ISSN 1688-2806

Universidad de la República
Facultad de Ingeniería

Modelado y detección de fraudes en redes inteligentes de distribución de energía eléctrica

Tesis presentada a la Facultad de Ingeniería de la Universidad de la República por

Fernando Viera

en cumplimiento parcial de los requerimientos para la obtención del título de Magister en Ingeniería Eléctrica.

Director de Tesis

Alicia Fernández ..... Universidad de la República

Tribunal

Félix Biscarri ..... Universidad de Sevilla
Gonzalo Casaravilla ..... Universidad de la República
Matías Di Martino ..... Duke University
Juan Pablo Kosut ..... Universidad de la República

Director Académico

Alicia Fernández ..... Universidad de la República

Montevideo
miércoles 14 octubre, 2020

Modelado y detección de fraudes en redes inteligentes de distribución de energía eléctrica, Fernando Viera.

ISSN 1688-2806

Esta tesis fue preparada en LaTeX usando la clase iietesis (v1.1).
Contiene un total de 98 páginas.
Compilada el miércoles 14 octubre, 2020.
http://iie.fing.edu.uy/

Agradecimientos

Mis agradecimientos van dirigidos a las personas que más cerca estuvieron durante el proceso y a quiénes me alentaron a seguir este camino.

A Lourdes, por su apoyo emocional y estar siempre presente.

A Javier, la persona que me ha enseñado el significado del esfuerzo, y de ser mejores cada día, gracias por seguir siendo ejemplo y referencia para mi.

A mis hermanos, por siempre estar en todas, por ser los primeros en escuchar y alentar.

Cuando hablo de mis hermanos, no puedo olvidarme de amigos como Juan Andrés, que más puedo decir si es como un hermano.

A Verónica, porque su actuar es ejemplo, sinónimo de esfuerzo y dedicación.

A Alicia Fernández, y Pablo Massaferro, del Instituto de Ingeniería Eléctrica de la Universidad de la República. Por estar siempre presentes, aconsejando y alentándome a seguir, siendo guía fundamental de este recorrido. Agradezco profundamente su participación durante toda la Maestría.

Expando también mi gratitud y agradecimiento a todos los docentes que impartieron los cursos de la actividad programada de la Maestría.

Finalmente, dedico esta tesis a los seres queridos que hoy ya no están y que seguro estarían muy orgullosos por este logro, parte de esto es gracias a su apoyo y compañía mientras estuvieron presentes.

Esta página ha sido intencionalmente dejada en blanco.

Resumen

Las pérdidas de energía ocurren durante las diferentes etapas de su entrega, tanto en la generación, como en transmisión y distribución. Estas pérdidas pueden ser clasificadas en pérdidas técnicas (TLs) y pérdidas no técnicas (NTLs). Una de las principales causas de las NTLs deriva de los diferentes tipos de fraudes que algunos clientes pueden llegar a cometer. Estos eventos pueden generar grandes pérdidas a las compañías de suministro eléctrico, además de poner en peligro la seguridad de las personas, resultando en un aspecto crítico en la gestión diaria de la empresa. La aparición de medidores inteligentes en las nuevas infraestructuras (Smart grids) trae consigo un nuevo abanico de oportunidades para el área de detección de fraudes en el consumo energético.

El objetivo del presente trabajo es estudiar la detección de fraudes en medidores inteligentes a través del modelado de distintos tipos de fraudes, y la generación de bases de datos sintéticas que permitan entrenar y evaluar algoritmos en el estado del arte. El alcance del trabajo incluye el preprocesado de los datos, adecuando los mismos para servir de entrada de los modelos considerados, incluyendo una propuesta basada en simular el balance en subestaciones. Se aborda también la generación sintética de fraudes antes del entrenamiento, explicando los tipos de fraudes considerados y la estadística que modela los mismos. El trabajo aborda distintas estrategias de detección de fraudes basadas en redes neuronales profundas, propuestas de inclusión de información de subestación, análisis de complementariedad, y fusión de modelos. A su vez, se analiza cuánto degrada la detección contar con un menor histórico de datos, resultado que complementa el análisis del efecto del cambio de granularidad en el desempeño. El trabajo incluye un capítulo de ensayos experimentales, los cuáles son ejecutados sobre una base de datos pública de consumo de energía residencial con frecuencia 30 minutal. Como ensayo final se evalúa el desempeño del mejor modelo obtenido en una prueba de campo, sobre una base de datos real con fraudes reales obtenida de la empresa estatal Uruguaya, UTE. Los modelos evaluados muestran que con datos 30 minutales y un histórico de 1 año y medio, se puede lograr muy buen desempeño. También se constató que la degradación puede ser del orden del 20 % cuando se baja a una granularidad diaria o cuando se reduce la historia a 1 mes. Asimismo, se vió que introducir información del balance de subestaciones puede mejorar en algunos puntos el desempeño. En cuanto el comportamiento de los distintos tipos de fraude se obtuvo que existía diferencias de desempeño, y que a futuro se podría considerar emplear estrategias de ensamblado de modelos para diseñar topologías especializadas en todos los fraudes.

Esta página ha sido intencionalmente dejada en blanco.

Tabla de contenidos

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 1

Introducción

El uso de los sistemas de redes inteligentes en el suministro eléctrico ha venido creciendo en los últimos años. Estos sistemas permiten gestionar la demanda de energía, no solo en un sentido (hacia el cliente), sino que de manera bidireccional donde los consumidores pueden convertirse también en pequeños productores de electricidad. Estas redes (Smart Grids) son básicamente redes de distribución eléctrica combinadas con elementos de telecomunicaciones, de manera que se propagan datos (información) hacia los clientes y desde los mismos hacia las empresas distribuidoras, lo que proporciona ventajas y nuevas posibilidades para ambas partes. Con el propósito de dotar de inteligencia a las 'Smart Grids', las mismas contienen entre otros dispositivos, contadores de lecturas digitales, también denominados contadores o medidores inteligentes, permitiendo conocer los consumos en tiempo real. Además de la información cruda que deriva de los consumos eléctricos, los datos obtenidos también permiten conocer hábitos de consumo, contribuir a la optimización del mismo, así como al ahorro energético.

A pesar de las oportunidades y ventajas que presentan las 'Smart Grids', existen algunos aspectos que exigen ser tratados con cuidado. Al igual que los medidores convencionales, los contadores inteligentes presentan algunas vulnerabilidades. En este caso, se pretende abordar la problemática que identificamos como fraudes eléctricos o pérdidas no técnicas. En particular se busca mejorar la detección de las maniobras usuales: enganches directo o derivaciones, manipulación del ICP, los precintos, los empalmes, etc., e introducir la detección de maniobras más sofisticadas como por ejemplo la intrusión a nivel de software, que aparecen con los medidores inteligentes.

Las pérdidas pueden clasificarse como pérdidas no técnicas y pérdidas técnicas, las pérdidas técnicas son inherentes al transporte de electricidad causado por diferentes componentes que conforman la red, en ellos se producen pérdidas energéticas propias de su funcionamiento. Las pérdidas no técnicas están vinculadas a los fraudes eléctricos, maniobras que consisten en alterar el conexionado eléctrico generando un 'bypass' o derivaciones de la misma, o en manipular los registros del consumo, con el objetivo de reducir parcialmente el importe facturado por el mismo. Este tipo de perdidas se hace cada vez mas considerable, creciendo año a año en empresas que no traten las mismas con la atención debida. Por ejemplo,

las costos causados por las pérdidas no técnicas (NTLs) están estimadas en $4.5 billones de dólares cada año en los Estados Unidos [4], y se estima que tomando el total de compañías del resto del mundo se pierde mas de 20 billones cada año por esta causa [5]. Las empresas de energía han venido utilizando diversas metodologías para afrontar el problema, sin embargo la mayoría de ellas requieren de una demanda excesiva de horas y recursos para su desarrollo. Entre estas tareas, cada vez menos costo-efectivas se incluyen inspecciones de líneas eléctricas 'in situ', análisis manual y comparación de datos maliciosos y fiables recabados por los medidores, y tareas de chequeo de equipamientos. Los recursos necesarios y horas acumuladas para la ejecución de estas tareas arrojan un costo considerable a la compañía, sin embargo no llega a ser suficiente considerando eventuales ataques de ciberseguridad.

Debido a estas limitaciones, se han comenzado a estudiar otras alternativas para la detección de pérdidas no técnicas, basadas en el crecimiento exponencial de la capacidad de computo y nuevas tecnologías de procesamiento de señales. Haciendo uso de estas herramientas, se busca un enfoque mas inteligente que involucre la automatización de las tareas más demandantes. El análisis de datos provenientes de las redes inteligentes (Smart grids) es útil en la detección de NTLs debido a los patrones anormales que se generan en las curvas de los clientes sospechosos. Durante los últimos 5 años, y acompasados del aumento en la disponibilidad de datos, se han publicado numerosas propuestas para abordar este problema. Los métodos utilizados abarcan diferentes técnicas para el análisis e interpretación de los datos. Las técnicas que se han venido utilizando en problemas de detección de NTLs incluyen Máquinas de Vectores de Soporte (SVMs), redes neuronales convolucionadas (CNNs), redes fully-connected, algoritmos combinados fully-connected - CNN, detección por balance energético, análisis mediante series temporales, etc. En particular, las estrategias basadas en algoritmos como SVMs y árboles de decisión son aptos para ser implementados en tiempo real, sin embargo algoritmos basados en redes convolucionadas logran codificar mejor la información en espacios de alta dimensión.

Este trabajo pone foco en la solución del problema de detección de fraudes (o NTLs) en medidores inteligentes de consumo de energía eléctrica. En este sentido, se propone el análisis de diversos modelos de aprendizaje automático que serán entrenados para resolver un problema del tipo discreto, donde se discriminará cada cliente en una de las dos clases predefinidas (fraude o normal). Haciendo uso de una base de datos que contiene los históricos de consumidores de Irlanda (ISSDA [20]), sobre la cual se generan fraudes sintéticos, el presente trabajo propone entrenar y testear el modelo 'Wide&Deep' [39] en el problema de detección inteligente de fraudes. Para finalizar el estudio, se evaluará cuánto se degrada el modelo al contar con un menor histórico de datos.

El informe inicia citando el estado del arte, ésta investigación inicial dará paso a la posterior selección de la base de datos a utilizar y modelos cuyo desempeño se estudiará. Luego de presentar una revisión del estado de los métodos y bases de datos disponibles, en el capítulo 3, se da una introducción a las técnicas de aprendizaje automático que serán utilizadas en los modelos considerados. El obje-

tivo del capítulo es dar un marco teórico y repasar los conceptos básicos que rigen sobre los componentes utilizados en los modelos. Se repasarán generalidades de las redes 'fully-connected', y redes 'CNN', así como la metodología para el cálculo del balance energético entre cliente y subestaciones eléctricas. El capítulo describe la topología 'Wide&Deep', formado por la combinación de los anteriores, y presenta el modelo que incluirá la información del balance energético. Asimismo, se describe las métricas más utilizadas en este tipo de problemas, así como las diferentes estrategias de fusión de modelos, métodos de regularización y balance de clases.

En el capítulo 4 se describen los datos que se utilizarán en los experimentos, incluyendo características tales como el origen de los mismos, granularidad, tamaño de la base, y fecha de adquisición. Se presenta la Base de datos CER (ISSDA [20]). El capítulo describe el modelado aleatorio de fraudes sintéticos, tomando la propuesta de modelar la incertidumbre en el inicio y finalización de las mismas. El capítulo expone toda la etapa de preprocesado requerida para la limpieza y utilización de los datos, además del proceso para generar la base de datos del balance energético.

El capítulo 5 tiene por objetivo evaluar el desempeño del modelo 'Wide&Deep' con la base de datos CER, testeando también los modelos que forman esta topología ('Fully-connected', y 'CNN'). En una segunda instancia se analizan los modelos que derivan de la inclusión de la información proporcionada por el balance de cargas (balance energético cliente-subestación). Se incluye además el análisis de complementariedad entre modelos, para su posterior fusión. Los experimentos realizados, servirán para medir la performance de los modelos seleccionados, permitiendo observar cómo se comportan con una base de datos de fraudes simulados, variando la granularidad de los datos, y el volumen histórico de los mismos. El resultado de cada ensayo será presentado por modelo, y por tipo de fraude, considerando las métricas vistas anteriormente (3.10).
El capítulo culmina los experimentos con un ensayo de Test para los mejores modelos encontrados. Del resultado en test se concluyó que el incluir la información de las subestaciones mejora la capacidad de detección del modelo 'Wide&Deep' original, considerando tanto el modelo sin fusión como ambos modelos fusionados ('WDSB-CNN' y 'WDSB-WD' (5.11.1)). De esta manera se concluyó que existe un aporte de información al agregar los datos del balance de carga de las subestaciones, así como también la existencia de complementariedad entre los modelos considerados en las fusiones 'WDSB-CNN' y 'WDSB-WD'.
Finalmente el capítulo plantea evaluar el grado de degradación que el modelo 'Wide&Deep' pudiese sufrir en caso de un recorte considerable en el volumen de muestras históricas por cliente. En este sentido, se realizó un ensayo con la base de datos 'CER' (30 minutal), pero con un 5 % del total de sus muestras, lo que equivale a 1 mes de datos. De los resultados obtenidos se concluyó que efectivamente esta acción impacta en el desempeño alcanzado con la base 'CER' completa. Se aprecia entonces una degradación del modelo entrenado, lo que se atribuye a la pérdida considerable de información.

Los ensayos culminan con una prueba de campo en la que se evalúa el desempeño del modelo 'Wide&Deep' sobre una base de datos de fraudes reales pro-

veniente de la empresa estatal Uruguaya, UTE. En este sentido, se realizan dos experimentos, en primer instancia entrenando el modelo con la base de datos 'CER' y testeando sobre la base de datos UTE, y un segundo ensayo con entrenamiento y test sobre la base de datos UTE.

Para finalizar el documento se presentan las conclusiones obtenidas de los experimentos realizados y se proponen posibles líneas de trabajo futuro en lo que respecta al área de la detección de NTLs en medidores inteligentes. En primera instancia, se podría apuntar a tener una base de datos real más sólida, con un mayor histórico de muestras y clientes etiquetados, de esta manera se podría entrenar un modelo que aprenda mejor la naturaleza de fraudes reales. En cuanto a los modelos considerados, queda a futuro realizar una prueba con la base de datos 'CER' sintética y la red 'Wide&Deep' con los datos de balance de carga entrando por un nuevo canal de su componente 'Deep', en lugar de ser entrada de la red 'Wide' (como fue implementado). Así mismo, se podría probar integrar la información del balance de carga en modelos más actuales, como ser los que se describen el los articulo [12] y [17].

Capítulo 2

Estado del Arte

2.1. Estudios recientes

Considerando el problema detección de NTLs en 'Smart Grids', se han identificado algunos artículos que muestran obtener buenos resultados, y que serán considerados como referencia de nuestro estudio para su posterior implementación. Existe un gran número de publicaciones que abordan el problema de detección de NTLs en 'Smart Grids' mediante la utilización de técnicas y datasets diferentes. Existen algunos que preponen el uso de la base de datos CER y topologías AMI [1, 2, 11, 12, 17, 22, 26, 28, 31–34, 36–38], otros con enfoques basados en redes neuronales [3, 9], enfoques basados en SVMs [4, 24, 25], y modelos basados en redes convolucionadas [17, 39]. Otros trabajos abordan estrategias particulares como en [21] donde se emplea una estrategia de fusión de datos. A su vez, se encuentran trabajos que abordan el estado del arte en sí, describiendo las diferentes generalidades de de las NTLs [35].

De la biografía consultada se destaca el articulo [39], que se tomará como la referencia principal para este trabajo. En el mismo, los autores proponen un modelo basado en un sistema modular implementado con una red neuronal convolucionada combinada con una red 'fully-connected'. El dataset con el que trabajan es obtenido de la "State Grid Corporation of China (SGCC [1])".

Otros trabajos, como en [38], aborda el problema desde el punto de vista del balance energético a través de un modelo de matemática lineal. El mismo toma datos de la 'Irish Social Science Data Archive (ISSDA)' [20], organismo que registra datos que surgen del proyecto "Customer Behavior Trials" llevado a cabo por la Comisión Reguladora de Energía (CER) en Irlanda.
Los métodos convencionales ('fully-connected', árboles de decisión, etc.) presentan bajas exactitudes al momento de detectar los casos de fraude, siendo una de las principales causas la unidimensionalidad de los datos tomados como entrada. Esta característica evita que los métodos empleados capturen la periodicidad en las lecturas de consumo energéticos. En [12], los autores presentan un modelo llamado BSBND (basado en vectores de soporte y redes GAN), el mismo esta motivado

[1] State Grid Corporation of China http://www.sgcc.com.cn/

por el poder que ofrecen las redes 'generative adversarial networks' (GANs) en aprender representaciones de alta dimensión de los datos. Esta componente es utilizada esencialmente para extraer 'features' a partir de los valores de largas filas de datos temporales por cliente. Al igual que en el artículo [38] y [17], aquí también toman la base de datos CER como fuente de datos para sus experimentos. Por otro lado, en el artículo [17] se propone un modelo híbrido para la detección de fraudes eléctricos conformado por una red convolucionada combinada con un modelo Random Forest (CNN-RF). La salida de esta red es un vector de 'features' que entra a una red de tipo 'Random Forest' quién decide si la entrada corresponde a un cliente fraudulento o no. Este artículo no solo es interesante por el método propuesto, sino también por utilizar la base de datos CER para realizar los ensayos sobre el mismo. Son varios los modelos que se ensayan en este articulo, las pruebas reportadas incluyen modelos del tipo 'Logistic Regression', 'SVMs', 'Random Forest', 'Gradient boosting decision trees', 'Métodos de Deep Learning' , y 'CNN-RF'.

Por otro lado en [17], los autores apuntan al uso de los clasificadores del tipo árboles de decisión para sustituir lo que normalmente sería una capa softmax, aumentando así el desempeño en la detección. Se concluye que los algoritmos basados en aprendizaje automático a partir redes neuronales convolucionadas combinadas con arboles de decisión, y redes GAN combinadas con SVM, demuestran tener mayor exactitud en la identificación de NTLs.

Además de las metodologías que se centran en extraer información de los datos en alta dimensión, existen otras, como en [39], donde además se propone un método que toma en consideración la naturaleza periódica de los datos de consumo energético. Esta técnica se basa en un modelo de dos módulos (un esquema detallado del sistema propuesto puede verse en la figura 5 del artículo), los modelos que allí se describen son los siguientes:

- Red Neuronal fully-connected (Softmax): aprende la naturaleza global de los datos unidimensionales.
- Red neuronal convolucionada profunda (DCNN): reconoce la periodicidad de los datos temporales del consumo eléctrico de clientes normales y la diferencia de la no-periodicidad de los clientes que cometen fraude en la red.

Los resultados obtenidos están representados en la Tabla III del artículo, los mismos se comparan contra métodos tradicionales como regresión logística (LR), árboles de decisión (random forest - RF), SVMs, red neuronal 'fully-connected' (Wide), y red neuronal convolucionada (CNN). El desempeño obtenido muestra un incremento en el valor del mismo al combinar módulos 'fully-connected' con una red 'CNN'. Al mismo tiempo se estudió el efecto generado por variables alfa, beta y gama utilizadas para configurar la red.

Una estrategia diferente se propone en el artículo "An anomaly detection framework for identifying energy theft and defective meters in smart grids" [38].Su estrategia de detección está basada en el estudio del balance de la Red NAN (red de medidores inteligentes instalados a pie de los consumidores).De esta manera

propone mitigar las NTLs identificando patrones característicos que resultan de la evaluación de coeficientes anómalos de consumidores presentes en los reportes de facturación generados por el Operador Local. Se introducen las métricas de factor de pérdida y término de error para estimar el valor de NTLs y capturar el ruido en la medida en las líneas de distribución y subestaciones. Una de las características que se presentan como novedad en el artículo es la capacidad que tiene el modelo propuesto en detectar NTLs de corta duración y frecuencia variables, dotándolo de robustez en la detección de NTLs. Esto es posible a través de la inclusión de las lecturas de subestaciones, o centros 'AMI' (como se llaman en el articulo), cuya funcionalidad es concentrar las lecturas de cierto grupo de clientes.
Los resultados obtenidos y posteriores verificaciones con ensayos de laboratorio indican que el modelo propuesto permite detectar satisfactoriamente clientes fraudulentos y medidores inteligentes defectuosos. Además se presentan dos nuevos esquemas que adoptan programación lineal para abordar algunos de los problemas asociados a las NTLs. Debido a que la programación lineal permite proteger contra ataques de contaminación, da robustez frente a factores no maliciosos, no presenta restricción debido a la dimensión de los datos provenientes del espacio de consumidores, y es capaz de detectar NTLs a partir de volúmenes pequeños de datos.

En cuanto a la extracción de características, los artículos [17] y [34] presentan modelos en que la componente 'CNN' se utiliza para extraer características entre gran cantidad de perfiles de consumo, para luego, a través del uso de técnicas como SVM, identificar las características de los consumidores. En [17], a pesar de tener excelente desempeño en procesos de extracción de características, los autores señalan que las redes 'CNN', por si solas, no son óptimas en tareas de clasificación. A su vez, resaltan esta etapa como un factor vital en el éxito de su modelo, haciendo énfasis en la problemática que deriva de la extracción de características basado en la experiencia, donde los datos son sensibles a condiciones climáticas, a estilos de vida según estación del año, al contexto operativo, etc.

En el desarrollo de modelos entrenados para la detección de NTLs, y como en la mayoría de las propuestas que implican aprendizaje automático, es necesaria una etapa preprocesado de los datos. Este proceso incluye la limpieza de datos, recuperación de valores perdidos, y el ordenamiento de los mismos. Por ejemplo, en [17] los autores proponen 2 métodos diferentes para recuperar datos perdidos, según el tipo, vieron que existen dos grupos; uno de ellos corresponde al grupo de valores perdidos que son contiguos; el otro caso son aquellos 'huecos' aislados (1 sólo valor perdido). Para los primeros se toma la decisión de eliminar al cliente en caso de presentar alguna porción de sus datos con más de 10 valores perdidos, y en el segundo caso la solución es rellenar el dato perdido con la media entre el valor pasado y futuro. Por último, en caso de que el valor en un instante dado sea 'null', se lo remplaza por 'NaN'. Otro aspecto clave es el re-ordenado (o 'reshape') de los datos, esto se observa en [12] y [39], donde los datos son preprocesados y convertidos a matrices (dimensión 2D) de ancho semanal, y cantidad de filas el número entero de semanas relevadas por cliente. O por ejemplo en [17], que toman cada fila de la matriz como 1 día donde cada columnas representa cada una de las

lecturas en ese día.

En cuanto a la normalización, en [17] se habla de la importancia de esta etapa para suavizar la sensibilidad que presentan las arquitecturas formadas por redes neuronales, para ello proponen el método 'MinMax' el cual re-escala los datos entre 0 y 1. Para generar los dataset de entrenamiento y test, los autores en [17] proponen un método similar al del articulo [39], donde transforman los datos del tipo serie temporal de 1D al espacio de 2D. Sin embargo, en lugar de considerar cada fila un periodo de 1 semana (como en [39]), cada fila de la matriz en este caso representa 1 día y las columnas cada una de las lecturas por día. Finalmente particionan el dataset con relación 80-20 (luego del oversampling 'SMOT') para entrenamiento y test respectivamente.

Durante el estudio de los artículos citados se prestó especial atención a las bases de datos utilizadas, siendo muchas veces éstas, puntos de partida al momento de investigar el estado del arte. En [39], las pruebas son realizadas sobre una base de datos real proveniente de la 'State Grid Corporation of China (SGCC)'. La misma es de acceso restringido y su contenido incluye datos de 42,372 consumidores durante 1035 días, del total de consumidores; 38,757 son consumidores normales; mientras que 3,615 (9 %) fueron etiquetados como anómalos. Los datos utilizados en [38] y en [17] son obtenidos de la red inteligente montada en Irlanda cuyos datos son suministrados por el 'Irish Social Science Data Archive' (ISSDA [20]), estos son utilizados para modelar los patrones normales de consumo eléctrico. Como los datos del tipo NTLs no existen en esta base de datos, la evaluación del método de detección de NTLs utiliza datos anómalos generados de manera sintética, modificando con ciertos patrones los datos normales. En particular, los autores del artículo [17] asumen que la base de datos CER no contiene NTLs, considerando que fue producto de una prueba de investigación en Irlanda en la cual los clientes fueron colaboradores voluntarios y por ende se los consideró consumidores honestos. Para encarar el problema resuelven generar 1200 clientes maliciosos como se describe en el artículo "Electricity theft detection in AMI using customers" [13]. Esta base de datos sin NTLs, la cual llamamos 'CER', será la que se utilizará como base de todos los experimentos a realizar en el presente trabajo.

En el artículo [12], se encontró un criterio de generación de 8 tipos de fraude distintos, el criterio adoptado es elegir aleatoriamente entre estos tipos de fraude e inyectar sobre la serie temporal de cada cliente seleccionado. Los autores en [12] diseñaron éstos 8 tipos de NTLs inspirados en el artículo [13], donde se explica además de las fórmulas utilizadas, la interpretación física de los NTLs considerados.

Debido a que las bases de datos tienen una tasa de muestreo fija, y generalmente se presentan en una única frecuencia, parte del trabajo del preprocesado de datos también implicará el cambio en la granularidad original. Por ejemplo, en el caso de la base de datos CER, debido a que la misma presenta una granularidad de lecturas 30 minutal, los autores en [17] proponen una reducción de la frecuencia para trabajar con datos de resolución horaria. Existen trabajos que estudian el impacto de la granularidad pero no en el rango de nuestro interés, por ejemplo en [10] se reporta el análisis de un modelo de Deep learning con base de datos china al variar la agregación de diaria a semanal, quincenal y mensual. En este

caso muestran que para el modelo propuesto no hay variaciones significativas en los indicadores de desempeño al tener una granularidad menor. En nuestro caso nos interesa trabajar con granularidad mas cercana a quinceminutal, siendo ésta la frecuencia de interés. A diferencia de los artículos analizados, uno de los objetivos será observar qué ocurre al aumentar la granularidad de los datos.

La naturaleza de desbalance en los datos puede tener un impacto negativo en la performance de métodos de aprendizaje supervisado. Existen diversos métodos para reducir o contrarretsar este impacto. Por ejemplo, en [12] y [14] se destaca la implementación de un método algo diferente a los vistos anteriormente. Allí se propone el método de clasificación de una clase, donde se entrena un modelo ('kernel-SVDD') que logra identificar NTLs aprendiendo únicamente de las muestras no anómalas. Este tipo de estrategias, dónde el modelo propuesto sólo aprende de las clases no fraudulentas, permite que el modelo entrenado no sea sensible a un tipo de fraude determinado, lo que supone una ventaja y robustez al modelo (debido a los posibles cambios de contexto y naturaleza del fraude). A su vez esta estrategia permite entrenar al modelo con datos reales sin necesidad de generar para el entrenamiento, por ejemplo en [12], el fraude sintético se genera únicamente en el grupo de Test. Sin embargo, este tipo de estrategias dificulta la manera de implementar una búsqueda del tipo "grid-search" tal como se describe en [14] y [8]. Existen otras opciones para contrarrestar este problema, una opción es proceder como en [17], allí se propone utilizar la técnica de 'oversampling' sintético de la clase minoritaria para contrarrestar el impacto del desbalance entre clases, además de usar la técnica de 'dropout' durante el entrenamiento de la red 'CNN'. Existe también la técnica de agregar un peso a cada clase durante el entrenamiento, esta técnica implica agregar un mayor peso a la clase minoritaria simplemente multiplicando el valor de 'loss' (en cada batch) por cierto factor que depende de las proporciones de las clases.

Revisando el estado del arte, también se prestó especial atención a las métricas que se utilizan para medir el desempeño de los modelos en problemas de este tipo. Tomando como referencia los artículos vistos, se observó que en algunos casos (ver [12]) utilizan los ratios FPR y TPR, obtenidos a partir de los valores de falsos positivos (FP), falsos negativos (FN), verdaderos positivos (TP), y verdaderos negativos (TN) . Por otro lado, en [17] se basan en las características de la curva ROC y la información proporcionada por la matriz de confusión resultantes de la prueba de test. En [39] se propone, entre otras métricas, relevar el MAP100. En el articulo, los autores usan dicha métrica para medir el desempeño del modelo. El uso de esta métrica requiere reordenar las etiquetas de test según el 'score' obtenido en el vector de predicción de fraude para cada cliente (obtenido a la salida del modelo). Luego de ordenar las etiquetas, se toman las 100 primeras para evaluar el desempeño (basado en la precisión obtenida en los primeros 100 clientes sospechosos). Una métrica de un estilo similar al anterior, se propone en [17], donde el objetivo final del artículo es presentar el resultado en una especie de lista de clientes ordenada acorde a la probabilidad de fraude calculada para cada uno de ellos.

En trabajos previos realizados por el grupo de investigación del IIE en cola-

boración con UTE [5–7, 16, 19, 27, 30], se realizaron distintos aportes al área de detección de anomalías para medidores convencionales, con estrategias de aprendizaje supervisado, y con especial foco en el problema del desbalance de los datos y la propuesta de métricas adecuadas al problema. En particular en el artículo [18] se propone una metodología basada en maximizar el retorno económico para determinar el punto óptimo de trabajo. Los autores proponen un enfoque enmarcado en la inferencia Bayesiana ('Bayesian-Risk'), dicha estrategia implica minimizar la tasa de error de Bayes, logrando que al clasificar cada muestra se alcance una solución óptima en términos de minimizar el error medio de clasificación.

Para poder evaluar los resultados del presente trabajo y tener una referencia base con la cual compararse, se revisaron los mejores valores logrados según las métricas consideradas en cada caso. Por ejemplo en [17], el AUC obtenidos en sus mejores modelos (CNN-RF, CNN-GBDT, CNN-SVM) alcanzaron valores de hasta 0.99 de AUC (con entrenamiento y test sobre la base de datos CER). Los autores atribuyen estos resultados al aporte de la red 'CNN', y su capacidad para detectar 'features' dentro de una gran cantidad de datos de medidas eléctricas. Modelos basado en vectores de soporte y redes GAN, como el que proponen en [12], logran resultados en DR (TPR) de 90.22 % y el más bajo logrado alcanzó un FPR de 6.77 %. En el caso del articulo [39], se reportaron resultados de hasta 0.80 en el valor de AUC, logrado con el modelo 'Wide&Deep' y R=5 (ver tabla IV). Este mismo modelo alcanzó un valor MAP100 de 0.95.

Capítulo 3

NTLs - Análisis y selección de modelos

3.1. Resumen.

En este capítulo se introducen los modelos seleccionados para abordar el problema de detección de fraudes en medidores inteligentes. Las topologías propuestas parten de los modelos vistos en el estado del arte, y de la simplificación de los mismos en modelos mas simples con el objeto de su comparación. Estos modelos más básicos son las conocidas redes 'fully-connected' y convolucionadas. Ambas redes serán ensayadas por separado y combinadas según se presenta en el modelo 'Wide&Deep'. Los modelos descriptos serán evaluados sobre una base de datos de diferentes frecuencias, de este modo se observará el efecto del cambio de resolución en la capacidad para detectar fraudes. El capítulo comienza con una breve descripción de las características principales de los modelos considerados, continuando con el abordaje de la dificultad adicional que surge de desbalance en las muestras, por último se propone y se presenta un método en el qué se incluye la información del consumo instantáneo total en cada subestación.

3.2. Redes 'fully-connected' - generalidades

Las redes neuronales conocidas como 'fully-connected' totalmente conectadas forman parte de los modelos de aprendizaje supervisado cuya característica principal es la de presentar diversas capas de neuronas interconectadas. Las mismas crean una red compuesta por diversas capas neuronales, ordenadas según su ubicación en la red, en este sentido se tienen capas de entrada, capas ocultas y de salida. La capa de entrada es la que recibe el vector de entrada 'x' unidimensional, la salida de cada neurona en esta capa será entrada de la 1^{er} capa oculta y así sucesivamente hasta llegar a la capa de salida. El comportamiento de la red es una sucesión de etapas con paso hacia delante conformadas como se muestra en la figura 3.1.

La ecuación que rige el comportamiento de cada neurona es la que conocemos para el 'perceptron', por lo tanto para cada una de las capas de la red se puede expresar su salida como: $y_j = \sum_{i=1}^{n} w_{i,j} x_i + b$, donde y_j es la salida de una

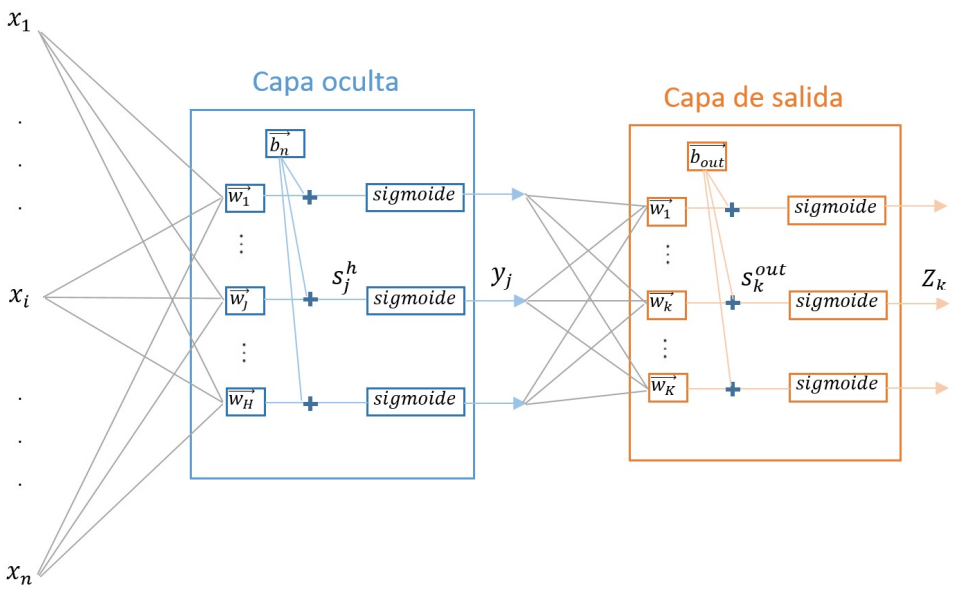

Figura 3.1: Ejemplo capas de red 'fully-conected'.

de las capas, específicamente en la neurona j-ésima, y 'n' es el largo del vector unidimensional de entrada a la capa. El elemento $w_{i,j}$ es el i-ésimo vector de pesos de la neurona j-ésima de la capa, y 'b' el valor del 'offset' de toda la capa. El vector producto a la salida (vector 'y') formado por cada elemento y_j será entrada de la capa siguiente de la red 'fully-connected' luego de pasar por una función de activación (por ejemplo la función 'sigmoide') que determina cuáles de los elementos de la salida 'y_j' tendrán repercusión en la capa siguiente de predicción. La función de activación que se utiliza en el modelo a emplear es la función 'ReLU' cuya ecuación es la siguiente: $u_j = f(y_j) = max(0, y_j)$, donde u_j es la salida de la función de activación y $max(0, y_j)$ el respectivo valor del mayor entre 0 e y_j, lo que implica quedarse con la salida y_j siempre que esta sea positiva. Además de cumplir la función de activación, el cálculo por ReLU puede también prevenir el sobreajuste.

Durante el entrenamiento se van ajustando los pesos de cada neurona mediante pasos de 'back-propagation', en este sentido el valor de $w_{i,j}$ se actualiza de acuerdo a la variación de la función de costo, es decir: $w(m+1) = w(m) + \triangle w(m)$, con $\triangle w(m) = -\eta.\delta J/\delta w$, J la función de costo, y η la tasa de aprendizaje o 'learning rate'.

Considerando ahora, el caso de estudio en cuestión, la entrada a la red serán las lecturas temporales del consumo eléctrico por cada cliente, es decir que cada vector de entrada representa una serie temporal unidimensional con las lecturas de consumo 30 minutales. Las mismas son tomadas desde el instante inicial de adquisición hasta el último dato guardado, para un cliente dado. El empleo de la red 'fully-connected' permitirá reconocer la co-ocurrencia entre características o 'features' de toda la serie temporal, lo que permite detectar un comportamiento

fraudulento o no del usuario.

3.3. Redes convolucionadas - generalidades

A diferencia de las redes densas ('fully-connected'), aquí los pasos 'forward' implican convoluciones con filtros de diferentes tamaños y profundidad. Por ejemplo, considerando una imagen de entrada 'u' y un filtro de convolución 'h', la salida resultante será la suma ponderada de cada píxel de la imagen trasladada en el tiempo y con los elementos del filtro de convolución. De esta manera, la ecuación que representa el cálculo es: $(u * h)(i, j) = \sum u(i - k, j - l)h(k, l)$. Resultando la salida en una serie de imágenes con profundidad (canales), igual al número de filtros de convolución utilizados. De esta manera a medida que se avance en la red de convolución, las neuronas capturan información de más alto nivel, información que es muy difícil de percibir a simple vista. En cuanto a la función de activación utilizada para filtrar cada píxel de la imagen de salida, se considera también la función de activación 'ReLU' cuya ecuación se menciona arriba. En la figura 7.2 puede observarse un ejemplo general de lo que ocurre con las dimensiones a medida que se avanza en la red de convolución (ver figura 3.2).

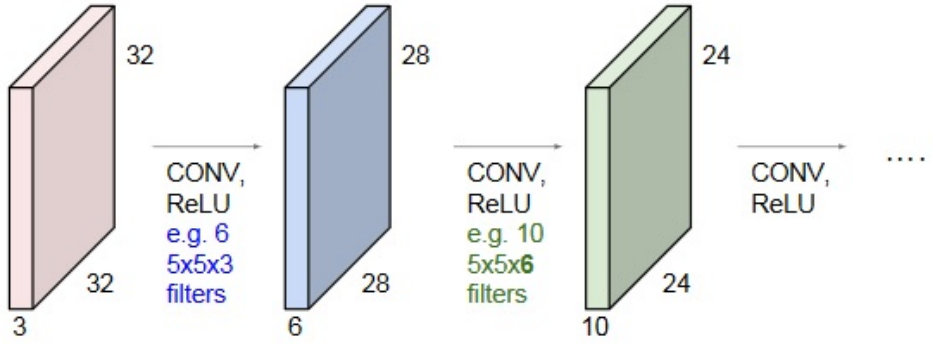

Figura 3.2: ej: Convolución de una entrada de 32x32 con filtros de 5x5 (32 → 28 → 24 ...), figura tomada de cs231n(Stanford) - Fei-Fei Li Justin Johnson & Serena Yeung.

En este tipo de modelos es usual utilizar capas 'max-pooling' a la salida de cada capa de convolución. La misma opera en cada mapa de activación (canal) por separado, y su uso permite comprimir la representación de la imagen a partir del sub-muestreo de la misma.

Para el tratamiento de los datos, en el contexto del presente trabajo, es necesaria la adaptación de la entrada unidimensional (serie temporal) a datos de dos dimensiones ordenados con una periodicidad de cierta cantidad de días. Debido a la naturaleza de los datos que se están tratando, y basándose en los desempeños obtenidos en el modelo del artículo [39], se decide fijar a 7 días el período en el que serán ordenados los vectores unidimensionales. De esta manera se obtiene como entrada una imagen de ancho 7 días (de lecturas) y alto la cantidad de semanas

Figura 3.3: Capa de max pooling, figura tomada de cs231n(Stanford) - Fei-Fei Li & Justin Johnson Serena Yeung.

que el cliente registró lecturas. Dicha transformación de los datos permitirá dar un indicio de los usuarios que siguen un comportamiento normal debido a la correlación semanal que debería existir en el patrón de consumo, y permitirá identificar como fraudulentos aquellos cuyo comportamiento tenga escasa correlación semana a semana. En las siguientes secciones se presentan lo modelos utilizados y los ensayos necesarios para evaluar el desempeño de los mismos en la detección de fraudes.

3.4. Modelo 1: Wide&Deep Convolutional Neural Networks

Se considerará el modelo presentado en [39] (en adelante llamado también 'modelo original' o 'Wide&Deep') con el fin de evaluar su desempeño en la detección de fraudes. El modelo se compone por una red 'fully-connected' en paralelo a una red de convolución (CNN). En particular, y tal como se señala en 3.3, el componente 'CNN' es capaz de captar la no periodicidad de los datos fraudulentos y la periodicidad en los datos de usuarios normales. El mismo toma como entrada datos de dos dimensiones (como imágenes) obtenidos a partir de la serie temporal de consumo eléctrico de cada cliente. Por otro lado, la componente de la red "wide"(fully-connected) toma datos unidimensionales, provenientes de la serie temporal que representa el consumo energético por cliente, siendo capaz de capturar las características globales de las medidas (correlación entre 'features'). A continuación (figura 3.4) se presenta una aproximación gráfica del modelo utilizado en [39]:

Los parámetros óptimos de funcionamiento son obtenidos a partir de ensayos del modelo 3.4, tomando como entrada al mismo las lecturas de consumos energéticos (cuyos fraudes serán simuladas sintéticamente).

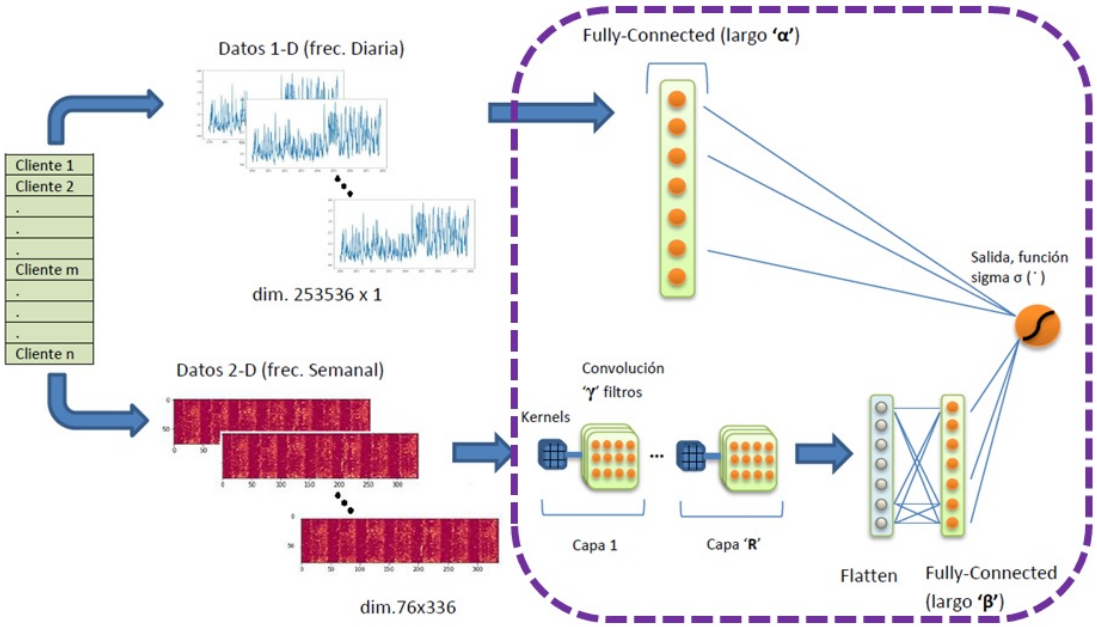

Figura 3.4: Red Wide&Deep.

Tal como se indica en el Capítulo 1, uno de los objetivos es evaluar el desempeño del modelo descripto en 3.4 y contrastar con el obtenido en el articulo de referencia [39]. Los parámetros con los cuáles se construye el mismo son los correspondientes a los resultados óptimos indicados en [39]. Se tomará entonces:

- $\alpha = 90$ (largo de la red Wide).
- $\gamma = 15$ (cantidad de filtros por capa de convolución).
- $\beta = 60$ (largo red fully-connected a la salida de la red CNN).
- $R = 5$ (capas de convolución de red CNN).

Tomando la referencia de los parámetros en [39], la topología del modelo a ensayar será la que se muestra en la figura (3.5).

Si bien en [39] se obtiene un número óptimo para los 30 'epochs' (para un 'training ratio' de 60 %) en la ejecución de sus experimentos, en nuestro caso se desarrollará una búsqueda de 'grid-search' para encontrar la combinación óptima del parámetro 'learning-rate', 'batch-size', y 'epochs' de la respuesta del modelo original.

3.5. Modelo 2: Red fully-connected

El siguiente modelo a considerar será la topología 'fully-connected' (ver 3.2) perteneciente al modelo original. En otros términos, se quita la capa 'CNN' al mo-

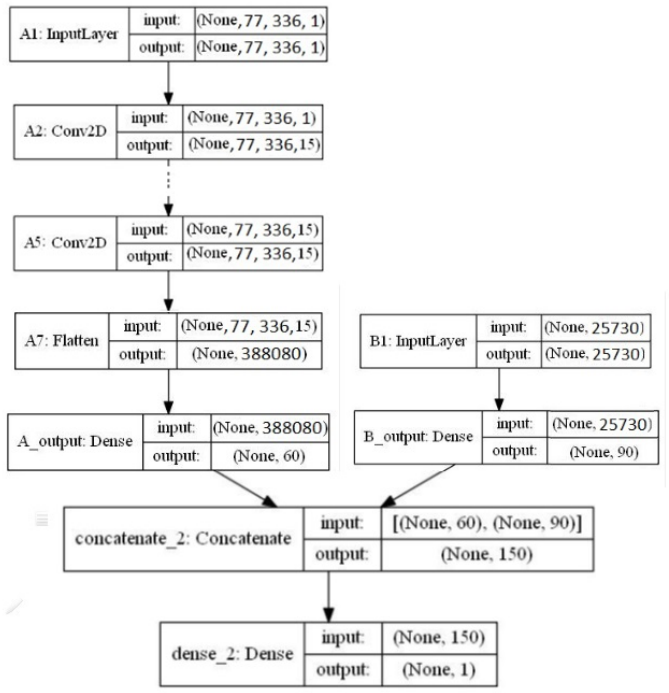

Figura 3.5: Representación modelo 'Wide&Deep' (red CNN + fully connected).

delo 'Wide&Deep' para comparar su desempeño contra el modelo completo ('Wide&Deep'). La configuración de la red 'fully-connected' a utilizar en la ejecución de los primeros experimentos se presenta en la imagen 3.6.

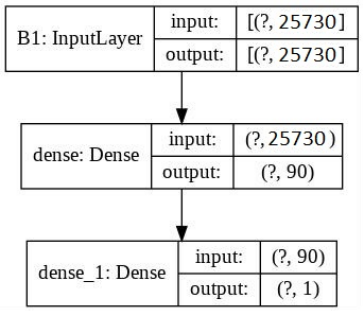

Figura 3.6: Representación de red 'fully-connected' (Wide).

Como se observa en la imagen 3.6, en la topología se respetará el ancho y cantidad de capas del modelo original. Su desempeño será evaluado según las métricas de medida que se exponen mas adelante en 3.10 y a diferentes frecuencias.

3.6. Modelo 3: Red CNN

Otro modelo a considerar será la red de Convolución (CNN) presente en el modelo original. El objetivo aquí será evaluar el efecto de quitar la capa 'Wide', y si es posible mantener el desempeño con respecto al modelo original para las diferentes frecuencias a considerar. Para realizar los ensayos se tomarán los parámetros 3.4. El modelo de red de convolución resultante puede verse en la imagen 3.7.

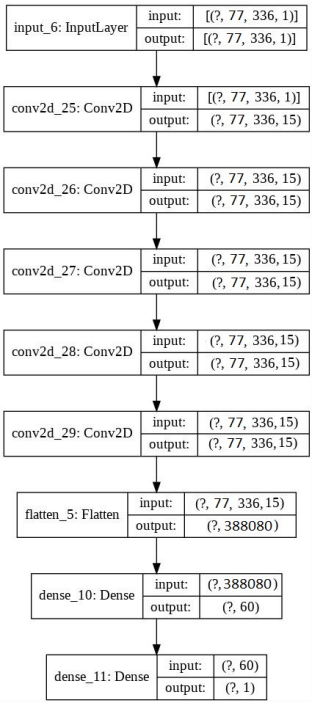

Figura 3.7: Representación modelo de red 'CNN'.

En ambos modelos se utiliza un método de aprendizaje adaptativo de descenso por gradiente conocido como 'Adam', el mismo consiste en el re-escalado de los pesos del gradiente (en cada dimensión), manteniendo la inercia (Momentum). El mismo es utilizado para entrenar modelos de aprendizaje profundo, y uno de los optimizadores más recomendados para trabajar con redes de este tipo debido a su gran efectividad para alcanzar buenos resultados rápidamente y su facilidad para ser configurado (sus hiper-parámetros por defecto funcionan bien en la mayoría de los casos).
Adam combina las mejores propiedades de los algoritmos AdaGrad (Algoritmo de Gradiente Adaptativo) y RMSProp (mantiene una media móvil con el cuadrado del gradiente para cada componente), para proveer un método de optimización

capaz de manejar matrices esparsas en problemas con alta presencia de ruido. Los pasos del algoritmo son los siguientes:

i) compute_gradient$(x_t) = \bigtriangledown f(x_t)$.
ii) first_moment= first_moment $*\beta_1 + (1 - \beta_1) \odot \bigtriangledown f(x_t)$.
iii) second_moment= second_moment $*\beta_2 + (1 - \beta_2) \odot \bigtriangledown f(x_t) \odot \bigtriangledown f(x_t)$.
iv) $x_{t+1} = x_t - \eta * first\_moment/(\sqrt{second\_moment} + 1e^-7)$.
Donde típicamente $\beta_1 = 0.9$, $\beta_2 = 0.999$ con η el 'learning rate'.

En el caso de los experimentos a realizar será utilizada la librería Adam optimizer de Keras cuyo "learning rate"por defecto es $\eta = 0{,}001$.

3.7. Regularización por Dropout

3.7.1. Aprendizaje con regularización Dropout

Existen diversas técnicas para evitar el sobreajuste de los modelos de redes neuronales, estas son llamadas técnicas de regularización y permiten alcanzar exactitudes mayores al permitir mejorar la generalización del modelo que se está entrenando. En este caso, se utilizará regularización por 'Dropout' para evaluar su efecto en el desempeño de los modelos considerados, y así evitar el sobreajuste de los mismos.

Regularización Dropout - Generalidades

La técnica de regularización por Dropout consiste en activar y desactivar aleatoriamente cierto porcentaje (fijado previamente) de neuronas (en capas ocultas) durante el entrenamiento entre cada batch de muestras. La figura 3.8 muestra una representación gráfica de lo que estaría ocurriendo en las capas ocultas de la red, antes y luego de aplicar la técnica.

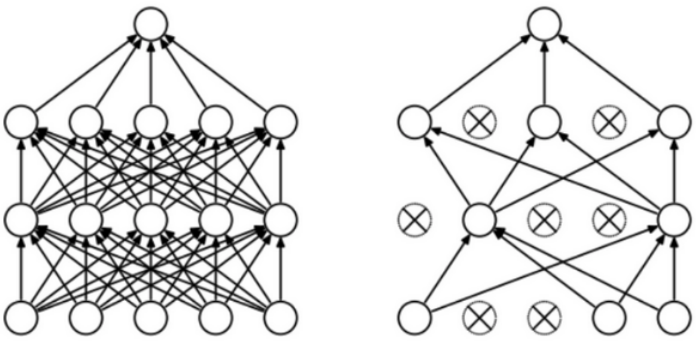

Figura 3.8: Técnica dropout, antes (izquierda) y luego de su aplicación (derecha), imagen tomada de [29].

3.8. Problema de clases desbalanceadas

Como se comentó anteriormente, este es un problema del tipo de clases desbalanceadas (% de clientes fraudulentos sobre totales entre 5-15 %). Esta particularidad puede jugar en contra al momento del entrenamiento, si simplemente se trata de minimizar la tasa de error o maximizar la tasa de acierto, ya que en estos casos, la penalización de equivocarse es la misma en hacerlo para una muestra fraudulenta y una no fraudulenta. De esta manera, el modelo puede llegar a no lograr aprender de los errores por clasificar erróneamente la clase minoritaria (fraudes), mientras que si lo hace para la clase dominante.

3.8.1. Solución propuesta: Balance de clases

Una manera de corregir este problema es agregando un mayor peso a la clase minoritaria, simplemente multiplicando el valor de 'loss' (en cada batch) por cierto factor que depende de las proporciones de las clases. Por ejemplo, considerando se tiene un espacio de 'm' datos de entrenamiento, donde 'n' son casos de uso normales y 'f' muestras fraudulentas (clase minoritaria). Entonces, los factores a utilizar en el entrenamiento son:

$$class\_weight\_n = (1/n) * 1/m * 1/2 \tag{3.1}$$

$$class\_weight\_m = (1/f) * 1/m * 1/2 \tag{3.2}$$

La técnica propuesta es una manera de corregir el problema que surge del desbalance entre clases del espacio de muestras, su efecto será evaluado en los ensayos al probar cómo responden los modelos al implementar el balance de clases.

3.9. Balance entre lecturas: subestaciones - consumidores

En el capítulo 2 se cita el articulo [38], el mismo propone un método de detección de fraudes basado en el análisis del balance energético, entre las lecturas tomadas de cada subestación de distribución y la suma total del consumo de los clientes bajo cada una de ellas. Una de las características que se presentan como novedad en el artículo es la capacidad que tiene el modelo propuesto en detectar fraudes de corta duración y frecuencia variables, dotándolo de robustez en la detección de fraudes. Para explicar la metodología es necesario entender la topología que debe presentar la red de distribución eléctrica, a modo de ejemplo se propone una red de distribución con la estructura que se muestra en la figura 3.9.

En la arquitectura se pueden observar dos tipos de redes:

- NAN: incluye los Smart meters instalados a pie de cada cliente, y los colectores de datos de cada subestación.

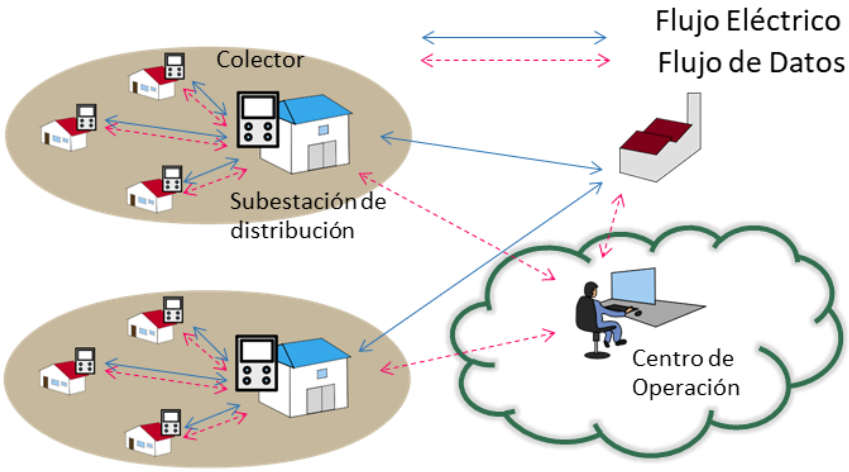

Figura 3.9: Ejemplo arquitectura AMI planteada en [38].

- WAN: La red de transmisión que conecta los colectores de datos con la central de operación (conocida como WAN).

El algoritmo de detección está basado en el estudio del balance de la red NAN (medidas a pie de los consumidores contra medida en el colector de cada subestación). Es decir, considerando un área constituida por 'N' clientes bajo la misma subestación, sean $p_{t_i,n}$ y c_{t_i} variables que indican los consumos energéticos del cliente n y del colector de la subestación respectivamente, en el intervalo de tiempo $t_i \in T = \{t_1, t_2, ..., t_T\}$. Se define el coeficiente de anomalía a_n para cada cliente, de modo que $a_n = 0$ implica que la lectura reportada por el consumidor n es confiable. Por otro lado, valores de $a_n > 0$ es indicador de que el consumidor n está cometiendo fraude. Teniendo en cuenta las consideraciones anteriores, el balance entre el consumo medido por cada subestación y la suma total de registros de cada cliente asociado a la subestación correspondiente, puede formularse a través de la siguiente ecuación:

$$(a_1 + 1)p_{t_i,1} + (a_2 + 1)p_{t_i,2} + ... + (a_N + 1)p_{t_i,N} + l_{t_i}c_{t_i} = c_{t_i} \tag{3.3}$$

Donde l_{t_i} es el coeficiente asociado a las pérdidas técnicas de la red de distribución.

3.9.1. Modelo 4: Red 'Wide&Deep' con información de subestaciones: 'Wide&Deep-sb'

El último modelo que se considerará es el modelo 'Wide&Deep-sb' (WDSB), el cuál tiene de base al modelo 'Wide&Deep', con la diferencia que en lugar de tener como entrada la serie temporal de las lecturas por la componente 'Wide', se toma la información de las subestaciones (obtenida según se explica en el capítulo 4). El

objetivo de considerar este modelo será evaluar el aporte que tiene la información del balance energético en el desempeño del modelo original. Al igual que en los modelos anteriores, este mantiene los parámetros descriptos en 3.4. La imagen 3.10 muestra una representación aproximada del modelo dando una intuición visual de su topología.

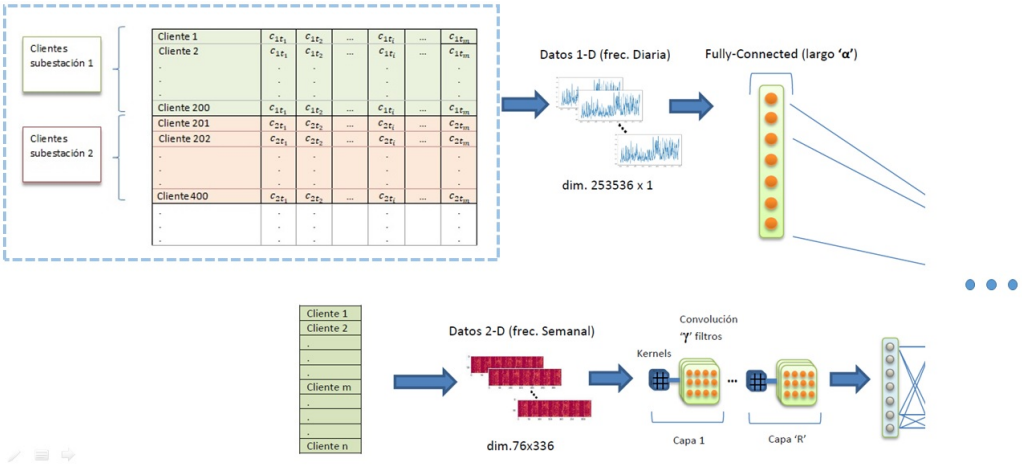

Figura 3.10: Representación modelo 'Wide&Deep-sb'.

3.9.2. Estrategias de fusiones de modelos

Con el propósito de aumentar el desempeño de los modelos considerados se propone abordar una estrategia de fusión. Algunas de las técnicas utilizadas se observan en el articulo [15], donde se detallan diferentes estrategias de fusión y combinación de clasificadores. La fusión de modelos implica incluir el resultado de dos modelos diferentes en la toma de decisión final. Por ejemplo, al considerar las salidas de ambos modelos a la vez para tener como resultado un nuevo vector de probabilidades. Este nuevo vector de salida será producto de la combinación lógica de ambas probabilidades (en cada cliente) y posterior normalización, ejecutando la operación cliente a cliente. Previo a intentar la estrategia de fusión para aumentar el desempeño del resultado final, es necesario hacer una estudio de complementariedad entre modelos. Este estudio puede realizarse a través de métodos visuales, graficando el resultado de un modelo contra el otro, e identificando una distribución no correlacionada entre los puntos. Con el fin de analizar si pudiese existir una solución conformada por la fusión entre modelos, y así mejorar el desempeño de las topologías consideradas, se analizará la complementariedad entre los modelos 'WDSB', 'CNN', y 'Wide&Deep', y se testeará el desempeño a través de las métricas descritas en la sección siguiente. Al verificarse su complementariedad, los mismos serán fusionados a través de uniones lógicas 'AND', 'OR', y a través de la 'SUMA' y normalización, de los vectores a la salida de cada modelo. Los modelos que se ensayarán serán dos, y están conformados por la unión 'WDSB-CNN' y 'WDSB-Wide&Deep'.

3.10. Métricas de medida

3.10.1. Métricas utilizadas en aplicación de detección de fraudes - NTLs

Dada la complejidad del problema, al ser un problema de clases desbalanceadas donde los datos que contienen anomalía son la clase minoritaria (5 % del total de muestras), es necesario emplear y combinar el conjunto de métricas adecuado y recomendado para este tipo de aplicaciones. Entre los métodos de medida del performance recomendados en problemas de detección de fraudes se encuentran:

- Matthews Correlation Coefficient (MCC)
- Fmeasure (Precision, Recall)
- Curva ROC y el área bajo la misma
- Evolución del valor de la función de pérdida calculada según la fórmula de la entropía cruzada para casos binarios (Binary Cross-Entropy)

Matthews Correlation Coefficient (MCC)

Si bien, la métrica 'MCC' no será considerada para evaluar los resultados, cabe destacar su utilidad para evaluar la performance en problemas de clasificación binaria. El 'MCC' es un valor que se calcula en base a los elementos de la matriz de confusión descripta arriba. El valor se calcula a partir de la siguiente formula:

$$MCC = \frac{(TP \times TN - FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.4}$$

El valor del 'MCC' varía entre -1 y 1, siendo 1 el valor resultante de una perfecta predicción y -1 el caso de una incorrecta predicción de todos los valores.

Los diferentes ensayos serán evaluados considerando las métrica 'Fmeasure', curvas 'ROC' y 'Precision-Recall', las cuales se describen a continuación.

Binary Cross-Entropy

La función de costo que se utiliza en este caso de clasificación binaria es la entropía cruzada (binaria). Sea p_i la probabilidad real (1 ó 0) de pertenecer a una clase, y $\hat{p_i}$ la probabilidad de pertenecer a la clase predicha, la entropía cruzada se calcula de la siguiente forma:

$$BCE = p_i log(\hat{p_i}) - (1 - p_i) log(1 - \hat{p_i}) \tag{3.5}$$

La misma decrece hasta cero a medida que la predicción se vuelve más exacta. Para determinar que tan bien evoluciona y responden los modelos empleados en el problema de clasificación de fraudes, será también tenido en cuenta la evolución de la función de costo en la etapa de test y entrenamiento.

3.10.2. F-measure

En las pruebas a realizar se considerará la siguiente métrica como medida de exactitud por ser la más apropiada para problemas de este tipo, donde las clases están desbalanceadas. La misma permitirá medir la performance en clasificación entre clases minoritarias y mayoritarias independientemente. Las ecuaciones a emplear para el cálculo de la misma son las siguientes:

$$Recall^p = \frac{TP}{TP + FN} \tag{3.6}$$

$$Recall^p = \frac{TP}{TP + FN} \tag{3.7}$$

$$Recall^n = \frac{TN}{TN + FP} \tag{3.8}$$

$$F_{value} = \frac{(1 + \beta^2) Recall^p \times Precision}{\beta^2 Recall^p + Precsision} \tag{3.9}$$

$Recall^p$ es la proporción de muestras positivas clasificadas correctamente (clientes que hacen fraude). La 'Precision' representa la proporción de muestras clasificadas como positivas que efectivamente lo son. La combinación de ambas es representada por el valor de 'F_value' (según se indica en la ecuación 3.9), el cual representa la media geométrica entre ellas, ponderadas por el parámetro 'β', que en este caso se tomara =1 (f1-score). El parámetro 'β' permite darle un peso relativo mayor al 'precision' sobre el 'recall' o viceversa. Por ejemplo dependiendo de los recursos humanos disponibles, puedo necesitar elegir un 'β' que priorice el 'precision' aunque disminuya el 'recall'.

3.10.3. Curvas ROC y Precisión-Recall

Entre las diferentes métricas para la evaluación del performance existen herramientas gráficas que también ayudan a entender la eficacia del modelo, un ejemplo es la curva o el análisis 'ROC' y la curva de 'Precision-Recall'. Ambos métodos toman en consideración el umbral o 'threshold' de decisión utilizado para evaluar si una muestra se clasifica como positiva o negativa. Este valor se fija entre 0 y 1 y se aplica sobre el resultado de probabilidades de pertenecer a una u otra clase a la salida del modelo.

- Curva ROC: El análisis o curva ROC es un ploteo de la tasa o proporción (true positive rate) de muestras clasificadas como "verdaderas-positivas" (eje-y) vs muestras "falsas positiva" (eje-x) evaluadas para cierto número de

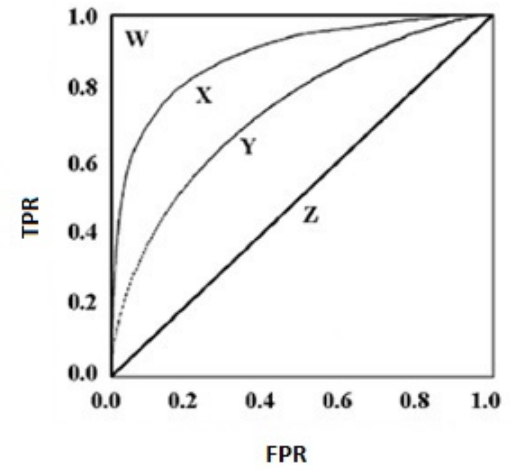

Figura 3.11: Área bajo la curva ROC.

umbrales ('thresholds') de decisión. El área bajo la curva ROC (AUC, 'area under the curve') es una medida de que tan bien se está distinguiendo entre los dos grupos. En la figura 3.11 se observa un ejemplo de dicha curva para 4 casos diferentes, en el ejemplo la gráfica muestra para $AUC(Z) = 0{,}5 < AUC(Y) = 0{,}65 < AUC(X) = 0{,}8 < AUC(W) = 1$ (caso ideal).

- Curva Precision-Recall: La curva es una gráfica de la precisión (eje-y) vs el valor de recall (eje-x) para diferentes umbrales de decisión.

En los experimentos del capítulo 5, los resultados serán evaluados por el área bajo la curva ROC (AUC).

Capítulo 4

NTLs - Caracterización y Síntesis

4.1. Resumen

En el presente capítulo se detallan las características de los fraudes a considerar, los parámetros que modelaran la realidad de los mismos, su implementación y generación sintética a partir de la base de datos disponible. Antes de generar los fraudes en la base de datos, es necesario el preprocesado de la misma, procedimiento que se explica en el presente capítulo. Finalmente, se presenta la base de datos con la que se trabajará en los ensayos, sus características y preprocesado sobre la misma. Se incluye también una descripción de como se propuso incluir la información de las subestaciones en el modelo 'Wide&Deep'.

4.2. NTLs en redes inteligentes

4.2.1. Tipo de fraudes - NTLs

Se conoce como fraude eléctrico a la acción que lleva a cabo el usuario de un servicio de suministro eléctrico para alterar la medida real de su contador, el mismo está asociado con la pérdida no técnica y su fin es no pagar parte o la totalidad de la energía que consume el cliente. Estos tipos de fraude se generan tanto en consumidores domésticos como industriales, esto se da en todo tipo de clases sociales, en aquellas con ingresaos medios y altos, como en hogares de ingresos bajos.

Los diferentes tipos de fraudes están mayormente caracterizados por el punto físico donde se comete el mismo. En este sentido el fraude puede establecerse:

- Antes del contador, este es el caso en el que se manipula el cableado aguas arriba del contador creando un 'bypass' que conecte directamente al usuario con la red eléctrica externas sin pasar por el contador. Esta práctica es muy común en viviendas familiares o en instalaciones empresariales, además de ser una práctica ilegal, este tipo de acciones incrementa el riesgo de accidentes con consecuencias graves tanto para la salud del infractor como para la integridad del domicilio (incendios, explosiones, etc).

- Dentro del contador, existen diversas técnicas que permiten la manipulación del mismo. En el caso particular de contadores analógicos, las técnicas más conocidas son el cambio en la posición normal de lectura, agregado de componentes electromagnéticos o cualquier otro elemento que impida el correcto movimiento de partes mecánicas. Para el caso de los medidores inteligentes (digitales) pueden aparecer técnicas más complejas como ser, entre otras, la inyección de código malintencionado que permita 'hackear' la lógica de medida del contador.

- En el interruptor de control de potencia (ICP), la manipulación del ICP corresponde a aquellos fraudes donde el cliente busca un consumo de potencia superior al contratado. En la mayoría de casos al tratarse de contadores inteligentes el ICP se incluye en el propio contador.

- En el suministro de otros usuarios, este es el caso en el que el infractor realiza una conexión a su domicilio desde la conexión de un vecino, pasando de esta manera su consumo por el contador del usuario fidedigno.

- En el suministro a servicios comunes, similar al caso anterior, en este caso la conexión que hace el defraudador es sobre la red que alimenta a los servicios comunes de un edificio, complejo, etc.

4.2.2. Consecuencias económicas

Los usuarios fraudulentos perjudican no solo a la empresa que suministra la energía eléctrica sino que sus actos tienen también repercusión en el resto de la sociedad que no comete fraude. Al no pagar por lo que se consume, los clientes mal intencionados no limitan ni optimizan su consumo energético generando pérdidas que impactan tanto en la rentabilidad del negocio eléctrico como en el medio ambiente, además de influir directamente en el aumento de las tarifas. Otra víctima de los fraudes cometidos es el estado, quien no percibe el retorno esperado por el porcentaje del consumo eléctrico que va grabado de impuestos, por lo que el retorno real a la cuenta del estado es menor al previsto.

Otra consecuencia directa de la actividad fraudulenta es el impacto en los elementos tanto de generación eléctrica como en la red de distribución ya que para el dimensionado de los mismos se tiene en cuenta únicamente el consumo eléctrico legal.

4.3. Bases de datos disponibles

Para evaluar los desempeños de los modelos a considerar en la detección de fraudes, es necesario contar con una base de datos confiable y fidedigna de lecturas de consumos de clientes reales. Debido a motivos de privacidad y seguridad las compañías proveedoras de energía eléctrica se han mostrado remisas a hacer

públicas sus bases de datos. Esto ha derivado en un reto adicional para los investigadores dando lugar en algunos casos la creación de datos sintéticos para pruebas. Sin embargo en los últimos 5 años, han aparecido algunas bases de datos del consumo a pie del cliente y de libre acceso bajo carácter de anonimato o semi-anonimato. Estas no presentan dimensiones mayores a las 10.000 muestras y en su mayoría faltan o son escasos los datos que representan casos de fraude.
Algunas de las bases de datos disponibles se muestran en la Tabla I del artículo [35]. En la misma se representa de manera esquemática el uso que se le ha dado a cada una de estas bases dentro del área de la investigación y análisis de datos de consumidores eléctricos. Allí se identifican los datasets "Customer Behavior Trials", "Low Carbon London" como las más interesantes por la característica de granularidad de muestreo (resolución de 30 minutos) y cantidad de muestras por encima de las 5000. Estas bases de datos surgen con los siguientes fines:

- Base de datos CER - Customer Behavior Trials: es un proyecto de medida energética inteligente lanzado por la Comisión Reguladora de Energía de Irlanda (CER - ISSDA) en el que se proponen capturar las conductas de los consumidores para determinar cómo el sistema de medidores inteligentes puede ayudar a caracterizar los perfiles de los mismos según diversidad demográfica, estilos de vida, y tamaño de los domicilios.

- Low Carbon London: este proyecto, de similares características que el anterior, involucra a 5000 domicilios dentro del área de Londres. Los datos son recolectados con el objeto de investigar el impacto sobre un amplio rango de tecnologías de energía renovable que operan dentro de la red de distribución de energía eléctrica de Londres.

A partir de los bancos de datos mencionados, se encontraron diferentes casos de uso que sirvieron como puntos de partidas, permitiendo identificar publicaciones que abordan diferentes tipos de problemas, como ser detección de fraudes, compresión de datos y extracción de características, predicción de carga, clustering según clase sociodemográfica del cliente, caracterización del cliente, y estimación de la carga pico en ciudades.
Si bien ambas bases de datos fueron utilizadas en diversidad de aplicaciones, como las ya mencionadas, éstas presentan un detalle común, y es que no presentan muestras anómalas. Dada la escasa disponibilidad de bases de datos que incluyan fraudes y la dificultad para conseguirlas, se decidió incluir una etapa de modelado y generación de fraudes sintéticos. Para ello, se identificará en detalle los tipos de fraudes a modelar y sus características, finalmente se incluirán dichos comportamientos en la base de datos sin fraudes. Las bases de datos resultantes quedarán disponibles para entrenar y evaluar los algoritmos estudiados en la tesis y para trabajos a futuro.

4.3.1. Base de datos CER: detalles técnicos

Los trabajos y ensayos realizados en este articulo son sobre la base de datos 'CER', la misma es una base de datos de consumos energéticos reales tomados sobre

redes con medidores inteligentes, y proviene de la entidad 'Irish Social Science Data Archive (ISSDA)', organismo que registra datos que surgen del proyecto 'Customer Behavior Trials' llevado a cabo por la Comisión Reguladora de Energía (CER) en Irlanda. Los clientes que participaron de esta prueba dieron su conformidad para formar parte del proyecto, por lo que se consideró razonable asumir que estas lecturas no contienen fraudes (asumiendo que pertenecen a clientes honestos). Esta base de datos contiene un gran número de clientes con diferentes características, cuyos datos fueron relevados por un largo periodo de tiempo, por estas razones la base de datos 'CER' se la considera una excelente fuente de investigación en el área de análisis de datos de medidores inteligentes.

En este caso, y gracias al apoyo del Instituto de Ingeniería Eléctrica de la Facultad de Ingeniería (UDELAR), se logró obtener la base de datos CER. Los datos contenidos en la misma son utilizados para modelar los patrones normales de consumo eléctrico. La base de datos CER presenta las siguientes características:

- Tiempo de adquisición: 1 año y medio
- Granularidad: 30 minutos
- x3 Columnas: client_id, [día] [tiempo] (3digitos+2digitos, ej:'00130'), consumo eléctrico (kWh)
- Total clientes utilizados: 6000
- Total medida / cliente: 25730

Los datos crudos son presentados por separado en 6 archivos (File[i].txt.zip con i=1 a 6) formato de texto con la información indicada arriba y con las siguientes características:

- Datos ordenados cronológicamente (según fueron adquiridos) y no por cliente
- Presencia de datos intermedios faltantes
- Datos con fechas de referencia incorrectas
- Las lecturas de un mismo cliente están distribuidas entre los 4 archivos entregados
- Cada cliente presenta un numero variable de lecturas tomadas
- No cuenta con datos fraudulentos

Dada la forma en la que están presentados los datos es necesario incluir una etapa de pre procesamiento de los mismos con el objeto de:

- Eliminar datos faltantes
- Ordenar los mismos en filas de clientes

- Eliminar datos con fechas incorrectas

Antes de generar la nueva base de datos con fraudes sintéticos es necesario preprocesar la base de datos CER. El objetivo es presentar los datos matricialmente, donde el histórico de consumo de cada cliente se dispone a lo largo de cada fila. De esta manera, cada fila es considerada como una serie temporal, que será alterada o no según el criterio de inyección de fraudes sintéticos.

4.4. Preprocesado de datos

A continuación se describe el procedimiento empleado para el preprocesado de la base de datos utilizada ('CER'), la misma será acondicionada para su posterior manipulación en la etapa de agregado de fraudes sintéticos.

4.4.1. Preprocesado de base de datos CER

El preprocesado de los datos fue realizado con un módulo creado en Python que reordena los consumos por cliente en el formato deseado, resultando en una matriz con la siguiente distribución:

$$Col_0 : client_{id}, Col_1 : lect_1, Col_2 : lect_2, ..., Col_i : lect_i, \ldots, Col_n : lect_n.$$

La misma matriz se guarda en una variable de la clase 'dataframe' perteneciente a la librería 'pandas'. Las etiquetas de los datos válidos para cada columna van de '19548' a '73001' (aproximadamente 1 año y medio de datos), donde los primeros 3 dígitos representan los 'días' y los últimos dos dígitos las tramas temporales de "media hora" (1 a 48).

A medida que se van integrando las lecturas de cada cliente al nuevo 'dataframe', las mismas se 'limpian' de manera de eliminar fechas incorrectas, interpolando las medidas faltantes, y rellenando con 0's a los extremos de la fila (es decir para los instantes anteriores a la primer muestra de un cliente y los instantes posteriores a la última muestra tomada del mismo cliente). Como producto se obtiene el 'dataframe' completo con los datos preprocesados aún sin fraudes sintéticos. La base de datos resultante es guardada en archivo 'process_data_CER' en formato 'pickle'.

4.4.2. Procesado de base de datos CER para incorporar lecturas de subestaciones

Con el fin de incorporar la información del balance de lecturas entre subestaciones - consumidores a partir de la base CER (sin considerar las pérdidas técnicas (l_{t_i})), se procedió a simular una arquitectura de distribución de subestaciones agrupando consumidores de a grupos de 200. Siendo el resultado final, una matriz (en formato de 'dataframe') del tamaño de la base de datos CER (preprocesada), donde para cada grupo de 200 consumidores, cada fila (asociada a cada cliente)

corresponde al vector de balance de cargas de la subestación (que contiene a esos 200 clientes). De este modo cada cliente (fila) asociado a una misma subestación, contiene el mismo vector que resulta del calculo del balance total de la subestación que los contiene (para todo instante t).
Considerando entonces las lecturas $p_{t_i,n}$ del cliente n contenido en la base de datos CER sin inyección de fraude, y llamando $pf_{t_i,n}$ a la lectura del mismo cliente n contenido en la base de datos CER con el 14 % de fraudes sintéticos, el calculo del balance para la subestación 'k-ésima' es de la forma:

$$\frac{\sum_{m=1+200\times(k-1)}^{200\times k}\left(p_{t_i,m}-p_{ft_i,m}\right)}{\sum_{1+200\times(k-1)}^{200\times k} p_{t_i,m}} = c_{kt_i} \tag{4.1}$$

Entonces el vector asociado a cada uno de los 200 clientes pertenecientes a la subestación 'k-ésima' para 'm' instantes de tiempo, esta formado por los elementos

$$[c_{kt_1}, c_{kt_2}, ..., c_{kt_i}, ..., c_{kt_m}].$$

El procesado necesario para armar la base de datos con el balance de las subestaciones se aborda en el código de los archivos 'CER_subs200.ipynb' y 'subs_diff.ipynb', cuyo resultado son los datos guardados en el dataframe 'process_data_CER_subs.pkl'.

4.5. Generación sintética de NTL's

Tal como se menciona anteriormente la base de datos considerada no contiene datos con características de fraude, por lo tanto será necesario implementar un módulo que inyecte fraudes del tipo NTLs en la base de datos original. Este modulo es parte fundamental del ensayo debido a su vital importancia en la calidad del modelo que se obtendrá, y su desempeño para detectar fraudes en redes inteligentes. Esto obliga a considerar un número considerable de variables a fin de emular un comportamiento fraudulento lo mas cercano a la realidad posible. Logrando una base de datos con fraudes simulados, estaremos en condiciones de entrenar y testear nuestro modelo con dicha base.

4.5.1. Características de fraudes NTL's

Para poder emular el patrón de comportamiento de los usuarios fraudulentos se recurrió a consultar directamente con el departamento de Tecnologías de la Información de la compañía estatal de energía eléctrica del Uruguay - UTE (administración nacional de Usinas y Trasmisiones Eléctricas). Luego de diversas reuniones e investigación de datos históricos se concluye que el comportamiento típico de fraude se puede clasificar principalmente en dos grupos:

- i- fraudes permanentes en el tiempo (a partir de un momento dado).
- ii- fraudes periódicos, mayoritariamente producidos durante la noche.

Particularmente se estima que los fraudes tipo (ii) se dan en aquellas familias donde el accionamiento se hace luego del horario de retorno habitual de los empleos, por lo que se supone exista un accionamiento manual con el cual el usuario comete el fraude en el momento que le es mas conveniente. A diferencia del tipo (i), normalmente ideado para empresas del ramo industrial o domicilios donde el consumo es alto durante todo el día, sin poder separarse en franjas de mayor o menor consumo.

Ambos tipos de fraudes presentan la particularidad de alterar la lectura del consumo de energía por un multiplicador que lleva la lectura real de la misma a un valor considerablemente mas bajo (pudiendo ser nulo).

4.5.2. Modelado de fraudes - NTL's

Además de la información recabada de la investigación y reuniones, también se consultaron los artículos académicos que trataran los tipos de fraudes que mas se adaptaran al comportamiento identificado. Particularmente el articulo [38] clasifica los fraudes en 5 grupos (considerando el primero de ellos como el estado normal) de acuerdo a la siguiente tabla:

Estado	Función de estado	Descripción
s_1	$p_{t_i,n} = 1xp^*_{t_i,n}$	Medidor Inteligente no comprometido ni en falta
s_2	$p_{t_i,n} = vp^*_{t_i,n}$; $v \in (0,0.95) \cup (1.05,2.5]$	Medidor Inteligente comprometido/falta permanente
s_3	$p_{t_i,n} = \delta_{t_i} p^*_{t_i,n}$; $\delta_{t_i} = \begin{cases} v \text{ si comienzo } \leq t_i \leq fin \\ 1 \text{ en caso contrario} \end{cases}$ Con v definido como en caso s_2	Medidor Inteligente comprometido/falta solo por cierto período de tiempo en el día
s_4	$p_{t_i,n} = \eta_{t_i} p^*_{t_i,n}$; $\eta_{t_i} = \begin{cases} 0 \text{ si comienzo } \leq t_i \leq fin \\ 1 \text{ en caso contrario} \end{cases}$	Medidor Inteligente envía medida nula solo por cierto período de tiempo en el día
s_5	$p_{t_i,n} = mediana(p^*_n)$	Medidor inteligente reporta la mediana de las lecturas reales

Figura 4.1: Descripción de tipos de fraudes considerados [38].

Dónde,

- $p^*_{ti,n}$ es la potencia medida del cliente (o smart meter) 'n', medida en el instante 't_i'
- v representa una fracción de la medida:
  1. $\nu \in (0,0{,}95)$
  2. $\nu \in (1{,}05, 2{,}5)$
- $\delta_{ti,n}$ y $\eta_{ti,n}$ se definen en la tabla

En el presente trabajo se plantea generar cada uno de los fraudes descriptos en la Tabla 4.1 tomando las siguientes consideraciones para el modelado del comportamiento fraudulento:

Primera consideración: Parámetros,

- % de fraude sobre la base: se considera un 14 %
- rango de disminución: entre 0.3 y 0.7. Selección sobre una distribución uniforme para los consumidores seleccionados.
- rango de duración (fraude tipo 3 y 4): 2 a 6 hs
- variación de uso: máx 1h
- momento de ocurrencia en el día (fraudes tipo 3 y 4): min 17hs - max 23h
- olvido (fraudes tipo 3 y 4): días en que el fraude no ocurre
- % de olvido (son los días en que el fraude no ocurre para fraudes tipo 3 y 4): min 5 % max 30 %)

Segunda consideración: En discusión y a propuesta del equipo de investigación del proyecto que surge del convenio IIE(UDELAR)-UTE, se resolvió modelar la naturaleza de los accionamientos (variables anteriores) de acuerdo a las distribuciones de probabilidad siguientes,

- Comienzo del fraude: Es el instante a partir del cual la serie temporal comienza a contener fraude. Este valor es asignado utilizando una distribución de probabilidad uniforme dentro del intervalo completo de la serie de datos, $C_i \sim U(d_{min}, d_{max})$.
- Inicio de accionamiento: Variable aleatoria con distribución gaussiana, $I \sim N(\mu_{ini}, \sigma_{ini})$. Modela el ruido del horario en que cada día se acciona el sistema de llaves. El valor medio ($\mu_{ini}[i]$) es fijo a lo largo de los días y es asignado a cada cliente utilizando una distribución de probabilidad uniforme $\mu_{ini} \sim U(t_{min}, t_{max})$.
- Duración de accionamiento: Variable aleatoria con distribución gaussiana $I \sim N(\mu_d[i], \sigma_d)$. Modela el ruido del horario en que cada día se acciona el sistema de llaves. El valor medio $\mu_d[i]$)es fijo a lo largo de los días y es asignado a cada cliente utilizando una distribución de probabilidad uniforme $\mu_d \sim U(t_{min}, t_{max})$.
- Olvido: Variable aleatorio con distribución de probabilidad Bernoulli, $O \sim Be(p)$. Modela la probabilidad de no cometer fraude un día.

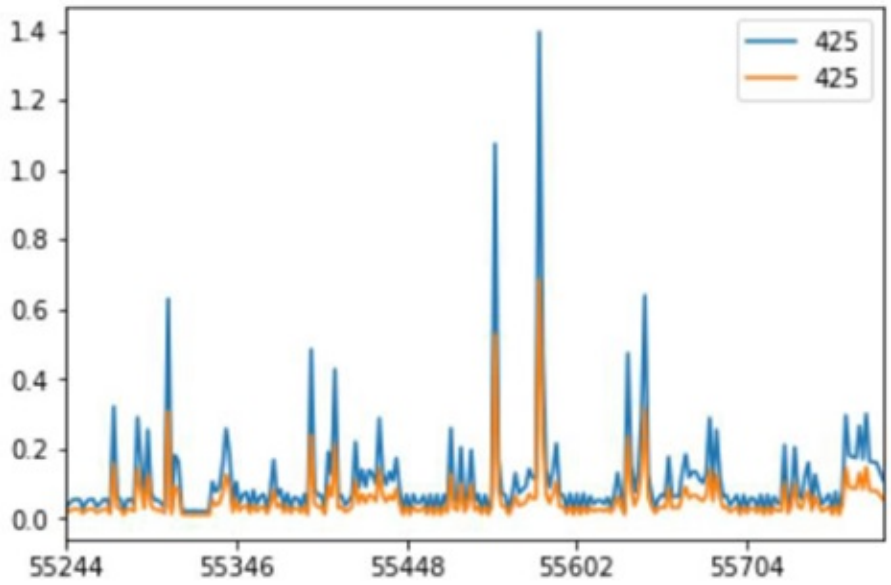

Figura 4.2: Fraude tipo 2 - curva consumo (kwh) vs lecturas.

4.5.3. Generación de fraudes en base de datos CER

Tomando las consideraciones anteriores se generó cada uno de los fraudes del tipo 2 al 5 obteniendo de este modo la base de datos de frecuencia 30 minutal llamada "process_data_CER_f14" en formato "pkl". Recordando que uno de los objetivos del articulo es evaluar qué ocurre con el desempeño del modelo al aumentar la granularidad de los datos, contar con una base de datos de frecuencias 30 minutal (CER con fraude) será de gran utilidad para cubrir el cometido.

Debido a que se está considerando el modelo 'Wide&Deep' del articulo de referencia [39], el cual se ensaya originalmente con la base de datos SGCC (State Grid Corporation of China [1]) de frecuencia diaria, es necesario pasar la base de datos CER con fraudes ('process_data_CER_f14') también a frecuencia diaria. Esto permitirá generar una base similar a la del articulo [39], logrando tener un punto de referencia que permita comparar el desempeño entre el modelo con la base de fraudes sintéticos 'CER' y la base 'SGCC' (utilizada en [39]). Se llama entonces 'process_data_CER_f14_daily' a la base de datos original (con fraudes sintéticos) llevada a frecuencias diarias.

A continuación se muestran ejemplos gráficos de las curvas de consumo resultantes al inyectar un tipo de fraude determinado en la base de datos 'CER' para frecuencias 30 minutales.

[1]State Grid Corporation of China http://www.sgcc.com.cn/

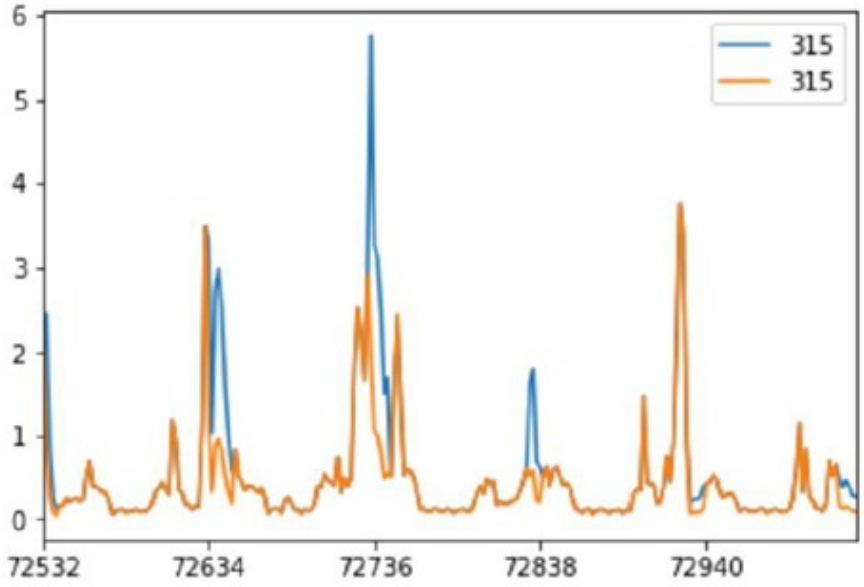

Figura 4.3: Fraude tipo 3 - curva consumo (kwh) vs lecturas.

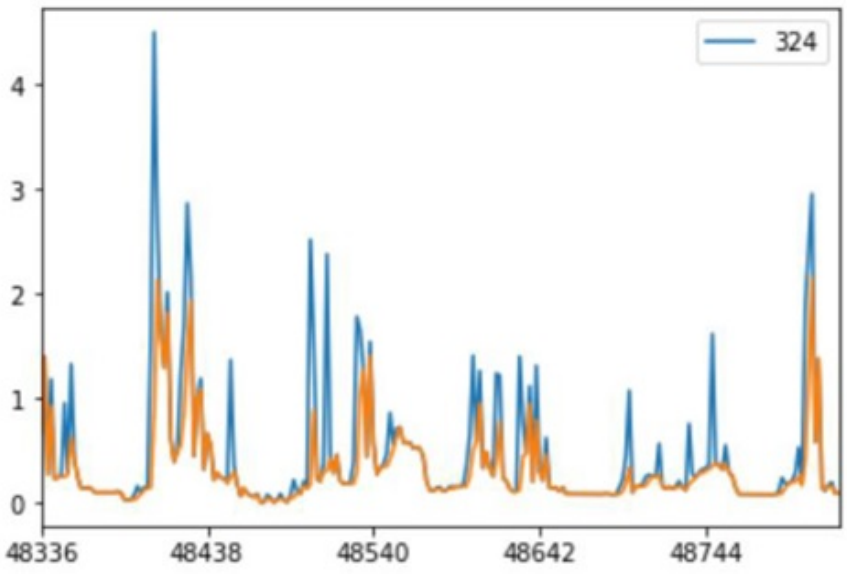

Figura 4.4: Fraude tipo 5 - curva consumo (kwh) vs lecturas (con n=4, siendo "n" el ancho de ventana para el filtro de mediana).

4.6. Bases de datos generadas

Antes de entrar en el capítulo de experimentos, es útil hacer una síntesis de las bases de datos que fueron generadas para ensayar los modelos propuestos. La tabla 4.6 muestra las bases de datos que se ponen a disposición, y que son resultado del preprocesado de la base de datos original y posterior generación de fraudes a diferentes frecuencias.

1. 'process_data_CER.pkl': preprocesado de base de datos original (4.4.1).
2. 'CER_f14_30.pkl': base de datos preprocesada con generación de fraudes en el 14 % de los clientes y frecuencia original.
3. 'CER_f14_1.pkl': base de datos (2) llevada a frecuencias diarias.
4. 'CER_subs.pkl': simulación de lecturas de subestaciones (según 4.4.2), obtenida a partir de base de datos (1) y (2)

Todas las bases de datos descriptas mantienen la misma cantidad de clientes, en particular las bases de datos 1, 2, y 4 son de frecuencia 30 minutal (muestras por cliente: 25725), mientras que la base de datos 3 es de frecuencia diaria (muestras por cliente: 538). Otra particularidad es que las bases de datos que presentan fraudes (2, 3, y 4) mantienen la misma proporcionalidad para cada tipo de fraude, es decir el 14 % de los clientes presentan fraudes y los mismos se reparten en iguales cantidades para cada tipo. Las bases de datos anteriores serán utilizadas en el siguiente capítulo para evaluar el desempeño de los modelos presentados en el Capítulo 3 según las métricas consideradas en 3.10.

4.6.1. Bases de datos real adquirida: UTE-DB

Para este trabajo se dispuso de la base de datos real suministrada en el marco del proyecto que deriva del convenio de colaboración UTE-IIE(UDELAR). La base de datos cuenta con lecturas obtenidos de la nueva red de medidores inteligentes instalada por UTE para uso de sus clientes. La información detallada de la base de datos en cuestión se presenta a continuación:

- Tiempo de adquisición: 1 mes
- Granularidad: 15 minutos
- Total clientes: 2912
- Total fraudes: 121
- Total medida / cliente: 2880

Debido a que la base recibida es de frecuencia 15 minutal, fue necesario el pre-procesamiento de la misma para pasar a granularidad 30 minutal. Para hacer el agregado se usó la librería de 'dataframes' de panda, obteniendo como producto final una base de datos con consumos y fraudes reales de granularidad 30 minutal (1440 medidas por cliente).

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 5

Experimentos y resultados

5.1. Resumen

Para evaluar el desempeño de los 3 modelos considerados (Wide&Deep, Fully-connected, y Deep 'CNN') se desarrollaron una serie de experimentos por etapas. Iniciando cada uno con la búsqueda de hiperparámetros para cada modelo (tomando entradas de diferentes resoluciones), y continuando con etapas de validación y test. Se evaluará también como influye el uso de técnicas de regularización por 'Dropout' y de compensación para problemas de clases desbalanceadas. Se continuará por la evaluación del desempeño del modelo 'Wide&Deep' tomando en su entrada 'Wide' la información (simulada) de las subestaciones, resultando en el modelo 'Wide&Deep-sb'. El resultado de cada ensayo será presentado por modelo y por tipo de fraude, considerando las métricas vistas anteriormente (3.10). Finalmente se evaluará la capacidad de complementariedad entre los modelos 'CNN', 'Wide&Deep' , y 'Wide&Deep_sb' (3.9.1). El capitulo culminará con un ensayo de Test para los mejores modelos encontrados. Los trabajos y ensayos realizados en este capítulo son sobre la base de datos 'CER_f14_30' (frecuencia 30 minutal) y 'CER_f14_1' (frecuencia diaria) presentada en el capítulo anterior (4.6).

5.2. Ensayos propuestos

Antes de estudiar el desempeño en test de los modelos presentados en el capítulo 3, se realizarán diversos ensayos de búsqueda de parámetros, con el fin de asegurar un punto de operación óptimo para cada uno de ellos. Las arquitecturas de los modelos son construidas utilizando la librería TensorFlow, en particular su API de alto nivel llamada 'Keras', la misma es utilizada para construir y entrenar modelos de aprendizaje profundo. En tal sentido, cada modelo será entrenado en un conjunto de 5148 muestras de entrenamiento (con relación 80-20 % train-validation), y 1287 muestras para test. El espacio de muestras considerado pertenece a las bases de datos 'CER_f14_1' (frecuencia diaria) y 'CER_f14_30' (frecuencia 30 minutal), según corresponda en cada caso. Luego de encontrar el punto óptimo de funcionamiento para cada modelo, se finalizará cada ensayo con una prueba de validación.

Finalmente se presentará un ensayo de test para los mejores modelos encontrados. Los ensayos que se desarrollarán a lo largo del capítulo son los siguientes:

1. Modelo 'Wide&Deep':
   - Ensayo 1: entrenamiento y validación con base de datos 'CER_f14_1' (frecuencia diaria). (ver sección 5.3)
   - Ensayo 2: entrenamiento y validación con base de datos 'CER_f14_30' (frecuencia 30 minutal). (ver sección 5.4)
   - Ensayo 3: estudio desempeño con 'dropout', con base de datos 'CER_f14_1' y 'CER_f14_30'. (ver sección 5.5)
   - Ensayo 4: estudio desempeño con balance de clases ('class-weight'), con base de datos 'CER_f14_1' y 'CER_f14_30'. (ver sección 5.6)
2. Modelo 'Fully-connected' (ver sección 3.5):
   - Ensayo 5: entrenamiento y validación con base de datos 'CER_f14_30' (frecuencia 30 minutal).
3. Modelo CNN (ver sección 3.6):
   - Ensayo 6: entrenamiento y validación con base de datos 'CER_f14_30' (frecuencia 30 minutal).
4. Modelo Wide&Deep-sb (ver sección 3.9.1):
   - Ensayo 7: entrenamiento y validación con base de datos 'CER_f14_30' (frecuencia 30 minutal), comparación resultados contra modelo 'CNN' estándar (ver sección 3.6), y 'Wide&Deep' original.
   - Ensayo 8: Análisis de complementariedad y fusión entre modelos 'Wide&Deep-sb'(ver sección 3.9.1), 'CNN', y 'Wide&Deep' original.
5. Desempeño en conjunto de Test:
   - Ensayo 9: entrenamiento de los modelos con muestras de entrenamiento y validación, y prueba de desempeño con muestras de test (base de datos 'CER_f14_30').
6. Prueba base de datos CER con histórico de 1 mes:
   - Ensayo 10: entrenamiento del modelo 'Wide&Deep' con base de datos 'CER_f14_30'.

5.3. Ensayo 1: Wide&Deep - Búsqueda parámetros y validación con base de datos CER de frecuencia diaria

Para evaluar el desempeño del modelo con lecturas de frecuencia diaria se utilizará la base de datos 'CER_f14_1'. La misma es generada a partir de la original (de frecuencia 30 minutal), tal como se indica en el capítulo 4. Para identificar el punto de funcionamiento óptimo del modelo se realizó un ensayo del tipo 'grid-search', el cual permite identificar los parámetros que generalizan mejor el problema (evitando el sobreajuste y limitando su varianza), y a su vez obtener un valor de AUC los mas cercano a 1 posible.
El modelo es entrenado siguiendo una grilla con los siguientes valores y parámetros:

- batch-size 32, 64, y 128
- learning-rate entre 0 y 0.1 (arreglo aleatorio de largo 5).

Del experimento de búsqueda de parámetros pudo apreciarse un sobreajuste del modelo (con muestras diarias) a partir de la época 25 (en la mayoría de los casos). Las curvas del 'loss' del modelo durante el entrenamiento se observan en la figura 5.1, en la misma se aprecia la evolución de los valores de 'loss' en 'train', y 'validation' durante el entrenamiento.

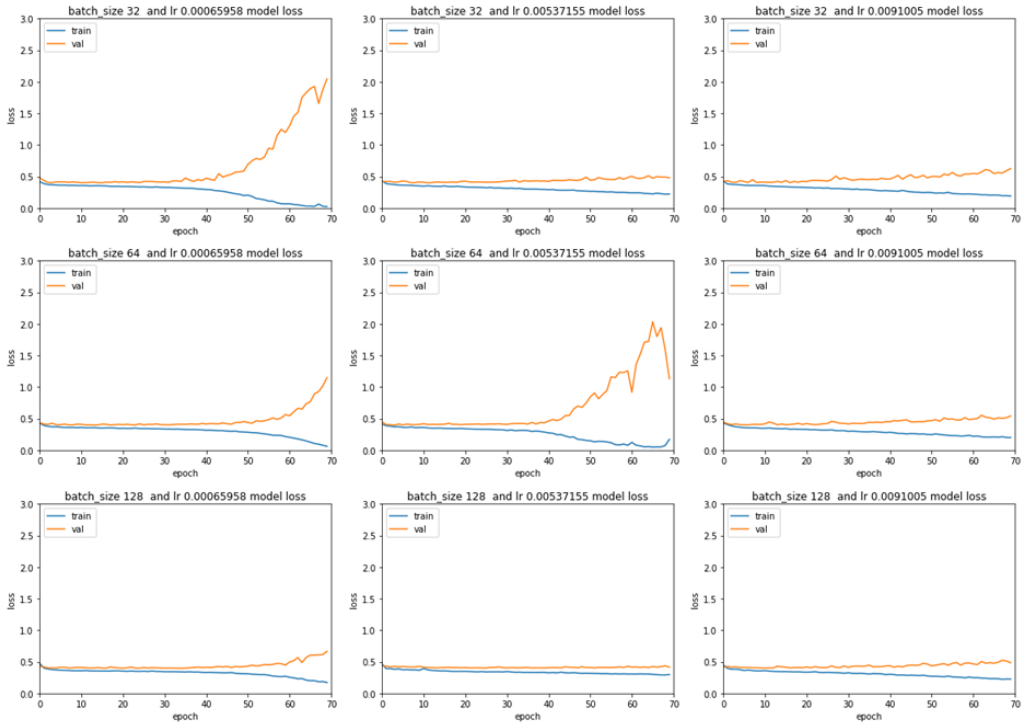

Figura 5.1: Ensayo1 - curvas de 'Loss' durante el entrenamiento (70 epochs).

Si se observa la curva de la evolución del AUC 5.2 para bacth-size de 64, se puede concluir que el modelo obtiene un buen desempeño para la base de datos

CER (con frecuencia diaria) para los parámetros de batch-size 64 y un learning-rate de 0.00066 ó 0.0091. En la imagen 5.2 puede verse cómo se comporta el valor del 'AUC', de 'validation', y 'train' durante el entrenamiento para un valor de batch-size 64.

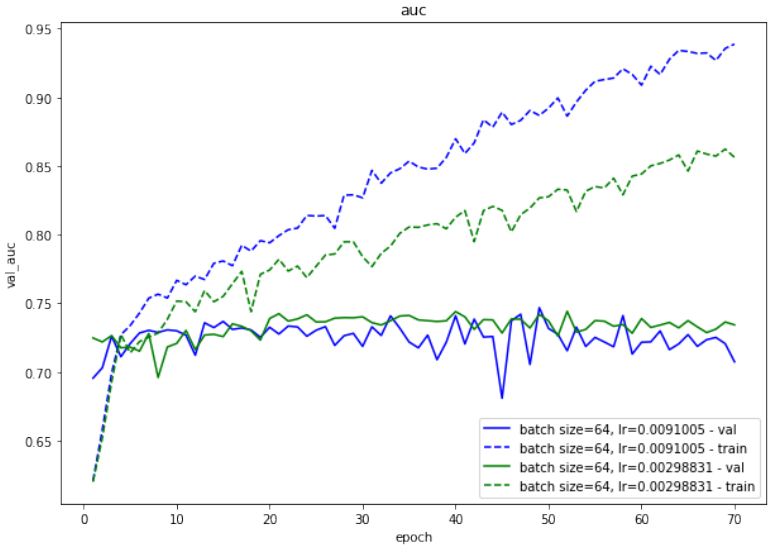

Figura 5.2: Ensayo1 - curva AUC de validación durante el entrenamiento (frecuencias diarias).

Al observar las curvas de 'loss' para batch-size igual a 32 (figura 5.1), se observa como el modelo comienza a sobreajustar a las muestras de entrenamiento a partir de las épocas 30. Considerando esto, puede afirmarse que se obtiene un balance óptimo de 'loss-AUC' en un punto cercano a las 25 para los tres valores de 'learning-rate' (imagen 5.2). Podría ser interesante también observar la evolución de la curva 'ROC' durante el entrenamiento, del área bajo su curva deriva la métrica del AUC.

5.3.1. Ensayo 1: curva ROC

La figura 5.3 muestra la evolución de la curva 'ROC' durante el entrenamiento, el comportamiento del modelo es evaluado en los puntos:

- batch-size: 64
- épocas: 25, y 250
- learning-rate: 0.00537

Entrenado el modelo correspondiente para los 2 instantes mencionados (25 y 250 épocas), las curvas de ROC obtenidas se observan en la imagen (5.3).

Rápidamente se verifica lo que indican las curvas de evolución del AUC, cuando el modelo comienza a sobreajustarse, el AUC disminuye y esto se refleja en la forma aplanada de la curva ROC (figura de la derecha en imagen 5.3). A continuación

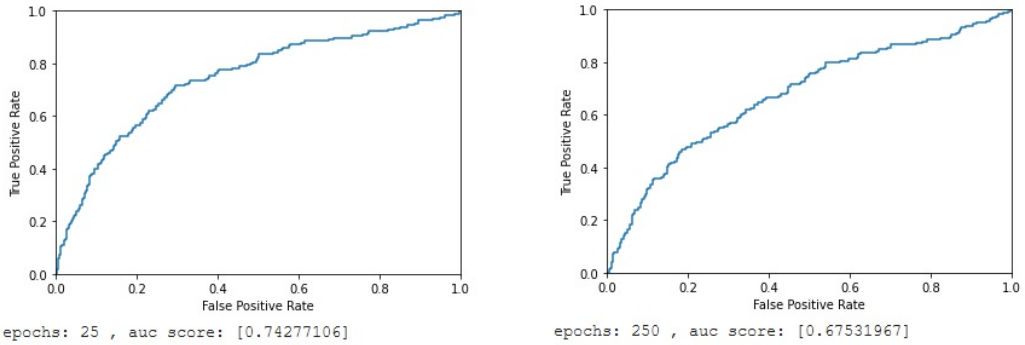

Figura 5.3: Ensayo1 - curva ROC para modelo con muestras de frecuencias diarias.

se observará cuál es el valor del AUC en los puntos óptimos identificados, y en particular se observará cómo responde el modelo para los diferentes tipos de fraude.

5.3.2. Ensayo 1: prueba con datos de Validación

Para validar el desempeño se toman los parámetros en los que el modelo responde satisfactoriamente, y se evaluará el mismo sobre un conjunto de muestras de validación (extraído de la base de datos 'CER_f14_1'). Los resultados siguientes corresponden al ensayo para los valores óptimos de 'batch-size', 'learning-rate', y número de épocas. Los mismos se presentan en la tabla 5.1.

Batch-size	epochs	Adam (lr)	AUC
32	6	0.00066	**0.72**
64	9	0.00131	**0.71**
128	6	0.00131	**0.71**

Tabla 5.1: Ensayo1 - resultados con datos de frecuencias diarias.

Ensayo 1: Curvas precision-recall y f1-score

Las curvas de 'precision-recall' y 'f1-score' dan una idea del desempeño del modelo en la detección de fraudes. La imagen 5.4 muestra cómo se comporta el modelo (que toma muestras diarias del subset de validación) en la detección de los diferentes tipos de fraudes simulados, para todo el rango del umbral de decisión. Las curvas de las gráficas 5.4 y 5.5, es el resultado que obtiene el modelo al ser entrenado de acuerdo a los parámetros del punto de operación considerado (bs: 128, lr: 0.00131, epochs: 6).

Analizando las curvas 'precision-recall' y 'f1-score' se puede observar el desempeño del modelo para cada tipo de fraude, y en particular se destaca la complementariedad entre ambas curvas. Es decir, observando las curvas para todos los tipos de fraude, por ejemplo el 'precision-recall', puede identificarse un intervalo

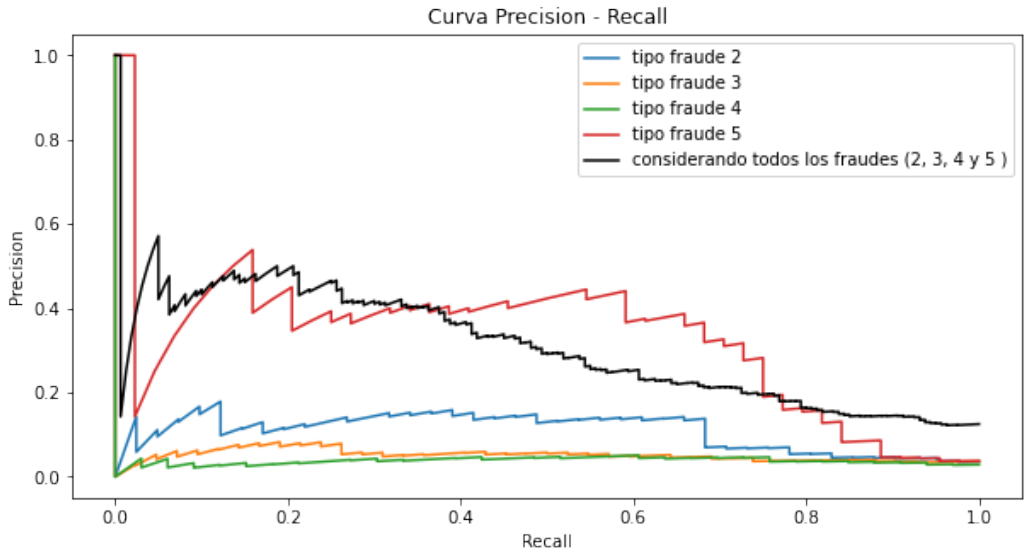

Figura 5.4: Ensayo1 - Curva precision-recall para cada fraude (frecuencias diarias).

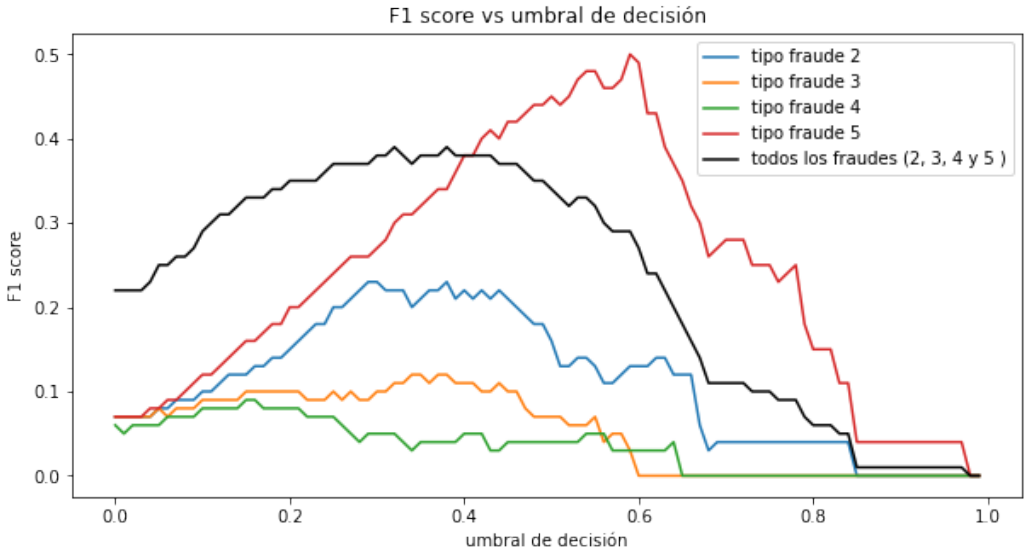

Figura 5.5: Ensayo1 - f1-score vs umbral de decisión para cada fraude (frecuencias diarias).

óptimo alrededor del punto [0.2, 0.4], mientras que en la curva del 'f1-score' puede identificarse un punto óptimo alrededor del umbral de decisión 0.35. Siguiendo el razonamiento anterior, parece lógico identificar el umbral óptimo de decisión en 0.35 (basándose en el valor de 'f1-score'), e identificar luego los valores de 'precision' y 'recall' en ese punto. Otra característica notoria en ambas curvas es la coherencia que presenta el comportamiento del modelo para cada tipo de fraude, siguiendo la misma tendencia en todo el umbral de decisión.
Considerando las respuestas del modelo frente a los diferentes tipos de fraude

(muestras de frecuencias diarias), puede afirmarse que su desempeño es mejor para el fraude tipo 5 (se computa la mediana entre lecturas), seguido de fraudes tipo 2 (lectura se multiplica por valor $\nu \in (0,0{,}95)$). Particularmente, los fraudes del tipo 5 y 2 son los del tipo permanentes, es decir una vez computado el fraude, el mismo se mantiene permanente en el tiempo.

5.4. Ensayo 2: Wide&Deep - Búsqueda parámetros y validación con base de datos CER de frecuencia 30 minutal

En este caso se realiza un segundo experimento (ensayo 2) sobre el modelo 'Wide&Deep', con la diferencia que en este caso se toman como muestras de entrada la base de datos 'CER_f14_30' (a su frecuencia original de 30 minutos).

El ensayo comienza con una búsqueda del tipo de 'grid-search' sobre los mismos parámetros del ensayo anterior. Las curvas de 'loss' obtenidas en este caso se presentan en la imagen 5.6. En las mismas se aprecia la evolución del desempeño del modelo durante su entrenamiento para cada punto.

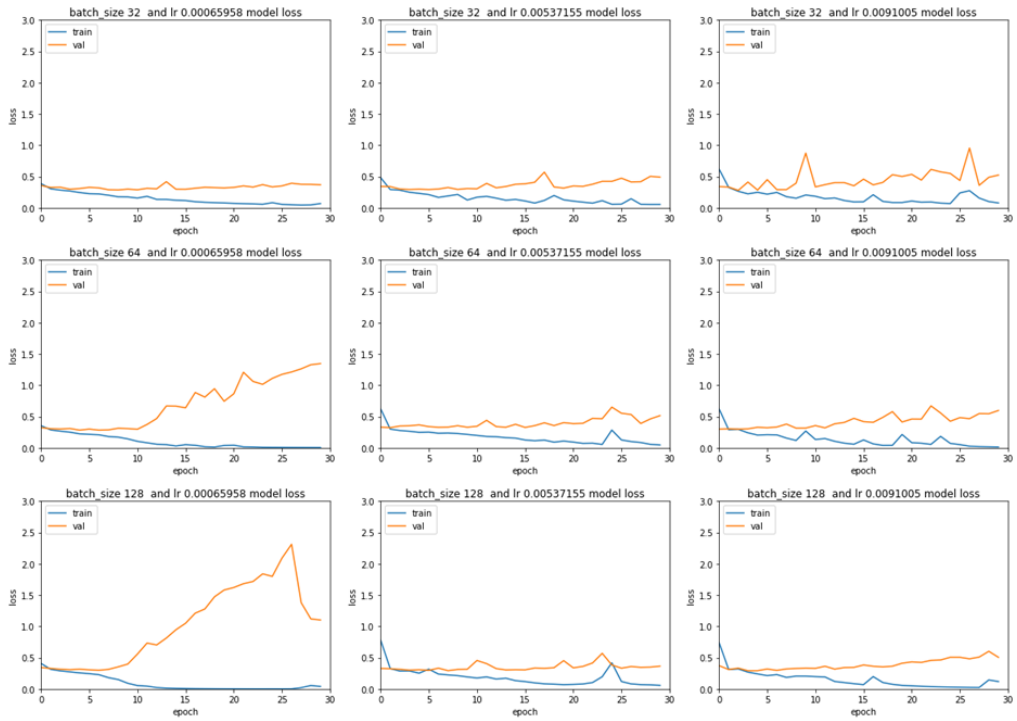

Figura 5.6: Ensayo2 - curvas de Loss durante el entrenamiento (20 epochs para frecuencia 30 minutal).

Al igual que en el caso anterior, la métrica tomada como criterio para seleccionar el mejor modelo es el 'AUC'. La figura 5.7 muestra la curva de entrenamiento

del 'AUC' (para batch-size de 64) durante el entrenamiento.

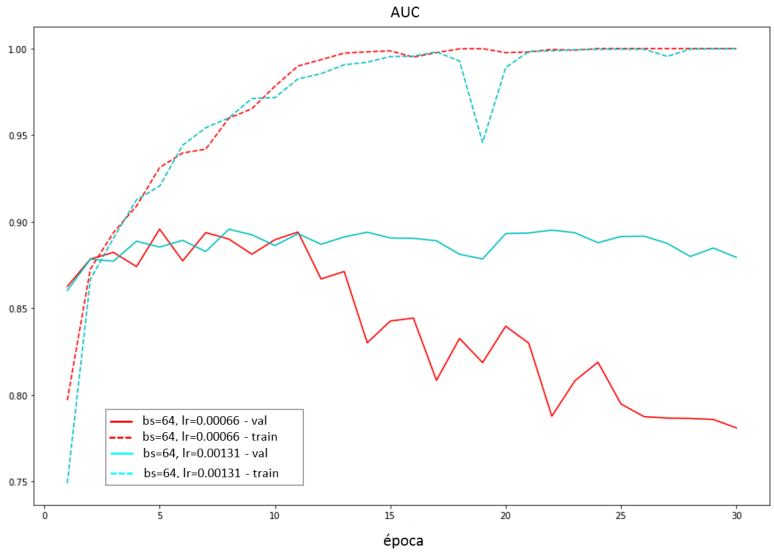

Figura 5.7: Ensayo2 - comportamiento del AUC de validación durante el entrenamiento (frecuencia 30 minutal).

A través de la observación conjunta de los gráficos de 'loss' y 'AUC', se pueden identificar los puntos de funcionamiento que optimizan el modelo. Considerando entonces la gráfica de 'AUC', se tiene que para los tres valores de batch-size, el AUC de validación es cercano a 0.90 entre las época 5 y 7 para valores de 'learning-rate' de 0.00131 y 0.000659. Sin embargo, es necesario observar la evolución del 'loss' durante el entrenamiento para identificar a partir de que momento el sistema comienza a sobre-entrenar, y ubicarse así antes de la zona de sobreajuste. En este sentido, para los valores de 'learning-rate' antes mencionados, se observa que el modelo comienza a sobreajustarse a partir de la época 6 para 'learning-rate' de 0.00131, lo que es notorio a simple vista si se observa el caso para batch-size de 64. Para un batch-size de 64, una buena configuración de parámetros sería con 'learning-rate' de 0.00066 y 7 épocas. A continuación se presentan las curvas ROC correspondientes, las mismas reflejan 2 etapas diferentes del aprendizaje durante el entrenamiento.

5.4.1. Ensayo 2: Curvas ROC

En este caso las diferencias entre curvas 'ROC' durante el entrenamiento serán menos notorias, ya que los valores de AUC son mayores en una etapa temprana del entrenamiento. En la siguiente imagen (ver fig 5.8) se muestra la evolución de la curva 'ROC' para 2 etapas contiguas de entrenamiento. La imagen representa la evolución de la curva para el modelo evaluado en los puntos:

- batch-size: 128

- épocas: 8, y 250
- learning-rate: 0.00066

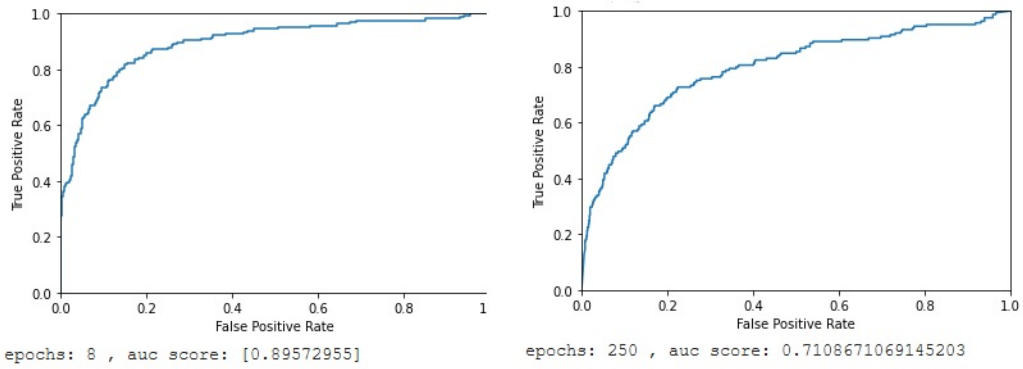

Figura 5.8: Ensayo2 - curva ROC (frecuencias 30 minutales).

Al igual que en el caso anterior, se verifica lo que indican las curvas de evolución del AUC, a medida que el modelo se sobreajusta, comienza a aplanarse la curva ROC (figura de la derecha en imagen 5.3) bajando de esta manera el valor de AUC.

5.4.2. Ensayo 2: desempeño con datos de validación

De manera similar al caso de frecuencias de muestras diarias, es necesario realizar una ronda de pruebas de Validación para evaluar el desempeño del modelo en los puntos óptimos encontrados, además se observarán las curvas 'precision-recall' y 'f1-score '. El ensayo se realizó para cada valor de 'batch-size', tomando 'learning-rate' y número de épocas que se consideran óptimos en cada caso. Los resultados obtenidos para un experimento de entrenamiento y validación se presentan en la tabla 5.2.

Batch-size	epochs	Adam (lr)	AUC
32	7	0.00298	**0.88**
64	7	0.00066	**0.88**
128	5	0.00066	**0.88**

Tabla 5.2: Ensayo2 - resultados con datos de frecuencias 30-minutales.

Para observar en detalle cómo responde el modelo para cada tipo de fraude se analiza a continuación las curvas asociadas a las métricas 'precision-recall' y 'f1-score' para todo el umbral de decisión.

Ensayo 2: Curvas 'precision-recall' y 'f1-score'

A continuación se observará el impacto que tiene el cambio en la frecuencia de las muestras de entrada al momento de detectar los diferentes tipos de fraudes.

Recordando el caso de muestras de frecuencia diaria, se vió como las curvas de 'precision-recall' y 'f1-score' indican una mejor desempeño del modelo para fraudes del tipo 5 y 2, en el entorno del umbral de decisión 0.35, se analizará entonces cómo afecta el cambio de frecuencia de las muestras en los valores mencionados (precision-recall y f1-score). Al igual que en el caso anterior se tomará el punto con el que se realizó el ensayo de evolución de la curva ROC para frecuencias 30 minutales. Luego de entrenado el modelo, el comportamiento obtenido frente a los diferentes tipos de fraudes se ve reflejado en las siguientes imágenes (5.9 y 5.10).

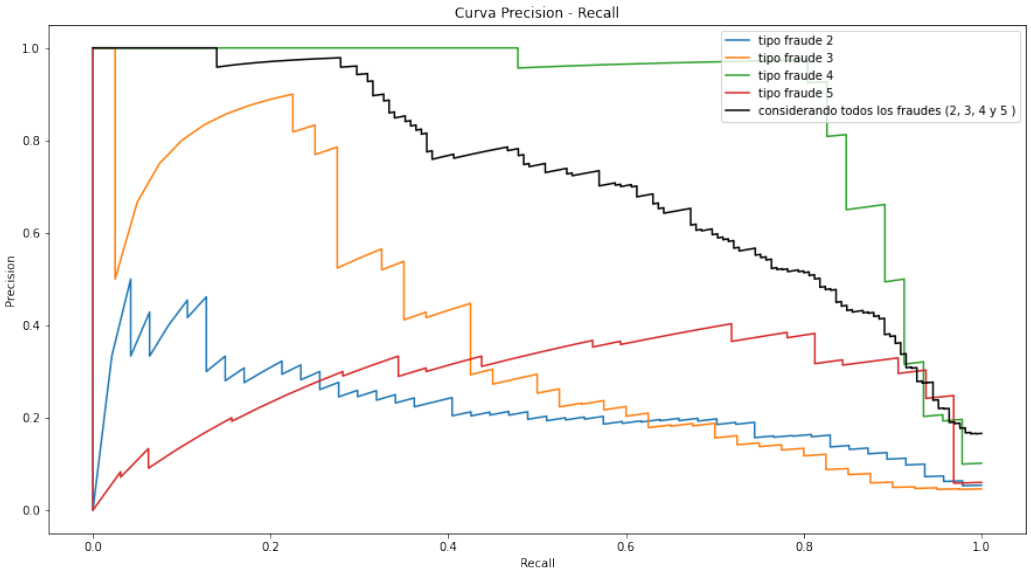

Figura 5.9: Ensayo2 - Curva precision-recall para cada fraude - modelo con muestras de frecuencias 30 minutales.

Analizando las curvas de 'precision-recall' y 'f1-score' obtenidas en este caso (frecuencia de lecturas 30 minutales) se distingue el traslado del intervalo de umbral de decisión. En el caso de muestras diarias el pico de 'f1-score' se da en 0.4, en este caso se amplia el rango aceptable, quedando éste entre [0.1 - 0.4]. Se observa también un incremento en el desempeño del modelo frente a fraudes del tipo 3 y 4. Se deduce en este caso que el pasar a muestras (lecturas) de mayor granularidad permite dotar al modelo de mayor capacidad para detectar este tipo de fraudes, sin perder capacidad de detección para fraudes del tipo 5. Particularmente, los fraude del tipo 4 son en los que el modelo obtiene mejor desempeño (el fraude se computa durante una ventana de tiempo 2-6hs y la lectura computada durante el fraude tiene valor 0).

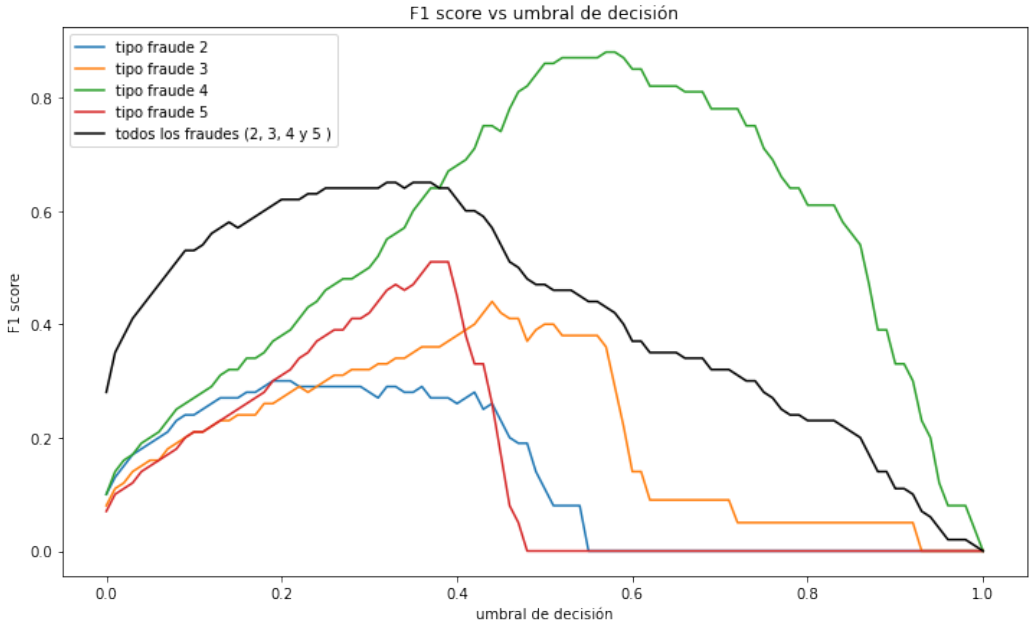

Figura 5.10: Ensayo2 - f1-score vs umbral de decisión para cada fraude - modelo con muestras de frecuencias 30 minutales.

5.5. Ensayo 3: Wide&Deep - Estudio desempeño con regularización por Dropout

En este experimento se evalúa cómo incide la técnica de regularización por 'dropout' (3.8) en el modelo 'Wide&Deep'. Para analizar cómo incide la misma en las diferentes frecuencias se tomará como entrada la base de datos 'CER_f14_1' (frecuencia diaria), y luego en un segundo experimento la base de datos 'CER_f14_30' (frecuencia 30 minutal). Ambos experimentos se realizan sobre un espacio de muestras de entrenamiento y validación fraccionados de igual manera que en los ensayos anteriores.

5.5.1. Ensayo 3: Prueba de Validación con técnica Dropout

Antes del ensayo propuesto se procedió con la búsqueda de parámetros óptimos para lograr el balance deseado de 'loss-AUC' y realizar la prueba de validación en un punto de buen desempeño. En las tablas 5.3 y 5.4 se muestran los resultados obtenidos en los puntos de mejor desempeño encontrados para los casos de frecuencia diaria y 30-minutal respectivamente.

Observando la tabla anterior, puede verse un leve incremento en el valor de AUC para muestras de frecuencias diarias y valores de batch-size 64 y 128. En el caso de muestras de frecuencias 30 minutales el modelo no logra incrementar sus valores mas allá de los obtenidos para el caso sin 'dropout'. Del ensayo realizado también se observó que agregar 'dropout' evita el sobreajuste temprano del modelo,

Batch-size	epochs	Adam (lr)	AUC
32	10	0.00298	**0.71**
64	14	0.0091	**0.73**
128	14	0.0091	**0.73**

Tabla 5.3: Ensayo3: Resultados con datos de frecuencias diarias tomando el modelo original con dropout.

Batch-size	epochs	Adam (lr)	AUC
32	7	0.00066	**0.89**
64	10	0.00066	**0.88**
128	10	0.00066	**0.88**

Tabla 5.4: Ensayo3 - resultados con datos de frecuencia 30-minutal tomando el modelo original con dropout.

permitiendo que el AUC tenga valores mas constantes a medida que avanza el entrenamiento.

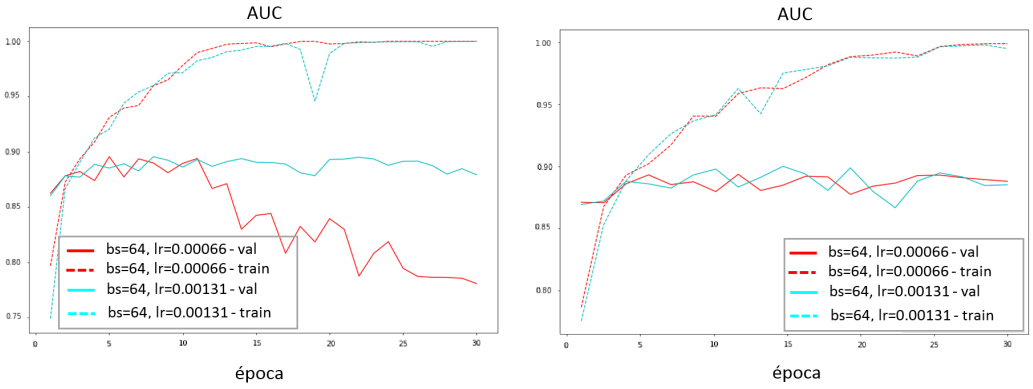

Figura 5.11: Ensayo3 - curva AUC durante el aprendizaje sin y con técnica dropout (a la derecha) para frecuencias 30 minutales.

5.6. Ensayo 4: Wide&Deep - Estudio desempeño con balance de clases

En esta sección se aborda el problema de clases desbalanceadas a través de la introducción de pesos a las muestras (ver 3.8.1). El estudio de desempeño se realizará tanto para muestras de frecuencias diarias como 30 minutales. Ambos experimentos se realizaron sobre un espacio de muestras de entrenamiento y validación fraccionados (80-20) de igual manera que en los ensayos anteriores.

Batch-size	epochs	Adam (lr)	AUC
32	9	0.00131	**0.72**
64	10	0.00066	**0.72**
128	15	0.00131	**0.72**

Tabla 5.5: Ensayo 4 - resultados con datos de frecuencias diarias, tomando el modelo original con balance de clases.

Batch-size	epochs	Adam (lr)	AUC
32	8	0.00066	**0.88**
64	7	0.00298	**0.88**
128	12	0.0091	**0.88**

Tabla 5.6: Ensayo 4 - resultados con datos de frecuencia 30-minutal, tomando el modelo original con balance de clases.

5.6.1. Ensayo 4: Prueba de Validación con balance de clases

El desempeño del modelo con balance de clases frente a muestras de validación, en los puntos óptimos encontrados, se presenta en las tablas (5.5 y 5.6).

Los resultados del experimento muestran como, para el caso de frecuencias diarias y de igual modo que en el caso con 'dropout', la técnica con balance de clases también logra mejorar levemente el valor de AUC. En el caso de frecuencias 30 minutales, los resultados son similares al resto de los modelos (con y sin 'dropout').

5.7. Ensayo 5: Red 'Fully-connected' - Entrenamiento y validación con base de datos CER 30 minutal

De lo observado en ensayos anteriores del modelo 'Wide&Deep', a partir de las curvas 'precision-recall' y 'f1-score', se había concluido el buen desempeño que presenta el mismo en reconocer fraudes del tipo 2, 4 y 5. Con el objeto de evaluar el desempeño de la red 'fully-connected' (aislada del modelo completo) frente a los diferentes tipos de fraude, se realizará una prueba de validación, lo que permitirá evaluar el desempeño de la misma.

5.7.1. Ensayo 5: prueba de Validación

Luego de una etapa de búsqueda de hiperparámetros, se prueba el desempeño del modelo con el conjunto de muestras de validación de la base de datos 'CER_f14_30' (30 minutal). Los resultados obtenidos, en los puntos óptimos de funcionamiento encontrados se presentan en la tabla 5.7.

Como puede observarse en los resultados obtenidos (ver tabla 5.7) se aprecia una baja en el desempeño respecto al modelo completo ('Wide&Deep'), lo cual

Batch-size	epochs	Adam (lr)	AUC
32	6	0.00298	**0.86**
64	5	0.00131	**0.85**
128	7	0.00298	**0.86**

Tabla 5.7: Ensayo 5 - resultados en validación con red fully-connected (frecuencia 30 minutal).

es de esperar ya que hay información presente que ya no es capturada por la componente 'Deep' ('CNN'), como ser el comportamiento periódico del cliente.

5.7.2. Ensayo 5: Curvas de precision-recall y f1-score

Las imágenes 5.12 y 5.13 muestran las curvas resultantes de 'precision-recall' y 'f1-score' obtenidas en este caso.

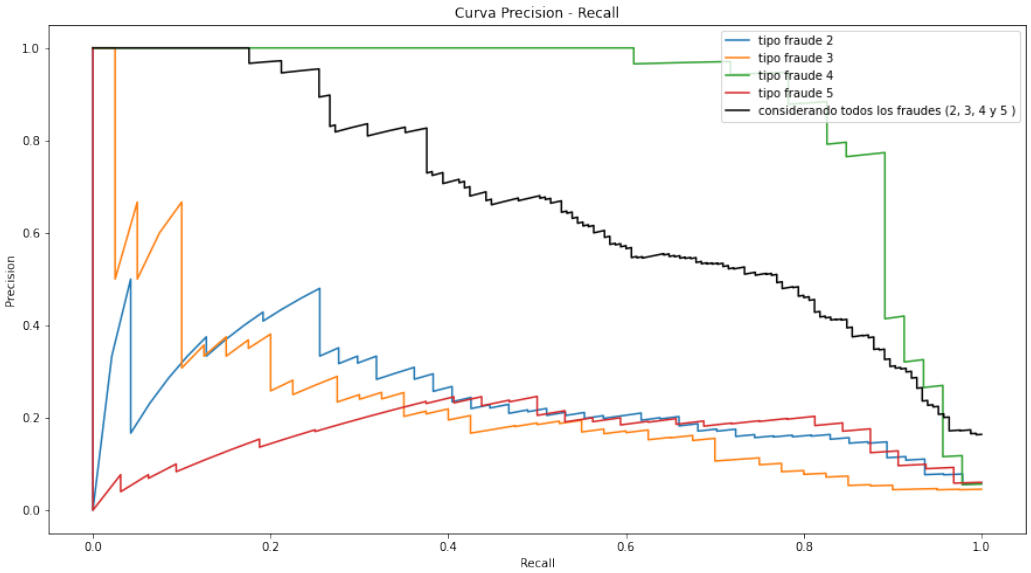

Figura 5.12: Ensayo 5 - curva precision-recall para cada fraude - modelo 'fully-connected' (wide) con muestras de frecuencias 30 minutales

Los resultados obtenidos sugieren un buen desempeño de la red 'Wide' en la detección de los fraudes del tipo 4. Particularmente se presenta una desmejora en el 'f1-score' de los fraudes del tipo 5 frente al modelo original, con un marcado corrimiento del 'thresshold' (umbral de decisión) en este tipo de fraudes (tipo 5) . Otra peculiaridad es el punto umbral donde se da el pico de 'f1-score' (alrededor de 0.2). Finalmente cabe destacar como en el entorno de 'Recall' entre [0, 0.4] se da una pérdida de 'score' de precisión para el tipo de fraude 3 (4.3). Este ultimo resultado refleja cómo la red densa es menos efectiva a las inyecciones de fraudes localizados (cometidos durante una ventana de tiempo acotada).

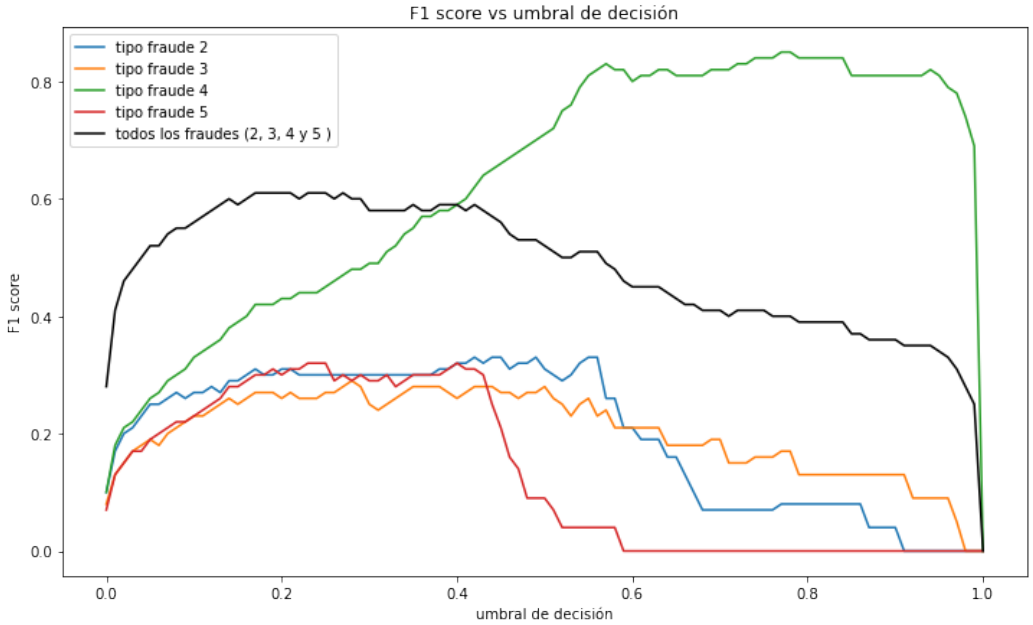

Figura 5.13: Ensayo 5 - f1-score vs umbral de decisión para cada fraude - modelo 'fully-connected' (wide) con muestras de frecuencias 30 minutales.

5.8. Ensayo 6: Red CNN - Prueba de desempeño con base de datos CER de frecuencia 30 minutal

Con el objeto de evaluar el desempeño de la red 'CNN' por si sola, se estudiará a continuación, su capacidad para detectar cada tipo de fraude.

5.8.1. Ensayo 6: Red CNN - Prueba de validación

Una vez encontrados los hiperparámetros que mejor ajustan al modelo, se procede a correr el test de validación con muestras 30 minutales. La siguiente tabla (5.8) presenta los resultados obtenidos por el modelo 'CNN' en un ensayo de validación para los puntos de mejor funcionamiento encontrados.

Batch-size	epochs	Adam (lr)	AUC
32	4	0.00131	**0.86**
64	5	0.00066	**0.86**
128	8	0.00066	**0.87**

Tabla 5.8: Ensayo 6 - resultados en Validación con red CNN (fercuencia 30 minutal).

De la observación de resultados, y de igual modo que en el caso de la red 'Wide', puede deducirse que existe una pequeña pérdida en el desempeño (según

el AUC). Tal efecto es producto de la pérdida de la componente 'Wide', la cuál se encarga de capturar la información presente en la correlación entre muestras de la serie temporal de cada cliente.

5.8.2. Ensayo 6: Curvas de 'precision-recall' y 'f1-score' para la red CNN

Para observar en más detalle cómo se comporta el modelo en los puntos de funcionamiento encontrados, resulta útil mirar las curvas 'precision-recall' y 'f1-score'. Las mismas pertenecen al ensayo de validación anterior, efectuado en uno de los puntos óptimos de funcionamiento previamente encontrados. Luego de entrenado el modelo, el comportamiento obtenido frente a los diferentes tipos de fraudes se ve reflejado en las siguientes imágenes (5.14 y 5.15).

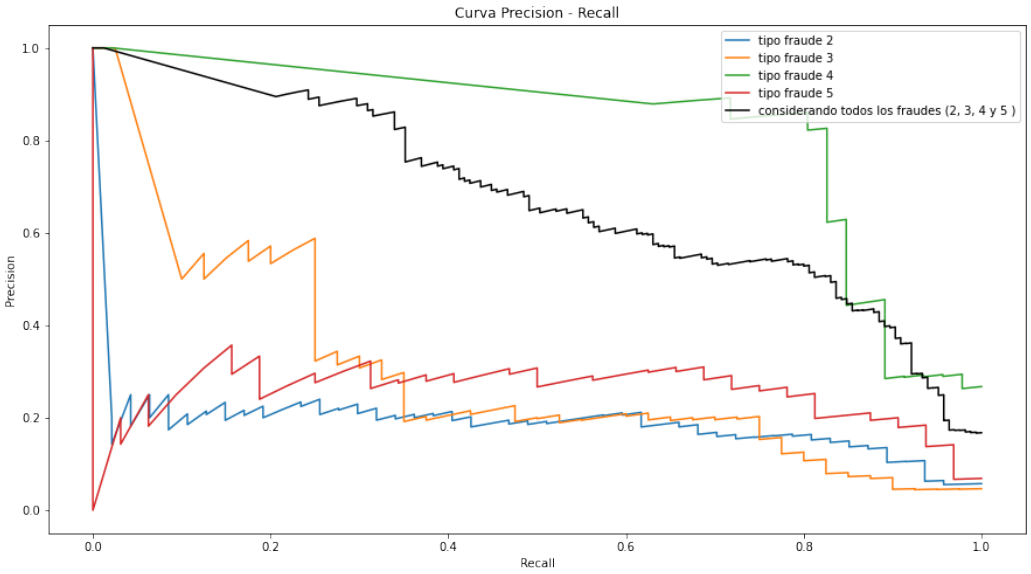

Figura 5.14: Ensayo 6 - curva precision-recall para cada fraude - modelo 'CNN' con muestras de frecuencias 30 minutales.

De los resultados obtenidos, se concluye que el modelo 'CNN' obtiene un comportamiento similar al del modelo completo (modelo original). En particular, se nota una mejoría (esperada) en la 'precision-score' de fraudes de tipo 3 (para 'Recall' menor a 0.3). A efectos de tener una mirada más objetiva, se presentará a continuación un análisis enfocado en la comparación de los mismos vistos hasta ahora.

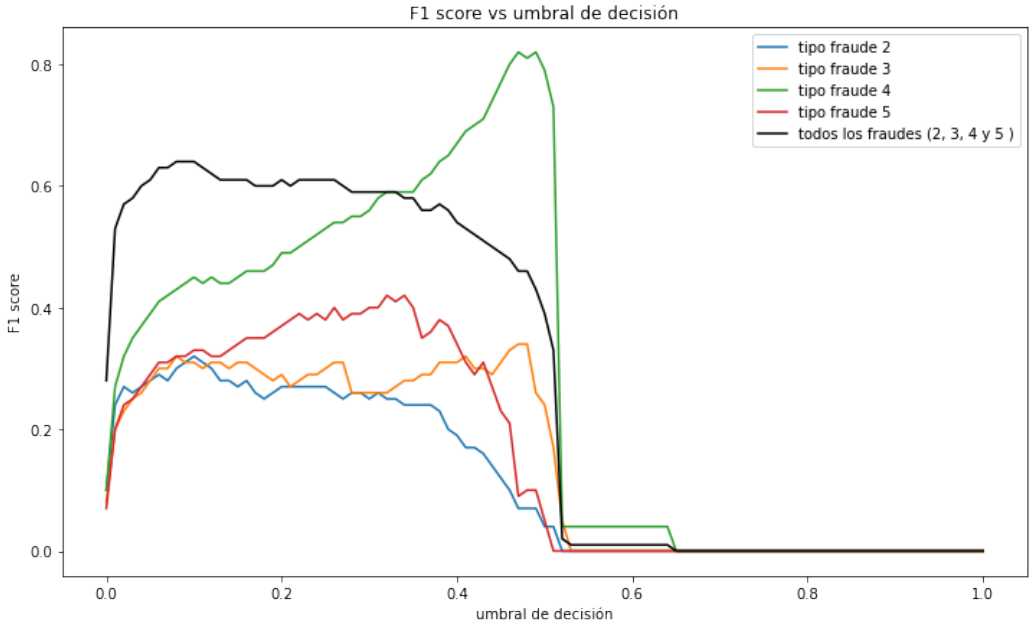

Figura 5.15: Ensayo 6 - f1-score vs umbral de decisión para cada fraude - modelo 'CNN' con muestras de frecuencias 30 minutales.

5.9. Comparativa entre modelos 1, 2, y 3 en la detección de fraudes sintéticos

Para obtener una visión mas amplia y clara del desempeño de los modelos vistos para cada tipo de fraude, se procederá a la realización de un ensayo en el que se entrenará y testeará (con muestras de validación) cada uno de los modelos (tomando entadas 30 minutales) en sus puntos óptimo de funcionamiento. La siguiente tabla (5.9) presenta los resultados obtenidos en los puntos de funcionamiento (umbral de decisión) dónde el 'f1-score' tiene su valor máximo.

De lo observado en las tablas, se deduce:

- Para todos los tipos de fraudes: el modelo 'Wide&Deep' es el que presenta mejor máximo de 'f1-score' y valor de 'Recall' y 'Precision' (en el punto donde se da el máximo 'f1-score').
- Para los fraudes tipo 2: el modelo 'Wide&Deep' es el que presenta mejor máximo de 'f1-score' y valor de 'Recall' y 'Precision' (en el punto donde se da el máximo 'f1-score').
- Para los fraudes tipo 3: considerando aquí el 'f1-score' como métrica preferencial del desempeño, el modelo 'Wide&Deep' y 'ConvNet' funcionan bien en este caso, sin embargo presenta menor 'Recall' que el modelo 'fully-connected'.
- Para los fraudes tipo 4: considerando el pico de 'f1-score', el mejor desem-

Modelo	max f1-score	Recall	Precision	AUC
Resultados en validación para **todos los tipos de fraude**				
Fully-connected	0.60	0.65	0.55	0.86
ConvNet	0.61	0.76	0.50	0.87
Wide&Deep	**0.64**	**0.76**	**0.55**	0.89
Resultados en validación para **tipo de fraude 2**				
Fully-connected	0.26	0.70	0.16	0.86
ConvNet	0.22	0.70	0.13	0.87
Wide&Deep	**0.30**	**0.74**	**0.18**	0.89
Resultados en validación para **tipo de fraude 3**				
Fully-connected	0.26	**0.48**	0.17	0.86
ConvNet	0.32	0.28	0.38	0.87
Wide&Deep	**0.44**	0.35	**0.58**	0.89
Resultados en validación para **tipo de fraude 4**				
Fully-connected	0.84	0.74	**0.97**	0.82
ConvNet	0.87	0.78	0.97	0.85
Wide&Deep	**0.87**	**0.85**	0.89	0.86
Resultados en validación para **tipo de fraude 5**				
Fully-connected	**0.55**	**0.84**	**0.40**	0.86
ConvNet	0.45	0.72	0.33	0.87
Wide&Deep	0.45	0.75	0.32	0.89

Tabla 5.9: Tabla comparativa desempeño en validación para **cada tipos de fraude**.

peño se da en el modelo 'Wide&Deep', superando también en la medida del 'Recall' al resto de los modelos.

- Para los fraudes del tipo 5: el modelo 'fully-connected' es el que mejor funciona (considerando el pico de 'f1-score').

Las imágenes 5.16 y 5.17 reflejan de manera gráfica cómo se comporta cada modelo según las curvas 'precision-recall' y 'f1-score' respectivamente.

Observando las mismas, se destacan las siguientes características:

- Los modelos 'ConvNet' y 'Wide&Deep' presentan los mejores valores de precisión entre [0.1, 0.7] del rango de 'recall' para fraudes del tipo 3.
- El modelo 'ConvNet' presenta menor precisión en un entorno de 'recall' de [0.6,0.9] para fraudes del tipo 4.
- El modelo 'fully-connected' presenta menor precisión que el resto de los modelos para los fraudes del tipo 5.

En cuanto al comportamiento de los modelos según el valor del 'f1-score', mirando las curvas presentes en la imagen (5.17) podría concluirse lo siguiente:

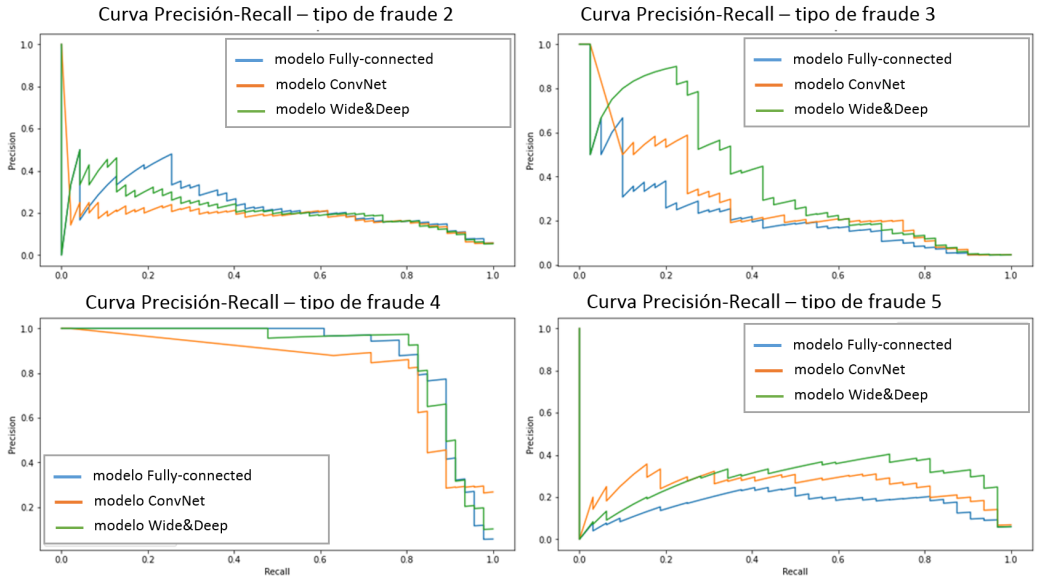

Figura 5.16: Curvas precision-recall de cada modelo con tipos de fraude 2, 3, 4, y 5.

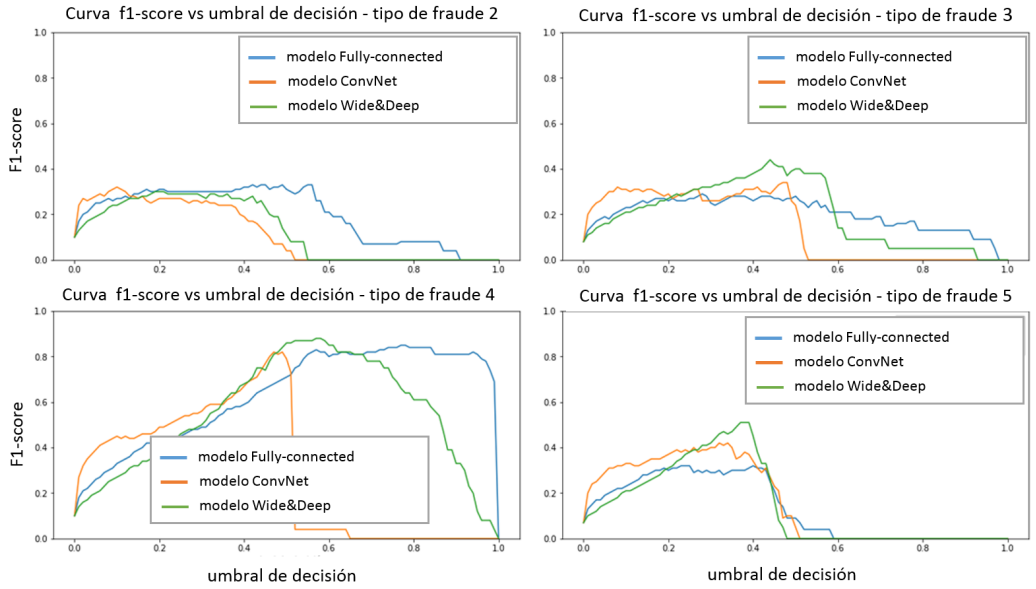

Figura 5.17: Curvas f1-score de cada modelo con tipos de fraude 2, 3, 4, y 5.

- Para fraudes del tipo 2, los 3 modelos presentan picos similares de 'f1-score'.
- El modelo 'Wide&Deep' presenta un leve margen de mejora en el valor de 'f1-score' frente al resto de los modelos para fraudes del tipo 3.
- Para fraudes del tipo 4, el modelo 'fully-connected' presenta mejores valores de 'f1-score' en puntos cercanos al umbral de decisión 0.5.
- El modelo 'Wide&Deep' presenta los mejores valores de 'f1-score' alrededor

del umbral 0.3 para fraudes del tipo 5.

Es también interesante observar el comportamiento de los 3 modelos detectando todos los tipos de fraudes a la vez, según las métricas de 'precision-recall' y 'f1-score', la imagen 5.18 muestra los desempeños.

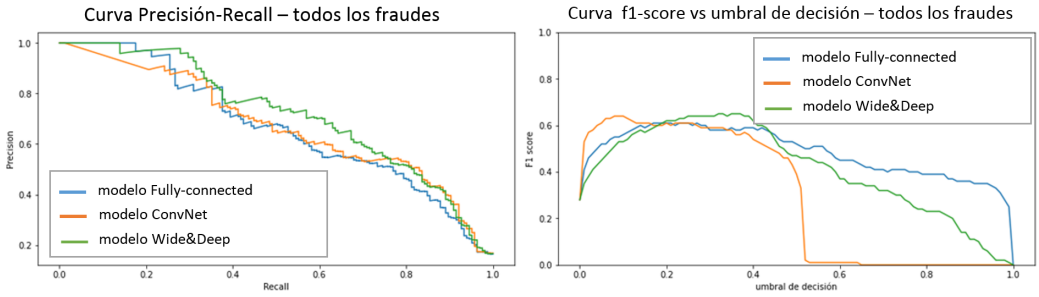

Figura 5.18: Ensayo 6 - curvas precision-recall y f1-score - comparativas entre los 3 modelos

Las curvas 5.18 reflejan una similitud en el comportamiento que presentan los 3 modelos frente a todos los tipos de fraude según las curvas 'precision-recall', sin embargo puede observarse un pequeño apartamiento en el desempeño según el 'f1-score' de los modelos 'Wide&Deep' y 'ConvNet' con respecto al modelo 'fully-connected' en el intervalo [0,0.5]. Observando aquí los picos del 'f1-score' para cada modelo, se podría concluir que integrar la red 'fully-connected' desplaza el mismo hacia el umbral 0.35, sin embargo los 3 modelos presentan valores muy similares en estos picos. Puede deducirse entonces que el comportamiento global (considerando todos los tipos de fraude) de los 3 modelos no presentan gran variación entre sí.

5.10. Ensayo 7: Modelo 'Wide&Deep-sb': Entrenamiento y validación con bases de datos 30 minutales

Para evaluar el aporte de agregar la información del balance de cargas de las subestaciones al modelo 'Wide&Deep' original se procederá a realizar un ensayo del modelo, con y sin la información proveniente del balance de las subestaciones. Para evaluar el desempeño del nuevo modelo, se armará una red del tipo 'Wide&Deep' donde se agregará la información del balance de subestaciones a la entrada del modelo 'Wide'. A éste modelo se lo comparará con uno formado únicamente por el modelo 'Deep' (red 'CNN'), y luego con el modelo 'Wide&Deep' original (con lecturas por cliente a la entrada de la componente 'Wide'). En todas las topologías se definirá la misma entrada a la red 'CNN', las lecturas de cada cliente (con fraudes) tal como se vino haciendo hasta ahora.

5.10.1. Ensayo 7: comparativo contra modelo 'CNN' ('Deep')

Antes de realizar la prueba de desempeño, se buscaron los hiperparámetros óptimos del modelo 'Wide&Deep-sb' (con balance de subestaciones). Una primer observación al realizar estas pruebas, fue el sobreajuste temprano al entrenar el modelo 'Wide&Deep-sb' (3.9.1), por lo que se decidió agregar capas de 'dropout' para evitar el efecto del sobreajuste. Una vez obtenidos los hiperparámetros para el funcionamiento óptimo del nuevo modelo con entrada el balance de subestaciones, se procede al ensayo comparativo (en validación) entre ambos modelos. Las siguientes tablas (5.10) incluyen los resultados obtenidos para cada tipo de fraude, medidos según las métricas propuestas anteriormente.

Modelo	max f1-score	Recall	Precision	AUC
Resultados de validación para **todos los tipos de fraude**				
ConvNet (Deep)	0.58	0.56	**0.60**	0.87
Wide&Deep-sb	**0.63**	**0.7**	0.57	0.88
Resultados de validación para **tipo de fraude 2**				
ConvNet (Deep)	0.22	0.40	0.16	0.87
Wide&Deep-sb	**0.27**	**0.55**	**0.18**	0.88
Resultados de validación para **tipo de fraude 3**				
ConvNet (Deep)	0.31	0.28	0.34	0.87
Wide&Deep-sb	**0.54**	**0.55**	**0.52**	0.88
Resultados de validación para **tipo de fraude 4**				
ConvNet (Deep)	0.92	0.85	**1.00**	0.87
Wide&Deep-sb	**0.92**	**0.91**	0.93	0.88
Resultados de validación para **tipo de fraude 5**				
ConvNet (Deep)	**0.40**	**0.69**	**0.28**	0.87
Wide&Deep-sb	0.38	0.56	0.28	0.88

Tabla 5.10: Ensayo 7 - comparativa de ambos modelos para cada tipo de fraude.

Los valores anteriores dan una idea del aporte de información del balance entre lecturas a pie de clientes y lecturas en su respectiva subestación. Al observar las tablas, se nota un leve incremento para los fraudes del tipo 2 y 3 en las métricas del máximo de 'f1-score' y 'precision', entre un 20 % y 50 % del 'f1-score' respectivamente. Se aprecia además un incremento en la precision del 70 % para los fraudes tipo 3. Esto es efecto de agregar esta nueva información a través de la red 'Wide', que por consecuencia impulsa en un leve incremento del AUC y en las métricas asociadas cuando se evalúa sobre todos los tipos de fraudes juntos. Cabe destacar que los fraudes del tipo 3 son aquellos de corta duración (entre 2 y 6 horas), lo que confirma la hipótesis de mejora en la capacidad del modelo en detectar fraudes de este tipo.

Curvas precision-recall y f1-score

A modo de obtener una representación visual de lo que ocurre en cada caso, se incluyen las curvas 'precision-recall' y 'f1-score' para ambos modelos, las imágenes siguientes (5.19, 5.20) muestran las curvas correspondientes para cada tipo de fraude.

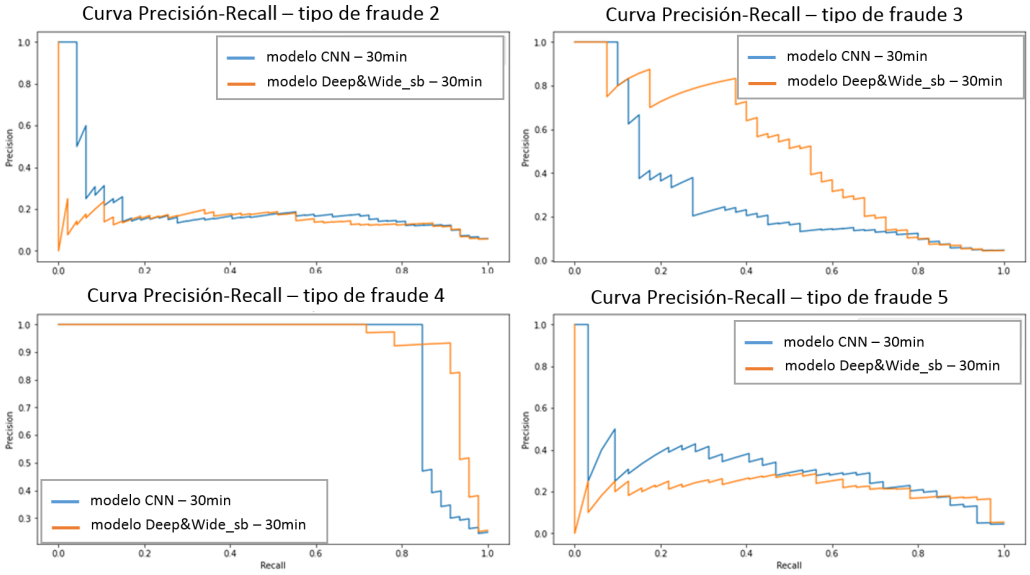

Figura 5.19: Ensayo 7 - curvas precision-recall para cada fraude - ConvNet (Deep) vs Wide&Deep (balance subest.)

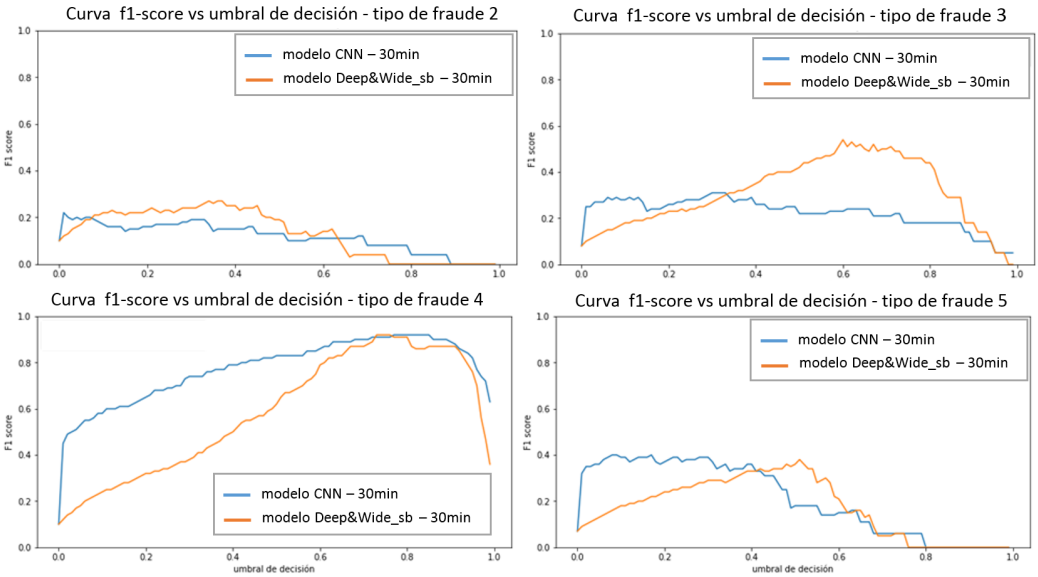

Figura 5.20: Ensayo 7 - curvas f1-score para cada fraude - ConvNet (Deep) vs Wide&Deep-sb (balance subest.)

Observando ambos tipos de curvas, para el caso de fraudes del tipo 3 se verifica una vez más el incremento notorio en el valor de las métricas de 'f1-score' y 'precision' (para todo el rango de 'recall'). Mirando detenidamente las imágenes y centrándose en la métrica del 'f1-score', puede llegarse a observar cómo el modelo con información de subestaciones mejora la detección de los fraudes tipo 2 (en el entorno del umbral 0.4). Por último, puede observarse un comportamiento muy similar (en valores picos) entre los dos modelos para el resto de tipo de fraudes, por lo que, mas allá de la leve mejora en precisión y 'f1-score' en los fraudes del tipo de variación corta, no es posible ver grandes mejorías para el resto de comportamientos.

Las curvas siguientes representan el comportamiento según métricas 'precision-recall' y 'f1-score' evaluando ambos modelos para todos los tipos de fraudes. Observando ambas imágenes (5.26) puede distinguirse un leve incremento en el desempeño general del modelo con la entrada de la información del balance de cargas. En particular el valor de 'precision' es algo mayor (que el modelo sin balance de cargas) para todo el rango de 'recall', mientras que el 'f1-score' también es mayor donde se presenta su pico (en el entorno del umbral 0.5).

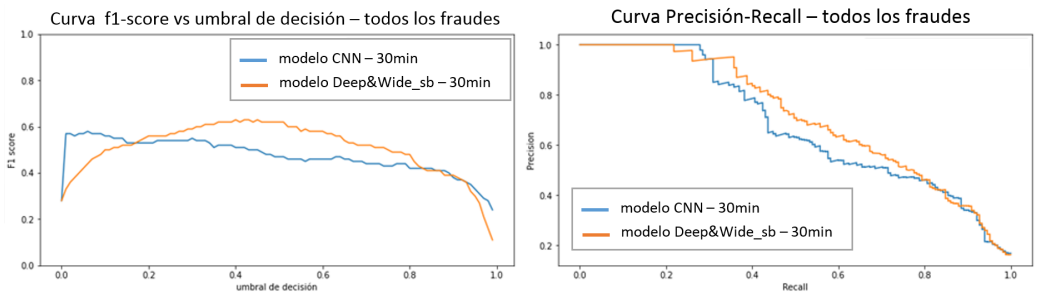

Figura 5.21: Curvas precision-recall y f1-score para todos los fraudes - ConvNet (Deep) vs Wide&Deep (balance subest.)

5.10.2. Ensayo 7: comparativo contra modelo 'Wide&Deep' estándar

En este caso se procederá a efectuar el mismo experimento que el caso anterior, pero en lugar de comparar el modelo 4 ('Wide&Deep-sb') contra el modelo 'CNN', se comparará contra el modelo 'Wide&Deep' original. Los resultados del ensayo, cuyos modelos fueron entrenados con los hiperparámetros de mejor desempeño se muestran en la tabla 5.12.

De los resultados obtenidos se concluye que ambos modelos presentan desempeños similares según el 'f1-score', en particular la diferencia mas notoria se da en el nivel de 'precision' que presenta el modelo 'Wide&Deep' clásico frente al 'Wide&Deep-sb' al detectar fraudes del tipo 5 (aproximación por la mediana). Sin embargo, el modelo 'Wide&Deep-sb' presenta mejores valores para el caso de fraudes tipo 3 (fraudes en ventanas de tiempo). Esto último sugiere una posible complementariedad entre los modelos, caso que se estudiará en el siguiente ensayo.

Modelo	max f1-score	Recall	Precision	AUC
Resultados de validación para **todos los tipos de fraude**				
Wide&Deep	**0.65**	**0.77**	**0.56**	0.89
Wide&Deep-sb	0.64	0.76	0.55	0.88
Resultados de validación para **tipo de fraude 2**				
Wide&Deep	0.27	**0.60**	0.17	0.89
Wide&Deep-sb	**0.32**	0.53	**0.23**	0.88
Resultados de validación para **tipo de fraude 3**				
Wide&Deep	0.44	0.40	0.50	0.89
Wide&Deep_sb	**0.64**	**0.52**	**0.81**	0.88
Resultados de validación para **tipo de fraude 4**				
Wide&Deep	0.89	**0.87**	0.91	0.89
Wide&Deep-sb	**0.91**	0.85	**0.98**	0.88
Resultados de validación para **tipo de fraude 5**				
Wide&Deep	**0.40**	**0.94**	0.25	0.89
Wide&Deep-sb	0.38	0.56	**0.28**	0.88

Tabla 5.11: Ensayo 7 - comparativa Wide&Deep vs Wide&Deep-sb para cada tipo de fraude.

5.11. Ensayo 8: Análisis de complementariedad entre modelo 'Wide&Deep-sb'(modelo 4) y modelos 1 y 3.

En esta sección se analizará el grado de aporte que tiene el agregar la información de las subestaciones como entrada al modelo 'Wide&Deep'. El caso se analizará para frecuencias 30 minutales, tomando como entrada la base de datos 'CER_f14_30'. Para iniciar el estudio, se hará un ensayo comparativo del resultado a la salida de los modelos correspondientes, comparando elemento a elemento el vector de predicción de la probabilidad de fraude (para cada cliente). En efecto, la salida de cada uno de los modelos 1 y 3 se graficará contra el vector de salida del modelo 'Wide&Deep-sb', y de esta manera se observará la distribución que adquieren en el plano, muestras positivas y negativas.

Tomando las comparaciones de a pares, es decir modelo 'CNN vs Wide&Deep-sb' y 'Wide&Deep vs Wide&Deep-sb' en un segundo ensayo, se presenta a continuación los resultados obtenidos.

En las imágenes 5.22 y 5.23 se ve representada la correlación que presentan los modelo 'Wide&Deep-sb' y 'CNN', la misma está representada por la distribución de los puntos correspondientes producto de la unión de un modelo y otro a través de los ejes 'x-y'.

Observando las imágenes anteriores puede identificarse la existencia de una relación complementaria entre ambas modelos. Particularmente, en los puntos etiquetados como negativos (no fraudes) se observa una concentración de los mismos en el origen (imagen derecha) con tendencia hacia el eje 'y'. Mientras que en la

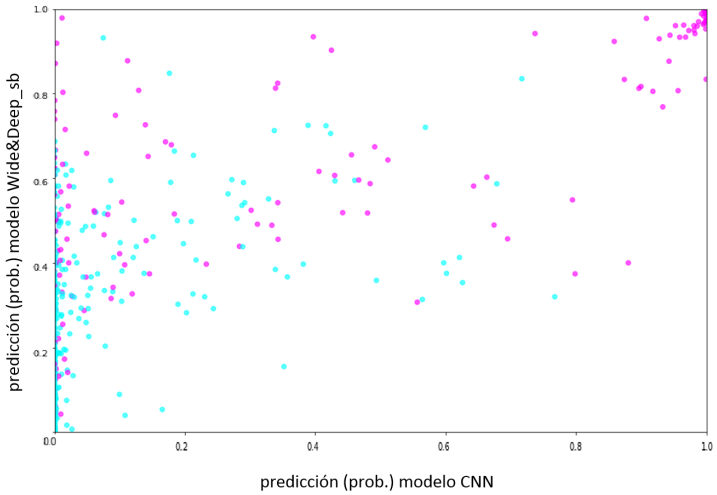

Figura 5.22: Ensayo 8 - densidad de muestras positivas y negativas - modelo Wide&Deep-sb vs CNN (modelo 1)

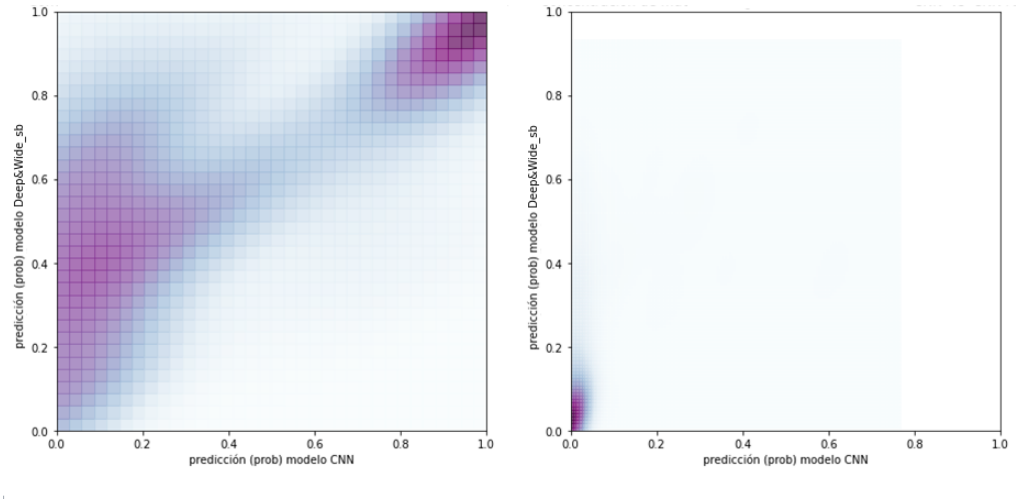

Figura 5.23: Ensayo 8 - densidad de muestras positivas y negativas - modelo Wide&Deep-sb vs CNN.

imagen de la izquierda (concentración de puntos etiquetados como fraudes), puede observarse una correlación débil entre ambos modelos con fuerte concentración de puntos en el cuadrante superior izquierdo del plano. Esto implicaría un posible aporte de complementariedad entre ambos modelos, característica que se estudiará mas adelante al realizar ensayos con diferentes fusiones de ambos modelos.
Al igual que en el caso anterior, se analizará a continuación el grado de complementariedad entre el modelo 'Wide&Deep-sb' (modelo 4) y 'Wide&Deep' original (modelo 1). En este sentido, se tendría la comparación entre dos modelos similares 'Wide&Deep' con la única diferencia las muestras que entran por la componente 'Wide' (información de subestaciones en el caso 'CNNSB', y serie temporal

de lecturas de cada cliente en el caso 'Wide&Deep' original). El resultado de la comparativa se observa en las imágenes 5.24 y 5.25.

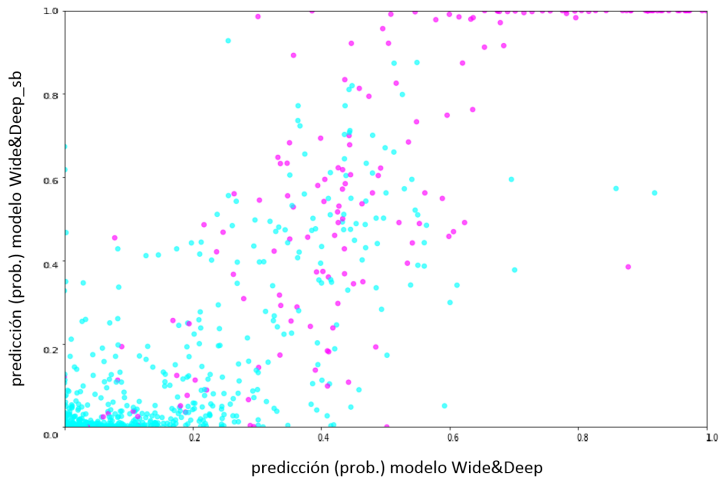

Figura 5.24: Ensayo 8 - densidad de muestras positivas y negativas - modelo Wide&Deep-sb vs Wide&Deep.

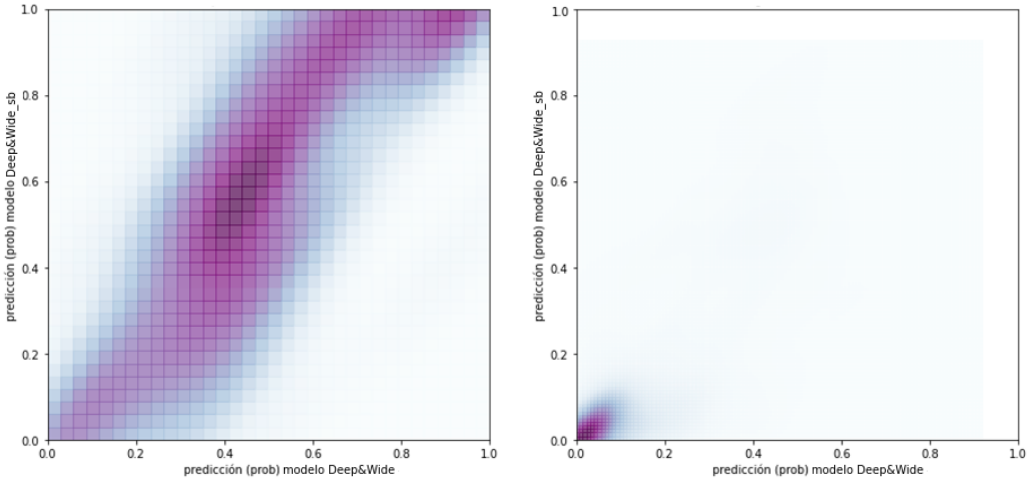

Figura 5.25: Ensayo 8 - densidad de muestras positivas y negativas - modelo Wide&Deep-sb vs Wide&Deep.

Tal como se aprecia en las imágenes 5.24 y 5.25 las muestras etiquetadas como positivas presentan una concentración cercana a la diagonal con un foco centrado en el punto [1,1], esto indica una complementariedad débil entre los modelos 'CNNSB' y 'Wide&Deep'. Para el caso de muestras negativas, el comparativo resulta similar al caso anterior, concentrándose la mayor parte de muestras negativas en el origen. Con el fin de analizar si pudiese existir una solución conformada por la fusión entre modelos, y así mejorar el desempeño de las métricas consideradas, se presenta el siguiente ensayo. En el mismo se fusionarán los modelos 'CNNSB', 'CNN' y

'Wide&Deep' con diferentes reglas y se testeará el desempeño a través de las métricas consideradas en el capitulo 3.

5.11.1. Ensayo 8: Fusión de modelos y pruebas de validación

En esta sección se analizará el desempeño resultante de la fusión de los modelos vistos. Los modelos serán fusionados a través de uniones lógicas 'AND', 'OR', y a través de la 'SUMA' y normalización de los valores probabilísticos (vectores a la salida de cada modelo). Los modelos que se ensayarán serán dos, y están conformados por la unión 'CNN-Wide&Deep-sb' y 'Wide&Deep-Wide&Deep-sb'. Para este ensayo, y antes de fusionar los modelos, se procedió a entrenar cada componente por separado (en su punto de funcionamiento óptimo), para unir finalmente los mismos aplicando cálculos lógicos (AND, OR y SUMA) sobre los vectores de salida de cada uno. A continuación se presentan algunos resultados de los ensayos realizados, los mismos incluyen gráficas 'precision-recall' y 'f1-score' para cada fusión, comparando contra el modelo estándar (sin lecturas de subestaciones). En las imágenes 5.26 y 5.27 puede apreciarse los diferentes comportamientos, destacándose las diferencias entre ellos.

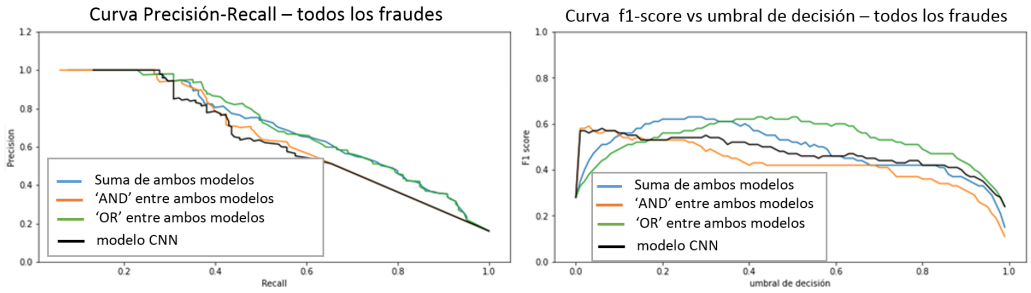

Figura 5.26: Ensayo 8 - Curvas 'precision-recall' y 'f1-score' para fusión 'CNN-Wide&Deep-sb'

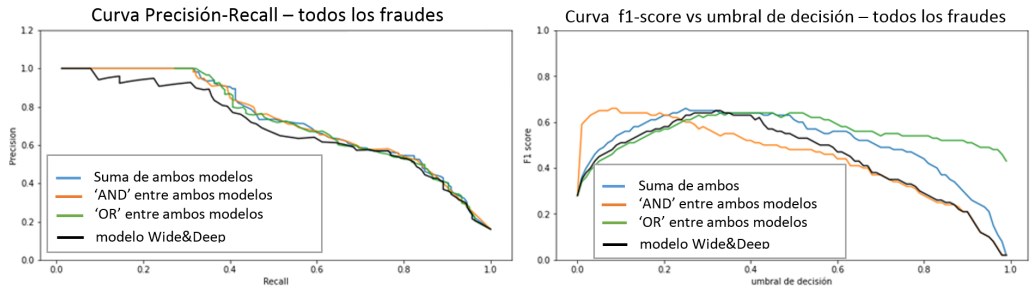

Figura 5.27: Ensayo 8 - Curvas 'precision-recall' y 'f1-score' para fusión 'Wide&Deep-Wide&Deep-sb'

Las gráficas anteriores reflejan los desempeños de los modelos 'CNN' (ConvNet) y 'Wide&Deep' contra el desempeño de las diferentes fusiones (suma, 'AND',

y 'OR') de ellos mismos junto al modelo 'Wide&Deep-sb' (modelo 4). Observando los resultados puede deducirse que, ambas fuciones, refelejan una mejora en el desempeño (resultante de la fusión 'suma' y 'OR') frente al modelo 'CNN'. Esta ganancia no es tan notoria comparando ambas fusiones contra el modelo 'Wide&Deep' original, donde los niveles del modelo original contra las fusiones se mantiene en picos y rangos similares. Por ultimo, se puede destacar en el caso de la figura 5.26, cómo se amplia el rango de umbrales óptimos para obtener un buen valor de 'f1-score' (en comparación a la curva del modelo 'CNN'). Con el objeto de ampliar en mas detalles, se presenta la siguiente tabla comparativa (5.12), en la que se reflejan los valores obtenidos de la fusión de ambos modelos (a través del método 'AND'), comparados contra los desempeños del modelo 'Wide&Deep' original.

Tabla comparativa 'Wide&Deep' vs modelo fusiones 'WDSB-CNN' y 'WDSB-WD'

Modelo	max f1-score	Recall	Precision
Resultados de validación para **todos los tipos de fraude**			
Wide&Deep	0.65	0.77	0.56
WDSB-CNN	0.63	0.65	**0.61**
WDSB-WD	**0.66**	**0.82**	0.54
Resultados de validación para **tipo de fraude 2**			
Wide&Deep	0.27	0.60	0.17
WDSB-CNN	0.26	0.47	0.18
WDSB-WD	**0.31**	**0.62**	**0.20**
Resultados de validación para **tipo de fraude 3**			
Wide&Deep	0.44	0.40	0.50
WDSB-CNN	0.43	0.48	0.40
WDSB-WD	**0.67**	**0.60**	**0.75**
Resultados de validación para **tipo de fraude 4**			
Wide&Deep	0.89	0.87	0.91
WDSB-CNN	**0.92**	0.85	**1.0**
WDSB-WD	0.91	**0.91**	0.91
Resultados de validación para **tipo de fraude 5**			
Wide&Deep	0.40	**0.94**	0.25
WDSB-CNN	**0.43**	0.56	**0.35**
WDSB-WD	0.37	0.75	0.25

Tabla 5.12: Ensayo 8 - comparativa Wide&Deep vs fusiones (con método suma) para cada tipo de fraude.

Los resultados observados en la tabla 5.12 reflejan una mejoría (según las métri-

cas consideradas) al detectar fraudes con fusiones a través de la suma. En la tabla se pueden observar que ambas fusiones superan en casi todos los casos el desempeño de la red 'Wide&Deep' estándar. Del análisis realizado se desprende la posibilidad de existencia de complementariedad entre el modelo 'Wide&Deep' original y su versión 'Wide&Deep-sb' con entradas las lecturas de las subestaciones. Recordando que todos los experimentos anteriores fueron realizados con muestras de validación, resta el ensayo final, el ensayo de todos los modelos probados anteriormente con muestras de test.

5.12. Ensayo de Test con base CER para todos los modelos

En este sección se procederá a presentar los resultados obtenidos del ensayo final de Test para todos los modelos analizados. Para dicho experimento será considerados los modelos con muestras 30 minutales, eligiendo los hiperparámetros óptimos para cada uno de ellos. Los mismos serán entrenados con las muestras de entrenamiento y validación utilizadas en todos los experimentos anteriores. Para evaluar el desempeño de los modelos en Test se clasificará un conjunto de muestras totalmente nuevo y desconocido. Para el caso de los modelos fusionados se considera el método 'suma' para la unión, ya que el mismo arrojó resultados satisfactorios en ensayos anteriores.

A todos los datos se aplicará la normalización 'MinMaxScaler', como se vino haciendo hasta ahora, la misma lleva los datos de entrenamiento (con validación) y test al rango entre 0 y 1 (con μ=0 y σ=1). Este re-escalado permite dotar de mayor robustez a los modelos frente al cambio de hiperparámetros y muestras en la etapa de entrenamiento de los mismos.

La tabla 5.13 muestra los resultados obtenidos con muestras de Test para todos los modelos analizados.

Modelo	max f1-score	Recall	Precision	Accuracy	AUC
DenseNet (Wide)	0.57	0.62	0.54	0.89	0.86
ConvNet (Deep)	0.58	0.62	0.53	0.89	0.88
Wide&Deep	0.58	0.68	0.51	0.88	0.87
Wide&Deep-sb	0.60	**0.74**	0.50	0.88	**0.90**
WDSB-CNN	**0.61**	0.64	0.59	0.90	0.89
WDSB-WD	0.60	0.57	**0.64**	**0.91**	0.88

Tabla 5.13: Ensayo 9 - desempeño en Test para todos los tipos de fraude.

Los valores de la tabla anterior indican una mejora potencial del modelo 'Wide&Deep' original al incluir la información de las subestaciones, considerando tanto el modelo sin fusión como ambos modelos fusionados aquí considerados. Cabe destacar aquí que el relevamiento de los valores, tomó como punto de partida el umbral de decisión donde el 'f1-score' se maximiza. Observando los valores de la

tabla puede concluirse que existe un aporte de información al agregar la información del balance de carga de las subestaciones, así como también la existencia de complementariedad entre los modelos considerados en las fusiones 'WDSB-CNN' y 'WDSB-WD'. En caso de tener que elegir un mejor modelo para la detección de fraudes en la base de datos CER, es muy probable que la recomendación sea trabajar con un modelo fusionado, donde se incluya la información que aporta el balance de cargas (clientes-subestaciones). En el caso de los ensayos se consideró siempre un modelo en el que la información de las subestaciones se implementó tomando en consideración que a cada subestación le correspondieran 200 clientes, por lo tanto, tomando éste último como un parámetro, queda cómo trabajo futuro evaluar cómo cambia el desempeño al variar el mismo.

5.13. Ensayo 10: prueba base de datos CER con histórico de 1 mes

En función de la prueba en campo a realizar (ver capítulo 6) con la base de datos real, la cuál presenta datos de duración 1 mes, el objetivo del ensayo será ver en qué grado se modifica el desempeño del modelo 'Wide&Deep' al considerar muestras con históricos de 1 mes. Para ello se deberá llevar la base de datos CER a la misma cantidad de características por cliente, es decir 1 mes. Para ejecutar este procedimiento se emplea el código en 'CER_to_1month', aquí se toma como entrada la base de datos 'CER_f14_30' y se hace un barrido cliente a cliente con una ventana de ancho 1 mes y fecha inicial aleatoria por cada cliente. De esta manera se obtiene la base de datos 'CER1month_f14', con esta última se entrena el modelo que desempeñará la tarea de clasificar las muestras de la base de datos real. El criterio de tomar aleatorio el comienzo del barrido por cada cliente se aplica con el objeto de emular lo mejor posible la realidad, ya que la base de datos de UTE contiene poca cantidad de datos relevados, lo que hace que un cliente identificado como fraudulento puede no haber cometido fraude el mes relevado. Luego de los ajustes necesarios, se procede con el ensayo de búsqueda de parámetros óptimos para el modelo a entrenar con el nuevo ancho de muestras de 1 mes, y su respectivo test de validación.

5.13.1. Ensayo 10: Entrenamiento, validación, y test con base de datos CER de ancho 1 mes.

Este ensayo permitirá encontrar el punto donde el modelo adaptado para muestras de 1 mes logra su mejor desempeño. Para la búsqueda de parámetros se tomarán los mismos rangos de los ensayos anteriores, considerando los mismos conjuntos de entrenamiento y validación. Los resultados de validación para los puntos óptimos obtenidos fueron los siguientes (5.14):

Por otro lado, la tabla 5.15 muestra los resultados obtenidos para el ensayo con muestras de Test de la base de datos CER.

Batch-size	epochs	Adam (lr)	AUC
32	10	0.00066	**0.68**
64	12	0.00066	**0.67**
128	15	0.00066	**0.68**

Tabla 5.14: Ensayo 10 - resultados en Validación con red Wide&Deep (fercuencia 30 minutal y ancho 1 mes).

Modelo	max f1-score	Recall	Precision	Accuracy	AUC
Wide&Deep	0.29	0.49	0.21	0.71	0.64

Tabla 5.15: Ensayo 10 - resultados en Test con red Wide&Deep (fercuencia 30 minutal y ancho 1 mes).

A partir de los resultados obtenidos puede concluirse que, el recortar la base de datos CER original de 1 año y medio de lecturas a muestras de 1 mes, impacta en el desempeño antes alcanzado. Se aprecia entonces una degradación del modelo entrenado, lo que se atribuye a la pérdida considerable de información (cerca de un 95 %) por cada cliente. En particular, es de esperar que la prueba de campo arroje resultados similares o hasta de menor desempeño, ya que los fraudes cometidos son reales (no simulados), y el comportamiento operativo es diferente al de la base de datos CER.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 6

Evaluación sobre datos de medidores inteligentes de UTE

6.1. Resumen

El capítulo presenta una prueba preliminar de campo con los datos reales que se consiguieron en el marco del convenio de colaboración UTE-IIE(UDELAR). Esta es la base de datos 'UTE-DB', la cual cuenta con información de 2912 clientes, y cuya información es tomada de los nuevos medidores inteligentes instalados por la empresa estatal UTE. Las pruebas a realizar son dos y se diferencian principalmente en los datos tomados para entrenar el modelo utilizado, ya sea con la base de datos CER ó la propia base de datos de UTE. Los ensayos incluyen una mirada cualitativa del tipo de fraude que presenta la base de datos real, con el fin de poder compararlos con los fraudes sintéticos considerados y así interpretar los resultados.

6.2. Prueba en campo con base de datos UTE

Los ensayos realizados son sobre la base de datos real 'UTE-db30' (frecuencia 30 minutal), cuyas características se describen en 4.6.1. El modelo que se considerará para esta prueba será el 'Wide&Deep' estándar, ya que la base de datos real no dispone de la información de las subestaciones.

Tal como se indica en el ensayo 10 (capítulo 5), la base de datos 'UTE-DB' cuenta con un histórico de 1 mes, lo que imponía ciertas limitaciones. Esto llevó a hacer un análisis con la base de datos CER tomando un histórico de 1 mes (ensayo 10), y de esta manera poder prever cómo respondería el modelo 'Wide&Deep'. Utilizando el mismo punto de funcionamiento que el obtenido para el ensayo 10 se procede a realizar los ensayos que se describen a continuación:

1. Primer ensayo: entrenamiento con base de datos 'CER1month_f14' y test sobre la base de datos 'UTE-DB' completa, sobre punto de funcionamiento encontrado con base de datos 'CER1month_f14'.

2. Segundo ensayo: entrenamiento y test con base de datos 'UTE-DB', sobre punto de funcionamiento encontrado con base de datos CER (de histórico un mes).

6.2.1. Primer ensayo: entrenamiento con base de datos 'CER1month_f14' y test sobre base de datos UTE-DB.

Para ejecutar este primer ensayo se entrenó el modelo 'Wide&Deep' con la base de datos 'CER1month_f14' en el punto de funcionamiento encontrado en el ensayo 10 (capítulo 5). El dataset de test fue la base de datos real 'UTE-DB' con 2912 clientes, donde 121 pertenecen a clientes no confiables (fraudes). Los resultados obtenidos mostraron como el desempeño no supera un AUC de 0.5 (aleatoriedad), no alcanzando el obtenido en la prueba de entrenamiento y validación con datos CER de 1 mes (5.14). Particularmente, se observó cómo el modelo detecta una gran cantidad de muestras no fraudulentas como positivas, además de tener un 'precision' en la detección considerablemente baja. En las imágenes 6.1, 6.2, y 6.3 se muestran las características de las muestras reales detectadas como verdaderos negativos, falsos positivos, y verdaderos positivos respectivamente.

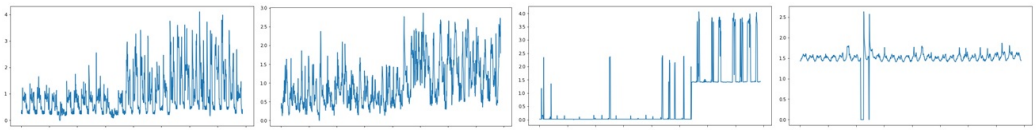

Figura 6.1: Muestras detectadas como verdaderos negativos

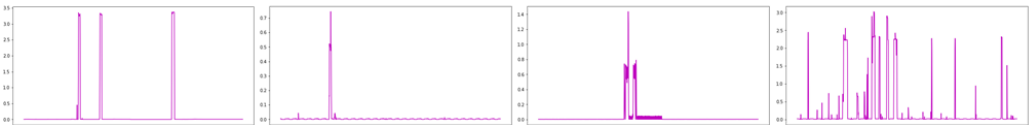

Figura 6.2: Muestras detectadas como falsos positivos

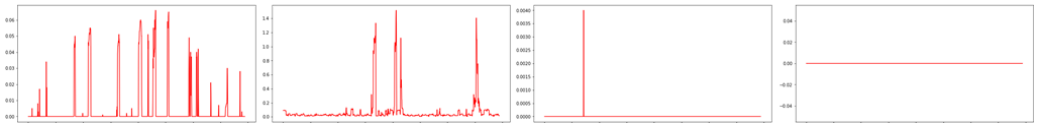

Figura 6.3: Muestras detectadas como verdaderos positivos

Observando las imágenes se puede concluir que el modelo entrenado con fraudes sintéticos de la base de datos CER, detecta como falsos positivos (ver figura 6.2) a clientes cuya curva de consumo presenta características similares a fraudes del tipo 4 (lectura nula durante cierto tiempo) combinada con fraudes tipo 2 (lectura alterada por tiempo indefinido). Sin perjuicio de lo anterior, puede verse que el modelo logra detectar algunos fraudes (ver figura 6.3) de naturaleza similar a los

detectados como falsos positivos. Por otro lado, observando las muestras de menor 'score', es decir las que el modelo identifica como verdaderas negativas (ver figura 6.1), se puede notar que existe cierta lógica ya que las curvas de consumo parecen ser de uso normal. Se concluye entonces que el modelo 'Wide&Deep' entrenado con base de datos CER (con fraudes sintéticos e histórico de un mes), no alcanza a tener un desempeño aceptable en test sobre la base de datos real Uruguaya. Esto puede deberse a diversos factores:

1. Fraudes de la base de datos CER son simulados y su naturaleza es sostenida en el tiempo, a diferencia de fraudes reales, cuya naturaleza no está matemáticamente determinada.

2. Escasa cantidad de muestras reales por cliente, dada la complejidad de una base de datos real.

3. Escasa cantidad de clientes relevados, se podría esperar mejor desempeño de un modelo que es entrenado para detectar fraudes reales y no fraudes simulados.

4. Diferentes contextos operativos entre la base de datos CER (con la cual se entrena) y base de datos real (base de test).

6.2.2. Segundo ensayo: entrenamiento y test con base de datos 'UTE-DB', sobre punto de funcionamiento encontrado con base de datos CER (de histórico un mes).

El presente ensayo considera únicamente la base de datos 'UTE-DB30' para el entrenamiento y test del modelo 'Wide&Deep' con ventana de un mes. A partir del siguiente experimento se determinará el grado de mejora que logra el modelo al entrenarse con muestras y fraudes reales de la base de datos UTE. En este sentido, se partirá la base de datos real 80-20 % para entrenamiento y test respectivamente, a su vez cabe destacar que los hiperparámetros del modelo a utilizar son los obtenidos con la base de datos 'CER1month_f14' (ver ensayo 10 del capítulo 5).

En la tabla (6.1) se muestran los resultados obtenidos según las métricas consideradas sobre el umbral de decisión óptimo ('f1-score' máximo). Se puede observar como aquí el AUC logra valores similares a los obtenidos al entrenar y testear con base de datos CER de un mes (capítulo 5, ensayo 10). Por otro lado, al observar la matriz de confusión (6.2) se verifica que el modelo incrementa su capacidad para identificar muestras no fraudulentas, bajando significativamente la cantidad de falsos positivos (respecto al caso del primer ensayo).

Modelo	max F1-score	Recall	Precision	Accuracy	AUC
Wide&Deep (UTE)	0.16	0.52	0.09	0.77	0.67

Tabla 6.1: Resultados modelo 'Wide&Deep' entrenado con muestras de 'UTE-DB30'

	negativo	positivo
negativo	439	121
positivo	11	12

Tabla 6.2: Matriz confusión - resultado en muestras de test.

Evaluando los resultados obtenidos, puede afirmarse que el modelo 'Wide&Deep' entrenado con la base de datos real obtuvo un desempeño aceptable. Mas aún, considerando que los datos tienen poca historia y presentan un alto grado de desbalance. Se logró tener un desempeño con un AUC de 0.67, logrando una tasa de detección superior al azar. Al ser una prueba de campo en la que se contaba con muestras de solo un mes de historia, es esperable que, con mas datos se logre tener un desempeño significativamente superior ya que el modelo así lo demuestra con la base de datos CER de histórico 1 año y medio. Si bien no se lograron sacar conclusiones contundentes, se mantiene la expectativa de que exista una mejora sustancial al agregar mas historia y mas datos reales para entrenar. En este sentido, se ve claramente que hay características de la propia base real que efectivamente agregan información al modelo. Lo que resulta prometedor, considerando la posibilidad de aplicar técnicas de re-entrenamiento que permiten agregar información real al modelo original (entrenado con una base de fraudes sintéticos CER_f14).

Capítulo 7

Conclusiones y trabajo a futuro

A partir del estudio del estado del arte se lograron identificar diferentes estrategias aplicables al problema de detección de fraudes en redes inteligentes que disponen de contadores inteligentes, identificando las bases de datos y métricas de desempeño más utilizadas. Del estudio realizado, se eligió partir del modelo 'Wide&Deep' [39] un modelo cuya principal característica es la combinación de redes neuronales profundas. Se identificó la base de datos CER, la cuál contaba con un volumen de datos razonables para abordar el entrenamiento y test de los modelos.

Por otro lado, a partir del artículo [38] y el aporte del equipo integrante del proyecto enmarcado en el convenio IIE (UDELAR) – UTE, se propusieron los criterios que permiten la generación aleatoria de fraudes sintéticos en la base de datos considerada. De esta manera, se obtuvo un 'dataset' de datos reales con fraudes simulados para entrenamiento y test de los modelos propuestos.
A partir de la base de datos CER - ISSDA [20] , se lograron generar 4 tipos de 'datasets' diferentes, dos de las cuales representan a la base de datos de frecuencia original (con y sin fraudes), otra de ellas que guarda los datos con fraudes a frecuencia diaria, y por último una que guarda la información simulada del balance de energía (subestaciones).
Para evaluar cómo repercute en la detección el cambio de granularidad en los datos, se procedió a ensayar el modelo Wide&Deep con la base de datos CER de granularidad diaria. De este ensayo, se concluyó que el bajar la resolución de las muestras degrada el desempeño del detector hasta un 20 %. Con el fin de complementar el ensayo anterior, se decidió analizar qué ocurre al bajar el histórico de la base de datos CER a 1 mes. Al hacer el ensayo correspondiente tomando una ventana mensual a su frecuencia original (30 minutos), se constató efectivamente una baja en el desempeño del modelo Wide&Deep, alcanzando valores similares a los obtenidos cuando se tomaron datos de granularidad diaria. Este es un resultado esperable, al contar con una cantidad mucho menor de volumen de datos, se considera razonable una degradación en el modelo.
La propuesta de incluir la información del balance de cargas en los modelos estudiados, requirió simular una estructura de datos típica de una topología AMI [23], donde los clientes se encuentran agrupados en subestaciones eléctricas indepen-

dientes. En este sentido, se logró diseñar una base de datos equivalente, a partir de la base de datos CER (con fraudes y sin fraudes), en la cual se incluye el total de energía acumulada por cada subestación en cada instante de muestreo. Esta idea, inspirada por el articulo [38], tuvo como objeto analizar el impacto que tiene el aporte de la información al considerar el balance de carga en las lecturas. De los ensayos realizados se obtuvieron las siguientes conclusiones, en primer lugar que el nuevo modelo 'Wide&Deep-sb' logró aumentar, en mayor o menor medida, las métricas consideradas en 3.10. Por ejemplo, el modelo 'Wide&Deep-sb' alcanzó un AUC de 0.90, además de aumentar el Recall hasta un valor de 0.74, ambos valores fueron los más altos obtenidos para ambas métricas.
Para estudiar el impacto de la combinación de modelos con información de las subestaciones, se decidió probar la estrategia de fusión, uniendo el modelo 'Wide&Deep-sb' con los modelos 'Wide&Deep' y 'CNN' (Deep) a través de uniones lógicas de 'AND', 'OR' y 'SUMA'. Luego del estudio de complementariedad entre los modelos candidatos, se procedió a evaluar el desempeño de las fusiones WDSB-WD y WDSB-CN. De la prueba realizada con muestras de validación se comprobó que ambos modelos fusionados mejoran en general el desempeño según las métricas consideradas (f1-score, precision, y recall (ver sección 3.10)). En particular se alanzaron valores de precision de 0.61, y recall de hasta 0.82 (casi un 8 % mayores que los obtenidos con Wide&Deep).
Del ensayo con todos los modelos sobre las muestras CER para Test, se confirmaron las observaciones hechas en las pruebas de validación, donde el desempeño de los modelos que incluyen la información del balance de cargas (Wide&Deep-sb, WDSB-WD , y WDSB-CN) fue algo mejor con respecto al de los modelos básicos, en todas las métricas consideradas. Particularmente, el modelo Wide&Deep-sb es el que obtuvo mejor valor de recall, además de lograr un valor de AUC de 90, el más alto obtenido.

En síntesis, como principales conclusiones o aportes del trabajo, se destaca la noción adquirida del estado del arte, en particular de las técnicas y estrategias aplicadas en este tipo de problemas, así como las características deseables de una base de datos y limitaciones existentes. Se constató que existen diversas técnicas aplicadas a este tipo de problemas, aún con mucho potencial de mejora, por lo que la detección de fraudes en redes inteligentes sigue siendo un área de estudio activa. De los ensayos realizados y modelos analizados vimos que el mejor desempeño global lo presentan los modelos que derivan de las fusiones WDSB-WD y WDSB-CNN, tal como se muestra en la tabla 5.13. En particular, los tres últimos modelos que son los que toman la información de las subestaciones, mejoraron el desempeño de los anteriores. Sin embargo cuando se analizan los distintos tipos de fraude, se ve que WDSB-CNN es mejor para fraudes del tipo 4 (como se muestra en la tabla 5.12), mientras que para fraudes del tipo 3 el modelo WDSB-DW adquiere mejor desempeño en este tipo de fraudes. Esto podría indicar que sería conveniente emplear estrategias de ensamblado de modelos a través de técnicas como 'voting', 'bagging', 'boosting', y 'stacking' para diseñar topologías especializadas en todos los fraudes. Otro camino válido podría ser tener un modelo independiente especia-

lizado para cada tipo de fraude. Por otro lado, puede concluirse que el agregado de los datos de balance aportan complementariedad (5.11) y una mejora en el desempeño. Al combinar los modelos seleccionados, se encontró que la regla de la suma alcanzó mejores resultados respecto a las alternativas 'OR' y 'AND.
En cuanto al efecto de reducir la granularidad de los datos o reducir su historia, ambas mostraron tener un impacto directo en el valor del AUC, reduciendo su valor.
Al incluir la técnica de regularización por 'dropout', se observó que la misma no aporta suficiente utilidad al desempeño del modelo Wide&Deep, sin embargo si fue útil utilizarla al agregar la información de subestaciones, ya que se observó que el modelo resultante se sobreajustaba tempranamente durante el entrenamiento. Cabe también destacar que el aplicar la técnica de balance de clases descrita en (3.8.1) no logró las mejoras esperadas en el modelo Wide&Deep, por lo que no fue usada tampoco en modelos posteriores.
En cuanto a la prueba de campo realizada con la base de datos de UTE y el modelo 'Wide&Deep' cuyos hiperparámetros fueron encontrados con la base de datos CER, se obtuvieron las siguientes conclusiones. Por un lado se constató que, entrenado con la base de datos CER (de historia mensual), el modelo no logra aprender a detectar fraudes de la base de datos UTE. Sin embargo, al utilizar muestras de esta última para entrenar el modelo, se alcanza un desempeño aceptable. Es de esperar que este mejore aún mas al contar con mayor histórico de lecturas y volumen de muestras fraudulentas.

Como trabajo futuro, se podría pensar en tener una base de datos real más sólida, con un mayor histórico de muestras y clientes etiquetados. De esta manera se podría entrenar un modelo del tipo 'Wide&Deep' con datos reales y fraudes reales, esperando que de esta manera el modelo aprenda mejor el contexto real. Trabajos similares se verían enriquecidos ampliando no sólo la cantidad de datos de entrada, sino sus características, sus propiedades. Por ejemplo, añadiendo información relacionada con la actividad económica del cliente, su localización geográfica, comportamiento, histórico que refleje su reputación, mejoraría sustancialmente la calidad de los resultados. Por otro lado, se podría realizar una prueba con la base de datos CER sintética y la red Wide&Deep con los datos de balance de carga entrando por un nuevo canal del componente Deep, en lugar de ser entrada de la red Wide. Así mismo, se podría probar integrar la información del balance de carga en modelos como los propuestos en [12] y [17].

Esta página ha sido intencionalmente dejada en blanco.

Referencias

[1] Christian Beckel, Leyna Sadamori, Thorsten Staake, and Silvia Santini. Revealing household characteristics from smart meter data. *Energy*, 78:397–410, 2014.

[2] Mohamed Chaouch. Clustering-based improvement of nonparametric functional time series forecasting: Application to intra-day household-level load curves. *IEEE Transactions on Smart Grid*, 5(1):411–419, 2013.

[3] Breno C Costa, Bruno LA Alberto, André M Portela, W Maduro, and Esdras O Eler. Fraud detection in electric power distribution networks using an ann-based knowledge-discovery process. *International Journal of Artificial Intelligence & Applications*, 4(6):17, 2013.

[4] Soma Shekara Sreenadh Reddy Depuru, Lingfeng Wang, and Vijay Devabhaktuni. Support vector machine based data classification for detection of electricity theft. In *2011 IEEE/PES Power Systems Conference and Exposition*, pages 1–8. IEEE, 2011.

[5] Matías Di Martino, Federico Decia, Juan Molinelli, and Alicia Fernández. Improving electric fraud detection using class imbalance strategies. In *ICPRAM (2)*, pages 135–141, 2012.

[6] Matías Di Martino, Alicia Fernández, Pablo Iturralde, and Federico Lecumberry. Novel classifier scheme for imbalanced problems. *Pattern Recognition Letters*, 34(10):1146–1151, 2013.

[7] Matías Di Martino, Guzmán Hernández, Marcelo Fiori, and Alicia Fernández. A new framework for optimal classifier design. *Pattern Recognition*, 46(8):2249–2255, 2013.

[8] Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134, 2016.

[9] Juan I Guerrero, Carlos León, Iñigo Monedero, Félix Biscarri, and Jesús Biscarri. Improving knowledge-based systems with statistical techniques, text mining, and neural networks for non-technical loss detection. *Knowledge-Based Systems*, 71:376–388, 2014.

[10] Md Hasan, Rafia Nishat Toma, Abdullah-Al Nahid, MM Islam, Jong-Myon Kim, et al. Electricity theft detection in smart grid systems: a cnn-lstm based approach. *Energies*, 12(17):3310, 2019.

[11] Tianyu Hu, Qinglai Guo, Xinwei Shen, Hongbin Sun, Rongli Wu, and Haoning Xi. Utilizing unlabeled data to detect electricity fraud in ami: A semisupervised deep learning approach. *IEEE transactions on neural networks and learning systems*, 30(11):3287–3299, 2019.

[12] Tianyu Hu, Qinglai Guo, Hongbin Sun, Tian-En Huang, and Jian Lan. Nontechnical losses detection through coordinated biwgan and svdd. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[13] Paria Jokar, Nasim Arianpoo, and Victor CM Leung. Electricity theft detection in ami using customers' consumption patterns. *IEEE Transactions on Smart Grid*, 7(1):216–226, 2015.

[14] Leandro Aparecido Passos Júnior, Caio César Oba Ramos, Douglas Rodrigues, Danillo Roberto Pereira, André Nunes de Souza, Kelton Augusto Pontara da Costa, and Joao Paulo Papa. Unsupervised non-technical losses identification through optimum-path forest. *Electric Power Systems Research*, 140:413–423, 2016.

[15] Josef Kittler. Combining classifiers: A theoretical framework. *Pattern analysis and Applications*, 1(1):18–27, 1998.

[16] Juan Pablo Kosut, Fernando Santomauro, Andrés Jorysz, Alicia Fernández, Federico Lecumberry, and Fernanda Rodríguez. Abnormal consumption analysis for fraud detection: Ute-udelar joint efforts. In *2015 IEEE PES Innovative Smart Grid Technologies Latin America (ISGT LATAM)*, pages 887–892. IEEE, 2015.

[17] Shuan Li, Yinghua Han, Xu Yao, Song Yingchen, Jinkuan Wang, and Qiang Zhao. Electricity theft detection in power grids with deep learning and random forests. *Journal of Electrical and Computer Engineering*, 2019, 2019.

[18] Pablo Massaferro, J Matías Di Martino, and Alicia Fernández. Fraud detection in electric power distribution: An approach that maximizes the economic return. *IEEE Transactions on Power Systems*, 35(1):703–710, 2019.

[19] Pablo Massaferro, Henry Marichal, Matias Di Martino, Fernando Santomauro, Juan Pablo Kosut, and Alicia Fernandez. Improving electricity non technical losses detection including neighborhood information. In *2018 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2018.

[20] James McBride. What is the irish social science data archive? *Irish Political Studies*, 17(sup1):1–3, 2003.

[21] Stephen McLaughlin, Brett Holbert, Ahmed Fawaz, Robin Berthier, and Saman Zonouz. A multi-sensor energy theft detection framework for advanced metering infrastructures. *IEEE Journal on Selected Areas in Communications*, 31(7):1319–1330, 2013.

[22] Fintan McLoughlin, Aidan Duffy, and Michael Conlon. A clustering approach to domestic electricity load profile characterisation using smart metering data. *Applied energy*, 141:190–199, 2015.

[23] Ramyar Rashed Mohassel, Alan Fung, Farah Mohammadi, and Kaamran Raahemifar. A survey on advanced metering infrastructure. *International Journal of Electrical Power & Energy Systems*, 63:473–484, 2014.

[24] Jawad Nagi, Keem Siah Yap, Sieh Kiong Tiong, Syed Khaleel Ahmed, and Malik Mohamad. Nontechnical loss detection for metered customers in power utility using support vector machines. *IEEE transactions on Power Delivery*, 25(2):1162–1171, 2009.

[25] Jawad Nagi, Keem Siah Yap, Sieh Kiong Tiong, Syed Khaleel Ahmed, and Farrukh Nagi. Improving svm-based nontechnical loss detection in power utility using the fuzzy inference system. *IEEE Transactions on power delivery*, 26(2):1284–1285, 2011.

[26] Franklin L Quilumba, Wei-Jen Lee, Heng Huang, David Y Wang, and Robert L Szabados. Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities. *IEEE Transactions on Smart Grid*, 6(2):911–918, 2014.

[27] Fernanda Rodriguez, Matías Di Martino, Juan Pablo Kosut, Fernando Santomauro, Federico Lecumberry, and Alicia Fernández. Optimal and linear f-measure classifiers applied to non-technical losses detection. In *Iberoamerican Congress on Pattern Recognition*, pages 83–91. Springer, 2015.

[28] Heng Shi, Minghao Xu, and Ran Li. Deep learning for household load forecasting—a novel pooling deep rnn. *IEEE Transactions on Smart Grid*, 9(5):5271–5280, 2017.

[29] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[30] Juan Tacón, Damián Melgarejo, Fernanda Rodríguez, Federico Lecumberry, and Alicia Fernández. Semisupervised approach to non technical losses detection. In *Iberoamerican Congress on Pattern Recognition*, pages 698–705. Springer, 2014.

[31] Souhaib Ben Taieb, Raphaël Huser, Rob J Hyndman, and Marc G Genton. Forecasting uncertainty in electricity smart meter data by boosting additive quantile regression. *IEEE Transactions on Smart Grid*, 7(5):2448–2455, 2016.

[32] Xing Tong, Chongqing Kang, and Qing Xia. Smart metering load data compression based on load feature identification. *IEEE Transactions on Smart Grid*, 7(5):2414–2422, 2016.

[33] Xing Tong, Ran Li, Furong Li, and Chongqing Kang. Cross-domain feature selection and coding for household energy behavior. *Energy*, 107:9–16, 2016.

[34] Yi Wang, Qixin Chen, Dahua Gan, Jingwei Yang, Daniel S Kirschen, and Chongqing Kang. Deep learning-based socio-demographic information identification from smart meter data. *IEEE Transactions on Smart Grid*, 10(3):2593–2602, 2018.

[35] Yi Wang, Qixin Chen, Tao Hong, and Chongqing Kang. Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Transactions on Smart Grid*, 10(3):3125–3148, 2018.

[36] Yi Wang, Qixin Chen, Chongqing Kang, and Qing Xia. Clustering of electricity consumption behavior dynamics toward big data applications. *IEEE transactions on smart grid*, 7(5):2437–2447, 2016.

[37] Yi Wang, Qixin Chen, Chongqing Kang, Qing Xia, and Min Luo. Sparse and redundant representation-based smart meter data compression and pattern extraction. *IEEE Transactions on Power Systems*, 32(3):2142–2151, 2016.

[38] Sook-Chin Yip, Wooi-Nee Tan, ChiaKwang Tan, Ming-Tao Gan, and KokSheik Wong. An anomaly detection framework for identifying energy theft and defective meters in smart grids. *International Journal of Electrical Power & Energy Systems*, 101:189–203, 2018.

[39] Zibin Zheng, Yatao Yang, Xiangdong Niu, Hong-Ning Dai, and Yuren Zhou. Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. *IEEE Transactions on Industrial Informatics*, 14(4):1606–1615, 2017.

Glosario

AMI Infraestructura de Medición Avanzada. 5

Aprendizaje supervisado técnica de aprendizaje automático que usa datos etiquetados para el entrenamiento. 9

Base de datos CER base de datos que surge de proyecto de organización de 'Customer Behavior Trials' llevado a cabo por la Comisión Reguladora de Energía (CER) en Irlanda. 3

Batch sub-grupo tomado del conjunto total de muestras (de entrenamiento), que es utilizado en una iteración por cada época de entrenamiento de un modelo. 9

Capa softmax función de normalización exponenecial, comprime los valores de un vector de entrada en un rango de 0 a 1.. 6

Curva ROC curva utilizada para evaluar el desempeño de un modelo (TPR vs FPR). 9

Datasets conjunto de datos almacenados en variable del tipo dataframe de librería panda. 5

Dropout técnica de regularización utilizada en el aprendizaje automático que permite reducir el sobreajuste del modelo . 9

Features propiedades mensurables o características de un fenómeno observado. 6

FPR tasa de muestras falsas positivas predichas. 9

Gradient boosting decision trees técnica de aprendizaje automático basado en la potenciación del gradiente. 6

Granularidad granularidad o resolución de la curva de consumo eléctrico. 3

Logistic Regression es un tipo de análisis de regresión utilizado para predecir el resultado de una variable en función de variables independientes o predictoras. 6

Matriz de confusión herramienta que permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado. 9

NaN proviene del acrónimo en inglés Not a Number. 7

Normalización re-escalado del valor de los datos para usar una única escala sin perder informqación. 8

NTLs pérdidas no técnicas. 5

Oversampling generar mayor cantidad de muestras de una clase determinada partiendo de un número reducido de la misma . 9

Performance medida del desempeño de un modelo según métricas consideradas. 3

Random Forest algoritmo de clasificación que consiste en la combinación de árboles de decisión . 6

Red NAN red de medidores inteligentes instalados a pie de los consumidores. 6

Smart Grids redes de distribución eléctricas ínteligentes: 1

SVMs algoritmo de aprendizaje supervisadollamado máquinas de vectores de soporte. 2, 5

TPR tasa de muestras verdaderas positivas predichas. 9

Índice de tablas

Esta página ha sido intencionalmente dejada en blanco.

Índice de figuras

Esta es la última página.
Compilado el miércoles 14 octubre, 2020.
http://iie.fing.edu.uy/