MÁSTER UNIVERSITARIO EN TECNOLOGÍAS FINANCIERAS: PAGOS Y BANCA DIGITAL

TRABAJO FIN DE MÁSTER

NextGen AML: ecosistema adversarial de IA para la detección y anticipación del blanqueo de capitales con GNNs y Reinforcement Learning

Autor: Álvaro Roldán Márquez
Director: José Ignacio Núñez García

Madrid
Julio de 2026

NEXTGEN AML: ECOSISTEMA ADVERSARIAL DE IA PARA LA DETECCIÓN Y ANTICIPACIÓN DEL BLANQUEO DE CAPITALES CON GNNS Y REINFORCEMENT LEARNING

Autor: Álvaro Roldán Márquez
Director: José Ignacio Núñez García

RESUMEN DEL PROYECTO

El blanqueo de capitales constituye uno de los grandes retos de la industria financiera, y su prevención se enfrenta a una limitación clara: los sistemas antiblanqueo tradicionales, basados en reglas rígidas, son fundamentalmente reactivos. Y es que estos sistemas se limitan a reconocer patrones ya conocidos y generan, además, una enorme cantidad de falsas alarmas que saturan a los equipos de cumplimiento, mientras que las organizaciones criminales evolucionan de forma constante y van siempre un paso por delante. De esta problemática surge la pregunta que motiva el presente trabajo: si, en lugar de esperar a reconocer el fraude, fuera posible anticiparse a él.

Con ese propósito, este Trabajo Fin de Máster propone NextGen AML, un ecosistema adversarial de inteligencia artificial orientado no solo a detectar, sino sobre todo a anticipar el blanqueo de capitales. La propuesta se articula en torno a tres componentes que interactúan entre sí. El primero es un agente atacante que, mediante aprendizaje por refuerzo (Reinforcement Learning), aprende a generar y camuflar distintas tipologías de blanqueo. El segundo es un agente defensor que, combinando redes neuronales de grafos (GNNs) y Transformers, detecta las operaciones sospechosas atendiendo tanto a la estructura de la red como al ritmo temporal de las transacciones. Y el tercero es un entorno de simulación, denominado Financial Sandbox, que genera el escenario financiero sintético sobre el que ambos agentes operan. La clave del planteamiento reside en que estos dos agentes coevolucionan, es decir, mejoran enfrentándose el uno al otro, de manera que el defensor aprende a reconocer amenazas cada vez más sofisticadas a medida que el atacante perfecciona sus estrategias.

Además de su desarrollo conceptual, la propuesta se acompaña de una prueba de concepto funcional, implementada en Google Colab, que traslada esta propuesta a código y la pone a funcionar de principio a fin. En ella, el Financial Sandbox genera un sistema financiero sintético con sus clientes, entidades y transacciones; el agente atacante introduce sobre él la actividad ilícita; y el agente defensor trata de detectarla, todo ello culminando en un proceso de coevolución entre ambos. Los resultados obtenidos muestran que las distintas tecnologías funcionan y, sobre todo, que interactúan de forma coherente, dando lugar a una dinámica de mejora mutua que reproduce de manera realista la tensión entre quien blanquea y quien detecta.

Conviene precisar, no obstante, el alcance del trabajo. NextGen AML no constituye una solución lista para desplegarse en una entidad real, sino una propuesta conceptual respaldada por una prueba de concepto de pequeña escala, construida sobre datos sintéticos. Su valor, por tanto, no reside en ofrecer un producto terminado, sino en demostrar de forma honesta que el enfoque es viable y en abrir un camino sobre el que seguir desarrollando esta idea. En esta línea, el trabajo proyecta también su evolución futura hacia el paradigma del Compliance-as-Code y hacia un escalado con datos reales.

En definitiva, este trabajo plantea un cambio de perspectiva en la prevención del blanqueo de capitales: el paso de una defensa que reacciona a una defensa que se anticipa. En una era en la que la tecnología, y con ella la inteligencia artificial, evoluciona a un ritmo cada vez mayor, y en la que también lo hacen los métodos empleados por los delincuentes, la mejor forma de no ir por detrás es aprender a adelantarse.

Palabras clave: blanqueo de capitales, inteligencia artificial, aprendizaje por refuerzo, redes neuronales de grafos, Transformers, sistema adversarial, Compliance-as-Code.

NEXTGEN AML: ECOSISTEMA ADVERSARIAL DE IA PARA LA DETECCIÓN Y ANTICIPACIÓN DEL BLANQUEO DE CAPITALES CON GNNS Y REINFORCEMENT LEARNING

Author: Álvaro Roldán Márquez
Supervisor: José Ignacio Núñez García

ABSTRACT

Money laundering is one of the major challenges facing the financial industry, and its prevention comes up against a clear limitation: traditional anti-money laundering systems, based on rigid rules, are fundamentally reactive. And the fact is that these systems merely recognise already known patterns and, moreover, generate a huge number of false alarms that overwhelm compliance teams, while criminal organisations evolve constantly and are always one step ahead. From this problem arises the question that motivates this work: whether, instead of waiting to recognise fraud, it might be possible to anticipate it.

With that aim, this Master's Thesis proposes NextGen AML, an adversarial artificial intelligence ecosystem geared not only towards detecting money laundering, but above all towards anticipating it. The proposal is built around three components that interact with one another. The first is an attacker agent that, through Reinforcement Learning, learns to generate and disguise different money laundering typologies. The second is a defender agent that, combining Graph Neural Networks (GNNs) and Transformers, detects suspicious operations by taking into account both the structure of the network and the temporal rhythm of the transactions. And the third is a simulation environment, called the Financial Sandbox, which generates the synthetic financial scenario on which both agents operate. The key to the approach lies in the fact that these two agents coevolve, that is, they improve by confronting each other, so that the defender learns to recognise increasingly sophisticated threats as the attacker refines its strategies.

In addition to its conceptual development, the proposal is accompanied by a functional proof of concept, implemented in Google Colab, which translates this proposal into code and puts it to work from start to finish. In it, the Financial Sandbox generates a synthetic financial system with its clients, entities and transactions; the attacker agent introduces the illicit activity into it; and the defender agent tries to detect it, all of this culminating in a process of coevolution between the two. The results obtained show that the different technologies work and, above all, that they interact coherently, giving rise to a dynamic of mutual improvement that realistically reproduces the tension between the one who launders and the one who detects.

It is worth clarifying, however, the scope of the work. NextGen AML is not a solution ready to be deployed in a real institution, but rather a conceptual proposal backed by a small-scale proof of concept, built on synthetic data. Its value, therefore, does not lie in offering a finished product, but in honestly demonstrating that the approach is viable and in opening up a path along which to keep developing this idea. Along these lines, the work also projects its future evolution towards the Compliance-as-Code paradigm and towards scaling up with real data.

In short, this work puts forward a change of perspective in the prevention of money laundering: the shift from a defence that reacts to a defence that anticipates. In an era in which technology,

and with it artificial intelligence, is evolving at an ever-increasing pace, and in which the methods used by criminals are evolving too, the best way not to fall behind is to learn to get ahead.

Keywords: money laundering, artificial intelligence, reinforcement learning, graph neural networks, Transformers, adversarial system, Compliance-as-Code.

Índice

Índice de figuras

Índice de tablas

Glosario de acrónimos

Sigla	Significado
AML	Anti-Money Laundering (prevención del blanqueo de capitales)
AMLA	Anti-Money Laundering Authority (Autoridad de Lucha contra el Blanqueo de Capitales y la Financiación del Terrorismo)
CDD	Customer Due Diligence (diligencia debida con respecto al cliente)
CPU	Central Processing Unit (unidad central de procesamiento)
CSV	Comma-Separated Values (valores separados por comas)
EDD	Enhanced Due Diligence (diligencia debida reforzada)
F1	F1-score (métrica que combina la precisión y la sensibilidad)
FATF	Financial Action Task Force (denominación en inglés del GAFI)
GAFI	Grupo de Acción Financiera Internacional (FATF en inglés)
GAT	Graph Attention Network (red de atención sobre grafos)
GCN	Graph Convolutional Network (red convolucional sobre grafos)
GNN	Graph Neural Network (red neuronal de grafos)
HTML	HyperText Markup Language (lenguaje de marcado de hipertexto)
IA	Inteligencia Artificial
KYC	Know Your Customer (conocimiento del cliente)
OCDE	Organización para la Cooperación y el Desarrollo Económicos (OECD en inglés)
ODS	Objetivos de Desarrollo Sostenible
PIB	Producto Interior Bruto
PSD2	Payment Services Directive 2 (segunda Directiva de Servicios de Pago)
RegTech	Regulatory Technology (tecnología aplicada al cumplimiento regulatorio)
RL	Reinforcement Learning (aprendizaje por refuerzo)
SupTech	Supervisory Technology (tecnología aplicada a la supervisión)
TFM	Trabajo Fin de Máster
UBO	Ultimate Beneficial Owner (titular real)
UE	Unión Europea
UNODC	United Nations Office on Drugs and Crime (Oficina de las Naciones Unidas contra la Droga y el Delito)

1. Introducción

1.1. La transformación digital del sistema financiero

Esta transformación ha venido impulsada por una serie de fenómenos que se refuerzan entre sí. El Open Banking ha conectado a las entidades con aplicaciones de terceros, de modo que los servicios ya no se prestan desde una única plataforma, sino que están repartidos entre múltiples sistemas. Los pagos instantáneos han reducido los tiempos de liquidación a prácticamente cero, permitiendo que una transferencia se complete en segundos y pueda atravesar distintas jurisdicciones antes de ser sometida a revisión. Por su parte, la consolidación de las criptomonedas ha abierto la posibilidad de transferir valor al margen del sistema bancario tradicional, incorporando niveles de pseudoanonimato que hace apenas unos años resultaban impensables.

Todo ello ha supuesto una mejora evidente para el usuario, que hoy dispone de servicios más rápidos, accesibles y económicos debido al aumento significativo de la competencia. Sin embargo, esta misma accesibilidad ha generado que dicho entorno sea considerablemente más difícil de supervisar. Cada nueva capa de inmediatez e interconexión amplía las oportunidades para el usuario legítimo, pero también incrementa el margen de actuación de quienes buscan explotar el sistema con fines ilícitos. Es importante partir de esta idea, ya que la transformación digital no constituye un mero telón de fondo, sino el elemento que explica por qué unos mecanismos de control diseñados para un entorno más lento han comenzado a quedar obsoletos.

1.2. El blanqueo de capitales y las limitaciones de los sistemas AML actuales

Dentro de este escenario, el blanqueo de capitales se ha consolidado como uno de los principales desafíos del sistema financiero, tanto por su alcance como por la sofisticación que ha alcanzado. Las estimaciones señalan que cada año se blanquea entre el 2 % y el 5 % del producto interior bruto (PIB) mundial (Oficina de las Naciones Unidas contra la Droga y el Delito [UNODC], s.f.), una cifra que permite hacerse una idea de la enorme cantidad de dinero que la economía criminal consigue infiltrar en el sistema financiero legal. Además, no se trata de un fenómeno estable, sino que está en constante evolución: las técnicas empleadas por los criminales han avanzado de igual manera que la transformación del propio sistema financiero. En la actualidad, el blanqueo rara vez se basa en operaciones simples o fácilmente identificables, sino en estructuras societarias complejas y opacas, en la fragmentación de movimientos en distintas transferencias de pequeño importe y en el uso encadenado de monederos digitales que dificultan el rastreo del origen de los fondos.

Frente a un problema que no deja de evolucionar, los sistemas tradicionales de prevención del blanqueo (AML) han mostrado límites cada vez más visibles. La mayoría funcionan a partir de reglas fijas y umbrales predefinidos: detectan patrones estandarizados que ya conocen, pero no lo que todavía no ha aparecido. Esa rigidez genera principalmente dos problemas. Por un lado, permite ser relativamente fácil de eludir, ya que basta con estructurar las transacciones por debajo de los umbrales de alerta para evitar la detección. Por otro lado, generan un volumen desproporcionado de falsos positivos (por encima del 90 %, según las estimaciones del sector), lo que satura a los equipos de cumplimiento y los obliga a dedicar buena parte de su tiempo a revisar casos que no entrañan ningún riesgo real, haciendo de este modo que se pierda una gran

eficiencia.

El resultado es un modelo que reacciona, pero que apenas se anticipa. La evidencia de estas limitaciones se puede observar en los resultados: pese al esfuerzo regulatorio y operativo desplegado, solo alrededor del 1 % de los beneficios del crimen organizado llega a ser confiscado en la Unión Europea (Europol, 2017). En otras palabras, la gran mayoría de los fondos provenientes de origen ilícito permanece fuera del alcance de las autoridades. Esta brecha entre los recursos implementados y los resultados obtenidos es, en el fondo, lo que justifica la búsqueda de un enfoque distinto, capaz no solo de identificar patrones conocidos, sino también de anticiparse a nuevas tipologías de blanqueo que aún no se han manifestado.

Por este motivo, dicha necesidad de pasar de la reacción a la anticipación constituye el punto de partida de este trabajo.

1.3. Motivación del proyecto

La elección de este tema no ha sido por azar, sino porque reúne dos ámbitos que considero especialmente relevantes: la tecnología aplicada a las finanzas y el trabajo real dentro del cumplimiento normativo. Gracias a haber tenido la oportunidad de cursar este Máster en Tecnologías Financieras de la Universidad Pontificia Comillas (ICAI), he adquirido una base sólida en análisis de datos, aprendizaje automático y monitorización de pagos, además de una visión clara del marco regulatorio que rige este sector. No obstante, ha sido mi experiencia profesional la que ha terminado de afianzar dicho interés.

En la actualidad trabajo en la consultora multinacional NTT DATA, dentro del área de Banking Financial Services, donde realizo tareas de diligencia debida reforzada (EDD), conocimiento del cliente (KYC) y análisis de operaciones de clientes Tier 1 para entidades del grupo BBVA en Alemania e Italia, uno de los principales grupos bancarios a escala europea. En el día a día realizo análisis a clientes y operaciones en detalle: comprobar si figuran en listas de sanciones, valorar si el origen o el destino de los fondos procede de zonas de alto riesgo o si los importes encajan con el perfil declarado. Asimismo, también reviso y reporto las alertas generadas por los sistemas internos que tiene el banco.

Ese contacto directo con la operativa me ha permitido ver dos cosas con mis propios ojos: la complejidad que tienen los esquemas que tratamos de detectar y lo limitadas que resultan, en muchas ocasiones, las herramientas con las que los combatimos.
Resulta difícil no percibir el desajuste que existe entre el esfuerzo que supone rastrear una sola operación sospechosa y lo poco que se anticipan unos sistemas que, casi siempre, reaccionan cuando el problema ya ha ocurrido y es que, tras ser conocedor del tiempo que se invierte en revisar alertas que acaban resultando falsas en su mayoría, se vuelve inevitable preguntarse si no existe una forma más inteligente y eficaz de afrontar dicho problema. Este trabajo nace precisamente de esa pregunta, y de la convicción de que el futuro de la prevención del blanqueo de capitales debería enfocarse en adelantarse, y no solo en la reacción y la corrección posterior.

1.4. Objetivos

Una vez argumentados el contexto y la motivación, es necesario concretar con claridad los objetivos de este trabajo. Para ello se establece un objetivo general que recoge su finalidad principal, junto con una serie de objetivos específicos que desglosan los pasos necesarios para alcanzarlo y que, al mismo tiempo, estructuran el desarrollo de la memoria.

1.4.1. Objetivo general

El objetivo general de este Trabajo Fin de Máster es proponer y argumentar, a nivel conceptual, un ecosistema adversarial de inteligencia artificial orientado a la detección y la anticipación del blanqueo de capitales. Dicho ecosistema se articula en torno a dos agentes que compiten entre sí, uno encargado de generar nuevas tipologías y métodos de blanqueo y otro destinado a detectarlas, ambos integrados en un entorno de simulación que actúa como marco de validación. La propuesta se complementa con una prueba de concepto sencilla que ilustra, sin ánimo de exhaustividad, la viabilidad del enfoque y la lógica que lo sustenta.

1.4.2. Objetivos específicos

A partir de ese objetivo general, el trabajo se concreta en cuatro objetivos específicos:

- Diseñar el agente atacante basado en aprendizaje por refuerzo y analizar cómo podría aprender a generar nuevas formas de blanqueo mediante un esquema de prueba y recompensa, ilustrándolo con una pequeña demostración que introduzca patrones conocidos (como el pitufeo o los flujos en círculo) en un grafo de transacciones sintético.
- Diseñar el agente defensor basado en redes neuronales sobre grafos (GNNs) y Transformers, justificando por qué la combinación de ambas tecnologías encaja con la naturaleza de red del blanqueo, e implementar una versión básica capaz de distinguir, sobre ese mismo grafo, las operaciones sospechosas de las legítimas.
- Analizar y justificar las ventajas de la propuesta frente a los sistemas actuales, valorando en qué medida un enfoque de estas características podría reducir los falsos positivos, anticiparse mejor a las amenazas y emplear de forma más eficiente los recursos de compliance.
- Plantear la evolución de la propuesta hacia el paradigma del Compliance-as-Code y señalar las líneas de trabajo futuras, reconociendo con honestidad el alcance de esta idea y los pasos que serían necesarios para convertir la prueba de concepto en una implementación más completa y realista.

En conjunto, estos objetivos combinan una vertiente conceptual, que constituye el núcleo del trabajo, con una parte práctica de carácter demostrativo, de forma que la propuesta esté bien fundamentada y pueda comprobarse en un caso sencillo.

1.5. Alcance y limitaciones

Conviene aclarar el alcance de este trabajo para ajustar las expectativas a lo que realmente se presenta. Este TFM es, ante todo, una propuesta conceptual: su principal aportación es diseñar y argumentar un ecosistema adversarial de IA basado en el conocimiento existente, y no construir un sistema operativo a escala real. La prueba de concepto asociada se desarrolla con datos sintéticos y modelos básicos, mediante scripts simples tanto del agente atacante como del defensor, con el único objetivo de mostrar la viabilidad lógica de la idea.

Esta delimitación implica varias restricciones. En primer lugar, no se utilizan datos reales de transacciones por su carácter confidencial y porque el alcance del trabajo no lo requiere. En segundo lugar, los modelos empleados son simplificados y no están optimizados, por lo que sus resultados deben interpretarse únicamente como una demostración. Por último, este trabajo no desarrolla una implementación real ni la compara con las herramientas utilizadas actualmente por las entidades financieras, ya que ello exigiría recursos, infraestructura, datos y capacidad computacional que no son propios de un estudiante de Máster.

No obstante, estas limitaciones no reducen el valor de la propuesta, sino que ayudan a situarla en su contexto real: una propuesta conceptual que explora una posible línea de investigación y desarrollo, y que puede servir como base para trabajos posteriores más avanzados o implementaciones a una mayor escala.

1.6. Metodología y plan de trabajo

Dado que este proyecto busca diseñar una solución tecnológica más que contrastar una hipótesis, la metodología se estructura siguiendo una lógica de resolución de problemas: en primer lugar, se identifica con precisión el problema, después se propone una solución, a continuación, se diseña y, por último, se valora. Sobre esa base se combinan tres formas de investigación complementarias.

La primera es la revisión bibliográfica, basada en el estudio de artículos académicos, publicaciones técnicas, fuentes documentadas y normativa, que sirve como base teórica del trabajo y garantiza que las decisiones de diseño se apoyen en información ya contrastada. La segunda es el análisis del estado del arte y la comparación entre tecnologías, que permite examinar las distintas opciones disponibles (tipos de redes sobre grafos, algoritmos de aprendizaje por refuerzo o arquitecturas basadas en Transformers) y justificar cada elección. La tercera incorpora un apoyo más reciente: el uso de inteligencia artificial generativa como herramienta para explorar ideas, recopilar literatura científica y detectar posibles conexiones entre investigaciones que podrían no ser evidentes a primera vista.

Sobre este último punto conviene ser explícito y es que, la inteligencia artificial no se ha empleado en ningún caso para redactar el contenido de la memoria, sino únicamente como apoyo para orientar la búsqueda y la exploración de fuentes, así como para la organización de la estructura y la orientación general del trabajo. La redacción, el análisis y las conclusiones son enteramente propios, de manera que la autoría y la integridad del trabajo quedan plenamente garantizadas.

En cuanto al plan de trabajo, el proyecto se ha organizado en cinco fases consecutivas, aunque con cierta interrelación entre ellas: una primera de revisión bibliográfica, seguida de un análisis del estado del arte, una tercera centrada en el diseño del ecosistema (el sandbox, los dos agentes y su interacción), una cuarta centrada en las métricas y las primeras pruebas, y una quinta de redacción final. Esta planificación se adapta a la duración habitual de un Trabajo Fin de Máster y permite volver a fases anteriores cuando sea necesario, ya que el proceso de diseño no suele seguir un orden totalmente lineal.

1.7. Estructura de la memoria

La memoria se divide en ocho capítulos que siguen una estructura lógica alineada con el desarrollo del trabajo. Tras esta introducción, el segundo capítulo presenta el marco teórico y el estado de la cuestión, desde el contexto regulatorio hasta la evolución de las soluciones tecnológicas, con el objetivo de identificar la brecha que la propuesta pretende cubrir.

El tercero presenta NextGen AML y sus elementos diferenciales, y el cuarto detalla el diseño del ecosistema y la articulación entre el entorno de simulación y los dos agentes. El quinto capítulo desarrolla una prueba de concepto acotada, cuyos resultados se analizan en el sexto. El séptimo proyecta la evolución de la idea hacia el Compliance-as-Code, y el octavo y último capítulo recoge las conclusiones y cierra el hilo argumental del trabajo.

2. Marco teórico y estado de la cuestión

2.1. Fundamentos del blanqueo de capitales y marco regulatorio (AML, KYC/EDD, GAFI, PSD2)

Según la Oficina de las Naciones Unidas contra la Droga y el Delito (UNODC, s.f.), el blanqueo de capitales puede definirse como el proceso mediante el cual los fondos procedentes de actividades ilícitas se introducen en el sistema financiero legal, ocultando su origen y dotándolos de una apariencia legítima. No se trata de un hecho aislado, sino de un proceso que los estudios sobre este fenómeno suelen estructurar en tres fases: la primera es la colocación, en la que el dinero de origen delictivo se introduce por primera vez en el circuito financiero, normalmente en forma de efectivo. La segunda es el encubrimiento, que busca ocultar su origen mediante múltiples transferencias, uso de sociedades interpuestas y movimientos entre distintas jurisdicciones dificultando de este modo su rastreo. La tercera es la integración, en la que los fondos ya "limpios" se reintroducen en la economía a través de inversiones, negocios o adquisiciones. Entender esta estructura es importante ya que cada fase ofrece puntos de control distintos y condiciona la forma en que los sistemas de prevención intentan detectar e intervenir estas operaciones.

Frente a este fenómeno surge lo que se conoce como prevención del blanqueo de capitales (Anti-Money Laundering o AML), un marco que engloba el conjunto de normas, obligaciones y procedimientos destinados a evitar que el sistema financiero sea utilizado con tales fines ilícitos. A nivel internacional, el principal referente de dicho marco es el Grupo de Acción Financiera Internacional (GAFI, o FATF por sus siglas en inglés), creado en 1989 en el seno del entonces G7. El GAFI no dicta normas de aplicación directa, sino que establece estándares (las conocidas como Cuarenta Recomendaciones) que los países adoptan y trasladan posteriormente a su propia legislación (GAFI, 2012). Su principal aportación ha sido consolidar el enfoque basado en riesgo, según el cual las entidades deben concentrar sus esfuerzos allí donde la probabilidad de blanqueo es mayor, en lugar de aplicar controles uniformes a todos los clientes y operaciones por igual.

Sobre esta base se articulan las obligaciones de diligencia debida, que constituyen el núcleo operativo de la prevención. La más relevante es el conocimiento del cliente (Know Your Customer o KYC), que obliga a las entidades a identificar y verificar la identidad de sus clientes, comprender su actividad económica y analizar el origen de los fondos, tanto al inicio de la relación como durante su desarrollo. En función del nivel de riesgo, se aplican distintos grados de control: cuando este es bajo o medio se realiza una diligencia debida ordinaria (Customer Due Diligence o CDD), mientras que, en situaciones de mayor riesgo como clientes con responsabilidad pública, jurisdicciones de alto riesgo u operaciones inusuales se exige una diligencia debida reforzada (Enhanced Due Diligence o EDD), más exhaustiva y documentada. Estas obligaciones se efectúan con la monitorización continua de las operaciones y con el deber de reportar aquellas consideradas sospechosas a las autoridades policiales competentes, lo que sitúa a las entidades financieras como la primera línea de defensa del sistema.

En el ámbito de la Unión Europea, estas exigencias se han ido desarrollando a través de distintas directivas durante las últimas décadas. Aun así, el modelo tradicional, en el que cada Estado miembro las transponía a su manera, ha dado paso recientemente a un marco mucho más armonizado: el paquete normativo de 2024, que introduce un código normativo único y directamente aplicable en los Veintisiete (Reglamento (UE) 2024/1624) y crea una autoridad supervisora común, la AMLA, con sede en Fráncfort (Alemania) (Reglamento (UE) 2024/1620). A este conjunto se suma, de forma complementaria, la segunda Directiva de

Servicios de Pago (PSD2), que, aunque su finalidad principal es regular los servicios de pago y abrir el sector a nuevos actores (como Bizum, Apple Pay o Google Pay) mediante el Open Banking, guarda una correlación estrecha con la prevención del blanqueo, ya que, al multiplicar los intervinientes y acelerar las operaciones, refuerza al mismo tiempo las exigencias de seguridad, autenticación y trazabilidad sobre las que se apoya el control (Directiva (UE) 2015/2366). En conjunto, AML, KYC/EDD, GAFI y PSD2 constituyen el marco regulatorio que establece las reglas del entorno en el que se desarrollan las soluciones tecnológicas posteriores.

2.2. Evolución de los sistemas de prevención del blanqueo

Una vez establecido el marco conceptual y regulatorio, es conveniente conocer cómo estas exigencias se han materializado en herramientas concretas, ya que los métodos de detección del blanqueo han evolucionado de forma significativa con el tiempo. Los primeros sistemas eran principalmente manuales y dependían de la experiencia de los analistas; sin embargo, con el aumento del volumen y la velocidad de las operaciones pronto resultaron insuficientes y dieron paso a soluciones automatizadas cada vez más complejas. En ese sentido, dicha evolución no ha sido lineal, sino que refleja un esfuerzo continuo por equilibrar dos objetivos que a menudo entran en tensión: detectar la mayor cantidad posible de actividad sospechosa y, al mismo tiempo, reducir el número de falsos positivos. Los siguientes apartados recorren este desarrollo, desde los sistemas basados en reglas hasta las técnicas de aprendizaje automático, como introducción a la propuesta de este trabajo.

2.2.1. Sistemas basados en reglas y umbrales

El primer paso en la evolución de los sistemas de prevención del blanqueo lo constituyen los modelos basados en reglas y umbrales, considerados la primera generación de soluciones AML automatizadas. Su funcionamiento se basa en una lógica muy sencilla: cuando una operación cumple determinadas condiciones previamente ya definidas, el sistema genera automáticamente una alerta. Algunos ejemplos habituales son los ingresos en efectivo por encima de un determinado importe (como el umbral de 10.000 euros), las transferencias dirigidas a jurisdicciones de alto riesgo o los movimientos que no encajan con el perfil declarado del cliente.

La principal ventaja de este enfoque reside en su simplicidad y su transparencia. Al tratarse de reglas explícitas y objetivas, resulta fácil comprender el motivo por el que se ha generado cada alerta, lo que simplifica las tareas de supervisión y auditoría. Sin embargo, esta misma característica constituye también su principal limitación. Por un lado, estos sistemas solo detectan aquello que se ha definido previamente, de modo que tienen dificultades para identificar nuevas estrategias de blanqueo. Asimismo, resultan relativamente fáciles de eludir, ya que basta con estructurar las operaciones por debajo de los umbrales establecidos, como la fragmentación de transferencias en pequeños importes (la conocida técnica del pitufeo) para así, pasar desapercibido. A ello se suma un tercer inconveniente: la generación de un elevado volumen de falsos positivos, puesto que muchas operaciones legítimas acaban cumpliendo alguna de las condiciones sin representar ningún riesgo real.

En consecuencia, aunque los sistemas basados en reglas siguen estando presentes en buena parte de las entidades por su fiabilidad y su bajo coste, pronto se hizo evidente la necesidad de complementarlos con enfoques más avanzados y capaces de adaptarse a nuevas formas de blanqueo.

2.2.2. Scoring y aprendizaje automático supervisado

Para superar dichas limitaciones, surge una segunda generación de sistemas apoyada en el scoring estadístico y, posteriormente, en el aprendizaje automático supervisado. El enfoque cambia de forma significativa: en lugar de aplicar reglas fijas, el sistema aprende a partir de casos reales previamente etiquetados. En otras palabras, aprende de operaciones que en el pasado se clasificaron como sospechosas frente a otras consideradas legítimas. A partir de esos ejemplos, el modelo identifica patrones y combinaciones de variables asociadas al riesgo y los utiliza para evaluar nuevas operaciones.

En este enfoque cobran protagonismo las llamadas variables de riesgo, es decir, los distintos atributos de una operación o de un cliente que el modelo tiene en cuenta (el importe, la frecuencia, el país de origen o destino, la antigüedad de la cuenta, etc.). Con ellas se entrenan modelos de clasificación que, en lugar de limitarse a una respuesta binaria, asignan a cada operación una puntuación de riesgo. Esto supone una mejora relevante respecto a los sistemas basados en reglas, ya que permite captar combinaciones mucho más complejas y priorizar las alertas en función de su probabilidad, facilitando el trabajo de los analistas.

No obstante, el aprendizaje supervisado también presenta limitaciones importantes. La principal es su dependencia de datos etiquetados. La razón es que los datos en el ámbito del blanqueo son muy escasos, en gran medida confidenciales y altamente desbalanceados, ya que los casos confirmados de blanqueo representan una proporción mínima frente al volumen total de operaciones legítimas. Esto dificulta con creces el entrenamiento y puede sesgar los modelos hacia los patrones más frecuentes. A esto hay que añadir una segunda limitación que también es relevante: al aprender únicamente de lo que ya se conoce, estos sistemas siguen siendo incapaces de detectar tipologías realmente nuevas, es decir, aquellas que no se asemejan a ningún caso previo. Dicho de otro modo, mejoran la detección de lo conocido, pero no resuelven el problema de la anticipación.

2.2.3. Aprendizaje no supervisado y detección de anomalías

En paralelo, y como respuesta a la dependencia de los datos etiquetados, surge un tercer enfoque basado en el aprendizaje no supervisado. A diferencia del anterior, este no necesita ejemplos previamente clasificados, sino que trabaja directamente sobre los datos en bruto para identificar su estructura. Sus dos aplicaciones más habituales son el clustering, que agrupa operaciones o clientes con un comportamiento similar, y la detección de anomalías, que parte de lo que se considera un comportamiento normal y señala aquellas observaciones que se desvían de él de forma significativa.

La gran ventaja de este planteamiento es que permite revelar patrones desconocidos, es decir, comportamientos que no se ajustan a reglas predefinidas ni a ningún caso etiquetado previamente. Por este motivo, resulta especialmente útil frente a nuevos métodos de blanqueo, que son precisamente los que escapan a los enfoques anteriores. Sin embargo, también presenta limitaciones importantes. La primera es su dificultad de interpretación: cuando el sistema marca una operación como anómala, no siempre resulta evidente por qué lo ha hecho, lo que complica la labor del analista y reduce la confianza en el resultado. Del mismo modo, destaca su tendencia a generar falsos positivos, y es que una anomalía no equivale necesariamente a un delito puesto que muchos comportamientos atípicos tienen una explicación perfectamente legítima y, aun así, son marcados por el sistema.

Con el aprendizaje no supervisado se cierra lo que podría considerarse la evolución "clásica" de la prevención del blanqueo. Cada generación ha aportado mejoras evidentes, pero ninguna ha conseguido resolver completamente el doble reto de anticiparse a lo desconocido sin disparar el número de falsas alertas. Es en este punto donde entran en juego las tecnologías de inteligencia artificial más avanzadas sobre las que se apoya la propuesta de este trabajo.

2.3. Tecnologías de inteligencia artificial relevantes para la detección

Las limitaciones de los enfoques anteriores explican el creciente interés por nuevas técnicas de inteligencia artificial que, en los últimos años, han cambiado muchos campos y que resultan especialmente adecuadas para la detección del blanqueo. No se trata de sustituir todo lo anterior, sino de dar un paso más y abordar el problema desde una perspectiva distinta, capaz de capturar tanto la estructura como la dimensión temporal de las operaciones y, sobre todo, de anticiparse a las amenazas que aún no se han manifestado.

Los siguientes apartados presentan las cuatro tecnologías en las que se basa la propuesta de este trabajo: las redes neuronales sobre grafos (GNNs), los Transformers y su mecanismo de atención, el aprendizaje por refuerzo y la inteligencia artificial adversarial. Cada una de ellas desempeña un papel concreto dentro del ecosistema que se planteará más adelante, ya sea en el agente defensor, en el agente atacante o en la dinámica que los enfrenta.

2.3.1. Redes neuronales sobre grafos (GNNs)

La primera de estas tecnologías, y la que constituye el núcleo del agente defensor, es la red neuronal sobre grafos (Graph Neural Networks o GNNs), un tipo de modelo diseñado para procesar datos que representan relaciones complejas en lugar de información aislada. Para entender su utilidad conviene partir de la forma en que representan la información, y es que, en lugar de organizar los datos en tablas, los estructuran como un grafo, es decir, como un conjunto de nodos conectados entre sí mediante aristas. Trasladado al ámbito financiero, esta representación resulta muy intuitiva: las cuentas o las entidades se representan como nodos y las transacciones que las relacionan, como las aristas que los unen.

Esta manera de estructurar los datos encaja bastante bien con el blanqueo de capitales, que es,

ante todo, un fenómeno de red. Las tipologías más habituales (cadenas de transferencias, abanicos en los que un origen reparte fondos entre múltiples destinos o flujos en círculo que regresan al punto de partida) no son otra cosa más que estructuras dentro de ese grafo. Mientras que un modelo tabular analiza cada operación de forma independiente y pierde por completo esa dimensión relacional, una GNN es capaz de captar dichos patrones estructurales, que suelen ser clave para identificar el comportamiento sospechoso.

El mecanismo que lo hace posible es la propagación de información entre nodos vecinos: cada nodo actualiza su representación combinando la información de los nodos con los que está conectado, de modo que la red aprende no solo a partir de las características de una cuenta o un cliente, sino también de las de su entorno. Sobre esta idea se han desarrollado distintas variantes, entre las que destacan las Graph Convolutional Networks (GCN) (Kipf y Welling, 2017), GraphSAGE, que permite generalizar a nodos no vistos durante el entrenamiento (Hamilton et al., 2017), y las Graph Attention Networks (GAT), que asignan distintos pesos a la influencia de cada vecino (Veličković et al., 2018).

La implementación de estas técnicas a la detección del blanqueo ha dado lugar a trabajos de referencia, como el de Weber et al. (2019), que emplearon redes convolucionales sobre grafos utilizando el conjunto de datos Elliptic, uno de los más conocidos para el análisis de transacciones de Bitcoin con fines forenses. Por todo ello, las GNNs se consolidan como la pieza central del agente defensor, encargado de modelar el sistema financiero como un grafo y de identificar sobre él las operaciones sospechosas.

2.3.2. Transformers y el mecanismo de atención

Si las GNNs aportan la visión estructural, los Transformers añaden una dimensión igualmente importante: la temporal. Esta arquitectura, presentada por Vaswani et al. (2017) en el ámbito del procesamiento del lenguaje natural, se diseñó para superar las limitaciones de los modelos secuenciales tradicionales en el tratamiento de dependencias a largo plazo y, desde entonces, se ha consolidado como una de las tecnologías más influyentes de la inteligencia artificial reciente.

Su elemento clave es el llamado mecanismo de atención, que permite al modelo asignar la importancia de cada elemento de una secuencia en relación con los demás, con independencia de la distancia que los separe. Dicho de otro modo, el modelo aprende a fijarse en lo que realmente importa dentro de una serie de datos, en lugar de tratarlos a todos por igual. Aunque esta idea nació en el ámbito del lenguaje, resulta igualmente útil cuando se traslada a secuencias de transacciones.

Y es que el comportamiento de un cliente no se entiende únicamente por la estructura de sus relaciones, sino también por el orden y el momento en que se producen sus operaciones. Una secuencia aparentemente normal puede resultar sospechosa precisamente por su dinámica temporal: por la rapidez con que se ejecutan, por su repetición en intervalos concretos o por coincidir con determinados patrones en el tiempo. Los Transformers permiten capturar esa dimensión secuencial que las GNNs, centradas en la estructura, no recogen por sí solas. Por este motivo, dentro de la propuesta actúan como complemento del agente defensor, combinando la visión estructural del grafo con la perspectiva temporal de las operaciones para lograr una detección más completa y efectiva.

2.3.3. Aprendizaje por refuerzo

Mientras que las tecnologías anteriores se centran principalmente en la detección, el aprendizaje por refuerzo (Reinforcement Learning o RL) constituye la base del agente atacante y aporta a la propuesta su capacidad anticipatoria. A diferencia del aprendizaje supervisado, que aprende de ejemplos ya etiquetados, el aprendizaje por refuerzo se basa en un esquema de prueba y error: un agente interactúa con un entorno, ejecuta acciones y recibe a cambio recompensas o penalizaciones que le indican si va por buen camino, de manera que aprende qué decisiones le acercan a su propósito en función de los resultados obtenidos (Sutton y Barto, 2018).

Su funcionamiento se apoya en varios conceptos fundamentales. El estado describe la situación del entorno en un momento determinado; la acción corresponde a cada una de las decisiones que puede tomar el agente; la recompensa es la señal, positiva o negativa, que recibe tras actuar; y la política es la estrategia que va aprendiendo para elegir, en cada ocasión, la acción que maximiza la recompensa a largo plazo. Al repetir este proceso una y otra vez, el agente acaba descubriendo por sí mismo comportamientos eficaces que nadie le ha enseñado de forma explícita. Uno de los hitos que popularizó este enfoque fue el trabajo de Mnih et al. (2015), en el que un único sistema aprendió a jugar a numerosos videojuegos utilizando únicamente la imagen de la pantalla y la puntuación obtenida como guía.

Aplicado al contexto de este trabajo, dicha lógica resulta especialmente atractiva. El agente atacante puede entenderse como un sistema de aprendizaje por refuerzo cuyo objetivo es aprender a generar nuevos métodos de blanqueo: cada intento de mover dinero ilícito sin ser detectado constituye una acción, y el hecho de lograr evadir o no al defensor funciona como señal de recompensa. De este modo, el atacante no se limita a reproducir esquemas conocidos, sino que explora activamente formas novedosas de eludir los controles, anticipándose así a amenazas que todavía no se han manifestado en la realidad. Es precisamente esta capacidad de imaginar lo que aún no existe la que diferencia la propuesta de los enfoques que solo reaccionan ante problemas ya existentes.

2.3.4. Inteligencia artificial adversarial

La última de estas tecnologías, y la que da sentido al conjunto, es la inteligencia artificial adversarial. Su idea central es simple pero potente: hacer competir a dos agentes entre sí, de forma que la mejora de uno obliga al otro a evolucionar y viceversa, generando así un proceso de perfeccionamiento mutuo que ninguno alcanzaría por separado.

El ejemplo más representativo de este enfoque son las redes generativas adversariales (Generative Adversarial Networks o GANs), propuestas por Goodfellow et al. (2014). En este marco conviven dos redes con objetivos opuestos: un generador, que trata de crear datos falsos lo bastante realistas como para resultar creíbles, y un discriminador, que intenta distinguir esos datos artificiales de los reales. A medida que el generador mejora su capacidad de engaño, el

discriminador se ve obligado a afinar su capacidad de detección, dando lugar a una dinámica de mejora mutua.

Este esquema sirve como base conceptual de NextGen AML. El agente atacante desempeña un papel equivalente al del generador, buscando crear nuevas tipologías de blanqueo que pasen desapercibidas, mientras que el agente defensor actúa como el discriminador, encargado de detectarlas. La interacción entre ambos da lugar a una dinámica de coevolución en la que atacante y defensor se retroalimentan y se obligan mutuamente a perfeccionarse: cada nueva estrategia de evasión obliga al defensor a adaptarse, y cada mejora en la detección empuja al atacante a idear estrategias más sofisticadas. Es esta lógica adversarial, trasladada al sector de la prevención del blanqueo de capitales, la que hace que la propuesta sea dinámica y la distingue de los sistemas estáticos descritos hasta ahora.

2.3.5. Entornos sintéticos y Compliance-as-Code

Una vez presentada la lógica adversarial que sustenta la propuesta, conviene completar este bloque tecnológico con dos elementos que, aunque no actúan directamente como modelos de detección, resultan fundamentales para hacer viable todo el planteamiento: los entornos sintéticos y el paradigma del Compliance as Code.

Los entornos sintéticos consisten en la generación de transacciones y escenarios ficticios que imitan el comportamiento de un sistema financiero real, pero sin contener información de clientes reales. Según Assefa et al. (2020), la generación de datos sintéticos ha ganado relevancia en el sector financiero como respuesta a las dificultades existentes para compartir información real, debido a las restricciones regulatorias y de privacidad. Su utilidad es doble. Por un lado, permiten entrenar y poner a prueba los modelos sin necesidad de usar datos reales, que en el ámbito del blanqueo son escasos y, sobre todo, altamente confidenciales. Además, ofrecen una solución al problema de la falta de casos etiquetados, ya que en un entorno controlado y artificial es posible diseñar a voluntad operaciones legítimas y fraudulentas con las que entrenar al sistema. Esta idea constituye uno de los pilares de la propuesta y se materializará más adelante en el Financial Sandbox que se describe en el capítulo 4.

La segunda pieza es el Compliance-as-Code, un paradigma estrechamente ligado al concepto de Rules as Code impulsado por la OCDE (Organización para la Cooperación y el Desarrollo Económicos) (Mohun y Roberts, 2020), que traduce las normas y los procesos de cumplimiento en código auditable, automático y actualizable. Esto resulta muy útil ya que en lugar de depender de procedimientos manuales o de interpretaciones dispersas, las obligaciones regulatorias se expresan directamente como reglas programables que pueden ejecutarse, verificarse y modificarse con rapidez si cambia alguna regulación. Más que una tecnología concreta, representa una oportunidad de evolución para la propuesta, que será desarrollada con mayor detalle en el capítulo 7.

2.4. La necesidad tecnológica identificada

A lo largo de este capítulo se ha analizado la evolución de los sistemas de prevención del blanqueo de capitales y se han presentado las principales tecnologías que actualmente permiten abordar este problema. Llegados a este punto, conviene detenerse y observar el panorama en su conjunto, ya que desde esa mirada global surge la brecha que justifica este proyecto.

El recorrido deja una conclusión clara: existen numerosas tecnologías con un gran potencial, pero dispersas. Los sistemas basados en reglas aportan transparencia; el aprendizaje supervisado permite identificar patrones complejos; el no supervisado descubre lo desconocido; las redes neuronales sobre grafos (GNNs) modelan la dimensión estructural; los Transformers incorporan la temporal; el aprendizaje por refuerzo aporta capacidad de anticipación; la inteligencia artificial adversarial ofrece un marco para la mejora mutua; y los entornos sintéticos y el Compliance-as-Code resuelven los problemas relacionados con los datos y con la implementación de las regulaciones vigentes. Cada una de estas tecnologías ha demostrado su valor por separado.

Sin embargo, dichas tecnologías suelen utilizarse como herramientas independientes que operan de forma separada. En la práctica, las entidades combinan estas herramientas como sistemas aislados que funcionan en paralelo, sin que ninguna empuje a las demás a mejorar. El resultado sigue siendo un conjunto de soluciones que reaccionan ante lo que ya ha ocurrido y que, en el mejor de los casos, detectan mejor lo conocido, pero no se anticipan a lo que está por venir.

Aquí es donde se localiza la brecha. No existe una integración efectiva de todas estas tecnologías dentro de un único ecosistema capaz de aprovechar las fortalezas de cada una de ellas. Del mismo modo, nunca se ha dado el salto de la detección posterior a la anticipación a través de una dinámica adversarial y coevolutiva, en la que un agente encargado de generar nuevas amenazas obligue de forma continua al encargado de detectarlas a mejorar, y viceversa. Es decir, falta un planteamiento que no solo reúna las tecnologías, sino que las haga interactuar entre sí para competir y evolucionar de forma conjunta.

Esa es, exactamente, la necesidad que NextGen AML pretende cubrir. La propuesta que se presenta en el siguiente capítulo surge precisamente de esta carencia y plantea un entorno capaz de integrar las tecnologías descritas en un marco adversarial capaz de pasar de reaccionar a anticiparse.

3. NextGen AML: la propuesta

3.1. Planteamiento general: un ecosistema adversarial de IA

NextGen AML surge como respuesta directa a la brecha descrita y se plantea, en esencia, como un ecosistema adversarial de inteligencia artificial. La idea central es sencilla: en lugar de utilizar las tecnologías presentadas en el capítulo anterior como capas independientes que funcionan por separado, se propone integrarlas en un único sistema en el que dos agentes inteligentes compiten entre sí dentro de un mismo entorno. De esta interacción nace el principal valor de la propuesta, ya que obliga al sistema a mejorar de forma constante y a anticiparse, en vez de limitarse a reaccionar.

El ecosistema se estructura en tres componentes, que se presentan a continuación, dejando su diseño para el capítulo 4. El primero es el agente atacante, construido sobre el aprendizaje por refuerzo. Su función es generar nuevas tipologías de blanqueo capaces de eludir los controles, aprendiendo por prueba y error: cada vez que consigue mover dinero ilícito sin ser detectado, recibe una señal positiva que refuerza dicha estrategia. De este modo, el atacante no se limita a reproducir esquemas ya conocidos, sino que explora de forma activa formas novedosas de esquivar los mecanismos de detección, actuando como una amenaza en constante evolución.

El segundo componente es el agente defensor, el cual combina las redes neuronales sobre grafos (GNNs) y los Transformers. Su misión es la contraria: detectar las operaciones sospechosas que el atacante introduce en el sistema. Para ello aprovecha las dos dimensiones que se describieron en el capítulo anterior, y es que las GNNs permiten captar la estructura de las relaciones entre cuentas y transacciones, mientras que los Transformers incorporan la dimensión temporal, es decir, el orden y la secuencia de las operaciones. Esta combinación le da una visión más completa que la de los enfoques tradicionales, que normalmente solo miran una parte del problema.

El tercer componente es el entorno de simulación, el cual consiste en lo denominado anteriormente como Financial Sandbox, en el que ambos agentes coexisten e interactúan. Se trata de un entorno sintético que reproduce el funcionamiento de un sistema financiero como si fuese real mediante transacciones y operaciones artificiales, sin recurrir en ningún momento a datos reales. Este espacio cumple una función clave: ofrece un terreno seguro y controlado en el que el atacante puede ensayar sus estrategias y el defensor puede aprender a reconocerlas, resolviendo a su vez los problemas de privacidad y de escasez de datos etiquetados mencionados anteriormente.
No obstante, el valor esencial de NextGen AML, no reside en ninguno de estos componentes, sino en la relación que tienen entre ellos (una dinámica de coevolución continua). El atacante y el defensor se enfrentan de forma repetida dentro del sandbox, de manera que cada uno empuja al otro a mejorar. Cuando el atacante descubre una nueva forma de blanqueo que logra pasar desapercibido, el defensor se ve obligado a aprender a detectarla; y en cuanto el defensor mejora y cierra esa vía, el atacante debe idear nuevas estrategias todavía más complejas. Este ciclo, al repetirse una y otra vez, hace que el sistema en su conjunto evolucione de forma autónoma y se mantenga actualizado frente a amenazas que ni siquiera han aparecido aún.

Es precisamente en este punto donde la propuesta da el salto a la anticipación de distintos métodos. Mientras que los sistemas actuales esperan a que una tipología se manifieste para poder reconocerla, NextGen AML la genera internamente antes de que ocurra, preparándose

para combatirla y detectarla con antelación. Conviene precisar, a pesar de todo, que esta coevolución continua es el objetivo conceptual de la propuesta. La prueba de concepto que se presenta más adelante solo reproduce una versión simplificada de esta dinámica, pensada para ilustrar la idea, no para replicar el sistema completo. Aun así, esta lógica adversarial es la que define NextGen AML y la que, como se verá a continuación, estructura todos sus elementos diferenciales.

3.2. Elementos diferenciales frente a los enfoques actuales

Una vez presentado todo el planteamiento, conviene detenerse en los elementos que diferencian a NextGen AML tanto de los sistemas tradicionales de prevención como de los enfoques de inteligencia artificial, que, como se ha visto, suelen aplicarse de forma aislada. En conjunto, estos elementos son los que aportan valor a la propuesta.

El primero y más importante es su carácter dinámico y coevolutivo. Frente a unos sistemas tradicionales completamente estáticos, que solo se actualizan si una persona modifica sus reglas o reentrena sus modelos, NextGen AML evoluciona por sí mismo gracias a la competencia continua entre atacante y defensor. Esto evita que el sistema quede obsoleto con el tiempo y le permite adaptarse de manera constante.

Directamente relacionado con lo anterior se encuentra su capacidad anticipatoria, que constituye quizá su rasgo más distintivo. Mientras que los enfoques actuales se limitan a detectar patrones ya conocidos, la propuesta es capaz de generar internamente amenazas que todavía no se han manifestado, de modo que se prepara para combatirlas antes de que lleguen a producirse en la realidad. Dicho de otra manera, el sistema deja de ir por detrás del delincuente para situarse, por primera vez, un paso por delante.

Un tercer elemento es la integración de la estructura y la temporalidad en un mismo análisis. La combinación de redes neuronales sobre grafos (GNNs) y Transformers permite al defensor analizar de forma continua cómo se relacionan las cuentas entre sí y en qué orden y momento se producen las operaciones. Esta doble perspectiva ofrece un enfoque más completo que el de los sistemas convencionales, que normalmente suelen centrarse en una única dimensión del problema.

A esto se le suma el uso de entornos sintéticos, que aporta ventajas tanto en lo práctico como en lo ético. Al entrenar y validar los modelos con datos artificiales, la propuesta evita los problemas de privacidad asociados a la información real y permite además superar la escasez de datos etiquetados, generando a voluntad propia las operaciones necesarias para enseñar al sistema. De este modo, se elimina una de las principales limitaciones de los enfoques tradicionales.

Dichas características se traducen, además, en dos beneficios relacionados con el cumplimiento. El primero es la reducción potencial de los falsos positivos, ya que un defensor más preciso y mejor entrenado debería ser capaz de reconocer con mayor acierto las operaciones realmente sospechosas respecto a las legítimas, aliviando la enorme carga de trabajo que hoy soportan los equipos de compliance. El segundo es, en consecuencia, una mayor eficiencia ya que permite concentrar los recursos en los casos que de verdad lo requieren.

Por último, la propuesta se alinea con el Compliance-as-Code, que interpreta las normativas y los controles en código auditable y actualizable, siendo el proyecto completamente adaptable. Esta idea, que se desarrollará en el capítulo 7, permite orientar la propuesta hacia un sistema de cumplimiento más automatizado y que cada vez está adquiriendo mayor relevancia dentro del ámbito financiero.

En conjunto, lo que diferencia a NextGen AML no es ninguna de estas características por sí sola, sino su combinación dentro de un mismo ecosistema. Es esa integración lo que da sentido a la propuesta y explica por qué este enfoque podría aportar algo realmente diferente en un ámbito tan exigente como es la prevención del blanqueo de capitales.

3.3. Alineación con los Objetivos de Desarrollo Sostenible

Más allá de describir la propuesta de manera técnica, conviene situar el proyecto dentro de un marco más amplio y comprobar en qué medida contribuye a los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030 de las Naciones Unidas (Organización de las Naciones Unidas, 2015). Los ODS constituyen un conjunto de 17 objetivos y 169 metas aprobados en 2015 para orientar los esfuerzos internacionales hacia un desarrollo más sostenible desde el punto de vista económico, social y ambiental (Sachs, 2012). Su importancia radica en que constituyen un marco de referencia común para gobiernos, instituciones y organizaciones de todo el mundo, lo que permite valorar el impacto de iniciativas como esta más allá de su dimensión tecnológica o económica, y contribuir a la orientación de las políticas hacia un desarrollo más justo y sostenible.

El vínculo más evidente se establece con el ODS 16, dedicado a la paz, la justicia y las instituciones sólidas. Dentro de este ODS, la meta 16.4 es la que persigue de forma explícita reducir de manera significativa los flujos financieros ilícitos, que es precisamente el objeto de este trabajo. Una herramienta capaz de detectar y anticipar mejor el blanqueo de capitales contribuye plenamente a este objetivo, ya que dificulta la circulación del dinero de origen delictivo y, con ello, debilita las organizaciones y las técnicas que utilizan los criminales.

Asimismo, la propuesta se alinea con el ODS 9, el cual está centrado en la industria, la innovación y la infraestructura. NextGen AML se incluye dentro de las corrientes RegTech y SupTech, basadas en la implementación de tecnologías avanzadas para mejorar el cumplimiento normativo y la supervisión financiera, aportando innovación a un ámbito que tradicionalmente ha evolucionado de forma más lenta.

De manera más indirecta, la propuesta también guarda relación con otros tres objetivos. El ODS 8, sobre el trabajo decente y el crecimiento económico, contribuye a la existencia de un sistema financiero más seguro y transparente, lo que favorece a la creación de un entorno económico más sano. El ODS 10, busca reducir las desigualdades entre los países, ya que lucha contra los flujos financieros ilícitos, que con frecuencia privan a los Estados de recursos necesarios para su desarrollo. Y el ODS 17, sobre las alianzas para lograr los objetivos, refleja la naturaleza colaborativa de la prevención del blanqueo, una actividad que requiere la cooperación entre entidades financieras, reguladores y organismos supervisores.

En cualquier caso, conviene ser realista sobre el alcance de esta contribución. NextGen AML es, sobre todo, una propuesta conceptual y, como tal, su aportación a los ODS es sobre todo

indirecta. Aun así, resulta relevante destacar que se trata de un trabajo orientado, en última instancia, a un fin socialmente valioso: combatir el blanqueo de dinero y dificultar la economía sumergida.

4. Diseño del ecosistema

4.1. Visión general de la arquitectura

Tras haber presentado NextGen AML de manera más general, este capítulo profundiza en su diseño y explica cómo encajan entre sí las piezas que lo componen, ya que como se ha dicho anteriormente, la principal aportación de la propuesta no reside en cada componente por separado, sino en la forma en que se conectan para funcionar como un único sistema cerrado. Como se adelantó en el capítulo anterior, la arquitectura se apoya en tres componentes (el Financial Sandbox, el agente atacante y el agente defensor) que no operan como capas independientes, sino que se relacionan formando un bucle continuo de generación, detección y aprendizaje.

El punto de partida del flujo es el agente atacante, que actúa dentro del Financial Sandbox generando operaciones de todo tipo, tanto lícitas como ilícitas. Estas operaciones no se producen de forma aislada, sino que se van registrando en forma de grafo de transacciones, en el que las cuentas se representan como nodos y los movimientos entre ellas como las aristas que los conectan. Ese grafo es la pieza que articula todo el sistema, y es que funciona como un lenguaje común a través del cual los dos agentes se comunican sin interactuar directamente entre sí. De este modo, el sandbox mantiene en todo momento una representación viva del sistema financiero simulado, que crece y cambia a medida que el atacante va introduciendo nuevos movimientos.

A partir de dicho grafo entra en juego el agente defensor, que analiza e intenta distinguir qué operaciones resultan sospechosas y cuáles responden a un comportamiento normal. Aquí se produce el segundo paso clave del flujo, ya que el resultado de este proceso no se limita a una simple etiqueta, sino que se reintroduce en el sistema como señal de aprendizaje para ambos agentes. Por un lado, el atacante recibe una recompensa que depende de si ha logrado o no pasar desapercibido, lo que le indica qué tipo de estrategias le conviene seguir utilizando. Por otro lado, el defensor aprende de sus aciertos y errores, ajustando su capacidad de detección de cara a las operaciones futuras. Con ello se completa el bucle de coevolución que da sentido a toda la arquitectura.

Es importante destacar de nuevo que este diseño en forma de bucle es lo que diferencia a la propuesta de un conjunto de tecnologías. Cada componente cumple una función específica y podría desarrollarse de manera autónoma, pero su valor real surge de la interacción entre ellos: el atacante necesita un defensor al que enfrentarse, el defensor necesita las operaciones que genera el atacante y ambos necesitan el sandbox como terreno donde coexistir. Es esa dependencia de los distintos componentes entre sí la que convierte al sistema en algo capaz de evolucionar por sí mismo.

Esta visión global que se resume de forma esquemática, funciona también como mapa para resto del capítulo. En los siguientes apartados se detalla cada uno de los componentes por separado: primero el Financial Sandbox, que constituye el entorno en el que todo ocurre; después el agente atacante, con su lógica y su capacidad de generar nuevas tipologías; y a continuación el agente defensor, encargado de detectarlas. Por último, el capítulo termina analizando cómo ambos agentes evolucionan juntos, lo que convierte todo el sistema en un conjunto integrado que va cambiando progresivamente.

4.2. El entorno de simulación: Financial Sandbox

El primero de los tres componentes, y la base sobre la que se sostienen los otros dos, es el Financial Sandbox. Se trata de un entorno de simulación que recrea un sistema financiero ficticio, en el que se generan cuentas y transacciones sintéticas y dentro del cual operan e interactúan tanto el agente atacante como el defensor. En esencia, se materializa la idea de los entornos sintéticos introducida en el apartado 2.3.5, aplicándola a esta propuesta: un espacio artificial que imita el comportamiento de un sistema real sin utilizar datos de clientes.

La necesidad de un entorno de este tipo responde a tres motivos principales. El primero es la confidencialidad, ya que los datos reales de transacciones son escasos y están fuertemente protegidos, de modo que trabajar con datos sintéticos permite experimentar sin comprometer información sensible. El segundo es el control, y es que un entorno simulado hace posible diseñar a voluntad los escenarios y las tipologías que se quieren estudiar, algo impensable cuando se depende de datos reales en los que el blanqueo aparece de manera limitada y poco estructurada. El tercero es la reproducibilidad, puesto que el sandbox ofrece un terreno estable sobre el que entrenar y evaluar los modelos, puedes volver a generar el mismo tipo de escenarios de forma consistente y controlada y, además, con la ventaja de conocer con total certeza qué operaciones son fraudulentas y cuáles no. Este enfoque no es nuevo: simuladores como PaySim, orientado a la detección de fraude en pagos móviles (Lopez-Rojas et al., 2016), o los generadores sintéticos más recientes desarrollados específicamente para la lucha contra el blanqueo (Altman et al., 2023), han demostrado la utilidad de la simulación en contextos donde los datos reales son escasos.

A nivel de diseño, el Financial Sandbox se organiza en torno al grafo de transacciones que ya se ha mencionado. Los nodos representan las cuentas o las entidades que participan en el sistema, mientras que las aristas representan las transacciones que se producen entre ellas. Ambas partes, contienen atributos e información que enriquecen la simulación, como el importe de cada operación, el canal, la procedencia o el destino, el momento en que se realiza o el tipo de movimiento del que se trata entre otras. Sobre esta estructura, el sandbox genera además una actividad de fondo que reproduce el comportamiento normal de un sistema financiero, de manera que las operaciones legítimas convivan con las que introduce el atacante y así el problema se pueda parecer a la realidad en la medida de lo posible

Por último, conviene destacar el papel que cumple este entorno en la dinámica atacante-defensor. El sandbox no es un simple almacén de datos, sino el escenario controlado en el que ambos agentes desarrollan su actividad: el atacante introduce sus operaciones y el defensor observa el grafo para tratar de detectarlas. Al gestionar el paso del tiempo, la generación de la actividad de fondo y el registro de cada transacción, el sandbox proporciona el ecosistema que hace posible la interacción entre los dos agentes. Es, en definitiva, la base del sistema, y de su buen diseño depende que todo el ecosistema funcione de forma coherente.

4.3. El agente atacante (Reinforcement Learning)

El segundo componente es el agente atacante, construido sobre el aprendizaje por refuerzo y encargado de introducir la capacidad de anticipación en el sistema. Su función es simular el comportamiento de un posible delincuente y aprender, por sí mismo, a generar formas de blanqueo capaces de eludir los controles. A diferencia de los enfoques tradicionales, que se

limitan a reaccionar ante lo ya conocido, este agente explora posibles escenarios futuros, actuando como motor de mejora continua para el defensor. En los dos subapartados siguientes se desarrolla primero su planteamiento como problema de aprendizaje por refuerzo y, posteriormente, su papel más relevante: la generación de nuevas tipologías de blanqueo.

4.3.1. Planteamiento conceptual

Desde un punto de vista conceptual, el agente atacante se plantea como un problema de aprendizaje por refuerzo, es decir, como un sistema que aprende a actuar mediante la interacción continua dentro de una plataforma y la información que recibe de sus resultados. Por ello, llevar el problema del blanqueo a este enfoque implica definir claramente sus elementos y adaptarlos al contexto de la propuesta.

El escenario con el que interactúa el agente es el Financial Sandbox descrito en el apartado anterior, que le ofrece el sistema financiero simulado sobre el que actuar. El estado representa la situación de ese sistema en un momento concreto, es decir, la configuración del grafo de transacciones en cada instante: las cuentas existentes, los movimientos realizados y toda la información que esté asociada a ellos. A partir de ese estado, el agente puede elegir entre un conjunto de acciones (dispone de distintas alternativas de qué hacer), que se corresponden con las operaciones o movimientos que puede ejecutar dentro del sandbox: crear nuevas cuentas dentro del sistema, realizar transferencias entre ellas con distintos importes, decidir el momento en que se ejecutan (fechas y horas), dividirlas en múltiples operaciones más pequeñas… entre otras muchas cosas.

El elemento clave del sistema es la recompensa, ya que es la señal que guía el aprendizaje. En este caso, el agente recibe una recompensa positiva cuando consigue mover fondos sin ser detectado por el defensor, y una penalización cuando la operación es identificada como sospechosa. A partir de esta señal, el agente aprende a distinguir qué estrategias o métodos son eficaces y cuáles no, reforzando aquellos que funcionan y descartando los que conducen a la detección. A través de la repetición continua de este proceso, el agente va construyendo su política, es decir, va ajustando las posibilidades de qué acción tomar en cada estado, subiendo la probabilidad de las jugadas que le funcionan y bajando la de las que le detectan. Del mismo modo, también va ciñendo la estrategia a nivel general con el objetivo de maximizar la recompensa a largo plazo.

De este modo, mediante un proceso de prueba y error, el agente va ajustando su comportamiento hacia la evasión. Las estrategias que le permiten pasar desapercibido se refuerzan, mientras que aquellas que terminan siendo detectadas se descartan, de manera que el agente aprende, poco a poco, a blanquear de forma cada vez más eficaz (descartando e implementando nuevas estrategias continuamente). Conviene insistir en que todo lo anterior se mantiene en el plano del diseño conceptual; la forma concreta en que esta lógica se traduce de manera técnica y se pone en práctica se detalla más adelante, en el capítulo 5.

4.3.2. Generación de nuevas tipologías de blanqueo

Si el planteamiento conceptual establece las reglas del juego, la verdadera aportación del agente atacante está en lo que puede hacer dentro de ellas: generar nuevas tipologías de blanqueo.

Guiado por la señal de recompensa, el agente no se limita a repetir esquemas predefinidos, sino que explora el amplio espacio de operaciones que tiene a su disposición y combina distintos patrones para encontrar formas cada vez más eficaces de mover fondos sin ser detectado. Es precisamente en esta capacidad donde se sitúa el mayor valor diferencial de la propuesta.
Para entenderlo, conviene partir de las tipologías de blanqueo más conocidas, que funcionan como bloques de construcción a partir de los cuales el agente puede operar. Entre ellas destacan el pitufeo o fragmentación de importes, que consiste en dividir una gran cantidad de dinero en numerosas transferencias pequeñas para mantenerlas por debajo de los umbrales de alerta; los flujos circulares, en los que el dinero recorre una cadena de cuentas hasta volver a su origen con apariencia legítima; los abanicos de dispersión, en los que un origen reparte fondos entre múltiples destinatarios; el encadenamiento de monederos digitales, que dificulta el rastreo mediante sucesivas billeteras; y las capas de intermediación, que consisten en utilizar cuentas "mula" o sociedades pantalla para ocultar el origen de los fondos. Muchos de estos patrones han sido formalizados en la literatura como estructuras concretas dentro del grafo de transacciones (Altman et al., 2023), lo que confirma que se trata de comportamientos identificables y, por tanto, reproducibles en un entorno simulado.

La clave está en que el agente no se conforma con aplicar estas tipologías tal cual, sino que puede modificarlas, encadenarlas y combinarlas de múltiples maneras. Por ejemplo, podría combinar varias tipologías dentro de una misma operación, comenzando por fragmentar los fondos en pequeñas cantidades, distribuyéndolos después entre numerosas cuentas, billeteras digitales, criptomonedas, plataformas de inversión o de intercambio de divisas… y haciéndolos circular posteriormente a través de distintos intermediarios antes de reagruparlos nuevamente. También podría modificar continuamente la forma en que ejecuta estas operaciones, de manera que dos esquemas con el mismo objetivo de colocación, presentasen patrones muy diferentes. Incluso podría crear redes de transacciones con múltiples niveles de profundidad, donde el agente elaborase estructuras de flujos complejas antes de alcanzar su destino final. En todos estos casos, la idea no sería repetir tipologías ya conocidas, sino explorar nuevas combinaciones y variantes que resulten más difíciles de detectar ya que se alejan de cualquier lógica conocida hoy día.

Aquí reside el valor diferencial de este enfoque, y es que, al estar guiado por la recompensa y no por un catálogo cerrado de esquemas, el atacante es capaz de generar amenazas que todavía no se han observado en la realidad o por lo menos no se conocen. En lugar de esperar a que aparezca una nueva tipología para estudiarla, el sistema la produce internamente y se anticipa a ella, lo que dota a la propuesta de una capacidad preventiva de la que carecen los sistemas tradicionales. Conviene precisar, no obstante, que esta capacidad se describe aquí de manera literaria. La prueba de concepto del capítulo 5 implementa únicamente una versión reducida de esta dinámica, suficiente para ilustrar su funcionamiento, aunque más simplificada que un sistema completo. Aun así, es precisamente esta capacidad de explorar lo que aún no existe lo que justifica el papel central del agente atacante dentro del ecosistema.

4.4. El agente defensor (GNNs y Transformers)

El tercer componente del ecosistema es el agente defensor, cuya función es la opuesta a la del atacante: detectar las operaciones sospechosas que este introduce en el sistema. Para ello, combina las dos tecnologías que ya se presentaron en el capítulo anterior, aprovechando lo mejor de cada una. Por un lado, las redes neuronales sobre grafos (GNNs) le permiten captar la

dimensión estructural del problema, es decir, cómo se relacionan y operan las cuentas, las estructuras y las transacciones entre sí. Por otro lado, los Transformers aportan la dimensión temporal ayudándole a representar el orden, momento y el ritmo de las operaciones para detectar si esa secuencia resulta sospechosa.

Los dos subapartados siguientes desarrollan en detalle cada una de estas vertientes: primero, la representación del sistema financiero como un grafo y, después, el análisis de los patrones de red junto con su evolución temporal.

4.4.1. Modelado del sistema financiero como grafo

Antes de poder detectar operaciones sospechosas, el agente defensor necesita representar el sistema financiero de una forma que le permita analizar las relaciones existentes entre cuentas y transacciones. Para ello utiliza un grafo, tal y como se introdujo en el apartado 2.3.1. La diferencia respecto a un enfoque tradicional es importante, ya que no se trata de analizar cada operación de manera aislada, sino de situarla dentro de la red de movimientos a la que pertenece. De hecho, buena parte de la investigación reciente sobre detección de fraude financiero ha adoptado precisamente este tipo de representación por su capacidad para captar relaciones complejas que otros modelos pasan por alto (Motie y Raahemi, 2024).

En este grafo, los nodos representan las cuentas, los clientes o las entidades que participan en el sistema, mientras que las aristas representan las transacciones que se producen entre ellos. Tanto unos como otras llevan asociada información que enriquece la representación.

En el caso de las operaciones, pueden considerarse atributos como el importe, el momento exacto en que se realiza (fecha y hora), el tipo de operación (ingreso o retirada en efectivo o electrónico, transferencia, pago con tarjeta, cambio de divisas, compra de cripto activos...), la dirección del flujo (origen y destino). A ello, se pueden añadir otras variables relevantes como la divisa utilizada, el canal a través del cual se ha ejecutado (banca online, aplicación móvil, oficina física o cajero automático) y la jurisdicción del origen y del destino de los fondos.

Si nos referimos a características de un cliente, pueden incluirse variables como el tipo de cliente (particular, empresa, ONG o proveedor de servicios financieros), su profesión, su país de residencia y la antigüedad de la relación con la entidad. También resulta relevante el historial de su actividad financiera, así como los ingresos y gastos habituales y las interacciones previas con otras cuentas o clientes dentro del sistema, lo que permite contextualizar su comportamiento dentro de la red.

Por lo que respecta a las cuentas y entidades, pueden considerarse parámetros como el tipo de cuenta (corriente, ahorro, corporativa o billetera digital), el saldo disponible y su historial de transacciones. De igual manera, es relevante analizar el número de conexiones con otras cuentas, su rol dentro del sistema (origen de fondos, intermediaria o destino final) y si actúa como cuenta puente o intermediaria dentro de determinadas estructuras de movimiento.

De este modo, el grafo no es una simple lista de movimientos, clientes o cuentas, sino que refleja cómo circula el dinero dentro del sistema y sobre todo cómo se relacionan e interactúan en función de las cualidades de cada uno.

La construcción de dicho grafo se realiza a partir de las operaciones que se generan en el Financial Sandbox. Cada vez que el atacante ejecuta un movimiento, este se incorpora como una nueva conexión (arista) entre las cuentas implicadas, de manera que el grafo va creciendo

y cambiando a medida que se suceden las transacciones. Sobre esta estructura, el defensor dispone de una información de red que un enfoque tabular perdería por completo, y es que puede observar los vecindarios y características de cada nodo (es decir, con qué otras cuentas se relaciona), las conexiones y los caminos que unen unas cuentas con otras y las subestructuras que se forman entre ellas, como cadenas, abanicos de dispersión o ciclos. Conviene ilustrarlo con un ejemplo: En una tabla se vería una transferencia de 6.000 euros como un registro más dentro de una base de datos. Sin embargo, con el grafo es posible observar más en detalle que esa misma cuenta ha recibido durante distintas semanas varias transferencias de importes similares procedentes de distintas cuentas; que posteriormente fragmenta y redistribuye una parte entre varios destinatarios y otros vuelven a circular a través de cuentas intermedias concurriendo finalmente en una misma entidad o beneficiario final (UBO). Analizada de forma aislada, ninguna de estas operaciones resultaría especialmente llamativa; sin embargo, al observar conjuntamente todas las relaciones y comportamientos, el grafo revela una estructura compleja de movimientos coordinados que puede ser indicativo de "marear fondos" como una estrategia de blanqueo de capitales.

En definitiva, esta representación del sistema financiero como un grafo constituye la base sobre la que el defensor aplica las redes neuronales sobre grafos (GNNs). Sin esta representación, la detección se limitaría a mirar operaciones sueltas; con ella, el defensor dispone del terreno necesario para analizar la estructura completa de la red, que es justamente lo que se aborda en el siguiente apartado.

4.4.2. Detección de patrones de red y de su dimensión temporal

Una vez construido el grafo, el agente defensor puede abordar su tarea principal: distinguir qué operaciones resultan sospechosas y cuáles responden a un comportamiento normal. Para ello no se fija en un único aspecto, sino que combina dos perspectivas, la estructural y la temporal, de manera que cada una aporte una parte de la información y, al unirse, ofrezcan una visión mucho más completa de lo que ocurre dentro de la red.

La dimensión estructural corre a cargo, como ya hemos visto, de las redes neuronales sobre grafos (GNNs), y su funcionamiento se apoya en la idea de propagación de información entre los nodos vecinos. Yendo más al detalle, cada cuenta (nodo) parte de sus propios atributos y, en cada paso o capa de la red, los combina con la información de las cuentas con las que está conectada. De este modo, en la primera capa un nodo "conoce" a sus vecinos directos; en la segunda, a los vecinos de sus vecinos; y así sucesivamente, de manera que la información se va difundiendo por el grafo y cada cuenta termina representada no solo por la interacción que hace o recibe, sino por el papel que ocupa dentro de la red. El resultado de este proceso es una representación numérica de cada nodo (lo que se conoce como embedding) que resume su comportamiento y su entorno. Gracias a ello, el defensor es capaz de reconocer los patrones de red típicos del blanqueo (cadenas de transferencias, abanicos de dispersión, flujos circulares o capas de intermediación) que, como ya se ha visto, no son otra cosa que subestructuras concretas dentro del grafo (Weber et al., 2019; Altman et al., 2023). Algunas variantes, como las redes de atención sobre grafos, afinan aún más este mecanismo asignando un peso distinto a cada vecino según lo relevante que sea (Veličković et al., 2018), de modo que una cuenta intermediaria conectada con decenas de orígenes reciba más atención que una conexión puntual y sin apenas importancia.

Sin embargo, la estructura por sí sola no es suficiente, y aquí es donde entran en juego los Transformers y su dimensión temporal, que constituyen una pieza igual de importante. Y es que el blanqueo no se entiende únicamente por cómo se conectan las cuentas, sino también por cuándo y en qué orden se mueve el dinero. Las operaciones de cada cuenta pueden ordenarse como una secuencia a lo largo del tiempo, y es esa secuencia la que procesa el Transformer mediante su mecanismo de atención. Dicho mecanismo permite al modelo decidir qué operaciones ya ejecutadas son relevantes para entender una operación concreta, con independencia de lo alejadas que estén en el tiempo (Vaswani et al., 2017). En la práctica, esto le permite captar dinámicas que en otro sistema pasarían desapercibidas: la rapidez con la que se encadenan los movimientos (por ejemplo, fondos que entran y salen de una cuenta en cuestión de minutos), transferencias similares que se repiten de forma periódica, incrementos repentinos de actividad o movimientos que se producen inmediatamente después de recibir un ingreso significativo. Esta perspectiva resulta especialmente útil en la fase de diversificación del blanqueo, donde los fondos suelen desplazarse de forma rápida y fragmentada para dificultar su seguimiento, y coincide con líneas de investigación que han demostrado la utilidad de incorporar la dimensión temporal al análisis de grafos dinámicos (Pareja et al., 2020).

La verdadera fortaleza del sistema aparece al combinar ambas dimensiones, ya que cada una, por separado, deja escapar información muy valiosa. Un análisis puramente estructural detectaría un abanico de dispersión, pero dejaría mucha incertidumbre ya que no se sabría si se ha ejecutado de golpe en segundos, de forma prolongada o durante mucho más espacio de tiempo. Del mismo modo, uno puramente temporal podría detectar una ráfaga de operaciones anómalas, pero sin comprender cómo se conectan entre sí las cuentas implicadas. Al unir ambas, el defensor obtiene una visión más completa del comportamiento que se ha ejecutado y puede valorar con mayor precisión el nivel de riesgo asociado a cada patrón: ese mismo abanico que reparte fondos entre veinte cuentas no levanta las mismas sospechas si se completa a lo largo de un año que si se ejecuta en apenas unos minutos o pocos días tras un único ingreso.

Finalmente, todo este proceso desemboca en una señal de sospecha, es decir, una puntuación o probabilidad que el defensor asigna a cada operación, cuenta o subestructura para indicar hasta qué punto la considera sospechosa. Cuando esa puntuación supera un determinado umbral, la operación se marca como sospechosa y quedaría pendiente de revisión, igual que ocurriría con una alerta en un sistema real. No obstante, dicha señal no termina ahí, sino que es justamente la que va alimentando el bucle del sistema, ya que sirve para determinar si el atacante ha logrado pasar desapercibido y, al mismo tiempo, para que el propio defensor aprenda de sus aciertos y de sus errores.

4.5. La dinámica de coevolución entre atacante y defensor

Una vez descritos los dos agentes por separado, queda por explicar el elemento que los conecta y que, en realidad, da sentido a todo el ecosistema: la dinámica de coevolución. Como se vio al hablar de la inteligencia artificial adversarial en el apartado 2.3.4, la idea de fondo consiste en enfrentar a dos sistemas con objetivos opuestos, de modo que el avance de uno obligue al otro a mejorar. Aplicada a esta propuesta, esta lógica es la que transforma los distintos componentes en un sistema integrado capaz de mejorar de forma continua.

El mecanismo es, en esencia, una carrera continua entre ambos agentes. Cada vez que el atacante descubre una nueva tipología que consigue evadir la detección, obliga al defensor a perfeccionarse para aprender a reconocerla; y cada vez que el defensor mejora y cierra esa vía,

empuja al atacante a idear estrategias todavía más sofisticadas. No obstante, conviene aclarar que ese ajuste mutuo no significa que ambos agentes funcionen igual, y es que cada uno aprende con una técnica distinta. El atacante se apoya en el aprendizaje por refuerzo, ajustando las probabilidades de qué operaciones ejecutar según la recompensa que obtiene, mientras que el defensor optimiza los pesos internos de sus modelos (GNNs y Transformers) para clasificar mejor cada operación. Lo que comparten no es el método, sino el hecho de mejorar enfrentándose y perfeccionándose el uno al otro, y eso es precisamente lo que se entiende por coevolución.

A nivel de diseño, este proceso puede describirse como un ciclo continuo compuesto por varias etapas. En primer lugar, el agente atacante observa el estado actual del Financial Sandbox, es decir, la situación del sistema financiero simulado en ese momento: las cuentas existentes y las transacciones previas registradas junto con sus atributos, los patrones que ya han sido detectados y el contexto general de la red. A partir de esa información, decide qué acciones ejecutar y genera nuevas operaciones dentro del entorno. Estas pueden ser legítimas o potencialmente ilícitas e incluir desde transferencias simples hasta estructuras más complejas que combinen fragmentación de fondos, dispersión entre múltiples destinatarios, utilización de cuentas intermediarias o movimientos escalonados en el tiempo.

Cada una de estas acciones modifica el estado del sistema y queda registrada en el grafo de transacciones. A medida que se incorporan nuevas operaciones, el grafo crece y evoluciona, reflejando tanto la actividad ordinaria generada por el sandbox como las estrategias introducidas por el atacante. De esta manera, el sistema dispone en todo momento de una representación actualizada de las relaciones existentes entre cuentas, clientes y entidades.

Una vez actualizado el grafo, entra en acción el agente defensor. Utilizando la información estructural proporcionada por las GNNs y la información temporal captada mediante los Transformers, analiza las nuevas operaciones y el contexto en el que se producen. Como resultado, asigna a cada operación, cuenta o subestructura una puntuación de riesgo que refleja el grado de sospecha asociado. Cuando dicha puntuación supera un determinado umbral, la actividad es clasificada como potencialmente ilícita.

El resultado de esta evaluación se utiliza entonces como mecanismo de retroalimentación para ambos agentes. Por un lado, el atacante recibe una recompensa o una penalización en función de si sus operaciones han conseguido pasar desapercibidas o han sido detectadas. Esta señal le permite identificar qué estrategias están funcionando mejor y cuáles resultan ineficaces, ajustando progresivamente las probabilidades de las acciones que ejecutará en futuras iteraciones. Por otro lado, el defensor compara sus predicciones con la realidad conocida dentro del entorno simulado y utiliza esa información para corregir errores, reforzar los patrones que ha identificado correctamente y mejorar su capacidad de detección frente a nuevos comportamientos.

Finalmente, una vez actualizados ambos agentes, el ciclo vuelve a comenzar sobre un entorno ligeramente diferente al anterior. El atacante dispone ahora de nuevas estrategias aprendidas y el defensor cuenta con una mayor experiencia para reconocerlas. Como consecuencia, cada iteración incrementa consecutivamente el nivel de sofisticación del enfrentamiento, generando una dinámica de mejora continua en la que ambos agentes evolucionan conjuntamente y se adaptan de forma progresiva a los avances de su oponente.

Es esta dinámica la que dota al sistema de su carácter adaptativo y anticipatorio, y la que lo diferencia de los enfoques tradicionales, que permanecen estáticos hasta que se actualizan manualmente. En lugar de esperar a que el delincuente dé el primer paso con nuevas tipologías, el propio ecosistema las genera, las detecta y aprende de ellas internamente. De este modo se completa el bucle de generación, detección y aprendizaje introducido en la visión general de la arquitectura del apartado 4.1. Conviene recordar, sin embargo, que esta coevolución continua representa el diseño conceptual al que aspira la propuesta; la prueba de concepto del capítulo 5 reproduce únicamente una versión muy reducida de esta idea, suficiente para ilustrar su funcionamiento. Aun así, es este proceso de adaptación mutua entre atacante y defensor el que constituye el núcleo de NextGen AML y el que aporta sentido al conjunto del ecosistema.

5. Prueba de concepto: implementación en Google Colab

5.1. Alcance y propósito de la prueba de concepto

Una vez presentado el diseño conceptual del ecosistema en el capítulo anterior, este capítulo da el paso de llevarlo a la práctica mediante una prueba de concepto, es decir, una demostración funcional que traduce esta propuesta a código y la pone a funcionar de principio a fin. Dicha demostración se ejecuta en Google Colab y materializa el ecosistema de NextGen AML a través de tres scripts encadenados que reproducen los tres componentes que forman esta idea descritos anteriormente. El primero es el Financial Sandbox, encargado de generar el sistema financiero sintético y de estructurarlo como un grafo de transacciones; el segundo es el agente atacante, que introduce en ese sistema toda la actividad transaccional, tanto las operaciones legítimas que actúan como camuflaje como las operaciones ilícitas de blanqueo; y el tercero es el agente defensor, que trata de detectar estas últimas. La interacción entre estos tres componentes da lugar a la dinámica de coevolución adversarial entre ambos agentes, que es precisamente donde se cierra el bucle descrito en el capítulo 4.

Conviene precisar con honestidad qué demuestra exactamente esta prueba y qué no pretende ser, ya que de ese equilibrio depende que se interprete correctamente. Por un lado, lo que demuestra es que el enfoque adversarial es viable y que los tres componentes no solo funcionan de manera aislada, sino que interactúan entre sí sobre un mismo grafo, de principio a fin y sin intervención manual ni humana. A lo largo de la ejecución, el sistema genera datos sintéticos, los representa como una red de distintos flujos, introduce sobre ella distintas tipologías de blanqueo, las detecta y mide su capacidad de detección, produciendo métricas y visualizaciones que permiten observar el comportamiento de ambos agentes. Dicho de otro modo, esta prueba práctica permite comprobar que lo planteado en el plano conceptual puede trasladarse con éxito a una implementación funcional.

Por otro lado, y con la misma claridad, conviene precisar lo que no demuestra este proyecto, ya que no se pretende reproducir un entorno bancario real ni constituir un sistema listo para su despliegue operativo en una entidad existente. Toda la información utilizada es sintética y el escenario se limita a un único banco ficticio llamado NextGen Bank, con un número reducido de cuentas y clientes junto con sus respectivos contactos, con los cuales operan a través de un número predefinido de entidades y transacciones, muy alejado del volumen, de la complejidad y de los requisitos regulatorios que se presentan en un entorno financiero auténtico. Se trata, por tanto, de una prueba de concepto (PoC) cuyo objetivo es validar que la viabilidad del enfoque sea totalmente posible, no desarrollar una solución comercial o de producción.

Precisamente esta simplificación constituye una de sus principales ventajas. Y es que, al ser una prueba de concepto, no se intenta resolver el problema a escala real, sino demostrar que la idea es factible y que merece la pena desarrollarla, reduciéndola a un tamaño manejable en el que se pueda observar, medir y entender con precisión. Asimismo, este enfoque abre una vía de desarrollo interesante de cara a una futura implementación profesional, ya que un entorno controlado y ficticio en el que se conoce con total certeza qué operaciones son lícitas y cuáles no, permite evaluar de forma rigurosa cada acierto y cada error del defensor. Sobre esa base, el sistema podría entrenarse de forma continua y enfrentarse una y otra vez a amenazas cada vez más sofisticadas generadas por el propio agente atacante, de manera que fuese perfeccionándose poco a poco antes de enfrentarse a datos de un entorno real.

Esta idea es el propósito de llevar el ecosistema más allá de la demostración, la cual se retoma y se desarrolla en el capítulo 7. No obstante, en los apartados siguientes se detalla cómo se ha construido esta demostración, comenzando por el entorno de desarrollo y las herramientas

empleadas, para pasar después a la generación del grafo y, posteriormente, a la implementación concreta de cada uno de los dos agentes.

El código completo de la prueba de concepto (los tres scripts descritos) se encuentra disponible en un repositorio de GitHub creado específicamente para esta propuesta, cuyo enlace puede consultarse en el Anexo A.

5.2. Entorno y herramientas

Toda la prueba de concepto se ha desarrollado sobre Google Colab, un servicio que permite ejecutar cuadernos de Python directamente en la nube, sin necesidad de instalar ningún software especializado. Esta elección se debe sobre todo a su accesibilidad, ya que basta con un navegador para ejecutar el código, aprovechar los recursos de cómputo que ofrece Google y visualizar los resultados, facilitando así que el trabajo pueda reproducirse con comodidad en cualquier equipo. A ello se añade que las librerías empleadas vienen ya preinstaladas en el entorno, de modo que no es necesario instalar nada por separado para poner el sistema en marcha.

Sobre este entorno, el proyecto se apoya en un conjunto de librerías de Python, cada una centrada en una función específica. Las librerías NumPy y Pandas se emplean para el cálculo numérico y el tratamiento de los datos, organizándolos mediante estructuras tabulares (los llamados DataFrames) que facilitan el poder operar con la información de una forma cómoda. Networkx es la librería con la que se construye y se gestiona el grafo, es decir, la que hace posible crear los nodos y las aristas y consultar las relaciones existentes entre ellos. Para las representaciones visuales se emplean dos herramientas complementarias: matplotlib, que genera las figuras estáticas en dos dimensiones (por ejemplo, las curvas de aprendizaje o las matrices de resultados), y plotly, encargada de crear visualizaciones tridimensionales e interactivas del grafo, exportadas posteriormente como archivos HTML que se pueden abrir y explorar en cualquier navegador. Por último, PyTorch es la librería sobre la que se construyen los modelos del agente defensor (tanto la GNN como el Transformer), y es la que permite entrenarlos y ajustar sus parámetros internos; conviene señalar que, dado el tamaño reducido de este proyecto, estos modelos funcionan sin necesidad de tarjeta gráfica, ejecutándose directamente sobre la CPU del entorno. A todo ello se suman algunas utilidades más sencillas, pero igualmente necesarias, como openpyxl, que es la que permite leer los ficheros Excel de entrada; random y secrets, que gestionan la aleatoriedad y la semilla del sistema; o datetime, que maneja las fechas y la ventana temporal en la que se distribuyen todas las transacciones.

Para que el sistema funcione, es necesario proporcionarle los datos de partida, que se aportan mediante dos ficheros Excel que deben subirse a Colab (en concreto, a la ruta /content/). El primero, llamado «Clientes_NextGen_Bank_y_asociados.xlsx», se apoya en dos hojas. La primera, «Personas Principales», recoge los 150 clientes del banco NextGen Bank junto con sus datos básicos (el identificador de cada cliente, su nombre completo y su nivel de riesgo). La segunda, «Contactos Asociados», recoge las diez personas asociadas que tiene cada cliente, con las que mantiene las distintas relaciones de las que después surgen los flujos de transacciones. El segundo Excel, llamado «Paises_Bancos_Riesgo.xlsx», asigna un nivel de riesgo a cada país y jurisdicción, así como a los bancos correspondientes a través de los cuales operan las distintas personas implicadas. Ambos ficheros constituyen la base de datos inicial

del Financial Sandbox, ya que a partir de ellos se genera el escenario sintético sobre el que posteriormente interactúan los dos agentes.

En cuanto a los resultados, la ejecución de los 3 scripts produce dos tipos de salidas. Las primeras son los ficheros HTML con las visualizaciones tridimensionales interactivas, que permiten observar la red desde distintos ángulos: el grafo del propio sandbox a modo de mapa de cuentas, los flujos de transacciones que va introduciendo el atacante y la detección que realiza el defensor. A ellas se añade un HTML de coevolución que integra varias de estas vistas mediante botones, de manera que se puede alternar entre ellas y seguir el proceso completo de forma guiada. Las segundas son los ficheros CSV con los datos generados (clientes, personas, entidades y transacciones), que dejan registrada toda la información en un formato tabular fácil de consultar. Tanto unos como otros pueden descargarse directamente desde Colab una vez generados.

Por último, conviene destacar dos aspectos que afectan al conjunto. El primero es la reproducibilidad, que se garantiza mediante una semilla: por defecto es aleatoria (aunque el sistema imprime la semilla efectiva que ha utilizado, de modo que pueda anotarse), pero también puede fijarse un valor concreto para repetir exactamente el mismo escenario tantas veces como se quiera. El segundo es que los tres scripts no operan de forma independiente, sino que se integran entre sí, y es que el defensor importa y reutiliza tanto el sandbox como el atacante, lo que permite que la demostración se ejecute de manera encadenada y coherente.

5.3. Generación del grafo de transacciones sintético

El primer componente de la demostración es el Financial Sandbox, cuya labor consiste en construir, a partir de los dos ficheros Excel descritos, el sistema financiero sintético sobre el que se apoya toda la prueba de concepto. En otras palabras, es el encargado de crear el escenario inicial de la simulación, es decir, el entorno con todos los participantes y sus características, y de definir cómo se genera y se representa la actividad que después tendrá lugar sobre él. Todo ello da lugar al grafo de transacciones que, posteriormente, analizará el agente defensor.

Antes de entrar en detalle, conviene señalar que todo el comportamiento del sandbox se gobierna desde un único bloque de configuración centralizado (denominado CONFIG), en el que se recogen los principales parámetros de la simulación. Esta organización permite modificar dicho escenario de forma sencilla sin necesidad de alterar el resto del código y facilita conocer, de un solo vistazo, los valores que determinan la generación de datos. A continuación, se muestra un extracto del primer script (financial_sandbox.py) con los parámetros más relevantes:

Figura 5.1. Extracto del bloque de configuración (CONFIG) del Financial Sandbox.

Como se puede observar, estos parámetros definen los aspectos principales del escenario. La semilla (SEMILLA) controla la reproducibilidad ya comentada; la ventana temporal la cual abarca 90 días (VENTANA_DIAS), empieza el 1 de enero de 2025 (FECHA_INICIO); y los tamaños fijan los 150 clientes (N_CLIENTES) y los 10 contactos de cada uno (CONTACTOS_POR_CLIENTE).
El modelo de generación de los flujos establece cuántas transacciones realiza cada cliente a lo largo del periodo (entre 20 y 100) y cuántas componen cada flujo (entre 1 y 5 operaciones), así como la probabilidad base de que un flujo sea complejo (PROB_FLUJO_COMPLEJO = 0,18, que después se modula según el riesgo) y el nivel de riesgo mínimo que debe tener un cliente para poder generar operaciones ilícitas (RIESGO_MIN_ILICITO = 3). Por último, el apartado de lo ilícito fija una proporción orientativa de operaciones ilícitas (PROPORCION_ILICITO = 0,08, es decir, alrededor del 8 % de todas las operaciones, que en la práctica se mantiene en torno al 8-10 %) y el umbral de referencia del pitufeo (PITUFEO_UMBRAL = 10.000). Estos dos últimos aportan un realismo controlado, ya que reproducen el hecho de que el blanqueo es minoritario frente a la enorme masa de actividad legítima, reproduciendo el comportamiento de un sistema financiero real y de que el fraccionamiento de los importes busca mantenerse por debajo de los umbrales de alerta.

Aclarado el marco de configuración, el sandbox genera los distintos participantes del sistema. En el centro se sitúa el propio banco, NextGen Bank, una entidad bancaria ficticia, regulada y de bajo riesgo, a través de la cual se canaliza toda la actividad. A su alrededor se generan sus 150 clientes, cada uno con un perfil propio que incluye un sector de actividad al que pertenecen (hostelería, tecnología, construcción, comercio, consultoría, inmobiliario, logística, agroalimentario, textil o sanitario), un nivel de riesgo KYC medido en una escala de seis niveles (de "Muy bajo" a "Muy alto"), una nacionalidad con su correspondiente nivel de riesgo de jurisdicción (asignada de forma coherente con el riesgo del cliente) y una información económica declarada (ingresos, gastos y saldo). Estos atributos no son únicamente un adorno descriptivo, sino que condicionan el comportamiento posterior de cada cliente durante la simulación.
Como se ha indicado anteriormente, a cada cliente se le asocian, además, 10 contactos (1.500 personas asociadas en total), que representan a las personas con las que se relaciona cada cliente y que podrán actuar como las llamadas cuentas "mula". Cada uno de estos contactos tiene asignada una entidad donde opera (su cuenta), de manera que las personas vinculadas a clientes de mayor riesgo tienden a operar más con criptomonedas y billeteras, un sesgo que aporta mayor realismo a la simulación. Por último, el sistema incorpora 64 entidades externas: 14 de ellas con nombre real (exchanges de criptomonedas como Binance o Coinbase, bancos como BBVA o Deutsche Bank, aplicaciones de inversión como Trade Republic o Interactive Brokers y wallets o billeteras digitales como MetaMask o Trust Wallet) y otras 50 procedentes del Excel de países. Cada entidad lleva asociado su tipo (banco, exchange, aplicación de inversión o wallet), su jurisdicción, su nivel de riesgo también medido en una escala de seis niveles (de "Muy bajo" a "Muy alto"), y su condición de regulada o no, información que posteriormente resulta relevante para evaluar el riesgo de cada operación.

Una vez definidos los participantes, se establece el modelo de la actividad que se produce entre ellos. Las operaciones se reparten a lo largo de la ventana temporal ya indicada (90 días desde el 1 de enero de 2025) y pueden ser de distintos tipos (ingreso o retirada de efectivo, transferencia, pago con tarjeta, cambio de divisas, compra de criptomonedas e inversión), ejecutándose a través de distintos canales coherentes con cada operación (cajero, aplicación móvil, web u oficina). Los movimientos se organizan en flujos que conectan a los clientes con

sus contactos y con las entidades, siguiendo unas reglas diseñadas para reproducir el funcionamiento de un sistema financiero real: cada flujo lo inicia un cliente y se compone de entre una a cinco operaciones encadenadas, y ese cliente únicamente puede operar o con su propia cuenta de otra entidad (siguiendo la lógica del sandbox en cuanto a pesos y coberturas: sesgo por riesgo hacia cripto/wallets en los perfiles peligrosos) o con sus contactos, nunca directamente con otro cliente o con los contactos de otro cliente. Toda la actividad queda además anclada en NextGen Bank mediante los ingresos y retiradas correspondientes, que se consideran siempre legítimos. Como ya se ha adelantado, el nivel de riesgo modula la aparición de flujos más complejos, de modo que solo los clientes de riesgo medio o superior generan estructuras propias de blanqueo (tipologías ya descritas en el capítulo 4), mientras que los de menor riesgo se limitan a operaciones sencillas.

Toda esta actividad se representa como un grafo, que es la estructura que da sentido al conjunto. Los nodos se corresponden con el banco central (NextGen Bank), los clientes, las personas o contactos y las distintas entidades, mientras que las aristas representan las transacciones que se producen entre ellos. Cada arista lleva asociada una información detallada sobre la operación que se ha realizado: el importe, el momento exacto en que se realiza, el tipo de operación, el canal, el titular remitente y el destinatario, la entidad implicada, la jurisdicción de origen y de destino, el sentido (si el dinero se envía o se recibe) y, sobre todo de cara a la evaluación, si se trata de una operación lícita o ilícita, junto con la tipología a la que pertenece. Estos tres últimos datos constituyen la "verdad" del sistema, que solo se conoce por tratarse de un entorno sintético y que, como ya se ha señalado, permitirá después medir con precisión los aciertos y errores del defensor.

Finalmente, todo este escenario puede visualizarse gracias a la representación tridimensional e interactiva del grafo en formato HTML que el sandbox exporta. En dicha representación, NextGen Bank ocupa el centro y a su alrededor se disponen, mediante capas, los clientes, las personas asociadas y el resto de entidades, con las transacciones enlazadas como arcos curvos que las conectan. Un código de color distingue el sentido de cada operación (un tono para las enviadas y otro para las recibidas) y resalta las de carácter ilícito, y cada elemento puede consultarse al pasar el ratón por encima, además de poder girar o acercar la imagen. Conviene precisar que este grafo constituye únicamente el "escenario base" (el tablero ya poblado de actores, pero todavía sin ninguna actividad), y es justamente el punto de partida sobre el que entrarán en juego las operaciones que introduce el agente atacante, que genera los flujos de todas las transacciones.

5.4. Implementación del agente atacante

El segundo componente de la demostración es el agente atacante, que materializa en código la estrategia de aprendizaje por refuerzo descrita en el apartado 4.3. Tiene un doble objetivo: por un lado, aprender cuáles de las tipologías de blanqueo consiguen evadir mejor la detección y, por otro, introducir esas operaciones dentro del sandbox, generando así toda la actividad transaccional sobre dicho escenario. Conviene señalar, antes de entrar en materia, que el atacante reutiliza el Financial Sandbox sin modificarlo, apoyándose en su misma lógica de flujos para que las operaciones ilícitas convivan de forma natural con las legítimas.

El funcionamiento del atacante se divide en dos fases claramente diferenciadas. La primera es la fase de entrenamiento, en la que el agente aprende una política, es decir, una distribución de probabilidad sobre las cinco tipologías existentes que le indica con qué frecuencia conviene

emplear cada una. Para ello emplea un algoritmo de gradiente exponenciado (Exponentiated Gradient), un mecanismo de optimización sencillo pero eficaz que, tras cada intento, sube la probabilidad de la tipología empleada si ha logrado evadir la detección y la baja si ha sido descubierta, es decir, va ajustando los valores y probabilidades del modelo para mejorar sus resultados. El proceso se repite a lo largo de 3.000 episodios simulados (N_EPISODIOS_ENTRENAMIENTO) hasta que la política converge, y se ejecuta sobre un sandbox temporal e independiente (distinto de la semilla original mediante SEMILLA_OFFSET_ENTRENAMIENTO), evitando así alterar el escenario. La segunda es la fase de generación, en la que el agente, con la política ya aprendida, aplica lo conveniente para inyectar los flujos ilícitos sobre el sandbox real.

Al igual que ocurre con el Financial Sandbox, todo el comportamiento del agente se controla mediante un bloque de configuración centralizado. A continuación, se muestra un extracto del segundo script (agente_atacante.py) con los parámetros más relevantes:

Figura 5.2. Extracto del bloque de configuración (CONFIG_ATACANTE) del agente atacante.

Como se puede observar, estos parámetros regulan tanto el proceso de aprendizaje como la generación de las operaciones. La tasa de aprendizaje (TASA_APRENDIZAJE = 0,06) determina la intensidad con la que la política se actualiza tras cada episodio, mientras que la probabilidad mínima (PROB_MIN = 0,03) impide que alguna tipología desaparezca por completo, garantizando siempre un nivel mínimo de exploración. El importe de referencia (IMPORTE_REF_BLANQUEO = 60.000) y el factor de magnitud (K_MAGNITUD = 1,5) intervienen en la recompensa, como se explicará más adelante, mientras que la temperatura de generación (TEMPERATURA_GENERACION = 0,7) y la calibración del flujo complejo (CALIBRACION_FLUJO_COMPLEJO = 1,25) gobiernan la fase de generación, y los pesos del defensor-proxy (PROXY_W_*) definen su comportamiento.

El agente puede generar las cinco tipologías de blanqueo ya presentadas en el capítulo 4, cada una implementada con su propia función de inyección que la construye dentro del grafo: el pitufeo, que fracciona un importe en múltiples operaciones pequeñas por debajo del umbral de alerta; el flujo circular, en el que el dinero recorre una cadena de cuentas hasta regresar a su

origen; el abanico de dispersión, que reparte los fondos entre numerosos destinatarios; la concentración, donde múltiples cuentas convergen en un mismo destino; y las capas de intermediación, que introducen varios intermediarios para dificultar el seguimiento del dinero. Durante este proceso, cada flujo queda etiquetado con su tipología y esquema correspondiente, constituyendo así la "verdad" del sistema que posteriormente permitirá evaluar el rendimiento del agente defensor.

Una pieza que es clave en el entrenamiento es el defensor-proxy interno. Conviene aclarar que no se trata del agente defensor real (basado en GNNs y Transformers), sino de un mecanismo heurístico cuya única finalidad es proporcionar una señal de aprendizaje al atacante durante la fase de entrenamiento. Su función es estimar lo detectable que resulta cada flujo a partir únicamente de señales estructurales observables: la fragmentación con importes justo por debajo del umbral (PROXY_W_UMBRAL), la rapidez con la que suceden las operaciones (PROXY_W_VELOCIDAD), el riesgo de las entidades implicadas (PROXY_W_RIESGO), el número de saltos realizados entre cuentas (PROXY_W_HOPS) y el retorno del dinero a la cuenta de origen (PROXY_W_RETORNO). Con estas señales, ponderadas y ajustadas por un sesgo (PROXY_BIAS), calcula la probabilidad de que el flujo sea detectado, lo que le permite dar al atacante la señal de recompensa que guía su aprendizaje sin necesidad de recurrir a un defensor completo.

Dicha recompensa, además, incorpora la magnitud del dinero blanqueado. En lugar de premiar por igual todas las evasiones, el sistema recompensa más cuanto mayor es el importe que el agente logra mover sin ser detectado y, del mismo modo, penaliza en mayor medida cuando es descubierto moviendo una cantidad elevada. Para ello, el importe blanqueado de cada flujo se normaliza con el valor de referencia definido previamente y modula la recompensa mediante el factor de magnitud, de manera que no solo se incentiva al agente a evadir, sino a hacerlo moviendo el máximo dinero posible. El núcleo de este aprendizaje es la actualización de la política que se muestra a continuación:

agente_atacante.py

```python
def _actualizar_politica(self, idx_tipologia, recompensa):
    """Sube/baja la probabilidad de la tipología usada según la recompensa
    (exponentiated gradient), renormalizando con un suelo para que ninguna se
    anule. La recompensa incorpora la magnitud del importe blanqueado (Cambio 1)."""
    cfg = self.cfg
    # prob *= exp(lr * recompensa) sobre la tipología elegida; renormalizar.
    nuevos = self.politica.copy()
    nuevos[idx_tipologia] *= math.exp(cfg["TASA_APRENDIZAJE"] * recompensa)
    nuevos = nuevos / nuevos.sum()
    nuevos = np.clip(nuevos, cfg["PROB_MIN"], None)
    nuevos = nuevos / nuevos.sum()
    self.politica = nuevos
    self.pesos = nuevos.copy()  # mantiene la escala acotada (estabilidad numérica)
```

Figura 5.3. Actualización de la política del agente atacante mediante gradiente exponenciado.

Ahora bien, si el agente se limitase a aplicar la política aprendida en la fase de generación, tendería a repetir casi siempre la misma tipología (la que mejor le funciona), lo que resultaría poco realista y empobrecería la dinámica. Para evitarlo, tal y como se ha mencionado antes en este apartado, en la generación se utiliza una política de Boltzmann, es decir, una función softmax con una temperatura (TEMPERATURA_GENERACION) aplicada sobre lo aprendido, que mantiene una tipología dominante, pero hace aparecer las cinco de forma

apreciable, tal y como haría un blanqueador real que diversifica sus estrategias para no dejar un patrón único. A ello se añade también la calibración del flujo complejo (CALIBRACION_FLUJO_COMPLEJO), que garantiza que la proporción de operaciones ilícitas permanezca estable en torno al 8-10 % del total generado (elegido así para que el escenario sea lo más real posible).

Por último, este segundo script del agente atacante produce una representación visual propia del proceso mediante un grafo tridimensional interactivo exportado en formato HTML, que muestra sobre el sistema cómo se van introduciendo los flujos de transacciones. La animación muestra de forma progresiva cómo van apareciendo los distintos flujos de transacciones (en fotogramas temporales, según N_FRAMES_ANIM), de modo que puede observarse el sistema cobrando vida a medida que el atacante despliega su actividad, distinguiendo en todo momento las operaciones lícitas de las ilícitas.

5.5. Implementación del agente defensor

El tercer y último componente es el agente defensor, que lleva a la práctica la lógica de detección descrita en el apartado 4.4. A diferencia de los anteriores, se construye como una red neuronal desarrollada en PyTorch que combina dos "cerebros": una GNN, encargada de la dimensión estructural (cómo se relacionan los distintos nodos y cuentas de las personas dentro del grafo), y un Transformer ligero, encargado de la dimensión temporal (el orden y el ritmo de las transacciones). Ambos alimentan dos cabezas de salida distintas, una que evalúa a nivel de cliente (si un cliente resulta sospechoso) y otra a nivel de transacción (si una operación concreta resulta sospechosa), permitiendo al sistema detectar el blanqueo desde dos ángulos a la vez. Conviene destacar, además, que tanto la GNN como el Transformer se han implementado directamente sobre PyTorch, sin recurrir a librerías externas de grafos. De esta manera se sigue manteniendo el control total sobre el modelo evitando dependencias.

El defensor parte del grafo generado por el atacante y construye, para cada nodo, un conjunto de características observables, cuidando de no utilizar nunca la "verdad" del sistema como dato de entrada. Entre ellas se encuentran, para cualquier cuenta, su grado de conexión (tanto de operaciones entrantes como salientes, además de su total), el número de transacciones en las que participa, los importes que mueve (recibidos, enviados y el valor promedio de las operaciones de esa cuenta), el número de contrapartes distintas con las que opera, si el dinero regresa a su cuenta de origen (señal de que delata un ciclo circular), cuántas de sus operaciones caen en la "zona de pitufeo" (es decir, importes que están justo por debajo del umbral) y la velocidad con la que encadena sus movimientos. A ello se añaden, en el caso de los clientes, sus atributos propios, como el nivel de riesgo KYC, el riesgo de su nacionalidad, sus ingresos, gastos y saldo declarados, su sector de actividad o la proporción de operaciones que se vuelcan hacia entidades de alto riesgo y hacia exchanges y wallets; y, en el de las entidades, su tipo, su nivel de riesgo y si está regulada o no.

Sobre esa base actúa la GNN, que propaga la información entre nodos vecinos: en cada capa, cada cuenta combina sus propios atributos con los de las cuentas con las que se relaciona, de manera que, tras varias capas, cada nodo queda representado no solo por lo que hace, sino por el papel que ocupa dentro de la red. De igual manera, el defensor construye para cada cliente una secuencia temporal con sus operaciones ordenadas cronológicamente y la procesa mediante el Transformer, cuyo mecanismo de atención permite captar el orden y el ritmo de cada flujo

(ráfagas, encadenamientos rápidos o movimientos inmediatamente posteriores a un ingreso) que la estructura por sí sola no puede capturar.

Todo el modelo se configura desde su propio bloque centralizado. A continuación, se muestra un extracto del tercer script (agente_defensor.py) con algunos de sus parámetros más relevantes:

agente_defensor.py

```python
CONFIG_DEFENSOR = {
    # ---- Semilla ------------------------------------------------------------
    "SEMILLA": None,
    # ---- GNN (cerebro estructural) ------------------------------------------
    "GNN_DIM": 64,            # dimensión oculta de la GNN
    "GNN_CAPAS": 2,           # nº de capas de convolución
    "DROPOUT": 0.3,
    # ---- Componente temporal (Transformer ligero) ---------------------------
    "TEMPORAL_DIM": 48,       # d_model (divisible por TEMPORAL_NHEAD)
    "TEMPORAL_NHEAD": 4,
    "TEMPORAL_CAPAS": 2,
    "TEMPORAL_FF": 96,        # dim_feedforward
    "MAX_LONGITUD_SECUENCIA": 64,  # longitud máx. de la secuencia por cliente
    "TIPO_TEMPORAL": "transformer",   # "transformer" | "gru"
    # ---- Entrenamiento ------------------------------------------------------
    "LR": 0.008,
    "WEIGHT_DECAY": 5e-4,
    "EPOCAS": 250,
    "PACIENCIA": 30,          # early stopping (épocas sin mejorar val)
    "SPLIT": (0.6, 0.2, 0.2), # train / val / test
    "ESTRATIFICADO": True,    # split estratificado por es_ilicito
    "USAR_FOCAL": False,      # False -> BCE con pos_weight (desbalance)
    # ---- Detección y evaluación ---------------------------------------------
    "UMBRAL_DETECCION": 0.5,
    "UMBRAL_ADAPTATIVO": False,
    "CRITERIO_ESQUEMA": "al_menos_una",
    # ---- Visualización ------------------------------------------------------
    "COLOR_TP": "#39FF14",    # verdadero positivo (acierto): verde neón
    "COLOR_FP": "#ffb300",    # falso positivo (falsa alarma): ámbar
    "COLOR_FN": "#ff2b4e",    # falso negativo (fuga): rojo
    "COLOR_TN": "#3b82f6",    # verdadero negativo (normal): azul
    # ---- COEVOLUCIÓN adversarial de 5 rondas (megaprompt 2) -----------------
    "N_RONDAS_COEVOLUCION": 5,          # nº de rondas de la coevolución
    "COEVOLUCION_WARM_START": True,     # el defensor arranca en caliente (pesos de la ronda previa)
    "COEVOLUCION_EPOCAS_RONDA": 120,    # épocas de reentrenamiento por ronda (rondas >= 2)
    "COEVOLUCION_TASA_APRENDIZAJE_POLITICA": 0.30,
    "COEVOLUCION_ENTROPIA_POLITICA": 0.10,   # suelo de exploración (peso mínimo por tipología)
    "COEVOLUCION_SIGILO_INICIAL": 0.0,  # sigilo de la ronda 1 (0 = sin camuflaje)
    "COEVOLUCION_SIGILO_MAX": 0.85,     # tope de sigilo (nunca camufla el 100 %)
    "COEVOLUCION_SIGILO_PASO": 0.17,    # incremento de sigilo por ronda si aún se detecta mucho
    "COEVOLUCION_COSTE_SIGILO": 0.5,    # peso del coste de sigilo (penaliza el volumen perdido)
    # (...) reuso de contrapartes, twins, semilla y rutas de salida
}
```

Figura 5.4. Extracto del bloque de configuración (CONFIG_DEFENSOR) del agente defensor.

Como puede observarse, la configuración define tanto la arquitectura del modelo como su entrenamiento. La GNN se dimensiona con (GNN_DIM = 64), que fija el tamaño del "resumen" numérico que se aprende de cada nodo (a mayor dimensión, más matices puede capturar), y con el número de capas (GNN_CAPAS = 2), se establece cuántas veces se propaga la información entre nodos vecinos y, por tanto, el alcance estructural que puede captar cada cuenta dentro del grafo. A ello se añade una regularización con (DROPOUT = 0,3), una técnica que "desactiva" al azar una fracción de la red durante el entrenamiento para evitar que memorice los datos y ayudar a que generalice mejor (de esta manera se evita el sobreajuste). El componente temporal se ajusta con su dimensión (TEMPORAL_DIM), su número de cabezas de atención (TEMPORAL_NHEAD), que permiten analizar simultáneamente distintos patrones

temporales, su número de capas (TEMPORAL_CAPAS) y una longitud máxima de secuencia (MAX_LONGITUD_SECUENCIA), que limita cuántas operaciones seguidas se analizan por cliente. A su vez, el entrenamiento se controla con la tasa de aprendizaje (LR), que regula la rapidez con la que el modelo aprende; el decaimiento de pesos (WEIGHT_DECAY), que lo penaliza por volverse demasiado complejo; el número de épocas (EPOCAS), o vueltas completas a los datos; y la paciencia de la parada temprana (PACIENCIA), entre otros parámetros.

El entrenamiento es supervisado y se realiza sobre los datos sintéticos etiquetados, aprovechando que en este entorno se conoce con certeza qué operaciones son ilícitas y cuáles no. Los datos se dividen en tres conjuntos (entrenamiento, validación y prueba, según el separador SPLIT, con proporciones del 60 %, el 20 % y el 20 %), de forma estratificada respecto a lo ilícito para que la escasa proporción de fraude esté bien repartida entre ellos. Y es que aquí aparece uno de los grandes retos del problema: el desbalance existente que hay entre las operaciones, ya que existen muchísimas transacciones lícitas y muy pocas ilícitas. Para que el modelo no ignore la minoría, la función de pérdida asigna un mayor peso a las operaciones ilícitas, de modo que equivocarse en una operación de blanqueo penalice más que hacerlo en una legítima. Además, la parada temprana detiene el entrenamiento en cuanto el modelo deja de mejorar, evitando que "memorice" los datos en lugar de aprender a generalizar.

El núcleo central de este proceso es la capa de la GNN, que implementa la convolución sobre el grafo mediante la fórmula Â · H · W. Conviene detenerse en lo que representa cada componente para entender qué hace realmente esta operación. La H es la matriz que contiene la información de todos los nodos, es decir, las características de cada cuenta descritas anteriormente. La Â es la matriz de adyacencia normalizada, que recoge quién está conectado con quién dentro de la red y actúa como el "mapa" de relaciones; al multiplicarla por H, cada cuenta combina su propia información con la de las cuentas con las que se relaciona, de modo que la información se propaga entre vecinos (el hecho de que esté "normalizada" simplemente evita que las cuentas con muchísimas conexiones dominen sobre el resto). Por último, la W es la matriz de los pesos de los parámetros que el modelo va ajustando durante el entrenamiento para aprender qué combinaciones de esa información son útiles para detectar el blanqueo. En conjunto, la fórmula hace que cada nodo se represente combinando lo que él mismo hace con lo que hacen sus vecinos, teniendo en cuenta sobre todo aquello que el modelo ha aprendido que de verdad importa. Su código se muestra a continuación:

agente_defensor.py

```python
class _CapaGCN(nn.Module):
    """Convolución de grafo: H' = Â · H · W (Â ya normalizada simétricamente)."""
    def __init__(self, d_in, d_out):
        super().__init__()
        self.lin = nn.Linear(d_in, d_out)

    def forward(self, H, A_hat):
        return self.lin(torch.sparse.mm(A_hat, H) if A_hat.is_sparse else A_hat @ H)

    # (...) dentro de DefensorNet: apila varias capas GCN y propaga
    def gnn_embeddings(self, X, A_hat):
        H = X
        for i, capa in enumerate(self.gcn):
            H = capa(H, A_hat)
            if i < len(self.gcn) - 1:
                H = self.dropout(F.relu(H))
        return H  # (N, GNN_DIM)
```

Figura 5.5. Implementación de la capa de convolución de grafos (GNN) del agente defensor.

Una vez entrenado, el defensor asigna a cada operación y a cada cliente una puntuación de sospecha, y cuando esta supera un umbral establecido (UMBRAL_DETECCION = 0,5), la marca como sospechosa. El sistema incorpora también un umbral adaptativo que, en lugar de mantener un valor fijo, se recalibra sobre la validación para maximizar el F1 (la métrica que combina la precisión, o acertar en las alertas, y la sensibilidad, o no dejar escapar fraude), lo que resulta especialmente útil cuando el atacante camufla sus operaciones y las puntuaciones de lo lícito y lo ilícito son tan parecidas que cuesta distinguirlas. A nivel de esquema, se considera que uno de los 150 clientes ha sido detectado si al menos una de sus transacciones ilícitas se marca como sospechosa. La calidad de la detección se mide comparando las predicciones con la "verdad": verdaderos positivos (aciertos), falsos positivos (falsas alarmas), falsos negativos (fugas) y verdaderos negativos, que en la visualización del HTML se distinguen mediante un código de colores.

El aspecto más ambicioso de la implementación es la coevolución adversarial de cinco rondas, en la que el agente atacante y este agente defensor se enfrentan de forma sucesiva con retroalimentación mutua. En cada ronda, el atacante genera un nuevo escenario de transacciones, el defensor la entrena y detecta, y su resultado real (no el del defensor proxy que emplea el atacante en su entrenamiento) realimenta la estrategia del atacante. Para que el defensor acumule experiencia entre rondas, se emplea un arranque en caliente (COEVOLUCION_WARM_START), haciendo que conserve los pesos aprendidos en la ronda anterior en lugar de empezar de cero. Por su parte, el atacante reorienta su política hacia las tipologías que mejores resultados tienen al evadir y sube su nivel de "sigilo", camuflando lo observable de sus operaciones. Este sigilo, no obstante, tiene un coste, y es que camuflar implica mover menos dinero (COEVOLUCION_COSTE_SIGILO), estableciendo de esta manera un compromiso entre capacidad de evasión y beneficio económico manteniendo la dinámica de un terreno lo más realista posible. Los resultados numéricos de esta coevolución se presentan en el capítulo 6.

Finalmente, el defensor genera sus propias visualizaciones: un grafo tridimensional interactivo en formato HTML que representa el resultado de la detección (distinguiendo los aciertos de los errores) y un segundo HTML de coevolución que integra las distintas vistas mediante varios botones, permitiendo alternar entre el escenario base, la actividad del atacante y la detección del defensor a lo largo de las 5 rondas establecidas.

6. Resultados y discusión

6.1. Resultados de la demostración

Una vez implementado el ecosistema, este apartado presenta los resultados obtenidos tras su ejecución completa, siguiendo el mismo orden de la demostración: en primer lugar, el escenario generado por el Financial Sandbox; después, la actividad introducida por el agente atacante; y, por último, el rendimiento del agente defensor y la evolución conjunta entre ambos. Conviene mencionar desde el inicio que todos los resultados corresponden a una ejecución realizada con la semilla 88, seleccionada como ejemplo. Y es que, como ya se explicó, la semilla puede fijarse con cualquier otro valor o generarse de forma aleatoria, lo que produciría resultados diferentes en cada ejecución, aunque el comportamiento general del sistema seguiría siendo el mismo. Se ha optado por fijarla para que los resultados puedan interpretarse sobre unos datos concretos.

El primer resultado corresponde al Financial Sandbox. A partir de los dos ficheros de Excel de entrada que se deben adjuntar antes de ejecutar el código, el sistema genera un total de 215 nodos, correspondientes a NextGen Bank en el centro, los 150 clientes y las 64 entidades externas, junto con las 1.500 relaciones que vinculan a cada cliente con sus contactos asociados. En esta fase no existe todavía ninguna transacción, ya que el sandbox únicamente implementa el entorno sobre el que posteriormente intervendrá el atacante. La Figura 6.1 muestra esta estructura inicial, con el banco ocupando el centro y las distintas entidades dispuestas a su alrededor, lo que permite hacerse una idea visual del sistema antes de que comience la actividad.

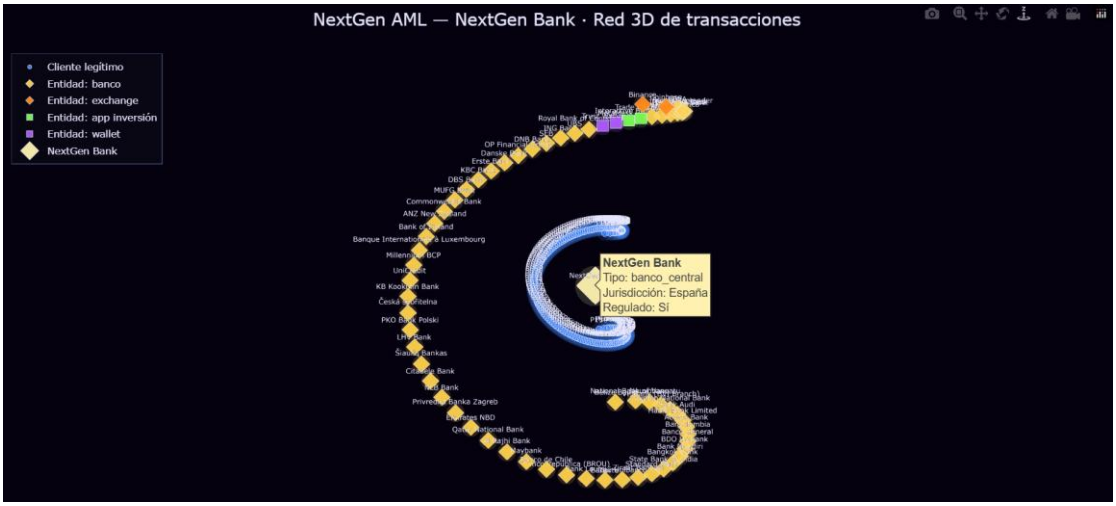

Figura 6.1. Grafo 3D del escenario base generado por el Financial Sandbox.

A partir de este escenario base, entra en juego el agente atacante, que genera todas las transacciones. En esta ejecución, el sistema mueve un volumen total de 38,25 M€ durante los tres meses simulados (de los 90 días de la ventana temporal), de los cuales 26,14 M€ (68 %) corresponden a operaciones lícitas y el resto a la actividad vinculada al blanqueo. Conviene distinguir aquí dos magnitudes diferentes para no confundir importes. Por un lado, el "ilícito recirculado" asciende a 12,11 M€ (un 32 %), pero esta cifra no significa que haya todo ese dinero ilícito, sino que es el mismo dinero contado varias veces, una por cada cuenta por la que pasa dentro de su flujo. Y es que, al saltar de cuenta en cuenta (lo que se conoce como multi-hop), un mismo importe se suma en cada paso, lo que hace que la cantidad total suba.
Por otro lado, el dinero blanqueado real, es decir, la cifra íntegra que se introduce en los esquemas sin contarlo varias veces, es de 5,89 M€. Asimismo, las operaciones ilícitas

representan aproximadamente el 10 % del total de transacciones, en la línea de lo buscado (que el fraude sea minoritario frente a la actividad legítima, tal y como ocurre en un sistema financiero real).

Respecto a lo que el atacante aprende, la Figura 6.2 muestra su entrenamiento por refuerzo. En el panel de la izquierda se observa cómo la recompensa acumulada crece progresivamente a lo largo de los 3.000 episodios, lo que indica que el agente mejora de verdad con la práctica. En el panel de la derecha se aprecia la convergencia de su política, es decir, la probabilidad que asigna a cada una de las cinco tipologías: en esta ejecución, la concentración (la tipología en la que varias cuentas vuelcan su dinero en un mismo destino) se impone como la preferida, alcanzando una probabilidad en torno al 0,85, mientras que las restantes conservan probabilidades reducidas, aunque nunca nulas, manteniendo el nivel mínimo de exploración como ya se ha mencionado. La Figura 6.3 muestra, sobre la red ya poblada, cómo el atacante despliega esos flujos de transacciones, distinguiendo las distintas operaciones.

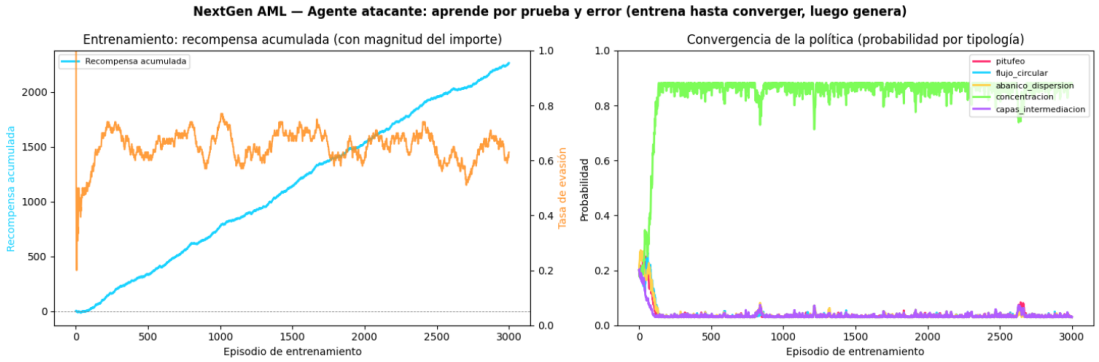

Figura 6.2. Aprendizaje del agente atacante por refuerzo.

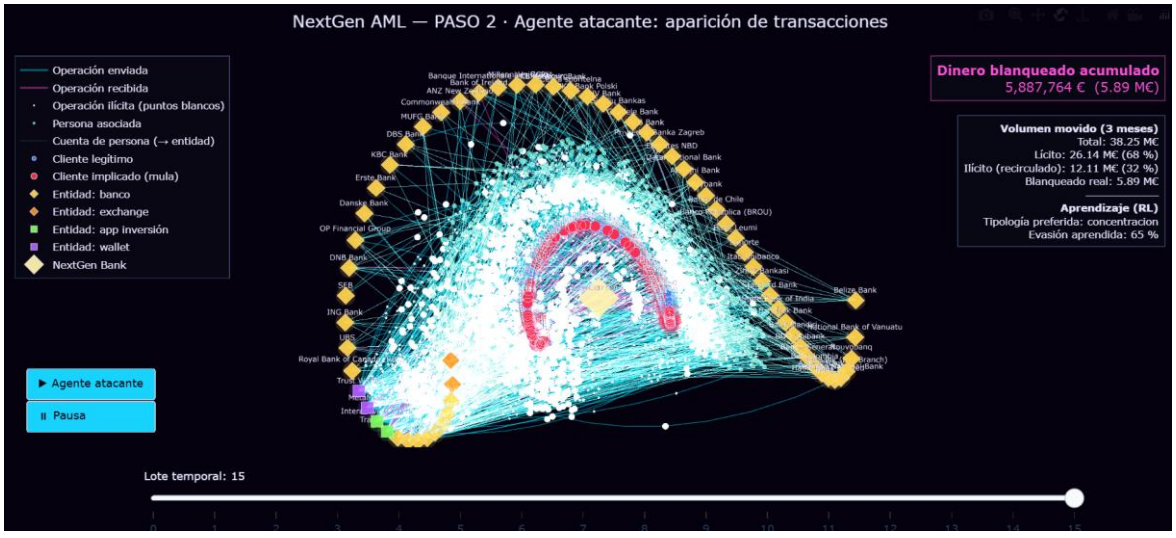

Figura 6.3. Flujos de transacciones generados por el agente atacante.

Llega entonces el turno del agente defensor, cuyo rendimiento en la primera ronda (antes de la coevolución) constituye el resultado central de la detección. Para valorarlo conviene aclarar primero qué significan las métricas empleadas. La precisión indica qué proporción de las operaciones que el defensor marca como sospechosas son realmente ilícitas (es decir, mide cuántas de sus alarmas son acertadas). El recall, o sensibilidad, indica qué proporción (%) del

fraude real logra detectar. Y el F1 es una métrica que combina ambas en un solo número, ofreciendo una visión equilibrada del rendimiento. En esta ejecución, a nivel de transacción el defensor obtiene una precisión de 0,89, un recall de 0,92 y un F1 de 0,91, cifras que se pueden calificar de altas y que indican que detecta bien y con pocas falsas alarmas. A nivel de cliente, los resultados son aún mejores (precisión de 1,00, recall de 0,92 y F1 de 0,96).

Estos valores pueden analizarse con mayor detalle en la matriz de confusión de la Figura 6.4, que clasifica las operaciones en cuatro categorías: verdaderos positivos (operaciones ilícitas que se han detectado), verdaderos negativos (operaciones lícitas correctamente ignoradas), falsos positivos (operaciones lícitas marcadas erróneamente como ilícitas, es decir, falsas alarmas) y falsos negativos (operaciones ilícitas que no son detectadas, las fugas). En concreto, refiriéndonos a las transacciones, el defensor acierta 165 operaciones ilícitas (verdaderos positivos) y 1.670 lícitas (verdaderos negativos), cometiendo únicamente 23 falsas alarmas y 13 fugas. A nivel de esquema, donde se considera que un esquema queda detectado si al menos una de sus transacciones ilícitas se marca como sospechosa, el defensor descubre 227 de los 229 existentes, una cobertura muy elevada. La Figura 6.5 representa gráficamente estos resultados mediante un código de colores que distingue los aciertos (verde), las falsas alarmas (ámbar) y las fugas (rojo), facilitando una interpretación visual del comportamiento del modelo.

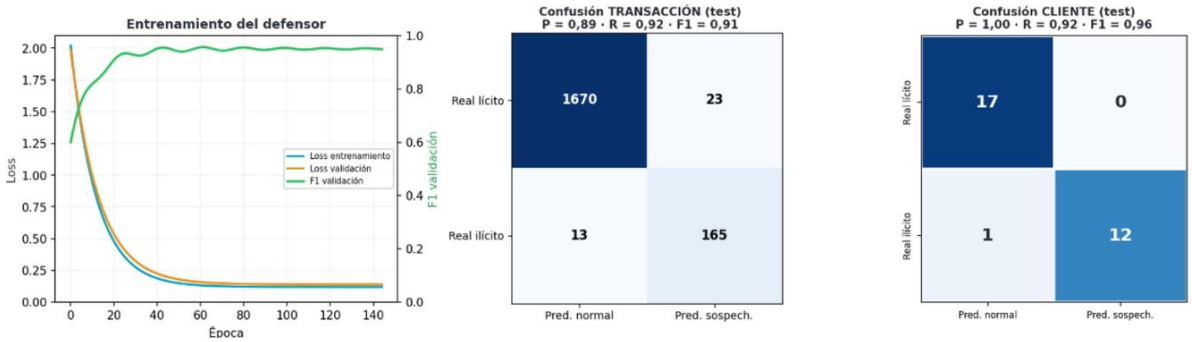

Figura 6.4. Entrenamiento y matrices de confusión del agente defensor.

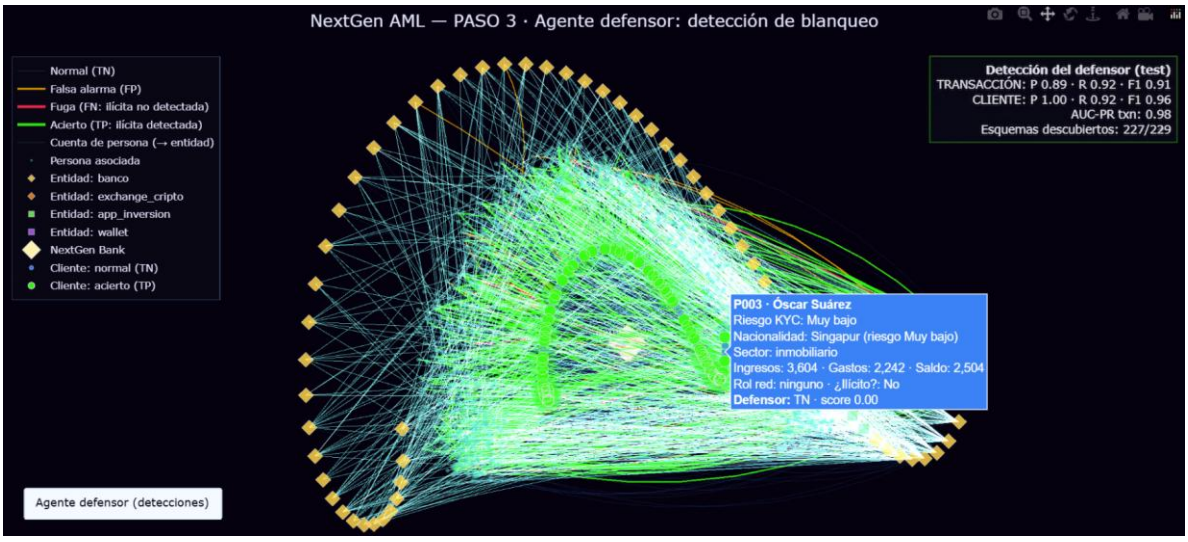

Figura 6.5. Detección del agente defensor sobre la red.

Ahora bien, el resultado más relevante de toda la dinámica es la coevolución adversarial de cinco rondas, en la que el atacante y el defensor se enfrentan de forma sucesiva mejorando el uno contra el otro. La Tabla 6.1 recoge la evolución ronda a ronda y la Figura 6.6 la representa gráficamente. Lo que se observa es que a medida que el atacante incrementa su nivel de "sigilo" (un parámetro que mide cuánto camufla sus operaciones, y que sube de 0,00 en la primera ronda a 0,68 en la quinta), la detección disminuye. En concreto, el F1 del defensor pasa de 0,943 en la primera ronda a 0,512 en la quinta, y de forma coherente la detección no alcanza el 100 % nunca, lo que resulta plenamente realista ya que un sistema que detectase absolutamente todo sería sospechoso de estar sobreajustado o de trabajar sobre un problema demasiado fácil.

Ronda	Sigilo	Recall	Precisión	F1	AUC-PR	Umbral	Esquemas (cob.)	% ilícita	Movido	Evasión
1	0.00	0.918	0.969	**0.943**	0.988	0.80	255/259 (98%)	10.2%	6.50M€	6.6%
2	0.17	0.893	0.912	**0.902**	0.957	0.85	251/256 (98%)	9.6%	4.98M€	8.9%
3	0.34	0.813	0.729	**0.769**	0.871	0.60	217/217 (100%)	8.6%	3.14M€	8.5%
4	0.51	0.926	0.578	**0.712**	0.776	0.70	200/203 (99%)	8.0%	2.65M€	7.1%
5	0.68	0.922	0.355	**0.512**	0.541	0.55	241/242 (100%)	9.0%	2.63M€	2.9%

Tabla 6.1. Resultados de la coevolución adversarial por ronda (semilla 88).

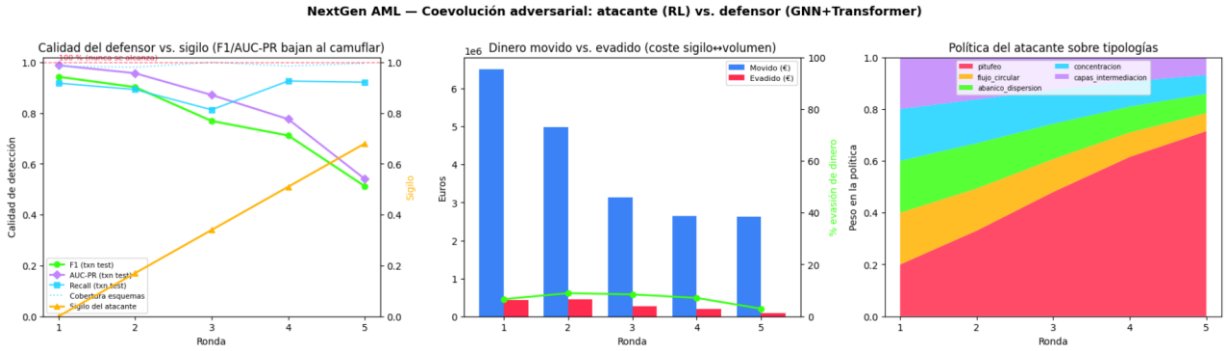

Figura 6.6. Evolución de la coevolución adversarial de las cinco rondas.

No obstante, el aspecto más interesante no es que la detección baje, sino el cómo lo hace. Al observar las cifras, se comprueba que lo que cae es la precisión (que pasa de 0,969 a 0,355), mientras que el recall se mantiene alto y estable en todas las rondas (entre 0,81 y 0,93). Dicho de otro modo, cuando el atacante camufla su actividad ilícita implementando operaciones legítimas, el defensor sigue detectando el fraude (por eso el recall aguanta más o menos igual), pero lo hace incrementando las falsas alarmas (por ello la precisión se hunde). Este comportamiento conecta directamente con uno de los grandes problemas en la detección del blanqueo de capitales, el de los falsos positivos. Lo que demuestra que el ecosistema es capaz de reproducir este mismo dilema de forma realista.

Del mismo modo, la coevolución también revela el coste que tiene el atacante para esconderse ya que como vemos en la Tabla 6.1, el volumen de dinero movido va disminuyendo conforme aumenta el sigilo (pasando de 6,50 M€ en la primera ronda a 2,63 M€ en la quinta), lo que refleja la relación entre camuflarse y mover dinero (cuanto más se esconde el atacante, menos capacidad de blanqueo tiene). A esto, se suma un cambio de estrategia porque mientras que el atacante de la primera ronda prefería la concentración, tras las cinco rondas su tipología dominante pasa a ser el pitufeo (el fraccionamiento de dinero en importes pequeños), precisamente porque ha aprendido que fragmentando los importes se evade mejor. Justo este cambio es la prueba de que la coevolución funciona y de que el atacante se adapta de verdad a su adversario.

En conjunto, estos resultados demuestran que las distintas tecnologías de este sistema realmente funcionan y, sobre todo, que interactúan entre sí de forma coherente, creando una dinámica de mejora mutua que es el corazón de la propuesta. No obstante, hay que recordar que estos son los resultados de una ejecución concreta (la SEMILLA 88) sobre datos sintéticos y simulados, y el valor real reside en mostrar que el enfoque propuesto es viable y se comporta como cabría esperar, no en obtener cifras extrapolables a un entorno real.

6.2. Ventajas frente a los sistemas tradicionales

A partir de los resultados obtenidos en la demostración, se pueden observar una serie de ventajas del enfoque de NextGen AML frente a los sistemas antiblanqueo tradicionales y frente a los sistemas de inteligencia artificial aislados. No obstante, conviene plantearlas con cautela como se matizará al final de este punto

La primera y más importante de todas es su carácter dinámico y coevolutivo. Como se sabe, los sistemas tradicionales, basados en reglas fijas, son estáticos: una vez definidas las reglas y umbrales, permanecen inmóviles hasta que un analista las vuelve a actualizar manualmente, lo que deja una ventana de tiempo en la que los criminales, que sí evolucionan, pueden adaptar sus estrategias antes de que se incorporen nuevas medidas de detección. En cambio, NextGen AML plantea un escenario en el que el agente defensor evoluciona continuamente frente a un agente atacante que modifica sus estrategias de forma progresiva, aprendiendo de esta forma a detectar estrategias cambiantes en lugar de patrones previamente conocidos. La demostración que se ha realizado muestra precisamente esta capacidad de adaptación mutua.

Como consecuencia de lo anterior, se origina una segunda ventaja: su capacidad anticipatoria. Como el atacante genera tipologías por su cuenta y busca activamente la forma más rentable de evadir la detección, el sistema puede exponer al defensor a amenazas que todavía no se han visto en el mundo real, en lugar de limitarse a reaccionar ante fraudes y métodos ya conocidos. Dicho de otro modo, permite entrenar a la defensa contra ataques del futuro, no solo contra los que se conoce hoy día.

Otro punto a favor es la combinación de la dimensión estructural y la temporal mediante la combinación de la GNN y el Transformer. Mientras que muchos enfoques se centran únicamente en el importe o en reglas sobre operaciones individuales, aquí se analiza tanto con quién se relaciona cada cuenta dentro de la red (la estructura) como el orden, el ritmo y el tiempo en que ocurren las operaciones (la temporalidad), ofreciendo una visión mucho más completa y difícil de engañar.

A ello se añade el uso de un entorno sintético, que resuelve dos de los mayores obstáculos del sector de AML: la privacidad y la disponibilidad de datos. Y es que trabajar con datos reales de clientes es legalmente complicado y, además, resulta muy difícil disponer de casos de blanqueo etiquetados con certeza. Al generar datos sintéticos con su "verdad" conocida, el ecosistema esquiva ambos problemas y permite entrenar y evaluar de forma rigurosa. Asimismo, ofrece la posibilidad de generar tantos datos como sean necesarios y de modelar de manera controlada los atributos y perfiles de los clientes, facilitando la creación de escenarios adaptados a distintos niveles de riesgo y tipologías de blanqueo.

Por último, de estas ventajas se desprenden dos beneficios prácticos relevantes. El primero es la posible reducción de los falsos positivos, uno de los principales problemas de los sistemas actuales (que generan enormes cantidades de alertas inútiles que saturan a los equipos de Compliance y a sus analistas). Al basarse en la información contextual, estructural y temporal, y no en umbrales fijos, el modelo tiene el potencial de discriminar con mayor precisión lo sospechoso de lo inocente, ganando así eficiencia. Y el segundo es que todo ello es compatible con el paradigma del Compliance-as-Code, al permitir que los mecanismos de cumplimiento normativo puedan expresarse y ejecutarse como un sistema automatizado y continuamente actualizado.

A pesar de todo, conviene ser prudente con todo esto. Estas ventajas son prometedoras, pero se han demostrado a escala de prueba de concepto y sobre datos sintéticos, de manera que deben entenderse como algo con fundamento, no como ventajas plenamente demostradas en un entorno real. Confirmarlas exigiría seguir desarrollando y validando el sistema, tal y como se explica en el capítulo 7.

6.3. Limitaciones del trabajo

En un trabajo de esta naturaleza, es imprescindible reconocer con honestidad las limitaciones de la propuesta, que delimitan su alcance real y evitan atribuirle capacidades que no tiene.

La limitación más evidente es el uso de datos sintéticos de escala reducida ya que el ecosistema no trabaja con datos bancarios reales, sino con un entorno ficticio construido alrededor de una única entidad financiera con 150 clientes, muy lejos de los millones de cuentas y participantes y de la complejidad de una entidad real. Los datos, por tanto, representan de forma simplificada la realidad, lo cual es útil para validar la prueba de concepto pero no para sacar conclusiones que sean trasladables.

En segundo lugar, el defensor-proxy que emplea el atacante durante su entrenamiento es una aproximación heurística, es decir, un detector simplificado basado en unas pocas señales, y no un modelo tan sofisticado como el defensor real. Esto es suficiente para que el atacante aprenda, pero supone una simplificación del agente defensor. Del mismo modo, las cinco tipologías de blanqueo implementadas representan únicamente una parte de las numerosas técnicas que existen en la realidad.

Asimismo, la propuesta no ha sido contrastada con datos reales ni evaluada conforme a criterios regulatorios, lo que conlleva un posible sesgo, ya que el modelo se evalúa sobre datos producidos bajo las mismas hipótesis con las que ha sido diseñado. Conviene recordar, además, que los resultados dependen de la semilla y de la configuración elegida ya que, aunque el comportamiento global del sistema permanece estable, los valores concretos de las métricas pueden variar. A ello se añade que tanto el modelado como la implementación incorporan diversas simplificaciones para que la demostración fuese viable.

Por último, conviene precisar el alcance del propio trabajo. La propuesta no está lista para su despliegue en producción a escala real ni para su comercialización, sencillamente porque no se dispone de los recursos que esto exigiría (equipos, infraestructuras de cómputo especializadas, acceso a datos reales, un proceso de validación normativo…) ni del conocimiento experto que requeriría un desarrollo profesional de esta naturaleza. De hecho, todo el proyecto se ha

construido a partir de los conocimientos y habilidades adquiridos a lo largo de este Máster, lo que marca de forma coherente sus límites.

Lejos de restar valor a la propuesta, reconocer estas limitaciones es una muestra de rigor y de saber hasta dónde llega el trabajo. El objetivo nunca fue crear un producto terminado, sino demostrar que el enfoque funciona y que vale la pena seguir desarrollándolo, algo que la demostración consigue dentro de sus propios límites. Precisamente de cómo dar ese siguiente paso, trata el capítulo siguiente.

7. Evolución futura: hacia el Compliance-as-Code

7.1. Del modelo de detección al paradigma Compliance-as-Code

Tal y como se ha visto a lo largo del trabajo, el agente defensor cumple una función concreta, pero limitada: analiza las operaciones y marca aquellas que presentan indicios de blanqueo. Sin embargo, la propuesta puede proyectarse un paso más allá, hacia el paradigma del Compliance-as-Code que ya se introdujo en el apartado 2.3.5 (según el cual las normas, políticas y procedimientos de Compliance se expresan como código ejecutable, auditable y fácilmente actualizable). Detectar lo sospechoso es solo una parte del cumplimiento ya que el objetivo último sería convertir todo el proceso de cumplimiento en algo programable y ejecutable. Esta visión se enmarca, además, dentro del ámbito de la RegTech, disciplina que persigue aplicar la tecnología para mejorar la monitorización, el reporte y el cumplimiento de las obligaciones regulatorias (Arner et al., 2017).

Bajo este enfoque, el ecosistema evolucionaría desde un sistema capaz de detectar hacia otro capaz de razonar, justificar y ejecutar decisiones de cumplimiento. Por una parte, lo que el agente defensor aprende durante la coevolución podría convertirse en reglas y modelos guardados por versiones, que reflejaran los patrones de riesgo que ha detectado. Por otro lado, las propias normas regulatorias (como los umbrales, las obligaciones legales y de reporte, los criterios de riesgo…) se expresarían igualmente como código auditable, y así, las decisiones de cumplimiento dejarían de depender de documentos que un analista debe interpretar y aplicar a mano para convertirse en controles programables, trazables y ejecutables.

Este planteamiento aporta ventajas de gran valor en el ámbito del Compliance. En primer lugar, mejora la trazabilidad y la auditabilidad, ya que cada decisión puede vincularse directamente a las reglas y modelos que la originaron, facilitando su revisión por parte de los auditores. La segunda es la automatización, que reduce el trabajo manual y evita que cada persona o departamento aplique criterios distintos. La tercera es la rapidez con la que pueden incorporarse los cambios normativos, puesto que, cuando cambia la regulación, se podría ajustar el código correspondiente para que los controles se ajustaran de forma inmediata, en lugar de depender de largos procesos de modificación. Y, en consecuencia, el sistema deja de depender tanto de que cada persona interprete las normas por su cuenta y a su manera, como ocurre hoy en día.

Conviene insistir, no obstante, en que todo lo anterior es una proyección a futuro y no algo que está implementado en este trabajo. La prueba de concepto se ha centrado en la detección adversarial, mientras que el Compliance-as-Code representa una posible línea hacia la que esa detección podría ampliarse. Se trata, por tanto, de señalar un posible camino de evolución, dejando su desarrollo real para más adelante.

7.2. Escalado hacia un sistema con datos reales

Más allá del paradigma anterior, el reto más evidente para llevar esta propuesta a la práctica consiste en pasar de una demostración sintética y controlada a un sistema capaz de operar con datos bancarios reales. Y es que una cosa es que la idea funcione en un entorno controlado, y otra muy distinta que esté lista para la realidad, algo que exige superar grandes obstáculos.

El primero y más delicado es el acceso a datos transaccionales reales. Acceder a movimientos reales de los clientes de un banco conlleva fuertes barreras de privacidad, confidencialidad y regulación, ya que se trata de información especialmente protegida cuyo uso está sujeto a estrictos límites legales. A ello se añade la enorme cantidad y variedad de esos datos, ya que un banco real maneja millones de operaciones muy distintas entre sí, muy diferente al tamaño y la uniformidad del escenario simulado. Un tercer obstáculo es que apenas hay casos reales etiquetados, y es que, a diferencia del entorno sintético (donde se sabe con certeza qué operaciones son ilícitas), en la realidad rara vez se cuenta con casos de blanqueo ya confirmados con los que entrenar y validar el modelo.

A estos retos de datos se suman otros de carácter técnico y práctico. Por una parte, la integración con los sistemas de monitorización y control que los bancos ya utilizan, que obligaría a que la solución funcionara junto a las herramientas existentes en lugar de sustituirlas. Por otra, la potencia de cálculo necesaria para procesar ese volumen de información, muy superior a la de una demostración que corre en un entorno sencillo. Y, por último, la validación con expertos y supervisores, imprescindible para que el sistema fuera aceptado y considerado fiable en un ámbito regulado.

Precisamente aquí cobra sentido la idea de que el entorno sintético podría servir como un terreno de entrenamiento seguro, es decir, un espacio en el que el sistema madura y se perfecciona frente a amenazas cada vez más sofisticadas antes de enfrentarse a datos reales. De este modo, la simulación no sería un sustituto de la realidad, sino un paso previo para prepararse de cara a ella. En cualquier caso, conviene dejar claro que este salto supondría un reto muy grande, pero, a pesar de ello, la prueba de concepto demuestra que la idea funciona y que merece la pena desarrollarla.

7.3. Líneas de trabajo futuras

Partiendo de todo lo anterior, pueden señalarse varias líneas de trabajo futuras que permitirían dar continuidad a la propuesta y acercarla poco a poco a un sistema aplicable.

En primer lugar, en el plano de los datos y las amenazas, se podrían incorporar más tipologías de blanqueo y hacerlas más complejas y variadas, de modo que el sistema se pudiese entrenar en un entorno más competitivo. En segundo lugar, desde el punto de vista de los modelos, se podrían mejorar tanto la parte de grafo como la temporal con arquitecturas más avanzadas que las usadas en la prueba de concepto, capaces de captar patrones aún más completos. Del mismo modo, convendría mejorar el propio agente atacante sustituyendo su defensor-proxy interno (el detector simplificado que le sirve como entrenamiento) por un enfrentamiento más directo y estrecho con el defensor real, reforzando así la coevolución.

Asimismo, de cara al cumplimiento, resultaría muy valioso dotar al sistema de explicabilidad, es decir, que no solo detecte, sino que además explique sus decisiones de forma comprensible, lo cual es esencial para un ámbito tan regulado como este. Otra línea bastante prometedora sería explorar el aprendizaje colaborativo entre entidades, lo que permitiría a varios bancos compartir patrones de blanqueo sin tener que compartir los datos de sus clientes, esquivando así las barreras de protección de datos. A ello se sumaría conectar el sistema con las herramientas de vigilancia que los bancos ya utilizan y, por último, avanzar hacia la puesta en práctica real del Compliance-as-Code que se ha descrito al principio de este capítulo.

En definitiva, todos estos enfoques señalan posibles oportunidades para seguir desarrollando la propuesta a partir de la prueba de concepto que se ha presentado, con el fin de acercarla a un sistema más completo y aplicable. En cualquier caso, se trata de un punto de partida sobre el que seguir trabajando, siendo conscientes de todo lo que queda por recorrer.

8. Conclusiones

A lo largo de este trabajo se ha abordado uno de los grandes problemas que hay en la lucha contra el blanqueo de capitales: el hecho de que los sistemas tradicionales, basados en reglas rígidas y de carácter reactivo, sean insuficientes frente a las amenazas que realmente ya existen. Y es que estos sistemas se limitan a reaccionar frente a patrones ya conocidos, generando además una enorme cantidad de falsas alarmas que saturan a los equipos de cumplimiento, mientras que las organizaciones criminales evolucionan constantemente yendo siempre un paso por delante. A partir de esta realidad, unida al interés personal que despertó en mí el mundo del Compliance durante mi experiencia profesional, surgió la pregunta que dio origen a este trabajo: ¿y si, en lugar de esperar a reconocer el fraude, fuéramos capaces de anticiparnos a él?

Con ese objetivo, este Trabajo Fin de Máster ha propuesto NextGen AML, un ecosistema adversarial de inteligencia artificial orientado a la detección y la anticipación del blanqueo de capitales. Dicho ecosistema está formado por tres componentes que interactúan entre sí: un agente atacante, que mediante aprendizaje por refuerzo aprende a generar y camuflar esquemas de blanqueo; un agente defensor, que combinando GNNs y Transformers detecta las operaciones sospechosas atendiendo tanto a la estructura de la red como al ritmo temporal de las operaciones; y un entorno de simulación, el Financial Sandbox, que genera el escenario sintético sobre el que ambos interactúan. La clave de la propuesta consiste en que estos dos agentes coevolucionan dentro del entorno ficticio, es decir, mejoran enfrentándose el uno al otro. Todo ello, además, se ha acompañado de una prueba de concepto, desarrollada en Google Colab, que traduce la idea a código y la pone a funcionar de principio a fin, demostrando así que la propuesta es viable.

Con todo ello puede afirmarse que se han cumplido los objetivos que se plantearon al principio del trabajo. El objetivo general, que era proponer y defender este ecosistema a nivel conceptual y respaldarlo con una prueba de concepto que lo mostrara en funcionamiento, se ha alcanzado a lo largo de los distintos capítulos. Y lo mismo ocurre con los cuatro objetivos específicos, que también se han cubierto: se ha diseñado el agente atacante y se ha mostrado cómo funciona con la demostración; se ha diseñado y construido una versión básica del agente defensor; se han analizado y justificado cada una de las ventajas de este enfoque frente a los sistemas actuales que existen contra blanqueo; y, por último, se ha planteado cómo podría evolucionar la propuesta hacia el Compliance-as-Code, junto con las principales líneas de trabajo para el futuro.

Más allá de haber cumplido estos objetivos, conviene detenerse en lo que, verdaderamente es el valor de este trabajo. En primer lugar, el cambio de enfoque que supone pasar de una detección que reacciona a una que se anticipa, y es que, al enfrentar al defensor con un atacante que inventa y mejora sus propias técnicas, el sistema puede prepararse frente a amenazas que aún no se han visto, en lugar de limitarse a perseguir las que ya se conocen. En segundo lugar, destaca el hecho de reunir en un mismo sistema tecnologías que hasta ahora se utilizaban por separado (el aprendizaje por refuerzo, las GNNs y los Transformers), y es que aquí no solo funcionan juntas, sino que se potencian mutuamente a través de la coevolución. Y, en tercer lugar, el haber demostrado de forma honesta y práctica que una idea que podría parecer solo teórica es en realidad viable y merece la pena seguir explorándola.

Ahora bien, es imprescindible ser claro y sincero sobre hasta dónde llega todo lo anterior. Este trabajo no es una solución lista para desplegarse en una entidad real, sino una propuesta conceptual acompañada de una prueba de concepto acotada, construida con datos sintéticos y a

pequeña escala. Por lo tanto, sus resultados no son cifras que puedan trasladarse tal cual al mundo real, sino la evidencia de que el enfoque funciona y se comporta como se esperaba. Precisamente por ello, su valor no está en ofrecer un producto terminado, sino en abrir un camino y en asentar unas bases sólidas sobre las que continuar con su desarrollo. Reconocerlo así no le resta mérito al trabajo, sino que forma parte del rigor con el que se ha hecho desde el principio.

Y es que vivimos en una era en la que la tecnología, y con ella la inteligencia artificial, cambia y evoluciona cada vez más rápido, y con ella también lo hacen los métodos que emplean los delincuentes para blanquear dinero. Como se ha visto, la tecnología necesaria para plantarles cara (la inteligencia artificial, los grafos, el aprendizaje por refuerzo…) ya existe, el reto está en atreverse a combinarla de otra forma y en dar el salto de una defensa que reacciona a una defensa que se adelanta. Puesto que, al final, en una lucha en la que el adversario no deja nunca de evolucionar (y ahora lo hace más deprisa que nunca), la mejor manera de no quedarse atrás es actuar antes que el adversario. En eso, y en nada más, reside la verdadera esencia de este trabajo: en la convicción de que el futuro de la lucha contra el blanqueo de capitales no estará en reaccionar más rápido, sino en anticiparse antes.

Referencias

Altman, E., Blanuša, J., von Niederhäusern, L., Egressy, B., Anghel, A., & Atasu, K. (2023). Realistic synthetic financial transactions for anti-money laundering models. En Advances in Neural Information Processing Systems (Vol. 36, pp. 29851–29874). Curran Associates. https://proceedings.neurips.cc/paper_files/paper/2023/hash/5f38404edff6f3f642d6fa5892479c42-Abstract-Datasets_and_Benchmarks.html

Arner, D. W., Barberis, J., & Buckley, R. P. (2017). FinTech, RegTech, and the reconceptualization of financial regulation. Northwestern Journal of International Law & Business, 37(3), 371–413. https://scholarlycommons.law.northwestern.edu/njilb/vol37/iss3/2/

Assefa, S. A., Dervovic, D., Mahfouz, M., Tillman, R. E., Reddy, P., & Veloso, M. (2020). Generating synthetic data in finance: Opportunities, challenges and pitfalls. En Proceedings of the First ACM International Conference on AI in Finance (ICAIF '20) (pp. 1–8). Association for Computing Machinery. https://doi.org/10.1145/3383455.3422554

Directiva (UE) 2015/2366 del Parlamento Europeo y del Consejo, de 25 de noviembre de 2015, sobre servicios de pago en el mercado interior y por la que se modifican las Directivas 2002/65/CE, 2009/110/CE y 2013/36/UE y el Reglamento (UE) n.º 1093/2010 y se deroga la Directiva 2007/64/CE. Diario Oficial de la Unión Europea, L 337, 23 de diciembre de 2015, 35–127. https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX:32015L2366

Europol. (2017). From suspicion to action: Converting financial intelligence into greater operational impact. Publications Office of the European Union. https://www.europol.europa.eu/publications-events/publications/suspicion-to-action-converting-financial-intelligence-greater-operational-impact

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 27, 2672–2680. https://arxiv.org/abs/1406.2661

Grupo de Acción Financiera Internacional (GAFI). (2012). Estándares internacionales sobre la lucha contra el lavado de activos, el financiamiento del terrorismo y de la proliferación: Las Recomendaciones del GAFI. GAFI. https://www.fatf-gafi.org/

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. Advances in Neural Information Processing Systems, 30, 1024–1034. https://arxiv.org/abs/1706.02216

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. En Proceedings of the 5th International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1609.02907

Lopez-Rojas, E. A., Elmir, A., & Axelsson, S. (2016). PaySim: A financial mobile money simulator for fraud detection. En Proceedings of the 28th European Modeling and Simulation Symposium (EMSS 2016) (pp. 249–255). https://www.msc-les.org/proceedings/emss/2016/EMSS2016_249.pdf

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533. https://doi.org/10.1038/nature14236

Mohun, J., & Roberts, A. (2020). Cracking the code: Rulemaking for humans and machines (OECD Working Papers on Public Governance N.º 42). OECD Publishing. https://doi.org/10.1787/3afe6ba5-en

Motie, S., & Raahemi, B. (2024). Financial fraud detection using graph neural networks: A systematic review. Expert Systems with Applications, 240, Artículo 122156. https://doi.org/10.1016/j.eswa.2023.122156

Oficina de las Naciones Unidas contra la Droga y el Delito (UNODC). (s.f.). Money laundering. Recuperado el 22 de junio de 2026, de https://www.unodc.org/unodc/en/money-laundering/overview.html

Organización de las Naciones Unidas. (2015). Transformar nuestro mundo: la Agenda 2030 para el Desarrollo Sostenible (Resolución A/RES/70/1). Naciones Unidas. https://sdgs.un.org/2030agenda

Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T. B., & Leiserson, C. E. (2020). EvolveGCN: Evolving graph convolutional networks for dynamic graphs. Proceedings of the AAAI Conference on Artificial Intelligence, 34(4), 5363–5370. https://arxiv.org/abs/1902.10191

Reglamento (UE) 2024/1620 del Parlamento Europeo y del Consejo, de 31 de mayo de 2024, por el que se crea la Autoridad de Lucha contra el Blanqueo de Capitales y la Financiación del Terrorismo y se modifican los Reglamentos (UE) n.º 1093/2010, (UE) n.º 1094/2010 y (UE) n.º 1095/2010. Diario Oficial de la Unión Europea, L, 19 de junio de 2024. https://eur-lex.europa.eu/eli/reg/2024/1620/oj

Reglamento (UE) 2024/1624 del Parlamento Europeo y del Consejo, de 31 de mayo de 2024, relativo a la prevención de la utilización del sistema financiero para el blanqueo de capitales o la financiación del terrorismo. Diario Oficial de la Unión Europea, L, 19 de junio de 2024. https://eur-lex.europa.eu/eli/reg/2024/1624/oj

Sachs, J. D. (2012). From millennium development goals to sustainable development goals. The Lancet, 379(9832), 2206–2211. https://doi.org/10.1016/S0140-6736(12)60685-0

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2.ª ed.). MIT Press. http://incompleteideas.net/book/the-book-2nd.html

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998–6008. https://arxiv.org/abs/1706.03762

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. En Proceedings of the 6th International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1710.10903

Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., & Leskovec, J. (2019). Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics. arXiv. https://arxiv.org/abs/1908.02591

Anexo A. Código de la prueba de concepto

Todo el código de la prueba de concepto se ha organizado en un repositorio de GitHub creado para este trabajo, que contiene los tres scripts (Financial Sandbox, agente atacante y agente defensor) listos para ejecutarse en Google Colab. Enlace al repositorio: https://github.com/aroldanmarquez-hue/SCRIPT---NextGen-AML---Alvaro-Roldan-Marquez