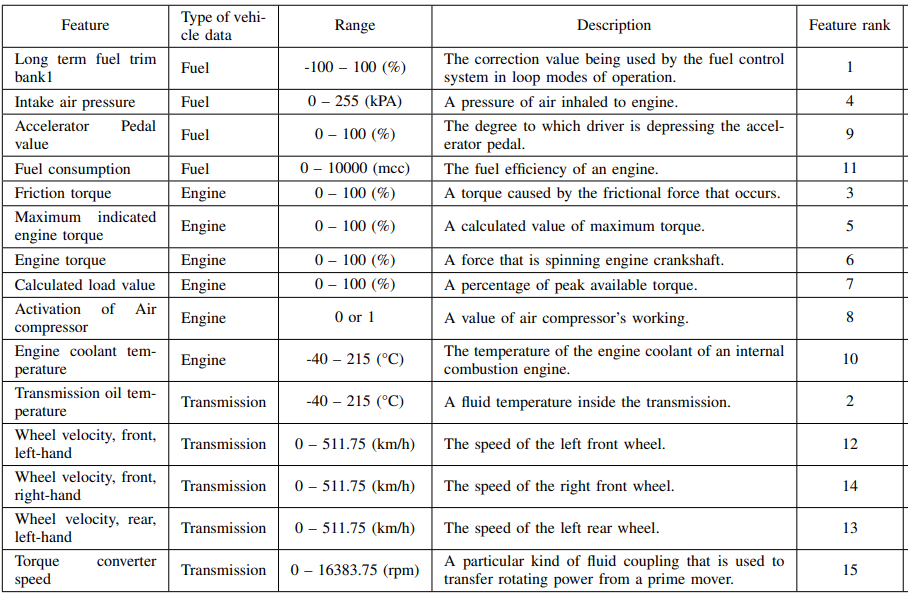
**Supervised**

**Know Your Master: Driver Profiling-based Anti-theft Method(2017)**

**Features: (InfoGainAttributeEval)**

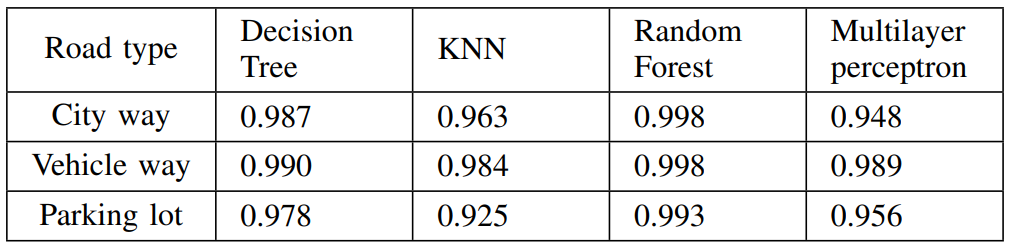
****

**+ Statistical features:** To minimize the effect of fluctuation of feature values and characterize the feature distribution effectively, they adopt the statistical values including mean, median, and standard deviation. They also adopt the median for highly skewed values. The statistical features are derived every period determined by sliding window.

*Thus, a record has 15 original features and 45 statistical features in total.*

*a window length of 60.*

**Average accuracy of driver identification:**

****

**Algorithm: Decision Tree** **Definition:** The decision tree is a classification algorithm where data is recursively divided into smaller parts based on the attributes that guarantee the highest information gain.

**Algorithm: Random Forest** **Definition:** Random forest is an ensemble learning method consisting of multiple decision trees. It addresses the over-fitting problem of individual decision trees by averaging their predictions.

**Algorithm: k-Nearest Neighbors (KNN)** **Definition:** KNN is a classification algorithm that assigns a class to a data point based on the majority class of its k-nearest neighbors, determined by distance.

**Algorithm: Multi-layer Perceptron (MLP)** **Definition:** MLP is a type of artificial neural network that generates hyperplanes by fully connecting nodes in lower layers to nodes in higher layers, enabling complex classification.

**Unsupervised**

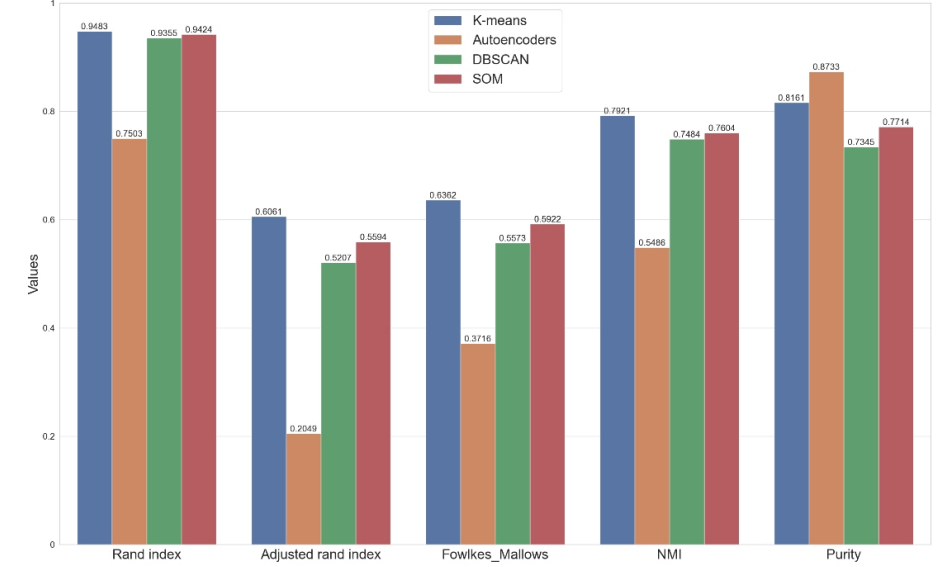
**USID - Unsupervised Identification of the Driver for Vehicle Comfort Functions(2024)**

**Features: (SelectKBest)**

engine coolant temperature, fuel level, ambient air temperature, latitude, longitude, altitude, engine rpm, air intake temperature, vehicle speed, and timing advance

**the Uniform Manifold Approximation and Projection (UMAP**) method was used to reduce the dimensionality of the data, The UMAP is a nonlinear dimensionality reduction technique used for embedding high-dimensional data into a lower-dimensional space while preserving the underlying structure and relationships within the data points.

*a window length of w = 30 with a stride of s = 1 is used.*

******

**Algorithm: k-means clustering** **Definition:** K-means clustering is an unsupervised machine learning algorithm used to partition a dataset into K distinct, non-overlapping clusters. The algorithm iteratively assigns data points to clusters based on their proximity to the mean of the cluster.

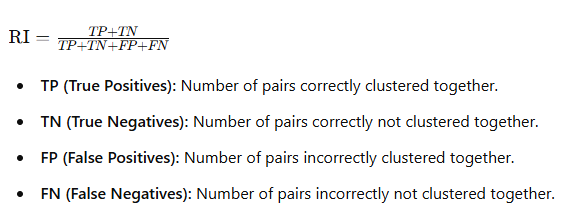
**Algorithm: Autoencoder** **Definition:** Autoencoders are a type of neural network used for feature extraction and dimensionality reduction. They consist of an encoder and a decoder that compress and reconstruct input data, respectively. In clustering, the encoder extracts relevant features from input sequences without requiring labels.

**Algorithm: DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** **Definition:** DBSCAN is a clustering algorithm that identifies clusters in noisy and irregularly shaped datasets. It uses two parameters, eps (the maximum distance for neighbors) and min-samples (the minimum number of points to form a cluster), to define dense regions and outliers.

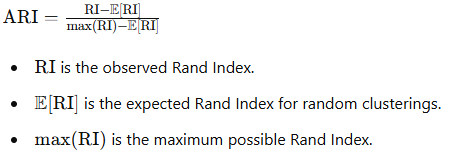
**Algorithm: Self-organizing maps (SOMs)** **Definition:** SOMs are a type of artificial neural network that transforms high-dimensional data into a two-dimensional map or grid. Each neuron in the map represents a cluster, allowing for visual representation and clustering of the data.

**Metric:**

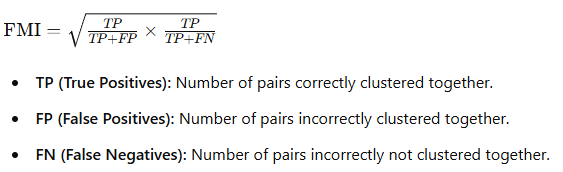
**Rand Index** **Definition:** The Rand Index measures the similarity between two clusterings by comparing the assignment of pairs of points. It accounts for the number of pairs that are either in the same or different clusters in both clusterings.



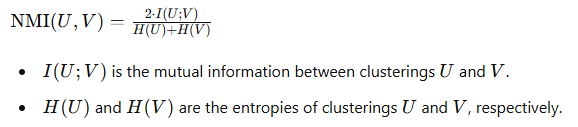
**Adjusted Rand Index** **Definition:** The Adjusted Rand Index corrects the Rand Index for the chance grouping of elements, providing a measure of similarity that accounts for random clustering.

******

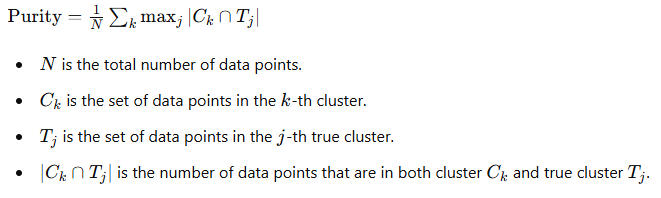
**Fowlkes-Mallows Index** **Definition:** The Fowlkes-Mallows Index evaluates the similarity between two clusterings by calculating the geometric mean of the precision and recall of the cluster pairs.

******

**Normalized Mutual Information (NMI)** **Definition:** NMI is an entropy-based metric that measures the similarity between two clusterings by quantifying the amount of shared information between them.

******

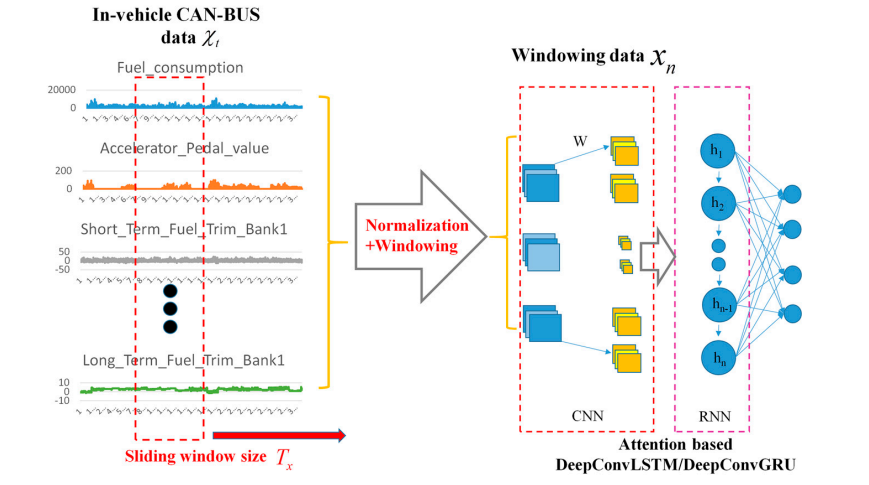
**Purity** **Definition:** Purity quantifies the degree of homogeneity within clusters by evaluating the extent to which each cluster contains only data points from a single true cluster, without considering the relationships between clusters.

******

**Deep Learning**

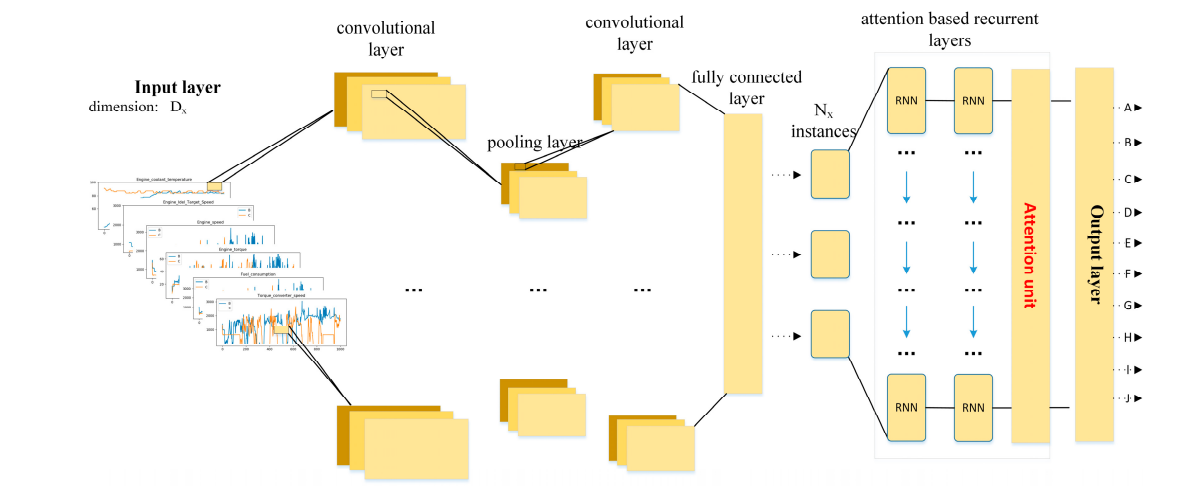
The in-vehicle’s CAN data include the steering wheel, vehicle speed, engine speed, brake position, etc., works on the same dataset.

This work used all 51 original features in the dataset without complex feature selection.

******

The architecture introduces an attention mechanism to enhance feature extraction from time series data.

1. **Input Layer**: Processes the CAN-BUS sensor data.
2. **Middle Layers**:
   * **Convolutional Layers**: Depth-wise separable convolutions extract features from input signals.
   * **Pooling Layers**: Reduce the dimensionality of feature maps to avoid overfitting.
   * **Recurrent Layers**: Utilize attention mechanisms for time series feature extraction.
3. **Output Layer**: Produces class probability distributions for driving behavior identification.



**Key models:**

**A Deep Learning Framework for Driving Behavior Identification on In-Vehicle CAN-BUS Sensor Data(2019)**

**DeepConvGRU–Attention**

* **Architecture**: Combines convolutional layers, GRU layers, and attention mechanisms.
  + **Layers**: (1 × 60) − 1 × 20 − 1 × 6 − 128 − 128 − 10
  + **Convolutional Layers**: Extract features from the input data.
  + **GRU Layers**: Capture temporal dependencies in the data.
  + **Attention Mechanism**: Focuses on relevant parts of the input sequence.
  + **Output Layer**: Produces class probability distributions.
* **Dropout**: 0.5
* **Activation Function**: ReLU
* **Optimizer**: Adam

**DeepConvLSTM–Attention**

* **Architecture**: Combines convolutional layers, LSTM layers, and attention mechanisms.
  + **Layers**: (1 × 60) − 1 × 20 − 1 × 6 − 128 − 128 − 10
  + **Convolutional Layers**: Extract features from the input data.
  + **LSTM Layers**: Capture temporal dependencies in the data.
  + **Attention Mechanism**: Focuses on relevant parts of the input sequence.
  + **Output Layer**: Produces class probability distributions.
* **Dropout**: 0.5
* **Activation Function**: ReLU
* **Optimizer**: Adam

**DeepConvGRU**

* **Architecture**: Combines convolutional layers and GRU layers without attention mechanisms.
  + **Layers**: (1 × 60) − 1 × 20 − 1 × 6 − 128 − 128
  + **Convolutional Layers**: Extract features from the input data.
  + **GRU Layers**: Capture temporal dependencies in the data.
  + **Output Layer**: Produces class probability distributions.
* **Dropout**: 0.5
* **Activation Function**: ReLU
* **Optimizer**: Adam

**DeepConvLSTM**

* **Architecture**: Combines convolutional layers and LSTM layers without attention mechanisms.
  + **Layers**: (1 × 60) − 1 × 20 − 1 × 6 − 128 − 128
  + **Convolutional Layers**: Extract features from the input data.
  + **LSTM Layers**: Capture temporal dependencies in the data.
  + **Output Layer**: Produces class probability distributions.
* **Dropout**: 0.5
* **Activation Function**: ReLU
* **Optimizer**: Adam

**CNN**

* **Architecture**: Convolutional neural network.
  + **Layers**: (1 × 60) − 1 × 20 − 1 × 6
  + **Convolutional Layers**: Extract features from the input data.
  + **Output Layer**: Produces class probability distributions.
* **Dropout**: 0.5
* **Activation Function**: ReLU
* **Optimizer**: Adam

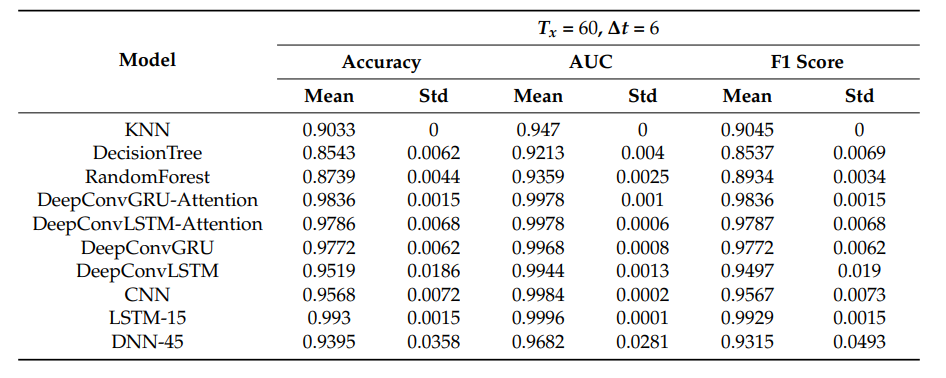
**LSTM**

* **Architecture**: Long short-term memory network.
  + **Layers**: 128 − 128
  + **LSTM Layers**: Capture temporal dependencies in the data.
  + **Output Layer**: Produces class probability distributions.
* **Dropout**: 0.5
* **Activation Function**: ReLU
* **Optimizer**: Adam

**DNN**

* **Architecture**: Deep neural network.
  + **Layers**: 1000 − 1000
  + **Fully Connected Layers**: Process the input data and produce class probability distributions.
* **Dropout**: 0.5
* **Activation Function**: ReLU
* **Optimizer**: Adam

‘(1\*60)’ represents the kernel size of input-to-state convolutional layer. ‘1\*20’ and ‘1\*6’ represent the corresponding kernel sizes of state-to-state convolutional layer and pooling layer. ‘128’ refers to the number of hidden states in the recurrent layers while ‘10’ represents the size of attention vector. ‘1000’ refers to the number of hidden states in the hidden layers of DNN.

******