Análisis de vulnerabilidades de tipo SQL Injection

Autor
Danilo Alfonso Timaran Hernandez
Wilson Andres Quinbien tero Cardona

Ingeniero de sistemas

Director
Gustavo Adolfo Alomia Peñafiel

Facultad de ingeniería
Ingeniería de sistemas
Universidad Santiago de Cali
Santiago de Cali - Colombia
2026

Análisis de vulnerabilidades de tipo SQL Injection

Analysis of SQL Injection vulnerabilities in web applications

Danilo Alfonso Timaran[1]
Danilo.timaran00@usc.edu.co

Wilson Andres Quintero Cardona[1]
Wilson.quintero00@usc.edu.co

Universidad Santiago de Cali, Facultad de Ingeniería, Programa de Ingeniería de Sistemas (1)

Resumen

Las vulnerabilidades de tipo SQL Injection representan una de las amenazas más persistentes para la seguridad de las aplicaciones web, a pesar de los avances en prácticas de desarrollo seguro y tecnologías de protección. El objetivo de este estudio es analizar la evolución, el impacto y las principales estrategias de detección y prevención de SQL Injection a partir de una revisión exploratoria de la literatura publicada entre 2018 y 2025. La metodología incluyó la búsqueda y selección de artículos revisados por pares en bases de datos especializadas, aplicando criterios de inclusión y exclusión definidos. Se analizaron dieciocho estudios, organizados según enfoques de detección, prevención, automatización y vacíos de investigación. Los resultados evidencian un predominio de modelos basados en aprendizaje automático y profundo, los cuales reportan altos niveles de precisión en la detección de ataques, así como la vigencia de las buenas prácticas de codificación segura como base de la prevención. No obstante, se identifican limitaciones relacionadas con la generalización de los modelos, la robustez adversarial y la implementación en sistemas heredados. Se concluye que la defensa multicapa, que integra desarrollo seguro, detección inteligente y monitoreo continuo, constituye la estrategia más efectiva para mitigar esta vulnerabilidad, y se destacan líneas futuras de investigación orientadas a contextos reales de aplicación.

Palabras Clave: SQL Injection, Vulnerabilidades, Ciberataques

Abstract

SQL Injection vulnerabilities is one of the most persistent threats to web application security, despite advances in secure development practices and protection technologies. The aim of this study was to analyze the evolution, impact, and main detection and prevention strategies related to SQL Injection through an exploratory review of literature published between 2018 and 2025. The methodology involved searching and selecting peer-reviewed articles from specialized databases, applying predefined inclusion and exclusion criteria. Eighteen studies were analyzed and organized according to detection approaches, prevention strategies, automation mechanisms, and research gaps. The results reveal a predominance of machine learning and deep learning–based models, which report high detection accuracy, as well as the continued relevance of secure coding practices as a foundational preventive measure. However, limitations related to model generalization, adversarial robustness, and deployment in legacy systems were identified. It is concluded that a multi-layered defense strategy, integrating secure development, intelligent detection, and continuous monitoring, represents the most effective approach to mitigating SQL Injection vulnerabilities, while future research should focus on real-world applicability and resilience.

Keywords: SQL Injection, Vulnerabilities, Cyber Attacks

1. INTRODUCCIÓN

El avance tecnológico de la última década ha generado que la información de un gran número de compañías a nivel mundial sea almacenada en sistemas web que permiten un acceso ágil y seguro. La universalización de los sistemas de la información exige que la seguridad informática asuma un rol más importante, debido a la consecuente exposición de los datos y al riesgo que supone la exposición de estos. Esta es la razón por la que los diferentes actores implicados en el sistema tecnológico desarrollan medidas de alarma y monitoreo para el constante control de dichos aspectos.

Bajo ese entendido, la vulnerabilidad SQL (*Structured Query Language*) *Injection* es una amenaza seria para la seguridad informática. Esta se define como un ataque que explota fallos en sistemas, aplicaciones o redes para acceder, modificar o destruir información sin autorización (OWASP, 2021) y surge cuando un atacante logra insertar código SQL malicioso a través de formularios web expuestos públicamente, engañando al intérprete para ejecutar comandos no autorizados o acceder a información sensible (Halfond et al., 2006). El impacto de estos ataques puede ser grande, comprometiendo la confidencialidad, integridad y disponibilidad de los datos en diversos contextos empresariales. Según Acunetix (2019), el 8% de las aplicaciones web analizadas eran vulnerables a ataques de SQL *Injection*, subrayando la persistencia de este problema. El análisis de esta vulnerabilidad en cuanto a su origen, evolución e impacto es fundamental para identificar y proponer medidas de prevención.

La creciente dependencia de las organizaciones respecto a aplicaciones web para la gestión de información crítica ha incrementado significativamente la exposición a amenazas de seguridad informática. Entre estas, las vulnerabilidades de tipo SQL Injection son algunas de las más críticas, a pesar de los avances en tecnologías de protección y en buenas prácticas de desarrollo de software. Diversos reportes y estudios académicos coinciden en que este tipo de vulnerabilidad continúa siendo una de las principales causas de brechas de seguridad en sistemas de información, afectando la confidencialidad, integridad y disponibilidad de los datos.

En ese sentido, es necesario consolidar y analizar el conocimiento existente sobre SQL Injection, su origen, evolución y las estrategias propuestas para su detección y prevención. Si bien existe una amplia cantidad de literatura que aborda esta problemática desde diferentes enfoques técnicos, no siempre se presenta de forma integrada ni accesible para estudiantes, desarrolladores y profesionales del área de la ingeniería de sistemas. Esto dificulta el análisis global del fenómeno y la adopción de medidas efectivas de mitigación en entornos reales. De esta forma, el objetivo de este trabajo es analizar la literatura científica publicada entre 2018 y 2025 sobre vulnerabilidades de tipo SQL Injection en aplicaciones web para identificar las principales estrategias de detección y prevención asociadas a esta vulnerabilidad.

Este trabajo se justifica entonces en la necesidad de realizar una revisión bibliográfica que permita sintetizar los principales aportes teóricos y prácticos relacionados con las vulnerabilidades de tipo SQL Injection, identificando patrones comunes, enfoques predominantes y vacíos en la literatura. A través de este análisis, se busca brindar una visión estructurada que facilite la toma de decisiones informadas en el diseño, desarrollo y aseguramiento de aplicaciones web.

Para cumplir su objetivo, el documento se estructura de la siguiente manera: en primer lugar, se presenta la introducción, en la cual se contextualiza la problemática y se justifica la relevancia del estudio. Posteriormente, se desarrolla el apartado SQL Injection y sus orígenes, donde se describen los fundamentos conceptuales y la evolución de esta vulnerabilidad. A continuación, se expone la metodología, detallando el enfoque de revisión bibliográfica, las fuentes de información y los criterios de selección utilizados. Seguidamente, se presentan los resultados y el análisis de la revisión de la literatura, organizados en categorías temáticas que permiten sintetizar lo más importante. Finalmente, se formulan las conclusiones, en las cuales se integran los resultados obtenidos.

2. OBJETIVOS

2.1. Objetivo general

Analizar la literatura científica publicada entre 2018 y 2025 sobre vulnerabilidades de tipo SQL Injection en aplicaciones web.

2.2. Objetivos específicos

1. Identificar los enfoques técnicos utilizados en la literatura científica para la detección de vulnerabilidades de tipo SQL Injection en aplicaciones web.
2. Caracterizar las estrategias de prevención y mitigación de SQL Injection reportadas en los estudios analizados.
3. Determinar los vacíos de investigación existentes en relación con la aplicabilidad, robustez y adaptación de las soluciones propuestas.

3. SQL INJECTION Y SUS ORÍGENES

La vulnerabilidad SQL *Injection* es una gran amenaza para los sistemas de información web que interactúan con bases de datos. Este tipo de ataque explota las debilidades en la programación de aplicaciones web, permitiendo a los atacantes manipular las consultas SQL para obtener acceso no autorizado a datos sensibles o realizar operaciones maliciosas en la base de datos (OWASP, 2021). Aunque existen medidas de seguridad como los certificados SSL para el cifrado de conexiones, la prevención de SQL Injection requiere un enfoque más específico centrado en el desarrollo seguro de aplicaciones. Los atacantes suelen utilizar herramientas de escaneo automatizadas para identificar vulnerabilidades potenciales, centrándose en puntos de entrada comunes como las pantallas de inicio de sesión (Halfond et al., 2006). Una vez detectada una vulnerabilidad, los atacantes pueden intentar explotarla mediante la inyección de código SQL malicioso, potencialmente obteniendo acceso no autorizado a información confidencial o incluso modificando o eliminando datos críticos.

La prevención de ataques SQL Injection es crucial y requiere la implementación de buenas prácticas de desarrollo seguro. Esto incluye la validación y sanitización adecuadas de las entradas del usuario, el uso de consultas parametrizadas o procedimientos almacenados y la aplicación del principio de mínimo privilegio en las conexiones a la base de datos (Shar & Tan, 2013). Además, es fundamental mantener actualizados los sistemas de gestión de bases de datos y aplicar parches de seguridad regularmente. La formación continua de los desarrolladores en técnicas de codificación segura y la realización de auditorías de seguridad periódicas son también elementos clave en la estrategia de prevención. Dada la sofisticación de los ciberdelincuentes y su capacidad para explotar incluso las vulnerabilidades más pequeñas, es esencial adoptar un enfoque proactivo y multifacético en la seguridad de las aplicaciones web para proteger eficazmente la información crítica de la empresa.

Dicho lo anterior, la prevención de ataques SQL Injection se convierte entonces en un reto para las organizaciones. Aunque la seguridad absoluta es inalcanzable, existen estrategias efectivas para mitigar estos riesgos. Un componente crucial de estas estrategias es la implementación de escaneos de vulnerabilidades regulares. Estos escaneos permiten a las organizaciones identificar y evaluar proactivamente las debilidades potenciales en sus sistemas antes de que puedan ser explotadas por atacantes (Antunes & Vieira, 2015). La frecuencia y el alcance de estos escaneos deben estar claramente definidos en las políticas de seguridad de la organización, con procesos establecidos para la priorización y remediación de las vulnerabilidades identificadas. Es importante destacar que, según el Proyecto de Seguridad de Aplicaciones Web Abiertas (OWASP, 2021), las vulnerabilidades de tipo SQL Injection consistentemente se ubican entre las amenazas más críticas, lo que subraya la necesidad de abordar estas vulnerabilidades con urgencia cuando se detectan.

Asimismo, la implementación de prácticas de codificación segura se consolida como otro pilar fundamental en la prevención de ataques SQL Injection. Este enfoque requiere un compromiso continuo con la formación y actualización de los equipos de desarrollo. Los desarrolladores deben estar bien versados en las mejores prácticas de seguridad, incluyendo técnicas como la parametrización de consultas, el uso de procedimientos almacenados y la implementación de controles de entrada rigurosos (Shar & Tan, 2013). Además, la adopción de marcos de desarrollo que incorporan características de seguridad por defecto puede proporcionar una capa adicional de protección. La educación continua y la certificación en seguridad de aplicaciones web para los desarrolladores no solo mejoran la calidad del código producido, sino que también fomentan una cultura de seguridad dentro de la organización (Geer, 2015).

A pesar de lo anterior, es menester adoptar un enfoque holístico que integre la seguridad en todo el ciclo de vida del desarrollo de software (SDLC). Esto incluye la realización de revisiones de código regulares, pruebas de penetración y el uso de herramientas de análisis estático y dinámico de código. La implementación de un proceso de Desarrollo y Operaciones de Seguridad (DevSecOps) puede facilitar la integración continua de prácticas de seguridad en el proceso de desarrollo y despliegue. Además, la adopción de un modelo de seguridad de defensa en profundidad, que incluya múltiples capas de controles de seguridad, puede proporcionar una protección más robusta contra ataques SQL Injection y otras amenazas cibernéticas (McGraw, 2006). Este enfoque multifacético no solo mejora la postura de seguridad de la

organización, sino que también contribuye a la creación de una infraestructura de tecnología de información más resiliente y adaptable a las amenazas emergentes.

4. METODOLOGÍA

Tipo de estudio

Este trabajo se enmarca bajo una revisión exploratoria de la literatura, cuyo propósito es identificar, analizar y sintetizar la evidencia científica reciente relacionada con las vulnerabilidades de tipo SQL Injection, así como las principales técnicas de detección, prevención y mitigación reportadas en aplicaciones web. Este enfoque se seleccionó en función de los estudios existentes, la evolución de las amenazas de seguridad y la diversidad de soluciones técnicas propuestas en el ámbito de la ingeniería de sistemas y la ciberseguridad.

Fuentes de información y estrategia de búsqueda

La búsqueda bibliográfica se realizó en bases de datos científicas reconocidas en ingeniería y seguridad informática: Scopus, IEEE Xplore, Web of Science y ACM Digital Library. Las búsquedas se llevaron a cabo mediante combinaciones de términos clave relacionados con SQL Injection, detección, prevención, aprendizaje automático y seguridad de aplicaciones web, empleando operadores booleanos y filtros por tipo de documento.

Periodo de análisis y criterios de selección

Con respecto a los criterios, se consideraron publicaciones comprendidas entre 2018 y 2025, con el fin de capturar investigaciones recientes que reflejen los avances tecnológicos actuales, incluyendo enfoques basados en aprendizaje automático, sistemas híbridos y prácticas DevSecOps. Estudios previos a este periodo fueron totalmente excluidos. Los criterios de inclusión contemplaron artículos revisados por pares, con enfoque explícito en SQL Injection y disponibilidad de texto completo. Además, se incluyeron estudios en inglés y en español. Por último, se excluyeron documentos sin metodología clara, fuentes no académicas y estudios duplicados.

Proceso de selección y análisis

El proceso de selección se desarrolló en fases de identificación, cribado y elegibilidad, lo que permitió depurar un conjunto final de estudios para el análisis. La información extraída de los estudios seleccionados fue sintetizada mediante un análisis cualitativo de tipo categorial, organizando los resultados en cuatro categorías analíticas: (a) detección basada en aprendizaje automático y profundo, (b) estrategias de prevención fundamentadas en buenas prácticas de codificación segura, (c) enfoques híbridos que integran firewalls de aplicaciones web con técnicas inteligentes, y (d) sistemas automatizados de monitoreo y respuesta, así como la identificación de vacíos de investigación.

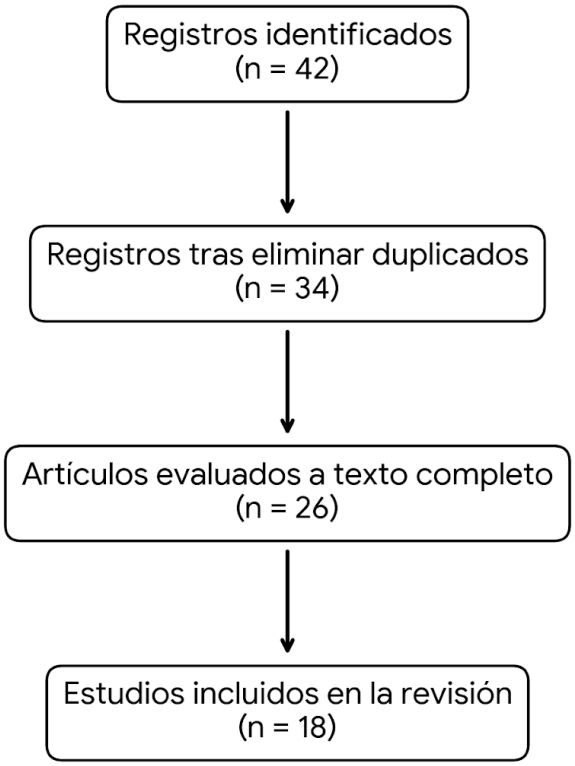
La categorización empleada en este estudio se fundamenta en clasificaciones propuestas en la literatura reciente sobre seguridad de aplicaciones web, donde se distinguen enfoques basados en detección inteligente, prevención mediante prácticas de desarrollo seguro, mecanismos híbridos y sistemas automatizados de defensa (Shar & Tan, 2013; McGraw, 2006; Paul et al., 2024; Salmawi, 2025). Estas categorías permiten estructurar la evidencia disponible y facilitar la comparación entre las distintas soluciones técnicas reportadas en la literatura.

4. **RESULTADOS**

Como resultado del proceso de búsqueda, cribado y selección descrito en la metodología, se identificaron estudios científicos publicados entre 2018 y 2025 que abordan la vulnerabilidad de tipo SQL Injection desde diversos enfoques técnicos y metodológicos. Tras aplicar los criterios de inclusión y exclusión, se seleccionaron 18 estudios para el análisis final, los cuales son el corpus de esta revisión.

La Figura 1 presenta el proceso de identificación, filtrado y selección de los estudios incluidos, evidenciando una depuración progresiva de la literatura hasta conformar el conjunto final analizado.

Figura 1. Proceso de identificación y selección de estudios incluidos en la revisión (2018–2025).



Fuente propia

Como se observa en la Figura 1, a partir de un conjunto inicial de estudios identificados, el proceso de cribado y elegibilidad permitió depurar progresivamente la literatura hasta conformar un corpus final de 18 estudios, los cuales cumplieron con los criterios definidos para esta revisión. A continuación, se presenta la caracterización de los estudios.

3.1 Caracterización general de los estudios incluidos

Los estudios seleccionados corresponden principalmente a artículos publicados en revistas científicas indexadas y actas de conferencias internacionales especializadas en seguridad informática, ingeniería de software y sistemas de información. En términos temporales, la mayor concentración de publicaciones se observa entre 2020 y 2025, lo que muestra un interés creciente en el desarrollo de mecanismos avanzados para la detección y prevención de ataques SQL Injection. A continuación se presentan los estudios incluídos en la tabla 1.

Tabla 1. Caracterización de los estudios incluidos en la revisión

Autor(es)	Año	Objetivo principal	Enfoque
Tang et al.	2020	Detectar SQL Injection mediante redes neuronales artificiales	ML
Xie et al.	2019	Detección de SQL Injection basada en CNN	ML
Paul et al.	2024	Detección, priorización y prevención de SQL Injection	Prevención
Coscia et al.	2024	Mejora de WAF mediante gramáticas proxy (PROGESI)	WAF
Muduli et al.	2024	Red neuronal para detección avanzada de SQL Injection	ML
Zhang et al.	2024	Prevención de inyecciones SQL de segundo orden	Híbrido
Dawadi & Srivastava	2023	WAF basado en deep learning	WAF
Fadlil et al.	2024	Mitigación de SQL Injection basada en OWASP	Prevención
Meshram	2025	Sistema automatizado de detección y prevención de SQLi	Automatizado
Abubakar et al.	2025	Detección SQLi mediante redes neuronales	ML
Akram et al.	2025	Técnica híbrida contra ataques SQL	Híbrido
Gan et al.	2025	Sistema integrado de detección y prevención basado en ML	ML
Salmawi	2025	Evaluación crítica de medidas de seguridad SQLi	Revisión
Kakisim	2024	Deep learning multivista para detección de SQLi	ML
Mohamed	2024	Detección y mitigación SQLi en aplicaciones de transporte	ML
Markulin et al.	2025	Uso de IA generativa para diseño resistente a SQLi	IA aplicada
Ikramaputra et al.	2025	WAF con ensemble de aprendizaje automático	WAF
Mallah & Quintero	2025	Robustez adversarial en detección SQLi basada en ML	Crítico

Fuente propia

Como se peude observar, desde el punto de vista del enfoque técnico, los estudios pueden agruparse en cuatro categorías principales:
(a) modelos de detección basados en aprendizaje automático y profundo,
(b) estrategias de prevención fundamentadas en buenas prácticas de codificación segura,
(c) mecanismos híbridos que integran firewalls de aplicaciones web con técnicas inteligentes, y
(d) sistemas automatizados de monitoreo y respuesta en tiempo real.

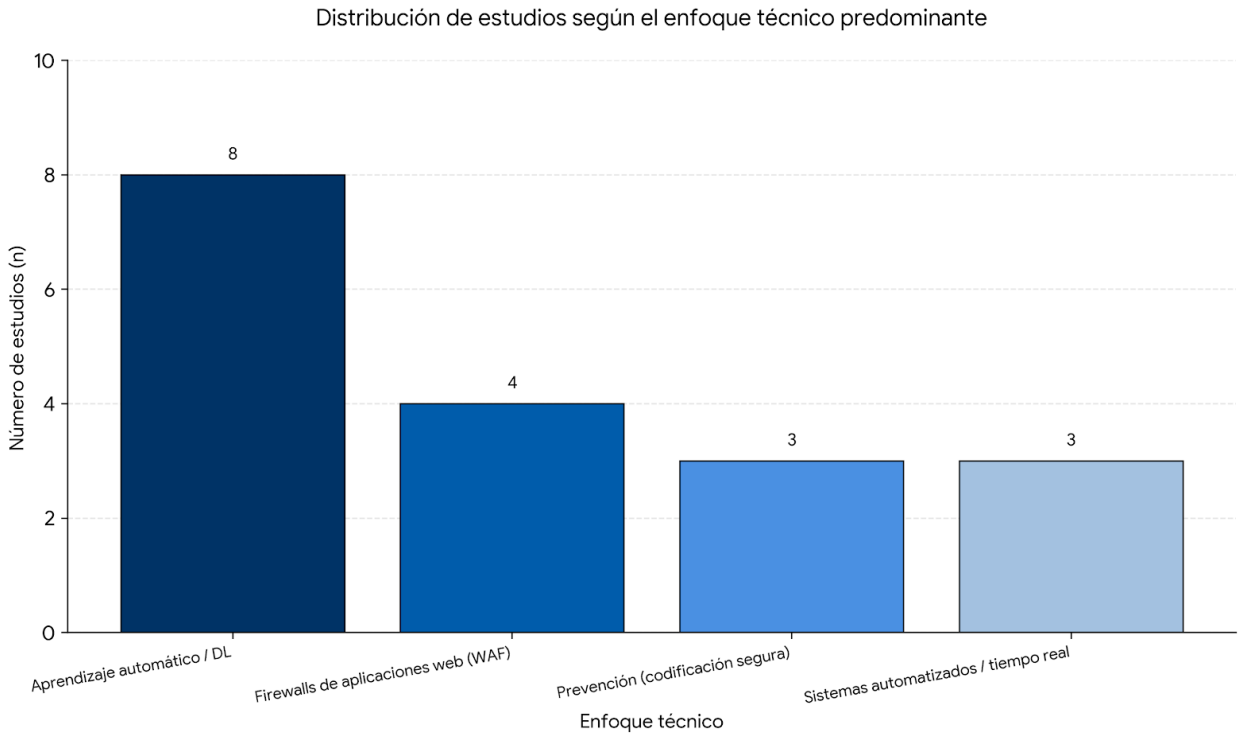
3.2 Técnicas de detección de vulnerabilidades SQL Injection

Los resultados de la revisión muestran un predominio de investigaciones centradas en técnicas de detección basadas en aprendizaje automático y profundo. Diversos estudios reportan niveles de precisión superiores al 97 % en la identificación de ataques SQL Injection, superando a los enfoques tradicionales basados en reglas y firmas (Tang et al., 2020; Muduli et al., 2024; Kakisim, 2024; Abubakar et al., 2025).

Asimismo, modelos como redes neuronales convolucionales, arquitecturas híbridas CNN-LSTM y enfoques basados en transformadores han demostrado una alta capacidad para identificar patrones complejos y ataques ofuscados (Xie et al., 2019; Zhang et al., 2024; Gan et al., 2025). Sin embargo, la literatura plantea que estos modelos dependen de conjuntos de datos representativos y actualizados, lo que pone sobre la mesa retos asociados al sobreajuste y a la generalización en entornos reales.

La Figura 2 ilustra la distribución de los estudios incluidos según el enfoque técnico predominante, lo que evidencia la alta representación de modelos basados en aprendizaje automático.

Figura 2. Distribución de los estudios incluidos según el enfoque técnico predominante.



Fuente propia

Como se presenta en la Figura 2, los estudios incluidos en la revisión se distribuyen principalmente en enfoques basados en aprendizaje automático y profundo, seguidos por soluciones apoyadas en firewalls de aplicaciones web, estrategias de prevención mediante buenas prácticas de codificación segura y sistemas de monitoreo automatizado. Esta distribución da cuenta de las principales líneas de investigación abordadas en la literatura reciente sobre SQL Injection.

3.3 Estrategias de prevención y mitigación

En cuanto a la prevención, los estudios coinciden en señalar que las buenas prácticas de codificación segura continúan

siendo un pilar fundamental para mitigar ataques SQL Injection. El uso de consultas parametrizadas, validación estricta de entradas y correcta gestión de privilegios se caracteriza como una defensa ampliamente aceptada (Paul et al., 2024; Fadlil et al., 2024; Salmawi, 2025).

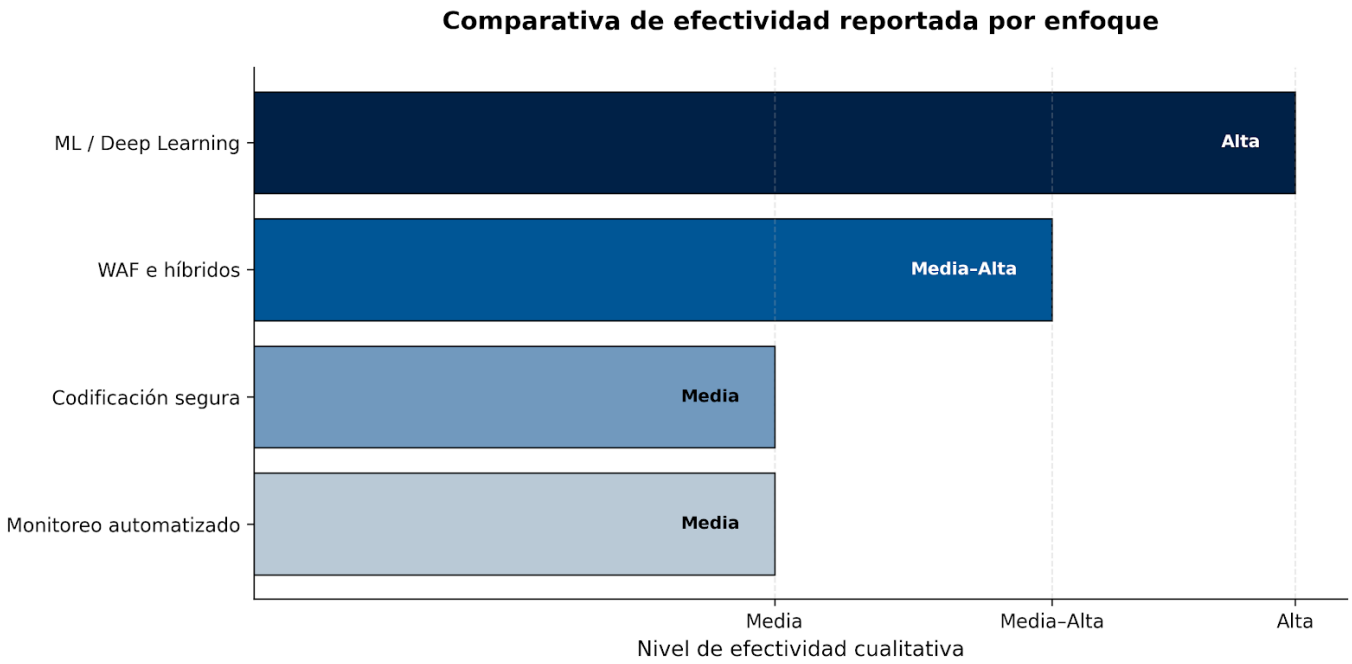
De igual forma, los firewalls de aplicaciones web han evolucionado hacia enfoques híbridos que incorporan aprendizaje automático para bloquear ataques mutados y de tipo cero día. Soluciones como PROGESI y WAFs con modelos ensemble muestran mejoras significativas en la detección de ataques avanzados (Coscia et al., 2024; Dawadi & Srivastava, 2023; Ikramaputra et al., 2025).

3.4 Sistemas automatizados y monitoreo en tiempo real

Un subconjunto de estudios aborda la integración de monitoreo continuo y respuesta automatizada como parte de una defensa multicapa. Sistemas que combinan análisis de tráfico, priorización de riesgos y contención automática permiten reducir el impacto de ataques SQL Injection en entornos productivos (Meshram, 2025; Mohamed, 2024). A pesar de lo anterior, la literatura también señala que estos sistemas incrementan la complejidad operativa y requieren personal capacitado para su supervisión y mantenimiento continuo.

3.5 Comparación de los enfoques identificados

La Figura 3 presenta una comparación cualitativa de la efectividad reportada de los principales enfoques analizados.



Fuente propia

Los resultados indican que los modelos basados en aprendizaje automático alcanzan una alta precisión en escenarios controlados, mientras que las estrategias tradicionales siguen siendo fundamentales como base de seguridad. De esta forma, los estudios coinciden en que la defensa multicapa constituye la estrategia más efectiva frente a la diversidad de ataques de SQL Injection (Akram et al., 2025; Salmawi, 2025).

3.6 Vacíos de investigación identificados

A partir del análisis de los estudios, se identificaron vacíos relevantes en la literatura, tal como los presenta la tabla 2.

Tabla 2. Vacíos identificados en la investigación

Enfoque	Sistemas legacy	Robustez adversarial	Entornos con pocos recursos
ML / DL	Bajo	Bajo	Bajo
WAF	Medio	Bajo	Medio
Codificación segura	Medio	N/A	Medio
Sistemas automatizados	Bajo	Bajo	Bajo

Fuente propia

La Tabla 2 sintetiza los vacíos encontrados y destaca la limitada atención prestada a la implementación de mecanismos de defensa en sistemas legacy, entornos con recursos restringidos y la robustez adversarial de los modelos basados en aprendizaje automático (Mallah & Quintero, 2025; Akram et al., 2025). Esto evidencia la necesidad de investigaciones futuras orientadas a mejorar la adaptabilidad, la sostenibilidad y la integración de soluciones de seguridad en contextos reales de desarrollo y operación.

5. DISCUSIÓN

La revisión permitió analizar la literatura publicada entre 2018 y 2025 sobre vulnerabilidades de tipo SQL Injection, evidenciando tanto avances como limitaciones en las estrategias de detección y prevención. En concordancia con los resultados obtenidos, la discusión se estructura en torno a la comparación de los enfoques identificados, su aplicabilidad y los vacíos que aún existen según lo encontrado en la investigación.

En primer lugar, los resultados confirman que los modelos basados en aprendizaje automático y profundo han adquirido un papel principal en la detección de ataques SQL Injection. Diversos estudios como el de Tang et al. (2020), Muduli et al. (2024), Kakisim (2024) y Abubakar et al. (2025) reportan precisiones superiores al 97 %, lo que representa una mejora frente a los enfoques tradicionales basados en reglas y firmas. Esta superioridad se explica por la capacidad de los modelos para identificar tendencias complejas y adaptarse a variantes de ataque ofuscadas. No obstante, la literatura también reporta que dichos modelos dependen fuertemente de la calidad y diversidad de los conjuntos de datos utilizados para su entrenamiento, lo que limita su generalización en entornos productivos heterogéneos (Mallah & Quintero, 2025).

En contraste, las estrategias tradicionales de prevención, fundamentadas en buenas prácticas de codificación segura, continúan siendo muy recomendadas como primera línea de defensa. El uso de consultas parametrizadas, validación de entradas y una gestión adecuada de privilegios se mantiene como un estándar efectivo para prevenir la mayoría de los ataques SQL Injection conocidos (Paul et al., 2024; Fadlil et al., 2024; Salmawi, 2025). Estos enfoques presentan la ventaja de su simplicidad, bajo costo de implementación y aplicabilidad transversal, aunque son insuficientes frente a ataques avanzados cuando se emplean de manera aislada.

La literatura también resalta el papel de los firewalls de aplicaciones web y los enfoques híbridos, los cuales integran mecanismos tradicionales con modelos inteligentes para mejorar la detección de ataques mutados y de tipo cero día.

Soluciones como PROGESI y los WAF basados en ensemble de aprendizaje automático evidencian una mayor robustez frente a ataques complejos (Coscia et al., 2024; Ikramaputra et al., 2025). Sin embargo, su efectividad depende de una buena configuración y mantenimiento continuo, lo que puede ser un reto para organizaciones con recursos limitados.

Adicionalmente, los sistemas automatizados de monitoreo y respuesta en tiempo real emergen como una tendencia relevante en la literatura reciente. Estos sistemas permiten detectar, priorizar y contener ataques de manera inmediata, reduciendo el impacto operativo y la exposición de los datos (Meshram, 2025; Mohamed, 2024). A pesar de sus beneficios, la implementación de estas soluciones incrementa la complejidad de la infraestructura y exige personal especializado, lo que restringe su adopción generalizada.

Los resultados de la revisión permiten reforzar la idea de que no existe una solución única y universalmente aplicable para la mitigación de SQL Injection. Tal como se evidencia en la Figura 3, los enfoques más efectivos corresponden a estrategias de defensa multicapa, que combinan prácticas de desarrollo seguro, mecanismos de detección inteligente y monitoreo continuo (Akram et al., 2025; Salmawi, 2025).

Finalmente, la identificación de vacíos de investigación, sobre todo en relación con sistemas legacy, entornos con recursos restringidos y la robustez adversarial de modelos basados en aprendizaje automático, pone en evidencia la necesidad de investigaciones futuras orientadas a la adaptabilidad y sostenibilidad de las soluciones propuestas (Mallah & Quintero, 2025). Estos vacíos explican por qué, a pesar de los avances tecnológicos, SQL Injection sigue siendo una amenaza para el panorama actual de la ciberseguridad.

5. CONCLUSIONES

A partir del análisis de la literatura científica publicada entre 2018 y 2025 sobre vulnerabilidades de tipo SQL Injection en aplicaciones web, se concluye que esta amenaza continúa siendo un riesgo para la seguridad de los sistemas de información, a pesar de los avances en técnicas de detección y prevención reportados en la literatura reciente. El estudio permitió identificar, caracterizar y analizar de manera sistemática las principales estrategias abordadas por la comunidad académica, así como los vacíos de investigación existentes. A continuación, se presentan las conclusiones por objetivos.

En relación con el primer objetivo específico, orientado a identificar los enfoques técnicos utilizados para la detección de vulnerabilidades de tipo SQL Injection, los resultados muestran un predominio de modelos basados en aprendizaje automático y profundo. Estos enfoques reportan altos niveles de precisión en escenarios controlados y demuestran una mayor capacidad para detectar ataques ofuscados en comparación con métodos tradicionales basados en reglas y firmas. No obstante, la literatura señala que su efectividad depende en gran medida de la calidad de los datos de entrenamiento y de su capacidad de generalización en entornos reales.

Respecto al segundo objetivo específico, que estaba enfocado en caracterizar las estrategias de prevención y mitigación de SQL Injection, se concluye que las buenas prácticas de codificación segura continúan siendo una buena base de defensa. El uso de consultas parametrizadas, validación estricta de entradas y gestión adecuada de privilegios se mantiene como una estrategia recomendada y efectiva. Asimismo, los firewalls de aplicaciones web y los enfoques híbridos que integran técnicas inteligentes fortalecen la protección frente a ataques avanzados, aunque su implementación requiere configuraciones y mantenimiento continuo.

En cuanto al tercer objetivo específico, orientado a determinar los vacíos de investigación existentes, la revisión permitió identificar limitaciones relacionadas con la aplicabilidad de las soluciones propuestas en sistemas legacy, la robustez adversarial de los modelos basados en aprendizaje automático y su adaptación a entornos con recursos restringidos.

Los resultados de esta revisión confirman que la defensa multicapa, que integra prácticas de desarrollo seguro, mecanismos de detección inteligente y monitoreo continuo, constituye la estrategia más efectiva para mitigar vulnerabilidades de tipo SQL Injection. Finalmente, se resalta la necesidad de investigaciones futuras orientadas a mejorar la adaptabilidad, sostenibilidad y despliegue práctico de las soluciones de seguridad en contextos reales de operación.

REFERENCIAS

Abubakar, A., Bisallah, H., & Kabir, K. (2025). Enhancing web application security through automated SQL injection detection using neural networks. *UNIABUJA Journal of Engineering and Technology*. https://doi.org/10.70118/ujet.2025.0202.34

Acunetix. (2019). Acunetix web application vulnerability report 2019. https://www.acunetix.com/white-papers/acunetix-web-application-vulnerability-report-2019/

Akram, S., Arif, B., & Wani, S. (2025). The implications for a hybrid detection technique against malicious SQL attacks on web applications. *Journal of Information Systems Engineering and Management*. https://doi.org/10.52783/jisem.v10i35s.6219

Antunes, N., & Vieira, M. (2015). On the metrics for benchmarking vulnerability detection tools. In *Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (pp. 505–516). IEEE.

Coscia, A., Dentamaro, V., Galantucci, S., Maci, A., & Pirlo, G. (2024). PROGESI: A proxy grammar to enhance web application firewall for SQL injection prevention. *IEEE Access, 12*, 107689–107703. https://doi.org/10.1109/ACCESS.2024.3438092

Dawadi, B., & Srivastava, D. (2023). Deep learning technique-enabled web application firewall for the detection of web attacks. *Sensors, 23*(4), 2073. https://doi.org/10.3390/s23042073

Fadlil, A., Riadi, I., & Mu'min, M. (2024). Mitigation from SQL injection attacks on web server using OWASP framework. *International Journal of Engineering*. https://doi.org/10.5829/ije.2024.37.04a.06

Gan, Y., Miraz, M., Sazan, S., Bhuiyan, M., & Yew, T. (2025). Designing an integrated system for SQL injection detection and prevention using a machine learning-based web exploration application. *International Journal of Information Technology & Decision Making*. https://doi.org/10.1142/S0219622026500239

Geer, D. (2015). Six key areas of investment for the science of cybersecurity. The Futurist, 49(1), 10–15.

Halfond, W. G., Viegas, J., & Orso, A. (2006). A classification of SQL injection attacks and countermeasures. In *Proceedings of the IEEE International Symposium on Secure Software Engineering* (pp. 13–15). IEEE.

Ikramaputra, A., Sudarsono, A., & Ningsih, N. (2025). Enhancing web application firewall with ensemble machine learning to detect SQL injection attacks. In *Proceedings of the International Electronics Symposium* (pp. 521–526). IEEE. https://doi.org/10.1109/IES67184.2025.11160726

Kakisim, A. (2024). A deep learning approach based on multi-view consensus for SQL injection detection. *International Journal of Information Security, 23*, 1541–1556. https://doi.org/10.1007/s10207-023-00791-y

Mallah, R., & Quintero, A. (2025). Adversarial threats and defense mechanisms in machine learning-based SQL injection

detection. In *Proceedings of the International Conference on Computing, Networking and Communications* (pp. 180–184). IEEE. https://doi.org/10.1109/ICNC64010.2025.10993834

McGraw, G. (2006) Software Security: Building Security In. 2006 17th International Symposium on Software Reliability Engineering, Raleigh, NC, USA, 7-10 November 2006, 5-6. https://doi.org/10.1109/ISSRE.2006.43

Meshram, P. (2025). SQL injection attack detection and prevention system. *International Journal for Research in Applied Science and Engineering Technology*. https://doi.org/10.22214/IJRASET.2025.74282

Mohamed, N. (2024). Securing transportation web applications: An AI-driven approach to detect and mitigate SQL injection attacks. *Journal of Transportation Security, 17*, 1–18. https://doi.org/10.1007/s12198-023-00269-x

Muduli, D., Shookdeb, S., Zamani, A., Saxena, S., Kanade, A., Parveen, N., & Shameem, M. (2024). SIDNet: A SQL injection detection network for enhancing cybersecurity. *IEEE Access, 12*, 176511–176526. https://doi.org/10.1109/ACCESS.2024.3502293

OWASP. (2021). *SQL injection*. Open Web Application Security Project. https://owasp.org/www-community/attacks/SQL_Injection

Paul, A., Sharma, V., & Olukoya, O. (2024). SQL injection attack: Detection, prioritization and prevention. *Journal of Information Security and Applications, 85*, 103871. https://doi.org/10.1016/j.jisa.2024.103871

Salmawi, H. (2025). Critical evaluation of SQL injection security measures in web applications. *Wasit Journal for Pure Sciences*. https://doi.org/10.31185/wjps.566

Shar, L. K., & Tan, H. B. K. (2013). Predicting SQL injection and cross-site scripting vulnerabilities through mining input sanitization patterns. Information and Software Technology, 55, 1767–1780. https://doi.org/10.1016/j.infsof.2013.04.002

Tang, P., Qiu, W., Huang, Z., Lian, H., & Liu, G. (2020). Detection of SQL injection based on artificial neural network. *Knowledge-Based Systems, 190*, 105528. https://doi.org/10.1016/j.knosys.2020.105528

Xie, X., Ren, C., Fu, Y., & Guo, J. (2019). SQL injection detection for web applications based on elastic-pooling CNN. *IEEE Access, 7*, 151475–151481. https://doi.org/10.1109/ACCESS.2019.2947527

Zhang, B., Ren, R., Liu, J., Jiang, M., Ren, J., & Li, J. (2024). SQLPsdem: A proxy-based mechanism towards detecting, locating and preventing second-order SQL injections. *IEEE Transactions on Software Engineering, 50*, 1807–1826. https://doi.org/10.1109/TSE.2024.3400404