



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

DESARROLLO DE UN MODELO PREDICTIVO DE ROBO A CASAS BASADO EN
REDES NEURONALES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL

PABLO ANDRÉS PLAZA ASCENCIO

PROFESORA GUÍA:
ASTRID CONTRERAS FUENTES

MIEMBROS DE LA COMISIÓN:
ÁNGEL JIMÉNEZ MOLINA
MAX ELMES ANGULO

Este trabajo ha sido financiado parcialmente por el proyecto Fondecyt 1181809

SANTIAGO DE CHILE
2020

**RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE:** Ingeniero Civil Industrial
POR: Pablo Andrés Plaza Ascencio
FECHA: 01/01/2020
PROFESORA GUÍA: Astrid Contreras Fuentes

DESARROLLO DE UN MODELO PREDICTIVO DE ROBO A CASAS BASADO EN REDES NEURONALES

En los últimos años se han desarrollado una amplia gama de métodos y modelos que ayuden a prevenir el actuar criminal. En varios países alrededor del globo han surgido investigaciones en pos de la seguridad social. Y es que, es un fenómeno que ha permanecido en la sociedad desde el origen de estas.

En particular, el delito de allanamiento es uno de los más preocupantes, ya que no solo hay daños materiales, sino que también psicológicos, dada la sensación de inseguridad con la que deben lidiar las víctimas de este tipo de delito. Se estima que en 2016 había 787 casos de allanamiento cada 100.000 habitantes para los primeros 10 países en donde este crimen era más recurrente [1].

La presente memoria tiene por objetivo determinar la probabilidad de ocurrencia del delito de robo a casas, el cual es una subcategoría del delito de allanamiento, por medio de un modelo basado en redes neuronales que permita la obtención de zonas de mayor y menor riesgo para una determinada locación. Para esto, se utiliza la base de datos de denuncias policiales de la ciudad de Dallas, la cual se encuentra pública en la web. Esta base contiene registros desde junio de 2014 hasta la fecha. A través de esta, se construyen distintas fuentes de información que permitan la construcción del modelo y la inclusión de diversos factores que afectan la predicción de robo según la literatura. Estas fuentes son: casas no robadas, datos de contexto extraídos de imágenes de Google Street View y de la red convolucional pre entrenada VGG16 en un proceso de extracción de características, puntos de interés cercanos, características demográficas y socioeconómicas, condiciones climáticas y cantidad de robos cercanos al delito en días previos a este.

Luego, se realiza una selección de características para la información contextual (imágenes) y espacial (información sociodemográfica y puntos de interés) con el objetivo de reducir la gran dimensionalidad de la base de datos y observar la importancia de las variables antes del uso del modelo final. Con esto se obtienen conjuntos con cantidades diferentes de características. Estos son utilizados para entrenar y evaluar el modelo final, el cual consiste en una red neuronal con múltiples entradas, las cuales son 3 en este caso (una para cada tipo de fuente de información). Las fuentes pasan por distintas capas, para luego unirse en una capa de concatenación, la cual finalmente origina una salida única que indica la probabilidad de robo para cada registro.

Como último paso, se evalúa la predicción del modelo para cada conjunto de características, comparando los resultados con el popular modelo de aprendizaje automático *Support Vector Machines*.

Agradecimientos

Con esta memoria culmina un ciclo de 19 años de educación en el cual hay muchísimo que agradecer. En primer lugar, quiero agradecer a mi familia más cercana, a mis padres, Marlene Ascencio y Carlos Plaza, y a mi hermano, Rodrigo Plaza quienes siempre estuvieron para apoyarme en todo sentido, enseñándome a confiar en mis capacidades y motivándome a superarme día a día. Ustedes fueron mi ejemplo a seguir durante todos estos años, por lo que este logro, indudablemente, también es suyo.

Quiero agradecer también a mis abuelos quienes además del apoyo presencial que me entregaron, sé que siempre me encontré en sus oraciones, hasta el día de hoy.

Por otro lado, agradezco a mis amigos y amigas, a los/as de mi colegio, a los del condominio y a los/as de universidad. Les agradezco por el constante apoyo en buenos y malos momentos. Son personas en las que sé que puedo confiar y sé que nuestra amistad seguirá por muchos años más.

Agradezco a todos quienes me ayudaron a la clasificación de casas. Sé que fue un trabajo tedioso, por lo que estoy muy agradecido de la paciencia y la perseverancia que le dedicaron. Sin la gran cantidad de gente que me ayudó, el trabajo se hubiese alargado aún más.

Agradezco también a los integrantes del WeSST Lab, lugar donde realicé esta memoria, específicamente a Cristian, Marcelo y Francisco, grandes personas quienes me recibieron de la mejor manera y siempre estuvieron dispuestos a ayudarme en caso de dudas. Mención especial a Francisco Díaz, quien fue un partícipe activo del presente trabajo aportando desde lo técnico y aclarando las no pocas interrogantes que me iban surgiendo a medida que avanzaba en el trabajo. Gracias Pancho por la paciencia y la buena onda.

También, quiero agradecer al profesor guía de esta memoria, Ángel Jiménez, quien confió en mí para realizar este trabajo y me apoyó en los momentos más difíciles del mismo.

Por último, agradezco a mi pareja y compañera de vida, quien estuvo conmigo durante toda la etapa universitaria, Andrea Faúndez. Gracias por confiar en mí y apoyarme en los altibajos que tuve durante la carrera. Gracias a ti soy mejor persona y siento que puedo hacer lo que me proponga.

Tabla de Contenido

1. Introducción.....	1
1.1 Motivación.....	1
1.2 Descripción del problema.....	2
1.3 Antecedentes: proyecto Fondecyt	5
1.4 Objetivos	5
1.4.1 Objetivo general.....	5
1.4.2 Objetivos específicos	5
1.5 Metodología	6
1.6 Resultados esperados y alcances.....	7
2. Marco teórico	9
2.1 Inteligencia artificial y aprendizaje automático	9
2.1.1 Aprendizaje supervisado	10
2.2 Modelos de aprendizaje automático.....	11
2.2.1 Regresión logística	11
2.2.2 Support vector machine (SVM).....	12
2.2.3 Random forest.....	14
2.3 Modelos de aprendizaje profundo	15
2.3.1 Redes neuronales	15
2.3.2 Red neuronal convolucional (CNN)	17
2.3.3 Red neuronal recurrente (RNN)	21
2.3.4 Funciones de activación: ReLU y Sigmoid.....	22
2.4 Métricas de evaluación.....	23
2.4.1 Matriz de confusión.....	23
2.4.2 Métricas utilizadas	24
2.5 Trabajos relacionados	25
2.5.1 Inicios en prevención del delito	25
2.5.2 Prevención del crimen a través del análisis de datos	26
2.5.3 Estudios y aplicaciones en la investigación criminal.....	26
2.5.4 Estudios de modelos predictivos de crimen	27
2.5.5 Estudios de modelos predictivos de robos a viviendas.....	28
3. Modelo y Recolección de Datos	31
3.1 Modelo.....	31
3.2 Base de datos de robos	33
3.3 Base de datos de puntos de no robo	34
3.3.1 Aplicación web: clasificador de imágenes	35
3.4 Base de datos espacial.....	37
3.4.1 Información demográfica y socioeconómica.....	37
3.4.2 Puntos de interés	40
3.5 Base de datos temporal	41
3.6 Base de datos de información contextual.....	42
3.6.1 VGG16	43
3.7 Base de datos final.....	44

3.8 Tecnologías involucradas.....	45
4. Selección de características	46
4.1 Composición de los conjuntos de características	50
4.1.1 Selección de 10.000 características	50
4.1.2 Selección de 1.000 características	50
4.1.3 Selección de 100 características.....	51
4.1.4 Selección de 10 características.....	51
4.2 Esquema general del procedimiento para la obtención de la probabilidad de robo a casas	52
5. Análisis de resultados	53
5.1 Ajuste de hiperparámetros de la red neuronal	53
5.2 Resultados del modelo	56
5.2.1 Tiempos de entrenamiento	56
5.2.2 Validación.....	57
5.2.3 Test	59
5.2.4 Características seleccionadas	66
6. Conclusiones y trabajo futuro	67
Bibliografía	69
Anexos	75

1. Introducción

1.1 Motivación

La delincuencia es un fenómeno que es relevante en prácticamente todas las ciudades del mundo, por lo que ocupa un lugar importante en la agenda política. Las formas de luchar contra esta han sido variadas, y abarcan soluciones a corto y a largo plazo. Una manera efectiva de disminuir la delincuencia, en cuanto a la asignación de recursos, es prevenirla, lo que implica anticiparse a las acciones delictivas antes de que sucedan.

Una de las formas de prevención que ha ido acaparando la atención dentro de la academia, es la predicción del crimen por medio de modelos de inteligencia artificial. Estos modelos logran aprender los patrones criminales a través de datos históricos. Diversos estudios se han originado en busca de este objetivo, y han sido utilizados para la movilización inteligente de fuerzas policiales. Por ejemplo, en Chile, se desarrolló un software que hace uso de modelos de inteligencia artificial para la predicción diaria de distintos tipos de delitos en todas las comunas del país, lo que conlleva a una distribución más efectiva de las fuerzas policiales, las cuales se concentran en las zonas de mayor riesgo según el software [2].

Aun cuando se han desarrollado modelos de distinta clase, existe un amplio abanico de soluciones no exploradas, puesto que el desarrollo de la inteligencia artificial continúa de manera acelerada en la actualidad. Esto motiva a investigar nuevas formas de realizar predicciones para la prevención del crimen, lo cual es el principal objetivo de la presente memoria.

Dada la extensa cantidad de tipos de crímenes existentes, se decide por concentrarse en uno de ellos, el allanamiento de morada o robo de vivienda (burglary of habitation en inglés). El principal motivo de la elección de este delito se debe a su naturaleza premeditada. Según diversos estudios [3], [4], [5], [6] las personas que cometen este delito seleccionan sus objetivos de forma racional, estudiando así, la estructura de la residencia, el sector circundante y el comportamiento de quienes la habitan, de manera de minimizar la probabilidad de ser capturados. Dado lo anterior, la evidencia sugiere que, al ser este un delito menos azaroso que otros, es posible descubrir patrones a través de los datos de robos, lo cual permitiría determinar aquellas residencias que tienen una mayor probabilidad de ser robadas, dadas las características de esta y sus alrededores.

Además, el robo de vivienda no solo tiene consecuencias materiales, sino que también psicológicas, dado que las personas víctimas de este delito, permanecen con la sensación de inseguridad luego de ocurrido el robo. Según un estudio de la empresa de seguridad *Verisure*, es común que las víctimas no duerman en sus moradas tras producirse un evento de robo, y que presenten un alto grado de inseguridad, teniendo dificultades para conciliar el sueño [7].

1.2 Descripción del problema

Este trabajo se realiza con los datos obtenidos de las denuncias policiales en la ciudad de Dallas. Esto se debe en primer lugar, a que dicha base de datos se encuentra disponible de forma pública en la página web: <https://www.dallasopendata.com/Public-Safety/Police-Incidents/qv6i-rrr7>, por lo cual es de fácil acceso. En segundo lugar, cada denuncia cuenta con la fecha y la dirección exacta no anonimizada del sitio del delito, lo cual es de suma importancia debido a las características del modelo desarrollado. Por último, se cuenta con las denuncias desde junio de 2014 hasta la fecha, siendo actualizada de manera diaria, lo que equivale a más de 500 mil registros. Esto presupone una cantidad adecuada de datos de robo de vivienda para la utilización de algoritmos de aprendizaje profundo [8].

En Estados Unidos, actualmente hay una tendencia a la baja en cuanto al crimen en casi todas sus categorías. En particular, el delito de allanamiento (burglary) disminuyó en un 7,6% entre 2016 y 2017 [9], sin embargo, representa un 18,2% del total de robos a la propiedad privada en el año 2017, lo cual equivale a más de 1,5 millones de casos de allanamiento. De esta forma, es relevante seguir disminuyendo esta cifra.

Los datos de delincuencia en la ciudad de Dallas, obtenidos en la página web antes mencionada, son un reflejo de lo que ocurre a nivel país. En la figura 1 se muestran las denuncias por categoría de delito. Estos datos pertenecen al periodo entre junio de 2014 y diciembre de 2018.

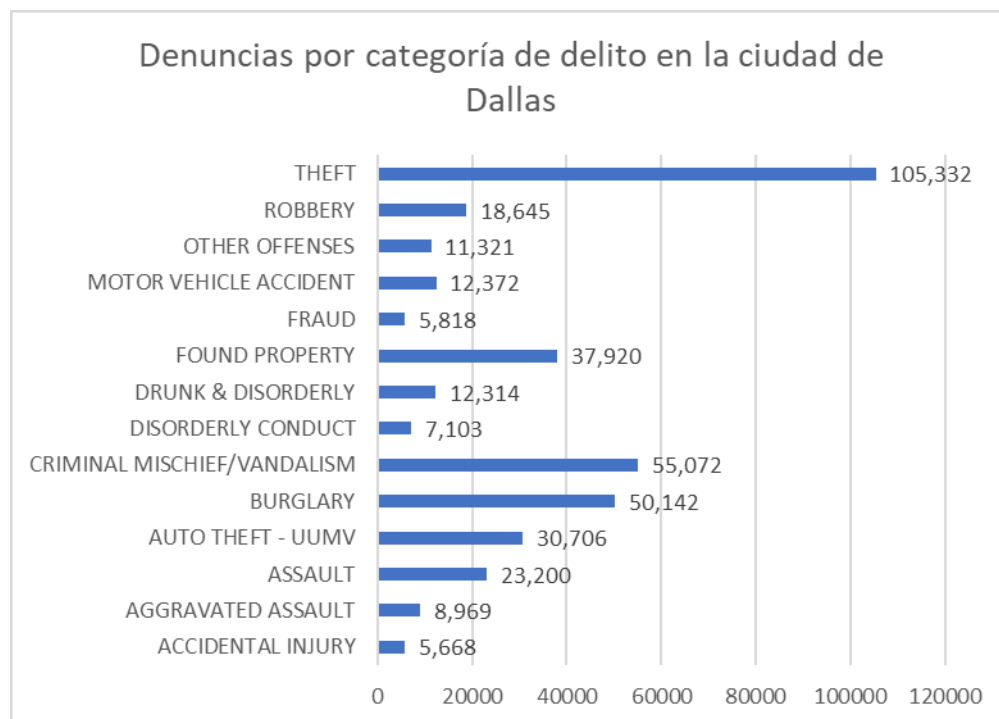


Figura 1.1: denuncias por categoría de delito desde junio de 2014 a diciembre de 2018.

Fuente: elaboración propia a través de la base de Dallas.

Como se muestra en la figura 1.1, el allanamiento (burglary) es el tercer tipo de delito que tiene más ocurrencia, luego de robo simple (theft) y vandalismo (criminal

mischief/vandalism). Se tienen más de 50 mil casos de delitos de allanamiento en los últimos años, lo que representa un 13% de la base de datos total. Es decir, cada 100 delitos, 13 son allanamientos en Dallas. Además, en la figura 1.2, se puede observar que la cantidad de denuncias de este delito por año ha tenido una leve baja en los últimos años, pero la cifra sigue siendo elevada en el año 2018.

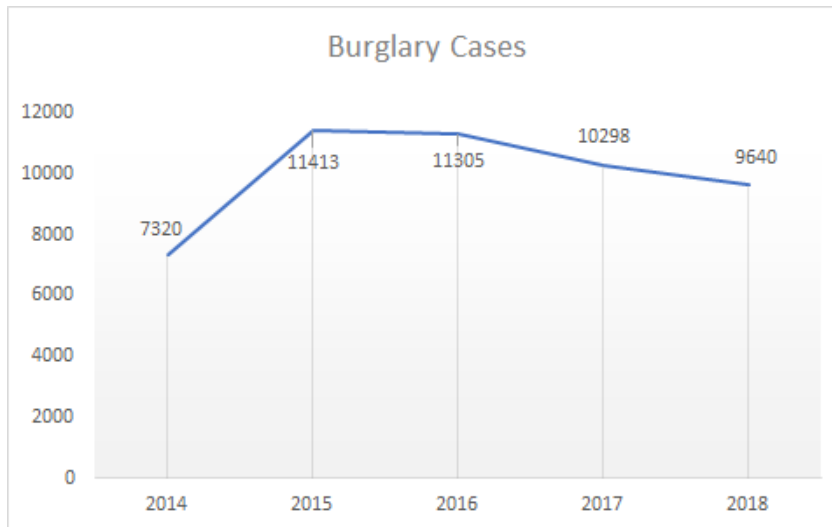


Figura 2: casos de allanamiento por año desde junio de 2014 a diciembre de 2018.
Fuente: elaboración propia a través de la base de Dallas.

Dentro de los delitos de allanamiento se tienen distintas categorías, entre las cuales se encuentran: robo a máquinas operadoras de dinero; robo de vivienda (burglary of habitation) con y sin entrada forzada; robo de edificio (burglary of building), donde se tienen robos a colegios, condominios, bencineras, entre otros, con y sin entrada forzada; e intento de robo de edificio y de vivienda. En la figura 1.3, se observan los tipos de allanamiento más recurrentes en la ciudad de Dallas.

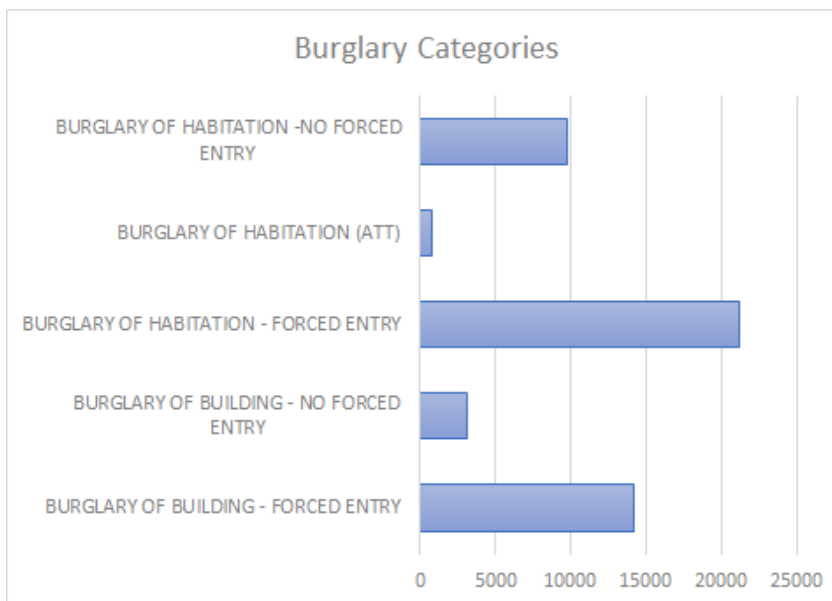


Figura 1.3: categorías de allanamiento con mayor cantidad de denuncias en la base de datos. Fuente: elaboración propia a través de la base de Dallas.

Se puede apreciar que son los robos con entrada forzada aquellos que registran mayor cantidad de denuncias, y dentro de estos, el robo de vivienda (burglary of habitation) registra más de 20 mil denuncias.

Por último, en la figura 1.4 se muestra el estado de los distintos casos de allanamiento, donde se puede ver que la mayor parte de estos delitos se encuentran suspendidos, lo cual quiere decir que después de pasado un tiempo no se tienen más sospechosos y, por tanto, el caso entra a dicha categoría, teniéndose una tasa de aprehendidos muy baja. Dado esto, se hace relevante poder predecir las zonas que tienen una mayor probabilidad de ocurrencia de este tipo de delito, de manera de prevenirlos o atrapar a delincuentes in fraganti.

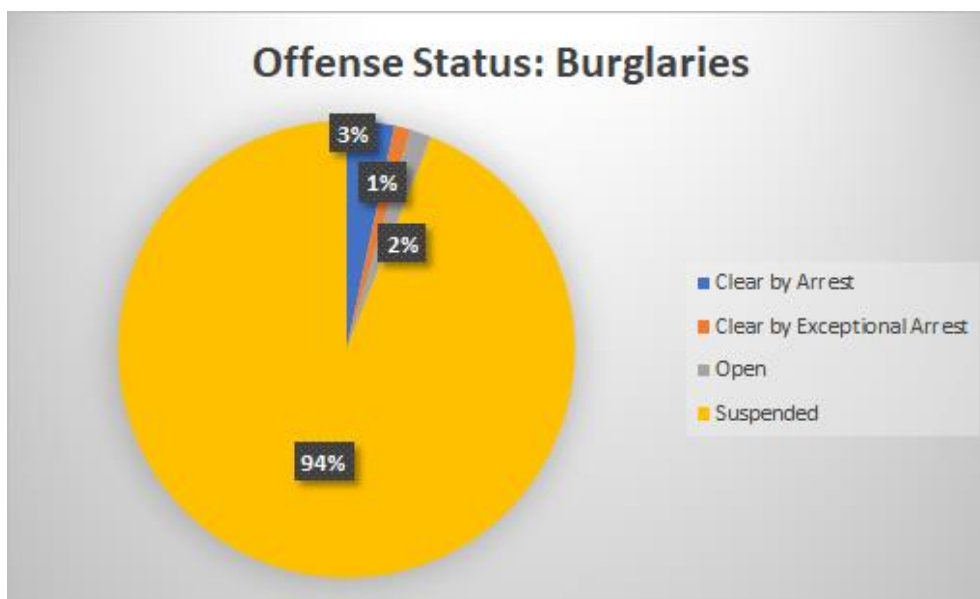


Figura 1.4: estatus de denuncias de allanamiento en Dallas, entre junio de 2014 a diciembre de 2018. Fuente: elaboración propia a través de la base de Dallas.

Es importante señalar que en este trabajo se utilizan datos de manera que estos sean lo más homogéneos posible, por lo que se decide considerar solo los casos de robo a casas, el cual es una subcategoría dentro del robo de vivienda. De esta forma se descartan los demás tipos de vivienda: condominios y departamentos. Esto, para lograr una mayor precisión en la predicción de robo. Los detalles con respecto al tratamiento de los datos se encuentran en la sección 3.

Dado lo anterior, el problema se aborda con el objetivo de predecir las probabilidades que tiene una casa de ser robada, a partir de las características de esta, del periodo de tiempo en el cual fue robada y del entorno en el cual se encuentra. Esto se puede realizar a partir de diversas técnicas de la ingeniería como el Aprendizaje Automático y el Aprendizaje Profundo. Los resultados podrían ser de ayuda a la policía de Dallas para la mejora en la eficiencia y efectividad en cuanto a la asignación de los recursos policiales, de manera de aumentar la prevención de estos siniestros en la ciudad.

1.3 Antecedentes: proyecto Fondecyt

Esta memoria se encuentra financiada por el Fondo Nacional de Desarrollo Científico y Tecnológico (Fondecyt) entregado por el Ministerio de Educación del Gobierno de Chile. Este Fondo fue ganado por la Universidad de Chile, siendo el Departamento de Ingeniería Civil Industrial y Eléctrica, las principales unidades académicas. El proyecto tiene código 1181809 y se titula “*Big data for sustainable smart cities with the aids of computational Intelligence*”.

El Big Data es reconocido actualmente como uno de los principales motores para promover la productividad y la eficiencia, y apoyar la innovación. Se espera que a través del análisis y el manejo de la gran cantidad de datos que al día se almacenan, se pueda generar valor para las diversas instituciones [10].

De esta manera, el objetivo de este Fondo es agregar valor al desarrollo sostenible de las ciudades a través de modelos que son posibles gracias a la gran cantidad de datos disponibles hoy en día y a la tecnología existente.

En este contexto es el profesor Ángel Jiménez (coinvestigador del Fondo y Profesor Asistente del departamento de Ingeniería Civil Industrial de la Universidad de Chile) quien está guiando la presente memoria. En particular, el trabajo se centra en el desarrollo de un modelo que permita determinar la probabilidad de robo a casas en la ciudad de Dallas, a través de los datos disponibles de este tipo de robo en los últimos años.

1.4 Objetivos

1.4.1 Objetivo general

El objetivo general de la memoria es determinar la probabilidad de robo a casas en la ciudad de Dallas, y determinar los factores que influyan en este tipo de delito, identificando así las casas más propensas a este. Para ello, se utiliza la base de datos pública de casos policiales antes mencionada, para diseñar y ejecutar un programa en python basado en redes neuronales que logre obtener las predicciones de robo de casas en las distintas áreas de la ciudad. Esto, con el propósito de ser un aporte a la amplia literatura que existe en el área de análisis criminal, y al Fondecyt en cuanto a sus objetivos de desarrollar ciudades más inteligentes.

La satisfacción de este objetivo será determinada según la factibilidad de construir este modelo, es decir, si efectivamente se pueden obtener probabilidades utilizando las fuentes de datos antes mencionadas. El modelo será evaluado a través de distintas métricas como accuracy, precision, recall y F1 que se explican en la sección 2 relativa al marco teórico.

1.4.2 Objetivos específicos

En primer lugar, se tienen objetivos relacionados a las estructuras de datos necesarias para generar el modelo:

- Construir una base de datos de casas que no han sido robadas en la ciudad de Dallas.
- Construir una base de datos de imágenes obtenidas desde Google Street View, tanto de casas robadas como no robadas.
- Para cada ubicación de casas robadas y no robadas, obtener los puntos de interés cercanos.
- Para cada registro de robo y no robo, obtener la zona censal en donde ocurre el evento.
- Para cada registro de robo y no robo, obtener información relativa al periodo (tiempo) en el que ocurre el suceso: fecha, condiciones climáticas y cantidad de robos de casas en la zona censal correspondiente.
- Caracterizar los lugares de Dallas en función de los puntos de interés, imágenes, condiciones demográficas y socioeconómicas, condiciones climáticas y ocurrencia del delito mencionado.

Luego, para encontrar patrones de robo dentro la ciudad y medir los resultados obtenidos:

- Estudiar características relevantes dentro de las fuentes de datos obtenidas, a fin de descubrir cuáles son más importantes para la predicción de robo.
- Desarrollar modelos de red neuronal en python, a través de las librerías correspondientes, lo cual permita la fusión, el entrenamiento y el aprendizaje de las distintas fuentes de datos.
- Obtener un clasificador de robo y no robo basado en las probabilidades de ocurrencia de robo.
- Ajustar los hiperparámetros de cada modelo con el objetivo de obtener una mejora en el rendimiento de cada uno.
- Analizar la capacidad predictiva de los modelos con el uso de métricas pertinentes.
- Comparar los resultados obtenidos entre el modelo de red neuronal propuesto (aprendizaje profundo) versus un modelo clásico de aprendizaje automático.

1.5 Metodología

Para cumplir con el objetivo propuesto de la memoria, se utilizará el proceso KDD (Descubrimiento de Conocimiento en Bases de Datos), el cual consiste en descubrir patrones novedosos y obtener información potencialmente útil y previamente desconocida, a través de los datos provenientes de algún repositorio.

Este proceso consta de 5 fases, las cuales son explicadas a continuación, así como su aplicación en esta memoria.

- a. *Selección de datos*: en la cual se determinan las fuentes de datos y el tipo de información a utilizar. Luego, los datos son extraídos desde las fuentes. Los datos que se utilizan en esta memoria son: denuncias de delitos de robos a casas en la ciudad de Dallas, Estados Unidos; puntos donde no ocurrieron robos en la ciudad de Dallas; imágenes obtenidas a partir de Google Street View, correspondientes a los lugares de robo y no robo. Se agrega también, puntos de interés cercanos a cada punto como centros comerciales, bencineras, estadios, restaurantes, bares, etc. Adicionalmente, se obtienen datos demográficos y socioeconómicos que ayuden a caracterizar las zonas de robo y no robo dentro de

Dallas. Por último, se utiliza información temporal, representada por la temperatura y nivel de precipitación de días anteriores, y robos en las cercanías de cada punto en días anteriores.

- b. *Preprocesamiento de los datos*: aquí se preparan y se limpian los datos extraídos para que puedan ser correctamente incluidos en los modelos posteriores.

En el caso de la información contextual proveniente de imágenes, se utilizan redes neuronales convolucionales para la extracción de características. Para la información temporal, se estructura en forma de series temporales, dado que se incluye información pasada. Los puntos de interés son categorizados de forma de obtener la cantidad de puntos cercanos por categoría para cada registro.

- c. *Transformación*: tratamiento preliminar de los datos. Se pueden agregar nuevas variables a partir de las presentes en la base, y en caso de ser necesario, se realizan operaciones como normalización o agregación de los datos.

En el caso de este trabajo, se realiza estandarización de los datos antes de la entrada a los modelos.

- d. *Minería de Datos*: esta es la fase de modelamiento, donde se aplican distintos métodos para extraer patrones e información relevante a partir de los datos.

Para esta memoria, en primer lugar, se utiliza la selección de características para la información espacial y contextual, de manera de crear conjuntos con diferentes cantidades de características. Luego, se construye una arquitectura de red neuronal con una entrada para cada fuente de información y una capa de concatenación para fusionar las entradas más adelante. Finalmente, se entrena un clasificador por medio de capas totalmente conectadas y una capa de salida con función de activación “sigmoid”, lo cual entrega las probabilidades para la clase robo. Con estas probabilidades se obtiene finalmente la clasificación de cada registro y, por tanto, la predicción de robo a casas.

- e. *Interpretación y evaluación*: se interpretan e identifican los patrones realmente interesantes, haciendo evaluación y medición de los modelos y resultados obtenidos.

Se compara el resultado de la red neuronal construida con un modelo clásico de aprendizaje automático como lo es SVM. En este caso, se usarán las métricas clásicas para evaluar los modelos de aprendizaje automático, a saber, *accuracy*, *precisión*, *recall* y *f1*, estas son explicadas en la sección 2 del trabajo.

1.6 Resultados esperados y alcances

Los resultados esperados de esta memoria son los siguientes:

- Caracterización de las distintas zonas de Dallas a través de la selección de variables provenientes de las fuentes de datos.
- Diseño de solución válido, tanto en los modelos utilizados para las distintas fuentes, como en la arquitectura utilizada para la fusión de estas.
- Análisis sobre factores que influyen en el robo a residencias en la ciudad de Dallas.
- Códigos funcionales en python que permitan replicar el trabajo realizado en esta memoria.

Con respecto a los alcances del trabajo, se encuentra determinar la probabilidad de robo de una propiedad en cualquier zona de la ciudad de Dallas mediante el desarrollo de un programa en python y de algoritmos de minería de datos e inteligencia artificial.

Además, el trabajo solo contempla la predicción para el robo a casas, descartando la predicción para otros delitos y para otros tipos de infraestructura. Por otro lado, sólo se estudia el efecto de algunos factores asociados a la predicción del robo a casas según la literatura. Estos son: ambientales, demográficos, socioeconómicos y temporales, descartándose, por ejemplo, entrevistas a victimarios, víctimas y testigos. En la sección 3 de modelo y recolección de datos, se explica el detalle de las variables escogidas.

Por último, el presente trabajo no tiene por objetivo alcanzar un cierto rendimiento en cuanto a la predicción de robo, sino que, se busca alcanzar el mejor rendimiento posible.

Se escoge la ciudad de Dallas, dado que la base de datos de denuncias policiales se encuentra pública en la web y cuenta con la dirección de cada hecho, lo que es un requisito clave para la realización de este trabajo. Además, es una ciudad grande dentro de Estados Unidos con más de 1,3 millones de habitantes y la tercera en el país con mayor área metropolitana. Esto, conlleva a que haya una gran cantidad de robos en comparación con otras ciudades, lo que es relevante a la hora de utilizar técnicas de aprendizaje automático.

2. Marco teórico

En esta sección se explican los conceptos y definiciones necesarios para comprender el trabajo realizado. Además, se exponen distintas aplicaciones presentes en la literatura y que son pioneras en cuanto a las tecnologías aquí utilizadas en el campo del análisis criminal.

2.1 Inteligencia artificial y aprendizaje automático

La inteligencia artificial hace referencia a la capacidad de las máquinas (generalmente, sistemas informáticos) para simular el comportamiento humano con respecto a procesos de inteligencia [11]. Debido a las mejoras constantes en tecnología, hoy en día se hace posible la obtención de una gran capacidad computacional a un precio accesible, lo que ha permitido un mayor uso de técnicas de inteligencia artificial en la academia y también en la industria. Por ejemplo, actualmente es utilizada en tecnologías de ciber seguridad, cirugía autónoma por medio de la robótica, robótica en la nube, traducción en tiempo real, compañeros virtuales, análisis de emociones en tiempo real, chatbots, procesamiento de lenguaje natural, sistemas autónomos y modelamiento predictivo a base de reconocimiento de patrones y aprendizaje automático. Esto último, es lo que se utiliza para cumplir con los objetivos de este trabajo.

Como señala la revista *The Wall Street Journal*, “en los últimos años se han unido los ingredientes necesarios para llevar a la IA a través del umbral: tecnologías informáticas potentes y económicas; enormes cantidades de datos; y algoritmos avanzados, especialmente el aprendizaje automático” [11].

El aprendizaje automático (machine learning en inglés), es una rama de la inteligencia artificial la cual utiliza métodos estadísticos para permitir la mejora de programas informáticos a través de la experiencia [12]. Aplicado al caso de los ordenadores, se trata de entregarles la habilidad de aprender sin que esté explícitamente programado [13]. Su objetivo es encontrar una función $f(x)$, donde tanto f como x pueden representar un problema de varias variables, que sea capaz de encontrar patrones en un conjunto de datos [14]. Existen 3 métodos de aprendizaje: supervisado, no supervisado y reforzado.

El aprendizaje supervisado se presenta al tener un problema de aprendizaje automático con etiquetas de clase. Esto quiere decir que la base de datos contiene ejemplos con una clasificación previa. Los algoritmos de aprendizaje automático supervisado son alimentados por estos datos con el objetivo de que aprendan patrones según su clasificación, luego se permite que el algoritmo prediga la clase para cada ejemplo, entregando una retroalimentación con respecto a si la predicción fue correcta o no. Este procedimiento se repite una cantidad dada de veces con el objetivo de que el modelo aprenda y haga predicciones cada vez mejores. El problema de predecir y hacer una clasificación de robo o no robo para una casa, como el que presenta este trabajo, corresponde a un problema de aprendizaje supervisado.

El aprendizaje no supervisado es opuesto al anterior, dado que los datos no poseen una clasificación explícita, por lo que los algoritmos de este tipo de aprendizaje buscan

encontrar patrones en los datos y agruparlos en conjuntos, de forma de dar sentido a la información. Los sistemas de recomendación como es el caso de Netflix o empresas de retail, al recomendar productos relacionados a las compras de sus clientes, son ejemplos de aprendizaje no supervisado.

Por último, está el aprendizaje reforzado, el cual se basa en la premisa de aprender de los errores. Los algoritmos pertenecientes a este tipo de aprendizaje son alimentados por señales distintas según el comportamiento del algoritmo. De esta forma, se refuerza al modelo para preferir comportamientos buenos, y así aprender a cometer menos errores en cada iteración. Algunos casos de uso más comunes se presentan en el entorno de los videojuegos, donde se les enseña a programas a jugar; y en simulación industrial, donde, por ejemplo, se automatizan procesos a través de aplicaciones robóticas.

2.1.1 Aprendizaje supervisado

Como se mencionó anteriormente, el aprendizaje supervisado se presenta al tener datos previamente clasificados y su objetivo es la correcta clasificación para nuevos datos, por lo tanto, corresponde al tipo de aprendizaje que se utiliza en la memoria al querer realizar una predicción de robo para casas en Dallas sobre registros que se sabe de antemano si corresponden a viviendas robadas o no robadas.

Sin importar el caso, el procedimiento para la resolución de un problema de aprendizaje supervisado consiste en la división del conjunto de datos en 3 subconjuntos:

- Conjunto de entrenamiento: se utiliza para la búsqueda de la función que ajuste a los datos.
- Conjunto de validación: es utilizado para observar una primera generalización del modelo. Dado lo anterior, este conjunto debe contener datos que no han sido utilizados previamente para el entrenamiento, de manera de comprobar el funcionamiento del algoritmo ante datos que no conoce. A través de los resultados de este conjunto, se prueban distintos modelos e hiperparámetros para lograr un mejor aprendizaje.
- Conjunto de prueba: los resultados en este conjunto representan el poder de generalización final del modelo. Dado esto, debe ser independiente de los dos anteriores y es utilizado para testear el ajuste de los hiperparámetros previamente realizado en el conjunto de validación.

El algoritmo es evaluado finalmente según las métricas que se presentan en el conjunto de prueba. Estas métricas son las comúnmente utilizadas en los algoritmos de aprendizaje automático y serán detalladas más adelante. Los algoritmos que se utilizan en la memoria siguen la lógica presentada.

Existen dos tipos de aprendizaje supervisado, lo cual dependerá del problema que se busque resolver. Estos son: problemas de regresión y de clasificación.

Los problemas de regresión se presentan cuando el valor de salida es continuo, esto quiere decir, que los datos presentan una clasificación por medio de valores no discretos, por lo tanto, el objetivo es encontrar una función f que mapee las observaciones x de tal

forma que $f(x)$ esté cerca de los valores observados. En la figura 2.1 se presenta un ejemplo de regresión.

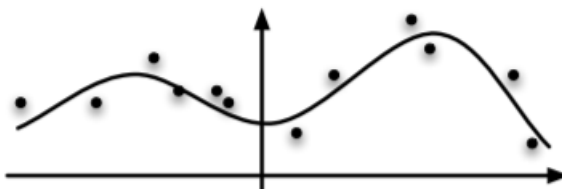


Figura 2.1: ejemplo de regresión. Los puntos representan las observaciones x y la línea continua la función f , que busca un ajuste lo más próximo a las observaciones. Fuente: [15].

Los problemas de clasificación, en cambio, son aquellos en los que el valor de salida o etiqueta es discreto. De esta forma, se busca una función que sea capaz de separar los conjuntos, para luego poder clasificar de forma correcta un dato entrante. Dependiendo de las etiquetas presentes en el conjunto de datos se tiene un problema de clasificación binario o de varias clases. En la clasificación binaria, el modelo predice sí o no (0 o 1), mientras que, en la clasificación de varias clases, el modelo predice más de dos clases. En la figura 2.2 se muestra un ejemplo de ambos problemas.

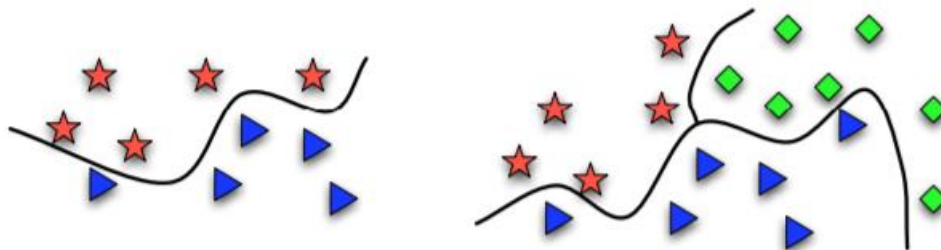


Figura 2.2: a la izquierda: problema de clasificación binaria. A la derecha: problema de clasificación para 3 clases. Fuente: [15].

El problema presentado en esta memoria corresponde a un problema de clasificación binaria con clases robo y no robo.

2.2 Modelos de aprendizaje automático

En esta sección se presenta, en primer lugar, uno de los modelos más populares e importantes del aprendizaje automático, la regresión logística. Luego, se presentan otros modelos que son utilizados en esta memoria.

2.2.1 Regresión logística

Este modelo es un tipo de regresión que es ampliamente utilizado para los problemas de clasificación binaria, en donde el output que se quiere obtener es 1 o 0, y en general representan una respuesta “positiva” o “negativa” respectivamente, ante la ocurrencia o no ocurrencia del fenómeno estudiado.

La ocurrencia del fenómeno estará dada por una cierta probabilidad, que en la regresión logística se expresa a través de la función logit:

$$p(x_i) = \frac{1}{1 + e^{-(\beta_0 + \sum \beta_i x_i)}}$$

Esta función tiene la característica de ser suave, por lo que se comporta de forma correcta en los puntos de corte y, por tanto, resulta ser ideal para problemas de clasificación. Esta clasificación será determinada según el punto de corte (\bar{X}) a través de la siguiente forma:

$$\begin{aligned} p(x_i) \geq \bar{X} &\rightarrow y_i = 1 \\ p(x_i) < \bar{X} &\rightarrow y_i = 0 \end{aligned}$$

Dadas estas características, es largamente utilizado en la literatura. En el caso de este trabajo, si bien las probabilidades de robo y no robo no se modelan de la misma forma, también se utilizan probabilidades para la clasificación final, con un punto de corte de 0.5.

2.2.2 Support vector machine (SVM)

SVM corresponde a un tipo de algoritmo de aprendizaje supervisado y, aunque puede ser utilizado tanto para problemas de clasificación como de regresión, es comúnmente utilizado en los primeros [16].

Al ser un problema de clasificación el que se aborda en este trabajo, la explicación del algoritmo se basa en la clasificación.

La idea principal del algoritmo es encontrar el hiperplano que mejor divida un conjunto de datos en clases. Un ejemplo para un problema lineal de clasificación binaria se aprecia en la figura 2.3.

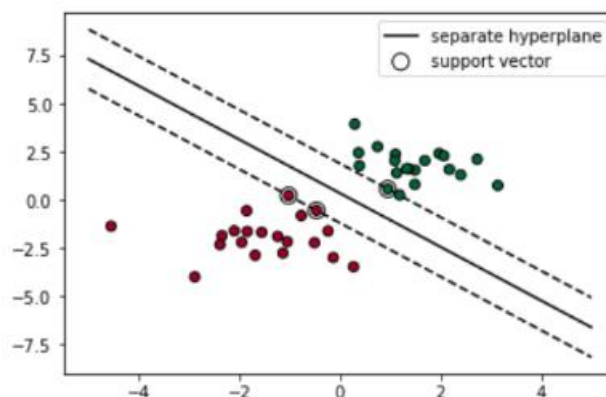


Figura 2.3: ejemplo del funcionamiento de SVM para un problema lineal de clasificación binaria. Fuente: [17].

Como se observa en la figura 2.3 los vectores de soporte están definidos por los puntos más cercanos al hiperplano que separa las clases, y tienen ese nombre, dado que, de eliminarse dichos puntos, la posición del hiperplano se modificaría. El hiperplano, por otro lado, es la línea (en este caso) que divide ambos conjuntos, por lo que el algoritmo de SVM tiene por objetivo encontrar esta división maximizando el margen entre una clase y

otra, es decir, maximizando las distancias perpendiculares entre los vectores de soporte y el hiperplano.

En el caso mostrado anteriormente, las clases son claramente separables, pero en la mayoría de los conjuntos de datos de la vida real, no se podrá hacer una división perfecta. En estos casos, se utilizan dos soluciones para abordar este problema:

- **Soft Margin:** se utiliza una penalización para los casos mal clasificados, buscando el equilibrio entre la maximización del margen y la minimización de la clasificación errónea. En la figura 2.4, se puede ver cómo varía la solución del algoritmo, modificando la penalización C .

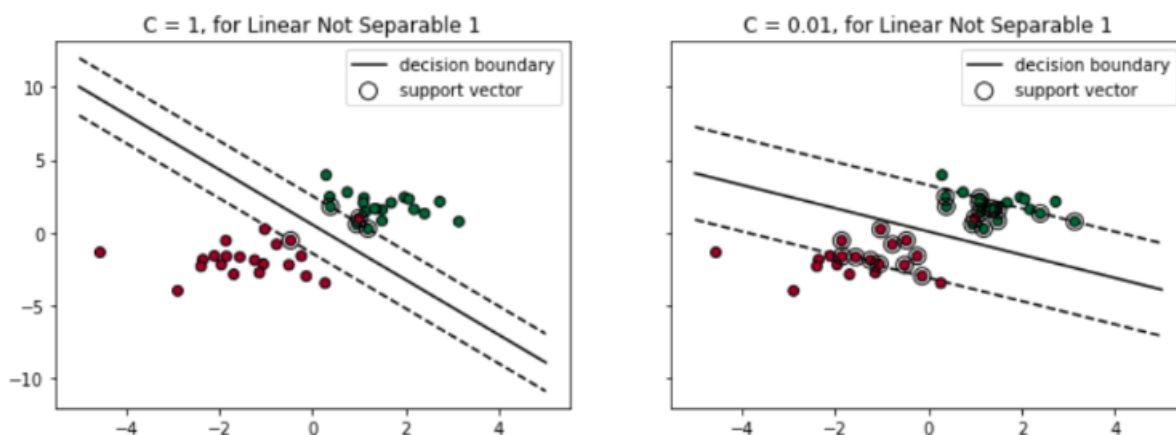


Figura 2.4: variación de SVM ante cambios en la penalización C . Fuente: [17].

- **Kernel Tricks:** se utiliza para aplicar transformaciones a funciones existentes y así crear nuevas para generar hiperplanos no lineales. En la figura 2.5, se aprecia cómo cambia la forma del hiperplano ante cambios en el límite de decisión.

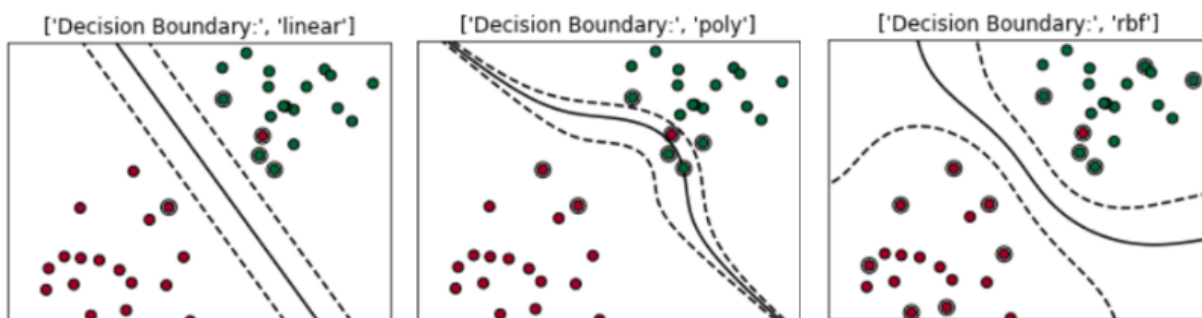


Figura 2.5: variación de la solución del SVM ante cambios en el límite de decisión. Fuente: [17].

Las figuras antes mostradas se crearon a partir de librerías especializadas en la implementación de SVM. En este trabajo, se hace uso de estas librerías en python para la implementación del algoritmo. Se prueba el ajuste de los hiperparámetros mencionados con el objetivo de encontrar un mejor rendimiento del modelo SVM.

2.2.3 Random forest

Random forest o bosque aleatorio es un modelo que se basa en la construcción de múltiples árboles de decisión. El árbol de decisión es una estructura que consta de distintos niveles en donde los datos se van separando de acuerdo con el cumplimiento de ciertas restricciones. Por ejemplo, si se considera el problema presente en la memoria de predicción de robo y se toman las variables de: tamaño de casa promedio, ingreso per cápita de la comuna y edad promedio de la población en dicha comuna, se puede construir un árbol de decisión de tal forma que el rango de probabilidad de robo varíe de acuerdo con la información de las variables. Finalmente, se obtiene una clasificación de acuerdo con las probabilidades obtenidas. En la figura 2.6, se puede ver un ejemplo de árbol de decisión. La clasificación es meramente manual y no representa los datos del problema.

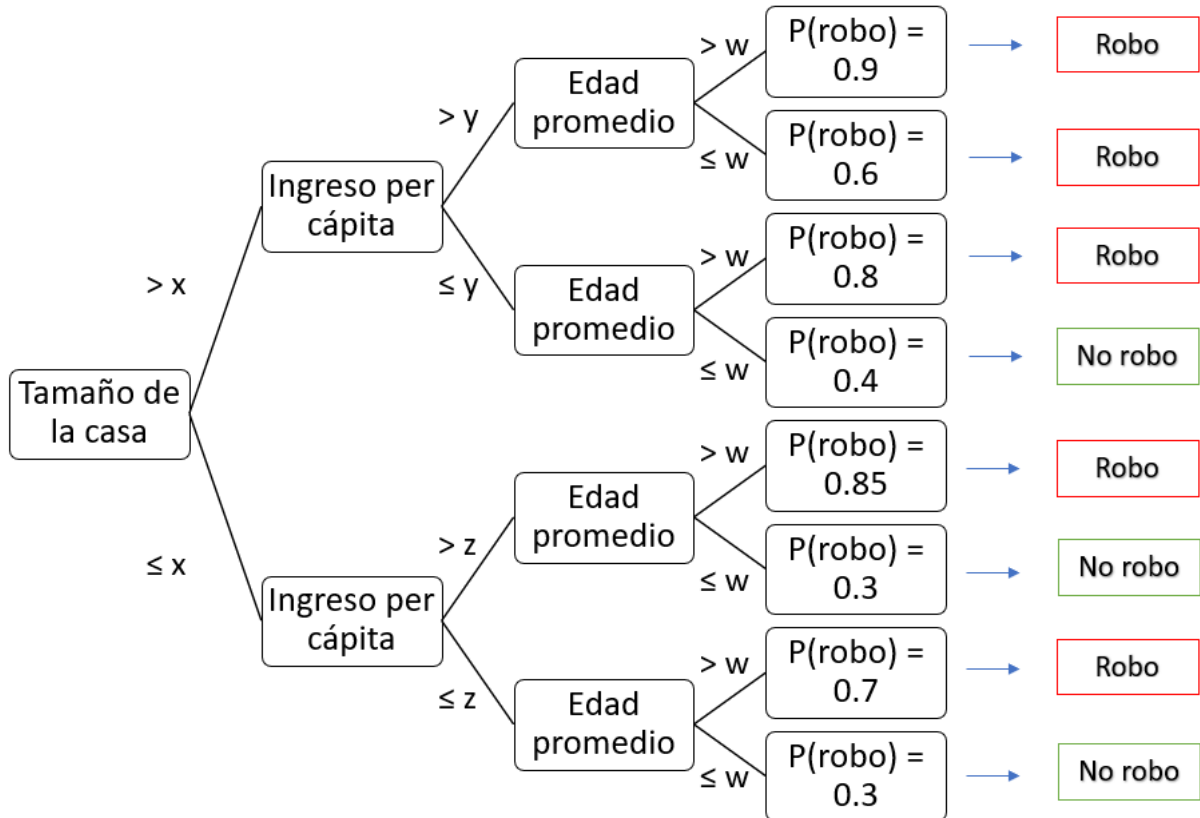


Figura 2.6: ejemplo de árbol de decisión para la predicción de robo a casas. Ejemplo ilustrativo. Fuente: Elaboración propia.

En la figura 2.6 se observa que sin importar el dato que se tenga, siempre se llegará a una clasificación final.

El bosque aleatorio combina muchos árboles de decisión diferentes, esto quiere decir, que las variables se pueden presentar en distinto orden, los puntos de corte de cada variable pueden ser diferentes o incluso se pueden tener más ramificaciones por nivel. La idea fundamental detrás, es que la clasificación de un solo árbol de decisión puede no ser muy precisa, pero cuando se combinan muchos, el resultado será más acertado en promedio. El nombre contiene la palabra “aleatorio”, dado que cada árbol considera un

subconjunto aleatorio de características al pasar por los distintos niveles y solo tiene acceso a un conjunto aleatorio de datos de entrenamiento [18]. Esto conduce a una mayor diversidad en el modelo y a predicciones generales más robustas. Para el caso de los problemas de clasificación, la predicción final para cada ejemplo se hace a través de voto mayoritario de los distintos árboles que conforman el bosque.

En el contexto de la memoria, el bosque aleatorio se utiliza para realizar una selección de características. Esto, debido a la gran cantidad de variables que se tienen y la intención de observar cómo influyen estas en la predicción de robo. Los detalles de su uso y los hiperparámetros presentes en el modelo serán explicados en la sección 4 de selección de características.

2.3 Modelos de aprendizaje profundo

El aprendizaje profundo (deep learning en inglés), es una rama del aprendizaje automático y funciona de manera similar a este último. Por tanto, para un gran abanico de problemas es posible la utilización tanto de técnicas de aprendizaje automático como de aprendizaje profundo, sin embargo, la diferencia principal entre ambos se relaciona con las capacidades de cada uno. En el aprendizaje automático, aún existe un trabajo manual importante por parte de la persona, de manera que, para que el algoritmo obtenga una predicción más precisa, es necesario entregarle más información. Por otro lado, los algoritmos de aprendizaje profundo están diseñados para que aprendan a través de su propio procesamiento de los datos [19]. Esto se logra a través de la construcción de modelos que imiten el cerebro humano, por medio de neuronas artificiales.

Para la predicción de robo, se utiliza una red neuronal con distintas entradas, la cual corresponde a un modelo de aprendizaje profundo. Además, se utilizan otras técnicas de este enfoque para el manejo de imágenes y de la información temporal. De esta forma, en primer lugar, se explica el funcionamiento general de las redes neuronales y luego se ahonda en los conceptos asociados a las técnicas utilizadas.

2.3.1 Redes neuronales

Las redes neuronales están inspiradas en el funcionamiento del cerebro humano por medio de las neuronas biológicas. La unidad básica de la red neuronal artificial es la neurona, y su funcionamiento, así como la comparación con la neurona biológica, se puede apreciar en la figura 2.7:

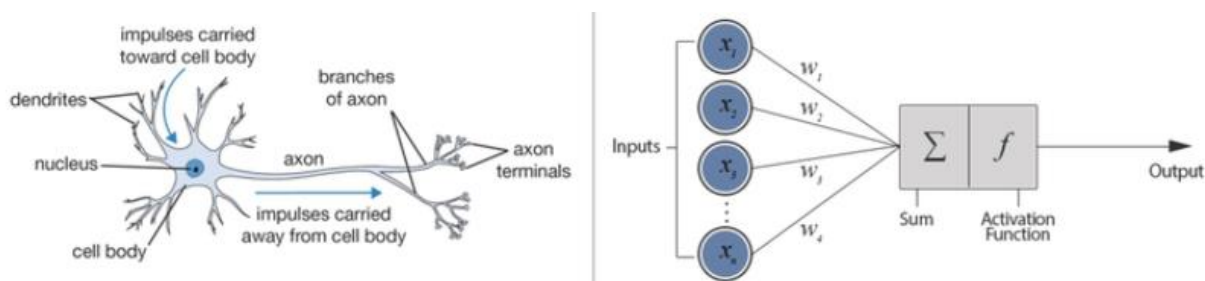


Figura 2.7: comparación entre neurona biológica (izquierda) y neurona artificial (derecha). Fuente: [20].

Como se aprecia en la figura 2.7, las entradas o inputs en la neurona artificial son las variables que se encuentren en el conjunto de datos. Estas representan a las dendritas. Luego, se tienen distintos pesos que multiplican a cada variable, para ser posteriormente sumados. Esto representa al núcleo y cuerpo de la neurona biológica. Finalmente, existe una función de activación que de sobrepasar un cierto umbral da como resultado una cierta salida, lo cual emula el funcionamiento de la neurona cerebral que recibe impulsos eléctricos. Si estos son lo suficientemente potentes y sobrepasan un cierto umbral de excitación, se transmitirá un nuevo impulso a través del axón, lo cual llegará a otras neuronas a través de sus terminales [21].

Lo anterior se conoce como red neuronal artificial de una sola capa o perceptrón, y corresponde al caso en el cual se tiene una sola neurona. Para problemas más complejos, se utilizan múltiples neuronas y una o más capas. Por lo tanto, para clarificar el funcionamiento de la neurona, y generalizando para múltiples neuronas, se tiene que la activación quedará descrita por:

$$u_j = \sum_{i=1}^n x_i w_{ij} \quad (2.1)$$

Donde u_j es la activación de la neurona de la neurona j , x_i es la variable de entrada i -ésima y w_{ij} es el peso sináptico de la variable i en la neurona j . La función de activación f es evaluada en la activación u sumado a un sesgo o bias, obteniéndose un valor de salida. Existen diversas funciones de activación que son mayormente utilizadas debido a sus buenas propiedades con respecto a continuidad y derivabilidad. De esta manera, la salida tendrá la siguiente forma:

$$y_j = f(u_j + b_j) \quad (2.2)$$

Donde y_j es la salida de la neurona j , u_j es la activación de la neurona j y b_j es el sesgo de la neurona j .

El esquema general de una red neuronal se muestra en la figura 2.8.

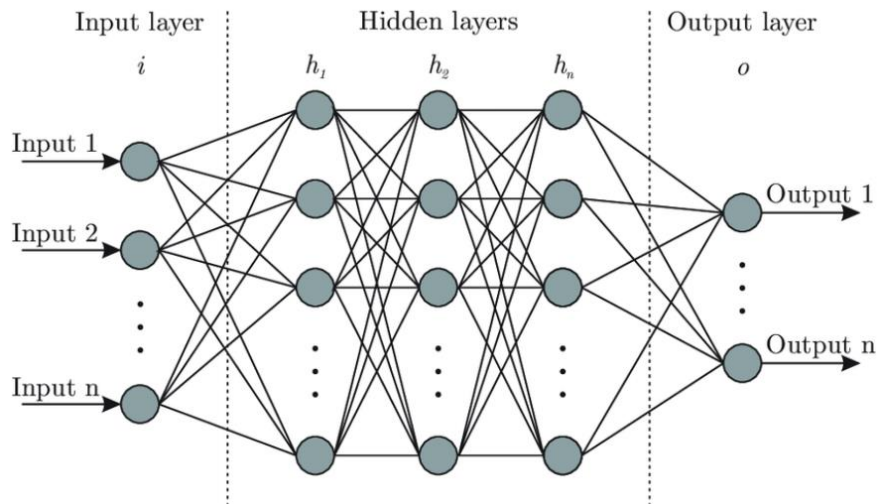


Figura 2.8: arquitectura general de una red neuronal artificial. La cantidad n no es necesariamente la misma para la entrada, las capas ocultas y la salida. Fuente [22].

De esta forma, las redes neuronales artificiales se componen por tres tipos de capas:

- Capa de entrada: donde entran las variables que se tienen en el conjunto de datos. La cantidad de neuronas en esta capa será igual a la cantidad de variables que se tengan.
- Capas ocultas: pueden ser una o más. Sirven para establecer las relaciones entre variables. Mientras más capas, más complejas serán las funciones que se consideran para la resolución de un problema. La cantidad de neuronas varía dependiendo de la cantidad de parámetros que se quiera incluir en el modelo.
- Capa de salida: salida final del modelo. En el caso de esta memoria, da como resultado la probabilidad de robo para una cierta casa. La cantidad de neuronas dependerá de la salida que se busque. Para problemas de múltiples clases, es común tener una neurona para cada clase, lo que entrega una probabilidad para cada una.

Como se mencionó anteriormente, las redes neuronales pueden variar en la cantidad de capas ocultas, y estas, además, pueden ser de distintos tipos. Las redes neuronales más comunes están conformadas por capas densas o fully connected, en donde sus neuronas están conectadas a todas las neuronas de la capa siguiente como se aprecia en la figura 2.8. Así, las neuronas de las capas ocultas y las de la capa de salida, en el caso de capas densas, quedan descritas por las ecuaciones 2.1 y 2.2.

2.3.2 Red neuronal convolucional (CNN)

Esta red se especializa en procesar datos que tienen una estructura similar a una cuadrícula, como una imagen. Es por esto que, generalmente, es utilizada para tareas de visión artificial, como en la clasificación y segmentación de imágenes [23].

Una imagen digital contiene una serie de píxeles dispuestos en forma de cuadrícula y sus valores indican qué tan brillante y de qué color es cada píxel. Esto se puede observar en la imagen 2.9.

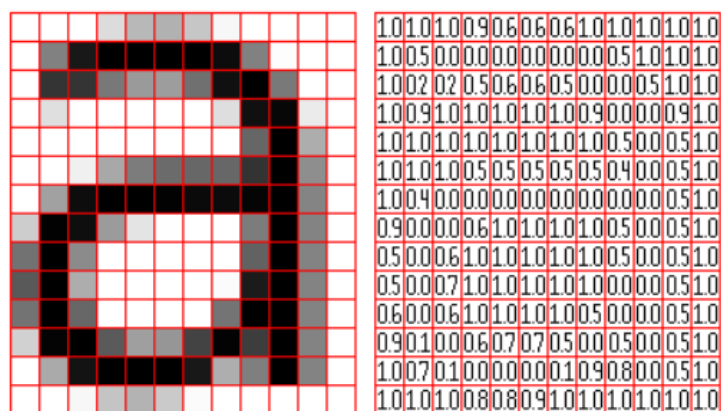


Figura 2.9: Representación de una imagen como una cuadrícula de píxeles. Fuente: [23].

Una red neuronal convolucional consta de una capa de entrada (imagen), seguida por las capas ocultas (convolución, agrupación y densa), y finalmente la salida. Esta disposición permite a la red aprender patrones más simples primero, como líneas o curvas, y luego patrones más complejos, como caras u objetos.

La capa de convolución lleva dicho nombre, puesto que realiza una operación llamada *convolución*. En este contexto, la convolución es una operación lineal que implica el producto punto de una matriz, llamada *kernel* (núcleo), y que representa el conjunto de pesos o parámetros que la red puede aprender, con otra matriz que representa una porción de la imagen, también llamada campo receptivo, y cuyo tamaño depende del tamaño del núcleo.

Lo anterior es representado en la figura 2.10:

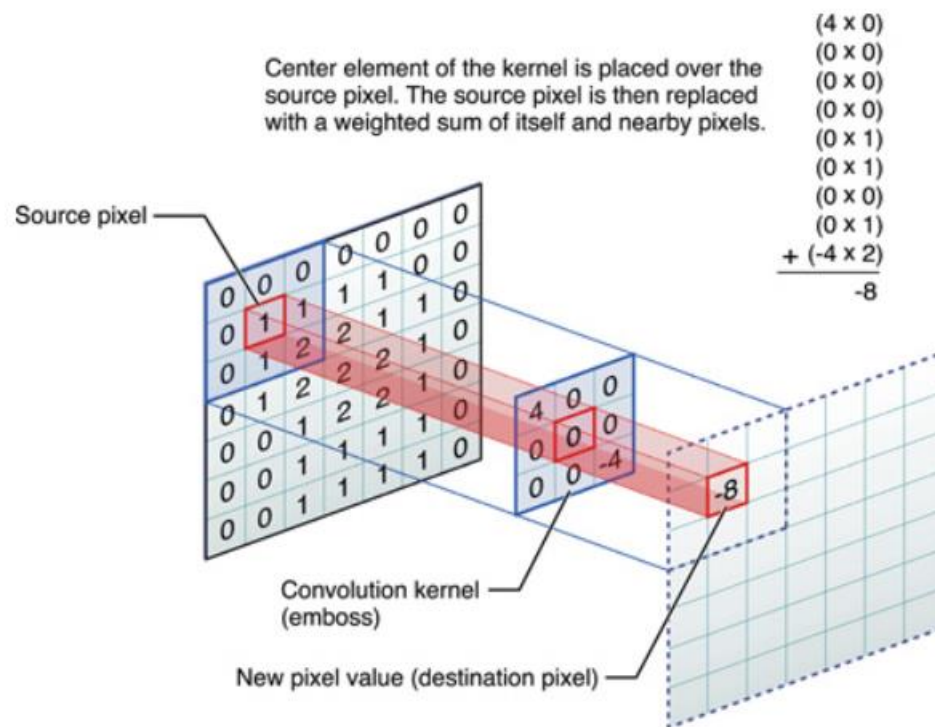


Figura 2.10: ejemplo de la convolución de una imagen. Esta operación da origen a una nueva matriz con diferentes valores. Fuente: [24].

A través de la figura 2.10, se puede observar una primera convolución. Luego el núcleo con los mismos valores se desplaza hasta abarcar toda la imagen y producir nuevos valores. A la matriz con estos nuevos valores se le conoce como *mapa de características* o *mapa de activación*.

La siguiente capa en una CNN es la capa de agrupación. En esta capa se reemplaza la salida de la capa convolucional, por una estadística resumida de las salidas cercanas. Esto se hace con el objetivo de reducir el tamaño de la representación, así como la capacidad de cálculo requerida. La capa de agrupación más popular es la agrupación máxima (*max pooling*), que informa la salida máxima de las porciones del mapa de características. La figura 2.11, muestra el funcionamiento de la agrupación máxima.

En la figura 2.11, se presentan los elementos principales de la capa de agrupación: el tamaño del filtro, que indica el tamaño de la porción del mapa de características a abarcar; y el paso, que indica la cantidad de elementos que se moverá para hacer una nueva agrupación. La salida de esta capa genera un mapa de características agrupado.

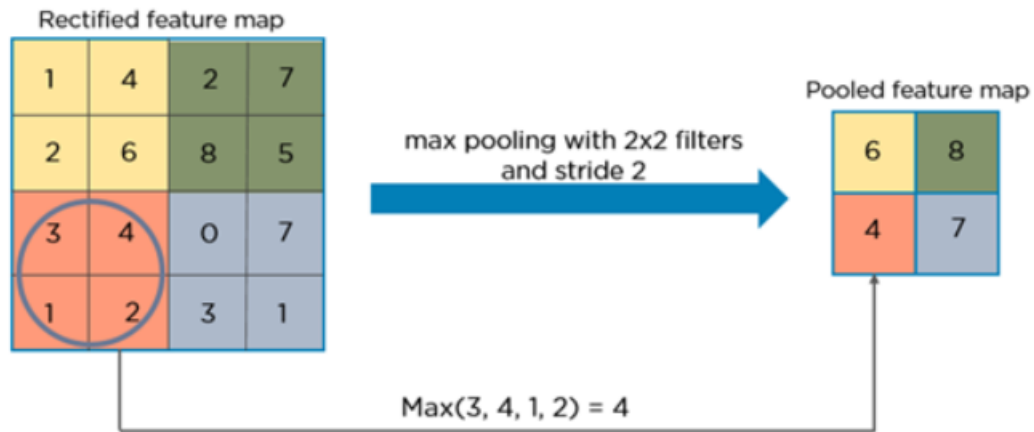


Figura 2.11: ejemplo de agrupación máxima con filtros de 2 x 2 y paso de 2. Fuente: [25].

Finalmente, se tiene una capa densa, en donde las neuronas de dicha capa se encuentran conectadas totalmente con la salida de la capa de agrupación y con las neuronas de la capa siguiente.

Una vez explicados los componentes principales de la red convolucional, es importante detallar cómo funciona en el contexto de aprendizaje profundo.

En la práctica, las redes convolucionales contienen más de una capa de cada tipo, de esta forma, se generan múltiples mapas de características a medida que se agregan capas de convolución y agrupación. Así las redes convolucionales constan de dos procesos: la extracción de características y la clasificación. El primero corresponde a las operaciones de convolución y agrupamiento, en las cuales, como se mencionó anteriormente, la red aprende patrones más básicos de la imagen. Luego, la clasificación está a cargo de las capas densas que aportan la no linealidad al modelo, y la red aprende formas más complejas y específicas de las imágenes, como objetos o rostros de animales o personas. La figura 2.12, muestra un ejemplo de lo anterior en el contexto de la clasificación de números.

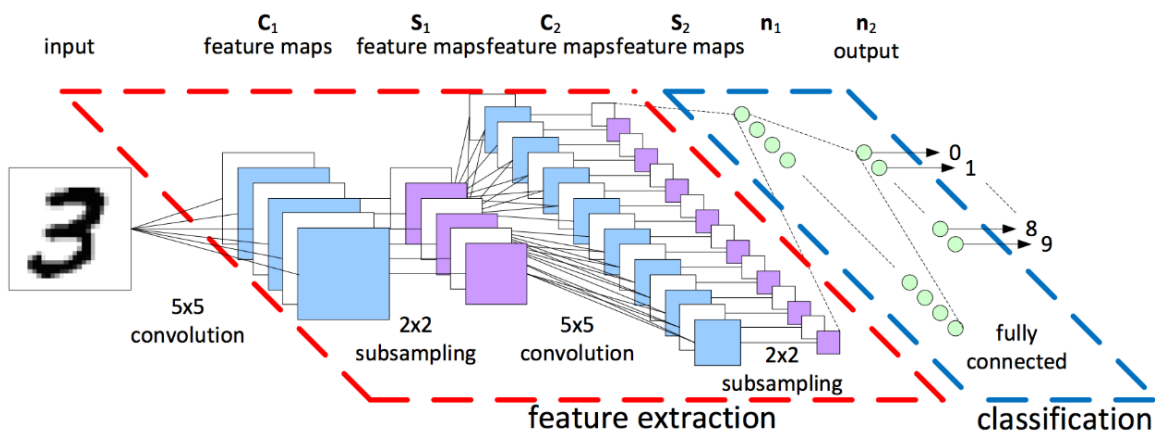


Figura 2.12: ejemplo de arquitectura de red neuronal convolucional para la clasificación de números. Fuente: [26].

Para el trabajo de memoria, se usará la extracción de características de una CNN pre entrenada para las imágenes de las casas robadas y no robadas. A continuación, se presenta la red elegida y se explica su contribución en el trabajo.

2.3.2.1 VGG-16

En la actualidad, existen muchas redes neuronales convolucionales con una arquitectura definida y que han sido utilizadas en múltiples investigaciones. El motivo de esto es que el proceso de diseño y construcción de una red es difícil y toma tiempo, además, las redes más populares han sido testeadas en conjuntos enormes de imágenes y han demostrado muy buenos resultados, por lo que es lógico pensar que se usen para trabajar tomando como base sus arquitecturas.

VGG-16 es una red neuronal convolucional de 16 capas de profundidad que fue presentada por K. Simonyan y A. Zisserman de la Universidad de Oxford en el contexto de la competencia de clasificación de imágenes ILSVRC-2014. El modelo alcanza un 92,7% de precisión de prueba top 5 en un subconjunto de Imagenet, es decir, la etiqueta de destino es una de las 5 predicciones principales del modelo para el 92,7% de las imágenes [27]. Además, fue entrenada durante semanas utilizando una GPU NVIDIA Titan Black debido al alto costo computacional [28]. Su arquitectura se muestra en la figura 2.13:

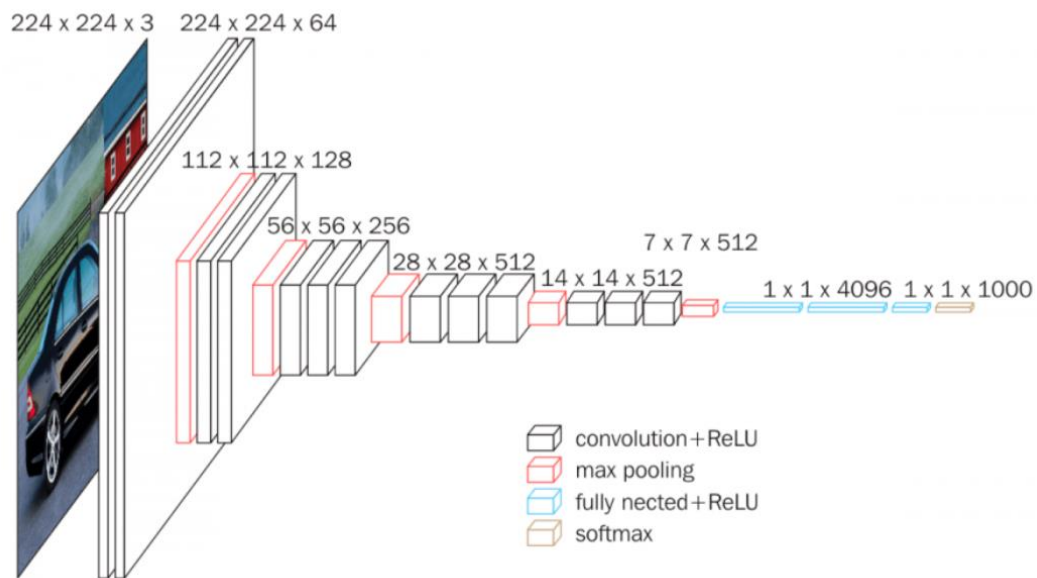


Figura 2.13: arquitectura de VGG-16 para la clasificación de imágenes. Fuente: [28].

Imagenet es un conjunto de datos, que cuenta con más de 15 millones de imágenes etiquetadas de alta resolución pertenecientes a 22.000 categorías. Por otro lado, ILSVRC (Desafío de reconocimiento visual a gran escala de Imagenet) es una competencia anual que se lleva a cabo desde el año 2010. En esta competencia se utiliza un subconjunto de Imagenet que consta de aproximadamente 1.2 millones de imágenes de entrenamiento, 50 mil imágenes de validación y 150 mil imágenes de prueba.

En [29] hacen uso de Alexnet para la extracción de características de las imágenes de los lugares de robo y no robo, en un problema similar al presentado en esta memoria.

En este caso, se utiliza VGG-16 puesto que muestra mejores resultados que Alexnet, es más actual (2014 vs 2012), y a través de la librería Keras es de rápida implementación, puesto que cuenta con el modelo pre entrenado, no así el de Alexnet.

El uso de una red pre entrenada en este trabajo se justifica, en primer lugar, por el tiempo de implementación, el cual es notablemente menor debido a que se utilizan los pesos aprendidos a través del conjunto en el cual fue entrenada la red (Imagenet en este caso), por tanto, la fase de entrenamiento es omitida para las imágenes. En segundo lugar, es de fácil implementación por medio de la librería Keras como se mencionó anteriormente, y, por último, el conjunto sobre el cual fue entrenada la red cuenta con imágenes de casas, autos, calles, rejas, árboles y otros objetos que están presentes en las imágenes de las casas que se utilizan en este trabajo, por lo tanto, la extracción de características con los pesos aprendidos en Imagenet son útiles en este caso (y probablemente mejor a la que se pueda obtener entrenando la red nuevamente con las imágenes de la memoria) [30].

2.3.3 Red neuronal recurrente (RNN)

Las redes neuronales comunes no pueden resolver problemas secuenciales, es decir, donde se deba utilizar información pasada en orden para obtener una predicción. Por ejemplo, las palabras sueltas en un ensayo no significan nada si no se consideran las palabras anteriores. Dada esta deficiencia, se creó una arquitectura de red neuronal para enfrentarla: las redes neuronales recurrentes [31].

Estas redes incorporan bucles en su estructura que permiten que la información persista. En la figura 2.14 se muestra a grandes rasgos el funcionamiento de esta red.

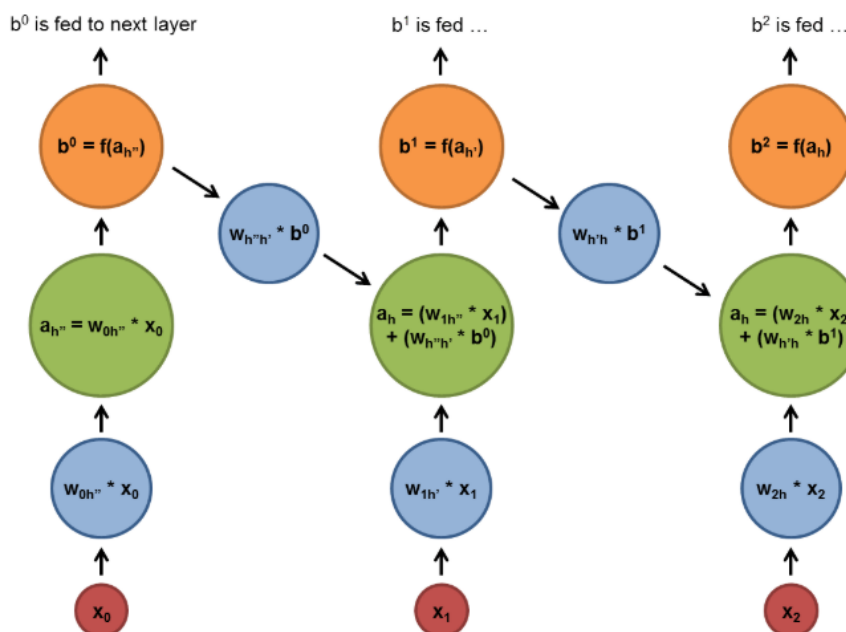


Figura 2.14: despliegue y funcionamiento de la estructura general de una RNN. Fuente: [32].

A través de la figura 2.14, se puede observar que las RNN están estrechamente relacionadas con secuencias y listas. Así, una RNN se puede considerar como copias múltiples de una misma red y en cada una se transmite un mensaje para la red que sigue.

En la figura, se ve que la salida b^t es multiplicada por un peso W y luego se incluye para calcular el siguiente estado a_h . De esta forma, se incluye la información de los periodos anteriores.

A pesar de que se incluye información del pasado, existe un problema con las RNN normales, y es que a medida que se avanza en los periodos de tiempo, se va perdiendo la información más antigua. Esto se conoce como *gradiente de fuga* y representa un obstáculo importante para el rendimiento de la RNN [32]. Para enfrentar esta situación se creó la red LSTM (memoria de corto y largo plazo), la cual se utiliza en esta memoria, y se explica a continuación.

2.3.3.1 LSTM

La LSTM resuelve el problema del gradiente de fuga dado que ayuda a preservar el error que puede propagarse a través del tiempo y de las capas. Al mantener este error constante, las RNN pueden continuar aprendiendo durante muchos periodos de tiempo [32].

Al ser entrenadas, estas redes aprenden qué deben desechar (olvidar) y qué deben mantener (recordar), con el objetivo de lograr un mejor rendimiento según el problema que se busca resolver.

El cambio de estas redes con las RNN normales guarda relación con las operaciones dentro de las celdas que representan los estados ocultos a_h . Sin embargo, lo más importante es entender por qué se utiliza la LSTM en la memoria, por lo que no se ahonda mayormente en el detalle de su estructura.

Esta RNN es utilizada para los datos temporales que se incluyen en la memoria, a saber, la información de la temperatura, nivel de precipitación y cantidad de robos. Estos datos son entregados a la red como series temporales.

2.3.4 Funciones de activación: ReLU y Sigmoid

La función de activación se utiliza para obtener la salida de cada nodo en una red neuronal. La función de activación se selecciona de acuerdo con los valores que se quieran conseguir.

La función ReLU (rectificador lineal) es de la más populares en la actualidad para algoritmos de aprendizaje profundo y entrega como resultado el máximo entre 0 y la entrada a la neurona, es decir, $\text{ReLU}(x) = \max(0, x)$. La razón principal de su masivo uso es la forma de esta, ya que tiene buenas propiedades, como función y derivada monótona, y los valores negativos se transforman de inmediato en 0, por lo que no posee cálculos en zonas complejas donde se presenten pendientes cercanas a 0 y el gradiente pueda desvanecerse (a diferencia de sigmoid o tanh). De esta forma, su costo computacional es muy bajo y se llega rápidamente a buenos resultados [33].

La función sigmoid, es utilizada especialmente para problemas en donde se busca obtener una probabilidad como salida, puesto que entrega valores en un rango de 0 a 1. La función es diferenciable y monótona, por lo que también presenta propiedades adecuadas para calcular el gradiente, sin embargo, puede hacer que el modelo se atasque en el entrenamiento, debido al desvanecimiento del gradiente [34].

En la figura 2.15 se observa el comportamiento de las funciones sigmoid y ReLU.

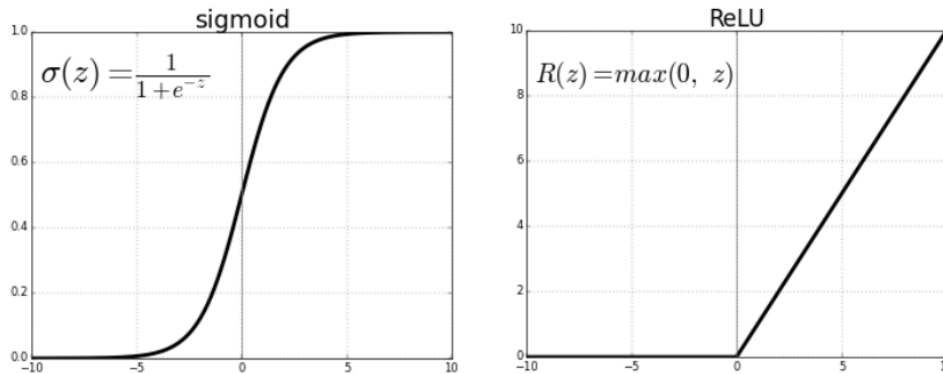


Figura 2.15: izquierda: función sigmoid. Derecha: función ReLU. Fuente: [35].

En este trabajo se utiliza la función de activación sigmoid para la salida de la capa final, dado que se busca la probabilidad de robo, por lo que el resultado tiene directa interpretación. Por otro lado, en las capas ocultas de la red neuronal se utiliza ReLU luego de pruebas con distintas funciones de activación. ReLU entrega mejores resultados con un tiempo de entrenamiento menor que otras funciones.

2.4 Métricas de evaluación

Para evaluar los resultados obtenidos por medio de modelos de aprendizaje automático y profundo, se utilizan distintas métricas que dan cuenta de la precisión de la clasificación. Se introduce en primer lugar, la matriz de confusión y luego 5 métricas que son comúnmente utilizadas y que se utilizan también en esta memoria.

2.4.1 Matriz de confusión

La matriz de confusión es una medida de rendimiento de un modelo de clasificación de aprendizaje automático. Esto implica que puede ser utilizada para problemas de clasificación de una o más clases. El problema de este trabajo corresponde a un problema de clasificación de 2 clases, por lo que se explicará desde este punto de vista. En la figura 2.16, se observa la matriz de confusión para dos clases (0 o 1).

En el eje y se tienen los valores predichos por el modelo, y en el eje x los valores reales de los datos. Se tienen 4 cuadrantes que se explican en el contexto de la predicción de robo a casas:

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figura 2.16: matriz de confusión para 2 clases. Fuente: [36].

- Verdadero positivo (TP): el modelo predice *robo* (1) y efectivamente la casa había sido robada (1).
- Falso positivo (FP): error tipo I. El modelo predice *robo* (1) y la casa no había sido robada (0).
- Falso negativo (FN): error tipo II. El modelo predice *no robo* (0) y la casa sí fue robada (1).
- Verdadero negativo (TN): el modelo predice *no robo* (0) y efectivamente la casa no había sido robada (0).

La suma total de todos los cuadrantes es igual al total de datos que se utilizan para evaluar el modelo (conjunto de prueba por lo general). El ideal es tener todos los datos entre verdaderos positivos y verdaderos negativos.

2.4.2 Métricas utilizadas

A través de los valores obtenidos en la matriz de confusión se definen las siguientes métricas:

- Accuracy: totalidad de aciertos sobre la totalidad de datos clasificados.

$$\frac{TP + TN}{TP + FP + FN + TN}$$

- Recall: totalidad de clases positivas predichas correctamente sobre la totalidad de datos positivos. Mientras más alto, significa que mayor es la cantidad de robos que se están clasificando correctamente.

$$\frac{TP}{TP + FN}$$

- Precision: totalidad de clases positivas predichas correctamente sobre la totalidad de datos predichos como positivos. Mientras más alto, significa que mayor es la cantidad de no robos que se están clasificando correctamente.

$$\frac{TP}{TP + FP}$$

- F1-Score: media armónica entre recall y precision. Es decir, captura ambas medidas al mismo tiempo. Es útil para comparar modelos donde uno posea alto recall y bajo precision, y el otro tenga bajo recall y alto precision.

$$\frac{2 * Recall * Precision}{Recall + Precision}$$

Todas las métricas varían entre 0 y 1, por lo que mientras más cercanas a 1, mejor será el modelo con respecto a esa métrica.

Por último, se define la función de pérdida: entropía cruzada binaria. Esta función se utiliza para ajustar los hiperparámetros del bosque aleatorio y la red neuronal. Es utilizada para problemas que involucran dos clases, sí y no (como en el caso de este trabajo), y mide qué tan lejos del valor verdadero se encuentra cada predicción. Luego, se promedia para todas las predicciones y se obtiene una pérdida final. De esta manera, se busca tener una pérdida lo más baja posible. En la figura 2.17, se observa la ecuación que define esta función.

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=0}^N (y * \log(\hat{y}_i) + (1 - y) * \log(1 - \hat{y}_i))$$

Figura 2.17: función de pérdida entropía cruzada binaria. Fuente: [37].

2.5 Trabajos relacionados

2.5.1 Inicios en prevención del delito

La batalla contra el crimen en occidente se inicia desde el sistema judicial criminal. El gobierno de Estados Unidos entre los años 1950 y 1980 aumentó el presupuesto y la cantidad de agencias de justicia penal como respuesta al crimen [38], sin embargo, no hubo una disminución significativa en la tasa criminal. A pesar de los debates en cuanto a la efectividad de las estrategias abordadas para la reducción del crimen, existe el consenso de que el sistema judicial en occidente está limitado en su capacidad para prevenir el crimen [39]. Debido a esto, se introdujo como alternativa expandir el rol normal de un ciudadano en la lucha contra el crimen, bajo la premisa de que estos deberían tener una mayor motivación en aceptar responsabilidad en la prevención del delito que el Estado, dado que, al ser una sociedad libre, los ciudadanos tienen un papel preponderante en la mantención del orden [40]. Este concepto se acuñó bajo el nombre de *prevención comunitaria del delito*, y se llevaron a cabo variados estudios e investigaciones respecto al tema [41], [42], [43], [44]. Así, diversas prácticas fueron incentivadas, relacionadas a comportamiento individual (como auto protección o comportamientos menos riesgosos), y colectivo (como patrullaje ciudadano, vigilancia de parte de vecinos y cohesión de la comunidad). A pesar de esto, era claro que sólo el actuar de la comunidad no era suficiente, además de tener consecuencias negativas para los individuos, como la pérdida parcial de su libertad y la continua sensación de miedo, dado que debían ellos mismos velar por su seguridad [39]. Por consiguiente, en los últimos años se han originado nuevos enfoques con el objetivo de fortalecer el actuar policial, de manera de lograr una asignación de

recursos más eficiente y así evitar hechos criminales antes de que ocurran. Esto ayuda también a la ciudadanía a no asumir la total responsabilidad por su seguridad.

2.5.2 Prevención del crimen a través del análisis de datos

Los programas de prevención comunitaria del delito tenían ciertas dificultades relacionadas a la calidad de vida de los ciudadanos, como se mencionó anteriormente, pero parecía una buena alternativa ante la capacidad limitada de la fuerza policial. No obstante, evaluaciones posteriores resaltaban la existencia de grandes diferencias entre las distintas comunidades y vecindarios en una ciudad, por lo que los programas rara vez eran viables o escalables más que a sectores puntuales [45]. De esta forma, en Estados Unidos, de la mano con el aumento del presupuesto a las agencias de inteligencia luego de los ataques terroristas en 2001, las autoridades locales comenzaron a monitorear más de cerca la actividad criminal, a través del análisis de las cantidades cada vez más grandes de registros de delitos [46]. Esto propició diversos estudios con respecto a la prevención del delito a través de técnicas de minería de datos y modelos estadísticos, bajo la creencia de que las conductas criminales siguen ciertos patrones que pueden ser identificables, tales como la alta concentración de eventos criminales en ciertos sectores apodados “*hot spots*” y a ciertos tipos de víctimas reiteradas, o el flujo de delitos según hora, día de la semana y estación del año [47]. Según [48], basado en la información criminal histórica y el registro de las relaciones criminales, un equipo de investigación criminal puede extraer información útil para detectar los hechos relacionados a la ocurrencia de un crimen y así evitar futuros eventos. De esta forma, a través del análisis de datos criminales inteligente se pueden descubrir patrones en la conducta criminal y así entender dónde, cuándo y por qué de los eventos criminales [49].

2.5.3 Estudios y aplicaciones en la investigación criminal

Brahan, JW, Lam, KP, Chan, H. y Leung, W. (1998) [50] describen un Sistema de Soporte de Decisiones Inteligente implementado para la policía de Hong Kong, el cual utiliza técnicas de aprendizaje automático y redes neuronales. Este sistema fue integrado de manera que tomaba como entrada información de distintas bases de datos y entregaba como salida las características del sospechoso, incidentes relacionados, eventos sospechosos cercanos, lista de posibles sospechosos, resumen de la actividad y una composición facial del victimario. Así, se entregan las primeras evidencias de que las técnicas relacionadas al análisis de datos son de gran utilidad y apoyo para el trabajo del investigador criminal.

Otros estudios también mencionan el uso de Sistemas de Soporte de Decisiones Inteligentes para complementar el trabajo de los analistas al momento de buscar patrones y similitudes en los eventos criminales. En [51] se describe un enfoque para la asociación automática de datos que disminuye por un factor de 0.33 la búsqueda en bases SQL, entregando además una mejor precisión que el sistema anterior. En [52] se describe un sistema que utiliza redes neuronales para el descubrimiento de características de patrones de eventos criminales, lo que permite un mejor uso de la fuerza policial, evitando el desperdicio de tiempo y recursos.

La evolución en el uso de los datos para un análisis criminal más eficiente dio paso a la implementación de técnicas de minería de datos en otros procesos de la investigación

criminal. En [53] hacen uso de estas técnicas para evaluar el desempeño de los investigadores de escenas del crimen (CSI) clasificándolos en 4 grupos a través del método k-means de acuerdo con la recolección de ADN y huellas dactilares. Por otro lado, en [54] utilizan Natural Language Processing (NLP) para extraer información relevante de las entrevistas que los investigadores hacen a testigos, víctimas, o a los mismos criminales. De esta manera, se acelera el proceso vinculado al análisis de estos reportes. Se reporta un accuracy entre el 94 y 96 por ciento y un recall entre el 85 y el 90 por ciento para esta última investigación.

2.5.4 Estudios de modelos predictivos de crimen

Con el aumento en la cantidad de datos criminales y la popularización del concepto de “*Big Data*”, han ido surgiendo, de manera continua, nuevos enfoques para la extracción de la información en los registros criminales, con el objetivo de un mejor entendimiento del actuar criminal y prever, en última instancia, futuros eventos [55]. De esta forma, la atención se ha trasladado desde el análisis de *hot spots* [56], [57], como el análisis de patrones de puntos; técnicas de minería de datos, como minería de texto y análisis espacial [58], hacia el desarrollo de modelos predictivos con énfasis en prevención del delito [59].

Para la construcción de los modelos mencionados, se han usado diversas técnicas y factores provenientes no solo del registro criminal, sino que, han complementado esta información con atributos espaciales provenientes de otras fuentes de datos. Ingilevich et al. (2018) [60] estudia distintas herramientas para lograr una aproximación de eventos criminales. En su estudio hace uso de modelos de *regresión lineal*, *regresión logística* y *aumento de gradiente*, utilizando factores espaciales como entrada. Esto bajo la premisa que los crímenes no ocurren de manera aleatoria, sino que dependen de dichos factores. Se incluye el producto interno bruto, la densidad poblacional, el desempleo y la población sin vivienda, además de otros patrones criminales espaciales como atributos relevantes en la predicción del delito. Luego, a través de la técnica de *clustering* (agrupamiento) se examinan sub-atributos de cada tipo de delito. Estos incluyen los números de población, estaciones de policía, escuelas, bares, y otros puntos de interés. Se concluye que la selección de características es la solución para crear modelos predictivos robustos y el aumento de gradiente es el modelo que entrega mejores resultados en la predicción de eventos criminales. Malathi y Baboo (2011) [61] utilizan técnicas de agrupamiento (*DBScan clustering*, *k-means clustering*) y técnicas de clasificación (árboles de decisión) para predecir tendencias de crimen para el año siguiente, utilizando 8 años de registros criminales. Nguyen, Sung y Hatua (2017) [62] construyen distintos clasificadores por medio de algoritmos de aprendizaje automático, incluyendo *support vector machines*, *bosque aleatorio*, *aumento de gradiente* y *redes neuronales*, que permiten predecir distintos tipos de delitos. A la información de los registros se agrega información demográfica del área, antecedentes educacionales, económicos y étnicos de las personas involucradas. Los datos son obtenidos de la policía de Portland, y se encuentran desbalanceados en cuanto a las distintas clases, por lo que se utilizan técnicas de muestreo para lidiar con este problema. El rendimiento de los modelos varía según las técnicas utilizadas, por lo que se concluye acerca de los mejores modelos a utilizar dependiendo de qué técnica de muestreo se utilice.

Además de los factores espaciales y demográficos, se han sido incluido factores temporales y contextuales en otras investigaciones. Cadena (2016) [63] estima un modelo

logit multinomial con el objetivo de determinar las comunas más expuestas a ser víctimas de determinados delitos a través de la medición de la incidencia de factores espaciales, temporales y elementos de contexto que caracterizan a las distintas comunas de la Región Metropolitana de Chile. De esta forma, para cada tipo de delito de mayor connotación social, se tiene un listado con las comunas más y menos propensas a que se presente cada tipo de delito. Cortez, Carrera, Kim y Jung (2018) [64] proponen una nueva arquitectura para la predicción de Eventos de Emergencia usando datos de la policía de Guatemala. Se propone una arquitectura basada en Redes Neuronales Recurrentes (RNN), específicamente una arquitectura LSTM (long short-term memory). Los resultados son que estas redes tienen un mejor rendimiento en comparación a análisis tradicionales de series de tiempo y a algunos métodos de aprendizaje automático.

Por otro lado, Kang, H-W. y Kang, H-B. [29] utilizan una “Red Neuronal Profunda” (DNN por sus siglas en inglés) de fusión para construir un predictor de ocurrencia de crímenes, basado en 3 tipos de fuentes de datos: espacial, temporal y contextual. La información contextual es obtenida por medio de imágenes provenientes de Google Street View de los lugares en donde ocurrieron delitos. Este modelo se hace cargo de los problemas más recurrentes en otros predictores, como las relaciones no lineales entre los datos, redundancia de variables y dependencia entre distintas fuentes de datos. Los resultados muestran un mejor rendimiento que otros modelos predictivos, alcanzando un accuracy del 84%, precisión de 74%, recall de 80% y AUC de 0.83. La figura 2.18 muestra la arquitectura de red neuronal utilizada en este artículo.

En este trabajo se utilizará una estructura similar a la expuesta en la figura 2.18 para la predicción de robo a casas, la cual es presentada en la sección 3.1.

Así como los modelos presentados en [63] y [29], otros estudios comenzaron a agregar características de contexto [65], [66], [67]. Estas características hacen referencia a elementos presentes en los lugares donde ocurren los robos. Esto se basa principalmente en la *Teoría de las ventanas rotas* [68], la cual señala que entornos urbanos que presentan desorden o señales de vandalismo son más propensos a ser víctimas de actos criminales en el corto plazo. Es por esto, que el trabajo de memoria también incluye elementos de contexto a través de imágenes de Google Street View, las cuales corresponden a las fotos de las casas tanto robadas como no robadas.

2.5.5 Estudios de modelos predictivos de robos a viviendas

El robo a viviendas es uno de los tipos de delito con mayor volumen en países occidentales, por lo que se requieren recursos significativos por parte de la fuerza policial para abordarlos. De esta manera, diversos estudios han tenido como objetivo principal la predicción de este tipo de robos. Por ejemplo, Miethe y McDowall (1993) [69] desarrollan un modelo que incorpora datos contextuales en los modelos de victimización criminal en cuanto a robos a viviendas en la ciudad de Seattle. Sus hallazgos se resumen en que mientras menor sea el nivel de seguridad de la zona y mientras mayor sea el atractivo de las residencias, hay un mayor riesgo de robo cuando las residencias pertenecen a un sector acomodado de la ciudad, en cambio, estos efectos no son significativos en áreas socialmente desorganizadas.

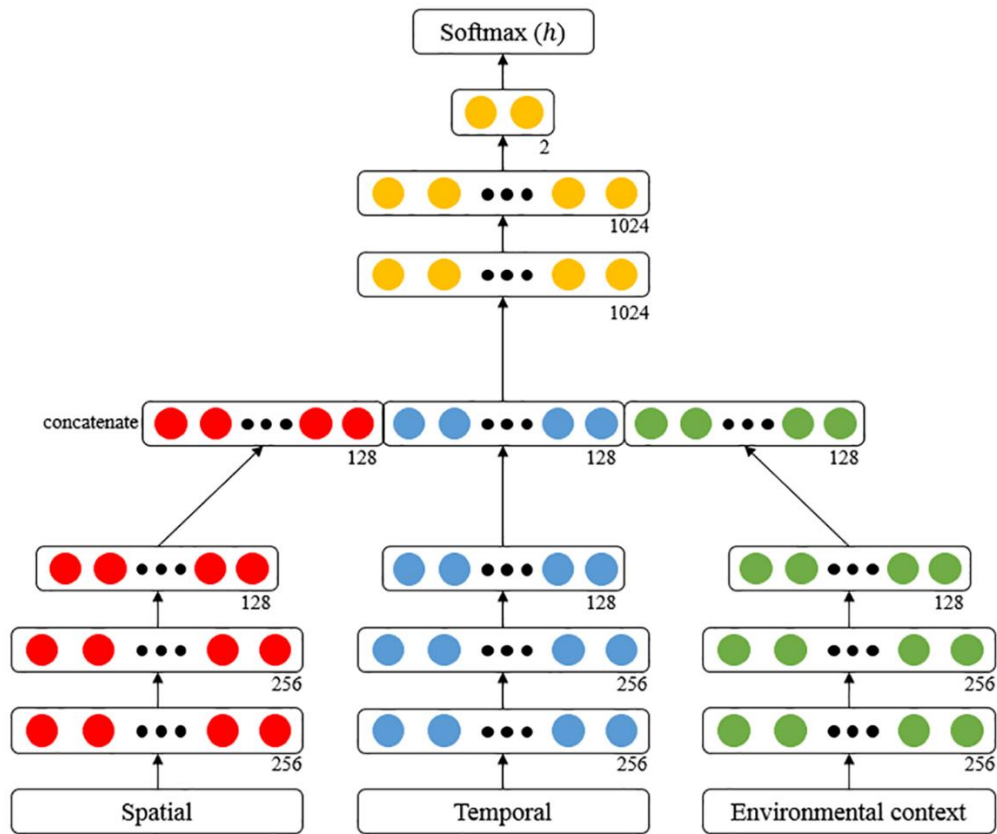


Figura 2.18: Estructura de Red Neuronal Profunda utilizada en [29]. Fuente [29].

Oatley y Ewart (2003) [70] se enfocan en la predicción de la probabilidad de robos para una cierta propiedad víctima de reiterados robos en el condado metropolitano de West Midlands en Inglaterra. Para esto, utilizan *redes bayesianas* incluyendo factores tales como, características del victimario, modus operandi, bienes robados e indicadores numéricos de la distribución del crimen en un área. El foco principal, era el desarrollo de un software que permitiera la predicción del robo a viviendas. Holm y Plynning (2018) [71] desarrollaron un modelo predictivo espacio temporal para robos a viviendas usando redes neuronales convolucionales en un área de estudio en Estocolmo, Suecia. Sus resultados son que el 43% de todos los robos son correctamente predichos. En [72] se desarrolla un método predictivo para obtener la probabilidad de robos residenciales a través de simulaciones. En él, consideran factores ambientales que según sus estudios influyen en el actuar de los delincuentes a la hora de cometer este tipo de atracos.

Otros estudios indagan en la elección de viviendas por parte de los criminales y qué factores influyen en esta elección. En [5] se estudian los factores de nivel de heterogeneidad étnica, viviendas unifamiliares, número potencial de objetivos en la zona de robo, proximidad entre el lugar de robo y la vivienda del ladrón, movilidad o rotación residencial, riqueza esperada por el ladrón y proximidad de la zona de robo al centro de la ciudad, a través de un modelo *logit*. Sus resultados indican que sólo los atributos de viviendas unifamiliares, heterogeneidad étnica, proximidad al centro de la ciudad y número potencial de objetivos en la zona de robo influyen en el robo a viviendas. Otros estudios coinciden en que los perpetrados de robos a viviendas basan su elección en la accesibilidad a la propiedad, la expectativa en cuanto a la riqueza del botín, la facilidad de escape y el nivel de vigilancia de la casa [73], [6] y [4].

Otros factores que se han estudiado son las condiciones climáticas. En [74] se investigó el efecto de la temperatura y de los niveles de precipitación en la ocurrencia de homicidios, violaciones y robos a propiedades. A través de un modelo que incluye controles económicos y efectos fijos según el condado, se encuentra que los niveles más altos de lluvia y nieve están asociados con una menor cantidad de robos a propiedades. La razón de esto es que, en los días con condiciones climáticas adversas, las personas tienden a quedarse en sus casas, mientras que en los días calurosos salen, momento en el cual los delincuentes aprovechan de entrar y robar.

Dado lo expuesto anteriormente en esta sección y en las anteriores, se puede dimensionar la influencia que ha tenido el análisis de datos y los modelos de aprendizaje automático en el área de criminología, y la evolución que ha tenido esta última a través de las mejoras tecnológicas y el registro de la información. El alcance ha sido tal, que la implementación de modelos predictivos o sistemas inteligentes que utilizan técnicas de aprendizaje automático ha llegado a distintas partes del globo como Chile [2], Reino Unido [4], China [75], y muchos otros.

Con respecto al robo a viviendas, los estudios señalados enfatizan el comportamiento racional de los criminales asociados a este tipo de robo, quienes maximizan su utilidad esperada al robar, considerando factores físicos, temporales y sociales. Así, minimizan su probabilidad de ser atrapados. De esta forma, se hace relevante el presente trabajo, ya que al ser un tipo de delito menos aleatorio que otros, se espera encontrar patrones que permitan predecir la probabilidad de robo a casas en Dallas.

3. Modelo y Recolección de Datos

A continuación, se detalla la estructura del modelo, seguido del procedimiento para la obtención de los datos y la confección final de cada una de las tres fuentes, las cuales son: espacial, temporal y contextual.

3.1 Modelo

En la sección anterior se expuso sobre diversas tecnologías y aplicaciones en el campo del análisis criminal. En variados estudios se prueba la influencia de factores demográficos y socioeconómicos en la predicción de robos. Por otro lado, en [74] se demuestra la influencia de los factores climáticos en cuanto al nivel de robo, donde a medida que aumenta la temperatura promedio en un sector, aumenta también la probabilidad de robo en este. Para el robo a casas, en particular, esto es explicado, dado que, a mayor temperatura, es más probable que las personas salgan de sus hogares, momento en el cual los que cometen este tipo de delito ingresan a la vivienda. Además, según estudios sobre el fenómeno de la revictimización [76], [77] existe una tendencia respecto a que zonas que han sido robadas en cierto periodo de tiempo vuelvan a ser víctimas de robo en el corto plazo, por lo cual la cantidad de robos cada zona en días anteriores es un factor que influye en la predicción de robo. Por último, según señala la teoría de “*Ventanas Rotas*” [68], el nivel de desorden físico en un sector, como suciedad en las calles, rayas en las murallas, u otro tipo de señales de vandalismo, se relacionan con conductas reprochables socialmente, las cuales pueden desencadenar en actos criminales, por lo cual es importante la inclusión de las imágenes de las casas robadas y no robadas como input del modelo.

De esta forma, se desarrolla un modelo multi modal, similar al expuesto en [29], específicamente en la figura 2.18, es decir, con entradas separadas para los tres tipos de fuentes: espacial, temporal y contextual, y con una clasificación final, que en el caso de la investigación mencionada es multiclase (más de dos), y en el caso de esta memoria es clasificación binaria: robo o no robo.

Para la correcta aplicación de este modelo se debe construir, en primer lugar, una base de datos de casas que no hayan sido robadas, de manera que el modelo logre un aprendizaje tanto de robos como de no robos, y así obtener la clasificación deseada. Además, se deben recolectar los datos necesarios para cada entrada y para ambas bases de datos. En el caso de la información espacial, estos son los datos demográficos y socioeconómicos que caracterizan a cada punto. Adicionalmente, para esta entrada, se agrega información de puntos de interés cercanos a cada lugar como aeropuertos, restaurantes, bencineras, malls, centros de belleza, estaciones de policía, entre otros. Esto, debido a lo presentado en [60], en el cual se realiza una clasificación con respecto a distintos puntos de interés para la predicción de robo, por lo que se cree que estos podrían influir en la predicción en este caso. Con respecto a la información temporal, en [74] se incluyen algunas condiciones climáticas. En esta memoria, se trabaja con la temperatura, el nivel de precipitación y la cantidad de robos en semanas anteriores. Por último, la información contextual se obtiene a partir de imágenes en Google Street View como se hizo en [29].

La arquitectura mostrada en la figura 2.18 está formada por tres ramas independientes, que representan los tres tipos de fuentes de datos. En cada una, se tienen 3 capas densas con 256 neuronas las dos primeras y 128 la última. Luego, las fuentes se fusionan en una capa de concatenación (128 x 3 neuronas), pasan por dos capas densas de 1024 neuronas cada una y finalmente por una capa densa de 2 neuronas con función de activación softmax, que da cuenta de las probabilidades de un acto delictivo. Dicha arquitectura fue utilizada para ese problema en particular y se ajusta a los datos de entrada, por lo que, en el presente trabajo, se desarrolla una arquitectura que se adecúe a los datos y al problema expuesto en esta memoria.

En la figura 3.1 se muestra la estructura de red neuronal utilizada para el presente trabajo.

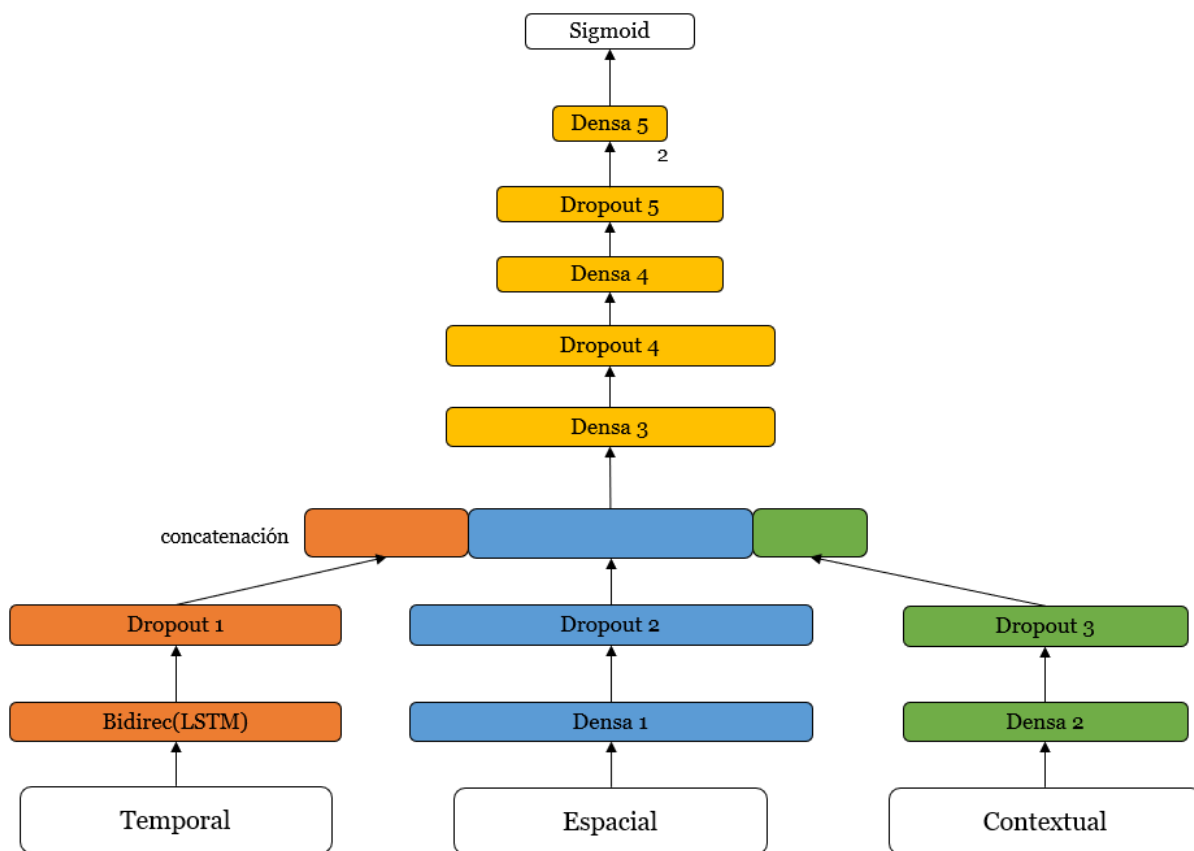


Figura 3.1: arquitectura del modelo propuesto para la predicción del robo a casas en la ciudad de Dallas. Arquitectura basada en figura 2.18. Fuente: autor.

Una de las principales diferencias del modelo expuesto en la figura 3.1 con respecto al modelo ocupado en [29], es que, en el caso de este trabajo, se utiliza una LSTM (presentada en la sección 2.3.4.1) para la base temporal antes de la concatenación de las distintas fuentes de información. Esto, dado que se busca extraer patrones en el robo de casas a través de información pasada y así hacer una predicción futura, por lo que se necesita una estructura que otorgue persistencia a la información. Además, se efectúa un ajuste de los hiperparámetros de la red neuronal, lo cual engloba una elección de la cantidad de capas y neuronas por cada capa de la red, técnicas de regularización, selección de las funciones de activación y elección del optimizador e inicializadores. Por último, la

función de activación utilizada en la capa final para obtener las probabilidades es la función “*sigmoid*”, la cual se utiliza para los problemas de clasificación binaria.

La configuración de hiperparámetros de la red neuronal se detalla en la sección 5.1, donde la cantidad de neuronas por capa varía de acuerdo con la cantidad de características a utilizar, es por esta razón por lo que en la figura 3.1 no se observa la cantidad de neuronas en cada capa (exceptuando por la capa densa final que es fija igual a 2). Aun así, la proporción de neuronas se mantiene para todas las características, la cual se observa a través de los tamaños de los rectángulos para las distintas fuentes de datos.

A continuación, se presenta la metodología respecto a la construcción de las distintas fuentes de datos.

3.2 Base de datos de robos

Esta memoria se realiza con la información presente en la base de datos pública de casos policiales de la ciudad de Dallas, que se puede obtener a través del siguiente link: <https://www.dallasopendata.com/Public-Safety/Police-Incidents/qv6i-rriz>. Esta base contiene registros de denuncias policiales de diversa índole desde junio de 2014 a diciembre de 2018. Cuenta con más de 500 mil registros (filas) de crímenes y cada uno es caracterizado con 100 variables. Sin embargo, la mayor parte de estas corresponden a formalidades a la hora de registrar la denuncia, y no aportan información relevante para la confección del trabajo. A pesar de esto, la base cuenta con la fecha del incidente (formato año-mes-día), la dirección del suceso y la localización en formato latitud y longitud para la mayoría de los casos. La fecha es necesaria para la construcción de la base temporal, mientras que los atributos de localización son necesarios para la obtención de las imágenes de los lugares de robo, para la construcción de la base espacial, y para la confección de la base de datos de lugares de no robo.

Se utilizan datos en el periodo de junio de 2014 a diciembre de 2018. Además, los delitos que interesan para este trabajo son los delitos de allanamiento (burglary), los cuales corresponden a 50.142 registros como se muestra en la figura 1.1. Además, se busca enfocar el trabajo de memoria al robo a casas, por lo que, en primer lugar, se consideran solo los robos de vivienda (burglary of habitation), considerando todas las subcategorías de este delito (con y sin entrada forzada e intentos de robo). A través del atributo *tipo de ubicación* (*type of location*) se identifican aquellos registros que corresponden a casas, descartándose los demás tipos de ubicaciones. En la figura 3.2 se observan las categorías de *tipo de ubicación* que presentan mayor cantidad de datos dentro de la categoría de robo de vivienda. De esta forma, solo se consideran para el análisis las categorías de residencias unifamiliares (*Single Family Residence*) tanto ocupadas como vacantes, descartando así departamentos y condominios, y aquellos datos mal clasificados dentro de la categoría de robo de vivienda, como senderos, colegios, restaurantes, instituciones religiosas, centros médicos, bares, hoteles, oficinas, aeropuertos y edificios. Así, se tiene un total de 17.008 registros correspondientes a robos a casas, lo cual representa un 33,92% del total de allanamientos.

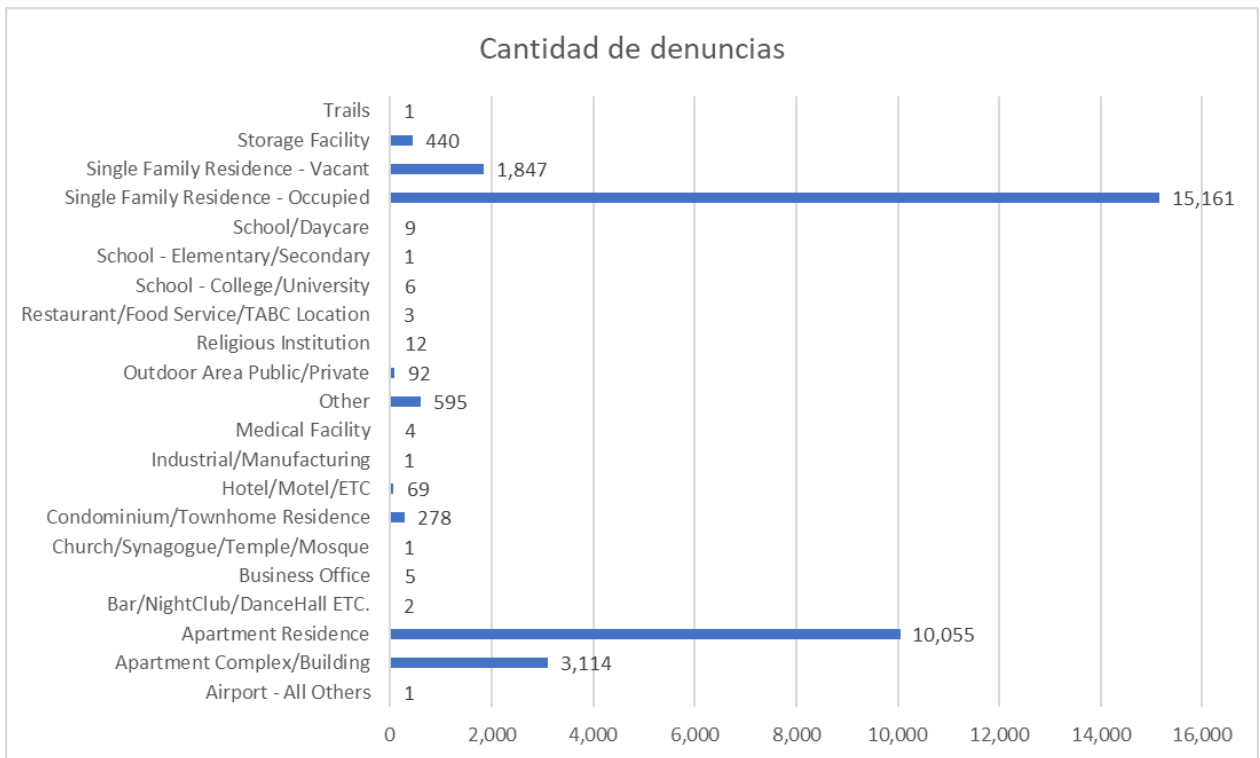


Figura 3.2: cantidad de denuncias por tipo de ubicación dentro de la categoría de robo de vivienda. Fuente: elaboración propia a través de la base de Dallas.

Como se dijo anteriormente, la localización es necesaria para la confección de las demás fuentes de datos, por lo que se descartan aquellos registros que no posean ni dirección ni coordenadas. En cuanto a los casos en los que se tiene la dirección, pero no las coordenadas, estas últimas son obtenidas a través del uso de la Api de geodificación de Google en Python 3.7. En la figura 3.3 se observa la cantidad de registros según su estado.

Estado de registros	Cantidad
Sin coordenadas ni dirección	166
Con dirección, pero sin coordenadas	1.158
Con coordenadas y dirección	15.684

} 16.842 registros

Figura 3.3: cantidad de registros según su estado. Fuente: elaboración propia a través de la base de Dallas.

De esta forma, se descartan 166 registros más de la base, por lo que se tiene un total de 16.842 registros que cumplen con los requerimientos de la memoria. Esto representa un 33,59% del total de delitos de allanamiento y un 3,41% del total de la base de denuncias.

3.3 Base de datos de puntos de no robo

La confección de un modelo de aprendizaje supone que para el entrenamiento del modelo se deben tener tanto ejemplos positivos como negativos, lo cual, en este caso, quiere decir registros de casas robadas y no robadas.

Al no existir un registro de casas que no han sido robadas, se debe idear la forma de construir esta base de datos, lo cual representa uno de los mayores desafíos de la memoria.

Para lograr esto, se utiliza el siguiente procedimiento:

1. Se utiliza el contorno de la ciudad de Dallas en formato SHP (shapefile) para generar puntos aleatorios dentro de la ciudad. Este formato contiene información de entidades geográficas que se pueden representar por medio de puntos, líneas o polígonos. En este caso, se tiene el polígono que contiene a la ciudad de Dallas. Esta información es obtenida a través de los servicios GIS de la ciudad, presente públicamente en la web.
Se generan 100.000 puntos aleatorios obteniéndose su latitud y longitud, por medio de un programa en python.
2. A través de la Api de Geocoding de Google, se obtiene la dirección de la calle más cercana a cada uno de los puntos por medio de geocodificación inversa. Se hace esto, dado que los sitios obtenidos pueden caer en lugares como ríos o terrenos baldíos, los cuales no son de utilidad en este caso.
3. A partir de las direcciones obtenidas, se obtiene la localización en formato latitud, longitud, lo cual es necesario para los pasos siguientes.
4. Se eliminan los puntos que estén repetidos.
5. Utilizando la base de robos, se descartan aquellos puntos que se encuentren a menos de 50 metros de algún punto de robo. Esto se hizo obteniendo, en primer lugar, el robo más cercano a cada punto generado aleatoriamente. Luego, se calcula la distancia entre ambos puntos y se evalúa si está a menos de 50 metros. Se utiliza el criterio de los 50 metros para asegurar que no se esté utilizando la misma casa tanto para robo como para no robo. Estos 50 metros corresponden aproximadamente a media cuadra [78].
Se obtienen 46.292 puntos que cumplen con los criterios anteriores.
6. Dado que sólo interesan aquellos puntos en donde hay casas, se opta por la construcción de una aplicación web, la cual, permite el etiquetado colaborativo de las imágenes de Google Street View correspondientes a los puntos obtenidos. Por medio de esta aplicación se obtuvo un total de 21.390 puntos que son casas. El funcionamiento y uso de esta se explica con mayor detalle en la sección 3.3.1.
7. Se agrega la fecha para cada registro de no robo, lo cual es relevante para la construcción de la base temporal.
Debido a la forma en que se construye esta base de datos, se puede decir que durante el periodo utilizado en este trabajo (junio de 2014 a diciembre de 2018) no ocurrieron robos en las casas obtenidas mediante la aplicación. De esta forma, las fechas se agregan de forma proporcional a las fechas del conjunto de robos, es decir, aquellas fechas con mayor cantidad de robos tendrán una mayor cantidad de no robos asociada.

3.3.1 Aplicación web: clasificador de imágenes

La creación de esta aplicación responde a la necesidad de identificar y seleccionar solo aquellos puntos correspondientes a casas. Esto no se podía asegurar, dada la generación aleatoria de los puntos.

Otras soluciones fueron descartadas dado el sesgo que podrían haber generado en el modelo. Por ejemplo, se pensó en la utilización de bases de datos de casas de acceso público en la ciudad de Dallas, como sitios de venta o arriendo, pero esto hubiese sido limitante, debido a que no todas las casas se encuentran en estos sitios.

La aplicación de clasificación de imágenes fue creada en lenguaje python y consiste en el uso de web scrapping para cargar la imagen proveniente de Google Street View a partir de la dirección de cada punto. Esta imagen se muestra en una interfaz web a la cual se puede acceder a través de cualquier navegador. La figura 3.4 muestra cómo se visualiza a través de Google Chrome.



Figura 3.4: visualización de la aplicación a través de Google Chrome.

Arriba de la imagen se tiene un contador con aquellas que faltan por etiquetar. Debajo de la imagen se tienen dos botones: “Hay fachada” o “No hay fachada”. Al momento de hacer clic en uno de los botones se guarda el registro en una base de datos sql y se carga otra imagen. En caso de que la persona no esté segura de que la imagen corresponda a una fachada o no, puede deslizar la imagen para pasarla sin responderla, sin embargo, esta debe ser clasificada para que el contador disminuya.

Para reducir el nivel de subjetividad y los errores involuntarios en la clasificación, cada imagen debe ser clasificada un mínimo de 3 veces para que sea etiquetada como casa o no casa de forma definitiva, donde la mayoría “gana” en caso de distintas clasificaciones para la misma imagen.

Excluyendo aquellos puntos que no tenían una imagen asociada en Google, se clasificó un total de 39.111 puntos, lo cual equivale a un mínimo de $39.111 \times 3 = 117.333$ etiquetas.

Al ser una gran cantidad de imágenes por clasificar se recurrió a la ayuda de muchas personas (en su mayoría conocidas), compartiendo el enlace a la aplicación web en redes sociales. La clasificación de casas duró 1 mes aproximadamente y participaron alrededor de 90 personas. En la tabla 3.1 se muestran los resultados obtenidos.

	Cantidad
Total casas	21.390
Total no casas	17.790
Sin URL	7.107

Tabla 3.1: resultados de la clasificación de casas para direcciones en las cuales no hubo robo.

De esta forma, se clasificaron 21.390 direcciones como casas, lo que representa un 54,70% del total de direcciones generadas.

3.4 Base de datos espacial

La información espacial consta de dos tipos de información. Por un lado, se tiene la información demográfica y socioeconómica de las zonas de robo y no robo, y por otra, la correspondiente a los puntos de interés cercanos a cada localización.

3.4.1 Información demográfica y socioeconómica

La información demográfica engloba aspectos relacionados al tamaño, estratificación y desarrollo de una comunidad humana, desde una perspectiva cuantitativa [79], mientras que la información socioeconómica hace referencia a factores sociales y económicos de una comunidad, tales como, estructura familiar, profesión, nivel de educación, ingreso, situación laboral, entre otros.

Para la obtención de esta información se hace necesario decidir sobre las áreas geográficas de las cuales se hará la extracción de los datos. En Estados Unidos las áreas se encuentran organizadas de forma jerárquica, la cual va de territorios más grandes a más pequeños como se muestra en la figura 3.5.

Se opta por caracterizar cada punto de robo y no robo por medio del “Census Tract” (zona censal), debido a que es de las divisiones más pequeñas, lo cual favorece a una mayor precisión en la descripción de cada casa, y debido a que la información está más completa en comparación a “Block Groups” donde hay muchos campos faltantes. Además, en Estados Unidos, las zonas censales están diseñadas para ser zonas relativamente homogéneas en cuanto a características de población, estado económico, y condiciones de vida [80], y son definidas con la intención de ser estables durante muchas décadas. El tamaño se mide según la cantidad de personas que abarca una zona censal, y generalmente varía entre 1.000 y 8.000 personas con un promedio de 4.000.

La zona censal no es un atributo presente en la base de robo, y tampoco en la de no robo, por lo que se extrae a través de la Api de Área de la Comisión Federal de Comunicaciones, la cual es una agencia independiente del gobierno de los Estados Unidos supervisada por el Congreso [81]. La obtención de la zona censal se hace a través de la latitud y longitud de cada punto.

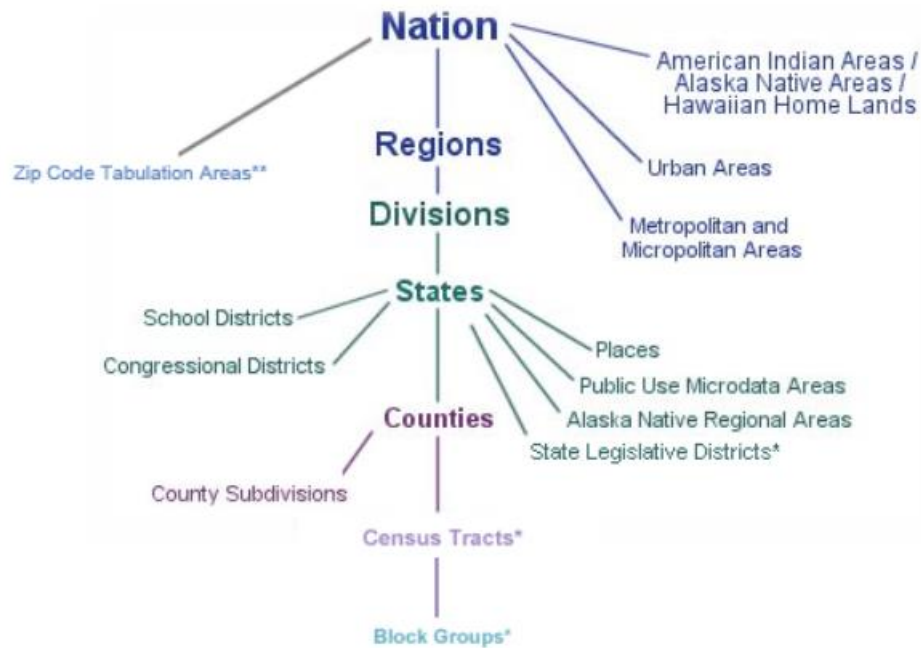


Figura 3.5: jerarquía geográfica en Estados Unidos. Fuente: United States Census Bureau.

Con el tramo censal definido para cada localización, se procede a extraer la información demográfica y socioeconómica. Esto se hace con los datos provenientes de la Encuesta a la Comunidad Americana 2017 (ACS), la cual era la más actual al momento de la captura de datos. Los resultados de la encuesta se encuentran públicos en la web en el portal American FactFinder, el cual corresponde a la Oficina del Censo de Estados Unidos en línea. Se extraen 4 archivos en formato csv (archivo delimitado por comas) con información de características económicas, sociales, de población y de vivienda, según zona censal.

A continuación, se enumeran los atributos seleccionados, presentándose en primer lugar, la descripción, y luego el código proveniente de la fuente y utilizado posteriormente en el modelo.

1. Ingreso medio por hogar - HCo3_EST_VCo2
2. Fuerza laboral, desde los 16 en adelante - HCo3_VCo4
3. Porcentaje de desempleados en fuerza laboral - HCo3_VCo7
4. Tasa de desempleo - HCo3_VCo12
5. Casa con hijo menor a 6 años con ambos padres en fuerza laboral - HCo3_VCo20
6. Porcentaje de gente que va solo en auto a trabajar - HCo3_VCo28_1
7. Porcentaje de gente que comparte auto para ir al trabajo - HCo3_VCo29_1
8. Porcentaje de gente que ocupa el trans público para ir al trabajo - HCo3_VCo30
9. Porcentaje de gente que camina al trabajo - HCo3_VCo31
10. Porcentaje de gente que utiliza otros medios para ir al trabajo - HCo3_VCo32
11. Porcentaje de gente que trabaja desde la casa - HCo3_VCo33
12. Porcentaje de gente con seguro de salud - HCo3_VCo131
13. Porcentaje de gente con seguro de salud privado - HCo3_VCo132
14. Porcentaje de gente con seguro de salud público - HCo3_VCo133_1

15. Porcentaje de familias y personas cuyo ingreso en los últimos 12 meses estuvo por debajo del nivel de pobreza - HCo3_VC171
16. Unidades totales de vivienda - HCo1_VCo3
17. Porcentaje de ocupación de viviendas - HCo1_VCo4
18. Porcentaje de estructuras construidas desde el 2014 en adelante - HCo3_VC27
19. Porcentaje de estructuras construidas desde el 2010 al 2013 - HCo3_VC28_2
20. Porcentaje de estructuras construidas entre 2000 y 2009 - HCo3_VC29_2
21. Porcentaje de estructuras construidas antes de 1939 - HCo3_VC36
22. Cantidad promedio de habitaciones por casa - HCo1_VC50
23. Porcentaje de viviendas sin habitaciones - HCo3_VC55
24. Porcentaje de viviendas con más de 5 habitaciones - HCo3_VC60
25. Porcentaje de viviendas ocupadas por sus dueños - HCo3_VC65
26. Porcentaje de viviendas ocupadas por inquilinos - HCo3_VC66
27. Porcentaje de viviendas sin vehículos disponibles - HCo3_VC85
28. Porcentaje de viviendas con 1 vehículo disponible - HCo3_VC86
29. Porcentaje de viviendas con 2 vehículos disponibles - HCo3_VC87
30. Porcentaje de viviendas con 3 o más vehículos disponibles - HCo3_VC88
31. Valor promedio de las casas ocupadas por sus dueños - HCo1_VC128
32. Porcentaje de unidades ocupadas por sus dueños con hipoteca - HCo3_VC133_2
33. Porcentaje de casas donde la mujer es sostenedora sin presencia de marido - HCo3_VC10
34. Porcentaje de casas con una o más personas menores de 18 años - HCo3_VC17
35. Porcentaje de casas con una o más personas mayores de 65 años - HCo3_VC18
36. Tamaño promedio de la casa - HCo1_VC21
37. Tamaño promedio de la familia - HCo1_VC22
38. Porcentaje de personas con secundaria completa o superior - HCo3_VC95
39. Porcentaje de personas con licenciatura o mayor - HCo3_VC96
40. Porcentaje de personas con discapacidad - HCo3_VC106
41. Porcentaje de casas con un computador al menos - HCo3_VC127
42. Porcentaje de casas con una suscripción a internet banda ancha - HCo3_VC218
43. Edad promedio de la población total - HCo1_EST_VC12
44. Porcentaje de población que son hombres - HCo1_EST_VC15
45. Porcentaje de población que son mujeres - HCo1_EST_VC16
46. Porcentaje de gente blanca de una raza - HCo1_EST_VC20
47. Porcentaje de gente negra o afroamericana - HCo1_EST_VC21
48. Porcentaje de gente india americana o de Alaska - HCo1_EST_VC22
49. Porcentaje de gente asiática - HCo1_EST_VC23
50. Porcentaje de gente nativa de Hawái u otra isla del pacífico - HCo1_EST_VC24
51. Porcentaje de gente de otra raza - HCo1_EST_VC25
52. Porcentaje de gente de dos razas o más - HCo1_EST_VC26
53. Porcentaje de gente hispana o latina (cualquier raza) - HCo1_EST_VC28
54. Porcentaje de gente sólo blanca (ni hispánica ni latina) - HCo1_EST_VC29
55. Porcentaje de población nunca casada - HCo1_EST_VC39
56. Porcentaje de población casada ahora, excepto separada - HCo1_EST_VC40
57. Porcentaje de población divorciada o separada - HCo1_EST_VC41
58. Porcentaje de población viuda - HCo1_EST_VC42
59. Porcentaje de población graduada o con un título profesional - HCo1_EST_VC50
60. Población total - HCo1_EST_VC01

Se seleccionan estas características, puesto que, en primer lugar, los estudios [60], [62] y [63] señalan la importancia de incorporar información espacial socioeconómica y demográfica en la predicción de crimen. Además, se busca agregar variables que sean heterogéneas entre sí, de forma de tener un panorama más completo de cada evento. Por último, estas características engloban a otras que no son seleccionadas lo cual ayuda a prevenir un problema de redundancia que, posteriormente, pueda afectar al modelo.

Finalmente, a través del manejo de bases de datos con la librería “pandas” en Python, se agregan estas características para cada localización en las bases de robo y no robo según zona censal.

3.4.2 Puntos de interés

Un punto de interés se define como “un punto de ubicación específica que alguien puede encontrar útil o interesante” [82]. En ese sentido, el común de la gente ocupa este término para referirse a hoteles, campings, estaciones de servicios, restaurantes y otras categorías útiles que suelen visualizarse en los GPS actuales.

En la mayoría de las investigaciones expuestas en la sección 2, y en relación con el comportamiento criminal en el robo a casas, se ha observado que estos criminales presentan una conducta racional maximizando su utilidad esperada. De esta forma, los ladrones de casas evalúan factores físicos, sociales y personales a la hora de cometer un delito, minimizando el riesgo de ser atrapados.

Los puntos de interés son de utilidad para la caracterización de una zona. Por ejemplo, una alta densidad de bares y restaurantes en un sector, pueden indicar una alta concurrencia de personas en la tardes y noches. Así, estos puntos podrían influir en la predicción de robo, en la medida que esto sea visto por parte de los delincuentes como un indicador de mayor vigilancia o de mayor resguardo policial en la zona.

Para la construcción de la base se utiliza la API Places – Nearby Search, la cual es una API de Google que permite obtener puntos de interés tomando como entrada un punto geolocalizado (latitud y longitud) y un radio a la redonda de este. Es importante señalar que cada solicitud entrega un máximo de 60 puntos, por lo que a través de distintas pruebas se decide obtener los puntos de interés en un radio de 250 metros para cada localización de robo y no robo, de manera de que los puntos que no se consideran debido al límite de la Api, sean los menos. Además, no todos los puntos de la ciudad están etiquetados por Google, pero dentro de las empresas que ofrecen este tipo de servicios, Google cuenta con una de las bases más robustas, por lo que este servicio es de los mejores disponibles y se piensa que esta limitación no afectará de gran forma a los resultados del modelo.

Las consultas a esta API tienen como salida un objeto JSON (notación de objetos de JavaScript), el cual cuenta con una serie de atributos, además del nombre y localización de cada lugar encontrado. Por ejemplo, la salida puede incluir para cada punto de interés, dirección completa, geometría del lugar, horario de apertura, fotos, calificación, categorías del lugar, entre otros. De estos atributos el que se considera para el trabajo de memoria es la categoría del lugar. Google cuenta con más de 100 categorías distintas [83], por lo que se decide agrupar todas estas en 9 macro categorías, las cuales se definieron a juicio del

memorista, buscando la homogeneidad dentro de cada. Estas categorías son: entretenimiento, salud, restaurantes, tiendas, belleza, servicios, ciudad, transporte y turismo, y otros. Las categorías que agrupa cada una de estas se detallan en Anexos A.

Finalmente, lo que se busca es obtener la cantidad de puntos de interés de cada macro categoría para cada localización de las bases de robo y no robo en un radio de 250 metros. Esto se lleva a cabo por medio de un programa escrito en python. En la tabla 3.2, se expone un ejemplo para los 3 primeros puntos de robo de cómo está formada la base de datos de puntos de interés.

id	lat	lng	entretenimiento	salud	restaurantes	tiendas	belleza	servicios	ciudad	transporte y turismo	otros
1	32.96	-96.77	2	12	5	11	2	2	2	4	18
2	32.72	-96.77	1	1	0	0	0	4	0	6	4
3	32.83	-96.91	2	1	1	9	4	11	0	7	25

Tabla 3.2: muestra de la base de datos de puntos de interés para 3 direcciones diferentes. Fuente: elaboración propia utilizando Api Places.

De esta forma, la base de datos espacial está conformada por 69 variables: 60 demográficas y socioeconómicas, y 9 correspondientes a las categorías de puntos de interés.

3.5 Base de datos temporal

En [74] se concluye que las condiciones climáticas influyen en el robo a propiedades. De esta forma, se incluyen las variables de mayor relevancia en el estudio: temperatura y nivel de precipitación.

Por otro lado, en [76] se señala que cuando ocurre un robo en una zona, los alrededores de esta tienen un mayor riesgo de ocurrencia de delito en el corto plazo, lo que se denomina victimización reiterada. Dado esto, se incluye la cantidad de robos ocurridos en la zona censal en la se encuentra cada punto.

Uno de los objetivos de la memoria es realizar una predicción de robo a futuro, por lo que, tanto las variables climáticas como la cantidad de robos, son utilizadas como series temporales, es decir, para cada punto se tiene información histórica anterior a la fecha del registro, que se utiliza para hacer una predicción hacia una ventana de tiempo futura. Así, se busca obtener la información de días anteriores tanto para las variables climáticas como para la cantidad de robos.

La temperatura y el nivel de precipitación son obtenidos desde la Administración Nacional Oceánica y Atmosférica (NOAA) del Departamento de Comercio de los Estados Unidos. Esta información proviene de una de las estaciones climatológicas de Dallas de nombre Dallas Love Field y contiene registros de hace más de 10 años. Además, los datos son almacenados por hora, sin embargo, para los fines de esta memoria, se utiliza el promedio diario.

La cantidad de robos en el día asociada a cada registro se obtiene a través de un programa en python que hace un conteo de la cantidad de registros por tramo y por fecha para luego almacenarlo en una variable.

Un aspecto importante en cuanto a la construcción de esta base de datos es cuánto tiempo hacia atrás se incluye por registro y cómo se incluye este periodo (por día, por semana o por mes). Esto también influye en la decisión respecto a la ventana de tiempo que se quiere predecir, dado que si los datos se almacenan de forma mensual no tiene sentido predecir de forma semanal o diaria. De esta forma, se opta en primer lugar, por un periodo de tiempo semanal, ya que, en la práctica, es de mayor utilidad hacer una predicción semanal que mensual, dado que de una semana a otra puede haber cambios significativos en las condiciones climáticas o en la cantidad de robos en una determinada zona. Por otro lado, se descarta una predicción diaria, puesto que la cantidad de valores faltantes es significativamente mayor comparada al rango semanal, lo que podría afectar en el rendimiento del modelo al momento de tratar con estos datos. En cuanto a la información pasada, se incluye la información de un mes hacia atrás para cada registro, puesto que los estudios mencionados manifiestan una influencia en el corto plazo del clima y la cantidad de robos en una zona con respecto a la predicción de robo. Así, para cada registro se tiene la temperatura y el nivel de precipitación promedio semanal de Dallas para las 4 semanas anteriores, y la cantidad total semanal de robos ocurridos en la zona censal correspondiente para las 4 semanas anteriores.

Finalmente, dado que en esta base se utiliza información secuencial a raíz de los eventos anteriores, se elimina de la base el primer mes (junio de 2014), ya que no se tienen los registros para mayo de ese año.

En resumen, los atributos incluidos en la base temporal son los siguientes:

- Temperatura: temperatura promedio semanal para las 4 semanas anteriores de cada registro, medida en grados Celsius.
- Precipitación: Nivel de precipitación promedio semanal para las 4 semanas anteriores de cada registro, medido en pulgadas.
- Cantidad de robos ocurridos: para cada localización, se tiene la suma total de robos y no robos semanales ocurridos en la zona censal para las 4 semanas anteriores de cada registro.

De esta forma, se tiene información de 4 semanas por cada uno de los 3 atributos anteriores, siendo un total de 12 variables temporales.

3.6 Base de datos de información contextual

La información contextual se refiere a los datos que dan contexto a una persona, entidad o evento [84]. En este caso, esta información está representada por las imágenes de las casas presentes en las bases de robo y no robo, las cuales dan contexto a los lugares donde ocurrieron estos eventos.

Para obtener las imágenes mencionadas se utiliza la Api de Google Street View y se realiza un programa en Python de manera de obtener estas de forma automática dada la gran cantidad de registros que se tienen. Por medio de esta Api, las imágenes son descargadas a través de la latitud y longitud de cada registro. Además, se utilizan los parámetros de tamaño y ángulo de la foto.

El objetivo principal en este caso es obtener la imagen de la fachada de cada casa, sin embargo, esto no siempre es seguro con una sola imagen en el mismo ángulo para todos los registros. Por ejemplo, si se solicitan las imágenes para cada punto con ángulo 0° , no siempre se tendrá la foto de la fachada de la casa, y ocurre lo mismo para los demás ángulos. De esta forma, se decide utilizar una vista panorámica del lugar del evento, descargando 4 imágenes por localización: 0° , 90° , 180° y 270° . Esto, además de asegurar la presencia de la fachada en las imágenes, entrega más información contextual al modelo, lo que podría ser un aporte en la predicción final. La figura 3.6 muestra un ejemplo de las 4 imágenes que son descargadas para uno de los ejemplos de la base de robo.



Figura 3.6: imágenes en 0, 90, 180 y 270 grados de uno de los registros de la base de robo. Fuente: fotos provenientes de la Api de Google Street View.

Por último, las imágenes son descargadas con tamaño 640 x 640 píxeles, que es el máximo permitido por la Api. Esto, con el objetivo de obtener la mayor resolución posible.

3.6.1 VGG16

Las imágenes recolectadas no son utilizadas directamente en el modelo de red neuronal, sino que, se hace una extracción de características como en [29] antes de la entrada al modelo. El motivo de esto es que actualmente existen distintos modelos pre entrenados que pueden ser utilizados rápidamente a través de librerías como Keras, lo que disminuye significativamente el tiempo de entrenamiento. La extracción de características se obtiene a través de la salida de la primera capa densa o fully connected, por lo que, de cada imagen, se obtiene un vector de 4.096 características lo que es equivalente al tamaño de salida de esta capa (cantidad de neuronas).

Para esto, el modelo utilizado es el VGG16 como se mencionó en la sección 2.3.2.1. El modelo se encuentra pre entrenado con las pesas del conjunto de datos “Imagenet”, el cual cuenta con más de 14 millones de imágenes pertenecientes a 1.000 clases distintas.

El modelo recibe imágenes de tamaño 224 x 224, por lo que en primer lugar se hace un redimensionamiento de las imágenes descargadas. Además, se utiliza un filtro anti-aliasing por medio de la interpolación lineal [85] de manera que la disminución en el tamaño de la imagen afecte lo menos posible a la calidad de esta y así no perjudique la información que pueda extraer la red.

Finalmente, las imágenes son transformadas a arreglos numpy, lo cual también es requisito para que puedan ser procesadas por el modelo. La implementación de esto se

hace en Google Colaboratory (detallado en la sección 3.8), debido al alto costo computacional en el que se incurre dada la gran cantidad de imágenes.

De esta forma, esta base cuenta con 16.384 características por registro, ya que cada uno cuenta con 4 imágenes asociadas y cada imagen entrega un vector de 4.096 características.

3.7 Base de datos final

Durante la adquisición de los datos necesarios para la construcción de las bases antes descritas, es común que para algunos registros no se tuviese la información o no hubiese una imagen proveniente de Google Street View, por lo que se opta por eliminar dichas localizaciones, lo que se traduce en una reducción en el tamaño de las bases de robo y no robo. Además, se elimina el mes de junio de 2014 de todas las bases de datos, debido a la construcción de la base temporal antes mencionada.

Debido a esta reducción la base de robos resultante cuenta con 16.234 casos, lo que representa un 32,37% del total de robos a propiedades y un 94,10% de la base anterior. Por otro lado, la base de no robos consta de 16.966 registros luego de la reducción, lo que representa un 79.32% del total de casas obtenidas por medio de la aplicación.

En consecuencia, los datos de entrada al modelo predictivo están constituidos por 33.200 registros, 16.234 robos (48,90%) y 16.966 no robos (51,10%), con 69 características espaciales (60 demográficas y socioeconómicas, y 9 de puntos de interés), 12 variables temporales y 16.384 características de información contextual, lo que da un total de 16.465 características. El detalle de estas variables se puede observar en la tabla 3.3.

Tipo de características	Características	Fuente
Espacial	- 60 demográficas y socioeconómicas	- American FactFinder (Oficina del Censo en línea)
Temporal	- 9 categorías de puntos de interés	- Places Api de Google
	- 8 de condiciones climáticas: 4 semanas de historia para el promedio de la temperatura y el nivel de precipitación	- Administración Nacional Oceánica y Atmosférica (NOAA)
Contextual	- 4 de cantidad de registros: 4 semanas de historia para la cantidad registros (robos y no robos) por zona censal	- Denuncias policiales en la ciudad de Dallas para los robos, y a través de la aplicación de clasificación de casas para no robos
	- Imágenes: 4 fotos por registro, en ángulos 0°, 90°, 180° y 270°. A través de la red VGG16 se obtienen 4.096 características por imagen, lo que resulta en un total de 16.384 características	- Api de Google Street View

Tabla 3.3: resumen de las características recolectadas y sus fuentes.

3.8 Tecnologías involucradas

Para la realización de algunas tareas relacionadas al proceso KDD, se utiliza Python como lenguaje de programación, dado que existe un gran soporte a nivel web y gran variedad de paquetes que facilitan la implementación de modelos de aprendizaje automático.

Adicionalmente, se utilizan API's de Google para obtener información de Dallas. “Una Api es un conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como capa de abstracción” [86]. Específicamente, se hace uso de: Api de Google Street View, la cual permite descargar imágenes de las direcciones de casas de Dallas utilizadas en la memoria; Geolocation, que es utilizada para georreferenciar una dirección y obtener coordenadas a partir de una dirección y viceversa; Api de Google Places, para la obtención puntos de interés cercanos a los registros; y la Api de Área de la Comisión Federal de Comunicaciones, para la obtención de la zona censal en la que se encuentra cada registro.

También, se utiliza Python para la implementación de los modelos de aprendizaje automático (Random Forest y Support Vector Machines) y aprendizaje profundo (Convolutional Neural Networks, Deep Neural Networks y Recurrent Neural Networks) utilizados en la memoria. Para los primeros se utiliza la librería *Sklearn*, mientras que para los de aprendizaje profundo se utiliza la librería *Keras*, la cual está precisamente diseñada para la rápida implementación de estos modelos. La utilización de lenguaje Python se justifica, en primer lugar, por el conocimiento que tiene el estudiante con respecto a este. Por otro lado, según información presente en la web y en la literatura, hoy en día, es ampliamente utilizado en la resolución de problemas similares al presentado en este trabajo, dada la vasta cantidad de bibliotecas especializadas relacionadas al aprendizaje automático y la facilidad de uso del lenguaje en sí [87].

Uno de los principales problemas en cuanto a la implementación de estos modelos (específicamente en cuanto al procesamiento de imágenes), se debe al costo computacional incurrido, lo que significa que el tiempo que demoran en ejecutarse estos modelos es muy grande, lo cual no permite obtener resultados en un tiempo óptimo. Para lidiar con esto, se utiliza Google Colaboratory (Colab), el cual es un entorno gratuito de Jupyter Notebook (por lo que se puede desarrollar código en Python y otros lenguajes), donde se pueden ejecutar modelos a través de la librería de Keras usando GPU (tarjetas gráficas) en vez de CPU, lo cual disminuye los costos de procesamiento y entrenamiento de los modelos de aprendizaje automático y profundo, al ser mucho más eficiente en la multiplicación y convolución de matrices [88]. Como alternativa, se encuentra Amazon Web Service, pero se elige Google Colab, dado el conocimiento del estudiante en el uso de Jupyter Notebook y a su uso gratuito.

4. Selección de características

En la sección anterior se detalló el procedimiento de la recolección de las distintas características que se utilizan como entrada al modelo. Estas características se clasifican en espaciales, temporales y contextuales y representan el 0,42%, 0,07% y 99,51% respectivamente, siendo un total de 16.465 características.

Dicha cantidad de características motiva el uso de algún método de reducción de variables, debido a que, en primer lugar, existe la posibilidad de presencia de correlación y redundancia entre variables lo cual podría afectar negativamente los resultados del modelo. En segundo lugar, el costo computacional crece a medida que el número de características aumenta, y, por ende, también lo hace el tiempo de ejecución del modelo, por lo que la reducción de características ayuda a mejorar este aspecto.

La selección de características es el proceso en el que se selecciona automática o manualmente las características que más contribuyen a la predicción de la variable de salida del modelo. Los beneficios de esta técnica son: reducir el sobreajuste, mejorar la precisión y reducir el tiempo de entrenamiento [89].

En este trabajo se utilizará el algoritmo *Recursive Feature Elimination* (RFE) para la selección de características, el cual, como su nombre lo indica, elimina características de forma recursiva. Esto se logra, en primer lugar, a través de la construcción de un modelo en todo el conjunto de predictores, calculando la importancia de cada uno. Luego, se eliminan los predictores menos importantes y se construye un nuevo modelo con los restantes y se vuelve a calcular la importancia de estos. En la práctica, el analista debe especificar el número de subconjuntos de predictores para evaluar, así como el tamaño de cada subconjunto [90].

El algoritmo RFE es aplicado sobre modelos de clasificación lineal o con estructura de árboles, dado que se debe identificar la importancia de las características para hacer la eliminación iterativa. Debido a esto, el algoritmo no funciona con modelos no lineales, como las redes neuronales. De esta forma, el modelo que se utiliza es el *Random Forest* (Bosque Aleatorio), previamente introducido en el capítulo 2. Se escoge este modelo para ser utilizado con RFE, puesto que, en primer lugar, el bosque aleatorio tiende a no excluir variables de la ecuación de predicción, lo cual es relevante para identificar aquellas características menos relevantes y eliminarlas. En segundo lugar, el bosque aleatorio tiene incorporado un método interno para la importancia de las características a través de un ranking, lo cual puede ser usado en la primera iteración del RFE cuando se tiene la totalidad de variables. Además, presenta robustez, facilidad de uso y un buen rendimiento en grandes conjuntos de características.

Se decide utilizar el RFE de manera de obtener distintas cantidades de características, y luego con estos subconjuntos, evaluar el modelo de red neuronal final a través de las métricas presentadas en 2.4. El RFE se utiliza sobre las características de la información espacial y contextual, excluyendo las variables temporales. Esto se debe a que la base temporal tiene una estructura de serie temporal distinta a las demás bases y se quiere observar la influencia de esta secuencia al momento de juntarse con las variables más importantes de los otros conjuntos de información. De esta forma, la cantidad total de características sobre las cuales se implementa el RFE son 16.453 (16.465 - 12), y se

eligen 4 escenarios para la evaluación del modelo según la metodología utilizada en [14]. Estos son:

1. 10.000 características
2. 1.000 características
3. 100 características
4. 10 características

Como se puede observar, la disminución de características por escenario es de un orden de magnitud y representan el 60,78%, 6,07%, 0,60% y 0,06% de la cantidad total de características respectivamente.

Antes de implementar el algoritmo RFE, se ejecuta el bosque aleatorio y se ajustan sus hiperparámetros en función de las métricas de pérdida y accuracy detalladas en la sección 2.4 con el objetivo de obtener un mejor modelo antes de la selección de características. Estos hiperparámetros son ajustados según los resultados del entrenamiento del modelo en el conjunto de validación. Este escenario es presentado en la tabla 4.1 y también es utilizado en el aprendizaje de la red neuronal final.

Nº de muestras de entrenamiento	Nº de muestras de validación	Nº de muestras de prueba	Fechas de entrenamiento	Fechas de prueba
26.224	1.381	5.595	07-2014: 12-2017	01-2018: 12-2018

Tabla 4.1: escenario para el ajuste de hiperparámetros del bosque aleatorio.

En principio, las muestras de validación y de entrenamiento se tienen en el mismo conjunto. Antes de entrenar el modelo, el conjunto de validación es excluido, representando un 5% del conjunto inicial. Estos datos corresponden a los meses de diciembre de los años 2015, 2016 y 2017.

El bosque aleatorio se ejecuta por medio de la librería *Sklearn*, en Python. Según la documentación de dicha librería [91], los hiperparámetros más importantes para ajustar son la cantidad de árboles de decisión que componen el bosque aleatorio (*n_estimators*) y el número máximo de características consideradas para dividir un nodo (*max_features*), por lo que se comienza con el ajuste de estos parámetros.

Dado lo anterior, en primer lugar, se prueban distintas configuraciones para la cantidad de árboles de decisión. En la figura 4.1 se puede observar el *log loss* (pérdida de entropía cruzada) y el *accuracy* para el conjunto de validación, conforme aumenta la cantidad de árboles. A medida que esta cantidad crece, también lo hace su costo computacional, lo que implica un aumento en el tiempo de ejecución. De esta forma, se opta por 240 árboles, dado que, solo los bosques aleatorios con alrededor de 800 árboles presentan mejores métricas, sin embargo, esto equivale a un tiempo de ejecución mucho mayor (14 minutos más aproximadamente).

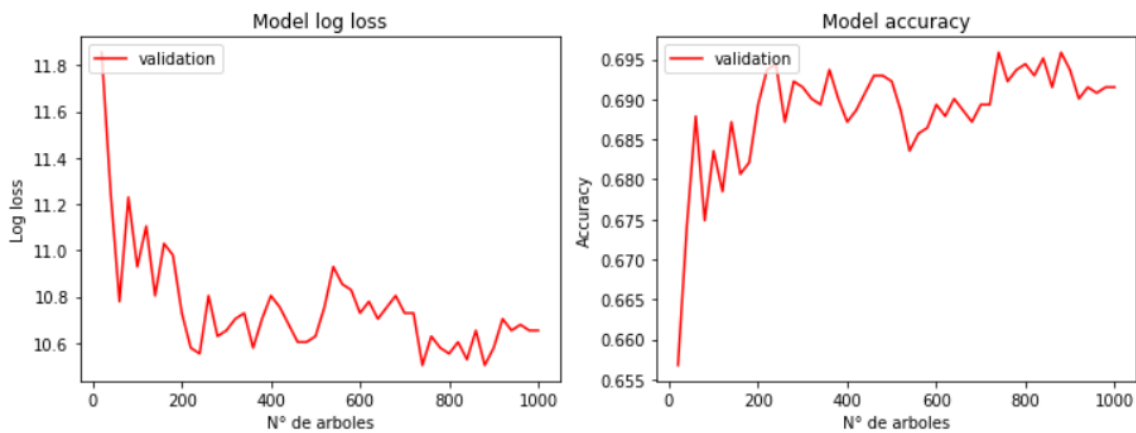


Figura 4.1: evolución de métricas *log loss* (izquierda) y *accuracy* (derecha) en el conjunto de validación a medida que se aumenta el número de árboles para el bosque aleatorio. Fuente: autor.

El siguiente parámetro que se ajusta es el número máximo de características consideradas para dividir un nodo. Según la documentación del modelo [91], las configuraciones “*sqrt*” o “*auto*” han mostrado buenos resultados empíricos para tareas de clasificación. Esta configuración utiliza una cantidad máxima de características para dividir un nodo igual a la raíz cuadrada de la cantidad total de características que se tengan. Dado esto, se realizan iteraciones del bosque aleatorio para configuraciones cercanas a dicha cantidad. Así, *max_features* toma valores entre 20 y 200, donde 128 es aproximadamente el valor de ‘*auto*’. La figura 4.2 muestra las métricas *log loss* y *accuracy* para las distintas configuraciones.

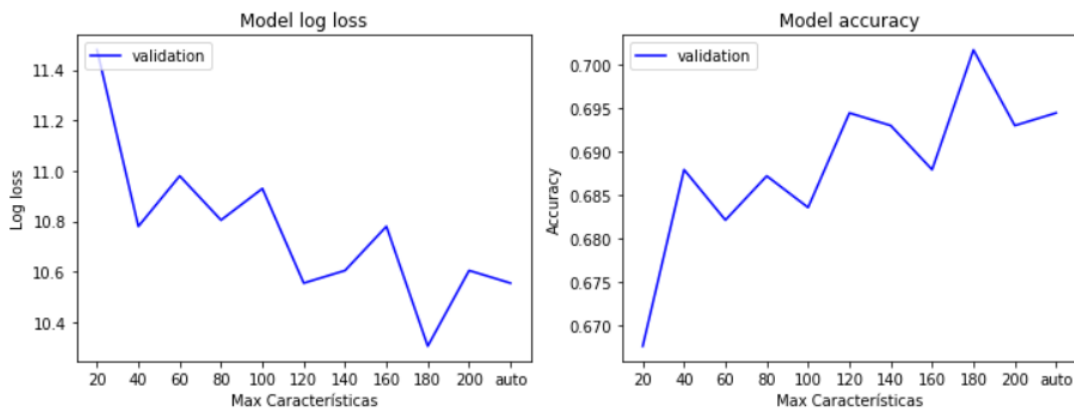


Figura 4.2: evolución de métricas *log loss* (izquierda) y *accuracy* (derecha) en el conjunto de validación a medida que se aumenta el número máximo de características. Fuente: autor.

El bosque aleatorio con *max_features* igual a 180 es el que muestra mejores métricas. Sin embargo, al momento de hacer la selección de 100 y 10 características, esta configuración sería inviable, puesto que este hiperparámetro debe estar entre 0 y la cantidad de características que se tiene. De esta forma, se elige la configuración ‘*auto*’, que también presenta buenas métricas.

Para el resto de los hiperparámetros del bosque aleatorio se realiza una grilla de búsqueda aleatoria y luego una grilla de búsqueda, de forma de obtener aquellos que

presenten mejores métricas. A través del primer método se puede definir una cuadrícula de rangos de hiperparámetros y muestrear aleatoriamente combinaciones (definiendo la cantidad deseada) de la cuadrícula, realizando K-Fold CV con cada combinación de valores [92], la cual es una técnica que cambia de manera aleatoria los conjuntos de entrenamiento y prueba de forma que las métricas obtenidas sean más robustas y no se correspondan sólo a un conjunto de prueba determinado. Con esto, se reducen los rangos para cada hiperparámetro y luego con el método de grilla de búsqueda se realizan todas las combinaciones posibles para obtener los mejores hiperparámetros.

De esta forma la primera grilla consta de 40 combinaciones de hiperparámetros generados aleatoriamente y cada una con 2 pruebas de validación cruzada ($cv=2$). Se tienen los siguientes rangos: entre 10 y 100 para la profundidad máxima (max_depth), sumado a *None* (predeterminado); entre 0,1% y 0,5% de la cantidad de muestras, sumado al valor predeterminado 2 para $min_samples_split$; entre 0,1% y 0,5% de la cantidad de muestras, más el valor predeterminado 1 para $min_samples_leaf$; y *True* o *False* para Bootstrap.

Luego de obtener los resultados de la primera grilla, los rangos se reducen, por lo que la segunda grilla consta de 8 combinaciones y $cv=3$. Los valores para cada hiperparámetro en este caso son: 20 y *None* para max_depth ; 0,1% y 2 para $min_samples_split$; y *True* o *False* para Bootstrap. Finalmente, las 2 mejores combinaciones se prueban en el conjunto de validación.

Así, los hiperparámetros seleccionados son los siguientes:

- Cantidad de árboles de decisión ($n_estimators$): 240. La cantidad de árboles afecta en la cantidad de predicciones distintas que se tengan. Es decir, a mayor cantidad de árboles, la predicción del bosque aleatorio será más robusta. Sin embargo, al aumentar esta cantidad, mayor será el tiempo de ejecución del modelo, dado que el costo computacional aumenta.
- Máximo de características ($max_features$): *auto*. Número máximo de características consideradas para dividir un nodo. Empíricamente se ha mostrado buenos resultados para la raíz cuadrada de la cantidad de características que se tienen, esto es *auto* o *sqrt* (en este caso, ese valor sería 128 aproximadamente).
- Profundidad máxima (max_depth): *None* (predeterminado). Representa la profundidad máxima de cada árbol en el bosque. A mayor profundidad, más divisiones tiene y el bosque captura más información de los datos. Al ser *None*, los nodos se expanden hasta que todas las hojas sean puras o hasta que las hojas contengan menos de $min_samples_split$ muestras [91].
- Número mínimo de muestras necesarias para dividir un nodo interno ($min_samples_split$): 2 (predeterminado). Al aumentar este parámetro, el bosque se vuelve más limitado, dado que tiene que considerar una mayor cantidad de muestras en cada división.
- Número mínimo de muestras necesarias para estar en un nodo hoja ($min_samples_leaf$): 1 (predeterminado). Parámetro similar al anterior, sin embargo, este describe el número mínimo de muestras en las hojas, es decir, la base del árbol. Al aumentar este valor el bosque también se vuelve más limitado.
- Bootstrap: *Verdadero*. Puede ser “verdadero” o “falso”. Si es verdadero, cada árbol se construye a partir de una muestra de los datos de entrenamiento con reemplazo.

Es razonable esta elección, puesto que, al haber construido la base de no robos de manera artificial, no se sabe la proporción real entre robos y no robos, por lo que es mejor dejar que el modelo lo decida.

Luego de fijar los hiperparámetros del bosque aleatorio, se procede a implementar el algoritmo RFE. Según el libro “Feature Engineering and Selection: A Practical Approach for Predictive Models” de Khun y Johnson [90]. A pesar de que el rendimiento predictivo del bosque aleatorio no disminuye con la presencia de variables altamente correlacionadas, la importancia de las características sí se ve afectada con este aspecto (Anexos B), por lo que es necesario quitar las variables redundantes antes de ejecutar el RFE.

Para identificar las variables altamente correlacionadas, se utiliza el índice de correlación de Pearson y se obtiene una matriz de correlaciones para cada par de variables (Anexos C). Esto, se hace solo para la base de información espacial, puesto que carece de sentido hacerlo para la base contextual, dado que la información proviene de una extracción de características previa, por lo que no es interpretable.

Con esto, se identifican y eliminan aquellas variables que presenten una correlación, en valor absoluto, superior a 0.9 [93]. Así, se eliminan 11 variables espaciales redundantes, lo que implica que son 16.442 características las utilizadas para el RFE, 58 espaciales y 16.384 contextuales.

Cabe señalar que se decide eliminar el 10% de las características en cada iteración del RFE. Esto, de manera que el algoritmo se ejecute más rápido, dada a la gran cantidad de características.

4.1 Composición de los conjuntos de características

En esta sección se describe y analiza por separado los conjuntos de características obtenidos a través del RFE sobre bosque aleatorio.

4.1.1 Selección de 10.000 características

Las 10.000 características más importantes obtenidas por el RFE corresponden a 55 características espaciales, y 9945 características contextuales. Esto corresponde al 94,83% de la información espacial y al 60,70% de la información contextual. De esta forma, solo 3 características de la información espacial son desechadas, mientras que una parte importante de las características contextuales son eliminadas por el algoritmo. Las 3 variables espaciales eliminadas son: porcentaje de gente nativa de Hawái u otra isla del pacífico, porcentaje de población casada excepto separada y porcentaje de población graduada o con un título profesional. Estas corresponden a características de población total y nativa.

4.1.2 Selección de 1.000 características

Los resultados de esta selección entregan 47 características espaciales (81,03% del total de espaciales) y 953 contextuales (5,82% del total de contextuales).

Nuevamente se puede apreciar que la mayor parte de las características espaciales son consideradas como relevantes para el algoritmo. Con respecto a los puntos de interés, 7 de las 9 categorías son eliminadas por este, quedando las categorías ‘otros’ y ‘turismo y transporte’.

4.1.3 Selección de 100 características

En este caso, la selección entrega un total de 39 características espaciales y 61 características contextuales. Esto representa el 67,24% del total de características espaciales y 0,36% del total de características contextuales, respectivamente.

La mayor parte de las características espaciales seleccionadas en este caso pertenecen a características económicas y de vivienda. La categoría ‘otros’, es la única seleccionada con respecto a los puntos de interés.

4.1.4 Selección de 10 características

Finalmente, las características que componen a esta selección son únicamente espaciales, lo que corresponde al 17,24% del total de estas características. La figura 4.3 muestra las características seleccionadas y su importancia. Esto último es obtenido implementando el bosque aleatorio para la predicción del robo a casas únicamente con estas características.

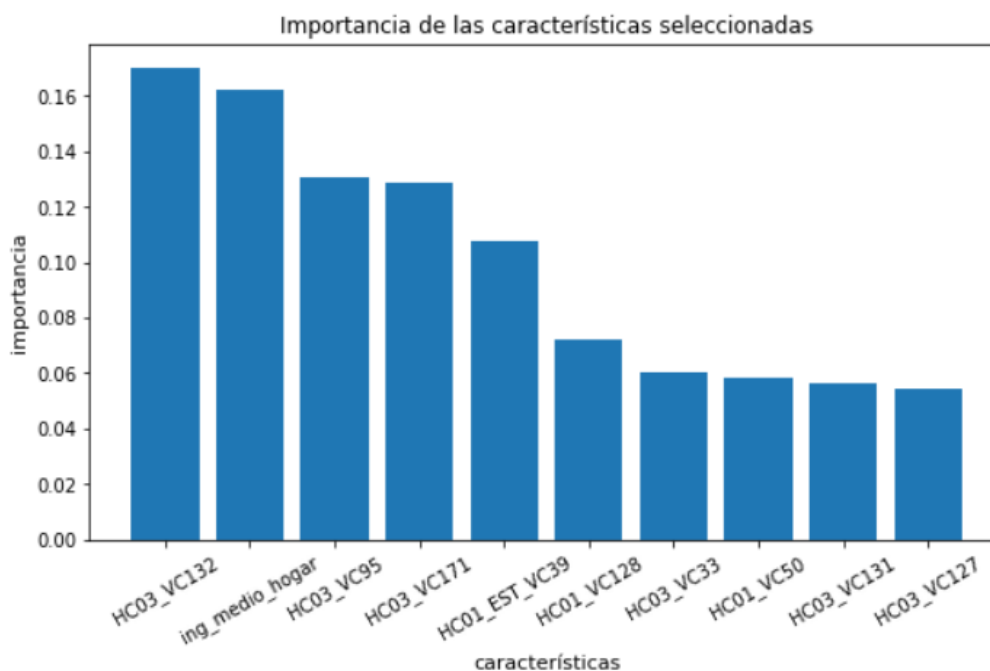


Figura 4.3: Importancia de las 10 características más relevantes según RFE para la predicción de robo. Fuente: autor.

Las características seleccionadas siguiendo el orden decreciente de importancia mostrado en la figura 4.3 son: porcentaje de personas con seguro de salud privado, ingreso medio de hogares, porcentaje de personas con secundaria completa o superior, porcentaje de familias y personas cuyo ingreso en los últimos 12 meses estuvo por debajo del nivel de

pobreza, porcentaje de población que nunca se ha casado, valor promedio de viviendas que están ocupadas por sus dueños, porcentaje de personas que trabaja desde su hogar, cantidad promedio de habitaciones por vivienda, porcentaje de personas con seguro de salud y porcentaje de viviendas con al menos un computador. La mayor parte de estas características son del tipo económicas (50%), cuatro de diez están relacionadas a las viviendas (40%) y solo está asociada a la población total y nativa.

4.2 Esquema general del procedimiento para la obtención de la probabilidad de robo a casas

Según lo expuesto en los capítulos 3 y 4, el procedimiento abordado en esta memoria para la obtención de probabilidades de robo a casas se puede observar en la figura 4.4.

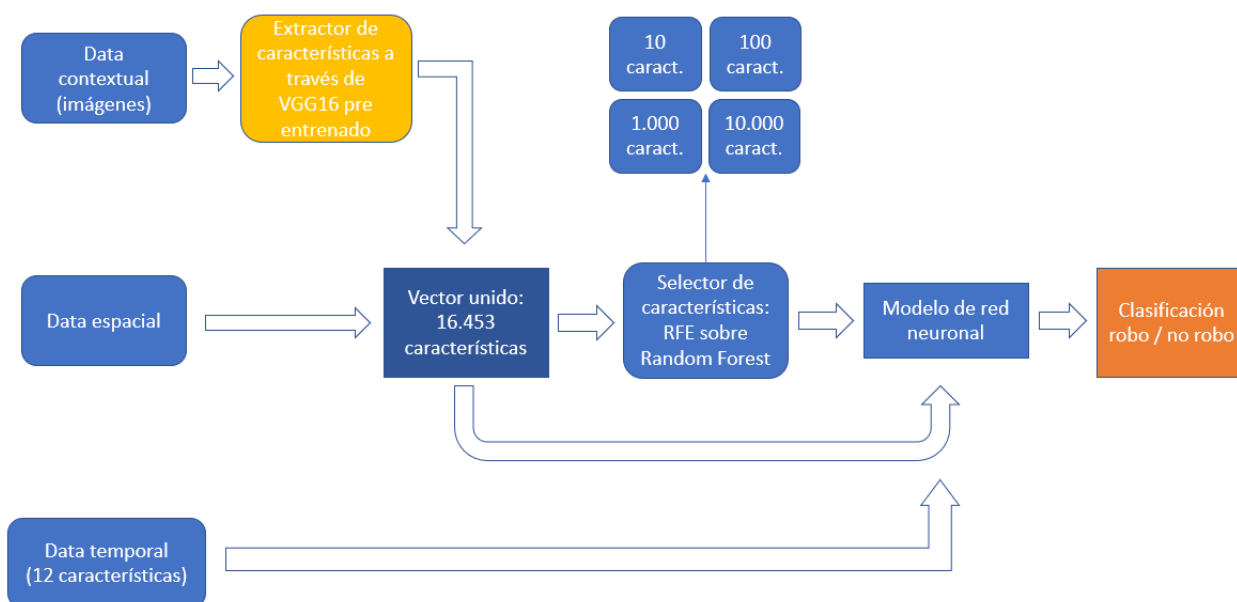


Figura 4.4: esquema general del procedimiento para la obtención de la probabilidad de robo a casas.

5. Análisis de resultados

En este capítulo se presentan los detalles y resultados del modelo de red neuronal final en la obtención de la clasificación de robo a casas. En primer lugar, se expone sobre el ajuste de los hiperparámetros de la red neuronal, lo cual tiene por objetivo lograr una mejor predicción. Luego, se presentan los resultados obtenidos en los distintos conjuntos de características reducidas y el conjunto de características completo, así como las técnicas usadas para obtener resultados más robustos. Los resultados son presentados primero para el conjunto de validación y después para el conjunto de prueba. Finalmente, se discute acerca de estos.

5.1 Ajuste de hiperparámetros de la red neuronal

Los hiperparámetros en una red neuronal son las variables que determinan la estructura de la red (número de capas y de neuronas por capa) y las que establecen cómo se entrena la red, como la tasa de aprendizaje o el optimizador a utilizar [94]. El hecho de encontrar los hiperparámetros adecuados para la red es muy relevante, puesto que determinan el rendimiento de esta en cuanto a métricas y tiempo de ejecución. Dado esto, antes de obtener resultados definitivos se procede al ajuste de los hiperparámetros.

El ajuste es realizado para el escenario en el cual se tienen las 1.000 características más importantes de acuerdo con los resultados del RFE detallados en la sección 4. Esto se debe a que, al tener que realizar una gran cantidad de iteraciones del modelo para encontrar los mejores hiperparámetros, el tiempo de ejecución de la red es importante, y este es significativamente mayor cuando se tiene la totalidad de características como se observa más adelante en la sección 5.2.1. Además, es importante señalar que este ajuste se realiza de acuerdo con las métricas obtenidas en el conjunto de validación (ver tabla 4.1) dado que al hacer esto, el conjunto de prueba queda excluido y, por tanto, el riesgo de seleccionar una configuración que se ajuste en exceso a este conjunto se reduce. De esta forma, los resultados en el conjunto de prueba son más robustos.

La gran cantidad de hiperparámetros en una red neuronal y el tiempo de entrenamiento del modelo complica el uso de una grilla que establezca distintas combinaciones como se hizo para el bosque aleatorio. Es por esto, que el ajuste de hiperparámetros se hace por medio de múltiples grillas, las cuales son modeladas a través de *for loops* en Python. Los diferentes hiperparámetros son agrupados de forma de hacer más eficiente el proceso, lo cual difiere de lo realizado anteriormente en el bosque aleatorio, ya que en este último la mayoría de los hiperparámetros se combinaban en una misma grilla, teniendo una mayor cantidad de combinaciones.

La tabla 5.1 muestra los hiperparámetros ajustados para el modelo presentado en la figura 3.1 (se sugiere al lector que se apoye en dicha figura). Además, la tabla expone los rangos probados para cada hiperparámetro y la configuración escogida junto a sus resultados en cuanto a “loss” y “accuracy” en el conjunto de validación. La tabla está ordenada de arriba hacia abajo de acuerdo con el orden en que se ajustaron los hiperparámetros, y cada fila representa aquellos que se ajustaron en una misma grilla.

Hiperparámetros	Rangos probados	Configuración escogida y resultados
Cantidad de capas ocultas	1 y 2 por cada entrada y para después de la capa de concatenación	1 capa después de cada entrada y 2 después de la capa de concatenación.
Neuronas por capa:		
- Bidirec-LSTM	- 5 a 15	- 12
- Densa 1 (Espacial)	- 24 a 64	- 56
- Densa 2 (Contextual)	- 12 a 48	- 16
- Densa 3	- 24 a 50	- 48
- Densa 4	- 10 a 36	- 12
		loss: 0,51, acc: 0,73
Tasa de aprendizaje	1e-1 a 1e-5	1e-2
Decaimiento	1e-7 a 1e-2	1e-5
		loss: 0,51, acc: 0,73
Optimizador	Los disponibles en la documentación de Keras	'Adam'
		loss: 0,51 acc: 0,72
Tamaño de lote	50 a 700	300
Número de épocas	15 a 50	35
		loss: 0,51, acc: 0,73
Inicializadores	Los disponibles en la documentación de Keras	'glorot_uniform'
		loss: 0,51, acc: 0,73
Función de activación (exceptuando capa final)	Las disponibles en la documentación de Keras	'relu'
		loss: 0,51, acc: 0,73
Regularizadores:		
- Nivel de dropout	- 0,0 a 0,4	- 0,2
- Kernel regularizer	- 1e-2 a 1e-6	- 1e-6
- Recurrent regularizer (capa LSTM)	- 1e-4 a 1e-6	- 1e-5
- Kernel constraint	- 1 a 4	- 2
- Recurrent constraint (capa LSTM)	- 1 a 3	- 2
		loss: 0,48, acc: 0,73

Tabla 5.1: hiperparámetros configurados junto con el rango testado y los resultados en el conjunto de validación para el seleccionado.

Como se mencionó en la sección 3.1, la cantidad de neuronas por capa, lo cual corresponde a un hiperparámetro relacionado a la arquitectura de la red, varía con respecto a la cantidad de características seleccionadas, por lo que la configuración de esta variable, mostrada en la tabla 5.1, corresponde sólo al conjunto de 1.000 características seleccionadas. Los demás hiperparámetros se mantienen igual para los modelos asociados a los otros conjuntos de características.

A continuación, se presenta una breve explicación de los hiperparámetros:

- Cantidad de capas ocultas y de neuronas por capa: Estos hiperparámetros son variables que afectan a la arquitectura del modelo. Mientras más capas se agregan a la red, más complejas son las funciones que puede generar, sin embargo, se ha demostrado que, para la mayor parte de los problemas, no se justifica el uso de más de 1 o 2 capas ocultas. Las neuronas en cada capa, por otro lado, afectan a la cantidad de parámetros que el modelo debe estimar. En general, se busca una

cantidad de parámetros similar a la cantidad de datos de entrenamiento que se tienen.

- Tasa de aprendizaje y decaimiento: la tasa de aprendizaje controla cuánto actualizar los pesos al final de cada lote, mientras que el decaimiento indica cuánto disminuye la tasa de aprendizaje luego de la actualización de lote. Una baja tasa de aprendizaje hace que el modelo se demore más en alcanzar un óptimo, mientras que una alta tasa puede hacer que la red no converja. Ambos hiperparámetros se ajustan en la misma grilla de manera de hacer más eficiente la búsqueda de una buena solución.
- Optimizador: se refiere a la función de optimización usada para alcanzar los pesos que entregan un mejor ajuste de la red. El optimizador “Adam” (estimación del momento adaptativo) ha demostrado empíricamente buenos resultados con bajos requisitos de memoria y en general, funciona bien incluso sin un extenso ajuste de hiperparámetros [95].
- Tamaño de lote y número de épocas: el tamaño de lote corresponde al número de muestras que la red observa antes de actualizar los pesos (parámetros). Cuando el tamaño de lote aumenta, el entrenamiento de la red será más rápido, puesto que los pesos se actualizan una menor cantidad de veces, sin embargo, esto puede generar que el modelo no aprenda lo suficiente. Por otro lado, el número de épocas es el número de veces que se le muestra a la red todo el conjunto de entrenamiento. A medida que aumentan las épocas el modelo aprende más de los datos, pero llega un momento en el cual la red no aprende más y, por tanto, no vale la pena seguir el entrenamiento.
- Inicializadores: este hiperparámetro determina los pesos iniciales que tomará la red antes de la primera iteración, por lo que influye en el esfuerzo que debe realizar el modelo para encontrar una solución óptima. Si se configura una inicialización deficiente el modelo puede incluso divergir.
- Función de activación: controla la no linealidad de las neuronas individuales y determina cuándo estas se activan (ver sección 2.3.1). La función “Relu” (rectificador lineal) es una de las más usadas, debido a su bajo costo computacional y a que las combinaciones de esta función son no lineales, por lo que cualquier otra función se puede aproximar con combinaciones de esta [96].
- Regularizadores: el objetivo de estos hiperparámetros es penalizar al modelo y, de esta forma, evitar el sobreajuste en el conjunto de entrenamiento. En este caso se implementan tres técnicas de regularización. En primer lugar, se tiene la capa de dropout o abandono la cual se inserta luego de las capas densas y de la LSTM. Esta capa inactiva aleatoriamente algunas de las neuronas de la capa predecesora. La cantidad de neuronas desactivadas está determinada por el nivel o tasa de abandono.

En segundo lugar, se tienen los regularizadores del kernel (kernel regularizers), lo cuales se insertan en la configuración de cada capa. Esta técnica, penaliza las matrices de pesos de los nodos. En este caso se ocupa la regularización “L2”, la cual es conocida como “regresión de cresta”, donde los pesos son disminuidos a cero, pero nunca cero. Si el valor de penalización es muy grande, entonces el ajuste será insuficiente [97]. En el caso de la capa LSTM, se utiliza además un regularizador recurrente que se adecúa a la estructura de esta capa.

Por último, se utilizan regularizadores de restricción (constraint regularizer), lo cuales también son fijados en cada capa. Estos, permiten establecer restricciones en los pesos de red durante la optimización. En este caso se utiliza la máxima

norma, la cual restringe los valores de los pesos, de manera que estos no sean muy positivos o negativos. Mientras menor sea este hiperparámetro, mayor será la restricción a pesos grandes, lo que podría generar un ajuste insuficiente. Al igual que en el caso anterior, la capa LSTM utiliza un regularizador recurrente.

5.2 Resultados del modelo

En esta sección se presentan los resultados obtenidos en el modelo final de red neuronal expuesto en la figura 3.1 para las distintas cantidades de características seleccionadas luego del ajuste de hiperparámetros. En primer lugar, se presentan los tiempos de entrenamiento para cada conjunto de características reducidas, así como para el conjunto total de características, luego se exponen los resultados para el conjunto de validación y posteriormente para el conjunto de prueba. El modelo es entrenado, validado y testeado según lo presentado en la tabla 4.1.

El entrenamiento del modelo está configurado para 35 épocas y un tamaño de lote de 300. Esto quiere decir que se tienen 35 iteraciones del modelo y los parámetros se actualizan en cada una, cada 300 muestras.

Es importante recordar que la arquitectura cambia con respecto a la cantidad de características, por lo que se tienen 5 modelos diferentes (1 por cada escenario de características seleccionadas y para el conjunto completo de características). Esto se debe a que la cantidad de parámetros a estimar en el modelo depende de la cantidad de capas, de la cantidad de neuronas por capa y de la cantidad de características del conjunto de datos. Por tanto, el cambio en la cantidad de características es compensado por un cambio en la arquitectura del modelo, en particular respecto a la cantidad de neuronas por capa, lo cual mantiene una cantidad de parámetros estable por estimar. Además, el conjunto de 10 características seleccionadas por el RFE se compone sólo de variables espaciales, por lo cual la entrada de información contextual (imágenes), es eliminada en este caso.

Para cada uno de estos modelos, se obtiene un conjunto de parámetros o pesos correspondiente a una de las 35 épocas en la que el modelo obtiene una mejor métrica de pérdida para el conjunto de validación. Con estos pesos se obtienen los resultados en el conjunto de validación y de prueba para cada set de características.

5.2.1 Tiempos de entrenamiento

La cantidad de características es una de las variables que influye en el número de parámetros que debe predecir la red neuronal. En la figura 5.1 se observan los tiempos de entrenamiento sobre el conjunto de validación para cada conjunto de características. Como se mencionó anteriormente, el entrenamiento es realizado con un tamaño de lote de 300 muestras y una cantidad de épocas igual a 35.

Como se puede apreciar, existe una gran diferencia entre entrenar el modelo con el 100% de las características (16.453) y entrenarlo con el 60% de estas (10.000). Además, hay una diferencia de casi 300 segundos entre los conjuntos de 10.000 y 1.000 características. Para los conjuntos más chicos, de 0,6% y 0,06% del total de características, se tiene una mínima diferencia en el tiempo de ejecución, por lo que no se tiene una gran

ganancia en términos de costo computacional al disminuir de 100 a 10 características seleccionadas.

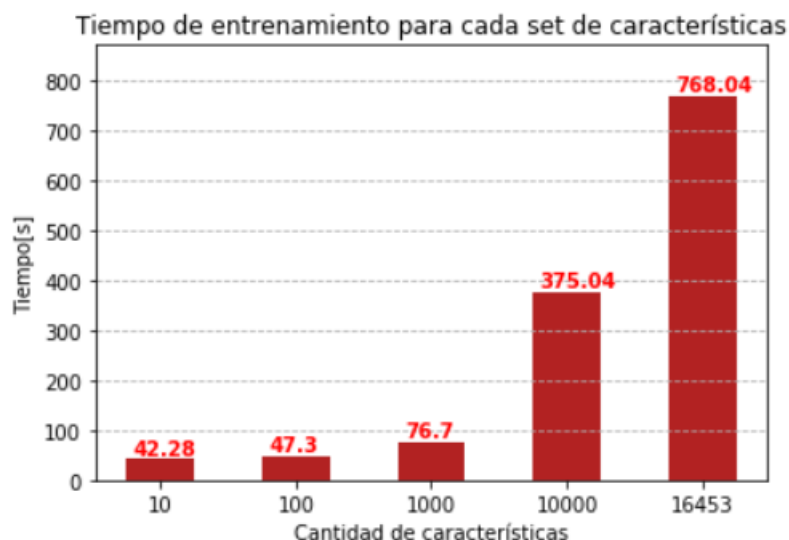


Figura 5.1: tiempo de entrenamiento en segundos sobre el conjunto de validación para los diferentes sets de características.

5.2.2 Validación

Tal como se presenta en la tabla 4.1, el conjunto de validación está compuesto por 1381 muestras que corresponden al 5% del conjunto inicial de entrenamiento. Estos datos son los robos y no robos con fecha diciembre de 2015, 2016 y 2017. Es importante señalar que se probaron 2 opciones más de conjunto de validación: todo el año 2016 (entrenamiento desde julio de 2014 a diciembre de 2015), y los últimos 3 meses del año 2017 (entrenamiento desde julio de 2014 a septiembre de 2017). Ambas presentaron resultados deficientes en comparación con el conjunto seleccionado finalmente.

Las figuras 5.2 y 5.3 muestran los resultados para el conjunto de validación con respecto a las métricas expuestas en la sección 2.4.2 para los diferentes conjuntos de características obtenidos en el RFE y para el conjunto completo de características.

Las características representan el 0,06%, 0,60%, 6,07% y 60,78% de la cantidad total de características iniciales, las cuales ascienden a un total de 16.453. Es importante que el lector recuerde que este valor corresponde al total de características espaciales y contextuales, excluyendo a las características temporales que son agregadas en todos los modelos como serie temporal.

Las métricas presentadas en las figuras 5.2 y 5.3 representan el mejor ajuste, con respecto a la función de pérdida, de cada uno de los modelos en una de las 35 iteraciones que se ejecutan. Es interesante observar que las mejores métricas para accuracy y precision, las presenta el conjunto de 100 características y luego, para las métricas de recall y f1, las mejores métricas las muestra el conjunto total de características. Esto es algo contraintuitivo, puesto que podría esperarse una mejora en las métricas a medida que aumenta la cantidad de características, ya que aumenta la cantidad de información con la

que cuenta el modelo. Sin embargo, lo anterior puede explicarse, puesto que se hizo un ajuste de hiperparámetros utilizando el conjunto de 1.000 características, por lo que es esperable que las métricas del conjunto de 1.000 características superen a las de otros conjuntos (como al de 10 y 10.000 en este caso). Por consiguiente, es posible que estos hiperparámetros se adapten mejor a ciertos conjuntos que a otros.

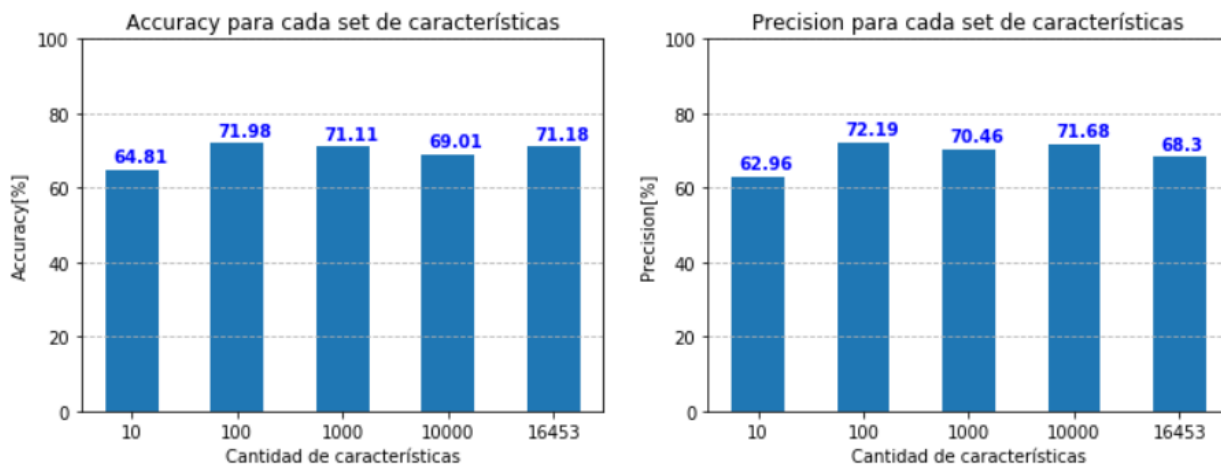


Figura 5.2: accuracy (izquierda) y precision (derecha) para cada set de características evaluadas en el conjunto de validación.

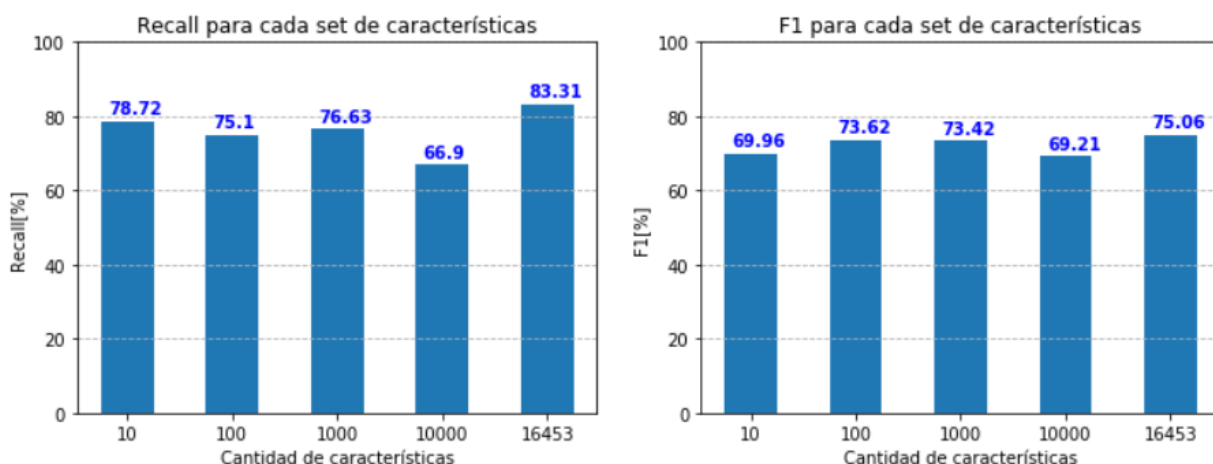


Figura 5.3: recall (izquierda) y F1 (derecha) para cada set de características evaluadas en el conjunto de validación.

Recordando lo que indica cada métrica, se tiene que, en primer lugar, accuracy indica qué tan buena es la predicción para ambas clases. Luego, precision indica qué tan buena es la predicción para las muestras de no robo, mientras que recall indica qué tan buena es la predicción para las de robo. Por último, F1 es la media armónica entre precision y recall, por lo que es una medida general con respecto a la predicción de ambas clases. Mientras que los conjuntos de 100 y 1.000 características presentan un comportamiento más o menos estable con respecto a las métricas, los conjuntos de 10, 10.000 y todas las características presentan grandes diferencias entre precision y recall. Los conjuntos de 10 y todas las características presentan, por un lado, bajo precision, pero alto recall, lo que indica que recuperan de mejor manera los casos de robo a casas, pero al hacer esto, clasifican como casas robadas algunas que realmente no lo habían sido, lo que

se traduce en un bajo precision. Caso contrario es el del conjunto de 10.000 características, el cual, recupera de mejor manera las muestras de no robo, clasificando erróneamente una mayor cantidad de casas como no robadas cuando en realidad sí fueron hurtadas.

Las diferencias mencionadas anteriormente son relevantes, porque un bajo precision implica un despliegue de fuerza policial en zonas donde no son requeridos, mientras que un bajo recall afecta directamente a las víctimas, puesto que no hay fuerza policial en lugares que sí se necesita.

Es importante recordar que las métricas obtenidas corresponden al mejor ajuste de 35 iteraciones del modelo en el conjunto de validación, donde los hiperparámetros fueron ajustados en función de los resultados en este conjunto, lo cual quiere decir que había que “conocerlo”. En la práctica, lo importante es la predicción que se haga a futuro, sobre muestras que aún no se conocen. Esta es la función que cumple el conjunto de prueba y sus resultados son presentados a continuación.

5.2.3 Test

El conjunto de prueba está compuesto por 5.595 datos de robo y no robo los cuales corresponden a las muestras del año 2018 (tabla 4.1). Se separó de esta forma, debido a que se busca testear el modelo en todos los meses del año, y así evitar sesgos temporales. Además, dado que se busca investigar la naturaleza serial de los eventos de robo y hacer una predicción a futuro, se opta por predecir el último año disponible, de manera de tener una mayor cantidad de muestras de entrenamiento.

Este conjunto es dividido en 4 periodos de tiempo diferentes: periodos de una semana, dos semanas, tres semanas y cuatro semanas. Esto, con el objetivo de comprobar si hay diferencias en cuanto a los resultados para la predicción de distintos tamaños de ventanas de tiempo. Para cada uno de los periodos de tiempo se excluye aquella que contenga la última semana de diciembre de 2018, dado que cuenta con una baja cantidad de registros en comparación con las demás semanas del año, por lo que se cree puede estar incompleta, dado que la base se descargó en las primeras semanas de enero de 2019. En la tabla 5.2, se muestra un resumen de los 4 escenarios en los cuales se obtiene una predicción.

Escenario	Ventana de tiempo	Número de conjuntos a predecir	Promedio de muestras por conjunto
1	1 semana	52	107
2	2 semanas	25	217
3	3 semanas	16	328
4	4 semanas	12	437

Tabla 5.2: resumen de escenarios en el conjunto de prueba.

En cada escenario se utiliza el promedio de las predicciones en los conjuntos para obtener finalmente 4 predicciones por cada conjunto de características seleccionadas por medio del algoritmo RFE.

En resumen, la metodología abordada para la obtención de resultados con el modelo de red neuronal por medio del conjunto de prueba es la siguiente:

1. Ajuste de la cantidad de parámetros a estimar para cada uno de los escenarios de características reducidas y para el conjunto total de atributos (ajustes en la arquitectura del modelo).
2. Para cada escenario de características seleccionadas, se obtiene el conjunto de parámetros que entregan una mejor métrica de pérdida del modelo (1 de 35 épocas) en el conjunto de validación.
3. Los parámetros obtenidos son evaluados en el conjunto de prueba, el cual se encuentra dividido en 4 diferentes ventanas de tiempo: periodos de una, dos, tres y cuatro semanas. Se obtiene la predicción promedio para cada caso.
4. Con esto, se obtienen 4 predicciones (para las 4 ventanas de tiempo), para cada uno de los 4 escenarios de características reducidas y para el conjunto de características completo (16.453).

Los resultados de los modelos en el conjunto de prueba para los escenarios expuestos en la tabla 5.2 son presentados a continuación en las tablas 5.3 a 5.6. Los números corresponden al promedio para cada métrica en cada conjunto de características. Los números de conjuntos predichos varían según el escenario. Los números mostrados se encuentran en porcentaje.

Escenario 1:

	Accuracy	Precision	Recall	F1
10	61.52	57.66	76.19	65.49
100	67.92	66.16	68.97	67.23
1000	69.17	66.04	74.16	69.72
10000	67.13	68.10	59.56	63.34
16453	67.53	62.62	81.06	70.55

Tabla 5.3: métricas de la predicción semanal en el conjunto de prueba (año 2018).

Escenario 2:

	Accuracy	Precision	Recall	F1
10	61.55	57.71	75.98	65.48
100	67.65	66.05	68.36	66.97
1000	69.07	66.00	74.11	69.75
10000	67.33	68.47	59.57	63.57
16453	67.75	62.83	81.19	70.78

Tabla 5.4: métricas de la predicción a 2 semanas en el conjunto de prueba (año 2018).

Escenario 3:

	Accuracy	Precision	Recall	F1
10	61.47	57.74	75.66	65.42
100	67.72	66.20	68.06	66.94
1000	69.06	66.03	73.78	69.67
10000	67.07	68.28	59.13	63.29
16453	67.48	62.67	80.82	70.55

Tabla 5.5: métricas de la predicción a 3 semanas en el conjunto de prueba (año 2018).

Escenario 4:

	Accuracy	Precision	Recall	F1
10	60.43	56.98	77.34	65.41
100	66.25	64.64	71.29	67.08
1000	67.67	64.84	76.42	69.67
10000	65.85	66.94	63.03	63.89
16453	66.03	61.56	82.52	70.22

Tabla 5.6: métricas de la predicción a 4 semanas en el conjunto de prueba (año 2018).

Como se puede observar en las tablas 5.3 a 5.6, los 3 primeros escenarios muestran métricas bastante similares entre sí, de hecho, no hay ninguna métrica que muestre una variación mayor a 1 punto porcentual. Sin embargo, el escenario 4 presenta algunas diferencias con respecto a los demás. El accuracy disminuye para todos los conjuntos de características, lo cual quiere decir que, en general, hay una peor clasificación del total de muestras en este escenario. Si se observan las otras métricas, precision disminuye, mientras que recall aumenta en todos los conjuntos de características. Esto quiere decir que el modelo predice una mayor cantidad de muestras como casas donde habrá robo, lo que implica un mayor recall, pero, a su vez, una mayor cantidad de casas no robadas son erróneamente clasificadas como robadas, lo que se traduce en una baja en precision.

Los resultados anteriores sugieren que una predicción para una ventana de tiempo más acotada entrega una mejor predicción en comparación a una más larga.

A pesar de las diferencias del resto de los escenarios con el cuarto, la tendencia con respecto a los resultados en los distintos conjuntos de características se mantiene. Prueba de esto, es la desviación estándar entre escenarios, que refleja la dispersión de las métricas en cada conjunto de características, es decir, las diferencias respectivas entre escenarios para cada posición en las tablas. Estas son:

- Entre 0.47 y 0.68 para accuracy
- Entre 0.32 y 0.65 para precision
- Entre 0.63 y 1.57 para recall
- Entre 0.04 y 0.24 para f1

Cada rango contempla 5 medidas (una para cada conjunto de características).

De esta forma, para una mejor observación de los resultados entre los conjuntos de características, se presenta el promedio de todos los escenarios en la tabla 5.7.

	Accuracy	Precision	Recall	F1
10	61.24	57.52	76.29	65.45
100	67.38	65.76	69.17	67.06
1000	68.74	65.73	74.62	69.70
10000	66.84	67.95	60.32	63.52
16453	67.20	62.42	81.40	70.52

Tabla 5.7: media de las predicciones entre los 4 escenarios. En negrita, se encuentra el modelo destacado.

Se destaca el modelo de 1.000 características, ya que presenta el mejor accuracy y posee la segunda mejor métrica en cuanto al f1, la cual corresponde a la medida utilizada

para comparar modelos que difieren en recall y precision. Con respecto a esta métrica, el modelo de 1.000 características sólo se encuentra por debajo del modelo que las incluye a todas, con una diferencia de menos de un punto porcentual. Sin embargo, la gran diferencia entre ambos con respecto al tiempo de entrenamiento (figura 5.1), influye en la decisión de considerar al modelo de 1.000 características como el destacado. En la práctica, es relevante tomar en consideración aspectos como el tiempo de entrenamiento al momento de seleccionar un modelo, puesto que este representa el costo computacional y, en caso de entrenar el modelo de forma recurrente, la diferencia entre utilizar el modelo de 1.000 características en contraposición al que las incluye todas, se traduce en un ahorro de recursos significativo.

Por otro lado, el comportamiento de los modelos para los conjuntos de 10 y todas las características en el conjunto de prueba se mantiene similar a lo que ocurre en el conjunto de validación (figuras 5.2 y 5.3), en el sentido de que ambos modelos presentan bajo precision, pero alto recall, es decir estos modelos no logran distinguir de gran manera las casas robadas de las no robadas, y la mayoría se clasifican como robadas. Esto, a pesar de tener como beneficio el hecho de que un porcentaje importante de las casas con un real riesgo de ser robadas contarán con el resguardo policial pertinente, tiene como deficiencia que una cantidad no menor de fuerzas policiales se desplegarán a lugares donde no deberían estar, lo que se traduce en asignación de recursos ociosos y, dada la capacidad limitada de fuerza policial existente, es probable que no se logre una cobertura total de las casas predichas como robo.

A diferencia de lo ocurrido en el conjunto de validación, en el conjunto de prueba se puede apreciar una leve tendencia del accuracy a mejorar a medida que la cantidad de características es mayor (excepto en el conjunto de 1.000 características que, presenta mejores resultados que en todos los demás conjuntos). Este resultado es esperable, puesto que el aumento en la cantidad de características es sinónimo de un aumento en la cantidad de información y, por ende, se espera un mejor rendimiento del modelo.

Como se mencionó anteriormente, una de las principales razones de este resultado, es decir, de una obtención de mejores métricas para el conjunto de 1.000 características, se puede explicar por el hecho de haber ajustado los hiperparámetros en este conjunto. La métrica más baja en este modelo (así como en la mayoría), es precision. Según lo explicado en la sección 2.4, esta métrica indica qué tan bien se está clasificando la clase negativa, en este caso, los registros de no robo. Este resultado se puede deber a la manera artificial que se usó para la confección de la base de no robos, en donde no se tiene la real proporción de robo y no robo, sino que se usó la generación de puntos aleatorios para tener las clases balanceadas. Esto podría explicar el hecho de que el modelo no logre capturar las principales características de las casas no robadas.

A continuación, se presenta el comportamiento del modelo seleccionado (1.000 características), a través de las matrices de confusión correspondientes a los distintos escenarios testeados (figuras 5.4 a 5.7). Esto, con el objetivo de hacer más visual los resultados. Los números en los cuadrantes corresponden al promedio de las predicciones hechas por escenario.

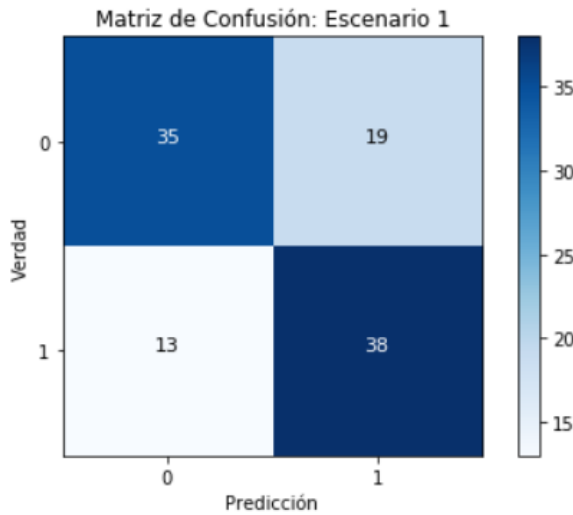


Figura 5.4: Matriz de confusión para el escenario 1 – 1.000 características

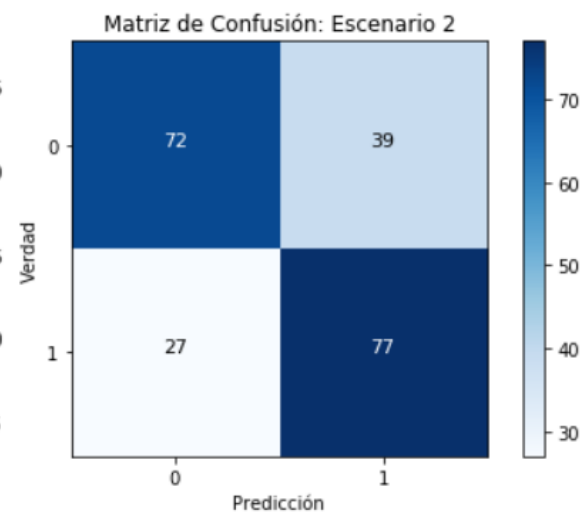


Figura 5.5: Matriz de confusión para el escenario 2 – 1.000 características

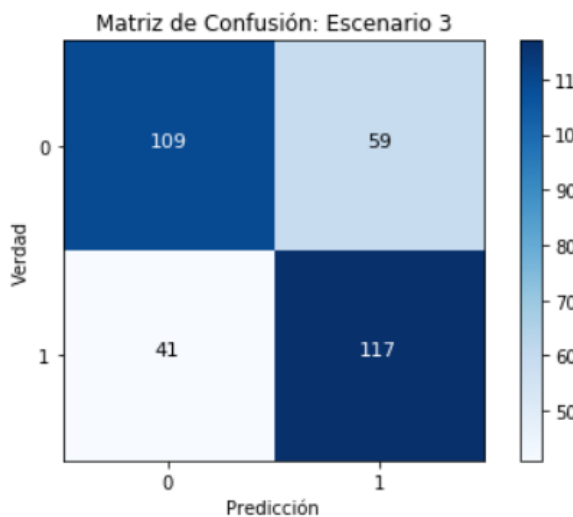


Figura 5.6: Matriz de confusión para el escenario 3 – 1.000 características

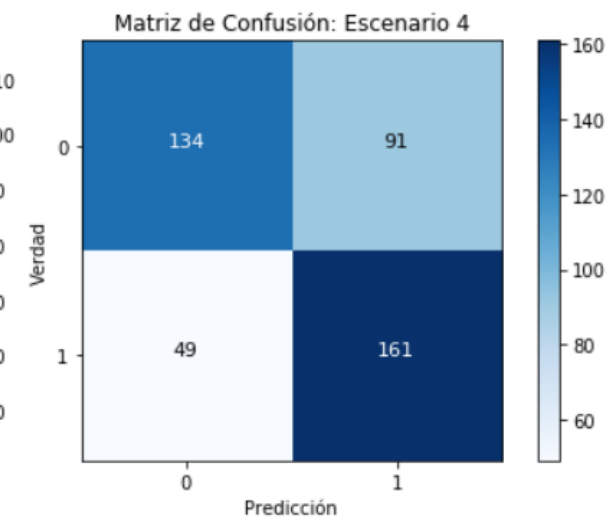


Figura 5.7: Matriz de confusión para el escenario 4 – 1.000 características

Como se puede observar, en todos los escenarios el modelo consigue una mejor predicción con respecto a las casas robadas, siendo los falsos negativos (predecir cero cuando en realidad era uno) el número más bajo en todas las figuras. Si lo llevamos a la práctica, esto quiere decir que habrá una menor cantidad de casas desprotegidas, puesto que las casas predichas como no robo, cuando en realidad si habían sido robadas, se encuentran en menor cantidad. Por otro lado, habrá una mayor cantidad de recursos asignados erróneamente, es decir, zonas en las cuales se asigna personal policial, y sin embargo no ocurre un robo, lo cual se aprecia a través de la alta cantidad de falsos positivos (predecir uno cuando en realidad era cero) y, por ende, una menor cantidad de casas predichas correctamente como no robadas.

Prácticamente todos los escenarios tienen estadísticas similares (lo cual se puede apreciar a través de los colores de los cuadrantes), excepto por la predicción a 4 semanas (escenario 4) como se mencionó anteriormente, donde se observa una mayor cantidad de

casas no robadas predichas como robadas en comparación a los demás escenarios. Este es el único indicio de que la predicción podría verse afectada según la ventana de tiempo que se considere.

5.2.3.1 Comparación: Red Neuronal vs SVM en el conjunto de 1.000 características

A continuación, se presenta una comparación entre los resultados del modelo de red neuronal propuesto y un modelo clásico de aprendizaje automático: SVM. Este modelo fue presentado en la sección 2.2.2 y es de los algoritmos más populares en el desarrollo de modelos predictivos. La comparación se realiza en el conjunto de prueba para el escenario 1, dada la poca variación entre escenarios. Además, se selecciona el conjunto de 1.000 características, dado que es aquel que presenta mejores resultados en un tiempo de entrenamiento relativamente bajo (figura 5.1). La tabla 5.8 muestra esta comparación a través del promedio de las métricas más relevantes para el escenario nombrado. La figura 5.8 hace más explícita la comparación a través de un gráfico de barras de estos promedios. Por otro lado, la tabla 5.9 muestra la desviación estándar (std) de cada métrica para ambos modelos. Es importante recordar que el escenario 1 consiste en la división del conjunto de prueba (año 2018) en semanas, por lo que se tiene un total de 52 predicciones correspondientes a las 52 semanas del año 2018. Esto también se visualiza de mejor manera a través de las figuras 5.9 y 5.10, donde se tienen los gráficos de bigote y caja para las métricas de cada modelo.

	<i>Red Neuronal</i>	<i>SVM</i>
<i>Accuracy</i>	69.17	71.17
<i>Precision</i>	66.04	69.38
<i>Recall</i>	74.16	71.90
<i>F1</i>	69.72	70.40

Tabla 5.8: Promedio de métricas para los modelos de red neuronal y SVM.

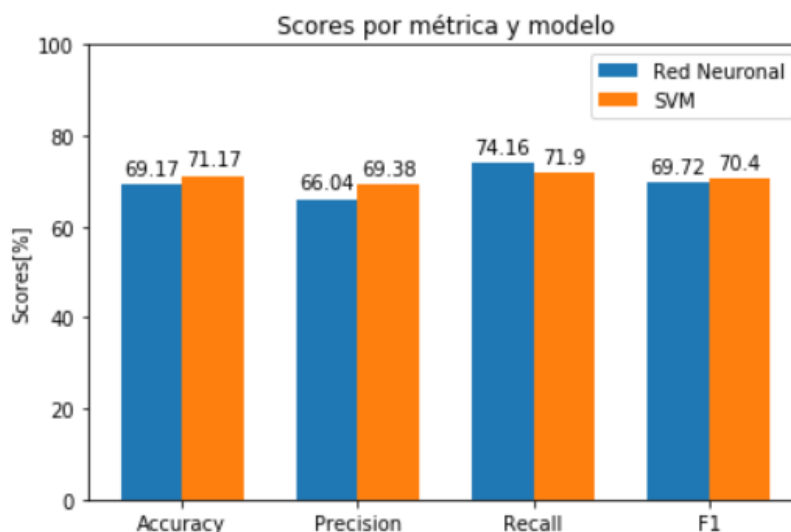


Figura 5.8: Comparativa de métricas para los modelos de red neuronal y SVM.

	<i>Red Neuronal</i>	<i>SVM</i>
<i>Std accuracy</i>	3.99	4.28
<i>Std precision</i>	3.99	4.93
<i>Std recall</i>	6.7	7.92
<i>Std F1</i>	4.27	5.33

Tabla 5.9: Desviación estándar de métricas para los modelos de red neuronal y SVM.

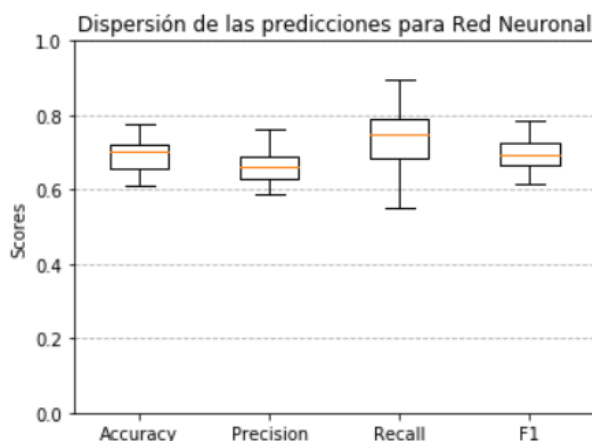


Figura 5.9: Box plot para las predicciones a 1 de la red neuronal

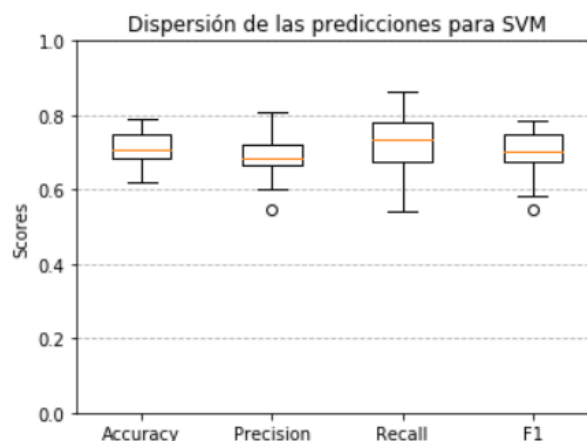


Figura 5.10: Box plot para las predicciones a 1 semana de SVM

A través de la tabla 5.8 y de la figura 5.8 se observa que, para la mayor parte de las métricas, el modelo de SVM supera a la red neuronal, excepto por la métrica de recall. Esta métrica se refiere a qué tan bien logra el modelo predecir la clase positiva, es decir, las casas robadas, mientras que precision es lo contrario, es decir, qué tan bien el modelo predice las zonas no robadas. Dado esto, en la práctica, el modelo de SVM lleva a una asignación menor de recursos ociosos, es decir, se tiene menos errores en la colocación de fuerza policial en zonas donde no es necesario tenerla. Sin embargo, el uso de la red neuronal conlleva a una mayor cobertura de fuerza policial en lugares donde sí es necesario tenerla, puesto que logra predecir la clase positiva de mejor manera. Ahora bien, el F1 es una métrica que es útil para comparar modelos en donde se tengan diferencias entre recall y precision, como es el caso presentado. Esta métrica es mayor en SVM que en la red neuronal, con una diferencia de casi 0,7 puntos porcentuales entre ambos modelos. A pesar de esto, es importante señalar que el tiempo de entrenamiento del SVM para el conjunto de 1.000 características seleccionadas corresponde 2.555 segundos, es decir, casi 43 minutos de entrenamiento, lo cual es significativamente mayor al minuto con 30 segundos que demora el entrenamiento en la red neuronal.

En la tabla 5.9 se observa que la desviación estándar es mayor en el modelo SVM en cada una de las métricas, lo cual indica que hay una mayor variación entre las predicciones semanales para dicho modelo. Esto se aprecia mejor en las figuras 5.9 y 5.10, donde se observa la dispersión de las métricas a través de gráficos de bigote y caja (box plot) para las predicciones de las 52 semanas del año 2018 para la red neuronal y SVM, respectivamente. Estos gráficos muestran cómo están distribuidos los datos y qué tan dispersos se encuentran. La primera línea indica el extremo inferior, es decir la predicción más baja, luego el rango entre esta línea y la línea inferior de la caja representa el rango de valores del cuartil inferior. La línea naranja indica la mediana, es decir, el valor que

define que hacia abajo se encuentra la mitad de los datos y hacia arriba la otra mitad. Luego, desde la línea superior de la caja hasta el extremo superior se tienen los valores que forman parte del cuartil superior. Los datos atípicos, están representados por puntos abajo o arriba de los extremos inferior y superior respectivamente.

En primer lugar, se puede observar que las medidas de accuracy son similares con respecto al rango en donde se mueven los datos (entre 0.6 y 0.8). Sin embargo, el cuartil superior está desplazado hacia arriba en el modelo SVM en comparación a la red neuronal, por lo que, la media resultante será mayor en este modelo como se observa en la figura 5.8. El recall, por otro lado, presenta la dispersión más grande en ambos modelos, por lo que se tiene una mayor variación de esta métrica dentro de las semanas de 2018, lo que sugiere información menos estable respecto a esta métrica. Por último, tanto las métricas de f_1 como de precision presentan una mayor dispersión para el modelo SVM, incluso se tienen valores atípicos en este modelo. Esto indica que, en lo que respecta a estas métricas, la red neuronal es más estable y, por tanto, más confiable.

5.2.4 Características seleccionadas

En la sección anterior se presentaron los resultados del modelo de red neuronal, además de la comparativa con el modelo SVM. Para esto se utilizaron los distintos conjuntos de características seleccionadas descritos en la sección 4.

Es importante recordar que en todos los conjuntos se tienen características contextuales y espaciales, excepto en el conjunto de 10 características donde, a través del bosque aleatorio, se seleccionan solo espaciales como las más importantes para la predicción de robo. Sin embargo, a través de los resultados expuestos, es relevante mencionar que este conjunto presenta las métricas más bajas, lo cual enfatiza el aporte de las características contextuales al modelo. Esto corrobora teorías como [68] y respalda los resultados de estudios como [66], [14] y [29] donde prueban la importancia de las características contextuales en la predicción del crimen.

Aun así, las características espaciales resultan ser las más relevantes en el modelo, dado que solo con 10 de estas características (las más importantes según el bosque aleatorio) se puede obtener un resultado cercano al 60% de accuracy. Estas características son en su mayoría indicadores económicos, como el *porcentaje de personas con seguro de salud privado*, *ingreso medio de hogares*, *porcentaje de familias y personas cuyo ingreso en los últimos 12 meses estuvo por debajo del nivel de pobreza* y *porcentaje de viviendas con al menos un computador*, lo cual indica que en la ciudad de Dallas, un factor relevante en la elección por parte de delincuentes con respecto a qué casas robar es la riqueza de estas y el bienestar económico del barrio en el que se encuentran. Por otro lado, dentro de estas 10 características seleccionadas se tienen dos que son indicadores de la forma de vida de las personas dentro de un hogar. Estos son: *porcentaje de población que nunca se ha casado* y *porcentaje de personas que trabaja desde su hogar*, lo cual podría señalar si una casa pasa más o menos tiempo sin personas en su interior, siendo potencial víctima de robo dado el comportamiento racional de los infractores como se señala en [98].

6. Conclusiones y trabajo futuro

En la presente memoria se ha expuesto el desarrollo del diseño y construcción de un modelo basado en redes neuronales que permita la predicción del delito de robo a casas. Este modelo fue desarrollado con los datos de la policía de Dallas, los cuales se encuentran de manera pública en la web. A pesar de esto, el trabajo está desarrollado de tal forma que el modelo podría hacer uso de datos de otras ciudades o locaciones para obtener una predicción del robo de casas en ese lugar.

Esta predicción tiene el objetivo de aportar información relevante para la toma de decisiones de organismos públicos, como fuerzas policiales locales. Por otro lado, el trabajo busca seguir aportando a la literatura en el uso de aprendizaje profundo en el campo del análisis criminal.

El trabajo es presentado en capítulos que engloban los 4 aspectos más relevantes de este según los objetivos propuestos en la memoria: construcción de las bases de datos necesarias, selección de características, implementación de un modelo basado en redes neuronales y análisis de resultados y comparativas entre modelos a través de métricas. Con respecto a esto, se puede concluir que:

1. Se construye satisfactoriamente una aplicación para la clasificación de casas, de manera de obtener la base de datos de casas no robadas. Se clasifica un total de 46.292 direcciones obtenidas aleatoriamente a través de un programa en python. De esta cantidad, 21.390 direcciones son clasificadas como casas.
2. Las bases de datos han podido ser construidas de manera automática a través de programas en python y el uso de Apis, lo cual significa un gran ahorro de tiempo en comparación a una construcción manual. Se utilizó información cruzada de: características socioeconómicas y demográficas obtenidas a través de la Encuesta a la Comunidad Americana 2017 (ACS); información de puntos de interés cercanos a cada registro por medio de la Api de Google Places; información contextual a través de imágenes de Google Street View; e información temporal por medio del organismo público de Estados Unidos NOAA. Luego de la limpieza de las bases, se cuenta con 16.234 registros de casas robadas y 16966 registros de casas no robadas (48,90% y 51,10% respectivamente).
3. Se realiza una selección de características a través del algoritmo Recursive Feature Selection, donde se obtienen conjuntos de 10, 100, 1.000 y 10.000 características, en donde las características demográficas y socioeconómicas muestran ser las más importantes. Destacan aspectos referentes a la riqueza de la casa y la posibilidad de que el hogar se encuentre desocupado. Utilizando el 6,07% de las características (conjunto de 1.000 variables) se puede obtener un accuracy de casi 70%.
4. Se obtiene satisfactoriamente un clasificador de robo a casas a través de las probabilidades de ocurrencia de estos eventos. El modelo que permite esto, es entrenado con los eventos de robo y no robo ocurridos entre junio de 2014 y diciembre de 2017, siendo testeado en los datos del año 2018. Este conjunto de prueba es separado en ventanas de tiempo de una, dos, tres y cuatro semanas. Los resultados señalan que para los primeros tres escenarios no existen diferencias importantes, sin embargo, en la predicción a 4 semanas, las métricas disminuyen. Esto sugiere que si la ventana de tiempo a predecir es muy grande los resultados serán peores en comparación a una ventana de tiempo más pequeña.

5. Se hace una comparativa entre el modelo de red neuronal y el modelo de aprendizaje automático SVM, utilizando el conjunto de 1.000 características. Los resultados indican que SVM obtiene mejores métricas respecto a accuracy, precision y f1, sin embargo, para todas las métricas, incluyendo recall donde la red neuronal presenta mejores resultados que SVM, la desviación estándar de SVM es mayor en comparación al modelo de red neuronal, lo que indica que los resultados presentan una mayor variación entre semanas, por lo que la predicción es menos confiable. Además, los tiempos de entrenamiento para ambos modelos difieren en casi 42 minutos (1 minuto y 17 segundos para la red neuronal, y 42 minutos y 36 segundos para SVM). Esto demuestra que en la práctica el uso de la red neuronal es más eficiente que el modelo SVM.

De esta forma, el objetivo general de esta memoria es cumplido satisfactoriamente, dado que se obtuvo una predicción de robo a casas a través de las fuentes de información seleccionadas, identificándose los principales factores que influyen en la decisión de los delincuentes (aspectos económicos y de vivienda).

A la luz de estos resultados, se señalan algunas extensiones que se podrían aplicar sobre la base de lo implementado:

- Los resultados de la clasificación de casas a través de la aplicación web creada, podrían ser de utilidad para la construcción de una base de datos de las direcciones de todas las casas en la ciudad de Dallas. Esto con el objetivo de capturar la real proporción de casas robadas y no robadas y así obtener un modelo más robusto.
- Se utilizó el modelo VGG16 para la extracción de características de las imágenes. Se propone utilizar otras arquitecturas presentes en la literatura con el objetivo de investigar las diferencias entre distintos modelos.
- El modelo de red neuronal fue optimizado por medio del ajuste de hiperparámetros, sin embargo, esto se hizo sólo para el conjunto de 1.000 características. En un trabajo futuro se podría realizar este ajuste para todos los conjuntos de características y observar el nuevo desempeño.
- Dado que los resultados en distintas ventanas de tiempo muestran un pequeño indicio en la variación de las métricas, se podría extender el trabajo a más ventanas de tiempo, de manera de estudiar este aspecto en mayor profundidad.
- Las redes neuronales son flexibles y hay muchos parámetros que pueden influir en una predicción, se sugiere profundizar estos aspectos, alterando tipos y cantidades de capas en conjunto con los demás hiperparámetros, con el objetivo de obtener mejores métricas.
- El modelo está hecho para obtener una predicción semanal del robo a casas. Se propone una implementación diaria o para cada hora, de forma de capturar el detalle en cuanto a la influencia del factor temporal en este tipo de delito.
- Se propone la implementación de este modelo en otros lugares o ciudades, y así observar las conductas delictivas según la locación.
- Finalmente, se propone la construcción de algún tipo de plataforma que permita una predicción automática en tiempo real en cuanto a las tendencias de robo a casas en los distintos sectores de una ciudad, la cual permita la visualización e interpretación rápida para la eficiente movilización de las fuerzas policiales.

Bibliografía

- [1] Knoema, «Burglary rate,» 2016. [En línea]. Available: <https://knoema.com/atlas/ranks/Burglary-rate>. [Último acceso: 25 11 2019].
- [2] R. Acevedo, «Primer software capaz de predecir dónde ocurrirán delitos en Chile,» *Beauchef Magazine*, 2018.
- [3] R. Wright, R. H. Logie y S. H. Decker, «Criminal Expertise and Offender Decision Making: An Experimental Study of the Target Selection Process in Residential Burglary,» *Journal of Research in Crime and Delinquency*, vol. 32, n° 1, pp. 39-53, 1995.
- [4] C. Nee y A. Meenaghan, «Expert Decision Making in Burglars,» *The British Journal of Criminology*, vol. 46, n° 5, pp. 935-949, 2006.
- [5] W. Bernasco y P. Nieuwbeerta, «How do residential burglars select target areas?: A new approach to the analysis of criminal location choice,» *British Journal of Criminology*, vol. 45, n° 3, pp. 296-315, 2005.
- [6] E. J. Palmer, A. Holmes y C. R. Hollin, «Investigating Burglars' Decisions: Factors Influencing Target Choice, Method of Entry, Reasons for Offending, Repeat Victimization of a Property and Victim Awareness,» *Security Journal*, vol. 15, n° 1, pp. 7-18, 2002.
- [7] Redacción Gestión, «¿Cuáles son las consecuencias psicológicas en víctimas de robo a viviendas?,» 5 marzo 2016. [En línea]. Available: <https://gestion.pe/tendencias/son-consecuencias-psicologicas-victimas-robo-viviendas-145384-noticia/>. [Último acceso: 26 agosto 2019].
- [8] A. Sharma, «Diferencias entre aprendizaje automático y aprendizaje profundo,» 15 noviembre 2018. [En línea]. Available: <https://www.datacamp.com/community/tutorials/machine-deep-learning>. [Último acceso: 26 agosto 2019].
- [9] FBI, «2017 Crime Statistics Released,» septiembre 24 2018. [En línea]. Available: <https://www.fbi.gov/news/stories/2017-crime-statistics-released-092418>. [Último acceso: 9 septiembre 2019].
- [10] J. Wu, S. Guo, J. Li y D. Zeng, «Big Data Meet Green Challenges: Big Data Toward Green Applications,» *IEEE Systems Journal*, vol. 10, n° 3, 2016.
- [11] I. Wladawsky-Berger, «Lo que el aprendizaje automático puede y no puede hacer,» *The Wall Street Journal*, 27 julio 2018.
- [12] R. Iriondo, «Aprendizaje automático e IA, ¿cuáles son las diferencias importantes entre los dos?,» 15 octubre 2018. [En línea]. Available: <https://medium.com/datadriveninvestor/differences-between-ai-and-machine-learning-and-why-it-matters-1255b182fc6>. [Último acceso: 26 agosto 2019].
- [13] S. Arthur L., «Eight-move opening utilizing generalization learning,» *IBM Journal*, 1959.
- [14] P. Cabargas, «Predicción de robo a vehículos basado en redes neuronales alimentadas por datos espacio temporales e imágenes de google street view,» Santiago, 2019.
- [15] A. Smola y S. Vishwanathan, «Introduction,» de *Introduction of Machine Learning*, New York, Cambridge University Press, 2008, pp. 3-12.
- [16] N. Bambrick, «Máquinas de vectores de soporte: una explicación simple,» julio 2016. [En línea]. Available: <https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html>. [Último acceso: 26 agosto 2019].
- [17] L. Chen, «Support Vector Machine — Simply Explained,» 7 enero 2019. [En línea]. Available: <https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496>. [Último acceso: 26 agosto 2019].
- [18] W. Koehrsen, «Random Forest Simple Explanation,» 27 diciembre 2017. [En línea]. Available: <https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d>. [Último acceso: 26 agosto 2019].

- [19] A. Badkar, «Diferencia entre aprendizaje automático y aprendizaje profundo,» noviembre 2017. [En línea]. Available: <http://www.iamwire.com/2017/11/difference-between-machine-learning-and-deep-learning/169100>. [Último acceso: 26 agosto 2019].
- [20] K. Willems, «Tutorial de Keras: Aprendizaje profundo en Python,» 4 febrero 2019. [En línea]. Available: <https://www.datacamp.com/community/tutorials/deep-learning-python>. [Último acceso: 26 agosto 2019].
- [21] J. Roell, «De la ficción a la realidad: una guía para principiantes sobre redes neuronales artificiales,» 12 junio 2017. [En línea]. Available: <https://towardsdatascience.com/from-fiction-to-reality-a-beginners-guide-to-artificial-neural-networks-d0411777571b>. [Último acceso: 26 agosto 2019].
- [22] F. Bre, J. M. Gimenez y V. D. Fachinotti, «Prediction of wind pressure coefficients on building surfaces using artificial neural networks,» *Energy and Buildings*, vol. 158, pp. 1429-1441, 2018.
- [23] Mishra y Mayank, «Convolutional Neural Networks, Explained,» 7 marzo 2019. [En línea]. Available: <https://www.datascience.com/blog/convolutional-neural-network>. [Último acceso: 27 agosto 2019].
- [24] Apple Inc., «Performing Convolution Operations,» 13 septiembre 2016. [En línea]. Available: <https://developer.apple.com/library/archive/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html>. [Último acceso: 27 agosto 2019].
- [25] A. Sharma, «¿Qué es la agrupación máxima en redes neuronales convolucionales?,» 9 noviembre 2018. [En línea]. Available: <https://www.quora.com/What-is-max-pooling-in-convolutional-neural-networks>. [Último acceso: 27 agosto 2019].
- [26] S. Duro, «Convolutional Neural Network with TensorFlow implementation,» 17 junio 2017. [En línea]. Available: <https://medium.com/data-science-group-iitr/building-a-convolutional-neural-network-in-python-with-tensorflow-d251c3ca8117>. [Último acceso: 27 agosto 2019].
- [27] Machine Learning Glossary, «Top-5 error rate,» 24 Diciembre 2017. [En línea]. Available: <https://machinelearning.wtf/terms/top-5-error-rate/>. [Último acceso: 20 Diciembre 2019].
- [28] Neurohive, «VGG16 – Convolutional Network for Classification and Detection,» 20 noviembre 2018. [En línea]. Available: <https://neurohive.io/en/popular-networks/vgg16/>. [Último acceso: 27 agosto 2019].
- [29] H.-W. Kang y H.-B. Kang, «Prediction of crime occurrence from multi-modal data using deep learning,» 24 abril 2017. [En línea]. Available: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0176244>.
- [30] F.-D. Cioloboc, «Why use a pre-trained model rather than creating your own?,» 4 enero 2019. [En línea]. Available: <https://medium.com/udacity-pytorch-challengers/why-use-a-pre-trained-model-rather-than-creating-your-own-d0e3a17e202f>. [Último acceso: 4 agosto 2019].
- [31] S. Banerjee, «An Introduction to Recurrent Neural Networks,» 23 mayo 2018. [En línea]. Available: <https://medium.com/explore-artificial-intelligence/an-introduction-to-recurrent-neural-networks-72c97bf0912>. [Último acceso: 27 agosto 2019].
- [32] C. Nicholson, «A Beginner's Guide to LSTMs and Recurrent Neural Networks,» [En línea]. Available: <https://skymind.ai/wiki/lstm>. [Último acceso: 27 agosto 2019].
- [33] K. Hara, D. Saito y H. Shouno, «Analysis of function of rectified linear unit used in deep learning,» de *2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney, 2015.
- [34] Wang y Chi-Feng, «The Vanishing Gradient Problem,» 8 enero 2019. [En línea]. Available: <https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484>. [Último acceso: 9 septiembre 2019].
- [35] S. DE SAGAR, «Activation Functions in Neural Networks,» 6 septiembre 2017. [En línea]. Available: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>. [Último acceso: 9 septiembre 2019].

- [36] S. Narkhede, «Understanding Confusion Matrix,» 9 mayo 2018. [En línea]. Available: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>. [Último acceso: 27 agosto 2019].
- [37] Peltarion, «Binary crossentropy,» [En línea]. Available: <https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/binary-crossentropy>. [Último acceso: 9 septiembre 2019].
- [38] H. Jacob y R. Lineberry, «Governmental Responses to Crime: Executive Summary,» 1982.
- [39] D. P. Rosenbaum, «Community crime prevention: A review and synthesis of the literature,» *Justice Quarterly*, vol. 5, n° 3, pp. 323-395, 1988.
- [40] T. Hope y M. Shaw, *Communities and crime reduction*, Norwich, 1988.
- [41] G. Hughes y A. Edwards, *Crime Control and Community*, Abingdon: Willan Publishing, 2002.
- [42] W. G. Skogan y M. G. Maxfield, *Coping With Crime - Individual and Neighborhood Reactions*, Washington, 1981.
- [43] T. Hope, «Community Crime Prevention,» *Crime and Justice*, vol. 19, pp. 21-89, 1995.
- [44] D. P. Rosenbaum, A. J. Lurigio y R. C. Davis, *The prevention of crime : social and situational strategies*, Wadsworth Publishing Co, 1998.
- [45] S. F. Bennett, «Community Organizations and Crime,» *The ANNALS of the American Academy of Political and Social Science*, vol. 539, n° 1, pp. 72-84, 1995.
- [46] H. Chen, W. Chung, J. J. Xu, G. Wang, Y. Qin y M. Chau, «Crime data mining: A general framework and some examples,» *Computer*, vol. 37, n° 4, pp. 50-56, 2004.
- [47] A. A. Braga, *PROBLEM-ORIENTED POLICING AND CRIME PREVENTION*, New York: Criminal Justice Press, 2008.
- [48] T. Wang, C. Rudin, D. Wagner y R. Sevieri, «Learning to Detect Patterns of Crime,» de *Lecture Notes in Computer Science*, Cambrigde, 2013, pp. 515-530.
- [49] D. Tyagi y S. Sharma, «An Approach to Crime Data Analysis: A Systematic Review,» *International Journal of Engineering Technologies and Management Research*, vol. 5, n° 2, pp. 67-74, 2018.
- [50] J. W. Brahan, K. P. Lam, H. Chan y W. Leung, «AICAMS: Artificial intelligence crime analysis and management system,» *Knowledge-Based Systems*, vol. 11, n° 5-6, pp. 355-361, 1998.
- [51] D. E. Brown y S. Hagen, «Data association methods with applications to law enforcement,» *Decision Support Systems*, vol. 34, n° 4, pp. 369-378, 2002.
- [52] S. T. Li, S. C. Kuo y F. C. Tsai, «An intelligent decision-support model using FSOM and rule extraction for crime prevention,» *Expert Systems with Applications*, vol. 37, n° 10, pp. 7108-7119, 2010.
- [53] R. Adderley, M. Townsley y J. Bond, «Use of Data Mining Techniques to Model Crime Scene Investigator Performance,» de *Applications and Innovations in Intelligent Systems XIV*, Londres, Springer London, 2007, pp. 19-32.
- [54] H. K. Chih, A. Iriberry y G. Leroy, «Crime information extraction from police and witness narrative reports,» de *2008 IEEE International Conference on Technologies for Homeland Security, HST'08*, Claremont, 2008.
- [55] G. Saltos y M. Cocea, «An Exploration of Crime Prediction Using Data Mining on Open Data,» *International Journal of Information Technology and Decision Making*, vol. 16, n° 5, pp. 1155-1181, 2017.
- [56] M. B. Short, M. R. D'Orsogna, V. B. Pasour, G. E. Tita, P. J. Brantingham, A. L. Bertozzi y L. B. Chayes, «A statistical model of criminal behavior,» *Mathematical Models and Methods in Applied Sciences*, vol. 18, pp. 1249-1267, 2008.
- [57] M. B. Short, M. R. D'Orsogna, P. J. Brantingham y G. E. Tita, «Measuring and Modeling Repeat and Near-Repeat Burglary Effects,» *Journal of Quantitative Criminology*, vol. 25, n° 3, pp. 325-339, 2009.

- [58] M. Helbich, J. Hagenauer, M. Leitner y R. Edwards, «Exploration of unstructured narrative crime reports: An unsupervised neural network and point pattern analysis approach,» *Cartography and Geographic Information Science*, vol. 40, n° 4, pp. 326-336, 2013.
- [59] B. C. Welsh y D. P. Farrington, «Science, politics, and crime prevention: Toward a new crime policy,» *Journal of Criminal Justice*, vol. 40, n° 2, pp. 128-133, 2012.
- [60] V. Ingilevich y S. Ivanov, «Crime rate prediction in the urban environment using social factors,» *Procedia Computer Science*, vol. 136, pp. 472-478, 2018.
- [61] M. A y S. Santhosh Baboo, «An Enhanced Algorithm to Predict a Future Crime using Data Mining,» *International Journal of Computer Applications*, vol. 21, n° 1, pp. 1-6, 2011.
- [62] T. Nguyen, A. H. Sung y A. Hatua, «Building a Learning Machine Classifier with Inadequate Data for Crime Prediction,» enero 2017. [En línea]. Available: https://www.researchgate.net/profile/Andrew_Sung6/publication/320575747_Building_a_Learning_Machine_Classifier_with_Inadequate_Data_for_Crime_Prediction.
- [63] P. Cadena, «Factores determinantes de los delitos de mayor connotación social en la región metropolitana. Análisis en base a un modelo logístico,» 2016. [En línea]. Available: <http://repositorio.uchile.cl/bitstream/handle/2250/142807/Factores-determinantes-de-los-delitos-de-mayor-connotaci%C3%B3n-social-en-la-Regi%C3%B3n-Metropolitana-An%C3%A1lisis-en-base.pdf?sequence=1>.
- [64] B. Cortez, B. Carrera, Y.-J. Kim y J.-Y. Jung, «An architecture for emergency event prediction using LSTM recurrent neural networks,» 1 mayo 2018. [En línea]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417417308606>.
- [65] E. L. Less, P. McKee, T. Toomey, T. Nelson, D. Erickson, S. Xiong y R. Jones-Webb, «Matching study areas using Google Street View: A new application for an emerging technology,» *Evaluation and Program Planning*, vol. 53, pp. 72-79, 2015.
- [66] N. Bearman y K. Appleton, «Using Google Maps to collect spatial responses in a survey environment,» *Area*, vol. 44, n° 2, pp. 160-169, 2012.
- [67] J. L. van Gelder, C. Nee, M. Otte, A. Demetriou, I. van Sintemaartensdijk y J. W. van Prooijen, «Virtual Burglary: Exploring the Potential of Virtual Reality to Study Burglary in Action,» *Journal of Research in Crime and Delinquency*, vol. 54, n° 1, pp. 29-62, 2017.
- [68] J. Q. Wilson y G. L. Kelling, «Broken Windows The police and neighborhood safety,» *The Atlantic Monthly*, 1982.
- [69] T. D. Miethe y D. McDowall, «Contextual effects in models of criminal victimization,» 1993. [En línea]. Available: <https://sci-hub.tw/https://academic.oup.com/sf/article-abstract/71/3/741/2232592>.
- [70] G. C. Oatley y B. W. Ewart, «Crimes analysis software: 'Pins in maps', clustering and Bayes net prediction,» *Expert Systems with Applications*, vol. 25, n° 4, pp. 569-588, 2003.
- [71] N. Holm y E. Plynning, «Spatio-temporal prediction burglaries using convolutional LSTM neural networks,» 2018. [En línea]. Available: <http://www.diva-portal.org/smash/get/diva2:1215575/FULLTEXT01.pdf>.
- [72] Y. Hwang, S. Jung y J. Lee, «Predicting residential burglaries based on building elements and offender behavior: Study of a row house area in Seoul, Korea,» enero 2017. [En línea]. Available: <https://www.sciencedirect.com/science/article/pii/S019897151630254X>.
- [73] T. Bennett y R. Wright, *Burglars on burglary : prevention and the offender*, Aldershot, 1984.
- [74] M. Reichhoff, «The effect of weather on crime: an investigation of weather and annual crime rates,» 2017. [En línea]. Available: <https://minds.wisconsin.edu/handle/1793/78481>. [Último acceso: 14 mayo 2019].
- [75] S. Jiang, K. C. Land y J. Wang, «Social Ties, Collective Efficacy and Perceived Neighborhood Property Crime in Guangzhou, China,» *Asian Journal of Criminology*, vol. 8, n° 3, pp. 207-223, 2013.
- [76] M. B. ROBINSON, «BURGLARY REVICTIMIZATION: The Time Period of Heightened Risk,» *British Journal of Criminology*, vol. 38, n° 1, pp. 18-87, 1998.

- [77] G. Laycock, «Revictimization, reducing the heat on hot victims,» Canberra, 1999.
- [78] Wikipedia, «Cuadra (urbanismo),» 5 agosto 2019. [En línea]. Available: [https://es.wikipedia.org/wiki/Cuadra_\(urbanismo\)](https://es.wikipedia.org/wiki/Cuadra_(urbanismo)). [Último acceso: 9 septiembre 2019].
- [79] J. Pérez y A. Gardey, «DEFINICIÓN DE DEMOGRAFÍA,» 2012. [En línea]. Available: <https://definicion.de/demografia/>. [Último acceso: 9 septiembre 2019].
- [80] Michigan State University, «Finding Census Tract Data: About Census Tracts,» [En línea]. Available: <https://libguides.lib.msu.edu/tracts>. [Último acceso: 9 septiembre 2019].
- [81] Federal Communications Commission, «API Documentation for Developers,» [En línea]. Available: https://geo.fcc.gov/api/census/#!/block/get_block_find. [Último acceso: 9 septiembre 2019].
- [82] Wikipedia, «Point of Interest,» 16 julio 2019. [En línea]. Available: https://es.wikipedia.org/wiki/Punto_de_inter%C3%A9s. [Último acceso: 9 septiembre 2019].
- [83] Google Maps Platform, «Places Types,» septiembre 6 2019. [En línea]. Available: https://developers.google.com/places/supported_types. [Último acceso: 9 septiembre 2019].
- [84] Techopedia, «Contextual Data,» [En línea]. Available: <https://www.techopedia.com/definition/31778/contextual-data>. [Último acceso: 9 septiembre 2019].
- [85] Cambridge in colour, «DIGITAL IMAGE INTERPOLATION,» [En línea]. Available: <https://www.cambridgeincolour.com/tutorials/image-interpolation.htm>. [Último acceso: 9 septiembre 2019].
- [86] Wikipedia, «Web API,» 5 septiembre 2019. [En línea]. Available: https://es.wikipedia.org/wiki/Web_API#targetText=Una%20API%20es%20una%20interfaz,funciones%20de%20un%20determinado%20software.. [Último acceso: 9 septiembre 2019].
- [87] Y. Zhang, «Why is Python a language of choice for data scientists?,» 30 enero 2017. [En línea]. Available: <https://www.quora.com/Why-is-Python-a-language-of-choice-for-data-scientists#targetText=Python%20is%20powerful%20language,complex%20data%20analysis%20and%20more..> [Último acceso: 9 septiembre 2019].
- [88] T. Dettmers, «Why are GPUs well-suited to deep learning?,» 18 Junio 2018. [En línea]. Available: <https://www.quora.com/Why-are-GPUs-well-suited-to-deep-learning>. [Último acceso: 24 Diciembre 2019].
- [89] R. Shaikh, «Feature Selection Techniques in Machine Learning with Python,» 28 octubre 2018. [En línea]. Available: <https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e>. [Último acceso: 28 agosto 2019].
- [90] M. Kuhn y K. Johnson, «Recursive Feature Elimination,» de *Feature Engineering and Selection: A Practical Approach for Predictive Models*, Boca Ratón, CRC Press LLC, 2019, pp. 248-252.
- [91] P. e. al., «Scikit-learn: Machine Learning in Python,» *Journal of Machine Learning Research*, vol. 12, pp. 2825--2830, 2011.
- [92] W. Koehrsen, «Hyperparameter Tuning the Random Forest in Python,» 10 enero 2018. [En línea]. Available: <https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74>. [Último acceso: 28 agosto 2019].
- [93] J. F. Hair, C. M. Ringle y M. Sarstedt, «Corrigendum to "Editorial Partial Least Squares Structural Equation Modeling: Rigorous Applications, Better Results and Higher Acceptance",» *Long Range Planning*, vol. 47, n° 6, p. 392, 2014.
- [94] P. Radhakrishnan, «What are Hyperparameters? and How to tune the Hyperparameters in a Deep Neural Network?,» 9 agosto 2017. [En línea]. Available: <https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a>. [Último acceso: 2 septiembre 2019].

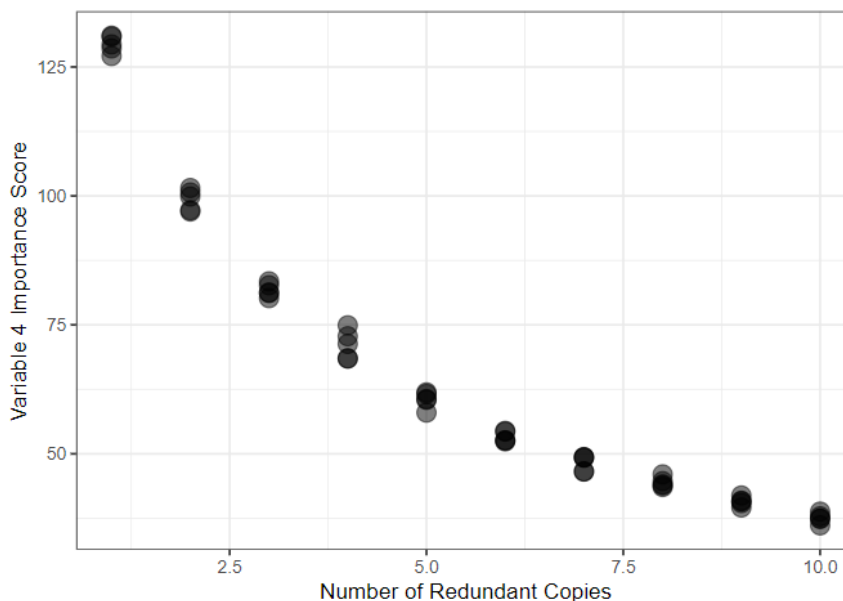
- [95] S. J. Reddi, S. Kale y S. Kumar, «ON THE CONVERGENCE OF ADAM AND BEYOND,» ICLR 2018 Conference, New York, 2018.
- [96] A. Sharma V., «Understanding Activation Functions in Neural Networks,» 30 marzo 2017. [En línea]. Available: <https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0#targetText=There%20are%20other%20variations%20too,are%20designing%20deep%20neural%20nets..> [Último acceso: 3 septiembre 2019].
- [97] A. Nagpal, «L1 and L2 Regularization Methods,» 13 octubre 2017. [En línea]. Available: <https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c#targetText=L2%20Regularization,term%20to%20the%20loss%20function..> [Último acceso: 3 septiembre 2019].
- [98] S. H. Langton y W. Steenbeek, «Residential burglary target selection: An analysis at the property-level using Google Street View,» *Applied Geography*, vol. 86, pp. 292-299, 2017.
- [99] INE, «Encuesta Nacional Urbana de Seguridad Ciudadana 2017,» Mayo 2018. [En línea]. Available: <http://www.ine.cl/docs/default-source/sociales/seguridad-ciudadana/2017/metodolog%C3%ADa/enusc-2018.pdf>.
- [100] G. Tanner, «Generating text using a Recurrent Neural Network,» 29 octubre 2018. [En línea]. Available: <https://towardsdatascience.com/generating-text-using-a-recurrent-neural-network-1c3bfee27a5e>. [Último acceso: 27 agosto 2019].
- [101] Wikipedia, «Decision tree learning,» 23 agosto 2019. [En línea]. Available: https://en.wikipedia.org/wiki/Decision_tree_learning#Advantages. [Último acceso: 28 agosto 2019].
- [102] F. Chollet, «Usage of optimizers,» 2017. [En línea]. Available: <https://keras.io/optimizers/>. [Último acceso: 3 septiembre 2019].
- [103] F. Chollet, «Usage of initializers,» 2017. [En línea]. Available: <https://keras.io/initializers/>. [Último acceso: 3 septiembre 2019].

Anexos

Anexo A: Macro categorías de clases de puntos de interés presentes en la base de Google.

- Entretenimiento: amusement_park, Aquarium, casino, art_gallery, bar, bowling_alley, movie_theater, museum, night_club, park, shopping_mall, stadium, zoo, rv_park, campground
- Salud: dentist, doctor, hospital, pharmacy, physiotherapist, health
- Restaurantes: bakery, café, meal_takeaway, restaurant, meal_delivery, food
- Tiendas: clothing_store, bicycle_store, book_store, home_goods_store, furniture_store,, jewelry_store, pet_store, shoe_store, store, electronics_store, hardware_store, liquor_store, department_store, supermarket, movie_rental, florist, laundry
- Belleza: beauty_salon, hair_care, gym, spa
- Servicios: painter, plumber, roofing_contractor, atm, electrician, bank, insurance_agency
- accounting, lawyer, locksmith, moving_company, funeral_home, storage, car_repair, veterinary_care, car_wash, finance, general_contractor.
- Ciudad: cemetery, church, courthouse, city_hall, Embassy, fire_station, hindu_temple, local_government_office, mosque, post_office, real_estate_agency, school, synagogue, library, political, place_of_worship, police.
- Transporte y turismo: bus_station, parking, train_station, transit_station, subway_station, gas_station, car_dealer, car_rental, taxi_stand, lodging, travel_agency, airport.
- Otros: administrative_area_level_1, 2, 3 y 4, colloquial_area, country, establishment, floor, geocode, intersection, locality, natural_feature, neighborhood, point_of_interest, post_box, postal_code, postal_code_prefix, postal_code_suffix, postal_town.

Anexo B: dilución de la importancia de características en el RFE a medida que aumenta el número de variables redundantes.



Anexo C: matriz de correlaciones de Pearson para las características espaciales.

