

Nuevo software AvArmy de detección y análisis de vulnerabilidades en servicios y aplicaciones web mediante técnicas de aprendizaje automático

Jesús García García

Trabajo de fin de Máster – Seguridad Informática

Director: Sergio Mauricio Martínez Monterrubio. PhD

18 de noviembre de 2020

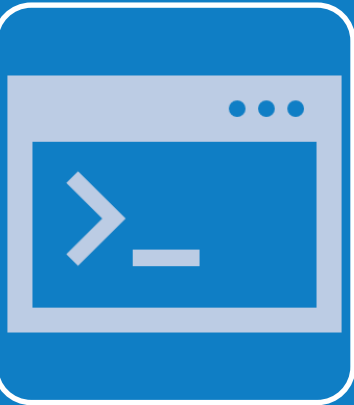
Índice

	Página
Introducción	3
Problema a resolver	6
Hipótesis (H_a y H_0)	7
Objetivos	8
Estado del Arte	11
Software AvArmy	14
Experimentos	15
Demo del software AvArmy	28
Conclusiones	29
Trabajo Futuro	33
Artículo científico	34
Bibliografía	35

Introducción



Esta investigación busca detectar y analizar vulnerabilidades en servicios y aplicaciones web mediante técnicas de aprendizaje automático.



Se ha desarrollado el software AvArmy, que es capaz de extraer métricas enfocadas a las vulnerabilidades más comunes en servicios y aplicaciones web realizando múltiples consultas de vulnerabilidades, extracción de sus resultados y análisis de datos y aplicación de técnicas de aprendizaje automático.

Introducción



Para demostrar la utilidad de dicho software, se ha creado un Dataset enfocado a los sectores estratégicos de España tomando como referencia un total de 10 sectores repartidos en 100 muestras diferentes.



Finalmente se han hecho las pruebas, las cuales han resultado satisfactorias demostrando así que es posible la detección y análisis de vulnerabilidades en servicios web con la inteligencia artificial. En el mercado no se conoce un software que tenga las mismas características.

Motivación

Se parte de la necesidad de valorar la cuantía de vulnerabilidades presentes en los diversos sectores estratégicos de España y de demostrar que las redes neuronales son efectivas para predecir vulnerabilidades.



Problema a resolver

¿Son igual de vulnerables todos los sectores estratégicos de España independientemente de que las empresas sean públicas o privadas?

¿Es posible crear un software robusto que se ayude de la inteligencia artificial y el análisis de datos para mejorar la detección de vulnerabilidades?

Hipótesis

Hipótesis nula (H_0)

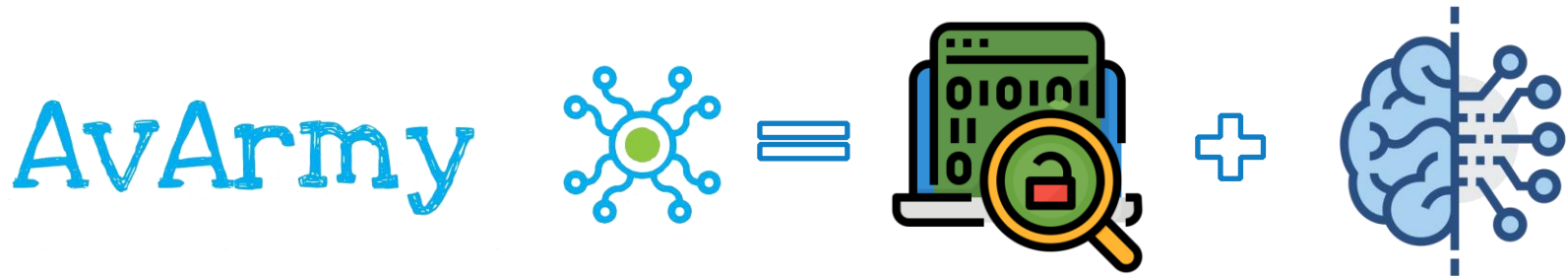
El software Zed Attack Proxy, Vega Vulnerability Scanner, Acunetix web vulnerability, Nessus y Metasploit WMAP son buenos para la detección y análisis de vulnerabilidad en servicios y aplicaciones web.

Hipótesis alternativa (H_A)

El software AvArmy es mejor para la detección y análisis de vulnerabilidades en servicios y aplicaciones web que Zed Attack Proxy, Vega Vulnerability Scanner, Acunetix web vulnerability, Nessus y Metasploit WMAP en los sectores estratégicos de España.

Objetivo General

Desarrollar un software que permita detectar y analizar vulnerabilidades presentes en servicios y aplicaciones web de diversos sectores estratégicos de España en aquellas empresas públicas y privadas, obteniendo una gran precisión aplicando técnicas de aprendizaje automática y fácil adaptabilidad a otros sectores estratégicos



Objetivos específicos

1

Identificar las vulnerabilidades que más riesgos presentan en el ámbito de los servicios y aplicaciones web.

2

Implementar el análisis de las vulnerabilidades haciendo uso del software AvArmy sobre los servicios y aplicaciones web que respondan al estándar de una URI además de todas las APIs y RestFul.

3

Identificar los algoritmos de aprendizaje automático más apropiados para la predicción de vulnerabilidades, partiendo de que las redes neuronales son las más apropiadas.

Objetivos específicos

4

Realizar un análisis de datos de los resultados e implementar los algoritmos de aprendizaje automático más aptos para la predicción de vulnerabilidades.

5

Analizar las diferentes relaciones entre las empresas públicas y privadas de los diferentes sectores estratégicos de España y sus vulnerabilidades en el ámbito de la seguridad informática.

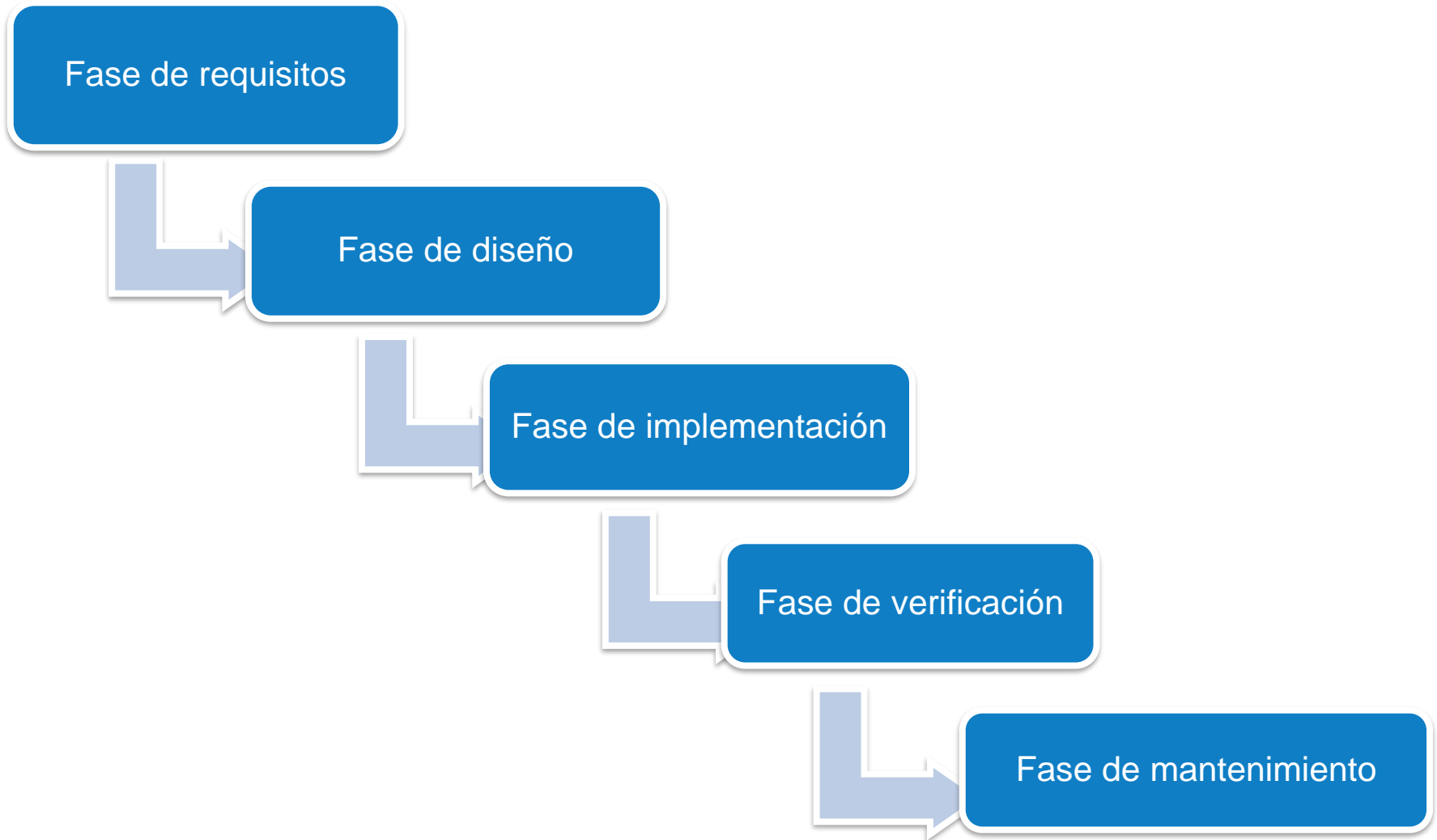
Estado del Arte

Aportación	Referencia
Las herramientas de análisis estático son unas de las mejores maneras de buscar vulnerabilidades en aplicaciones y servicios.	Nunes, P. M. (2017). An empirical study on combining diverse static analysis tools for web security vulnerabilities based on development scenarios. <i>Computing</i> 101, 161–185.
Los scripts en Python son confiables siempre que se verifique la integridad y la detección de vulnerabilidades.	Peng, S. L. (2019). Python Security Analysis Framework in Integrity Verification and Vulnerability Detection. <i>Wuhan Univ. J. Nat. Sci</i> , 24, 141–148.
Existen varias vulnerabilidades al ejecutar la computación en la nube (capa de virtualización) que deben ser examinadas.	Modi, C. A. (2017). Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review. <i>J Supercomput</i> , 73, 1192–1234.

Estado del Arte

Aportación	Referencia
La evaluación de las capacidades de los escáneres resulta dificultosa.	Román Muñoz, F. G. (2018). An algorithm to find relationships between web vulnerabilities. J Supercomput, 74, 1061–1089.
La aplicación del aprendizaje automático en la detección de vulnerabilidades no conocidas es útil.	Román Muñoz, F. S. (2018). Enlargement of vulnerable web applications for testing. J Supercomput, 74, 6598–6617.

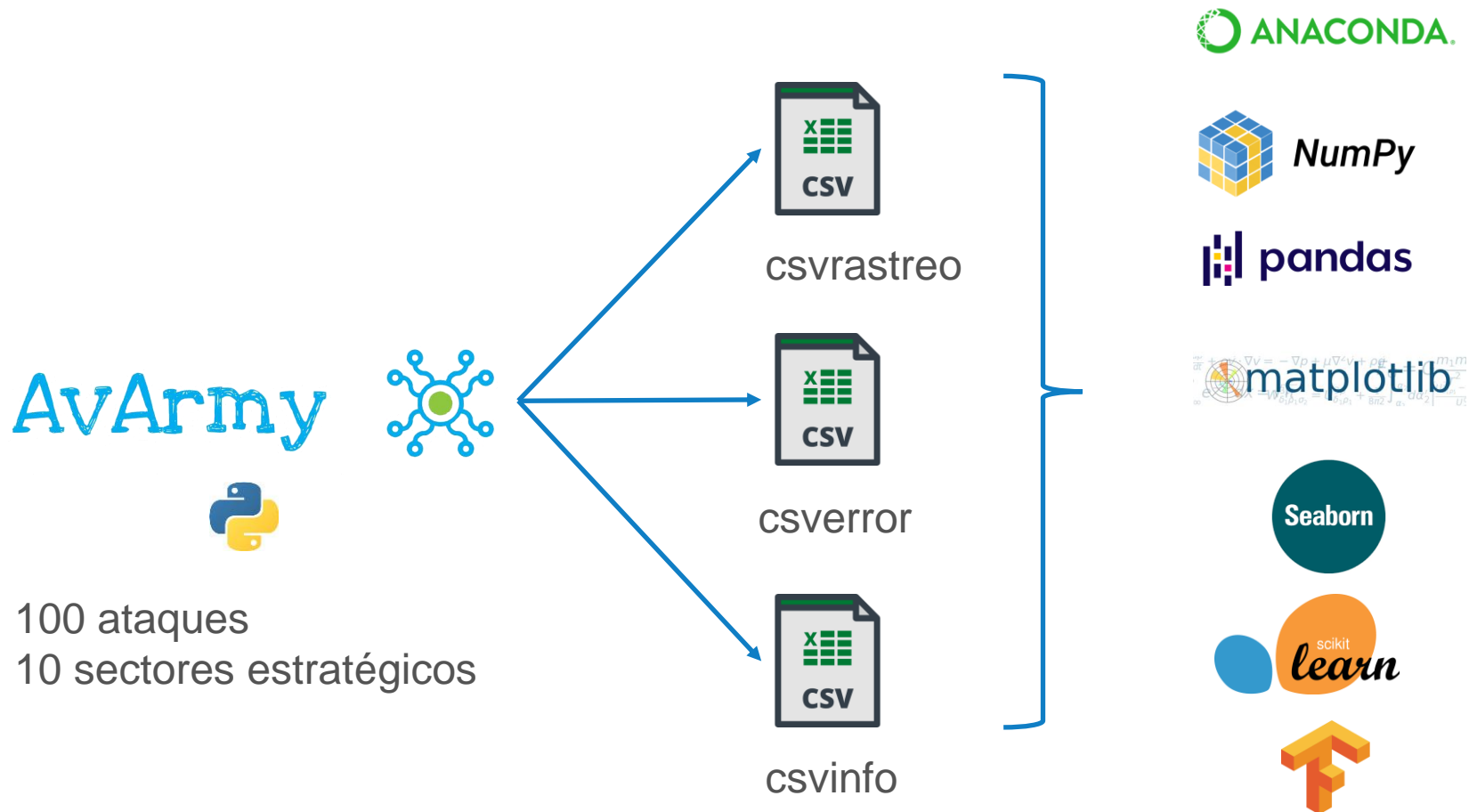
Metodología desarrollo en Cascada



Software AvArmy

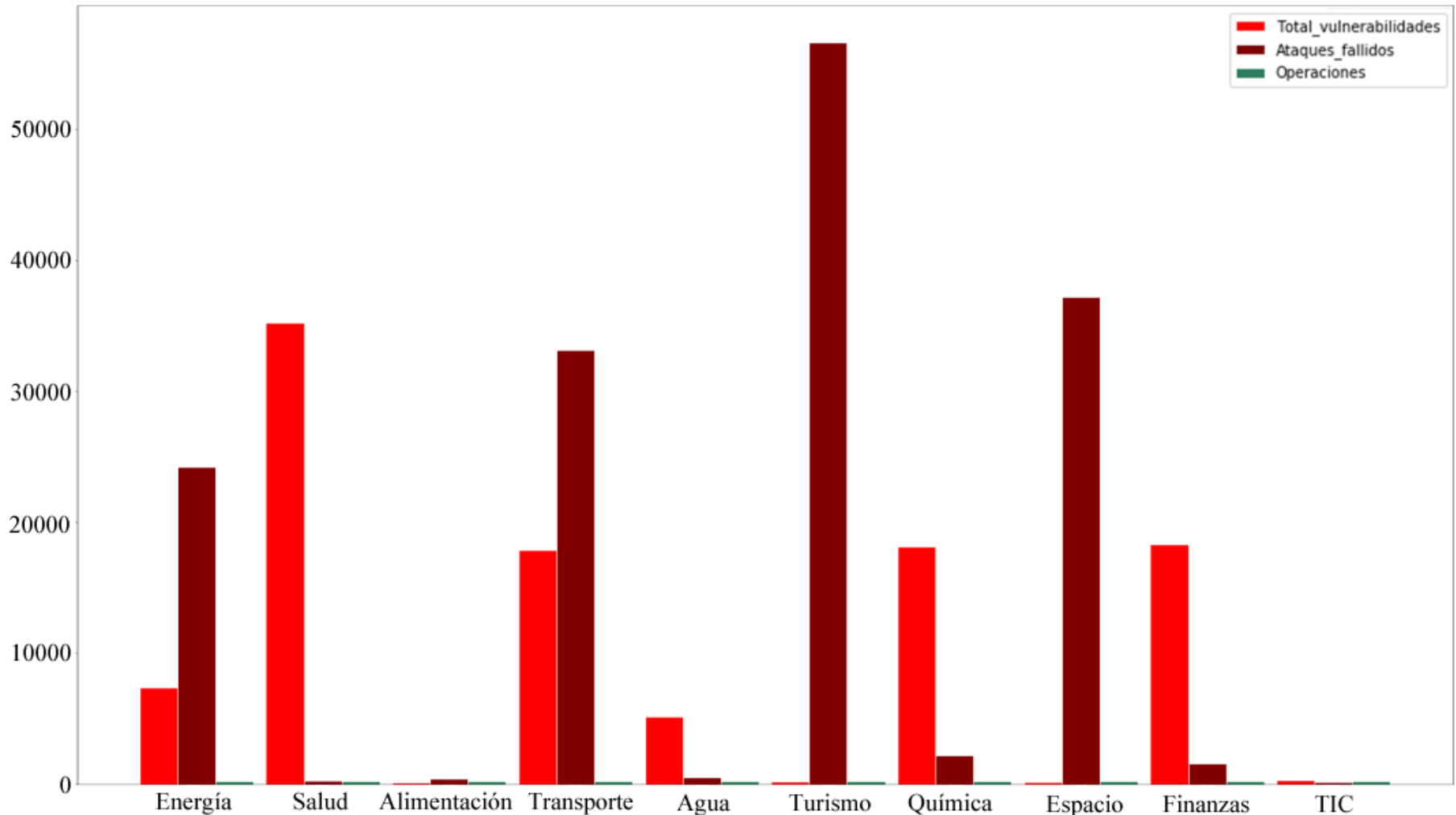


Experimentos – Recolección y procesamiento de datos



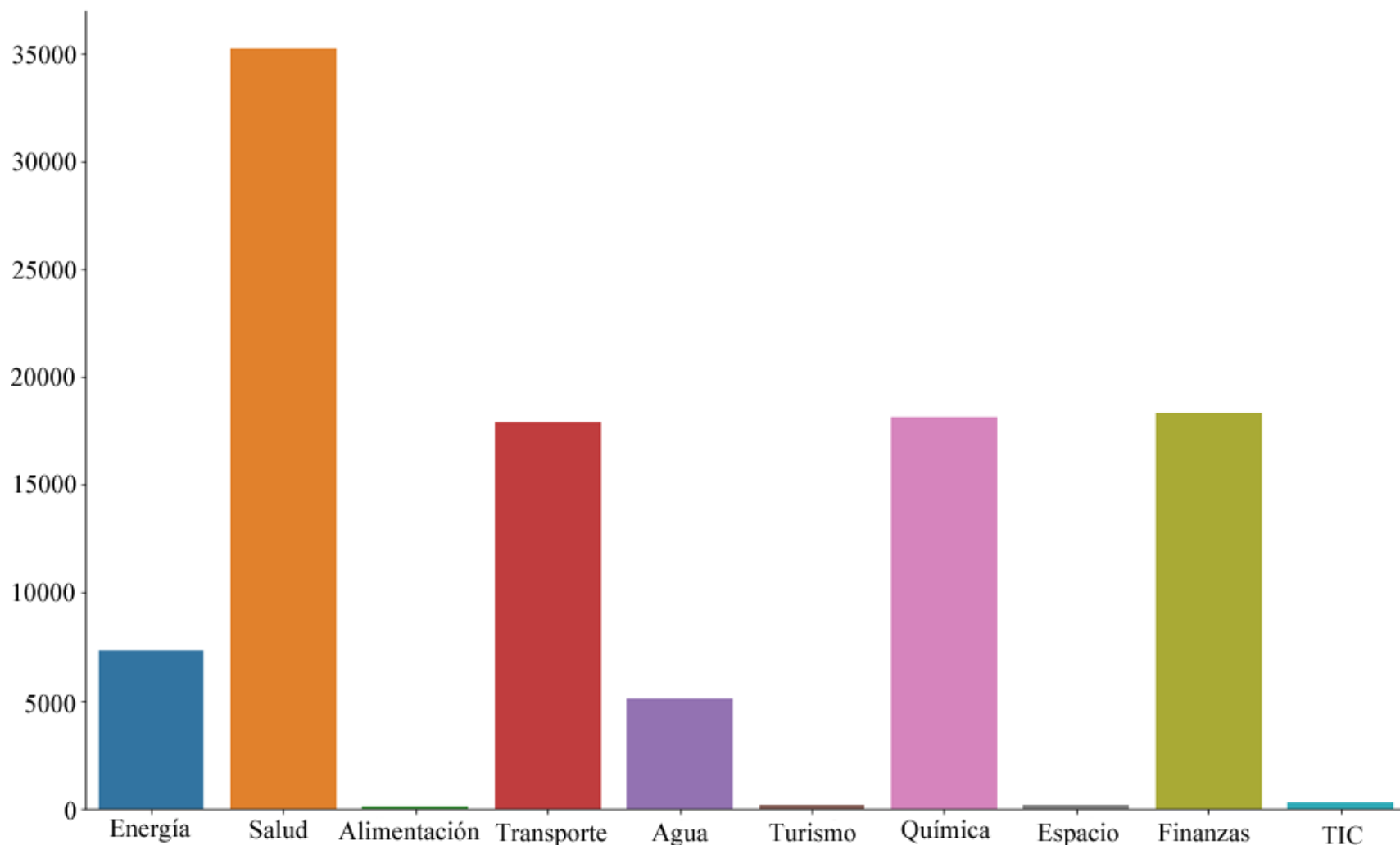
Experimentos – Resultado análisis de datos

Total de vulnerabilidades vs ataques fallidos vs operaciones



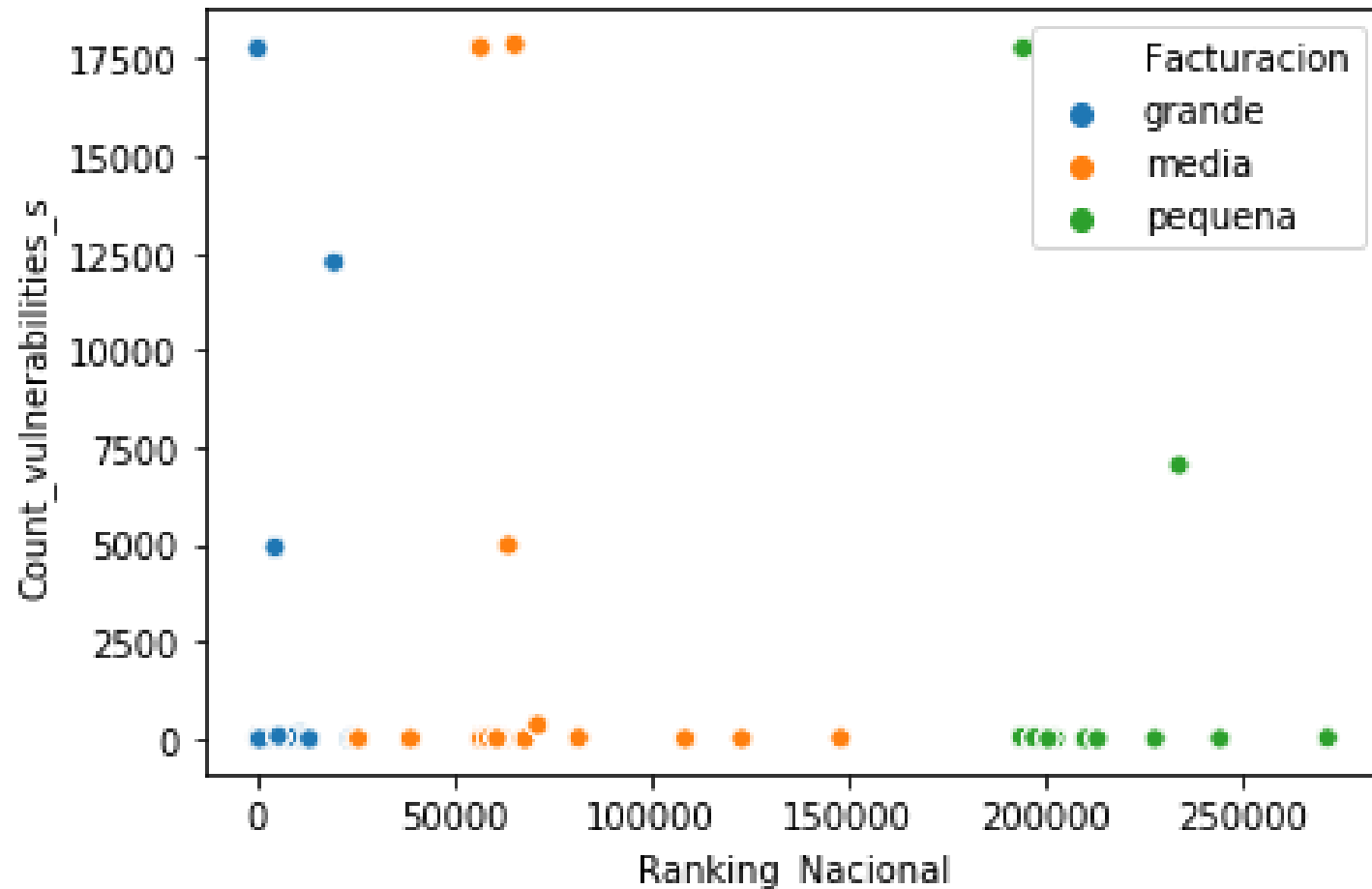
Experimentos – Resultado análisis de datos

Total de vulnerabilidades por sector



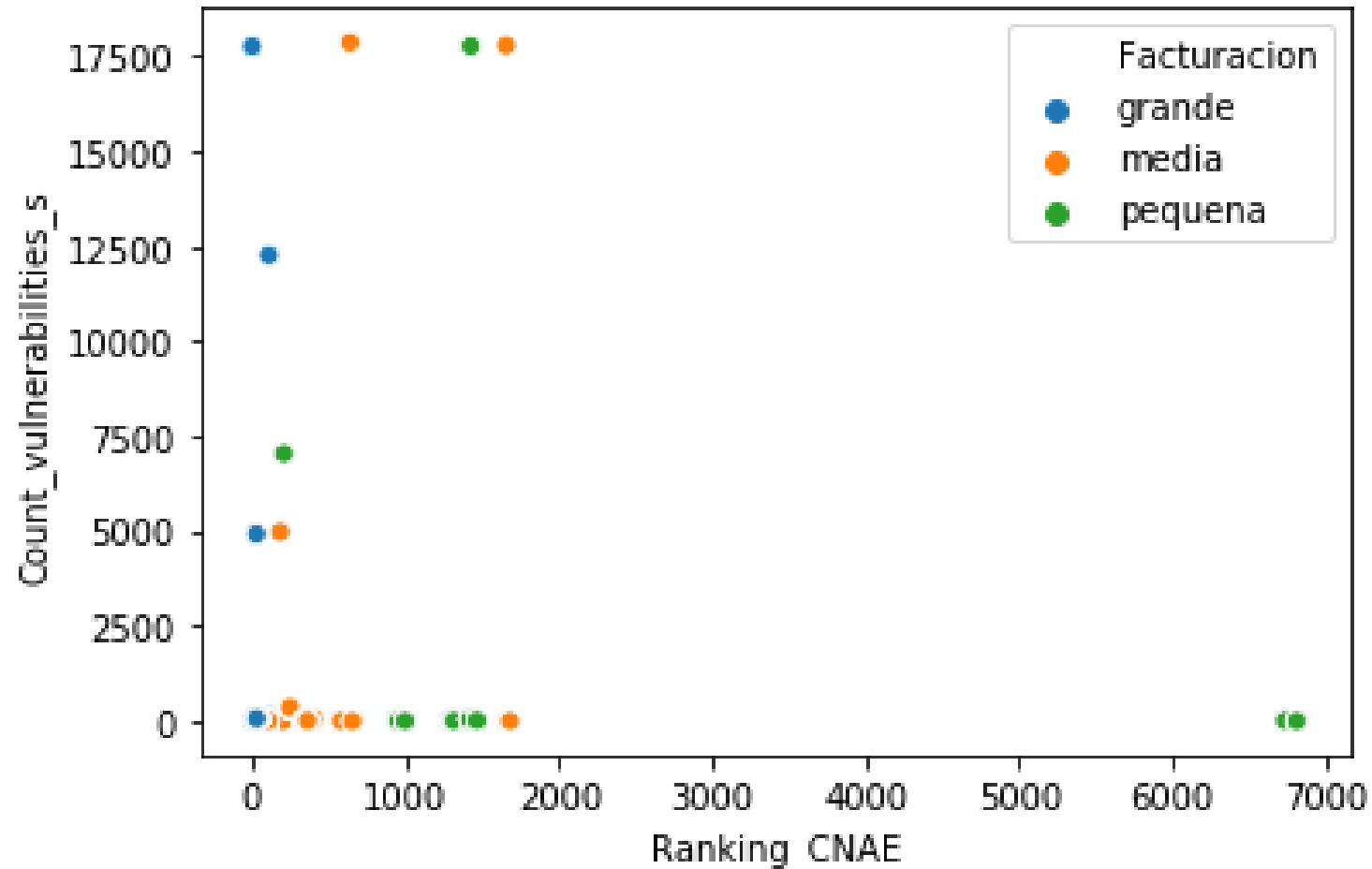
Experimentos – Resultado análisis de datos

Total de vulnerabilidades por facturación (Ranking Nacional)



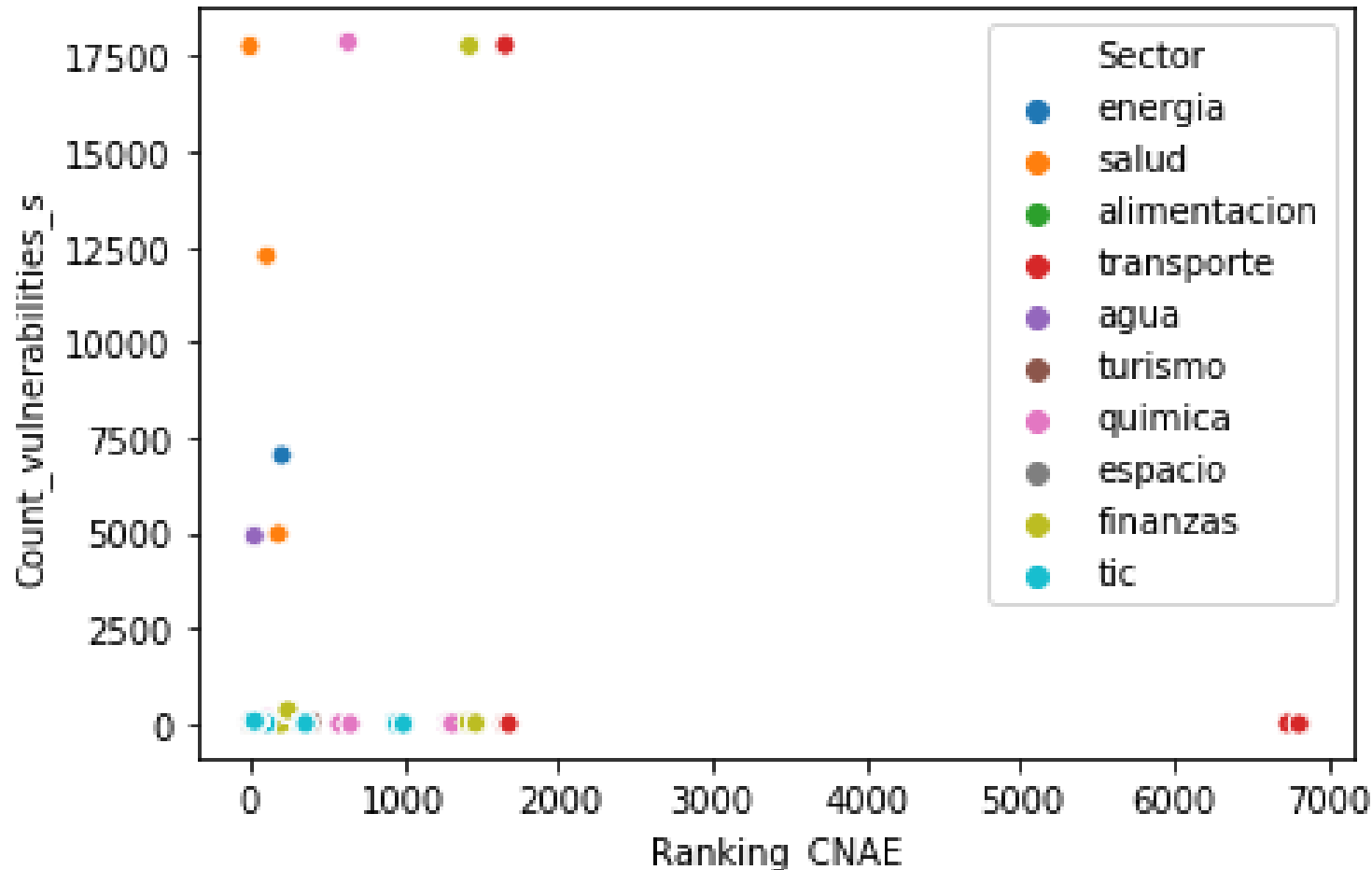
Experimentos – Resultado análisis de datos

Total de vulnerabilidades por facturación (Ranking CNAE)



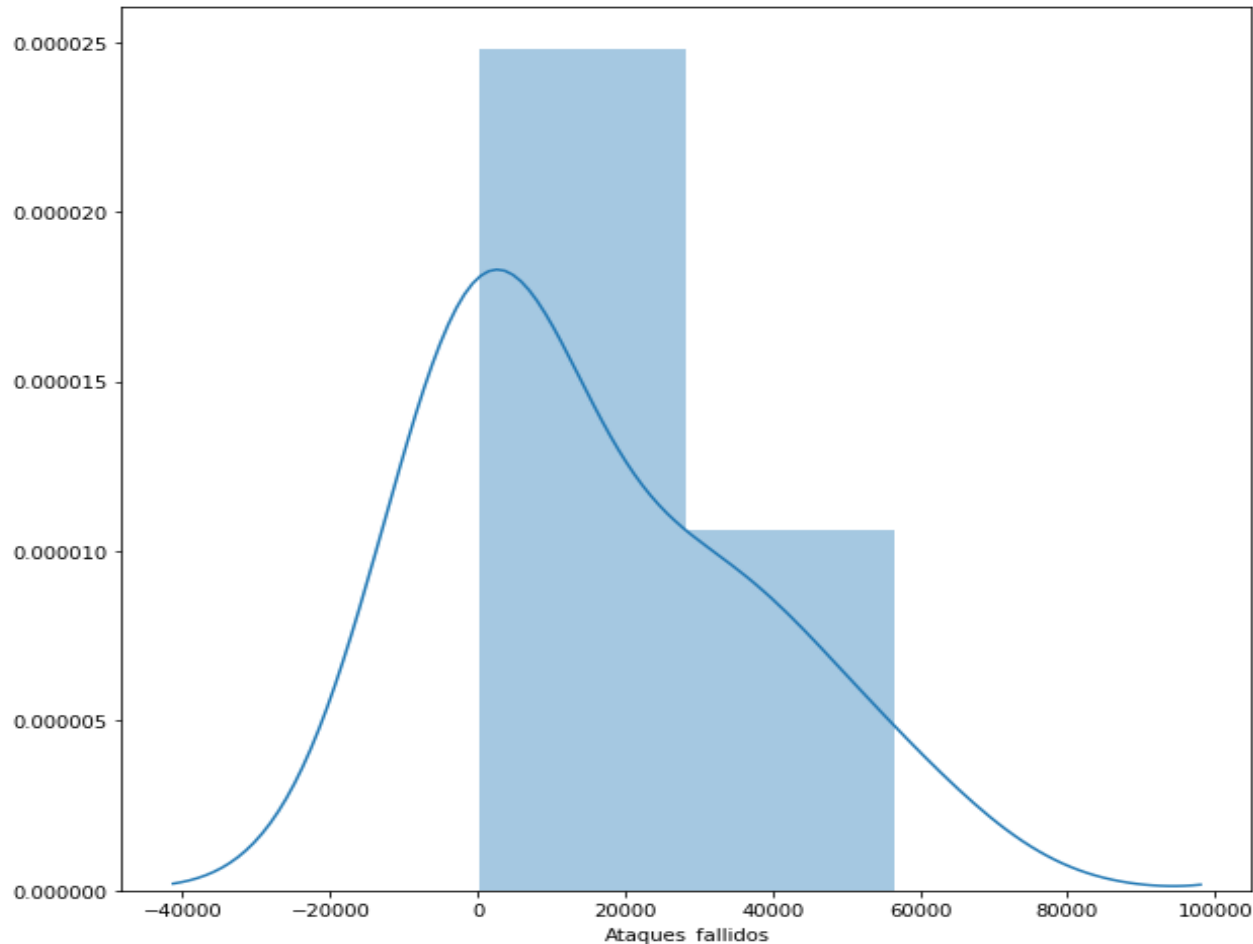
Experimentos – Resultado análisis de datos

Total de vulnerabilidades por sectores (Ranking CNAE)



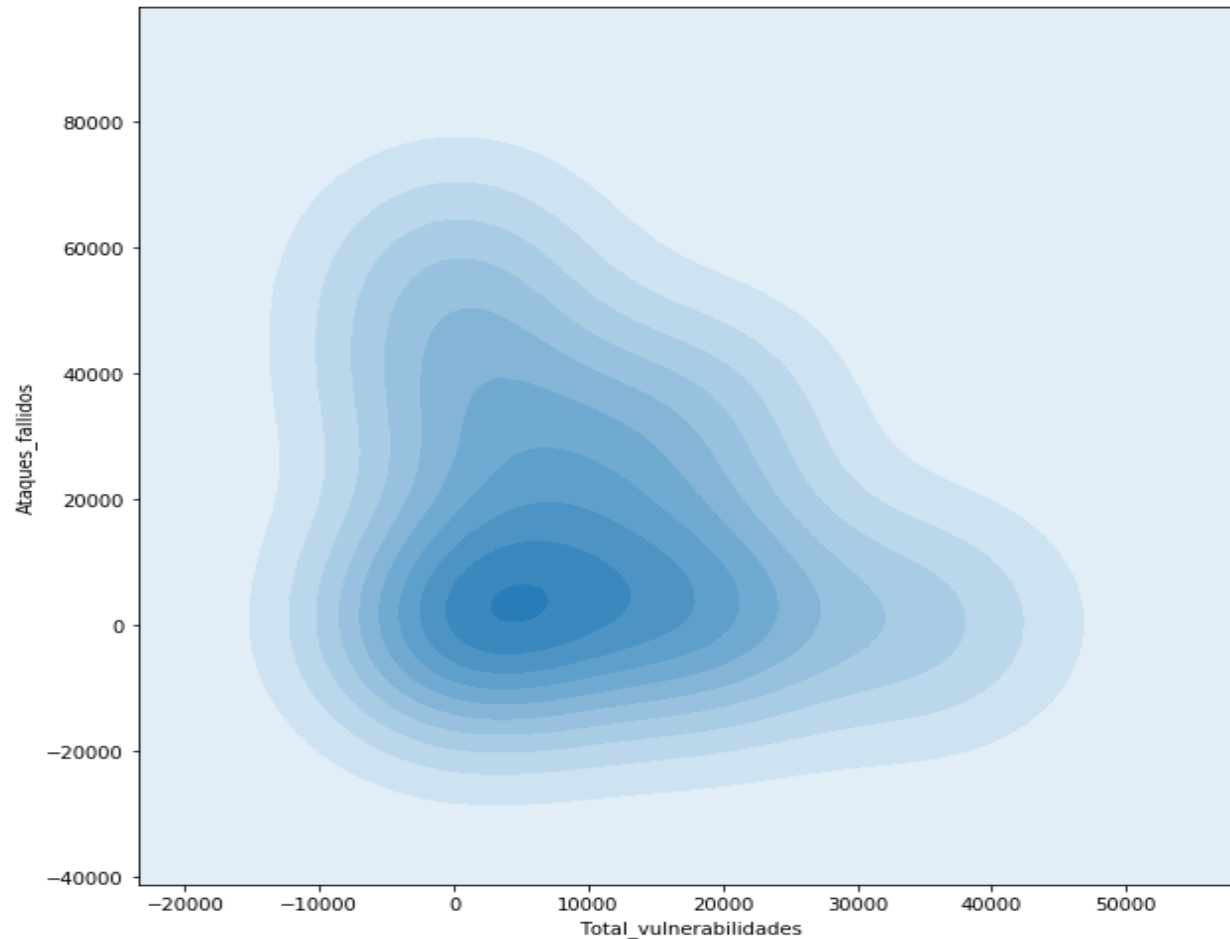
Experimentos – Resultado análisis de datos

Distribución de ataques fallidos



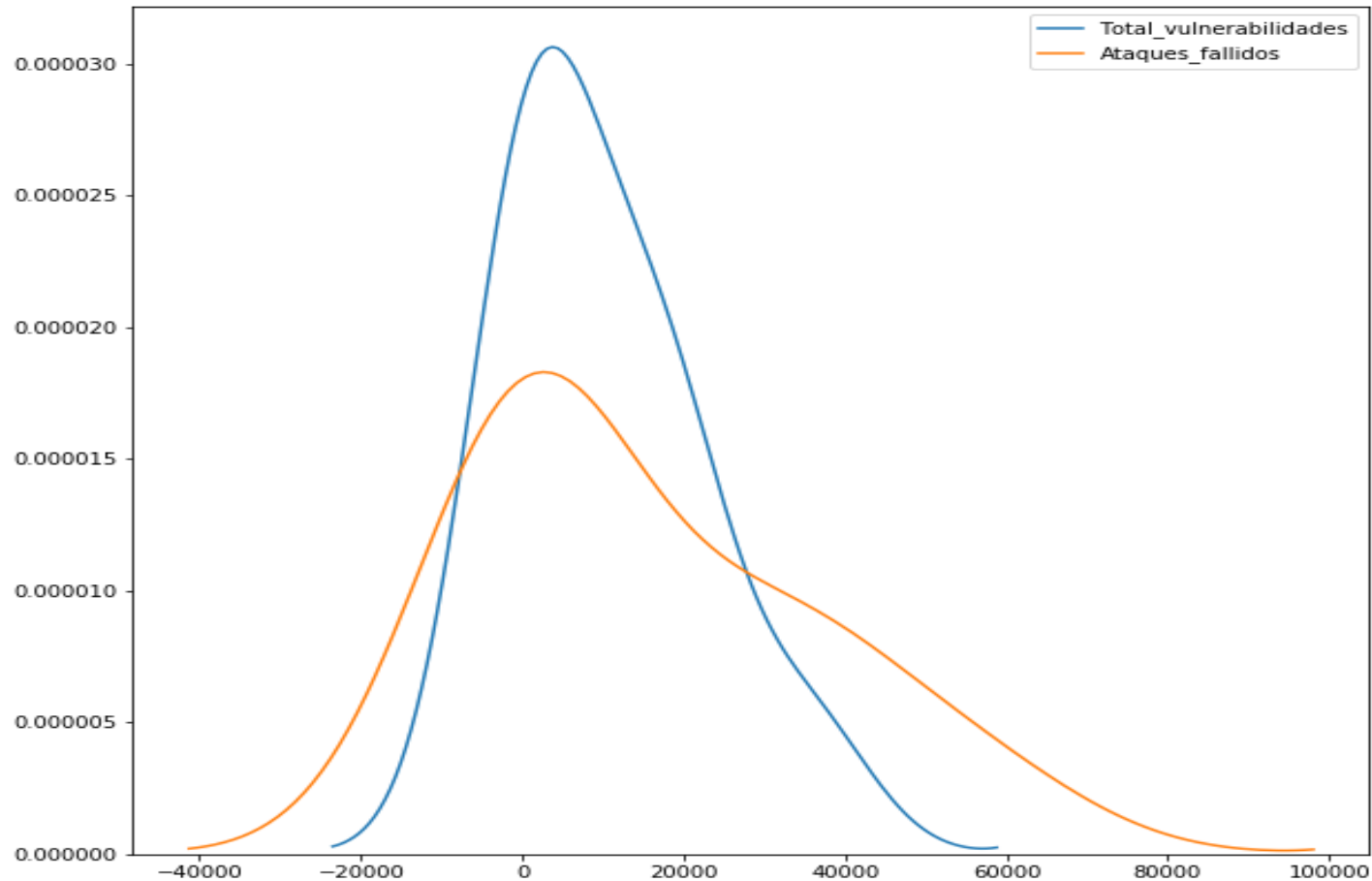
Experimentos – Resultado análisis de datos

Distribución total de vulnerabilidades vs ataques fallidos



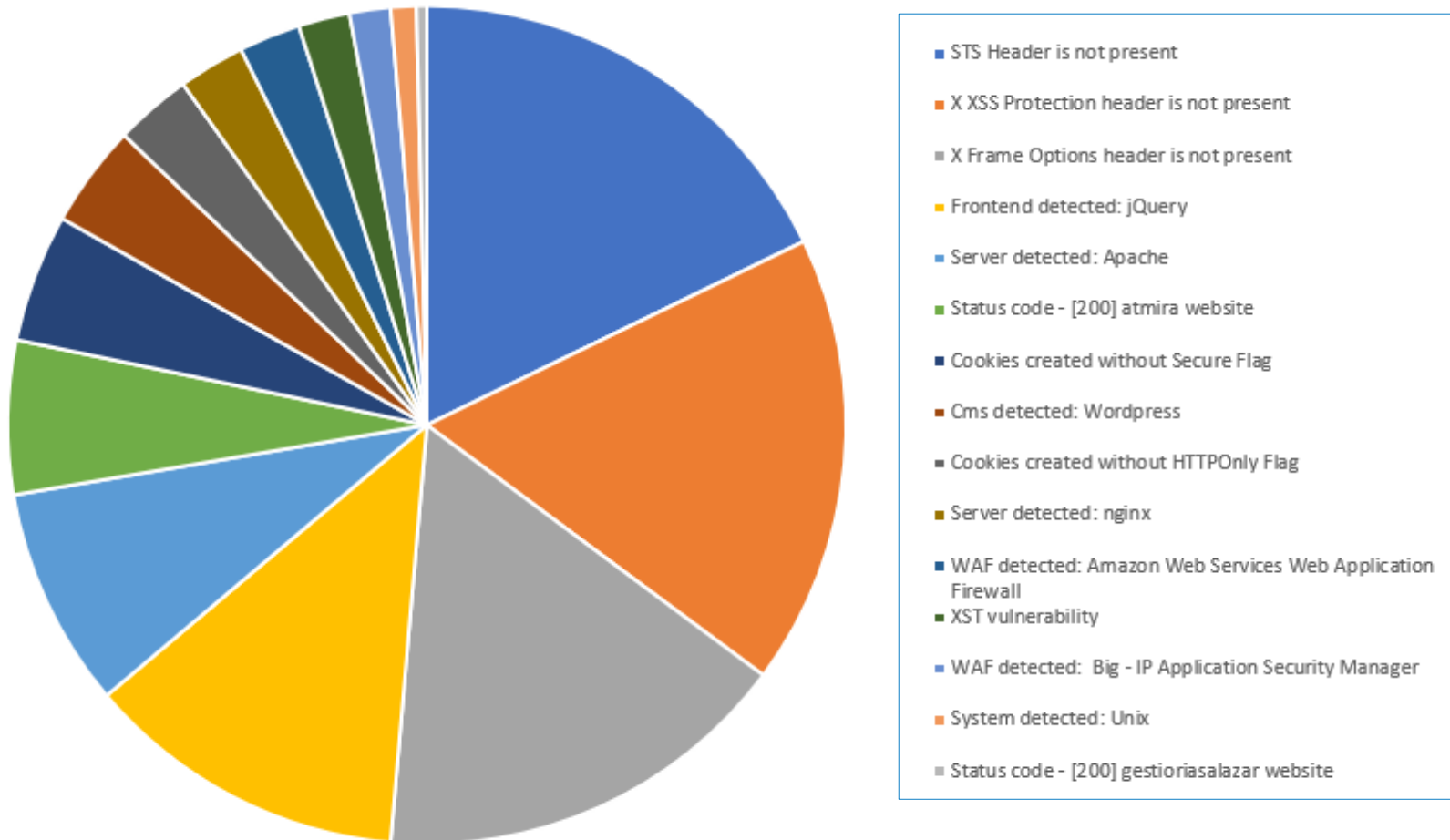
Experimentos – Resultado análisis de datos

Relación ataques perpetrados vs ataques fallidos

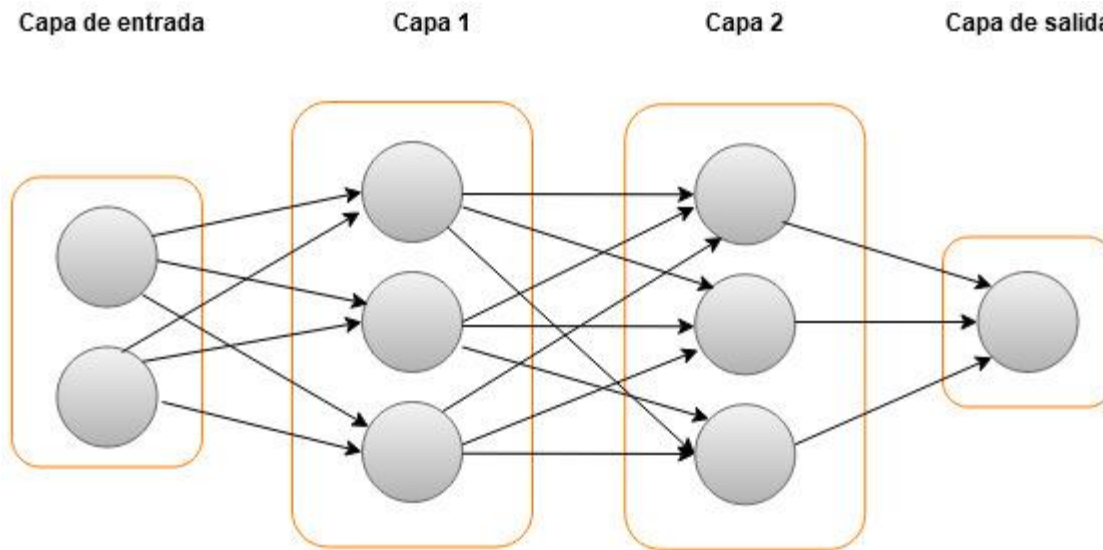


Experimentos – Resultado análisis de datos (V)

Top 15 rastreos más frecuentes



Experimentos – Modelado de la red neuronal



Características

Modelo secuencial

1 capa de entrada

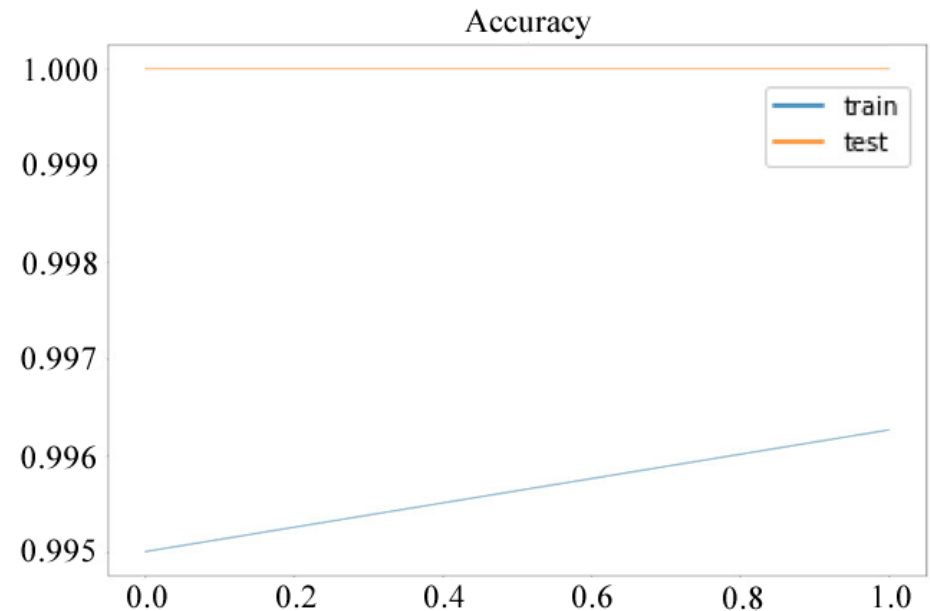
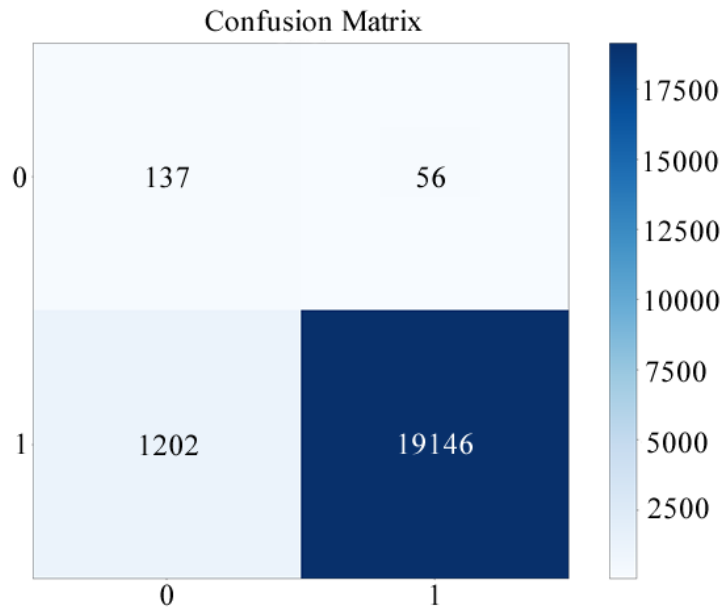
2 capas ocultas

1 capa de salida

Función de activación ReLU y SoftMax (capa de salida)

Optimizador Adam

Experimentos – Evaluación del modelo

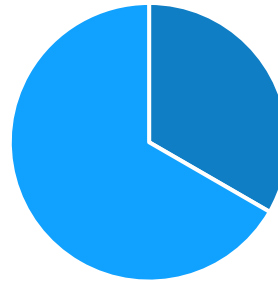


Métrica	Red Neuronal
Precisión	99%
Exhaustividad	94%
Valor F1	96%
Exactitud	93%

Experimentos – Eficacia del software

Para determinar la eficacia del software AvArmy se ha tomado una muestra aleatoria de 504 rows o filas de datos de cada uno de los sectores. Se sabe que 168 ataques no han sido exitosos y 336 sí han sido exitosos.

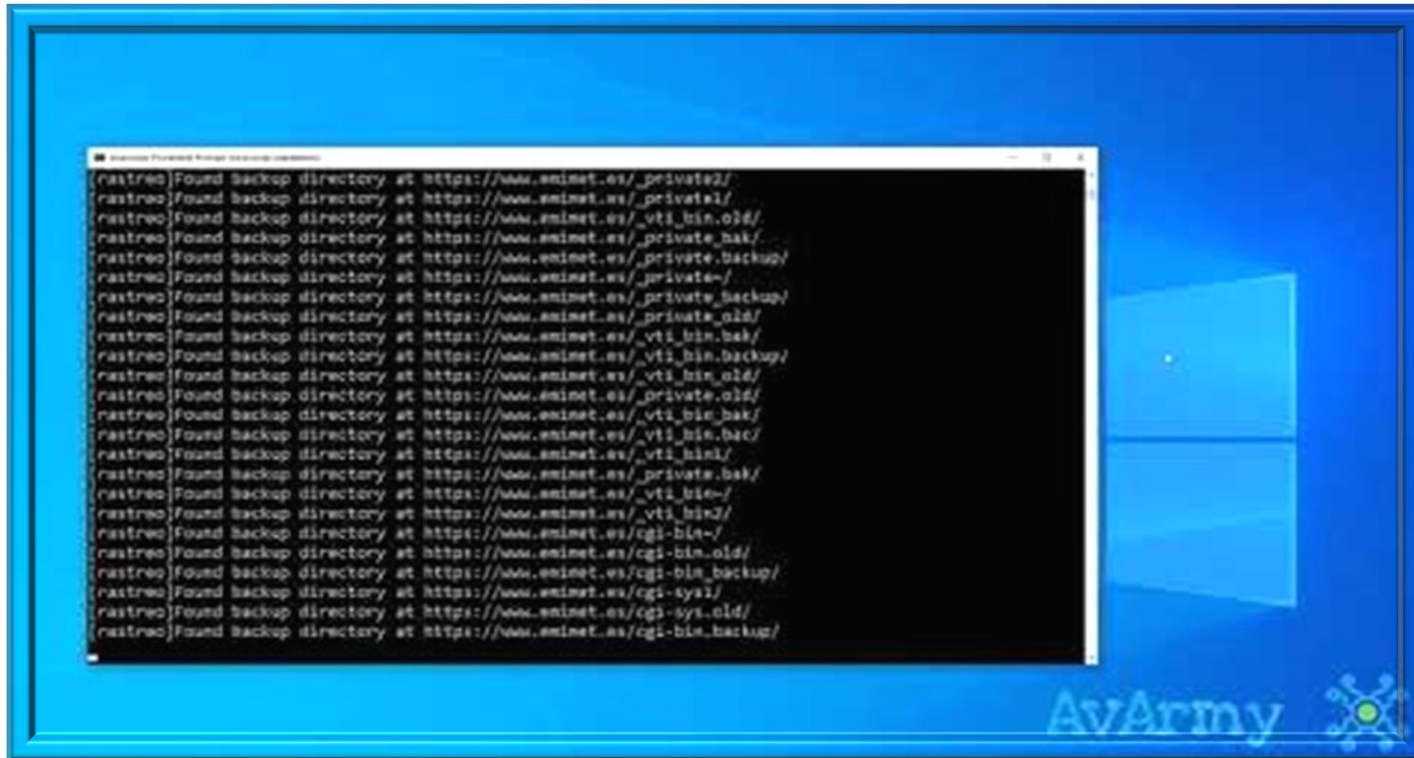
Totalidad de ataques



■ Total de Ataques No Exitosos ■ Total de Ataques Exitosos

$$\frac{\text{Total de Ataques Exitosos}}{\text{Total de muestra tomada}} \times 100 = \frac{336}{504} = 67\%$$

Demo del software AvArmy



Conclusión 1

Se ha conseguido el objetivo principal de esta investigación que fue la construcción del software AvArmy. Dicho software detecta vulnerabilidades y demuestra que las redes neuronales funcionan mejor que otro tipo de software que no las utiliza. Por tanto, el proyecto ha cumplido con todos los objetivos que la UNIR requiere para el desarrollo de un software:

- a) Identificación de requisitos.
- b) Descripción de la herramienta desarrollada.
- c) Evaluación de la usabilidad y aplicación para resolver un problema.



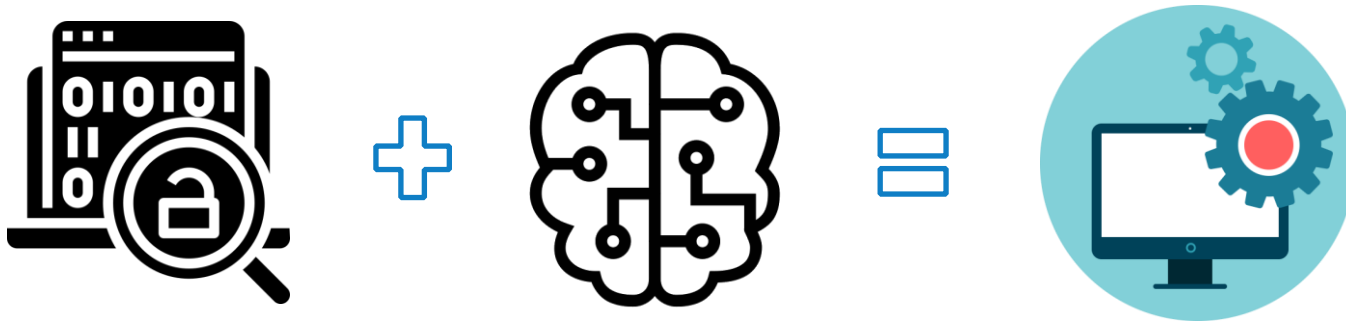
Conclusión 2

Se ha hallado que las empresas del sector salud (35199 vulnerabilidades), finanzas (18293 vulnerabilidades), química (18118 vulnerabilidades) y transporte (17874 vulnerabilidades) son vulnerables o susceptibles a ataques informáticos. Los experimentos muestran que dichos resultados son realmente alarmantes y se llega a la conclusión de que estos sectores aún no se han protegido correctamente invirtiendo en Seguridad Informática.



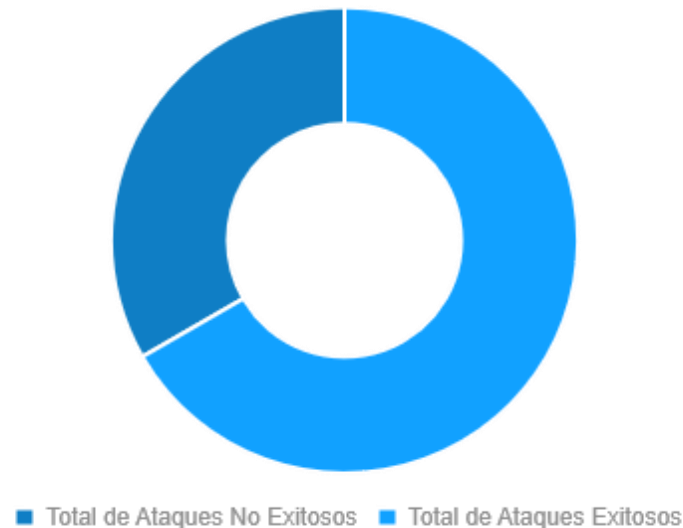
Conclusión 3

Se ha descubierto que la detección de las vulnerabilidades de Seguridad Informática con el campo del Análisis de Datos e inteligencia artificial crean un software más robusto.



Conclusión 4

Se ha demostrado que el software funciona y es eficaz en la detección de vulnerabilidades con un 70% de efectividad, ya que de cada 10 ataques 7 de ellos son exitosos. El software AvAvmy también puede aplicarse en cualquier otro sector estratégico español.



Trabajo Futuro



Utilizar el software en entorno producción real (GCP, AWS, Azure, etc).



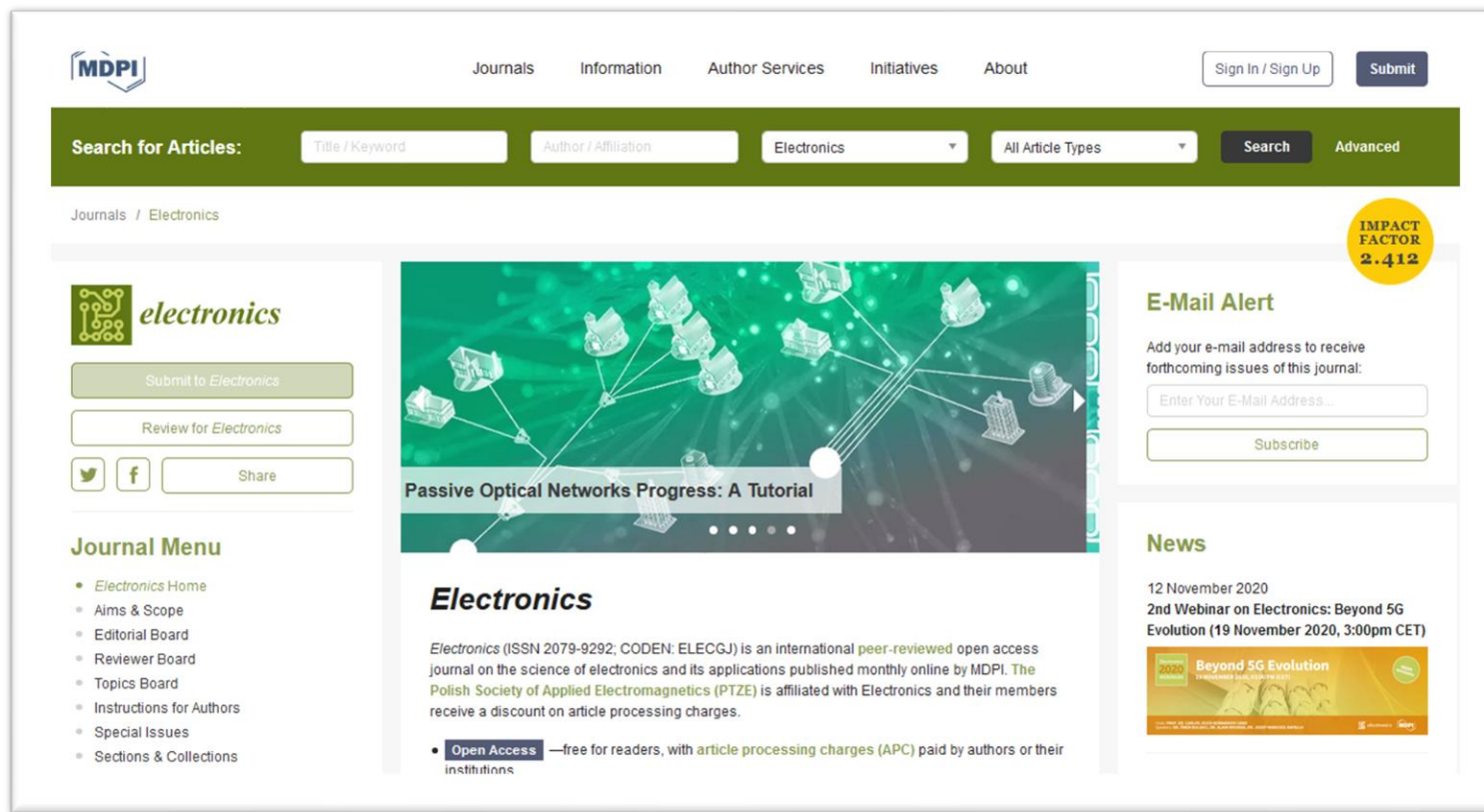
Incorporar más vulnerabilidades.



Mejora de la interfaz de usuario de la aplicación.

Artículo científico

Derivado de esta investigación se ha sometido un artículo científico a la revista MDPI.



The screenshot displays the MDPI Electronics journal homepage. At the top, the MDPI logo is on the left, and navigation links for Journals, Information, Author Services, Initiatives, and About are in the center. On the right, there are buttons for 'Sign In / Sign Up' and 'Submit'. Below this is a green search bar with the text 'Search for Articles:' and input fields for 'Title / Keyword', 'Author / Affiliation', a dropdown menu set to 'Electronics', and another dropdown for 'All Article Types'. A 'Search' button and a link to 'Advanced' search are also present. The main content area features a large banner for 'Passive Optical Networks Progress: A Tutorial' with a green background and network diagrams. To the left of the banner is a sidebar with the 'electronics' logo, a 'Submit to Electronics' button, a 'Review for Electronics' button, social media icons for Twitter and Facebook, and a 'Share' button. Below this is a 'Journal Menu' with links to Electronics Home, Aims & Scope, Editorial Board, Reviewer Board, Topics Board, Instructions for Authors, Special Issues, and Sections & Collections. To the right of the banner, there is an 'E-Mail Alert' section with a text input field and a 'Subscribe' button. Below that is a 'News' section dated 12 November 2020, announcing a '2nd Webinar on Electronics: Beyond 5G Evolution (19 November 2020, 3:00pm CET)' with a corresponding thumbnail image. A yellow circular badge in the top right corner of the page indicates an 'IMPACT FACTOR 2.412'. At the bottom of the page, a bullet point highlights 'Open Access' as free for readers, with article processing charges (APC) paid by authors or their institutions.



Bibliografía

- Abadi, M. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. CoRR abs/1603.04467.
- Acunetix. (2020). Acunetix. Recuperado el 25 de Mayo de 2020, de Acunetix: <https://www.acunetix.com/vulnerability-scanner/>
- Daraio, C. (09 de 06 de 2018). PRODUCTIVITY AND EFFICIENCY ANALYSIS SOFTWARE: AN EXPLORATORY BIBLIOGRAPHICAL SURVEY OF THE OPTIONS. Journal of Economic Surveys. Obtenido de <https://doi.org/10.1111/joes.12270>
- elEconomista. (01 de 07 de 2020). elEconomista. Recuperado el 07 de 2020, de elEconomista: <https://ranking-empresas.eleconomista.es/>
- Gantt, H. (1910). Work, Wages and Profit. Engineering Magazine., ISBN 0-87960-048-9.
- Goutte, Cyril & Gaussier, Eric. (2005). A Probabilistic Interpretation of Precision, Recall and. 345-359.
- Hand, David & Christen, Peter. (2017). A note on using the F-measure for evaluating record linkage algorithms. Statistics and Computin. 10.1007/s11222-017-9746-6.

Bibliografía

- KPMG. (05 de 2018). Ciberseguridad. Los doce sectores estratégicos definidos en España Ciberataques. Recuperado el 15 de 08 de 2020, de KPMG: <https://www.tendencias.kpmg.es/2018/05/ciberataques-colapsar-pais/doce-sectores-ampliar-imagen/>
- m4ll0k. (2018). WAScan. Obtenido de WAscan: <https://github.com/m4ll0k/WAScan>
- Metasploit. (2020). Recuperado el 25 de 05 de 2020, de <https://www.metasploit.com/>
- Modi, C. A. (2017). Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review. J Supercomput, 73, 1192–1234. Obtenido de <https://doi.org/10.1007/s11227-016-1805-9>
- Nunes, P. M. (2017). An empirical study on combining diverse static analysis tools for web security vulnerabilities based on development scenarios. Computing 101, 161–185. Obtenido de <https://doi.org/10.1007/s00607-018-0664-z>
- OWASP. (2020). Recuperado el 25 de 05 de 2020, de <https://owasp.org/www-project-zap/>

Bibliografía

- Peng, S. L. (2019). Python Security Analysis Framework in Integrity Verification and Vulnerability Detection. Wuhan Univ. J. Nat. Sci, 24, 141–148. Obtenido de <https://doi.org/10.1007/s11859-019-1379-5>
- Pressman, R. (2005). Ingeniería del software. Un enfoque práctico. McGraw-Hill.
- Provost, Foster & Fawcett, Tom & Kohavi, Ron. (2001). The Case Against Accuracy Estimation for Comparing Induction Algorithms. Proceedings of the Fifteenth International Conference on Machine Learning.
- Pylint. (2020). Recuperado el 24 de 08 de 2020, de <https://pypi.org/project/pylint/>
- Román Muñoz, F. G. (2018). An algorithm to find relationships between web vulnerabilities. J Supercomput, 74, 1061–1089. Obtenido de <https://doi.org/10.1007/s11227-016-1770-3>



GRACIAS!



unir

LA UNIVERSIDAD
EN INTERNET