

MAESTRÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

Proyecto de Tesis:

MACHINE LEARNING EN LA DETECCIÓN DE FRAUDES DE COMERCIO ELECTRÓNICO APLICADO A LOS SERVICIOS BANCARIOS

Autor: Fredi William Álvarez

Año: 2020

ÍNDICE

ÍNDICE DE ILUSTRACIONES 5

CAPÍTULO UNO 7

ANTECEDENTES Y JUSTIFICACIÓN 7

1.1. TEMA: 7

1.2. DEFINICIÓN DEL PROBLEMA 7

1.3. OBJETIVOS 11

1.3.1 Objetivo General 11

1.3.2 Objetivos Específicos 11

1.4. JUSTIFICACIÓN 12

1.5. ALCANCES 13

CAPÍTULO DOS 14

FUNDAMENTOS TEÓRICOS 14

2.1. ECOMERCE EN AMERICA LATINA 15

2.2. FRAUDE 18

2.2.1. Tipos de Fraude 19

2.3. FRAUDE ELECTRÓNICO 22

2.3.2. Tipos de Fraudes Electrónicos 26

2.4. VISIÓN GENERAL DE LA SEGURIDAD BANCARIA 28

2.5. BIG DATA 31

2.5.1. Características 32

2.5.2. Tipos de Datos Big Data 33

2.5.3. Fases 34

2.5.4. Componentes 36

2.5.5. Aplicación de Big Data 37

2.6. MACHINE LEARNING 38

2.7. PRINCIPALES TÉCNICAS DE MACHINE LEARNING 40

2.7.1. Aprendizaje supervisado o predictivo 40

2.7.2. Aprendizaje no supervisado o descriptivo 42

2.7.3. Aprendizaje por Refuerzo 43

2.8. ALGORITMOS SUPERVISADOS CLÁSICOS DE MACHINE LEARNING 45

2.8.1. Random Forest 45

2.8.2. SVM (Máquinas de Soporte Vectorial) 46

2.8.3. AdaBoost 47

2.8.4. Regresión Lineal 48

2.8.5. Regresión Logística (RL) 48

CAPÍTULO TRES 50

DETECCIÓN DE FRAUDES 50

3.1. SISTEMAS DE DETECCIÓN DE FRAUDES (SDF) 51

3.1.1. Entrenamiento Supervisado 55

3.1.2. Actualización de los modelos 56

3.1.3. Inclusión de características adicionales a la data de la transacción 56

3.1.4. Como medir la precisión de un SDF 57

3.2. ALGORITMOS PARA DETECCIÓN DE FRAUDES 58

Métodos supervisados 59

Métodos no supervisados 62

CAPÍTULO 4 64

PLANTEAMIENTO DE LA SOLUCIÓN 64

4.1. Arquitectura de la propuesta 64

4.1.1. Apache Kafka 65

4.1.2. Apache Spark 66

4.1.3. Apache Cassandra 69

4.2. Proceso 70

4.2.1. Construcción del Modelo 70

4.2.2. Implementación del modelo 73

4.3. La estrategia de Aprendizaje 74

4.4. Monitoreo 77

4.5. Experimentación 81

4.5.1. Resultados 82

CAPÍTULO 5 88

CONCLUSIONES Y RECOMENDACIONES 88

5.1. Conclusiones y Recomendaciones 88

5.2. Limitaciones 90

5.3. Trabajos Futuros 91

BIBLIOGRAFÍA 92

# ÍNDICE DE ILUSTRACIONES

[Ilustración 1. Los 5 riesgos más probables en 2017 8](#_Toc35937141)

[Ilustración 2. Relación entre IA, ML y DL 13](#_Toc35937142)

[Ilustración 3. Crecimiento eCommerce en América Latina (US$ MM) 16](#_Toc35937143)

[Ilustración 4.- Ventas eCommerce por mercado 17](#_Toc35937144)

[Ilustración 5.- Resumen de métricas dentro del proceso de detección de fraudes en América Latina 18](#_Toc35937145)

[Ilustración 6.- Resumen de las áreas afectadas por fraude 20](#_Toc35937146)

[Ilustración 7. Conocimiento de Amenazas o Fraudes Electrónicos 25](#_Toc35937147)

[Ilustración 8. Responsables del Fraude Electrónico en Latinoamérica 30](#_Toc35937148)

[Ilustración 9. Valor promedio robado por incidente de Fraude Electrónico en Latinoamérica (en USD) 30](#_Toc35937149)

[Ilustración 10. Características de BIG DATA 33](#_Toc35937150)

[Ilustración 11. Cómo aprende la máquina (ML) 39](#_Toc35937151)

[Ilustración 12. Metodología de aplicación ML (Contreras, 2016) 40](#_Toc35937152)

[Ilustración 13. Principales algoritmos de aprendizaje supervisado 43](#_Toc35937153)

[Ilustración 14. Principales algoritmos de aprendizaje no supervisado 44](#_Toc35937154)

[Ilustración 15. Componentes de un sistema de aprendizaje por refuerzo 45](#_Toc35937155)

[Ilustración 16. Principales algoritmos de aprendizaje por refuerzo 46](#_Toc35937156)

[Ilustración 17. Elementos de un Sistema de Detección de Fraudes 55](#_Toc35937157)

[Ilustración 18. Componentes de Apache Spark 68](#_Toc35937158)

[Ilustración 19. Arquitectura de Cassandra. 70](#_Toc35937159)

[Ilustración 20. Flujo construcción de un modelo de aprendizaje automático 72](#_Toc35937160)

[Ilustración 21. Características consideradas para la construcción del modelo 74](#_Toc35937161)

[Ilustración 22. Arquitectura implementación del modelo 75](#_Toc35937162)

[Ilustración 23.- Ventana de tiempo de cada uno de los modelos (Feedback y Delayed) 77](#_Toc35937163)

[Ilustración 24. Captura de pantalla del listado de transacciones sospechosas en el monitor 79](#_Toc35937164)

[Ilustración 25. Captura de pantalla del detalle de una transacción sospechosa 80](#_Toc35937165)

[Ilustración 26. Captura de pantalla de los datos presentados para el contacto con el cliente 80](#_Toc35937166)

[Ilustración 27. Captura de pantalla del mensaje de confirmación de una transacción confirmada como legítima 81](#_Toc35937167)

[Ilustración 28. Distribución de la muestra agrupada por días 82](#_Toc35937168)

[Ilustración 29. Distribución de transacciones fraudulentas en el conjunto de datos de muestra. 83](#_Toc35937169)

[Ilustración 30. Distribución de transacciones fraudulentas detectadas como sospechosas 84](#_Toc35937170)

[Ilustración 31. Distribución de transacciones fraudulentas no detectadas como sospechosas. 84](#_Toc35937171)

[Ilustración 32. Resumen comparativo de fraudes detectados vs no detectados 85](#_Toc35937172)

[Ilustración 33. Matriz de confusión para el conjunto de datos de muestra 86](#_Toc35937173)

RESUMEN

Uno de los principales riesgos a los que están sometidas las entidades financieras son los ataques de fraudes electrónicos. Billones de dólares en pérdidas son absorbidas cada año por las entidades financieras debido a transacciones fraudulentas. Para combatir el fraude de manera efectiva es primordial comprender los mecanismos utilizados por los atacantes para cometer estos actos ilícitos.

El diseño de algoritmos de detección de fraude es particularmente difícil debido a la distribución no estacionaria de los datos, las distribuciones de clases altamente desequilibradas (fraude o legítimo) y la disponibilidad de pocas transacciones etiquetadas por los investigadores de fraude; al igual que la falta de disponibilidad de los datos de transacciones por razones de confidencialidad.

El presente trabajo plantea un modelo para detección de fraudes en transacciones financieras. Debido a la falta de información pública de transacciones financieras, el modelo fue planteado utilizando datos de transacciones de autorización de tarjetas de crédito, pero puede ser fácilmente adaptado para otro tipo de transacciones financieras.

El modelo planteado considera los principales retos en el diseño de un sistema de detección de fraudes: *a)* clases altamente desequilibradas, *b)* distribución de estacionaria de los datos y *c)* la incorporación en línea de la retroalimentación de los investigadores de fraude ante las transacciones etiquetadas como sospechosas.

La implementación del modelo planteado en el presente trabajo en un conjunto de datos de prueba permitió predecir exitosamente la mayoría de casos de transacciones fraudulentas con un mínimo porcentaje de falsos negativos.

# CAPÍTULO UNO

# ANTECEDENTES Y JUSTIFICACIÓN

## TEMA:

MACHINE LEARNING EN LA DETECCIÓN DE FRAUDES DE COMERCIO ELECTRÓNICO APLICADO A LOS SERVICIOS BANCARIOS

## DEFINICIÓN DEL PROBLEMA

Según una publicación realizada en el *World Economic Forum, Zurich* se indica dentro de los 5 riesgos más probables en 2017 a los Incidentes masivos de Fraude y robo de datos.



Ilustración 1. Los 5 riesgos más probables en 2017

La necesidad de las empresas y entidades bancarias que sus usuarios gestionen operaciones y transacciones por internet eleva el potencial riesgo que dichos procesos sean manipulados por hackers poniendo en peligro la integridad de los datos de los usuarios y la información reservada de las entidades bancarias (Morales, 2018).

El fraude contra el sistema financiero constituye un gran problema para los bancos ya que ocasionan pérdida económica, pérdida de imagen y desconfianza de los clientes.

Tras un historial de fraudes electrónicos presente en varias instituciones financieras ecuatorianas, la exigencia de tener un seguro fue impuesta por la Superintendencia de Bancos, el 17 de enero del 2012, en la resolución JB-2012-2090. No obstante, una de las principales razones para utilizar los servicios electrónicos de una institución financiera es la seguridad informática que ésta le brinda al usuario.

Cuando no se puede prevenir un fraude, es necesario poder detectarlo lo antes posible. El problema de la prevención y detección de un fraude, se magnifica por una serie de características y limitaciones. En primer lugar, se debe tener cuidado de no procesar demasiadas transacciones legítimas o bloquear incorrectamente tarjetas o cuentas genuinas. En segundo lugar, las instituciones financieras procesan un gran número de transacciones, de las cuales sólo un pequeño porcentaje es fraudulento (cerca del 0,1%) (Juszczak, Adams, Hand, Whitrow, & Weston, 2008), (Dal Pozzolo A. , 2015). En tercer lugar, sólo un número limitado de transacciones pueden ser revisados por investigadores de fraudes, por lo que las instituciones financieras necesitan automatizar su proceso de detección.

Los fraudes en tarjetas de crédito se pueden dar de diversas maneras (Anderson, 2007) y aunque no se puede dar una clasificación oficial debido a la evolución de las técnicas para fraude se puede mencionar como ejemplo:

La *tarjeta de crédito robada* es el tipo de fraude más común en el que el estafador suele intentar gastar tanto como sea y lo más rápido posible. La detección de un fraude de este tipo normalmente se basa en el descubrimiento de un patrón de uso inesperado de la tarjeta de crédito con respecto a la práctica común.

El fraude con *tarjeta no presente* se suele observar en el comercio electrónico. Aquí el estafador necesita la información sobre una tarjeta de crédito, pero no la tarjeta en sí. Este fraude exige una pronta detección ya que, a diferencia del caso anterior, el titular oficial de la tarjeta no es consciente de que sus propios datos han sido robados.

Otro tipo de fraude corresponde a la solicitud de una tarjeta de crédito con información personal falsa. Este tipo de fraude se produce más raramente ya que podría ser detectado durante la petición mediante el control de la información de la persona que solicita la tarjeta.

Cuando un método de fraude es detectado, los estafadores adaptan sus estrategias y/o generan unas nuevas. De igual manera todos los días se suman nuevos criminales participando en el juego y probando nuevas y viejas estrategias. Debido a esto es importante actualizar las herramientas de detección, pero también mantener activas las viejas herramientas de detección. El intercambio de ideas para actualizar o generar herramientas de detección es difícil ya que los estafadores podrían beneficiarse de ello probando sus estrategias. Por la misma razón, los conjuntos de datos sobre transacciones financieras no suelen estar disponibles públicamente para la comunidad investigadora ( Jha, Guillen, & Westland, 2012) (Zareapoor & Shamsolmoali, 2015).

Es difícil tener algunas cifras sobre el impacto del fraude, ya que las instituciones financieras no suelen revelar la cantidad de pérdidas debidas a fraudes. Otro problema en la estimación de la pérdida de fraude es que puede medir la pérdida de sólo aquellos fraudes que se han detectado, y no es posible evaluar el tamaño de los fraudes no reportados/no detectados. Otros fraudes se reportan mucho tiempo después de que el criminal ha completado el crimen

La Asociación para servicios de compensación de pagos (APACS) ha calculado que las pérdidas totales a través del fraude de tarjetas de crédito en el Reino Unido han crecido rápidamente de £122 millones en 1997 a £440,3 millones en 2010. Según el informe Nilson, las pérdidas globales de crédito, débito y fraude con tarjeta prepagada alcanzaron $11,27 mil millones en 2012 (un 14,6% más de 2011). Las pérdidas por fraude bruto representaron el 5,22% del volumen total, frente al 5,07% en 2011. En 2012, sólo en los Estados Unidos las pérdidas por fraude alcanzaron $5,33 mil millones. En 2014 las transacciones fraudulentas de tarjetas en todo el mundo han alcanzado alrededor de $11 mil millones al año, y Estados Unidos puede tener en cuenta aproximadamente la mitad de eso (Nexis, s.f.).

El Banco Central Europeo (BCE) informa que, en 2012, 1 euro por cada € 2.365 gastado en tarjetas de crédito y débito emitidas dentro de SEPA (La Unión Europea, Islandia, Liechtenstein, Mónaco, Noruega y Suiza) se perdió por fraude. Se estimó que el valor total del fraude alcanzó los 1,33 billones de euros en 2012, registrando un aumento del 14,8% en comparación con 2011. En particular, el 60% de estos fraudes provenían de pagos de tarjeta no presente (CNP) (es decir, pagos por correo postal, teléfono o Internet), 23% de terminales de punto de venta (POS) y 17% de cajeros automáticos. La introducción del estándar de seguridad EMV (tarjetas con chip) ha reducido la cuota de fraude (0,048% en 2008 y 0,038% en 2012) en el número total de transacciones. Sin embargo, de 2011 a 2012, los fraudes de CNP han aumentado en un 21%, tras el aumento de los pagos del CNP, que aumentó alrededor del 15% al 20% al año entre 2008 y 2012, mientras que las otras transacciones aumentaron cerca del 4%.

El BCE también informó que las tarjetas de crédito son más afectadas que la tarjeta de débito, calculando que por cada 1000 euros se perdió 1 euro por fraude en tarjetas de crédito, mientras que la pérdida en tarjeta de débito fue de 1 euro por cada 5400 euros.

Otro dato interesante es que el fraude CNP suele ser más frecuente en los mercados de tarjetas maduras, mientras que el fraude POS es más común en los mercados menos desarrollados.

Una vez identificados los riesgos de fraude a los que se enfrentan las entidades financieras, es necesario contar con herramientas informáticas, que permitan identificar dentro del gran número de registros de transacciones, patrones de comportamiento que no son usuales y/o que corresponden a actividades potencialmente fraudulentas. La detección a tiempo de un fraude permite evitar daños, proteger la reputación, los activos corporativos e incrementar la confianza por parte de los clientes.

El presente proyecto propone que, dadas las características de una transacción y el historial de transacciones fraudulentas obtenidas en el pasado, permita obtener un modelo de aprendizaje que permita clasificar la transacción como fraudulenta o íntegra.

## OBJETIVOS

### Objetivo General

Proponer un sistema de detección de fraudes en el comercio electrónico online usando Machine Learning aplicado a servicios bancarios.

### Objetivos Específicos

* Determinar los desafíos actuales para la detección de fraudes electrónicos en los servicios bancarios en línea.
* Analizar las diferentes técnicas de Machine Learning que podrían ser utilizadas para minimizar el riesgo de fraudes electrónicos a nivel de servicios bancarios.
* Implementar un prototipo que permita validar la solución propuesta al detectar las características de transacciones fraudulentas y evitar que las operaciones mal intencionadas lleguen a concretarse.
* Establecer los aportes que brinda la inteligencia artificial en cuanto a las herramientas que se pueden emplear en casos de detección de fraudes en los servicios de la banca electrónica.

## JUSTIFICACIÓN

El aprendizaje automático o Machine Learning identifica patrones entre millones de datos de diferente naturaleza (Big Data) y predice comportamientos a través de algoritmos con capacidad de aprender y evolucionar basándose en su propia experiencia (Bisong, 2019).

Ahora bien, Machine Learning y Deep Learning son términos asociados a la Inteligencia Artificial que en ocasiones se dificulta su distinción, sin embargo, su diferencia radica en la sofisticación que posee el uno del otro. Machine Learning es una tecnología de aprendizaje automático mediante la organización de datos y el reconocimiento de patrones (Bejerano, 2017), mientras que Deep Learning se basa en el aprendizaje profundo más avanzado construido a partir del principio de redes neuronales imitando al cerebro humano a través de la superposición de capas no lineales de procesamiento de datos (Bisong, 2019).



Ilustración 2. Relación entre IA, ML y DL

En este caso se utiliza Machine Learning, debido a que estos sistemas pueden examinar ciertos patrones basado en una serie de requisitos y reglas, esto logra que al momento de aplicarlos el software este en la capacidad de adaptar las premisas y crear otras nuevas a fin de optimizar la tasa de aciertos. Es importante acotar que el Deep Learning, practica un análisis por capas o unidades neuronales arrojando resultados que se revelan en forma cuantitativa, es decir, expresa ponderación de acierto y desacierto.

Según datos del Instituto Nacional de Estadísticas y Censos (INEC) de una encuesta aplicada a 3777 empresas en el año 2015, el 8,4% vendieron sus productos a través de la red, mientras el 14,6% hicieron alguna compra; frente al 8,7% y 14% del 2013 respectivamente. Las entidades bancarias son un recurso para facilitar este tipo de intercambio, sin embargo, se está convirtiendo en un riesgo que puede conducir a pérdidas masivas, debido a la aparición y evolución de estafas de phishing, infección de malware y sitios web de fantasmas.

Para la banca, la idea es definir servicios que se ajusten a las necesidades e intereses de los clientes, tanto en el estudio de los hábitos de consumo como de los canales donde los usuarios de banca muestran un mayor compromiso; por esta razón es de vital importancia que las instituciones financieras cuenten con un protocolo de seguridad ante fraudes en servicios electrónicos.

## ALCANCES

* Se explorarán técnicas de Machine Learning para determinar la que mejor se adapte a nuestro modelo, minimizando así el riesgo de fraude electrónico en el servicio bancario.
* Se desarrollará un prototipo que detectará las características de transacciones fraudulentas evitando así que estas operaciones se concreten.
* Se investigarán los aportes que brinda la inteligencia artificial para detección de fraudes en los servicios de la banca electrónica.

# CAPÍTULO DOS

# FUNDAMENTOS TEÓRICOS

En la actualidad se usa con frecuencia operaciones de comercio electrónico y estas a su vez van ligadas a los servicios bancarios, realizando de esta manera, un sin número de operaciones bancarias entre las más comunes: el uso de tarjetas de crédito para realizar compras desde cualquier lugar del mundo, pago de servicios básicos, recargas de saldo a celulares, transferencias bancarias, etc. Sin embargo, este tipo de facilidades requieren de sistemas de seguridad robustos para evitar la sustracción de información delicada y personal de cada usuario, las entidades bancarias están en la obligación de brindar toda la seguridad que las operaciones electrónicas realizadas por sus clientes requieren para que la operación se desarrolle de manera correcta (Security Standards Counsil, 2018).

Un sistema de seguridad para este tipo de operaciones puede basarse en algoritmos de detección mediante técnicas de Machine Learning (ML), que se encargan de detectar si un tipo de movimiento realizado por cierto usuario es habitual o no, en cuyo caso se deben tomar las medidas pertinentes. Una de las grandes ventajas de usar un algoritmo de ML es que cuantos más movimientos realice un usuario, más precisión tendrá el algoritmo a la hora de determinar la veracidad del movimiento; determinando así si es una operación lícita o un fraude (Dal Pozzolo A. , 2015).

Tomando en cuenta que uno de los activos más valioso es la información, al usar técnicas de Machine Learning se busca identificar las posibles operaciones o transacciones que no son realizados por los usuarios idóneos para desarrollarlas



## ECOMERCE EN AMERICA LATINA

La estructura cultural de los países latinoamericanos, el modelo de negocios digitales o eCommerce, ha ido, poco a poco y exitosamente expandiéndose en las naciones de América Latina. Actualmente, en la región el 60% de la población tiene algún tipo de acceso a internet y aproximadamente el 42% de los hogares tienen una computadora de los cuales el 10% tiene internet de banda ancha fija. América Latina presenta una situación de conectividad tal que es posible acceder a un teléfono celular no tan costoso y accesible a la mayoría.

A continuación, se presenta un gráfico que ilustra el crecimiento del comercio electrónico en América Latina (Cibersource, 2017).

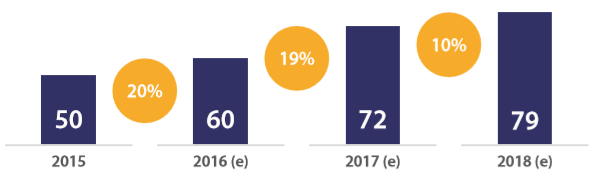


Ilustración 3. Crecimiento eCommerce en América Latina (US$ MM)

En la Ilustración 4 se muestra un gráfico de la distribución de las transacciones de eCommerce en la región de América Latina donde se destaca que el comercio digital está concentrado principalmente en 3 países: Brasil, México y Argentina (Cibersource, 2017).

Ilustración 4.- Ventas eCommerce por mercado

El rubro que más se ha beneficiado con la incorporación y crecimientos del comercio electrónico es el turístico ya que el 70% de sus transacciones provienen de algún canal online (web, app).

El comercio electrónico móvil (mCommerce), representa el 48% de todo el comercio electrónico en América Latina: el uso de app móviles para comercio electrónico alcanzó un 19% del total de las transacciones, mientras que un 29% de las operaciones fueron realizadas a través de la web móvil. La categoría de mCommerce es el rubro que presentará el mayor crecimiento en los próximos años (Estudios: Latinoamérica, la región de más rápido crecimiento en M-Commerce, 2018).

**Estructura del proceso Risk Manager Pipe-Line para Fraude en América Latina**

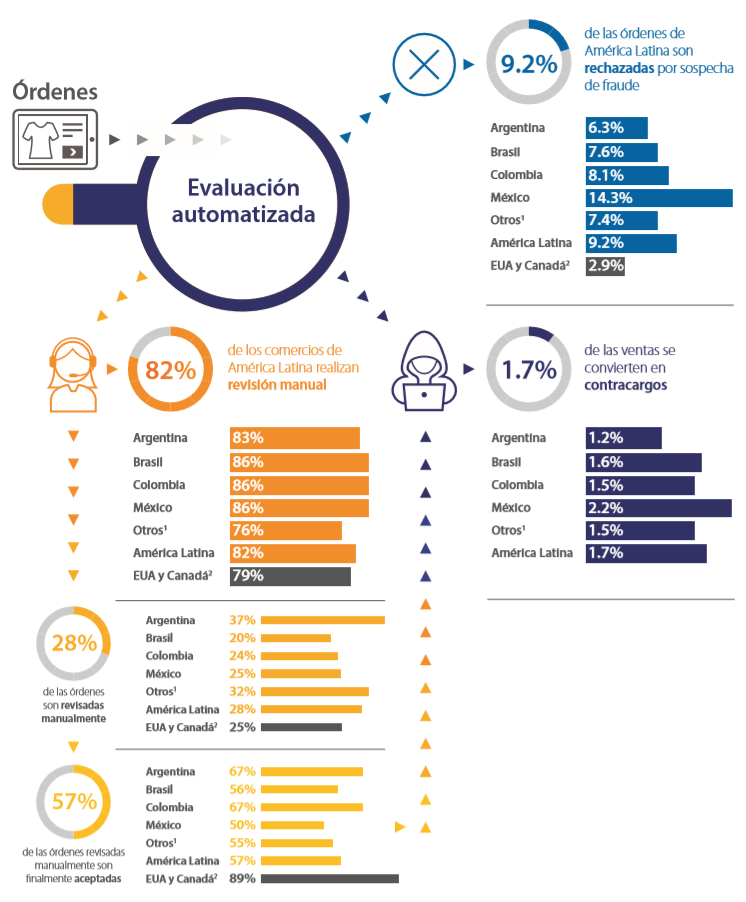


Ilustración 5.- Resumen de métricas dentro del proceso de detección de fraudes en América Latina

## FRAUDE

Según la Real Academia de la Lengua Española (RAE) el fraude se define como:

*“Acción contraria a la verdad y a la rectitud, que perjudica a la persona contra quien se comete”*

Ésta definición no abarca todas las características que comúnmente están presentes en un acto fraudulento.

Una de las mejores definiciones de fraude fue propuesta por Véronique Van Vlasseler (Social Network Analysis for Fraud Detection, 2013) que define al fraude como:

*“Fraud is a uncommon, well-considered, imperceptibly concealed, time-evolving and often carefully organized crime witch appears in many types and forms”*

*“El fraude es un crimen poco común, bien considerado, imperceptiblemente oculto, que evoluciona en el tiempo y, a menudo, cuidadosamente organizado, que aparece en muchos tipos y formas”*

De la definición anterior se pueden rescatar 5 características principales:

**Poco** **común.**  La mayoría de los fraudes detectados son considerados eventos raros dentro de su universo de eventos.

**Imperceptible oculto**. Los eventos fraudulentos generalmente no son detectados en el momento que ocurren y en algunos casos nunca lo son.

**Evoluciona en el tiempo**. Los defraudadores acostumbran a refinar sus métodos de ataque o engaño de manera que les permita seguir con sus actividades sin ser detectados.

**Cuidadosamente organizado**. Generalmente un fraude no es realizado por una sola persona, es realizado por un conjunto de personas o una organización donde cada miembro interpreta un papel o rol dentro de la acción de fraude.

**Diferentes tipos o formas**. Los fraudes pueden darse en diferentes tipos y formas; en la siguiente ilustración se presenta un resumen de las diferentes áreas sobre las que afecta el fraude (Carreño Lopez, 2017).

Ilustración 6.- Resumen de las áreas afectadas por fraude



### Tipos de Fraude

**Robo de Identidad**

Los tipos más comunes de fraudes que causan preocupación son el robo de identidad (71%), el phishing (66%) y el robo de cuentas (63%). Aquí, las tarjetas de crédito son el objetivo más popular, ya que un estafador no necesita mucho para realizar una transacción de "tarjeta no presente".

En el robo de identidad tradicional, el objetivo de los delincuentes es realizar transacciones utilizando una identidad diferente. En lugar de tener que idear una identidad completamente nueva para hacer esto, simplemente se apoderan de una existente. Esto es más fácil de hacer, y generalmente mucho más rápido.

Para cometer el robo de identidad o apropiarse de la identidad de alguien, los estafadores se enfocan en información personal, como nombres, direcciones y direcciones de correo electrónico, así como información de cuentas o tarjetas de crédito. Esto les permite, por ejemplo, ordenar artículos en línea con un nombre falso y pagar utilizando la información de la tarjeta de crédito de otra persona o debitando la cuenta de otra persona.

**Fraude Amistoso**

Mediante este método, los clientes piden bienes o servicios y pagan por ellos, preferiblemente utilizando un método de pago "pull" como una tarjeta de crédito o débito directo. Entonces solicitan deliberadamente una devolución de cargo, alegando que su tarjeta de crédito o los detalles de su cuenta fueron robados. Son reembolsados, pero se quedan con los bienes o servicios. Este método de fraude es particularmente frecuente con los servicios, como los de los juegos de azar o los entornos para adultos.

El fraude amistoso también tiende a combinarse con el reenvío. Aquí es donde los delincuentes que usan datos robados para pagar sus compras no quieren que se los envíen a sus domicilios. En su lugar, utilizan servicios o personas intermediarias cuyas direcciones se utilizan para realizar las compras.

**Fraude limpio**

El nombre de fraude limpio es engañoso, porque no hay nada claro al respecto. El principio básico de fraude limpio es que una tarjeta de crédito robada se utiliza para realizar una compra, pero la transacción se manipula de tal manera que se evitan las funciones de detección de fraude. Aquí se requiere mucho más conocimiento que con el fraude amistoso, donde el único objetivo es cancelar el pago una vez que se haya realizado la compra.

En el fraude limpio, los delincuentes utilizan análisis sólidos de los sistemas de detección de fraude desplegados, además de un gran conocimiento sobre los legítimos propietarios de sus tarjetas de crédito robadas. Luego se ingresa una gran cantidad de información correcta durante el proceso de pago para que la solución de detección de fraude sea engañada.

Antes de cometer fraude limpio, a menudo se realizan pruebas de tarjeta. Esto implica realizar compras baratas de prueba en línea para verificar que los datos robados de la tarjeta de crédito funcionan.

**Fraude por triangulación**

Durante el fraude de triangulación, el fraude se realiza a través de tres puntos. El primero es un escaparate en línea falso, que ofrece productos de alta demanda a precios extremadamente bajos. En la mayoría de los casos, se agrega un cebo adicional, como la información de que las mercancías solo se enviarán de inmediato si se pagan con una tarjeta de crédito. La tienda falsificada recopila datos de direcciones y tarjetas de crédito, este es su único propósito. La segunda esquina del triángulo de fraude implica el uso de otros datos de tarjetas de crédito robadas y el nombre recopilado para ordenar productos en una tienda real y enviarlos al cliente original. El tercer punto en el triángulo de fraude consiste en utilizar los datos de tarjetas de crédito robadas para realizar compras adicionales.

Los datos de los pedidos y los números de las tarjetas de crédito son ahora casi imposibles de conectar, por lo que el fraude generalmente permanece sin ser descubierto durante un período de tiempo más prolongado, lo que genera mayores daños.

**Fraude mercantil**

El fraude mercantil es otro método que debe mencionarse. Es muy simple: los productos se ofrecen a precios baratos, pero nunca se envían. Los pagos, por supuesto, se mantienen. Este método de fraude también existe al por mayor. No es específico de ningún método de pago en particular, pero esto es, por supuesto, donde los métodos de pago sin devolución de cargo donde los métodos de pago sin devolución de cargo entran en vigencia.

**Pago por anticipado**

En otras ocasiones los delincuentes envían correos electrónicos fraudulentos en los que piden ayuda a los destinatarios para transferir grandes cantidades de dinero a terceros y les prometen un porcentaje si aceptan procesar la transferencia utilizando sus cuentas personales. La dinámica del fraude por pago anticipado consiste en engañar a las posibles víctimas persuadiéndolos de que recibirán un beneficio sustancial, a cambio de proporcionar un pago modesto por adelantado. En esencia, este tipo de fraude abarca fraudes de mercadotecnia masiva, fraudes de herencia, fraudes de beneficencia o desastre falso, loterías falsas y esquemas piramidales.

## FRAUDE ELECTRÓNICO

El fraude electrónico o delito informático es una actividad indebida basada en la manipulación fraudulenta de elementos informáticos y sistemas de comunicación, para obtener un beneficio no autorizado (Téllez, 2004).

Las cifras publicadas por el APWG **(Anti-Phishing Working Group**) indican que esta actividad ha aumentado un 5,7% en los últimos 12 años, siendo los países de Latinoamérica los más afectados (Group, 2019). Pese a la recesión económica que varios países han sufrido en los últimos años, los mercados de seguridad de TI han crecido considerablemente, empresas en Latinoamérica han aumentado su concientización sobre las amenazas virtuales y consecuentemente han realizado mayores inversiones en Seguridad de TI en los últimos años. Una de las exigencias de regulaciones internacionales a partir del 2010 es el estándar de seguridad de la información para la industria de tarjetas de pago o PCI DSS (Security Standards Counsil, 2018).

Los principales obstáculos en el crecimiento de los mercados de Seguridad de TI son (Frost & Sullivan, 2010):

* Falta de una medición cuantificable del Retorno de la Inversión (ROI) para soluciones de seguridad de TI.
* Falta de presupuesto para seguridad de TI.
* Escasa importancia de las amenazas existentes.
* Inestabilidad política y económica.

Los servicios bancarios a través de Internet y el aumento de los trabajadores remotos, son dos fuertes impulsores globales de fraude virtual. Los niveles de conocimiento de amenazas en toda la región Latinoamericana son relativamente bajos, particularmente cuando se comparan con los de países más desarrollados. Este bajo conocimiento tiene como consecuencia una pobre adopción de soluciones de prevención, aumentando a la vez el riesgo de fraude en Internet (Frost & Sullivan, 2010).

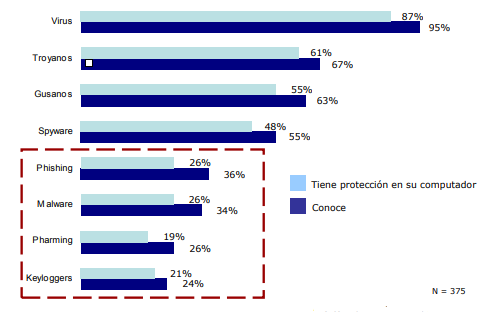


Ilustración 7. Conocimiento de Amenazas o Fraudes Electrónicos

Las amenazas con un nivel de conocimiento relativamente bajo y consecuentemente un bajo nivel de protección, incluyen ataques de Phishing, Malware, Pharming y Keyloggers. Todas estas amenazas tienen niveles de conocimiento inferiores al 40% y niveles de protección por debajo del 30% (Ilustración 7Ilustración 7*)*, aumentando así, las vulnerabilidades del usuario por ataques de fraude electrónico (Frost & Sullivan, 2010).

Cuantos más comerciantes en línea y clientes de comercio electrónico hay, más víctimas potenciales hay para los estafadores. La internacionalización del comercio electrónico está permitiendo que los delincuentes en línea altamente especializados se vuelvan activos internacionalmente. Mientras que los ataques de fraude en línea se utilizaban para dirigirse principalmente a bancos y proveedores de pagos, ahora están muy bien equipados para hacer frente a tales amenazas, aprovechando las medidas de protección técnica y los servicios avanzados de detección de fraude, así como las normas reglamentarias para la industria financiera (Security Standards Counsil, 2018). Por lo tanto, a los atacantes se les hace más difícil superar los principales obstáculos para saquear las instituciones financieras.

Aunque las tiendas en línea tienden a estar menos protegidas, también procesan los datos de los clientes y reciben información financiera confidencial, los mecanismos de protección utilizados por muchos comerciantes aún no son muy modernos: no tienden a realizar verificaciones en vivo de la información ingresada por el cliente ni a implementar sistemas sofisticados de administración de riesgos (Masoud, 2013).

El móvil es el nuevo escritorio, los usuarios se están alejando cada vez más de las PC y computadoras portátiles tradicionales hacia teléfonos inteligentes y tabletas (Fundación Telefónica, 2016). El problema aquí es que los mecanismos de protección para dispositivos móviles no son tan completos como los diseñados para computadoras tradicionales. Otro factor que a menudo se pasa por alto es que las pequeñas pantallas de los teléfonos móviles hacen que sea mucho más fácil encontrar sitios web fraudulentos: los usuarios tampoco pueden ver los detalles (Porter Felt & Wagner). Los patrones de comportamiento en dispositivos móviles también son diferentes: los usuarios de teléfonos inteligentes están acostumbrados a controlar todo a través de unos pocos toques, por lo que las funciones de seguridad complejas no son prácticas. El pago se realiza generalmente mediante métodos de un solo clic.

La gestión de riesgos para los clientes móviles también tiende a ser problemática, ya que ya no es posible evaluar simplemente su ubicación; después de todo, el objetivo de los dispositivos móviles es brindar a los usuarios libertad de movimiento. Las amenazas de malware en teléfonos inteligentes y tabletas también siguen siendo un tema interesante. Aunque se ha hablado durante años, los expertos creen que la gran plaga de los virus de teléfonos inteligentes aún está por venir. En 2014, hubo alrededor de 400,000 nuevos virus para dispositivos móviles (Cordero Perez, 2014).

Las estrategias de KYC (know your client) son importantes (James, 2019), pero no son suficientes por sí mismas. La clasificación de los clientes es algo positivo. ¿Pero qué pasa si una cuenta de cliente es hackeada? En tales casos, no es el cliente de confianza que realiza las transacciones, sino el estafador, que utiliza nombre del cliente. Por lo tanto, además de las conocidas funciones de KYC, las instituciones deben utilizar soluciones de detección de fraudes para reconocer cuándo un cliente realiza compras o transacciones inusualmente frecuentes con montos inusualmente altos o pequeños.



### Tipos de Fraudes Electrónicos

**Phishing**

Es sinónimo de suplantación de identidad, es la obtención fraudulenta de información crítica de los clientes por medio de una página web similar a la auténtica, los principales datos críticos a ser obtenidos pueden ser nombres de usuario, contraseñas, números de tarjetas de crédito.

Generalmente por medio de un correo electrónico, SMS o redes sociales se simula una comunicación oficial de la empresa, direccionando a una página web similar a la de la empresa real para que el usuario ingrese los datos que pueden ser extraídos de manera fraudulenta (Aite, 2014).

Smishing y vishing son formas de phishing más sofisticadas y utilizan mensajes de texto y llamadas telefónicas para atraer a las víctimas. Este tipo de fraude también se puede utilizar para dirigirse a corporaciones y otros comerciantes. Los sitios de comerciantes de eCommerce han sido un objetivo, ya que normalmente contienen datos personales valiosos o información de tarjetas de pago almacenada que se puede usar para compras fraudulentas (Koshrow-Pour).

Pharming, en el cual los navegadores manipulados dirigen a los clientes confiados a sitios web fraudulentos. Se realiza la explotación de una vulnerabilidad en el software de los servidores o en el equipo de los propios usuarios que permite al atacante redirigir un nombre de dominio a sitios web no deseados.

**Malware**

Es la abreviatura de “*Malicious Software*”; este término abarca a todo programa o código malicioso cuyo objetivo es dañar el sistema operativo o causar un mal funcionamiento (Koshrow-Pour).

El malware se está volviendo más inteligente. No importa qué tan amplia sea la protección técnica, los estafadores usan malware inteligente para mantenerse al día, esto significa que las amenazas seguirán aumentando, y no solo en los dispositivos móviles. Todos los canales de comercio electrónico, ya sean ventas por teléfono o ventas a través de plataformas de socios, están constantemente bajo fuego. La razón de esto es que, en los últimos años, la escena del malware se ha vuelto extremadamente profesional (Aite, 2014). Sin embargo, con los virus, no es el número lo que es aterrador: es el hecho de que ahora es posible no solo clonar virus, sino modificarlos de tal manera que formen entidades completamente nuevas: las que no se pueden detectar por los mecanismos de seguridad existentes. Considerando que, en 2014, hubo alrededor de 82 millones de nuevos virus al año, se estima que ya existen más de 166 millones[[1]](#footnote-1).

**Keyloggers**

Son pequeños programas maliciosos utilizados para capturar cualquier tipo de actividad realizada en el computador, de esta manera se podría conocer todo lo digitado por la víctima, esta actividad se puede grabar en un registro en el mismo ordenador o a su vez enviarlo a un equipo remoto que fue configurado por el atacante (Pathak, Pawar, & Patil, 2015).

**Spyware**

Comúnmente conocido como software espía, para este ataque se utiliza un software que es instalado sin autorización en la máquina de la víctima para monitorear las actividades que esta realiza desde un acceso remoto (Koshrow-Pour).

**Spam**

Otro tipo de fraude involucra el envío de correo no deseado donde los correos electrónicos no solicitados o las publicaciones de los grupos de noticias no deseados se envían sin el consentimiento del receptor, con frecuencia son maliciosos y en ocasiones los delincuentes pretenden ser instituciones o compañías financieras solicitando información personal o credenciales de acceso a sus cuentas (Mendez, Fdez-Riverola, Díaz, & Corchado, 2007). A la luz de eso, algunos expertos sugieren que la única solución real en la lucha contra el spam es aumentar los costos de transmisión para los remitentes.

**Hacking**

El hacking malicioso o más conocido como cracking es uno de los delitos informáticos más antiguos relacionados con el acceso ilegal a un sistema informático (Seguridad en Sistemas y Técnicas de Hacking, 2011). Estos tipos de ataques suelen ser sofisticados y utilizan técnicas de cobertura de rastro, como las computadoras de retransmisión, para que parezca que el ataque se origina localmente o en otra localización y hace que sea más difícil rastrearlas. Los piratas informáticos obtienen acceso no autorizado a grandes cantidades de datos confidenciales con el objetivo de robar información, causar daños monetarios y de reputación a la entidad objetivo (Donaldson, Williams, & Siegel, 2019).

## VISIÓN GENERAL DE LA SEGURIDAD BANCARIA

La seguridad bancaria y por ende de las transacciones electrónicas es una responsabilidad conjunta entre el usuario y la entidad financiera, no obstante el 43% de los usuarios en Latinoamérica consideran a los bancos como la entidad responsable de la seguridad de las transacciones electrónicas (Ilustración 8).

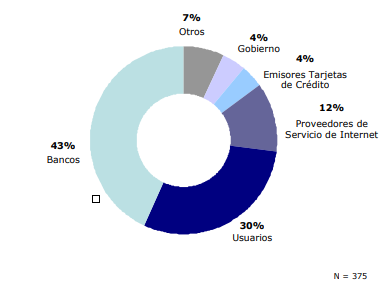


Ilustración 8. Responsables del Fraude Electrónico en Latinoamérica

La clonación de tarjetas crédito y débito, fue el fraude más común en Latinoamérica en 2010, seguido por fraudes virtuales incluyendo compras en línea. Las compras no autorizadas por Internet se están convirtiendo en el ataque más común entre los usuarios de transacciones electrónicas. Adicionalmente, hay un alto número de transacciones fraudulentas sobre cajeros automáticos (Frost & Sullivan, 2010).

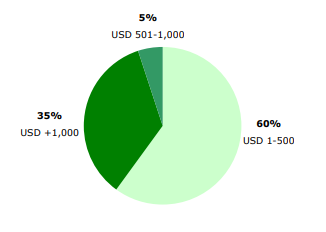


Ilustración 9. Valor promedio robado por incidente de Fraude Electrónico en Latinoamérica (en USD)

Es necesario que las entidades financieras puedan inculcar una cultura de seguridad informática para minimizar los riesgos de fraudes electrónicos, para esto tanto los bancos como los proveedores de servicios transaccionales invertirán de manera constante en soluciones de seguridad para la red y en servicios gestionados de seguridad especializados en prevención y detección de fraudes electrónicos (Bhatla, Prabhu, & Amit, 2003).

Entre las principales recomendaciones para minimizar actividades de fraude electrónico publicadas por la APWG están:

* Análisis en tiempo real de riesgo individual de transacciones.
* Soluciones de autenticación multi-factor y servicios profesionales.
* Soluciones constantes anti-phising y anti-pharming.
* Monitoreo de malware en cajeros automáticos.

El continuo avance de la tecnología ha beneficiado en gran medida a las entidades financieras donde la seguridad y control de riesgos es uno de sus principales ejes del negocio, con la finalidad de disminuir los ataques y fraudes ML ayuda a identificar posibles amenazas para mitigar de manera eficaz los riesgos de las mismas.

Usando técnicas de ML se pueden crear diferentes modelos de detección de fraudes electrónicos ya que se cuenta con mucha más información, de esta manera lo que en décadas anteriores tomaba varias horas o incluso días ahora toma segundos, siendo el sector financiero uno de los más beneficiados (Dal Pozzolo A. , 2015).

Al implementar algoritmos de ML, los costos para las empresas pueden reducir de manera considerable, mejorando las funcionalidades y las ofertas. Los algoritmos de ML aplicado a servicios bancarios no sólo ayudan en la prevención de robo de dinero sino también de información confidencial del cliente; así como recopilar y analizar gran cantidad de datos en menor tiempo (Calle, 2018).

Los sistemas anteriores para detectar de fraudes electrónicos basaban su funcionamiento en un conjunto de reglas y estándares partiendo de la premisa que toda actividad fraudulenta tenía un comportamiento similar, por ejemplo, provenían de ciertos grupos de direcciones IP sospechosas, inicios de sesión inusuales, actividades electrónicas más frecuentes, etc (Dal Pozzolo A. , 2015). Con el desarrollo de la tecnología es importante no solamente enfocarse en el listado de reglas sino brindar a los sistemas de seguridad una retroalimentación mediante el uso de ML o aprendizaje automático permitiendo detectar actividades poco usuales de fraude de baja incidencia y no detectables por las técnicas anteriores.

Las técnicas de ML ayudan a distinguir actividades fraudulentas internas y externas en la entidad financiera, si bien el sistema bancario maneja grandes cantidades de dinero también gestiona grandes volúmenes de datos, considerando a la información como un activo importante para el desarrollo de la actividad financiera, cabe recalcar que una actividad de fraude electrónico no sólo es para extraer dinero sin autorización sino también información crítica y protegida.

## BIG DATA

En la actualidad todas las áreas de negocio están atravesando una transformación tecnológica e ideológica en función de la gran cantidad de datos generados y la evolución de la tecnología, día a día surgen nuevos tipos de datos y necesidades que los sistemas y servicios actuales no pueden procesar de manera idónea, generando de esta manera un tiempo de respuesta más grande con inversión más elevada.

Con el avance de la tecnología las técnicas utilizadas para interpretación de grandes volúmenes de datos van encaminadas a Técnicas de Big Data, ya que el tiempo de procesamiento disminuye notablemente al igual que su inversión. Big Data es una técnica utilizada para el análisis de información que no puede ser procesada por métodos y herramientas tradicionales; simplificando de esta manera el tiempo de respuesta (Hernández, 2015). Esta técnica es aplicada en diversas áreas industriales como, por ejemplo: hidrografía, meteorología, estadísticas y censos, análisis de redes sociales y obviamente temas críticos como transacciones electrónicas y bancarias.



### Características

Uno de los cambios más significativos en las técnicas de Big Data es la captura de datos, entre las características más importantes se muestran en la Ilustración 10 (Riahi & Riahi, 2018):



Ilustración 10. Características de BIG DATA

**Volumen**

Se maneja un gran volumen de datos, se espera que para el año 2020 se tenga 35 zetabytes de datos guardados, considerando que Twitter genera 7 terabytes; por tal razón es importante el estudio y análisis de las técnicas de Big Data para ayudar a las diferentes organizaciones a procesar dicha información (Grable & Lyons, 2018).

**Velocidad**

El volumen de datos capturados puede ser procesado y analizado en tiempo real generando así que el tiempo de respuesta sea demasiado rápido como para lograr obtener la información correcta en el momento preciso (Grable & Lyons, 2018).

**Variedad**

Las técnicas de Big Data no se limitan sólo a datos estructurados, sino que incluyen datos del tipo no estructurado como archivos de audio, logs, etc (Grable & Lyons, 2018).

**Veracidad**

Es la confianza o fiabilidad que los datos generados por Big Data suponen en la toma de decisiones de la empresa (Riahi & Riahi, 2018).

**Valor**

El valor de la información escondida dentro de los datos tradicionales y no tradicionales varía significativamente, lo fundamental es identificar que data es valiosa y empezar con el análisis y procesamiento de la misma de manera rentable y eficiente (Riahi & Riahi, 2018).



### Tipos de Datos Big Data

Considerando la clasificación de los diversos tipos de datos que realiza la consultora IBM se tienen 5 grandes grupos (Mysore, Khupat, & Jain, 2013):

**Web and Social Media**: Incluye contenido web e información que es obtenida de las redes sociales como:

* Facebook,
* Twitter,
* Blogs
* LinkedIn, etc.

**Machine-to-Machine (M2M):** M2M se refiere a las tecnologías que permiten conectarse a otros dispositivos. M2M utiliza dispositivos como sensores o medidores que capturan algún evento en particular (velocidad, temperatura, presión, variables meteorológicas, variables químicas como la salinidad, etc.) los cuales transmiten a través de redes alámbricas, inalámbricas o híbridas a otras aplicaciones que traducen estos eventos en información significativa.

**Big Transaction Data**: Incluye registros de facturación en telecomunicaciones registros detallados de las llamadas (CDR), etc. Estos datos transaccionales están disponibles en formatos tanto semiestructurados como no estructurados.

**Biometrics:** Información biométrica en la que se incluye huellas digitales, escaneo de la retina, reconocimiento facial, genética, etc. En el área de seguridad e inteligencia, los datos biométricos han sido información importante para las agencias de investigación.

**Human Generated**: Las personas generamos diversas cantidades de datos como la información que guarda un call center al establecer una llamada telefónica, notas de voz, correos electrónicos, documentos electrónicos, estudios médicos, etc.



### Fases

La implementación de una solución Big Data comprende las siguientes fases (Galeano Cruz & Domínguez Rivera, 2017):

**Fase Inicial**

En esta fase se determina si la implementación de la solución es real y factible, es necesario planificar la implementación de la solución de Big Data analizando los siguientes aspectos:

*Detectar las necesidades*. - Es válida la comparación de la tecnología nueva y la existente, tomando en cuenta la variedad de los datos, la velocidad, el almacenamiento, el volumen de datos.

*Justificar la inversión. -* Una de las razones más válidas para justificar la inversión es la parte tecnológica, ya que procesar y almacenar una mayor cantidad de información en menor tiempo representa a la empresa mayor rendimiento y ahorro en costo.

*Evaluar las limitaciones. -* Además de tomar en consideración la infraestructura de la empresa, su madurez tecnológica y sus posibilidades (presupuesto y recursos, entre otros); también hay que valorar las cuestiones legales, en especial en lo concerniente a las exigencias en materia de privacidad de datos.

**Fase de planificación**

En esta fase se determinan los recursos asociados y necesarios para la ejecución del proyecto, así como el presupuesto destinado al proyecto que se establece en función de la capacidad económica y necesidades de la empresa, entre los recursos más relevantes se tiene:

* Gestores
* Diseñadores y arquitectos de datos
* Implementadores
* Operadores de datos

**Fase de diseño de infraestructura de la solución**

Es la arquitectura propia del sistema a implementar, es decir el diseño más óptimo para atacar las necesidades de la empresa tomando en cuenta la escalabilidad, costo, calidad y migración de datos. El diseño de la solución consta de dos secciones:

* Infraestructura
* Arquitectura

**Fase de implementación**

Conocida como fase productiva ya que en esta fase se realiza el despliegue de la solución tanto a nivel de arquitectura como de infraestructura tomando en cuenta la respectiva seguridad del sistema (accesos, niveles de privacidad y aplicación de la normativa vigente); se considerará:

* Instalación de la infraestructura.
* Implementación de la arquitectura.
* Puesta en marcha operativa.

### Componentes

Para que la solución Big Data implementada trabaje de manera óptima se sugiere que la solución cuente con los siguientes componentes (Mora, 2016):

**Fuentes:** las más habituales son los registros históricos de la compañía, los almacenes de datos, los dispositivos inteligentes, los sistemas de gestión de datos, Internet y el Internet de las Cosas (IoT).

**Capa de almacenamiento**: su función es la de recoger y transformar los datos sin perder de vista la normativa legal. Además, tiene que dar acceso a los datos independientemente de su formato, volumen, frecuencia u origen.

**Capa de análisis**: se encarga de leer los datos almacenados. Mediante la utilización de los modelos, los algoritmos y las herramientas adecuadas, proporciona visibilidad sobre los datos para que puedan ser consultados en la capa de consumo.

**Capa de consumo**: son muchos los proyectos y usuarios que se benefician del conocimiento extraído en todo este proceso. La forma de consumir los datos dependerá del destinatario, pero será habitual verlos en forma de *reporting* o visualización en tiempo real.

### Aplicación de Big Data

El campo de aplicación de proyectos basados en técnicas de Big Data es numeroso, entre las principales aplicaciones se destacan (Ladrero, 2017):

* Segmentación de clientes.
* Optimización de procesos de negocio.
* Cuantificación y optimización del rendimiento personal y deportivo.
* Mejoras en la Salud Pública.
* Investigación.
* Rendimiento de máquinas y dispositivos.
* Optimización de Seguridad Cibernética.
* Mejorar la experiencia del usuario.
* Análisis de operaciones en tiempo real.
* 360° Visión del Cliente.

## MACHINE LEARNING

Machine Learning es una disciplina científica que maneja sistemas inteligentes, es decir que aprenden automáticamente al identificar ciertos patrones presentes en los datos. Para este aprendizaje ML usa algoritmos que se encargan de revisar datos mediante ejemplos o instrucciones predefinidas para así predecir comportamientos futuros permitiendo además la incorporación de información adicional y reajustar el resultado. ML maneja conocimiento inductivo obteniendo un enunciado general en base a enunciados que describen casos particulares (Mohri, Rostamizadeh, & Talwalkar, 2018).

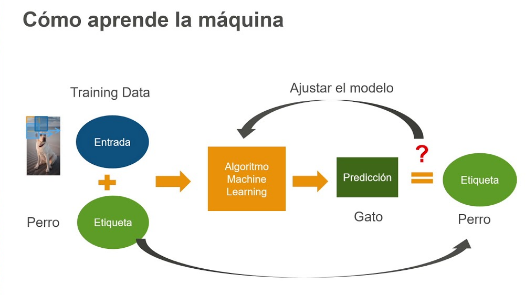


Ilustración 11. Cómo aprende la máquina (ML)

Una de las definiciones más conocidas de ML es la de Tom M. Mitchell:

*“Se dice de un programa informático que aprende de la experiencia E con respecto algún conjunto de tareas T y la medida de rendimiento P si su rendimiento en las tareas en T, medido por P, mejora con la experiencia E. ”* (Mitchell, 1997)

El proceso de ML es similar al de Data Mining, ya que los dos sistemas buscan patrones entre los datos iniciales; Data Mining extrae los datos para la comprensión humana mientras que ML utiliza esos datos para detectar patrones y reajustar las acciones del programa.

Los algoritmos utilizados en el proceso de ML deben ser eficientes y retroalimentados, los puntos a tomar en cuenta son el tiempo de ejecución y el espacio utilizado; pues al utilizar una gran cantidad de datos estos puntos pueden suponer la diferencia entre un buen y un mal modelo. La cantidad de datos necesarios para generar el algoritmo es fundamental ya que mientras mayor sea la cantidad de datos el algoritmo se vuelve más robusto y eficiente obteniendo así patrones más reales y facilitando la retroalimentación o recalibración del modelo (Rodríguez Arbonès ).

Es importante resaltar que el resultado del rendimiento de un algoritmo es bueno o malo dependiendo del contexto de negocio en el que se quiera aplicar. Por ejemplo, si se trabaja en un modelo predictivo de riesgo de Cáncer y lo queremos utilizar para informar al paciente de la probabilidad de contraer la enfermedad, podríamos informarle de que es necesaria una precisión y exactitud de más del 90% para evitar impactos emocionales en el paciente. Pero si por el contrario necesitamos la predicción para incluir al paciente en algún nivel de los programas de prevención y promoción tal vez requiera una precisión y exactitud de entre 50% y 60%. (Contreras, 2016)

El proceso a seguir para la construcción de un modelo de ML se visualiza en laIlustración 12*.*

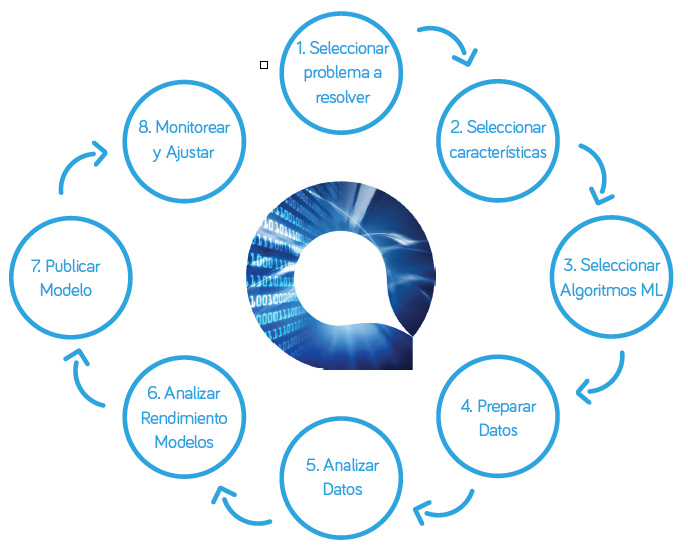


Ilustración 12. Metodología de aplicación ML (Contreras, 2016)

* Seleccionar problema a resolver: Definir el problema real que se desea resolver, analizando pros, contras, costos y beneficios de alto nivel.
* Seleccionar características: Establecer las características o variables que se utilizará para atacar el problema, se recomienda que sean de fácil acceso y extracción.
* Seleccionar modelo ML: Definir el algoritmo de ML que más se adapte a la resolución del problema planteado.
* Preparar datos: La extracción, la consolidación y el escalamiento de características que conforman el archivo de datos a utilizar.
* Analizar datos: Depurar los datos o características obtenidas y en base a los resultados seleccionar el algoritmo que puede resolver de mejor manera el problema planteado
* Analizar rendimiento del modelo (evaluación del modelo): El 60% de los datos recolectados serán datos de entrenamiento; el 20% para realizar una validación cruzada del rendimiento del modelo para depurarlo y reajustarlo; y el 20% restante de los datos recolectados para una prueba del modelo generado y comprobar el rendimiento obtenido.
* Publicar modelo: Publicar el modelo y usarlo en un ambiente productivo.
* Monitorear y ajustar: Una vez que el modelo se encuentre en producción, se debe monitorear su desempeño y hacer los ajustes necesarios para mejorarlo.

## PRINCIPALES TÉCNICAS DE MACHINE LEARNING

Los algoritmos de ML se clasifican generalmente en:

* Aprendizaje supervisado
* Aprendizaje no supervisado.
* Aprendizaje por Refuerzo.

### Aprendizaje supervisado o predictivo

La máquina aprende no sólo de los propios datos finales (inputs), sino que es posible darle modelos o datos adicionales ya categorizados (outputs) para que el aprendizaje sea mucho más fiable (Murphy, 2012).

Las aplicaciones de aprendizaje supervisado requieren algoritmos especializados que detecten patrones en los datos. Estos algoritmos pueden implementarse en lenguajes de programación como Python, pero al igual que en el análisis de contenido automatizado, si se aplican sobre grandes cantidades de datos requieren plataformas distribuidas para el procesamiento en paralelo. Para superar las dificultades que implica el desarrollo de código y el despliegue de centros de cómputo en la nube, ha prosperado una serie de servicios comerciales que permiten el aprendizaje automático de manera mucho más sencilla (Arcila Calderón, Barbosa Caro, & Cabezuelo Lorenzo, 2016).

El aprendizaje supervisado consiste en que, partiendo de una muestra:

construida por n realizaciones de un par de variables , se construye una función con la cual, dado un vector de entrada , se puede predecir con cierto grado de confianza la variable . Para cada observación de , a la variable se le llama variable de entrada, explicativa o input y a variable dependiente u output (Bourel, 2012).

Usualmente los problemas de aprendizaje supervisado se clasifican en:

1. Clasificación: se predice la clasificación de las inputs en un conjunto de categorías prefijadas como determinar que una noticia es de deportes, salud, cine, etc.
2. Regresión: predice un valor real en base a los valores que están ya almacenados. Por ejemplo, el ganador de una carrera de caballos, tomando en cuenta las carreras anteriores de los mismos.

La *Ilustración 13* muestra de manera resumida los algoritmos más utilizados en el aprendizaje supervisado.

Ilustración 13. Principales algoritmos de aprendizaje supervisado

### Aprendizaje no supervisado o descriptivo

Sólo se dan los datos finales (inputs) a la máquina para que encuentre patrones interesantes a partir de esos datos (Murphy, 2012).

A diferencia del aprendizaje supervisado, el no supervisado utiliza procedimientos inductivos, extrayendo conocimiento sólo de los datos, como en el caso del análisis de clusters para la clasificación

Los algoritmos de aprendizaje no supervisados ​​se trabajan sólo con las variables de entrada para modelar una distribución con esos datos y así obtener más información de estas input.

Los problemas de aprendizaje no supervisado se clasifican en (Unsupervised Machine Learning: What is, Algorithms, Example, s.f.):

* Clustering: dividen los datos en grupos según algunas de sus características
* Asociación: manejan reglas o características que definan grandes cantidades de datos por ejemplo en las compras en línea un usuario que suele comprar a, también suele comprar b.

La *Ilustración 14* muestra de manera resumida los algoritmos más utilizados en el aprendizaje no supervisado.

Ilustración 14. Principales algoritmos de aprendizaje no supervisado

### Aprendizaje por Refuerzo

En el aprendizaje por refuerzo (AR) el agente no cuenta con los datos de entrada y la respuesta esperada. En éste caso el algoritmo intenta obtener la mayor recompensa posible ante un determinado estado y una acción tomada para dicho estado. El agente debe descubrir que acciones le brindan la mayor recompensa ante determinado estado, una medida numérica, un número alto representa un mayor nivel de recompensa.

En ciertos casos las acciones no solamente afectan la recompensa a corto plazo, sino que también las recompensas a mediano y largo plazo se ven afectadas por dichas decisiones. (Merino, 2019)

Dos de las ventajas que representa llevar el aprendizaje por refuerzo a sistemas de Inteligencia Artificial es que el computador no se cansa y/o aburre y que es capaz de ejecutar sus tareas extremadamente rápido. Ambas características permiten obtener un gran beneficio al proceso de “prueba y error” considerando solamente aquellas decisiones que le brindaron la mejor recompensa y descartando aquellas que no fueron lo suficiente buenas permitiendo que vaya perfeccionando su estrategia.

Otra de las ventajas del aprendizaje por refuerzo es que resuelve el problema de correlacionar las acciones inmediatas con consecuencias a largo plazo (Sutton & Barto, 2018)

A continuación, en la Ilustración 15 se describen los principales componentes de un sistema de aprendizaje por refuerzo (Sutton & Barto, 2018):

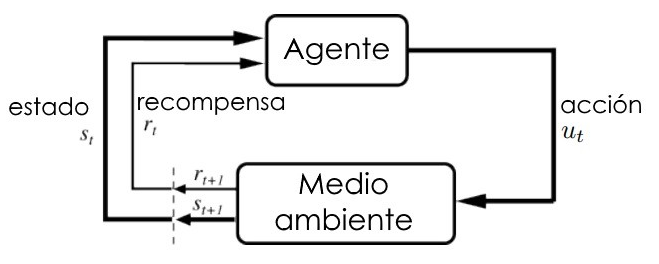


Ilustración 15. Componentes de un sistema de aprendizaje por refuerzo

El agente percibe un conjunto *s* de estados distintos en su entorno y tiene un conjunto de acciones *u* disponibles para interactuar.

En un tiempo determinado *t* el agente percibe un estado determinado *st* y ejecuta una acción posible ut obteniendo como resultado un nuevo estado st+1 = ut(st)

El entorno responde a la acción tomada por el agente con una recompensa o castigo rt+1(st, ut). Mientras mayor es el número mayor es el beneficio.

El agente no necesita conocer con anticipación el resultado de la acción tomada, no sabe qué sucederá cuando seleccione una acción determinada ante un estado definido. Un buen aprendizaje será aquel que le permita al agente determinar qué acciones le permiten obtener los resultados más eficaces y así conseguir sus objetivos (Aprendizaje por refuerzo: algoritmo Q Learning , 2019).

Ilustración 16. Principales algoritmos de aprendizaje por refuerzo

## ALGORITMOS SUPERVISADOS CLÁSICOS DE MACHINE LEARNING

### Random Forest

Random forest es un método que combina varios árboles de decisión independientes probados sobre conjuntos de datos aleatorios con igual distribución. Es decir que dichos árboles de decisión construidos a partir de datos de entrada ligeramente distintos irán alterando el conjunto inicial de datos de entrada.

**Ventajas**

* En la actualidad es uno de los algoritmos más eficientes. ​
* Maneja con mucha facilidad grandes bases de datos (Breiman, 2001).
* Puede manejar cientos de variables de entrada sin excluir ninguna.
* Estima las variables que son importantes en la clasificación.
* Tiene un método eficaz para estimar datos perdidos y mantener la exactitud cuando una gran proporción de los datos está perdida.
* Tanto la fase de entrenamiento como de test se las pueden realizar de manera simultánea.
* Con este algoritmo se puede resolver problemas de clasificación y regresión.

**Desventajas**

* Es difícilmente interpretable por el ser humano
* Random forest sobreajusta a ciertos grupos de datos con tareas de clasificación/regresión ruidosas (Segal, 2004).

### SVM (Máquinas de Soporte Vectorial)

El algoritmo de máquinas de soporte vectorial o máquinas de vectores de soporte basa su funcionamiento en la ubicación de puntos de entrenamiento (inputs) en un hiperplano o espacio vectorial de dimensión mayor, logrando determinar así los puntos (cuya clase inicialmente es desconocida) que pertenecen a una clase u otra; según la ubicación en el hiperplano de separación.

**Ventajas**

* Es una poderosa y robusta herramienta destinada a labores de clasificación.
* Por lo general SVM son utilizados en redes neuronales.
* Los algoritmos de SVM presenta sencillez en los modelos.

**Desventajas**

* EL SVM está diseñado en un principio para problemas de clasificación binaria (dos clases).
* La clasificación es algo lenta debido a su algoritmo básico de entrenamiento

### AdaBoost

El algoritmo boosting genera un clasificador muy preciso a partir de clasificadores débiles, el algoritmo AdaBoost, proviene de ***Adaptive Boosting***, y basa su funcionamiento en una estrategia de ponderar de manera diferente a cada uno de los ejemplos, durante el entrenamiento. Aquellos ejemplos que fueron clasificados incorrectamente en una iteración, reciben una ponderación más alta en la siguiente iteración. Por el contrario, los ejemplos que fueron clasificados correctamente en esa iteración, recibirán menos importancia en la siguiente. Esta estrategia permite que AdaBoost se enfoque, iteración tras iteración, en aquellos ejemplos que aún no han sido clasificados correctamente por el ensamble de clasificadores débiles.

**Ventajas**

* Es un algoritmo intuitivo.
* Sencillo de programar.
* Es robusto al sobreentrenamiento, es decir se puede entrenar al modelo por muchas iteraciones.
* El clasificar débil no requiere conocimiento previo.

**Desventajas**

* Se requieren lograr cientos de clasificadores débiles para un desempeño mínimo aceptable.
* Es vulnerable al ruido.
* Al entrenar el modelo por muchas ocasiones el algoritmo puede aprender a clasificar perfectamente los ejemplos que son usados en la etapa de entrenamiento, no obstante, se puede volver deficiente para clasificar ejemplos nuevos.

### Regresión Lineal

Conocido también como algoritmo de ajuste lineal es un modelo matemático usado para aproximar la relación de dependencia entre una variable dependiente , las variables independientes y un término aleatorio §.

Para utilizar un algoritmo de regresión lineal es necesario el cumplimiento de (Walpole, Raymond, & Myers):

* La relación entre las variables es lineal.
* Los errores en la medición de las variables explicativas son independientes entre sí.
* Los errores tienen varianza constante.
* Los errores tienen una esperanza matemática igual a cero
* El error total es la suma de todos los errores.

En conclusión, lo que se realiza es dibujar una línea recta que indique la tendencia de un conjunto de datos continuos.

**Ventajas**

* Se puede predecir el comportamiento de alguna variable en un determinado punto.
* Entrega información útil para confirmar hipótesis acerca de si dos variables están relacionadas.
* Proporciona información gráfica para probar la fuerza de una posible relación y agilizar la toma de decisiones.
* Desarrolla proyectos para la búsqueda de mejoras de calidad.

### Regresión Logística (RL)

La RL es uno de los algoritmos más utilizados en la producción científica contemporánea, es adecuado cuando la variable de respuesta es politómica es decir admite varias categorías de respuesta (De la Fuente, 2011).

**Ventajas**

* Es robusto al ruido y se puede realizar selección de atributos.
* El resultado final se puede interpretar como una probabilidad.
* Se puede usar con grandes volúmenes de datos.

**Desventajas**

* Es más útil cuando existen sólo dos posibles respuestas (variable de respuesta dicotómica).
* Puede ser entrenado con resultados satisfactorios siempre cuando los atributos son lineales y las clases idealmente reparables.

Considerando la evolución constante de la tecnología en el aprendizaje automático y el surgimiento de nuevas técnicas utilizadas por delincuentes para ejecutar transacciones fraudulentas se ve la necesidad de explotar las bondades del manejo de grandes volúmenes de datos en tiempo real aplicando herramientas de Big Data como Spark en conjunto con algoritmos de Machine Learning que permita obtener como resultado altos índices de efectividad en la detección temprana de una operación ilegítima;

Las transacciones en línea manejadas por el sistema Financiero al ser un punto crítico del negocio están sujetas a estándares internacionales como PCI-DSS (Security Standards Counsil, 2018) en el cual es mandatorio el correcto monitoreo transaccional antifraudes; brindando a los clientes seguridad en sus operaciones en línea.

# CAPÍTULO TRES

# DETECCIÓN DE FRAUDES

Para abordar el problema de detección de fraudes se han realizado varios estudios y hasta la fecha sigue siendo un gran tema de interés para la comunidad científica. A medida que pasa el tiempo las técnicas plantadas tienden a pertenecer al campo de la inteligencia artificial, redes neuronales, minería de datos e aprendizaje automático. A continuación, se presenta una tabla de las diferentes técnicas que han sido presentadas para tratar el problema de la detección de fraudes.

|  |  |
| --- | --- |
| Técnica | Área de Fraude |
| Sistemas inteligentes: redes neuronales, inteligencia difusa, algoritmos genéticos, programación genética, estrategias evolutivas y optimización por enjambres de partículas | *Telecomunicaciones, seguros, revisión de cuentas, atención médica, transacciones de tarjetas de crédito, comercio online, apuestas y verificación de identidad.* |
| Técnicas inspiradas en la naturaleza. | *Email, spam, phishing e intrusion en redes.* |
| Técnicas de minería de datos sobre series espaciales y temporales | *Seguros médicos* |
| Técnicas basadas en reglas y técnicas de aprendizaje supervisado y no supervisado | *Llamadas sobre IP (VoIP)* |
| Minería de datos y estadística | *Detección del fraude financiero, seguro de hogar, seguro de motos y seguros médicos* |



## **SISTEMAS DE DETECCIÓN** DE FRAUDES (SDF)



La detección de fraudes en tarjetas de débito y crédito se basa en el análisis de las transacciones almacenadas. Los datos de las transacciones contienen una gran cantidad de atributos que dificultan la detección manual de patrones o comportamientos de fraude.

A continuación, se realiza un análisis de la detección de fraudes basado en el conocimiento (Expert Driven) y la detección de fraudes basado en los datos (Data Driven) (Dal Pozzolo A. , 2015)

Los métodos ED utilizan el conocimiento y experiencia de investigadores de fraudes para definir un conjunto de reglas que permiten predecir la posibilidad de una transacción fraudulenta. Así por ejemplo según su experiencia pueden definir una regla donde si el monto de una transacción es superior a USD 20.000 y proviene de un sitio web de compras en línea tiene una probabilidad de 0.99 de que sea un incidente fraudulento; de esta manera se puede definir un determinado número de reglas para diferentes escenarios. Este tipo de reglas se pueden clasificar como reglas bloqueantes y reglas de probabilidad. Los expertos asignan un número que representa el nivel de probabilidad de una incidencia fraudulenta, luego definen el bloqueó de las operaciones cuya puntuación sobrepase un límite que implique un alto riesgo de fraude. Aunque la principal ventaja de las reglas ED es su sencillez para definirlas y entenderlas también presenta algunas desventajas como:

* Pueden llegar a ser subjetivas (depende del investigador o experto).
* Analizan solamente correlaciones simples entre los atributos de la operación.
* Pueden llegar a detectar solamente patrones conocidos de fraude.
* Algunas reglas pueden volverse obsoletas debido a la evolución de las técnicas de fraude.

Los métodos orientados a Datos (DD) en cambio se basan en utilizar técnicas de Machine Learning para aprender a reconocer patrones de operaciones fraudulentas en base al análisis los datos de operaciones anteriores. Igualmente, estos métodos presentan algunas ventajas:

* Puede encontrar correlaciones complejas entre los datos.
* Puede aprender de grandes cantidades de datos.
* Llega a reconocer anomalías en operaciones genuinas permitiendo detectar nuevos tipos de fraude.
* Se adapta a la evolución de los fraudes en el tiempo.

Los métodos DD también presentan algunas desventajas:

* Es necesario una gran cantidad de datos para entrenamiento
* Los modelos no son fácilmente interpretables.

Mientras en los Sistemas de Detección de Fraudes utilizan los métodos DD para reducir el número de falsos positivos obteniendo así alertas precisas; los métodos ED se utilizan para reducir el número de falsos positivos y asegurarse que todos o la mayoría de casos fraudulentos sean detectados. La combinación de ambas estrategias es la que ofrece la mejor solución.

El diseño de un SDF basado en modelos DD presenta tres grandes características que se deben tomar en cuenta y resolver:

* El porcentaje de transacciones fraudulentas es muy pequeño en relación al total de transacciones dadas en un periodo de tiempo. También conocida como “problema de desbalanceo de datos” donde la mayoría de las transacciones tienden al estado genuino y muy pocas son tomadas como transacciones fraudulentas. Esto representa un problema ya que la mayoría de algoritmos de ML no están diseñadas para trabajar con datos fuertemente desbalanceados.
* La constante evolución de las estrategias para realizar fraude en transacciones bancarias. Debido al constante cambio del comportamiento financiero de los clientes y la evolución de las técnicas empleadas para realizar fraudes los SDF tienen que estar constantemente actualizándose mediante el procesamiento de la nueva información y desechando información que ya no es útil en el proceso de detección de los nuevos patrones de comportamiento. Los FDS que no son actualizados o revisados frecuentemente pierden precisión en sus predicciones y llegan a ser obsoletos a largo plazo.
* El estado real (genuino o fraudulento) de la mayoría de las transacciones es conocida algunos días después de sucedida. Debido a la gran cantidad de transacciones generadas diariamente en el sistema financiero, es imposible que todas estas transacciones sean verificadas por el limitado número de recursos humanos destinados a esta labor. Los SDF deben generar un determinado número de alertas a ser revisadas por los investigadores, los mismos que luego de contactarse con el cliente respectivo pueden etiquetar la transacción como fraudulenta o genuina. Este reducido número de transacciones que están disponibles en un tiempo casi inmediato es la única información “en tiempo real” que puede ser utilizado para entrenar y actualizar el modelo de ML. El resto de transacciones van a ser útiles cuando haya transcurrido varias semanas y al no tener queja de los clientes puedan ser etiquetadas como transacciones legítimas

A continuación, se describe la arquitectura de un Sistema de Detección de Fraudes y sus diferentes capas. Como puede ver la *Ilustración 17* un FDS consta de 5 capas; los primeros 4 elementos son automatizados mientras que el último requiere de intervención humana para tomar las alertas y generadas y confirmar o descartar el posible fraude.

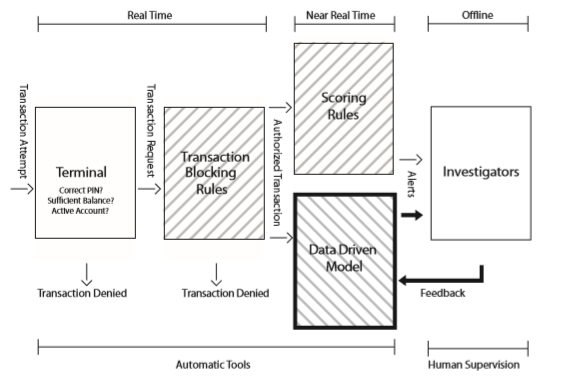


Ilustración 17. Elementos de un Sistema de Detección de Fraudes

1. *Terminal*: Representa la primera capa en el SDF. En esta fase de seguridad se verifica el estado de la tarjeta, el pin correcto, el número de intentos y si el cliente tiene el cupo o saldo necesario para procesar la transacción. Todas las validaciones son realizadas en línea y si alguna falla la transacción es rechaza inmediatamente.
2. *Transaction Blocking Rules*: Las reglas son definidos por investigadores expertos en fraudes para bloquear las transacciones antes de su procesamiento. También son verificadas en línea, son muy claras y precisas y bloquean la transacción en caso de que estas reglan detecten un intento de fraude. Las transacciones que pasen este filtro de seguridad son autorizadas y el pago es ejecutado. Para enviar la transacción al siguiente componente dentro el sistema los datos son enriquecidos con información adicional (por ejemplo: promedio de consumo mensual, número de transacciones al día del cliente, etc).
3. *Scoring Rules (EDR)*: Son reglas generadas por expertos en base a su experiencia. Este componente actúa como un clasificador basado en reglas que asigna una calificación para cada transacción; mientras más alta es la calificación más alta es la probabilidad de que se trate de una operación fraudulenta. Contiene reglas de tipo IF que son fáciles de definir y entender (por ejemplo: Si el monto es mayor a 5000, el lapso de tiempo transcurrido desde última transacción del cliente es menor a 5 minutos y el país de origen de la operación es diferente a la anterior: score=0.9). Aunque son reglas simples requieren revisión y ajuste manual cuando la precisión del modelo empiece a bajar.
4. *Data Driven Model(DDM)*: Consiste en un modelo predictivo para asignar una calificación a cada transacción. Aplica algoritmos de ML para poder estimar la probabilidad de una transacción fraudulenta. El modelo es más robusto que las EDR, pero su interpretación es muy compleja por lo que vuelve una caja negra. Los DDM son capaces de considerar todo el conjunto de características de la transacción mientras que las condiciones de los EDR solo consideran algunas de ellas.
5. *Investigators*: Son las personas encargadas de verificar las alertas generadas tanto por EDR como por DDM. Debido a la limitación del número de personas encargadas a ésta tarea el número de alertas a ser revisadas deben ser solamente las de más alta calificación. El verificador debe comunicarse con el cliente y luego de comprobar la autenticidad de la transacción debe etiquetar la respectiva operación como genuina o fraudulenta, el resultado de aplicar una etiqueta a la operación se lo conoce como *feedback*. Al ser un proceso largo y tedioso los SDF deben generar la menor cantidad de falsos positivos. El feedback generado por los investigadores es utilizado por DDM para continuar con el proceso de aprendizaje y evolución continua de sus modelos.

### Entrenamiento Supervisado

Las transacciones con el feedback de los investigadores representan una pequeña parte del total de transacciones realizadas cada día. El resto de operaciones procesadas apenas van pueden ser etiquetadas varios días después cuando pase el periodo que el cliente necesita para reportar una transacción no reconocida luego de revisar su estado de cuenta y se pueda asumir que las transacciones son genuinas. Una vez ocurrido se puede usar éstas transacciones para entrenar un nuevo DDM.

### Actualización de los modelos

La actualización de un EDM se realiza de forma manual mientras que los DDM deben ser actualizados automáticamente. En los componentes DDM la información de feedback de los expertos obtenida durante el día es usado para entrenar y actualizar el modelo que calificará las transacciones del día siguiente.



### Inclusión de características adicionales a la data de la transacción

Cuando una transacción ha sido autorizada, los datos de ésta operación son almacenados en una BD con las características propias de esa autorización (ID/Numero, monto, fecha, adquiriente, etc). A partir de estos datos es posible calcular nuevas variables que pueden ser agregadas al conjunto de datos original que permitirá enriquecer la información antes de que ser procesada en módulos de EDM y DDM. A este proceso se lo puede llamar como “enriquecimiento de características (feature augmentation)”. Este proceso permite aumentar características como promedio de monto, número de transacciones del cliente en el mes, número de transacciones en la misma tienda, etc.

La información agregada permite tener una visión del comportamiento del cliente en un determinado lapso de tiempo (semanal, mensual, anual); El componente DDM ocupará esta información para detectar situaciones anómalas al comportamiento de los usuarios. El conjunto de características originales y agregadas de la transacción es conocido como “*vector de características*”.

Ya que el identificador de la tarjeta/cliente es utilizado únicamente para poder calcular las características adicionales, una vez obtenidas, se debe eliminar éste dato del vector de características con el fin de que el entrenamiento del modelo sea realice con operaciones completamente independientes.



### Como medir la precisión de un SDF

El rendimiento de los sistemas de detección no puede ser medido en términos del rendimiento de los sistemas de clasificación. En un sistema de detección lo que más importa es que el sistema pueda calificar correctamente la pequeña cantidad de incidencias útiles (fraudulentas) TP por sobre el resto de las operaciones. Debido a la limitación de recursos los investigadores no pueden verificar todas las transacciones que hayan sido marcadas como fraudulentas, sino enfocarse en aquellas *n* transacciones que tienen la calificación más alta (n está definido en función a la cantidad de investigadores destinados al proceso de verificación). El objetivo de un buen sistema de detección no es predecir la etiqueta de clasificación para cada operación entrante, sino retornar un número de calificación correcta para aquella clase que importa (fraude).

En este contexto un buen algoritmo de detección es aquel que es capaz de calificar con un valor alto a las incidencias relevantes (transacciones fraudulentas) y calificar con un valor bajo a aquellas incidencias no importantes (transacciones genuinas).

Una buena medida para los sistemas de detección es conocida como Average Precision (AP) (Fan & Zhu, 2011).

Donde *N* es el número total de observaciones en el conjunto de datos y se define el valor de Precisión (*Pk*) y Recall (*Rk*) como:

Donde *N+* es el número de casos positivos (fraudes) en el conjunto de datos original y *TPk* es el número de verdaderos positivos en las primeras k posiciones de la lista de calificaciones.

Un algoritmo que pueda detectar todos los fraudes de un conjunto de datos con la más alta calificación será aquel que obtenga un AP = 1.



## ALGORITMOS PARA DETECCIÓN DE FRAUDES

Para el caso de detección de fraudes en transacciones financieras se han propuesto varios algoritmos ML *supervisados* y *sin supervisión* [ (Brause, Langsdorf, & Hepp, 1999), (Fan, Prodromidis, Stolfo, & Chan, 1999), (Bhattacharyya, Jha, Tharakunnel, & Westland, 2011)]. En el caso de los algoritmos supervisados el modelo es entrenado con datos ya clasificados para descubrir patrones asociados a transacciones genuinas y transacciones fraudulentas. Por otro lado, los algoritmos sin supervisión permiten detectar anomalías en los datos que no corresponden con los de la mayoría de las transacciones. Las técnicas no supervisadas generalmente generan una mayor cantidad de alertas falsas por lo que es conveniente utilizar una combinación de ambos métodos.

Para el tema de detección de fraudes se debe afrontar el problema desde dos puntos de vista: a nivel de cliente y a nivel de transacción.

Los modelos basados en el comportamiento del cliente utilizan métodos “sin supervisión” que tratan de encontrar el comportamiento habitual del cliente en el tiempo. Estos modelos solo consideran el historial anterior de cada tarjeta, pero no intentan identificar patrones globales de comportamientos fraudulentos; solo intentan detectar cambios en el comportamiento. El problema es que un cambio de comportamiento no necesariamente se debe a una actividad fraudulenta, por ejemplo, en época de navidad el cliente puede cambiar sus hábitos de gastos en comparación con el resto del año.

Los modelos que trabajan en un nivel transaccional intentan estimar la posibilidad de fraude en una transacción sin conocer el comportamiento de la tarjeta o el cliente. Esta estrategia solo considera transacciones independientes sin tomar en cuenta el historial de las transacciones asociadas al cliente o cuenta. Como alternativa, es posible trabajar a nivel de transacción e incluir el comportamiento del cliente en el modelo agregando variables de comportamiento como se indicó en la sección anterior.



## Métodos supervisados

En los algoritmos supervisados se utilizan ejemplos fraudulentos y no fraudulentos para predecir la clase de una nueva observación. Las técnicas supervisadas desarrolladas para la detección de fraudes pueden agruparse en: métodos de perfilado, clasificación, orientado al costo.

*Métodos de perfilado*

Cuando las transacciones etiquetadas están disponibles, es posible establecer un perfil de la distribución de las variables relevantes de una operación genuina o fraudulenta. Esto significa que es posible crear diferentes perfiles para cada clase. En este punto, cada nueva transacción se puede comparar para ver qué perfil es más similar.

Los perfiles también se pueden crear mediante métodos basados en reglas. Las reglas pueden ser definidas a partir de expertos en fraude o modelos estadísticos y tienen la ventaja de ser fáciles de entender e implementar. Se define un conjunto de reglas para cada perfil, y si una nueva transacción coincide con estas reglas, se asume que tiene el mismo perfil.

A medida que las actividades delictivas y el comportamiento genuino del cliente evolucionan, los perfiles fraudulentos también tienen que actualizarse. Esto significa que las reglas basadas en estadísticas deben actualizarse periódicamente. Adicionalmente se pueden ir agregando nuevas reglas mientras que se actualizan las reglas actuales. Los perfiles deben actualizarse para reflejar los patrones dinámicos de actividad delictiva, así como los cambios en el comportamiento legítimo del usuario.

*Métodos de clasificación*

La clasificación es la manera más común de abordar el problema la detección de fraudes. Entre los algoritmos de clasificación que se vienen utilizando se tiene: redes neuronales, regresión logística, reglas de asociación, máquinas de vectores de soporte y árboles de decisión.

Los árboles de decisión han encontrado muchas aplicaciones en la detección de fraudes, tanto para la detección de fraudes en tarjetas de crédito como para la puntuación de riesgo crediticio. Sin embargo, como se explicó anteriormente, los métodos supervisados en general sufren el problema de los tamaños de clase no balanceados: las transacciones genuinas superan por mucho en número a las fraudulentas.

Las redes neuronales presentan la desventaja de ser cajas negras, ya que no es posible que un ser humano entienda cómo se comportan. Reglas de asociación y árboles de decisión son ampliamente utilizados en la detección de fraude porque a diferencia de las técnicas de caja negra permiten una mejor comprensión de la clasificación. Esto es importante ya que los analistas, comúnmente, no son especialistas en aprendizaje automático y necesitan entender los mecanismos de clasificación para poder confiar en ellos y usarlos.

También se plantea el uso del modelo gráfico probabilístico en la detección de fraudes, como las Redes de Creencias Bayesianas (BBN).

Entre todos los algoritmos propuestos en la literatura, se recomienda el uso de *Random Forest*, ya que varios estudios han demostrado que logra los mejores resultados entre los diferentes clasificadores [ (Whitrow, Hand, Juszczak, Weston, & Adams, 2009), (Bhattacharyya, Jha, Tharakunnel, & Westland, 2011), (Correa Bahnsen, Stojanovic, Aouada, & Ottersten, 2013), (Viola & Jones, 2001)].

*Métodos sensibles al costo*

Dado que las empresas están interesadas en reducir la pérdida monetaria debida a actividades fraudulentas, hay un gran número de algoritmos de clasificación sensibles a costos. Estos algoritmos son capaces de tener en cuenta los diferentes costos de clasificar erróneamente una transacción como fraude o legítimo. El aprendizaje sensible al costo es una forma alternativa de abordar el problema de la clasificación desbalanceada que consiste en asignar mayores costos a los errores que se han cometido en la clase minoritaria. Algunos de algoritmos sensibles al costo disponibles en la actualidad son: AsymBoost, AdaCost, CSB, DataBoost, AdaUBoost y SMOTEBoost [ (Fan, Prodromidis, Stolfo, & Chan, 1999), (Ting, 2000), (Guo & Viktor, 2004), (Shawe-Taylor, 1999), (Chawla, Lazarevic, Hall, & Bowyer, 2003)].

Los algoritmos tradicionales, como AdaCost, asumen que los costos son fijos y dependientes de la clase, sin embargo, en detección de fraudes, el costo es proporcional al importe de la transacción. Cuanto mayor sea la cantidad, mayor será la pérdida potencial en caso de fraude. El costo de un falso negativo no es fijo, sino proporcional al importe de la transacción. En un trabajo desarrollado por (Mahmoudi & Duman, 2015) se propone utilizar el análisis discriminante modificado de Fisher para considerar el costo dependiente de ejemplo para cada transacción a fin de maximizar el beneficio total. Una alternativa es utilizar árboles de decisión dependientes de los costos para maximizar los ahorros [ (Sahin, Bulkan, & Duman, 2013), (Correa Bahnsen, Stojanovic, Aouada, & Ottersten, 2013)]. Los métodos basados en el costo son útiles cuando el objetivo principal es minimizar algunos costos y maximizar los beneficios de la detección o el ahorro ya que la función de pérdida tiene en cuenta la pérdida financiera ocurrida en cada predicción.

## Métodos no supervisados

Los métodos no supervisados se utilizan cuando no hay conjuntos previos de observaciones legítimas y fraudulentas. Puesto que no se basan en ejemplos de fraude o transacciones genuinas, las estrategias no supervisadas tienen la ventaja de ser independientes de su selección, y son capaces, en teoría, de descubrir fraudes aún no observados, que no han sido detectados por un experto. Sin embargo, no se ven afectados por el problema del desbalanceo de datos.

Las técnicas empleadas por la detección de fraudes pueden contener una combinación de métodos de detección basado en perfiles y anomalías. Modelan una línea base que representa el comportamiento normal y luego intentan detectar las observaciones que muestran la mayor desviación de esta línea base. Uno de estos métodos es *Peer Group Analysis* que agrupa a los clientes en diferentes perfiles e identifica a las transacciones que salen del perfil del cliente como fraudulentas. Otros modelan el comportamiento del cliente por medio de mapas auto-organizativos.

Para detectar valores atípicos o anomalías, primero es importante definir cuándo un ejemplo es un valor atípico:

"U*na observación marginada, o valor atípico, es una que parece desviarse marcadamente de otros miembros de la muestra en la que se produce*"[[2]](#footnote-2)

"U*na observación (o subconjunto de observaciones) que parece ser incompatible con el resto de ese conjunto de datos*".[[3]](#footnote-3)

Cuando es posible definir una región o distribución de los datos que representan el comportamiento normal, todas las observaciones que caen fuera pueden marcarse como atípico. Este enfoque, sin embargo, viene con varios desafíos:

* Definir una región normal que abarque todos los comportamientos normales posibles es muy difícil,
* El comportamiento normal sigue evolucionando y una noción existente de comportamiento normal podría no ser suficientemente representativa en el futuro,
* El límite entre el comportamiento normal y el marginado generalmente es muy difuso,
* Los fraudes se adaptan para hacer que las observaciones periféricas parezcan normales, haciendo así que la tarea de definir el comportamiento normal sea más difícil.

Los valores atípicos se pueden clasificar en 2 grupos: los valores atípicos locales y valores atípicos globales.

Un valor atípico global es una observación anómala para todo el conjunto de datos. En el caso de la detección de fraudes, un ejemplo de valor atípico global puede ser una transacción de USD 100.000 cuando todas las demás transacciones del conjunto de datos tienen una cantidad menor.

Un valor atípico local es una observación que parece anómala en comparación con los subgrupos de los datos. Por ejemplo, cuando una tarjeta se utiliza principalmente en un área geográfica restringida, una compra realizada por la misma tarjeta en el país extranjero es un valor atípico local dentro de sus transacciones, pero no es anómala con respecto a todas las transacciones posibles.

# CAPÍTULO 4

# PLANTEAMIENTO DE LA SOLUCIÓN



## Arquitectura de la propuesta

Los Sistemas de detección de Fraudes se basan en operaciones manuales y automáticas. Las operaciones manuales son realizadas por personal de la entidad financiera denominadas investigadores de fraude, mientras que los componentes automáticos se implementan mediante algoritmos que funcionan en tiempo real y casi tiempo real. Las operaciones en tiempo real tienen lugar antes de que se autorice el pago, mientras que las operaciones casi en tiempo real se ejecutan después de que se produjo el pago.

La propuesta planteada se implementa para las operaciones cuya autorización ha sido emitida y propone un modelo de ML corriendo sobre componentes basados en herramientas estándar del ecosistema de Apache:

* Kafka[[4]](#footnote-4)
* Spark Streaming[[5]](#footnote-5)
* Cassandra[[6]](#footnote-6)

Se escogieron estas herramientas ya que corren en un mismo ecosistema y presentan similares sistemas de tolerancia a fallos y tareas distribuidas. Además, los componentes planteados permiten escalamiento horizontal cuando se tenga una gran cantidad de transacciones a ser procesadas.

A continuación, se describen las principales características de los componentes utilizados en la propuesta, así como sus principales ventajas y desventajas.

### Apache Kafka

La plataforma de Apache Kafka se trata de un sistema de mensajes Open Source de tipo publish-subscribe basado en una arquitectura P2P (Peer to Peer). Tiene un sistema de administración capaz de recuperar mensajes de múltiples fuentes. Es un sistema de mensajería distribuido, particionado y replicado. Estas características, añadidas al que es muy rápido en lecturas y escrituras lo convierten en una herramienta excelente para comunicar streams de información que se generan a gran velocidad y que deben ser gestionados por una o varias aplicaciones. Entre las principales características a destacar se mencionan:

* Funciona como un servicio de mensajería, categoriza los mensajes en topics.
* Los procesos que publican se denominan brokers y los subscriptores son los consumidores de los topics.
* Utiliza un protocolo propio basado en TCP y Apache Zookeeper para almacenar el estado de los brokers. Cada broker mantiene un conjunto de particiones (primaria y secundaria) de cada topic.
* Se pueden programar productores/consumidores en diferentes lenguajes: Java, Scala, Python, Ruby, C++, etc.
* Es escalable y tolerante a fallos.
* Se puede utilizar para servicios de mensajería (tipo ActiveMQ o RabbitMQ), procesamiento de streams, web tracking, trazas operacionales, etc.
* Escrito en Scala.
* Creado por LinkedIn.
* Tiene mucha popularidad en la actualidad (muchas grandes empresas lo tienen incorporado en sus arquitecturas: Netflix, Microsoft, Bancos, aseguradoras, empresas ticketing, etc).
* Dispone de un API de "Producer". Facilita que una aplicación publique una secuencia de mensajes en uno o más topics de diferentes formas.
* Dispone de un API de "Consumer". Facilita que una aplicación pueda suscribirse a uno o más topics y así poder procesar la secuencia de mensajes.
* Dispone de un API de "Streams". Facilita procesar un flujo consumiendo un flujo de entrada de uno o más topics y produciendo un flujo para uno o más topics de salida.
* Dispone de un API de "Connector". Permite implementar/ejecutar productores o consumidores reutilizables con el objetivo de conectar topics con aplicaciones o sistemas de datos existentes.
* En caso de necesidad (por ejemplo, interrupción del sistema), las transacciones pueden recuperarse también durante un intervalo de tiempo (establecido por el usuario) posterior a su procesamiento.

### Apache Spark

Apache Spark es una tecnología de computación en clúster excepcionalmente diseñada para el cómputo rápido. Básicamente Spark una implementación de Map-Reduce en memoria, habilitada para streaming, que distribuye automáticamente el cálculo entre los recursos asignados y agrega los resultados en un sistema de archivos distribuido.

Spark es una de las empresas de Hadoop creada en 2009 en AMPLab de UC Berkeley por Matei Zaharia. Fue Open Sourced en 2010 bajo una licencia BSD.

Spark proporciona un conjunto de interfaces API para lenguajes de alto nivel, por ejemplo: Java, Scala, Python y R para el desarrollo de aplicaciones. Spark utiliza Hadoop de dos maneras diferentes: una es para almacenamiento y la segunda es para el manejo de procesos.

El objetivo de Apache Spark de lograr un manejo más rápido de la información se pudo lograr gracias a la manera en que fue diseñado en base a sus componentes (Ilustración 18).



Ilustración 18. Componentes de Apache Spark

**Apache Spark Core**

Spark Core es el motor de ejecución general básico para la plataforma Spark en el que se basan todas las demás funcionalidades. Proporciona registros en memoria y conjuntos de datos conectados en marcos de almacenamiento externos.

**Spark SQL**

Spark SQL es un segmento sobre Spark Core que presenta otra abstracción de información llamada SchemaRDD, que ofrece ayuda para sincronizar información estructurada y no estructurada.

**Spark Streaming**

Spark Streaming utiliza la capacidad de programación rápida de Spark Core para realizar Streaming Analytics. Procesa información en grupos reducidos y realiza cambios RDD (conjuntos de datos distribuidos resilientes) en esos grupos de información a pequeña escala.

**MLIB (Machine Learning Library)**

MLlib es una estructura de aprendizaje automático distribuido sobre Spark en vista de la arquitectura distribuida basada en memoria de Spark. Spark MLlib es nueve veces más rápido que la versión de disco Hadoop de Apache Mahout.

**GraphX**

GraphX ​​es un marco distribuido de procesamiento de gráficos de Spark. Proporciona una API para comunicar el cálculo del gráfico que puede mostrar los diagramas caracterizados por el cliente utilizando la API de abstracción Pregel. Asimismo, proporciona un tiempo de ejecución optimizado y mejorado.

Apache Spark, principalmente diseñado para Data Science, es considerando el más grande proyecto Open Source para procesamiento de datos; a continuación, se describen algunas de sus principales características:

* Rápido en procesamiento de datos, 10 veces más rápido en disco y 100 veces más rápido en memoria.
* Su abstracción RDD proporciona tolerancia a fallos garantizando ninguna pérdida de datos.
* Soporte para múltiples lenguajes de programación.
* Procesamiento en memoria que resulta en alta velocidad de cómputo y flujo de datos acíclico.
* Procesamiento de flujo de datos en tiempo real con Spark Streaming.
* Flexible para funcionar de forma independiente y se puede integrar con Hadoop Yarn Cluster Manager.
* Rentable para Big Data como una necesidad mínima de almacenamiento y centro de datos.
* Análisis futurista con herramientas integradas para aprendizaje automático, consultas interactivas y transmisión de datos.
* Comunidad de Apache activa y en expansión para asistencia rápida



### Apache Cassandra

Cassandra es una popular base de datos NoSQL de código abierto. Es una base de datos distribuida diseñada para proporcionar escalabilidad, capaz de admitir la replicación en múltiples nodos o centros de datos. Ofrece escalabilidad lineal, tolerancia a fallos, baja latencia al realizar consultas. Los datos se almacenan en múltiples nodos organizados en forma de anillo (es decir, no hay un maestro y cada nodo es tan importante como los demás), evitando así un solo punto de falla. Ilustración 19



Ilustración 19. Arquitectura de Cassandra.

Cassandra está siendo utilizando con éxito en una variedad de escenarios como análisis, análisis de series de tiempo, monitoreo, venta minorista, comercio electrónico, etc. Generalmente se usa en entornos con altos volúmenes de escritura.

A continuación, se mencionan algunas de las características más importantes de Cassandra:

* Está diseñada para ejecutarse en un grupo de nodos para proporcionar alta disponibilidad, tolerancia a fallas y escalabilidad.
* Trabaja en modo master-less o multi-master. Las escrituras se distribuyen entre los nodos utilizando una función hash y las lecturas se canalizan a nodos específicos.
* En Cassandra, si un nodo se cae, las escrituras se redirigen hacia otros nodos y el sistema continúa funcionando.
* Debido a su arquitectura multi-master, Cassandra es linealmente escalable, duplicando el número de nodos en un clúster.
* Almacena las escrituras entrantes en RAM para proporcionar un mejor rendimiento.
* Cassandra se considera un sistema de alta disponibilidad y tolerante a la partición.



## Proceso

La construcción de un Sistema de Detección de Fraude se divide en dos fases:

* + La primera fase implica la construcción de un modelo que permita obtener la probabilidad de una transacción fraudulenta.
  + La segunda fase la implementación de éste modelo en un ambiente de tiempo real que permita hacer las predicciones sobre transacciones entrantes en vivo.

### Construcción del Modelo

Para proceder a la construcción de un correcto modelo de aprendizaje automático supervisado se deben seguir los siguientes pasos:

1. Realizar el proceso de feature engineering (ingeniería de características) para transformar los datos históricos en un conjunto de características y etiqueta de clasificación para ser utilizado en un algoritmo de aprendizaje automático supervisado.
2. Dividir el conjunto de datos disponible en dos partes, una para construir el modelo y otra para probar el modelo.
3. Construir el modelo con las características y etiquetas de entrenamiento.
4. Probar el modelo con el conjunto de datos de prueba para obtener predicciones y luego comparar las predicciones obtenidas del modelo con las etiquetas del conjunto de pruebas.
5. Ajustar el modelo hasta obtener un índice de precisión deseado.



Ilustración 20. Flujo construcción de un modelo de aprendizaje automático

**Feature Engineering**

Feature Engineering (ingeniería de características) hace referencia al proceso de transformar datos sin procesar en entradas para un algoritmo de aprendizaje automático. La ingeniería de características depende en gran medida del tipo de caso de uso y las posibles fuentes de datos.

Para el caso de fraudes en tarjetas de crédito las características de cada transacción se pueden dividir en:

* Características asociadas a la transacción: fecha y hora, monto, comercio, etc.
* Características asociadas a la tarjeta y su dueño: número de tarjeta, tipo de tarjeta, edad del dueño, etc.
* Características obtenidas a partir del histórico de transacciones: número de transacciones en un periodo de tiempo, promedio de monto, suma del monto de las transacciones, etc.



Ilustración 21. Características consideradas para la construcción del modelo



### Implementación del modelo

La figura a continuación se muestra un diagrama de la arquitectura a alto nivel de la solución propuesta. Los eventos de las transacciones son entregados a través del sistema de mensajería de Kafka. Spark streaming procesa y verifica las transacciones en busca de características fraudulentas utilizando MLib con el modelo implementado y asigna una probabilidad de fraude. Luego se utiliza Cassandra para almacenar las transacciones enriquecidas con los datos agregados y almacena la tabla con el top N de las transacciones con mayor índice de probabilidad de fraude. Esta tabla es luego tomada por el sistema monitor que muestra los datos al personal encargado de comunicarse con el dueño de la tarjeta y etiquetar la transacción como genuina o fraudulenta que a posterior será utilizado en el siguiente periodo de entrenamiento como feedback.



Ilustración 22. Arquitectura implementación del modelo



## La estrategia de Aprendizaje

La solución propone un FDS donde la herramienta permite la interacción con los investigadores de fraude. El papel de los investigadores de fraude es centrarse en las transacciones más sospechosas y contactar a los titulares de tarjetas. Para esto el sistema automático recibe una retroalimentación (legítimo o fraude) de solo un pequeño subconjunto de transacciones que activaron una alerta. Para el resto de las transacciones no se reciben comentarios a menos que el titular de la tarjeta informe un fraude. Esto significa que se puede suponer que las transacciones no alertadas son genuinas solo después de un tiempo. Ésta estrategia permite integrar los datos de las transacciones para las que se tiene el feedback de los investigadores y las reportadas por el cliente con cierto retraso.

La clasificación se basa en el algoritmo de Random Forest, que ha demostrado ser particularmente eficaz en escenarios de detección de fraude [ (Whitrow, Hand, Juszczak, Weston, & Adams, 2009), (Bhattacharyya, Jha, Tharakunnel, & Westland, 2011), (Correa Bahnsen, Stojanovic, Aouada, & Ottersten, 2013), (Viola & Jones, 2001)]..

La estimación del riesgo final de una transacción está dada en función de los resultados de la probabilidad de una transacción positiva (fraudulenta) para cada uno de los 2 clasificadores:

* Un clasificador basado en Random Forest *(Ft)* que considera las observaciones generadas en los últimos n días y para el cual los investigadores devolvieron un feedback
* Un clasificador *“Delayed” (Dt)* basado en Balanced Random Trees (BRTs) entrenado con las transacciones anteriores para los cuales luego de n días se puede asumir razonablemente que tienen etiquetas conocidas. Este clasificador necesita un conjunto de datos mucho más grande para su entrenamiento

Aunque los dos clasificadores se actualizan una vez al día, son utilizados continuamente en el módulo de “scoring” para evaluar el riesgo de fraude.



Ilustración 23.- Ventana de tiempo de cada uno de los modelos (Feedback y Delayed)

Para el entrenamiento de los modelos se utiliza un enfoque de ventana deslizante para actualizar F*t* y *Dt* en base a nuevas transacciones. El modelo *Ft* es entrenado con las transacciones de los últimos n días; mientras que el modelo *Dt* implementa una estrategia de actualización que mantiene los datos correspondientes al resto de días y descarta los más antiguos.

Dada una transacción entrante *i* en el tiempo *t*, codificada por un vector de características *xi*, los clasificadores *Ft* y *Dt* producen respectivamente las probabilidades posteriores *PFt (+ | xi) y PDt (+ | xi)*, donde + denota un evento fraudulento. La probabilidad posterior agregada *PAt (+ | xi)* se obtiene mediante un promedio ponderado de probabilidades posteriores de los clasificadores individuales:

donde *At* es el modelo general que envuelve *Ft* y *Dt;* y *wA* = 0.5 (Dal Pozzolo, Boracchi, Caelen, Alippi, & Bontempi, 2015).

A continuación, se describe el flujo del proceso dado un DStream, para cualquiera de sus componentes RDD:

* Si el día ha terminado se vuelven a entrenar los modelos F y D, se guarda en Cassandra las alertas topN del día, se encera AlertTable y descartan las transacciones innecesarias;
* Para cualquier intervalo de tiempo dado en la ventana de tiempo definida, se recupera la información sobre transacciones anteriores de la tabla de transacciones tableTrx en base a su identificador de la tarjeta.
* Una vez que las características de la transacción han sido enriquecidas, la transacción puede clasificarse de acuerdo a los modelos F y D y las alertas topN más riesgosas actualizadas en la tabla de alertas.
* Esta tabla de alertas va a ser leída posteriormente por el monitor de alertas para que los investigadores puedan contactar al titular de la tarjeta agregando así el feedback respectivo.

## Monitoreo

En este módulo, las transacciones se ponen a disposición de los investigadores en un panel de monitoreo de transacciones riesgosas, donde pueden ver el puntaje de fraude devuelto por la función de predicción y algunas características originales de la transacción, como el identificador de la tarjeta, fecha y hora de la transacción, etc. En el monitor, las transacciones se clasifican según su puntaje de fraude para que los investigadores puedan revisar las más sospechosas. Cuando el investigador desea contactarse con el dueño de la tarjeta selecciona la fila y aparecerá una ventana con los datos de la transacción y los datos de contacto del tarjeta-habiente. Después de la verificación, la transacción se marca como genuina o fraudulenta y se convierte en una retroalimentación o feedback. Las transacciones que no son revisadas por los investigadores permanecen sin etiquetar durante *n* días y luego, si los titulares de la tarjeta no las informan como fraudulentas, son etiquetadas como genuinas.

A continuación, se adjuntan las pantallas del módulo de monitoreo de las transacciones más riesgosas.

En la Ilustración 24 se presenta la captura de una pantalla del módulo de monitoreo con el listado de las transacciones con un score más alto, esta pantalla es utilizada por las personas encargadas de comunicarse con el cliente y verificar la legitimidad de las transacciones; los datos que se muestran en la pantalla muestran al usuario un primer nivel de información importante de la transacción como la fecha y hora de la transacción, el monto de la operación y los scores obtenidos de cada clasificador y del promedio de los 2 clasificadores.



Ilustración 24. Captura de pantalla del listado de transacciones sospechosas en el monitor

En la Ilustración 25 se muestra el detalle de una transacción y el botón para que el usuario consulte los datos de contacto del cliente y la Ilustración 26 visualiza los datos del cliente y la opción para establecer la operación como legítima o fraudulenta.

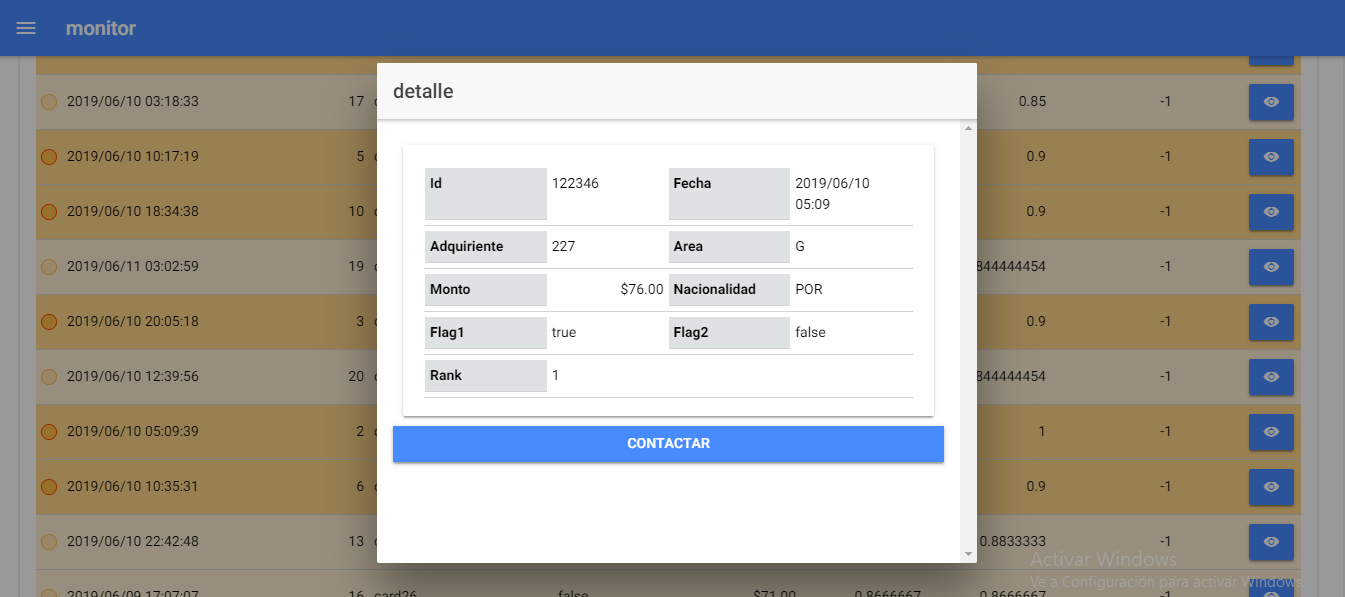


Ilustración 25. Captura de pantalla del detalle de una transacción sospechosa

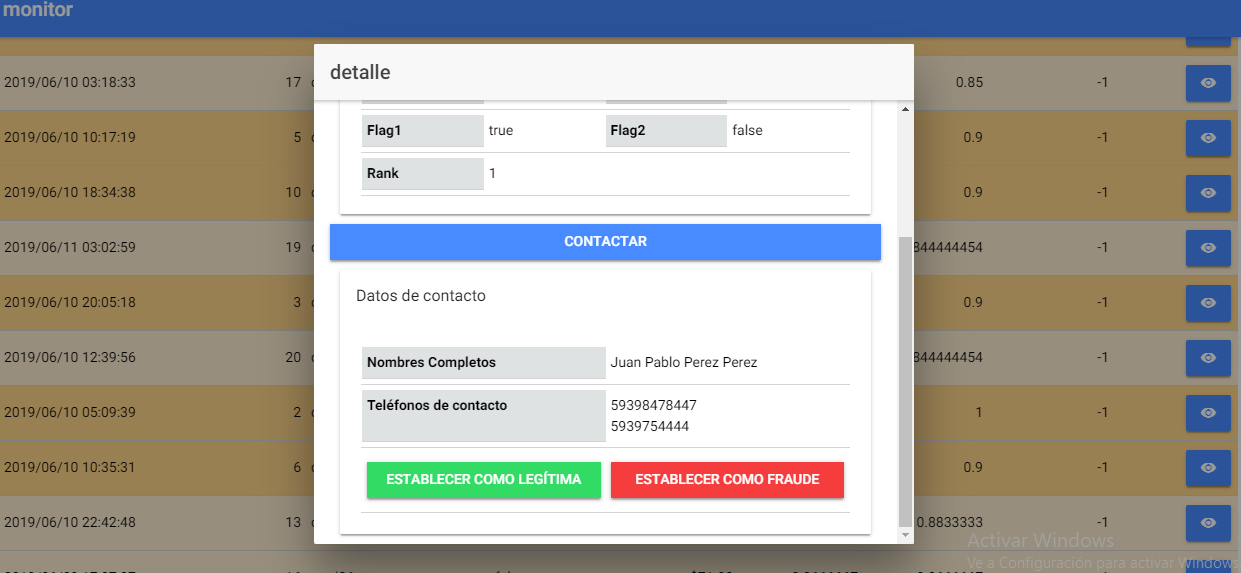


Ilustración 26. Captura de pantalla de los datos presentados para el contacto con el cliente

Finalmente, una vez que el usuario contactó al cliente, se procede a grabar en la transacción el feedback entregado por el cliente *(Ilustración 27).*

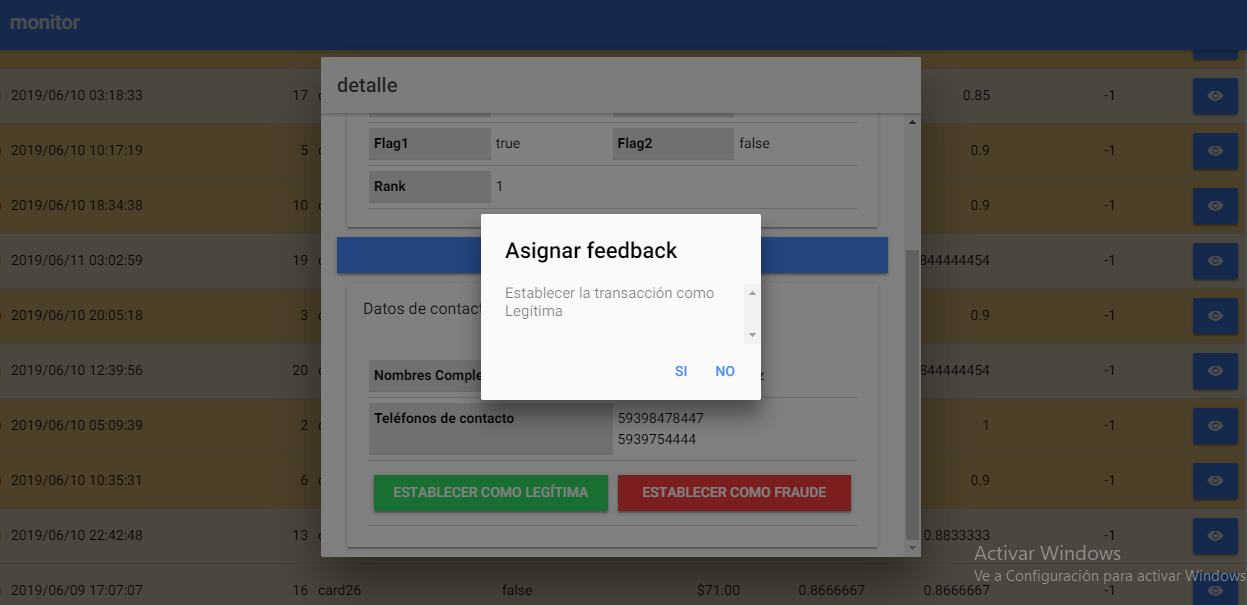


Ilustración 27. Captura de pantalla del mensaje de confirmación de una transacción confirmada como legítima

## Experimentación

Debido a la complejidad para conseguir transacciones financieras reales, para la implementación del prototipo de la solución planteada, se tomó como datos las transacciones de tarjetas de crédito en un período de 21 días con una muestra de 10000 transacciones.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Transacciones | Transacciones Fraudulentas | Transacciones Legales |
| Valor | 10000 | 409 | 99591 |
| Porcentaje | 100% | 0.41% | 99.59% |

Tabla 1. Muestra de ejemplo utilizado para elaboración y prueba del prototipo

Como se puede observar en la tabla la distribución de las clases es sumamente desbalanceada hacia la clase negativa (transacciones legales); apenas el 0.41% de las transacciones pertenecen son catalogadas como fraudulentas.

A continuación, un diagrama de la distribución de las transacciones de la muestra agrupado por día.

Ilustración 28. Distribución de la muestra agrupada por días

Ilustración 29. Distribución de transacciones fraudulentas en el conjunto de datos de muestra.

### Resultados

Los resultados presentados a continuación son obtenidos a partir del scoring generado para cada transacción que es obtenido a partir de la combinación de los scorings de los 2 clasificadores presentados en el Capítulo 3 *Ft  y Dt*

Para la obtención de los resultados se procedió a entrenar el modelo con las transacciones de los primeros 3 días, a partir del cuarto día el sistema empezó a generar la tabla de transacciones sospechosas que deben ser verificadas con el cliente por parte del grupo de investigadores que son los encargados de comunicarse con el dueño de la tarjeta para confirmar si la transacción es legítima completando así el proceso de feedback de las transacciones que serán utilizadas en el entrenamiento y refinamiento del modelo.

Luego de la ejecución del proceso para todas las transacciones del conjunto de datos se obtuvieron los siguientes resultados:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Transacciones sospechosas | Transacciones fraudulentas detectadas | Transacciones fraudulentas no detectadas | Transacciones Sospechosas Legítimas | Transacciones Legitimas no sospechosas |
| Valor | **640** | **264** | **22** | **376** | **80332** |
| Porcentaje | **100%** | **92.30%** | **7.70%** | **58.75%** | **99.21%** |

Ilustración 30. Distribución de transacciones fraudulentas detectadas como sospechosas

Ilustración 31. Distribución de transacciones fraudulentas no detectadas como sospechosas.

Ilustración 32. Resumen comparativo de fraudes detectados vs no detectados

Para calcular el rendimiento del modelo propuesto se procede a establecer la matriz de confusión; los resultados de las predicciones presentadas fueron calculadas en función a los scores de los 2 clasificadores planteados:

Para obtener un solo score de predicción del modelo completo, se calcula un promedio de las probabilidades de cada clasificador *Dt y Ft*. La fórmula utilizada para calcular este promedió se encuentra en la *Sección 4.3*. El valor resultante es un número decimal entre 0 y 1; a mayor valor, más alta es la probabilidad de que la transacción sea fraudulenta. Las 20 transacciones con mayor puntaje son enviadas y actualizadas en el monitor para que sean verificadas con el cliente

Para el cálculo de los resultados en la matriz de confusión se tomaron las transacciones reportadas en el monitor como transacciones que se predijeron como fraudulentas.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Clase Real | |
| Fraude (+) | Legítimo (-) |
| Predicción | Fraude (+) | TP | FP |
| Legítimo(-) | FN | TN |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Clase Real | |
| Fraude (+) | Legítimo (-) |
| Predicción | Fraude (+) | 264 | 376 |
| Legítimo(-) | 22 | 80332 |

Ilustración 33. Matriz de confusión para el conjunto de datos de muestra

Donde:

TP: son los casos que pertenecen a la clase y el clasificador los definió en esa clase.

FN: son los casos que si pertenecen a la clase y el clasificador no los definió en esa clase.

FP: son los casos que no pertenecen a la clase, pero el clasificador los definió en esa clase.

TN: son los casos que no pertenecen a la clase y el clasificador definió que no pertenecen a esa clase.

Exactitud (Accuracy): es la proporción del número total de predicciones que son correctas, se determina utilizando la siguiente ecuación:

La medida de exactitud no funciona bien en conjuntos de datos de datos desequilibrados ya que, si se tiene una muestra con el 95% de clase negativas y 5% de clase positiva, al clasificar todas las ocurrencias como negativas, se tendría un valor de exactitud de 0.95 o 95% que aparenta ser una buena medida.

Para resolver éste problema se considera la medida de Exactitud Equilibrada (Balanced Accuracy) (Mower, 2005). Balanced Accuracy normaliza las predicciones verdaderas positivas y las negativas verdaderas por el número de muestras positivas y negativas respectivamente y las divide para 2. La métrica de Balanced Accuracy se calcula mediante la siguiente formula:

Con los valores obtenidos de la Matriz de confusión se procedió a obtener las siguientes métricas:

|  |  |
| --- | --- |
| Métrica | Valor |
| Exactitud | 0.995086056 |
| Precisión | 0.4125 |
| TPR (Sensibilidad) | 0.923076923 |
| TNR (Especificidad) | 0.999726212 |

El valor de exactitud indica que el 99.5% de los registros fueron clasificados exitosamente, pero al ser un conjunto de datos desequilibrado se procede a obtener el valor de exactitud equilibrada.

= 0.9614015

Por lo tanto, se puede deducir que el nivel de exactitud de modelo planteado tiene una exactitud del 96.14%.

La medida de Precisión indica la proporción de identificaciones positivas fue correcta. Un modelo que no tiene falsos positivos tiene una precisión de 1.

El valor de precisión de 0.4125 nos indica que el 41% de registros clasificados como positivos, fueron realmente positivos; los mismos que al ser presentados en el sistema monitor pueden ser validados con el cliente.

La Sensibilidad es una métrica que mide la habilidad de detectar correctamente una transacción fraudulenta entre el total de transacciones fraudulentas.

Por lo tanto, el valor de 0.9230 indica que el modelo tiene una alta sensibilidad (92.3%) para detectar los verdaderos casos de fraude.

La Especificidad o Tasa de Verdaderos Negativos, en cambio mide la capacidad del modelo para marcar como legítimas aquellas transacciones que efectivamente no son fraudulentas.

Con un valor de 0.999 indica que el modelo planteado tiene un porcentaje de más del 99% de verdaderos negativos.

# CAPÍTULO 5

# CONCLUSIONES Y RECOMENDACIONES



## Conclusiones y Recomendaciones

La detección de eventos fraudulentos es una tarea particularmente desafiante y compleja, al ser eventos raros y al estar en constante evolución son difíciles de modelar. El aumento del volumen de transacciones que ocurren todos los días exige el uso de herramientas automáticas para apoyar la detección y los recursos humanos destinados a apoyar ésta tarea deben estar enfocados a investigar los casos más sospechosos.

Una solución estándar para tratar los problemas de clasificación que tienen una distribución de clases desequilibrada (como la detección de fraude) es reequilibrar las clases antes de entrenar un modelo. Una técnica de reequilibrio popular en los sistemas de aprendizaje automático es el sub-muestreo (undersampling).

Los sistemas tradicionales de detección de fraudes con tarjetas de crédito se han centrado tradicionalmente en buscar características como el monto de la transacción, el punto de venta, la ubicación, etc. Pero a partir de estas variables básicas es posible calcular nuevas características agregadas para mejorar el modelo del comportamiento del titular de la tarjeta

Generalmente los sistemas tradicionales utilizan una pequeña muestra de transacciones históricas de cada tarjeta para crear variables a nivel de cuenta. Debido a que es computacionalmente exigente calcular los agregados, estas características generalmente son calculadas fuera de línea y luego se agregan al vector de características que representa la transacción cuando está autorizada. La introducción de tecnologías de Big Data permite superar estos problemas, es decir, calcular los agregados en tiempo real y utilizar un conjunto más amplio de transacciones históricas.

La implementación de la solución con Kafka, Spark y Cassandra puede proporcionar escalabilidad fácil y tolerancia a fallas, para recibir, agregar y clasificar transacciones a una alta tasa de procesamiento.

Se debe tener precaución sobre la madurez de las soluciones de Big Data para implementaciones a gran escala. Las soluciones de Big Data están respaldadas por una creciente comunidad de código abierto que conduce a una evolución muy rápida y, al mismo tiempo, a una alta tasa de nuevos lanzamientos. Si, por un lado, esto garantiza una depuración rápida, por otro lado, puede provocar inestabilidad en las soluciones existentes, especialmente cuando se intenta combinar varias funcionalidades de diferentes herramientas en la misma plataforma.

La adopción de una solución de Big Data introduce una serie de parámetros que tienen un impacto en los resultados de computación y clasificación resultantes. Para obtener una solución eficiente a un problema de detección específico, se debe tener una correcta administración de los parámetros tanto a nivel de software como de hardware.

## Limitaciones

A continuación, se presentan algunas limitaciones con las que el enfoque cuenta actualmente. En primer lugar, no se divulgará datos delicados de la red y las transacciones, cuidando así la integridad, identidad y privacidad de la información que se manejará para la realización de esta investigación. En segundo lugar, la generación del modelo se realizará en base a los datos obtenidos de las transacciones de tarjetas de crédito de una entidad financiera. Finalmente, la implementación del prototipo se desarrollará sobre una plataforma en Open Source; no obstante, se toman en cuenta herramientas actuales y que están siendo ampliamente utilizadas en el área de Big Data.

## Trabajos Futuros

A de esta tesis se plantean futuras líneas o trabajos de investigación permitan enriquecer y fortalecer el modelo planteado:

En primer lugar, se planifica aplicar todo el modelo propuesto en un entorno real y proponer mejoras sobre el mismo. En segundo lugar, es importante que los resultados del modelo planteado sean revisados por los investigadores expertos en fraude para obtener sus observaciones y mejorar las predicciones del modelo.

En tercer lugar, para la implementación del presente prototipo se creó un proceso batch que se encarga de cargar las transacciones a partir de un archivo de texto y enviarlos a Kafka para para simular el ingreso de una transacción. Al prototipo presentado se debe agregar una interfaz que se encargue de comunicar el sistema de adquiriencia y autorización de la entidad financiera con el sistema de mensajería de Apache Kafka.

Finalmente, apuntamos a ampliar el modelo para incluir nuevas herramientas para manejo de Big Data y Machine Learning.

# BIBLIOGRAFÍA

Jha, S., Guillen, M., & Westland, C. (2012). Employing transaction aggregation strategy to detect credit card fraud. En *Expert Systems with Applications* (págs. 12650-12657). Elsevier.

Aite. (2014). *Financial Institutions, Merchants, and the Race Against Cyberthreats.*

Anderson, R. (2007). The Credit Scoring Toolkit: Theory and Practice for Retail Credit Risk Management and Decision Automation. *Oxford University Press*.

*Aprendizaje por refuerzo: algoritmo Q Learning* . (03 de 2019). Obtenido de http://www.cs.us.es/~fsancho/?e=109

Arcila Calderón, C., Barbosa Caro, E., & Cabezuelo Lorenzo, F. (2016). *Técnicas de Big Data: Análisis de textos a gran escala para la investigación científica y periodística.* Obtenido de http://www.elprofesionaldelainformacion.com/contenidos/2016/jul/12.pdf

Barnett, V., & Lewis, T. (1984). Outliers in statistical data. Wiley Series in Probability and Mathematical Statistics. *Applied Probability and Statistics, Chichester*.

Bayardo, J. (1998). *Efficiently mining long patterns from databases.* Obtenido de Computing Science.

Bejerano, P. G. (08 de 02 de 2017). *Diferencias entre machine learning y deep learning*. Obtenido de https://blogthinkbig.com/diferencias-entre-machine-learning-y-deep-learning

Bhatla, T. P., Prabhu, V., & Amit, D. (2003). Understanding Credit Card Frauds. *Cards Business Review*.

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*.

Bisong, E. (2019). *Building Machine Learning and Deep Learning Models on Google Cloud Platform.* OTTAWA, ON, Canada: Apress.

Bourel, M. (2012). *Métodos de agregación de modelos y aplicaciones.*

Brause, R., Langsdorf, T., & Hepp, M. (1999). *Neural data mining for credit card fraud detection. In Tools with Artiﬁcial Intelligence, Proceeding.* IEEE.

Breiman, L. (2001). *Universidad de California.* Obtenido de Random Forest: https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

Calle, J. P. (30 de 05 de 2018). *IA y machine learning, soluciones para enfrentar el fraude financiero*. Obtenido de https://www.riesgoscero.com/blog/ia-y-machine-learning-soluciones-para-enfrentar-el-fraude-financiero

Carreño Lopez, A. (2017). *Detección de sucesos raros con Machine Learning.* Madrid: Universidad Politécnica de Madrid.

Chawla, N., Lazarevic, A., Hall, L., & Bowyer, K. (2003). Smoteboost: Improving prediction of the minority class in boosting. *Knowledge Discovery in Databases: PKDD 2003*.

Cibersource. (2017). *Reporte de fraude online para América Latina 2017.* Visa Merchant Sales & Solutions.

Contreras, F. (2016). *Machine Learning.* Obtenido de Zemsania Global Group: https://zemsaniaglobalgroup.com/recursos-zemsania/whitepapers/DTS/Machine\_learning.pdf

Cordero Perez, C. (09 de 2014). *Virus para celulares se expanden más rápido que para computadoras.* Obtenido de https://www.nacion.com/tecnologia/moviles/virus-para-celulares-se-expanden-mas-rapido-que-para-computadoras/REPTKMGOD5HCJNX3UZQPHOHANI/story/

Correa Bahnsen, A., Aouada, D., & Ottersten, B. (2015). Exampledependent cost-sensitive decision trees. *Expert Systems with Applications*.

Correa Bahnsen, A., Stojanovic, A., Aouada, D., & Ottersten, B. (2013). Cost sensitive credit card fraud detection using bayes minimum risk. *Machine Learning and Applications (ICMLA). 2013 12th International Conference*.

Dal Pozzolo, A. (2015). *Adaptive Machine Learning for Credit Card Fraud Detection.*

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2015). Credit card fraud detection and concept-drift adaptation with delayed supervised information. *International Joint Conference on Neural Networks (IJCNN), IEEE*.

De la Fuente, S. (2011). *Regresión Logística.* Facultad de Ciencias Económicas y Empresariales. Universidad Autónoma de Madrid.

Donaldson, S. E., Williams, C. K., & Siegel, S. (2019). *Undertanding Security Issues.*

Dzomira, S. (2014). ELECTRONIC FRAUD (CYBER FRAUD) RISK IN THE BANKING INDUSTRY, ZIMBABWE. *Risk Governance & Control: Financial Markets & Institutions*.

*Estudios: Latinoamérica, la región de más rápido crecimiento en M-Commerce*. (03 de 2018). Obtenido de América Retail: https://www.america-retail.com/estudios-consumidores/estudios-latinoamerica-la-region-de-mas-rapido-crecimiento-en-m-commerce/

Fan, G., & Zhu, M. (2011). *Detection of rare items with target. Statistics and Its Interface.*

Fan, W., Prodromidis, A., Stolfo, S., & Chan, P. (1999). Distributed data mining in credit card fraud detection. *Intelligent Systems and their Applications*.

Fan, W., Stolfo, S., Zhang, J., & Chan, P. (1999). Adacost: misclassiﬁcation cost-sensitive boosting. *ICML*.

Frost, & Sullivan. (2010). *Retas clave contra el Fraude electrónico en las Instituciones Bancarias y Financieras de Latinoamérica.*

Fundación Telefónica. (2016). *La Sociedad de la Información en España 2015.* Barcelona: Editorial Ariel, S.A.

Galeano Cruz, L., & Domínguez Rivera, D. A. (2017). *Prototipo de Laboratorio Hadoop para Análisis Big Data en la Institución Universitaria Politécnico Grancolombiano.* Bogotá.

Grable, J. E., & Lyons, A. C. (2018). An Introduction to Big Data. *ECONOMICS & INVESTMENT MANAGEMENT*.

Group, A.-P. W. (2019). *Phishing Activity Trends Report - 2nd Quarter 2019.*

Grubbs, F. (1969). Procedures for detecting outlying observations in samples. *Technometrics*.

Guo, H., & Viktor, H. (2004). Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *ACM SIGKDD Explorations Newsletter*.

Hartigan, J., & M, W. (1979). Algorithm AS136: A k-means clustering algorithm. *Applied Statistics*.

Hernández, S. M. (2015). *Near real time fraud detection with Apache Spark.* Barcelona.

James, C. (08 de 2019). *Know Your Client (KYC)*. Obtenido de https://www.investopedia.com/terms/k/knowyourclient.asp

Juszczak, P., Adams, N., Hand, D., Whitrow, C., & Weston, D. (2008). Off-the-peg and bespoke classifiers for fraud detection. En *Computational Statistics & Data Analysis - Volume 52* (págs. 4521-4532). Elsevier.

Koshrow-Pour, M. (s.f.). *Encyclopedia of Criminal Activities and the Deep Web.* IGI Global.

Ladrero, I. (04 de 2017). *10 ejemplos de usos reales de Big Data Analytics*. Obtenido de https://www.baoss.es/10-ejemplos-usos-reales-big-data/

Mahmoudi, N., & Duman, E. (2015). Detecting credit card fraud by modiﬁed ﬁsher discriminant analysis. *Expert Systems with Applications*, 2510–2516.

Masoud, E. Y. (2013). *The Effect of Perceived Risk on Online Shopping in Jordan .* European Journal of Business and Management .

Mendez, J., Fdez-Riverola, F., Díaz, F., & Corchado, J. (2007). Sistemas inteligentes para la detección y filtrado de correo spam: una revisión. *Inteligencia Artificial. Revista Iberoamericana*.

Merino, M. (01 de 2019). *Conceptos de inteligencia artificial: qué es el aprendizaje por refuerzo* . Obtenido de https://www.xataka.com/inteligencia-artificial/conceptos-inteligencia-artificial-que-aprendizaje-refuerzo

Mitchell, T. (1997). *Machine Learning.* McGraw-Hill.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning . Second edition.* Cambridge, MA : The MIT Press.

Mora, L. (05 de 2016). *Qué es Big Data: fases y elementos*. Obtenido de https://www.ve.com/es/blog/que-es-big-data-fases-elementos

Morales, L. (2018). *BIGDATA: 5 MÉTODOS PARA DETECTAR POSIBLES FRAUDES*. Obtenido de https://www.grupo-novatech.com/bigdata-5-metodos-para-detectar-posibles-fraudes/

Mower, J. (2005). PREP-Mt: predictive RNA editor for plant mitochondrial genes. *BMC Bioinformatics*.

Murphy, K. (2012). *Machine learning. A probabilistic perspective.* London: The MIT Press.

Mysore, D., Khupat, S., & Jain, S. (09 de 2013). *Introducción a la clasificación y la arquitectura de big data*. Obtenido de Arquitectura y Patrones de Big Data, Parte 1: https://www.ibm.com/developerworks/ssa/library/bd-archpatterns1/index.html

Nexis, L. (s.f.). *True cost of fraud 2014 study*. Obtenido de http://www.lexisnexis.com/ risk/insights/true-cost-fraud.aspx

Pathak, N., Pawar, A., & Patil, B. (2015). A Survey on Keylogger: A malicious Attack. *International Journal of Advanced Research in Computer Engineering & Technology*.

Porter Felt, A., & Wagner, D. (s.f.). *Phishing on Mobile Devices.* Obtenido de https://pdfs.semanticscholar.org/94b2/44c518f431f84d2e00317709c98771a91eca.pdf

Riahi, Y., & Riahi, S. (2018). *Big Data and Big Data Analytics: Concepts, Types and Technologies.*

Rodríguez Arbonès , D. (s.f.). *Machine Learning for Fraud Detection.* Univerisidad de Copenhagen.

Sahin, Y., Bulkan, S., & Duman, E. (2013). A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications*.

Security Standards Counsil. (2018). *Requisitos y procedimientos de evaluación de seguridad. Version 3.2.1.* Obtenido de https://es.pcisecuritystandards.org/minisite/env2/

Segal, M. R. (2004). *Machine Learning Benchmarks and Random Forest Regression.* Center for Bioinformatics & Molecular Biostatistics.

Seguridad en Sistemas y Técnicas de Hacking. (2011). *¿Qué es el Hacking?* Obtenido de https://thehackerway.com/about/

Shawe-Taylor, G. (1999). Optimizing classiﬁers for imbalanced training sets. *Advances in Neural Information Processing Systems 11*.

Sibson, R. (1973). SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal (British Computer Society)*.

*Social Network Analysis for Fraud Detection*. (2013). Obtenido de https://www.kdnuggets.com/2013/11/lecture-social-network-analysis-for-fraud-detection.html

Sutton, R., & Barto, A. (2018). *Reinforcement Learning: An Introduction.* MIT Press, Cambridge.

Tasoulis, D., Adams, N., & Hand, D. J. (2006). Unsupervised clustering in streaming data. *ICDM Workshops*.

Téllez, J. (2004). *Derecho Informático.* México: McGraw-Hill, 3ra edición.

Ting, K. (2000). An empirical study of metacost using boosting algorithms. *Machine Learning: ECML 2000*.

*Unsupervised Machine Learning: What is, Algorithms, Example*. (s.f.). Obtenido de Guru99: https://www.guru99.com/unsupervised-machine-learning.html

Viola, P., & Jones, M. (2001). Fast and robust classiﬁcation using asymmetric adaboost and a detector cascade. *Advances in Neural Information Processing System*.

Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M., & Baesens, B. (2015). Apate: A novel approach for automated credit card transaction fraud detection using network-based extensions. *Decision Support Systems*.

Walpole, R. E., Raymond, H., & Myers, S. (s.f.). *Probabilidad y Estadística para Ingenieros.* México: Prentice-Hall Hispanoamericana, S.A.

Whitrow, C., Hand, D., Juszczak, P., Weston, D., & Adams, M. (2009). Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*.

Zareapoor, M., & Shamsolmoali, P. (2015). Application of Credit Card Fraud Detection: Based on Bangging Ensemble Classifier. *Procedia Computer Science*, 679-685.

1. Financial Institutions, Merchants, and the Race Against Cyberthreats [↑](#footnote-ref-1)
2. Definición realizada por F.E. Grubbs (Grubbs, 1969) [↑](#footnote-ref-2)
3. Definición realizada por V. Barnett and T. Lewis (Barnett & Lewis, 1984) [↑](#footnote-ref-3)
4. http://kafka.apache.org/ [↑](#footnote-ref-4)
5. https://spark.apache.org/streaming/ [↑](#footnote-ref-5)
6. http://cassandra.apache.org/ [↑](#footnote-ref-6)