ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

EVALUACIÓN DE ALGORITMOS DE MINERÍA DE DATOS PARA DETECCIÓN Y PREDICCIÓN DE ATAQUES DE INYECCIÓN SQL EN BIG DATA

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SOFTWARE

ANDRÉS MAURICIO LLUMIQUINGA GUAMBA

DIRECTORA: PhD. GABRIELA LORENA SUNTAXI OÑA

DMQ, octubre 2022

CERTIFICACIONES

Yo, Andrés Mauricio Llumiquinga Guamba , declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

ANDRES MAURICIO LLUMIQUINGA GUAMBA

Certifico que el presente trabajo de integración curricular fue desarrollado por Andrés Mauricio Llumiquinga Guamba, bajo mi supervisión.

PhD. Gabriela Lorena Suntaxi Oña
DIRECTORA DE PROYECTO

Certificamos que revisamos el presente trabajo de integración curricular.

Nombre1 Nombre2 Apellido1 Apellido2
REVISOR 1 DEL TRABAJO
DE INTEGRACIÓN CURRICULAR

Nombre1 Nombre2 Apellido1 Apellido2
REVISOR 1 DEL TRABAJO
DE INTEGRACIÓN CURRICULAR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

ANDRÉS MAURICIO LLUMIQUINGA GUAMBA

DRA. GABRIELA LORENA SUNTAXI OÑA

BRYAN ANDRÉS PALMA PONCE

EDISON JAVIER QUIMBIAMBA GUASGUA

STEVEN JAVIER RIVERA TENELANDA

DEDICATORIA

Dedico con gran felicidad y amor mi tesis a mis padres, mi hijo y a mi abuelita, ya que, sin su compañía y su amor incondicional no lo hubiera logrado. Sus plegarias para no desviarme del camino correcto, sus grandes consejos me sabrán guiar y proteger de todo mal que me enfrente bien sea en mi vida profesional o como persona. Por estos grandes motivos les doy mi trabajo como ofrenda por su crianza, su amor y su paciencia, les amo.

Continuaré perseverante esforzándome por mis metas, para seguir siendo su orgullo en todo lado y anhelo que dios me dé el tiempo necesario para poder recompensar el hombre que me hicieron y el gran profesional que seguiré siendo.

AGRADECIMIENTOS

A Dios y a la virgen del Cisne porque jamás me han dejado solo y han sido muy buenos conmigo y por las grandes bendiciones que he podido recibir de su parte.

A mi padre Antonio, gracias por inculcarme valores de respeto, humildad, también por consentirme, y forjarme como una gran persona.

A mi madre Viviana, a quien no solo la amo con todo mi corazón si no que la admiro demasiado, gracias por todos los cursos que me diste los cuales fueron muy importantes para conseguir mi primer trabajo, además por jamás dudar de mis habilidades, al contrario, siempre me apoyaste y me apoyas en cada una de mis decisiones. Gracias también por ser una gran abuelita y estar para mi hijo incondicionalmente. Jamás cambiara la mujer que dios me dio como mamá.

A mi abuelita Rosita por comprarme mi computadora la cual me ayudo a culminar mis estudios con éxito y fue mi primera herramienta para realizar todos mis deberes, proyectos, etc.... También le agradezco por siempre tenerme en sus oraciones y de su gran crianza que me supo dar.

A mis hermanos Antony y Aarón porque siempre cuando estaba triste ellos sabían cómo alegrar mis días y ver las cosas de diferentes maneras.

A la Dra. Gabriela Suntaxi directora del proyecto integrador, quien supo no solo ser nuestra tutora si no una amiga, sin contar con la gran paciencia que nos guio para culminar con éxito nuestro proyecto.

A una persona muy especial en mi vida, Karla Tipanta gracias por tu apoyo, confianza y por haberme dado un hijo hermoso,sano e inteligente.

CONTENIDO

RESUMEN

Las aplicaciones web han crecido exponencialmente en los últimos 15 años, dado que las aplicaciones web hacen las vidas más convenientes debido a sus grandes capacidades para cubrir los requerimeintos de los usuarios, sin dejar de lado sus grandes capacidades de almacenamiento de datos. Sin embargo, su gran accesibilidad da como resultado posibles vulnerabilidades. Por ejemplo, vulnerabilidades en las funciones de entradas de datos por los usuarios. Los ataques más comunes en explotar estas vulnerabilidades según OWASP Top Ten son: inyección SQL (SQLIAs) y secuencias de comandos entre sitios (XSS). Por lo tanto, cualquier sistema u organización empresarial de cualquier tipo son usualmente atacadas por parte de sus usuarios o terceros con el objetivo de acceder a información con alto nivel de confidencialidad.

El Centro de Respuesta a Incidentes de Seguridad Informática de la Escuela Politécnica Nacional (CSIRT-EPN) existe la necesidad de detectar diferentes vulnerabilidades en sus servidores de bases de datos, para prevenir ataques de inyección de código SQL (SQLIAs) de terceros.

Para evitar y descubrir ataques de inyección SQL se puede implementar analizadores de código para esas entradas realizadas. Sin embargo, esta técnica es muy limitada frente a la capacidad del ataque. Por lo tanto, este proyecto permite la detección y prevención de ataques de inyección SQL a través del análisis de registros. Además su implementación fue realizada con la metodología CRISP-DM de minería de datos, debido a que es la más común en el área de minería de datos, por su flexibilidad y su enfoque en la detección y visualización de datos para encontrar patrones sospechosos. El algoritmo implementado para la predicción fue el SVM (Máquinas de vectores de soporte) utilizando como principal fuente de información el registro de transacciones del servidor de la CSIRT-EPN.

Para las interfaces de visualización de datos se realizó gráficos cuantitativos mediante diagramas de barras u otros, los cuales permiten visualizar mejor si el sistema ha estado bajo ataque.

PALABRAS CLAVE: SVM, CRISP-DM, Patrones, OWASP Top Ten, Inyección SQL

ABSTRACT

Web applications have grown exponentially in the last 15 years since they make lives more convenient due to their outstanding capabilities to meet users'requirements without neglecting their significant data storage capabilities. However, its great accessibility results in possible vulnerabilities. For example, vulnerabilities in user input functions. The most common attacks to exploit these vulnerabilities, according to OWASP Top Ten, are: SQL injection (SQLIAs) and cross-site scripting (XSS). Therefore, any system or business organization of any kind is usually attacked by its users or third parties to access information with high confidentiality.

The Computer Security Incident Response Center of the National Polytechnic School (CSIRT-EPN) needs to detect different vulnerabilities in its database servers, to prevent third-party SQL injection attacks (SQLIAs).

Code analyzers can be implemented for entries made by users to prevent and discover SQL injection attacks. However, this technique is minimal against the ability of the attack. Therefore, this project allows the detection and prevention of SQL injection attacks through log analysis. In addition, its implementation was carried out with the CRISP-DM data mining methodology since it is the most common in the data mining area due to its flexibility and focus on detecting and data to find suspicious patterns. The algorithm implemented for the prediction was the SVM (Support Vector Machines) ,using the transaction log of the CSIRT-EPN server as the primary information.

For the data visualization interfaces, quantitative graphs were made using bar diagrams or others, which allow better visualization of whether the system has been under attack.

KEYWORDS: SVM, CRISP-DM, Patterns, OWASP Top Ten, Cross Site Scripting, SQL Injection.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El OWASP Top 10 2021 contribuye bastante a entender la importacia de que el software tenga altos niveles de seguridad frente a los diferentes tipos de ataques. El ataque más común y conocido por la mayoria de los desarrolladores es el ataque de inyección SQLIA. OWASP lo etiquetó como A03:2021. Según la última publicación los ataques de inyección descienden hasta la tercera posición. Cabe recalcar que el 94 % de las aplicaciones fueron analizadas para la detección de algún tipo de inyección, dando como resultado una tasa de incidencia máxima del 19 % y de 3.37 % de promedio [1].

Los desarroladores implementan diferentes acciones para evitar los ataques de SQLIAs, pero no son lo suficientemente capaces de prevenir en su gran parte o totalidad dichos ataques. Por lo tanto, en este componente se realizará el estudio de un algoritmo de minería de datos que permita detectar y predecir ataques de inyección SQL. El algoritmo de minería de datos que se analizará en este trabajo es SVM (SUPPORT VECTOR MACHINES), el cual fue previamente seleccionado por criterios como: su enfoque al aprendizaje supervisado, su porcentaje de precisión al momento de identificar y predecir datos etiquetados como los ataques de SQLIAs en Big Data. A continuacón, se desarrollará un modelo para poder identificar y predecir ataques inyección SQL a partir del algoritmo SVM. La evaluación del modelo se realizará a través de métricas como la precisión, falsos positivos, falsos negativos. Estas métricas serán reflejadas a traves de valores númericos o gráficos.

Posteriormente se procederá a desarrollar un prototipo web en el cual se utilizará el modelo previamente creado, para evaluar y documentar los resultados obtenidos en un informe final de un caso de estudio real. Finalmente, para obtener un funcionamiento óptimo del sistema, se procederá a crear una guía de ayuda en la cual se detallará como implementar correctamente el prototipo en el servidor destinado para el caso de estudio.

1.1 OBJETIVO GENERAL

Generar un prototipo web, el cual posibilite la identifiación de ataques por inyección de código SQL implementando técnicas de extracción de datos y analizar los resultados obtenidos con datos de prueba reales de un caso de estudio, con la finalidad de estimar la efectividad del algoritmo de minería de datos SVM.

1.2 OBJETIVOS ESPECÍFICOS

- ❒ Evaluar la efectividad de la detección y predicción de los ataques de inyección SQL obtenidos por el algoritmo SVM.
- ❒ Generar un prototipo de sistema, el cual posibilite la identifiación y predicción de ataques a sistemas de bases de datos manejados por el CSIRT-EPN.
- ❒ Generar gráficos para mejorar la visualización de los resultados cuantitativos obtenidos.

1.3 ALCANCE

El presente componente primero busca estudiar y evaluar el algoritmo de minería de datos SVM en ataques de inyección SQL a través de CRISP-DM, la cual es una metodología conocida de extracción de datos o minería de datos. Segundo la implementación de un prototipo de software que permita detectar y predecir los ataques de inyección SQL en Big Data utilizando SVM mediante la metodología de Programación Extrema (XP). Tercero la evaluación del prototipo para un caso de estudio específico con datos reales. Cuarto la creación de una guía de implementación del prototipo en el servidor destinado para el caso de estudio.

2 MARCO TEÓRICO

En esta primera parte del proyecto se introduce a la parte teórica de la seguridad de la información, para resaltar su conceptos fundamentales. Segundo, se intrujo un concepto importante, como el BIG DATA. En el cual se repasará la teoría que son necesarios para el proyecto. Tercero se abarcara los ataques SQLIAs y sus diferentes tipos de ataques. Cuarto, detalla el desarrollo de la revisión sistemática implementada de la busqueda de diferentes investigaciones sobre la detección y predicción de ataques SQLIAs. Quinto, se presenta CRIPS-DM la metodología de extracción de datos aplicada a este proyecto. Es decir, se enumeran y resumen cada una de sus fases. Finalmente, se identificará varias herramientas que se pueden usar para ejecutar el proyecto.

2.1 SEGURIDAD DE LA INFORMACIÓN

En los últimos años, la importancia de la seguridad de la información ha aumentado debido a la necesidad de proteger los activos de información [2]. Es decir, los activos son considerados como todo objeto que tiene valor para la organización. Por tanto, se entiende por seguridad de la información (SI) el mantenimiento de la confidencialidad, integridad y disponibilidad de la información [2]. Esta consideración se basa en las tres categorías definidas en el estándar de gestión de seguridad de la información ISO 27001, pero en el área de la informática se le conoce como la Triada CIA (Confidentiality, Integrity, Availability), las cuales constan de:

- *Confidencialidad:* Al referirnos a la confidencialidad hacemos mención de las propiedades de la información, las cuales no deben ser puestas a disposición por personas o entidades no autorizadas.
- *Disponibilidad:* La disponibilidad hace mención a la propiedad de ser accesible por personas o entidades autorizadas.

- ❒ *Integridad:* La integridad se enfoca en que la información se conserve inalterada ante cualquier acontecimiento malicioso o accidental.

2.2 BIG DATA

Big data se refiere a la gestión especializada del procesamiento de grandes cantidades de datos, también conocida como ciencia de datos. Su método consiste en una serie de herramientas computacionales y estadísticas para sintetizar y analizar los datos recopilados gracias al gran volumen de información que se genera cuando los individuos interactúan a través de plataformas virtuales, dispositivos electrónicos y teléfonos móviles. La importancia del big data se basa en leer y analizar los comportamientos y relaciones de los diversos stakeholders para agilisar la toma de decisiones en las organizaciones y empresas, permitiéndoles desarrollar estrategias y acciones dirigidas a ellos.

2.2.1 Características

Diversos autores han determinado una serie de características en común para que un sistema de bases de datos, así como los datos que almacena, sean considerados dentro de un esquema de Big Data [3]. Estas características en el área de la informática son conocidas como las 3V's del Big Data las cuales son:

a. Volumen

Para los datos principales, tenemos que lidiar con grandes cantidades de datos de baja densidad sin estructura. Puede tener datos desconocidos con valores de diferentes fuentes. Para la mayoria de organizaciones, esto significa un gran tamaño de información por lo cual les dificulta en su manipulación.

b. Velocidad

La velocidad es la medición del tiempo en la cual los datos son recibidos y procesados. En la actualidad los productos inteligentes tienen la necesidad de que sus datos fluyan en tiempo real de manera rápida, además requieren evaluación y acción en el mismo.

c. Variedad

La variedad hace énfasis a sus diferentes tpos de datos, dado que comunmente se

trabaja con tipos de datos tradicionales, los cuales están acopalados a bases de datos relacionales.

Debio al aumento de datos a nivel global, están disponibles nuevos tipos de datos, más conocidos como no estructurales. Estos datos son texto, audio y vídeo, los cuales, necesitan un paso adicional para ser procedados y deducir el significado y la compatibilidad con los metadatos.

Debido a la gran cantidad de los datos que se manipula en la actualidad, un aspecto clave dentro de la gestión de dichos datos es la seguridad que rodea a los motores de base de datos utilizados. Por esta razón, es necesario identificar los principales ataques y amenazas que existen con respecto a las bases de datos

2.3 ATAQUES DE INYECCIÓN SQL

La gestión de la información es un tema muy relevante en la actualidad debido a su alto estatus dentro de la empresa. Como tal, hay muchas aplicaciones web y de escritorio que se ejecutan sobre bases de datos relacionales, generalmente gobernadas por un lenguaje de consulta estructurado conocido como SQL. Sin embargo, el uso de este lenguaje puede hacer que su aplicación sea vulnerable a entradas de consultas maliciosas conocidas como ataques de inyección SQL.

Los ataques de inyección SQL se enfocan al proceso de insertar sentencias de consulta SQL dentro de una consulta aceptada por el sistema, con el fin de manipular los procesos lícitos de la aplicación.

Al utilizar las vulnerabilidades de inyección de SQL, los atacantes tienen la posibilidad de enviar parámetros maliciosos a la base de datos de back-end. Es decir, la declaración SQL original cambiará y se podrá realizar ataques maliciosos, por ejemplo las acciones más comunes al entrar a la aplicación es: leer, modificar, eliminar información confidencial en la base de datos de back-end [1].

2.3.1 Tipos de ataques de inyección SQL

a. Ataque basado en tautologías

Descripción: Este tipo de ataque se utiliza para evitar la autenticación al sistema, dado que ignora la clausula **WHERE**. El ejemplo de tautología más conocido por la gente es la expresión "1=1". La lógica de este ataque se enfoca en colocar cualquier condición y concatenarla con los criterios OR y "1=1, ya que siempre va hacer verdadero, entonces toda la consulta será verdadera [1].

Ejemplo:

```
SELECT * FROM ACCOUNTS WHERE login='' OR 1=1 -- AND pass=''
```

b. **Consultas de unión**

Descripción: Este ataque se encarga de transformar la lógica y, por lo tanto, permite que la estructura de la consulta original cambie el conjunto de resultados. El ataque comienza al encontrar una entrada vulnerable en la cual se adjunta una segunda consulta con el conector “unión”, con esta acción se puede obtener resultados de diferentes tablas, es decir nos da como respuesta la unión de los datos de la primera consulta y de la segunda consulta.

Ejemplo:

```
SELECT * FROM TblSupplier WHERE NameSupplier = ' ' UNION ALL
SELECT * From TblConsumer WHERE 1 =1
```

c. **Consultas adicionales Descripción:** Este tipo de ataque no cambia la lógica de la consulta original. Por el contrario, la base de datos recibe múltiples consultas SQL porque se agrega una nueva consulta que es completamente diferente de la primera consulta.

Ejemplo:

```
SELECT * FROM TblUser WHERE UserName = ''; DROP TABLE TblUser AND pass =''
```

d. **Procedimientos almacenados Descripción:** Este ataque se basa en inyectar código SQL malicioso en la base de datos a través de procedimientos almacenados

existentes. Tenga en cuenta que los procedimientos almacenados tienen la capacidad de interactuar con el sistema operativo. Por lo tanto, es importante que un atacante pueda interactuar con estos procedimientos almacenados para tomar el control o bloquear el servidor.

Ejemplo:

```
SELECT * FROM news WHERE news_cat = 'sport'; SHUTDOWN;
```

e. Codificaciones alternativas

Descripción: Este ataque tiene como objetivo confundir la base de datos con diferentes codificaciones en las sentencias SQL. Un ejemplo muy común es cuando los atacantes usan códigos hexadecimal, ASCII, etc. en una declaración SQL. Al realizar esta poderosa técnica da como resultado el evitar la validación básica realizada por la aplicación [1].

Ejemplo:

```
SELECT usr_id FROM gestion.ges_usuario WHERE usr_nombre= 'legalUser'; exec(
    CHAR(0x73687574646f776e)) -- AND usr_password=''
```

2.3.2 Técnicas de detección de ataques de inyección SQL

Una técnica común es crear una lista de todos los campos de entrada cuyos valores se pueden usar para construir la consulta que genera el error, para prevenir inserciones de sentencias maliciosas de SQL. La primera prueba generalmente se basa en agregar una comilla simple (') o un punto y coma (;) sobre el campo a probar. El primero se usa como cadena de terminación en SQL. Esta es la pregunta incorrecta ya que no está filtrada por la aplicación. El segundo se utiliza para finalizar la instrucción SQL. Si no lo hace, es muy probable que se produzcan errores.

2.3.3 Técnicas de prevención de ataques de inyección SQL

Para la prevención de los ataques de inyección SQL, los desarrolladores pueden implementar diferentes estrategias como:

- ❒ Usar instrucciones predefinidas, consultas de parámetros o procedimientos almacenados para asegurar que los diferentes campos de entrada de los usuarios no se consideren requisitos genuinos.
- ❒ Limitar los privilegios de los administradores en base de datos, en particular para las cuentas de solicitud.

2.4 MINERÍA DE DATOS

En la actualidad la minería de datos es un proceso muy común por sus efectivos resultados cuando son ejecutados de forma correcta. Comúnmente es aplicado cuando se desea detectar estructuras interesantes como patrones, secuencias, etc. La minería de datos esta enfocada en dos aspectos importantes, las cuales son: A gran escala y A pequeña escala.

La minería de datos es la extracción automática de información general utilizando datos y algoritmos para simular relaciones ocultas en grandes cantidades de datos. Se espera que la información obtenida también pueda hacer predicciones y facilitar el análisis de datos.

2.4.1 Técnica de minería de datos

Las técnicas de extracción de datos se enfoca en generar modelos predecibles, los cuales ayuden a responder preguntas sobre datos futuros.

En cambio existe un modelo descriptivo, el cual se encarga de proveer información sobre las diferentes de los datos y sus caracteristicas. Hay varias técnicas en minería de datos que se pueden categorizar [4]:

a. Clasificación de datos

Esta es una técnica basada en la construcción de un modelo a partir de etiquetas y categorías conocidas. El modelo está entrenado para identificar características a partir de datos conocidos. El modelo generado se utilizará con nuevos datos sin clasificar.

Al ser evaluado por el modelo, determina la clase a la que pertenece cada dato. Se muestran los esquemas de los algoritmos de árbol de decisión y SVM (máquina de vectores de soporte) en la Figura 2.1.

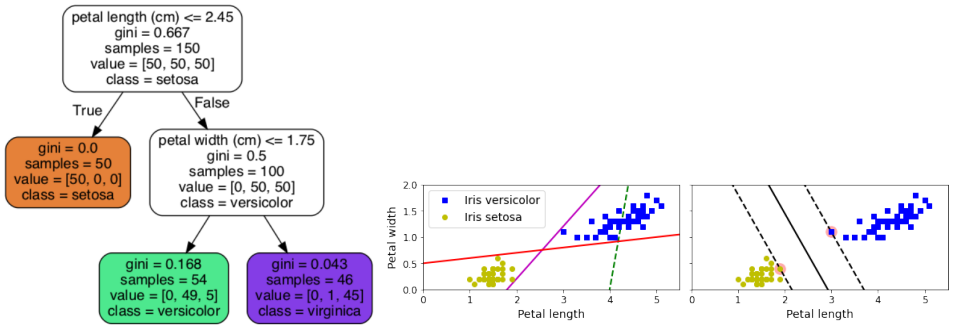


Figura 2.1: Los algoritmos de árboles de decisión (izq.) y el algoritmo SVM (der.) para la clasificación de flores basándose en las características de sus pétalos [5]

b. Predicción de datos

Los datos que faltan en el conjunto se pueden esbozar utilizando esta técnica. Esta técnica es diferente de la clasificación. El propósito de este último es asignar etiquetas a varias clases de datos para que los datos mostrados puedan asociarse con cada una de esas clases. La predicción se centra en la probabilidad de que ciertos patrones, características o valores ocurran en los datos que no están presentes en el conjunto de datos original.

c. Agrupación de datos

Esta técnica se basa en dividir los datos en grupos que comparten características similares. Esta es una técnica independiente en la que las etiquetas de clase no están predefinidas y el algoritmo utilizado las determina automáticamente en función de la similitud entre los datos.

d. Análisis de anomalías

Esta técnica intenta determinar cuáles de estos datos se comportan de manera anormal en comparación con otros datos dentro de un conjunto de datos, según su comportamiento y propiedades. La detección de anomalías es un proceso complejo debido al nivel de ruido latente en los datos o la falta de comprensión empresarial que puede dificultar la identificación precisa de diferentes valores en los datos.

e. Reglas de asociación

Esta técnica le permite encontrar relaciones entre diferentes datos. Utilizado principalmente en aspectos comerciales donde se estudia el análisis de identificación del modelo de cliente.

2.4.2 Proceso de minería de datos

La minería de datos comienza con identificar datos. Para hacer esto, necesita saber qué datos desea, dónde encontrarlos y cómo obtenerlos. Una vez que los datos están disponibles, deben ingresarse a la base de datos en el formato adecuado y procesarse. Esta es considerada una de las tareas más difíciles en la minería de datos. Una vez que los datos están en el formato correcto, debe seleccionar los datos que desea y eliminar los datos que no necesita. La minería de datos podría verse más como un proceso que como un conjunto de herramientas [6]. El proceso en cuestión consta de los siguientes pasos:

a. Dominio de negocio

Se debe identificar y comprender los campos de requisitos e identificar el propósito para el cual solicita acceso a los requisitos de minería de datos.

b. Obtención de datos

Seleccione un registro, anotando las variables requeridas para realizar una búsqueda de conocimiento.

c. Limpieza y preprocesamiento de datos

Identifique pautas para minimizar el ruido en datos seleccionados. Este ruido puede ser causado por cosas como la copia innecesaria de datos, etc..

d. Reducción de datos y proyección

Encuentre una función que pueda mostrar correctamente los datos en el contexto. Con la reducción de tamaño se reduce la complejidad requerida para la información relevante.

e. Análisis exploratorio

Seleccione métodos y algoritmos de extracción de datos para encontrar plantillas o información relacionada en sus datos. Este paso determina las condiciones y parámetros de los datos analizados.

f. Interpretación

Explique el patrón utilizado. Este paso aplica un método iterativo de minería de datos. Esto se debe a que si los resultados obtenidos no cumplen con las metas de rendimiento esperadas, es necesario volver al paso anterior.

g. Actuar sobre el conocimiento descubierto

Este nuevo conocimiento se puede utilizar para negociar en otros sistemas o simplemente para registrar e informar.

2.4.3 Herramientas de minería de datos

Las herramientas de extracción de datos utilizadas en el proceso de extracción de conocimiento se puede clasificar en:

- ❒ Técnicas de verificación en donde se limita a comprobar hipótesis dada por el usuario.
- ❒ Método de descubrimiento en donde se encuentra patrones de forma automática, en el cual se incluyen todas las técnica de predicción.

En el campo de la minería de datos, existen varias herramientas de uso común que facilitan el desarrollo de minería de datos como:

- ❒ Yale- Facilita la visualización de los datos de dispersión en 2D y 3D, estas representaciones de datos se muestran en formato de coordenadas paralelas.
- ❒ Weka- Esta herramienta es una de las herramientas más completas en el proceso de minería de datos ya que puede realizar reprocesamiento, agrupamiento, clasificación, regresión, visualización y selección de datos.
- ❒ Ramses- Es la herramienta más utilizada por las agencias gubernamentales ya que permite una gestión automatizada y confiable de los riesgos asociados al comercio internacional.

3 REVISIÓN SISTEMÁTICA DE LA LITERATURA DE TÉCNICAS DE DETECCIÓN Y PREDICCIÓN DE SQLIA

3.1 METODOLOGÍA

Para la realización de esta investigación, se aplicó la metodología propuesta por Kitchenham en [7]. La metodología va dirigida diferentes pasos, conel fin de realizar una revisión sistemática buena. Estos pasos se describen a continuación:

a. Preguntas de investigación

Las preguntas de investigación se formularon con la finalidad de determinar las técnicas que existen en la actualidad para la detección y predicción de ataques de inyección SQL.

De esta forma se obtuvó las siguientes preguntas de investigación:

RQ1 ¿Cuáles son las técnicas que se están utilizando para la detección y predicción de SQLIAs?

RQ2 ¿Cuáles son las técnicas mas utilizadas para detección y predicción de SQLIAs?

RQ3 ¿Es posible clasificar las técnicas para la detección y predicción de SQLIA?

Mediante las preguntas RQ1 y RQ2 se llevó a cabo un análisis de publicaciones que plantean nuevas técnicas para la detección y predicción de SQLIAs.

Para contestar la pregunta RQ3 se analizó los resultados obtenidos en las preguntas anteriores para sentar una clasificación de las técnicas identificadas.

b. Proceso de búsqueda

Para efectuar la búsqueda se utilizó la siguiente cadena de búsqueda en base a las preguntas de investigación planteadas:

(detection OR prediction) AND (SQLIA OR (SQL AND injection)) AND NOT (survey OR review)

Esta cadena fue acoplada de manera que cumpliera con las especificaciones de búsqueda de cada librería o base de datos utilizada. Sin embargo, el esquema general de los términos y conectores utilizados se mantuvo fijo en cada búsqueda.

c. Fuentes y bases de datos para la búsqueda

La búsqueda fue efectuada en las bases de datos más nombradas dentro del área de la Ciencia de la Computación. Las librerías digitales utilizadas fueron:

- ACM Digital Library
- IEEE Xplore
- ScienceDirect
- SpringerLink

Después se efectuó una búsqueda mas exhaustiva en el motor de búsqueda bibliográfica de Google Scholar. Esta búsqueda se la realizó con el fin de obtener artículos que no hayan sido publicados en las librerías digitales consideradas inicialmente.

d. Criterios de inclusión y exclusión

Para la inclusión de un artículo se tomaron en cuenta los siguientes criterios:

- Artículos publicados entre el 1 de enero de 2012 y el 13 de abril de 2022
- Artículos cuyos títulos cumplieran la cadena de búsqueda considerada para la búsqueda
- Artículos publicados en conferencias o revistas especializadas

Una vez determinados los artículos que cumplieron con los criterios de inclusión, se empleo los siguientes criterios de exclusión para la selección de artículos:

- Artículos que traten sobre la detección de ataques de inyección SQL en tecnologías o entornos específicos.
- Artículos que traten sobre la detección de otros ataques además de los ataques de inyección SQL

- ❒ Artículos que no propongan de técnicas específicas para la detección de ataques de inyección SQL. Por ejemplo, revisiones sistemáticas de la literatura o artículos científicos que presenten comparativas entre técnicas existentes
- ❒ Artículos que no tengan DOI
- ❒ Artículos que tengan menos 5 páginas de contenido sin tomar en cuenta la sección de referencias bibliográficas
- ❒ Artículos escritos en un idiomas distinto al inglés

La aplicación de los criterios de exclusión se la realizó de manera manual realizando un escaneo sobre los artículos obtenidos

e. Selección de artículos

La selección de artículos se la realizó en 6 fases de búsqueda en las cuales se fueron aplicando los distintos criterios de inclusión y exclusión hasta llegar a los artículos que fueron seleccionados para formar parte de la revisión sistemática.

En la primera fase de búsqueda, se seleccionaron todos los resultados que incluyeran la cadena de búsqueda en cualquier lugar del artículo, ya sea en el título, resumen, contenido, metadatos, etc. En esta primera búsqueda se obtuvieron un total de 4048 resultados, desglosados de la siguiente manera:

Tabla 3.1: Tabla de números de artículos seleccionados .

Artículos	Total
Science Direct	213
SpringerLink	973
IEEE Xplore	274
ACM Digital Library	1478
Google Scholar	1110

En la segunda fase de búsqueda se filtraron solo los resultados comprendidos entre el 1 de enero de 2012 y el 13 de abril de 2022. En esta búsqueda, los resultados disminuyeron a 2957 resultados, desglosados de la siguiente manera:

Tabla 3.2: Tabla de números de artículos filtrados por fecha.

Artículos	Total
Science Direct	403
SpringerLink	745

Artículos	Total
IEEE Xplore	212
ACM Digital Library	831
Google Scholar	766

En la tercera fase de búsqueda se seleccionaron los resultados en donde la cadena de búsqueda se encontrase solo en el título, para validar que los papers seleccionados hagan referencia al tema a investigar. Así, se redujo el resultado de la búsqueda a 233 resultados, desglosados de la siguiente manera:

Tabla 3.3: Tabla de números de artículos filtrados por título.

Artículos	Total
Science Direct	8
SpringerLink	12
IEEE Xplore	53
ACM Digital Library	28
Google Scholar	132

En la cuarta fase de búsqueda se descartaron los resultados que no fuesen propiamente artículos científicos, por ejemplo, aquellos artículos que hayan sido publicados en revistas indexadas o conferencias especializadas. En esta fase se obtuvieron 198 artículos desglosados de la siguiente manera:

Tabla 3.4: Tabla de números de artículos filtrados por tipos.

Artículos	Total
Science Direct	8
SpringerLink	12
IEEE Xplore	53
ACM Digital Library	28
Google Scholar	97

En la quinta fase de búsqueda, se procedió a filtrar los artículos duplicados. En este filtrado se logró obtener un total de 156 artículos.

En la sexta fase de búsqueda, fueron tomados los 156 artículos obtenidos hasta la quinta fase de búsqueda y se realizó un análisis manual de cada artículo, aplicando

los criterios de exclusión definidos anteriormente. De esta búsqueda se obtuvieron finalmente 51 artículos que fueron seleccionados para la revisión.

En la Fig. 3.1 se muestra de manera resumida el proceso de filtrado de los artículos mediante las diversas búsquedas en las que se fueron aplicando los criterios de inclusión y exclusión especificados anteriormente.

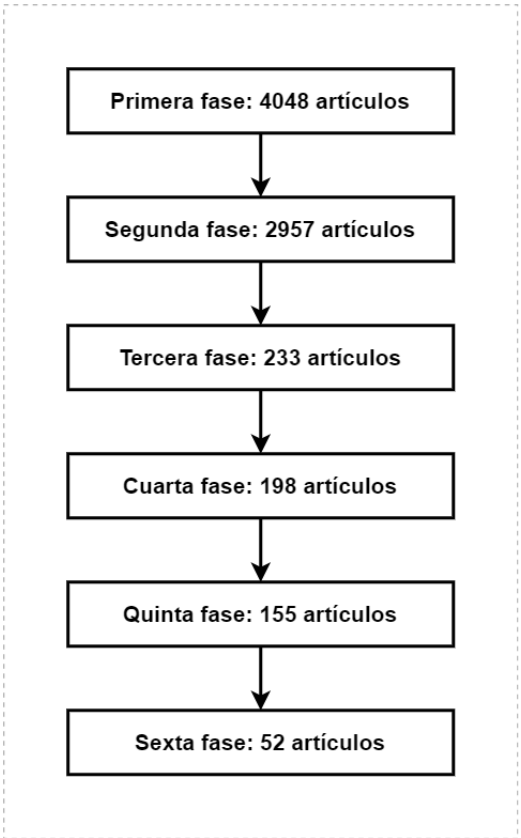


Figura 3.1: Resumen del proceso de búsqueda y selección de artículos. Fuente: Elaborada por el autor

f. Evaluación de calidad Para la evaluación de calidad se utilizaron los criterios propuestos en [8]. Así, se obtuvieron las siguientes preguntas y criterios de puntuación para realizar la evaluación de calidad:

QA1 **¿El artículo describe los objetivos de investigación de manera clara?**

- ❒ La pregunta puntúa 1 si se indica en el abstract o en la introducción de manera explícita la técnica a desarrollar en el artículo.
- ❒ La pregunta puntúa 0.5 si se indica en el abstract o en la introducción de manera implícita la técnica a desarrollar en el artículo.
- ❒ La pregunta puntúa 0 si no se indica en el abstract o en la introducción de manera la técnica a desarrollar en el artículo.

QA2 **¿El artículo describe una revisión de literatura, antecedentes y contexto de investigación?**

- ❒ La pregunta puntúa 1 si se describen al menos dos temas diferentes que den contexto de la investigación en el artículo.
- ❒ La pregunta puntúa 0.5 si se describe un solo tema que dé contexto de la investigación en el artículo.
- ❒ La pregunta puntúa 0 si no se describe ningún tema que dé contexto de la investigación en el artículo.

QA3 **¿El artículo muestra trabajos relacionados de trabajos anteriores para mostrar la principal contribución de la investigación?**

- ❒ La pregunta puntúa 1 si existe una sección de trabajos relacionados en el artículo o se describen de manera detallada algunos trabajos relacionados.
- ❒ La pregunta puntúa 0.5 si existe un indicio o breve referencia a trabajos relacionados en el artículo pero no se los describe de manera detallada.
- ❒ La pregunta puntúa 0 si no existe ningún indicio o mención a trabajos relacionados dentro del artículo.

QA4 **¿El artículo describe la arquitectura propuesta o la metodología usada?**

- ❒ La pregunta puntúa 1 si existe una descripción clara, completa y detallada de la arquitectura o metodología propuesta en el artículo.
- ❒ La pregunta puntúa 0.5 si existe una descripción inconsistente, incompleta o ambigua de la arquitectura o metodología propuesta en el artículo.
- ❒ La pregunta puntúa 0 si no existe una descripción de la arquitectura o metodología propuesta en el artículo.

QA5 **¿El articulo tiene resultados de la investigación?**

- ❒ La pregunta puntúa 1 si se evalúa de manera detallada la metodología o técnica propuesta en el artículo y se muestran los resultados de dicha evaluación.
- ❒ La pregunta puntúa 0.5 si se solo se evalúa la metodología o técnica propuesta en el artículo sin mostrar los resultados o solo se muestran los resultados de la metodología o técnica propuesta sin mostrar la evaluación.
- ❒ La pregunta puntúa 0 si no se muestran ni la evaluación ni los resultados de la metodología o técnica propuesta en el artículo.

QA6 **¿El artículo muestra conclusiones que son relevantes al propósito/problema de investigación?**

- ❒ La pregunta puntúa 1 si la conclusión de la investigación muestra concordancia con los objetivos propuestos en el artículo.
- ❒ La pregunta puntúa 0.5 si la conclusión de la investigación difiere un poco de los objetivos propuestos en el artículo.
- ❒ La pregunta puntúa 0 si la conclusión de la investigación difiere completamente de los objetivos propuestos en el artículo o si no existen conclusiones en el artículo.

QA7 **¿El articulo recomienda trabajo o mejoras a realizar para el futuro?**

- ❒ La pregunta puntúa 1 si los trabajos futuros se detallan claramente y se relacionan directamente con la investigación principal del artículo.
- ❒ La pregunta puntúa 0.5 si los trabajos futuros no se detallan claramente o no se relacionan directamente con la investigación principal del artículo.
- ❒ La pregunta puntúa 0 si no se indican trabajos futuros en el artículo.

g. Extracción de datos y resultados

Los datos extraídos de cada estudio fueron:

- ❒ Información bibliográfica (título, año de publicación, conferencia o revista, autores)
- ❒ Número de citas en el Google Scholar
- ❒ Nombre o descripción de la técnica utilizada para la detección de SQLIA
- ❒ Clasificación a la que pertenece la técnica utilizada para la detección de SQLIA
- ❒ Algoritmo utilizado para la detección de SQLIA (si aplica)
- ❒ Tamaño y fuente del conjunto de datos utilizado para la aplicación de la técnica (si aplica)
- ❒ Tipos de SQLIA que abarca la técnica descrita

3.1.1 Resultados

a. Resultados de búsqueda

Los 51 artículos seleccionados se describen en la Tabla 8.1. Por cada artículo se muestran: las citas obtenidas en Google Scholar, el año de publicación, el nombre de la técnica utilizada para la detección o predicción de SQLIA, la clasificación a la que pertenece la técnica utilizada el o los algoritmos utilizados (si aplica), el tamaño y fuente del conjunto de datos utilizado para la evaluación de la técnica propuesta (si aplica), los tipos de SQLIA que abarca la técnica descrita. Los artículos fueron ordenados por el número de citas que obtuvieron en Google Scholar. Para los artículos que no cumplan alguno de los criterios se colocarán las iniciales "N/Aïndicando que no aplica el criterio en dicho artículo.

En base a la Figura 3.2 se observa que la técnica de Machine Learning es utilizada en 19 artículos para la detección y predicción de SQLIA, y las técnicas menos usadas son: Sistema de detección de intrusión, técnicas híbridas y otras técnicas. En base a la lectura de los artículos se determinó una clasificación para las técnicas de detección y predicción de SQLIA. Esta clasificación se puede apreciar en la quinta columna de la Tabla 8.1 y se explica a mayor profundidad en el apartado **c.** de la Sección 3.1.2 .

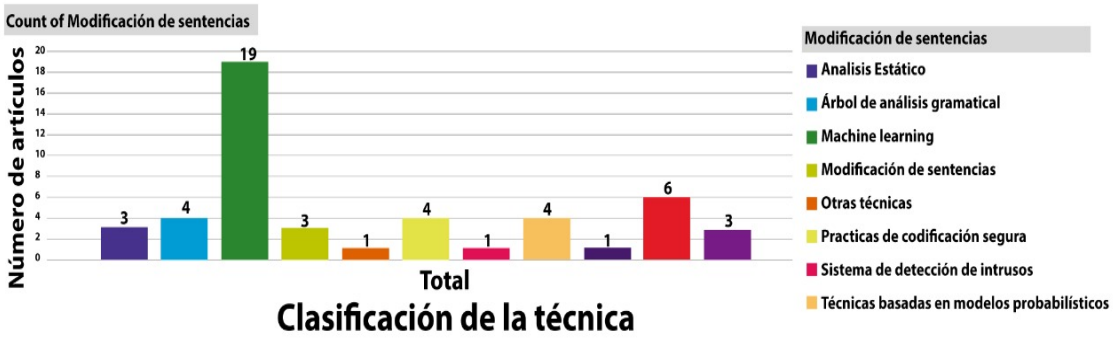


Figura 3.2: Distribución de clasificación de las técnicas de detección de SQLIA en los artículos analizados. Fuente: Elaborado por el autor

b. Resultados de la evaluación de calidad

En la Figura. 3.3, se muestran los puntajes de la evaluación de calidad de los artículos analizados

Para esta revisión se determinó que todos los artículo con una calificación mayor a 5 puntos de 7 posibles, son considerados como artículos de alta calidad.

Como se puede observar en la Tabla 3.3, el promedio de la evaluación de calidad es aproximadamente de 5.75, por lo que se puede determinar que de manera general, los artículos poseen una buena calidad.

En la Fig. 8.2 se observa que a pesar de mantener una calidad relativamente alta, existen 4 artículos que muestran una baja calificación. Por contra parte, 12 artículos alcanzaron una calificación perfecta, lo que resulta en un buen indicador en cuanto a la calidad de los estudios realizados para proponer nuevas técnicas para la detección y predicción de SQLIAs.

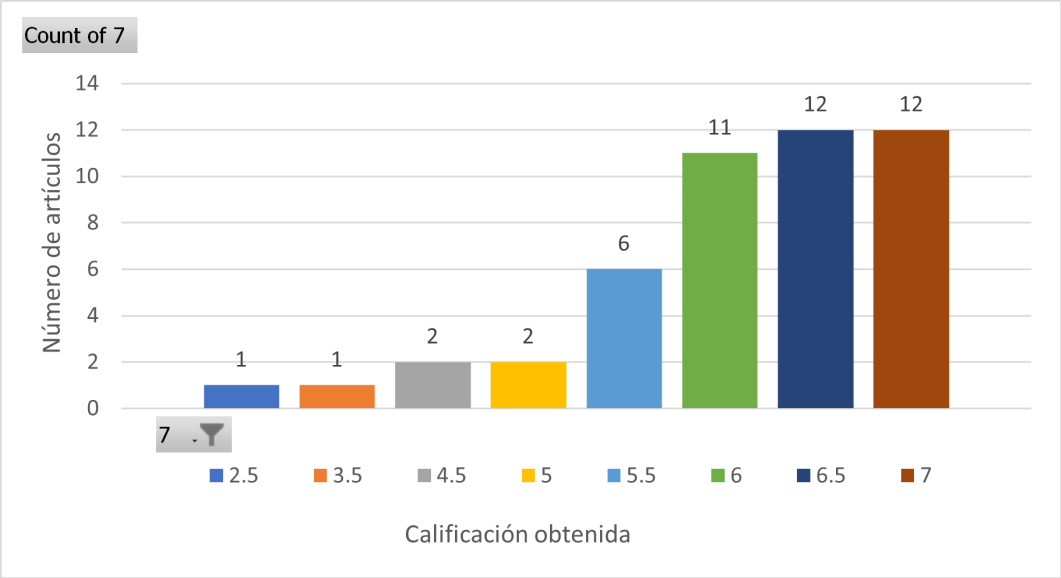


Figura 3.3: Distribución de calificaciones de la evaluación de calidad de los artículos analizados. Fuente: Elaborado por el autor

3.1.2 Discusión de las preguntas de investigación

En esta sección se discuten las respuestas a las preguntas de investigación descritas en el apartado **a.** de la Sección 3.1

a. ***¿Cuáles son las técnicas mas utilizadas para detección y predicción de SQLIAs***

De acuerdo con la investigación realizada, las técnicas con mayor impacto resultan ser aquellas que hacen usos de algoritmos de machine learning. Aproximadamente el 38 % de los artículos analizados utilizan algoritmos de machine learning.

b. ***¿Cuáles son las técnicas que se están utilizando para la detección y predicción de SQLIAs***

Para responder esta pregunta, se puede observar la Tabla 8.1, donde se resumen las técnicas que han sido propuestas para la detección y predicción de SQLIAs en los últimos 10 años.

c. *¿Es posible clasificar las técnicas para la detección y predicción de SQLIAs*

Como se observa en la Tabla 8.1 se realizó una clasificación de las técnicas para la detección y predicción de SQLIAs. En este estudio se determinó la siguiente clasificación:

- ❒ **Análisis estático:** los enfoques estáticos detectan o contrarrestan la posibilidad de un ataque de inyección SQL en la fase de compilación. Este enfoque se centra en escanear la aplicación y aprovechar el análisis del flujo de información para detectar los códigos que podrían tener vulnerabilidades[9].
- ❒ **Modificación de sentencias:**esta técnica se centra en reconstruir las consultas en tiempo de ejecución utilizando una clave criptográfica que es inaccesible para los atacantes. Esta técnica permite a los desarrolladores crear consultas SQL utilizando palabras clave aleatorias en lugar de normales, donde un proxy entre la aplicación web y la base de datos intercepta las sentencias SQL y desaleatoriza las palabras clave[9].
- ❒ **Árbol de análisis gramatical:** esta técnica comprueba en tiempo de ejecución si las consultas entrantes se ajustan a un modelo de consulta esperado. El modelo se decide en tiempo de ejecución donde examina las estructuras de la consulta antes y después de las peticiones del cliente, es decir, se encarga de asegurar las sentencias SQL vulnerables comparándolo con un árbol de análisis sintáctico de una sentencia con el de la original y únicamente permitirá que se ejecute una sentencia con una comparación coincidente[9].
- ❒ **Técnicas Taint-based:** esta técnica aplica varias políticas de seguridad marcando los datos no fiables y rastreando sus flujos a través de los programas mediante un análisis sensible y minucioso al contexto para rechazar las consultas SQL si estas tienen una entrada no fiable[9].
- ❒ **Técnicas basadas en modelos probabilísticos:** las técnicas basadas en modelos probabilísticos se los realiza en tiempo de ejecución, donde se asume que el valor de una sentencia SQL está relacionada con la presencia o ausencia de vulnerabilidades en su estructura y de esta manera permitir la detección de un ataque de inyección SQL[10].
- ❒ **Sistemas de detección de intrusos:** los sistemas de detección de intrusos se basan en una técnica de aprendizaje automático que se entrena utilizando un

conjunto de consultas típicas en aplicaciones web. La técnica empieza construyendo modelos de las consultas típicas y luego las supervisa las consultas que ingresan a la aplicación en tiempo de ejecución para identificar las consultas que no coinciden con el modelo construido[10].

- **Técnicas híbridas:** algunas técnicas combinan un análisis estático durante el desarrollo con la combinación de una supervisión dinámica en tiempo de ejecución[9], como tal es el caso de AMNESIA [11], que asocia un modelo de consulta con la ubicación de cada consulta en la aplicación y luego monitoriza la aplicación para detectar si alguna consulta se desvía del modelo esperado[10].
- **Prácticas de Codificación Segura:** las principales vulnerabilidades de inyección SQL se deben a la insuficiente validación de las entradas. Por lo tanto, la solución directa para eliminar estas vulnerabilidades es aplicar prácticas de codificación segura. Algunos ejemplos de las mejores prácticas son: comprobar el tipo de entrada en la consulta, codificación de las entradas, coincidencias positivas de patrones e identificar todas las fuentes de la entrada[10].

3.1.3 Conclusiones

A partir de esta revisión, se pudo realizar una clasificación de los diferentes tipos de técnicas de SQLIA, donde se encontró una cierta prevalencia en las técnicas que utilizan Machine Learning. La gran mayoría de los artículos que fueron evaluados demostraron tener un alto nivel de calidad en cuanto al contenido y al detalle mostrado sobre la técnica propuesta. Es importante notar el avance que ha existido en cuanto a investigaciones en el campo de las técnicas para la detección y predicción de SQLIA, ya que como se puede apreciar, existe mucha información en este ámbito. En un futuro es posible profundizar en otros aspectos como los SQLIA en entornos específicos u otros ataques similares como podrían ser los ataques Cross Site Scripting(XSS).

4 METODOLOGÍA

El proyecto se realizó utilizando la metodología Kitchenham [12] para la revisón sistemática de la literatura, CRISP-DM [13] para la mineria de datos y para la genearicón de un prototipo web se utilizó la metodología de Programación Extrema [14].

4.1 METODOLOGÍA DE REVISIÓN SISTEMÁTICA DE LA LITERATURA

Las metodología escogida para la revisión sistemática de la literatura fue las propuesta por el autor Kitchenham [12], ya que es una metodología que propone hacia el desarrollo de software. La estructura de esta metodología se divide en tres secciones generales: planificación, implementación y reporte. A continuación, se describen cada una de estas secciones y sus respectivas fases.

4.1.1 Planificación de la revisión

4.1.1.1 Identificación de la necesidad de una revisión

La descripción de esta metodología fue desarrollada como parte de un trabajo común entre compañeros.

En esta etapa es importante asegurarse de que se requiere una revisión sistemática. Como tal, los investigadores deben identificar y revisar las revisiones sistemáticas existentes de los fenómenos de interés en función de los criterios apropiados. Para verificar esto, se crea una lista de verificación que contiene preguntas para ayudar a identificar la necesidad de verificación.

4.1.1.2 Elaboración de un protocolo de revisión

Un protocolo de revisión es responsable de especificar los métodos utilizados para desarrollar revisiones sistemáticas. La importancia de los registros es reducir el potencial de sesgo del investigador. El componente del protocolo incluye todos los elementos de la revisión más información de programación adicional.

- ❒ Preguntas de investigación con las que se pretende responder con la revisión
- ❒ Estrategia de búsqueda para investigación primaria, incluidos términos de búsqueda, recursos para buscar, incluidas bases de datos, revistas científicas, conferencias.
- ❒ Criterios y procedimientos de selección de estudios. Los criterios de selección de estudios especifican los criterios para incluir o excluir un estudio de la revisión sistemática.
- ❒ Listas de comprobación y procedimientos de evaluación de la calidad de los estudios.
- ❒ Estrategia de extracción de datos.
- ❒ Síntesis de los datos extraídos.

4.1.2 Realización de la revisión

Cuando los investigadores han acordado el protocolo a seguir, se puede empezar la revisión, esto implica las siguientes fases:

4.1.2.1 Identificación de la investigación

El objetivo principal de una revisión sistemática es encontrar tantos estudios clave como sea posible que sean relevantes para la pregunta de investigación utilizando una estrategia de búsqueda imparcial. Por lo tanto, en comparación con las pruebas tradicionales, se toman algunos pasos adicionales:

a. Generar una estrategia de búsqueda

La estrategia de búsqueda suele ser iterativa y se beneficia de búsquedas preliminares para identificar revisiones sistemáticas existentes y evaluar el alcance de estudios

potencialmente relevantes. Se recomienda dividir la pregunta en pasos separados como: población, intervención, resultados, diseños de estudio. Luego se crea una lista de sinónimos y abreviaturas. A continuación, cree cadenas de búsqueda complejas utilizando combinaciones booleanas AND y OR.

b. Sesgo de publicación

El sesgo es un problema en las publicaciones que tienen más probabilidades de publicar resultados positivos que negativos. La definición de resultados positivos o negativos depende del investigador, quien los categoriza según su valoración valorativa. Por lo tanto, el investigador debe informarse sobre la problemática y explorar la literatura, conferencias o contactar con expertos e investigadores que el área de interés que puedan guiarlo en la investigación y no recaer en el sesgo de publicación.

c. Gestión de la bibliografía y recuperación de documentos

La gestión de la bibliografía permite gestionar un gran número de referencias que se pueden obtener de una investigación bibliografía exhaustiva. Por lo tanto, es importante tener un sistema para esto, por ejemplo, se pueden utilizar paquetes bibliográficos como Reference Manager o Endnote o simplemente un Excel con toda esta información.

d. Documentación de la búsqueda

Todo el proceso de preparación de una revisión sistemática debe ser transparente y rastreable, por lo que la revisión debe: documentarse con suficiente detalle para repetirse, documentarse a medida que se realiza y los cambios se deben anotar y justificar, los resultados de la búsqueda sin filtrar deben almacenarse y estar disponibles para su análisis para mantenerse en el futuro.

4.1.2.2 Selección de estudios primarios

Si hay estudios primarios potencialmente relevantes, el siguiente paso es evaluar su significado real.

a. Criterios para la selección de estudios

Los criterios de selección de los estudios tienen por objetivo el de identificar los estudios primarios que aportan pruebas directas a la consulta de investigación. Los criterios de inclusión y exclusión deben tomar como referencia la pregunta de investiga-

ción. Estos criterios deben probarse para asegurar que su interpretación es fiable y que los estudios están clasificados correctamente.

b. Proceso de selección de estudios

La selección de estudios es un proceso de varios pasos que comienza con los criterios de selección interpretados por el investigador, a menos que no se pueda excluir un estudio debido a que se recibieron copias incompletas. Por lo tanto, una vez obtenido el texto completo, mantenida la lista de estudios excluidos e identificadas las razones de la exclusión, se debe tomar una decisión final de inclusión/exclusión.

4.1.2.3 Evaluación de la calidad del estudio

Los criterios de inclusión y exclusión son importantes considerar en la evaluación de la calidad de los estudios primarios:

- ❒ Proporcione criterios de inclusión/exclusión aún más detallados.
- ❒ Como medio para considerar la importancia de los estudios individuales al momento de sintetizar los resultados.
- ❒ Interpretar los resultados y determinar la solidez de las conclusiones.
- ❒ Recomendaciones para guiar futuras investigaciones.

4.1.2.4 Extracción y seguimiento de datos

En esta etapa se diseñan formularios de extracción de datos que cumplen la tarea de capturar con precisión la información que los investigadores obtuvieron de la investigación primaria. Para minimizar el sesgo, las formas de extracción de datos deben definirse y probarse al definir un protocolo de estudio.

a. Diseño de formularios de extracción de datos

Los formularios de extracción de datos deben diseñarse para garantizar la recopilación de la información necesaria para responder las preguntas de revisión y los criterios de calidad del estudio. En la mayoría de los casos, la minería de datos se define como una serie de valores numéricos que se deben extraer para cada estudio. Una

recomendación importante es el uso de formularios electrónicos, ya que facilitan el análisis posterior.

b. Contenido de los formularios para la recolección de datos

Además de cualquier pregunta necesaria para responder la pregunta de la revisión sistemática y los criterios de evaluación de la calidad, los formularios de recopilación de datos deben contener información relevante, que incluye:

- ❒ Nombre de la ediciones
- ❒ Fecha de la obtención de los datos
- ❒ Designación, creador, revista, detalles de trabajos

c. Procedimientos para la extracción de datos

Es importante que el proceso de extracción de datos de la investigación primaria se realice de forma independiente por dos o más investigadores. Luego, los datos extraídos deben compararse y los desacuerdos deben resolverse por consenso entre los investigadores. Es recomendable utilizar un formulario aparte para anotar y corregir los errores o inconsistencias indicados.

d. Múltiples publicaciones de los mismos datos

Es importante recordar que múltiples publicaciones con los mismos datos no se incluyen en una revisión sistemática, ya que estos informes duplicados pueden influir seriamente en la investigación. En caso de publicación repetida, se recomienda utilizar la última publicación para una revisión periódica.

e. Datos no publicados, datos que faltan y datos que requieren manipulación

Si existiera el caso de que se dispone de información de estudios que se están desarrollando o están en curso, debe incluirse siempre y cuando sea posible información de calidad sobre el estudio. Los informes no siempre presentan todos los datos relevantes, también pueden estar mal redactados y ser ambiguos, por lo tanto, es necesario ponerse en contacto con los autores para obtener la información necesaria. En ciertas ocasiones los estudios primarios no proporcionan todos los datos, pero en algunas situaciones se pueden recrear estos datos necesarios a partir de la manipulación de los datos publicados. Por lo tanto, si se diera el caso de manipulación de datos, es importante someterlos a un análisis de sensibilidad para su posterior uso.

4.1.2.5 Síntesis de datos

La síntesis de datos consiste en reunir y resumir los resultados obtenidos de la investigación primaria. La síntesis de los resultados puede ser descriptiva (en lugar de cuantitativa). En algunos casos, también es posible completar la síntesis descriptiva con resúmenes cuantitativos. El uso de métodos que utilizan estadísticas para desarrollarlos se denomina metanálisis.

a. Síntesis descriptiva

La información obtenida de la investigación a saber hace enfásis de la intervención, la población, el contexto, el tamaño de la muestra, los resultados y la calidad del estudio deben tabularse adecuadamente, teniendo siempre en cuenta el tema de la revisión. Estas tablas deben tener una estructura que permita mostrar similaridades y diferencias entre los resultados de la investigación.

b. Sensibilidad del análisis

La realización de un análisis de sensibilidad es importante cuando se realiza un meta-analisis. El meta-analisis se utiliza para proporcionar una estimación global del efecto de tratamiento y su variabilidad en el estudio. En estos casos lo ideal es la repetición de varios subconjuntos de estudios primarios con el objetivo de determinar si estos resultados son robustos o no. Los tipos de subconjuntos serian:

- ❒ Solo estudios primarios de alta calidad
- ❒ Estudios primarios de tipos particulares
- ❒ Estudios primarios para los que la extracción de datos no presento dificultades

4.1.3 Presentación de informes

Esta fase es una de las más importantes porque permite compartir de manera efectiva los resultados de la revisión sistemática. Las revisiones sistemáticas generalmente se presentan en al menos dos formas:

- ❒ En un apartado de tesis doctoral
- ❒ En una conferencia

4.2 METODOLOGÍA DE MINERÍA DE DATOS CRISP-DM

CRISPDM es una metodología que describe las fases típicas de un proyecto y las tareas requeridas en cada fase [13]. CRISPDM puede verse como un modelo de minería de datos que ayuda a los expertos en el campo a resolver problemas. CRISPDM consta de seis capas, algunas de las cuales son bidireccionales. Cualquiera puede editar y editar de nuevo. Esto significa que los segmentos de pasos no están necesariamente colocados en la secuendia que se muestra en el diagrama. El modelo CRISP-DM es flexible y fácil de adaptar a las operaciones de su organización, permitiéndole crear modelos de minería de datos que satisfagan sus necesidades específicas.

A continuación se explica cada una de las fases:

1. **Comprensión del negocio**

   Esta primera fase es quizás el conjunto de tareas más importante para abarcar los objetivos y requisitos del proyecto desde un pensamiento enfocado al area comercial y traducirlos en objetivos técnicos y planes del proyecto. Sin incluir estos objetivos, ningún algoritmo, por complejo que sea, puede producir resultados fiables. Para extraer datos de manera más efectiva, debe tener una buena comprensión del problema que está tratando de resolver para poder recopilar los datos correctos e interpretar los resultados correctamente. En esta etapa, es muy importante traducir el conocimiento obtenido de la empresa en problemas de extracción de datos y desarrollar un plan de preparación para alcanzar los objetivos de la empresa. A continuación, se describen cada tarea que esta compuesta por esta fase:

   - ❒ **Determinar los objetivos de negocio**

     Esta es la primera tarea desarrollada y su propósito es identificar el problema que se necesita resolver, identificar por qué se debe utilizar la extracción de datos y definir criterios de éxito. Su problema puede ser diferente como detectar fraudes con tarjetas de crédito, detectar intentos de entrar en nuestros sistemas, asegurar el éxito de ciertas campañas publicitarias, etc. Por ejemplo, la cantidad de fraudes realizadas y la respuesta de los clientes a las campañas publicitarias.

   - ❒ **Evaluación de la situación**

     Esta tarea requiere un panorama diferenciado de la situación, considerando los siguientes aspectos, antes de iniciar el proceso de minería de datos: ¿Resolución

de problemas? Qué tan beneficioso es un programa de minería de datos En esta fase, se identifican las necesidades de minería de datos y los problemas comerciales [13].

- ❒ **Determinar los objetivos de la minería de datos**

  El propósito de esta asignación es presentar los objetivos comerciales en relación con los objetivos de un proyecto de minería de datos. Por ejemplo, si el objetivo de una empresa es desarrollar una campaña publicitaria para aumentar la atribución de préstamos hipotecarios, los objetivos de minería de datos pueden incluir la identificación de perfiles de clientes potenciales relacionados con préstamos.

- ❒ **Realizar el plan del proyecto**

  El propósito de esta última tarea en la primera fase de CRISP-DM es desarrollar un plan de proyecto que defina los pasos a realizar y los métodos a utilizar en cada paso.

2. **Compresión de los datos**

   La fase empieza con la recopilación inicial de datos, la familiarización con los datos, la identificación de problemas de calidad de los datos, la obtención de información temprana sobre los datos y el descubrimiento de subconjuntos interesantes para formular hipótesis sobre la información oculta.

   La compresión empresarial y la compresión de datos van de la mano. Para formular problemas de minería de datos y planes de proyectos, necesitamos al menos cierta compresión de los datos disponibles.

   - ❒ **Recolectar los datos iniciales**

     En esta tarea se debe recopilar los datos sin procesar y su integridad para su posterior procesamiento. El propósito de esta tarea es crear un informe que enumere los datos recopilados, su ubicación, las técnicas utilizadas para recopilarlos y cualquier problema y solución específicos del proceso [13].

   - ❒ **Descripción de los datos**

     A continuación, de recopilar los datos primarios, se deben describir. Este proceso incluye determinar la cantidad de datoso registros, definir los datos, qué expresa cada campo y explicar el formato original.

   - ❒ **Exploración de los datos**

Una vez recopilados los datos primarios, deben describirse. Después de ingresar la información, se confirmará. El objetivo es encontrar una estructura general de los datos. El resultado de esta misión es el Informe de análisis de datos.

- ❒ **Verificar la calidad de los datos**

  Esta función verifica la consistencia de los datos de los valores de campo individuales, el conteo y la distribución de ceros, y encuentra valores atípicos que pueden introducir ruido en el proceso. En este punto, se trata de garantizar la integridad y precisión de los datos.

3. **Preparación de los datos**

   Esta fase contiene todas las acciones para crear el conjunto de datos final (los datos que se introducirán en las herramientas de modelado) a continuación de los datos sin procesar. Las tareas de preparación de datos se pueden realizar varias veces en cualquier orden. Además incorporan el seleccionar tablas, conjuntos de datos y atributos, limpiar datos, crear nuevos atributos y transformar datos para herramientas de modelado [13].

   - ❒ Seleccionar los datos [13]
   - ❒ Limpiar los datos [13]
   - ❒ Construir los datos [13]
   - ❒ Integrar los datos [13]
   - ❒ Establecer un formato adecuado para los datos [13]

4. **Modelado**

   Esta fase se enfoca en la generación de un modelo que pueda lograr los objetivos del proyecto. En esta fase debe:

   - ❒ Escoger la técnica de modelado
   - ❒ Producir un plan de prueba
   - ❒ Levantar el modelo
   - ❒ Valorar el modelo

   Al encontrarnos ante un problema de clasificación. Aplicamos un método de aprendizaje supervisado, específicamente el clasificador SVM, para predecir ataques de

inyección SQL utilizando el conjunto de datos original. Construya el modelo dividiendo las observaciones en dos conjuntos, el uno de entrenamiento y el otro de prueba. Para hacer esto, tomamos el 70 % de las observaciones del conjunto de datos original y las etiquetas de clase asociadas del conjunto de entrenamiento. El 30 % restante se utilizará para el conjunto de prueba.

5. **Evaluación**

   Esta etapa se enfoca en evaluar la cercanía del modelo a los objetivos comerciales previamente establecidos. La métrica de evaluación más común es la precisión.

   Durante esta fase se considera:

   - ❒ Dimensionar los resultados
   - ❒ Verificar el proceso
   - ❒ Establecer los próximos pasos

6. **Despliegue o implementación**

   Esta fase se enfoca en entregar los resultados a los usuarios finales y mantenerlos después de que se complete la implementación.

   Esta fase debe realizar las siguientes tareas:

   - ❒ Proponer la implantación
   - ❒ Planificar el monitoreo de despliegue y mantenimiento
   - ❒ Elaborar el informe final
   - ❒ Verificar el proyecto

4.3 METODOLOGÍA DE DESARROLLO DE SOFTWARE XP

Basado en uno de los paradigmas más utilizados en el desarrollo de software, la programación extrema es un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planificación, diseño, codificación y prueba [15]. Como se muestra en la Figura 4.1.

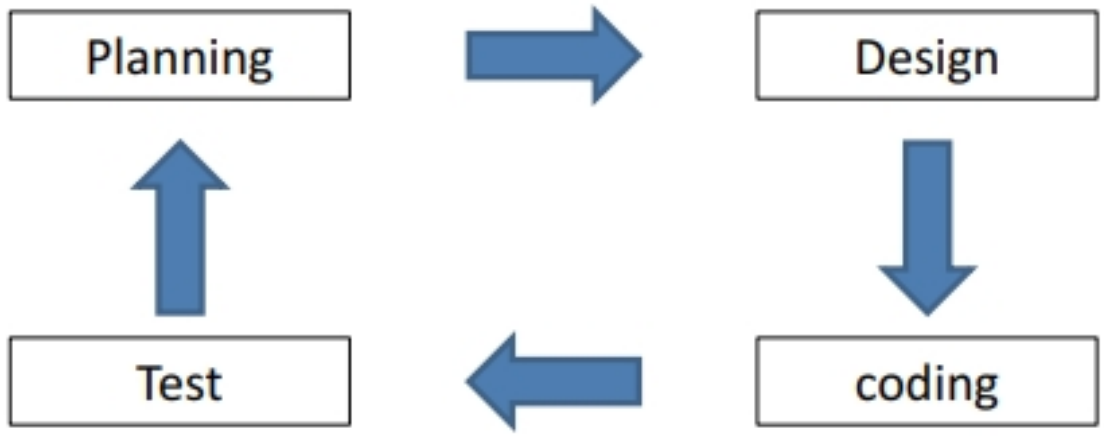


Figura 4.1: Proceso de la programación extrema [15].

4.3.1 Planeación

Esta es la etapa inicial de la metodología XP, donde se establece una comunicación continua entre el equipo de desarrollo y el cliente, con la fin de capturar los requisitos del sistema. Además, se puede establecer el alcance del proyecto y la fecha en la cual sera entregada del sistema considerando las prioridades de desarrollo y el tiempo estimado para cada historia de usuario.

4.3.1.1 Historias de usuario

Las historias de usuario deben escribirse en un lenguaje común que todos (clientes, desarrolladores y usuarios) puedan entender y expresar los requisitos que debe cumplir el sistema. Además, Las historias de usuario deben tener prioridades asignadas en relación con el valor del cliente en características y funcionalidad. Cada historia de usuario se evalúa y su valor se mide en semanas de desarrollo. Tenga en cuenta que puede modificar y crear historias de usuario [14]. Después de crear historias de usuario, se deben agrupar y seleccionar en el orden en que se desarrollaron. Debe especificar una fecha de entrega posterior, así como otros aspectos y detalles de su proyecto.

4.3.2 Diseño

El diseño de XP se basa en un principio muy particular, el cual la mayoria de las perosnas le conocen como Keep It Simple. El cual significa que un diseño simple es más valioso que un diseño complicado. También tenga en cuenta que el diseño debe guiar la ejecución de las historias de usuario y no debe tener en cuenta las funciones adicionales que los desarrolladores asumen. [14].

La metodología XP recomienda el uso de tarjetas CRC (Clase-Responsabilidad-Colaborador) como herramienta para pensar el software en un contexto orientado a objetos. La tarjeta CRC debe contener información de clase, responsabilidad y colaborador que le permita identificar y organizar las clases orientadas a objetos que son importantes para su incremento de software actual [14].

4.3.3 Codificación

Una vez que se han desarrollado las historias de usuario y se ha completado el trabajo de diseño preliminar, el equipo no pasa al código. Al contrario se procede a generar un conjunto de pruebas unitarias que prueban cada historia en la versión actual (software incremental). Una vez que se escriben las pruebas unitarias, los desarrolladores pueden concentrarse en lo que se debe implementar para que se aprueben las pruebas. Una vez que el código esté completo, puede probar sus unidades inmediatamente y proporcionar comentarios inmediatos a los desarrolladores. Un concepto importante en la codificación (y uno de los aspectos más discutidos de XP) es la programación en pares.

XP recomienda que dos personas trabajen juntas en la computadora para codificar la historia. Esto genera un mecanismo de solución de problemas en tiempo real. resolución (dos personas son mejores que una) y control de calidad en tiempo real (el código se verifica durante la construcción). Una vez que la pareja de desarrolladores ha terminado su trabajo, el código desarrollado se inyecta en otro trabajo. Esta palnificación de integración continua ayuda a identificar errores e interfaces en una etapa temprana.

4.3.4 Pruebas

Las pruebas unitarias creadas deben implementarse utilizando un marco que permita la automatización (para que puedan repetirse fácilmente). Esto fomenta la implementación de una estrategia de prueba de regresión cada vez que cambia el código (lo cual es común dada la mentalidad de refactorización de XP). Las pruebas de aceptación son comunmente conocidas como pruebas de clientes, las define el cliente y se enfocan en las características y funciones generales que el cliente puede ver y verificar. Estos se derivan de historias de usuarios implementadas como parte del lanzamiento del software.

5 DESARROLLO E IMPLEMENTACIÓN

Este proyecto se desarrolla e implementa a través de un proceso de minería aplicando CRISP-DM para obtener un modelo que pueda definir la efectividad del algoritmo SVM en la predicción de ataques de inyección SQL en BIGDATA. Adicionalmente, se desarrolla un prototipo utilizando la metodología XP.

5.1 DESARROLLO DEL MODELO DE MINERÍA DE DATOS

El presente proyecto integrador fue realizado mediante las diferentes fases que presenta la metodología CRISP-DM. Cabe recalcar que esto fue posible debido a los paquetes de registros de transacciones otorgados por la CSIRT-EPN. A continuación se muestra la ejecución de la metodología CRISP-DM en cada una de sus fases:

5.1.1 Comprensión del negocio

a. Determinación de los objetivos comerciales

A partir de la investigación realizada en el CSIRT de la Escuela Politécnica Nacional se pudieron identificar los siguientes objetivos principales:

- ❒ Determinar una técnica efectiva para detectar y predecir de manera efectiva los ataques de SQLIAs en los diversos sistemas que se encuentran en la institución
- ❒ Obtener un sistema automatizado para monitorear los posibles ataques de inyección SQL según la técnica determinada.

b. Evaluación de la situación

Para evaluar la situación del CSIRT de la EPN es fundamental los siguientes factores:

- ❒ **Personal:** Se cuenta con la supervisión de los miembros del CSIRT para realizar

la gestión y asesoría necesaria dentro del proceso de extracción de datos.

- ❒ **Datos:** Los datos que una organización proporciona para la minería de datos se obtienen de una muestra diversa de registros producidos por los sistemas de información utilizados por la organización.
- ❒ **Riesgos:** Es importante notar que los datos utilizados provienen de un entorno de producción. Estos datos resultan ser de uso interno de la organización, y por lo tanto no pueden ser divulgados de manera externa. Si los datos utilizados se difunden podrían resultar en una brecha de seguridad.

c. Determinación de los objetivos de minería de datos

El proyecto debe producir un modelo capaz de clasificar los datos provenientes de las diversas fuentes de datos. Los datos pueden clasificarse en datos con anomalías (posibles SQLIA) y datos sin anomalías. Para generar este modelo deben utilizarse los datos provistos por los registros y logs que actualmente se encuentran en el CSIRT. En la creación del modelo se debe considerar el factor que sobresale el cual es la sentencia SQL ingresada al sistema. Ya que el sistema solo con leer la sentencia podra identificar si es un ataque de inyección SQL o no es ataque.

d. Producción de un plan de proyecto

En la producción de un plan de proyecto se elaboró un cronograma con cada una de las fases de la metodología. En este cronograma se especifican los tiempos, recursos, y riesgos asociados en cada fase. El plan del proyecto se obseva en la Tabla 5.1

Tabla 5.1: Plan de proyecto de minería de datos aplicando las fases de la metodología CRISP-DM

Fase	Tiempo	Recursos	Riesgos
Comprensión del negocio	1 semana	Miembros del CSIRT	Acuerdos de confidencialidad
Comprensión de los datos	2 semanas	Miembros del CSIRT	Anomalías en los datos
Preparación de los datos	3 semanas	equipo de desarrollo	Falta de comprensión en los datos, anomalías en los datos
Modelado	2 semanas	Equipo de desarrollo	Dificultad para encontrar modelos adecuados
Evaluación	1 semana	Miembros del CSIRT y equipo de desarrollo	Dificultad para lograr una implementación adecuada

Fase	Tiempo	Recursos	Riesgos
Despliegue	1 semana	Equipo de desarrollo	Dificultad para lograr una implementación adecuada

5.1.2 Comprensión de los datos

En esta etapa de la metodología CRISP-DM, se encarga de recopilar datos primarios que sean de alta calidad y permitan la resolución de problemas. A esto le sigue la descripción, investigación y verificación de los datos obtenidos con el fin de eliminar información innecesaria que impide la creación de un conjunto de datos valioso [1].

a. **Recopilación de datos iniciales**

La obtención de los datos iniciales se realizo en el ambiente de producción de la CSIRT, las operaciones ejecutadas por los usuarios al sistema fueron almacenadas en sus respectivos logs. Los cuales son fundamentales para entender y trabajar los datos en el proyecto de detección de ataques de inyección SQL.

A continuación, se lista toda la información contenida dentro de un registro de transacción:

%an = application name (nombre de la aplicación)

%un = nombre de usuario

%db = nombre de base de datos

%h = nombre del servidor

%pi = identificador del proceso

%tm = marca de tiempo sin milisegundos

%ms = marca de tiempo con milisegundos

%ic = etiqueta de comando

%eq = sentencia SQL

%c = identificador de la sesión

Estos parámetros obtenidos en los logs trabajados contienen demasiada información, los cuales son complejos la momento de entender, dado que no aportan suficiente

valor para el objetivo planteado del proyecto, el cual es la detección de inyección SQL al sistema. Por tal motivo se procedió a realizar un filtrado de los diferentes parámetros para identificar cuales de ellos si nos otorgarán valor para trabajarlos.

La información que nos será útil para poder realizar la detección será el:

%i = command tag (etiqueta de comando)

Está sentencia es la única que se necesitará, ya que los datos de entrenamiento constan con una etiqueta la cual es 0 o 1. Está etiqueta nos ayuda a que el modelo puedo comprender e identificar que las sentencias con etiqueta 1 son ataques y 0 no ataques.

La estructura de los logs obtenido por la CSIRT es:

```
...
"log": {
"file": {
"path": "...",
},
"offset": "...",
"flags": ["..."]
},
"fileset": {
"name": "...",

},
"message":
"timestamp=... ,process_id=... ,session_number=... ,user=... ,db
=... ,app=....,client=... ,LOG: duration: ... ms bind ...:
SELECT...
timestamp=... ,process_id=...,session_number=...,user=...,db=...,
app=...,client=...,DETAIL ...
...
...
...",
"fileset":{
```

```
"name": "...",
},
"error": {
"message":"Provided Grok expressions do not match field value:[
timestamp=... ,process_id=... ,session_number=... ,user=... ,db
=... ,app=....,client=... ,LOG: duration: ... ms bind ...:
SELECT...
timestamp=... ,process_id=...,session_number=...,user=...,db=...,
app=...,client=...,DETAIL ...
...
...
...]",
}
},
"input": {
"type": "...",
},
...
```

b. **Descripción de los datos**

Como se puede visualizar en la Tabla 5.2, se encuentra una simulación de una transacción normal al sistema.

Tabla 5.2: Descripción de los Atributos de los Registros de Transacciones

Valor	Efecto
%i	La etiqueta de comando devuelve la sentencia ejecutada como, por ejemplo: SELECT, INSERT, UPDATE, DELETE

d. **Verificación de calidad de datos**

En esta fase, se efectúa la verificaciones de los datos, para determinar la consistencia de los valores individuales de los campos como: la cantidad y distribución de los valores nulos, y para encontrar valores fuera de rango, los cuales pueden causar molestias en el proceso.

5.1.3 Preparación de los datos

De la información recopilada. Los datos están listos para aplicarles técnicas de minería de datos.

a. **Selección de datos**

Esta fase selecciona los atributos de registro más importantes para aplicar modelos de minería de datos. Puede ver los datos seleccionados en la Tabla 5.3

Tabla 5.3: Datos seleccionados para la aplicación de la metodología CRISP-DM

Atributo	Descripción del atributo	Tipo de dato
anómalo	Indica si el registro refleja alguna anomalía que podría considerarse como un SQLIA	Cadena de caracteres (típico/atípico)

b. **Limpieza de datos**

Luego, se genera un .csv, solo con los atributos especificados en la Tabla 5.3 para crear un nuevo grupo de datos con los atributos que ayudarán a entrenar el modélo para su respectiva clasificación eficaz. Esto se hizo usando la biblioteca pandas. Se eliminaron ciertos registros de transacciones capturados con valores nulos en etiquetas de etiquetas para evitar afectar la detección de anomalías.

c. **Construcción e integración de los datos**

Para tener un formato de consulta SQL coherente, cada consulta contiene parámetros para un análisis más detallado en la siguiente fase. Los datos proporcionados para este estudio provienen de una sola fuente y no requieren integración con otras fuentes.

El resultado de esta fase fue un archivo con 3 millones de registros de sentencias SQL almacenados en un archivo de implementación CSV.

5.1.4 Modelado

Se ejecutó el algoritmo SVM con Python por su facilidad de la manipulación de grandes volúmenes de datos. SVM (Support Vector Machine) es una técnica útil para la clasificación.

El objetivo de SVM es clasificar los datos respectivamente a su etiqueta con la cual fue entrenada el modelo. Este modelo tiene valor objetivo generar una etiqueta de predicción la cual nos ayuda a clasificar los datos como ataques de inyección y no ataques de inyección SQL.

- **Uso de SVM**

  El término SVM[13] se usa típicamente para describir la clasificación con métodos de vector de soporte y la regresión de vector de soporte se usa para describir la regresión con métodos de vector de soporte.

  En está investigación, nos hemos enfocado en las propiedas de la consulta de la sentencia SQL, mediante la libreria pipeline de Spark, la cual nos permitió generar un tokenizador de cadena que crea tokens de la consulta original y la consulta inyectada, y crea una matriz de tokens tanto de la consulta original como de la inyectada, si la longitud de las matrices de ambas consultas es igual, eso significa que no hay inyección de SQL. De lo contrario, hay inyección. En resumen, la clasificación de los ataques y no ataques se realiza analizando los conjuntos de datos de la descomposición de las difetentes sentencias SQL.

- **Algoritmo**

  Pasos para generación del algoritmo:

  Paso 1. Descomposición de las sentencias en valores númericos.

  Paso 2. Seleccionar una cantidad razonable como conjunto de entrenamiento.

  Paso 3. Introducir el conjunto de entrenamiento en el proceso SVM-Train para la generación del modelo.

  Paso 4 Ejecutar el modelo usando el clasificador SVM.

  Paso 5 Obtener etiqueta 0 o 1, nos dará la precisión de nuestro algoritmo.

  Se obtuvieron varios conjuntos de datos de varias fuentes para entrenar el modelo SVM. Las etiquetas para estos conjuntos de datos oscilaron entre el conjunto anómalo (1) y el conjunto no anómalo (0), como se muestra en la Tabla 5.4:

- **Evaluación del modelo**

  En la evaluación del modelo se procedió a utilizar la librería de Evaluador de Clasificación Multiclase. Procederemos a utilizar la función evaluate(), la cual recibe como

Secuencia de texto	Descripción	Etiqueta	Cantidad
Sentencias Sql Normales	Sentencias no anómalas que usadas para la consulta de información	0	11382
Sentencias Sql anómalas	Sentencias que contiene código SQL	1	19537

Tabla 5.4: Etiquetado de cadenas SQL

parámetro la columna de predicciones y se obtiene como salida el porcentaje de precisión del modelo a evaluar. Adicional se creo una una matriz de confusión con los datos de la Tabla 5.4, con el fin de obtener metricas cuantitativas de evaluación. Estos datos obtenidos se separan en Verdaderos positivos, falsos negativos, falsos positivos, verdadedores negativos.

5.1.5 Evaluación

Una vez entrenado el modelo, se obtiene una matriz de confusión como se visualiza a continuación Figura 5.1. Para mayor entendimiento de las métricas obtenidas leer la Tabla 5.5

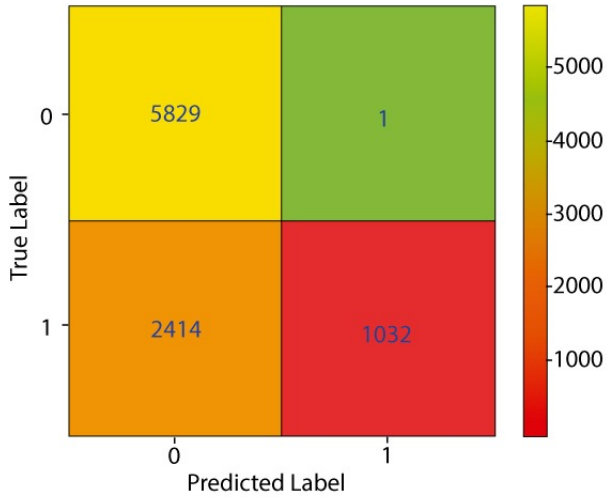


Figura 5.1: Matriz de confusión del Modelo - SVM Fuente: Autoría propia

Verdaderos Positivos	TP= 5829	Falsos Negativos	FN= 1
Falso Positivos	FP = 2414	Verdadero Negativo	TN=1032

Tabla 5.5: Resultados de la matriz de confusión - Modelo SVM

Con los datos cuantitativos de la Tabla 5.5, se realizan la siguientes mediciones :

❒ **Exactitud A:** Porcentaje global de elementos clasificados correctamente.

$$A = \frac{TP - TN}{TP + FP + FN + TN}$$

$$A = \frac{5829 + 1032}{5829 + 2414 + 1 + 1032}(100\,\%) = 73.96\,\%$$

❒ **Sensibilidad o TPR(Tasa de Verdadero positivos):** Número de elementos correctamente identificados como positivos del total de verdaderos positivos.

$$TPR = \frac{TP}{TP + TN}$$

$$TPR = \frac{5829}{5829 + 1032}(100\,\%) = 84.95\,\%$$

❒ **Precisión P**: El número de elementos correctamente identificados como positivos de todos los elementos identificados como positivos.

$$P = \frac{TP}{TP + FP}$$

$$P = \frac{5829}{5829 + 2414}(100\,\%) = 70.71\,\%$$

❒ **Especificidad TNR (Tasa negativa verdadera):** Este es el número de artículos correctamente identificados como negativos del número total de negativos.

$$TNR = \frac{TN}{TN + FP}$$

$$TNR = \frac{1032}{1032 + 2414}(100\,\%) = 29.94\,\%$$

Una vez que se entrena el modelo, las comprobaciones se realizan mediante sentencias SQL regulares y ataques de inyección SQL, como se muestra en la Figura 5.2.

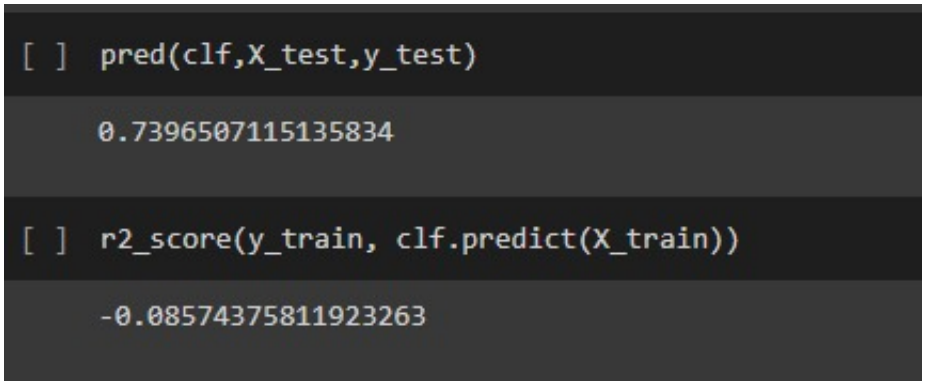


Figura 5.2: Evaluación simple del modelo SVM

Finalmente nuestro algoritmo SVM ha sido probada en un conjunto de datos de cadenas de consultas SQL. Para la prueba de la muestra se manejo 1 millon de datos de sentencias SQL para ser analizadas, lo cual nos dio como resultado los siguientes valores de la Tabla 5.6.

Cadena de texto	Cantidad	Porcentaje
Sentencias SQL Normales	739600	73.96 %
Sentencias SQL anómalas	260400	26.04 %

Tabla 5.6: Resultados de análisis de las muestras

5.1.6 Despliegue

Esta fase tiene como objetivo principal aclarar al cliente como poner en funcionamiento el proyecto que se ha realizado en las fases anteriores. En este proyecto decidimos utilizar la metodología de Programación Extrema con el fin de presentar un prototipo de sistema para visualizar los resultados del modelo SVM creado por la metodología CRISP-DM.

Esta fase consta de las siguientes tareas como:

- ❒ Planificación de despliegue
- ❒ Planificación del control y del mantenimiento
- ❒ Creación de un informe final
- ❒ Revisión final del proyecto

5.2 DESARROLLO DE LA APLICACIÓN WEB

Para el desarrollo de un sistema web se identificaron las principales funciones y características que debe tener este sistema. Comprender estos aspectos es muy importante al implementar una aplicación web, ya que depende de la facilidad con la que los usuarios puedan interactuar con el sistema.

5.2.1 Planeación

En XP, se ejecuta un plan dinámico en lugar de un plan global que detalla cada tarea. En este proyecto, solo se abarco una visión global de alto nivel y se ejecuta una planificación detallada para cada iteración o entrega, dado a esta característica permite que la planificación sea mucho más precisa y nos permite obtener más control sobre nuestro proyecto, porque siempre sabemos lo que está sucediendo en él.

5.2.1.1 Historias de usuario

Para el desarrollo de un sistema web se identificaron las principales funciones y características que debe tener este sistema. Comprender estos aspectos es muy importante al implementar una aplicación web, ya que depende de la facilidad con la que los usuarios puedan interactuar con el sistema. Una vez que se identificaron los requisitos, se crearon historias de usuarios y se agruparon en la Tabla 5.7 según la prioridad y el esfuerzo.

5.2.1.2 Plan de entrega del proyecto

Esta sección define el plan de entrega del proyecto en función de las historias de usuario de la Tabla. 5.7. En función de la prioridad y el esfuerzo de cada historia de usuario, el tiempo de entrega se estima como se muestra en la Tabla 5.8.

Finalmente, la asignación de pares de codificación y asignación de tareas para cada iteración se realiza como se muestra en la tabla 5.9.

Código	Titulo	Descripción	Prioridad
HU-01	Control de acceso	Como administrador deseo poder mantener un control de acceso dentro del sistema para evitar la divulgación de información sensible de la organización	2
HU-02	Carga de datos	Como usuario deseo poder realizar la carga de un log de cualquier tamaño , para que pueda ser procesado y limpiado de manera que pueda ser utilizado para detectar sentencias de ataque de inyección SQL	4
HU-03	Análisis	Como usuario deseo evaluar diferentes modelos de Machine Learning para detectar ataques de inyección SQL	3
HU-04	Visualización de logs anómalos	Como administrador del sistema deseo visualizar las sentencias SQL detectadas como posibles ataques de inyección SQL para aumentar la seguridad de los sistemas de donde se obtuvo la información	1

Tabla 5.7: Historias de usuario para el desarrollo de prototipo

Código historias de usuario	Iteración	Prioridad	Duración en semanas
HU-01	1	1	1
HU-02	1	2	1
HU-03	2	3	1
HU-04	2	4	1

Tabla 5.8: Plan de entrega de proyecto

Integrantes	Pareja	Historias de usuario
Edison Quimbiamba, Ándres Palma	1	HU-01, HU-03
Steven Rivera, Ándres Llumiquinga	1	HU-02, HU-04

Tabla 5.9: Asignación de parejas y responsabilidades.

5.2.2 Diseño

5.2.2.1 Arquitectura del sistema software

Un prototipo web se crea conceptualmente con una arquitectura cliente/servidor que permite que un cliente interactúe con un servidor a través de solicitudes HTTP. Junto con el prototipo, se basa en el patrón arquitectónico Model View Controller "MVC"que ayuda a separar los datos de la lógica comercial de su aplicación como se muestra en la Figura 5.3 .

El patrón MVC sugiere tres componentes: vistas, controladores y modelos. De esta forma, es más fácil para el usuario interactuar con el componente, a la vez que se facilita la visualización de la información, reduciendo la complejidad del desarrollo del prototipo y posterior mantenimiento.

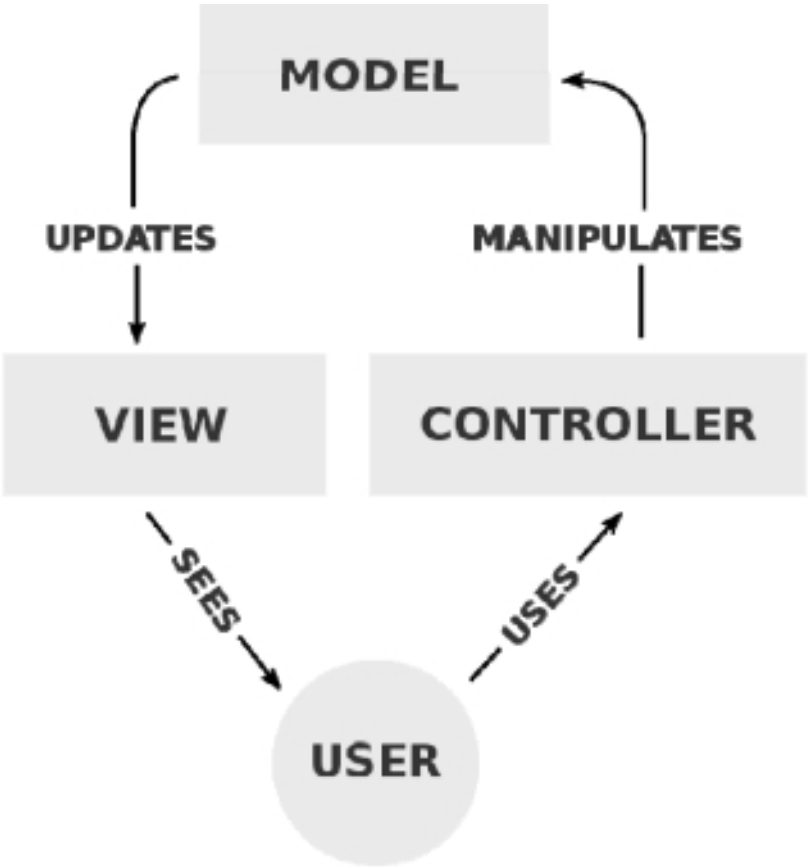
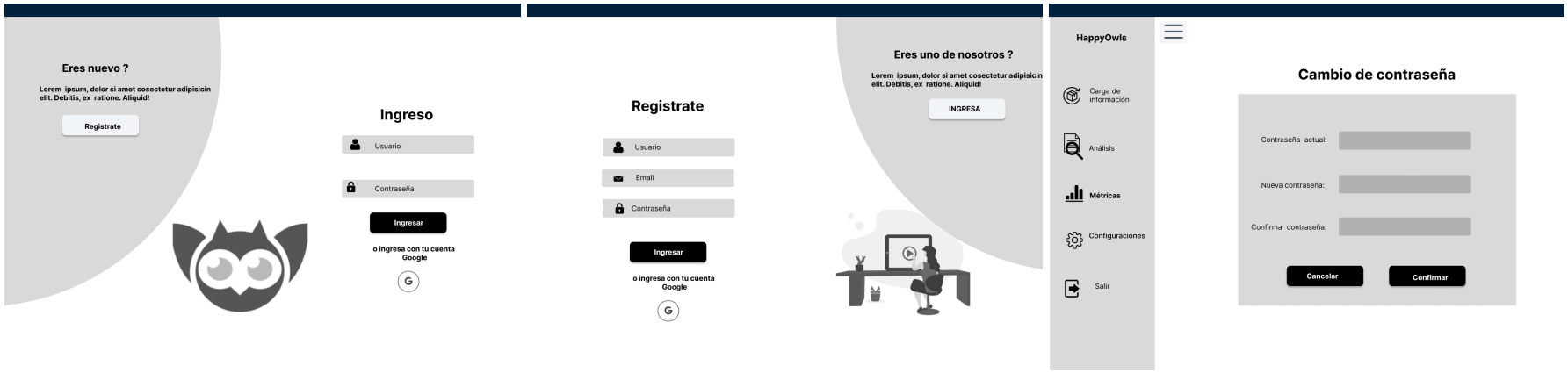


Figura 5.3: Patron MVC Fuente [18]

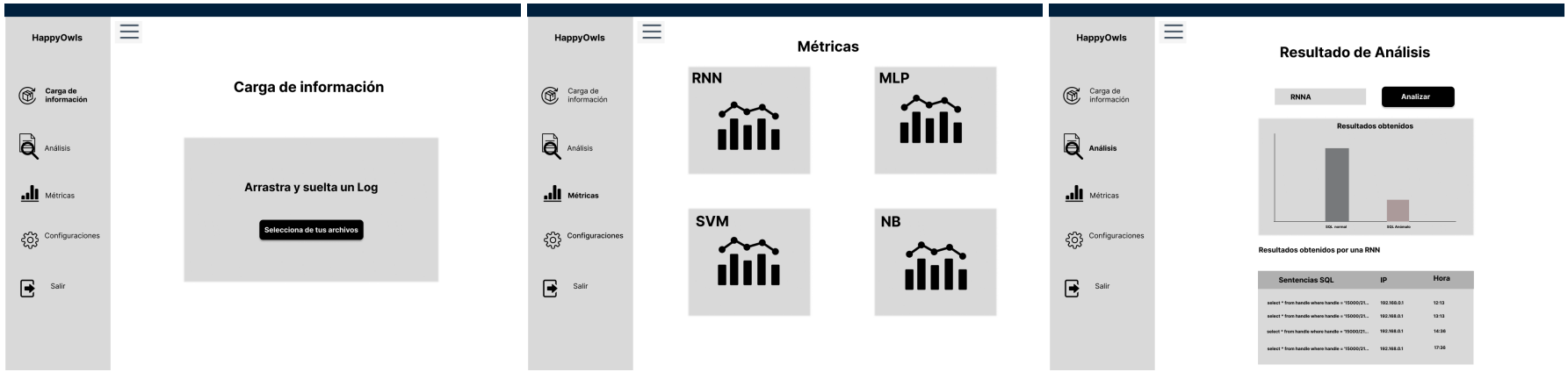
5.2.2.2 Diseño interfaces

El propósito de esta etapa es implementar el proceso de interacción se lleve a cabo de una forma sencilla e intuitiva. Para ello, se han desarrollado las siguientes interfaces de baja fidelidad, es decir, son prototipos que no tendrán el mismo aspecto real de producción, pero su funcionalidad sera la misma. Desarrollo de interfaces basadas en requisitos definidos en la Tabla 5.7. En la Figura 5.4, se muestra la interfaz de control de acceso al sistema web y en la Figura 5.5, se muestra una interfaz para el módulo de análisis de datos.



(a) Login **(b)** Registro **(c)** Cambio de contraseña

Figura 5.4: Interfaces para control de acceso



(a) Carga de información **(b)** Métricas de algoritmos **(c)** Análisis

Figura 5.5: Módulos del prototipo web

Después de definir el prototipo de baja fidelidad, el siguiente capítulo analizará la fase de codificación y aclarará los requisitos para crear una aplicación web.

5.2.3 Codificación

Una vez que se definen los requisitos y la arquitectura de la aplicación web, se construye un grupo de iteraciones, las cuales son consideradas dos, según el plan definido en la fase anterior. La primera iteración relacionada con la configuración del control de acceso y la segunda iteración donde se crean los módulos de análisis de datos.

5.2.3.1 Primera iteración

El propósito de esta iteración es generar una extensión que permita a los usuarios registrarse y abrir secciones, controlando así su acceso a la información.

La Figura 5.6 contiene una lista de tareas relacionadas con la historia seleccionada en esta iteración.

Figura 5.6: Lista de tareas para la iteración 1, Fuente: Autoria Propia

5.2.3.2 Segunda iteración

Esta iteración tiene como objetivo generar un aumento de funcionalidad para la selección de algoritmos de minería de datos y la visualización de los resultados obtenidos a partir del análisis de los datos proporcionados por la organización..

La Figura 5.6 contiene una lista de tareas relacionadas con la historia seleccionada en esta iteración.

Figura 5.7: Lista de tareas para la iteración 2, Fuente: Autoria Propia

6 ANÁLISIS DE RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

Esta sección presenta los resultados de un estudio sistemático de la literatura, la evaluación de modelos de clasificación utilizando datos reales y de entrenamiento, y el desarrollo de aplicaciones web para el procesamiento de información. Así como conclusiones y recomendaciones.

6.1 ANÁLISIS DE RESULTADOS

La revisión sistemática de la literatura realizada en este proyecto nos ha permitido determinar diferentes tipos de técnicas de ataques SQLIAs. Además pudimos concluir que el 38 %, es decir 19 de los artículos analizados utilizaban algoritmos de aprendizaje automático, con el objetivo de predecir ataques de SQLIAs.

Las técnicas de aprendizaje automático incluyen el algoritmo SVM. En este proyecto se desarrolló un modelo basado en maquinas de vectores de soporte. Es un tipo de separación en forma de hiperplano que se especializa en procesar secuencias de datos con etiquetas, con el objtetivo de poder predecir lo que sucederá en el futuro en base a datos anteriores. De esta forma, la inteligencia artificial tiene la capacidad de poder identificar si es un ataque o no es un ataque de inyección SQL. El algoritmo SVM aplicado a la detección de ataques de inyección tiene una precisión del 73,96 %, una sensibilidad del 84,95 %, una precisión del 70,71 % y una especificidad del 29,94 %. En conclusión, podemos decir que este modelo es el menos óptimo cuando se trata de clasificar una gran cantidad de datos y textos, ya que no es correcto utilizarlo para detectar ataques de inyección SQL.

Como se puede visualizar en la Figura 6.1, al momento de detectar 1 millon de datos nos da como resultado la obtención de solo haber detectado 1 ataque de inyección y todas como sentencias normales es decir 999999 de sentencias SQL normales.

Figura 6.1: Resultado en interfaz web Fuente: Autoria Propia

6.2 CONCLUSIONES

- ❒ Nuestro proyecto ayuda proporcionar o mejorar una aplicación para elevar el nivel de seguridad, basada en la clasificación de cadenas de consulta sin ataques y con ataques, utilizando el algortimo de clasificación SVM.
- ❒ Las pruebas funcionales durante la fase de evaluación del método CRISP-DM mostraron que el algortimo SVM implementado en el sistema no da una gran presición.
- ❒ La implementación del modelo de minería de datos fue guiado por las diferentes fases de CRISP-DM. La metodología CRISP-DM simplificó el proceso al considerar las diversas fases y su simplicidad, permitiéndonos avanzar y retroceder entre las diversas fases. Su simplicidad se pudo demostrar al momento de crear el modelo, exisitian datos con valr null, lo cual esto no permitia una predicción de alto nivel, por lo cual se regreso a los pasos de limpieza y se pudo eliminar esos datos inecesarios. A continuación de la correción de esos datos, se continuo con las siguientes tareas de la metodología CRISP-DM.
- ❒ El algoritmo SVM es muy propenso a errores cuando se trata de datos muy ruidosos,

es decir tiene varios campos de características superpuestas para diferentes etiquetas. Pero como en este proyecto su etiqueta era simple 1 igual a ataque y 0 a no ataque, se suponía que iba a tener una respuesta óptima al ejecutarse. Sin embargo, uno de los principales defectos es que no tiene tanta efectividad cuando es el momento de analizar una gran cantidad de datos como la de este proyecto.

- ❒ Las aplicaciones web desarrolladas están limitadas por variables computacionales, especialmente cuando se analizan grandes cantidades de información. Sin embargo, los algoritmos seleccionados se pueden utilizar en BIG DATA utilizando técnicas de multiprocesamiento de la información, por lo que se pueden considerar factores como la velocidad de análisis, la diversidad de datos y el volumen de datos para obtener un comportamiento deseable en el mundo real.

- ❒ pySpark es una gran herramienta para lo que es manejo de grandes volumenes de datos, dado que nos da respuestas de predicciones en el minimo tiempo, recalcando su gran rendimiento y eficacia. Minimizando el consumo de recursos computacionales. En este proyecto se utiliza grandes volumenes de datos para el entrenamiento y la clasificación. Por lo cual, esta herramiento nos facilitó bastante el trabajo y el consumo de recursos, los cuales fueron limitados. Nuestro algoritmo SVM muestra el resultado de rendimiento en precisión, que es del 73.96 %.

6.3 RECOMENDACIONES

- ❒ Se recomienda usar la libreria Pipeline de Spark para la clasificación de textos, dado a su facilidad de descomposición de textos y transformación a valores numéricos.

- ❒ Se recomiendo utilizar el algoritmo SVM para conjuntos de datos moderados, es decir no tengan tanta complejidad en su estructura, para obtener un rendimiento óptimo del modelo.

- ❒ Se recomienda implementar diferentes técnicas de validaciones para el incremento de la presición de la predicción de datos.

- ❒ Se recomienda investigar a profundidad los objetivos especificos a los cuales se va a implementar los algoritmos de machine learning, dado que las etiquetas o el número de etiquetas de los datos, va a inferir mucho en la precisión de los algoritmos que se

inplementarán. Esto significa que la selección del algoritmo correcto, va de la mano de los datos a analizar y la respuesta al problema a solucionar.

- ❒ Para iniciar una búsqueda bibliográfica sistemática, es necesario identificar el problema a abordar y confrontar el problema a resolver con la investigación. Además, debido a que la recopilación de información debe ser completa tanto para los estudios publicados como para los no publicados, se recomienda establecer criterios de inclusión y exclusión con el fin de evitar sesgos en elegir estudios relevantes.

7 REFERENCIAS BIBLIOGRÁFICAS

[1] *SQL Injection*, en, https://owasp.org/www-community/attacks/SQL_Injection.

[2] R. Von Solms y J. Van Niekerk, «From information security to cyber security,» *computers & security*, vol. 38, págs. 97-102, 2013.

[3] N. T. Mitsuo Wada, «Infrared Spectroscopy of Metal-MgO Single Crystalline Composite Films,» *Physics*, vol. 29, págs. L1497-L1499, 2014.

[4] S. Agarwal, «Data mining: Data mining concepts and techniques,» en *2013 international conference on machine intelligence and research advancement*, IEEE, 2013, págs. 203-207.

[5] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. .ºReilly Media, Inc.", 2019.

[6] M. K. Obenshain, «Application of data mining techniques to healthcare data,» *Infection Control & Hospital Epidemiology*, vol. 25, n.º 8, págs. 690-695, 2004.

[7] B. Kitchenham y S. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering*, 2007.

[8] R. K. Jamra, B. Anggorojati, D. I. Sensuse, R. R. Suryono et al., «Systematic Review of Issues and Solutions for Security in E-commerce,» en *2020 International Conference on Electrical Engineering and Informatics (ICELTICs)*, IEEE, 2020, págs. 1-5.

[9] Y.-C. Chung, M.-C. Wu, Y.-C. Chen y W.-K. Chang, «A Hot Query Bank approach to improve detection performance against SQL injection attacks,» *computers & security*, vol. 31, n.º 2, págs. 233-248, 2012.

[10] W. G. Halfond, J. Viegas, A. Orso et al., «A classification of SQL-injection attacks and countermeasures,» en *Proceedings of the IEEE international symposium on secure software engineering*, IEEE, vol. 1, 2006, págs. 13-15.

[11] W. G. J. Halfond y A. Orso, «AMNESIA: Analysis and Monitoring for NEutralizing SQL-Injection Attacks,» en *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering*, ép. ASE '05, Long Beach, CA, USA: Association for Computing Machinery, 2005, págs. 174-183, ISBN: 1581139934. DOI: 10.1145/1101908.1101935. dirección: https://doi.org/10.1145/1101908.1101935.

[12] B. Kitchenham, «Procedures for performing systematic reviews,» *Keele, UK, Keele University*, vol. 33, n.º 2004, págs. 1-26, 2004.

[13] R. Wirth y J. Hipp, «CRISP-DM: Towards a standard process model for data mining,» en *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, Manchester, vol. 1, 2000, págs. 29-39.

[14] R. S. Pressman y J. M. Troya, «Ingenierı´a del software,» 1988.

[15] T. N. Takaaki Goto Kensei Tsuchida, «EPISODE: An Extreme Programming Method for Innovative Software Based on Systems Design,» *Physics*, vol. 3, págs. 780-784, 2014.

[16] I. Lee, S. Jeong, S. Yeo y J. Moon, «A novel method for SQL injection attack detection based on removing SQL query attribute values,» *Mathematical and Computer Modelling*, vol. 55, n.º 1, págs. 58-68, 2012, Advanced Theory and Practice for Cryptography and Future Security, ISSN: 0895-7177. DOI: https://doi.org/10.1016/j.mcm.2011.01.050. dirección: https://www.sciencedirect.com/science/article/pii/S0895717711000689.

[17] C. I. Pinzón, J. F. De Paz, Á. Herrero, E. Corchado, J. Bajo y J. M. Corchado, «idMAS-SQL: Intrusion Detection Based on MAS to Detect and Block SQL injection through data mining,» *Information Sciences*, vol. 231, págs. 15-31, 2013, Data Mining for Information Security, ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2011.06.020. dirección: https://www.sciencedirect.com/science/article/pii/S0020025511003148.

[18] M.-Y. Kim y D. H. Lee, «Data-mining based SQL injection attack detection using internal query trees,» *Expert Systems with Applications*, vol. 41, n.º 11, págs. 5416-5430, 2014, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2014.02.041. dirección: https://www.sciencedirect.com/science/article/pii/S0957417414001171.

[19] H. Shahriar y M. Zulkernine, «Information-Theoretic Detection of SQL Injection Attacks,» en *2012 IEEE 14th International Symposium on High-Assurance Systems Engineering*, 2012, págs. 40-47. DOI: 10.1109/HASE.2012.31.

[20] S. Som, S. Sinha y R. Kataria, «Study on sql injection attacks: Mode detection and prevention,» *International Journal of Engineering Applied Sciences and Technology*, vol. 1, n.º 8, págs. 23-29, 2016.

[21] I. Balasundaram y E. Ramaraj, «An Efficient Technique for Detection and Prevention of SQL Injection Attack using ASCII Based String Matching,» *Procedia Engineering*, vol. 30, págs. 183-190, 2012, International Conference on Communication Technology and System Design 2011, ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2012.01.850. dirección: https://www.sciencedirect.com/science/article/pii/S1877705812008600.

[22] T. Latchoumi, M. S. Reddy y K. Balamurugan, «Applied machine learning predictive analytics to SQL injection attack detection and prevention,» *European Journal of Molecular & Clinical Medicine*, vol. 7, n.º 02, pág. 2020, 2020.

[23] P. Tang, W. Qiu, Z. Huang, H. Lian y G. Liu, «Detection of SQL injection based on artificial neural network,» *Knowledge-Based Systems*, vol. 190, pág. 105 528, 2020, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2020.105528. dirección: https://www.sciencedirect.com/science/article/pii/S0950705120300332.

[24] Q. Li, W. Li, J. Wang y M. Cheng, «A SQL Injection Detection Method Based on Adaptive Deep Forest,» *IEEE Access*, vol. 7, págs. 145 385-145 394, 2019. DOI: 10.1109/ACCESS.2019.2944951.

[25] N. M. Sheykhkanloo, «Employing Neural Networks for the Detection of SQL Injection Attack,» en *Proceedings of the 7th International Conference on Security of Information and Networks*, ép. SIN '14, Glasgow, Scotland, UK: Association for Computing Machinery, 2014, págs. 318-323, ISBN: 9781450330336. DOI: 10.1145/2659651.2659675. dirección: https://doi.org/10.1145/2659651.2659675.

[26] D. Kar, S. Panigrahi y S. Sundararajan, «SQLiDDS: SQL Injection Detection Using Query Transformation and Document Similarity,» en *Distributed Computing and Internet Technology*, R. Natarajan, G. Barua y M. R. Patra, eds., Cham: Springer International Publishing, 2015, págs. 377-390, ISBN: 978-3-319-14977-6.

[27] A. Ghafarian, «A hybrid method for detection and prevention of SQL injection attacks,» en *2017 Computing Conference*, 2017, págs. 833-838. DOI: 10.1109/SAI.2017.8252192.

[28] X. Xie, C. Ren, Y. Fu, J. Xu y J. Guo, «SQL Injection Detection for Web Applications Based on Elastic-Pooling CNN,» *IEEE Access*, vol. 7, págs. 151 475-151 481, 2019. DOI: 10.1109/ACCESS.2019.2947527.

[29] Y. Wang y Z. Li, «SQL injection detection via program tracing and machine learning,» en *International Conference on Internet and Distributed Computing Systems*, Springer, 2012, págs. 264-274.

[30] H. Gu, J. Zhang, T. Liu et al., «DIAVA: A Traffic-Based Framework for Detection of SQL Injection Attacks and Vulnerability Analysis of Leaked Data,» *IEEE Transactions on Reliability*, vol. 69, n.º 1, págs. 188-202, 2020. DOI: 10.1109/TR.2019.2925415.

[31] R. A. Katole, S. S. Sherekar y V. M. Thakare, «Detection of SQL injection attacks by removing the parameter values of SQL query,» en *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, 2018, págs. 736-741. DOI: 10.1109/ICISC.2018.8398896.

[32] K. Ross, M. Moh, T.-S. Moh y J. Yao, «Multi-Source Data Analysis and Evaluation of Machine Learning Techniques for SQL Injection Detection,» en *Proceedings of the ACMSE 2018 Conference*, ép. ACMSE '18, Richmond, Kentucky: Association for Computing Machinery, 2018, ISBN: 9781450356961. DOI: 10.1145/3190645.3190670. dirección: https://doi.org/10.1145/3190645.3190670.

[33] K. N. Durai, R. Subha y A. Haldorai, «A Novel Method to Detect and Prevent SQLIA Using Ontology to Cloud Web Security,» *Wireless Personal Communications*, vol. 117, n.º 4, págs. 2995-3014, 2021.

[34] J. O. Atoum y A. J. Qaralleh, «A hybrid technique for SQL injection attacks detection and prevention,» *International Journal of Database Management Systems*, vol. 6, n.º 1, pág. 21, 2014.

[35] N. M. Sheykhkanloo, «A learning-based neural network model for the detection and classification of SQL injection attacks,» *International Journal of Cyber Warfare and Terrorism (IJCWT)*, vol. 7, n.º 2, págs. 16-41, 2017.

[36] S. Bangre, A. Jaiswal et al., «SQL Injection Detection and Prevention Using Input Filter Technique,» *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 1, n.º 2, págs. 145-150, 2012.

[37] M. Hasan, Z. Balbahaith y M. Tarique, «Detection of SQL Injection Attacks: A Machine Learning Approach,» en *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, 2019, págs. 1-6. DOI: 10.1109/ICECTA48151.2019.8959617.

[38] Z. Xiao, Z. Zhou, W. Yang y C. Deng, «An approach for SQL injection detection based on behavior and response analysis,» en *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, 2017, págs. 1437-1442. DOI: 10.1109/ICCSN.2017.8230346.

[39] D. Kar, K. Agarwal, A. K. Sahoo y S. Panigrahi, «Detection of SQL injection attacks using Hidden Markov Model,» en *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, 2016, págs. 1-6. DOI: 10.1109/ICETECH.2016.7569180.

[40] L. Yan, X. Li, R. Feng, Z. Feng y J. Hu, «Detection Method of the Second-Order SQL Injection in Web Applications,» en *Proceedings of the Third International Workshop on Structured Object-Oriented Formal Language and Method - Volume 8332*, Berlin, Heidelberg: Springer-Verlag, 2013, págs. 154-165, ISBN: 9783319049144. DOI: 10.1007/978-3-319-04915-1_11. dirección: https://doi.org/10.1007/978-3-319-04915-1_11.

[41] Z. C. S. S. Hlaing y M. Khaing, «A Detection and Prevention Technique on SQL Injection Attacks,» en *2020 IEEE Conference on Computer Applications(ICCA)*, 2020, págs. 1-6. DOI: 10.1109/ICCA49400.2020.9022833.

[42] P. Li, L. Liu, J. Xu et al., «Application of Hidden Markov Model in SQL Injection Detection,» en *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, 2017, págs. 578-583. DOI: 10.1109/COMPSAC.2017.64.

[43] T. Oosawa y T. Matsuda, «SQL injection attack detection method using the approximation function of zeta distribution,» en *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, págs. 819-824. DOI: 10.1109/SMC.2014.6974012.

[44] K. Wang e Y. Hou, «Detection method of SQL injection attack in cloud computing environment,» en *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2016, págs. 487-493. DOI: 10.1109/IMCEC.2016.7867260.

[45] Y.-C. Chung, M.-C. Wu, Y.-C. Chen y W.-K. Chang, «A Hot Query Bank approach to improve detection performance against SQL injection attacks,» *Computers & Security*, vol. 31, n.º 2, págs. 233-248, 2012, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2011.11.007. dirección: https://www.sciencedirect.com/science/article/pii/S016740481100143X.

[46] P. Kumar, «The multi-tier architecture for developing secure website with detection and prevention of sql-injection attacks,» *International Journal of Computer Applications*, vol. 62, n.º 9, 2013.

[47] D. Chen, Q. Yan, C. Wu y J. Zhao, «SQL Injection Attack Detection and Prevention Techniques Using Deep Learning,» *Journal of Physics: Conference Series*, vol. 1757, n.º 1, pág. 012 055, ene. de 2021. DOI: 10.1088/1742-6596/1757/1/012055. dirección: https://doi.org/10.1088/1742-6596/1757/1/012055.

[48] G. Singh, D. Kant, U. Gangwar y A. P. Singh, «Sql injection detection and correction using machine learning techniques,» en *Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) Volume 1*, Springer, 2015, págs. 435-442.

[49] C.-c. Shi, T. Zhang, Y. Yu y W. Lin, «A new approach for SQL-injection detection,» en *Instrumentation, Measurement, Circuits and Systems*, Springer, 2012, págs. 245-254.

[50] R. M. Nadeem, R. M. Saleem, R. Bashir y S. Habib, «Detection and prevention of SQL injection attack by dynamic analyzer and testing model,» *International Journal of Advanced Computer Science and Applications*, vol. 8, n.º 8, págs. 209-214, 2017.

[51] L. Xiao, S. Matsumoto, T. Ishikawa y K. Sakurai, «SQL Injection Attack Detection Method Using Expectation Criterion,» en *2016 Fourth International Symposium on Computing and Networking (CANDAR)*, 2016, págs. 649-654. DOI: 10.1109/CANDAR.2016.0116.

[52] R. M. Nadeem, R. M. Saleem, R. Bashir y S. Habib, «Detection and Prevention of SQL Injection Attack by Dynamic Analyzer and Testing Model,» *International Journal of Advanced Computer Science and Applications*, vol. 8, n.º 8, 2017. DOI: 10.14569/IJACSA.2017.080827. dirección: http://dx.doi.org/10.14569/IJACSA.2017.080827.

[53] M. S. Aliero e I. Ghani, «A component based SQL injection vulnerability detection tool,» en *2015 9th Malaysian Software Engineering Conference (MySEC)*, 2015, págs. 224-229. DOI: 10.1109/MySEC.2015.7475225.

[54] H. Zhang, B. Zhao, H. Yuan, J. Zhao, X. Yan y F. Li, «SQL Injection Detection Based on Deep Belief Network,» en *Proceedings of the 3rd International Conference on Computer Science and Application Engineering*, ép. CSAE 2019, Sanya, China: Association for Computing Machinery, 2019, ISBN: 9781450362948. DOI: 10.1145/3331453.3361280. dirección: https://doi.org/10.1145/3331453.3361280.

[55] G. Bafghi, «A Simple and Fast Technique for Detection and Prevention of SQL Injection Attacks (SQLIAs),» *International Journal of Security and Its Applications*, vol. 7, n.º 5, págs. 53-66, 2013.

[56] Sangeeta, S. Nagasundari y P. B. Honnavali, «SQL Injection Attack Detection using ResNet,» en *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2019, págs. 1-7. DOI: 10.1109/ICCCNT45670.2019.8944874.

[57] T.-Y. Wu, J.-S. Pan, C.-M. Chen y C.-W. Lin, «Towards SQL injection attacks detection mechanism using parse tree,» en *Genetic and Evolutionary Computing*, Springer, 2015, págs. 371-380.

[58] L. Saoudi, K. Adi e Y. Boudraa, «A rejection-based approach for detecting SQL injection vulnerabilities in web applications,» en *International Symposium on Foundations and Practice of Security*, Springer, 2019, págs. 379-386.

[59] R. Kozik, M. Choraś y W. Hołubowicz, «Hardening Web Applications against SQL Injection Attacks Using Anomaly Detection Approach,» en *Image Processing & Communications Challenges 6*, Springer, 2015, págs. 285-292.

[60] N. M. Sheykhkanloo, «A Pattern Recognition Neural Network Model for Detection and Classification of SQL Injection Attacks,» *International Journal of Computer and Information Engineering*, vol. 9, n.º 6, págs. 1436-1446, 2015, ISSN: eISSN: 1307-6892. dirección: https://publications.waset.org/vol/102.

[61] O. Hubskyi, T. Babenko, L. Myrutenko y O. Oksiiuk, «Detection of sql injection attack using neural networks,» en *International scientific-practical conference*, Springer, 2020, págs. 277-286.

[62] D. E. Nofal y A. A. Amer, «SQL Injection Attacks Detection and Prevention Based on Neuro-Fuzzy Technique,» en *International Conference on Advanced Intelligent Systems and Informatics*, Springer, 2019, págs. 722-738.

[63] N. Gandhi, J. Patel, R. Sisodiya, N. Doshi y S. Mishra, «A CNN-BiLSTM based Approach for Detection of SQL Injection Attacks,» en *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, 2021, págs. 378-383. DOI: 10.1109/ICCIKE51210.2021.9410675.

[64] R. A. Dalimunthe y S. Sahren, «Intrusion detection system and modsecurity for handling sql injection attacks,» en *International Conference on Social, Sciences and Information Technology*, vol. 1, 2020, págs. 187-194.

[65] A. O. Agbakwuru y D. O. Njoku, «SQL Injection Attack on Web Base Application: Vulnerability Assessments and Detection Technique,» *International Research Journal of Engineering and Technology*, vol. 8, n.º 3, págs. 243-252, 2021.

8 ANEXOS

Tabla 8.1: Artículos seleccionados para la revisión

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
1	[16]	166	2012	Técnica basada en Removing Sql query attribute values	Machine learning	—	— SQL 5.0	—
2	[17]	75	2013	Técnica basada en idMas-SQL	Machine learning	CNN	Base de datos SQL 5.0	Todos
3	[18]	66	2014	Técnica basada en Data-mining based Sql injection attacj deteccion in internal query tress	Machine learning	CNN	PostgreSql v 9.2.3	Todos
4	[19]	52	2012	Técnica basada en Detection of SQL injection attacks	Machine learning	CNN	PostgreSql v 9.2.3	Todos
5	[20]	43	2016	Técnica basada en análisis estático	Prácticas de codificación segura	N/A	N/A	Todos
6	[21]	42	2016	Técnica basada en basada ASCII based string matching	Técnicas híbridas	String Matching	Generador de claves basado en texto, graficos SQL utilizando FMS	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
7	[22]	38	2020	Técnica basada en Machine learning predictive analytics to SQL injection attac	Machine Learning	SVM	N/A	Todos
8	[23]	28	2020	Técnica basada en Artificial neural network	Machine Learning	CNN	Generador de URL	Todos
9	[24]	27	2019	Técnica basada en Adaptive Deep Forest	Machine Learning	AdaBoost	Exploit-Db y wooyun-Db	Todos
10	[25]	27	2019	Técnica basada en Neural networks	Machine Learning	CNN	Generador de URL	Todos
11	[26]	26	2015	Técnica basada en SALiDDs	Técnicas taint-based	K-meas	N/A	Todos
12	[27]	24	2017	Técnica basada en análisis estático y dinámico	Técnicas Hibridas	N/A	N/A	Todos
13	[28]	24	2019	Técnica basada en Elastic pooling - CNN	Machine learning	CNN	Registros de web reales en entorno de producción	Todos
14	[29]	23	2012	Técnica basada en Progam tracing and machine learning	Técnicas híbridas	N/A	N/A	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
15	[30]	23	2019	Técnica basada en Diava	Modificación de sentencias	N/A	Almacenamiento en la nube, análisis de trafico de red	Todos
16	[31]	22	2019	Técnica basada en Removing the parameter values of SQL query	Modificación de sentencias	N/A	Aplicaciones web vulnerables	Todos
17	[32]	22	2019	Técnica basada en Machine learning for SQL injection detection	Modificación de sentencias	N/A	Aplicaciones web vulnerables	Todos
18	[33]	17	2020	Técnica basada en Ontology to cloud web security	Practicas de codificación segura	N/A	Información guardada en la nube	Todos
19	[34]	16	2014	Técnica basada en análisis estático y dinámico	Técnicas híbridas	N/A	N/A	Todos
20	[35]	15	2017	Técnica basada en redes neuronales	Machine learning	CNN	Generador y clasificador de URLs	Todos
21	[36]	14	2012	Técnicas basadas en filtrado de atributos	Prácticas de códificación	N/A	N/A	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
22	[37]	14	2019	Técnica basada en clasificadores de machine learning	Machine learning	23 clasificadores	Ejemplos de sentencias SQL de W3School (benignos) y sentencias del OWASP SecLists Project	Todos
23	[38]	13	2017	Técnica basada en analisis de comportamiento	Otras técnicas	N/A	N/A	Solo los 6 tipos de SQLIA más básicos
24	[39]	13	2016	Técnica basada en el modelo oculto de Markov	Técnicas basadas en modelos probabilísticos	HMM	Datos reales de una configuración de prueba	Todos
25	[40]	13	2014	Tecnica basada en análisis estático y dinamico	Técnicas híbridas	N/A	N/A	Todos
26	[41]	12	2020	Técnica basada en creacion de lexicos y tokenizacion de cadenas	Árbol de análisis gramatical	N/A	N/A	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
27	[42]	12	2017	Técnica basada en el modelo oculto de Markov	Técnicas basadas en modelos probabilísticos	HMM	Datos reales de una configuración de prueba	Todos
28	[43]	12	2014	Técnica basada en la funcion de distribución Zeta	Técnicas basadas en modelos probabilísticos	N/A	Datos de ejemplo	Todoss
29	[44]	12	2016	Técnica basada en creación de reglas	Técnicas taint-based	N/A	N/A	Todos
30	[45]	12	2012	Técnica asado en creacioón de banco de consultas	Árbol de análisis gramatical	N/A	N/A	Todos
31	[46]	11	2013	Tecnica basada en análisis estático y dinamico	Técnicas híbridas	N/A	N/A	Todos
32	[47]	11	2021	Técnica basada en deep learning	Machine learning	CNN y MLP	Datos de ejemplo obtenidos de internet	Todos
33	[48]	11	2015	Técnica basada en machine learning	Machine learning	K-means	No especifica	Todos
34	[49]	10	2015	Técnica basada en creación de librerias de conocimiento	Árbol de análisis gramatical	N/A	N/A	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
35	[50]	9	2017	Técnica basada en Dynamic Analyzer and Testing Model	Taint-based Technique	N/A	datos reales de una configuración de prueba	Todos
36	[51]	9	2016	Técnica basada en Expectation Criterion	Probabilístico	N/A	datos de ejemplo	Todos
37	[52]	9	2013	Técnica basada en la detección del lado del cliente utilizando cuatro métricas de entropía condicional	Practicas de Codificación Segura	N/A	datos reales de una configuración de prueba	Todos
38	[53]	8	2015	Técnica basada en herramientas de detección de vulnerabilidades basada en Rastreo de la web, análisis de los ataques y elaboración de informes, análisis de los ataques y elaboración de informes	Análisis Estático	N/A	datos reales de una configuración de prueba	Todos
39	[54]	7	2019	Técnica basada en Deep Belief Network	Machine Learning	Deep Belief Network (DBN)	datos de ejemplo	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
40	[55]	6	2013	Técnica basada en modelos de consulta válidos obtenidos de una aplicación web	Análisis Estático y Dinámico	N/A	datos reales de una configuración de prueba	Todos
41	[56]	6	2019	Técnica basada en ResNet	Machine Learning	ResNet	uso de una herramienta (no especifica) y datos de internet	Todos
42	[57]	6	2015	Técnica basada en Dynamic SQLIAs Detection (DSD)	Parse Tree	Dynamic SQLIAs Detection (DSD)	datos reales de una configuración de prueba	Todos
43	[58]	5	2019	Técnica basada en rechazo	Análisis Estático	N/A	datos reales de una configuración de prueba	Todos
44	[59]	4	2015	Técnica basada en anomalias de rechazo	Analisis Estático	Linear Discriminant Analysis(LDA)	datos generados por un servicio HTTP	todos
45	[60]	4	2015	Técnica basada en una Red Neuronal	Machine Learning	Red Neuronal	datos de ejemplo	Todos
46	[61]	4	2020	Técnica basada en Artificial Neural Networks	Machine Learning	Artificial Neural Networks	datos obtenidos de sitios de internet	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
47	[62]	3	2019	Técnica basada en Neuro-Fuzzy	Machine Learning	Adaptive Neuro-Fuzzy Inference System (ANFIS) / Fuzzy C-Means (FCM) / ScaledConjugate Gradient (SCG)"	datos reales de una configuración de prueba	Todos
48	[63]	1	2021	Técnica basada en CNN-BiLSTM	Machine Learning	CNN-BiLSTM	datos obtenidos de sitios de internet	Todos
49	[64]	1	2020	Técnica basada en un Sistema de Detección de Intrusos y Cortafuegos(ModSecurity)	Sistema de detección de intrusos	N/A	Datos de ejemplo	Todos
50	[65]	1	2021	Técnica basada en fuzzy rule-based classification system (FRBCS)	Machine Learning	Algoritmo Genético Simple	datos reales de una configuración de prueba	Todos

Tabla 8.2: Puntaje de la evaluación de calidad de los artículos analizados .

#	QA1	QA2	QA3	QA4	QA5	QA6	QA7	Total
[16]	1	1	1	1	1	1	1	7
[17]	1	1	1	1	1	1	0	6
[18]	1	1	1	1	1	0.5	0	5.5
[19]	1	1	1	1	1	1	0	6
[20]	1	1	1	1	1	1	1	7
[21]	1	1	1	1	1	1	0	6
[22]	1	0.5	0.5	1	1	0.5	1	5.5
[23]	1	1	1	1	1	1	1	7
[24]	1	1	1	1	1	1	0	6
[25]	1	1	1	0	1	1	0.5	5.5
[26]	1	1	1	1	1	1	1	7
[27]	1	1	1	1	1	1	1	7
[28]	1	1	1	1	1	1	0.5	6.5
[29]	1	1	1	1	1	1	0.5	6.5
[30]	1	1	1	1	1	1	0	6
[31]	1	1	0	1	1	1	0	5
[32]	1	1	1	1	1	1	1	7
[33]	1	1	1	1	1	1	1	7
[34]	1	1	0.5	1	1	1	1	6.5
[35]	1	1	1	1	1	1	0	6
[36]	1	1	0.5	1	1	1	1	6.5
[37]	1	1	1	1	1	1	1	7
[38]	0.5	1	1	1	1	1	1	6.5
[39]	1	0.5	1	1	1	1	1	6.5
[40]	1	1	1	1	1	1	1	7
[41]	1	1	0	1	1	0.5	0	4.5
[42]	1	1	1	1	1	1	1	7
[43]	1	1	0	1	1	1	1	6
[44]	1	1	0	1	1	1	0	5
[45]	1	1	1	1	1	1	1	7
[46]	1	1	1	1	1	0.5	1	6.5
[47]	1	1	0.5	1	1	1	1	6.5
[48]	1	1	0.5	1	1	0.5	1	6
[49]	1	1	0	1	1	0.5	1	5.5
[50]	1	1	0.5	1	1	1	0.5	6
[51]	1	0,5	1	0,5	1	1	1	6

#	QA1	QA2	QA3	QA4	QA5	QA6	QA7	Total
[52]	1	0	0	0.5	0	0.5	0.5	2.5
[53]	1	0.5	1	1	1	1	1	6.5
[54]	1	1	1	0.5	1	1	0	5.5
[55]	1	1	1	1	1	1	0.5	6.5
[56]	1	1	1	1	1	1	0	6
[57]	1	0.5	0.5	1	1	1	0.5	5.5
[58]	0.5	0.5	0	0.5	1	1	1	4.5
[60]	1	1	1	1	1	1	1	7
[61]	1	0.5	0.5	1	1	1	0	5
[62]	1	1	1	1	1	1	1	7
[63]	1	1	1	1	1	1	1	7
[64]	1	0.5	0	1	1	0	0	3.5
[65]	1	1	1	1	1	1	0.5	6.5