Facultad 2

Detección de Fraude Bancario mediante técnicas de Aprendizaje Profundo

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: Franklin Pacheco Domínguez

Tutores: Dr.C. Héctor Raúl González Diez, P.T.

Ing. Vladimir Milián Núñez, P.A.

La Habana, noviembre de 2023

"Año 65 de la Revolución"

Agradecimientos

frase

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título "*Detección de Fraude Bancario mediante técnicas de Aprendizaje Profundo*", concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firman la presente a los <día> días del mes de <mes> del año <año>.

Franklin Pacheco Domínguez

_____
Firma del Autor

Dr.C. Héctor Raúl González Diez

_____
Firma del Tutor

Ing. Vladimir Milián Núñez

_____
Firma del Tutor

Resumen

En este trabajo de diploma se abordan técnicas de Deep Learning para detectar transacciones bancarias fraudulentas. Uno de los problemas actuales de las instituciones bancarias es la gran cantidad de operaciones a procesar y el reducido tiempo que se tiene para realizar estas operaciones. El objetivo del presente trabajo consiste en el desarrollo de modelos orientados a la detección de anomalías en transacciones bancarias fraudulentas. Para ello se realiza una fundamentación teórica previa de los principales conceptos y algoritmos relacionados con el campo de acción, así como las métricas, herramientas y tecnologías que se usan para su estudio y aplicación. Se selecciona como metodología KDD (Knowledge Discovery in Databases) para guiar el ciclo de vida del proyecto. Se obtienen los modelos que mejor resultado ofrecen en la detección de fraude sobre el set de datos propuesto.

Palabras clave: fraude bancario, anomalía, aprendizaje profundo, transacciones bancarias, autoencoder, modelo, red neuronal artificial.

Abstract

This diploma work addresses Deep Learning techniques to detect fraudulent banking transactions. One of the current problems of banking institutions is the large number of operations to process and the short time available to carry out these operations. The objective of this work is the development of models aimed at detecting anomalies in fraudulent banking transactions. To do this, a prior theoretical foundation is carried out on the main concepts and algorithms related to the field of action, as well as the metrics, tools and technologies used for their study and application. KDD (Knowledge Discovery in Databases) is selected as the methodology to guide the project life cycle. The models that offer the best results in detecting fraud on the proposed data set are obtained.

Keywords: bank fraud, anomaly, deep learning, banking transactions, autoencoder, model, artificial neural network.

CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

El fraude bancario es un problema persistente que representa una amenaza para la estabilidad financiera y la confianza de los clientes en el sistema bancario. La detección temprana y eficaz de las actividades fraudulentas se ha convertido en una prioridad para las instituciones financieras en todo el mundo. Puede causar pérdidas económicas tanto a los clientes como a los bancos, lo que afecta negativamente a su rentabilidad y solvencia. El fraude bancario puede dañar la reputación de los bancos y erosionar la confianza de los clientes, lo que puede provocar una menor participación en el sistema financiero y una menor demanda de productos y servicios bancarios. Según una encuesta de PwC, el 49% de los clientes globales dijeron que perderían la confianza en su banco si fueran víctimas de fraude, y el 30% cambiarían de banco (PricewaterhouseCoopers, s. f.).

En la actualidad, con el uso y normalización de las tecnologías se está viviendo un cambio en el paradigma social, en el cual los datos masivos cobran vital importancia. Si se tienen en consideración los grandes volúmenes de datos generados sólo en los pasados años y los avances adquiridos en la esfera científica, es de suma relevancia notar que dichos datos sobrepasan claramente la capacidad de comprensión, almacenamiento y recolección de los mismos sin el uso de herramientas adecuadas. Limitando así las capacidades de detección del fraude en instituciones con fines de crédito.

Con la finalidad de solucionar dicho dilema, siendo la detección de anomalías una técnica de Minería de Datos (MD) con un vasto conjunto de aplicaciones de análisis de datos y transacciones bancarias (TB), se propone hacer uso de la misma como vía para extraer conocimiento a partir de los datos existentes. En este contexto, se puede adquirir dicho conocimiento a través de la búsqueda de patrones estadísticamente confiables, ideas o conceptos derivados de los datos originales.

Las instituciones bancarias son consideradas un eslabón esencial para la economía de un país, así como para la población que recibe sus servicios. En general, sus actividades de financiamiento conllevan varios tipos de préstamos, tales como: para viviendas, proyectos, financiamiento a corto plazo, pequeñas y medianas empresas, comercio y de otros tipos. También es posible que solo se concentren en transacciones específicas con clientes que cumplan ciertos requisitos y con ciertos sectores industriales. En un sentido amplio, se considera a estas instituciones el objetivo principal de aquellos individuos que son promotores del fraude bancario.

El aumento del fraude ha sido el resultado de la expansión de la banca electrónica y de varios entornos de pago en línea resultando en pérdidas anuales de miles de millones de dólares. En esta era de

pagos digitales, la detección de fraude bancario se ha convertido en uno de los objetivos más importantes. No se puede negar que el futuro se dirige hacia una cultura sin efectivo y nuestro país es

parte de este futuro, dicha afirmación se evidencia en el artículo “Retomando la bancarización” del cual citamos un fragmento: “El mundo avanzó a la bancarización con fuertes inversiones en equipamiento tecnológico y en terminales de puntos de venta. En Cuba no hemos tenido esa celeridad, porque se nos obstaculiza el acceso a nuevas tecnologías. Sin embargo, a partir de la innovación, con los recursos disponibles, los compañeros de Etecsa, Enzona y el sistema bancario nos hemos vinculado y tenemos posibilidades de seguir avanzando en la bancarización de las transacciones”, dijo Alonso Vázquez (*Retomando la bancarización*, 2023).

Como resultado, los métodos de pago típicos ya no se utilizarán en el futuro, y por lo tanto no serán útiles para expandir un negocio. Los clientes no siempre visitarán el negocio con dinero en efectivo en sus bolsillos. Ahora están otorgando una prima a pagos con tarjeta de débito y crédito. Como resultado, las empresas necesitan actualizar su entorno para garantizar que puedan aceptar todo tipo de pagos. En los próximos años, se espera un gran incremento del fraude bancario ya que todas las operaciones financieras se estiman serán realizadas mediante transferencias bancarias.

Como resultado, las instituciones bancarias deberían priorizar equiparse con un sistema automatizado de detección de fraude para impedir que el número de víctimas de fraude siga aumentando de forma constante.

A medida que la tecnología avanza, los delincuentes también han evolucionado y con ellos sus métodos para cometer fraude, en la actualidad hay un mayor número de posibles víctimas dado al aumento de operaciones financieras y en el futuro con la eliminación del efectivo en nuestro país es muy probable que estos ciberdelincuentes fijen su mirada en algunos sectores específicos que son los más vulnerables de la sociedad cubana, como es el caso del sector de la tercera edad o personas con discapacidad visual entre otros, estos sectores en una sociedad bancarizada son los más propensos a sufrir perdidas lo que eleva la importancia de la detección de actividades fraudulentas como mecanismo de defensa para proteger a los sectores antes mencionados. Las técnicas tradicionales de detección de fraude basadas en reglas predefinidas y estadísticas no siempre son capaces de adaptarse a las tácticas cambiantes de los estafadores. En este sentido, las redes neuronales profundas han demostrado ser una herramienta prometedora para abordar esta problemática (Alarfaj et al., 2022).

Las técnicas de detección de anomalías actuales han demostrado ser insuficientes para combatir eficazmente este problema. Aquí es donde entran en juego las redes neuronales profundas. A pesar de su potencial, la aplicación de estas redes en la detección de fraudes bancarios es un área relativamente inexplorada y presenta sus propios desafíos.

Se plantea como **Problema de investigación:** ¿Cómo pueden los modelos de redes neuronales profundas contribuir a la detección de fraude bancario?

Objeto de estudio: La detección de fraude bancario.

Campo de acción: Métodos de detección de anomalías para la identificación de fraude bancario.

Objetivo general: Desarrollar un algoritmo de detección de fraude bancario basado en modelos de redes neuronales profundas para identificar anomalías en las transacciones financieras.

Objetivos específicos:

1. Investigar y revisar la literatura existente sobre técnicas de detección de fraude bancario, con un enfoque en las aplicaciones de modelos de redes neuronales profundas.
2. Recopilar y preparar datos históricos de transacciones bancarias que sean representativos y adecuados para el entrenamiento y evaluación de modelos de detección de fraude.
3. Diseñar y desarrollar modelos de redes neuronales profundas personalizados para la detección de anomalías en las transacciones financieras, considerando la arquitectura de la red, las funciones de activación y otros hiperparámetros relevantes.
4. Evaluar la eficacia de los modelos de detección de fraude basados en redes neuronales profundas utilizando métricas como la precisión, la sensibilidad y la especificidad, comparándolos con enfoques tradicionales de detección de fraude.
5. Analizar los resultados para comprender cómo los modelos de redes neuronales profundas identifican y clasifican las anomalías en las transacciones bancarias, y determinar su capacidad para adaptarse a las tácticas cambiantes de los estafadores.
6. Proponer recomendaciones y pautas para la implementación de sistemas de detección de fraude bancario basados en redes neuronales profundas en entornos bancarios reales.

El **alcance** de esta tesis se extiende a varios aspectos clave de la detección de fraudes bancarios utilizando técnicas de aprendizaje profundo. Primero, se realizará un estudio de las técnicas de aprendizaje profundo y su aplicación en la detección de fraudes bancarios. Esto implicará el desarrollo y la implementación de modelos de aprendizaje profundo para identificar transacciones fraudulentas.

Los **resultados esperados** son desarrollar modelos de aprendizaje profundo eficaces que puedan identificar transacciones fraudulentas con precisión, identificar y discutir los desafíos y oportunidades asociados con la implementación de técnicas de aprendizaje profundo en la detección de fraudes bancarios. En última instancia, se espera que esta tesis contribuya a la comprensión de cómo las técnicas de aprendizaje profundo pueden ser utilizadas para mejorar la detección y prevención de fraudes bancarios, beneficiando tanto a las entidades financieras como a los clientes.

Impacto:

1. **Impacto Académico**: Esta tesis contribuirá al cuerpo de conocimientos en el campo de la detección de fraudes bancarios utilizando técnicas de aprendizaje profundo. Los hallazgos y conclusiones de esta investigación podrían ser utilizados como base para futuras investigaciones en este campo.
2. **Impacto Práctico**: Los modelos de aprendizaje profundo desarrollados en esta tesis podrían ser implementados por las instituciones financieras para mejorar su capacidad de detectar y prevenir el fraude bancario. Esto podría resultar en una reducción significativa de las pérdidas financieras causadas por el fraude.
3. **Impacto Social**: Al mejorar la detección y prevención del fraude bancario, esta tesis podría ayudar a fortalecer la confianza del público en el sistema financiero. Esto podría tener un impacto positivo en la economía en general.

En resumen, esta tesis tiene el potencial de tener un impacto significativo tanto en el ámbito académico como en el práctico, beneficiando a las instituciones financieras, a los clientes y a la sociedad en general.

Método teórico

Histórico-lógico determina las tendencias actuales de los sistemas de detección, de los modelos de desarrollo, las técnicas, lenguajes y herramientas a utilizar en la investigación.

Método empírico

Observación se lleva a cabo para el proceso de detección de fraude bancario para obtener información sobre las etapas, los actores, y requerimientos del sistema entre otras.

Los aportes prácticos y sociales de la investigación

Se espera desarrollar un algoritmo para la detección del fraude bancario mediante la detección de anomalías.

El documento se encuentra estructurado en: Resumen, introducción, 3 capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía y anexos.

CAPÍTULO 1: La detección del fraude en transacciones bancarias: Aborda todas las bases teóricas relacionadas para el desarrollo del algoritmo, referenciando a todas las investigaciones de las cuales se extrajo información.

Capítulo 2: Preparación para el proceso de minería de datos: Refleja todo lo relacionado con los primeros 7 pasos de la metodología que se aplicará para la minería de datos.

Capítulo 3: Etapa de interpretación/evaluación de los datos: Describe las últimas 2 fases de la metodología, las cuales abordan la evaluación y resultados comparativos de los algoritmos teniendo en cuenta las métricas establecidas, además de la definición del método informático que se desarrolló.

CAPÍTULO 1: LA DETECCIÓN DEL FRAUDE EN TRANSACCIONES BANCARIAS.

Este capítulo contiene los fundamentos para la comprensión de las relaciones entre los conceptos expuestos. esta investigación comprende la detección de fraudes dentro de instituciones bancarias, así como una presentación de modelos de Deep Learning (DL) como mejor solución al problema de detección de fraude.

1.1 Detección de Fraude Bancario

La detección de fraude bancario es un área crítica en el ámbito financiero que involucra la identificación y prevención de actividades fraudulentas con el objetivo de salvaguardar los activos financieros, la integridad del sistema bancario y la confianza de los clientes. Este desafío se ha vuelto cada vez más complejo con la evolución de las tecnologías y la sofisticación de los métodos utilizados por los delincuentes financieros. Desde la perspectiva de la ingeniería informática, la detección de fraude bancario implica la aplicación de diversas técnicas y herramientas para analizar grandes volúmenes de datos y descubrir patrones anómalos que puedan indicar actividades fraudulentas.

El fraude bancario es un delito o estafa que se realiza mediante prácticas ilegales por parte de funcionarios internos de bancos o grupos externos para obtener información y datos privados de terceros, posibilidad de usurpar identidades o directamente robar dinero de usuarios de una entidad bancaria (GraphEverywhere, 2019).

Existen varios autores que abordan sobre este tema, no obstante, para los oficios de estos estudios, en base a la bibliografía consultada, fueron seleccionados por su nivel de especificidad los de los siguientes conceptos:

Fraude electrónico: El fraude electrónico o delito informático es una actividad indebida basada en la manipulación fraudulenta de elementos informáticos y sistemas de comunicación, para obtener un beneficio no autorizado.

Dato: Un dato es la representación de una variable que puede ser cuantitativa o cualitativa que indica un valor que se le asigna a las cosas y se representa a través de una secuencia de símbolos, números o letras.

Dato en informática: Es la expresión general que describe aquellas características de la entidad sobre la que opera. Los programas y aplicaciones tienen como función el procesa-miento de datos, ya que cada lenguaje de programación tiene un conjunto da datos a partir de los cuales trabaja. Toda la información que entra y sale de un ordenador lo hace en forma de datos. Dentro de los archivos existen

datos que son paquetes más pequeños de otros datos llamados registros (reunidos por características iguales o similares)

Entidad Financiera: Una entidad financiera es una agrupación cuyo giro es ofrecer servicios financieros en el área de la banca, valores y seguros. Su oferta considera desde la intermediación, comercialización de seguros, créditos y asesoramiento, entre otros.

La detección y prevención del fraude bancario, está magnificada dentro de un espectro de características y limitaciones.

- Primeramente, se debe tener sumo cuidado de no bloquear incorrectamente tarjetas o cuentas genuinas o con el procesamiento de demasiadas transacciones legítimas.
- En segundo lugar, las instituciones financieras procesan un volumen inmenso de transacciones, de las cuales sólo un pequeño porcentaje es fraudulento (cerca del 0,1%).
- En tercer lugar, el número de transacciones que pueden ser revisadas por los investigadores de fraudes es limitado, por lo existe la necesidad de automatizar el proceso de detección.

1.1.1 Los fraudes bancarios más habituales en la actualidad

Los tipos más comunes de fraude bancario que podemos encontrar en la actualidad son violaciones a la seguridad, privacidad y obtención de datos bancarios de terceros por medio de software malicioso o estrategias complejas de defraudación. Existen diversos tipos de fraudes electrónicos en el sector bancario que debemos conocer para ser perjudicados(GraphEverywhere, 2019).

Entre ellas podemos encontrar el **Phishing**. Este es un tipo de fraude que ocurre a través de plataformas en línea en las que un tercero viola nuestros patrones de seguridad para obtener información valiosa como claves de usuarios bancarios, cuentas bancarias, número de tarjetas de crédito y códigos de autorización para utilizar nuestras herramientas financieras para su beneficio (GraphEverywhere, 2019).

También podemos encontrar el **Pharming**. Este modelo más sofisticado consta en la replicación de las estructuras digitales de las entidades bancarias. En este tipo de delitos los usuarios son engañados e ingresan a plataformas que parecen ser las plataformas oficiales de los bancos. Esta usurpación les permite a los delincuentes obtener nuestros datos y tomar control total sobre nuestras finanzas personales (GraphEverywhere, 2019).

Existen adicionalmente softwares especialmente diseñados para obtener nuestros datos y contraseñas. Estos son conocidos como **Key Logger**. Este tipo de programas puede captar toda la información que

tecleamos en nuestros dispositivos. Los ordenadores suelen infectarse con este tipo de software malicioso mediante programas que se ejecutan en segundo plano. Adicionalmente a esto existen diversas operaciones de fraude electrónico que se produce en el mundo del Ecommerce que vale la pena conocer (GraphEverywhere, 2019).

1.1.2 Enfoques de Detección de Fraude Bancario desde la Ingeniería Informática:

1. **Minería de Datos y Análisis Estadístico:** La aplicación de técnicas de minería de datos y análisis estadístico permite identificar patrones y comportamientos inusuales en los datos financieros. Los modelos pueden aprender de transacciones históricas para reconocer anomalías en tiempo real.
2. **Aprendizaje Automático y Aprendizaje Profundo:** Los algoritmos de aprendizaje automático, incluidas las redes neuronales y otras técnicas de aprendizaje profundo, son cada vez más esenciales en la detección de fraude. Estos modelos pueden detectar patrones complejos y adaptarse a cambios en las tácticas de fraude.
3. **Análisis de Comportamiento del Usuario:** El seguimiento del comportamiento del usuario, especialmente en plataformas en línea, puede ayudar a identificar actividades sospechosas. Esto incluye el análisis de patrones de navegación, ubicaciones inusuales de inicio de sesión y cambios en el comportamiento de transacciones.
4. **Integración de Tecnologías Emergentes:** La incorporación de tecnologías emergentes como blockchain y biometría fortalece las medidas de seguridad y dificulta la manipulación de datos o identidades.

1.1.4 Desafíos en la Detección de Fraude Bancario:

1. **Sofisticación de los Métodos de Fraude:** Los perpetradores de fraudes bancarios han evolucionado y adoptado métodos cada vez más sofisticados, como el phishing, el malware bancario y otras técnicas avanzadas. La detección eficiente debe adaptarse a estos métodos cambiantes.
2. **Gran Volumen de Datos:** Las instituciones financieras generan y procesan enormes cantidades de datos diariamente. Manejar esta gran cantidad de información de manera eficiente y efectiva es esencial para identificar patrones indicativos de actividades fraudulentas.

3. **Velocidad en la Detección:** La rapidez en la detección es crucial para evitar pérdidas significativas. Los sistemas deben ser capaces de analizar transacciones en tiempo real y tomar decisiones instantáneas para prevenir o limitar el impacto del fraude.
4. **Falsos Positivos y Negativos:** En la detección de fraude, existe el desafío constante de equilibrar la precisión del modelo para evitar falsos positivos (clasificar incorrectamente una transacción legítima como fraudulenta) y falsos negativos (no identificar una transacción fraudulenta).

Tras ser identificados los desafíos para la detección de fraude bancario, es de vital importancia contar con herramientas informáticas. Dichas herramientas deben permitir la identificación de patrones de comportamiento inusuales y/o que corresponden a actividades potencialmente fraudulentas dentro del gran número de registros de transacciones.

En el mundo digital actual, la detección de fraude bancario representa un campo dinámico y desafiante. Desde la perspectiva de la ingeniería informática, se necesita una combinación de enfoques tecnológicos avanzados, análisis de datos efectivos y colaboración interdisciplinaria para construir sistemas robustos y adaptativos capaces de proteger la integridad financiera y la confianza de los clientes en el sistema bancario. La evolución continua de estas estrategias es esencial para hacer frente a la creciente sofisticación de las amenazas en el panorama financiero moderno.

1.2 Detección de Anomalías

La detección de anomalías es un campo crucial en la ciencia de datos y la ingeniería informática, con aplicaciones extendidas en diversas industrias, desde la detección de fraudes hasta el mantenimiento predictivo. En este contexto, la detección de anomalías se refiere a la identificación de patrones inusuales o atípicos en conjuntos de datos, lo que puede indicar eventos o comportamientos anómalos. Desde la perspectiva de la ingeniería informática, la implementación efectiva de técnicas de detección de anomalías requiere una combinación de algoritmos avanzados, procesamiento eficiente de datos y adaptabilidad a patrones cambiantes.

Desafíos en la Detección de Anomalías:

1. **Naturaleza Evolutiva de los Datos:** Los datos en tiempo real y aquellos provenientes de entornos dinámicos pueden cambiar con el tiempo. Los algoritmos de detección de anomalías deben adaptarse a estas variaciones para mantener su eficacia.

2. **Equilibrio entre Falsos Positivos y Negativos:** Como en la detección de fraude, encontrar el equilibrio adecuado entre identificar anomalías reales y evitar falsos positivos o negativos es un desafío constante.
3. **Gran Volumen de Datos:** Manejar grandes volúmenes de datos en tiempo real puede ser computacionalmente intensivo. La eficiencia en el procesamiento es esencial para garantizar respuestas rápidas y precisas.
4. **Interpretación de Resultados:** Comprender la causa de una anomalía detectada puede ser igualmente importante que identificarla. La interpretación de los resultados puede requerir un análisis adicional para contextualizar las anomalías en el dominio específico.

Enfoques de Detección de Anomalías desde la Ingeniería Informática:

1. **Aprendizaje No Supervisado:** Algoritmos no supervisados, como el algoritmo de K-means o el DBSCAN, pueden identificar patrones inusuales sin la necesidad de etiquetas previas. Estos métodos son útiles cuando las anomalías son raras y difíciles de clasificar.
2. **Aprendizaje Supervisado:** En situaciones donde se tienen datos etiquetados, los modelos de aprendizaje supervisado pueden entrenarse para reconocer patrones normales y, por lo tanto, identificar anomalías con mayor precisión.
3. **Redes Neuronales y Aprendizaje Profundo:** Las redes neuronales, especialmente aquellas diseñadas para la detección de anomalías, pueden aprender representaciones complejas y capturar patrones sutiles en los datos, siendo efectivas en escenarios con información de alta dimensionalidad.
4. **Técnicas de Transformación de Datos:** La aplicación de técnicas como PCA (Análisis de Componentes Principales) puede reducir la dimensionalidad de los datos y resaltar patrones anómalos. Esto es especialmente útil cuando se enfrenta con conjuntos de datos complejos.
5. **Monitorización Continua y Actualización de Modelos:** La detección de anomalías es un proceso continuo. La monitorización en tiempo real y la actualización periódica de los modelos son esenciales para mantener la efectividad a lo largo del tiempo.

Aplicaciones Prácticas de la Detección de Anomalías:

1. **Seguridad Informática:** Identificación de comportamientos anómalos en la red que pueden indicar ataques cibernéticos.
2. **Mantenimiento Predictivo:** Detección de anomalías en el rendimiento de maquinaria o sistemas para prevenir fallas y realizar mantenimiento de manera proactiva.

3. **Finanzas:** Identificación de transacciones inusuales que podrían indicar fraude o actividades financieras sospechosas.
4. **Salud:** Detección de patrones anómalos en datos médicos para la identificación temprana de enfermedades o condiciones médicas.

Conceptos asociados:

Anomalía: Un cambio dentro de un patrón de datos, un valor atípico o un evento que se encuentre fuera de una tendencia estándar; una desviación de algo esperado o algo que no se ajusta a las expectativas. Una anomalía, o un valor atípico en un patrón, pueden indicar algo que está fuera de la norma o algo que posiblemente no esté bien. Ellas pueden ser puntuales/globales, contextuales o colectivas (ServiceNow, 2022).

Anomalías puntuales/globales: Un único punto de datos que se ha identificado como demasiado lejos del resto (ServiceNow, 2022).

Anomalías contextuales: Anomalía anormal en el contexto de un conjunto de datos, pero normal en el contexto de otro conjunto de datos. Este es el tipo más común de anomalía contextual en datos de series temporales (ServiceNow, 2022).

Anomalías colectivas: Cuando un subconjunto completo de datos es anómalo al compararlo con un conjunto más amplio de datos; los puntos de datos individuales no se tienen en cuenta cuando se identifican anomalías colectivas (ServiceNow, 2022).

Detección de anomalías: La identificación de un valor atípico raro o un punto de datos fuera de las tendencias de un conjunto de datos. Las anomalías pueden indicar eventos sospechosos, fallos, defectos o fraude (ServiceNow, 2022).

La detección de anomalías, desde la perspectiva de la ingeniería informática, es una herramienta esencial para abordar una variedad de desafíos en distintos campos. Con la continua evolución de algoritmos y la disponibilidad de datos, la detección de anomalías se posiciona como una disciplina en constante crecimiento, ofreciendo soluciones innovadoras para problemas complejos en el análisis de datos y la toma de decisiones en tiempo real.

1.3 Aplicación de Deep Learning (DL) en sistemas bancarios.

La aplicación de técnicas de Deep Learning (DL) en sistemas bancarios marca un punto crucial en la convergencia de la inteligencia artificial y las finanzas. En la búsqueda constante de eficiencia, seguridad y personalización, las instituciones financieras han recurrido al poder del aprendizaje

profundo para transformar fundamentalmente la forma en que operan y ofrecen servicios. En este contexto, exploraremos cómo la aplicación de DL en sistemas bancarios ha redefinido la eficiencia operativa, la seguridad financiera, y la experiencia del cliente.

El aprendizaje profundo ha impulsado una transformación operativa sin precedentes en el sector bancario. Desde la automatización de procesos rutinarios hasta la optimización de flujos de trabajo complejos, los modelos de DL han permitido a las instituciones financieras mejorar la eficiencia y reducir costos. La capacidad de analizar grandes volúmenes de datos de manera rápida y precisa ha llevado a una toma de decisiones más ágil y fundamentada.

1.3.1 Deep Learning

El aprendizaje profundo es una rama del aprendizaje automático que es un subconjunto de la inteligencia artificial y es una Red neuronal efectiva de tres o más capas (Pandey, 2017).

Mientras que una red neuronal de una sola capa todavía puede producir clasificaciones precisas, más capas ocultas pueden promover la optimización y mejorar la precisión (Pandey, 2017).

El objetivo del DL es estudiar redes neuronales artificiales. La técnica estándar se refiere al tamaño de las redes neuronales, y se considera el modelo de retro propagación (Alarfaj et al., 2022).

Algoritmos de aprendizaje profundo (DL) aplicados en redes informáticas, detección de intrusiones, banca, seguros, redes de telefonía móvil, fraude en atención sanitaria detección, detección médica y de malware, detección por vídeo vigilancia, seguimiento de ubicación, detección de malware en Android, domótica y predicción de enfermedades cardíacas. Exploramos la aplicación práctica de ML, particularmente los algoritmos DL, identificar robos de tarjetas de crédito en la industria bancaria en este papel (Alarfaj et al., 2022).

En los últimos años, los enfoques de aprendizaje profundo han recibido atención significativa debido a resultados sustanciales y prometedores en diversas aplicaciones, como la visión por computadora, el procesamiento del lenguaje natural y la voz (Alarfaj et al., 2022).

El DL se utiliza en muchas aplicaciones de inteligencia artificial (IA) que mejoran la automatización al realizar tareas analíticas y físicas sin intervención humana.

La tecnología de DL impulsa muchos productos y servicios cotidianos, como asistentes digitales, controles de TV habilitados por voz y detección de fraude con tarjeta de crédito, así como tecnologías emergentes como vehículos autónomos.

Los algoritmos de DL pueden procesar datos no estructurados, como texto e imágenes, y automatizar la extracción de características, eliminando parte de la dependencia de expertos humanos.

Los algoritmos pueden detectar anomalías en puntos de datos específicos e identificar incidentes inusuales que parecen sospechosos por ser diferentes a los patrones establecidos de comportamiento.

La mayoría de los sectores pueden beneficiarse de la detección de anomalías. Por ejemplo, el comercio electrónico puede identificar cualquier comportamiento extraño o problemas de calidad de productos, como fallos en los precios o cambios anormales en la estacionalidad

1.3.2 Métodos de aprendizaje automático

Para generar conocimiento a partir de datos, es necesaria la aplicación de algoritmos de aprendizaje automático, que pueden clasificarse en supervisados o no supervisados, en función del uso de la etiqueta de clase para realizar el aprendizaje.

- Métodos supervisados: cuando se conoce la existencia de anomalías en los datos, y se sabe cuáles son las técnicas usadas son de clasificación supervisada. En este tipo de problemas se tienen dos conjuntos de datos, uno de entrenamiento y otro de test, como se dispone de toda la información, los datos están etiquetados en función de si son anomalía o no, donde se construye un modelo que aprenda a distinguir ente un dato anómalo y uno legítimo.
- Métodos semi-supervisados: cuando se conoce la existencia de anomalías, pero no se encuentran en el conjunto de datos
- Métodos no supervisados cuando se dispone de anomalías en el conjunto, pero no están etiquetadas, no se conoce a priori si un dato es una anomalía o no, es decir, tanto anomalías como comportamientos legítimos están mezclados. En este campo existen también varias alternativas.

1.3.3 Ventajas de los métodos supervisados:

Precisión: Los algoritmos de aprendizaje supervisado pueden predecir resultados con alta precisión.

Automatización: Permite una mayor automatización y extracción de información más exhaustiva de los datos.

Aplicabilidad: Se puede utilizar para resolver una amplia variedad de problemas del mundo real a escala, como la clasificación de spam en una carpeta distinta de la bandeja de entrada.

Eficiencia: Es altamente valorado por las empresas, ya que permite a los sistemas de inteligencia artificial tomar decisiones empresariales de manera más rápida.

Confiabilidad: Las decisiones tomadas por los sistemas de inteligencia artificial son más precisas y, por tanto, confiables.

Reducción de costos: Supone un avance importante para reducir costos y mejorar las soluciones.

Debido a las ventajas mencionadas se eligió el método supervisado como el mejor debido a la naturaleza de los datos.

1.3.4 Redes Neuronales Artificiales

El modelo simplificado y abstracto de las redes neuronales artificiales surge de intentar imitar el comportamiento de las neuronas que se encuentran en el cerebro. En las últimas décadas las Redes Neuronales Artificiales (ANN) han recibido un interés particular como una tecnología para minería de datos, puesto que ofrece los medios para modelar de manera efectiva y eficiente problemas grandes y complejos. Los modelos de ANN son dirigidos a partir de los datos, es decir, son capaces de encontrar relaciones (patrones) de forma inductiva por medio de los algoritmos de aprendizaje basado en los datos existentes más que requerir la ayuda de un modelador para especificar la forma funcional y sus interacciones.

El aprendizaje es la clave de la adaptabilidad de la red neuronal y esencialmente es el proceso en el que se adaptan las sinapsis, para que la red responda de un modo distinto a los estímulos del medio. Entrenar una red neuronal es un proceso que modifica los pesos w y el sesgo estadístico b que se obtiene de la interacción entre dos capas, con el fin de que la red pueda a partir de datos de entrada, generar una salida (Ameijeiras Sánchez et al., 2021).

En el artículo “JOURNAL OF OPTOELECTRONICS LASER” se evidencia la utilidad de este modelo con una métrica de exactitud de 84% (Sharma & Lavavanshi, 2022).

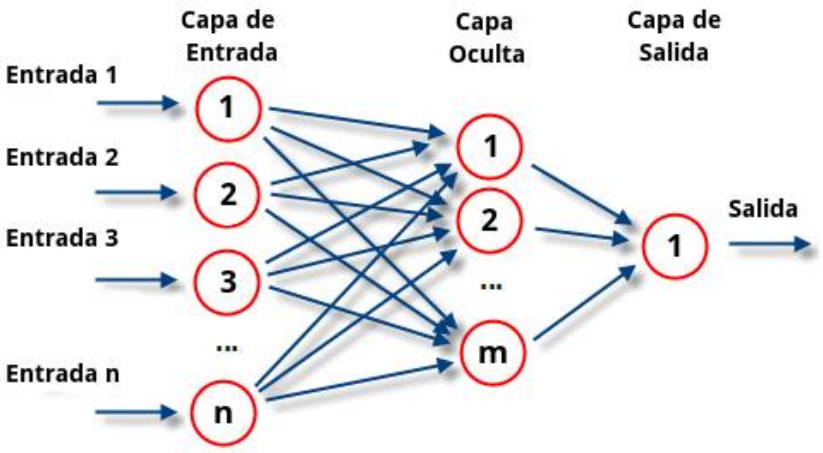

Figura 1. Ejemplo de ANN

1.3.5 Autoenconders

Un autoencoder (AE) es un modelo computacional basado en redes neuronales compuesto por un codificador y un decodificador, usado para tareas de aprendizaje no supervisado. El principal objetivo de este modelo según es aprender a codificar y decodificar una entrada dada. Esto es particularmente útil cuando se trata de representar los datos con un conjunto más pequeño de características y cuando se trata de reconstruirlos mientras se eliminan las características no deseadas. Existen varios tipos de autocodificadores y pueden emplearse para resolver diversos problemas.

Es posible entrenar un autocodificador para que devuelva versiones modificadas de los datos de entrada. Los autocodificadores codifican los datos y luego los reconstruyen mediante el decodificador. Asimismo, existen autocodificadores capaces de generar información (Arroyo et al., 2020).

La detección de anomalías (*outliers*) con Autoencoders es una estrategia no supervisada para identificar anomalías cuando los datos no están etiquetados. Si bien esta estrategia hace uso de *Autoencoders*, no utiliza directamente su resultado como forma de detectar anomalías, sino que emplea el error de reconstrucción producido al revertir la reducción de dimensionalidad. El error de reconstrucción como estrategia para detectar anomalías se basa en la siguiente idea: los métodos de reducción de dimensionalidad permiten proyectar las observaciones en un espacio de menor dimensión que el espacio original, a la vez que tratan de conservar la mayor información posible. La forma en que consiguen minimizar la perdida global de información es buscando un nuevo espacio en el que la mayoría de observaciones puedan ser bien representadas (Rodrigo, 2020).

Dado que la búsqueda de ese nuevo espacio ha sido guiada por la mayoría de las observaciones, serán las observaciones más próximas al promedio las que mejor puedan ser proyectadas y en consecuencia mejor reconstruidas. Las observaciones anómalas, por el contrario, serán mal proyectadas y su reconstrucción será peor. Es este error de reconstrucción (elevado al cuadrado) el que puede emplearse para identificar anomalías (Rodrigo, 2020).

En el artículo “An Autoencoder Based Model for Detecting Fraudulent Credit Card Transaction” se utiliza un algoritmo de este tipo para detectar fraude en transacciones de tarjetas de crédito con una exactitud del 99% (Lin & Jiang, 2021).

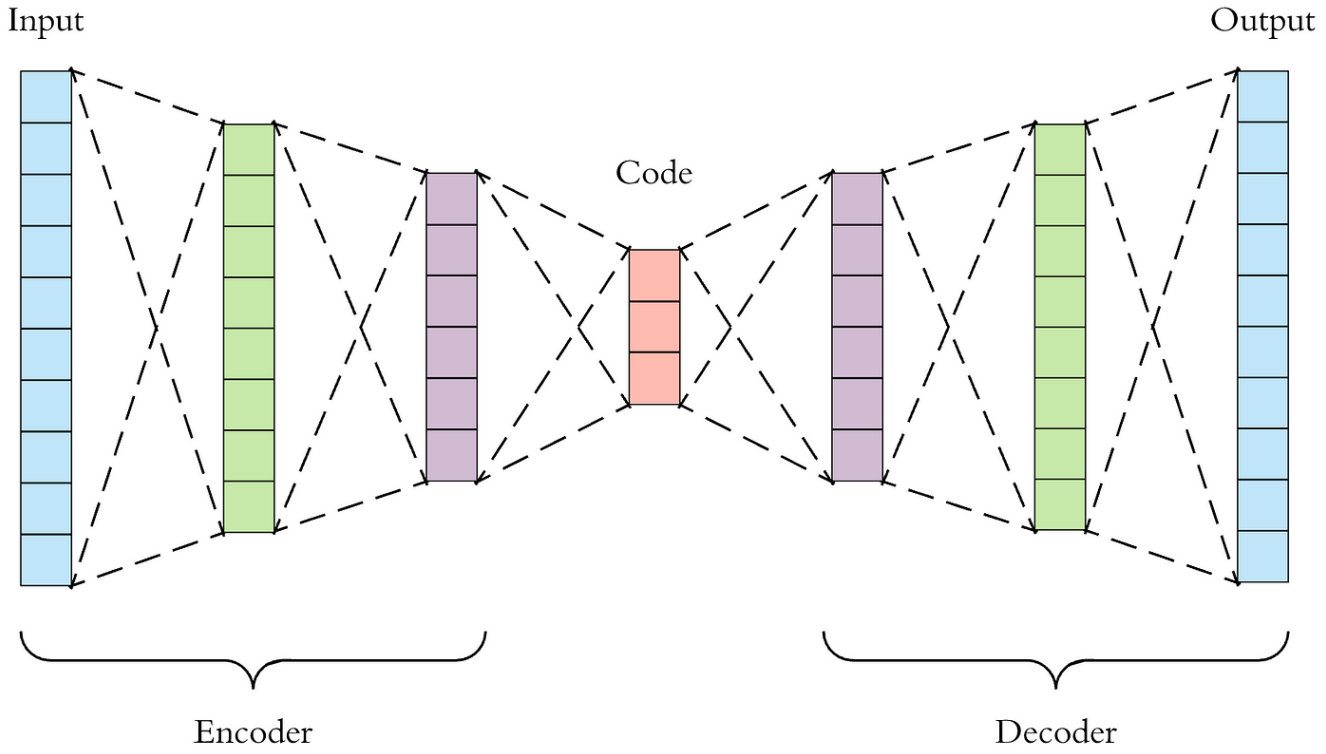

Figura 2. Ejemplo de Autoencoder

1.3.6 Redes Neuronales Convolucionales

Una red neuronal convolucional según (Durán Suárez, 2017) es un tipo de red multicapa que consta de diversas capas convolucionales y de pooling (submuestreo) alternadas

Las redes neuronales convolucionales (CNN), son la elección popular de las redes neuronales para el análisis de imágenes visuales, aunque su capacidad permite extraer características ocultas en datos de alta dimensión con estructura compleja y ha permitido su uso como extractores de características en la detección de valores atípicos para el conjunto de datos secuenciales y de imágenes. Esta más

enfocado a tareas de aprendizaje supervisado sobre conjuntos de datos previamente etiquetados (Abroyan, 2017).

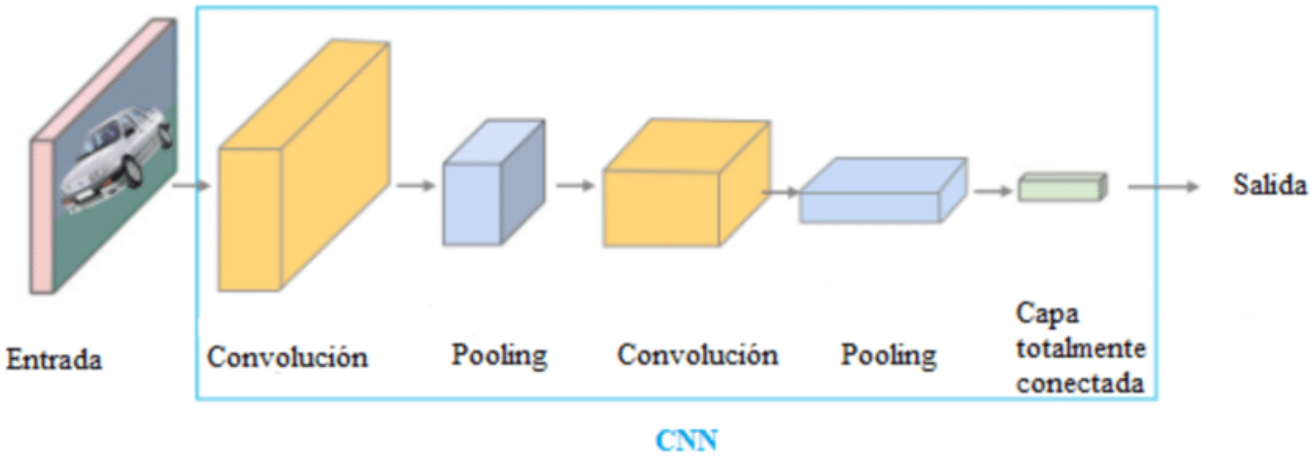

Figura 3. Ejemplo de Red Neuronal Convolucional

Se hace uso de este modelo para la detección de fraude bancario en el artículo “Credit Card Fraud Detection System Using CNN" se obtuvo una accuracy del 95% (Shanmugapriya et al., 2022).

1.3.7 Modelos Generativos

El objetivo principal de las Redes generativas antagónicas (GAN) (Goodfellow et al., 2014) es generar datos desde cero. Para ello las GAN emplean dos redes neuronales y las enfrentan mutuamente. La primera red es el "generador" y la segunda es el "discriminador". Ambas redes fueron entrenadas con un mismo conjunto de datos, pero la primera debe intentar crear variaciones de los datos que ya ha visto, en el caso de los rostros de personas que no existen, debe crear variaciones de los rostros que ya ha visto. La red discriminatoria debe identificar si ese rostro que está viendo forma parte del entrenamiento original o si es un rostro falso que creó la red generativa. Mientras más lo hace, la red generativa se hace mejor creando y a la red discriminadora se le hace más difícil detectar si el rostro es falso.

La red generadora necesita la discriminadora para saber cómo crear una imitación tan realista que la segunda no logre distinguir de una imagen real. Una red generadora por sí sola crearía solo ruido aleatorio, el concepto es que la red discriminadora hace de guía sobre cuáles imágenes crear y ayuda a la red generativa a aprender los aspectos que comprenden una imagen real. El modelo entrena a ambas redes y las enfrenta en una dura competición para que se mejoren a sí mismas. Eventualmente, el discriminador será capaz de identificar la más pequeña diferencia entre lo que es real y lo que fue generado, y la red generativa será capaz de crear imágenes que el discriminador no puede distinguir.

Se utilizan algunas variantes de este algoritmo con buenos resultados en el artículo “A Survey on GAN Techniques for Data Augmentation to Address the Imbalanced Data Issues in Credit Card Fraud Detection” (Strelcenia & Prakoonwit, 2023).

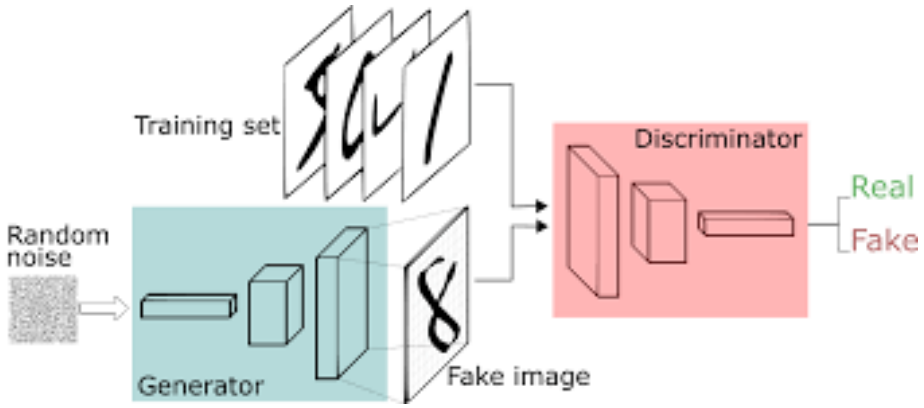

Figura 4. Ejemplo de Red Generativa Antagónica

1.3.8 Redes Neuronales Recurrentes

Las Redes Neuronales Recurrentes (RNN) se han utilizado sobre todo en datos secuenciales, como datos de series temporales, datos de audio y habla, y lenguaje. A diferencia de las redes de avance, las RNN utilizan una memoria interna para procesar las entradas entrantes. Las RNN se utilizan en el análisis de datos de series temporales en varios campos. En general, la RNN procesa las series de entrada de una en una, durante su funcionamiento. Las unidades de la capa oculta contienen información sobre la historia de la entrada en el vector de estado. Las RNN se pueden entrenar utilizando el método de propagación en a través del tiempo (BPTT). Con el método BPTT, la diferenciación de la pérdida en cualquier momento refleja los pesos de la red en el paso de tiempo anterior. El entrenamiento de las RNN es más difícil que el de las redes neuronales de avance (FFNN) y el periodo de entrenamiento de las RNN es más largo (Ozbayoglu et al., 2020).

Se hace mención de este modelo por su relevancia dentro del DL, aunque no se encontraron referentes de su uso para detectar fraude bancario en la investigación en cuestión. Sin embargo, el hecho de que

se mencione este modelo en particular sugiere que es un modelo importante en el campo del DL y puede tener aplicaciones en otros campos.

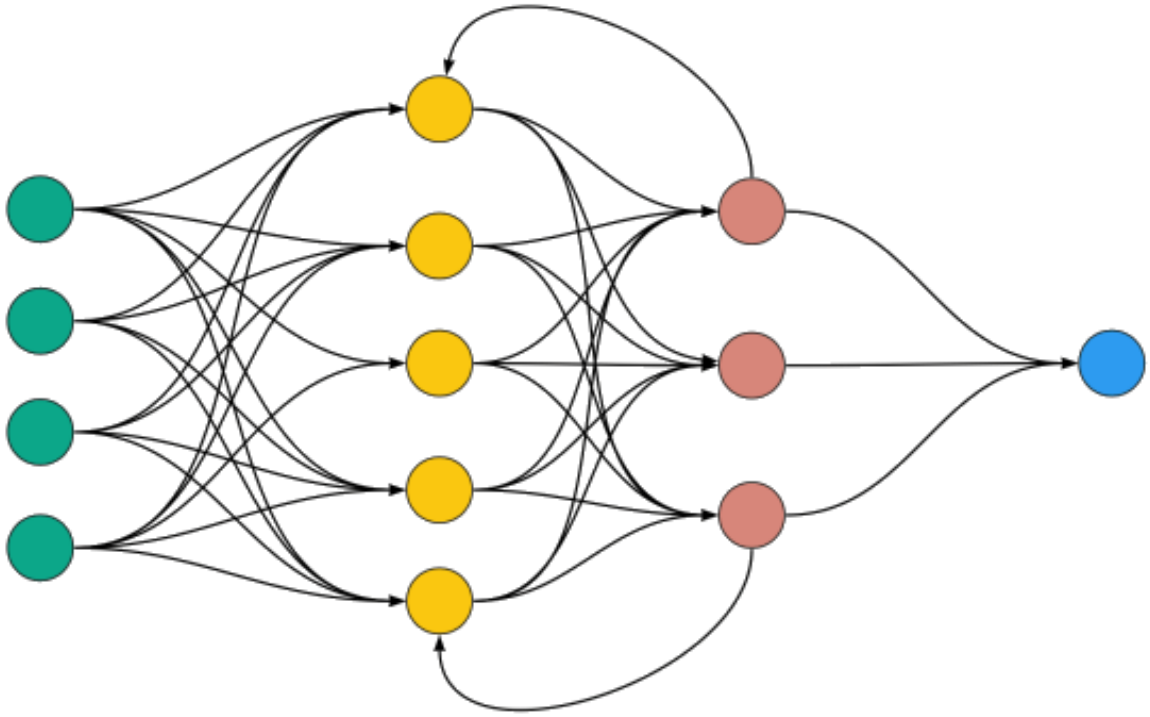

Figura 5. Ejemplo de Red Neuronal Recurrente

1.3.9 Ventajas de usar algoritmos de DL para detección de fraude bancario

Detección más rápida y eficiente: Los modelos de DL pueden identificar rápidamente patrones sospechosos y comportamientos que podrían tomar meses a los humanos para ser establecidos.

Reducción en el tiempo de revisión manual: El tiempo invertido en revisar manualmente la información puede ser drásticamente reducido cuando se deja que los modelos analicen toda la información.

Mejores predicciones en grandes conjuntos de información: Mientras más información se le suministre a un modelo de DL, más entrenado llega a ser. Esto quiere decir que, aunque es un reto encontrar patrones en grandes conjuntos de información para los humanos, es totalmente lo opuesto para los modelos de DL.

Solución económica y efectiva: En vez de contratar más agentes de operaciones de riesgo, solo se necesita un algoritmo de DL para analizar toda la información suministrada, independientemente cuan grande sea el volumen de los datos.

Adaptabilidad: Los algoritmos de DL permiten adaptarse y mejorar constantemente a medida que se recopilan más datos y se descubren nuevos métodos de fraude.

Detección de patrones sutiles: Los algoritmos aprenden de manera autónoma y pueden detectar patrones sutiles que podrían pasar desapercibidos para un análisis humano.

1.3.10 Métricas para comparar modelos de DL:

Accuracy (Exactitud): Es la proporción de predicciones correctas respecto al total de predicciones.

$$Accuracy = \frac{VP + VN}{VP + FP + FN + VN}$$

Donde;

VP, corresponde a los verdaderos positivos

VN, corresponde a los verdaderos negativos

FP, corresponde a los falsos positivos

FN, corresponde a los falsos negativos

Balanced Accuracy o Precisión Balanceada: es una métrica que se utiliza para evaluar el rendimiento de un modelo de clasificación, especialmente en casos de conjuntos de datos desbalanceados. Se define como el promedio de la sensibilidad (tasa de verdaderos positivos) y la especificidad (tasa de verdaderos negativos)

$$Specificity = \frac{TN}{TN+FP}$$

$$Balanced\ Accuracy = \frac{(ReCall + Specificity)}{2}$$

La precisión balanceada es una métrica útil para evaluar el rendimiento de un modelo en conjuntos de datos desequilibrados.

Precision (Precisión): Es la proporción de predicciones positivas que fueron correctas.

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

Un modelo que no produce falsos positivos tiene una precisión de 1.0. La precisión es útil cuando el costo de los falsos positivos es alto, como en el diagnóstico médico o la detección de fraude.

Recall (Exhaustividad): Es la proporción de casos positivos reales que fueron identificados correctamente.

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

Un modelo que no produce falsos negativos tiene una recall de 1.0. La recall es útil cuando el costo de los falsos negativos es alto, como en la detección de enfermedades graves.

F1 Score: Es una medida que combina la precisión y la exhaustividad. Es útil en situaciones donde los falsos positivos y los falsos negativos tienen un impacto similar.

$$F1 = \frac{2 * Precision * ReCall}{Precision + ReCall}$$

AUC-ROC (Área bajo la curva del operador receptor): Es una métrica de rendimiento para problemas de clasificación binaria. Un AUC-ROC de 1 representa un modelo perfecto, mientras que un AUC-ROC de 0.5 representa un modelo aleatorio.

Log Loss (Pérdida logarítmica): Es una métrica de rendimiento para problemas de clasificación binaria. Cuanto más cerca de 0, mejor es el modelo.

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

El log loss es una métrica útil para evaluar la calidad de un modelo de clasificación binaria, especialmente cuando las clases no están equilibradas

En resumen, la aplicación de Deep Learning en sistemas bancarios representa un paradigma transformador en la industria financiera. Desde la mejora operativa hasta la seguridad avanzada y la personalización de servicios, el aprendizaje profundo está forjando un camino hacia un sistema bancario más inteligente, eficiente y centrado en el cliente. A medida que la tecnología evoluciona, la integración ética y estratégica de DL continuará definiendo el futuro de la banca moderna.

1.3.11 Trabajos Relacionados

La minería de datos y el aprendizaje automático son métodos populares para estudiar y combatir los casos de fraude con tarjetas de crédito.

Existe una gran cantidad de estudios que explotaron la fuerza de la minería de datos y el aprendizaje automático para prevenir las actividades fraudulentas con tarjetas de crédito.

Por su parte, en (Adewumi & Akinyelu, 2017), los autores han realizado una revisión de las técnicas mejoradas de detección de fraudes con tarjetas de crédito. Precisamente, el trabajo se centró en las recientes técnicas de detección de fraudes de tarjetas de crédito basadas en el aprendizaje automático e inspiradas en la naturaleza propuestas en la literatura. Los autores han proporcionado una imagen de la tendencia reciente en la detección de fraudes con tarjetas de crédito. Además, esta revisión ha descrito algunas limitaciones y contribuciones de las técnicas de detección de fraude con tarjetas de crédito existentes, y también ha proporcionado la información básica necesaria para los investigadores en este dominio.

Como conclusión, el trabajo podría servir como guía y trampolín para las instituciones financieras y las personas que buscan técnicas nuevas y efectivas de detección de fraude con tarjetas de crédito.

En (Dhankhad et al., 2018), los autores proponen diferentes algoritmos de aprendizaje automático supervisados para detectar transacciones fraudulentas con tarjetas de crédito utilizando un conjunto de datos del mundo real. Además, han empleado estos algoritmos para implementar un superclasificador utilizando métodos de aprendizaje por conjuntos. Identificaron las variables más importantes que pueden conducir a una mayor precisión en la detección de transacciones fraudulentas con tarjetas de crédito.

Por su parte, en (Awoyemi et al., 2017), los autores han propuesto analizar el rendimiento de Naive Bayes, KNN y regresión logística en datos de fraude de tarjetas de crédito muy sesgados. Los autores desarrollaron una técnica híbrida de submuestreo y sobre muestreo de los datos asimétricos. Las tres técnicas se aplican a los datos sin procesar y preprocesados. Con una implementación realizada en Python, los resultados muestran una precisión óptima para Naive Bayes, KNN y regresión logística de 97,92%, 97,69% y 54,86% respectivamente.

En (Yee et al., 2018), los autores emplearon técnicas de aprendizaje automático para predecir las transacciones sospechosas y no sospechosas automáticamente mediante el uso de clasificadores. La combinación de técnicas de aprendizaje automático y minería de datos pudo identificar las transacciones genuinas y no genuinas al aprender los patrones de los datos. Los autores analizaron la clasificación basada en supervisión que utiliza clasificadores de redes bayesianas, como, K2, Tree Augmented Naïve Bayes (TAN) y Naïve Bayes, logística y clasificadores J48. Después de preprocesar

el conjunto de datos mediante la normalización y el análisis de componentes principales, todos los clasificadores lograron una precisión superior al 95% en comparación con los resultados obtenidos antes de preprocesar el conjunto de datos.

En esta línea, en (Khare & Sait, 2018), los autores han investigado el rendimiento del árbol de decisiones, Random Forest, SVM y regresión logística en datos de fraude de tarjetas de crédito muy sesgados. El conjunto de datos de transacciones con tarjetas de crédito proviene de titulares de tarjetas europeos que contienen 284.786 instancias. Estas técnicas se han aplicado a los datos sin procesar y preprocesados. El rendimiento de las técnicas se evaluó en función de la precisión, sensibilidad, y especificidad. Los resultados indicaron que la precisión óptima para la regresión logística, el árbol de decisión, Random Forest y SVM con valores de 97,7%, 95,5% y 98,6%, 97,5% respectivamente. Como se puede ver, Random Forest obtuvo un desempeño superior al resto de los algoritmos.

Finalmente, en (Sailusha et al., 2020), los autores implementaron algoritmos de random forest y el Adaboost para la detección de fraude en tarjetas de crédito. Los resultados de los dos algoritmos se basan en la accuracy, precision, recall y F1-score. Se compararon los algoritmos Random Forest y Adaboost y el algoritmo que ha demostrado el mejor desempeño para detectar el fraude fue Random Forest.

1.4 Metodología de Minería de Datos (MD)

La Minería de Datos (MD) se ha convertido en una disciplina esencial para descubrir patrones, tendencias y conocimientos útiles en conjuntos de datos masivos. La aplicación efectiva de la MD requiere una metodología estructurada que guíe el proceso desde la selección de datos hasta la interpretación de resultados.

El Descubrimiento de conocimiento en bases de datos (kdd, del inglés Knowledge Discovery in Databases) es básicamente un proceso automático en el que se combinan descubrimiento y análisis. El proceso consiste en extraer patrones en forma de reglas o funciones, a partir de los datos, para que el usuario los analice. Esta tarea implica generalmente preprocesar los datos, hacer minería de datos (data mining) y presentar resultados. KDD se puede aplicar en diferentes dominios, por ejemplo, para determinar perfiles de clientes fraudulentos (evasión de impuestos), para descubrir relaciones implícitas existentes entre síntomas y enfermedades, entre características técnicas y diagnóstico del estado de equipos y máquinas, para determinar perfiles de estudiantes “académicamente exitosos” en términos de sus características socioeconómicas y para determinar patrones de compra de los clientes en sus canastas de mercado (Rajan et al., 2021).

1.4.1 Selección de la metodología de Minería de Datos.

Metodologías dominantes para el proceso de la minería de datos:

- Knowledge Discovery in Databases (KDD): Es una metodología propuesta por Fayyad en 1996, propone 5 fases: Selección, preprocesamiento, transformación, minería de datos y evaluación e implantación. Es un proceso iterativo e interactivo (Xu et al., 2021).
- SEMMA: Acrónimo a las cinco fases: (Sample, Explore, Modify, Model, Assess). La metodología es propuesta por SAS Institute Inc, la define como: "… proceso de selección, exploración y modelamiento de grandes cantidades de datos para descubrir patrones de negocios desconocidos…"
- Cross-Industry Standard Process for Data Mining (CRISP-DM): Iniciativa financiada por la Comunidad Europea la cual se ha unido para desarrollar una plataforma para Minería de Datos. Persigue como objetivos: Fomentar la interoperabilidad de las herramientas a través de todo el proceso de minería de datos y eliminar la experiencia misteriosa y costosa de las tareas simples de minería de datos.

Se selecciono KDD dadas las siguientes razones:

- Proceso Iterativo e Interactivo: KDD es un proceso iterativo e interactivo, lo que permite la revisión y el refinamiento continuo del modelo.
- Fases bien definidas: KDD propone 5 fases: Selección, preprocesamiento, transformación, minería de datos y evaluación e implantación. Estas fases proporcionan una estructura clara para el proceso de minería de datos.
- Adaptabilidad: KDD es adaptable a diferentes tipos de datos y objetivos de minería de datos.
- Enfoque integral: KDD no solo se centra en la minería de datos, sino que también considera los pasos previos necesarios para preparar los datos y los pasos posteriores para evaluar e implementar los resultados.
- Interpretación y Evaluación: KDD incluye pasos para la interpretación de los patrones minados y la evaluación de los resultados obtenidos.

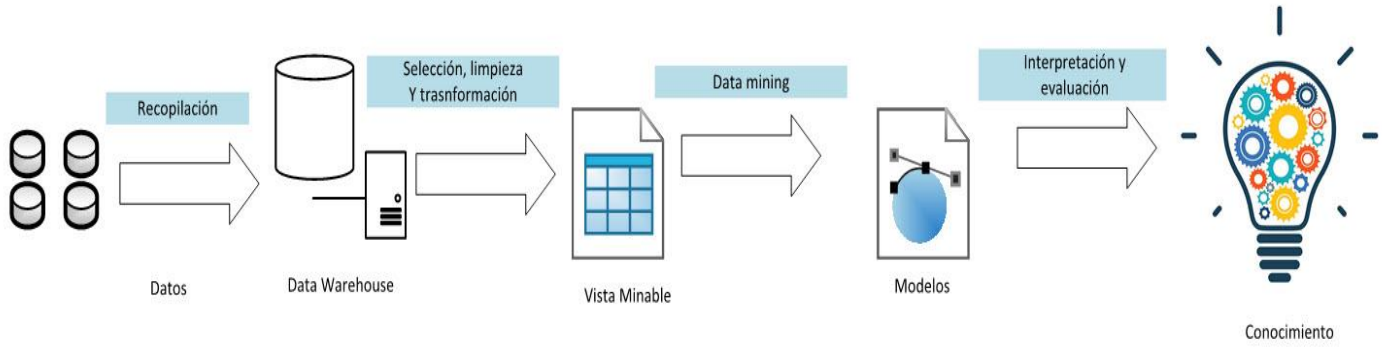

Figura 6. Metodología KDD

1.4.2 Etapas del proceso KDD

El proceso KDD es interactivo e iterativo, involucra numerosos pasos con la intervención del usuario en la toma de muchas decisiones. Se resume en las siguientes etapas:

- Selección.
- Preprocesamiento/limpieza.
- Transformación/reducción.
- Minería de datos (data mining).
- Interpretación/evaluación.

1.5 Tecnologías y herramientas.

En el ámbito de la Minería de Datos (MD), varias tecnologías y herramientas son utilizadas para llevar a cabo las diferentes etapas del proceso. A continuación, se mencionan algunas de las tecnologías y herramientas comúnmente empleadas en la práctica de la MD:

Plataformas y servicios clave de transmisión de datos:

Apache Kafka es una plataforma de transmisión de eventos de código abierto y distribuida que se utiliza para la integración de datos en tiempo real y el procesamiento de flujos de datos. Es utilizada por miles de empresas para crear tuberías de datos de alto rendimiento, integración de datos, análisis en tiempo real y aplicaciones críticas.

Kafka es capaz de manejar grandes volúmenes de datos y proporciona una plataforma unificada para la manipulación en tiempo real de fuentes de datos. Algunas de las características clave de Kafka incluyen alta capacidad de procesamiento, escalabilidad, almacenamiento permanente, alta disponibilidad y una amplia gama de herramientas para el análisis y procesamiento de datos.

1.5.1 Lenguajes de programación:

R: Es un ambiente de programación formado por un conjunto de herramientas muy flexibles que pueden ampliarse fácilmente mediante paquetes, librerías o definiendo nuestras propias funciones**.** Además, es **gratuito y de código abierto,** un Open Source parte del proyecto GNU, como Linux o Mozilla Firefox (*Formación*, 2022).

C++: Es de alto nivel, siendo un lenguaje ensamblador e híbrido, altamente didáctico. C++ colabora en que los programadores puedan escribir otros programas portátiles y rápidos. Unix, por ejemplo, es un sistema operativo escrito en Lenguaje C. Tiene compatibilidad con bibliotecas enriquecidas y se caracteriza por su velocidad en comparación a otras opciones disponibles (*LLC*, 2022).

JavaScript: Es un robusto lenguaje de programación que se puede aplicar a un documento HTML y usarse para crear interactividad dinámica en los sitios web. Fue inventado por Brendan Eich, cofundador del proyecto Mozilla, Mozilla Foundation y la Corporación Mozilla (DOCS, M. W., 2023).

Java: Es un lenguaje multiplataforma, orientado a objetos y centrado en la red que se puede utilizar como una plataforma en sí mismo. Es un lenguaje de programación rápido, seguro y confiable para codificar todo, desde aplicaciones móviles y software empresarial hasta aplicaciones de big data y tecnologías del lado del servidor (Services, A. W., 2022).

C#: Es un lenguaje de programación **orientado a componentes**, orientado a objetos. C# proporciona construcciones de lenguaje para admitir directamente estos conceptos, por lo que se trata de un lenguaje natural en el que crear y usar componentes de software. Desde su origen, C# ha agregado características para admitir nuevas cargas de trabajo y prácticas de diseño de software emergentes (BillWagner, 2023).

Python: Python es un lenguaje donde su código se ejecuta en el navegador al cargar la página, es independiente de la plataforma y orientado a objetos, está listo para realizar cualquier tipo de programa desde aplicaciones de Windows hasta servidores de red o incluso páginas web. Es un lenguaje interpretado, lo que ofrece ventajas como la velocidad de desarrollo e inconvenientes como una velocidad más baja al ser ejecutado (Álvarez, M., 2013).

Python es uno de los lenguajes de programación más populares para el aprendizaje profundo. Ofrece una amplia variedad de bibliotecas y herramientas para el procesamiento de datos y el aprendizaje automático, como TensorFlow, Keras, PyTorch y Theano. Estas bibliotecas son muy populares entre los científicos de datos y los desarrolladores debido a su facilidad de uso y su capacidad para manejar grandes volúmenes de datos.

Figura 7. Lenguaje de programación Python

La **última versión estable** de Python es la 3.11.0, que fue lanzada el 24 de octubre de 2022. Python es el lenguaje de programación preferido para el deep learning debido a su simplicidad y sus bibliotecas especializadas como Numpy, Scipy, Pandas, Matplotlib, Theano, TensorFlow y Keras. Algunas de las características de Python que lo hacen ideal para el deep learning son:

- **Simplicidad**: Python es un lenguaje de programación fácil de aprender y usar. Su sintaxis es clara y concisa, lo que facilita la lectura y escritura del código.
- **Flexibilidad**: Python es un lenguaje multiparadigma que admite programación orientada a objetos, programación funcional y programación estructurada. Esto lo hace ideal para el desarrollo de aplicaciones complejas.
- **Bibliotecas especializadas**: Python cuenta con una gran cantidad de bibliotecas especializadas para el deep learning, como Numpy, Scipy, Pandas, Matplotlib, Theano, TensorFlow y Keras. Estas bibliotecas simplifican el proceso de desarrollo de modelos de redes neuronales profundas.
- **Comunidad activa**: Python tiene una gran comunidad de desarrolladores que contribuyen al desarrollo y mantenimiento de bibliotecas especializadas para el deep learning. Esto significa

que siempre hay una gran cantidad de recursos disponibles en línea para ayudar a los desarrolladores a resolver problemas.

1.5.2 Algunas bibliotecas populares de Python para deep learning:

- **TensorFlow**: Es ampliamente considerada como una de las mejores bibliotecas de Python para aplicaciones de deep learning. Desarrollado por el equipo Google Brain, proporciona una amplia gama de herramientas flexibles, bibliotecas y recursos comunitarios. Los principiantes y los profesionales por igual pueden usar TensorFlow para construir modelos de deep learning, así como redes neuronales.
- **Pytorch**: Otra de las bibliotecas populares de Python para deep learning es Pytorch, que es una biblioteca de código abierto creada por el equipo de investigación en inteligencia artificial de Facebook en 2016. PyTorch le permite llevar a cabo muchas tareas y es especialmente útil para aplicaciones de deep learning como NLP y visión por computadora.
- **NumPy**: Es una biblioteca popular utilizada en la computación científica y el análisis numérico en Python. Proporciona soporte para matrices multidimensionales y funciones matemáticas.
- **Scikit-Learn**: Es una biblioteca popular utilizada en el aprendizaje automático en Python. Proporciona herramientas simples y eficientes para la minería y análisis de datos.
- **SciPy**: Es una biblioteca popular utilizada en la computación científica en Python. Proporciona soporte para funciones matemáticas avanzadas y algoritmos numéricos.
- **Pandas**: Es una biblioteca popular utilizada en la manipulación y análisis de datos en Python. Proporciona estructuras de datos flexibles y herramientas para trabajar con datos.
- **Theano**: Es una biblioteca popular utilizada en la definición, optimización y evaluación eficiente de expresiones matemáticas que involucran matrices multidimensionales.
- **Keras**: Es una biblioteca popular utilizada en la construcción rápida y sencilla de modelos de redes neuronales profundas.

Librería/Framework	Versión	Descripción
TensorFlow	2.15	TensorFlow es una biblioteca de Python ampliamente considerada como una de las mejores para aplicaciones de aprendizaje profundo. Fue desarrollada por el equipo de Google Brain y proporciona una amplia gama de herramientas flexibles, bibliotecas y recursos de la comunidad.
Keras	2.15.0	Keras es una biblioteca de código abierto que proporciona una interfaz de Python para redes neuronales artificiales. Keras actúa como una interfaz para la biblioteca TensorFlow.
PyTorch	2.1	PyTorch es una de las bibliotecas de Python más populares para el aprendizaje profundo, es una biblioteca de código abierto creada por el equipo de investigación de IA de Facebook en 2016.
Theano	1.0.5	Theano es una biblioteca de Python que te permite definir, optimizar y evaluar eficientemente expresiones matemáticas que involucran arrays multidimensionales. Está construido sobre NumPy.
Scikit-learn	1.3.2	Scikit-learn es una API de aprendizaje profundo escrita en Python, que se ejecuta en la plataforma de aprendizaje automático TensorFlow. Fue desarrollado con un enfoque en permitir una experimentación rápida y proporcionar una experiencia de desarrollador agradable.
Pandas	2.1.3	Pandas es un paquete de Python que proporciona estructuras de datos rápidas, flexibles y expresivas diseñadas para hacer que trabajar con datos “relacionales” o “etiquetados” sea fácil e intuitivo.
NumPy	1.26.2	NumPy es el paquete fundamental para la computación científica con Python.
Matplotlib	1.26.2	Matplotlib es una biblioteca completa para crear visualizaciones estáticas, animadas e interactivas en Python.

Tabla 1. Descripción de librerías y sus respectivas versiones

1.5.3 IDE’s para deep learning:

PyCharm: Es una IDE popular para Python que admite el desarrollo de aplicaciones de aprendizaje automático y deep learning. PyCharm proporciona herramientas para la depuración, el análisis de código y la refactorización. También tiene una amplia gama de complementos y bibliotecas que pueden ayudar en el desarrollo de aplicaciones de aprendizaje automático.

Spyder: Es una IDE de código abierto que se enfoca en la ciencia de datos y el análisis numérico. Spyder proporciona herramientas para la depuración, la edición de código y la visualización de datos. También tiene una amplia gama de bibliotecas preinstaladas para análisis de datos.

Jupyter Notebook: Es una aplicación web que permite crear y compartir documentos que contienen código, ecuaciones, visualizaciones y texto narrativo. Jupyter Notebook es una herramienta popular para el aprendizaje automático y el análisis de datos debido a su capacidad para integrar código, texto y visualizaciones en un solo documento.

Visual Studio Code: Es un editor de código fuente desarrollado por Microsoft que admite múltiples lenguajes de programación, incluido Python. Visual Studio Code proporciona herramientas para la depuración, la refactorización y la integración con Git.

Google Colaboratory: Es un entorno gratuito basado en la nube que permite escribir, ejecutar y compartir código en Python. Google Colaboratory es una herramienta popular para el aprendizaje automático debido a su capacidad para ejecutar código en la nube sin necesidad de configurar un entorno local.

1.5.4 IDE’s propuestos a utilizar:

Para el desarrollo de este proyecto utilizaremos **Google Colaboratory** no tiene una “versión” en el sentido tradicional, ya que es un servicio en línea y se actualiza continuamente.

Google Colaboratory, también conocido como Colab, es un cuaderno basado en la nube que te permite escribir y ejecutar código Python en tu navegador. Proporciona una amplia gama de herramientas y recursos para científicos de datos, incluyendo conjuntos de datos listos para usar, plantillas y funciones de colaboración. Colab es una excelente herramienta para científicos de datos que desean escribir rápidamente código y análisis para proyectos de datos sin necesidad de configuración local.

Algunas de las características de Colab incluyen:

Acceso gratuito a GPU: Colab proporciona acceso gratuito a GPU a los usuarios, lo que se puede utilizar para acelerar el entrenamiento de modelos de deep learning.

Fácil compartición: Puedes compartir fácilmente tus cuadernos de Colab con compañeros de trabajo o amigos, lo que les permite comentarlos o incluso editarlos.

Integración con Google Drive: Los cuadernos que creas en Colab se almacenan en tu cuenta de Google Drive, lo que te permite acceder a ellos desde cualquier lugar.

No requiere configuración: Colab no requiere ninguna configuración adicional para empezar a trabajar. Todo lo que necesitas es una cuenta de Google y un navegador web.

Figura 8. Logo del entorno de desarrollo Google Colab

Acceso a bibliotecas populares: Colab viene preinstalado con muchas bibliotecas populares para el aprendizaje automático y la ciencia de datos, como TensorFlow, Keras, Pandas y NumPy.

Conclusiones del capítulo.

En este capítulo se realizó una revisión de los modelos de Deep learning para la detección de fraude en transacciones bancarias, donde se encontraron variantes de modelos para este propósito.

Se definió KDD como metodología para la Mineria de datos. Así como se definió el lenguaje de programación **Python 3.10** para el desarrollo y obtención de los resultados junto a las librerías **Keras** y **TensorFlow** y **Google Colab** como IDE.

Estas herramientas y tecnologías forman parte de un conjunto diverso y en constante evolución, que permite a los profesionales de la MD abordar una amplia gama de problemas y desafíos en el análisis de datos. En el próximo capítulo se abarcará la metodología KDD secuencialmente atendiendo a los pasos definidos anteriormente.

CAPITULO 2: PREPARACION PARA EL PROCESO DE MINERIA DE DATOS

La minería de datos es un proceso poderoso que tiene el potencial de extraer conocimientos valiosos de grandes conjuntos de datos. Sin embargo, antes de que podamos comenzar a minar estos datos, es esencial prepararlos adecuadamente para garantizar la precisión y la eficacia de nuestros resultados.

En este capítulo, exploraremos en profundidad la fase de preparación en el proceso de minería de datos. Esta fase es crucial, ya que los datos sin procesar a menudo están incompletos, ruidosos e inconsistentes. La preparación de los datos implica una serie de pasos que incluyen la limpieza de los datos, la integración de los datos, la selección de los datos y la transformación de los datos.

2.1 Entendimiento, conocimientos previos e identificación de la meta.

En este paso se desarrolla un entendimiento de la aplicación de dominio, los conocimientos previos y la identificación de la meta del proceso de KDD desde el punto de vista del cliente.

Para lograr un correcto desarrollo y comprensión del dominio de la aplicación, aprender los conocimientos previos relevantes e identificar los objetivos del usuario final partimos del conjunto de informaciones que dan entrada a este paso de la secuencia KDD.

Entrada:

- Problema a resolver: ¿Cómo automatizar la detección de fraude bancario?
- Objetivo general: Desarrollar modelos de DL que permitan detectar fraudes es transacciones bancarias.

Objetivo del negocio: Detectar fraude en transacciones bancarias, criterio de éxito cuantitativo: “Número de detecciones de fraude”.

Situación actual: La detección del fraude es realizada de manera manual mediante el estudio de los patrones de comportamiento de cada usuario en un determinado período de tiempo.

Con la aplicación de la minería de datos a este problema se descubre información que no se esperaba obtener, las combinaciones de distintas técnicas otorgan efectos inesperados que se transforman en un valor añadido a la empresa. Enormes bases de datos pueden ser analizadas mediante la tecnología de minería de datos. Los resultados son fáciles de entender: personas sin un conocimiento previo en ingeniería informática pueden interpretar los resultados con sus propias ideas. Contribuye a la toma de decisiones tácticas y estratégicas para detectar la información clave. Los modelos son probados y comprobados usando técnicas estadísticas antes de ser usados, para que las predicciones que se

obtienen sean confiables y válidas. En su mayoría, los modelos se generan, construyen y entrenan de manera rápida.

Objetivo de la minería de datos: Determinar un método de clasificación binaria mediante el uso de patrones de comportamiento de los clientes respecto a su capacidad de cometer fraude bancario.

Plan del proyecto:

La última tarea de esta fase tiene como objetivo desarrollar el plan de proyecto considerando los pasos que se deben seguir y los métodos por emplear en cada paso.

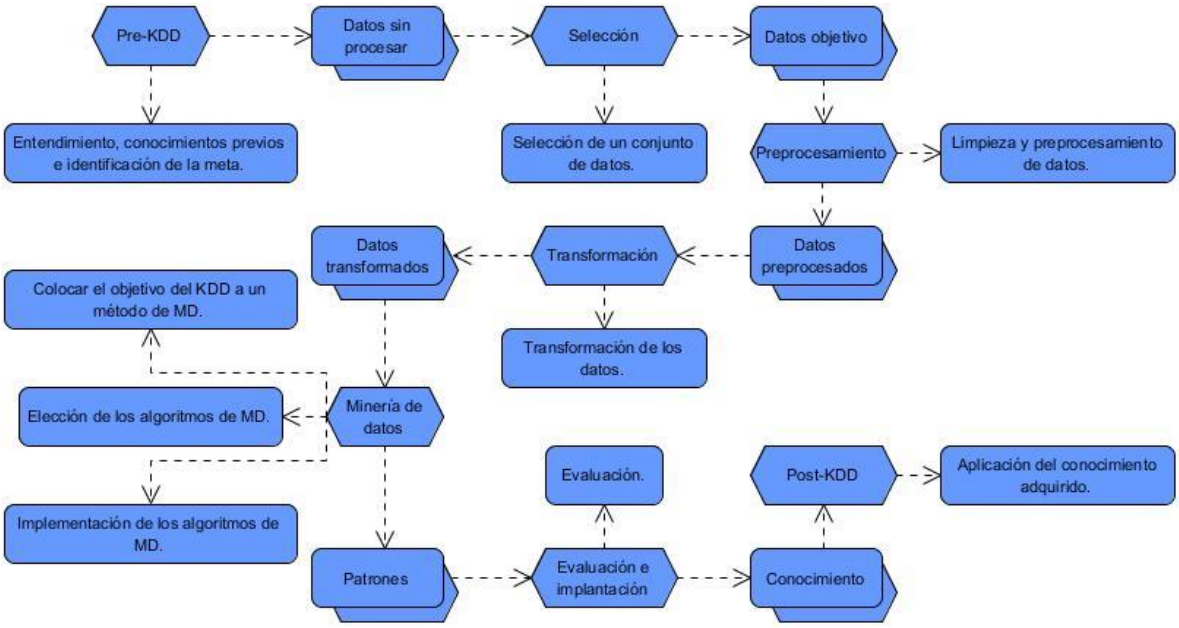

Figura 9. Metodología que guía la realización del proyecto

Salida: Comprensión del problema.

2.2 Etapa de selección.

En la etapa de selección, una vez identificado el conocimiento relevante y prioritario y definidas las metas del proceso KDD, desde el punto de vista del usuario final, se crea un conjunto de datos objetivo, seleccionando todo el conjunto de datos o una muestra representativa de este, sobre el cual se realiza el proceso de descubrimiento. La selección de los datos varía de acuerdo con los objetivos del negocio.

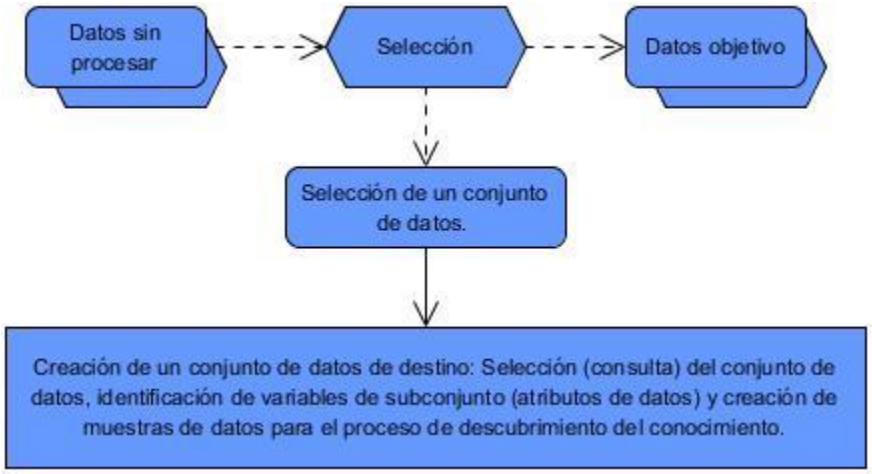

Figura 10. Fase de Selección

Conjunto de datos:

El conjunto de conjuntos de datos sobre fraude de cuentas bancarias (BAF) se publicó en NeurIPS 2022 y comprende un total de 6 conjuntos de datos tabulares sintéticos diferentes sobre fraude de cuentas bancarias. BAF es un banco de pruebas realista, completo y sólido para evaluar métodos nuevos y existentes en ML y ML justo, ¡y el primero de su tipo!

Este conjunto de conjuntos de datos es: realista, basado en un conjunto de datos del mundo real actual para la detección de fraude;

Sesgado, cada conjunto de datos tiene distintos tipos controlados de sesgo;

Desequilibrado, este entorno presenta una prevalencia extremadamente baja de clase positiva;

Dinámico, con datos temporales y cambios de distribución observados;

Preservación de la privacidad, para proteger la identidad de los solicitantes potenciales, hemos aplicado técnicas de privacidad diferencial (adición de ruido), codificación de características y entrenado un modelo generativo (CTGAN).

Cada conjunto de datos se compone de: un millón de casos; 30 funciones realistas utilizadas en el caso de uso de detección de fraude; una columna de "mes", que proporciona información temporal sobre el conjunto de datos; atributos protegidos (grupo de edad, situación laboral y % de ingresos).

Conjunto de datos sobre fraude en cuentas bancarias (NeurIPS 2022) fue proporcionado por **Kaggle**.

Se puede obtener el conjunto de datos en: https://www.kaggle.com/datasets

Salida: Datos de destino / conjunto de datos.

2.3 Etapa de preprocesamiento/limpieza.

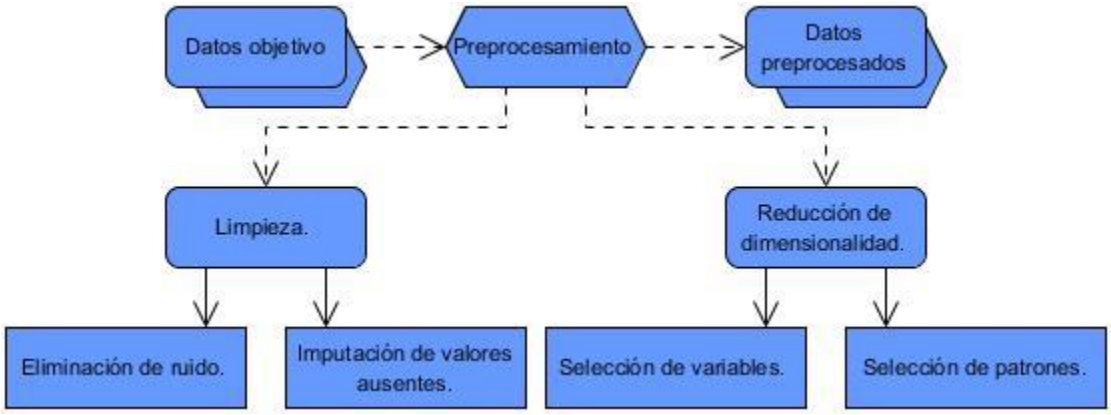

Figura 11. Fase de Procesamiento

En la etapa de preprocesamiento/limpieza (data cleaning) se analiza la calidad de los datos, se aplican operaciones básicas como la remoción de datos ruidosos, se seleccionan estrategias para el manejo de datos desconocidos (missing y empty), datos nulos, datos duplicados y técnicas estadísticas para su reemplazo.

En esta etapa, es de suma importancia la interacción con el usuario o analista. Los datos ruidosos (noisy data) son valores que están significativamente fuera del rango de valores esperados; se deben principalmente a errores humanos, a cambios en el sistema, a información no disponible a tiempo y a fuentes heterogéneas de datos.

Los datos desconocidos empty son aquellos a los cuales no les corresponde un valor en el mundo real y los missing son aquellos que tienen un valor que no fue capturado. Los datos nulos son datos desconocidos que son permitidos por los sistemas gestores de bases de datos relacionales (sgbdr).

En el proceso de limpieza todos estos valores se ignoran, se reemplazan por un valor por omisión, o por el valor más cercano, es decir, se usan métricas de tipo estadístico como media, moda, mínimo y máximo para reemplazarlos.

```
[ ]  #       sin duplicados y sin campos vacios
     df_sin_duplicados=df.drop_duplicates()
     df=df_sin_duplicados.dropna()
```

Figura 12. Fragmento de código para la limpieza

Salida: Datos preprocesados.

2.4 Etapa de transformación/reducción

En la etapa de transformación/reducción de datos, se buscan características útiles para representar los datos dependiendo de la meta del proceso. Se utilizan métodos de reducción de dimensiones o de transformación para disminuir el número efectivo de variables bajo consideración o para encontrar representaciones invariantes de los datos.

Los métodos de reducción de dimensiones pueden simplificar una tabla de una base de datos horizontal o verticalmente. La reducción horizontal implica la eliminación de tuplas idénticas como producto de la sustitución del valor de un atributo por otro de alto nivel, en una jerarquía definida de valores categóricos o por la discretización de valores continuos (por ejemplo, edad por un rango de edades).

La reducción vertical implica la eliminación de atributos que son insignificantes o redundantes con respecto al problema, como la eliminación de llaves, la eliminación de columnas que dependen funcionalmente (por ejemplo, edad y fecha de nacimiento). Se utilizan técnicas de reducción como agregaciones, compresión de datos, histogramas, segmentación, discretización basada en entropía, muestreo, entre otras.

Se realizó la división de los datos en los conjuntos de entrenamiento y prueba, en un 80% y 20% respectivamente.

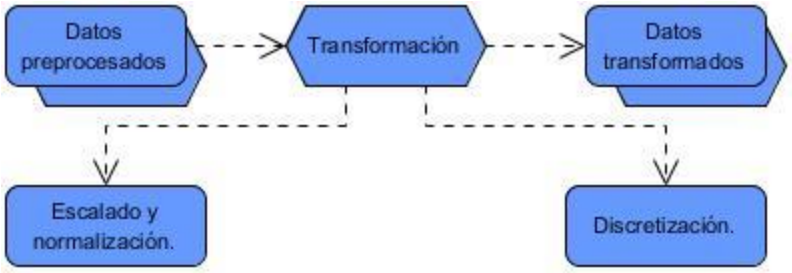

Figura 13. Fase de Transformación

```
[ ]
    colunmas_a_normalizar=['name_email_similarity', 'prev_address_months_count',
                           'current_address_months_count', 'customer_age', 'days_since_request',
                           'intended_balcon_amount', 'zip_count_4w', 'velocity_6h', 'velocity_24h',
                           'velocity_4w', 'bank_branch_count_8w', 'date_of_birth_distinct_emails_4w',
                           'credit_risk_score', 'bank_months_count', 'proposed_credit_limit',
                           'session_length_in_minutes', 'x1', 'x2']
    scaler=MinMaxScaler()
    df[colunmas_a_normalizar]=scaler.fit_transform(df[colunmas_a_normalizar])
```

Figura 14. Fragmento del código para transformar los datos

Salida: datos transformados.

2.5 Etapa de minería de datos

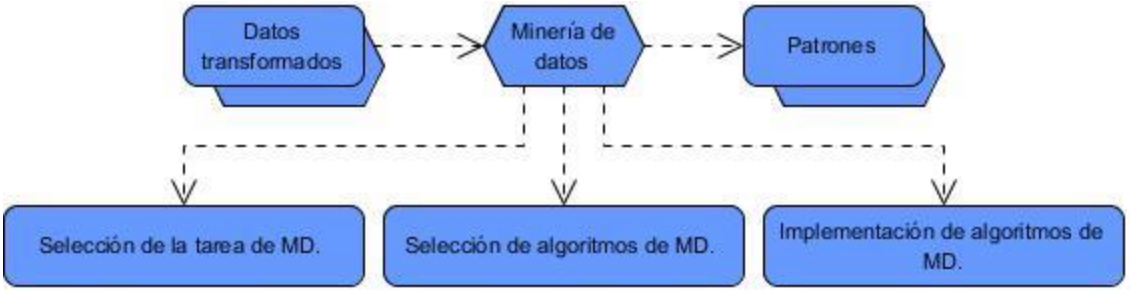

Figura 15. Fase de aprendizaje

Selección de la tarea de minería de datos: Decisión sobre qué métodos aplicar para la clasificación, el agrupamiento, la regresión u otra tarea.

En función de la tarea que realizan, los algoritmos de DL se pueden dividir en algoritmos de clasificación, regresión, agrupación y asociación:

- Algoritmos de clasificación: se utilizan cuando la etiqueta toma valores discretos dentro de un conjunto finito de resultados. La clasificación puede ser binaria o múltiple.
- Algoritmos de regresión: su objetivo es establecer una relación entre un cierto número de características y una variable objetivo continua.
- Algoritmos de agrupación (clustering): es un procedimiento de agrupación de una serie de datos de acuerdo con un criterio, por lo general distancia o similitud entre casos.
- Algoritmos de asociación: los algoritmos de reglas de asociación tienen como objetivo encontrar relaciones dentro un conjunto de transacciones, en concreto, ítems o atributos que tienden a ocurrir de forma conjunta.

Debido al enfoque del proyecto centrado en la detección de fraude en transacciones bancarias y al hecho de que la predicción es a veces referida como una minería de datos supervisada, el método propuesto a usar en el presente proyecto es la clasificación. Usada en varias tareas dentro de las que se encuentra el fraude financiero. Específicamente la clasificación binaria ya que el conjunto de datos se encuentra previamente etiquetado en 0 (transacción normal) y 1 (transacción fraudulenta).

Resultado: Método de clasificación binaria.

Selección de algoritmos de minería de datos: La selección de algoritmos de minería de datos es un paso crucial en el proceso de minería de datos. El objetivo de este paso es seleccionar el método adecuado para la búsqueda de patrones que haga coincidir los métodos con el objetivo del proceso y decida sobre los modelos apropiados y sus parámetros. La elección del algoritmo de minería de datos

correcto es importante porque puede afectar significativamente la calidad de los resultados de la minería de datos. Existen varios factores que deben considerarse al seleccionar un algoritmo de minería de datos, como la naturaleza de los datos, el tamaño del conjunto de datos, el tipo de patrones que se buscan y el objetivo del proceso de minería de datos.

Implementación de los algoritmos de MD:

La implementación de algoritmos de minería de datos es un paso importante en el proceso de minería de datos. En este caso, se implementó un modelo personalizado Autoencoder y una Red Neuronal Profunda. El Autoencoder es una arquitectura de red neuronal que se utiliza para la reducción de dimensionalidad y la detección de anomalías. Por otro lado, las Redes Neuronales Profundas son un tipo de red neuronal que se utiliza para problemas de clasificación y regresión.

La elección de estos modelos específicos puede haber sido influenciada por la naturaleza de los datos, el tamaño del conjunto de datos, el tipo de patrones que se buscan y el objetivo del proceso de minería de datos.

Resultado: Patrones.

Conclusiones del capítulo

Se definieron las metas y procedimientos a utilizar para poder guiar la experimentación según la secuencia de la metodología KDD. Para la obtención de los resultados se caracteriza y selecciona el conjunto de datos sujeto. Se realiza la limpieza y preprocesamiento del conjunto de datos, para definir si la detección de fraude realiza la transformación de los datos, y así comparar las salidas de los modelos con las predicciones.

Se determina la clasificación como método de MD por el enfoque que posee.

Concluido el trabajo realizado en este capítulo es posible proceder a la evaluación de los algoritmos. La evaluación y aplicación del conocimiento conforman los pasos finales de la secuencia KDD aplicada para el desarrollo de este proyecto. Para ello se realizarán experimentos donde sus resultados serán detallados y representados en el capítulo siguiente.

CAPITULO 3 ETAPA DE INTERPRETACIÓN/EVALUACIÓN DE DATOS

El presente capítulo abarca la evaluación de los modelos de DL seleccionados en los pasos anteriores de la secuencia KDD y se definen los parámetros y métricas.

3.1 Evaluación

Inicialmente se prueba cada modelo entrenado, analizando las métricas de **accuracy**, **precisión**, **recall**, **f1-score** y **balanced accuracy score**. Basados en el estudio del estado del arte podemos afirmar que estas son algunas de las métricas más utilizadas en este campo, las cuales permitirán establecer comparaciones con otras investigaciones similares

3.1.1 Modelo ANN

Modelo	**Accuracy**	**Precision**	**ReCall**	**F1-S**	**BAS**
ANN	0.99	0.85	0.33	0.47	0.61

Tabla 2. Métricas del modelo ANN

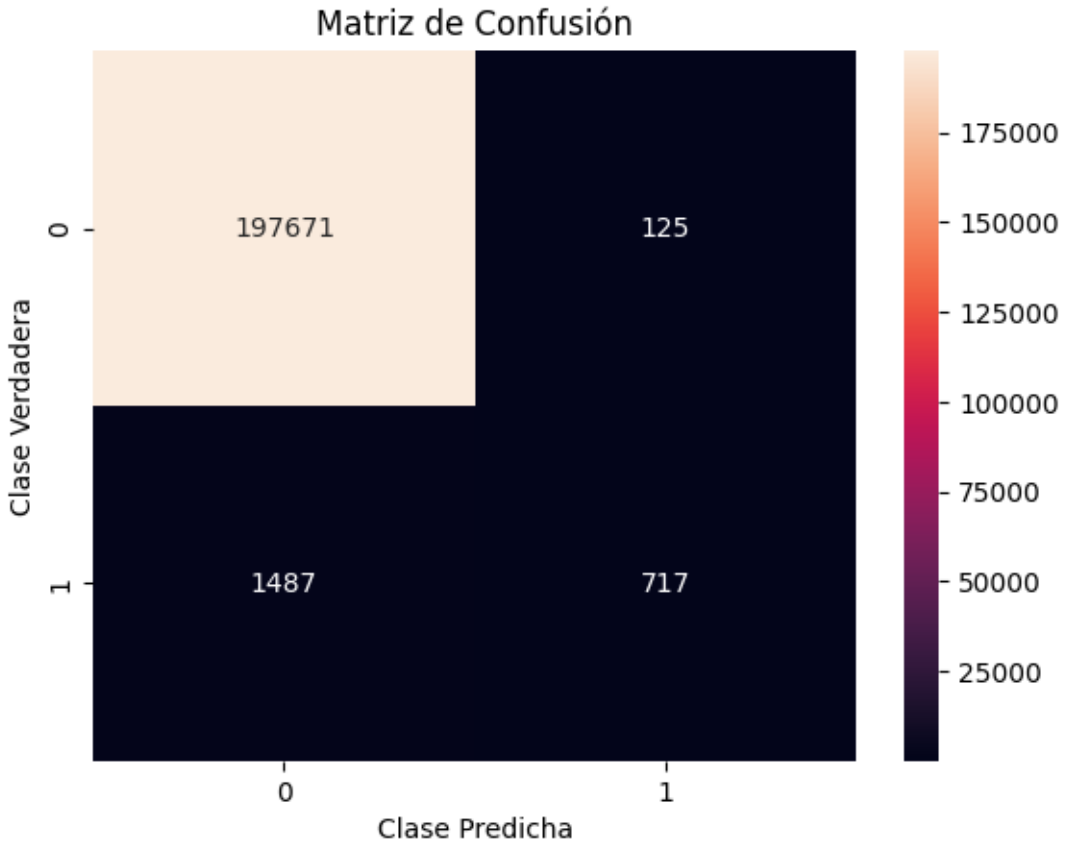

Figura 16. Matriz de Confusión ANN

Se evidencia que el modelo acierta 717 veces, a pesar de presentar una elevada tasa de FP. También se muestran resultados de una elevada exactitud y precisión.

3.1.2 Modelo Autoencoder

Modelo	**Accuracy**	**Precision**	**ReCall**	**F1-S**	**BAS**
AEs	0.94	0.01	0.06	0.02	0.50

Tabla 3. Métricas del modelo AEs

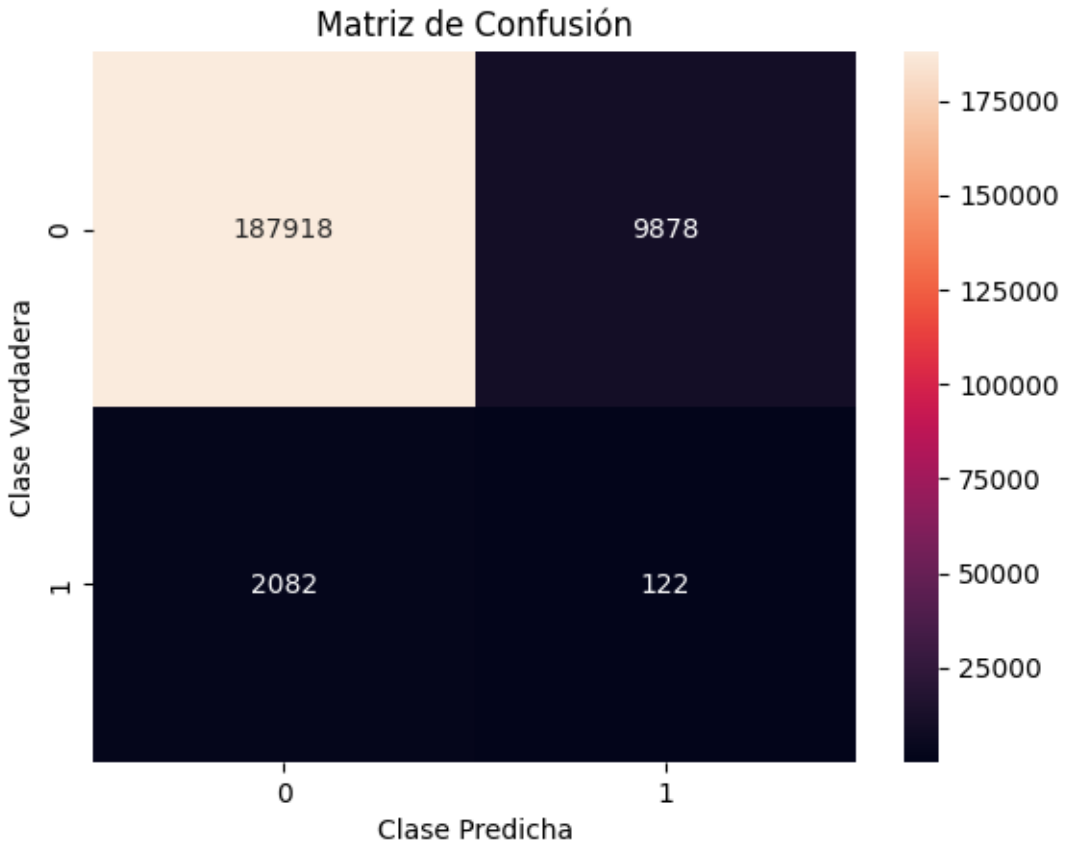

Figura 17. Matriz de Confusión AEs

El modelo acierta 122 muestras y falla en 9878, lo que denota poca efectividad, además de tener un elevado número de FP.

3.1.3 Comparación de los modelos desarrollados

Modelo	**Accuracy**	**Precision**	**ReCall**	**F1-S**	**BAS**
ANN	0.99	0.85	0.33	0.47	0.61
AEs	0.94	0.01	0.06	0.02	0.50

Tabla 4. Comparación entre los modelos implementados

Se establece una comparación de los modelos donde claramente el modelo ANN presenta mejores resultados en todas las métricas, siendo la exactitud la métrica de menor diferencia comparativa entre los modelos planteados con una diferencia del 5%. Se define el modelo ANN como el más óptimo de los modelos implementados.

3.1.4 Comparación con modelos de ML

En este sub epígrafe se utilizan modelos de ML desarrollados en el trabajo de curso" Método para la detección del fraude en transacciones bancarias con escenarios de Flujo de Datos." Con el objetivo de evaluar los resultados obtenidos.

Modelo	**Accuracy**	**Precision**	**ReCall**	**F1-S**	**BAS**
ANN	0.99	0.85	0.33	0.47	0.61
AEs	0.94	0.01	0.06	0.02	0.50
ANN (F)	-	-	1.00	0.00	-
DT	-	-	0.61	0.26	-
GBC	-	-	0.16	0.26	-
IF	-	-	0.61	0.05	-
KNN	-	-	0.00	0.00	-
LR	-	-	0.72	0.05	-
NBC	-	-	0.88	0.00	-
RF	-	-	0.83	0.01	-
SVM	-	-	0.00	0.00	-

Tabla 5. Comparación de los modelos implementados con modelos de ML

ANN = Artificial Neural Network

DT = Decision Trees

GBC = Gradient Boosting

IF = Isolation Forest

KNN = K-Nearest Neighbors

LR = Logistic Regression

NBC = Navie Baiyes Classifier

RF = Random Forest

SVM = Support Vector Machine

En este caso solo se pudo establecer comparación entre las métricas recall y f1-score, el modelo DT muestra resultados de la métrica recall superiores al modelo implementado ANN, pero en la métrica f1-s el modelo implementado ANN obtiene mejores resultados que el modelo DT.

3.1.5 Comparación con otros modelos

Modelo	**Accuracy**	**Precision**	**ReCall**	**F1-S**	**BAS**
ANN	0.99	0.85	0.33	0.47	0.61
AEs	0.94	0.01	0.06	0.02	0.50
RL	0.95	0.05	0.90	0.09	-
RF	0.99	0.11	0.54	0.19	-
NB	0.98	0.06	0.49	0.11	-
ANN (foránea)	0.99	0.35	0.87	0.49	-

Tabla 6. Comparación con otros modelos

RL = Regresión Logística

RF = Random Forest

NB = Navie Bayes

ANN = Artificial Neural Network

En este caso se pueden comparar los modelos implementados teniendo en cuenta un número mayor de métricas. El modelo RL presenta un recall de 0.90 y una exactitud de 0.95, pero la precisión disminuye de manera considerable a 0.05 y el f1-s a 0.09. Teniendo en cuenta la baja precisión que existe en relación con una alta sensibilidad, se puede evidenciar una buena detección de la clase objetivo, pero se está incluyendo muestras de la otra clase.

En el modelo RF se aprecia un aumento de la precisión y disminución de la sensibilidad o recall en comparación con el modelo RL, también disminuye la precisión y el f1-s al compararlo con el modelo ANN implementado.

El modelo NB evidencia una disminución tanto en la precisión como en recall, lo cual indica que dicho modelo no logra identificar la clase correctamente (Londoño Morales & Carmona Mora, 2021).

Las comparaciones anteriores y los resultados expuestos permiten la obtención de las siguientes observaciones en general.

Modelo de mayor Accuracy: ANN, RF y ANN (foránea) presentan una accuracy de 0.99.

Modelo de mayor Precision: ANN.

Modelo de mayor Recall: NBC.

Modelo de mayor F1-S: ANN (foránea)

Teniendo en cuenta todas las métricas se recomienda usar el modelo ANN

Conclusiones del capitulo

Los resultados de las distintas comparaciones permiten definir la efectividad del modelo ANN en la detección de fraude, presentando niveles de estabilidad en sus métricas. El modelo ANN se mantuvo estable dentro de los 5 mejores algoritmos durante la realización de las comparaciones, se logró la implementación de un modelo para la detección de Fraude Bancario, permitiendo su uso y mejoramiento.

CONCLUCIONES FINALES

Se realizó una caracterización y sistematización del estado del arte referido al problema de la detección de fraudes bancarios mediante la implementación de modelos de DL. Se definió la metodología de la MD, la cual, permite una lógica en la obtención de los resultados. Se determinó usar el lenguaje de programación Python en su versión 3.10 para el desarrollo y obtención de los resultados junto a las librerías necesarias y los IDE’s a utilizar.

Se desarrolló un método para algoritmos basados en el aprendizaje supervisado y la computación distribuida para la detección de anomalías en operaciones bancarias. Se validó la solución implementada mediante el diseño de experimentos sobre conjuntos de datos de referencia, comparando los resultados con otros algoritmos.

RECOMENDACIONES

Para futuras implementaciones se recomienda: implementar métodos híbridos de DL utilizando como base los modelos transformer, CNN y Aes. Además, se recomienda desarrollar un nuevo modelo de DL enfocado exclusivamente en la detección de fraude bancario.

REFERENCIAS BIBLIOGRÁFICAS

Abroyan, N. (2017). Neural networks for financial market risk classification. *Frontiers in Signal Processing*, *1*(2), 62-66.

Adewumi, A. O., & Akinyelu, A. A. (2017). A survey of machine-learning and nature-inspired based credit card fraud detection techniques. *International Journal of System Assurance Engineering and Management*, *8*, 937-953.

Alarfaj, F. K., Malik, I., Khan, H. U., Almusallam, N., Ramzan, M., & Ahmed, M. (2022). Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *IEEE Access*, *10*, 39700-39715.

Álvarez, M. (2013). *Python*. https://desarrolloweb.com/home/python

Ameijeiras Sánchez, D., Valdés Suárez, O., & González Diez, H. (2021). Algoritmos de detección de anomalías con redes profundas. Revisión para detección de fraudes bancarios. *Revista Cubana de Ciencias Informáticas*, *15*(4), 244-264.

Arroyo, A. C., Serrano, S. R., & Torres, G. S. (2020). Remoción de lluvia en imágenes por medio de una arquitectura de autoencoder. *Investigación e Innovación en Ingenierías*, *8*(1), 140-167.

Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. *2017 international conference on computing networking and informatics (ICCNI)*, 1-9.

BillWagner. (2023, febrero 15). *Un paseo por C#: Información general*. https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/

Dhankhad, S., Mohammed, E., & Far, B. (2018). Supervised machine learning algorithms for credit card fraudulent transaction detection: A comparative study. *2018 IEEE international conference on information reuse and integration (IRI)*, 122-125.

DOCS, M. W. (2023, julio 18). *Fundamentos de JavaScript—Aprende desarrollo web | MDN*. https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics

Durán Suárez, J. (2017). *Redes neuronales convolucionales en R: Reconocimiento de caracteres escritos a mano*.

Formación. (2022). Máxima Formación. https://www.maximaformacion.es/blog-dat/que-es-r-software/

Goodfellow, P.-A., Mirza, X., & Warde-Farley, O. (2014). Goodfellow I. *Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., Generative adversarial nets, Advances in neural information processing systems*, *27*.

GraphEverywhere, E. (2019, diciembre 23). Fraude Bancario | Que es el fraude bancario y cómo evitarlo. *GraphEverywhere*. https://www.grapheverywhere.com/que-es-el-fraude-bancario-y-como-evitarlo/

Khare, N., & Sait, S. Y. (2018). Credit card fraud detection using machine learning models and collating machine learning models. *International Journal of Pure and Applied Mathematics*, *118*(20), 825-838.

Lin, T.-H., & Jiang, J.-R. (2021). Credit card fraud detection with autoencoder and probabilistic random forest. *Mathematics*, *9*(21), 2683.

LLC. (2022). edX. https://www.edx.org/es/aprende/programacion-c-mas-mas

Londoño Morales, L. M., & Carmona Mora, M. (2021). *Modelos de machine learning para la detección de fraude financiero*.

Ozbayoglu, A. M., Gudelek, M. U., & Sezer, O. B. (2020). Deep learning for financial applications: A survey. *Applied Soft Computing*, *93*, 106384.

Pandey, Y. (2017). Credit Card Fraud Detection using Deep Learning. *International Journal of Advanced Research in Computer Science*, *8*(5).

PricewaterhouseCoopers. (s. f.). *PwC*. PwC. Recuperado 3 de diciembre de 2023, de https://www.pwc.com/gx/en.html

Rajan, R., Rajest, S., & Singh, B. (2021). Spatial data mining methods databases and statistics point of views. *Innovations in Information and Communication Technology Series*, 103-109.

Retomando la bancarización: ¿Cómo marcha el proceso? (+ Video) - Cubadebate. (2023, noviembre 24). Cubadebate - Cubadebate, Por la Verdad y las Ideas. http://www.cubadebate.cu/especiales/2023/11/24/retomando-la-bancarizacion-como-marcha-el-proceso-video/

Rodrigo. (2020). *Detección de anomalías: Autoencoders y PCA*. https://cienciadedatos.net/documentos/52_deteccion_anomalias_autoencoder_pca.html

Sailusha, R., Gnaneswar, V., Ramesh, R., & Rao, G. R. (2020). Credit card fraud detection using machine learning. *2020 4th international conference on intelligent computing and control systems (ICICCS)*, 1264-1270.

ServiceNow. (2022). *¿Qué es la detección de anomalías?* ServiceNow. https://www.servicenow.com/es/products/it-operations-management/what-is-anomaly-detection.html

Services, A. W. (2022). *¿Qué es Java? - Explicación del lenguaje de programación Java - AWS*. Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/java/

Shanmugapriya, P., Shupraja, R., & Madhumitha, V. (2022). Credit Card Fraud Detection System Using CNN. *Int. J. Res. Appl. Sci. Eng. Technol.* https://www.academia.edu/download/82425160/Credit_Card_Fraud_Detection_System_Using_CNN.pdf

Sharma, D., & Lavavanshi, S. (2022). DETECTION OF CREDIT CARD FRAUD USING A NOVEL LSTM, GRU, AND ANN MODELS. *JOURNAL OF OPTOELECTRONICS LASER*, *41*(12).

Strelcenia, E., & Prakoonwit, S. (2023). A Survey on GAN Techniques for Data Augmentation to Address the Imbalanced Data Issues in Credit Card Fraud Detection. *Machine Learning and Knowledge Extraction*, *5*(1), 304-329.

Xu, W., Jang-Jaccard, J., Singh, A., Wei, Y., & Sabrina, F. (2021). Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset. *IEEE Access*, *9*, 140136-140146.

Yee, O. S., Sagadevan, S., & Malim, N. H. A. H. (2018). Credit card fraud detection using machine learning as data mining technique. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *10*(1-4), 23-27.