

UNIVERSIDAD ANDRÉS BELLO
FACULTAD DE INGENIERÍA
INGENIARÍA EN COMPUTACIÓN E INFORMÁTICA



**Universidad
Andrés Bello®**

**Detección de trayectos anómalos en vehículos usando
deep learning**

Richard Soria Mora

Proyecto de título para optar al título de Ingeniería en Computación e Informática.

Santiago - Chile

2019

Índice general

I Capítulo 1	13
1. Introducción	14
1.1. Introducción	14
1.2. Descripción del problema	16
1.3. Alcance	18
1.3.1. Objetivo general	18
1.3.2. Objetivos específicos	18
1.3.3. Límites	19
1.3.4. Supuestos	19
II Capítulo 2	20
2. Fundamentación del tema	21
2.1. Estado del arte	21
2.2. Generalidades de aprendizaje automático	23
2.2.1. Aprendizaje supervisado	23
2.2.2. Aprendizaje no supervisado	24
2.2.3. ¿Cuándo utilizar aprendizaje automático?	25
2.3. Redes neuronales artificiales	25
2.3.1. Ventajas de redes neuronales artificiales	26
2.3.2. Aprendizaje en las redes neuronales artificiales	27
2.4. Perceptrón multicapa	27
2.4.1. Algoritmo de aprendizaje	31

2.4.2. Descenso de gradiente	32
2.5. Redes Neuronales Autoencoder	33
2.5.1. Autoencoder	33
2.5.2. Autoencoder apilado	35
2.6. Redes neuronales convolucionales	36
2.6.1. Operación convolucional	37
2.6.2. Convolucional transpuesta	38
2.7. Redes Neuronales Autoencoder Convolucionales apilados	39
2.8. Series de tiempo	40
2.9. Datos outlier o atípicos	40
2.9.1. Método Convex Hull	40
2.9.2. Basado en Clustering(K-means)	42
2.9.3. Basado en Clustering(PAM)	43
2.9.4. Basado en densidad (LOF)	43
2.10. Comparación de métodos clásicos vs redes neuronales	45
2.11. Inteligencia artificial explicable (XAI)	46
2.12. Árbol de decisión	47
III Capítulo 3	49
3. Materiales y métodos	50
3.1. Modelo de desarrollo de software y machine learning	50
3.1.1. Modelo cascada	51
3.1.2. Modelo CRISP-DM(Cross Industry Standard Process for Data Mining)	52
3.2. Acta de constitución del proyecto	53
3.3. Planificación de proyecto	55
3.3.1. Gestión de alcance	55
3.3.2. Plan de tiempo	57
3.3.3. Plan de costos	60
3.3.4. Plan de calidad	60
Detección de series de tiempo anómalos	2

3.3.5. Plan de riesgos	61
3.4. Materiales	62
3.4.1. Hardware	62
3.4.2. Software	62
3.5. Comprensión de los datos	63
3.5.1. Recolección de datos	63
3.5.2. Descripción de datos	63
3.6. Preparación de datos	65
3.6.1. Selección de variables	65
3.6.2. Limpieza de datos	66
3.6.3. Exploración de los datos	66
3.7. Transformación de datos a series de tiempo	67
3.8. Criterio para la detección de series de tiempo outlier	69
3.9. Evaluación del modelo	70
3.9.1. Evaluación cuantitativa	70
3.10. Evaluación cualitativa	72

IV Capítulo 4 **73**

4. Resultado y discusión	74
4.1. Solución de alto nivel	74
4.2. Análisis de software	75
4.2.1. Requerimientos funcionales	75
4.2.2. Requerimientos no funcionales	76
4.3. Diseño de software	76
4.3.1. Descripción de la arquitectura	76
4.3.2. Vista de escenario (Diagrama de casos de uso)	77
4.3.3. Vista lógica (Diagrama de clases)	78
4.3.4. Vista de desarrollo (Diagrama de componentes)	78
4.3.5. Vista física (Diagrama de despliegue)	79
4.3.6. Vista de procesos (Diagrama de secuencia)	79

4.4.	Interfaz gráfica de módulo historial de rutas	80
4.5.	Pruebas unitarias	81
4.6.	Conjunto de datos	81
4.7.	Método base (LOF)	82
4.7.1.	Configuración Local Outlier Factor(LOF)	82
4.7.2.	Ejemplos de serie de tiempo con LOF	84
4.8.	Modelos propuestos	86
4.8.1.	Red autoencoder apilada	86
4.8.2.	Red autoencoder convolucional apilada	86
4.8.3.	Configuración de modelos	88
4.9.	Resultados cuantitativos	89
4.9.1.	Autoencoder apilado aplicado a dataset 1	89
4.9.2.	Autoencoder convolucional apilado aplicado a dataset 1	89
4.9.3.	Autoencoder apilado aplicado a dataset 2	90
4.9.4.	Autoencoder convolucional apilado aplicado a dataset 2	90
4.9.5.	Autoencoder apilado aplicado a dataset 3	90
4.9.6.	Autoencoder convolucional apilado aplicado a dataset 3	91
4.9.7.	Autoencoder apilado aplicado a dataset 4	91
4.9.8.	Autoencoder convolucional apilado aplicado a dataset 4	91
4.10.	Resultados cualitativos	92
4.11.	Resultados de las series de tiempo ilustradas en mapa	93
4.11.1.	Dataset 1(Autoencoder apilado)	94
4.11.2.	Dataset 1(Autoencoder convolucional apilado)	96
4.11.3.	Dataset 2(Autoencoder apilado)	98
4.11.4.	Dataset 2(Autoencoder convolucional apilado)	102
4.11.5.	Dataset 3(Autoencoder apilado)	106
4.11.6.	Dataset 3(Autoencoder convolucional apilado)	108
4.11.7.	Dataset 4(Autoencoder apilado)	110
4.11.8.	Dataset 4(Autoencoder convolucional apilado)	114
4.12.	Arquitectura del mejor modelo	118

4.13. Métricas de aprendizaje y validación del mejor modelo	119
4.13.1. Métrica de función de costo	119
4.13.2. Métrica de coeficiente de determinación	120
4.13.3. Métrica de varianza explicada	120
4.14. Criterio para la detección de series de tiempo outlier	121
4.15. Resultado del árbol de decisión	122
4.15.1. Explicación del árbol de decisión	122

V Capítulo 5 **126**

5. Conclusiones	127
5.1. Discusión de los resultados	127
5.2. Conclusiones del proyecto	128
5.3. Trabajo a futuro	128
A. Exploración de datos	130
A.1. Comportamiento de variables	130
A.1.1. Variable latitud	130
A.1.2. Variable longitud	131
A.1.3. Variable altitud	131
A.1.4. Variable velocidad	132
A.1.5. Variable número de satélites	132
B. Ejemplos de series de tiempo	133
B.0.1. Dataset 1(Autoencoder apilado)	133
B.0.2. Dataset 1(Autoencoder convolucional apilado)	136
B.0.3. Dataset 2(Autoencoder apilado)	140
B.0.4. Dataset 2(Autoencoder convolucional apilado)	145
B.0.5. Dataset 3(Autoencoder apilado)	150
B.0.6. Dataset 3(Autoencoder convolucional apilado)	153
B.0.7. Dataset 4(Autoencoder apilado)	156
B.0.8. Dataset 4(Autoencoder convolucional apilado)	163

C. Arquitecturas de redes neuronales	168
C.1. Dataset 1	168
C.1.1. Autoencoder apilado	168
C.1.2. Autoencoder convolucional apilado	169
C.2. Dataset 2	172
C.2.1. Autoencoder apilado	172
C.2.2. Autoencoder convolucional apilado	173
C.3. Dataset 3	176
C.3.1. Autoencoder convolucional apilado	177
C.4. Dataset 4	180
C.4.1. Autoencoder apilado	180
C.4.2. Autoencoder convolucional apilado	182
D. Curvas de aprendizaje y validación	185
D.1. Dataset 1(Autoencoder apilado)	185
D.1.1. Arquitectura A	185
D.1.2. Arquitectura B	187
D.1.3. Arquitectura C	188
D.2. Dataset 1(Autoencoder convolucional apilado)	189
D.2.1. Arquitectura A	189
D.2.2. Arquitectura B	190
D.2.3. Arquitectura C	191
D.3. Dataset 2(Autoencoder apilado)	192
D.3.1. Arquitectura A	192
D.3.2. Arquitectura B	193
D.3.3. Arquitectura C	194
D.4. Dataset 2(Autoencoder convolucional apilado)	195
D.4.1. Arquitectura A	195
D.4.2. Arquitectura B	196
D.4.3. Arquitectura C	197
D.5. Dataset 3(Autoencoder apilado)	198

D.5.1. Arquitectura A	198
D.5.2. Arquitectura B	199
D.5.3. Arquitectura C	200
D.6. Dataset 3(Autoencoder convolucional apilado)	201
D.6.1. Arquitectura A	201
D.6.2. Arquitectura B	202
D.6.3. Arquitectura C	203
D.7. Dataset 4(Autoencoder apilado)	204
D.7.1. Arquitectura A	204
D.7.2. Arquitectura B	205
D.7.3. Arquitectura C	206
D.8. Dataset 4(Autoencoder convolucional apilado)	207
D.8.1. Arquitectura A	207
D.8.2. Arquitectura B	208
D.8.3. Arquitectura C	209

Índice de figuras

1.1.	Ejemplo de outlier	17
2.1.	Esquema de aprendizaje supervisado	24
2.2.	Esquema de aprendizaje no supervisado	24
2.3.	Programación tradicional vs aprendizaje automático	25
2.4.	Neurona biológica vs neurona artificial	26
2.5.	Modelo de perceptrón simple	28
2.6.	Modelo del perceptrón multicapa	29
2.7.	Método de descenso de gradiente	33
2.8.	Arquitectura Autoencoder	34
2.9.	Arquitectura Autoencoder Stack	35
2.10.	Operación convolucional extraída de Goodfellow y cols. (2016)	37
2.11.	Convolucional transpuesta	39
2.12.	Arquitectura de red autoencoder convolucional	39
2.13.	Ejemplo de outlier	41
2.14.	Metodo convex hull	41
2.15.	Algoritmo k-means	43
2.16.	Algoritmo local outlier factor	45
2.17.	Árbol de decisión extraído de Pedregosa y cols. (2011)	48
3.1.	Modelo cascada	51
3.2.	Modelo CRISP-DM	52
3.3.	EDT parte 1	56
3.4.	EDT parte 2	56

3.5.	Cronograma del proyecto, pagina 1	58
3.6.	Cronograma del proyecto, pagina 2	58
3.7.	Cronograma del proyecto, pagina 3	59
3.8.	Cronograma del proyecto, pagina 4	59
3.9.	Transformación de datos a 1-D	68
3.10.	Transformación de datos a 2-D	68
4.1.	Solución de alto nivel	75
4.2.	Modelo 4+1 de Kruchten	76
4.3.	Diagrama de casos de uso	77
4.4.	Diagrama de clases	78
4.5.	Diagrama de componentes	78
4.6.	Diagrama de despliegue	79
4.7.	Diagrama de secuencia	79
4.8.	GUI01-Inicio sesión	80
4.9.	GUI02-Historial de rutas	80
4.10.	Pruebas unitarias	81
4.13.	Modelo propuesto	87
4.46.	Arquitectura de autoencoder apilado	118
4.47.	Función de costo error cuadrático medio	119
4.48.	Métrica coeficiente de determinación	120
4.49.	Métrica varianza explicada	120
4.50.	Umbral para la detección de series de tiempo outlier	121
A.1.	Histograma latitud	130
A.2.	Histograma longitud	131
A.3.	Histograma altitud	131
A.4.	Histograma velocidad	132
A.5.	Histograma número de satélites	132
C.1.	Arquitectura A	168
C.2.	Arquitectura B	168

C.3. Arquitectura C	169
C.4. Arquitectura A	169
C.5. Arquitectura B	170
C.6. Arquitectura C	171
C.7. Arquitectura A	172
C.8. Arquitectura B	172
C.9. Arquitectura C	173
C.10.Arquitectura A	173
C.11.Arquitectura B	174
C.12.Arquitectura C	175
C.13.Arquitectura A	176
C.14.Arquitectura B	176
C.15.Arquitectura C	176
C.16.Arquitectura A	177
C.17.Arquitectura B	178
C.18.Arquitectura C	179
C.19.Arquitectura A	180
C.20.Arquitectura B	181
C.21.Arquitectura C	181
C.22.Arquitectura A	182
C.23.Arquitectura B	183
C.24.Arquitectura C	184

Índice de cuadros

1.1. Tabla de objetivos	18
2.1. Tabla de comparación de algoritmos	46
3.1. Tabla de gestión de alcance	57
3.2. Tabla de costos	60
3.3. Tabla de activos	60
3.4. Tabla de costos total	60
3.5. Tabla de riesgos	61
3.6. Tabla de descripción notebook	62
3.7. Tabla de descripción servidor	62
3.8. Tabla de descripción librerías	63
3.9. Tabla de descripción de variables	65
3.10. Tabla de descripción cuantitativa de variables	67
3.11. Tabla de valoración, fuente:Autor	72
4.1. Tabla de requerimientos funcionales	75
4.2. Tabla de requerimientos no funcionales	76
4.3. Tabla de resumen de conjuntos de entrenamientos	81
4.4. Tabla de resultados autoencoder apilado	89
4.5. Tabla de resultados autoencoder convolucional apilado	89
4.6. Tabla de resultados autoencoder apilado	90
4.7. Tabla de resultados autoencoder convolucional apilado	90
4.8. Tabla de resultados autoencoder apilado	90

4.9. Tabla de resultados autoencoder convolucional apilado	91
4.10. Tabla de resultados autoencoder apilado	91
4.11. Tabla de resultados autoencoder convolucional apilado	92
4.12. Tabla de resultados cualitativos	92
4.13. Tabla de regla de decisión para series de tiempo outlier	123
4.14. Tabla de regla de decisión para series de tiempo normales	124
5.1. Tabla de resultados de objetivos específicos	128

Parte I

Capítulo 1

Capítulo 1

Introducción

1.1. Introducción

En los últimos años la humanidad ha generado una gran cantidad de datos por parte de empresas y usuarios, los cuales siguen creciendo debido a la masificación de los sistemas de información. Este gran volumen de datos típicamente necesita ser canalizado y analizado para ser transformado en información útil para las personas involucradas en el análisis de datos, estos volúmenes de datos son en general enormes y además poco estructurados, por lo cual son muy difíciles de analizar manualmente. Las técnicas de Big Data permiten extraer información útil de estas grandes bases de datos combinando técnicas como modelos predictivos, inteligencia artificial y estadística, considerando la eficiencia computacional.

Una alternativa reciente dentro del análisis Big Data es el uso del aprendizaje automático, el cual consiste en aprender a reconocer automáticamente patrones a partir de cantidades ingentes de datos, con los cuales es factible tomar decisiones de forma autónoma que ayudan a agilizar tareas y procedimientos disminuyendo costos y tiempo.

Un aspecto clave a considerar es que estas grandes cantidades de datos existe información errónea que se desvía de su comportamiento habitual, denominados en la literatura especializada, como datos outlier, datos atípicos o datos anómalos, para este proyecto se usará la denominación datos outlier. Típicamente los datos outlier son descartados porque generan desvíos al momento de generalizar el comportamiento de un conjuntos de datos,

pero también suscitan gran interés en la detección de estos datos outlier como es el monitoreo de actividades criminales, la detección de fraudes en tarjetas de crédito, etc.(Taquias y cols. (2014)). Existen diferentes técnicas para la detección de datos outlier como técnicas estadísticas, técnicas basadas en distancia y técnicas de agrupamiento (Urgiles y Amoroso (2018)). Sin embargo en los últimos años el uso del deep learning que lleva a cabo el proceso de aprendizaje de automático usando una red neuronal con múltiples capas, ha obtenido resultados con alta precisión en distintas áreas como diagnósticos médicos, sistemas financieros, sistemas de recomendación, etc.(González (2015)). Dado el excelente rendimiento del deep learning, en este trabajo se propone el uso de redes neuronales profundas no supervisadas para la detección de datos anómalos. Específicamente para este proyecto se proponen dos modelos para la detección de comportamientos anómalos en series de tiempo usando deep learning.

El primer modelo que se propone es una red de autoencoders apilados, esta red captura una representación de los principales patrones de los datos de entrada en un espacio de variables latentes comúnmente llamado espacio latente, para luego reconstruir la salida de la red en base al espacio latente, dado que los datos outlier a menudo se corresponden con características no representativas, es probable que la red autoencoder apilado no pueda reconstruir los valores outlier utilizando el espacio latente, esta reconstrucción nos permite identificar los valores outlier como aquellos registros que no se explican bien usando el espacio latente. El segundo modelo propuesto es una variante de la red autoencoder apilado, la red de autoencoders convolucionales apilados, donde el principio es el mismo que la red autoencoder apilado. Este modelo usa la operación convolucional, la cual permite extraer características de los datos de entrada a través de una matriz, lo cual nos permite conocer los patrones locales de las características.

Para el desarrollo e implementación de los modelos se tomo como caso de estudio el sector de la telemetría junto con la empresa Waypoint dedicada a este sector. La telemetría genera una gran cantidad de información relevante solo en el monitoreo de vehículos donde también existe datos outlier generados por dispositivos GPS instalados en los vehículos, cada vehículo envía información de su posición y los estados de sus sensores en un determinado tiempo. El problema en cuanto al monitoreo de vehículos

con GPS, es que éstos pueden dar indicaciones erróneas si se encuentran en una zona sin cobertura digital, las cuales son almacenadas en la base de datos de la empresa.

1.2. Descripción del problema

Waypoint Telecomunicaciones S.A es una empresa dedicada a la entrega de servicios de telemetría y localización GPS de activos, bienes, vehículos y personas. El principal servicio que brinda es la gestión y localización instantánea e histórica de vehículos, personas o activos a través de dispositivos GPS para el control de reportes asociados al negocio. Waypoint tiene a cargo una flota aproximada de 30.000 vehículos de distintos clientes en todo Chile, por tal motivo cuenta con el módulo historial de rutas en donde el cliente puede visualizar el recorrido de un determinado vehículo, esto es posible porque cada vehículo tiene asociado un dispositivo GPS el cual envía información de su posición y los estados de sus sensores en un determinado tiempo, sin embargo por motivos como interferencia de señal o zonas sin cobertura digital estos datos son enviados a la plataforma errados y guardando valores engañosos difíciles de detectar. Dado que existen millones de registros en la base de datos, la realización de un procesamiento manual de estos millones de registros es infactible, mas aún no se tiene criterio totalmente especificado de como identificar un dato como outlier.

Los clientes de Waypoint monitorean los trayectos realizados por sus vehículos basado en el historial de rutas, el módulo historial de ruta muestra los trayectos de los vehículos entre una fecha de inicio y fecha final de un determinado vehículo, estos trayectos están formados por un conjunto de lecturas de GPS en un determinado instante, a estas lecturas la llamaremos observaciones las cuales tienen 6 características (Fecha, Latitud, Longitud, Altitud, Velocidad, Número de satélites). Aquellos observaciones que tengan un comportamiento distinto o anómalo a otros son llamados observaciones outlier, un comportamiento anómalo respecto al dispositivo GPS es la posición del vehículo (latitud, longitud, altitud) que se ve reflejado en la visualización de un trayecto de vehículo donde algunos de los puntos georreferenciados en el mapa (punto rojo) están alejados de aquellos puntos georreferenciados que tienen un comportamiento normal (punto azul) como en la figura 1.1.

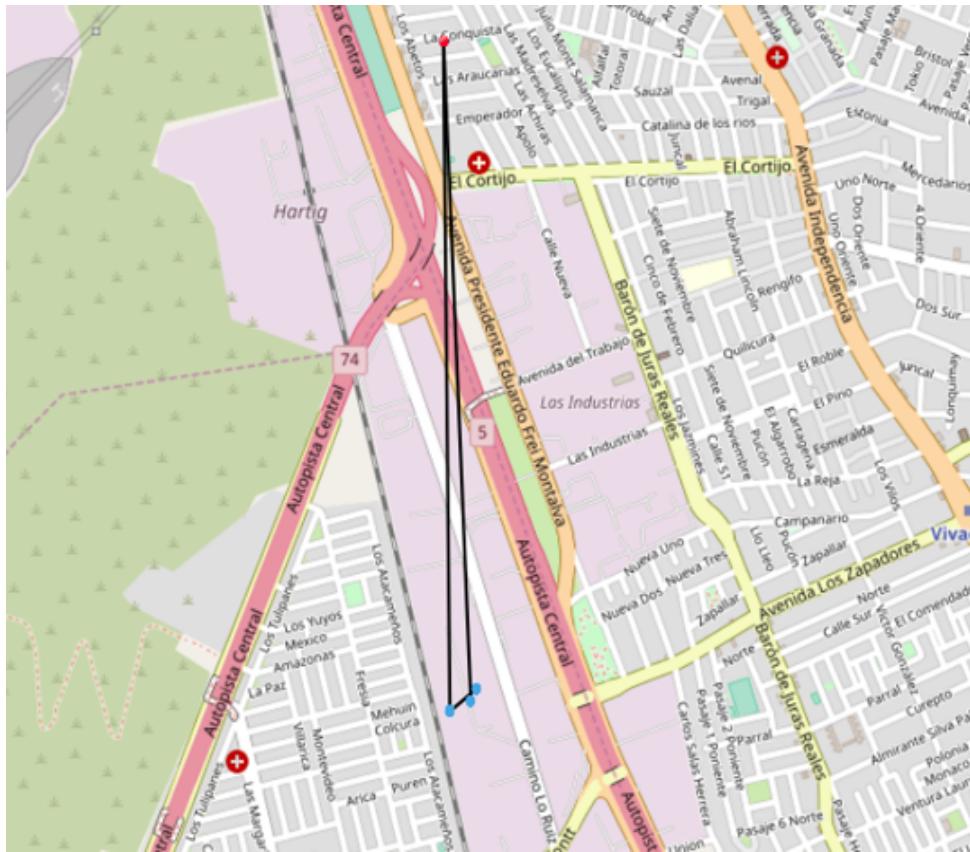


Figura 1.1: Ejemplo de outlier

Pero también están aquellos trayectos donde algunas observaciones se comportan de forma anómala pero no son detectados de forma visual sino tienen que ser examinados los valores de esa observación para determinar que tiene un comportamiento anómalo, tal caso es con la variable velocidad donde las respectivas empresas que usan la plataforma auditán a sus conductores con los datos de la velocidad que son proporcionados por la plataforma, si los datos son incorrectos estos proyectan falencias a los conductores que son sancionados y amonestados. El módulo historial de rutas no cuenta con una función predefinida que detecte estos comportamientos anómalos en los trayectos de los vehículos por lo cual es necesario el desarrollo de un API(Application Programming Interface) que ayude a la detección de trayectos outlier y complementariamente detectar aquellas observaciones anómalas, retornando una etiqueta de aquellas observaciones que sean detectadas como outlier y no outlier.

1.3. Alcance

1.3.1. Objetivo general

Automatizar la detección de trayectos outlier de vehículos basado en el historial de rutas de la flota de vehículos de Waypoint.

1.3.2. Objetivos específicos

1. Entender las variables que puedan ser usados como predictoras para identificar trayectos outlier.
2. Desarrollar e implementar modelos de red neuronal que detecte trayectos de vehículos outlier exitosamente, basados en el historial de rutas de vehículos de Waypoint.

Id	Situación actual	Resultado esperado	Métrica	Criterio de éxito
1	No se tiene un análisis de las variables predictoras.	Realizar un análisis de cada variable.	Número de variables a analizar.	Analizar al menos 3 variables.
2	No se cuenta con ningún modelo para la detección de trayectos outlier.	Modelo de red neuronal entrenado e implementado en el módulo historial de rutas.	-Número de trayectos outlier detectados. -Coeficiente de determinación	-El modelo detectó al menos un 30 % de trayectos outlier. -La precisión del modelo debe ser sobre el 90 %.

Cuadro 1.1: Tabla de objetivos

1.3.3. Limites

- El modelo de red neuronal solo detecta trayectos outlier e indicara aquellas observaciones que son outlier.
- Para la detección de trayectos outlier solo se tomará en cuenta los datos del historial de rutas de vehículos mas no los datos que se estén generando en tiempo real.

1.3.4. Supuestos

- Las observaciones de los trayectos deben estar ordenados en orden cronológico y completos.
- Se cuenta con los datos históricos de trayectos para el desarrollo e implementación de la red neuronal.

Parte II

Capítulo 2

Capítulo 2

Fundamentación del tema

2.1. Estado del arte

La detección de series de tiempo outlier ha tomado interés debido a que esta tiene un alto impacto en amplia variedad de aplicaciones y por lo tanto se ha estudiado ampliamente en múltiples áreas de investigación, a continuación se describen trabajos relacionados a la detección de series de tiempo outlier.

Yan y Yu (2015) plantea un diseño para la detección de anomalías con enfoque de aprendizaje no supervisado para observaciones multivariadas, usan una red stacked denoising autoencoders (SDAE), la cual aprende características mas robustas para el ruido de la entrada, las características aprendidas del SDAE se toman como entrada para una máquina de aprendizaje extremo (ELM) para clasificar si una observación es anómala o no, una ELM tiene conexiones entre las neuronas de entrada y las ocultas que se generan y arreglan al azar.

Singh (2017), centra su trabajo en la detección de anomalías en series de tiempo para observaciones univariadas. El modelo propuesto de Singh tiene un enfoque de aprendizaje no supervisado, donde combinan una red de memoria a largo y corto plazo(LSTM) y una red neuronal recurrente (RNN). El modelo LSTM-RNN es entrenado solamente con datos normales y sin anomalías, de esta forma el modelo aprende el comportamiento normal de las series de tiempo; así cuando el modelo se utilice para predicciones de datos nuevos, este dará un error de predicción más alto en regiones con anomalías en comparación con

regiones normales.

Ma y cols. (2017) proponen un método para detectar observaciones anómalas en series de tiempo, donde solo se aplica a observaciones univariadas. El modelo divide las series de tiempo en subsecuencias mediante una ventana deslizante, entonces se aplica un modelo de predicción autorregresivo (AR) a cada subsecuencia para predecir el valor del siguiente punto, calculan un intervalo de confianza de predicción (PCI) a partir de los datos históricos del vecino más cercano, si el valor predicho cae fuera del PCI, se considerará un valor outlier.

Lu y cols. (2017) proponen la detección de series de tiempo outlier multivariantes con un enfoque no supervisado usando deep learning, su modelo integra una red denoising autoencoder (DAE) que se usa para extraer automáticamente características de datos sin procesar y una red recurrente (RNN) que se usa para modelar la estructura temporal, el objetivo de la red DAE es capturar la diferencia intrínseca en la densidad entre los valores outlier y las instancias normales, mientras que de la red RNN es obtener información contextual y, por lo tanto, mejorar la construcción de las características. Para la identificación de outlier se uso el error de reconstrucción, si la serie de tiempo tiene un error de reconstrucción alto es considerado outlier.

Kieu y cols. (2018) presento dos modelos para la detección de series de tiempo anómalos para observaciones multivariantes usando una red neuronal convolucional basado en autoencoder(2DCNN-AE) y una red Long Short Term Memory basado en autoencoder(LSTM-AE). Además proponen un método para enriquecer series temporales multidimensionales para capturar diferentes aspectos de los cambios temporales en las series temporales y transformándolas en matrices para que pueden ser procesadas por 2DCNN-AE, para la red LSTM-AE se tomaron las matrices y se concateno cada fila transformándolo en un vector unidimensional. Para la identificación de outlier se uso el error de reconstrucción, si la serie de tiempo tiene un error de reconstrucción alto es considerado outlier.

Los modelos que proponen Yan y Yu (2015), Lu y cols. (2017) y Kieu y cols. (2018), utilizan redes neuronales autoencoder, los tres autores usan esta arquitectura para extraer características representativas de las series de tiempo a través de una red encoder, las cuales servirán para diferenciar las series de tiempo outlier; los modelos propuestos

para este proyecto tienen como finalidad extraer las características representativas para luego ser reconstruidas por una red decoder. Además Singh (2017), Kieu y cols. (2018) y Lu y cols. (2017) usan redes RNN y LSTM, los autores usan esta arquitectura para proporcionar una mejor comprensión del contexto completo de las trayectorias pasadas, para este proyecto no se utilizará estas redes porque lo que se busca es encontrar relaciones locales de las observaciones. Los modelos propuestos por Singh (2017) y Ma y cols. (2017) se orientan a series de tiempo univariadas, por lo cual no son aplicables a este proyecto, ya que se utilizan series de tiempo multivariadas.

2.2. Generalidades de aprendizaje automático

Aprendizaje automático o machine learning es una rama de la inteligencia artificial que se encarga del estudio, desarrollo e implementación de algoritmos que aprenden a reconocer patrones o conclusiones obtenidas de una gran cantidad de datos, sin que el ser humano tenga que escribir instrucciones para ello. Una característica muy importante de los algoritmos de aprendizaje automático es que tienen la capacidad de generalizar y asociar nuevos casos basándose en la experiencia aprendida del conjunto de datos utilizados en su entrenamiento. Específicamente el aprendizaje automático se divide en dos tipos de aprendizaje, el aprendizaje supervisado y el aprendizaje no supervisado, los cuales se describirán a continuación.

2.2.1. Aprendizaje supervisado

El aprendizaje supervisado es un tipo de aprendizaje que entrena un algoritmo a partir de datos de entrenamiento, se le indica que es lo que debería generar el algoritmo como salida, con esta información el algoritmo calcula cuál debería ser el cambio en sus parámetros de acuerdo a Pajares y De La Cruz (2010). El objetivo del aprendizaje supervisado es crear una función capaz de predecir el valor correspondiente dado una entrada válida, conocer la salida del conjunto de entrenamientos implica que el aprendizaje del algoritmo es supervisado por un experto, de ahí el nombre de aprendizaje supervisado a continuación se muestra esquema de aprendizaje supervisado en la figura 2.1.

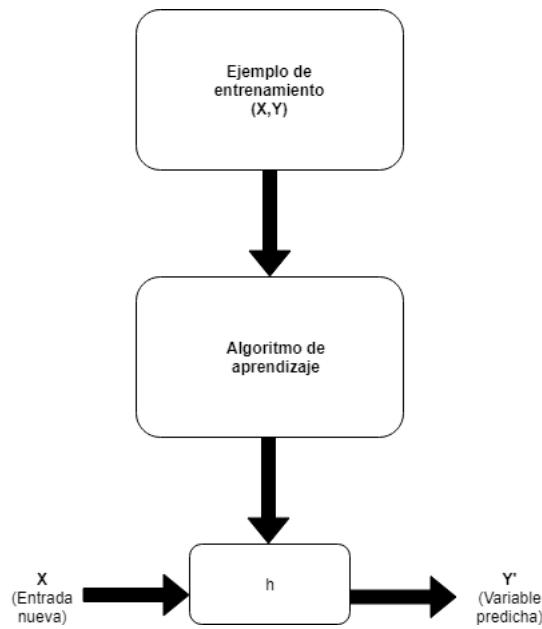


Figura 2.1: Esquema de aprendizaje supervisado

2.2.2. Aprendizaje no supervisado

De igual forma que el aprendizaje supervisado, el aprendizaje no supervisado es un tipo de aprendizaje que entrena el algoritmo a partir de datos de entrada, pero este no requiere de información externa sobre si la respuesta es correcta o no a una entrada, solo necesita los datos de entrada para ajustar los parámetros del algoritmo, según señala Pajares y De La Cruz (2010). El objetivo del aprendizaje no supervisado es encontrar patrones a partir de las propiedades de los ejemplos de entrenamiento se trata de agrupar y clasificar según las propiedades de los ejemplos, se muestra esquema de aprendizaje no supervisado en la figura 2.2.



Figura 2.2: Esquema de aprendizaje no supervisado

2.2.3. ¿Cuándo utilizar aprendizaje automático?

Se puede desarrollar soluciones utilizando la programación tradicional que utiliza bucles y condicionales para determinados problemas, por ejemplo, determinar un valor de predicción de interés a través de sencillas reglas, cálculos o pasos predeterminados que pueden programarse sin necesidad de ningún tipo de aprendizaje basado en datos. Pero cuando el problema no se puede resolver mediante sencillas reglas debido a que el problema depende de muchas variables o la relación entre las variables es compleja y ambigua, el programador no podrá codificar las reglas de manera precisa. Para estos casos es útil el aprendizaje automático, la figura 2.3 muestra la diferencia entre programación tradición y aprendizaje automático.

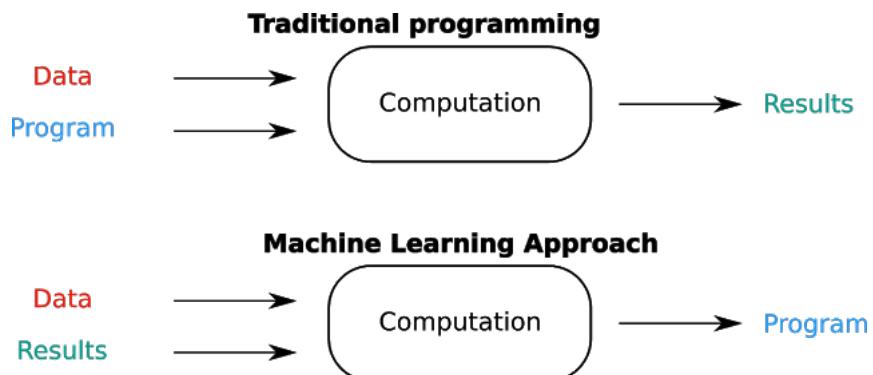


Figura 2.3: Programación tradicional vs aprendizaje automático

2.3. Redes neuronales artificiales

Las redes neuronales artificiales son sistemas conexionistas inspirados en la capacidad humana de procesar información, esta capacidad humana se debe a la naturaleza biológica de nuestro cerebro, por lo tanto, las redes neuronales artificiales imitan esta característica basando en soportes artificiales iguales a las neuronas biológicas del cerebro.

Una neurona biológica recibe información de otra neurona a través de las sinapsis de sus dendritas y cada sinapsis representa la unión de un axón de otra neurona. Para la transmisión de información entre neuronas se lleva a cabo un proceso electro-químico, en este proceso la información es transmitida hasta llegar al cuerpo de la neurona, allí se suma todos los impulsos electromagnéticos provenientes de otras neuronas y se aplica una

función de activación a este. La neurona se activará si el resultado cumple con un umbral determinado de esta manera es como funcionan las redes neuronales.

Homologando este proceso biológico a un proceso artificial se tiene un conjunto de neuronas representados por nodos que están interconectados entre sí, las interconexiones son representadas por los axones que producen un número variable de conexiones(sinapsis) que la llamaremos pesos sinápticos, representación de neurona biología y neurona artificial en la figura 2.4.

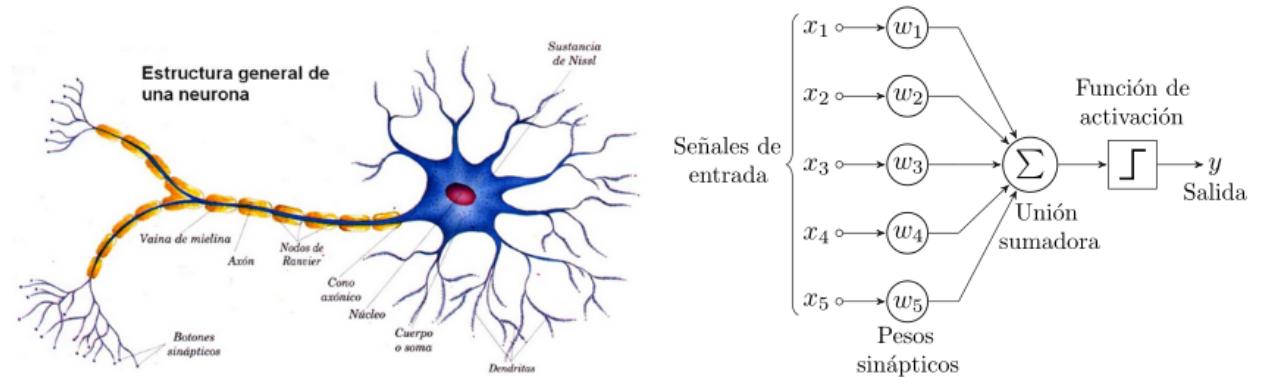


Figura 2.4: Neurona biológica vs neurona artificial

2.3.1. Ventajas de redes neuronales artificiales

Como las redes neuronales artificiales se asemejan a las del cerebro, están tienen las mismas características como lo refiere Maren y cols. (1990).

- **Aprendizaje adaptativo.** - Capacidad de aprender a realizar tareas basadas en un entrenamiento o una experiencia inicial.
- **Auto-organización.** – Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
- **Tolerancia a fallos.** - La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.
- **Operación en tiempo real.** – Los computadores neuronales pueden ser realizados

en paralelo, y se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.

- **Fácil inserción dentro de la tecnología existente.** - Se puede obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes.

2.3.2. Aprendizaje en las redes neuronales artificiales

Aprendizaje supervisado

Con este tipo de aprendizaje, alimentamos a la red con un conjunto de datos de entrada y respuesta correcta, los datos se propagan hacia delante hasta alcanzar la capa de salida. Entonces se compara la salida de la red con la respuesta correcta, esto ejecutara una acción correctiva ajustando los pesos de la red para que esta produzca una respuesta más probable a la respuesta correcta, esto se itera hasta que la red aprenda la respuesta deseada según Hernández Orallo y cols. (2004).

Aprendizaje no supervisado

Con este tipo de aprendizaje, alimentamos a la red con solo un conjunto de datos de entrada y no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. Estas redes deben encontrar las características y correlaciones que se puedan establecer entre los datos de entrada, este tipo de aprendizaje es útil para tarea de agrupamiento, detección de outlier, etc. de acuerdo a Hernández Orallo y cols. (2004)

2.4. Perceptrón multicapa

El perceptrón simple fue inicialmente investigado por Rosenblatt en 1962, esta red consta de una capa de entrada de n neuronas y una capa de salida con una o más neuronas de salida, no tiene capas ocultas. Esta neurona está limitada a la resolución de problemas linealmente separables, además que utiliza señales binarias y usa la función de activación escalón, en la figura 2.5 se presenta el modelo de un perceptrón simple. El perceptrón recibe

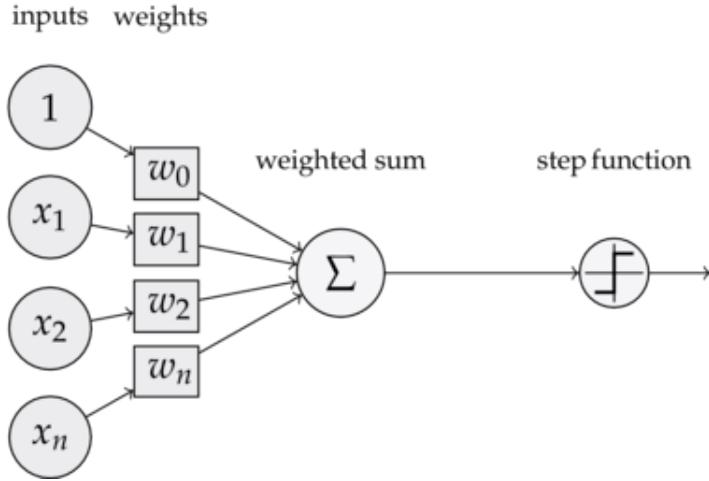


Figura 2.5: Modelo de perceptrón simple

señales de entrada o atributos de la fuente de datos representados por un vector X tal que $X = \{x_1, x_2, x_3, \dots, x_n\}$, y los pesos representados por un vector $W = \{w_0, w_1, w_2, \dots, w_n\}$ la figura 2.5 incluye una entrada externa adicional denominada bias la cual la denotaremos por b y tiene un valor igual a $b = 1 * w_0$, su finalidad es poder aumentar o disminuir el umbral de excitación de la neurona, entonces el valor de una neurona esta dado por la ecuación 2.1.

$$z = x_1w_1 + x_2w_2 + x_3w_3 + \dots + x_nw_n + b = \sum_{i=1}^n (x_iw_i) + b \quad (2.1)$$

Donde z es una combinación lineal del vector de entrada X y el vector de pesos W , luego se aplica la función de activación escalón dado por la ecuación 2.2, debiendo superar un umbral u , para la función escalón se toma como umbral el valor 0.

$$f(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases} \quad (2.2)$$

Al aplicar la función escalón al valor z obtenemos la salida de la neurona representado por a , ecuación 2.3.

$$a = f(z) \quad (2.3)$$

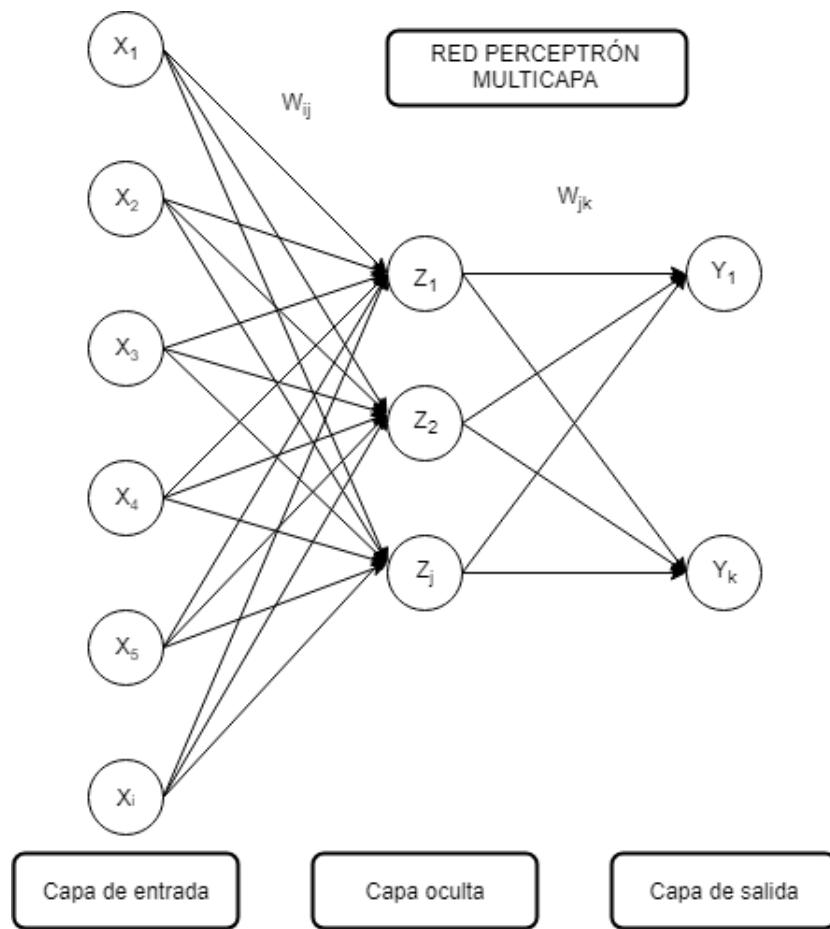


Figura 2.6: Modelo del perceptrón multicapa

Como se mencionó anteriormente el perceptrón simple está limitado a la resolución de problemas linealmente separables, se ha demostrado que muchos conjuntos de datos no son linealmente separables y que pueden ser modelados por el perceptrón multicapa(MLP) obteniendo una precisión alta en los resultado. Esta red neuronal está formada por múltiples capas en donde cada capa cuenta con un numero de neuronas. El modelo del perceptrón simple se replica en cada neurona de la red perceptrón multicapa, en la figura 2.6 se muestra la red perceptrón multicapa.

Entonces de la misma forma que el perceptrón simple, para cada neurona de la capa oculta se tiene una combinación lineal:

$$z_1 = x_1 w_{11} + x_2 w_{21} + x_3 w_{31} + \dots + x_i w_{i1} + b_1$$

$$z_2 = x_1 w_{12} + x_2 w_{22} + x_3 w_{32} + \dots + x_i w_{i2} + b_2$$

$$z_3 = x_1 w_{13} + x_2 w_{23} + x_3 w_{33} + \dots + x_i w_{i3} + b_3$$

.

.

$$z_j = x_1 w_{1j} + x_2 w_{2j} + x_3 w_{3j} + \dots + x_i w_{ij} + b_j$$

$$\forall i, j \in W$$

Esto se puede representar de forma matricial, la cual nos permite obtener ecuaciones más sencillas, definimos una matriz W que representa los pesos que llegan de la capa de entrada a la capa oculta, una matriz X que representa los datos de entrada, un vector B que representa el bias de cada combinación lineal y el vector Z que es la combinación lineal de cada neurona, entonces se obtiene la siguiente expresión:

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_j \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_i \end{pmatrix} \begin{pmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1j} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2j} \\ w_{31} & w_{32} & w_{33} & \dots & w_{3j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{i1} & w_{i2} & w_{i3} & \dots & w_{ij} \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_j \end{pmatrix}$$

La representación en matrices quedaría como la ecuación 2.4.

$$Z = XW + B \tag{2.4}$$

En el caso del perceptrón multicapa usa funciones de activación como la sigmoide, tangente hiperbólica, relu, etc. Antona Cortés (2017), la cual le da la propiedad de poder resolver problemas que no son linealmente separables, si asumimos la vectorización del

vector Z podemos considerar que la función de activación se aplica a cada elemento del vector, la salida de cada neurona esta dada por el vector A .

$$A = f(z) = \begin{pmatrix} f(z_1) \\ f(z_2) \\ f(z_3) \\ \vdots \\ f(z_j) \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_j \end{pmatrix}$$

La salida de la capa oculta viene dada por la ecuación 2.5.

$$A = f(Z) \quad (2.5)$$

Este procedimiento se aplica entre las capas intermedias u ocultas hasta llegar a la capa de salida, este procedimiento es más conocido como feed-forward o alimentación hacia adelante(Pajares y De La Cruz (2010)).

2.4.1. Algoritmo de aprendizaje

Para poder aprender las redes neuronales se sirven de un algoritmo de aprendizaje, un algoritmo de aprendizaje es un método adaptativo por el cual la red se auto modifica de tal forma que se implementa un comportamiento deseado, el perceptrón multicapa utiliza el algoritmo de aprendizaje backpropagation o propagación hacia atrás(Pajares y De La Cruz (2010)). La característica importante de este algoritmo de aprendizaje es la capacidad de adaptar los pesos de las neuronas de las capas intermedias, podemos utilizar cualquier numero de capas intermedias ya que el algoritmo es bastante general, para la actualización de los pesos se usa el método del descenso de gradiente de acuerdo a Hernández Orallo y cols. (2004).

2.4.2. Descenso de gradiente

El descenso de gradiente es un método que minimiza una función y es uno de los algoritmos mas populares en el aprendizaje automático, una red neuronal esta asociada a una función de costo la cual indica el error que comete la red neuronal, la función de costo E depende de los pesos W de la red neuronal entonces podemos usar $E(w)$ para definir la función de costo. El objetivo es encontrar los valores óptimos de W tal que se obtiene un mínimo global de la función de costo $E()$, en consecuencia el aprendizaje de redes neuronales es un problema de optimización, ver ecuación 2.6.

$$\operatorname{argmin} E(w) \quad (2.6)$$

Para encontrar estos valores óptimos se hace uso de las derivadas parciales las cuales permitirán modificar los pesos de la red. Entonces la deriva parcial nos dará un vector gradiente representado por la ecuación 2.7:

$$\Delta E = \frac{\partial E(w)}{\partial w_{ij}} \forall i, j \in W \quad (2.7)$$

El descenso de gradiente modifica los pesos en base al peso anterior, la ecuación 2.8 muestra como se modifica los pesos de la red.

$$w_{i+1} = w_i - \alpha \Delta E \quad (2.8)$$

Donde α es la tasa de aprendizaje que se fija a priori y es constante para todo el entrenamiento, adaptamos el descenso de gradiente hasta encontrar una secuencia de valores W que se acercan al mínimo de la función, en la figura 2.7 se muestra el método de descenso de gradiente.

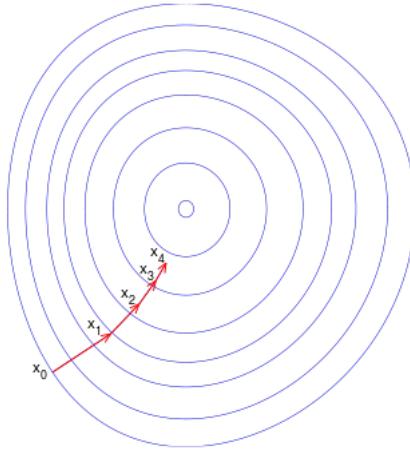


Figura 2.7: Método de descenso de gradiente

2.5. Redes Neuronales Autoencoder

2.5.1. Autoencoder

Un autoencoder es una red neuronal artificial que intenta copiar su entrada a su salida, internamente comprime los datos de entrada a una capa oculta h denominado espacio latente h con la cual reconstruye los datos de entrada (Goodfellow y cols. (2016)). Esta compresión de datos se basa en reducir el número de neuronas de la capa oculta, de esta forma la red es forzada a aprender una representación de los principales patrones de los datos de entrada.

Dado que los datos outlier a menudo se corresponden con características no representativas, es probable que la red autoencoder no pueda reconstruir los valores outlier utilizando el espacio latente, esta reconstrucción nos permite marcar como valores outlier aquellos datos que no se explican bien usando el espacio latente.

El proceso de reducir el número de neuronas de la capa oculta se denomina encoding, de igual forma que el perceptrón, el autoencoder utiliza una combinación lineal de los datos de entrada y los pesos de la red, representados por X y W respectivamente y una función de activación s , como en la ecuación 2.9.

$$h = s(XW + B) \quad (2.9)$$

El espacio latente h que representa las características más resaltantes de los datos de entrada, se usa para reconstruir los datos de entrada originales, para ello se aumenta el número de neuronas de la capa oculta a este proceso lo denominaremos decoding, \hat{x} es una reconstrucción de la entrada x dado el espacio latente h como en la ecuación 2.10.

$$\hat{x} = s(W'h + B') \quad (2.10)$$

La matriz de pesos W' es equivalente a la matriz transpuesta W^T ($W' = W^T$) los parámetros del autoencoder (W, W, B, B') son optimizados tal que minimizan la función de costo, típicamente se usa el error cuadrático medio(MSE), la cual se expresa en la ecuación 2.11.

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{x} - x)^2 \quad (2.11)$$

El error de reconstrucción es una puntuación por cada ejemplo que puede ser calculada como el error cuadrático medio entre el dato de entrada y dato reconstruido, esta puntuación nos indica cuan bien ha sido reconstruido el dato de entrada, a mayor sea el error de reconstrucción mayor será la posibilidad de que el dato sea outlier, en la figura 2.8 se muestra la arquitectura de una red autoencoder.

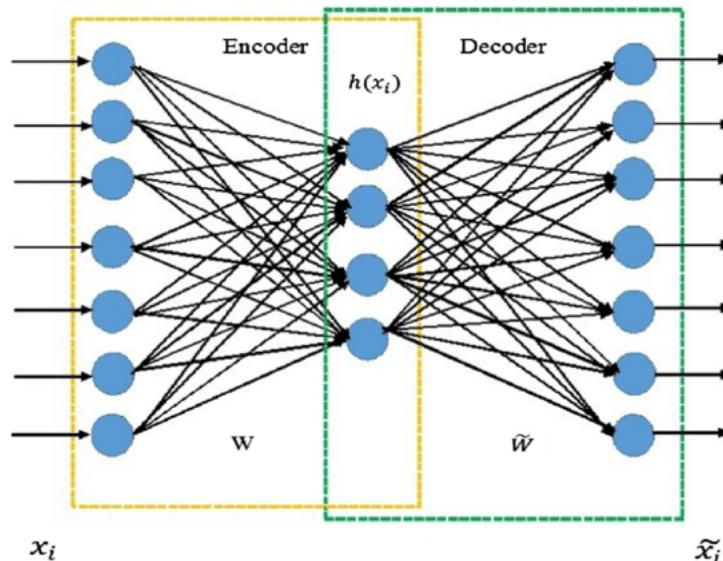


Figura 2.8: Arquitectura Autoencoder

2.5.2. Autoencoder apilado

Los autoencoder se pueden apilar para formar una red profunda, llamaremos autoencoder apilado a aquella red que tiene múltiples capas en el encoder y decoder, al agregar más capas ayuda al autoencoder a aprender codificaciones más complejas. La arquitectura de un autoencoder apilado es simétrica con respecto a la capa central oculta, entonces el autoencoder es visto como dos subredes, arquitectura de red autoencoder apilado figura 2.9.

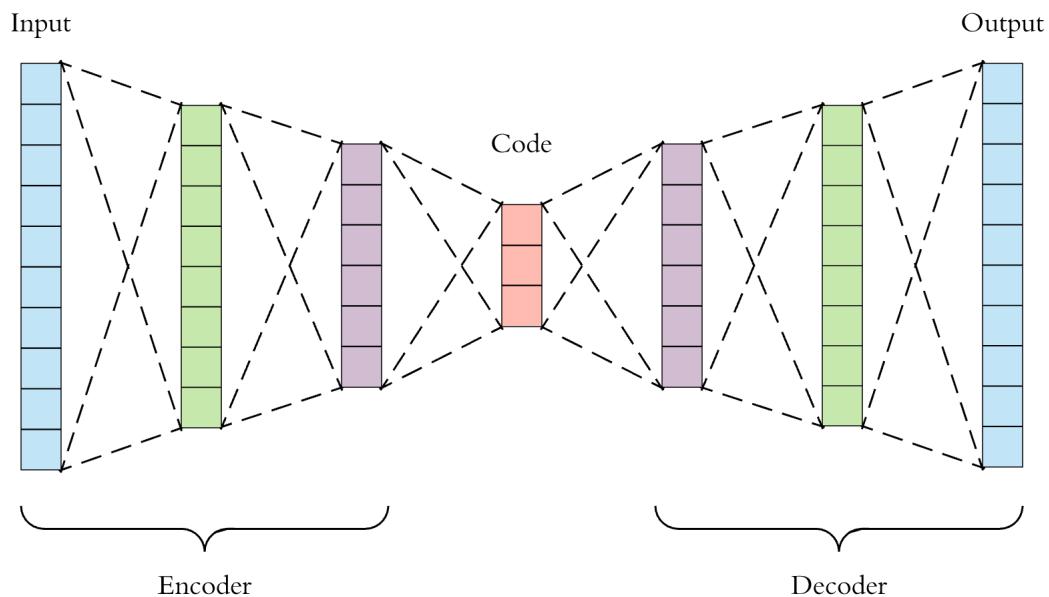


Figura 2.9: Arquitectura Autoencoder Stack

Red Encoder

Formada por la capa de entrada y un conjunto de capas ocultas, esta subred comprime los datos de entrada X a un espacio latente, el entrenamiento de dicha red se realiza de una capa a la vez. Cada capa se entrena como un encoder y reduciendo el número de neuronas, cada capa recibe como entrada la representación latente de la capa anterior hasta llegar a la capa oculta h que describe el espacio latente de la red, red encoder se representa mediante la ecuación 2.12.

$$h = f(x) \quad (2.12)$$

Red Decoder

Formada por un conjunto de capas ocultas y la capa de salida, esta subred reconstruye la entrada original \hat{x} en base al espacio latente h , el entrenamiento de dicha red se realiza de una capa a la vez. Cada capa se entrena como un decoder aumentando el número de neuronas, cada capa recibe como entrada la reconstrucción de la capa anterior hasta llegar a la capa de salida, la cual tiene el número de neuronas iguales al de la capa de entrada, red decoder se representa mediante la ecuación 2.13.

$$\hat{x} = g(h) \quad (2.13)$$

EL autoencoder apilado se puede representar por la ecuación 2.14.

$$g(f(x)) = \hat{x} \quad (2.14)$$

Sin embargo hay que tener cuidado de no hacer el autoencoder potente, por ejemplo si el encoder aprende a asignar cada entrada a un solo número arbitrario y el decoder aprender el mapeo inverso, tal autoencoder reconstruirá los datos de entrenamiento a la perfección, pero no habrá aprendido ninguna representación de datos útil en el proceso según señala Géron (2017).

2.6. Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN) surgieron del estudio de la corteza visual del cerebro y se utilizan típicamente para reconocimiento visual de imágenes, también se pueden utilizar para el procesamiento de datos que tienen una topología similar a una cuadrícula conocida, por ejemplo para datos de series de tiempo, se puede considerar como una cuadricula 1-D que toma muestras a intervalos de tiempo Goodfellow y cols. (2016). El nombre de red neuronal convolucional es porque emplea una operación matemática llamada convolucional, estas redes son muy potentes debido a que pueden detectar características simples y componer características más complejas usando filtros.

2.6.1. Operación convolucional

La operación convolucional es la multiplicación entre un filtro (kernel) y la entrada de la red, el filtro sirve para extraer las mismas características en cualquier parte de la entrada, con esto se consigue reducir el número de conexiones y números de parámetros a entrenar, el filtro contiene los pesos de la red. Además, el filtro debe recorrer la imagen, esto lo hace con un strider que son los pasos que da el filtro para recorrer la entrada, en la figura 2.10 se ilustra la operación convolucional y a continuación se muestra un ejemplo de la operación convolucional.

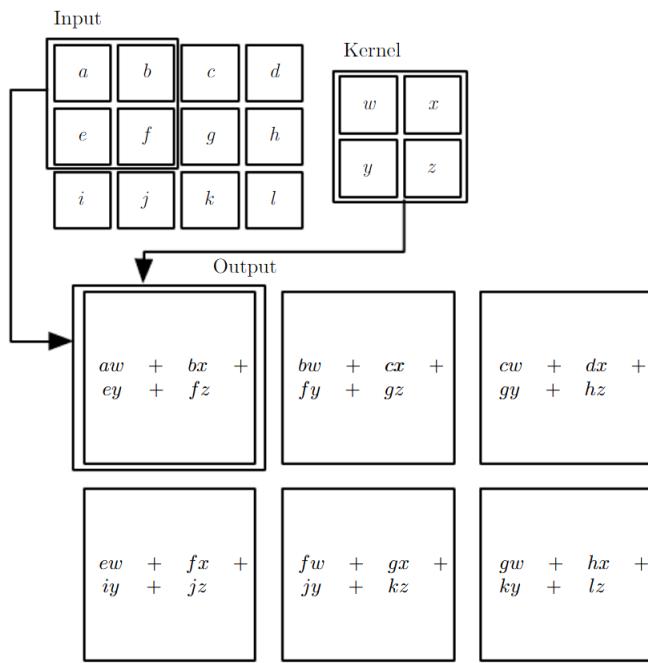


Figura 2.10: Operación convolucional extraída de Goodfellow y cols. (2016)

La entrada de la red X lo representamos como una matriz de 3x4 y el filtro W como una matriz 2x2 con los pesos de la red:

$$X = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{pmatrix} W = \begin{pmatrix} w & b \\ y & z \end{pmatrix}$$

El filtro multiplica los pesos de la red con los datos de entrada, recorriendo la imagen con un strider = 1, se obtiene las siguientes combinaciones lineales y de igual forma que los

anteriores modelos de red neuronal, se aplica una función de activación s de la siguiente forma:

$$\begin{aligned} a_1 &= s(aw + bx + ey + fz) \\ a_2 &= s(bw + cx + fy + gz) \\ a_3 &= s(cw + dx + gy + hz) = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{pmatrix} \\ a_4 &= s(ew + fx + iy + jz) \\ a_5 &= s(fw + gx + jy + kz) \\ a_6 &= s(gw + hx + ky + lz) \end{aligned}$$

Estas combinaciones son representadas en una matriz, siendo la salida de la capa convolucional para luego ser las entradas de la siguiente capa convolucional.

2.6.2. Convolucional transpuesta

Convolucional transpuesta o también llamada deconvolucional es la inversa de la operación convolucional, generalmente se usa para hacer una transformación que va en la dirección opuesta a una convolución normal manteniendo un patrón de conectividad compatible con dicha convolución.

Dado una matriz de entrada C , desde la perspectiva de un solo valor en la entrada, tomamos este valor y lo distribuimos a una vecindad de puntos en la salida, entonces se define un kernel k y para cada celda de salida multiplicamos el valor de entrada por el peso correspondiente del kernel. Repetimos este proceso para cada valor en la entrada y acumulamos valores en cada celda de salida, obteniendo una matriz de salida C' . En la figura 2.11 se ilustra el procedimiento de la convolucional transpuesta.

En azul, es la matriz de entrada C que se desea ampliar, en verde oscuro, es la matriz de salida C' y los cuadrados sombreados hacen referencia al kernel k con el que se convoluciona la matriz de entrada. Los cuadrados blancos representan valores nulos que se añaden a la matriz de entrada para que se pueda realizar el aumento de dimensionalidad.

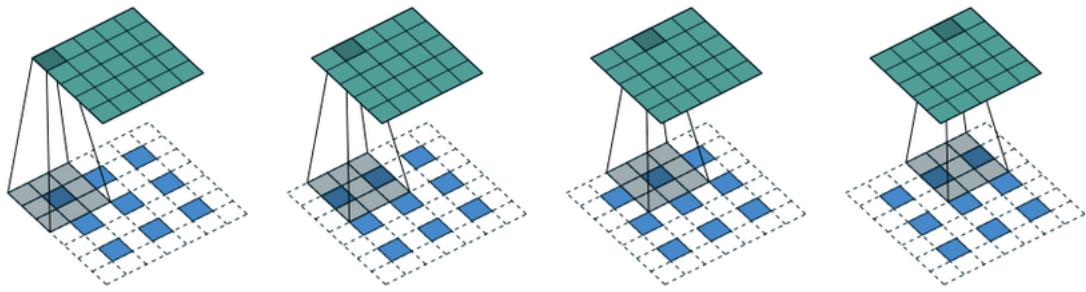


Figura 2.11: Convolucional transpuesta

2.7. Redes Neuronales Autoencoder Convolucionales apilados

Redes autoencoder convolucionales apilados(AE-CNN) son una variante de la red autoencoder apilado que usa capas convolucionales en vez de capas totalmente conectadas(Capas Densas), Las AE-CNN comparten sus pesos en todas las ubicaciones de la entrada de la red permitiendo preservar la localidad espacial y siguen el mismo esquema que la red autoencoder apilada descrita en la sección 2.5.2. Esta formado por una red encoder que usa capas convolucionales y una red decoder usa capas deconvolucionales, en la figura 2.12 se muestra la arquitectura de la red autoencoder convolucional apilado.

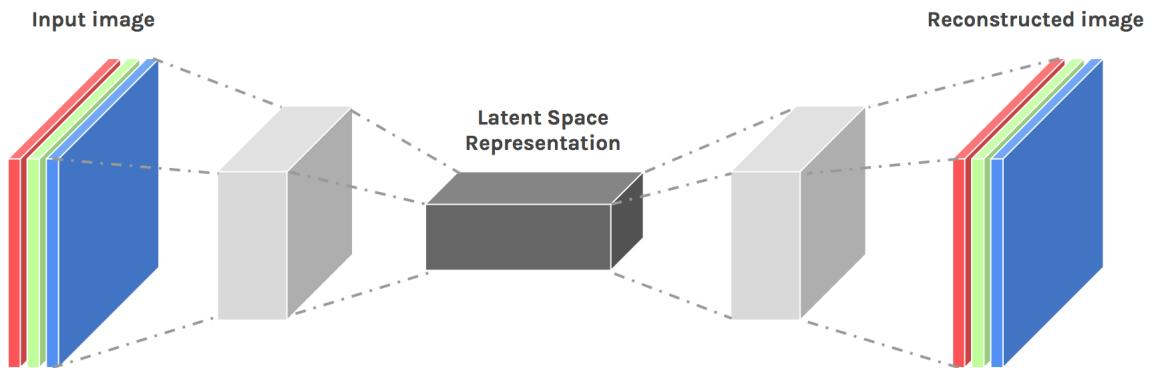


Figura 2.12: Arquitectura de red autoencoder convolucional

2.8. Series de tiempo

Las series de tiempo se definen como un conjunto de observaciones tomadas en un período de tiempo particular ordenadas y que toman valores en diferentes momentos del tiempo, los modelos de serie de tiempo tienen un enfoque netamente predictivo. Una serie de tiempo esta representado por $X_T = X_1, X_2, X_3, \dots, X_t$, donde X_t es una observación tomada en el instante t . Las observaciones pueden ser univariadas o multivariadas para este proyecto se trabajará con series de tiempo multivariantes.

Las observaciones pueden comportarse de diferentes formas a través del tiempo, donde algunas observaciones de la serie se comportan de forma anómala respecto a las demás observaciones, por lo cual a aquellos los denominaremos como observaciones anómalas.

2.9. Datos outlier o atípicos

Un dato outlier o valor atípico es aquella instancia en un conjunto de datos que tiene un patrón muy diferente a los encontrados en la mayoría de los datos estos datos son de naturaleza anómalo y escasa, por lo usual, son difíciles de detectar en grandes bases de datos. Los métodos de detección de outlier se pueden orientar a outlier univariadas y multivariadas. Para el caso de outlier univariadas se usan técnicas estadísticas como enfoques paramétricos y enfoques no paramétricos; para el caso de outlier multivariadas es mucho más difícil detectarlos debido a la dimensionalidad de los datos. La detección de datos outlier multivariados solo es posible realizando un análisis multivariable y las interacciones entre las diferentes variables. Un ejemplo de dato outlier es el punto que está en la parte superior derecha figura 2.13.

A continuación se describirán algunos métodos para la detección de outlier tomando como referencia el trabajo de Urgiles y Amoroso (2018).

2.9.1. Método Convex Hull

Este método define un polígono convexo que cubre todos los puntos, aquellos puntos que se encuentren fuera del polígono convexo se consideran outlier, ver figura 2.14, existen

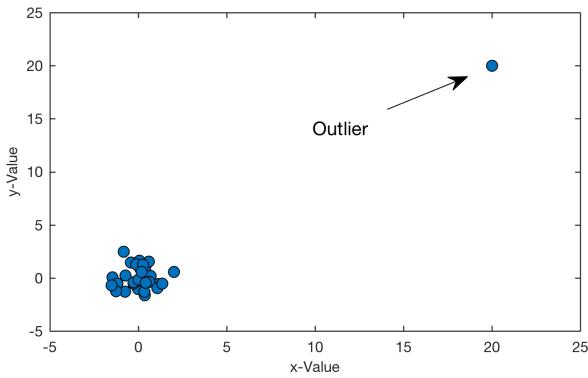


Figura 2.13: Ejemplo de outlier

distintos algoritmos para calcular el polígono y determinar cual es el mejor, el algoritmo mas utilizado y fácil de implementar es el Graham Scan(Kong y cols. (1990)). Este algoritmo encuentra los puntos que formaran parte del polígono, aquellos puntos que tienen mayor y menor valor deben pertenecer al polígono. Se puede definir el convex hull como un conjunto de puntos $X = x_1, x_2, x_3, \dots, x_k$, su envoltura convexa C viene dada por la ecuación 2.15:

$$C(X) = \left\{ \sum_{i=1}^k \alpha_i x_i \mid x_i \in X; \alpha_i \in \mathbf{R}; \alpha_i x_i \geq 0; \sum_{i=1}^k \alpha_i = 1 \right\} \quad (2.15)$$

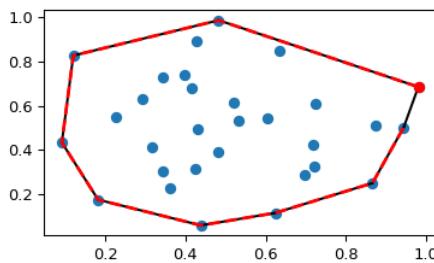


Figura 2.14: Metodo convex hull

2.9.2. Basado en Clustering(K-means)

Clustering es una técnica de análisis exploratorio que consiste en la división de los datos en grupos de objetos con características similares, entre ellos está el algoritmo k-means cuyo objetivo es agrupar k grupos de partición(clúster) de un conjunto de datos, cada clúster contiene un centroide y cada dato debe ser asignado al clúster con el centroide mas próximo, se debe especificar el número de clúster k .

Entonces dado un conjunto de datos de entrada $X = \{x_1, x_2, x_3, \dots, x_n\}$, donde cada observación es un vector de d dimensiones, el algoritmo construye k clúster ($S = \{S_1, S_2, S_3, \dots, S_k\}$) en base a los datos de entrada ($k \leq n$). K-means minimiza la distancia entre los datos y el centroide de cada clúster, cada clúster contiene un subconjunto de los datos de entrada, la función de costo esta dada por la ecuación 2.16.

$$\min_S E(u_i) = \min_S \sum_{i=1}^k \sum_{x_j \in S_i} |x_j - u_i|^2 \quad (2.16)$$

Donde u_i corresponde al centroide de cada clúster, la cual es la media de puntos del cluster S_i . En cada actualización de los centroides imponemos la condición necesaria de extremo a la función $E(u_i)$ que, para la función cuadrática 2.16 es:

$$\frac{\partial E}{\partial u_i} = 0 \Rightarrow u_i^{t+1} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \quad (2.17)$$

Finalmente se toma el promedio de los elementos de cada grupo como nuevo centroide. Las principal ventaja del método k-means es que es un método sencillo y rápido, pero es necesario decidir el valor de k y el resultado final depende de la inicialización de los centroides, ejemplo de k-means en la figura 2.15.

K-means se basa en la suposición de que normalmente los datos pertenecen a grandes y densos clúster, mientras que los valores outlier pertenecen a clúster muy pequeños. Cabe indicar que los grupos producidos por el algoritmo k-means son sensibles al ruido y a los propios valores outlier.

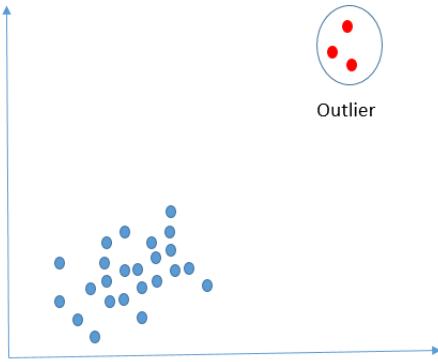


Figura 2.15: Algoritmo k-means

2.9.3. Basado en Clustering(PAM)

PAM (Partitioning Around Medoids) o K-medoids es un algoritmo de agrupamiento que utiliza los objetos más centrales localizados en un clúster (llamados medoide) en lugar de la media del clúster lo cual resulta más eficiente, al igual que K-means intenta minimizar la distancia entre puntos que se añadirían a un cluster y el medoide. PAM es un algoritmo mas robusto en presencia de ruido y valores outlier además, este algoritmo trabaja bien en bases de datos pequeñas pero es lento en grandes. Al igual que K-means, los valores pertenecen a clúster muy pequeños, lo que los diferencia es que K-meloids es más robusto al ruido y valores outlier que K-means, ya que este se basa en una métrica mas robusta de ubicación a datos cercanos como es la medoide.

2.9.4. Basado en densidad (LOF)

Los enfoques basados en la densidad calculan la densidad de regiones en los datos en base k vecinos y declaran los objetos en regiones de baja densidad como valores outlier, el local outlier factor (LOF) usa la estimación de densidad de los objetos localizado en regiones de baja densidad. Entonces un punto presenta una densidad alta si sus vecinos se encuentran muy próximos a él, mientras que diremos que tiene una densidad baja cuando sus vecinos estén muy próximos entre sí, pero no se encuentren cerca de él. LOF trabaja bien en bases de datos pequeñas, pero es lento en base de datos grandes, tal como se puede apreciar en la figura 2.16.

Primero definimos $k - \text{distancia}$ de un objeto o tal que $\text{dist}_k(o)$ es la distancia entre o y su k -th vecino mas cercano.

Luego establecemos $k - \text{distancia-vecindad}$ de o tal que $N_k(o) = \{o' | o' \in D, \text{dist}(o, o') \leq \text{dist}_k(o)\}$

Luego definimos la $\text{reach} - \text{distance}$ o distancia de alcance de o a o' de la siguiente forma: $\text{reach} - \text{distance}_k(o \leftarrow o') = \max\{\text{dist}_k(o), \text{dist}(o, o')\}$. Ahora tenemos definidos estos conceptos para cualquier k , al momento de ejecutar el algoritmo se debe definir k para mantener la condición de densidad. Definimos el *local reach density* o densidad de alcance local de un punto o , según la ecuación 2.18.

$$\text{lrd}_k(o) = \frac{N_k(o)}{\sum_{o' \in N_k(o)} \text{reach} - \text{distance}_k(o \leftarrow o')} \quad (2.18)$$

Finalmente, definimos el local outlier factor de un punto o , siguiendo la ecuación 2.19.

$$\text{LOF}_k(o) = \frac{\sum_{o' \in N_k(o)} \frac{\text{lrd}_k(o')}{\text{lrd}_k(o)}}{N_k(o)} \quad (2.19)$$

Cuanto menor sea el *local reach density* de o , y cuanto mayor sea el *local reach density* del k -th vecino mas cercano de o , mayor será el LOF.

$$\text{LOF} \approx 1 \Rightarrow \text{no outlier}$$

$$\text{LOF} \gg 1 \Rightarrow \text{outlier}$$

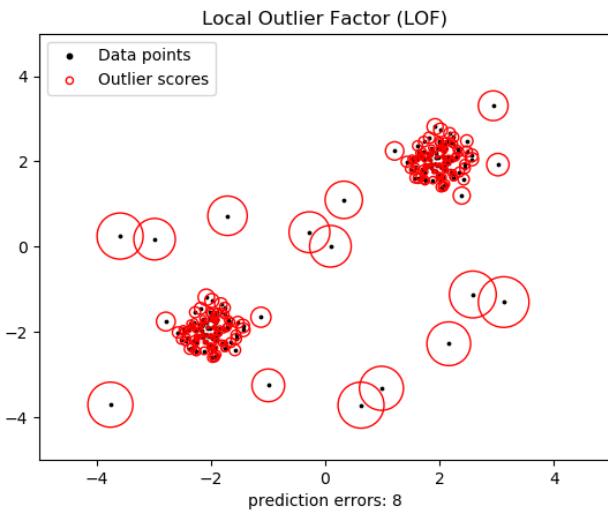


Figura 2.16: Algoritmo local outlier factor

2.10. Comparación de métodos clásicos vs redes neuronales

El objetivo de la detección de datos outlier es separar las observaciones con comportamiento normal de algunas observaciones que tienen un comportamiento anómalo, típicamente en estos problemas se conoce este comportamiento anómalo pero no se tienen identificados aquellos datos que son outlier. Por lo tanto, estamos ante un problema de aprendizaje no supervisado donde solo se tiene el conjunto de datos de entrenamiento.

En los problemas de aprendizaje automático se asume que la estructura de los datos son linealmente no separable esto significa que no existe un línea o hiperplano que pueda separar los datos. Las técnicas mencionadas anteriormente exploran los valores outlier en sus espacios de datos originales y se han demostrado que funcionan bien en distribuciones linealmente separables, sin embargo, tienden a rendir menos que las estructuras no lineales según Lyudchik (2016).

La principal propiedad de las redes neuronales es que estas pueden resolver problemas que no son linealmente separables a comparación de las técnicas mencionadas, esta es la diferencia por la cual las redes neuronales son muy populares. Una de las ventajas de las redes neuronales mencionadas en la sección 2.3.1, es la capacidad de generalizar lo

aprendido a través de una función objetivo, esto permite predecir el valor correspondiente de una entrada valida.

Sin embargo, existen algoritmos que realizan un aprendizaje perezoso(lazy learning) como el local outlier factor(lof), Aha (1997) lo define como un marco en el que los datos de entrenamiento se almacenan simplemente para uso futuro en lugar de utilizarlos para construir una descripción general y explícita de la función objetivo. Esto significa que cada vez que realicemos una predicción sobre datos nuevos, el algoritmo tendrá que entrenar con todos los datos de entrenamiento porque el algoritmo no construye una función objetivo, este proceso es infactible para base de datos grandes,en la tabla 2.1 se compara los métodos mencionados.

	Genera función objetivo	Se aplica a Big Data	Se usa en datos con patrones no lineales
Convex Hull	Si	No	No
K-means	Si	Si	No
k-medoids	Si	Si	No
Local outlier factor	No	No	No
Red neuronales artificiales	Si	Si	Si

Cuadro 2.1: Tabla de comparación de algoritmos

2.11. Inteligencia artificial explicable (XAI)

Inteligencia artificial explicable o Explainable Artificial Intelligence (XAI) es una tendencia que ha tomado fuerza en los últimos años, los algoritmos de inteligencia artificial tienen un alto nivel de capacidad para resolver problemas elevados, pero no tienen la capacidad de explicación. Además, los algoritmos de aprendizaje automático son llamados algoritmos de caja negra porque no se entiende como se generó la salida. Estos algoritmos utilizan matemáticas complejas que son muy difíciles de entender o comprender para los usuarios humanos, esto lleva a que los algoritmos de aprendizaje automático carezcan de transparencia. Bajo este contexto es necesario una explicación e interpretabilidad de los algoritmos de aprendizaje automático según Došilović y cols. (2018).

- **Interpretabilidad** se refiere a la medida en que se puede observar una causa y un efecto dentro de un sistema.
- **Explicación** es la medida en que los mecanismos internos de una máquina o sistema de aprendizaje profundo pueden explicarse en términos humanos.

2.12. Árbol de decisión

El árbol de decisión es un método de aprendizaje supervisado muy útil para la clasificación y regresión, debido a la facilidad de organización y comprensión del conocimiento en forma gráfica y analítica que permite representar todos los eventos que surgen de una decisión asumida en una estructura jerárquica. La decisión final que toma el árbol de decisión se puede determinar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas, esto permite analizar totalmente las posibles consecuencias de tomar una decisión(Hernández Orallo y cols. (2004)). Existen diferentes algoritmos para la construcción de arboles de decisión, el mas popular es el algoritmo de CART(Classification and Regression Trees) que construye árboles binarios utilizando la característica y el umbral que producen la mayor ganancia de información en cada nodo según Hernández Orallo y cols. (2004), en la figura 2.17 se muestra un ejemplo de árbol de decisión aplicado a un conjunto de datos de plantas de Iris.

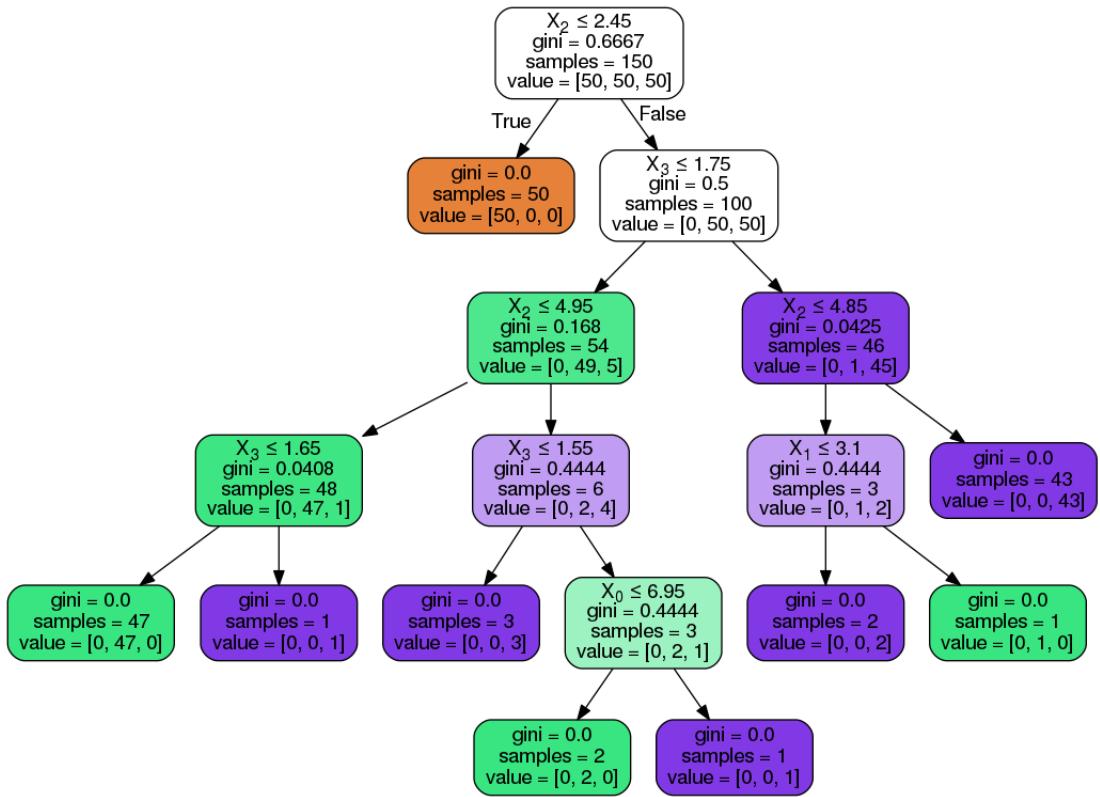


Figura 2.17: Árbol de decisión extraído de Pedregosa y cols. (2011)

El árbol de decisión utiliza un modelo de caja blanca, esto quiere decir que la salida de una evento se explica fácilmente mediante la lógica booleana, esto es de gran ayuda para poder interpretar y explicar modelos de caja negra como son las redes neuronales, donde los resultados de estos modelos son difíciles de interpretar. En nuestro caso vamos a usar este modelo para dar explicación e interpretabilidad a las redes neuronales validadas de este proyecto de forma que puedan darnos una explicación en términos de variables de porque una determinada serie de tiempo es outlier o normal.

Parte III

Capítulo 3

Capítulo 3

Materiales y métodos

3.1. Modelo de desarrollo de software y machine learning

Waypoint cuenta con el módulo historial de rutas donde sus clientes pueden visualizar el monitoreo histórico de sus vehículos, esta es la funcionalidad principal de este módulo. El objetivo es integrar una API Rest que pueda detectar puntos outlier en los trayectos de vehículos por ende se desarrollara un prototipo del módulo historial de ruta, de esta forma no se desarrolla todo el sistema, pero si condensara la idea principal. Para ello ya se tienen los requerimientos funcionales definidos y las cuales no cambiaran a lo largo del proyecto. Para este proyecto se utilizara el modelo cascada ya que es fácil de implementar para proyectos pequeños. Además de este, se propuso el modelo CRISP-DM(Cross Industry Standard Process for Data Mining) para el desarrollo del modelo de machine learning, la particularidad que tiene este modelo es que comparte procesos análogos con el modelo cascada, esto nos permite tener un control paralelo de los 2 modelos tanto en planificación y desarrollo.

3.1.1. Modelo cascada

Pressman y Troya (1988) define el modelo cascada como un enfoque sistemático y secuencial para el desarrollo del software, que comienza con la especificación de los requerimientos por parte del cliente y avanza a través de planeación, modelado, construcción y despliegue, para concluir con el apoyo del software terminado, ver figura 3.1.

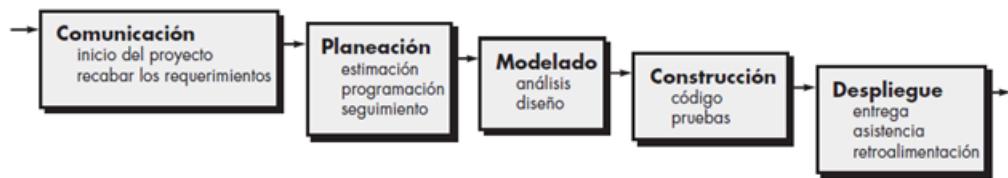


Figura 3.1: Modelo cascada

1. **Comunicación:** En esta fase se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir.
2. **Planeación:** En esta fase se planifica el proyecto, el tiempo que tome cumplir con los objetivos, programar las actividades para el proyecto y el seguimiento de proyecto.
3. **Modelado:** En esta fase se define la estructura interna del software, también se determina el diseño de alto nivel o arquitectónico y diseño detallado.
4. **Construcción:** En esta fase se programan los requisitos especificados haciendo uso de las estructuras de datos diseñadas en la fase anterior y seguida por las pruebas de funcionamiento al software .
5. **Despliegue:** Una vez terminado el desarrollo del software y ser aceptado por el cliente, se procede a implementar.

3.1.2. Modelo CRISP-DM(Cross Industry Standard Process for Data Mining)

Para poder aplicar técnicas de machine learning es fundamental contar con grandes cantidades de información y que están sean canalizadas por los algoritmos de machine learning, por ende es necesario un preprocesamiento de información que viene desde la comprensión del negocio hasta el despliegue de la aplicación.

Una metodología que cubre todos estos procesos es CRISP-DM(Cross Industry Standard Process for Data Mining) figura 3.2, que normaliza los procesos de minería de datos, de forma similar se hace en la ingeniería de software. A continuación, se describen cada una de las fases en que se divide CRIPS-DM Gallardo Arancibia (2009).

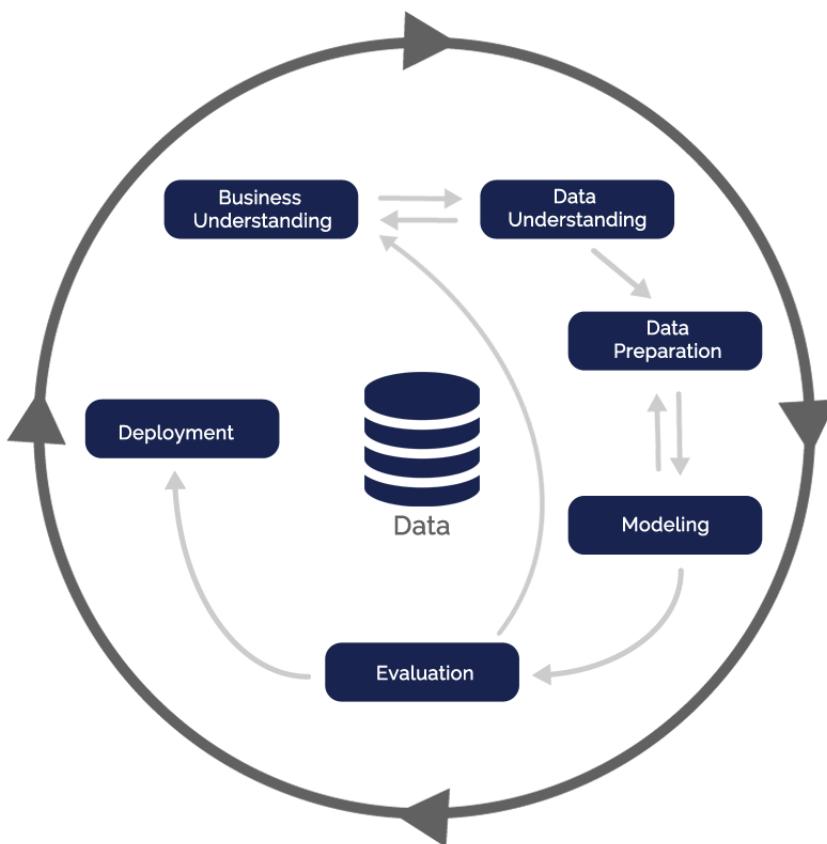


Figura 3.2: Modelo CRISP-DM

1. **Comprensión del negocio:** En esta fase se enfoca en la comprensión de los objetivos y requisitos del proyecto, también se establece las herramientas y el plan

de proyecto. Además, en esta fase es muy importante entender el problema que se desea resolver porque esto permitirá recolectar los datos correctamente.

2. **Comprensión de los datos:** En esta fase empieza con la recolección de datos con el objetivo de familiarizarse con ellos, describir, entender cada dato y verificar la calidad de los estos, todo esto para poder establecer relaciones mas evidentes entre los datos y definir las primeras hipótesis.
3. **Preparación de los datos:** Una vez terminado la recolección de los datos se procede a construir el conjunto de datos para adaptarlos a las técnicas modelado, en esta fase se utiliza la limpieza de datos que es optimizar la calidad de los datos para prepáralos en la fase de modelado.
4. **Modelado:** En esta fase se seleccionan las técnicas de modelado mas apropiadas para el tipo de problema a resolver, se tiene que considerar el objetivo principal del proyecto ya que nos indicara el tipo de problema.
5. **Evaluación :** En esta fase se ha construido uno o mas modelos que deben ser evaluados con factores como la exactitud y generalidad del modelo. Además, se debe comparar el modelo obtenido con los objetivos del proyecto.
6. **Implantación:** Una vez que se obtiene le mejor modelo que satisface el objetivo general, se tiene que presentar los resultados de manera comprensible de tal forma que el usuario pueda usarlo y se tiene que documentar todo el proceso y resultados.

3.2. Acta de constitución del proyecto

El Acta de Constitución del Proyecto es un documento en el que se define el alcance, los objetivos y los participantes del proyecto. Da una visión preliminar de los roles y responsabilidades, de los objetivos, de los principales interesados.

ACTA DE CONSTITUCIÓN DEL PROYECTO	
Proyecto:	Detección de series de tiempo anómalas en trayectos de vehículos usando deep learning
Patrocinador:	Waypoint Telecomunicaciones
Aprobado por:	Billy Peralta
Descripción del producto a entregar	
El proyecto busca entregar un modelo de aprendizaje automático usando deep learning, que detecte comportamientos anómalos en los trayectos de vehículos e implementarlo en el módulo historial de rutas a través de una interfaz de programación(API).	
Alineamientos del proyecto	
Objetivo del proyecto	Objetivos específicos del proyecto
Automatizar la detección de trayectos anómalos en vehículos basado en el historial de rutas de la flota de vehículos de Waypoint.	<ul style="list-style-type: none"> - Entender las variables que puedan ser usadas como predictoras para identificar trayectos anómalos. - Desarrollar e implementar modelos de red neuronal que detecte trayectos de vehículos anómalos exitosamente, basado en el historial de rutas de vehículos de Waypoint.
Propósito del proyecto	
Con este modelo buscamos la automatización de detección de trayectos anómalos, las cuales son lecturas erróneas del dispositivo GPS producidas por interferencia de señal o zonas sin cobertura digital, que ocasionan sanciones y amonestaciones a los conductores de los vehículos que son auditados por las empresas.	
Factores críticos de éxito del proyecto	
<ol style="list-style-type: none"> 1.- Adquisición de los datos por parte de la empresa Waypoint. 2.- Validar el modelo de aprendizaje automático para comprobar que los resultados son válidos y suficientemente satisfactorios. 	
Requerimientos de alto nivel	
<ol style="list-style-type: none"> 1.- El API desarrollado debe integrarse con el modulo historial de rutas. 2.- El historial de rutas debe indicar que trayectos son anómalos junto con sus valores correspondientes. 	
Interesados clave	
<ol style="list-style-type: none"> 1.- Gerencia de desarrollo e investigación de la empresa Waypoint. 2.- Equipo de proyecto. 	
Riesgos	
<ol style="list-style-type: none"> 1.- Demora en el proceso de entrega de datos. 2.- Retraso en el desarrollo del modelo de aprendizaje automático. 	
Hitos principales del proyecto	
<ol style="list-style-type: none"> 1.- Aprobación del acta de constitución por parte del interesado. 2.- Entrega de los datos para el entrenamiento del modelo de aprendizaje automático. 3.- Entrega de un prototipo funcional del modulo historial de rutas y con una funcionalidad que detecte trayectos de vehículos anómalos. 	
Hitos principales del proyecto	
La persona asignada para el proyecto es el estudiante Richard Soria Mora.	

3.3. Planificación de proyecto

Para el desarrollo de la planificación del proyecto se ha utilizado la guía Pmbok que es un instrumento desarrollado por el Project Management Institute(Rose (2013)), que establece un criterio de buenas prácticas relacionadas con la gestión, la administración y la dirección de proyectos, los planes abarcados para este proyecto:

- Plan de alcance
- Plan de tiempo
- Plan de costos
- Plan de calidad
- Plan de riesgos

3.3.1. Gestión de alcance

La gestión de alcance define y controla lo que se incluye y no en el proyecto, además se describen las fases que se llevaron a lo largo del proyecto, en esta etapa se deja en claro cuáles serán los entregables del proyecto, ver tabla 3.1.

Estructura de desglose de trabajo (EDT)

La estructura de desglose de trabajo es un diagrama que divide el trabajo del proyecto en componentes más pequeños y fáciles de manejar, estos componentes representan un trabajo específico, a continuación, se muestre el EDT del proyecto.

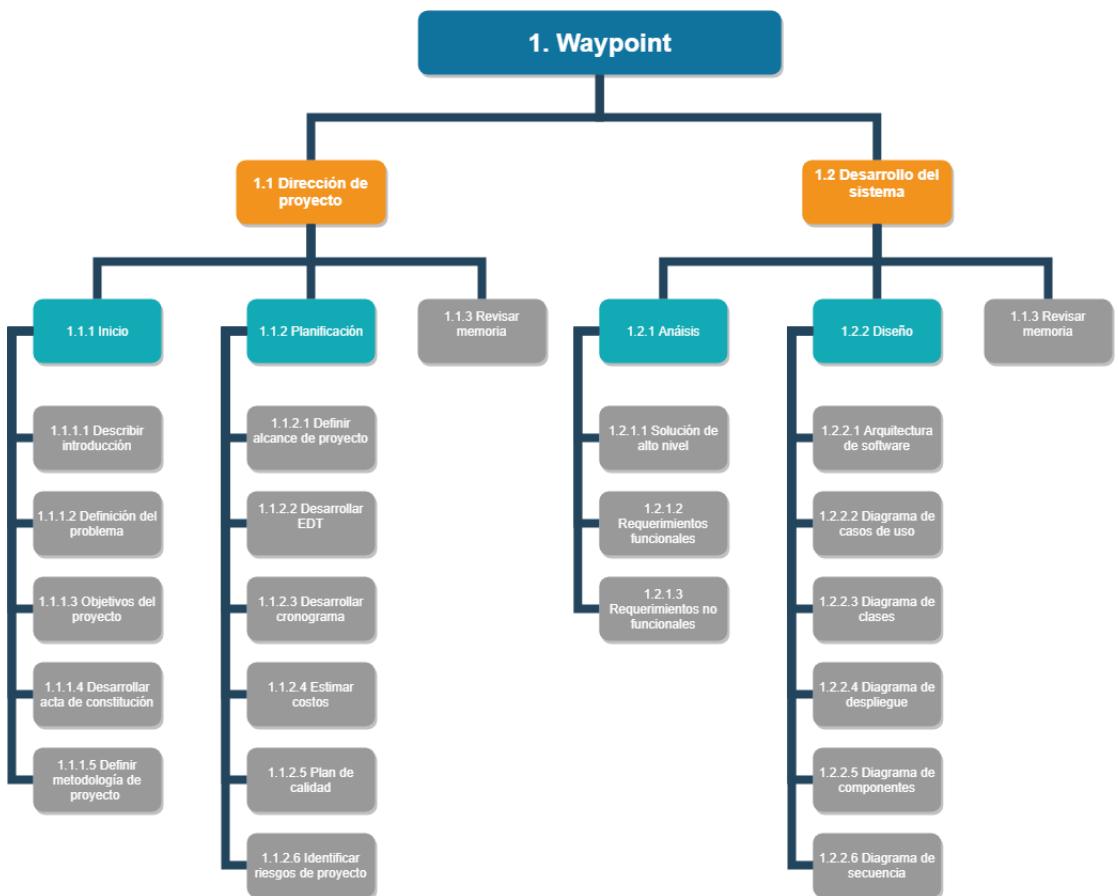


Figura 3.3: EDT parte 1

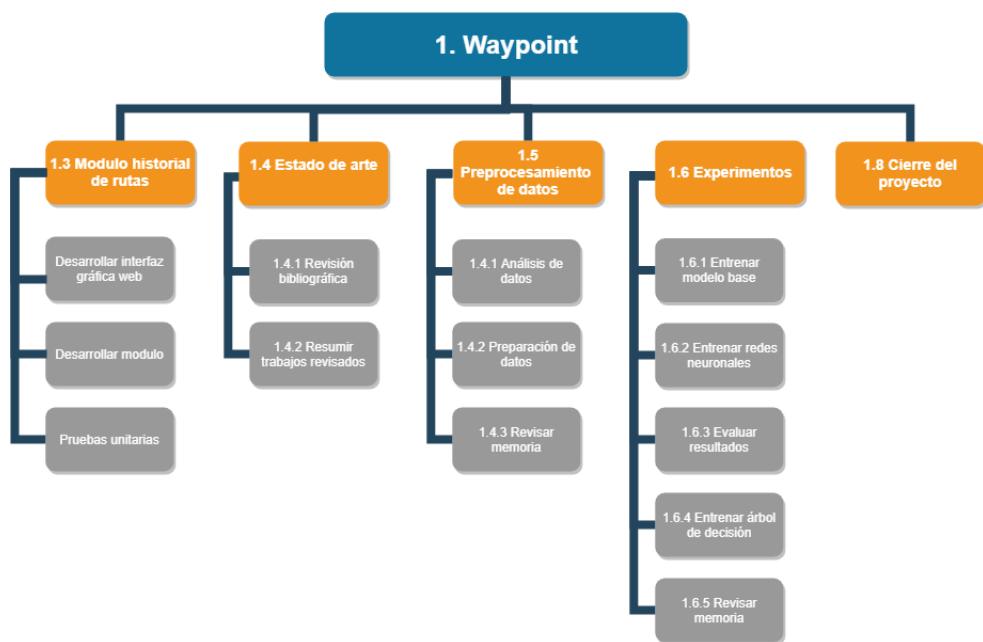


Figura 3.4: EDT parte 2

Fase	Objetivo	Entregable
Planificación	Programar y estimar actividades necesarias para alcanzar los objetivos.	- Documento de gestión de proyecto.
Análisis de software	Establecer las funcionalidades del software.	- Requerimientos funcionales. - Requerimientos no funcionales.
Diseño de software	Definir la organización del software.	- Casos de uso. - Diagrama de clases. - Diagrama de despliegue. - Diagrama de componentes. - Diagrama de secuencia.
Desarrollo de software	Desarrollar el módulo historial de ruta.	- Modulo historial de rutas
Entrenar modelo	Explorar la información y entrenar modelos machine learning.	- Documento con el resultado de los experimentos.

Cuadro 3.1: Tabla de gestión de alcance

3.3.2. Plan de tiempo

La gestión de tiempo administra el tiempo de las actividades que se definieron en el EDT, busca conocer cuanto durarán cada actividad con el fin de tener un plazo de tiempo determinado. Para establecer los tiempos de cada actividad en el plan de tiempo, se homologó las actividades en base al proyecto de Rodrigo (2018), a continuación, se muestra le cronograma del proyecto.

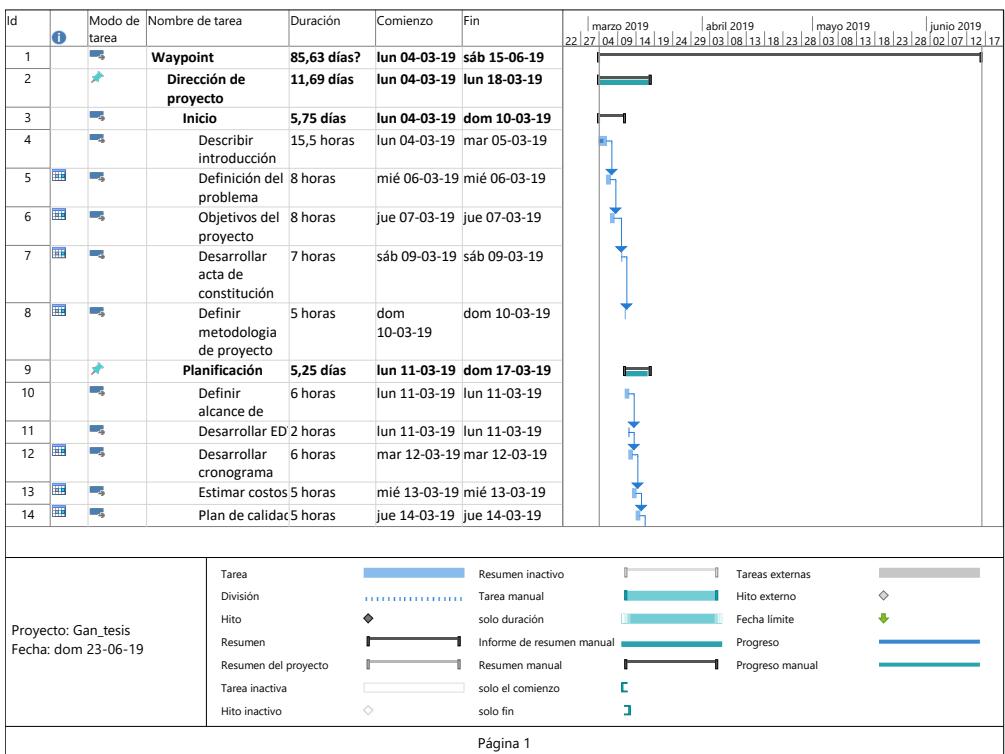


Figura 3.5: Cronograma del proyecto, pagina 1

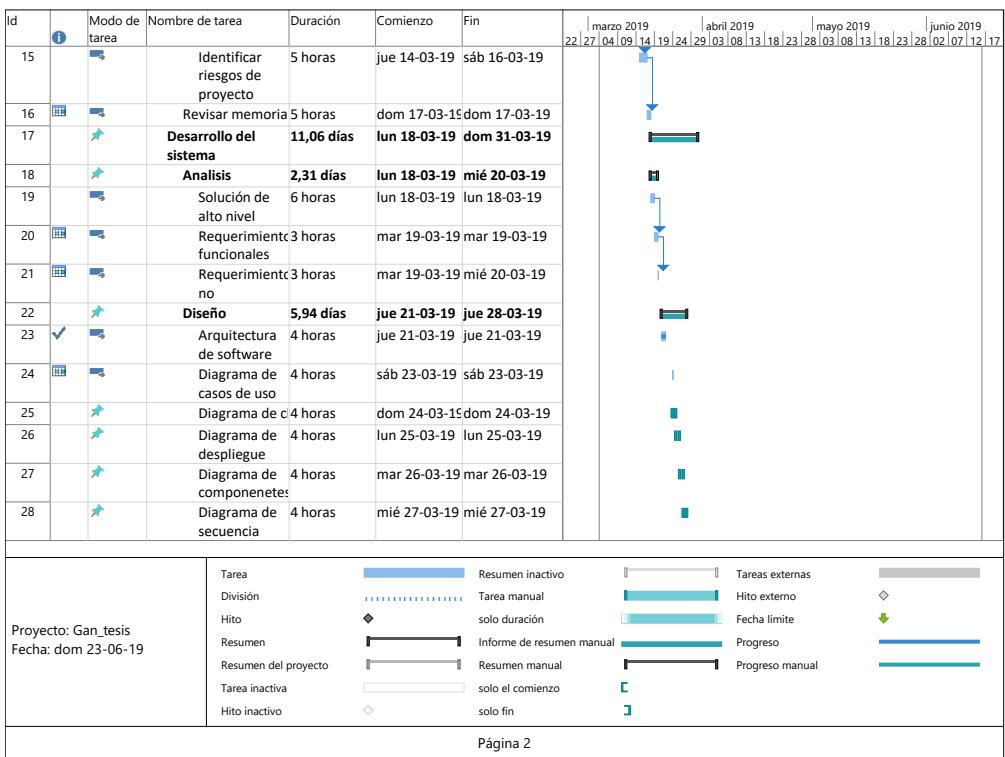


Figura 3.6: Cronograma del proyecto, pagina 2

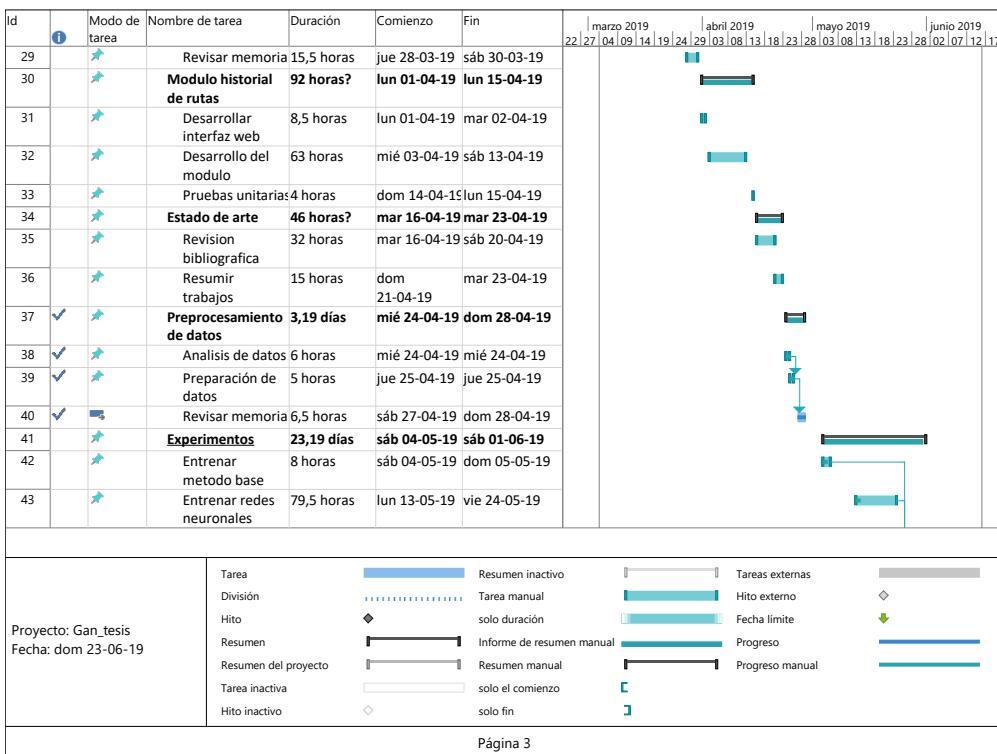


Figura 3.7: Cronograma del proyecto, pagina 3

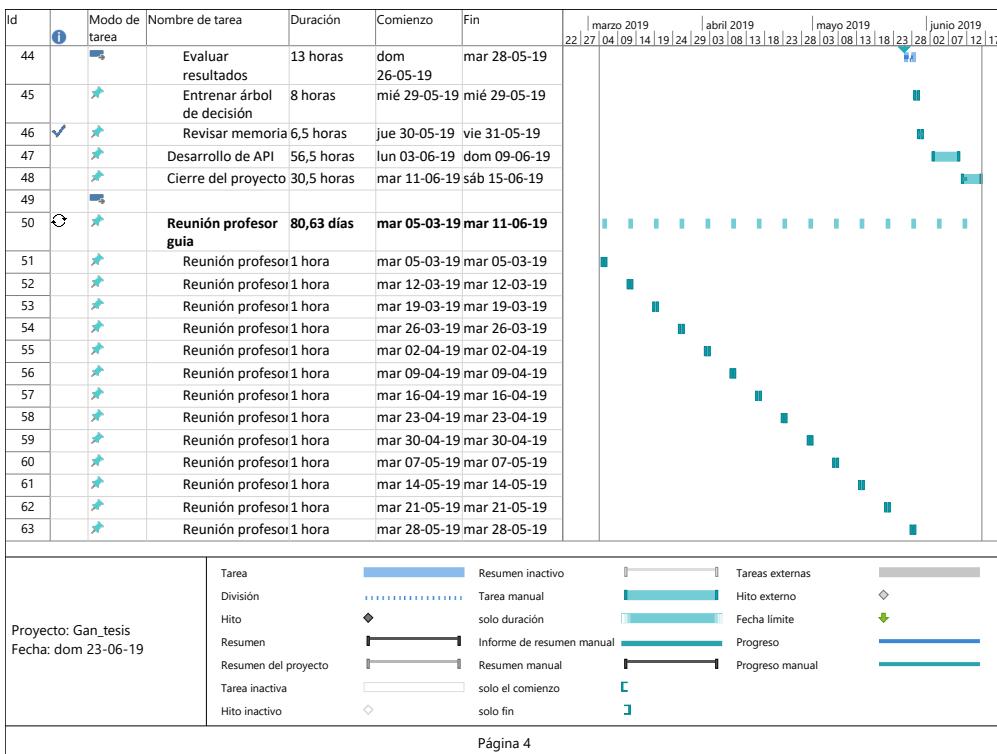


Figura 3.8: Cronograma del proyecto, pagina 4

3.3.3. Plan de costos

La gestión de costos incluye los procesos de estimar, presupuestar y controlar los costos que se definen para el proyecto, además de como fluirá en efectivo durante el periodo del proyecto.

Los costos asociados a los recursos humanos están basados en el cronograma de la carta Gantt como se muestra en la tabla 3.2. A continuación, se muestra los costos por

Duración horas	486 h
Tarifa por hora	\$ 7000
Valor estimado	\$ 3.402.000

Cuadro 3.2: Tabla de costos

los activos fijos del proyecto en la tabla 3.3.

Items	Estimación
Notebook ASUS FX504 SERIES	\$ 870.000
Microsoft Project 2016	\$ 129.990
Servicio de Internet Movistar	\$ 168.000
Total	\$ 1.167.990

Cuadro 3.3: Tabla de activos

En la tabla 3.4 se presentan el total de costos del proyecto.

RR.HH	\$ 3.402.000
Activos fijos	\$ 1.167.990
Total	\$ 4.569.990

Cuadro 3.4: Tabla de costos total

3.3.4. Plan de calidad

La gestión de calidad es la etapa donde se determinan políticas de calidad, con el objetivo de que el proyecto satisfaga las necesidades del proyecto. Para este proyecto se utilizaron los siguientes puntos bases para la calidad:

- Utilizar las buenas prácticas propuestas por el PMBOOK.
- Cumplir con los plazos establecidos.

- Seguir una metodología para el desarrollo del modelo de machine learning.

3.3.5. Plan de riesgos

El propósito de la gestión de riesgos es identificar aquellos riesgos que pudieran perjudicar al proyecto tanto en tiempo como costos y cuantificarlo de acuerdo a su grado de probabilidad de ocurrencia, para disminuir este grado de ocurrencia se debe realizar un plan de mitigación y contingencia, tabla 3.5.

Riesgo	Evento disparador	Mitigación	Contingencia	Prioridad
Entrega tardía de los datos de prueba por parte de la empresa Waypoint.	La empresa no envía los datos de prueba con respecto al tiempo acordado.	Proponer una fecha límite para la entrega de datos de prueba.	- Comunicarse con la empresa y acelerar el proceso de entrega de datos. - Buscar base de datos similares a los datos de prueba.	Alto
Entrenamiento de los modelos de machine learning son detenidos por motivos de mantenimiento o falla del servidor.	El servidor es reiniciado o apagado por motivos de mantenimiento.	Guardar el modelo entrenado cada 1 hora para luego volver a entrenar desde el punto en que se dejó.	Comunicarse con el encargado del servidor para que dé solución al problema sucedido.	Alto
Incorrecta configuración del modelo de machine learning.	Se realiza un monitoreo del modelo entrenándose y este no tiene el comportamiento esperado.	Revisar con detalle el script del modelo de machine learning antes de ser ejecutado.	Detener la ejecución del modelo y revisar la configuración del modelo para volver a ejecutarlo.	Medio

Cuadro 3.5: Tabla de riesgos

3.4. Materiales

3.4.1. Hardware

Para el desarrollo del proyecto se usaron 2 tipos de equipos, el primero es un notebook personal que se uso para el análisis de las variables candidatas, también se uso para el desarrollo del código fuente de los scripts que entrena los modelos de aprendizaje automático. Además se uso el notebook para el desarrollo del modulo historial de rutas, en la tabla 3.6 se especifica las características del notebook.

Característica	Descripción
Marca	Asus
Modelo	TUF Gaming FX504GD-E4653T
Procesador	Intel Core i7-8750H
Memoria RAM	DRAM DDR4 2666 8G
Almacenamiento	SATA 1TB HDD
Tarjeta Gráfica	NVIDIA GeForce GTX 1050, 4GB

Cuadro 3.6: Tabla de descripción notebook

Para el segundo equipo se uso un servidor proporcionado por la universidad para la experimentación de los modelos de aprendizaje automático, este servidor cuenta con muchos recursos para realizar cálculos con bases de datos grandes además de que el acceso a este es de forma remota, en la tabla 3.7 se especifica las características del servidor.

Característica	Descripción
Sistema Operativo	Ubuntu 16.04.2 LTS
Procesador	Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz
Memoria RAM	125GiB
Tarjeta Gráfica	Tesla P4

Cuadro 3.7: Tabla de descripción servidor

3.4.2. Software

Se uso como lenguaje de programación principal Python3.7 tanto para desarrollo del modelo de aprendizaje automático y el modulo historial de rutas, además de framework de trabajo TensorFlow y Django, en la tabla 3.8 se describen las librerías usadas.

Librería	Descripción
TensorFlow 1.12	TensorFlow es una biblioteca de código abierto para aprendizaje automático a través de un rango de tareas.
Numpy 1.16.1	NumPy es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.
Matplotlib 3.0.3	Matplotlib es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python.
Pandas 0.24.2	En Computación y Ciencia de datos, pandas es una biblioteca de software escrita como extensión de NumPy para manipulación y análisis de datos para el lenguaje de programación Python.
Django 2.2	Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo–vista–template.
Jupyter Notebook 5.7.8	Aplicación web que permite desarrollar código interactivo en python.

Cuadro 3.8: Tabla de descripción librerías

3.5. Comprensión de los datos

3.5.1. Recolección de datos

La empresa Waypoint Telecomunicaciones concedió los datos de prueba para el desarrollo de este proyecto, se recolectó datos de diferentes empresas que estaban asociadas a Waypoint. El módulo histórico de rutas guarda indicando la fecha y hora en que se envió los datos del GPS a la base de datos de Waypoint, para nuestro caso de estudio se tomó el histórico de rutas entre el año 2016 y 2018, los datos fueron exportados en archivos .csv.

3.5.2. Descripción de datos

De igual forma que la recolección de datos, la empresa Waypoint Telecomunicaciones concedió la descripción de las variables que usa el dispositivo GPS, en la tabla 3.9 se muestra la descripción de cada variable.

Dato	Tipo de dato	Descripción
fecha	long	Fecha y hora de transmisión GPS en milisegundos, con zona horaria UTC.
latitud	double	Coordenadas decimales para la latitud GPS.
longitud	double	Coordenadas decimales para la longitud GPS.
altitud	double	Altura en el espacio sobre la superficie del nivel del mar en metros.
velocidad	double	Velocidad calculada por equipo GPS en Km/hr.
cog	double	Ángulo de orientación sobre la superficie terrestre medido en grados de 0 a 360.
nsat	double	Cantidad de satélites a la vista que tiene el equipo GPS en el momento. Ayuda a determinar la calidad de la señal GPS. Mientras más satélites, mejor precisión.
realtime	boolean	Tipo de registro, tiempo real igual a true, backup igual a false.
input	int[]	Arreglo de status de entradas digitales. Existen 4 entradas, ejemplo: [input1, input2, input3, input4] donde el valor puede ser 1 (activa) o 0 (desactiva).
hdop	double	Determina la geometría de los satélites en el cielo, ayuda a definir la calidad de la señal GPS. Mientras más cercano a 1, mejor calidad de señal y precisión.
ignición	int	Determina el encendido (1) o apagado del motor (0).
adc	double[]	Arreglo de señales análogas, se usa mayormente para indicar los valores de temperaturas. Existen 4, ejemplo [adc1, adc2, adc3, adc4]. El valor -200 es que no posee habilitada esa señal.

continua en la siguiente pagina

Dato	Tipo de dato	Descripción
power	int	Determina si el equipo está funcionando con la batería del vehículo (1) o con batería interna (0).
horometro	long	Cantidad de segundos funcionando con el motor encendido.
odometro	long	Cantidad de metros recorridos, calculado por el equipo mediante el uso del GPS.
panico	int	Señal que detecta si el botón de pánico fue presionado (1). Esta variable es usada en particular por algunos modelos. Para los otros, se usan las inputs estándar de pánico.
bateria	double	Valor del voltaje que tiene la batería del móvil donde está instalado el equipo GPS. Sólo algunos equipos miden y entregan esta variable por defecto.
bateriaint	double	Valor del voltaje de la batería interna del equipo GPS. Sólo algunos equipos miden y entregan esta variable por defecto.
patente	string	Patente del vehículo.
descripVehículo	string	Información adicional del vehículo.

Cuadro 3.9: Tabla de descripción de variables

3.6. Preparación de datos

3.6.1. Selección de variables

En esta etapa se seleccionan las variables que se usarán para entrenar el modelo de aprendizaje automático, en la descripción del problema 1.2 la empresa Waypoint identificó dos tipos de outlier, el primero lo llamaremos outlier espacial donde el algunos

puntos geográficos están alejados de aquellos puntos que tienen un comportamiento normal y el segundo lo llamaremos outlier de velocidad donde las lecturas de la velocidad de los vehículos son incorrectos produciendo sanciones a los conductores. Teniendo estas 2 tipos de outlier identificados por parte de la empresa, se hizo una retroalimentación de expertos humanos para la selección de variables, quedando como candidatas 6 variables(Fecha,Latitud,Longitud,Altitud,Velocidad,Numero de satélites), para la detección de outlier espacial solo se hará uso de las variables fecha, latitud, longitud y para la detección de outlier de velocidad se hará uso de todos todas las variables.

3.6.2. Limpieza de datos

En esta etapa se corrige o elimina registros de datos erróneos de los datos de entrenamiento como también la normalización de los datos para estandarizar el rango independiente de las variables, los datos proporcionados por la empresa están completos ni presentan datos erróneos. En cuanto a la normalización de datos se uso el método min-max normalization, que se aplica a cada variable de los datos y viene dada por la ecuación 3.1, donde x en variable, $\min(x)$ es el valor mínimo de la variable y $\max(x)$ es el valor máximo de la variable .

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3.1)$$

3.6.3. Exploración de los datos

El objetivo de esta sección es encontrar una estructura general de los datos a través de pruebas estadísticas básicas, que revelen propiedades en los datos recién adquiridos. Para ello analizamos el tipo de variable y una descripción univariable, la cual nos ayudara a comprender mejor los datos, en la tabla 3.10 se muestra el análisis univariable.

Análisis univariable de datos

La variable latitud muestra un valor máximo de 22.4405, la cual no coincide con la localización geográfica de los vehículos que transitan en Chile y la cual es un posible error de lectura del dispositivo GPS, de igual forma pasa con la variable longitud y su valor máximo de 0 que representa la localización de África. En cuanto a la variable altitud esta

muestra un valor máximo de 3.574640e+14, la cual indica una lectura errónea del GPS y es también un valor outlier si se compara con la mediana de esta.

La variable velocidad muestra un valor negativo -9, la cual es un error de lectura ya que la velocidad es un valor positivo, mientras que el número de satélites tiene un valor máximo de 315, la cual indica una lectura errónea del GPS y es también un valor outlier si se compara con la mediana de esta.

Nombre variable	Tipo variable	Valor mínimo	Valor máximo	Media	Mediana
Latitud	Cualitativa continua	-53.67396	22.4405	-33.40148	-33.588202
Longitud	Cualitativa continua	-75.75141	0	-71.35929	-71.182729
Altitud	Cualitativa continua	-1044	3574.640e+11	1104.757e+4	207
Velocidad	Cualitativa continua	-9	201	11.01984	0
Número de satélites	Cualitativa discreta	0	315	9.875347	9

Cuadro 3.10: Tabla de descripción cuantitativa de variables

El comportamiento de cada variable a través de histogramas se puede ver en el apéndice A.

3.7. Transformación de datos a series de tiempo

Los datos extraídos de la base de datos de Waypoint son registros de vehículos ordenados temporalmente, cada registro es una observación en un tiempo t , para los dos modelos propuestos se realiza diferentes transformaciones, los datos se deben transformar en series de tiempo.

La red autoencoder apilada recibe como entrada series de tiempo multidimensional, una serie de tiempo T es una secuencia de observaciones $T = \{S_1, S_2, S_3, \dots, S_k\}$, donde $S_k = \{s_i^1, s_i^2, s_i^3, \dots, s_i^d\}$ es una observación multidimensional $S_k \in R^d$, se considero el tamaño de la ventana k que tomare con conjunto de observaciones y b son los pasos que se desliza la ventana, aplicando la ventana deslizante sobre los datos se obtiene un

conjunto de series de tiempo, en la figura 3.9 se ilustra el procedimiento.

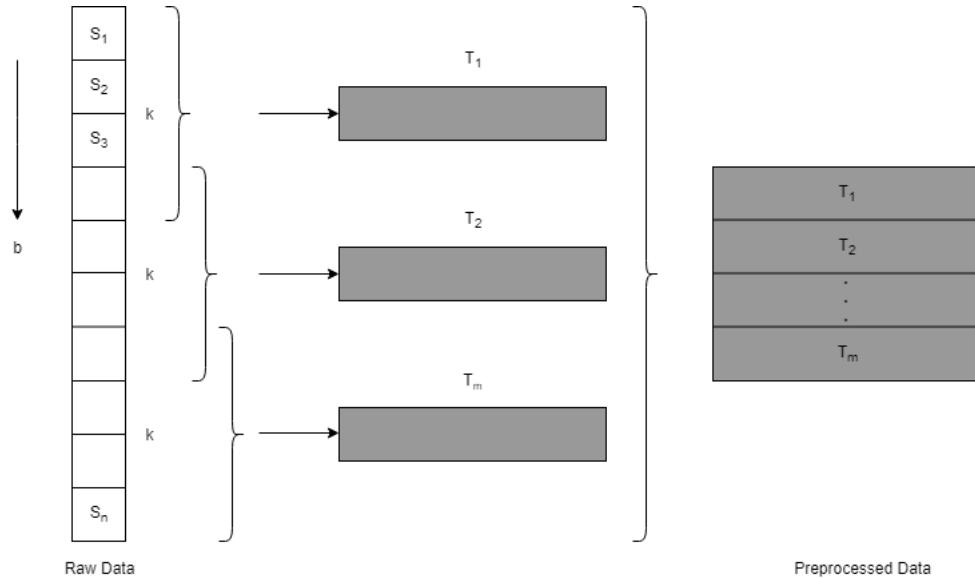


Figura 3.9: Transformación de datos a 1-D

La red autoencoder convolucional también recibe como entrada series de tiempo pero en una cuadricula de dos dimensiones, de igual forma que la red autoencoder utilizamos la ventana deslizante, solo que ahora se transforma la serie de tiempo T en una cuadricula de 2 dimensiones, en la figura 3.10 se ilustra el procedimiento.

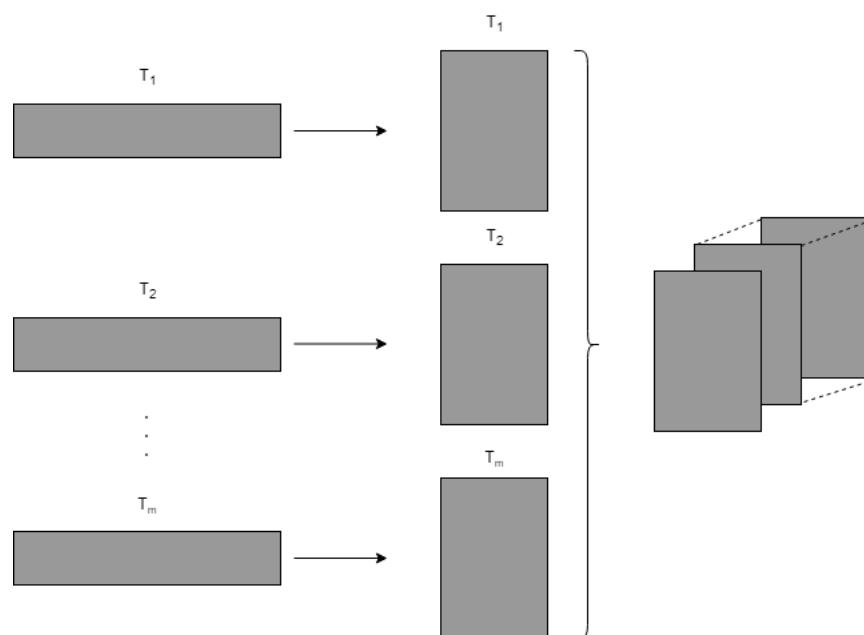


Figura 3.10: Transformación de datos a 2-D

3.8. Criterio para la detección de series de tiempo outlier

Para la detección de series de tiempo outlier usamos el error de reconstrucción, el modelo ajusta sus parámetros para minimizar la función costo de modo que sea más probable que la red pueda reconstruir los patrones mas comunes y para los patrones anómalos estén menos reproducidos por el modelo, podemos derivar el error de reconstrucción(ER) en la ecuación 3.2..

$$\text{Error reconstrucción}(ER) = |\hat{x} - f(g(x))|^2 \quad (3.2)$$

Donde x es la serie de tiempo, \hat{x} es la serie de tiempo reconstruida por el modelo y $f(g(x))$ es el modelo.

Por lo tanto, las series de tiempo que representan patrones con baja reconstrucción son considerados series de tiempo normales, mientras que aquellas series de tiempo que representen patrones con alto error de reconstrucción son considerados series de tiempo outlier.

Para decidir si una serie de tiempo es outlier o no, se establece un umbral de reconstrucción τ , de modo que si el error de reconstrucción supera el umbral es considerado outlier y si el error de reconstrucción en menor al umbral es considerado normal, se define el criterio de detección de outlier en la ecuación 3.3.

$$\text{Outlier} = \begin{cases} 1 & \text{si } ER \geq \tau \\ -1 & \text{si } ER < \tau \end{cases} \quad (3.3)$$

Complementariamente se puede establecer un criterio para la detección de aquellas observaciones de la serie de tiempo que tengan una reconstrucción alta, se tomará el mismo criterio que se uso en detección de series de tiempo outlier, es decir usaremos el error de reconstrucción *pero aplicado a las observaciones de la series tiempo outlier*. Para definir si una observación es outlier o no, se tomará el percentil de 90 % de los datos debido a que los rangos de errores de reconstrucción de las observaciones varía para cada serie de tiempo y por la tanto no se puede fijar en un umbral constante. Específicamente el error de reconstrucción de cada observación(ER_o) es calculada por la ecuación 3.2 y se

toma como umbral u el percentil de 90 % de los datos dado por la ecuación 3.4 , donde n es el numero de observaciones de la serie de tiempo e i el percentil.

$$u = n \frac{i}{100} \quad (3.4)$$

Para decidir si es una observación es outlier o no, usamos el umbral u , de modo que si el error de reconstrucción supera el umbral es considerado outlier y si el error de reconstrucción en menor al umbral es considerado normal, se define el criterio de detección de observaciones outlier en la ecuación 3.5.

$$\text{Outlier Observación} = \begin{cases} 1 & \text{si } \text{ER}_o \geq u \\ 0 & \text{si } \text{ER}_o < u \end{cases} \quad (3.5)$$

3.9. Evaluación del modelo

3.9.1. Evaluación cuantitativa

Para poder evaluar el modelo correctamente, se separó una muestra de datos del total de datos, con esta muestra de datos se entreno los modelos de aprendizaje automático. Se uso el método Hold-Out para entrenar y evaluar el modelo, que consiste en separar de manera aleatoria una parte de los datos de los que se dispone, obteniendo:

- Conjunto de entrenamiento (70 %)
- Conjunto de evaluación (30 %)

Con el conjunto de entrenamiento el modelo aprende y con el conjunto evaluación el modelo evalúa el rendimiento, una representación del aprendizaje y rendimiento del modelo es a través una curva de aprendizaje y curva de validación respectivamente.

Métrica de aprendizaje

La métrica de aprendizaje es el error cuadrático medio la cual se utiliza en un procedimiento iterativo, mostrando la convergencia de la función costo respecto al número de iteraciones. El error cuadrático medio(MSE) dada en la ecuación 3.6, nos da un promedio

del error entre la diferencia de los datos originales y datos reconstruidos, donde x es la serie de tiempo, \hat{x} es la serie de tiempo reconstruida por el modelo y n es el número de observaciones.

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{x}_i - x_i)^2 \quad (3.6)$$

Métrica de validación

La métrica de validación mide que tan efectivo es el modelo en un procedimiento iterativo, mostrando el rendimiento del modelo respecto al número de iteraciones. Las métricas de validación utilizadas en este proyecto son el coeficiente de determinación (R^2) y la varianza explicada (VE).

Coeficiente de determinación(R^2) Este coeficiente mide la bondad del ajuste realizado por el modelo y permite decidir si el ajuste lineal es suficiente, el coeficiente de determinación se define en la ecuación 3.7, donde x es el valor de entrada, \hat{x} es el valor pronosticado por el modelo, \bar{x} es el promedio de los valores de x y n es el número de observaciones.

$$R^2 = \frac{scE}{scG} = \frac{\sum_{i=1}^n (\hat{x}_i - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (3.7)$$

El coeficiente de determinación es igual a la proporción de la varianza explicada(scE) respecto de la varianza total(scG) se cumple que $scE < scG$ y se verifica que $0 < R^2 < 1$, por lo tanto cuanto más cerca de 1 se sitúe R^2 , mayor será el ajuste del modelo. De forma inversa, cuanto más cerca de cero, menos ajustado estará el modelo y será menos fiable.

Varianza explicada (VE) Varianza explicada mide la proporción a la que un modelo explica la variación (dispersión) de un conjunto de datos determinado, la variación explicada se define en la ecuación 3.8.

$$VE = 1 - \frac{S_e^2}{S_x^2} \quad (3.8)$$

Donde S_e^2 es la varianza del error residual que presentan los valores de x comparados con sus pronósticos \hat{x} , representamos el error residual como $e = x_i - \bar{x}$ y n es el número de

observaciones.

$$S_e^2 = \frac{\sum_{i=1}^n (e_i - \bar{e})^2}{n} \quad (3.9)$$

Y S_x^2 es la varianza total de x y \bar{x} es el promedio de los valores de x .

$$Var(x) = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n} \quad (3.10)$$

3.10. Evaluación cualitativa

Como se menciono anteriormente el criterio para establecer si una serie de tiempo es outlier o no se basa en el error de reconstrucción, por lo tanto aquellas series de tiempo con mayor error de reconstrucción son considerados como outlier y aquellas series de tiempo con menor reconstrucción son considerados como normales, para la evaluación cualitativa se tomarán las últimas 100 series de tiempo con mayor error de reconstrucción. Se ilustrará cada serie de tiempo en un mapa web y se georreferencia cada observación en el mapa indicando si es outlier o no, junto con sus valores correspondiente de cada observación. Posteriormente un experto en la materia evaluará cada serie de tiempo, y por consiguiente dará una valoración de si la serie de tiempo es outlier, dudoso o no es outlier, en la tabla 3.11 se muestra la valoración.

Valoración	Puntuación
Outlier	1
Dudoso	0.5
No outlier	0

Cuadro 3.11: Tabla de valoración, fuente:Autor

Parte IV

Capítulo 4

Capítulo 4

Resultado y discusión

4.1. Solución de alto nivel

Las tecnologías de Cloud presentan varias ventajas entre ellas es la integración de servicios a las infraestructuras existentes de las empresas y que además son muy adecuados para exponer los modelos de aprendizaje automático por su escalabilidad, capacidad de procesamiento y disponibilidad.

El modulo historial de ruta se comunica con el modelo de aprendizaje automático alojado en un cloud a través de un API (application programming interface) para realizar inferencias de los trayectos que se obtiene de la base de datos de Waypoint y detectar trayectos anómalos junto a aquellas observaciones que son anómalas, en la figura 4.1 se presenta la solución de alto nivel.

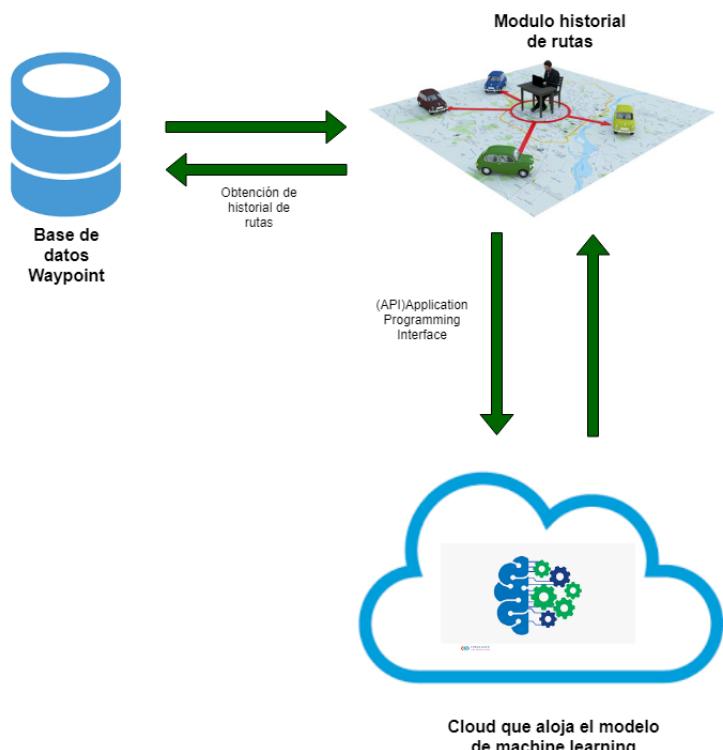


Figura 4.1: Solución de alto nivel

4.2. Análisis de software

4.2.1. Requerimientos funcionales

Requerimiento funcional	Descripción
RF 01	El módulo histórico de rutas debe desplegar el trayecto de un vehículo en un mapa interactivo.
RF 02	El módulo histórico de rutas debe mostrar los vehículos asociados a un cliente.
RF 03	El módulo histórico de rutas debe desplegar el trayecto entre fecha de inicio y una fecha final.
RF 04	Al desplegar el trayecto de vehículo se debe indicar los puntos outlier.
RF 05	El módulo histórico de rutas debe desplegar un cuadro con los valores del trayecto.

Cuadro 4.1: Tabla de requerimientos funcionales

4.2.2. Requerimientos no funcionales

Requerimiento no funcional	Descripción
RNF 01	Para el despliegue del mapa se debe usar el API Leaflet versión 1.4.0.
RNF 02	La conexión a la cloud debe realizarse por el puerto localhost:5000.
RNF 03	El envío de información de los trayectos al API debe ser en formato json.

Cuadro 4.2: Tabla de requerimientos no funcionales

4.3. Diseño de software

4.3.1. Descripción de la arquitectura

La arquitectura de software es la abstracción del diseño y implementación de la estructura de un sistema a alto nivel, usando varios elementos arquitectónicos. Para este proyecto se usó el modelo “4+1” de Kruchten en figura 4.2, este modelo describe la arquitectura de software mediante 5 vistas.

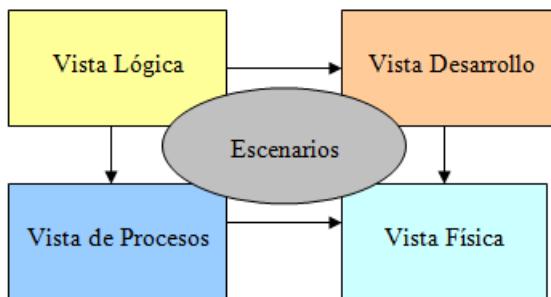


Figura 4.2: Modelo 4+1 de Kruchten

- **Vista lógica:** describe la funcionalidad del sistema y usa un diseño orientado a objetos.
- **Vista de procesos:** describe los aspectos dinámicos del sistema y se enfoca en el comportamiento del sistema en tiempo de ejecución.

- **Vista de desarrollo:** describe la organización de los componentes del software en su ambiente de trabajo e ilustra el sistema desde la perspectiva del programador.
- **Vista física:** describe el mapeo del sistema en el hardware, así como las conexiones físicas entre componentes.
- **Vista de escenario:** tiene la función de unir y relacionar las 4 vistas, los escenarios describen interacciones entre procesos.

4.3.2. Vista de escenario (Diagrama de casos de uso)

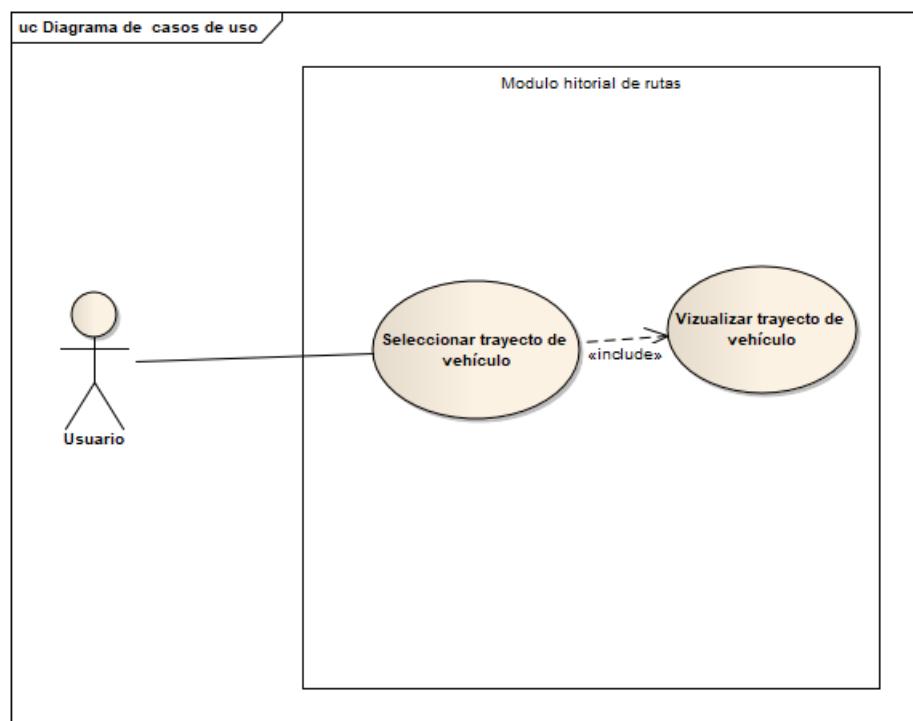


Figura 4.3: Diagrama de casos de uso

4.3.3. Vista lógica (Diagrama de clases)

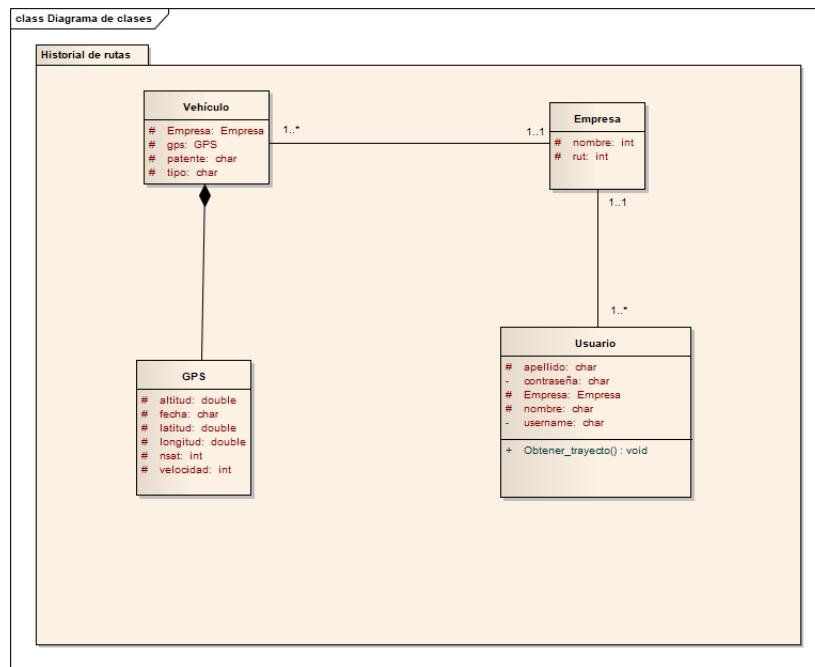


Figura 4.4: Diagrama de clases

4.3.4. Vista de desarrollo (Diagrama de componentes)

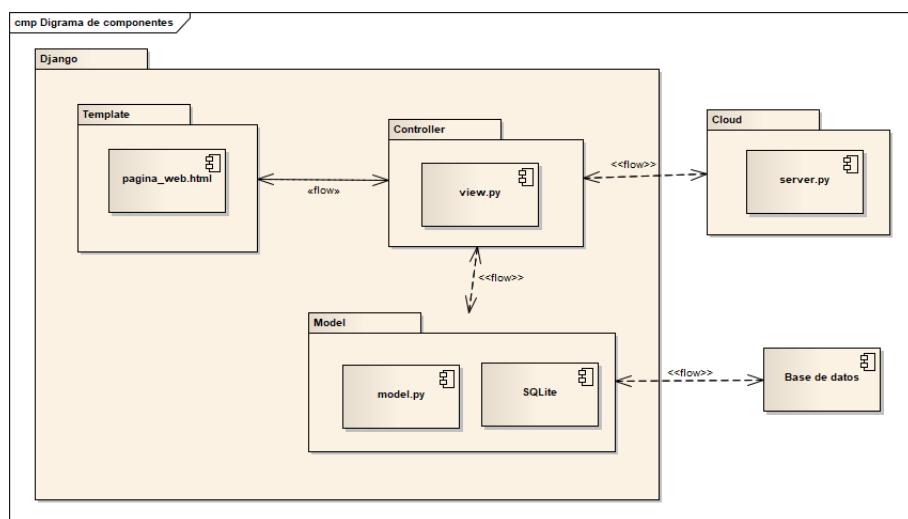


Figura 4.5: Diagrama de componentes

4.3.5. Vista física (Diagrama de despliegue)

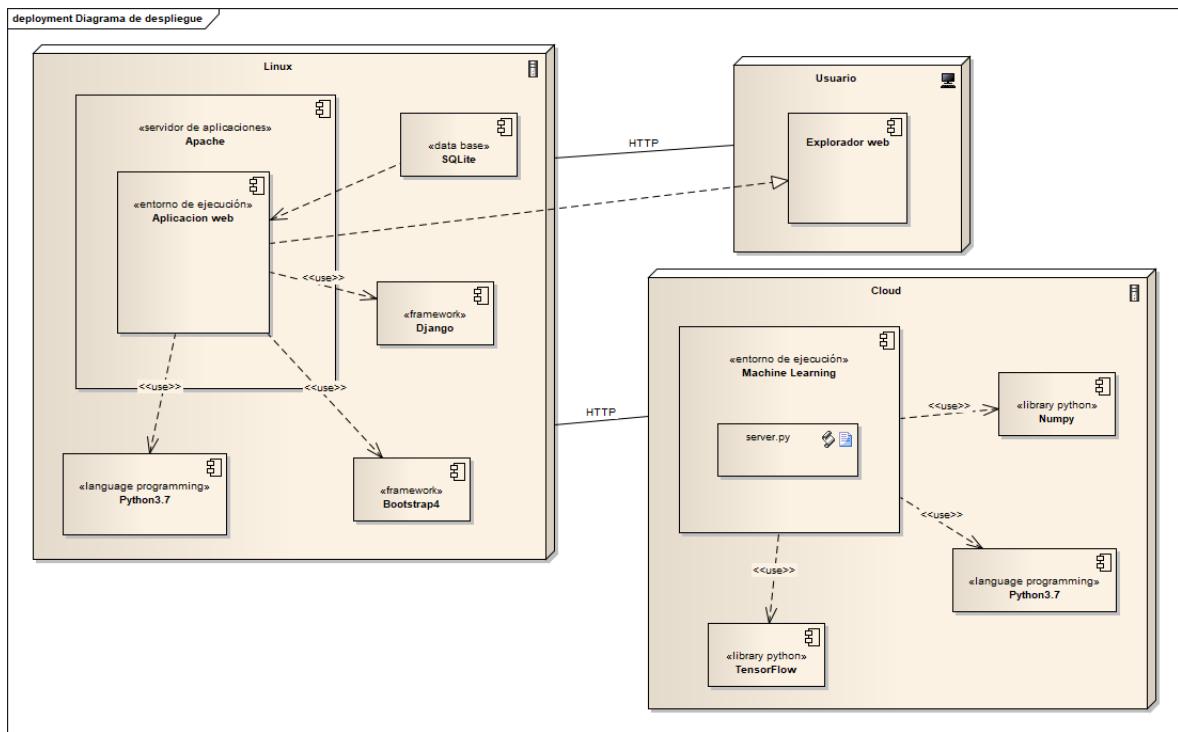


Figura 4.6: Diagrama de despliegue

4.3.6. Vista de procesos (Diagrama de secuencia)

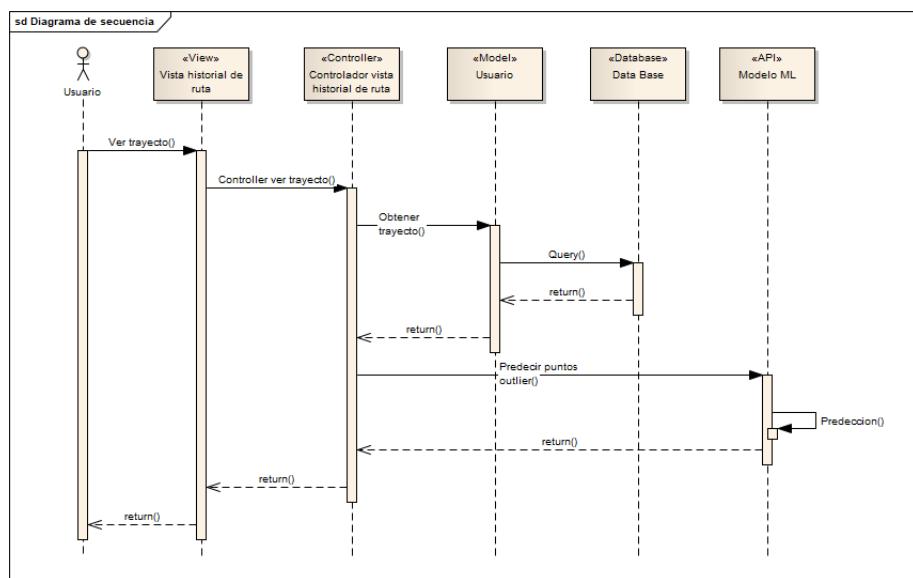


Figura 4.7: Diagrama de secuencia

4.4. Interfaz gráfica de módulo historial de rutas

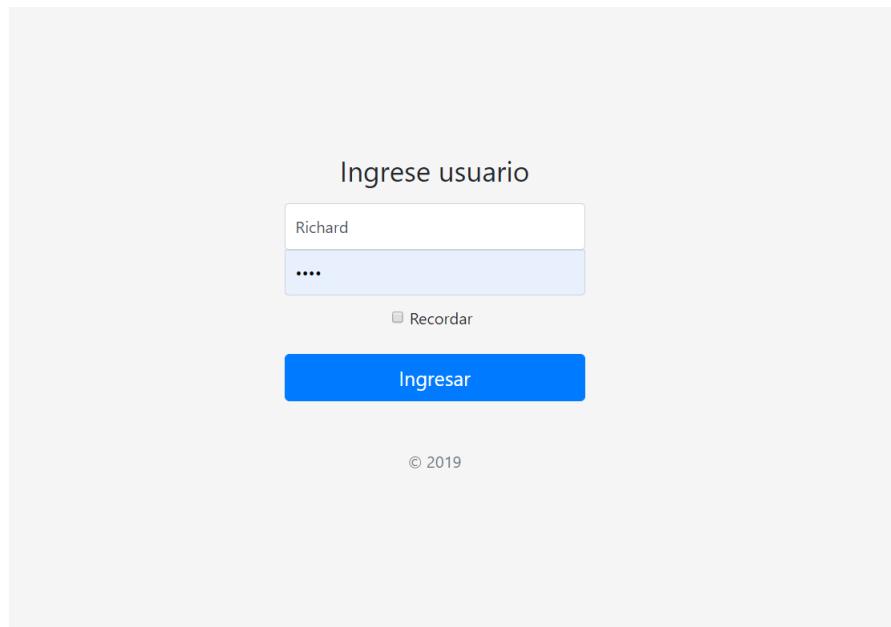


Figura 4.8: GUI01-Inicio sesión

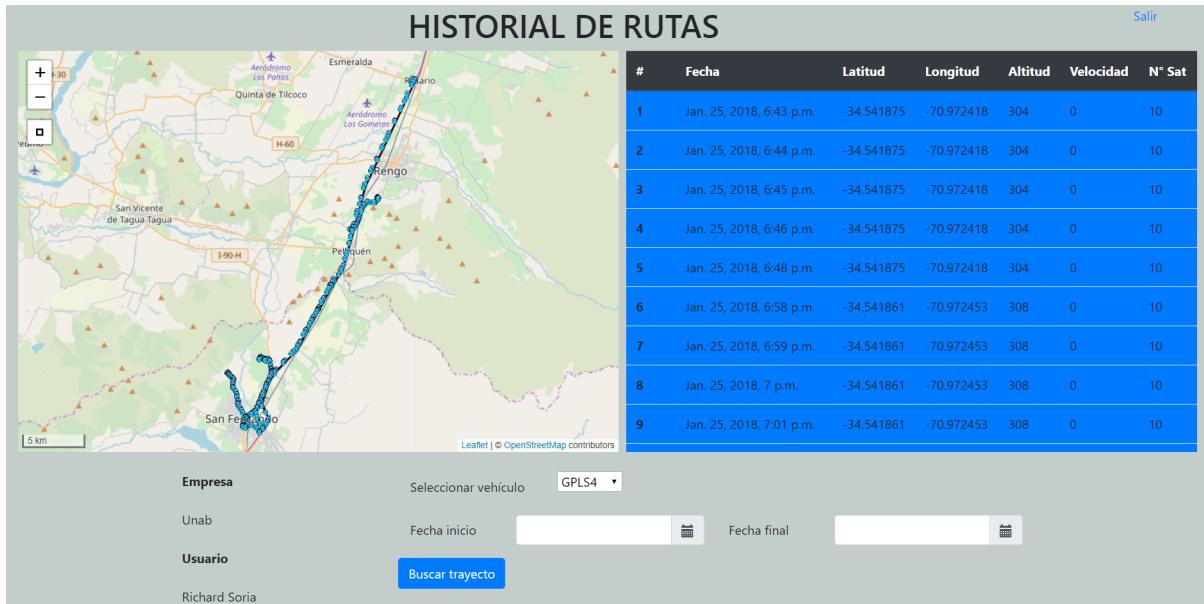


Figura 4.9: GUI02-Historial de rutas

4.5. Pruebas unitarias

Las pruebas unitarias fueron testeadas por la herramienta Coverage.py, la cual mide la cobertura del código, en la figura 4.10 se muestra los resultados del las pruebas unitarias. El detalle de las pruebas se encuentran en la carpeta htmlcov, ubicada en el directorio raíz del modulo historial de rutas.

Module ↓	statements	missing	excluded	coverage
historial_rutas__init__.py	0	0	0	100%
historial_rutas\admin.py	7	0	0	100%
historial_rutas\migrations\0001_initial.py	7	0	0	100%
historial_rutas\migrations\0002_remove_usuario_name.py	4	0	0	100%
historial_rutas\migrations__init__.py	0	0	0	100%
historial_rutas\models.py	31	9	0	71%
Total	49	9	0	82%

Figura 4.10: Pruebas unitarias

4.6. Conjunto de datos

Para el entrenamiento de los modelos propuestos se creo 4 conjuntos de entrenamientos(dataset), dado que el objetivo es detectar trayectos de vehículos outlier los modelos deben recibir como entrada trayectos de vehículos que los podemos representar como series de tiempo que definimos en la sección 2.8. Se identifico 2 tipos de outlier, para cada uno de ellos se tomo trayectos cortos conformados por 5 observaciones y trayectos largos conformados por 40 observaciones, en la empresa Waypoint los 2 tipos de outlier se identificaron en trayectos cortos, aunque creemos que es factible que hayan trayectos largos que sean anómalos y que aportan mas información para la detección de estos outlier, en la tabla 4.3 se muestra un resumen de los conjuntos de datos que usaremos.

	Tipo de outlier	Tipo de trayecto
Dataset 1	Outlier espacial	Trayectos cortos
Datseta 2	Outlier espacial	Trayectos largos
Datseta 3	Outlier velocidad	Trayectos cortos
Dataset 4	Outlier velocidad	Trayectos largos

Cuadro 4.3: Tabla de resumen de conjuntos de entrenamientos

4.7. Método base (LOF)

Se considero un método base de aprendizaje no profundo - Local Outlier Factor(LOF), en la sección 2.9.4 se explica este método. Para este método solo se uso el dataset 4 que contiene los outlier de velocidad, el método LOF recibe como entrada las serie de tiempo y devuelve un vector con valores 1 y -1, el valor 1 indica que la serie de tiempo es outlier y el valor -1 indica que la serie de tiempo es normal. Cada observación cuenta con seis variables fecha(f), latitud(lat),longitud(lon),altitud(alt),velocidad(v) y numero de satélites(sat), para la detección de observaciones outlier se uso regresiones lineales, cada variable se toma como una variable dependiente que depende de las variables mencionadas y la variable que se tomo como dependiente tomara valores del tiempo anterior, si aplicamos este método para cada variable se obtiene seis regresiones lineales de la siguiente forma.

$$f_t = F(f_{t-1}, lat_t, lon_t, alt_t, v_t, sat_t)$$

$$lat_t = F(f_t, lat_{t-1}, lon_t, alt_t, v_t, sat_t)$$

$$lon_t = F(f_t, lat_t, lon_{t-1}, alt_t, v_t, sat_t)$$

$$alt_t = F(f_t, lat_t, lon_t, alt_{t-1}, v_t, sat_t)$$

$$v_t = F(f_t, lat_t, lon_t, alt_t, v_{t-1}, sat_t)$$

$$sat_t = F(f_t, lat_t, lon_t, alt_t, v_t, sat_{t-1})$$

Luego se aplica las regresiones lineales calculadas a las observaciones de la serie de tiempo, obteniendo una predicción de la serie de tiempo. Para decidir cual observación es outlier se utilizo el criterio del error de reconstrucción explicado en la sección 3.5. Para la evaluación del modelo solo le utilizo la evaluación cualitativa debido que este algoritmo tiene un aprendizaje perezoso el no tiene un proceso de aprendizaje.

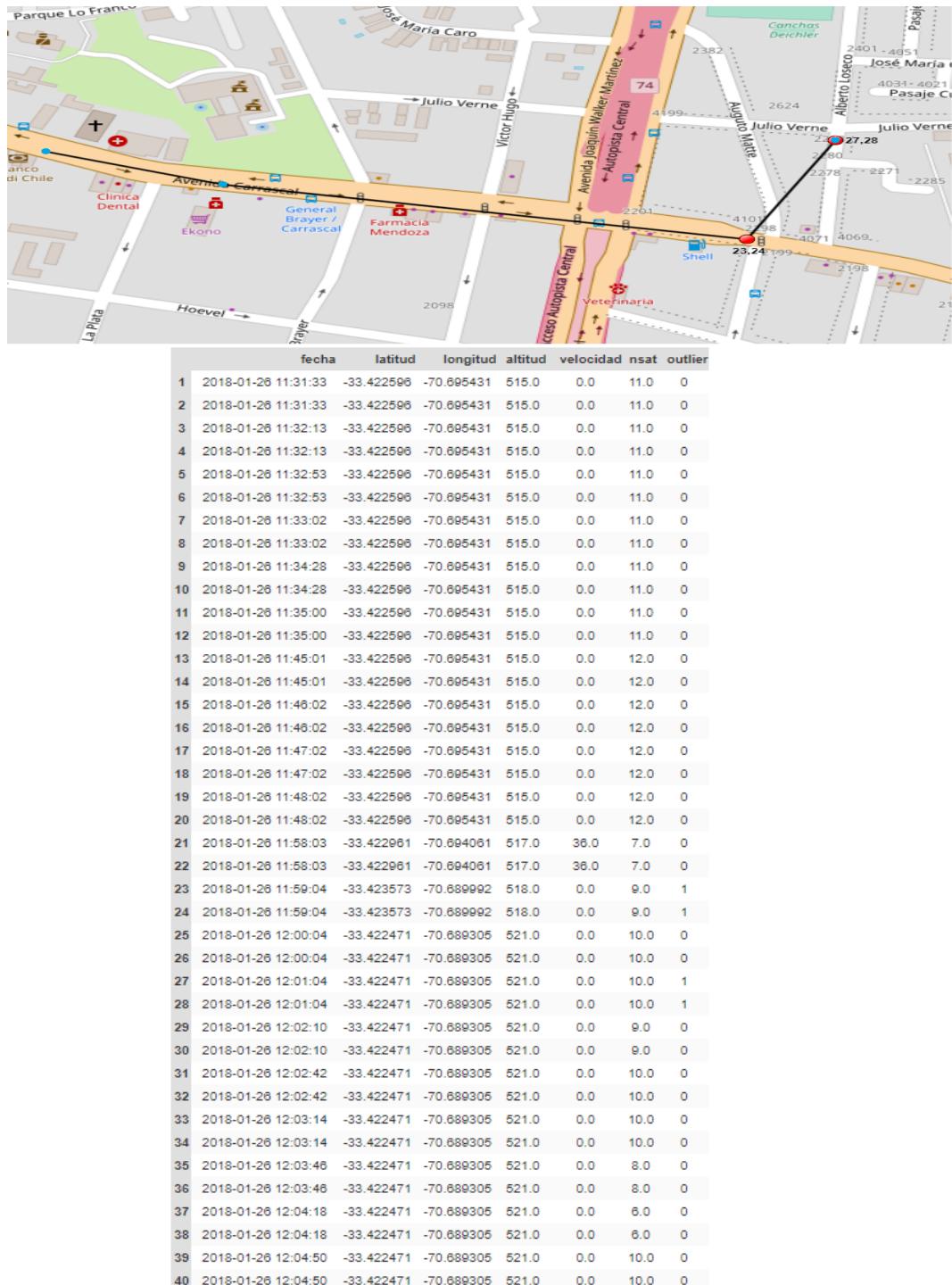
4.7.1. Configuración Local Outlier Factor(LOF)

El LOF se implemento en python 3.7 utilizando la librería Scikit-Learn, dentro de la configuración del LOF se uso los parámetros por defecto Pedregosa y cols. (2011), excepto en el número de vecinos, donde se uso 20 vecinos cercanos. En la evaluación cualitativa del LOF se detecto 0 % de series de tiempo outlier, este resultado se debe a que el LOF

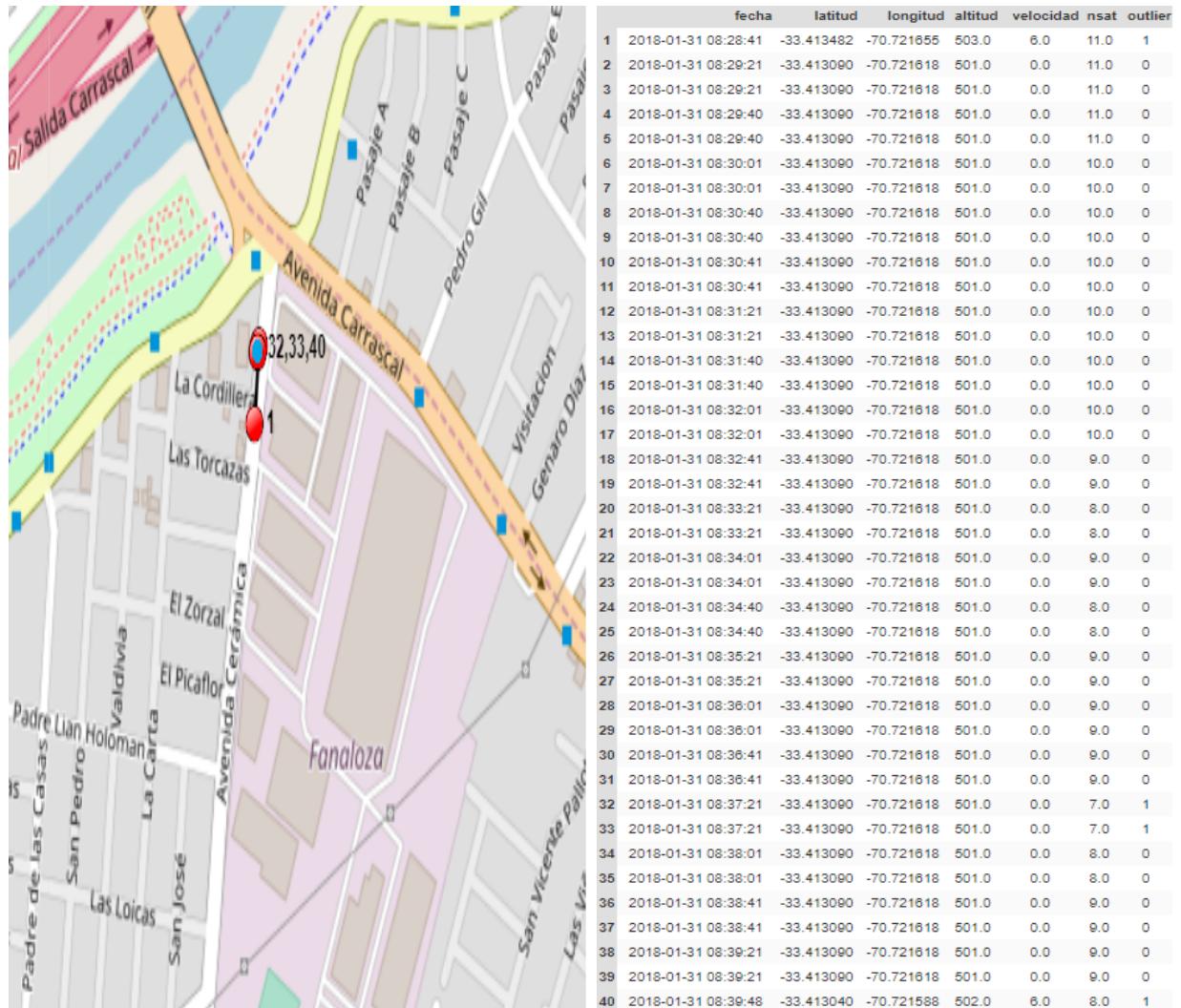
no detecta outlier en datos secuenciales sino en instancias independientes; y además que no relaciona las variables como lo hace una red neuronal, sino que detecta outlier en base a la densidad de los datos. A continuación solo mostrara 2 ejemplos de series de tiempo normales pero que el LOF detecto como anómalos.

4.7.2. Ejemplos de serie de tiempo con LOF

Este trayecto muestra una continuidad normal en todas las observaciones, las observaciones 23, 24, 27 y 28 no presentan un cambio brusco en la velocidad, sin embargo son detectadas como outlier.



Este trayecto muestra una continuidad normal en todas las observaciones, las observaciones 1, 32, 33 y 40 no presentan un cambio brusco en la velocidad, sin embargo son detectadas como outlier.



4.8. Modelos propuestos

4.8.1. Red autoencoder apilada

El primero modelo usa capas densamente conectadas, esto quiere decir que todas las neuronas de una capa L están conectadas con todas las neuronas de la capa siguiente $L + 1$, por lo tanto la entrada de una neurona de la capa $L + 1$ es la correlación de todas las salidas de las neuronas de la capa L . El modelo recibe como entrada una serie de tiempo que tiene un conjunto de observaciones multivariadas y correlaciona todas las observaciones las cuales cuentan con seis variables cada una, haciendo que el número de conexiones dependa del número de observaciones y variables. Para detectar si una serie de tiempo es outlier, el modelo correlaciona el contexto de la serie de tiempo, osea el modelo correlaciona todas las variables de todas las observaciones.

4.8.2. Red autoencoder convolucional apilada

Una serie de tiempo es outlier cuando sus observaciones tienen un comportamiento anómalo respecto a las demás observaciones, por lo tanto no es necesario correlacionar todas las observaciones debido a que un cambio anómalo de una observación se basa en las observaciones adyacentes. El segundo modelo propuesto correlaciona un conjunto de observaciones vecinas, lo cual reduce el número de conexiones. El modelo usa un filtro o ventana deslizante para correlacionar las observaciones vecinas, para ello usa las capas convolucionales que usan filtros que correlacionan las observaciones, este procedimiento se aplica a las capas siguientes. Con este método no se fuerza al modelo a buscar correlaciones entre todas las observaciones, al contrario, nos permite correlacionar las observaciones que deseamos. A continuación se explica el diseño de los filtros que se usaron en la red neuronal convolucional apilada, ver figura 4.13.

Partiremos con la entrada de la red autoencoder convolucional apilada que recibe como entrada una serie de tiempo en forma de cuadrícula de dos dimensiones, ver sección 3.7 para ver transformación.

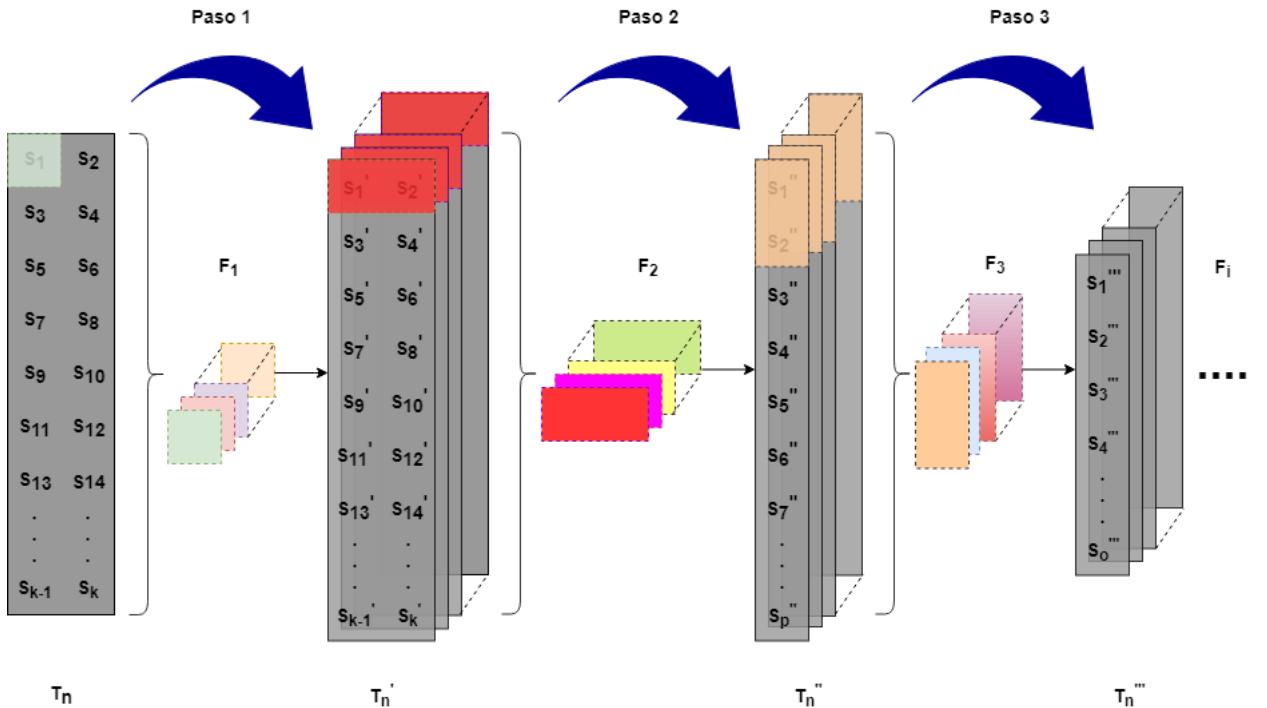


Figura 4.13: Modelo propuesto

Paso 1

Dado una serie de tiempo $T_n = \{S_1, S_2, S_3, \dots, S_k\}$, donde $S_k = \{s_i^1, s_i^2, s_i^3, \dots, s_i^d\}$ es una observación multidimensional $S_k \in R^d$, primero se busca la correlación entre las variables de la observación S_k , por lo tanto el tamaño del primer filtro es d y se aplica el filtro F_1 a cada observación obteniendo una serie de tiempo transformada $T_n' = \{S'_1, S'_2, S'_3, \dots, S'_k\}$, donde S'_k es la correlación de las variables de S_k y también es unidimensional $S'_k \in R$.

Paso 2

Dado la serie de tiempo transformada T_n' , buscamos la correlación para cada fila de la serie transformada, la cual esta formada por una par de observaciones. Como S'_k es unidimensional el tamaño del filtro es dos. Se aplica a cada fila el filtro F_2 y se obtiene una nueva serie de tiempo transformada $T_n'' = \{S''_1, S''_2, S''_3, \dots, S''_k\}$, donde S''_k es unidimensional.

Paso 3

A diferencia de los pasos anteriores donde el tamaño de los filtros siempre son los mismos, en esta paso se busca la correlación entre observaciones con diferentes tamaños

de filtros, desde esta paso uno puede diseñar libremente el tamaño de los filtros y los pasos a mover el filtro. Entonces dado la serie de tiempo transformada T_n'' se aplica un filtro F_3 y se obtiene una nueva serie de tiempo transformada $T_n''' = \{S_1''', S_2''', S_3''', \dots, S_k'''\}$, este paso se repite hasta obtener un espacio latente.

4.8.3. Configuración de modelos

Para los dos modelos propuestos: red autoencoder apilado y red autoencoder convolucional apilado se uso la función de activación tangente hiperbólica y descenso de gradiente estocástico como algoritmo optimizador. En cuanto a las métricas de evaluación se uso el coeficiente de determinación y varianza explicada, para la función de costo se uso el error cuadrático medio, todo esta explicado en el punto 3.10.

Una arquitectura de una red neuronal es la forma como se organizan las neuronas en su interior. Se entrenaron tres arquitecturas diferentes para cada dataset, manteniendo las configuraciones mencionadas pero variando el número de capas y número de neuronas para cada arquitectura, todos las arquitecturas se entrenaron con el mismo número de épocas(300) para mantener las mismas condiciones de aprendizaje. Para especificar cada arquitectura, usaremos la siguiente notación que denota los tipos de capa que se utilizaron:

- D = Capas densas o totalmente conectadas
- C = Capas convolucionales
- TC = Capas transpuestas o deconvolucionales

Las metricas utilizadas son denotadas de la siguiente manera:

- MSE = Error cuadrático medio
- R^2 = Coeficiente de determinación
- EVS = Varianza explicada

4.9. Resultados cuantitativos

4.9.1. Autoencoder apilado aplicado a dataset 1

La arquitectura C obtuvo el mejor resultado, la cual cuenta con 6D; la arquitectura A y la arquitectura B cuentan con 4D.

	MSE		R^2		EVS	
	Train	Valid	Train	Valid	Train	Valid
Arquitectura A	3.888e-7	1.3088e-7	0.99997	0.999985	0.99997	0.99998
Arquitectura B	3.6265e-7	1.234e-7	0.99997	0.999987	0.99997	0.99999
Arquitectura C	2.9139e-7	1.0942e-7	0.99998	0.999987	0.99998	0.99998

Cuadro 4.4: Tabla de resultados autoencoder apilado

4.9.2. Autoencoder convolucional apilado aplicado a dataset 1

La arquitectura B obtuvo el mejor resultado, la cual cuenta con 2C, 2D y 2TC; la arquitectura A cuenta con 3C, 2D y 3TC y la arquitectura C cuenta con 2C, 2D y 2TC. La arquitectura B y C cuentan con un filtro de menor tamaño que el de la arquitectura C.

	MSE		R^2		EVS	
	Train	Valid	Train	Valid	Train	Valid
Arquitectura A	2.7117e-7	1.0446e-7	0.999983	0.9999879	0.999983	0.9999913
Arquitectura B	2.4559e-7	9.1347e-8	0.999985	0.9999897	0.999985	0.9999935
Arquitectura C	2.5914e-7	9.4499e-8	0.999984	0.9999894	0.999984	0.9999931

Cuadro 4.5: Tabla de resultados autoencoder convolucional apilado

4.9.3. Autoencoder apilado aplicado a dataset 2

La arquitectura B obtuvo el mejor resultado, la cual cuenta con 8D; la arquitectura A y C cuentan con 6D.

	MSE		R^2		EVS	
	Train	Valid	Train	Valid	Train	Valid
Arquitectura A	1.335e-6	8.3177e-7	0.999915	0.99990	0.999915	0.99991
Arquitectura B	8.633e-7	4.099e-7	0.999945	0.99995	0.999946	0.99996
Arquitectura C	8.8594e-7	4.56e-7	0.999943	0.99994	0.999944	0.99995

Cuadro 4.6: Tabla de resultados autoencoder apilado

4.9.4. Autoencoder convolucional apilado aplicado a dataset 2

La arquitectura A obtuvo el mejor resultado, la cual cuenta con 4C, 2D y 4TC; la arquitectura B cuenta con 3C, 2D y 3TC y la arquitectura C cuenta con 4C, 2D y 4TC, el filtro de la arquitectura A es de menor tamaño que el de las otras arquitecturas.

	MSE		R^2		EVS	
	Train	Valid	Train	Valid	Train	Valid
Arquitectura A	1.7698e-6	1.3554e-6	0.9999	0.9998	0.9999	0.9998
Arquitectura B	2.9682e-6	2.3015e-6	0.9998	0.9997	0.9998	0.9997
Arquitectura C	5.4754e-6	4.3883e-6	0.9997	0.9995	0.9997	0.9995

Cuadro 4.7: Tabla de resultados autoencoder convolucional apilado

4.9.5. Autoencoder apilado aplicado a dataset 3

La arquitectura B obtuvo el mejor resultado, la cual cuenta con 4D; la arquitectura A cuenta con 6D y la arquitectura C cuenta con 4D.

	MSE		R^2		EVS	
	Train	Valid	Train	Valid	Train	Valid
Arquitectura A	1.2654e-4	1.1692e-4	0.9976	0.9959	0.9976	0.9960
Arquitectura B	4.9402e-6	4.2375e-6	0.9999	0.9999	0.9999	0.9999
Arquitectura C	8.8641e-5	7.4975e-5	0.9983	0.9975	0.9983	0.9976

Cuadro 4.8: Tabla de resultados autoencoder apilado

4.9.6. Autoencoder convolucional apilado aplicado a dataset 3

La arquitectura B obtuvo el mejor resultado, la cual cuenta con 3C, 2D y 3TC; la arquitectura A cuenta con 3C, 2D y 3TC y la arquitectura C cuenta con 2C, 2D y 2TC, el filtro de la arquitectura B es de menor tamaño que el de las arquitecturas A y C.

	MSE		R^2		EVS	
	Train	Valid	Train	Valid	Train	Valid
Arquitectura A	1.6204e-6	1.5368e-6	0.99996	0.99995	0.99996	0.99995
Arquitectura B	1.2402e-6	1.1958e-6	0.99997	0.99996	0.99997	0.99996
Arquitectura C	1.2624e-5	1.1092e-5	0.99980	0.99960	0.99980	0.99960

Cuadro 4.9: Tabla de resultados autoencoder convolucional apilado

4.9.7. Autoencoder apilado aplicado a dataset 4

La arquitectura C obtuvo el mejor resultado, la cual cuenta con 10D; la arquitectura A cuenta con 8D y la arquitectura B cuenta con 6D.

	MSE		R^2		EVS	
	Train	Valid	Train	Valid	Train	Valid
Arquitectura A	2.255e-4	2.1979e-4	0.9957	0.9922	0.9957	0.9922
Arquitectura B	2.9738e-4	2.8812e-4	0.9943	0.9898	0.9944	0.9899
Arquitectura C	1.3529e-4	1.278e-4	0.9974	0.9955	0.9974	0.9955

Cuadro 4.10: Tabla de resultados autoencoder apilado

4.9.8. Autoencoder convolucional apilado aplicado a dataset 4

La arquitectura A obtuvo el mejor resultado, la cual cuenta con 4C, 2D y 4TC; la arquitectura B cuenta con 4C, 2D y 4TC y la arquitectura C cuenta con 3C, 2D y 3TC, el filtro de la arquitectura A es de menor tamaño que el de las arquitecturas B y C.

	MSE		R^2		EVS	
	Train	Valid	Train	Valid	Train	Valid
Arquitectura A	1.0563e-4	9.2084e-5	0.9980	0.9980	0.9980	0.9980
Arquitectura B	8.2154e-5	6.8964e-5	0.9984	0.9976	0.9984	0.9977
Arquitectura C	8.7996e-5	7.5401e-5	0.9983	0.9974	0.9983	0.9974

Cuadro 4.11: Tabla de resultados autoencoder convolucional apilado

4.10. Resultados cualitativos

La evaluación cualitativa solo se aplico a la mejor arquitectura del autoencoder apilado y autoencoder convolucional apilado de cada dataset. En los resultados se muestra que el peor resultado cualitativo se obtuvo del autoencoder apilado convolucional aplicado al dataset 2, analizando los resultados del autoencoder se observo que la gran mayoría de los trayectos se focalizan en un región específica del mapa, es muy probable que el autoencoder encontró patrones específicos de geolocalización en esa región con respecto a otras regiones.

Dado que este es un problema de aprendizaje no supervisado, no se sabe cuál sera el resultado para los modelos propuestos, en los 4 dataset se encontró cambios bruscos de velocidad entre observaciones consecutivas mayores a 100 km/h, las cuales se tomaron como observaciones anómalas. También los modelos propuestos encontraron los outlier espaciales en los 4 dataset. Se contó como outlier detectados los cambios bruscos de velocidad y los outlier espaciales, obteniendo los siguientes resultados en la tabla 4.12.

	Red neuronal	Outlier detectados
Data set 1	Autoencoder apilado(Arq. C)	51.5 %
	Autoencoder convolucional apilado(Arq. B)	39 %
Data set 2	Autoencoder apilado(Arq. B)	47 %
	Autoencoder convolucional apilado(Arq. A)	4 %
Data set 3	Autoencoder apilado(Arq. B)	20.5 %
	Autoencoder convolucional apilado(Arq. B)	15 %
Data set 4	Autoencoder apilado(Arq. C)	87.5 %
	Autoencoder convolucional apilado (Arq. A)	68.5 %

Cuadro 4.12: Tabla de resultados cualitativos

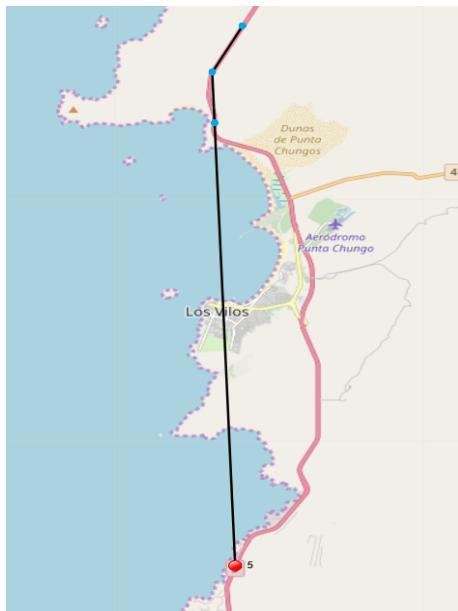
4.11. Resultados de las series de tiempo ilustradas en mapa

En esta sección se mostraran 2 ejemplos visuales de series de tiempo outlier detectadas por el modelo y 2 ejemplos de series de tiempo que no son outlier pero que el modelo detecto como outlier con sus respectivos mapas y valores, estos ejemplos visuales son extraidos de los 100 series de tiempo con mas error de reconstrucción. Esto es aplicado para la mejor red neuronal de cada dataset que se evaluó en el anterior punto. En el apéndice B se muestran más ejemplos.

4.11.1. Dataset 1(Autoencoder apilado)

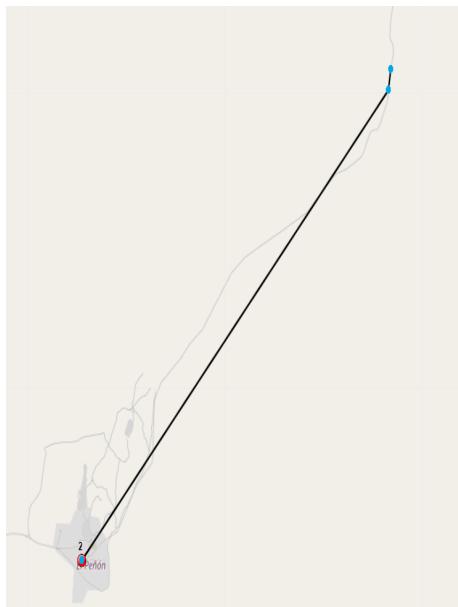
Ejemplos de trayectos outlier

Este trayecto muestra una discontinuidad evidente, la observación 5 sale de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-31.823953	-71.506471	127.0	0
2	-31.838311	-71.515168	81.0	0
3	-31.854066	-71.514546	57.0	0
4	-31.990640	-71.508551	26.0	0
5	-31.990640	-71.508551	26.0	1

Este trayecto muestra una discontinuidad evidente, la observación 2 sale de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-24.413796	-69.497358	1799.0	0
2	-24.413772	-69.497349	1800.0	1
3	-24.413772	-69.497349	1800.0	0
4	-24.286228	-69.360710	2142.0	0
5	-24.280403	-69.359710	2119.0	0

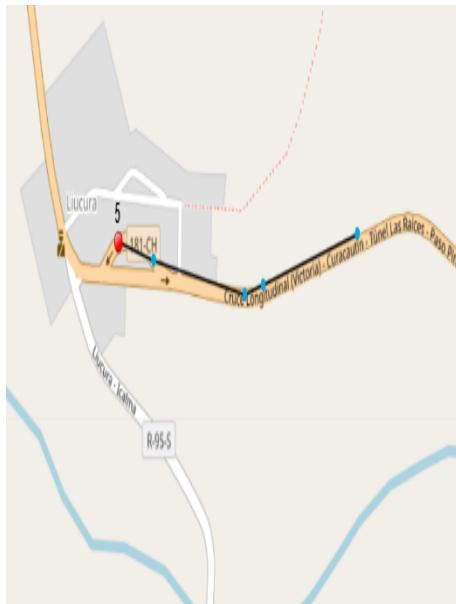
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo la observación 5 es detectada como outlier.



	latitud	longitud	altitud	outlier
1	-38.701807	-71.098793	1087.0	0
2	-38.702308	-71.099600	1085.0	0
3	-38.702503	-71.099688	1085.0	0
4	-38.702833	-71.100051	1085.0	0
5	-38.702866	-71.100183	1084.0	1

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo la observación 5 es detectada como outlier.



	latitud	longitud	altitud	outlier
1	-38.645260	-71.081258	1045.0	0
2	-38.645720	-71.084020	1042.0	0
3	-38.645801	-71.084550	1042.0	0
4	-38.645505	-71.087213	1039.0	0
5	-38.645336	-71.088241	1038.0	1

4.11.2. Dataset 1(Autoencoder convolucional apilado)

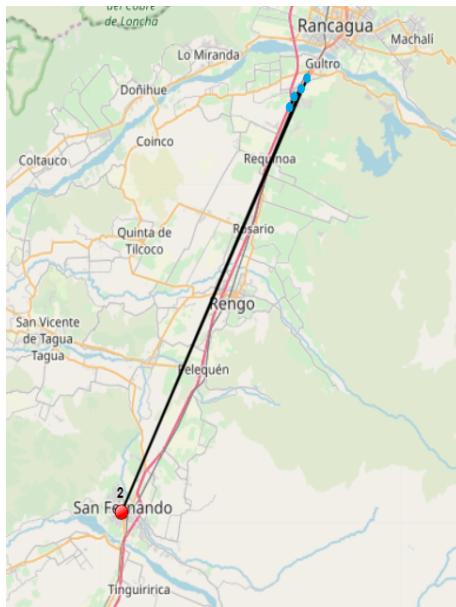
Ejemplos de trayectos outlier

Este trayecto muestra una discontinuidad evidente, la observación 1 sale de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-20.252940	-69.511786	1166.0	1
2	-20.260555	-69.770096	1034.0	0
3	-20.260555	-69.770096	1034.0	0
4	-20.260555	-69.770096	1034.0	0
5	-20.260555	-69.770096	1034.0	0

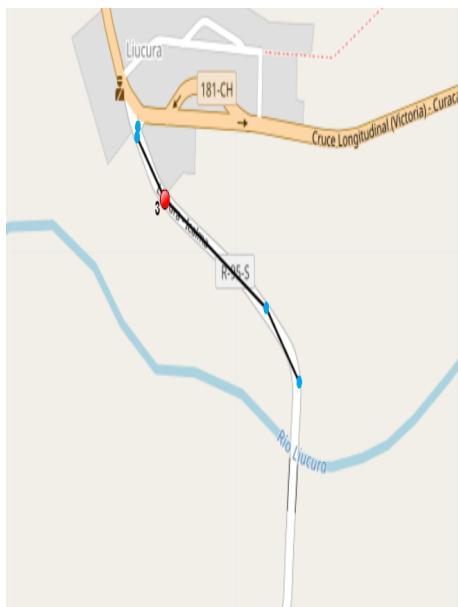
Este trayecto muestra una discontinuidad evidente, la observación 2 sale de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-34.216669	-70.773638	502.0	0
2	-34.588081	-70.990638	338.0	1
3	-34.225573	-70.781238	491.0	0
4	-34.232500	-70.789638	484.0	0
5	-34.241089	-70.795175	465.0	0

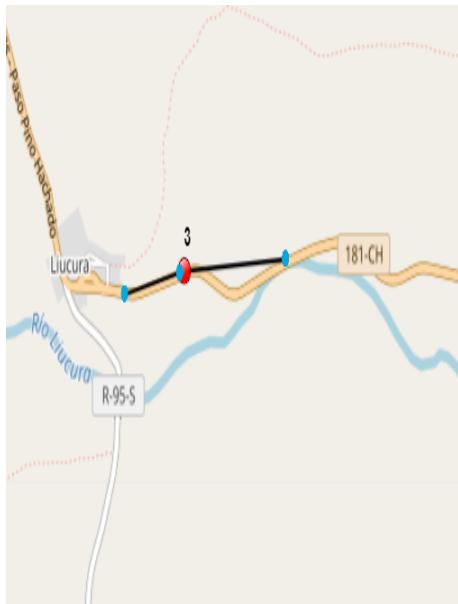
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo la observación 3 es detectada como outlier.



	latitud	longitud	altitud	outlier
1	-38.645723	-71.089463	1036.0	0
2	-38.645819	-71.089476	1037.0	0
3	-38.646408	-71.088803	1036.0	1
4	-38.647451	-71.086283	1037.0	0
5	-38.648151	-71.085460	1039.0	0

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo la observación 3 es detectada como outlier.



	latitud	longitud	altitud	outlier
1	-38.644848	-71.072844	1048.0	0
2	-38.644848	-71.072844	1048.0	0
3	-38.645183	-71.080639	1042.0	1
4	-38.645233	-71.080929	1043.0	0
5	-38.645811	-71.085161	1038.0	0

4.11.3. Dataset 2(Autoencoder apilado)

Ejemplos de trayectos outlier

Este trayecto muestra una discontinuidad evidente, las observaciones 5,3,21,28 salen de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-34.536135	-70.942731	312.0	0
2	-34.256618	-70.803918	450.0	0
3	-34.543380	-70.951210	307.0	1
4	-34.247343	-70.798623	460.0	0
5	-34.550751	-70.959778	311.0	1
6	-34.238566	-70.793516	469.0	0
7	-34.558868	-70.965903	320.0	0
8	-34.234960	-70.791511	473.0	0
9	-34.232381	-70.789273	482.0	0
10	-34.568048	-70.970711	329.0	0
11	-34.229128	-70.784069	483.0	0
12	-34.572810	-70.973036	334.0	0
13	-34.572810	-70.973036	334.0	0
14	-34.221795	-70.777820	493.0	0
15	-34.220941	-70.777121	494.0	0
16	-34.573677	-70.973338	334.0	0
17	-34.573677	-70.973338	334.0	0
18	-34.577721	-70.975618	337.0	0
19	-34.212545	-70.769926	505.0	0
20	-34.209105	-70.766993	510.0	0
21	-34.580661	-70.979630	335.0	1
22	-34.581806	-70.981600	333.0	0
23	-34.204010	-70.762651	512.0	0
24	-34.583856	-70.983899	334.0	0
25	-34.195603	-70.756535	519.0	0
26	-34.584165	-70.984000	335.0	0
27	-34.585895	-70.984588	336.0	0
28	-34.189038	-70.748813	519.0	1
29	-34.587958	-70.985320	338.0	0
30	-34.588881	-70.985650	338.0	0
31	-34.183550	-70.739253	508.0	0
32	-34.590120	-70.986085	340.0	0
33	-34.177256	-70.730776	514.0	0
34	-34.173558	-70.728790	516.0	0
35	-34.594305	-70.987588	344.0	0
36	-34.169169	-70.726999	512.0	0
37	-34.595240	-70.987893	344.0	0
38	-34.595240	-70.987893	344.0	0
39	-34.595308	-70.987851	344.0	0
40	-34.595308	-70.987851	344.0	0

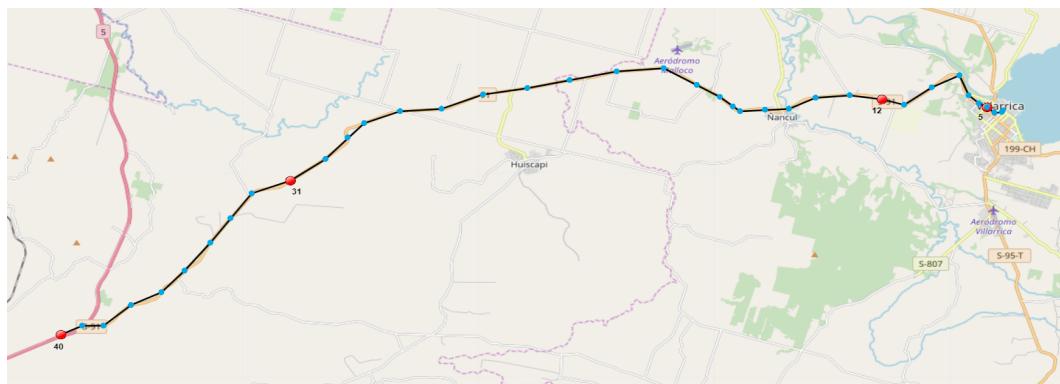
Este trayecto muestra una discontinuidad evidente, las observaciones 30, 32, 37, 40 salen de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-36.776421	-73.057411	11.0	0
2	-36.776421	-73.057411	11.0	0
3	-36.776421	-73.057411	11.0	0
4	-36.776421	-73.057411	11.0	0
5	-36.776421	-73.057411	11.0	0
6	-36.776421	-73.057411	11.0	0
7	-36.776370	-73.057446	7.0	0
8	-36.776366	-73.057498	7.0	0
9	-36.776456	-73.057640	6.0	0
10	-36.777375	-73.058133	4.0	0
11	-36.777420	-73.058103	4.0	0
12	-36.777428	-73.058100	4.0	0
13	-36.777477	-73.057990	5.0	0
14	-36.777444	-73.057933	5.0	0
15	-36.777455	-73.057950	5.0	0
16	-36.777466	-73.057961	6.0	0
17	-36.777472	-73.057968	6.0	0
18	-36.777465	-73.057956	6.0	0
19	-36.777462	-73.057950	6.0	0
20	-36.777462	-73.057950	6.0	0
21	-36.777462	-73.057950	6.0	0
22	-36.777390	-73.057840	7.0	0
23	-36.777390	-73.057840	7.0	0
24	-36.777390	-73.057840	7.0	0
25	-33.374988	-70.721756	486.0	0
26	-33.374988	-70.721756	486.0	0
27	-33.374981	-70.721705	507.0	0
28	-33.374981	-70.721705	507.0	0
29	-33.374981	-70.721705	507.0	0
30	-33.374981	-70.721705	507.0	1
31	-33.374981	-70.721705	507.0	0
32	-33.374981	-70.721705	507.0	1
33	-33.374981	-70.721705	507.0	0
34	-33.374981	-70.721705	507.0	0
35	-33.374981	-70.721705	507.0	0
36	-33.374981	-70.721705	507.0	0
37	-33.374981	-70.721705	507.0	1
38	-33.374951	-70.721701	499.0	0
39	-33.374950	-70.721706	499.0	0
40	-33.374928	-70.721758	498.0	1

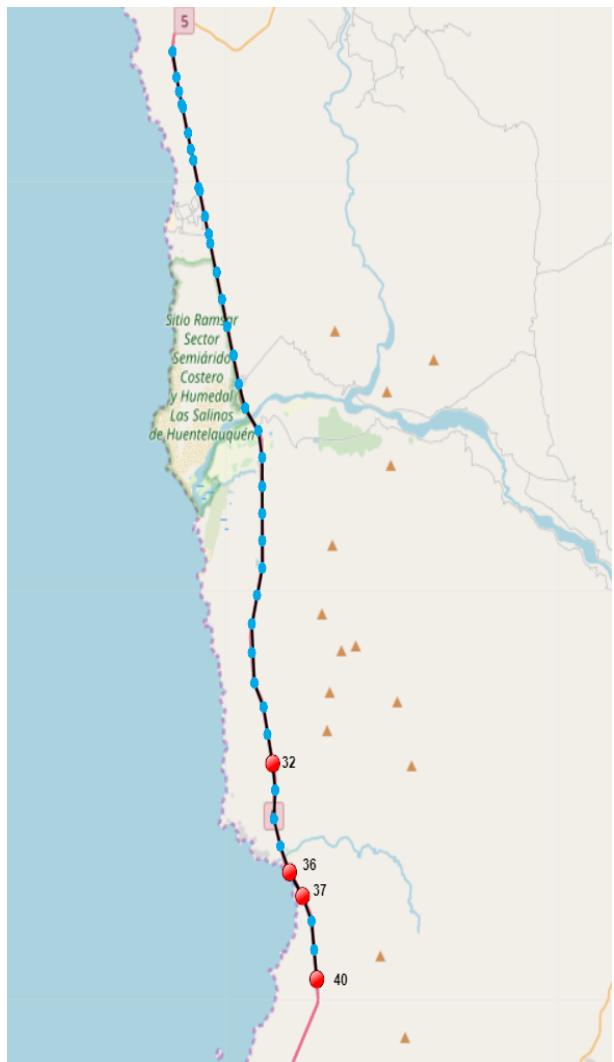
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo las observaciones 5,12,31,40 son detectadas como outlier.



	latitud	longitud	altitud	outlier
1	-39.279484	-72.226635	227.0	0
2	-39.279481	-72.226652	228.0	0
3	-39.280048	-72.226673	229.0	0
4	-39.280296	-72.229577	230.0	0
5	-39.278505	-72.232218	230.0	1
6	-39.278436	-72.232330	230.0	0
7	-39.277193	-72.235305	226.0	0
8	-39.274508	-72.239273	238.0	0
9	-39.267978	-72.242700	257.0	0
10	-39.271996	-72.252961	293.0	0
11	-39.277621	-72.263294	286.0	0
12	-39.275786	-72.271383	283.0	1
13	-39.274471	-72.283373	276.0	0
14	-39.275333	-72.296328	276.0	0
15	-39.279096	-72.306032	267.0	0
16	-39.279258	-72.315186	277.0	0
17	-39.279808	-72.324450	299.0	0
18	-39.278358	-72.327272	305.0	0
19	-39.274058	-72.331920	307.0	0
20	-39.270975	-72.340431	287.0	0
21	-39.265361	-72.352890	268.0	0
22	-39.266661	-72.370318	243.0	0
23	-39.269521	-72.387843	229.0	0
24	-39.272098	-72.403665	244.0	0
25	-39.274140	-72.420085	231.0	0
26	-39.279118	-72.435671	230.0	0
27	-39.279863	-72.451085	211.0	0
28	-39.283851	-72.464411	227.0	0
29	-39.288535	-72.470608	196.0	0
30	-39.295671	-72.479037	214.0	0
31	-39.303071	-72.491985	214.0	1
32	-39.307331	-72.506173	207.0	0
33	-39.315501	-72.514140	213.0	0
34	-39.323619	-72.521770	206.0	0
35	-39.332918	-72.531280	197.0	0
36	-39.340231	-72.540138	175.0	0
37	-39.344408	-72.551448	168.0	0
38	-39.351400	-72.561460	150.0	0
39	-39.351233	-72.569557	146.0	0
40	-39.354270	-72.577438	150.0	1

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo las observaciones 32,36,37,40 son detectadas como outlier.



	latitud	longitud	altitud	outlier
1	-31.456234	-71.571399	164.0	0
2	-31.465419	-71.569128	177.0	0
3	-31.470974	-71.567484	173.0	0
4	-31.475615	-71.566118	155.0	0
5	-31.476646	-71.565813	149.0	0
6	-31.485863	-71.563091	138.0	0
7	-31.492036	-71.561271	149.0	0
8	-31.496103	-71.560068	166.0	0
9	-31.506126	-71.557103	109.0	0
10	-31.507166	-71.556799	107.0	0
11	-31.516481	-71.554054	90.0	0
12	-31.522678	-71.552224	75.0	0
13	-31.526601	-71.551070	75.0	0
14	-31.536919	-71.548021	64.0	0
15	-31.547005	-71.545044	61.0	0
16	-31.557088	-71.542054	62.0	0
17	-31.567626	-71.538901	68.0	0
18	-31.577728	-71.535960	68.0	0
19	-31.586838	-71.532848	13.0	0
20	-31.594968	-71.526096	6.0	0
21	-31.604785	-71.524083	10.0	0
22	-31.615378	-71.523905	43.0	0
23	-31.625304	-71.523746	75.0	0
24	-31.635379	-71.523573	99.0	0
25	-31.645236	-71.523941	98.0	0
26	-31.655416	-71.526538	94.0	0
27	-31.665880	-71.529143	110.0	0
28	-31.676376	-71.529419	120.0	0
29	-31.687081	-71.528095	122.0	0
30	-31.696025	-71.523206	165.0	0
31	-31.706015	-71.520740	141.0	0
32	-31.716373	-71.518251	152.0	1
33	-31.726575	-71.517168	137.0	0
34	-31.736844	-71.517619	106.0	0
35	-31.746849	-71.514071	34.0	0
36	-31.756348	-71.509191	12.0	1
37	-31.765091	-71.502561	62.0	1
38	-31.774473	-71.497539	101.0	0
39	-31.784825	-71.496329	115.0	0
40	-31.795246	-71.495089	98.0	1

4.11.4. Dataset 2(Autoencoder convolucional apilado)

Ejemplos de trayectos outlier

Este trayecto muestra una discontinuidad evidente, las observaciones 1,5,6,7 salen de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-33.423048	-70.779109	459.0	1
2	-33.423106	-70.779183	445.0	0
3	-33.422933	-70.779078	445.0	0
4	-33.423093	-70.778830	456.0	0
5	-33.423093	-70.778830	456.0	1
6	-33.423390	-70.779006	458.0	1
7	-33.423302	-70.779010	463.0	1
8	-36.815610	-73.062275	43.0	0
9	-36.815610	-73.062275	43.0	0
10	-36.815610	-73.062275	43.0	0
11	-36.815610	-73.062275	43.0	0
12	-36.815610	-73.062275	43.0	0
13	-36.815610	-73.062275	43.0	0
14	-36.815610	-73.062275	43.0	0
15	-36.815610	-73.062275	43.0	0
16	-36.815581	-73.062300	35.0	0
17	-36.815581	-73.062300	35.0	0
18	-36.815581	-73.062300	35.0	0
19	-36.815581	-73.062300	35.0	0
20	-36.815581	-73.062300	35.0	0
21	-36.815581	-73.062300	35.0	0
22	-36.815581	-73.062300	35.0	0
23	-36.815581	-73.062300	35.0	0
24	-36.815581	-73.062300	35.0	0
25	-36.815581	-73.062300	35.0	0
26	-36.815581	-73.062300	35.0	0
27	-36.815581	-73.062300	35.0	0
28	-36.815581	-73.062300	35.0	0
29	-36.815581	-73.062300	35.0	0
30	-36.815590	-73.062295	30.0	0
31	-36.815590	-73.062295	30.0	0
32	-36.815590	-73.062295	30.0	0
33	-36.815590	-73.062295	30.0	0
34	-36.815590	-73.062295	30.0	0
35	-36.815590	-73.062295	30.0	0
36	-36.815590	-73.062295	30.0	0
37	-36.815590	-73.062295	30.0	0
38	-36.815590	-73.062295	30.0	0
39	-36.815590	-73.062295	30.0	0
40	-36.815590	-73.062295	30.0	0

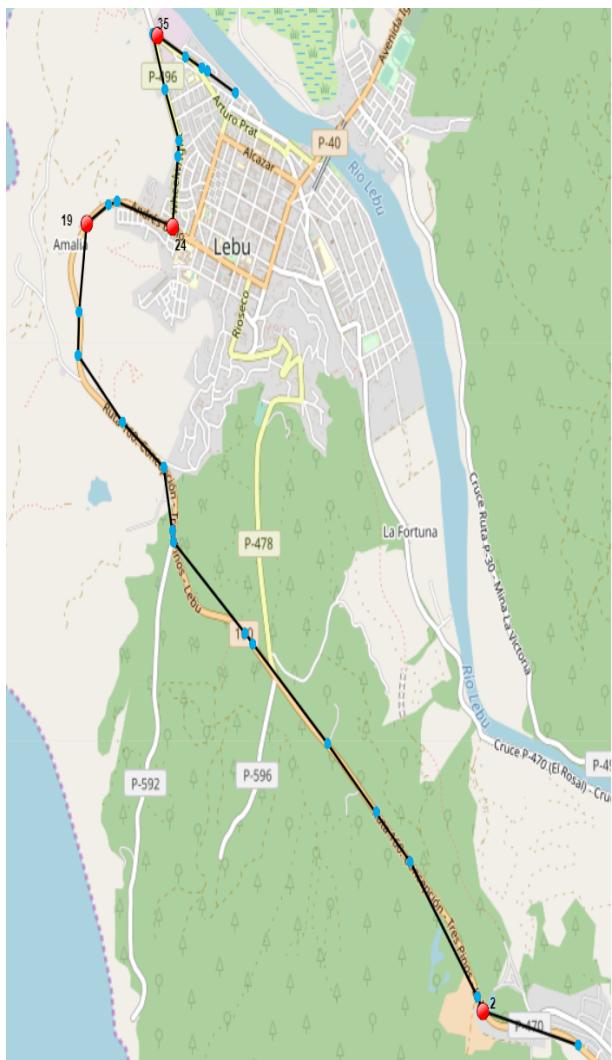
Este trayecto muestra una discontinuidad evidente, las observaciones 30, 38, 39, 40 salen de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-33.469985	-70.771001	473.0	0
2	-33.469985	-70.771001	473.0	0
3	-33.469985	-70.771001	473.0	0
4	-33.469985	-70.771001	473.0	0
5	-33.469985	-70.771001	473.0	0
6	-33.469985	-70.771001	473.0	0
7	-33.469985	-70.771001	473.0	0
8	-33.469985	-70.771001	473.0	0
9	-33.469985	-70.771001	473.0	0
10	-33.469985	-70.771001	473.0	0
11	-33.470008	-70.771000	471.0	0
12	-33.470195	-70.770932	469.0	0
13	-33.470206	-70.770898	468.0	0
14	-33.470238	-70.770946	468.0	0
15	-33.470347	-70.771091	468.0	0
16	-33.470347	-70.771251	468.0	0
17	-33.470347	-70.771251	468.0	0
18	-33.470358	-70.771268	467.0	0
19	-33.470375	-70.771154	467.0	0
20	-33.470388	-70.771163	467.0	0
21	-33.470433	-70.771176	467.0	0
22	-33.470437	-70.771170	472.0	0
23	-33.470437	-70.771170	472.0	0
24	-33.470437	-70.771170	472.0	0
25	-33.470400	-70.771180	472.0	0
26	-33.470327	-70.771093	471.0	0
27	-33.470335	-70.771097	471.0	0
28	-33.470347	-70.771099	471.0	0
29	-33.470383	-70.771108	470.0	0
30	-33.470391	-70.771110	471.0	1
31	-33.470387	-70.771106	471.0	0
32	-33.470405	-70.771111	471.0	0
33	-33.470403	-70.771111	471.0	0
34	-33.470403	-70.771111	471.0	0
35	-33.470403	-70.771111	471.0	0
36	-33.470403	-70.771111	471.0	0
37	-33.470403	-70.771111	471.0	0
38	-23.512723	-70.401990	88.0	1
39	-23.512723	-70.401990	88.0	1
40	-23.512751	-70.401975	87.0	1

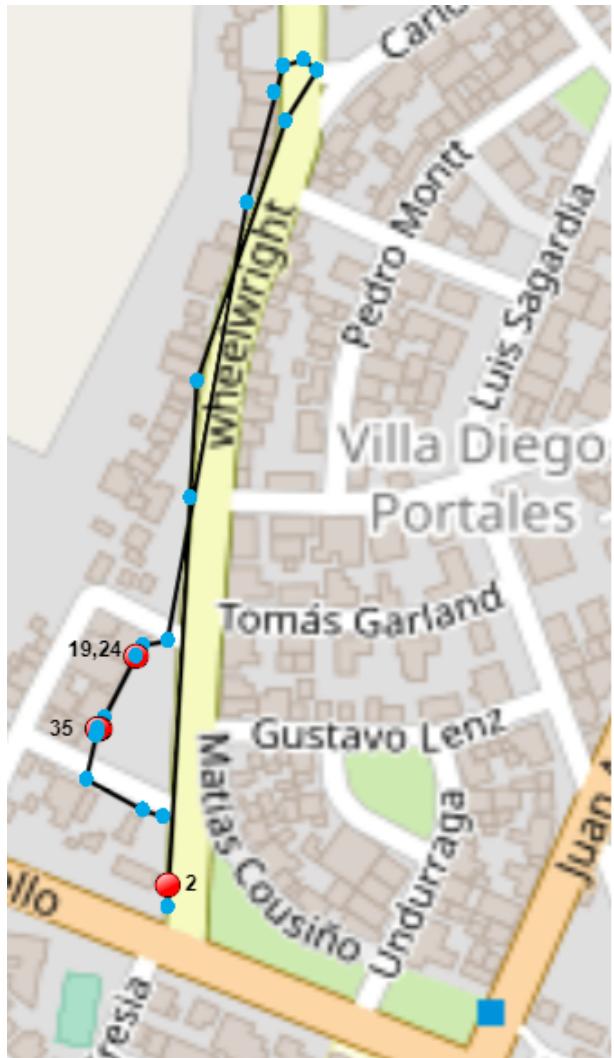
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo las observaciones 2,19,24,35 son detectadas como outlier.



	latitud	longitud	altitud	outlier
1	-37.644981	-73.628571	177.0	0
2	-37.643455	-73.636100	178.0	1
3	-37.642860	-73.636563	177.0	0
4	-37.636953	-73.641858	175.0	0
5	-37.634781	-73.644528	178.0	0
6	-37.631770	-73.648425	195.0	0
7	-37.627438	-73.654343	191.0	0
8	-37.626963	-73.654958	189.0	0
9	-37.622963	-73.660566	152.0	0
10	-37.622470	-73.660635	149.0	0
11	-37.622470	-73.660635	149.0	0
12	-37.619640	-73.661331	123.0	0
13	-37.619640	-73.661331	123.0	0
14	-37.617720	-73.664638	114.0	0
15	-37.614780	-73.668103	101.0	0
16	-37.614780	-73.668103	101.0	0
17	-37.612898	-73.668035	94.0	0
18	-37.608973	-73.667430	63.0	0
19	-37.608973	-73.667430	63.0	1
20	-37.608146	-73.665760	48.0	0
21	-37.608038	-73.665080	42.0	0
22	-37.608038	-73.665080	42.0	0
23	-37.609146	-73.660695	8.0	0
24	-37.609146	-73.660695	8.0	1
25	-37.608973	-73.660643	7.0	0
26	-37.606087	-73.660220	6.0	0
27	-37.605386	-73.660188	6.0	0
28	-37.605386	-73.660188	6.0	0
29	-37.603115	-73.661298	6.0	0
30	-37.600790	-73.662213	5.0	0
31	-37.600708	-73.662196	5.0	0
32	-37.600708	-73.662196	5.0	0
33	-37.600673	-73.662120	5.0	0
34	-37.600690	-73.662022	5.0	0
35	-37.600793	-73.661865	3.0	1
36	-37.601748	-73.659605	2.0	0
37	-37.602180	-73.658358	1.0	0
38	-37.602318	-73.657881	1.0	0
39	-37.603265	-73.655666	2.0	0
40	-37.603265	-73.655666	2.0	0

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo las observaciones 2,19,24,35 son detectadas como outlier.

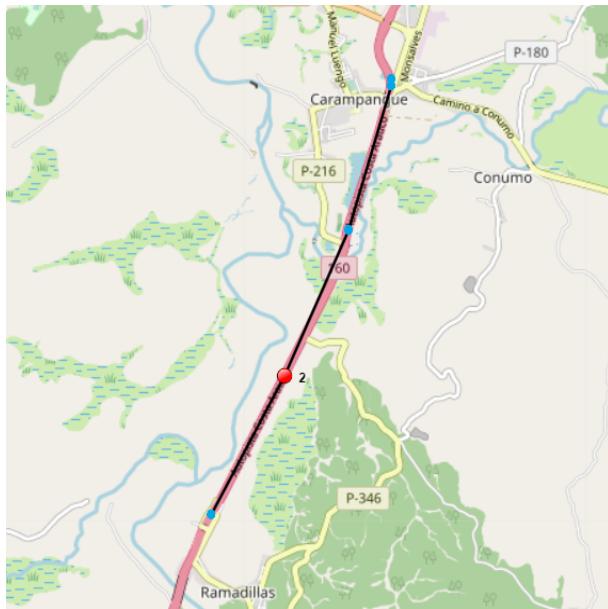


	latitud	longitud	altitud	outlier
1	-37.609085	-73.660723	11.0	0
2	-37.608995	-73.660728	11.0	1
3	-37.607501	-73.660615	8.0	0
4	-37.607046	-73.660580	9.0	0
5	-37.606046	-73.660168	10.0	0
6	-37.606046	-73.660168	10.0	0
7	-37.605853	-73.660010	10.0	0
8	-37.605853	-73.660010	10.0	0
9	-37.605812	-73.660076	10.0	0
10	-37.605812	-73.660076	10.0	0
11	-37.605832	-73.660171	10.0	0
12	-37.605832	-73.660171	10.0	0
13	-37.605936	-73.660216	9.0	0
14	-37.605936	-73.660216	9.0	0
15	-37.606360	-73.660351	9.0	0
16	-37.608053	-73.660725	8.0	0
17	-37.608053	-73.660725	8.0	0
18	-37.608071	-73.660841	8.0	0
19	-37.608110	-73.660872	8.0	1
20	-37.608110	-73.660872	8.0	0
21	-37.608420	-73.661066	9.0	0
22	-37.608420	-73.661066	9.0	0
23	-37.608407	-73.661058	9.0	0
24	-37.608392	-73.661050	9.0	1
25	-37.608346	-73.661021	9.0	0
26	-37.608346	-73.661021	9.0	0
27	-37.608346	-73.661021	9.0	0
28	-37.608346	-73.661021	9.0	0
29	-37.608346	-73.661021	9.0	0
30	-37.608346	-73.661021	9.0	0
31	-37.608346	-73.661021	9.0	0
32	-37.608346	-73.661021	9.0	0
33	-37.608346	-73.661021	9.0	0
34	-37.608346	-73.661021	9.0	0
35	-37.608395	-73.661058	8.0	1
36	-37.608395	-73.661058	8.0	0
37	-37.608586	-73.661110	9.0	0
38	-37.608710	-73.660843	9.0	0
39	-37.608737	-73.660750	9.0	0
40	-37.608737	-73.660750	9.0	0

4.11.5. Dataset 3(Autoencoder apilado)

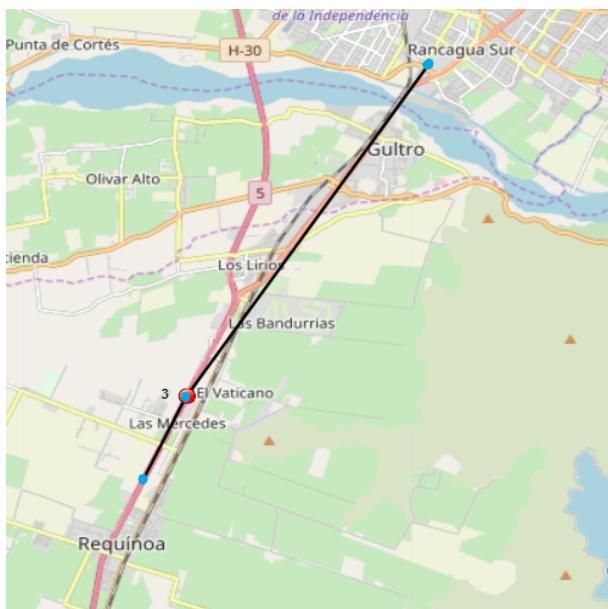
Ejemplos de trayectos outlier

Este trayecto muestra una continuidad normal, sin embargo la observación 2 muestra un cambio brusco de velocidad respecto a la anterior observación y luego regresa a la velocidad promedio.



	fecha	latitud	longitud	altura	velocidad	nsat	outlier
1	2018-01-18 13:56:45	-37.299645	-73.261120	2.0	131.0	9.0	0
2	2018-01-18 13:57:25	-37.285501	-73.251615	8.0	176.0	9.0	1
3	2018-01-18 13:58:05	-37.270796	-73.243498	6.0	133.0	9.0	0
4	2018-01-18 13:58:45	-37.256138	-73.237983	10.0	133.0	9.0	0
5	2018-01-18 13:58:47	-37.255475	-73.237985	10.0	132.0	9.0	0

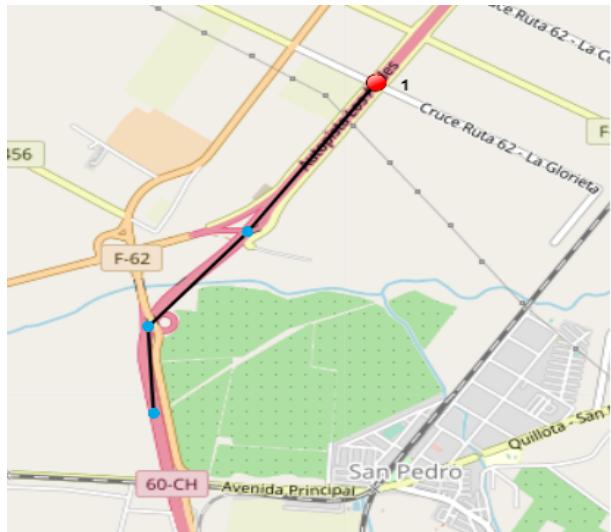
Este trayecto muestra una discontinuidad evidente, las observación 3 sale de la continuidad de las demás observaciones.



	fecha	latitud	longitud	altura	velocidad	nsat	outlier
1	2016-12-19 08:21:01	-34.185661	-70.748155	515.0	0.0	10.0	0
2	2016-12-19 08:24:20	-34.185821	-70.748408	502.0	2.0	9.0	0
3	2016-12-19 08:34:21	-34.254470	-70.802811	451.0	130.0	9.0	1
4	2016-12-19 08:34:22	-34.254767	-70.802979	451.0	130.0	9.0	0
5	2016-12-19 08:35:22	-34.271658	-70.812619	431.0	113.0	9.0	0

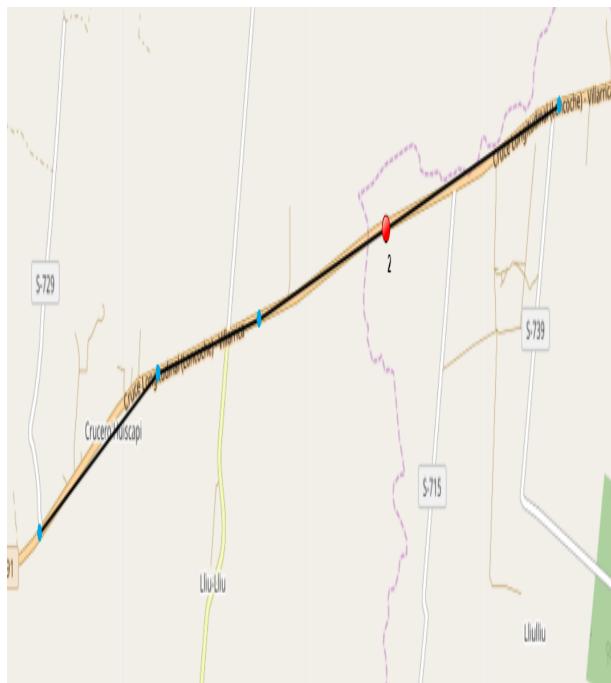
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo la observación 1 es detectada como outlier la cual mantiene una velocidad promedio sin cambios bruscos.



	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2018-02-01 08:59:54	-32.920185	-71.277839	91.0	167.0	9.0	1
2	2018-02-01 09:00:16	-32.927188	-71.283868	87.0	149.0	10.0	0
3	2018-02-01 09:00:34	-32.931720	-71.288510	82.0	114.0	10.0	0
4	2018-02-01 09:00:35	-32.931720	-71.288510	82.0	114.0	10.0	0
5	2018-02-01 09:01:15	-32.935803	-71.288214	82.0	62.0	7.0	0

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo la observación 2 es detectada como outlier la cual mantiene una velocidad promedio sin cambios bruscos.

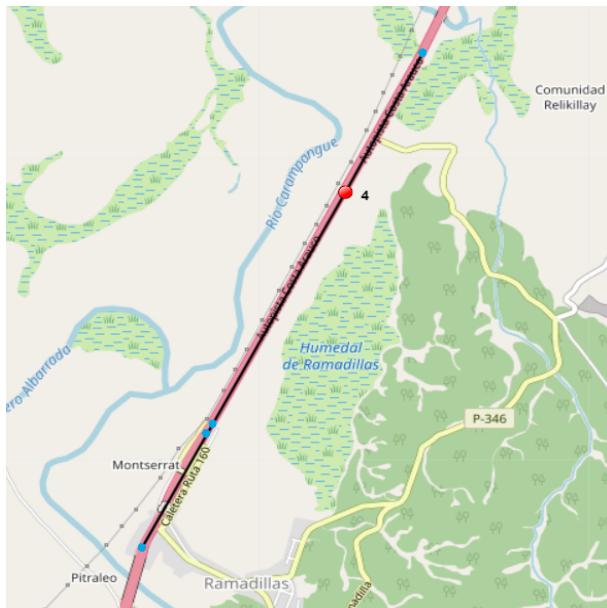


	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2017-09-01 21:52:08	-39.266591	-72.369378	258.0	157.0	10.0	0
2	2017-09-01 21:52:48	-39.269906	-72.390235	242.0	165.0	10.0	1
3	2017-09-01 21:53:28	-39.272368	-72.405483	259.0	107.0	10.0	0
4	2017-09-01 21:54:07	-39.273873	-72.417707	246.0	100.0	9.0	0
5	2017-09-01 21:54:48	-39.278154	-72.431950	241.0	128.0	10.0	0

4.11.6. Dataset 3(Autoencoder convolucional apilado)

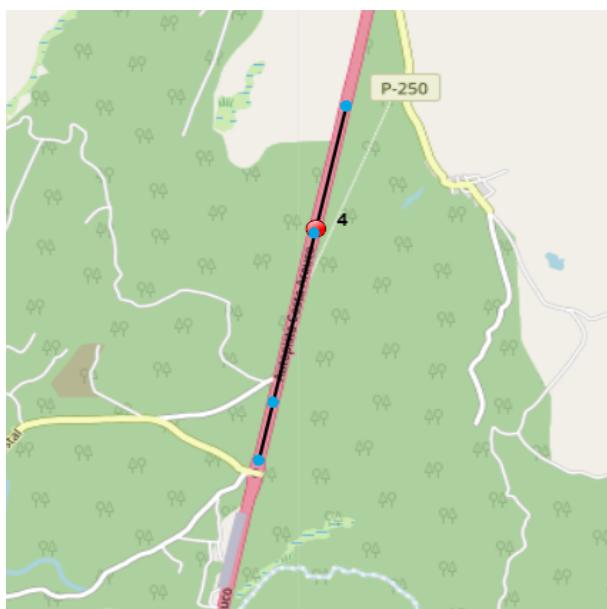
Ejemplos de trayectos outlier

Este trayecto muestra una continuidad normal, sin embargo la observación 4 muestra un cambio brusco de velocidad respecto a la anterior observación y luego regresa a la velocidad promedio.



	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2018-01-17 12:06:27	-37.305420	-73.264621	20.0	73.0	10.0	0
2	2018-01-17 12:07:07	-37.298798	-73.260396	15.0	129.0	9.0	0
3	2018-01-17 12:07:09	-37.298233	-73.259948	15.0	133.0	9.0	0
4	2018-01-17 12:07:47	-37.284762	-73.251110	14.0	173.0	9.0	1
5	2018-01-17 12:08:09	-37.276738	-73.245958	11.0	138.0	9.0	0

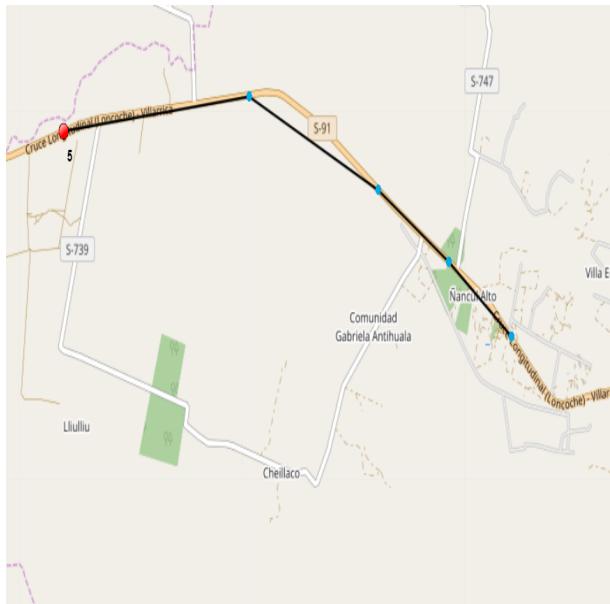
Este trayecto muestra una continuidad normal, sin embargo la observación 4 muestra un cambio brusco de velocidad con respecto a la siguiente observación.



	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2017-12-29 16:24:49	-37.416077	-73.348448	141.0	147.0	9.0	0
2	2017-12-29 16:25:03	-37.411095	-73.347082	139.0	148.0	9.0	0
3	2017-12-29 16:25:43	-37.396380	-73.343113	123.0	158.0	9.0	0
4	2017-12-29 16:25:44	-37.395995	-73.343007	123.0	157.0	9.0	1
5	2017-12-29 16:26:23	-37.385340	-73.340125	138.0	50.0	7.0	0

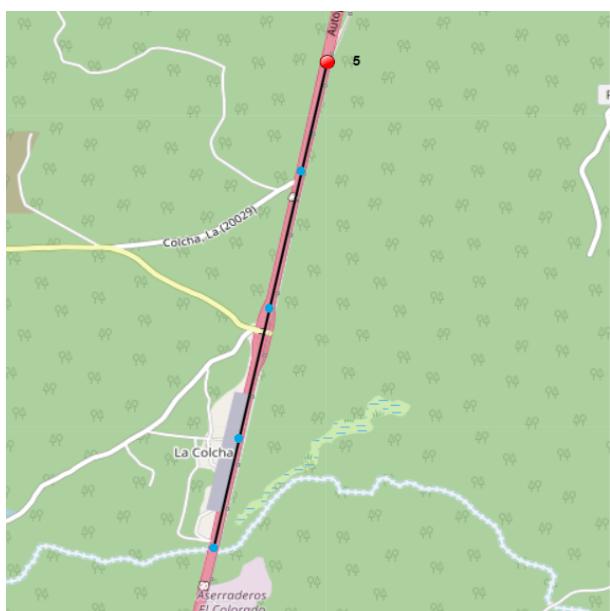
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, la observación 5 no tiene un cambio brusco de velocidad sin embargo es detectada como outlier.



	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2017-10-13 21:19:02	-39.276736	-72.329447	305.0	100.0	11.0	0
2	2017-10-13 21:19:42	-39.273240	-72.335630	300.0	50.0	10.0	0
3	2017-10-13 21:20:22	-39.269923	-72.342538	285.0	125.0	10.0	0
4	2017-10-13 21:21:02	-39.265593	-72.355268	268.0	88.0	10.0	0
5	2017-10-13 21:21:42	-39.267206	-72.373541	244.0	163.0	9.0	1

Este trayecto muestra una continuidad normal en todas las observaciones, la observación 5 no tiene un cambio brusco de velocidad sin embargo es detectada como outlier.

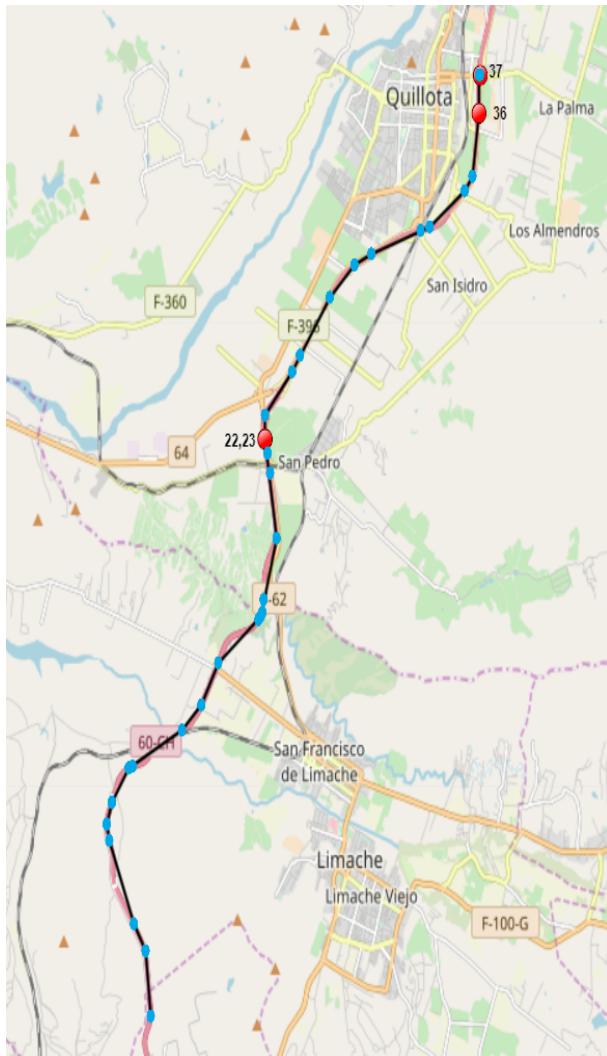


	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2018-01-24 17:24:19	-37.429343	-73.352057	146.0	133.0	10.0	0
2	2018-01-24 17:24:36	-37.423185	-73.350393	157.0	150.0	10.0	0
3	2018-01-24 17:24:59	-37.415925	-73.348421	153.0	111.0	8.0	0
4	2018-01-24 17:25:23	-37.408193	-73.346321	155.0	148.0	9.0	0
5	2018-01-24 17:25:39	-37.402070	-73.344676	146.0	159.0	8.0	1

4.11.7. Dataset 4(Autoencoder apilado)

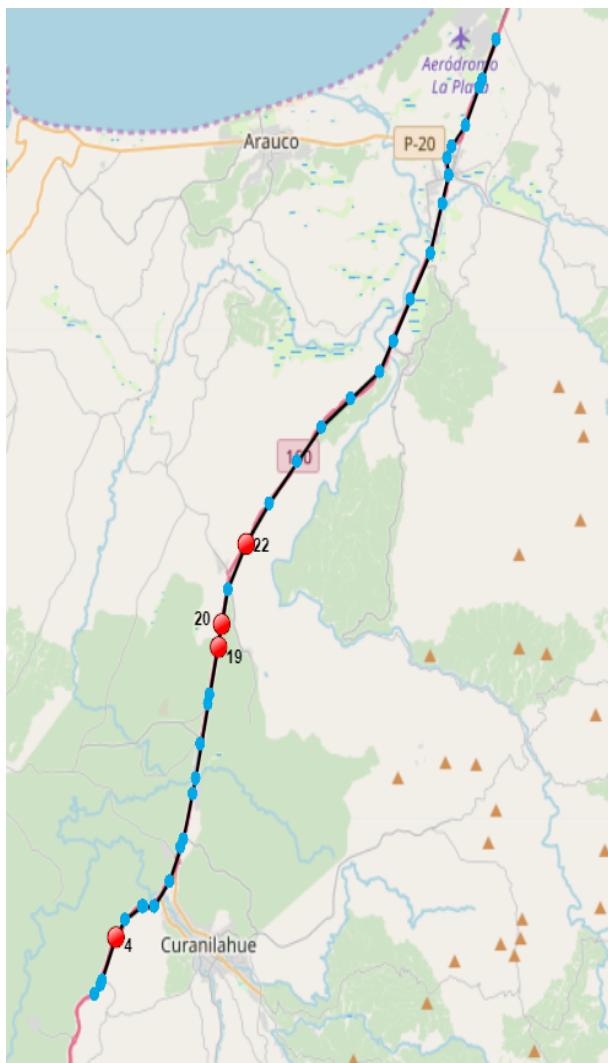
Ejemplos de trayectos outlier

Este trayecto muestra una continuidad normal, sin embargo las observaciones 22,23 muestran un cambio brusco de velocidad con respecto a las velocidades adyacentes, la observación 37 muestra un cambio brusco de velocidad con respecto a la observación 36.



	fecha	latitud	longitud	altura	velocidad	nsat	outlier
1	2018-02-01 08:33:04	-33.026723	-71.318473	124.0	137.0	8.0	0
2	2018-02-01 08:33:30	-33.016562	-71.319941	106.0	154.0	6.0	0
3	2018-02-01 08:33:31	-33.016562	-71.319941	106.0	154.0	6.0	0
4	2018-02-01 08:33:44	-33.012098	-71.323018	103.0	160.0	7.0	0
5	2018-02-01 08:34:24	-32.998755	-71.329436	92.0	140.0	9.0	0
6	2018-02-01 08:34:31	-32.996403	-71.330000	87.0	139.0	7.0	0
7	2018-02-01 08:34:41	-32.992712	-71.328748	75.0	142.0	7.0	0
8	2018-02-01 08:34:42	-32.992712	-71.328748	75.0	142.0	7.0	0
9	2018-02-01 08:35:01	-32.997310	-71.324196	70.0	139.0	5.0	0
10	2018-02-01 08:35:02	-32.997310	-71.324196	70.0	139.0	5.0	0
11	2018-02-01 08:35:04	-32.997095	-71.323411	71.0	138.0	5.0	0
12	2018-02-01 08:35:44	-32.981237	-71.310036	62.0	132.0	6.0	0
13	2018-02-01 08:36:02	-32.977417	-71.304978	68.0	132.0	6.0	0
14	2018-02-01 08:36:24	-32.970692	-71.300534	84.0	143.0	9.0	0
15	2018-02-01 08:37:02	-32.963713	-71.289811	145.0	118.0	6.0	0
16	2018-02-01 08:37:03	-32.963229	-71.289395	147.0	120.0	5.0	0
17	2018-02-01 08:37:05	-32.962680	-71.289053	150.0	123.0	7.0	0
18	2018-02-01 08:37:13	-32.960498	-71.288483	154.0	130.0	6.0	0
19	2018-02-01 08:37:44	-32.950663	-71.285293	103.0	138.0	9.0	0
20	2018-02-01 08:38:13	-32.940502	-71.288960	86.0	138.0	6.0	0
21	2018-02-01 08:38:24	-32.937323	-71.287701	87.0	95.0	8.0	0
22	2018-02-01 08:39:04	-32.934893	-71.288131	83.0	0.0	9.0	1
23	2018-02-01 08:39:13	-32.934893	-71.288131	83.0	0.0	9.0	1
24	2018-02-01 08:39:44	-32.931070	-71.288090	81.0	107.0	8.0	0
25	2018-02-01 08:40:13	-32.924160	-71.281118	85.0	124.0	9.0	0
26	2018-02-01 08:40:24	-32.921690	-71.278988	87.0	119.0	9.0	0
27	2018-02-01 08:41:04	-32.912407	-71.271014	93.0	115.0	9.0	0
28	2018-02-01 08:41:28	-32.907371	-71.264666	99.0	117.0	6.0	0
29	2018-02-01 08:41:30	-32.907371	-71.264666	99.0	117.0	6.0	0
30	2018-02-01 08:41:44	-32.905650	-71.259995	100.0	123.0	8.0	0
31	2018-02-01 08:42:24	-32.901667	-71.247066	117.0	107.0	7.0	0
32	2018-02-01 08:42:30	-32.901327	-71.244768	117.0	114.0	7.0	0
33	2018-02-01 08:43:04	-32.895404	-71.235221	121.0	115.0	8.0	0
34	2018-02-01 08:43:13	-32.893170	-71.233126	119.0	113.0	9.0	0
35	2018-02-01 08:43:14	-32.893170	-71.233126	119.0	113.0	9.0	0
36	2018-02-01 08:43:44	-32.883116	-71.231621	123.0	151.0	8.0	1
37	2018-02-01 08:44:14	-32.876997	-71.231075	130.0	33.0	8.0	1
38	2018-02-01 08:44:15	-32.876920	-71.231116	130.0	33.0	7.0	0
39	2018-02-01 08:44:17	-32.876784	-71.231376	130.0	36.0	7.0	0
40	2018-02-01 08:44:20	-32.876943	-71.231746	130.0	46.0	9.0	0

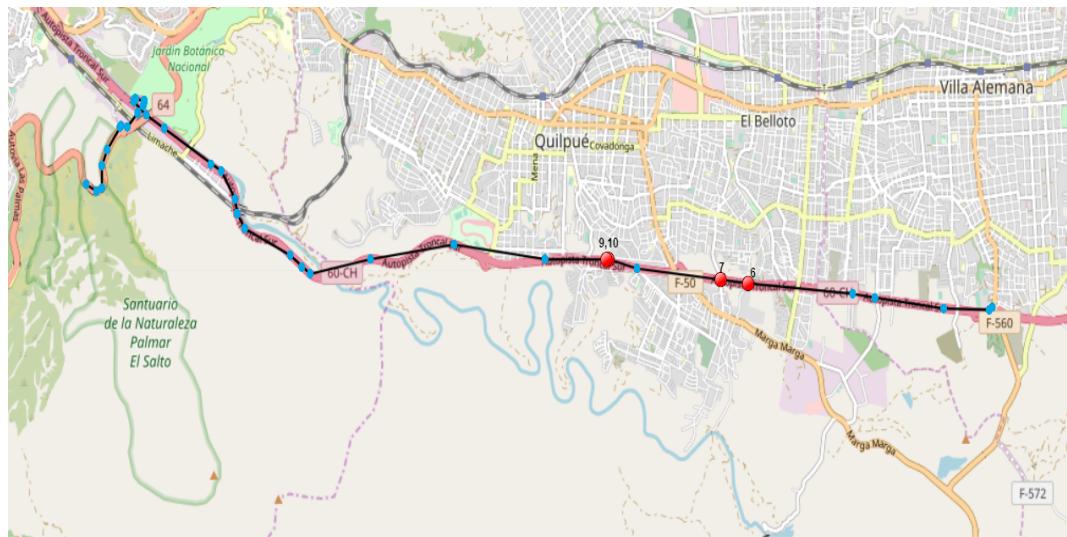
Este trayecto muestra una continuidad normal, sin embargo entre las observaciones 19 y 20 se muestra un cambio brusco de velocidad y las observaciones 4 y 22 no muestran un cambio brusco de velocidad.



	fecha	latitud	longitud	altura	velocidad	nsat	outlier
1	2017-12-30 13:05:24	-37.489888	-73.396593	178.0	102.0	7.0	0
2	2017-12-30 13:05:59	-37.487258	-73.394191	180.0	80.0	9.0	0
3	2017-12-30 13:06:04	-37.486372	-73.393762	178.0	84.0	6.0	0
4	2017-12-30 13:06:44	-37.473855	-73.387041	170.0	161.0	9.0	1
5	2017-12-30 13:06:59	-37.469131	-73.383172	189.0	140.0	6.0	0
6	2017-12-30 13:07:20	-37.465036	-73.375870	183.0	129.0	7.0	0
7	2017-12-30 13:07:21	-37.465036	-73.375870	183.0	129.0	7.0	0
8	2017-12-30 13:07:24	-37.465148	-73.374630	187.0	134.0	7.0	0
9	2017-12-30 13:07:35	-37.464790	-73.370028	178.0	128.0	8.0	0
10	2017-12-30 13:07:36	-37.464790	-73.370028	178.0	128.0	8.0	0
11	2017-12-30 13:08:04	-37.457713	-73.363093	125.0	141.0	6.0	0
12	2017-12-30 13:08:36	-37.448400	-73.358198	161.0	130.0	9.0	0
13	2017-12-30 13:08:44	-37.445990	-73.357156	161.0	125.0	10.0	0
14	2017-12-30 13:09:24	-37.433118	-73.353021	140.0	151.0	10.0	0
15	2017-12-30 13:09:36	-37.428381	-73.351755	138.0	163.0	10.0	0
16	2017-12-30 13:10:04	-37.418948	-73.349205	140.0	139.0	10.0	0
17	2017-12-30 13:10:36	-37.407373	-73.346088	145.0	137.0	10.0	0
18	2017-12-30 13:10:44	-37.404656	-73.345360	137.0	146.0	9.0	0
19	2017-12-30 13:11:24	-37.390788	-73.341608	138.0	150.0	9.0	1
20	2017-12-30 13:12:04	-37.384083	-73.339738	143.0	5.0	7.0	1
21	2017-12-30 13:12:44	-37.374742	-73.337310	141.0	143.0	5.0	0
22	2017-12-30 13:13:24	-37.361657	-73.329252	131.0	167.0	11.0	1
23	2017-12-30 13:14:04	-37.349851	-73.318318	114.0	131.0	9.0	0
24	2017-12-30 13:14:44	-37.338195	-73.306443	104.0	133.0	11.0	0
25	2017-12-30 13:15:24	-37.327878	-73.295120	62.0	133.0	9.0	0
26	2017-12-30 13:16:03	-37.320028	-73.282316	92.0	138.0	10.0	0
27	2017-12-30 13:16:44	-37.312497	-73.269098	47.0	121.0	7.0	0
28	2017-12-30 13:17:24	-37.303488	-73.263301	10.0	103.0	11.0	0
29	2017-12-30 13:18:04	-37.291648	-73.255621	9.0	155.0	11.0	0
30	2017-12-30 13:18:44	-37.278225	-73.246873	8.0	160.0	8.0	0
31	2017-12-30 13:19:24	-37.264485	-73.240948	8.0	146.0	11.0	0
32	2017-12-30 13:19:49	-37.255772	-73.237981	15.0	125.0	11.0	0
33	2017-12-30 13:19:50	-37.255772	-73.237981	15.0	125.0	11.0	0
34	2017-12-30 13:20:04	-37.251351	-73.239313	10.0	137.0	11.0	0
35	2017-12-30 13:20:16	-37.247553	-73.237036	10.0	123.0	11.0	0
36	2017-12-30 13:20:17	-37.247553	-73.237036	10.0	123.0	11.0	0
37	2017-12-30 13:20:44	-37.241936	-73.230953	9.0	131.0	11.0	0
38	2017-12-30 13:21:17	-37.230905	-73.224463	8.0	155.0	11.0	0
39	2017-12-30 13:21:24	-37.228541	-73.223030	7.0	140.0	11.0	0
40	2017-12-30 13:22:04	-37.217025	-73.216833	11.0	110.0	12.0	0

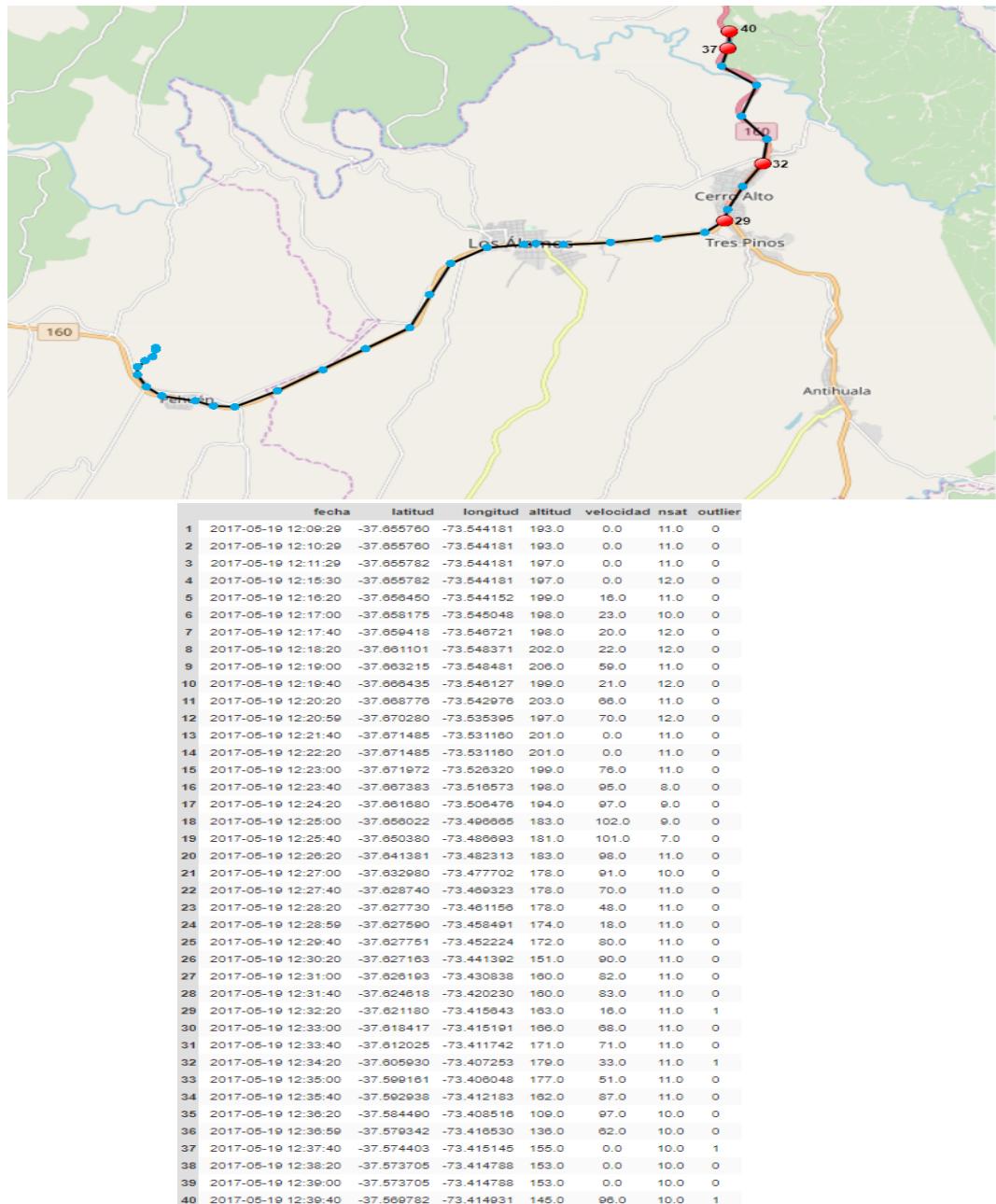
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, las observaciones 6, 7, 9 y 10 no presentan un cambio brusco en la velocidad, sin embargo son detectados como outlier.



	fecha	latitud	longitud	altura	velocidad	nsat	outlier
1	2018-01-22 14:45:57	-33.067990	-71.371581	171.0	53.0	10.0	0
2	2018-01-22 14:46:01	-33.068256	-71.372160	170.0	57.0	7.0	0
3	2018-01-22 14:46:31	-33.068035	-71.379479	183.0	111.0	6.0	0
4	2018-01-22 14:47:01	-33.067011	-71.390623	175.0	121.0	10.0	0
5	2018-01-22 14:47:11	-33.066468	-71.394259	177.0	125.0	10.0	0
6	2018-01-22 14:47:51	-33.065397	-71.411271	162.0	154.0	10.0	1
7	2018-01-22 14:48:01	-33.064933	-71.415800	169.0	154.0	10.0	1
8	2018-01-22 14:48:31	-33.063733	-71.429321	166.0	112.0	8.0	0
9	2018-01-22 14:49:01	-33.062845	-71.433879	165.0	20.0	7.0	1
10	2018-01-22 14:49:11	-33.062808	-71.434275	164.0	19.0	6.0	1
11	2018-01-22 14:49:51	-33.062838	-71.444221	173.0	120.0	9.0	0
12	2018-01-22 14:50:31	-33.061178	-71.458979	168.0	133.0	9.0	0
13	2018-01-22 14:51:11	-33.062708	-71.472619	103.0	117.0	10.0	0
14	2018-01-22 14:51:41	-33.064338	-71.492448	59.0	124.0	7.0	0
15	2018-01-22 14:51:45	-33.063573	-71.493709	56.0	133.0	9.0	0
16	2018-01-22 14:51:51	-33.062266	-71.485609	53.0	132.0	8.0	0
17	2018-01-22 14:52:17	-33.059461	-71.493021	33.0	94.0	8.0	0
18	2018-01-22 14:52:25	-33.057855	-71.494271	25.0	99.0	7.0	0
19	2018-01-22 14:52:31	-33.056330	-71.494539	23.0	106.0	7.0	0
20	2018-01-22 14:52:44	-33.053318	-71.496808	18.0	125.0	9.0	0
21	2018-01-22 14:52:49	-33.052621	-71.498564	18.0	133.0	9.0	0
22	2018-01-22 14:53:11	-33.049697	-71.505989	16.0	133.0	10.0	0
23	2018-01-22 14:53:21	-33.047168	-71.508918	16.0	95.0	9.0	0
24	2018-01-22 14:53:34	-33.045820	-71.511053	20.0	38.0	5.0	0
25	2018-01-22 14:53:37	-33.045580	-71.510949	20.0	34.0	6.0	0
26	2018-01-22 14:53:40	-33.045481	-71.510653	19.0	35.0	7.0	0
27	2018-01-22 14:53:50	-33.045650	-71.509403	20.0	40.0	7.0	0
28	2018-01-22 14:53:51	-33.045740	-71.509341	20.0	40.0	6.0	0
29	2018-01-22 14:53:52	-33.045845	-71.509318	20.0	42.0	6.0	0
30	2018-01-22 14:53:52	-33.046168	-71.509505	19.0	50.0	8.0	0
31	2018-01-22 14:54:02	-33.046992	-71.510291	17.0	64.0	8.0	0
32	2018-01-22 14:54:03	-33.047125	-71.510414	17.0	66.0	8.0	0
33	2018-01-22 14:54:23	-33.048501	-71.512091	33.0	40.0	10.0	0
34	2018-01-22 14:54:30	-33.048450	-71.513154	41.0	43.0	10.0	0
35	2018-01-22 14:54:32	-33.048517	-71.513256	42.0	42.0	10.0	0
36	2018-01-22 14:55:11	-33.050928	-71.515321	65.0	64.0	7.0	0
37	2018-01-22 14:55:22	-33.052770	-71.516176	78.0	76.0	9.0	0
38	2018-01-22 14:55:34	-33.056077	-71.516188	86.0	79.0	7.0	0
39	2018-01-22 14:55:38	-33.055395	-71.517074	90.0	82.0	8.0	0
40	2018-01-22 14:55:46	-33.054648	-71.518798	104.0	80.0	6.0	0

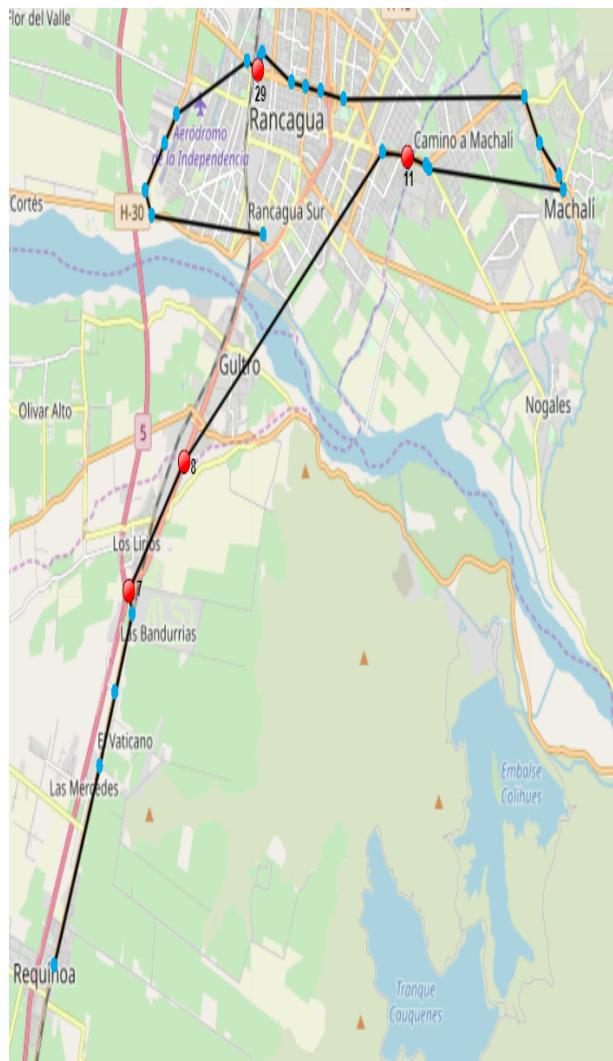
Este trayecto muestra una continuidad normal en todas las observaciones, las observaciones 28, 32, 37 y 40 no presentan un cambio brusco en la velocidad, sin embargo son detectados como outlier.



4.11.8. Dataset 4(Autoencoder convolucional apilado)

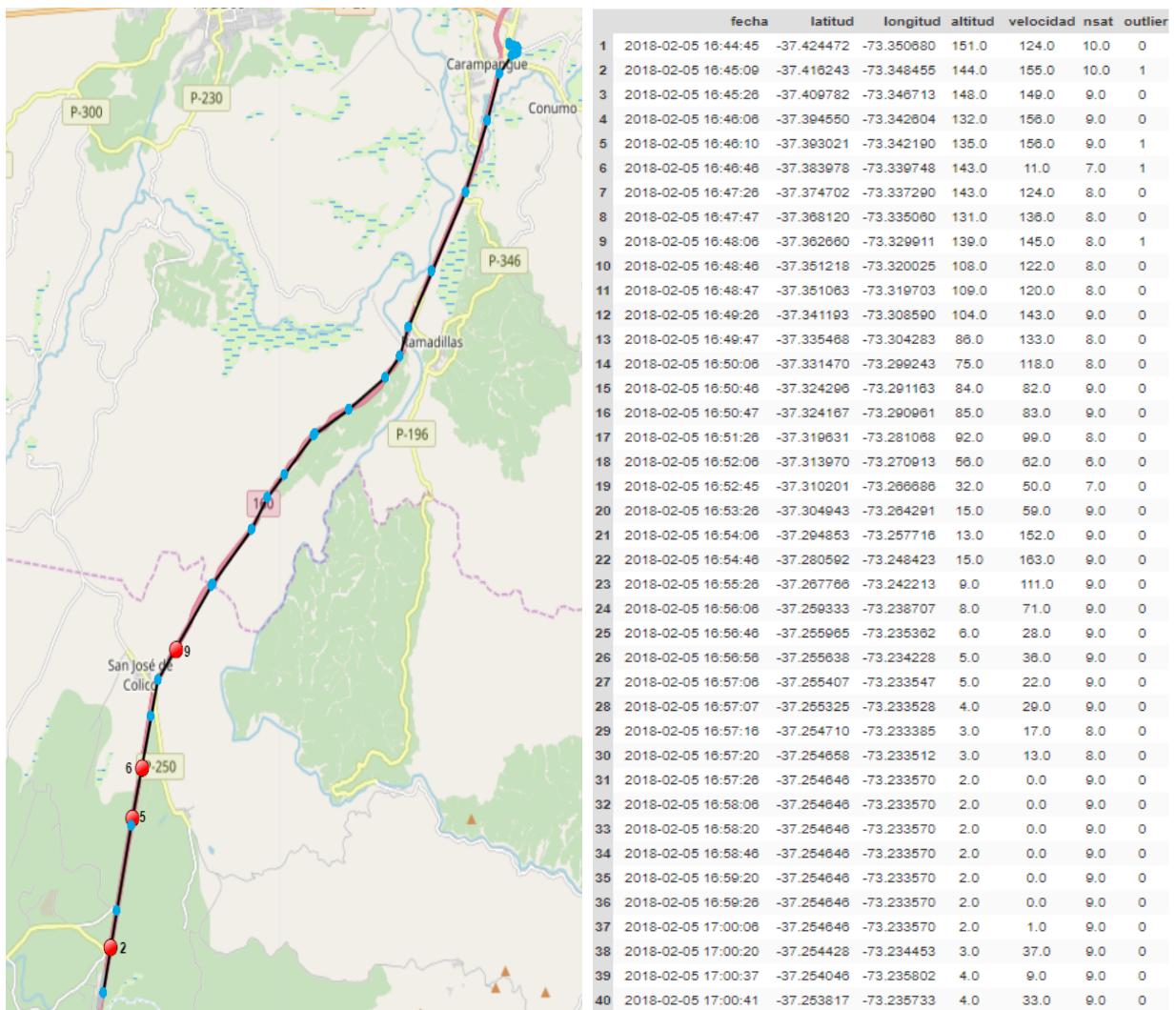
Ejemplos de trayectos outlier

Este trayecto muestra una discontinuidad evidente en las observaciones 7 y 8 con respecto a la siguiente observación, mientras que las observaciones 11 y 29 no presenten cambios bruscos de velocidad.



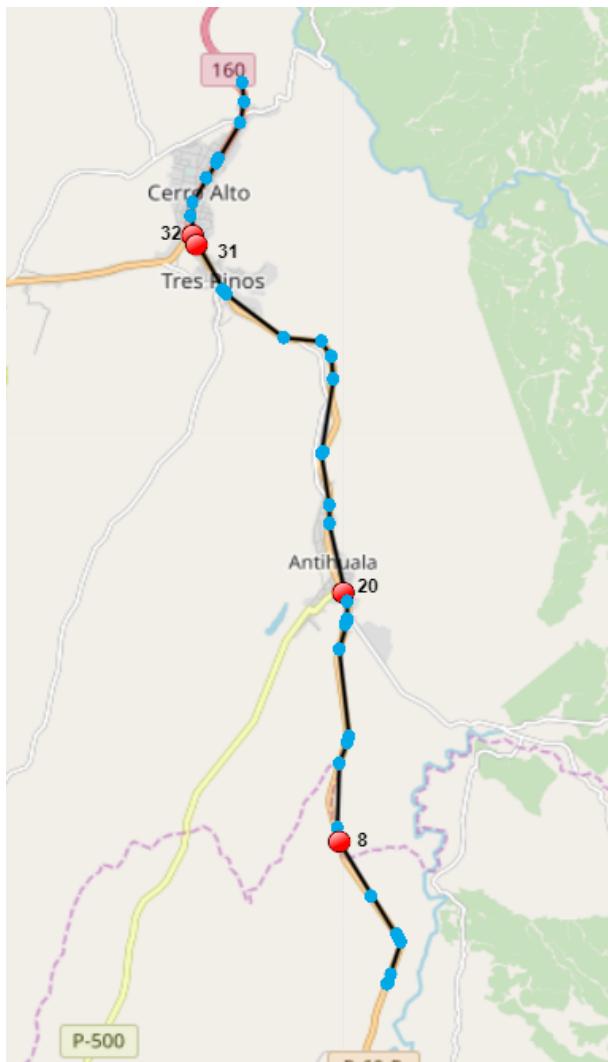
	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2017-06-10 10:38:51	-34.283958	-70.814451	416.0	0.0	9.0	0
2	2017-06-10 10:39:59	-34.283958	-70.814451	416.0	0.0	9.0	0
3	2017-06-10 10:49:59	-34.257215	-70.800098	451.0	18.0	12.0	0
4	2017-06-10 10:50:00	-34.257215	-70.800098	451.0	18.0	12.0	0
5	2017-06-10 10:51:00	-34.247308	-70.795058	463.0	75.0	12.0	0
6	2017-06-10 10:52:00	-34.236688	-70.789648	474.0	71.0	12.0	0
7	2017-06-10 10:53:00	-34.233773	-70.790779	474.0	12.0	10.0	1
8	2017-06-10 10:54:39	-34.216420	-70.773241	499.0	112.0	10.0	1
9	2017-06-10 11:04:41	-34.174881	-70.710411	522.0	0.0	10.0	0
10	2017-06-10 11:04:42	-34.174881	-70.710411	522.0	0.0	10.0	0
11	2017-06-10 11:05:42	-34.175576	-70.702669	525.0	65.0	10.0	1
12	2017-06-10 11:06:42	-34.176490	-70.696663	529.0	4.0	10.0	0
13	2017-06-10 11:07:42	-34.177089	-70.695805	532.0	0.0	10.0	0
14	2017-06-10 11:17:43	-34.179976	-70.653194	560.0	0.0	9.0	0
15	2017-06-10 11:17:44	-34.179976	-70.653194	560.0	0.0	9.0	0
16	2017-06-10 11:18:44	-34.178141	-70.654573	555.0	47.0	9.0	0
17	2017-06-10 11:19:44	-34.173668	-70.660608	544.0	45.0	9.0	0
18	2017-06-10 11:20:44	-34.187555	-70.665563	536.0	58.0	9.0	0
19	2017-06-10 11:30:44	-34.187885	-70.722798	510.0	0.0	9.0	0
20	2017-06-10 11:30:44	-34.187885	-70.722798	510.0	0.0	9.0	0
21	2017-06-10 11:31:45	-34.186703	-70.729918	503.0	31.0	8.0	0
22	2017-06-10 11:32:45	-34.185998	-70.734808	503.0	34.0	9.0	0
23	2017-06-10 11:33:45	-34.185381	-70.739105	503.0	13.0	8.0	0
24	2017-06-10 11:42:45	-34.181478	-70.748419	489.0	0.0	8.0	0
25	2017-06-10 11:42:46	-34.181478	-70.748419	489.0	0.0	8.0	0
26	2017-06-10 11:43:46	-34.181478	-70.748419	489.0	0.0	8.0	0
27	2017-06-10 11:44:46	-34.182241	-70.749092	489.0	37.0	8.0	0
28	2017-06-10 11:45:46	-34.183842	-70.749463	489.0	0.0	8.0	0
29	2017-06-10 11:46:54	-34.183901	-70.750008	489.0	0.0	9.0	1
30	2017-06-10 11:47:26	-34.182801	-70.753333	487.0	57.0	9.0	0
31	2017-06-10 11:57:26	-34.169770	-70.775724	488.0	15.0	9.0	0
32	2017-06-10 11:57:26	-34.169800	-70.775751	488.0	14.0	9.0	0
33	2017-06-10 11:58:27	-34.173778	-70.779492	485.0	42.0	8.0	0
34	2017-06-10 11:59:27	-34.180088	-70.785468	484.0	56.0	7.0	0
35	2017-06-10 12:00:27	-34.183523	-70.783704	484.0	61.0	5.0	0
36	2017-06-10 12:10:27	-34.185828	-70.748103	511.0	0.0	7.0	0
37	2017-06-10 12:10:28	-34.185828	-70.748103	511.0	0.0	7.0	0
38	2017-06-10 12:11:28	-34.185959	-70.748054	503.0	0.0	8.0	0
39	2017-06-10 12:12:28	-34.185875	-70.748054	499.0	0.0	7.0	0
40	2017-06-10 12:13:28	-34.185875	-70.748071	496.0	0.0	8.0	0

Este trayecto muestra una continuidad normal, sin embargo las observaciones 5 y 6 muestran un cambio brusco de velocidad, las observaciones 2 y 9 no muestran un cambio brusco de velocidad.



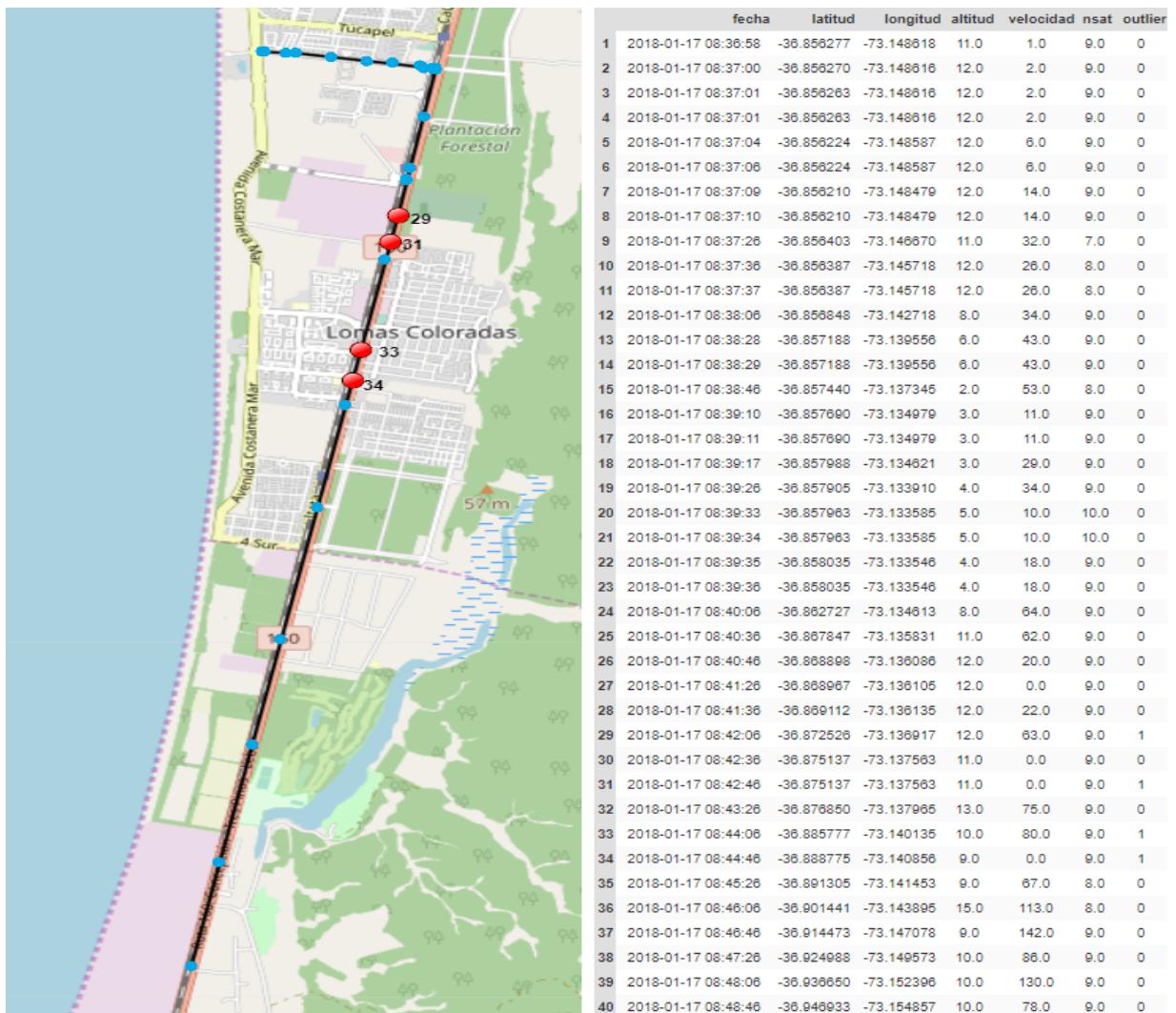
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, las observaciones 8, 20, 31 y 32 no presentan un cambio brusco en la velocidad, sin embargo son detectados como outlier.



	fecha	latitud	longitud	altura	velocidad	nsat	outlier
1	2018-01-30 15:40:56	-37.727798	-73.380995	100.0	78.0	8.0	0
2	2018-01-30 15:40:58	-37.727423	-73.380823	101.0	79.0	7.0	0
3	2018-01-30 15:41:03	-37.726350	-73.380321	102.0	75.0	8.0	0
4	2018-01-30 15:41:30	-37.721950	-73.378715	105.0	70.0	7.0	0
5	2018-01-30 15:41:32	-37.721398	-73.378893	105.0	80.0	7.0	0
6	2018-01-30 15:41:38	-37.720800	-73.379273	103.0	90.0	7.0	0
7	2018-01-30 15:42:16	-37.715267	-73.383961	103.0	66.0	8.0	0
8	2018-01-30 15:42:50	-37.707345	-73.389325	118.0	140.0	6.0	1
9	2018-01-30 15:42:56	-37.705605	-73.389873	119.0	142.0	5.0	0
10	2018-01-30 15:43:22	-37.698400	-73.389210	129.0	142.0	5.0	0
11	2018-01-30 15:43:32	-37.693350	-73.388026	133.0	111.0	7.0	0
12	2018-01-30 15:43:38	-37.692585	-73.387803	134.0	103.0	6.0	0
13	2018-01-30 15:44:16	-37.679970	-73.389198	132.0	125.0	7.0	0
14	2018-01-30 15:44:32	-37.676463	-73.388185	142.0	44.0	7.0	0
15	2018-01-30 15:44:56	-37.675973	-73.388040	137.0	0.0	7.0	0
16	2018-01-30 15:45:36	-37.675973	-73.388040	137.0	0.0	8.0	0
17	2018-01-30 15:46:16	-37.675973	-73.388040	137.0	0.0	8.0	0
18	2018-01-30 15:46:23	-37.675760	-73.387990	138.0	17.0	7.0	0
19	2018-01-30 15:46:44	-37.673096	-73.387958	135.0	73.0	7.0	0
20	2018-01-30 15:46:56	-37.671957	-73.388588	138.0	30.0	6.0	1
21	2018-01-30 15:47:36	-37.662172	-73.391211	156.0	116.0	7.0	0
22	2018-01-30 15:47:44	-37.659486	-73.391088	180.0	118.0	7.0	0
23	2018-01-30 15:48:14	-37.652107	-73.392388	185.0	117.0	7.0	0
24	2018-01-30 15:48:15	-37.651833	-73.392228	186.0	120.0	6.0	0
25	2018-01-30 15:48:56	-37.641300	-73.390418	174.0	75.0	6.0	0
26	2018-01-30 15:49:07	-37.638243	-73.390783	166.0	118.0	6.0	0
27	2018-01-30 15:49:18	-37.636016	-73.392620	164.0	93.0	6.0	0
28	2018-01-30 15:49:36	-37.635358	-73.398993	161.0	112.0	5.0	0
29	2018-01-30 15:50:16	-37.629056	-73.408993	152.0	87.0	8.0	0
30	2018-01-30 15:50:18	-37.628623	-73.406568	153.0	82.0	8.0	0
31	2018-01-30 15:50:56	-37.622000	-73.413968	148.0	100.0	6.0	1
32	2018-01-30 15:51:05	-37.620833	-73.414868	155.0	12.0	6.0	1
33	2018-01-30 15:51:36	-37.618135	-73.415182	158.0	61.0	8.0	0
34	2018-01-30 15:51:47	-37.616225	-73.414731	160.0	64.0	8.0	0
35	2018-01-30 15:52:16	-37.612873	-73.412235	170.0	55.0	8.0	0
36	2018-01-30 15:52:48	-37.610503	-73.410651	169.0	28.0	8.0	0
37	2018-01-30 15:52:56	-37.609837	-73.410161	171.0	48.0	8.0	0
38	2018-01-30 15:53:36	-37.604683	-73.406446	180.0	85.0	8.0	0
39	2018-01-30 15:53:48	-37.601740	-73.405683	178.0	100.0	8.0	0
40	2018-01-30 15:54:04	-37.599047	-73.406096	176.0	40.0	8.0	0

Este trayecto muestra una continuidad normal en todas las observaciones, las observación 29, 31, 33 y 34 no presenten un cambio brusco en la velocidad, sin embargo son detectados como outlier.



4.12. Arquitectura del mejor modelo

En base a los resultados cualitativos, el mejor modelo para la detección de series de tiempo outlier es el autoencoder apilado usando todas las variables en trayectos largos de 40 observaciones. Se muestra en detalle la arquitectura de este modelo en la figura 4.46.

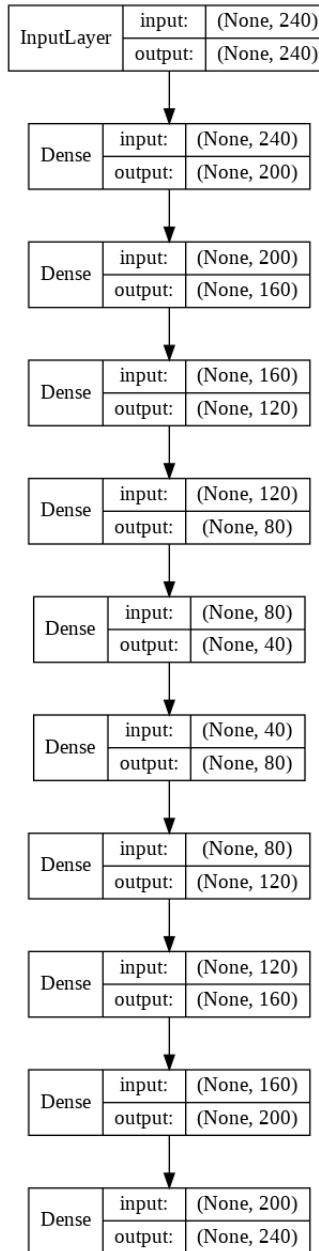


Figura 4.46: Arquitectura de autoencoder apilado

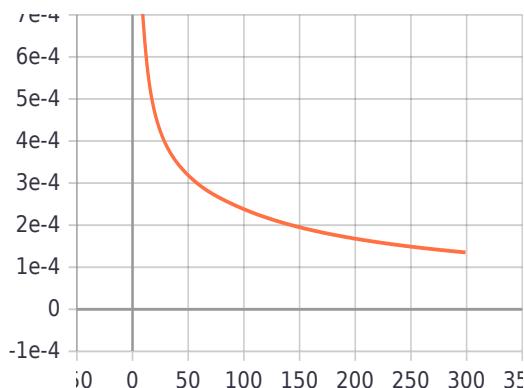
En el apéndice C se ilustra todas las arquitecturas de las redes neuronales autoencoder apilado y autoencoder convolucional apilado que se desarrollaron.

4.13. Métricas de aprendizaje y validación del mejor modelo

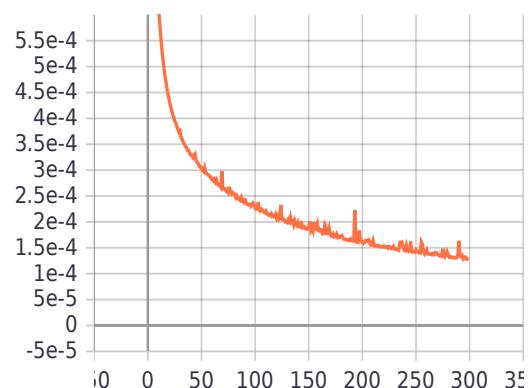
La métrica de aprendizaje y la de validación son descritas en la sección ?? y 3.9.1 respectivamente, a continuación se muestra la métrica de aprendizaje y validación del mejor modelo.

4.13.1. Métrica de función de costo

En la figura 4.47a se observa que la función de costo converge paulatinamente hasta la iteración 300. En la figura 4.47b se observa que la función de costo converge paulatinamente hasta la iteración 300.



(a) Función de costo en entrenamiento

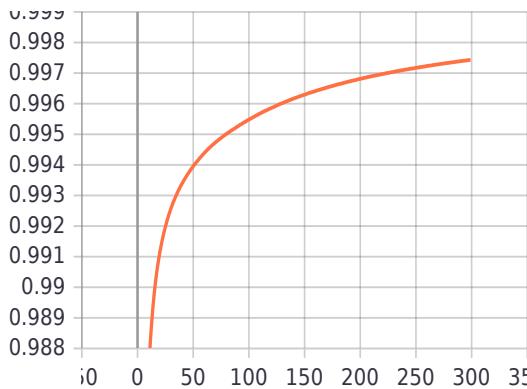


(b) Función de costo en validación

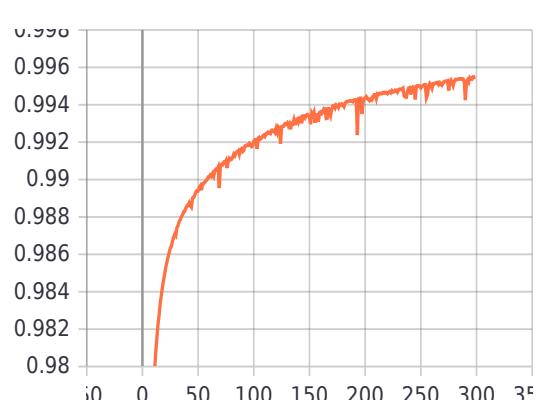
Figura 4.47: Función de costo error cuadrático medio

4.13.2. Métrica de coeficiente de determinación

En la figura 4.48a se observa que el coeficiente de determinación sube gradualmente hasta la iteración 300. En la figura 4.48b se observa que el coeficiente de determinación sube gradualmente hasta la iteración 300.



(a) Coeficiente de determinación en entrenamiento

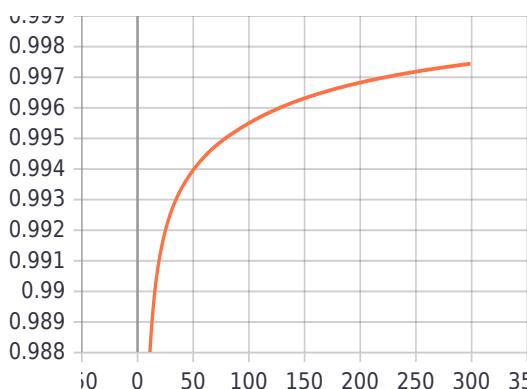


(b) Coeficiente de determinación en validación

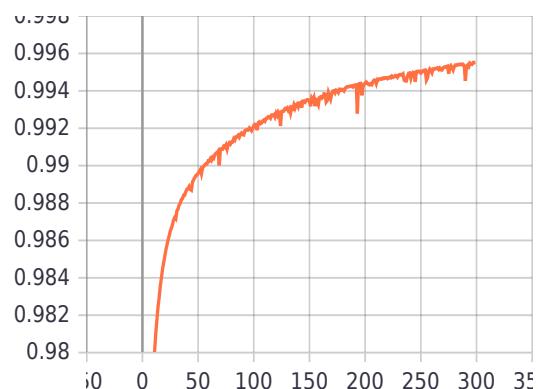
Figura 4.48: Métrica coeficiente de determinación

4.13.3. Métrica de varianza explicada

En la figura 4.48a se observa que la varianza explicada sube gradualmente hasta la iteración 30. En la figura 4.48b se observa que la varianza explicada sube gradualmente hasta la iteración 300, desde esta iteración la puntuación de la variación explicada realizado por el modelo permanece mas cerca al valor específico 0.998.



(a) Varianza explicada en entrenamiento



(b) Varianza explicada en validación

Figura 4.49: Métrica varianza explicada

En el apéndice D se ilustra todas las métricas de aprendizaje y validación de las redes neuronales autoencoder apilado y autoencoder convolucional apilado que se desarrollaron.

4.14. Criterio para la detección de series de tiempo outlier

El criterio para la detección de series de tiempo es definido en la sección 3.8, se explico que se tenia que establecer un umbral para discernir de series de tiempo outlier y normales, para ello nos basamos en la suposición de que las series de tiempo outlier se encuentran en minoría. A través de un histograma y con los errores de reconstrucción de todas las series de tiempo previamente calculadas, se gráfica una representación de distribuciones de frecuencias de los errores de reconstrucción. De esta forma se puede establecer un umbral, en la figura 4.50 se muestra como se fijo el umbral τ para el autoencoder apilado aplicado al data set 4, ya que esta configuración obtuvo el mejor resultado, dando como umbral $\tau = 0,0004$.

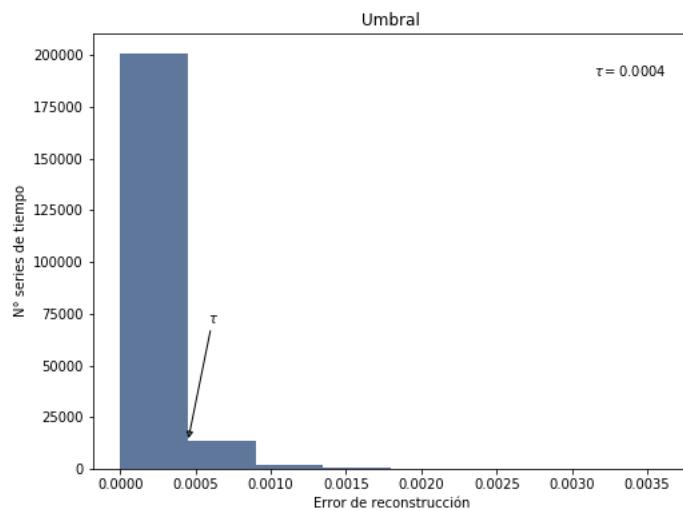


Figura 4.50: Umbral para la detección de series de tiempo outlier

4.15. Resultado del árbol de decisión

Como se menciono en la sección 2.11 los algoritmos de inteligencia artificial carecen de la capacidad de explicación. Para poder entender las decisiones de la red neuronal se uso un árbol de decisión, el cual nos genera una regla de decisiones que nos ayuda a entender las decisiones de la red neuronal. Como el árbol de decisión tiene un aprendizaje supervisado, necesitamos del etiquetado de los datos de entrenamiento y validación, para ello utilizaremos el umbral calculado en la sección 4.14, donde aquellas series de tiempo mayores al umbral son etiquetados como outlier, mientras que las series de tiempo menores al umbral son etiquetados como normales.

El árbol de decisión se implemento en python 3.7 utilizando la librería Scikit-Learn, dentro de la configuración del árbol de decisión se uso los parámetros por defecto, excepto la máxima profundidad, donde se tomo como máximo 20 niveles de profundidad, el árbol de decisión obtuvo una precisión de 92 %, el árbol completo se encuentra en el archivo time_series_tree.pdf dentro del directorio raíz del proyecto.

4.15.1. Explicación del árbol de decisión

La entrada del árbol de decisión es una serie de tiempo, la cual esta formado por 40 observaciones, para denotar las observaciones y las variables de cada observación se usara la siguiente expresión: fecha_x, lat_x, lon_x, alt_x, vel_x, nsat_x, donde lat es la variable latitud, lon en la variable longitud, alt es la variable altitud, vel es la variable velocidad y nsat es el número de satélites; y x que denota el numero de observación que va desde 1 hasta 40. Para explicar el árbol de decisión, se tomo una 1 hoja pura con decisión de serie de tiempo outlier y otra hoja pura con decisión de serie de tiempo normal.

Ejemplo outlier

En la tabla 4.13 se muestra las reglas de decisión, las velocidades de las observaciones 6 y 8 son en promedio menores o iguales a 41.5km/h a diferencia de las velocidades de las observaciones 13 y 15 que son en promedio mayores o iguales a 79.5, las cuales indican un cambio brusco de velocidad. También las velocidades de las observaciones 17, 19, 21,

26, 29 y 31 son en promedio menores o iguales 29km/h y muestran un cambio brusco de velocidad con respecto a las velocidades de las observaciones 13 y 15.

	Condición
1	vel_6 \leq 44,5
2	vel_8 \leq 38,5
3	vel_13 \geq 78,5
4	vel_15 \geq 80,5
5	vel_17 \leq 26,5
6	vel_19 \leq 13,5
7	vel_21 \leq 34,5
8	vel_26 \leq 32,5
9	vel_29 \leq 23,5
10	vel_31 \leq 43,5

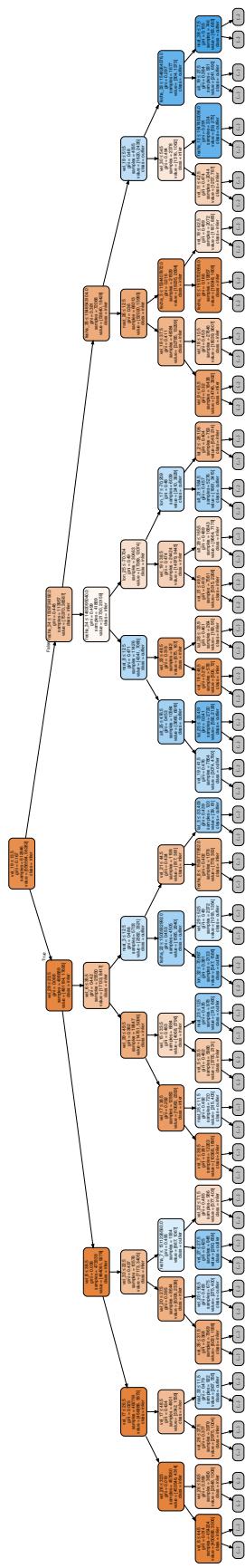
Cuadro 4.13: Tabla de regla de decisión para series de tiempo outlier

Ejemplo no outlier

En la tabla 4.14 se muestra las reglas de decisión, las velocidades de las observaciones 1 al 19 son en promedio menores o iguales a 45 km/h, la cual corresponde a una velocidad promedio normal con la que transitan los vehículos.

	Condición
1	vel_6 \leq 44,5
2	vel_7 \leq 42,5
3	vel_8 \leq 38,5
4	vel_9 \leq 43,5
5	vel_11 \leq 72,5
6	vel_13 \leq 43,5
7	vel_15 \leq 36,5
8	vel_17 \leq 26,5
9	vel_19 \leq 13,5
10	vel_20 \leq 26,5
11	vel_21 \leq 34,5
12	vel_22 \leq 42,5
13	vel_26 \leq 32,5
14	vel_28 \leq 43,5
15	vel_29 \leq 23,5
16	vel_30 \leq 42,5
17	vel_31 \leq 43,5
18	vel_33 \leq 53,5
19	vel_34 \leq 47,5
20	lon_37 \leq -64,114

Cuadro 4.14: Tabla de regla de decisión para series de tiempo normales



Parte V

Capítulo 5

Capítulo 5

Conclusiones

5.1. Discusión de los resultados

Inicialmente la empresa Waypoint identifico 2 tipos de outlier en datos de sensores multivariados de GPS, los cuales llamamos outlier espacial y outlier velocidad, la primera usa las variables geográficas y la segunda usa todas las variables. Se crearon 4 dataset donde el objetivo del dataset 1 y dataset 2 fue detectar outlier espaciales en trayectos cortos y largos respectivamente usando los 2 modelos propuestos, sin embargo al aplicar el dataset 2 con trayectos largos a la red autoencoder convolucional apilado se obtuvo un rendimiento de 5 % y la gran mayoría de trayectos se focalizan en una zona específica del mapa.

El objetivo del dataset 3 y dataset 4 fue detectar outlier de velocidad en trayectos cortos y largos respectivamente usando los 2 modelos propuestos, en los 2 dataset mencionados se pudo detectar el outlier espacial, pero también los modelos detectaron cambios bruscos o repentinos de velocidad en los trayectos y que fueron detectados como outlier.

Al analizar con mas detalle estos cambios bruscos de velocidad, se determino que este comportamiento anómalo puede tener una similitud con comportamientos anómalos de conductores por ejemplo, cuando los conductores encuentran ubicaciones peligrosas en la carretera y usan los frenos con fuerza para disminuir su velocidad o cuando frenan debido a algún imprevisto en la carretera que los obligue a frenar bruscamente, tales maniobras pueden ser denominados comportamientos anómalos de conductores y aparecen en la serie

de tiempo como anómalos.

5.2. Conclusiones del proyecto

Una vez concluido todas las fases planteadas en la sección 3.1, se obtuvo una solución que cubre el problema identificado, donde el objetivo general y objetivos específicos se cumplieron en su totalidad, en la tabla 5.1 se muestra los resultados de los objetivos específicos.

1. Entender las variables que puedan ser usados como predictoras para identificar trayectos outlier.
2. Desarrollar e implementar modelos de red neuronal que detecte trayectos de vehículos outlier exitosamente, basados en el historial de rutas de vehículos de Waypoint.

ID	Situación actual	Criterio de éxito	Resultado
1	No se tiene un análisis de las variables predictoras.	Analizar al menos 3 variables.	Análisis univariable para cada variable.
2	No se cuenta con ningún modelo para la detección trayectos outlier.	-El modelo detecto al menos un 30 % de trayectos outlier. -La precisión del modelo debe ser sobre el 90 %.	Modelo de red neuronal entrenado e implementado en el módulo historial de rutas.

Cuadro 5.1: Tabla de resultados de objetivos específicos

5.3. Trabajo a futuro

Para un trabajo a futuro se plantea extender el dataset 4 agregando nuevas variables como aceleración y dirección del vehículo, las cuales potencialmente pueden mejorar la capacidad de los modelos y enriquecer la detección de comportamientos anómalos de conductores, por ejemplo identificando localizaciones de trayectos donde los conductores frecuentemente usen los frenos con fuerza o hacen cambios repentinos de dirección esto puede significar carreteras peligrosas y por ello es importante identificarlos, ya que pueden ayudar a mejorar el diseño y superficie de las carreteras y ayudar a prevenir accidentes.

El solo poder detectar comportamientos anómalos de conductores abre muchas aristas de aplicación como recomendación de rutas personalizadas advirtiendo de carreteras peligrosas, en el monitoreo de vehículos detectando interrupciones en el trayecto las cuales van en contra de la política de seguridad vial de una determinada empresa, en la mejora del diseño de carreteras etc., todas estas aplicaciones se pueden centralizar en un sistema de transporte inteligente.

Apéndice A

Exploración de datos

A.1. Comportamiento de variables

A.1.1. Variable latitud

Esta variable presenta una distribución normal, se observa que los valores de los datos están entre -45 y -15 los cuales pertenecen a la localización de Chile en un mapa geográfico, además de valores -50 que pertenecen a la localización en Chile.

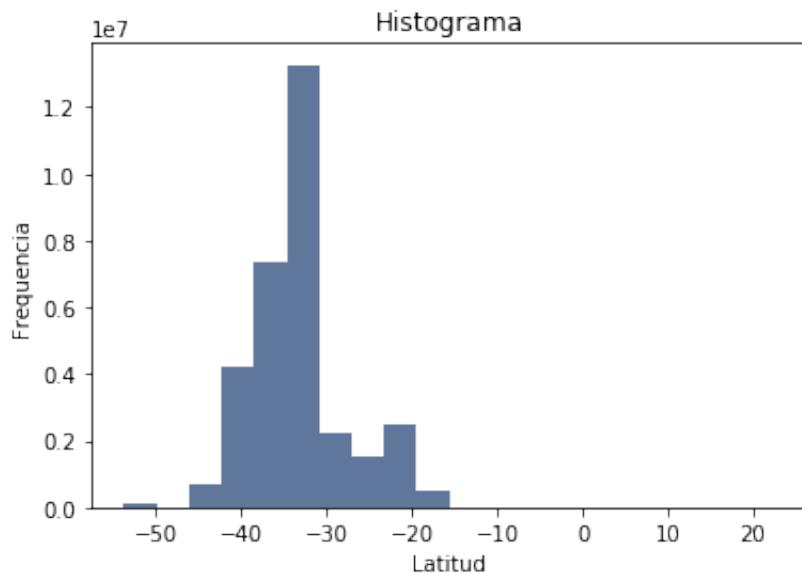


Figura A.1: Histograma latitud

A.1.2. Variable longitud

Esta variable presenta una distribución donde los valores están concentrados entre los valores -75 y -65 los cuales pertenecen a la localización de Chile en un mapa geográfico.

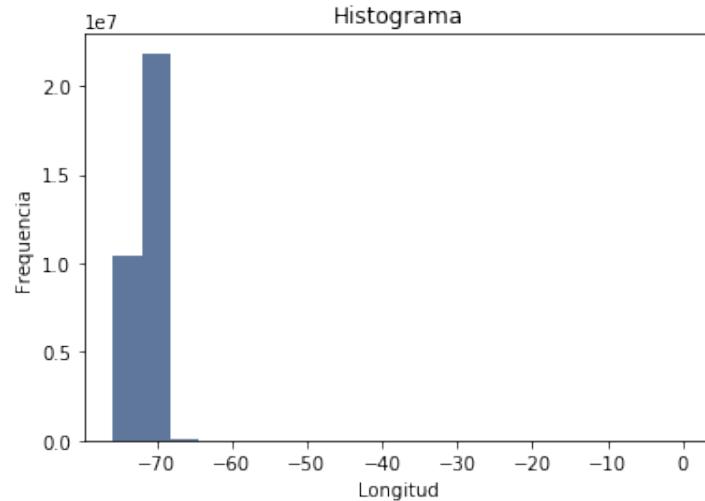


Figura A.2: Histograma longitud

A.1.3. Variable altitud

Esta variable presenta una distribución donde los valores están concentrados entre los valores 0 y 0.3.

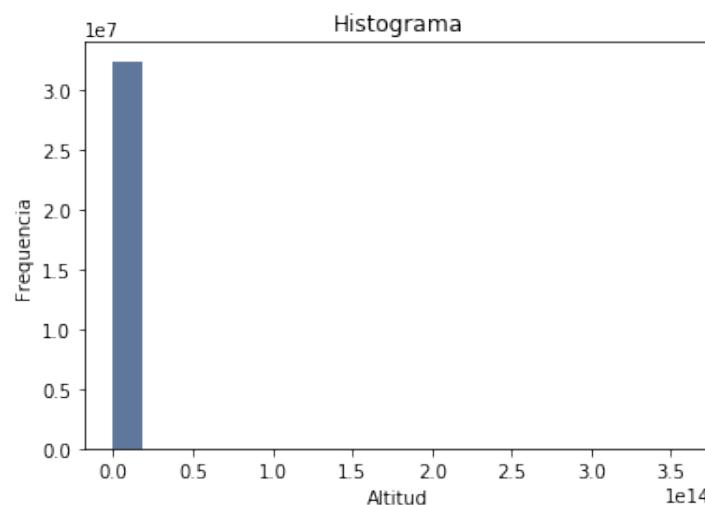


Figura A.3: Histograma altitud

A.1.4. Variable velocidad

Esta variable presenta una distribución donde los valores están concentrados entre los valores 0 y 130. Se puede observar que hay pocos datos con valores mayores a 100, lo que da a conocer que los vehículos inusualmente llegan a una velocidad sobre los 100 km/h.

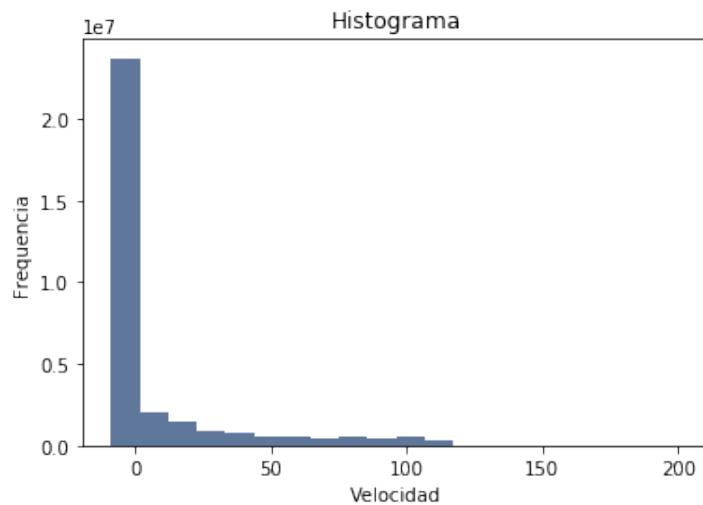


Figura A.4: Histograma velocidad

A.1.5. Variable número de satélites

Esta variable presenta una distribución donde los valores están concentrados entre los valores 0 y 40.

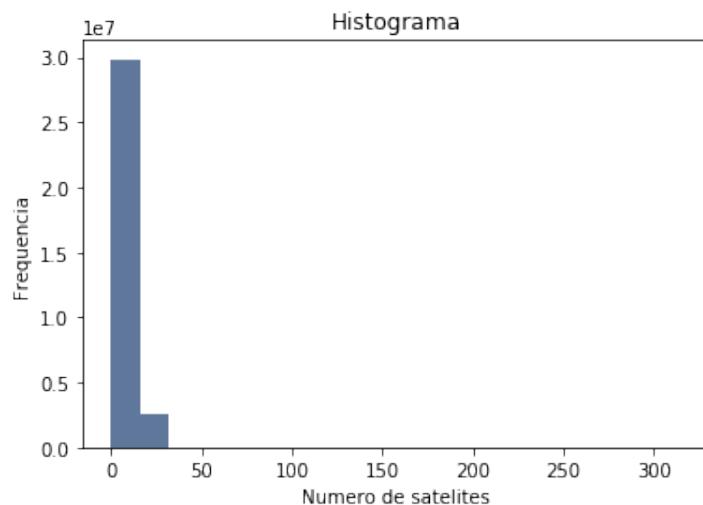


Figura A.5: Histograma número de satélites

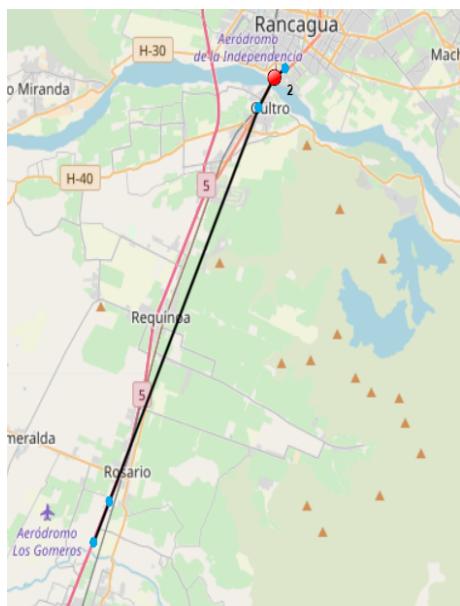
Apéndice B

Ejemplos de series de tiempo

B.0.1. Dataset 1(Autoencoder apilado)

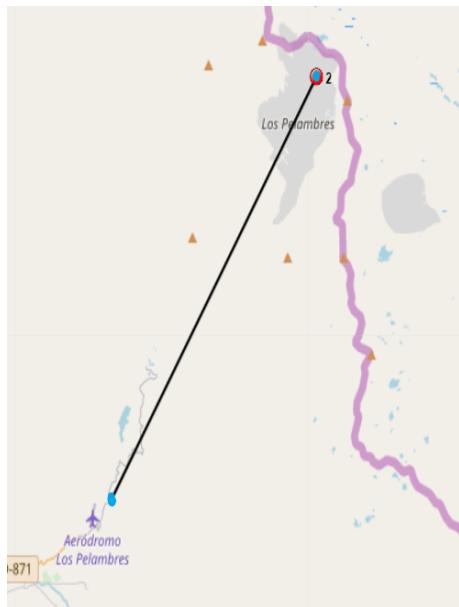
Ejemplos de trayectos outlier

Este trayecto muestra una discontinuidad evidente, la observación 2 sale de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-34.187734	-70.747370	516.0	0
2	-34.191601	-70.753629	514.0	1
3	-34.203193	-70.762143	514.0	0
4	-34.357149	-70.847020	333.0	0
5	-34.372735	-70.855953	319.0	0

Este trayecto muestra una discontinuidad evidente, la observación 4 sale de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-31.865801	-70.605883	1295.0	0
2	-31.865071	-70.606081	1298.0	0
3	-31.864850	-70.606261	1300.0	0
4	-31.705029	-70.483393	3532.0	1
5	-31.705016	-70.483268	3531.0	0

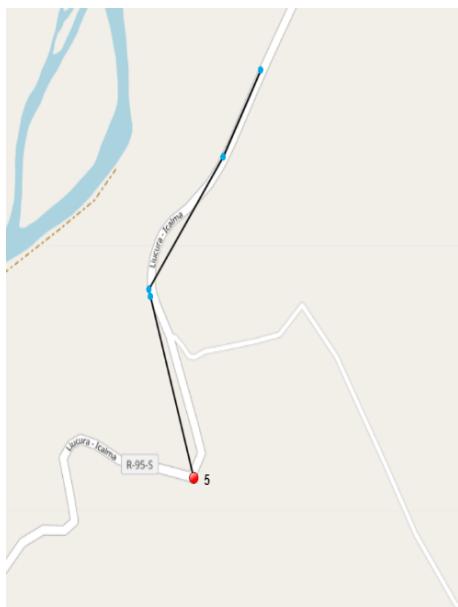
Este trayecto muestra una discontinuidad evidente, la observación 1 sale de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-34.358763	-70.847926	330.0	1
2	-34.588081	-70.990638	338.0	0
3	-34.588081	-70.990638	338.0	0
4	-34.368590	-70.853608	321.0	0
5	-34.588081	-70.990638	338.0	0

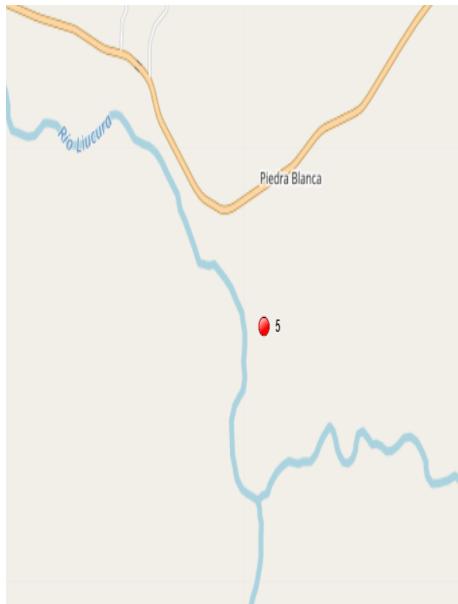
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo la observación 5 es detectada como outlier.



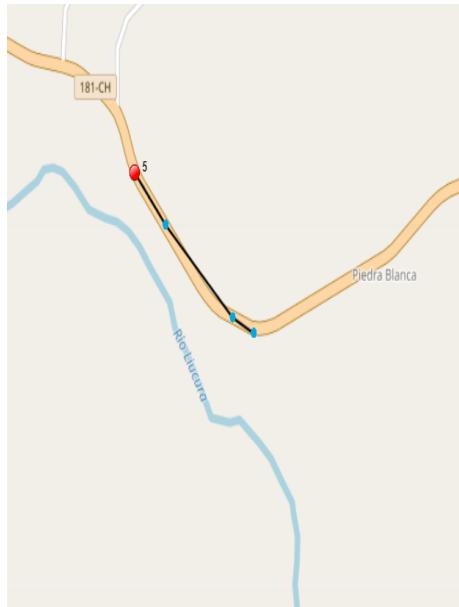
	latitud	longitud	altitud	outlier
1	-38.716963	-71.111091	1067.0	0
2	-38.718360	-71.112076	1070.0	0
3	-38.720508	-71.114023	1075.0	0
4	-38.720621	-71.113995	1075.0	0
5	-38.723561	-71.112840	1090.0	1

Este trayecto muestra observaciones fijas, sin embargo la observación 5 es detectada como outlier.



	latitud	longitud	altitud	outlier
1	-38.658257	-71.03591	1133.0	0
2	-38.658257	-71.03591	1133.0	0
3	-38.658257	-71.03591	1133.0	0
4	-38.658257	-71.03591	1133.0	0
5	-38.658257	-71.03591	1133.0	1

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo la observación 5 es detectada como outlier.

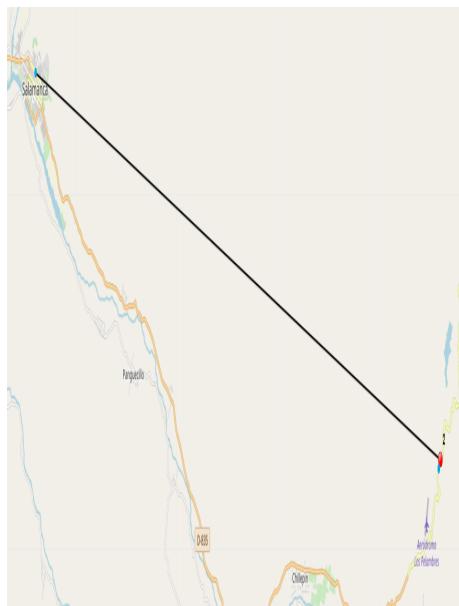


	latitud	longitud	altitud	outlier
1	-38.653730	-71.039598	1129.0	0
2	-38.653730	-71.039598	1129.0	0
3	-38.653390	-71.040591	1124.0	0
4	-38.651205	-71.043744	1097.0	0
5	-38.649961	-71.045213	1085.0	1

B.0.2. Dataset 1(Autoencoder convolucional apilado)

Ejemplos de trayectos outlier

Este trayecto muestra una discontinuidad evidente, la observación 2 sale de la continuidad de las demás observaciones.



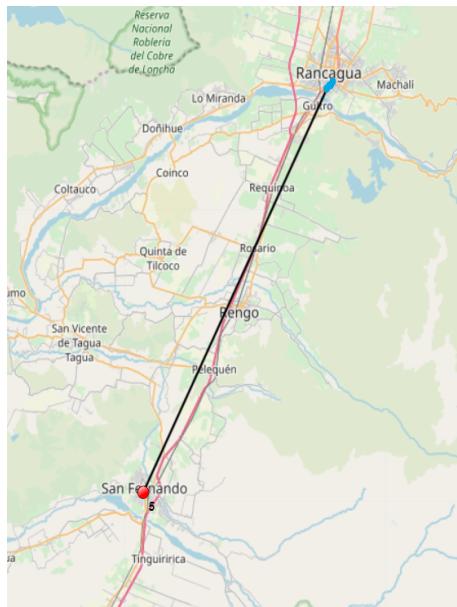
	latitud	longitud	altitud	outlier
1	-31.860540	-70.606580	1344.0	0
2	-31.858513	-70.605276	1360.0	1
3	-31.777200	-70.964386	513.0	0
4	-31.777200	-70.964386	513.0	0
5	-31.777194	-70.964395	513.0	0

Este trayecto muestra una discontinuidad evidente, la observación 5 sale de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-34.588081	-70.990638	338.0	0
2	-34.588081	-70.990638	338.0	0
3	-34.588081	-70.990638	338.0	0
4	-34.588081	-70.990638	338.0	0
5	-34.166140	-70.743188	484.0	1

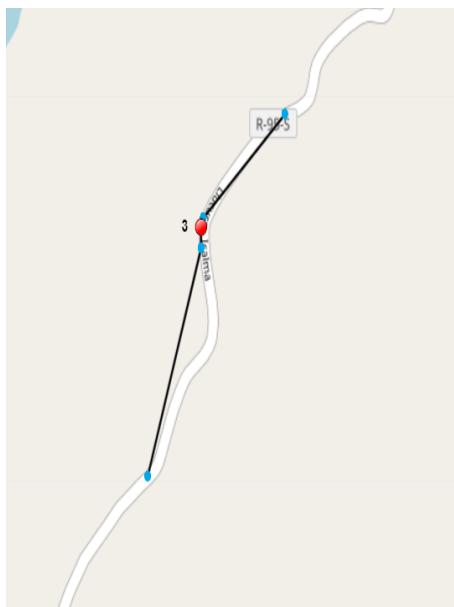
Este trayecto muestra una discontinuidad evidente, la observación 5 sale de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-34.180390	-70.735098	512.0	0
2	-34.182906	-70.738441	511.0	0
3	-34.183551	-70.739315	508.0	0
4	-34.186695	-70.743608	513.0	0
5	-34.588081	-70.990638	338.0	1

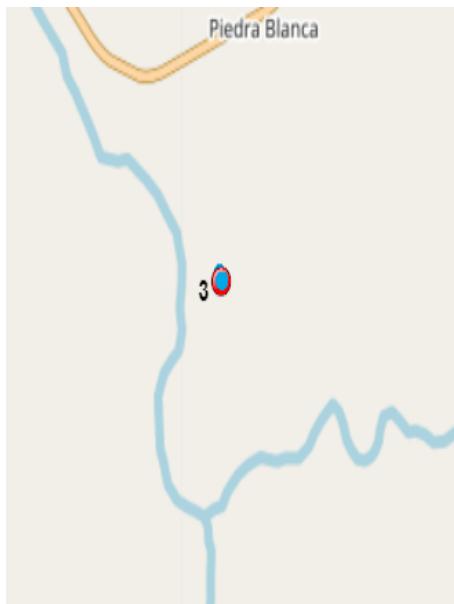
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo la observación 3 es detectada como outlier.



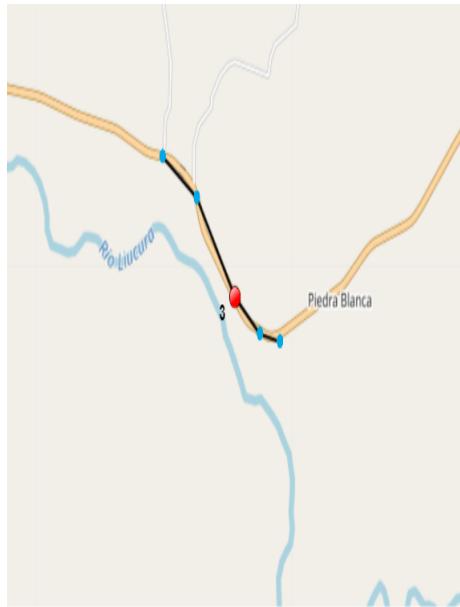
	latitud	longitud	altitud	outlier
1	-38.702866	-71.100183	1084.0	0
2	-38.704016	-71.101856	1080.0	0
3	-38.704120	-71.101901	1080.0	1
4	-38.704346	-71.101914	1082.0	0
5	-38.706894	-71.103001	1102.0	0

Este trayecto muestra las observaciones fijas, sin embargo la observación 3 es detectada como outlier.



	latitud	longitud	altitud	outlier
1	-38.658761	-71.035694	1123.0	0
2	-38.658946	-71.035629	1122.0	0
3	-38.658971	-71.035593	1122.0	1
4	-38.658970	-71.035561	1122.0	0
5	-38.658950	-71.035604	1122.0	0

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo la observación 3 es detectada como outlier.



	latitud	longitud	altitud	outlier
1	-38.647463	-71.048651	1075.0	0
2	-38.648833	-71.045776	1076.0	0
3	-38.652291	-71.042524	1107.0	1
4	-38.653513	-71.040370	1124.0	0
5	-38.653810	-71.038603	1130.0	0

B.0.3. Dataset 2(Autoencoder apilado)

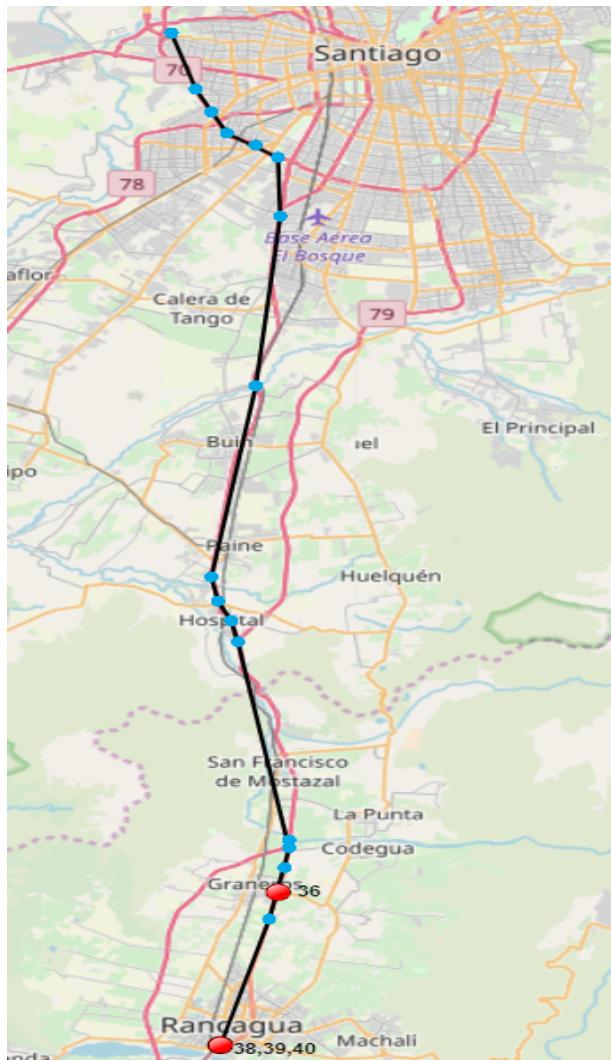
Ejemplos de trayectos outlier

Este trayecto muestra una discontinuidad evidente, las observaciones 9,38,39,40 salen de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-33.469985	-70.771001	473.0	0
2	-33.469985	-70.771001	473.0	0
3	-33.469985	-70.771001	473.0	0
4	-33.469985	-70.771001	473.0	0
5	-33.469985	-70.771001	473.0	0
6	-33.469985	-70.771001	473.0	0
7	-33.469985	-70.771001	473.0	0
8	-33.469985	-70.771001	473.0	0
9	-33.469985	-70.771001	473.0	1
10	-33.469985	-70.771001	473.0	0
11	-33.470008	-70.771000	471.0	0
12	-33.470195	-70.770932	469.0	0
13	-33.470206	-70.770898	468.0	0
14	-33.470238	-70.770946	468.0	0
15	-33.470347	-70.771091	468.0	0
16	-33.470347	-70.771251	468.0	0
17	-33.470347	-70.771251	468.0	0
18	-33.470358	-70.771268	467.0	0
19	-33.470375	-70.771154	467.0	0
20	-33.470388	-70.771163	467.0	0
21	-33.470433	-70.771176	467.0	0
22	-33.470437	-70.771170	472.0	0
23	-33.470437	-70.771170	472.0	0
24	-33.470437	-70.771170	472.0	0
25	-33.470400	-70.771160	472.0	0
26	-33.470327	-70.771093	471.0	0
27	-33.470335	-70.771097	471.0	0
28	-33.470347	-70.771099	471.0	0
29	-33.470383	-70.771108	470.0	0
30	-33.470391	-70.771110	471.0	0
31	-33.470387	-70.771106	471.0	0
32	-33.470405	-70.771111	471.0	0
33	-33.470403	-70.771111	471.0	0
34	-33.470403	-70.771111	471.0	0
35	-33.470403	-70.771111	471.0	0
36	-33.470403	-70.771111	471.0	0
37	-33.470403	-70.771111	471.0	0
38	-23.512723	-70.401990	88.0	1
39	-23.512723	-70.401990	88.0	1
40	-23.512751	-70.401975	87.0	1

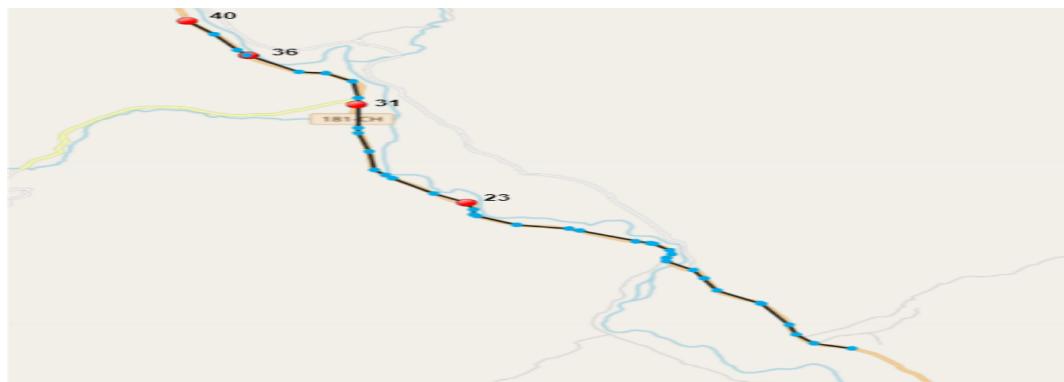
Este trayecto muestra una discontinuidad evidente, las observaciones 36, 38, 39, 40 salen de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-33.422883	-70.778663	479.0	0
2	-33.422883	-70.778663	479.0	0
3	-33.422883	-70.778663	479.0	0
4	-33.422883	-70.778663	479.0	0
5	-33.422883	-70.778663	479.0	0
6	-33.422883	-70.778663	479.0	0
7	-33.422883	-70.778663	479.0	0
8	-33.422883	-70.778663	479.0	0
9	-33.422883	-70.778663	479.0	0
10	-33.422883	-70.778663	479.0	0
11	-33.422883	-70.778663	479.0	0
12	-33.422883	-70.778663	479.0	0
13	-33.422883	-70.778663	479.0	0
14	-33.465871	-70.762625	476.0	0
15	-33.483160	-70.753411	485.0	0
16	-33.499166	-70.743653	484.0	0
17	-33.507741	-70.725519	505.0	0
18	-33.517247	-70.711577	517.0	0
19	-33.561983	-70.710941	544.0	0
20	-33.689576	-70.725576	515.0	0
21	-33.689576	-70.725576	515.0	0
22	-33.689576	-70.725576	515.0	0
23	-33.689576	-70.725576	515.0	0
24	-33.689761	-70.725510	514.0	0
25	-33.689761	-70.725510	514.0	0
26	-33.689761	-70.725510	514.0	0
27	-33.689761	-70.725510	514.0	0
28	-33.689845	-70.725478	514.0	0
29	-33.833432	-70.753156	369.0	0
30	-33.852038	-70.748668	373.0	0
31	-33.886667	-70.741416	375.0	0
32	-33.882881	-70.736308	384.0	0
33	-34.032690	-70.705467	498.0	0
34	-34.037878	-70.705938	500.0	0
35	-34.052853	-70.707330	493.0	0
36	-34.070998	-70.712236	480.0	1
37	-34.091867	-70.717901	469.0	0
38	-34.185816	-70.748173	522.0	1
39	-34.185886	-70.748088	518.0	1
40	-34.185895	-70.748196	512.0	1

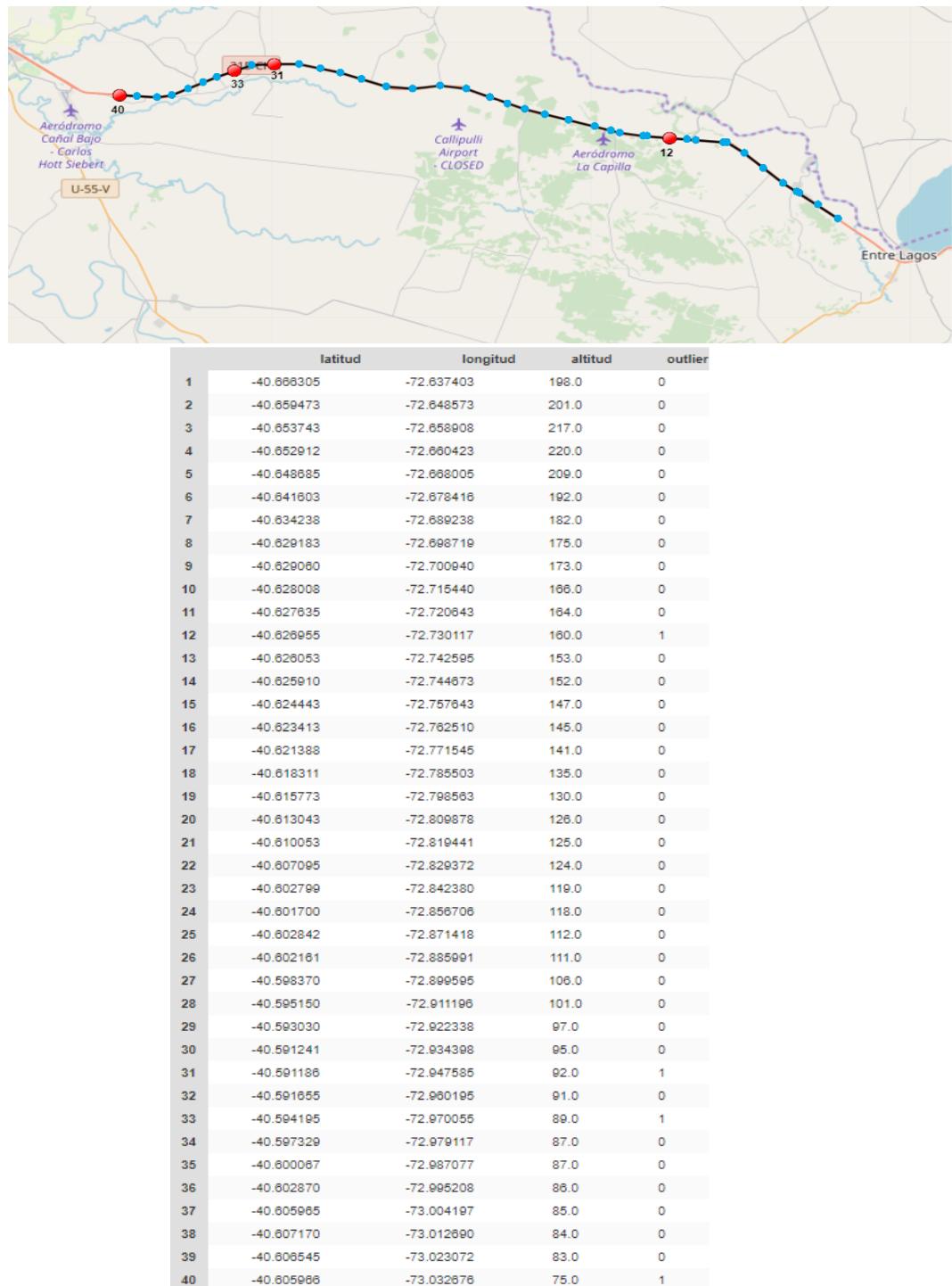
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo las observaciones 23,31,36,40 son detectadas como outlier.

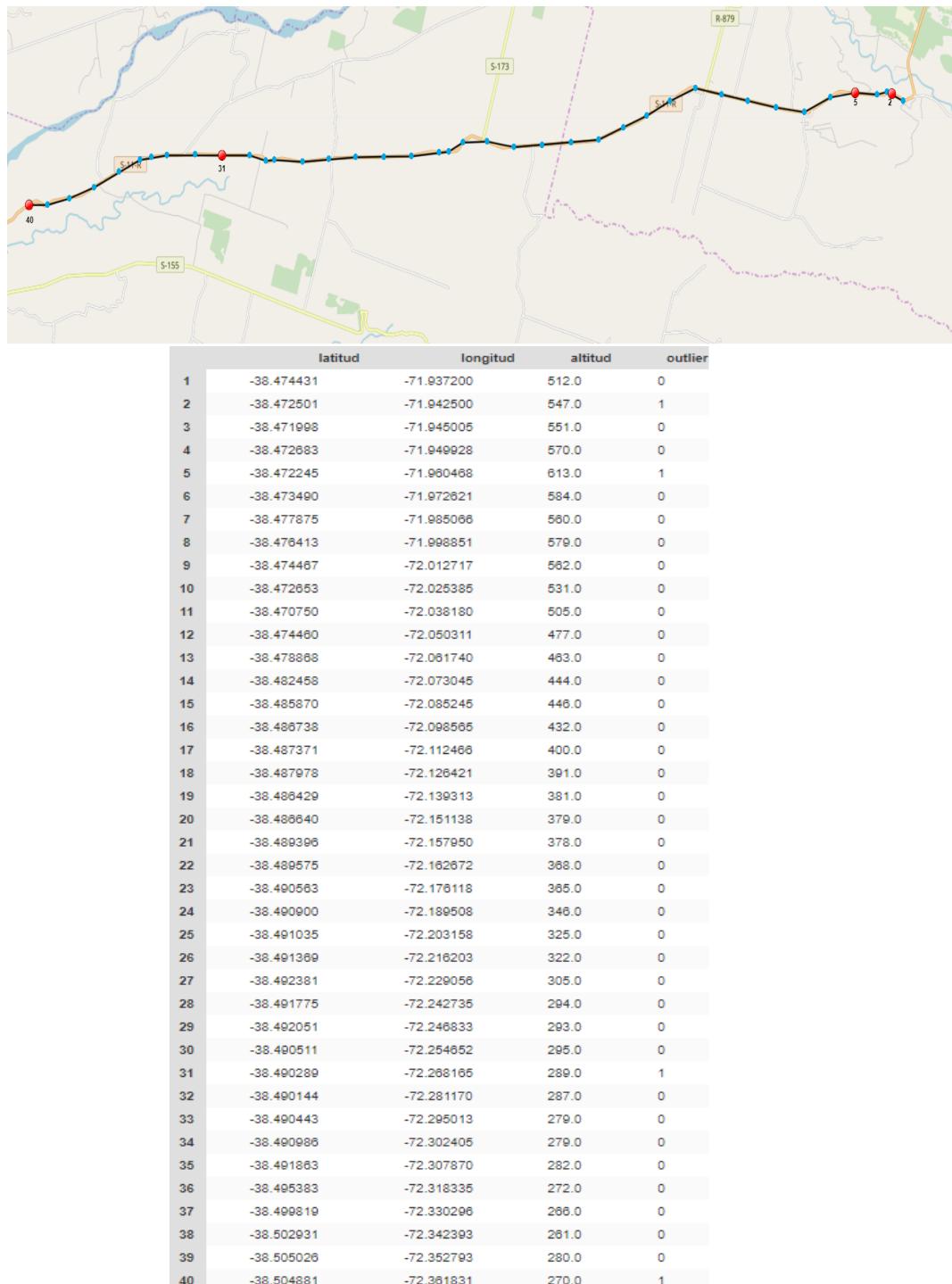


	latitud	longitud	altitud	outlier
1	-38.627045	-71.106975	1039.0	0
2	-38.625038	-71.114048	1040.0	0
3	-38.621208	-71.117458	1040.0	0
4	-38.617138	-71.119039	1042.0	0
5	-38.608275	-71.124056	1038.0	0
6	-38.607903	-71.124359	1037.0	0
7	-38.602715	-71.132591	1014.0	0
8	-38.597588	-71.134961	1011.0	0
9	-38.594129	-71.137098	997.0	0
10	-38.590403	-71.142256	979.0	0
11	-38.588877	-71.142163	977.0	0
12	-38.587263	-71.141351	981.0	0
13	-38.585663	-71.141464	995.0	0
14	-38.583070	-71.144725	1000.0	0
15	-38.582705	-71.145478	1001.0	0
16	-38.581978	-71.148273	1005.0	0
17	-38.577455	-71.158953	1002.0	0
18	-38.576643	-71.160963	995.0	0
19	-38.574922	-71.170921	976.0	0
20	-38.571353	-71.178493	986.0	0
21	-38.570620	-71.179075	984.0	0
22	-38.568478	-71.179061	977.0	0
23	-38.565718	-71.180564	963.0	1
24	-38.561760	-71.186624	971.0	0
25	-38.555380	-71.194515	965.0	0
26	-38.554168	-71.195764	965.0	0
27	-38.551808	-71.197834	965.0	0
28	-38.544141	-71.198978	960.0	0
29	-38.536254	-71.200868	961.0	0
30	-38.534108	-71.201045	961.0	0
31	-38.524390	-71.201391	959.0	1
32	-38.521451	-71.200971	954.0	0
33	-38.514481	-71.202053	941.0	0
34	-38.511049	-71.207190	935.0	0
35	-38.510491	-71.212473	941.0	0
36	-38.503701	-71.222116	926.0	1
37	-38.503506	-71.222273	926.0	0
38	-38.501405	-71.223910	927.0	0
39	-38.494791	-71.228315	911.0	0
40	-38.489013	-71.233471	928.0	1

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo las observaciones 12,31,33,40 son detectadas como outlier.



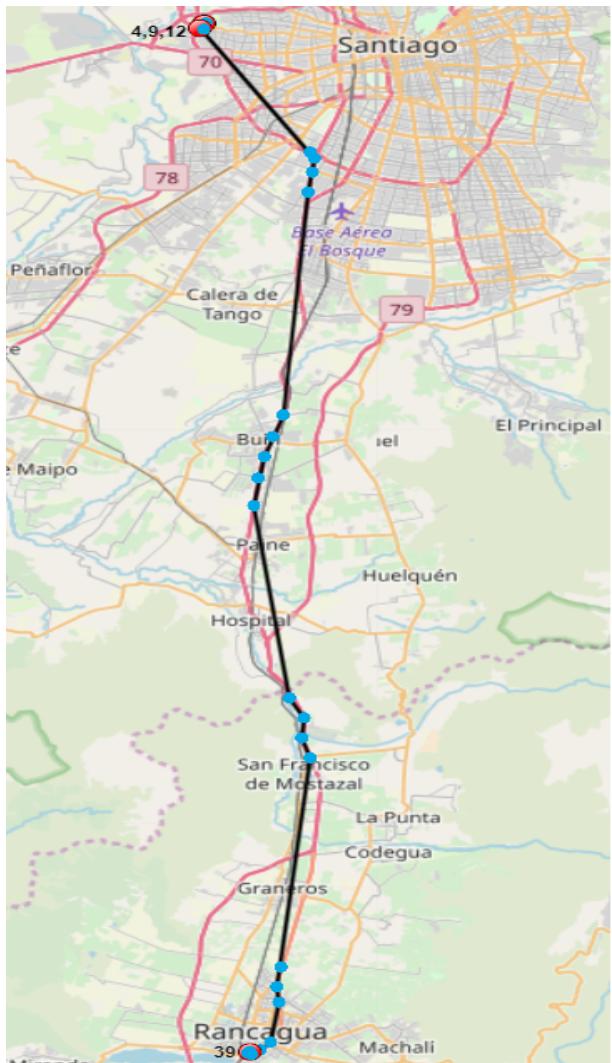
Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo las observaciones 2,5,31,40 son detectadas como outlier.



B.0.4. Dataset 2(Autoencoder convolucional apilado)

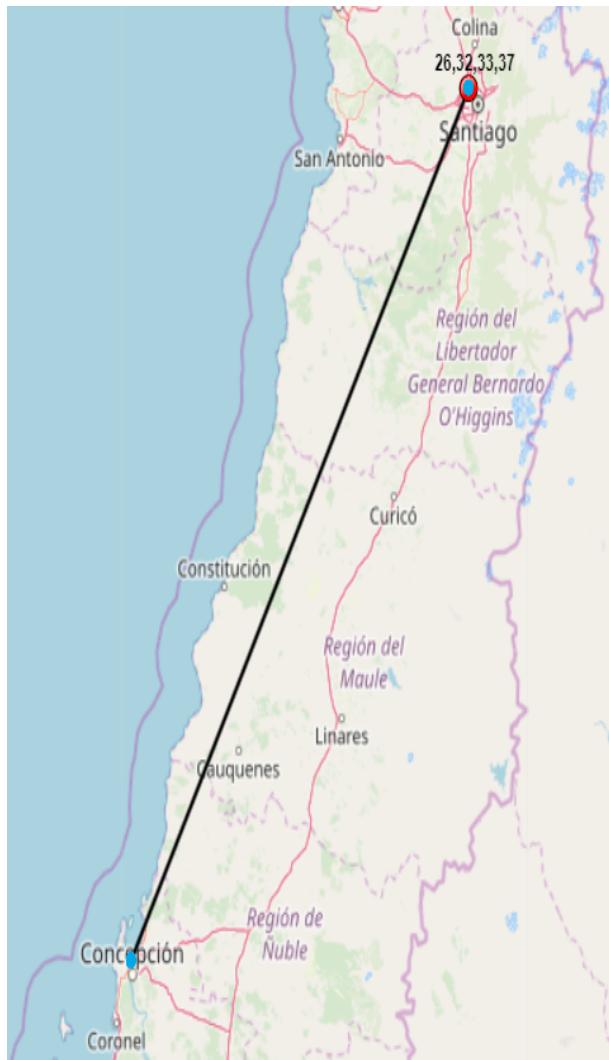
Ejemplos de trayectos outlier

Este trayecto muestra una discontinuidad evidente, las observaciones 4,9,12 salen de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-33.422700	-70.778280	485.0	0
2	-33.422700	-70.778280	485.0	0
3	-33.422700	-70.778280	485.0	0
4	-33.422700	-70.778280	485.0	1
5	-33.422700	-70.778280	485.0	0
6	-33.422700	-70.778280	485.0	0
7	-33.422700	-70.778280	485.0	0
8	-33.422700	-70.778280	485.0	0
9	-33.422616	-70.778276	486.0	1
10	-33.422805	-70.777295	477.0	0
11	-33.427093	-70.777841	480.0	0
12	-33.425380	-70.781183	480.0	1
13	-33.518872	-70.708003	521.0	0
14	-33.523241	-70.705603	526.0	0
15	-33.523262	-70.706678	534.0	0
16	-33.548652	-70.708736	539.0	0
17	-33.714396	-70.725506	503.0	0
18	-33.730137	-70.732323	482.0	0
19	-33.744803	-70.738553	485.0	0
20	-33.761017	-70.741655	447.0	0
21	-33.781616	-70.744996	424.0	0
22	-33.923695	-70.721239	413.0	0
23	-33.938918	-70.712278	438.0	0
24	-33.953940	-70.713531	453.0	0
25	-33.968810	-70.707793	466.0	0
26	-34.122044	-70.726656	468.0	0
27	-34.137758	-70.729414	481.0	0
28	-34.148884	-70.728581	491.0	0
29	-34.179331	-70.733693	518.0	0
30	-34.184431	-70.740595	504.0	0
31	-34.187891	-70.746688	516.0	0
32	-34.186275	-70.749538	513.0	0
33	-34.185680	-70.749206	513.0	0
34	-34.185863	-70.748835	513.0	0
35	-34.185783	-70.748371	516.0	0
36	-34.185827	-70.748266	516.0	0
37	-34.185827	-70.748266	516.0	0
38	-34.185827	-70.748266	516.0	0
39	-34.185827	-70.748266	516.0	1
40	-34.185827	-70.748266	516.0	0

Este trayecto muestra una discontinuidad evidente, las observaciones 26,32,33,37 salen de la continuidad de las demás observaciones.



	latitud	longitud	altitud	outlier
1	-36.776421	-73.057411	11.0	0
2	-36.776421	-73.057411	11.0	0
3	-36.776421	-73.057411	11.0	0
4	-36.776421	-73.057411	11.0	0
5	-36.776421	-73.057411	11.0	0
6	-36.776421	-73.057411	11.0	0
7	-36.776370	-73.057446	7.0	0
8	-36.776366	-73.057498	7.0	0
9	-36.776456	-73.057640	6.0	0
10	-36.777375	-73.058133	4.0	0
11	-36.777420	-73.058103	4.0	0
12	-36.777428	-73.058100	4.0	0
13	-36.777477	-73.057990	5.0	0
14	-36.777444	-73.057933	5.0	0
15	-36.777455	-73.057950	5.0	0
16	-36.777466	-73.057961	6.0	0
17	-36.777472	-73.057968	6.0	0
18	-36.777465	-73.057956	6.0	0
19	-36.777462	-73.057950	6.0	0
20	-36.777462	-73.057950	6.0	0
21	-36.777462	-73.057950	6.0	0
22	-36.777390	-73.057840	7.0	0
23	-36.777390	-73.057840	7.0	0
24	-36.777390	-73.057840	7.0	0
25	-33.374988	-70.721756	496.0	0
26	-33.374988	-70.721756	496.0	1
27	-33.374981	-70.721705	507.0	0
28	-33.374981	-70.721705	507.0	0
29	-33.374981	-70.721705	507.0	0
30	-33.374981	-70.721705	507.0	0
31	-33.374981	-70.721705	507.0	0
32	-33.374981	-70.721705	507.0	1
33	-33.374981	-70.721705	507.0	1
34	-33.374981	-70.721705	507.0	0
35	-33.374981	-70.721705	507.0	0
36	-33.374981	-70.721705	507.0	0
37	-33.374981	-70.721705	507.0	1
38	-33.374951	-70.721701	499.0	0
39	-33.374950	-70.721706	499.0	0
40	-33.374928	-70.721758	498.0	0

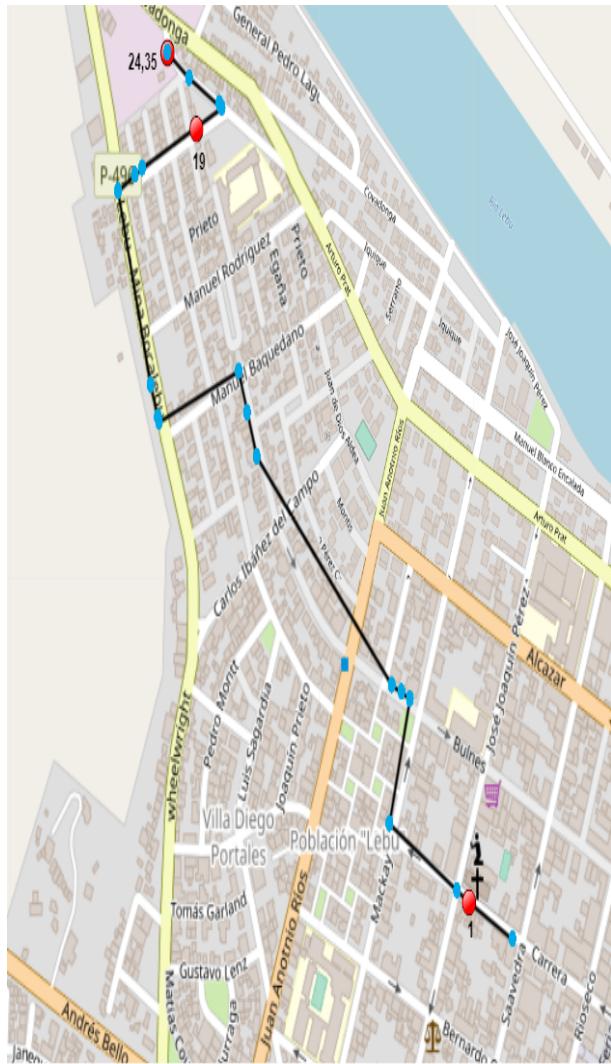
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo las observaciones 2,19,24,35 son detectadas como outlier.



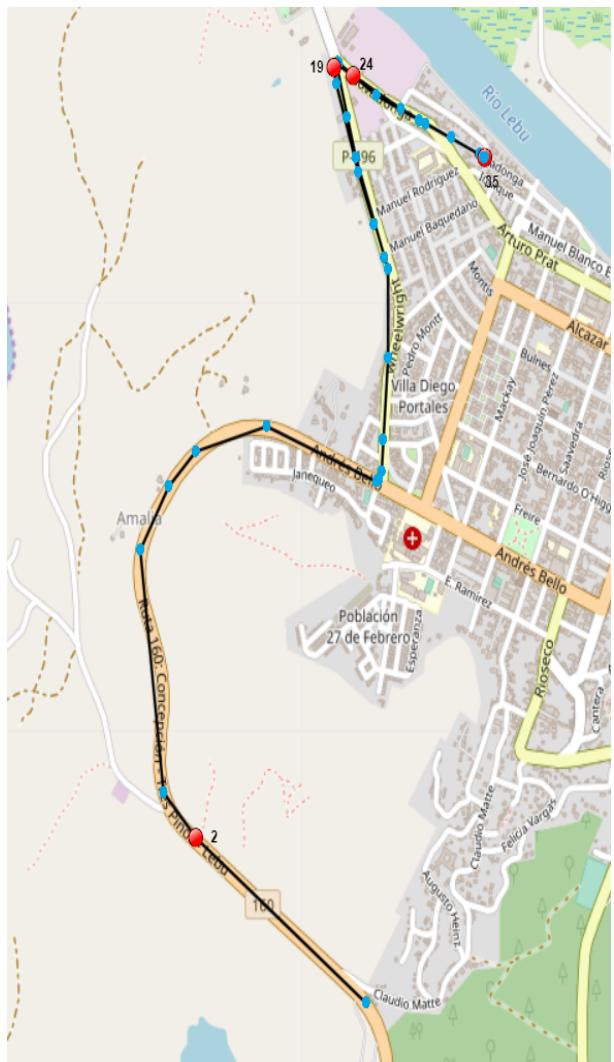
	latitud	longitud	altitud	outlier
1	-37.614330	-73.668042	99.0	0
2	-37.614330	-73.668042	99.0	1
3	-37.613505	-73.667947	97.0	0
4	-37.608735	-73.667095	57.0	0
5	-37.608735	-73.667095	57.0	0
6	-37.608225	-73.666025	49.0	0
7	-37.608033	-73.664871	40.0	0
8	-37.608033	-73.664871	40.0	0
9	-37.609195	-73.660707	9.0	0
10	-37.609195	-73.660707	9.0	0
11	-37.609170	-73.660693	9.0	0
12	-37.606987	-73.660557	4.0	0
13	-37.605950	-73.660177	4.0	0
14	-37.605488	-73.660166	3.0	0
15	-37.605198	-73.660269	3.0	0
16	-37.604468	-73.660658	3.0	0
17	-37.602028	-73.661688	4.0	0
18	-37.601358	-73.661998	5.0	0
19	-37.600707	-73.662173	3.0	1
20	-37.600707	-73.662173	3.0	0
21	-37.600670	-73.662103	3.0	0
22	-37.600710	-73.661970	2.0	0
23	-37.600770	-73.661878	2.0	0
24	-37.600908	-73.661637	1.0	1
25	-37.601563	-73.660055	0.0	0
26	-37.601845	-73.659286	0.0	0
27	-37.602163	-73.658371	0.0	0
28	-37.602550	-73.657261	-1.0	0
29	-37.602682	-73.656063	-1.0	0
30	-37.603248	-73.655861	0.0	0
31	-37.603248	-73.655861	0.0	0
32	-37.603465	-73.655826	0.0	0
33	-37.603905	-73.656038	0.0	0
34	-37.604295	-73.654861	-1.0	0
35	-37.605346	-73.651768	1.0	1
36	-37.606360	-73.649703	0.0	0
37	-37.606413	-73.649518	1.0	0
38	-37.606413	-73.649518	1.0	0
39	-37.604863	-73.647476	7.0	0
40	-37.603680	-73.645600	4.0	0

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo las observaciones 2,19,24,35 son detectadas como outlier.



	latitud	longitud	altitud	outlier
1	-37.608180	-73.654398	6.0	0
2	-37.607908	-73.655188	5.0	1
3	-37.607828	-73.655390	5.0	0
4	-37.607336	-73.656601	5.0	0
5	-37.607336	-73.656601	5.0	0
6	-37.606411	-73.656231	5.0	0
7	-37.606411	-73.656231	5.0	0
8	-37.606388	-73.656378	5.0	0
9	-37.606313	-73.656558	5.0	0
10	-37.604631	-73.658953	5.0	0
11	-37.604310	-73.659115	5.0	0
12	-37.604000	-73.659277	5.0	0
13	-37.604373	-73.660695	6.0	0
14	-37.604336	-73.660720	6.0	0
15	-37.604103	-73.660841	6.0	0
16	-37.602667	-73.661418	8.0	0
17	-37.602546	-73.661118	7.0	0
18	-37.602510	-73.660986	7.0	0
19	-37.602215	-73.660021	7.0	1
20	-37.602068	-73.659583	6.0	0
21	-37.602050	-73.659590	6.0	0
22	-37.602030	-73.659615	6.0	0
23	-37.601842	-73.660153	6.0	0
24	-37.601653	-73.660543	7.0	1
25	-37.601653	-73.660543	7.0	0
26	-37.601653	-73.660543	7.0	0
27	-37.601653	-73.660543	7.0	0
28	-37.601653	-73.660543	7.0	0
29	-37.601653	-73.660543	7.0	0
30	-37.601653	-73.660543	7.0	0
31	-37.601653	-73.660543	7.0	0
32	-37.601653	-73.660543	7.0	0
33	-37.601653	-73.660543	7.0	0
34	-37.601653	-73.660543	7.0	0
35	-37.601653	-73.660543	7.0	1
36	-37.601653	-73.660543	7.0	0
37	-37.601653	-73.660543	7.0	0
38	-37.601653	-73.660543	7.0	0
39	-37.601653	-73.660543	7.0	0
40	-37.601653	-73.660543	7.0	0

Este trayecto muestra una continuidad normal en todas las observaciones, sin embargo las observaciones 2,19,24,35 son detectadas como outlier.

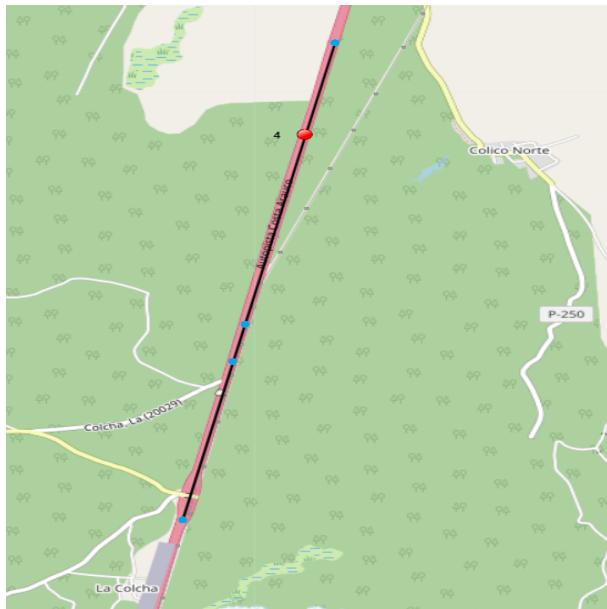


	latitud	longitud	altitud	outlier
1	-37.619756	-73.661183	121.0	0
2	-37.616422	-73.666860	103.0	1
3	-37.615483	-73.667920	99.0	0
4	-37.610575	-73.668697	77.0	0
5	-37.609281	-73.667751	66.0	0
6	-37.608551	-73.666866	58.0	0
7	-37.608040	-73.664483	40.0	0
8	-37.609165	-73.660861	9.0	0
9	-37.609135	-73.660743	8.0	0
10	-37.608966	-73.660650	8.0	0
11	-37.608310	-73.660620	7.0	0
12	-37.606650	-73.660453	7.0	0
13	-37.604851	-73.660475	6.0	0
14	-37.604625	-73.660568	5.0	0
15	-37.602558	-73.661521	4.0	0
16	-37.601770	-73.661837	4.0	0
17	-37.600883	-73.662187	4.0	0
18	-37.600677	-73.662083	4.0	0
19	-37.600916	-73.661610	1.0	1
20	-37.601578	-73.660005	1.0	0
21	-37.601878	-73.659203	0.0	0
22	-37.602172	-73.658363	0.0	0
23	-37.602581	-73.657235	0.0	0
24	-37.602585	-73.657243	1.0	1
25	-37.602585	-73.657243	1.0	0
26	-37.602565	-73.657258	1.0	0
27	-37.602565	-73.657258	1.0	0
28	-37.602531	-73.657280	1.0	0
29	-37.602537	-73.657273	2.0	0
30	-37.602517	-73.657393	2.0	0
31	-37.601820	-73.659415	1.0	0
32	-37.601317	-73.660833	0.0	0
33	-37.600646	-73.662141	0.0	0
34	-37.600658	-73.662195	0.0	0
35	-37.600732	-73.662236	1.0	1
36	-37.601058	-73.662156	2.0	0
37	-37.602873	-73.661431	0.0	0
38	-37.603928	-73.660908	0.0	0
39	-37.603928	-73.660908	0.0	0
40	-37.603928	-73.660908	0.0	0

B.0.5. Dataset 3(Autoencoder apilado)

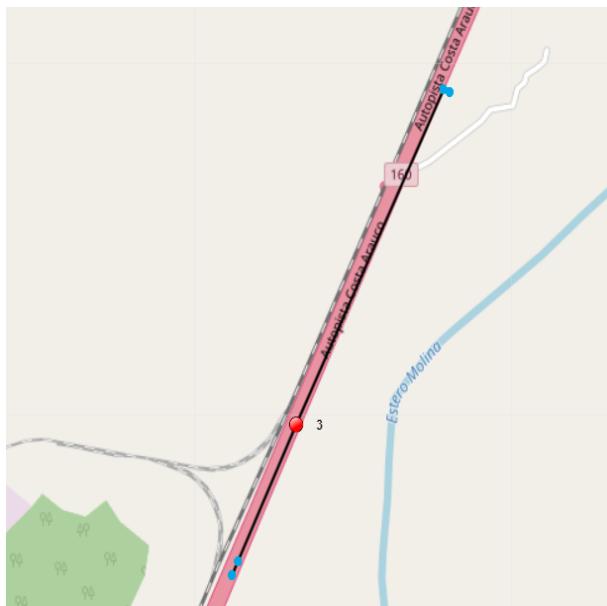
Ejemplos de trayectos outlier

Este trayecto tiene una continuidad normal, sin embargo la observación 4 muestra un cambio brusco de velocidad con respecto a la siguiente observación.



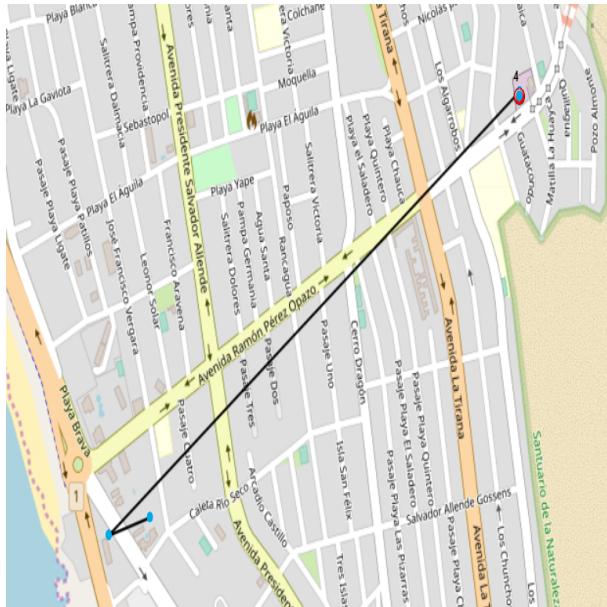
	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2017-12-30 13:10:04	-37.418948	-73.349205	140.0	139.0	10.0	0
2	2017-12-30 13:10:36	-37.407373	-73.346086	145.0	137.0	10.0	0
3	2017-12-30 13:10:44	-37.404656	-73.345360	137.0	146.0	9.0	0
4	2017-12-30 13:11:24	-37.390788	-73.341608	138.0	150.0	9.0	1
5	2017-12-30 13:12:04	-37.384083	-73.339738	143.0	5.0	7.0	0

Este trayecto muestra una continuidad normal, sin embargo la observación 3 muestra un cambio brusco de velocidad con respecto a la siguiente observación.



	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2018-01-19 14:45:06	-37.208093	-73.211608	8.0	162.0	10.0	0
2	2018-01-19 14:45:08	-37.207725	-73.211390	8.0	162.0	10.0	0
3	2018-01-19 14:45:16	-37.204315	-73.209376	7.0	170.0	10.0	1
4	2018-01-19 14:45:43	-37.195997	-73.204265	8.0	30.0	10.0	0
5	2018-01-19 14:45:46	-37.196033	-73.204058	7.0	22.0	10.0	0

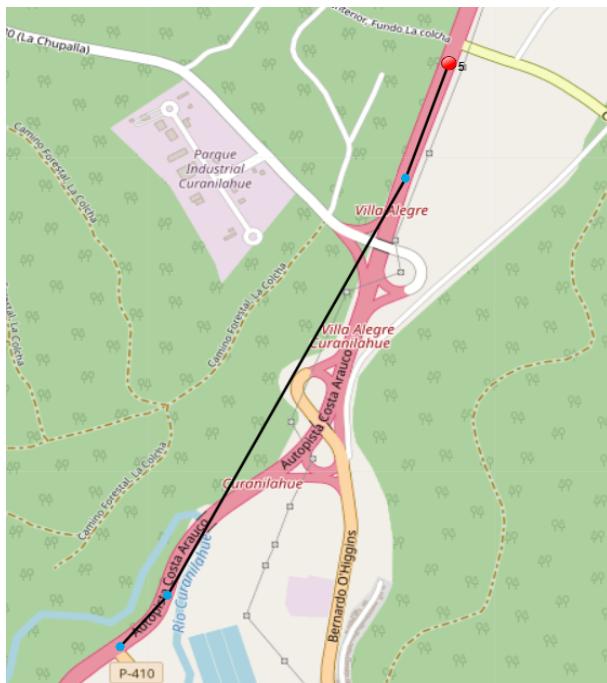
Este trayecto muestra una discontinuidad evidente, las observación 4 sale de la continuidad de las demás observaciones.



	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2017-08-02 20:40:05	-20.256353	-70.132721	19.0	0.0	6.0	0
2	2017-08-02 20:57:00	-20.256605	-70.133638	-19.0	0.0	6.0	0
3	2018-01-03 15:24:56	-20.250612	-70.124635	103.0	0.0	6.0	0
4	2018-01-03 15:26:00	-20.250654	-70.124636	114.0	0.0	8.0	1
5	2018-01-03 15:26:27	-20.250654	-70.124636	114.0	0.0	8.0	0

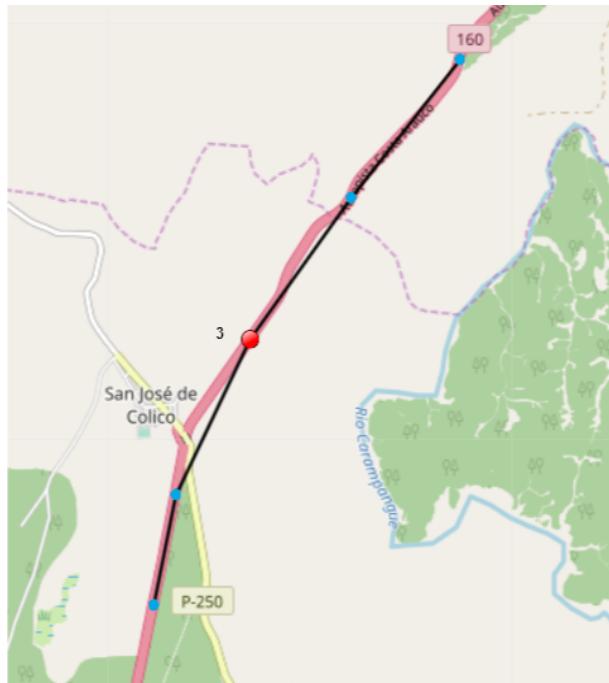
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones y no tienen cambios bruscos de velocidad, sin embargo la observación 5 es detectada como outlier.



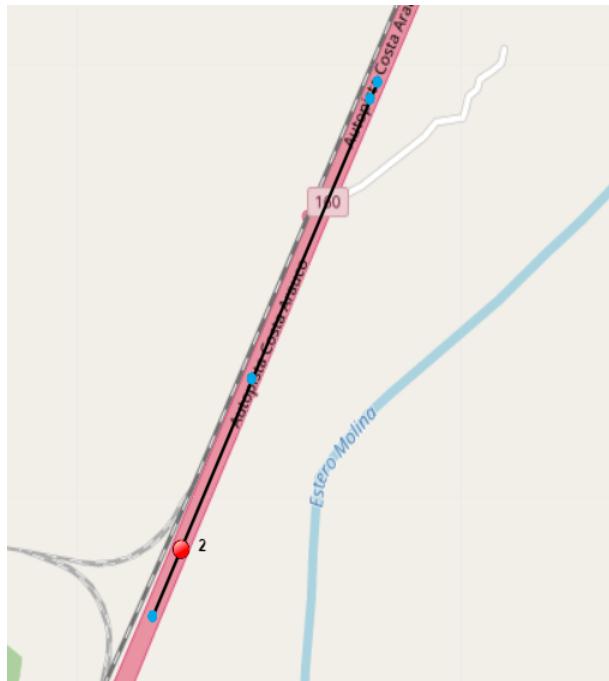
	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2018-01-18 13:46:34	-37.462286	-73.367607	135.0	16.0	8.0	0
2	2018-01-18 13:46:35	-37.462286	-73.367607	135.0	16.0	8.0	0
3	2018-01-18 13:46:46	-37.460858	-73.366098	127.0	101.0	8.0	0
4	2018-01-18 13:47:26	-37.449261	-73.358563	155.0	146.0	9.0	0
5	2018-01-18 13:47:35	-37.446080	-73.357183	157.0	154.0	8.0	1

Este trayecto muestra una continuidad normal en todas las observaciones y no tienen cambios bruscos de velocidad, sin embargo la observación 3 es detectada como outlier..



	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2017-12-30 13:12:04	-37.384083	-73.339738	143.0	5.0	7.0	0
2	2017-12-30 13:12:44	-37.374742	-73.337310	141.0	143.0	5.0	0
3	2017-12-30 13:13:24	-37.361657	-73.329252	131.0	167.0	11.0	1
4	2017-12-30 13:14:04	-37.349851	-73.318316	114.0	131.0	9.0	0
5	2017-12-30 13:14:44	-37.338195	-73.306443	104.0	133.0	11.0	0

Este trayecto muestra una continuidad normal en todas las observaciones y no tienen cambios bruscos de velocidad, sin embargo la observación 2 es detectada como outlier..

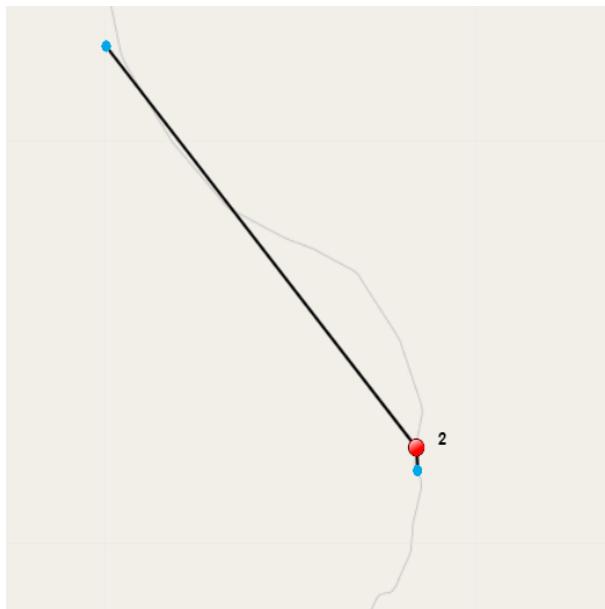


	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2018-02-07 18:24:08	-37.206523	-73.210755	12.0	147.0	10.0	0
2	2018-02-07 18:24:12	-37.205146	-73.209945	12.0	153.0	9.0	1
3	2018-02-07 18:24:22	-37.201688	-73.207907	10.0	149.0	10.0	0
4	2018-02-07 18:24:39	-37.196023	-73.204515	10.0	151.0	10.0	0
5	2018-02-07 18:24:40	-37.195682	-73.204310	10.0	151.0	10.0	0

B.0.6. Dataset 3(Autoencoder convolucional apilado)

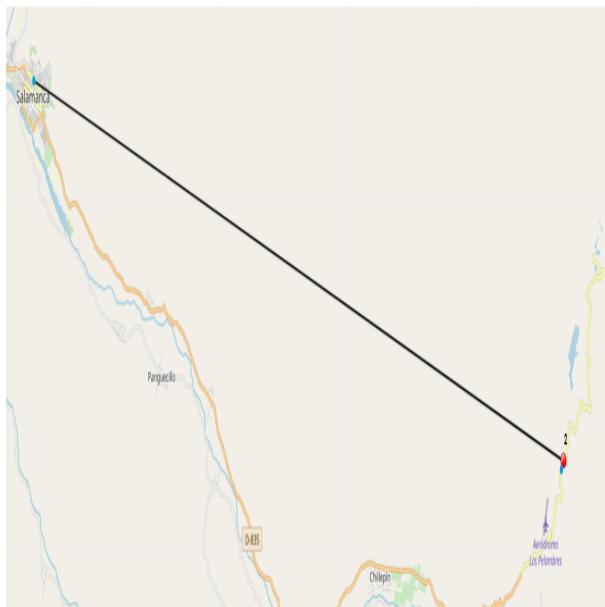
Ejemplos de trayectos outlier

Este trayecto muestra una discontinuidad evidente, las observación 2 sale de la continuidad de las demás observaciones.



	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2018-11-15 11:25:51	-24.272770	-69.359488	2097.0	71.0	7.0	0
2	2018-11-15 11:26:17	-24.268080	-69.359843	2077.0	74.0	9.0	1
3	2018-12-24 18:47:17	-24.188046	-69.433201	1738.0	0.0	13.0	0
4	2018-12-24 20:35:58	-24.187990	-69.433211	1741.0	0.0	11.0	0
5	2018-12-24 20:50:08	-24.187990	-69.433211	1741.0	0.0	11.0	0

Este trayecto muestra una discontinuidad evidente, las observación 2 sale de la continuidad de las demás observaciones.



	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2017-06-06 19:25:26	-31.860540	-70.606580	1344.0	46.0	7.0	0
2	2017-06-06 19:25:49	-31.858513	-70.605276	1360.0	45.0	7.0	1
3	2017-08-07 17:02:11	-31.777200	-70.964386	513.0	0.0	7.0	0
4	2017-08-07 17:13:13	-31.777200	-70.964386	513.0	0.0	11.0	0
5	2017-08-07 17:13:18	-31.777194	-70.964395	513.0	3.0	11.0	0

Este trayecto muestra una continuidad normal, sin embargo la observación 2 muestra un cambio brusco de velocidad con respecto a la anterior observación y luego regresa a la velocidad promedio.



	fecha	latitud	longitud	altura	velocidad	nsat	outlier
1	2018-01-18 13:56:45	-37.299645	-73.261120	2.0	131.0	9.0	0
2	2018-01-18 13:57:25	-37.285501	-73.251615	8.0	176.0	9.0	1
3	2018-01-18 13:58:05	-37.270796	-73.243498	6.0	133.0	9.0	0
4	2018-01-18 13:58:45	-37.256138	-73.237983	10.0	133.0	9.0	0
5	2018-01-18 13:58:47	-37.255475	-73.237985	10.0	132.0	9.0	0

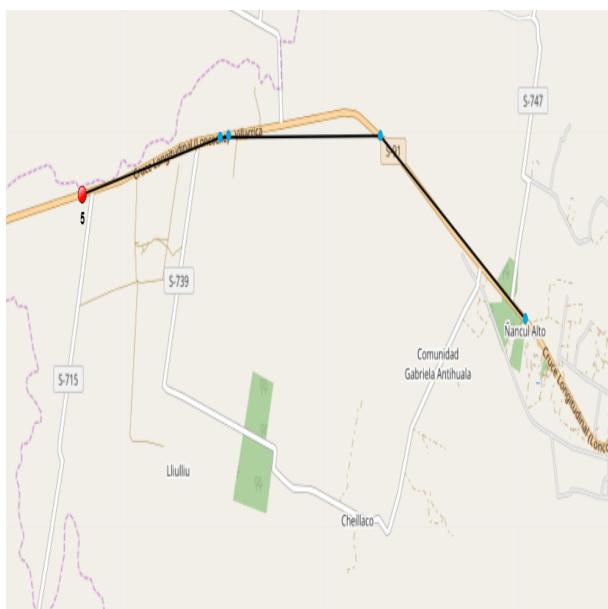
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, la observación 5 no tiene un cambio brusco de velocidad sin embargo es detectada como outlier.



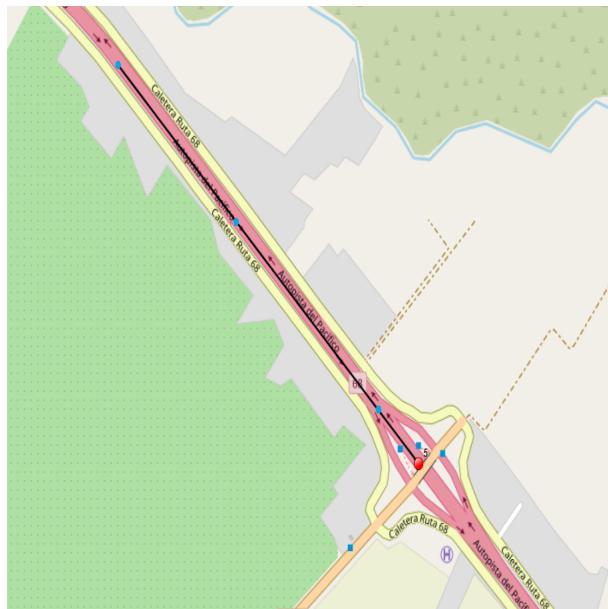
	fecha	latitud	longitud	altura	velocidad	nsat	outlier
1	2018-01-31 17:13:42	-37.312838	-73.269565	55.0	52.0	7.0	0
2	2018-01-31 17:14:22	-37.308192	-73.265601	14.0	60.0	8.0	0
3	2018-01-31 17:15:02	-37.302866	-73.262970	7.0	84.0	9.0	0
4	2018-01-31 17:15:42	-37.293496	-73.256845	5.0	88.0	9.0	0
5	2018-01-31 17:16:22	-37.281088	-73.248765	5.0	165.0	9.0	1

Este trayecto muestra una continuidad normal en todas las observaciones, la observación 5 no tiene un cambio brusco de velocidad sin embargo es detectada como outlier.



	fecha	latitud	longitud	altura	velocidad	nsat	outlier
1	2017-08-10 09:26:57	-39.274353	-72.333182	309.0	130.0	10.0	0
2	2017-08-10 09:27:38	-39.266404	-72.349480	267.0	148.0	10.0	0
3	2017-08-10 09:28:18	-39.266439	-72.366582	250.0	154.0	10.0	0
4	2017-08-10 09:28:19	-39.266510	-72.367568	250.0	153.0	10.0	0
5	2017-08-10 09:28:51	-39.268953	-72.383093	236.0	154.0	10.0	1

Este trayecto muestra una continuidad normal en todas las observaciones, la observación 5 no tiene un cambio brusco de velocidad sin embargo es detectada como outlier.

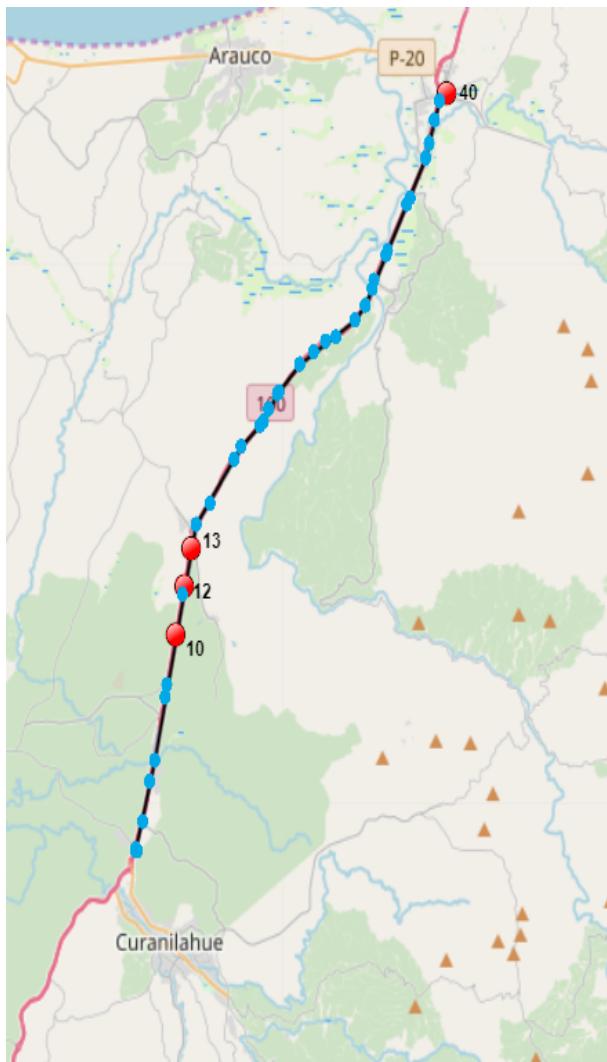


	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2017-12-26 14:17:12	-33.414384	-71.053501	173.0	135.0	16.0	0
2	2017-12-26 14:17:13	-33.414384	-71.053501	173.0	135.0	16.0	0
3	2017-12-26 14:17:23	-33.416903	-71.050458	172.0	150.0	16.0	0
4	2017-12-26 14:17:33	-33.419896	-71.046786	174.0	161.0	16.0	0
5	2017-12-26 14:17:36	-33.420746	-71.045752	174.0	160.0	16.0	1

B.0.7. Dataset 4(Autoencoder apilado)

Ejemplos de trayectos outlier

Este trayecto muestra una continuidad normal, sin embargo la observaciones 12,13 muestran un cambio brusco de velocidad, las observaciones 40 y 10 no muestran cambios de velocidad bruscos.



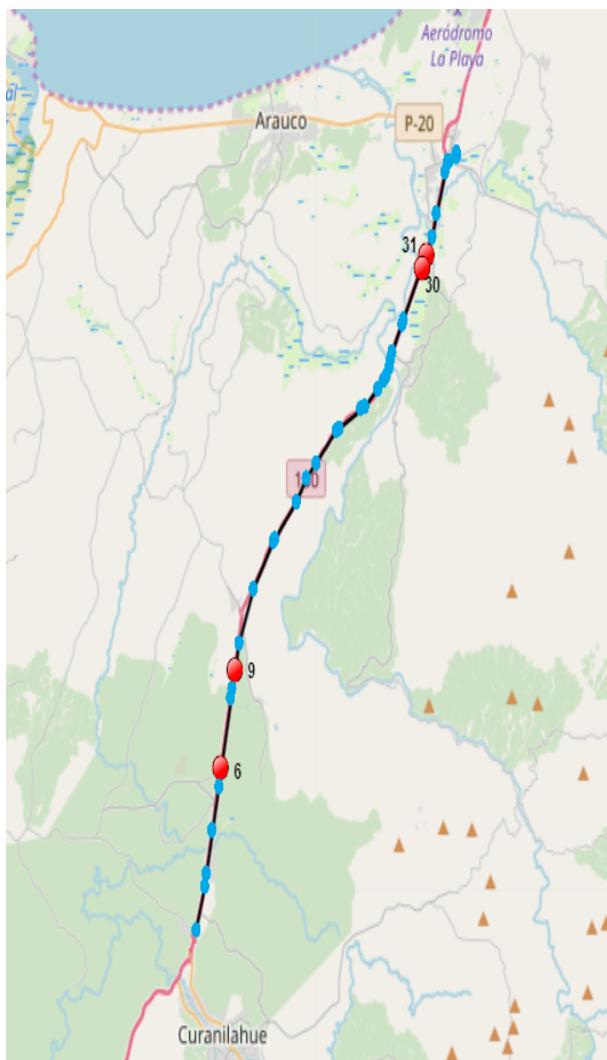
	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2018-01-30 16:51:46	-37.452448	-73.358682	160.0	52.0	8.0	0
2	2018-01-30 16:51:48	-37.452356	-73.359126	159.0	46.0	8.0	0
3	2018-01-30 16:51:49	-37.452273	-73.359238	159.0	47.0	8.0	0
4	2018-01-30 16:51:52	-37.451975	-73.359390	158.0	38.0	7.0	0
5	2018-01-30 16:52:26	-37.444843	-73.356671	168.0	138.0	8.0	0
6	2018-01-30 16:52:52	-37.434583	-73.353414	150.0	171.0	8.0	0
7	2018-01-30 16:53:06	-37.429013	-73.351921	140.0	175.0	9.0	0
8	2018-01-30 16:53:46	-37.412501	-73.347491	146.0	174.0	8.0	0
9	2018-01-30 16:53:52	-37.409638	-73.346693	148.0	158.0	8.0	0
10	2018-01-30 16:54:25	-37.396095	-73.343070	135.0	170.0	10.0	1
11	2018-01-30 16:54:52	-37.386078	-73.340378	145.0	130.0	7.0	0
12	2018-01-30 16:55:06	-37.383883	-73.339811	145.0	9.0	6.0	1
13	2018-01-30 16:55:46	-37.374137	-73.337248	153.0	142.0	9.0	1
14	2018-01-30 16:56:05	-37.367921	-73.334955	133.0	130.0	9.0	0
15	2018-01-30 16:56:26	-37.362467	-73.329955	139.0	138.0	9.0	0
16	2018-01-30 16:57:08	-37.351206	-73.320063	113.0	121.0	9.0	0
17	2018-01-30 16:57:19	-37.347847	-73.316925	118.0	122.0	11.0	0
18	2018-01-30 16:57:46	-37.342228	-73.309943	108.0	108.0	10.0	0
19	2018-01-30 16:57:51	-37.340963	-73.308518	108.0	115.0	11.0	0
20	2018-01-30 16:58:05	-37.338050	-73.306416	108.0	98.0	11.0	0
21	2018-01-30 16:58:23	-37.334140	-73.302673	86.0	118.0	11.0	0
22	2018-01-30 16:58:26	-37.333715	-73.302183	86.0	113.0	11.0	0
23	2018-01-30 16:59:05	-37.326227	-73.293918	70.0	105.0	9.0	0
24	2018-01-30 16:59:08	-37.326227	-73.293918	70.0	105.0	9.0	0
25	2018-01-30 16:59:27	-37.323106	-73.288455	88.0	92.0	11.0	0
26	2018-01-30 16:59:46	-37.320481	-73.283948	92.0	110.0	9.0	0
27	2018-01-30 16:59:59	-37.319251	-73.279705	93.0	103.0	9.0	0
28	2018-01-30 17:00:26	-37.315197	-73.271867	78.0	111.0	7.0	0
29	2018-01-30 17:00:45	-37.311198	-73.267630	47.0	97.0	7.0	0
30	2018-01-30 17:01:06	-37.306925	-73.265159	22.0	92.0	8.0	0
31	2018-01-30 17:01:17	-37.304777	-73.264161	18.0	79.0	9.0	0
32	2018-01-30 17:01:46	-37.297823	-73.259873	13.0	144.0	9.0	0
33	2018-01-30 17:01:49	-37.296845	-73.259015	12.0	149.0	9.0	0
34	2018-01-30 17:02:21	-37.284857	-73.251213	12.0	170.0	9.0	0
35	2018-01-30 17:02:25	-37.283340	-73.250225	11.0	170.0	9.0	0
36	2018-01-30 17:02:53	-37.273218	-73.244041	14.0	137.0	9.0	0
37	2018-01-30 17:03:08	-37.269015	-73.242728	10.0	154.0	9.0	0
38	2018-01-30 17:03:25	-37.262988	-73.240268	10.0	126.0	9.0	0
39	2018-01-30 17:03:46	-37.258025	-73.238221	11.0	76.0	9.0	0
40	2018-01-30 17:04:11	-37.255988	-73.235580	11.0	8.0	10.0	1

Este trayecto muestra una discontinuidad evidente, las observaciones 32,33,37,39 salen de la continuidad de las demás observaciones.



	fecha	latitud	longitud	altura	velocidad	nsat	outlier
1	2017-05-13 23:40:00	-34.186191	-70.743118	508.0	3.0	8.0	0
2	2017-05-13 23:40:01	-34.186195	-70.743103	508.0	3.0	7.0	0
3	2017-05-13 23:40:02	-34.186188	-70.743088	508.0	1.0	8.0	0
4	2017-05-13 23:40:03	-34.186188	-70.743088	508.0	1.0	8.0	0
5	2017-05-13 23:40:09	-34.186171	-70.743071	505.0	1.0	9.0	0
6	2017-05-13 23:40:16	-34.186186	-70.743081	508.0	1.0	10.0	0
7	2017-05-13 23:40:17	-34.186188	-70.743081	509.0	1.0	9.0	0
8	2017-05-13 23:40:23	-34.186181	-70.743083	508.0	3.0	10.0	0
9	2017-05-13 23:40:57	-34.185833	-70.743114	508.0	8.0	4.0	0
10	2017-05-13 23:41:24	-34.185206	-70.742998	504.0	0.0	11.0	0
11	2017-05-13 23:41:28	-34.185215	-70.742991	502.0	2.0	11.0	0
12	2017-05-13 23:41:31	-34.185198	-70.742918	502.0	10.0	10.0	0
13	2017-05-13 23:41:36	-34.185253	-70.742595	499.0	31.0	8.0	0
14	2017-05-13 23:42:16	-34.186081	-70.737138	501.0	47.0	10.0	0
15	2017-05-13 23:42:31	-34.186273	-70.735828	503.0	0.0	11.0	0
16	2017-05-13 23:42:56	-34.186273	-70.735828	503.0	0.0	11.0	0
17	2017-05-13 23:43:32	-34.186816	-70.731778	506.0	57.0	11.0	0
18	2017-05-13 23:43:36	-34.186930	-70.730910	507.0	58.0	10.0	0
19	2017-05-13 23:43:53	-34.187266	-70.728660	508.0	19.0	11.0	0
20	2017-05-13 23:43:56	-34.187408	-70.728626	508.0	21.0	8.0	0
21	2017-05-13 23:44:05	-34.188143	-70.728540	508.0	40.0	11.0	0
22	2017-05-13 23:44:16	-34.189030	-70.727428	509.0	43.0	11.0	0
23	2017-05-13 23:44:52	-34.173478	-70.726928	517.0	81.0	9.0	0
24	2017-05-13 23:44:52	-34.173478	-70.726928	517.0	81.0	9.0	0
25	2017-05-13 23:44:57	-34.174268	-70.727225	516.0	83.0	9.0	0
26	2017-05-13 23:45:36	-34.180390	-70.735098	512.0	91.0	10.0	0
27	2017-05-13 23:45:52	-34.182906	-70.738441	511.0	98.0	10.0	0
28	2017-05-13 23:45:56	-34.183551	-70.739315	508.0	97.0	10.0	0
29	2017-05-13 23:46:16	-34.186695	-70.743608	513.0	93.0	8.0	0
30	2017-05-13 23:46:22	-34.588081	-70.990638	338.0	0.0	8.0	0
31	2017-05-13 23:46:23	-34.588081	-70.990638	338.0	0.0	8.0	0
32	2017-05-13 23:46:56	-34.191395	-70.753501	517.0	94.0	11.0	1
33	2017-05-13 23:47:23	-34.588081	-70.990638	338.0	0.0	10.0	1
34	2017-05-13 23:47:37	-34.199833	-70.759638	513.0	99.0	11.0	0
35	2017-05-13 23:47:56	-34.204093	-70.762908	512.0	102.0	11.0	0
36	2017-05-13 23:48:16	-34.208226	-70.766430	512.0	100.0	11.0	0
37	2017-05-13 23:48:23	-34.588081	-70.990638	338.0	0.0	9.0	1
38	2017-05-13 23:48:56	-34.216669	-70.773638	502.0	104.0	11.0	0
39	2017-05-13 23:49:23	-34.588081	-70.990638	338.0	0.0	10.0	1
40	2017-05-13 23:49:37	-34.225573	-70.781238	491.0	107.0	11.0	0

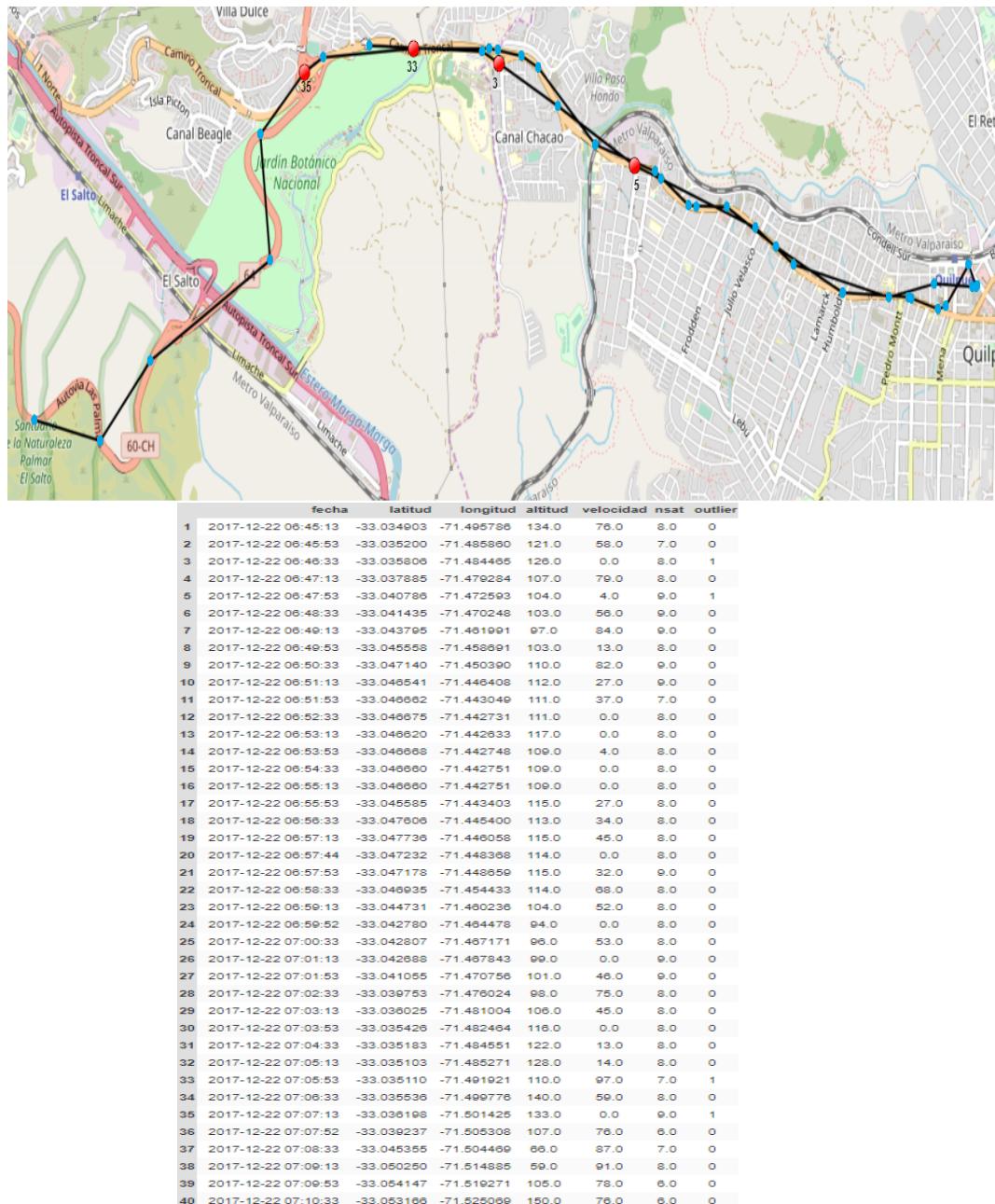
Este trayecto muestra una continuidad normal, sin embargo las observaciones 6,30,31 no muestran un cambio brusco de velocidad, la observación 9 muestra un cambio de velocidad con respecto a sus observaciones adyacentes.



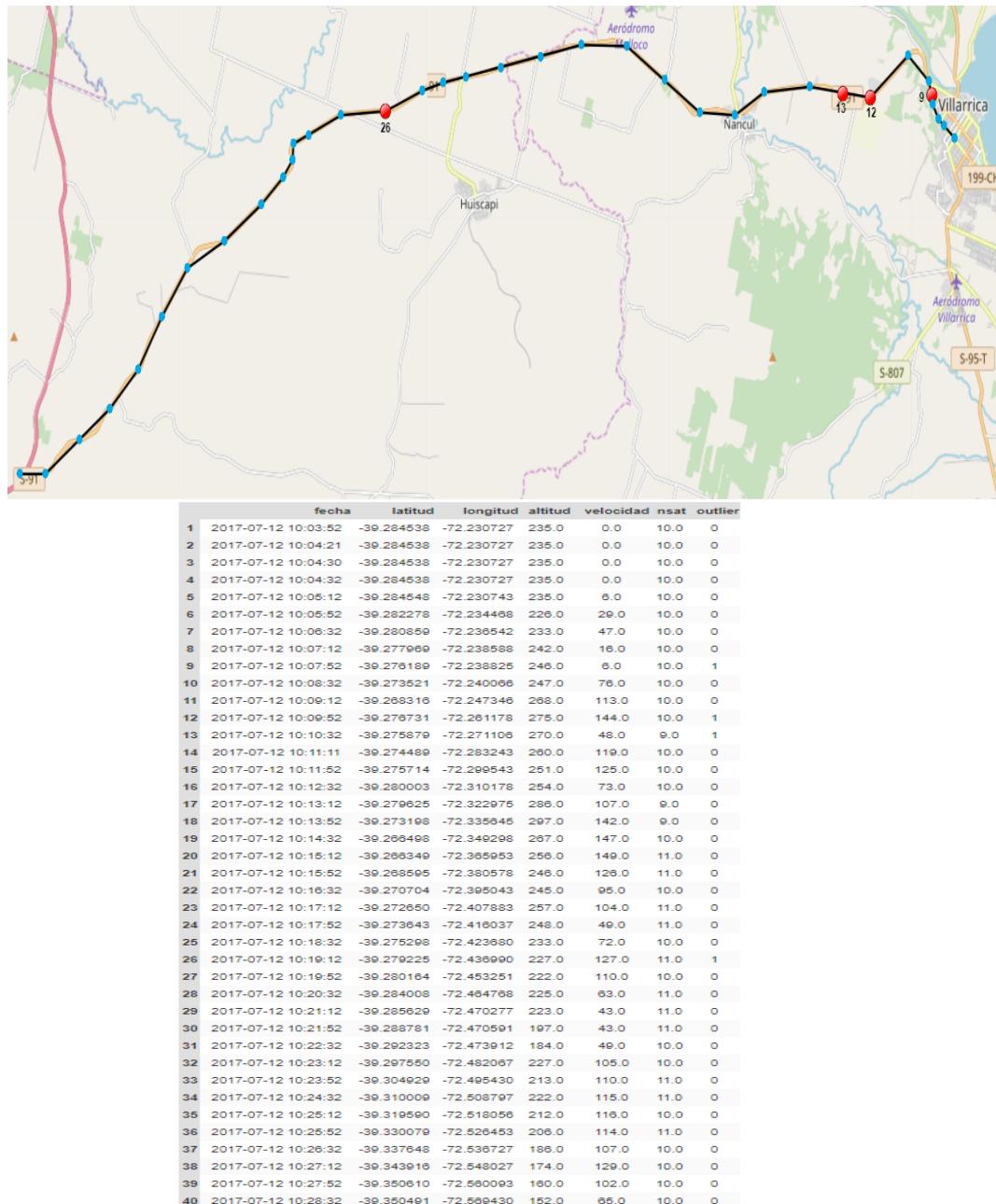
	fecha	latitud	longitude	altitud	velocidad	nsat	outlier
1	2018-01-26 16:37:49	-37.449772	-73.358738	155.0	78.0	7.0	0
2	2018-01-26 16:38:29	-37.438843	-73.354538	154.0	150.0	7.0	0
3	2018-01-26 16:38:36	-37.435833	-73.353742	151.0	153.0	7.0	0
4	2018-01-26 16:39:09	-37.424405	-73.350718	144.0	136.0	7.0	0
5	2018-01-26 16:39:36	-37.413740	-73.347803	140.0	166.0	7.0	0
6	2018-01-26 16:39:49	-37.408760	-73.346458	141.0	166.0	7.0	1
7	2018-01-26 16:40:29	-37.391576	-73.341825	132.0	157.0	6.0	0
8	2018-01-26 16:40:37	-37.388996	-73.341120	134.0	123.0	7.0	0
9	2018-01-26 16:41:09	-37.384253	-73.339782	141.0	6.0	6.0	1
10	2018-01-26 16:41:48	-37.377618	-73.338131	144.0	140.0	6.0	0
11	2018-01-26 16:42:29	-37.364035	-73.331258	141.0	143.0	6.0	0
12	2018-01-26 16:43:09	-37.352013	-73.321602	123.0	147.0	8.0	0
13	2018-01-26 16:43:11	-37.351571	-73.320905	128.0	140.0	8.0	0
14	2018-01-26 16:43:49	-37.342386	-73.310123	114.0	148.0	7.0	0
15	2018-01-26 16:44:12	-37.336513	-73.305368	101.0	123.0	7.0	0
16	2018-01-26 16:44:29	-37.332490	-73.300910	87.0	116.0	8.0	0
17	2018-01-26 16:45:09	-37.324055	-73.299045	97.0	123.0	7.0	0
18	2018-01-26 16:45:12	-37.323513	-73.289990	101.0	123.0	8.0	0
19	2018-01-26 16:45:44	-37.319078	-73.279258	91.0	141.0	7.0	0
20	2018-01-26 16:45:49	-37.318426	-73.277223	88.0	137.0	7.0	0
21	2018-01-26 16:46:12	-37.313891	-73.270918	68.0	118.0	7.0	0
22	2018-01-26 16:46:22	-37.311956	-73.268436	57.0	100.0	6.0	0
23	2018-01-26 16:46:29	-37.310675	-73.267228	48.0	83.0	8.0	0
24	2018-01-26 16:46:48	-37.308421	-73.265678	33.0	45.0	8.0	0
25	2018-01-26 16:46:49	-37.308316	-73.265648	31.0	44.0	8.0	0
26	2018-01-26 16:47:09	-37.306072	-73.264908	20.0	51.0	8.0	0
27	2018-01-26 16:47:20	-37.304630	-73.264112	20.0	63.0	8.0	0
28	2018-01-26 16:47:49	-37.297437	-73.259458	28.0	148.0	9.0	0
29	2018-01-26 16:47:52	-37.296423	-73.258792	27.0	153.0	9.0	0
30	2018-01-26 16:48:29	-37.283201	-73.250148	23.0	171.0	9.0	1
31	2018-01-26 16:48:38	-37.279816	-73.247935	22.0	165.0	9.0	1
32	2018-01-26 16:48:49	-37.275618	-73.245312	21.0	149.0	9.0	0
33	2018-01-26 16:49:09	-37.269538	-73.242985	21.0	135.0	8.0	0
34	2018-01-26 16:49:48	-37.259151	-73.238858	20.0	71.0	9.0	0
35	2018-01-26 16:50:16	-37.256595	-73.237348	15.0	3.0	9.0	0
36	2018-01-26 16:50:29	-37.256358	-73.236787	14.0	38.0	9.0	0
37	2018-01-26 16:51:00	-37.255246	-73.233560	9.0	30.0	9.0	0
38	2018-01-26 16:51:09	-37.254668	-73.233437	9.0	17.0	9.0	0
39	2018-01-26 16:51:10	-37.254668	-73.233437	9.0	17.0	9.0	0
40	2018-01-26 16:51:11	-37.254637	-73.233530	9.0	16.0	8.0	0

Ejemplos de trayectos no outlier

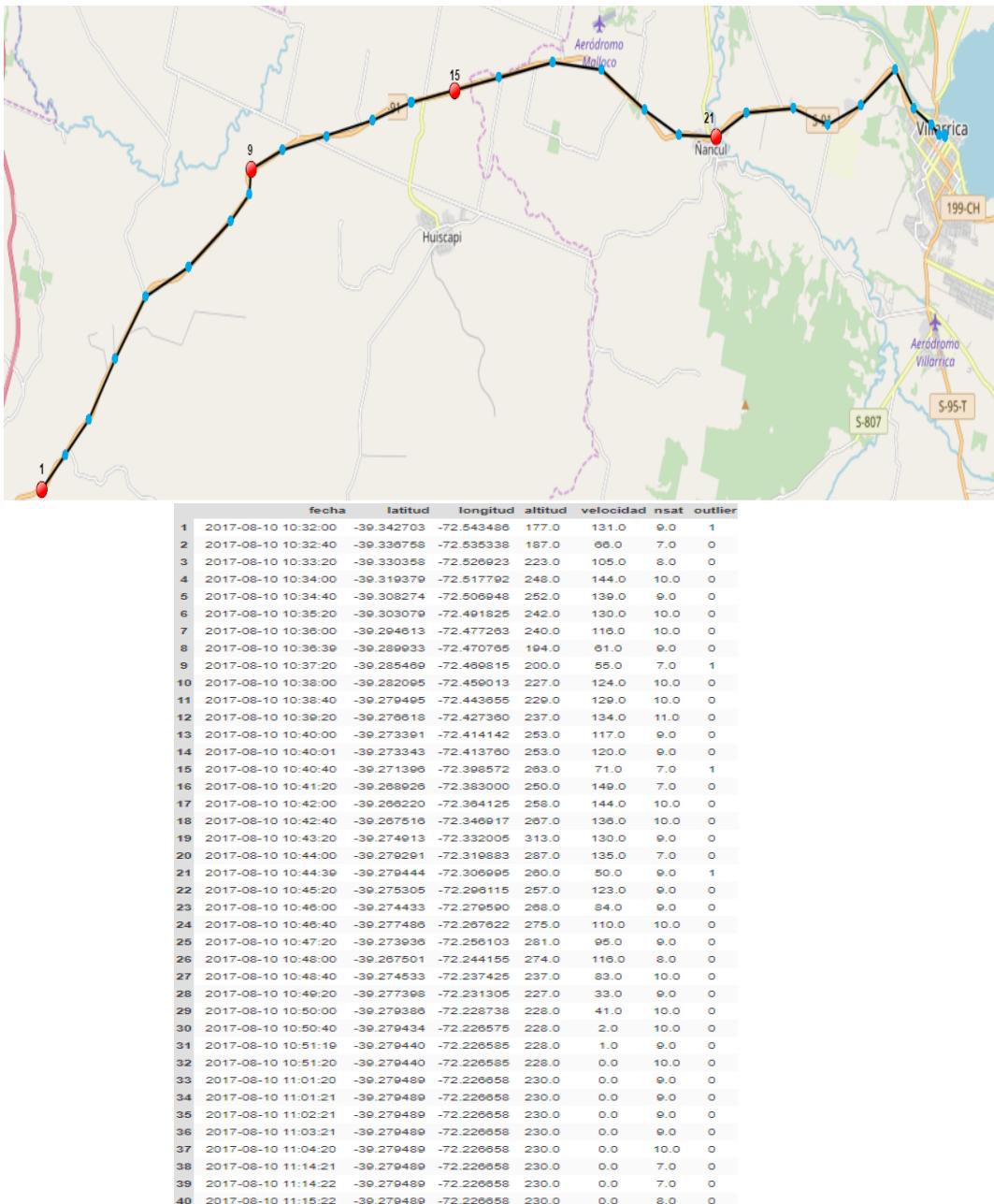
Este trayecto muestra una continuidad normal en todas las observaciones, las observaciones 3, 5,33 y 35 no presentan un cambio brusco en la velocidad, sin embargo son detectados como outlier.



Este trayecto muestra una continuidad normal en todas las observaciones, las observaciones 9, 12, 13 y 26 no presenten un cambio brusco en la velocidad, sin embargo son detectados como outlier.



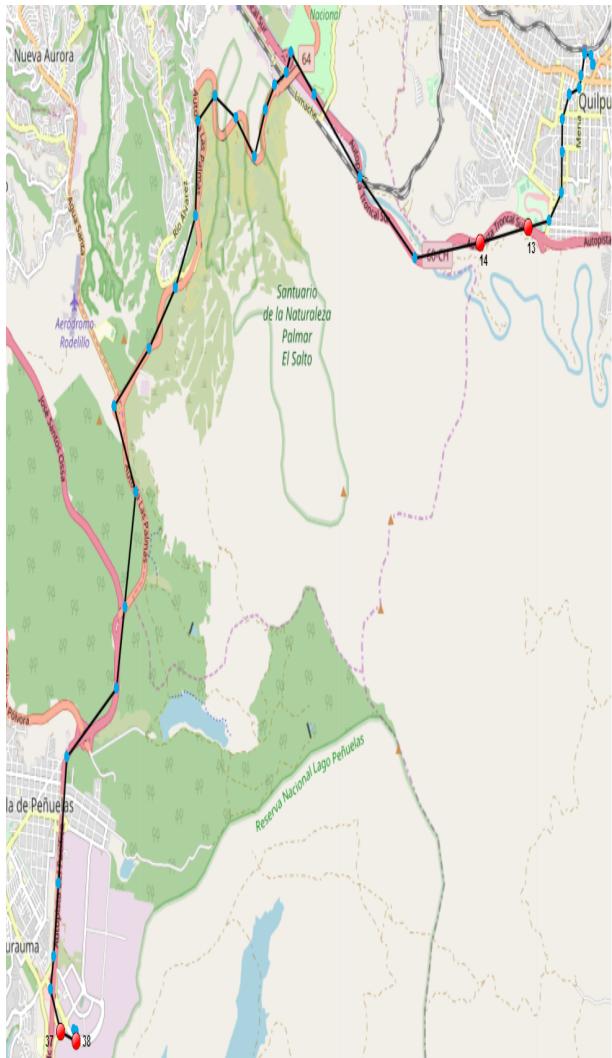
Este trayecto muestra una continuidad normal en todas las observaciones, las observaciones 1, 9, 15 y 21 no presentan un cambio brusco en la velocidad, sin embargo son detectados como outlier.



B.0.8. Dataset 4(Autoencoder convolucional apilado)

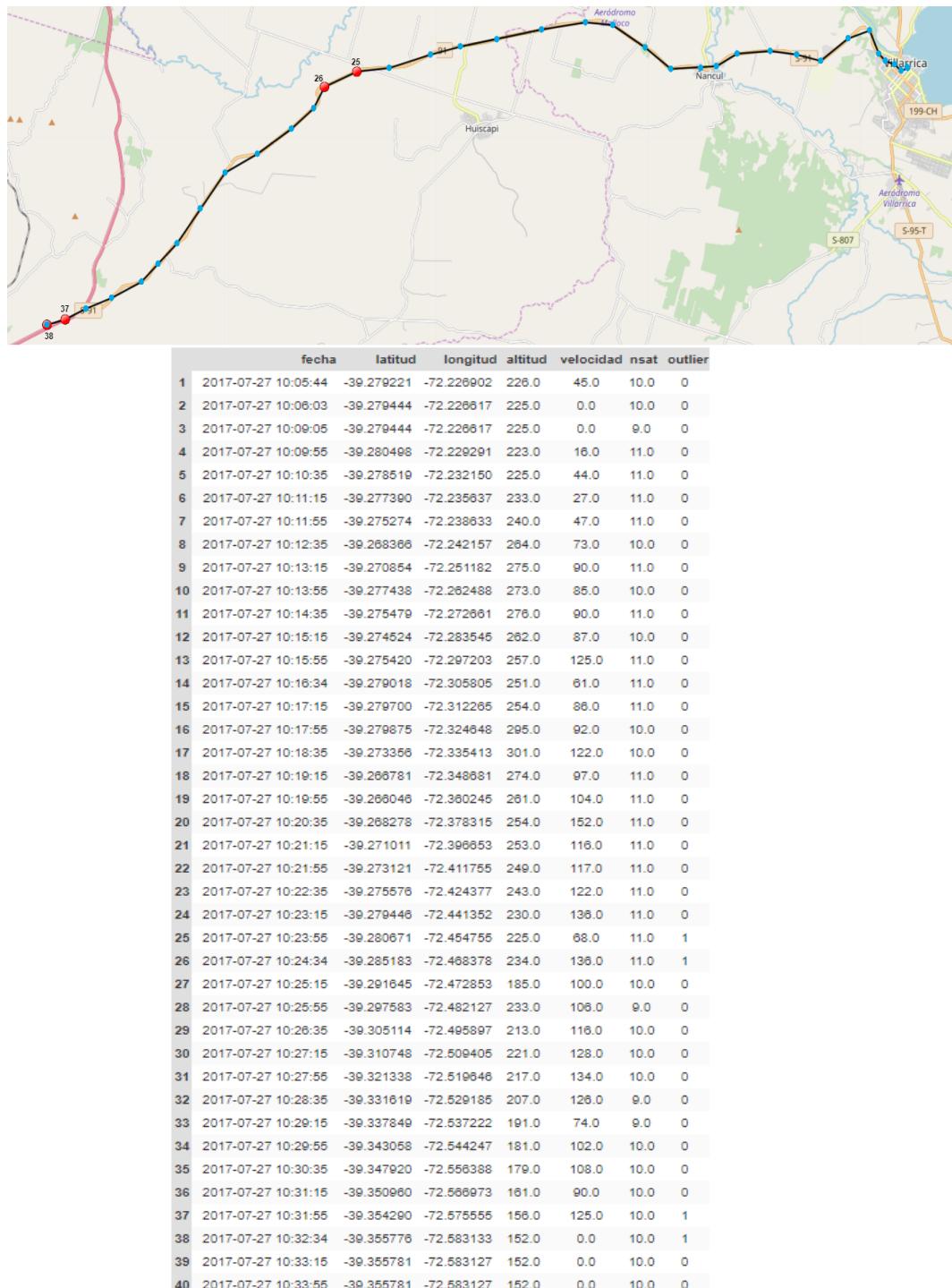
Ejemplos de trayectos outlier

Este trayecto muestra una continuidad normal, sin embargo las observaciones 13 y 14 muestran un cambio brusco de velocidad, las observaciones 37 y 38 no tiene cambios bruscos de velocidad.

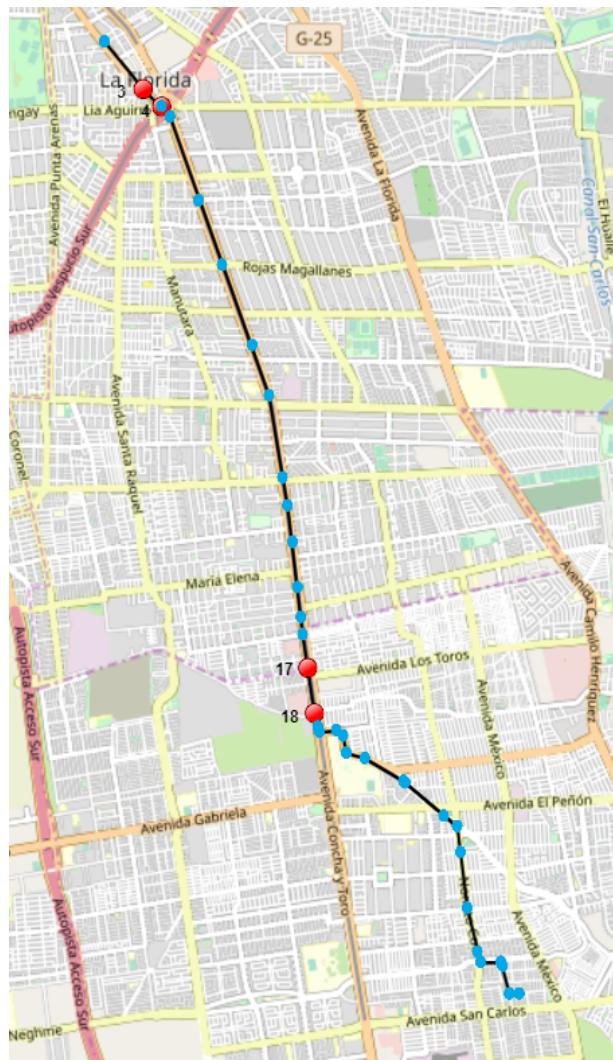


	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2017-11-27 19:16:17	-33.046605	-71.442841	106.0	0.0	8.0	0
2	2017-11-27 19:16:58	-33.046603	-71.442843	106.0	0.0	8.0	0
3	2017-11-27 19:17:38	-33.046601	-71.442864	107.0	4.0	9.0	0
4	2017-11-27 19:18:18	-33.045785	-71.443223	107.0	1.0	8.0	0
5	2017-11-27 19:18:58	-33.045587	-71.444578	106.0	20.0	9.0	0
6	2017-11-27 19:19:38	-33.047626	-71.445403	106.0	0.0	9.0	0
7	2017-11-27 19:20:18	-33.048748	-71.445886	107.0	42.0	9.0	0
8	2017-11-27 19:20:58	-33.049438	-71.448120	115.0	39.0	8.0	0
9	2017-11-27 19:21:38	-33.051738	-71.449583	134.0	44.0	9.0	0
10	2017-11-27 19:22:18	-33.054740	-71.449640	130.0	52.0	6.0	0
11	2017-11-27 19:22:58	-33.058506	-71.449836	148.0	53.0	8.0	0
12	2017-11-27 19:23:38	-33.061001	-71.452456	171.0	65.0	9.0	0
13	2017-11-27 19:24:18	-33.061580	-71.457108	179.0	6.0	9.0	1
14	2017-11-27 19:24:58	-33.063048	-71.467733	136.0	129.0	8.0	1
15	2017-11-27 19:25:38	-33.064483	-71.482185	67.0	127.0	9.0	0
16	2017-11-27 19:26:18	-33.056978	-71.494423	33.0	127.0	7.0	0
17	2017-11-27 19:26:58	-33.049416	-71.504508	21.0	132.0	8.0	0
18	2017-11-27 19:27:38	-33.045536	-71.509691	24.0	47.0	6.0	0
19	2017-11-27 19:28:18	-33.047370	-71.510648	26.0	12.0	8.0	0
20	2017-11-27 19:28:58	-33.048570	-71.513354	34.0	13.0	8.0	0
21	2017-11-27 19:29:38	-33.050892	-71.515288	50.0	17.0	6.0	0
22	2017-11-27 19:30:18	-33.055193	-71.517816	76.0	71.0	6.0	0
23	2017-11-27 19:30:58	-33.051545	-71.521873	124.0	64.0	7.0	0
24	2017-11-27 19:31:38	-33.049607	-71.526549	177.0	77.0	7.0	0
25	2017-11-27 19:32:18	-33.051895	-71.530338	224.0	90.0	7.0	0
26	2017-11-27 19:32:58	-33.060611	-71.530718	289.0	83.0	10.0	0
27	2017-11-27 19:33:38	-33.067203	-71.535281	314.0	82.0	10.0	0
28	2017-11-27 19:34:18	-33.072828	-71.541068	346.0	74.0	7.0	0
29	2017-11-27 19:34:58	-33.078127	-71.548748	337.0	63.0	9.0	0
30	2017-11-27 19:35:38	-33.086040	-71.544063	327.0	103.0	8.0	0
31	2017-11-27 19:36:18	-33.096720	-71.546318	326.0	95.0	8.0	0
32	2017-11-27 19:36:58	-33.104148	-71.548274	357.0	102.0	6.0	0
33	2017-11-27 19:37:38	-33.110526	-71.559335	352.0	113.0	8.0	0
34	2017-11-27 19:38:18	-33.122121	-71.561188	363.0	105.0	9.0	0
35	2017-11-27 19:38:58	-33.128957	-71.562173	347.0	27.0	9.0	0
36	2017-11-27 19:39:38	-33.131918	-71.562941	338.0	76.0	8.0	0
37	2017-11-27 19:40:18	-33.135760	-71.560643	330.0	7.0	9.0	1
38	2017-11-27 19:40:58	-33.136658	-71.557168	337.0	83.0	10.0	1
39	2017-11-27 19:41:38	-33.135802	-71.557268	338.0	20.0	7.0	0
40	2017-11-27 19:42:18	-33.135618	-71.557763	344.0	0.0	7.0	0

Este trayecto muestra una continuidad normal, sin embargo la observación 25 muestra un cambio velocidad no brusco pero es diferente a la velocidad promedio de sus observaciones adyacentes, las observaciones 37 y 38 muestran un cambio brusco de velocidad.



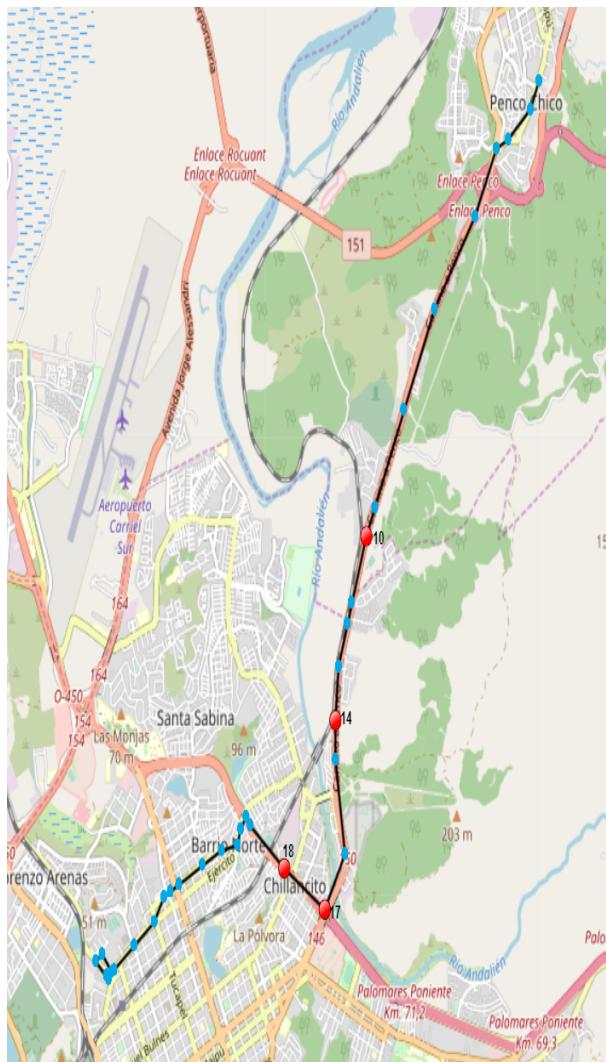
Este trayecto muestra una continuidad normal, sin embargo las observación 3 muestra un cambio velocidad no brusco pero es diferente a la velocidad promedio de sus observaciones adyacentes, lo mismo sucede con la observación 18.



	fecha	latitud	longitud	altitud	velocidad	nsat	outlier
1	2018-01-09 16:54:08	-33.517421	-70.604920	587.0	0.0	9.0	0
2	2018-01-09 16:54:10	-33.517421	-70.604920	587.0	1.0	9.0	0
3	2018-01-09 16:54:48	-33.521178	-70.600866	594.0	71.0	9.0	1
4	2018-01-09 16:55:28	-33.522587	-70.598924	600.0	0.0	9.0	1
5	2018-01-09 16:55:55	-33.522587	-70.598924	600.0	0.0	9.0	0
6	2018-01-09 16:56:08	-33.523566	-70.598071	602.0	58.0	9.0	0
7	2018-01-09 16:56:48	-33.530328	-70.595141	606.0	67.0	9.0	0
8	2018-01-09 16:57:28	-33.535670	-70.592810	613.0	68.0	8.0	0
9	2018-01-09 16:58:08	-33.542315	-70.589793	617.0	61.0	9.0	0
10	2018-01-09 16:58:48	-33.546370	-70.587949	622.0	69.0	8.0	0
11	2018-01-09 16:59:28	-33.553148	-70.586563	632.0	27.0	9.0	0
12	2018-01-09 17:00:08	-33.555445	-70.586174	634.0	68.0	9.0	0
13	2018-01-09 17:00:48	-33.558413	-70.585653	638.0	58.0	9.0	0
14	2018-01-09 17:01:28	-33.562181	-70.585151	641.0	0.0	9.0	0
15	2018-01-09 17:02:08	-33.564560	-70.584780	644.0	51.0	9.0	0
16	2018-01-09 17:02:48	-33.565946	-70.584544	645.0	24.0	9.0	0
17	2018-01-09 17:03:28	-33.568656	-70.584020	652.0	0.0	9.0	1
18	2018-01-09 17:04:08	-33.572412	-70.583370	658.0	59.0	8.0	1
19	2018-01-09 17:04:48	-33.573558	-70.583123	657.0	0.0	10.0	0
20	2018-01-09 17:05:28	-33.573558	-70.583123	657.0	0.0	10.0	0
21	2018-01-09 17:05:52	-33.574075	-70.582884	658.0	24.0	9.0	0
22	2018-01-09 17:06:08	-33.573895	-70.581073	661.0	40.0	9.0	0
23	2018-01-09 17:06:14	-33.574346	-70.580425	662.0	44.0	9.0	0
24	2018-01-09 17:06:26	-33.575660	-70.580078	664.0	37.0	9.0	0
25	2018-01-09 17:06:48	-33.576158	-70.578305	667.0	15.0	9.0	0
26	2018-01-09 17:07:27	-33.578032	-70.574301	669.0	65.0	9.0	0
27	2018-01-09 17:07:28	-33.578146	-70.574158	669.0	65.0	9.0	0
28	2018-01-09 17:08:08	-33.580823	-70.570220	674.0	37.0	9.0	0
29	2018-01-09 17:08:27	-33.581722	-70.568813	677.0	6.0	9.0	0
30	2018-01-09 17:08:28	-33.581803	-70.568768	677.0	16.0	9.0	0
31	2018-01-09 17:08:48	-33.583926	-70.568447	680.0	56.0	9.0	0
32	2018-01-09 17:09:27	-33.588481	-70.567690	684.0	0.0	9.0	0
33	2018-01-09 17:09:29	-33.588508	-70.567683	685.0	9.0	9.0	0
34	2018-01-09 17:10:08	-33.592030	-70.566688	691.0	52.0	9.0	0
35	2018-01-09 17:10:19	-33.592855	-70.566406	692.0	16.0	9.0	0
36	2018-01-09 17:10:43	-33.592858	-70.564251	694.0	18.0	9.0	0
37	2018-01-09 17:10:48	-33.593103	-70.564183	693.0	28.0	9.0	0
38	2018-01-09 17:11:28	-33.595473	-70.563460	697.0	11.0	9.0	0
39	2018-01-09 17:11:29	-33.595483	-70.563428	697.0	11.0	9.0	0
40	2018-01-09 17:12:08	-33.595413	-70.562500	700.0	0.0	9.0	0

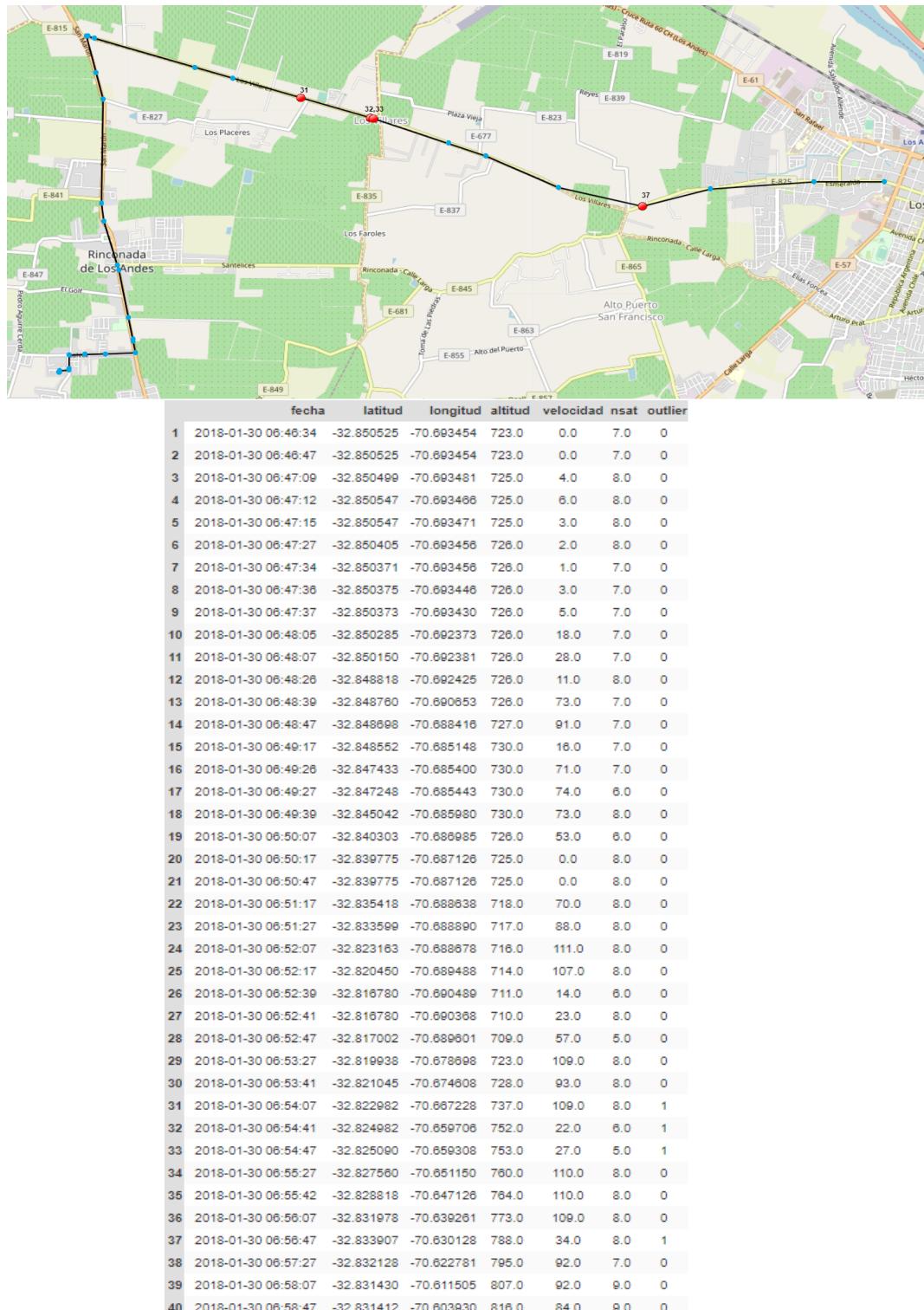
Ejemplos de trayectos no outlier

Este trayecto muestra una continuidad normal en todas las observaciones, las observaciones 10, 14, 17 y 18 no presentan un cambio brusco en la velocidad, sin embargo son detectados como outlier.



	fecha	latitud	longitud	altura	velocidad	nsat	outlier
1	2017-05-09 11:47:33	-36.746591	-72.993072	52.0	0.0	9.0	0
2	2017-05-09 11:48:13	-36.748847	-72.994397	68.0	43.0	10.0	0
3	2017-05-09 11:48:53	-36.751135	-72.997955	59.0	26.0	10.0	0
4	2017-05-09 11:49:33	-36.751847	-72.99787	53.0	0.0	10.0	0
5	2017-05-09 11:50:13	-36.751823	-72.99788	52.0	11.0	10.0	0
6	2017-05-09 11:50:53	-36.757086	-73.003353	45.0	84.0	9.0	0
7	2017-05-09 11:51:33	-36.764230	-73.009803	20.0	85.0	9.0	0
8	2017-05-09 11:52:13	-36.771830	-73.014741	12.0	83.0	10.0	0
9	2017-05-09 11:52:53	-36.779398	-73.019456	10.0	83.0	10.0	0
10	2017-05-09 11:53:33	-36.781688	-73.020853	10.0	0.0	10.0	1
11	2017-05-09 11:54:13	-36.786787	-73.023143	9.0	65.0	10.0	0
12	2017-05-09 11:54:53	-36.788342	-73.023815	7.0	45.0	10.0	0
13	2017-05-09 11:55:33	-36.791627	-73.025205	5.0	0.0	10.0	0
14	2017-05-09 11:56:13	-36.795875	-73.025795	7.0	76.0	10.0	1
15	2017-05-09 11:56:53	-36.798817	-73.025823	9.0	59.0	10.0	0
16	2017-05-09 11:57:33	-36.806133	-73.024211	8.0	80.0	10.0	0
17	2017-05-09 11:58:13	-36.810448	-73.027518	11.0	0.0	11.0	1
18	2017-05-09 11:58:53	-36.807272	-73.033968	16.0	71.0	11.0	1
19	2017-05-09 11:59:33	-36.803902	-73.036528	18.0	48.0	11.0	0
20	2017-05-09 12:00:13	-36.803275	-73.040158	20.0	0.0	11.0	0
21	2017-05-09 12:00:53	-36.804183	-73.041078	21.0	19.0	11.0	0
22	2017-05-09 12:01:33	-36.805405	-73.041563	21.0	0.0	11.0	0
23	2017-05-09 12:02:13	-36.805801	-73.044217	27.0	0.0	11.0	0
24	2017-05-09 12:02:52	-36.806986	-73.047261	26.0	53.0	11.0	0
25	2017-05-09 12:03:33	-36.808486	-73.050956	24.0	15.0	12.0	0
26	2017-05-09 12:04:13	-36.808518	-73.051053	25.0	0.0	11.0	0
27	2017-05-09 12:04:53	-36.809030	-73.052308	26.0	0.0	12.0	0
28	2017-05-09 12:05:33	-36.809478	-73.053452	27.0	25.0	12.0	0
29	2017-05-09 12:06:13	-36.811398	-73.054890	22.0	24.0	11.0	0
30	2017-05-09 12:06:53	-36.813076	-73.058229	19.0	18.0	12.0	0
31	2017-05-09 12:07:33	-36.815108	-73.061468	14.0	22.0	11.0	0
32	2017-05-09 12:08:13	-36.815256	-73.061812	16.0	25.0	12.0	0
33	2017-05-09 12:08:53	-36.813805	-73.063313	16.0	23.0	11.0	0
34	2017-05-09 12:09:33	-36.814319	-73.064418	12.0	33.0	12.0	0
35	2017-05-09 12:10:13	-36.815491	-73.062621	12.0	8.0	12.0	0
36	2017-05-09 12:10:49	-36.815713	-73.062295	12.0	0.0	12.0	0
37	2017-05-09 12:10:52	-36.815713	-73.062295	12.0	0.0	11.0	0
38	2017-05-09 12:20:53	-36.815713	-73.062295	12.0	0.0	10.0	0
39	2017-05-09 12:20:54	-36.815713	-73.062295	12.0	0.0	10.0	0
40	2017-05-09 12:21:54	-36.815746	-73.062298	15.0	0.0	11.0	0

Este trayecto muestra una continuidad normal en todas las observaciones, las observaciones 31, 32, 33 y 37 no presenten un cambio brusco en la velocidad, sin embargo son detectados como outlier.



Apéndice C

Arquitecturas de redes neuronales

C.1. Dataset 1

C.1.1. Autoencoder apilado

Arquitectura A

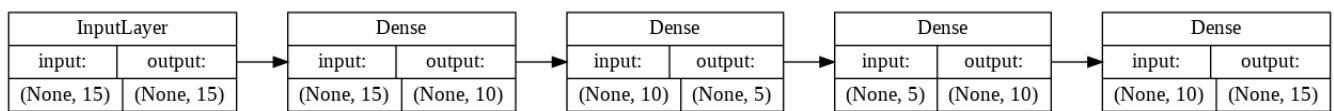


Figura C.1: Arquitectura A

Arquitectura B

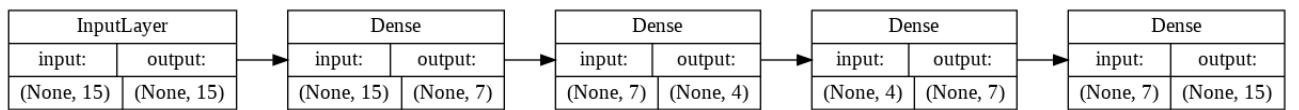


Figura C.2: Arquitectura B

Arquitectura C

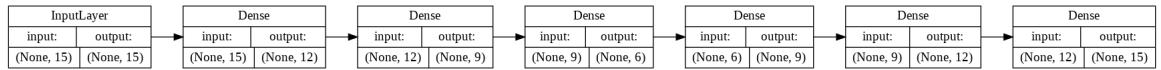


Figura C.3: Arquitectura C

C.1.2. Autoencoder convolucional apilado

Arquitectura A

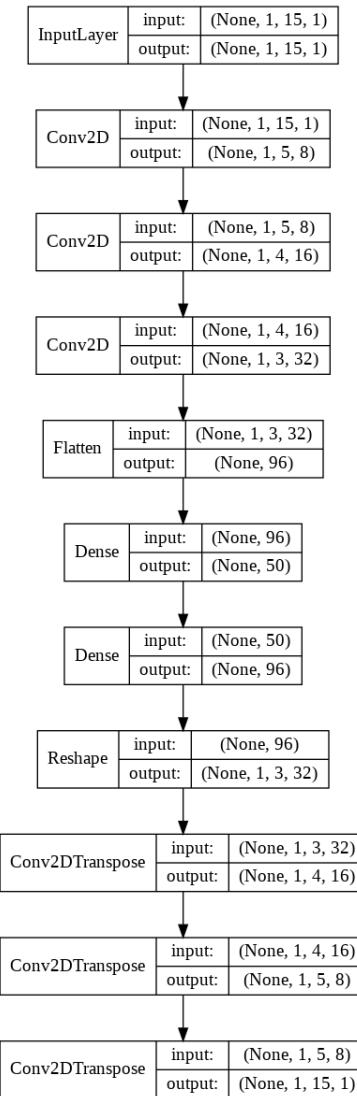


Figura C.4: Arquitectura A

Arquitectura B

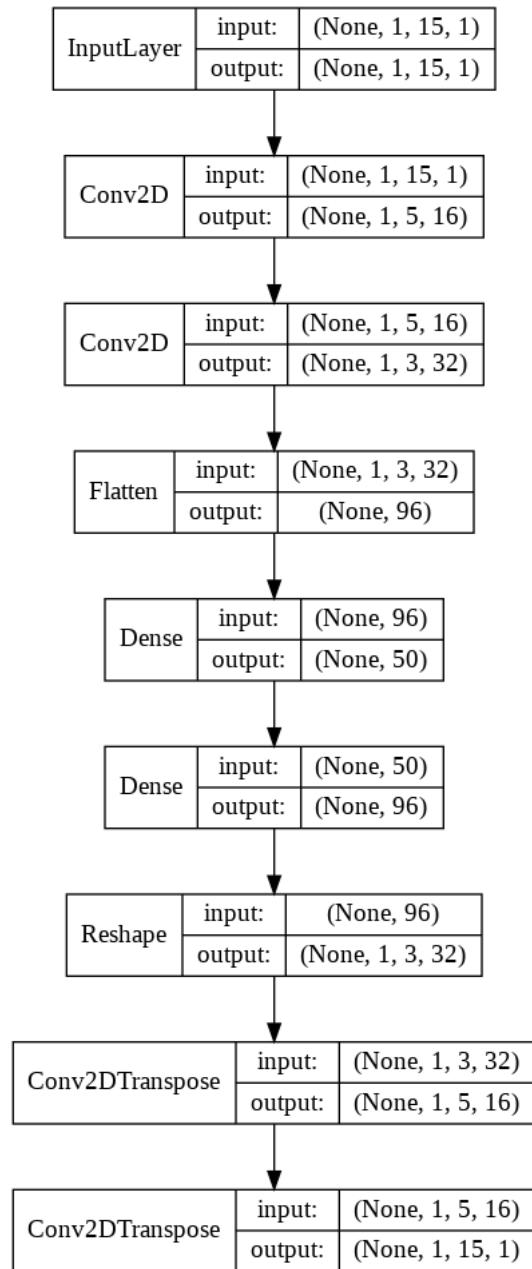


Figura C.5: Arquitectura B

Arquitectura C

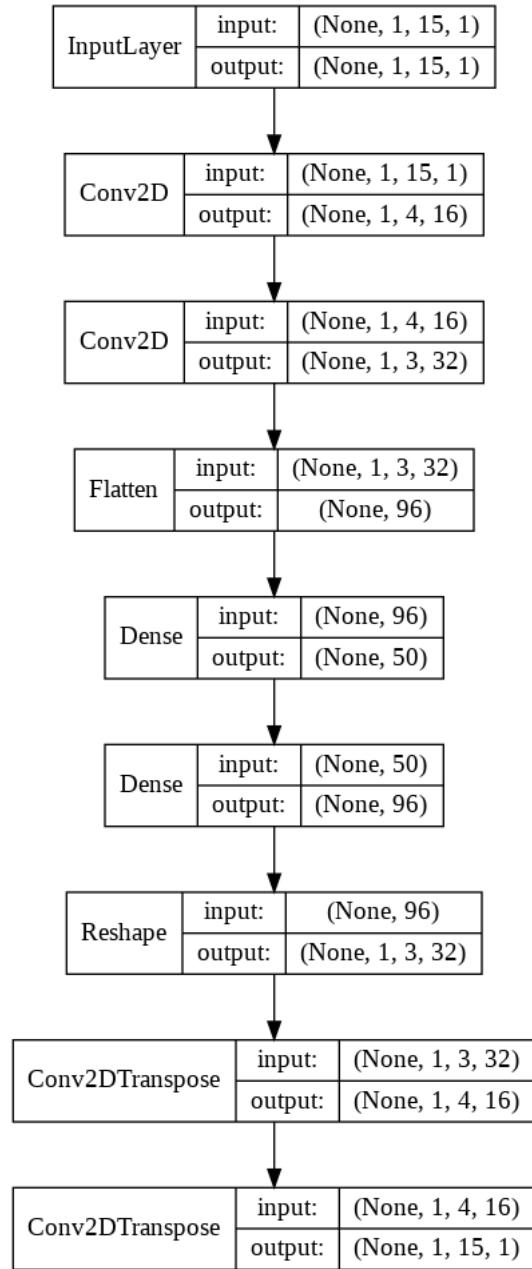


Figura C.6: Arquitectura C

C.2. Dataset 2

C.2.1. Autoencoder apilado

Arquitectura A

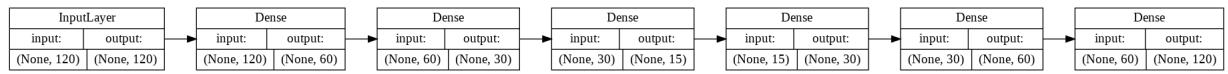


Figura C.7: Arquitectura A

Arquitectura B

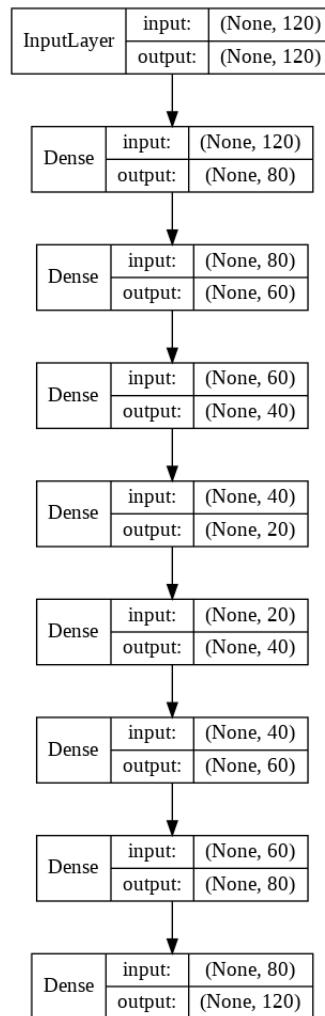


Figura C.8: Arquitectura B

Arquitectura C

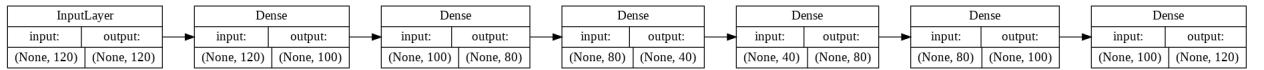


Figura C.9: Arquitectura C

C.2.2. Autoencoder convolucional apilado

Arquitectura A

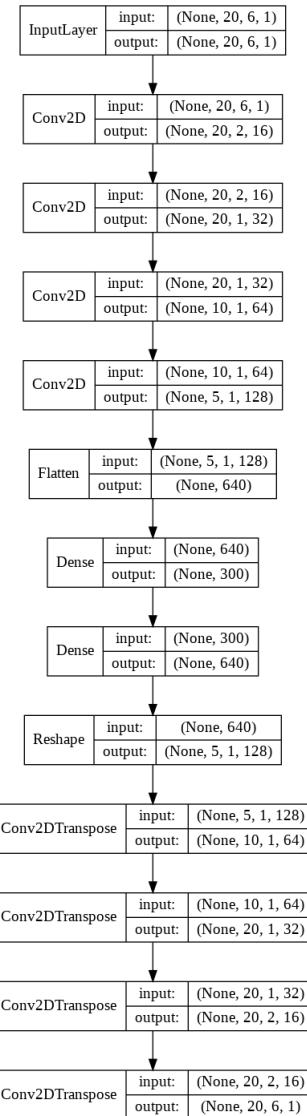


Figura C.10: Arquitectura A

Arquitectura B

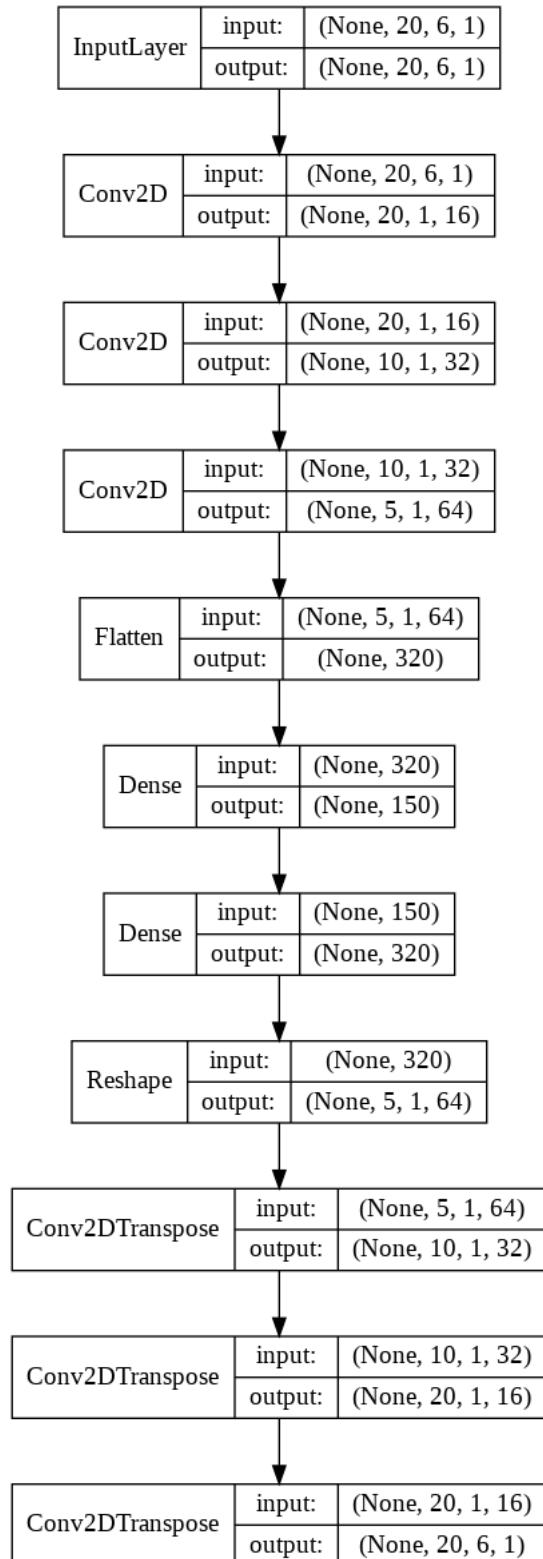


Figura C.11: Arquitectura B

Arquitectura C

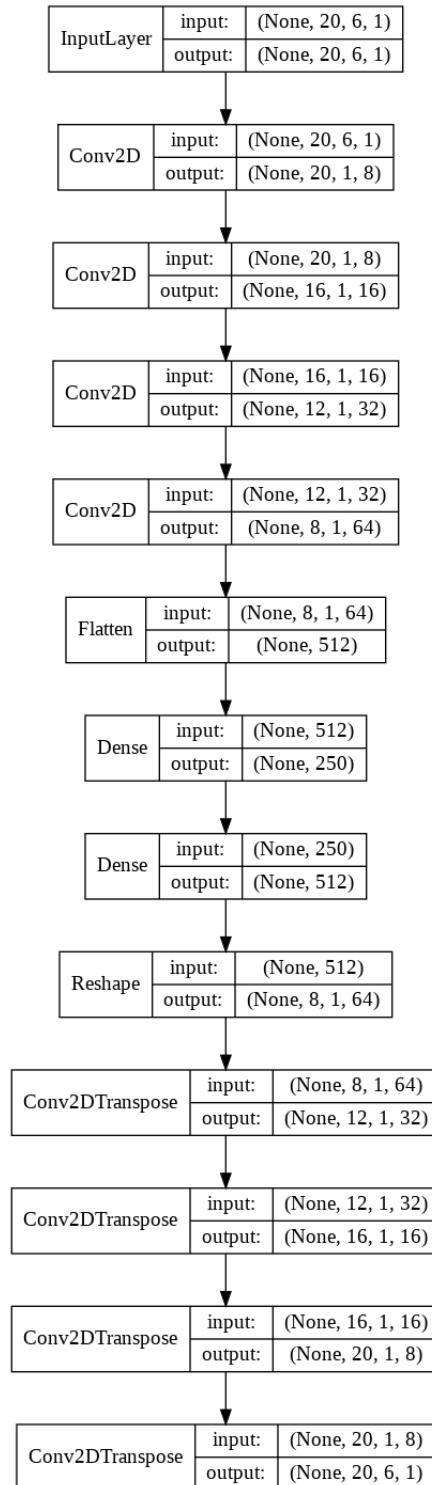


Figura C.12: Arquitectura C

C.3. Dataset 3

Arquitectura A

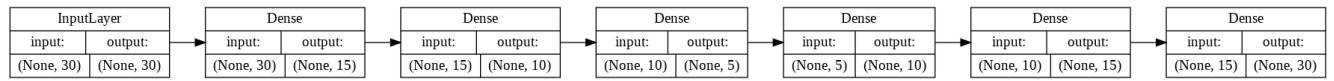


Figura C.13: Arquitectura A

Arquitectura B

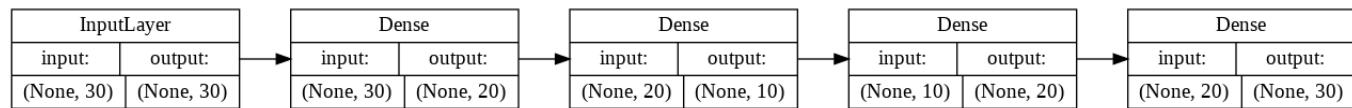


Figura C.14: Arquitectura B

Arquitectura C

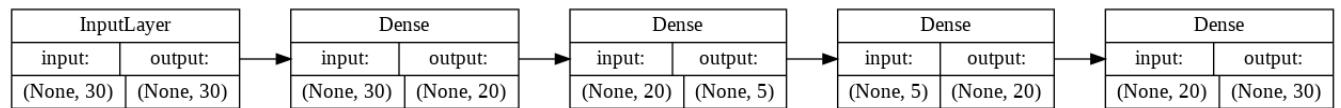


Figura C.15: Arquitectura C

C.3.1. Autoencoder convolucional apilado

Arquitectura A

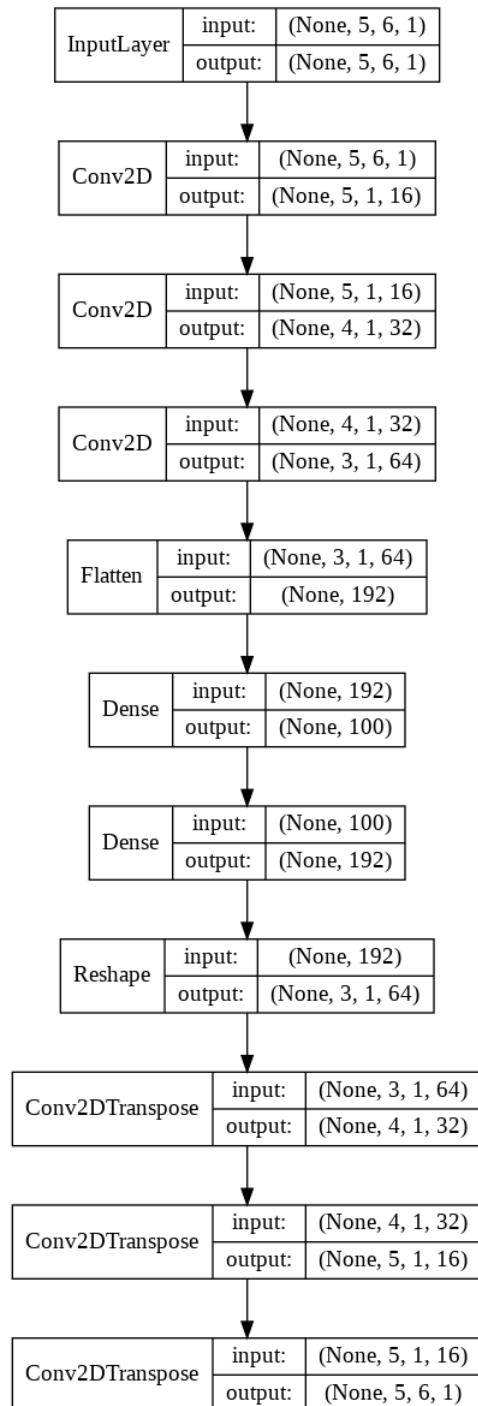


Figura C.16: Arquitectura A

Arquitectura B

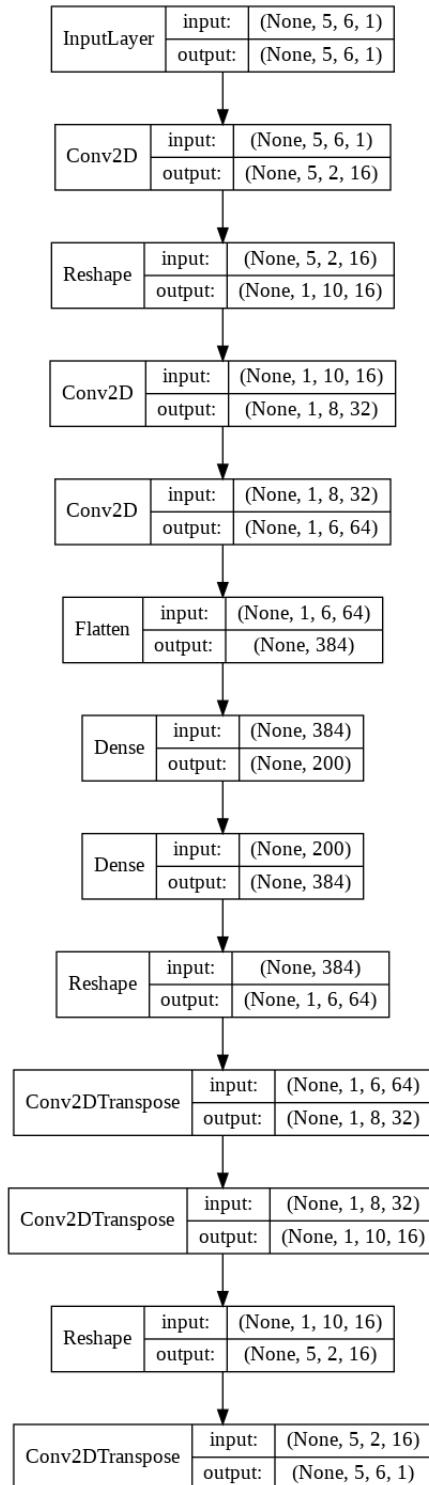


Figura C.17: Arquitectura B

Arquitectura C

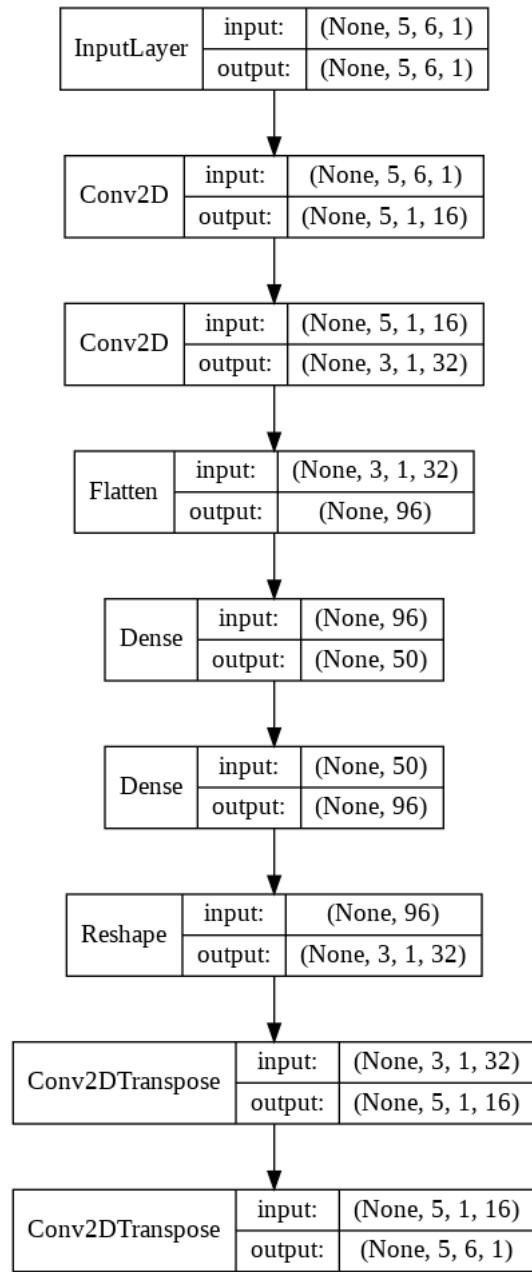


Figura C.18: Arquitectura C

C.4. Dataset 4

C.4.1. Autoencoder apilado

Arquitectura A

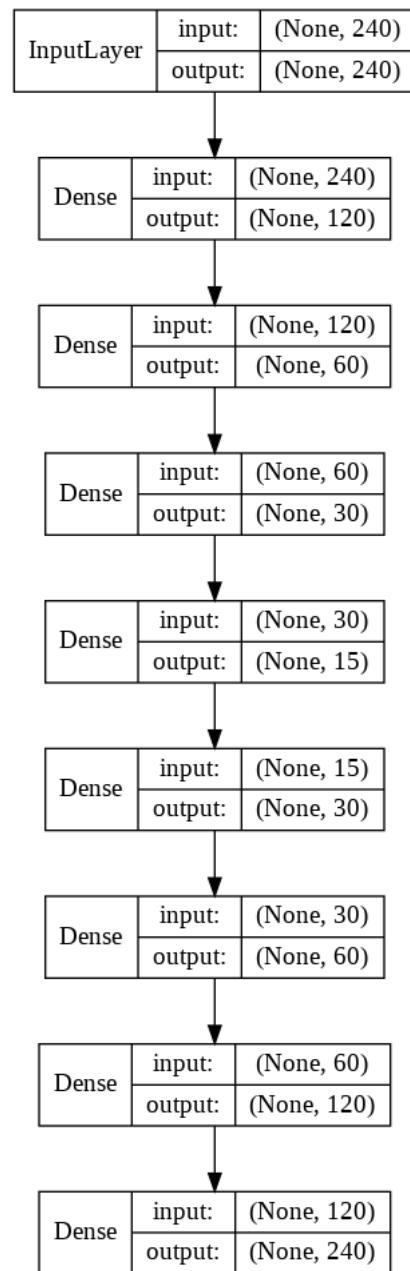


Figura C.19: Arquitectura A

Arquitectura B

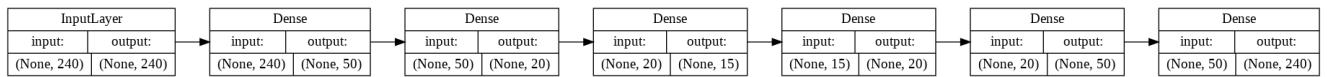


Figura C.20: Arquitectura B

Arquitectura C

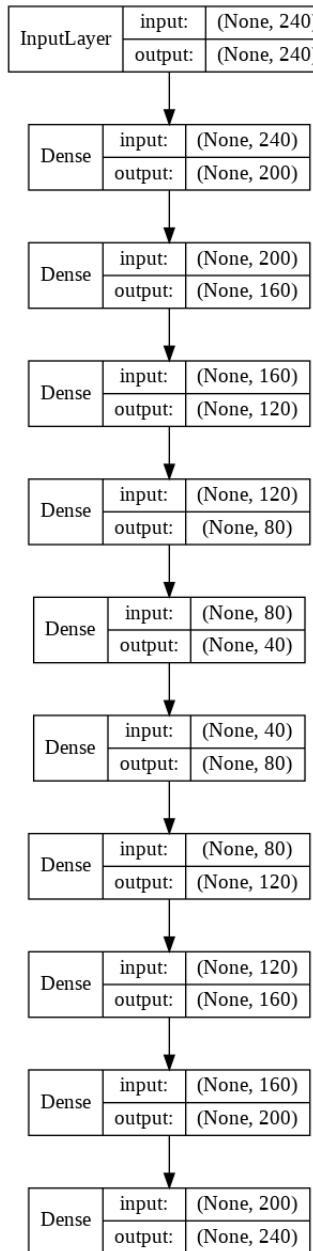


Figura C.21: Arquitectura C

C.4.2. Autoencoder convolucional apilado

Arquitectura A

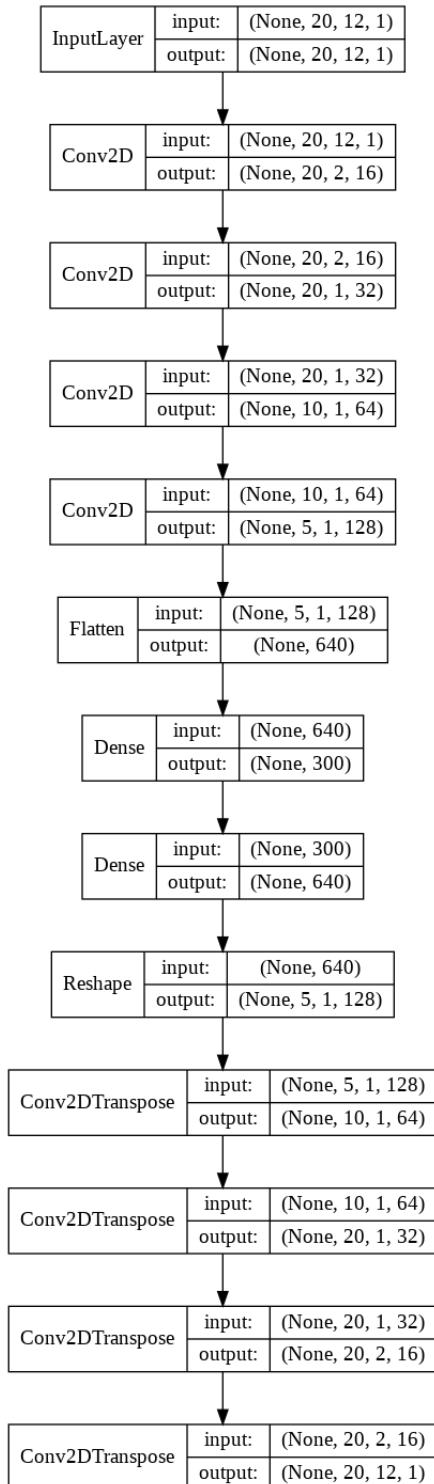


Figura C.22: Arquitectura A

Arquitectura B

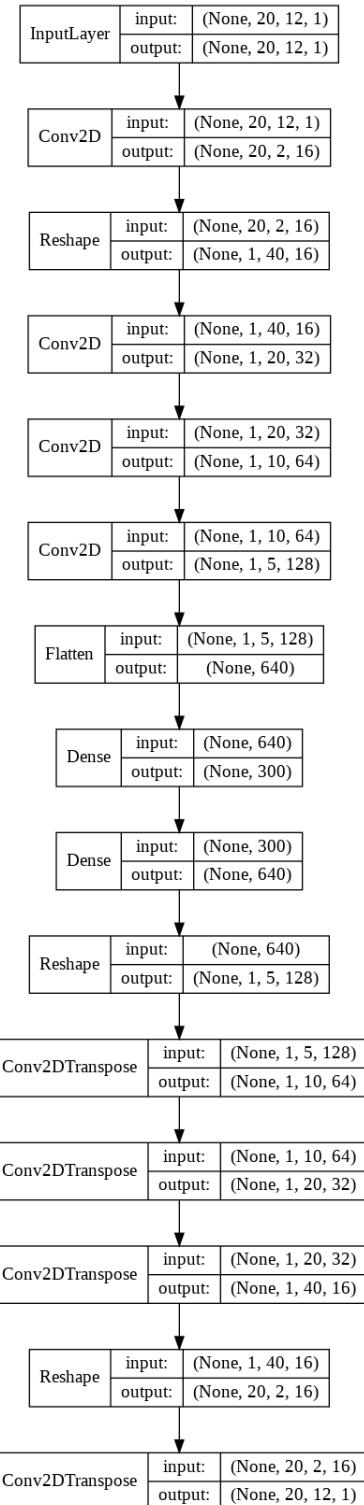


Figura C.23: Arquitectura B

Arquitectura C

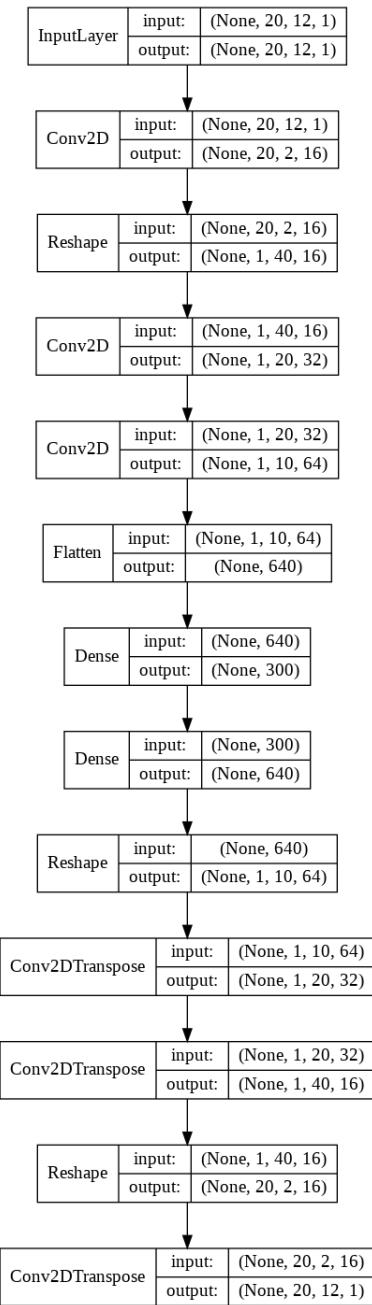


Figura C.24: Arquitectura C

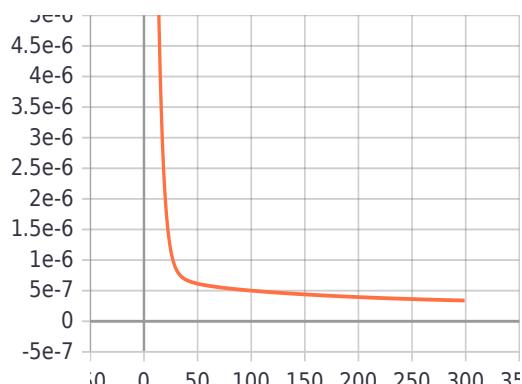
Apéndice D

Curvas de aprendizaje y validación

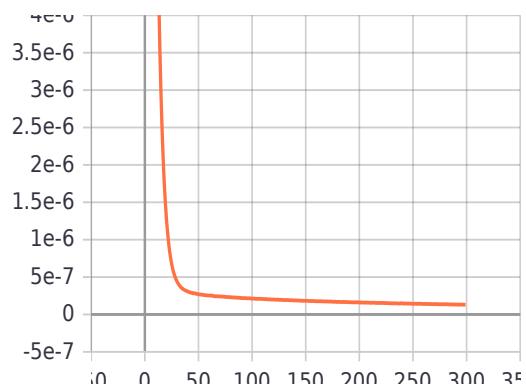
D.1. Dataset 1(Autoencoder apilado)

D.1.1. Arquitectura A

Curva de función de costo

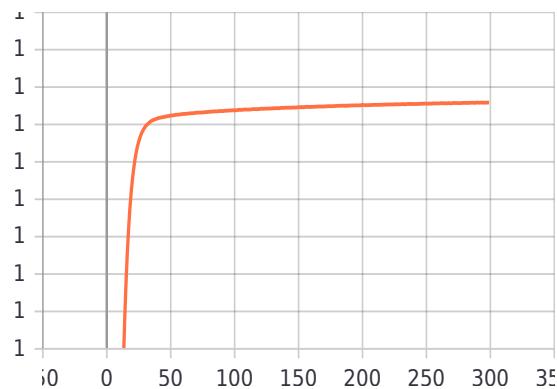


(a) Función de costo en entrenamiento

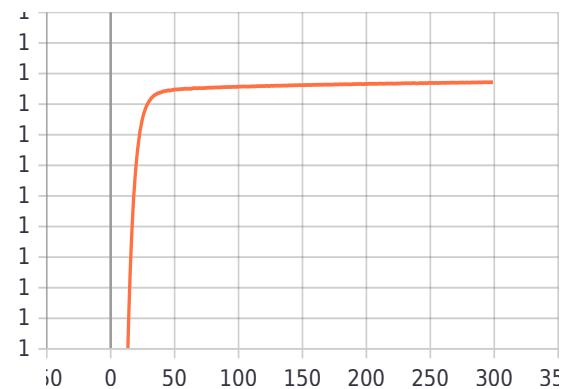


(b) Función de costo en validación

Curva de coeficiente de determinación

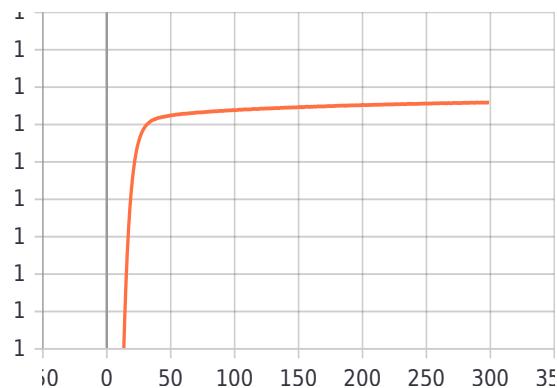


(a) Coeficiente de determinación en entrenamiento

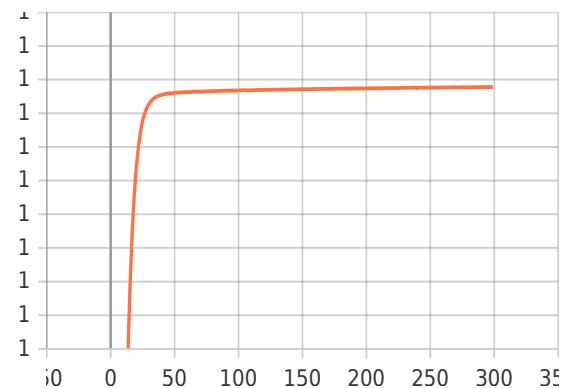


(b) Coeficiente de determinación en validación

Curva de varianza explicada



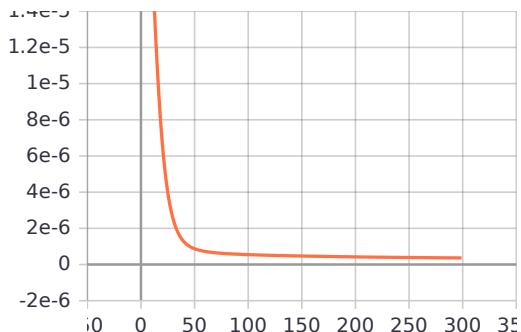
(a) Varianza explicada en entrenamiento



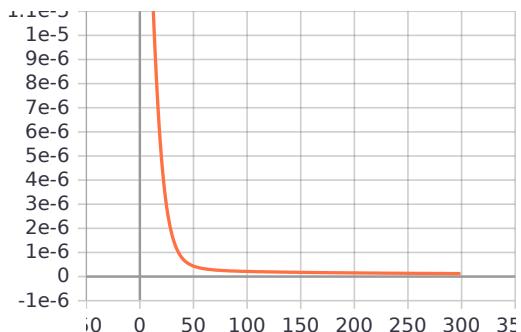
(b) Varianza explicada en validación

D.1.2. Arquitectura B

Curva de función de costo

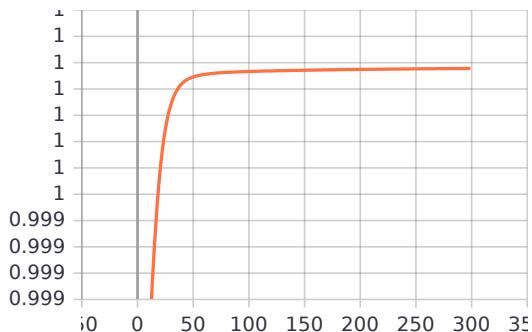


(a) Función de costo en entrenamiento

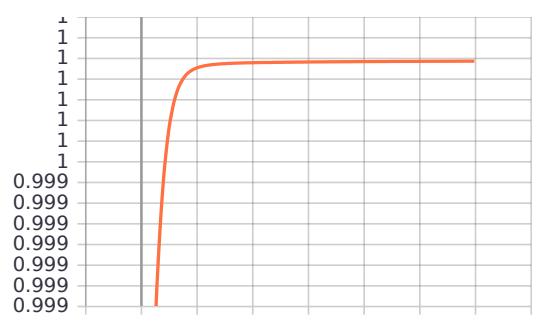


(b) Función de costo en validación

Curva de coeficiente de determinación

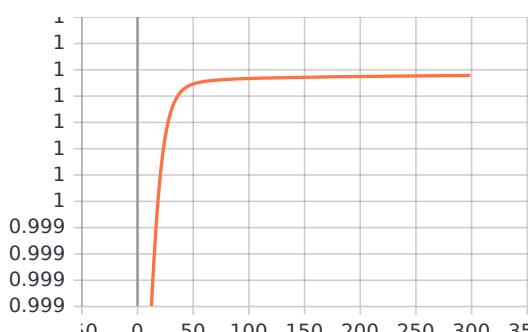


(a) Coeficiente de determinación en entrenamiento

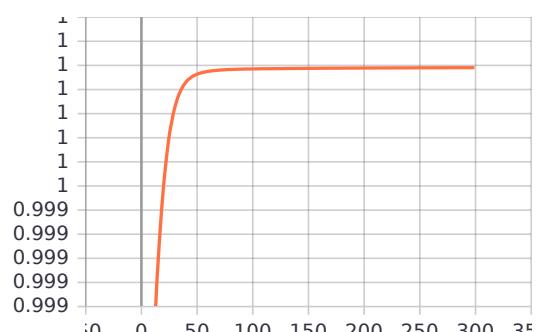


(b) Coeficiente de determinación en validación

Curva de varianza explicada



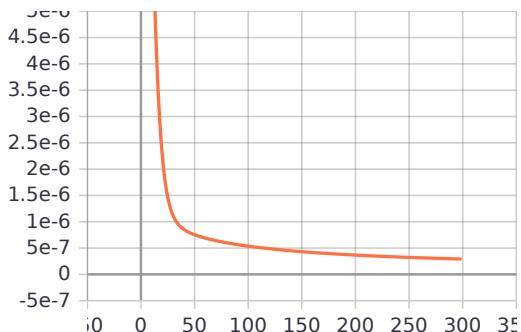
(a) Varianza explicada en entrenamiento



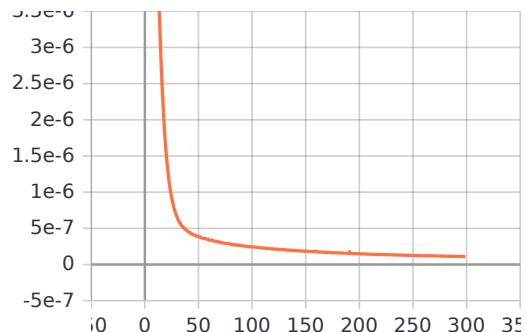
(b) Varianza explicada en validación

D.1.3. Arquitectura C

Curva de función de costo

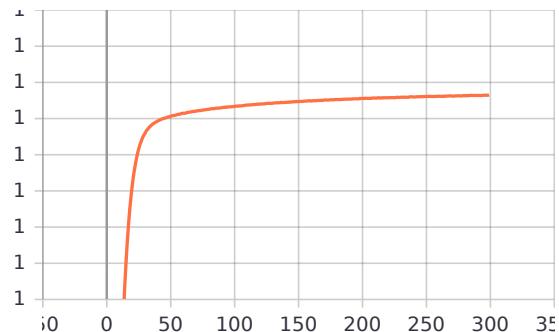


(a) Función de costo en entrenamiento



(b) Función de costo en validación

Curva de coeficiente de determinación

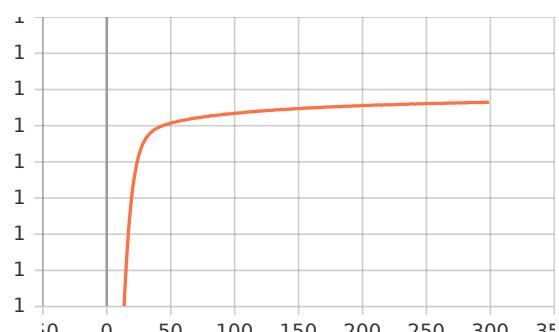


(a) Coeficiente de determinación en entrenamiento

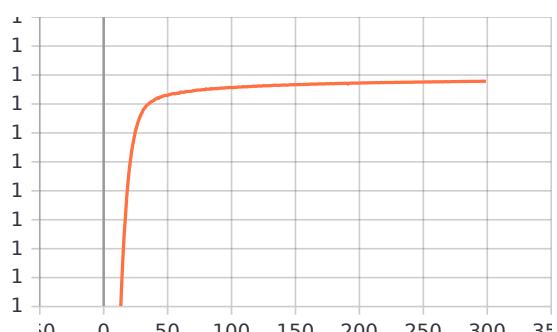


(b) Coeficiente de determinación en validación

Curva de varianza explicada



(a) Varianza explicada en entrenamiento

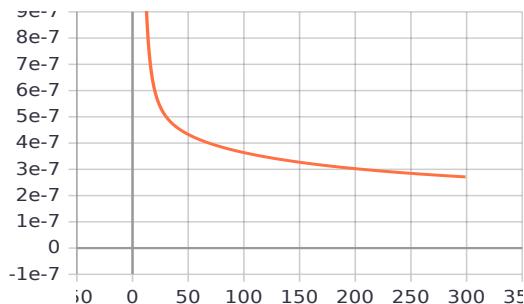


(b) Varianza explicada en validación

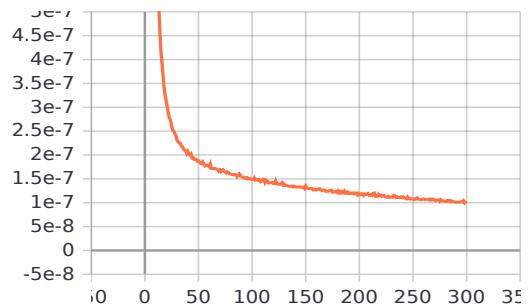
D.2. Dataset 1(Autoencoder convolucional apilado)

D.2.1. Arquitectura A

Curva de función de costo

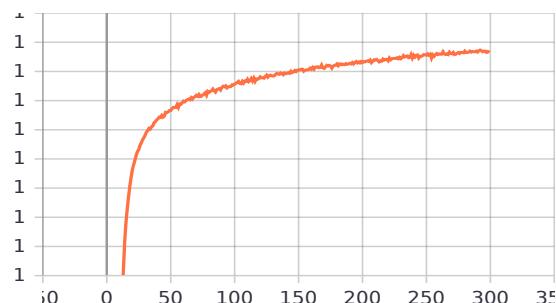


(a) Función de costo en entrenamiento

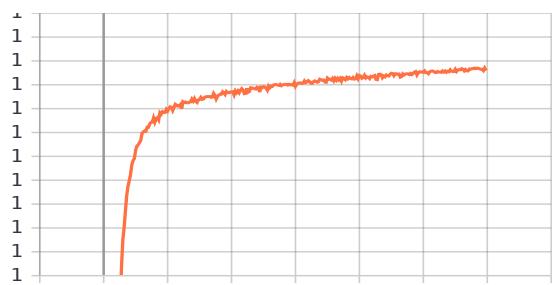


(b) Función de costo en validación

Curva de coeficiente de determinación

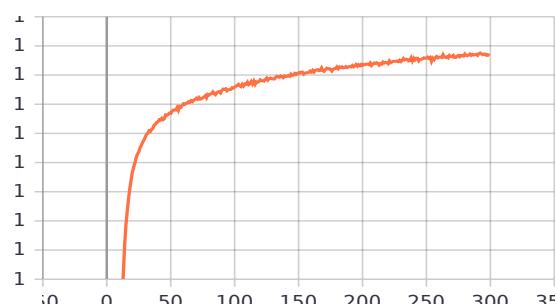


(a) Coeficiente de determinación en entrenamiento

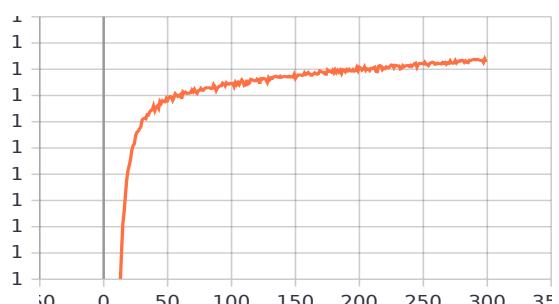


(b) Coeficiente de determinación en validación

Curva de varianza explicada



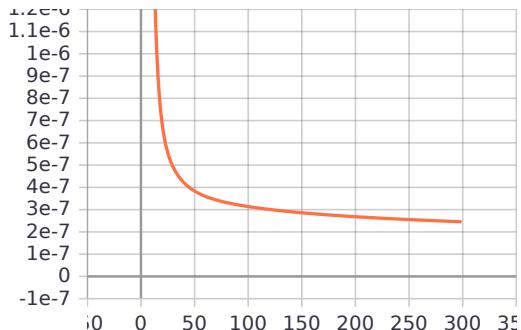
(a) Varianza explicada en entrenamiento



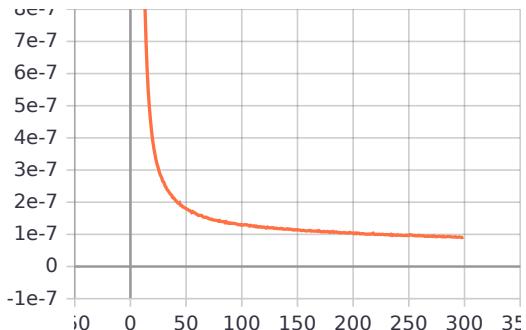
(b) Varianza explicada en validación

D.2.2. Arquitectura B

Curva de función de costo

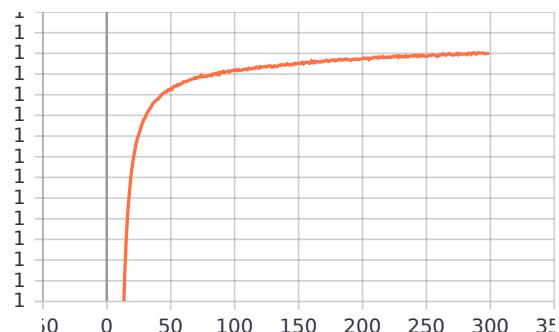


(a) Función de costo en entrenamiento

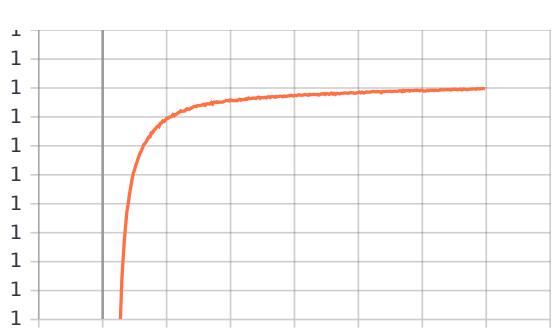


(b) Función de costo en validación

Curva de coeficiente de determinación

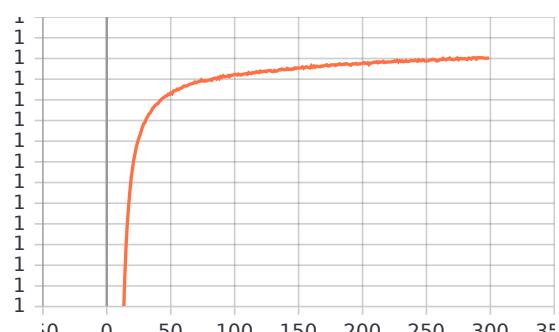


(a) Coeficiente de determinación en entrenamiento

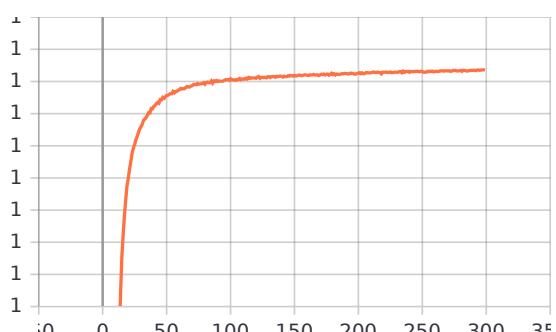


(b) Coeficiente de determinación en validación

Curva de varianza explicada



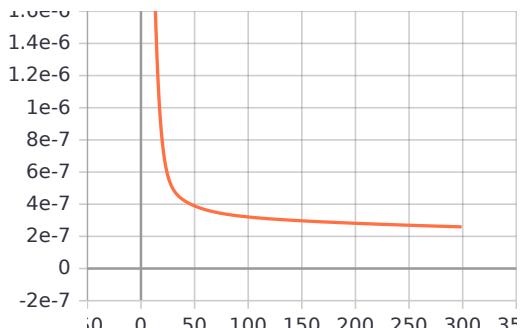
(a) Varianza explicada en entrenamiento



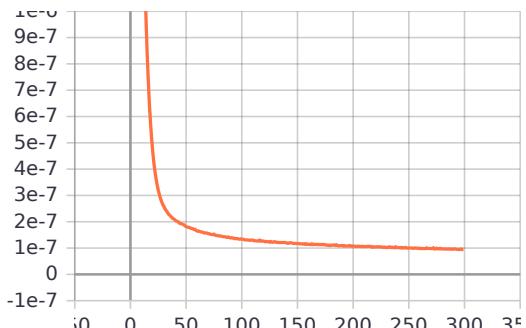
(b) Varianza explicada en validación

D.2.3. Arquitectura C

Curva de función de costo

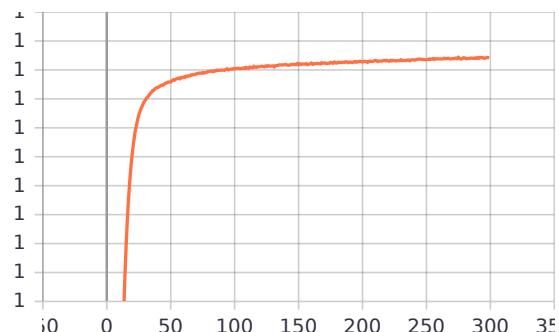


(a) Función de costo en entrenamiento

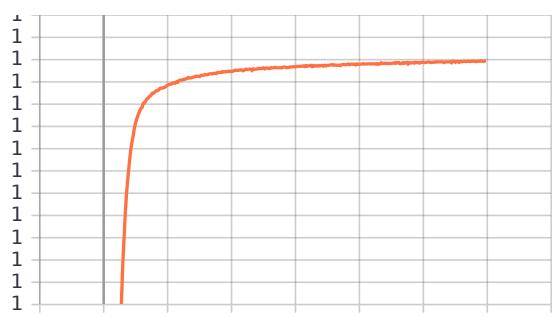


(b) Función de costo en validación

Curva de coeficiente de determinación

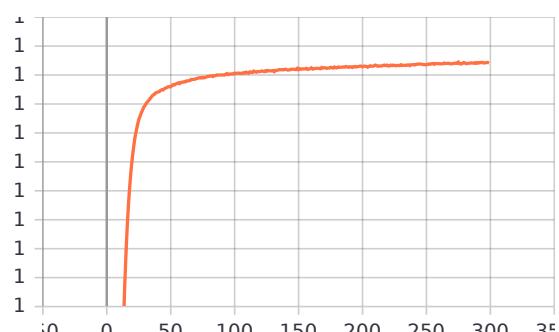


(a) Coeficiente de determinación en entrenamiento

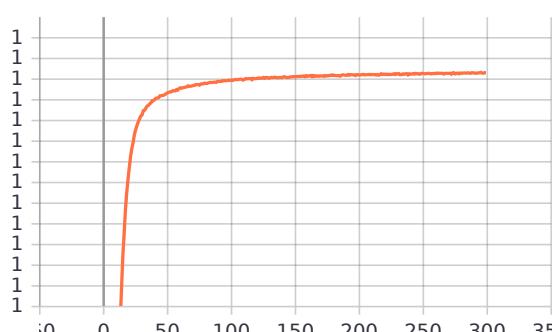


(b) Coeficiente de determinación en validación

Curva de varianza explicada



(a) Varianza explicada en entrenamiento

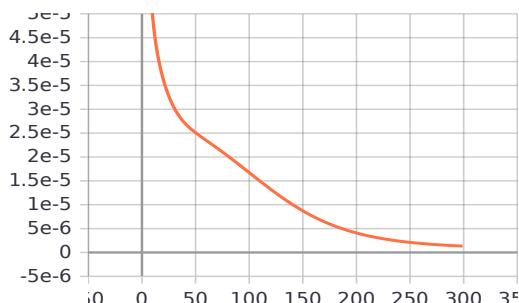


(b) Varianza explicada en validación

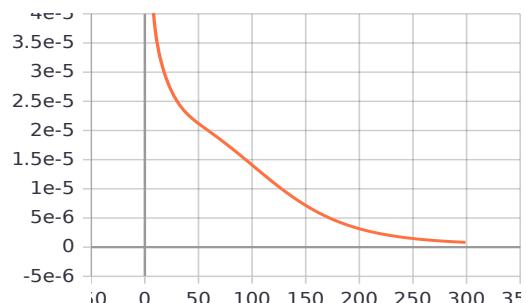
D.3. Dataset 2(Autoencoder apilado)

D.3.1. Arquitectura A

Curva de función de costo

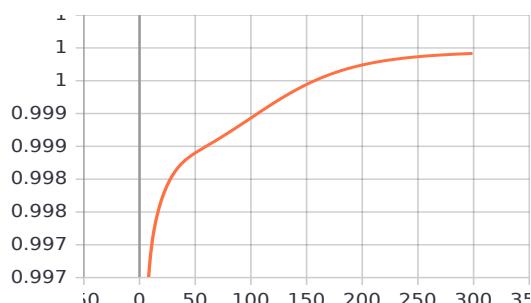


(a) Función de costo en entrenamiento

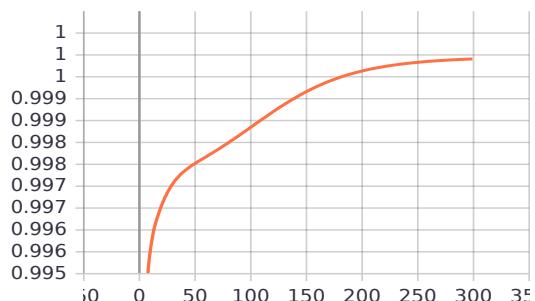


(b) Función de costo en validación

Curva de coeficiente de determinación

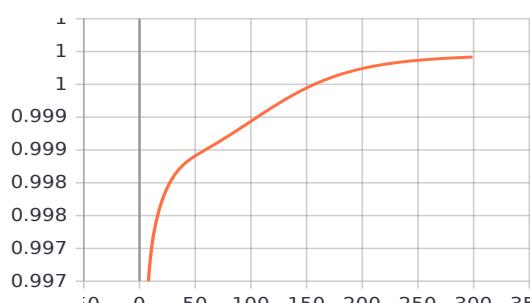


(a) Coeficiente de determinación en entrenamiento

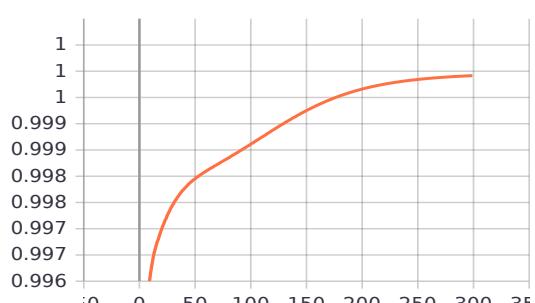


(b) Coeficiente de determinación en validación

Curva de varianza explicada



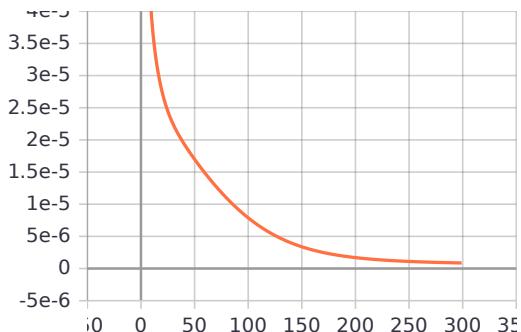
(a) Varianza explicada en entrenamiento



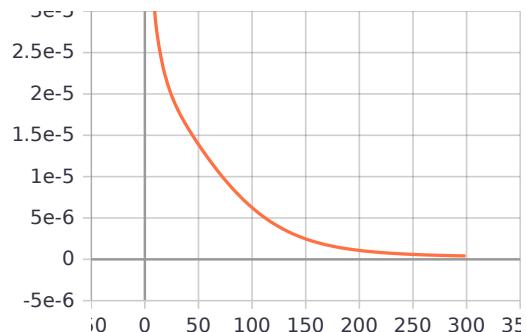
(b) Varianza explicada en validación

D.3.2. Arquitectura B

Curva de función de costo

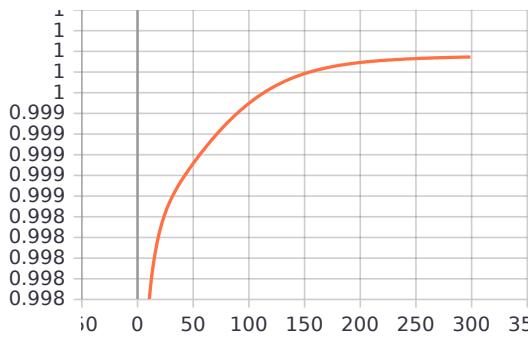


(a) Función de costo en entrenamiento

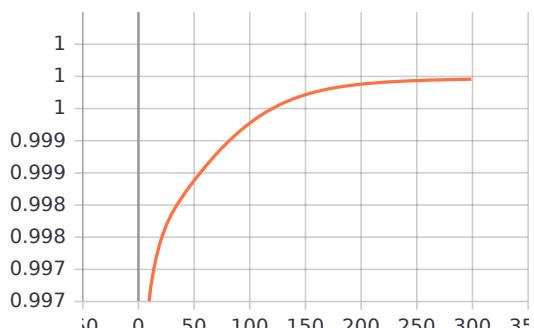


(b) Función de costo en validación

Curva de coeficiente de determinación

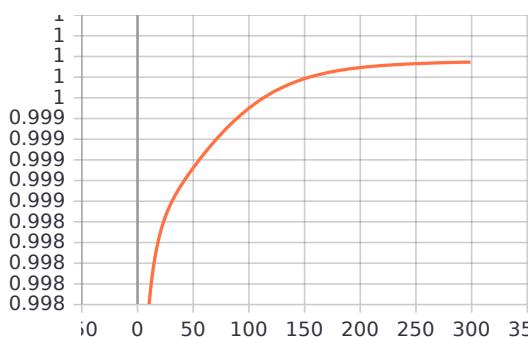


(a) Coeficiente de determinación en entrenamiento

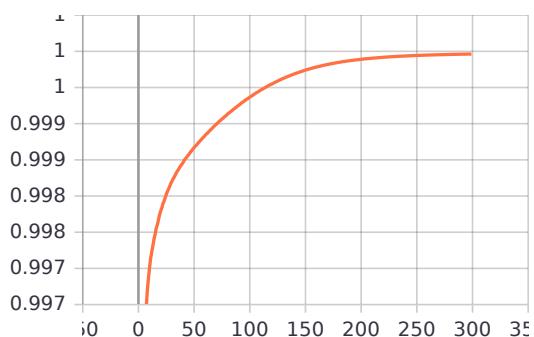


(b) Coeficiente de determinación en validación

Curva de varianza explicada



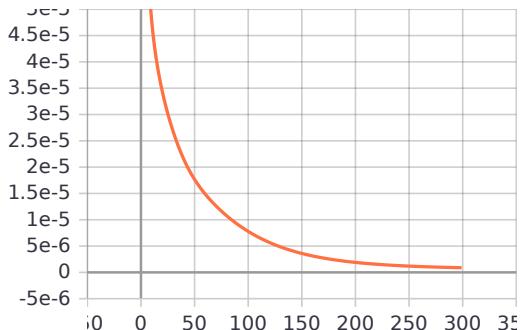
(a) Varianza explicada en entrenamiento



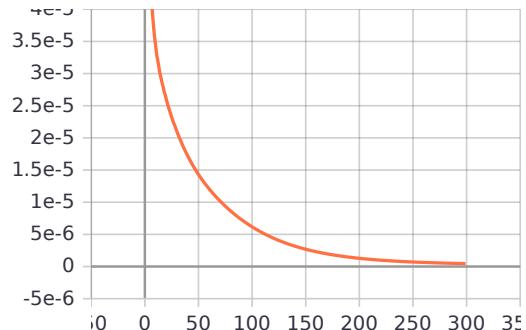
(b) Varianza explicada en validación

D.3.3. Arquitectura C

Curva de función de costo

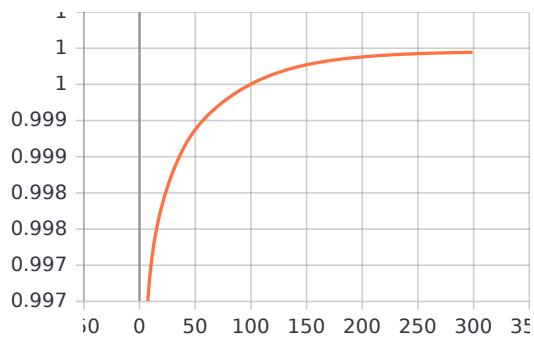


(a) Función de costo en entrenamiento

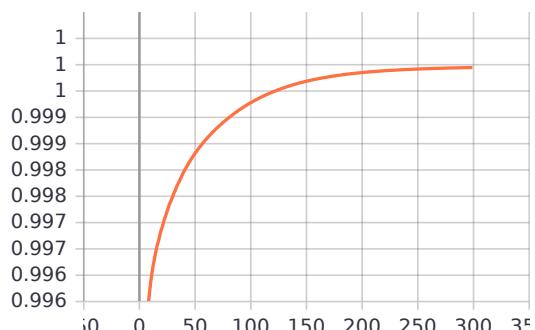


(b) Función de costo en validación

Curva de coeficiente de determinación

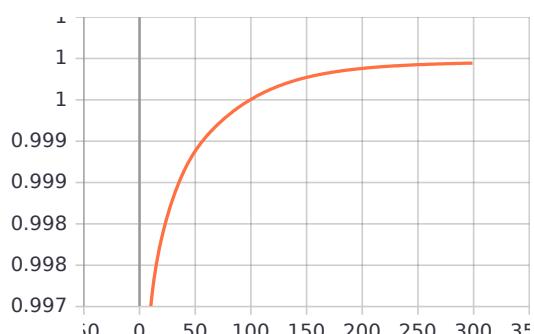


(a) Coeficiente de determinación en entrenamiento

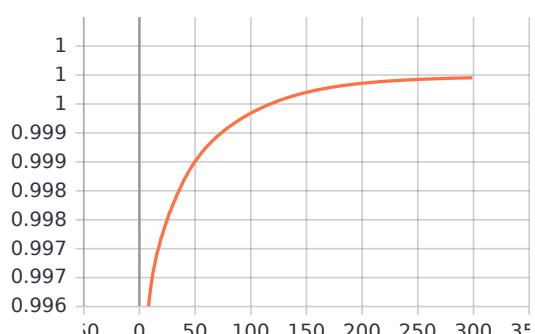


(b) Coeficiente de determinación en validación

Curva de varianza explicada



(a) Varianza explicada en entrenamiento

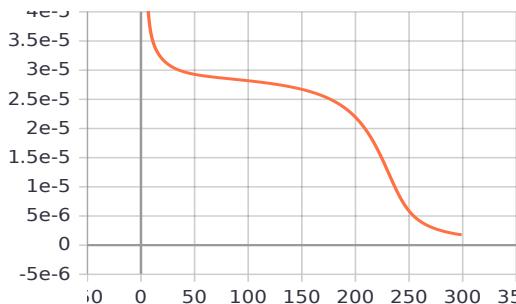


(b) Varianza explicada en validación

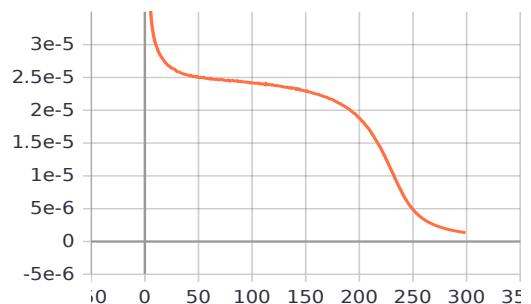
D.4. Dataset 2(Autoencoder convolucional apilado)

D.4.1. Arquitectura A

Curva de función de costo

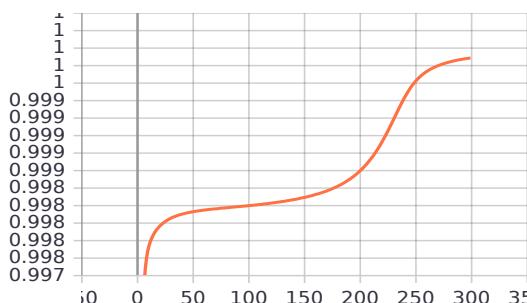


(a) Función de costo en entrenamiento

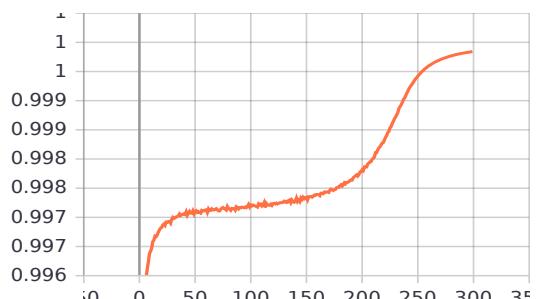


(b) Función de costo en validación

Curva de coeficiente de determinación

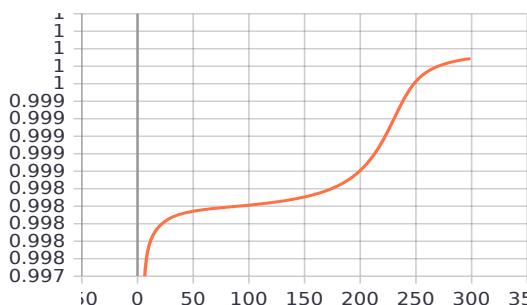


(a) Coeficiente de determinación en entrenamiento

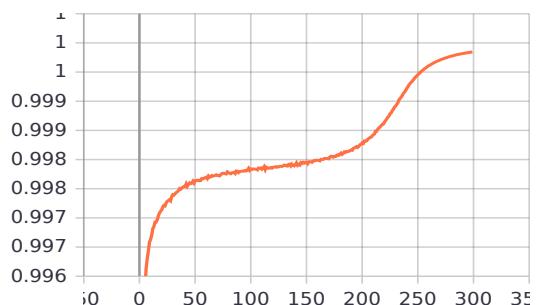


(b) Coeficiente de determinación en validación

Curva de varianza explicada



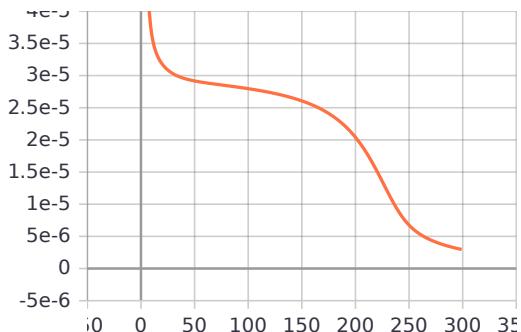
(a) Varianza explicada en entrenamiento



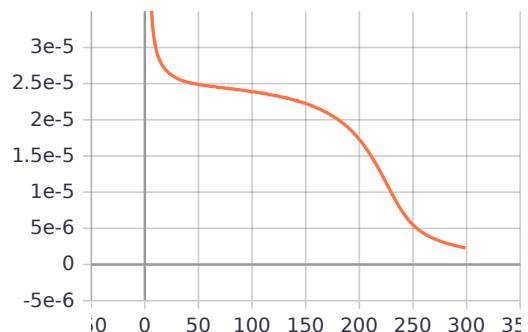
(b) Varianza explicada en validación

D.4.2. Arquitectura B

Curva de función de costo

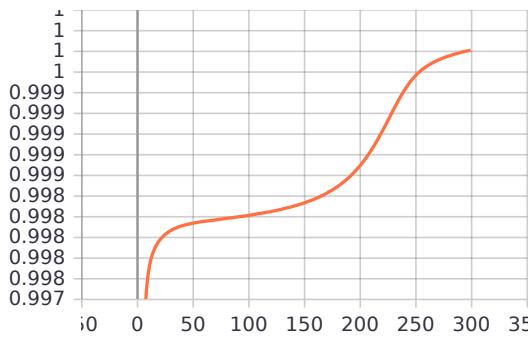


(a) Función de costo en entrenamiento

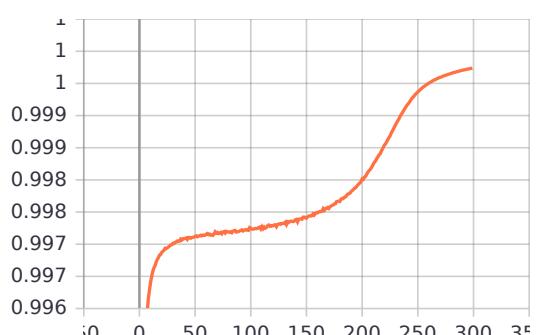


(b) Función de costo en validación

Curva de coeficiente de determinación

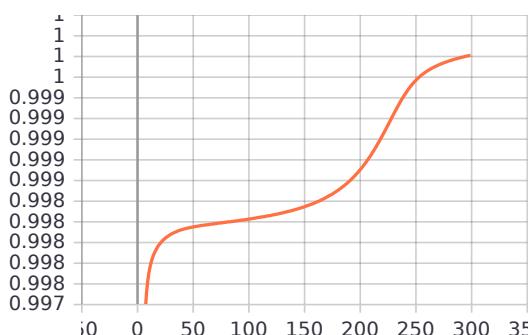


(a) Coeficiente de determinación en entrenamiento

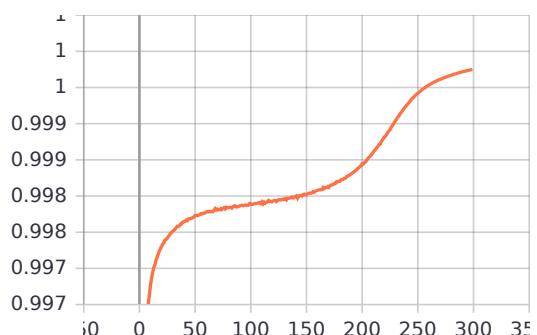


(b) Coeficiente de determinación en validación

Curva de varianza explicada



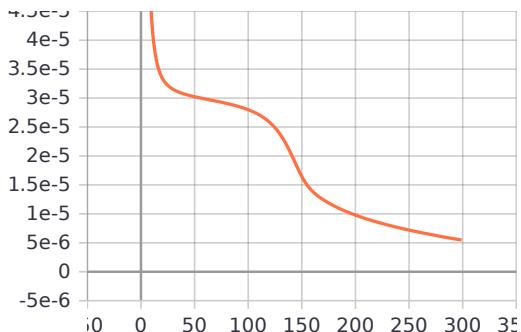
(a) Varianza explicada en entrenamiento



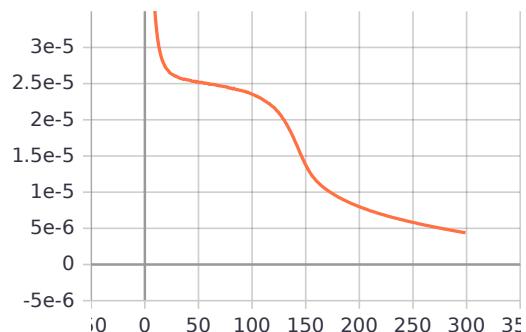
(b) Varianza explicada en validación

D.4.3. Arquitectura C

Curva de función de costo

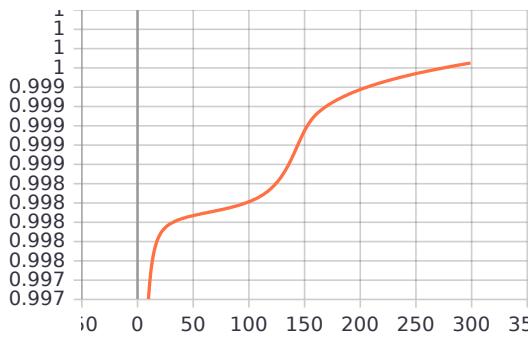


(a) Función de costo en entrenamiento

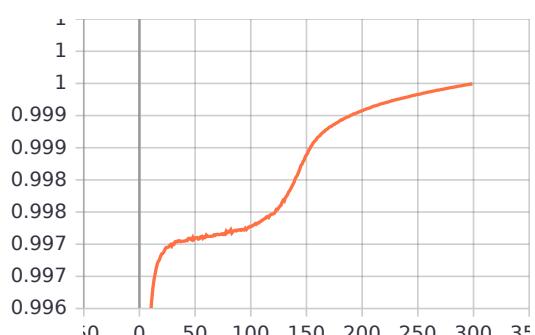


(b) Función de costo en validación

Curva de coeficiente de determinación

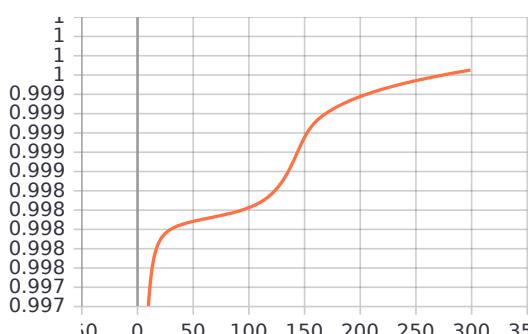


(a) Coeficiente de determinación en entrenamiento

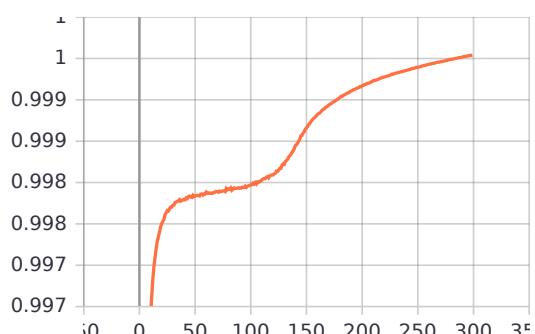


(b) Coeficiente de determinación en validación

Curva de varianza explicada



(a) Varianza explicada en entrenamiento

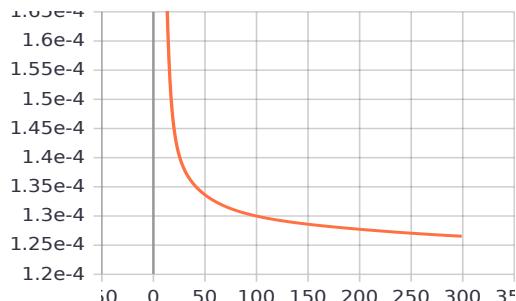


(b) Varianza explicada en validación

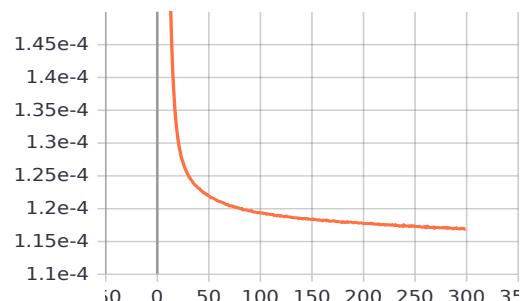
D.5. Dataset 3(Autoencoder apilado)

D.5.1. Arquitectura A

Curva de función de costo

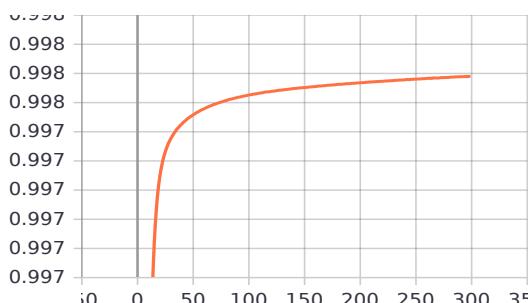


(a) Función de costo en entrenamiento

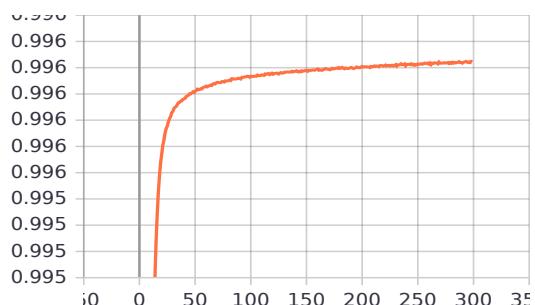


(b) Función de costo en validación

Curva de coeficiente de determinación

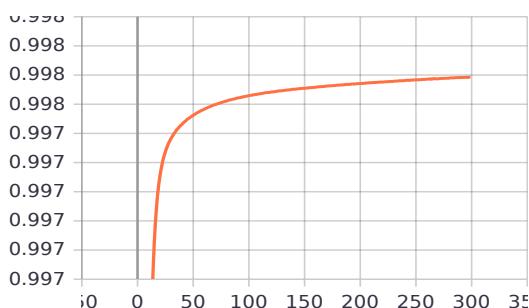


(a) Coeficiente de determinación en entrenamiento

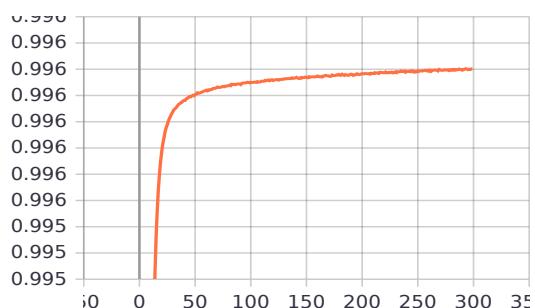


(b) Coeficiente de determinación en validación

Curva de varianza explicada



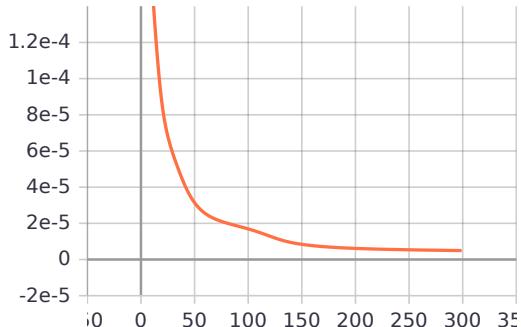
(a) Varianza explicada en entrenamiento



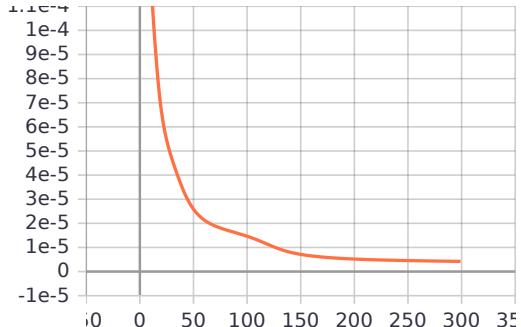
(b) Varianza explicada en validación

D.5.2. Arquitectura B

Curva de función de costo

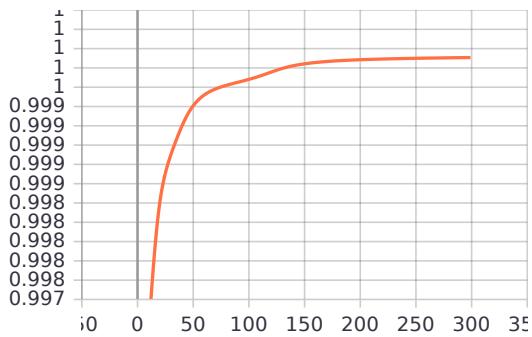


(a) Función de costo en entrenamiento

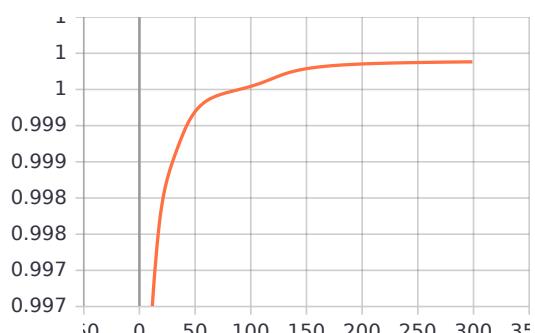


(b) Función de costo en validación

Curva de coeficiente de determinación

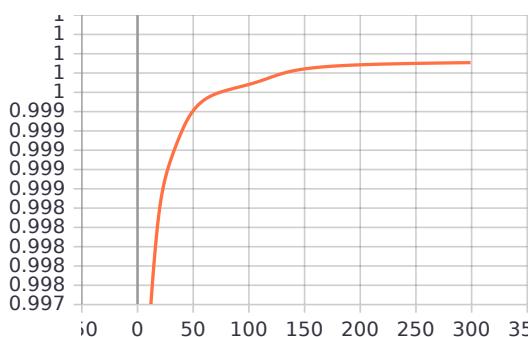


(a) Coeficiente de determinación en entrenamiento

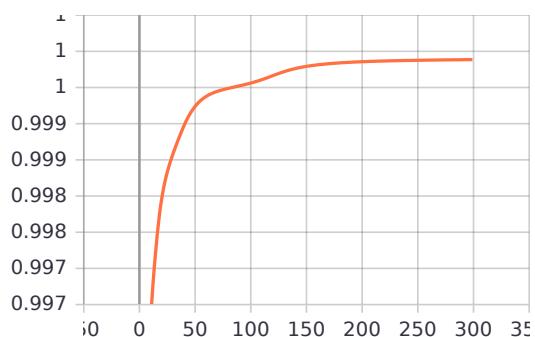


(b) Coeficiente de determinación en validación

Curva de varianza explicada



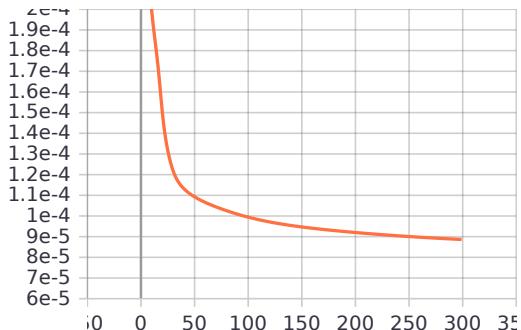
(a) Varianza explicada en entrenamiento



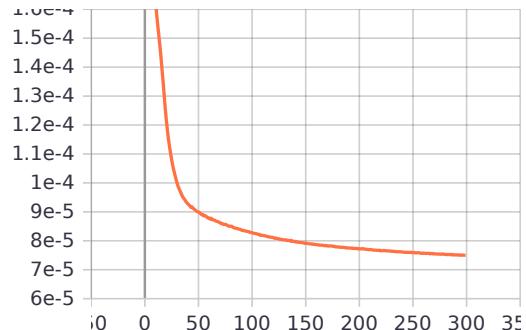
(b) Varianza explicada en validación

D.5.3. Arquitectura C

Curva de función de costo

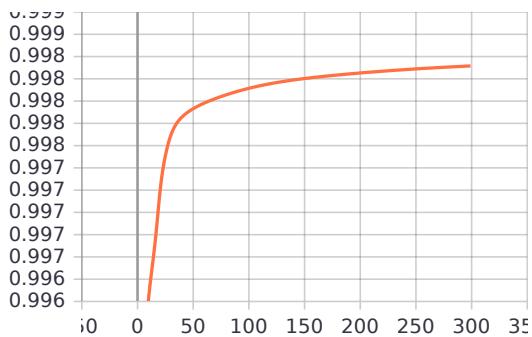


(a) Función de costo en entrenamiento

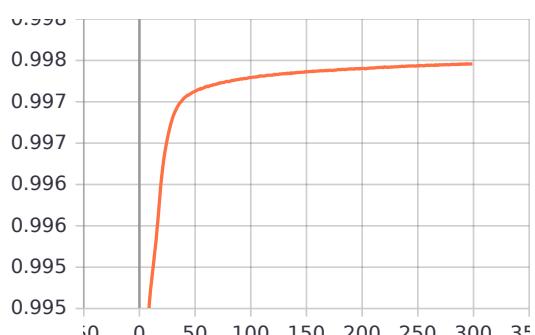


(b) Función de costo en validación

Curva de coeficiente de determinación

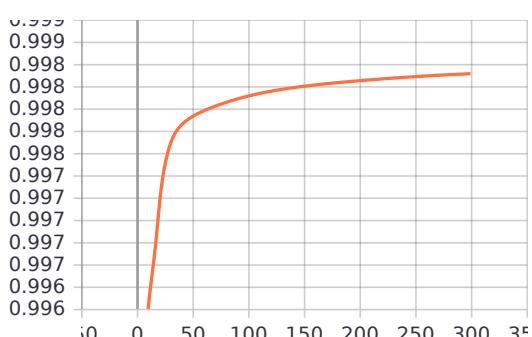


(a) Coeficiente de determinación en entrenamiento

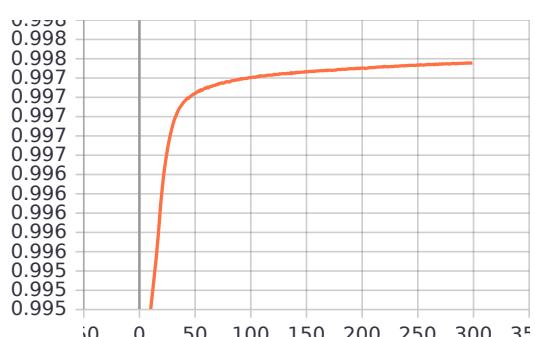


(b) Coeficiente de determinación en validación

Curva de varianza explicada



(a) Varianza explicada en entrenamiento

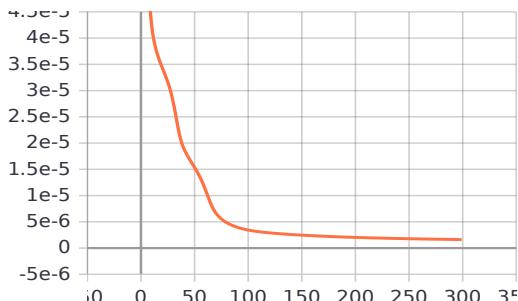


(b) Varianza explicada en validación

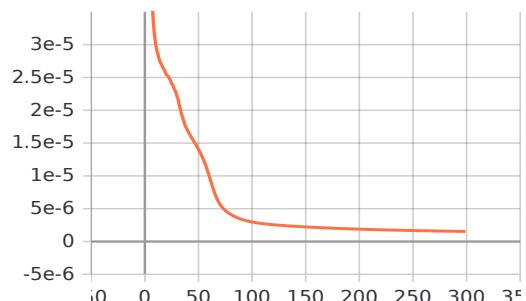
D.6. Dataset 3(Autoencoder convolucional apilado)

D.6.1. Arquitectura A

Curva de función de costo

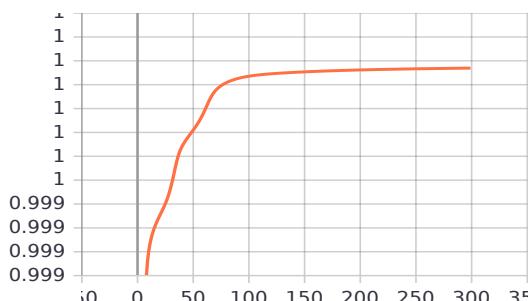


(a) Función de costo en entrenamiento

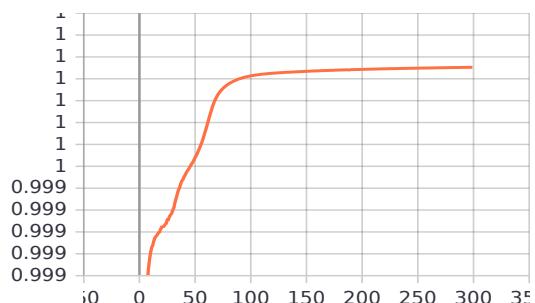


(b) Función de costo en validación

Curva de coeficiente de determinación

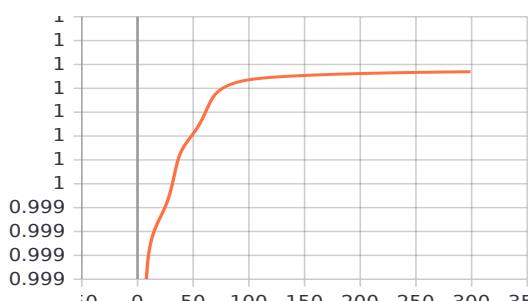


(a) Coeficiente de determinación en entrenamiento

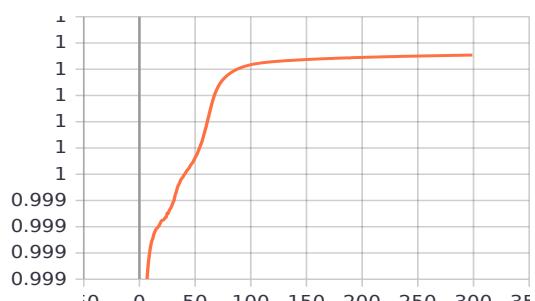


(b) Coeficiente de determinación en validación

Curva de varianza explicada



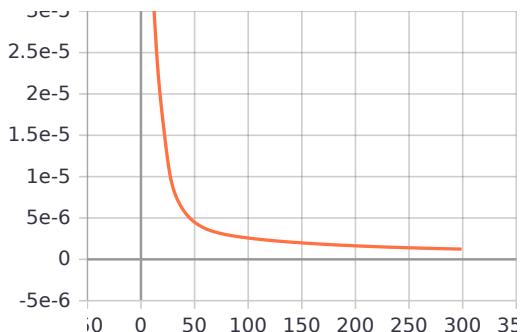
(a) Varianza explicada en entrenamiento



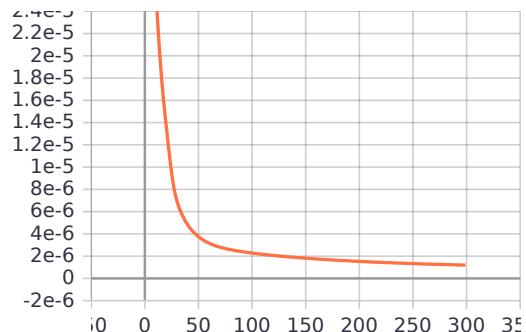
(b) Varianza explicada en validación

D.6.2. Arquitectura B

Curva de función de costo

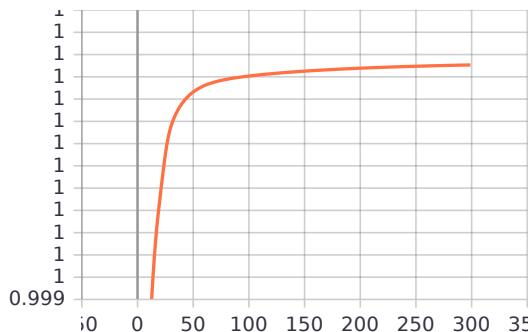


(a) Función de costo en entrenamiento

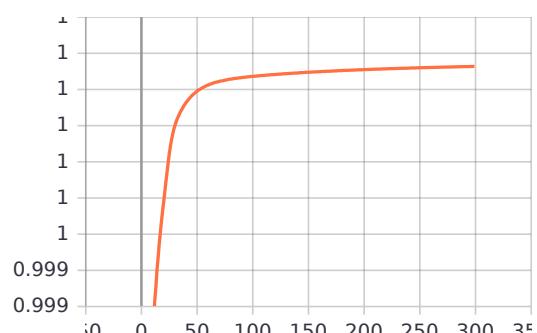


(b) Función de costo en validación

Curva de coeficiente de determinación

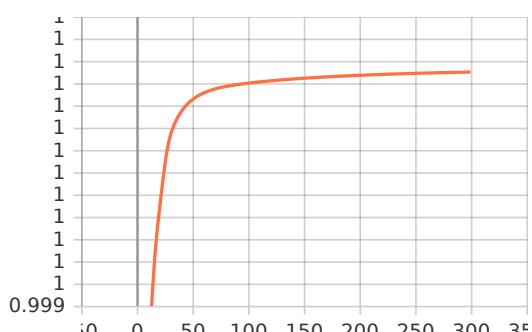


(a) Coeficiente de determinación en entrenamiento

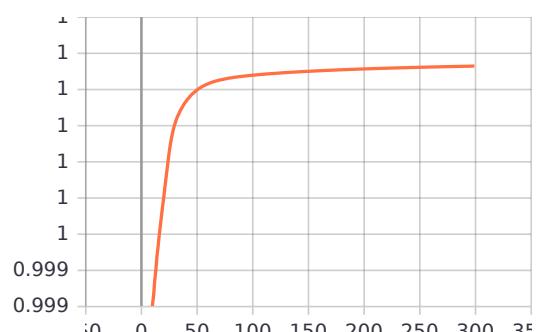


(b) Coeficiente de determinación en validación

Curva de varianza explicada



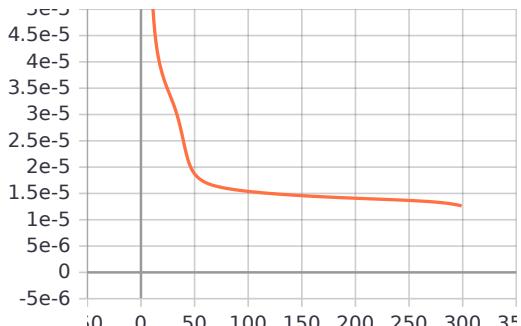
(a) Varianza explicada en entrenamiento



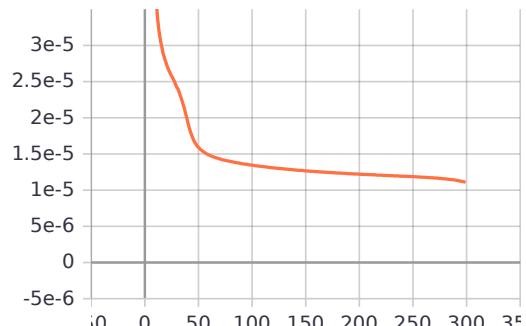
(b) Varianza explicada en validación

D.6.3. Arquitectura C

Curva de función de costo

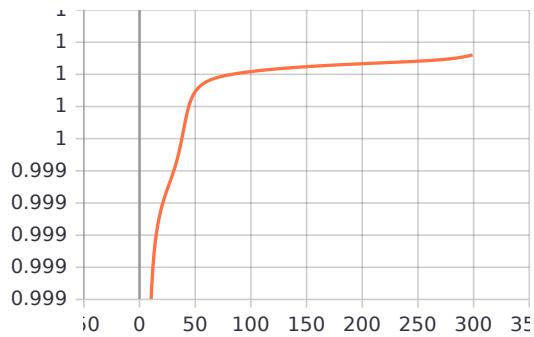


(a) Función de costo en entrenamiento

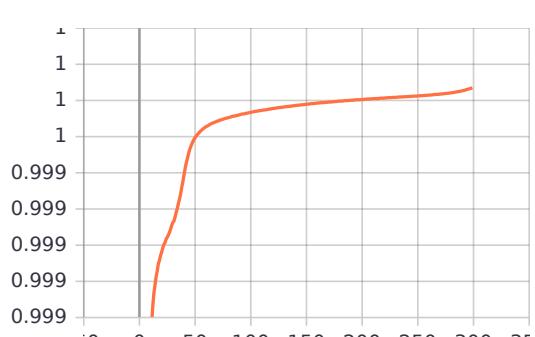


(b) Función de costo en validación

Curva de coeficiente de determinación

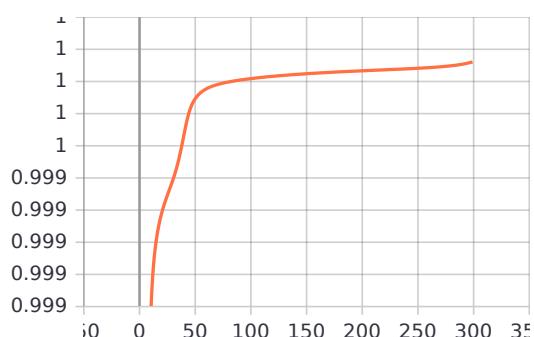


(a) Coeficiente de determinación en entrenamiento

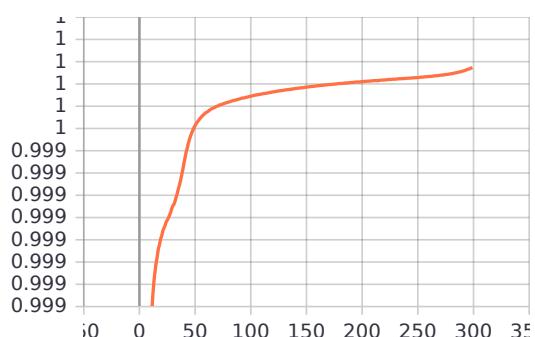


(b) Coeficiente de determinación en validación

Curva de varianza explicada



(a) Varianza explicada en entrenamiento

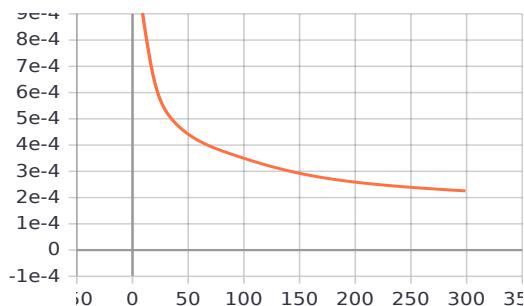


(b) Varianza explicada en validación

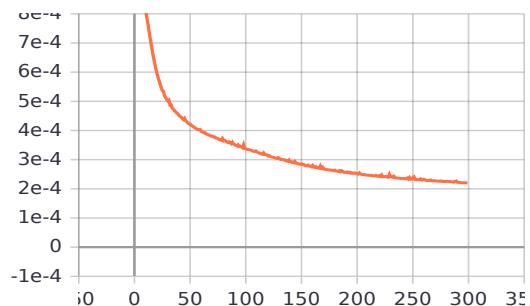
D.7. Dataset 4(Autoencoder apilado)

D.7.1. Arquitectura A

Curva de función de costo

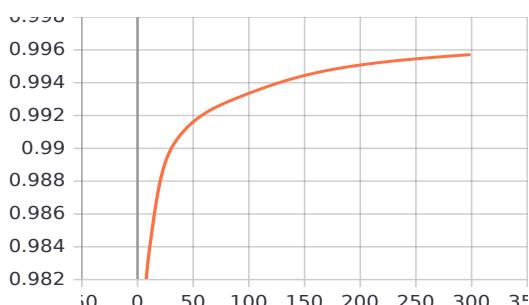


(a) Función de costo en entrenamiento

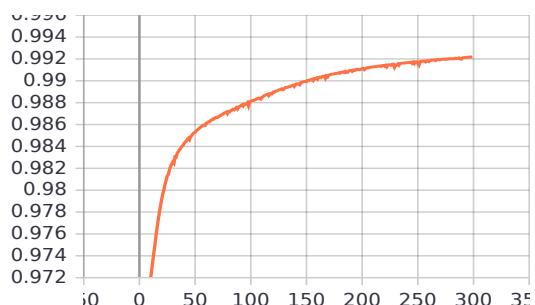


(b) Función de costo en validación

Curva de coeficiente de determinación

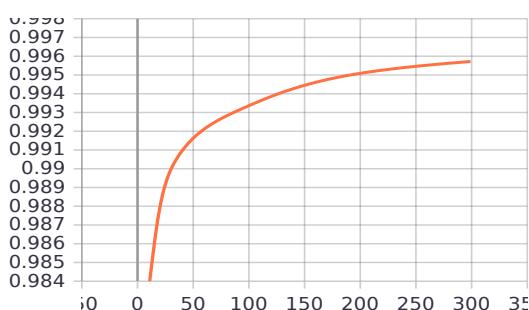


(a) Coeficiente de determinación en entrenamiento

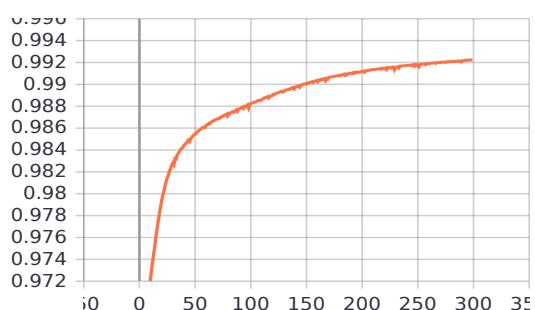


(b) Coeficiente de determinación en validación

Curva de varianza explicada



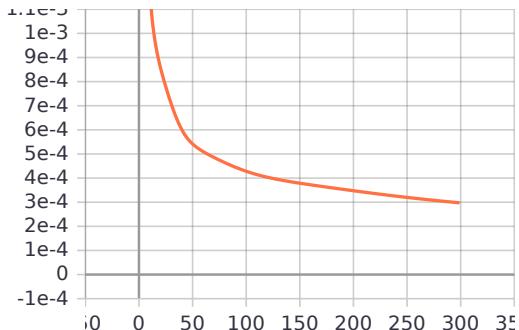
(a) Varianza explicada en entrenamiento



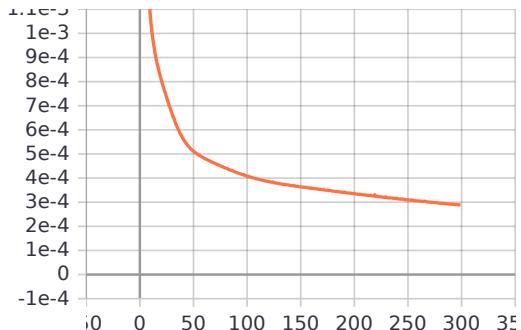
(b) Varianza explicada en validación

D.7.2. Arquitectura B

Curva de función de costo

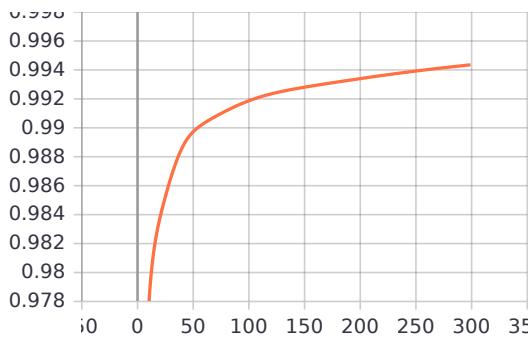


(a) Función de costo en entrenamiento

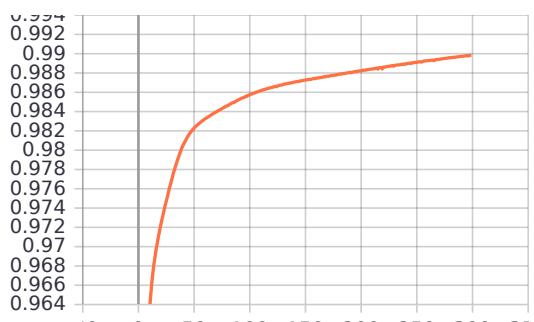


(b) Función de costo en validación

Curva de coeficiente de determinación

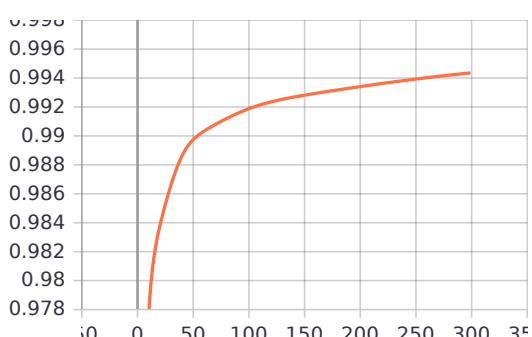


(a) Coeficiente de determinación en entrenamiento

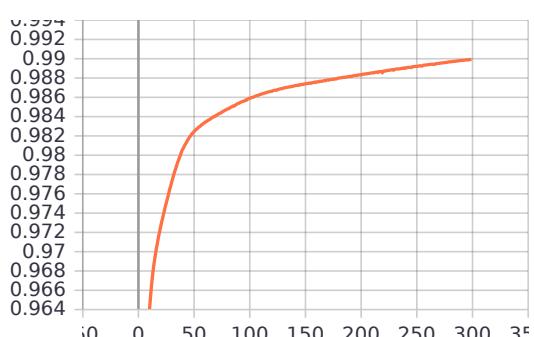


(b) Coeficiente de determinación en validación

Curva de varianza explicada



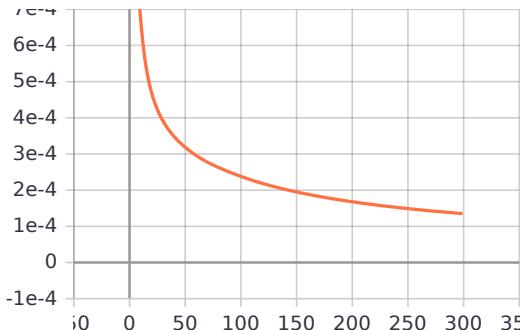
(a) Varianza explicada en entrenamiento



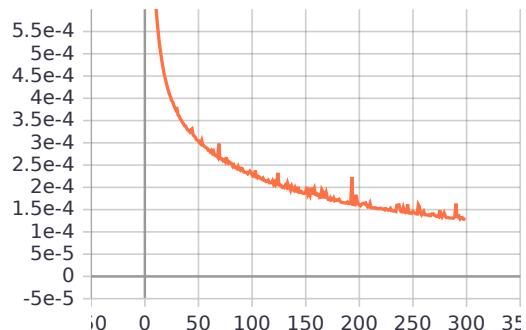
(b) Varianza explicada en validación

D.7.3. Arquitectura C

Curva de función de costo

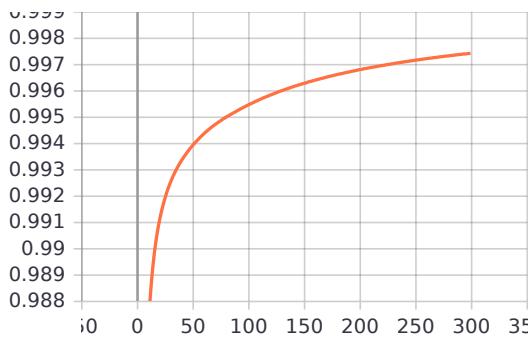


(a) Función de costo en entrenamiento

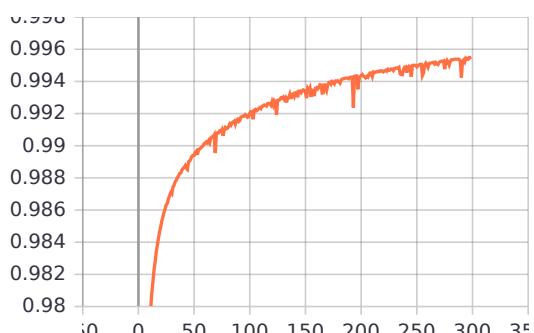


(b) Función de costo en validación

Curva de coeficiente de determinación

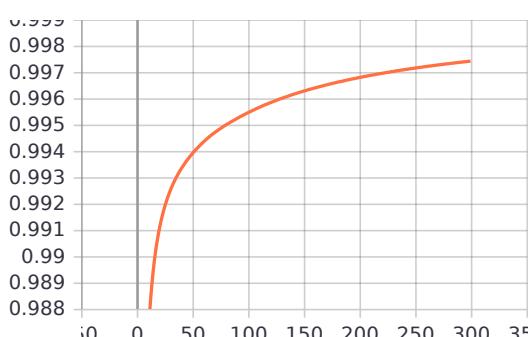


(a) Coeficiente de determinación en entrenamiento

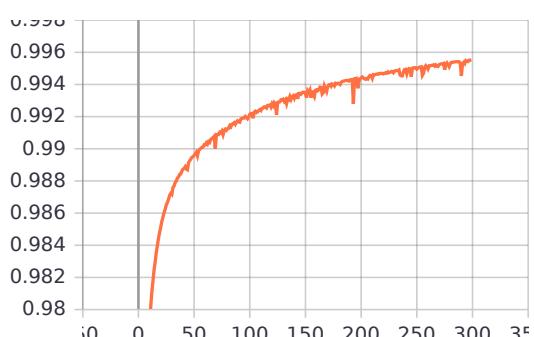


(b) Coeficiente de determinación en validación

Curva de varianza explicada



(a) Varianza explicada en entrenamiento

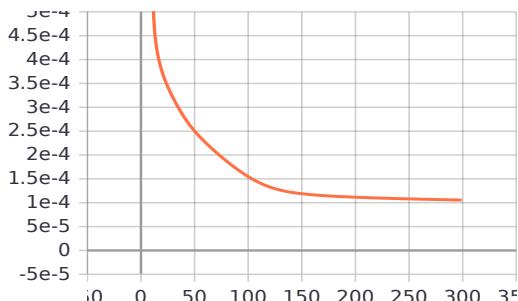


(b) Varianza explicada en validación

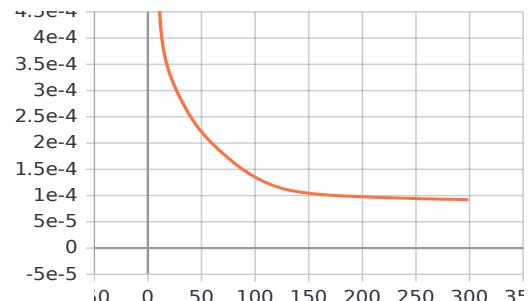
D.8. Dataset 4(Autoencoder convolucional apilado)

D.8.1. Arquitectura A

Curva de función de costo

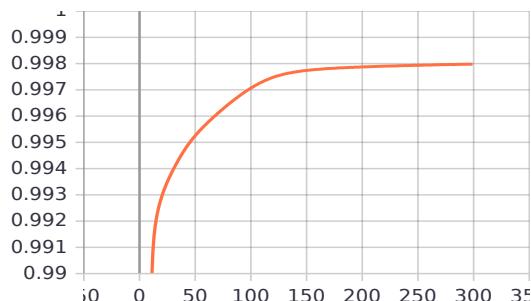


(a) Función de costo en entrenamiento

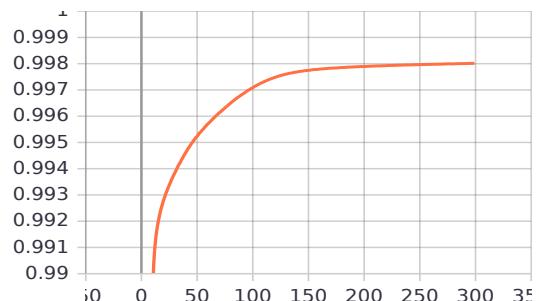


(b) Función de costo en validación

Curva de coeficiente de determinación

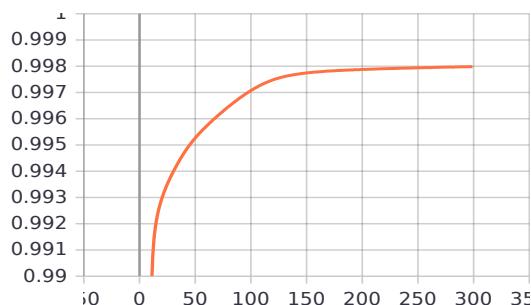


(a) Coeficiente de determinación en entrenamiento

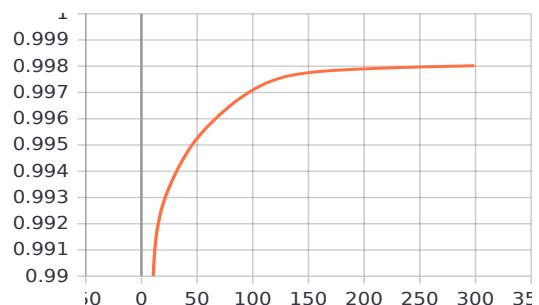


(b) Coeficiente de determinación en validación

Curva de varianza explicada



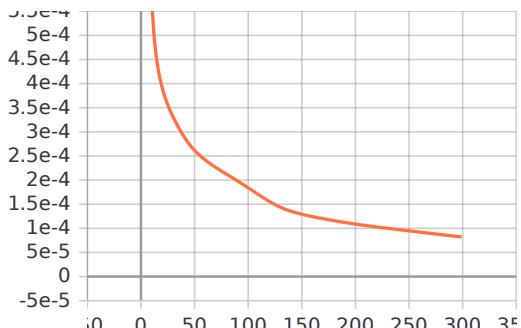
(a) Varianza explicada en entrenamiento



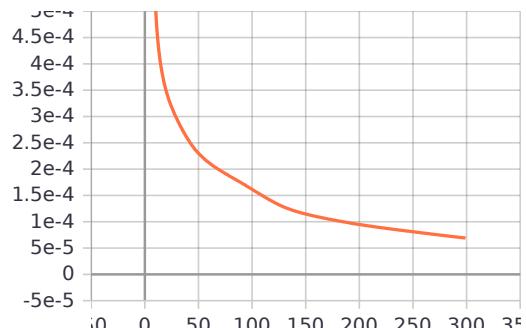
(b) Varianza explicada en validación

D.8.2. Arquitectura B

Curva de función de costo

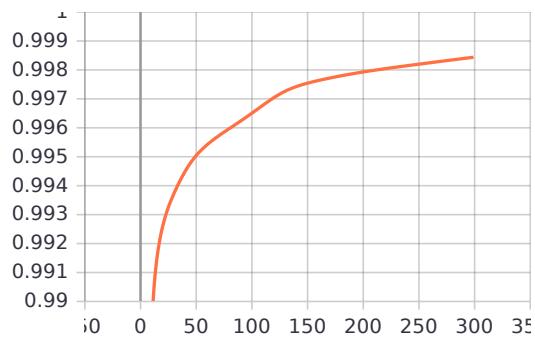


(a) Función de costo en entrenamiento

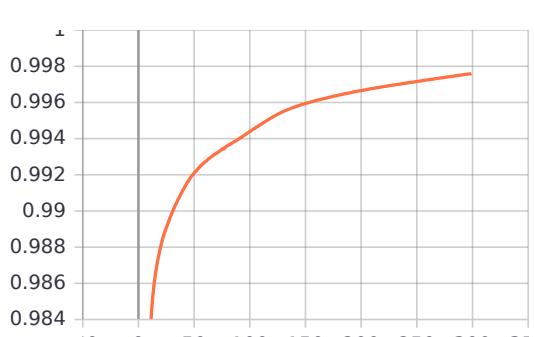


(b) Función de costo en validación

Curva de coeficiente de determinación

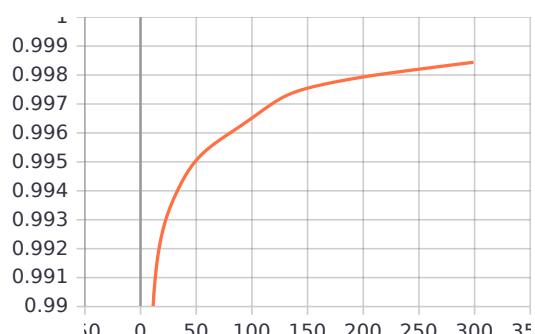


(a) Coeficiente de determinación en entrenamiento

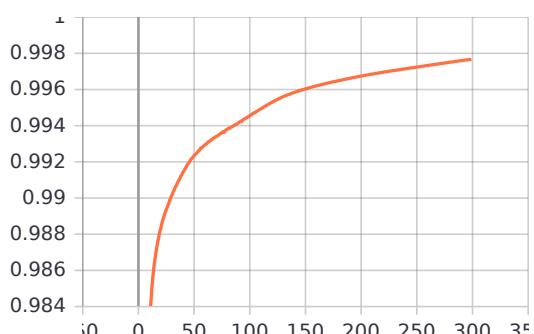


(b) Coeficiente de determinación en validación

Curva de varianza explicada



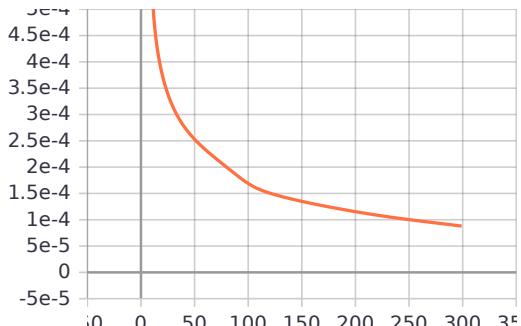
(a) Varianza explicada en entrenamiento



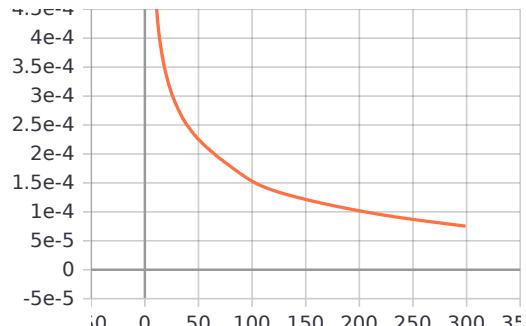
(b) Varianza explicada en validación

D.8.3. Arquitectura C

Curva de función de costo

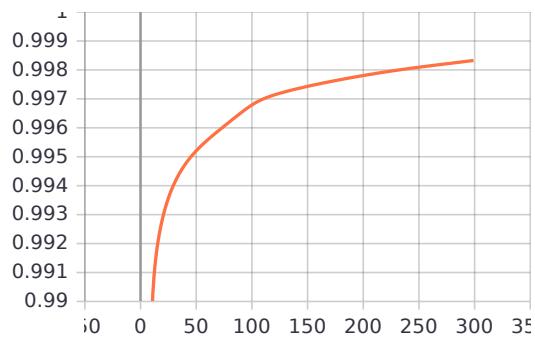


(a) Función de costo en entrenamiento

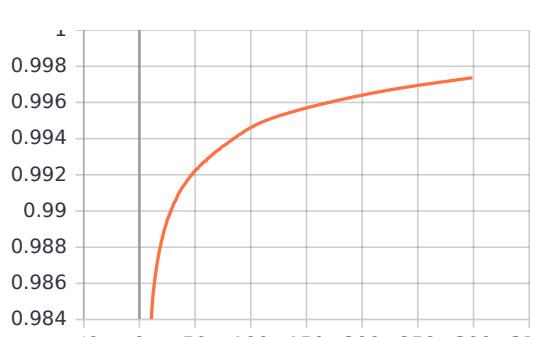


(b) Función de costo en validación

Curva de coeficiente de determinación

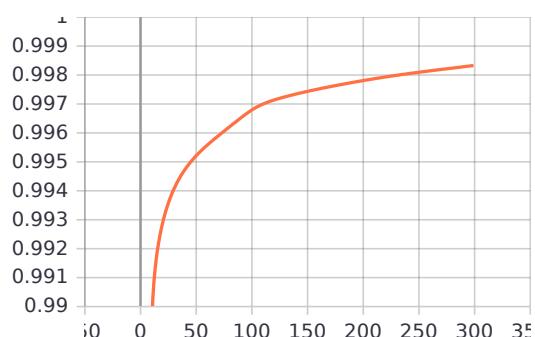


(a) Coeficiente de determinación en entrenamiento

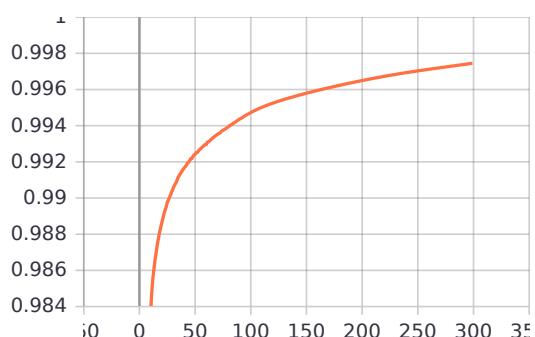


(b) Coeficiente de determinación en validación

Curva de varianza explicada



(a) Varianza explicada en entrenamiento



(b) Varianza explicada en validación

Referencias

Aha, D. W. (1997). Lazy learning.

Antona Cortés, C. (2017). *Herramientas modernas en redes neuronales: la librería keras* (B.S. thesis).

Došilović, F. K., Brčić, M., y Hlupić, N. (2018). Explainable artificial intelligence: A survey. En *2018 41st international convention on information and communication technology, electronics and microelectronics (mipro)* (pp. 0210–0215).

Gallardo Arancibia, J. A. (2009). *Metodología para el desarrollo de proyectos en minería de datos crisp-dm* (Inf. Téc.).

Géron, A. (2017). *Hands-on machine learning with scikit-learn and tensorflow: concepts, tools, and techniques to build intelligent systems*. .O'Reilly Media, Inc.

González, F. A. (2015). Machine learning models in rheumatology. *Revista Colombiana de Reumatología*, 22(2), 77–78.

Goodfellow, I., Bengio, Y., y Courville, A. (2016). *Deep learning*. MIT Press.
(<http://www.deeplearningbook.org>)

Hernández Orallo, J., FERRI RAMIREZ, C., y RAMIREZ QUINTANA, M. J. (2004). Introducción a la minería de datos.

Kieu, T., Yang, B., y Jensen, C. S. (2018). Outlier detection for multidimensional time series using deep neural networks. En *2018 19th ieee international conference on mobile data management (mdm)* (pp. 125–134).

- Kong, X., Everett, H., y Toussaint, G. (1990). The graham scan triangulates simple polygons. *Pattern Recognition Letters*, 11(11), 713–716.
- Lu, W., Cheng, Y., Xiao, C., Chang, S., Huang, S., Liang, B., y Huang, T. (2017). Unsupervised sequential outlier detection with deep architectures. *IEEE transactions on image processing*, 26(9), 4321–4330.
- Lyudchik, O. (2016). *Outlier detection using autoencoders* (Inf. Téc.).
- Ma, L., Gu, X., y Wang, B. (2017). Correction of outliers in temperature time series based on sliding window prediction in meteorological sensor network. *Information*, 8(2), 60.
- Maren, A. J., Harston, C. T., y Pap, R. M. (1990). Handbook of neural computing applications.
- Pajares, M., y De La Cruz, J. (2010). Aprendizaje automático un enfoque práctico. *RA-MA SA Editorial y Publicaciones. ISBN-10: 8499640117.*
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Du-chesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Pressman, R. S., y Troya, J. M. (1988). *Ingeniería del software* (n.º 001.64 P74s.). McGraw Hill.
- Rodrigo, B. (2018). *Intercambio de juguetes usados y análisis del desarrollo en la personalidad infantil.*
- Rose, K. H. (2013). A guide to the project management body of knowledge (pmbok® guide)—fifth edition. *Project management journal*, 44(3), e1–e1.
- Singh, A. (2017). *Anomaly detection for temporal data using long short-term memory (lstm).*
- Taquias, D., Villagra, A., y Errecalde, M. L. (2014). Detección de plagio con adversarios. En *Xvi workshop de investigadores en ciencias de la computación*.

- Urgiles, C. M. F., y Amoroso, M. S. O. (2018). Revisión de algoritmos para la detección de valores atípicos. *Killkana Técnica*, 2(1), 19–26.
- Yan, W., y Yu, L. (2015). On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach. En *Proceedings of the annual conference of the prognostics and health management society*.