

# Robot Judge: An Automated Judge Based on Machine Learning for Food Claim Cases in Lima

Aradiel Castañeda, Hilario<sup>1</sup>; Acosta de la Cruz, Pedro Raúl<sup>1</sup>; Mas Azahuanche, Guillermo Antonio<sup>2</sup>; Flores Salinas, José Alberto<sup>1</sup>; Carhuay Pampas, Enrique Grerorio<sup>1</sup>; Torre camones, Anival Alfredo<sup>1</sup>  
Universidad Nacional de Ingeniería, Peru, haradiel@uni.edu.pe, pacosta@uni.edu.pe, Jflores@uni.edu.pe, ecarhuay@uni.edu.pe, atorrec@uni.edu.pe  
Universidad Nacional del Callao, Peru, gamasa@unac.edu.pe

**Summary** – *The automated judge is an innovative artificial intelligence solution conceived with the purpose of anticipating and issuing a ruling in specific cases, particularly those related to food claims.*

**Keywords**— *Judicial Processes, Judge, Robot, NTLK, Machine Learning, ChatGPT*

## I. INTRODUCCIÓN

En la era de la inteligencia artificial, la tecnología continúa revolucionando diversos campos, incluido el sistema judicial. Con el propósito de agilizar los procesos judiciales y proporcionar respuestas rápidas y precisas en casos relacionados con demandas alimentarias, hemos desarrollado una innovadora solución: el juez automatizado.

El juez automatizado es una aplicación avanzada de inteligencia artificial que utiliza herramientas de deep learning. Este sistema ha sido concebido específicamente para anticipar y emitir fallos en casos de demandas alimentarias. Al entrenar al juez automatizado con un conjunto de datos proporcionado en un archivo CSV, y luego realizar pruebas con otro conjunto de datos también en formato CSV, podemos obtener un veredicto final basado en una evaluación minuciosa y precisa.

Para entrenar al juez automatizado, utilizamos técnicas de aprendizaje profundo y algoritmos sofisticados. El conjunto de datos de entrenamiento, que se presenta en formato CSV, contiene información detallada sobre casos previos de demandas alimentarias. Estos datos incluyen antecedentes, pruebas presentadas, argumentos legales y los fallos correspondientes emitidos por jueces reales. Al procesar y analizar este conjunto de datos a través de algoritmos de deep learning, el juez automatizado puede aprender patrones, considerar precedentes y evaluar factores clave para tomar decisiones informadas.

Por otro lado, para probar la precisión del juez automatizado, utilizamos un conjunto de datos de prueba también en formato CSV. Este conjunto de datos contiene casos de demandas alimentarias adicionales, pero con los fallos reales ocultos. Al someter estos casos al juez automatizado, podemos evaluar la capacidad del sistema para generar veredictos

precisos y compararlos con los fallos reales posteriormente revelados.

Esta metodología basada en el uso de herramientas de deep learning y la manipulación de archivos CSV nos permite desarrollar un juez automatizado confiable y eficiente en el ámbito de las demandas alimentarias. Al aprovechar los avances en inteligencia artificial y aprendizaje automático, buscamos acelerar los procesos legales, al tiempo que garantizamos un veredicto justo y equitativo en estos casos específicos.

## II. ANTECEDENTES

El artículo "¿Adoptar un abogado robot? El modelo de aceptación de la tecnología de abogados robot de inteligencia artificial para la industria legal: un estudio exploratorio" explora la adopción de abogados robot en la industria legal utilizando un modelo de aceptación de tecnología. El estudio tiene como objetivo comprender la disposición de la industria legal aceptar y utilizar abogados robot basados en inteligencia artificial [1][2].

El artículo propone que la adopción de abogados robot puede beneficiar a la industria legal al reducir costos, mejorar la eficiencia y brindar respuestas rápidas y precisas a los problemas legales. Se utiliza un modelo de aceptación de tecnología para analizar los factores que influyen en la adopción de esta tecnología por parte de los profesionales legales.

El estudio explora aspectos como la percepción de utilidad y facilidad de uso de los abogados robot, la influencia de los factores sociales y la compatibilidad con las prácticas existentes en la industria legal. Además, se investiga la confianza y la preocupación ética que puedan surgir en relación con la adopción de abogados robot.

El artículo plantea que la adopción de abogados robot basados en inteligencia artificial puede ser beneficiosa para la industria legal en términos de eficiencia y costo. Sin embargo, también se abordan aspectos como la percepción de utilidad, facilidad de uso, factores sociales, compatibilidad y preocupaciones éticas relacionadas con esta adopción. El estudio busca proporcionar información y conocimientos para

**Digital Object Identifier:** (only for full papers, inserted by LEIRD).  
**ISSN, ISBN:** (to be inserted by LEIRD).  
**DO NOT REMOVE**

comprender mejor la disposición de la industria legal aceptar y utilizar abogados robot en su práctica diaria.

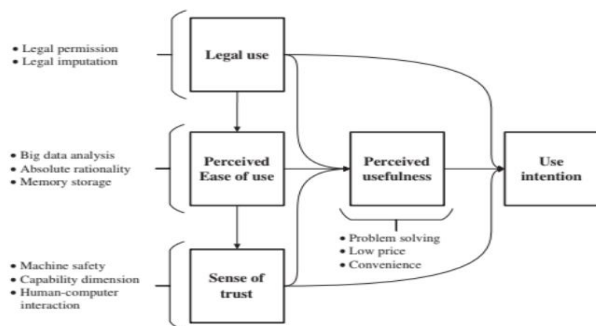


Fig. 1 Modelo de aceptación tecnológica de un robot abogado de inteligencia artificial

### III.HERRAMIENTAS

En el ámbito del derecho procesal penal, la automatización y el uso de chatbots han demostrado ser herramientas prometedoras para agilizar y mejorar la calidad de los procesos legales. En este contexto, el desarrollo del JuezBot se ha convertido en una aplicación destacada que aprovecha diversas herramientas y librerías para su funcionamiento eficiente y efectivo.

Python, reconocido por su versatilidad y facilidad de uso, ha sido el lenguaje de programación principal utilizado en el desarrollo del JuezBot. Esta elección se basa en la amplia variedad de bibliotecas disponibles, lo que permite emplear herramientas especializadas para el procesamiento de datos y el aprendizaje automático.

El JuezBot se beneficia de la librería torch, que es ampliamente utilizada en el campo del aprendizaje profundo. Torch proporciona una estructura y herramientas para la creación y entrenamiento de redes neuronales, lo cual es fundamental para el procesamiento y análisis de datos complejos relacionados con el derecho procesal penal.

Se emplea la librería MySQL para la obtención y gestión de la base de datos de casos, que almacena preguntas y respuestas relacionadas con el derecho procesal penal en categorías específicas. Esto facilita al chatbot ofrecer respuestas rápidas y precisas a las consultas de los usuarios.

La librería Flask se ha utilizado para desarrollar la interfaz interactiva del JuezBot. Flask es una herramienta flexible y ligera para construir aplicaciones web, lo que permite a los usuarios interactuar de manera intuitiva con el chatbot y recibir respuestas en tiempo real.

Para mejorar la capacidad de respuesta y generar respuestas coherentes y contextualmente relevantes, se ha integrado la

biblioteca OpenAI, en particular el modelo GPT-3.5, en el JuezBot. OpenAI utiliza técnicas de aprendizaje profundo y modelos de lenguaje natural para generar respuestas basadas en el contexto y la semántica de las preguntas planteadas.

Además de las mencionadas, el JuezBot también hace uso de otras librerías clave, como nltk (Natural Language Toolkit), que proporciona herramientas para el procesamiento del lenguaje natural, y csv, utilizada para guardar los datos generados y mantener una estructura organizada de la información.

En resumen, el JuezBot aprovecha una combinación de herramientas y librerías, como pandas, MySQL, Flask, OpenAI (GPT-3.5), nltk, csv, torch y modelneuronal, para brindar una solución integral en el ámbito del derecho procesal penal. Esta combinación de tecnologías permite al JuezBot procesar, analizar y generar respuestas precisas y contextualmente relevantes a las consultas de los usuarios. Al automatizar y agilizar los procesos legales, el JuezBot se presenta como una herramienta innovadora que mejora la experiencia de los usuarios y contribuye a la eficiencia y equidad en la administración de justicia.

TABLA I  
HERRAMIENTAS DEL PROYECTO

Herramientas y aplicaciones utilizadas		
Python	OpenAI	matplotlib
HTML	JavaScript	seaborn
HeidiSQL	MySQL	sklearn
NLTK	Heidi	numpy
csv	torch	VStudiocode
modelneuronal	RandomForest	

### III.METODOLOGÍA DEL PROYECTO

El presente proyecto se basa en una metodología descriptiva y explorativa-investigativa con el objetivo de implementar un bot del derecho procesal penal denominado JuezBot utilizando Python, una base de datos, un dataset, herramientas de procesamiento de machine learning y la biblioteca openAI. A continuación, se detallan los objetivos principales y secundarios del proyecto:

#### Objetivos Principales:

- **Lograr un veredicto:** El objetivo principal del JuezBot es lograr emitir un veredicto preciso y fundamentado en casos específicos relacionados con demandas alimentarias. A través del uso de herramientas de inteligencia artificial, aprendizaje automático y procesamiento del lenguaje natural, el JuezBot

busca evaluar la información proporcionada por los usuarios en casos de demandas alimentarias, analizar las leyes y normativas pertinentes, así como considerar precedentes legales y criterios aplicables para generar un veredicto justo y acorde a la situación particular del caso. El objetivo final es proporcionar una respuesta confiable y precisa en el ámbito de las demandas alimentarias, brindando una solución innovadora que agilice y mejore el proceso legal en este tipo de casos.

- Usar las herramientas de Machine Learning para una mejor precisión del robot: El objetivo de utilizar herramientas de Machine Learning en el JuezBot es mejorar la precisión y confiabilidad del chatbot en el ámbito del derecho procesal penal. Mediante el análisis de datos, entrenamiento de modelos y algoritmos de aprendizaje automático, se busca que el JuezBot sea capaz de comprender y evaluar de manera más precisa la información proporcionada por los usuarios, así como de generar respuestas contextualmente relevantes y fundamentadas en precedentes legales. El uso de Machine Learning permite al JuezBot adaptarse y mejorar continuamente su desempeño a medida que se retroalimenta con nuevos datos y casos, lo que contribuye a brindar una asistencia legal más precisa y eficiente.

- Facilitar el proceso de casos: A partir de lo que conoce el Bot, poder predecir los casos y a partir de eso brindar una nueva herramienta de solución a la hora de un caso de demanda de alimentos, que esta herramienta pueda servir en la sociedad y demostrar a las personas como la tecnología se desarrolla para el bien en estos campos.

*Objetivos Secundarios:*

- Desarrollar una interfaz intuitiva y amigable para el usuario: El objetivo de integrar las páginas web en el JuezBot es proporcionar a los usuarios una experiencia cómoda y accesible, a través de una interfaz intuitiva y la posibilidad de acceder a recursos adicionales. Esto mejora la comodidad y satisfacción de los usuarios al interactuar con el robot en casos de demandas alimentarias, permitiéndoles acceder a la información legal y obtener respuestas precisas desde cualquier dispositivo con conexión a internet.

- Implementar el uso de ChatGPT: Esto si bien no es objetivo principal sirve de mucho a la hora de guiar al usuario después de dar el veredicto, es una forma en la que el usuario puede interactuar con un bot para tener más idea de lo que tiene que realizar a continuación.

TABLA II  
METODOLOGIA DEL PROYECTO

Metodología	
1.	Recopilación de papers de Machine Learning
2.	Recopilación de datasets
3.	Investigación y planificación
4.	Desarrollo de código en python
5.	Pruebas del código
6.	Corrección y mejora del código
7.	Resultados y análisis de esto
8.	Retroalimentación

Para llevar a cabo el proyecto, se siguieron los siguientes pasos:

1. Recopilación de papers e información relacionada con el uso de robots que a partir de aplicar machine learning pueden capacitar al usuario.
2. Se llevó a cabo una investigación exhaustiva sobre las funcionalidades y características necesarias para desarrollar un juez robot que desarrolle inteligencia automatizada y de un veredicto.
3. Utilizando el lenguaje de programación Python, se desarrolló el código necesario para implementar el JuezBot.
4. Se realizaron pruebas exhaustivas del bot y de los datasets (train y test) para identificar posibles errores y bugs.
5. Se realizaron iteraciones y ajustes en el código para aprovechar al máximo las capacidades de generación de texto de la inteligencia artificial
6. Se evaluaron los resultados obtenidos a través de pruebas y evaluaciones de usuarios reales
7. La retroalimentación después de los resultados obtenidos

Cada uno de estos pasos fue necesario para la realización del proyecto, ya que permitieron una planificación adecuada y una ejecución efectiva para el desarrollo de JuezBot.

IV.DISEÑO E IMPLEMENTACION JUEZ ROBOT

El Juezbot es un sistema basado en inteligencia artificial que utiliza técnicas de machine learning para responder consultas y predecir el resultado de una sentencia en casos de denuncia de alimentos. Se ha desarrollado utilizando el lenguaje de programación Python y se han utilizado Jupyter Notebook y varias bibliotecas para el entrenamiento del modelo predictivo. Además, se ha implementado una página web que permite a los usuarios interactuar con el Juezbot y obtener predicciones de sentencias. Para aplicar ello se desarrollo la arquitectura del sistema y luego la implementación técnica de la funcionalidad.

## Arquitectura del sistema

El sistema del Juezbots consta de los siguientes componentes principales:

TABLA III  
DATASET PARA EL ENTRENAMIENTO DEL MODELO PREDICTIVO

id_expediente	Cal SUR	Solicitud Porcentaje	Org.Jurisdiccional	Tipo de presentante	Documento	Edad	Deposito Judicial	Arancel	Anexos	Acompañados	Observacion RF	veredicto
01018-2018-0-1815-JP-FC-07	1317	60	N2	Demandado	Escrito	39	0	0	0	0	0 30	1
02045-2019-0-1863-JP-FC-09	1445	70	N5	Demandante	Escrito	25	0	0	0	1	0 45	1
03078-2020-0-1902-JP-FC-06	1221	50	N3	Demandado	Escrito	25	0	0	1	0	0 0	0
04059-2021-0-1951-JP-FC-04	1547	80	N4	Demandante	Escrito	26	0	0	1	0	0 10	1
05098-2019-0-2008-JP-FC-09	1358	55	N8	Demandado	Escrito	25	0	0	0	1	0 20	1

**Entrenamiento del modelo predictivo:** Se recopiló un conjunto de datos de 200 demandas alimenticias en Lima para el entrenamiento y 50 para el testeo. Se utilizaron técnicas de procesamiento de lenguaje natural (NLP) como Stemming y la biblioteca NLTK para preparar los datos. Se probaron varios modelos, incluyendo Logistic Regression (Regresión Logística), Decision Tree (Árbol de Decisión), Random Forest (Bosque Aleatorio) y Gradient Boosting (Refuerzo Gradiente). Tras comparar la precisión, se seleccionó el modelo Random Forest como el más adecuado para el Juezbots.

**Modelo predictivo:** Utilizando el modelo Random Forest seleccionado, se desarrolló un modelo predictivo capaz de recibir variables relacionadas con la denuncia de alimentos y predecir el resultado de la sentencia. Se realizaron ajustes y optimizaciones para maximizar la precisión y confiabilidad del modelo.

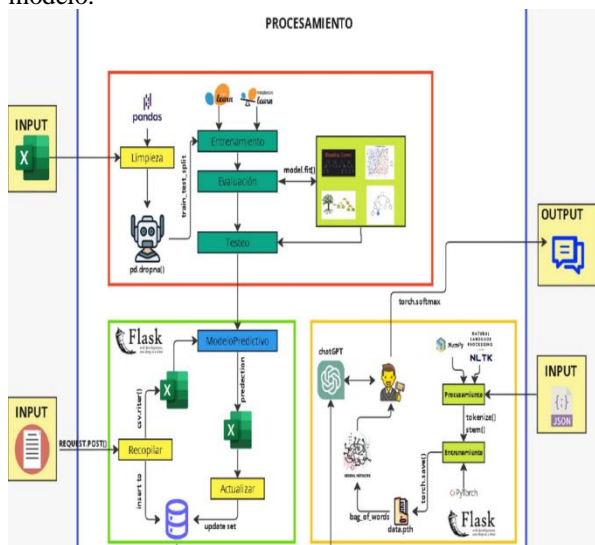


Fig. 2 Arquitectura del juez robot

**Interfaz de usuario y página web:** Se creó una página web que sirve como interfaz de usuario para el Juezbots. La página web permite a los usuarios enviar los datos de entrada requeridos por el modelo predictivo y recibir la predicción de la sentencia en respuesta. Se utilizó HTML, CSS y JavaScript para desarrollar la página web y garantizar una experiencia de usuario intuitiva y agradable[4][5][6][7].

**Integración con ChatGPT:** Para mejorar la capacidad de respuesta del Juezbots, se integró con ChatGPT. Utilizando la información obtenida de la predicción de sentencia y los resultados, el Juezbots puede responder consultas adicionales realizadas por los usuarios.

### Implementación técnica

La implementación del Juezbots se basó en las siguientes tecnologías y bibliotecas:

**Python:** Se utilizó Python como lenguaje de programación principal para el desarrollo del Juezbots y el entrenamiento del modelo predictivo.

**Jupyter Notebook:** Se utilizó Jupyter Notebook para el desarrollo, experimentación y entrenamiento del modelo predictivo.

**Bibliotecas de machine learning:** Se utilizaron diversas bibliotecas populares, como scikit-learn, para la implementación y el entrenamiento de los modelos predictivos, con énfasis en el modelo Random Forest.

**Procesamiento de lenguaje natural (NLP):** Se utilizó la biblioteca NLTK en conjunto con técnicas de Stemming para preparar los datos y realizar el entrenamiento del modelo predictivo.

**HTML, CSS y JavaScript:** Estas tecnologías se emplearon para desarrollar la página web y la interfaz de usuario del Juezbots, brindando una experiencia interactiva y agradable para los usuarios[3].

**ChatGPT:** Se integró ChatGPT para permitir al Juezbots responder consultas adicionales basadas en la información obtenida de la predicción de la sentencia y los resultados.

## VIII. CONCLUSIONES

En primer lugar, el entrenamiento y testeo exitoso del JuezBot ha demostrado su capacidad para procesar casos de demanda de alimentos como input y proporcionar un veredicto o resolución final como output. Esto es una prueba del funcionamiento efectivo del chatbot en el ámbito del derecho procesal penal, lo que puede ser una herramienta valiosa para los abogados, denunciantes y denunciados.

En segundo lugar, el JuezBot ha demostrado ser altamente eficiente al automatizar tareas que anteriormente requerían un esfuerzo manual considerable. La capacidad de respuesta rápida y precisa, tanto a través del modelo de lenguaje ChatGPT como con respuestas programadas específicas, ha permitido ahorrar tiempo valioso a los usuarios al obtener la información que necesitan de manera ágil.

En tercer lugar, la mejora en la calidad y objetividad de las respuestas es otra conclusión importante del proyecto. El uso de tecnologías como Python, OpenAI y una base de datos ha permitido al JuezBot proporcionar respuestas consistentes y objetivas en el ámbito del proceso penal. Los criterios legales y las normativas pertinentes se han integrado en el sistema, asegurando que los usuarios reciban información precisa y confiable.

En cuarto lugar, es importante destacar el potencial de mejora continua del JuezBot. La integración de herramientas de inteligencia artificial y tecnologías emergentes, como "affective computing", permitiría al chatbot reconocer y adaptarse a las emociones de los usuarios, mejorando así su experiencia y satisfacción. Esto demuestra que el JuezBot está preparado para enfrentar los desafíos futuros y seguir evolucionando.

En quinto lugar, la integración del JuezBot en una página web ha sido una decisión estratégica que ha proporcionado beneficios significativos tanto para los abogados como para los denunciantes y denunciados. La facilidad de acceso a través de una plataforma en línea ha mejorado la accesibilidad y la disponibilidad del servicio, lo que permite a los usuarios obtener respuestas rápidas y precisas en cualquier momento y lugar.

Por último, el avance tecnológico y la exploración de nuevas funcionalidades demuestran que el JuezBot tiene el potencial de seguir evolucionando y mejorando en el ámbito legal. En conclusión, el desarrollo e implementación del JuezBot ha demostrado ser una solución efectiva y eficiente para brindar apoyo y asistencia en el proceso penal. Con su capacidad de automatización, precisión en las respuestas y la integración en una página web, el JuezBot ha proporcionado beneficios significativos para los usuarios.

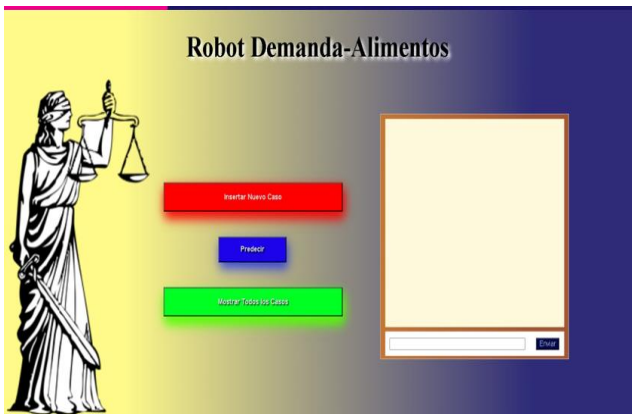


Fig. 3 Interfaz del robot, con ayuda de Flask

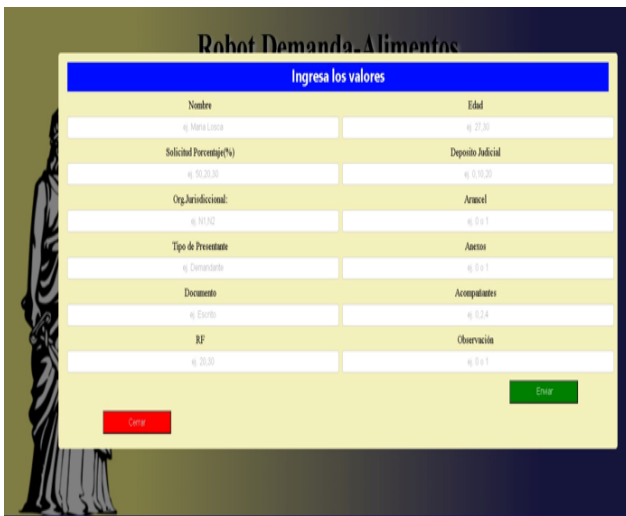


Fig.4 Valores modificables a través de la interfaz



Fig.5 Resultados finales en base al veredicto

## RECONOCIMIENTOS

Reconocimiento a la Universidad Nacional de Ingeniería por el apoyo en el Proyecto de Investigación y al equipo de investigación.

## REFERENCIAS

- [1] Kim, M., Jeon, S., Shin, H., Choi, W., Chung, H., & Nah, Y. (2019). Movie Recommendation based on User Similarity of Consumption Pattern Change. 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), 317–319. <https://doi.org/10.1109/AIKE.2019.00064>
- [2] Delgado, A. (2020). Design of a Web Application for the Detection of Diabetes using Machine Learning. *International Journal of Emerging Trends in Engineering Research*, 8(7), 3974–3978. <https://doi.org/10.30534/ijeter/2020/169872020>
- [3] Varshney, S. (2016). *Building Trading Bots Using Java*. Apress. <https://doi.org/10.1007/978-1-4842-2520-2>
- [4] Scalabrino, S., Mastropaolo, A., Bavota, G., & Oliveto, R. (2021). An Adaptive Search Budget Allocation Approach for Search-Based Test Case Generation. *ACM Transactions on Software Engineering and Methodology*, 30(3), 1–26. <https://doi.org/10.1145/3446199>
- [5] Sonekar, S. v., Singh, P., Koche, P., & Chauvhan, P. (2021). Design and Development of an Intelligent Robot for Improving Crop Productivity Using Machine Learning. In *Studies in Computational Intelligence* (Vol. 960, pp. 63–70). [https://doi.org/10.1007/978-981-16-0598-7\\_5](https://doi.org/10.1007/978-981-16-0598-7_5)
- [6] Xu, N., & Wang, K. J. (2021). Adopting robot lawyer? The extending artificial intelligence robot lawyer technology acceptance model for legal industry by an exploratory study. *Journal of Management and Organization*, 27(5). <https://doi.org/10.1017/jmo.2018.81>
- [7] Rodríguez Oconitrillo, L. R., Vargas, J. J., Camacho, A., Burgos, Á., & Corchado, J. M. (2021). RYEL: An Experimental Study in the Behavioral Response of Judges Using a Novel Technique for Acquiring Higher-Order Thinking Based on Explainable Artificial Intelligence and Case-Based Reasoning. *Electronics*, 10(12), 1500. <https://doi.org/10.3390/electronics10121500>