ANÁLISIS COMPARATIVO DE COMUNIDADES EN ETHEREUM: DE POW A POS

A PREPRINT

Ignacio Argel
FIUBA
iargel@fi.uba.ar

Manuel V. Battan
FIUBA
mvbattan@fi.uba.ar

Lucas Bilo
FIUBA
lbilo@fi.uba.ar

Nahuel H. Spiguelman
FIUBA
nspiguelman@fi.uba.ar

May 29, 2024

ABSTRACT

El cambio de Proof of Work a Proof of Stake en la blockchain descentralizada de Ethereum tiene un impacto directo en la función desempeñada por los mineros, ya que son reemplazados por Stakers como nuevos validadores de la red. La disparidad en las dinámicas de este nuevo protocolo plantea interrogantes sobre si ha ocasionado modificaciones en el comportamiento de los usuarios que operan en él, ya sean stakers o participantes frecuentes de la red. Las investigaciones previas en Ethereum han priorizado aspectos técnicos, dejando un vacío en el análisis de interacciones entre cuentas y la formación de comunidades. Esto es especialmente notorio en la ausencia de estudios que aborden cómo el comportamiento de usuarios y comunidades ha cambiado con la migración de PoW a PoS. Para este análisis utilizaremos un modelo de Deep Learning cuya finalidad es crear embeddings de nodos que representen, de una manera inteligente, las relaciones entre addresses en la red. A su vez, el método propuesto es superador en términos de limitaciones computacionales y de utilización de features respecto a algoritmos tradicionales para analizar grafos.

1 Introducción

La blockchain Ethereum ha sido una de las innovaciones más significativas en el mundo de la tecnología financiera en la última década. Desde su lanzamiento en 2015, Ethereum ha operado bajo un protocolo de consenso conocido como Proof of Work (PoW), el cual ha demostrado ser robusto y confiable en la validación de transacciones y la seguridad de la red, pero presentando problemas como el alto consumo de energía y la centralización de la minería de bloques. Debido a estos problemas, Ethereum ha migrado hacia el protocolo de consenso conocido como Proof of Stake (PoS), el cual incluye varios cambios en la manera de garantizar la integridad durante las transacciones, mejorando la escalabilidad y la eficiencia de la red. Algunas preguntas significativas que surgen a partir del cambio de paradigma: ¿Existe un cambio en el comportamiento de los usuarios debido a esta transición? ¿Cuáles son los comportamientos invariantes o que siguen siendo vigentes después del cambio? ¿Cómo variaron las relaciones entre usuarios y comunidades?

El presente proyecto tiene como objetivo principal contestar estas preguntas. A través de un análisis exhaustivo, buscamos proporcionar una visión clara de los patrones de comportamiento de quienes operan sobre la red de Ethereum en sus dos protocolos de consenso principales, PoW y PoS. Al hacerlo, esperamos contribuir a la comprensión de las implicaciones de esta transición para la comunidad Blockchain, los desarrolladores y los usuarios de Ethereum.

1.1 Investigación

1.1.1 Artículos relacionados

Ethereum - Yellow paper
Para establecer una base sólida y comprender exhaustivamente los aspectos fundamentales de Ethereum, incluyendo su red y el proceso de transacciones, optamos por examinar el documento original de Ethereum.

ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER

El documento proporciona una riqueza de información crucial, detallando temas fundamentales como los campos de las transacciones, la naturaleza de los bloques, el protocolo de consenso (entonces basado en Prueba de Trabajo, PoW), y otros aspectos de suma importancia.

Artículos muy relacionados

Como una sugerencia inicial y como punto de partida para abordar el tema, se nos recomendó revisar el siguiente documento académico acerca de la detección de comunidades ilícitas en la red de Bitcoin. Aunque en sus conclusiones se menciona la falta de resultados satisfactorios, el documento proporciona una definición de comunidades, propone diversas técnicas y evalúa el rendimiento mediante métricas específicas. Una idea inicial era encarar el trabajo sobre esto mismo, pero en ethereum, relacionándolo con cambio de protocolo de consenso.

Illegal Community Detection in Bitcoin Transaction Networks

Luego de eso, buscamos artículos recientes que recopilaran algo de la situación actual científica respecto a la red de ethereum y su análisis como un grafo. Este paper nos dio un punto de partida en lo que respecta a la diversidad de tipos de cuentas y tipos de transacciones dentro de la red. Viendo esto, nos dimos cuenta que debíamos limitar el scope del grafo, ya que los distintos tipos de cuenta y transacciones se comportan de una manera muy diferente. Debido a esto, decidimos focalizarnos en las transacciones que tienen un value, transaccionado el token Ether.

Graph Analysis of the Ethereum Blockchain Data:A Survey of Datasets, Methods, and Future Work

El último paper y estrechamente vinculado que hemos identificado se enfoca en la identificación de comunidades dentro de redes sociales. En este contexto, se sugiere la aplicación de la matriz de covarianza de salida de baja dimensionalidad sobre los datos de transacciones, seguido por la extracción de los K autovectores. Posteriormente, se emplea el algoritmo de K-means para la partición de los nodos en K grupos distintos. Es interesante que la conclusión y las pruebas definitivas, se hacen sobre un grafo de tamaño reducido.

Community Detection in Blockchain Social Networks

Una hipótesis surgida de estos análisis es si es la mejor opción buscar comunidades utilizando K-means, ya que el algoritmo obligatoriamente separa en K clusters, sin dejar a nadie fuera de una comunidad.

1.1.2 Papers complementarios

Si bien estos papers no fueron foco principal de nuestra investigación, los revisamos de forma rápida viendo si encontrábamos insights inspiradores o novedosos.

Este paper es uno de los primeros trabajos científicos que definen una comunidad, lo que la hace una lectura muy interesante.

Community structure in social and biological networks

En el siguiente se presenta el primer estudio empírico a gran escala sobre el uso de Inline Assembly en contratos inteligentes de Ethereum, analizando millones de contratos desde diversas perspectivas técnicas. Proporciona nuevas observaciones y percepciones valiosas para el desarrollo de contratos inteligentes y la evolución de Solidity y sus compiladores.

Large-Scale Empirical Study of Inline Assembly on 7.6 Million Ethereum Smart Contracts

******INSERTE SUS PAPERS LEIDOS ACA*******

1.2 Materiales y métodos

1.2.1 Fuente de datos

La primera complicación con la que nos enfrentamos fue la búsqueda de una fuente de datos íntegra, sin limitaciones significativas en términos de cantidad y accesibilidad. Inicialmente, exploramos la posibilidad de utilizar APIs con el propósito de obtener muestras extensas y almacenarlas. Sin embargo, la restricción en el número de solicitudes permitidas, junto con la complejidad inherente de realizar una cantidad considerable de peticiones y la gestión de almacenamiento para tal volumen de datos, nos llevó a desechar esta estrategia.

Ejemplos de APIs revisadas (entre otras):

- Blockchain Data
- Etherscan

Posteriormente, consideramos la posibilidad de replicar un nodo. Contábamos con diversas alternativas, si bien replicar un nodo plantea ciertas dificultades técnicas y de almacenamiento. Las alternativas que evaluamos fueron las siguientes:

- Nodo completo (Full node): Este tipo de nodo conserva y administra una réplica integral de la cadena de bloques de Ethereum, aunque con ciertos recortes (pruning). Esto abarca todas las transacciones efectuadas en la red, así como el estado actualizado de todas las cuentas y contratos inteligentes hasta un punto en el tiempo determinado. La implementación de este tipo de nodo requiere una capacidad de almacenamiento de aproximadamente 1.5 TB.
- Nodo de archivo (Archive node): Se trata de una categoría de nodo que preserva todo el historial completo de la cadena de bloques desde su bloque inicial hasta el más reciente. Esto incluye todas las transacciones, contratos inteligentes, eventos y estados históricos de todas las cuentas. La variante más ligera de este tipo de nodo requiere alrededor de 10 TB de almacenamiento hasta mayo de 2022.

Sin embargo, esta opción fue desechada debido a la limitación de recursos de almacenamiento disponibles y a la falta de necesidad de poseer los datos en su totalidad.

Por último, hemos identificado un dataset en BigQuery que, aunque no está sincronizado en tiempo real, se actualiza diariamente. Las consultas realizadas en este conjunto de datos son gratuitas y, para propósitos prácticos, su límite es inalcanzable. Los campos disponibles en este conjunto de datos son los siguientes:

- **block_hash**: El identificador único de un bloque en la blockchain, el cual es generado a partir de la información contenida en el bloque mediante una función de hash.
- **block_number**: El número de secuencia del bloque dentro de la blockchain.
- **block_timestamp**: Es la hora y fecha exacta en la que el bloque fue minado.

- **transaction_hash**: El identificador único de una transacción específica, el cual es generado a partir de la información de la transacción mediante una función de hash.
- **transaction_index**: La posición de la transacción dentro del bloque.
- **nonce**: Un número utilizado para garantizar que cada transacción sólo pueda ser procesada una vez. Es específico de la cuenta del remitente y se incrementa con cada transacción enviada desde esa cuenta.
- **from_address**: La dirección de Ethereum del remitente de la transacción.
- **to_address**: La dirección de Ethereum del destinatario de la transacción. Puede ser la dirección de un usuario o la dirección de un smart contract.
- **value**: Cantidad de Ether transferida en la transacción, está expresada en wei.
- **value_lossless**: Es el mismo valor que el campo "value" pero está expresado de otra forma para evitar cierta precisión en algunas aplicaciones.
- **gas**: es la tarifa que se paga en la transacción. Está expresada en wei.
- **gas_price**: Es el precio en Ether de una unidad de gas usada en la transacción. Está expresada en wei.
- **max_fee_per_gas**: es la cantidad máxima de gas que el usuario remitente está dispuesto a pagar para que su transacción se incluya en un bloque. Está expresada en wei.
- **max_priority_fee_per_gas**: Es determinada por el usuario remitente y es opcional. Indica la cantidad de gas que se está dispuesto a pagar a los mineros como "propina" por la transacción. Está expresada en wei.
- **transaction_type**: Indica el tipo de la transacción. Puede ser 0 para transacciones regulares o 2 para transacciones EIP-1559.
- **chain_id**: Identificador único para la cadena de bloques (Ethereum Mainnet, Testnet, entre otros).
- **r y s**: Son componentes de la firma digital de la transacción. En conjunto se utilizan para verificar que la transacción fue firmada por el propietario de la clave privada correspondiente a la dirección de origen de la transacción.
- **v**: Es un parámetro de la firma digital de la transacción, se usa para la recuperación de la clave pública.
- **y_parity**: Es un parámetro de la firma digital de la transacción y es una versión más compacta y moderna de lo que indica el campo v. Se introdujo en las transacciones de tipo EIP-1559 y EIP-2930.

A continuación, se adjunta una transacción con datos reales:

"block_hash": "0xbe6dd03c251417b51020f5358c5981fd1627575c256dd98d6d24dd7687997d32",
"block_number": "17329020",
"block_timestamp": "2023-05-24 12:23:35.000000 UTC",
"transaction_hash": "0x5fe28c92492cfd4107b867da66036cf108a2a7b09baed29fdc543d2ba5778574",
"transaction_index": "79",
"nonce": "4894815",
"from_address": "0x56eddb7aa87536c09ccc2793473599fd21a8b17f",
"to_address": "0x86ba5d7600e1ce97fdf821e96a0b4ec729843ab2",
"value": "9208340000000000",
"value_lossless": "9208340000000000",
"gas": "207128",
"gas_price": "32342796700",
"input": "0x",
"max_fee_per_gas": "102000000000",
"max_priority_fee_per_gas": "2000000000",
"transaction_type": "2",
"chain_id": "1",
"access_list": [],
"r": "0x8e7de580875602ab722aaeb890eb2bfb88443e34338fd2936408e89d4e5b456c",
"s": "0x937cd39f66fb771a8c1216b296f1c32b42c5dc013054b808596cad794ca084a",
"v": "0x0",
"y_parity": null

Con acceso a este conjunto de datos y la capacidad de realizar consultas sobre cualquier ventana temporal desde el inicio de la cadena hasta el día anterior al actual, tenemos la información necesaria para nuestros análisis.

1.2.2 node2vec

node2vec es un algoritmo para aprendizaje de representaciones de nodos en redes. Su objetivo es mapear los nodos de un grafo a un espacio vectorial continuo de baja dimensión, preservando las propiedades y estructuras del grafo original. Funciona en dos fases principales:

Generación de Trayectorias Aleatorias: node2vec utiliza una combinación de técnicas de paseo aleatorio (random walks) sesgado para explorar el grafo. Ajusta dos parámetros, p y q, que permiten controlar el equilibrio entre la exploración de nuevas partes del grafo (similar a un paseo de profundidad primero, DFS) y la explotación de áreas cercanas al nodo inicial (similar a un paseo de amplitud primero, BFS). El parámetro p controla la probabilidad de regresar al nodo anterior, y q ajusta la probabilidad de visitar nodos más alejados.

Optimización de la Representación: Una vez generadas las trayectorias, node2vec aplica técnicas de aprendizaje no supervisado similares a las usadas en el procesamiento de lenguaje natural, como Skip-gram, para optimizar los vectores de los nodos. El objetivo es maximizar la probabilidad de co-ocurrencia de nodos que están cerca en las trayectorias generadas.

De esta manera, node2vec logra capturar tanto la estructura local como global del grafo, produciendo representaciones vectoriales útiles para tareas posteriores como la clasificación de nodos, detección de comunidades y predicción de enlaces.

1.2.3 NetworkX

NetworkX es un paquete de Python para la creación, manipulación y estudio de la estructura, dinámica y funciones de redes complejas. Es ampliamente utilizado en investigación y aplicaciones prácticas debido a su versatilidad y facilidad de uso. NetworkX permite trabajar con grafos (redes) de cualquier tamaño y tipo, incluidos grafos dirigidos, no dirigidos y multigrafos. Ofrece una amplia gama de algoritmos para análisis de redes, como medidas de centralidad, detección de comunidades, algoritmos de búsqueda de caminos, y más. Además, se integra bien con otras bibliotecas científicas de Python, como NumPy, SciPy y Matplotlib, facilitando el análisis y visualización de datos de redes.

1.2.4 UMAP

UMAP, Uniform Manifold Approximation and Projection, es un algoritmo de reducción de dimensionalidad no lineal que destaca por su eficiencia y habilidad para preservar las estructuras y relaciones locales en datos de alta dimensión. A través de la construcción de un grafo ponderado basado en la conectividad local de los puntos en el espacio de alta dimensión, UMAP busca optimizar una representación en un espacio de menor dimensión que conserve estas relaciones. Su capacidad para capturar estructuras complejas y no lineales, combinada con su eficaz preservación de la estructura global y local de los datos, lo convierte en una herramienta poderosa para visualizar y explorar conjuntos de datos grandes y complejos, haciéndolo especialmente relevante en campos como la ciencia de datos y el aprendizaje automático.

1.3 Pregunta a responder

Durante la etapa inicial de revisión de literatura relacionada con la temática, percibimos, por lo expuesto en éstas, una notable carencia de investigación exhaustiva sobre el análisis de comunidades en Ethereum. En general, los estudios más completos y con mejores resultados se han desarrollado predominantemente sobre la blockchain de Bitcoin.

Al inicio de nuestra investigación, surgió la idea de analizar el comportamiento de los usuarios en la red de Ethereum en relación con el cambio de protocolo de consenso implementado en septiembre de 2022.

Como mencionamos previamente, este cambio introduce una diferencia tanto técnica como teórica en el proceso de aprobación de nuevos bloques de transacciones dentro de la red.

Dicha diferencia podría ser un factor motivador en la modificación del comportamiento de los usuarios que utilizan la red diariamente para realizar acuerdos o transferencias de activos.

Por lo tanto, la pregunta principal que se busca responder en este trabajo es: ¿Realmente se produjo un cambio en la composición de las comunidades? ¿Cómo ha impactado este cambio en el uso de la blockchain de Ethereum, considerando que el nuevo protocolo permite (y además fomenta) una mayor participación general de los usuarios en la comunidad?

1.4 Resultado esperado

Para responder a las preguntas planteadas, es necesario desarrollar una serie de conceptos que abarquen desde la formulación de nuestro propio concepto de comunidad hasta los métodos para identificarlas.

En este estudio, al igual que en muchos de los trabajos previamente revisados, se considerará a priori como característica principal de la relación entre dos nodos, la cantidad de transacciones entre ellos. Esta relación puede no ser simétrica y, si es necesario, podríamos incluir otros datos transaccionales que aporten valor a la hora de definir la importancia de la misma.

Con base en esta relación, se buscará identificar comunidades en las que las interacciones entre dos o más nodos pertenecientes a la misma comunidad sean más significativas que las interacciones con nodos externos a dicha comunidad. Cabe mencionar que la definición de comunidades puede ser difusa, dependiendo del concepto inicial de comunidad, del método de definición y de la naturaleza del problema. Además, adherimos a la idea presentada en la literatura revisada, de que un nodo no necesariamente debe pertenecer a una comunidad. Si pertenece, es a una única comunidad, pero podría no estar en ninguna debido a la baja calidad de sus relaciones con otros nodos de la red.

Con esto en mente, se propone analizar cualitativa y cuantitativamente las comunidades detectadas y su composición, en un periodo de tiempo determinado, antes y después de la implementación del cambio al protocolo de consenso Proof of Stake.

Para comprender mejor el problema, es esencial conocer los datos, lo que permitirá formular preguntas más específicas y refinar las hipótesis a medida que se analicen los datos y, consecuentemente, las comunidades. Finalmente, todo esto contribuirá a construir un método para agrupar distintos nodos en un grafo de transacciones de Ethereum, permitiendo identificar sus comunidades.

2 Metodología

2.1 Análisis exploratorio

Una vez conseguidos los datos, debíamos empezar a analizar los mismos, con el fin de conocer mejor los datos, su evolución y en base a eso tomar la decisión de que subset de datos pre y post cambio de protocolo de consenso vamos a elegir para la comparación final.

Para esto, decidimos plantear una serie de métricas:

- Cantidad de nodos
- Cantidad de transacciones
- Gas
- Value
- Grados de los nodos (in/out)
- Pagerank
- Métricas calculadas mediante el algoritmo HITS

Todas estas métricas se calcularon para los años 2022 y 2023 de forma agregada por semana. Las agregaciones ploteadas en este caso son mínimo, máximo, media y desviación estándar.

Usando esto nos apoyamos para un análisis exploratorio y la toma de la decisión de en qué ventana de tiempo nos íbamos a centrar.

2.2 Análisis de transacciones

En el siguiente gráfico se presenta la evolución diaria de la cantidad de transacciones desde enero de 2020 hasta finales de 2023. Se señala el lanzamiento de la versión de Proof of Stake y se marcan con una línea negra punteada los periodos que se analizarán. Hemos seleccionado estos intervalos para mantener la estacionalidad temporal, ya que corresponden a la misma época del año, pero con un cambio en el protocolo. Esto nos permite comparar principalmente el comportamiento de los usuarios frente al nuevo protocolo en comparación con el de Proof of Work.

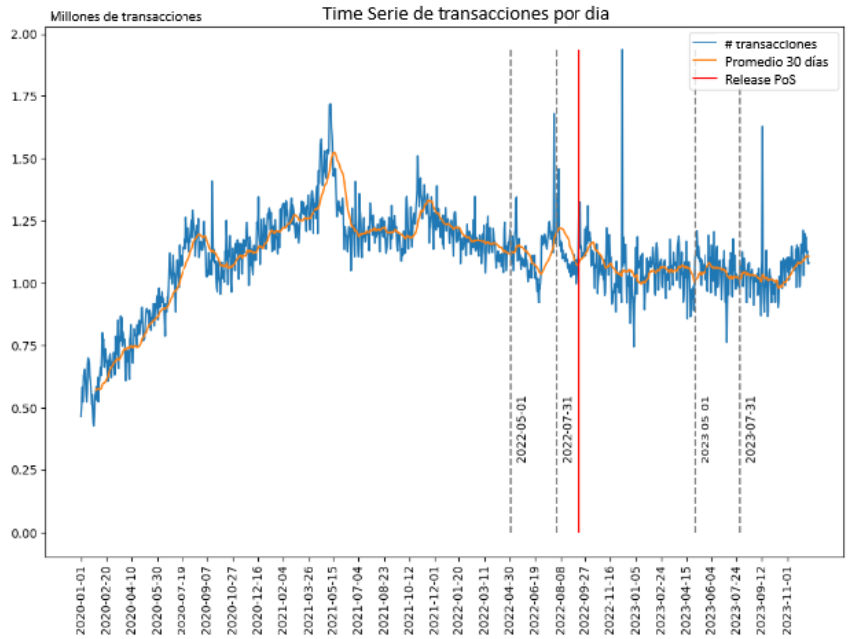

Figure 1: Evolución con media móvil de la cantidad de transacciones por dia.

Una vez elegidos los intervalos, procedemos a compararlos analiticamente mediante el uso de las métricas previamente nombradas.

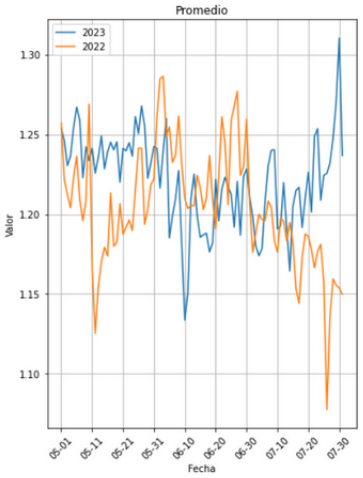

Figure 2: Promedio transacciones totales.

Este gráfico presenta la evolución del promedio de transacciones (in+out) por nodo activo por día para ambos periodos.

En conjunto con lo observado en el gráfico de la cantidad de transacciones diarias, se evidencia que, en el periodo de 2022, hay un incremento en la cantidad total de transacciones hacia el final del periodo, mientras que el promedio de transacciones por nodo disminuye. Esto sugiere una centralización de la red al final de este periodo, muy probablemente debido a factores externos, como noticias y fluctuaciones en el precio.

Continuando con el análisis de estos periodos, observamos en los gráficos de la desviación estándar que los valores son significativamente superiores en órdenes de magnitud a los de la media. Esto indica que, aunque la media

se aproxima a 1, la desviación estándar es elevada, lo que sugiere la existencia de numerosos casos en los que la cantidad de transacciones, tanto recibidas como enviadas, es exorbitante.

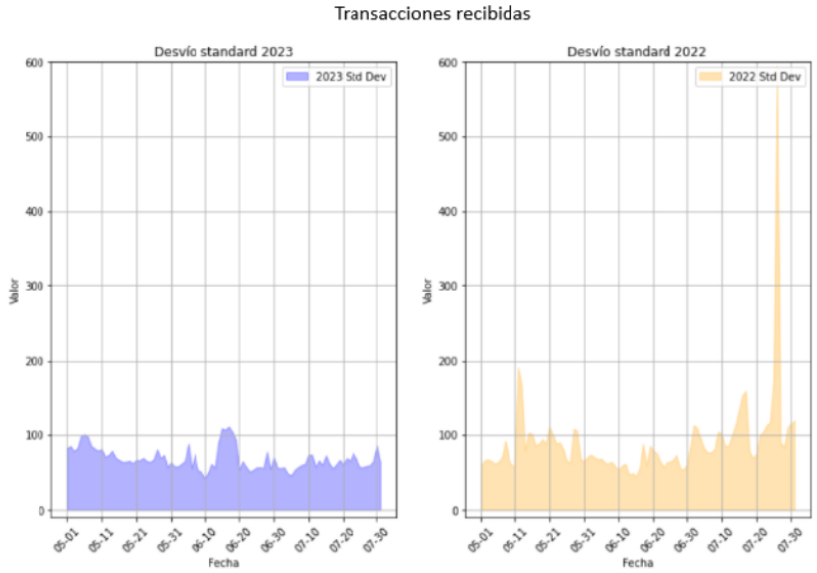

Figure 3: Evolución del desvió estándar para transacciones recibidas.

Este gráfico respalda la hipótesis de que, a finales de julio, la red experimentó una centralización, evidenciada por un valor atípico en la desviación estándar de la cantidad de transacciones recibidas. Esto sugiere que algunos nodos concentraron una proporción significativa de las transacciones realizadas como destinatarios.

En el caso de las transacciones realizadas, se observa un valor considerablemente más bajo en general para las desviaciones. Esto podría indicar que la realización de transacciones es mucho más homogénea que la recepción de las mismas. Sin embargo, se observa un pico en la desviación de las transacciones realizadas el 10 de junio de 2023.

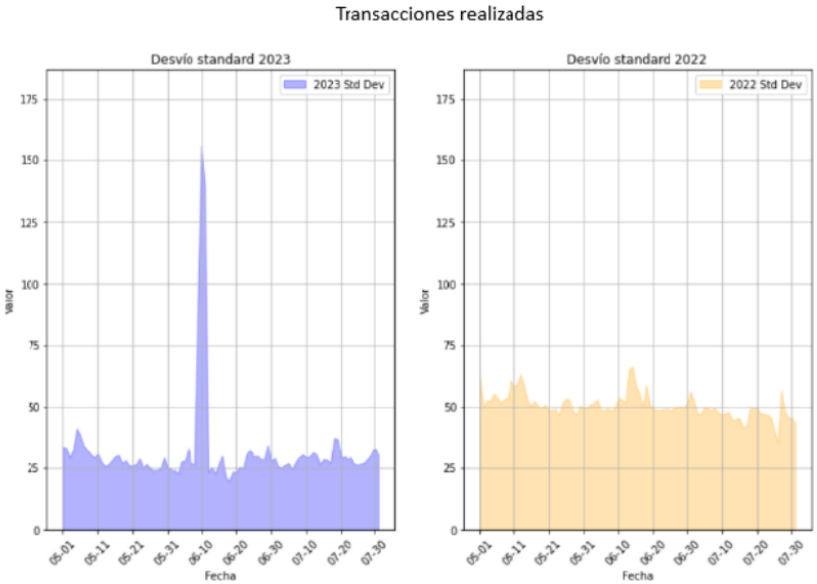

Figure 4: Evolución del desvió estándar para transacciones realizadas.

Como último aspecto a destacar del gráfico de desviación en transacciones realizadas, se puede observar que la desviación promedio disminuyó, pasando de aproximadamente 50 a un poco más de 25. Esto también indica una homogeneización en la red tras el cambio de protocolo.

2.3 Gephi

El análisis exploratorio de datos es una fase crucial en la investigación y comprensión de estructuras complejas, como las redes transaccionales. En este contexto Gephi, un software de visualización de redes, permite representar de manera gráfica y comprensible la composición y dinámica de las redes.

Gephi funciona mediante la manipulación de datos de red, donde los nodos representan address de wallet y las aristas reflejan las transacciones entre ellas. Para llevar a cabo esta representación visual, Gephi ofrece una variedad de algoritmos de posicionamiento de nodos en el espacio visual, lo que facilita la identificación de patrones, comunidades y centralidades dentro de la red.

Estos algoritmos utilizan técnicas de optimización y modelado matemático para distribuir los nodos de manera óptima en el espacio visual, teniendo en cuenta diversas métricas y atributos de la red, como la distancia entre nodos, la fuerza de las relaciones y la distribución de la densidad de conexiones.

Al emplear Gephi en nuestro análisis exploratorio, hemos optado por aprovechar estas funcionalidades para obtener una representación visual clara y significativa de la composición de la red en cuestión. Esta herramienta nos permite no solo visualizar la red en su totalidad, sino también explorar detalles específicos, identificar subgrupos o comunidades y detectar patrones emergentes que podrían pasar desapercibidos en un análisis puramente estadístico.

A continuación, exponemos ejemplos visualizados utilizando esta herramienta, acompañados de los supuestos y cuestionamientos que suscitaron.

Como prueba inicial, optamos por representar visualmente un grafo conformado por los nodos que han participado en más de 10 transacciones enviadas o recibidas durante el periodo comprendido entre mayo y junio de 2022. Esta selección se justifica debido a que la cantidad de nodos involucrados en la red torna impracticable su visualización sin un filtrado previo. Para esta prueba utilizamos Force Atlas 2, debido a su rápida convergencia, aun con data extensa.

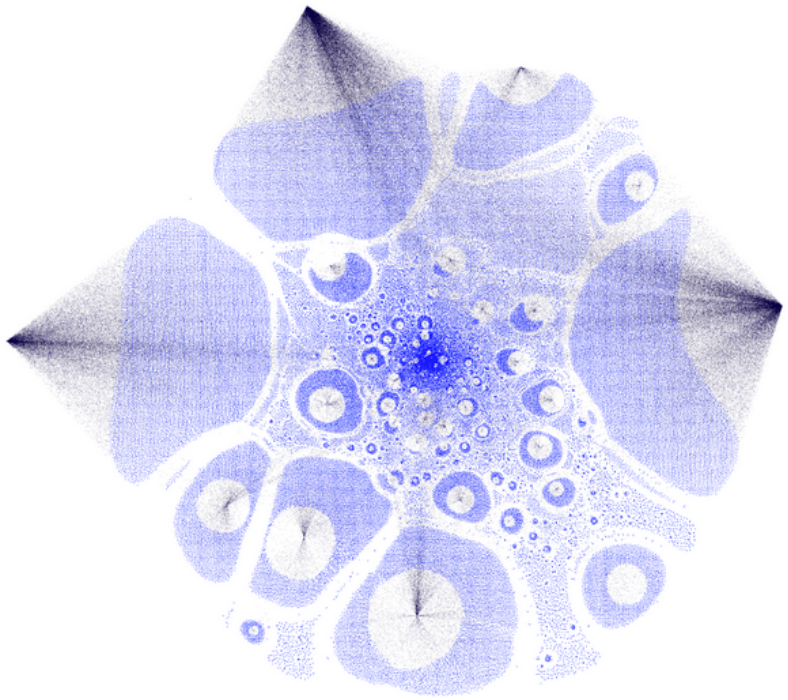

Figure 5: Visualización de transacciones utilizando Gephi con Force Atlas 2.

3 Modelos

3.1 Introducción

Dada la investigación hecha al inicio del trabajo, nos dimos cuenta que una buena aproximación para atacar el problema es generar embeddings y sobre estos, ejecutar algoritmos de clustering. Esta forma de proceder dio buenos resultados en varios de los papers analizados previamente, pero con varias limitaciones de procesamiento. Una primera aproximación fue hacer uso de modelos como node2vec, el cual trabaja generando embeddings de nodo teniendo en cuenta las relaciones que tiene con los demás agentes en la red, y sobre el resultado obtenido intentar hacer clustering como kmeans o hdbscan o simplemente utilizar algoritmos de búsqueda de comunidades como lo es Louvain o Infomap. Esto nos dio buenos resultados en el proceso de análisis pero rápidamente nos limitó para escalar la solución a varios días de transacciones que es necesario para intentar detectar comunidades en ventanas temporales no tan pequeñas. Para solucionar este problema intentamos generar embeddings de una forma robusta y escalable sin que ésta dependiera directamente de una limitación de hardware. Es así que iniciamos el desarrollo de un modelo de Deep Learning que genere los embeddings en reemplazo de node2vec, salvando las limitaciones previamente mencionadas.

3.2 Heurística para el filtrado de datos

Utilizamos cierto filtrado en los datos, para poder evitar aquellas transacciones más aisladas que nos podrían traer cierto ruido a los modelos. El filtrado fue por un mínimo de transacciones, el cual tiene en cuenta para cada nodo tanto las transacciones de remitente como las de destinatario. Este mínimo lo obtuvimos de la siguiente manera:

Dada la porción de datos a analizar, calculamos la cantidad total de transacciones para cada nodo. Luego, dado un porcentaje (entre 0 y 1) y el total de transacciones, se calcula un umbral con el cual buscamos el máximo de transacciones de los nodos que se encuentran debajo de este umbral. En otras palabras, buscamos la máxima cantidad de transacciones de los nodos que se encuentran debajo de un porcentaje del total de transacciones.

Finalmente, este máximo que obtenemos será utilizado como valor mínimo para eliminar aquellas transacciones que tengan menos de esta cantidad de transacciones.

3.3 Uso de *representation learning* sobre la red

En el camino de búsqueda de comunidades, un camino a desarrollar es aplicar algoritmos de representation learning para generar un espacio vectorial que mantenga las propiedades relacionales que presenta la red originalmente.

Inicialmente decidimos hacer una primera prueba de concepto utilizando node2vec en base a los relativos buenos resultados expuestos en los trabajos previamente analizados. Para este caso en particular, nosotros generamos el grafo de transacciones, con una ventana de tiempo, utilizando NetworkX. Con este grafo, ejecutamos node2vec con el fin de tener estas representaciones para cada nodo con participación activa durante este periodo.

Los primeros resultados fueron satisfactorios, nos permitieron generar embeddings de algunos pocos días con un tiempo de procesamiento considerable para poder analizar una muestra inicial pequeña del comportamiento de los nodos de la red, tal como se vio en varios de los trabajos analizados en la primera instancia.

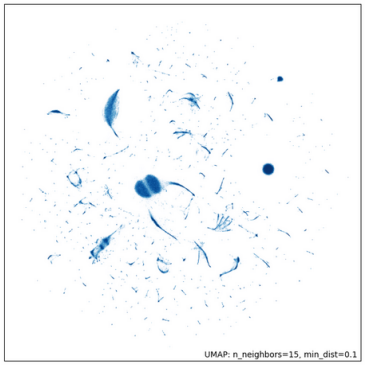

Figure 6: Visualización de transacciones utilizando Gephi con Force Atlas 2.

Se adjunta un ejemplo de ejecución para 12 horas de transacciones (~ 500.000 transacciones) visualizada mediante la representación comprimida resultante del algoritmo UMAP.

Las limitaciones aparecieron cuando quisimos aumentar el tamaño de esa ventana temporal para poder analizar varios días de transacciones, siendo que node2vec hace uso intensivo de la memoria de la máquina en la que se está corriendo. Por ejemplo, hicimos una prueba con 2 días (1 y 2 de marzo de 2022) y el algoritmo no llegó a un resultado satisfactorio por una deficiencia en el uso de la memoria.

4 Proceso de validación