

Sistema de aprendizaje automático

Campo técnico

- 5 Esta invención se refiere a un módulo de historial categórico para un sistema de aprendizaje automático, particularmente para el uso en un sistema de aprendizaje automático para detectar anomalías en patrones de datos, por ejemplo, para detectar transacciones fraudulentas. Ciertos ejemplos se refieren a un sistema de aprendizaje automático para el uso en el procesamiento de transacciones en tiempo real.

10

Antecedentes de la invención

- Los pagos digitales han experimentado un crecimiento explosivo en los últimos veinte años y más de tres cuartas partes de los pagos globales utilizan algún tipo de tarjeta de pago o
- 15 billetera electrónica. Los sistemas de puntos de venta se están volviendo progresivamente digitales en lugar de basarse en efectivo. En pocas palabras, los sistemas globales de comercio ahora dependen en gran medida de plataformas de procesamiento electrónico de datos. Esto presenta muchos desafíos de ingeniería que en general quedan ocultos para un usuario común. Por ejemplo, las transacciones digitales deben completarse en tiempo
- 20 real, es decir, con un nivel mínimo de retraso en los dispositivos informáticos en el punto de compra. Las transacciones digitales también deben ser seguras y resistentes a ataques y explotación. El procesamiento de transacciones digitales también está limitado por el desarrollo histórico de los sistemas electrónicos globales de pagos. Por ejemplo, gran parte de la infraestructura todavía está configurada en torno a modelos que fueron diseñados
- 25 para arquitecturas de computador central que se utilizaban hace más de 50 años.

- A medida que aumentan las transacciones digitales, también se hacen evidentes nuevos riesgos de seguridad. Las transacciones digitales presentan nuevas oportunidades para el fraude y la actividad maliciosa. En 2015, se estimó que el 7 % de las transacciones digitales
- 30 eran fraudulentas y esa cifra solo ha aumentado con la transición de más actividad económica a Internet. Se estima que las pérdidas por fraude son cuatro veces mayores que la población mundial (por ejemplo, en dólares estadounidenses) y están aumentando.

- Las instituciones de servicios financieros están siendo sometidas a un mayor escrutinio regulatorio debido a que los métodos tradicionales de prevención del fraude como, por ejemplo, la autenticación de identidad (por ejemplo, contraseñas, biometría digital, identificación nacional y similares) han demostrado ser ineficaces para prevenir vectores
- 5 de fraude como, por ejemplo, identidades sintéticas y estafas. Estos vectores de amenaza de fraude mucho más complejos requieren muchos más análisis en un tiempo muy corto (menos de 50 ms) y a menudo se basan en un tamaño de muestra de datos mucho más pequeño para la estafa o el fraude en sí. Esto plantea importantes desafíos técnicos.
- 10 Si bien riesgos como el fraude son un problema económico para las empresas involucradas en el comercio, la implementación de sistemas técnicos para procesar transacciones es un desafío de ingeniería. Tradicionalmente, los bancos, comerciantes y emisores de tarjetas desarrollaban reglas o procedimientos “en papel” que los empleados implementaban
- 15 manualmente para marcar o bloquear ciertas transacciones. A medida que las transacciones se volvieron digitales, un enfoque para construir sistemas técnicos para procesar transacciones ha sido proporcionar a los ingenieros informáticos estos conjuntos de criterios desarrollados y pedirles que los implementen utilizando representaciones
- 20 digitales de las transacciones, es decir, convertir las reglas escritas a mano en declaraciones lógicas codificadas que puedan aplicarse a datos de transacciones electrónicas. Este enfoque tradicional ha tropezado con varios problemas a medida que los volúmenes de transacciones digitales han crecido. En primer lugar, cualquier
- 25 procesamiento aplicado debe realizarse en “tiempo real”, por ejemplo, con latencias de milisegundos. En segundo lugar, es necesario procesar muchos miles de transacciones cada segundo (por ejemplo, una “carga” común puede ser de 1000 a 2000 por segundo), y la carga puede variar inesperadamente con el tiempo (por ejemplo, el lanzamiento de un
- 30 nuevo producto o un grupo de entradas puede fácilmente incrementar un nivel de carga promedio en varios múltiplos). En tercer lugar, los sistemas de almacenamiento digital de los procesadores de transacciones y los bancos suelen estar aislados o particionados por razones de seguridad, pero las transacciones digitales a menudo involucran una red
- interconectada de sistemas comerciales. En cuarto lugar, ahora es posible realizar análisis a gran escala de los fraudes reales denunciados y de los fraudes previstos. Esto demuestra que los enfoques tradicionales para la detección del fraude son deficientes: la precisión es baja y los falsos positivos son elevados. Esto tiene entonces un efecto físico en el

procesamiento de transacciones digitales: se rechazan más compras genuinas en puntos de venta y en línea, y aquellos que intentan explotar los nuevos sistemas digitales a menudo salen airosos.

- 5 En los últimos años, se ha adoptado un enfoque más de aprendizaje automático para el procesamiento de datos de transacciones. A medida que los modelos de aprendizaje automático maduran en el ámbito académico, los ingenieros han comenzado a intentar aplicarlos al procesamiento de datos de transacciones. Sin embargo, esto vuelve a plantear problemas. Incluso si a los ingenieros se les proporciona un modelo de aprendizaje
- 10 automático académico o teórico y se les pide que lo implementen, esto no es sencillo. Por ejemplo, entran en juego los problemas de los sistemas de procesamiento de transacciones a gran escala. Los modelos de aprendizaje automático no tienen el lujo de un tiempo de inferencia ilimitado como en el laboratorio. Esto significa que simplemente no es práctico implementar ciertos modelos en un entorno de tiempo real, o que necesitan
- 15 una adaptación significativa para permitir el procesamiento en tiempo real en los niveles de volumen que experimentan los servidores del mundo real. Además, los ingenieros deben lidiar con el problema de implementar modelos de aprendizaje automático en datos que están aislados o particionados en función de la seguridad del acceso y en situaciones donde la velocidad de las actualizaciones de datos es extrema. Los problemas que
- 20 enfrentan los ingenieros que construyen sistemas de procesamiento de transacciones pueden entonces considerarse similares a los que enfrentan los ingenieros de redes o bases de datos; es necesario aplicar modelos de aprendizaje automático, pero cumpliendo con las limitaciones de rendimiento del sistema y tiempo de respuesta de consultas establecidas por la infraestructura de procesamiento. No existen soluciones fáciles para
- 25 estos problemas. En efecto, el hecho de que muchos sistemas de procesamiento de transacciones sean confidenciales, propietarios y basados en tecnologías antiguas, significa que los ingenieros no tienen el conjunto de conocimientos desarrollado en estos campos vecinos y a menudo enfrentan desafíos que son exclusivos del campo del procesamiento de transacciones. Además, el campo del aprendizaje automático práctico a gran escala aún
- 30 es joven y existen pocos patrones de diseño establecidos o libros de texto en los que los ingenieros puedan confiar.

Los sistemas de detección de fraude generan una puntuación de riesgo para las nuevas transacciones en tiempo real (es decir, latencia). ≤ 100 milisegundos), donde los puntajes de riesgo altos indican una probabilidad de fraude, error o uso no autorizado de una cuenta. Las instituciones financieras utilizan estos puntajes dentro de su lógica de decisión al momento de decidir si aprueban las transacciones (normalmente esto será para rechazar la transacción si el puntaje de riesgo es más alto que cierto umbral). Las instituciones financieras generalmente desean que el conjunto de transacciones por encima de un cierto umbral de riesgo contenga la mayor proporción posible de fraude (es decir, para proporcionar la “tasa de detección máxima”) con la tasa mínima posible de clasificaciones de falsos positivos (es decir, para proporcionar la “máxima precisión”).

Un enfoque conocido en la materia *per se*, implica el uso de técnicas de aprendizaje automático como, por ejemplo, redes neuronales recurrentes (RNN), para detectar anomalías en patrones de comportamiento. Los expertos en la materia observarán que la salida de una RNN es un estado siguiente, y este estado siguiente depende del estado anterior y de la entrada proporcionada a la RNN. De esta manera, el estado de la RNN se basa no sólo en la información instantánea relativa a una transacción dada, sino también en el historial de las transacciones anteriores, conservando así una memoria de los patrones que se han observado. Esto puede usarse para detectar un comportamiento anómalo que pueda ser indicativo de una transacción fraudulenta. Los términos “comportamiento” y “conductual” se utilizan en el presente documento para referirse a un patrón de actividad o acciones.

Esta detección de actividad anómala puede proporcionar mejoras a la seguridad del sistema, en comparación con los métodos “tradicionales” de prevención de fraude (por ejemplo, contraseñas, biometría digital, cédula de identidad nacional, etc.). Por supuesto, se observará que las técnicas de análisis de patrones de comportamiento pueden utilizarse en combinación con uno o más de estos métodos “tradicionales”, según corresponda.

En general, un sistema de aprendizaje automático determinado puede configurarse para gestionar transacciones relacionadas con muchas entidades diferentes, por ejemplo, diferentes titulares de tarjetas. El patrón de comportamiento de cada entidad puede ser diferente y, por lo general, el historial de cada una debe conservarse de forma

independiente para poder detectar un comportamiento anómalo para esa entidad. Como tal, los sistemas de aprendizaje automático para canales de procesamiento de transacciones de gran volumen generalmente pueden almacenar datos de estado de cada entidad (es decir, el titular de la tarjeta).

5

El Solicitante ha observado que el tiempo que transcurre entre transacciones consecutivas para una entidad determinada puede variar, a menudo de manera significativa. Por ende, el grado de influencia que el estado anterior debe tener a la hora de determinar la legitimidad de una nueva transacción. Las solicitudes de patente anteriores del Solicitante, publicadas como WO/2022/008130 y WO/2022/008131, cada una de las cuales se incorpora en el presente documento como referencia, describen una disposición en la que los datos de estado de una entidad determinada se deterioran en función del período de tiempo que ha transcurrido desde que se almacenaron esos datos de estado (es decir, desde la última transacción para esa entidad).

10

15

Los científicos de datos que crean modelos de aprendizaje automático generalmente realizan un gran esfuerzo para diseñar modelos y extraer características de aprendizaje automático para esos modelos, que produzcan los mejores equilibrios entre tasa de detección y precisión.

20

Sin embargo, el solicitante ha observado que se pueden realizar mejoras adicionales en los sistemas técnicos que realizan los análisis pertinentes. En particular, el Solicitante ha observado que la novedad de las interacciones de una entidad, por ejemplo, la novedad de una relación entre un titular de cuenta y un comerciante en particular es útil para detectar un comportamiento anómalo. La novedad de la interacción comprador-vendedor es una señal importante para maximizar el rendimiento de estas puntuaciones. Las relaciones establecidas son buenos indicadores de que la nueva actividad es genuina (es decir, realizada por el comprador de buena fe), mientras que las relaciones nuevas indican una mayor sospecha sobre la nueva actividad (que normalmente puede tomarse junto con otras señales de riesgo).

25

30

La presente invención busca aportar mejoras al sistema de aprendizaje automático para extraer automáticamente la novedad de las interacciones, por ejemplo, entre comprador y vendedor.

5 Breve descripción de la invención

De acuerdo con un primer aspecto, las modalidades de la presente invención proporcionan un sistema de aprendizaje automático para procesar datos correspondientes a una transacción entrante asociada con una entidad, comprendiendo el sistema de aprendizaje automático un módulo de historial categórico configurado para procesar al menos algunos de dichos datos. El módulo de historial categórico comprende:

- i) una memoria configurada para almacenar datos de estado para una pluralidad de categorías indexadas por un identificador de categoría respectivo, en la que los datos de estado almacenados en la memoria para cada identificador de categoría corresponden a una transacción anterior asociada con la entidad y la categoría respectiva;
- ii) una etapa lógica de deterioro configurada para modificar datos de estado almacenados en la memoria, en función de una diferencia de tiempo entre un momento de la transacción entrante y un momento de una transacción anterior, estando configurada dicha etapa lógica de deterioro para:
 - a) recuperar dichos datos de estado y aplicar una función de deterioro a los datos de estado para generar una versión deteriorada de los datos de estado, cuando la memoria contiene datos de estado para el par identificador de entidad y categoría asociado con la transacción entrante, en la que la función de deterioro depende de la diferencia de tiempo, y generar la versión deteriorada de los datos de estado; y
 - iii) una etapa lógica de actualización configurada para: actualizar la salida de datos de estado de la etapa lógica de deterioro utilizando un tensor de entrada, generado a partir de los datos correspondientes a la transacción entrante para generar datos de estado actualizados; para almacenar dichos datos de estado actualizados en la memoria para el par identificador de entidad y categoría asociado con la transacción entrante; y para generar un tensor de salida que comprenda los datos de estado actualizados;
 - iv) en el que el sistema de aprendizaje automático está configurado para asignar el tensor de salida del módulo de historial categórico a un valor escalar

representativo de una probabilidad de que la transacción entrante presente una anomalía dentro de una secuencia de acciones; y

v) en el que el valor escalar se utiliza para determinar si se aprueba o rechaza la transacción entrante.

5

El primer aspecto de la invención también se extiende a un método de procesamiento de datos asociados con una transacción propuesta asociada con una entidad. El método comprende:

i) almacenar, en una memoria, datos de estado para una pluralidad de categorías indexadas por un identificador de categoría respectivo, en el que los datos de estado almacenados en la memoria para cada identificador de categoría corresponden a una transacción anterior asociada con la entidad y la categoría respectiva;

ii) modificar datos de estado almacenados en la memoria, en función de una diferencia de tiempo entre un momento de la transacción entrante y un momento de una transacción anterior, en el que el paso de modificar los datos de estado comprende:

a) recuperar dichos datos de estado y aplicar una función de deterioro a los datos de estado para generar una versión deteriorada de los datos de estado, cuando la memoria contiene datos de estado para el par identificador de entidad y categoría asociado con la transacción entrante, en el que la función de deterioro depende de la diferencia de tiempo, y generar la versión deteriorada de los datos de estado; y

iii) actualizar la salida de datos de estado utilizando un tensor de entrada, generado a partir de los datos correspondientes a la transacción entrante para generar datos de estado actualizados; almacenar dichos datos de estado actualizados en la memoria para el par identificador de entidad y categoría asociado con la transacción entrante; y generar un tensor de salida que comprende los datos de estado actualizados;

iv) asignar el tensor de salida a un valor escalar representativo de una probabilidad de que la transacción entrante presente una anomalía dentro de una secuencia de acciones; y

v) utilizar el valor escalar para determinar si se aprueba o rechaza la transacción entrante.

El primer aspecto de la invención se extiende además a un medio legible por computadora no transitorio, que comprende instrucciones que, cuando son ejecutadas por un procesador, hacen que el procesador lleve a cabo un método de procesamiento de datos asociado con una transacción propuesta asociada con una entidad. El método comprende:

- 5 i) almacenar, en una memoria, datos de estado para una pluralidad de categorías indexadas por un identificador de categoría respectivo, en el que los datos de estado almacenados en la memoria para cada identificador de categoría corresponden a una transacción anterior asociada con la entidad y la categoría respectiva;
- ii) modificar datos de estado almacenados en la memoria, en función de una
10 diferencia de tiempo entre un momento de la transacción entrante y un momento de una transacción anterior, en el que el paso de modificar los datos de estado comprende:
 - a) recuperar dichos datos de estado y aplicar una función de
 deterioro a los datos de estado para generar una versión deteriorada de los datos
 de estado, cuando la memoria contiene datos de estado para el par identificador
15 de entidad y categoría asociado con la transacción entrante, en el que la función de deterioro depende de la diferencia de tiempo, y generar la versión deteriorada de los datos de estado; y
 - iii) actualizar la salida de datos de estado utilizando un tensor de entrada,
 generado a partir de los datos correspondientes a la transacción entrante para generar
20 datos de estado actualizados; almacenar dichos datos de estado actualizados en la memoria para el par identificador de entidad y categoría asociado con la transacción entrante; y generar un tensor de salida que comprende los datos de estado actualizados;
 - iv) asignar el tensor de salida a un valor escalar representativo de una probabilidad de que la transacción entrante presente una anomalía dentro de una
25 secuencia de acciones; y
 - v) utilizar el valor escalar para determinar si se aprueba o rechaza la transacción entrante.

El primer aspecto de la invención se extiende además a un producto de software
30 informático, que comprende instrucciones que, cuando son ejecutadas por un procesador, hacen que el procesador lleve a cabo un método de procesamiento de datos asociado con una transacción propuesta asociada con una entidad. El método comprende:

- i) almacenar, en una memoria, datos de estado para una pluralidad de categorías indexadas por un identificador de categoría respectivo, en el que los datos de estado almacenados en la memoria para cada identificador de categoría corresponden a una transacción anterior asociada con la entidad y la categoría respectiva;
- 5 ii) modificar datos de estado almacenados en la memoria, en función de una diferencia de tiempo entre un momento de la transacción entrante y un momento de una transacción anterior, en el que el paso de modificar los datos de estado comprende:
- a) recuperar dichos datos de estado y aplicar una función de deterioro a los datos de estado para generar una versión deteriorada de los datos
- 10 de estado, cuando la memoria contiene datos de estado para el par identificador de entidad y categoría asociado con la transacción entrante, en el que la función de deterioro depende de la diferencia de tiempo, y generar la versión deteriorada de los datos de estado; y
- iii) actualizar la salida de datos de estado utilizando un tensor de entrada,
- 15 generado a partir de los datos correspondientes a la transacción entrante para generar datos de estado actualizados; almacenar dichos datos de estado actualizados en la memoria para el par identificador de entidad y categoría asociado con la transacción entrante; y generar un tensor de salida que comprende los datos de estado actualizados;
- iv) asignar el tensor de salida a un valor escalar representativo de una
- 20 probabilidad de que la transacción entrante presente una anomalía dentro de una secuencia de acciones; y
- v) utilizar el valor escalar para determinar si se aprueba o rechaza la transacción entrante.
- 25 Por consiguiente, se observará que las modalidades de la presente invención proporcionan una disposición mejorada, en la que los datos de estado relacionados con transacciones anteriores de una categoría coincidente se almacenan en la memoria, y los datos de estado almacenados para una categoría dada se recuperan la próxima vez que haya una transacción que coincida con esa categoría para esa misma entidad, es decir, para el par
- 30 identificador de entidad y categoría asociado con la nueva transacción entrante. Las categorías que se pueden utilizar se explican con más detalle a continuación, sin embargo, en general, las categorías pueden incluir, por ejemplo, el comerciante, el tipo de comerciante, un monto de transacción, una hora del día o alguna combinación de estos.

Por lo tanto, el módulo de historial categórico proporciona un mecanismo para rastrear patrones en transacciones de una categoría particular. A diferencia de los mecanismos convencionales como, por ejemplo, una celda de deterioro temporal que generalmente recupera el estado de la última transacción (es decir, la transacción inmediatamente anterior para esa entidad), el módulo de historial categórico recupera selectivamente datos de estado correspondientes a la misma categoría que la transacción actual, que puede no ser la transacción más reciente, y puede haber transcurrido un período de tiempo significativo desde que se procesó la última transacción de esa categoría y, mientras tanto, puede haber ocurrido cualquier número de transacciones intermedias correspondientes a otras categorías. Esto permite extraer señales que brindan información sobre la novedad de las interacciones entre una entidad y una categoría particular, cuando dicha novedad (o patrones de comportamiento relacionados con esa novedad) pueden ser indicativos de un comportamiento anómalo (por ejemplo, fraude).

Ventajosamente, las modalidades de la presente invención pueden proporcionar mejoras significativas en la velocidad y/o confidencialidad de la implementación. En concreto, los científicos de datos ya no necesitan diseñar manualmente características de este tipo, lo que significa que no es necesario divulgarlas para la gestión del modelo como entradas al clasificador, y el tiempo de implementación de los modelos puede reducirse.

Las modalidades de la presente invención también pueden proporcionar de forma ventajosa mejoras en el rendimiento de detección. El módulo de historial categórico puede aprender una lógica de interacción más sutil que la que los seres humanos son capaces de diseñar a mano en un tiempo finito. Específicamente, el módulo de historial categórico puede aprender que solo la actividad de un tipo particular debe calificar como generadora de confianza para puntuar la actividad del tipo actual. El solicitante ha descubierto que el uso de la presente invención normalmente se traduce en mejoras significativas en la detección en relación con los enfoques que pueden conocerse en la materia *per se*.

Las modalidades de la presente invención se pueden aplicar a una amplia variedad de transacciones digitales, incluyendo, entre otros, pagos con tarjeta, transferencias denominadas “telegráficas”, pagos entre pares, pagos mediante el Sistema de

Compensación Automatizado de Banqueros (BACS) y pagos mediante la Cámara de Compensación Automatizada (ACH). Los resultados del sistema de aprendizaje automático se pueden utilizar para prevenir una amplia variedad de comportamientos fraudulentos y delictivos como, por ejemplo, fraude con tarjetas, fraude con aplicaciones, fraude con pagos, fraude comercial, fraude con juegos y lavado de dinero.

Como se describió anteriormente, el módulo de historial categórico recupera datos de estado correspondientes a la misma categoría que la transacción entrante. En general, la memoria puede almacenar datos de estado para cada categoría posible (por ejemplo, para cada comerciante o cada tipo de comerciante), aunque en la práctica este puede no ser el caso. Por ejemplo, pueden imponerse limitaciones al número de categorías para las que se pueden almacenar datos de estado en la memoria. Esto puede deberse a una limitación de hardware o software para una implementación determinada, o puede ser una decisión de diseño para limitar la cantidad de datos de estado retenidos a una cantidad particular. Por lo tanto, es posible que una transacción entrante corresponda a una categoría para la cual no existen datos de estado para el identificador de categoría correspondiente almacenado en la memoria en ese momento. En algunas modalidades, la etapa lógica de deterioro está configurada además para:

generar nuevos datos de estado y emitir dichos nuevos datos de estado, cuando la memoria no contiene datos de estado para el par identificador de entidad y categoría asociado con la transacción entrante. Los nuevos datos de estado pueden, por ejemplo, comprender típicamente un tensor cero; sin embargo, se observará que se podría utilizar algún otro tensor predeterminado para los nuevos datos de estado si fuera apropiado.

En general, los datos de la transacción entrante se pueden convertir en un tensor de entrada adecuado para su entrada al resto del sistema de aprendizaje automático. Los expertos en la materia observarán que el tensor de entrada puede verse como un tensor o vector de características, es decir, un tensor de valores cada uno de los cuales corresponde al grado en que la transacción entrante tiene ciertas características. En algunas modalidades, el sistema de aprendizaje automático comprende además una primera etapa de red neuronal configurada para aplicar una capa de red neuronal, que tiene una pluralidad respectiva de pesos aprendidos a partir de los datos correspondientes a la transacción entrante para generar el tensor de entrada, y para proporcionar dicho tensor

de entrada a la etapa lógica de actualización del módulo de historial categórico. En otras palabras, la primera red neuronal puede describirse como una capa de inserción. Esta primera red neuronal puede, en algunas modalidades, comprender un primer perceptrón multicapa (MLP).

5

De manera similar, el tensor de salida producido por la etapa lógica de actualización se asigna al valor escalar, como se describe anteriormente. En algunas modalidades, el sistema de aprendizaje automático comprende además una segunda etapa de red neuronal configurada para aplicar una segunda capa de red neuronal que tiene una pluralidad respectiva de pesos aprendidos al tensor de salida generado por la etapa lógica de actualización del módulo de historial categórico, para generar el valor escalar representativo de la probabilidad de que la transacción entrante presente una anomalía dentro de una secuencia de acciones. En algunas modalidades, esta segunda red neuronal puede comprender un segundo perceptrón multicapa (MLP).

15

Los expertos en la materia observarán que una MLP es un tipo de red neuronal de propagación hacia adelante, que generalmente se construye a partir de al menos una capa de entrada, una capa oculta y una capa de salida, en la que cada capa comprende una cantidad de nodos. Los nodos de la capa oculta y de salida tienen una función de activación (por ejemplo, una función de activación no lineal). Los nodos de cada capa están conectados a los nodos de la capa posterior (generalmente, cada nodo de una capa se conecta a cada nodo de la capa siguiente) con un peso determinado. Estos pesos pueden aprenderse durante una fase de entrenamiento, por ejemplo, durante el aprendizaje supervisado como, por ejemplo, mediante retropropagación.

25

En algunas modalidades, la etapa lógica de actualización comprende: una tercera etapa de red neuronal configurada para generar los datos de estado actualizados, utilizando los datos de estado deteriorados y el tensor de entrada derivado de la transacción entrante. En otras palabras, los datos de estado actualizados también pueden ser generados por una capa de red neuronal. Esta capa de red neuronal también puede utilizar pesos entrenados. La tercera etapa de la red neuronal puede, en algunas modalidades, comprender una red neuronal recurrente.

30

La cantidad de tiempo que transcurre entre transacciones consecutivas para una entidad determinada (por ejemplo, el titular de una tarjeta) que pertenece a una categoría particular puede variar significativamente. Por ejemplo, si las categorías incluyen la identidad de un comerciante y se almacenan datos estatales para cada comerciante (es decir, los datos estatales de la interacción anterior entre comprador y vendedor que involucró al titular de la tarjeta y al comerciante), es posible que la tarjeta se use varias veces por semana para un comerciante, mientras que esa misma tarjeta puede pasar semanas o meses entre transacciones para otro comerciante. De manera similar, diferentes entidades pueden tener transacciones a ritmos muy diferentes y los patrones de comportamiento de una entidad determinada pueden variar con el tiempo.

Se ha observado que la cantidad de tiempo transcurrido desde que se almacenaron los datos de estado relevantes, recuperados de la memoria para la combinación particular de entidad y categoría de una transacción entrante, puede tener un impacto significativo en su relevancia al momento de tomar una determinación en cuanto a la legitimidad de la nueva transacción entrante. La etapa lógica de deterioro modifica los datos de estado para reducir su influencia.

La función de deterioro se utiliza para disminuir el impacto de los datos de estados anteriores, dependiendo de su antigüedad. En un conjunto de modalidades, la etapa lógica de deterioro está configurada para modificar los datos de estado almacenados en la memoria, para la entidad asociada con la transacción entrante, en función de una diferencia de tiempo entre el tiempo de la transacción entrante y el tiempo de una transacción más reciente para esa entidad. En otras palabras, en dichas modalidades, toda la tabla de datos de estado de una entidad se deteriora cada vez que hay una transacción para esa entidad. Ventajosamente, este enfoque no requiere ningún registro de cuándo se vio por última vez cada categoría para esa entidad, sino que se utiliza una única marca de tiempo para cada entidad, donde esa marca de tiempo corresponde a la última transacción para esa entidad (independientemente de la categoría a la que se relacione esa transacción). En algunas de estas modalidades, la memoria está configurada para almacenar una marca de tiempo para los datos de estado almacenados en dicha memoria para cada entidad. El deterioro de la tabla de estados para la entidad específica puede reducir ventajosamente la latencia (en comparación con el deterioro de la tabla para todas

las entidades) al tiempo que elimina la necesidad de almacenar marcas de tiempo para cada par entidad-categoría (como en el caso en el que solo se deterioran los datos de estado para el par entidad-categoría específico, como se describe con respecto a un conjunto alternativo de modalidades a continuación).

5

En un conjunto alternativo de modalidades, la etapa lógica de deterioro está configurada para modificar todos los datos de estado almacenados en la memoria para cada entidad, en función de una diferencia de tiempo entre el momento de la transacción entrante y el momento de una transacción inmediatamente anterior. En otras palabras, en tales

10 modalidades, la tabla de datos de estado completa para todas las entidades se deteriora cada vez que hay una transacción, independientemente de para qué entidad fue esa transacción, utilizando una marca de tiempo asociada con la última transacción (independientemente de quién involucró esa transacción). Se observará, sin embargo, que esto puede imponer requisitos computacionales significativos para mitigar aumentos no
15 deseados en la latencia.

En un conjunto alternativo adicional de modalidades, la etapa lógica de deterioro está configurada para modificar únicamente los datos de estado almacenados en la memoria para el par identificador de entidad y categoría asociado con la transacción entrante, en el
20 que la etapa lógica de deterioro modifica dichos datos de estado en función de una diferencia de tiempo entre el tiempo de la transacción entrante y un tiempo de la transacción anterior, para el par identificador de entidad y categoría asociado con la transacción entrante. Con este arreglo, solo se deterioran los datos de estado de la combinación particular de entidad y categoría, cada vez que exista una transacción para
25 esa misma combinación. En algunas de estas modalidades, la memoria está configurada para almacenar una marca de tiempo para cada dato de estado almacenado en dicha memoria. Las marcas de tiempo pueden formar parte de los datos estatales o pueden almacenarse por separado. Ventajosamente, este enfoque no requiere deteriorar toda la tabla en cada transacción, lo que puede permitir un procesamiento más rápido a cambio
30 de un mayor uso de memoria para almacenar las marcas de tiempo.

Existen varias opciones para implementar el almacenamiento de los datos de estado en la memoria. Sin embargo, en algunas modalidades, la memoria está configurada como una

pila, de manera que los datos de estado recuperados se eliminan de la pila y los datos de estado actualizados se almacenan en la parte superior de la pila. De acuerdo con dichas modalidades, las interacciones entidad-categoría más recientes se pueden encontrar en la parte superior de la pila, y las interacciones menos recientes en la parte inferior de la pila.

5

En algunas modalidades, el módulo de historial categórico está configurado de manera que, cuando la memoria está llena, los datos de estado almacenados en relación con al menos un identificador de categoría se eliminan. En algunas de estas modalidades, el módulo de historial categórico está configurado de manera que, cuando la memoria está

10 llena, los datos de estado almacenados en relación con un identificador de categoría visto menos recientemente se eliminan. Con estos arreglos, los datos más antiguos se “descartan” de la memoria para hacer espacio para datos nuevos. En efecto, esto proporciona una función de “olvido” adicional en la que, asumiendo una selección apropiada del tamaño de memoria, se eliminan los datos que son demasiado antiguos. En

15 general, el tamaño de la memoria puede seleccionarse para garantizar que los datos que sería deseable conservar no se borren rutinariamente de la memoria.

En algunas modalidades, los datos de estado almacenados para cada par identificador de entidad y categoría comprenden un tensor de valores. Cada valor del tensor puede

20 corresponder a alguna característica o combinación de características que puedan tener las transacciones, y la magnitud del valor proporciona una medida de con qué fuerza la transacción (o el patrón de transacciones anteriores) exhibe esas características.

En algunas de estas modalidades, la función de deterioro aplica un multiplicador de deterioro respectivo (por ejemplo, un deterioro exponencial) a cada uno de los valores. Se

25 observará que se pueden aplicar diferentes multiplicadores (por ejemplo, diferentes deterioros exponenciales) a cada valor. El uso de diferentes multiplicadores de deterioro para diferentes valores permite que la fuerza de la memoria se establezca de manera diferente para diferentes características. Puede ser deseable que el historial de ciertas

30 características se conserve durante períodos de tiempo más largos que el de otras. Por ejemplo, los datos estatales pueden relacionarse con (es decir, depender de) un monto de transacción promedio para las transacciones del par identificador de entidad y categoría, y puede ser útil conservar la influencia de este promedio durante meses a la vez (por

ejemplo, 6 o 12 meses). Mientras tanto, los datos estatales que dependen de la velocidad de las transacciones (es decir, la tasa de transacciones por unidad de tiempo) pueden ser de interés durante un período de tiempo más corto (por ejemplo, en el transcurso de 1 o 2 horas), por lo que puede ser preferible limitar su influencia más allá de dicho período. Se observará que, en la práctica, los valores almacenados en los datos de estado pueden no tener una correspondencia directa (es decir, uno a uno) con ninguna característica o características específicas, sino que cada uno de ellos es una función de una o más características. En otras palabras, los valores numéricos producidos (por ejemplo, a través de una capa de red neuronal) se almacenan como datos de estado, y cada valor numérico puede tener una dependencia compleja de varias características. La relación entre esas características y el valor de los datos de estado correspondientes (es decir, el valor numérico) atribuido se puede aprender durante el entrenamiento. Por lo tanto, debe entenderse que no existe necesariamente un deterioro especificado por el usuario de los datos de estado relacionados con alguna característica específica, sino que más bien el sistema de aprendizaje automático puede aprender que los datos de estado históricos derivados de ciertas características (como los ejemplos descritos anteriormente) o combinaciones de dichas características, deben conservarse por más tiempo que otros.

Los multiplicadores del deterioro se pueden seleccionar según sea necesario para adaptar el grado de retención de memoria para cada característica, según corresponda. Estos multiplicadores del deterioro pueden, al menos en algunas modalidades, estar predeterminados, es decir, se pueden establecer de antemano. De acuerdo con algunas de estas modalidades, el sistema de aprendizaje automático puede aprender que los valores derivados de ciertas características o combinaciones de características deben estar sujetos a uno o más de estos factores de deterioro en diferentes grados, es decir, los factores de deterioro influyen en un proceso de entrenamiento de extremo a extremo.

Sin embargo, en algunas modalidades, se pueden aprender al menos algunos (y potencialmente todos) de los multiplicadores de deterioro. En lugar de proporcionar una selección de factores de deterioro por adelantado, el sistema de aprendizaje automático puede aprender por sí mismo los mejores factores de deterioro a utilizar y para qué datos de estado se deben utilizar esos factores de deterioro aprendidos.

- Generalmente, los multiplicadores de deterioro se pueden organizar en un tensor (o vector, etc.) de manera que los deterioros de tiempo se puedan multiplicar elemento por elemento con los datos de estado que se van a deteriorar. Cuando los datos de estado categórico se almacenan en una tabla donde cada columna puede indexarse por el
- 5 identificador de categoría, el tensor (o vector, etc.) de deterioro en el tiempo se puede multiplicar elemento por elemento con cada columna de la tabla. Por supuesto, se observará que la tabla podría, de manera equivalente, configurarse de manera que los datos de estado se almacenen en filas indexadas por el identificador de categoría.
- 10 Como se describió anteriormente, el módulo de historial categórico permite conservar patrones en todas las transacciones para pares de entidades y categorías específicas. En algunas modalidades, el sistema de aprendizaje automático comprende además un módulo de celda de deterioro temporal, configurado para procesar al menos algunos de los datos correspondientes a la transacción entrante. El módulo de celda de deterioro
- 15 temporal comprende:
- una segunda memoria configurada para almacenar datos de segundo estado correspondientes a una transacción inmediatamente anterior; y
 - una segunda etapa lógica de deterioro configurada para modificar los datos del segundo estado almacenados en la segunda memoria, en función de una diferencia de
- 20 tiempo entre un momento de la transacción entrante y un momento de la transacción inmediatamente anterior. Los expertos en la materia observarán que, a diferencia del módulo de historial categórico, el módulo de celda de deterioro temporal actúa transacción por transacción para una entidad particular, independientemente de la categoría a la que se relacionen esas transacciones.
- 25 De acuerdo con dichas modalidades, el módulo de historial categórico y el módulo de celda de deterioro temporal pueden disponerse en paralelo entre sí y proporcionar un análisis complementario de la transacción entrante. El módulo de historial categórico ayuda a detectar anomalías (por ejemplo, fraude) en función de la novedad de las interacciones
- 30 entre la entidad (por ejemplo, titular de la tarjeta) y la categoría (por ejemplo, comerciante, tipo de comerciante, hora del día, monto de la transacción, etc.), mientras que el módulo de celda de deterioro en el tiempo ayuda a detectar anomalías en función del comportamiento de transacciones recientes de esa entidad en todas las categorías.

En algunas de dichas modalidades, en las que el módulo de celda de deterioro temporal comprende, además:

- 5 una cuarta etapa de red neuronal configurada para determinar datos del estado siguiente, utilizando datos de estado anterior modificados por la segunda etapa lógica de deterioro y un segundo tensor de entrada derivado de la transacción entrante, estando dicha cuarta etapa de red neuronal configurada para almacenar los datos del estado siguiente en la segunda memoria. La cuarta etapa de la red neuronal puede, en algunas modalidades, comprender una red neuronal recurrente.

10

Los datos correspondientes a la transacción entrante se suministran a cada uno de los módulos de historial categórico y de celda de deterioro temporal, en las modalidades en las que se proporcionan ambos. Cada uno de estos módulos puede recibir datos diferentes o puede existir alguna superposición (o potencialmente total) en los datos que recibe cada

15 módulo. En general, se espera que los dos módulos actúen sobre características diferentes, por lo que los datos proporcionados a cada módulo pueden ser específicos de los respectivos módulos.

20

Cuando se utiliza una capa de red neuronal (por ejemplo, una MLP) con pesos aprendidos para generar el tensor de entrada a partir de la transacción entrante, la misma capa de red neuronal puede generar el tensor de entrada de manera que una primera porción del tensor de entrada se suministre al módulo de historial categórico y una segunda porción del tensor de entrada se suministre al módulo de celda de deterioro temporal.

25

En algunas modalidades en las que se utiliza un módulo de celda de deterioro temporal, el sistema de aprendizaje automático está configurado para asignar datos de salida tanto del módulo de celda de deterioro temporal como del módulo de historial categórico al valor escalar representativo de la probabilidad de que la transacción entrante presente una anomalía dentro de una secuencia de acciones.

30

En algunas modalidades, la memoria está configurada para almacenar datos de estado para una pluralidad de entidades, en la que para cada entidad se almacena una pluralidad de categorías indexadas por un identificador de categoría respectivo, en la que los datos de

estado almacenados en la memoria para cada identificador de categoría corresponden a una transacción anterior asociada con la entidad respectiva y la categoría respectiva.

Se observará que las “categorías” a las que se hace referencia en el presente documento se relacionan con algún aspecto de la transacción, es decir, las categorías pueden verse como grupos o etiquetas que pueden aplicarse a transacciones que cumplen algunos criterios. Por ejemplo, las categorías pueden representar diferentes entidades secundarias (por ejemplo, comerciantes) con las que la entidad (o entidad “principal”, que puede ser, por ejemplo, el titular de una tarjeta) realiza transacciones. Existen varias categorías diferentes que podrían utilizarse. En algunas modalidades, cada una de la pluralidad de categorías comprende uno o más de: una entidad secundaria; un tipo de entidad secundaria; un valor de transacción; una banda de valor de transacción; un día de la semana; una hora del día; una hora del día; y una ventana de tiempo. En un conjunto particular de modalidades, la pluralidad de categorías puede comprender una combinación de una o más de estas categorías.

Cuando se observan desde un segundo aspecto, las modalidades de la presente invención proporcionan un método de procesamiento de datos asociados con una transacción propuesta. El método comprende:

20 recibir un evento entrante de un sistema de procesamiento de transacciones de un cliente, estando asociado el evento entrante con una solicitud de decisión de aprobación para la transacción propuesta;

analizar el evento entrante para extraer datos de la transacción propuesta, incluyendo la determinación de una diferencia de tiempo entre la transacción propuesta y una transacción anterior;

25 aplicar el sistema de aprendizaje automático de las modalidades del primer aspecto de la invención, para generar el valor escalar representativo de una probabilidad de que la transacción propuesta presente una anomalía dentro de una secuencia de acciones basada en los datos extraídos y la diferencia de tiempo, comprendiendo dicha aplicación acceder a la memoria para recuperar al menos los datos de estado de la entidad y el identificador de categoría asociado con la transacción;

determinar una salida binaria basada en el valor escalar generado por el sistema de aprendizaje automático, representando la salida binaria si la transacción propuesta se aprueba o rechaza; y

devolver la salida binaria al sistema de procesamiento de transacciones del cliente.

5

El segundo aspecto de la invención se extiende a un medio legible por computadora no transitorio, que comprende instrucciones que, cuando son ejecutadas por un procesador, hacen que el procesador lleve a cabo un método de procesamiento de datos asociado con una transacción propuesta. El método comprende:

10 recibir un evento entrante de un sistema de procesamiento de transacciones de un cliente, estando asociado el evento entrante con una solicitud de decisión de aprobación para la transacción propuesta;

analizar el evento entrante para extraer datos de la transacción propuesta, incluyendo la determinación de una diferencia de tiempo entre la transacción propuesta y

15 una transacción anterior;

aplicar el sistema de aprendizaje automático de las modalidades del primer aspecto de la invención, para generar el valor escalar representativo de una probabilidad de que la transacción propuesta presente una anomalía dentro de una secuencia de acciones basada en los datos extraídos y la diferencia de tiempo, comprendiendo dicha
20 aplicación acceder a la memoria para recuperar al menos los datos de estado de la entidad y el identificador de categoría asociado con la transacción; determinar una salida binaria basada en el valor escalar generado por el sistema de aprendizaje automático, representando la salida binaria si la transacción propuesta se aprueba o rechaza; y

devolver la salida binaria al sistema de procesamiento de transacciones del cliente.

25

El segundo aspecto de la invención se extiende a un producto de software informático que comprende instrucciones que, cuando son ejecutadas por un procesador, hacen que el procesador lleve a cabo un método de procesamiento de datos asociado con una transacción propuesta. El método comprende:

30 recibir un evento entrante de un sistema de procesamiento de transacciones de un cliente, estando asociado el evento entrante con una solicitud de decisión de aprobación para la transacción propuesta;

analizar el evento entrante para extraer datos de la transacción propuesta, incluyendo la determinación de una diferencia de tiempo entre la transacción propuesta y una transacción anterior;

- 5 aplicar el sistema de aprendizaje automático de las modalidades del primer aspecto de la invención, para generar el valor escalar representativo de una probabilidad de que la transacción propuesta presente una anomalía dentro de una secuencia de acciones basada en los datos extraídos y la diferencia de tiempo, comprendiendo dicha aplicación acceder a la memoria para recuperar al menos los datos de estado de la entidad y el identificador de categoría asociado con la transacción;
- 10 determinar una salida binaria basada en el valor escalar generado por el sistema de aprendizaje automático, representando la salida binaria si la transacción propuesta se aprueba o rechaza; y
- devolver la salida binaria al sistema de procesamiento de transacciones del cliente.

- 15 El módulo de historial categórico es novedoso e inventivo por sí mismo y, por tanto, cuando se observa desde un tercer aspecto, las modalidades de la invención proporcionan un módulo de historial categórico para el uso en un sistema de aprendizaje automático para procesar datos correspondientes a una transacción entrante asociada con una entidad, estando configurado el módulo de historial categórico para procesar al menos
- 20 algunos de dichos datos. El módulo de historial categórico comprende:

- i) una memoria configurada para almacenar datos de estado para una pluralidad de categorías indexadas por un identificador de categoría respectivo, en la que los datos de estado almacenados en la memoria para cada identificador de categoría corresponden a una transacción anterior asociada con la entidad y la categoría respectiva;
- 25 ii) una etapa lógica de deterioro configurada para modificar datos de estado almacenados en la memoria, en función de una diferencia de tiempo entre un momento de la transacción entrante y un momento de una transacción anterior, estando configurada dicha etapa lógica de deterioro para:
- a) recuperar dichos datos de estado y aplicar una función de
- 30 deterioro a los datos de estado para generar una versión deteriorada de los datos de estado, cuando la memoria contiene datos de estado para el par identificador de entidad y categoría asociado con la transacción entrante, en la que la función

de deterioro depende de la diferencia de tiempo, y generar la versión deteriorada de los datos de estado; y

- 5 iii) una etapa lógica de actualización configurada para: actualizar la salida de datos de estado de la etapa lógica de deterioro utilizando un tensor de entrada, generado a partir de los datos correspondientes a la transacción entrante para generar datos de estado actualizados; para almacenar dichos datos de estado actualizados en la memoria para el par identificador de entidad y categoría asociado con la transacción entrante; y para generar un tensor de salida que comprenda los datos de estado actualizados.

- 10 El tercer aspecto de la invención se extiende adicionalmente a un método de procesamiento de datos asociados a una transacción propuesta asociada a una entidad. El método comprende:

- 15 i) almacenar, en una memoria, datos de estado para una pluralidad de categorías indexadas por un identificador de categoría respectivo, en el que los datos de estado almacenados en la memoria para cada identificador de categoría corresponden a una transacción anterior asociada con la entidad y la categoría respectiva;

- ii) modificar datos de estado almacenados en la memoria, en función de una diferencia de tiempo entre un momento de la transacción entrante y un momento de una transacción anterior, en el que el paso de modificar los datos de estado comprende:

- 20 a) recuperar dichos datos de estado y aplicar una función de deterioro a los datos de estado para generar una versión deteriorada de los datos de estado, cuando la memoria contiene datos de estado para el par identificador de entidad y categoría asociado con la transacción entrante, en el que la función de deterioro depende de la diferencia de tiempo, y generar la versión deteriorada de los datos de estado; y

- 25 iii) actualizar la salida de datos de estado utilizando un tensor de entrada, generado a partir de los datos correspondientes a la transacción entrante para generar datos de estado actualizados; almacenar dichos datos de estado actualizados en la memoria para el par identificador de entidad y categoría asociado con la transacción entrante; y generar un tensor de salida que comprende los datos de estado actualizados;
- 30

El tercer aspecto de la invención se extiende además a un medio legible por computadora no transitorio que comprende instrucciones que, cuando son ejecutadas por un

procesador, hacen que el procesador lleve a cabo un método de procesamiento de datos asociado con una transacción propuesta asociada con una entidad. El método comprende:

- i) almacenar, en una memoria, datos de estado para una pluralidad de categorías indexadas por un identificador de categoría respectivo, en el que los datos de estado almacenados en la memoria para cada identificador de categoría corresponden a una transacción anterior asociada con la entidad y la categoría respectiva;
- ii) modificar datos de estado almacenados en la memoria, en función de una diferencia de tiempo entre un momento de la transacción entrante y un momento de una transacción anterior, en el que el paso de modificar los datos de estado comprende:
 - a) recuperar dichos datos de estado y aplicar una función de deterioro a los datos de estado para generar una versión deteriorada de los datos de estado, cuando la memoria contiene datos de estado para el par identificador de entidad y categoría asociado con la transacción entrante, en el que la función de deterioro depende de la diferencia de tiempo, y generar la versión deteriorada de los datos de estado; y
 - iii) actualizar la salida de datos de estado utilizando un tensor de entrada, generado a partir de los datos correspondientes a la transacción entrante para generar datos de estado actualizados; almacenar dichos datos de estado actualizados en la memoria para el par identificador de entidad y categoría asociado con la transacción entrante; y generar un tensor de salida que comprende los datos de estado actualizados;

El tercer aspecto de la invención se extiende además a un producto de software informático que comprende instrucciones que, cuando son ejecutadas por un procesador, hacen que el procesador lleve a cabo un método de procesamiento de datos asociado con una transacción propuesta asociada con una entidad. El método comprende:

- i) almacenar, en una memoria, datos de estado para una pluralidad de categorías indexadas por un identificador de categoría respectivo, en el que los datos de estado almacenados en la memoria para cada identificador de categoría corresponden a una transacción anterior asociada con la entidad y la categoría respectiva;
- ii) modificar datos de estado almacenados en la memoria, en función de una diferencia de tiempo entre un momento de la transacción entrante y un momento de una transacción anterior, en el que el paso de modificar los datos de estado comprende:

a) recuperar dichos datos de estado y aplicar una función de deterioro a los datos de estado para generar una versión deteriorada de los datos de estado, cuando la memoria contiene datos de estado para el par identificador de entidad y categoría asociado con la transacción entrante, en el que la función de deterioro depende de la diferencia de tiempo, y generar la versión deteriorada de los datos de estado; y

iii) actualizar la salida de datos de estado utilizando un tensor de entrada, generado a partir de los datos correspondientes a la transacción entrante para generar datos de estado actualizados; almacenar dichos datos de estado actualizados en la memoria para el par identificador de entidad y categoría asociado con la transacción entrante; y generar un tensor de salida que comprende los datos de estado actualizados;

Se observará que las características opcionales descritas anteriormente con respecto a las modalidades del primer aspecto de la invención se aplican igualmente, cuando sea técnicamente apropiado, al segundo y tercer aspecto de la invención.

Cuando sea técnicamente apropiado, se podrán combinar modalidades de la invención. Se describen en el presente documento modalidades que comprenden determinadas características/elementos. La divulgación también se extiende a modalidades separadas que consisten, o consisten esencialmente, en dichas características/elementos.

Las referencias técnicas como, por ejemplo, patentes y solicitudes se incorporan en el presente documento como referencia.

Cualquier modalidad específica y explícitamente citada en el presente documento puede formar la base de una exención de responsabilidad, ya sea sola o en combinación con una o más modalidades adicionales.

En el contexto de esta memoria descriptiva, “que comprende” debe interpretarse como “que incluye”.

Los aspectos de la invención que comprenden ciertos elementos también pretenden extenderse a modalidades alternativas que “consisten” o “consisten esencialmente” en los elementos relevantes.

- 5 El término “memoria” debe entenderse como cualquier medio adecuado para el almacenamiento de datos e incluye tanto la memoria volátil como la no volátil, según sea apropiado para la aplicación prevista. Los expertos en la materia observarán que esto incluye, entre otros, memoria de acceso aleatorio (RAM), RAM estática (SRAM), RAM dinámica (DRAM), almacenamiento magnético, almacenamiento de estado sólido y
- 10 memoria flash. Se observará que también se pueden utilizar combinaciones de uno o más de aquellos para el almacenamiento de datos, según sea técnicamente apropiado (por ejemplo, utilizando una memoria volátil de acceso más rápido para datos a los que se accede con frecuencia).
- 15 El término “datos” se utiliza en diferentes contextos en el presente documento para referirse a información digital como, por ejemplo, la representada por estructuras de bits conocidas dentro de uno o más lenguajes de programación. En uso, los datos pueden referirse a información digital que se almacena como secuencias de bits dentro de la memoria de la computadora. Ciertos modelos de aprendizaje automático pueden operar
- 20 en matrices estructuradas de datos de un formato de bits predefinido. Los expertos en la materia comprenderán fácilmente que estos pueden denominarse matrices, matrices multidimensionales, vectores, tensores u otros términos similares. Cabe señalar que, para los métodos de aprendizaje automático, las matrices o tensores multidimensionales, por ejemplo, con una extensión definida en varias dimensiones, se pueden “aplanar” para
- 25 representarlos (por ejemplo, dentro de la memoria) como una secuencia o vector de valores almacenados, de acuerdo con el formato predefinido (por ejemplo, número entero o de punto flotante de n bits, con o sin signo). Por lo tanto, el término “tensor” tal como se utiliza en el presente documento cubre matrices multidimensionales con una o más dimensiones (por ejemplo, vectores, matrices, matrices volumétricas, etc.).
- 30

Los principios de la presente invención se aplican independientemente del formato de datos particular elegido. Los datos pueden representarse como matrices, vectores, tensores o cualquier otro formato adecuado. Para facilitar la referencia, estos términos se

utilizan indistintamente en el presente documento, y las referencias a un “vector” o “vectores” de valores deben entenderse como extendidas a cualquier tensor o tensores de valores n-dimensionales, según corresponda. De manera similar, las referencias a un “tensor” o “tensores” de valores deben entenderse extendidas a los vectores, que los expertos en la materia entienden simplemente como tensores unidimensionales. Los principios de la presente invención pueden aplicarse independientemente del formato de las estructuras de datos utilizadas para estas matrices de valores. Por ejemplo, los datos de estado se pueden almacenar en la memoria como tensores unidimensionales (es decir, vectores) o como un tensor con una dimensionalidad de dos o más (es decir, tensores), y los expertos en la materia comprenderán fácilmente que se pueden realizar modificaciones adecuadas a los elementos de procesamiento de datos para manejar el formato de datos seleccionado. Las posiciones relativas entre varios valores de estado, por ejemplo, cómo se ordenan dentro de un vector o tensor, generalmente no importan, y el alcance de la presente invención no está limitado a ningún formato o estructura de datos en particular.

El término “representación numérica estructurada” se utiliza para referirse a datos numéricos en forma estructurada como, por ejemplo, una matriz de una o más dimensiones que almacena valores numéricos con un tipo de datos común como, por ejemplo, números enteros o valores flotantes. Una representación numérica estructurada puede comprender un vector o un tensor (como se utiliza en la terminología del aprendizaje automático). Una representación numérica estructurada normalmente se almacena como un conjunto de ubicaciones de memoria indexadas y/o consecutivas, por ejemplo, una matriz unidimensional de flotantes de 64 bits puede representarse en la memoria de una computadora como una secuencia consecutiva de ubicaciones de memoria de 64 bits en un sistema informático de 64 bits.

El término “datos de transacción” se utiliza en el presente documento para referirse a datos electrónicos asociados con una transacción. Una transacción comprende una serie de comunicaciones entre diferentes sistemas electrónicos para implementar un pago o intercambio. En general, los datos de transacciones pueden incluir datos que indican eventos (por ejemplo, acciones realizadas en el tiempo) que se relacionan con el procesamiento de transacciones y pueden ser informativos para dicho procesamiento. Los datos de transacción pueden comprender datos estructurados, no estructurados y

semiestructurados, o cualquier combinación de ellos. Los datos de transacción también pueden incluir datos asociados con una transacción como, por ejemplo, los datos utilizados para procesar una transacción. En ciertos casos, los datos de transacciones pueden usarse ampliamente para referirse a acciones realizadas con respecto a uno o más dispositivos electrónicos. Los datos de transacción pueden adoptar distintas formas dependiendo de la implementación precisa. Sin embargo, se pueden convertir diferentes tipos y formatos de datos mediante el procesamiento previo o posterior, según corresponda.

El término “interfaz” se utiliza en el presente documento para referirse a cualquier interfaz física y/o lógica que permite una o más entradas y salidas de datos. Una interfaz puede implementarse mediante una interfaz de red adaptada para enviar y/o recibir datos, o mediante la recuperación de datos de una o más ubicaciones de memoria, según lo implementado por un procesador que ejecuta un conjunto de instrucciones. Una interfaz también puede comprender acoplamientos físicos (de red) a través de los cuales se reciben datos como, por ejemplo, hardware para permitir comunicaciones cableadas o inalámbricas a través de un medio particular. Una interfaz puede comprender una interfaz de programación de aplicaciones y/o una llamada o retorno de método. Por ejemplo, en una implementación de software, una interfaz puede comprender pasar datos y/o referencias de memoria a una función iniciada a través de una llamada de método, donde la función comprende código de programa de computadora que es ejecutado por uno o más procesadores; en una implementación de hardware, una interfaz puede comprender una interconexión cableada entre diferentes chips, conjuntos de chips o partes de chips. En los dibujos, una interfaz puede estar indicada por un límite de un bloque de procesamiento que tiene una flecha hacia adentro y/o hacia afuera que representa una transferencia de datos.

Los términos “componente” y “módulo” se utilizan indistintamente para referirse a una estructura de hardware que tiene una función específica (por ejemplo, en forma de asignación de datos de entrada a datos de salida) o una combinación de hardware general y software específico (por ejemplo, código de programa de computadora específico que se ejecuta en uno o más procesadores de propósito general). Un componente o módulo puede implementarse como un chipset empaquetado específico, por ejemplo, un circuito integrado de aplicación específica (ASIC) o una matriz de puertas programables en campo

(FPGA) programada, y/o como un objeto de software, clase, instancia de clase, script, porción de código o similar, tal como se ejecuta en uso por un procesador.

El término “modelo de aprendizaje automático” se utiliza en el presente documento para referirse al menos a una implementación ejecutada por hardware de un modelo o función de aprendizaje automático. Los modelos conocidos dentro del campo del aprendizaje automático incluyen modelos de regresión logística, modelos Naïve Bayes, bosques aleatorios, máquinas de vectores de soporte y redes neuronales artificiales. Se pueden proporcionar implementaciones de clasificadores dentro de una o más bibliotecas de programación de aprendizaje automático, incluyendo, entre otras, scikit-learn, TensorFlow y PyTorch.

El término “asignar” se utiliza en el presente documento para referirse a la transformación o conversión de un primer conjunto de valores de datos en un segundo conjunto de valores de datos. Los dos conjuntos de valores de datos pueden ser matrices de diferentes tamaños, siendo una matriz de salida de menor dimensionalidad que una matriz de entrada. Las matrices de entrada y salida pueden tener tipos de datos comunes o diferentes. En ciertos ejemplos, la asignación es una asignación unidireccional a un valor escalar.

El término “arquitectura de red neuronal” se refiere a un conjunto de una o más redes neuronales artificiales que están configuradas para realizar una tarea particular de procesamiento de datos. Por ejemplo, una “arquitectura de red neuronal” puede comprender una disposición particular de una o más capas de red neuronal de uno o más tipos de red neuronal. Los tipos de redes neuronales incluyen redes neuronales convolucionales, redes neuronales recurrentes y redes neuronales de propagación hacia adelante. Las redes neuronales convolucionales implican la aplicación de una o más operaciones de convolución. Las redes neuronales recurrentes implican un estado interno que se actualiza durante una secuencia de entradas. De este modo, se considera que las redes neuronales recurrentes incluyen una forma de conexión recurrente o de retroalimentación mediante la cual un estado de la red neuronal recurrente en un momento o iteración determinados (por ejemplo, t) se actualiza utilizando un estado de la red neuronal recurrente en un momento o iteración anteriores (por ejemplo, $t-1$). Las

redes neuronales de propagación hacia adelante implican operaciones de transformación sin retroalimentación, por ejemplo, las operaciones se aplican en una secuencia unidireccional desde la entrada hasta la salida. Las redes neuronales de propagación hacia adelante incluyen “redes neuronales” simples y “redes neuronales completamente conectadas”. Los expertos en la materia comprenderán que un “perceptrón multicapa” es un término utilizado para describir una capa completamente conectada y es un caso especial de una red neuronal de propagación hacia adelante.

El término red neuronal “profunda” se utiliza para indicar que la red neuronal comprende varias capas de red neuronal en serie (se debe tener en cuenta que esta terminología “profunda” se utiliza tanto con redes neuronales de propagación hacia adelante como con redes neuronales recurrentes). Ciertos ejemplos descritos en el presente documento hacen uso de redes neuronales recurrentes y completamente conectadas.

Una “capa de red neuronal”, como se define normalmente en las herramientas y bibliotecas de programación de aprendizaje automático, puede considerarse una operación que asigna datos de entrada a datos de salida. Una “capa de red neuronal” puede aplicar uno o más parámetros como, por ejemplo, pesos, para asignar datos de entrada a datos de salida. También se pueden aplicar uno o más términos de sesgo. Los pesos y sesgos de una capa de red neuronal se pueden aplicar utilizando una o más matrices o arreglos multidimensionales. En general, una capa de red neuronal tiene una pluralidad de parámetros cuyo valor influye en cómo la capa asigna los datos de entrada a los datos de salida. Estos parámetros pueden entrenarse de forma supervisada optimizando una función objetivo. Generalmente esto implica minimizar una función de pérdida. Ciertos parámetros también pueden ser entrenados previamente o fijados de otra manera. Los parámetros fijos pueden considerarse como datos de configuración que controlan el funcionamiento de la capa de red neuronal. Una capa o arquitectura de red neuronal puede comprender una mezcla de parámetros fijos y aprendibles. Una capa de red neuronal recurrente puede aplicar una serie de operaciones para actualizar un estado recurrente y transformar los datos de entrada. La actualización del estado recurrente y la transformación de los datos de entrada pueden implicar transformaciones de uno o más de un estado recurrente anterior y los datos de entrada. Una capa de red neuronal recurrente se puede entrenar desenrollando una unidad recurrente modelada, como se

puede aplicar en herramientas y bibliotecas de programación de aprendizaje automático. Aunque una red neuronal recurrente puede considerarse compuesta por varias (sub)capas para aplicar diferentes operaciones de activación, la mayoría de las herramientas y bibliotecas de programación de aprendizaje automático se refieren a la aplicación de la red neuronal recurrente en su conjunto como una “capa de red neuronal” y aquí se seguirá esta convención. Por último, una capa de red neuronal de propagación hacia adelante puede aplicar uno o más de un conjunto de pesos y sesgos a los datos de entrada para generar datos de salida. Esta operación se puede representar como una operación matricial (por ejemplo, donde se puede incluir un término de sesgo añadiendo un valor de 1 a los datos de entrada). Como alternativa, se puede aplicar un sesgo a través de una operación de adición separada. Como se explicó anteriormente, el término “tensor” se utiliza, según las bibliotecas de aprendizaje automático, para referirse a una matriz que puede tener varias dimensiones, por ejemplo, un tensor puede comprender un vector, una matriz o una estructura de datos de mayor dimensionalidad. En un ejemplo preferido, los tensores descritos pueden comprender vectores con un número predefinido de elementos.

Para modelar funciones no lineales complejas, una capa de red neuronal como la descrita anteriormente puede ir seguida de una función de activación no lineal. Las funciones de activación comunes incluyen la función sigmoidea, la función tanh y las unidades lineales rectificadas (RELU). Existen muchas otras funciones de activación que pueden aplicarse. Se puede seleccionar una función de activación según las pruebas y preferencias. Las funciones de activación pueden omitirse en determinadas circunstancias y/o formar parte de la estructura interna de una capa de red neuronal.

Las arquitecturas de redes neuronales de ejemplo descritas en el presente documento pueden configurarse mediante entrenamiento. En ciertos casos, los parámetros “aprendibles” o “entrenables” pueden entrenarse utilizando un enfoque llamado retropropagación. Durante la retropropagación, las capas de red neuronal que componen cada arquitectura de red neuronal se inicializan (por ejemplo, con pesos aleatorios) y luego se utilizan para hacer una predicción utilizando un conjunto de datos de entrada de un conjunto de entrenamiento (por ejemplo, un llamado pase “hacia adelante”). La predicción se utiliza para evaluar una función de pérdida. Por ejemplo, una salida de “verdad fundamental” puede compararse con una salida prevista y la diferencia puede formar

parte de la función de pérdida. En ciertos ejemplos, una función de pérdida puede basarse en una diferencia absoluta entre un valor escalar previsto y una etiqueta de verdad fundamental binaria. El conjunto de entrenamiento puede comprender un conjunto de transacciones. Si se utilizan métodos de descenso de gradiente, la función de pérdida se

5 utiliza para determinar un gradiente de la función de pérdida con respecto a los parámetros de la arquitectura de la red neuronal, donde luego el gradiente se utiliza para propagar hacia atrás una actualización de los valores de los parámetros de la arquitectura de la red neuronal. Normalmente, la actualización se propaga de acuerdo con la derivada de los pesos de las capas de la red neuronal. Por ejemplo, se puede determinar un

10 gradiente de la función de pérdida con respecto a los pesos de las capas de la red neuronal y utilizarlo para determinar una actualización de los pesos que minimice la función de pérdida. En este caso, se pueden utilizar técnicas de optimización como, por ejemplo, el descenso de gradiente, el descenso de gradiente estocástico, Adam, etc. para ajustar los pesos. La regla de la cadena y las funciones de autodiferenciación se pueden aplicar para

15 calcular de manera eficiente el gradiente de la función de pérdida, trabajando hacia atrás a través de las capas de la red neuronal a su vez.

Breve descripción de los dibujos

20 A continuación, se describirán ciertas modalidades de la presente invención con referencia a los dibujos adjuntos, en los que:

Las FIG. 1A a 1C son diagramas esquemáticos que muestran diferentes ejemplos de infraestructuras electrónicas para el procesamiento de transacciones;

25 Las FIG. 2A y 2B son diagramas esquemáticos que muestran diferentes ejemplos de sistemas de almacenamiento de datos para el uso por un sistema de procesamiento de transacciones de aprendizaje automático;

Las FIG. 3A y 3B son diagramas esquemáticos que muestran diferentes ejemplos de datos de transacciones.

30 La FIG. 4 es un diagrama esquemático que muestra componentes de ejemplo de un sistema de procesamiento de transacciones de aprendizaje automático;

Las FIG. 5A y 5B son diagramas de secuencia que muestran un conjunto de ejemplos de procesos realizados por diferentes entidades informáticas sobre datos de transacciones;

La FIG. 6 es un diagrama esquemático que muestra un sistema de aprendizaje automático de acuerdo con una modalidad de la presente invención; y

La FIG. 7 es un diagrama esquemático que muestra el funcionamiento del módulo de historial categórico utilizado en el sistema de aprendizaje automático de la FIG. 6.

5

Descripción detallada

Se describen en el presente documento ciertas modalidades ejemplares que se relacionan con un sistema de aprendizaje automático para el uso en el procesamiento de

10 transacciones. En ciertas modalidades, se aplica un sistema de aprendizaje automático en canales de procesamiento de transacciones de alto volumen en tiempo real, para proporcionar una indicación de si una transacción o entidad coincide con patrones de actividad o acciones previamente observados y/o previstos, por ejemplo, una indicación de si una transacción o entidad es “normal” o “anómala”. El término “conductual” se utiliza en

15 el presente documento para referirse a este patrón de actividad o acciones. La indicación puede comprender un valor escalar normalizado dentro de un rango predefinido (por ejemplo, 0 a 1) que luego se puede utilizar para prevenir fraudes y otros usos indebidos de los sistemas de pago. Los sistemas de aprendizaje automático pueden aplicar modelos de aprendizaje automático que se actualizan a medida que se obtienen más datos de

20 transacciones, por ejemplo, que se entrenan constantemente en función de nuevos datos, a fin de reducir los falsos positivos y mantener la precisión de la métrica de salida. Los presentes ejemplos pueden ser particularmente útiles para prevenir el fraude, en casos en los que no se puede determinar la presencia física de una tarjeta de pago (por ejemplo, transacciones en línea denominadas “sin tarjeta presente”) o para transacciones

25 comerciales en las que las transacciones de alto valor pueden ser rutinarias y donde puede ser difícil clasificar patrones de comportamiento como “inesperados”. Como tal, los presentes ejemplos facilitan el procesamiento de transacciones, ya que estas transacciones se realizan principalmente “en línea”, es decir, se realizan digitalmente a través de una o más redes de comunicaciones públicas.

30

Ciertas modalidades descritas en el presente documento permiten que los modelos de aprendizaje automático se adapten para que sean específicos de patrones de comportamiento entre ciertos pares de entidades (como, por ejemplo, titulares de

cuentas) y categorías (como, por ejemplo, comerciantes, montos de transacciones, horas del día y otros). Por ejemplo, los modelos de aprendizaje automático pueden modelar patrones de comportamiento específicos de pares entidad-categoría. Los sistemas de aprendizaje automático descritos en el presente documento pueden proporcionar sistemas de aprendizaje automático que se actualizan dinámicamente a pesar de los grandes flujos de transacciones y/o a pesar de la necesidad de segregar diferentes fuentes de datos.

Como se describió anteriormente, las modalidades de la presente invención se pueden aplicar a una amplia variedad de transacciones digitales, incluyendo, entre otros, pagos con tarjeta, transferencias denominadas “telegráficas”, pagos entre pares, pagos mediante el Sistema de Compensación Automatizado de Banqueros (BACS) y pagos mediante la Cámara de Compensación Automatizada (ACH). Los resultados del sistema de aprendizaje automático se pueden utilizar para prevenir una amplia variedad de comportamientos fraudulentos y delictivos como, por ejemplo, fraude con tarjetas, fraude con aplicaciones, fraude con pagos, fraude comercial, fraude con juegos y lavado de dinero.

Los presentes sistemas de aprendizaje automático ejemplares, por ejemplo, tal como están configurados y/o entrenados como se describe a continuación, están configurados para detectar novedades en las interacciones entidad-categoría.

A continuación, se describe una modalidad ejemplar no limitante de un sistema de aprendizaje automático de acuerdo con una modalidad de la invención. Las FIG. 1A a 5B proporcionan contexto para el sistema de aprendizaje automático. Un sistema de aprendizaje automático en forma de una arquitectura de red neuronal específica como se muestra en las FIG. 6 y 7.

Ejemplo de sistema de procesamiento de transacciones

Las FIG. 1A a 1C muestran un conjunto de sistemas de procesamiento de transacciones ejemplares 100, 102, 104. Estos sistemas de procesamiento de transacciones ejemplares se describen para proporcionar contexto para las invenciones analizadas en el presente documento, pero no deben considerarse como limitantes; la configuración de cualquier implementación puede diferir en función de los requisitos específicos de esa

implementación. Sin embargo, los sistemas de procesamiento de transacciones ejemplares descritos permiten a los expertos en la materia identificar ciertas características técnicas de alto nivel, que son relevantes para la descripción a continuación. Los tres sistemas de procesamiento de transacciones ejemplares 100, 102, 104 muestran diferentes áreas
5 donde pueden ocurrir variaciones.

Las FIG. 1A a 1C muestran un conjunto de dispositivos cliente 110 que están configurados para iniciar una transacción. En este ejemplo, el conjunto de dispositivos cliente 110 comprende un teléfono inteligente 110-A, una computadora 110-B, un sistema de punto
10 de venta (POS) 110-C y un dispositivo comercial portátil 110-D. Estos dispositivos cliente 110 proporcionan un conjunto de ejemplos no exhaustivos. Generalmente se puede utilizar cualquier dispositivo electrónico o conjunto de dispositivos para realizar una transacción. En un caso, la transacción comprende una compra o un pago. Por ejemplo, la compra o el pago pueden ser una compra o un pago en línea o móvil realizado a través del
15 teléfono inteligente 110-A o la computadora 110-B, o pueden ser una compra o un pago realizado en las instalaciones de un comerciante, como por ejemplo a través del sistema POS 110-C o el dispositivo portátil del comerciante 110-D. La compra o el pago pueden ser por bienes y/o servicios.

20 En las FIG. 1A a 1C, los dispositivos cliente 110 están acoplados comunicativamente a una o más redes informáticas 120. Los dispositivos cliente 110 pueden estar acoplados comunicativamente de diversas maneras, incluso mediante una o más redes cableadas y/o inalámbricas, incluyendo redes de telecomunicaciones. En los ejemplos preferidos, todas las comunicaciones a través de la o las redes de computadoras están protegidas, por
25 ejemplo, mediante protocolos de seguridad de la capa de transporte (TLS). En la FIG. 1A se muestran dos redes de computadoras, 120-A y 120-B. Pueden ser redes separadas o partes diferentes de una red común. La primera red informática 120-A acopla comunicativamente los dispositivos cliente 110 a un servidor comercial 130. El servidor comercial 130 puede ejecutar un proceso informático que implementa un flujo de proceso para la transacción.
30 Por ejemplo, el servidor comercial 130 puede ser un servidor *back end* que maneja solicitudes de transacciones recibidas desde el sistema POS 110-C o el dispositivo comercial portátil 110-D o puede ser utilizado por un comerciante en línea para implementar un sitio web donde se pueden realizar compras. Se observará que los

ejemplos de las FIG. 1A a 1C son simplificaciones necesarias de arquitecturas reales; por ejemplo, puede haber varios dispositivos servidor interactuantes que implementan un comerciante en línea, incluyendo dispositivos servidor separados para proporcionar páginas de lenguaje de marcado de hipertexto (HTML) que detallan un producto y/o servicio y para manejar un proceso de pago.

En la FIG. 1A, el servidor comercial 130 está acoplado comunicativamente a un conjunto adicional de dispositivos servidor *back end* para procesar la transacción. En la FIG. 1A, el servidor comercial 130 está acoplado comunicativamente a un servidor procesador de pagos 140 a través de una segunda red 120-B. El servidor procesador de pagos 140 está acoplado comunicativamente a un primer dispositivo de almacenamiento de datos 142 que almacena datos de transacción 146 y a un segundo dispositivo de almacenamiento de datos 144 que almacena datos auxiliares 148. Los datos de transacción 146 pueden comprender lotes de datos de transacción relacionados con diferentes transacciones que se realizan durante un período de tiempo. Los datos auxiliares 148 pueden comprender datos asociados con las transacciones como, por ejemplo, registros que almacenan datos del comerciante y/o del usuario final. En la FIG. 1A, el servidor de procesamiento de pagos 140 está acoplado comunicativamente a un servidor de aprendizaje automático 150 a través de la segunda red 120-B. El servidor de aprendizaje automático 150 implementa un sistema de aprendizaje automático 160 para el procesamiento de datos de transacciones. El sistema de aprendizaje automático 160 está dispuesto para recibir datos de entrada 162 y asignarlos a datos de salida 164 que son utilizados por el servidor procesador de pagos 140 para procesar una transacción particular como, por ejemplo, una que surge de los dispositivos cliente 110. En un caso, el sistema de aprendizaje automático 160 recibe al menos datos de transacción asociados con la transacción particular y proporciona una alerta o salida numérica que es utilizada por el servidor procesador de pagos 140 para determinar si la transacción debe ser autorizada (es decir, aprobada) o rechazada. Como tal, la salida del sistema de aprendizaje automático 160 puede comprender una etiqueta, una alerta u otra indicación de fraude o actividad maliciosa o anómala general. La salida puede incluir una indicación probabilística como, por ejemplo, una puntuación o una probabilidad. En un caso, los datos de salida 164 pueden comprender un valor numérico escalar. Los datos de entrada 162 pueden comprender además datos derivados de uno o más de los datos de transacción 146 y los datos auxiliares 148. En un caso, los datos de

salida 164 indican un nivel de desviación de un patrón de comportamiento esperado específico basado en observaciones o mediciones pasadas. Por ejemplo, esto puede indicar fraude o comportamiento delictivo, ya que a menudo difiere significativamente de los patrones de comportamiento observados, especialmente a gran escala. Los datos de salida 164 pueden formar una medida de comportamiento. El patrón de comportamiento esperado puede definirse, ya sea explícita o implícitamente, basándose en interacciones observadas entre diferentes entidades dentro del flujo del proceso de transacción como, por ejemplo, usuarios finales o clientes, comerciantes (incluyendo los puntos de venta y las ubicaciones o entidades *back end* donde aquellos pueden diferir) y bancos.

10

El sistema de aprendizaje automático 160 puede implementarse como parte de un proceso de procesamiento de transacciones. Más adelante se describe un ejemplo de canalización de procesamiento de transacciones con respecto a las FIG. 5A y 5B. Un canal de procesamiento de transacciones puede comprender comunicaciones electrónicas entre los dispositivos cliente 110, el servidor comercial 130, el servidor procesador de pagos 140 y el servidor de aprendizaje automático 150. También pueden intervenir otros dispositivos servidor como, por ejemplo, servidores bancarios que proporcionan autorización de un banco emisor. En ciertos casos, los dispositivos cliente 110 pueden comunicarse directamente con el servidor procesador de pagos 140. En uso, un canal de procesamiento de transacciones normalmente debe completarse en uno o doscientos milisegundos. En general, los tiempos de procesamiento de subsegundos pueden considerarse en tiempo real (por ejemplo, los seres humanos normalmente perciben eventos en un lapso de 400 ms). Además, 100-200 ms puede ser la latencia máxima deseada del tiempo de ida y vuelta completo para el procesamiento de transacciones; dentro de este lapso, el tiempo asignado para el sistema de aprendizaje automático 160 puede ser una pequeña fracción de esta cantidad total como, por ejemplo, 10 ms (es decir, menos del 5-10 % del tiempo de procesamiento objetivo), ya que la mayor parte del tiempo puede reservarse para otras operaciones en el flujo de procesamiento de transacciones. Esto presenta una restricción técnica para la implementación del sistema de aprendizaje automático 160. Además, en implementaciones del mundo real, los volúmenes de procesamiento promedio pueden ser del orden de 1000 a 2000 por segundo. Esto significa que la mayoría de los sistemas de aprendizaje automático “listos para usar” no son adecuados para implementar el sistema de aprendizaje automático 160. Esto significa, además, que la mayoría de los enfoques de

30

aprendizaje automático descritos en artículos académicos no pueden implementarse dentro del canal de procesamiento de transacciones mencionado anteriormente sin adaptaciones no obvias.

- 5 Además, los sistemas de aprendizaje automático existentes no extraen la novedad de la interacción entidad-categoría. Como se explicará más adelante con referencia a las FIG. 6 y 7, un sistema de aprendizaje automático de acuerdo con una modalidad de la presente invención se puede utilizar para proporcionar una serie de beneficios técnicos.
- 10 La FIG. 1B muestra una variación 102 del sistema de procesamiento de transacciones de ejemplo 100 de la FIG. 1A. En esta variación 102, el sistema de aprendizaje automático 160 se implementa dentro de la infraestructura informática del procesador de pagos, por ejemplo, ejecutado por el servidor del procesador de pagos 140 y/o ejecutado en un servidor acoplado localmente dentro de la misma red local que el servidor del procesador
- 15 de pagos 140. La variación 102 de la FIG. 1B puede ser preferible para procesadores de pagos más grandes, ya que permite tiempos de respuesta más rápidos, mayor control y seguridad mejorada. Sin embargo, funcionalmente, el canal de procesamiento de transacciones puede ser similar al de la FIG. 1A. Por ejemplo, en el ejemplo de la FIG. 1A, el sistema de aprendizaje automático 160 puede iniciarse mediante una llamada de interfaz
- 20 de programación de aplicaciones (API) externa segura como, por ejemplo, una llamada de API de Transferencia de Estado de Representación (REST) utilizando el Protocolo de Transferencia de Hipertexto Seguro (HTTPS), mientras que en la FIG. 1B, el sistema de aprendizaje automático 160 puede iniciarse mediante una llamada API interna, pero donde una API final común puede manejar ambas solicitudes (por ejemplo, la API REST HTTPS
- 25 puede proporcionar un contenedor externo para la API interna).

- La FIG. 1C muestra otra variación 104 del sistema de procesamiento de transacciones de ejemplo 100 de la FIG. 1A. En esta variación 104, el sistema de aprendizaje automático 160 está acoplado comunicativamente a los dispositivos de almacenamiento de datos locales
- 30 170. Por ejemplo, los dispositivos de almacenamiento de datos 170 pueden estar en la misma red local que el servidor de aprendizaje automático 150 o pueden comprender una red de almacenamiento local accesible al servidor de aprendizaje automático 150. En este caso, existe una pluralidad de dispositivos de almacenamiento de datos locales 170-A a

170-N, donde cada dispositivo de almacenamiento de datos almacena datos auxiliares particionados 172. Los datos auxiliares particionados 172 pueden comprender parámetros para uno o más modelos de aprendizaje automático. En un caso, los datos auxiliares 172 pueden comprender un estado para modelos de aprendizaje automático, donde el estado
5 puede relacionarse con una entidad específica como, por ejemplo, un usuario o un comerciante. Puede ser necesario aplicar la partición de los datos auxiliares 172 para cumplir con los requisitos de seguridad establecidos por un tercero como, por ejemplo, el procesador de pagos, uno o más bancos y/o uno o más comerciantes. En uso, el sistema de aprendizaje automático 160 accede a los datos auxiliares 172-A a 172-N a través de la
10 pluralidad de dispositivos de almacenamiento de datos locales 170-A a 170-N, basándose en los datos de entrada 162. Por ejemplo, los datos de entrada 162 pueden recibirse mediante una solicitud de API desde una fuente particular y/o pueden comprender datos que identifican que se utilizará una partición particular para manejar la solicitud de API. En las FIG. 2A y 2B se presentan más detalles de diferentes sistemas de almacenamiento que
15 se pueden aplicar para cumplir con los requisitos de seguridad 2A and 2B.

Ejemplos de configuraciones de almacenamiento de datos

Las FIG. 2A y 2B muestran dos configuraciones de almacenamiento de datos de ejemplo
20 200 y 202 que pueden ser utilizadas por un sistema de aprendizaje automático de ejemplo 210 para el procesamiento de datos de transacciones. Los ejemplos de las FIG. 2A y 2B son dos ejemplos no limitantes que muestran diferentes opciones disponibles para implementaciones, y se pueden seleccionar configuraciones particulares según circunstancias individuales. El sistema de aprendizaje automático 210 puede comprender
25 una implementación del sistema de aprendizaje automático 160 descrito en los ejemplos anteriores de las FIG. 1A a 1C. Los ejemplos de las FIG. 2A y 2B permiten el procesamiento de datos de transacciones que están protegidos mediante parámetros criptográficos heterogéneos, por ejemplo, para que el sistema de aprendizaje automático 210 procese de
30 forma segura datos de transacciones para entidades heterogéneas. Se observará que las configuraciones de las FIG. 2A y 2B no pueden utilizarse si el sistema de aprendizaje automático 160 se implementa para un único conjunto de transacciones seguras y datos auxiliares, por ejemplo, dentro de un sistema de procesamiento de transacciones interno o como un sistema alojado para el uso por un único procesador de pagos.

La FIG. 2A muestra un sistema de aprendizaje automático 210 acoplado comunicativamente a un bus de datos 220. El bus de datos 220 puede comprender un bus de datos interno del servidor de aprendizaje automático 150 o puede formar parte de una red de área de almacenamiento. El bus de datos 220 acopla comunicativamente el sistema de aprendizaje automático 210 a una pluralidad de dispositivos de almacenamiento de datos 230, 231, 232. Los dispositivos de almacenamiento de datos 230, 231, 232 pueden comprender cualquier dispositivo de almacenamiento de datos conocido como, por ejemplo, discos duros magnéticos y dispositivos de estado sólido. Aunque los dispositivos de almacenamiento de datos 230, 231, 232 se muestran como dispositivos diferentes en la FIG. 2A alternativamente, pueden formar diferentes áreas físicas o porciones de almacenamiento dentro de un dispositivo de almacenamiento de datos común. En la FIG. 2A, la pluralidad de dispositivos de almacenamiento de datos 230, 231, 232 almacenan datos de transacciones de entidad 240, datos de transacciones de historial categórico 241 (por ejemplo, una tabla de datos de estado almacenados para pares entidad-categoría, como se describe con más detalle más adelante) y datos auxiliares 242.

En la FIG. 2A, un primer conjunto de dispositivos de almacenamiento de datos 230 almacena datos de transacciones de entidades 240; un segundo conjunto de dispositivos de almacenamiento de datos 231 almacena datos de transacciones de historial categórico 241; y un tercer conjunto de dispositivos de almacenamiento de datos 232 almacena datos auxiliares 242.

Los datos auxiliares 242 pueden comprender uno o más parámetros de modelo para un conjunto de modelos de aprendizaje automático (como, por ejemplo, parámetros entrenados para una arquitectura de red neuronal y/o parámetros de configuración para un modelo de bosque aleatorio) y datos de estado para esos modelos. En un caso, los diferentes conjuntos de datos de transacciones de entidades 240-A a N, datos de transacciones de historial categórico 241-A a N y datos auxiliares 242-A a N, están asociados con diferentes entidades que utilizan de forma segura y colectiva los servicios proporcionados por el sistema de aprendizaje automático 210, por ejemplo, estos pueden representar datos para diferentes bancos que necesitan mantenerse separados como

parte de las condiciones para proporcionar servicios de aprendizaje automático a esas entidades.

La FIG. 2B muestra otra forma en que se pueden almacenar diferentes conjuntos de datos de transacciones de entidades 240-A a N, datos de transacciones de historial categórico 241-A a N y datos auxiliares 242-A a N. En la FIG. 2B el sistema de aprendizaje automático 210 está acoplado comunicativamente, a través del canal de transferencia de datos 250, a al menos un dispositivo de almacenamiento de datos 260. El canal de transferencia de datos 250 puede comprender un bus de almacenamiento local, una red de área de almacenamiento local y/o un acoplamiento de almacenamiento seguro remoto (por ejemplo, superpuesto sobre redes inseguras como Internet). En la FIG. 2B, se proporciona una capa de almacenamiento lógico seguro 270 utilizando el dispositivo de almacenamiento de datos físicos 260. La capa de almacenamiento lógico seguro 270 puede ser un sistema virtualizado que aparece como dispositivos de almacenamiento físico separados para el sistema de aprendizaje automático 210 mientras que en realidad se implementa de forma independiente en al menos el único dispositivo de almacenamiento de datos 260. La capa de almacenamiento lógico 270 puede proporcionar particiones cifradas independientes 280 para datos relacionados con grupos de entidades (por ejemplo, relacionados con diferentes bancos emisores, etc.) y los diferentes datos de transacciones de entidades 240-A a N, los datos de transacciones de historial categórico 241-A a N y los datos auxiliares 242-A a N pueden almacenarse en las particiones correspondientes 280-A a N. En ciertos casos, las entidades pueden crearse dinámicamente a medida que se reciben transacciones para su procesamiento en función de los datos almacenados por uno o más de los sistemas servidor que se muestran en las FIG. 1A a 1C.

Ejemplo de datos de transacción

Las FIG. 3A y 3B muestran ejemplos de datos de transacciones que pueden procesarse mediante un sistema de aprendizaje automático como, por ejemplo, 160 o 210. La FIG. 3A muestra cómo los datos de transacción pueden comprender un conjunto de registros ordenados en el tiempo 300, donde cada registro tiene una marca de tiempo y comprende una pluralidad de campos de transacción. En ciertos casos, los datos de transacciones

pueden agruparse y/o filtrarse según la marca de tiempo. Por ejemplo, la FIG. 3A muestra una partición de los datos de la transacción en datos de la transacción actual 310 que están asociados a una transacción actual y datos de transacciones “más antiguas” o históricas 320 que están dentro de un rango temporal predefinido de la transacción actual. El rango de tiempo se puede establecer como un hiperparámetro de cualquier sistema de aprendizaje automático. Como alternativa los datos de transacciones “más antiguos” o históricos 320 se pueden establecer como un cierto número de transacciones. También son posibles mezclas de ambos enfoques.

La FIG. 3B muestra cómo los datos de transacción 330 para una transacción particular pueden almacenarse en forma numérica para su procesamiento por uno o más modelos de aprendizaje automático. Por ejemplo, en la FIG. 3B, los datos de transacción tienen al menos campos: monto de la transacción, marca de tiempo (por ejemplo, como una época Unix), tipo de transacción (por ejemplo, pago con tarjeta o débito directo), descripción o identificador del producto (es decir, relacionado con los artículos que se compran), identificador del comerciante, identificador del banco emisor, un conjunto de caracteres (por ejemplo, caracteres Unicode dentro de un campo de longitud de caracteres predefinida), identificador de país, etc. Cabe señalar que se puede recibir una amplia variedad de tipos y formatos de datos y procesarlos previamente para obtener representaciones numéricas apropiadas. En ciertos casos, los datos de transacción de origen como, por ejemplo, los generados por un dispositivo cliente y enviados al servidor comercial 130, se procesan previamente para convertir tipos de datos alfanuméricos en tipos de datos numéricos para la aplicación de uno o más modelos de aprendizaje automático. Otros campos presentes en los datos de la transacción pueden incluir, entre otros, un número de cuenta (por ejemplo, un número de tarjeta de crédito), una ubicación donde se realiza la transacción y una manera (por ejemplo, en persona, por teléfono, en un sitio web) en que se ejecuta la transacción.

Como se describe con más detalle a continuación, el sistema de aprendizaje automático 600 de las FIG. 6 y 7 busca identificar novedades en las interacciones entre una entidad (por ejemplo, el titular de la tarjeta) y una o más categorías (por ejemplo, comerciante, tipo de comerciante, monto de la transacción, hora del día, día de la semana, etc.). Los campos de los datos de la transacción se pueden utilizar directamente para identificar la

categoría o categorías relevantes. Adicional o alternativamente, la categoría o categorías asociadas con la transacción pueden inferirse o derivarse de los datos de la transacción. Se observará que dichas categorías derivadas pueden obtenerse de varias maneras. A modo de ejemplo no limitante, esto puede incluir: procesamiento previo simple como la

5 clasificación de una característica numérica; procesamiento previo que combina varias categorías; o un modelo de aprendizaje automático que aprende a componer categorías útiles y se entrena junto con el sistema de aprendizaje automático principal 600.

Ejemplo de sistema de aprendizaje automático

10

La FIG. 4 muestra un ejemplo 400 de un sistema de aprendizaje automático 402 que puede usarse para procesar datos de transacciones. El sistema de aprendizaje automático 402 puede implementar uno o más de los sistemas de aprendizaje automático 160 y 210. El sistema de aprendizaje automático 402 recibe datos de entrada 410. La forma de los datos

15 de entrada 410 puede depender de qué modelo de aprendizaje automático esté siendo aplicado por el sistema de aprendizaje automático 402. En un caso en el que el sistema de aprendizaje automático 402 está configurado para realizar una detección de fraude o anomalía en relación con una transacción, por ejemplo, una transacción en curso como se describió anteriormente, los datos de entrada 410 pueden comprender datos de

20 transacción como, por ejemplo, 330 (es decir, datos que forman parte de un paquete de datos para la transacción) así como datos derivados de datos de transacción de entidad (como, por ejemplo, 300 en la FIG. 3A) y/o datos derivados de datos de historial categórico (como se describe con más detalle a continuación) y/o datos derivados de datos auxiliares (como, por ejemplo, 148 en las FIG. 1A a 1C o 242 en las FIG. 2A y 2B). Los datos auxiliares

25 pueden comprender datos secundarios vinculados a una o más entidades identificadas en los datos primarios asociados con la transacción. Por ejemplo, si los datos de transacción para una transacción en curso identifican a un usuario, un comerciante y uno o más bancos asociados con la transacción (como un banco emisor para el usuario y un banco comerciante), como por ejemplo a través de identificadores únicos presentes en los datos

30 de transacción, entonces los datos auxiliares pueden comprender datos relacionados con estas entidades de transacción. Los datos auxiliares pueden utilizarse para enriquecer la información disponible para las capas de la red neuronal. Los datos auxiliares también pueden comprender datos derivados de registros de actividad como, por ejemplo,

registros de interacción y/o registros de autenticación. En un caso, los datos auxiliares se almacenan en uno o más registros de datos estáticos y se recuperan de estos registros en función de los datos de transacción recibidos. Además, o alternativamente, los datos auxiliares pueden comprender parámetros del modelo de aprendizaje automático que se recuperan en función del contenido de los datos de la transacción. Por ejemplo, los modelos de aprendizaje automático pueden tener parámetros que sean específicos para uno o más de los siguientes: el usuario, el comerciante y el banco emisor, y estos parámetros pueden recuperarse en función de cuál de ellos se identifique en los datos de la transacción. Por ejemplo, uno o más usuarios, comerciantes y bancos emisores pueden tener incrustaciones correspondientes, que pueden comprender representaciones tensoriales recuperables o asignables para dichas entidades. Por ejemplo, cada usuario o comerciante puede tener una representación tensorial (por ejemplo, un vector de punto flotante de tamaño 128-1024) que puede recuperarse de una base de datos u otro almacenamiento de datos o puede generarse mediante una capa de inserción, por ejemplo, basada en un índice de usuario o comerciante.

Los datos de entrada 410 se reciben en una interfaz de datos de entrada 412. La interfaz de datos de entrada 412 puede comprender una interfaz API como, por ejemplo, una interfaz API interna o externa como se describió anteriormente. En un caso, el servidor procesador de pagos 140 como se muestra en las FIG. 1A a 1C realiza una solicitud a esta interfaz, donde la carga útil de la solicitud contiene los datos de la transacción. La interfaz API puede definirse para que sea independiente en cuanto a la forma de los datos de la transacción o su fuente. La interfaz de datos de entrada 412 está acoplada comunicativamente a una plataforma de modelo de aprendizaje automático 414. En un caso, una solicitud realizada a la interfaz de datos de entrada 412 desencadena la ejecución de la plataforma de modelo de aprendizaje automático 414 utilizando los datos de transacción suministrados a la interfaz. La plataforma de modelo de aprendizaje automático 414 está configurada como un entorno de ejecución para la aplicación de uno o más modelos de aprendizaje automático a los datos de entrada 410. En un caso, la plataforma de modelo de aprendizaje automático 414 está dispuesta como un contenedor de ejecución para una pluralidad de diferentes modelos de aprendizaje automático seleccionables. Por ejemplo, un modelo de aprendizaje automático puede definirse utilizando un lenguaje de definición de modelos (por ejemplo, similar a, o utilizando,

lenguajes de marcado como, por ejemplo, eXtended Markup Language - XML). Los lenguajes de definición de modelos pueden incluir (entre otros, de manera independiente o en combinación), SQL, TensorFlow, Caffe, Thinc y PyTorch. En un caso, el lenguaje de definición del modelo comprende un código de programa de computadora que se puede ejecutar para implementar uno o más de los procesos de entrenamiento e inferencia de un modelo de aprendizaje automático definido. Los modelos de aprendizaje automático pueden comprender, por ejemplo, entre otros, arquitecturas de redes neuronales artificiales, modelos de conjunto, modelos de regresión, árboles de decisión como, por ejemplo, bosques aleatorios, modelos de gráficos y redes bayesianas. Más adelante se describe un ejemplo de modelo de aprendizaje automático basado en una red neuronal artificial con referencia a la FIG. 6. La plataforma de modelo de aprendizaje automático 414 puede definir definiciones de entrada y salida comunes (es decir, compartidas) de manera que diferentes modelos de aprendizaje automático se apliquen de manera común (es decir, compartida).

En el presente ejemplo, la plataforma de modelo de aprendizaje automático 414 está configurada para proporcionar al menos una única salida escalar 416. Esto puede normalizarse dentro de un rango predefinido como, por ejemplo, de 0 a 1. Cuando se normaliza, la salida escalar 416 puede verse como una probabilidad de que una transacción asociada con los datos de entrada 410 sea fraudulenta o anómala. En este caso, un valor de "0" puede representar una transacción que coincide con los patrones normales de actividad de uno o más de un usuario, comerciante y banco emisor, mientras que un valor de "1" puede indicar que la transacción es fraudulenta o anómala, es decir, no coincide con los patrones de actividad esperados (aunque los expertos en la materia sabrán que el rango normalizado puede diferir como, por ejemplo, estar invertido o dentro de límites diferentes y tener el mismo efecto funcional). Cabe señalar que, si bien un rango de valores puede definirse como 0 a 1, los valores de salida pueden no estar distribuidos uniformemente dentro de este rango; por ejemplo, un valor de "0,2" puede ser una salida común para un evento "normal" y un valor de "0,8" puede verse como superior a un umbral para un evento "anómalo" o fraudulento típico. El modelo de aprendizaje automático implementado por la plataforma de aprendizaje automático 414 puede así implementar una forma de asignación entre datos de entrada de alta dimensionalidad (por ejemplo, los datos de transacción y cualquier dato auxiliar recuperado) y una salida de

valor único. En un caso, por ejemplo, la plataforma de aprendizaje automático 414 puede estar configurada para recibir datos de entrada para el modelo de aprendizaje automático en un formato numérico, en la que cada modelo de aprendizaje automático definido está configurado para asignar datos de entrada definidos de la misma manera. El modelo de aprendizaje automático exacto que aplica la plataforma de modelo de aprendizaje automático 414 y los parámetros para ese modelo se pueden determinar en función de los datos de configuración. Los datos de configuración pueden estar contenidos dentro de, y/o identificados mediante, los datos de entrada 410 y/o se puede establecer en función de uno o más archivos de configuración que son analizados por la plataforma de aprendizaje automático 414.

En ciertos casos, la plataforma del modelo de aprendizaje automático 414 puede proporcionar resultados adicionales según el contexto. En ciertas implementaciones, la plataforma del modelo de aprendizaje automático 414 puede configurarse para devolver un “código de motivo” que captura una explicación amigable para humanos de la salida de un modelo de aprendizaje automático en términos de atributos de entrada sospechosos. Por ejemplo, la plataforma de modelo de aprendizaje automático 414 puede indicar cuál de uno o más elementos o unidades de entrada dentro de una representación de entrada influyó en la salida del modelo, por ejemplo, una combinación de un canal de “cantidad” que está por encima de un umbral aprendido y un conjunto de elementos o unidades de “comerciante” (como, por ejemplo, una incrustación o índice) que están fuera de un grupo determinado. En los casos en que la plataforma de modelo de aprendizaje automático 414 implementa un árbol de decisiones, estas salidas adicionales pueden comprender una ruta a través del árbol de decisiones o una importancia de característica agregada basada en un conjunto de árboles. Para arquitecturas de redes neuronales, esto puede comprender activaciones de salida de capa y/o filtros de capa con activaciones positivas.

En la FIG. 4, ciertas implementaciones pueden comprender un sistema de alerta opcional 418 que recibe la salida escalar 416. En otras implementaciones, la salida escalar 416 puede pasarse directamente a una interfaz de datos de salida 420 sin procesamiento posterior. En este último caso, la salida escalar 416 puede empaquetarse en una respuesta a una solicitud original a la interfaz de datos de entrada 412. En ambos casos, los datos de salida 422 derivados de la salida escalar 416 se proporcionan como una salida del sistema

de aprendizaje automático 402. Los datos de salida 422 se devuelven para permitir el procesamiento final de los datos de la transacción. Por ejemplo, los datos de salida 422 pueden devolverse al servidor del procesador de pagos 140 y usarse como base de una decisión para aprobar o rechazar la transacción. Dependiendo de los requisitos de implementación, en un caso, el sistema de alerta 418 puede procesar la salida escalar 416 para devolver un valor binario que indica si la transacción debe aprobarse o rechazarse (por ejemplo, “1” equivale a rechazar). En ciertos casos, se puede tomar una decisión aplicando un umbral a la salida escalar 416. Este umbral puede depender del contexto. En ciertos casos, el sistema de alerta 418 y/o la interfaz de datos de salida 420 también pueden recibir entradas adicionales como, por ejemplo, datos explicativos (por ejemplo, el “código de motivo” analizado anteriormente) y/o los datos de entrada originales. La interfaz de datos de salida 420 puede generar un paquete de datos de salida para los datos de salida 422 que combina estas entradas con la salida escalar 416 (por ejemplo, al menos para el registro y/o revisión posterior). De manera similar, una alerta generada por el sistema de alerta 418 puede incluir y/o basarse adicionalmente en las entradas adicionales mencionadas anteriormente, por ejemplo, además de la salida escalar 416.

El sistema de aprendizaje automático 402 se puede utilizar en un modo “en línea” para procesar un gran volumen de transacciones dentro de un rango de tiempo definido estrechamente. Por ejemplo, en condiciones normales de procesamiento, el sistema de aprendizaje automático 402 puede procesar solicitudes en 7 a 12 ms y ser capaz de gestionar entre 1000 y 2000 solicitudes por segundo (estas son restricciones medianas de las condiciones operativas del mundo real). Sin embargo, el sistema de aprendizaje automático 402 también puede utilizarse en un modo “fuera de línea”, por ejemplo, proporcionando una transacción histórica seleccionada a la interfaz de datos de entrada 412. En un modo fuera de línea, los datos de entrada pueden pasarse a las interfaces de datos de entrada en lotes (es decir, grupos). El sistema de aprendizaje automático 402 también puede implementar modelos de aprendizaje automático que proporcionen una salida escalar para una entidad, así como, o en lugar de, una transacción. Por ejemplo, el sistema de aprendizaje automático 402 puede recibir una solicitud asociada con un usuario identificado (por ejemplo, un titular de una tarjeta o cuenta de pago) o un comerciante identificado y estar dispuesto para proporcionar una salida escalar 416 que indique una probabilidad de que el usuario o comerciante sea fraudulento, malicioso o anómalo (es

decir, una amenaza o riesgo general). Por ejemplo, esto puede formar parte de un proceso de monitoreo continuo o periódico, o una solicitud única (por ejemplo, como parte de una solicitud de un servicio). La provisión de una salida escalar para una entidad particular puede basarse en un conjunto de datos de transacción hasta e incluyendo una última

5 transacción aprobada dentro de una secuencia de datos de transacción (por ejemplo, datos de transacción para una entidad similar a la que se muestra en la FIG. 3A).

Ejemplo de flujo de proceso de transacción

10 Las FIG. 5A y 5B muestran dos posibles flujos de proceso de transacción 500 y 550 respectivamente. Estos flujos de procesos pueden tener lugar en el contexto de los sistemas de procesos de transacción de ejemplo 100, 102, 104 que se muestran en las FIG. 1A a 1C, así como otros sistemas. Los flujos de proceso 500 y 550 se proporcionan como un ejemplo de un contexto en el que se puede aplicar un sistema de procesamiento de

15 transacciones de aprendizaje automático. Sin embargo, no todos los flujos de proceso de transacción seguirán necesariamente los procesos que se muestran en las FIG. 5A y 5B y los flujos de procesos pueden cambiar entre implementaciones, sistemas y con el tiempo. Los flujos de proceso de transacción de ejemplo 500 y 550 reflejan dos casos posibles: un primer caso representado por el flujo de proceso de transacción 500 donde se aprueba una

20 transacción, y un segundo caso representado por el flujo de proceso de transacción 550 donde se rechaza una transacción. Cada flujo de proceso de transacción 500, 550 involucra el mismo conjunto de cinco sistemas y dispositivos que interactúan: un POS o dispositivo de usuario 502, un sistema comercial 504, un sistema procesador de pagos (PP) 506, un sistema de aprendizaje automático (ML) 508 y un sistema de banco emisor 510. El

25 dispositivo POS o de usuario 502 puede comprender uno de los dispositivos cliente 110, el sistema comercial 504 puede comprender el servidor comercial 130, el sistema procesador de pagos 506 puede comprender el servidor procesador de pagos 140, y el sistema de aprendizaje automático 508 puede comprender una implementación del sistema de aprendizaje automático 160, 210 y/o 402. El sistema del banco emisor 510 puede

30 comprender uno o más dispositivos servidor que implementan funciones de transacción en nombre de un banco emisor. Los cinco sistemas y dispositivos interactuantes 502 a 510 pueden estar acoplados comunicativamente mediante uno o más canales de comunicación internos o externos como, por ejemplo, redes 120. En ciertos casos, algunos de estos

sistemas pueden combinarse, por ejemplo, un banco emisor también puede actuar como procesador de pagos y, por lo tanto, los sistemas 506 y 510 pueden implementarse con un sistema común. En otros casos, se puede realizar un flujo de proceso similar

5 específicamente para un comerciante (por ejemplo, sin involucrar a un procesador de pagos o un banco emisor). En este caso, el sistema de aprendizaje automático 508 puede comunicarse directamente con el sistema comercial 504. En estas variaciones, el flujo general del proceso de transacción funcional puede seguir siendo similar al que se describe a continuación.

10 El flujo del proceso de transacción en ambas FIG. 5A y 5B comprenden una serie de procesos comunes (es decir, compartidos) 512 a 528. En el bloque 512, el POS o dispositivo de usuario 502 inicia una transacción. Para un dispositivo POS, esto puede comprender un cajero que utiliza un dispositivo frontal para intentar tomar un pago electrónico; para un dispositivo de usuario 502, esto puede comprender un usuario que realiza una compra en

15 línea (por ejemplo, haciendo clic en “completar” dentro de un carrito en línea) utilizando una tarjeta de crédito o débito, o una cuenta de pago en línea. En el bloque 514, los detalles del pago se reciben como datos electrónicos por el sistema comercial 504. En el bloque 516, la transacción es procesada por el sistema comercial 504 y se realiza una solicitud al sistema procesador de pagos 506 para autorizar el pago. En el bloque 518, el

20 sistema procesador de pagos 506 recibe la solicitud del sistema comercial 504. La solicitud puede realizarse a través de un canal de comunicaciones propietario o como una solicitud segura a través de redes públicas (por ejemplo, una solicitud HTTPS a través de Internet). Luego, el sistema procesador de pagos 506 realiza una solicitud al sistema de aprendizaje automático 508 para obtener una puntuación o probabilidad para el uso en el

25 procesamiento de la transacción. El bloque 518 puede comprender además la recuperación de datos auxiliares, para combinarlos con los datos de transacción que se envían como parte de la solicitud al sistema de aprendizaje automático 508. En otros casos, el sistema de aprendizaje automático 508 puede tener acceso a dispositivos de almacenamiento de datos que almacenan datos auxiliares (por ejemplo, similares a las

30 configuraciones de las FIG. 2A y 2B) y así recuperar estos datos como parte de operaciones internas (por ejemplo, en función de identificadores proporcionados dentro de los datos de transacción y/o como se define como parte de un modelo de aprendizaje automático implementado).

El bloque 520 muestra una operación de inicialización del modelo que ocurre antes de cualquier solicitud del sistema procesador de pagos 506. Por ejemplo, la operación de inicialización del modelo puede comprender la carga de un modelo de aprendizaje

5 automático definido y parámetros que instancian el modelo de aprendizaje automático definido. En el bloque 522, el sistema de aprendizaje automático 508 recibe la solicitud del sistema procesador de pagos 506 (por ejemplo, a través de una interfaz de entrada de datos como, por ejemplo, 412 en la FIG. 4). En el bloque 522, el sistema de aprendizaje automático 508 puede realizar cualquier procesamiento previo definido antes de la

10 aplicación del modelo de aprendizaje automático inicializado en el bloque 520. Por ejemplo, en el caso de que los datos de transacción aún conserven datos de caracteres como, por ejemplo, un comerciante identificado por una cadena de caracteres o una descripción de transacción de caracteres, estos pueden convertirse en datos numéricos estructurados adecuados (por ejemplo, convirtiendo datos categóricos de cadena en un

15 identificador a través de una operación de búsqueda u otra asignación, y/o asignando caracteres o grupos de caracteres a incrustaciones vectoriales). Luego, en el bloque 524, el sistema de aprendizaje automático 506 aplica el modelo de aprendizaje automático instanciado, suministrando al modelo datos de entrada derivados de la solicitud recibida. Esto puede comprender la aplicación de la plataforma de modelo de aprendizaje

20 automático 414 como se describe con referencia a la FIG. 4. En el bloque 526, el modelo de aprendizaje automático instanciado genera una salida escalar. Esto puede procesarse para determinar una decisión binaria de “aprobación” o “rechazo” en el sistema de aprendizaje automático 508 o, en un caso preferido, se devuelve al sistema procesador de pagos 506 como respuesta a la solicitud realizada en el bloque 518.

25

En el bloque 528, la salida del sistema de aprendizaje automático 508 es recibida por el sistema procesador de pagos 506 y se utiliza para aprobar o rechazar la transacción. La FIG. 5A muestra un proceso donde la transacción se aprueba en función de la salida del sistema de aprendizaje automático 508; la FIG. 5B muestra un proceso en el que se rechaza la

30 transacción en función del resultado del sistema de aprendizaje automático 508. En la FIG. 5A, en el bloque 528, se aprueba la transacción. Luego, en el bloque 530, se realiza una solicitud al sistema bancario emisor 532. En el bloque 534, el sistema bancario emisor 532 aprueba o rechaza la solicitud. Por ejemplo, el sistema bancario emisor 532 puede aprobar

la solicitud si un usuario final o titular de una tarjeta tiene fondos suficientes y aprobación para cubrir el costo de la transacción. En ciertos casos, el sistema del banco emisor 532 puede aplicar un segundo nivel de seguridad; sin embargo, esto puede no ser necesario si el banco emisor confía en la detección de anomalías realizada por el procesador de pagos
5 utilizando el sistema de aprendizaje automático 508. En el bloque 536, la autorización del sistema bancario emisor 510 se devuelve al sistema procesador de pagos 506, que a su vez envía una respuesta al sistema comercial 504 en el bloque 538, y el sistema comercial 504 a su vez responde al POS o dispositivo de usuario 502 en el bloque 540. Si el sistema bancario emisor 510 aprueba la transacción en el bloque 534, entonces la transacción
10 puede completarse y se puede devolver una respuesta positiva a través del sistema comercial 504 al POS o dispositivo de usuario 502. El usuario final puede experimentar esto como un mensaje “autorizado” en la pantalla del POS o del dispositivo del usuario 502. El sistema comercial 504 puede luego completar la compra (por ejemplo, iniciar el procesamiento interno para completar la compra).

15

En un momento posterior, uno o más del sistema comercial 504 y el sistema de aprendizaje automático 508 pueden guardar datos relacionados con la transacción, por ejemplo, como parte de los datos de transacción 146, 240 o 300 y/o datos de historial categórico 241 (como se analizará con más detalle a continuación). Esto se muestra en los
20 bloques discontinuos 542 y 544. Los datos de transacción y/o los datos del historial categórico se pueden guardar junto con una o más de las salidas del sistema de aprendizaje automático 508 (por ejemplo, la probabilidad de fraude o anomalía escalar) y un resultado final de la transacción (por ejemplo, si fue aprobada o rechazada). Los datos guardados pueden almacenarse para usarse como datos de entrenamiento para los
25 modelos de aprendizaje automático implementados por el sistema de aprendizaje automático 508 (por ejemplo, como base para los datos de entrenamiento que se muestran en una o más de las FIG. 8 y 10). También se puede acceder a los datos guardados como parte de futuras iteraciones del bloque 524, por ejemplo, pueden formar parte de futuros datos auxiliares. En ciertos casos, es posible que no se conozca el
30 resultado final de la transacción en el momento de realizarla. Por ejemplo, una transacción solo puede etiquetarse como anómala mediante una revisión posterior por parte de un analista y/o un sistema automatizado, o en función de la retroalimentación de un usuario (por ejemplo, cuando el usuario informa un fraude o indica que una tarjeta de pago o una

cuenta se vieron comprometidas a partir de una fecha determinada). En estos casos, las etiquetas de base real con el propósito de entrenar el sistema de aprendizaje automático 508 se pueden recopilar a lo largo del tiempo después de la transacción misma.

- 5 Pasando ahora al flujo de proceso alternativo de la FIG. 5B, en este caso uno o más del sistema de aprendizaje automático 508 y el sistema procesador de pagos 506 rechazan la transacción basándose en la salida del sistema de aprendizaje automático 508. Por ejemplo, una transacción puede rechazarse si la salida escalar del sistema de aprendizaje automático 508 está por encima de un umbral recuperado. En el bloque 552, el sistema
- 10 procesador de pagos 506 emite una respuesta al sistema comercial 504, que se recibe en el bloque 554. En el bloque 554, el sistema comercial 504 toma medidas para evitar que la transacción se complete y devuelve una respuesta apropiada al POS o al dispositivo del usuario 502. Esta respuesta se recibe en el bloque 556 y se puede informar al usuario final o cliente que su pago ha sido rechazado, por ejemplo, mediante un mensaje de
- 15 “Rechazado” en la pantalla. Es posible que se le solicite al usuario final o al cliente que utilice un método de pago diferente. Aunque no se muestra en la FIG. 5B, en ciertos casos, el sistema bancario emisor 510 puede ser informado de que se ha rechazado una transacción relacionada con un titular de cuenta en particular. El sistema del banco emisor 510 puede ser informado como parte del proceso que se muestra en la FIG. 5B o puede
- 20 informarse como parte de una actualización periódica (por ejemplo, diaria). Si bien es posible que la transacción no pase a formar parte de los datos de transacción 146, 240 o 300 (ya que no fue aprobada), aún puede ser registrada al menos por el sistema de aprendizaje automático 508 como se indica en el bloque 544. Por ejemplo, como en la FIG. 5A, los datos de la transacción se pueden guardar junto con la salida del sistema de
- 25 aprendizaje automático 508 (por ejemplo, la probabilidad de fraude o anomalía escalar) y un resultado final de la transacción (por ejemplo, que fue rechazada).

Sistema de aprendizaje automático con módulo de historial categórico

- 30 Ciertos ejemplos descritos en el presente documento como, por ejemplo, los sistemas de aprendizaje automático 160, 210, 402 y 508 en las FIG. 1A a 1C, 2A y 2B, FIG. 4 y las FIG. 5A y 5B, pueden implementarse como una plataforma modular que permita utilizar diferentes modelos y configuraciones de aprendizaje automático para proporcionar el procesamiento

de transacciones descrito en el presente documento. Esta plataforma modular puede permitir el uso de diferentes modelos y configuraciones de aprendizaje automático a medida que las tecnologías mejoran y/o en función de características particulares de los datos disponibles.

5

Un sistema de aprendizaje automático 600 de acuerdo con una modalidad de la presente invención se muestra en las FIG. 6 y 7, como se describe con más detalle a continuación.

Como se analizó anteriormente, debe notarse que, si bien en la modalidad particular
10 descrita a continuación los datos de estado se almacenan en una tabla de vectores, los datos de estado podrían estructurarse fácilmente como tensores n-dimensionales y/o pueden almacenarse en alguna otra estructura o formato de datos, por ejemplo, en una matriz multidimensional. De manera similar, los diversos vectores (por ejemplo, vector de entrada, vector de salida, etc.) descritos con respecto a esta modalidad también pueden
15 ser tensores n-dimensionales y pueden almacenarse utilizando cualquier estructura de datos apropiada.

El sistema de aprendizaje automático 600 recibe datos de entrada 601 y los asigna a una salida escalar 602. Este procesamiento general sigue el mismo marco descrito
20 anteriormente. El sistema de aprendizaje automático 600 comprende una primera etapa de procesamiento (o “modelo inferior”) 603 y una segunda etapa de procesamiento (o “modelo superior”) 604.

El sistema de aprendizaje automático 600 se aplica a los datos asociados con al menos una
25 transacción propuesta para generar una salida escalar 602 para la transacción propuesta. La salida escalar 602 representa una probabilidad de que la transacción propuesta presente una anomalía de comportamiento, por ejemplo, que la transacción propuesta incorpore un patrón de acciones o eventos que difiera de los patrones esperados o típicos de acciones o eventos. En ciertos casos, la salida escalar 602 representa una probabilidad
30 de que la transacción propuesta presente una anomalía en una secuencia de acciones, donde las acciones incluyen al menos transacciones anteriores para esa entidad y el par entidad-categoría, como se describe con más detalle a continuación. La salida escalar 602 se puede utilizar para completar una decisión de aprobación para la transacción propuesta,

por ejemplo, como se describe con referencia a las FIG. 5A y 5B, pueden utilizarse para decidir si se aprueba o rechaza la transacción propuesta.

Los datos de entrada pueden incluir datos de tiempo de transacción y datos de características de transacción. Los datos de tiempo de transacción pueden comprender datos derivados de una marca de tiempo como, por ejemplo, la que se muestra en la FIG. 3B o cualquier otro formato de datos que represente una fecha y/o hora de una transacción. La fecha y/o hora de la transacción se puede establecer como una hora en la que se inició la transacción en un dispositivo informático cliente como, por ejemplo, 110 o 502, o se puede establecer como una hora en la que se recibió una solicitud en el sistema de aprendizaje automático 600, por ejemplo, similar a la hora de una solicitud recibida en el bloque 522 en las FIG. 5A y 5B. Los datos de tiempo de transacción pueden comprender datos de tiempo para una pluralidad de transacciones como, por ejemplo, la transacción propuesta actual y un conjunto histórico de una o más transacciones anteriores. Los datos de tiempo del conjunto histórico de una o más transacciones anteriores pueden recibirse con una solicitud y/o recuperarse de un dispositivo de almacenamiento acoplado comunicativamente al sistema de aprendizaje automático 600.

En la disposición de la FIG. 6, los datos de tiempo de transacción se convierten en datos de tiempo relativo para la aplicación de una o más arquitecturas de redes neuronales. En particular, los datos de tiempo de transacción se convierten en un conjunto de valores de diferencia de tiempo, donde los valores de diferencia de tiempo representan una diferencia de tiempo Δt entre la transacción propuesta actual y una transacción anterior (generalmente, la transacción anterior más reciente). Por ejemplo, la diferencia horaria puede comprender una diferencia horaria normalizada en segundos, minutos u horas. La diferencia de tiempo Δt se puede calcular restando una marca de tiempo de la transacción anterior (o transacciones, donde se calculan y utilizan varias diferencias) de la marca de tiempo de la transacción propuesta. Los valores de diferencia de tiempo se pueden normalizar dividiéndolos por una diferencia de tiempo máxima predefinida y/o recortándolos en un valor de diferencia de tiempo máxima.

En un modo de inferencia, el sistema de aprendizaje automático 600 detecta conjuntos de características de los datos de entrada 601 utilizando los modelos superior e inferior 603,

604 como se describe con más detalle a continuación. La salida del modelo superior 604 es el valor escalar 602, que se utiliza para determinar si se debe bloquear o no la transacción (por ejemplo, comparando el valor escalar 602 con un umbral).

- 5 Como se muestra en la FIG. 6, el sistema de aprendizaje automático 600 comprende una capa de incrustación basada en una red neuronal de propagación hacia adelante dentro del modelo inferior 603. En esta modalidad particular, la capa de incrustación es un primer perceptrón multicapa 610, sin embargo, se observará que este podría ser en cambio cualquier otro tipo de red neuronal de propagación hacia adelante como, por ejemplo, una
- 10 capa de transformador o una unidad lógica aritmética neuronal.

- El primer perceptrón multicapa 610 comprende una arquitectura de red neuronal completamente conectada que comprende una pluralidad de capas de red neuronal (por ejemplo, de 1 a 10 capas) para procesar previamente los datos para la transacción
- 15 propuesta desde la entrada 601. Aunque aquí se describe un perceptrón multicapa, en ciertas implementaciones se puede omitir el procesamiento previo (por ejemplo, si los datos de transacción recibidos en los datos de entrada 601 ya están en un formato adecuado) y/o solo se puede asignar una única capa de asignación lineal. En ciertos ejemplos, el procesamiento previo puede verse como una forma de “incrustación” o capa
- 20 de “asignación inicial” para los datos de transacción de entrada. En ciertos casos, una o más de las capas de red neuronal del primer perceptrón multicapa 610 pueden proporcionar escalamiento aprendido y/o normalización de los datos de transacción recibidos. En general, la arquitectura de red neuronal completamente conectada del primer perceptrón multicapa 610 representa una primera etapa de procesamiento previo
- 25 de características aprendidas, que convierte los datos de entrada asociados con la transacción propuesta en un vector de características para su posterior procesamiento. En ciertos casos, la arquitectura de red neuronal completamente conectada puede aprender ciertas relaciones entre los elementos de los datos de características de la transacción, por ejemplo, ciertas correlaciones, y generar una representación eficiente que tenga en cuenta
- 30 estas correlaciones. En un caso, los datos de transacción de entrada pueden comprender números enteros y/o de punto flotante y la salida 612 del primer perceptrón multicapa 610 puede comprender un vector de valores entre 0 y 1. El número de elementos o unidades (es decir, el tamaño del vector de salida) del primer perceptrón multicapa 610 se puede

establecer como un hiperparámetro configurable. En ciertos casos, el número de elementos o unidades puede estar entre 32 y 2048.

5 La salida 612 producida por el primer perceptrón multicapa 610 se divide en dos porciones 612a, 612b que se proporcionan a una celda de deterioro temporal 614 y a un módulo de historial categórico 616 respectivamente. Sin embargo, se observará que, si bien estos se muestran como totalmente separables, se prevén arreglos en los que hay al menos cierta superposición en las partes de la salida 612 que se alimentan a cada una de las celdas de deterioro temporal 614 y el módulo de historial categórico 616.

10

Se debe tener en cuenta que los datos de transacción de entrada de la entrada 601 pueden comprender datos de la transacción propuesta (por ejemplo, tal como se reciben a través de uno o más de un dispositivo cliente, un dispositivo POS, un dispositivo servidor del comerciante y un dispositivo servidor del procesador de pagos) y datos asociados con la transacción propuesta que no están contenidos en paquetes de datos asociados con la transacción propuesta. Por ejemplo, como se explicó con referencia a las FIG. 1A a 1C y 2A y 2B, la entrada 601 puede comprender además datos auxiliares (como, por ejemplo, 148 y 242), donde los datos auxiliares son recuperados por el sistema de aprendizaje automático 600 (como, por ejemplo, se muestra en la FIG. 1C) y/o por el servidor procesador de pagos 140 (como, por ejemplo, se muestra en las FIG. 1A o 1B). El contenido exacto que se incluye en los datos de entrada 601 puede variar entre implementaciones. Los presentes ejemplos se refieren a una arquitectura técnica general para el procesamiento de datos de transacciones, más que a la forma exacta de dichos datos. En general, como el sistema de aprendizaje automático 600 comprende un conjunto de capas de red neuronal, los parámetros pueden aprenderse en función de cualquier configuración de datos de entrada deseada o en función de los datos de entrada que estén disponibles. Los presentes ejemplos tratan del diseño de ingeniería de dicha arquitectura técnica para permitir el procesamiento de transacciones a la velocidad y escala descritas en el presente documento.

30

En el ejemplo de la FIG. 6, la salida del primer perceptrón multicapa 610 es recibida por la celda de deterioro temporal 614. La celda de deterioro temporal 614 está configurada para generar datos de salida o_t basados en datos de estado previos h_{t-1} , la diferencia de tiempo

Δt y la porción de la salida 612a suministrada por el primer perceptrón multicapa 610. La celda de deterioro temporal 614 puede implementar las técnicas descritas en WO/2022/008130 y WO/2022/008131, cada una de las cuales se incorpora en el presente documento como referencia.

5

Los datos del estado anterior h_{t-1} son un vector de valores correspondiente al estado del sistema de la transacción anterior. Los expertos en la materia observarán que, en general, los datos de estado anteriores h_{t-1} se almacenan en la memoria y proporciona un historial de los patrones de transacciones que han tenido lugar hasta ahora. Los datos de estado se
10 almacenan para cada entidad principal (por ejemplo, el titular de la tarjeta) y, por lo tanto, los datos de estado de una entidad principal determinada proporcionan información sobre la cadena de transacciones anterior para esa entidad principal.

El grado de influencia que los datos históricos del estado deberían tener en una decisión
15 relacionada con una transacción propuesta, puede depender de la diferencia temporal Δt . Si ha transcurrido un largo período de tiempo desde la transacción anterior, algunos datos estatales pueden no ser tan relevantes a la hora de evaluar si la transacción propuesta es anómala o fraudulenta. La celda de deterioro temporal 614 aplica una función de deterioro a los datos de estado anteriores h_{t-1} basado en la diferencia de tiempo Δt .

20

Una vez que se hayan obtenido los datos del estado anterior h_{t-1} se ha deteriorado en función de la diferencia de tiempo Δt , la celda de deterioro temporal 614 realiza una asignación para generar datos de salida o_t . La celda de deterioro temporal 614 también almacena los nuevos datos de estado h_t en memoria para la próxima transacción.

25

El proceso de asignación llevado a cabo por la celda de deterioro temporal 614 puede basarse en aprendizaje automático. Por ejemplo, la celda de deterioro temporal 614 puede usar una asignación de red neuronal (por ejemplo, una o más funciones parametrizadas) para asignar los datos de estado anteriores h_{t-1} y la salida 612a del primer perceptrón
30 multicapa 610 a una salida vectorial de tamaño fijo de un tamaño predefinido. Se observará, sin embargo, que se pueden utilizar otros métodos que no requieren ninguna red neuronal ni ninguna otra técnica de aprendizaje automático.

Una parte 612b de la salida del primer perceptrón multicapa 614 se introduce en el módulo de historial categórico 616, y es utilizada específicamente por una capa de actualización 619. A diferencia de la celda de deterioro temporal 614 que opera utilizando un único vector de datos de estado para cada entidad, el módulo de historial categórico 616 construye una tabla (por ejemplo, una tabla de búsqueda) de varios estados, uno para cada categoría secundaria con la que la entidad primaria ha interactuado, como se puede entender mejor con referencia a la FIG. 7.

En aplicaciones típicas de fraude, la entidad principal representa al pagador o titular de la cuenta, mientras que una categoría secundaria puede representar al beneficiario de los fondos, por ejemplo, un comerciante. Sin embargo, debe entenderse que la categoría secundaria puede ser cualquier campo categórico en los datos. Podría ser, por ejemplo, un campo numérico o de fecha y hora que se haya agrupado en diferentes categorías; por ejemplo, "cantidad" podría agruparse en las categorías "baja", "media" y "alta"; las fechas y horas podrían agruparse en horas del día o días de la semana, etc. Las categorías también pueden ser una combinación de varias categorías (por ejemplo, la identidad del comerciante concatenada con un día de la semana). Las categorías no necesitan ser ordinales. De ahora en adelante, en este ejemplo específico, nos referiremos al campo categórico secundario como "comerciante" para mayor claridad en la notación. Se debe tener en cuenta que todos los estados seguirán asociados con la entidad principal, de manera que los datos de estado se almacenen y recuperen para un par entidad-categoría en particular. Se entenderá que, en general, habrá una tabla de este tipo para cada entidad primaria.

En una implementación ejemplar basada en software, esta tabla es un tensor de forma 2D (*merchant_state_size*, *max_num_merchant_state*) cuyas columnas representan estados individuales para diferentes comerciantes (se observará, por supuesto, que se podrían utilizar diferentes categorías secundarias en otros ejemplos). Se entenderá que '*max_num_merchant_state*' es el número máximo de comerciantes (o equivalentemente, otros valores de categoría como, por ejemplo, día de la semana, monto o banda de transacción, etc.) para los que se almacenan datos de estado, es decir, el número de columnas en la tabla. Mientras tanto, '*merchant_state_size*' es el número de valores independientes almacenados en cada columna, es decir, el número de filas de la tabla.

Como se mencionó anteriormente, se debe entender que generalmente habrá una tabla de datos de estado para cada entidad principal (es decir, para cada titular de cuenta), de manera que los datos del comerciante almacenados (es decir, la categoría secundaria) sean específicos de esa entidad principal.

5

La tabla de este ejemplo tiene un tamaño fijo y, por lo tanto, tiene un límite en la cantidad de estados comerciales que puede almacenar. Como entrada toma la tabla de búsqueda G_{t-1} (para esta entidad) del paso de tiempo anterior $t-1$ y un vector de actualización de tamaño *merchant_state_size*. Como salida, devuelve un vector que contiene el estado del comerciante actualizado. Este vector puede, en algunas modalidades, incluir también el estado de comerciante deteriorado, por ejemplo, proporcionando un vector que es una concatenación del estado de comerciante deteriorado y un estado de comerciante actualizado (para el comerciante actual), es decir, de tamaño $2 * \text{merchant_state_size}$.

10

Como se muestra en la FIG. 7, para cada transacción el módulo de historial categórico 616 realiza una lógica de deterioro y actualización, de manera similar a la celda de deterioro temporal estándar 614. El módulo de historial categórico 616 primero aplica un deterioro de tiempo exponencial en todos los estados almacenados en la tabla. G_{t-1} , donde el deterioro se basa en la diferencia de tiempo Δt desde la última transacción.

20

La función de deterioro permite un uso selectivo y ajustado de datos de transacciones anteriores (es decir, puede aplicar un grado particular de “olvido”) en función de la diferencia de tiempo Δt entre la transacción propuesta y una transacción anterior, donde los datos de la transacción anterior son una función agregada de vectores de características de transacción para una pluralidad de transacciones anteriores.

25

La función de deterioro se puede aplicar como una función de deterioro temporal exponencial vectorizada de la forma $f(s) = [e^{(-s/w_1)}, e^{(-s/w_2)}, \dots, e^{(-s/w_d)}]$, es decir, como una serie de cálculos de deterioro temporal parametrizados con $[w_1, w_2 \dots w_d]$ que comprenden un conjunto de pesos que representan una “longitud de deterioro” en segundos y donde d es el tamaño del vector de estado recurrente (generalmente puede ser igual a la longitud del vector de datos de estado en el que opera la función de deterioro). En este caso, Δt se pasa a la función como s para generar un vector de longitud

30

d , por ejemplo, se puede pasar como un valor flotante escalar que representa la cantidad de segundos transcurridos entre el evento anterior (por ejemplo, transacción) para la misma entidad y el evento actual (por ejemplo, la transacción propuesta). Los factores de deterioro (es decir, los pesos) pueden aprenderse durante un proceso de entrenamiento.

5

Los pesos de deterioro pueden ser parámetros fijos (por ejemplo, configurables manualmente) y/o pueden comprender parámetros entrenables. En el caso de una función exponencial, las longitudes de deterioro pueden incluir (o estar entrenadas para incluir) una mezcla de diferentes valores que representan deterioros desde unos pocos minutos

10 hasta unas pocas semanas, de manera que se capture tanto el comportamiento a corto como a largo plazo. La función de deterioro temporal se aplica a la tabla de datos de estado de la iteración anterior utilizando una multiplicación elemento por elemento (por ejemplo, se calcula el producto Hadamard) entre el vector de deterioro temporal y cada entrada de datos de estado (los datos de estado almacenados para cada categoría, es

15 decir, columna de la tabla). En este ejemplo, los datos de estado comprenden una tabla bidimensional de vectores, cada uno de los cuales tiene la misma longitud que la salida del vector de características de la transacción. Esto controla efectivamente cuánto de los datos de estado de la iteración anterior se recuerda y cuánto se olvida.

20 Se puede utilizar un rango de constantes de deterioro temporal para deteriorar la tabla. Por ejemplo, un deterioro en una escala de tiempo corta suprimirá las contribuciones de memoria de la actividad anterior, mientras que un deterioro en una escala de tiempo larga retendrá las contribuciones de memoria de la actividad anterior. Esto permite que las redes comparen actividades nuevas y antiguas para determinar la novedad. Además, las

25 actualizaciones de estado son una función de los atributos de entrada de los eventos. Esto significa que algunos elementos de estado solo pueden actualizarse para ciertos tipos de eventos. Esto se ilustra gráficamente en la FIG. 7 según las diferentes agrupaciones de filas en la tabla, donde las tres filas superiores se retienen durante un período de tiempo relativamente largo, las tres filas del medio se retienen durante un período de tiempo

30 intermedio y las tres filas inferiores se retienen durante un período de tiempo relativamente corto. Se entenderá que esto es sólo un ejemplo, sin embargo, pueden aplicarse diferentes períodos de tiempo a cada parte de los datos estatales según corresponda.

La red puede aprender que solo se ha visto antes actividad de un tipo particular (por ejemplo, transacciones de bajo valor) con un comerciante determinado y, por lo tanto, que el evento actual es novedoso incluso si el comerciante ya ha estado presente antes. Por ejemplo, podría aprender a ignorar la presencia de transacciones de bajo valor en el pasado con este comerciante al evaluar el riesgo de una actividad actual de alto valor. Los deterioros reales aplicadas pueden aprenderse o pueden predeterminarse (y pueden ser definidas por el usuario, por ejemplo, durante la configuración inicial o la optimización del sistema). En cualquier caso, el sistema de aprendizaje automático puede aprender qué pesos de deterioro deben aplicarse a qué combinaciones de características de comportamiento.

Si bien en esta modalidad ejemplar se deteriora toda la tabla para una entidad dada, se prevén otras modalidades en las que solo se deteriora la columna seleccionada actualmente. Para lograr esto, se puede almacenar una marca de tiempo para cada par entidad-categoría y la diferencia de tiempo Δt se toma como el tiempo entre la transacción actual y el momento en el que se almacenaron los datos de la instancia más reciente de ese par entidad-categoría en la tabla.

Después del paso de deterioro, el módulo de historial categórico 616 extrae la columna de estado del comerciante que coincide con el merchantId actual (o el vector cero si es un nuevo comerciante, como se describe a continuación). El módulo de historial categórico 616 consulta la tabla de búsqueda con el merchantId actual para ver si la tabla contiene datos de estado (es decir, si existe una columna) para este comerciante. Si es así, los datos de estado almacenados en relación con ese merchantId se recuperan de la tabla. Por el contrario, si no existe en la tabla, se puede agregar un nuevo estado “predeterminado” (por ejemplo, un vector cero) a la tabla.

Los datos de estado (recuperados o nuevos según corresponda) se actualizan luego mediante la capa de actualización 619, utilizando la porción de salida 612b del perceptrón multicapa 610 (que a su vez se deriva del vector de entrada x_t como se describió anteriormente) y se insertan nuevamente en la tabla. Si bien se pueden adoptar diferentes enfoques para el paso de actualización (es decir, para los pasos llevados a cabo por la capa

de actualización 619), en una implementación, el vector de datos de estado deteriorado para el par entidad-categoría y el vector de entrada del primer perceptrón multicapa 610 (es decir, la porción de salida 612b) se suman entre sí mediante la adición de elementos. Esto da como resultado un vector de salida y_t .

5

La tabla se administra como una pila: las columnas actualizadas se insertan al principio de la tabla y la columna actualizada menos recientemente se encuentra al final de la tabla. Debido a los límites de tamaño de la tabla, si la tabla está llena, se agregará un nuevo estado de comerciante descartando un estado del comerciante visto menos

10 recientemente. Como se puede ver en la FIG. 7, los datos de estado actualizados para esta transacción se insertan al frente de la tabla y el “espacio” dejado atrás (como se ilustra con la línea discontinua) se cierra.

Las salidas o_t , y_t de la celda de deterioro temporal 614 y del módulo de historial categórico 15 616 son recibidas por el modelo superior 604, y específicamente por una segunda capa de salida de red neuronal de propagación hacia adelante. En esta modalidad particular, la capa de salida es un segundo perceptrón multicapa 620, sin embargo, se observará que éste podría ser cualquier otro tipo de red neuronal de propagación hacia adelante como, por ejemplo, una capa de transformador o una unidad lógica aritmética neuronal. En esta 20 modalidad particular, los datos de entrada 601 también se alimentan al segundo perceptrón multicapa 620. En este ejemplo, las salidas o_t , y_t de la celda de deterioro temporal 614 y del módulo de historial categórico 616 se pasan directamente al segundo perceptrón multicapa 620. Sin embargo, se prevén otras modalidades (no mostradas) en las que este paso se realiza indirectamente a través de una o más capas de red neuronal 25 intervinientes adicionales.

El segundo perceptrón multicapa 620 se utiliza para generar la salida escalar 602. El segundo perceptrón multijugador 620 comprende una arquitectura de red neuronal completamente conectada que comprende a su vez una pluralidad de capas de red 30 neuronal para recibir las salidas o_t , y_t de la celda de deterioro temporal 614 y del módulo de historial categórico 616, así como de los datos de entrada 601, y para asignar esas salidas o_t , y_t al valor escalar 602. Esto puede comprender una serie de capas que reducen la dimensionalidad. Una última función de activación en la pluralidad de capas de red

neuronal puede comprender una función de activación sigmoidea para asignar una salida al rango de 0 a 1. El segundo perceptrón multijugador 620 puede ser entrenado para extraer correlaciones entre características que son emitidas por un mecanismo de atención (por ejemplo, que puede utilizar una pluralidad de cabezales de atención de una manera conocida en la materia *per se*) y aplicar una o más funciones no lineales para generar finalmente el valor escalar 602.

El valor escalar 602 puede luego ser utilizado por otra parte del canal de procesamiento de transacciones (no mostrado) al decidir si se debe permitir que la transacción continúe normalmente o si en cambio se debe detener y marcar como fraudulenta o anómala. Por ejemplo, el sistema puede comparar el valor escalar 602 con un umbral adecuado, de manera que una puntuación que exceda el umbral se marque como anómala.

En una implementación preferida, los datos de estado dependen de la entidad, es decir, son específicos de un usuario, titular de cuenta o cuenta comercial en particular. De esta manera, la entidad apropiada puede identificarse como parte del procesamiento previo de datos de transacción y el sistema de aprendizaje automático 600 puede configurarse para esa entidad. En un caso, el sistema de aprendizaje automático 600 puede aplicar los mismos parámetros para las arquitecturas de red neuronal para cada entidad, pero puede almacenar datos de estado por separado y solo puede recuperar datos de transacciones históricas y/o datos auxiliares que estén asociados con (por ejemplo, indexados por) esa entidad. Como tal, el sistema de aprendizaje automático 600 puede comprender un almacén de estados de entidad, por ejemplo, como se describe con referencia a las FIG. 2A y 2B, mediante las cuales se pueden dividir de manera efectiva los datos de diferentes entidades. En otros casos, los parámetros de la función de deterioro pueden compartirse entre varias entidades. Esto puede resultar ventajoso para reducir la cantidad de parámetros que se pueden aprender para cada entidad de todo el sistema de aprendizaje automático. Por ejemplo, los pesos de deterioro temporal exponencial descritos anteriormente pueden ser fijos (o aprendidos) y compartidos entre varias entidades, mientras que los datos de estado pueden ser únicos para cada entidad.

Entrenamiento de un sistema de aprendizaje automático para procesar datos de transacciones

- En ciertos ejemplos, los sistemas de aprendizaje automático como, por ejemplo, los
- 5 descritos en el presente documento, pueden entrenarse utilizando datos de entrenamiento etiquetados. Por ejemplo, se puede proporcionar un conjunto de entrenamiento que comprenda datos asociados con transacciones que se hayan etiquetado como “normales” o “fraudulentas”. En un caso, estas etiquetas pueden proporcionarse en función de la actividad fraudulenta denunciada, es decir, utilizando
- 10 informes anteriores de fraude. Por ejemplo, los datos asociados a transacciones que han sido aprobadas y procesadas sin informes posteriores de fraude pueden etiquetarse como “0” o “normal”; mientras que los datos asociados a transacciones que han sido rechazadas y/o transacciones que luego han sido informadas como fraudulentas y de otra manera marcadas o bloqueadas pueden etiquetarse como “1” o anómalas.
- 15
- Se puede llevar a cabo un proceso de entrenamiento de extremo a extremo, de manera que el sistema de aprendizaje automático pueda aprender valores apropiados para todos y cada uno de los parámetros que se puedan aprender. Esto puede incluir, por ejemplo, el entrenamiento de los parámetros de cualquier perceptrón multicapa y/o los pesos de
- 20 deterioro para las funciones de deterioro. Se observará que cualquier algoritmo de entrenamiento adecuado, conocido en la materia *per se*, pueden utilizarse y los principios de la presente invención no están limitados a ningún algoritmo de entrenamiento específico. Sin embargo, a modo de ejemplo, un algoritmo de entrenamiento adecuado puede basarse en el descenso de gradiente estocástico y puede utilizar la retropropagación
- 25 del error en la predicción del modelo para mejorar iterativamente los parámetros del modelo con el fin de encontrar aquellos que minimicen la función objetivo. En un caso, la función objetivo (que mide el error en la predicción del modelo) puede definirse como la entropía cruzada binaria entre el valor predicho 602 y la etiqueta de base real ('0' o '1').
- 30 Los expertos en la materia observarán que cuando se hace referencia a las distintas filas y columnas de la tabla de datos de estado, los roles respectivos de éstas podrían intercambiarse de manera que los datos de estado se almacenen en filas indexadas por el

identificador de categoría con los valores de datos de estado, para las distintas características, almacenados en las columnas en su lugar.

Se observará que las modalidades de la presente invención pueden proporcionar una
5 disposición en la que el módulo de historial categórico incorpora la extracción de señales del monitoreo de interacción a la arquitectura. Ventajosamente, esto puede conducir a mejoras en la velocidad y/o confidencialidad de la implementación. En concreto, los científicos de datos ya no necesitan diseñar manualmente características de este tipo, lo que significa que no es necesario divulgarlas para la gestión del modelo como entradas al
10 clasificador, y el tiempo de implementación de los modelos se reduce.

Además, las modalidades de la presente invención pueden proporcionar mejoras en el rendimiento de detección. El módulo de historial categórico puede aprender una lógica de interacción más sutil que la que los seres humanos son capaces de diseñar a mano en un
15 tiempo finito. Específicamente, el módulo de historial categórico puede aprender patrones de actividad del usuario específicos de cada categoría y detectar anomalías basadas en datos históricos para el par entidad-categoría relevante. Este enfoque puede proporcionar mejoras de detección significativas en relación con otros enfoques conocidos en la materia
per se.

20

Si bien se han descrito en detalle modalidades específicas de la presente invención, los expertos en la materia observarán que las modalidades descritas en detalle no limitan el alcance de la invención reivindicada.