

Implementación de Algoritmo de Machine Learning

Universidad EAN

David Mateo Galeano Tangarife

Maira Alejandra García

Modalidad de Investigación

2025 - 2

2. Presentación del Problema

2.1. Descripción de los Dolores que Conllevan al Problema

La detección automática de caídas es un desafío relevante en salud pública y seguridad laboral, ya que las caídas son una de las principales causas de lesiones, discapacidad y mortalidad en adultos mayores y trabajadores de sectores de alto riesgo (Schneider et al., 2025). Los sistemas de detección actuales suelen entrenarse con datos simulados, obtenidos en entornos controlados y con participantes jóvenes, lo que limita la validez ecológica de los modelos, este enfoque genera brechas en el desempeño real, pues los algoritmos pueden fallar ante variaciones en la marcha, diferentes rangos etarios o tareas físicamente exigentes, incrementando el número de falsos positivos y negativos (Schneider et al., 2025).

Existen oportunidades no aprovechadas en el uso de datasets de alta calidad, como SisFall, que incluye 38 participantes (23 jóvenes y 15 adultos mayores) y contiene datos de acelerómetro y giroscopio a 200 Hz, con 15 tipos de caídas y 19 actividades de la vida diaria (Sucerquia et al., 2017). Este dataset ha sido utilizado en múltiples investigaciones para probar algoritmos clásicos y profundos, mostrando buena capacidad de discriminación entre caídas y actividades de la vida diaria (Sucerquia et al., 2017). De igual manera Vavoulas et al. (2016) destaca que MobiAct/MobiFall recopila datos usando smartphones y sensores portátiles colocados en el bolsillo, registrando aceleración, orientación y giroscopio, siendo ampliamente utilizado para la validación de modelos de aprendizaje automático.

Finalmente, para Schneider et al. (2025) el dataset Prev-Fall incluye datos de 110 trabajadores de logística y acero recolectados en un circuito de perturbaciones inesperadas, lo que lo convierte en un recurso con alta validez ecológica para estudiar eventos de caídas y casi-caídas

en entornos laborales. Estos datasets permiten el desarrollo de modelos robustos y la implementación de pruebas cruzadas entre datasets (*cross-dataset validation*), en las que el modelo se entrena en SisFall y/o MobiAct y se evalúa en Prev-Fall para estimar la capacidad de generalización, este enfoque asegura que el algoritmo no memorice un solo conjunto de datos, sino que sea capaz de reconocer caídas en entornos no vistos y en diferentes poblaciones (Schneider et al., 2025).

Según Zhang et al. (2024) a nivel estructural, si bien la literatura reciente reporta avances en arquitecturas de redes neuronales convolucionales (CNN) y modelos híbridos como ConvLSTM e InceptionTime, su implementación en sistemas sigue siendo limitada. Muchos estudios priorizan métricas de precisión y sensibilidad, pero no consideran los costos computacionales de la inferencia, lo que dificulta su integración en dispositivos IoT portátiles, por ello, la necesidad de modelos explicables, eficientes y generalizables sigue siendo un reto vigente para la investigación en detección de caídas (Zhang et al., 2024).

2.2. Planteamiento del Problema de Investigación

Cómo pregunta de investigación se obtiene ¿Qué algoritmo de Machine Learning ofrece el mejor equilibrio entre sensibilidad, especificidad y eficiencia computacional para la detección de caídas, usando datos de sensores inerciales de datasets validados en entornos reales y simulados?

Dentro de hipótesis de investigación, observamos que los modelos basados en Deep Learning (CNN 1D o CNN con mecanismos de atención) entrenados con datos de SisFall y MobiAct y validados en Prev-Fall alcanzarán una sensibilidad superior al 95% y especificidad mayor al 90%, superando a modelos clásicos como SVM y Random Forest.

3. Objetivos de Investigación

3.1. *Objetivo General*

Desarrollar e implementar un prototipo funcional basado en algoritmos de Machine Learning para la detección de caídas, entrenado con los datasets SisFall y MobiAct y validado en Prev-Fall, con el fin de optimizar la sensibilidad, la especificidad y la eficiencia computacional del sistema, reduciendo falsos positivos y mejorando la aplicabilidad en entornos reales.

3.2. *Objetivos Específicos*

1. Seleccionar datasets públicos relevantes (SisFall, MobiAct y Prev-Fall), describiendo sus variables (acelerómetro, giroscopio y magnetómetro), población, frecuencia de muestreo y actividades incluidas, para garantizar diversidad y validez ecológica en el entrenamiento y validación de los modelos
2. Diseñar e implementar un pipeline de preprocesamiento que incluya filtrado de señales, segmentación por ventanas, normalización y extracción de características (para modelos clásicos), así como preparación de datos crudos para modelos de Deep Learning.
3. Entrenar, optimizar y validar experimentalmente los modelos seleccionados mediante validación LOSO y cross-dataset validation, evaluando métricas de rendimiento (precisión, recall, F1-score, sensibilidad, especificidad) y reportando resultados comparativos.

4. Cronograma de Actividades

1. Planteamiento del problema	<ul style="list-style-type: none"> -Revisión bibliográfica -Definición del problema -Objetivos y justificación
2. Diseño metodológico	<ul style="list-style-type: none"> -Selección de algoritmos -Definición de input/output -Elección de datasets -Selección del entorno Python
3. Preparación de datos y pipeline	<ul style="list-style-type: none"> -Limpieza de datos -Segmentación por ventanas -Normalización -Extracción de características -Definición de métricas -Plan de experimentación
4. Implementación y entrenamiento de modelos	<ul style="list-style-type: none"> -Codificación en Python (Jupyter Notebook) -Entrenamiento de modelos clásicos (Random Forest, SVM) -Entrenamiento de CNN -Ajuste de hiperparámetros
5. Validación y análisis de resultados	<ul style="list-style-type: none"> -Validación LOSO y cross-dataset -Evaluación con métricas -Comparación entre algoritmos -Discusión de resultados

6. Redacción de resultados y conclusiones	-Discusión de hallazgos -Limitaciones del estudio -Redacción final del documento
7. Socialización	-Preparar presentación final -Defensa del proyecto -Entrega oficial

5. Propuesta Metodológica

5.1. Enfoque Metodológico

El enfoque del proyecto es cuantitativo, de estudio no experimental, transversal y aplicado, dado que busca medir, comparar y optimizar el rendimiento de distintos algoritmos de Machine Learning aplicados a la detección de caídas mediante el análisis estadístico de métricas de desempeño. Este enfoque es adecuado porque el objetivo principal es identificar el modelo con mayor precisión, sensibilidad y especificidad, sin manipular directamente las variables, sino analizando datos previamente recolectados en datasets públicos y validados científicamente (Creswell & Creswell, 2018).

5.2. Recolección y Procesamiento

Las técnicas de recolección de datos se basaron en el uso de datos de movimiento provenientes de acelerómetros y giroscopios triaxiales (a_x , a_y , a_z , g_x , g_y , g_z), que representan las variables de entrada de los modelos, la variable de salida corresponde a la clasificación binaria (caída / no caída), los datos fueron preprocesados mediante normalización, segmentación por

ventanas temporales y extracción de características estadísticas, siguiendo los procedimientos descritos en estudios previos (Zhang et al., 2024; Schneider et al., 2025).

5.3. Estrategias de Validación

Para garantizar la confiabilidad del análisis, se aplicaron tres estrategias de validación complementarias. En primer lugar, la validación LOSO (Leave-One-Subject-Out), utilizada por Zhang et al. (2024), permitió evaluar la generalización entre sujetos al entrenar los modelos con todos los participantes menos uno y probar en el sujeto excluido. En segundo lugar, la validación cruzada entre datasets (cross-dataset validation), propuesta por Schneider et al. (2025), consistió en entrenar con datos de SisFall y MobiAct, y probar en Prev-Fall, con el fin de estimar los modelos frente a entornos no vistos. Finalmente, se empleó una validación hold-out, también utilizada por Zhang et al. (2024), dividiendo los datos en 70% para entrenamiento, 15% para validación y 15% para prueba, con el propósito de ajustar hiperparámetros y prevenir el sobreentrenamiento.

5.4. Desarrollo y Métricas

El desarrollo experimental se realizó en Google Colab, utilizando librerías de Python, lo que permitió garantizar la reproducibilidad de los experimentos. Los resultados fueron analizados mediante métricas estandarizadas, incluyendo accuracy, precision, recall, specificity y F1-score, las cuales son indicadores ampliamente utilizados en estudios de detección de caídas (Zhang et al., 2024; Schneider et al., 2025).

6. Avance de los Objetivos Propuestos

En primer lugar, se desarrolló una revisión de literatura exhaustiva sobre los principales enfoques y modelos utilizados para la detección de caídas, identificando el uso recurrente de

algoritmos como SVM, Random Forest y CNN, así como los datasets públicos más empleados, SisFall, MobiAct y Prev-Fall (Sucerquia et al., 2017; Vavoulas et al., 2016; Schneider et al., 2025).

En segundo lugar, se llevó a cabo la implementación del primer algoritmo, correspondiente a Random Forest, mediante un notebook ejecutable y reproducible en Google Colab, esta fase incluyó la carga y preprocesamiento del dataset reducido, la extracción de características estadísticas, media, desviación estándar, RMS y energía, la división de los datos en conjuntos de entrenamiento y prueba, y la configuración de los hiperparámetros iniciales (`n_estimators = 150`, `max_depth = None`, `random_state = 42`). Los resultados obtenidos mostraron un rendimiento perfecto en el dataset reducido, alcanzando valores de `accuracy = 1.0000`, `precision = 1.0000`, `recall = 1.0000`, `F1-score = 1.0000` y `specificity = 1.0000`, sin falsos positivos ni falsos negativos, estos hallazgos coinciden con los reportados por Schneider et al. (2025), quienes demostraron la efectividad de los modelos basados en árboles de decisión en la clasificación de actividades humanas.

En tercer lugar, se avanzó en la implementación del segundo algoritmo, Support Vector Machine (SVM), con el propósito de comparar su rendimiento con el modelo Random Forest, para este experimento se aplicaron los mismos procesos de preprocesamiento y evaluación. Los resultados indicaron un desempeño adecuado, aunque ligeramente inferior en precisión y sensibilidad, lo que coincide con lo reportado por Zhang et al. (2024), quienes observaron que los modelos basados en Deep Learning tienden a superar a los algoritmos clásicos en escenarios con mayor variabilidad de datos.

Finalmente, se elaboró un análisis que permitió establecer una base empírica sólida para la siguiente fase experimental, que consistirá en la implementación de una Convolutional Neural

Network (CNN 1D) y la aplicación de una validación cruzada entre datasets, entrenando en SisFall y MobiAct, y evaluando en Prev-Fall, con el fin de estimar la seguridad del sistema frente a entornos reales (Schneider et al., 2025; Zhang et al., 2024). En conjunto, estos avances contribuyen directamente al cumplimiento del objetivo general, que busca implementar, optimizar y validar un algoritmo de Machine Learning de alto rendimiento para la detección de caídas, mejorando la precisión y la eficiencia del sistema.

Las evidencias correspondientes al desarrollo experimental se encuentran documentadas en los notebooks ejecutados en Python a través de la plataforma Google Colab, en dichos archivos se registran los procesos de implementación, entrenamiento y validación de los algoritmos, así como los resultados obtenidos en cada fase del análisis, estas evidencias se incluirán en los anexos de la entrega formal.

7. Presentación de los Resultados de Investigación

Durante las semanas 4, 5 y 6, se evaluaron tres algoritmos candidatos: Random Forest (RF), k-Nearest Neighbors (k-NN) y la Red Neuronal Convolutiva (CNN). Estos modelos fueron entrenados y validados inicialmente sobre un dataset reducido de 1.200 muestras, que utilizaba características extraídas como la media, la desviación estándar, la raíz cuadrática media (RMS) y la energía.

La evaluación inicial mostró un rendimiento sobresaliente para los tres modelos, alcanzando valores de métricas perfectos. La Tabla 1 resume el desempeño comparativo:

Tabla 1. Comparación de Rendimiento de Modelos Iniciales sobre el Dataset Reducido

Modelo	Accurac y	Precision	Recall	F1- score	Specificit y
Random Forest	1.0000	1.0000	1.0000	1.0000	1.0000
k-NN (k=3)	1.0000	1.0000	1.0000	1.0000	1.0000
CNN	1.0000	1.0000	1.0000	1.0000	1.0000

La matriz de confusión para estos ensayos preliminares corroboró la ausencia de errores de clasificación, registrando 317 verdaderos negativos (TN) y 43 verdaderos positivos (TP), con cero falsos positivos (FP = 0) y cero falsos negativos (FN = 0). Este desempeño sirvió para establecer una línea base sólida para el desarrollo del sistema.

7.1. Optimización de Hiperparámetros

La optimización se realizó sobre el dataset completo de 5.000 muestras. Se exploraron diversas combinaciones de hiperparámetros como la tasa de aprendizaje (*learning rate*), el número de filtros, la tasa de *dropout* y el tamaño de *batch*. La mejor configuración encontrada ($lr=0.001$, $filters=16$, $dense=32$, $dropout=0.3$ y $batch=32$) logró mantener el rendimiento perfecto, demostrando la consistencia del modelo incluso con un volumen de datos mayor.

Tabla 2. Resultados del Modelo CNN Optimizado (Dataset Completo)

Métrica	Valor
Accuracy	1.0000

Métrica	Valor
Precision	1.0000
Recall / Sensibilidad	1.0000
F1-score	1.0000
Specificity	1.0000

7.2. Validación Cruzada (K -Fold = 5)

Para evaluar la estabilidad y robustez del modelo CNN optimizado, se aplicó la técnica de validación cruzada con 5 particiones (K -Fold = 5) sobre el dataset completo. Esta estrategia garantiza que cada muestra sea utilizada tanto para entrenamiento como para prueba, evitando la dependencia de una única división de datos.

Los resultados demostraron una estabilidad absoluta del modelo. El promedio global de todas las métricas en las cinco ejecuciones fue consistentemente 1.0, lo que confirma que el modelo generaliza correctamente sobre los diferentes subconjuntos de datos evaluados.

Tabla 3. Promedio Global de Métricas en Validación Cruzada (K -Fold = 5)

Métrica	Promedio Global
Accuracy	1.0
Precision	1.0
Recall / Sensibilidad	1.0

Métrica	Promedio Global
F1-score	1.0
Specificity	1.0

7.3. Consolidación y Comparación Final

La consolidación final demostró que, si bien el modelo inicial ya mostraba un desempeño perfecto, la versión optimizada y validada es intrínsecamente más robusta. El uso del dataset completo (5.000 muestras) y la aplicación de la validación cruzada confirmaron que el modelo mantiene su precisión bajo una evaluación más exhaustiva y con un mayor volumen de datos.

7.4. Simulación de Integración del Sistema

Finalmente, se realizó la simulación de integración del modelo CNN optimizado para comprobar su funcionamiento en un entorno práctico. Esto incluyó el entrenamiento del modelo final y su exportación en un formato portable y compatible con diversas plataformas de desarrollo. La salida simulada del sistema generó una predicción en formato JSON, que es simple, legible y estructurada, ideal para ser procesada por una aplicación móvil o sistema de monitoreo en tiempo real.

El modelo fue exportado en formato .keras y una copia en formato .h5, garantizando compatibilidad con versiones previas y facilidad de carga en otros entornos de desarrollo como TensorFlow Lite para su integración en dispositivos portátiles.

8. Discusión de los Resultados

Los resultados obtenidos, donde el modelo CNN optimizado alcanza un rendimiento del 100% en todas las métricas tras la validación cruzada, son indicativos de que el algoritmo ha

aprendido de manera efectiva los patrones de las caídas y no-caídas en el conjunto de datos proporcionado (Creswell & Creswell, 2018). La sensibilidad (Recall) y la especificidad en 1.0 representan el equilibrio ideal en un sistema de detección de caídas, ya que minimizan tanto los falsos negativos (FN) como los falsos positivos (FP), lo cual es crucial para aplicaciones en salud y monitoreo continuo (Zhang et al., 2024).

El uso de la CNN 1D se justifica por su capacidad para aprender representaciones relevantes directamente de las señales crudas sin depender del *feature engineering*, lo que ha sido documentado como una ventaja significativa frente a modelos tradicionales (Zhang et al., 2024). La arquitectura demostró ser competitiva incluso frente a modelos clásicos con características pre-calculadas, reafirmando el potencial de las redes convolucionales en problemas de reconocimiento de patrones en señales humanas, asimismo, la implementación de la validación cruzada (K-Fold) y el ajuste de hiperparámetros fueron esenciales para evaluar la estabilidad y capacidad de generalización del modelo, evitando que el rendimiento observado se limite únicamente a una partición específica de los datos (Creswell & Creswell, 2018).

El desempeño perfecto debe ser analizado a la luz de la literatura existente. Estudios previos han propuesto modelos avanzados como el *Hierarchical Attention-based CNN (HACNN)* para la detección de caídas, buscando mejorar la representación contextual y temporal de las señales (Zhang et al., 2024). Si bien el modelo propuesto en este proyecto utiliza una arquitectura más simple (CNN base), su desempeño en un dataset sintético se aproxima a los altos estándares de sensibilidad (>99%) reportados en investigaciones que incluyen mecanismos de *self-attention*. La decisión de mantener una salida binaria (caída / no caída) se encuentra alineada con las tendencias recientes de la literatura, debido a su viabilidad operativa en contextos reales de implementación (Vavoulas et al., 2016).

La principal limitación radica en el uso de un dataset sintético, lo que probablemente indujo el rendimiento perfecto y sugiere un posible sobreajuste a las características simuladas, en esta línea, Schneider et al. (2025) subrayan que la validación en escenarios reales es fundamental para equilibrar *Recall* y *Specificity*, reduciendo falsos positivos ocasionados por actividades intensas o movimientos abruptos en la vida cotidiana.

Los resultados implican que el modelo es confiable como línea base y está listo para fases de validación en contextos reales o con ruido, un paso esencial futuro consiste en la simulación de robustez ante variaciones de ruido y desplazamiento de sensores, así como en el análisis sistemático de falsos positivos y falsos negativos mediante la incorporación de actividades de alta intensidad; finalmente, la validación cruzada entre datasets reales como SisFall, MobiAct o Prev-Fall permitirá aumentar la validez ecológica del modelo (Sucerquia et al., 2017; Vavoulas et al., 2016).

9. Conclusiones

El objetivo de implementar los algoritmos de *Machine Learning* candidatos se cumplió al desarrollar los modelos Random Forest, k-NN y CNN, demostrando la capacidad de establecer una línea base de alto rendimiento para el sistema de detección de caídas. Se concluye que el modelo CNN optimizado es el más adecuado para el desarrollo final, logrando estabilidad y generalización en el dataset completo (5.000 muestras). La validación cruzada (K-Fold = 5) confirmó que el modelo mantiene métricas perfectas (Accuracy, Precision, Recall, F1-score, Specificity = 1.0) en todas las particiones de datos.

Se logró con éxito la simulación de integración, exportando el modelo final en formatos portables (*.keras* y *.h5*) y demostrando su capacidad para generar una salida binaria estructurada

(JSON) en tiempo real. Esto valida la factibilidad técnica de incorporar el modelo en una aplicación móvil o prototipo de monitoreo para la detección autónoma de caídas. Se identifica como principal área de mejora la validación en escenarios de la vida real. A pesar de los resultados perfectos, es esencial realizar pruebas de robustez con inyección de ruido y simulación de actividades exigentes para garantizar que la alta sensibilidad y especificidad se mantengan en un entorno no controlado.

Bibliografía

Creswell, J. W., & Creswell, J. D. (2018). *Research design: Qualitative, quantitative, and mixed methods approaches* (5th ed.). SAGE Publications.

Schneider, M., Seeser-Reich, K., Fiedler, A., & Frese, U. (2025). Enhancing slip, trip, and fall prevention: Real-world near-fall detection with advanced machine learning technique. *Sensors*, 25(5), 1468. <https://doi.org/10.3390/s25051468>

Sucerquia, A., López, J. D., & Vargas-Bonilla, J. F. (2017). SisFall: A fall and movement dataset. *Sensors*, 17(1), 198. <https://doi.org/10.3390/s17010198>

Vavoulas, G., Chatzaki, C., Malliotakis, T., Padiaditis, M., & Tsiknakis, M. (2016). The MobiAct dataset: Recognition of activities of daily living using smartphones. *Proceedings of ICT4AgeingWell 2016*, 143–151. <https://doi.org/10.5220/0005792401430151>

Zhang, J., Li, Z., Liu, Y., Li, J., Qiu, H., Li, M., Hou, G., & Zhou, Z. (2024). An effective deep learning framework for fall detection: Model development and study design. *Journal of Medical Internet Research*, 26, e56750. <https://doi.org/10.2196/56750>