UNIVERSIDAD
DE GRANADA

TOURISM MANAGEMENT IN SMART VILLAGES: DEVELOPMENT OF A METHODOLOGY WITH SENSORS AND MACHINE LEARNING

DANIEL BOLAÑOS MARTÍNEZ

Doctoral Dissertation

Programme in Information and Communication Technologies

PhD Advisor

María Bermúdez Edo

DEPARTMENT OF SOFTWARE ENGINEERING

SCHOOL OF TECHNOLOGY AND TELECOMMUNICATIONS ENGINEERING

UNIVERSIDAD DE GRANADA

Tourism Management in Smart Villages: Development of a Methodology with Sensors and Machine Learning

DOCTORAL DISSERTATION
presented to obtain the
DOCTOR OF PHILOSOPHY DEGREE
in the
Doctoral Programme in Information and Communication Technologies
by

Daniel Bolaños Martínez

PhD Advisor María Bermúdez Edo
Department of Software Engineering
School of Technology and Telecommunications Engineering

Granada, Thursday 1st May, 2025

Editor: Universidad de Granada. Tesis Doctorales
Autor: Daniel Bolaños Martínez
ISBN: 978-84-1195-908-7
URI: https://hdl.handle.net/10481/108583

A mis abuelos, siempre conmigo.

ACKNOWLEDGMENTS

Aunque normalmente no me gusta extenderme en los agradecimientos, siempre disfruto reflexionando sobre las experiencias y personas que, de forma directa o indirecta, contribuyen a la elaboración de un trabajo final de estudios -y una tesis doctoral, no es para menos-. Quisiera comenzar expresando mi gratitud a mi familia, ya que sin ella no sería quien soy. Especialmente a mi padre, por inculcarme la pasión por la docencia universitaria; a mi madre, por su constante cariño; y a mi hermano, por todos los buenos momentos que compartimos.

Para mí, este trabajo comenzó en 2021, un año repleto de desafíos. Tenía aún pendientes algunas asignaturas del DGIIM, el TFG sin empezar, España enfrentaba una nueva variante del COVID y Granada vivía un enjambre sísmico no visto desde hace décadas. Los nervios me impedían concentrarme, por lo que decidí refugiarme en Cástaras, un pequeño pueblo de la Alpujarra donde tenemos una casa. Esa experiencia me permitió centrarme en mis proyectos, ganar independencia personal, y compartir momentos muy especiales con mi abuela durante los fines de semana, cuando podía trasladarse desde Granada.

Esa conexión con la Alpujarra me impulsó, en 2022, a postularme para un contrato de investigación, sugerido por mi amigo Pedro, a quien siempre agradeceré por ese gesto. Conseguí el puesto, y una vez dentro, tuve el privilegio de conocer a personas extraordinarias, tanto en el ámbito académico como en el personal. Agradezco al equipo de Smart Poqueira por su cálida acogida: a Alberto, por su sencillez y liderazgo; a Blanca, por su simpatía y compañerismo; a José Luis, por su templanza y valiosos consejos; y a Julián, por ser el mejor compañero de rutas que podría desear.

No puedo dejar de mencionar a María, quien ha sido tanto tutora como directora de tesis, y gracias a quien me he podido formar como investigador. Es, sin duda, una de las personas más trabajadoras que he tenido el placer de conocer. Aunque nuestras exigencias nos llevan en ocasiones a tensiones, siempre logramos encontrar puntos de acuerdo. Sin su apoyo, este trabajo y mis primeros años como docente hubieran sido mucho más difíciles.

Finalmente, agradezco de corazón a mis amigos por su apoyo emocional. A Alberto Durán, por ser no solo un gran amigo de quedadas, sino también un excelente compañero de trabajo con el que espero compartir muchos años en el departamento. A Elio, gracias a quien las semanas son mucho más llevaderas. A mi grupo de amigos del instituto -David, Inma, Miguel y Yeray-; y a mis "colegas de Telegram", antiguos compañeros de universidad y amigos que, a pesar de estar fuera de Granada, son un apoyo fundamental en mi día a día.

ABSTRACT

Machine learning (ML) uses algorithms to analyze data features and identify patterns. Deep learning (DL) is a subset of ML that uses neural networks (NN) to analyze complex relationships, often outperforming traditional models. In the DL field, transformer-based architectures introduce attention mechanisms to improve model performance. Transformers excel in time-series forecasting, and image processing. A few works have adapted transformers for tabular data, but remain ineffective for small datasets. ML and DL models are implemented through ML pipelines, consisting of different steps: data collection, validation, and preprocessing; followed by model training, tuning, evaluation, and visualization; and ending with the model deployment. The validation and preprocessing steps include feature selection and normalization, respectively. Feature selection determines the most influential features, while normalization ensures consistency in data distribution. Key challenges identified in this thesis include: (1) The lack of transformers adapted for small tabular datasets, which is important in contexts with limited data collection, such as questionnaires. (2) Within the data validation step, feature selection in ML pipelines is often focused on individual features rather than entire datasets, making resource allocation decisions difficult. (3) Within the data preprocessing step, normalization methods are applied without proper assessment, despite their impact on model accuracy and explainability.

This work presents an ML pipeline that integrates data from various sources and fuses them. During the data validation step, we conduct an ablation study at the dataset level to assess the influence of each data source on the tested models, thereby addressing the challenge (2). In the data preprocessing step, we apply different normalization methods to analyze their impact on model performance, addressing challenge (3). Finally, we develop a transformer-based model with multiple attention layers specifically designed for limited data and integrate it during the model training/tuning step to tackle the challenge (1). Additionally, we evaluate other models for comparison and draw conclusions in the evaluation and visualization step.

The study implements the pipeline in a case study on smart villages, an adaptation of the smart city concept to rural communities. While smart cities have been the subject of numerous studies, smart villages are an emerging concept that has attracted attention but remains underexplored. The literature highlights that solutions designed for large cities may not be directly applicable to small villages. One significant difference between smart cities and smart villages is their population size and infrastructure. Unlike cities, where data are more easily collected through both automated systems and manual records, villages typically produce much smaller datasets. This limitation

makes it challenging to apply DL models effectively, as they require vast amounts of data to achieve reliable performance.

The results of this thesis provide insights into ML pipelines for vehicle mobility in smart villages. We deploy IoT devices, including LPR cameras, to collect vehicle behavior and contextual data such as holiday calendars, socio-economic factors, and visitor demographics. Unlike prior studies, we integrate LPR data with contextual information to improve clustering analysis. We evaluate normalization methods and their impact on cluster interpretation, identifying behavioral patterns that differentiate residents from visitors. After identifying vehicle patterns, we propose supervised classification tasks to predict different vehicles' behaviors. For example, we propose a model that predicts how many nights a visitor spends in the area. Using this idea as a basis, we propose an ablation study at the dataset level, evaluating the level of improvement of the resulting models. Unlike conventional ablation studies, which focus on assessing the contribution of NN layers, our approach analyses the impact of datasets composed of different features from a common information source. Finally, to address data scarcity, we develop a transformer that combines vehicle data with visitor questionnaires, predicting repeat tourist visits.

This proposal guides researchers in selecting validation and preprocessing techniques, such as feature selection and normalization. It advances research by applying transformers to small tabular datasets, improving predictive models in data-limited scenarios. It also helps stakeholders in smart villages analyze mobility patterns. Normalization reveals distinct visitor clusters, providing insights for strategies to promote overnight stays and encourage non-registered residents to register. An ablation study identifies socio-economic status and entry points as key predictors of visitor overnights, optimizing data use in tourism forecasting. Finally, our transformer model, which tracks repeat tourists, could aid urban planning and transport policies.

RESUMEN EXTENDIDO EN ESPAÑOL

En la Parte I de esta tesis doctoral, se presenta la introducción, fundamentos, objetivos, metodología, resultados, conclusiones y trabajo futuro del proyecto realizado. En la Parte II se presentan las tres publicaciones que conforman la tesis por compendio. A continuación, se ofrece un resumen extendido en español de los contenidos de ambas partes, centrandonos en la motivación, los objetivos las contribuciones (junto con un resumen de las publicaiones con sus resultados) y las conclusiones.

MOTIVACIÓN

El aprendizaje automático o *machine learning* (ML) utiliza algoritmos que analizan datos para detectar patrones e inferir relaciones en diversos ámbitos. Una de las taxonomías más populares, distingue en tres clases de ML: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo [1, 2]. El aprendizaje supervisado, utiliza datos etiquetados para tareas como clasificación. Por ejemplo, predecir si un turista volverá a una ciudad [3]. El aprendizaje no supervisado, mediante técnicas como el *clustering*, agrupa datos sin etiquetar para revelar patrones ocultos. Por ejemplo, encontrar comportamientos comunes entre los vehículos que se mueven por una carretera [4]. El aprendizaje por refuerzo, donde un agente aprende a tomar decisiones óptimas mediante la interacción con un entorno y la retroalimentación de recompensas. Por ejemplo, entrenar un vehículo para conducir de forma autónoma evitando obstáculos [5]. Aunque hemos seguido esta taxonomía, existen taxonomías alternativas que también se utilizan habitualmente en la actualidad. Por ejemplo, el aprendizaje profundo o *deep learning* (DL) y aprendizaje superficial o *shallow learning* (SL) [6]. En la última década, el DL ha potenciado ML con redes neuronales o *neural networks* (NN) basadas en arquitecturas multicapa (lo que le da la profundidad al DL), capaces de analizar relaciones complejas y mejorar la eficacia de algoritmos tradicionales de SL en campos como educación, salud y turismo [7, 8, 9], aunque con la limitación de requerir grandes volúmenes de datos para su entrenamiento.

Un avance reciente en el campo de DL, son los *transformers*, los cuales fueron desarrollados inicialmente para el procesamiento de lenguaje natural, pero también se han aplicado al campo de imágenes y series temporales [10, 11]. Los *transformers*, integran mecanismos de atención que permiten capturar dependencias de largo alcance y relaciones contextuales [12], algo que las NN tradicionales no logran con la misma eficacia [13, 14]. Recientemente, su uso se ha expandido a problemas de datos tabulares, que incluyen datos estructurados en filas y columnas [15, 16]. Los mecanismos de atención podrían ayudar a las NN a adquirir conocimiento a partir de conjuntos

de datos pequeños, como por ejemplo ya se ha comprobado para el caso del aprendizaje de imágenes [10, 17]. No obstante, aunque se han desarrollado mecanismos de atención para datos tabulares [15, 16], estas soluciones aún no están adaptadas a conjuntos de datos limitados [18].

Para aplicar modelos de ML, incluidos modelos de DL, se desarrollan *pipelines*, que incluyen una secuencia de procesos que comienza con la recopilación de datos y termina con el despliegue del modelo [19]. Este proceso (ver Figura 1) consta de tres fases generales:

i. Adquisición y preparación de datos.
   - Recolección de datos, donde los datos se recopilan, formatean y almacenan.
   - Validación de datos, que garantiza la calidad y coherencia de los datos entrantes mediante el análisis exploratorio de la distribución de los datos, la detección de anomalías o selección de características.
   - Preprocesamiento de datos, que transforma los datos en un formato adecuado para el entrenamiento, utilizando técnicas como normalización o reducción de la dimensionalidad.

ii. Entrenamiento y optimización del modelo.
   - Entrenamiento del modelo, donde los algoritmos aprenden a partir de los datos.
   - Ajuste del modelo, donde se optimizan los hiperparámetros para mejorar la precisión y/o el rendimiento.
   - Evaluación y visualización del modelo, donde el modelo entrenado se evalúa utilizando métricas como precisión y exactitud. En este paso, también es posible aplicar métodos de visualización de resultados para refinar aún más el modelo y para explicarlo.

iii. Despliegue del modelo evaluado para su uso.
   - El paso final implica desplegar el modelo evaluado, el cual será refinado y actualizado continuamente con nuevos datos, reiniciando el ciclo del proceso.

En la fase de adquisición y preparación de datos (i), dos tareas importantes son la selección de características y la normalización. La selección de características evalúa el impacto y peso de cada característica en el proceso de aprendizaje e identifica las características más influyentes en el modelo [20, 21, 22]. Esta fase, junto con la visualización, puede ayudar a determinar la explicabilidad del modelo. Aunque algunos estudios aplican selección de características para elegir las más influyentes en el modelo [23, 24]. Sin embargo, no se considera la importancia de hacer este análisis a nivel de conjunto de datos en lugar de solo características individuales. En el contexto actual, donde se valoran cada vez más las prácticas de ML eficientes y sostenibles [25], realizar estudios de ablación sobre cada conjunto de datos puede indicar si vale la pena invertir tiempo y recursos en la recopilación de datos de una fuente específica

[26, 27]. Respecto a la normalización, ha recibido poca atención, ya que los investigadores suelen aplicar un solo método de normalización sin evaluar su idoneidad. No obstante, esta fase es importante, ya que cada técnica de normalización transforma la escala y distribución de los datos, afectando el rendimiento del algoritmo de ML seleccionado y alterando sus resultados. Diferentes salidas para un mismo algoritmo pueden generar inconsistencias al interpretar los resultados y al analizar la explicabilidad de los modelos obtenidos [28].

Por lo tanto, los desafíos generales de ML identificados que esta tesis busca abordar son: (1) en el campo de DL, no existen transformers adaptados a pequeños conjuntos de datos tabulares, una limitación particularmente significativa en aplicaciones donde la recopilación de datos es restringida, como los estudios basados en cuestionarios con un número limitado de respuestas. Dentro de la fase de adquisición y preparación de datos en un pipeline de ML que integra conjuntos de datos contextuales: (2) la selección de fuentes de información representa un desafío, ya que la mayoría de los estudios en la literatura se centran en la selección de características individuales en lugar de evaluar conjuntos de datos completos de cada fuente, lo que podría ayudar a decidir si vale la pena invertir en una fuente de datos específica; y (3) la tarea de normalización no ha recibido suficiente atención, a pesar de ser determinante para la explicabilidad del modelo.

Nuestra propuesta es construir un pipeline de ML que integre datos de diversas fuentes y los fusione una vez recopilados. Durante la fase de validación de datos, realizamos un estudio de ablación a nivel de conjunto de datos para examinar la influencia de cada fuente en los modelos que probamos, abordando el desafío (2). En la fase de preprocesamiento de datos, empleamos distintos métodos de normalización para explorar su impacto en los resultados del modelo, abordando el desafío (3). Finalmente, desarrollamos un modelo basado en transformers con diferentes capas de atención específicamente diseñadas para datos limitados para abordar el desafío (1). Además, probamos otros modelos para compararlos y extraer conclusiones durante la fase de evaluación y visualización.

Para validar nuestras propuestas, hemos utilizado un caso de estudio de smart villages, un concepto derivado de las smart cities que ha cobrado impulso en la última década, especialmente desde la iniciativa "The EU Action for Smart Villages" en 2017 [29, 30]. Mientras que las ciudades inteligentes han sido ampliamente estudiadas, los pueblos inteligentes presentan desafíos particulares debido a su menor infraestructura y la generación de conjuntos de datos más reducidos. Esto dificulta la aplicación directa de soluciones urbanas y de modelos de aprendizaje profundo, que suelen requerir grandes volúmenes de datos. Nuestra propuesta aborda estas limitaciones adaptando tecnologías de Internet de las Cosas (IoT en inglés) y aprendizaje automático a entornos rurales, optimizando la gestión de recursos y ayudando a la toma de decisiones en estos contextos.

OBJETIVOS

El objetivo de esta tesis es desarrollar herramientas y metodologías de ML para optimizar la construcción de *pipelines* de ML, con un enfoque específico en la gestión de datos turísticos en smart villages. Para ello, nos planteamos los siguientes objetivos:

1. Adaptar la recolección de datos de ciudades inteligentes a pueblos, ajustándola a entornos con limitaciones de datos y recursos.
2. Combinar datos de sensores con información contextual (festivos, procedencia, factores socioeconómicos) para enriquecer el análisis de datos.
3. Explorar métodos de normalización y su influencia en el desempeño de algoritmos de ML según la distribución de datos.
4. Identificar los conjuntos de datos más importantes mediante estudios de ablación, que permitan equilibrar rendimiento y costes de recolección de datos.
5. Desarrollar arquitecturas de DL que integren mecanismos de atención y funcionen con conjuntos de datos limitados.
6. Validar los modelos de ML generados sobre un caso de estudio de turismo rural. Así como aplicarlos a problemas o situaciones reales.

CONTRIBUCIONES

Para abordar los distintos objetivos, en un primer estudio, definimos la infraestructura de recogida de datos que incluye la red sensórica de cámaras LPR (*License Plate Recognition*) y la definición de las distintas fuentes de datos heterogéneas. Además, en este estudio empleamos algoritmos de *clustering* para analizar patrones en los datos, y descubrir que pueden decirnos los datos que tenemos. Para construir el *pipeline* de ML, realizamos análisis exploratorio durante la validación, seleccionamos algoritmos que se ajusten a la distribución de los datos, y preprocesamos los datos mediante diferentes técnicas de normalización. En un segundo estudio, abordamos un problema de clasificación. Realizamos estudios de ablación para encontrar conjuntos de datos que optimicen el modelo, reduciendo el número de características, y reduciendo el tiempo de entrenamiento. A diferencia de los estudios de ablación convencionales, que se centran en evaluar la contribución de las capas NN, nuestro enfoque analiza el impacto de los conjuntos de datos compuestos por diferentes características procedentes de una fuente de información común. Se genera un modelo que predice cuántas noches pasará un visitante en la zona, midiendo cómo influye cada conjunto/fuente de datos. Finalmente, en un tercer estudio, abordamos la limitación de datos típica en áreas rurales mediante la propuesta y creación de una red neuronal con atención, capaz de manejar conjuntos de datos pequeños, integrando cuestionarios de visitantes y datos de comportamiento vehicular. Así, buscamos predecir qué turistas regresarán a corto plazo.

Por lo tanto, las contribuciones de esta tesis se pueden resumir en:

- Creación de un dataset que fusiona datos de vehículos detectados por cámaras LPR, con datos de contexto socioeconómicos, geográficos, y procedentes de calendarios de festivos nacionales.
- Exploración de diferentes métodos de normalización y su influencia en algoritmos de ML según la distribución de los datos.
- Desarrollo de estudios de ablación para identificar los conjuntos de datos más importantes que permitan equilibrar rendimiento y costes de recolección de datos.
- Propuesta y validación de una nueva arquitectura de *transformer*, o red neuronal con atención, personalizada para conjuntos de datos pequeños.
- Creación de un modelo de *clustering* para identificar a residentes y turistas en varios grupos según su comportamiento.
- Creación de un modelo predictivo del número de noches de estancia de los vehículos que ingresan en una zona rural.
- Creación de un modelo predictivo para diferenciar a los visitantes que volverán a visitar una zona rural de los que no.

Breve resumen artículos del compendio

A continuación, se ofrece un breve resumen de las publicaciones que contienen los resultados de los estudios que abordan los objetivos mencionados anteriormente y que se adjuntan en la Parte II de esta tesis:

Clustering Pipeline for Vehicle Behavior in Smart Villages

Este trabajo desarrolla un *pipeline* de ML usando algoritmos de *clustering* para analizar la movilidad vehicular en una zona de turismo rural, combinando datos de sensores LPR con fuentes contextuales heterogéneas. El *pipeline* propuesto abarca ocho fases: recolección, limpieza, fusión, normalización, reducción dimensional, algoritmos de *clustering*, evaluación y visualización. Aunque el ML *pipeline* presentado anteriormente constaba de siete fases. En nuestra propuesta, hemos descartado la fase de despliegue del modelo, al encontrarse fuera del ámbito de la tesis. Además, hemos subdividido otras fases, como la de normalización, reducción de dimensionalidad (presentes en la fase de preprocesamiento) o la de visualización (separada de la fase de evaluación). Estas divisiones nos permiten analizar de forma más precisa las técnicas presentes en cada fase dentro de nuestro pipeline. La fase de validación por ejemplo, la hemos subdividido en dos para hacer frente a algunos desafíos como la fusión de fuentes heterogéneas o la limpieza de datos, los cuales son importantes en nuestro caso de estudio.

Durante nueve meses, se recopilan más de 50.000 vehículos con cuatro sensores LPR, junto con datos contextualizados de festivos basados en calendarios nacionales y locales, procedencia de vehículos y factores socioeconómicos del lugar de origen. Combinar datos heterogéneos exige procesos de ingeniería y preprocesamiento de datos.

La normalización es importante: se comparan técnicas como min-max, Z-score, ℓ^2 y MAD, encontrando que la elección influye mucho en los resultados del *clustering*. Por ejemplo, min-max funcionó bien para segmentar individuos y analizar comportamiento de visita, además de detectar visitantes que actúan como residentes. Por otro lado, la normalización ℓ^2 podría ser útil en situaciones específicas que requieran una distinción de la región de origen.

Se aplican técnicas de reducción dimensional para reducir la complejidad mientras se conservan las variables más relevantes. Se evalúan varios algoritmos de *clustering* (*K*-Means, Agglomerative *clustering*, DBSCAN, Gaussian Mixtures) y se eligió Gaussian Mixtures basándonos en un análisis previo de la distribución de los datos, y en los resultados. Usamos las métricas Bayesian Information Criterion (BIC) y Akaike Information Criterion (AIC) con el método del codo para definir la cantidad óptima de grupos. El análisis identifica patrones como turistas de estancia corta y de larga duración y residentes, y muestra la relevancia de variables como frecuencia de visita, número de noches o distancia recorrida.

Predicting Overnights in Smart Villages: The Importance of Context Information

Esta publicación amplía el trabajo previo al pasar de un enfoque no supervisado a uno supervisado en un problema de clasificación. Se utiliza parcialmente el dataset de la publicación anterior, añadiendo algunas variables nuevas relativas al momento del día y lugar de entrada del vehículo a la zona. El conjunto de datos cubre 17 meses, extendiendo la duración anterior. Las 35 variables proceden, a su vez, de 5 fuentes de datos diferentes: uno base con información de las LPR, uno de métricas de visita y tres de contexto (festivos, socioeconómicos y condiciones de entrada del vehículo a la zona).

Se emplean varios modelos de ML (árboles de decisión, regresión logística, máquinas de vectores de soporte, *gradient boosting*, y *transformers* para datos tabulares) con optimización de hiperparámetros y métodos de *ensemble* (*stacking*, *bagging*, *voting*). Además, se introduce una fase de estudios de ablación en la que se eliminan los conjuntos de datos menos relevantes, reduciendo el tiempo de procesamiento un 22.2% y la complejidad del modelo en un 80%, con solo una ligera disminución en la eficacia predictiva. El caso de estudio pretende predecir la duración de la estancia en zonas rurales turísticas, medida en número de pernoctaciones. En los resultados finales, obtenemos que los factores socioeconómicos (ingresos por origen) y aspectos relacionados con el punto de entrada del vehículo a la zona (cámara de detección y momento del día) mejoran las predicciones de manera notable.

SASD: Self-Attention for Small Datasets – A Case Study in Smart Villages

La tercera publicación aborda la limitación de trabajar con pocos datos en problemas de aprendizaje supervisado con datos tabulares. Se presenta una nueva arquitectura de *transformers*, *Self-Attention for Small Datasets* (SASD), que es una red neuronal que

usa capas de *self-attention* para valorar la importancia de las variables y manejar la escasez de datos y el ruido de los mismos.

La arquitectura combina capas lineales, *batch normalization*, *dropout* y mecanismos de *self-attention* al inicio y al final de la red, favoreciendo la captura de relaciones de largo alcance y acelerando la convergencia. Se probaron variantes como *multi-head attention*, menos útiles en datos tabulares pequeños, así como otras distribuciones de las capas de atención.

Se comparó SASD con modelos clásicos (*random forest*, *K*-NN, *gradient boosting*), incluyendo otras NN y *transformers* (redes neuronales recurrentes, TabNet, TabTransformer), usando las métricas de *precisión*, *recall*, F1-*score* y tiempo de entrenamiento. SASD mejoró hasta en un 3% el F1-*score* al resto de modelos evaluados. Se hicieron estudios de ablación sobre las capas de *self-attention* variando tanto el número de capas como su posición en la arquitectura. También se hicieron estudios de ablación sobre el resto de capas propuestas, confirmando que la activación ReLU (rectified linear unit) y las capas dropout mejoran los resultados del modelo.

SASD se usó para predecir la probabilidad de que un turista vuelva en los siguientes 12 meses. Este estudio muestra el potencial de integrar mecanismos de *self-attention* en redes neuronales para casos con datos limitados, y la importancia de posicionarlos en el lugar correcto de la arquitectura. Además de utilizar otras capas para acelerar la convergencia, añadir no linealidad al aprendizaje y evitar el sobreaprendizaje, abordando directamente los problemas de entrenamiento con datos limitados.

CONCLUSIONES

Esta tesis presenta una metodología para diseñar un pipeline de ML que permite recopilar, analizar y evaluar el comportamiento de vehículos en entornos rurales inteligentes. El planteamiento cubre desde la adquisición de datos mediante cámaras LPR y fuentes contextuales hasta la creación de diversos modelos de ML basados en la información fusionada, cumpliéndose todos los objetivos propuestos en esta tesis:

- El primer objetivo se cumplió trasladando el sistema de adquisición de datos propio de ciudades inteligentes a la realidad de zonas rurales. Para ello, se construyó una infraestructura IoT con cuatro cámaras LPR, sometidas a procesos de validación y control para asegurar la fiabilidad de los datos. Tras la limpieza, la información se almacenó para el analisis posterior.
- El segundo objetivo se cumplió mejorando la calidad de la información al fusionar fuentes de datos heterogéneas que complementaran a las cámaras LPR. Se incluyeron días festivos nacionales, procedencia de vehículos y factores socioeconómicos. Estos conjuntos de datos no han sido utilizados en la literatura estudiada, dónde normalmente se utilizan solo los datos de las LPRs. Esta integración de datos enriqueció el análisis, demostrando que la fusión de datos supera los resultados de usar solo LPRs.

- El tercer objetivo se cumplió al analizar distintos métodos de normalización y su efecto en el desempeño de los modelos. Se aplicaron estas técnicas, observando que min-max ofreció una segmentación muy detallada y facilitó la detección de conductas atípicas, mientras que ℓ^2 resultó útil para distinguir el lugar de procedencia en ciertos casos.
- El cuarto objetivo se cumplió al identificar y analizar las variables y conjuntos de datos más influyentes en cada estudio. Para ello, se evaluó la explicabilidad en modelos no supervisados mediante análisis de clusters y, además, se realizaron estudios de ablación en cada conjunto de datos dentro de distintos modelos de clasificación. Esto permitió descartar variables y/o fuentes de datos que suponen altos costos de recolección sin aportar demasiado valor analítico.
- El quinto objetivo se cumplió diseñando y desarrollando nuevas arquitecturas de DL para mejorar las predicciones con datos limitados. Se creó una arquitectura de *transformers*, que mostró un rendimiento superior a otros algoritmos de clasificación, incluidos otros *tranformers*, al aplicarse al conjunto de datos limitados. Este resultado destaca el potencial de las capas de *self-attention*, y su correcto posicionamiento dentro de la arquitectura, en escenarios con datos limitados.
- El último objetivo se cumplió al validar los modelos predictivos en situaciones reales de turismo rural. Todos los modelos se entrenaron y probaron con datos obtenidos de sensores desplegados en un entorno rural funcional situado en la comarca del Barranco de Poqueira, Granada, España, y que abarca los tres municipios de Pampaneira, Bubión y Capileira. También se creó un modelo de *clustering* para agrupar a los vehículos que frecuentan la zona según su comportamiento, un modelo de clasificación para predicción del número de noches que un turista pasará en la zona y otro para predecir si un turista volverá a la zona en los siguientes 12 meses.

Este trabajo plantea un enfoque novedoso al aplicar *transformers* en datos tabulares con pocos datos. Se demuestra que las capas de *self-attention* y su posición en la arquitectura mejoran los resultados al trabajar con información limitada, ayudando a científicos y analístas de datos a superar las dificultades de recolección. Este trabajo también orienta en la elección de técnicas de preprocesamiento según la distribución de los datos. De igual modo, beneficia a administradores de zonas rurales, quienes pueden diseñar estrategias para retener turistas y fomentar las pernoctaciones de los mismos, así como proponer incentivos para residentes no registrados.

En el futuro, tenemos previsto ampliar nuestro trabajo a través de varias vías de investigación: refinaremos las arquitecturas de NN propuestas y validaremos el modelo SASD en conjuntos de datos de referencia públicos más allá del ámbito turístico, comparándolo con algoritmos clásicos y otros de última generación; desarrollaremos herramientas avanzadas de visualización que combinen algoritmos de *clustering* y lógica difusa para desvelar relaciones complejas entre datos e integraremos métodos de explicabilidad como SHAP (*Shapley Additive Explanations*) y LIME (*Local*

Interpretable Model-agnostic Explanations) para interpretar las decisiones del modelo; ampliaremos nuestra infraestructura IoT con sensores de conteo de personas, meteorológicos, de calidad del aire y de nivel de residuos, y diseñaremos estrategias de fusión de estos flujos heterogéneos en un sistema unificado; e implementaremos aprendizaje federado para entrenar modelos de forma distribuida y preservar la privacidad, colaborando con expertos en turismo y responsables políticos de distintas regiones, aprovechando su retroalimentación para perfeccionar nuestras técnicas y documentando casos de estudio que muestren el despliegue y la adaptación del sistema en escenarios reales de aldeas inteligentes.

CONTENTS

LIST OF FIGURES

ACRONYMS

AIC	**A**kaike **I**nformation **C**riterion
ANN	**A**rtificial **N**eural **N**etwork
ANPR	**A**utomatic **N**umber-**P**late **R**ecognition
AP	**A**ffinity **P**ropagation
API	**A**pplication **P**rogramming **I**nterface
ASC	**A**ttributed **S**pectral **C**lustering
AUC	**A**rea **U**nder the **C**urve
BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
BIC	**B**ayesian **I**nformation **C**riterion
CH	**C**alinski-**H**arabasz
CNN	**C**onvolutional **N**eural **N**etwork
DBI	**D**avies-**B**ouldin **I**ndex
DBSCAN	**D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise
DGT	**D**irección **G**eneral de **T**ráfico
DL	**D**eep **L**earning
DTW	**D**ynamic **T**ime **W**arping
EM	**E**xpectation **M**aximization
EU	**E**uropean **U**nion
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
GAN	**G**enerative **A**dversarial **N**etwork
GBM	**G**radient **B**oosting **M**achine
GDP	**G**ross **D**omestic **P**roduct
GPS	**G**lobal **P**ositioning **S**ystem
GPT	**G**enerative **P**re-trained **T**ransformer
HDBSCAN	**H**ierarchical **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise
IC	**I**nformation **C**riterion
INE	**I**nstituto **N**acional de **E**stadística
IO	**I**ndoor-**O**utdoor
IoT	**I**nternet **O**f **T**hings

IP	**I**nternet **P**rotocol
IQR	**I**nter**Q**uartile **R**ange
K-NN	**K**-**N**earest **N**eighbors
LASSO	**L**east **A**bsolute **S**hrinkage and **S**election **O**perator
LDA[1]	**L**atent **D**irichlet **A**llocation
LDA[2]	**L**inear **D**iscriminant **A**nalysis
LIME	**L**ocal **I**nterpretable **M**odel-agnostic **E**xplanations
LOF	**L**ocal **O**utlier **F**actor
LPR	**L**icense **P**late **R**ecognition
LSTM	**L**ong **S**hort- **T**erm **M**emory
MAD	**M**edian **A**bsolute **D**eviation
ML	**M**achine **L**earning
NLP	**N**atural **L**anguage **P**rocessing
NN	**N**eural **N**etwork
OPTICS	**O**rdering **P**oints **T**o **I**dentify the **C**lustering **S**tructure
OvR	**O**ne-vs-**R**est
OvO	**O**ne-vs-**O**ne
PCA	**P**rincipal **C**omponent **A**nalysis
RNN	**R**ecurrent **N**eural **N**etwork
ROC	**R**eceiver **O**perating **C**haracteristic
SASD	**S**elf **A**ttention for **S**mall **D**atasets
SC	**S**ilhouette **C**oefficient
SHAP	**Sh**apley **A**dditive **E**xplanations
SSB	**S**um of **S**quared **B**etween
SSW	**S**um of **S**quared **W**ithin
STING	**S**tatistical **IN**formation **G**rid
SVM	**S**upport **V**ector **M**achine
TN	**T**rue **N**egative
TP	**T**rue **P**ositive
t-SNE	t-**S**tochastic **N**eighbor **E**mbedding
VAE	**V**ariational **A**uto**E**ncoder
ViT	**Vi**sion **T**ransformer

Part I

PHD DISSERTATION

It presents the introduction, theoretical fundamentals, objectives, methodology, main results, conclusions and future lines of research of the doctoral thesis.

1

INTRODUCTION

Machine learning (ML) is currently applied to solve problems in various fields of knowledge, such as healthcare [33], transportation [34], education [35] and industry [36]. ML comprises a series of continuously evolving algorithms that analyze features of data points to identify patterns and discover inferences [37]. One of the most used taxonomies of ML comprises three classes: supervised, unsupervised learning and reinforcement learning [2]. Supervised learning involves training models on labeled datasets, where each data point includes both input features and corresponding output labels, enabling tasks such as classification and regression. Particularly, classification is used to categorize data into predefined classes. For example, predicting whether a tourist will revisit a particular city based on past behavior [3]. Unsupervised learning deals with unlabeled data, where the outcomes are unknown, focusing on discovering hidden patterns or intrinsic distributions within the data. A popular task in unsupervised learning is clustering, which groups data points that follow a similar behavior or pattern. For example, by analyzing toll station payment data, we could identify common behaviors among the different types of vehicles [4]. Finally, reinforcement learning, where an agent learns to make optimal decisions through interaction with an environment and reward feedback. For example, training a vehicle to drive autonomously while avoiding obstacles [5]. Although we have followed this taxonomy, there are alternative taxonomies that are also commonly used today. For example, deep learning (DL) and shallow learning (SL) [6].

In the last decade, DL has significantly advanced SL by defining a set of algorithms based on multilayer architectures (SL only use one layer algorithms) to identify patterns through the analysis of complex relationships within data. These algorithms are commonly referred to as artificial neural networks (ANNs or NNs). NNs have exhibited strong performance in numerous fields, including education [7], healthcare [8, 38], and tourism [39, 9], often surpassing classical methods [40, 41]. For example, an ANN that recommends tourist attractions based on the behavior of tourists, outperforms other traditional classifiers such as K-NN or random forests [42]. On the downside NN need a large amount of training data. One recent advancement in the field of NN is transformer-based architectures, which introduce a new concept, attention, to improve model performance [12]. Transformers use attention mechanisms

that dynamically assign varying levels of importance to different data points based on their relationships within the dataset. Although initially developed for natural language processing (NLP), they excel as well in tasks where the order of appearance of data points matters, such as time-series forecasting or signal processing [11].

Attention mechanisms enable transformers to effectively capture long-range data dependencies and contextual relationships that traditional NNs, such as recurrent neural networks (RNNs) struggle to address [13]. For instance, transformer-based models with attention mechanisms outperform classical RNN-based models such as long short-term memory (LSTM) in traffic flow prediction, demonstrating superior accuracy and reliability [14]. Transformers have also been applied to image processing tasks [10, 43], taking advantage of their ability to analyze spatial relationships and patterns. Recently, their use has expanded to tabular data problems, which include structured rows and columns [15, 16]. Attention mechanisms could also help NNs acquire knowledge from small datasets [44], for example, in the field of image learning [10, 17]. Although attention mechanisms for NN have also been developed for tabular data, such as numerical or alphanumeric data from sensors or questionnaires [15, 16], these solutions are not fully adapted to small datasets [18].

To apply ML and DL models, researchers develop ML pipelines, which includes a sequence of processes that starts with data collection and ends with the model's deployment [19]. This process (see Figure 1) consists of three general steps:

i. Data acquisition and preparation.
   - Data collection, where data is collected, formatted, and stored.
   - Data validation, which ensures the quality and consistency of the incoming data by conducting an exploratory analysis of the data distribution, detecting anomalies or feature selection.
   - Data preprocessing, which transforms the data into a format suitable for training, using techniques such as normalization or dimensionality reduction.

ii. Model training and optimization.
   - Model training, where algorithms learn from the data.
   - Model tuning, where hyperparameters are adjusted to improve performance.
   - Model evaluation and visualization, where the trained model is assessed using metrics such as precision and accuracy. At this step, it is also possible to apply methods of visualization of the results to further refine the model.

iii. Deployment of the evaluated model for use.
   - The final step, involves deploying the evaluated model, which will be continuously refined and updated with new data, restarting the process loop.

In the data preprocessing sub-step, two important tasks are feature selection and normalization. Feature selection assess the impact and weight of each feature on the

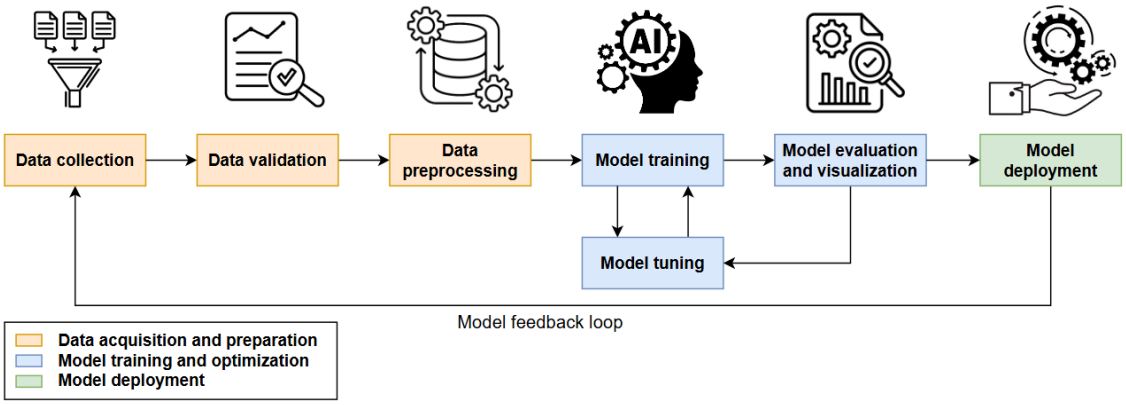


Figure 1: ML pipeline general flow.

learning process and identifies the most influential in the model [20, 21, 22]. This step, along with visualization, can help determine the explainability of the model. Although some research applies feature selection to choose the most influential features in the model, none of them considers the importance of looking at the dataset level rather than just single features. In today's environment, where efficient and sustainable ML practices are increasingly valued [25], conducting ablation studies on each independent context dataset can indicate whether investing time and resources in data collection for a specific source is worthwhile [26, 27]. The normalization stage has received little attention. Researchers often apply a single method without assessing its suitability. However, this step is important, as each normalisation technique transforms the scale and distribution of the data, affecting the performance of the selected ML algorithm and altering its results. Different outputs for the same algorithm can lead to inconsistencies when interpreting the results and analysing the explainability of the models obtained [28].

Hence, we have spotted some challenges that this thesis aims to address: (1) within the DL field, there are no transformers adapted to small tabular datasets, a limitation that is particularly significant in practical applications where data collection is restricted, such as works that rely on questionnaires with a limited number of respondents. Within the data acquisition and preparation step (i) of an ML pipeline that integrates contextual datasets: (2) the feature selection step poses a challenge, as most of the works in the literature are conducted on individual features rather than on the complete datasets from each source, which might otherwise help decide whether to invest time and resources in a specific data source; and (3) normalization task has not received sufficient attention, even though it is critical to both the results and the explainability of the model.

Our approach consists of building an ML pipeline that integrates data from various sources and fuses them. During data validation step, we perform an ablation study at the dataset level to examine the influence of each source on the model, thus

addressing the challenge (2). In data preprocessing step, we employ different normalization methods to explore their impact on the model results, which addresses the challenge (3). Finally, we propose a transformer-based architecture with different layers including: self-attention, dropout, batch-normalization and ReLU layers. We propose a configuration specifically tailored to limited data to address the challenge (1). In addition, we test other architectures and algorithms for comparison and draw conclusions during the evaluation and visualization step.

To develop our proposals, we have used a smart village use case. We selected this use case because of the importance of the smart cities/villages concepts and the relatively small datasets associated with smart villages compared to smart cities. Smart cities have gained momentum due to the increasing integration of Internet of Things (IoT) technologies in urban environments. In 2023, there were approximately 15.9 billion IoT devices worldwide. Statista projected this number to grow to 20.1 billion this year and reach 39.6 billion by 2033[1]. These devices form a vast, interconnected network that generates extensive data across various social domains. For example, IoT platforms monitor aspects of urban life [45], defining what is called smart cities. Smart cities use technologies and data-driven solutions to enhance urban services, infrastructure, and residents' quality of life [46]. By integrating IoT devices, cities can monitor and manage processes in real time, improving areas such as transportation, waste management, and public safety [47, 48, 49]. For example, license plate recognition (LPR) cameras allow detailed analysis of vehicle behavior and traffic flow [50, 51]. This data can be merged with information from events, parking details, and weather patterns [52, 21] to improve the results of the analysis. Researchers feed the fused data into ML pipelines. Figure 2 shows devices such as cameras, and other sensors collecting data. Researchers apply ML algorithms to extract patterns and useful information from data. Most studies focus on mobility patterns to reduce traffic congestion [53, 54] and cluster vehicles for urban management [55], while others examine air pollution, climate change, and pedestrian routes [56, 57]. ML models have also been developed to predict traffic or visitor flows in the cities [58, 59].

Over the past decade, urban planners applied the smart city concept to rural areas [60]. Rural development programs in the 2010s introduced smart villages, a concept that gained further momentum in 2017 with "The EU Action for Smart Villages" [29, 30]. Smart villages are a convenient use case for our proposal because they have distinct characteristics compared to urban environments. While smart cities have been the subject of numerous studies, smart villages are an emerging concept that has attracted attention but remains underexplored. The literature highlights that solutions designed for large cities may not be directly applicable to small villages [60, 61]. For example, villages with narrower or predominantly pedestrian streets cannot directly adopt approaches that rely on multiple lanes of traffic. Another significant difference between smart cities and smart villages is their population size and infrastructure. Unlike cities, where data are more easily collected through both auto-

1 https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

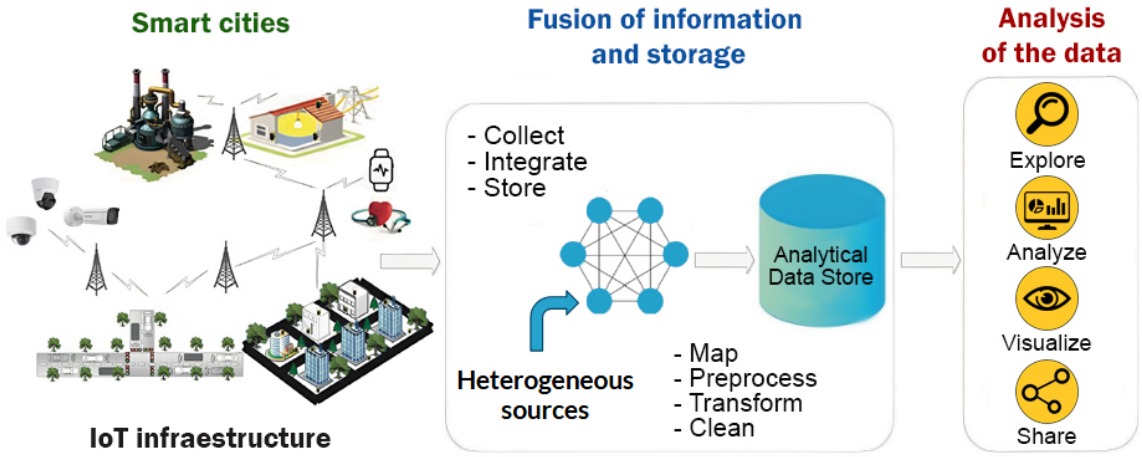


Figure 2: Data flow in the IoT within smart cities. Adapted from [31].

mated systems and manual records, villages typically produce much smaller datasets. This limitation makes it challenging to apply DL models effectively, as they require vast amounts of data to achieve reliable performance.

In particular, we apply our proposal to a rural case study on vehicle mobility, confronting challenges in ML pipelines and smart villages. We use IoT devices to collect data on vehicle behavior. Particularly, we installed four LPR cameras in a rural area to build an IoT infrastructure that collects data. We also collect contextual information such as holiday calendars, vehicle provenance, visitors' socio-economic factors, and questionnaires responses on demographic and behavioral aspects related to vehicle usage and visitor patterns. Some studies have used LPR cameras for mobility clustering [62, 63, 64, 65], but none combine LPR data with this contextual information such as visitor's provenance, or socio-economic details. First, we use clustering algorithms to identify common clusters and patterns in collected data. To construct the ML pipeline, we selected algorithms that fit the data distribution, performed exploratory analysis during data validation step. Next, we analyze the resulting clusters in model visualization step. We focus on data preprocessing step, testing several normalization methods, and examining their effects on the final cluster visualization, tackling the challenge (3). We obtain different results depending on the normalization methods used. We also address model explainability, analyzing the features that most influence them. This process enable us to develop a model that distinguishes residents from visitors based on their behavior. We use this information to guide subsequent classification tasks. We concentrate on feature selection by performing ablation studies to find the datasets that most improve the model, reduce features, shorten processing time, and boost evaluation metrics tackling the challenge (2). We tested this approach with a model that predicts how many nights a visitor will spend in the area. We examine the influence of different datasets in improving the classification model. Once we study the influence of normalization and feature selection in data acquisition and preparation step, we shift our focus to improving the model.

In this case, we address the challenge (1) by proposing an attention-based NN architecture that handles small datasets. To apply this architecture, we use data visitor questionnaires capturing visitation intentions, and merge them with the rest of collected information. With this, we get a model to predict which tourists will return in the short term.

Our results introduce a new research direction by applying transformers to tabular datasets with limited data. It improves predictive models when researchers can not collect enough data to train a model. Our work guides researchers and developers to select preprocessing techniques based on data type, distribution, and sources when designing ML pipelines. Our results also benefit stakeholders in smart villages. The normalization method reveals distinct movement patterns, such as visitors acting as residents or long-stay international visitors. Policymakers can use these insights to design strategies that retain specific tourists, considering factors such as income and origin, and to promote overnight stays. Moreover, the data guide policies that engage non-registered residents through incentives like tax breaks or social programs. In Spain, stakeholders apply this information for tasks such as licensing pharmacies, investing in public health, and scheduling security forces during seasonal fluctuations. We also conducted an ablation study at the dataset level to analyze data from various sources and expert opinions. This process identifies the datasets that exert the greatest influence on predictions when estimating how many overnights a visitor stays in the area. The analysis reveals that the socio-economic status of a visitor, and the point of entry to the rural area, influence the outcome more than the other datasets. Our work aids scientists who develop predictive models in tourism. By identifying the key databases, we help them allocate resources efficiently when budgets and time are limited. Finally, we build a transformer model for limited datasets that tracks tourism repeaters. Policymakers can use this results to design local tourism strategies for urban planning or transport.

This thesis consists of two main parts. The Part I focuses on the PhD Dissertation: Chapter 2 presents the fundamentals that support this thesis, Chapter 3 outlines the objectives, and Chapter 4 details the methodology. Chapter 5 summarizes the key findings from the publications, while Chapters 6 and 7 discuss the conclusions and potential future work, respectively. The Part II of this document presents the three publications that form the core of this thesis:

- *Clustering Pipeline for Vehicle Behaviour in smart villages.*
- *Predicting Overnights in smart villages: The Importance of Context Information.*
- *SASD: Self-Attention for Small Datasets - A Case Study in smart villages.*

2

FUNDAMENTALS

This Section presents the fundamentals on which the methodology and experimentation proposed for this doctoral thesis is based. Section 2.1 reviews the basics of ML, distinguishing in supervised, unsupervised, and reinforcement learning taxonomy. Section 2.3 describes popular clustering algorithms used to detect patterns in data. Next, Section 2.2 explores supervised learning, examining popular classification algorithms. Section 2.4 discusses DL, emphasizing transformers architectures with multihead and self-attention mechanisms. Finally, Section 2.5 presents the theoretical methods of the ML pipeline steps.

2.1 ML TAXONOMY

ML is responsible for extracting meaningful patterns from a dataset with the objective of inferring the underlying statistical distribution [37]. Traditionally, researchers used computer algorithms to learn a model from a previously selected dataset of interest [2, 66]. Nowadays, they develop algorithms that work independently of any specific domain [6]. Once the model is trained, they can use it to make predictions without programming it for a specific task. Within the classification of ML algorithms, there are several taxonomies. The most popular taxonomy considers three types of ML learning classes: supervised learning, unsupervised learning, and reinforcement learning [1, 2].

- **Supervised Learning:** is an approach that uses labeled examples (with known correct answers) to learn a function that can predict these labels [1]. Specifically, a supervised learning algorithm receives input examples and labels that identify each example. Representative algorithms include linear regression [67], Bayesian probabilistic methods [68], decision trees [69], support vector machines [70], or supervised NN [71]. It is commonly applied in classification (e.g., spam detection [72] or image recognition [73]) and regression tasks (such as predicting numerical values in finance or medicine [74]).
- **Unsupervised Learning:** is an approach that uses unlabeled data, seeking to discover hidden structures or patterns without known information in advance [1]. Common unsupervised learning algorithms include clustering methods,

which group examples based on their feature values, inter-correlations, and intrinsic structure. Popular examples are *K*-Means [75], Affinity Propagation [76], DBSCAN [77], or Gaussian Mixtures [78]. Typical applications include data segmentation (e.g., customer segmentation in marketing [79]), anomaly detection (e.g., identifying unusual patterns in network traffic [80] or detecting fraudulent transactions [81]), and exploring datasets to find meaningful relationships without external guidance.

- **Reinforcement Learning:** is an approach in which an agent learns to make decisions by interacting with an environment [82]. The agent performs actions and receives rewards or penalties based on the results, adjusting its behavior to maximize the long-term cumulative reward. Popular algorithms of this type are Q-learning or deep reinforcement learning methods (such as Deep Q-Network [83] or Actor-Critic methods [84]). It is applied to learn optimal control strategies for sequential tasks (e.g., controlling robots for precise manipulation [85] or managing adaptive behaviors in autonomous vehicles [5]).

In this thesis we will be focus on the unsupervised and supervised learning. Although we have followed this taxonomy, there are alternative taxonomies that are also commonly used today. For example, generative and discriminative learning, which focus on whether a model learns the data distribution or the decision boundary [86]. Generative algorithms adopt a probabilistic approach and can generate new samples from the learned distribution, while discriminative algorithms are optimized for classification by focusing on the separation between categories. Another popular taxonomy distinguishes between deep and shallow learning. DL approaches use multiple layers to automatically learn complex patterns and representations from raw data, while shallow learning algorithms use a single layer and typically rely on manual feature engineering to extract relevant information [6].

2.2 POPULAR SUPERVISED ALGORITHMS

The classification algorithms presented below are classified according to one of the most popular supervised learning taxonomies, which divides the algorithms into [37, 1, 87]:

- **Decision Trees [69]:** algorithms in this category separate data into branches based on specific characteristics, which makes it easier to understand how each classification decision is made. Representative algorithms include Decision Trees and Random Forest [88].
- **Support Vector Machines (SVM) [70]:** these methods seek to find the optimal hyperplane that maximizes class separation, making them useful for both classification and regression tasks. A popular examples are the Linear SVM [89], Kernel SVM [90] or Least Squares SVM [91].
- **Nearest Neighbors [92]:** this technique makes predictions based on the proximity of data points. By adjusting the parameter *K*, which determines the number

of nearest neighbours considered, the sensitivity of the algorithm, i.e. the responsiveness of the model to variations in the local structure of the data, can be controlled. A smaller K results in greater sensitivity, capturing local variations, while a larger K produces smoother, less sensitive decision boundaries. The most widely recognised method is the K-Nearest Neighbors (K-NN) [93], but existes others such as Condensed NN [94] or Distance-Weighted K-NN [95].

- **Logistic Regression [96]:** this method models the probability of a categorical outcome by leveraging predictor features, thus serving as a fundamental tool for binary and multiclass classification tasks. Some examples are Logistic Regression [96], Regularized Logistic Regression [97] or Multinomial Logistic Regression [98].
- **Bayesian Probabilistic Methods [68]:** these probabilistic approaches utilize Bayes' theorem to infer the likelihood of outcomes. For instance, Gaussian Naive Bayes [99] operates under the assumption of feature independence for classification. Other examples are: Latent Dirichlet Allocation (LDA[1]) [100], or Bayesian Networks (Belief Networks) [101].
- **Gradient Boosting Machines [102]:** this ensemble technique builds sequential tree-based models that incrementally improve performance by correcting errors made by previous models. Notable algorithms include Light GBM [103], which leverages gradient boosting on decision trees; XGBoost [104], which uses advanced optimization techniques and sparsity-aware learning; and CatBoost [105], which effectively handles categorical features through ordered boosting and symmetric trees.
- **Neural Networks [71]:** this family of algorithms models complex nonlinear relationships through interconnected layers of perceptrons. Key examples include the MLP Classifier [106] for general-purpose classification, Recurrent Neural Networks (RNN) [107] for sequential data processing, and Long Short-Term Memory networks (LSTM) [108] which address the vanishing gradient issue by maintaining long-term dependencies.

2.3 POPULAR UNSUPERVISED ALGORITHMS

The clustering algorithms presented below are classified according to one of the most popular unsupervised learning taxonomies [109, 110, 1].

- **Partitional Clustering:** this clustering technique decomposes a dataset into distinct clusters through an iterative process of distance calculations between individuals, and typically uses centroids. Examples of algorithms that utilize this technique include K-Means [75] and MiniBatch$K$Means, which is a scalable version of K-Means that updates clusters using small random batches until convergence is achieved [111]. Another algorithm that falls into this category is ISODATA [112], which employs iterative self-organizing data analysis.
- **Hierarchical Clustering:** this clustering method constructs clusters in either an agglomerative or divisive manner by adding or removing individuals, respec-

tively [1]. Some examples include SLINK [113] implements the single linkage method by constructing the dendrogram without computing all pairwise distances, while CLINK [114] optimizes complete linkage clustering by minimizing the maximum distance between elements of different clusters. Other popular example is BIRCH [115], an algorithm that uses an unbalanced height tree to dynamically split data points.

- **Density-Based Clustering:** this technique identifies dense regions of objects in the data space separated by low-density regions. It is known to handle noise well and adapt to arbitrary shapes in the data. The algorithm most common in this category is DBSCAN [77], along with improved versions of it, such as OPTICS [116] and HDBSCAN [117], which compute a density function for each cluster found. Other examples include Mean-shift [118], which creates clusters based on regions of maximum density attraction and can be considered a version of K-Means using density functions, making it adaptable to arbitrary shapes of clusters.
- **Distribution-Based Clustering:** this technique creates clusters based on the probability that each individual belongs to the same distribution, the Gaussian distribution is the most widely used distribution based on the expectation maximization algorithm [119]. These algorithms result in Gaussian Mixture models [78], which are also classification algorithms. In some cases, they are a generalization of *K*-Means, with each individual having a probability of belonging to each cluster.
- **Grid-Based Clustering:** this clustering approach involves dividing the space into a finite number of cells, followed by defining clustering operations within the quantized space. Some popular algorithms that utilize this method include STING [120], WaveCluster [121], and CLIQUE [122].
- **Message-Passing Clustering:** this category of clustering creates clusters by exchanging messages between different data points until convergence. An example of this approach is the Affinity Propagation (AP) algorithm [76], which has been further improved by proposals such as IWC-KAP [123] and ScaleAP [124].
- **Spectral Clustering:** this method uses the spectral radius of a similarity matrix of the data in a multidimensional problem. Dimensionality reduction techniques, such as Principal Component Analysis (PCA), are used to obtain a linearly separable problem. There are different versions of Spectral Clustering algorithms [125], depending on how the eigenvectors are selected from the Laplacian of the similarity matrix [126]. Newer versions, such as Attributed Spectral Clustering (ASC), improve the degree of affinity between nodes in the same density region [127]. Another algorithm is Self-Tuning Spectral Clustering [128], which adjusts the parameters of the similarity matrix according to the local density of the data, facilitating clustering in problems with varying scales.

DL has transformed the analysis of complex data through DL architectures capable of capturing hierarchical representations [6]. Technically, a NN consists of several layers of interconnected neurons, where each neuron applies a nonlinear activation function to a weighted sum of its inputs [129]. Transformers [12] are NN architectures that utilize attention mechanisms to enable the modeling of long-range dependencies and parallel processing of input data (see Figure 3).

Structurally, they follow an encoder-decoder structure [130, 131], where both components are composed of multiple identical layers. An encoder-decoder network generates features length yet contextually appropriate output sequences to correspond to a given input sequence [132]. Each encoder layer consists of a multi-head self-attention mechanism followed by a position-wise feed-forward network, with residual connections and layer normalization applied after each operation [12]. The decoder stack mirrors the encoder but includes an additional attention layer that allows it to attend to the encoder output while maintaining autoregressive generation through masked self-attention. Token embeddings and positional encodings provide sequence order information [133], ensuring effective representation learning.

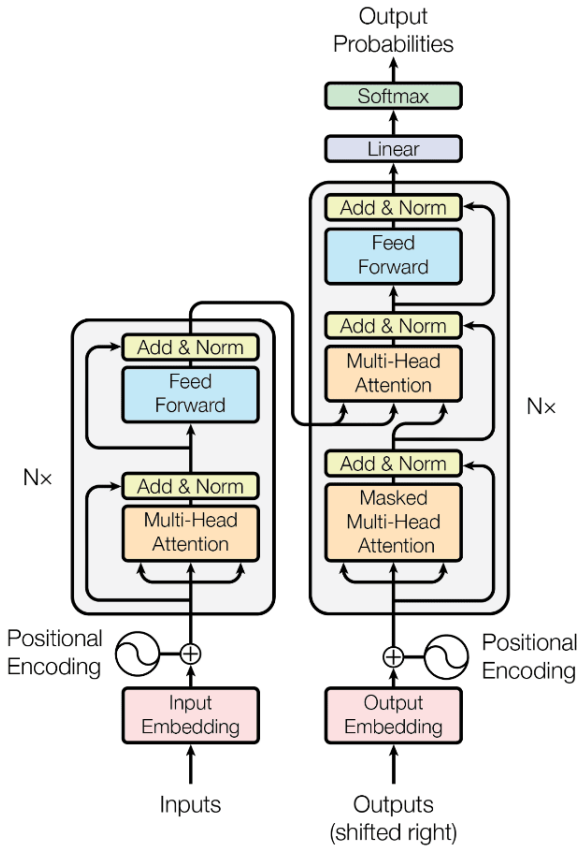


Figure 3: Transformer model architecture. Source [12].

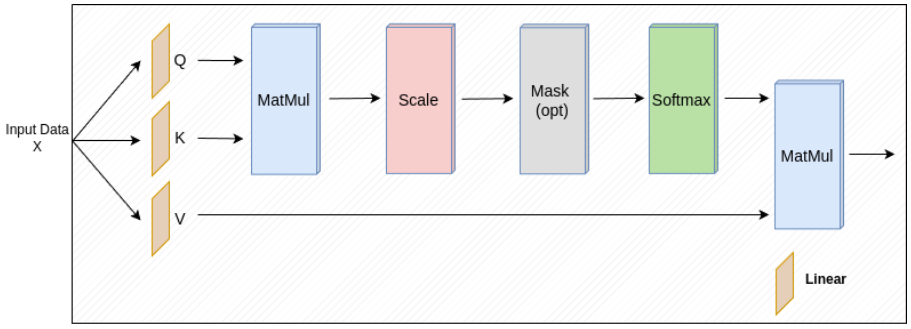


Figure 4: Self-attention layer. Adapted from [12].

Next, we examine self-attention and multihead-attention mechanisms, which are an important basis in transformers architectures.

2.4.1 *Self-Attention Mechanism*

Self-attention, initially used for language translation, is an attention mechanism where the input sequence itself serves as the queries, keys, and values. This allows the model to weigh the importance of the words in a sentence, capturing long-term dependencies and enhancing the contextual representation of each word [134]. For an input sequence $X = [x_1, \ldots, x_n]$, where x_i is a vector of features for the i-th word, the self-attention mechanism (see Figure 4) is calculated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

This Equation 1, is known as "Scaled Dot-Product Attention" and is characterized by:

- Linearly transform the input X to obtain the matrices of queries (Q), keys (K), and values (V), respectively:

  $$Q = XW_Q, \quad K = XW_K, \quad V = XW_V,$$

  where W_Q, W_K, and W_V are weight matrices that facilitate the transformation of the input data X into different representations for the purpose of computing attention scores.
- Compute the dot product QK^T to obtain the attention scores, which are then scaled by the inverse square root of the dimension of the keys (d_k), i.e., $1/\sqrt{d_k}$, to prevent the scores from becoming too large.
- Finally, apply the softmax function to each row of the scaled QK^T matrix, normalizing the weights to sum 1, and use these to weight the values (V) through matrix multiplication.

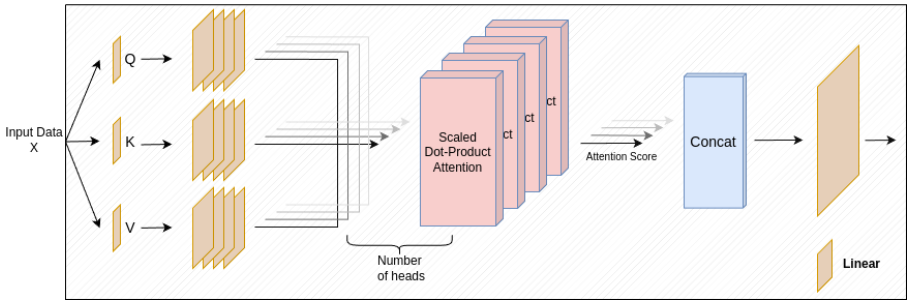


Figure 5: Multihead-attention layer. Adapted from [12].

2.4.2 *Multihead-Attention Mechanism*

The multihead-attention mechanism [12] shortens the processing time of the self-attention mechanism by processing multiple attention tasks simultaneously. Initially, the Q, K, and V vectors are projected into h separate sets, named Q_i, K_i, and V_i for each $i = 1, \ldots, h$. The Equation 1 formula is applied to each projection set. Following this, the results are merged by first concatenating them and then applying a linear projection (see Figure 5). The Equation 2 defines the overall multi-head attention mechanism. In multihead-attention, each head focuses on different parts of the input sequence. Hence, the model captures a wider range of dependencies than a single instance of self-attention.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{2}$$

where each head_i is defined as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V).$$

Here, W_i^Q, W_i^K, and W_i^V represent the projection matrices for transforming Q, K, and V in the i-th head, and W^O is the matrix used for the final projection.

The literature highlights several transformer-based models for different types of data. For image data, the Vision Transformer (ViT) [135] has demonstrated impressive performance in computer vision tasks by treating images as sequences of patches and applying self-attention mechanisms to capture global context. For text data, models like Bidirectional Encoder Representations from Transformers (BERT) [136] and the Generative Pre-trained Transformer (GPT) series [137] have revolutionized natural language processing by leveraging self-attention to model contextual relationships in language. For tabular data, TabNet [15] integrates transformers in neural networks, including an attention mechanism for feature selection, making it particularly efficient in tasks involving high-dimensional structured data and time series. TabTransformer [16] leverages transformer architectures to model interactions between

categorical and numerical features, using self-attention mechanisms to capture complex dependencies in tabular data.

2.5 OVERALL ML PIPELINE

The construction of an ML model follows an ML pipeline (see Figure 1), which starts with the preparation of the data and ends with the deployment of the model [19]. Below are the theoretical fundamentals of each step of the ML pipeline construction process.

2.5.1 *Data Collection*

Data collection is the first step in any ML pipeline. During this step, data is processed into a format that subsequent steps can handle. The quality, quantity, and variety of data largely determine the performance and robustness of models [138]. The sources of data vary depending on the application domain (e.g., can range from IoT sensors to questionnaires, administrative records, or public databases). There are various methods for merging multiple data into a unified set. For example, early fusion techniques, where data are integrated directly at the initial stage, and late fusion, which combines the outputs of individual models, have proven to be effective in handling heterogeneous data sources [139]. Some works have also developed hybrid approaches that take advantage of the best of both methods to optimise model performance and robustness [140]. In addition, there are strategies for collecting various types of data (tabular, text and images) that have proven useful in different use cases [139]. When dealing with unstructured data, it is common to use transformations such as text embeddings based on transformer architectures [12] or the application of Convolutional Neural Networks (CNN) for feature extraction in images [141]. These methods allow the efficient integration and processing of different types of data, adapting to the specific needs of each application.

2.5.2 *Data Validation*

Data validation is a step in the ML pipeline process that ensures the quality and consistency of the data collected. This step explores the integrity of the data by detecting anomalies and exploratory analysis of the distribution, identifying changes in the data and studying the statistics of the data sets used. Data validation also produces statistics around the features of the data, and allows selecting which features will go to the data preprocessing step [19].

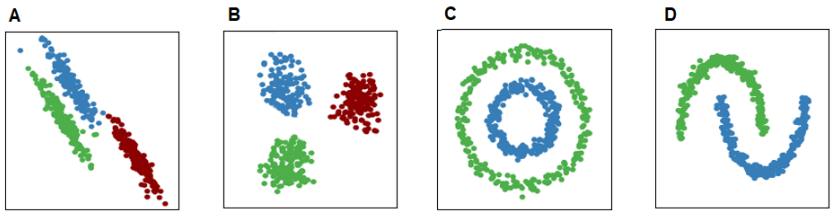


Figure 6: (A) Three 1D flat manifolds (segments); (B) Three 0D flat manifolds (points); (C) Two 1D non-flat manifolds (circles); (D) Two 1D non-flat manifolds (arcs). Adapted using scikit-learn code[1].

Data Distribution

In data analysis, Riemannian geometry, by considering curvature, allows to identify the distribution of the data and to assess whether the data are best represented in a flat or non-flat variety. This approach aids dimensionality reduction and the interpretation of variability [142]. In mathematics, a Riemannian manifold [143] is a geometric object that can be described locally as Euclidean space. Curvature is an intrinsic measure of a manifold, indicating how much the manifold curves at each point. In this context, we say that a manifold is flat if its curvature is zero at all points, that is, if the manifold is locally indistinguishable from a flat Euclidean space. If the curvature is not zero at any point of the manifold, the manifold is non-flat. In data analysis, we refer to flat and non-flat geometry as the measurement of distances between points by Euclidean or non-Euclidean geometric methods, respectively [144]. In flat geometry, the distance is measured following a straight line between two points, while in non-flat geometry, the distance is measured following a curve. We can detect whether our data follow flat or non-flat geometry by representing the data in a scatter plot, where each point represents an individual in the population. Visually we can only represent 3 dimensions, which normally are the most representative features of the cluster, or the firsts principal components of a dimensional reduction algorithm. Figure 6, shows four images of different data distribution. If the figure is circular, rectangular or elliptical (e.g., images A,B), the data follows a flat geometry. However, if the figure has an irregular, twisted or folded shape (e.g., images C,D), the data follows a non-flat geometry [144].

Anomaly Detection

Outliers are defined as data points that significantly deviate from the majority of the dataset due to errors or variations, and they can greatly influence statistical results [145]. Excluding extreme data points before analysis can reduce distortions caused by data anomalies and improve the results by addressing inconsistencies [146]. However, it is important to recognize that such removal might also eliminate valid observations,

1 https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

potentially introducing bias in ML models [147]. Various statistical methods and ML algorithms are employed for this purpose. The most well-known are:

1. **Isolation Forest** [148]: This algorithm works by randomly selecting features and splitting values between the maximum and minimum values, effectively isolating outliers based on the number of splits required. One of its advantages is its efficiency on large datasets, as it does not require the calculation of distances or densities. Its performance is due to the construction of isolation trees that recursively explore random partitions, making anomalies, being less represented, easier to separate from the rest of the data.
2. **Interquartile Range (IQR) Method**: This statistical approach identifies outliers as those data points falling outside the interval $[Q_1 - K \cdot \text{IQR}, Q_3 + K \cdot \text{IQR}]$, where Q_1 and Q_3 represent the first and third quartiles (i.e. the 25th and 75th percentiles), and the interquartile range (IQR) is the number obtained by subtracting Q_1 from Q_3, thereby encompassing the central 50% of the data [146]. Adjusting the multiplier K allows setting the threshold for identifying outliers; a smaller K value results in a narrower range of what is considered normal, thereby classifying more points as outliers, while a larger K value reduce the number of points classified as outliers. Typically, K is defined such as 1.5 because it is roughly equivalent to $\pm 3\sigma$ in a normal distribution, identifying about 1% of the data as outliers [149], where σ represents the standard deviation.
3. **Z-Score Method**: This technique calculates the standard score for each observation (x_i) using the formula $z_i = \frac{x_i - \bar{x}}{\sigma}$, where σ is the standard deviation, and $\bar{x}$ is the mean of the dataset. Typically, a data point is considered an outlier if its absolute z-score (z_i) exceeds 2.5 [150]. This threshold can be adjusted to increase or decrease the percentage of detected outliers, providing flexibility based on the specific requirements of the analysis.
4. **Local Outlier Factor (LOF) [151]**: This algorithm measures the anomaly of a data point by comparing the local density of its neighbors. LOF calculates the ratio of the point's local density to the average local density of its nearest neighbors. A value of LOF significantly greater than 1 indicates that the point is an outlier, as its environment is much less dense compared to its neighbors. This approach is especially useful in scenarios where the data distribution is uneven or when clusters of varying density exist, allowing for more adaptive anomaly detection.

Feature Selection

Feature selection is classified for the most popular taxonomies into three main approaches: filtering, wrapping and embedding [152]. Filter methods individually evaluate the relevance of each feature using statistical metrics (e.g., chi-square, data frequency or information gain). Wrapper methods optimize selection by exploring combinations of features based on model performance [153]. Embedded methods

integrate selection within training, most notably approaches such as Least Absolute Shrinkage and Selection Operator (LASSO) [154] and implicit selection in decision tree models [69]. These techniques have been widely studied to improve the explainability and computational efficiency of models [155].

2.5.3 *Data Preprocessing*

Data preprocessing is a step in the ML pipeline that transforms raw data into a format suitable for training models. This process involves applying statistical techniques, such as normalization and dimensionality reduction, and converting labels into vector representations (e.g., one-hot or multi-hot encoding) [19]. Since preprocessing is performed only once before training rather than at every epoch, it is typically executed as an independent step to ensure efficient model training.

Data Normalization

Normalization adjusts the range of each feature to a common interval (e.g., $[0, 1]$ or $[-1, 1]$), enabling fair comparisons in subsequent pipeline steps by preventing larger-scaled features from dominating the analysis. In general, a feature X is transformed into X', where X' is the scaled version that facilitates equitable comparisons. The choice of the normalization algorithm usually depends on the specific application and the dataset used, as different methods may yield different results and interpretations. For example, in clustering analysis, normalization is useful because many distance measures, such as the Euclidean distance, can be affected by the scale of the features; scaling all features to a common range ensures that no feature, particularly those with larger numerical values, disproportionately influences the similarity calculations between data points. Different normalization methods can yield variations in the results. Some popular methods [156] are:

1. **Min-max normalization [157]:** Uses the minimum (X_{min}) and maximum (X_{max}) of the attribute X domain to scale the values to the range $[0, 1]$. This method preserves the relative distances between points, which is beneficial for distance-based algorithms.
$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$
2. **Z-score standardization [157]:** Transforms X so that the mean (μ) is 0 and the standard deviation (σ) is 1. This approach is particularly useful when the data is assumed to follow a normal distribution and helps to mitigate differences in scales among attributes.
$$X' = \frac{X - \mu}{\sigma}$$
3. **Median Absolute Deviation (MAD) normalization [158]:** Normalizes X such that the median of each attribute becomes 0 and the MAD becomes 1. This

method is robust to outliers since it uses the median instead of the mean as the central tendency measure.

$$X' = \frac{X - \text{median}(X)}{\text{MAD}(X)}$$

4. ℓ^2 **normalization [159]:** Scales X by dividing by the Euclidean norm ($||X||_2 = \sqrt{\sum_{i=1}^{n} X_i^2}$), ensuring that all feature vectors have the same length. This technique is widely used in machine learning and information retrieval, especially in contexts where the direction of the vector is more important than its magnitude.
$$X' = \frac{X}{||X||_2}$$
5. **Decimal scaling normalization [156]:** This method normalizes X by dividing by a power of 10 such that the maximum absolute value of the normalized data is less than 1. It is defined as:
$$X' = \frac{X}{10^j}$$
where j is the smallest integer such that $\max(|X'|) < 1$. Although simple to implement, its effectiveness may be limited in the presence of extreme values.
6. **Logarithmic transformation [160]:** Used for strictly positive data, this method applies to X a logarithmic transformation to compress a wide range of values and reduce skewness in the distribution. The value c is a constant (often $c = 1$) to ensure that the logarithm of zero is not computed. It is expressed as:
$$X' = \log(X + c)$$

Dimensionality Reduction

Dimensionality reduction decreases the number of features, simplifying data analysis and visualization, while improving the efficiency of ML algorithms [161]. Several techniques have been developed for this purpose, including Fisher's Linear Discriminant Analysis (LDA[2]), which maximize class separability by projecting data into a lower-dimensional space where the distances between class means are maximized and the variance within each class is minimized [162], Isometric Mapping, which preserves the distribution of the data in a lower-dimensional space [144], and t-distributed Stochastic Neighbor Embedding (t-SNE), renowned for its ability to preserve neighborhood relationships among data points in datasets that exhibit nonlinear structures [163]. Despite the variety of methods, PCA is the most popular due to its computational efficiency and the clear interpretability of its principal components [164].

PCA method condenses the information provided by multiple features $(X_1, \ldots, X_p)$ from a sample into fewer features, finding a number s of underlying factors that approximately explain the same variance as the original features with $s < p$. Each

of the new features $(Z_1, \ldots, Z_p)$ are called principal components, which are linear combinations of the original features. Each Z_i is defined as:

$$Z_i = \Phi_{1i} X_1 + \Phi_{2i} X_2 + \cdots + \Phi_{pi} X_p$$

Each Φ represents the weight or importance that each feature X_i has in each Z_i and explains the information collected by each of the principal components [165]. It is advisable to apply prior normalization to the data because this method is highly sensitive to features with different scales. Moreover, PCA only works with numerical data, so it is necessary to preprocess any categorical features in the input dataset [166].

2.5.4 *Model training and tuning*

The model training and tuning step is the core of the ML pipeline [19]. In this step, we train the model using the features collected and transformed in previous steps while searching for the best hyperparameters of each ML algorithm. The hyperparameters are the settings and values determined before the learning process that influence the training behavior. Examples of hyperparameters include the learning rate and the number of epochs in NNs, the number of leaves or the depth in random forests, the number of kernels in SVMs, the number of nearest neighbors (K) in K-Nearest Neighbors, the number of clusters (K) in K-Means clustering, and both the number of mixture components and the covariance type (e.g., diagonal, full, tied, spherical) in Gaussian Mixture Models.

Often, it is not feasible to determine in advance a single best algorithm. Instead, multiple algorithms and their respective hyperparameters are evaluated to identify the most optimal combination for the given dataset [2]. While the distribution of the data can provide insights into which algorithms might perform well, it is advisable to conduct comparative analyses across various algorithms to validate their effectiveness and ensure consistent and reliable results under varying data conditions [167].

Ablation studies also allow researchers to assess the importance of various components within a DL architecture by adding or removing NN layers and observing the resulting impact on performance. This technique has become an important tool for analysing the roles of different elements in ML systems [168]. In recent developments, ablation studies have been applied to quantify the contributions of different model components, as seen in methods such as permutation importance in Random Forests [88] and Shapley values in Shapley Additive Explanations (SHAP) [169]. These analyses provide a theoretical basis for understanding both the individual and collective effects of model components on performance and optimisation [170].

2.5.5 *Model evaluation and visualization*

The performance of a model is evaluated using various metrics adapted to the specific ML algorithm employed [37]. There is no single metric to measure this performance; instead, numerous performance metrics exist within the ML field [171]. Consequently, the selection of an appropriate metric depends on the specific problem, its domain, and real-world constraints [165].

Supervised learning metrics

Binary classification tasks use two classes: positive (belonging to one class) and negative (not belonging to that class) [172]. In this context, an ML algorithm's ability to classify positive and negative classes, is measured using the following counts: TP (true positive), TN (true negative), FP (false positive), and FN (false negative). TP is the number of examples correctly classified as positive. Conversely, TN is the number of examples correctly classified as negative. FP counts the number of negative examples that were incorrectly classified as positive, and FN is the number of positive examples incorrectly classified as negative.

Classification metrics [172, 173] utilize these counts to provide comprehensive performance measures, as outlined below:

- **Accuracy**: The proportion of correctly classified examples, both positive and negative.
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$
- **Precision**: The proportion of true positives among all predicted positives.
$$\text{Precision} = \frac{TP}{TP + FP}$$
- **Recall (Sensitivity)**: The proportion of true positives out of all actual positives.
$$\text{Recall} = \frac{TP}{TP + FN}$$
- **Specificity**: The proportion of true negatives out of all actual negatives.
$$\text{Specificity} = \frac{TN}{TN + FP}$$
- **F1-Score**: The harmonic mean of precision and recall, providing a balance between the two metrics.
$$\text{F1-Score} = \frac{2TP}{2TP + FP + FN}$$
- **Area Under the Curve (AUC)**: Evaluates the model's ability to distinguish between classes. First, the ROC (Receiver Operating Characteristic) curve is generated by plotting sensitivity versus specificity at various threshold settings [173].

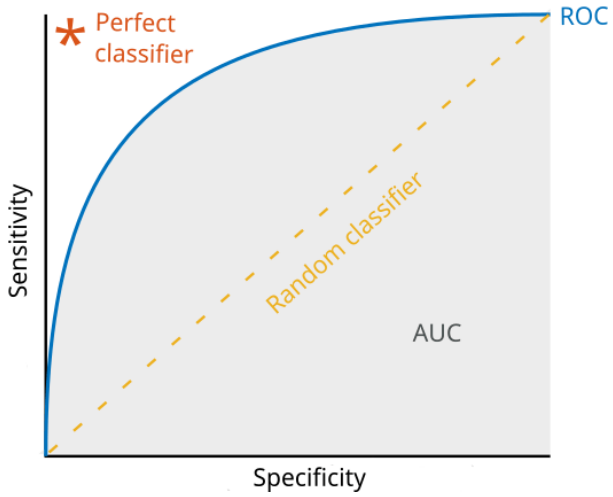


Figure 7: ROC curve. Source mathworks[2].

Then, AUC is computed as the area under this ROC curve, yielding a value between 0 and 1, where higher values indicate better model performance. Figure 7 shows a sample ROC curve alongside that of a random classifier, and it displays the corresponding AUC value.

The metrics described above for binary classification extend to the multilabel case [174, 175], where each instance may belong to multiple categories simultaneously. In multilabel classification, one can calculate evaluation measures for each label individually and then aggregate them using strategies such as micro-averaging, macro-averaging, or the weighted strategy.

- **Micro-averaging:** This technique aggregates the counts of TP, FP, and FN across all labels, then calculates the metrics from these totals. It emphasizes labels with more instances, which proves useful when dealing with imbalanced label frequencies.
- **Macro-averaging:** This approach computes the performance metrics for each label separately and then averages them. It treats each label equally regardless of frequency, offering a balanced evaluation across all labels.
- **Weighted Strategy:** This approach calculates the metrics for each class (label) and then computes a weighted average based on the number of true instances for each label. This strategy works best for unbalanced problems because it assigns more importance to labels that occur more frequently.

Sometimes, instead of using a native multi-class classifier, it is useful to decompose the multi-class problem into several binary problems, allowing the use of binary classifiers. Two widely used decomposition strategies are One-vs-Rest (OvR) and One-vs-One (OvO) [175]:

2 https://www.mathworks.com/help/deeplearning/ug/compare-deep-learning-models-using-ROC-curves.html

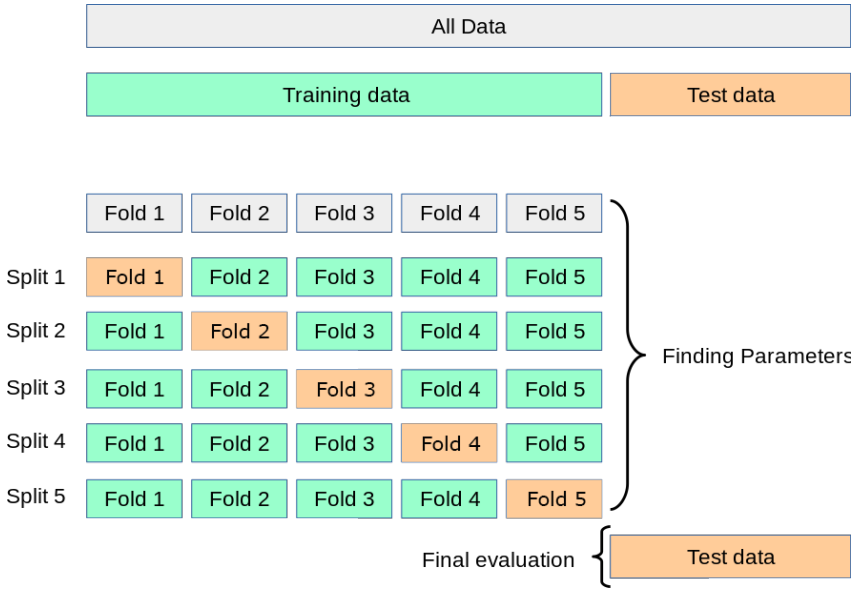


Figure 8: 5-fold cross validation. Source scikit-learn[3].

- **OvR:** Trains a distinct binary classifier for each label, treating that label as positive and all others as negative. This approach is easy to implement and works well in many scenarios.
- **OvO:** Trains a separate classifier for each pair of labels. With K labels, one ends up training $\frac{K(K-1)}{2}$ classifiers. Although it requires more models, this strategy can sometimes improve results, especially when label interactions are important.

To evaluate the generalization performance of a machine learning model by making better use of the available data k-fold cross-validation is commonly employed [176]. It helps estimate how well the model will perform on unseen data while reducing variability compared to a single train-test split. In this technique, the dataset is first split into a training set (in most works 80% of the data) and a test set (the remaining 20%). The training set is then divided into k equally sized folds. During each of the k iterations, one fold is held out for validation while the remaining $(k-1)$ folds are used to train the model. This ensures that every fold is used once as the validation set. The final performance is usually determined by averaging the chosen metric (such as accuracy) over all k iterations. Figure 8, shows an example of 5-fold cross-validation: after the initial training/test split, the training set is divided into five subsets, and the model is trained and evaluated five times, each time using a different subset as the validation set and the remaining four as the training set.

3 https://scikit-learn.org/stable/modules/cross_validation.html

Unsupervised learning metrics

Clustering is difficult to evaluate, as we do not know the ground-true, i.e. we do not have labeled data, to evaluate whether the clustering algorithm has grouped each individual in the right cluster. However, there are some metrics that could give some insight into how good the clustering is based on the distances between groups or the balance between groups, or the density of individuals in each group. The three most popular internal evaluation metrics in the literature [177] are silhouette coefficient, calinski-harabasz score, and davies-bouldin index. All of these metrics are based on distances between data points and are commonly used to evaluate the effectiveness of any clustering algorithm, working especially well in algorithms that work with distances, such as those included in the hierarchical, partitional, or spectral categories.

- **Silhouette Coefficient** (SC): measures the similarity, based on distances, of an individual to its own cluster compared to other clusters [178]. The coefficient value ranges between $[-1, 1]$, where 1 represents a good clustering division and a value close to -1 represents a poor division.

  The silhouette coefficient of one data point $i \in C_i$ is:

  $$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \; if \; |C_i| > 1, \;\; s(i) = 0 \; if \; |C_i| = 1$$

  Where C_i represents the cluster to which the data point i belongs, and $|C_i|$ is the cluster size, i.e. the total number of points contained in C_i.

  $$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} ||j - i||, \;\; b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} ||j - i||$$

  Where $a(i)$ is the average distance between a data point i and all other data points in the same cluster C_i, and $b(i)$ is the average distance between a data point i and all data points in the nearest cluster other than C_i.

  For n the total number of data points, the global silhouette coefficient is defined as:

  $$SC = \frac{1}{n} \sum_{i=1}^{n} s(i)$$

- **Calinski-Harabasz Score** (CH): like the silhouette coefficient, measures how similar an individual is to its group relative to other groups [179]. A higher value minimizes the intracluster covariance of individuals and maximizes the intercluster covariance. In cluster analysis, the within-group variance and between-group variance can be calculated by sum-of-squares within a cluster (SSW) and sum-of-squares between clusters (SSB) respectively.

Sum of Squared Within (SSW): minimizes the distance between individuals in the same cluster (cohesion).

$$SSW = \sum_{i=1}^{k} \sum_{i \in C_i} ||i - m_i||^2$$

where k is the number of clusters, i is a point of cluster C_i and m is the centroid of a cluster C_i.

Sum of Squared Between (SSB): maximizes the distance between individuals from different clusters (separation).

$$SSB = \sum_{j=1}^{k} |C_j| ||m_j - \bar{x}||^2$$

where k is the number of clusters, $|C_j|$ is the number of elements in a cluster j, m_j is the centroid of the cluster j and $\bar{x}$ is the mean of the dataset.

The CH score is the division between both variances:

$$CH = \frac{SSB(n-k)}{SSW(k-1)}$$

where k is the number of clusters and n is the sample size.

- **Davies-Bouldin Index** (DBI): Small values indicate compact clusters with well-differentiated centers that are far apart from each other [180].

$$DBI = \frac{1}{k} \sum_{i=1, i \neq j}^{k} \max\left(\frac{\sigma_i + \sigma_j}{||C_j - C_i||}\right)$$

where k is the number of clusters, σ_p is the average distance between each point in a cluster p and the centroid of its cluster (with $p \in \{i, j\}$) and $||C_j - C_i||$ is the distance between the centroids of the two clusters.

These distance-based metrics may not be suitable for algorithms relying on the Expectation Maximization (EM) method, such as the Gaussian Mixture algorithm, because EM models data using probability distributions rather than distances among data points. Consequently, we might see inaccuracies when comparing the performance of such algorithms if we employ these metrics. Instead of distance-based metrics, distribution-based algorithms typically use statistical criteria to decide the optimal number of clusters or components that best fit the data [181].

- **Information Criterion (IC)**: IC measures how well a statistical model fits the data distribution while penalizing overfitting [182].

$$IC(k) = -2 \cdot L(\hat{\theta}_k) + c_N \cdot k$$

where θ_k is the estimator of the parameter vector for the mixture model of order k, L is the log-likelihood function, N is the number of observations, and c_N is an increasing function of N. The optimal number of clusters is the one that minimizes the IC.

Below are two of the most well-known variations of IC used in the literature [183]:

- **Akaike information criterion (AIC):** AIC is a specific instance of the general information criterion (IC), where $c_N = 2$. This criterion is known for overestimating the model order.

$$AIC(k) = -2 \cdot L(\hat{\theta}_k) + 2 \cdot k$$

- **Bayesian information criterion (BIC):** Attempts to mitigate AIC's tendency to overestimate. The penalty term depends on the sample size N, so as $N \to \infty$, the penalty grows larger and avoids overestimating the mixture order as much as AIC does [184].

$$BIC(k) = -2 \cdot L(\hat{\theta}_k) + \log N \cdot k$$

Data visualization

Data visualization is important in ML pipelines, as it helps to explore data statistics and study the distribution of the data at any step of the pipeline [185, 37]. However, this process is not limited to the steps of data acquisition and preparation [19]; it can also be applied in the model evaluation step to make decisions about hyperparameter settings, algorithms, and normalization used in ML pipeline. These graphical visualizations facilitate the interpretation of the results and increase the explainability of the models [186]. For example, the elbow method is used to visually determine the optimal number of clusters by identifying the inflection point in the evaluation curve [187]. Figure 9, shows an example of an elbow method graph in which the optimal cluster number (in this example, four) achieved by a K-Means algorithm is evaluated based on the distance between clusters. Similarly, the use of scatter and box plots helps to show the clusters performed by clustering methods based on chosen features. Histograms and radar plots in ablation studies clearly illustrate the changes in performance when modifying or removing features [149, 188]. Combining these visual techniques with metric evaluations allows researchers to optimize the configuration and performance of the model [189].

2.5.6 *Model deployment*

After developing a ML model, it can be deployed into production using various techniques:

- **Containerization:** This involves packaging the model and all its dependencies into isolated environments, ensuring consistency across platforms [190].

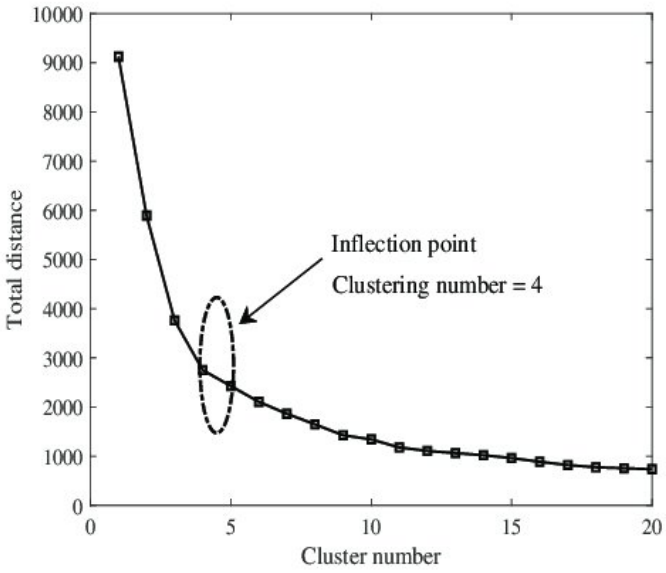


Figure 9: Elbow method for *K*-Means example. Source [32].

- **Microservices Architecture:** The application is divided into small, independent services that are developed, deployed, and scaled separately, which enhances flexibility and fault tolerance [191].
- **Serverless Computing:** This approach abstracts the underlying infrastructure by dynamically allocating resources as needed, simplifying scalability and reducing management overhead [192].
- **Edge Computing:** This technique deploys the model closer to the data source, reducing latency and improving responsiveness in real-time applications [193].

These deployment techniques, although widely applied, fall outside the scope of this thesis.

3

OBJECTIVES

The objective of this thesis is to develop ML tools and methodologies to optimize ML pipelines, with a specific focus on tourism data management in smart villages. To this end, we set the following objectives:

1. **Adapt smart city data collection to villages, adjusting it to the data and resource constrained environments.** Evaluating how to deploy sensors in areas with limited infrastructure, while ensuring that the data collected reflect travel patterns and visitor behaviors of smaller communities.
2. **Combine sensor data with contextual information to enrich data analysis.** Exploring how data sources such as holiday calendars, vehicle provenance or socioeconomic factors can improve the analysis of tourism patterns.
3. **Explore normalization methods and their influence on the performance of ML algorithms according to data distribution.** Evaluating statistical techniques (such as min-max, z-score, MAD or ℓ^2) in data preprocessing step and their influence in different clustering models using visualization tools.
4. **Identify the most important datasets sources through ablation studies.** Expanding the feature selection task in the data validation step, balancing algorithms' performance and data collection costs, while enhancing the explainability of ML models.
5. **Develop DL architectures that integrate attention mechanisms and work with limited datasets.** Designing NNs with self-attention and other layers to speed up convergence, add nonlinearity to learning and avoid overfitting, directly addressing data-limited training problems.
6. **Validate the ML algorithms and methodologies, designed on a case study of rural tourism.** Applying them to real problems or situations, testing the proposed approaches in cases such as: identifying residents and tourists in various groups according to their behavior, predicting the number of nights of stay of vehicles entering a rural area, or classifying visitors who will revisit a rural area from those who will not.

4

METHODOLOGY

The development of this thesis follows a theoretical-practical methodology, combining the design of an IoT device infrastructure with the development of ML pipelines. Therefore, we need a strategy that follows the guidelines of the traditional scientific method, adapted to address the specific requirements of the study. The steps and adaptations of the scientific method applied in this study are detailed below:

1. **Review Literature:** Conduct a systematic review of the most relevant publications on ML pipelines, data analysis, DL architectures, and smart villages.
2. **Detect Research Gaps and identify Proposals:** Identify specific areas that require further research based on the literature review. Identify proposals that address these gaps.
3. **Formulate Hypotheses:** Define hypotheses that focus on applying ML pipelines to integrate heterogeneous data sources and address challenges.
4. **Refine Hypotheses Through Scientific Publications:** Present hypotheses and proposals at conferences, workshops, and forums, to gather feedback, refine initial hypotheses.
5. **Demonstrate Contributions via ML Pipelines:**
    - **Develop ML Pipelines:** Design and implement ML pipelines that integrate collected data, including ablation studies, data preprocessing, model training, and validation steps.
    - **Assess Performance Metrics:** Use key evaluation metrics such as accuracy, precision, recall, F1-score, and AUC to measure the effectiveness of the developed pipelines and individual models.
    - **Compare Benchmarks:** Evaluate performance by comparing different models within the pipeline to highlight the strengths and weaknesses of each approach.
    - **Validate in Real Environments with Sensor Data:** Analyze collected data through the ML pipeline to confirm the applicability of the proposed solutions in real-world scenarios.

Case study setup for validation

We determined the locations for installing LPR cameras by analyzing viewing angles, lighting, and vehicular flow to maximize data capture and minimize costs. The vehicle tracking infrastructure consists of four Hikvision LPR IP devices equipped with vehicle detection sensors. These 2MP devices feature automatic number-plate recognition (ANPR) with deep learning, a 2.8-12 mm varifocal lens, and a 50 m IR range. We developed this infrastructure across three villages in the Barranco de Poqueira region (Pampaneira, Bubión, and Capileira) in Sierra Nevada, Granada, Spain. To cover the entrances and exits of each village, we strategically positioned the four cameras, as shown in Figure 10. The locations include (i) the entrance to Pampaneira from the western part of the Alpujarra, (ii) the entrance to Pampaneira from the eastern part of the Alpujarra, (iii) the entrance to Bubión via a single road, and (iv) the entrance to Capileira via a single road. By leveraging the road structure, we monitor the mobility of all vehicles circulating in the Poqueira area using only four LPRs. As Capileira has no exit at the top, its entrance camera also monitors the exits from the municipality. As there is only one road linking the municipalities of Capileira and Bubión, the Capileira camera also functions as an exit camera for Bubión. This configuration covers all the entrances and exits of every village, and consequently the vehicle movements within each village, eliminating the need for six LPRs. After installation, we continuously monitored and adjusted the cameras to ensure optimal performance.

Figure 10: Setup of the 4 LPR that obtain the data from the license plates of the vehicles.

We design a questionnaire consisting of 17-questions, and we conducted it in Pampaneira village. We collected 522 questionnaires by interviewing drivers in the parking area, gathering demographic and behavioral data related to vehicle usage and visitor patterns. The surveyor visually confirms the license plate numbers to ensure

that the vehicle corresponds to the respondent. This information allows us to merge the data from the questionnaires with the information collected by the LPRs. The questionnaires took place in January, March, and July 2023, targeting a visitor population that excluded local residents. In order to maintain a proportion of visitors surveyed equivalent to the existing percentage in the LPRs data, a prior analysis of the origin of vehicles to the area was carried out based on the geographical location and the Gross Domestic Product (GDP). The areas were defined as follows: Area 1 (Areas nearby with low GDP), Area 2 (Intermediate areas with low to medium GDP), Area 3 (Intermediate areas with high GDP), Area 4 (Distant areas with high GDP), and Area 5 (Distant areas with low GDP). The questions collected cover the following points:

- One of the questions included was the intention to visit the area (in number of visits) in the next 12 months.
- Six questions related to LPR information (entry time, estimated exit time, number of visits in the past year, overnight stays, license plate number, and residential postcode) functioned as control variables to validate the data against LPR camera records.
- Eight questions which contains personal information such as age, gender, annual income, education level, number of passengers, and employment status.
- Two tax-related questions: one about how much money visitors would be willing to pay if a parking toll is installed in the area, and another about whether they intend to visit under those circumstances.

5

RESULTS

The results of this thesis are presented through several case studies, each of which uniquely contributes to understanding and addressing the challenges of data collection and integration, assessing the explainability of features, or developing models with limited datasets.

5.1 CLUSTERING PIPELINE FOR VEHICLE BEHAVIOR IN SMART VILLAGES

This study focuses on developing a clustering process to analyze vehicle mobility in a rural tourism region by integrating LPR sensor data with various heterogeneous contextual data sources. The data processing pipeline comprises eight steps: data collection, cleaning, fusion, normalization, dimensionality reduction, clustering, evaluation, and visualization. Although the previously presented ML *pipeline* consisted of seven steps. In our proposal, we have discarded the model deployment step, as it is outside the scope of the thesis. In addition, we have subdivided other steps, such as normalization, dimensionality reduction (present in the preprocessing step) or visualization (separate from the evaluation step). These divisions allow us to analyze more precisely the techniques present in each step within our pipeline. The validation step for example, we have subdivided it in two to address some challenges such as the fusion of heterogeneous sources or data cleaning, which are important in our case study.

Over a nine-month period, we gather data using four strategically placed LPR sensors, resulting in more than 50,000 unique vehicle records enriched with contextual information such as vehicle mobility during public holidays[1] based on national and local calendars, vehicle provenance information from the Spanish General Directorate of Traffic (DGT)[2], and socio-economic factors of vehicle origin obtained from the Spanish National Statistics Institute (INE)[3]. This publication details the design of the IoT infrastructure for data collection, and defines the features we utilize in constructing the study's primary database, that we expand in the following works.

1 https://python-holidays.readthedocs.io/en/latest/
2 https://sede.dgt.gob.es/
3 https://www.ine.es/dynt3/inebase/es/index.htm?padre=7132&capsel=5693

The integration of heterogeneous datasets, introduces a challenge that requires developing data engineering methods and cleaning process. Normalization serves as an important component of the pipeline. We compare normalization techniques such as min-max normalization, Z-score standardization, and MAD. For each normalization method, we analyzed the cumulative variance for each component of the PCA. ℓ^2 obtained the highest cumulative variance, indicating that it retained the most information in only two components, followed by the min-max method. Z-score standardization and MAD obtained a higher value for a larger number of components, which makes it difficult to visualize and explain the model, and will be eliminated from further study. We evaluate various clustering algorithms, including *K*-Means, Agglomerative clustering, DBSCAN, and Gaussian Mixture Models. Based on the geometry of data distribution and preliminary experiments, we select Gaussian Mixture Models as the main model. We use the BIC and AIC metrics, along with the elbow method, to determine the optimal number of clusters. Under min-max normalization, the optimal model is achieved with seven components (BIC: –87,585, AIC nearly identical), while ℓ^2 normalization reaches its best configuration with four components (BIC: –269,828, with a corresponding AIC value). We found that min-max normalization was the most effective for precisely segmenting individuals and analyzing their visiting behavior in the area. It also excelled at identifying atypical behavior in individuals who are not registered residents but behave like residents. Additionally, ℓ^2 normalization can be useful in specific cases where distinguishing the region of origin is necessary. The results show that the choice of the normalization method significantly affects clustering outcomes.

The experiments successfully identifies distinct behavioral patterns, such as short-term tourists and long-term residents, and demonstrates the explainability of features like visit frequency, total nights or distance traveled for these clusters. This analysis could assist area managers in crafting tailored strategies to keep certain tourists, considering their income and origin, and promoting overnight stays. Additionally, these patterns could inform policies to engage non-registered residents in the community, such as tax breaks or social programs.

5.2 Predicting overnights in smart villages: the importance of context information

This publication extends prior work by employing other ML pipelines while transitioning from an unsupervised to a supervised learning paradigm for a classification problem. The study explore the integration of contextual information with LPR data to predict vehicle stay durations in rural tourist areas, measured by the number of overnights. The analysis is based on data collected over a 17-month period, significantly extending the time-frame of the previous study. The 35 features that composed the new dataset come from five different data sources: one base dataset (extracted from the LPR database), one visit-specific calculated metrics dataset, and three context datasets (holiday, socio-economic, and spatio-temporal vehicle entry).

The study employed ML models such as decision trees, naive bayes, SVM, gradient boosting models, and tabular transformers to build the predictive model. We carried out hyperparameter optimization to select the most efficient models adapted to the dataset's features. Additionally, we applied various ensemble methods like stacking, bagging, and voting to enhance model performance. Among the evaluated classifiers, NN models such as MLP and TabNet and gradient boosting algorithms such as LightGBM, XGBoost, and CatBoost produced the best results. LightGBM delivered the best performance, achieving an AUC of approximately 0.8012 and an F1 score of around 0.7273 on the test validation set. Gradient boosting algorithms also achieved faster processing times in average compared to MLP and TabNet executions. Using different ensemble strategies, we combined LightGBM, XGBoost, and CatBoost getting an improvement in AUC, reaching up to 0.8025. Bagging experiments also showed that using 200 estimators for XGBoost provided a result of 0.8024 AUC score, although at a significant increase in processing time.

We incorporated an ablation study step into the pipeline for each designed dataset. Unlike conventional ablation studies, which focus on assessing the contribution of NN layers, our approach analyses the impact of datasets composed of different features from a common information source. Our experiments show that removing less relevant datasets reduced processing time by 22.2% and decreased model complexity by 80%, with only a minimal impact on predictive performance (AUC decreased by 0.01). Furthermore, this publication introduced a systematic methodology for evaluating the contributions of features at the dataset level rather than individually, thereby addressing a gap in traditional feature selection approaches. For instance, socio-economic indicators such as gross income by vehicle provenance and entry context features, such as the camera that detects the vehicle's entry into the area or the time of day when it occurs, significantly improved the model's predictions. This research is useful for scientists developing predictive models in the field of tourism. By identifying the most important databases, our results guide them in strategically allocating their resources to obtain and handle specific datasets. This becomes particularly advantageous when they encounter resource constraints concerning finances and time allocation for a given project.

5.3 SASD: Self-Attention for Small Datasets – A Case Study in Smart Villages

The third publication addresses the methodological challenge of working with small datasets, a common limitation in rural tourism studies. This study introduces SASD (Self-Attention for Small Dataset) architecture, a NN designed to leverage the limited data from visitor questionnaires. The model incorporates self-attention layers to evaluate feature relevance, addressing challenges such as data sparsity and noise.

The architecture combines linear layers, batch normalization, dropout, and self-attention mechanisms. From different ablation studies, we found that the best configuration

of self-attention layers was at the beginning and at the end of the NN, optimising the model's ability to capture long-term dependencies and contextual relationships between features. The study evaluated SASD by comparing its performance with state-of-the-art algorithms, including widely used classification models such as random forest, *K*-NN, and gradient boosting, as well as advanced DL models like RNN, TabNet, and TabTransformer. Metrics such as precision, recall, F1-score, and training time were used for benchmarking. To prove our proposal, we perform two additional experiments. In the first experiment, we use a baseline multiclass NN without attention. This architecture obtained the best F1-score, accuracy, precision and recall results in 270 epochs, with a weighted average F1-score of 0.74. In the second experiment, we added multi-headed attention layers in place of self-attention layers in SASD, named Multihead-Attention Small Dataset (MASD), to introduce a parallelization of the tasks. We used four heads of attention in the construction of MASD based on experimentation with values between 2-64. This architecture obtained the best F1-score, accuracy, precision and recall results in 210 epochs, with a weighted average F1-score of 0.71. Algorithms such as RNN or LSTM, as well as TabTransformer or TabNet, got worse results than the previously described architectures.

Specifically, SASD outperforms traditional classification algorithms by up to 3% on the weighted average F1-score. Our configuration achieves a value of 0.75 for the weighted F1 score metric. Additionally, this model has fewer epochs (converges only in 120 epochs) and consequently a lower preprocessing time than the other NNs (up to 32% faster than the baseline NN without attention version). The research also included ablation studies to assess the impact of architectural components like ReLU activation and dropout layers, demonstrating that their inclusion improved generalization and prevented overfitting. From a practical perspective, the SASD model was applied to predict tourists' intentionality to revisit within 12 months. The results revealed patterns useful for marketing strategies and optimizing tourism infrastructure in smart villages. This study demonstrates the potential of integrating attention mechanisms into NNs for small datasets, addressing a critical need in ML research for rural contexts.

6

CONCLUSIONS

This thesis presents a methodology aimed at developing ML pipelines to collect, analyze and evaluate the behavior of vehicles in smart villages. The approach encompasses the entire workflow, from data acquisition and dataset construction using LPR cameras integrated with various contextual data sources, to the creation of various ML models with the merged data. The techniques presented in this methodology address the challenges inherent in rural environments and effectively achieve each of the objectives presented in the dissertation.

The first objective focused on extrapolating the sensor data collection design used in smart cities to the context of smart villages. To this end, we designed an IoT infrastructure using a minimal number of LPR cameras. By leveraging the road layout, a single camera can monitor both the entrance of one village and the exit of another, thereby reducing the number of devices needed to four. After collecting the data, we validated it and compiled it into a dataset that forms a strong foundation for further analysis and model development. In addition, to adapt our solutions to rural environments with limited infrastructure, we developed the SASD architecture. This model performs well with limited data, significantly reducing the time and resources required for tasks such as survey data collection. Finally, we conducted ablation studies to identify the most influential datasets, further minimizing the expenditure of invested resources.

The second objective was to improve the quality and depth of the data by integrating various heterogeneous sources of contextual data. This integration was intended to complement the information collected by the LPR cameras with additional data that would allow for a more comprehensive analysis of vehicle mobility patterns in the area. Contextual datasets included national holidays, vehicle provenance information, and socio-economic indicators. In addition, on-site questionnaires were conducted to enrich the dataset, providing a deeper understanding of visitors mobility behaviors. The resulting dataset has been used to solve different problems using ML models, obtaining in all cases better results than those that would have been obtained only using LPR data.

The third objective focused on normalization. We first conducted an analysis of the distribution of the data and its geometry. We explored various normalization algorithms; including min-max, Z-score standardization, ℓ^2, and MAD to assess their impact on data distribution and model behavior. To achieve this, we applied the four most common normalization techniques and conducted PCA analysis accounted for most of the variance regardless of the method. An exploratory visual analysis of the first two components revealed that min-max and ℓ^2 normalization yielded notably different data clusters. Z-score and MAD required more than two components, complicating model visualization and explanation, so they were excluded from subsequent analysis. In particular, min-max normalization proved most effective for detailed segmentation of individuals and for detecting atypical behaviors such as individuals not registered as residents but exhibiting resident-like patterns while ℓ^2 normalization may be advantageous in scenarios where distinguishing the region of provenance is important.

The fourth objective was identifying and analyzing the most influential features and sources of data within the dataset. This was done by assessing the explainability of features in unsupervised models using cluster analysis, which helped us assess the impact of each feature on every cluster. In addition, ablation studies were performed on each dataset of the data collected to assess their importance in different classification models. This analysis allowed the identification of essential data sources, thus eliminating data collection efforts that would entail excessive monetary and time costs without providing substantial analytical value.

The fifth objective was to create deep learning models to improve predictions with small datasets. We developed a transformer-based model for tabular data, called SASD. The SASD model demonstrated superior performance to other popular classification algorithms when applied to the limited dataset composed in part of responses obtained from questionnaires. The results highlight the potential of advanced deep learning techniques and self-attention layer aggregation in data-limited scenarios.

The last objective focused on validating the developed ML pipelines in real rural tourism flow scenarios. All ML pipelines in the paper were built and tested using real data collected from sensors deployed in a real smart village environment. This validation process not only employed different ML algorithms adapted to various learning paradigms, but also ensured that each learning task addressed significant issues relevant to policymakers managing rural areas.

OTHER PUBLICATIONS DERIVED FROM THE THESIS

In addition to the published articles that support this thesis as a compendium, two additional JCR-indexed works related to open source data and software, as well as open datasets and conference papers[1].

1 Displayed citation, download, and indicator counts reflect data up to April 22, 2025.

Data publication

The first JCR publication presents a dataset for vehicle tracking in the Barranco de Poqueira region using four LPR cameras. The dataset, covering February to October 2022 (now updated until August 2023), includes raw data, visit-level aggregation, and vehicle-level aggregation enriched with contextual information, making it valuable for mobility, urban planning, tourism, and socio-economic studies.

[i] Bolaños-Martinez, D., Bermudez-Edo, M., Garrido, J. L., & Delgado-Márquez, B. L. (2024). Spatio-temporal dynamics of vehicles: Fusion of traffic data and context information. *Data in Brief*, 53, 110084. *[JCR ESCI 2023 - JIF Q3; IF 1.0]*. DOI: https://doi.org/10.1016/j.dib.2024.110084. Number of citations: 1 (Source, Google Scholar).

In addition to this data paper, other open datasets have been published on Zenodo [2], including raw datasets for questionnaires and LPRs, as well as an additional dataset with the intersection of both.

[ii] Bolaños-Martinez, D., Bermudez-Edo, M., Garrido, J. L., Delgado Márquez, B. L., Urriza, J. I., & Aragon-Correa, J. A. (2023). Federation of Vehicular Data in Smart Villages with Socioeconomic Information. Federation of Vehicular Data in Smart Villages with Socioeconomic Information. DOI: https://doi.org/10.5281/zenodo.14262136. Number of downloads: 320. Number of citations: 1 (Source, Google Scholar)

[iii] Urriza, J. I., Bolaños-Martinez, D., Delgado Márquez, B. L., Garrido, J. L., Bermudez-Edo, M., & Aragon-Correa, J. A. (2023). PorqueiraSurveys: A Dataset on Economic Impact Surveys in the region of Barranco del Poqueira in the Alpujarra Granadina. PorqueiraSurveys: A Dataset on Economic Impact Surveys in the region of Barranco del Poqueira in the Alpujarra Granadina. DOI: https://doi.org/10.5281/zenodo.8328348. Number of downloads: 13. Number of citations: 1 (Source, Google Scholar)

[iv] Bolaños-Martinez, D., Urriza, J. I., Delgado Márquez, B. L., Garrido, J. L., Bermudez-Edo, M., & Aragon-Correa, J. A. (2023). PoqueiraVehicleLPR: A Dataset of Vehicle Detection Sensors in the region of Barranco de Poqueira in the Alpujarra Granadina. PoqueiraVehicleLPR: A Dataset of Vehicle Detection Sensors in the region of Barranco de Poqueira in the Alpujarra Granadina. DOI: https://doi.org/10.5281/zenodo.8356386. Number of downloads: 12. Number of citations: 0 (Source, Google Scholar).

[v] Durán-López, A., Bolaños-Martinez, D., Bermudez-Edo, M., Delgado Márquez, B. L., & Aragon-Correa, J. A. (2024). Smart Poqueira: Predicting Rural Parking Lot Feasibility with Sensor-Questionnaire Integration. Smart Poqueira: Predicting Rural Parking Lot Feasibility with Sensor-Questionnaire Integration. DOI: https://doi.org/10.5281/zenodo.11112791. Number of downloads: 67. Number of citations: 0 (Source, Google Scholar).

2 https://zenodo.org/

Software publication

The second JCR publication, where the author is listed as the second, introduces RouteRecoverer, a tool to address LPR sensor limitations by reconstructing vehicle routes and recovering missing license plate digits, improving data quality and filling gaps in routes caused by incomplete detections.

[vi] Durán-López, A., Bolaños-Martinez, D., Delgado-Márquez, L., & Bermudez-Edo, M. (2024). RouteRecoverer: A tool to create routes and recover noisy license plate number data. *Software Impacts*, 20, 100636. *[JCR ESCI 2023 - JIF Q3; IF 1.3]*. DOI: https://doi.org/10.1016/j.simpa.2024.100636. Number of citations: 0 (Source, Google Scholar).

Conferences

Additionally, several research and dissemination works have been presented at national and international conferences during the development of the thesis.

One such work, presented at 14th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI 2022). This preliminary study on traffic data, which was later utilized in the first publication, involved initial data analysis using clustering algorithms and the construction of a simplified version of the dataset that was ultimately used for the main research.

[vii] Bolaños-Martinez, D., Bermudez-Edo, M., & Garrido, J. L. (2022, November). Clustering study of vehicle behaviors using license plate recognition. In *International Conference on Ubiquitous Computing and Ambient Intelligence* (pp. 784-795). Cham: Springer International Publishing. DOI: https://doi.org/10.1007/978-3-031-21333-5_77. Number of citations: 4 (Source, Google Scholar).

Another work, presented at Jornadas sobre la Enseñanza Universitaria de la Informática (JENUI 2023) conference. Although the topic is related to education, specifically a role-based active and collaborative learning methodology, it employed data visualization tools (e.g., boxplots and radar plots) and statistical tests such as the Kruskal-Wallis test to analyze the results.

[viii] Bolaños-Martinez, D., et al. "Un enfoque innovador para el aprendizaje activo y colaborativo basado en juegos de rol". En: Cruz Lemus, José Antonio; Medina Medina, Nuria; Rodríguez Fórtiz, María José (eds.). Actas de las XXIX Jornadas sobre la Enseñanza Universitaria de la Informática, Granada, 5-7 de julio de 2023. Granada: Asociación de Enseñantes Universitarios de la Informática, 2023, pp. 335-342. URI: http://hdl.handle.net/10045/137219. Number of citations: 1 (Source, Google Scholar).

Dissemination

The work has also been disseminated at various events, including the II International Artificial Intelligence Forum in Andalusia 2024 in Jaen, the European Night of Researchers 2024 in Granada, and the CITIC Coffee 2024 in Granada.

We have also had an impact on local and national newspapers as a result of numerous publications in press media such as: Europa Press[3], Canal UGR[4], Granada Hoy[5], and IDEAL[6].

[ix] Europa Press Andalucía. (2022, noviembre 22). Colocan en Pampaneira (Granada) sensores que miden los vehículos que entran y salen del pueblo. Europa Press. https://www.europapress.es/andalucia/noticia-colocan-pampaneira-granada-sensores-miden-vehiculos-entran-salen-pueblo-20221122184149.html

[x] UGRDivulga. (2024, febrero 23). Investigadores de la UGR utilizan cámaras y técnicas de Inteligencia Artificial para analizar los patrones de comportamiento de los coches en la Alpujarra. Canal ugr.es. https://canal.ugr.es/noticia/investigadores-de-la-ugr-utilizan-camaras-y-tecnicas-de-inteligencia-artificial-para-analizar-los-patrones-de-comportamiento-de-los-coches-en-la-alpujarra/

[xi] UGRDivulga. (2024, octubre 15). Investigadores de la UGR diseñan una IA para predecir la duración de estancias turísticas en la Alpujarra. Canal ugr.es. https://canal.ugr.es/noticia/investigadores-de-la-ugr-disenan-una-ia-para-predecir-la-duracion-de-estancias-turisticas-en-la-alpujarra/

[xii] Redacción Granada Hoy. (2024, octubre 15). ¿Cuánto tiempo se queda un turista en la Alpujarra? La IA ya lo predice. Granada Hoy. https://www.granadahoy.com/vivir/tiempo-queda-turista-alpujarra-ia_0_2002566526.html

[xiii] IDEAL. (2024, febrero 23). Investigadores de la UGR utilizan cámaras y técnicas de IA para analizar los coches en la Alpujarra. IDEAL. https://www.ideal.es/miugr/investigadores-ugr-utilizan-camaras-tecnicas-ia-analizar-20240223110813-nt_amp.html

3 https://www.europapress.es/
4 https://canal.ugr.es/ugrnews/
5 https://www.granadahoy.com/
6 https://www.ideal.es/

7

FUTURE WORK

In the future, we plan to expand our work by refining NN proposed architectures, experimenting with new visualization and explainability methods, integrating additional sensors in our designed infrastructure, and implementing federated learning techniques.

We will test the SASD model on datasets beyond the tourism case. We plan to use publicly available benchmark datasets recognized by the research community, representing various scenarios with and without data constraints. Furthermore, we will evaluate alternative DL methods by comparing our model's performance against both classic algorithms and the state-of-the-art techniques available at the time.

We will enhance our techniques for visualizing dense data by developing tools that reveal more complex relationships between data points. By combining clustering algorithms with fuzzy logic, we aim to establish clear boundaries between groups that follow distinct patterns, to improve classification models in tasks involving the separation of data points with close features. Additionally, we will integrate explainability tests such as SHAP, Local Interpretable Model-agnostic Explanations (LIME), and other approaches to clarify how the model makes decisions.

We will expand our IoT infrastructure by integrating new types of sensors. For instance, we plan to add sensors that count people, monitor weather conditions, control air quality, or measure waste levels in the area. We will explore innovative methods for fusing these heterogeneous data sources into a cohesive system that satisfies the specific requirements of each sensor.

We plan to integrate federated learning into our model training process, enabling distributed analysis of local data while preserving privacy. To this end, we intend to collaborate with tourism experts and policymakers in other geographic areas to incorporate their data and knowledge into our system. We will use their feedback to refine our methods, and we will develop case studies that document the deployment process and illustrate how our system adapts to real-world challenges to serve as an example for the rest of the scientific community working in the field of smart villages.

BIBLIOGRAPHY

[1] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[2] Tom M Mitchell and Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

[3] Paolo Fantozzi, Guglielmo Maccario, and Maurizio Naldi. Uncovering tourist visit intentions on social media through sentence transformers. *Information (2078-2489)*, 15(10), 2024.

[4] Yutong Wang. Expressway traveler classification algorithm based on toll data. In *International Conference on Smart Transportation and City Engineering (STCE 2023)*, volume 13018, pages 749–755. SPIE, 2024.

[5] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems*, 23(6):4909–4926, 2021.

[6] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[7] Ali Tarhini, Mariam AlHinai, Adil S Al-Busaidi, Srikrishna Madhumohan Govindaluri, and Jamil Al Shaqsi. What drives the adoption of mobile learning services among college students: An application of sem-neural network modeling. *International Journal of Information Management Data Insights*, 4(1):100235, 2024.

[8] Tin-Chih Toly Chen, Hsin-Chieh Wu, and Min-Chi Chiu. A deep neural network with modified random forest incremental interpretation approach for diagnosing diabetes in smart healthcare. *Applied Soft Computing*, 152:111183, 2024.

[9] Jijie Fan, Weikai Lu, Satyvaldieva Baktygul Abduraimovna, Jinlong Cheng, and Haoyi Fan. Graph-guided neural network for tourism demand forecasting. *IEEE Access*, 11:134259–134268, 2023.

[10] Lorenzo Brigato and Luca Iocchi. A close look at deep learning with small data. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2490–2497. IEEE, 2021.

[11] Saidul Islam, Hanae Elmekki, Ahmed Elsebai, Jamal Bentahar, Nagat Drawel, Gaith Rjoub, and Witold Pedrycz. A comprehensive survey on applications of transformers for deep learning tasks. *Expert Systems with Applications*, 241:122666, 2024.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[13] Jasmin Praful Bharadiya. Exploring the use of recurrent neural networks for time series forecasting. *International Journal of Innovative Science and Research Technology*, 8(5):2023–2027, 2023.

[14] Karimeh Ibrahim Mohammad Ata, Mohd Khair Hassan, Ayad Ghany Ismaeel, Syed Abdul Rahman Al-Haddad, Sameer Alani, et al. A multi-layer cnn-gruskip model based on transformer for spatial- temporal traffic flow prediction. *Ain Shams Engineering Journal*, 15(12):103045, 2024.

[15] Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6679–6687, 2021.

[16] Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

[17] Bjorn Barz and Joachim Denzler. Deep learning on small datasets without pretraining using cosine loss. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1371–1380, 2020.

[18] Shriyank Somvanshi, Subasish Das, Syed Aaqib Javed, Gian Antariksa, and Ahmed Hossain. A survey on deep tabular learning. *arXiv preprint arXiv:2410.12034*, 2024.

[19] Hannes Hapke and Catherine Nelson. *Building machine learning pipelines*. O'Reilly Media, 2020.

[20] Qiang Liu, Jiade Zhang, Jingna Liu, and Zhi Yang. Feature extraction and classification algorithm, which one is more essential? an experimental study on a specific task of vibration signal diagnosis. *International Journal of Machine Learning and Cybernetics*, pages 1–12, 2022.

[21] Bingchun Liu, Jiayi Pei, and Zhecheng Yu. Stock price prediction through grawd-bilstm model with air quality and weather factors. *International Journal of Machine Learning and Cybernetics*, pages 1–18, 2023.

[22] Aniruddha Maiti, Sai Shi, and Slobodan Vucetic. An ablation study on the use of publication venue quality to rank computer science departments: Publication quality is strongly correlated with the subjective perception of research strength. *Scientometrics*, 128(8):4197–4218, 2023.

[23] Sonali Mhatre, Saloni Patil, Navya Mishra, Vaibhav Mungelwar, and Harshada Patil. Automl based tourism prediction and maximising revenue. In *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, pages 1193–1202. IEEE, 2024.

[24] Pouya Khani, Elham Moeinaddini, Narges Dehghan Abnavi, and Amin Shahraki. Explainable artificial intelligence for feature selection in network traffic classification: A comparative study. *Transactions on Emerging Telecommunications Technologies*, 35(4):e4970, 2024.

[25] Verónica Bolón-Canedo, Laura Morán-Fernández, Brais Cancela, and Amparo Alonso-Betanzos. A review of green artificial intelligence: Towards a more sustainable future. *Neurocomputing*, page 128096, 2024.

[26] Shaolong Sun, Mingchen Li, Shouyang Wang, and Chengyuan Zhang. Multi-step ahead tourism demand forecasting: The perspective of the learning using privileged information paradigm. *Expert Systems with Applications*, 210:118502, 2022.

[27] Scott Peters and Peter Keller. Applications and issues of big data in tourism research. 2022.

[28] Peter Trebuňa, Jana Halčinová, Milan Fil'o, and Jaromír Markovič. The importance of normalization and standardization in the process of clustering. In *2014 IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 381–385. IEEE, 2014.

[29] Christiane Kirketerp de Viron. Eu action for-smart villages. 2017.

[30] Angel Paniagua. Smart villages in depopulated areas. *Smart village technology: Concepts and developments*, pages 399–409, 2020.

[31] Ejaz Ahmed, Ibrar Yaqoob, Ibrahim Abaker Targio Hashem, Imran Khan, Abdelmuttlib Ibrahim Abdalla Ahmed, Muhammad Imran, and Athanasios V Vasilakos. The role of big data analytics in internet of things. *Computer Networks*, 129:459–471, 2017.

[32] Jiachi Zhang, Liu Liu, Yuanyuan Fan, Lingfan Zhuang, Tao Zhou, and Zheyan Piao. Wireless channel propagation scenarios identification: A perspective of machine learning. *IEEE Access*, 8:47797–47806, 2020.

[33] Felix Krones, Umar Marikkar, Guy Parsons, Adam Szmul, and Adam Mahdi. Review of multimodal machine learning approaches in healthcare. *Information Fusion*, 114:102690, 2025.

[34] Eduardo Fernandes Montesuma, Fred Maurice Ngolè Mboula, and Antoine Souloumiac. Recent advances in optimal transport for machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[35] Chao Wang, Tao Li, Zhicui Lu, Zhenqiang Wang, Tmader Alballa, Somayah Abdualziz Alhabeeb, Maryam Sulaiman Albely, and Hamiden Abd El-Wahed Khalifa. Application of artificial intelligence for feature engineering in education sector and learning science. *Alexandria Engineering Journal*, 110:108–115, 2025.

[36] Shuvo Dip Datta, Mobasshira Islam, Md Habibur Rahman Sobuz, Shakil Ahmed, and Moumita Kar. Artificial intelligence and machine learning applications in the project lifecycle of the construction industry: A comprehensive review. *Heliyon*, 2024.

[37] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[38] Showmick Guha Paul, Arpa Saha, Md Zahid Hasan, Sheak Rashed Haider Noori, and Ahmed Moustafa. A systematic review of graph neural network in healthcare-based applications: Recent advances, trends, and future directions. *IEEE Access*, 2024.

[39] Mingchen Li, Chengyuan Zhang, Shaolong Sun, and Shouyang Wang. A novel deep learning approach for tourism volume forecasting with tourist search data. *International Journal of Tourism Research*, 25(2):183–197, 2023.

[40] Andrei Margeloiu, Nikola Simidjievski, Pietro Lio, and Mateja Jamnik. Weight predictor network with feature selection for small sample tabular biomedical data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9081–9089, 2023.

[41] Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? *Advances in Neural Information Processing Systems*, 36, 2024.

[42] Keshav Gupta, Vikas Kumar, Abhishek Jain, Pranita Singh, Amit Kumar Jain, and MSR Prasad. Deep learning classifier to recommend the tourist attraction in smart cities. In *2024 2nd International Conference on Disruptive Technologies (ICDT)*, pages 1109–1115. IEEE, 2024.

[43] Lirong Yin, Lei Wang, Siyu Lu, Ruiyang Wang, Youshuai Yang, Bo Yang, Shan Liu, Ahmed AlSanad, Salman A AlQahtani, Zhengtong Yin, et al. Convolution-transformer for image feature extraction. *CMES-Computer Modeling in Engineering & Sciences*, 141(1), 2024.

[44] Rhett N D'souza, Po-Yao Huang, and Fang-Cheng Yeh. Structural analysis and optimization of convolutional neural networks with a small sample size. *Scientific reports*, 10(1):834, 2020.

[45] Asif Ali Laghari, Kaishan Wu, Rashid Ali Laghari, Mureed Ali, and Abdullah Ayub Khan. A review and state of art of internet of things (iot). *Archives of Computational Methods in Engineering*, pages 1–19, 2021.

[46] Essam H Houssein, Mahmoud A Othman, Waleed M Mohamed, and Mina Younan. Internet of things in smart cities: Comprehensive review, open issues and challenges. *IEEE Internet of Things Journal*, 2024.

[47] CV Suresh Babu, CS Akkash Anniyappa, and Abhipsa Raut. Toward seamless mobility: Integrating connected and autonomous vehicles in smart cities through digital twins. In *Digital Twins for Smart Cities and Villages*, pages 169–187. Elsevier, 2025.

[48] Iftikhar Hussain, Adel Elomri, Laoucine Kerbache, and Abdelfatteh El Omri. Smart city solutions: Comparative analysis of waste management models in iot-enabled environments using multiagent simulation. *Sustainable Cities and Society*, 103:105247, 2024.

[49] Al-Siyam Rahman, Md Sadik Tasrif Anubhove, Mohammad Zeyad, SM Masum Ahmed, and Md Abul Ala Walid. Study of city crimes with smart solutions integrating into smart cities, internet of things (iot), and cybersecurity systems. In *2024 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC)*, pages 1–6. IEEE, 2024.

[50] Wenhao Li, Chengkun Liu, Tao Wang, and Yanjie Ji. An innovative supervised learning structure for trajectory reconstruction of sparse lpr data. *Transportation*, 51(1):73–97, 2024.

[51] Qinghai Lin, Jinyong Chen, Guilong Li, and Zhaocheng He. Signal timing parameters inference method at intersections using license plate recognition data. *IET Intelligent Transport Systems*, 2022.

[52] Changxi Ma, Xiaoting Huang, and Jiangchen Li. A review of research on urban parking prediction. *Journal of Traffic and Transportation Engineering (English Edition)*, 2024.

[53] SM Swapno, SM Nobel, Preeti Meena, VP Meena, Ahmad Taher Azar, Zeeshan Haider, and Mohamed Tounsi. A reinforcement learning approach for reducing traffic congestion using deep q learning. *Scientific Reports*, 14(1):1–20, 2024.

[54] Shupei Wang, Ziyang Wang, Rui Jiang, Feng Zhu, Ruidong Yan, and Ying Shang. A multi-agent reinforcement learning-based longitudinal and lateral control of cavs to improve traffic efficiency in a mandatory lane change scenario. *Transportation Research Part C: Emerging Technologies*, 158:104445, 2024.

[55] Jonas Hamann, Tobias Hagen, and Siavash Saki. Understanding traffic patterns using clustered semantic trajectories and local geographic units. *Transportation Research Procedia*, 82:2911–2930, 2025.

[56] Murat Bakirci. Smart city air quality management through leveraging drones for precision monitoring. *Sustainable Cities and Society*, 106:105390, 2024.

[57] Xian Li, Ziyi Zhao, and Xudong Zeng. Applying bim and pedestrian simulation for architectural flow line optimization in subway stations: an empirical analysis at chongqing ranjiaba station. *Journal of Asian Architecture and Building Engineering*, pages 1–22, 2025.

[58] Zohreh Doborjeh, Nigel Hemmington, Maryam Doborjeh, and Nikola Kasabov. Artificial intelligence: a systematic review of methods and applications in hospitality and tourism. *International Journal of Contemporary Hospitality Management*, 34(3):1154–1176, 2022.

[59] Peter Madzík, Lukáš Falát, Lukáš Copuš, and Marco Valeri. Digital transformation in tourism: bibliometric literature review based on machine learning approach. *European Journal of Innovation Management*, 26(7):177–205, 2023.

[60] Suresh Renukappa, Subashini Suresh, Wala Abdalla, Nisha Shetty, Nagaraju Yabbati, and Rahul Hiremath. Evaluation of smart village strategies and challenges. *Smart and Sustainable Built Environment*, 13(6):1386–1407, 2024.

[61] Pedro Flores-Crespo, Maria Bermudez-Edo, and Jose Luis Garrido. Smart tourism in villages: Challenges and the alpujarra case study. *Procedia Computer Science*, 204:663–670, 2022.

[62] Wenbin Yao, Caijun Chen, Hongyang Su, Nuo Chen, Sheng Jin, and Congcong Bai. Analysis of key commuting routes based on spatiotemporal trip chain. *Journal of Advanced Transportation*, 2022, 2022.

[63] Hu Yang, Bao Guo, Changxin Yan, Zhiqiang Chen, and Pu Wang. A new mobility field and gradient-based traffic signal control approach applicable to large-scale road networks. *Transportation Safety and Environment*, page tdae024, 2024.

[64] Hu Yang, Changxin Yan, Zhiqiang Chen, and Pu Wang. A k-shape clustering based transformer-decoder model for predicting multi-step potentials of urban mobility field. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[65] Yuting Wang, Zhaocheng He, Wangyong Xing, and Chengchuang Lin. Understanding congestion risk and emissions of various travel behavior patterns based on license plate recognition data. *Sustainability*, 17(2):551, 2025.

[66] Jaime G Carbonell, Ryszard S Michalski, and Tom M Mitchell. An overview of machine learning. *Machine learning*, pages 3–23, 1983.

[67] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[68] Gregory F Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9:309–347, 1992.

[69] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

[70] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[71] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[72] Nitin Jindal and Bing Liu. Review spam detection. In *Proceedings of the 16th international conference on World Wide Web*, pages 1189–1190, 2007.

[73] Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007.

[74] Sanford Weisberg. *Applied linear regression*, volume 528. John Wiley & Sons, 2005.

[75] Edward W Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769, 1965.

[76] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[77] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, volume 96, pages 226–231, 1996.

[78] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663):3, 2009.

[79] Tushar Kansal, Suraj Bahuguna, Vishal Singh, and Tanupriya Choudhury. Customer segmentation using k-means clustering. In *2018 international conference on computational techniques, electronics and mechanical systems (CTEMS)*, pages 135–139. IEEE, 2018.

[80] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016.

[81] Mohiuddin Ahmed, Abdun Naser Mahmood, and Md Rafiqul Islam. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55:278–288, 2016.

[82] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[83] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[84] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

[85] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[86] Tony Jebara. *Machine learning: discriminative and generative*, volume 755. Springer Science & Business Media, 2012.

[87] Sotiris B Kotsiantis, Ioannis Zaharakis, P Pintelas, et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.

[88] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[89] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226, 2006.

[90] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[91] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9:293–300, 1999.

[92] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[93] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[94] Peter Hart. The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 14(3):515–516, 1968.

[95] Belur V Dasarathy. Nearest neighbor (nn) norms: Nn pattern classification techniques. *IEEE Computer Society Tutorial*, 1991.

[96] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*. John Wiley & Sons, 2013.

[97] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78, 2004.

[98] Chanyeong Kwak and Alan Clayton-Matthews. Multinomial logistic regression. *Nursing research*, 51(6):404–410, 2002.

[99] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. Seattle, USA, 2001.

[100] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[101] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[102] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[103] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[104] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[105] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[106] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

[107] Larry R Medsker, Lakhmi Jain, et al. Recurrent neural networks. *Design and Applications*, 5(64-67):2, 2001.

[108] Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

[109] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. Clustering algorithms and validity measures. In *Proceedings Thirteenth International Conference on Scientific and Statistical Database Management. SSDBM 2001*, pages 3–22. IEEE, 2001.

[110] Adil Fahad, Najlaa Alshatri, Zahir Tari, Abdullah Alamri, Ibrahim Khalil, Albert Y Zomaya, Sebti Foufou, and Abdelaziz Bouras. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, 2(3):267–279, 2014.

[111] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010.

[112] Geoffrey H Ball. Isodata, a novel method of data analysis and pattern classification. *stanford research institute*, pages AD–699616, 1965.

[113] Robin Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, 16(1):30–34, 1973.

[114] Daniel Defays. An efficient algorithm for a complete link method. *The computer journal*, 20(4):364–366, 1977.

[115] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114, 1996.

[116] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.

[117] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.

[118] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[119] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[120] Wei Wang, Jiong Yang, Richard Muntz, et al. Sting: A statistical information grid approach to spatial data mining. In *VLDB*, volume 97, pages 186–195. Citeseer, 1997.

[121] Gholamhosein Sheikholeslami, Surojit Chatterjee, and Aidong Zhang. Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *VLDB*, volume 98, pages 428–439, 1998.

[122] Anna Forster and Amy L Murphy. Clique: Role-free clustering with q-learning for wireless sensor networks. In *2009 29th IEEE International Conference on Distributed Computing Systems*, pages 441–449. IEEE, 2009.

[123] Ahmed M Serdah and Wesam M Ashour. Clustering large-scale data based on modified affinity propagation algorithm. *Journal of Artificial Intelligence and Soft Computing Research*, 6(1):23–33, 2016.

[124] Hiroaki Shiokawa. Scalable affinity propagation for massive datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9639–9646, 2021.

[125] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[126] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[127] Kamal Berahmand, Mehrnoush Mohammadi, Azadeh Faroughi, and Rojiar Pir Mohammadiani. A novel method of spectral clustering in attributed networks by constructing parameter-free affinity matrix. *Cluster Computing*, 25(2):869–888, 2022.

[128] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. *Advances in neural information processing systems*, 17, 2004.

[129] Anders Krogh. What are artificial neural networks? *Nature biotechnology*, 26(2):195–197, 2008.

[130] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[131] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[132] James H Martin and Daniel Jurafsky. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*, volume 23. Pearson/Prentice Hall Upper Saddle River, 2009.

[133] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[134] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[135] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[136] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[137] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[138] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal*, 32(4):791–813, 2023.

[139] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.

[140] Dhanesh Ramachandram and Graham W Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE signal processing magazine*, 34(6):96–108, 2017.

[141] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[142] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[143] Manfredo Perdigao Do Carmo and J Flaherty Francis. *Riemannian geometry*, volume 2. Springer, 1992.

[144] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[145] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[146] Ch Sanjeev Kumar Dash, Ajit Kumar Behera, Satchidananda Dehuri, and Ashish Ghosh. An outliers detection and elimination framework in classification task of data mining. *Decision Analytics Journal*, 6:100164, 2023.

[147] Xiaoying Chen, Bo Zhang, Ting Wang, Azad Bonni, and Guoyan Zhao. Robust principal component analysis for accurate outlier sample detection in rna-seq data. *BMC bioinformatics*, 21:1–20, 2020.

[148] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[149] John Wilder Tukey et al. *Exploratory data analysis*, volume 2. Springer, 1977.

[150] Peter J Rousseeuw and Mia Hubert. Robust statistics for outlier detection. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 1(1):73–79, 2011.

[151] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[152] Avrim L Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271, 1997.

[153] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[154] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[155] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[156] SGOPAL Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.

[157] Henderi Henderi, Tri Wahyuningsih, and Efana Rahwanto. Comparison of min-max normalization and z-score normalization in the k-nearest neighbor (knn) algorithm to test the accuracy of types of breast cancer. *International Journal of Informatics and Information Systems*, 4(1):13–20, 2021.

[158] Kemal Polat and Umit Sentürk. A novel ml approach to prediction of breast cancer: Combining of mad normalization, kmc based feature weighting and adaboostm1 classifier. In *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–4. Ieee, 2018.

[159] Mohammed Ayub and El-Sayed M El-Alfy. Impact of normalization on bilstm based models for energy disaggregation. In *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, pages 1–6. IEEE, 2020.

[160] Oliver N Keene. The log transformation is special. *Statistics in medicine*, 14(8):811–819, 1995.

[161] Ian T Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.

[162] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[163] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[164] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[165] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An introduction to statistical learning: with applications in r, 2013.

[166] Joaquín Amat Rodrigo. Análisis de componentes principales (principal component analysis, PCA) y t-SNE, 2017. Accessed: 2023-3-29, available under a Attribution 4.0 International (CC BY 4.0).

[167] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[168] Allen Newell. A tutorial on speech understanding systems. In *Speech recognition: Invited papers presented at the 1974 IEEE Symposium*, pages 3–54. Academic Press, 1975.

[169] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[170] Richard Meyes, Melanie Lu, Constantin Waubert de Puiseau, and Tobias Meisen. Ablation studies in artificial neural networks. *arXiv preprint arXiv:1901.08644*, 2019.

[171] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[172] Miroslav Kubat. *An introduction to machine learning*. Springer, 2017.

[173] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[174] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.

[175] Francisco Herrera, Francisco Charte, Antonio J Rivera, María J Del Jesus, Francisco Herrera, Francisco Charte, Antonio J Rivera, and María J del Jesus. *Multilabel classification*. Springer, 2016.

[176] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[177] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu. Understanding of internal clustering validation measures. In *2010 IEEE international conference on data mining*, pages 911–916. IEEE, 2010.

[178] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[179] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[180] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[181] Ana Oliveira-Brochado, Francisco Vitorino Martins, et al. Assessing the number of components in mixture models: a review. *FEP Working Papers*, (194), 2005.

[182] Christian Olivier, F Jouzel, and AE Matouat. Choice of the number of component clusters in mixture models by information criteria. In *Proc. Vision Interface*, pages 74–81, 1999.

[183] Zhengyu Hu. *Initializing the EM algorithm for data clustering and sub-population detection*. PhD thesis, The Ohio State University, 2015.

[184] Jean-Patrick Baudry. *CLADAG 2015. Book of Abstracts*, chapter Estimation and model selection for model-based clustering with the conditional classification likelihood. 2015. ISBN: 978888467749-9.

[185] William S Cleveland. *Visualizing data*. Hobart press, 1993.

[186] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[187] Trupti M Kodinariya, Prashant R Makwana, et al. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.

[188] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

[189] Junpeng Wang, Shixia Liu, and Wei Zhang. Visual analytics for machine learning: A data perspective survey. *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[190] Claus Pahl. Containerization and the paas cloud. *IEEE Cloud Computing*, 2(3):24–31, 2015.

[191] Nicola Dragoni, Ivan Lanese, Stephan Thordal Larsen, Manuel Mazzara, Ruslan Mustafin, and Larisa Safina. Microservices: How to make your application scale. In *Perspectives of System Informatics: 11th International Andrei P. Ershov Informatics Conference, PSI 2017, Moscow, Russia, June 27-29, 2017, Revised Selected Papers 11*, pages 95–104. Springer, 2018.

[192] Ioana Baldini, Paul Castro, Kerry Chang, Perry Cheng, Stephen Fink, Vatche Ishakian, Nick Mitchell, Vinod Muthusamy, Rodric Rabbah, Aleksander Slominski, et al. Serverless computing: Current trends and open problems. *Research advances in cloud computing*, pages 1–20, 2017.

[193] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017.

Part II

PUBLICATIONS

It includes the three publications that support the contributions and results obtained in the thesis.

1

PUBLICATIONS

The following is a list of the publications that are part of this doctoral thesis, which will be presented in the next chapters[1].

Publication [A] adapts smart city data collection techniques to smart villages by designing a sensor infrastructure and creates a dataset that fuses sensor data with contextual information such as holiday calendars, vehicle provenance, and socio-economic factors. In this work, we also study how normalization methods affect ML pipeline performance and extract behavior patterns of visitor and resident vehicles using clustering models. In publication [B], we conduct an ablation study to improve feature selection in an ML pipeline that uses datasets instead of variables, and we applied it for predicting visitor overnight stays. Finally, in publication [C], we present a transformer-based DL model that improves classification in limited-data scenarios such as tracking visitor repeaters in tourism.

[A] Bolaños-Martinez, D., Bermudez-Edo, M., & Garrido, J. L. (2024). Clustering pipeline for vehicle behavior in smart villages. *Information Fusion*, 104, 102164. *[JCR SCIE 2023 - JIF D1; IF 14.8]*. DOI: https://doi.org/10.1016/j.inffus.2023.102164. Number of citations: 10 (Source, Google Scholar), 8 (Source, Altmetric).

[B] Bolaños-Martinez, D., Garrido, J. L., & Bermudez-Edo, M. (2024). Predicting overnights in smart villages: the importance of context information. *International Journal of Machine Learning and Cybernetics*, 1-20. *[JCR SCIE 2023 - JIF Q2; IF 3.1]*. DOI: https://doi.org/10.1007/s13042-024-02337-7. Number of citations: 1 (Source, Google Scholar), 12 (Source, Altmetric).

[C] Bolaños-Martinez, D., Durán-López, A., Garrido, J. L., Delgado-Márquez, B., & Bermudez-Edo, M. (2025). SASD: Self-Attention for Small Datasets—A case study in smart villages. Expert Systems with Applications, 271, 126245. *[JCR SCIE 2023 - JIF Q1; IF 7.5]*. DOI: https://doi.org/10.1016/j.eswa.2024.126245. Number of citations: 0 (Source, Google Scholar), 9 (Source, Altmetric).

1 Displayed citation, download, and indicator counts reflect data up to April 22, 2025.

2

CLUSTERING PIPELINE FOR VEHICLE BEHAVIOR IN SMART VILLAGES

[A] Bolaños-Martinez, D., Bermudez-Edo, M., & Garrido, J. L. (2024). Clustering pipeline for vehicle behavior in smart villages. *Information Fusion*, 104, 102164.

DOI: 10.1016/j.inffus.2023.102164.

- Status: Published.
- Impact Factor (JCR SCIE 2023): 14.8.
- Category: Computer Science, Artificial Intelligence. Rank: 4 / 197 (JIF D1).
- Category: Computer Science, Theory & Methods. Rank: 2 / 144 (JIF D1).
- Number of citations: 10 (Source, Google Scholar).
- Attention score: 8 (Source, Altmetric).
  Mentioned by
  - 6 X users.
  - 2 Redditors.
  - 2 Bluesky users.

  Citations
  - 7 Dimensions.

  Readers on
  - 57 Mendeley.
- Related works: [i], [vii].
- Open source data/software: [ii], [iv], [vi].
- Press notes: [ix], [x], [xiii].

This article is available in **open access** at the following link. Also available in ResearchGate[1], Digibug[2] and Zenodo.

1 https://www.researchgate.net/
2 https://digibug.ugr.es/

Clustering Pipeline for Vehicle Behavior in Smart Villages

Daniel Bolaños-Martinez[a,b,*], Maria Bermudez-Edo[a,b], Jose Luis Garrido[a,b]

[a]*Department of Software Engineering, Computer Science School, University of Granada, C/ Periodista Daniel Saucedo Aranda S/N, 18071 Granada, Spain.*
[b]*Research Centre for Information and Communication Technologies (CITIC-UGR), University of Granada, Granada, Spain.*

Abstract

Smart cities and villages present a plethora of opportunities for fusing and managing multi-source data. However, in the analysis of mobility patterns, the use of only one data source (i.e., road sensors) without considering other contextual data sources, limits the understanding of the process. To address this gap, we propose a pipeline that integrates multiple data sources, providing valuable information for pattern extraction, mainly based on vehicle mobility behavior and provenance. Our research also highlights the critical role of selecting the appropriate normalization algorithm to scale input features from heterogeneous data sources, which has not received sufficient attention in the literature. We conducted our analysis using data from four License Plate Recognition (LPR) cameras, spanning nine months, and incorporating several databases that include provenance, gross income, and holiday information, resulting in a dataset of over 50,000 vehicles. Using this data and our clustering pipeline, we identified various traffic patterns among residents and visitors in a rural touristic area. Our findings assist data analysts in choosing algorithms for analyzing heterogeneous datasets. Moreover, policymakers could use our results to adjust the resources, such as new parking zones.

Keywords: Internet of Things (IoT), sensors, clustering, smart villages, explainability

1. Introduction

Currently, there are 13.4 billion Internet of Things (IoT) devices. Statista predicted that this figure will increase to 29.4 billion by 2030[1]. These devices form an interconnected network that produces extensive data in numerous social domains. Access to a large volume of data collected by various sensors makes it possible to supervise and manage different aspects of society, including evacuation systems, smart environments, and transportation [1, 2, 3, 4]. This trend boosted cities to deploy sensor networks and IoT platforms, for example, to monitor the flow of vehicles on their roads. The data obtained by these sensors have led to numerous studies in several areas, such as traffic behavior [5, 6, 7, 8]. Extracting and combining information from multiple sources, not only sensor data, but also information stored on the Internet, can lead to a better understanding of the problem to be solved. For instance, traffic in cities is partially dependent on local holidays. Some approaches have enhanced the analysis of traffic data (from vehicle counter sensors) with context information to understand the traffic conditions on roads using events data, parking information, or weather conditions [9, 10].

However, most solutions using License Plate Recognition (LPR) sensors [11, 12] did not use additional contextual datasets. Only few works combine LPR with location information [13, 14], but none of them include other con-

*Corresponding author.
Email addresses: danibolanos@ugr.es (Daniel Bolaños-Martinez), mbe@ugr.es (Maria Bermudez-Edo), jgarrido@ugr.es (Jose Luis Garrido)

[1] https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

textual information. They also did not explore calculated variables that enhance the raw data, such as distance traveled or visit frequency. Furthermore, in traffic analysis works [15], and in ML pipelines, in general, [16, 17], the normalization stage was understudied. They usually apply one normalization method without studying the suitability of that method. Moreover, the smart city trend had yet to reach villages, as the solutions found for large cities did not always apply directly to small villages. For example, solutions monitoring traffic behavior in large cities with numerous streets and several traffic lines in some avenues do not extrapolate to villages with mostly pedestrian streets and just one road with a single lane in each direction. Additionally, even if we try to add some explanation to the behavioral cluster in smart villages, the residency of vehicle owners is not straightforward. Recent movements of people relocating from cities to villages or spending extended periods in second residences have made actual residency information unclear in rural areas.

The contribution of this article was twofold. First, we explored the integration of LPR sensor data with contextual information from multiple sources (such as holidays, provenance, or demographic information). One of these sources incorporated data on the origin of each vehicle, which could enhance the results by adding the economic status of the region of origin or the distance traveled to reach the area. Second, we conducted an exploration of different normalization algorithms. To achieve that, we utilized various visualization tools to determine the optimal algorithms based on empirical tests.

In particular, this paper proposes a clustering pipeline based on vehicle behavior in small villages, with information from license plate recognition (LPR) devices and contextual information, such as owners' residence location. We applied the study directly to each individual (vehicle) and defined their spatio-temporal behavior based on their spatial frequencies of visitation. To that end, we fused several datasets and calculated new valuable variables such as the time spent in the area; total distance traveled there, etc. Our pipeline comprised eight steps: data collection, cleaning, fusion, normalization, dimensionality reduction, clustering, evaluation, and visualization. We applied the proposed pipeline to a touristic rural region, with the problems mentioned above of a single small road and the lack of reliable residency information. We paid special attention to optimizing the normalization algorithms to our data. Furthermore, we analyzed the results with residential information and identified the variables that had the most influence on each cluster. With this information, we explained the behavior patterns of each cluster.

Our results are useful for policymakers to improve tourism policy and bring benefits to the area. For example, policymakers could tailor parking fees in the area by identifying visitor clusters and their average stay duration. They could also designate schoolyards or streets for parking during peak tourist periods to reduce road congestion. This work is also useful for developers and data scientists to formalize and choose the clustering and normalization algorithms for their analyses.

The remainder of the paper is organized as follows. In Section 2 related work is summarized. Section 3 presents the theoretical bases discussed throughout the paper, describing the main normalization and clustering algorithms and metrics. Section 4 presents the unsupervised learning pipeline, including a background of the use case, the sensor setup, and the different sources of information used for the construction of the dataset, and Sections 5 and 6 show the analysis and discussion of the results. Finally, Section 7 concludes the paper.

2. Related Work

The concept of information fusion has been applied to the specific problem of tourism flows and smart cities. These approaches used data analysis techniques to combine multiple sources of information, providing valuable insights for developing smart tourism applications in cities and designing sustainable environments. Smart city applications were built on top of data, and data fusion provided a wide variety of techniques to improve the input data for an application [18]. Examples of these techniques included data association, state estimation, unsupervised machine learning, or statistical inference. For example, combining different tourist information was used to predict the tourist flow with graph neural networks [19]. The data used in the solution were composed of tourist infrastructure information, such as camping and tourist housing from OpenStreetMap and the National Statistics Institute (Spanish: Instituto Nacional de Estadistica, INE);

reports released by the Spanish Ministry of Transportation (SMT); and human mobility data, including the number of movements between administrative areas per hour extracted from geotagged Twitter data. Most of these applications were focused either on user recommendations or tourist flow, but little attention was paid to studying the individual behavior of the tourist inside an area (for a detailed survey, see [20, 18]).

The increasing deployment of IoT platforms in smart cities has boosted the proliferation of sensors, including those that monitor traffic. These sensor data allow us to analyze vehicle behavior. The most common works in this area were to analyze mobility patterns in order to improve traffic congestion [5, 7], and to aggregate vehicles to obtain useful conclusions for urban management [21, 6].

To infer mobility patterns from raw data, unsupervised ML has been widely adopted. Clustering analysis was used to detect behavioral patterns in the field of pedestrian-vehicle mobility, and in the field of indoor-outdoor (IO) positioning systems [22]. Algorithms such as GaussianMixture were used to perform segment analysis, where individuals were defined by their movement routines, and the data was related to the frequency and period of stay in different areas. From the movement information provided by smart cards, several papers applied this algorithm to identify market segments based on temporal travel patterns [23], defined tourist patterns based on frequency and areas where transactions were made [24], or identified changes in functional areas of cities over time [25].

Some studies highlighted the importance of employing normalization techniques, such as in the context of time series analysis [26, 27]. In the field of pattern extraction, and specifically in other clustering frameworks, some works use one normalization [15, 17, 16]. However, to the best of our knowledge, no work has studied the influence of using different normalization algorithms.

Few works related to clustering analysis in mobility use LPR cameras as the main source of information [28]. For example, [28] analyzed commuting patterns by constructing the spatio-temporal similarity matrix using the Dynamic Time Warping (DTW) algorithm and subsequently analyzed the characteristics of commuting patterns with the density-based spatial clustering of applications with noise (DBSCAN) algorithm. Similarly, [12] analyzed the change in traffic patterns during the pandemic using K-Means. However, none of these works combined LPR data with vehicle provenance nor studied the touristic behavior of the vehicle.

3. Fundamentals

In clustering pipelines, besides choosing the right algorithm and evaluation metrics, sometimes other analyses are needed. For example, to analyze attributes in different scales, such as nights ranging from 0 to 269 and gross income per capita from 12,638 to 79,327, we had to normalize them first. Sometimes, it is worth reducing the dimensionality to simplify the data matrix and facilitate their understanding by the human mind [29]. The most used dimensionality reduction algorithm is Principal Component Analysis (PCA), and it can be used with at least five variables and five samples [30]. Data distributions come in various shapes (scattered, curved, flat), and understanding this geometry can help choose appropriate clustering algorithms.

3.1. Main clustering algorithms

Unsupervised machine learning automates the knowledge discovery process without needing labeled or previously classified data [18]. Most taxonomies group the algorithms into at least five categories [31], although we have identified seven, as some of them did not fit in the 5 elements taxonomy:

- **Partitional Clustering:** decomposes a dataset into distinct clusters through an iterative process of distance calculations between individuals.
- **Hierarchical Clustering:** constructs clusters in either an agglomerative or divisive manner by adding or removing individuals, respectively.
- **Density-based Clustering:** identifies dense regions of objects in the data space separated by low-density regions.
- **Distribution-based Clustering:** creates clusters based on the probability that each individual belongs to the same distribution.
- **Grid-based Clustering:** divides the space into a finite number of cells.

- **Message-Passing Clustering:** creates clusters by exchanging messages between different data points until convergence.

- **Spectral Clustering:** uses the spectral radius of a similarity matrix of the data in a multidimensional problem.

Table 1 shows the main algorithms in each category described in this section, and examples of applications for each algorithm, in the field of mobility pattern analysis in the last three years (2020-2023).

3.2. Clustering performance

The three most popular internal evaluation metrics in the literature [44] are silhouette coefficient, calinski-harabasz score, and davies-bouldin index. All of these metrics are based on distances between data points and are commonly used to evaluate the effectiveness of virtually any clustering algorithm, working especially well in algorithms that work with distances, such as those included in the hierarchical, partitional, or spectral categories.

These distance-based metrics may not be suitable for algorithms that use the Expectation Maximization (EM) method, such as the GaussianMixture algorithm. This is because the EM method models the data using probability distributions rather than distances between data points. Therefore, we might get some imprecision when comparing the performance of algorithms of this type if we use these metrics. Instead of using distance-based metrics, distribution-based algorithms typically use statistical criteria to determine the optimal number of clusters or components that best fit the data [45]. One of these metrics is the information criterion (IC), which measures how well a statistical model fits the data distribution while penalizing overfitting [46].

$$IC(k) = -2 \cdot L(\hat{\theta}_k) + c_N \cdot k \tag{1}$$

where $\hat{\theta}_k$ is the estimator of the parameter vector relating to the mixture model with order k, L the log-likelihood function, N the number of observations, and c_N an increasing function of N. The optimal number of clusters is the one that minimizes the IC.

The following are two of the best-known variations of information criteria used in the literature [47]:

- **Akaike information criterion** (AIC): AIC is a particular specification of the general information criterion (IC), in which $c_N = 2$. This criterion is known to overestimate the order of the model.

$$AIC(k) = -2 \cdot L(\hat{\theta}_k) + 2 \cdot k \tag{2}$$

- **Bayesian information criterion** (BIC): Tries to overcome the overestimate of AIC. The penalty term depends on the sample size N, so as $N \to \infty$ the penalty is larger and does not overestimate the order of the mixture as much as AIC does [48].

$$BIC(k) = -2 \cdot L(\hat{\theta}_k) + \log N \cdot k \tag{3}$$

3.3. Principal Component Analysis

The Principal Component Analysis (PCA) method condenses the information provided by multiple variables $(X_1, \ldots, X_p)$ from a given sample into a smaller number of variables, finding a number s of underlying factors that explain approximately the same variance as the original variables with $s < p$. Each of the new variables $(Z_1, \ldots, Z_p)$ are called principal components, which are linear combinations of the original variables. We define each Z_i as:

$$Z_i = \Phi_{1i}X_1 + \Phi_{2i}X_2 + \ldots + \Phi_{pi}X_p \tag{4}$$

Each Φ represents the weight or importance that each variable X_i has in each Z_i and, explains the information collected by each of the principal components [49]. It is advisable to apply prior normalization to the data, since this method is highly sensitive to variables of different scales. Furthermore, the PCA only works with numerical data, so it is necessary to perform a previous preprocessing on categorical variables that may exist in the input dataset [50].

3.4. Normalization

Normalization compresses or expands the values of each variable to fit them in the same range of values, normally [0,1], or [-1, 1], making them comparable in subsequent processes (PCA or ML algorithms). The choice of the normalization algorithm usually depends on the specific application and the dataset used, as different methods

Clustering Category	Algorithms	Application	Related work
Partitional	K-Means, MiniBatchKMeans, ISODATA	Target classes, analyze patterns	[12, 15, 32]
Hierarchical	Agglomerative clustering, Divisive clustering, BIRCH	Behavioral patterns, feature extraction	[33, 34, 35]
Density-based	DBSCAN, OPTICS, HDBSCAN, MeanShift	Complexity reduction, anomaly detection	[7, 36, 28, 37]
Distribution-based	Gaussian Mixture	Density estimation, outlier detection	[23, 24, 25]
Grid-based	STING, WaveCluster, CLIQUE	Spatial-based segmentation	Not found
Message passing-based	Affinity Propagation, IWC-KAP, ScaleAP	Clustering indoor location patterns	[38, 39, 40]
Spectral	Spectral Clustering, ASC	Graph partitioning, image segmentation	[41, 42, 43]

Table 1
Examples of works using clustering to infer mobility pattern in 2020-2023.

may yield different results and interpretations. For example, in clustering analysis, normalization can be particularly important for comparing similarities between characteristics based on certain distance measures. Among the most commonly used normalization methods are min-max normalization and z-score standardization [51, 52]. We have also tested two other methods that are commonly used in the literature [53, 54] and occasionally produce better results than min-max or z-score.

1. **Min-max normalization:** Uses the minimum and maximum in the attribute domain to normalize the values to the interval, $[0, 1]$ keeping the distances for each data point X.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5}$$

2. **Z-score standardization:** scales the values so that the mean (μ) of the data domain is 0 and the standard deviation (σ) is equal to 1.

$$X' = \frac{X - \mu}{\sigma} \tag{6}$$

3. **Median Absolute Deviation (MAD) normalization:** normalizes the data such that the median of each attribute is 0 and the median absolute deviation is equal to 1.

$$X' = \frac{X - median(X)}{MAD(X)} \tag{7}$$

Where $median(X)$ is the median of the values in attribute X, and $MAD(X)$ is the median absolute deviation of X.

4. ℓ^2 **normalization:** normalizes the data by dividing it by its Euclidean norm. This ensures that all feature vectors have the same length and is commonly used in machine learning and information retrieval. The formula for ℓ^2 normalization is shown below:

$$X' = \frac{X}{\|X\|_2} \tag{8}$$

Where $\|X\|_2$ is the Euclidean norm of, X given by $\sqrt{\sum_{i=1}^{n} X_i^2}$.

3.5. Dataset geometry

In data analysis, we refer to flat and non-flat geometry as the measurement of distances between points by Euclidean or non-Euclidean geometric methods, respectively. In flat geometry, the distance is measured following a straight line between two points, while in non-flat geometry, the distance is measured following a curve. We can detect whether our data follow flat or non-flat geometry by representing the data in a scatter plot, where each point represents an individual in the population. Visually we can only represent 3 dimensions, which normally are the most representative variables of the cluster, or the firsts principal components of a dimensional reduction algorithm. If the resulting figure shows a roughly circular, rectangular, or elliptical shape, the data are likely to follow a flat geometry. However, if the figure has an irregular, twisted, or folded shape, the data are likely to follow a non-flat geometry. From different studies [55], it has been found that partitional or distribution category clustering algorithms work best with data cases that follow a

flat geometry, while density-based and message-passing algorithms work best with non-flat geometries.[2]

4. Clustering Pipeline

We designed an information fusion pipeline to analyze vehicle behavior that divides the analysis into different stages. In general, the pipeline begins with extracting and collecting data from heterogeneous sources and finally produces a grouping result from a clustering model based on the decisions made along the pipeline (see in Fig. 1). Table 2, describes the different stages proposed in the pipeline and the experimental values considered in each stage. The pipeline consists of the following stages: data collection, data cleaning, data fusion, preprocessing, dimension reduction, clustering, evaluation, and visualization.

4.1. Background

Recent years have seen a growing trend of urban exodus, with many people leaving the cities searching for a quieter life. This trend has been boosted by COVID-19 [56]. With the rise of telecommuting, this trend is likely to continue in the future. These migratory flows include both foreign immigrants and the arrival of resident citizens from other parts of the country [57]. In our use case, we take data from 3 small villages in the Alpujarra, an area close to a national park, and attracting tourists from diverse backgrounds [58]. It is especially favored by local and foreign retirees and "neo-rurals", individuals drawn by environmental concerns or a quieter lifestyle, often becoming residents for extended periods [59]. These groups, referred to as "false residents" [60] or non-registered residents, maintain their vehicle registrations from previous residences. Understanding the patterns of the vehicles in the zone is the first step to generating suitable policies to preserve the area's sustainability.

4.2. Data collection

The main source of information for our work was the vehicle tracking system, particularly the license plate recognition (LPR) cameras. The data were collected by four Hikvision LPR IP devices with Automatic number-plate recognition (ANPR) based on Deep Learning. The devices have a 2MP resolution, 2.8-12 mm varifocal optics, and IR LEDs with a range of 50 m.

To cover the entrances and exits of each village in the target area, we strategically positioned the four cameras, as shown in Fig. 2. The locations were (i) entrance to Pampaneira from the western part of the Alpujarra, (ii) entrance to Pampaneira from the eastern part of the Alpujarra, (iii) entrance to Bubión via a single road, and (iv) entrance to Capileira via a single road. By taking advantage of the road structure, we could monitor the mobility of all vehicles in the area using only four LPRs, minimizing the cost and complexity of the system. The information collected by the cameras was stored on a cloud platform. The rest of the data were collected from different datasets described in Section 4.4.

4.3. Data Cleaning

In the field of the IoT, the production of sensor data can often be inaccurate and lead to the loss of some records. In our case, we presented two cleaning steps for the main dataset (LPR cameras). The first step, "license plate matching", aimed to reduce the error rate of incomplete or wrongly detected license plates by the LPRs. About 2% of the stored 1,050,760 records had missing values in the license plate number. For example, if we had a record with a correct license plate 0000AAA, and another record with the value 0#00AAA, missing the second digit, we could, by probability, infer that both records belong to the same plate number and assign the correct value, 0000AAA, to both records. In our case, we assigned the same plate number to all those records whose license plate matches at least four characters out of seven in the same position. The second step, "route recovery", aimed to reduce the percentage of vehicles not detected by any LPR device. These errors occurred when the camera did not detect a vehicle that passes through the road. This error was difficult to detect, but in our setup, if a vehicle moves on the road from camera 1 to 3, and camera 2 (in the middle of the unique road connecting cameras 1 and 3), did not detect the car, we could infer that the car has passed through camera 2. In our process, if the vehicle was detected in less than 30 minutes in two non-consecutive cameras, our system infers that the vehicle is still in the area and calculates its time of stay based on the new registered values.

[2] https://scikit-learn.org/stable/modules/clustering.html

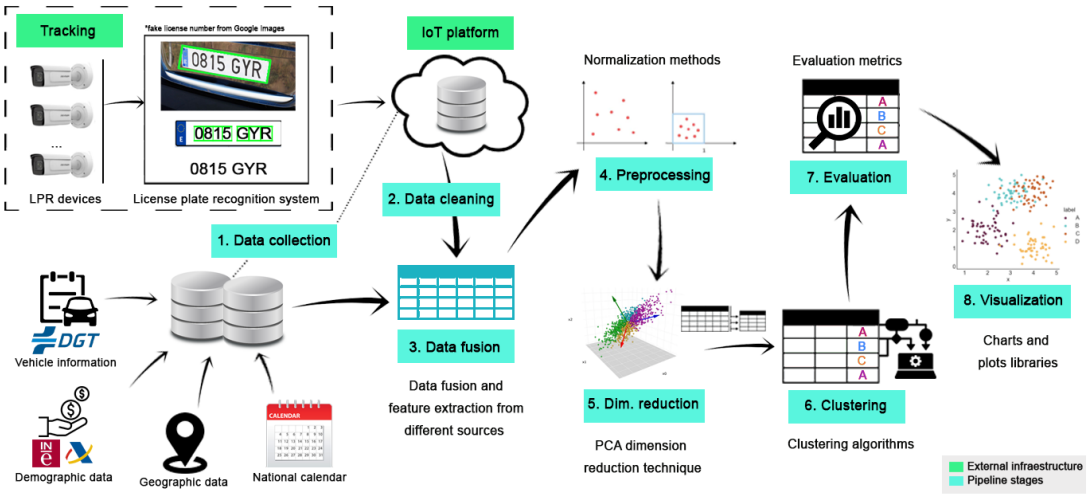


Fig. 1. Overview of the clustering pipeline.

Fig. 2. Setup of the 4 LPR that obtain the data from the license plates of the vehicles.

Stage	Configuration parameters	Experimental values
Data Collection	Data collection from different sources	Storage in own BD and external IoT platform
Data Cleaning	Recovery and treatment of lost data	1. License plate matching 2. Recover movement of vehicles not detected by any camera in their total route
Data Fusion	Fusion of information data and feature extraction	Detailed process in Table 3
Preprocessing	Normalization methods	Min-max normalization, z-score standarization, MAD normalization, ℓ^2 normalizacion
Dimension reduction	Dimension reduction techniques	Principal Component Analysis (PCA)
Clustering	Clustering algorithms	K-Means, MiniBatchKMeans, Agglomerative clustering, BIRCH, DBSCAN, HDBSCAN, MeanShift, Gausian Mixture, Spectral Clustering
Evaluation	Evaluation metrics	Silhouette, Davies–Bouldin, Calinski–Harabasz, number of clusters, Bayesian Information Criterion, Akaike Information Criterion
Visualization	Visualization plots	box plot, scatter-plot, elbow method, PCA variance plot

Table 2
Configuration of each stage of the pipeline with the values used in this study.

4.4. Data Fusion

Combining data from provenance, mobility in the area, and the holiday calendar offered the opportunity to gain an understanding of the region, its inhabitants, and visitors. This section explains each source of information and the feature extraction and construction process of each dataset to allow the merging. We will detail the structure and variables obtained for each data source, creating a joint database. Table 3 schematically shows the information fusion process we followed.

License Plate Recognition Data

The LPRs return information on four variables: the vehicle license plate (license_plate), the time stamp (time_stamp), and a variable (direction) indicated as "IN" when a vehicle enters the village or "OUT" when it exits. Each camera is uniquely identified by its (camera_id). The dataset contains information for nine months (February to October 2022). In total, we have 1,050,760 records, of which 25.69% correspond to the camera PAMPANEIRA 1 (i), 29.25% to PAMPANEIRA 2 (ii), 19.16% to BUBION (iii) and 25.9% to CAPILEIRA (iv) (see in Fig. 2). We grouped the records based on the new vehicle identifier (num_plate_ID), taking into account the mobility behavior of each vehicle. For each vehicle, we built a record per each time the vehicle visits the area, containing the date of entry (entry_time_stamp) and exit (exit_time_stamp) to the area and a list of all the cameras (route) by which it has been registered during its stay, this allows us to calculate the total distance traveled in kilometers (total_distance). This calculation is based on the road distance between each installed device, which we recorded in a small dataset. By summing up the distances between the cameras that a vehicle has passed by, we could determine the distance covered within the area. From the above records, we could also calculate the duration of stay (avg_visit) expressed in days and the number of nights spent there. In case of missing data, i.e., we could not calculate the time of entry or exit of a vehicle in the area, we removed the individual from the dataset.

After that, we performed a grouping at the license plate level so that each row corresponded to a different individual. In this way, we fused the information of all the vehicle visits in the area. Finally, we obtained a dataset with the total number of visits (total_entries), the average time (avg_visit) in days, the complete vehicle routing (route), the total accumulated distance traveled (total_distance), the standard deviation of the average time of each visit (std_visit) in days, the total time spent (total_time) in the area, and the total number of nights spent there (nights). From the new record structure, we could calculate the visits of each vehicle in different weeks (visits_dif_weeks) and months (visits_dif_months) to study the fidelity of the individual in the area. Finally, we obtained

Phase	Tasks	Values
	Calendar Data	
Importing Data	Read the dataset with information on public holidays at national level in Spain	270 days, 3 attributes (date, day_type, holiday_period)
Set holiday periods	Establish the important holiday periods in Spain: Summer Holiday, Christmas and Holy Week	Summer Holiday (from 1 aug. to 31 aug.) Christmas (from 12 dec. to 6 jan.) Holy Week (from 10 apr. to 17 apr.)
Encode variables	Convert categorical holiday periods into binary variables	270 days, 5 attributes (date, day_type, Summer, Christmas, Holy_Week)
	License Plate Recognition Data	
Importing Data	Read the cleaned dataset produced from the detection of vehicle license plates	1,050,760 rows, 4 attributes (license_plate, time_stamp, direction, camera_id)
Calculate associate variables	Calculate variables combining the 4 cameras + LPR location	(license_plate, entry_time_stamp, exit_time_stamp, route, total_distance)
Group information	Group the information for each record by vehicle	50,901 rows, 10 attributes (license_plate, total_entries, avg_visit, std_visit, total_time, nights, route, total_distance, visits_dif_weeks, visits_dif_months)
	Vehicle information Data	
Importing Data	Reads the dataset with vehicle information and its origin	45,132 license plates, 4 attributes (license_plate, postcode, co2_emissions, num_seats)
	Demographic and Economic data	
Importing Data	Reads demographic information about the region of origin of the vehicle	11,752 regions, 4 attributes (postcode, population, gross_income, disposable_income)
Merging Data	Merge the two sources	INE
Validate Data	Validate information common to the two sources	INE
	Geographic data	
Importing Data	Reads information regarding the region of origin of the vehicle	11,752 regions, 7 attributes (postcode, autonomous_community, province, county, district, town, km_to_dest)
Merging Data	Mix and validate information from the two sources used	geopy and pgeocode
Standardize values	Treatment of equivalences between names of regions in different co-official languages	Elimination of accents, spaces and translation to Spanish of all values related to region names
Validate Data	Validate postcodes and geolocation	geopy, pgeocode and INE
	Fusion Dataset	
Merging Data	Unification of header names and data formats, Mix postcode and license plate fields, Delete rows with some null fields	49,224 vehicles, 22 attributes (license_plate, total_entries, avg_visit, std_visit, total_time, nights, route, total_distance, visits_dif_weeks, visits_dif_months, co2_emissions, num_seats, postcode, autonomous_community, province, county, district, town, km_to_dest, population, gross_income, disposable_income)
Generate new variables	Calculate variables related to the type of dates in the calendar during the period of stay of each vehicle	49,224 vehicles, 27 attributes (license_plate, total_entries, avg_visit, std_visit, total_time, nights, route, total_distance, visits_dif_weeks, visits_dif_months, co2_emissions, num_seats, postcode, autonomous_community, province, county, district, town, km_to_dest, population, gross_income, disposable_income, total_holiday, total_workday, entry_in_holiday, total_high_season, total_low_season)
Exporting Data	Obtaining the resultant dataset	CLUSTERING_VEHICLES BD

Table 3
Detailed schematic of the data fusion stage in the pipeline.

a dataset with 50,901 vehicle records and ten attributes.

Vehicle Information Data

The Spanish Directorate-General for Traffic (DGT) provided us with data relating to vehicle information[3] including details such as the vehicle's CO_2 emissions (co2_emissions), the number of seats (num_seats), and the postcode of the vehicle's address (postcode). Each vehicle was associated with a fiscal address used to pay road tax. This generally matched the driver's place of origin, although as described in Section 4.1, this was not entirely true. This dataset helped us understand the distribution of vehicle types and ownership in the different regions. We had a dataset with 45,132 vehicles registered in Spain and four attributes. Unfortunately, we did not have this information for vehicles registered outside of Spain. The percentage of foreigners in the data sample was less than 9.5%. Therefore, we determined these individuals exclusively by their mobility behavior in the area. All information related to vehicle information, demographic, economic, and calendar holidays was restricted to Spanish-registered vehicles.

[3] https://sede.dgt.gob.es/es/vehiculos/informe-de-vehiculo/

Demographic and Economic data

We accessed data regarding population size (population), average gross income (gross_income), and average disposable income (disposable_income) per person for each region linked to a postcode (postcode). This information came from the National Statistics Institute (Spanish: Instituto Nacional de Estadística, INE)[4]. The data were available for regions with more than 1,000 inhabitants and were updated until 2020. The information collected in this database allowed us to understand each region's economic and demographic characteristics, which was valuable for analyzing patterns in the data related to the drivers' economic capacity and willingness to travel. We obtained a database with 11,752 postcode records from Spain and four attributes.

National calendar data

We obtained the holiday data using a holiday library, which also allowed the creation of custom calendars for local holidays, long weekends, and bank holidays. The library was designed to quickly and efficiently generate holiday sets specific to each country and subdivision (such as state or province)[5]. It aimed to determine whether a particular date was a public holiday and to set national and regional holidays for multiple countries. As we mentioned before, due to the small percentage of foreign individuals in the sample and the complexity of dealing with a different set of holidays for different vehicles, we restricted the analysis of the holidays to Spain. However, we included Saturdays and Sundays in the holidays, so we also considered the idea of a weekly holiday for any origin. For each day, represented by a date (date), we specified with a binary variable whether it is a holiday or working day (day_type). In addition, holiday periods were defined to establish high and low tourist seasons based on the three most important national holidays in Spain: Summer, Christmas, and Holy Week[6], which represented a binary variable, indicating whether the date belonged to that holiday period (Summer, Christmas, Holy Week). We obtained a database with 270 days and five attributes.

Geographic data

We obtained the geographic origin of the vehicles using the postcode and two libraries: pgeocode and geopy. pgeocode[7] allowed fast and efficient queries of GPS coordinates, region name, and municipality name from postcodes. geopy[8] is a Python client that provided access to several popular geocoding web services. We used data from both sources to validate and complement each other's vehicle location information at different levels, such as municipality, county, or suburb. Furthermore, we also used data from the INE[9] to verify the province and autonomous community code of the vehicle, which was directly related to the postcode. Hence, we created a database that contained, for each postcode, information about (autonomous_community), (province), (county), (district), (town), and the distance in kilometers between the origin of the vehicle and the destination region (km_to_dest). We obtained a database with 11,752 postal code records and nine attributes.

Merge of all the processed datasets

Finally, we fused all constructed databases, crossing the information from the license plate and postcode variables. After merging the tables, we eliminated records with any of the aforementioned attributes null. The information from the national calendar allowed us to add to the vehicle database information related to the stay and its total number of holidays (total_holiday), workdays (total_workday), high season (total_high_season), low season (total_high_season) and a binary variable indicating whether the vehicle enters the area on a holiday or a workday (entry_in_holiday). The resulting dataset contains information on the behavior in the area for 49,224 vehicles and 27 attributes.

4.5. Preprocessing

Our dataset contains 27 attributes with different scales and units. Hence, some variables may be more influential than others in our analysis. To solve this problem, we will apply normalization to the data. Normalization must be applied to numerical data, so we

[4] https://www.ine.es/dynt3/inebase/es/index.htm?padre=7132&capsel=5693

[5] https://python-holidays.readthedocs.io/en/latest/

[6] https://es.statista.com/temas/3585/vacaciones-en-espana/#topicOverview

[7] https://pgeocode.readthedocs.io/en/latest/

[8] https://geopy.readthedocs.io/en/latest/

[9] https://www.ine.es/daco/daco42/codmun/cod_ccaa_provincia.htm

must first convert the categorical variables (in our use case: route, postcode, autonomous_community, province, county, district, town) to numerical values. In particular, the numeric variable, total_distance, kept the information of the kilometers traveled in the variable route. The rest of the categorical variables related to the provenance: town, postal code, etc., and we converted them into the variable km_to_dest. We removed the variables co2_emissions and num_seats, because they had a high percentage of missing values (about 25%), which could introduce noise. During this phase, we also excluded vehicles with a total stay time (total_time) of less than 1 hour. This subset comprised 16.98% (8360 vehicles) of the entire dataset. Given their role as transient passers-by in the area and their brief stays, which did not contribute to any discernible benefits for the locality, we omitted them from our analysis. We finally obtained a dataset with 40,864 vehicles and 17 numerical attributes: total_entries, avg_visit, std_visit, total_time, nights, total_distance, visits_dif_weeks, visits_dif_months, km_to_dest, population, gross_income, disposable_income, total_holiday, total_workday, entry_in_holiday, total_high_season, total_low_season.

4.6. Dimensionality reduction

We reduced the dataset's dimensionality to improve efficiency in clustering. This involved simplifying the feature matrix by removing low-variance features that would not contribute much to our goal of clustering different vehicle behaviors. We used PCA to reduce dimensionality. We found that removing variables with very high correlation substantially improved the results and the performance of the clustering models for our data. Furthermore, correlated variables increased the data's variance, making the visual interpretation of the PCA results difficult, as the first principal components might not have accurately reflected the underlying structure of the data.

4.7. Clustering and evaluation

Our study explored all the algorithms mentioned in Section 3.1 to determine the optimal approach for pattern recognition and evaluated whether they could find a realistic solution.

4.8. Visualization

Data visualization was essential in our work, as it helped to determine and make decisions about parameter settings, algorithms, and normalization methods. It also made our machine learning results more understandable. For instance, we used the elbow method to find the best number of clusters for various algorithms. This method plots the number of clusters and a given evaluation metric. The number of clusters at the curve's bend ("elbow") balances the model's complexity and accuracy. We used scatter plots to visualize the first two principal components for each normalization method, helping us grasp the data's structure and cluster distribution. Box plots were another tool we used to show how features were distributed within clusters. This allowed us to spot common patterns in each cluster.

4.9. Data Privacy and Security

The LPR cameras sent the license plates to a secured server on our provider's premises. We only used the anonymized dataset (see Section 4.4), which we openly published[10]. The other datasets were public, except the DGT dataset. The DGT shared with us sensitive data with license plates and its associate owner's postal code only for research purposes. This information was stored encrypted and was accessible only to authorized researchers. Furthermore, we used clustering, which means that we did not evaluate the individual behavior of each person but considered them part of a group. Hence, the privacy of the activities of the individuals is not compromised.

5. Results

To model traffic behavior and distinguish between residents and visitors. We labeled vehicles as 1 for those registered in the area (resident) and 0 for others. We identified several variables with non-significant correlations (correlation < 0.2): avg_visit, std_visit, and population, and removed them. We showed the results relating to these analyses in Appendix A (Fig. A.1 and Table A.1).

[10] https://zenodo.org/record/8356386

5.1. *Preprocessing and Dimension reduction results: Normalization selection*

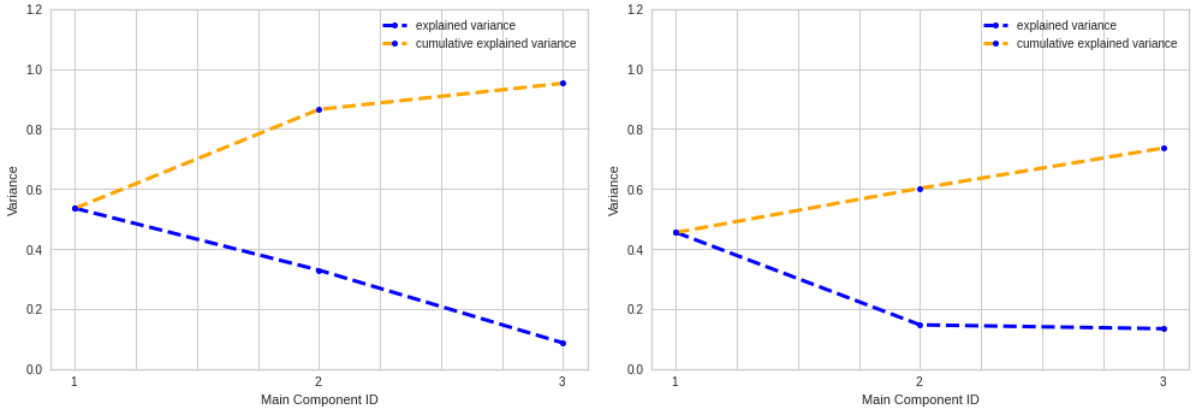
We performed preprocessing and dimension reduction stages together because they are interdependent. We found that removing highly correlated variables before applying PCA improved the variance explained and the scatter plots of PCA components. Specifically, we removed variables with a correlation coefficient > 0.9: total_entries, nights, visit_dif_weeks, visit_dif_months, km_to_POQ, gross_income, entry_in_holiday, total_distance and total_high_season Appendix A (Fig. A.2).

After applying the four most common normalizations to the data (see in Section 3), we applied PCA analysis. Fig. 3 showed the variance carried by each PCA component for each normalization. We could appreciate that two components explained most of the variance in all normalizations. Hence, we performed an exploratory visual analysis by plotting the first two principal components to study their underlying geometry. In Fig. 4, we overlaid on the plots, in red, the points representing the vehicles of the registered residents, in blue, non-registered residents.

The normalization method that obtained the highest cumulative variance was ℓ^2, indicating that it retained the most information in only two components (see in Fig. 3 (d)). In addition, the variance of each dimension was high compared to the other techniques analyzed, suggesting that the data were well distributed in both dimensions. The graph in Fig. 4 (d) shows a clear separation between the two groups, and the registered residents (in red) were well confined. The min-max normalization method obtained the second-best cumulative variance and the highest variance for each dimension, preserving a reasonable amount of information in only two components (see in Fig. 3 (a)). The graph also shows a clear separation between the two groups, and the actual residents were defined along a vertical line on the left cluster in Fig. 4 (a). In contrast, the MAD normalization method had a lower cumulative variance and variance for each dimension (see in Fig. 3 (c)) than the ℓ^2 and min-max normalization methods. The 2-dimensional scatter plot showed no apparent clusters (see in Fig. 4 (c)), and the actual residents were highly dispersed, which made it unusable for our analysis. We had similar results in a scatter plot of three principal components. Finally, the mean normalization, z-score, method presented the lowest cumulative variance, indicating that it lost more information during dimensionality reduction than other techniques (see in Fig. 3 (b)). The graph shows that the actual residents were grouped together, but for the 2-components, there were no apparent significant clusters (see in Fig. 4 (c)). The trend of the cumulative variance explained was rising, suggesting that the current normalization method could be enhanced by including more components. By adding more dimensions, it may be possible to identify a dimension where the group of registered residents conformed to a clearer distribution. PCA typically worked better with z-score standardization than with min-max normalization. However, normalization techniques that better handled outliers (such as z-score) may not always have been effective for all datasets because they tried to distribute the individuals uniformly, softening the outliers. For example, we observed that the min-max normalization method performed better than the z-score standardization, possibly due to the presence of small clusters that z-score detected as outliers. In particular, the dataset have a low proportion of registered residents (less than 2% of the total sample), which could be considered outliers (see in Table A.1). In these cases, the min-max normalization method, which was more sensitive to small clusters, may have given better results. With all this information, we decided to apply the two best normalizations for our data (ℓ^2 and min-max) and compare the results obtained in the clustering.

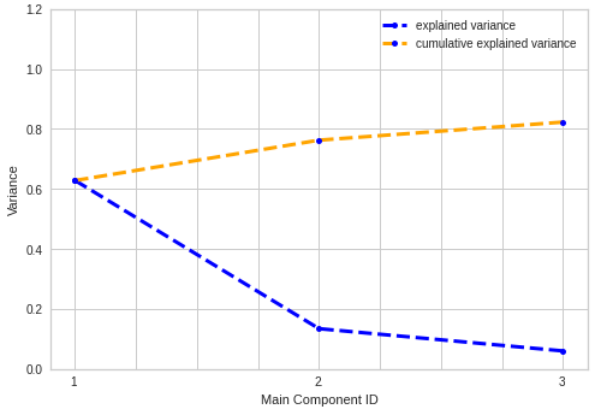
5.2. *Exploration of Clustering Algorithm categories*

From the scatter plots in Fig. 4, we observed that the data points were spread relatively flat. This suggested that the data points were concentrated in a lower-dimensional space within the original feature space. In other words, the data appeared to exist in a more compressed space rather than being spread out across multiple dimensions. Hence, partition and distribution-based clustering models were the most suitable for this geometry (see Section 3.5). We tested various algorithms from other categories to verify this. However, we did not report the results because none of the tested techniques identified a cluster for the correctly registered residents. For example, density and spectral-based algorithms performed poorly, probably because of the non-flat geometry but also because they worked best for detecting outliers. Hierarchical algorithms performed poorly, probably because of

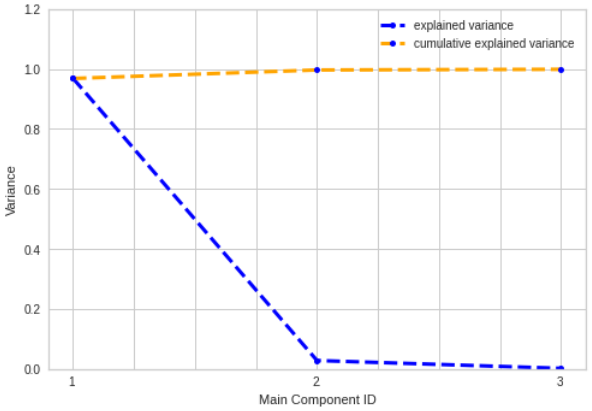


(a) Min-max normalization.

(b) Z-score standarization.

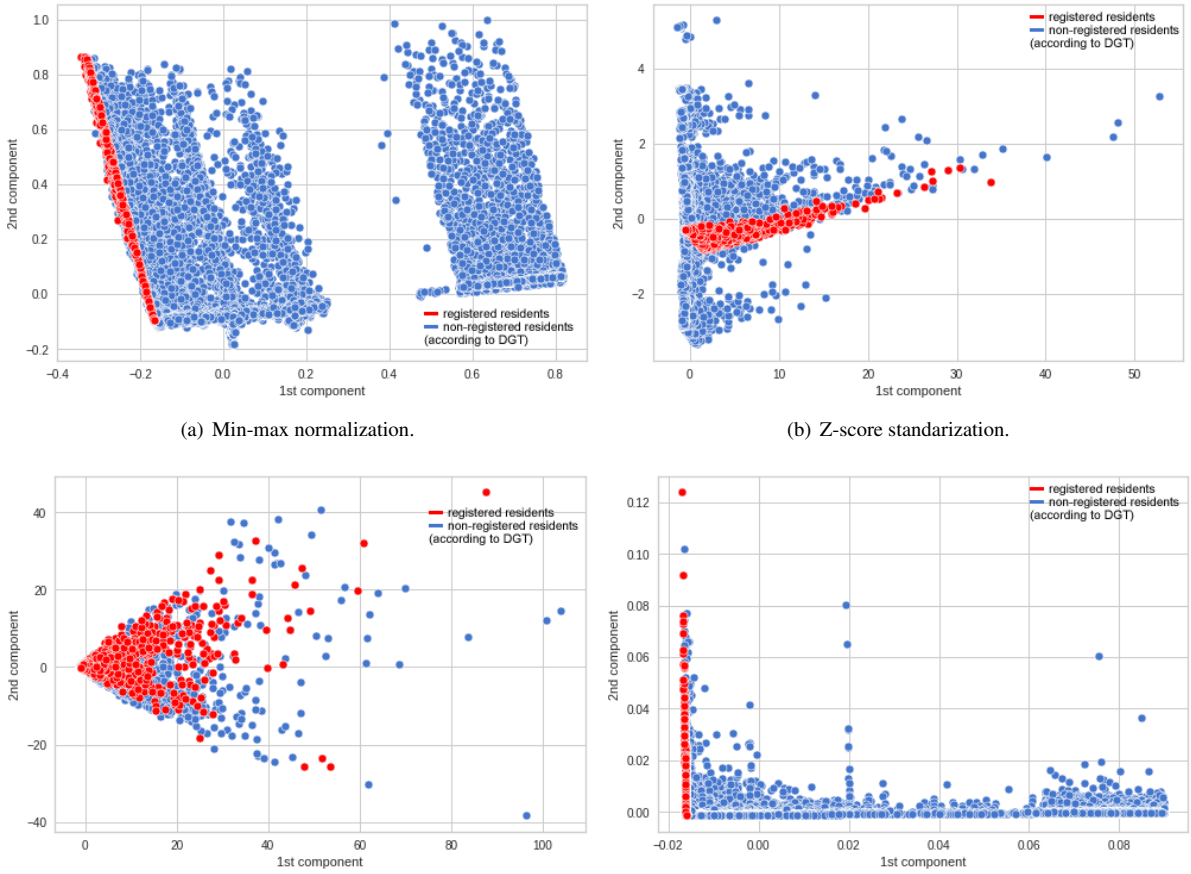


(c) MAD normalization.



(d) ℓ^2 normalization.

Fig. 3. Variance with 3 principal components.



(a) Min-max normalization.

(b) Z-score standarization.

(c) MAD normalization.

(d) ℓ^2 normalization.

Fig. 4. Scatter-plot of the first two principal components for the different normalizations.

the non-flat geometry, but they also had difficulties with highly concentrated datasets, creating distinct groups only when the separation was obvious. Consequently, we focused on the partitional and distribution-based algorithms, which worked well with flat geometry data. In particular, we tried Gaussian Mixture, K-Means, and MiniBatchK-Means.

Gaussian Mixture models were more flexible and could handle different cluster shapes and sizes, while K-Means assumed a spherical shape of the clusters and a uniform size. In addition, Gaussian Mixture models could estimate the probability that a data point belonged to a cluster, which could be useful in specific applications where we needed to make decisions based on uncertain data or when we wanted to assign a data point to multiple clusters with different probabilities. In the tests carried out, we discovered that K-Means and MiniBatchKMeans were not able to find any cluster that contained the majority of individuals of registered residents (see in Fig. 4 (a) and (d)). This was because the distribution of these individuals followed an elliptical geometry, which was not amenable to partition-based algorithms directly. Based on these results, we used the Gaussian Mixture clustering algorithm given the geometry of our data and the distribution followed by registered residents.

5.3. Evaluation results

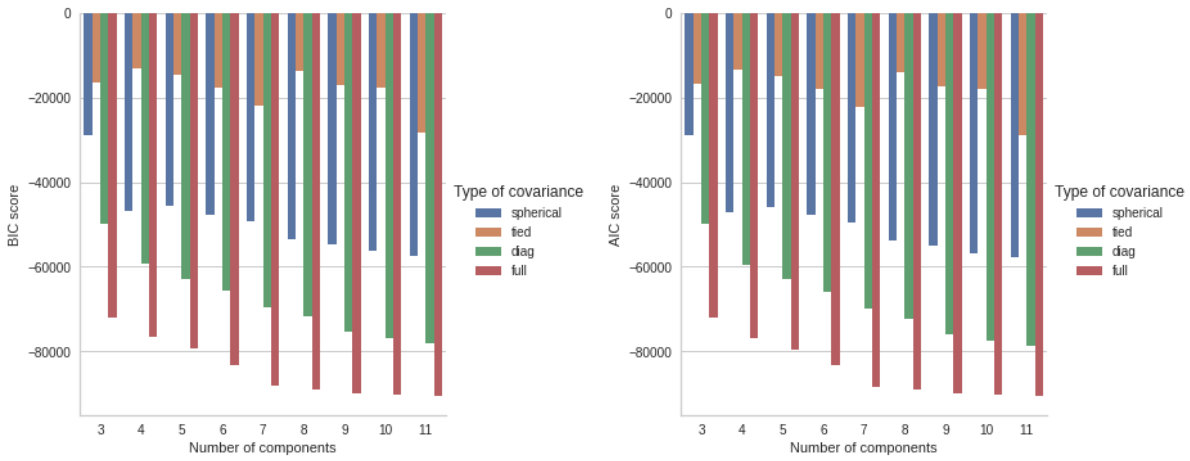
After choosing the algorithm, we had to configure its settings and hyperparameters. For the GaussianMixture algorithm, a 'mixture' meant a blend of multiple Gaussian distributions, with each component representing one of these distributions [61]. We could adjust the number of mixture components, determining how many Gaussian distributions to use for modeling the data. Another configurable aspect was the covariance type, which influenced how variables in the data were correlated, impacting the model's accuracy and efficiency. The common types of covariance were:

- Full: all components have their own covariance matrix. This means that each component can have a complex correlation structure between the different variables.
- Tied: all components share the same overall covariance matrix. This can be useful if different variables are highly correlated.
- Diagonal: each component has its own diagonal in the covariance matrix. This means that the correlation structure between the different variables is limited to correlations between pairs of variables.
- Spherical: each mixture component has its own unique variance. This means that the correlation structure between the different variables is limited to the variance of each variable individually.

To select the best hyperparameters, we calculated the performance of the resulting model with the metrics presented in Section 3.2, which were appropriate for clustering algorithms based on distributions (BIC and AIC). In the next subsections, we performed the evaluation for the different types of covariance of the GaussianMixture algorithm on the two normalizations chosen in the previous subsection: min-max and ℓ^2 normalization.

5.3.1. Evaluation results: Min-max normalization

Fig. 5 represents the values of the BIC and AIC metrics with respect to the number of components and type of covariance used as parameters of the GaussianMixture algorithm. We noted that the 'full' covariance type was the one that minimized both metrics in all cases, so it was the one chosen for the subsequent analysis. This value meant that each component had its own overall covariance matrix, which meant it could capture any correlation between variables. We noted no significant differences between the values obtained for AIC and BIC scores. Therefore, we calculated the elbow method on the BIC score to select the optimal number of mixture components. In Fig. 6, we could detect two "elbow" points. One occurred at seven components (-87,585 BIC), marking a 4591 unit difference from the preceding six components (-82,994 BIC) and a 1632 unit difference from the following eight components (-89,217 BIC). The other point was at four components (-76,798 BIC), with a 4407 unit difference from the preceding three components (-72,391 BIC) and a 3098 unit difference from the subsequent five components (-79,896 BIC). The change from seven components to their previous value was more substantial than the change from three to four, and the difference with the following eight components was less pronounced, indicating a more abrupt change in slope.



(a) BIC score. (b) AIC score.

Fig. 5. Information criteria for the GaussianMixture on min-max normalization.

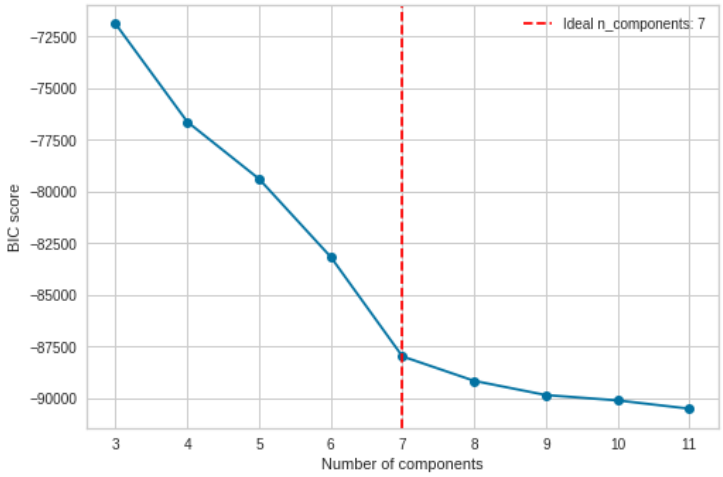


Fig. 6. Elbow method for BIC using min-max normalization.

5.3.2. Evaluation results: ℓ^2 normalization

Fig. 7 represents the values of the BIC and AIC metrics with respect to the number of components and type of covariance, used as parameters of the GaussianMixture algorithm for ℓ^2 normalization. We observed that the 'tied' covariance type was slightly superior for three components, but the 'full' covariance type was again the best for more than three components. Similarly to the min-max normalization, there was no significant difference between the AIC and BIC score values. Therefore, we calculated the elbow method on the BIC score and the 'full' covariance type. Fig. 7 shows a clear change in four components, showing an increase of 36,977 units in the BIC score (the highest in the graph), going from three components (-232,851 BIC) to four components (-269,828 BIC).

5.4. Visualization results

Once we selected the clustering algorithm and the hyperparameters, we discussed the visualization of the generated clusters over the two chosen normalizations: min-max normalization and ℓ^2.

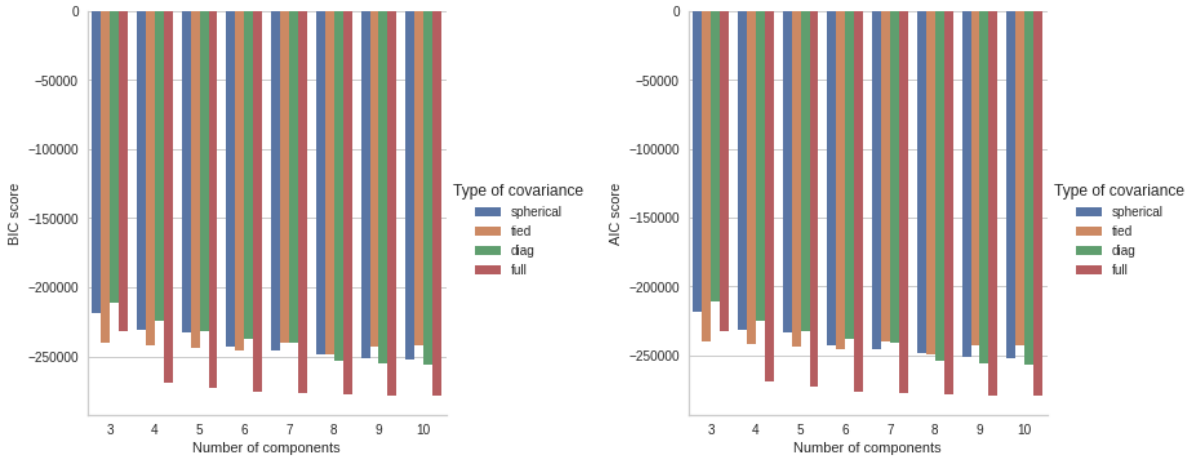
5.4.1. Visualization: Min-max normalization

Fig. 9 (a) shows a 2D scatter plot, where each axis represented 1st and 2nd principal components. Fig. 9 (b) highlights registered residents in red. Fig. 9 (c) displays a 3D scatter plot with 3 principal components in each axis. Table 4 shows vehicle percentages and registered resident counts in 7 clusters. Cluster 3 correctly grouped over 96% of individuals, and cluster 5 contained nearly 45% of the total sample. Cluster 3, with the most registered residents, represented around 14% of the total population.

Fig. 10 presents the box plots for the 7 clusters for the nights (Fig. 10 (a)) and km_to_dest (Fig. 10 (b)) variables, which showed significant differences in explaining the groups. Figs. 11 and 12 presents the box plots of the most relevant variables for the 7 clusters obtained. Table 5 complements Figs. 11 and 12, indicating the exact number of the mean of each variable in each cluster. To facilitate visualization, we separated some of the box plots according to the value of the variable nights, which seemed to discriminate well between 2 groups of clusters: (0, 1, 2, 5) with lower values and (3, 4, 6) with higher values (see in Fig. 10 (a)). Clusters 3,4,6 had a number of nights close to the behavior of a resident in the area and represented 27.44% of the data (see Table 4). Clusters 0,1,2,5 had visitor behavior because they spent fewer nights in the area and represented 72.56% of the total sample.

For clusters 3, 4, and 6, a key factor was the distance in kilometers from the vehicle's registered address to the area (see Fig. 11 (c)). Despite significant differences in origin, these three clusters exhibited similar patterns in terms of nights spent, indicating that they resided or stayed in the area. Cluster 3, with an average distance of 19.39 km (see Table 5), primarily consisted of vehicles registered in the study area (registered residents) and nearby villages. Cluster 6, with an average distance of 1747.30 km for the variable km_to_dest, comprised non-registered residents from abroad, as defined in Section 4.1. Cluster 4, with an average distance of 318.36 km, represented individuals from other regions of Spain who were also non-registered residents, as discussed in the same section. Additionally, the gross income variable was significantly higher in cluster 4 compared to clusters 3 and 6 (almost 34% higher) (see Fig. 12 (c)). This suggests that a majority of individuals in cluster 4 (non-registered residents from other Spanish regions) came from regions with above-average incomes. Residents living farther away (clusters 4 and 6) had lower average values for total_distance, total_high_season, and total_entries (see Table 5). This is because they tended to visit less often, cover shorter distances in the area, and have fewer visits during the high season compared to residents in closer proximity (cluster 3) (see Fig. 11 (e) and Fig. 12 (a, e)).

Clusters 0, 1, 2, and 5 represented different visitor behaviors (see in Table 5). Cluster 0, with an average distance of 128.55 km, corresponded to visitors from the province, typically staying 1.57 nights. They made an average of 1.54 visits, mostly during weekends and holidays, and around 65% of these visits occurred in high season (see in Fig. 12 (b, f)). Cluster 1, averaging 1742.97 km, consisted of foreign visitors who stayed for only 0.26 nights. They tended to visit during low seasons, primarily using the main road to reach the first village in the area and not visiting the other villages. Cluster 2, with an average distance of 474.21 km, attracted visitors from outside the province, spending around 1.55 nights. This cluster had the highest average gross income (see in Fig. 12 (d)) and visits the area during high season, likely by tourists from northern Spain. Cluster 5, averaging 253.70 km, represented visitors from other nearby provinces. They



(a) BIC score. (b) AIC score.

Fig. 7. Information criteria for the GaussianMixture on ℓ^2 normalization.

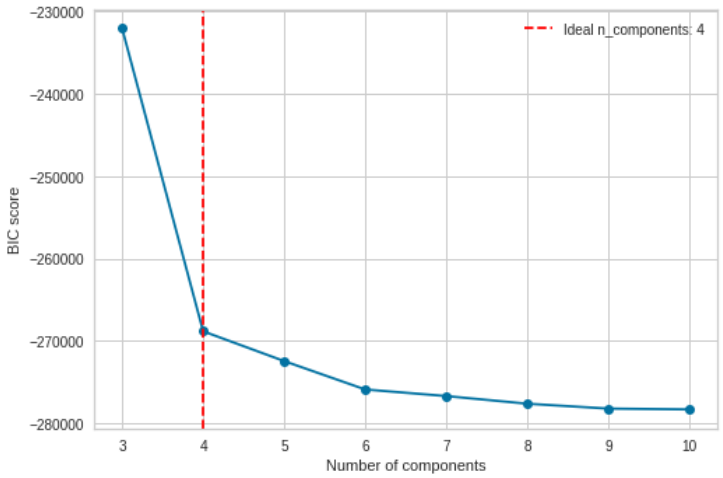
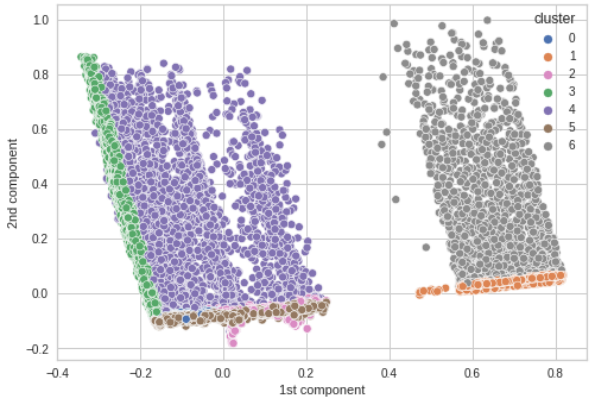
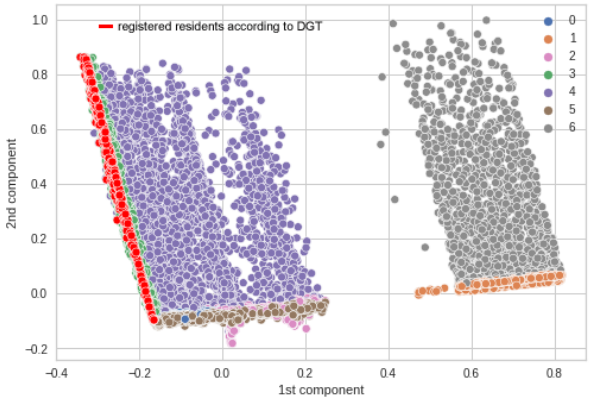


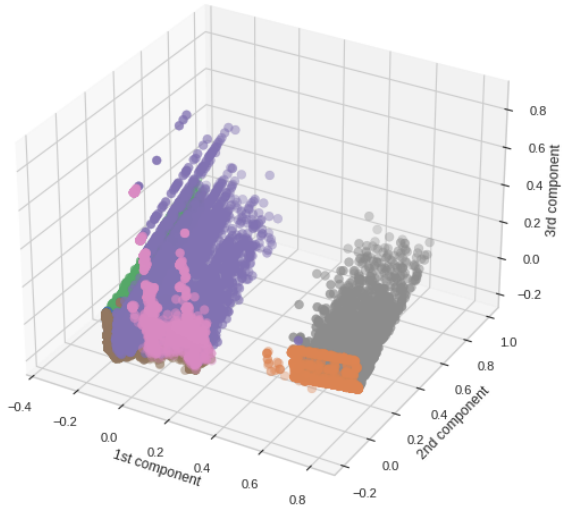
Fig. 8. Elbow method for BIC using ℓ^2 normalization.



(a) Segmentation for 7 mixture components.



(b) Highlighted registered residents.

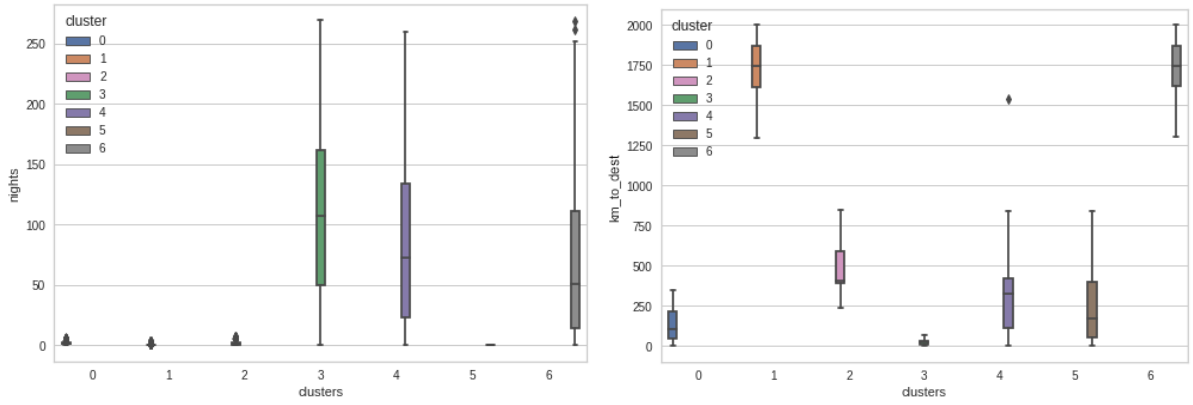


(c) 3D plot with 3rd component.

Fig. 9. Scatter-plot of the first three components (PCA) using min-max normalization.

Data points	Nº cluster						
	0	1	2	3	4	5	6
Percentage of sample	14.13%	5.74%	8.06%	13.47%	10.30%	44.63%	3.67%
Real Residents	8	0	0	641	3	9	0
Rest of individuals	5766	2347	3293	4862	4205	18,230	1500

Table 4
Clusters based on registered resident labels using min-max normalization.



(a) Total nights of stay by cluster.

(b) Distance in Km. to the area by cluster.

Fig. 10. box plots for min-max normalization (I).

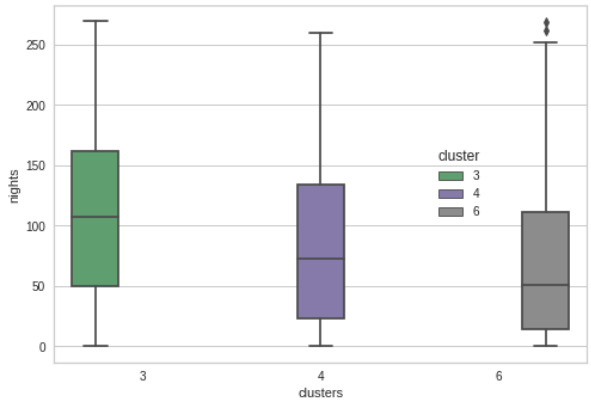
rarely stayed overnight (0 nights on average) and predominantly visited during the day, making up 44.63% of the sample (see Table 4). Only 27% of their visits occurred during high season (see Fig. 12 (f)), suggesting day trips from neighboring provinces.

5.4.2. Visualization: ℓ^2 normalization

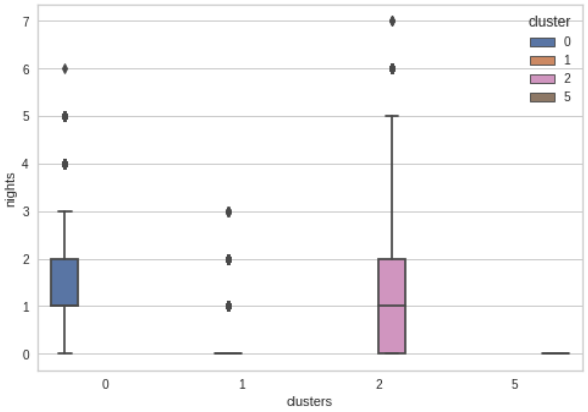
Fig. 13 shows the data distribution using ℓ^2 normalization. Fig. 13 (a) depicts a 2D scatter plot of principal components (1st and 2nd axes). In Fig. 13 (b), registered residents are marked in red, and Fig. 13 (c) presents a 3D scatter plot. Table 6 shows cluster details: Cluster 0 accurately includes over 89% of registered residents, representing 10.30% of the total population. Cluster 3 contains 75.95% of the total sample.

Fig. 14 shows the box plots of the relevant variables for the 4 clusters, and Table 7 displays the mean of each of these variables in each cluster. We distinguished two clusters that contained a high value of the variable "nights" (cluster 0 and 2), while the rest of the clusters (clusters 1 and 3) had a low value. Although there were outliers (see in Fig. 14 (a)) that increased the mean number of nights for these clusters (clusters 1 and 3), 50% of the individuals had a number of nights lower than 2 for cluster 3 and lower than 15 nights for cluster 1.

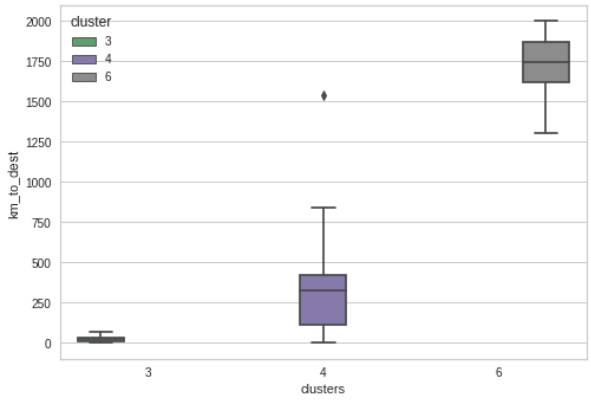
Cluster 0, which included over 89% of area residents, had an average stay of 144.93 nights, covering an average distance of 25.54 km. Most non-registered residents in this cluster were from the province (see Fig. 14 (b)). Cluster 2 represented non-registered residents from outside Granada, making up only 5.25% of the total sample. They stayed an average of 84.62 nights and came from an average distance of 598.01 km. For both groups, total_distance, total_high_season, and total_entries (see Table 7) were inversely proportional to km_to_dest, indicating that visitors from further away tended to visit during the low season, move less within the area, and visit fewer times a year (see Fig. 14 (c, d, f)). Cluster 1 comprised foreign visitors and some non-registered foreign residents, covering an average distance of 1750.68 km. They stayed an average of 22.81 nights, with only 17% of stays in the high season. Cluster 3, the largest group



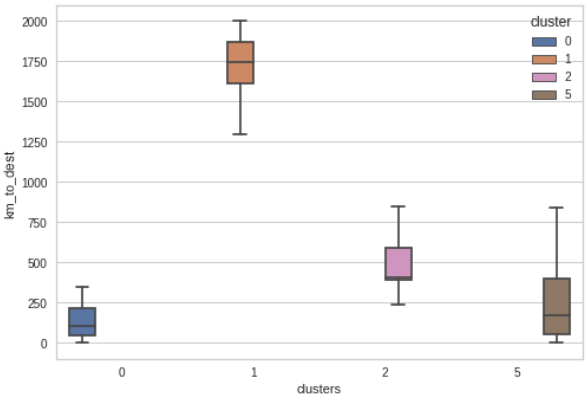
(a) Nights of stay for clusters **3,4,6**.



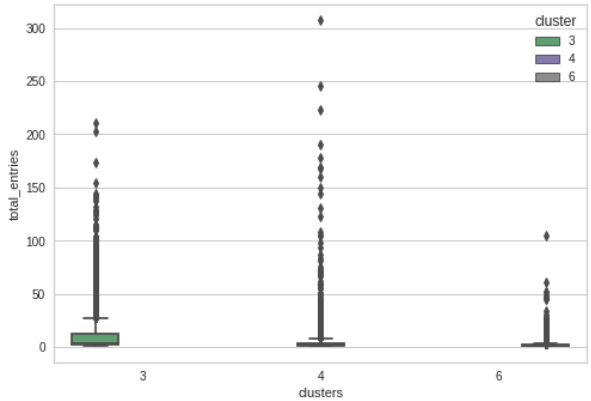
(b) Nights of stay for clusters **0,1,2,5**.



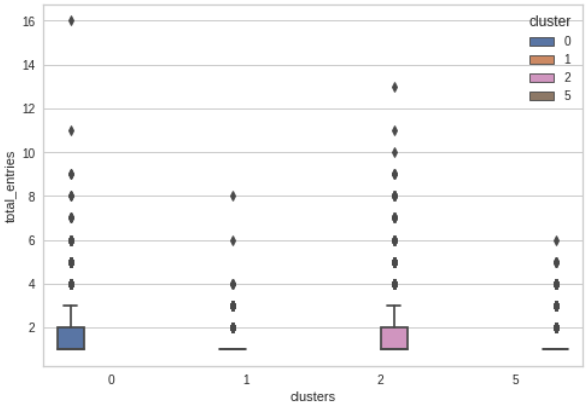
(c) Distance to the area for clusters **3,4,6**.



(d) Distance to the area for clusters **0,1,2,5**.

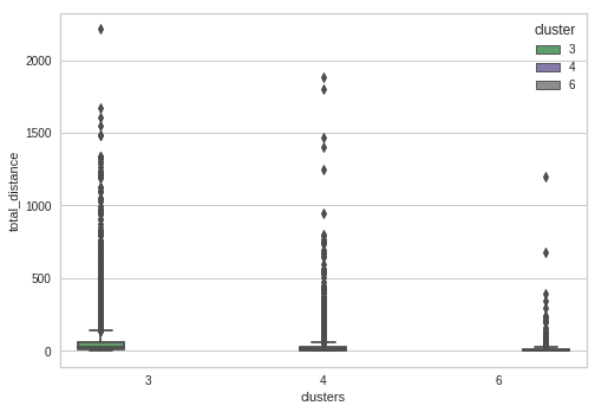


(e) Total entries for clusters **3,4,6**.

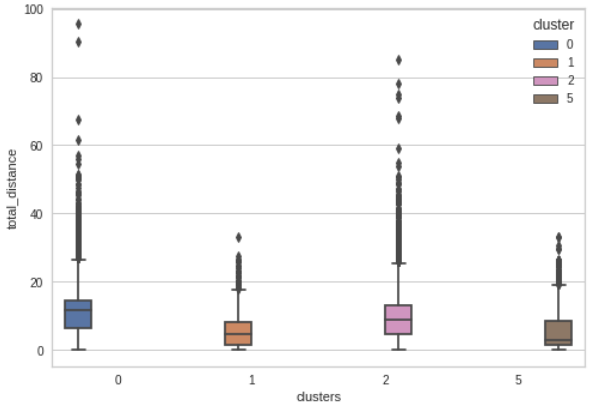


(f) Total entries for clusters **0,1,2,5**.

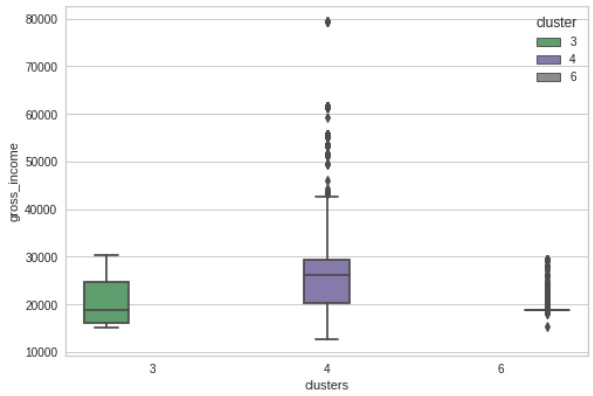
Fig. 11. Box plots for min-max normalization (II).



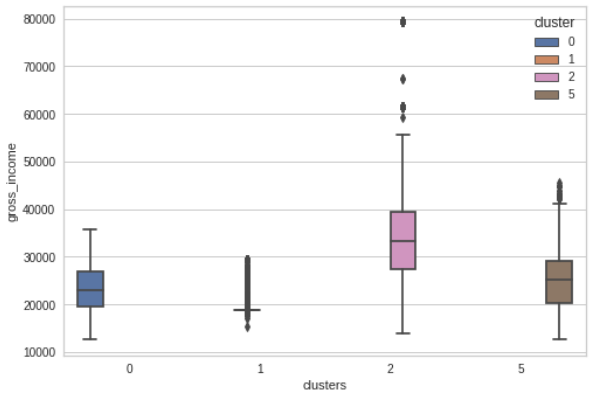
(a) Distance run in area for clusters **3,4,6**.



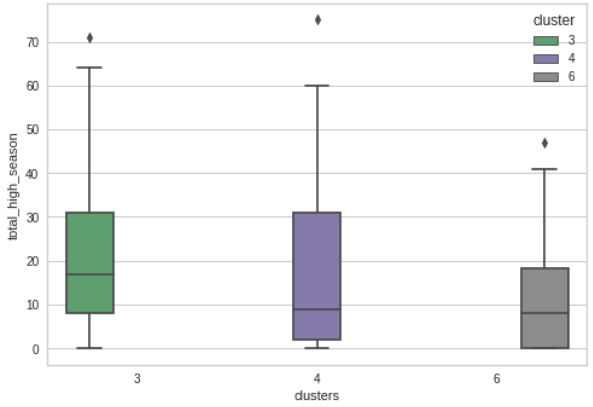
(b) Distance run in area for clusters **0,1,2,5**.



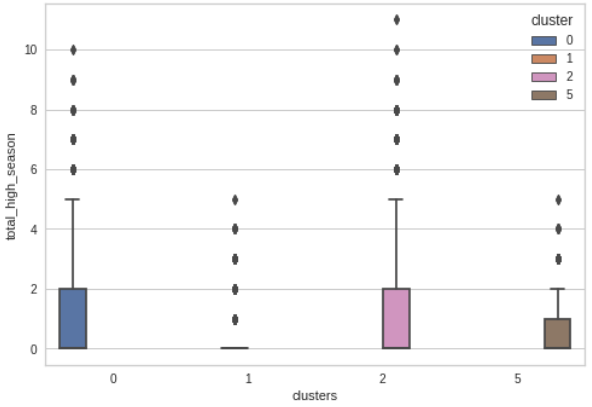
(c) Avg. gross income for clusters **3,4,6**.



(d) Avg. gross income for clusters **0,1,2,5**.

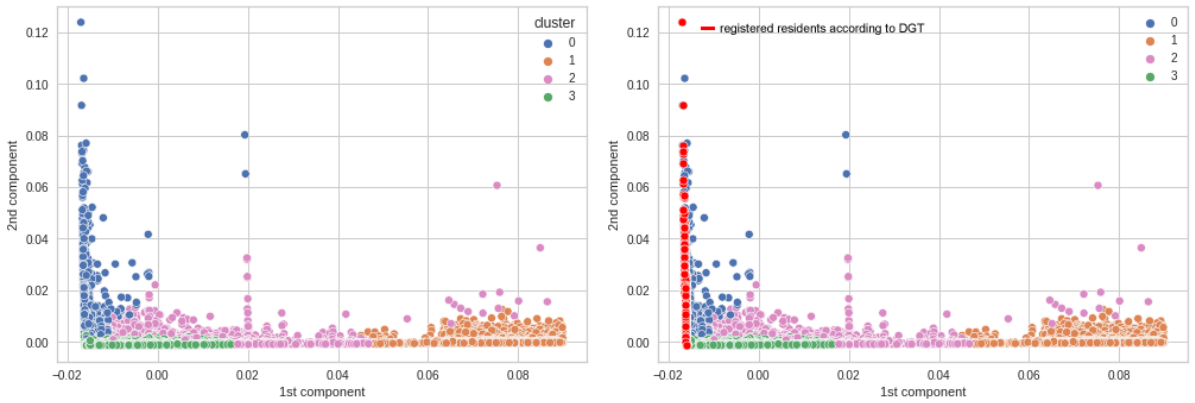


(e) Total high season for clusters **3,4,6**.



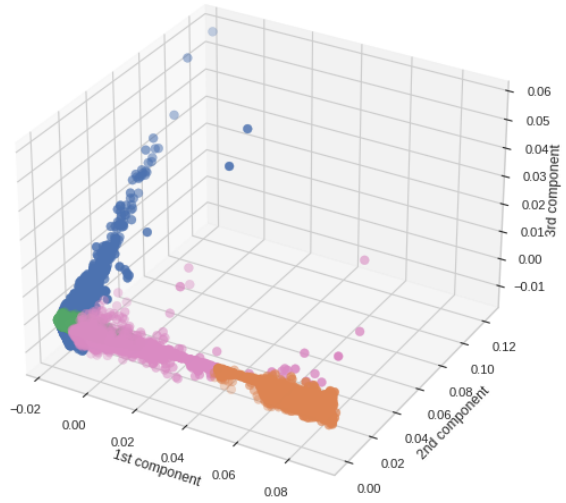
(f) Total high season for clusters **0,1,2,5**.

Fig. 12. Box plots for min-max normalization (III).



(a) Segmentation for 4 mixture components.

(b) Highlighted registered residents.



(c) 3D plot with 3rd component.

Fig. 13. Scatter-plot of the first three components (PCA) using ℓ^2 normalization.

Variables	N° cluster						
	0	1	2	3	4	5	6
nights	1.57	0.26	1.55	108.62	84.66	0.00	68.73
km_to_dest	128.55	1742.97	474.21	19.39	318.36	253.70	1747.30
total_entries	1.54	1.12	1.58	10.34	4.36	1.12	2.71
total_distance	11.64	4.90	10.67	70.24	30.77	4.86	14.42
gross_income	23,085.36	19,482.10	35,547.66	20,972.17	26,902.26	25,151.75	19,179.54
total_high_season	1.01	0.31	1.14	18.85	15.10	0.31	11.24

Table 5
Mean of variables for each cluster performed using min-max normalization.

Data points	N° cluster			
	0	1	2	3
Percentage of sample	10.30%	8.50%	5.25%	75.95%
Real Residents	589	0	0	62
Rest of individuals	3620	3473	2146	30,974

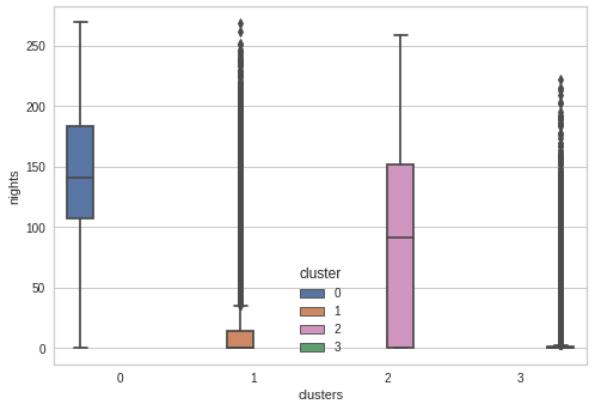
Table 6
Clusters based on actual resident labels using ℓ^2 normalization.

(75.95% of the sample), had an average stay of 4.82 nights (although most did not stay overnight). They covered an average distance of 240.01 km and rarely visited in the high season (28% of the total stay) (see Fig. 14 (f)). It also had the highest income, with an average of 26,158.32.

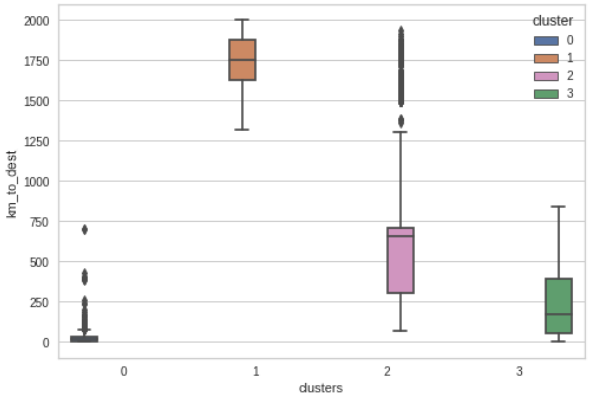
6. Discussion

Table 8 shows the equivalence by clusters and percentage of the total set for the two normalizations analyzed. Additionally, it briefly describes the general profile of individuals in each cluster. For the group of registered residents, we could see that both normalization methods grouped them into a single cluster (cluster 3 in min-max and 0 in ℓ^2). However, there was a 3.17% difference in the size of these clusters, with the ℓ^2 cluster size being smaller. The min-max normalization distinguished between foreign visitors and foreign non-registered residents (clusters 1 and 6, respectively), while the ℓ^2 normalization grouped all foreign individuals into a single cluster (cluster 1). The clusters of national non-registered residents were also similar in both normalization methods (cluster 4 in min-max and 2 in ℓ^2). Still, there was a 5.05% difference in the size of these clusters, with the size of the ℓ^2 cluster also being smaller. Finally, the ℓ^2 normalization grouped all national visitors into a single cluster (cluster 3), while the min-max normalization divided these into three distinct clusters (clusters 0, 2, and 5). It should be noted that in the ℓ^2 normalization, cluster 3 is larger than the sum of clusters 0, 2, and 5, because it contained individuals with resident behaviors that were not included in the other clusters. This explained the significant differences in the sample sizes of clusters 0 and 2 compared to their equivalents in the min-max normalization.

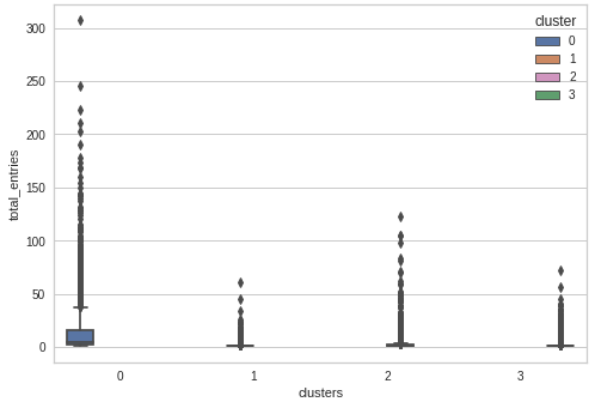
Fig. 15 shows a hierarchical graph comparing the equivalences presented in Table 8 between the two normalizations. We could quickly discern the descriptions that corresponded to each normalization for each cluster type. The min-max normalization seemed more efficient since it allowed a more detailed segmentation of individuals than ℓ^2, and ℓ^2 showed more outliers in the box plots for all the variables. While min-max seemed to distinguish the residents from the visitors, with the variable representing the number of nights spent in the area, ℓ^2 seemed to have a clear segmentation based on the distance to their home. Hence, for our purposes, min-max offered better segmentations. In addition, min-max detected atypical behaviors of individuals not officially registered as residents of the area, but that behaved as residents. In contrast, the ℓ^2 normalization could be useful for excluding foreigners from the analysis and focusing only on comparing registered and non-registered residents at the national level, grouping visitors in a single cluster. Our work, as many in machine learning in real environments, has some limitations related to uncontrolled variables. In particular, we acknowledge that there could be some rented cars



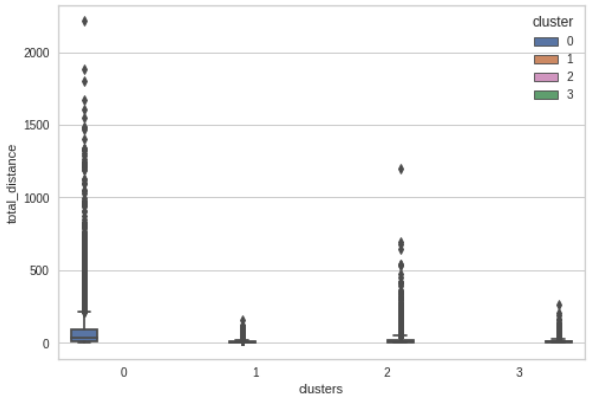
(a) Total nights of stay by cluster.



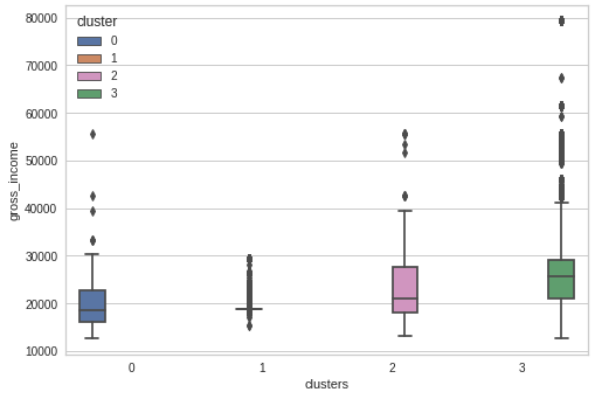
(b) Distance in Km. to the area by cluster.



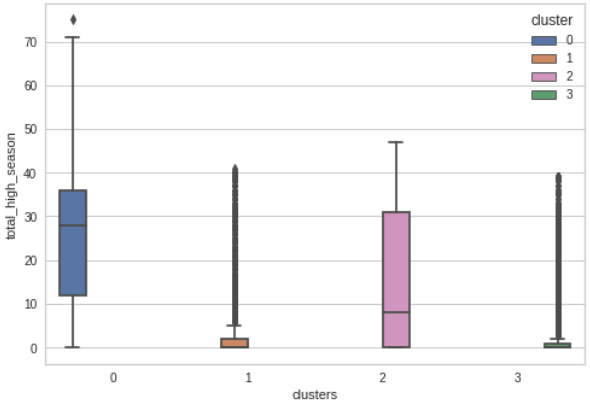
(c) Total entries to the area by cluster.



(d) Distance run in area by cluster.



(e) Average gross income by cluster.



(f) Total high season by cluster.

Fig. 14. Box plots for ℓ^2 normalization.

Variables	Nº cluster			
	0	1	2	3
nights	144.93	22.81	84.62	4.82
km_to_dest	25.54	1750.68	598.01	240.01
total_entries	13.18	1.52	3.53	1.49
total_distance	95.43	6.89	26.43	8.01
gross_income	20,268.41	19,018.55	22,886.88	26,158.32
total_high_season	24.83	3.87	14.47	1.36

Table 7
Mean of variables for each cluster performed using ℓ^2 normalization.

Normalization								
Min-max	Nº cluster	3	1	6	4	0	2	5
	% sample	11.17%	6.11%	3.05%	8.55%	15.04%	8.58%	47.50%
	Description	Registered residents	International visitors	Non-registered international residents	Non-registered national residents	Visitors from Granada who stay for 1-2 nights	National visitors	Visitors from Granada who do not stay night
	Additional characteristics	Long stays and travel frequently in the area	No overnight and visits mostly in low season	Long stays and above-average distance of provenance	Long stays and above-average income and visits	Overnights mostly in high season and weekends	Overnights mostly in high season and above-average income	No overnight and visits mostly in low season
ℓ^2	Nº cluster	0	1		2	3		
	% sample	8.55%	8.76%		4.36%	78.33%		
	Description	Registered residents	International visitors		Non-registered national residents	National visitors		
	Additional characteristics	Long stays and travel frequently in the area	Medium-short stays and visits mostly in low season		Long stays and above-average income and visits	Short stays and visits mostly in low season		

Table 8
Equivalence of the clusters made for each normalization.

with a national plate number that does not match the occupants' provenance; unfortunately, we could not access any rented car database. Likewise, we could not find any good local event calendars, which could affect the traffic.

In summary, our methodology comprises eight steps (see Figure 1). Initially, we gathered data from various sources, cleaned it, and merged it based on vehicle licenses. In this merge step, we also calculated additional variables from the existing ones (e.g., route and total distance in the area). Next, we followed a systematic sequence involving preprocessing, reducing dimensions, and clustering. Ultimately, we evaluate outcomes through visualization techniques. This approach enriches LPR data with contextual information, uncovering novel patterns within the data. Additionally, it facilitates the comparison of algorithm performance, such as comparing different normalization algorithms in the performance of vehicle-behavior clustering. In smart villages, it is important to select suitable LPR locations to cover the towns entries and exits, and it is also important to consider that the official residence could only be partially reliable.

7. Conclusions

The paper presented an effective pipeline for clustering analysis, using data from different sensors and sources to detect registered and non-registered residents and visitors and their behavior in a given area. We selected an optimal clustering algorithm based on the data distribution and two potential normalization algorithms. We found that the min-max normalization was the most effective for detailed segmentation of individuals and their visiting behavior in the area and detection of atypical behavior of individuals not registered as residents of the area but showing resident behavior. The ℓ^2 normalization could be useful in specific situations requiring a distinction from the region of origin. This analysis could assist area managers in crafting tailored strategies to keep certain tourists, considering their income and origin, and promoting overnight stays. This could boost the local economy and reduce traffic. Additionally, these patterns could inform policies to engage non-registered residents in the community, such as tax breaks or social programs. In Spain, this data is crucial for tasks like licensing pharmacies, investing in public health, and scheduling security forces based on sea-

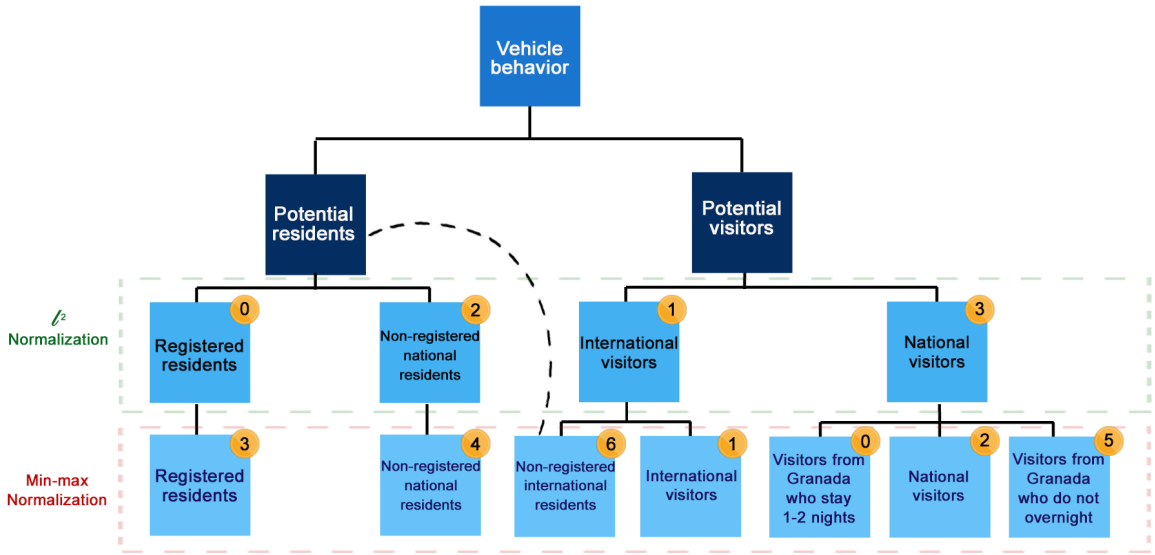


Fig. 15. Hierarchical graph of the two clusters made for each normalization.

sonal fluctuations. Our pipeline and analysis could also assist data analysts in improving their solutions and making informed decisions. In the future, we aim to conduct an independent clustering analysis on the dataset of passing vehicles in the area. The objective is to identify movement patterns and promote longer stays within the vicinity. Likewise, we will try to find useful datasets that could enhance the results, such as vacation accommodation occupancy or local events, although in small villages, it could be a challenge to find good datasets.

CRediT authorship contribution statement

Daniel Bolaños-Martinez: Conceptualization, Investigation, Formal analysis, Methodology, Software, Validation, Visualization, Writing – original draft. **Maria Bermudez-Edo:** Conceptualization, Investigation, Methodology, Project administration, Resources, Supervision, Writing – review & editing. **Jose Luis Garrido:** Investigation, Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

Conflict of interest/Competing interests: The authors have no competing interests to declare relevant to this article's content.

Data Availability Statement

I have shared a link to my data in the Manuscript File.

Acknowledgments

This publication is part of the R&D&i Project Ref. PID2019-109644RB-I00 funded by Ministerio de Ciencia e Innovación/ Agencia Estatal de Investigación/ 10.13039/501100011033, and the R&D&i Project Ref. C-SEJ-128-UGR23 funded by Junta de Andalucía and "ERDF A way of making Europe", and also by the project "Thematic Center on Mountain Ecosystem & Remote sensing, Deep learning-AI e-Services University of Granada-Sierra Nevada" (LifeWatch-2019-10-UGR-01), which has been co-funded by the Ministry of Science and Innovation through the FEDER funds from the Spanish

Pluriregional Operational Program 2014-2020 (POPE), LifeWatch-ERIC action line. The project has also been co-financed by the Provincial Council of Granada. Funding for open access charge: Universidad de Granada / CBUA

References

[1] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Computer networks 54 (15) (2010) 2787–2805.

[2] M. Bermudez-Edo, P. Barnaghi, K. Moessner, Analysing real world data streams with spatio-temporal correlations: Entropy vs. pearson correlation, Automation in Construction 88 (2018) 87–100.

[3] F. M. Garcia-Moreno, M. Bermudez-Edo, E. Rodríguez-García, J. M. Pérez-Mármol, J. L. Garrido, M. J. Rodríguez-Fórtiz, A machine learning approach for semi-automatic assessment of iadl dependence in older adults with wearable sensors, International Journal of Medical Informatics 157 (2022) 104625.

[4] R. P. Centelles, F. Freitag, R. Meseguer, L. Navarro, S. F. Ochoa, R. M. Santos, A lora-based communication system for coordinated response in an earthquake aftermath, Multidisciplinary Digital Publishing Institute Proceedings 31 (1) (2019) 73.

[5] M. A. Mondal, Z. Rehena, Identifying traffic congestion pattern using k-means clustering technique, in: 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), IEEE, 2019, pp. 1–5.

[6] M. Lin, X. Zhao, Application research of neural network in vehicle target recognition and classification, in: 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), IEEE, 2019, pp. 5–8.

[7] M. L. M. Peixoto, A. H. Maia, E. Mota, E. Rangel, D. G. Costa, D. Turgut, L. A. Villas, A traffic data clustering framework based on fog computing for vanets, Vehicular Communications 31 (2021) 100370.

[8] Z. Ning, J. Huang, X. Wang, Vehicular fog computing: Enabling real-time traffic management for smart cities, IEEE Wireless Communications 26 (1) (2019) 87–93.

[9] Ş. Kolozali, M. Bermudez-Edo, N. FarajiDavar, P. Barnaghi, F. Gao, M. I. Ali, A. Mileo, M. Fischer, T. Iggena, D. Kuemper, et al., Observing the pulse of a city: A smart city framework for real-time discovery, federation, and aggregation of data streams, IEEE Internet of Things Journal 6 (2) (2018) 2651–2668.

[10] O. Golovnin, Data-driven profiling of traffic flow with varying road conditions.

[11] G. Yang, D. Coble, C. Vaughan, C. Peele, A. Morsali, G. F. List, D. J. Findley, Waiting time estimation at ferry terminals based on license plate recognition, Journal of Transportation Engineering, Part A: Systems 148 (9) (2022) 04022064.

[12] W. Yao, J. Yu, Y. Yang, N. Chen, S. Jin, Y. Hu, C. Bai, Understanding travel behavior adjustment under covid-19, Communications in Transportation Research (2022) 100068.

[13] P. Wang, J. Lai, Z. Huang, Q. Tan, T. Lin, Estimating traffic flow in large road networks based on multi-source traffic data, IEEE Transactions on Intelligent Transportation Systems 22 (9) (2020) 5672–5683.

[14] Z. Liu, Y. Liu, Q. Meng, Q. Cheng, A tailored machine learning approach for urban transport network flow estimation, Transportation Research Part C: Emerging Technologies 108 (2019) 130–150.

[15] H. Sun, Y. Chen, J. Lai, Y. Wang, X. Liu, Identifying tourists and locals by k-means clustering method from mobile phone signaling data, Journal of Transportation Engineering, Part A: Systems 147 (10) (2021) 04021070.

[16] C. Morris, J. J. Yang, A machine learning model pipeline for detecting wet pavement condition from live scenes of traffic cameras, Machine Learning with Applications 5 (2021) 100070.

[17] J. Enes, R. R. Expósito, J. Fuentes, J. L. Cacheiro, J. Touriño, A pipeline architecture for feature-based unsupervised clustering using multivariate time series from hpc jobs, Information Fusion 93 (2023) 1–20.

[18] B. P. L. Lau, S. H. Marakkalage, Y. Zhou, N. U. Hassan, C. Yuen, M. Zhang, U.-X. Tan, A survey of data fusion in smart city applications, Information Fusion 52 (2019) 357–374.

[19] F. T. Sáenz, F. Arcas-Tunez, A. Muñoz, Nation-wide touristic flow prediction with graph neural networks and heterogeneous open data, Information Fusion 91 (2023) 582–597.

[20] Z. Doborjeh, N. Hemmington, M. Doborjeh, N. Kasabov, Artificial intelligence: a systematic review of methods and applications in hospitality and tourism, International Journal of Contemporary Hospitality Management 34 (3) (2022) 1154–1176.

[21] D. Bolaños-Martinez, M. Bermudez-Edo, J. L. Garrido, Clustering study of vehicle behaviors using license plate recognition, in: Proceedings of the International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2022), Springer, 2022, pp. 784–795.

[22] M. Mallik, A. K. Panja, C. Chowdhury, Paving the way with machine learning for seamless indoor-outdoor positioning: A survey, Information Fusion (2023).

[23] O. Cats, F. Ferranti, Unravelling individual mobility temporal patterns using longitudinal smart card data, Research in Transportation Business & Management 43 (2022) 100816.

[24] A. Gutiérrez, A. Domènech, B. Zaragozí, D. Miravet, Profiling tourists' use of public transport through smart travel card data, Journal of Transport Geography 88 (2020) 102820.

[25] Z. Wang, H. Liu, Y. Zhu, Y. Zhang, A. Basiri, B. Büttner, X. Gao, M. Cao, Identifying urban functional areas and their dynamic changes in beijing: using multiyear transit smart card data, Journal of Urban Planning and Development 147 (2) (2021) 04021002.

[26] F. T. Lima, V. M. Souza, A large comparison of normalization methods on time series, Big Data Research (2023) 100407.

[27] M. Nicholson, R. Agrahari, C. Conran, H. Assem, J. D. Kelleher, The interaction of normalisation and clustering in sub-domain definition for multi-source transfer learning based time series anomaly detection, Knowledge-Based Systems 257 (2022) 109894.

[28] W. Yao, C. Chen, H. Su, N. Chen, S. Jin, C. Bai, Analysis of key commuting routes based on spatiotemporal trip chain, Journal of Advanced Transportation 2022 (2022).

[29] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometrics and intelligent laboratory systems 2 (1-3) (1987) 37–52.

[30] C. C. D. Oliveira, V. M. D. A. Calado, G. Ares, D. Granato, Statistical approaches to assess the association between phenolic compounds and the in vitro antioxidant activity of camellia sinensis and ilex paraguariensis teas, Critical reviews in food science and nutrition 55 (10) (2015) 1456–1473.

[31] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Clustering algorithms and validity measures, in: Proceedings Thirteenth International Conference on Scientific and Statistical Database Management. SSDBM 2001, IEEE, 2001, pp. 3–22.

[32] W. Yao, M. Zhang, S. Jin, D. Ma, Understanding vehicles commuting pattern based on license plate recognition data, Transportation Research Part C: Emerging Technologies 128 (2021) 103142.

[33] S. Pasupathi, V. Shanmuganathan, K. Madasamy, H. R. Yesudhas, M. Kim, Trend analysis using agglomerative hierarchical clustering approach for time series big data, The Journal of Supercomputing 77 (2021) 6505–6524.

[34] B. YU, J. XIONG, A novel wsn traffic anomaly detection scheme based on birch, Journal of Electronics & Information Technology 44 (1) (2022) 305–313.

[35] K. Kim, Spatial contiguity-constrained hierarchical clustering for traffic prediction in bike sharing systems, IEEE Transactions on Intelligent Transportation Systems 23 (6) (2021) 5754–5764.

[36] X. Bai, Z. Ma, Y. Hou, D. Yang, A data-driven iterative multi-attribute clustering algorithm and its application in port congestion estimation, Available at SSRN 4086627 (2022).

[37] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, J. C.-W. Lin, G. Fortino, Deep learning for pedestrian collective behavior analysis in smart cities: A model of group trajectory outlier detection, Information Fusion 65 (2021) 13–20.

[38] A. J. Martín, I. M. Gordo, J. J. G. Domínguez, J. Torres-Sospedra, S. L. Plaza, D. G. Gómez, Affinity propagation clustering for older adults daily routine estimation, in: 2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, 2021, pp. 1–7.

[39] S. Zhao, K. Zhao, Y. Xia, W. Jia, Hyper-clustering enhanced spatio-temporal deep learning for traffic and demand prediction in bike-sharing systems, Information Sciences 612 (2022) 626–637.

[40] F. S. de Moura, C. T. Nodari, Application of the affinity propagation clustering technique to obtain traffic accident clusters at macro, meso, and micro levels, arXiv preprint arXiv:2202.05175 (2022).

[41] B. Priambodo, A. Ahmad, R. A. Kadir, Predicting traffic flow propagation based on congestion at neighbouring roads using hidden markov model, IEEE Access 9 (2021) 85933–85946.

[42] J. Park, J. Jeong, Y. Park, Ship trajectory prediction based on bi-lstm using spectral-clustered ais data, Journal of marine science and engineering 9 (9) (2021) 1037.

[43] H. Li, J. S. L. Lam, Z. Yang, J. Liu, R. W. Liu, M. Liang, Y. Li, Unsupervised hierarchical methodology of maritime traffic pattern extraction for knowledge discovery, Transportation Research Part C: Emerging Technologies 143 (2022) 103856.

[44] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, Understanding of internal clustering validation measures, in: 2010 IEEE international conference on data mining, IEEE, 2010, pp. 911–916.

[45] A. Oliveira-Brochado, F. V. Martins, et al., Assessing the number of components in mixture models: a review, FEP Working Papers (194) (2005).

[46] C. Olivier, F. Jouzel, A. Matouat, Choice of the number of component clusters in mixture models by information criteria, in: Proc. Vision Interface, 1999, pp. 74–81.

[47] Z. Hu, Initializing the em algorithm for data clustering and sub-population detection, Ph.D. thesis, The Ohio State University (2015).

[48] J.-P. Baudry, CLADAG 2015. Book of Abstracts, 2015, Ch. Estimation and model selection for model-based clustering with the conditional classification likelihood, iSBN: 978888467749-9.

[49] G. James, D. Witten, T. Hastie, R. Tibshirani, An introduction to statistical learning, Vol. 112, Springer, 2013.

[50] J. A. Rodrigo, Análisis de componentes principales (principal component analysis, PCA) y t-SNE, accessed: 2023-3-29, available under a Attribution 4.0 International (CC BY 4.0) (2017).

[51] H. Henderi, T. Wahyuningsih, E. Rahwanto, Comparison of min-max normalization and z-score normalization in the k-nearest neighbor (knn) algorithm to test the accuracy of types of breast cancer, International Journal of Informatics and Information Systems 4 (1) (2021) 13–20.

[52] S. Patro, K. K. Sahu, Normalization: A preprocessing stage, arXiv preprint arXiv:1503.06462 (2015).

[53] K. Polat, U. Sentürk, A novel ml approach to prediction of breast cancer: Combining of mad normalization, kmc based feature weighting and adaboostm1 classifier, in: 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ieee, 2018, pp. 1–4.

[54] M. Ayub, E.-S. M. El-Alfy, Impact of normalization on bilstm based models for energy disaggregation, in: 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), IEEE, 2020, pp. 1–6.

[55] R. Gallardo García, B. Beltrán, D. Vilariño, C. Zepeda, R. Martínez, Comparison of clustering algorithms in text clustering tasks, Computación y Sistemas 24 (2) (2020) 429–437.

[56] S. D. Whitaker, Did the covid-19 pandemic cause an urban exodus?, Cleveland Fed District Data Brief (20210205) (2021).

[57] V. Pinilla, M.-I. Ayuda, L.-A. Sáez, Rural depopulation and the migration turnaround in mediterranean western europe: a case study of aragon, Journal of Rural and Community Development 3 (1) (2008).

[58] Á. D. R. Escudero, La alpujarra granadina: un espacio rural diverso y complejo. de sierra nevada al litoral, in: Nuevas realidades rurales en tiempos de crisis: territorios, actores, procesos y políticas: XIX Coloquio de Geografía Rural de la Asociación de Geógrafos Españoles y II Coloquio Internacional de Geografía Rural, Universidad de Granada, 2018, pp. 782–794.

[59] A. Bertuglia, S. Sayadi, A. Guarino, C. López, et al., Reverse migration: from the city to the countryside. the spanish case of alpujarra granadina., Agriregioneuropa 7 (27) (2011) 62–64.

[60] V. Rodriguez, G. Fernandez-Mayoralas, F. Rojo, International retirement migration: Retired europeans living on the costa del sol, spain, Population Review 43 (1) (2004) 1–36.

[61] D. Reynolds, Gaussian Mixture Models, Springer US, Boston, MA, 2009, pp. 659–663.

Appendix A. Supplementary Correlation and Variable Statistics

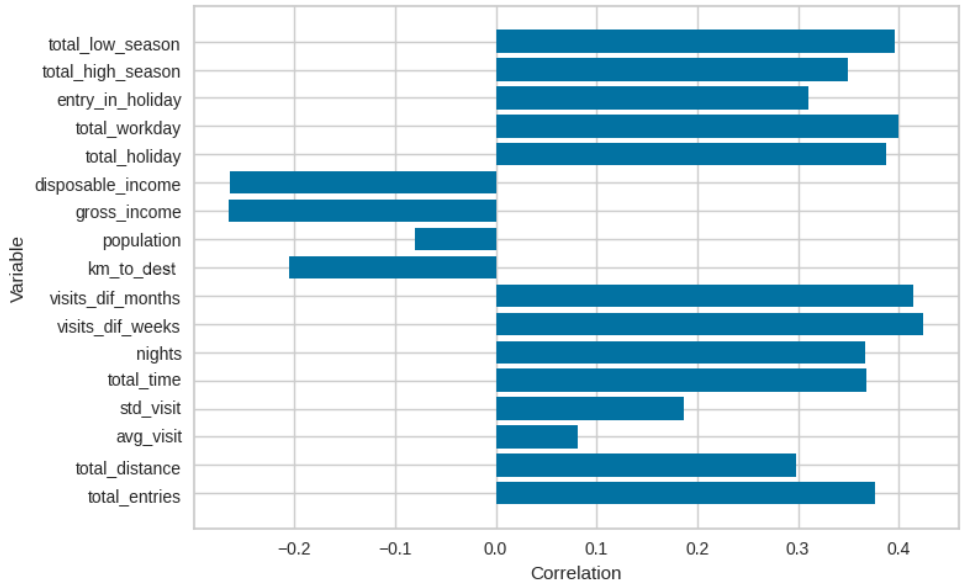


Fig. A.1. Correlation between the registered resident label and the rest of the variables.

	nights		total_distance		total_entries		entry_in_holiday	
	Residents	Others	Residents	Others	Residents	Others	Residents	Others
mean	158.47	19.99	205.82	13.60	19.46	2.35	4.26	0.72
std	72.37	48.07	238.52	47.78	23.57	6.58	5.49	1.70
	gross_income		km_to_dest		visits_dif_weeks		total_high_season	
	Residents	Others	Residents	Others	Residents	Others	Residents	Others
mean	16,084	25,007.07	1.02	374.73	4.57	1.48	27.53	3.84
std	0.00	7671.19	0.59	486.97	4.03	1.97	14.75	9.00
	total_holiday		avg_visit		std_visit		population	
	Residents	Others	Residents	Others	Residents	Others	Residents	Others
mean	52.54	6.83	23.60	10.54	20.26	4.15	406.66	19,8175.90
std	23.71	15.06	34.85	31.87	23.35	16.05	121.16	56,7183.30

Table A.1
Mean and std. deviation for registered residents and rest of individuals in dataset.

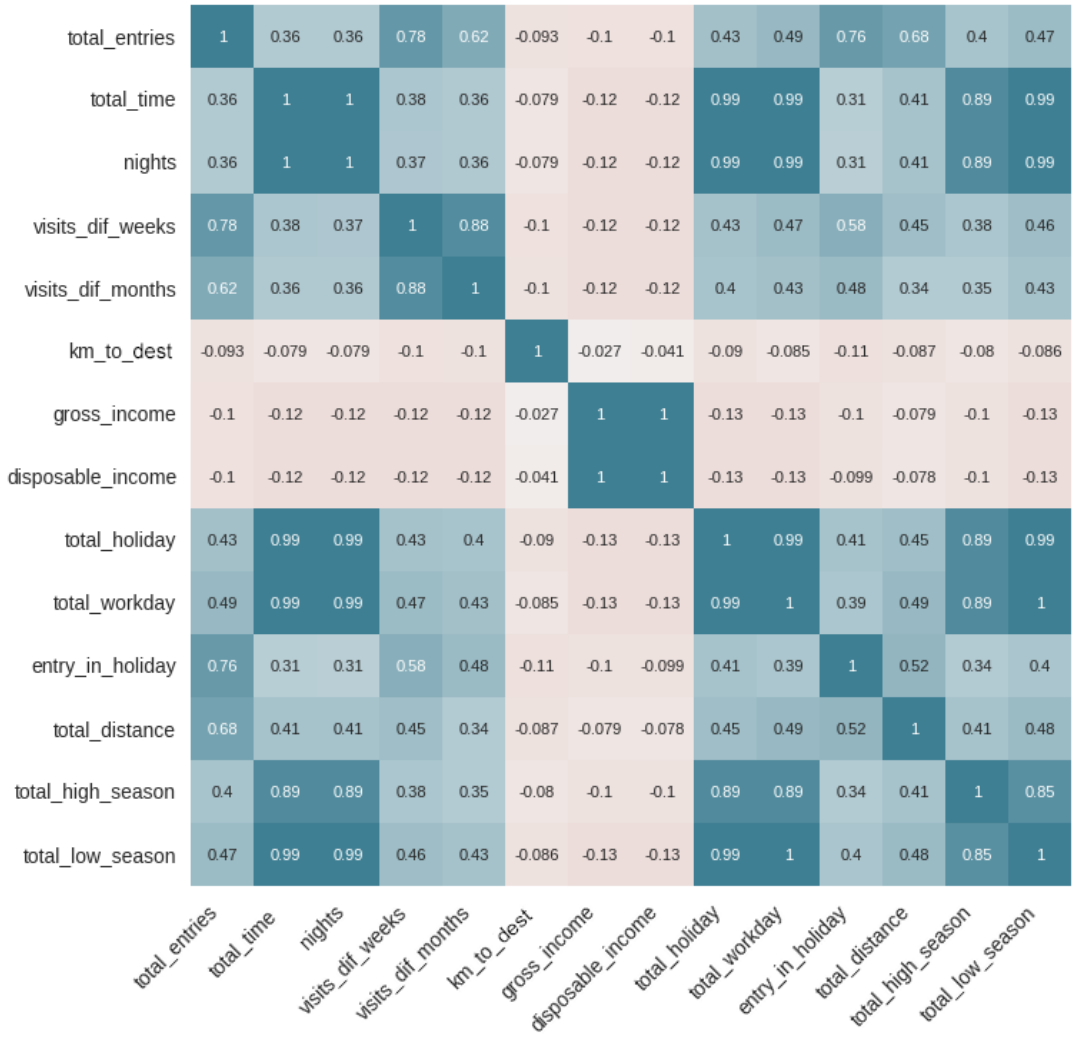


Fig. A.2. Correlation matrix for all variables in the proposed dataset.

3

PREDICTING OVERNIGHTS IN SMART VILLAGES: THE IMPORTANCE OF CONTEXT INFORMATION

[B] Bolaños-Martinez, D., Garrido, J. L., & Bermudez-Edo, M. (2024). Predicting overnights in smart villages: the importance of context information. *International Journal of Machine Learning and Cybernetics*, 1-20.

DOI: 10.1007/s13042-024-02337-7.

- Status: Published.
- Impact Factor (JCR SCIE 2023): 3.1.
- Category: Computer Science, Artificial Intelligence. Rank: 86 / 197 (JIF Q2).
- Number of citations: 1 (Source, Google Scholar).
- Attention score: 12 (Source, Altmetric).
  Mentioned by
  - 11 X users.
  - 1 Redditors.
  - 2 Bluesky users.

  Citations
  - 1 Dimensions.

  Readers on
  - 10 Mendeley.
- Related works: [i], [vii].
- Open source data/software: [ii], [iv], [vi].
- Press notes: [xi], [xii].

Predicting Overnights in Smart Villages: The Importance of Context Information

Daniel Bolaños-Martinez[a,b,*], Jose Luis Garrido[a,b], Maria Bermudez-Edo[a,b]

[a]*Department of Software Engineering, Computer Science School, University of Granada, C/ Periodista Daniel Saucedo Aranda S/N, 18071 Granada, Spain.*
[b]*Research Centre for Information and Communication Technologies (CITIC-UGR), University of Granada, Granada, Spain.*

Abstract

The tourism industry increasingly employs sensors and machine learning for tasks such as demand prediction and mobility forecasting. However, some challenges in data collection remain, especially with information privacy and resource management. We propose a vehicle classification model based on License Plate Recognition (LPR) sensor data, incorporating contextual datasets not explored in the existing literature to predict the number of nights a vehicle will stay in a mountain tourist area. We also study the importance of each dataset in the results. Our analysis utilizes data from four LPR cameras spanning 17 months. We compare different classification models optimized through ensemble techniques. Additionally, an ablation study assesses the impact of each dataset, with variables categorized by expert knowledge into seasonal, socio-economic or visit-related. Optimal dataset selection demonstrates a 22.2% reduction in processing time and an 80% decrease in the number of variables, with only a slight decrease of 0.01 in the Area Under the Curve (AUC) compared to using all available variables. This research provides information to develop tourism prediction models, guiding which datasets and calculated variables are the most important while balancing the processing time and AUC.

Keywords: Tourism forecasting, Sensors, Internet of Things, Machine learning

1. Introduction

In recent years, the use of machine learning has gained prominence in the tourism industry, especially in areas such as forecasting tourist demand, predicting mobility flows, and tourist segmentation [1, 2, 3]. The growing deployment of Internet of Things (IoT) platforms in smart cities has driven the proliferation of sensors to monitor, for example, the traffic on the roads [4, 5]. These sensors can include Global Positioning System (GPS) or License Plate Recognition (LPR) devices and other smart devices [6, 7, 8]. The data collected from these sources provide valuable information on the mobility patterns of visitors and traffic behavior [9, 10].This information can aid stakeholders in the creation of new actions, processes and services [11]. For example, LPRs streamline toll collection on roads, automatically deducting fees from drivers' accounts[1]. They are also used to control parking, performing the control of vehicles entering and leaving by avoiding the conventional paper ticket[2].

In particular, most cities have integrated LPRs to monitor vehicles within their jurisdictions, primarily for law enforcement purposes related to traffic restrictions, but also for traffic monitoring and tourist enhancement. However, most researches using LPR sensors overlook the in-

*Corresponding author.
Email addresses: danibolanos@ugr.es (Daniel Bolaños-Martinez), jgarrido@ugr.es (Jose Luis Garrido), mbe@ugr.es (Maria Bermudez-Edo)

[1] https://www.rac.co.uk/drive/advice/legal/the-dartford-crossing-charge/

[2] https://www.hubparking.com/features/license-plate-recognition-lpr/

clusion of additional contextual datasets [12, 13, 14, 15]. Only a few investigations integrate LPR with location information [16, 7], and none of them consider other contextual data. Furthermore, some works investigate feature selection and assessment of the relevance of each variable in prediction models [17, 18, 19]. These analyses focus on examining the impact and weight of each variable [20, 21], usually using methods such as chi-square or data frequency [22, 23]. However, none of them consider the importance of looking at complete datasets instead of just single variables. The importance of the whole dataset is relevant, as accessing specific datasets presents significant challenges. For instance, these challenges may include the associated data costs and privacy concerns regarding the data [24, 25, 26, 27].

This paper proposes an ablation test to study the importance of different datasets related to LPRs and contextual information in the field of tourism, as well as the use of calculated variables not used in the previous literature. In particular, we have created a vehicle classification on the number of nights spent in a rural tourist region in high mountains. The model uses anonymized variables from LPR devices and is enriched with contextual information, such as the owners' residence location, and calculated variables, such as the kilometers traveled in the area. Using the collected data, we leverage historical information on vehicle visits to the area to predict the behavior of their current visit in terms of number of nights they will spend in the area. We compared different classification models commonly used in the literature and optimized the best ones using ensemble techniques. Variables are divided into different datasets based on context and data source, including seasonal, socio-economic, and geographic factors. Using the created datasets, an ablation study highlights which datasets influence the model most. Finally, we present the datasets and variables that minimize processing time while maintaining a high value for the model evaluation metric.

Hence, the contribution of this article is twofold. First, we present a predictive model of the number of nights a vehicle stays in the area, taking advantage of contextual variables not considered in the existing literature. Second, we propose an ablation study focusing on datasets from various expert knowledge domains and data sources to assess the most valuable variables for our model.

The remainder of the paper is organized as follows. Section 2 describes the related work. Section 3 presents the methodology discussed throughout the paper. Section 4 explains the experiments realized, including a background of the use case, and Section 5 shows the analysis and discussion of the results. Finally, Section 6 concludes the paper.

2. Related work

Over the past few years, the utilization of digital tools within the tourism industry, particularly in promoting and managing services, has notably risen. These approaches use data analysis techniques, combining information to create smart tourism applications that enhance the visitors' experience. These applications use various techniques such as machine learning models and feature selection to improve the applications [28]. It is popular to estimate tourist flows when the main source of information is historical data, even mixing other sources such as search engines or social media [29, 1, 30]. However, with the emergence of the Smart Cities paradigm, significant progress has been made in merging sensor data installed in cities and external data sources that notably improve the performance of the models [31]. The fusion of sensor data with other data sources, such as geotagged social-networks data, holiday information, or weather information [2, 32, 20], allows building more robust and efficient machine learning models.

In particular, most of the deployments in Smart Cities incorporate LPRs sensors with or without contextual information, which makes it possible to analyze how vehicles move. Research with LPRs focuses mainly on clustering vehicles to obtain useful information for city management [8, 33, 4]. Some studies center on prediction models with these sensors, and they mainly focus on traffic predictions at intersections in networks [12, 13, 25]. While a few studies have built variables based on the frequency and duration of visits using raw sensor data, there is a scarcity of studies utilizing additional contextual datasets [12, 13, 14, 15]. Few works combine LPR with other contextual information, such as the location flows in the area, extracted from GPS trajectory or Cellphone Location (CL) [7, 16]. Moreover, the related literature omits calculated variables that could improve the raw data, such as the distance traveled or the number of nights spent in a

tourist area. The scarcity of research on tourism forecasting may be attributed to data collection, modeling, and storage challenges, such as data privacy and the complexity of federating data [26]. When using sensor data, issues become more pronounced, making it difficult to manage data collections or design the installation system [27]. In the case of LPR, which includes license plate numbers, privacy becomes a significant concern, leading to limited availability of such information [25].

Regarding the importance of the variables or the set of variables, the problem of feature selection is a crucial aspect in enhancing the performance of classification models. Existing literature explores this issue in various domains, such as sentiment analysis [34, 22] and even in tourism [23]. However, these studies predominantly rely on algorithmic approaches such as chi-square, data frequency, or information gain to identify influential features and eliminate those with negligible effects. Another method for assessing variable importance in machine learning models is the ablation study [18, 21]. Research employing ablation studies examines the impact of models or optimization algorithms on different phases of learning by removing some of the steps in the pipeline [35], as well as the inclusion and exclusion of attribute information vectors [36]. None of these studies have applied ablation tests at the data level to complete datasets. We address the gaps in the literature in two ways. First, we create a predictive model using LPRs, calculated data, and contextual information not used in previous studies, such as time spent in the area or gross income. Second, we perform an ablation test at the dataset level.

3. Methodology

The methodology proposed to detect the relevance of each dataset in developing a forecasting model of visitor flows starts with a basic dataset, which is the raw information coming from LPR sensors; constructs the machine learning model; and calculates the model's performance. Then, it adds different datasets one by one and calculates the improvement of each dataset in the performance of the model. Fig. 1 illustrates the methodology in six steps. These steps follow a machine learning pipeline, with commonly used steps, such as steps 2-5: learning methodology selection, cleaning, normalization, algorithm, and metric selection. The emphasis of this methodology resides in steps 1 and 6. In step 1, we create a dataset calculating new variables from the raw LPR data not previously used in the literature, and we merge this dataset with context datasets. In step 6, we perform an ablation test based on the datasets. In particular, the methodology starts in the first round, selecting all the datasets in step 1, follows steps 2 to 5, calculates the results in step 6, and saves them. Then, we select in each of the following rounds a dataset together with the base dataset in step 1, perform steps 3 and 4, calculate the results with the algorithms and metrics selected in steps 2 and 5 of the first round, and save the results in step 6. We can repeat the steps with any combination of datasets in step 1, normally combining the datasets with best results. Finally, we compared all the results of each round and analyzed which datasets had better results. Optionally, we can perform a feature selection over the best datasets to reduce the processing time. In the following subsections, we explain the six steps.

3.1. Dataset construction

The first step is to retrieve and create the different datasets. The main source of information is the different LPRs that cover the touristic area. The LPR raw data consists of the license plate number and timestamp at which the corresponding LPR detects the vehicle. Adding the LPR identifier, we have the LPRs raw dataset. It is important to anonymize the license plate with a unique value representing each vehicle. As the main advantage of the LPRs is the identification of the vehicles, all the datasets will be centered on the vehicle. That is, each row will represent a vehicle. We can create three types of datasets:

- **Base dataset:** only uses LPR data and contains calculated variables from the raw sensor data that could be considered basic for the study. This dataset aggregates the vehicles' information to center the dataset on the vehicle. For tourism studies, it is important to separate the information into visits, considering each visit as the time a vehicle spends between entering and leaving the touristic area, which normally includes several LPRs. We can aggregate the data by averaging the values or by accumulating the values. For example, the number of visits will be created by accumulating the values of each visit. The average

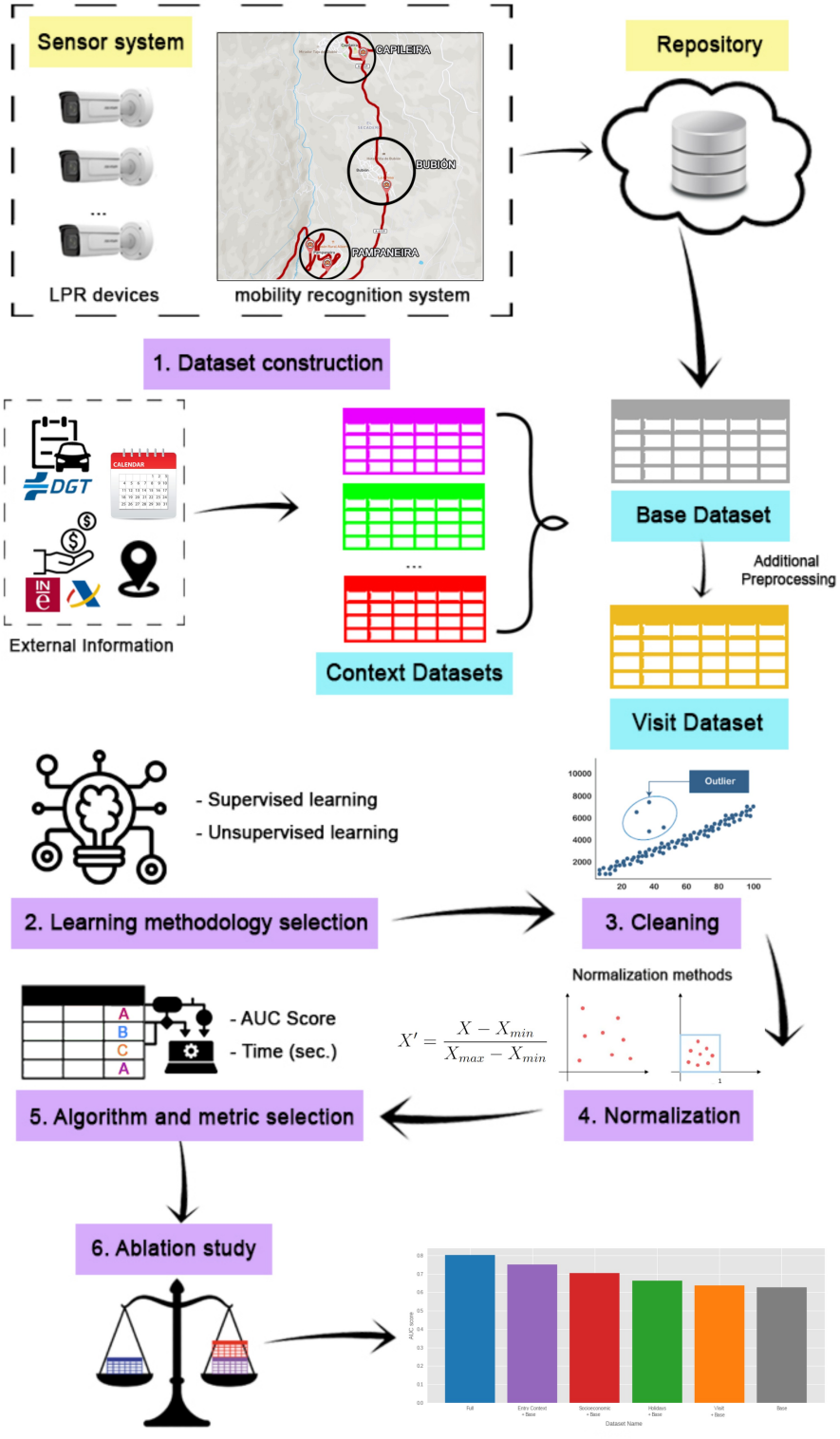


Fig. 1. Structure diagram that outlines the methodology used.

time spent on each visit will be calculated as the average of the individual times. This dataset represents the behavior of each vehicle in the touristic area.

- **Visit dataset:** extends the base dataset with calculated variables from the raw data that complement the basic information about vehicle visits in the area. These variables include detailed information on the behavior of a vehicle in the tourist area. Still, it is complementary information that we believe is important but not as important as the base dataset, such as the nights spent in the area.

- **Context datasets:** are external information that could enhance the results. For example, if a vehicle is detected during a public holiday or a working day. This information is dependent on the datasets that the researchers or analysts could access.

3.2. Learning technique selection

This step chooses the appropriate learning method to address the problem under study. For example, decide whether the problem is a regression or a classification problem and which variables and datasets could be relevant to that problem. For example, regression is adequate for continuous variables, while classification is adequate for categorical variables. Additionally, regression tends to have problems with large data variability, providing predictions with a large margin of error [37].

3.3. Cleaning

As the LPRs have an inherent percentage of detection error, these errors add noise to the data. Hence, it is necessary to perform an anomaly elimination prior to model training, eliminating vehicles with strange behavior. This step identifies and eliminates samples with atypical behaviors (anomalies that introduce noise), keeping only the most relevant and representative data for the analysis. Eliminating these anomalies improves the quality and metrics values of the machine-learning models.

There are several anomaly detection techniques; one of the most used is the dimensionality reduction with Principal Component Analysis (PCA) [38] with the use of an algorithm on these two components to detect and eliminate possible anomalies in the data [39]. For example, Isolation Forest is an anomaly detection algorithm based on the concept of isolating observations that are rare and different from the rest of the data. This algorithm builds multiple random decision trees to partition the data, and anomalies generally end up isolated in smaller regions of the feature space, easily detected. There are other anomaly detection algorithms that could be used. Furthermore, we could, using expert information, eliminate data that is above or below a threshold in one variable. For example, for vehicles that spend only 5 minutes in the area, we can consider them as pass-by vehicles and not tourists in the area, so we can eliminate them. These pass-by vehicles do not generate any revenue or consume any resources or infrastructures other than the road, so they are not relevant for tourism managers.

3.4. Normalization

When working with different sources of information, the existence of attributes at different scales erroneously increases the influence of some variables over others in the grouping process. Normalization compresses or expands the values of each variable to adjust them to the same range of values of the other variables, making them comparable. The choice of normalization algorithm usually depends on the specific application and the data distribution, as different methods may yield different results and interpretations. In the literature, several normalization methods have been used, among which min-max normalization and Z-score normalization [40, 41] stand out. Additionally, other applicable methods, such as MAD, logarithmic, or l_2 transformations, have been explored [42, 43].

3.5. Algorithms and metrics selection

Choosing the algorithms that build the model and defining the appropriate evaluation metrics is essential. In this process, different options and families of machine learning algorithms could be explored in order to identify the most suitable for a specific problem. To do so, it is advisable to perform comparative tests between different types of algorithms, balancing their performance and processing time. It is also advisable to use algorithms from different families, from traditional [44, 45, 19] to deep learning [46, 47], and paying special attention to the algorithms used in the field of tourism. We have identified the following families recently used in tourism studies:

- **Decision Trees:** The main representative algorithms of this family are Decision Tree and Random Forest [48]. These algorithms, although simple in their rule-based implementation, are still used in the field of predicting tourist flow behavior due to their speed and interpretability of results [49, 50, 51].

- **Nearest Neighbors:** K-Nearest Neighbors [52] allows optimization and improvement by adjusting the parameter K, which indicates the number of nearest neighbors for distance calculation. It has recently been used in tourism decision and recommendation systems [53, 54].

- **Logistic Regression:** Logistic Regression [55] predicts categorical outcomes based on predictor variables, aiding decision-making. It is currently used in tourism planning and development [56, 57].

- **Gradient Boosting:** One of the most representative algorithms of this family is Light GBM [58], based on gradient boosting, iteratively improves model performance by combining several decision trees. It is widely used in various fields with problems of large datasets and complex feature interactions including tourism [59]. XGBoost Classifier [60] uses sparse data algorithms, weighted quantile sketching, and optimization techniques to enhance classification outcomes. In recent years, XGBoost has been used in predicting tourism trends based on sentiment analysis [61, 62], and improve the performance of other combined algorithms [63]. CatBoost Classifier [64] employs ordered reinforcement and symmetric or forgetting trees to handle categorical variables efficiently. It has been recently used for hotel bookings studies [65, 66].

- **Neural Networks:** One of the most used neural networks in tourism, when there is a large amount of data is the MLP Classifier [67], based on multiple perceptron layers [46, 47]. The recent algorithm TabNet [68] employs transformers in neural networks including an attention mechanism in feature selection, making it especially effective for tasks with high-dimensional structured data and time series. It has been used for example for customer purchases [69].

- **Bayesian Probabilistic Models:** For example, Gaussian Naive Bayes [55] is a probabilistic classification algorithm based on Bayes' theorem. It is used in recommendation systems [70] and as a comparison with other algorithms [49].

- **Support Vector Machines:** For instance, Linear SVM [71] is used primarily for classification and regression. It has been used for forecasting tourist [72], and it is often used as a benchmark algorithm to compare it with other algorithms [45, 73].

Each family of algorithms has its own characteristics and advantages [74]. Still, we can combine them using Bagging, Voting, and Stacking ensemble strategies to test whether some algorithms work better with each other to improve the overall performance of the model [75, 76]. These combination techniques allow us to leverage the strengths of each algorithm and mitigate their weaknesses, which could increase the evaluation metrics values and generalization of the prediction model [77].

The common validation of the resulting models is the fold cross-validation technique, which divides the dataset into different subsets. For example, 5-cross validation divides the dataset into 5 subsets. The model undergoes training and testing five times, each iteration using a distinct subset as the test set and the remaining four as the training set. It is advisable to select several metrics to compare the algorithms and test their performance because some are general metrics, such as F1-score or Area Under the Curve (AUC), providing a measure of overall performance, taking into account the balance between the ratios of false positives and true positives [78]. The balance between these two ratios allows us to assess the model's ability to distinguish between right and wrong classifications fairly. Other metrics provide nuanced results in terms of detecting the positives better than the negatives, etc., and together they provide an overview that can serve to study the robustness of the results. The common metrics are Accuracy, Precision, Recall, Specificity, F1-score, and AUC score.

3.6. Ablation study

The core of this methodology is to perform an ablation test at the level of the dataset. In particular, the proposal is to select the relevant datasets that enhance the results.

Ablation studies help in the selection of these datasets. The ablation study evaluates the model's performance after eliminating a variable or set of variables in each run. By comparing the results obtained with and without a specific set of variables, we can determine its relative importance and contribution to the model. This ablation could be performed in two steps:

- Individual evaluation of each dataset: First, evaluate the model's performance using each dataset individually. In this way, we can analyze each dataset's impact on the prediction model. The variables defined in the base dataset are included in all evaluations, as they are considered to be the minimum information needed from the cameras to create a prediction model. The other datasets are additional and do not have the capability to build a robust model with the necessary information to obtain a good value for the chosen evaluation metric. This analysis identifies which dataset or combination of datasets has the most significant influence on the final model.

- Correlation analysis and variable selection: Subsequently, select the datasets that have proven most influential in the previous phase and perform a correlation study among its variables to determine which provides the most relevant information to the study. The aim is to achieve a model performance as close as possible to that obtained by using all the variables in the model. We can optimize the model and reduce its complexity by identifying the most significant variables.

4. Experiment

We center our use case in a rural tourist area, facing problems of over-tourism and protection of a nearby national park. In particular, we cover three rural villages near the Sierra Nevada National Park in Granada, Spain. These municipalities, known for their scenic appeal, face challenges of over-tourism, leading to traffic congestion. We have deployed an LPR camera system to monitor the traffic. The system, consisting of four Hikvision LPR IP cameras with Deep Learning-based ANPR, monitors vehicle movement at key entry and exit points (see in Fig. 2). Due to the road structure, four cameras can cover all the entrances and exits to the three villages. This approach optimizes costs and system complexity while providing comprehensive data on mobility in the area. The road ends in the town of Capileira, and there is a fork in the road in the map's southern area, allowing entry into the area through two towns, Pampaneira and Bubion. The data spans from February 2022 to June 2023 (17 months). We apply the methodology described in Section 3 to build the model for forecasting the number of nights a vehicle entering the area will stay there.

4.1. Datasets definition

The structure of our 35 variables can be found in Table 1, and are divided into 5 datasets, one base dataset, one visit dataset, and three context datasets (holiday, socio-economic, and entry) as follows:

- **Base dataset:** consists of two variables that contain the minimum information needed to represent the spatio-temporal frequency of a vehicle in the area: total entries (number of visits), and average time of visit.

- **Visit dataset:** These variables, not found in LPR literature, provide information on the spatio-temporal behavior of the vehicles and nights in the area.

- **Holiday dataset:** contains different variables related to vacations and working days based on external information from national calendars. High season days include the most important holiday periods in Spain: Summer Holiday, Christmas, and Holy Week[3]. The national calendar data are from the Python library "holidays"[4].

- **Socio-Economic dataset:** contains variables related to the vehicle's provenance based on external information from a private dataset provided by the Spanish Directorate-General for Traffic (DGT)[5]. We obtained the distance to the area from their provenance information and two libraries: pgeocode[6] and

[3] https://es.statista.com/temas/3585/vacaciones-en-espana/#topicOverview

[4] https://python-holidays.readthedocs.io/en/latest/

[5] https://sede.dgt.gob.es/es/vehiculos/informe-de-vehiculo/

[6] https://pgeocode.readthedocs.io/en/latest/

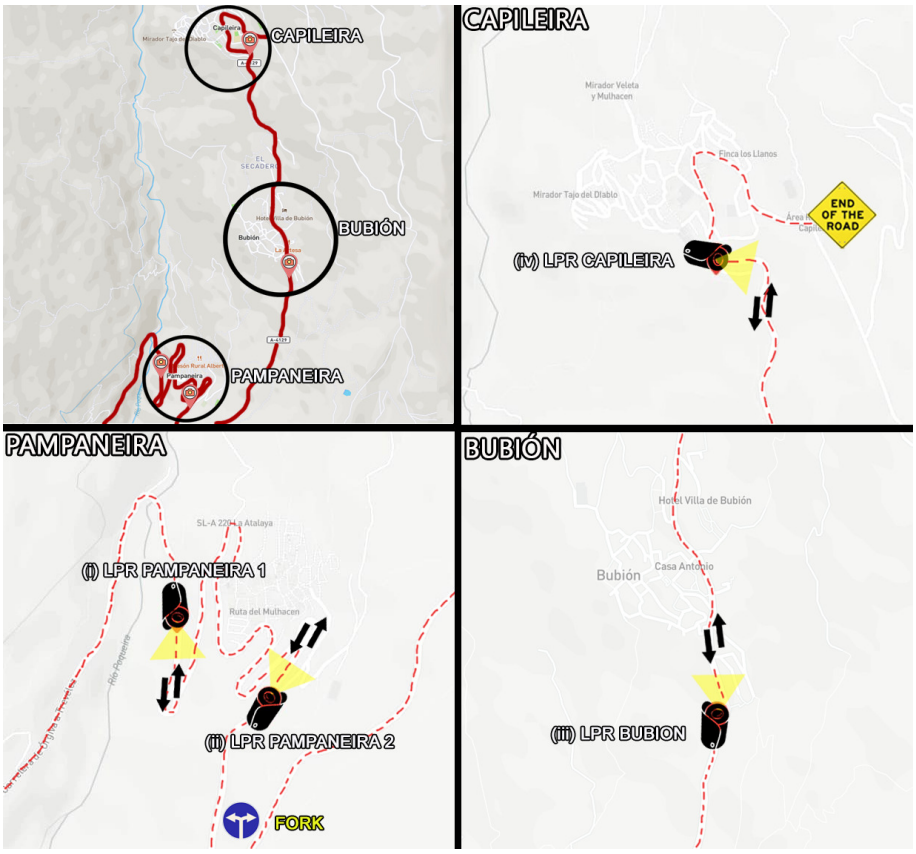


Fig. 2. Setup of the 4 LPR that obtains the vehicle data.

geopy[7]. The population and gross income are obtained from the website of the National Statistics Institute (Spanish: Instituto Nacional de Estadistica, INE)[8].

- **Entry dataset:** obtains information from the spatio-temporal information of the current visit of the vehicle. This information corresponds only to data obtained from LPR cameras at the time of vehicle entry and is unrelated to its behavior within the area. For the variable "current_entry_east", we choose between the two east entrances to the area (LPR Pampaneira 2 or LPR Bubion). This decision is based on the fact that these two cameras cover the only access points from the eastern region to the area, given that there is a single road connecting the villages. If the vehicle does not enter through either, it enters from LPR Pampaneira 1 on the west side of the road. Entering from the east or west is important because, normally, vehicles access the mountain region from the west. If a vehicle comes from the east, it suggests it comes from nearby mountain villages.

4.2. Learning methodology selection

We aim to predict the number of nights a vehicle will spend in the zone (variable current_entry_nights). Hence, we use the last visit of each vehicle to label the dataset with the dependent variable current_entry_nights. To train the model, we use historical information about the vehicle (stored in visit and holiday datasets), socio-economic data, and information about the current context at the time of entry to the area: collected in the entry context dataset and two holiday dataset variables (current_entry_in_holiday and current_entry_in_high_season variables). However, in cases where vehicles only visit the area once, the variables associated with their previous stays do not exist. In this case, the variables recorded in visit and holiday datasets have a value of zero. Despite appearing to lack information, this may help the model consider the behaviors of vehicles visiting the area for the first time.

[7] https://geopy.readthedocs.io/en/latest/

[8] https://www.ine.es/dynt3/inebase/es/index.htm?padre=7132&capsel=5693

Our problem falls under supervised learning and could fit into regression or classification. We chose the classification because we have large data variability. In addition, for our problem, predicting exact details such as length of stay in hours is unnecessary; hence, applying regression would not be appropriate. Since our output variable is numerical, we must categorize or discretize it into intervals to apply classification algorithms. We define three different intervals to obtain classification labels (day, short, and long visits) so that the model prediction provides relevant information to policymakers.

4.3. Cleaning

To detect anomalies, we apply PCA to the two most relevant components and use the Isolation Forest algorithm. Furthermore, we have opted to exclude vehicles that spend less than 3.5 hours within the area. This exclusion significantly enhances the models' evaluation metrics. It is crucial to note that our model does not incorporate this information during training, resulting in a 3.5-hour forecast delay. Vehicles departing before this time are treated as passing through.

4.4. Normalization

All timedelta variables have been converted to numerical values (in units of hours) to prepare them for the application of normalization methods and subsequent machine learning algorithms. We use the five normalization methods most used in the literature (min-max, Z-score, MAD, logarithmic, and l_2) to check which one is the most suitable for our data and analysis.

4.5. Algorithms and metrics selection

We employ various classification algorithms to construct the prediction model and evaluate which one yields the best results for our specific dataset and problem. In particular, we use the most commonly used in tourism, as described in subsection 3.5: Decision Tree, Random Forest, K-Nearest Neighbors, Logistic Regression, Light GBM, CatBoost, XGBoost, MLP Classifier, TabNet, Gaussian Naive Bayes, and Linear SVM. Additionally, we employ Bagging, Voting, and Stacking ensemble strategies. Our approach incorporates a five-fold cross-validation technique. We utilize the most common evaluation metrics: Accuracy, Precision, Recall, Specificity,

Name	Variable	Type	Description	Data source
Base	total_entries	Integer	Total number of entries.	LPR cameras
	avg_visit	Timedelta	Average visit time.	
Visit	visit_time	Timedelta	Total time of stay.	LPR cameras
	std_visit	Timedelta	Standard deviation of the average visit time.	
	distance	Float	Total distance traveled in kilometers within the area.	
	nights	Integer	Number of nights.	
	avg_nights	Float	Average number of nights.	
	std_nights	Float	Standard deviation of the average number of nights.	
	fidelity	Float	Relative number of visits after maintaining fidelity of at least five days.	
	visits_dif_weeks	Integer	Number of different weeks with at least one visit.	
	visits_dif_months	Integer	Number of different months with at least one visit.	
Holiday Context	num_holiday	Integer	Number of holidays spent.	National calendar
	avg_holiday	Float	Average number of holidays spent.	
	std_holiday	Float	Standard deviation of the average number holidays spent.	
	num_workday	Integer	Number of workdays spent.	
	avg_workday	Float	Average number of workdays spent.	
	std_workday	Float	Standard deviation of the average number workdays spent.	
	num_high_season	Integer	Number of high season days spent.	
	avg_high_season	Float	Average number of days of high season spent.	
	std_high_season	Float	Standard deviation of the average number of days of high season spent.	
	num_low_season	Integer	Number of low season days spent.	
	avg_low_season	Float	Average number of days of low season spent.	
	std_low_season	Float	Standard deviation of the average number of days of low season spent.	
	entry_in_holiday	Integer	Number of entries on holiday.	
	entry_in_high_season	Integer	Number of entries in high season.	
	current_entry_in_holiday	Boolean	Binary value indication if the current entry is on a holiday.	
	current_entry_in_high_season	Boolean	Binary value indication if the current entry is on a high season day.	
Socio-Economic Context	km_to_dest	Float	Distance in kilometers between the origin of the vehicle and the destination region.	Geographic data (DGT)
	population	Integer	Population size of the city/town of the provenance of the vehicle.	Demographic and Economic data (INE)
	avg_gross_income	Float	Average gross income of the area of origin of the vehicle.	
Entry Context	current_entry_east	Boolean	Binary value indicating if the current entry is from LPR Pampaneira 2 (ii) or LPR Bubion (iii) of Fig 2.	LPR cameras
	current_entry_hour_morning	Boolean	Binary value indicating if the current entry is during the time interval from 6 a.m. to 12 p.m.	
	current_entry_hour_afternoon	Boolean	Binary value indicating if the current entry is during the time interval from 12 p.m. to 6 p.m.	
	current_entry_hour_night	Boolean	Binary value indicating if the current entry is during the time interval from 6 p.m. to 12 a.m.	
	current_entry_hour_dawn	Boolean	Binary value indicating if the current entry is during the time interval from 12 a.m. to 6 a.m.	

Table 1
Definition of the variables from different datasets.

F1-score, and AUC score, with a primary focus on the AUC metric for discussions. Furthermore, we conduct validation on the training set to compare with the test set, determining potential overfitting, and calculate processing time to select the best algorithm.

4.6. Ablation study

Finally, we employ the ablation study method to evaluate the impact of the different datasets presented in Section 4.1 on our selected model. And then perform a correlation analysis to select the most relevant variables within the best datasets.

5. Results and discussion

We develop all the experiments using a computer system equipped with 40 Intel(R) Xeon(R) Silver 4210 CPUs operating at 2.20GHz, and a total memory capacity of 93GB RAM. For the GPU-accelerated computations, we use one of the server NVIDIA GeForce RTX 3090 graphics cards with 24GB memory, with CUDA 8.6 support. The programming environment is Anaconda with Python 3.9.12. We use the Python library scikit-learn[9] to implement the algorithms: Decision Tree, Random Forest, K-

[9] https://scikit-learn.org/stable/

Nearest Neighbors, Logistic Regression, MLP Classifier, Gaussian Naive Bayes and Linear SVM. We use lightgbm library[10] to implement Light GBM, catboost library[11] for CatBoost Classifier, xgboost library[12] for XGBoost Classifier and pytorch_tabnet library[13] for TabNet Classifier.

The cleaning step in the experiments consists first on removing the passing-by vehicles. Then, we apply the outliers' detection technique, detecting 6% of the sample (3,424 vehicles) as outliers. In Fig. 3, the first two PCA components of the dataset are visualized, with the outliers detected by the Isolation Forest algorithm in red. The final sample is 26,490 vehicles. Then, we discretize the classification variable into three classes because they split the data into three explainable visitor groups. Thus, we have one class with 7,532 vehicles that spend 0 nights in the area (day visits), 11,915 vehicles that spend between 1 and 5 nights (short visits), and 7,043 vehicles with more than 5 nights of stay (extended visits). We also tried different normalization techniques proposed in Section 4.4 and found that the min-max normalization yielded the best results for all algorithms. Hence, we use it for the rest of the analysis. To calculate the different metrics of the multiclass problem, we use the weighted strategy, that calculates the metrics for each label and finds their weighted average according to the number of true instances for each label. This strategy is good when we work with unbalanced problems [79]. In this case, the weighted recall is equal to the accuracy, so we only show one of them (accuracy) in results.

Table 2 and Table 3 show the evaluation metrics and processing time of the eleven algorithms on the entire dataset of the train and test sets, respectively. Regarding the general metrics, AUC and F1-score, we can see that F1-score consistently performs worse than AUC in all the algorithms. This is due to the fact that, unlike AUC, F1-score diminishes its value with unbalanced data [80]. In our data, one of the classes is higher than the other two, hence we have unbalanced data. F1-score is around 0.06 and 0.1 points below AUC in all the algorithms, except in Decision Tree, that the difference is lower and Gaussian Naive Bayes, that the difference is higher. However, both of them yield worst results than the rest of the classifiers. Hence, for our imbalanced data, focusing on the AUC, we can observe that the best algorithms are the ones of the family Neural Networks: MLP and TabNet, and those of the Gradient Boosting family: Light GBM, XGBoost, CatBoost, achieving AUC values close to or equal to 0.80 for the test set. Light GBM algorithm outperforms MLP, TabNet, XGBoost and CatBoost Classifiers in terms of AUC, and MLP, TabNet and CatBoost in terms of processing time. This information can be seen graphically in Fig. 5, which shows the Receiver Operating Characteristic (ROC) curve of the different algorithms. Fig. 4, shows a radar plot displaying the comparative AUC score of the different algorithms in training and test. The decision tree-based algorithms show higher overfitting, unlike the rest of the algorithm families, which present similar values in both test and training data. This suggests that the algorithms of the other families can generalize correctly.

Although the AUC and F1-score are complete metrics that consider the model's general performance, other metrics can provide nuances of the models. We can see also that all the metrics in Table 3 follow the same tendency in all the algorithms, being the highest metric the specificity and the lowest the precision for all the classifiers. Hence, there is no distinction between classifiers, but in general we can say that for our data the probability of correctly classifying into a class (recall/accuracy) is lower than detecting that the vehicle does not belong to a class (specificity). Gaussian Naive Bayes performs poorly in all the metrics, so this algorithm is unsuitable for our problem.

As we observe that Light GBM, XGBoost, CatBoost and MLP perform better than the others and do not overfit, we attempt to further improve the results by employing ensemble algorithms. We discard TabNet from these analyses since its processing time is much higher than the others, 322% higher to the slowest one (MPL), and produces worst AUC values (0.001 lower than the worst classifier -MPL-). Table 4 shows different tests for stacking and voting ensemble strategies. Different algorithms using the stacking strategy fail to overcome the AUC value of the Light GBM algorithm alone. Although stacking has the advantage of combining the strengths of multiple models, possibly due to the complexity of the interaction between the different models, it failed in this case.

[10] https://lightgbm.readthedocs.io/en/latest/Python-API.html
[11] https://catboost.ai/en/docs/
[12] https://xgboost.readthedocs.io/en/stable/
[13] https://dreamquark-ai.github.io/tabnet/

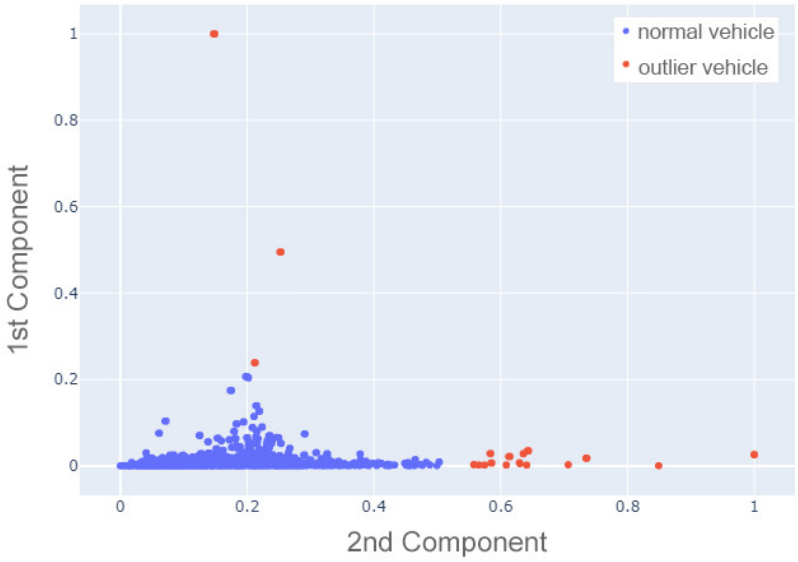


Fig. 3. Outliers detected by the Isolation Forest algorithm

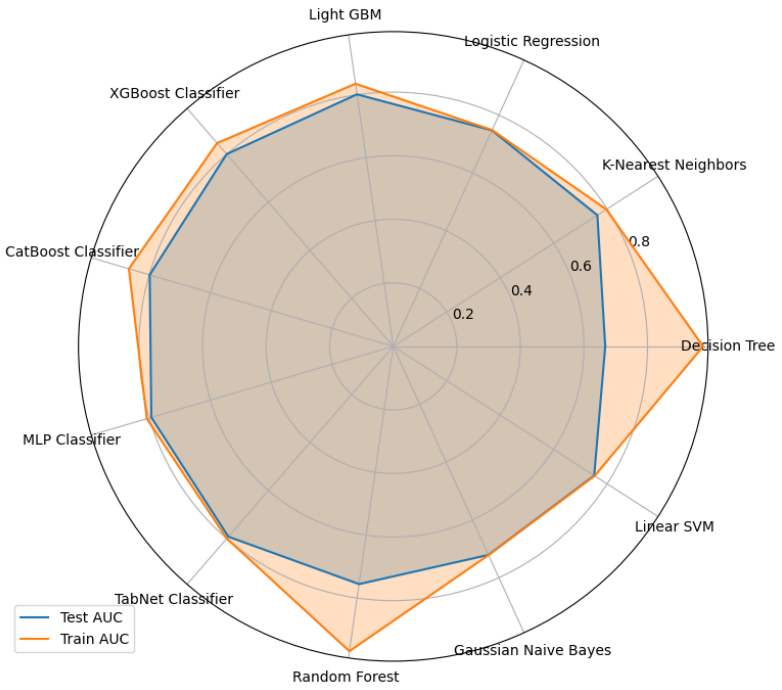


Fig. 4. Analysis of the AUC score for the training and test sets of each model used

Model	Train					
	Accuracy	**Precision**	**Specificity**	**F1-score**	**AUC score**	**Time (sec.)**
Decision Tree	**0.958333**	**0.959635**	**0.922121**	**0.958984**	**0.974974**	3.267257
K-Nearest Neighbors	0.716940	0.709744	0.776718	0.713324	0.797081	74.509236
Logistic Regression	0.670055	0.630828	0.714945	0.649850	0.748468	9.158642
Light GBM	0.762174	0.759875	0.805486	0.761023	0.834851	31.024371
XGBoost Classifier	0.778482	0.775499	0.811930	0.776988	0.846249	11.245760
CatBoost Classifier	0.806087	0.803573	0.827197	0.804828	0.867066	30.655587
MLP Classifier	0.735344	0.729904	0.786039	0.732614	0.807940	216.282925
TabNet Classifier	0.726793	0.721770	0.779259	0.724273	0.799871	898.752648
Random Forest	**0.958333**	0.958341	0.918823	0.958337	0.968851	916.299046
Gaussian Naive Bayes	0.541402	0.466800	0.763862	0.501341	0.721748	**2.285743**
Linear SVM	0.670045	0.625760	0.712187	0.647146	0.753376	1,816.589914

Table 2
Results of evaluation metrics for train validation set for the main analyzed algorithms

Model	Test					
	Accuracy	**Precision**	**Specificity**	**F1-score**	**AUC score**	**Time (sec.)**
Decision Tree	0.650057	0.649379	0.752672	0.649718	0.667007	3.154011
K-Nearest Neighbors	0.693054	0.685357	0.762559	0.689184	0.763476	22.021935
Logistic Regression	0.668063	0.628419	0.714020	0.647635	0.747285	9.448737
Light GBM	**0.728803**	**0.725935**	**0.787206**	**0.727366**	**0.801207**	29.222903
XGBoost Classifier	0.726652	0.722683	0.783490	0.724665	0.800391	11.009313
CatBoost Classifier	0.726463	0.723133	0.784656	0.724794	0.798610	31.696878
MLP Classifier	0.721933	0.715672	0.778150	0.718789	0.793184	212.845502
TabNet Classifier	0.721819	0.716757	0.771291	0.719279	0.792242	898.881885
Random Forest	0.693620	0.692453	0.772221	0.693036	0.755623	775.686355
Gaussian Naive Bayes	0.541072	0.466456	0.763785	0.501001	0.721246	**2.387914**
Linear SVM	0.667950	0.613512	0.710953	0.644822	0.751633	1,746.591185

Table 3
Results of evaluation metrics for test validation set for the main analyzed algorithms

The voting strategy using the three gradient boosting algorithms has achieved better results than Light GBM alone. However, it only improves AUC by 0.001, while the processing time increased by 126%.

Tables 5 and 6 show a detailed AUC performance and processing time for the validation set using a bagging ensemble strategy with different estimators for MLP/CatBoost, and Light GBM/XGBoost Classifiers, respectively. Fig. 6 and 7 show the comparison of the AUC score and processing time, respectively, between the top 4 algorithms for different estimator values. We notice that the MLP processing time increases exponentially when the number of estimators increases, while the AUC value hardly improves. Hence, we stopped the experiments at 500 bagging estimators. The CatBoost Classifier achieves better AUC score and processing time performance than MLP, but it is still worse compared to Light GBM and XGBoost. The processing time for Light GBM and XGBoost Classifiers demonstrates a gradual increase. Although XGBoost is superior to Light GBM, by a value of 0.0002 AUC score. Light GBM has a better processing time on average, whose difference becomes more pronounced as we increase the number of estimators. In Fig. 8, the AUC score values are depicted across a range of 1 to 500 estimators for ease of viewing. The graph reveals an inflection point, or an "elbow," at around 200 estimators for both, Light GBM and XGBoost. The processing time of XGBoost at that point, although it doubles the processing time of Light GBM it is negligible (less than 894 seconds). Hence, we use Bagging XGBoost with 200 estimators as the optimal choice for our model. For use cases with higher constrains in processing time, Light GBM could be a better choice.

Then, we proceed to perform the ablation study on the different datasets we defined in Section 4.1 with XGBoost with 200 estimators. Fig. 9, shows the value of the AUC score for the five datasets, along with a comparison with the dataset containing all the 35 variables defined in Table 1, which we will refer to as "full dataset". Fig. 10, shows the corresponding ROC curves, and Table 7 shows the detailed results. We can observe that the Entry context + Base dataset (in purple) with a value of 0.752177 provides the most information to the problem in terms of AUC performance. It is followed by the Socio-Economic

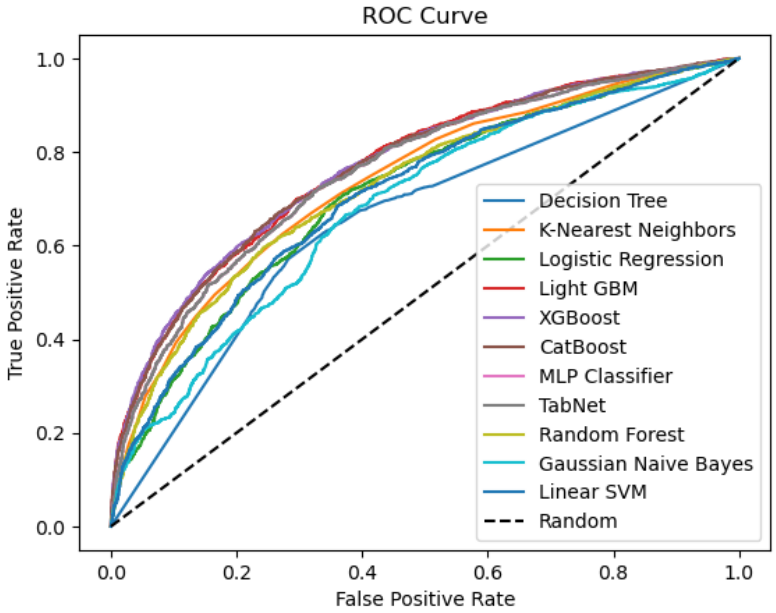


Fig. 5. ROC curve for the eleven machine learning algorithms used

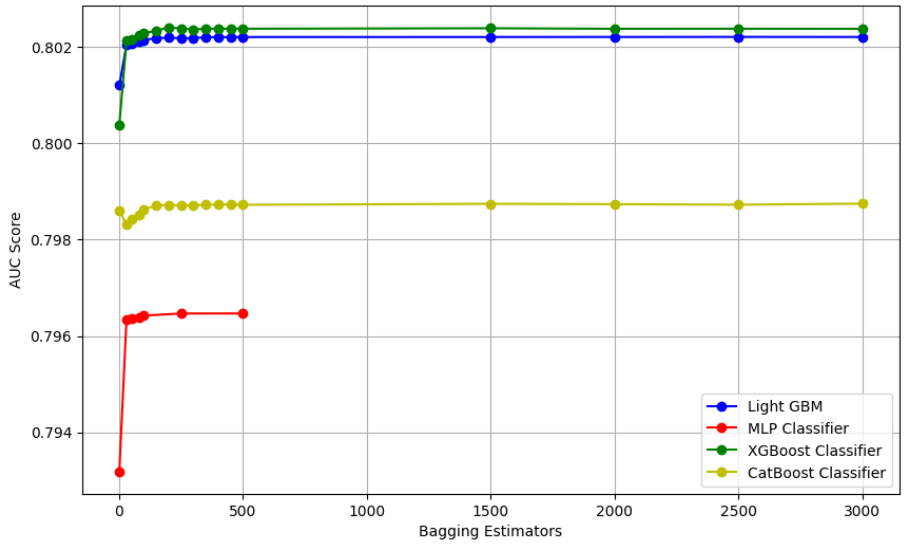


Fig. 6. AUC score comparison: Light GBM, MLP, XGBoost and CatBoost Classifiers. We run MLP Classifier only from 1 to 500 estimators because the improvement in AUC when augmenting the estimators was negligible, while the processing time increase exponentially

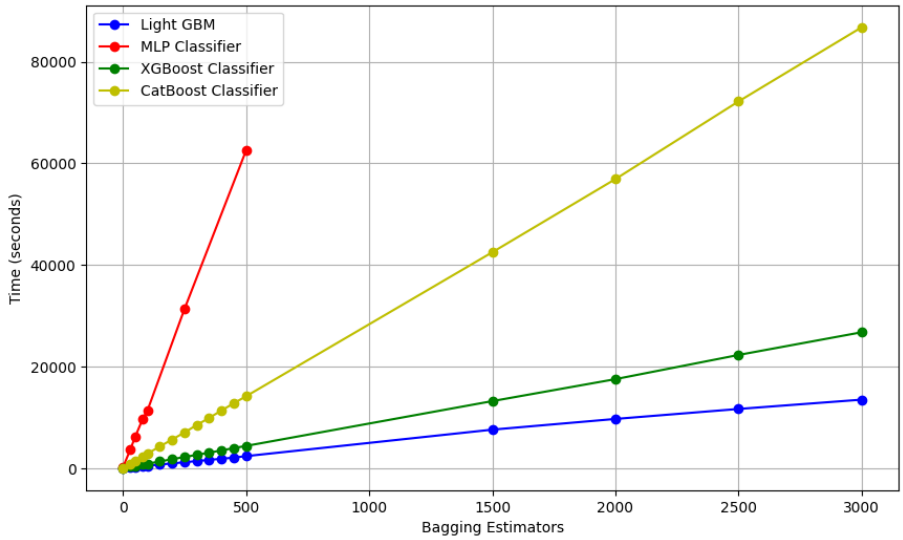


Fig. 7. Time comparison: Light GBM, MLP, XGBoost and CatBoost Classifiers. We run MLP Classifier only from 1 to 500 estimators because the improvement in AUC when augmenting the estimators was negligible, while the processing time increase exponentially

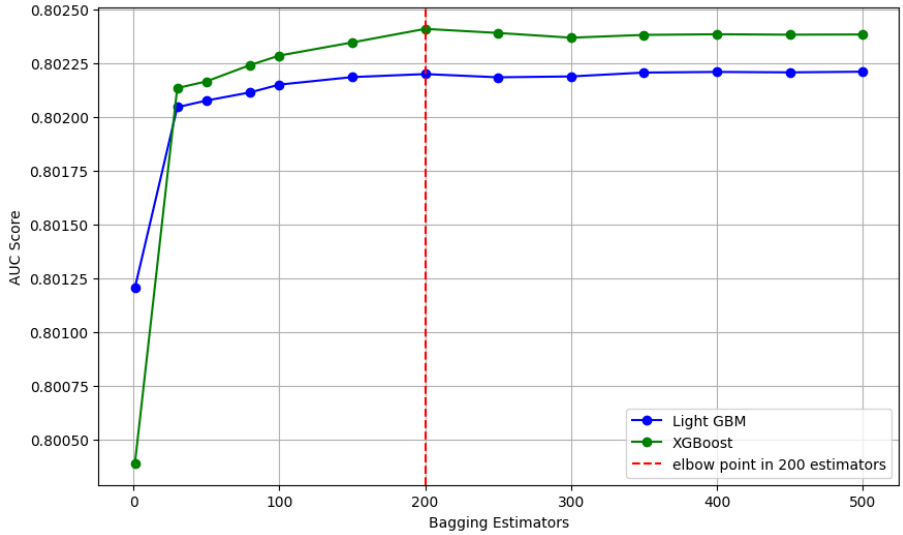


Fig. 8. Elbow Point for Light GBM and XGBoost models in 200 bagging estimators

Models	Ensemble Strategy	Weights	Meta Classifier	Test	
				AUC score	Time (sec.)
Light GBM + MLP Classifier	Voting	(50,50)	-	0.800563	251.985319
Light GBM + MLP Classifier	Voting	(70,30)	-	0.801654	251.753127
Light GBM + MLP Classifier	Voting	(30,70)	-	0.798382	252.117382
Light GBM + XGBoost Classifier	Voting	(50,50)	-	0.801786	38.766567
Light GBM + XGBoost Classifier	Voting	(70,30)	-	0.801819	38.615071
Light GBM + CatBoost Classifier	Voting	(50,50)	-	0.801958	57.349756
Light GBM + CatBoost Classifier	Voting	(70,30)	-	0.802192	57.267933
XGBoost Classifier + CatBoost Classifier	Voting	(50,50)	-	0.801727	39.244806
XGBoost Classifier + CatBoost Classifier	Voting	(70,30)	-	0.801775	38.833770
XGBoost Classifier + CatBoost Classifier	Voting	(30,70)	-	0.800954	39.590995
Light GBM + MLP Classifier + KNN	Voting	(33,33,33)	-	0.795801	267.582535
Light GBM + MLP Classifier + KNN	Voting	(60,30,10)	-	0.800964	267.642321
Light GBM + XGBoost + CatBoost	Voting	(33,33,33)	-	0.800966	67.303823
Light GBM + XGBoost + CatBoost	Voting	(40,30,30)	-	**0.802460**	66.017960
Light GBM + XGBoost + CatBoost	Voting	(40,20,40)	-	0.802397	68.945931
Light GBM + XGBoost + CatBoost	Voting	(50,30,20)	-	0.802422	67.708285
Light GBM + XGBoost + CatBoost	Voting	(60,30,10)	-	0.802217	67.008461
Light GBM + MLP Classifier	Stacking	-	RandomForest	0.752192	2,209.391480
Light GBM + MLP Classifier	Stacking	-	Light GBM	0.777118	2,212.529160
Light GBM + XGBoost + CatBoost	Stacking	-	Light GBM	**0.780084**	693.482353
Light GBM + XGBoost + CatBoost	Stacking	-	XGBoost	0.778326	697.756069
Light GBM + XGBoost + CatBoost	Stacking	-	CatBoost	0.769178	714.536195
Light GBM + MLP Classifier + KNN	Stacking	-	Light GBM	0.776217	2,814.143213
Light GBM + MLP Classifier + KNN + RandomForest	Stacking	-	Light GBM	0.776080	6,647.374783

Table 4
Other ensemble strategies analysis for algorithms with the best results

context + Base dataset with a value of 0.707177.

In order to approach the maximum AUC score value achieved by the XGBoost algorithm when using all variables (0.802409), we combined both datasets. We also performed a correlation analysis to identify the most relevant variables of both datasets contributing the most information. Fig. 11, displays the variables present in the Entry Context and Socio-Economic datasets, along with their correlation with the dependent variable. After analyzing this figure, we decided to eliminate variables with correlations falling within the range of [-0.09, 0.09], meaning with less than 10% correlation. Consequently, we have removed the variables: population, current_entry_hour_afternoon, and current_entry_hour_dawn.

In Fig. 12 we present a bar chart, again with the AUC values, for the two datasets that achieved the best results: Entry and Socio-Economic context (+ Base). Additionally, we introduce a combined dataset (Entry Context + Socio-Economic + Base) containing all 10 variables from the original datasets. Furthermore, we have created a simplified (reduction) combined dataset based on the correlation analysis, which includes the following 7 variables from the Entry Context, Socio-Economic and Base datasets: total_entries, avg_visit, km_to_dest, avg_gross_income, current_entry_east, current_entry_hour_morning, and current_entry_hour_night. We present the ROC curve for these improved datasets in Fig. 13.

Finally, Table 7 compares the AUC score values, number of variables, and processing time of each dataset. It can be observed that the combination of context datasets formed by Entry Context, Socio-Economic, and Base with 10 variables reaches a close value (a difference of 0.01) to the use of the full dataset. Furthermore, if we use the reduced version, denoted "Socio-Economic + Entry Context + Base (reduction)", we observe a 22.2% reduction in processing time and a reduction of 80% in the number of variables used (from 35 to 7) with respect to the full dataset. This combination represents only a loss of 0.0121 in the maximum AUC score obtained with the model.

Our work has some limitations related to bias in training. As mentioned in Section 4.3, we have excluded vehicles with a stay time of less than 3.5 hours. This causes the model to be unable to make predictions at the precise moment of the vehicle's entry into the area but with a delay of at least 3.5 hours from its arrival. Adding these vehicles

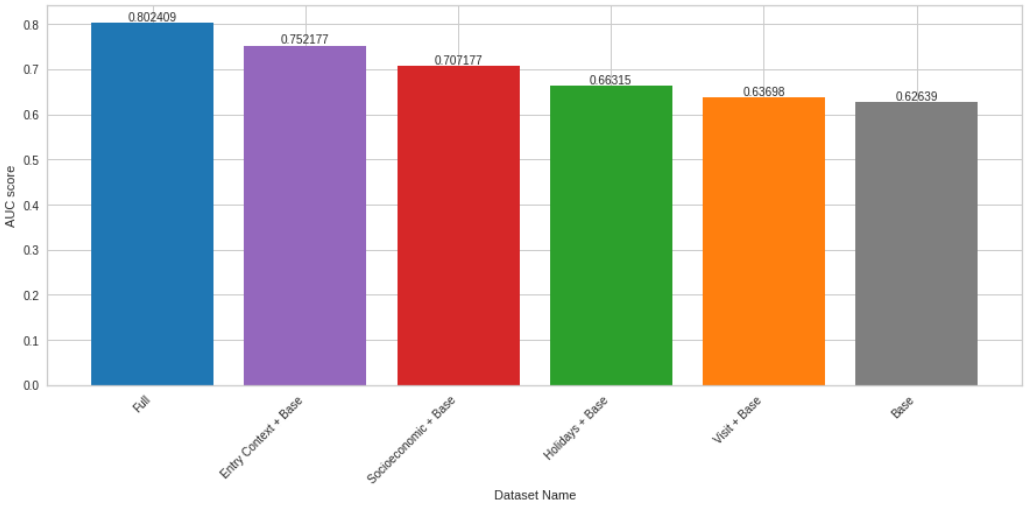


Fig. 9. Bar chart with the value of AUC obtained for each defined dataset using Bagging XGBoost with 200 estimators

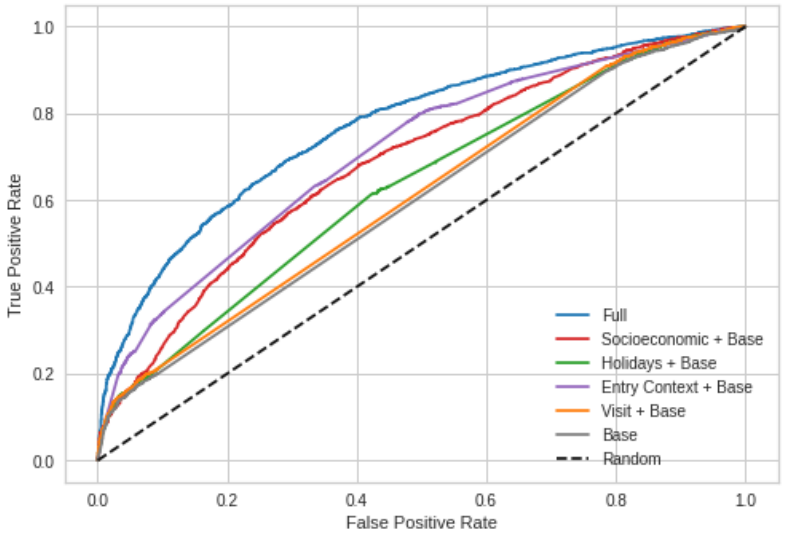


Fig. 10. ROC curve for the defined datasets using Bagging XGBoost with 200 estimators

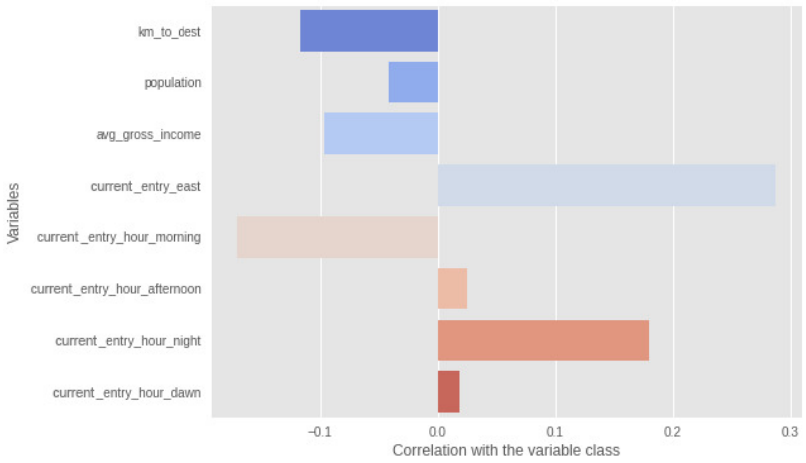


Fig. 11. Correlation analysis for variables in the Entry Context and Socio-Economic datasets

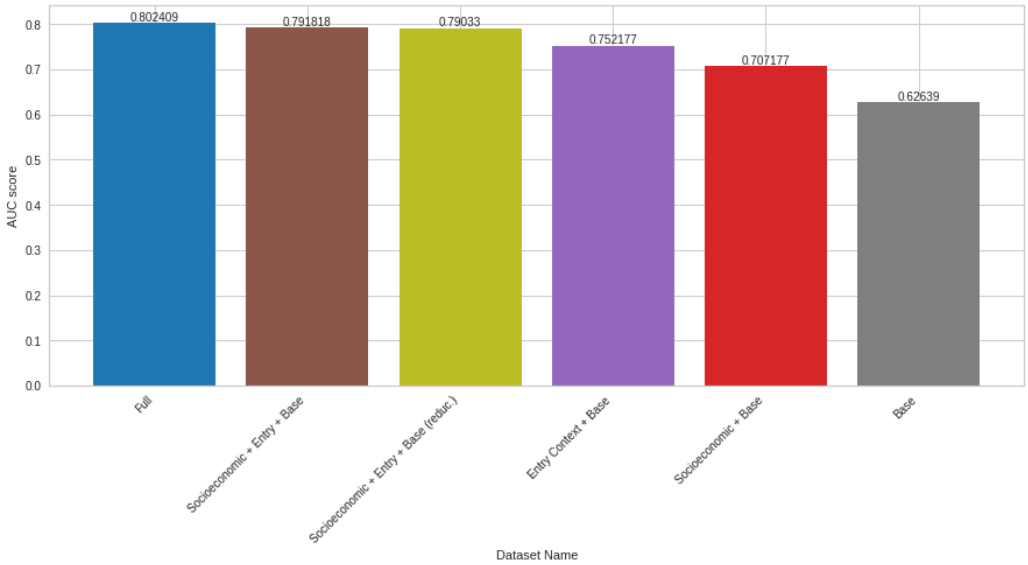


Fig. 12. Bar chart with the AUC value obtained for the detailed correlation study using Bagging XGBoost with 200 estimators

Model	Bagging Estimators	Test	
		AUC score	Time (sec.)
MLP Classifier	1	0.793184	212.845502
Bagging MLP Classifier	30	0.796351	3,616.870512
Bagging MLP Classifier	50	0.796359	6,164.704876
Bagging MLP Classifier	80	0.796375	9,611.036603
Bagging MLP Classifier	100	0.796423	11,358.486290
Bagging MLP Classifier	250	0.796466	31,299.27071
Bagging MLP Classifier	500	**0.796467**	62,506.932641
CatBoost Classifier	1	0.798610	31.696878
Bagging CatBoost	30	0.798313	878.978161
Bagging CatBoost	50	0.798423	1,439.435556
Bagging CatBoost	80	0.798525	2,266.095045
Bagging CatBoost	100	0.798630	2,857.258595
Bagging CatBoost	150	0.798714	4,257.845871
Bagging CatBoost	200	0.798723	5,631.068212
Bagging CatBoost	250	0.798711	7,011.509500
Bagging CatBoost	300	0.798716	8,482.161146
Bagging CatBoost	350	0.798725	9,873.686051
Bagging CatBoost	400	0.798727	11,327.662459
Bagging CatBoost	450	0.798733	12,761.623385
Bagging CatBoost	500	0.798725	14,139.715798
Bagging CatBoost	1500	0.798745	42,504.386149
Bagging CatBoost	2000	0.798738	56,916.862839
Bagging CatBoost	2500	0.798727	72,213.213867
Bagging CatBoost	3000	**0.798749**	86,811.868792

Table 5
Ensemble bagging analysis for MLP and CatBoost Classifier.

resulted in a 0.05-point deterioration in the AUC score. Nevertheless, the decision regarding the time threshold was made, balancing the model's performance and sufficient time to enable policymakers to make quick decisions on the ground. We believe that this 3.5-hour window allows flexibility in making decisions.

6. Conclusion

This paper presents a machine learning model for tourist prediction, focusing on a classification algorithm designed to forecast the number of nights a vehicle stays in a rural tourist region located in high mountain areas. The LPR sensor data is enriched with contextual information, such as the owners' residence location. Defining datasets and storing vehicle visit history enables predicting the number of nights with a few hours of delay from vehicle entry. Additionally, we conduct an ablation study, exploring datasets derived from various data sources and expert knowledge, analytically demonstrating the value each dataset contributes to the model. The results reveal a significant reduction of 22.2% in processing time and an 80% decrease in the number of variables used, compared to applying the model on all variables and rendering only a 0.01 loss of the AUC score. Our work is useful for scientists developing predictive models in the field of tourism. By identifying the most important databases, our results guide them in strategically allocating their resources to obtain and handle specific datasets. This becomes particularly advantageous when they encounter resource constraints concerning finances and time allocation for a given project. Our proposal could be extended with additional information from other datasets.

CRediT authorship contribution statement

Daniel Bolaños-Martinez: Methodology, Validation, Investigation, Resources, Software, Writing - Original Draft. **Jose Luis Garrido:** Conceptualization, Investigation, Resources, Writing- Review & Editing, Super-

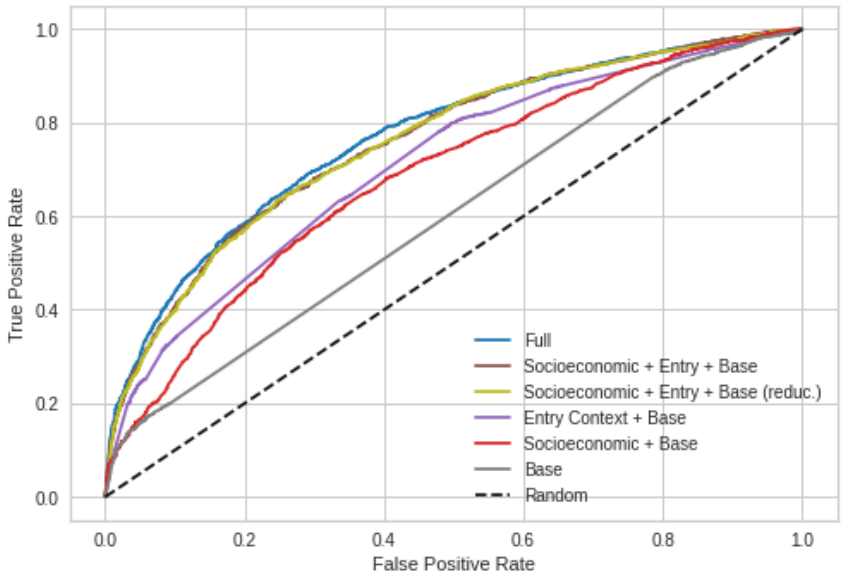


Fig. 13. ROC curve for the detailed datasets using Bagging XGBoost with 200 estimators

vision. **Maria Bermudez-Edo:** Conceptualization, Investigation, Methodology, Resources, Writing- Review & Editing, Supervision.

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.

Consent for publication

All authors agreed with the content and that all gave explicit consent to submit.

Ethics approval and consent to participate

Before installing cameras, we secured agreements with authorities and the installation company to comply with laws. LPR cameras send anonymized data to a secure server. Except for DGT data, all other variables are public. DGT only shared sensitive data for research purposes and under an agreement. This information is stored encrypted, and it is accessible only to authorized researchers.

Data Availability Statement

The dataset used to support the results is not available at this time, but we plan to publish them in the near future.

Acknowledgments

This publication is part of the grant PID2023-149185OB-I00 funded by MICIU/AEI/10.13039/501100011033 and by ERDF/EU; and also by R&D&i Project Ref. C-SEJ-128-UGR23, co-financed by the Regional Ministry of University, Research and Innovation and by the European Union under the Andalusia ERDF Program 2021-2027.

Model	Bagging Estimators	Test	
		AUC score	Time (sec.)
Light GBM	1	0.801207	29.222903
Bagging Light GBM	30	0.802045	126.960913
Bagging Light GBM	50	0.802076	220.738479
Bagging Light GBM	80	0.802114	332.957847
Bagging Light GBM	100	0.802150	464.916159
Bagging Light GBM	150	0.802185	694.649814
Bagging Light GBM	200	0.802199	921.147756
Bagging Light GBM	250	0.802184	1,152.046603
Bagging Light GBM	300	0.802188	1,382.276393
Bagging Light GBM	350	0.802206	1,612.056429
Bagging Light GBM	400	0.802209	1,841.251040
Bagging Light GBM	450	0.802207	2,060.768677
Bagging Light GBM	500	0.802210	2,354.839286
Bagging Light GBM	1500	0.802211	7,571.803962
Bagging Light GBM	2000	**0.802214**	9,682.179090
Bagging Light GBM	2500	0.802213	11,647.081261
Bagging Light GBM	3000	0.802212	13,480.760810
XGBoost Classifier	1	0.800391	11.009313
Bagging XGBoost	30	0.802134	269.519583
Bagging XGBoost	50	0.802165	438.245013
Bagging XGBoost	80	0.802241	698.889167
Bagging XGBoost	100	0.802285	865.735364
Bagging XGBoost	150	0.802346	1,302.160010
Bagging XGBoost	200	**0.802409**	1,814.137277
Bagging XGBoost	250	0.802390	2,178.676117
Bagging XGBoost	300	0.802368	2,607.685399
Bagging XGBoost	350	0.802381	3,037.155903
Bagging XGBoost	400	0.802384	3,477.172626
Bagging XGBoost	450	0.802382	3,936.966248
Bagging XGBoost	500	0.802383	4,365.816685
Bagging XGBoost	1500	0.802393	13,190.436564
Bagging XGBoost	2000	0.802382	17,518.338156
Bagging XGBoost	2500	0.802383	22,283.368221
Bagging XGBoost	3000	0.802381	26,753.253852

Table 6
Ensemble bagging analysis for Light GBM and XGBoost Classifier

Dataset Name	AUC score	Num. Variables	Time (sec.)
Basic Datasets			
Full	0.802409	35	1,814.137277
Entry Context + Base	0.752177	7	1,187.963586
Socio-Economic + Base	0.707177	5	1,399.354359
Holidays + Base	0.663150	18	1,386.297427
Visit + Base	0.636980	11	1,396.513810
Base	0.626390	2	1,157.062184
Improved Datasets			
Socio-Economic + Entry Context + Base	0.791818	10	1,424.717826
Socio-Economic + Entry Context + Base (reduction)	0.790330	7	1,411.681485

Table 7
AUC score, number of variables and processing time for the different datasets using XGBoost model with bagging and 200 estimators

References

[1] H. Laaroussi, F. Guerouate, et al., Deep learning framework for forecasting tourism demand, in: 2020 IEEE international conference on technology management, operations and decisions (ICTMOD), IEEE, 2020, pp. 1–4.

[2] F. T. Sáenz, F. Arcas-Tunez, A. Muñoz, Nation-wide touristic flow prediction with graph neural networks and heterogeneous open data, Information Fusion 91 (2023) 582–597.

[3] Z. Zhai, P. Liu, L. Zhao, J. Qian, B. Cheng, An efficiency-enhanced deep learning model for city-wide crowd flows prediction, International Journal of Machine Learning and Cybernetics 12 (2021) 1879–1891.

[4] M. Lin, X. Zhao, Application research of neural network in vehicle target recognition and classification, in: 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), IEEE, 2019, pp. 5–8.

[5] Z. Ning, J. Huang, X. Wang, Vehicular fog computing: Enabling real-time traffic management for smart cities, IEEE Wireless Communications 26 (1) (2019) 87–93.

[6] W. Yao, C. Chen, H. Su, N. Chen, S. Jin, C. Bai, Analysis of key commuting routes based on spatiotemporal trip chain, Journal of Advanced Transportation 2022 (2022).

[7] Z. Liu, Y. Liu, Q. Meng, Q. Cheng, A tailored machine learning approach for urban transport network flow estimation, Transportation Research Part C: Emerging Technologies 108 (2019) 130–150.

[8] O. Cats, F. Ferranti, Unravelling individual mobility temporal patterns using longitudinal smart card data, Research in Transportation Business & Management 43 (2022) 100816.

[9] M. A. Mondal, Z. Rehena, Identifying traffic congestion pattern using k-means clustering technique, in: 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), IEEE, 2019, pp. 1–5.

[10] M. L. M. Peixoto, A. H. Maia, E. Mota, E. Rangel, D. G. Costa, D. Turgut, L. A. Villas, A traffic data clustering framework based on fog computing for vanets, Vehicular Communications 31 (2021) 100370.

[11] D. Buhalis, Technology in tourism-from information communication technologies to etourism and smart tourism towards ambient intelligence tourism: a perspective article, Tourism Review 75 (1) (2020) 267–272.

[12] J. Tang, J. Zeng, Y. Wang, H. Yuan, F. Liu, H. Huang, Traffic flow prediction on urban road network based on license plate recognition data: combining attention-lstm with genetic algorithm, Transportmetrica A: Transport Science 17 (4) (2021) 1217–1243.

[13] J. Tang, J. Zeng, Spatiotemporal gated graph attention network for urban traffic flow prediction based on license plate recognition data, Computer-Aided Civil and Infrastructure Engineering 37 (1) (2022) 3–23.

[14] G. Yang, D. Coble, C. Vaughan, C. Peele, A. Morsali, G. F. List, D. J. Findley, Waiting time estimation at ferry terminals based on license plate recognition, Journal of Transportation Engineering, Part A: Systems 148 (9) (2022) 04022064.

[15] W. Yao, J. Yu, Y. Yang, N. Chen, S. Jin, Y. Hu, C. Bai, Understanding travel behavior adjustment under covid-19, Communications in Transportation Research (2022) 100068.

[16] P. Wang, J. Lai, Z. Huang, Q. Tan, T. Lin, Estimating traffic flow in large road networks based on multi-source traffic data, IEEE Transactions on Intelligent Transportation Systems 22 (9) (2020) 5672–5683.

[17] Q. Liu, J. Zhang, J. Liu, Z. Yang, Feature extraction and classification algorithm, which one is more essential? an experimental study on a specific task of vibration signal diagnosis, International Journal of Machine Learning and Cybernetics (2022) 1–12.

[18] R. Meyes, M. Lu, C. W. de Puiseau, T. Meisen, Ablation studies in artificial neural networks, arXiv preprint arXiv:1901.08644 (2019).

[19] J. A. Gómez-Pulido, J. M. Romero-Muelas, J. M. Gómez-Pulido, J. L. Castillo Sequera, J. Sanz Moreno, M.-L. Polo-Luque, A. Garcés-Jiménez, Predicting infectious diseases by using machine learning classifiers, in: I. Rojas, O. Valenzuela, F. Rojas, L. J. Herrera, F. Ortuño (Eds.), Bioinformatics and Biomedical Engineering, Springer International Publishing, Cham, 2020, pp. 590–599.

[20] B. Liu, J. Pei, Z. Yu, Stock price prediction through gra-wd-bilstm model with air quality and weather factors, International Journal of Machine Learning and Cybernetics (2023) 1–18.

[21] A. Maiti, S. Shi, S. Vucetic, An ablation study on the use of publication venue quality to rank computer science departments: Publication quality is strongly correlated with the subjective perception of research strength, Scientometrics 128 (8) (2023) 4197–4218.

[22] N. Saraswathi, T. S. Rooba, S. Chakaravarthi, Improving the accuracy of sentiment analysis using a linguistic rule-based feature selection method in tourism reviews, Measurement: Sensors 29 (2023) 100888.

[23] D. R. Anamisa, F. A. Mufarroha, A. Jauhari, Feature selection to increase the attractiveness of visitors in bangkalan tourism, madura based on chi-square method, in: AIP Conference Proceedings, Vol. 2679, AIP Publishing, 2023.

[24] S. Sun, M. Li, S. Wang, C. Zhang, Multi-step ahead tourism demand forecasting: The perspective of the learning using privileged information paradigm, Expert Systems with Applications 210 (2022) 118502.

[25] X. Zhan, R. Li, S. V. Ukkusuri, Link-based traffic state estimation and prediction for arterial networks using license-plate recognition data, Transportation Research Part C: Emerging Technologies 117 (2020) 102660.

[26] H. Song, H. Liu, Predicting tourist demand using big data, Analytics in smart tourism design: Concepts and methods (2017) 13–29.

[27] S. Peters, P. Keller, Applications and issues of big data in tourism research (2022).

[28] P. Madzík, L. Falát, L. Copuš, M. Valeri, Digital transformation in tourism: bibliometric literature review based on machine learning approach, European Journal of Innovation Management 26 (7) (2023) 177–205.

[29] T. Peng, J. Chen, C. Wang, Y. Cao, A forecast model of tourism demand driven by social network data, IEEE Access 9 (2021) 109488–109496.

[30] J.-W. Bi, Y. Liu, H. Li, Daily tourism volume forecasting for tourist attractions, Annals of Tourism Research 83 (2020) 102923.

[31] B. P. L. Lau, S. H. Marakkalage, Y. Zhou, N. U. Hassan, C. Yuen, M. Zhang, U.-X. Tan, A survey of data fusion in smart city applications, Information Fusion 52 (2019) 357–374.

[32] D. Bolaños-Martinez, M. Bermudez-Edo, J. L. Garrido, Clustering pipeline for vehicle behavior in smart villages, Information Fusion (2023) 102164.

[33] D. Bolaños-Martinez, M. Bermudez-Edo, J. L. Garrido, Clustering study of vehicle behaviors using license plate recognition, in: Proceedings of the International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2022), Springer, 2022, pp. 784–795.

[34] L. Zheng, H. Wang, S. Gao, Sentimental feature selection for sentiment analysis of chinese online reviews, International journal of machine learning and cybernetics 9 (2018) 75–84.

[35] C. Sun, H. Li, M. Song, D. Cai, B. Zhang, S. Hong, Adaptive model training strategy for continuous classification of time series, Applied Intelligence (2023) 1–19.

[36] B. Swaminathan, S. Palani, S. Vairavasundaram, Feature fusion based deep neural collaborative filtering model for fertilizer prediction, Expert Systems with Applications 216 (2023) 119441.

[37] Y. S. Abu-Mostafa, M. Magdon-Ismail, H.-T. Lin, Learning from data (2012).

[38] G. James, D. Witten, T. Hastie, R. Tibshirani, An introduction to statistical learning: with applications in r (2013).

[39] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 eighth ieee international conference on data mining, IEEE, 2008, pp. 413–422.

[40] H. Henderi, T. Wahyuningsih, E. Rahwanto, Comparison of min-max normalization and z-score normalization in the k-nearest neighbor (knn) algorithm to test the accuracy of types of breast cancer, International Journal of Informatics and Information Systems 4 (1) (2021) 13–20.

[41] S. Patro, K. K. Sahu, Normalization: A preprocessing stage, arXiv preprint arXiv:1503.06462 (2015).

[42] A. S. Eesa, W. K. Arabo, A normalization methods for backpropagation: a comparative study, Science Journal of University of Zakho 5 (4) (2017) 319–323.

[43] S. Bektaş, Y. Şişman, The comparison of l1 and l2-norm minimization methods, International Journal of the Physical Sciences 5 (11) (2010) 1721–1727.

[44] L. Mendoza-Pittí, J. M. Gómez-Pulido, M. Vargas-Lombardo, J. A. Gómez-Pulido, M.-L. Polo-Luque, D. Rodréguez-Puyol, Machine-learning model to predict the intradialytic hypotension based on clinical-analytical data, IEEE Access 10 (2022) 72065–72079.

[45] O. Gutiérrez, J. C. Sancho Núñez, M. Homaei, J. Díaz, Aplicación de técnicas de reducción de dimensionalidad y balanceo en ciberseguridad, 2022.

[46] E. E. Misengo, D. D. Prastyo, H. Kuswanto, Modeling and forecasting monthly tourist arrivals to the united states and indonesia using arima hybrids of multilayer perceptron models, in: AIP Conference Proceedings, Vol. 2540, AIP Publishing, 2023.

[47] S. JATMIKA, S. PATMANTHARA, A. P. WIBAWA, The model of local wisdom for smart wellness tourism with optimization multilayer perceptron, Journal of Theoretical and Applied Information Technology 102 (2) (2024).

[48] J. Ali, R. Khan, N. Ahmad, I. Maqsood, Random forests and decision trees, International Journal of Computer Science Issues (IJCSI) 9 (5) (2012) 272.

[49] N. Ariyani, A. Fauzi, F. Umar, Predicting and determining antecedent factors of tourist village development using naive bayes and tree algorithm, International Journal of Applied Sciences in Tourism and Events 7 (1) (2023) 1–15.

[50] L. Peng, L. Wang, X.-Y. Ai, Y.-R. Zeng, Forecasting tourist arrivals via random forest and long short-term memory, Cognitive Computation 13 (2021) 125–138.

[51] N. Celiker, C. O. Guzeller, Predicting organizational citizenship behaviour in hospitality businesses with decision tree method, International Journal of Hospitality & Tourism Administration 25 (2) (2024) 436–474.

[52] L. E. Peterson, K-nearest neighbor, Scholarpedia 4 (2) (2009) 1883.

[53] E. H. Rachmawanto, C. A. Sari, H. Pramono, W. S. Sari, Visitor prediction decision support system at dieng tourism objects using the k-nearest neighbor method, Journal of Applied Intelligent System 7 (2) (2022) 183–192.

[54] D. R. Anamisa, A. Jauhari, F. A. Mufarroha, K-nearest neighbors method for recommendation system in bangkalan's tourism, ComTech: Computer, Mathematics and Engineering Applications 14 (1) (2023) 33–44.

[55] P. Tsangaratos, I. Ilia, Comparison of a logistic regression and naïve bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size, Catena 145 (2016) 164–179.

[56] H. Siroosi, G. Heshmati, A. Salmanmahiny, Can empirically based model results be fed into mathematical models? mce for neural network and logistic regression in tourism landscape planning, Environment, Development and Sustainability 22 (4) (2020) 3701–3722.

[57] D. Devianto, S. Maryati, H. Rahman, Logistic regression model for entrepreneurial capability factors in tourism development of the rural areas with bayesian inference approach, in: Journal of Physics: Conference Series, Vol. 1940, IOP Publishing, 2021, p. 012022.

[58] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, Advances in neural information processing systems 30 (2017).

[59] D. Zhao, Z. Hu, Y. Yang, Tourist trajectory prediction based on improved lightgbm, in: International Conference on Statistics, Data Science, and Computational Intelligence (CSDSCI 2022), Vol. 12510, SPIE, 2023, pp. 54–59.

[60] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[61] J. Kang, X. Guo, L. Fang, X. Wang, Z. Fan, Integration of internet search data to predict tourism trends using spatial-temporal xgboost composite model, International Journal of Geographical Information Science 36 (2) (2022) 236–252.

[62] Y. Hu, L. Shao, L. La, H. Hua, Using investor and news sentiment in tourism stock price prediction based on xgboost model, in: 2021 IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science (BCD), IEEE, 2021, pp. 20–24.

[63] H. Li, H. Gao, H. Song, Tourism forecasting with granular sentiment analysis, Annals of Tourism Research 103 (2023) 103667.

[64] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, A. Gulin, Catboost: unbiased boosting with categorical features, Advances in neural information processing systems 31 (2018).

[65] Y. Chen, C. Ding, H. Ye, Y. Zhou, Comparison and analysis of machine learning models to predict hotel booking cancellation, in: 2022 7th International Conference on Financial Innovation and Economic Development (ICFIED 2022), Atlantis Press, 2022, pp. 1363–1370.

[66] J. Tang, J. Cheng, M. Zhang, Forecasting airbnb prices through machine learning, Managerial and Decision Economics 45 (1) (2024) 148–160.

[67] L. K. Hansen, P. Salamon, Neural network ensembles, IEEE transactions on pattern analysis and machine intelligence 12 (10) (1990) 993–1001.

[68] S. Ö. Arik, T. Pfister, Tabnet: Attentive interpretable tabular learning, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 35, 2021, pp. 6679–6687.

[69] S. Kim, W. Shin, H.-W. Kim, Predicting online customer purchase: The integration of customer characteristics and browsing patterns, Decision Support Systems 177 (2024) 114105.

[70] D. Hermanto, M. Ziaurrahman, M. Bianto, A. Setyanto, Twitter social media sentiment analysis in tourist destinations using algorithms naive bayes classifier, in: Journal of Physics: Conference Series, Vol. 1140, IOP Publishing, 2018, p. 012037.

[71] T. Joachims, Training linear svms in linear time, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, pp. 217–226.

[72] E. Purnaningrum, M. Athoillah, Svm approach for forecasting international tourism arrival in east java, in: Journal of Physics: Conference Series, Vol. 1863, IOP Publishing, 2021, p. 012060.

[73] D. A. Otchere, T. O. A. Ganat, R. Gholami, S. Ridha, Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: Comparative analysis of ann and svm models, Journal of Petroleum Science and Engineering 200 (2021) 108182.

[74] G. Bonaccorso, Machine learning algorithms: Popular algorithms for data science and machine learning (2018).

[75] L. Breiman, Bagging predictors, Machine learning 24 (1996) 123–140.

[76] G. Sigletos, G. Paliouras, C. D. Spyropoulos, M. Hatzopoulos, W. Cohen, Combining information extraction systems using voting and stacked generalization., Journal of Machine Learning Research 6 (11) (2005).

[77] T. G. Dietterich, Ensemble methods in machine learning, in: International workshop on multiple classifier systems, Springer, 2000, pp. 1–15.

[78] J. A. Hanley, B. J. McNeil, The meaning and use of the area under a receiver operating characteristic (roc) curve., Radiology 143 (1) (1982) 29–36.

[79] A. Gupta, N. Tatbul, R. Marcus, S. Zhou, I. Lee, J. Gottschlich, Class-weighted evaluation metrics for imbalanced data classification, arXiv preprint arXiv:2010.05995 (2020).

[80] L. A. Jeni, J. F. Cohn, F. De La Torre, Facing imbalanced data–recommendations for the use of performance metrics, in: 2013 Humaine association conference on affective computing and intelligent interaction, IEEE, 2013, pp. 245–251.

4

SASD: SELF-ATTENTION FOR SMALL DATASETS - A CASE STUDY IN SMART VILLAGES

[C] Bolaños-Martinez, D., Durán-López, A., Garrido, J. L., Delgado-Márquez, B., & Bermudez-Edo, M. (2025). SASD: Self-Attention for Small Datasets—A case study in smart villages. Expert Systems with Applications, 271, 126245.

DOI: 10.1016/j.eswa.2024.126245.

- Status: Published.
- Impact Factor (JCR SCIE 2023): 7.5.
- Category: Computer Science, Artificial Intelligence. Rank: 24 / 197 (JIF Q1).
- Category: Engineering, Electrical & Electronic. Rank: 25 / 353 (JIF D1).
- Category: Operations Research & Management Science. Rank: 6 / 106 (JIF D1).
- Number of citations: 0 (Source, Google Scholar).
- Attention score: 9 (Source, Altmetric).
  Mentioned by:
  - 8 X users.
  - 1 Redditors.
  - 1 Bluesky users.

  Readers on:
  - 11 Mendeley.
- Open source data/software: [iii], [v], [vi].

SASD: Self-Attention for Small Datasets - A Case Study in Smart Villages

Daniel Bolaños-Martinez[a,b,*], Alberto Durán-López[a,b], Jose Luis Garrido[a,b], Blanca Delgado-Márquez[c], Maria Bermudez-Edo[a,b]

[a]*Research Centre for Information and Communication Technologies (CITIC-UGR), University of Granada, Granada, Spain.*
[b]*Department of Software Engineering, Computer Science School, University of Granada, C/ Periodista Daniel Saucedo Aranda S/N, 18071 Granada, Spain.*
[c]*Department of Business Management and European Institute of Sustainability Management, University of Granada, Campus of Cartuja, 18071 Granada, Spain.*

Abstract

Understanding repeat visitation patterns in tourism is important for optimizing economic benefits, as loyal visitors significantly contribute to the stability and growth of destinations. However, this area remains underexplored, especially in smart villages where data limitations challenge traditional machine learning (ML) approaches. Although neural networks (NN) have proven effective in various research fields, they struggle with small datasets. We propose a ML application for tracing repeat visitors using NN suitable for small datasets. Specifically, we designed SASD (Self-Attention for Small Dataset), a deep learning architecture that incorporates self-attention layers to address data limitations. We applied SASD to predict tourists' visit intentionality in the next 12 months in a smart village region, using as training data, information from License Plate Recognition sensors, and questionnaires. We evaluated its performance against various ML algorithms; Decision Trees, Random Forests, K-NN, Logistic Regression, Gradient Boosting, Naive Bayes, SVM, MLP, RNN, and LSTM, TabNet and TabTransformer. Our results demonstrate greater accuracy, recall, precision, and F1-score. Specifically, SASD outperforms other models by up to 3% on the weighted average F1 score. Our results also confirm that in NN, the incorporation of self-attention layers accelerates convergence and reduces processing time by 32%. The best results are achieved with two self-attention layers placed at the beginning and end of the NN. Our results provide insights for policymakers, business managers, local communities, and environmental organizations, enabling informed decisions and optimal resource allocation for tourism development.

Keywords: Self-Attention, Deep Learning, Internet of Things, Tourism Development, Repeat Tourism, Sensors

1. Introduction

Tracing repeat tourism involves identifying tourists who return to the same destination one or multiple times [1]. Repeat tourists tend to show higher levels of satisfaction and loyalty, which translates into greater economic benefits for destinations through consistent spending and positive word-of-mouth promotion [2, 3]. In addition, repeat visitors can contribute to more stable demand, which assists policymakers in infrastructure planning and resource allocation [4]. In parallel, the concept of smart villages - rural communities that use digital technologies to improve the lives of their people and the economy - has gained attention [5]. By applying machine learning (ML) techniques, researchers can predict the number of individuals that revisit an area and thus improve resource

*Corresponding author.
Email addresses: danibolanos@ugr.es (Daniel Bolaños-Martinez), albduranlopez@ugr.es (Alberto Durán-López), jgarrido@ugr.es (Jose Luis Garrido), bdelgado@ugr.es (Blanca Delgado-Márquez), mbe@ugr.es (Maria Bermudez-Edo)

management in future events [6, 7]. We propose a ML application for tracing repeat visitors using neural networks for small datasets by incorporating self-attention layers to address data limitations.

Neural Networks (NN) have gained prominence in numerous research fields, such as education [8, 9], health [10, 11, 12] and tourism [13, 14, 15]. Although NNs were originally applied to image-related tasks [16, 17], they have since demonstrated effectiveness across a broader range of applications, including object recognition, text classification, and sound and video data processing [18, 19, 20]. Recent studies also show that, even with tabular data, NNs outperform classical algorithms [21, 22].

Usually, NNs work best when trained with a large amount of data. However, there are cases where the amount of data is small, but we want to take advantage of the performance of NN. Several tools can help NN to acquire knowledge from limited datasets, such as weight extraction [21] or attention mechanisms [23]. Attention-based models on small datasets have been tested in the field of image learning [24, 25]. While NN with attention mechanisms have also been developed for tabular data, such as numerical or alphanumeric data coming from sensors or questionnaires [26, 27], these solutions are not fully adapted to small datasets [28], which remains a limitation in our specific problem. Specifically, we propose **S**elf-**A**ttention for **S**mall **D**ataset (**SASD**), a NN that provides a solution to a tabular sparse data problem in the area of smart villages.

The rise in the installation of IoT sensors for building smart cities has increased the volume of data, and hence the analysis of such data, for example, in the field of vehicle monitoring [29, 30] using Global Positioning System (GPS) or License Plate Recognition (LPR) systems, and other smart devices [31, 32]. This trend has also been used recently in smart villages [5]. IoT derived data can be combined with context data or questionnaires [33, 34]. However, questionnaires are difficult and costly to obtain but remain necessary for predicting certain human behaviors [10, 35]. Their restriction lies in the cost of implementation (economic and temporal) because the sample of respondents may not be sufficient, presenting various challenges in constructing deep learning models on this data [35, 36]. We design the architecture as a classification model to predict the intentionality of returning in the next 12 months of visitors in a rural tourist area. We trained our model with a limited number of questionnaires and data from LPR sensors providing the vehicle behavior in the area. We use the information from the questionnaires to label the intentionality of future visits and use it as a variable to predict. We compare the performance of this model with several popular classification algorithms in the literature, including two versions of Recurrent Neural Networks (RNN), and two tabular data transformers (TabNet and TabTransformer). In addition, we apply specific optimization techniques on the SASD model, including variations in the number and position of the attention layers, and performing an ablation test on dropout and ReLU layers. We also try an alternative version that replaces the self-attention layers with multihead-attention layers, the **M**ultihead-**A**ttention **S**mall **D**ataset (**MASD**) model.

Our results show that two self-attention layers, one at the beginning of the NN and one at the end, provide the best results. Comparative results demonstrate that SASD outperforms other popular classification algorithms in terms of precision, accuracy and F1-score, underscoring its value as an effective tool for anticipating tourists' visit intentions with scarce data. The MASD model performs worse than SASD model.

Over the last decade, tourism development strategies have increasingly emphasized sustainable tourism development, seeking to balance economic, environmental, and social benefits [37, 38]. However, recent research continues to stress the need for deeper exploration into frameworks and mechanisms that can effectively achieve this balance [39, 40]. Emerging solutions in sustainable tourism development integrate sensor data and machine learning to enhance visitor experiences, optimize resource use, minimize environmental impacts, and predict traffic flows and tourism demand [41, 42, 43]. Our interest in predicting repeat visitors using small datasets within a rural context aligns with this progressive shift toward sustainable tourism. By employing precise forecasting models to analyze tourist patterns, policymakers can optimize the allocation of public resources to support tourism growth while mitigating potential adverse impacts [44]. For business managers, accurate demand predictions facilitate efficient resource allocation [45], while local communities and environmental organizations gain a clearer understanding of the role that preserving cultural heritage and natural ecosystems plays in attracting repeat visitors

[46].

Traditional tourism development research has long underscored the role of domestic economic, financial, and political risks in shaping sustainable tourism. Political stability, for instance, positively impacts tourism revenues in less-developed countries, although the effect is less pronounced in developed nations [47]. Similarly, reducing economic, financial, and political risks is critical to attracting international tourists and fostering sustainable tourism development [48]. Geopolitical risks and currency depreciation negatively affect tourism in the short term, whereas favorable economic policies yield positive long-term outcomes [49]. While these factors are undeniably significant in shaping tourism patterns, our model controls their influence by focusing on a specific rural area in a developed country, making predictions under the assumption of relatively stable domestic conditions.

The remainder of the paper is organized as follows. Section 2 describes the related work. Section 3 presents the theoretical basis for each of the layers we use in our NN classification model. Section 4 explains the use case, including data collection methodology and the preprocessing of the dataset. Section 5 defines the proposed NN configuration and the experimental procedure. Section 6 shows the analysis and discussion of the results, and Section 7 concludes the paper.

2. Related work

Machine learning, and especially deep learning models, need a great amount of data, which is sometimes impossible to obtain. For small datasets, the depth of the network matters less than how the knowledge transfer is executed [23]. A few studies seek for solution in NN with limited training data [24, 25]. However, most examples in this area apply NN on image datasets. Some works opt to use transformers or incorporate layers of attention to improve knowledge transfer with images [50, 51]. A few works use NN models with scarce tabular data, outperforming other classical ML algorithms. For example, in [21], the authors combine a classification NN with two auxiliary networks in charge of producing the weights to be used by the first layer of the model. However, they do not use attention, which may be a more robust method for feature extraction [52].

With the rise of deep learning, many authors have employed artificial NNs such as BackPropagation Neural Network (BPNN), Long Short-Term Memory (LSTM), or Neural Network Autoregressive (NNAR) to build prediction models for tourist flows [53, 54, 55]. However, most of these articles rely on historical tourist flow data obtained from large public databases and do not consider other contextual factors [56]. A few studies include data obtained from LPR and/or GPS sensors in their NN models [57, 58], but do not incorporate contextual data. The fusion of sensor data with other data sources, such as geolocated social media data, information about holidays, or meteorological data [59, 60, 61], allows for the construction of more robust and efficient ML models.

In the tourism domain, some studies have integrated techniques to improve visitors' forecasting, such as attention mechanisms. [62] use Convolutional Neural Network (CNN) with multihead-attention. [63] developed a transformer-based model that utilizes an adaptive evolution algorithm to fine-tune the model's parameters. [64] introduce a transformer-based framework combining time series decomposition, temporal fusion transformers, and hyperparameter optimization. All these works focus on forecasting tourist flow demand, meaning they do not address the issue of whether individuals will repeat their visit. Additionally, they rely on extensive historical datasets, which means they do not encounter the challenges related to the limited data in our smart village use case. Although there are solutions for tabular data such as TabNet [26] and TabTransformer [27], they are prone to overfitting on small datasets due to their architecture [28].

In the field of tracing repeaters in tourism, we find few works of application to ML. [7] applies traditional classification algorithms to predict the visit intentionality of hotel guests, but it still does not explore new architectures such as transformers. [65] use transformer models tailored for text analysis. They analyze sentences, specifically tweets from X (former Twitter), to predict if an individual will visit or not a destination. However, our data is not textual, and in general, this methodology cannot be applied to rural villages, where textual data (tweets) are nearly nonexistent [66].

Our case study has a dataset limited in the number of instances. So our architecture must address a gap found in the literature: applying attention layers in NNs to improve the performance of the models when trained with small

tabular datasets. While these layers have been used in NN for image classification, their use for classifying tabular data remains unexplored.

3. SASD

The proposed multiclass classification model, depicted in Fig. 1, utilizes common layers in NNs, using a hybrid architecture that combines linear layers, batch normalization, ReLU (Rectified Linear Unit) activations, dropout and introduces self-attention layers. Attention mechanisms dynamically evaluate the relevance of different segments within the input data, enhancing the model's sensitivity and ability to learn about the relationships and dependencies in the sequence [67]. By evaluating the significance of each element in relation to its context and connections to others, the attention mechanism enables the model to focus on the most relevant parts of the input sequence, [68]. The decision to incorporate attention layers is motivated by their adaptability and versatility, which are important in handling noisy data, such as sensor data or questionnaire data, that are generally subjective.

The common layers use in the literature are linear, batch normalization, ReLU and dropout. A linear layer is a component in NNs that transforms input data into output features that can be used for further processing or making predictions. The core functionality of a linear layer lies in its ability to apply a linear transformation to the input data it receives. After this linear layer, batch normalization and ReLU activations take place. Firstly, batch normalization is a technique that aims to stabilize and accelerate the training process by normalizing the inputs of each layer within a network. It works by adjusting and scaling the activations of a previous layer, based on the mean and variance of the current mini-batch of data [69]. Secondly, the ReLU adds non-linearity to our model, enabling it to learn and model complex relationships in the data. In this way, it helps maintain the essential characteristics of the data while minimizing the impact of noise [70]. The use of batch normalization followed by ReLU activations demonstrates that it solves the "Vanishing Gradient Problem" since the input is normalized, and it maintains positive and constant gradients, accelerating learning in NN [69, 71], which in the case of having small datasets is essential. Moreover, the combined use of batch normalization with the dropout layer speeds the convergence. It reduces overfitting, as the mutual information and correlation coefficient between any pair of neurons are reduced [72]. Specifically, dropout layers are particularly beneficial in scenarios involving noisy data because there is a risk that the NN learns the noise as if it were a significant feature of the problem, leading to overfitting and, consequently, poor generalization to new data [73].

Our model addresses the problem of optimizing the NN architecture for multiclass classification tasks with scarce training data through the exploration of integrating and positioning attention layers within the network's architecture. The attention mechanism captures complex dependencies and performs contextual analysis of the input data. Our work has focused on investigating how the inclusion of attention layers and the position in the model affects the performance of the model. From a hybrid architecture that combines linear layers, batch normalization, ReLU activations and dropout, we have maintained the core of the model while systematically varying the position of attention layers throughout the network: at the beginning, in the middle, at the end, etc. Fig. 1 shows one of the configurations, the one that reaches the best results, with two self-attention layers, one at the beginning and one at the end of the network. To adapt the model to our small dataset, we apply batch normalization, ReLU activation, and sequential dropout. This combination accelerates convergence [69], adds nonlinearity for better learning of variable relationships [70], and avoids overfitting [73], directly solving the frequent problems of training limited data [71]. We have also experimented with models that, instead of self-attention layers, have multihead-attention layers.

Mathematical representation of self-attention and multihead-attention

Self-attention is an attention mechanism where the input sequence itself serves as the queries, keys, and values. This allows the model to weigh the importance of the words in a sentence, capturing long-term dependencies and enhancing the contextual representation of each word. For an input sequence $X = [x_1, x_2, \cdots, x_n]$, where x_i is a vector of features for the i-th word, the self-attention mechanism (see Fig. 2) is calculated as follows:

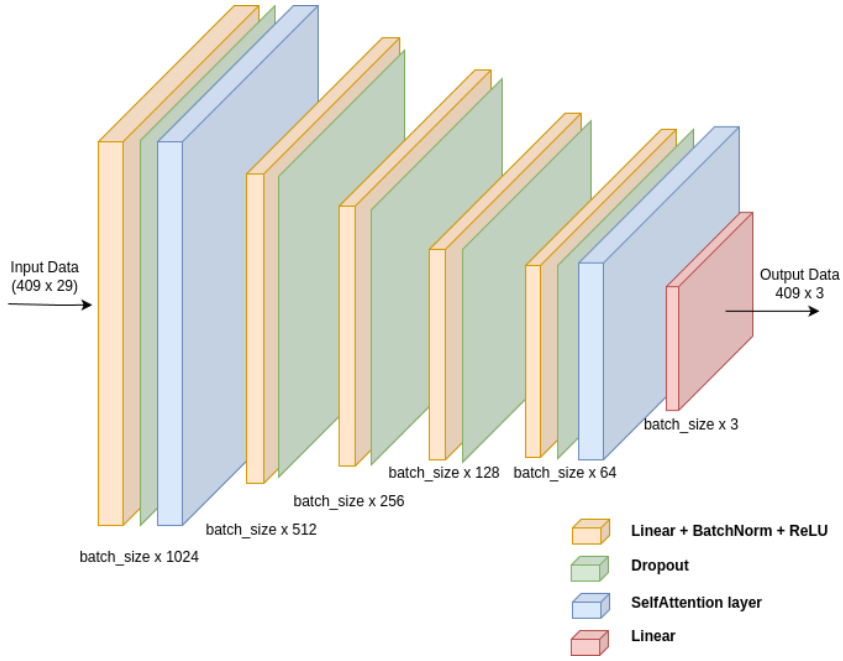


Fig. 1. SASD: proposed model with two self-attention layers, one at the beginning and one at the end of the network.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$

This Equation 1, is known as "Scaled Dot-Product Attention" and is characterized by:

- Linearly transform the input X to obtain the matrices of queries (Q), keys (K), and values (V), respectively:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V, \qquad (2)$$

  where W_Q, W_K, and W_V are weight matrices that facilitate the transformation of the input data X into different representations for the purpose of computing attention scores.

- Compute the dot product QK^T to obtain the attention scores, which are then scaled by the inverse square root of the dimension of the keys (d_k), i.e., $1/\sqrt{d_k}$, to prevent the scores from becoming too large.

- Finally, apply the softmax function to each row of the scaled QK^T matrix, normalizing the weights to sum 1, and use these to weight the values (V) through matrix multiplication.

The multihead-attention mechanism shortens the processing time of the self-attention mechanism by processing multiple attention tasks simultaneously. Initially, the Q, K, and V vectors are projected into h separate sets, named Q_i, K_i, and V_i for each $i = 1, \ldots, h$. Then, the scaled dot-product attention Equation 1 is applied to each projection set. Following this, the results are merged by first concatenating them and then applying a linear projection. See Fig. 3 and Equation 3.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \qquad (3)$$

where each head_i is calculated as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \qquad (4)$$

Here, W_i^Q, W_i^K, and W_i^V represent the projection matrices for transforming Q, K, and V in the i-th head, and W^O

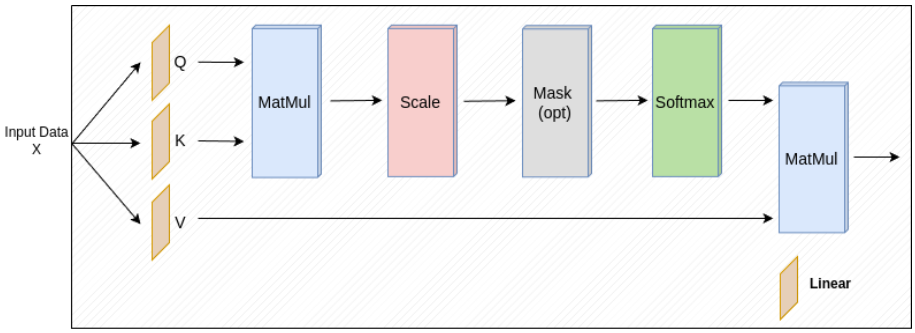


Fig. 2. Self-attention layer

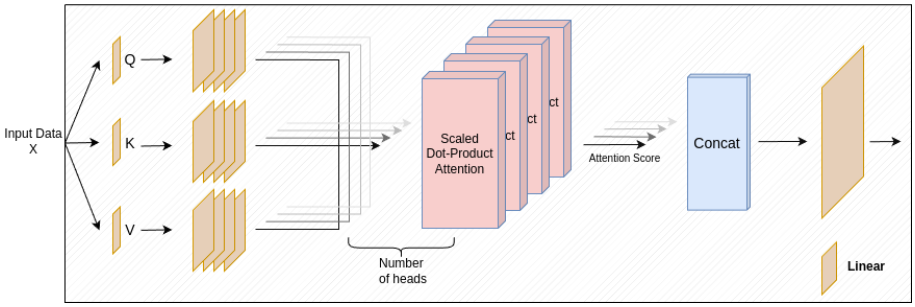


Fig. 3. Multihead-attention layer

is the matrix used for the final projection.

In multihead-attention, each head focuses on different parts of the input sequence. Hence, the model captures a wider range of dependencies than a single instance of self-attention.

4. Use case

The application of our model is focused on the visit intentionality of different visitors to a tourist area in the next 12 months. This intentionality is gathered through questionnaires, which, due to the inherent cost of the questionnaires, led to a scarce dataset. To define the profile of the respondents, we use visitor behavior data from previous visits, matching the license plate of the questionnaires with the previous visits. Data from previous visits is obtained from a sensor network installed in the area. The system comprises four Hikvision LPR IP cameras with Deep Learning-based Automatic Number Plate Recognition (ANPR). We focus on a rural tourist area, specifically in three villages near the Sierra Nevada National Park in Granada, Spain. These cameras record vehicle license plates at each village's entry and exit points, as depicted in Fig. 4. Due to the road layout, deploying four cameras strategically covers all access points to the villages, optimizing costs and system complexity. The LPRs record the license plate number of vehicles and the timestamp at the moment they pass by. To anonymize this information, we replace each license plate number with a unique integer value. The dataset covers all the traffic in the area from February 2022 to July 2023 (17 months).

Fig. 5 shows the data collection process, preprocessing performed on the datasets to obtain clean data, and data analysis. In the next subsections, we explain the four main phases of Fig. 5, highlighted in yellow. In the first phase, we collected and preprocessed data to obtain dataset variables. The overall sample size of the questionnaires limits the size of the merged data from sensors and external sources. Thus, data scarcity will significantly constrain our predictive model. The anomaly detection phase aims to identify and remove data points that could introduce noise in the distribution. During the data transformation phase, we apply various scaling changes to normalize the data distance. Finally, we select an ML model and evaluate it using different metrics.

4.1. Data collection and preprocessing

This process involves collecting data from sensors and heterogeneous data sources, such as web pages or datasets. In our case, we collected the data from the LPR cameras and combined their information with other contextual datasets, including vehicle information, demographic and economic data, national holidays calendar and geographic data. The information collected includes more than 85,000 vehicles. In addition, we collected 522 questionnaires from one of the villages. We surveyed different drivers in the parking area, collecting demographic information and visiting details through various questions.

With the license plate variable we cross-referenced LPRs data, contexts datasets and questionnaires, resulting in a dataset of 522 instances, limited by the number of questionnaires. Then, we select the most relevant variables for our problem using a correlation study. Additionally, we remove rows containing missing information, resulting in 514 vehicles with complete data from all sources. We only use one variable (pred_label) from the questionnaires, which contains the respondent's intention to visit the area in the next 12 months. The datasets are designed from the vehicle's perspective and contain the average or accumulated information (depending on the variable) of its behavior in all the visits to the area. We consider a visit the time that a vehicle spends between entering and leaving the area. We built five datasets with 30 variables, described in Table 1. These datasets are:

- **Vehicle behavior:** consists of variables that capture details for representing the spatio-temporal frequency of a vehicle in the given area. These variables are derived from raw LPR data. These variables involve additional computations based on fundamental camera variables (such as total_entries, which count the entries to the area in different visits. We separate each visit as the action of going out of the area and not coming back in at least 30 minutes. These variables offer insights into vehicle behavior within the area. These unique variables, which are not usually included in the LPR literature, provide information on the spatio-temporal patterns of vehicles and activities during the stay in the area.

- **Holiday context:** contains different variables related

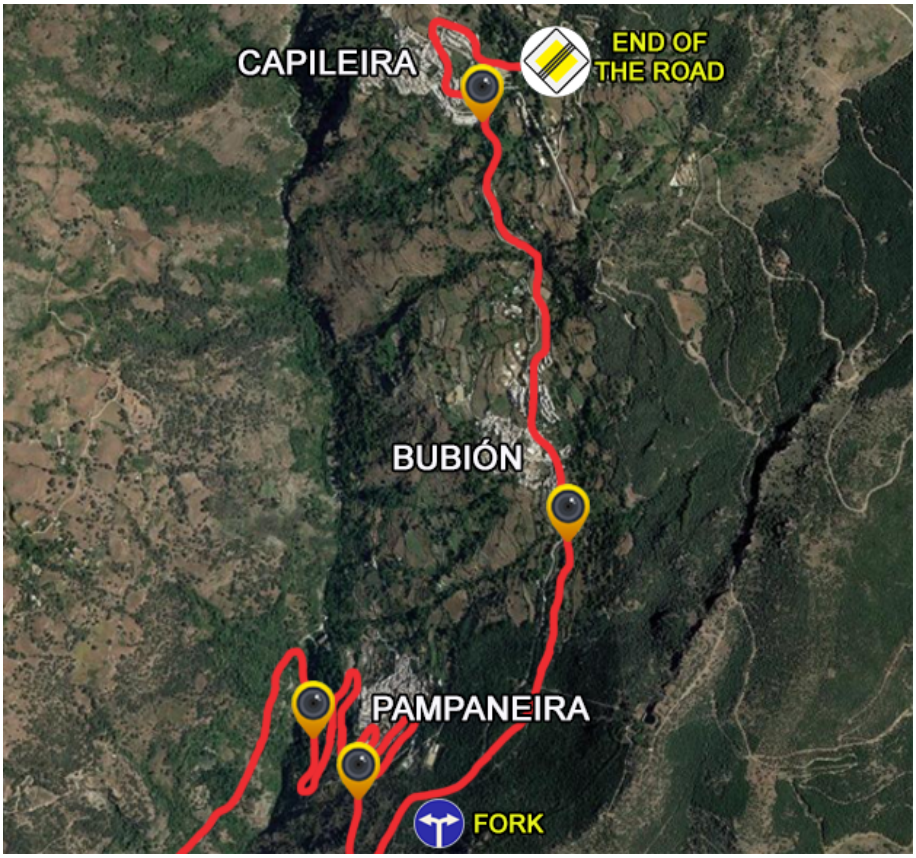


Fig. 4. Configuration of the four-LPR system that gathers vehicle data.

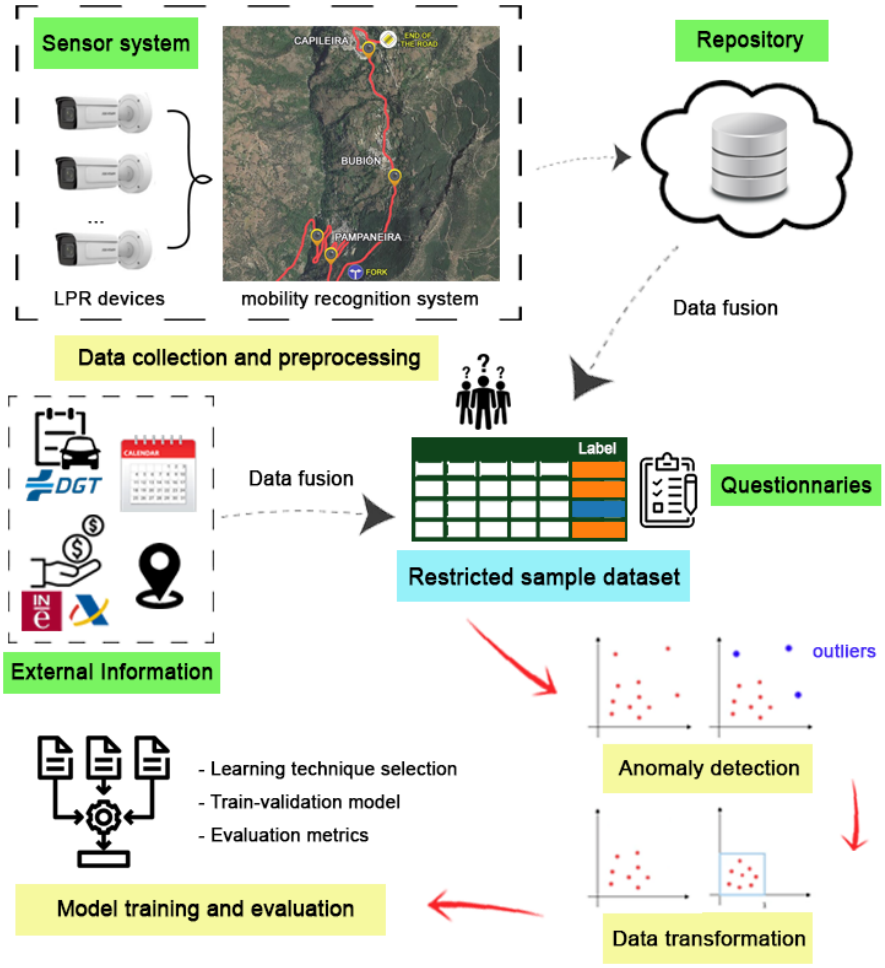


Fig. 5. Schematic of the data collection, preprocessing methodology used, and data analysis.

to vacations and working days based on external information from national calendars. High season days include the most important holiday periods: Summer Holiday, Christmas, and Holy Week[1]. The national calendar data are from the Python library "holidays"[2].

- **Geographic data** contains variables related to the vehicle's provenance based on external information from a private dataset provided by the Spanish Directorate-General for Traffic (DGT)[3]. We obtained the distance to the area from their provenance information and two libraries: pgeocode[4] and geopy[5]. The variable 'area' is a number from 1 to 5 that indicates the assigned zone for a vehicle based on its provenance region. This assignment considers both the geographical location and the Gross Domestic Product (GDP). The areas are defined as follows: Area 1 (Areas nearby with low GDP), Area 2 (Areas a bit further away with low to medium GDP), Area 3 (Intermediate areas with high GDP), Area 4 (Distant areas with high GDP), and Area 5 (Remote areas with low GDP).

- **Demographic and Economic data:** contains further variables related to the vehicle's provenance. These variables are the population and gross income. They are obtained from the website of the National Statistics Institute (Spanish: Instituto Nacional de Estadistica, INE)[6], the data are available for regions with more than 1000 inhabitants and are updated until 2020.

- **Questionnaires:** conducted in January, March, and July 2023 on a population excluding local residents and maintaining the same proportion as the LPR dataset among visitors from the five different areas defined above. The questionnaires were taken in a parking lot and contained 15 questions, collecting a variety of demographic and behavioral information related to vehicle usage and visitor patterns. Additionally, the surveyor visually confirmed the vehicle license plate number to ensure data accuracy. One of the questions included was the intention to visit the area (in number of visits) in the next 12 months, which is the variable to predict in our problem. Six of the remaining questions are related to LPR information: time of entry into the zone, approximate time of exit from the zone, number of visits to the zone in the last year, overnight stays during their visit, license plate number, and residential postcode. These were employed as control questions to verify and validate the questionnaire data with the LPR camera data. The other eight variables: age, gender, annual income, education level, number of passengers, employment status, and two additional variables related to tax payment requested by policymakers, were not used in this work, as they were intended for municipal reports and other applications.

4.2. *Anomaly detection*

Outliers are defined as data points that deviate significantly from the majority due to errors or variations, and can greatly influence the statistical results [74, 75]. In the context of LPR data (visiting behavior) and questionnaires (intentionality to repeat), we refer to outliers as people who give contradictory answers or show inconsistent behavior, which may occur despite the application of a screening phase. For example, a visitor who has indicated that they will return several times in the future, but their behavior in the area is similar to that of people who have never repeated visits. This approach avoids biasing the results and improves the robustness of the analysis by addressing and resolving inconsistencies or outliers in the data prior to the analysis [75]. For this task, we tested 3 popular outlier detection algorithms. The first one is the Isolation Forest, which creates multiple random decision trees to identify and eliminate potential anomalies in the data [76]. This method is effective as anomalies tend to be isolated in smaller regions of the feature space, making them easier to detect. This algorithm allows the specification of the contamination percentage of the dataset, so

[1] https://es.statista.com/temas/3585/vacaciones-en-espana/#topicOverview
[2] https://python-holidays.readthedocs.io/en/latest/
[3] https://sede.dgt.gob.es/es/vehiculos/informe-de-vehiculo/
[4] https://pgeocode.readthedocs.io/en/latest/
[5] https://geopy.readthedocs.io/en/latest/
[6] https://www.ine.es/index.htm

Data type	Variable	Type	Description
Vehicle behavior (LPR cameras)	visit_time	Time	Total time of stay.
	distance	Float	Total distance traveled in kilometers within the area.
	nights	Integer	Number of nights.
	visits_dif_weeks	Integer	Number of different weeks with at least one visit.
	visits_dif_months	Integer	Number of different months with at least one visit.
	fidelity	Float	Number of visits after maintaining fidelity of at least five days.
	total_entries	Integer	Total number of entries.
	avg_visit	Time	Average visit time.
	std_visit	Time	Standard deviation of the average visit time.
	avg_nights	Float	Average number of nights.
	std_nights	Float	Standard deviation of the average number of nights.
Holiday context (National calendar)	num_holiday	Integer	Number of holidays spent.
	avg_holiday	Float	Average number of holidays spent.
	std_holiday	Float	Standard deviation of the average number holidays spent.
	num_workday	Integer	Number of workdays spent.
	avg_workday	Float	Average number of workdays spent.
	std_workday	Float	Standard deviation of the average number workdays spent.
	num_high_season	Integer	Number of high season days spent.
	avg_high_season	Float	Average number of days of high season spent.
	std_high_season	Float	Standard deviation of the average number of days of high season spent.
	num_low_season	Integer	Number of low season days spent.
	avg_low_season	Float	Average number of days of low season spent.
	std_low_season	Float	Standard deviation of the average number of days of low season spent.
	entry_in_holiday	Integer	Number of entries on holiday.
	entry_in_high_season	Integer	Number of entries in high season.
Demographic and Economic data (INE)	population	Integer	Population size of the city/town of the provenance of the vehicle.
	avg_gross_income	Float	Average gross income of the area of origin of the vehicle.
Geographic data (DGT)	km_to_dest	Float	Distance in kilometers between the origin of the vehicle and the destination region.
	area	Integer	Label indicating the area of origin of the vehicle among the 5 defined for the problem.
Questionnaires	pred_label	Integer	Label of the predicted class that determines the intentionality of a repeat visit to the area.

Table 1
Definition of the variables from different datasets.

it will adapt its parameters to eliminate the points necessary to reach that sample percentage. We also consider the interquartile range method (IQR). It identifies outliers beyond the range defined by $[Q_1 - 1.5 \cdot \text{IQR}, Q_3 + 1.5 \cdot \text{IQR}]$, where Q_1 and Q_3 are the first and third quartiles of the data, respectively [77, 75]. Adjusting the value 1.5 allows us to control the sensitivity to outliers. The last method used is based on the z-scores of the observations given by $z_i = \frac{x_i - \bar{x}}{s}$, where s is the standard deviation. The classical rule marks a point as an outlier if its Z-score exceeds 2.5. More precisely, the rule marks x_i as peripheral if $|z_i|$ exceeds 2.5 [77], but the threshold can be adapted to increase or decrease the percentage of outliers.

4.3. Data transformation

Time variables have been converted into numerical values (expressed in hours) to facilitate the application of feature scaling techniques and subsequent utilization in ML algorithms. Various scaling methods have been employed in the literature, including the popular min-max normalization and standardization [78, 79], and alternative methods based on IQR or considering the maximum absolute value of each variable [80]. In our study, we evaluate these four methods to determine the most suitable one for our data and analysis.

4.4. Model training and evaluation

In this use case, our goal is to predict tourist visit intentionality over the next 12 months. We opted for a classifi-

cation approach in supervised learning due to its effectiveness in categorizing labeled data. Regression, which tends to struggle with large data variability like ours, would yield predictions with a significant margin of error [81]. For our problem, predicting exact details, such as the exact number of intentional visits, is unnecessary; hence, applying regression would not be appropriate. We discretized the variable pred_label to create a classification problem, dividing it into 3 separate classes: "Do not know or will not repeat visit" (214 vehicles), "Will return 1 or 2 times" (202 vehicles), and "Will return 3 or more times" (98 vehicles). We chose a three-class rather than a two-class approach (return or not return) to capture detailed visitor behaviors, which assists policymakers in making decisions in the area.

To evaluate the model, we divide the data into three sets: training, validation and test, with a ratio of 70-10-20, respectively, as in the literature. We train the NN using the training set. We fit it with cross-validation using the Cross-Entropy Loss function [82], which evaluates the discrepancy between model predictions and actual labels. We then evaluate the model with the test set, comparing predictions with the actual labels to compute evaluation metrics such as precision, recall, F1 score, and accuracy [83]. We use the F1-score metric as a reference metric since it is commonly used for evaluating unbalanced data models [84]. We also use weighted average metrics for model evaluation. First, we calculated the specific metric for each class separately. Then, we consider the weights to obtain the average, assigning more weight to classes with a higher number of true instances for each label. This approach provides a better interpretation for cases where classes are unbalanced [85].

5. Experimental procedure

For training our model, we utilized the Adam optimizer with a learning rate of 6e-5, we determined this value by experimentation within a scale interval of e-3 to e-10, knowing that smaller learning rates can improve convergence stability, particularly for models trained on limited or small datasets [86]. We employed a batch size of 1048, which is supported by our NVIDIA GeForce RTX 3090 GPU environment with a total memory of 25.45 gigabytes, compatible with CUDA compute architecture 8.6. For programming, we use an anaconda environment compatible with Python 3.9.18. Table 2 shows the complete layer structure of our model. Alongside each layer, we provide the value of internal parameters and hyperparameters. The -1 value in the output shape column indicates a dynamic dimension that adapts automatically to the batch size of the data. The parameter p in dropout layers represents the probability of an element being zeroed, indicating the likelihood of dropout occurring for each individual element within the layer. The specified hyperparameter value for self-attention layers represents that the number of output dimension for the query and key convolution layers is one-fourth of the number of input dimension. We use the sigmoid activation function [87] in the self-attention layers, since it is the option that obtained the best results in all the cases evaluated.

We also use different algorithms popular in various classification problems to create the model and compare their performance with our proposed NN model. These algorithms come from different families, each with unique characteristics and advantages [88]. Recent research highlights that, in certain scenarios, traditional algorithms can outperform neural networks [89], in some cases models such as MultiLayer Perceptron (MLP) outperform tabular data transformers [90]. It is therefore important to conduct a comprehensive review of a range of traditional and deep learning approaches, including:

- **Decision Trees:** Decision Tree and Random Forest [91].
- **Nearest Neighbors:** K-Nearest Neighbors [92].
- **Logistic Regression:** Logistic Regression [93].
- **Gradient Boosting:** Gradient Boosting Classifier [94].
- **Neural Networks:** MLP Classifier [95], RNN [96], and Long Short-Term Memory (LSTM) [97].
- **Tabular Data Transformers:** TabNet [26], and TabTransformer [27].
- **Bayesian Probabilistic Models:** Gaussian Naive Bayes [93].
- **Support Vector Machines (SVM):** Linear SVM [98].

Layer name	Output shape	Num. trainable params	Hyperparameters
Linear-1	[-1,1024]	30,720	-
BatchNorm1d-2	[-1,1024]	2,048	-
ReLU-3	[-1,1024]	0	-
Dropout-4	[-1,1024]	0	$p = 0.6$
Self-Attention-5	[-1,1024]	1,574,400	out_dim=in_dim/4 activation=sigmoid
Linear-6	[-1, 512]	524,800	-
BatchNorm1d-7	[-1, 512]	1,024	-
ReLU-8	[-1, 512]	0	-
Dropout-9	[-1, 512]	0	$p = 0.6$
Linear-10	[-1, 256]	131,328	-
BatchNorm1d-11	[-1, 256]	512	-
ReLU-12	[-1, 256]	0	-
Dropout-13	[-1, 256]	0	$p = 0.5$
Linear-14	[-1, 128]	32,896	-
BatchNorm1d-15	[-1, 128]	256	-
ReLU-16	[-1, 128]	0	-
Dropout-17	[-1, 128]	0	$p = 0.3$
Linear-18	[-1, 64]	8,256	-
BatchNorm1d-19	[-1, 64]	128	-
ReLU-20	[-1, 64]	0	-
Dropout-21	[-1, 64]	0	$p = 0.3$
Self-Attention-22	[-1, 64]	6,240	out_dim=in_dim/4 activation=sigmoid
Linear-23	[-1, 3]	195	-

Table 2
Full configuration of the layers of the proposed self-attention model.

6. Results and discussion

Among the 3 outlier techniques tested (see Subsection 4.2), the one that yielded the best results was the application of the Isolation Forest algorithm across all variables. For this algorithm, we specify an outlier detection of 20% for each class, which aligns with several studies [75, 99]. By removing the same percentage of outliers per class, we preserve the class proportions of the original dataset avoiding potential biases in the analysis. The final sample consists of 410 vehicles, categorized as follows: "Do not know or will not repeat visit" (171 vehicles), "Will return 1 or 2 times" (161 vehicles), and "Will return 3 or more times" (78 vehicles). The resulting dataset has a ratio of approximately 40%-40%-20% between the three classes, our test dataset follows the same proportions. Next, we perform a transformation of the data, where the best results for all metrics were obtained by min-max normalization, followed by standardization, and finally scaling using robust statistics.

To prove our proposal, we perform 3 experiments[7]. In the first experiment, we use a baseline multiclass NN. This architecture obtained the best F1 score, accuracy, precision and recall results in 270 epochs, with a weighted average F1 score of 0.74. In the second experiment, we added self-attention layers (SASD) to the baseline NN. In the last experiment, we added multi-headed attention layers (MASD) in place of self-attention layers in SASD to introduce a parallelization of the tasks. We used 4 heads of attention in the construction of MASD based on experimentation with values between 2 and 64. This architecture obtained the best F1 score, accuracy, precision and recall results in 210 epochs, with a weighted average F1 score of 0.71. Baseline NN and MASD models' results for the precision, recall and accuracy metrics as well as the F1 score values for the three classes can be seen in Table 5.

[7]Models were run for 0–500 epochs, with final values based on metrics results convergence.

Results for SASD

We analyzed SASD, the NN with self-attention, across different epochs, analyzing how the number and placement of self-attention layers affect the model's efficiency. The best results were achieved using two self-attention layers: one at the beginning and one at the end of the network. This configuration accelerates convergence, maximizes metrics, and captures global dependencies early, enabling better contextualization in subsequent layers.

Hence, this first attention layer acts as a data exploration task, identifying the key characteristics in the model early. Additionally, we have added a self-attention layer before the last linear layer that acts as a refinement mechanism, adjusting the model output to focus on the aspects of the data that are more important in the final classification problem. This can be seen as a final exploitation task, which focuses on the key characteristics to make precise predictions. The comparative table of the different ablation test results can be found in Appendix A (see Table A.1).

Fig. 6 illustrates a comparison of training-validation sets for accuracy (left) and loss (right) per epoch metrics for the network with two self-attention layers at the initial and final positions, which were the configuration with the best results in the previous experiment. We can observe that the highest accuracy on the validation set is achieved between epochs 90 and 150. Regarding the loss graph, we can see that the model starts to overfit from around epoch 160 (the difference between train and validation graphs is high). From epoch 400 onwards, the validation line remains constant, indicating that the model stops improving.

Table 3 presents the evaluation metrics and processing time for epochs between 90 and 150. The best value is achieved at 120 epochs, with a score of 0.75. The experiment with self-attention layers achieves its maximum accuracy value in a smaller number of epochs, almost half of the experiment without attention layers. The processing time is shorter, and the F1 score metric value improves by 0.01 points.

As the results show that SASD outperforms MASD, we will perform the ablation test and fine-tuning over SASD. We conducted two ablation studies to analyze the influence of different model layers on our case study with a small dataset. The first study explores the effects of dropout layers and their probability values [24], and the second study examines the impact of the ReLU layer between BatchNorm and Dropout [69]. Our experiments reveal that decreasing dropout enhance model performance and that the ReLU layer improves the model's ability to generalize by introducing necessary nonlinearity. Further details on the dropout layers results can be found in Appendix A (see Table A.2). The parameter settings and final layer order obtained can be seen in Table 2.

Baseline classification algorithms

Table 4 presents the weighted metrics of precision, recall, and F1 score, accuracy, and processing time for the different popular classification models that are still used in the tourism literature [6, 7], the RNN and LSTM algorithms, and TabNet and TabTransformer tabular data models. We build the RNN model using a sequential RNN layer, which processes the input step-by-step. Additionally, we add a Dense layer with ReLU activation, follow by another Dense layer with 3 units with softmax activation for classification. Similarly, we build the LSTM model with the same structure but replaced the RNN layer with an LSTM layer, which includes gates (input, forget, and output) that regulate the flow of information, allowing them to capture long-term dependencies more effectively [100]. For traditional algorithms, we perform an 80/20 train-test split. SVM and logistic regression achieve similar values for all the metrics and outperformed the rest of these algorithms. RNNs and LSTMs achieve lower performance metrics compared to SVM or logistic regression because deep learning algorithms require larger amounts of data to train effectively, which is not the case of this dataset. Additionally, they are generally better suited for time series data, given their ability to capture and model the temporal dependencies inherent in such data [101]. The two tabular data models obtain higher values than the average of the traditional algorithms, surpassing even the RNNs. However, they fail to outperform MLP, Logistic Regression and SVM, aligning our results with those obtained in the work [90].

Discussion of the results

Table 5 summarises the results, comparing the results of the three NNs, including our two proposed: SASD and MASD, and the two best baseline classification algorithms. The best model is SASD, achieving a value

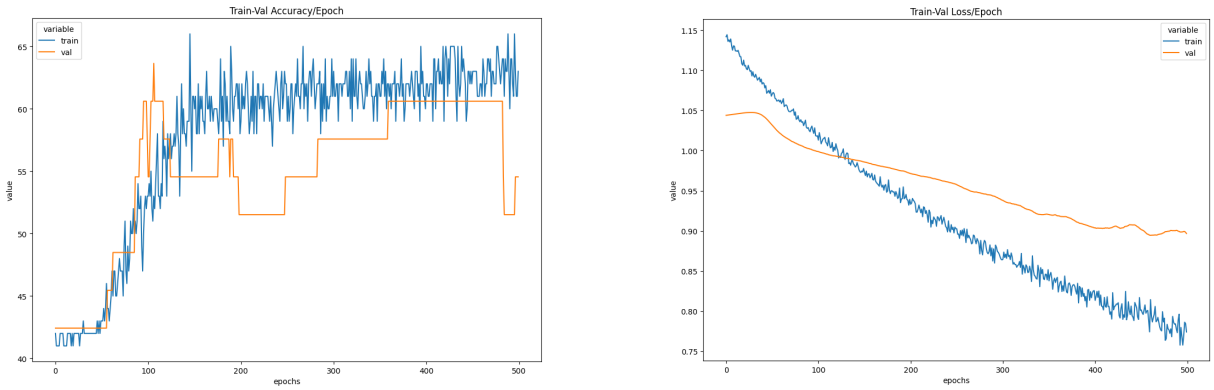


Fig. 6. Comparison of training-validation sets for accuracy (left) and loss (right) per epoch metrics for SASD.

	F1 score per class			weighted avg metrics				
Epochs	**Don't know or won't repeat visit (34)**	**Will return 1 or 2 times (32)**	**Will return 3 or more times (16)**	**precision**	**recall**	**F1 score**	**accuracy**	**time (sec.)**
90	0.66	0.36	0.77	0.67	0.60	0.57	0.60	**4.4260814**
100	0.66	0.59	0.77	0.68	0.65	0.65	0.65	5.0372171
110	0.69	0.68	**0.81**	0.75	0.71	0.71	0.71	5.8176632
120	**0.72**	0.73	**0.81**	**0.80**	**0.74**	**0.75**	**0.74**	6.2430282
130	0.70	0.73	0.79	0.78	0.73	0.73	0.73	6.8389513
140	0.70	**0.75**	0.80	0.78	**0.74**	0.74	**0.74**	7.4455154
150	0.70	0.74	0.77	0.76	0.73	0.73	0.73	8.0596223

Table 3
Results of processing time and metrics for SASD.

of 0.75 for the weighted F1 score metric. Additionally, this model has fewer epochs (converges more rapidly) and consequently a lower preprocessing time than the other NNs (up to 32% faster than the without attention version). Although multi-attention is, in general, a better option, in our case, when applied on a small dataset, multihead increases the complexity of the model, worsening the performance [24]. SASD is only surpassed by 0.01 points in the F1 score for the classification of the second class by the model without attention. The three neural network models outperform the baseline algorithms. However, as previously mentioned, these algorithms have significantly lower processing times, averaging around 450 times lower. Values of epochs higher than 120 might seem excessive for a dataset of 400 samples. However, in our case study, classification label is obtained from questionnaires that are considered a subjective variable, which might introduce noise that could affect the late convergence of the model [102, 103].

Our proposal pave the way for other researchers to explore the possibility of adding attention layers to other NNs architectures (such as RNNs), and to take all the advantages of the NN even when the amount of data is not enough to train a regular NN. Furthermore, this allows stakeholders involved in tourism to make predictions on small tabular datasets in order to take appropriate measures in the field of tourism development.

Practical implication for stakeholders

The findings of this paper hold significant policy implications. By predicting repeat visitors using deep learning techniques, the results inform sustainable tourism development strategies. Various stakeholders, including local governments, tourism businesses, communities, and environmental organizations, perceive these implications differently, shaping their strategies and practices accordingly.

First, local governments and tourism authorities play a

Model	weighted avg metrics				
	precision	**recall**	**F1 score**	**accuracy**	**time (sec.)**
Decision Tree	0.57	0.54	0.53	0.54	0.0030653
Random Forest	0.70	0.70	0.69	0.70	0.1659531
K-Nearest Neighbors	0.60	0.57	0.56	0.57	**0.0006871**
Logistic Regresion	0.78	**0.72**	**0.72**	**0.72**	0.0120337
Gradient Boosting	0.58	0.60	0.59	0.57	0.5109689
MLP Classifier	0.74	0.71	0.71	0.71	0.4370239
RNN	0.66	0.55	0.56	0.55	1.9156559
LSTM	0.67	0.60	0.57	0.60	2.1864752
TabNet	0.73	0.65	0.63	0.65	7.9505872
TabTransformer	0.71	0.70	0.70	0.70	12.2232128
Gaussian Naive Bayes	0.58	0.55	0.44	0.55	0.0010070
Linear SVM	**0.79**	**0.72**	**0.72**	**0.72**	0.0039882

Table 4
Results of processing time and metrics for the popular algorithms.

Model	F1 score per class			weighted avg metrics				
	Don't know or won't repeat visit (34)	**Will return 1 or 2 times (32)**	**Will return 3 or more times (16)**	**precision**	**recall**	**F1 score**	**accuracy**	**time (sec.)**
Baseline NN 270 epochs	**0.72**	**0.74**	0.79	0.79	**0.74**	0.74	**0.74**	9.1983194
SASD NN 120 epochs	**0.72**	0.73	**0.81**	**0.80**	**0.74**	**0.75**	**0.74**	6.2430282
MASD NN 210 epochs	0.71	0.67	**0.81**	0.74	0.71	0.71	0.71	11.7865441
TabTransformer	0.71	0.65	0.79	0.71	0.70	0.70	0.70	12.2232128
MLP Classifier	0.70	0.68	0.81	0.74	0.71	0.71	0.71	0.4370239
Logistic Regression	0.71	0.71	0.77	0.78	0.72	0.72	0.72	**0.0120337**
Linear SVM	0.70	0.72	0.77	0.79	0.72	0.72	0.72	0.0039882

Table 5
Results of processing time and metrics for 3 versions of our model and the best popular algorithms.

key role in guiding sustainable tourism development. By utilizing predictive models, they can enhance destination management strategies, optimize resource allocation, and improve visitor experience. Accurate predictions on visitor behavior also aid in effective urban planning and infrastructure development, ensuring sustainable growth in tourism [104]. Many of the local governments' decisions regarding tourism have been biased by the decision makers' expectations because data acquisition is expensive, and its scope is often limited. Our findings are useful for developing accurate data-driven decision-making and improving tourism governance and policy formulation [44].

Second, local tourism businesses and operators, such as hotels, often have difficulties getting enough data to accurately forecast visitor demand and tailor service offerings accordingly. Our findings are useful to optimize capacity utilization, improve operational efficiency, and enhance overall tourist satisfaction. Data-driven strategies in tourism operations management are highlighted for their importance, demonstrating how predictive models can lead to better service customization and resource management [45].

Third, local communities and residents play a vital role in the field of tourism by preserving cultural heritage and mitigating environmental impacts. The growing tensions between the tourist industry and local residents have shown the importance of community involvement in tourism planning processes to ensure the balance between economic and social sustainability of the tourism destinations [46]. Our findings can be useful to assist communities in understanding the tourist influx and providing opportunities to ask for effective sustainable tourism practices that benefit residents and visitors alike goals [105].

Fourth, environmental and conservation organizations

advocate for sustainable tourism practices to preserve natural ecosystems and cultural landscapes. The role of predictive analytics in guiding conservation efforts and minimizing ecological footprints associated with tourism activities is emphasized [106]. Our model can support environmental organizations in understanding the most effective visitor management strategies to promote responsible tourism practices and enhance the sustainability of tourism destinations [107].

7. Conclusion

This paper introduces a NN model (SASD) designed to address the challenge of tracing repeat visitors in the tourism sector. By leveraging LPR sensor data and related questionnaires, our proposal forecasts future visitation behaviors. We found that the best combination of attention layers in SASD model is when we place two layers of self-attention, the first one after the first normalization-regularization block and the second one at the end. The effectiveness of the proposed NN outperforms popular models in all classification metrics analyzed, achieving an increase of up to 0.03 in the weighted average F1 score. For our case study, our model outperforms other popular transformer models for tabular data found in the literature. Incorporating self-attention layers in the model improves performance on almost all metrics. It accelerates convergence, which is reflected in fewer epochs, and reduces processing time by 32% compared to the NN without attention layers. In addition, the integration of progressively decreasing dropout layers yields noticeable improvements, raising F1 scores by 0.01 points compared to the fixed probability dropout and by 0.08 points relative to models without dropout layers. The results highlight the potential of attention layers to optimize predictive models with small data, allowing data scientists and other researchers to address the limitations of data collection. For future research, we plan to expand our dataset. However, collecting additional data requires significant resources and careful logistical and budgetary planning. A larger dataset could allow the model to capture broader patterns and improve its accuracy. Although it might be necessary to modify the current architecture which is tuned to small datasets. In addition, we want to test the generalization of our model on other datasets to evaluate the suitability for other small datasets.

CRediT authorship contribution statement

Daniel Bolaños-Martinez: Methodology, Validation, Investigation, Resources, Software, Writing- Review & Editing. **Alberto Durán López:** Investigation, Writing-Review & Editing. **Jose Luis Garrido:** Investigation, Resources, Writing- Review & Editing, Supervision. **Blanca Delgado-Márquez:** Investigation, Writing-Review & Editing, Supervision, Project administration. **Maria Bermudez-Edo:** Conceptualization, Investigation, Methodology, Resources, Writing- Review & Editing, Supervision, Project administration.

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 Statement

The dataset for the questionnaires used to support these results can be found at [108]. In addition, a sample limited to 9 months of sensor data is available at [109], in the future we plan to update it with the 17-month dataset used.

Acknowledgments

This research work is funded by the Spanish Ministry of Economy and Competitiveness (Agencia Estatal de Investigación) through the project: Ref. PID2023-149185OB-I00/MCIN/AEI/10.13039/501100011033;
and also by R&D&i Project Ref. C-SEJ-128-UGR23, co-funded by the Regional Ministry of University, Research and Innovation and by the European Union under the Andalusia ERDF Program 2021-2027.

References

[1] C. Juaneda, Repeat tourism, Springer International Publishing, Cham, 2016, pp. 787–788. doi:10.1007/978-3-319-01384-8_300.
URL https://doi.org/10.1007/978-3-319-01384-8_300

[2] S. Acharya, M. Mekker, J. De Vos, Linking travel behavior and tourism literature: Investigating the impacts of travel satisfaction on destination satisfaction and revisit intention, Transportation research interdisciplinary perspectives 17 (2023) 100745.

[3] E. F. Amissah, E. Addison-Akotoye, S. Blankson-Stiles-Ocran, Service quality, tourist satisfaction, and destination loyalty in emerging economies, Marketing tourist destinations in emerging economies: Towards competitive and sustainable emerging tourist destinations (2022) 121–147.

[4] U. F. Alfarhan, K. Nusair, First-time, first-repeat and multiple-repeat visitors: a conditional counterfactual quantile expenditure decomposition analysis, Current Issues in Tourism 25 (15) (2022) 2377–2383.

[5] E. A. M. Sampetoding, E. Mahendrawathi, Digital transformation of smart village: A systematic literature review, Procedia Computer Science 239 (2024) 1336–1343.

[6] P. Sharma, U. Meena, G. K. Sharma, Application of data mining algorithms for tourism industry, in: Intelligent Computing and Applications: Proceedings of ICICA 2019, Springer, 2021, pp. 481–495.

[7] A. Dursun-Cengizci, M. Caber, Using machine learning methods to predict future churners: an analysis of repeat hotel customers, International Journal of Contemporary Hospitality Management (2024).

[8] M. Özbey, M. Kayri, Investigation of factors affecting transactional distance in e-learning environment with artificial neural networks, Education and Information Technologies 28 (4) (2023) 4399–4427.

[9] A. Tarhini, M. AlHinai, A. S. Al-Busaidi, S. M. Govindaluri, J. Al Shaqsi, What drives the adoption of mobile learning services among college students: An application of sem-neural network modeling, International Journal of Information Management Data Insights 4 (1) (2024) 100235.

[10] T. Das, S. Mobassirin, S. M. M. Hossain, A. Das, A. Sen, K. M. A. Kamal, K. Deb, Patient questionnaires based parkinson's disease classification using artificial neural network, Annals of Data Science (2023) 1–44doi:10.1007/s40745-023-00482-4.

[11] T.-C. T. Chen, H.-C. Wu, M.-C. Chiu, A deep neural network with modified random forest incremental interpretation approach for diagnosing diabetes in smart healthcare, Applied Soft Computing 152 (2024) 111183.

[12] S. G. Paul, A. Saha, M. Z. Hasan, S. R. H. Noori, A. Moustafa, A systematic review of graph neural network in healthcare-based applications: Recent advances, trends, and future directions, IEEE Access (2024).

[13] D. Bolaños-Martinez, J. L. Garrido, M. Bermudez-Edo, Predicting overnights in smart villages: the importance of context information, International Journal of Machine Learning and Cybernetics (2024) 1–20.

[14] M. Li, C. Zhang, S. Sun, S. Wang, A novel deep learning approach for tourism volume forecasting with tourist search data, International Journal of Tourism Research 25 (2) (2023) 183–197.

[15] J. Fan, W. Lu, S. B. Abduraimovna, J. Cheng, H. Fan, Graph-guided neural network for tourism demand forecasting, IEEE Access 11 (2023) 134259–134268.

[16] M. Nahiduzzaman, M. O. F. Goni, R. Hassan, M. R. Islam, M. K. Syfullah, S. M. Shahriar, M. S. Anower, M. Ahsan, J. Haider, M. Kowalski, Parallel cnn-elm: A multiclass classification of chest x-ray images to identify seventeen lung diseases including covid-19, Expert Systems with Applications 229 (2023) 120528. doi:10.1016/j.eswa.2023.120528.

[17] P. Shivakumara, C. P. Kumar, J. J. Nemade, K. Michael, A. Kumar, B. S. Anami, U. Pal, A new u-net based system for multi-cultural wedding

image classification, Expert Systems with Applications 237 (2024) 121562. doi:10.1016/j.eswa.2023.121562.

[18] N. Hussain, M. A. Khan, M. Sharif, S. A. Khan, A. A. Albesher, T. Saba, A. Armaghan, A deep neural network and classical features based scheme for objects recognition: an application for machine inspection, Multimedia Tools and Applications (2024) 1–23.

[19] A. Onan, Gtr-ga: Harnessing the power of graph-based neural networks and genetic algorithms for text augmentation, Expert systems with applications 232 (2023) 120908.

[20] A. Kanev, M. Nazarov, D. Uskov, V. Terentyev, Research of different neural network architectures for audio and video denoising, in: 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Vol. 5, IEEE, 2023, pp. 1–5.

[21] A. Margeloiu, N. Simidjievski, P. Lio, M. Jamnik, Weight predictor network with feature selection for small sample tabular biomedical data, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 9081–9089. doi:10.1609/aaai.v37i8.26090.

[22] D. McElfresh, S. Khandagale, J. Valverde, V. Prasad C, G. Ramakrishnan, M. Goldblum, C. White, When do neural nets outperform boosted trees on tabular data?, Advances in Neural Information Processing Systems 36 (2024). doi:10.48550/arXiv.2305.02997.

[23] R. N. D'souza, P.-Y. Huang, F.-C. Yeh, Structural analysis and optimization of convolutional neural networks with a small sample size, Scientific reports 10 (1) (2020) 834. doi:10.1038/s41598-020-57866-2.

[24] L. Brigato, L. Iocchi, A close look at deep learning with small data, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 2490–2497. doi:10.48550/arXiv.2003.12843.

[25] B. Barz, J. Denzler, Deep learning on small datasets without pre-training using cosine loss, in: Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2020, pp. 1371–1380. doi:10.1109/WACV45572.2020.9093286.

[26] S. Ö. Arik, T. Pfister, Tabnet: Attentive interpretable tabular learning, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 35, 2021, pp. 6679–6687. doi:10.1609/aaai.v35i8.16826.

[27] X. Huang, A. Khetan, M. Cvitkovic, Z. Karnin, Tabtransformer: Tabular data modeling using contextual embeddings, arXiv preprint arXiv:2012.06678 (2020). doi:10.48550/arXiv.2012.06678.

[28] S. Somvanshi, S. Das, S. A. Javed, G. Antariksa, A. Hossain, A survey on deep tabular learning, arXiv preprint arXiv:2410.12034 (2024).

[29] M. Lin, X. Zhao, Application research of neural network in vehicle target recognition and classification, in: 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), IEEE, 2019, pp. 5–8. doi:10.1109/ICITBS.2019.00010.

[30] Z. Ning, J. Huang, X. Wang, Vehicular fog computing: Enabling real-time traffic management for smart cities, IEEE Wireless Communications 26 (1) (2019) 87–93. doi:10.1109/MWC.2019.1700441.

[31] W. Yao, C. Chen, H. Su, N. Chen, S. Jin, C. Bai, Analysis of key commuting routes based on spatiotemporal trip chain, Journal of Advanced Transportation 2022 (2022). doi:10.1155/2022/6044540.

[32] O. Cats, F. Ferranti, Unravelling individual mobility temporal patterns using longitudinal smart card data, Research in Transportation Business & Management 43 (2022) 100816. doi:10.1016/j.rtbm.2022.100816.

[33] J. Tang, J. Zeng, Spatiotemporal gated graph attention network for urban traffic flow prediction based on license plate recognition data, Computer-Aided Civil and Infrastructure Engineering 37 (1) (2022) 3–23. doi:10.1111/mice.12688.

[34] G. Yang, D. Coble, C. Vaughan, C. Peele, A. Morsali, G. F. List, D. J. Findley, Waiting time estimation at ferry terminals based on license plate recognition, Journal of Transportation Engineering, Part A: Systems 148 (9) (2022) 04022064. doi:10.1061/JTEPBS.0000722.

[35] F. Kitsios, E. Mitsopoulou, E. Moustaka, M. Kamariotou, User-generated content behavior and digital tourism services: A sem-neural network model for information trust in social networking sites, International Journal of Information Management Data Insights 2 (1) (2022) 100056. doi:10.1016/j.jjimei.2021.100056.

[36] Y. Gao, Forecast model of perceived demand of museum tourists based on neural network integration, Neural Computing and Applications 33 (2021) 625–635. doi:10.1007/s00521-020-05012-4.

[37] S. Gössling, M. Balas, M. Mayer, Y.-Y. Sun, A review of tourism and climate change mitigation: The scales, scopes, stakeholders and strategies of carbon management, Tourism Management 95 (2023) 104681.

[38] D. Streimikiene, B. Svagzdiene, E. Jasinskas, A. Simanavicius, Sustainable tourism development and competitiveness: The systematic literature review, Sustainable development 29 (1) (2021) 259–271.

[39] S. Zeinab Aliahmadi, A. Jabbarzadeh, L. A. Hof, A multi-objective optimization approach for sustainable and personalized trip planning: A self-adaptive evolutionary algorithm with case study, Expert Systems with Applications 261 (2025) 125412. doi:https://doi.org/10.1016/j.eswa.2024.125412.
URL https://www.sciencedirect.com/science/article/pii/S0957417424022796

[40] M. H. Kolaee, A. Jabbarzadeh, S. M. J. M. A. e hashem, Sustainable group tourist trip planning: An adaptive large neighborhood search algorithm, Expert Systems with Applications 237 (2024) 121375. doi:https://doi.org/10.1016/j.eswa.2023.121375.
URL https://www.sciencedirect.com/science/article/pii/S0957417423018778

[41] F. Xu, N. Nash, L. Whitmarsh, Big data or small data? a methodological review of sustainable tourism, Journal of Sustainable Tourism 28 (2) (2020) 144–163.

[42] Z. Doborjeh, N. Hemmington, M. Doborjeh, N. Kasabov, Artificial intelligence: a systematic review of methods and applications in hospitality and tourism, International Journal of Contemporary Hospitality Management 34 (3) (2022) 1154–1176.

[43] P. Madzík, L. Falát, L. Copuš, M. Valeri, Digital transformation in tourism: bibliometric literature review based on machine learning approach, European Journal of Innovation Management 26 (7) (2023) 177–205. doi:10.1108/EJIM-09-2022-0531.

[44] M. Novotny, R. Dodds, P. R. Walsh, Understanding the adoption of data-driven decision-making practices among canadian dmos, Information Technology & Tourism (2023) 1–15.

[45] O. Troisi, A. Visvizi, M. Grimaldi, Digitalizing business models in hospitality ecosystems: toward data-driven innovation, European Journal of Innovation Management 26 (7) (2023) 242–277.

[46] B. F. Bichler, Designing tourism governance: The role of local residents, Journal of Destination Marketing & Management 19 (2021) 100389.

[47] D. Jaisinghani, M. Kaur, M. Joshi, S. Sharma, A. D. Ahmed, Impact of political risk and economic growth on the tourism industry: Evidence from a dynamic threshold panel model, Tourism Economics 30 (6) (2024) 1448–1464.

[48] S. A. Athari, U. V. Alola, A. A. Alola, A global perspective of the role of domestic economic, financial and political risks in inbound tourism, International Journal of Emerging Markets 18 (10) (2023) 4191–4213.

[49] G. G. Reivan-Ortiz, P. T. Cong, W.-K. Wong, A. Ali, H. T. T. Thu, S. Akhter, Role of geopolitical risk, currency fluctuation, and economic policy on tourist arrivals: temporal analysis of brics economies, Environmental Science and Pollution Research 30 (32) (2023) 78339–78352.

[50] Z. Dai, H. Liu, Q. V. Le, M. Tan, Coatnet: Marrying convolution and attention for all data sizes, Advances in neural information processing systems 34 (2021) 3965–3977. doi:10.48550/arXiv.2106.04803.

[51] Z. Lu, H. Xie, C. Liu, Y. Zhang, Bridging the gap between vision transformers and convolutional neural networks on small datasets, Advances in Neural Information Processing Systems 35 (2022) 14663–14677. doi:10.48550/arXiv.2210.05958.

[52] X. Pan, C. Ge, R. Lu, S. Song, G. Chen, Z. Huang, G. Huang, On the integration of self-attention and convolution, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 815–825. doi:10.48550/arXiv.2111.14556.

[53] Y. Li, H. Cao, Prediction for tourism flow based on lstm neural network, Procedia Computer Science 129 (2018) 277–283. doi:10.1016/j.procs.2018.03.076.

[54] H. Laaroussi, F. Guerouate, et al., Deep learning framework for forecasting tourism demand, in: 2020 IEEE international conference on technology management, operations and decisions (ICTMOD), IEEE, 2020, pp. 1–4. doi:10.1109/ICTMOD49425.2020.9380612.

[55] E. S. Silva, H. Hassani, S. Heravi, X. Huang, Forecasting tourism demand with denoised neural networks, Annals of Tourism Research 74 (2019) 134–154. doi:10.1016/j.annals.2018.11.006.

[56] Y. Dong, L. Xiao, J. Wang, J. Wang, A time series attention mechanism based model for tourism demand forecasting, Information Sciences 628 (2023) 269–290. doi:10.1016/j.ins.2023.01.095.

[57] C. Van Hinsbergen, A. Hegyi, J. Van Lint, H. Van Zuylen, Bayesian neural networks for the prediction of stochastic travel times in urban networks, IET intelligent transport systems 5 (4) (2011) 259–265. doi:10.1049/iet-its.2009.0114.

[58] Y. Yao, Y. Cao, A neural network enhanced hidden markov model for tourism demand forecasting, Applied Soft Computing 94 (2020) 106465. doi:10.1016/j.asoc.2020.106465.

[59] F. T. Sáenz, F. Arcas-Tunez, A. Muñoz, Nationwide touristic flow prediction with graph neural networks and heterogeneous open data, Information Fusion 91 (2023) 582–597. doi:10.1016/j.inffus.2022.11.005.

[60] D. Bolaños-Martinez, M. Bermudez-Edo, J. L. Garrido, Clustering pipeline for vehicle behavior in smart villages, Information Fusion 104 (2024) 102164. doi:10.1016/j.inffus.2023.102164.

[61] B. Liu, J. Pei, Z. Yu, Stock price prediction through gra-wd-bilstm model with air quality and weather factors, International Journal of Machine Learning and Cybernetics (2023) 1–18doi:10.1007/s13042-023-02008-z.

[62] D.-K. Kim, S. K. Shyn, D. Kim, S. Jang, K. Kim, A daily tourism demand prediction framework based on multi-head attention cnn: The case of the foreign entrant in south korea, in: 2021 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2021, pp. 01–10. doi:10.48550/arXiv.2112.00328.

[63] B. Wu, L. Wang, Y.-R. Zeng, Interpretable tourism demand forecasting with temporal fusion

transformers amid covid-19, Applied Intelligence 53 (11) (2023) 14493–14514.

[64] X. Li, Y. Xu, R. Law, S. Wang, Enhancing tourism demand forecasting with a transformer-based framework, Annals of Tourism Research 107 (2024) 103791.

[65] P. Fantozzi, G. Maccario, M. Naldi, Uncovering tourist visit intentions on social media through sentence transformers., Information (2078-2489) 15 (10) (2024).

[66] R. Arthur, H. T. Williams, Scaling laws in geo-located twitter data, PloS one 14 (7) (2019) e0218454.

[67] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017). doi:10.48550/arXiv.1706.03762.

[68] E. Y. Zhang, A. D. Cheok, Z. Pan, J. Cai, Y. Yan, From turing to transformers: A comprehensive review and tutorial on the evolution and applications of generative transformer models, Sci 5 (4) (2023) 46.

[69] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International conference on machine learning, pmlr, 2015, pp. 448–456. doi:10.48550/arXiv.1502.03167.

[70] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 807–814.

[71] T. Ergen, A. Sahiner, B. Ozturkler, J. Pauly, M. Mardani, M. Pilanci, Demystifying batch normalization in relu networks: Equivalent convex optimization models and implicit regularization, arXiv preprint arXiv:2103.01499 (2021). doi:10.48550/arXiv.2103.01499.

[72] G. Chen, P. Chen, Y. Shi, C.-Y. Hsieh, B. Liao, S. Zhang, Rethinking the usage of batch normalization and dropout in the training of deep neural networks, arXiv preprint arXiv:1905.05928 (2019). doi:10.48550/arXiv.1905.05928.

[73] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (1) (2014) 1929–1958.

[74] X. Chen, B. Zhang, T. Wang, A. Bonni, G. Zhao, Robust principal component analysis for accurate outlier sample detection in rna-seq data, BMC bioinformatics 21 (2020) 1–20.

[75] C. S. K. Dash, A. K. Behera, S. Dehuri, A. Ghosh, An outliers detection and elimination framework in classification task of data mining, Decision Analytics Journal 6 (2023) 100164.

[76] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 eighth ieee international conference on data mining, IEEE, 2008, pp. 413–422. doi:10.1109/ICDM.2008.17.

[77] P. J. Rousseeuw, M. Hubert, Robust statistics for outlier detection, Wiley interdisciplinary reviews: Data mining and knowledge discovery 1 (1) (2011) 73–79. doi:10.1002/widm.2.

[78] H. Henderi, T. Wahyuningsih, E. Rahwanto, Comparison of min-max normalization and z-score normalization in the k-nearest neighbor (knn) algorithm to test the accuracy of types of breast cancer, International Journal of Informatics and Information Systems 4 (1) (2021) 13–20. doi:10.47738/ijiis.v4i1.73.

[79] S. Patro, K. K. Sahu, Normalization: A preprocessing stage, arXiv preprint arXiv:1503.06462 (2015). doi:10.48550/arXiv.1503.06462.

[80] S. Eddamiri, F. Z. Bassine, V. Ongoma, T. Epule Epule, A. Chehbouni, An automatic ensemble machine learning for

wheat yield prediction in africa, Multimedia Tools and Applications (2024) 1–27doi:10.1007/s11042-024-18142-x.

[81] Y. S. Abu-Mostafa, M. Magdon-Ismail, H.-T. Lin, Learning from data (2012).

[82] Y. Ho, S. Wookey, The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling, IEEE access 8 (2019) 4806–4813. doi:10.1109/ACCESS.2019.2962617.

[83] J. Lever, M. Krzywinski, N. Altman, Points of significance: Classification evaluation, Nature Methods 13 (2016) 603–604. doi:10.1038/nmeth.3945.

[84] N. W. S. Wardhani, M. Y. Rochayani, A. Iriany, A. D. Sulistyono, P. Lestantyo, Cross-validation metrics for evaluating classification performance on imbalanced data, in: 2019 international conference on computer, control, informatics and its applications (IC3INA), IEEE, 2019, pp. 14–18. doi:10.1109/IC3INA48034.2019.8949568.

[85] A. Gupta, N. Tatbul, R. Marcus, S. Zhou, I. Lee, J. Gottschlich, Class-weighted evaluation metrics for imbalanced data classification, arXiv preprint arXiv:2010.05995 (2020).

[86] D. Masters, C. Luschi, Revisiting small batch training for deep neural networks, arXiv preprint arXiv:1804.07612 (2018). doi:10.48550/arXiv.1804.07612.

[87] C. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, Activation functions: Comparison of trends in practice and research for deep learning, arXiv preprint arXiv:1811.03378 (2018). doi:10.48550/arXiv.1811.03378.

[88] G. Bonaccorso, Machine learning algorithms: Popular algorithms for data science and machine learning (2018).

[89] S. Yingze, S. Yingxu, Z. Xin, Z. Jie, Y. Degang, Comparative analysis of the tabnet algorithm and traditional machine learning algorithms for landslide susceptibility assessment in the wanzhou region of china, Natural Hazards 120 (8) (2024) 7627–7652.

[90] G. Zabërgja, A. Kadra, J. Grabocka, Tabular data: Is attention all you need?, arXiv preprint arXiv:2402.03970 (2024). doi:10.48550/arXiv.2402.03970.

[91] J. Ali, R. Khan, N. Ahmad, I. Maqsood, Random forests and decision trees, International Journal of Computer Science Issues (IJCSI) 9 (5) (2012) 272.

[92] L. E. Peterson, K-nearest neighbor, Scholarpedia 4 (2) (2009) 1883. doi:10.4249/scholarpedia.1883.

[93] P. Tsangaratos, I. Ilia, Comparison of a logistic regression and naïve bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size, Catena 145 (2016) 164–179. doi:10.1016/j.catena.2016.06.004.

[94] J. H. Friedman, Greedy function approximation: a gradient boosting machine, Annals of statistics (2001) 1189–1232doi:10.1214/aos/1013203451.

[95] L. K. Hansen, P. Salamon, Neural network ensembles, IEEE transactions on pattern analysis and machine intelligence 12 (10) (1990) 993–1001. doi:10.1109/34.58871.

[96] L. R. Medsker, L. Jain, et al., Recurrent neural networks, Design and Applications 5 (64-67) (2001) 2.

[97] A. Graves, A. Graves, Long short-term memory, Supervised sequence labelling with recurrent neural networks (2012) 37–45.

[98] T. Joachims, Training linear svms in linear time, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, pp. 217–226. doi:10.1145/1150402.1150429.

[99] D. Kim, J. Hwang, J. Lee, K. Kim, Y. Kim, Odim: Outlier detection via likelihood of under-fitted generative models (2024). arXiv:2301.04257.

[100] A. Sherstinsky, Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network, Physica D: Nonlinear Phenomena 404 (2020) 132306.

[101] Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: Lstm cells and network architectures, Neural computation 31 (7) (2019) 1235–1270.

[102] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning (still) requires rethinking generalization, Communications of the ACM 64 (3) (2021) 107–115.

[103] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, S. Lacoste-Julien, A closer look at memorization in deep networks (2017). arXiv:1706.05394.
URL https://arxiv.org/abs/1706.05394

[104] S. Novianti, E. Susanto, W. Rafdinal, Predicting tourists' behaviour towards smart tourism: the case in emerging smart destinations, Journal of Tourism Sustainability 2 (1) (2022) 19–30.

[105] T. H. Lee, F.-H. Jan, Can community-based tourism contribute to sustainable development? evidence from residents' perceptions of the sustainability, Tourism Management 70 (2019) 368–380.

[106] N. Kongbuamai, Q. Bui, H. M. A. U. Yousaf, Y. Liu, The impact of tourism and natural resources on the ecological footprint: a case study of asean countries, Environmental Science and Pollution Research 27 (2020) 19251–19264.

[107] S. E. Bibri, J. Krogstie, Environmentally data-driven smart sustainable cities: Applied innovative solutions for energy efficiency, pollution reduction, and urban metabolism, Energy Informatics 3 (1) (2020) 29.

[108] J. I. Urriza, D. Bolaños-Martinez, B. L. Delgado Márquez, J. L. Garrido, M. Bermudez-Edo, J. A. Aragon-Correa, PoqueiraSurveys: A Dataset on Economic Impact Surveys in the region of Barranco del Poqueira in the Alpujarra Granadina. (Sep. 2023). doi:10.5281/zenodo.8328348.

[109] D. Bolaños-Martinez, M. Bermudez-Edo, J. L. Garrido, B. L. Delgado Márquez, U. Julián Ignacio, A.-C. Juan Alberto, Federation of Vehicular Data in Smart Villages with Socioeconomic Information (Dec. 2023). doi:10.5281/zenodo.10245475.

Appendix A. Supplementary experiments

Model		weighted avg metrics				
Num. self-attention layers	**Epochs**	**precision**	**recall**	**F1 score**	**accuracy**	**time (sec.)**
0 (baseline NN)	270	0.79	0.74	0.74	0.74	9.1983194
1 (after the first linear layer)	40	0.78	0.71	0.71	0.71	**1.9677348**
1 (after the last linear layer)	115	0.79	0.72	0.72	0.72	5.1671507
2 (after first-last linear layers)	120	**0.80**	**0.74**	**0.75**	**0.74**	6.2430282
2 (after middle linear layers)	230	0.74	0.71	0.71	0.71	17.1272461
3 (after first-middle-last linear layers)	70	0.77	0.73	0.73	0.73	5.4467063
3 (after middle linear layers)	250	0.74	0.71	0.71	0.71	21.9595783
4 (after first-middle-last linear layers)	75	0.73	0.71	0.71	0.71	6.9297996
5 (after every linear layer)	55	0.72	0.68	0.69	0.68	6.2177124

Table A.1
Comparison of SASD metrics varying the number and position of attention layers.

Model	weighted avg metrics				
Dropout configuration	**precision**	**recall**	**F1 score**	**accuracy**	**time (sec.)**
Decreasing dropout (0.6-0.3)	**0.80**	**0.74**	**0.75**	**0.74**	6.2430282
Equal dropout (0.6)	0.59	0.54	0.44	0.54	6.2656787
Equal dropout (0.5)	0.73	0.70	0.70	0.70	6.2142571
Equal dropout (0.4)	0.72	0.71	0.71	0.71	6.2526220
Equal dropout (0.3)	0.77	**0.74**	0.74	**0.74**	6.2565201
Equal dropout (0.2)	0.77	**0.74**	0.74	**0.74**	6.4881895
No dropout layers	0.67	0.67	0.67	0.67	**5.9259279**

Table A.2
Comparison of SASD metrics with different dropout configurations.

UNIVERSIDAD
DE GRANADA