AutoSensing: Automated Feature Engineering and Learning for Classification Task of Time-Series Sensor Signals

Arijit Ukil, Soma Bandyopadhyay
TCS Research and Innovation
Tata Consultancy services, India
{arijit.ukil, soma.bandyopadhyay}@tcs.com

Abstract

The ever-growing need for machine learning systems demands automation of machine learning pipeline for classification task of time-series sensor signals considering the widespread use and availability of sensor-based systems for Internet of Things (IoT) applications. We present AutoSensing, which automates the machine learning pipeline with the principal focus on automated feature engineering and learning that automatically generates rich feature set and subsequently discovers a distinct feature subset along with constructing the learned model. One of the main difficulties in feature engineering is automated feature subset selection under practical computational time. We present a novel meta-heuristic search for feature subset selection that effectively discovers the distinct feature subset by minimizing the irrelevant search space. AutoSensing is a complete sensor-agnostic machine learning pipeline. We demonstrate the performance efficacy of our scheme using a number of publicly available diverse standard time series datasets from heterogeneous sensors and sources. AutoSensing consistently outperforms the state-of-the-art methods like AutoML and related deep learning based network.

1 Introduction

The wide-scale deployment of Internet of Things (IoT) applications along with the availability of large number of wearable devices, as well as the natural accessibility of time series signals influence us to develop analytics solution for time series signals. However, time series signals in IoT applications are generated from heterogeneous sources and are applied in diverse applications. For example, smart healthcare applications involve number of sensor signals like Electrocardiogram (ECG), accelerometer, Electroencephalogram (EEG), and many others. These sensor signals along with powerful analytics algorithms construct a smart healthcare eco-system to detect or predict different health disorders like Atrial fibrillation, fetal health condition. Similarly, other IoT applications like smart energy management [Ukil. et al. 2014]. One of the main challenges for practical deployment of such IoT applications is the necessity of rapid development and timely deployment of larger spectrum of applications with little or no expert intervention under practical constraint of limited computational time (equivalent to speedy training process).

There are different issues and challenges to construct end-to-end automated machine learning pipeline for heterogeneous sensor signals. We observe that expert involvement in the learning process and expert-intervened feature selection are expensive, time consuming and affects scalability. The salient aspects of AutoSensing are: 1. Generation of rich features that can characterize the time-series sensor signals in different domains like statistical, spectral, wavelet, 2. Searching for the feature subset, which is to be completed without human intervention and under practical computational time, 3. Integrated feature engineering with model construction.

AutoSensing automatically generates rich feature set $\mathbb{F}_{super}$ by different types of transformations and subsequently discovers distinct feature subset $\mathbb{F}_{Distinct} \subseteq \mathbb{F}_{super}$. It discovers the feature subset ($\mathbb{F}_{Distinct}$) by our novel meta-heuristic search which searches efficiently and effectively over feature superset $\mathbb{F}_{super}$ and suggest trained model $\mathbb{M}_{Suggested} = \{\mathbb{F}_{Distinct}, \mathbb{H}_{suggested} \in \mathbb{H}\}$, ($\mathbb{H}$ is the classifier or learner, e.g. Random Forest, Supprt Vector Machine) without human-in-loop and with linear computational complexity: $\mathcal{O}(\Pi)$, Π= number of features in $\mathbb{F}_{super}$. In fact, AutoSensing performs the following pipeline: Feature generation → Feature selection → Model selection → Learned (Trained) model construction.

Current automated machine learning approach like AutoML [Feurer et al., 2015] or PoSH Auto-sklearn [Feurer et al., 2018] focuses on feature selection from feature superset generated elsewhere. Unless feature generation process is an integral part of the analytics method and feature subset selection is computationally light, practical purposes of sensor signal analytics scalability would not be served and automation of learning process would be incomplete.

Learning model construction with inbuilt representation mapping by deep learning based networks [Palaz et al., 2015] require good number of hand-tuned hyperparameter setting. Our experimentations over diverse domain and type of sensor signals demonstrate that AutoSensing consistently outperforms the relevant state-of-the-art methods [Feurer et al., 2015], [Palaz et al., 2015].

2 Problem Formulation

We formalize few definitions.

Definition 1. $\mathcal{Z}_{train} = \{x_q, y_q\}$, $q = 1,2, \dots, Q$, x_q be the time-series sensor signals. $x_q \in \mathbb{R}^T$, class labels $y_q \in [1, \mathcal{C}]$.

Definition 2. The training set consists of model generation set $\mathcal{Z}_{model-generat}$ and model validation $\mathcal{Z}_{validate}$, $\mathcal{Z}_{train} = \mathcal{Z}_{model-generation} + \mathcal{Z}_{validate}$. Let, $\mathcal{Z}^k_{validate} \subset \mathcal{Z}_{train}$, $\mathcal{Z}^k_{model-generat} \subset \mathcal{Z}_{train}$, $\mathcal{Z}^k_{model-gener} + \mathcal{Z}^k_{validate} = \mathcal{Z}_{train}$, $k = 1,2, .. K$, $\mathcal{Z}^k_{validate}$ is the validation and $\mathcal{Z}^k_{model-generat}$ is model generation datasets for each of the stratified *K*-fold cross-validation.

Definition 3. Unsupervised feature space $\mathbb{F}_{super} = \{\mathbb{F}_1, \mathbb{F}_2, \dots, \mathbb{F}_\Pi\} \in \mathbb{R}^\Pi$ transforms each of the input time series to Π dimension vectors: $\mathbb{R}^T \rightarrow \mathbb{R}^\Pi$.

Definition (AutoSensing): AutoSensing discovers feature subset $\mathbb{F}_{Distinct} \subseteq \mathbb{F}_{super}$, $\mathbb{F}_{Distinct} \in \mathbb{R}^\pi, \pi \leq \Pi$ and suggests model $\mathbb{M}_{Suggested} = \{\mathbb{F}_{Distinct}, \mathbb{H}\}$, ($\mathbb{H}$ is the classifier, e.g. Random Forest) such that:

$$\mathbb{M}_{Suggested} = arg \max_{\mathbb{M}} \frac{1}{K} \sum_{k=1}^{K} \mathbb{P}(\mathbb{M}(\mathcal{Z}^k_{model-generat}), \mathcal{Z}^k_{validate}) \quad (1)$$

Where, $\mathbb{P}$ is the reward function ($\mathbb{P}$ can be accuracy, sensitivity, specificity) and $\mathbb{P}$ is computed when model $\mathbb{M}$ is constructed with training set $\mathcal{Z}_{model-generat}$ and validated by the validation set $\mathcal{Z}_{validate}$ for all k cross-validation folds; under the constraints: human effort budget $\mathcal{B} = 0$, training or learning time $\mathcal{T}_{learn} \leq \mathbb{T}$.

Our approach is to find the distinct feature set $\mathbb{F}_{Distinct}$ (with the given classifier $\mathbb{H}$) from feature superset $\mathbb{F}_{super}$ that on the average maximizes the given award function (say, accuracy or sensitivity) over stratified *K*-fold cross-validation. It is shown in [Komiyama. et al. 2018] that structural estimation for model selection is properly demonstrated in cross-validation investigation. Cross-validation is unbiased as $\mathcal{Z}^k_{model-generat}$ is independent of $\mathcal{Z}^k_{validate}$.

3 AutoSensing Scheme

AutoSensing scheme broad view is illustrated in Figure 1. From the given training sensor signals $\mathcal{Z}_{train}$, feature superset $\mathbb{F}_{super}$ is generated. Subsequently feature subset $\mathbb{F}_{Distinct} \subseteq \mathbb{F}_{super}$ is discovered and corresponding model $\mathbb{M}_{Suggested} = \{\mathbb{F}_{Distinct}, \mathbb{H}\}$ is constructed, $\mathbb{H}$ is the classifier, e.g. Random Forest. AutoSensing requires only the training dataset $\mathcal{Z}_{train}$ to construct $\mathbb{M}_{Suggested}$. Reward function $\mathbb{P}$ is user choice. By default $\mathbb{P}$ is accuracy metric.

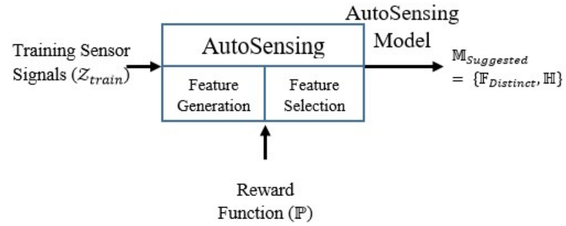

Figure 1. AutoSensing Architecture- Illustration.

4 Automated Feature Generation

In this Section, we briefly illustrate the feature generation procedure, where we implement rich signal processing and statistical functions to generate $\mathbb{F}_{super} = \{\mathbb{F}_1, \mathbb{F}_2, \dots, \mathbb{F}_\Pi\} \in \mathbb{R}^\Pi$ from time-series training sensor signals x_q, $\forall q, q = [1,2, \dots, Q]$, $x_q \in \mathbb{R}^T$such that: $\mathbb{R}^T \rightarrow \mathbb{R}^\Pi$. $\mathbb{F}_{super}$ consists of micro-features (where x_q is segmented with each segment consists of (approximately) $\mathcal{W}$ number of samples: $x_q = x_q^{segmented} = (\overbrace{x_q^1, x_q^2 \dots, x_q^{\mathcal{W}}}^{segment\ \#\ 1}, \overbrace{x_q^{\mathcal{W}+1}, x_q^{\mathcal{W}+2} \dots, x_q^{2\mathcal{W}}}^{segment\ \#\ 2}, \dots)$ and macro-features (where complete x_q is considered). We exploit the spectral, temporal and wavelet domain characterization to generate the feature space $\mathbb{F}_{super}$ [Gray and Davisson, 2010]. Micro-features are capable of extracting the non-stationarity of the time-series. As depicted in Figure 2, first x_q is segmented (Layer 1). Next in Layer 2, base transformation (signal processing functions: β number of functions) are applied to get the micro-features. In Layer 3, statistical functions are applied on Layer 2 output (micro-feature generation), or directly on the time series (macro-feature generation).

Examples of the features in $\mathbb{F}_{super}$: 1. variance (FFT-coefficients (x_q)): *macro-feature*, 2. $median\left(Kurtosis\left(x_q^{seg}\right)\right)$: Kurtosis of each of the segments is calculated and mean operation is performed over it: *micro-feature* 3. Shannon entropy (x_q): *macro-feature*.

The above description of feature generation is specific for time-series sensor signal classification task. The feature superset generation depends on the type of the data or signal. Different dimensionality reduction techniques like Principal Component Analysis (PCA) might be useful for other types of datasets. Moreover, we intend to emphasize that in presence of noise in x_q, micro-features have the ability to minimize the signal distortion. For example, number of segments in x_q is 20 and three of the segments are distorted due the presence of noise. Features like

$median\left(Kurtosis(x_q^{seg})\right)$ has minimum impact due to the presence of such distortion.

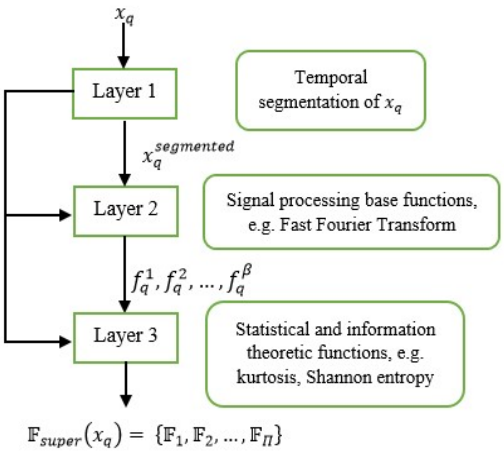

Figure 2. Automated Feature Superset ($\mathbb{F}_{super}$) Generation.

Base Transformation at Layer 2	**Statistical Functions at Layer 3**
• Spectral centroid • Hjorth complexity • MFCC • Daubechies 4 wavelet transform • FFT co-efficients • Time-series samples (x_q)	• Mean • Variance • Kurtosis • Skewness • Shannon entropy • Ljung–Box test

Table 1. Few Examples of the Functions Used for Feature Superset ($\mathbb{F}_{super}$) Generation.

4 Metaheuristic Searching for Feature Selection: Patently Greedy Search

One of our main contributions is developing a novel meta-heuristic search based feature subset selection method that maximizes the exploitation of the relevant search spaces and minimizes the exploration of unreliable searching spaces.

In order to reduce the computational time for searching, greedy search algorithms like forward selection, backward elimination were proposed [Kohavi and John. 1997]. All of the greedy search methods suffer from nesting effect. In the greedy search algorithms, each of the features are sequentially examined and eliminated if the addition (for forward selection) or deletion (for backward elimination) of the feature does not yield a better outcome. Features once discarded cannot be considered later, which results in sub-optimal feature combination selection.

We propose our novel patiently greedy searching algorithm that is computationally as efficient as the greedy search ($\mathcal{O}(\Pi)$), while minimizing the nesting effect. It is a metaheuristic search, where 1. We guide the searching procedure by eliminating the irrelevant search space, 2. We do not discard the features completely, instead features are investigated over a window (patience window Λ). Thus, nesting effect is diminished.

Elimination of irrelevant search space: Initially, we find for the relevant search space with the intuition that a ranked or ordered feature set would provide better exploration of the search space. The initial guess of representation strength is derived using filter based techniques (independent of learner methods) like minimum Redundancy Maximum Relevancy (mRMR) [Peng et al., 2005]. $mRMR$ satisfies relevancy property $\mathbb{A}$ and redundancy property $\mathbb{B}$ from the set of features $\mathbb{F}_{super}$ to the target class $y_q, \forall q$, $\mathcal{Y} = [y_1, y_2, \dots, y_Q]$ based on mutual information (mutual information: $\mathbb{I}(x, y) = \sum_{x\in X}\sum_{y\in Y} p(xy)\log\frac{p(xy)}{p(x)p(y)})$, where $\mathbb{A} = \frac{1}{\Pi}\sum_{\mathbb{F}_i \subseteq \mathbb{F}_{super}} \mathbb{I}(\mathbb{F}_i, \mathcal{Y})$, $\mathbb{B} = \frac{1}{\Pi^2}\sum_{\mathbb{F}_i \subseteq \mathbb{F}_{super}} \mathbb{I}(\mathbb{F}_i, \mathbb{F}_j)$, $\mathbb{F}_{ordered} = \arg\max_{\mathbb{F}_{super}} (|\mathbb{A} - \mathbb{B}|)$. There exists bijection from $\mathbb{F}_{super}$, $\mathbb{F}_{super} \xrightarrow{mRMR} \mathbb{F}_{ordered}$. We can safely assume that the features with higher ranking in $\mathbb{F}_{ordered}$ characterizes the distribution of $\mathcal{Z}_{train}$ more properly with high probability.

Exploring relevant search space: Feature selection, i.e. finding $\mathbb{F}_{Distinct}$ from $\mathbb{F}_{Super}$ is performed using wrapper method, where a feature is selected with respect to its performance over the available training examples by a certain classifier ($\mathbb{H}$). Patiently greedy search searches over $\mathbb{F}_{ordered}$ such a way that feature elimination is not instantaneous. We define a patience window $\Lambda(\Pi)$, where $\Lambda = \lceil\beta\sqrt{\Pi}\rceil$, $\beta = (0, 1]$, $\Lambda \in \mathbb{Z}^+$. An initial low performing feature would not be discarded entirely, and that feature would be selected if that feature in combination of other features demonstrate incremental performance gain within the patience window.

Let, $\mathbb{M}^i = \{\mathbb{F}_{ordered}^i, \mathbb{H}\}$ be trained on $\mathcal{Z}_{model-generat}^k$ and validated $\mathcal{Z}_{validate}^k$; reward function $\mathbb{P}$ (say, accuracy) is computed. The performance of a feature $\mathbb{F}_{ordered}^i$ is computed as the mean of the reward for the K-number of cross-validation:

$$\rho^i = \frac{1}{K}\sum_{k=1}^{K} \mathbb{P}\left(\mathbb{M}^i\left(\mathcal{Z}_{model-generat}^k\right), \mathcal{Z}_{validate}^k\right) \qquad (2)$$

<u>Algorithm: Patiently Greedy Search for Feature Selection</u>

Input: $\mathbb{F}_{ordered}, \mathbb{H}, \mathcal{Z}_{train}, \mathcal{Y}, \mathbb{P}, \Lambda$ (patience window)

Method:

1. $\alpha = \emptyset, \beta = \emptyset, \gamma = 0$
2. FOR *i*=*1* to Π
    a. $\beta = \beta \cup \{i\}$
    b. $\alpha = \alpha \cup \{\rho^\beta\}$, where ρ^β corresponds to feature $\mathbb{F}_{ordered}^\beta$ and ρ^β is computed as per Equation (2)
    c. $\gamma = \gamma + 1$
        i. If $\gamma == \Lambda$:
            1. $\mathrm{ind_{select}} = \operatorname{argmax}_\beta \alpha$
            2. $\alpha = \rho^{\mathrm{ind_{select}}}$,
            3. $\gamma = 0, \beta = \mathrm{ind_{select}}$

Output: $\mathbb{F}_{Distinct} = \mathbb{F}_{ordered}^{\text{ind}_{\text{select}}}$, denoting the distinct feature set $\mathbb{F}_{Distinct}$ discovered from $\mathbb{F}_{ordered}$ of indices $\text{ind}_{\text{select}}$.

Example: For simplicity of explanation, let the total number of features in $\mathbb{F}_{super}$ be 8 ($\mathbb{F}_{super} = \{\mathbb{F}_1, \mathbb{F}_2, \dots, \mathbb{F}_8\}$, classifier $\mathbb{H}$ ((say, Random Forest), patience window $\Lambda = 3$. The ordered feature set is $\mathbb{F}_{super} \xrightarrow{mRMR} \mathbb{F}_{ordered}$, $\mathbb{F}_{ordered} = \{\mathbb{F}_5, \mathbb{F}_8, \mathbb{F}_1, \mathbb{F}_6, \mathbb{F}_2, \mathbb{F}_3, \mathbb{F}_7, \mathbb{F}_4\}$. We compute $\rho_{\{\mathbb{F}_{ordered}^{\{1\}}, \mathbb{H}\}}, \rho_{\{\mathbb{F}_{ordered}^{\{1,2\}}, \mathbb{H}\}}, \rho_{\{\mathbb{F}_{ordered}^{\{1,2,3\}}, \mathbb{H}\}}$. Say, $\rho_{\{\mathbb{F}_{ordered}^{\{1,2,3\}}, \mathbb{H}\}}$ is maximum. Next, we compute the $\rho_{\{\mathbb{F}_{ordered}^{\{1,2,3,4\}}, \mathbb{H}\}}$, $\rho_{\{\mathbb{F}_{ordered}^{\{1,2,3,4,5\}}, \mathbb{H}\}}, \rho_{\{\mathbb{F}_{ordered}^{\{1,2,3,4,5,6\}}, \mathbb{H}\}}$ and it is found that $\mathbb{F}_{ordered}^{\{1,2,3,4,\}}$ maximizes the reward function. Next, $\rho_{\{\mathbb{F}_{ordered}^{\{1,2,3,4,7\}}, \mathbb{H}\}}$, $\rho_{\{\mathbb{F}_{ordered}^{\{1,2,3,4,7,8\}}, \mathbb{H}\}}$ are computed, we find $\rho_{\{\mathbb{F}_{ordered}^{\{1,2,3,4,7\}}, \mathbb{H}\}}$ is the maximum. Output $\mathbb{M}_{Suggested} = \{\mathbb{F}_{Distinct} = \{\mathbb{F}_1, \mathbb{F}_5, \mathbb{F}_6, \mathbb{F}_7, \mathbb{F}_8\}, \mathbb{H}\}$, where $\mathbb{F}_{ordered}^1 = \mathbb{F}_5, \mathbb{F}_{ordered}^2 = \mathbb{F}_8, \mathbb{F}_{ordered}^3 = \mathbb{F}_1, \mathbb{F}_{ordered}^4 = \mathbb{F}_6, \mathbb{F}_{ordered}^7 = \mathbb{F}_7$.

Observations: The proposed algorithm has some nice and elegant properties.

1. **Filter-based ranking order is respected:** Selected features in $\mathbb{F}_{Distinct}$ is in monotonically increasing order of feature ranks in $\mathbb{F}_{ordered}$.

2. **Nesting effect is minimized:** $\mathbb{F}_{ordered}^{\{1,2,3\}}$ would not be selected in greedy search, when $\rho_{\{\mathbb{F}_{ordered}^{\{1,2\}}, \mathbb{H}\}} < \rho_{\{\mathbb{F}_{ordered}^{\{1,2,3\}}, \mathbb{H}\}}$.

3. **Short-term memory long-term gain:** Patience window Λ acts as a short-term but volatile memory to allow the features to stay alive for a period equivalent of Λ. The fittest combination survives for the next Λ period.

4. **Graph-theoretic modeling:** We can further model our approach of the proposed metaheursitc feature subset selection algorithm from graph theoretic view. Initially, features of $\mathbb{F}_{super}$ form undirected graph (Figure 4A). We rank the features by filter-based feature ranking method (mRMR). Ranked features $\mathbb{F}_{ordered}$ form directed acyclic graph as depicted in Figure 4B, 4C (in compliance with the cited example) respectively (please note that: $\mathbb{F}_5 = \mathbb{F}_{ordered}^1$, $\mathbb{F}_8 = \mathbb{F}_{ordered}^2, \dots$). Real-valued edge weights $\rho^i: E \to \mathbb{R}$ are derived from the reward values associated with the valid features using the given classifier $\mathbb{H}$ following Equation (2). The greedy search based feature selection on ordered feature set works as depicted in Figure 4D. Let, ρ^1 be the reward value with $\mathbb{F}_{ordered}^1$. The edge weight $E_{(\mathbb{F}_{ordered}^1, \mathbb{F}_{ordered}^2)}$ be $\rho^{1,2}$, which is computed considering $\mathbb{F}_{ordered}^1, \mathbb{F}_{ordered}^2$ as the feature set at the model $\mathbb{M}^{1,2}$. If $\rho^{1,2} > \rho^1$, node 2 or $\mathbb{F}_{ordered}^2$ is included. Next edge weight is $\rho^{1,2,3}$ and it is found $\rho^{1,2,3} \ngtr \rho^{1,2}$. Consequently, node 3 or $\mathbb{F}_{ordered}^3$ is bypassed. Similarly, $\mathbb{F}_{ordered}^4$ and $\mathbb{F}_{ordered}^7$ are bypassed. Resultant directed acyclic graph is formed and shown in Figure 4E and the selected feature subset by greedy search is: $\mathbb{F}_{greedy-searc} = \{\mathbb{F}_{ordered}^1, \mathbb{F}_{ordered}^2, \mathbb{F}_{ordered}^5, \mathbb{F}_{ordered}^6, \mathbb{F}_{ordered}^8\}$. This notion is inspired from shortest path search.

Our proposed metaheuristic search groups the nodes to form individual trees (please note that now undirected graphs are formed) and number of forests are constructed, where number of trees at each of the forests is patience window Λ (except the last tree, which may contain less number of trees). Approximately $\frac{\Pi}{\Lambda}$ number of forests are formed. All the forests together form directed acyclic graph as depicted in Figure 4F. Each of the forests can be considered as a node in the greedy search and the edge weight of a forest is the maximum edge weight of its trees. Subsequent search is equivalent to greedy search (equivalent to the depiction in Figure 4D, 4E). The selected feature subset by proposed patiently greedy search is: $\mathbb{F}_{distinct} = \{\mathbb{F}_{ordered}^1, \mathbb{F}_{ordered}^2, \mathbb{F}_{ordered}^3, \mathbb{F}_{ordered}^4, \mathbb{F}_{ordered}^7\} = \{\mathbb{F}_1, \mathbb{F}_5, \mathbb{F}_6, \mathbb{F}_7, \mathbb{F}_8\}$.

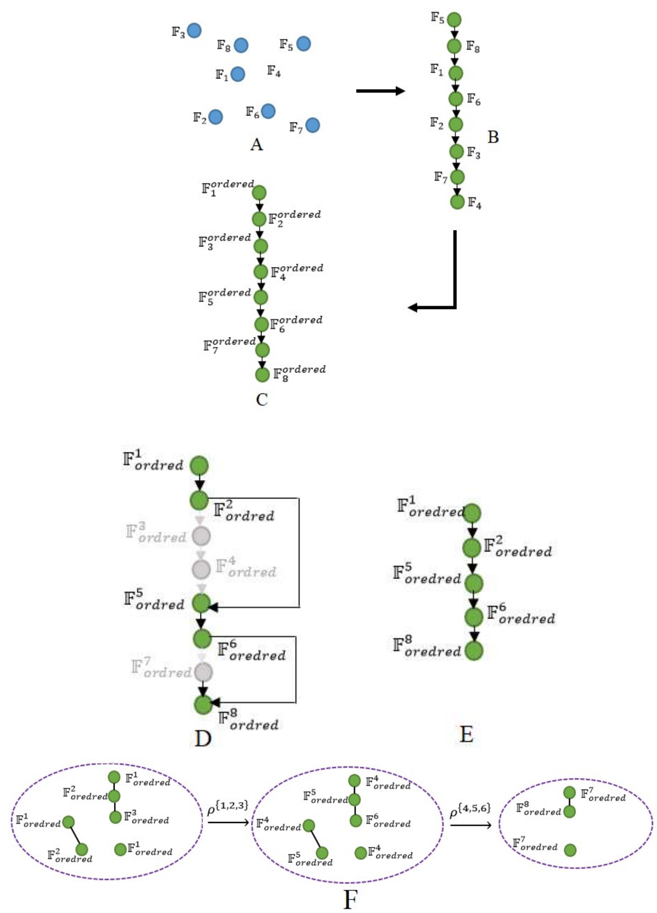

Figure 4. Graph Theoretic Representation, A. Feature Superset, B, C. Ordered Feature Set, D, E. Greedy Search, F. Proposed Metaheuristic Search-based Feature Subset Selection.

5 Experimental Evaluation

Our experimentation focuses on evaluating the performance efficacy of the proposed algorithm on stratified K-fold cross-validation (K =5), which provides the intuitive idea on the goodness of the learned model [Komiyama. et al. 2018]. For this experimentation study, total features are 392, $\Pi = 392$, consisting of micro and macro features [Ukil. et al. 2019]. Number of features selected π in $\mathbb{F}_{Distinct} \in \mathbb{R}^{\pi}$ varies from 4 (in TS_8) to 53 (in TS_4), much less than the number of features in $\mathbb{F}_{super} \in \mathbb{R}^{\Pi}$. We have considered Random Forest (any other classifiers like Support Vector Machine, AdaBoost, etc… can also be considered) as the classifier ($\mathbb{H}$) with number of trees = 100. We have used three different publicly available machine learning databases: UCR (UC Riverside) [Chen et al., 2015], UCI (UC Irvine) [Gorman. et al. 1988] and Physionet [Clifford. et al. 2017], [Goldberger. et al. 2000], [Liu. et al. 2016]. We have considered diverse sensor signals as described in Table 2, which are not only significantly different in characteristics, but also many of them are noisy (TS_5, TS_7), with smaller training examples (TS_1, TS_2, TS_3, Ts_6, TS_8, TS_9), with imbalanced class (TS_5). One of the key aspects of the proposed method is its ability to equally provide good performance over large as well as smaller number of training instances.

Our experimentation results as described in Table 3 reveal that AutoSensing consistently outperforms the closest and relevant state-of-the-art methods: 1. Widely recognized Auto-sklearn from AutoML [Feurer et al., 2015] as well as 2. Convolution neural network-based deep learning approach [Palaz et al., 2015]. We have performed stratified 5-fold cross-validation and demonstrated mean performance score with $\mathbb{P}$ = 'accuracy' and standard deviation over the folds. We have depicted sensitivity and specificity metrics as part of our experimental results to illustrate the complete confusion matrix.

In Deep learning based method where we consider the network as followed in [Palaz et al., 2015], both bias and variance figures are observed to be relatively higher, which is probably due to the practical constraints of unavailability of large training examples. The obtained empirical results in Table 3 demonstrate the trend of better model learning capability of AutoSensing over AutoML. AutoSensing provides considerable worthy and consistent performance over heterogeneous sensor signal types. In this work, we have considered ensemble flavor Auto-sklearn implementation of AutoML (https://automl.github.io/auto-sklearn/stable/) with $\mathbb{F}_{\text{super}}$ input. AutoML consists of 14 classifiers and 15 different feature recommendation methods along with relevant hyperparameters (tuning total 110 hyperparameters).We find that running time of AutoML implementation is much higher than AutoSensing. For example, on Intel Corporation Xeon Processor E7 with 128 GB RAM, SONAR dataset requires around 4 hours of run (training including feature subset selection) time by AutoSensing, whereas Auto-sklearn requires 29 hours of run time.

Data source	Class-1: Training size	Class-0: Training size	Dataset name	Signal type
TS_1	125	125	Computers	Electrical
TS_2	74	56	PPG	Optical
TS_3	104	35	Earthquakes	Vibration
TS_4	681	639	FordA	Engine noise
TS_5	2488	665	PCG	Audio
TS_6	40	20	Lightning_2	RF
TS_7	5154	3374	ECG	Electrical
TS_8	111	97	SONAR	Sound
TS_9	18	18	Toe_Seg_2	Motion

Table 2. Characteristics of the Time-series Sensor Signal Datasets in the Experimentation Purpose: Diversity of Sensor Types, Number of Instances, Class Distribution.

Dataset	Performance	AutoSensing	Auto-sklearn	Deep learning
TS_1	Accuracy	$\mathbf{0.88} \pm .05$	$0.80 \pm .02$	$0.61 \pm .06$
	Sensitivity	$\mathbf{0.92} \pm .06$	$0.80 \pm .04$	$0.26 \pm .11$
	Specificity	$0.82 \pm .00$	$078 \pm .04$	$\mathbf{0.99} \pm .07$
TS_2	Accuracy	$\mathbf{0.7}\pm.05$	$0.58 \pm .07$	$0.54 \pm .07$
	Sensitivity	$0.73\pm.04$	$0.57 \pm .14$	$\mathbf{0.83} \pm .10$
	Specificity	$\mathbf{0.74}\pm.10$	$0.63 \pm .15$	$0.40 \pm .13$
TS_3	Accuracy	$\mathbf{0.83} \pm .04$	$0.77 \pm .01$	$0.25 \pm .21$
	Sensitivity	$\mathbf{0.44} \pm .09$	$0.19 \pm .05$	$0.43 \pm .35$
	Specificity	$0.92 \pm .02$	$\mathbf{0.97} \pm .01$	$0.45 \pm .23$
TS_4	Accuracy	$\mathbf{0.96} \pm .01$	$0.95 \pm .01$	$0.86 \pm .03$
	Sensitivity	$\mathbf{0.96} \pm .01$	$0.94 \pm .01$	$0.72 \pm .07$
	Specificity	$0.95 \pm .01$	$0.94 \pm .02$	$\mathbf{0.99} \pm .01$
TS_5	Accuracy	$\mathbf{0.92}\pm.01$	$0.91 \pm .02$	$0.79 \pm .16$
	Sensitivity	$\mathbf{0.76}\pm.005$	$0.71 \pm .04$	$0.67 \pm .20$
	Specificity	$\mathbf{0.96}\pm.05$	$\mathbf{0.96} \pm .02$	$0.85 \pm .13$
TS_6	Accuracy	$\mathbf{0.92} \pm .10$	$0.79 \pm .11$	$0.74 \pm .08$
	Sensitivity	$\mathbf{0.93} \pm .18$	$0.86 \pm .13$	$0.60 \pm .22$
	Specificity	$\mathbf{0.80} \pm .13$	$0.70 \pm .1$	$0.54 \pm .45$
TS_7	Accuracy	$\mathbf{0.89} \pm .006$	$0.87 \pm .01$	$0.76 \pm .03$
	Sensitivity	$0.70\pm.01$	$\mathbf{0.83} \pm .02$	$0.50 \pm .06$
	Specificity	$0.85\pm.004$	$\mathbf{0.93} \pm .01$	$\mathbf{0.93} \pm .03$
TS_8	Accuracy	$\mathbf{0.88}\pm.04$	$0.83\pm.05$	$\mathbf{0.88} \pm .10$
	Sensitivity	$0.85\pm.07$	$0.82\pm.1$	$\mathbf{0.89} \pm .10$
	Specificity	$\mathbf{0.91}\pm.04$	$0.88\pm.04$	$0.80 \pm .17$
TS_9	Accuracy	$\mathbf{0.97} \pm .04$	$0.91 \pm .02$	$0.84 \pm .10$
	Sensitivity	$\mathbf{0.98} \pm .06$	$0.75 \pm .06$	$0.56 \pm .19$
	Specificity	$\mathbf{0.99} \pm .02$	$0.96 \pm .04$	$\mathbf{0.99} \pm .05$

Table 3. Autosensing in Comparison with Relevant State-of-the-art Methods: Auto-sklearn [Feurer et al., 2015] and Deep Learning [Palaz et al., 2015], Performance Figures in Terms of Mean $\pm$ Standard Deviation over Stratified K-Fold (K= 5).

Another relevant state-of-the-art algorithm HIVE-COTE [Bagnall et al., 2015], an ensemble learning method claims to exhibit better performance on time-series signals. Cross-validated mean accuracy of HIVE-COTE (results available for only UCR datasets at http://www.timeseriesclassification.com/Resamples.csv)- Computers: 0.82, Earthquake: 0.75, Ford_A: 0.96, Lighning_2: 0.80, Toe_seg_2: 0.97. We find that AutoSensing (Table 3) demonstrates considerable better performance over HIVE-COTE.

6 AutoSensing Case Study: Smart Healthcare

Number of portable sensors are available to record physiological signals, like ECG, EEG. Such sensors along with physical signals like accelerometer send the sensed signals to edge devices (e.g. smartphone). These sensor signals are fed to AutoSensing which analyzes the sensor signals for screening the health conditions. The architecture is illustrated in Figure 5. AutoSensing provides inference and subsequently sends alarms to the patient as well as to the concerned medical caregivers for awareness and action.

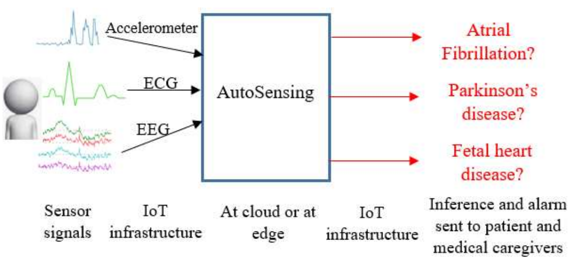

Figure 5. Smart Healthcare with AutoSensing.

7 Conclusion

We have proposed a complete automation of machine learning pipeline to perform classification tasks for time series sensor signals emphasizing automated feature engineering and model construction. Our proposed feature selection algorithm solves the nesting effect of traditional greedy search to a larger extent while not compromising on the computational cost, ensuring both effectivity and efficiency. We have described a graph-theoretic modeling of the proposed feature subset selection method. We put faith that extending the graph-theoretic approach will surely enable us to deduce more elegant solutions. In a complete machine learning pipeline, pre-processing and noise removal are vital steps, which will be taken up as future scope of work. AutoSensing is sensor-agnostic and the empirical evidences show that it empowers rapid development and prototyping of IoT analytics applications like smart healthcare, smart building, smart engine health monitoring, remote elderly care and many related others.

References

[Bagnall et al., 2015] Anthony Bagnall, Jason Lines, Jon Hills and Aaron Bostrom. Time-series classification with COTE: The collective of transformation-based ensembles. In: TKDE, 2015.

[Chen et al., 2015] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The UCR Time Series Classification Archive. 2015, www.cs.ucr.edu/~eamonn/time_series_data, http://www.cs.ucr.edu/~eamonn/time_series_data

[Clifford. et al. 2017] Gari D Clifford, et al. AF Classification from a Short Single Lead ECG Recording: The Physionet Computing In Cardiology Challenge 2017. In IEEE CinC, 2017.

[Feurer et al., 2015] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, Frank Hutter. Efficient and Robust Automated Machine Learning. In NIPS. 2015.

[Feurer et al., 2018] Feurer, M., Eggensperger, K., Falkner, S., Lindauer, M., & Hutter, F. (2018, July). Practical automated machine learning for the automl challenge 2018. In International Workshop on Automatic Machine Learning at ICML.

[Goldberger. et al. 2000] A.L. Goldberger, et al. Physiobank, Physiotoolkit, and Physionet: Components of A New Research Resource for Complex Physiologic Signals. In Circulation, 2000.

[Gorman. et al. 1988] R. Paul Gorman and Terrence J. Sejnowski. Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets. In Neural Networks, 1988. URL: http://archive.ics.uci.edu/ml/machine-learning-databases/undocumented/connectionist-bench/sonar/sonar.all-data

[Gray and Davisson, 2010] Robert M. Gray and Lee D. Davisson. An Introduction to Statistical Signal Processing. Cambridge University Press. 2010.

[Kohavi and John. 1997] Ron Kohavi, George H. John. Wrappers for Feature Subset Selection. In Elsevier Artificial Intelligence, 1997.

[Komiyama. et al. 2018] Junpei Komiyama, Hajime Shimao. Cross Validation Based Model Selection via Generalized Method of Moments. In North American Summer Meeting of the Econometric Society, 2018.

[Liu. et al. 2016] Chengyu Liu, et al. An Open Access Database for the Evaluation of Heart Sound Algorithms. In Physiological Measurement, 2016. https://www.physionet.org/challenge

[Palaz et al., 2015] Dimitri Palaz, Mathew Magimai.-Doss, Ronan Collobert. Convolutional Neural Networks-Based Continuous Speech Recognition Using Raw Speech Signal. In IEEE ICASSP, 2015.

[Peng et al., 2005] Hanchuan Peng, Fuhui Long, C. Ding. Feature Selection Based On Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. In IEEE TPAMI, 2005.

[Ukil. et al. 2014] Arijit Ukil, Soma Bandyopadhyay and Arpan Pal. Sensitivity inspector: Detecting privacy in smart energy applications. In IEEE Symposium on Computers and Communications (ISCC), pp. 1-6, 2014.

[Ukil. et al. 2019] Arijit Ukil, Pankaj Malhotra, Soma Bandyopadhyay, Tulika Bose, Ishan Sahu, Ayan Mukherjee, Lovekesh Vig, Arpan Pal, and Gautam Shroff. Fusing Features based on Signal Properties and TimeNet for Time Series Classification. In ESANN, 2019.