

En líneas generales

0.1. Precedentes en detección de caídas

0.1.1. Primeros pasos, cálculos basados en cotas

Sobre cálculos basados en cotas, tenemos el FallIndex (?), Kangas(?) lo compara con el uso ya sea de aceleración vertical o modulo de la aceleración y de nuevo obtiene resultados dispares según el tipo de caída simulada. Este índice probablemente lo usemos como algoritmo de comparación dado que no requiere información de postura/posición del acelerómetro.

En (?) se compara el rendimiento de los métodos basados en cotas sobre la aceleración registrada por un acelerómetro usando un teléfono como fuente de medida y análisis. Se desprenden los resultados: Los metodos con buenos resultados en pruebas de laboratorio/controladas no dan buenos resultados en experimentos de uso real y viceversa (se achaca al valor de las cotas, calibradas con demasiada especificidad para el experimento en laboratorio) Los métodos con buena sensibilidad tienen especificidad muy mala y al revés. Si queremos 100 % de sensibilidad la especificidad puede descender hasta el 11 % (Bourke 1LFT) y para un 98 % de especificidad una sensibilidad del 14 % (Kangas1D) Incluso el mejor de los algoritmos, Chen, da unos valores de especificidad 24 % y sensibilidad 94 %. Claramente insuficientes para la detección de caidas. Los moviles se llevan en la mano o en la cintura.

Al ser estos métodos computacionalmente eficientes, son utilizados en sistemas embebidos donde tanto potencia de cálculo como la potencia son altamente restringidos. El acercamiento de usar relojes o pulseras para la captura de movimiento se presenta ya por (?) usando tanto el smartphone como una pulsera para la captura, aunque el procesamiento se realiza en el smartphone. Comprara los resultados con iFall y Fade con resultados muy dispares según el tipo de caída aunque usa algoritmos básicos basados en modulo de la aceleración o FallIndex (crean un dataset llamado project gravity solo con gente joven)

Luque(?) analiza multitud de sistemas de detección de caídas que usan un teléfono inteligente android para al menos una de las etapas (captura de datos, procesado y detección o proxy hacia un servidor externo) que hagan uso o no de terceros dispositivos.

Los algoritmos estudiados son: 1- Monitorizado básico (usar módulo del vector aceleración y un threshold X) 2- Fall Index (Yoshida, T.; Mizuno, F.; Hayasaka, T.; Tsubota, K.; Wada, S.; Yamaguchi, T. A Wearable Computer System for a Detection and Prevention of Elderly Users from Falling. In Proceedings of the 12th International Conference on Biomedical and Medical Engineering (ICBME), Singapore, Singapore, 7–10 December 2005.) 3- PerFallD (Dai, J.; Bai, X.; Yang, Z.; Shen, Z.; Xuan, D. PerFallD: A Pervasive Fall Detection System using Mobile Phones. In Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Mannheim, Germany, 29 March–2 April 2010; pp. 292–297) (usa también un giroscopio) 4- iFall (Sposaro, F.; Tyson, G. iFall: An Android Application for Fall Monitoring and Response. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2009), Minneapolis, MN, USA, 2–6 September 2009; pp. 6119–6122.)

Realmente el monitorizado básico no está tan lejos del resto de métodos e incluso son mejores que algunas implementaciones propietarias comerciales. Recalca: alta variabilidad de los resultados según el tipo de caída (están demasiado optimizados para un tipo concreto, no son de uso general), también hace hincapié en los problemas de usabilidad por parte de usuarios no expertos.

0.1.2. IA para detección de caídas

Usando inteligencia artificial, Shi (?, ?) describe un método basado en acelerómetro triaxial llevado en la cintura para predecir caídas, tratando una caída como una actividad humana más y usando métodos similares a los usados para la predicción de actividad humana (CNN). Casilari (paper ejemplar en estructura) (?, ?) usa una red de 4 capas convolucionales para lo mismo, y compara el rendimiento sobre multitud de datasets obteniendo que si bien se puede optimizar el rendimiento para un conjunto en concreto logrando resultados de precisión y especificidad del orden del 100 %, no es posible generar un modelo que generalice a todos los datasets [estudia también las precisiones variando el tamaño de la ventana y el threshold de potencia del modulo de la aceleración y llega a la conclusión de que ventanas de 1 segundo son las que mejor se comportan, casa con la definición de episodio de los resultados obtenidos del análisis del dataset casero. Hassan (?, ?) mejora los resultados, sin embargo ambos usan gran potencia de cálculo y no es aplicable al uso de sistemas embebidos. Madrano (?, ?) usa K-NearestNeighbours

y SVM para detectar actividad y varios tipos de caídas, mostrando buenos resultados con SVM que además parece ser capaz de distinguir caídas para las que no ha sido entrenado.

Finalmente, en (?, ?) tenemos un resumen con los últimos avances y resultados. Introduce otras fuentes de información como es el riesgo biológico, o predisposición a la caída debido a la edad y otros factores de deterioro de la salud. Muy interesante. CUIDADO!

Un estudio similar al de Luque realiza Aziz(? , ? , ?) Llega a conclusiones similares respecto a la alta variabilidad de los resultados, especialmente en los métodos basados en cotas. Sin embargo encuentra que SVM llega a generalizar incluso con tipos de caídas para las que no había sido entrenado. En la segunda obra analiza en más detalle la detección con SVM usando caídas reales y sugiere usar una combinación de métodos basados en cotas junto con técnicas de IA para obtener mejores estimadores.

0.2. GRU vs LSTM para predicción de series

Respecto al uso de LSTM para detectar anomalías en series temporales, Wang (?, ?) usa LSTM para identificar anomalías en la señal de un motor (aunque el usa el error de reconstrucción de la descomposición y recomposición wavelet de la señal como entrada a una triple red LSTM, nuestro enfoque es el contrario: usar LSTM a modo de transformada wavelet y luego comparar el error de recomposición).

Li(? , ?) usa redes con varias capas LSTM bidireccionales para la predicción de actividad humana y caídas usando acelerómetro y radar, capturando en la muñeca mientras consigue resultados comparables a las mejores técnicas de clasificación.

Qin(? , ?) estudia el comportamiento de varias redes recurrentes para la predicción de la saturación de oxígeno en el agua y obtiene que las GRU son las que mejores predicciones (menos error) obtienen (Sobre LSTM e incluso RNN bidireccional). Kofi (?, ?) compara LSTM y GRU para predecir mercado de valores con topologías muy variadas (número de celdas, de capas, stateless o no) y encuentra que GRU tiene mejor tasa de aciertos teniendo en cuenta el coste computacional (y muchas veces sin tenerlo) y que no siempre tener dos capas de RNN da mejores resultados.

0.3. Pruning y técnicas para reducir complejidad de modelos

TODO