracting information from the data. Modern data platforms (MDP) appear suitable for supporting management systems and are able to perform future prospective analyses. This paper presents a data platform called Artificial Intelligence Modern Data Platform (AIMDP), based on Big Data, artificial intelligence for management, and efficient data handling. The different components of AIMDP intervene in the data acquisition phase and implement algorithms capable of analysing massive data collected from heterogeneous sources. In addition, the entire platform is geared towards data management and exploitation with a layer of security and data governance that allows the integrity and privacy of the databases to be maintained. The proposed platform is designed to be used by users who are not experts in data science. To this end, it implements a user-oriented workflow that has effectively been introduced in a use case of two Spanish hospitals to extract knowledge from their historical data, which had been siloed and had never been explored by any hospital researchers or doctors.

© 2023 Published by Elsevier B.V.

1. Introduction

Nowadays, companies and researchers are increasingly interested in developing robust systems that efficiently enable data storage and knowledge discovery. Modern data platforms (MDP) are computer-based systems which enable organizations to become data-driven [1].

These systems provide developers and end-users with tools for storing, processing and analysing complex data from heterogeneous sources and domains. Modern data platforms have been widely used in different domains such as smart cities, financial technology, materials science, energy, and many others [1–6].

Building these sorts of systems raises many challenges, such as processing and analysing heterogeneous data sources, providing end-users (mostly non-expert users) with a user-friendly tool, treatment of massive data and data governance. These challenges become even more complex when the data to be analysed is electronic health records (EHRs). It is mainly because medical

databases can contain data from different sources, which results in the data being mostly semi-structured and unstructured. Medical databases store historical data about patients. We can find very few studies in the literature related to developing MDP oriented to healthcare [7–11]. We summarize the main drawbacks to address for implementing these kinds of systems in the healthcare domain below:

- EHRs are often stored in data silos, where data comes from different sources, and the data are mostly comprised of semi-structured and unstructured data.
- Providing end-users (mostly users without knowledge about data science) with user-friendly services and tools, enabling end-users to focus only on the desired analysis.
- The enormous amount of data collected in medical data silos makes processing and analysing difficult.
- Establishing appropriate data governance policies in order to provide better availability, integrity, usability and security of the data.

Considering the problems above, it would be useful to endow healthcare-related centres and users with powerful tools to manage EHRs.

* Corresponding author at: Department of Computer Science and Artificial Intelligence, University of Granada, 18071, Granada, Spain.

E-mail addresses: cjferba@decsai.ugr.es, carlos.basso@ucl.ac.uk (C. Fernandez-Basso).

Our study aims to develop a multi-purpose MDP-based architecture that enables professionals in many areas of knowledge who are not experts in data science to handle large amounts of data flexibly and efficiently. The tool would allow end-users to perform more detailed analyses of their data. To this end, this study proposes a new architecture called an Artificial Intelligence Modern Data Platform (AIMDP). The new architecture integrates the main features of the MDP with Data Science (DS) capabilities. We must remark that our research group has already implemented the algorithms comprised in the platform [12–17]. These capabilities are based on supervised and unsupervised techniques in a distributed environment (Big Data).

Although AIMDP is able to work with data of different natures, to showcase the feasibility of the proposed data platform, we have conducted a real-world use case using two healthcare data sets from two hospitals of the Spanish national health system service. Since AIMDP allows end-users to process and analyse health-related data, it could facilitate possible diagnostics and analysis, the subsequent treatment, and the premature prevention of potential diseases. Following on, we summarize the main contributions of this paper:

- **Heterogeneous data sources** - The proposed architecture can handle semi-structured and unstructured data from different sources. This feature and a dynamic database structure allow the end-users to analyse data from different research areas without worrying about the structure and provenance of the data.
- **User-friendly** - AIMDP provides end-users with an interactive and intuitive interface. In this way, the tool can be readily used by users who do not have any expertise in data mining (non-expert users).
- **Bigdata-based techniques** - AIMDP enables data processing and analysing through Data Science capabilities. All the algorithms have been implemented using the distributed programming paradigm.
- **Data governance** - The availability, accessibility, integrity, usability and security of the data in the platform are carried out through an innovative security layer that allows the users of the platform to work with sensitive data.

The rest of the paper is structured as follows: a review of previous research related to this topic is presented in Section 2. Sections 3 and 4 presents the proposed data platform (AIMDP) for artificial intelligence applications and how users can use the data platform workflow modules, respectively. Section 5 presents a real-world use case using AIMDP and discusses the obtained results. Finally, in Section 6, the conclusions and future research are presented.

2. Related research

The *platform concept* has been well known since the early 2010s when there were already discussions [18] about the different connotations of this term. The connotation used in this paper is the computational one, which complements the *data platform* term, one of the crucial concepts of this study. AIMDP, the platform that is being introduced in this paper, is a data platform that adopts a Modern Data Architecture schema.

In the related research section, an introduction to the data platform concept is made. The most relevant terms related to this concept, such as *Cloud Computing*, *Big Data* or *Modern Data Architecture*, are described from the perspective of the data platform concept. AIMDP is also detailed within these relevant terms. In addition, some applications of data platforms are mentioned, introducing real examples of previous proposals made by other authors in this field. The application in health care is explained in

detail, as there is a lot of potential in this area, which is the main one on which the use case of Section 5 is based. The capabilities of the recently developed data platforms are discussed separately and compared to those of AIMDP.

The objective of this section is for the reader to be able to firstly understand the concept of data platform and its context and secondly know the main aspects of the proposed AIMDP data platform before the technical description made in Sections 3 and 4. And finally, appreciate the main differences between AIMDP and other data platforms that appear in the scientific literature.

2.1. Data platform

Nowadays, data storing and treatment operations are experiencing a shift [19] from data warehouses to data lakes. This shift is produced by three main factors: 1. The increase in the amount of semi-structured and unstructured data, which are data warehouses unfriendly. 2. The popularity of the micro-service architecture over the monolithic one, which does not have an associated central database. 3. The inability to fulfil the 5 Vs (variety, volume, velocity, veracity and value) requirements.

Regarding the data warehousing and data lake concepts, it is possible to combine these two technologies, even in a cloud-based environment, under the name of Modern Data Platform [19]. This is more capable than a data centre, as it provides more features that address the needs of the new data consumers. Some of these features are, amongst others, that in a data platform, there is no need to provide a schema for the incoming data or the use of Spark [20], and will offer more flexibility since it will be able to deal with large and semi-structured data sets and use multiple parallel tasks for processing [21].

In this paper, the data platform concept is considered one of the main aspects. The standardization of this term is specially promoted by the commercial sector. This is because data platforms as products are a current trend. However, in recent years, a significant number of data platforms have also been developed in the scientific literature (OPEN GOVERNMENT DATA [2], Financial BDP [3], CiDAP [4], CALIBER [7,8], WikiHealth [9], ENTISO-E [22,23], DEGS [24], D2D BIG DATA [25], TELCO [26]).

In the scientific literature, there are different definitions and interpretations of what a data platform is. For [27], a data platform is a web-based interface to collect, store, host and manage datasets with specific uniform organizations. A data platform is capable of storing a large amount of data in an organized structure. Users of the data platform are allowed to contribute, modify, query, analyse and export the datasets, which is crucial for accelerating the evaluation of building performance and energy efficiency. Similarly, in [10], 4 different data platforms are analysed and compared. The authors set the 3 dimensions of data control as the most important parameters to analyse in a data platform: data access (who can access the data), data use (primary and secondary uses of data, who decides these uses, and how they are legitimized and approved through data governance strategies), and data governance (the process by which stewardship responsibilities are conceptualized and carried out).

In the book [19], a data platform is capable of accomplishing all the operations established by [27], which are needed to implement data analysis. For [19,27], the data analysis process is not included as a mandatory feature of data platforms. In other words, the purpose of a data platform is to ingest, store, process, and make data available for analysis. These operations should be independent of the type of data and be performed in a cost-efficient manner. However, other authors that propose implementations of their own data platforms consider this procedure part of a data platform [3,4,9,26]. Moreover, the authors in [3,5,9,19,26] include a layer-based layout, where the data platform is composed of

modules that work independently and share information through their connections, building a workflow. AIMDP also implements a layer-based architecture with independent modules, described in detail in Section 3.

MongoDB [28] includes a definition of a data platform that comprehends most of the ideas introduced by the previously referenced authors. The definition that they propose is: “a data platform is an integrated set of technologies that collectively meet an organization’s end-to-end data needs. It enables the acquisition, storage, preparation, delivery, and governance of your data, as well as a security layer for users and applications”.

2.2. Heterogeneous data integration

As it is discussed in [29], in the current article, structured data is considered as a group of concepts that possesses a set of attributes and relationships with other concepts in the set or source. This is the source type included in standard relational databases. Semi-structured data has some similarities, but each instance of a concept may have different properties or relationships. Examples of this type of data are XML or JSON documents. Unstructured data does not have a structured representation of its concepts, properties, or relationships. Most common examples of these sources are text files or multimedia content such as images [30], audio, or video.

Working with heterogeneous data and its integration constitutes a complicated and well-known problem. In this article, two main kinds of integration are distinguished: 1. Transforming available data into data with similar properties and structure. 2. Allowing the coexistence of different types of data on the platform.

If data scientists and data architects consider following the first approach, it will result in a simpler database system and a possible unification of the data processing techniques of the platform. As included in [29], progress has recently been made to transform unstructured data into semi-structured and structured data. Giving this unstructured data a ‘structure’ can be achieved through techniques that consist of extracting keywords (from sources such as text, videos, or images) and establishing a representation schema and mathematical models. This can simplify the architecture of the data system and enable simultaneous query on heterogeneous data and interest schema property extraction (synonymies, homonymies, hyponymies, overlappings...), thus enriching the data and associated ontologies. However, this requires quality assurance in the metadata and a data pre-processing stage that is not trivial, especially working in a Big Data scenario.

In this platform, data integration is based on the second approach, the coexistence of heterogeneous data sources. Having different types of data available on the platform may require more effort in different aspects, such as database management or algorithm implementation. However, AIMDP grants the possibility of carrying out data processing, knowledge extraction and analysis of results based on the techniques described in Section 4. The implementation is done so that the user can work with unstructured, semi-structured and structured data and will always have available advanced techniques to perform the required experiments.

2.3. Cloud computing in data platforms

Although MongoDB [28] explicitly establishes that this is not a mandatory technology in a data platform, the Cloud Computing paradigm improves its quality. Some of the features [19] that this paradigm introduces in data platforms are elasticity (they grow and shrink as the needs of the data platform change), modularity

(custom resources in storage and computing), availability (resources are available at any time) and faster development (faster introduction of new features in the production environment) of the resources. Being able to use third-party Infrastructure as a Service (IaaS) and Cloud Computing services in a data platform is a great advantage since it allows scalability and flexibility that cannot be guaranteed with a standard server infrastructure. The data platform proposals in [9,25] use third-party services on demand, allowing data platform administrators to increase the storage and compute capabilities of their systems based on their traffic and needs.

In the proposed data platform, regarding the NIST definition of Cloud Computing [31], AIMDP implements a solution for each of the characteristics that define a Cloud Computing infrastructure. It adopts a Software as a Service (SaaS) model since the tools of the data platform are accessible from various client devices through a web-based user interface and the consumer does not manage the underlying cloud infrastructure (network, servers, operating systems, storage, etc.). AIMDP can also be defined as a community cloud because its infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns. A more precise description of how AIMDP’s cloud components are configured is included in Section 3.3.

2.4. Big data platform

The amount of data has increased dramatically in recent decades as society has become more involved with technology, and companies have discovered that producing and storing data can generate considerable profit. Big Data has applications in healthcare, electronics, biology, banking, meteorology and many others fields that affect people’s daily lives. Implementing Big Data architectures for organizations requires obtaining expensive software licenses, preparing a large and sophisticated infrastructure, and paying for experts who know how to use the system and organize and integrate the generated data for analytics [32]. Data platform technologies can be used to help developers to meet these challenges. This is where the term Big Data platform (BDP) comes in, which is capable of providing the same services and features as data platforms but working with massive data sets. Moreover, it is worth mentioning that using the data science tools that computational frameworks such as Apache Spark offer can lead the user to perform advanced knowledge extraction and create meaningful visualizations from these big-sized datasets. These computational frameworks are often based on the concept of distributed learning [33], which establishes that training data is located and processed in different nodes. Those nodes are distributed autonomous computers, and they communicate among themselves over a network. Many relevant algorithms, such as those introduced in [34] or Apache Spark MLlib are based on distributed learning concept. Some of the platforms mentioned previously [4,9,25,26] are defined as Big Data platforms, and they include big data tools like Spark, Hadoop Distributed File System, Hive, OpenStack, etc.

Big Data platform services and tools are usually combined so the user can use them without worrying about what is going on behind the scenes in terms of availability, security or performance. AIMDP, the data platform described in this article, has some of the characteristics of big data platforms, as is described in Section 3.

2.5. Modern data architecture

As mentioned previously, some authors describe a layer-based architecture as the core of their data platforms. In this article, we

include the paradigm Modern Data Architecture (MDA), defined by MongoDB in [28]. It consists of a software architecture that helps to ensure that the user does not have to worry about what is taking place in the core of the system, since it specifies how the data platform and data analysis technologies are structured and carried out.

The AIMDP data platform follows a similar structure to the MDA. This makes the system capable of implementing data analysis and processing techniques using the components of the analytics layer. One of the Big Data characteristics of the data platform is that it implements powerful and big-size-data-proof ML/artificial intelligence(AI) techniques. Something similar appears in [7], where the authors describe the CALIBER phenotyping approach and use it to produce 51 phenotyping algorithms that take part of the CALIBER data platform [8]. This platform includes data resources and tools, algorithms, and specialized infrastructure and supports access to anonymized United Kingdom National Health Service (NHS) data under licence from the Clinical Practice Research Datalink (CPRD) [35], WikiHealth [9] and TELCO [26] also include a differentiated data analysis layer.

On the other hand, due to the generalist and scalable nature of the data platform proposed in this article, it may have been used in different fields of knowledge. AIMDP allows users to load data with different structures since it works with NO-Sql database schemas and implements generic Apache Spark ML algorithms. Including algorithms developed in recent years is also straightforward because the AI module is independent of the rest of the modules of the data platform. This feature is described in more detail in Section 3.

2.6. Potential in health care

Especially after the Covid-19 pandemic outbreak, doctors and other researchers in the medical field have become crucial since it has been proven once again that their research work has helped to save many lives and managed government actions and decisions [36]. The authors in [37,38] show how multi-disciplinarity, involving different subjects of study in one activity (medical and data analysis in this case), can produce real outcomes in people's health.

Another problem related to health data and, especially, big health data is the existence of data silos [10]. These storages contain data collected from health institutions, biomedical and genomics research and so on, the use of which is limited. This means that all the research potential found in the data has been lost. In order to perform knowledge extraction from these data sets, data platforms such as AIMDP are very useful since they allow any user with permission to provide and mine all the available data in the platform.

Multi-disciplinarity and the data silo problem were two of the main reasons that motivated the development of the AIMDP data platform and carry out the use case described in Section 5. Although the system can work with data of a different nature [39], the use case included in this article concerns medical data.

For more than a decade, the researchers of the group have been working with hospitals of the Spanish national health service. Working with doctors introduces the necessity of building a data platform which is user-friendly, and easy-to-use [40]. The objective becomes to be used correctly by users with basic knowledge of data science and programming, so they can still exploit all the features and extract the maximum potential of the data platform [41,42]. This reveals one of the main features of the system, accessibility.

2.7. Data platforms comparison

The characteristics of AIMDP are described in detail in Section 3. The data platform shares points of view with some of the data platforms mentioned in this section but also includes some innovative aspects. A comparison of the most relevant characteristics and capabilities of the data platforms mentioned in this section can be found in Table 1.

The aspects selected for the comparison are the following: 1. **Heterogeneous data collection:** data of different structures (stream, structured, unstructured), data variety, and several data sets. 2. **Data governance:** Availability, integrity, usability and security of data. 3. **Big Data:** Use of Big Data technologies and big-sized data sets. 4. **AI tools:** Use of AI technologies and algorithms for prediction, regression, recommendation, etc. 5. **System security:** Elements and techniques that guarantee the security of the data platform as a whole. 6. **Multi-purpose:** Scalability and flexibility of the system and if it can be used for different purposes with data of diverse nature and fields of knowledge. 7. **User friendly:** Usability of the system and an easy-to-use user interface for accessing the features of the system. 8. **3rd parties IaaS support:** Support of 3rd parties IaaS such as Google Cloud, Azure, AWS, etc. 9. **Tested with real users:** Features of the system used by different users with real data.

Considering the results of Table 1, AIMDP is one the most complete data platform amongst the selected ones. Data governance, heterogeneous data collection and security are characteristics that most data platforms fulfil, as they are central aspects of the definition of the concept. However, features such as AI or big data tools are not supported on all platforms. AIMDP not only includes these features but implements an advanced computing infrastructure that allows the running of Big Data-friendly algorithms. It is compatible with open-source tools such as Spark and allows newer algorithms to be added without much effort, leveraging the infrastructure defined in the following section and producing remarkable results in terms of quality and efficiency.

The main difference between AIMDP and other data platforms is that it is multi-purpose and easy to use. Thanks to the ease of use, the dynamism and the ability to implement generic algorithms, the platform can be classified as multi-application, that is, it can be used to build different applications in many areas such as energy, medicine, social networks and so on, without being associated with any particular field.

As can be seen in the comparison table, AIMDP presents a limitation in the use third-party IaaS, whilst CALIBER and PatientsLikeMe does not. This decision has been taken because the data platform is designed to work with data with several levels of privacy. To guarantee data security, a set of private and centralized servers are used. However, in future challenges, described in Section 6, the development of a hybrid system is contemplated. This system allows critical data to be managed on a private server or a set of decentralized servers (allowing the use of federated learning) and the rest of the data on third-party servers.

Additional limitations of the platform are related to: first, the import of external data since, although it is dynamic, the management of this information may be improved by the direct connection to the ontology-based enrichment system; second, the lack of an AutoML tool so that the user can query the system and it automatically generates the entire pipeline. These two limitations are planned to be addressed in a future (see Section 6.1).

In Section 5, the features from Table 1 are put to the test in a real use case.

Table 1

Comparison of the main features of the mentioned data platforms. Legend: ✓ - Feature supported, ✗ - Feature not supported, ~ - Feature not fully supported or with explicit limitations, ? - Unknown, information not available about the feature.

Name	Heterogeneous data collection	Data governance	Big data	AI tools	System security	Multi-purpose	User friendly	3rd party IaaS support	Tested with real users
AIMDP	✓	✓	✓	✓	✓	✓	✓	✗	✓
ENTSO-E [22]	✗	✓	✗	✗	?	✗	~	✗	✓
CALIBER [7]	✓	✓	~	✓	✓	✗	?	✗	✓
D2D Big Data [25]	✓	~	✓	✓	~	✗	✗	✓	✗
WikiHealth [9]	✓	✓	✓	✓	✓	✗	✓	✓	✗
CIDAP [4]	✓	~	✓	✓	✗	✗	✗	✗	✗
TELCO [26]	✓	~	✓	✓	✓	✗	✗	✗	✗
PatientsLikeMe [10,11]	✓	✓	✗	✗	✓	✗	✓	?	✓

3. AIMDP architecture design

As explained in Section 2.3, the concept of a data platform has become widespread in recent years. Many data processing companies have oriented their developments towards the implementation of such platforms to optimize the work of their users. One of the main limitations of most platforms is the difficulty in adapting any kind of problem to this type of platform. Due to the complexity of some databases, mainly in the health sector, it is too tedious to achieve the assembly of this type of data structure within the architecture of the platform.

In our study, we have eliminated this weak point, creating a data platform that achieves complete separation of each of its layers, grouping its functionalities into different modules. In this way, the platform is capable of adapting to any type of problem, regardless of its scope. AIMDP is capable to recognize the category of data, whether it is structured, semi-structured or unstructured and to perform a basic adjustment so data from different sources can be stored in the platform and become available for experimentation. This functionality is achieved through the implementation of the Acquisition layer and the Application layer, the functioning of which are explained in detail in Sections 3.2 and 3.7. This provides versatility when performing any type of work, whether it is oriented towards health, as is our use case in Section 5, or other types of work, such as those oriented towards energy efficiency, etc.

In our data platform, we have achieved that the different layers that it is made up of being independent. Thanks to this independence, in the event of an error in any of the layers, the platform is not completely affected, thus considerably reducing critical errors. The independence of each layer has made it possible for the platform to adapt to the different changes that occur throughout the life cycle of the platform in real-time without affecting the rest. Fig. 1 shows the different layers that make up our data platform. The following sub-sections explain the architecture and functionality of each layer.

3.1. Source layer

In the source layer of AIMDP, the data platform is enriched with data extracted through a series of extraction, transformation and loading (ETL) processes and stored so that it is available for the users of the platform. To perform this task, we are using Pandas [43]. Pandas is a highly advantageous open-source python library for data science. It is renowned for its speed, power, and flexibility, as well as its ease of use in data analysis. These strengths make it a valuable tool for data scientists and analysts, who can quickly and effectively manipulate and analyse complex datasets. Additionally, its open-source nature means it is constantly being improved and updated by a community of dedicated developers, making it a reliable and cutting-edge

option for data analysis tasks. Overall, Pandas is a highly effective and widely-used python library for data science.

In our platform, it is possible to include sources such as data repositories, as well as data lakes, which store a large amount of data that is retained until it is needed to be used. The data stored in these repositories is unstructured. Another storage system we use in our data platform is data warehousing, which is designed to store data in a structured way, as there are also less flexible databases which follow a relational structure. For this task, we use Oracle [44] tools, although any available data warehouse could be used. In this way, we manage to work with both structured and unstructured data.

In this layer, it is also possible to find simpler sources from which a large amount of information is obtained. These are the databases themselves, the import of files, physical media, as well as the use of web services for data collection. With the help of all the above elements, plus the collection of data from sources such as hospitals or sensors in office buildings, the complete data storage ecosystem of our architecture is established.

One of the main advantages that our platform offers over others is that the data is not obtained from a single data source. AIMDP provides the possibility of obtaining data from almost any repository (Github [45], Data Repositories, etc...), sensor (temperature, heating, cool, etc...) or storage device (USB, hard disk, etc...), even if the data is siloed and has strict privacy requirements. Through a series of data extraction, transformation and loading (ETL) operations, AIMDP is capable of adapting to the data set so that it can be stored to be subsequently used by any layer of the platform. The data platform configures the database by identifying the data structure of each source, managing to adapt nearly any data structure regardless of the complexity it offers.

In the following sub-section, we explain how to obtain the data of the different elements explained in this section and how to access the information already stored in the data platform.

3.2. Acquisition layer

The data acquisition layer focuses on the creation of the data storage architecture. This is necessary so that the platform can obtain the different data that the user needs to perform different experiments.

This layer is where the necessary ecosystem is created to supply the application layer with the different databases. It also implements a series of application programming interfaces (API) capable of retrieving and accessing stored information that can be interpreted and subsequently processed efficiently. The API has been created using the flask api-restful tool [46] because it offers several advantages. Its implementation in Python makes it easy to use and integrate with other Python-based technologies. Additionally, the flask API-restful tool is well-known for its simplicity and flexibility, allowing developers to quickly and easily build robust, scalable API solutions. Overall, the decision to use

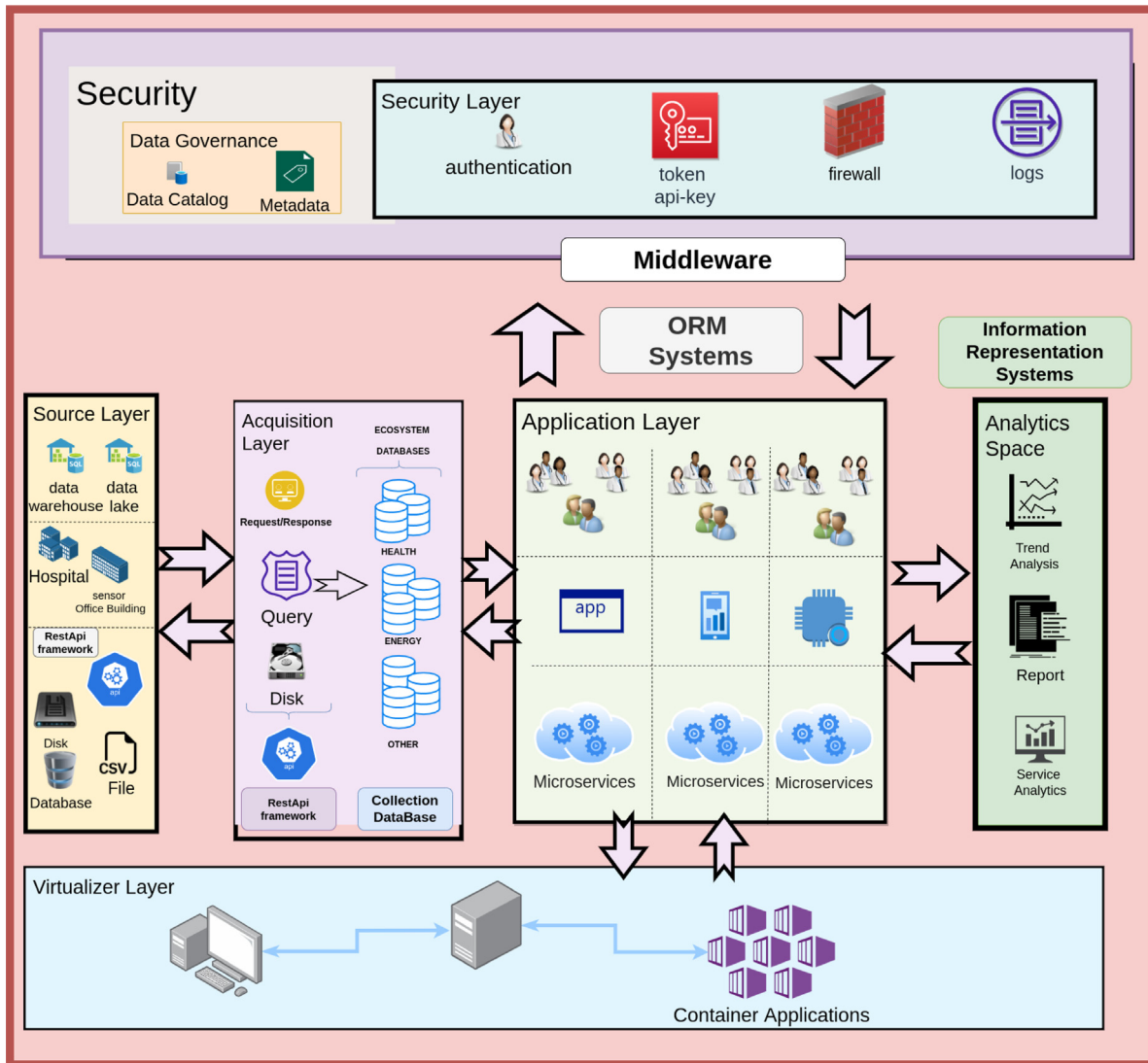


Fig. 1. AIMDP architecture design.

flask API-restful was driven by its strong support for Python and its powerful features for building APIs.

In this paper, we have created a storage architecture with the help of MongoDB [47], in order to establish an ecosystem where all the databases are placed and managed to work on the different stored data structures. Each database that makes up this architecture contains a series of collections.

These collections are usually of two types, one where the raw or processed data is stored in key-value documents and another collection where the metadata of the variables of the first collection is stored. Metadata collection is very important because it provides the necessary information for the user to use the data in a structured way in the platform. Through an intelligent process, AIMDP obtains the knowledge of the metadata collection and makes it available to the users so that they can carry out different experiments having all the necessary information about the data.

One of the advantages of this study that differentiates it from others is the possibility of having each of the structures independently within the same database storage ecosystem. This is achieved through the creation of a sub-layer where data is stored in multiple collections, maintaining data and metadata. Thus, when required by the user, the platform loads metadata into the memory for the user to work with. This produces better

maintainability and the possibility of producing independent experiments using the data hosted within our storage architecture, achieving more efficient and faster results. Regarding this, the data stored does not necessarily have to be of the same type since the databases are independent. This allows data and structures from different research areas to be part of AIMDP and processed with its tools. Additionally, a feature that is offered to the users is that the data from different areas of knowledge can coexist, being able to assign each group of platform users a subset of databases independently, depending on the permission of the user.

Thanks to the creation of this layer and the storage architecture, one of the objectives we set is fulfilled, which is that users can focus on their experimentation without spending a lot of time adapting, transforming and manually exploring the data stored on the platform.

3.3. Data virtualisation layer

The virtualisation layer is based on a cloud architecture that some data platforms implement. We have set up the platform with the help of Docker [48]. This tool automates the deployment of applications inside software containers. This provides an additional layer of abstraction and the automation of application

virtualisation across multiple operating systems. In this way, we achieve a modularized platform.

Docker-driven modularization guarantees that any of the modules, which are independent of each other, can be maintained at any time without affecting the rest of the containers. This is a great advantage because it allows the system to be more stable and respond effectively when critical failures occur, guaranteeing the integrity and security of the information in the modules that are not affected by the failure. Due to this modular approach, it is possible to introduce any application considered useful for the platform. Another aspect to highlight is the possibility of integrating this layer with widespread services such as AWS [49], Google [50] and Azure [51], as it is mentioned in Section 6.1. With this, it would be possible to access the powerful cloud computing resources of these third-party IaaS, thus increasing the potential of AIMDP in terms of the virtualization layer and computing and storage capabilities.

Due to the modularization of this layer and the integration of Docker, the maintainability of the different work ecosystems that make up the platform is achieved. AIMDP provides the possibility of extending its size just by adding any programming interface that contributes new knowledge to the platform.

3.4. Analytics space

The analytical space layer is responsible for the generation of different elements to improve the visualization and understanding of the data stored in the data platform. These elements can be simple, such as PDF/HTML reports, automatic charts, or tools that allow users to customize how they view and explore the data they are using. However, AIMDP also supports more complex display options such as graphs, nodes, or data cubes [16]. In this layer, there are a number of micro-services that communicate with the API of the platform, which analyses the data used by the user within AIMDP. This layer performs visualization tasks, reports the results of the different experiments, explores data, OLAP cubes [52], etc.

One of the key features of the analytical space layer is its ability to generate complex visualizations of data. This allows users to gain a deeper understanding of the data stored in the platform and to explore it in new and meaningful ways.

In addition to generating charts and graphs, the layer also supports the use of OLAP cubes, which are specialized data structures that allow users to quickly and easily perform complex data analysis tasks. By leveraging these powerful tools, users can quickly and easily gain insights into their data that would be difficult or impossible to uncover using other methods.

The purpose of this AIMDP layer is to offer more flexibility to the user to understand, visualize and analyse data. This layer solves the problem of exploring data from a wide variety of sources, a common drawback, as mentioned in Section 2.7. Thus, the elements of the layer are better adapted to the needs of the users' project, allowing a personalized visualization of the data available to them.

In conclusion, the analytical space layer of AIMDP provides users with flexible tools for understanding, visualizing, and analysing their data. This allows them to overcome the limitations of traditional data platforms and to gain a deeper understanding of the data they are working with, as mentioned in Section 2.7. By providing a personalized approach to data visualization and analysis, the analytical space layer is better suited to the needs of users' projects and can help them to extract valuable insights from their data.

3.5. Data governance

The main function of the data governance layer is to ensure the integrity of data, the elimination of leaks in data transmission and the availability of data for registered users. Alongside the security layer, it guarantees that only authorized users are able to access each of the databases.

AIMDP's data catalogue follows a hierarchy that goes from the most generic option, where the nature of the project is included, to the multiple databases offered by each of the projects of the data platform.

Policies have been established using the middleware authentication provided by the Django framework used in our system. Our middleware ensures and maintains the integrity of the data within the system's database by performing automated checks on a regular basis to verify the integrity of the data and eliminate any errors within it.

The middleware authentication provided by Django is an essential part of the security protocols in place for our system. It ensures that only authorized users are able to access the data stored in the database and that the data is kept safe and secure. In addition to providing authentication, the middleware also performs regular checks on the data to ensure its integrity. This helps to ensure that the data remains accurate and reliable, even as it is being accessed and used by different users within the system. Overall, the middleware authentication provided by Django is an important tool for maintaining the security and integrity of the data within our system.

In this way, users can only access the parts of the ecosystem they are authorized to, eliminating inconsistencies within the database itself. This is a key feature of the platform, as parts of the data within a single database are restricted to users without permission, ensuring controlled, personalized, and secure access.

3.6. Security layer

One of the objectives of the layer is to reduce the loss of information and critical integrity errors that may exist in the different databases, preventing them from affecting the rest of the database. This layer is crucial in the functioning of the platform since sensitive data can be included in AIMDP, as shown in Section 5.

In addition, the elements of the layer guarantee great flexibility to access the databases of the platform. We have divided the layer into three levels, network, application and user or research group:

- At the network level, AIMDP uses a virtual private network (VPN). This allows users and administrators to communicate securely and access platform resources without loss of information.
- Regarding the application level, security is based on Docker containers [48]. We have created an internal network where applications using AIMDP data can communicate when running within the same environment. This is another way to prevent sensitive information from leaving the control of the platform.
- At the user or research group level, an authentication system has been implemented. This system generates internal tokens exclusively for each session in which the user performs an experiment or uses any of the AIMDP tools. This provides the AIMDP administrators with a way to differentiate each user or research group and set custom permissions.

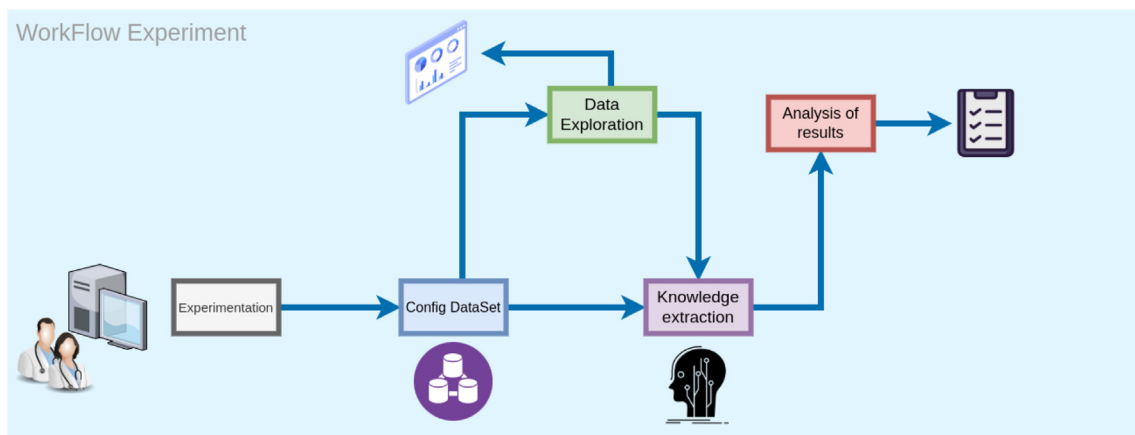


Fig. 2. AIMDP experimentation workflow.

3.7. Application space

This layer controls the communication of the different layers that make up AIMDP with the applications that use the platform services.

Regarding the acquisition data layer, we obtain the necessary data for its operation through a series of application interface programs. Through the API, different users can interact independently with the storage architecture of the platform, using a series of micro-services that have been created. Micro-services are small programs that execute specific tasks within the data platform. The small size of these tasks makes the micro-services independent of each other, being able to coexist on the platform without depending on each other.

This layer provides the possibility to connect with the analytics space layer, where the information provided to the different applications can be represented in a more customizable way, as explained in Section 3.4. Security and data governance layers, described in detail in Sections 3.5 and 3.6, allow this layer to interact with this information. They offer the possibility of allowing access to certain knowledge of the platform by assigning a series of credentials to each of the platform users, either individually or through work groups.

The virtualisation layer also plays an important role, as explained in Section 3.3. This is where the different applications that make up this layer are deployed on the server. Each of the applications that compose AIMDP is independently assigned a space within the platform server so that the crash or removal of one application does not affect the rest.

Finally, due to the interaction between the different layers, an ecosystem is achieved in such a way that data stored in AIMDP can be used through user-friendly interfaces, following a workflow that allows users to focus exclusively on extracting information from their data. The user interface has been created using Django, a popular framework for developing web applications in python. While other technologies could also be used for this purpose, Django was chosen for its powerful features and strong support for python. However, it is important to note that the developer is not limited exclusively to Django and could potentially use other technologies as well.

This user-friendly paradigm that allows the user to obtain results quickly and efficiently is one of the main aspects of the following section, which offers a deeper study of how the workflow has been implemented.

4. AIMDP user-friendly workflow

One of the problems that most non-programming-expert researchers have when performing experiments using a data platform is the lack of a guided workflow that takes them to step by step to achieve a goal in the execution of their work. They have a great deal of expert knowledge about their data, but most of them are not experts in pre-processing and extracting all the knowledge from the data through the application of data science and computational techniques. One of the objectives of this study is to solve this problem. To do this, we follow the paradigm of ease of use, as explained in Section 2.6.

In this section, we explain how the AIMDP workflow is structured to help the user focus exclusively on extracting insights from their data. This section focuses on the application layer, as it allows the user to perform different experiments in a simple way within our AIMDP architecture. As Fig. 2 shows, the workflow is divided into different stages or modules. In the next sub-section, each stage of the workflow is explained, exposing each of its characteristics.

4.1. Experimentation module

The experimentation module is the first stage of the workflow. In addition to the application layer, security, governance and acquisition layers are involved in the operation of this module.

The acquisition layer allows the display of the different experiments in this module which are available to the user. With the help of the security layer, it restricts access to each experiment and, as explained in Sections 3.5 and 3.6, prevents data loss. The governance layer is also involved in this module as it achieves data integrity, avoids redundant or erroneous data within the databases, and controls data availability.

The creation of the experiments is dynamically stored in the acquisition layer, associating it exclusively to the owner user through the security layer. Our application layer stores only the metadata, such as the experiment's name, the database selected in our experiment and the parameters chosen to filter the different variables of the experiment itself.

The great advantage of this storage system is that the only thing stored in the database supported by the AIMDP system is plain text, which considerably reduces our application's necessary percentage of storage. This way of storing experimentation does not involve the storage of records from the hospital's data collections, which eliminates data redundancy through the governance

layer. All the management within our application layer is done with the python framework Django.

In this module, users can create a new experiment by selecting a specific database or uploading an experiment already performed by the active user. This gives the user the ability to reuse the work of other experiments, which can be useful for re-running experiments with some changes or not having to enter all filters and workflow parameters.

4.2. Data-set configuration module

In this module, the user can configure which variables from the database selected in the previous module are going to be used in the experiment, using a variable selection tool. It is also possible to filter the variables, taking into account whether they are numeric or categorical. In the case of numeric variables a numerical range can be selected, and in the case of categorical variables, some particular values within the domain of the variable can be selected.

One of the main problems of data processing systems is the great variety of databases, each of which has different variables, data types and even representation structures. AIMDP solves this problem by dynamically and intelligently assembling the variables that make up the database on which the experimentation is based, thanks to the storage architecture described in Section 3.2. Therefore, regardless of the user's specialization, AIMDP can automatically upload the data to the data platform.

With the AIMDP filtering tool, the user can select both the data set and the variables needed for experimentation. The filters and parameters selected in this module are stored in a configuration file, which prevents the user from having to configure these filters multiple times. In this way, the platform gains in efficiency and we prevent the server from suffering load saturation, reducing the percentage of critical errors on the platform.

At this point, users can perform exploratory data analysis on the data they have selected or direct their experimentation towards knowledge extraction.

4.3. Knowledge extraction module

One of the main barriers of other data platforms is that extracting knowledge from the data is often quite tedious and inflexible due to the numerous constraints presented by these algorithms, as well as the need for extensive knowledge in the field of data science and programming. The elimination of this drawback is one of the goals set in the development of AIMDP, which guides the user through extracting knowledge from the selected dataset for the current experiment.

The user can select any of the algorithms offered by AIMDP implemented[2] in our APIRest. For the execution of this kind of algorithm, advanced knowledge in data science is required. Therefore, a user help system has been implemented to show the user the necessary help to run this algorithm and to see if this type of algorithm fits the data used in our experimentation.

Currently, our AIMDP allows us to execute cluster algorithms in a sequential or parallelized way due to the first adaptation offered by our tool in distributed environments implemented in the Spark framework. Once selected, each of these algorithms can configure automatically executed by a Big-Data-proof computational cluster, following the conditions required by our help system for the execution of the algorithm in question.

Another advantage of our AIMDP architecture is its high encapsulation. This feature allows us to increase the range of algorithms needed to adapt to different experiments without interfering with developing other application layers that interact with our architecture.

The main goal of this module is that the user can focus on the configuration of his dataset and does not have to spend a large part of his time delving into the field of data science. Once the algorithm has been run, the user can check and analyse the results using the results analysis module, detailed in the results 4.5.

4.4. Exploration module

One of the reasons why this type of platform is indispensable is the possibility of representing data in a way that makes it easy for the user to visualize and obtain information from it. In AIMDP, a module includes this type of interpretation, which is the data exploration module. In this module, users can comprehensively analyse and visualize the dataset selected for experimentation. Users can select any data or variable from the dataset and analyse it in a way that allows them to conclude their studies.

This module allows the user to visualize their variables' behaviour in the advanced stages of running the experimentation. It allows the user to extend their knowledge of the selected variables, discovering hidden or undetected behaviours and allowing them to decide to improve the execution of the different algorithms.

The analytics space layer supports this module. This layer achieves the objective of encapsulating the functionality within AIMDP in order to be able to extend its behaviour throughout the architecture life cycle without affecting different applications that are interacting with AIMDP. Some of the more straightforward visualization tools are offered by the analytics space layer in AIMDP.

Currently it allows the user to generate pie charts and histograms, depending on the ranges and types of variables selected for exploration. It also includes more complex visualization tools to see the possible relationships between two or more variables, using box plots, scatter plots and others, depending on the type of variables to be compared (see Table 2).

As mentioned above, AIMDP also allows the user to visualize more complex data using graphing tools, nodes or data cubes, giving the possibility to develop external visualization libraries and interact with our architecture [16].

4.5. Result analysis module

The results analysis module is usually fundamental in all data platforms. It is where users can draw their conclusions from the experiments performed and decide whether the execution of an experiment achieves the expected results or whether a new, different experiment should be performed.

This module analyses the results obtained after running an algorithm from the knowledge extraction module. It generates a series of automatic tables and graphs depending on the algorithm selected by the users and the data used in the experiment. Depending on the algorithm executed, the system will generate a series of visualization elements to be displayed to the user. These dynamic and intelligent charts and graphs help the user interpret the experiment's results, which is crucial in the development of their studies.

The application layer manages this stage of the workflow. In this way, independence is achieved when displaying the results to the user, as the application layer decides how to display these results. This decision is because, depending on the problem, the different application layers using our architecture will be free to display the results in one way or another.

Our AIMDP will only manage the processing of the data and will send results that the application layer will be in charge of representing. As the application layer manages these results,

Table 2
Algorithms included in AIMDP.

Name	Cite	Kind of algorithm	Aim	source	Big data	Efficiency
BDARE	[53]	Association Rules	No supervised	Own development	✓(Spark)	Exponential
FIM in Streaming	[12]	Frequent items set mining	No supervised	Own development	✓	Exponential
FARE	[14]	Fuzzy association Rules	No supervised	Own development	✓(Spark)	Exponential
OBTD	[15]	Topic detection	No supervised	Own development	✗	Cubic
CDMA	[16]	Multidimensional analysis	No supervised	Own development	✗	Cubic
FSD	[17]	Fuzzy multidimensional analysis	No supervised	Own development	✗	Quadratic
MLIB LR	[54]	Logistic regression	supervised	MLIB	✓	Linear
MLIB DT	[54]	Decision tree	supervised	MLIB	✓	Linear
MLIB RF	[54]	Random Fore	supervised	MLIB	✓	Logarithmic
MLIB SVM	[54]	Support Vector machine	supervised	MLIB	✓	Cubic
MLIB GBT	[54]	Gradient-Boosted Trees	supervised	MLIB	✓	Logarithmic
XGBOOST	[55]	XGBOOST	supervised	Xgboost	✓	O(tdxlogn)
scikit-learn PCA	[56]	PCA	No supervised	scikit-learn	✗	Cubic
scikit-learn Kmeans	[56]	K-means	No supervised	scikit-learn	✓	Quadratic
Distributed K-Prototypes	[57]	Clustering	No supervised	Implementation based on [57]	✗	Linear

users can store each of the results provided by this module in its reserved storage area. In our case, it can be consulted at any time for future studies. The possibility of reusing previous reports generated after the execution of an experiment is another advantage of AIMDP.

In the next section of this paper, we include an experiment that goes through all the workflow stages described in this section. To test this workflow and the architecture described in Section 3, we have included a real use case with health-oriented data silos.

5. Use case: IA experiment applied to a spanish health service data silo using AIMDP

In fields such as medicine, a great deal of data tends to remain siloed and not used in data analysis tasks due to privacy and standardization problems [10]. Data silos [10,58] are repositories that cannot be accessed from the outside because of privacy, regulations and other issues. These data sets with unexploited potential may cause losses in financial or medical organizations [59], due to delays in scientific progress in different areas. The problem of data silos motivated the use case introduced in this paper. Since the main collaborators and users of the data platform are doctors and researchers from Spanish hospitals, the data silos in their storage system can be analysed and processed. The AIMDP platform can also provide an improvement in the research work of the collaborators of the hospital since they do not have to expend hours improving their programming skills and they can focus on their real study area. This is achieved with the tools described in Section 3.7, which allow users to perform data mining and analysis tasks without the need for programming skills since usability and ease of use are the central aspects of the AIMDP data platform.

In order to validate and evaluate the AIMDP data platform, the architecture and all the modules presented in Section 3, a real use case has been carried out. It consists of setting up all the features of AIMDP so it can be used by real medical collaborators of two hospitals in the Spanish health service: the Costa del Sol and San Cecilio hospitals. These researchers are the first real users of the system. In order to test all the features the following tasks have been carried out: 1. Two health data sets that have never been used for data analysis (data silos) with medical information of patients in these two hospitals have been uploaded to the data platform. 2. The system has been deployed on a server of the University of Granada. 3. An account for each of the users has been created. 4. A pre-processing and enrichment of data has been carried out. 5. An experimentation workflow has been defined, based on the aspects described in Section 4, so the users can perform a whole experiment using all the modules of the data platform. 6. An experiment using the data set has been done and the results have been analysed.

5.1. Data sources

The data sets uploaded into the data platform used for the experimentation follow the structure of the Minimum Basic Data Set (MBDS) of Andalusia [60]. The structure of these data sets is based on EHR and they contain the health records of real patients of the two hospitals. These records come from different sections of the hospitals (emergency, mental health, maternity, etc.) and each record is associated with a patient's hospitalization episode. The data stored in these registers have different formats because they come from a wide diversity of data sources. Some of the most representative sources and features can be seen in Table 3.

Characteristics about the dimension of the raw data sets can be found in Table 4. Since AIMDP implements features of a data lake, raw data is uploaded and kept in the storage architecture described in the acquisition layer (Section 3.2). This is useful since the data platform implements Big Data tools and it is capable of working with large amounts of data and takes advantage of information that could be lost during the data processing stage. Nevertheless, during the data loading process, some custom operations are carried out automatically on the data sets. This generates new data sets that are also stored in the platform. These operations are described in the following sub-section.

5.2. Data processing

As has been mentioned previously, the data sets considered for the use case described in this paper have the structure of the Minimum Basic Data Set (MBDS) of Andalusia. With this in mind, a series of operations are carried out on data at the same time the data is being uploaded into the AIMDP storage architecture, details of which can be checked in 3.2. The operations are transformation, pre-processing, enrichment and loading. All these operations are hidden from the user, maintaining the usability of the system.

5.2.1. Data transformation and pre-processing

During the transformation stage, factor variables are mapped from integers to understandable labels, using the manuals and supporting documents of MBDS [60]. During this phase, the data is transformed into key-value documents. This is an efficient way to store the considered data sets because there are very dispersed variables.

The next stage is the pre-processing of data. Here, new variables are computed using existing variables, since the information represented in these variables is usually not data analysis-friendly. Variables used for these operations are history codes, postcodes, dates, etc. Some examples of this kind of pre-processing can be found in Table 5.

Table 3
Data source features description.

Features	Codification	Method for enrichment
Diagnoses	International Classification of Diseases (ICD)	External API
Origin of patients	Spanish health service system	Database
Reason for discharge	International Classification of Diseases (ICD)	External API
Reason for admission	International Classification of Diseases (ICD)	External API
Surgical procedures	International Classification of Procedures	External API
Diagnostic tests	International Classification of Test	External API
Services/departments	Spanish health service system	Database
Other data ^a	Andalusian health service system (SAS)	Database

^aRest of the variables related to the management and information of the Andalusian hospital system.

Table 4
Characteristics of the data sets of the use case.

Data set	Records	Features
EHR - Costa del Sol Hospital	75.000	273
EHR - San Cecilio Hospital	220.000	273

5.2.2. Data enrichment and loading

Data enrichment techniques applied to the data sets use the classifications and codes included in the International Statistical Classification of Diseases and Related Health Problems 10th Revision (ICD-10) [61]. This process is carried out on variables such as diagnoses, surgical procedures or external causes of morbidity and mortality. A mapping with 3 different levels of granularity is made with these variables, as is shown in Table 6. Dig level 1 provides a more wide and more generic diagnosis, while level 3 maps the diagnosis to a more specific and precise one.

After all the mentioned processing has been applied to each data set, they are uploaded into the storage system of the data platform. This process is described in Sections 3.1 and 3.2. For each data set, a copy of the raw data transformed into key-value documents and a processed one are uploaded.

5.3. Data experimentation

In this section, a complete clustering experiment is presented using the AIMDP data platform. This experiment has been shown to the users of the data platform as an example or guide of how an experiment should be performed, so it is meant to be understandable and show the main characteristics of an experimentation workflow. The proposed clustering problem consists of extracting information about the underlying structure of childbirths in the Costa del Sol Hospital between the years 2016 and 2020. A secondary objective is the preparation of the data for future data analysis using complex AI algorithms such as [62–66].

Before the configuration of the data set and algorithms of the experiment, the user must access the URL of the system using a web browser. This web application is deployed on a University of Granada server. The user then has to log in using the authentication module, described in Section 3.6. This is also a key piece in the data governance management of the platform, defined in Section 3.5. This is because once users have been logged in, it is possible to control which data is available for each user, what

they can do with the data and so on. The user is now capable of creating and running an experiment following the steps included in Fig. 2. All the modules that appear in the figure are described in detail in Section 4.

The first step is the creation of an experiment and the selection of the parent data set on which the experiment is based. This can be done using the experimentation module, which is defined in Section 4.1. The parent data set contains the processed and enriched data of the hospital, obtained after applying the methods described in the previous Section 5.2 to the data set with raw data. This processed data set is selected because these types of algorithms give a worse performance using raw data. The second data set of Table 1 is selected as the parent data set, assuming the user has access to it. Next, the user can configure a sub-dataset using the variable selection and filtering tool of the system, which does not require any programming skills from the user. Available metadata information is also useful since it ensures that the user possesses all the needed information about each variable. These operations are carried out in a very efficient way since, as it is mentioned in Section 3.7, the system can perform the filtering and selection operations by only using the metadata of variables.

These filtering and selection operations are carried out by the data set module, described in Section 4.2. The selected variables and filters for the clustering problem-solving can be found in Table 7. It is worth mentioning that only childbirth episodes are considered since PESO1N, the variable that contains information about the weight of the babies is selected and missing values are removed from the experiment. After the sub-data set has been built, all the filters and selected variables are stored in a configuration file that can be loaded for future experimentation, as it is described in Sections 4.1 and 4.2.

At this point, the user can perform an EDA using the tools of the module detailed in Section 4.4 or proceed with the experiment and use the tools available in the knowledge extraction module of Section 4.3, where the clustering algorithms can be configured. The EDA module has been used to explore the data and purpose of the clustering problem, so the following step is to select the algorithm and set up its parameters. For solving a clustering problem, there are some available algorithms in Spark MLlib. From this library, algorithms such as K-means or Bisecting K-means are supported currently in AIMDP, but these do not take advantage of the categorical variables' information. To solve this problem, a Spark-based version of the classic K-Prototypes has been developed, based on the proposal [57].

Table 5
Example of the pre-processing of temporary data from the hospital database.

Date admission	Date discharge	Birthdate	
20/4/2016 10:16:40	23/4/2016 18:23:12	11/2/1991	
10/10/2019 22:45:20	13/10/2019 19:20:45	7/2/1987	
↓			
Days of admission to labour	Season	Age	Patient type
2	spring	31	adult
3	autumn	35	adult

Table 6
Example of the processing of an instance of C00.4 diagnosis.

Diagnosis				
K35.20				
↓				
Dig Level1	Dig Level2	Dig Level3	Application to	Group diagnosis
Diseases of the digestive system	Diseases of appendix	Acute appendicitis	(Acute) appendicitis with generalized peritonitis NOS	Major gastrointestinal disorders and peritoneal infections with and without cc/mcc

Table 7
Selected variables for the clustering experiment.

MBDS code	Description	Type	Filtered domain
PAISNAC	Mother's country of Birth	Categorical	Argentina, Spain, Morocco, Paraguay, United Kingdom
EDAD	Mother's age in years	Numerical	All available data
TIEMPOING	Hospital admission time during childbirth in days	Numerical	All available data
PESO1N	Weight of the baby in kg	Numerical	All available data
TGESTAC	Gestation time in weeks	Numerical	All available data

Once the K-Prototypes algorithm has been selected, an automatically generated HTML form is shown to the user. This form has the parameters needed for the execution of the algorithm, and each form attribute is populated with default values. The HTML form and the values chosen for the parameters of the K-Prototypes algorithm's execution can be found in Fig. 5. After all the parameters have been set, the user can run the algorithm by pressing a button that sends all the information to the system. The connection to the computer cluster and the generation of the result plots and tables is also automatic, so the user completes the whole experimentation workflow without seeing a single line of code.

5.4. Results and discussion

Since the solved problem is a clustering one, the results are very graphic and easy to understand. This is the aim of the experiment: to be understandable for non-experts and to bring closer the AIMDP data platform and its capabilities.

The data platform offers a series of tables and plots generated automatically, depending on the parameters and the results provided by the algorithm, in this case, the K-Prototypes clustering algorithm. The most relevant table, which includes the main interpretation of results, contains the labels of each cluster found by the algorithm, the number of individuals placed on each cluster of data and a summary of the value of each variable in the cluster. The mean and the mode are computed for numerical and categorical variables, respectively. This interpretation can be found in Table 8. From this table, it is possible to deduce some information and propose some hypotheses:

- The algorithm found 4 clusters of different sizes. Clusters 3 and 4 are the most populated groups.
- The most common countries of birth in the first two clusters are UK and Morocco, respectively, whilst the most common in the last two clusters is Spain. The algorithm made a clear separation in the groups based on this variable.
- There is no significant difference regarding the mean weight of the baby, gestation time and hospital admission time variables (PESO1N, TGESTAC, TIEMPOING).
- Looking at the mother's age variable, the mean in the first two clusters is similar, whilst the mean in the third and fourth one is clearly different, with a difference of approximately 10 years.

- The weight of the babies is higher in cluster 2.
- Bearing all of this knowledge extracted by the algorithm in mind, the following hypothesis is proposed if 4 clusters are considered: 1. The first cluster contains mothers from the UK, which is the smallest group found. The weight of their babies is similar to the Spanish mothers' babies from clusters 3 and 4. The age is similar to the one of cluster 2 and the mean between clusters 3 and 4. 2. The second cluster contains mothers from Morocco whose babies' weight is higher and triplicates the size of the first clusters in terms of individuals. 3. The third and fourth clusters aggregates Spanish mothers of different ages, with cluster number 3 being associated with younger mothers. Looking at the sizes of the last two clusters, it is possible to observe that there are more mothers that belong to the cluster with a higher value of age.

These results can also be observed graphically in the plots generated automatically by the system. In Fig. 4, which contains a plot of the distribution of the considered numerical variables, it is possible to see the importance that the algorithm gives to the mother's age variable (EDAD), which is very important in the delimitation of clusters 3 and 4. On the other hand, in Fig. 3, the relationship between only the categorical and the rest of the numerical variables is summarized in a box plot. This provides information about the distribution of the rest of the selected countries of origin, Argentina and Paraguay, which are distributed amongst the four clusters, with the minority being in selected data. Looking at the plot, the importance of age is also visible, as it has different distributions amongst the clusters. As the last aspect to remark, the user is also able to see that the weight of the baby and the gestation time are variables with a considerable number of outliers, which must be considered in future analysis operations using this data set.

This experimentation process has been put on into a video, which can be accessed by all the registered users of the platform, serving as an example and a user guide. The problem that has been approached and the results of the algorithm were presented to the health-specialist collaborators of the research team. Since all the hospital personnel were capable of understanding and checking the potential of AIMDP data platform, we see this as a new way of validation of the experimentation module and the system as a whole.

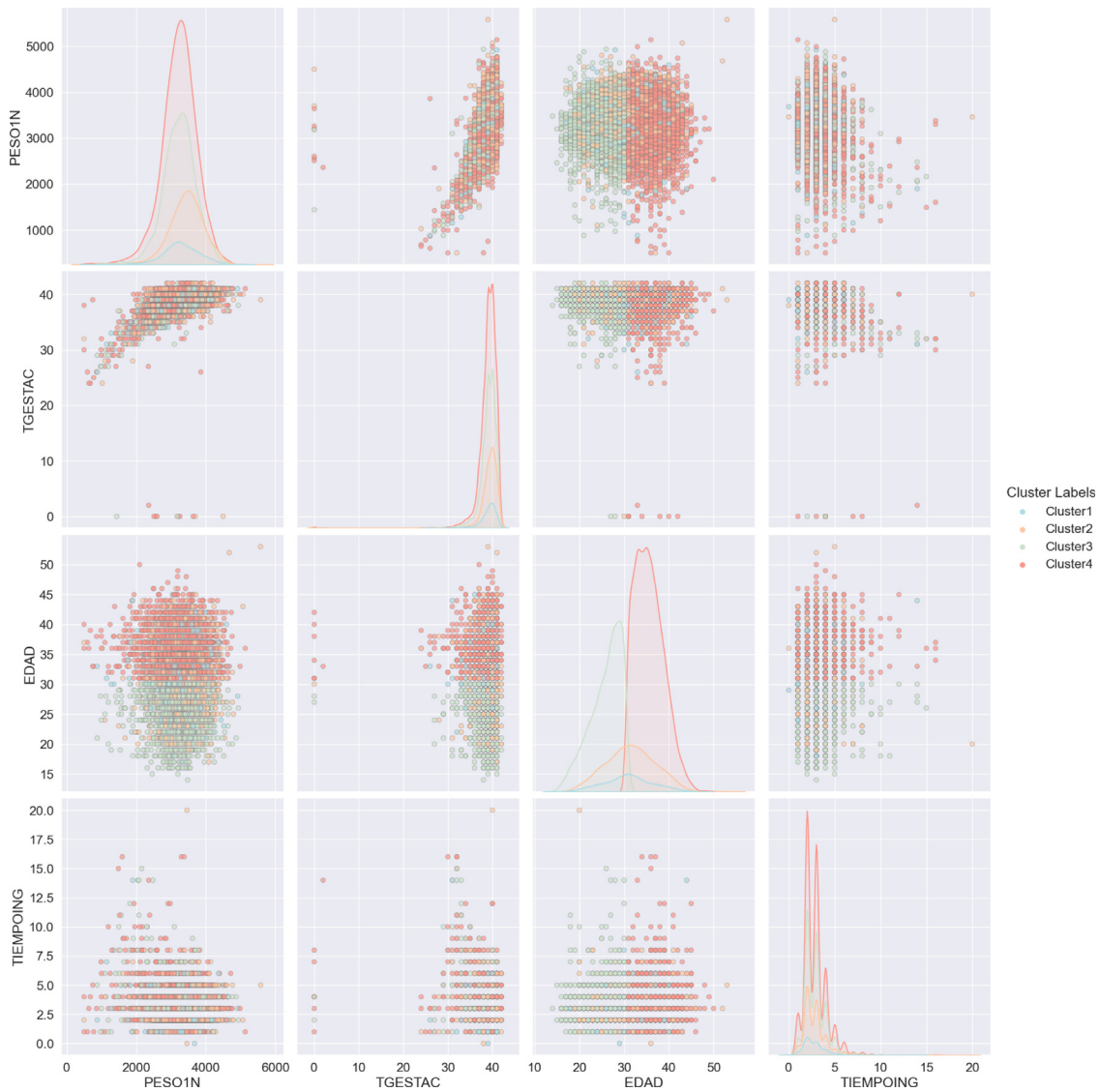


Fig. 3. Pairplot of the numerical variables. Two variables scatterplot and density plot in the main diagonal.

Table 8
Results interpretation.

Cluster labels	Total individuals	Weight of the baby	Gestation time	Mother's age	Hospital admission time	Mother's country of birth
Cluster 1	391	3274.50	39.00	30.90	2.73	UK
Cluster 2	1234	3444.69	39.29	31.45	2.77	Morocco
Cluster 3	2463	3234.20	39.03	25.80	2.87	Spain
Cluster 4	4016	3228.36	38.96	35.57	2.86	Spain

In addition to this example, we can see other results of algorithms implemented in the system in the state of the art. Among them, we can highlight works with fuzzy association rules in Big data for the extraction of hidden knowledge related to comorbidity in diagnoses [39].

6. Conclusions and future research

6.1. Future challenges

Through the platform proposed in this paper and the use case developed, it has been possible to see how the use of the Big Data platform can help in different processes involving various aspects of data management. Specifically, in our use case, we have seen

how it improves knowledge extraction from data silos. Among the main advantages of our system are its application for managing massive amounts of data, that it is a user-friendly system and its ability to import heterogeneous data from different systems. Furthermore, to complete this capacity, integration with third-party IaaS will be implemented in order to be able to work in a hybrid way and improve the capacity of the platform.

However, as demonstrated in the use case, a large amount of data is not used due to security and anonymity constraints. In this sense, several approaches can be used to tackle this problem, such as federated learning, allowing the extraction of knowledge from data in a decentralized way. We can also include an additional layer that takes care of this process, thus preserving the modularity of the platform and facilitating its development and

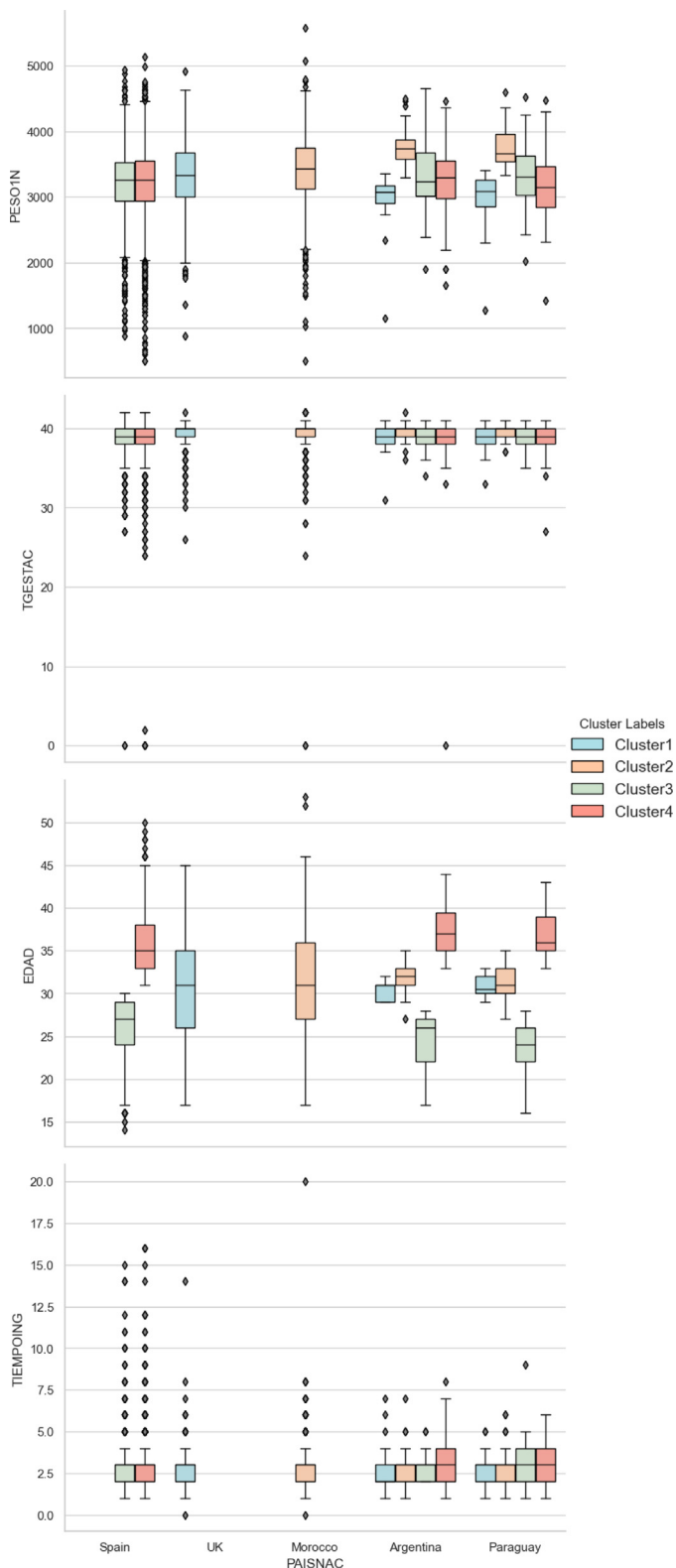


Fig. 4. Boxplot of the numerical and categorical variables. Numerical variables as rows and categorical as columns.

maintenance. However, we must remark that the data have been anonymized beforehand by the Andalusian Health System (SAS by its acronym in Spanish) for the use case presented. These approaches help maintain data anonymity and exploit hidden knowledge.

Regarding the integration of heterogeneous data, new methods have been implemented for the enrichment, and ‘structuring’ of unstructured data [29]. In addition, these methods pursue the objective of detrimental response time, which makes them more suitable for Big Data environments such as the one proposed in this article. Working in this direction will provide the AIMDP with greater flexibility, and further progress can be made in one of the platform’s strong points: the treatment of heterogeneous data sources.

Other challenges that arise are the integration of data through data flows that can be found in a multitude of applications such as health, energy, and social networks. For this, it is necessary to adapt the integration of this type of data, and its processing and to have algorithms that allow the analysis of trends [62].

Finally, we have seen how a friendly and intuitive system for the end user improves the system’s understanding, use and productivity. However, there are still challenges in the field of the use of data science tools by non-expert users in this field. Therefore, as enhancements to the system, an explainable artificial intelligence (XAI) layer has to be implemented to improve the interpretability and explainability of many of the methods that are available in our tool.

Using this future module, end-users will better understand the results obtained. Moreover, it will also be possible to improve the user experience by improving the generation and parameters of the algorithms by adding an intelligent AutoML process to select the best parameters and algorithms for users to obtain the desired results.

6.2. Conclusions

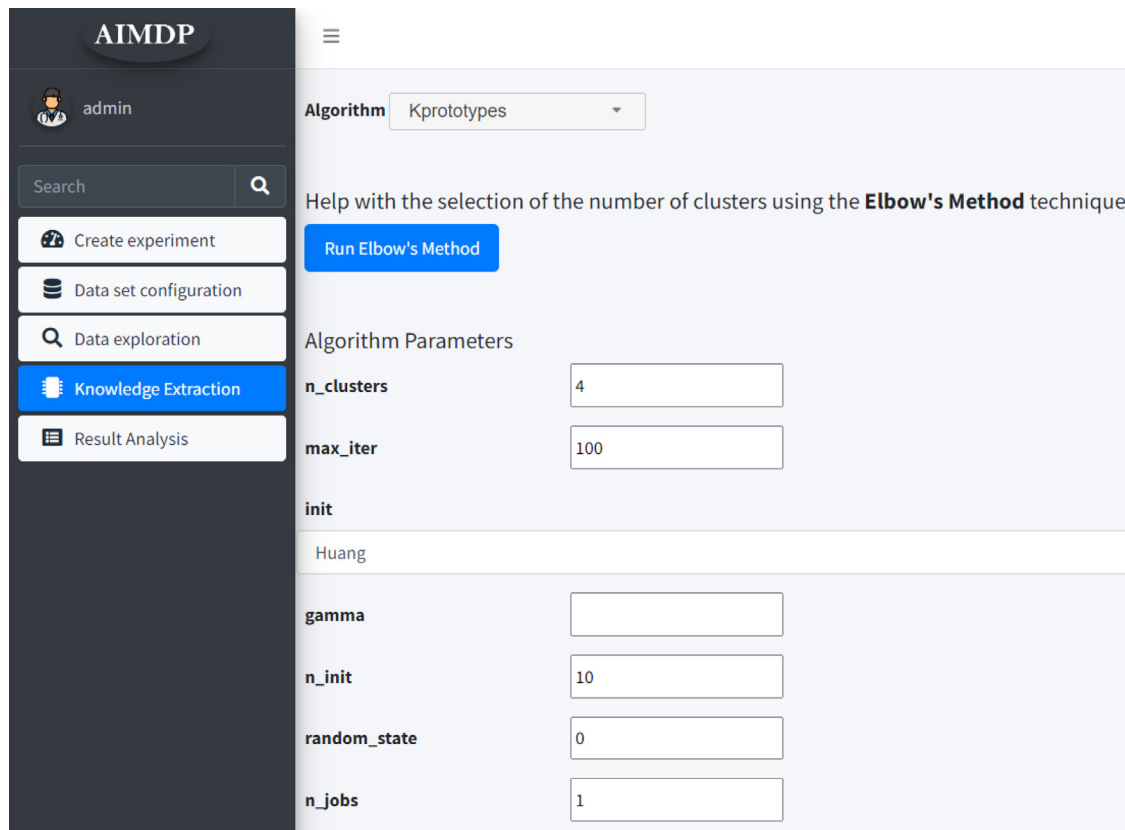
The proposed Modern Data Platform, AIMDP, has been effectively introduced in a real use case of a healthcare data silo, explaining its main capabilities. It has been demonstrated that it is possible to have a system capable of allowing a non-expert user to extract hidden knowledge using innovative technologies such as Big Data. Furthermore, this system has already included routines for data import [63], processing, data enrichment [39] and AI techniques to extract knowledge from large datasets [64–67]. The system offers a suitable framework for efficient and fault-tolerant data management due to the robustness of the systems used, such as micro-services, Spark etc.

Data is the heart of a system, although it is only stored in many cases, and the full knowledge it holds is not extracted. This is why platforms such as the one presented in this paper are necessary for the improvement of users in the different fields of energy, health, and economy to be able to analyse their data without a complex process in addition to being able to use innovative technologies and large computing systems in a transparent way.

This research opens the door to new implementations, improvements and applications of AIMDP to different fields, taking advantage of big data platforms and using the available Data Mining and Machine Learning algorithms that the scientific community has been developing in recent years. The main objective is to bring end users closer to the use of these new tools that can effectively help to manage and also to process the large volumes of data generated by social networks, medical records, images, sensors and other information external to the system, taking advantage of distributed computing and artificial intelligence in a simple, transparent and guided process.

CRedit authorship contribution statement

Alberto S. Ortega-Calvo: Supervision, Original draft preparation, Investigation, Software, Writing, Visualization. **Roberto Morcillo-jimenez:** Original draft preparation, Investigation,



The screenshot shows the AIMDP web-based application interface. On the left is a dark sidebar with the AIMDP logo and a user profile for 'admin'. Below the profile is a search bar and a list of navigation options: 'Create experiment', 'Data set configuration', 'Data exploration', 'Knowledge Extraction' (highlighted in blue), and 'Result Analysis'. The main content area has a light blue background. At the top, there is a dropdown menu for 'Algorithm' set to 'Kprototypes'. Below it is a text box with the instruction 'Help with the selection of the number of clusters using the **Elbow's Method** technique' and a blue button labeled 'Run Elbow's Method'. Underneath is a section titled 'Algorithm Parameters' containing several input fields: 'n_clusters' (value: 4), 'max_iter' (value: 100), 'init' (value: Huang), 'gamma' (empty), 'n_init' (value: 10), 'random_state' (value: 0), and 'n_jobs' (value: 1).

Fig. 5. Set up of the parameters of the K-Prototypes algorithm. Extracted from AIMDP data platform web-based application.

Software, Writing, Visualization. **Carlos Fernandez-Basso:** Conceptualization, Supervision, Writing – review & editing. **Karel Gutiérrez-Batista:** Conceptualization, Writing – review & editing. **Maria-Amparo Vila:** Supervision, Resources, Investigation, Writing – review & editing. **Maria J. Martín-Bautista:** Supervision, Resources, Investigation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

The research reported in this paper was partially supported by the BIGDATAMED project, which has received funding from the Andalusian Government, Spain (Junta de Andalucía) under grant agreement No P18-RT-1765. In addition, this research has been partially supported by the Ministry of Universities through the EU-funded Margarita Salas programme NextGenerationEU.

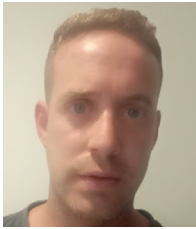
References

- [1] M. Tanifuji, A. Matsuda, H. Yoshikawa, Materials data platform - a FAIR system for data-driven materials science, in: 2019 8th International Congress on Advanced Applied Informatics, IIAI-AAI, 2019, pp. 1021–1022, <http://dx.doi.org/10.1109/IIAI-AAI.2019.00206>.
- [2] I. Vieira, A. Alvaro, A centralized platform of open government data as support to applications in the smart cities context, *ACM SIGSOFT Softw. Eng. Notes* 42 (4) (2018) 1–13.
- [3] Y. Liu, J. Peng, Z. Yu, Big data platform architecture under the background of financial technology: In the insurance industry as an example, in: *Proceedings of the 2018 International Conference on Big Data Engineering and Technology*, 2018, pp. 31–35.
- [4] B. Cheng, S. Longo, F. Cirillo, M. Bauer, E. Kovacs, Building a big data platform for smart cities: Experience and lessons from santander, in: 2015 IEEE International Congress on Big Data, IEEE, 2015, pp. 592–599.
- [5] M.D. Ruiz, J. Gomez-Romero, C. Fernandez-Basso, M.J. Martín-Bautista, Big data architecture for building energy management systems, *IEEE Trans. Ind. Inform.* (2021).
- [6] X. Fei, K. Li, W. Yang, K. Li, Analysis of energy efficiency of a parallel AES algorithm for CPU-GPU heterogeneous platforms, *Parallel Comput.* 94 (2020) 102621.
- [7] S. Denaxas, A. Gonzalez-Izquierdo, K. Direk, N.K. Fitzpatrick, G. Fatemifar, A. Banerjee, R.J. Dobson, L.J. Howe, V. Kuan, R.T. Lumbers, et al., UK phenomics platform for developing and validating electronic health record phenotypes: CALIBER, *J. Am. Med. Inform. Assoc.* 26 (12) (2019) 1545–1559.
- [8] University College London, CALIBER data platform official website, 2022, <https://www.ucl.ac.uk/health-informatics/research/caliber/accessing-caliber-resources>. (Accessed 14 June 2022).
- [9] Y. Li, C. Wu, L. Guo, C.-H. Lee, Y. Guo, Wiki-health: A big data platform for health sensor data management, in: *Cloud Computing Applications for Quality Health Care Delivery*, IGI Global, 2014, pp. 59–77.
- [10] T. Kariotis, M.P. Ball, B.G. Tzovaras, S. Dennis, T. Sahama, C. Johnston, H. Almond, A. Borda, Emerging health data platforms: From individual control to collective data governance, *Data & Policy* 2 (2020).
- [11] PatientsLikeMe, PatientsLikeMe official website, 2022, <https://www.patientslikeme.com/>. (Accessed 17 June 2022).
- [12] C. Fernandez-Basso, A.J. Francisco-Agra, M.J. Martín-Bautista, M.D. Ruiz, Finding tendencies in streaming data using Big Data frequent itemset mining, *Knowl.-Based Syst.* 163 (2019) 666–674, <http://dx.doi.org/10.1016/j.knosys.2018.09.026>.
- [13] C. Fernandez-Basso, M.D. Ruiz, M.J. Martín-Bautista, A fuzzy mining approach for energy efficiency in a Big Data framework, *IEEE Trans. Fuzzy Syst.* 28 (11) (2020) 2747–2758, <http://dx.doi.org/10.1109/TFUZZ.2020.2992180>.

- [14] C. Fernandez-Basso, M.D. Ruiz, M.J.M. Bautista, Spark solutions for discovering fuzzy association rules in Big Data, *Internat. J. Approx. Reason.* 137 (2021) 94–112, <http://dx.doi.org/10.1016/j.ijar.2021.07.004>.
- [15] K. Gutiérrez-Batista, J.R. Campaña, M.A.V. Miranda, M.J. Martín-Bautista, An ontology-based framework for automatic topic detection in multilingual environments, *Int. J. Intell. Syst.* 33 (7) (2018) 1459–1475, <http://dx.doi.org/10.1002/int.21986>.
- [16] K. Gutiérrez-Batista, J.R. Campaña, M.A.V. Miranda, M.J. Martín-Bautista, Building a contextual dimension for OLAP using textual data from social networks, *Expert Syst. Appl.* 93 (2018) 118–133, <http://dx.doi.org/10.1016/j.eswa.2017.10.012>.
- [17] K. Gutiérrez-Batista, M.A. Vila, M.J. Martín-Bautista, Building a fuzzy sentiment dimension for multidimensional analysis in social networks, *Appl. Soft Comput.* 108 (2021) 107390, <http://dx.doi.org/10.1016/j.asoc.2021.107390>.
- [18] A. Helmond, The platformization of the web: Making web data platform ready, *Soc. Media + Soc.* 1 (2) (2015) 2056305115603080.
- [19] D. Zburivsky, L. Partner, *Designing Cloud Data Platforms*, Simon and Schuster, 2021.
- [20] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, et al., Mllib: Machine learning in apache spark, *J. Mach. Learn. Res.* 17 (1) (2016) 1235–1241.
- [21] A. Spark, Apache spark, 17, (1) 2018, p. 2018, Retrieved January.
- [22] L. Hirth, J. Mühlendorff, M. Bulkeley, The ENTSO-E transparency platform—A review of Europe's most ambitious electricity data platform, *Appl. Energy* 225 (2018) 1054–1067.
- [23] ENTSO-E, ENTSO-E transparency platform official website, 2022, <https://www.entsoe.eu/data/transparency-platform/>. (Accessed 14 June 2022).
- [24] C. Scheidt-Nave, P. Kamtsiuris, A. Gößwald, H. Hölling, M. Lange, M.A. Busch, S. Dahm, R. Dölle, U. Ellert, J. Fuchs, et al., German health interview and examination survey for adults (DEGS)-design, objectives and implementation of the first data collection wave, *BMC Pub. Health* 12 (1) (2012) 1–16.
- [25] X. Wang, Y. Zhang, V.C. Leung, N. Guizani, T. Jiang, D2D big data: Content deliveries over wireless device-to-device sharing in large-scale mobile networks, *IEEE Wirel. Commun.* 25 (1) (2018) 32–38.
- [26] X. Hu, M. Yuan, J. Yao, Y. Deng, L. Chen, Q. Yang, H. Guan, J. Zeng, Differential privacy in telco big data platform, *Proc. VLDB Endow.* 8 (12) (2015) 1692–1703.
- [27] N. Luo, M. Pritoni, T. Hong, An overview of data tools for representing and managing building information and performance data, *Renew. Sustain. Energy Rev.* 147 (2021) 111224.
- [28] MongoDB, What is a data platform? 2021, <https://www.mongodb.com/what-is-a-data-platform>. (Accessed 31 May 2022).
- [29] F. Cauteruccio, P.L. Giudice, L. Musarella, G. Terracina, D. Ursino, L. Virgili, A lightweight approach to extract interschema properties from structured, semi-structured and unstructured sources in a big data scenario, *Int. J. Inf. Technol. Decis. Mak.* 19 (03) (2020) 849–889.
- [30] J. Chen, N. Yang, M. Zhou, Z. Zhang, X. Yang, A configurable deep learning framework for medical image analysis, *Neural Comput. Appl.* 34 (10) (2022) 7375–7392.
- [31] P. Mell, T. Grance, et al., The NIST Definition of Cloud Computing, Computer Security Division, Information Technology Laboratory, National ..., 2011.
- [32] M.D. Assunção, R.N. Calheiros, S. Bianchi, M.A. Netto, R. Buyya, Big data computing and clouds: Trends and future directions, *J. Parallel Distrib. Comput.* 79 (2015) 3–15.
- [33] P.A. Forero, A. Cano, G.B. Giannakis, Consensus-based distributed support vector machines, *J. Mach. Learn. Res.* 11 (5) (2010).
- [34] J. Chen, K. Li, Q. Deng, K. Li, S.Y. Philip, Distributed deep learning model for intelligent video surveillance systems with edge computing, *IEEE Trans. Ind. Inform.* (2019).
- [35] CPRD, Clinical practice research datalink official website, 2022, <https://cprd.com/>. (Accessed 14 June 2022).
- [36] G.A. Williams, S.M.U. Díez, J. Figueras, S. Lessof, et al., Translating evidence into policy during the COVID-19 pandemic: bridging science and policy (and politics), *Eurohealth* 26 (2) (2020) 29–33.
- [37] V. Palanisamy, R. Thirunavukarasu, Implications of big data analytics in developing healthcare frameworks—A review, *J. King Saud Univ. Comput. Inf. Sci.* 31 (4) (2019) 415–425.
- [38] C.S. Kruse, A. Stein, H. Thomas, H. Kaur, The use of electronic health records to support population health: a systematic review of the literature, *J. Med. Syst.* 42 (11) (2018) 1–16.
- [39] C. Fernandez-Basso, K. Gutiérrez-Batista, R. Morcillo-Jiménez, M.-A. Vila, M.J. Martín-Bautista, A fuzzy-based medical system for pattern mining in a distributed environment: Application to diagnostic and co-morbidity, *Appl. Soft Comput.* 122 (2022) 108870.
- [40] J. Waring, C. Lindvall, R. Umeton, Automated machine learning: Review of the state-of-the-art and opportunities for healthcare, *Artif. Intell. Med.* 104 (2020) 101822.
- [41] E. LeDell, S. Poirier, H2O automl: Scalable automatic machine learning, in: *Proceedings of the AutoML Workshop At ICML*, Vol. 2020, 2020.
- [42] B. Raef, R. Ferdousi, A review of machine learning approaches in assisted reproductive technologies, *Acta Inform. Medica* 27 (3) (2019) 205.
- [43] W. McKinney, pandas: powerful Python data analysis toolkit, 2008, URL <https://pandas.pydata.org/>.
- [44] Oracle, Oracle corporation, 1977, URL <https://www.oracle.com/>.
- [45] GitHub, GitHub, inc., 2008, URL <https://github.com/>.
- [46] M. Grinberg, Flask API-RESTful, 2013, URL <https://flask-api-restful.readthedocs.io/en/latest/>.
- [47] MongoDB, Inc., MongoDB, 2007, URL <https://www.mongodb.com/>.
- [48] Docker, Docker official website, 2022, <https://www.docker.com/>. (Accessed 27 June 2022).
- [49] Amazon.com, Inc., Amazon web services, 2006, URL <https://aws.amazon.com/>.
- [50] Google, Google LLC, 1998, URL <https://www.google.com/>.
- [51] Microsoft Corporation, Microsoft azure, 2010, URL <https://azure.microsoft.com/>.
- [52] W.H. Inmon, OLAP cubes, *Commun. ACM* 39 (9) (1996) 90–98, <http://dx.doi.org/10.1145/237257.237273>.
- [53] C. Fernandez-Basso, M.D. Ruiz, M.J. Martín-Bautista, Extraction of association rules using big data technologies, *Int. J. Des. Nat. Ecodynamics* 11 (3) (2016) 178–185.
- [54] Z. Zhang, J. Jiang, W. Wu, C. Zhang, L. Yu, B. Cui, Mllib*: Fast training of glms using spark mllib, in: 2019 IEEE 35th International Conference on Data Engineering, ICDE, IEEE Computer Society, 2019, pp. 1778–1789.
- [55] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, et al., Xgboost: extreme gradient boosting, 1, (4) 2015, pp. 1–4, R package version 0.4-2.
- [56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, *J. Mach. Learn. Res.* 12 (2011) 2825–2830.
- [57] M.A.B. HajKacem, C.E.B. N'Cir, N. Essoussi, KP-S: a spark-based design of the K-prototypes clustering for big data, in: 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications, AICCSA, IEEE, 2017, pp. 557–563.
- [58] J. Kim, H. Ha, B.-G. Chun, S. Yoon, S.K. Cha, Collaborative analytics for data silos, in: 2016 IEEE 32nd International Conference on Data Engineering, ICDE, IEEE, 2016, pp. 743–754.
- [59] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, Y. Liu, {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning, in: 2020 USENIX Annual Technical Conference, USENIX ATC 20, 2020, pp. 493–506.
- [60] SAS, Minimum basic data set (MBDS) of Andalusia, 2022, <https://www.sspa.juntadeandalucia.es/servicioandaluzdesalud/profesionales/sistemas-de-informacion/cmbd-andalucia>. (Accessed 20 June 2022).
- [61] WHO, ICD-10 - International statistical classification of diseases and related health problems 10th revision, 2022, <https://icd.who.int/browse10/2019/en/#/>. (Accessed 27 June 2022).
- [62] C. Fernandez-Basso, A.J. Francisco-Agra, M.J. Martín-Bautista, M.D. Ruiz, Finding tendencies in streaming data using big data frequent itemset mining, *Knowl.-Based Syst.* 163 (2019) 666–674.
- [63] C. Fernandez-Basso, M.D. Ruiz, M.J. Martín-Bautista, A fuzzy mining approach for energy efficiency in a Big Data framework, *IEEE Trans. Fuzzy Syst.* 28 (11) (2020) 2747–2758.
- [64] M.D. Ruiz, D. Sánchez, M. Delgado, M.J. Martín-Bautista, Discovering fuzzy exception and anomalous rules, *IEEE Trans. Fuzzy Syst.* 24 (4) (2015) 930–944.
- [65] K. Gutiérrez-Batista, J.R. Campaña, M.-A. Vila, M.J. Martín-Bautista, An ontology-based framework for automatic topic detection in multilingual environments, *Int. J. Intell. Syst.* 33 (7) (2018) 1459–1475.
- [66] I. Diaz-Valenzuela, V. Loia, M.J. Martín-Bautista, S. Senatoro, M.A. Vila, Automatic constraints generation for semisupervised clustering: experiences with documents classification, *Soft Comput.* 20 (6) (2016) 2329–2339.
- [67] C. Fernandez-Basso, M.D. Ruiz, M.J. Martín-Bautista, Spark solutions for discovering fuzzy association rules in Big Data, *Internat. J. Approx. Reason.* 137 (2021) 94–112.



Alberto S. Ortega-Calvo received his Computer Engineering degree and Master's degree in Data Science in 2020 and 2021, respectively. He completed both at the University of Granada, for which he is currently actively researching together with the IdBIS (Intelligent Databases and Information Systems) research group on projects such as P18-RT-1765, whose main topics are Big Data, medical data analysis and machine learning. He is currently posing a thesis on topics related to medical data processing and federated learning.



Roberto Morcillo Jiménez received a degree in Computer Engineering and M. in Computer Engineering from the University of Granada, Spain. He worked as an Assistant Professor in the Department of Computer Science and Artificial Intelligence between 2016 and 2019. He is doing a Ph.D. in Computer Science and Artificial Intelligence at the University of Granada from 2019. He is currently working as an Assistant Professor in the Department of Computer Languages and Systems. He is Associated Research of Intelligent Data Bases and Information Systems (IDBIS) research group at the University of Granada. His research interests comprise Deep Learning, Reinforcement Learning, Building Energy Efficiency and Control, and Information Systems.



Carlos Fernandez-Basso received the degree in computer science, the M.Sc. degree in data science, and the Ph.D. degree in computer science from the University of Granada, Granada, Spain, in 2014, 2015, and 2020, respectively. He is currently a Postdoctoral Fellow with Causal Cognition Lab, University College London, London, U.K. He was a Lead Developer in the EU FP7 Project Energy IN TIME in the topics of building simulation and control, data analytics, and machine learning, and in the COPKIT Project in the topics of cybercrime, Big Data, and machine learning. From 2016 to 2018, he collaborated with the Data Science Institute, Imperial College London, London, U.K., where he has carried out research stays.



Karel Gutiérrez-Batista received a degree in computer science and an M.Sc. degree in data science from the University of Camagüey, Cuba. He worked as an Assistant Professor between 2009 and 2014 at the Department of Computer Science at the University of Camagüey. He received his Ph.D. in computer science in 2018 from the University of Granada, Spain. He is currently working as a Postdoctoral Fellow in the Department of Computer Science and Artificial Intelligence at the University of Granada, Spain. He is an Associated Research of Intelligent Data Bases and Information Systems (IDBIS) research group at the University of Granada. His research interests comprise Multidimensional Data Analysis, Deep Learning, Data Mining, and Natural Language Processing.



Maria-Amparo Vila Miranda is Professor of Computer Science and Artificial Intelligence, University of Granada since 1992, she was previously associate professor and analyst programmer of the Computer Centre at the same university. She has developed her research and teaching primarily in the area of databases and intelligent information systems, and its main lines of research are : treatment of imprecision in information systems using fuzzy logic, knowledge discovery in databases using techniques "Soft Computing"-based, and ubiquitous computing and knowledge mobilization. She has been the advisor of 27 doctoral theses, she has been or is responsible for more than 10 research projects and she has published numerous research papers, which highlights more than 100 papers in SCI journals. She has been responsible for the research group of the Andalusian ICT-113 Approximate Reasoning and Artificial Intelligence) from 1994 to 1997 and head of the group TIC174 (Databases and Intelligent Information Systems) since its inception in 2000 until today . From the point of view of university management she has been Vice-Director of Organization at the School of Computer Science for 3 years (1994-1997) and Director of the Department of Computer Science and Artificial Intelligence, University of Granada from 1997 to 2004. She has also developed related tasks and quality assessment processes in this regard has been: a member of the speech area of Information Technology and Communications in the Andalusian Research Plan III, II and evaluator within the Quality Plan Universities (University Council ersities MEC) has been evaluating also for ANECA within the institutional evaluation program, having served as president of various committees of external and Commissioner Assessment Andalusian Autonomous Accessories (CAECA) being responsible and chairwoman of the subcommittee of the area of Technical Education's. She has been also chairman of the evaluation committee of undergraduate Architecture and Engineering within the program VERIFICA of ANECA. This committee is responsible for assessments and guidelines teachers of all grades of Computer Engineering and Telecommunications have been evaluated in Spain.



Dr. Maria J. Martin-Bautista is a Full Professor at the Department of Computer Science and Artificial Intelligence at the University of Granada, Spain, since 1997. She is a member of the IDBIS (Intelligent Data Bases and Information Systems) research group. Her current research interests include Data Science and Big Data Analytics in Data, Text, Web and Social Networks, Intelligent Information Systems, Knowledge Representation and Uncertainty. She has supervised several Ph.D. Thesis and published more than 100 papers in high impact international journals and conferences. She has participated in more than 20 R+D projects and has supervised several research technology transfers with companies. She has served as a program committee member for several international conferences.

5 Conclusiones y Trabajos Futuros

La implementación exitosa de estos objetivos aseguró la calidad de los resultados obtenidos plasmados en tres artículos JCR y otro enviado a la revista Applied Intelligence. El desarrollo de una herramienta sólida, eficiente y accesible para la resolución de problemas específicos en el campo de la Ciencia de Datos requiere de un enfoque riguroso y multidisciplinario, que permita ejecutar una metodología, la estructuración de los datos y la plataforma de datos de manera sólida y segura.

Al cumplir con los objetivos de esta tesis doctoral, se podría lograr una herramienta valiosa para el avance del conocimiento científico en diferentes áreas de la investigación, permitiendo a los investigadores ejecutar experimentos con total libertad y sin necesidad de profundizar en el conocimiento de la Ciencia de Datos o de los propios algoritmos encargados del tratamiento de los datos.

Sin embargo, como se ha demostrado en el caso de uso [24], una gran cantidad de datos no se utilizan debido a restricciones de seguridad y anonimato. En este sentido, se pueden utilizar en un futuro varios enfoques para abordar este problema:

- Aprendizaje federado [26], permite la extracción de conocimiento de los datos de manera descentralizada y encriptada. Incluyendo una capa adicional que se encargue de este proceso, se preserva la modularidad de la plataforma y se facilita su desarrollo y mantenimiento.
- Integración de los datos, en otros trabajos se han implementado nuevos métodos para el enriquecimiento y estructuración de los datos como en [4]. Además, estos métodos persiguen el objetivo de reducir el tiempo de respuesta, lo que los hace más adecuados para entornos de Big Data. Trabajar en esta dirección en el futuro proporcionará a AIMDP una mayor flexibilidad y se podrá avanzar aún más en uno de los puntos fuertes de la plataforma: el tratamiento de fuentes de datos heterogéneos.
- Usabilidad del sistema, hemos visto cómo un sistema amigable e intuitivo para el usuario final mejora la comprensión, el uso y la productividad de sus investigaciones. Sin embargo, todavía existen desafíos en el campo del uso de herramientas en Ciencia de Datos por parte de usuarios no expertos. Por lo tanto, como mejoras al sistema, se podría implementar una capa de XAI[3] (Explainable Artificial Intelligence, Inteligencia Artificial Explicativa) para mejorar la interpretabilidad y explicabilidad de muchos de los métodos disponibles en nuestra herramienta.
- Mejorar la selección de los datos y la configuración de los parámetros de los algoritmos mediante la adición de un proceso de AutoML [28] (Automatic Machine Learning, Aprendizaje Automático Automatizado) para seleccionar los mejores parámetros y algoritmos para que los usuarios obtengan los mejores resultados para su experimentación.

Bibliografía

- [1] Rakesh Agrawal, Tomasz Imieliński, y Arun Swami. Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, 22(2):207 – 216, 1993.
- [2] Marcos D Assunção, Rodrigo N Calheiros, Silvia Bianchi, Marco AS Netto, y Rajkumar Buyya. Big data computing and clouds: Trends and future directions. *Journal of parallel and distributed computing*, 79:3–15, 2015.
- [3] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, y Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82 – 115, 2020. Cited by: 2623; All Open Access, Green Open Access.
- [4] Francesco Cauteruccio, Paolo Lo Giudice, Lorenzo Musarella, Giorgio Terracina, Domenico Ursino, y Luca Virgili. A lightweight approach to extract interschema properties from structured, semi-structured and unstructured sources in a big data scenario. *International Journal of Information Technology & Decision Making*, 19(03):849–889, 2020.
- [5] Bin Cheng, Salvatore Longo, Flavio Cirillo, Martin Bauer, y Ernoe Kovacs. Building a big data platform for smart cities: Experience and lessons from santander. En *2015 IEEE International Congress on Big Data*, páginas 592–599, 2015.
- [6] Mark Coeckelbergh. *AI Ethics*. Springer, 1st edition, mayo 2023.
- [7] Carlos Fernandez-Basso, Karel Gutiérrez-Batista, Roberto Morcillo-Jiménez, Maria-Amparo Vila, y Maria J. Martin-Bautista. A fuzzy-based medical system for pattern mining in a distributed environment: Application to diagnostic and co-morbidity. *Applied Soft Computing*, 122:108870, June 2022.
- [8] Iztok Fister, Iztok Fister, Dušan Fister, Vili Podgorelec, y Sancho Salcedo-Sanz. A comprehensive review of visualization methods for association rule mining: Taxonomy, challenges, open problems and future ideas. *Expert Systems with Applications*, 233, 2023.
- [9] Aurelien Geron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow 3e: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 3rd edition, noviembre 2022.
- [10] Anne Helmond. The platformization of the web: Making web data platform ready. *Social media+ society*, 1(2):2056305115603080, 2015.
- [11] Erik P. Hess, George A. Wells, Allan Jaffe, y Ian G. Stiell. A study to derive a clinical decision rule for triage of emergency department patients with chest pain: Design and methodology. *BMC Emergency Medicine*, 8, 2008.
- [12] Xueyang Hu, Mingxuan Yuan, Jianguo Yao, Yu Deng, Lei Chen, Qiang Yang, Haibing Guan, y Jia Zeng. Differential privacy in telco big data platform. *Proc. VLDB Endow.*, 8(12):1692–1703, aug 2015.
- [13] Brendan Tierney John D Kelleher. *Data Science*. The MIT Press, illustrated edition, April 2018.
- [14] Timothy Kariotis, Mad Price Ball, Bastian Greshake Tzovaras, Simon Dennis, Tony Sahama, Carolyn Johnston, Helen Almond, y Ann Borda. Emerging health data platforms: From individual control to collective data governance. *Data & Policy*, 2:e13, 2020.
- [15] Jinkyu Kim, Heonseok Ha, Byung-Gon Chun, Sungroh Yoon, y Sang K. Cha. Collaborative analytics for data silos. En *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, páginas 743–754, 2016.

- [16] Holger Kruse y Jiří Šponer. Highly accurate equilibrium structure of the c2h symmetric n1-to-o2 hydrogen-bonded uracil-dimer. *International Journal of Quantum Chemistry*, 118(15):1, 2018. Cited by: 7.
- [17] Yi Liu, Jiawen Peng, y Zhihao Yu. Big data platform architecture under the background of financial technology: In the insurance industry as an example. En *Proceedings of the 2018 International Conference on Big Data Engineering and Technology*, BDET 2018, página 31–35, New York, NY, USA, 2018. Association for Computing Machinery.
- [18] Na Luo, Marco Pritoni, y Tianzhen Hong. An overview of data tools for representing and managing building information and performance data. *Renewable and Sustainable Energy Reviews*, 147:111224, 2021.
- [19] Viktor Mayer-Schönberger y Kenneth Cukier. *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. John Murray Publishers Ltd, octubre 2013.
- [20] Bożena Małysiak-Mrozek, Marek Stabla, y Dariusz Mrozek. Soft and declarative fishing of information in big data lake. *IEEE Transactions on Fuzzy Systems*, 26(5):2732–2747, 2018.
- [21] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, et al. Mllib: Machine learning in apache spark. *The Journal of Machine Learning Research*, 17(1):1235–1241, 2016.
- [22] Roberto Morcillo-Jimenez, Karel Gutiérrez-Batista, y Juan Gómez-Romero. TSxtend: A tool for batch analysis of temporal sensor data. *Energies*, 16(4):1581, February 2023.
- [23] Roberto Morcillo-Jimenez, Karel Gutiérrez-Batista, y Juan Gómez-Romero. Tsxtend: A tool for batch analysis of temporal sensor data. *Energies*, 16(4), 2023.
- [24] Alberto S. Ortega-Calvo, Roberto Morcillo-Jimenez, Carlos Fernandez-Basso, Karel Gutiérrez-Batista, Maria-Amparo Vila, y Maria J. Martin-Bautista. Aimdp: An artificial intelligence modern data platform. use case for spanish national health service data silo. *Future Generation Computer Systems*, 143:248–264, 2023.
- [25] Venketesh Palanisamy y Ramkumar Thirunavukarasu. Implications of big data analytics in developing healthcare frameworks – a review. *Journal of King Saud University - Computer and Information Sciences*, 31(4):415–425, 2019.
- [26] Sharnil Pandya, Gautam Srivastava, Rutvij Jhaveri, M. Rajasekhara Babu, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, Spyridon Mastorakis, Md. Jalil Piran, y Thippa Reddy Gadekallu. Federated learning for smart cities: A comprehensive survey. *Sustainable Energy Technologies and Assessments*, 55:102987, 2023.
- [27] Seyedmostafa Safavi, Seyed Mohammad Javadi, y Ahmad Khonsari. Big data platforms: a survey. *Journal of Big Data*, 6(1):79, 2019.
- [28] J. Enrique Sierra-Garcia y Matilde Santos. Federated discrete reinforcement learning for automatic guided vehicle control. *Future Generation Computer Systems*, 150:78 – 89, 2024.
- [29] Pang-Ning Tan. *Introduction to Data Mining, Global Edition*. Financial Times Prentice Hall, n.º 1 edition, marzo 2019.
- [30] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Pr, n.º 2 edition, abril 2001.
- [31] Xiaofei Wang, Yuhua Zhang, Victor CM Leung, Nadra Guizani, y Tianpeng Jiang. D2d big data: Content deliveries over wireless device-to-device sharing in large-scale mobile networks. *IEEE Wireless Communications*, 25(1):32–38, 2018.
- [32] Gemma A Williams, Sara M Ulla Díez, Josep Figueras, Suszy Lessof, et al. Translating evidence into policy during the covid-19 pandemic: bridging science and policy (and politics). *Eurohealth*, 26(2):29–33, 2020.

- [33] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, y Ion Stoica. Spark: cluster computing with working sets. En *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, páginas 10–10. USENIX Association, 2010.
- [34] Danil Zburivsky y Lynda Partner. *Designing Cloud Data Platforms*. Simon and Schuster, 2021.