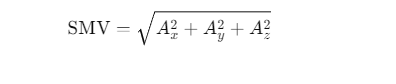
**Tìm hiểu sơ bộ về source code**

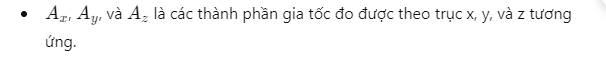
[**SVD\_applied\_fall\_detection**](https://github.com/heeryoncho/SVD_applied_fall_detection)

<https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/el.2018.6117>

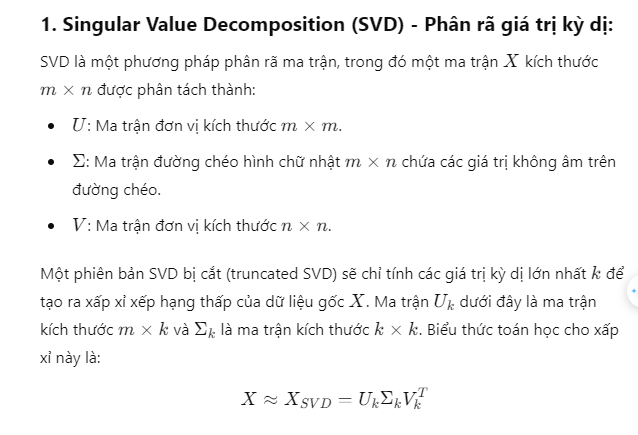
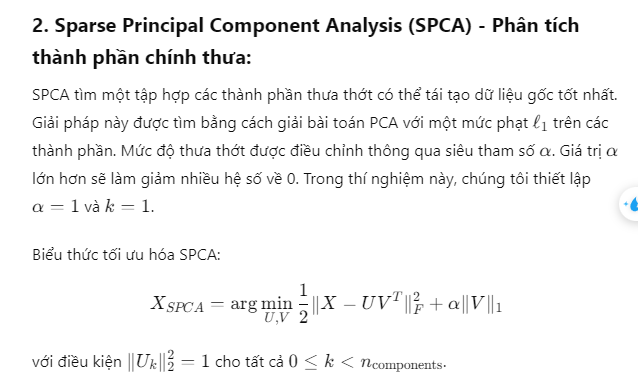
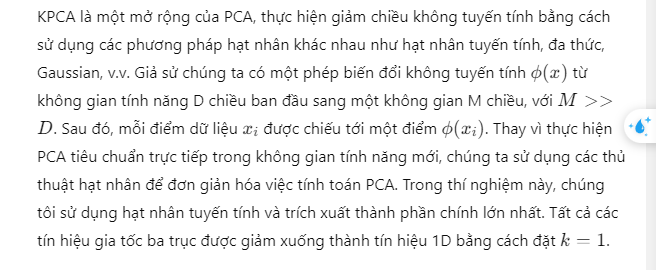
### ****Phương pháp giảm chiều****

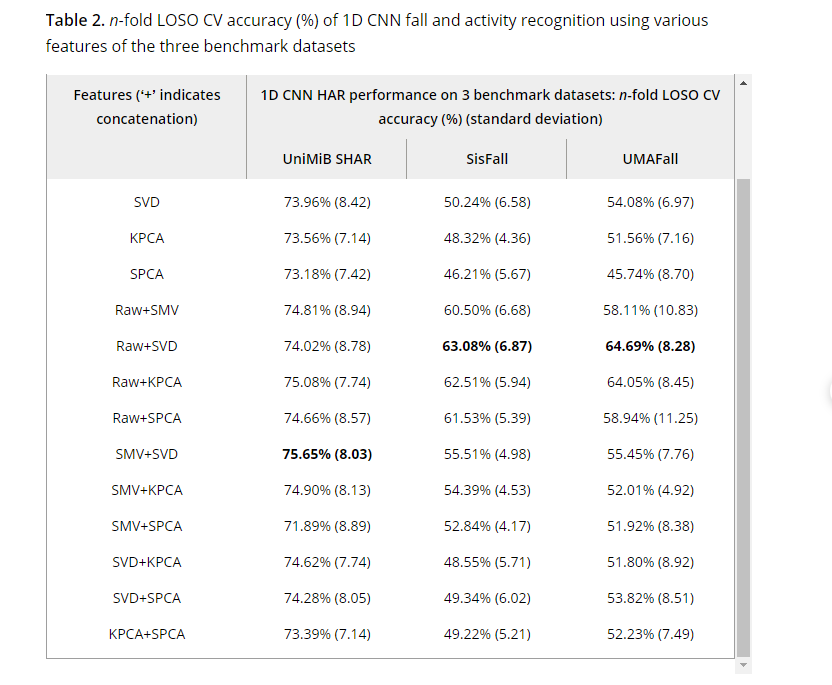
Trước khi trình bày chi tiết về ba phương pháp giảm chiều, chúng ta sẽ định nghĩa một đặc trưng cơ bản gọi là **Signal Magnitude Vector (SMV)**. SMV được tính như là một chuẩn Euclide của các giá trị gia tốc trên các trục x, y, và z. Công thức tính SMV được biểu diễn như sau:





**Các phương pháp giảm chiều chính:**

1. **Singular Value Decomposition (SVD) - Phân rã giá trị kỳ dị:**
   * SVD là một phương pháp phân rã ma trận, trong đó ma trận gốc XXX được phân tách thành ba ma trận: UUU, Σ\SigmaΣ, và VTV^TVT. Đây là một cách hiệu quả để giảm số chiều dữ liệu bằng cách loại bỏ những thành phần ít quan trọng hơn, giữ lại những giá trị kỳ dị lớn nhất.
2. **Sparse Principal Component Analysis (SPCA) - Phân tích thành phần chính thưa:**
   * SPCA là một biến thể của phân tích thành phần chính (PCA) nhằm tìm ra một tập hợp các thành phần có độ thưa thớt tốt nhất có thể tái tạo dữ liệu ban đầu. Phương pháp này kết hợp PCA với một mức phạt thưa thớt (sparsity penalty) để điều chỉnh và giảm thiểu các thành phần không cần thiết.
3. **Kernel Principal Component Analysis (KPCA) - Phân tích thành phần chính sử dụng hạt nhân:**



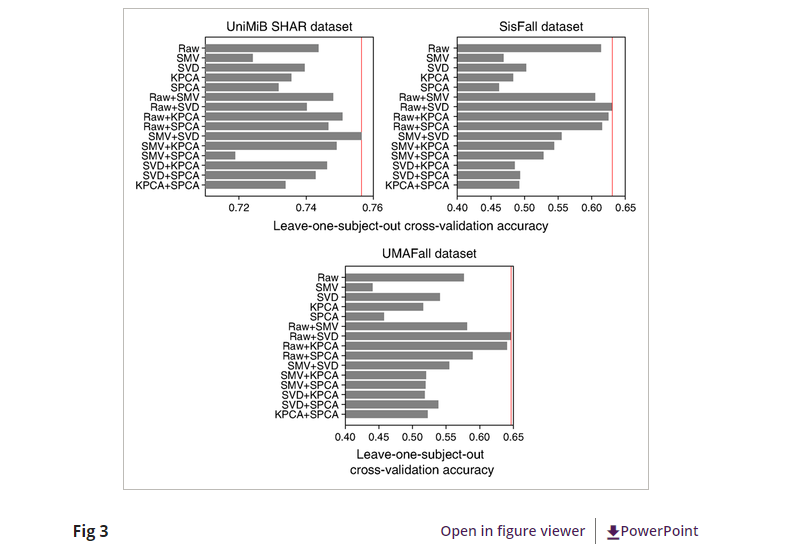
Trong bảng kết quả bạn đã cung cấp (Bảng 2), các giá trị biểu thị độ chính xác trung bình của phương pháp n-fold LOSO CV (Leave-One-Subject-Out Cross-Validation) của mạng CNN 1D dựa trên các đặc trưng tín hiệu khác nhau khi thực hiện nhận dạng hoạt động và phát hiện ngã trên ba bộ dữ liệu chuẩn (UniMiB SHAR, SisFall và UMAFall). Đây là các thông tin cụ thể:

**Giải thích các cột trong bảng:**

1. **Features ('+' indicates concatenation):**
   * Cột này liệt kê các đặc trưng tín hiệu được sử dụng trong thí nghiệm. Các đặc trưng này có thể là tín hiệu gốc (raw), SMV (Signal Magnitude Vector), SVD (Singular Value Decomposition), KPCA (Kernel Principal Component Analysis), SPCA (Sparse Principal Component Analysis), hoặc sự kết hợp giữa hai đặc trưng (ví dụ: Raw+SMV).
2. **1D CNN HAR performance on 3 benchmark datasets: n-fold LOSO CV accuracy (%) (standard deviation):**
   * Các cột này biểu thị hiệu suất của mô hình CNN 1D trong việc nhận dạng hoạt động và phát hiện ngã trên ba bộ dữ liệu chuẩn: UniMiB SHAR, SisFall, và UMAFall.
   * Các giá trị được biểu thị là độ chính xác trung bình (%) và độ lệch chuẩn (trong ngoặc) của phương pháp n-fold LOSO CV. Độ chính xác trung bình cho biết tỉ lệ phần trăm các dự đoán đúng trên tổng số dự đoán, trong khi độ lệch chuẩn chỉ ra mức độ phân tán của kết quả xung quanh giá trị trung bình.

**Cách đọc bảng:**

1. **Baseline CNN on Raw [2]:** Đây là kết quả tham chiếu (baseline) từ một nghiên cứu trước đó sử dụng CNN 2D với đặc trưng tín hiệu thô (Raw). Hiệu suất này được dùng để so sánh với các phương pháp khác.
   * Đối với bộ dữ liệu UniMiB SHAR, baseline đạt 74.97%.
2. **Các đặc trưng tín hiệu và kết quả của chúng:**
   * **Raw:** Đặc trưng tín hiệu thô, khi sử dụng trong CNN 1D cho độ chính xác khác nhau trên từng bộ dữ liệu. Ví dụ: trên UniMiB SHAR, độ chính xác là 74.37% với độ lệch chuẩn 8.07.
   * **SMV:** Đặc trưng tín hiệu từ vector độ lớn tín hiệu, cho kết quả thấp hơn so với Raw, đặc biệt trên bộ dữ liệu UMAFall chỉ đạt 44.04%.
   * **SVD, KPCA, SPCA:** Các phương pháp giảm chiều khác nhau khi sử dụng độc lập thường cho kết quả thấp hơn so với khi kết hợp với đặc trưng tín hiệu thô (Raw). Ví dụ: SVD trên SisFall đạt 50.24%, thấp hơn so với khi kết hợp Raw+SVD (63.08%).
3. **Kết hợp các đặc trưng tín hiệu:**
   * **Raw+SVD, SMV+SVD:** Các tổ hợp này cho kết quả cao nhất. Ví dụ, Raw+SVD trên SisFall đạt 63.08% và trên UMAFall đạt 64.69%.
   * **Raw+KPCA, SMV+SPCA:** Các kết hợp này cũng cải thiện độ chính xác so với các đặc trưng riêng lẻ, nhưng nhìn chung không tốt bằng Raw+SVD.



**Hình 3** cung cấp biểu đồ thanh ngang biểu thị độ chính xác của mô hình CNN 1D sử dụng phương pháp n-fold LOSO CV (Leave-One-Subject-Out Cross-Validation) trong việc nhận dạng hoạt động và phát hiện ngã dựa trên các đặc trưng tín hiệu khác nhau trên ba bộ dữ liệu chuẩn: **UniMiB SHAR**, **SisFall**, và **UMAFall**.

#### Các thông tin từ Hình 3:

1. **Trục ngang**: Biểu thị độ chính xác của các mô hình với các đặc trưng khác nhau.
2. **Trục dọc**: Liệt kê các đặc trưng tín hiệu được sử dụng trong các thí nghiệm, bao gồm các đặc trưng như Raw (tín hiệu gốc), SMV (Signal Magnitude Vector), SVD (Singular Value Decomposition), KPCA (Kernel Principal Component Analysis), SPCA (Sparse Principal Component Analysis) và các kết hợp của chúng.
3. **Đường thẳng đứng màu đỏ**: Chỉ ra độ chính xác cao nhất đạt được trong số các đặc trưng khác nhau.

#### Phân tích từng bộ dữ liệu:

1. **UniMiB SHAR Dataset**:
   * Đặc trưng **SMV+SVD** (75.65%) đạt độ chính xác cao nhất, vượt qua cả các đặc trưng riêng lẻ như Raw (74.37%) và SVD (73.96%).
   * Kết hợp **Raw+KPCA** (75.08%) cũng đạt kết quả tốt, nhưng thấp hơn so với SMV+SVD.
   * Các đặc trưng giảm chiều khác như KPCA và SPCA khi sử dụng riêng lẻ có độ chính xác thấp hơn.
2. **SisFall Dataset**:
   * Kết hợp **Raw+SVD** (63.08%) có độ chính xác cao nhất, tốt hơn nhiều so với sử dụng riêng lẻ các đặc trưng như SMV (46.89%) hay SVD (50.24%).
   * Kết hợp **Raw+KPCA** (62.51%) cũng có kết quả khả quan, nhưng thấp hơn một chút so với Raw+SVD.
   * Các đặc trưng giảm chiều khác như KPCA và SPCA khi sử dụng riêng lẻ đạt kết quả thấp hơn.
3. **UMAFall Dataset**:
   * **Raw+SVD** (64.69%) và **Raw+KPCA** (64.05%) có độ chính xác cao nhất, với Raw+SVD vượt nhẹ hơn.
   * Các đặc trưng giảm chiều như SVD và SPCA khi sử dụng riêng lẻ có độ chính xác thấp hơn (ví dụ: SVD chỉ đạt 54.08%).

**Tổng quan về các thuật toán và quy trình:**

1. **Chuẩn hóa dữ liệu (Normalization) bằng StandardScaler:**
   * **Mục đích**: Chuẩn hóa dữ liệu để các đặc trưng có cùng trung bình (mean = 0) và phương sai (variance = 1), giúp các thuật toán học máy không bị lệch về các đặc trưng có giá trị lớn.
   * **Cách thức**: Sử dụng StandardScaler từ thư viện sklearn.preprocessing để chuyển đổi dữ liệu ban đầu sao cho nó có phân phối chuẩn.
2. **Giảm chiều dữ liệu với Kernel PCA:**
   * **Mục đích**: Sử dụng Principal Component Analysis (PCA) với hàm hạt nhân (kernel) để giảm chiều dữ liệu xuống còn 1 chiều, giúp giảm độ phức tạp tính toán mà vẫn giữ được thông tin quan trọng.
   * **Cách thức**: KernelPCA từ sklearn.decomposition với kernel='rbf' (hàm hạt nhân Radial Basis Function) để tìm ra thành phần chính trong không gian dữ liệu biến đổi.
3. **Giảm chiều dữ liệu với Sparse PCA:**
   * **Mục đích**: Giảm chiều dữ liệu với PCA hiếm (Sparse PCA) nhằm chọn ra các thành phần chính với số lượng nhỏ nhất các đặc trưng, giúp tăng cường độ hiếm của mô hình.
   * **Cách thức**: SparsePCA từ sklearn.decomposition, áp dụng kỹ thuật "Coordinate Descent" (method='cd') để tối ưu hóa việc chọn các thành phần chính.
4. **Giảm chiều dữ liệu với Truncated SVD:**
   * **Mục đích**: Giảm chiều dữ liệu sử dụng phương pháp Singular Value Decomposition (SVD) cắt ngắn để phân tích dữ liệu thành các thành phần có ý nghĩa nhất.
   * **Cách thức**: TruncatedSVD từ sklearn.decomposition để giảm chiều dữ liệu thành 1 thành phần (n\_components=1).
5. **Vector độ lớn tín hiệu (Signal Magnitude Vector - SMV):**
   * **Mục đích**: Tính toán độ lớn tín hiệu từ các trục gia tốc (x, y, z) để chuyển đổi dữ liệu thành một đại diện vô hướng duy nhất giúp đơn giản hóa quá trình xử lý.
   * **Cách thức**: Sử dụng numpy.linalg.norm để tính toán độ lớn (magnitude) của các giá trị vector gia tốc từ các trục.

**Quy trình và các hàm chi tiết:**

1. **Đọc và lọc dữ liệu từ tệp CSV (pandas.read\_csv)**:
   * Đọc dữ liệu từ các tệp CSV chứa thông tin gia tốc từ bộ dữ liệu UMAFall.
   * Sử dụng on\_bad\_lines='skip' để bỏ qua các dòng có lỗi định dạng.
2. **Lọc dữ liệu dựa trên điều kiện (pandas.DataFrame.loc)**:
   * Lọc dữ liệu để chỉ lấy các mẫu được thu từ điện thoại và cảm biến Accelerometer (df.loc[(df[5] == 0) & (df[6] == 0)]).
3. **Chuyển đổi giá trị không hợp lệ thành NaN và xóa chúng (pandas.to\_numeric và DataFrame.dropna)**:
   * Sử dụng pd.to\_numeric(errors='coerce') để chuyển đổi các giá trị không thể chuyển thành số thành NaN.
   * Xóa các hàng chứa giá trị NaN để đảm bảo dữ liệu hợp lệ cho việc xử lý.
4. **Chuẩn hóa dữ liệu (StandardScaler.fit\_transform)**:
   * Áp dụng StandardScaler để chuẩn hóa dữ liệu, giúp mô hình học máy không bị ảnh hưởng bởi sự chênh lệch về thang đo của các đặc trưng.
5. **Giảm chiều dữ liệu với Kernel PCA, Sparse PCA, Truncated SVD**:
   * **Kernel PCA (KernelPCA.fit\_transform)**: Giảm chiều dữ liệu phi tuyến tính với hàm hạt nhân (kernel) để giữ lại các thông tin quan trọng nhất.
   * **Sparse PCA (SparsePCA.fit\_transform)**: Giảm chiều dữ liệu tuyến tính với độ hiếm tối ưu, giúp chọn lọc các thành phần có ý nghĩa nhất.
   * **Truncated SVD (TruncatedSVD.fit\_transform)**: Giảm chiều dữ liệu sử dụng phương pháp cắt ngắn phân tích giá trị kỳ dị để phân tích dữ liệu thành các thành phần có ý nghĩa nhất.
6. **Tạo danh sách các chỉ số mẫu (numpy.arange và numpy.array.reshape)**:
   * Tạo các danh sách chỉ số mẫu (idx) để chọn các mẫu dữ liệu từ các cửa sổ thời gian.
   * Sử dụng reshape để định dạng lại các mẫu thành dạng cần thiết cho việc xử lý tiếp theo.
7. **Lưu trữ dữ liệu đã xử lý (pickle.dump)**:
   * Lưu dữ liệu đã xử lý (X) và nhãn (y) vào các tệp sử dụng pickle để sử dụng trong các bước xử lý sau này.

Quy tắc đặt tên từng các file model  
umafall\_[dữ liệu đầu vào]\_[thuật toán]\_[mô hình]\_[phiên bản]  
  
**LOSO CV** là viết tắt của **Leave-One-Subject-Out Cross-Validation**. Đây là một kỹ thuật kiểm tra chéo (cross-validation) đặc biệt được sử dụng trong các bài toán học máy liên quan đến dữ liệu từ nhiều đối tượng khác nhau.  
**accurate**1.raw\_model: 0.56 xấp xỉ 0.576

2.smv\_raw\_model: xấp xỉ 0.576

3.raw\_kpca\_model: xấp xỉ 0.52 đến 0.5478

4.raw\_spca\_model:fail

5.raw\_svd\_model:fail

6.kpca: xấp xỉ 0.4

7.kpca\_spca:0.53 đến 0.62

8. svd:0.50

9. spca:0.56

10. smv:0.44

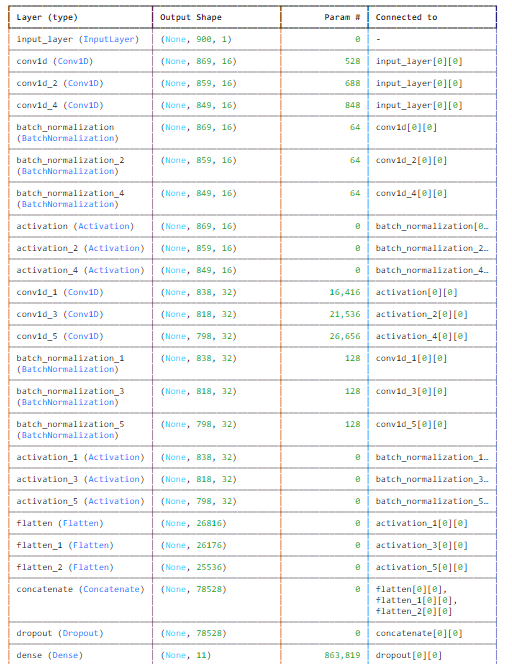
11. smv\_svd:0.51

12. svd\_kpca:0.52

13. svd\_spca:0.53

14. smv\_kpca:0.41

15. smv\_spca:



Cấu trúc bảng trên   
**Layer (type)**: Loại và tên của từng lớp trong mô hình, ví dụ như InputLayer, Conv1D, BatchNormalization, Activation, Flatten, Dense, v.v.

**Output Shape**: Hình dạng của đầu ra (output) từ mỗi lớp. Đây là số lượng mẫu (samples) và kích thước của mỗi mẫu dữ liệu sau khi đi qua lớp đó. Ví dụ, (None, 900, 1) có nghĩa là lớp đầu vào (input layer) có số lượng mẫu không xác định (None) và mỗi mẫu có kích thước 900x1.

**Param #**: Số lượng tham số (parameters) có thể học được của mỗi lớp. Các tham số này bao gồm trọng số (weights) và độ lệch (biases) của các lớp, đóng vai trò quan trọng trong việc điều chỉnh mô hình để phù hợp với dữ liệu đầu vào.

**Connected to**: Thông tin về kết nối giữa các lớp trong mô hình. Nó cho biết mỗi lớp được kết nối với lớp nào trước đó hoặc đầu vào nào. Ví dụ, conv1d\_1[0][0] cho biết lớp này được kết nối từ đầu ra của lớp conv1d\_1.

### Ý nghĩa của một số lớp chính trong mô hình:

* **InputLayer**: Lớp đầu vào, nhận dữ liệu đầu vào với kích thước được xác định bởi người dùng, trong trường hợp này là (None, 900, 1).
* **Conv1D (Convolutional 1D Layer)**: Lớp tích chập 1D được sử dụng để trích xuất các đặc trưng từ dữ liệu đầu vào. Các lớp này có tham số học được (số lượng tham số tương ứng với số lượng filters và kích thước của kernel) để phát hiện các mẫu khác nhau trong dữ liệu.
* **BatchNormalization**: Lớp chuẩn hóa dữ liệu đầu ra của các lớp tích chập để tăng tốc độ huấn luyện và cải thiện độ ổn định của mô hình.
* **Activation (ReLU)**: Lớp kích hoạt được sử dụng để giới thiệu phi tuyến tính vào mô hình, cho phép mạng học các mối quan hệ phi tuyến tính từ dữ liệu. ReLU (Rectified Linear Unit) là một hàm kích hoạt phổ biến.
* **Flatten**: Lớp chuyển đổi đầu ra 3D từ các lớp tích chập thành một vector 1D, chuẩn bị cho việc đầu vào vào các lớp kết nối đầy đủ (Dense layers).
* **Concatenate**: Lớp kết hợp nhiều đầu vào hoặc đầu ra từ các lớp khác nhau, tạo thành một đầu ra kết hợp.
* **Dropout**: Lớp bỏ bớt ngẫu nhiên một tỷ lệ phần trăm của các đơn vị kết nối (neurons) trong một lớp để ngăn ngừa quá khớp (overfitting).
* **Dense**: Lớp kết nối đầy đủ (Fully connected layer) thường được sử dụng ở cuối mạng để tổng hợp tất cả các đặc trưng học được thành các kết quả đầu ra, ví dụ như phân loại các nhãn đầu ra.