ISSN 1688-2784

Universidad de la República
Facultad de Ingeniería

Detección de pérdidas no técnicas en redes eléctricas en un contexto de migración tecnológica y maximizando el retorno económico

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

Pablo Massaferro Saquieres

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

Directores de Tesis

Alicia Fernández ..... Universidad de la República
J. Matías Di Martino ..... Duke University

Tribunal

Marcelo Fiori ..... Universidad de la República
José Lezama ..... Universidad de la República
Mario Vignolo ..... Universidad de la República
Joaquín Luque (Revisor Externo) ..... Universidad de Sevilla
Álvaro Pardo (Revisor Externo) . Universidad Católica del Uruguay

Director Académico

Alicia Fernández ..... Universidad de la República

Montevideo
9 de marzo de 2022

Detección de pérdidas no técnicas en redes eléctricas en un contexto de migración tecnológica y maximizando el retorno económico, Pablo Massaferro Saquieres.

ISSN 1688-2784

Esta tesis fue preparada en LaTeX usando la clase iietesis (v1.1).
Contiene un total de 142 páginas.
Compilada el sábado 9 abril, 2022.
http://iie.fing.edu.uy/

Los hombres de tu planeta —dijo el principito— cultivan cinco mil rosas en un mismo jardín... sin embargo no encuentran lo que buscan.

Antoine de Saint-Exupéry

Agradecimientos

En primer lugar quiero agradecer a mis tutores y directores de tesis Alicia Fernández y Matías Di Martino por su generosidad en compartir conmigo su tiempo, su conocimiento y su experiencia. En particular, quiero agradecer a Alicia por confiar en mi capacidad y darme ánimo para llevar adelante este proceso de formación y trabajo. A Matías por estar siempre presente más allá de la distancia y por sus invaluables aportes.

En este proceso de investigación aplicada tuve la suerte de trabajar en conjunto con un grupo formidable de ingenieros de UTE que participaron de diferentes formas de este proceso. Quiero agradecer a Juan Pablo Kosut, Fernando Santomauro, Andrés Jorysz, Marcelo Álvarez, Gonzalo Caudullo, Ibero Fomichov, Agustín Heberling y Alexander Martins.

El desarrollo de esta tesis fue posible gracias al poyo de UTE, de la Agencia Nacional de Investigación e Innovación (ANII), de la Comisión Académica de Posgrados (CAP) y la Comisión Sectorial de Investigación Científica (CSIC) de la Universidad de la República.

También quiero agradecer a mis compañeras y compañeros del Instituto de Ingeniería Eléctrica con quienes he tenido oportunidad de conversar e intercambiar ideas tanto en ámbitos formales como en otros espacios.

Volver a estudiar teniendo tres hijos pequeños fue un gran desafío, tanto para mí como para mi familia. Le agradezco a mi esposa Sofía García por ser una vez más mi compañera de aventuras.

Por último quiero agradecer a mi madre Ana María Saquieres y a mi padre Rodolfo Massaferro por su esfuerzo en mi formación temprana lo que me permite hoy estar culminando mi formación de doctorado.

A mis hijos Emiliano, Agustina y Juan Ignacio.

Resumen

La corriente eléctrica es parte fundamental de la vida en las sociedades modernas. Su distribución implica el uso de una red física de conductores que deja este bien de mercado expuesto al hurto y a los sistemas de medición expuestos al fraude. Según informes del Banco Interamericano de Desarrollo (BID) las pérdidas totales de energía eléctrica en América Latina y el Caribe (ALC) alcanzan el 17 % de la energía generada. Sin tener en cuenta las pérdidas técnicas (dentro de los sistemas de distribución), el perjuicio económico causado para las economías de ALC asciende a 11 mil millones de dólares anuales. Dado el impacto que las pérdidas no técnicas (NTL) generan, su detección y regularización es de gran importancia. Es necesario definir a qué clientes inspeccionar y para tal tarea las empresas se basan en el análisis de datos y consumos de sus clientes.

En esta tesis se aborda el problema de detección de pérdidas no técnicas basado en técnicas de aprendizaje automático supervisado. Los aportes principales de esta tesis se pueden resumir en: (i) la incorporación explicita de los costos y el retorno potencial asociado a las actividades de inspección y su uso en la optimización de algoritmos de aprendizaje automático; (ii) el análisis de las rutas de inspección como parte de una estrategia global; (iii) en un contexto de cambios en la infraestructura de medición, se propone una arquitectura de aprendizaje profundo que, utilizando datos de consumo multirresolución, sea capaz de mejorar la detección de NTL en contadores inteligentes; (iv) se organizó y analizó una base de datos con consumo real y fraude de magnitud sin precedentes, para validar hipótesis presentadas en trabajos anteriores y validadas en pequeñas bases de datos.

A pesar de la relevancia del impacto económico de NTL, antes de esta tesis, ningún trabajo se centró en el modelado de costos y la inclusión de estimaciones de retorno económico de las actividades de campo. En esta tesis se propone un nuevo enfoque del problema centrado en maximizar el retorno económico de las actividades de inspección. También se aborda por primera vez la inclusión de estrategias de optimización de ruteo de vehículos dentro de un esquema de detección de NTL. Los resultados experimentales son validados sobre grandes bases de datos generadas en conjunto con la empresa de generación y distribución de energía de Uruguay (UTE).

En los últimos años, el movimiento hacia la medición inteligente ha creado nuevos desafíos y oportunidades para la detección automática de NTL. En concreto, ha habido varias propuestas para tratar datos de contadores inteligentes con deep learning en los últimos tres años. Hasta donde sabemos, todos los trabajos

utilizan exclusivamente datos de contadores inteligentes o datos de contadores de menor resolución temporal, pero no existen trabajos que aborden el problema de la coexistencia de estos datos en el proceso de cambio de tecnologías de medida. Este proceso puede llevar años y la información que se genera es muy valiosa. En esta tesis se presenta por primera vez en NTL una arquitectura de aprendizaje profundo para tratar datos de consumo en multirresolución, y se demuestra que el uso combinado de datos de medidores inteligentes con registros mensuales previos genera mejores resultados que el uso único de consumo en alta resolución. La propuesta se prueba con datos reales de UTE y datos sintéticos ampliamente utilizados en la literatura. Además, utilizando una de las mayores bases de datos de acceso académico reportadas, se prueban varias hipótesis sobre la inclusión de información adicional y el desempeño de los principales algoritmos utilizados en NTL. Ideas derivadas del problema de los ataques adversarios también se utilizan en esta tesis para interpretar modelos de aprendizaje profundo.

Los resultados obtenidos en esta tesis, que dio lugar a cinco artículos científicos, han sido implementados y transferidos a la industria a través de acuerdos entre UTE y UdelaR. UTE utiliza los algoritmos propuestos y los avances científicos y tecnológicos desarrollados para la detección automática de NTL.

Abstract

Electric current is a fundamental part of life in modern societies. Its distribution implies using a physical network of conductors that expose this market good to theft and the measurement systems to fraud. According to the Inter-American Development Bank (IDB) reports, total electricity losses in Latin America and the Caribbean (LAC) reach 17 % of the energy generated. Without considering the technical losses (within the distribution systems), the economic damage caused to the LAC economies amounts to 11 billion dollars annually. Given the impact that non-technical losses (NTL) generate, their detection and regularization are of great importance. It is necessary to define which clients to visit to inspect the electrical installation. Companies perform data analysis, including consumption curves and other relevant customer information for this task.

This thesis addresses the problem of non-technical loss detection based on supervised machine learning techniques. The main contributions of this thesis can be summarized as (i) the explicit incorporation of costs and the potential economic return associated with inspection activities and their use in optimizing machine learning algorithms; (ii) analysis of inspection routes as part of a global strategy; (iii) in a context of changes in the metering infrastructure, a deep learning architecture is proposed that using multiresolution consumption data, is capable of improving NTL detection in smart meters; (iv) a database with real consumption and fraud of unprecedented magnitude was organized and analyzed, to validate hypotheses presented in previous works and validated in small databases.

Despite the relevance of NTL´s economic impact, prior to this thesis, no work focused on cost modeling and the inclusion of estimates of the economic return of inspection activities. This thesis proposes a new approach to the problem, focused on maximizing the economic return of inspection activities. The inclusion of vehicle routing optimization strategies within an NTL detection scheme is also addressed for the first time. The experimental results are validated on large databases generated with the Uruguayan energy generation and distribution company (UTE).

In recent years, the move towards smart metering has created new challenges and opportunities for automatic NTL detection. In particular, there have been several proposals to treat data from smart meters with deep learning in the last three years. As far as we know, all the papers exclusively use smart meter data or lower temporal resolution meter data, but there are no papers that address the problem of the coexistence of this data in the process of changing measurement

technologies. This process can take years, and the information that is generated is very valuable. In this thesis, a deep learning architecture is presented for the first time in NTL to treat consumption data in multiresolution, and it is shown that the combined use of data from smart meters with previous monthly records generates better results than the only use of consumption in high resolution. The proposal is tested with real data from UTE and synthetic data widely used in the literature. In addition, and using one of the largest academic access databases reported, several hypotheses about the inclusion of additional information and the performance of the main algorithms used in NTL are tested. Ideas derived from the problem of adversary attacks are also used in this thesis to interpret deep learning models.

The results obtained in this thesis, which gave rise to five scientific papers, have been implemented and transferred to the industry through agreements between UTE and UdelaR. UTE uses the proposed algorithms and the scientific and technological advances developed for NTL automatic detection.

Tabla de contenidos

Capítulo 1

Introducción

1.1. Motivación

1.1.1. Contexto e impacto del problema

La corriente eléctrica es una parte fundamental de la vida en las sociedades modernas. La iluminación, la refrigeración de alimentos, la calefacción, los sistemas motrices de las industrias, los centros de supercomputación, todo requiere en cierta medida de energía eléctrica para su funcionamiento o su manufactura. La generación y distribución de energía eléctrica fue la base de la segunda revolución industrial hacia fines del siglo XIX. En la década de 1870 fueron varias las iniciativas para generar corriente eléctrica utilizando movimiento mecánico con base en los fenómenos electromagnéticos. Para 1881 el sistema de generación de energía eléctrica de Charles Brush [9] ya se utilizaba en varias ciudades de Estados Unidos como San Francisco, Nueva York, Boston, Filadelfia y Cleveland entre otras. Los sistemas de generación y distribución de energía eléctrica comienzan a ser comercializados en todo el mundo. El sistema de corriente continua de Brush llega a Ciudad de México en 1881, a La Plata (Argentina) en 1883 y a Montevideo en 1886 [96]. Si bien los sistemas de generación y distribución han avanzado mucho desde sus comienzos hasta ahora, la distribución de energía eléctrica requiere el uso de una red física de conductores que transporta la energía desde el punto de generación hasta el punto de consumo. A medida que la red eléctrica fue expandiéndose y los tendidos de cables cubrían grandes distancias, este nuevo bien de mercado quedó expuesto al hurto. Los tendidos de cables eléctricos atraviesan miles de kilómetros y el control de posibles conexiones indebidas se tornó una tarea inmanejable. A principios del siglo XX se comenzó a legislar en todo el mundo al respecto de este tema. En Uruguay se promulgó el artículo 343 del Código Penal en diciembre de 1933, donde se diferencia entre dos tipos de consumo ilegal de energía: el hurto y la estafa (o fraude).

Independientemente de las implicancias legales que pueda tener el hurto de energía eléctrica, ¿cuál es el impacto de estos hechos? Pensemos en cualquier empresa que produce y comercializa un producto. Las empresas tienen sistemas de

producción y para controlar sus costos generan índices de rendimientos sobre los insumos. Incluyen índices de pérdidas del producto que pueden estar derivados de defectos o ineficiencias productivas. En las etapas de distribución también hay costos asociados, métricas de desempeño e índices de pérdidas que deben ser controlados. En el mercado eléctrico, el índice de pérdidas de energía es un indicador de eficiencia muy importante que tiene un gran impacto en los resultados económicos, como veremos más adelante. Un porcentaje de la energía producida en las centrales de generación (térmicas, eólicas, nucleares, hidroeléctricas) no llega a ser facturada. Una parte de esa energía se debe a pérdidas en las líneas de transmisión o en las subestaciones de transformación por efecto Joule. A esta energía se la denomina pérdidas técnicas. Al resto de la energía consumida pero no facturada, se la denomina pérdidas no técnicas, NTL por su sigla en inglés (*non-technical losses*). En este último grupo podemos diferenciar entre robo, fraude, errores de medición y errores de facturación. Ambos tipos de pérdidas son gestionadas por las empresas con el objetivo de generar un mayor margen de ganancia, minimizando costos de generación y aumentando la facturación.

Las estimaciones de pérdidas económicas de las compañías eléctricas dependen fuertemente del mercado y de la infraestructura de generación y distribución. En algunos mercados probablemente las empresas tengan que sobredimensionar sus inversiones en infraestructura para evitar que las NTL generen inestabilidades o sobrecargas de la red. En un mercado con excedentes de energía, las NTL podrían condicionar estrategias comerciales de exportación o almacenamiento (embalses, generación de hidrógeno). Las tarifas también pueden verse afectadas por las pérdidas no técnicas, no solo como una estrategia de compensación de resultados económicos, sino que también puede afectar la segmentación de clientes para planes especiales. El control de las pérdidas no técnicas también genera costos operativos al requerir de una estructura para realizar inspecciones en las instalaciones de los clientes. Incluso puede generar inversiones de infraestructura para minimizar o disuadir la realización de actividades ilegales (medidores inteligentes, protección de tendidos eléctricos, etc). Según informes del Banco Interamericano de Desarrollo (BID) las pérdidas totales en América latina y el Caribe (ALC) alcanzan el 17.0 % de la energía generada [51]. La Organización Latinoamericana De Energía (OLADE) en su reporte anual de 2019 muestra cómo la situación de pérdidas de energía ha permanecido casi invariante en los últimos cinco años [13]. La generación de energía eléctrica en ALC en 2019 fue de 1.602 TWh según datos de la OLADE. Suponiendo un 7.0 % de pérdidas técnicas, las pérdidas no técnicas en ALC ascienden a 160 TWh, unos 11 mil millones de dólares [51] [1]. En el caso de Uruguay, las pérdidas en distribución en 2019 fueron de 18.1 % [98]. Los niveles de pérdidas varían entre países en desarrollo y economías del primer mundo. El organismo europeo regulador de energía (CEER) reporta en "2nd CEER Report on Power Losses" pérdidas totales en 2020 que varían entre 2.5 % y 14.0 % dependiendo del país (a excepción de Kosovo) [79]. Todos los países, a excepción de España, Grecia y Finlandia, reportan disminución en la pérdidas en los últimos años. Las pérdidas de transmisión no superan el 2.0 % en ninguno de los 35 países

[1] Para la realización de la estimación se utilizan los precios de la energía de 2019.

que se incluyen en el informe. Las pérdidas en distribución en España en 2018 fueron de 8.5 % mientras que en Alemania las pérdidas totales fueron del 4.5 %. El Banco Mundial reporta pérdidas globales de energía en distribución del 8.3 % [78]. Esta pérdida ha sido estimada en 96 mil millones de dólares anuales [62].

Según el reporte anual de la empresa de generación y distribución de energía uruguaya UTE, en 2020 se vendió energía por 1.401 millones de dólares. En la memoria anual de UTE de 2019 se reporta un total de pérdidas de 19.7 %, correspondiendo 18.1 % a distribución y comercialización. Estos valores incluyen pérdidas técnicas y NTL. Las pérdidas técnicas para la red uruguaya pueden estimarse entorno al 6.0 % de la energía distribuida[2] lo que deja un 12.0 % de NTL. Una estimación muy simplificada de las pérdidas económicas puede hacerse tomando como hipótesis que toda la energía de NTL pudiera ser facturada. Bajo esa hipótesis las pérdidas económicas para el caso de Uruguay serían de 205 millones de dolares anuales. Sin embargo, este razonamiento no es correcto ya que no toda esa energía puede ser facturada, básicamente por dos razones: 1) aproximadamente la mitad de esa energía es consumida en zonas de vulnerabilidad socioeconómica con conexiones irregulares, donde los usuarios no pueden afrontar el costo total de la energía por sus propios medios (la memoria anual de 2019 de UTE reporta un 6.4 % por este motivo [98]); 2) el uso de energía en los casos de hurto o fraude es abusivo dado que no se paga por lo que se consume y una vez regularizado, el consumo se adecua a las posibilidades económicas del cliente o se reincide en actividades ilegales. Por este motivo, en un escenario de control total de pérdidas, el consumo total sería menor, generando ahorros en los costos de generación y ganancias por la energía facturada. La empresa ha comunicado en varios medios de prensa que su estimación en 2021 de esta pérdida es de 100 millones de dólares anuales (declaraciones de la presidenta de UTE Ing. Silvia Imaldi a El Observador 13/08/21).

Por otro lado, el uso indebido o no regulado de energía eléctrica es un peligro para la salud y la seguridad de las personas. En particular, las instalaciones precarias en contextos de vulnerabilidad socioeconómica son posibles fuentes de incendios o de choques eléctricos. Instalaciones sin llaves de corte automático, llaves diferenciales o seccionamiento alguno, son claramente peligrosas. Si bien esto es una realidad, también existen instalaciones muy ingeniosas realizadas por técnicos con expertise generando importantes fraudes tanto para uso residencial como productivo. Existen principalmente dos grandes problemas en el control de NTL, la regularización de consumos de usuarios en condiciones socioeconómicas vulnerables y la identificación de NTL en el resto de los usuarios.

1.1.2. Detección de pérdidas no técnicas

Dado el impacto que las NTL generan, las empresas de distribución de energía cuentan con personal abocado a la detección de pérdidas. La detección efectiva de un consumo indebido de energía eléctrica implica una inspección de la instalación

[2]Según informes realizados por el sector de Planificación y Estudios de Distribución de UTE.

in situ. Esta inspección es realizada en general por dos funcionarios especializados en la tarea. Los datos más utilizados para analizar las pérdidas están dados por las mediciones de consumo. Las empresas cuentan con medidores de energía en la acometida de cada cliente y esto les permite computar el consumo a ser facturado (comerciales o residenciales). Uruguay comenzó en 2019 una migración progresiva hacia medidores inteligentes instalando 600.000 hasta 2021, lo que representa algo menos de la mitad de sus clientes activos. Esta transformación requiere de una gran inversión y trae importantes oportunidades comerciales, en particular como fuente de información para la detección de pérdidas no técnicas. A esta nueva infraestructura de medición inteligente se la conoce por la sigla AMI (*advanced metering infrastructure*).

La energía consumida por los clientes es facturada de forma mensual. Esta información ha sido la principal fuente de análisis para determinar qué clientes deben recibir una inspección. Algunas de las estrategias utilizadas en el pasado para generar una inspección estaban vinculadas a la detección de una disminución de consumo por debajo de algún umbral o seleccionar registros de consumos con muy poca varianza. Sobre finales de la primera década del siglo XXI se comienza a investigar el uso de aprendizaje automático (o reconocimiento de patrones) para el análisis de los datos de los clientes y la identificación de posibles fraudes [71, 73, 74, 85]. En los últimos 10 años, la detección automática de NTL ha sido un campo muy activo a nivel académico [69, 90]. Al igual que en otras áreas del análisis de datos, recientemente ha evolucionado hacia técnicas de aprendizaje profundo y *big data*. En el capítulo 2 se presentan los principales avances que ha habido en esta materia. Previo a la realización de esta tesis el Instituto de Ingeniería Eléctrica de la Universidad de la República, en colaboración con la empresa UTE, abordó el problema de detección automática de fraudes utilizando un enfoque clásico de reconocimiento de patrones sobre consumos mensuales [23, 24, 56].

1.2. Objetivos y alcance

Esta tesis tiene como objetivo avanzar en la comprensión del problema de las pérdidas no técnicas, aportando nuevas estrategias que vinculen el aprendizaje automático con la gestión del negocio.

1.2.1. Preguntas de investigación

- ¿Cómo definir la cantidad de inspecciones necesarias considerando las restricciones operativas? ¿Cuáles son las métricas más apropiadas para evaluar desempeño en detección automática de NTL?
- ¿Cómo integrar modelos de costos de inspección en sistemas de detección automática de NTL? ¿Cuánto incide el método de ruteo en la selección óptima de clientes a inspeccionar?
- En un contexto de migración de los sistemas de medición de consumo de energía a medidores inteligentes, ¿cómo incide la resolución de la medida en

el desempeño de los algoritmos de detección? ¿Cómo aprovechar la coexistencia de mediciones en diferentes resoluciones en el problema de NTL?

- ¿Cuánto aporta la extracción de características de los perfiles de consumo? ¿Qué aprenden las redes neuronales al usar datos crudos?

- Dadas las ventajas de trabajar con bases de datos etiquetadas y la dificultad de generarlas, ¿cuál es la incidencia del tamaño de las bases de datos de entrenamiento en el desempeño de los algoritmos de detección de NTL?

1.2.2. Alcance de la tesis

Esta tesis está centrada en el uso de los perfiles de consumo de los clientes para la detección automática de pérdidas no técnicas. Se utilizan tanto datos mensuales de medidores electromecánicos o electrónicos como lecturas quinceminutales de medidores inteligentes. Otras informaciones vinculadas a los contratos de los clientes y su historial son también utilizadas. Queda por fuera del alcance de esta tesis el uso de otras fuentes de información como microbalances de subestaciones, registros de alarmas de medidores inteligentes, variables climáticas u otros datos socioeconómicos.

1.3. Principales aportes de la tesis y publicaciones arbitradas

En el marco de esta tesis se propuso por primera vez en NTL utilizar como métrica de evaluación el retorno económico. El método propuesto fue publicado en IEEE Transactions on Power Systems [63] y ha sido citado, entre otros, por grupos de investigación del MIT [2] y de una de las universidades más prestigiosas de China, Zhejiang University. El grupo de trabajo del profesor Dr. Fushuan Wen complementa nuestro trabajo explícitamente en [18]. También es de destacar la generación, en conjunto con UTE, de una de las bases de datos de NTL más grandes que se reporta en la literatura a nivel académico. Esto nos permitió probar varias hipótesis y sacar conclusiones robustas sobre el problema de NTL con datos de consumos mensuales [65].

Durante la tesis se realizaron dos proyectos de investigación y transferencia tecnológica a la industria. Los desarrollos tecnológicos DAICE y DeepDAICE están operativos en la empresa UTE e incluyen varias de las propuestas formuladas en esta tesis.

1.3.1. Publicaciones arbitradas

- Massaferro, P., Di Martino, J. M., & Fernández, A. (2019). Fraud detection in electric power distribution: An approach that maximizes the economic return. *IEEE Transactions on Power Systems*, 35(1), 703-710.

- Massaferro, P., Di Martino, J. M., & Fernández, A. (2022). Fraud detection on power grids while transitioning to smart meters by leveraging multi-resolution consumption data. *IEEE Transactions on Smart Grids* (accepted).

- Massaferro, P., Marichal, H., Di Martino, J.M., Santomauro, F., Kosut, J. P., & Fernández, A. (2018, August). Improving electricity non technical losses detection including neighborhood information. In *2018 IEEE Power & Energy Society General Meeting (PESGM)* (pp. 1-5).

- Massaferro, P., Di Martino, J. M., & Fernández, A. (2021, February). NTL Detection: Overview of Classic and DNN-based Approaches on a Labeled Dataset of 311k Customers. In *2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)* (pp. 1-5).

- Massaferro, P., Di Martino, J. M., & Fernandez, A. (2021, September). NTL Detection: Optimization of Inspection Routes Weighing Mobility Cost and Detection Likelihood. In *2021 IEEE PES Innovative Smart Grid Technologies Conference-Latin America (ISGT Latin America)* (pp. 1-5).

1.3.2. Desarrollos tecnológicos transferidos a la industria

- DAICE: El Detector Automático de Irregularidades en Consumos Eléctricos, es una herramienta desarrollada en el marco del proyecto "Implantación de un sistema de detección automática de irregularidades en el uso de energía eléctrica" financiado por el Fondo Sectorial de Energía de la Agencia Nacional de Investigación e Innovación (ANII-FSE), 2016-2018. A lo largo del desarrollo del doctorado se han incluido en esta herramienta varios resultados académicos de esta tesis, incluyendo un modo de operación de máximo retorno económico.

- DeepDAICE: Herramienta de detección de fraudes basada en aprendizaje profundo sobre datos de medidores inteligentes. Este desarrollo se enmarca en el proyecto "Detección de anomalías en medidores inteligentes" (2019-2021) como parte de un convenio entre Facultad de Ingeniería y UTE. Esta herramienta incluye los aportes discutidos en el capítulo 6 de esta tesis.

1.4. Estructura de la tesis

El resto de esta tesis se estructura de la siguiente forma:

- Capítulo 2: Se plantea el problema de tesis y se hace una revisión del estado del arte en detección automática de pérdidas no técnicas de energía eléctrica. Se plantea una discusión sobre las métricas de desempeño para problemas de clasificación con desbalance y específicamente para evaluar resultados en detección de NTL.

- Capítulo 3: Se utilizan las ventajas de una de las bases de datos de NTL de acceso académico más grandes del mundo para validar varias hipótesis. Se responden las siguientes preguntas: i) ¿Cuál es el efecto del tamaño de la base de entrenamiento sobre los resultados? ii) ¿Sirve realizar extracción de características sobre datos de consumos? iii) ¿Cuál es el impacto del uso de información adicional de los clientes? iv) ¿Pueden usarse las ideas de ataques adversarios para generar intuición sobre lo que aprenden las redes neuronales en NTL?

- Capítulo 4: Se propone un abordaje del problema para maximizar el retorno económico teniendo en cuenta los costos de inspección y estimando el retorno económico esperado para cada inspección. Se propone un método para dimensionar las secciones operativas encargadas de las inspecciones teniendo en cuenta modelos de costos generales de funcionamiento.

- Capítulo 5: Se introduce el problema de ruteo de vehículos en la gestión de pérdidas no técnicas, realizando una formulación matemática de optimización combinatoria. Se muestra el impacto de la gestión de las inspecciones desde el punto de vista de los costos operativos derivados de la programación de actividades de inspección.

- Capítulo 6: Se presenta una arquitectura de aprendizaje profundo para el análisis de datos de consumos proveniente de medidores inteligentes. Se propone una estrategia para el manejo de datos en diferentes resoluciones temporales sacando provecho del histórico de datos mensuales en clientes que comienzan a ser telemedidos.

- Capítulo 7: Se resumen las principales conclusiones de la tesis.

Capítulo 2

Estado del arte en detección automática de NTL

2.1. Introducción

En los últimos veinte años se han propuesto diferentes enfoques basados en aprendizaje automático para la detección de NTL. Glauner et al. [36] presenta una revisión de los trabajos más relevantes publicados hasta 2017. Complementariamente en 2018 Messinis y Hatziargyriou [69] presentan una revisión exhaustiva que incluye métodos orientados a la red (topología y balances de energía), orientados a datos e híbridos. Dentro de los métodos orientados a datos podemos distinguir según el uso de información etiquetada. En el caso de NTL, la etiqueta puede ser el resultado de una inspección de forma binaria o nominal. Incluso podría ser la cantidad de energía perdida u alguna otra variable de interés como veremos en el capítulo 4. A los métodos de aprendizaje automático orientados a datos que hacen uso completo de las etiquetas se los denomina métodos de aprendizaje supervisado. En NTL el uso de aprendizaje supervisado es uno de los enfoques más utilizados debido a que las empresas de distribución de energía realizan inspecciones *in situ* para el control de pérdidas y generan bases de datos etiquetadas. Si bien el acceso a estas bases es restringido, existen varios grupos de investigación a nivel mundial que trabajan en conjunto con empresas de distribución de energía y han tenido un acceso parcial a esta información [3, 10, 17, 39, 48, 56, 65, 91]. El acceso a esta información no es abierto por ser información sensible, no solo por la privacidad de los datos personales, sino también por incluir datos que revelan información del negocio de una empresa que, en muchos casos, se encuentran en competencia. Esto ha llevado a muchos investigadores a generar bases de datos que modelan el problema de forma sintética. Uno de los métodos más utilizados es tomar una base de datos de consumos reales sin fraude y de acceso libre o académico y simular fraude modificando las curvas de consumo [6, 20, 29, 47, 53, 84, 101]. La única base de datos pública con fraudes reales y datos diarios es la que hizo disponible Smart Grid Corporation of China (SGCC) [109]. Sin embargo, esta base no cuenta con las fechas de las inspecciones, por lo que no hay forma de segmentar los

intervalos de tiempo de datos válidos (los previos a las inspecciones). Complejas arquitecturas de aprendizaje profundo presentan excelentes resultados superando el 90 % de precisión sobre esta base [1, 44, 109], mientras que trabajos realizados por investigadores en conjunto con empresas de distribución de energía han alcanzado resultados reales en campañas de inspección entre el 15 % y el 47 % de precisión [11, 48]. Estas diferencias dejan en evidencia lo importante que es contar con datos reales y bien segmentados en el tiempo para generar bases de datos de entrenamiento. De no contar con la fecha de inspección, los algoritmos también aprenden sobre el comportamiento posterior a una inspección. Estos patrones no estarán presentes a la hora de realizar predicciones sobre nuevos datos. Dependiendo de la realidad de cada empresa, para algunas regiones o ciudades es probable que la cantidad de datos etiquetadas sea muy pequeña para aplicar algunos algoritmos de aprendizaje supervisado o directamente no se cuenta con dicha información. Encontramos también en la literatura abordajes al problema de NTL con algoritmos de aprendizaje no supervisado o semisupervisado [20, 46, 102].

La aparición de los medidores inteligentes generó nuevas oportunidades para el análisis de datos en el sector eléctrico. Aplicaciones vinculadas a predicción de carga, gestión de la demanda y análisis de la demanda han avanzado enormemente gracias a la aplicación de herramientas de aprendizaje automático. Una reseña completa sobre el uso de datos de medidores inteligentes en estos temas se puede leer en [104]. Dentro del análisis de consumos eléctricos, NTL es un problema muy desafiante con permanente actividad académica. El aumento exponencial de datos vinculados al uso de infraestructuras avanzadas de medidas (AMI) genera nuevos desafíos y oportunidades.

La revisión bibliográfica de los diferentes abordajes al problema se divide en tres secciones: i) un enfoque clásico, predominante hasta el año 2018, que incluye extracción de características, bases de datos de tamaño pequeño o moderado (unos pocos miles) y uso de algoritmos como *support vector machine* (SVM); ii) detección de pérdidas no técnicas basada en algoritmos de *bagging* y *boosting* como *random forest* o *extreme gradient boosting*; y iii) abordajes de NTL basados en aprendizaje profundo con el uso de redes neuronales convolucionales y redes neuronales recurrentes.

2.2. Enfoque clásico

En [69] se describe el enfoque aprendizaje supervisado clásico, donde, utilizando conocimiento del problema, se extraen características de los datos para luego entrenar un algoritmo de clasificación. Hasta 2018 este fue el enfoque predominante. Incluso en el último artículo de revisión publicado por la revista Energies en 2020 [90] no se hace referencia a trabajos con arquitecturas de aprendizaje profundo. Sin embargo, en los últimos tres años son varias la propuestas de aprendizaje profundo para NTL donde la extracción de características es realizada en las primeras capas de los algoritmos de clasificación. El uso de datos crudos ha generado muy buenos resultados. En el capítulo 3 se realiza una comparación entre el uso de características extraídas de los datos y su uso directo para el entrenamiento de mo-

delos de detección. Los modelos derivados de algoritmos de aprendizaje automático han sido criticados por ser una especie de caja negra y carecer de interpretabilidad para los técnicos que trabajan directamente en los problemas abordados. La ingeniería de características de alguna forma permitía generar una intuición de qué es lo que los algoritmos capturan como información relevante. El uso de datos crudos con redes neuronales dificulta aún más la interpretabilidad de los modelos. En la literatura no se han encontrado trabajos que aborden este tema para NTL. Utilizando ideas provenientes de las técnicas de ataques adversarios en la sección 3.3.4 de esta tesis, generamos un método para facilitar la interpretabilidad de los modelos entrenados. Y en la sección 6.4 se generan visualizaciones de capas de activación de redes convolucionales también con el objetivo de interpretar parcialmente los modelos.

Dentro del enfoque clásico de aprendizaje automático podemos distinguir las siguientes etapas: preprocesamiento de datos, extracción de características, selección de características, entrenamiento de modelos y clasificación.

2.2.1. Extracción de características

En el enfoque clásico de reconocimiento de patrones en datos, luego de que los datos son visualizados y preprocesados, se realizan operaciones sobre ellos para generar variables relevantes para el problema estudiado. La creación de estas características requiere de conocimiento específico del problema. En NTL se han propuesto y probado muchas características, sin embargo, algunas son comunes en diferentes trabajos. Algunas de las características forman parte de reglas de generación de inspecciones utilizadas históricamente por la compañías, por ejemplo, la variación de consumo interanual o la relación de consumo entre las estaciones del año. A continuación se listan algunas de las características extraídas de las curvas de consumo de energía de trabajos de referencia.

- Consumo promedio y desviación estándar [24, 65, 75, 76, 90].
- Consumo máximo y consumo mínimo [70, 76].
- Diferencia entre los coeficientes de la transformada de Fourier del último año respecto al anterior [24, 65].
- Diferencia entre los coeficientes de la transformada Wavelet del último año respecto al anterior [25, 27, 65].
- Coeficientes de aproximaciones polinómicas de la serie temporal de consumos [24, 65].
- Ratios de variación de consumo entre trimestres. Último trimestre comparado con diferentes períodos previos. [16, 24, 65, 95].
- Número de cruces por valor medio [27, 59, 70].
- Factor de potencia, relación entre consumo de energía activa y reactiva [87, 88].

Dada la variabilidad de usos de energía eléctrica y las modalidades posible de fraude este es un problema de clases no separables. Por más potente que sea un algoritmo la información proveniente de las curvas de consumo tiene un desempeño limitado. Es por eso que muchos trabajos incluyen dentro de los datos de análisis otras informaciones a las cuales las empresas de distribución de energía tienen acceso. La forma de utilizar esta información depende de los algoritmos que se utilizan para realizar los modelos de clasificación. Existen varias propuestas en la literatura, que incluimos en las siguientes secciones. Es importante considerar que más de la mitad de las publicaciones en el área trabajan con datos simulados, lo que dificulta enormemente el uso de información adicional de los clientes. La inclusión de información adicional busca capturar patrones de comportamiento fraudulento por fuera del efecto en la curva de consumo. Estos patrones pueden estar vinculados a actividades delictivas en ciertas regiones, a antecedentes de los clientes o incluso a actividades comerciales. Las revisiones del estado del arte de Messinis [69] y Saeed [90] no recogen el uso de estos datos adicionales. Se resumen a continuación algunas de las características utilizadas en la literatura:

- Información de ubicación geográfica:
  - Ciudad, región y/o barrio [11, 17, 39, 56].
  - Densidad de inspecciones de la región y prevalencia de fraude histórica de la región [10, 35].
  - Coordenadas geográficas de los medidores [11, 67].
- Información contractual:
  - Tipo de contrato (residencial, comercial) [10].
  - Rubro de negocio [10, 39].
  - Potencia contratada [39, 56, 65, 67].
  - Tensión de alimentación y cantidad de fases [17].
  - Retraso en el pago de facturas de energía [56, 63, 67].
  - Información de inspecciones previas [39, 56, 65].
  - Tipo de medidor [10, 17, 65].
- Información adicional:
  - Consumo de subestaciones [20, 43, 48].
  - Temperatura ambiente [48].

2.2.2. Selección de características

En algunos algoritmos como SVM la inclusión de características que aportan muy poca o nula información sobre el problema deteriora el desempeño de los

modelos. Por esta razón, en el enfoque clásico es necesario determinar qué subconjunto de características debe ser utilizado durante el entrenamiento del modelo. Costa [17] realiza selección de características por ganancia de información. Ramos et al. analizan en [85] varias técnicas de selección como: *particle swarm optimization* (PSO), *differential evolution* (DE) y *genetic algorithm* (GA). Kosut et al. [56] utilizan métodos *filter* y *wrapper* para la selección de características. Los métodos *filter* permiten evaluar la correlación de las características con las etiquetas y descartar aquellas que no aportan información suficiente. Los métodos *wrapper* permiten evaluar el desempeño de un algoritmo al ser entrenado con subconjuntos de características. Estos últimos se basan en estrategias de búsqueda *greedy* como *hill-climbing* o *best-first* [54]. Algunos algoritmos realizan la selección de características de forma intrínseca o embebida, es el caso de *random forest* ampliamente utilizado como un algoritmo de referencia por su desempeño tanto en precisión como en velocidad de entrenamiento.

2.2.3. Algoritmos de aprendizaje automático

SVM - Máquina de vectores de soporte

SVM es uno de los algoritmos de aprendizaje automático más populares, tanto para clasificación como para regresión. SVM busca generar hiperplanos que, en el espacio de características, generen la máxima separación entre clases. En un problema de clasificación binaria, si las muestras son linealmente separables se seleccionan dos hiperplanos paralelos que separen las muestras y se maximiza su distancia. Al maximizar la distancia entre hiperplanos unas pocas muestras quedan incluidas en dichos planos (en los margenes del espacio de separación). A estas muestras se les llama vectores de soporte. Sea el hiperplano separador $w^T x - b = 0$, y la etiqueta $y_i \in \{1, -1\}$ los vectores de soporte son las muestras de entrenamiento que cumplen $|w^T x_k - b| = 1$. Si la distancia $w^T x_k - b$ es mayor que uno, las muestras son clasificadas como positivas y si es menor que -1 como negativas.

Sin embargo, pocos problemas en la práctica son linealmente separables, al menos en el espacio de características. Para lidiar con este problema y permitir una clasificación con errores o con *outliers* existe una técnica llamada *soft-marging*. Básicamente se generan los hiperplanos permitiendo errores de clasificación que son ponderados por su distancia a los hiperplanos de toma de decisión. La tolerancia de los errores es ponderada por un parámetro C, el objetivo es minimizar la siguiente expresión

$$\frac{1}{N}\left[\sum_{i=1}^{N} max(0, 1 - y_i(w^T x - b))\right] + C||w||^2. \tag{2.1}$$

Valores pequeños de C permiten una mayor cantidad de errores pero pueden conducir a un modelo que generaliza mejor en datos no vistos. SVM encuentra su mayor potencial en problemas no separables linealmente al introducir el *kernel trick*. Si los datos no son separables linealmente podríamos aplicar transformaciones a las características generando nuevas características en un espacio de dimensiones

mayor donde sí fuesen separables. Podríamos, entonces, encontrar un hiperplano separador en este espacio de mayor dimensión. Sin embargo, aplicarle transformaciones polinomiales a todas las características y sus combinaciones en la práctica sería muy costoso. La función de *kernel* toma los vectores de entrada y devuelve el producto escalar de los vectores en el nuevo espacio de características, esto permite evitar el cálculo explícito. Uno de los *kernels* más utilizados en SVM es el *kernel* Gaussiano RBF (gaussian radial basis function). El radio de influencia de los vectores de soporte está dado por el parámetro gamma. Valores grandes de *gamma* definen un radio pequeño disminuyendo el efecto del uso del *kernel*.

En detección de NTL los modelos SVM han sido ampliamente utilizados tanto SVM lineal como RBF-SVM y variantes como CS-SVM, donde se puede asignar un costo diferente a cada clase, o OneclassSVM, donde se entrena un modelo solo con la clase negativa (la mayoritaria) y se busca detectar *outliers* [21, 22, 24, 43, 49, 52, 53, 75, 87]. En la práctica, entrenar RBF-SVM implica realizar una búsqueda de parámetros C y gamma que puede llegar a ser muy costosa. En particular, todos los trabajos que aplican SVM en NTL utilizan bases de datos de pequeño o mediano porte, de unas pocas miles de muestras como máximo. El algoritmo no escala bien al aumentar significativamente la cantidad de muestras. Esto se debe a que SVM es resuelto con programación cuadrática y esto da un orden entre $O(n_f \times n_s^2)$ y $O(n_f \times n_s^3)$ siendo n_f la cantidad de características y n_s la cantidad de muestras.

Árboles de decisión

Los árboles de decisión son algoritmos muy versátiles que se utilizan en infinidad de problemas. Se entrenan con facilidad y además permiten extraer reglas simples para la comprensión del problema que se aborda. A diferencia de otros algoritmos, su salida puede ser interpretada y visualizada de forma simple. Además, los árboles de decisión permiten incluir variables categóricas que, en algunos problemas como el de NTL, son muy utilizadas (en NTL pueden ser: tipo de medidor, tipo de contrato, rubro de negocio, etc.). Los árboles de decisión van dividiendo las muestras de forma recursiva, seleccionando aquellas características que aportan más información. Para esto utilizan el coeficiente de Ginni o medidas de ganancia de información. Los más utilizados en NTL han sido C4.5, CART y QUEST [16, 25, 56, 59, 70]. Algoritmos que combinan varios árboles de decisión formando un ensamble de clasificadores, muy utilizados en los últimos años, son presentados en la siguiente sección.

Redes neuronales artificiales (ANN)

Si bien en los últimos cinco años las redes neuronales han tenido un auge muy importante, el desarrollo de esta teoría tiene casi 80 años. El primer modelo computacional de redes neuronales, inspirado en cómo las neuronas biológicas transmiten señales para generar lógica en base a estímulos, fue presentado en 1943 por McCulloch y Pitts [68]. En 1958 Frank Rosemblatt propone el Perceptrón, un algoritmo para reconocimiento de patrones basado en el uso de dos capas de neuronas artificiales. El uso de estos modelos tuvo varios avances a lo largo del

tiempo, pero durante décadas uno de los principales problemas fue la falta de suficientes datos para entrenar y un costo computacional muy elevado. Fueron varios los avances tecnológicos y algorítmicos que generaron nuevos impulsos en el uso de las redes neuronales, en particular, en la década de 1980 con el entrenamiento por *backpropagation* (un método eficiente para minimizar la función de error por descenso por el gradiente). Pero el surgimiento de SVM concentró la atención de la comunidad científica en torno a los algoritmos de reconocimiento de patrones y sus fundamentos matemáticos. En el siglo XXI, el desarrollo de internet y sus aplicaciones dio lugar a la generación de bases de datos de gran porte para el entrenamiento de redes neuronales. El incremento en la capacidad de cómputo de las CPU que se ha duplicado cada dos años desde 1990 (Ley de Moore) permitió que nuevamente se pusiera foco en el desarrollo de arquitecturas de redes neuronales. Pero, sin duda, uno de lo mayores avances tecnológicos para el crecimiento de las aplicaciones con redes neuronales ha sido el uso de tarjetas gráficas (GPU) para el cómputo en el entrenamiento de estos modelos.

Básicamente el modelo de una neurona es una combinación lineal de las entradas más un *bias* y una función de activación no lineal:

$$h(X) = g(\sum_{i=1}^{n} w_i x_i + b), \tag{2.2}$$

donde x_i representa el vector de entradas, w_i los pesos entrenables, b el *bias* y a la función g se le llama función de activación. El uso de funciones de activación no lineales permite que se puedan generar grafos acíclicos direccionados de funciones capaces de presentar modelos más complejos. El perceptrón multicapa (MLP por su sigla en inglés) es una de las arquitecturas más utilizadas en aprendizaje automático. En particular, dentro del problema de detección de pérdidas no técnicas, ha sido utilizado como clasificador [17, 63, 81] y como regresor [29]. Ford et al. [29] utilizan un MLP para hacer *forecasting* del consumo de cada cliente minimizando RMSE (*root mean square error*): si la diferencia de previsiones supera un cierto umbral se clasifica como fraudulento. Este trabajo además es uno de los primeros en realizar simulaciones de fraude sobre la base de datos de consumos de medidores inteligentes de Irlanda CER [14]. Esta base de datos de medidores inteligentes ha sido ampliamente utilizada para la síntesis de fraudes y también es utilizada en el capítulo 6 de esta tesis. MLP es utilizado como clasificador de referencia en esta tesis en los capítulos 3 y 4.

Optimum path forest (OPF)

OPF es un algoritmo de aprendizaje automático basado en grafos propuesto por Papa et al. [80] en 2009. Cada muestra representa un nodo de un grafo cuyas coordenadas son los valores de las características. Un camino (*path*) es una secuencia de nodos del grafo y se asigna una función de costo a cada camino. Cada arco del grafo tiene asignado un costo dado por la distancia entre muestras, la distancia entre muestras de clases diferentes es infinita. Los caminos de menor distancia son denominados como prototipos, clasificar una nueva muestra es encontrar el prototipo con la menor distancia que incluye la nueva muestra. Este algoritmo ha sido

utilizado en varios trabajos de detección de NTL dentro del grupo de investigación de su autor en la Universidad de Campinas [85–88].

2.3. Detección de NTL basada en ensambles de clasificadores

Los métodos de ensamble de clasificadores están basados en la idea de que la combinación de varios modelos puede dar mejores resultados que la predicción de un único modelo. De hecho, en la práctica, métodos de ensamble como *random forest* han mostrado ser algoritmos muy robustos y con desempeños muy competitivos en varias áreas [32]. Los ensambles de árboles de decisión han sido algoritmos muy utilizados en los últimos tres años para abordar el problema de detección de NTL, ya sea entrenados directamente con los datos [41, 84] o combinados con algoritmos de aprendizaje profundo [1, 60].

2.3.1. *Bagging*

Una forma de generar diferentes modelos utilizando un mismo tipo de algoritmos es entrenar diferentes instancias sobre subconjuntos de los datos. El nombre *bagging* viene de *boostrap aggregating*. Las muestras para conformar cada subconjunto son muestreadas con remplazo (*boostrapping*), o sea que en el subconjunto habrá muestras repetidas dándole mayor peso a algunas de ellas de forma aleatoria. Luego de que todos los modelos son entrenados, la predicción de una nueva muestra se hace agregando las salidas. Para clasificación se puede utilizar la moda (*hard voting*) y para regresión el promedio de las salidas. Este agregado de resultados reduce tanto la varianza como el sesgo de la predicción [32].

Random forest (RF) es una implementación optimizada de un ensamble de árboles de decisión. En general, entrenado con el método de *bagging*, si bien tiene la opción de realizar muestreo sin remplazo (en ese caso método *pasting*). En particular *random forest* agrega otra aleatoriedad al modelo al realizar una selección aleatoria de las características utilizadas para tomar decisiones en cada nodo del árbol. En [91] se ataca el problema de detección de fraude en redes eléctricas con *bagged trees* y se compara su desempeño con RF y SVM. Es posible hacer todavía más aleatorio el entrenamiento de un ensamble de arboles de decisión si, además, en cada nodo el umbral de decisión para las características también es aleatorio. A estos ensambles se les llama *extra-trees* [33]. Gunturi et al. (2021) [41] prueban varios métodos de ensambles de árboles de decisión sobre fraude simulado en la base CER (se utilizan 48 datos por día). Reportan el desempeño de los métodos de *bagging*, RF y *extra-trees*, y los comparan con ensambles del método *boosting*. En [60] se propone un *framework* que combina una primera etapa de extracción de características con redes convolucionales (CNN) y luego entrena un clasificador *random forest*. En dicho trabajo también se realiza simulación de fraude sobre datos de consumo de medidores inteligentes de la base CER. En esta tesis se utiliza el algoritmo *random forest* en el capítulo 4 como clasificador y como regresor para

estimar el retorno económico de las inspecciones. En el capítulo 3 se utiliza una base de datos de 311k inspecciones reales para comparar desempeño entre algoritmos de ensamble (*random forest* y *gradient boosting*) con algoritmos de redes neuronales.

2.3.2. *Boosting*

Boosting hace referencia a cualquier ensamble de algoritmos "débiles" (*weak learners*), que combinados generan un algoritmo más robusto y de mejor desempeño. En general, los métodos de ensamble *boosting* son algoritmos de entrenamiento secuencial donde cada *weak learner* es entrenado con el objetivo de resolver los errores cometidos por su antecesor. Uno de los algoritmos de *boosting* más utilizados *gradient boosting* que puede ser utilizado tanto para clasificación como para regresión. Cada nuevo estimador es entrenado para ajustar el residuo del modelo anterior de forma de que la suma de las salidas de cada *weak learner* (en este caso árboles de decisión) se aproxime al valor objetivo. Veamos el caso de regresión, dado un vector de entrada x_i buscamos estimar el valor y_i y la estimación $\hat{y}_i$ es realizada por M estimadores

$$\hat{y}_i = F_M(x_i) = \sum_{m=1}^{M} h_m(x_i), \tag{2.3}$$

donde h_m es un árbol de decisión.

Podemos escribir la salida de cada etapa de forma *greedy*

$$F_m(x_i) = F_{m-1}(x_i) + h_m(x_i). \tag{2.4}$$

Entrenar en el siguiente paso implica minimizar una función de pérdida (*loss*)

$$h_m = \underset{h}{arg\,min} \sum_{i=1}^{n} l(y_i, F_{m-1}(x_i) + h_m(x_i)). \tag{2.5}$$

Supongamos que utilizamos el error cuadrático medio como función de *loss* (MSE) $L = \frac{1}{n}(y - F(x))^2$. Es fácil ver que el residuo es proporcional al negativo del gradiente de la función de *loss*

$$\frac{\delta L}{\delta F} = -\frac{2}{n}(y - F(x)) = -\frac{2}{n}h_m(x). \tag{2.6}$$

En cada iteración el residuo es minimizado calculando el gradiente de la función de costo. Esto puede verse como un descenso por el gradiente, de allí el nombre del algoritmo. En los últimos tres años *gradient boosting* y en particular sus implementaciones optimizadas *extreme gradient boosting* (XGB) y *light gradient boosting* (LigthGB) se encuentran entre los algoritmos más utilizados para abordar el problema de NTL. En [84] se combina el uso de XGB, LigthGB y CatBoost para detección de fraude simulado sobre la base CER. Buzau et al. [10] trabajan con una base de datos de 57k clientes de ENDESA con inspecciones e integran diferentes

fuentes de información para entrenar modelos de clasificación. Muestran que XGB obtiene mejores resultados que SVM, vecinos más cercanos (KNN) y regresión logística (LR). Gunturi [41] compara el desempeño de diferentes ensambles incluyendo XGB, LigthGB, CatBoost y Adaboost. Este último algoritmo, Adaboost, a diferencia de los anteriores no trabaja sobre el residuo sino que directamente modifica el peso de las muestras con errores en la predicción. De esta forma, en la siguiente iteración el estimador definirá las reglas de decisión ponderando los errores previos. En [60], además de combinar CNN con RF, también muestran que combinar CNN con *gradient boosting* genera resultados igual de competitivos. El problema de desbalance implícito en el problema de detección de fraude puede ser abordado de diferentes maneras (ver 2.6). En [3] utilizan rusboost (*random under sampling boosting*) como clasificador para tratar NTL. En cada iteración del algoritmo, las muestras son subsampleadas para entrenar un árbol con clases balanceadas.

El el capítulo 3 de esta tesis se realiza una comparación con diferentes abordajes del problema de detección de NTL sobre datos de consumos mensuales y características extras. Se comparan algoritmos de aprendizaje automático con algunas arquitecturas de redes neuronales, los mejores resultados son alcanzados por XGB. En el capítulo 5 de esta tesis se utiliza XGB como regresor y como estimador de probabilidades.

2.4. Detección de NTL basada en redes neuronales y aprendizaje profundo

2.4.1. Redes neuronales de convolución - CNN

Las redes neuronales de convolución surgen del estudio de la corteza visual del cerebro humano en 1980 con el neocognitron [31]. Estas redes tienen un gran avance en 1998 cuando Yan Lecun las incluye en un algoritmo de reconocimiento de documentos utilizando *back propagation* y capas de *pooling* [58]. El método de *back propagation* permite calcular el gradiente de la función de pérdida respecto a cada parámetro entrenable de la red neuronal. Se puede pensar como la aplicación de la regla de la cadena de derivadas parciales en funciones anidadas. Teniendo los valores de las derivadas se pueden aplicar diferentes algoritmos de descenso por el gradiente para minimizar la pérdida.

Las redes convolucionales en 2D han demostrado ser muy potentes para identificar patrones en imágenes, superando ya desde hace algunos años la capacidad humana para detectar objetos sobre bases de datos estandarizadas como Imagenet [89]. Las CNN reducen drásticamente la cantidad de parámetros entrenables en comparación con las redes *fully conected* (de los MLP). Esto se basa en el hecho de que las imágenes tienen estructuras locales (líneas, formas, colores, etc.), que pueden ser capturadas por un conjunto de filtros (*kernels*). La detección de pequeñas partes permite codificar estructuras más complejas de forma jerárquica. Cada filtro es equivalente a una neurona que tiene conexiones solamente con una región de la

imagen. La operación de convolución permite que cada filtro opere con todas las zonas de la imagen compartiendo los valores de los parámetros entrenables (*kernel weights*).

Para el caso de una imagen de dos dimensiones I, la discretización de la operación de convolución con un *kernel* K se presenta en la siguiente ecuación

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n),$$

donde m y n son las dimensiones del *kernel*, (i,j) las coordenadas de un punto de la entrada y S la capa de salida de la convolución. Cada capa de convolución está formada por un banco de filtros que se caracteriza por sus dimensiones de largo y alto, típicamente 3x3, y la profundidad queda dada por la dimensión del volumen de datos a la entrada. En el caso de aprendizaje profundo se suelen utilizar varias capas de convolución concatenadas.

La activación de cada neurona está dada por una función no lineal, por ejemplo *rectified linear unit* (ReLU). Las no linealidades permiten representaciones más complejas y son un punto clave del éxito del aprendizaje profundo.

Las capas de convolución son el bloque principal de una arquitectura de un clasificador CNN. Otras capas muy utilizadas en un clasificador CNN son: *pooling, batch normalization, dense* y *softmax* (o sigmoide para dos clases). Las capas de *pooling* permiten reducir la dimensionalidad de los datos. Una vez que se extraen características de los datos se puede entrenar un clasificador tipo MLP concatenando algunas capas densas. Para hacer esto, se aplana el volumen de datos de la última capa de convolución y se usa como entrada de la primera capa totalmente conectada.

Señales codificadas como series temporales pueden ser representadas como imágenes con el objetivo de utilizar el potencial de las redes CNN en dos dimensiones. Un claro caso de esto es el uso de espectrogramas para el análisis de señales de audio. Los mejores resultados en *speach recognition, music information retrial, source separation* se basan en el uso de representaciones en 2D de señales de audio. En NTL se ha propuesto realizar una representación matricial del consumo de cada cliente asignando el consumo de cada semana a una fila diferente de la matriz [109]. Este enfoque fue propuesto para datos de SGCC, dado que los datos son de consumos diarios, los datos de entrada a la red tienen dimensión (S,7) siendo S la cantidad de semanas de la base. En [105] utilizan aprendizaje profundo con CNN para 148 semanas de datos de SGCC. Hasan et al. [44] complementan este enfoque utilizando CNN para extraer características sobre los mismos datos y agregando a continuación algunas capas de LSTM (long short term memory). Sin embargo, la mayoría de los patrones de uso de energía acontecen con resoluciones menores a un día. Por lo que un enfoque con representación matricial de los consumos pero de mayor resolución temporal podría aprovechar de mejor forma la potencia de herramientas como las redes de convolución en dos dimensiones. Siguiendo esta línea, en esta tesis se propone utilizar datos de consumos de mayor frecuencia de muestreo para generar imágenes que representen los perfiles de consumo de los clientes. Utilizando 96 muestras de consumo diario durante un período

de tres meses se forman imágenes de 90x96 (capítulo 6). La invarianza espacial generada por la operación de convolución en 2D permite, con este enfoque, identificar patrones anómalos con independencia al instante de su ocurrencia, a diferencia de lo que ocurre con las MLP y otros métodos de aprendizaje automático donde cada entrada es una característica.

Las redes de convolución también son utilizadas en NTL para el tratamiento de series temporales en 1D de baja resolución. En [6] utilizan como datos de entrada los últimos 12 consumos mensuales y muestran que, en su simulación, CNN obtiene un mejor desempeño que otros algoritmos como RF, MLP y LSTM. En [1] se propone el uso de una arquitectura AlexNet de redes de convolución para la extracción de características y luego la clasificación se hace utilizando el algoritmo de *boostring* LGB.

2.4.2. Redes neuronales recurrentes - RNN

A diferencia de las redes *feedforward* (MLP, CNN), las redes neuronales recurrentes (RNN por su sigla en inglés) son diseñadas específicamente para trabajar con secuencias de datos de largo variable. Estas redes han sido ampliamente utilizadas para procesamiento del lenguaje natural (NLP), para predicción climática y, en general, para el tratamiento de series temporales. RNN no son las únicas redes capaces de manejar secuencias de datos. MLP puede ser utilizada para manejar secuencias cortas y para manejar secuencias muy largas puede utilizarse CNN [32], como ya vimos en este capítulo. En las redes *feedforward* las activaciones solo avanzan en dirección entrada-salida. En las redes recurrentes, las activaciones también tienen conexiones hacia atrás, lo que permite generar memoria sobre los datos previamente utilizados en la secuencia. En una red MLP cada entrada tiene asociado un conjunto de parámetros de entrenamiento, las RNN comparten parámetros con todos los datos de la secuencia. Una celda de estado interna de cada bloque RNN permite ir acumulando información de contexto de toda la secuencia de datos. El estado interno es actualizado en el tiempo, utilizando el estado interno anterior y la entrada actual con una función de recurrencia

$$h_t = f_W(h_{t-1}, x_t), \tag{2.7}$$

donde h_t es el estado interno en el instante t y x_t la entrada. Notar que los pesos W de la función no dependen de t. Para el entrenamiento de los modelos con redes recurrentes se utiliza BPTT (*back propagation through time*). El gradiente de la función de *loss* respecto a los parámetros del modelo es calculado para cada instante de tiempo, comenzando por el último, y el gradiente total es la suma de cada una de las contribuciones de la secuencia. La red recurrente más utilizada en la actualidad es la LSTM (*long short term memory*) [45] que apareció para mitigar el problema de desvanecimiento del gradiente que generan las redes recurrentes. En esta arquitectura, además de tener un estado interno h_t, se tiene una celda de memoria c_t. La información que se acumula en memoria es controlada por dos compuertas f_t (*forget gate*) e i_t (*input gate*) que son operadas por un conjunto de parámetros entrenables del modelo como se ve en la figura 2.1. La salida viene de la celda de memoria modulada por la compuerta o_t.

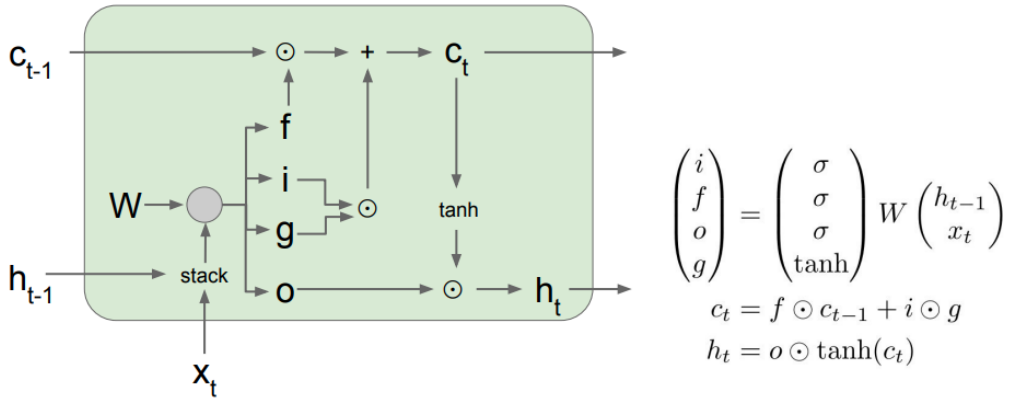


Figura 2.1: Grafo computacional de una celda de la red LSTM (imagen extraída de la lectura 10 del curso CS231n de Stanford University).

Arquitecturas de aprendizaje profundo basadas en LSTM también han sido utilizadas para abordar el problema de NTL. En [11] proponen el uso de una arquitectura de dos entradas. En la primera entrada hay una red LSTM que es alimentada con vectores de datos semanales que contienen el consumo diario, la cantidad de medidas nulas y los datos faltante de dicha semana. La segunda entrada son variables categóricas normalizadas que alimentan un MLP. Los datos utilizados en dicho trabajo corresponden a 105k clientes de la empresa española ENDESA. Se utiliza como métrica de desempeño el área debajo de la curva precision-recall. Los resultados experimentales son comparados con los ensambles RF, XGB y con Wide and Deep [109]. Otras redes recurrentes también han sido utilizadas en NTL. En [72] se utiliza GRU (gater recurrent units), que, al igual que LSTM, resuelve el problema de desvanecimiento del gradiente pero utilizando únicamente dos compuertas.

En [48] se presenta un trabajo con la corporación china de redes inteligentes (SGCC) donde los autores accedieron a una base de datos de 300k clientes (más completa que la disponible a nivel público [109]). En dicho trabajo proponen una arquitectura con tres redes recurrentes LSTM para tres entradas de series temporales diferentes: consumos individuales de SM, consumo de subestaciones y temperatura ambiente. Este trabajo también se compara con los algoritmos RF, XGB y Wide and Deep utilizando como métrica de desempeño f_β con $\beta = 0{,}5$. Si bien hay enfoques para incluir nueva información en arquitecturas de aprendizaje profundo, no ha habido aportes a la literatura de NTL para utilizar medidas en multiresolución y abordar los períodos donde las tecnologías de medida conviven. Estos períodos pueden durar años, por lo que el uso de esta información puede ser muy valioso para la compañías de distribución de energía. En el capítulo 6 de esta tesis se presenta una propuesta en esta línea.

2.5. Abordajes complementarios en NTL

Se incluyen en esta sección algunos abordajes complementarios de NTL, ya sea por el uso de algoritmia no incluida en las secciones previas o por el tratamiento de datos adicionales. La Universidad de Sevilla ha realizado varios trabajos en colaboración con la empresa de distribución de energía ENDESA. Guerrero et al. [39] se proponen mejorar el desempeño obtenido al inspeccionar clientes que tienen algún consumo nulo (regla previa de generación de inspecciones). Para ello, incluyen como fuente de datos las anotaciones realizadas por los inspectores en sus visitas a las instalaciones de los clientes. Este es el único trabajo que aplica *text mining* en NTL. Se generan diccionarios (incluso en varios idiomas: español, catalán, etc.) y se utilizan técnicas de NLP para realizar un primer filtrado de clientes. Luego se extraen reglas de un árbol de decisión y se combinan con *clustering* hecho con SOM (self organized maps). Otro trabajo reciente que realiza una primera etapa de filtrado con información adicional es [20] donde se estiman las pérdidas de energía de las subestaciones realizando estimaciones del estado estático de la red con datos de PMUs (phasor measurement units). Los datos derivan de simular una red con el estándar IEEE70 y simular fraude sobre datos de consumos de la base irlandesa CER. Una vez seleccionadas las subestaciones se generan *clusters* con SOM y un modelo de clasificación binaria (normal/fraudulento) con MLP para cada *cluster*.

Las redes GAN (generative adversarial networks) [37] han tomado un gran interés académico en los últimos años por su gran habilidad para generar datos con la misma estadística que los datos de entrada. Son redes que se entrenan de forma no supervisada. Las GAN están compuestas de dos redes neuronales: un generador y un discriminador. El generador es entrenado para crear datos similares a los datos de entrada, que puedan confundir al discriminador mientras que el discriminador es entrenado para detectar estos datos falsos. Para abordar el problema NTL Hu et al. [47] utilizan una red BiWGAN (bidirectional Wasserstein GAN) que permite entrenar un *encoder* para extraer características relevantes de los datos. Estas características son utilizadas para entrenar un clasificador SVM de una clase. La propuesta es probada con datos sintéticos y se compara con el uso de otros tipos de redes GAN.

Poco se ha escrito sobre la privacidad de datos en NTL. Algunas compañías, si bien cuentan con información de sus clientes, no están habilitadas legalmente a utilizarla sin consentimiento expreso del cliente o pueden utilizar datos con una resolución menor a la que poseen. En [105] se propone el uso de una técnica de cifrado homomórfico para preservar la privacidad de los datos utilizados por los algoritmos de NTL.

2.6. Métricas de desempeño y desbalance

2.6.1. Métricas utilizadas en NTL

La detección automática de pérdidas no técnicas debe generar como salida una lista de clientes a ser inspeccionados. Por lo que, independientemente de los diferen-

tes abordajes que pueda tener el problema, es necesario generar una clasificación binaria: inspeccionar o no. Desde esta perspectiva podemos definir el problema como un problema de clasificación de dos clases: los usuarios normales y los usuarios con NTL (o irregulares). Una irregularidad es un fraude, un hurto o un medidor dañado por una causa ajena al usuario. En clasificación binaria se suelen definir las clases como positivos (P) y negativos (N), esto se debe históricamente a su uso en la predicción y testeo de enfermedades. Un sistema automático de detección de NTL puede generar dos tipos de errores, inspeccionar a un usuario normal (falso positivo) o no inspeccionar a uno irregular (falso negativo). En este punto podemos preguntarnos: ¿cuál es la forma de medir si un algoritmo está haciendo bien su trabajo? La métrica más utilizada en problemas de clasificación es la exactitud (*accuracy* en inglés). Pero el problema de NTL es un problema de clases desbalanceadas, la cantidad de positivos es mucho menor que la cantidad de negativos. Supongamos que existe una prevalencia del 5 % de usuarios con irregularidades, un sistema que clasifica a todos los clientes como negativos (normales) obtendría una exactitud del 95 %. Este podría ser un excelente valor, pero no hay que realizar ninguna inspección y las pérdidas no técnicas permanecen igual o continúan en aumento por la falta de controles y sanciones. Claramente es necesario tener métricas que evalúen el desempeño poniendo énfasis en la clase positiva. La precisión de las tareas de inspección es un dato muy relevante, es medir qué porcentaje de las inspecciones realizadas son efectivamente irregularidades (true positives). Por otra parte, si solo se realizan inspecciones a 10 clientes y el algoritmo alcanza una precisión del 100 % se puede estar recuperando un porcentaje muy bajo de las NTL que no genere un impacto real en el negocio. Siguiendo con la lógica binaria, otra métrica relevante para problemas de clases desbalanceadas es *Recall*, también llamada *sensitivity* o *True Positive Rate* (TPR). *Recall* es el porcentaje de detección de positivos en el universo de todos los positivos.

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

$$Recall = \frac{TP}{P} = \frac{TP}{TP + FN}.$$

Donde TP (true positives) representa el número de positivos con predicción acertada, TN (true negatives) es el número de negativos con acierto y FP (false positives) y FN (false negatives) la cantidad de errores en cada clase. La precisión tiene en cuenta el error de predicción de la clase positiva mientras que *recall* incluye los positivos no detectados. Es fácil notar que si se pudiera inspeccionar a todos los clientes se obtendría $Recall = 1$. Claramente es necesario encontrar un balance entre estas métricas. La media armónica es una estrategia utilizada como forma de generar un balance entre *precision* y *recall*, esta métrica se llama $F_{measure}$.

Una versión de esta métrica con la media armónica ponderada se denomina F_β y permite priorizar alguno de los dos errores de clasificación. $F_{measure}$ es un caso particular de F_β cuando $\beta = 1$.

$$F_\beta = (1 + \beta^2)\frac{Recall \times Precision}{\beta^2 Precision + Recall}$$

Esta métrica fue propuesta para considerar el desbalance en NTL [24], seleccionando valores de $\beta < 1$ de forma de priorizar los resultados de precisión. Siguiendo esta línea en [48] se utiliza como métrica $f_{0,5}$.

Estas métricas, como otras tantas basadas en la matriz de confusión, permiten comparar resultados dado un umbral de decisión. O sea, es necesario definir un punto de trabajo para calcular las métricas. En algunos problemas donde el punto de trabajo no es conocido o involucra variables que exceden el análisis, se pueden utilizar métricas de desempeño de comparación global de algoritmos. La más conocida es el área debajo de la curva ROC (*receiver operating characteristic*). La curva ROC se obtiene al graficar la proporción de positivos encontrados (TPR) contra la proporción de falsos positivos (FPR) variando el umbral de decisión.

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN}$$

ROC-AUC ha sido utilizada en varios trabajos previos en NTL [3, 10, 35, 109] y se utiliza en esta tesis para comparar desempeño de modelos y realizar ajuste de hiperparámetros en los procesos de entrenamiento. Sin embargo, algunos autores sostienen que para problemas de clasificación binaria con datos desbalanceados es más adecuado el uso del área debajo de la curva *Precision-Recall* (PR-AUC) [32, 92]. PR-AUC ha sido poco utilizada para NTL, esta métrica es reportada en [1,11] y es utilizada para realizar la selección de modelos en [11]. En el capítulo 6 de esta tesis se monitorea PR-AUC para controlar el sobreajuste de los modelos durante el entrenamiento y como métrica de selección de modelos en el proceso de ajuste de hiperparámetros.

Los algoritmos de clasificación que son capaces de generar un puntaje (*score*) a la salida, además de permitir el cálculo de métricas donde se desplaza el umbral de decisión (ROC-AUC y PR-AUC), permiten realizar clasificación por *ranking*. Algunas métricas de *ranking* han sido utilizadas y reportadas en NTL. Es el caso de $MAP@N$ (*mean average precision at N inspections*), que computa la precisión promedio para un número dado de inspecciones, el *top* N del *ranking* de clasificación [109]. En el capítulo 6 se reporta otra métrica de *ranking*, $P@10\,\%$ la precisión obtenida al clasificar como positivas el $10\,\%$ de muestras de la base de evaluación.

Sin embargo, y más allá de las métricas de evaluación que se utilicen, lo que está implícito detrás de este tipo de problemas con desbalance de clases, es el costo asociado a la toma de decisión. ¿Por qué no utilizar la exactitud? Porque el costo de no detectar un positivo (FN) es mayor que el costo de errar una predicción positiva (FP). El problema económico, que es la principal motivación de los trabajos académicos citados en este capítulo, no había sido tratado como parte de un *framework* de detección automática de NTL previo al trabajo realizado en esta tesis. En el capítulo 4 se compara el desempeño de un sistema de clasificación al definir el umbral de clasificación maximizando un $F_{measure}$ o definirlo realizando una clasificación sensible a costos que involucre una estimación del retorno económico conjuntamente con modelos de costos de las actividades. En el capítulo 5 se

ahonda en el modelado de costos, incluyendo técnicas de optimización de ruteo de vehículos.

2.6.2. Tratamiento del desbalance

Conjuntamente con una selección adecuada de las métricas de evaluación, también puede ser necesario definir un método de tratamiento del desbalance de clases en los datos de entrenamiento. Los métodos de tratamiento del desbalance pueden ser divididos en tres categorías: i) métodos a nivel de datos, ii) métodos a nivel de algoritmo, y iii) métodos híbridos [57]. En la primera categoría se encuentran los métodos más utilizados en problemas de clasificación con desbalance: métodos de submuestreo, sobremuestreo o métodos combinados de muestreo.

Métodos de balance a nivel de datos

La selección del método adecuado depende de la naturaleza de los datos. Si se cuenta con suficientes datos se puede submuestrear la clase mayoritaria. En NTL uno de los mayores problemas es el costo de generar grandes bases de datos, por lo que esta estrategia podría estar descartando información relevante del problema. En [34] se utiliza submuestreo a diferentes tasas para evaluar su impacto sobre una gran base de datos. Un método de sobremuestreo ampliamente usado en problemas con desbalance es SMOTE (*synthetic minority oversampling technique*) [15]. El sobremuestreo de la clase minoritaria puede generar patrones que no representen necesariamente un elemento de la clase fraudulenta. Sin embargo, su uso en NTL es bastante extendido y ha reportando buen desempeño en algunos trabajos recientes [41, 44, 60]. Un método combinado de muestreo es utilizado en [1] con el nombre SMOTEENN. Este método sobremuestrea la clase minoritaria con SMOTE y submuestrea la clase mayoritaria eliminando aquellas muestras que dentro de sus vecinos tengan más muestras de la clase minoritaria (aquellas muestras más cercanas a la frontera), utiliza KNN (k-*nearest neighbors*). El sobremuestreo de la clase minoritaria también puede realizarse utilizando GANs (*generative adversarial networks*). Con este método una red neuronal aprende a generar nuevas muestras que siguen la distribución de probabilidad de los datos de entrada [37]. Aldegheishem et al. [1] proponen para NTL una arquitectura que combina dos tipos de redes GAN: *conditional generative adversarial network* (CGAN) y *wasserstein generative adversarial network gradient penalty* (WGAN-GP) mostrando que la combinación de ambas técnicas obtiene mejores resultados que su uso individual. Esta técnica es utilizada en una etapa de preprocesamiento, por lo que entra en la categoría i) de tratamiento del desbalance a nivel de datos.

Métodos de balance a nivel de algoritmos

Dentro de los métodos que trabajan el desbalance a nivel de algoritmo, se destaca el uso de clasificación sensible a costo. En clasificación sensible a costo se puede asignar un peso a cada muestra o directamente un peso a cada clase. El

más utilizado para reducir el sesgo hacia la clase mayoritaria de los clasificadores es asignar un peso mayor a los errores de la clase minoritaria. Este método evita la degradación de los datos generada por las técnicas de muestreo. En [47] se utiliza este enfoque para tratar el desbalance. El peso de la clase minoritaria es asignado con el ratio de muestras por clase de la base de entrenamiento ($\#negativos/\#positivos$).

Métodos de balance híbridos

En los métodos híbridos se realiza muestreo de datos dentro del algoritmo de entrenamiento. Se destaca el muestreo en ensambles de clasificadores como, por ejemplo, RUSBoost (*random undes samplig adaBoost*) [93] utilizado por Ávila [3] en NTL.

En algunos trabajos que utilizan aprendizaje profundo en NTL, para realizar una clasificación por *ranking* no se utiliza ningún método de control del desbalance [11, 109]. Sin embargo, la estimación de probabilidades de pertenencia a cada clase se ve afectada por el desbalance [11].

En esta tesis se trata el problema de desbalance utilizando clasificación sensible a costos y métricas de evaluación que contemplan el desbalance.

Capítulo 3

Comparativa de enfoques clásicos y basados en aprendizaje profundo.

3.1. Introducción

En la última década la comunidad de aprendizaje automático ha ofrecido varias soluciones al problema de detección de NTL [69]. Sin embargo, uno de los principales obstáculos consiste en recopilar y acceder a datos etiquetados para evaluar y comparar la validez de las soluciones propuestas. En colaboración con UTE, recopilamos datos de 311 mil clientes con inspecciones en el sistema de suministro y medición de energía, creando uno de los conjuntos de datos completamente etiquetados para uso académico más grandes del mundo. Las inspecciones fueron realizadas por personal experto durante seis años, y cada cliente recibió al menos una inspección *in situ* por parte de un electricista certificado. Los datos se recopilaron en todo el territorio de Uruguay entre enero de 2014 y julio de 2020.

En el presente capítulo, explotamos de dos maneras estos valiosos datos recopilados. Primero, revisamos trabajos previos, comparamos y validamos los hallazgos probados en bases de datos mucho más pequeñas y menos diversas. En segundo lugar, comparamos y analizamos nuevos algoritmos de redes neuronales profundas, que se han adoptado más recientemente para prevenir NTL [47, 60, 109]. Comenzamos describiendo los datos recopilados y los algoritmos probados. Luego presentamos y discutimos los resultados obtenidos. Seguimos analizando las vinculaciones con el estado del arte y concluimos discutiendo nuestros hallazgos y perspectivas para el futuro.

3.2. Métodos

3.2.1. Datos

Se recopiló un conjunto de datos completamente etiquetado de clientes residenciales y comerciales en un esfuerzo conjunto entre la Facultad de Ingeniería y UTE. 311k clientes fueron inspeccionados por electricistas certificados para eva-

luar cualquier signo de falla o actividades fraudulentas en la instalación eléctrica y el medidor del cliente. Las inspecciones se realizaron en todo el país, entre enero de 2014 y julio de 2020. En 36,6k de las 311k inspecciones (11,8 %) se detectaron irregularidades. Después de la inspección, si se detectaba alguna anomalía, se asignaba la etiqueta *irregular* al cliente y, de lo contrario, se etiquetaba al cliente como *normal*. Además del consumo eléctrico mensual y su etiqueta, utilizamos para cada cliente información adicional: *potencia contratada* representa la potencia máxima contratada por el cliente; *(latitud, longitud)* la ubicación geográfica del medidor; *pago atrasado* los días acumulados de demora en el pago de facturas; e *historial de fraudes* el número de irregularidades anteriores detectadas, entre otras características [63]. Un subconjunto de las características adicionales se ilustra en la figura 3.2. Para preservar el anonimato de los clientes, truncamos la información de latitud y longitud con una precisión de 1 km.

3.2.2. Extracción de características

Supongamos que estudiamos un conjunto de n clientes, de los cuales sabemos: $C = (c_1, ..., c_m)$ el consumo mensual medido para m meses consecutivos. En el presente trabajo, establecimos $m = 36$ y definimos el último consumo C_m como el anterior a la inspección. Además del consumo mensual, las características complementarias son accesibles, como se describió anteriormente, las denotamos como $v = (v_1, ..., v_p)$. La etiqueta (resultado real de la inspección) se representa como $y = 1$ o $y = 0$ para la clase positiva (irregular) y negativa, respectivamente.

Uno de los primeros enfoques de NTL consistió en crear a mano un conjunto de características de la curva de consumo [24, 50]. Este paso consiste en definir una representación $(u_1, ..., u_k) = f(c_1, ..., c_m)$, mapeando el vector de entrada de consumo mensual en k características $\{u_i\}$. Ejemplos evaluados en este trabajo son (i) la media de consumo y la desviación estándar, (ii) los ratios estacionales, definidos como la razón de consumo entre la temporada del año pasado y la actual (trimestre actual *versus* mismo trimestre del año anterior), (iii) coeficientes de Fourier (los primeros cinco son una opción común para una señal de consumo de tres años), (iv) coeficientes de *wavelet*, (v) y los coeficientes de una aproximación polinomial.

3.2.3. Clasificación

Considerando como entrada los datos de consumo brutos $(c_1, ..., c_m)$ o características calculadas $(u_1, ..., u_k)$, el siguiente paso es mapear la entrada X en una etiqueta $\hat{y}$. Para definir la clasificación independiente de la selección de características, de ahora en adelante se denota las características de entrada como X. Varias técnicas de clasificación son populares para la detección de NTL. Evaluamos el conjunto de opciones más frecuentes: (i) regresión logística (LR) [10, 34, 48], (ii) máquinas de vectores de soporte (SVM) [10,34,52,53]) *random forest* (RF) [35,48], (iv) *gradient boosting* (GB) [84], y (v) *extreme gradient boosting* (XGB) [10,48,84].

3.2.4. Clasificación basada en DNN (*deep neural network*

Más recientemente, el notable éxito del aprendizaje profundo transformó el campo, y la mayoría de las estrategias están cambiando del paradigma de clasificación y extracción de características a un aprendizaje *end-to-end*. En este contexto, las características y las decisiones de umbrales se descubren conjuntamente de una manera basada en datos. En el presente trabajo, probamos las alternativas basadas en DNN más populares y relevantes: (i) una red neuronal convolucional (CNN), (ii) una red recurrente de memoria larga de corto plazo (LSTM) y (iii) una red neuronal completamente conectada (MLP). La arquitectura de redes se ilustra en la figura 3.1. Probamos arquitecturas entrenadas exclusivamente en la señal de consumo y redes entrenadas de dos entradas para explotar simultáneamente la señal de consumo de energía y la información complementaria.

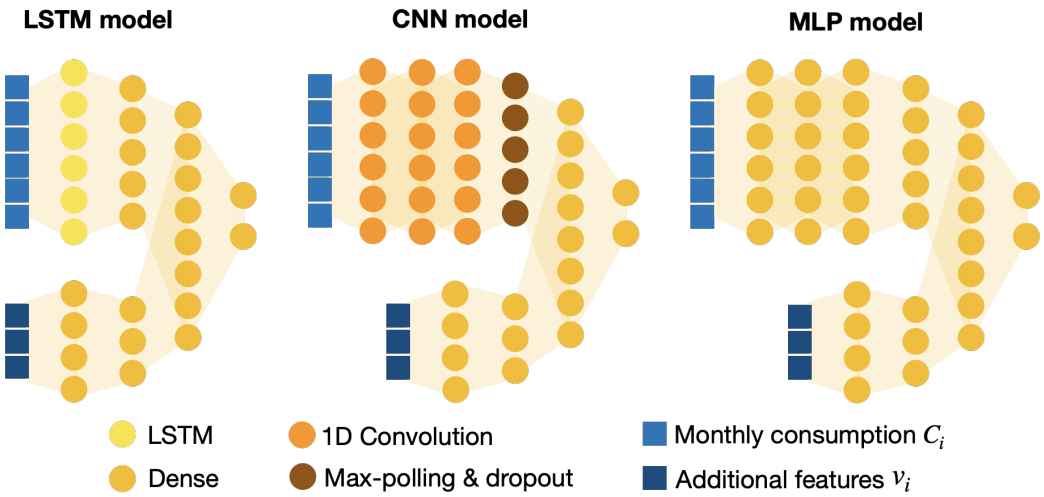


Figura 3.1: Modelos DNN probados. Probamos tres tipos de modelos: (i) una red recurrente con memoria larga de corto plazo (izquierda), (ii) una red convolucional (centro) y (iii) una red con capas completamente conectadas (derecha). Para cada modelo, comparamos el rendimiento aprendiendo exclusivamente de la señal de consumo y su combinación con las características adicionales disponibles.

3.2.5. Ataques adversarios

Uno de los aspectos más desafiantes de las soluciones basadas en DNN es comprender e interpretar lo que estos modelos están aprendiendo. En contraste con las características expertas, donde intuitivamente tenemos patrones en mente que queremos representar, las redes neuronales aprenden patrones óptimos directamente de los datos. Para comprender algunas de las características y patrones que están aprendiendo los modelos, implementamos un algoritmo de ataque adversario como se describe a continuación. Entonces, al transformar sintéticamente un consumo normal en uno fraudulento (y viceversa), podemos comprender algunos

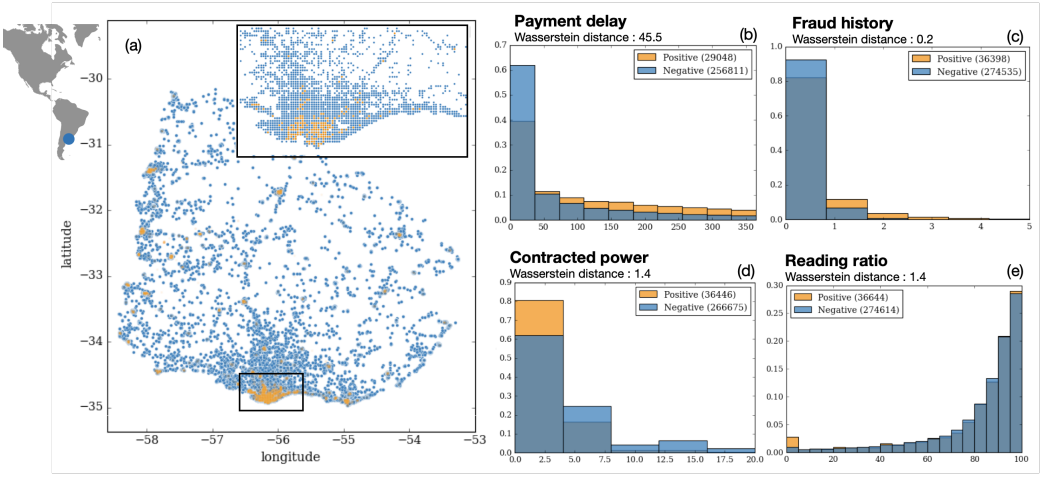


Figura 3.2: Datos completamente etiquetados. (a) La geolocalización de un subconjunto de las 311k muestras etiquetadas, en naranja/azul, se ilustra en muestras positivas / negativas. Los gráficos (b) - (e) muestran la distribución de un conjunto de características adicionales en ambas clases; para cada uno, se informa la distancia de *Wasserstein* entre las distribuciones de la clase positiva y negativa. Cuanto mayor sea la distancia, mayor será la diferencia entre las dos distribuciones, lo que significa que más relevante es la función para detectar el fraude. El número de muestras utilizadas para estimar cada distribución se proporciona en cada gráfico (ya que algunas variables no están disponibles para todos los clientes de la base). El índice de lectura se refiere a la proporción de datos obtenidos del medidor en el sitio (en algunos casos donde no se puede acceder a la lectura del medidor, el valor se estima realizando una regresión con datos históricos).

de los patrones que capturan los modelos y contrastarlos con nuestra intuición del problema.

En [38] se presenta un método de perturbación lineal para el ataque adversario. Sea $J(\theta, X, y)$ la función de pérdida (*loss*) y θ los parámetros del modelo. Dado que el modelo es diferenciable, podemos ajustar la entrada para producir un descenso/ascenso de gradiente en la salida prevista del modelo $\tilde{X} = X + \epsilon\, sign(\nabla_X J(\theta, X, y))$, ϵ representa el paso de perturbación. Se puede aprovechar una aplicación iterativa de este procedimiento para transformar las entradas hasta que la predicción del modelo cambie la categoría predicha (ver sección 3.3.4).

3.3. Resultados y discusión

3.3.1. Características adicionales

La figura 3.2 ilustra la distribución de los datos en todo Uruguay (a) y la distribución de un subconjunto de características adicionales (b)-(e). Se proporciona la distancia entre las distribuciones asociadas con la clase positiva y negativa. Como se muestra en la figura 3.2 (b), la distribución del retraso en el pago se desplaza hacia la derecha para la clase positiva, lo que sugiere que existe una correlación positiva entre un retraso en el pago de la factura y la ocurrencia de fraude. De

manera similar, observamos que el historial de fraude tiende a ser una indicación de mayores probabilidades de fraude (figura 3.2 (c)). En el caso de la potencia contratada (figura 3.2 (d)), se observa un mayor porcentaje de fraude en el conjunto de clientes con menor potencia contratada. Otra observación interesante es el número comparativamente grande de muestras fraudulentas para las que la función de índice de lectura es 0 (figura 3.2 (e)).

Usando *extreme gradient busting* (XGB) como modelo de clasificación, comparamos el desempeño de detección de fraude cuando solo se considera la curva de consumo y al incluir características adicionales como se describe anteriormente (comparamos el algoritmo de clasificación en los siguientes experimentos). La figura 3.3 muestra las curvas de *precision-recall* cuando XGB se entrena exclusivamente con la curva de consumo (línea punteada), cuando incluimos la información adicional del contrato, como el retraso en el pago (línea discontinua) y, finalmente, cuando utilizamos toda la información, incluida la geolocalización (línea continua). Este resultado valida resultados similares observados en el pasado [35, 67]. En este experimento, solo consideramos a aquellos clientes para los que se disponía de información de consumo de tres años antes de la inspección, de las 311k muestras, 168k verificaron esta condición.

3.3.2. ¿Características hechas a mano o datos sin procesar?

La tabla 3.1 proporciona los resultados obtenidos para un conjunto de estrategias de clasificación, entrenadas para predecir a partir de la curva de consumo sin procesar o características hechas a mano (en este caso concatenamos las características descritas en 3.2). Podemos ver que para todos los algoritmos utilizados, los resultados son superiores cuando se utilizan directamente los datos sin procesar. Concluimos, entonces, que extraer características expertas no es la mejor manera de trabajar con datos en NTL. Dado que en experimentos anteriores no observamos ninguna ventaja evidente de las funciones de extracción de características, en el resto de esta sección, entrenamos y probamos nuestros modelos usando como entrada la curva de consumo sin procesar.

3.3.3. Tamaño del conjunto de entrenamiento

Dado que tuvimos la oportunidad única de entrenar con un conjunto de datos muy grande de más de $150k$ muestras, evaluamos cómo el tamaño del conjunto de entrenamiento afecta el rendimiento. Observamos que por encima de las $80k$ muestras la ganancia en el desempeño se vuelve marginal (ver figura 3.3).

3.3.4. ¿Qué están aprendiendo los modelos basados en DNN?

Entrenamos modelos DNN usando la curva de consumo y combinándolo con las características adicionales descritas en 3.2. En la figura 3.1 se proporciona una descripción esquemática de los modelos. Las tres arquitecturas evaluadas (un

Algoritmo	datos	AUC _PR	AUC _ROC	Fmeasure	MCC
LR	Raw	25,5	64,7	32,0	0,188
	Features	19,5	61,9	29,6	0,146
SVM	Raw	25,4	64,5	31,9	0,184
	Features	19,4	61,7	29,6	0,148
RF	Raw	**27,3**	67,2	33,6	0,212
	Features	26,5	66,0	33,3	0,212
GB	Raw	27,2	**67,5**	**33,9**	**0,234**
	Features	27,1	67,1	33,7	0,232
XGB	Raw	26,9	66,5	33,5	0,220
	Features	26,5	65,7	32,5	0,204

Tabla 3.1: Evaluación de desempeño de distintas técnicas de clasificación y extracción de características. "*Features*" es un conjunto de 30 características expertas concatenadas que se detallan en Métodos, mientras que "*Raw*" son los 36 datos de consumo mensual normalizados.

Algorithm	PR _AUC	ROC _AUC	Fm	MCC	Precision	Recall
MLP *	25,8	65,5	32,0	0,202	29,5	35,1
CNN *	**28,6**	**67,9**	34,1	0,232	33,0	**35,3**
LSTM *	28,2	66,3	**34,2**	**0,241**	**35,5**	32,3
LR	34,1	71,3	37,9	0,268	33,5	43,6
SVM	34,2	71,4	38,2	0,284	38,6	37,9
RF	38,8	75,2	41,3	0,310	36, 1	**48,3**
GB	40,3	75,9	**42,3**	**0,328**	**41,0**	43,6
XGB	**40,5**	**76,2**	42,0	0,318	37,3	48,1
MLP	37,0	73,2	39,8	0,299	38,7	40,9
CNN	38,2	73,8	40,0	0,299	37,9	42,4
LSTM	38,4	73,6	40,4	0,301	37,3	44,2

Tabla 3.2: Evaluación de todos los algoritmos presentados entrenados con el historial de consumo y los datos adicionales. Los primeros 3 algoritmos (*) se entrenaron utilizando solo la serie de tiempo de consumo, complementando los resultados de la Tabla 3.1.

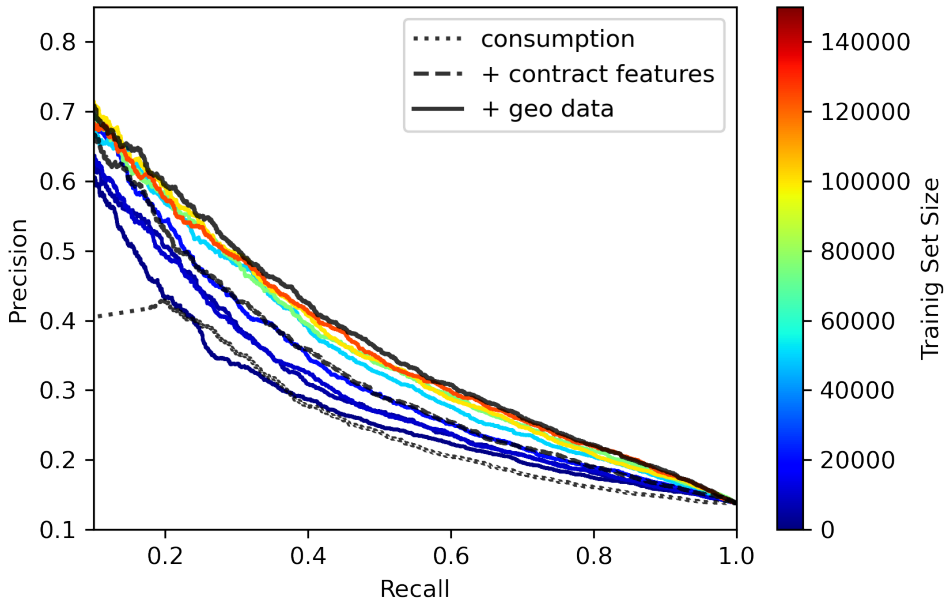


Figura 3.3: Curva de *precision-recall* cuando se combinan características contractuales adicionales y datos de consumo, y cuando se modifica el tamaño del conjunto de entrenamiento. El entrenamiento con diferentes conjuntos de características se ilustra con diferentes estilos de línea, la curva punteada fina representa el desempeño de un modelo entrenado solo en la curva de consumo, la curva punteada gruesa representa un modelo que incluye información adicional del contrato del cliente (ver 3.2) y, finalmente, la línea continua representa la precisión cuando se considera toda la información (incluida la geolocalización). Además, las líneas continuas de diferentes colores representan el desempeño a medida que cambiamos el tamaño del conjunto de entrenamiento. Comenzamos con 150k muestras de entrenamiento (rojo) y disminuimos el tamaño de la configuración de entrenamiento hasta 10k muestras (azul). Este experimento se realizó considerando XGB como algoritmo de clasificación.

modelo recurrente, uno convolucional y uno completamente conectado) se comportaron de manera similar. Superan los métodos clásicos como SVM, pero presentan resultados competitivos en comparación con alternativas modernas como XGB. El uso de DNN puede ser una mejor opción al trabajar con datos de mayor dimensión como es el caso de los medidores inteligentes y que veremos en el capítulo 6.

Para comprender algunos de los patrones que capturan los modelos basados en DNN, utilizamos ideas de *adversarial attacks* para modificar los datos originales paulatinamente hasta que se modifique la predicción de la red (ver 3.2). En la figura 3.4 se ilustran ejemplos de muestras fraudulentas transformadas en típicas (a)-(b) y viceversa (c)-(d). Se observa cómo transformar un perfil normal en uno fraudulento implica una caída en el consumo (figura 3.4 (g)) y una variabilidad atípica (figura 3.4 (h)). Por otro lado, para convertir un ejemplo fraudulento en uno normal, se introduce una variabilidad estacional suave (figura 3.4 (e)), y en lugar de disminuir, el consumo oscila (figura 3.4 (f)).

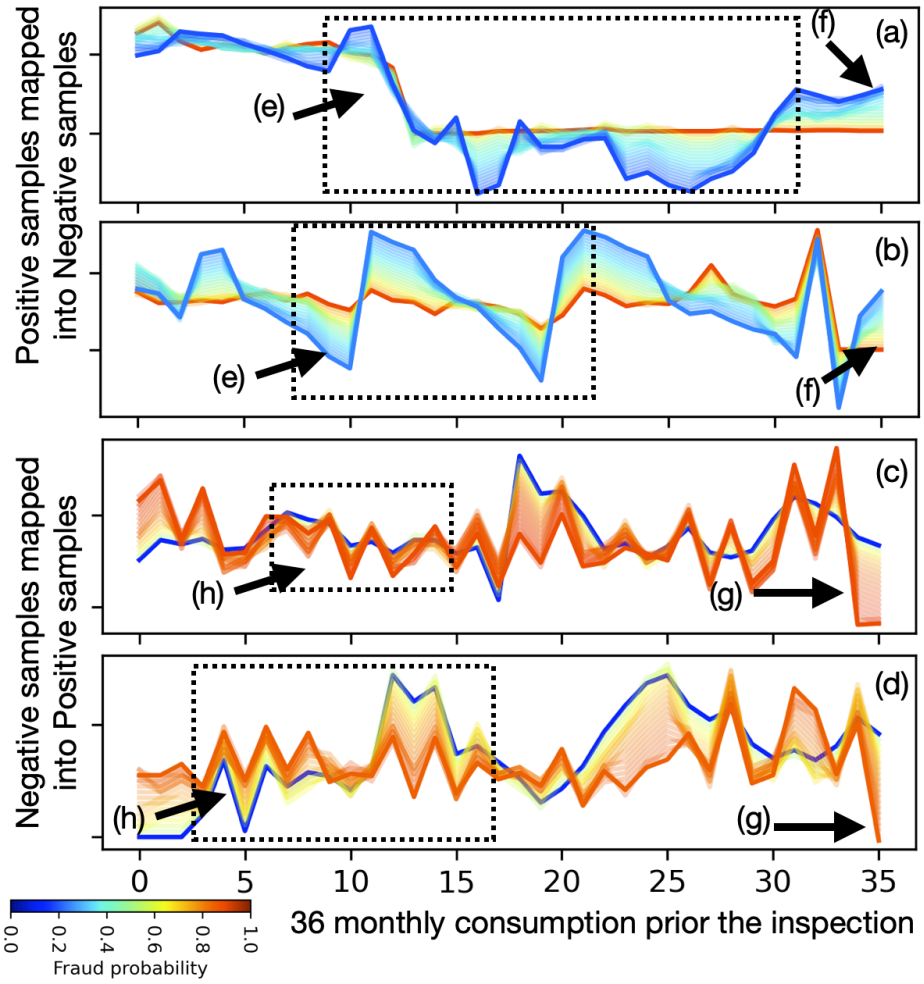


Figura 3.4: Transformar un perfil normal en uno irregular y viceversa. Los ejemplos (a)-(b) representan curvas inicialmente fraudulentas (curva roja en el fondo). A medida que avanza el ataque adversario (ver 3.2), podemos observar cómo evoluciona la curva y el color se vuelve más cercano al azul (lo que significa que la predicción cambia hacia la clase normal). Cada curva se colorea utilizando la salida del modelo XGB. Por el contrario, (c)-(d) muestran perfiles inicialmente normales (curvas azules en la parte posterior), transformados iterativamente en perfiles fraudulentos (curvas rojas en el frente).

3.4. Comparación con trabajos relacionados

En la revisión de trabajos de Messinis et al. [69], las bases de datos se clasifican según su tamaño, definidas como grandes aquellas con más de 1.000 registros. Esta tesis complementa el trabajo de ellos comparando un conjunto de soluciones relevantes en una nueva base de datos completamente etiquetada de más de $300k$ clientes en América del Sur. A diferencia del trabajo anterior, no asumimos que los clientes sean normales por defecto [48], y solo consideramos las etiquetas verdaderas resultado de una inspección minuciosa en el sitio. También comparamos enfoques novedosos basados en DNN con los clásicos, validando en una nueva y extensa base de datos algunos hallazgos previos [48, 53]. Por ejemplo, validamos que las características categóricas son importantes y descriptivas, y muestran una distancia de *Wasserstein* significativa entre la distribución entre clases. También validamos cómo la localización geográfica es útil y complementaria [34, 35, 53, 67]. Hay escasos trabajos que presenten resultados en un conjunto de datos de esta magnitud y el presente trabajo complementa sus hallazgos [10, 34, 48]. Los avances actuales en DNN están proporcionando un poder notable para extraer características de una manera basada en datos, y nuestros resultados sugieren que las características anteriores hechas a mano [24] ya no brindan ventajas claras. Nuestros hallazgos únicos son: (i) por encima de los $80k$ clientes etiquetados en una base de entrenamiento, el aumento de desempeño de los modelos es marginal (esto tiene enormes implicaciones prácticas cuando una empresa planifica y diseña los esfuerzos de recopilación de datos); (ii) comparar en el mismo conjunto de datos enfoques clásicos y alternativas basadas en DNN; y (iii) utilizar ideas de *adversarial attacks* como una forma de entender qué características de las curvas de consumos aprenden las redes para clasificar a los clientes con comportamientos irregulares.

3.5. Conclusiones

Presentamos y analizamos un nuevo conjunto de datos completamente etiquetado de más de 311k clientes de todo Uruguay. Comparamos los enfoques clásicos con extracción de características expertas con los métodos modernos basados en redes neuronales profundas. Observamos que este último podría identificar patrones generales mientras optimiza los umbrales de decisión de una manera unificada basada en datos. Se verificó un rendimiento similar en diferentes arquitecturas de redes (recurrente, convolucional y completamente conectado). Aprovechamos las ideas de ataques adversarios para explorar algunos de los patrones más importantes capturados por las redes. Además, observamos que después de aproximadamente $80k$ clientes etiquetados, la ganancia de desempeño de los algoritmos es marginal. Esta conclusión probablemente esté relacionada con el hecho de que estamos trabajando con curvas de consumo mensual, que son de baja dimensión en comparación con los datos recopilados por los medidores inteligentes. Dado que la mayor parte de la infraestructura está cambiando a medidores inteligentes (de hecho en 2021 en Uruguay más del 40 % de los medidores son inteligentes), el trabajo futuro incluye reevaluar las estrategias resumidas en el presente capítulo en este nuevo escena-

rio futuro. La coexistencia de tecnologías de medición es un desafío para explorar nuevos enfoques con algoritmos de resolución múltiple. Este punto se aborda en el capítulo 6 de esta tesis.

Capítulo 4

Enfoque de máximo retorno económico

4.1. Introducción

Si bien en la bibliografía hay numerosas propuestas para clasificar clientes como sospechosos de fraude, al momento de escribir este trabajo no había propuestas para definir el número de inspecciones a realizar teniendo en cuenta los costos de dichas tareas y la variabilidad en el retorno esperado. En este capítulo se propone el diseño de una estrategia de detección NTL que usa una medida de desempeño, que a diferencia de las métricas estándar insensibles a los costos, como *precision*, la tasa de positivos verdaderos (TPR), el AUC y el *F-measure* (ver 2.6), maximiza el rendimiento económico efectivo. Para ello, se consideran tanto los montos recuperados como el costo de inspección. Además, la estrategia propuesta se puede utilizar para dimensionar la infraestructura a cargo de la recuperación de energía. Esto permite contribuir no solo a las decisiones de corto plazo, por ejemplo, qué cliente se debe inspeccionar primero, sino también a la elaboración de estrategias a largo plazo, como pueden ser el presupuesto del área de recuperación. El problema está formulado en un marco de riesgo bayesiano. La validación experimental se presenta utilizando un gran conjunto de datos de usuarios reales de UTE. Los resultados obtenidos muestran que el método propuesto puede aumentar las ganancias y proporcionar un abordaje eficiente y realista para NTL. Además, la estrategia propuesta es general y se puede adaptar fácilmente a otros problemas prácticos.

El objetivo de este capítulo es desarrollar una solución de aprendizaje automático que, basada en los perfiles de consumo de los clientes y los costos del servicio, produzca una lista reducida, óptima, de clientes a ser inspeccionados. La optimización consiste en la maximización del retorno económico de las inspecciones. En otras palabras, nuestro objetivo es detectar a los clientes que cometen fraude, pero también proporcionar una lista de inspección que contenga aquellos clientes para los cuales el retorno económico es potencialmente mayor.

Las principales contribuciones de este capítulo son: (i) presentar un enfoque para la detección de NTL concebido para maximizar el retorno económico; (ii) proponer un método flexible que pueda optimizarse para modelos de costos realistas, para que esta solución se pueda utilizar como una herramienta de gerenciamiento;

(iii) contextualizar la solución propuesta en un marco de riesgo Bayesiano, lo que permite que investigadores de otras áreas puedan adaptar fácilmente la propuesta a sus disciplinas específicas; y (iv) estudiar y comparar nuestra solución con otros enfoques de clasificación basados en costos.

4.2. Estrategia propuesta

Supongamos que x_i representa un vector de columna con los valores de característica asociados a la muestra i-esima, e y_i su etiqueta. Por ejemplo, en el contexto de NTL, x_i puede representar el historial de consumo mensual, o la concatenación del historial de consumo y características numéricas adicionales (por ejemplo, las coordenadas geográficas del cliente asociadas a la muestra) [67]. Nos centramos en un problema de clasificación binaria donde $y_i \in \{-1, 1\}$. La etiqueta $y_i = 1$ (llamada clase positiva) se asocia a un comportamiento fraudulento, mientras que la etiqueta $y_i = -1$ (clase negativa) se asocia a un cliente normal.

Cuando las probabilidades posteriores $P(y_i = 1|\mathbf{x_i})$, y $P(y_i = -1|\mathbf{x_i})$ están disponibles para una muestra dada $\mathbf{x_i}$, el criterio de clasificación que minimiza el error de clasificación sobre un conjunto dado $X = \{\mathbf{x_1}...\mathbf{x_n}\}$ es:

$$\hat{y}_i = \underset{\tilde{y}}{\operatorname{argmax}}\{P(\tilde{y}|\mathbf{x_i})\}. \tag{4.1}$$

Donde $\hat{y}_i$ denota la etiqueta predicha, mientras que y_i representa la etiqueta real (disponible o no). La regla de clasificación anterior se conoce como regla Bayes [26]. Es fácil probar que esta estrategia conduce a una solución de clasificación óptima en términos de minimizar el error de clasificación promedio.

Aunque la estrategia anterior puede parecer atractiva, no se adapta a problemas con clases desbalanceadas. Resultados poco prácticos se obtienen cuando se aplica para NTL. En particular, debido a que el problema de NTL es muy desbalanceado (solo un pequeño porcentaje del total de clientes realiza actividades fraudulentas [24, 36, 76]). Como ya se mencionó en el capítulo 2 si solo 5 % de los clientes están cometiendo fraude, entonces, un clasificador trivial que predice siempre clase negativa alcanzaría una exactitud de 95 %, a pesar de que no proporciona ningún acierto en la detección de fraude.

Como alternativa, estamos interesados en minimizar la pérdida financiera teniendo en cuenta: (a) el costo de realizar inspecciones individuales y (b) el daño de no detectar un caso fraudulento. Esta información podría usarse para definir si el número de inspecciones realizadas es suficiente, o si la reasignación de recursos es necesaria en una división de la empresa.

4.2.1. Maximizando el retorno económico

Dejemos que m denote la cantidad de inspecciones que se realizarán y $X_m \subset X$ un subconjunto arbitrario de m muestras de X. Como antes, $P(y_i = 1|\mathbf{x_i})$ denota la probabilidad de que una muestra dada $\mathbf{x_i}$ esté cometiendo fraude, a_i representa la cantidad de dinero que el cliente i^{th} podría estar robando (si lo hace), y c_i

el costo de inspeccionar al cliente i^{th}. Dadas las definiciones anteriores, nuestro enfoque consiste en obtener el subconjunto óptimo $\hat{X}_m = \{\mathbf{x_{i_1}}, ..., \mathbf{x_{i_m}}\}$ tal que

$$\hat{X}_m = \underset{X_m}{\operatorname{argmax}} \left\{ \sum_{k=1}^{m} a_{i_k} P(y_{i_k} = 1 | x_{i_k}) - \sum_{k=1}^{m} c_{i_k} \right\}. \tag{4.2}$$

4.3. Implementando una solución para NTL

La ecuación (4.2) es intuitiva y expresa matemáticamente el objetivo de maximizar el rendimiento económico. Tres aspectos cruciales deben ser abordados antes de que sea prácticamente aplicable. Por ejemplo, necesitamos estimar: (i) la probabilidad de clase *a posteriori* $P(y_i = 1|\mathbf{x_i})$, (ii) la cantidad de fraude potencial a_i, y (iii) el número óptimo de inspecciones m_{op}. A continuación, abordamos estas definiciones, y proponemos una solución práctica para estimar estas cantidades con la información accesible en el contexto de la detección de NTL.

4.3.1. Estimación empírica de la probabilidad *a posteriori*

Un enfoque ingenuo sería explotar los datos de entrenamiento $(\mathbf{x_i}, y_i)$ para estimar directamente la función de densidad $P(y = 1|\cdot)$, por ejemplo, aplicando un *kernel* no paramétrico basado en el método [94]. Esta familia de métodos es impracticable en varias aplicaciones, específicamente cuando la dimensión del espacio de las características es moderado o grande [77]. Ciertamente, no es la elección correcta en el contexto de NTL donde la dimensión del espacio de la característica es moderadamente grande [24, 36, 106].

Inspirados por el trabajo de Zadrozny y Elkan [108], proponemos estimar la probabilidad posterior de fraude en dos pasos. Primero, se entrena un método de clasificación para estimar una función de puntuación $s(\mathbf{x})$. Luego, se define una función de calibración $g : [0,\ 1] \rightarrow [0,\ 1]$ de manera que $P(y = 1|\mathbf{x}) = g(s(\mathbf{x}))$. Los algoritmos de clasificación como SVM, RF o redes neuronales son extremadamente eficientes en el aprendizaje de la distribución de diferentes clases en el espacio de características. La mayoría de estas técnicas proporcionan como resultado una función de puntuación $s(\mathbf{x})$. Cuando $s(\mathbf{x}) \approx 0$ la muestra $\mathbf{x}$ muy probablemente pertenezca a la clase negativa, mientras que por otro lado, $s(\mathbf{x}) \approx 1$ indica que la muestra probablemente pertenece a la clase positiva. Luego, la clasificación de las nuevas muestras de entrada se puede realizar de acuerdo con: *si* $s(\mathbf{x}_i) < \lambda$ *entonces* $\hat{y}_i = -1$*, de lo contrario,* $\hat{y}_i = 1$. La mayoría de los métodos establecen por defecto $\lambda = 0{,}5$, pero también es común ajustar este umbral para maximizar una medida de desempeño particular (por ejemplo, la precisión, $F_{measure}$ o AUC).

Aunque la puntuación $s(x)$ proporciona información valiosa para la clasificación, no puede interpretarse directamente como la probabilidad de pertenecer a una clase determinada. Por ejemplo, $s(\mathbf{x}) = 0{,}2$ no implica necesariamente que $P(y = 1|\mathbf{x}) = 0{,}2$.

La calibración del *score* se puede definir como la tarea de obtener una función de calibración $g : [0,\ 1] \rightarrow [0,\ 1]$ tal que $P(y = 1|g(s(\mathbf{x})) = s)$ converge a s si el

número de muestras tiende a infinito [108]. Una de las técnicas más utilizadas en la calibración de probabilidad es *Platt scaling*, propuesto originalmente por John Platt para calibrar SVM [82]. Este es un método paramétrico basado en el ajuste de la probabilidad a través de la regresión logística, como se describe en la ecuación (4.3). Los parámetros A y B se aprenden de manera supervisada utilizando los datos de entrenamiento disponibles.

$$P(y = 1|x) = \frac{1}{1 + e^{As(x)+B}}. \tag{4.3}$$

Recientemente se demostró que una extensión simplificada llamada *temperature-scalling*, es el método más simple y eficiente para calibrar la salida de las redes neuronales [42].

En el presente trabajo, el método de calibración de *Platt scalling* se compara con un enfoque no paramétrico basado en regresión isotónica. Luego, la calibración se realiza utilizando la implementación *Platt scalling* incluida en la biblioteca *scikit-learn* para el *score* de salida de los algoritmos *random forest* y SVM. Para NN implementamos *temperature-scalling* agregando una capa adicional antes de la función de activación de *SoftMax* (para minimizar la entropía cruzada binaria).

4.3.2. Estimación de la pérdida potencial de fraude

Para estimar el fraude potencial por cliente (magnitud de impacto económico) a_i, proponemos dos alternativas. La primera idea utiliza información directa del contexto de NTL, que requiere solo el conocimiento de la máxima potencia contratada de los clientes. La segunda alternativa, utiliza los registros de recuperación económica obtenidos para clientes fraudulentos inspeccionados en el pasado.

Estimación de fraude utilizando exclusivamente información de facturación

Una instalación doméstica típica incluye, además del medidor de potencia, una llave que limita la potencia máxima que se puede consumir de la red eléctrica (y protege la infraestructura eléctrica en caso de fallas).

Sobre la base de la observación empírica de los datos, asumimos lo siguiente: los clientes fraudulentos reducen sus facturas eléctricas en una cantidad que es aproximadamente proporcional a su consumo real, es decir, $a_i \propto e_i.$, donde e_i representa la cantidad total de energía (en kWh) que el cliente realmente consume. Observamos empíricamente que la energía promedio consumida es aproximadamente proporcional a la potencia máxima contratada Mp_i. Esta es una observación interesante desde una perspectiva práctica, ya que la última es información disponible (mientras que la primera es estrictamente desconocida en el contexto de la detección de fraudes). Por lo tanto, la magnitud potencial del fraude se puede estimar en $a_i \propto Mp_i$.

Estimación de fraude usando los registros del retorno económico

Un enfoque alternativo para la estimación del posible fraude es considerar este problema también como un problema de aprendizaje supervisado. En otras pala-

bras, si el daño económico se puede recuperar en la práctica (después de realizar las inspecciones reales), podemos usar esta información para predecir $a_i(x)$.

Lo formulamos como un problema de regresión en el que, al usar la información de los clientes x, predecimos el volumen del fraude a_i. Con este fin, recopilamos medidas reales de la pérdida económica asociada a los casos de fraude. En 2017 se realizaron alrededor de 50k inspecciones y se analizó la información asociada a 3k casos de fraude.

RF, SVM y una red neuronal fueron los algoritmos utilizados para la solución numérica del problema de regresión. Estos algoritmos fueron entrenados en casos fraudulentos utilizando el daño económico real como el resultado objetivo. El valor utilizado en la práctica fue el valor de las multas impuestas por UTE a los clientes. Este valor es una recuperación económica real y es calculado por la empresa de forma de representar el daño causado por el cliente.

4.4. Ingresos y costos: definición de la capacidad operativa óptima

El número óptimo de inspecciones que se deben realizar se determina por la cantidad esperada de ganancia económica en comparación con el costo asociado para realizar las inspecciones. Este equilibrio entre ganancia y costo se puede formular en el marco descrito anteriormente,

$$\hat{X}_m = \underset{N, X_m}{\operatorname{argmax}} \left\{ \sum_{k=1}^{m} a_{i_k} P(y_{i_k} = 1 | x_{i_k}) - \sum_{k=1}^{m} c[N, k] \right\}, \tag{4.4}$$

$c[N, k]$ indica el costo de realizar la inspección k^{th} cuando la infraestructura está diseñada para realizar un número nominal de inspecciones N. A continuación vemos cómo se puede definir el costo de inspecciones $c[N, k]$.

4.4.1. Modelo de costo

La definición del número de inspecciones que deben realizarse ayuda a establecer y diseñar la infraestructura operativa. Un modelo de costo realista debe incluir al menos un componente fijo y uno variable. Recordemos las definiciones estándar en microeconomía. El costo marginal (MC) representa la derivada de la función de costo con respecto a la cantidad, y el costo promedio (AC) el costo total dividido por el número de unidades. En el presente trabajo utilizamos una curva de costos diseñada para una capacidad operativa de N inspecciones, asumimos que el costo fijo es proporcional a N y un costo adicional cuando se opera sobre la capacidad diseñada:

$$c[m] = \begin{cases} \alpha m + \alpha N \frac{\gamma}{1-\gamma} & m \leq N \\ \beta m + \alpha N \frac{1}{1-\gamma} & m > N. \end{cases} \tag{4.5}$$

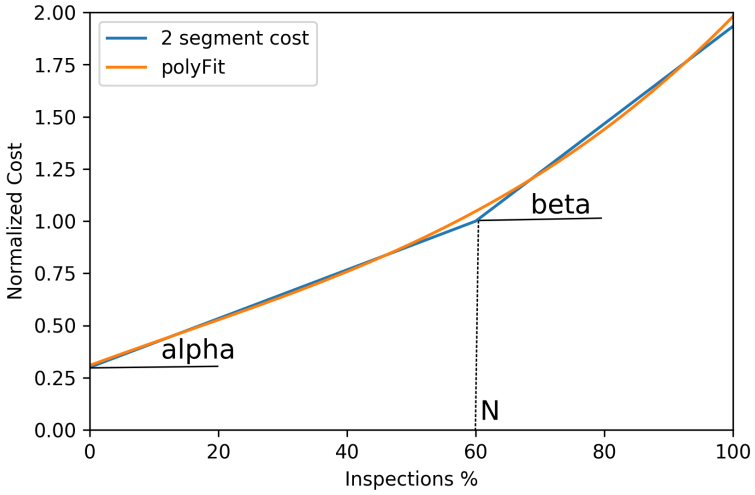


Figura 4.1: Costos de inspección suponiendo que la infraestructura está planificada para un número nominal de inspecciones de N. El modelo de costos incluye: costos fijos, variables y extras. También se considera una aproximación suave continua del modelo de dos segmentos.

Definiciones: $\gamma = c[0]/c[N]$ proporción de costos fijos, α y β el costo marginal por debajo y por encima de la capacidad diseñada respectivamente. Se utiliza una aproximación suavizada del modelo anterior como se ilustra en la figura 4.1 (aproximación polinomial de tercer orden).

Además de la lista óptima de usuarios a inspeccionar X_m, consideramos ahora el número nominal de inspecciones N también como un parámetro de optimización. Como se describió anteriormente, un modelo realista para los costos de inspección es variable y depende de la capacidad nominal N establecida por la infraestructura de la empresa. Los componentes de la ecuación (4.4) asociados a la ganancia potencial $a_{i_k} P(y_{i_k} = 1|\mathbf{x_{i_k}})$ son independientes de la curva de costo. Por lo tanto, podemos clasificar principalmente el retorno potencial estimado de cada cliente. Luego, se puede calcular el número óptimo de inspecciones m dado el valor nominal N. Una ventaja práctica importante del enfoque propuesto es que la tarea de aprendizaje más compleja (asociada a la estimación de $a_{i_k} P(y_{i_k} = 1|\mathbf{x_{i_k}})$) se realiza solo una vez, lo que permite probar en forma eficiente varios modelos de costos. En contraste, los algoritmos que toman en cuenta el costo de cada muestra internamente, por ejemplo, como se propone en [5], deben ser entrenados desde cero para cada modelo de costo, lo que genera rutinas de entrenamiento que consumen mucho tiempo. Cabe destacar que el enfoque propuesto es independiente de la cantidad de datos con la que se cuente y del tipo de algoritmo de clasificación utilizado. En particular, este enfoque económico se puede aplicar tanto sobre datos de medidores convencionales como sobre datos de medidores inteligentes.

4.5. Resultados experimentales

4.5.1. Base de datos

NTL_10K_S, esta base de datos está compuesta por las curvas de consumo histórico de diez mil clientes distribuidos a lo largo de todo Montevideo. Para proporcionar una primera ronda de experimentos en condiciones totalmente controladas, se simulan fraudes sintéticos sobre el consumo real de este conjunto de clientes. Por lo tanto, la recuperación económica real puede medirse con precisión sobre este conjunto. Los casos fraudulentos se simulan en el 10,0 % de las muestras distribuidas uniformemente en el tiempo. El porcentaje de energía robada también se distribuye al azar.

NLT_50K_R, el conjunto de datos consta de 50k clientes en Montevideo inspeccionados durante 2017. La porción de clientes fraudulentos es de aproximadamente 6,0 %. Este conjunto de datos incluye el consumo histórico de energía y características adicionales como la potencia de pico contratada y las coordenadas geográficas. Después de detectar clientes fraudulentos, se aplican multas y sanciones económicas (cuyo valor depende de la estimación de la empresa sobre la magnitud del fraude). Utilizamos esta información financiera para estimar los valores reales de la recuperación económica asociada a cada cliente fraudulento.

4.5.2. Detalles de implementación

Se utiliza un período de tres años de consumo eléctrico mensual como vector de características. Los experimentos sobre el conjunto de datos de *NLT_50K_R* también incluyen características adicionales como el registro de irregularidades anteriores, mora y coordenadas geográficas. Se proporciona un análisis detallado de estas características en [67] donde nos enfocamos en el impacto de considerar diferentes subconjuntos de características para NTL. Los algoritmos se entrenan utilizando el 70 % de los datos, mientras que el otro 30 % se separa para test. Esta partición se realiza aleatoriamente diez veces, se reporta el desempeño promedio en estos experimentos. Los algoritmos de clasificación y regresión se seleccionan de las bibliotecas de Python: *sikit-learn*, *Keras* y *Tensorflow*.

Entrenamiento

Los parámetros (C, γ) para SVM y $(n\_estimators, max\_features)$ para *random forest* se obtienen realizando una búsqueda del valor máximo de AUC en una grilla de puntos. La red totalmente conectada, tiene tres capas ocultas con 100 neuronas por capa, se utiliza ReLU como función de activación. La red se optimiza utilizando el descenso de gradiente estocástico (SGD) y la entropía cruzada para la función de pérdida. La capa de activación final es una unidad sigmoide cuyo valor se considera para estimar la probabilidad de fraude para cada muestra. Se utiliza *temperature-scaling* para la calibración. Para SVM y *random forest*, se usa *platt scalling* para la calibración.

Además, se presenta un experimento donde se evalúa la incidencia del método de calibración en los resultados. Los algoritmos *platt scaling* y regresión isotónica se comparan utilizando la implementación contenida en la biblioteca *scikit-learn*.

Para estimar el volumen de robo potencial a_i por medio de regresión, se consideran un SVR, un regresor *random forest* y una red neuronal. La red utilizada para la regresión tiene la misma arquitectura de la red de clasificación descrita anteriormente. El error cuadrático medio se considera como la pérdida a optimizar y el rendimiento económico como la salida. Para entrenar el regresor *random forest*, se utilizan 400 árboles con el criterio de error cuadrado mínimo. Cada árbol usa todas las características, pero las muestras se eligen al azar con reemplazo. Para SVR, se considera un *kernel* RBF ($\gamma = 0{,}01$) y la penalización por error se establece en $C = 1$.

Optimización del número de inspecciones

Una vez que se obtiene la probabilidad de fraude p_i y la estimación de la cantidad de daño económico a_i para cada cliente i, se puede calcular el número óptimo de inspecciones m dada una curva de costos, $c_N(m)$ es el costo total de realizar m inspecciones a una capacidad nominal de N.

Además, el método anterior se puede usar para estimar la capacidad óptima de la división a cargo de la detección de fraudes. Por ejemplo, se pueden probar diferentes valores N (definiendo una familia de funciones de costo c_N). Por otra parte, N también se puede elegir para maximizar la ganancia general,

$$N_{op} = \text{argmax}_N \left(\sum_{i=1}^{m(N)} G[i] - c_N[m(N)] \right), \tag{4.6}$$

donde G_i representa la ganancia potencial asociada a cada cliente individual (en orden descendente). Como mostraremos en los siguientes experimentos, la expresión dada en la ecuación (4.6) puede evaluarse empíricamente para encontrar la capacidad óptima de operación y el punto de operación asociado a ella.

4.5.3. Resultados en la base *NTL_10K_S*

Se comparan tres estrategias de detección de fraudes. Como línea de base consideramos una solución que maximiza la medida $F_{measure}$. Este criterio tiene como objetivo encontrar un equilibrio óptimo entre la clasificación de *recall* y *precision*, para problemas desbalanceados (ver capítulo 2). Esta es una de las muchas medidas adecuadas que pueden considerarse [69]. Se proporciona una segunda solución que estima el fraude potencial asociado a cada cliente a_i a partir de su máxima potencia contratada M_p. La tercera solución se obtiene estimando la ganancia potencial de los clientes a_i como un problema de regresión supervisada como se describe en 4.3.2.

Utilizaremos la siguiente notación para las soluciones descritas anteriormente como:

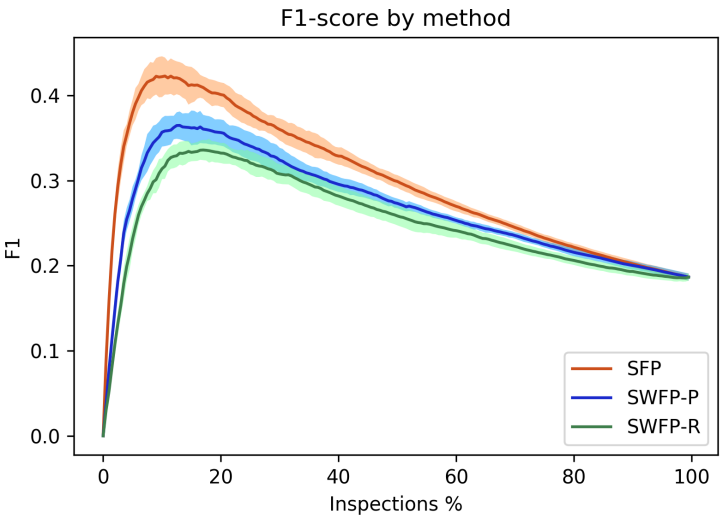


Figura 4.2: $F_{measure}$ para las propuestas SFP, SWFP-P y SWFP-R. Se usa *random forest* como algoritmo de clasificación y regresión.

- SFP (*sort fraud probability*): se genera una lista ordenada utilizando las probabilidades de fraude. El umbral de clasificación se define para maximizar la métrica de $F_{measure}$.
- SWFP-P (*sort weighted fraud probabilities*): la lista ordenada de clientes se genera ponderando la probabilidad con la potencia máxima contratada.
- SWFP-R (*sort weighted fraud probabilities using regression algorithms*): la lista ordenada de clientes se genera ponderando la probabilidad con algoritmos de regresión.

La figura 4.2 presenta $F_{measure}$ para las soluciones descritas anteriormente. El eje horizontal representa el número m de clientes a inspeccionar (es decir, etiquetados como fraudulentos). Como se esperaba, la solución SFP obtiene el mejor desempeño en términos de la medida $F_{measure}$. Por otro lado, la figura 4.3 muestra el ingreso acumulado (es decir, la ganancia económica sin tener en cuenta los costos de inspección). Y finalmente, la figura 4.4 muestra la ganancia neta (es decir, la ganancia menos el costo) con respecto al número de inspecciones m para un perfil de costo fijo.

Las figuras 4.2, 4.3 y 4.4 muestran los resultados obtenidos utilizando *random forest* como herramienta de clasificación y regresión. La tabla 4.1 muestra los resultados de las tres soluciones propuestas (SFP, SWSP-P y SWSP-R), comparando dos métodos de calibración de probabilidad. El experimento indica que *Platt scalling* genera un mejor resultado. Los resultados complementarios para los algoritmos SVM y NN se informan en la tabla 4.2.

En [4] se proponen algoritmos destinados a maximizar el rendimiento económico en el contexto del fraude con tarjetas de crédito. Algunos de estos métodos se implementan y están disponibles públicamente en la biblioteca CostCla, cost sensitive classificaction library [12]. Comparamos nuestras soluciones con dos métodos

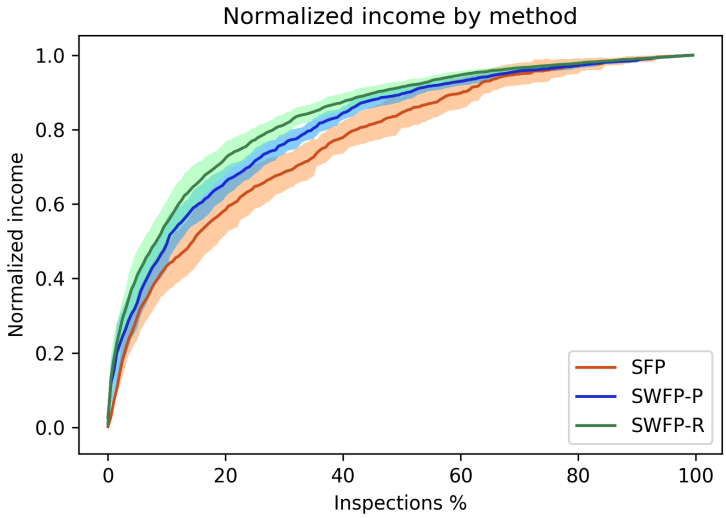


Figura 4.3: Retorno para las propuestas SFP, SWFP-P y SWFP-R.

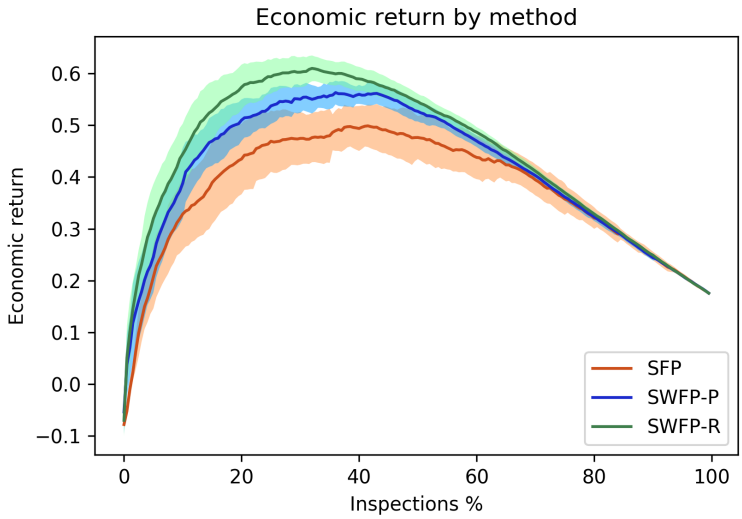


Figura 4.4: Retorno económico neto (normalizado) para SFP, SWFP-P y SWFP-R.

Tabla 4.1: Comparación de métodos de calibración *Platt scaling* y regresión isotónica para las tres soluciones propuestas mediante el uso del algoritmo de RF sobre el conjunto de datos *NTL_10K_S*.

Calibración	method	E. Return %	$F_{measure}$ %	precision %	recall %
Isotónica	SFP	31,5	42,9	**44,5**	41,3
	SWFP-P	55,0	33,5	22,3	**67,6**
	SWFP-R	58,2	31,5	21,0	62,7
Platt scalling	SFP	32,3	**43,6**	41,1	46,5
	SWFP-P	55,7	32,8	21,9	65,4
	SWFP-R	**61,0**	32,7	22,7	58,4

Tabla 4.2: Rentabilidad económica neta para las soluciones SFP, SWFP-P y SWFP-R implementadas con diferentes algoritmos de clasificación/regresión. Las soluciones que proponemos también se comparan con las soluciones basadas en costos propuestas por Bahnsen et al. [4, 5] CostCla.CSRP y CostCla.CSDT.

Alorithm	Method	E. Return %	Inspections %	$F_{measure}$
SVM	SFP	54,6 %	44,5 %	27,2 %
RF	SFP	33,4 %	10,5 %	**42,3** %
NN	SFP	41,4 %	27,5 %	30,5 %
SVM	SWFP-P	50,6 %	28,5 %	21,2 %
RF	SWFP-P	56,3 %	36,0 %	30,6 %
NN	SWFP-P	51,1 %	34,0 %	19,9 %
SVM	SWFP-R	55,6 %	40,5 %	24,3 %
RF	SWFP-R	**61,0 %**	32,0 %	30,6 %
NN	SWFP-R	55,7 %	29,5 %	21,7 %
CostCla.CSRP		49,2 %	58,5 %	25,2 %
CostCla.CSDT		43,2 %	59,5 %	21,6 %

del estado del arte en clasificación sensible a costos incluidos en CostCla: el árbol de decisión sensible al costo (CSDT) y los parches aleatorios sensibles al costo (CSRP), ver tabla 4.2. Además, medimos y comparamos la eficacia de estas soluciones en términos de tiempo de ejecución. Dado que los algoritmos de *CostCla* requieren el conocimiento previo del costo para cada muestra individual, estos algoritmos deben volver a entrenarse cada vez que se actualiza el modelo de costos (lo que se vuelve muy ineficiente cuando se desea probar una gran cantidad de modelos de costos). Por ejemplo, para obtener los resultados informados en la tabla 4.2, SWFP-R exigió aproximadamente 9 segundos, mientras que CSDT requirió 4,2 horas.

Los experimentos presentados anteriormente se obtienen para un modelo de costo fijo (N fijo). Como se explicó en la sección anterior, las curvas de costo $c_N(m)$ con capacidad nominal N pueden compararse simultáneamente. La figura 4.5 ilustra la ganancia económica neta para diferentes valores de N (capacidad operativa) y m (número real de inspecciones realizadas). El máximo global recupera 68,6 % del valor monetario total que se está robando. Esta solución se obtiene cuando se inspeccionan 33,5 % de los clientes.

Al comparar experimentos con un valor fijo respecto a un valor variable de N, se abordan dos situaciones prácticas diferentes. En el primer caso (N es fijo) se asume que la empresa tiene una infraestructura fija (por ejemplo, un número determinado de inspectores y vehículos) y desea saber qué clientes deben ser inspeccionados para maximizar el rendimiento económico. Un segundo escenario es cuando la empresa quiere determinar cuál debería ser su infraestructura óptima, en este caso, N es un parámetro que también se puede optimizar.

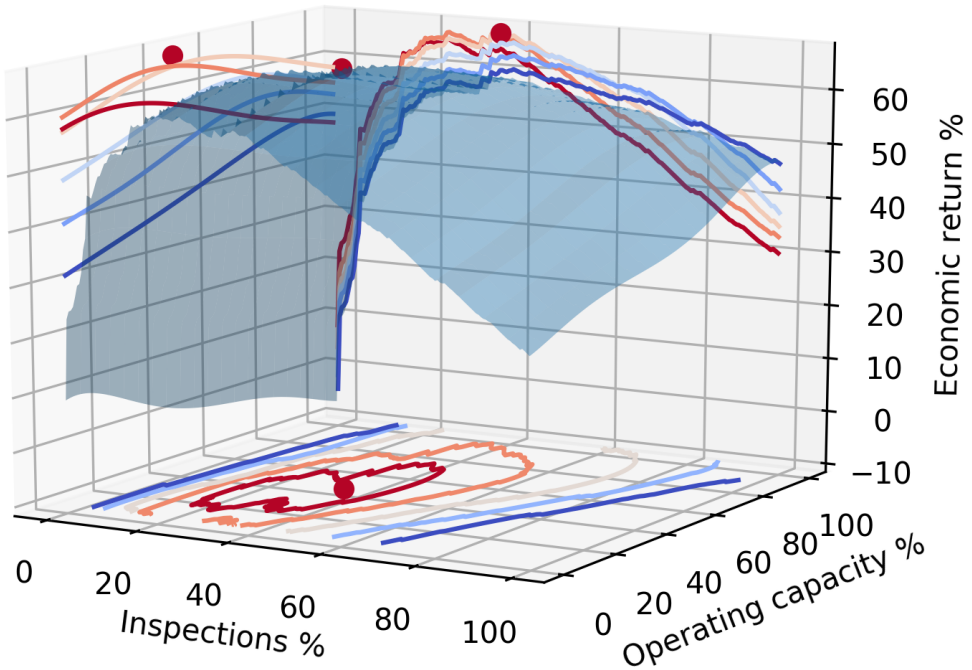


Figura 4.5: Retorno económico en función del número de inspecciones y el tamaño de la capacidad operativa. Resultados sobre el conjunto de datos de *NTL_10K_S* usando el método RF / SWFP-R. El punto rojo representa el rendimiento económico máximo y su proyección se muestra en los tres planos.

4.5.4. Resultados en la base *NTL_50K_R*

Se realizaron experimentos similares en el conjunto de datos NTL_50K_R. Este conjunto es extremadamente heterogéneo y, en particular, el número de clientes en todo el rango de potencia contratada M_p es muy desbalanceado. Para evitar el sesgo de clasificación para ciertos valores de la potencia contratada, se consideran las técnicas de sobremuestreo (se aplican exclusivamente a los datos seleccionados para el entrenamiento). Una vez que los datos de entrenamiento están balanceados, los experimentos se realizan de la misma forma que para el conjunto de datos previos.

La tabla 4.3 reporta el rendimiento económico más alto obtenido para cada solución y proporciona medidas de desempeño adicionales (el modelo de costo es fijo en este experimento). Nuevamente, se puede ver que el método SFP logra el valor más alto de $F_{measure}$. La figura 4.6 muestra el retorno económico neto para las soluciones SFP, SWFP-P y SWFP-R.

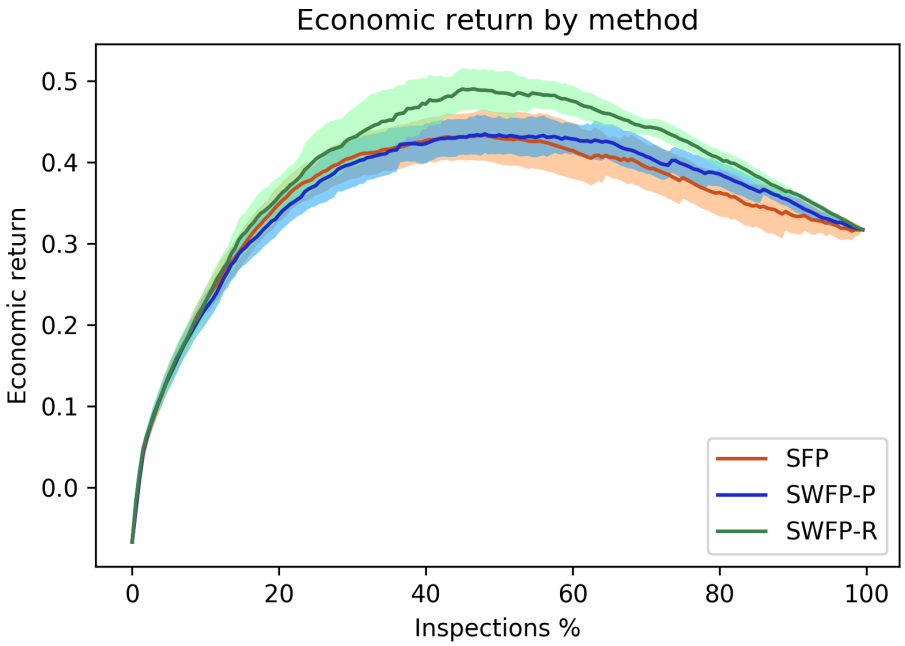


Figura 4.6: Retorno económico neto normalizado frente al número de inspecciones (modelo de costo fijo). Se utiliza *random forest* para la clasificación/regresión.

	SFP	SWFP-P	SWFP-R
Economic Return %	25,8	43,5	**49,0**
Inspections %	12,0	48,0	46,5
$F_{measure}$ %	**37,1**	24,9	26,1
precision %	**32,8**	14,8	15,6
recall %	42,7	77,3	**78,8**

Tabla 4.3: Máximo retorno económico alcanzado por SFP, SWFP-P y SWFP-R. Se utiliza *random forest* para clasificación y regresión.

4.6. Discusión y conclusiones del enfoque de máximo retorno económico

Los resultados experimentales muestran cómo el retorno económico puede mejorarse drásticamente cuando las soluciones de aprendizaje automático se desarrollan teniendo en cuenta los aspectos económicos.

RF, SVM y las redes neuronales demostraron ser algoritmos adecuados para implementar los esquemas propuestos. Aunque todos ellos son extremadamente eficientes y adecuados para esta tarea, RF fue el algoritmo que proporcionó de manera consistente el mayor desempeño para los datos disponibles. Como se esperaba, cuando se seleccionan las medidas de clasificación (como la $F_{measure}$) como criterios de maximización, los algoritmos se vuelven óptimos solo con respecto a

esa medida. Esto puede llevar a pérdidas económicas sustanciales cuando se está desarrollando una solución de detección de fraude. Es importante destacar que, debido a la gran cantidad de usuarios, incluso un aumento porcentual moderado del desempeño económico relativo representa una gran cantidad de ganancias.

Observamos que las soluciones que maximizan el impacto económico desarrolladas en el contexto de la detección de fraudes con tarjetas de crédito superan los enfoques clásicos (compare, por ejemplo, los resultados reportados en la tabla 4.2 para SFP con CSRP y CSDR). Sin embargo, en el contexto de NTL y para los datos recopilados a través de los clientes uruguayos, observamos que las estrategias propuestas (tanto SWFP-R como SWFP-P) superan a otros métodos del estado del arte. Además, las soluciones propuestas son computacionalmente más convenientes que las estrategias CSRP y CSDR. Varias curvas de costo realistas y el tamaño de la infraestructura requerida pueden optimizarse adicionalmente como se ilustra en la figura 4.5. Esto es extremadamente útil ya que, además de detectar el fraude, el enfoque propuesto puede utilizarse para simular el impacto económico asociado a diferentes decisiones de gestión. Además, el enfoque propuesto se centra en el diseño de un criterio de optimización que es económicamente significativo, y es independiente del conjunto de características o del método de clasificación disponible.

Capítulo 5

Optimización de rutas de inspección

5.1. Introducción

En el capítulo anterior vimos un nuevo enfoque del problema de NTL que muestra lo relevante de definir cuál es la métrica que se quiere maximizar a la hora de crear los planes de inspección. En particular, se puede estimar el retorno económico esperado de una inspección y realizar un plan de inspecciones que, teniendo en cuenta los costos, maximiza el resultado económico definiendo un punto de operación.

En este capítulo retomamos este enfoque pero ahora el modelo de costos no es genérico (como en el capítulo 4), sino que se propone un modelo de costos más realista, donde la definición de rutas de inspección juega un papel fundamental. Las inspecciones deben ser realizadas por cuadrillas de técnicos que recorren el territorio en un vehículo. Estas actividades tienen costos y restricciones que son abordadas en este capítulo dentro de un problema de optimización.

El problema de minimizar el costo de una ruta en un grafo (dado un conjunto de nodos) es conocido como el problema del viajante (TSP - *travel salesman problem*). Es un problema clásico de optimización combinatoria del tipo NP-completo. Si bien en este trabajo buscamos minimizar los costos de la ruta, el problema difiere en un punto muy importante: los clientes a inspeccionar no están definidos. Determinar la lista de nodos de la ruta es parte del problema de optimización. Por otro lado, existe una limitante temporal que impide realizar una única ruta para solucionar el problema (turnos de trabajo). Múltiples rutas son necesarias modelando el trabajo en equipo o una secuencia de viajes.

Las preguntas que nos hacemos son, dado un conjunto de datos de consumos eléctricos de clientes residenciales y comerciales: i) ¿cuál es el método de planificación de inspecciones que genera el mayor retorno económico para la empresa?, ii)¿cuántas inspecciones es necesario realizar?, iii) ¿cuáles son las rutas de inspección necesarias para obtener el mejor resultado?

5.2. Trabajos relacionados del área de optimización

Desde hace varias décadas existe una actividad académica muy importante en el área de optimización de rutas con múltiples variantes y variadas aplicaciones en el área de ruteo de vehículos (VRP - *vehicle routing problem*) [97].

El problema de determinar una ruta sobre un subconjunto de puntos optimizando un *score* es conocido como *orienteering problem* (OP) [40]. Este es una combinación de selección de nodos y de encontrar la ruta más corta entre los nodos. El objetivo es maximizar el *score* total de los nodos visitados. Este problema podría verse también como la combinación de dos problemas clásicos de optimización combinatoria: el problema del viajante (TSP) y el problema de la mochila (*knapsack problem*) [99]. El primero consiste en, dado un conjunto de ciudades, visitarlas todas realizando la menor ruta posible. Mientras tanto *knapsack problem* consiste en, dado un conjunto de elementos con diferentes valores y tamaños, guardar en la mochila un conjunto que maximice el valor total. En la literatura este problema también ha sido abordado como *TSP with profits* [28] agregando restricciones a TSP.

Sin embargo, dada la restricción temporal del turno de trabajo para la realización de una ruta, la solución que maximiza el retorno probablemente sea realizando varias rutas. Algunos abordajes a problemas similares han sido denominados *team orienteering problem* (T-OP) [61, 83]. En [61] se utiliza una heurística basada en *simulated annealing*, donde el compromiso entre exploración y explotación de las posibles soluciones es controlado con un parámetro ("temperatura") que va decreciendo con las iteraciones.

Si tomamos el problema simplificado (TSP): dado un conjunto reducido de puntos, supongamos 20, se quiere hallar la ruta que, pasando solo una vez por los 20 puntos, minimiza la distancia recorrida. Las opciones de este problema son permutaciones de 20, o sea, 20! (20 factorial) del orden de $2{,}4 \times 10^{18}$. Existen varias soluciones a este problema que aseguran mínimos locales. Un método muy eficiente desde el punto de vista computacional es la adaptación de *self organized maps* pasando de configuración de red a anillo planteada por Brocki et al. [8]. Este algoritmo cuenta con una implementación para *Python* en *Github* [100], la cual es utilizada en este trabajo.

En particular en el área de NTL no se han encontrado trabajos que tengan en cuenta las rutas de inspección. Los trabajos se basan en maximizar métricas de detección de fraudes [69]. En el capítulo 4 abordamos el problema de clasificación sensible a costos por muestra y lo comparamos con propuestas utilizadas en problemas de detección de fraudes con tarjetas de crédito [63]. En dicho trabajo se plantea un método para maximizar el retorno económico teniendo en cuenta un costo de inspección que es independiente de la ruta de inspección. Sin embargo, el método propuesto permite estimar, utilizando aprendizaje supervisado, el retorno económico de un conjunto de clientes. Recientemente en [65] presentamos una comparación exhaustiva de algoritmos de detección de NTL sobre una de las bases de datos más grandes utilizadas en el área. Utilizando estos dos trabajos se genera una nueva base de datos con estimaciones del retorno económico espera-

do para cada cliente. Estas estimaciones, junto con los cálculos de costos de ruta de inspección, permiten desarrollar diferentes propuestas. Estas son evaluadas con datos de retornos económicos reales provistos por UTE para todas las inspecciones contenidas en la base de datos.

5.3. Formulación del problema

5.3.1. Formulación general

El conjunto X está formado por n vectores, donde cada uno representa a un cliente y está formado por la curva de consumos e información contractual. Sea m el número de inspecciones a realizar y $X_m \subset X$ un subconjunto de m muestras de X. $P(y_i = 1|\mathbf{x_i})$ representa la probabilidad de que $\mathbf{x_i}$ esté cometiendo un fraude, a_i representa la cantidad de dinero que se gana por detectar esa irregularidad, si existe, y c_i el costo de inspeccionar al cliente i. El conjunto óptimo sería, $\hat{X}_m = \{\mathbf{x_{i_1}}, \ ..., \ \mathbf{x_{i_m}}\}$ tal que

$$\hat{X}_m = \underset{X_m}{\operatorname{argmax}} \left\{ \sum_{k=1}^{m} a_{i_k} P(y_{i_k} = 1|x_{i_k}) - \sum_{k=1}^{m} c_{i_k} \right\}. \tag{5.1}$$

5.3.2. Estimación de retorno

En este problema, a diferencia de otros trabajos que tienen en cuenta un beneficio en cada visita, no se utiliza el beneficio real de visitar a un cliente sino una estimación de este. Esto se debe a que el retorno no se conoce *a priori*. Si el problema fuera entregar productos ya vendidos, el retorno es conocido. En este problema existe una incertidumbre sobre el retorno de las actividades planificadas.

Para estimar el retorno se utiliza el método descrito en el capítulo 4 entrenando el algoritmo *extreme gradient boosting* (XGBoost) que es el que reporta mejores resultados sobre la base de datos a utilizar (resultados del capítulo 3). XGBoost, utilizado tanto para clasificación como para regresión, es una implementación eficiente del algoritmo *gradient boosting* [30] (ver capítulo 2 por más detalles de este algoritmo).

Utilizando un conjunto de datos etiquetado de entrenamiento se aprende a estimar la probabilidad de fraude $P(y_{i_k} = 1|x_{i_k})$ a partir de datos de los clientes. Para realizar un modelo que sea capaz de estimar el retorno económico de detectar un fraude (a_{i_k}), se entrena un regresor utilizando solo datos de la clase positiva. Si bien este enfoque permite utilizar algoritmos diferentes para cada una de estas dos tareas, utilizamos también como regresor XGBoost. De aquí en más la estimación de retorno económico para el cliente i_k es $r_{i_k} = a_{i_k} P(y_{i_k} = 1|x_{i_k})$.

5.3.3. Modelo de costos y restricciones

Para representar la función de costos y las restricciones veamos algunas características de la operativa:

1. Las rutas de inspecciones son diarias. El tiempo de una ruta está limitado a la duración de un turno de trabajo.
2. Todas las rutas se realizan con vehículos que comienzan y terminan en el mismo punto (UTE).
3. Varias rutas pueden realizarse en paralelo, pero cada cliente debe ser visitado una sola vez luego de programar su visita.
4. Las rutas son realizadas por dos técnicos en un vehículo de la empresa. Esto implica un costo por tiempo y un costo por distancia.

Al plantear el problema como en la ecuación 5.1, las soluciones incluirían una única ruta con la secuencia a inspeccionar. Pero, al existir una restricción de tiempo, la solución puede incluir varias rutas. Parte de la solución del problema es establecer cuál es la cantidad de rutas e inspecciones a realizar. Sea m_k la cantidad de inspecciones a realizar en la ruta k y M la cantidad de rutas a realizar, la cantidad de inspecciones total es $N = \sum_k^M m_k$.

La restricción que genera las M rutas es el tiempo de un turno y la podemos expresar de forma paramétrica:

$$m_k\beta + \sum_{i=1}^{m_k+1} (\alpha d_{ik}) < T \;\; \forall k \in 1, 2, ..., M \tag{5.2}$$

siendo d_{ik} la distancia entre los puntos i y $i-1$ de la ruta de inspección k. Los puntos $i = 0$ e $i = m_k + 1$ corresponden al origen (el punto de salida de todas las rutas). α es un coeficiente de tiempo/distancia que representa el costo del tiempo vinculado a la trayectoria de la camioneta. Se supone una velocidad promedio constante y, como simplificación, se utiliza la distancia euclideana entre la ubicación de los clientes. β es un coeficiente que representa el tiempo promedio de una inspección (ejemplo: 10 minutos por inspección).

5.3.4. Formulación del problema de optimización

La formulación de este problema es una mezcla entre *team orientering problem* (T-OP) y *capacitated profitable tour problem* (C-PTP). Ambos problemas son tratados en el capítulo 10 del libro "Vehicle Routing: Problems, Methods, and Applications" [97].

Simplificaremos la notación cambiando el subíndice de clientes de i_k a i (ejemplo, r_{i_k} pasa a ser r_i), esta notación nos permite separar el identificador del cliente del índice k que ocupa en la lista de la ruta. Si vemos cada ruta como un grafo $R = (V, A)$, V es el conjunto de vértices i y A el conjunto de arcos (i, j). Definimos entonces las siguientes variables de decisión:

- y_{ik} variable binaria que vale 1 si el cliente i es visitado en la ruta k.
- x_{ijk} variable binaria que vale 1 si en la ruta k luego de visitar al cliente i se visita al cliente j. Esta variable representa la matriz de adyacencias de un grafo para cada ruta, $(i, j) \in A$.

El objetivo, entonces, es maximizar la recuperación de energía r_i minimizando los costos de inspección c_i como se ve en la ecuación 5.3.

$$maximizar \sum_{i \in V} \sum_{k=1}^{M} r_i y_{ik} - \sum_{i,j \in A} \sum_{k=1}^{M} c_{i,j} x_{ijk} \tag{5.3}$$

$$s.to \sum_{k=1}^{M} \sum_{j=1}^{n} x_{0jk} = \sum_{k=1}^{M} \sum_{i=1}^{n} x_{i0k} = M \tag{5.4}$$

$$\sum_{(i,j) \in V_k} (\alpha d_{ij} + \beta) x_{ijk} < T \ \ \forall k \in [1, 2, ..., M] \tag{5.5}$$

$$\sum_{k=1}^{M} y_{ik} \leq 1 \ \ \forall i \in [1, 2, ..., n] \tag{5.6}$$

$$\sum_{i=1}^{n} x_{ilk} = \sum_{i=1}^{n} x_{lik} = y_{lk} \ \ \forall l \in [1, 2, ..., n] \ \ \forall k \in [1, 2, ..., M] \tag{5.7}$$

$$c_{i,j} = c_{time}(\alpha d_{ij} + \beta) + c_{distance} d_{ij} \tag{5.8}$$

La ecuación 5.4 restringe todas las rutas (M) a comenzar y llegar desde el punto 0, que en este caso corresponde a la ubicación del Palacio de la Luz. La segunda restricción 5.5 limita el tiempo del recorrido de todas las rutas a no superar T (tiempo total del turno, en este caso 7,5 hs.). La restricción de la ecuación 5.6 asegura que los clientes sean visitados a lo sumo una vez, mientras que la restricción presentada en la ecuación 5.7 asegura conectividad en las rutas, haciendo que todo cliente l tenga una conexión entrante y una conexión saliente si forma parte de una ruta. En la ecuación 5.8 se muestran las componentes del costo de realizar la inspección j luego de inspeccionar i. La constante c_{time} es el costo vinculado a las horas de los funcionarios y $c_{distance}$ el costo del recorrido vinculado al costo del combustible y la depreciación del transporte.

5.4. Enfoque propuesto

No hay soluciones exactas que se puedan aplicar cuando el problema de ruteo cuenta con más de unos pocos cientos de nodos. El enfoque propuesto debe escalar a miles de nodos. En la siguiente sección presentaremos resultados sobre un conjunto de datos de más de 15k nodos. El carácter NP-completo del problema lleva a que las soluciones sean un conjunto de heurísticas con diferentes enfoques, que aseguran a lo sumo un buen resultado en un mínimo local. Algunas de estas alternativas fueron mencionadas en la sección 5.2. Con el objetivo de maximizar el retorno económico en las rutas de inspección proponemos tres métodos que llamaremos de la siguiente forma: *Naive*, *Nearest Neighbor top M* y SOM/TSP.

5.4.1. Método *Naive*

Este método consiste en realizar un *ranking* de clientes a visitar ordenados por la estimación de monto de recuperación de energía. Las rutas comienzan en el punto de origen y van agregando clientes siguiendo el orden del *ranking* hasta que visitar al cliente m_k y volver al origen cumplan la restricción de tiempo de ruta. La cantidad de rutas a realizar es aquella que da la mayor ganancia según la ecuación 5.3. La complejidad de este algoritmo es muy baja, ya que implica solamente el cálculo de n distancias, siendo n el largo de la base.

5.4.2. Método *Nearest Neighbor top M*

El método *Naive* no busca minimizar la distancia y, por ende, el costo es solo una consecuencia del orden elegido para maximizar la recuperación de energía. Proponemos, entonces, un método simple que busque minimizar la distancia recorrida maximizando el retorno. En este método los clientes no serán visitados necesariamente en el orden en el que figuran en el *ranking* de recuperación de pérdidas (recordar que el *ranking* es realizado haciendo una estimación). Desde el punto de origen y en adelante, se evalúa la distancia a los M clientes de mayor recuperación estimada (no visitados) y se selecciona el más cercano. De esta forma no es necesario calcular la matriz de distancias de la base que es $O(n^2)$, en este caso de distancias simétricas, implica el cálculo de $n(n-1)/2$ distancias. En este método se calculan $M \times n$ distancias. La lista de opciones del *top* M asegura visitar en la primeras rutas clientes con una alta estimación de recuperación económica y el uso de *Nearest Neighbor* permite disminuir los costos de inspección, realizando una mayor cantidad de inspecciones por ruta.

5.4.3. Método SOM/TSP

Este método implica resolver el problema del viajante (*travel salesman problem*) para un conjunto de m clientes utilizando el *self organized maps* (SOM) que explicaremos a continuación. El conjunto de m clientes es seleccionado utilizando el *ranking* de recuperación económica estimada. Determinar la cantidad de inspecciones a realizar implica realizar la optimización de rutas para diferentes valores de m sobre la misma lista ordenada de clientes y seleccionar el mejor resultado.

Self organized maps

Self organized maps, también conocido como *Kohonen maps* [55], es un tipo de red neuronal (ANN) de entrenamiento no supervisado que permite disminuir la dimensionalidad de los datos (generalmente a dos dimensiones) y de esta forma organizar y visualizar datos. En su formulación clásica, cada neurona de la red se visualiza como un nodo en una grilla de dos dimensiones. Cada neurona tiene asociado un vector de pesos (W) de la misma dimensión que los datos de entrada. El objetivo del entrenamiento es generar una nueva distribución en el espacio mapeado

que preserve la relación de similitud de los datos de entrada. El entrenamiento se realiza de la siguiente forma:

1. Los pesos son inicializados de forma aleatoria o utilizando las primeras dos componentes de PCA.

2. Para cada dato de entrada se selecciona la neurona más cercana midiendo la distancia del dato a todos los vectores de peso de la red. Al nodo seleccionado se le llama BMU (*best matching unit*).

3. Los pesos de la BMU y los nodos vecinos son actualizados con la siguiente ecuación:

$$W_v(k+1) = W_v(k) + \theta(u, v, k)\alpha(k)(X_i - W_v(K)) \tag{5.9}$$

   Donde k representa el paso de la iteración, u la neurona BMU, v son los nodos de la red. Para dar la actualización entorno a BMU se utiliza un factor de vecindad dado por θ. En un caso simple este coeficiente vale 1 para BMU y los nodos adyacentes y 0 para el resto. Sin embargo, lo más utilizado es hacer la convolución con un *kernel* gaussiano y, de esta forma, ajustar el impacto para cada vecino. Los pesos entonces W_v son actualizados en dirección del dato X_i, utilizando un coeficiente de aprendizaje $alpha(k)$ monótonamente decreciente con las iteraciones.

4. Los datos pueden ser seleccionados por algún método de muestreo o utilizados secuencialmente. Se realizan varias iteraciones hasta la convergencia del método. Esto se puede asegurar seleccionando el coeficiente de entrenamiento correctamente.

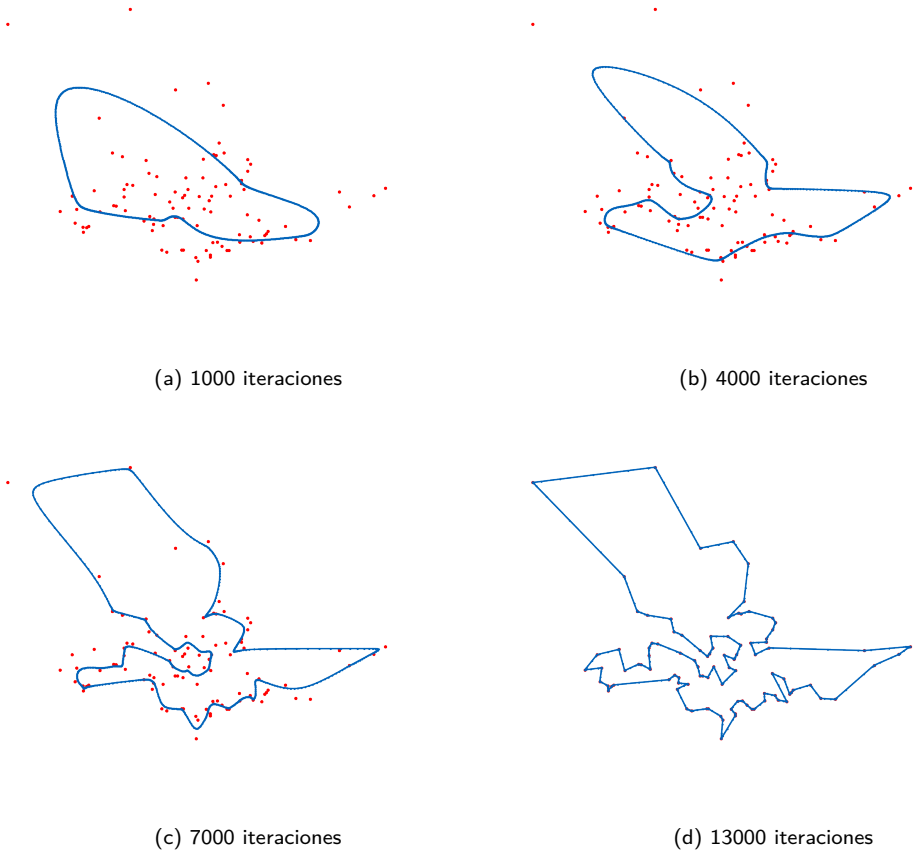
Adaptación de SOM a TSP

Para adaptar la técnica SOM al problema del viajante, la idea principal es modificar la estructura de proximidad de la red pasando de una topología de grilla a un anillo. De esta forma, cada nodo queda en una secuencia. Los datos de entrada y los vectores de peso son de dimensión dos (latitud y longitud). Se asocia a cada BMU la ubicación de un cliente, por lo que cada nodo del anillo se irá acercando a un cliente conforme aumentan las iteraciones.

El valor de α, grande al comienzo, permite exploración y cuando este disminuye con las iteraciones, el algoritmo queda en una etapa de explotación. Para mejorar el ajuste local, el coeficiente de vecindad también debe ser decreciente con las iteraciones. En este trabajo se utiliza la implementación de *Python* de Diego Vicente [100]. El código fue adaptado para manejar entradas y salidas en los formatos necesarios para trabajar con el problema específico.

Ruteo con SOM/TSP

El método de ruteo para máximo retorno económico con SOM/TSP queda entonces de la siguiente forma:



(a) 1000 iteraciones

(b) 4000 iteraciones

(c) 7000 iteraciones

(d) 13000 iteraciones

Figura 5.1: Evolución del ruteo con *self organizad maps* para 100 puntos en Montevideo.

1. Se seleccionan los m clientes de mayor retorno estimado.

2. Se ajusta una ruta que minimice la distancia entre los m clientes usando SOM/TSP.

3. La ruta es dividida en subrutas de forma de cumplir con las restricciones de tiempo (restricción de la ecuación 5.5). En este punto, se toma como hipótesis que si r_o es la ruta óptima entre los nodos $[1..m]$ entonces $r_o[1..k]$ es ruta óptima de $[1..k]$ $\forall k < m$.

4. Se computa el costo total de todas las rutas que permiten inspeccionar a los m clientes y el retorno real obtenido. Se vuelve al punto 1 para cubrir el rango de m que permita identificar cuál es la cantidad de inspecciones con mayor retorno.

En la figura 5.1 se muestra un ejemplo de aplicación de SOM/TSP para resolver el ruteo de 100 clientes aleatorios en Montevideo.

5.5. Experimentos y resultados

A continuación se describen los datos utilizados en los experimentos y una comparativa de los resultados obtenidos al aplicar los tres métodos descritos en la sección 5.4.

5.5.1. Base de datos

La base de datos utilizada cuenta con 168k inspecciones a clientes de UTE realizadas durante los últimos tres años. El porcentaje de clientes con fraude con retorno económico real es de 15.5 % (unos 26k). Esta base es dividida en un conjunto de entrenamiento y uno de test. El conjunto de test corresponde a las últimas 20k inspecciones realizadas en 2020. Utilizando el conjunto de entrenamiento se realiza un modelo capaz de predecir el retorno económico esperado para cada cliente como se describe en el capítulo 4 (referencia [63]). Una vez realizada la predicción sobre el conjunto de test se conforma una base de datos conteniendo los siguientes campos:

- ID: identificador del cliente.
- Long: coordenada geográfica de longitud.
- Lat: coordenada geográfica de latitud.
- retorno_estimado: retorno económico estimado como función no lineal de las variables disponibles.
- y: etiqueta binaria sobre la presencia de fraude.
- retorno_real: retorno económico real (multa).

A los efectos del problema planteado con un único punto de origen de rutas, se decide utilizar la información de los clientes de Montevideo, descartando los demás departamentos. Montevideo es el departamento que concentra la mayor cantidad de inspecciones, 15k de 20k en esta base.

Los clientes de la base de datos pueden ordenarse en un *ranking* dado por la estimación de retorno (usando el campo retorno_estimado). Si los algoritmos utilizados para la estimación de retorno no fueran capaces de extraer patrones de las curvas de consumo, la suma acumulada de retorno_real sería aproximadamente lineal. En la figura 5.2 se ve la suma acumulada desde 0 inspecciones hasta inspeccionar toda la base siguiendo el *ranking* de fraude dado por retorno_estimado.

5.5.2. Resultados experimentales

Los datos utilizados para el cálculo de retorno económico y de distancias son datos reales obtenidos en el marco de un convenio de la Facultad de Ingeniería con la empresa UTE. Los parámetros del modelo de costo son estimaciones realizadas

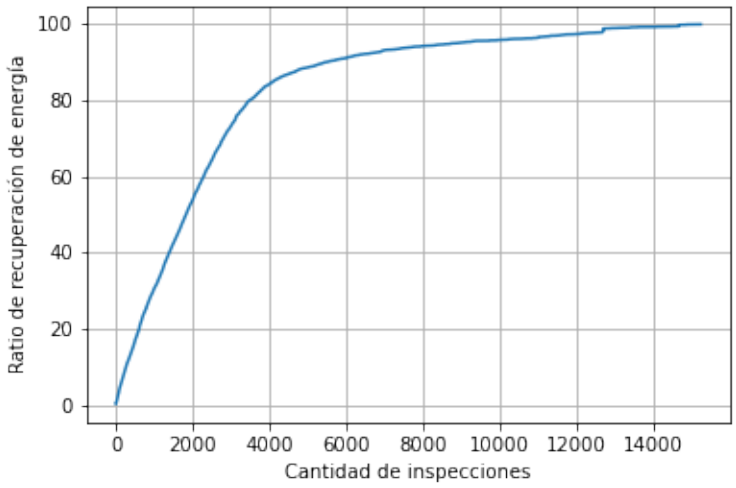


Figura 5.2: Proporción de retorno real acumulado sobre base de test.

exclusivamente para este trabajo. Para los experimentos presentados a continuación se utilizaron los siguientes parámetros de costos en las ecuaciones 5.5 y 5.8: $c_{distance} = 10\$/km$, $c_{time} = \$1000/h$, $\alpha = 5min/km$, $\beta = 10min$.

Para los experimentos realizados con SOM/TPS se utilizó un coeficiente de aprendizaje $\alpha(k) = 0{,}8(1 - 0{,}3 \times 10^{-5})^k$. El coeficiente de vecindad del BMU es calculado utilizando un filtro gaussiano cuya varianza disminuye con las iteraciones de la siguiente forma $sigma(k) = C(1 - 0{,}3 \times 10^{-4})^k$ donde la constante C es proporcional al número de muestras. La red se genera con un total de $8 \times M$ neuronas siendo M la cantidad de clientes. Los experimentos se realizaron para valores de $M = [50, 100, 200, 500, 1000, 2500, 4000, 5000, 7000, 10000, 12000, 15000]$.

En las figuras 5.3,5.4 y 5.5 se presenta, a modo de ejemplo, el resultado de ruteo obtenido para las primeras tres rutas a realizar con cada uno de los métodos propuestos. Analizando los gráficos se puede ver que la cantidad de puntos dentro de las rutas que optimizan el recorrido (métodos 2 y 3) son muy superiores a las que no (método 1) y que el método SOM/TSP genera recorridos por zonas. El algoritmo SOM/TSP al lograr un recorrido más eficiente permite realizar más inspecciones dentro de cada ruta. En la figura 5.6 se presenta el histograma de inspecciones por ruta para los tres métodos.

El objetivo es maximizar el retorno económico por lo que evaluamos las tres estrategias calculando el valor de la ecuación 5.3 para diferentes cantidades de inspecciones y hallamos el valor máximo. En la figura 5.7 se presentan las curvas de ratio de retorno económico obtenidas con los tres métodos. Los resultados se expresan como un porcentaje del dinero total que se recuperó realmente. Obviamente no existe ninguna solución que alcance el 100 %, ya que todas las inspecciones tienen un costo asociado. En la tabla 5.1 se presentan los resultados de los principales indicadores del problema. El método que alcanza el mejor resultado es SOM/TSP, generando una ganancia del 76,3 % realizando 206 rutas de inspección con un pro-

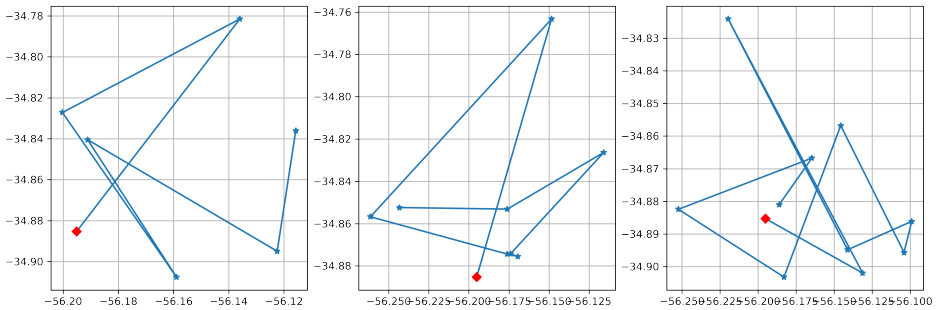


Figura 5.3: Ejemplo de las primeras tres rutas realizadas con el método *Naive*. El punto rojo es la UTE, origen y destino de los recorridos.

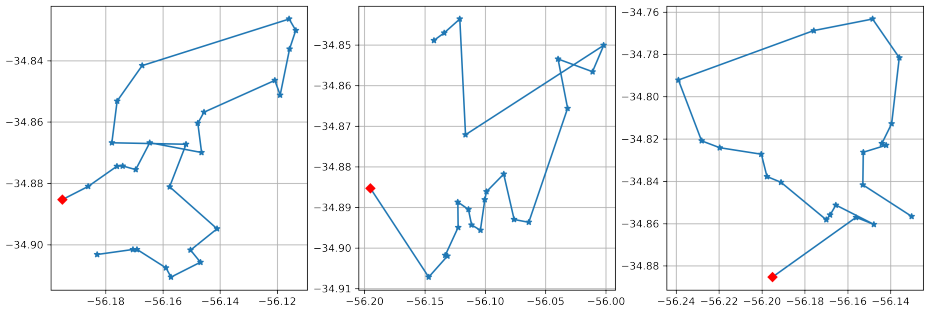


Figura 5.4: Ejemplo de las primeras tres rutas realizadas con el método *Nearest Neighbor Top M*, con M=30.

medio de 34,8 inspecciones por turno.

Método	Naive	Nearest Neighbord Top M	**SMO/TSP**
Retorno total	42,2 %	66,4 %	**76,3 %**
Cantidad de rutas	460	232	206
Cantidad de inspecciones	3895	4520	7000
Inspecciones/Ruta	8,5	19,4	34,8
Precisión	63,9 %	57,9 %	40,8 %

Tabla 5.1: Resultados obtenidos sobre la base de test al evaluar las estrategias de ruteo en su punto de máximo retorno.

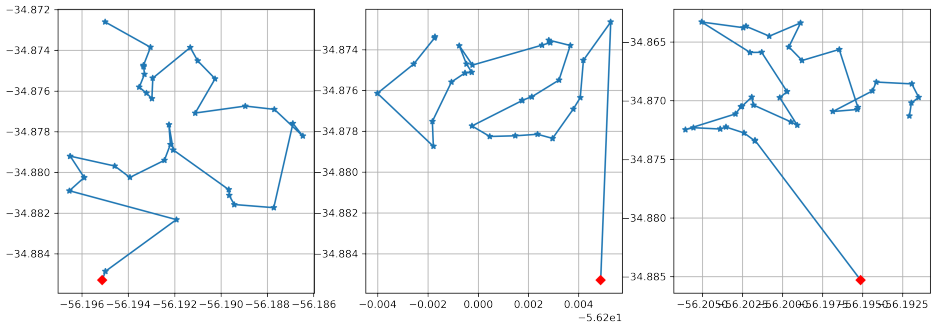


Figura 5.5: Ejemplo de las primeras tres rutas realizadas con el método SOM/TSP para el punto de operación de máximo retorno con 7.500 inspecciones.

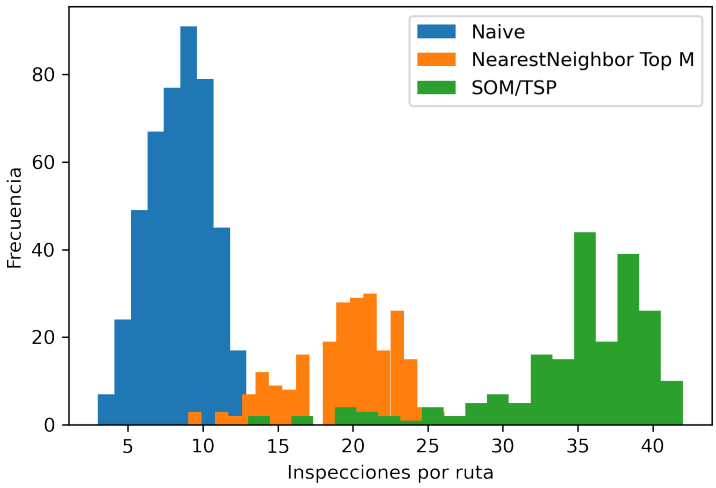


Figura 5.6: En este gráfico se presenta el histograma de inspecciones por ruta luego de elegir el punto de trabajo que maximiza el retorno. El valor medio de inspecciones por ruta es de $8{,}5$, $19{,}4$ y $34{,}8$ para los métodos *Naive, Nearest Neighbor* y SOM/TSP respectivamente.

5.6. Conclusiones y trabajos futuros

En este capítulo se planteó la resolución de un problema de ingeniería concreto: optimizar las rutas de inspección de fraude en clientes consumidores de energía eléctrica, de forma de maximizar el retorno económico. Con base en la propuesta del capítulo 4, se realizó una estimación del perjuicio económico que pueden generar algunos clientes en función de datos de consumo e información adicional. Se trabajó con una nueva base de datos de 168k inspecciones generada en forma conjunta con la empresa UTE.

Uno de los principales aportes de este capítulo es la formalización matemática del problema, junto con una revisión de trabajos relevantes en el área de ruteo de vehículos (VRP). Los problemas de ruteo son problemas NP-completo lo que im-

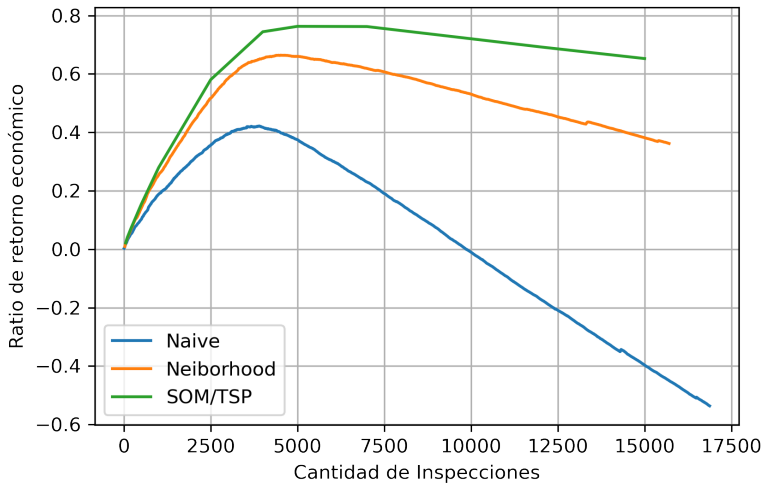


Figura 5.7: Ratio de recuperación económica por método para diferentes números de inspecciones totales. El ratio es el cociente entre la ganancia obtenida con el método ($ingresos - costos$) y el total de ingresos posibles de la base.

plica que no se pueda abordar con soluciones exactas cuando aumenta la cantidad de puntos a unos pocos cientos. Se proponen tres métodos de ruteo, dos de ellos basados en heurísticas simples y un tercero basado en *self organized maps*.

Los resultados obtenidos muestran lo relevante de abordar este tipo de problemas utilizando métricas que reflejen los objetivos principales. En general, los trabajos que realizan clasificación o identificación de fraudes se centran en métricas estándar de clasificación (*precisión, recall*, $F_{measure}$) o métricas que comparan modelos modificando umbrales de clasificación como el área debajo de las curvas ROC y *precision-recall*. En este capítulo se utiliza como métrica el retorno económico. La selección de clientes a inspeccionar tiene en cuenta el compromiso entre retorno y costo. Una estrategia de ruteo de inspecciones no optimizada podría generar pérdidas económicas muy importantes, generando resultados donde la ganancia no alcanza para pagar los costos del trabajo (ver valores negativos de la curva azul en figura 5.7). Incluso, teniendo como objetivo minimizar los costos de rutas, la diferencia en resultados puede ser muy grande. Los experimentos presentados muestran que el método SOM/TSP permite realizar un promedio de 34,8 inspecciones por ruta mientras que el método *Nearest Neighbor TOP M* alcanza una media de 19,4 inspecciones. La diferencia de casi 10 puntos porcentuales en recuperación económica entre ambos métodos se puede traducir en millones de pesos en el problema real.

En trabajos futuros se podrían explorar otras soluciones utilizadas en T-OP (*team organized problem*) y en P-TSP (*travel salesman problem with profits*) por ejemplo *simulated annealing*. También se puede agregar heurística de ganancia marginal para descartar nodos de las rutas y comparar resultados con otros algoritmos del estado del arte incluidos en OR-Tools de Google.

Capítulo 6

Detección de fraudes en medidores inteligentes utilizando curvas de consumo en multiresolución

6.1. Introducción

Acompañar el avance tecnológico es un desafío para las empresas. La migración hacia nuevas tecnologías en el área de la energía es un proceso costoso y que lleva tiempo, pero debe realizarse para garantizar la eficiencia y la competitividad de los servicios. En particular, la migración hacia una infraestructura de medición avanzada (AMI) es un proceso que puede llevar varios años. Estos cambios implican no solo una gran inversión, sino también un plan de sustitución de equipos de medida a lo largo de toda la red de distribución. Incluso, dado el volumen de la inversión y el acceso a financiación, muchas empresas realizan estos cambios en etapas. Esto implica que las empresas convivan con datos de medida provenientes de diferentes fuentes para diferentes clientes, por ejemplo, algunos clientes reportan datos mensuales de su consumo mientras otros clientes pueden ser monitoreados de forma remota con datos cada 15 minutos. Desde el momento que se instala el primer medidor inteligente en una red comienzan a convivir datos de diferente resolución temporal. Los datos de medición inteligente comienzan en fechas diferentes según el cliente, generando bases de datos de series temporales con largos variables.

En la literatura hay numerosas propuestas para abordar este problema de NTL, tanto con datos de medidores inteligentes como con datos de menor resolución generados con tecnologías anteriores [69, 90]. Si bien hay algunos enfoques para incluir nueva información en arquitecturas de aprendizaje profundo [11, 48], no ha habido aportes a la literatura de NTL para utilizar medidas en multiresolución y abordar los períodos donde las tecnologías de medida conviven. Esta coexistencia puede ser estática (por una estrategia de negocio) o puede ser dinámica, si se está en un proceso de migración tecnológica. Los períodos de migración pueden durar años, por lo que el uso de esta información puede ser muy valioso para la compañías de distribución de energía.

Surgen entonces las siguientes preguntas: ¿cómo manejar el volumen de datos generado por los medidores inteligentes?, ¿cómo combinar la información histórica mensual con los datos de los *smart meters* (SM)?, ¿cómo aprovechar todo lo aprendido en NTL para datos mensuales sobre la nueva infraestructura de medida?

En este capítulo abordamos el problema de la transición tecnológica a AMI aprovechando el uso de datos de multiresolución temporal para la detección de fraudes en la red eléctrica. Se propone una nueva arquitectura de aprendizaje profundo para combinar datos de diferente resolución temporal.

6.2. Propuesta

6.2.1. Multiresolución

En una etapa de migración tecnológica de los sistemas de datos, es tan importante mantener los registros históricos como generar valor agregado con la nueva información disponible. En general, se cuenta con varios años de datos de consumos mensuales de energía (kWh/mes) previo a la instalación de medidores inteligentes. Con la nueva AMI se puede tener datos de consumos de energía cada quince minutos para los mismos clientes desde la instalación de los SM. El objetivo es ser capaces de identificar un comportamiento irregular (fraude o rotura) analizando las curvas de consumo. Para ello, proponemos utilizar una arquitectura de redes neuronales de convolución con dos entradas. La serie temporal de datos mensuales pasa por una red convolucional de una dimensión de donde se extraen características relevantes [65]. En una segunda entrada se utilizan los datos de consumos de energía cada 15 minutos de los últimos tres meses para conformar una imagen de 90x96 píxeles (ver figura 6.1). Cada una de las 96 columnas corresponde al consumo de energía en un momento del día y cada fila corresponde a un día diferente. Los experimentos fueron realizados para diferentes largos de datos de entrada, concluyendo que a partir de 90 días la mejora en el desempeño es marginal.

Las redes convolucionales en 2D han demostrado ser muy potentes para identificar patrones en imágenes. En estas redes la detección de patrones es invariante a la traslación, lo que en nuestro enfoque se traduce en invarianza temporal. El objetivo es aprender a identificar un patrón de fraude independientemente de cuando haya sucedido. De esta forma, el algoritmo aprende a identificar consumos de energía sospechosos y luego puede encontrar un patrón similar en otro cliente en otra región de la imagen (otra fecha).

Las CNNs reducen drásticamente la cantidad de parámetros entrenables en comparación con las redes totalmente conectadas. Esto se basa en el hecho de que las imágenes tienen estructuras locales (líneas, formas, colores, etc.) que pueden ser capturadas por un conjunto de filtros (*kernels*). La detección de pequeñas partes permite codificar estructuras más complejas de forma jerárquica. Cada filtro es equivalente a una neurona que tiene conexiones solamente con una región de la imagen. La operación de convolución permite que cada filtro opere con todas las zonas de la imagen, compartiendo los valores de los parámetros entrenables (*kernel weights*).

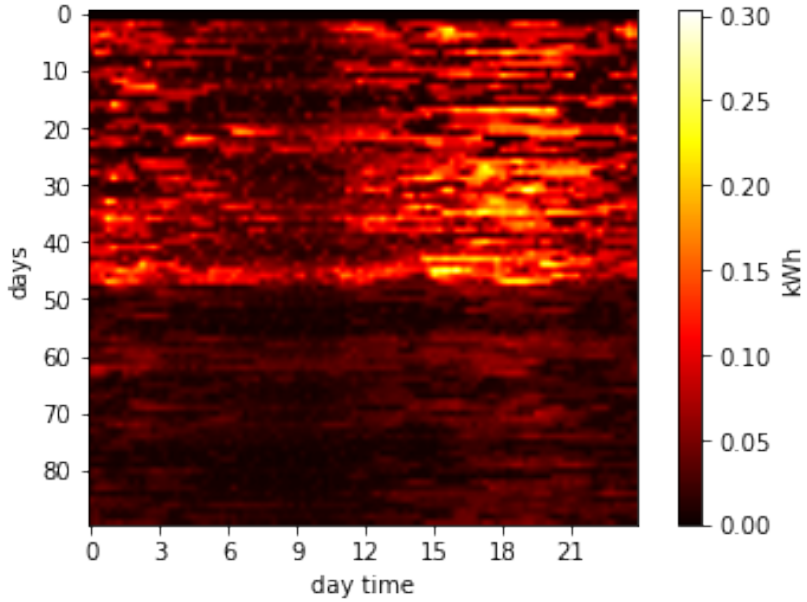


Figura 6.1: Ejemplo de consumo fraudulento de energía en 2D. El eje de las *y* corresponde a los días y el tiempo avanza a medida que se desciende.

Para el caso de una imagen de dos dimensiones I la operación de convolución con un *kernel* K se presenta en la siguiente ecuación

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n),$$

donde m y n son las dimensiones del *kernel*, (i,j) las coordenadas de un punto de la entrada y S la capa de salida de la convolución. Cada capa de convolución está formada por un banco de filtros que se caracteriza por su dimensiones de largo y alto, típicamente 3x3 y la profundidad queda dada por la dimensión del volumen de datos a la entrada. En el caso de aprendizaje profundo se suelen utilizar varias capas de convolución concatenadas. De esta forma, con una primera capa de 64 filtros de 3x3, el volumen de datos a la salida será de profundidad 64. Un filtro de 3x3 en la segunda capa tiene profundidad 64, esto equivale a 576 parámetros entrenables en cada filtro.

La activación de cada neurona está dada por una función no lineal, en este caso utilizamos *rectified linear unit* (ReLU). Las no linealidades permiten representaciones más complejas y son un punto clave del éxito del aprendizaje profundo.

Las capas de convolución son el bloque principal de una arquitectura de un clasificador CNN. Otras capas muy utilizadas en un clasificador CNN son: *pooling, batch normalization, dense* y *softmax* (o sigmoide para dos clases). Las capas de *pooling* permiten reducir la dimensionalidad de los datos. En este capítulo se utiliza *maxpooling*. Una vez que se extraen características de los datos, se puede entrenar un clasificador tipo MLP concatenando algunas capas densas. Para hacer esto se aplana el volumen de datos de la última capa de convolución y se usa como

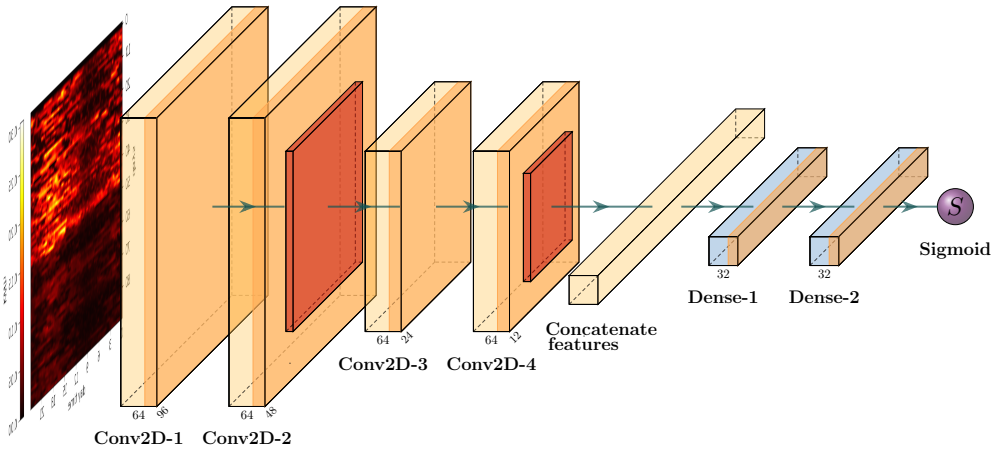


Figura 6.2: Arquitectura CNN2D para detección de fraude con datos de consumo de energía activa quinceminutales.

entrada de la primera capa totalmente conectada (FC). Un ejemplo de arquitectura del detector de NTL con CNN2D se presenta en la figura 6.2.

6.2.2. Arquitectura

En este capítulo utilizamos redes convolucionales, sin embargo, las ideas principales aquí presentadas son independientes de esta elección y podrían implementarse con otras opciones arquitectónicas, como redes neuronales recurrentes. En la sección 6.3 comparamos el rendimiento de diferentes arquitecturas clasificando los datos de consumo de energía de alta resolución.

La detección de fraudes utilizando solamente datos de consumos de energía mensual es una tarea difícil. La representación de tres años de consumo queda reducida solamente a 36 valores. Si bien la extracción manual de características es muy utilizada en el tratamiento de series temporales, para este problema, el uso de los datos crudos en una red de convolución de una dimensión, o algoritmos de *machine learnig* como *xgboost* o *random forest*, han demostrado tener mejores resultados (ver capítulo 3). Proponemos en este capítulo la extracción de características usando una red convolucional en una dimensión (CNN1D). Al igual que con la red de convolución en 2D, basta con agregar algunas capas densas y una capa de salida con una neurona y una función sigmoide para tener un clasificador. Probaremos el desempeño de ambas redes en la siguiente sección.

La arquitectura de multiresolución propuesta combina la última capa de activación de cada red convolucional a la entrada de un MLP y entrena todos los parámetros utilizando una única función de *loss* (figura 6.3).

En NTL el etiquetado de datos es realizado mediante inspecciones *in situ* de personal especializado. En caso de constatar un fraude, la instalación es reparada

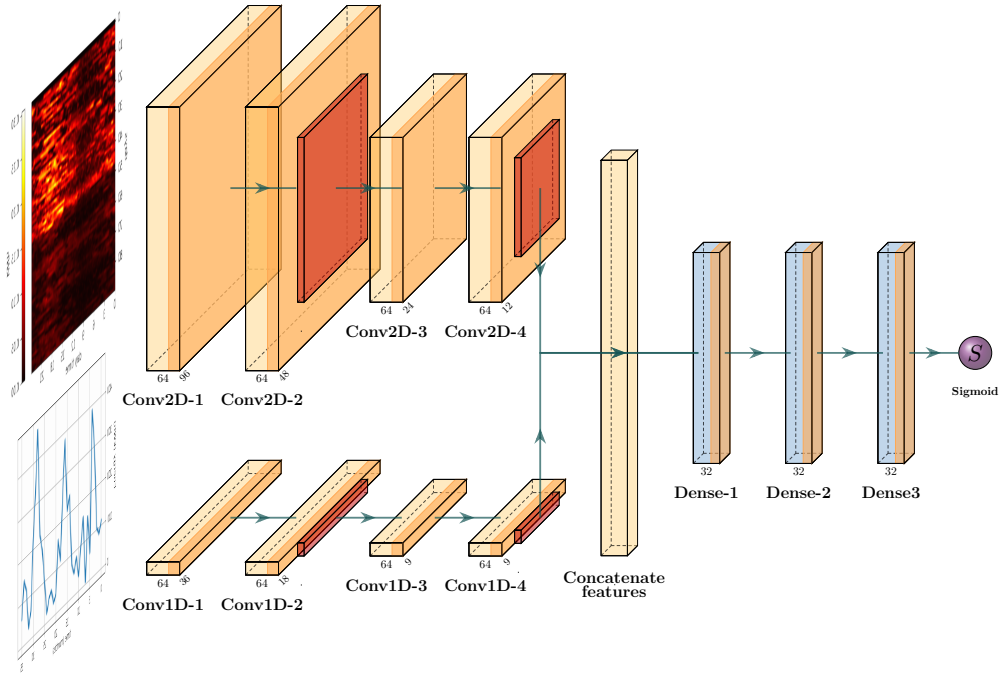


Figura 6.3: Arquitectura de multiresolución para detección de NTL. En la parte superior se representa la entrada de datos de alta resolución en formato de imagen de 90x96 y su procesamiento con cuatro capas de convolución 2D de 64 filtros. En anaranjado se representa la activación ReLU y en rojo las capas de *pooling*. La entrada inferior corresponde a las series de consumos de baja frecuencia (mensual). Estos datos pasan por una red de convolución de 1D con tres capas de 64 filtros. En verde se representa la concatenación de características y por último una serie de capas de redes neuronales totalmente conectadas con activación ReLU. La salida es un *score* dado por una función sigmoide.

y existen sanciones para el cliente. Esto implica que los datos para entrenamiento solamente son válidos previo a la fecha de inspección. Este punto, fundamental para el análisis del problema, parece no ser tenido en cuenta por varias investigaciones que trabajan con datos reales (ejemplo con datos de SGCC). Si tomamos como entrada de alta resolución datos de los últimos 90 días previos a la inspección, veremos que los períodos de datos válidos para entrenamiento difieren entre los clientes. Este problema se resuelve al utilizar redes de convolución que generan invarianza temporal con el enfoque propuesto.

6.2.3. Métricas

Siempre que se entrena un clasificador es necesario definir alguna métrica para optimizar. En particular, al hacer *finetunnig* de los hiperparámetros de un algoritmo, es necesario definir una forma de comparar desempeños. En la gran mayoría de los problemas de clasificación se utiliza por defecto *accuracy*. Pero, como ya se

mencionó en el capítulo 2, esta métrica pierde sentido al trabajar con un problema desbalanceado, donde es importante detectar la clase minoritaria. Al revisar la bibliografía es claro que no hay un consenso sobre las métricas a utilizar en NTL. Por más que se utilicen métricas de comparación de desempeño global de algoritmos, como el área debajo de la curva ROC o de la curva *precision-recall*, en algún momento es necesario definir un umbral y clasificar nuevas muestras para continuar con las tareas de inspección. Al final, lo que importa es el desempeño de las inspecciones realizadas como ya fue presentado en capítulos previos.

Es probable que al momento de entrenar un algoritmo de clasificación no se sepa con exactitud cuál es la cantidad de inspecciones que requiere hacer la empresa. Esto puede depender de muchos factores estratégicos y operativos. Este es uno de los fundamentos de por qué puede ser una buena estrategia utilizar métricas como AUC_ROC o AUC_PR para comparar modelos.

En particular, en problemas de clasificación binaria con desbalance de clases, la curva P-R da una imagen más informativa sobre el desempeño de un algoritmo [19]. En este trabajo utilizamos AUC_PR como métrica de comparación de modelos y selección de hiperparámetros. Utilizamos también como métrica de desempeño principal en la evaluación de experimentos $P@10\,\%$, que es la precisión obtenida al realizar inspecciones etiquetando como positivos al $10\,\%$ de la base. Para facilitar la comparación de modelos y vincular esta métrica con la curva P-R veamos cuál es el lugar geométrico en el plano PR de los puntos de operación para un cantidad fija de inspecciones ($N_i = TP + FP$). Sea N_P la cantidad de positivos y N_i la cantidad de inspecciones a realizar podemos expresar $Recall = TP/(TP+FN) = TP/N_P$ y $Precision = TP/(TP+FP) = TP/N_i$. Combinando ambas igualdades se obtiene

$$Precision = \frac{N_P}{N_i} Recall,$$

que es una recta que pasa por el origen con pendiente N_P/N_i.

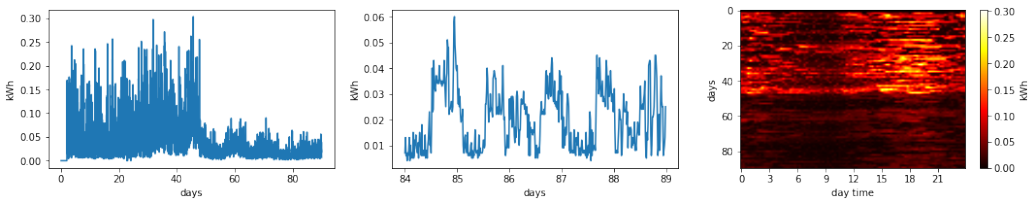
Agregamos la recta de operación al $10\,\%$ en los gráficos de comparación de modelos con curvas P-R.

6.3. Experimentos

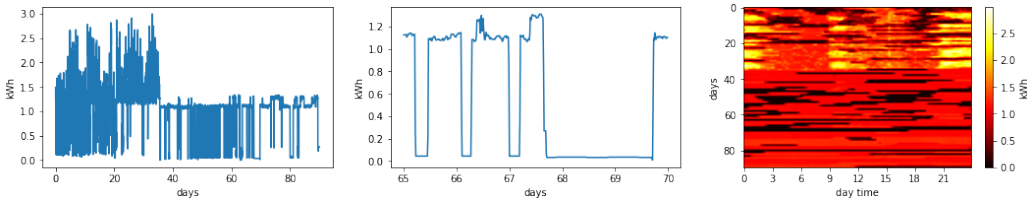
6.3.1. Datos

UTE_SM: base de datos de medidores inteligentes con fraude real

A partir del año 2019 la empresa estatal de generación y distribución de energía de Uruguay (UTE) comenzó una campaña de renovación de medidores de energía pasando de medidores electrónicos o electromecánicos a medidores inteligentes. La base de datos utilizada para los experimentos en este capítulo cuenta con los datos correspondientes a 10.596 clientes con SM, inspeccionados por técnicos de UTE entre enero de 2019 y diciembre de 2020. De cada cliente se cuenta con su curva de energía activa consumida cada 15 minutos desde la fecha de instalación hasta la fecha de inspección. Además se cuenta con el histórico de consumo mensual de



(a) Ejemplo de fraude generado al poner un *bypass* en la bornera del medidor. El resultado es similar al fraude simulado tipo 1.



(b) Esta anomalía corresponde a una intervención intermitente del medidor inteligente similar al fraude simulado tipo 3.

Figura 6.4: Ejemplos de fraudes reales de la base de datos UTE_SM. Se presenta la serie temporal completa de 90 días, un fragmento de cinco días durante la ocurrencia del fraude y la representación de los 90 días de medidas quinceminutales en una imagen.

los tres años previos a la fecha de inspección. Los datos están etiquetados con 1 si se constató un fraude en la instalación (clase positiva) y 0 si son normales (clase negativa). La base tiene 772 casos positivos que representan el 7,3 % del total. En los experimentos utilizamos 90 días de datos por cliente. El 90 % de los medidores de la base tiene más de 90 días de datos válidos. Para el resto de los clientes se realiza *zero padding*.

CER_NTL: Conjunto de datos de base CER con fraude sintético

CER_NTL es un conjunto de datos de consumo de energía eléctrica de fraude sintético. Se crean patrones fraudulentos para simular fraudes típicos en datos de consumo real de una base de datos de acceso público. El conjunto de datos CER fue creado por la Comisión Reguladora de Energía de Irlanda para analizar el comportamiento de los clientes en el consumo de energía [14]. La base de datos contiene, entre otra información, el registro de energía activa consumida cada media hora en 6.435 hogares durante un período de 17 meses. Para la simulación de fraude se tuvo en cuenta la experiencia de técnicos de UTE y otros trabajos académicos [53, 84, 107]. Las lecturas de medición pueden verse afectadas total o parcialmente. Los ataques cibernéticos o las modificaciones de los medidores pueden afectar a la AMI de forma permanente o por períodos de tiempo. El uso de un puente (*byPass*) en la bornera del medidor o una doble acometida sin medidor pueden tener un interruptor de activación que permite el fraude por ventanas de tiempo. El porcentaje de energía robada depende de cómo se lleve a cabo el fraude. Se han presentado modelos similares en otros trabajos [53, 103, 107]. En este capítulo, usamos un conjunto de variables aleatorias para modelar el comportamiento humano cuando ocurre un fraude.

- Fraude tipo 1: Disminución proporcional constante en el tiempo.

$$\hat{p}_{t_i,n} = \nu p_{t_i,n};\ \ \nu \in [min, max]$$

donde $p_{t_i,n}$ es el valor de consumo de energía del cliente n en el instante t_i y $\hat{p}$ el consumo modificado con fraude. Para cada cliente n se asigna un valor fijo de ν con distribución uniforme $\nu \sim U[0{,}3, 0{,}7]$ y una fecha de comienzo de fraude $t_{f,n}$ aleatoria dentro de los 17 meses de datos.

- Fraude tipo 2: Disminución proporcional en ventanas de tiempo diarias.

$$\hat{p}_{t_i,n} = \delta_{t_i} p_{t_i,n}\ \ /\ \ \delta_{t_i} = \begin{cases} \alpha\ si\ t_{start} \leq t_i \leq t_{start} + l \\ 1\ \text{ en otro caso} \end{cases}$$

donde α modela el olvido en el accionamiento del fraude de algunos días, tomando valores $[\nu, 1]$ con distribución de *Bernoulli*. Sea $F \sim Bernoulli(p)$,

$$\alpha = (F-1)\nu + F.$$

El fraude se comete diariamente en una ventana de tiempo. t_{start} y l modelan el ruido en comienzo y duración de la ventana. $t_{start} \sim N(\mu_{ini}, \sigma_{ini})$, se asume el valor medio de la distribución fijo ($\mu_{ini}[n]$) para cada cliente y asignado en forma aleatoria con probabilidad uniforme dentro de una rango horario $\mu_s \sim U(t_{min}, t_{max})$. La duración de accionamiento también es una variable aleatoria con distribución *gaussiana* $l \sim N(\mu_d[i], \sigma_d)$. 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 $\mu_d \sim U(l_{min}, l_{max})$

- Fraude tipo 3: Disminución total en ventanas de tiempo con franja horaria de máximo consumo. Este fraude modela el reporte nulo de consumo y es un caso particular del fraude tipo 2 con $\alpha = 0$

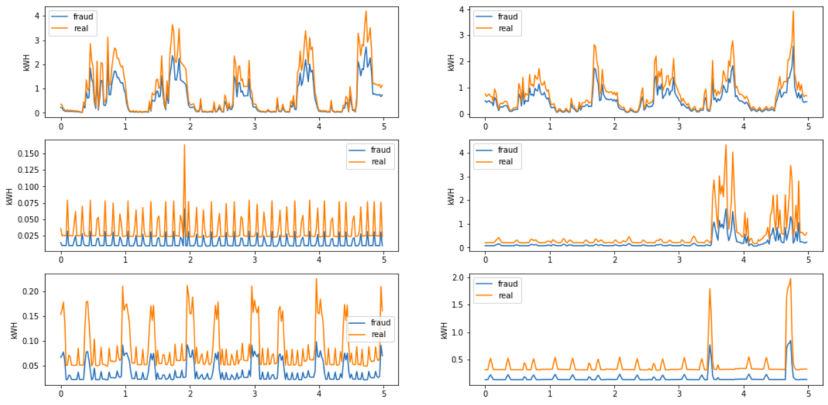
$$\hat{p}_{t_i,n} = \delta_{t_i} p_{t_i,n}\ \ /\ \ \delta_{t_i} = \begin{cases} 0\ si\ t_{start} \leq t_i \leq t_{start} + l \\ 1\ \text{ en otro caso} \end{cases}$$

En este modelo se utiliza $t_{min} = 5pm$ y $t_{max} = 11pm$ para que la ventana de fraude sea dentro del horario de cresta de la demanda energética.

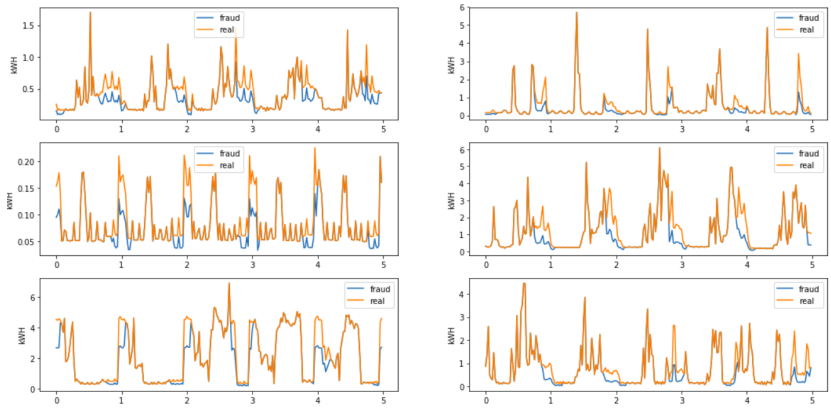
- Fraude tipo 4: Disminución total en ventanas de tiempo sin franja horaria. Este fraude es similar al fraude tipo 3 con la diferencia de que las ventanas de fraude pueden estar en cualquier momento del día. O sea, la asignación de μ_s no tiene restricciones de franjas horarias. Igualmente cada cliente tiene su patrón de realización de fraude.

Para los experimentos de este capítulo se utilizaron los siguientes parámetros en la creación de la base sintética:

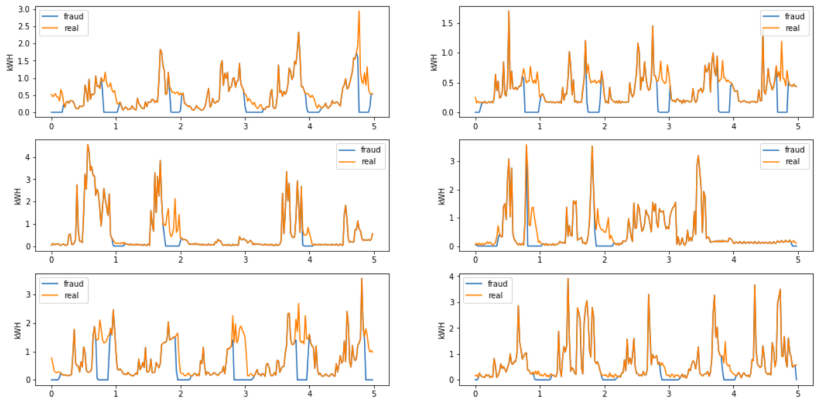
- 8 % de fraudes totales.



(a) Fraude tipo 1



(b) Fraude tipo 2



(c) Fraude tipo 3

Figura 6.5: Ejemplos de generación de fraude sintético. Se grafican cinco días consecutivos de consumos de energía activa para diferentes perfiles de consumo de la base CER. En color anaranjado los consumos originales y en azul la modificación introducida según tipo de fraude.

- 4 % de fraudes tipo 1 con rango de robo entre 30 % y 70 %, según el cliente.
- 2 % de fraudes tipo 2 con el mismo porcentaje de fraude en ventanas comprendidas entre las 17 y las 23 hs. de cada día. La ventanas duran en media entre dos y seis horas ($\mu_s \sim U[2,6]$), con una varianza en su inicio de una hora ($\sigma_s = 1$) y una varianza en la duración diaria de una hora ($\sigma_d = 1$). Además el accionamiento diario no se ejecuta con probabilidad $f \sim U[0,0{,}2]$ modelando olvidos para accionamientos manuales.
- 1 % de fraudes tipo 3 con idéntica parametrización que fraude 2 pero con robo total en ventanas de tiempo.
- 1 % de fraudes tipo 4 con ventanas de mayor duración y sin tener en cuenta los horarios cresta (horarios de mayor demanda con precios diferenciados). Para este último tipo, la duración diaria del fraude puede ir desde las cuatro a las 12 horas de duración con una varianza de una hora, tanto en media como en duración para cada cliente.

6.3.2. Procedimiento

Conjunto de datos de test

En aprendizaje automático se suele separar un conjunto de datos, al cual se le llama conjunto de test. Este conjunto no participa de ninguna de las etapas de entrenamiento y solo se utiliza para reportar los resultados finales de desempeño de los algoritmos. La selección de este conjunto debe ser hecha de forma que represente el problema en el que se está trabajando. Típicamente, en los problemas de clasificación, el conjunto de test se conforma con muestras aleatorias del conjunto de datos disponibles. Sin embargo, en problemas como NTL, lo que se debe medir es cuán bueno es un algoritmo para detectar fraudes en inspecciones futuras. Una selección aleatoria podría incluir sesgos vinculados a campañas de inspecciones direccionadas a cierto tipo de fraudes. En caso de incluir ubicación geográfica como característica, también podría generar un sesgo sobre detección de fraudes que se dan en vecindarios. Es por esta razón que el conjunto de test es seleccionado realizando una partición de los datos según las fechas de inspección.

Se divide el conjunto de datos dejando un 15 % para test y el resto para entrenamiento. El conjunto de datos de test tiene un 8 % de fraudes. Del conjunto de entrenamiento se toma un 20 % de los datos como conjunto de validación. Los hiperparámetros de los algoritmos son ajustados de forma de maximizar el desempeño sobre el conjunto de validación.

Modelos

Evaluaremos de forma experimental las cinco arquitecturas que se describen a continuación:

- **CNN1D**: Clasificador de series de datos de consumos mensuales con la siguiente estructura de capas:

- $[Conv1D+BatchNormalization+ReLU+Conv1D+BatchNormalization+ ReLU + MaxPooloing + Dropout] \times N/2$
- $Flatten + [Dense + BatchNormalization + ReLU] \times M + Dropout$
- $Dense(1) + Sigmoid$

donde N es la cantidad de capas de convolución y M la cantidad de capas totalmente conectadas.

- **Wide&Deep**: Arquitectura de aprendizaje profundo para detección de NTL propuesta en [109]. En la propuesta original se utilizan datos de consumos diarios. En esta implementación utilizamos los datos de alta resolución del medidor inteligente, 96 muestras diarias para UTE_SM y 48 para la base CER_NTL. Se utilizan los 90 días previos a cada inspección.

- **LSTM**: Se implementa una red neuronal recurrente para clasificación de series temporales. Esta arquitectura es utilizada para el tratamiento de datos de medidores inteligentes en [11,48] con resolución diaria. Nuestra implementación utiliza un vector con todas las medidas del día (96 o 48) generando 90 pasos temporales para LSTM.

- **CNN2D**: Clasificador imágenes de consumos eléctricos con datos de medidores inteligentes. La estructura de capas es exactamente la misma que CNN1D con la diferencia que utiliza convoluciones en dos dimensiones y, por ende, *maxpooling* en 2D (ver figura 6.2).

- **CNN_MR**: Esta arquitectura de dos entradas permite identificar pérdidas no técnicas en consumos de energía utilizando la combinación de datos históricos de consumos mensuales con las lecturas de los medidores inteligentes. Las capas de convolución para ambas entradas de datos son las mismas que las utilizadas en los modelos CNN1D y CNN2D. Una vez extraídas las características de las imágenes y la serie temporal de datos mensuales, con las capas de convolución, estas se concatenan para entrenar un MLP (ver figura 6.3). Los pesos de inicialización de las capas de convolución son los obtenidos al entrenar CNN1D y CNN2D. La búsqueda de hiperparámetros para esta arquitectura solo se realiza en las capas totalmente conectadas del MLP como se muestra en la tabla 6.1.

Las capas de normalización se agregan de forma de evitar desvanecimiento del gradiente durante el entrenamiento. La normalización es realizada calculando la media y la desviación estándar de cada *minibatch* de datos, luego de cada una de las capas de la red. Se resta la media y se divide por la varianza. Esto implica la inclusión de dos parámetros más de entrenamiento por cada capa de activación. La implementación utilizada (*Keras*) también computa la estadística con ventanas deslizantes para luego ser utilizada para normalizar el conjunto de test agregando otros dos parámetros más por cada capa e activación.

Hiperparámetros	CNN1D	CNN2D	CNN_MR
capas CNN	[3;4;5]	[3;4;5]	-
filtros	[32;64]	[32;64]	-
kernel size	[3;5]	[(3;3)]	-
learning rate	$[10^{-2} : 10^{-4}]$	$[10^{-2} : 10^{-5}]$	$[10^{-3} : 10^{-5}]$
dropout	[0; 0,3]	[0;0,3;0,5]	[0; 0,3]
capas FC	[2;3]	[2]	[2;3;4]
neuronas FC	[32;64]	[32;64;128]	[32;64]

Tabla 6.1: Rango de valores para búsqueda de hiperparámetros según la arquitectura.

Ajuste fino de parámetros

Dentro de la búsqueda de hiperparámetros hemos incluido, además del *learning rate*, variables que determinan la complejidad de la red, como la cantidad de capas ocultas y la cantidad de filtros o neuronas por capa. Por otro lado, también se incluyen capas de *dropout* como agente de regularización para disminuir el *overfitting*. A los efectos de poder abarcar un amplio espacio de opciones se utiliza búsqueda aleatoria combinada con *early stoping*. Durante cada entrenamiento se monitorea el AUC_PR en el conjunto de datos de evaluación. El desbalance de clases es tenido en cuenta durante el entrenamiento asignando diferentes pesos a cada clase al computar la función de *loss*. Los rangos de valores para cada parámetro se presentan en la tabla 6.1. El entrenamiento de los modelos fue realizado utilizando recursos de cluster.uy, el centro de supercomputación de Uruguay.

6.3.3. Proceso de entrenamiento

La primera etapa del entrenamiento del modelo multiresolución consiste en entrenar los modelos CNN2D y CNN1D. Los pesos obtenidos en las capas de convolución de estos modelos se utilizan para inicializar el modelo CNN_MR. En la segunda etapa, se congelan los pesos de la capa de convolución y se entrenan las capas completamente conectadas. En la figura 6.6 se ve en rojo cómo la *loss* de validación disminuye en la segunda etapa de entrenamiento. Para lidiar con el sobreajuste durante el proceso de entrenamiento, la métrica AUC_PR se monitorea en un conjunto de datos de validación. En ambas etapas del entrenamiento se realiza una búsqueda de hiperparámetros como se indicó anteriormente. Los modelos de mayor desempeño son los que utilizan cuatro capas de convolución y 64 filtros, los más complejos. El proceso de ajuste también determinó el uso de 30 % de *dropout* para la regularización de modelos. En la tabla 6.2 presentamos los resultados obtenidos para el modelo multiresolución sobre los conjuntos de datos de entrenamiento y validación de UTE_SM.

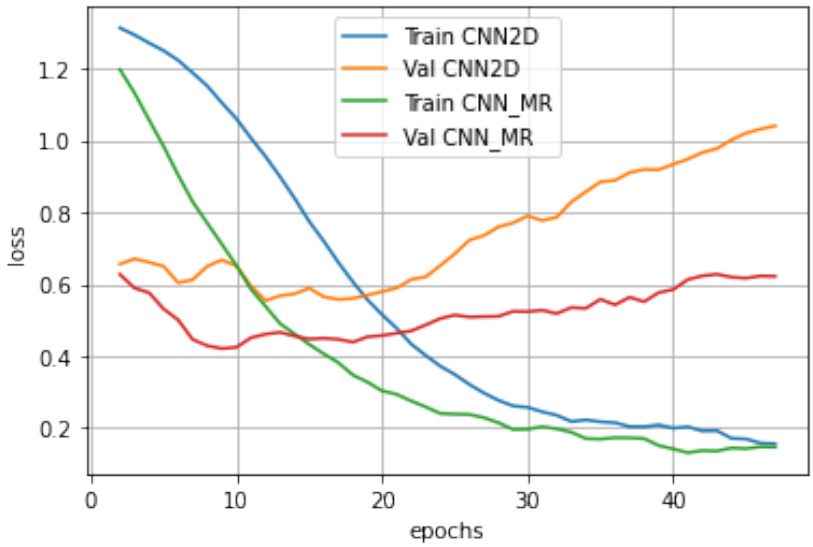


Figura 6.6: Resultados de *loss* para entrenamiento y validación de los modelos CNN2D and CNN_MR.

Conjunto de datos	Precision	Recall	$F_{measure}$	AUC_PR	AUC_ROC
Validación	0,15	0,23	0,18	0,16	0,70
Entrenamiento	0,20	0,35	0,25	0,20	0,77

Tabla 6.2: Resultados del modelo CNN_MR en los conjuntos de datos de entrenamiento y validación para UTE_SM.

6.3.4. Resultados

Se presentan los resultados obtenidos luego de entrenar los modelos de detección de fraude en redes eléctricas sobre las dos bases presentadas. Las tablas incluyen las métricas de evaluación global de los algoritmos AUC_PR y AUC_ROC. También se reporta P@10 % (*precision*), *recall* y $F_{measure}$, fijando el umbral de decisión de forma de predecir como positivos a un 10 % del conjunto de datos de test.

Resultados UTE_SM

En análisis de datos es muy importante contar con datos de casos reales para poder llegar a resultado concluyentes. La base UTE_SM cuenta con más de 10k inspecciones realizadas por personal experto en las instalaciones de medida. Sobre esta base de datos se entrenaron un total de 60 modelos para cada una de las tres arquitecturas monitoreando AUC_PR sobre un conjunto de datos de validación. En la tabla 6.3 se reportan los resultados obtenidos sobre el conjunto de datos de test de la base de datos UTE_SM. Los resultados muestran que el algoritmo CNN2D entrenado con tres meses de datos de medidores inteligentes logra mejores

Modelo	Precision	Recall	$F_{measure}$	AUC_PR	AUC_ROC
CNN1D	0,10	0,12	0,11	0,11	0,56
Wide& Deep	0,15	0,17	0,16	0,12	0,55
LSTM	0,18	0,21	0,19	0,14	0,61
CNN2D	0,19	0,21	0,19	0,15	0,66
CNN_MR	**0,20**	**0,23**	**0,22**	**0,18**	**0,69**

Tabla 6.3: Resultados sobre datos de test de UTE_SM. Se consideran 36 meses de consumo como entrada de baja resolución (para CNN1D y CNN_MR) y 90 días de datos de medidores inteligentes como entrada de alta resolución (para *Wide&Deep*, LSTM, CNN2D y CNN_MR).

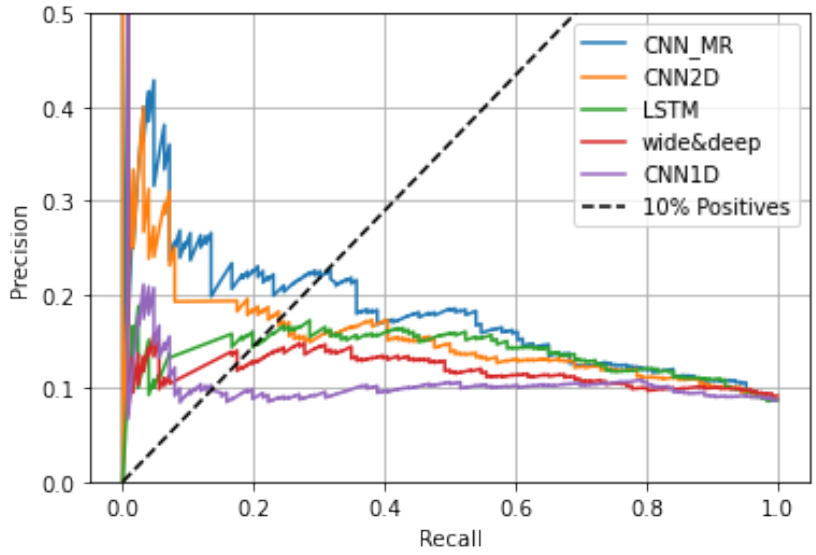


Figura 6.7: Curvas *precision-recall* para el conjunto de datos de test de UTE_SM

resultados que los obtenidos por CNN1D usando 36 meses con valores de baja frecuencia. The CNN_MR El modelo multiresolución supera a los componentes individuales y también supera a otras arquitecturas utilizadas en la detección NTL con datos de medidores inteligentes, como LSTM y *Wide&Deep*. Observando las curvas PR en la figura 6.7 se puede ver que CNN_MR no solo es mejor en el punto de trabajo (línea punteada negra), es mejor para cualquier umbral de decisión. CNN_MR logra aprovechar los datos históricos, disponibles previos a la instalación de medidores inteligentes, para identificar patrones de consumo anormales.

Resultados sobre CER_NTL

Como en los experimentos anteriores, el algoritmo de multiresolución también logra los mejores resultados, superando al modelo que considera datos de baja resolución (CNN1D) y a los modelos entrenados con datos de alta resolución (*Wide&Deep*, LSTM y CNN2D). Esto es cierto para cualquier punto de operación

Modelo	Precision	Recall	$F_{measure}$	AUC_PR	AUC_ROC
CNN1D	0,34	0,38	0,36	0,28	0,76
Wide&Deep	0,36	0,40	0,38	0,36	0,79
LSTM	0,37	0,41	0,39	0,41	0,82
CNN2D	0,40	0,45	0,42	0,49	0,84
CNN_MR	**0,46**	**0,52**	**0,49**	**0,55**	**0,86**

Tabla 6.4: Resultados sobre el conjunto de datos de test de CER_NTL. Se consideran 17 consumos mensuales como entrada de baja resolución(en CNN1D y CNN_MR) y 90 días de mediciones cada media hora como entrada de alta resolución (*Wide&Deep*, LSTM, CNN2D y CNN_MR).

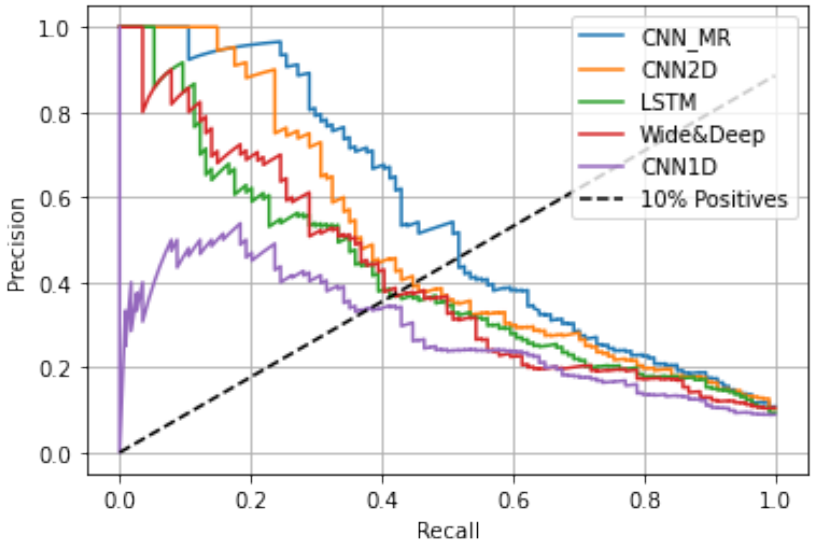


Figura 6.8: Curvas de *precision-recall* para el conjunto de datos de test de CER_NTL.

(ver figuras 6.8 y 6.9). En cuanto al desempeño de clasificación, CNN2D obtiene un P@10 % de 0.40 mientras que CNN1D alcanza 0,34. Teniendo en cuenta que el conjunto de datos tiene solo un 8 % de fraude, los resultados más que cuadriplican una clasificación aleatoria. Asimismo, otras arquitecturas como LSTM o *Wide&Deep* entrenadas con tres meses de datos de alta resolución logran rendimientos superiores a los obtenidos con tres años de datos de CNN1D. El más relevante de los resultados es que la arquitectura de resolución múltiple propuesta (CNN_MR) supera el rendimiento de todos los algoritmos probados, alcanzando una precisión de 0,46 (tabla 6.4).

Para analizar qué aporta cada fuente de datos y por qué el modelo de multiresolución logra mejores resultados, presentamos en la tabla 6.5 el desempeño de P@10 % para cada uno de los cuatro tipos de fraude incluidos en la base. Se puede ver que el algoritmo CNN1D es superior detectando fraudes de disminución proporcional y constantes en el tiempo (tipo 1) mientras que CNN2D es mucho

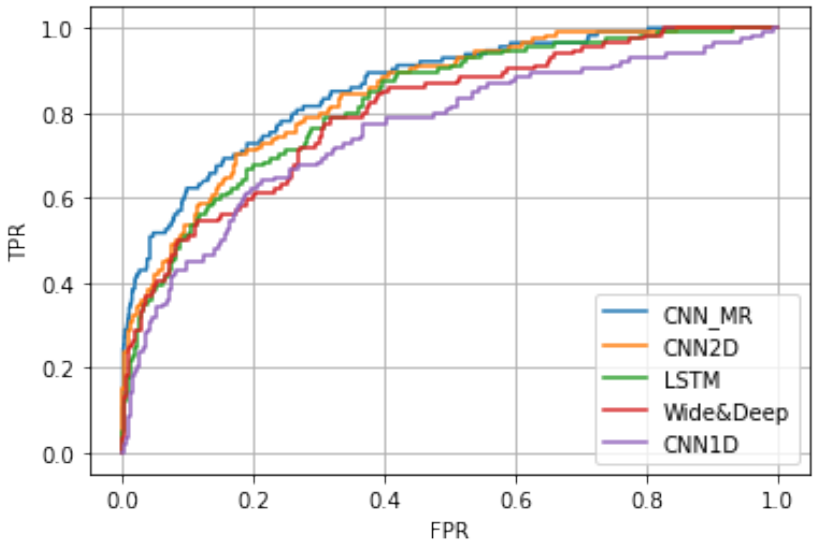


Figura 6.9: Curvas ROC para el conjunto de datos de test de CER_NTL.

Modelo	Fraude 1	Fraude 2	Fraude 3	Fraude 4	todos
CNN1D	**0,52**	0,08	0,07	0,63	0,34
Wide&Deep	0,328	0,269	0,714	0,63	0,36
LSTM	0,34	0,19	0,86	0,63	0,37
CNN2D	0,22	**0,35**	**1,00**	0,94	0,40
CNN_MR	0,36	0,31	**1,00**	**1,00**	**0,46**

Tabla 6.5: Resultados de precisión (P@10 %) por tipo de fraude para la base de datos CER_NTL.

mejor detectando fraudes en ventanas de tiempo dentro de cada día. El modelo de multiresolución no solo tiene el mejor desempeño global sino que también logra buen desempeño en los cuatro tipos de fraude. El fraude tipo 2 es el más difícil, coincidiendo con ser el que menores modificaciones introduce sobre la curva de medida original (ver figura 6.5). El fraude tipo 3, con disminución total del reposte de consumo dentro del horario cresta, es detectado en un 100 % y para el tipo 4 obtiene un 93,7 % con el modelo de multiresolución. CNN1D también logra resultados importantes detectando fraude tipo 4, lo que se debe a que en este las ventanas son de larga duración afectando significativamente el consumo mensual.

Resultados del sistema de detección multiresolución integrando información adicional

Evaluamos agregar a la arquitectura propuesta una tercera entrada con una red MLP para procesar información de características adicionales sobre los clientes de forma análoga a lo visto en el capítulo 3. Utilizando la base UTE_SM y nueve características extras se logra mejorar aún más el desempeño sobre la base de test

obteniendo una mejora en AUC_PR del 10 %.

6.4. Discusión y conclusiones

El desempeño relativo de los modelos probados de forma experimental coincide tanto sobre fraude real (UTE_SM) como sobre el simulado (CER_NTL). A pesar de la relativa coherencia entre los modelos, se puede observar una brecha de desempeño significativa al detectar perfiles fraudulentos reales en relación a los casos simulados. Si bien los diferentes tipos de fraude simulados tienen una base física para su efecto en las curvas de consumo y, en ese sentido, simulan con precisión diferentes configuraciones eléctricas asociadas con el fraude, la falta de un modelo asociado con el comportamiento humano y los patrones de uso de energía de un cliente fraudulento podría explicar la discrepancia en el desempeño.

Los experimentos respaldan nuestra hipótesis principal de que la información de múltiples resoluciones se puede aprovechar durante las transiciones tecnológicas de la infraestructura para minimizar las NTL. Los resultados son consistentes tanto para experimentos simulados en un conjunto de datos disponible públicamente como en datos reales recopilados en el campo por la empresa UTE. Nuestros resultados también sugieren que la elección de la arquitectura (es decir, las capas particulares que componen la solución de red) no es crítica, pudiéndose obtener resultados comparables utilizando otros componentes como, por ejemplo, redes recurrentes LSTM.

Una de las ventajas de utilizar representaciones del consumo en imágenes y redes convolucionales es la posibilidad de interpretar parte del modelo como se muestra en la figura 6.10, donde se visualizan algunas capas de activación como mapas de características.

Presentamos una arquitectura de redes neuronales convolucionales de resolución múltiple para la detección de fraudes en redes inteligentes, inspirada en el cambio reciente de la infraestructura de la red, donde la nueva generación de medidores inteligentes está reemplazando versiones anteriores de medidores digitales y electromecánicos. Nuestro enfoque de resolución múltiple muestra un rendimiento superior a otros algoritmos de detección NTL entrenados exclusivamente en datos de medidores inteligentes (CNN2D, LSTM y *Wide&Deep*). Estos resultados se verificaron con bases de datos con fraudes reales y sintéticos.

También observamos que, cuando se comparan de forma independiente, tres meses de datos de alta resolución tienen más poder predictivo que 36 meses de datos de baja resolución. Esta es una gran noticia para las empresas de distribución de energía, ya que proporciona una prueba cuantitativa de que los nuevos medidores inteligentes mejoran la capacidad de detectar y prevenir las NTL. Además, muestra que incluso unos pocos meses de datos de alta resolución pueden superar las lecturas mensuales de más de un año, lo que sugiere que las empresas no tienen que esperar demasiado luego de que se instalen nuevos medidores para comenzar a detectar actividades anormales. Finalmente, nuestro modelo modular puede aprovechar ambos conjuntos de datos de infraestructuras anteriores y futuras. Los componentes de alta y baja resolución de nuestro modelo pueden potencialmen-

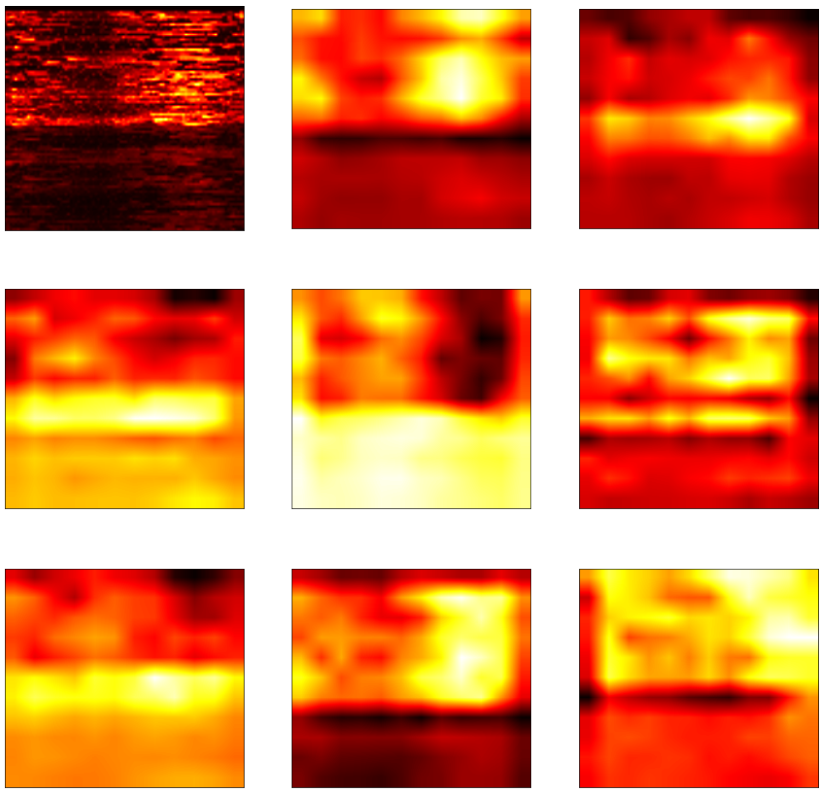


Figura 6.10: Arriba a la izquierda: un ejemplo de la imagen 2D asociada al perfil de consumo de alta resolución de un cliente fraudulento; las imágenes restantes ilustran ejemplos de capas de activación para esta entrada de prueba para un subconjunto aleatorio de *kernels* del modelo CNN2D. Como podemos observar (ver, por ejemplo, la activación de la parte superior derecha), algunas características aprendidas están asociadas con cambios abruptos de consumo, lo que sugiere que algunos núcleos están asociados con la detección de bordes direccionales.

te ajustarse y entrenarse de forma independiente (congelando partes de la red) para aprovechar conjuntos de datos heterogéneos que combinan clientes con una variedad de medidores.

Capítulo 7

Conclusiones

Las pérdidas no técnicas de energía son un problema importante para las economías en desarrollo. En particular, en América Latina y el Caribe las pérdidas anuales alcanzan los 11 mil millones de dólares. El uso de aprendizaje automático para la detección de NTL puede ayudar a las compañías de distribución de energía a minimizar pérdidas y reducir sobrecostos. A continuación se resumen los principales aportes de esta tesis y las perspectivas en esta temática.

7.1. Conclusiones y principales aportes

Los aportes principales de esta tesis incluyen la publicación de trabajos científicos en revistas y congresos arbitrados así como también la transferencia tecnológica de estos desarrollos teóricos. Fruto de la investigación asociada a esta tesis se publicaron cinco artículos científicos [63–67], los aportes específicos asociados a cada trabajo se discutirán a continuación. Complementando la producción académica, los aportes de esta tesis dieron lugar a implementaciones prácticas que están siendo utilizadas actualmente por el Departamento de Recuperación de Energía de UTE para el monitoreo de consumos eléctricos y la prevención y detección de pérdidas no técnicas.

7.1.1. Sobre el impacto económico y las estrategias de inspección

Si bien la mayoría de los trabajos científicos que abordan la detección automática de NTL hacen referencia a la relevancia del impacto económico, los datos económicos que se manejan son parciales o imprecisos. En esta tesis cruzamos información de diferentes organismos internacionales como el Banco Mundial, el Banco Interamericano de Desarrollo, el organismo europeo regulador de energía, la Organización Latinoamericana De Energía y memorias anuales de balances de empresas públicas, con el objetivo de mostrar el impacto real que genera NTL en las diferentes economías. Si bien una de las principales motivaciones del control de pérdidas es económica, previo a esta tesis no había trabajos que se enfocaran en modelar el problema de forma de maximizar el retorno económico. En esta tesis se

propone un abordaje al problema de NTL que tiene en cuenta el retorno económico potencial de cada inspección a realizar (capítulo 4). En función de la estimación de energía recuperada que realizan las empresas para cada fraude detectado, es posible realizar modelos de regresión para estimar el impacto de un fraude potencial. La inclusión de diferentes modelos de costos permite a las empresas estudiar el dimensionamiento de sus estructuras de trabajo. Los modelos de costos de las rondas de inspección tampoco han sido considerados previamente en trabajos sobre NTL. Sin embargo, claramente no es lo mismo inspeccionar 30 clientes que viven contiguos sobre una misma calle que inspeccionar 30 clientes distribuidos en una ciudad. Los algoritmos de detección de NTL, en general, estiman la probabilidad de fraude y sobre un *ranking* de clientes seleccionados se generan las rutas de inspección. ¿Cuál es la mejor estrategia para el negocio? ¿Seleccionar una zona y concentrar toda la fuerza de trabajo allí ahorrando tiempo de traslado y combustible o inspeccionar a los clientes con mayor probabilidad de fraude seleccionando algún criterio de ruteo? En esta tesis se muestra el impacto económico que genera la selección de la estrategia de ruteo en las rondas de inspección. Se formula el problema de optimización, incluyendo las principales restricciones, lo que permite generar un modelo de costos más realista. Incluso limitaciones operativas vinculadas a relaciones laborales pueden ser incluidas en la formulación del problema de optimización. Se propone el uso de *self organised maps* adaptado al problema del viajante como estrategia de ruteo (capítulo 5). Los clientes son seleccionados según la estimación del retorno económico esperado en cada inspección. Clientes cuya estimación de retorno sea baja y visitarlos modifique significativamente los costos de traslado y los tiempos de trabajo, no serán visitados. Las estrategias propuestas pueden ser utilizadas con cualquier algoritmo de clasificación automática con el cual se puedan estimar probabilidades. Por lo que puede utilizarse tanto sobre modelos de árboles de decisión como sobre complejas arquitecturas de redes neuronales. Estos enfoques pueden disminuir de forma significativa la precisión de las inspecciones (considerando la clasificación binaria estándar) y deberían ser utilizadas si están enmarcadas en una estrategia global del negocio con buenas estimaciones del impacto económico de los fraudes y de los costos asociados.

7.1.2. Sobre los algoritmos de detección automática y los datos

En los enfoques de resolución de problemas basados en datos, la calidad de los datos y su preprocesamiento son fundamentales para obtener resultados que generalicen bien. Debido a lo restringido del acceso a datos reales de NTL, la mayoría de los trabajos en el área se basan en la simulación de fraudes. Este tipo de estudios utilizan básicamente curvas de consumo de energía activa y, si bien las modificaciones realizadas modelan el efecto de diversos modos de fraude, no modelan el comportamiento de un consumidor fraudulento. Por otra parte, los sistemas de detección de fraude en NTL obtienen mayor desempeño al incluir otras fuentes de información (información contractual, antecedentes, ubicación geográfica, balances) que difícilmente pueden ser simuladas en NTL. En esta tesis se presenta una de las bases de datos de NTL con fraude real más grandes que han sido utilizadas

hasta el momento en el ámbito académico. Comparamos los enfoques clásicos con extracción de características expertas con los métodos modernos basados en redes neuronales profundas (capítulo 3). Se verifica un rendimiento similar en diferentes arquitecturas de redes (recurrente, convolucional y completamente conectado) al trabajar con consumos mensuales. Aprovechamos las ideas de ataques adversarios para explorar algunos de los patrones más importantes capturados por las redes. Además, observamos que después de aproximadamente $80k$ clientes etiquetados, la ganancia de desempeño de los algoritmos es marginal. Concluimos que la extracción de características expertas de curvas de consumo no mejora el desempeño en la detección de NTL. El enfoque orientado a datos y la extracción intrínseca de características obtiene mejores resultados, tanto sobre datos de consumos mensuales (capítulo 3) como sobre datos de medidores inteligentes (capítulo6). También concluimos que la inclusión de la ubicación geográfica mejora el desempeño en la detección y que el algoritmo más eficiente para tratar con datos de consumos mensuales y variables categóricas en NTL es *extreme gradient boosting* (XGB).

El uso de datos de medidores inteligentes supone nuevos desafíos y oportunidades para los sistemas de detección de NTL. El volumen de datos a manejar aumenta en el orden de miles. Diferentes arquitecturas de aprendizaje profundo han sido propuestas en los últimos tres años para abordar este problema. Lo que ningún trabajo reporta hasta el momento es cómo manejar las etapas de migración tecnológica. La instalación de un nuevo sistema de medición para una compañía de distribución de energía puede llevar varios años. En esta tesis se presenta una arquitectura de aprendizaje profundo para análisis de consumos eléctricos en multiresolución temporal aplicada a la detección de fraudes. El método permite utilizar todas las medidas de consumo de energía disponibles por las empresas de distribución durante los períodos de evolución hacia sistemas de medida inteligente. Los datos de medidores inteligentes son tratados como imágenes y los consumos mensuales históricos como series temporales. Se proponen tres algoritmos: CNN1D para clasificación de series temporales, CNN2D para clasificación de imágenes y CNN_MR para clasificación de datos de consumos en multiresolución. Los algoritmos son probados sobre datos con fraudes reales y datos con fraudes sintéticos. Los resultados experimentales muestran que se logran mejores desempeños con tres meses de datos de medidores inteligentes que con tres años de datos de medidores clásicos. Además la combinación de ambas fuentes de datos logra resultados superiores a utilizar solo datos de medidores inteligentes.

7.2. Perspectivas y trabajos futuros

El problema de detección automática de pérdidas de energía eléctrica es un campo con constante actividad académica. La evolución de las compañías hacia infraestructuras avanzadas de medida (AMI por su sigla en inglés) y avance en el uso de aprendizaje profundo han dinamizado aún más esta área. Si bien esta tesis da respuesta a las preguntas que la motivaron, deja abiertas nuevas preguntas y muestra algunos caminos en los que se podría profundizar.

En esta tesis se discute sobre la relevancia de la métrica a utilizar en NTL

y se propone un enfoque alternativo basado en retorno económico. A futuro se podría explorar una métrica de desempeño que evalúe la precisión en la detección de fraude teniendo en cuenta restricciones operativas. Es difícil saber de antemano cuál será la cantidad de inspecciones a realizar pero sí son conocidos los extremos de la capacidad operativa, por lo que computar el área bajo curvas ROC o PR para todos los umbrales puede no ser representativo. Por ejemplo, se podría utilizar como métrica el área debajo de la curva PR para los umbrales que cumplen con las restricciones operativas.

Independientemente de la métrica que se esté maximizando a la hora de seleccionar los clientes a inspeccionar, la operativa requiere definir rutas de inspección todos los días. ¿Cada cuánto deberían incluirse nuevos clientes a inspeccionar en esas listas y en qué cantidades? La eficiencia de las rutas depende claramente de esto, si dos vecinos son incluidos en dos días diferentes será necesario ir nuevamente hasta al mismo punto en una nueva ruta aumentando los costos. La respuesta a esta pregunta es parte del trabajo futuro en esta línea de investigación. Los próximos pasos sobre el estudio del problema de ruteo podrían incluir el estudio de nuevas estrategias basadas en *graph neural networks*.

En esta tesis se propuso una arquitectura de aprendizaje profundo con la capacidad de manejar múltiples fuentes de datos, incluyendo consumos de medidores inteligentes. Son muy pocos los trabajos en el área que incluyen el tratamiento de alarmas de medidores inteligentes. Estas alarmas también pueden ser tratadas como series temporales y, en algunos casos, por problemas en las instalaciones pueden generarse de a miles (por ejemplo, corriente diferencial). Los técnicos que trabajan con estos datos saben que estas alarmas también esconden problemas vinculados al fraude. Generar estrategias para la inclusión de esta información en los sistemas de detección y medir su impacto es parte de los próximos pasos. A futuro también sería interesante explorar técnicas de aumento de datos para instalaciones con medidores inteligentes donde las técnicas de aprendizaje profundo pueden llegar a obtener mejores resultados aún.

Los patrones de consumo a nivel residencial responden a la suma del consumo de un conjunto de electrodomésticos que, si bien es muy amplio, está restringido por el mercado y por los hábitos. Los avances en el área de monitoreo no invasivo de cargas (NILM) pueden ser utilizados también dentro de NTL. Por ejemplo, la potencia de los termotanques está normalizada y utilizando técnicas de desagregación de fuentes podría ser estimada para cada cliente. Desviaciones de dichos parámetros podrían indicar modificaciones en los sistemas de medida. No hay trabajos académicos hasta el momento en dicha línea.

Apéndice A

Estudio de la inclusión de información geográfica

A.1. Introducción

En este apéndice analizamos la efectividad de diferentes conjuntos de características para la detección de pérdidas no técnicas, incluyendo especialmente características con foco en la ubicación geográfica de los consumidores. La estructura general de la solución implementada consta de dos bloques principales: (a) cálculo de las características y (b) clasificación del cliente (como sospechoso o normal). Para la tarea de clasificación, consideramos el algoritmo *random forest* como un algoritmo de clasificación estándar en el reconocimiento de patrones. El primer paso, la representación de características, es donde nos enfocamos en esta sección y se describirá en detalle a continuación.

A.2. Métricas de desempeño

Este trabajo se basa en el estudio y en la propuesta de mejoras del artículo de Patrick Glauner "Neighborhood Features Help Detecting Electricity Theft in Big Data Sets" [35]. A los efectos de comparar resultados se utiliza la misma métrica de desempeño que en dicho trabajo. Glauner utiliza una aproximación de ROC-AUC y se calcula a partir de un único umbral de decisión:

$$AUC = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right). \tag{A.1}$$

Por completitud y teniendo en cuenta el carácter desbalanceado del problema, también se reportan *Recall*, *Precision* y F_β

$$Recall = \frac{TP}{TP+FN}, \qquad Precision = \frac{TP}{TP+FP}, \tag{A.2}$$

y

$$F_\beta = (1+\beta^2).\frac{Recall.Precision}{\beta^2 Recall + Precision}. \tag{A.3}$$

Tabla A.1: Características adicionales

Cod	Característica	Descripción
01	Proporción de medidas reales	Proporción de los datos que son tomados *in situ* por empleados de UTE.
02	Potencia contratada	Potencia máxima contratada que es limitada en la instalación del cliente por una llave termomagnética.
03	Irregularidades previas	Cantidad de antecedentes de fraude.
04	Días desde la última inspección	Cantidad de días desde la fecha de la última inspección (en caso de haber tenido).
05	Días desde instalación	Cantidad de días desde la fecha de instalación del medidor que está en uso.
06	Mora	Cantidad total anual de días de atraso en el pago de facturas.
07	Días de contrato	Cantidad de días desde la fecha de firma del contrato vigente.
08	Estado del contrato	Estado del contrato: activo, inactivo.

A.3. Características propuestas

Nos enfocamos en clientes comerciales y residenciales para los cuales se obtuvieron los siguientes datos: (i) consumo mensual (total de energía facturada en el mes); (ii) información de facturación adicional como: la potencia contratada, el estado del contrato (activo / inactivo); (iii) ubicación geográfica de los clientes; y (iv) información adicional generada por las inspecciones *in situ* realizadas por técnicos de UTE.

Se seleccionaron dos años de consumo mensual de todo el perfil de consumo del cliente, el mes de inicio fue el mismo para todos los clientes con el fin de evitar el impacto de la estacionariedad climática.

Además, se analizó el uso de un conjunto de características adicionales que en trabajos previos habían mostrado ser relevantes para este problema [56]. Estas se muestran y definen en la tabla A.1 y fueron previamente seleccionadas de un conjunto mayor de características.

Características de vecindad. Como primer set de características analizamos las ocho propuestas por [35]. Con el objetivo de hacer coincidir las dimensiones de las grillas propuestas con las del artículo referenciado, se divide la ciudad de Montevideo con las cuadrículas presentadas en la tabla A.2, luego se calculan las densidades de fraudes (A.4) y las densidades de inspecciones (A.5) para cada grilla.

$$inspected\_ratio = \frac{\#inspected}{\#customers} \tag{A.4}$$

$$NTL\_ratio = \frac{\#NTL}{\#inspected} \tag{A.5}$$

Tabla A.2: Área por celda para cada grilla

Tamaño grilla	área(km2)
3x3	25.3
6x6	6.3
12x12	1.6
24x24	0.4

La figura A.1 representa los valores de las características inspected_ratio y NTL_ratio sobre la grilla más fina (24×24 celdas, cada celda tiene una área de $0{,}4km^2$).

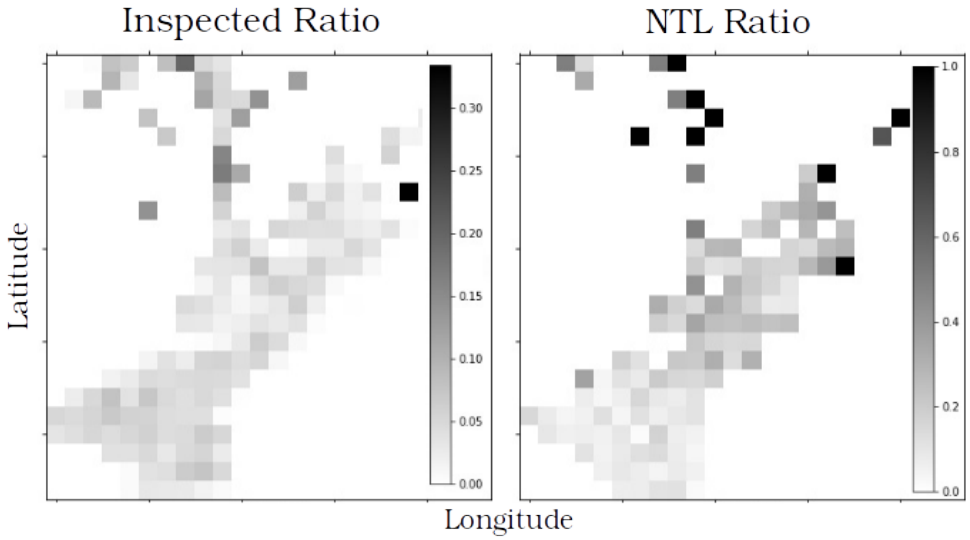


Figura A.1: Densidad de inspecciones y densidad de irregularidades sobre el área de trabajo seleccionada en la ciudad de Montevideo.

Optimización de la grilla.

Además de las características propuestas en [35], en el presente trabajo analizamos el impacto de aprender una topografía de cuadrícula óptima a partir de los datos recopilados mediante la inspección de clientes en la ciudad de Montevideo.

Aprendemos de los datos: (i) cuáles son las posibles grillas que capturan la información de vecindad deseada, y (ii) de este conjunto de grillas posibles, cuál es el subconjunto óptimo de características que debemos mantener.

Para crear diferentes estructuras de cuadrícula que se puedan adaptar mejor a la estructura geográfica de los datos, necesitamos establecer un conjunto de unidades de partición básicas. Por ejemplo, considerando los tamaños de las cuadrículas regulares presentadas en la sección anterior, podemos comenzar considerando (3×3), (6×6), (12×12) y (24×24) como unidades de partición básicas. Entonces, podemos generar nuevos conjuntos de cuadrículas combinando estos conjuntos de tamaños de particiones básicas. Por ejemplo, podemos considerar cuadrículas de dimensiones (3×12), (24×6), (9×3) o cualquier otra combinación posible obtenida de las particiones de tamaño básico $\{3, 6, 12, 24\}$ como se ilustra en la figura A.2.

Una vez definidas las nuevas grillas es posible calcular ambas características (*inspected_ratio* y *NTL_ratio*) en cada una de las grillas. Luego, a partir de este gran conjunto de características, podemos encontrar cuáles son los subconjuntos más discriminativos. Ilustremos lo anterior con un ejemplo simple. Supongamos que generamos nuevas grillas arbitrarias usando como tamaños básicos $\{s_1, \cdots, s_N\}$ y calculamos dos nuevas características en cada una de esas nuevas cuadrículas.

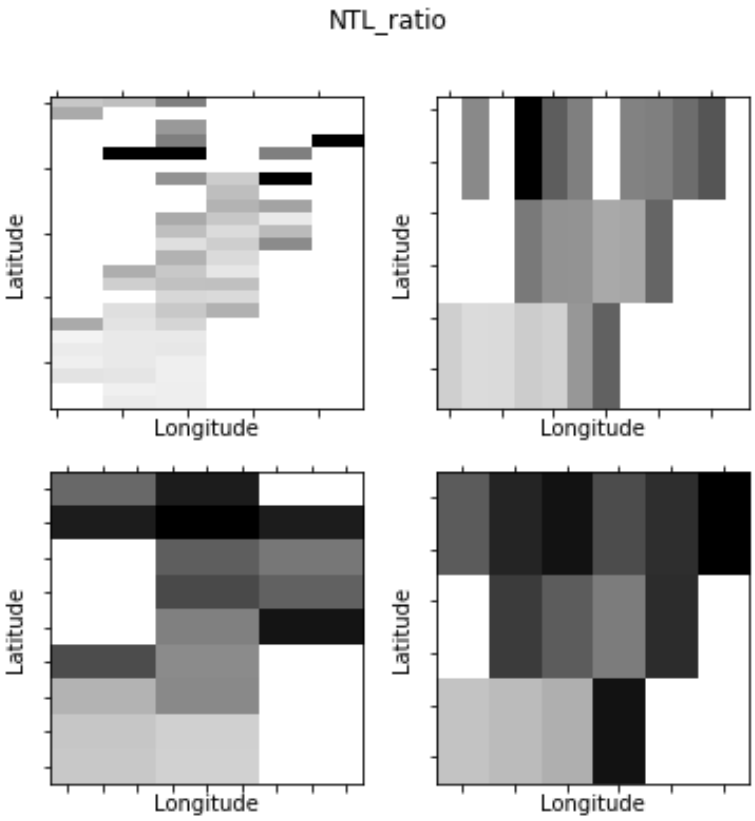


Figura A.2: Algunos ejemplos de grillas generadas aleatoriamente sobre las particiones básicas $\{3, 6, 12, 24\}$

Tenemos en total N^2 posibles grillas, entonces tendremos $2 \times N^2 \stackrel{def}{=} M$ posibles características. Además podemos definir $\sum_{i=1}^{M-1} \frac{M!}{i!(M-i)!} + 1$ diferentes subconjuntos de características, por ejemplo, para $N = 4$, tenemos un total de posibles subconjuntos de características del orden de 10^9.

En este trabajo, encontramos subconjuntos adecuados de características de vecindad definidas en grillas no homogéneas mediante la evaluación de subconjuntos aleatorios de características. Para ello, seguimos los siguientes seis pasos:

1. Establecer las unidades de partición básicas de la cuadrícula, es decir, cuáles son los tamaños unitarios a considerar.
2. Definir un conjunto de grillas aleatoriamente para las características NTL_ratio y Inspected_ratio.
3. Calcular NTL_ratio y Inspected_ratio en sus respectivos conjuntos de grillas.
4. Actualizar el conjunto de características, incluyendo las nuevas características (calculadas) en el conjunto de consumos y características adicionales.
5. Entrenar el clasificador en este nuevo espacio de características usando el conjunto de datos de entrenamiento.

6. Evaluar el modelo anterior en un conjunto de datos de prueba y calcular el AUC. [1]

7. Repetir este procedimiento N veces y definir el conjunto de cuadrículas óptimas como el conjunto que logró el AUC más alto.

A.4. Algoritmo de clasificación

Una vez establecido el espacio de características, es necesario proceder con una segmentación del mismo, identificando las regiones del espacio de características asociadas a cada una de las clases. Para ello, en el presente trabajo se considera el clasificador *random forest*. Este clasificador, muy robusto y popular, consiste en una combinación de árboles de clasificación simples aplicados a subconjuntos de datos generados mediante la selección aleatoria de características. El resultado de cada árbol se utiliza para definir la predicción por mayoría de votos. Este método es resistente al ruido y la generación de subconjuntos de datos reduce los efectos potenciales del sobreajuste. Los parámetros que se utilizan comúnmente para optimizar el modelo de entrenamiento son: el número de árboles y el número de características aleatorias por árbol. Se puede encontrar una descripción y un análisis detallados del método *random forest* en la referencia [7].

A.5. Experimentos

A.5.1. Datos

La base de datos analizada en el presente trabajo se generó a partir de un conjunto de 6029 clientes que fueron inspeccionados *in situ* por técnicos de UTE durante el año 2015 en la ciudad de Montevideo. La región geográfica en la que se seleccionaron los clientes tiene una extensión de $25{,}2km^2$ y un total de $115K$ clientes activos. Las inspecciones realizadas permitieron etiquetar a cada uno de los clientes analizados como irregular (etiqueta 1) o normal (etiqueta -1), en total 11,6 % de los clientes inspeccionados presentan irregularidades asociadas a algún tipo de NTL.

Gracias a un trabajo conjunto con la empresa uruguaya UTE, en el presente trabajo pudimos probar y evaluar las ideas presentadas en un nuevo y actualizado conjunto de datos. Estos datos se recopilaron después de meses de inspecciones *in situ* durante dos años (2015-2017).

A.5.2. Resultados

Este trabajo se centra en la evaluación de diferentes conjuntos de características en aras de mejorar el desempeño en las tareas de detección de irregularidades en los consumos de energía eléctrica. Específicamente, queremos evaluar explícitamente

[1] Los pasos 5 y 6 se pueden realizar en una sola base de datos usando validación cruzada.

el impacto de incluir información geográfica de clientes. Con ese fin, se diseñaron y probaron cinco experimentos sobre la base de datos descrita anteriormente. Cada experimento se realizó utilizando una validación cruzada de 10 particiones, y ese procedimiento se repitió además 10 veces (mezclando aleatoriamente la base de datos). Se utilizó *random forest* como algoritmo clasificador. Los cinco conjuntos de características evaluados son:

1. set "C": 24 consumos mensuales.
2. set "C + A": set C más ocho características adicionales.
3. set "NF": set C + A más ocho características de vecindad calculadas sobre una grilla predefinida [35].
4. set "ONF": set C + A más un conjunto de características de vecindad optimizadas (según describimos en la sección anterior).
5. set "GC": set C + A más las coordenadas geográficas sin procesar (latitud y longitud).

Cada experimento se realizó imponiendo cinco proporciones diferentes de NTL en el conjunto de entrenamiento, lo que ayuda a lidiar con el problema del desbalance de clases. Para cada experimento se reporta, AUC, *accuracy*, *recall*, *precision* y F_1. Las tablas A.3- A.4 y A.5, muestran los resultados obtenidos al utilizar los conjuntos de características "C", "NF" y "ONF" respectivamente. La figura A.3 muestra los valores de AUC obtenidos para cada uno de los conjuntos de características evaluados sobre una base balanceada (NTL prop. de 50 %). Cada experimento se repitió 10 veces (cada vez se realizó una validación cruzada de 10 veces), el AUC mínimo, máximo, medio y la desviación estándar se representan en la gráfica.

Tabla A.3: Resultados usando el conjunto de características "C".

NTL prop.	AUC	Accuracy	Precision	Recall	F_1
12 %	0,521	0,884	0,443	0,050	0,090
20 %	0,542	0,875	0,354	0,110	0,168
30 %	0,575	0,837	0,263	0,234	0,248
40 %	**0,607**	0,750	0,208	0,423	0,279
50 %	0,601	0,594	0,162	0,610	0,256

A.5.3. Discusión y conclusiones

El análisis de los experimentos aquí presentados revela interesantes conclusiones. Primero, los resultados obtenidos usando características de vecindario (" NF ") están en el mismo orden de magnitud que los obtenidos en [35]. Este es un resultado interesante ya que evaluamos estas características en un conjunto de datos completamente independiente, en un país diferente. En segundo lugar, verificamos

Tabla A.4: Resultados usando el conjunto de características "NF".

NTL prop.	AUC	Accuracy	Precision	Recall	F_1
12 %	0,541	0,880	0,401	0,100	0,161
20 %	0,567	0,860	0,315	0,187	0,235
30 %	0,597	0,824	0,264	0,303	0,282
40 %	0,627	0,759	0,226	0,456	0,302
50 %	**0,633**	0,642	0,184	0,620	0,284

Tabla A.5: Resultados usando el conjunto de características "ONF".

NTL prop.	AUC	Accuracy	Precision	Recall	F_1
12 %	0,538	0,885	0,485	0,087	0,148
20 %	0,570	0,873	0,379	0,178	0,242
30 %	0,605	0,837	0,294	0,305	0,299
40 %	0,634	0,769	0,237	0,460	0,313
50 %	**0,646**	0,645	0,190	0,649	0,294

Tabla A.6: Resultados de AUC promedio de 10 experimentos con validación cruzada de 10 particiones

NTL prop.	C	C+A	set NF	set ONF	set GC
12 %	0,521	0,526	0,541	0,538	0,542
20 %	0,542	0,547	0,567	0,570	0,574
30 %	0,575	0,587	0,597	0,605	0,607
40 %	**0,607**	0,621	0,627	0,634	0,639
50 %	0,601	**0,623**	**0,633**	**0,646**	**0,646**

Tabla A.7: Todas las métricas para todos los conjuntos de datos

	C	C+A	set NF	set ONF	set GC
AUC	0,607	0,623	0,633	**0,646**	**0,646**
F_1	0,279	0,273	0,284	0,294	0,294
Recall	0,423	0,644	0,620	0,649	0,650
Precision	0,208	0,173	0,184	0,190	0,190
Accuracy	0,750	0,608	0,642	0,645	0,643

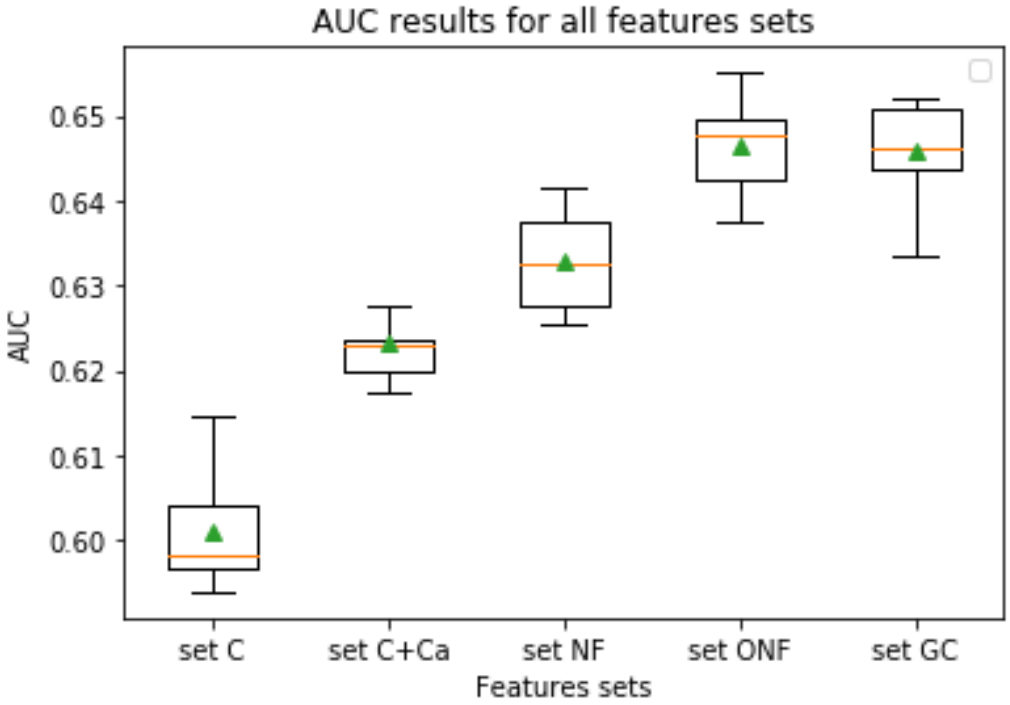


Figura A.3: AUC para los diferentes conjuntos de características evaluados. El cuadro se extiende desde los valores del cuartil inferior al superior de los datos, con una línea en la mediana y un triángulo en el valor medio.

que la inclusión de características adicionales mejora significativamente las tasas de detección de NTL, compare, por ejemplo, los resultados obtenidos con las características " C " y " C + A ". En tercer lugar, investigamos que las características de vecindario se pueden mejorar definiendo topografías de grillas óptimas, lo que nos permite capturar de una manera más flexible la estructura geográfica de una ciudad. En cuarto lugar, mostramos que si consideramos la ubicación geográfica de los clientes (es decir, la latitud y longitud de su dirección) se pueden obtener resultados muy competitivos. Esta última observación es muy importante desde un punto de vista práctico, ya que la inclusión de este tipo de información se puede realizar de una forma muy sencilla.

Apéndice B

Transferencia tecnológica y desarrollos aplicados en UTE

B.1. DAICE: Detección automática de irregularidades en consumos eléctricos

Previo al comienzo de esta tesis participé en el desarrollo de una herramienta para la detección automática de irregularidades en consumos de energía eléctrica para la empresa estatal UTE. Previo y durante el transcurso de la tesis el transcurso de esta tesis hemos realizado varios trabajos en conjunto con UTE desde el Instituto de Ingeniería Eléctrica de la Facultad de Ingeniería (UdelaR) en el marco de proyectos de investigación y convenios de colaboración. Se resumen a continuación algunos de los cambios realizados a dicha herramienta.

B.1.1. Cambios en DAICE

El departamento de recuperación de energía cuenta con la herramienta DAICE (Detector Automático de Irregularidades en Consumos Eléctricos) desarrollado por este grupo de investigación en el marco de proyectos conjuntos. El sistema que había sido testeado en Montevideo en 2018 con una prueba de campo en la zona Goes alcanzaba el 15.4 % de precisión y superaba resultados previos de 9.8 % en esa zona para inspecciones programadas con criterios analíticos.

Durante este convenio la herramienta fue mejorada en varios aspectos. Se simplificó su uso eliminando variables usadas solamente con fines académicos. Se agregaron algoritmos que demostraron un mejor desempeño como es el caso de *extreme gradient boosting* (ver capítulo 3).

Dentro de las mejoras realizadas al sistema existente se destaca la inclusión de un nuevo modo de uso. La versión DAICE 1.1 incluye el modo de funcionamiento de Máximo Retorno Económico descrito en el capítulo 4 y publicado como producto de esta tesis en la revista IEEE Transactions on Power Systems [63].

Otra de las principales mejoras del sistema es el uso de bases de datos etiquetadas de mayor tamaño y cobertura. En trabajos previos llegamos a trabajar

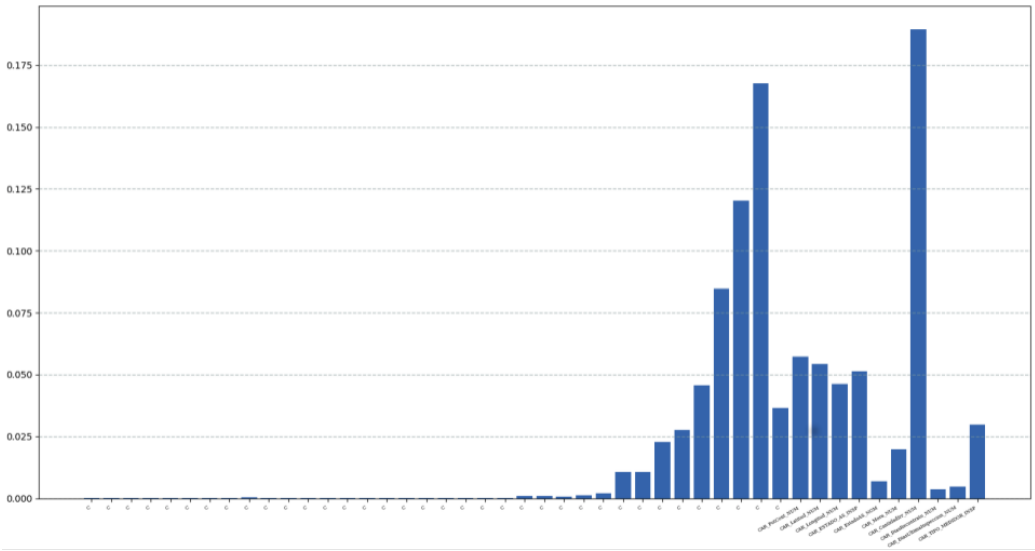


Figura B.1: Importancia de las características utilizadas en DAICE 1 para la construcción de los modelos con *random forest*.

con hasta 30k inspecciones y durante este proyecto pudimos estudiar bases de hasta 300k inspecciones, cubriendo además todo el territorio nacional (figura B.2). Previamente, las consultas a bases de datos se hacían desde Microsoft Access y el equipo del Departamento de Recuperación de Energía (DRE) realizó una ingeniería inversa para realizar consultas con SQL Oracle. Esto disminuyó drásticamente el tiempo de extracción de datos y mejoró la calidad de las bases de investigación. Dentro de las consultas se pudieron agregar otros datos como, por ejemplo, el tipo de medidor de cada punto de servicio. Todas las características fueron evaluadas para cuantificar el aporte a la discriminación de clases de los modelos. En la figura B.1 se ve la importancia de cada una de las características utilizadas en DAICE destacándose los últimos seis meses de consumo, la ubicación geográfica, los antecedentes, el tipo de medidor y el estado del acuerdo de servicio.

Los analistas de UTE pudieron comparar resultados al cruzar modelos de diferentes partes del país, llegando a la conclusión de que lo más conveniente era generar modelos generales basados en toda la información disponible.

La versión que se encuentra en producción también eliminó las interfaces de usuario, dándole versatilidad de uso al analista. Los talleres de formación realizados con UTE permitieron al equipo de DRE apropiarse de la herramienta y participar de la programación de la nueva versión.

B.1.2. Pruebas de campo

Fueron realizadas varias pruebas de campo. En mayo de 2019 se realizó una campaña de inspecciones llamada Etapa 1 realizando más de 800 inspecciones en zona Goes obteniendo 19.9 % de precisión. El sistema fue utilizado nuevamente en una campaña de validación en mayo de 2020 que incluyó más de 3k inspecciones

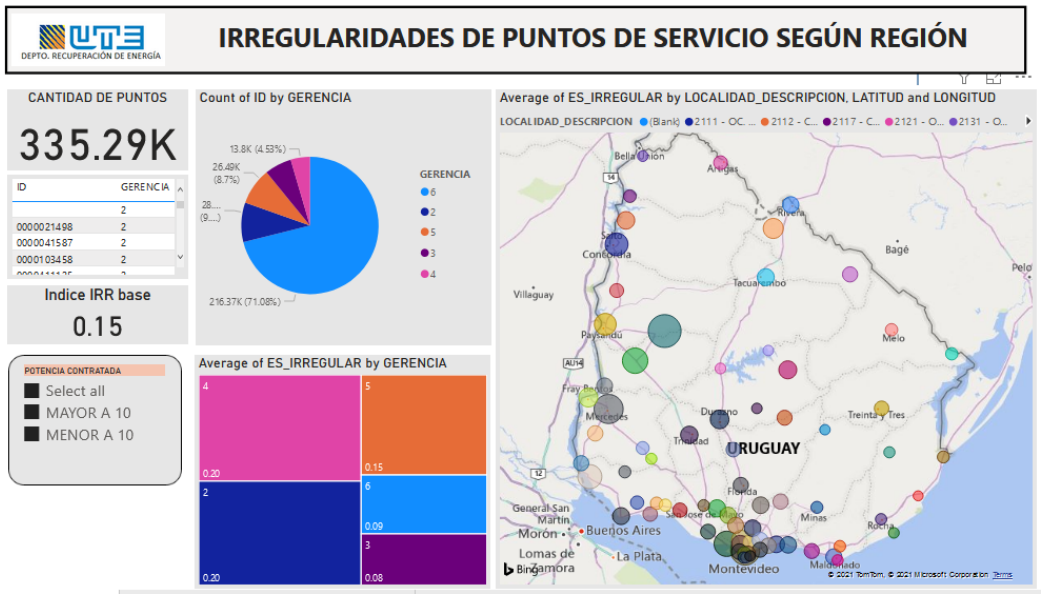


Figura B.2: Visualización de inspecciones históricas que conforman la base de datos de entrenamiento y evaluación de algoritmos.

en todo el país obteniendo una precisión del 16.0 %. Dado que la estimación de fraudes es del orden del 6.0 % el resultado supera la aleatoriedad en un factor de 2.7.

Si bien durante la pandemia que comenzó en marzo de 2020 se siguieron haciendo inspecciones, el porcentaje de efectividad cayó, lo que, entre otros factores puede deberse a que los protocolos no permitían el ingreso a las propiedades de los clientes. También fueron detectados algunos errores en los procedimientos de extracción de datos utilizados en dichas campañas.

B.2. DeepDAICE: Aprendizaje profundo aplicado a detección de irregularidades en consumos eléctricos

Este apéndice es parte del informe final del proyecto "Detección de anomalías en medidores inteligentes", entregado a UTE en abril de 2021 en el marco del convenio UTE-FIng.

En el contexto de un convenio, y en paralelo con la investigación del uso de algoritmos de *deep learning* para la detección de fraudes en medidores inteligentes, se comenzó con el desarrollo de una nueva herramienta para su uso con datos de consumos quinceminutales para UTE. Debido al volumen de datos que implica la nueva infraestructura de telemedida de UTE fue necesario definir una forma de manejo de datos escalable y un *framework* capaz de procesar y manipular toda esta información. A continuación se describe este nuevo sistema llamado DeepDAICE, desarrollado en colaboración con el DRE de UTE.

B.2.1. Manejo de datos

Los datos de consumos mensuales utilizados en el sistema DAICE podían ser manejados en memoria incluso para cientos de miles de clientes. Un PC con 12G de RAM o una GPU del cluster.uy podían manejar estos datos con un abordaje clásico de *machine learning*. Pero pasar de una medida por mes a un dato cada 15 minutos genera un volumen de datos 3k veces superior. Para poder procesar estos datos o utilizarlos para entrenar algoritmos de aprendizaje automático es necesario manejar en memoria una porción reducida. A esto se le llama trabajo por lotes (*batch*). La información permanece en disco duro y es accedida de a lotes según se defina en cada proceso. La estructura de archivos tiene que ser capaz de crecer sin afectar el desempeño del sistema. De esta forma se puede pasar de trabajar con bases de datos de unos pocos Gbytes a bases de Tbytes.

Dado que UTE se encuentra en un proceso de migración hacia medidores inteligentes, la cantidad de datos disponibles varía para cada cliente. Es por eso que las series de datos a manejar son de largo variable y es algo a tener en cuenta no solo para el manejo de datos, sino también para el tipo de algoritmos a utilizar.

El formato de datos utilizado también influye en el costo de cómputo necesario y en el tiempo de ejecución de las rutinas. Si se entrena con una base de 32k clientes un algoritmo con 100 iteraciones con *batches* de 32 clientes, el sistema necesita leer de disco 100.000 veces, por lo que los tiempos de lectura dependiendo de la aplicación podrían ser relevantes.

En este contexto definimos utilizar el formato HDF5, un formato de datos jerárquicos diseñado para almacenar y organizar grandes volúmenes de datos. Gracias al indexado de tablas B-tree este formato es ampliamente utilizado para el manejo de datos de series temporales. Al ser un archivo binario, la velocidad de acceso a variables categóricas es muy superior que utilizar archivos CSV. Si bien existen otros formatos como *parquet* o *feather* que son algo más rápidos, optamos por HDF5 por lo intuitivo de su trabajo y con librerías muy utilizadas en el manejo de datos en Python como lo es *pandas*. La estructura jerárquica nos permite ampliar la base con nuevas informaciones de clientes e incluso ampliarla fácilmente sin necesidad de crear el archivo nuevamente.

La extracción de datos en UTE se hace generando un archivo CSV por cliente. El equipo de informática de UTE genera rápidamente estos archivos que luego son procesados para incluirse en las bases de deepDAICE. Ampliar la base es simplemente agregar archivos en un directorio y correr un procesos de verificación que agrega *keys* en un HDF5.

B.2.2. Plataforma

Dado que UTE es una de las principales empresas financiadoras del centro nacional de supercomputación cluster.uy, optamos por utilizar dicha infraestructura para realizar las tareas de entrenamiento de modelos y clasificación.

Cluster.uy cuenta con 28 nodos de cómputo, cada nodo tiene 40 CPUs Xeon Gold 6138, 128GB de memoria RAM, una GPU NVIDIAP100 y un espacio en disco de estado sólido de 300GB.

deepDAICE se mantiene en un repositorio en gitLAB de Facultad de Ingeniería que está accesible tanto para los funcionarios del DRE UTE como para los investigadores del grupo de Facultad. Utilizando el control de versiones GIT el código se mantiene actualizado en la partición de UTE del *cluster* y se ejecuta en un nodo utilizando la potencia de la GPU.

B.2.3. Algoritmos

Se implementaron funciones que permiten la alimentación de datos en *batches* tanto para datos extraídos por el sistema DAICE como para la lectura de datos de medidores inteligentes.

Breve descripción del código:

- **class_ntl_sm.py**: Generador de datos de la clase *Sequence* que recibe una lista de punto de servicios (PS) y devuelve lotes de datos. El código permite acceder aleatoriamente a datos en las diferentes épocas de entrenamiento. También acceder a otros datos de los clientes en otros formatos, por ejemplo, utilizando los archivos de la versión anterior DAICE. De esta forma puede utilizar arquitecturas más complejas que integran los datos quinceminutales con información contractual o medidas de consumo mensual histórico. Con el parámetro *mode* se selecciona el formato de datos generados. Los modos de generación incluyen: serie temporal 1D, representación matricial de consumo de alta resolución, *wide&deep* de alta resolución y cualquiera de los anteriores agregando datos de consumos mensuales, y otro conjunto de datos de características extras.

- **models.py**: Se usa también como librería y permite crear los modelos de aprendizaje automático. Contiene las siguientes funciones:

  - crear_modelo_CNN2: Crea modelo CNN2D según parametrización incluyendo cantidad de filtros por capa, cantidad de capas, *learning rate*, tipo de activación *dropout* y cantidad de neuronas y capas en etapas finales con redes *fully conected*.
  - crear_modelo_CNN2_extra: Crea un modelo CNN2D con dos entradas procesando las características extra (no consumos) de las bases de datos DAICE en paralelo con el HDF5.
  - crear_modelo_cnn_multi: Crea un modelo de multiresolución como el presentado en el capítulo 6 combinando datos de las extracciones de DAICE y del HDF5 de medidores inteligentes.
  - crear_modelo_wide_deep: Implementación del algoritmo *wide&deep* [109].
  - cnn1d: Permite generar un modelo de redes de convolución en una dimensión para el procesamiento de las series temporales de consumos.
  - best_model_cnn1d: Utilizando la función anterior realiza una búsqueda aleatoria de parámetros para maximizar el área debajo de la curva *precision-recall*.

  - crear_modelo_fullDAICE: Crea un modelo de detección de tres entradas que combina una arquitectura CNN1D para datos mensuales, una CNN2D para medidas de consumo de alta resolución y una red MLP para características extras.

- Código de entrenamiento de modelos: El hecho de contar con funciones para la creación de modelos facilita la búsqueda de hiperparámetros que maximicen el desempeño. El repositorio incluye varias rutinas de entrenamiento y selección de modelos.

B.2.4. Resultados de deepDAICE

Base de datos

Se conformó una base de datos de 10596 inspecciones a puntos de servicio con medidores inteligentes de la marca KAIFA, realizadas entre setiembre de 2018 y diciembre de 2020. La mayoría de las inspecciones corresponden a 2020, como se puede ver en el histograma de la figura B.3. La base que denominaremos UTESMDB se conforma por dos archivos *data.H5* y *data_DAICE.csv*. Cada PS tiene una *key* de la estructura de datos HDF5 en el archivo *data.H5*, conteniendo todos los datos disponibles para el cliente en las fechas previas a la inspección. En el archivo *data_DAICE.csv* cada cliente es una fila de datos conteniendo los consumos mensuales desde enero de 2014 y las siguientes características: *potencia contratada (latitud, longitud)*: la ubicación geográfica del medidor; *pago atrasado*: los días acumulados de demora en el pago de facturas; *historial de fraudes*: el número de irregularidades anteriores detectada; tipo de medidor; estado del acuerdo de servicio; días desde la renovación del contrato y proporción de lecturas mensuales reales.

La base contiene 772 irregularidades detectadas en medidores KAIFA, lo que da un porcentaje de irregularidades del 7.3 %. Dada las características del problema y el contexto de migración de tecnología, el tiempo de datos disponible varía entre clientes. En la figura B.4 se ve cómo disminuye el tamaño de la base si se restringe el largo mínimo de análisis a un numero dado de días. En la figura se puede ver que si se quiere trabajar con clientes que tengan al menos 300 días de datos, la base pasa a tener 4.500 clientes con una proporción de fraude del 10 %.

El uso de algoritmos que incluyen capas de convolución en las primeras etapas permite que los modelos tengan invarianza temporal en la detección de patrones. De esta forma se puede trabajar con series temporales de largos diferentes y aprovechar al máximo la información disponible.

Resultados

Para realizar experimentos que representen de mejor forma el caso real, se divide la base dejando las últimas inspecciones (inspecciones posteriores al 15 de octubre de 2020) para realizar los test. El conjunto de entrenamiento (todas las inspecciones previas al 15 de octubre de 2020) se divide también dejando un 20 % para ser utilizado en la validación de los modelos.

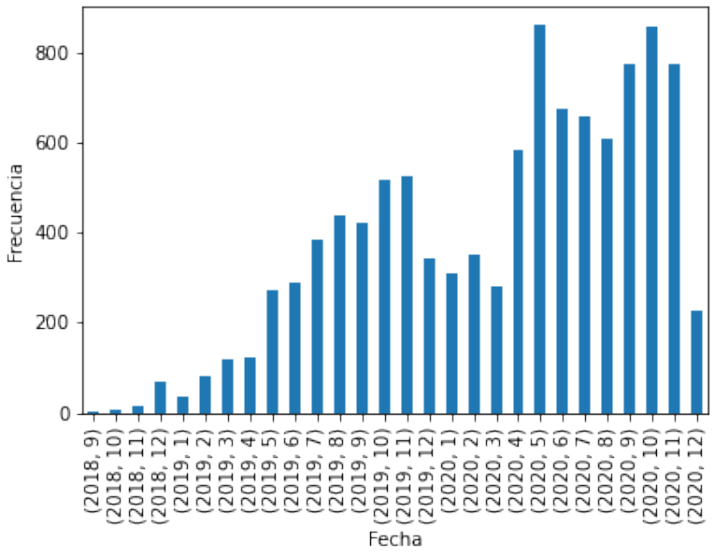


Figura B.3: Distribución de cantidad de clientes de la base UTESMDB según fecha de inspección.

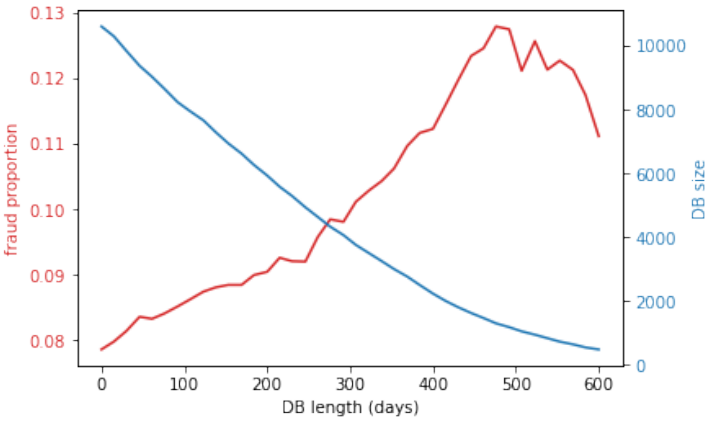


Figura B.4: Variación de tamaño de base y proporción de fraudes dado un largo mínimo de datos en días.

El ajuste de hiperparámetros, como, por ejemplo, el *learning rate* se realiza buscando maximizar el área debajo de la curva *precision-recall*. Existen muchos parámetros que pueden ser ajustados con el objetivo de mejorar los resultados, por lo que no pueden ser explorados de forma exhaustiva. Se pueden realizar búsquedas aleatorias u otras técnicas de optimización para ajustar estos parámetros. Técnicas más avanzadas de *finetunning* serán utilizadas en trabajos futuros. En la figura B.5 se puede ver el proceso de entrenamiento desde Tensorboard. Notar que algunas de las iteraciones de búsqueda no llegan a completar todas las épocas de entrenamiento, esto se debe al uso de *early stopping*, método que permite ahorrar tiempo de cómputo cuando un algoritmo no mejora su desempeño luego de un número de

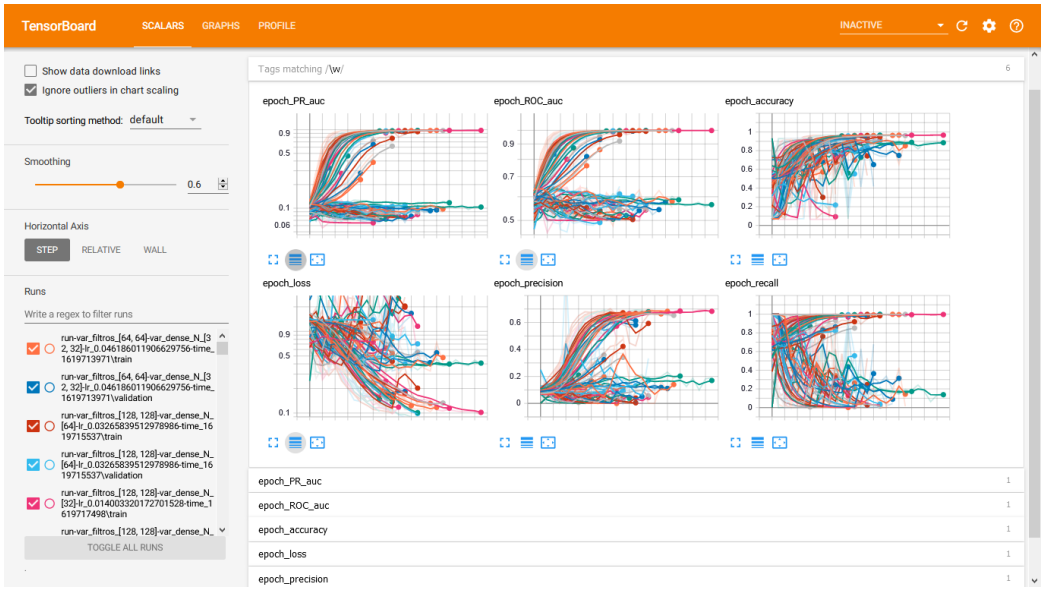


Figura B.5: Imagen del tablero de métricas de Tensorboard durante el entrenamiento de deepDAICE en cluster.uy.

Tabla B.1: Resultados experimentales del área bajo la curva *precision-recall* de los algoritmos XGB de DAICE y CNN2D de deepDAICE evaluados sobre los conjuntos de datos de validación y test.

	AUC_PR VAL	AUC_PR TEST
CNN2D	0.157	0139
XGB	0.102	0.101

épocas (paciencia).

Para evaluar la capacidad de los modelos de deepDAICE utilizaremos como línea base el algoritmo XGB (*extremme gradient boosting*), el cual obtuvo los mejores resultados en DAICE para datos mensuales. Se evalúa la misma base de datos pero en lugar de utilizar los datos de medidores inteligentes se utilizan los consumos mensuales de los últimos tres años.

En la figura B.6 se muestran las curvas de *precision-recall* de los modelos obtenidos con los algoritmos XGB y CNN2D. Se ve como deepDAICE es capaz de obtener mejores resultados con tres meses de datos que lo que logra el algoritmo de DAICE con tres años de información. Los resultados de precisión media (AUC_PR) se presentan en la tabla B.1 y muestran cómo para un conjunto de test de datos no vistos, el nuevo sistema (deepDAICE) entrenado con tres meses de datos supera al sistema DAICE entrenado con datos de tres años, en un 38 %.

Otros experimentos con más resultados experimentales de deepDAICE se presentan en el capítulo 6.

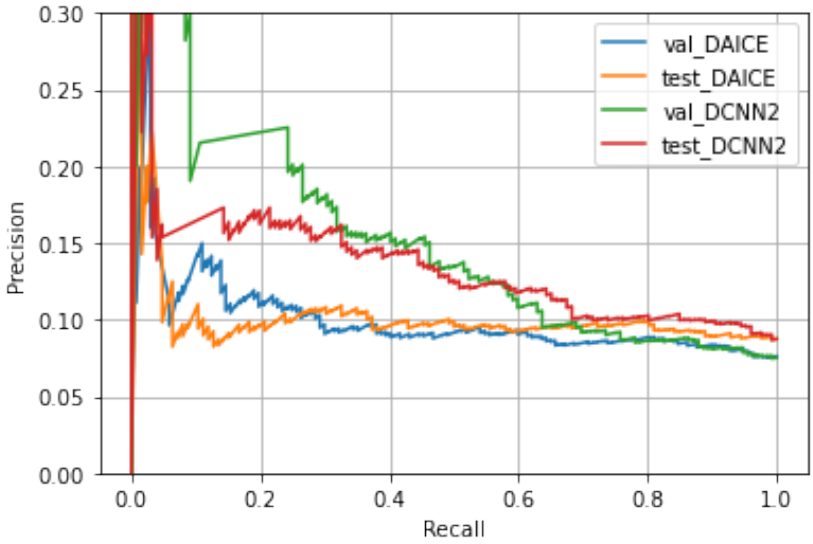


Figura B.6: Resultados de los algoritmos XGB con 36 meses y CNN2D con 90 días sobre los conjuntos de datos de validación y test

B.3. CER_NTL: Ambiente de simulación de fraudes y generación de modelos de aprendizaje profundo

B.3.1. Repositorio NTL_SmartMeters

El repositorio inicialmente creado para realizar una capacitación del equipo de trabajo del DRE de UTE está a la fecha disponible en gitlab de Facultad de Ingeniería (https://gitlab.fing.edu.uy/ute-fing/NTL_SmartMeters).

En la carpeta *tools* se encuentran dos archivos de Python que pueden ser utilizados como librerías desde otros programas. A modo de ejemplo, en la carpeta *Notebooks* se encuentran varios ejemplos de uso de estas herramientas.

La librería *NTLSM_lib_data* contiene dos funciones que permiten crear diferentes tipos de fraudes como los que se utilizan en el capítulo 6. La función *fraudet* permite generar fraude con periodicidad en su ejecución y en ventanas de tiempo, mientras que la función *fraudepc* permite generar fraude aleatorio proporcional al consumo. La función *downres* permite modificar la resolución temporal de los medidores inteligentes para evaluar su impacto en el desempeño de los algoritmos de detección automática de fraude. La estructura de carpetas de repositorio se puede ver en la imagen B.7.

B.3.2. Algoritmos de detección usados en NTL_SmartMeters

Un posible enfoque para detección de fraude en este tipo de datos es generar una imagen a partir de la serie temporal de consumos y entrenar un algoritmo de clasificación con redes convolucionales. Si bien en [109] se utiliza una estructura

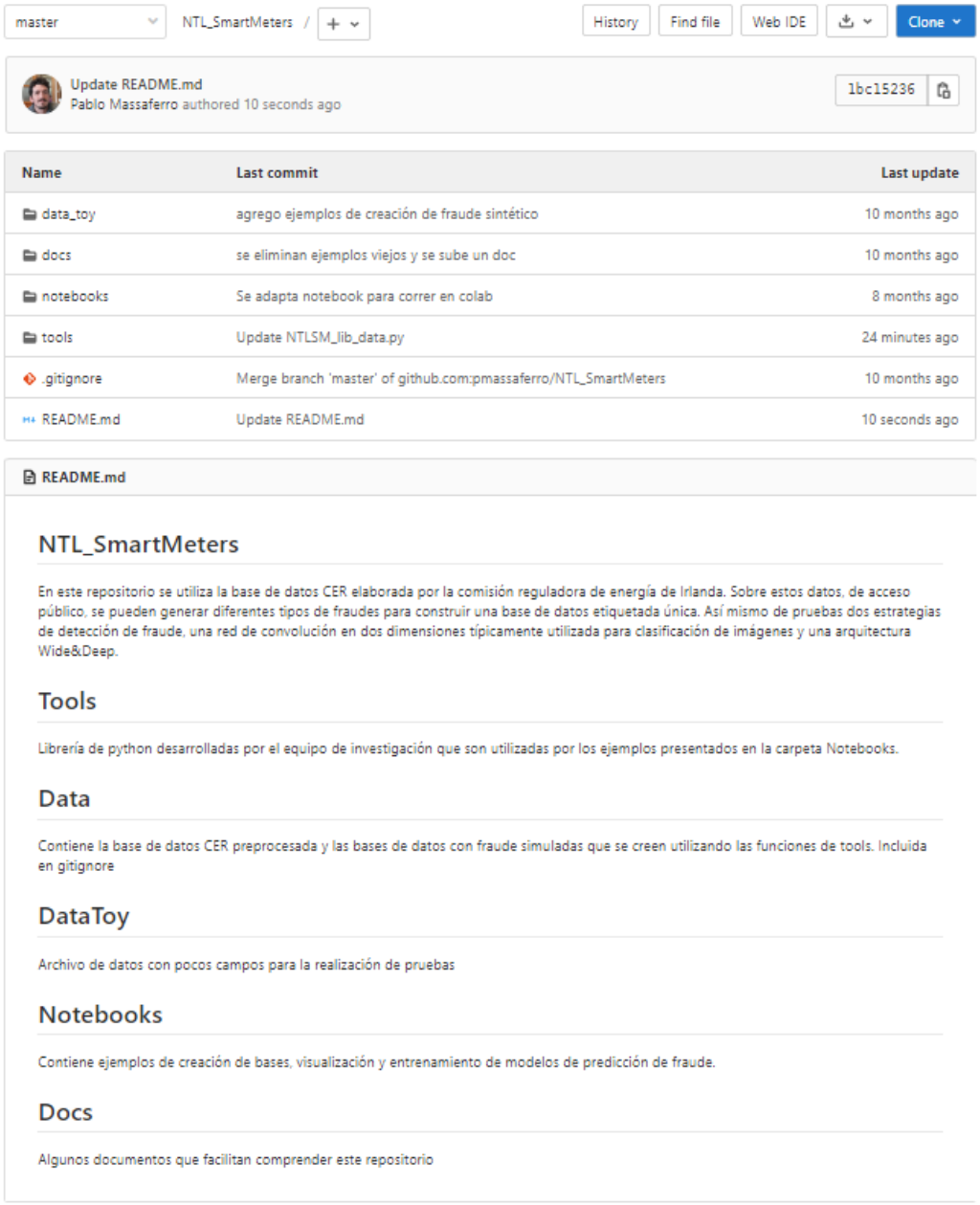


Figura B.7: Repositorio gitlab.fing.edu.uy/ute-fing/NTL_SmartMeters

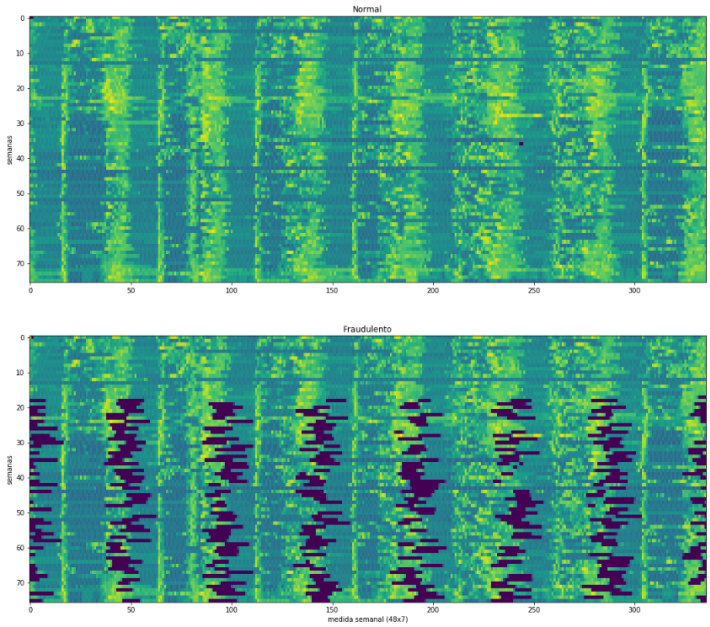


Figura B.8: Representación matricial de consumo de energía con y sin fraude. El fraude simulado es del tipo total periódico

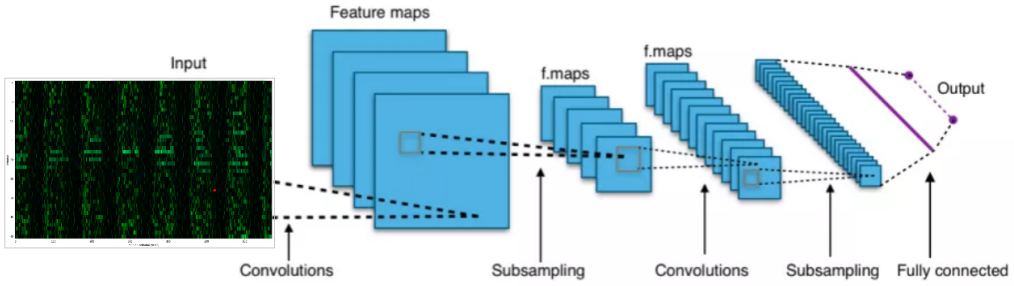


Figura B.9: Esquema de arquitectura utilizada para redes convolucionles en dos dimensiones (CNN2D).

matricial de los datos de entrada, al tratarse de consumos diarios la matriz ordenada con consumos semanales solo tiene siete valores por fila. En este caso, se trabajó con consumos cada 30 minutos durante 72 semanas. Esto permitió generar una imagen de 336x72 como se muestra en la figura B.8.

En este enfoque cada cliente analizado representa una imagen. El conjunto de datos de entrenamiento es utilizado para ajustar los valores de los filtros utilizados en cada capa del algoritmo. Esto se muestra de forma esquemática en la figura B.9.

El segundo algoritmo incluido en el repositorio es *wide&deep* [109], cuya arquitectura se presenta de forma esquemática en la figura B.10

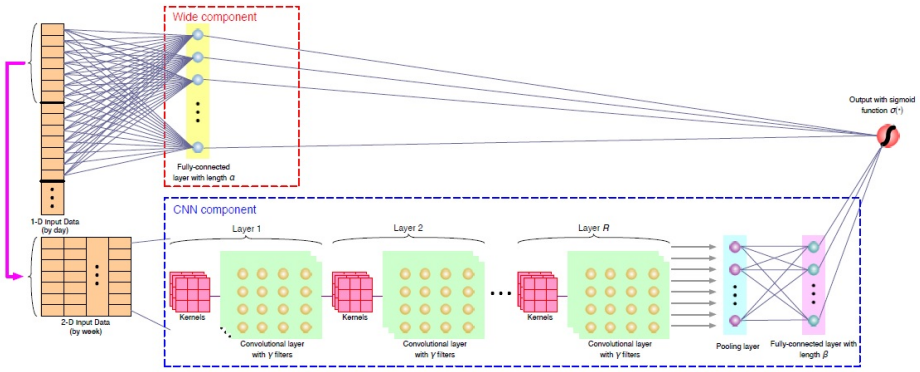


Figura B.10: red *deep&wide* - Imagen tomada del artículo citado en [109]

B.3.3. Resultados sobre datos de la base CER

Los algoritmos presentados en esta sección, e incluidos en el repositorio transferido a UTE, permiten obtener buenos resultados de detección de fraude. Para esto se generó una base de datos con fraudes utilizando las librerías desarrolladas. Esta base incluye un 14 % de irregularidades, este valor es similar a los encontrados en bases de datos de UTE utilizadas en DAICE y al valor reportado en la base SGCC (única base de datos pública con fraudes reales).

En la figura B.11 se presentan las métricas de desempeño más relevantes del problema durante el entrenamiento del algoritmo CNN2D. Una vez que los hiperparámetros del sistema son ajustados, los resultados de precisión promedio (área bajo la curva de *precision-recall*) alcanzan valores del orden de 50 % en detección global de fraudes. Este resultado aumenta en un factor mayor a tres la detección aleatoria. Incluso existen umbrales de decisión en los cuales se puede alcanzar desempeños con precisiones superiores, detectando menos fraudes (ver curvas PR en figura B.12).

En este tipo de problemas donde el desbalance de clases es importante, una de las métricas más relevantes es el área debajo de la curva PR. En la figura B.12 se presenta la curva PR obtenida para cada uno de los algoritmos presentados en esta sección. Viendo las curvas PR se puede notar que el desempeño entre ambos algoritmos es muy similar y, por otro lado, que existen puntos de operación con precisiones mayores al 50 % para la base CER con fraude simulado al 14 %.

Como se muestra en la tabla B.2 la diferencia de resultados entre ambos algoritmos no es estadísticamente significativa. Como ya hemos visto en otros experimentos, los mayores cambios de desempeño, luego de que se optimiza un algoritmo, están dados por la inclusión de nueva información.

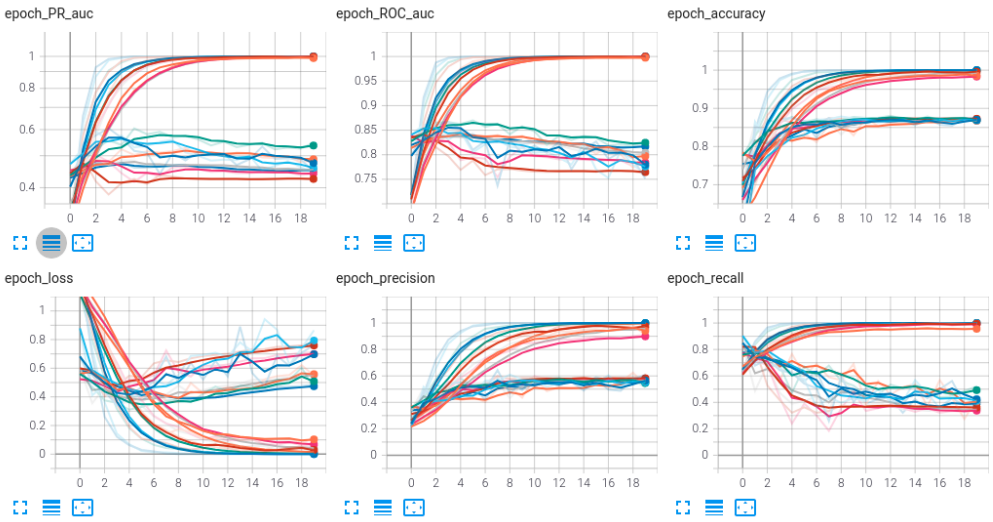


Figura B.11: Visualización de métricas en Tensorboard durante el proceso de entrenamiento del algoritmo CNN2D. Incluye resultados sobre datos de entrenamiento y validación para las métricas AUC_ROC, AUC_PR (área debajo de la curva PR), exactitud, *precision*, *recall* y la medida de la función de error de entrenamiento (*loss*)

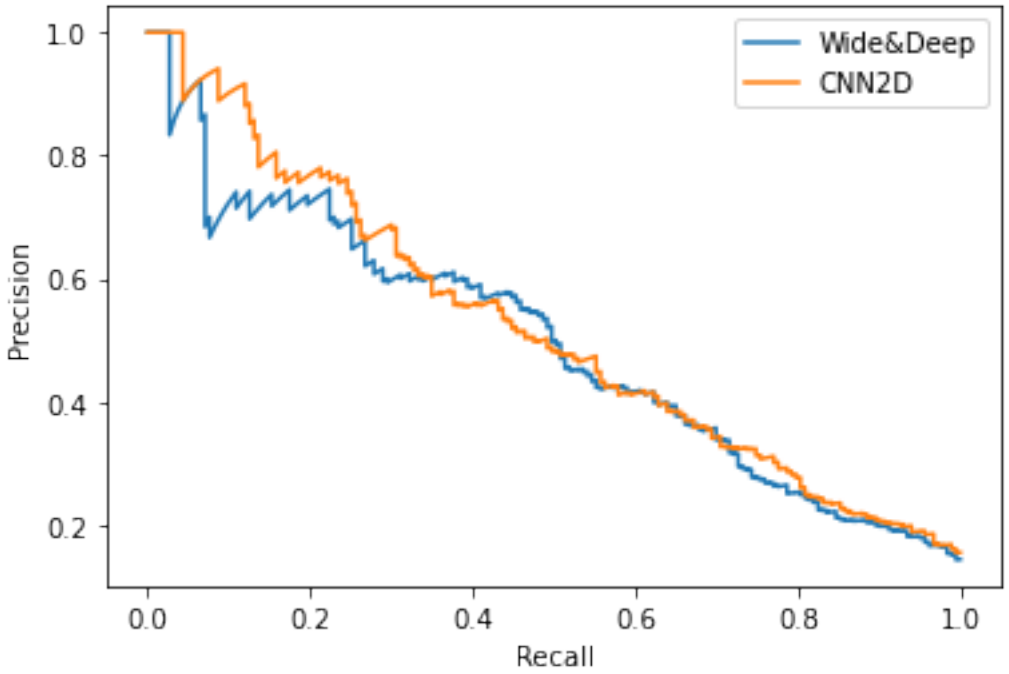


Figura B.12: Resultado de clasificación sobre un subconjunto de test de la base CER con 14 % de fraude sintético.

Algoritmo	AUC_PR	AUC_ROC	Precision	Recall	Accuracy
CNN2D	0.520	0.853	0.430	0.739	0.824
Wide&Deep	0.515	0.851	0.425	0.726	0.822

Tabla B.2: Resultados experimentales de evaluación de los algoritmos CNN2D y *wide&deep* en detección de fraudes simulados sobre consumos reales de la base de datos CER

Referencias

[1] Abdulaziz Aldegheishem, Mubbashra Anwar, Nadeem Javaid, Nabil Alrajeh, Muhammad Shafiq, and Hasan Ahmed. Towards sustainable energy efficiency with intelligent electricity theft detection in smart grids emphasising enhanced neural networks. *IEEE Access*, 9:25036–25061, 2021.

[2] Meshal Alharbi, Saud Alghumayjan, Mansour Alsaleh, Devavrat Shah, and Ahmad Alabdulkareem. Electricity non-technical loss detection: Enhanced cost-driven approach utilizing synthetic control. In *2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2021.

[3] Nelson Fabian Avila, Gerardo Figueroa, and Chia-Chi Chu. Ntl detection in electric distribution systems using the maximal overlap discrete wavelet-packet transform and random undersampling boosting. *IEEE Transactions on Power Systems*, 33(6):7171–7180, 2018.

[4] Alejandro Correa Bahnsen, Djamila Aouada, and Bjorn Ottersten. Ensemble of example-dependent cost-sensitive decision trees. *arXiv preprint arXiv:1505.04637*, 2015.

[5] Alejandro Correa Bahnsen, Djamila Aouada, and Björn Ottersten. Example-dependent cost-sensitive decision trees. *Expert Systems with Applications*, 42(19):6609–6619, 2015.

[6] Rajendra Rana Bhat, Rodrigo Daniel Trevizan, Rahul Sengupta, Xiaolin Li, and Arturo Bretas. Identifying nontechnical power loss via spatial and temporal deep learning. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 272–279. IEEE, 2016.

[7] L Breiman. Random forests machine learning. 45: 5–32. *View Article PubMed/NCBI Google Scholar*, 2001.

[8] Łukasz Brocki and Danijel Koržinek. Kohonen self-organizing map for the traveling salesperson problem. In *Recent Advances in Mechatronics*, pages 116–119. Springer, 2007.

[9] Charles Brush. Magneto electric machine, U.S. Patent 189 997, Apr. 1877.

[10] Madalina Mihaela Buzau, Javier Tejedor-Aguilera, Pedro Cruz-Romero, and Antonio Gómez-Expósito. Detection of non-technical losses using smart meter data and supervised learning. *IEEE Transactions on Smart Grid*, 10(3):2661–2670, 2018.

[11] Madalina-Mihaela Buzau, Javier Tejedor-Aguilera, Pedro Cruz-Romero, and Antonio Gómez-Expósito. Hybrid deep neural networks for detection of non-technical losses in electricity smart meters. *IEEE Transactions on Power Systems*, 35(2):1254–1263, 2019.

[12] Alejandro C. Bahnsen. Costsensitiveclassification. `https://github.com/albahnsen/CostSensitiveClassification`, 2020.

[13] Tatiana Castillo, Fabio García, Luis Mosquera, Targelia Rivadeneira, Andrés Schuschny, Katherine Segura, and Marco Yujato. Panorama energético de américa latina y el caribe. *Organización Latinoamericana de Energía*, 2020.

[14] Commission for Energy Regulation (CER). CER Smart Metering Project - Electricity Customer Behaviour Trial, 2009-2010 [dataset]., 2012. 1st Edition. Irish Social Science Data Archive. SN:0012-00. `https://www.ucd.ie/issda/data/commissionforenergyregulationcer/`.

[15] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[16] Bernat Coma-Puig, Josep Carmona, Ricard Gavalda, Santiago Alcoverro, and Victor Martin. Fraud detection in energy consumption: A supervised approach. In *2016 IEEE international conference on data science and advanced analytics (DSAA)*, pages 120–129. IEEE, 2016.

[17] 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.

[18] Xueyuan Cui, Shengyuan Liu, ZZ Lin, Jien Ma, Fushuan Wen, Yi Ding, Li Yang, Wenchong Guo, and Xiaofeng Feng. Two-step electricity theft detection strategy considering economic return based on convolutional autoencoder and improved regression algorithm. *IEEE Transactions on Power Systems*, 2021.

[19] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[20] Matheus Alberto de Souza, José LR Pereira, Guilherme de O Alves, Bráulio C de Oliveira, Igor D Melo, and Paulo AN Garcia. Detection and identification of energy theft in advanced metering infrastructures. *Electric Power Systems Research*, 182:106258, 2020.

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

[22] Soma Shekara Sreenadh Reddy Depuru, Lingfeng Wang, Vijay Devabhaktuni, and Robert C Green. High performance computing for detection of electricity theft. *International Journal of Electrical Power & Energy Systems*, 47:21–30, 2013.

[23] M Di Martino, G Hernández, M Fiori, and A Fernández. A new framework for optimal classifier design. *Pattern Recognition*, 46(8):2249–2255, 2013.

[24] Matias Di Martino, Federico Decia, Juan Molinelli, and Alicia Fernández. Improving electric fraud detection using class imbalance strategies. In *International Conference on Pattern Recognition and Methods, 1st. ICPRAM.*, pages 135–141, 2012.

[25] Matías Di Martino, Federico Decia, Juan Molinelli, and Alicia Fernández. A novel framework for nontechnical losses detection in electricity companies. In *Pattern Recognition-Applications and Methods*, pages 109–120. Springer, 2013.

[26] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* Wiley, New York, 2. edition, 2001.

[27] Lucas Teles Faria, Joel David Melo, and Antonio Padilha-Feltrin. Spatial-temporal estimation for nontechnical losses. *IEEE Transactions on Power Delivery*, 31(1):362–369, 2016.

[28] Dominique Feillet, Pierre Dejax, and Michel Gendreau. Traveling salesman problems with profits. *Transportation science*, 39(2):188–205, 2005.

[29] Vitaly Ford, Ambareen Siraj, and William Eberle. Smart grid energy fraud detection using artificial neural networks. In *2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG)*, pages 1–6. IEEE, 2014.

[30] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[31] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[32] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* O'Reilly Media, 2019.

[33] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[34] Patrick Glauner, Andre Boechat, Lautaro Dolberg, Radu State, Franck Bettinger, Yves Rangoni, and Diogo Duarte. Large-scale detection of non-technical losses in imbalanced data sets. In *Innovative Smart Grid Technologies Conference (ISGT), 2016 IEEE Power & Energy Society*, pages 1–5. IEEE, 2016.

[35] Patrick Glauner, Jorge Meira, Lautaro Dolberg, Radu State, Franck Bettinger, Yves Rangoni, and Diogo Duarte. Neighborhood features help detecting electricity theft in big data sets. In *Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies.* IEEE, 2016.

[36] Patrick Glauner, Jorge Meira, Petko Valtchev, Radu State, and Franck Bettinger. The challenge of non-technical loss detection using artificial intelligence: A survey. *International Journal of Computational Intelligence Systems 10.1 (2017): 760-775.*, 2017.

[37] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[38] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[39] Juan Ignacio Guerrero, Inigo Monedero, Felix Biscarri, Jesus Biscarri, Rocio Millan, and Carlos Leon. Non-technical losses reduction by improving the inspections accuracy in a power utility. *IEEE Transactions on Power Systems*, 33(2):1209–1218, 2017.

[40] Aldy Gunawan, Hoong Chuin Lau, and Pieter Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332, 2016.

[41] Sravan Kumar Gunturi and Dipu Sarkar. Ensemble machine learning models for the detection of energy theft. *Electric Power Systems Research*, 192:106904, 2021.

[42] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*, 2017.

[43] Yonghe Guo, Chee-Wooi Ten, and Panida Jirutitijaroen. Online data validation for distribution operations against cybertampering. *IEEE Transactions on Power Systems*, 29(2):550–560, 2013.

[44] 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.

[45] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[46] Tianyu Hu, Qinglai Guo, Xinwei Shen, Hongbin Sun, Rongli Wu, and Haoning Xi. Utilizing unlabeled data to detect electricity fraud in ami: A semi-supervised deep learning approach. *IEEE transactions on neural networks and learning systems*, 2019.

[47] 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.

[48] Wenjie Hu, Yang Yang, Jianbo Wang, Xuanwen Huang, and Ziqiang Cheng. Understanding electricity-theft behavior via multi-source data. In *Proceedings of The Web Conference 2020*, pages 2264–2274, 2020.

[49] Rong Jiang, Rongxing Lu, Ye Wang, Jun Luo, Changxiang Shen, and Xuemin Shen. Energy-theft detection issues for advanced metering infrastructure in smart grid. *Tsinghua Science and Technology*, 19(2):105–120, 2014.

[50] Rong Jiang, Harry Tagaris, Andrei Lachsz, and Mark Jeffrey. Wavelet based feature extraction and multiple classifiers for electricity fraud detection. In *IEEE/PES Transmission and Distribution Conference and Exhibition*, volume 3, pages 2251–2256. IEEE, 2002.

[51] Raúl Jiménez, Tomás Serebrisky, and Jorge Mercado. Dimensionando las pérdidas de electricidad en los sistemas de transmisión y distribución en américa latina y el caribe. *Banco Interamericano de Desarrollo. División de Energía*, 2014.

[52] Anish Jindal, Amit Dua, Kuljeet Kaur, Mukesh Singh, Neeraj Kumar, and Sukumar Mishra. Decision tree and svm-based data analytics for theft detection in smart grid. *IEEE Transactions on Industrial Informatics*, 12(3):1005–1016, 2016.

[53] 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.

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

[55] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[56] 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 *Innovative Smart Grid Technologies Latin America (ISGT LATAM), 2015 IEEE PES*, pages 887–892. IEEE, 2015.

[57] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[58] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[59] Carlos León, Félix Biscarri, Iñigo Monedero, Juan Ignacio Guerrero, Jesús Biscarri, and Rocío Millán. Variability and trend-based generalized rule induction model to ntl detection in power companies. *IEEE Transactions on Power Systems*, 26(4):1798–1807, 2011.

[60] 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.

[61] Shih-Wei Lin. Solving the team orienteering problem using effective multi-start simulated annealing. *Applied Soft Computing*, 13(2):1064–1073, 2013.

[62] Northeast Group LLC. Electricity theft and non-technical losses: Global markets, solutions, and vendors, 2017.

[63] 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.

[64] Pablo Massaferro, J Matías Di Martino, and Alicia Fernandez. Ntl detection: Optimization of inspection routes weighing mobility cost and detection likelihood. In *2021 IEEE PES Innovative Smart Grid Technologies Conference-Latin America (ISGT Latin America)*, pages 1–5. IEEE, 2021.

[65] Pablo Massaferro, J Matías Di Martino, and Alicia Fernández. Ntl detection: Overview of classic and dnn-based approaches on a labeled dataset of 311k customers. In *2021 IEEE NA Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2021.

[66] Pablo Massaferro, J Matías Di Martino, and Alicia Fernández. Fraud detection on power grids while transitioning to smart meters by leveraging multi-resolution consumption data. *IEEE Transactions on Smart Grids (accepted)*, 2022.

[67] Pablo Massaferro Saquieres, Henry Marichal, Matías Di Martino, Fernando Santomauro, Juan Pablo Kosut, and Alicia Fernández. Improving electricity non technical losses detection including neighborhood information. In *2018 IEEE PES General Meeting (GM) - IEEE Power and Energy Society, Portland, Oregon, USA, 5-9 aug*, pages 1–5. IEEE, 2018.

[68] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[69] George M Messinis and Nikos D Hatziargyriou. Review of non-technical loss detection methods. *Electric Power Systems Research*, 158:250–266, 2018.

[70] Iñigo Monedero, Félix Biscarri, Carlos León, Juan I Guerrero, Jesús Biscarri, and Rocío Millán. Detection of frauds and other non-technical losses in a power utility using pearson coefficient, bayesian networks and decision trees. *International Journal of Electrical Power & Energy Systems*, 34(1):90–98, 2012.

[71] Inigo Monedero, Felix Biscarri, Carlos Leon, JuanI. Guerrero, Jesus Biscarri, and Rocio Millan. Using regression analysis to identify patterns of non-technical losses on power utilities. In Rossitza Setchi, Ivan Jordanov, RobertJ. Howlett, and LakhmiC. Jain, editors, *Knowledge-Based and Intelligent Information and Engineering Systems*, volume 6276 of *Lecture Notes in Computer Science*, pages 410–419. Springer Berlin Heidelberg, 2010.

[72] Mahmoud Nabil, Muhammad Ismail, Mohamed Mahmoud, Mostafa Shahin, Khalid Qaraqe, and Erchin Serpedin. Deep learning-based detection of electricity theft cyber-attacks in smart grid ami networks. In *Deep Learning Applications for Cyber Security*, pages 73–102. Springer, 2019.

[73] J Nagi, AM Mohammad, Keem Siah Yap, Sieh Kiong Tiong, and Syed Khaleel Ahmed. Non-technical loss analysis for detection of electricity theft using support vector machines. In *2008 IEEE 2nd International Power and Energy Conference*, pages 907–912. IEEE, 2008.

[74] 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.

[75] 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, 2010.

[76] 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.

[77] Nasser M Nasrabadi. Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901, 2007.

[78] International Energy Agency OECD. Electric power transmission and distribution losses, 2021.

[79] Council of European Energy Regulators. 2nd ceer report on power losses, 2020.

[80] Joao P Papa, Alexandre X Falcao, and Celso TN Suzuki. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, 19(2):120–131, 2009.

[81] Luis AM Pereira, Luis CS Afonso, João P Papa, Zita A Vale, Caio CO Ramos, Danillo S Gastaldello, and André N Souza. Multilayer perceptron neural networks training through charged system search and its application for non-technical losses detection. In *2013 IEEE PES Conference on Innovative Smart Grid Technologies (ISGT Latin America)*, pages 1–6. IEEE, 2013.

[82] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[83] Marcus Poggi, Henrique Viana, and Eduardo Uchoa. The team orienteering problem: Formulations and branch-cut and price. In *10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'10)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2010.

[84] Rajiv Punmiya and Sangho Choe. Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. *IEEE Transactions on Smart Grid*, 10(2):2326–2329, 2019.

[85] Caio César Oba Ramos, Andrá Nunes de Sousa, Joao Paulo Papa, and Alexandre Xavier Falcao. A new approach for nontechnical losses detection based on optimum-path forest. *IEEE Transactions on Power Systems*, 26(1):181–189, 2010.

[86] Caio César Oba Ramos, André Nunes de Souza, Alexandre Xavier Falcao, and João Paulo Papa. New insights on nontechnical losses characterization through evolutionary-based feature selection. *IEEE Transactions on Power Delivery*, 27(1):140–146, 2011.

[87] Caio César Oba Ramos, Andre Nunes De Souza, Danilo Sinkiti Gastaldello, and João Paulo Papa. Identification and feature selection of non-technical losses for industrial consumers using the software weka. *Industry Applications (INDUSCON), 2012 10th IEEE/IAS International Conference on*, pages 1–6, 2012.

[88] Caio CO Ramos, Douglas Rodrigues, André N de Souza, and Joao P Papa. On the study of commercial losses in brazil: a binary black hole algorithm for theft characterization. *IEEE Transactions on Smart Grid*, 9(2):676–683, 2016.

[89] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[90] Muhammad Salman Saeed, Mohd Wazir Mustafa, Nawaf N Hamadneh, Nawa A Alshammari, Usman Ullah Sheikh, Touqeer Ahmed Jumani, Saifulnizam Bin Abd Khalid, and Ilyas Khan. Detection of non-technical losses in power utilities—a comprehensive systematic review. *Energies*, 13(18):4727, 2020.

[91] Muhammad Salman Saeed, Mohd Wazir Mustafa, Usman Ullah Sheikh, Touqeer Ahmed Jumani, and Nayyar Hussain Mirjat. Ensemble bagged tree based classification for reducing non-technical losses in multan electric power company of pakistan. *Electronics*, 8(8):860, 2019.

[92] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

[93] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1):185–197, 2009.

[94] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[95] Josif V Spirić, Slobodan S Stanković, Miroslav B Dočić, and Tatjana D Popović. Using the rough set theory to detect fraud committed by electricity customers. *International Journal of Electrical Power & Energy Systems*, 62:727–734, 2014.

[96] Pablo Thomasset. Historia del alumbrado público del uruguay. *Ingeniería*, 1(87):25–35, 2020.

[97] Paolo Toth and Daniele Vigo. *Vehicle routing: problems, methods, and applications*. SIAM, 2014.

[98] UTE. Memoria anual 2019 ute. *Administración Nacional de Usinas y Trasmisiones Eléctricas*, 2020.

[99] Pieter Vansteenwegen, Wouter Souffriau, and Dirk Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10, 2011.

[100] Diego Vicen. Solving the traveling salesman problem using self-organizing maps. https://github.com/DiegoVicen/som-tsp, 2018.

[101] Joaquim L Viegas, Paulo R Esteves, R Melício, VMF Mendes, and Susana M Vieira. Solutions for detection of non-technical losses in the electricity grid: A review. *Renewable and Sustainable Energy Reviews*, 80:1256–1268, 2017.

[102] Joaquim L Viegas, Paulo R Esteves, and Susana M Vieira. Clustering-based novelty detection for identification of non-technical losses. *International Journal of Electrical Power & Energy Systems*, 101:301–310, 2018.

[103] Fernando Gustavo Viera Nuñez. Modelado y detección de fraudes en redes inteligentes de distribución de energía eléctrica. Master's thesis, Udelar.FI, IIE, 2020.

[104] Yi Wang, Qixin Chen, Tao Hong, and Chongqing Kang. Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Transactions on Smart Grid*, 2018.

[105] Donghuan Yao, Mi Wen, Xiaohui Liang, Zipeng Fu, Kai Zhang, and Baojia Yang. Energy theft detection with energy privacy preservation in the smart grid. *IEEE Internet of Things Journal*, 6(5):7659–7669, 2019.

[106] Keem Siah Yap, Sieh Kiong Tiong, Jawad Nagi, Johnny S. P. Koh, and Farrukh Nagi. Comparison of supervised learning techniques for non-technical loss detection in power utility. *International Review on Computers and Software (I.RE.CO.S.)*, 7(2):1828–6003, 2012.

[107] 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.

[108] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699. ACM, 2002.

[109] 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, 2018.

Índice de tablas

Índice de figuras

Esta es la última página.
Compilado el sábado 9 abril, 2022.
http://iie.fing.edu.uy/