Departamento – Centro de Ciencias Básicas de Sistemas de Información

Trabajo Práctico que presenta **Luis Enrique Ostos Ríos** para optar por el grado de
Maestría en Informática y Tecnologías Computacionales

Análisis de grandes cantidades de datos por medio de técnicas de máquinas de aprendizaje para la Ciberseguridad

Comité tutorial

Luis Eduardo Bautista Villalpando, Ph.D. - Tutor

Dr. Juan Muñoz López - Co-Tutor

MC. Edgar Oswaldo Díaz – Asesor

Aguascalientes, Ags, a 19 de junio de 2020

Autorizaciones

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

CARTA DE VOTO APROBATORIO INDIVIDUAL

M. en C. Jorge Martín Alférez Chávez
DECANO (A) DEL CENTRO **DE CIENCIAS BÁSICAS**

PRESENTE

Por medio del presente como ***TUTOR*** designado de la estudiante ***LUIS ENRIQUE OSTOS RÍOS*** con ID *243346* quien realizó *trabajo práctico* titulado: ***ANÁLISIS DE GRANDES CANTIDADES DE DATOS POR MEDIO DE TÉCNICAS DE MÁQUINAS DE APRENDIZAJE PARA LA CIBERSEGURIDADR***, un trabajo propio, innovador, relevante e inédito y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que *el* pueda proceder a imprimirlo así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"
Aguascalientes, Ags., a *18* de *junio* de *2020*.

Luis Eduardo Bautista Villalpando, Ph.D.
Tutor de *tesis*

c.c.p.- Interesado
c.c.p.- Secretaría Técnica del Programa de Posgrado

Elaborado por: Depto. Apoyo al Posgrado.
Revisado por: Depto. Control Escolar/Depto. Gestión de Calidad.
Aprobado por: Depto. Control Escolar/ Depto. Apoyo al Posgrado.

Código: DO-SEE-FO-07
Actualización: 01
Emisión: 17/05/19

M. en C. Jorge Martín Alférez Chávez
DECANO (A) DEL CENTRO **DE CIENCIAS BÁSICAS**

_____
P R E S E N T E

Por medio del presente como ***TUTOR*** designado de la estudiante ***LUIS ENRIQUE OSTOS RÍOS*** con ID *243346* quien realizó *trabajo práctico* titulado: ***ANÁLISIS DE GRANDES CANTIDADES DE DATOS POR MEDIO DE TÉCNICAS DE MÁQUINAS DE APRENDIZAJE PARA LA CIBERSEGURIDADR***, un trabajo propio, innovador, relevante e inédito y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que *el* pueda proceder a imprimirlo así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

A T E N T A M E N T E
"Se Lumen Proferre"
Aguascalientes, Ags., a *18* de *junio* de *2020.*

Dr. Juan Muñoz López
Tutor de *tesis*

c.c.p.- Interesado
c.c.p.- Secretaría Técnica del Programa de Posgrado

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

M. en C. JORGE MARTÍN ALFÉREZ CHÁVEZ

DECANO DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE

Por medio del presente como Asesor designado del estudiante **LUIS ENRIQUE OSTOS RIOS** con ID 243346 quien realizo el trabajo de tesis titulado: **ANÁLISIS DE GRANDES CANTIDADES DE DATOS POR MEDIO DE TÉCNICAS DE MÁQUINAS DE APRENDIZAJE PARA LA CIBERSEGURIDAD**, y con fundamento en el Articulo 175, Apastado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que él puede proceder a imprimir, y así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

ATENTAMENTE

"Se Lumen Proferre"

Aguascalientes, Ags., a 23 de junio de 2020

MITC. Edgar Oswaldo Diaz

Asesor de trabajo práctico

UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES

DICTAMEN DE LIBERACION ACADEMICA PARA INICIAR LOS TRAMITES DEL EXAMEN DE GRADO

Fecha de dictaminación dd/mm/aa: 22/06/2020

NOMBRE: LUIS ENRIQUE OSTOS RÍOS **ID** 243346

PROGRAMA: MAESTRÍA EN INFORMÁTICA Y TECNOLOGÍAS COMPUTACIONALES **LGAC (del posgrado):** INGENIERÍA DE SISTEMAS DECISIONALES PARA MEJORAR PROCESOS ORGANIZACIONALES

TIPO DE TRABAJO: ( ) Tesis (X) Trabajo práctico

TITULO: ANÁLISIS DE GRANDES CANTIDADES DE DATOS POR MEDIO DE TÉCNICAS DE MÁQUINAS DE APRENDIZAJE PARA LA CIBERSEGURIDAD

IMPACTO SOCIAL (señalar el impacto logrado): DESARROLLO DE TECNICAS PARA LA CIBERSEGURIDAD EN ORGANIZACIONES

INDICAR SI/NO SEGÚN CORRESPONDA:

Elementos para la revisión académica del trabajo de tesis o trabajo práctico:

SI	El trabajo es congruente con las LGAC del programa de posgrado
SI	La problemática fue abordada desde un enfoque multidisciplinario
SI	Existe coherencia, continuidad y orden lógico del tema central con cada apartado
SI	Los resultados del trabajo dan respuesta a las preguntas de investigación o a la problemática que aborda
SI	Los resultados presentados en el trabajo son de gran relevancia científica, tecnologica o profesional según el área
SI	El trabajo demuestra más de una aportación original al conocimiento de su área
SI	Las aportaciones responden a los problemas prioritarios del país
SI	Generó transferecia del conocimiento o tecnológica
SI	Cumpe con la ética para la investigación (reporte de la herramienta antiplagio)

El egresado cumple con lo siguiente:

SI	Cumple con lo señalado por el Reglamento General de Docencia
SI	Cumple con los requisitos señalados en el plan de estudios (créditos curriculares, optativos, actividades complementarias,estancia, predoctoral, etc)
SI	Cuenta con los votos aprobatorios del comité tutoral, en caso de los posgrados profesionales si tiene solo tutorpodrá liberar solo el tutor
SI	Cuenta con la carta de satisfacción del Usuario
SI	Coincide con el titulo y objetivo registrado
SI	Tiene congruencia con cuerpos académicos
SI	Tiene el CVU del Conacyt actualizado
NO	Tiene el artículo aceptado o publicado y cumple con los requisitos institucionales (en caso que proceda)

En caso de Tesis por artículos científicos publicados

	Aceptación o Publicación de los articulos según el nivel del programa
	El estudiante es el primer autor
	El autor de correspondencia es el Tutor del Núcleo Académico Básico
	En los artículos se ven reflejados los objetivos de la tesis, ya que son producto de este trabao de investigación.
	Los artículos integran los capítulos de la tesis y se presentan en el idioma en que fueron publicados
	La aceptación o publicación de los artículos en revistas indexadas de alto impacto

Con base a estos criterios, se autoriza se continúen con los trámites de titulación y programación del examen de grado

Sí X

No ____

FIRMAS

Elaboró:

* NOMBRE Y FIRMA DEL CONSEJERO SEGÚN LA LGAC DE ADSCRIPCION: DR. JOSÉ MANUEL MORA TAVARES

NOMBRE Y FIRMA DEL SECRETARIO TÉCNICO: **MTRO. JORGE EDUARDO MACIAS LUEVANO**

*** En caso de conflicto de intereses, firmará un revisor miembro del NAB de la LGAC correspondiente distinto al tutor o miembro del comité tutoral, asignado por el Decano**

Revisó:

NOMBRE Y FIRMA DEL SECRETARIO DE INVESTIGACIÓN Y POSGRADO: DRA. HAYDEE MARTÍNEZ RUVALCABA

Autorizó:

NOMBRE Y FIRMA DEL DECANO: M. EN C. JORGE MARTÍN ALFÉREZ CHÁVEZ

Nota: procede el trámite para el Depto. de Apoyo al Posgrado

En cumplimiento con el Art. 105C del Reglamento General de Docencia que a la letra señala entre las funciones del Consejo Académico: .... Cuidar la eficiencia terminal del programa de posgrado y el Art. 105F las funciones del Secretario Técnico, llevar el seguimiento de los alumnos.

Agradecimientos

Deseo expresar mi agradecimiento a mis profesores que me apoyaron con su experiencia y conocimientos en esta investigación.

Al Dr. Bautista que me apoyo con los conocimientos de aprendizaje automático, los algoritmos que me ayudo a entender y el poder visualizar y entender la ciencia del aprendizaje automático.

Al profesor Oswaldo y al Dr. Muñoz por permitirme realizar un proyecto en un ámbito laboral para una organización de renombre.

A mi familia que siempre me han impulsado a proponerme nuevas metas y alcanzarlas y que siempre me apoyaron para la finalización de mis estudios. Mi madre que siempre está para mí y me hace creer que el cielo no es el límite. Y a mi hermano que siempre me muestra alegría y satisfacción por mis logros.

Índice General

Indice de tablas

Indice de figuras

I. Resumen

Después del capital humano, el activo principal de una organización es la información, con la que se genera valor y con la que se permite realizar sus actividades. La información se extrae, se recolecta e integra para poder ser almacenada donde es posible procesarla, y posteriormente analizarla para convertirse en un elemento valioso para la toma de decisiones. Esta información debe ser protegida contra robos, ataques, clonación, fraude y destrucción, por lo que se requiere implementar acciones, tanto preventivas como correctivas.

Ante un ataque es necesario reducir el tiempo empleado en validar diferentes hipótesis de seguridad tales como, el posible origen del ataque, que aplicaciones de servicios fueron los objetivos del ataque y los datos afectados, así como los incidentes en el que una persona o entidad no autorizada han accedido a la información, también conocidas como brechas de seguridad.

Así pues, se busca minimizar la necesidad de un soporte externo o lateral, es decir, soporte extra que trabaje a la par del ya existente en los sistemas. Además, se busca disminuir la latencia de observación en tiempo real y la toma de decisiones que presenta el Instituto Nacional de Estadística y Geografía (INEGI) tras la posible invasión, así como ampliar el mapa de revisión de amenazas cibernéticas conocidas y definidas por parte de la organización.

Todo esto con el fin de cubrir la necesidad de identificar las direcciones IPs y DNS maliciosos, los cuales son dominios de internet que realizan ataques a hacia la organización.

I. Resumen (Abstract)

After human capital, the main asset of an organization is the information, in which, value is generated and this allows to perform its activities. Information is extracted, gather and integrated in order to be stored where is possible, to be processed and then analyze, this in order it to be transformed in a valuable asset for decision making. This information must be protected against theft, attacks, cloning, fraud and destruction, because of this, it is required implement preventive and corrective actions.

In the case of an attack occurs, it is necessary to reduce the time used to validate different security hypothesis, such as the possible origin of the attack, the applications of the services target by the attack, and the affected data, as well as incidents in which an unauthorized person or entity has accessed to the information, also known as security breaches.

The above, in the search for minimize the need for external or lateral support, that means, an extra support that works alongside with the team that already exist in the system. In addition, it also seeks to decrease the latency in real-time to perform the observation and decision making of the Instituto Nacional de Estadística y Geografía (INEGI) after the invasion, as well as expanding the revision map in order to known and defined cyber threats in the organization.

All the above in order to cover the need to identify malicious IP addresses and DNS, which are internet domains that execute attacks in the organization.

II. Introducción

En las organizaciones, continuamente se presentan en la infraestructura ataques e invasiones que constituyen amenazas de seguridad, los cuales tienen origen tanto externo como interno. Dicha infraestructura está compuesta por el conjunto de software y hardware sobre el que se soportan y se comunican los servicios de la organización.

El Instituto Nacional de Estadística y Geografía (INEGI) requiere de un análisis de datos de servidores de aplicaciones, con el fin de evaluar si son atacados por usuarios externos o internos. Considerando un análisis previamente de las bitácoras de los servidores en un ambiente controlado que no afecte la información de la organización, para una evaluación del estado de seguridad de la infraestructura.

Para ello, se debe realizar la identificación de direcciones IPs y DNS maliciosas, en las que se detecten actividades que se estimen que están encaminadas a dañar o robar la información. Para realizar el análisis de los accesos a los servidores y la clasificación de las redes se ha propuesto el análisis de grandes volúmenes de datos mejor conocido en inglés como big data analytics.

Para realizar el análisis de los datos, que se han recolectado y tienen relación con la actividad de los servidores de aplicaciones, se ha utilizado una técnica de aprendizaje automático, el cual, a través de algoritmos que realizan tareas de entrenamiento y aprendizaje, permite llevar a cabo un proceso de análisis e interpretación de la información. Dicha técnica ha permitido generar un método para fortalecer la seguridad de la información relacionada con un ámbito tecnológico, conocido como Ciberseguridad.

El presente documento está estructurado de la siguiente manera: planteamiento, marco teórico, descripción de la solución propuesta, así como de las herramientas que fueron utilizadas, descripción de los experimentos que ayudaron a la validación del modelo y por último los resultados y conclusiones que se lograron obtener con la aplicación del modelo que se describe dentro de este documento.

Capítulo 1. Planteamiento de la problemática a atender a través del trabajo práctico

El Instituto Nacional de Estadística y Geografía (INEGI), se encarga de producir, integrar y dar a conocer la información estadística y geográfica de la población y la economía, la cual abarca todos los aspectos que caracterizan el territorio, para el Gobierno de México.

Actualmente el INEGI cuenta con un centro de datos, en el cual por medio de tecnologías web (recursos que utilizan los conceptos de red, sus conexiones y comunicaciones entre equipos conectados), recolecta, procesa y almacena la información estadística y geográfica, en servidores de aplicaciones.

En el área encargada de la administración de los servidores de aplicaciones con tecnología web (conocido como el Departamento de Administración de Servicios en Web dentro de la Coordinación General de Informática), existe actualmente un proceso para la revisión de las bitácoras de acceso a las aplicaciones publicadas en internet, con la finalidad de identificar segmentos de red que tienen eventualidades de tipo “códigos de errores con base a la norma w3c” (STANDARDS, 2019), por ejemplo, 300, 400 o 500, de tal forma los tiempos de respuesta para realizar un reporte que permita la toma de decisiones referentes a si un evento en particular es una posible vulnerabilidad orientada a la “denegación de servicios”, además de tener retrasos significativos, a lo cual es necesario identificar una forma de automatizar la lectura de las bitácoras en los servidores del centro de datos institucional, permitiendo que la recolección, y análisis de información sea de forma ágil. El caso práctico descrito a continuación, tiene la intención de implementar una estrategia para el análisis de la información, con algoritmos de aprendizaje automatizado.

Ante la posible vulnerabilidad definida en el presente documento, se busca cubrir los siguientes problemas específicos con la utilización de la técnica de aprendizaje automático que se describe en el documento:

1. Tiempo empleado en validar diferentes hipótesis (posible origen del ataque, cuál sería la aplicación y los datos afectado, etc.) a la respuesta de una o varias invasiones (brechas o accesos permitidos o no permitidos) en la organización.
2. Tiempo que ocupa el equipo de trabajo en analizar conjuntos de datos complejos.
3. Necesidad de un soporte externo o extra al ya existente en la organización.
4. Demora en el análisis en tiempo real y en la toma de decisiones de la organización tras el ataque.
5. Desconocimiento de las direcciones IPs y DNS maliciosos que representan una amenaza continua.
6. Accesos ocultos de usuarios.

1.1 Funcionamiento actual

El INEGI provee al usuario datos estadísticos y geográficos en cuanto al territorio nacional de México, por medio de censos, conteos y registros administrativos, entre otros indicadores, que permiten conocer las características de los estados y municipios que conforman nuestro país ayudando a la toma de decisiones. Los datos son almacenados, procesados en las bases de datos de la organización y difundidos en servidores de aplicaciones en internet, de tal forma un usuario puede realizar la consulta y descarga de la información pública en la dirección (www.inegi.org.mx).

Las solicitudes de tipo web browser (request/response), que emite un usuario, son recibidas por la infraestructura tecnológica implementada en la organización con puntos de control de seguridad informática. En este flujo, dentro de las bitácoras en los servidores de aplicaciones, son registrados los direccionamientos de IP´s (segmentos de red), internos y externos, que el webmaster (administrador de servidores web) ejecuta por medio de un proceso manual, para validar por medio de percepción los códigos de error con base a la norma W3c HTTP que arrojen 300, 400 o 500, con la finalidad de identificar comportamientos que indiquen una posible vulnerabilidad (STANDARDS, 2019).

La figura 1, describe el proceso actual de comunicación usuario y servidor al momento de solicitar información proveniente de los servicios de la organización.

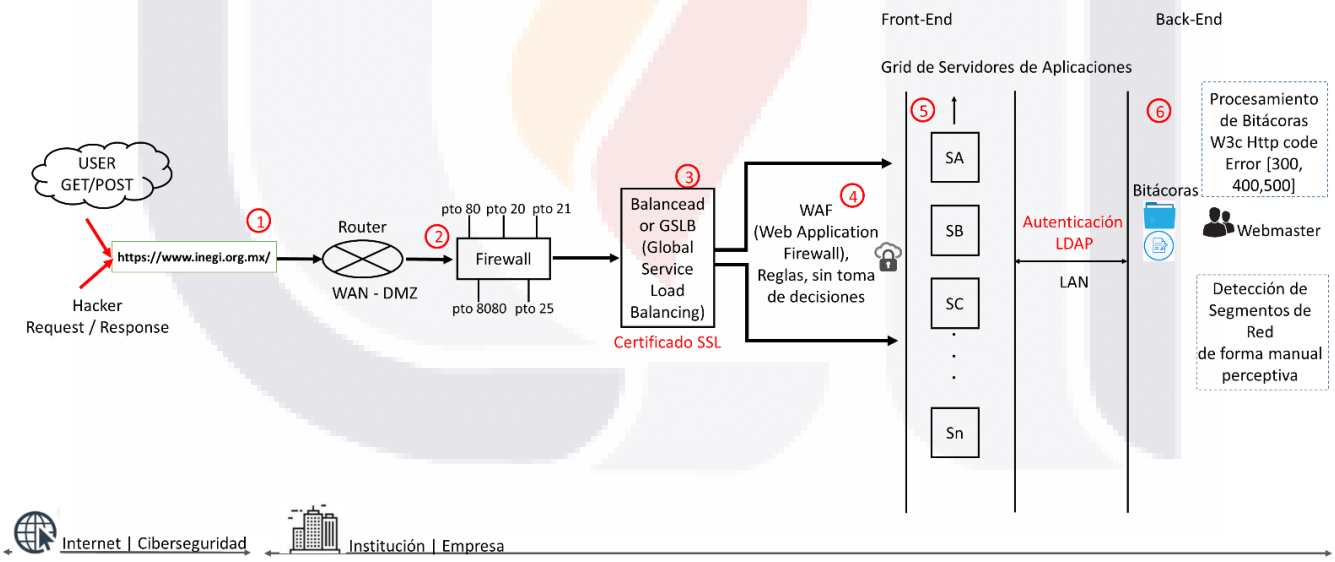

Figura 1. Flujo de información actual, User - Servidor

Las diferentes etapas que describen el flujo actual donde la solicitud de información fluye dentro de la organización es el siguiente:

1. Los usuarios (usuarios seguros y hackers) mediante métodos GET o POST, intentan acceder a la información de los servidores iniciando con la conexión al último punto entre internet y la distribución de información de la compañía (INEGI).

2. El acceso es filtrado por el firewall físico que limita sus accesos de acuerdo con los puertos que se encuentran habilitados.
3. Los accesos que logran ser aceptados son distribuidos por el balanceador definido por la organización, GSLB (Global Services Load Balancing). El cual puede ser operado en la Oficina Central o mediante un sitio alterno definido por su DRP (Disaster Recovery Plan).
4. Los accesos nuevamente son filtrados mediante sus protocolos definidos dentro de su WAF (Web Application Firewall) y así verificar que es una solicitud legal.
5. Se da la lectura de datos. En este punto se encuentra la decisión de quien puede entrar al flujo de los datos y acceder a los servidores listados en la organización. Cada uno de los datos de acceso son almacenados en los logs de los servidores.
6. Los accesos y actividades son almacenados en las bitácoras de los servidores de aplicaciones, aquí es donde son registrados los direccionamientos de IP´s (segmentos de red), internos y externos, que el webmaster revisa por medio de un proceso manual, para validar por medio de percepción los códigos de error con base a la norma W3c HTTP que arrojen 300, 400 o 500, con la finalidad de identificar comportamientos que indiquen una posible vulnerabilidad.

El Instituto Nacional de Estadística y Geografía (INEGI) con el fin de evaluar si la organización es atacada por entidades externas o internas, requiere de un análisis de la información de los servidores de los datos que se generan, se procesan y se distribuyen.

Capítulo 2. Objetivos

El objetivo principal de este trabajo de investigación es; *cubrir la posible vulnerabilidad en la revisión de las bitácoras en los servidores del centro de datos institucional, que actualmente efectúa el grupo de administración para las aplicaciones con tecnologías web pero de una forma* ***automatizada***.

2.1 Objetivos específicos

Los objetivos específicos que se cubren son:

1. Identificar segmentos de red nacional y locales maliciosas (los cuales son dominios de internet que buscan realizar ataques a la organización dañando o robando la información).
2. Implementar una estrategia con base a técnicas de aprendizaje automático (Machine Learning), para poder realizar un entrenamiento de modelo de análisis, con el fin de identificar patrones de comportamiento en los accesos, para que la organización posea un método para fortalecer sus esquemas de seguridad en proyectos con tecnología web, haciéndolos capaces de anticipar problemas en los accesos, mediante una correcta toma de decisiones.

Capítulo 3. Fundamentación teórica

3.1 Ciberseguridad

La seguridad informática es el conjunto de tecnologías y procesos diseñados para proteger computadoras, redes, programas y datos de ataques, accesos no autorizados, cambios o destrucción. Los sistemas de ciberseguridad están compuestos por sistemas seguridad de red y en el equipo (host) (Buczak, 2016).

Cada sistema de ciberseguridad se debe componer al menos de los siguientes elementos:

- ✓ Una configuración mínima de firewall, el cual es un dispositivo de seguridad de la red que monitorea el tráfico de red y decide si permite o bloquea el tráfico específico en función de un conjunto definido de reglas de seguridad.
- ✓ De configuración mínima de antivirus y un sistema de detección de intrusos (IDS por sus siglas en inglés) (Buczak, 2016).

Ya sea que formen parte de una empresa o trabajen para una organización gubernamental, los investigadores de seguridad en cómputo de la actualidad son especialistas en seguridad y analistas de inteligencia, los cuales se enfrentan a una presión extraordinaria para responder rápidamente a una amplia gama de amenazas, desde ataques cibernéticos, amenazas terroristas, criminalidad y ocupaciones no éticas. Estos ataques son especialmente relevantes, ya que las sociedades están aumentando su dependencia del ciberespacio para el crecimiento económico, el bienestar social, las operaciones gubernamentales y el monitoreo y gestión de la infraestructura crítica (IBM Corporation, 2017).

Existe una gran variedad de fuentes, tanto de amenazas cibernéticas como de métodos de ataque. Para contrarrestar estas fuentes, las organizaciones necesitan algún tipo de “inteligencia” que ayude a fortalecer la prevención de amenazas internas y externas.

Actualmente, las organizaciones se enfrentan a cinco fuentes críticas de peligro cibernético, estas son:

1. Ciberespionaje comercial industrial: robo de secretos comerciales e industriales por parte de una empresa o incluso un país extranjero como en el caso suscitado en 2014 donde el gobierno de Estados Unidos presentó cargos contra cinco miembros del ejército de China por robar secretos comerciales a empresas estadounidenses (Schmidt & Michael, 2014).
2. Gobierno de espionaje cibernético: las instancias gubernamentales pueden ser objeto de ataques que afecten la infraestructura crítica energética, de salud o transporte, como fue en el caso del ataque cibernético de denegación de servicio (DDoS por sus siglas en inglés) que sufrió el sitio de la Administración Pública Federal del gobierno de México: www.gob.mx durante 45 minutos, en junio de 2016, donde también la infraestructura del sitio quedó vulnerable durante ese periodo de tiempo. Esto significó un nulo acceso de los ciudadanos a la única ventanilla de comunicación digital con el gobierno federal (IBM Corporation, 2017).

3. Crimen organizado: Se han presentado ataques a instituciones bancarias y organizaciones por parte de grupos del crimen organizado, tal fue el caso que se presentó en Europa en el 2016, donde un grupo criminal internacional atacó con virus informáticos los cajeros de entidades bancarias lo cual provoco pérdidas financieras en el conjunto de la Unión Europea, entre los más afectados fue España, Portugal y Reino Unido (IBM Corporation, 2017).
4. Actividad terrorista: diversos grupos terroristas, como es el caso del grupo islámico al-Qaeda, se han aprovechado de la utilidad de las herramientas en el ciberespacio para alimentar las diversas dimensiones de su activismo, mediante el enaltecimiento, la radicalización, la captación, el reclutamiento, la canalización de personas hacia sus zonas de combate y el diseño y ejecución de sus acciones terroristas (IBM Corporation, 2017).
5. Hacktivismo de los activos informáticos: esta actividad es la que se ha presentado para irrumpir en sistemas informáticos debido a un propósito político y/o social, esto con el fin de captar la atención de la sociedad y así ganar visibilidad para una causa en particular. Los objetivos por lo regular son agencias del gobierno, corporaciones multinacionales o cualquier otra organización la cual posea una ideología diferente a estos grupos de personas. Dos de los grupos más conocidos en esta práctica son Anonymous y Lulz Security. Un ejemplo de esos ataques ocurrió el 16 de enero del 2013, donde la página de la SEDENA, fue hackeada por el grupo Anonymous. El grupo Anonymous modifico la página de inicio y coloco un video en donde se observaban imágenes de la toma de protesta del presidente Enrique Peña Nieto (IBM Corporation, 2017).

3.2 Big data

El término Big Data es usado para describir grandes cantidades de datos en el orden de gigabytes y terabytes; los cuales se pueden originar a partir de sensores del Internet de las cosas (IoT por sus siglas en ingles), redes sociales, streaming, registros de transacciones, etc. En un mundo impulsado por la constante generación de información a través de los diferentes tipos de dispositivos tecnológicos que se encuentran disponibles actualmente (Goodfellow et al., 2018).

La tecnología de Big Data nace a partir de la incapacidad de las arquitecturas de datos tradicionales de procesar de manera eficiente, los exorbitantes volúmenes de nuevos conjuntos de datos generados diariamente (Chang, 2015). Las características de Big Data que fuerzan a nuevas arquitecturas relacionados con este paradigma tecnológico son:

- ✓ Volumen (el tamaño del conjunto de datos);
- ✓ Variedad (datos de múltiples repositorios, dominios o tipos);
- ✓ Velocidad (tasa de flujo a partir de cual se generan);
- ✓ Variabilidad (el cambio en otras características) (Gartner, 2015).

Cada una de las características anteriores da como resultado diferentes arquitecturas o diferentes procesos del ciclo de vida de los datos para lograr la eficiencia necesaria en el procesamiento de grandes volúmenes de datos (ISO/IEC, 2015).

3.3 Aprendizaje automático

El aprendizaje automático o Machine learning (por su traducción en inglés), se describe como una variedad de paradigmas de aprendizaje, algoritmos, resultados teóricos y aplicaciones. Este es un campo multidisciplinario que utiliza técnicas, métodos y herramientas de la inteligencia artificial, la probabilidad y estadística, la teoría de la complejidad computacional, teoría de control, teoría de la información, filosofía, psicología, neurobiología y otros campos (Mitchell, 1997).

Se entiende que una computadora puede “aprender” con base a la experiencia presentada en una tarea bajo un rendimiento determinado, todo esto con la idea que mejore con la experiencia (Mitchell, 1997).

El aprendizaje automático es una disciplina científica del ámbito de la Inteligencia Artificial (IA) que implanta métodos y modelos matemáticos en los sistemas para que estos aprendan automáticamente. Por aprender se refiere a identificar tipos de patrones complejos en millones de datos de forma más concreta. El aprendizaje automático trata de crear aplicaciones capaces de generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos (Marsland, 2009).

El aprendizaje automático consiste en hacer que la computadora modifique o adapte sus acciones para mejorar su precisión, a través de la medición por la forma en que las acciones elegidas reflejen decisiones correctas. Algo importante a considerar en el momento de hablar del aprendizaje automático, es mencionar la complejidad computacional de los métodos utilizados al producir algoritmos. La complejidad comúnmente se divide en dos partes: complejidad del entrenamiento y complejidad del algoritmo utilizado por la aplicación con el que es sometido al entrenamiento (Marsland, 2009).

La teoría e implementación de un método de aprendizaje automático comprende el entrenamiento del modelo (una función paramétrica que describe el mapeo de entradas a salidas), validaciones de entradas (verificación de las propiedades de los valores ingresados al modelo) y cambios en la arquitectura con el fin de defender contra ataques (dado a que también es posible defenderse de ataques modificando la estructura del sistema de aprendizaje) (Goodfellow et al., 2018).

A menudo se realizan los siguientes pasos durante el proceso de aprendizaje automático:

1) Identificar los atributos de la clase (características) y clases de los datos de entrenamiento.

2) Identificar un subconjunto de los atributos necesarios para la clasificación.

3) Aprender el modelo usando datos de entrenamiento.

4) Y finalmente usar el modelo entrenado para clasificar los datos desconocidos (Buczak, 2016).

Actualmente, los aprendizajes automáticos son utilizados por muchas aplicaciones de gran importancia como núcleo de su objetivo de operación. Los algoritmos de búsqueda,

los algoritmos automatizados del comercio financiero, análisis de datos, vehículos autónomos y detección de malware dependen críticamente de los algoritmos de aprendizaje automático, que interpretan sus respectivas entradas de dominio para proporcionar salidas inteligentes, que facilitan el proceso de toma de decisiones de los usuarios o de los sistemas automatizados. Debido a que los aprendizajes automáticos son cada vez más utilizados en contextos donde los adversarios maliciosos tienen un incentivo para interferir con el funcionamiento de un sistema que hacen uso de este, es cada vez más importante proporcionar protecciones, o "garantías de solidez", contra la manipulación adversaria (Gartner, 2015).

Las técnicas de aprendizaje automático se han aplicado con éxito a varios problemas del mundo real en áreas tan diversas como el análisis de imágenes Web, semántica, bioinformática, procesamiento de textos, procesamiento de lenguaje natural, telecomunicaciones, finanzas, diagnóstico médico, etc. (Gama, 2005).

Un área en la que el aprendizaje automático desempeña un papel clave es la de extracción de datos, esto debido al amplio uso para el desarrollo de modelos de asociación, predicción de agrupamiento, diagnóstico y regresión. El objetivo principal del aprendizaje automático es construir un modelo computacional a partir de la experiencia pasada de lo que se ha observado. Para ello se estudia desde la adquisición automatizada de conocimientos de dominio en búsqueda de la mejora del rendimiento de los sistemas como resultado de la experiencia (Gama, 2005).

El enfoque del aprendizaje automático generalmente consta de dos fases: 1) entrenamiento y 2) pruebas.

En la fase de entrenamiento se identifican sus características con las que el modelo podrá clasificar la información desconocida. En el caso de una minuciosa detección, es en la fase de entrenamiento donde se aprende a usar ejemplos apropiados para el conjunto de datos de entrenamiento. En el caso de una detección de anomalía, el tráfico normal de patrones es definido en esta fase (Manu, 2018).

En caso de que en la fase de entrenamiento se detecte un uso indebido, este se aprenderá mediante el uso de los modelos que conforman el entrenamiento. En la fase de prueba, los nuevos datos se procesan en el modelo y el dato a analizar se clasifica según si pertenece a una de las clases de uso indebido (Manu, 2018).

Si el dato a analizar, no pertenece a ninguna de las clases de uso indebido, se clasifica como normal (Manu, 2018).

En los métodos supervisados, se utilizan un conjunto de ejemplos de entrenamiento con respuestas correctas (objetivos) y basándose en ese conjunto de entrenamiento, el algoritmo generaliza para responder correctamente a todas las entradas posibles. Esto también es llamado aprender de ejemplos (Manu, 2018).

En los métodos no supervisados, no son proporcionadas respuestas correctas, lo que se busca es que el algoritmo intente identificar similitudes entre las entradas para que las entradas que tienen algo en común se clasifiquen juntas. El enfoque estadístico para el método no supervisado se conoce como estimación de densidad. Los métodos supervisados y no supervisados de entrenamientos de aprendizaje automático y manejo de datos de

prueba en Big Data son; K-Means Clustering, Hierarchical Clustering y Probabilistic Clustering, los cuales se utilizan para procesar esta gran cantidad de datos por medio de diferentes clasificadores para obtener información útil y predecir patrones. En el dominio de la ciberseguridad es posible utilizar estos algoritmos y patrones para predecir posibles tendencias de ataques (Manu, 2018).

3.3.1 Algoritmo de aprendizaje automático de detección de anomalía, T-digest

La forma más común para detectar anomalías hoy en día, es por medio de una alerta de umbral, la cual es configurada manualmente para enviar una alerta a la presencia de posibles anomalías (Dunning, 2014).

La entrada a dicha alarma es una medida numérica de algún tipo. La idea básica es que cada vez que esta medición excede un umbral que se ha establecido, durante un cierto período de tiempo, se activa la alarma (Dunning, 2014).

Este enfoque funciona si el sistema que se observa posee un patrón de mediciones bien entendidas y el número de mediciones de diferentes tipos no es enorme. Pero este enfoque puede volverse difícil de llevar a cabo de manera efectiva, si se tiene una gran cantidad de mediciones con comportamientos que no son comprendidos. Como resultado, si se tiene una gran cantidad de mediciones que son impredecibles o indefinidas es difícil obtener resultados óptimos, los cuales se encuentra comúnmente en entornos de interés del mundo real. Esa es una razón por la que necesitamos nuevas formas de abordar la detección de anomalías (Dunning, 2014).

Para lo anterior, es posible establecerse el umbral de detección de anomalías muy bajo para detectar la mayoría o todas las anomalías, pero esto puede dar lugar a una alta tasa de falsos positivos. Sin embargo, lidiar con falsas alarmas también tiene un costo. Debe tener suficientes recursos para responder a las alarmas y determinar si son falsos positivos o no. Demasiadas falsas alarmas se convierten en una distracción, se desperdicia tiempo y potencialmente abruman al humano que necesita responder ante las alarmas. Por consecuencia, la persona encargada de las alarmas podría habituarse a ellas, lo que aumenta el peligro de que no se responda adecuadamente a una anomalía verdadera (Dunning, 2014).

Se debe elegir un umbral para controlar las alarmas totales en un período de tiempo determinado. Teóricamente, se debería poder establecer este umbral examinando la distribución de la medición en condiciones normales y seleccionando un valor del umbral para obtener la frecuencia deseada de alarmas (Dunning, 2014).

Esa acción puede parecer fácil, pero calcular incrementalmente un cuartil extremo con precisión con memoria limitada puede ser difícil, especialmente si se necesita hacer esto para una gran cantidad de situaciones relacionadas.

T-digest es un algoritmo utilizado para estimar cuartiles extremos en grandes conjuntos de datos en línea. Los percentiles son una escala muy natural para hablar sobre la configuración del umbral. Sin embargo, traducir un percentil en un umbral puede ser

complicado con memoria y tiempo limitados. Ahí es donde T-digest puede ayudar a definir dichos valores (Dunning, 2014).

El método T-digest es el que se utiliza en este caso de estudio, debido a que es un método rápido cuando se trata de analizar grandes volúmenes de datos en tiempo real. Este método ha sido probado con buenos resultados en ambientes que utilizan big data.

El algoritmo T-digest fue creado por Ted Dunning, con el fin de poder estimar con precisión cuartiles extremos para conjuntos de datos muy grandes con un uso limitado de memoria. El algoritmo se encuentra disponible en el repositorio abierto *https://github.com/tdunning/t-digest* y ha sido publicado por Maven Central (Dunning, 2014).

Una de las ventajas de T-digest es la precisión, especialmente para cuartiles extremos. En lugar de tener que clasificar una gran cantidad de muestras para estimar un cuartil de interés, un valor de entrada se puede analizar en línea usando un T-digest para encontrar el umbral correspondiente a cualquier cuartil (Dunning, 2014).

3.4 Visualización de datos

Es necesario hablar de la visualización de datos ya que el análisis de la información depende de lo que se puede ver. Al trabajar de un método de detección de anomalías es necesario poder visualizar los picos en un grupo de datos (Cairo, 2017).

La visualización de datos está basada en una idea simple, pero también poderosa: el cerebro humano no está bien preparado para manejar con soltura símbolos arbitrarios y abstractos, como los números. Somos capaces de interpretar el sentido de grandes cantidades de cifras sólo indirectamente; por ejemplo, cuando las representamos proporcionalmente por medio de la variación de ciertas propiedades de objetos visuales, como su altura, longitud, tamaño, ángulo, grosor o color (Cairo, 2017).

La visualización a partir de los datos pretende construir un conjunto gráfico, sintético o complementario, que destaque lo más significativo o los asuntos clave, que permitan entender, establecer agrupaciones, relaciones o tendencias estadísticas, que reduzcan al mínimo la entropía y facilite el obtener conclusiones o pruebas para su interpretación (Ochoa & Sancho, 2014).

Algunas técnicas y algoritmos de extracción de datos son difíciles de entender y utilizar para los tomadores de decisiones. La visualización puede hacer que los datos y los resultados de la extracción sean más accesibles, lo que permite la comparación y la verificación de los resultados. La visualización también se puede utilizar para dirigir algunos algoritmos de extracción de datos y poder tener información entendible para la toma de decisiones (Kantardzic, 2011).

Algo necesario para poder definir como visualizar e interpretar los datos es poder definir su tipo así como poder determinar su clasificación lo cual se presenta en la siguiente sección.

3.4.1 Tipos de datos

Existen diferentes tipos de datos de acuerdo con sus características. Entre ellos ubicamos los siguientes que a continuación se describen:

3.4.1.1 Datos numéricos

Los valores de tipo numérico, incluyen valores reales que pueden ser variables o enteros, tales como la edad, velocidad o longitud. Existen dos propiedades importantes para este tipo de valores:

1. Poseen un orden relacional, es decir los valores de este tipo tendrán un orden relacional, por ejemplo: 2 es menor a 5 y 5 es menor a 7 y,
2. Además poseen una distancia relacional entre ellos, por ejemplo: la distancia presente entre 4.2 y 2.3 es 1.9 (Kantardzic, 2011).

3.4.1.2 Datos categóricos

Los datos categóricos, también llamados simbólicos, son variables que no poseen relación entre ninguna de ellas. Los datos de las variables categóricas pueden ser iguales o no iguales. Esos valores son soportados por la relación de igualdad, ejemplo: azul es igual a azul o rojo es diferente a negro (Kantardzic, 2011).

Una variable categórica con *n* valores puede convertirse en *n* variables numéricas binarias, es decir, una variable binaria para cada valor categórico. Estas variables categóricas codificadas se conocen como "variables ficticias" en las estadísticas. Otra manera de clasificar una variable, basada en sus valores, es observarlas como si fueran variables continuas o variables discretas (Kantardzic, 2011).

3.4.2 Clasificación de variables

Cada valor esta descrito de acuerdo con sus características con lo que se agruparan de acuerdo sus características: en variables cuantitativas y en variables discretas (Kantardzic, 2011).

3.4.2.1 Variables continuas

Las variables continuas son también conocidas como variables cuantitativas o métricas. Son medibles usando una escala de intervalos o una escala de relación. Ambas escalas permiten a la variable subyacente ser definida o medible teóricamente con una infinidad de precisión (Kantardzic, 2011).

3.4.2.2 Variables discretas

Las variables discretas son también llamadas variables cualitativas. Tales variables son medibles, o sus valores son definidos, usando una de dos tipos de escalas no métricas - nominales u ordinales (Kantardzic, 2011).

3.4.2.3 Variables ordinales

Una escala ordinal consiste en graduaciones ordenadas y discretas, por ejemplo, clasificaciones. Una variable ordinal es una variable categórica por lo cual un orden relacional es definido pero no una distancia relacional. Algunos ejemplos de variables ordinales, son el rango de un estudiante de una clase o las posiciones de medalla de oro, plata y bronce en una competencia deportiva (Kantardzic, 2011).

La escala ordenada no tiene por qué ser necesariamente lineal; por ejemplo, la diferencia entre los estudiantes clasificados en cuarto y quinto lugar no tiene que ser idéntica a la diferencia entre los estudiantes en el puesto 15 y 16. Todo lo que puede establecerse a partir de una escala ordenada para atributos ordinales con relaciones mayores que iguales o menores que las relaciones (Kantardzic, 2011).

3.4.2.4 Variables especiales

Un tipo de variable especial sería las variables periódicas. Una variable periódica es una característica que existe en la relación de la distancia, pero no hay una relación de orden. Algunos ejemplos de este tipo de variables son los días de la semana, los días del mes o los días del año. Por ejemplo el lunes y el martes, viéndose como valores de una característica, están más cerca que el lunes y el jueves, sin embargo el lunes puede ser anterior o posterior al viernes (Kantardzic, 2011).

Capítulo 4. Metodología de Extracción, Transformación y Carga

Para la aplicación de la solución se utilizó la metodología de extracción, transformación y carga de la información a tablas o repositorios para su análisis en la toma de decisiones (ETL por sus siglas en inglés) (Miller, 2019).

ETL es un proceso el cual se encarga de extraer información (la cual no está optimizada para analizar), y necesita ser alojada en una terminal central. El proceso ETL cubre las tareas necesarias para el trabajar con big data (Miller, 2019).

El proceso tradicional de ETL se inicia cuando los datos son extraídos de las bases de datos, que por ejemplo puedan ser utilizados en un procesamiento de transacciones en línea (OLTP por sus siglas en inglés), hoy más comúnmente conocido como "bases de datos transaccionales". Las aplicaciones OLTP tienen un alto rendimiento y soportan un gran número de solicitudes de lectura y escritura, he aquí un motivo por el cual ETL es de un alto interés al utilizar big data (Miller, 2019).

Una vez que la extracción ha sido terminada, los datos se deben transformar en un área de prueba, estas transformaciones cubren tanto la limpieza de datos como la optimización de los datos para el análisis (Miller, 2019).

Los datos transformados luego se cargan en una base de datos de procesamiento analítico en línea (OLAP por sus siglas en inglés), hoy más comúnmente conocida como solo una base de datos analíticos. La información puede ser alojada en diferentes modelos de bases de datos, pero es importante que se realice este paso, previo a cualquier análisis necesario para una toma de decisiones (Miller, 2019).

Esta metodología se aplica en el caso práctico siguiendo estos pasos:

1. Extracción de datos que han sido registrados en los logs de los servidores.
2. Transformación de los datos a información entendible y procesable; dado a que los datos fueron registrados en los logs como datos crudos no procesables, se requiere una transformación lo cual se procede mediante un script. En conjunto con la transformación de los registros se procede a una limpieza de datos para tener solo información útil para el proceso.
3. Carga de los datos, donde la información que ya se encuentra en una estructura y con los registros necesarios es almacenada en tablas, en este caso en elasticsearch.

Realizando estos pasos definidos en la metodología ETL se puede proceder al análisis y el procesamiento, lo cual se efectúa por medio de un método de aprendizaje automático.

4.1 Solución propuesta

Se propone un método que permita el análisis de los datos que se han recolectado sobre la actividad de los servidores. De acuerdo con la estructura del flujo de información de la organización, el método se sitúa en el punto 4 de la figura 1 (anteriormente mostrada), donde se permite decidir si el acceso al servicio será permitido o no. Para esto, se propone una técnica de aprendizaje automático, a través de un algoritmo de entrenamiento y aprendizaje que haga posible identificar patrones en el comportamiento de los datos y detectar anomalías identificadas como accesos inválidos.

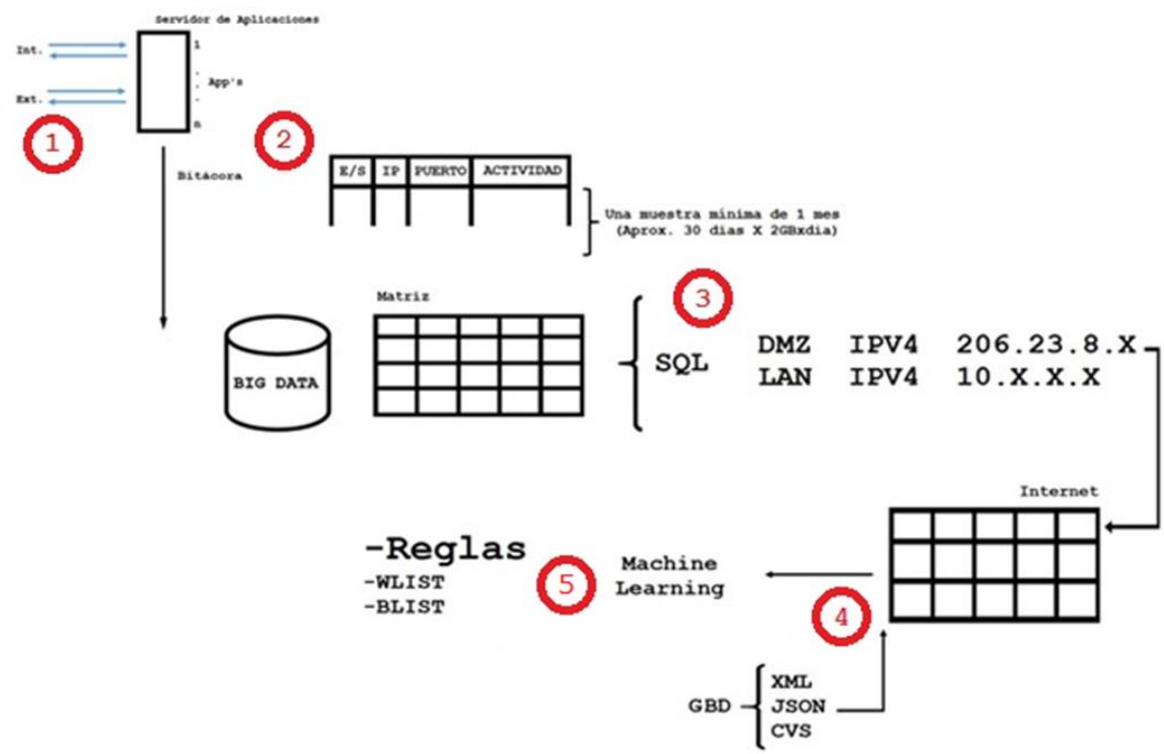

Figura 2. Flujo de solución propuesta

La figura 2 describe el flujo de la solución propuesta, el cual es descrito a continuación:

1. Extraer los datos a procesar en el mismo formato con el que anteriormente ya fue almacenada en los logs de los servidores (como parte de su proceso normal).
2. Limpiar la información contenida en los logs de los servidores con el fin de tener información limpia y procesable. Este es un punto de gran valor debido a que la información como se encuentra en los logs no es posible procesarla ni analizarla de manera inmediata, además de que dichos logs contienen datos que no son necesarios en el modelo y hacen más difícil su procesamiento. Debido a que en cualquier proceso donde se involucra Big Data, los datos reunidos no cuenta con un formato ni una lectura entendible por lo que es necesario llevarlo a una estructura utilizable. Para realizar la limpieza de datos se realiza una lectura

completa de los logs de servidores, donde son descartados los registros que carecen de valor para el sistema y se generan columnas de la información necesaria para poder realizar el entrenamiento del sistema.

3. Identificar los registros que corresponden a las direcciones IPs y conformar una lista para su análisis. Al estar analizando una gran cantidad de registros es necesarios agrupar todos en una lista, la cual será sometida a un análisis respecto al algoritmo de aprendizaje automático.
4. Analizar la información por medio de una técnica de aprendizaje automático que permita “enseñar” a un equipo con alta capacidad de procesamiento (con supervisión o sin supervisión) aprender la ruta del proceso para obtener la información de IPs externas
5. Filtrar los registros de acuerdo con el algoritmo de aprendizaje automático T-digest para la detección de anomalías. Se fijan percentiles del 95% al 99% (de acuerdo con la observación se considerará cual será el más eficiente) y se evalúan los valores de los campos. Todo valor que exceda los parámetros determinados en el percentil elegido, será declarado un posible acceso anómalo y es considerado como brecha de seguridad en el tráfico de datos.

Empleando este proceso se permite a la organización:

- ✓ Tener una herramienta para realizar una limpieza de datos de los logs de servidores, la cual en este caso es utilizada para detección de anomalías, sin embargo se está dando una herramienta que puede ser utilizada en futuros proyectos que involucre analizar la información de los accesos a servidores.
- ✓ Determinar, dentro de los datos recabados de los logs de servidores, cuales son registros normales y cuales son registros anómalos. En el caso de algún registro sea determinado como acceso anómalo, el proceso indica que se ha detectado una amenaza a la seguridad. Todos estos registros se continúan almacenado en la base de datos para alimentar el entrenamiento del algoritmo.

A continuación, se hace una descripción detallada de los pasos de la metodología empleada en la solución con su respectiva fundamentación.

4.1.1 Recolección de datos

4.1.1.1 Datos en “crudo” (Raw Data)

Los logs de servidores que se utilizan en el trabajo se encuentran en un formato que se denomina datos en “crudo” (raw data, por su traducción en inglés). Dicha información no posee un formato de fácil acceso y procesable por los métodos a sugerir y en los cuales sus campos no se encuentran estructurados ni definidos claramente (Kantardzic, 2011).

Los datos en "crudo" se refieren a cualquier objeto de datos que no se haya sometido a un procesamiento completo, ya sea manualmente o mediante un software informático automatizado. Los datos sin procesar se pueden recopilar de diversos procesos y recursos de TI. En este caso fueron recopilados de logs de servidores del INEGI.

Los datos en "crudo" también se conocen como datos de origen, datos primarios o datos atómicos (Kantardzic, 2011).

Estos datos son principalmente datos de repositorio no estructurados o sin formato. Los datos, son extraídos, analizados, procesados y utilizados por personas o aplicaciones de software especialmente diseñadas para extraer conclusiones, hacer proyecciones o extraer información significativa (Kantardzic, 2011).

Todos los datos en "crudo" a usar en la minería de datos suelen ser grandes cantidades de información; muchos están relacionados con la generación de datos por los patrones de uso por los usuarios y tienen el potencial de ser desordenados. Es necesario poder generar un proceso de interpretación para su lectura y estructuración (normalización) con el fin de poder entender la información, analizarla y obtener resultados (Kantardzic, 2011).

Algunos métodos de extracción de datos, generalmente aquellos que se basan en el cálculo de la distancia entre puntos en un espacio n-dimensional, pueden necesitar datos normalizados para obtener los mejores resultados. La normalización de datos es el proceso sistemático de la descomposición de los datos para eliminar la redundancia en la información y las características no deseadas que pueden ser generadas en el momento de insertar, actualizar y eliminar registros (Kantardzic, 2011).

Algunos métodos de extracción de datos, generalmente aquellos que se basan en el cálculo de la distancia entre puntos en un espacio n-dimensional, pueden necesitar para obtener mejores resultados el uso de datos normalizados. Los datos normalizados son aquellos que han sido sometidos a un proceso sistemático de la descomposición de los datos, para eliminar la redundancia en la información y de las características no deseadas, que pueden ser generadas en el momento de insertar, actualizar y eliminar registros.

Si los valores no están normalizados, los datos se considerarán con criterios incorrectos, en promedio, valores más grandes.

La figura 3, presenta un fragmento de la información recolectada de uno de los logs de los servidores del INEGI la cual se encuentra sin formato y como se menciona son datos en “crudo”.

```
#Software: IIS Advanced Logging Module
#Version: 1.0
#Start-Date: 2019-02-13 00:00:00.089
#Filter: (URI-Stem != ^/status\.html)
#Fields:  date-local time-local s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username NS-IPClient c
2019-02-12 18:00:00.307 10.153.2.72 POST /app/mapa/Denue/Default.aspx/busquedaPuntosOffset - 80 - "148.2
2019-02-12 18:00:00.542 10.153.2.72 POST /app/mapa/Denue/Default.aspx/RegistroConsultas - 80 - "148.222.
2019-02-12 18:00:00.776 10.153.2.72 POST /app/mapa/denue/Default.aspx/busquedaEstablecimientos - 80 - "1
2019-02-12 18:00:01.354 10.153.2.72 GET /app/mexicoenmapas/servicios/cdt.ashx http://gaiamapas.inegi.org
2019-02-12 18:00:01.370 10.153.2.72 GET /app/mexicoenmapas/servicios/cdt.ashx http://gaiamapas.inegi.org
2019-02-12 18:00:02.588 10.153.2.72 POST /app/mapa/inv/Default.aspx/CentroLoc_Ageb - 80 - "177.234.12.15
2019-02-12 18:00:02.698 10.153.2.72 POST /app/mapa/inv/Default.aspx/CentroLoc_Ageb - 80 - "189.158.97.23
2019-02-12 18:00:03.604 10.153.2.72 GET /app/mapa/denue/default.aspx - 80 - "201.137.176.37" "Mozilla/5.
2019-02-12 18:00:03.604 10.153.2.72 GET /app/mapa/denue/ - 80 - "201.137.176.37" "Mozilla/5.0 (Windows N
2019-02-12 18:00:03.604 10.153.2.72 GET /app/mapa/denue/ - 80 - "201.137.176.37" "Mozilla/5.0 (Windows N
2019-02-12 18:00:03.667 10.153.2.72 POST /app/mapa/denue/Default.aspx/busquedaEstablecimientos - 80 - "1
2019-02-12 18:00:03.760 10.153.2.72 POST /app/mapa/inv/Default.aspx/CentroLoc_Ageb - 80 - "189.158.97.23
2019-02-12 18:00:03.901 10.153.2.72 POST /app/mapa/denue/Default.aspx/MenuAreaGeo - 80 - "201.137.176.37
2019-02-12 18:00:03.932 10.153.2.72 POST /app/mapa/denue/Default.aspx/MenuPersonalOcup - 80 - "201.137.1
2019-02-12 18:00:03.932 10.153.2.72 POST /app/mapa/denue/Default.aspx/MenuActividadEco - 80 - "201.137.1
2019-02-12 18:00:04.307 10.153.2.72 GET /app/mapa/denue/default.aspx - 80 - "201.137.176.37" "Mozilla/5.
2019-02-12 18:00:04.307 10.153.2.72 GET /app/mapa/denue/ - 80 - "201.137.176.37" "Mozilla/5.0 (Windows N
2019-02-12 18:00:04.307 10.153.2.72 GET /app/mapa/denue/ - 80 - "201.137.176.37" "Mozilla/5.0 (Windows N
2019-02-12 18:00:04.385 10.153.2.72 POST /app/mapa/inv/Default.aspx/CentroLoc_Ageb - 80 - "177.234.12.15
2019-02-12 18:00:04.870 10.153.2.72 GET /app/mapa/denue/default.aspx - 80 - "201.137.176.37" "Mozilla/5.
2019-02-12 18:00:04.870 10.153.2.72 GET /app/mapa/denue/ - 80 - "201.137.176.37" "Mozilla/5.0 (Windows N
2019-02-12 18:00:04.870 10.153.2.72 GET /app/mapa/denue/ - 80 - "201.137.176.37" "Mozilla/5.0 (Windows N
2019-02-12 18:00:05.073 10.153.2.72 GET /app/mapa/denue/Images/IconActEco/esf_act_11.png - 80 - "201.137
2019-02-12 18:00:05.104 10.153.2.72 GET /app/mapa/denue/Images/IconActEco/esf_act_21.png - 80 - "201.137
```

Figura 3. Ejemplo de datos en "crudo" en log de servidores.

4.1.2 Limpieza de datos

Al tener disponible la información de los logs de servicios se lleva a cabo el proceso denominado limpieza de datos, el cual es la tarea de estructurar la información extraída con el objetivo de obtener los campos y los datos necesarios a utilizar para la evaluación.

Para este caso, el proceso de limpia de datos se llevó a cabo por medio de una aplicación desarrollada en el lenguaje de programación Python. Dicha aplicación inicia con la eliminación de registros que sirven de encabezados y son descriptivos en el log inicial. Quedando solamente los registros del log. De esta forma se puede segmentar cada uno de los campos que son presentados. Cada valor se encuentra separado por un espacio por lo que se segmentan, tomando esa consideración, en campos diferentes. Una vez separados, el programa se encarga de seleccionar los campos que se necesitan utilizar en el proceso.

Después de haber realizado el proceso de limpieza de datos extraídos de los logs que se representan en la figura 3. Dicha figura muestra un extracto de los datos en crudo de los logs y los cuales son listados en la tabla 1:

Tabla 1. Datos extraidos y descripción

Dato extraído	Descripción
Date Time	Fecha y hora de acceso a la información. La fecha y la hora se encuentran separados por un espacio vacío.
Method	Método de conexión con el que el usuario accede y trabaja con la información del servidor, este puede ser método GET o método POST
Server	Ruta definida dentro del servidor donde se encuentra la información consultada o alojada
Status[c1]	Status que arroja el servidor con relación al acceso a la información
Status_win32[c2]	Es una bandera que indica cuando una descarga se realizó de manera exitosa (status 0) y no fue abortada (status 64) del cliente)
sc_bytes[c3]	Cantidad de información en bytes que se mandan desde el servidor hacia el cliente
cs_bytes[c4]	Cantidad de información en bytes que se mandan desde el cliente hacia al servidor
time_taken[c5]	Tiempo que dura el enlace establecido con el servidor
NS-IPClient	IP del cliente que está accediendo a la información

La figura 4 muestra cómo se visualiza la información, después de que se ejecuta el proceso de limpieza de datos sobre el log y se generan cada uno de los campos listados en la tabla:

	A	B	C	D	E	F	G	H	I
1	10/01/2020 00:00	GET	https://www.inegi.org.mx/inegi/quienes_somos.htm	200	0	1561	1052	15	200.68.136.203
2	10/01/2020 00:00	GET	https://www.inegi.org.mx/sistemas/olap/consulta/general_ver4/MDXQueryDatos.asp?proy=sh_pty	200	0	2592	592	6	10.74.1.236
3	10/01/2020 00:00	GET	https://www.inegi.org.mx/datos	200	0	3340	728	109	187.201.24.13
4	10/01/2020 00:01	GET	https://inegi.org.mx/app/mapa/inv	200	0	39750	619	31	187.248.71.2
5	10/01/2020 00:01	GET	https://www.inegi.org.mx/datos/?t=015	200	0	1946	746	46	200.56.17.98
6	10/01/2020 00:01	GET	https://www.inegi.org.mx/programas/ccpv/2020/default.htm	200	0	1716	555	15	187.188.64.104
7	10/01/2020 00:02	GET	https://www.inegi.org.mx/app/buscador/default.html?q=anuario+estadistico+de+aguascaliente	200	0	16357	770	15	201.162.167.67
8	10/01/2020 00:02	GET	https://inegi.org.mx	200	0	7721	581	15	189.183.15.182
9	10/01/2020 00:02	GET	https://www.inegi.org.mx/app/spc/plazasenconcurso.asp	200	0	778	909	77	189.217.43.213
10	10/01/2020 00:03	GET	https://www.inegi.org.mx/datos/?t=012	200	0	1847	769	31	201.175.202.80
11	10/01/2020 00:03	GET	https://www.inegi.org.mx/app/spc/Registro.asp	200	0	1977	547	15	189.230.38.18
12	10/01/2020 00:03	GET	https://www.inegi.org.mx/temas/estructura	200	0	19800	660	15	189.189.179.37
13	10/01/2020 00:04	GET	https://www.inegi.org.mx/app/indicadores/?ind=620448221	200	0	1615	809	15	187.216.130.147
14	10/01/2020 00:04	GET	http://gaiamapas.inegi.org.mx/mdmCache/service/wms&SERVICE=WMS&VERSION=1.3.0&REQUEST=	200	0	13146	1084	0	177.241.101.178
15	10/01/2020 00:04	GET	http://gaiamapas.inegi.org.mx/mdmCache/service/wms&SERVICE=WMS&VERSION=1.3.0&REQUEST=	200	0	4930	1135	15	189.203.188.206
16	10/01/2020 00:05	GET	https://www.inegi.org.mx	200	0	2026	752	0	187.216.130.147
17	10/01/2020 00:05	GET	https://www.inegi.org.mx/app/buscador/default.html?q=mapa+digital+de+mexic	200	0	3726	764	78	200.194.58.11
18	10/01/2020 00:05	GET	https://www.inegi.org.mx/temas/uma/?fbclid=IwAR16k79TrqForkzkpFtOXQjNVZ-xMa_lZDpd2cYNtFI	200	0	1504	604	15	200.68.137.237
19	10/01/2020 00:05	GET	https://www.inegi.org.mx/temas/inpc	200	0	11728	547	109	177.236.63.85
20	10/01/2020 00:06	GET	http://gaiamapas.inegi.org.mx/mdmCache/service/wms&SERVICE=WMS&VERSION=1.3.0&REQUEST=	200	0	2913	1259	15	189.252.158.211
21	10/01/2020 00:06	GET	https://www.inegi.org.mx/temas/inpc	200	0	1488	793	15	200.52.75.34
22	10/01/2020 00:06	GET	https://www.inegi.org.mx/temas/uma	200	0	3762	586	62	189.136.226.132
23	10/01/2020 00:07	GET	https://www.inegi.org.mx/datos	200	0	523	509	15	187.150.161.74

Figura 4. Información del log post proceso de limpia

4.1.2.1 Reglas de datos

A continuación se describe cuáles serían las reglas por seguir (lineamientos necesarios) en los registros para que el sistema pueda procesarlos. Aún cuando se está realizando una limpieza de datos es necesario que se cumpla con un mínimo de requerimientos en sus registros para poder funcionar:

- ✓ Línea de log que inicie con una fecha/hora (en ese específico orden) de la creación del registro. Registro que no inicie con esos dos valores (fecha/hora) será descartado del sistema.
- ✓ La hora solo se permite que este definida como hh:mm:ss o hh:mm:ss:ms:ms el sistema no podrá laborar con valores diferentes a los descritos para las horas.
- ✓ Todo registro para ser procesado debe contar con:
  - Método de acceso (GET/POST)
  - Ruta en el servidor a la cual se accedió, esta debe empezar por un **http:** o **https:**
  - cinco códigos de retorno, los cuales hacen referencia al código de ejecución, tiempos, y cantidad de bytes transmitidos. En caso de que falte un valor el sistema asignara un valor de cero al campo Time_taken (que corresponde al tiempo que duro la conexión con el servidor) dado a que cada uno se encuentra separado solo por un espacio y no hay forma de hacer distinción entre las columnas.
  - Una dirección IP que intenta acceder a los registros

4.1.3 Preparación de datos

Teniendo listo la recolección y la limpieza de los datos, se procede a preparar cada uno de los datos a un formato entendible para el algoritmo a utilizar.

Para ello es necesario definir cuáles son los distintos tipos de datos que pueden ser los utilizados y distinguir cual son los valores que pueden adoptar y como se pueden tratar.

4.1.3.1 Definición de estructura de datos a explorar

En la tabla 2 se describe el diccionario de datos, integrado por el nombre del campo, su tipo de variable que lo define y la descripción de cada uno.

Tabla 2. Campos, tipo de variables y descripción

Campo	Tipo de variable	Descripción
Date Time	Continuas	Fecha y hora de acceso a la información. La fecha y la hora se encuentran separados por un espacio vacío.
Method	Categórico	Método con el que se trabaja la información en el servidor (GET y POST)
Server	Categórico	Ruta dentro del servidor donde la información es consultada o alojada
Status	Continuas	Status arrojado del acceso al servidor
Status_win32	Continuas	Es una bandera la cual nos indica cuando una descarga se realizó de manera exitosa (status 0) y no fue abortada (status 64) por parte del cliente
SC_bytes	Numéricos	Cantidad de información en bytes que se manda desde el server hacia el cliente
CS_bytes	Numéricos	Cantidad de información en bytes que se manda desde el cliente al servidor
Time_taken	Numéricos	Tiempo que duro el enlace con el servidor
NS-IPClient	Categórico	IP del cliente que accede a la información

4.1.4 Normalización de datos

Como ya se mencionó previamente, la normalización de datos es el proceso sistemático de la descomposición de los datos para eliminar la redundancia en la información y las características no deseadas que pueden ser generadas en el momento de insertar, actualizar y eliminar registros. Es un proceso de varios pasos que coloca los datos en forma tabular, eliminando los datos duplicados de las tablas de relaciones (Kantardzic, 2011).

La normalización se utiliza principalmente para dos propósitos:

1. Eliminar datos redundantes (inútiles).
2. Asegurar que las dependencias de datos tengan sentido, es decir, los datos se almacenan de forma lógica.

Para los campos SC_bytes, CS_bytes y Time_taken se prueba con el método de normalización de mínimos y máximos (Kantardzic, 2011).

El método de mínimos y máximos se enfoca que una característica X, está en un rango entre 150 y 250. Entonces, el método anterior de normalización dará toda la información normalizada entre .15 y .25, pero esta acumulará los valores en un subintervalo de un rango completo.

Para obtener la distribución de valores en un intervalo completo normalizables, por ejemplo, considerando 0 y 1, podemos usar la fórmula 1 de mínimo y máximo que se muestra a continuación:

$$v´(i) = (v(i)-min[v(i)]/(max[v(i)]-min[v(i)]) \quad (1)$$

Donde el valor mínimo y el valor máximo de los valores de la característica X son computados automáticamente en un grupo, o son estimados por un experto en un dominio específico. Una transformación similar puede ser usada para normalizar el intervalo de -1 a 1 (Kantardzic, 2011).

En la figura 5 se muestra como los datos de los campos SC_bytes, CS_bytes y Time_taken fueron procesados mediante la normalización, con una precisión de 6 decimales:

	A	B	C	D	E	F	G	H	I
1	GET	https://www.inegi.org.mx/	200	0	0.000672	0.011132	0.000572	187.237.25.37	
2	GET	https://www.inegi.org.mx/400.html?aspxerrorpath=/ineg	200	0	0.008309	0.004873	0.000357	189.215.224.158	
3	GET	https://www.inegi.org.mx/app/descarga/?ti=13&ag=00	200	0	0.001548	0.013491	0.000069	187.160.9.127	
4	GET	https://www.inegi.org.mx/400.html?aspxerrorpath=/ineg	200	0	0.000758	0.005104	0.000499	189.215.224.158	
5	GET	https://www.inegi.org.mx/componentes/biinegi/buscad	200	0	0.001044	0.016518	0.00021	189.218.223.233	
6	GET	https://www.inegi.org.mx/app/descarga/?ti=13&ag=00	200	0	0.006562	0.012748	0.000142	187.160.9.127	
7	GET	https://www.inegi.org.mx/app/descarga/?ti=13&ag=00	200	0	0.001426	0.01285	0.000142	187.160.9.127	
8	GET	https://www.inegi.org.mx/default.html	200	0	0.010637	0.019211	0.000714	189.181.223.236	
9	GET	https://www.inegi.org.mx/default.html	200	0	0.001327	0.018724	0.00021	189.181.223.236	
10	GET	https://www.inegi.org.mx/app/descarga/?ti=13&ag=00	200	0	0.001202	0.013286	0.000425	187.160.9.127	
11	GET	https://www.inegi.org.mx/app/descarga/?ti=13&ag=00	200	0	0.006933	0.013132	0.000069	187.160.9.127	
12	GET	https://www.inegi.org.mx/default.html	200	0	0.000962	0.018749	0.000499	189.181.223.236	
13	GET	https://www.inegi.org.mx/app/descarga/?ti=13&ag=00	200	0	0.00946	0.012927	0.000069	187.160.9.127	
14	GET	https://www.inegi.org.mx/app/descarga/?ti=13&ag=00	200	0	0.010637	0.012799	0.000572	187.160.9.127	
15	GET	https://www.inegi.org.mx/default.html	200	0	0.001423	0.019493	0.000069	189.181.223.236	
16	GET	https://www.inegi.org.mx/400.html?aspxerrorpath=/ineg	200	0	0.002845	0.005925	0.00021	189.215.224.158	
17	GET	https://www.inegi.org.mx/400.html?aspxerrorpath=/ineg	200	0	0.000866	0.005771	0.000499	189.215.224.158	
18	GET	https://www.inegi.org.mx/default.html	200	0	0.000758	0.019108	0.000425	189.181.223.236	
19	GET	https://www.inegi.org.mx/app/descarga/?ti=13&ag=00	200	0	0.001525	0.013107	0.000069	187.160.9.127	
20	GET	https://www.inegi.org.mx/400.html?aspxerrorpath=/ineg	200	0	0.001866	0.008233	0	189.215.224.158	
21	GET	https://www.inegi.org.mx/default.html	200	0	0.000776	0.01916	0.000567	189.181.223.236	
22	GET	https://www.inegi.org.mx/400.html?aspxerrorpath=/ineg	200	0	0.053368	0.006694	0.00064	189.215.224.158	
23	GET	https://www.inegi.org.mx/app/descarga/?ti=13&ag=00	200	0	0.001423	0.013081	0.000069	187.160.9.127	

Figura 5. Infromación de log post proceso de normalización.

Hasta este punto los datos han sido recolectados, sometidos a un proceso de limpia y preparados en un formato específico. Sin embargo este punto nos demuestra que la propuesta de utilizar la normalización de los datos no es la óptima para este análisis.

Nota importante: Con el proceso de normalización de los datos, se detectó que los campos seleccionados (SC_bytes, CS_bytes y Time_taken) arrojaban muchos valores diferentes, por lo que no se podía establecer un pico en los datos, ya que eran varios picos sin un patrón.

Tomando este resultado y comparándolo con los datos normales (datos originales obtenidos del proceso de limpia), se opta por considerar a los valores sin normalizar, esto en los campos SC_bytes, CS_bytes y Time_taken.

4.1.5 Almacenamiento de datos

Una vez realizada la limpieza de datos de los logs se puede cubrir el tema de almacenamiento. Dicha información es trabajada en una base de datos NoSQL llamada elasticsearch (Elasticsearch, 2020).

El termino NoSQL no hace referencia a que no posee SQL si no a “Not only SQL” esto significa que la tecnología empleada en dichas bases de datos no solo opere con SQL y que las tecnologías empleadas puedan coexistir entre ellas. Plantea modelos de datos específicos de esquemas flexibles que se adaptan a los requisitos de las aplicaciones más modernas.

A diferencia de las bases de datos tradicionales, en la mayoría de los sistemas que utilizan NoSQL, no se implementan mecanismos rígidos de consistencia que garanticen que cualquier cambio llevado a cabo en el sistema distribuido sea visto, al mismo tiempo, por todos los nodos y asegurando, también, la no violación de posibles restricciones de integridad de los datos.

4.1.6 Determinación de Datos Normales y Anómalos

Ya que se cuenta con los datos almacenados y con cierta estructura que permite distinguir cada uno de ellos y poder analizarlos, se procede a determinar que puede ser normal o anómalo dentro de la muestra (Dunning, 2014).

Para poder determinar dentro de una muestra que valores son normales, primero hay que identificar y describir el comportamiento y patrones de la muestra completa, y así poder tener un particular punto de observación sobre lo que se intenta evaluar. Una forma de poder determinar si los valores son normales es mediante un modelo probabilístico (Dunning, 2014).

Si se asume que los valores siguen un comportamiento particular, se describe entonces dicho valor de comparación como lo que sería normal y todo aquel valor que no siga ese comportamiento se podría determinar como algo anómalo (Dunning, 2014).

Descubrir patrones normales no solo depende de un buen modelo de aprendizaje automático, también depende de la interacción humana para decidir cuando los valores pasan a ser de normales a anómalos y de esa forma que tenga sentido con la situación deseada. Gráficamente hablando un valor pasa a ser anómalo cuando este representa un pico con el resto de los datos graficados (Dunning, 2014).

4.1.7 Aplicación del algoritmo de aprendizaje automático T-digest

Para poder determinar cuáles datos son normales o anómalos dentro de las muestras que son introducidas al sistema, se procede con un algoritmo de aprendizaje automático.

Dicho algoritmo busca detectar los accesos que representan alertas de brechas de seguridad en el tráfico de los datos, esto significa implementar un algoritmo para la detección de anomalías. El algoritmo que se utiliza es el llamado T-digest (Dunning, 2014).

Una de las formas más comunes para detectar anomalías hoy en día es configurar manualmente una alarma lo que en inglés se conoce como; “manually-set threshold alarm”, lo anterior para alertar de posibles anomalías. Esta alarma seria alimentada por valores de medición numéricos de cualquier tipo. La idea de dicha alarma es que cuando un valor de los que alimentan el sistema sobrepase el límite establecido se active la alarma (Dunning, 2014).

Sin embargo esta idea se complica si la cantidad de datos a analizar es enorme. La solución que se pensaría como la más lógica sería el aumentar el valor del límite establecido, sin embargo eso significaría generar valores como verdaderos que resultarían siendo alarmas no detectadas. De igual forma si el límite se reduce para tratar de detectar más fallas, provocaría más falsas alertas. Por lo que la selección de dicho límite o umbral debe ser lo más apropiada y certera posible, de lo contrario el sistema arrojaría un gran número de falsas alarmas (Dunning, 2014).

Por lo que utilizando el algoritmo T-digest buscamos reducir el número de falsas alarmas con el fin de volver el sistema lo óptimo posible. Así pues, para entender correctamente el uso del algoritmo T-digest es necesario entender los conceptos Cuartiles y Percentiles, los cuales se describen a continuación.

4.1.7.1 Cuartiles

Los cuartiles corresponden a los valores que tiene una variable y que cumplen con la función de dividir los datos ordenados en cuartos o cuatro partes con igual valor porcentual. Se distinguen en principio tres cuartiles, que se denotan regularmente con la letra Q: Q1, Q2 y Q3 (Dunning, 2014).

Q1 - primer cuartil, representa un valor por debajo del cual quedan un cuarto o 25% de los valores, previamente ordenados.

Q2 - segundo cuartil, es considerado como la mediana.

Q3 - tercer cuartil, representa a su vez el valor por debajo del que queda el 75% de todos los datos (Dunning, 2014).

4.1.7.2 Percentiles

Los percentiles, son otras de las medidas de posición más comunes y empleadas. Técnicamente, son definidos como ciertos valores que dividen en cien partes idénticas porcentualmente hablando los datos que han sido ordenados de forma sucesiva de menor a mayor. En cuanto a su denotación, ésta corresponde a la forma P1, P2…. Pn, no obstante son leídas como Percentil 10, Percentil 90, etc. (Dunning, 2014).

4.1.7.3 Algoritmo T-digest

Como se mencionó anteriormente, una de las ventajas de T-digest es la precisión, especialmente para cuartiles extremos. En lugar de tener que clasificar una gran cantidad de muestras para estimar un cuartil de interés, un valor de entrada se puede analizar en línea usando un T-digest para encontrar el umbral correspondiente a cualquier cuartil (Dunning, 2014).

Un modelo de detector de anomalías basado en el límite o umbral, se basa en la suposición de que la entrada tiene una distribución casi simple, de modo que un percentil particular siempre estará en un punto particular. Si la suposición no se cumple, el umbral que es calculado por el algoritmo T-digest, generara una gran variedad de valores dispersos entre anomalías verdaderas y anomalías falsas a medida que los valores cambien en el paso del tiempo

Utilizando el algoritmo T-digest el valor de entrada es enrutado a través del sistema siguiendo el algoritmo estimado el cual fue establecido como un cuartil (Dunning, 2014).

En el sistema a implementar, una vez que la información ha sido limpiada y estructurada, se pueden tomar los campos SC_bytes (columna 6) y Time_taken (columna 8) para generar los limites, los cuales serán utilizados por el algoritmo T-digest.

La tabla 3 y 4 muestran los límites generados por cada uno de los percentiles del 95%-99% para los campos SC_bytes y Time_taken.

Tabla 3. Campo SC_bytes, percentiles y límites

Field SC_bytes	
Percentage	**Limits**
95%	1748.4643313953472
96%	2460.1616300940427
97%	2627.6483529554566
98%	4397.32837026648
99%	4614.8560903316875

Tabla 4. Campo Time_taken, percentiles y límites

Field Time_taken	
Percentage	**Limits**
95%	30.17992938620314
96%	31.814115742614323
97%	61.98478937598172
98%	267.2683383030674
99%	21207.895508021724

En este punto es donde es necesario el ajuste supervisado, donde se elige el límite óptimo por medio de técnica heurística (observación).

En el campo SC_bytes podemos ver que al llegar el percentil de 98% (P98) existe un salto considerable de los valores que se reflejan, un pico.

Sin embargo en el caso del campo Time_taken aún cuando hay un aumento entre el percentil de 97% y el percentil de 98%, el pico más grande se ve en el percentil 99%. Pese a eso se puede ver que el pico principal representa una cantidad considerable (no mínima) de registros superiores a 98%. Esto lo podemos confirmar con los valores a utilizar en la evaluación del modelo.

A continuación se muestran unas gráficas generadas en Kibana, con la información que se cargó en la base de datos de elasticsearch. Kibana es una aplicación desarrollada como código abierto para frontend y se encarga de proporcionar herramientas para la visualización de datos y apoyar en la busqueda de los datos que han sido indexados en Elasticsearch.

En la figura 6, se muestran los 3 campos que se consideraron inicialmente en evaluación con sus valores generados en el transcurso de 2 meses; SC_bytes (columna 6), CS_bytes (columna 7) y Time_taken (columna 8).

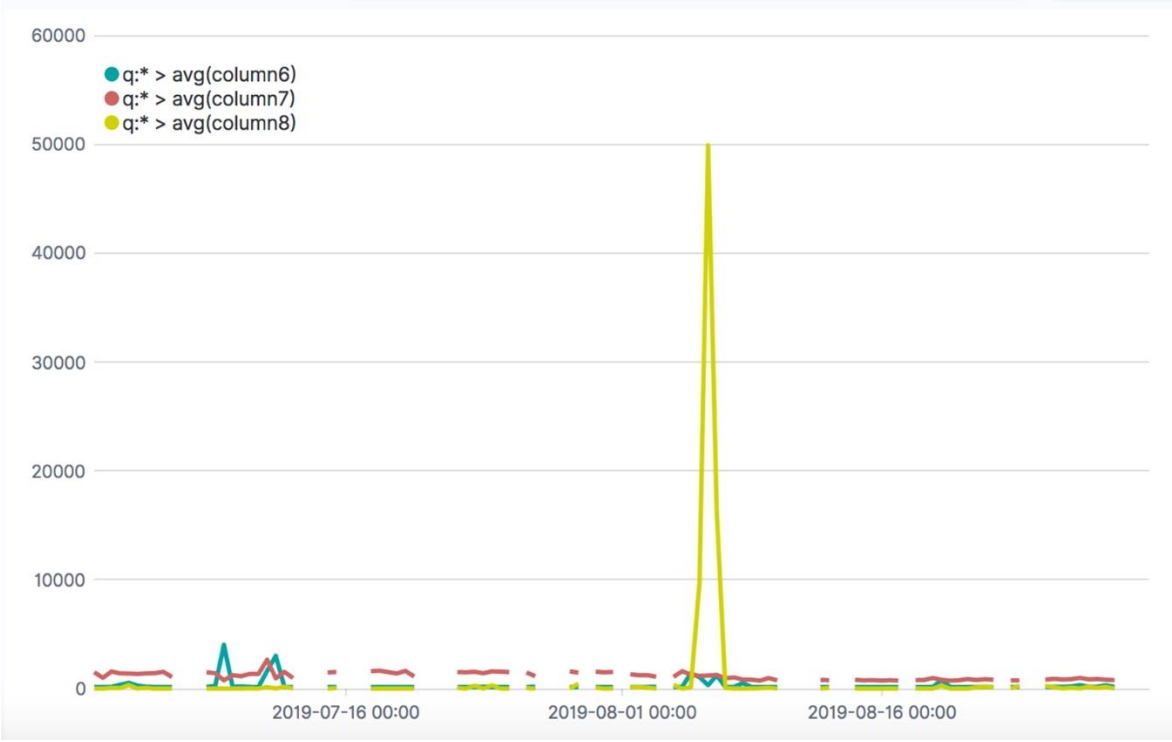

Figura 6. Presentación de los tres campos y su correlación.

En la figura 7, se muestra la extracción de solo los datos del campo SC_bytes. En esa gráfica se puede notar como existe un pico contra todos los datos presentes en la gráfica.

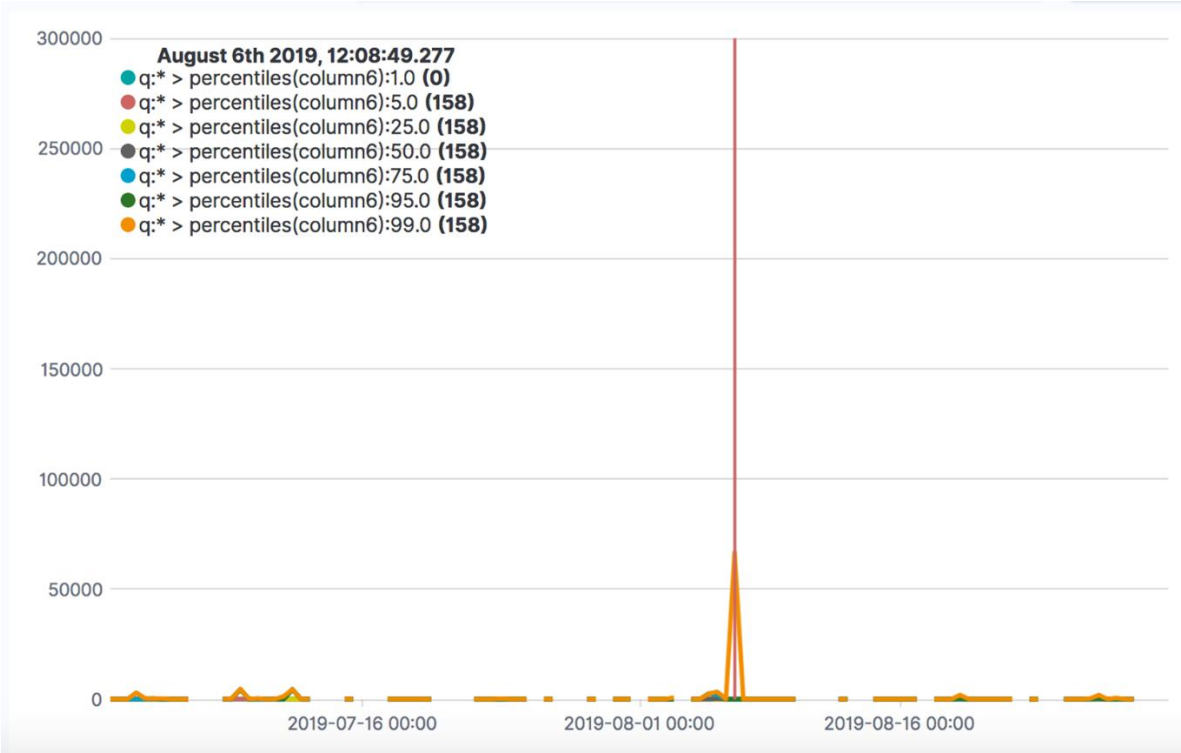

Figura 7. Cuartiles del campo SC_bytes.

En la figura 8, se muestra la extracción de solo los datos del campo CS_bytes. En esa gráfica se puede notar como existe una gran variedad de picos en los valores, sin embargo, no existe un salto considerable que pudiera considerarse de comportamiento anómalo. Por tal motivo se descarta el campo CS_bytes, esto con el fin de reducir la cantidad de falsas alarmas.

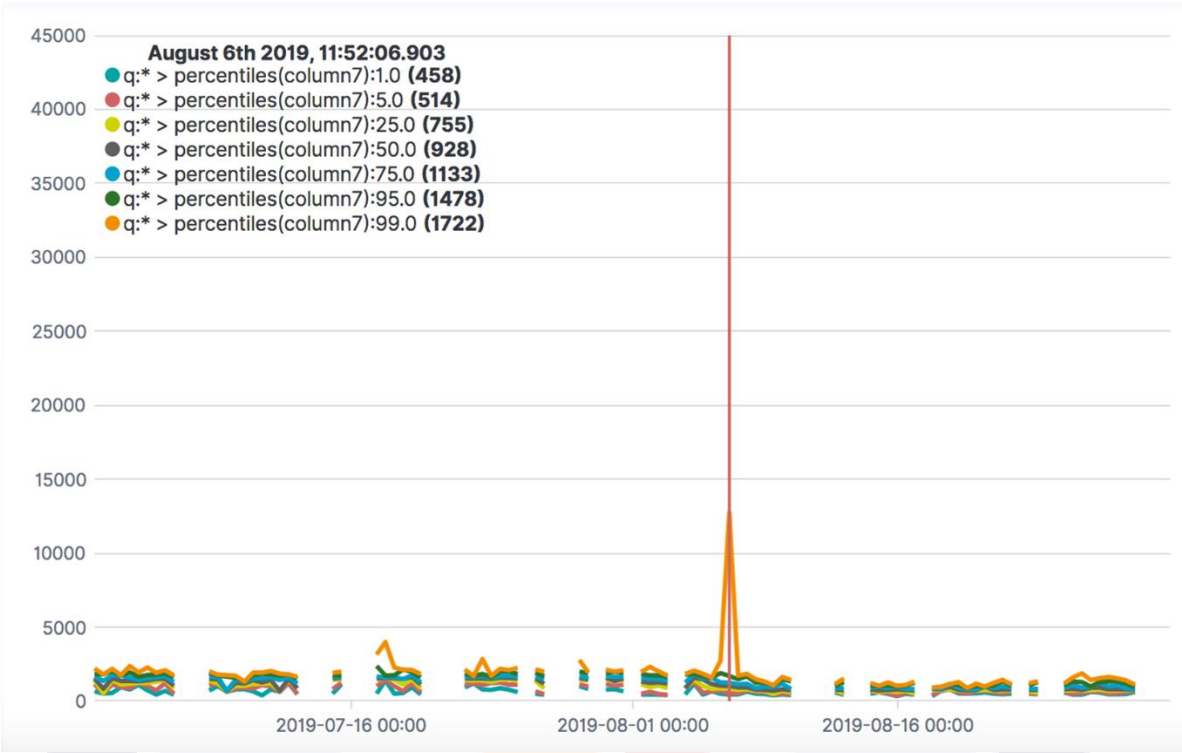

Figura 8. Cuartiles del campo CS_bytes.

En la figura 9, se muestra la extracción de solo los datos del campo Time_taken. En esa gráfica se puede notar como existe un pico contra todos los datos presentes en la gráfica.

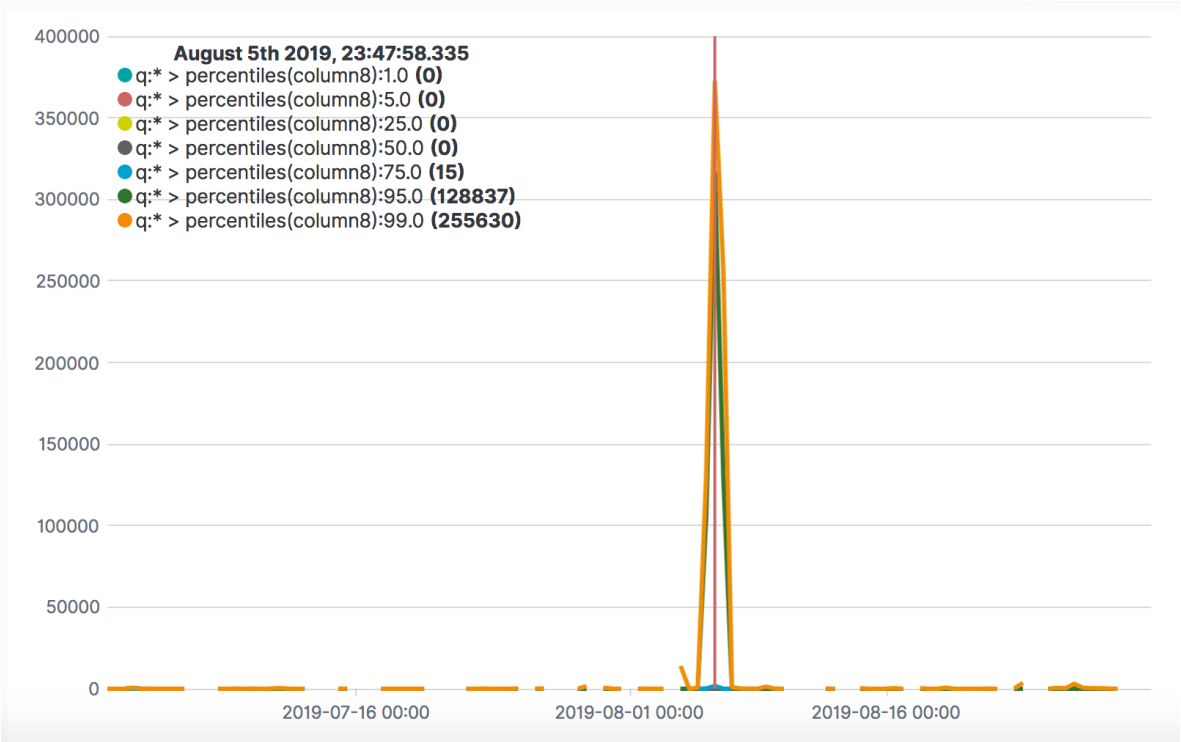

Figura 9. Cuartiles del campo Time_taken.

La idea básica detrás de cualquier detector de anomalías es poder construir un modelo, que pueda detectar y estimar lo que es “normal" y poder encontrar lo que sería una desviación dentro del grupo de datos. El modelo para el detector de anomalías basado en el límite o umbral, se basa en la suposición de que la información de entrada tiene una distribución casi estacionaria y simple, de modo que un percentil particular siempre estará en un punto particular. Si esta suposición no se cumple, entonces el umbral calculado por el T-digest dará como resultado que la tasa de positivos verdaderos y falsos varíe a medida que cambie la distribución de la señal (Dunning, 2014).

En la figura 10 se muestra un extracto del archivo final que se genera una vez que el algoritmo es implementado, por cuestiones graficas el ultimo campo refleja si el registrado es aceptado o rechazado de acuerdo con el algoritmo T-digest.

547	02:59.5	GET	https://www	500	-2147024883	222	781	15	10.152.21.8	ACCEPTED
548	03:15.5	GET	https://www	500	-2147024883	222	704	0	10.210.100.2	ACCEPTED
549	03:15.6	GET	https://www	500	-2147024883	222	704	0	10.210.100.2	ACCEPTED
550	03:25.9	GET	https://www	500	-2147024883	222	781	15	10.210.100.2	ACCEPTED
551	03:25.9	GET	https://www	500	-2147024883	222	781	31	10.210.100.2	ACCEPTED
552	03:30.2	GET	https://www	500	-2147024883	222	696	15	10.210.100.2	ACCEPTED
553	03:30.2	GET	https://www	500	-2147024883	222	696	15	10.210.100.2	ACCEPTED
554	03:40.0	GET	https://www	500	-2147024883	222	608	0	10.152.21.8	ACCEPTED
555	03:53.8	GET	https://www	500	-2147024883	222	727	15	10.210.100.2	ACCEPTED
556	03:53.8	GET	https://www	500	-2147024883	222	727	15	10.152.21.8	ACCEPTED
557	04:11.6	GET	https://www	500	-2147024883	222	957	31	10.152.21.8	ACCEPTED
558	04:11.6	GET	https://www	500	-2147024883	222	957	0	10.152.21.8	ACCEPTED
559	04:39.5	GET	https://www	500	-2147023667	332250	950	18734	10.210.100.2	REJECTED
560	04:52.4	GET	https://www	500	-2147024883	222	775	15	10.152.21.8	ACCEPTED
561	04:57.1	GET	https://www	500	-2147024883	222	787	15	10.210.100.2	ACCEPTED
562	05:00.9	GET	https://www	500	-2147024883	222	691	15	10.210.100.2	ACCEPTED
563	05:00.9	GET	https://www	500	-2147024883	222	691	15	10.210.100.2	ACCEPTED
564	05:23.5	GET	https://www	500	-2147024883	222	1011	31	10.152.21.8	ACCEPTED
565	05:24.4	GET	https://www	500	-2147024883	222	1011	31	10.152.21.8	ACCEPTED
566	05:30.6	GET	https://www	500	-2147024883	222	787	31	10.210.100.2	ACCEPTED
567	06:02.1	GET	https://www	500	-2147024883	222	671	78	10.152.21.8	ACCEPTED
568	06:02.2	GET	https://www	500	-2147024883	222	671	0	10.152.21.8	ACCEPTED
569	07:13.6	GET	https://www	500	-2147024883	222	676	15	10.152.21.8	ACCEPTED

Figura 10. Información post implementación del algoritmo T-digest.

El algoritmo detecta cual es el umbral para fijar que valor se consideraría anormal (anomalía en la información), sumado a que se debe determinar si la condicional se fija con solo un campo o todos los campos. Eso es algo que se determina por observación en el tipo de datos de cada uno de los campos seleccionados, si son del mismo tipo se consideraría un valor booleano de conjunción “AND” lo que significa que deben ser todos los campos en la evaluación, pero si son de diferente tipo se utilizaría un valor booleano de conjunción “OR” que solo busca que se cumpla un campo en la evaluación.

Como se puede ver en el registro 559 de la figura 10, se presenta un pico en el campo SC_bytes y en el campo Time_taken, esos picos superan los límites establecidos en el percentil 98, por lo tanto el resultado es un rechazo de registro. Se está estableciendo en el sistema, que si se supera el límite del campo SC_bytes o (or) del campo Time_taken, el registro marcado como anomalía, si es en ambos casos de igual forma se rechaza el registro. La consideración or se toma dado a que los campos SC_bytes y Time_taken poseen diferente tipo de datos, uno corresponde a cantidad de bytes transmitidos (SC_bytes) y el otro al tiempo que permaneció activa la conexión (Time_taken), si ambos campos hubieran sido del mismo tipo se optaría por la consideración and.

En la tabla 5 se puede ver el total de los valores introducidos para evaluar una vez terminado el algoritmo, así como un conteo de datos que fueron superiores a los percentiles definidos (95%-99%). Donde cont8_9X significa la cantidad de eventos que se presentaron en el Time_taken para el percentil indicado, los cuales pueden ser desde 95% hasta 99%.

Tabla 5. Total de registros de muestra y definición de totales de límite de percentiles.

Final Lenght	229039

Percentil	**Eventos**
cont8_95	46380
cont8_96	29816
cont8_97	16909
cont8_98	**5548**
cont8_99	3372

Capítulo 5. Resultados de la intervención

Se realizaron dos experimentos con información obtenida de los logs extraídos de los servidores de aplicaciones del INEGI, proporcionados por el área encargada de la administración de los servidores de aplicaciones, donde se buscó entrenar el sistema y poder determinar valores anormales de acuerdo con el algoritmo de aprendizaje automático de detección de anomalías, T-digest. Con lo que se detectó el umbral correcto para determinar anomalías de la información introducida.

Dichos experimentos utilizaron diferentes valores de entrada y diferente cantidad de información, además de encontrarse dentro de diferentes periodos de tiempo.

5.1 Caso de estudio uno

5.1.1 Componentes de la plataforma del experimento

Infraestructura:

Laboratorio implementado dentro de la Universidad Autónoma de Aguascalientes, las características se describen a continuación.

Máquina virtual

- ✓ 6 vCPUs
- ✓ Memory 16GB
- ✓ Hard disk 1 160 GB
- ✓ Hard disk 2 150 GB
- ✓ Ubuntu server 18

Almacenamiento de información:

- ✓ NoSQL – Elasticsearch
- ✓ Almacenamiento local

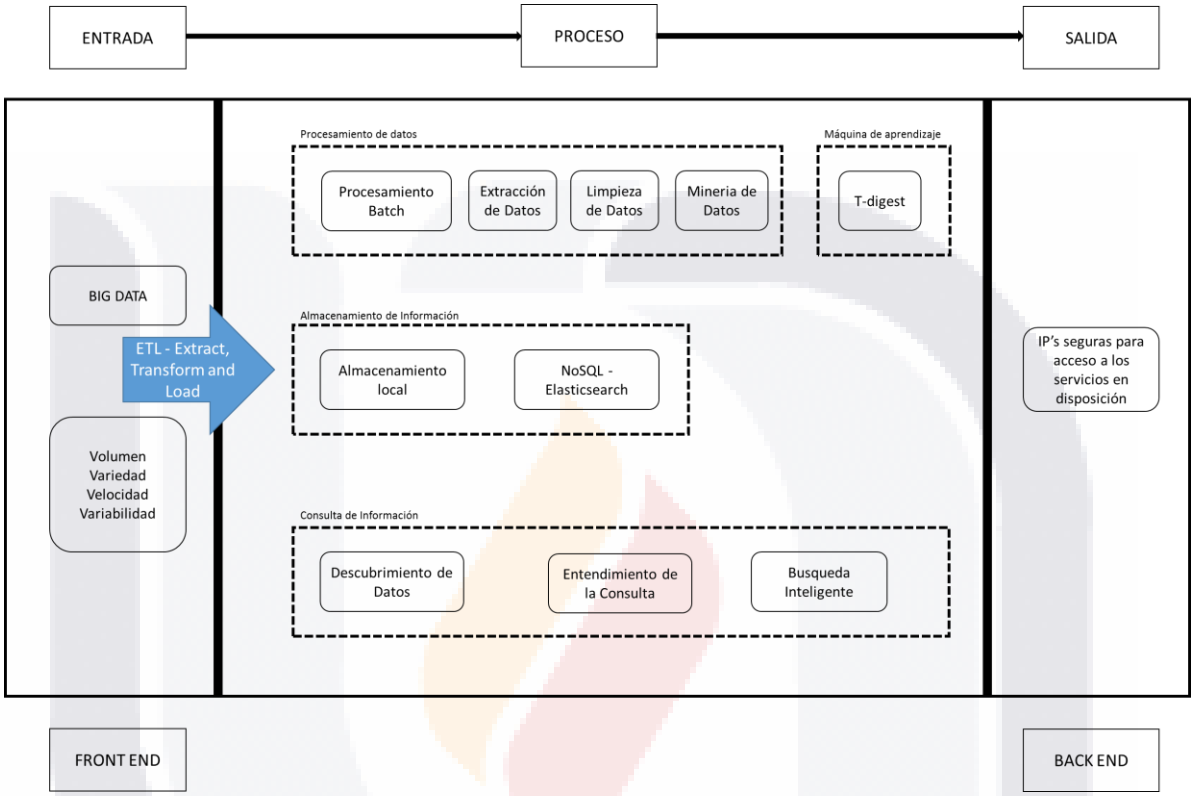

Figura 11. Diagrama descriptivo del caso práctico.

5.1.2 Descripción técnica del caso de uso (experimento)

En el primer caso se procesaron 67692 registros de los cuales correspondían a los meses julio y agosto del año 2019, con el fin de alimentar el algoritmo de aprendizaje automático.

Estos registros se sometieron al proceso de limpieza de datos con el fin de poder eliminar información no necesaria para el algoritmo. Una vez detallado los datos necesarios, se agruparon en los campos definidos anteriormente (Date Time, Method, Server, Status, Status_win32, SC_bytes, CS_bytes, Time_taken, NS-IPClient).

Con la información lista en cada uno de los campos, se sometió al algoritmo definido en Python, el cual arrojo los límites necesarios para cada uno de los percentiles a usar en el análisis.

Las tablas 6, 7 y 8, muestran los límites generados por cada uno de los percentiles del 95%-99% para los campos SC_bytes, CS_bytes y Time_taken:

Tabla 6. Límite calculado por percentil del campo bytes transferidos Servidor-Cliente (SC_bytes)

Field SC_bytes	
Percentage	Limits
95%	1748.4643313953472
96%	2460.1616300940427
97%	2627.6483529554566
98%	**4397.32837026648**
99%	4614.8560903316875

Tabla 7. Límite calculado por percentil del campo bytes transferidos Cliente-Servidor (CS_bytes)

Field CS_bytes	
Percentage	Limits
95%	1704.2274904807557
96%	1737.0725992103135
97%	1778.598772023296
98%	1851.7484977027664
99%	**1970.4102486986703**

Tabla 8. Límite calculado por percentil del campo de tiempo que duro la conexión (Time_taken)

Field Time_taken	
Percentage	Limits
95%	30.17992938620314
96%	31.814115742614323
97%	61.98478937598172
98%	**267.2683383030674**
99%	21207.895508021724

En la tabla 7 vemos los valores obtenidos correspondientes a campo CS_bytes. El campo CS_bytes no nos proporciona límites a considerar en el proceso de limpieza debido a que entre cada uno de los percentiles (95-99) no existe un salto considerable que pudiera considerarse de comportamiento anómalo. Por tal motivo se descarta el campo CS_bytes lo cual reducirá la cantidad de falsas alarmas.

El campo Time_taken posee un salto mayor en el percentil 99 sin embargo el primer salto mayor es el del percentil 98, considerando que las alertas por un percentil 99 pueden

ser consideradas dentro del percentil 98 se selecciona el percentil 98 como límite umbral en dicho campo.

Considerando solo los campos SC_bytes y Time_taken como límites para el algoritmo y tomando el percentil 98 se procede a una evaluación con diferentes registros generados. A continuación se muestra un extracto del archivo final que se genera una vez que el algoritmo es implementado, por cuestiones graficas el ultimo campo refleja si el registrado es aceptado o rechazado de acuerdo con el algoritmo T-digest.

555	03:53.8	GET	https://www	500	-2147024883	222	727	15	10.210.100.2	ACCEPTED
556	03:53.8	GET	https://www	500	-2147024883	222	727	15	10.152.21.8	ACCEPTED
557	04:11.6	GET	https://www	500	-2147024883	222	957	31	10.152.21.8	ACCEPTED
558	04:11.6	GET	https://www	500	-2147024883	222	957	0	10.152.21.8	ACCEPTED
559	04:39.5	GET	https://www	500	-2147023667	332250	950	18734	10.210.100.2	REJECTED
560	04:52.4	GET	https://www	500	-2147024883	222	775	15	10.152.21.8	ACCEPTED
561	04:57.1	GET	https://www	500	-2147024883	222	787	15	10.210.100.2	ACCEPTED
562	05:00.9	GET	https://www	500	-2147024883	222	691	15	10.210.100.2	ACCEPTED

Figura 12. Muestra del archivo final generado por el algoritmo.

Como se puede ver en la figura 12, en el registro 559, se presenta un pico en el campo SC_bytes y en el campo Time_taken esos picos superan los límites establecidos en el percentil 98, por lo tanto el resultado es un rechazo de registro.

5.2 Caso de estudio dos

5.1.1 Componentes de la plataforma del experimento

Infraestructura:

Laboratorio implementado dentro de la Universidad Autónoma de Aguascalientes.

Máquina virtual

- ✓ 6 vCPUs
- ✓ Memory 16GB
- ✓ Hard disk 1 160 GB
- ✓ Hard disk 2 150 GB
- ✓ Ubuntu server 18

Almacenamiento de información:

- ✓ NoSQL – Elasticsearch
- ✓ Almacenamiento local

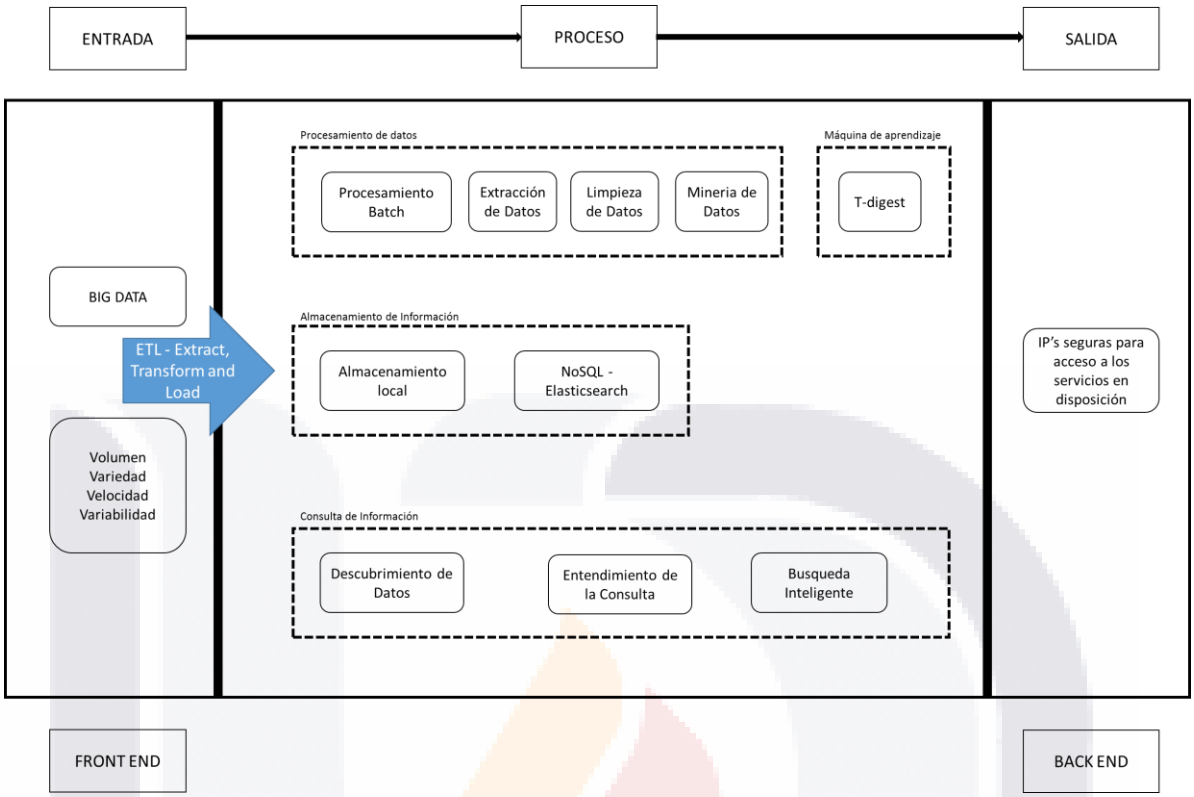

Figura 13. Diagrama descriptivo del caso práctico.

5.1.2 Descripción técnica del caso de uso (experimento)

En el segundo caso se procesaron 82416715 registros de los cuales correspondían a los meses enero y febrero del año 2020, con el fin de alimentar el algoritmo de aprendizaje automático.

Estos registros se sometieron al proceso de limpieza de datos con el fin de poder eliminar información no necesaria para el algoritmo. Una vez detallado los datos necesarios, se agruparon en los campos definidos anteriormente (Date Time, Method, Server, Status, Status_win32, SC_bytes, CS_bytes, Time_taken, NS-IPClient).

Con la información lista en cada uno de los campos, se sometió al algoritmo definido en Python, el cual arrojo los límites necesarios para cada uno de los percentiles a usar en el análisis.

Las tablas 9, 10 y 11, muestran los límites generados por cada uno de los percentiles del 95%-99% para los campos SC_bytes, CS_bytes y Time_taken:

Tabla 9. Límite calculado por percentil del campo bytes transferidos Servidor-Cliente (SC_bytes)

Field SC_bytes	
Percentage	Limits
95%	46250.24886275025
96%	59195.3121778115
97%	71434.29774107791
98%	**129842.60378960375**
99%	185212.34017595067

Tabla 10. Límite calculado por percentil del campo bytes transferidos Cliente-Servidor (CS_bytes)

Field CS_bytes	
Percentage	Limits
95%	1092.9373670295458
96%	1121.0679943688763
97%	1158.8972444202864
98%	1224.4618150553954
99%	**1351.517868925034**

Tabla 11. Límite calculado por percentil del campo de tiempo que duro la conexión (Time_taken)

Field Time_taken	
Percentage	Limits
95%	452.46656619864194
96%	542.0252157573887
97%	720.2336036273365
98%	**1079.5563676536065**
99%	2105.6521038427763

En las tablas 9, 10 y 11 se muestran los campos: bytes transferidos Servidor-Cliente (SC_bytes), bytes transferidos Cliente-Servidor (CS_bytes) y de tiempo que duró la conexión (Time_taken), respectivamente, con su percentil y el límite establecido en cada uno. El campo CS_bytes no nos proporciona límites a considerar en el proceso de limpieza debido a que entre cada uno de los percentiles (95-99) no existe un salto considerable que pudiera considerarse de comportamiento anómalo. Por tal motivo se descarta el campo CS_bytes lo cual reducirá la cantidad de falsas alarmas.

El campo Time_taken posee un salto mayor en el percentil 99 sin embargo el primer salto mayor es el del percentil 98, considerando que las alertas por un percentil 99 pueden ser consideradas dentro del percentil 98 se selecciona el percentil 98 en dicho campo.

Este proceso llevo 4.658 horas, debido a la cantidad de datos que se tuvieron que procesar al leer los archivos de logs, limpieza de datos, procesamiento en un archivo limpio y la alimentación del algoritmo T-digest. Esto se considera normal dado a que el algoritmo no se fue alimentando al día de la creación sino hasta el final de los dos meses lo cual acumuló una gran cantidad de datos.

5.3 Análisis y notas de los resultados

1. Debido a que el volumen de información a procesar es extremadamente grande, y a las limitantes de la infraestructura donde se realizaron los experimentos, se realizó la implantación del algoritmo T-digest solo con una fracción de esta. Sin embargo, bajo las condiciones antes descritas, se permitió un tiempo aceptable de procesamiento aunque se encontró que los tiempos varían significativamente de acuerdo con el volumen de datos a procesar.
2. Originalmente se consideró trabajar con datos normalizados previo a la utilización del algoritmo T-digest, pero una vez que la información se sometió a un análisis, se observó que en los campos existían demasiados valores ignorados por el algoritmo, existiendo aumentos en los datos y los cuales no seguían un patrón específico. Por tal motivo, se compararon este tipo de datos contra datos sin normalizar, dando como resultado que estos últimos eran lo mejor para detectar un patrón en los picos de acuerdo al algoritmo.
3. Ambos experimentos dan como resultado un comportamiento similar durante la implantación del algoritmo T-digest en los percentiles (95-99) así como en el límite máximo. De la misma forma, en ambos experimentos se descartó el campo CS_bytes dado a que el algoritmo no generaba detección de anomalías con los valores de entrada, ya que dichos valores mantenían un comportamiento estable lo que imposibilitaba determinar el percentil ideal para activar la alarma de registros rechazados en base a este campo.
4. Todo los componentes desarrollados en este proyecto se pueden descargar del siguiente repositorio:
   https://github.com/luisenrique1212/caso_practico_LEOR.git

Capítulo 6. Conclusiones y trabajos futuros

A continuación se presentan las conclusiones del presente caso práctico de acuerdo con los resultados obtenidos:

1. La implantación del algoritmo T-digest muestra ser una buena técnica de aprendizaje automático para la detección de posibles accesos que representen amenazas en los servidores del INEGI. Sin embargo, es necesario la implantación del trabajo realizado en el presente caso práctico en un ambiente de producción para poder validar completamente los trabajos aquí realizados. Desafortunadamente, debido a los eventos a causa de la pandemia del virus COVID-19 a nivel mundial, no fue posible realizar dicha etapa, por lo cual, se recomienda ampliamente la continuación de este trabajo utilizando herramientas para el procesamiento de grandes cantidades de datos como las que posee el instituto, para la prueba de su funcionamiento en tiempo real.
2. Se logró la reducción del tiempo que se emplea en validar las diferentes hipótesis para validar posibles amenazas de ataques en instituto. Cada uno de los registros son validados con el modelo y marcados aquellos que se presentan como registros anómalos. Así pues, al poder clasificarlos se elimina la necesidad de analizar de forma tradicional, todos los accesos buscando eventos peligrosos, reduciendo substancialmente el tiempo necesario para esto.
3. El análisis rápido de conjuntos de datos complejos, creando múltiples vistas de análisis sobre los datos que son de interés por medio de la implantación del algoritmo T-digest, sirve a los equipos de trabajo, conformado por el Departamento De Administración De Servicios En Web dentro de la Coordinación General De Informática, como una posible herramienta de apoyo de gran utilidad para el análisis de posibles amenazas informáticas a sus servidores.
4. De acuerdo a los resultados obtenidos, se piensa que es posible eliminar o minimizar la necesidad de un soporte externo o extra al ya existente en el instituto, para el establecimiento de la seguridad en los servidores de los cuales se analizó la información, donde el modelo arroja valores concisos y verificables durante su análisis sobre datos específicos como la transmisión de datos y tiempos de conexión.
5. Por otro lado, creemos que la implantación de los trabajos realizados en el presente caso práctico, podría minimizar el tiempo de respuesta en el análisis en tiempo real y en la toma de decisiones del instituto frente a un ataque. Esto debido a que el modelo detecta si el acceso es anormal, y de acuerdo con los experimentos realizados que permitieron validar el modelo, confirmar que esos accesos anormales correspondían a accesos inválidos.
6. Se consiguió ampliar el mapa de revisión de amenazas cibernéticas ya conocido y definido por la organización, dado a que la identificación de direcciones IPs y DNS maliciosos genera un mapeo geoespacial. El mapeo geoespacial es una técnica que permite aplicar métodos analíticos, con el fin de poder detectar amenazas en puntos geográficos específicos.
7. El desarrollo de los trabajos presentados en el presente documento, a través de la implantación del algoritmo de aprendizaje automático T-digest, permitió descubrir accesos inválidos de usuarios internos, a través del análisis de conjuntos masivos

de datos. Esto se realizó a través del análisis de direcciones que accedieron a los servidores las cuales, al presentar accesos inválidos, se sometieron a su respectiva validación con respecto al algoritmo planteado reflejando el mismo comportamiento que presentaban los accesos externos inválidos.

Glosario

Aprendizaje automático. Es una variedad de paradigmas de aprendizaje, algoritmos, resultados teóricos y aplicaciones.

Ciberseguridad. Es el conjunto de tecnologías y procesos diseñados para proteger computadoras, redes, programas y datos de ataques, accesos no autorizados, cambios o destrucción.

Dato en “crudo”. Son los datos que no se han sometido a un procesamiento completo de limpia, ya sea manualmente o mediante un software informático automatizado, por lo que no poseen un formato o alguna estructura en especial. También se les conoce como datos de origen, datos primarios o datos atómicos.

DNS. Sistema de Nombres de Dominio (DNS por sus siglas en inglés). El cual es un sistema de bases de datos distribuidas en la red que cumplen la función de traducir una solicitud de ciertos nombres de host a direcciones de IP específicos, los cuales pueden ser entendidos por cualquier equipo.

GSLB. Servidor de balanceo de carga global (GSLB por sus siglas en inglés) es un mecanismo de equilibrio de carga sobre el protocolo de DNS, el cual es rápido y confiable ya que utiliza el protocolo UDP y la respuesta que da al cliente es prácticamente en tiempo real.

IP. Protocolo de Internet (IP por sus siglas en inglés), el cual es un código que sirve para identificar al usuario dentro de la red.

Método GET. Es un método de envío de datos a través de Internet donde el cliente o usuario recibe información de en un servidor, archivo o base de datos y la puede visualizar.

Método POST. Es un método de envío de datos a través de Internet donde el cliente o usuario se encarga de enviar información para ser procesada o actualizada en un servidor, archivo o base de datos.

Normalización. Es el proceso sistemático de la descomposición de los datos para eliminar la redundancia en la información y las características no deseadas que pueden ser generadas en el momento de insertar, actualizar y eliminar registros. Es un proceso de varios pasos que coloca los datos en forma tabular, eliminando los datos duplicados de las tablas de relaciones

W3C. Es una comunidad internacional donde las organizaciones, personal y el público en general trabajan conjuntamente para desarrollar estándares utilizados en internet.

Bibliografía

Buczak, A. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEE Communications Survey's & Tutorials. Vol 18, No 2. p1153

Buczak, A. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEE Communications Survey's & Tutorials. Vol 18, No 2. p1154

Cairo, A. (2017). Data visualization: An image can be worth more than a thousand numbers, but not always more than a thousand words. El Profesional de la Información. Vol. 26 Issue 6. p1026

Chang, W. (2015). NIST Big Data Interoperability Framework: Definitions. NIST Special Publication. Volume 1, 1500-1, p 4

Dunning, T. (2014), Practical Machine Learning A New Look at Anomaly Detection, O'Reilly Media, Inc, First Edition, p8-11

Dunning, T. (2014), Practical Machine Learning A New Look at Anomaly Detection, O'Reilly Media, Inc, First Edition, p14-24

Dunning, T. (2018), Computing Extremely Accurate Quantiles Using t-Digests, p18-21.

Elasticsearch (2020). Elastic. Recuperado de https://www.elastic.co/what-is/elasticsearch

Elasticsearch (2020). Kibana. Recuperado de https://www.elastic.co/what-is/kibana

Gama, J. (2005). Machine Learning. Portugal: University of Porto, André C.P.I.F. de Carvalho.

Gartner. (2015). NIST Big Data Interoperability Framework: Definitions. NIST Special Publication. Volume 1, 1500-1. p iii

Goodfellow, I.; McDaniel, P. and Papernot, N. (2018). Making Machine Learning Robust Against Adversarial Inputs. Communications of the ACM, Vol 61 Issue 7, p56.

Goodfellow, I.; McDaniel, P. and Papernot, N. (2018). Making Machine Learning Robust Against Adversarial Inputs. Communications of the ACM, Vol 61 Issue 7, p60.

IBM Corporation. (2017). Accelerate the data-to-decision process by rapidly transforming data into actionable insight. IBM i2 Enterprise Insight Analysis. p1

IBM Corporation. (2017). Counter and mitigate more attacks with cyber threat hunting. IBM i2 Enterprise Insight Analysis for Cyber Threat Hunting. p1

ISO/IEC. (2015). Big Data Preliminary Report 2014. ISO/IEC JTC 1 Information Technology. p5

Kantardzic, M. (2011). Data Mining: Concepts, Models, Methods, and Algorithms, Institute of Electrical and Electronics Engineers. Second Edition. p449

Kantardzic, M. (2011). Data Mining: Concepts, Models, Methods, and Algorithms, Institute of Electrical and Electronics Engineers. Second Edition. p30-34

Manu, N. (2018). Analysis of Machine Learning Methodologies in Big Data Applications. International Journal of Recent Research Aspects. Vol. 5 Issue 1. p3

Marsland S. (2009). Machine Learning An Algorithm Perspective. Boca Raton, FL USA: CRC Press Taylor & Francis Group, A Chapman & HALL BOOK.

Miller, K. (18 de Agosto, 2019). ETL Database. Your central database for all things ETL: advice, suggestions, and best practices. Recuperado de https://www.stitchdata.com/etldatabase/etl-process/

Mitchell T. M., (1997). Machine Learning. Singapore: MCFraw-Hill International Editions, Computer Science Series.

Ochoa, M. and Sancho, V. (2014). An approach to taxonomy of data visualization. Revista Latina de Comunicación Social. Issue 69. p489

STANDARDS (2019). W3C Leading the web to its full potential. Recuperado de https://www.w3.org/standards/

Schmidt, Michael S., (2014), 5 IN CHINA ARMY FACE U. S. CHARGES OF CYBERATTACKS, New York Times, Vol. 163 (Issue 56507), pA1-A8