

ANÁLISIS DE PATRONES Y DETECCIÓN DE ANOMALÍAS EN DATOS ON-CHAIN DE BITCOIN MEDIANTE TÉCNICAS DE MACHINE LEARNING SOBRE UNA ARQUITECTURA BASADA EN RPC

Christian Roberto Tapia Gaibor

Pontificia Universidad Católica del Ecuador

Maestría en Sistemas de Información (Mención Ciencia de Datos)

Tutor de Tesis: Rafael Melgarejo Heredia

Quito, Ecuador

2026-08

Índice General

Contenido

Introducción 6

Planteamiento del Problema y Pregunta Principal 6

Justificación del Proyecto 6

Objetivos de la Investigación 7

Capítulo I: Marco Teórico 8

1.1 Introducción del Capítulo 8

1.2 Estrategia de revisión de literatura 8

1.3 Antecedentes de investigación 9

1.4 Bases conceptuales 10

1.5 Teorías, modelos o enfoques relacionados 11

1.6 Variables, dimensiones o categorías de análisis 12

1.7 Brechas de investigación 12

1.8 Síntesis del marco teórico 13

Capítulo II: Marco Metodológico 14

2.1 Metodología CRISP-DM 14

2.2 Población, Muestra y Fuente de Datos 14

2.3 Variables Principales y Técnicas de recolección 14

2.4 Método Analítico, Modelos y Métricas 15

2.5 Diseño de la Investigación 17

2.6 Arquitectura de la solución 17

Capítulo III: Resultados 26

3.1 Configuración de Hiperparámetros 27

3.1.1. Isolation Forest (IF) 28

3.1.2. K-Means 29

3.1.3. DBSCAN 30

3.2 Resultados de Detección de Anomalías 31

3.3 Resultados de Clustering 32

3.3.1. K-Means 32

3.1.2. DBSCAN 35

3.3 Difusión en Telegram 37

3.7 Benchmark de Tiempos 39

Capítulo IV: Discusión de Resultados 41

4.1 Discusión y Contraste de Hallazgos 41

4.1.1 Discusión de resultados 41

4.1.2 Contraste con estudios previos 43

Conclusiones y Recomendaciones 45

Conclusiones 45

Recomendaciones 45

Referencias Bibliográficas 46

Índice de Tablas

Tabla 1 18
Tabla 2 24
Tabla 3 28
Tabla 4 29
Tabla 5 30
Tabla 6 31
Tabla 7 33
Tabla 8 35
Tabla 9 39
Tabla 10 40

Índice de Figuras

Figura 1 19
Figura 2 20
Figura 3 21
Figura 4 22
Figura 5 34
Figura 6 35
Figura 7 36
Figura 8 37
Figura 9 38

Introducción

Planteamiento del Problema y Pregunta Principal

PREGUNTA PRINCIPAL DE INVESTIGACIÓN:

¿Se puede implementar un sistema independiente para la detección de anomalías y patrones en el mempool de Bitcoin en tiempo real mediante técnicas de Machine Learning basado en un nodo propio y llamadas RPC?

PLANTEAMIENTO DEL PROBLEMA:

Actualmente una parte importante de los servicios disponibles son de plataformas centralizadas que comercializan sus servicios a costos elevados, tienen restricciones a la información en bruto y otras limitaciones. Se requiere disponer de herramientas accesibles, descentralizadas y eficientes para realizar análisis on-chain a la red de Bitcoin.

Existe un vacío en la literatura y la práctica sobre la aplicación de técnicas de Machine Learning no supervisado en la mempool en tiempo real utilizando un nodo podado propio, ya que la mayoría de los estudios previos utilizan datasets históricos estáticos.

Justificación del Proyecto

Al disponer de nodos propios con acceso directo a la información ya es una alternativa viable que permite obtener los datos directo de la fuente con la integridad requerida, menor latencia y un control soberano sobre el procesamiento de la información. Sin embargo, una alternativa más viable puede ser el utilizar para el análisis un nodo podado de Bitcoin, pero este tipo de nodos se han explorado poco en las investigaciones referentes a tráfico on-chain y más aún sobre la mempool en tiempo real.

La aplicación de técnicas de Machine Learning no supervisado es limitada en cuanto a la caracterización de patrones y detección de anomalías en la red de Bitcoin y en especial a los datos obtenidos directamente de un nodo propio. La mayoría de los estudios se han realizados con información estática o datasets históricos, sin tomar en cuenta a la información en tiempo real de la mempool.

Por lo que se ha identificado un vacío tanto en la literatura como en la práctica sobre el desarrollo de una solución de análisis de datos on-chain la red de Bitcoin, junto con la aplicación de técnicas de Machine Learning no supervisado que permitirá alternativas

soberanas para las personas en cuanto a la detección automatizada de patrones y anomalías en la red. Es por ello, que se plantea el desarrollo de este tipo de sistema que aportaría tanto al ámbito investigativo como tecnológico, para futuras investigaciones generando mayor factibilidad en la adquisición de datos en tiempo real, reduciendo considerablemente los costos relacionados con el análisis manual usando herramientas tradicionales, con la posible contribución de desarrollo de aplicaciones en el ámbito de la FinTech.

Objetivos de la Investigación

OBJETIVO GENERAL:

Desarrollar un sistema independiente para la detección de anomalías y patrones en la mempool de la red de Bitcoin en tiempo real mediante técnicas de Machine Learning no supervisado basado en un nodo propio y llamadas RPC.

OBJETIVOS ESPECÍFICOS:

1. Implementar un pipeline de adquisición de datos en tiempo real desde un nodo propio de Bitcoin mediante llamadas RPC para la obtención de información de la mempool.

2. Analizar el flujo de datos on-chain de la mempool mediante el análisis de métricas y patrones obtenidos de dichas transacciones.

3. Aplicar técnicas de Machine Learning no supervisado (K-Means, DBSCAN, Isolation Forest) para la detección de anomalías y patrones dentro de los datos obtenidos desde la mempool.

4. Evaluar la efectividad del modelo desarrollado mediante métricas de desempeño aplicadas a la detección de patrones y anomalías de datos en tiempo real.

Capítulo I: Marco Teórico

1.1 Introducción del Capítulo

En este capítulo se fundamentó teóricamente la investigación realizada mediante la revisión bibliográfica de la literatura científica relacionada con blockchain, Bitcoin, mempool, datos on-chain y las diferentes técnicas de Machine Learning no supervisado. Durante este análisis también se identificó la brecha de investigación que existe entre los conceptos, los antecedentes y los enfoques metodológicos que sustentan al desarrollo de esta propuesta.

En la actualidad, Las redes de blockchain que existen generan cantidades enormes de volúmenes de datos sobre las transacciones, mismas que pueden ser analizadas para su comprensión en cuanto al comportamiento de los usuarios y detectar los patrones más relevantes. Particularmente, la red de blockchain de Bitcoin representa uno de los ecosistemas que más se han estudiado debido a su principal característica de descentralización frente a las demás blockchain, el hecho de que la disponibilidad de sus datos sea pública y la gran relevancia dentro del área de finanzas digitales la vuelven una fuente de datos apetecible. En este sentido, las diferentes técnicas de machine learning pueden ofrecer oportunidades que permitan identificar comportamientos inusuales o anómalos, lo que beneficia en la toma de decisiones basadas en esta evidencia.

1.2 Estrategia de revisión de literatura

La estrategia para la revisión de literatura se la realizó mediante la búsqueda sistemática, tanto de artículos científicos como libros en bases de datos especializadas, priorizando que sean estudios de alto impacto y que conlleve la revisión por pares, también se consideró que las publicaciones estén relacionadas con blockchain, Bitcoin, análisis de datos, mempool y Machine Learning.

Entre las principales fuentes que se consultaron están: IEEE Xplore, Scopus y SciELO. Para la búsqueda de los diferentes artículos científicos se utilizaron combinaciones de términos tanto en inglés como en español, tales como: Bitcoin, Blockchain, Mempool, On-Chain Data, Pruned Node, Machine Learning, Isolation Forest, Clustering, Anomaly Detection y Real-Time Processing.

Los criterios de inclusión que se utilizaron para filtrar las investigaciones publicadas principalmente entre 2020 y 2025, están relacionadas con los términos: análisis de blockchain,

obtención de datos desde nodos, análisis de mempool, aprendizaje automático y detección de anomalías. Además, se excluyeron todos los estudios que se enfocaban exclusivamente en aspectos legales, regulatorios o financieros que no aportaron directamente al problema de investigación a pesar de que estar dentro de los términos buscados.

Como resultado de todo el proceso de búsqueda, se obtuvo una selección de una base documental que contienen artículos internacionales y regionales, los cuáles permitieron construir los antecedentes y fundamentos conceptuales de la investigación.

1.3 Antecedentes de investigación

El crecimiento que ha tenido las tecnologías de blockchain han impulsado el desarrollo de investigaciones que se orientan al análisis de los datos de las transacciones y al beneficio que se puede obtener de la información generada por estas redes descentralizadas. Diversos estudios se han concentrado en la obtención, el almacenamiento y el procesamiento eficiente de los datos de la blockchain. Wang et al. (2022) presentaron una propuesta de un esquema eficiente de almacenamiento que se basó en las características de UTXO, mientras que Matzutt et al. (2021) desarrollaron su propuesta con CoinPrune para reducir significativamente los altos requerimientos de almacenamiento y sincronización de los nodos de Bitcoin.

Con respecto a la adquisición y posterior análisis de los datos de la blockchain, Zheng et al. (2020) desarrollaron XBlock-ETH para realiza la extracción y luego la exploración de la información proveniente de red de Ethereum, demostrando que tiene gran potencial para el análisis los datos on-chain. De igual manera, Jagannath et al. (2021) y Gundlach et al. (2021) utilizaron métricas on-chain para la construcción de modelos predictivos que se basaron en el aprendizaje profundo y que se aplicaron al comportamiento de las criptomonedas.

Por otra parte, el realizar análisis de la mempool ha empezado a recibir atención debido a su importancia dentro de la transmisión y las confirmaciones de las transacciones que circulan por ella. Imtiaz et al. (2021) realizaron el análisis del impacto que tiene la sincronización de mempool y la conectividad de los nodos sobre el rendimiento de la red Bitcoin, mientras que Hossain y Altarawneh (2025) se concentraron en estudiar la relación entre la congestión, las tarifas y la eficiencia de la validación en Ethereum.

En cuanto a la utilización de Machine Learning, las investigaciones recientes han demostrado que la utilidad de técnicas no supervisadas sirve para caracterizar los patrones y detectar los comportamientos anómalos. Vlahavas, Karasavvas y Vakali (2024) aplicaron K-

Means para la identificación de grupos de comportamiento en transacciones de Bitcoin. De una manera complementaria, Yahia et al. (2024) compararon los algoritmos de detección de anomalías como: Isolation Forest, Local Outlier Factor y One-Class Support Vector Machine, en el cual encontrando resultados favorables para el algoritmo LOF.

Finalmente, Hasan et al. (2024) incorporaron a sus investigaciones las técnicas de Inteligencia Artificial Explicable mediante SHAP para interpretar los modelos de detección de anomalías en Bitcoin, mientras que Siddamsetti y Srivenkatesh (2023) exploraron el uso de las técnicas: Isolation Forest, CBLOF y Autoencoders para identificar las posibles actividades de fraudes en los datos de la blockchain.

1.4 Bases conceptuales

Blockchain

La tecnología de Blockchain es un registro distribuido que permite almacenar la información de forma descentralizada, de una manera transparente e inmutable. Su estructura está basada en bloques que están enlazados criptográficamente para garantizar la integridad de la información, reduciendo la necesidad de intermediarios.

Bitcoin

Bitcoin es la aplicación pionera y que tuvo éxito sobre la tecnología blockchain y su operación es mediante una red distribuida o descentralizada de nodos que validan y registran cada transacción. El funcionamiento de bitcoin genera grandes volúmenes de información que es pública y que pueden ser utilizados para realizar análisis y estudios de comportamiento.

Mempool

La mempool es el conjunto de todas las transacciones que están pendientes de confirmación dentro de la red de bitcoin. Mediante su comportamiento podemos ver reflejado: el estado de la congestión, las prioridades de procesamiento y la dinámica temporal de las transacciones antes de que sean minadas e incorporadas a un bloque.

Datos On-Chain

Los datos on-chain es toda la información que corresponde a los registros directos en la blockchain, la cual incluye: todos los bloques, las transacciones, las direcciones, los valores transferidos y las métricas derivadas del comportamiento de la red.

Machine Learning no supervisado

El aprendizaje no supervisado es la agrupación de las técnicas capaces de identificar a los patrones, las relaciones y las anomalías sin necesidad de etiquetas previas. Estas técnicas pueden resultar especialmente útiles cuando se dispone para el análisis de grandes volúmenes de datos y se desconocen previamente las categorías existentes, es decir, no existen etiquetas en los datos.

Core

Bitcoin Core es la implementación oficial del software del protocolo Bitcoin que permite la validación de forma independiente de bloques y de transacciones, mantiene la sincronización con la red y puede acceder a la información mediante la interfaz Remote Procedure Call (RPC).

Pruned Node

El nodo en modo podado es el cual conserva únicamente los bloques más recientes y necesarios para la validación, el cual elimina automáticamente todos los bloques antiguos con el fin de reducir significativamente los requerimientos del almacenamiento, sin que afecte la capacidad de captura y análisis en tiempo real de la información de la mempool que es requerida para el estudio.

1.5 Teorías, modelos o enfoques relacionados

Esta investigación se fundamenta en tres enfoques principales.

El primer enfoque corresponde a los sistemas distribuidos, la cual proporciona la base conceptual para poder comprender el funcionamiento en conjunto de blockchain y Bitcoin como redes descentralizadas.

El segundo enfoque corresponde al procesamiento de los datos en tiempo real, debido a que la mempool genera en cada instante información dinámica de todas las transacciones que pueden ser capturadas y analizadas continuamente.

El tercer enfoque corresponde al aprendizaje automático no supervisado, el cual es utilizado para descubrir los patrones ocultos y detectar las anomalías sin la necesidad de disponer de los datos previamente etiquetados.

Dentro del tercer enfoque destacan los algoritmos como: K-Means, Isolation Forest, Local Outlier Factor y Autoencoders, los cuales se han utilizado ampliamente en

investigaciones recientes relacionadas con criptomonedas y la detección de los comportamientos anómalos.

1.6 Variables, dimensiones o categorías de análisis

Las investigaciones consideran como principales categorías de análisis a los datos provenientes de la red Bitcoin los cuales son capturados desde un nodo propio de bitcoin.

Entre las variables potenciales para la investigación basadas en variables utilizadas en otros estudios para el análisis de anomalías y clusterización se encuentranas siguientes: el número de entradas y las salidas por transacción, el tamaño de la transacción, el monto transferido, las comisiones, el tiempo de permanencia en mempool y la frecuencia de actividad observada.

A partir de estas variables mencionadas se busca identificar los patrones de comportamiento y las posibles anomalías mediante las técnicas de Machine Learning no supervisado. Los resultados que se obtienen serán comunicados a través de los mecanismos que se han automatizado para la visualización y las notificaciones.

1.7 Brechas de investigación

En la revisión de literatura se evidencio importantes avances en el análisis de blockchain, el uso de las métricas on-chain y la aplicación de las técnicas de aprendizaje automático no supervisado para la predicción y la detección de las anomalías.

Sin embargo, la mayoría de los estudios que se han realizado se centran en datos históricos que previamente han sido almacenados en aplicaciones financieras que tienen relación con la predicción de precios. Asimismo, las investigaciones realizadas sobre la mempool suelen enfocarse en problemas de sincronización, la propagación o la eficiencia de la red, sin integrar en estos estudios las técnicas de Machine Learning para el análisis automatizado de los patrones transaccionales en tiempo real.

En el análisis de los nodos propios de Bitcoin se logró identificar una oportunidad de investigación en la integración de los datos y las técnicas de aprendizaje no supervisado para detectar los patrones y las anomalías dentro de la mempool de Bitcoin. Se identificó también que existe una limitada exploración de los mecanismos de comunicación automatizadas que permitan difundir los resultados que se han obtenido a los usuarios finales.

1.8 Síntesis del marco teórico

La literatura que ha sido revisada demuestra que la blockchain y Bitcoin constituyen uno de los entornos ricos en información y que es susceptible de análisis mediante las técnicas avanzadas de ciencia de datos. Los estudios existentes han aportado los fundamentos sobre: el almacenamiento eficiente, la captura de datos on-chain, el comportamiento de la mempool y la aplicación de algoritmos de Machine Learning.

No obstante, permanecen las oportunidades para la integración de estos componentes dentro de una solución que unifique y permita capturar datos desde los nodos propios, permita el análisis en tiempo real y permita la detección de comportamientos anómalos mediante técnicas de aprendizaje no supervisado.

La presente investigación pretende la contribución al conocimiento existente mediante el desarrollo de una solución con enfoque orientado al análisis automatizado de la red de Bitcoin y a la comunicación práctica de los resultados que se han obtenido.

Capítulo II: Marco Metodológico

2.1 Metodología CRISP-DM

La investigación se desarrollará siguiendo la metodología CRISP-DM (Cross Industry Standard Process for Data Mining), debido a su amplia utilización en proyectos de Ciencia de Datos. Las fases consideradas serán: comprensión del problema, comprensión de los datos, preparación de los datos, modelado, evaluación y presentación de resultados. Esta metodología permitirá estructurar el proceso de adquisición, análisis y detección de anomalías sobre datos provenientes de la mempool de Bitcoin.

2.2 Población, Muestra y Fuente de Datos

Población: Todas las transacciones registradas en la mempool de la red Bitcoin.

Muestra: Conjunto de snapshots de la mempool y transacciones asociadas capturadas mediante un nodo propio durante un período continuo de observación de al menos 10 días.

Fuente de datos: Registros y datos obtenidos directamente desde la red Bitcoin a través de llamadas RPC realizadas a un nodo propio configurado en modo podado.

2.3 Variables Principales y Técnicas de recolección

Las variables seleccionadas corresponden a los atributos estructurales, los económicos y los temporales de las transacciones presentes en la mempool de Bitcoin, se obtuvieron directamente mediante la interfaz RPC de un nodo propio. Se seleccionaron en base a la evidencia de su utilización en investigaciones previas relacionadas con análisis on-chain y detección de anomalías; y también en base a la compatibilidad con algoritmos de aprendizaje no supervisado basados en variables numéricas. Dentro de todas las variables que se pueden obtener de la mempool se descartaron las variables identificadoras, las categóricas o altamente correlacionadas con las variables ya seleccionadas.

Variables para identificar patrones y anomalías en las transacciones de Bitcoin:

- Fee rate (sat/vB)
- Tamaño de transacción
- Número de inputs
- Número de outputs

- Valor transferido
- Tiempo de permanencia en mempool

Variables para el análisis del comportamiento de la mempool

- Tamaño de la mempool
- Número de transacciones pendientes
- Cantidad de transacciones confirmadas por bloque
- Tiempo promedio de permanencia

Estas variables seleccionadas para el análisis se encuentran respaldadas por las investigaciones sobre el comportamiento de las transacciones de Bitcoin y su mempool. Vlahavas et al. (2024) emplea características transaccionales como la tasa de comisión, el número de inputs, el número de outputs y el valor transferido para identificar patrones mediante técnicas de aprendizaje no supervisado, mientras que Zhang et al. (2025) también emplea a las variables asociadas con inputs, outputs y valores transaccionales para explorar modos de comportamiento en Bitcoin. Gundlach et al. (2021) analizan la relación entre el estado de la mempool y los tiempos de confirmación de las transacciones, lo que sustenta la decisión en esta investigación sobre considerar variables como el tamaño de la mempool y el tiempo de permanencia. En definitiva, los artículos científicos analizados permitieron fundamentar la selección de las variables sobre transacciones y variables temporales utilizadas en esta investigación para la identificación de patrones y anomalías.

Técnica de recolección: Se obtendrá los datos mediante la extracción automatizada de datos mediante llamadas RPC que se realizará a un nodo propio de Bitcoin, directamente a la mempool.

2.4 Método Analítico, Modelos y Métricas

Método analítico: Se utilizará el método de análisis exploratorio de datos y Machine Learning no supervisado.

En la investigación de Vlahavas et al. (2024), aplicaron técnicas basadas en K-Means para agrupar transacciones de Bitcoin e identificar diferentes comportamientos transaccionales. Incorporaron también **DBSCAN** como un enfoque basado en densidad, debido a que permite identificar agrupamientos sin requerir establecer previamente el número de clústeres. De esta

manera, K-Means permite analizar estructuras de agrupamiento basadas en proximidad a centroides, mientras que DBSCAN proporciona una perspectiva basada en densidad.

Las métricas que se usaron para la evaluación de los resultados obtenidos mediante el entrenamiento de los modelos K-Means y DBSCAN son: Silhouette, Davies-Bouldin y Calinski-Harabasz. Esta elección tiene como finalidad el analizar la calidad de los agrupamientos desde varias perspectivas. El uso de la métrica Silhouette está respaldada por Vlahavas et al. (2024), quienes la emplearon para evaluar los grupos obtenidos durante el análisis no supervisado de transacciones de la mempool de Bitcoin. El uso de estas métricas permite que la evaluación interna sea más robusta tanto para K-Means como para DBSCAN, evitando determinar la calidad del agrupamiento a partir de un único criterio.

Clustering:

- K-Means
- DBSCAN

Métricas:

- Silhouette Score
- Davies-Bouldin Index

Las investigaciones sobre transacciones de criptomonedas realizadas por de Yahia et al. (2024) y Hasan et al. (2024), utilizan los enfoques para identificar observaciones que se pueden considerar como anómalos a los algoritmos Isolation Forest (IF). Motivo por el cual se seleccionaron para utilizar en esta investigación.

Las métricas de evaluación de los modelos Isolation Forest (IF) que se consideraron adecuadas son las siguientes: anomaly scores, la cantidad y proporción de observaciones clasificadas como anómalas y la consistencia de los resultados entre diferentes configuraciones del modelo. Estas técnicas son respaldadas por Hasan et al. (2024) quienes evidenciaron la importancia de evaluar los comportamientos identificados como anómalos dentro del análisis de transacciones blockchain.

Anomalías:

- Isolation Forest

Métricas:

- Proporción de outliers detectados

- Consistencia entre modelos
- Estabilidad temporal de anomalías
- Revisión manual de casos representativos

2.5 Diseño de la Investigación

La presente investigación adopta un enfoque cuantitativo debido a que se basa en la recopilación, el procesamiento y el análisis de datos numéricos obtenidos desde la red de Bitcoin mediante un nodo propio. A partir de estos datos se obtendrán diferentes métricas que están relacionadas con la actividad de la mempool, las cuales serán analizadas mediante las técnicas de Machine Learning no supervisado para identificar los posibles patrones y las anomalías. Además, los resultados obtenidos serán evaluados mediante métricas cuantificables que permitan medir el desempeño de los modelos desarrollados.

La investigación presenta un diseño exploratorio debido a que se pretende el análisis y lograr comprender el comportamiento de los datos on-chain que provienen de la mempool de Bitcoin en tiempo real, siendo este un ámbito en el que todavía se presenta pocas investigaciones, especialmente cuando se obtienen los datos directamente desde un nodo propio y estos son procesados mediante las técnicas de Machine Learning no supervisado. Por lo tanto, el estudio pretende la identificación de los patrones, las relaciones y las posibles anomalías que permitan generar el conocimiento sobre este tipo de datos.

2.6 Arquitectura de la solución

2.6.1. Visión General de la Arquitectura del Sistema

El diseño de la solución está basado en software modular, desacoplado y de alto rendimiento con el propósito de resolver la dependencia de proveedores centralizados y permitir la detección de anomalías y patrones en tiempo real sobre la mempool de Bitcoin.

La aplicación integra una arquitectura hexagonal la cual integra un nodo de Bitcoin Core en modo podado (pruned node) que funciona con un motor de extracción basado en llamadas JSON-RPC y tiene un sistema de almacenamiento relacional basado en PostgreSQL. El núcleo de la solución es un pipeline de Machine Learning (ML) no supervisado (K-Means, DBSCAN, Isolation Forest).

Las tecnologías que se utilizaron en el desarrollo de la solución para la conexión RPC al nodo Bitcoin y entrenamiento de modelos con información recopilada de la mempool de Bitcoin es:

Tabla 1

Versiones principales utilizadas en el desarrollo de la solución

Tecnología	Versión	Función principal
Python	3.13	Lenguaje principal de la Factory
Scikit-Learn	1.9.0	Entrenamiento y evaluación local de modelos ML
PySpark / Apache Spark	3.5.9	Procesamiento y entrenamiento distribuido
SynapseML	1.1.3	Isolation Forest sobre infraestructura Spark
Py4J	0.10.9.9	Comunicación entre Python y la JVM utilizada por Spark
NumPy	2.5.0	Operaciones numéricas
Pandas	3.0.4	Manipulación y preparación de datos
SciPy	1.18.0	Operaciones científicas utilizadas por componentes de ML
FastAPI	0.141.1	API utilizado para comunicación con componentes de la Factory
Uvicorn	0.52.1	Servidor ASGI de la API
SQLAlchemy	2.0.51	Acceso y persistencia sobre base de datos
psycopg2-binary	2.9.12	Driver de conexión con PostgreSQL
python-bitcoinrpc	1.0	Comunicación RPC con el nodo Bitcoin
python-telegram-bot	22.8	Publicación de resultados en Telegram

PyYAML	6.0.3	Lectura de archivos de configuración YAML
OpenJDK	21 / 21.0.12	Runtime Java requerido por Spark/Hadoop

Nota. Elaboración propia.

2.6.2. Arquitectura Hexagonal (Puertos y Adaptadores)

Las razones de la elección de usar la Arquitectura Hexagonal responden a los criterios de escalabilidad y disponibilidad:

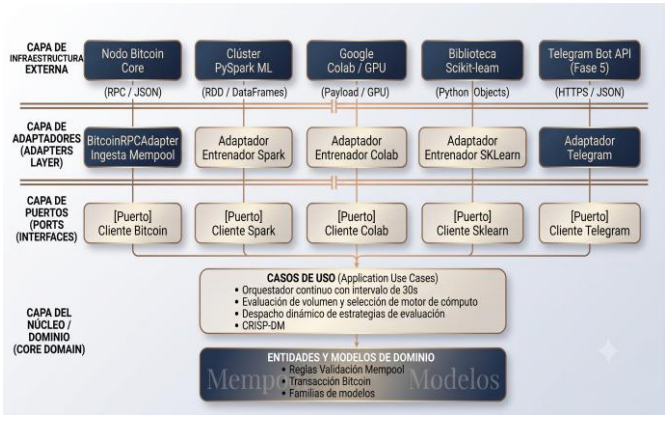
La lógica principal del sistema comprendida por: la extracción de variables, reglas de limpieza en los datos sobre las transacciones en la mempool y la inferencia de modelos de ML son completamente aisladas de los detalles tecnológicos externos. De esta manera el núcleo de la investigación no se ve afectado, por los posibles cambios en la API del nodo Bitcoin Core o la migración de librerías de Machine Learning.

El núcleo de la solución interactúa solo mediante contratos abstractos (Puertos) con las implementaciones concretas (Adaptadores)como pueden ser la base de datos, al nodo Bitcoin e incluso al notificador en Telegram, convirtiéndolos en componentes intercambiables.

Al aislar el dominio de la solución, se facilita la posibilidad de realizar pruebas unitarias y de integración de todo el sistema, así como validaciones del pipeline de ML, donde se puede utilizar datos simulados en la fase de pruebas unitarias, sin depender del estado cambiante o la latencia de la red real de Bitcoin en fases preliminares.

Figura 1

Arquitectura Hexagonal de la solución



Nota. Elaboración propia.

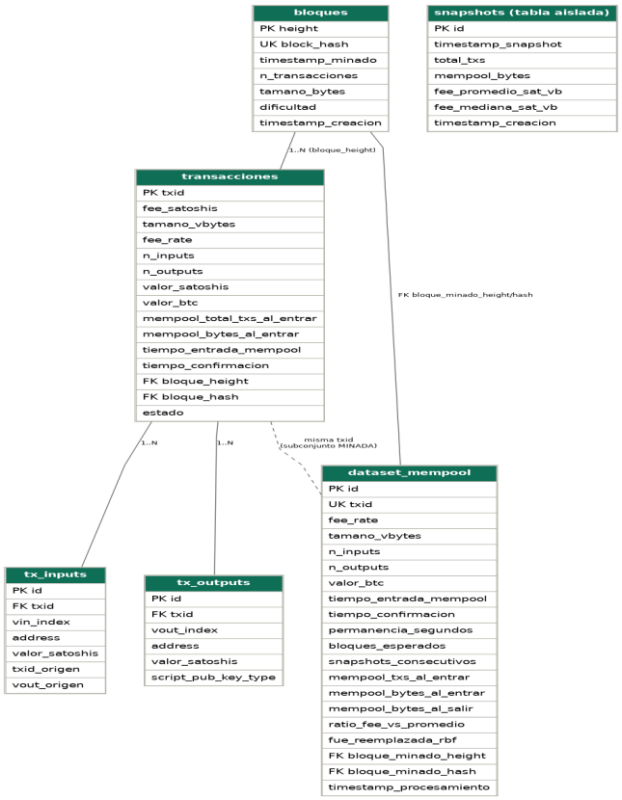
2.6.3. Diseño y Estrategia de Persistencia en PostgreSQL

Las transacciones de la mempool tienen datos que son fuertemente relacionados entre sí, tales como: comisión, tamaño en vBytes, tasa de comisión, locktime. También tiene componentes semiestructurados o variables de múltiples inputs y outputs. PostgreSQL permite almacenar datos mediante el tipo de dato JSONB, lo cual permite estructurar los metadatos principales mediante esquemas relacionales estrictos.

La mempool de Bitcoin es volátil y experimenta picos de tráfico elevados según las épocas. PostgreSQL permite la integridad de los datos procesados durante la alta tasa de escritura, mediante mecanismos robustos de concurrencia como Multi-Version Concurrency Control.

Figura 2

Arquitectura de Base de Datos para adquisición de datos



Nota. Elaboración propia.

2.5.6 Conexión RPC a datos On-Chain

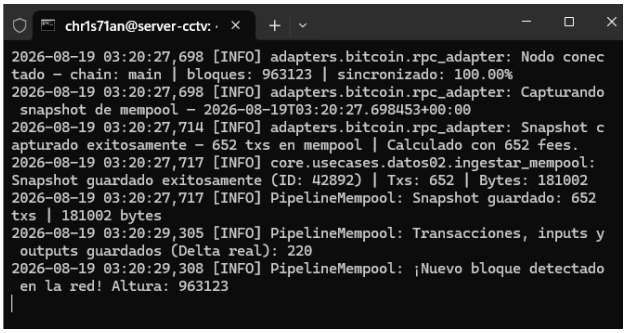
La conexión con el nodo Bitcoin se realiza mediante RPC (Remote Procedure Call), lo que permite a la aplicación ejecutar solicitudes directamente al Bitcoin Core implementado en

el servidor y acceder al mismo para obtener la información solicitada para la ingesta de datos. Es importante mencionar que los timestamps obtenidos del nodo de Bitcoin utilizan como referencia temporal UTC, mientras que la solución desarrollada registra a los eventos de la ingesta de datos utilizando la zona horaria local de Ecuador (UTC−5), por lo que, al contrastar algunos registros se puede observar la diferencia de horario que corresponde a cinco horas adelantadas a nuestro horario (UTC-5).

El nodo de Bitcoin se encuentra implementado en el servidor en modo podado con un valor de 100 GB, lo que quiere decir que del total del que requiere la blockchain de Bitcoin que se acerca a 1 TB, el nodo podado tendrá acceso y podrá validar transacciones que requieran acceso a los bloques que se almacenen en el espacio de 100 GB, para los fines de esta investigación no representan ninguna limitante, ya que las consultas de los datos on-chain se puede realizar tanto a los bloques del blockchain disponibles en el nodo, como también a la mempool de Bitcoin, que corresponde a las transacciones temporales válidas que ha sido recibido por el nodo, pero que aún no han sido incluidas en un bloque por uno de los mineros. Estas transacciones permanecen en espera de confirmación y su composición puede cambiar continuamente conforme ingresan las nuevas transacciones o las transacciones existentes llegan a ser confirmadas o descartadas.

Figura 3

Captura de ingesta de nodo Bitcoin en tiempo real (UTC) de datos on-chain



Nota. Elaboración propia 18/8/2026 – 22:24

En la base de datos de la solución desarrollada se van almacenando los datos capturados de la mempool en tiempo real, estos datos se guardan en las diferentes tablas de la arquitectura de datos, sin embargo, para esta investigación se utilizarán 3 tablas de la arquitectura destinadas a capturar las variables necesarias para cada modelo de Machine Learning no supervisado, los datasets están conformados de la siguiente manera:

- Modelos de Clustering: Dentro de la solución se realiza un join en base al timestamp entre las dos tablas para luego se preparada para los modelos de clustering

    Tabla snapshots

    Tabla bloques

- Modelos de Anomalías: Esta tabla se encuentra lista para ser preparada para los modelos de anomalías

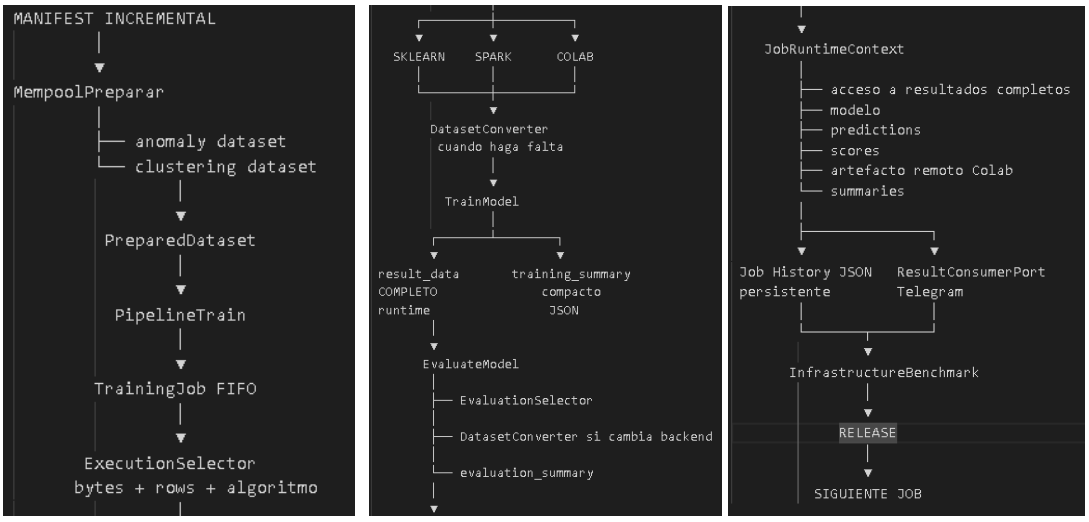
    Tabla dataset_mempool

2.6.5 Integración del Pipeline de Machine Learning sobre Nodos Propios

La arquitectura integra el ciclo de vida de datos definido por la metodología CRISP-DM:

Figura 4

Ciclo de Vida CRISP-DM de la solución



Nota. Elaboración propia.

Un Adaptador de Entrada ejecuta consultas JSON-RPC hacia el nodo local podado. Esto elimina la latencia de red que se suele tener en la entrega de información de APIs de terceros, así como los límites de peticiones (rate limits) y los costos monetarios asociados.

El Núcleo del Dominio extrae métricas clave por transacción, entre algunas variables utilizadas en la investigación tenemos:

- Fee Rate(satoshis/vByte).

- Relación Entrada/Salida (Input/Output ratio).
- Tamaño en vBytes y uso de características específicas.

Los vectores de características se persisten en PostgreSQL y simultáneamente se alimentan al Adaptador de Machine Learning. Los modelos no supervisados calculan:

- Score de Anomalía: Aplicadas por Isolation Forest.
- Asignación de Clústeres: Aplicadas por K-Means o BSCAN para etiquetado de patrones.

Este entrenamiento de cada modelo se realiza en diferentes infraestructuras, los cuales asan por unos selectores los cuáles en base al tipo de modelo, tamaño del dataset, número de filas del dataset, selecciona en que infraestructura se entrenara.

Las etiquetas de anomalías y conglomerados se persisten junto a la transacción correspondiente en PostgreSQL para su posterior visualización y validación científica, una vez consolidado en la base de datos, se realiza backups periódicos en intervalos de 10 días al sistema Hadoop, para un almacenamiento distribuido.

Finalmente, entrenados los modelos, se procede a enviar la información al canal de Telegram para la distribución de los resultados de los modelos

2.6.6 Tipos de Infraestructura

Las infraestructuras que tiene implementado la solución para el entrenamiento de los modelos según el tipo de datasets son los siguientes:

Scikit-learn: Infraestructura local orientada al procesamiento en memoria con recursos del servidor. Está orientada principalmente para datasets pequeños que pueden ser procesados con los recursos disponibles en una sola máquina.

Apache Spark: infraestructura distribuida utilizada para procesar grandes volúmenes de datos. Para esta solución se encuentra integrada con **Hadoop/YARN**, permitiendo distribuir las tareas entre los recursos disponibles y trabajar directamente con los datos almacenados en HDFS.

Google Colab: Infraestructura externa utilizada como alternativa de ejecución en la nube, lo que permite disponer de recursos computacionales diferentes a los existentes en la infraestructura local, para datasets mucho más grandes de lo que pueda procesar el clúster.

La solución fue diseñada para que pueda identificar los límites de cada infraestructura para el entrenamiento de los modelos, donde los selectores eligen a partir del tamaño del dataset en bytes y el número de filas del dataset, los criterios se tomaron a criterio propio con la premisa de al entrenar los modelos observar el comportamiento del servidor, del clúster y de la infraestructura externa con Google Colab y en base a los benchmarks de tiempo ir afinando los criterios de selección, las siguientes tablas muestra dichos criterios.

Tabla 2

Criterios generales para la selección de infraestructura según el volumen del dataset

Tamaño del dataset	Número de registros	Infraestructura seleccionada
≤ 512 MiB	≤ 500.000	Scikit-Learn
> 512 MiB	Cualquier cantidad	Apache Spark
≤ 512 MiB	> 500.000	Apache Spark
> 512 MiB	Cualquier cantidad	Google Colab
≤ 512 MiB	> 500.000	Google Colab

Nota. Elaboración propia

En el caso específico de LOF, se preserva la ejecución mediante la infraestructura de Scikit-learn implementada en la solución, mientras que los algoritmos que disponen de soporte distribuido se pueden ejecutar en Apache Spark y los recursos administrados por YARN.

Finalmente, está limitación se convierte en una línea de posible investigación y escalamiento para la aplicación de entrenamiento de modelos de Machine Learning. La arquitectura actual permite incorporar en el futuro nuevas implementaciones como, por ejemplo, la de futuras versiones distribuidas de LOF o nuevos motores de procesamiento, sin necesidad de modificar el núcleo de la solución desarrollada. De esta manera, la restricción identificada durante la investigación queda contenida en la capa de infraestructura y no compromete la extensibilidad general de la solución.

2.6.7 Independencia Operativa y Financiera

Al tener un nodo propio de Bitcoin y con comunicación directa mediante RPC, se eliminan los posibles costos generados por conexiones a APIs comerciales, garantizando el acceso directo a los datos sin filtrar.

El hecho de usar un nodo en modo pruned y con la eficiencia de PostgreSQL permite ejecutar la solución en infraestructura modesta sin sacrificar rendimiento.

La separación clara de responsabilidades que se realiza mediante la Arquitectura Hexagonal ayuda en varios aspectos como: la auditoría del código, la reproducibilidad de los experimentos de ML y la futura escalabilidad del sistema a entornos de producción.

Capítulo III: Resultados

En este capítulo, se tratará sobre los resultados del entrenamiento obtenido por los modelos seleccionados previamente sobre los datos de la mempool de Bitcoin, con pruned de 100 GB, el cuál realizando un cálculo rápido podemos saber el tiempo de información conservada en el nodo, donde altura actual es el último bloque minado y prune height es el bloque más antiguo que tiene el nodo:

Altura actual: 962844

Prune height: 903375

962844 – 903375 +1 = 59470 bloques disponibles en el nodo

Sabiendo que la fecha de los datos para este cálculo fue el 16/08/2026 y que el intervalo promedio de minado de cada bloque es de 10 minutos, entonces:

59470 x 10 = 594700 minutos

594700 / (60 x 24) = 413 días de historial en el nodo de Bitcoin

Los resultados se presentan a partir de fuentes de evidencia previamente delimitadas en la solución con el fin de asegurar trazabilidad, reproducibilidad y consistencia metodológica. La fuente oficial para los resultados obtenidos de los modelos corresponde a los archivos de job_history, los cuales concentran los resultados de entrenamiento, evaluación, comparación de evaluadores y benchmark de infraestructura. Así también, el linaje de los datos se sustenta en el manifest incremental almacenado en HDFS junto a los backups realizados de la base de datos, que permite relacionar cada ejecución con el backup exacto de datos utilizado.

Para el entrenamiento de los modelos tanto de la familia de clustering como de anomalías, se realizaron backups en formato Parquet de la base de datos que contiene datos de la mempool de Bitcoin, los resultados de los entrenamientos se pueden revisar en el canal de Telegram, las fechas de los backups para trazabilidad se muestra cómo se generaron a continuación:

- Ventana 1

  Fecha de generación: 18/08/2026 - 22:07

Nombre Backup: incremetal-manual-2026-08-18-214132-621453

Rango de fechas del dataset: 27-07-2026 al 05-08-2026

- Ventana 2

  Fecha de generación: 18/08/2026 -23:25

Nombre Backup: incremental-manual-2026-08-18-221243-657451
Rango de fechas del dataset: 06-08-2026 al 15-08-2026

3.1 Configuración de Hiperparámetros

Con el objetivo de determinar la configuración óptima de los hiperparámetrosy evaluar la estabilidad estadística de todos los modelos sobre datos de transacciones del mempool, se ejecutaron cuatro corridas experimentales sobre la misma ventana temporal de 2 días, este dataset fue recuperado de los datos de la mempool con fecha 27/07/2026 al 28/07/2026 desde el servidor Hadoop en formato Parquet. Recordando que lo datos de la mempool son temporales y en cada bloque minado las transacciones ingresan con sus datos y todo rastro anterior es eliminado, sin embargo, la solución actualmente permanece capturando los datos desde el 27/08/2026. Para todos los modelos que aplique se mantuvo el random_state=42.

Para temas de trazabilidad de la investigación se cita las fechas y horas que se entrenaron los modelos y el nombre del backup, en total se generaron 4 backups de la misma temporalidad de la mempool para la configuración de los hiperparámetros, todos los datos son obtenidos de la base de datos de la aplicación:

- Entrenamiento 1:

  Fecha: 18/08/2026 - 16:48

  Nombre Backup: incremental-manual-2026-08-18-164321-401974

- Entrenamiento 2

Fecha: 18/08/2026 - 16:59

  Nombre Backup: incremental-manual-2026-08-18-165525-392850

- Entrenamiento 3

  Nombre Backup: incremental-manual-2026-08-18-170719-638304

  Fecha: 18/08/2026 - 17:11

- Entrenamiento 4:

  Fecha: 18/08/2026 - 17:19

  Nombre Backup: incremental-manual-2026-08-18-171453-768075

Todos los resultados del entrenamiento de cada modelo junto con sus métricas de evaluación se encuentran publicados en el canal de Telegram. Cabe recalcar que en todas las tablas de resultados de entrenamientos y evaluaciones de los modelos están en formato de decimales con (,) y unidad de miles con (.)

3.1.1. Isolation Forest (IF)

Los experimentos realizados con Isolation Forest sobre el dataset de 2 días, permitieron analizar el comportamiento del modelo sobre un conjunto de 412 562 registros, manteniendo inicialmente una configuración base de 100 estimadores, maxSamples=256, Bootstrap=No y una tasa de contaminación de 0,01. Luego se aumentó a maxSamples=512 y se hizo tres variaciones de la contaminación valores de 0,001; 0,01; 0,02; sabiendo que el primer entrenamiento se mantuvo en 0,001.

Con esta variación de los hiperparámetros se puede observar que manteniendo la contaminación y variando el max_samples se obtienen el mismo número de anomalías, pero si varía los scores mínimo y máximo. Al mismo tiempo al dejar con 512 y variar la contaminación se observa que el aumento de anomalías es proporcional al valor de la contaminación elegida, ya que las 4126 anomalías encontradas con el 1%, aumenta al doble exacto con el 2%, entonces viendo estos resultados en el mismo dataset, se considera mejor usar una contaminación del 1% con un max_sample de 512 y mantener los demás hiperparámetros a la línea base.

Tabla 3

Resultados de calibración de hiperparámetros de Isolation Forest (IF)

Parameter / metric	IF-1	IF-2	IF-3	IF-4
Run ID	manual-2026-08-18-164321-401974	manual-2026-08-18-165525-392850	manual-2026-08-18-170719-638304	manual-2026-08-18-171453-768075
Records	412.562	412.562	412.562	412.562
Features	12	12	12	12
n_estimators	100	100	100	100
max_samples	256	512	512	512
contamination	0,01	0,01	0,001	0,02
max_features	1	1	1	1
bootstrap	No	No	No	No
random_state	42	42	42	42
total_anomalies	4.126	4.126	413	8.252
normal_records	408.436	408.436	412.149	404.310
anomaly_ratio	0,010001	0,010001	0,001001	0,020002
normal_ratio	0,989999	0,989999	0,998999	0,979998

Parameter / metric	IF-1	IF-2	IF-3	IF-4
anomaly_scores_min	-0,192181	-0,208789	-0,101025	-0,250347
anomaly_scores_max	0,274143	0,264639	0,372403	0,223081
Training time (s)	**3,518**	**3,7676**	**3,8268**	**3,7158**

3.1.2. K-Means

En K-Means al entrenar el modelo con el mismo dataset de temporalidad de dos días, y para entrenar al modelo se fue variando el número de clústeres, se empezó con un k=2 hasta llegar a k=5. En donde, se pudo ir observando como la métrica de Silhouette fue disminuyendo, en los resultados para k=3 y k=5 se obtuvo los valores más bajos, mientras que en k=2 fue el valor más alto con 0,364829, seguido por k=4 con un valor de 0,306735, lo que indica que la cohesión interna y la separación entre clústeres se mantiene relativamente bien para los datos de la mempool analizados. Sin embargo, al contrastar con Davies-Bouldin su mejor valoración estuvo para k=4 con un valor de 1,128377. mientras que la métrica de Calinski-Harabasz fue degradándose cada que se incrementaba el número de clústeres, siendo un claro indicio que la relación intercluster se va degradando cada que se aumenta el hiperparámetro k. Entonces después de observar estos resultados se determina que la mejor configuración para K-Means es para k=4, por sus valores de las métricas de evaluación relativamente buenos con un Silhouette de 0,306735, Devies-Bouldinde 1,128377 y Calinski-Harabasz con un 2137,08. Sabiendo que los demás hiperparámetros se mantienen en línea base.

Tabla 4

Resultados de calibración de hiperparámetros de K-Means

Parameter / metric	KM-1	KM-2	KM-3	KM-4
Run ID	manual-2026-08-18-164321-401974	manual-2026-08-18-165525-392850	manual-2026-08-18-170719-638304	manual-2026-08-18-171453-768075
Records	5.760	5.760	5.760	5.760
Features	8	8	8	8
n_clusters	2	3	4	5
init	k-means++	k-means++	k-means++	k-means++
metric	euclidean	euclidean	euclidean	euclidean
n_init	10	10	10	10
max_iter	300	300	300	300
tol	0,0001	0,0001	0,0001	0,0001

Parameter / metric	KM-1	KM-2	KM-3	KM-4
random_state	42	42	42	42
Clusters detected	2	3	4	5
Noise records	0	0	0	0
Noise ratio	0	0	0	0
Silhouette	0,364829	0,284699	0,306735	0,277802
Davies-Bouldin	1,192749	1,289539	1,128377	1,169953
Calinski-Harabasz	2.658,31	2.265,91	2.137,08	1.979,36
Training time (s)	**0,5843**	**0,8261**	**0,5567**	**0,3605**

3.1.3. DBSCAN

En el entrenamiento de DBSCAN se mantuvo los hiperparámetros base y se verifico los resultados del entrenamiento con la variación del hiperparámetro eps para ir variando la distancia a los vecinos. Donde los valores fueron mejorando en consideración al valor obtenido con un eps=0,5. Sin embargo, Silhouette obtuvo la mejor métrica con 0,241262, luego Davies-Bouldin estuvo muy cercana a las dos mejores métricas con un valor de 0,698084, mientras que Calinski-Harabasz obtuvo la mejor métrica con 256,504654. Donde qué valor de eps queda seleccionado con un valor de 1,5, ya que muestra que logró mayor cohesión de los agrupamientos en base a los valores obtenidos de las métricas y una buena relación de separación entre los agrupamientos.

Tabla 5

Resultados de calibración de hiperparámetros de DBSCAN

Parameter / metric	DB-1	DB-2	DB-3	DB-4
Run ID	manual-2026-08-18-164321-401974	manual-2026-08-18-165525-392850	manual-2026-08-18-170719-638304	manual-2026-08-18-171453-768075
Records	5.760	5.760	5.760	5.760
Features	8	8	8	8
eps	0,5	1	1,2	1,5
min_samples	5	5	5	5
metric	euclidean	euclidean	euclidean	euclidean
Clusters detected	40	8	9	7
Noise records	179	25	14	7
Noise ratio	0,031076	0,00434	0,002431	0,001215
Silhouette	-0,349231	0,180717	0,162833	0,241262
Davies-Bouldin	0,987209	0,681166	0,689362	0,698048
Calinski-Harabasz	80,210072	223,445827	197,676244	256,504654

Parameter / metric	DB-1	DB-2	DB-3	DB-4
Training time (s)	**0,774**	**0,4037**	**0,4328**	**0,4639**

3.2 Resultados de Detección de Anomalías

3.2.1. Isolation Forest

El entrenamiento de este modelo se realizó en ventanas temporales distintas y el análisis corresponden a periodos diferentes de observación de la mempool y no fueron diseñadas como conjuntos equivalentes para realizar una comparación directa entre ellas. Los resultados pertenecen a cada ventana temporal de datos de la mempool, por lo que los resultados se analizan de manera independiente. La finalidad es interpretar el comportamiento observado en la mempool sin considerar una ventana como referencia o control de la otra.

Tabla 6

Desempeño de Isolation Forest en dos ventanas de tiempo diferentes

Parameter / metric	Ventana 1	Ventana 2
Temporalidad	27-07-2026 al 05-08-2026	06-08-2026 al 15-08-2026
Run ID	manual-2026-08-18-214132-621453	manual-2026-08-18-221243-657451
Infrastructure	Spark	sklearn
Records	2.846.280	196.385
Dataset size	260,310855 MiB	17,779721 MiB
Features	12	12
Hiperparámetros		
n_estimators / numEstimators	100	100
max_samples / maxSamples	512	512
contamination	0,01	0,01
contaminationError	0,0001	—
max_features / maxFeatures	1	1
bootstrap	No	No
random_state / randomSeed	42	42
Resultados		
Total anomalies	28.507	1.964
Normal records	2.817.773	194.421
Anomaly ratio	0,010016	0,010001
Normal ratio	0,989984	0,989999
anomaly_scores_min	0,311878	-0,220341
anomaly_scores_max	0,818049	0,279608
Status	SUCCESS	SUCCESS

Nota. Elaboración propia con datos de mempool de Bitcoin

En la primera ventana se observa que el entrenamiento fue realizado en la infraestructura de Spark, esto debido a su tamaño de 260 MiB y casi llegando a los 3 millones

de filas. Cabe recalcar que 1000 B = 1 KB y que 1024 B = 1KiB. El modelo IF se entrenó con maxSamples= 512 y una contaminación de 1%. Dentro de sus resultados se puede observar que el modelo determino 28507 anomalías con la contaminación ya configurada al 1%, estos resultados los realizo sobre una infraestructura distribuida con un dataset de 2,8 millones de filas. Mirando los scores min=0,311878 y max=0,818049, se identifica que el comportamiento que IF no fue binario en el origen, pues el modelo calculo los grados de atipicidad, donde genero su umbral y en base a él clasifico a las anomalías, tomando en cuenta que en Spark con SynapseML, la forma de determinar la atipicidad es el valor más alto se convierte en atípico, es decir, que el valor del anomaly_scores_max=0,818049 representaría el grado de atipicidad más alto.

En la segunda ventana se observa claramente una disminución del tráfico de la mempool debido a su menor número de filas que representan a transacciones a la espera, en este caso con 196 385 filas y un tamaño de 17,8 MiB, por lo que en este caso el selector de la aplicación determino que se puede entrenar en la infraestructura de sklearn, el modelo se mantuvo con los hiperparámetros establecidos en la configuración, por lo que su maxSample fue igual a 512 y su contaminación fue de 1%. En sus resultados obtuvo 1964 anomalías que representan a la configuración establecida, sin embargo, esto indica que encontró las observaciones que fueron las más fáciles de separar, con sus scores de anomalía min=-0,220341 y su max=0,278608, sabiendo que la forma de clasificar una anomalía para Scikit-Learn es diferente a la de Spark con SynapseML, donde el valor más atípico es el que tiende a la parte negativa de los valores calculados, es decir, el valor -0,220341 s el valor que representa la mayor atipicidad.

3.3 Resultados de Clustering

3.3.1. K-Means

El entrenamiento de este modelo se realizó en ventanas temporales distintas y el análisis corresponden a periodos diferentes de observación de la mempool y no fueron diseñadas como conjuntos equivalentes para realizar una comparación directa entre ellas. Los resultados pertenecen a cada ventana temporal de datos de la mempool, por lo que los resultados se analizan de manera independiente. La finalidad es interpretar la estructura global observado en la mempool sin considerar una ventana como referencia o control de la otra.

Tabla 7

Desempeño de K-Means en dos ventanas de tiempo diferentes

Parámetro / métrica	Ventana 1	Ventana 2
Temporalidad	27-07-2026 al 05-08-2026	06-08-2026 al 15-08-2026
Run ID	manual-2026-08-18-214132-621453	manual-2026-08-18-221243-657451
Infraestructura	sklearn	sklearn
Registros	28.558	11.722
Tamaño del dataset	1,132804 MiB	452,598633 KiB
Features	8	8
Hiperparámetros		
n_clusters	4	4
init	k-means++	k-means++
metric	euclidean	euclidean
n_init	10	10
max_iter	300	300
tol	0,0001	0,0001
random_state	42	42
Resultados		
Clústeres detectados	4	4
Silhouette	0,284359	0,276642
Davies-Bouldin	1,126796	1,160319
Calinski-Harabasz	7.628,03	2.980,55
Estado	SUCCESS	SUCCESS

Nota. Elaboración propia con datos de mempool de Bitcoin

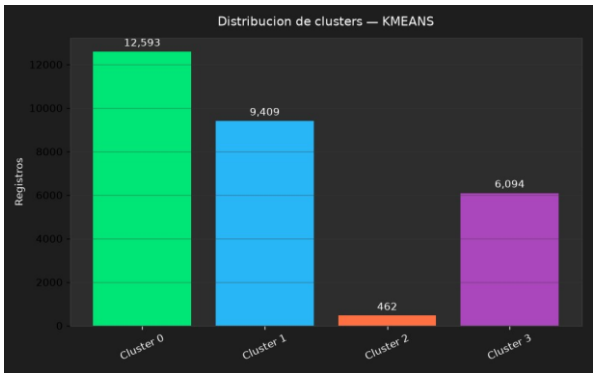
En la Ventana 1 se pude observar el tamaño del dataset de 1,13MiB y de 28558 filas, este dataset fue procesado por la infraestructura de sklearn porque cumplieron los límites de los selectores para ser procesado completamente en esta infraestructura, así también se mantuvieron los hiperparámetros seleccionados en la configuración, con el valor de número de clústeres k=4, se puede divisar que el valor de Silhouette con 0,284359, muestra que existe separación entre los clústeres con cierto grado de solapamiento o continuidad, puesto que este valor no indica que la separación este fuertemente marcada entre los diferentes clústeres formados por el modelo. El resultado obtenido de 1,126796 con Davis-Bouldin indica una lectura similar a Silhouette, donde la relación entre compactación interna y la separación entre agrupamientos no muestra una división que este muy marcada, mientras que Calinski-Harabasz si muestra que existe la relación entre la dispersión interna y la separación de los agrupamientos globales en los datos de la mempool, aunque el resultado no muestra el grado directamente, pues hay que ir determinando con cada entrenamiento para saber si es un valor fuerte o medio de agrupamiento global.

Al observar las distribuciones de los clústeres, se identifica claramente que, de los cuatro clústeres, los dos primeros abarcan la mayoría de observaciones lo que juntos representan el 77,04% de las observaciones totales y si incluimos al cuarto clúster

representarían el 98,4% de todas las transacciones de la mempool, con el tercer clúster que representa un patrón minoritario de transacciones; si relacionamos con los resultados de Silhouette, se observa que tienen un cierto grado de solapamiento donde sus fronteras no están fuertemente marcadas.

Figura 5

Gráfica de distribución de clústeres distribuidos por K-Means en la Ventana 1



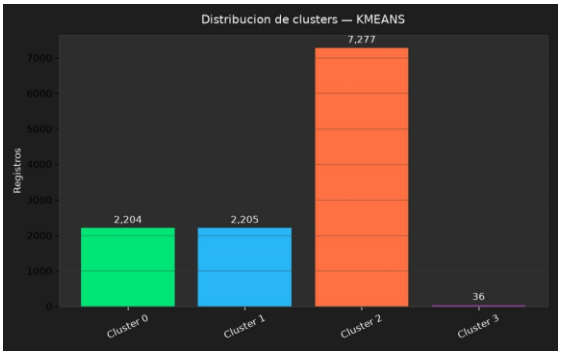
Nota. Elaboración propia

En los resultados de la Ventana 2, el entrenamiento se realizó sobre un dataset de menor dimensión con 11722 registros y un tamaño de 452,59 KiB, este dataset debido a su dimensión fue procesado en la infraestructura sklearn, donde se mantuvo los hiperparámetros de la configuración en la que k=4. Los resultados que se pueden observar muestran para Silhouette un valor de 0,276642 indicando que existen una estructura de agrupamiento, pero con fronteras no tan marcadas o cierto grado de solapamientos, en Davis-Bouldin se obtiene un valor de 1,160319 indicando una relación con la dispersión intra-clúster con una cierta cercanía entre las distancias de los agrupamientos, cual complementa a lo que indico Silhouette, mientras tanto Calinski-Harabasz entrega un valor de 2980,55, representando la relación entre la separación de los agrupamientos con la dispersión de cada clúster.

En el análisis de las transacciones que el modelo distribuyo en los clústeres, se puede observar que el clúster 2 obtiene la mayor proporción de las transacciones representado por 62,1% , después observamos que el clúster 0 y 1 tienen una proporción similar entre ellos y juntos representan el 37,6% de las transacciones globales y al final el clúster 3 tiene una proporción muy pequeña representada por el 0.31% y al relacionar con el resultado de Silhouette vemos que esta ventana temporal tiene una estructura de agrupamiento pero con fronteras que tienen cierto grado de solapamiento.

Figura 6

Gráfica de distribución de clústeres distribuidos por K-Means en la Ventana 2



Nota. Elaboración propia

3.1.2. DBSCAN

El entrenamiento de este modelo se realizó en ventanas temporales distintas y el análisis corresponden a periodos diferentes de observación de la mempool y no fueron diseñadas como conjuntos equivalentes para realizar una comparación directa entre ellas. Los resultados pertenecen a cada ventana temporal de datos de la mempool, por lo que los resultados se analizan de manera independiente. La finalidad es interpretar la naturalidad de los agrupamientos en la mempool sin considerar una ventana como referencia o control de la otra.

Tabla 8

Desempeño de DBSCAN en dos ventanas de tiempo diferentes

Parámetro / métrica	Ventana 1	Ventana 2
Temporalidad	27-07-2026 al 05-08-2026	06-08-2026 al 15-08-2026
Run ID	manual-2026-08-18-214132-621453	manual-2026-08-18-221243-657451
Infraestructura	sklearn	sklearn
Registros	28.558	11.722
Tamaño del dataset	1,132804 MiB	452,598633 KiB
Features	8	8
eps	1,5	1,5
min_samples	5	5
metric	euclidean	euclidean
Clústeres detectados	12	8
Registros de ruido	31	22
Proporción de ruido	0,001086	0,001877
Silhouette	0,171535	0,145961
Davies-Bouldin	0,543851	0,913064
Calinski-Harabasz	363,732195	584,093598
Estado	SUCCESS	SUCCESS

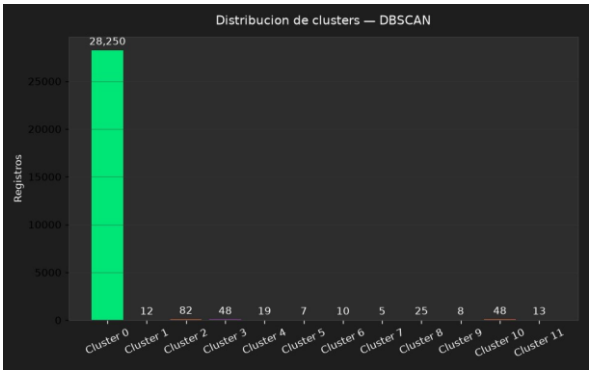
Nota. Elaboración propia con datos de mempool de Bitcoin

En la Ventana 1 se entrenó el modelo manteniendo los hiperparámetros seleccionado en la sección de configuración, en la que se eligió que el valor para eps sea igual a 1,5 y manteniendo la línea base para los demás hiperparámetros. Esta configuración se aplicó a un dataset de 28558 registros con un tamaño de 1,13 MiB, en el que el modelo determino doce clústeres con una valoración de Silhouette de 0,171535 representando agrupaciones con fronteras débilmente marcadas. La métrica de Davies-Bouldin entrego un valor de 0.543851 dando una relación razonable de su compactación de las agrupaciones frente a la separación que hay entre ellos. Calinski-Harabasz por su parte entrego una valoración de 363,732195 representando la relación entre las dispersiones internas y las externas. Entendiendo que esta métrica por si sola, no entrega el grado de influencia que pueda tener en la mempool, por lo que al comparar más entrenamientos con diferentes ventanas temporales se podrá observar cómo va cambiando la estructura bajo la misma configuración.

La distribución de los clústeres tiene marcado el clúster 0 como el más predominante para las transacciones de esta ventana temporal, representando el 98,92% de las transacciones globales, mientras que el restante está distribuido entre los 11 clústeres restantes, lo que tiene coherencia con los resultados de las métricas de evaluación analizadas anteriormente. Entre los clústeres de baja densidad se puede distinguir al clúster 7, el cual tiene 5 transacciones que coincide con el hiperparámetro configurado, lo que significa que este clúster alcanzo con el mínimo para ser generado. A esto se suma los 31 registros que no pudieron ser agrupados y que no cumplieron con los requisitos mínimos por lo que fueron catalogados como ruido.

Figura 7

Gráfica de distribución de clústeres identificados por DBSCAN en la Ventana 1



Nota. Elaboración propia

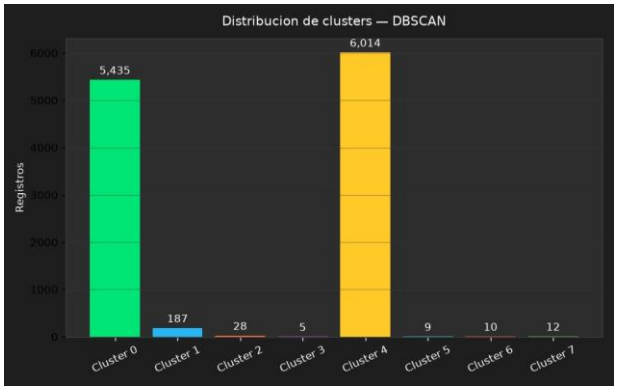
En la Ventana 2 se entrenó al modelo con un dataset de 11722 registros y con un tamaño de 452,58 KiB, por lo que el dataset cumplió los criterios del selector para ser entrenado en la

infraestructura de sklearn, para este entrenamiento se mantuvieron los hiperparámetros configurados en donde se mantienen la línea base y el hiperparámetro eps se asignó el valor de 1,5 para las distancias entre los vecinos. Para los cual, el modelo determino ocho clústeres y que el resultado que se obtuvo en la métrica de evaluación Silhouette fue de 0,145961 indicando que la relación entre la diferencia de distancias con sus vecinos y su distancia máxima lo que muestra solapamientos entre los agrupamientos, mientras que, Davies-Bouldin entrega un valor de 0,913064 mostrando una relación razonable entre la compactación de los agrupamientos y la distancia entre dichos agrupamientos siendo un complemento al resultado de Silhouette, mientras tanto, Calinski-Harabasz entrega una valoración de 584,093598 representando la relación de las dispersiones internas de las agrupaciones y la separación entre ellos.

La estructura de densidad de esta segunda ventana está concentrada principalmente en dos agrupamientos que corresponden al 0 y 4 que representan el 94,67% del total de las transacciones, mientras que alrededor de ellos se muestran agrupamientos con poca densidad en incluso detecto una cantidad de 22 transacciones que no pertenecieron a ningún agrupamiento y fue catalogado como ruido.

Figura 8

Gráfica de distribución de clústeres identificados por DBSCAN en la Ventana 2



Nota. Elaboración propia

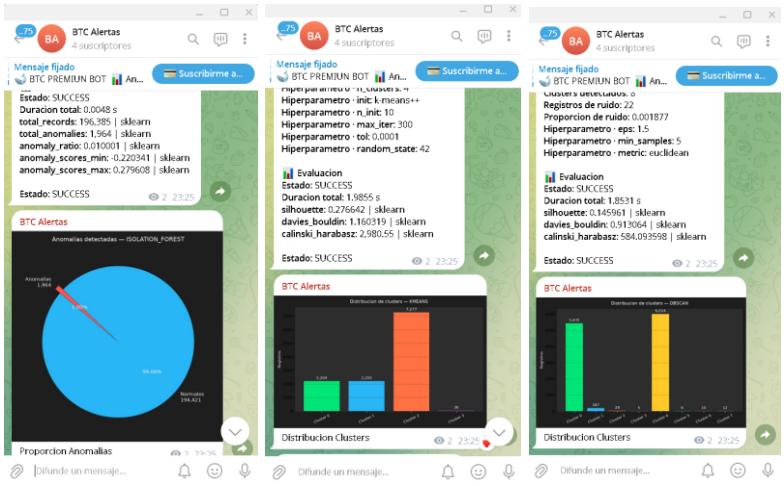
3.3 Difusión en Telegram

Como etapa final del proceso de entrenamiento, así como etapa final de la metodología CRISP-DM, se implementó un mecanismo de notificación automática mediante un canal de Telegram, el cual permitirá que la información sea difundida a penas se tenga los reportes de entrenamiento de los modelos con el dataset seleccionado.

Las notificaciones que se envían al canal de Telegram incluyen información relacionada con la ejecución del experimento: la infraestructura utilizada, hiperparámetros del modelo, estado de ejecución, duración y métricas de evaluación. Durante la fase de despliegue, se realizan gráficas de los resultados obtenidos por los modelos y que también son transmitidas al canal de Telegram, como se muestra en la siguiente figura

Figura 9

Difusión de resultados mediante Telegram de IF, K-Means, DBSCAN



Nota. Elaboración propia

La Figura presenta los resultados obtenidos del entrenamiento de los modelos Isolation Forest, K-Means y DBSCAN respectivamente en la primera ventana temporal de los 10 primeros días y que se difundieron en la última fase de despliegue mediante el canal de Telegram. Estas visualizaciones permiten observar en las gráficas generadas directamente los resultados obtenidos y que se analizaron anteriormente, donde: Isolation Forest muestra la proporción de anomalías detectadas, K-Means y DBSCAN muestran la distribución de los patrones de agrupamiento que se identificaron. Lo que se tiene como información complementaria a los resultados cuantitativos obtenidos sin la necesidad de ingresar a la infraestructura donde se generaron estos resultados.

De esta manera, se demuestra flujo desde la adquisición y preparación hasta la difusión de los resultados de entrenamiento realizados con los datos de la mempool, pasando por todas las fases de la metodología CRISP-DM implementadas en la solución hasta la difusión de los resultados obtenidos hacia un mecanismo externo de consulta.

3.7 Benchmark de Tiempos

Durante la etapa de calibración de hiperparámetros, las cuatro ejecuciones fueron realizadas sobre infraestructura sklearn y bajo el mismo dataset, lo que permitió observar el comportamiento en las ventanas temporales de los modelos en condiciones controladas. Isolation Forest presentó tiempos de entrenamiento entre 3,518 y 3,8268 segundos, mientras que su evaluación se mantuvo alrededor de una centésima de segundo. K-Means registró entrenamientos entre 0,3605 y 0,8261 segundos y evaluaciones entre 0,5359 y 0,7247 segundos. Por otra parte, DBSCAN presentó tiempos de entrenamiento entre 0,4037 y 0,774 segundos y evaluaciones entre 0,4618 y 0,582 segundos. Estos resultados muestran que, bajo el mismo entorno de ejecución y volumen de datos, los tiempos permanecieron relativamente estables entre las diferentes configuraciones evaluadas.

Tabla 9

Benchmark de tiempos de ejecución de configuración de Hiperparámetros

Métrica	**Ejecución 1**	**Ejecución 2**	**Ejecución 3**	**Ejecución 4**
Run ID	164321-401974	165525-392850	170719-638304	171453-768075
Infraestructura	Sklearn	Sklearn	Sklearn	Sklearn
Isolation Forest – entrenamiento (s)	3,518	3,7676	3,8268	3,7158
Isolation Forest – evaluación (s)	0,0098	0,0107	0,011	0,0116
K-Means – entrenamiento (s)	0,5843	0,8261	0,5567	0,3605
K-Means – evaluación (s)	0,6904	0,7247	0,5359	0,5659
DBSCAN – entrenamiento (s)	0,774	0,4037	0,4328	0,4639
DBSCAN – evaluación (s)	0,5148	0,5136	0,582	0,4618

Nota. Elaboración propia

En las ejecuciones sobre las ventanas temporales reales, los tiempos reflejan además el efecto del tipo de dataset y de la infraestructura seleccionada automáticamente. En la Ventana 1, Isolation Forest procesó aproximadamente 2,8 millones de registros mediante Spark, registrando 636,17 segundos de entrenamiento y 292,5122 segundos de evaluación, mientras que en la Ventana 2, con un volumen considerablemente menor, fue ejecutado mediante sklearn con 1,9492 segundos de entrenamiento y 0,0048 segundos de evaluación. K-Means y DBSCAN permanecieron en sklearn en ambas ventanas, registrando variaciones en sus tiempos de acuerdo con las características y volumen de cada dataset. Estos resultados no se utilizan para determinar que una infraestructura sea superior a otra, debido a que las

ventanas corresponden a conjuntos de datos diferentes, sino para evidenciar el comportamiento temporal observado por la Factory al seleccionar y ejecutar cada infraestructura de acuerdo con las condiciones de procesamiento.

A pesar de que los dataset cumplieron criterios para ser entrenado en estas dos infraestructuras como sklearn y Spark, en las etapas de pruebas end to end, se utilizó la infraestructura de Google Colab, donde se procesó dos datasets de 24 registros y 8 features con tiempos de 18,95 segundos y el otro con 19,32 segundos. Por ser una infraestructura externa, el tiempo entre comunicarse y entrenar en la infraestructura de Google Colab, registra un incremento en los segundos frente a las infraestructuras internas, por lo que el verdadero impacto se podrá mediar al entrenar un dataset que supere los criterios de selección con tamaños superiores a 512 MiB.

Tabla 10

Benchmark de tiempos de ejecución de entrenamiento de modelos

Proceso	**Ventana 1**	**Ventana 2**
Temporalidad	27-07-2026 al 05-08-2026	06-08-2026 al 15-08-2026
Run ID	manual-2026-08-18-214132-621453	manual-2026-08-18-221243-657451
Isolation Forest – infraestructura	Spark	sklearn
Isolation Forest – entrenamiento (s)	636,17	1,9492
Isolation Forest – evaluación (s)	292,5122	0,0048
K-Means – infraestructura	sklearn	sklearn
K-Means – entrenamiento (s)	1,547	2,1524
K-Means – evaluación (s)	20,2737	1,9855
DBSCAN – infraestructura	sklearn	sklearn
DBSCAN – entrenamiento (s)	7,6374	1,194
DBSCAN – evaluación (s)	11,517	1,8531

Nota. Elaboración propia

Capítulo IV: Discusión de Resultados

4.1 Discusión y Contraste de Hallazgos

4.1.1 Discusión de resultados

Los resultados obtenidos mediante el entrenamiento de los modelos no supervisados evidencian que la mempool de Bitcoin presenta una estructura heterogénea cuya organización varía entre ventanas temporales. Esta característica se pudo observar especialmente con los resultados obtenidos en el modelo DBSCAN, en el cual se mantuvieron constantes los hiperparámetros eps=1,5 y min_samples=5, lo cual permitió lograr la identificación de 12 regiones de densidad en la primera ventana y 8 en la segunda. Mientras los registros clasificados como ruido permanecieron de cierta manera estable, pasando de 0,1086% a 0,1877%en la segunda. Estos resultados se obtuvieron en conjunto con una reducción considerable del volumen de transacciones. Por lo tanto, aunque las regiones densas experimentaron cambios en su organización interna en cada ventana de temporalidad, la proporción de observaciones que no logró integrarse a ellas mantuvo un comportamiento similar en ambos periodos.

Esta estabilidad del ruido no debe interpretarse como que existiera alguna estabilidad del mercado de Bitcoin, si no como una estabilidad relativa del comportamiento espacial de las transacciones de la mempool bajo los criterios de densidad establecidos en el modelo. Es decir, fuera de la variabilidad de los registros de cada ventana de temporalidad utilizada aproximadamente un 0.2% de las observaciones permanecieron fuera de las regiones densas que fue detectada por DBSCAN.

Los resultados que se obtuvieron con el modelo K-Means entregan una interpretación complementaría, durante la calibración de los hiperparámetros se observó que los valores bajos del número de clústeres (k) entregaban una representación muy general del espacio de características, mientras que el incremento progresivo de k comenzaba a fragmentar estructuras que no presentaban fronteras claramente separadas y los resultados de las métricas de evaluación comenzaban a degradarse. Pero, con las configuraciones seleccionadas de los hiperparámetros, los coeficientes de Silhouette permanecieron próximos a 0,284359 para la primera ventana y 0.276642 para la primera ventana, lo que indica que los grupos obtenidos por el modelo si presentan proximidad entre sí. Por lo que, los patrones de las transacciones observadas se pueden interpretar como estructuras parcialmente continuas, debido a que existen regiones generales de agrupamiento, pero sus fronteras internas no son rígidas y pueden

modificarse a medida que cambia la composición o temporalidad de las transacciones presentes en la mempool.

La diferencia entre los resultados obtenidos de K-Means y DBSCAN refuerza esta interpretación. K-Means puede describir la estructura global de las ventanas de temporalidad con las que se entrenó mediante un número ya establecido de centroides, mientras que DBSCAN encuentra estas regiones a partir de la densidad local de las transacciones sin la necesidad de establecer este parámetro de antemano. El hecho de que DBSCAN haya identificado 12 y 8 regiones locales bajo un mismo criterio de densidad al mantener los hiperparámetros y en ventanas de temporalidad diferente, muestra que dentro de las regiones globales representadas por K-Means podrían coexistir diferentes subestructuras transaccionales. Por esta razón, se ha identificado que la organización de la mempool no se puede representarse completamente mediante una división rígida en unos pocos grupos como sucede con K-Means, pues su estructura indica que tiene diferentes niveles dependiendo de la escala desde la cual se analice.

La calibración de los hiperparámetros del modelo DBSCAN permitió observar un comportamiento entre sensibilidad y calidad interna del agrupamiento. Cuando se colocaba valores reducidos de eps producían una elevada fragmentación del espacio, Provocando un incremento considerable del número de clústeres y que provocaba valores desfavorables en las métricas de evaluación. Por el contrario, valores mayores de eps conectaban a regiones próximas, lo que reducía el número de agrupamientos y de cierta manera mejoraba las métricas internas, pero sucedía a costa de perder resolución sobre las estructuras locales. La elección del valor de eps=1,5 representa, un punto de equilibrio entre ambos comportamientos. Los resultados obtenidos con esta configuración no muestran clústeres perfectamente separados al menos en la primera ventana de temporalidad ejecutada, pero esta característica puede representar precisamente a la naturaleza continua y dinámica de las transacciones que circulan por la mempool de Bitcoin y no necesariamente a una deficiencia del modelo.

Algo importante de diferenciar en los resultados de los diferentes modelos son las transacciones clasificadas como ruido por DBSCAN y las transacciones clasificadas como anómalas por Isolation Forest. Pues en DBSCAN, estas transacciones clasificadas como ruido suceden cuando no presenta suficiente densidad de vecindad para integrarse a una región. En cambio, en Isolation Forest se analiza la facilidad con la que una transacción podría ser aislada con respecto del conjunto general. Es por esto qué estos dos resultados son completamente diferentes y no se puede asumir que hallaron lo mismo.

Para el entrenamiento de Isolation Forest, en los hiperparámetros se estableció la contaminación en 0,01, por lo que no se podría interpretar como un hallazgo en los resultados del modelo, a pesar de que este valor se obtuvo en las dos ventanas de entrenamiento, en los resultados del anomaly ratio de las métricas de evaluación. Lo que si permitió es identificar el subconjunto de transacciones con un comportamiento anómalo en cada ventana de entrenamiento con este valor de anomalía establecido, lo cual resulta especialmente útil para encontrar a las anomalías en un escenario muy cambiante como lo es la mempool de Bitcoin, sabiendo que el objetivo no es validar anomalías que ya se conocían, sino más bien el objetivo es localizar transacciones que se apartan con mayor intensidad del comportamiento predominante.

Los resultados permiten identificar las características de la mempool de Bitcoin con sus variaciones, como la combinación de estructuras generales, regiones locales que mantienen distinta densidad y transacciones con diferentes grados de anomalías. Los patrones que se identificaron no presentan fronteras completamente estáticas, más bien al contrario, presentan una composición dinámica que responde a la temporalidad de las transacciones que estuvieron presentes en cada ventana de temporalidad. Por lo que esta característica de la mempool justifica el hecho de usar varios modelos de machine learning no supervisado, ya que en la investigación cada modelo aporto diferentes perspectivas que se complementaron para brindar un panorama más completo del comportamiento de la mempool de Bitcoin. Isolation Forest permitió identificar observaciones atípicas, K-Means permitió representar la estructura global mediante agrupamientos predefinidos y DBSCAN permitió identificar estructuras locales emergentes a partir de la densidad y observaciones que no lograron integrarse a ellas.

4.1.2 Contraste con estudios previos

Los resultados que se obtuvieron presentan correspondencia con los resultados de la investigación de Vlahavas et al. (2024), quienes demostraron que las técnicas de clustering no supervisado permiten identificar patrones sobre las características de las transacciones Bitcoin. Así también, en esta investigación también se identificó estructuras diferenciables; pero, al analizar que las transacciones que llegan a la mempool hasta ser minadas en un nuevo bloque se pudo observar agrupamientos con fronteras no tan definidas y variaciones del número de transacciones entre ventanas temporales. Por lo que, estos resultados complementan el enfoque de la investigación de Vlahavas et al. al mostrar que los patrones transaccionales se pueden identificarse no solo cuando ya han llegado a blockchain, sino que también durante la etapa de

su permanencia en la mempool y previa a la confirmación, donde presentan un comportamiento mucho más dinámico.

En cuanto a la detección de anomalías, la investigación de los autores Yahia et al. (2024) evidenciaron que la utilidad de algoritmos no supervisados como Isolation Forest y LOF sirve para identificar observaciones atípicas en datos de criptomonedas. La presente investigación trasladó este enfoque para aplicar el modelo Isolation Forest y aplicarlo directamente a los datos transaccionales obtenidos desde la mempool en tiempo real, el cual se aplicó sobre grandes volúmenes de registros y se clasifico las transacciones con mayor anomalía bajo un criterio de contaminación definido. También esta investigación identifico que no solamente se debe considerar la capacidad del modelo a utilizar, sino también la compatibilidad con la infraestructura que se tenga disponible para la ejecución del modelo establecido, pues esta limitación de escalabilidad se experimentó con el modelo LOF.

Finalmente, el nodo de Bitcoin podado permitió obtener los datos de la mempool lo cual mantiene relación con el planteamiento de Matzutt et al. (2021), quienes demostraron que se puede reducir los requerimientos de almacenamiento de la blockchain en el nodo mediante estrategias de poda. En esta investigación se comprobó que un nodo podado puede tranquilamente proporcionar la información necesaria ya sea de los bloques ya minados hasta la temporalidad que permita la poda del nodo, así como también la información de la mempool, lo cual permitió alimentar el pipeline completo para la ingesta de datos mediante conexión RPC, y posteriormente su almacenamiento, procesamiento y entrenamiento de los modelos de Machine Learning no supervisados.

Conclusiones y Recomendaciones

Conclusiones

Se obtuvo información on-chain de la mempool de Bitcoin en tiempo real, mediante un pipeline de ingesta con llamadas RPC a un nodo propio configurado en modo poda.

Se desarrolló una arquitectura basada en CRISP-DM, integrando la ingesta, almacenamiento, preparación, entrenamiento, evaluación y despliegue de los resultados.

Se identificaron patrones diferenciados en la mempool: K-Means presentó agrupamientos moderados, DBSCAN presentó un núcleo principal con microagrupamientos e Isolation Forest evidenció perfiles anómalos diferenciados frente al comportamiento normal.

Se evaluó la efectividad de los modelos mediante métricas de desempeño y benchmarks, permitiendo analizar los resultados y establecer criterios para la selección de la infraestructura de procesamiento.

Recomendaciones

Se recomienda ampliar las ventanas temporales para el entrenamiento de los modelos según como avance la ingesta de datos de la mempool, lo que permitiría analizar los datos en momentos de alta congestión y contrastar con los resultados de esta investigación.

Profundizar en la configuración de los hiperparámetros con varias ventanas de temporalidad de preferencia superiores a 10 días para una mejor caracterización de la variabilidad de la mempool de Bitcoin.

Incorporar nuevos modelos de Machine Learning no supervisado junto con métodos de explicabilidad para una investigación más profunda, así como también evaluar la incorporación del modelo LOF, para sistemas distribuidos.

En base a los benchmarks que toma la aplicación, definir de manera específica los criterios de selección de la infraestructura para el entrenamiento de los diferentes modelos y actualizar los criterios base que están definidos actualmente.

Referencias Bibliográficas

Achraf Yahia, Yassine Mouhssine, Abdelkader El Alaoui, & Said Ouatik El Alaoui. (2024). Exploring machine learning-based methods for anomalies detection evidence from cryptocurrencies. 20, 3951–3964. https://doi.org/10.1007/s41060-024-00703-w

Aragón, B. M., & Núñez, J. A. (2019). Estimation of the multivariate distribution of exchange rate yields against the dollar of the Cryptocurrencies Bitcoin, Ripple and Ether. Revista Mexicana de Economia y Finanzas Nueva Epoca, 14(3), 447–457. https://doi.org/10.21919/remef.v14i3.409

Ascanio, L. C. S., & García, J. A. A. (2020). Predicting the Price of Bitcoin, and more. Suma de Negocios, 11(24), 42–52. https://doi.org/10.14349/sumneg/2020.V11.N24.A5

Gundlach, R., Gijsbers, M., Koops, D., & Resing, J. (2021). Predicting confirmation times of Bitcoin transactions. https://jochen-hoenicke.de/queue/

Hasan, M., Rahman, M. S., Janicke, H., & Sarker, I. H. (2024). Detecting anomalies in blockchain transactions using machine learning classifiers and explainability analysis. Blockchain: Research and Applications, 5(3). https://doi.org/10.1016/j.bcra.2024.100207

Imtiaz, M. A., Starobinski, D., Trachtenberg, A., & Younis, N. (2021). Churn in the Bitcoin Network. IEEE Transactions on Network and Service Management, 18(2), 1598–1615. https://doi.org/10.1109/TNSM.2021.3050428

Jagannath, N., Barbulescu, T., Sallam, K. M., Elgendi, I., McGrath, B., Jamalipour, A., Abdel-Basset, M., & Munasinghe, K. (2021). An On-Chain Analysis-Based Approach to Predict Ethereum Prices. IEEE Access, 9, 167972–167989. https://doi.org/10.1109/ACCESS.2021.3135620

López-Zambrano, C. R., Camberos-Castro, M., & Villarreal-Peralta, E. M. (2021). The determinants of trust and perceived risk on bitcoin users. Retos(Ecuador), 11(22), 199–214. https://doi.org/10.17163/ret.n22.2021.01

Matzutt, R., Kalde, B., Pennekamp, J., Drichel, A., Henze, M., & Wehrle, K. (2021). CoinPrune: Shrinking Bitcoin's Blockchain Retrospectively. IEEE Transactions on Network and Service Management, 18(3), 3064–3078. https://doi.org/10.1109/TNSM.2021.3073270

Moreno-Arboleda, F. J., Rodríguez-Camacho, J. S., & Giraldo-Muñoz, D. (2021). Comparación de dos Plataformas de Blockchain: Bitcoin y Hyperledger Fabric. INGENIERÍA Y COMPETITIVIDAD, 24(1). https://doi.org/10.25100/iyc.v24i1.11027

Navas Bayona, W. I., Loor Zambrano, H. Y., & Amen Chinga, C. R. (2020). LA CONSOLIDACIÓN DEL BLOCKCHAIN EN LAS EMPRESAS COMO MÉTODO DE PAGO PARA SUS TRANSACCIONES. Investigación & Negocios, 13(22), 135. https://doi.org/10.38147/invneg.v13i22.108

Siddamsetti, S., Tejaswi, C., & Maddula, P. (2024). Anomaly Detection in Blockchain Using Machine Learning. 619–634. journal.esrgroups.org

Sharifi, A. S., & Ghorbanpour, M. Z. (2022). Proposing a multiple infrastructure model for the utilization of bitcoin. Retos(Ecuador), 12(23), 111–124. https://doi.org/10.17163/ret.n23.2022.07

Siddamsetti, S., & Srivenkatesh, M. (2023). Deep Blockchain Approach for Anomaly Detection in the Bitcoin Network. Original Research Paper International Journal of Intelligent Systems and Applications in Engineering IJISAE, 2024(1), 581–595. www.ijisae.org

S M Mostaq Hossain, & A. Altarawneh. (2025). Unraveling Ethereum's Mempool The Impact of Fee Fairness, Transaction Prioritization, and Consensus Efficiency. 2025 IEEE International Conference on Blockchain and Cryptocurrency (ICBC),. https://doi.org/10.110/ICBC64466.2025.11185068

Trivedi, S., Mehta, K., & Sharma, R. (2021). Systematic Literature Review on Application of Blockchain Technology in E-Finance and Financial Services. J. Technol. Manag. Innov., 16(3). http://jotmi.org

Vlahavas, G., Karasavvas, K., & Vakali, A. (2024). Unsupervised clustering of bitcoin transactions. Financial Innovation, 10(1). https://doi.org/10.1186/s40854-023-00525-y

Wang, X., Wang, C., Zhou, K., & Cheng, H. (2022). ESS: An Efficient Storage Scheme for Improving the Scalability of Bitcoin Network. IEEE Transactions on Network and Service Management, 19(2), 1191–1202. https://doi.org/10.1109/TNSM.2021.3127187

YAHIA Achraf, MOUHSSINE Yassine, ELALAOUI Abdelkader, & OUATIK ELALAOUI Said. (2024). Leveraging Machine learning for anomaly detection methods in cryptocurrency: A Data-Driven Study. 1–1. https://doi.org/10.1109/icoa62581.2024.10754421

Zheng, P., Zheng, Z., Wu, J., & Dai, H. N. (2020). XBlock-ETH: Extracting and exploring blockchain data from ethereum. IEEE Open Journal of the Computer Society, 1(1), 95–106. https://doi.org/10.1109/OJCS.2020.2990458

Zhang, Y., Li, Y., Zhang, J., & Tessone, C. J. (2025). Bitcoin transaction behavior modes exploration based on balance data. EPJ Data Science, 14(1). https://doi.org/10.1140/epjds/s13688-025-00564-8