

A Survey of Course Code Representations for Machine Learning-Based Cybersecurity Task

Alumno: Briceño Quiroz Anthony Angel

Trabajo: Ppt sobre Survey a elección

Curso: Metodología de la Investigación

Fecha de entrega: 17/09/2025

Introducción:

Las vulnerabilidades de software son fallos que comprometen la seguridad y permiten ataques maliciosos. Con la creciente dependencia tecnológica, es vital que los proveedores fortalezcan la seguridad. La inteligencia artificial (IA) y el aprendizaje automático (ML) pueden ayudar a identificar estas vulnerabilidades antes de implementar el software, ahorrando tiempo y dinero.



Los modelos de aprendizaje automático (ML) requieren la conversión del código fuente en datos numéricos para su análisis.

Puntos clave:

- El artículo revisa técnicas para representar código fuente.
- Se examinan lenguajes utilizados y su popularidad en ciberseguridad.
- Se identifican lagunas de investigación en el campo.

Conclusión: La investigación busca mejorar la comprensión de la representación de código fuente en ML, enfocándose en áreas poco exploradas.

Motivación:



Problema:

Las vulnerabilidades de software son un riesgo de seguridad crítico.

Solución Propuesta:

Usar Aprendizaje Automático para detectar **vulnerabilidades** de forma temprana.

Brecha:

Para que el ML funcione, el código fuente debe estar convertido a formato numérico ("**representación**"). La forma en que se hace esta representación afecta en gran parte al rendimiento del modelo.

Objetivo del Survey:

Analizar y mapear el estado del arte: que representaciones, lenguajes y modelos se están usando en la ciberseguridad basada en ML.

Metodología:

Para llevar a cabo nuestra revisión sistemática, que implica 3 actividades

Planificar

Realizar

Presentar

Preguntas de Investigación:

- ¿Cuáles son las representaciones de código más utilizadas?
- ¿Ciertas tareas de ciberseguridad utilizan sólo o principalmente un tipo de representación del código fuente?
- ¿Qué tareas de ciberseguridad cubren las técnicas que se han creado?
- ¿Qué lenguajes de programación son los principales objetivos de las técnicas basadas en el aprendizaje automático para las tareas de ciberseguridad?
- ¿Qué modelos se utilizan habitualmente con diferentes representaciones?

Método de búsqueda

("machine learning" OR "deep learning" OR "artificial intelligence") AND ("security" OR "vulnerability") AND ("code")

Se buscaron en 3 BD:

- ACM Library
- IEEE xplora
- Springer Link

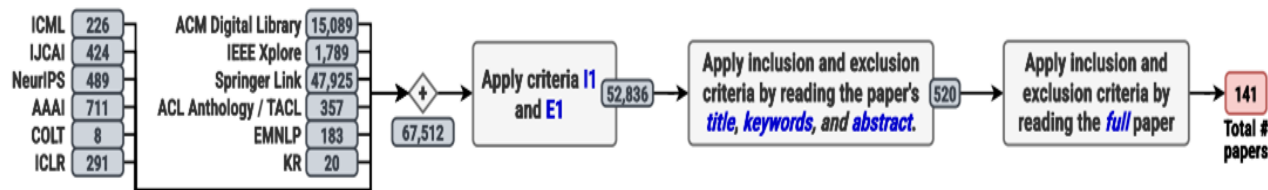
La búsqueda dio un resultado de 67512 artículos,

Criterio de Inclusion y Exclusion:

Inclusion Criteria	Exclusion Criteria
I1 Written between 2012 and May 2023	E1 Duplicated studies
I2 A full paper	E2 Books, reference work entries, reference works
I3 Focused on ML for cybersecurity tasks	E3 Position papers, short papers, tool demo papers, keynotes, reviews, tutorials, and panel discussions
I4 Contains information regarding the source code's representation	E4 Studies not in English
	E5 Survey/comparative studies.

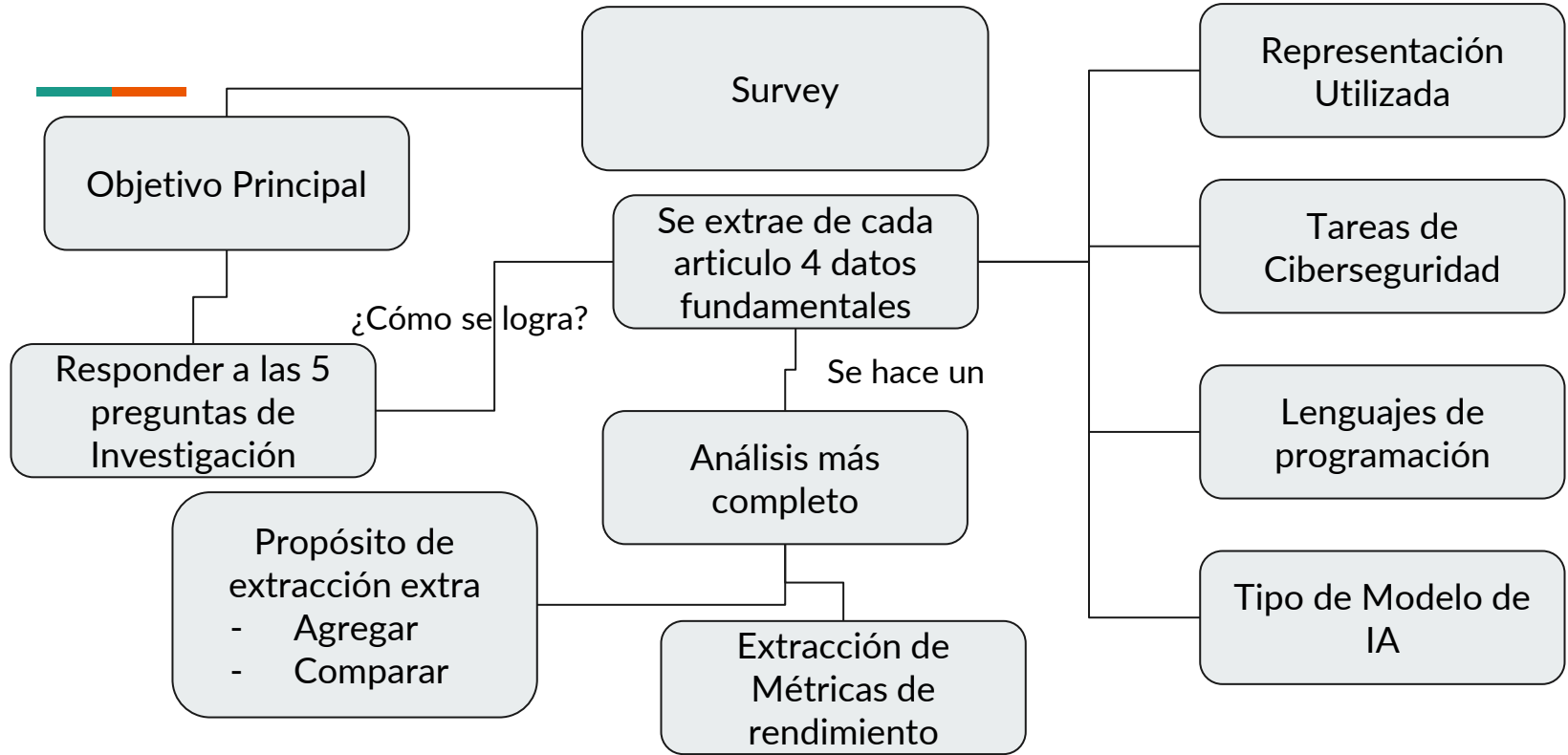
Casey,Santos, et al. "Tabla. 1 Criterio de Inclusión y Exclusión" A Survey of Source Code Representations for Machine Learning-Based Cybersecurity Tasks (2025)

Selección de Artículos:



Casey,Santos, et al. "Fig. 1 Resumen de las tres etapas de nuestro proceso de búsqueda" A Survey of Source Code Representations for Machine Learning-Based Cybersecurity Tasks (2025)

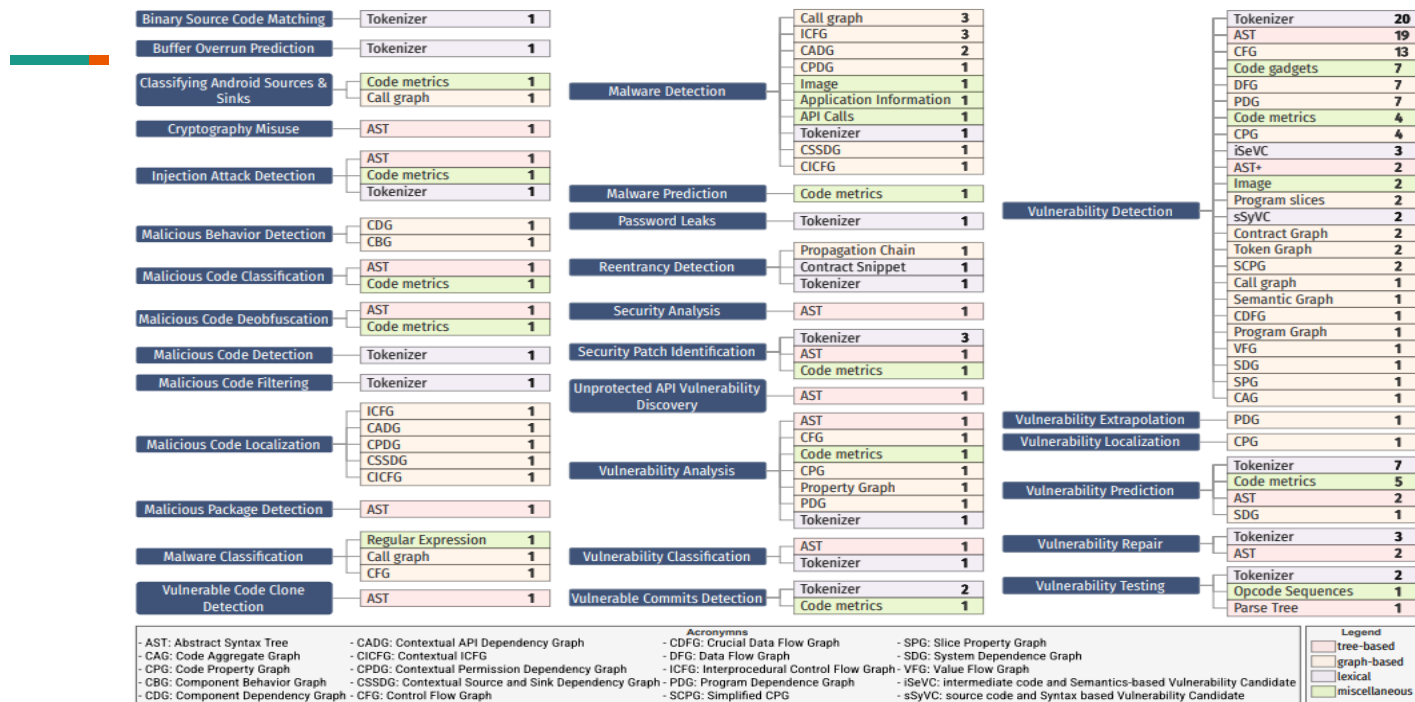
Extracción de Información:



RQ01: ¿Cuáles son las representaciones de código fuente más utilizadas?

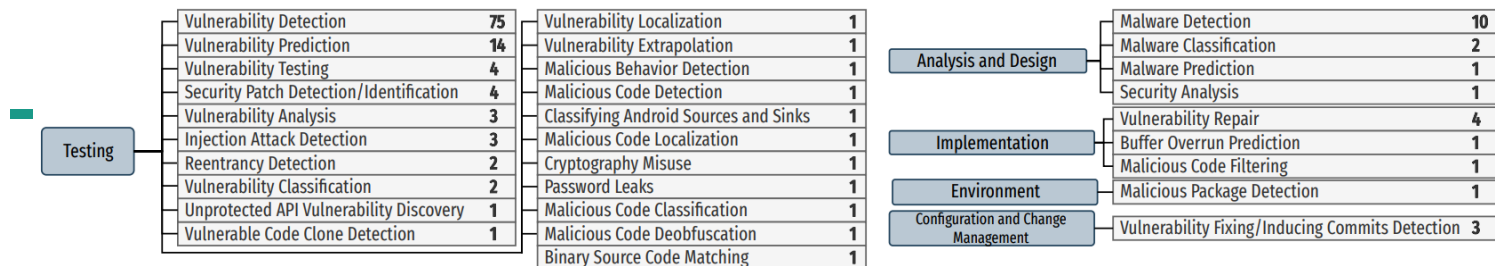
Categorías	1	2	3	4	5
Basada en arboles	Árbol de Sintaxis Abstracta	Árbol de Análisis Sintáctico	AST+		
Basada en grafos	(CFG - Control Flow Graph)	PDG - Program Dependence Graph	DFG - Data Flow Graph	Call Graph	CPG - Code Property Graph
Representaciones léxicas	Tokenizador (Tokenizer):	iSeVC y sSyVC	Fragmento de Contrato (Contract Snippet)		
Representaciones Misceláneas	Imagen (Image)	Expresión Regular (Regular Expression)	Gadgets de Código (Code Gadgets)	Información de la Aplicación (Application Information) y Llamadas a API (API Calls)	Métricas de Código (Code Metrics)

RQ02: ¿Ciertas tareas usan solo un tipo de representación?



Casey,Santos, et al. “Relación entre representaciones y tareas ” A Survey of Source Code Representations for Machine Learning-Based Cybersecurity Tasks (2025)

RQ3: ¿Qué tareas de ciberseguridad cubren las técnicas basadas en el aprendizaje automático?



Casey,Santos, et al. “Fig. 3: Tareas de ciberseguridad en el ciclo RUP”A Survey of Source Code Representations for Machine Learning-Based Cybersecurity Tasks (2025)

RQ4: ¿Qué lenguajes de programación son los principales objetivos de las técnicas basadas en el aprendizaje automático para tareas de ciberseguridad?


Lang. #Papers	Lang. #Papers	Lang. #Papers	Lang. #Papers	Lang. #Papers	Lang. #Papers	Lang. #Papers	Lang. #Papers	Lang. #Papers	Lang. #Papers
C 81 (57.4%)	JS 12 (8.5%)	Python 6 (4.3%)	C# 2 (1.4%)	Gecko 1 (0.7%)	Powershell 1 (0.7%)	SM 1 (0.7%)			
C++ 50 (35.5%)	Solidity 12 (8.5%)	CSS 3 (2.1%)	SQL 2 (1.4%)	Go 1 (0.7%)	Ruby 1 (0.7%)	XML 1 (0.7%)			
Java 36 (25.5%)	PHP 8 (5.7%)	Rust 3 (2.1%)	TS 2 (1.4%)	HTML 1 (0.7%)	Smali 1 (0.7%)	XUL 1 (0.7%)			

JS = JavaScript; TS = TypeScript; SM = SpiderMonkey.

Casey,Santos, et al. “Tabla 3: Lenguajes cubiertos por las técnicas existentes” A Survey of Source Code Representations for Machine Learning-Based Cybersecurity Tasks (2025)

RQ5: ¿Qué modelos se utilizan habitualmente con diferentes representaciones?

Categorías de Modelos:

- 
1. Basado en Secuencia (Mas popular)
 - CNNs, Transformers y LSTMs
 - La popularidad de estos tipo de modelo se debe a la potencia que tienen
 2. Basado en Características
 3. Basados en Arboles
 4. Basado en Grafos
 5. Basado en Redes Neuronales

A pesar de la popularidad de los modelos de secuencia, el modelo individual más utilizado en general fue la **Máquina de Vectores de Soporte (SVM)**. La razón de su éxito es su gran capacidad para aprender y discriminar las características que diferencian distintas clases de código.

Discusión

La investigación muestra que usar **grafos para representar** el código es más **efectivo**. Esto ayuda a los modelos a entender mejor las relaciones semánticas y lógicas, lo que es importante para detectar vulnerabilidades. Tratar el código como texto simple no es efectivo. La **calidad** de las características extraídas es clave para el rendimiento del modelo. Aunque los **tokenizadores** son limitados, pueden ser **útiles** si se mejoran.

Referencias:



ACM Reference Format: Beatrice Casey, Joanna C. S. Santos, and George Perry. 2025. A Survey of Source Code Representations for Machine Learning-Based Cybersecurity Tasks. ACM Comput. Surv. 57, 8, Article 217 (April 2025), 41 pages. <https://doi.org/10.1145/3721977>