Combining Static and Dynamic Features for Multivariate Sequence Classification

A Paper Review

E. Bisong[1] Y. Deng[2]

[1]School of Computer Science
Carleton University

[2]Sprott School of Business
Carleton University

DATA 5000, 2018

Outline

Problem Statement

- Despite the fact that both static and dynamic features may contribute to the classification, they are rarely used together.
- One of the reasons is that most machine learning methods are not suitable for processing static and dynamic data simultaneously.
- This paper investigates whether there is a better way for extracting useful information from both data modalities to improve the overall classification performance.

Data Modalities

- unimodal: only *static* **or** *dynamic* features
- multimodal or bimodal: both *static* **and** *dynamic* features

static feature_1 **[age]**	static feature_2 **[gender]**	dynamic feature_id_1 **[heartbeats]**
45	M	1
32	F	2
67	M	3

id	timestamp	heartbeats
1	2005-10-2	40 bpm
2	2005-10-3	50bpm
2	2005-10-3	52bpm
2	2005-10-3	48bpm
3	2005-10-4	60bpm

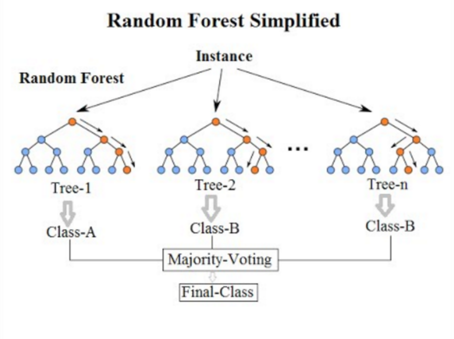
Figure: A dataset of static and dynamic features

Paper Contribution

- To devise a data augmentation technique where static features are concatenated with the data representation provided by a dynamic model.
- The paper refers to such an approach as hybrid and show that the hybrid way of stacking models is in general more beneficial than ensemble methods.

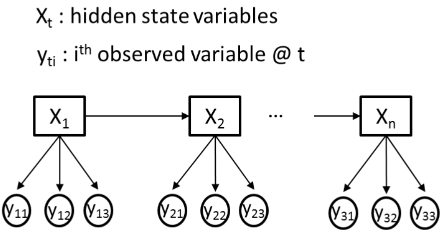
Random Forest

- Random Forest is an ensemble machine learning algorithm of Decision Trees.
- Random Forest is employed as a stand-alone discriminative classifier and as a final predictor, which combines the lower layer features in the ensemble and hybrid architectures.



Hidden Markov Models

- HMM is class of probabilistic graphical generative approaches. They are used to generate samples from a joint distribution of observed and unobserved features.
- The most commonly used is the 1-order Markov process with discrete hidden states: the probability of a given state depends only on the previous state, while ignoring the rest.



Long Short Term Memory

- LSTM does not make the Markovian assumption.
- In LSTM, the neuron is replaced by a more complicated structure, called an LSTM unit. This allows the LSTM network to learn long-term dependencies in the data making the LSTM a perfect fit for sequential data.

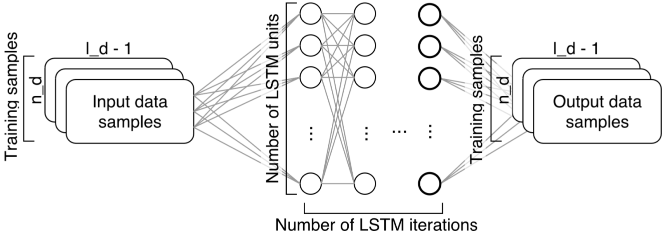


Figure: LSTM RNN structure

Methodology: Overview

- Three broad approaches are evaluated in this work including the main contribution, which is a hybrid model for multivariate sequence classification. The method groups are as follows:
  - Stand-alone Models on Unimodal data
  - Stand-alone Models on Bimodal data
  - Multiple Models on Bimodal data

  Let's briefly go through them.

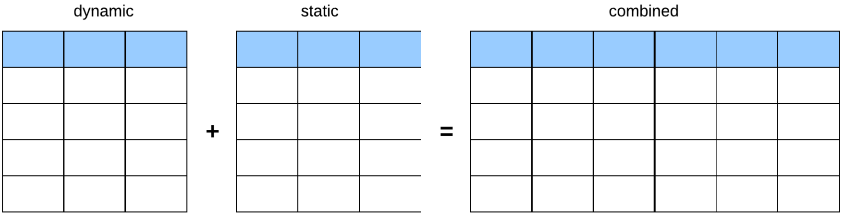
- **Random Forest on Static or Dynamic Features**: The most straightforward way to handle a multimodal dataset is to build a model on static features only. In this work such an approach is referred to as RF_s.

- **Hidden Markov Models on Dynamic Features**: The method denoted as HMM_d is a direct application of HMM to sequential data.
- **Long Short-Term Memory on Dynamic Features**: The method denoted as $LSTM_d$ is also a direct application of LSTM to sequential data.

- **Random Forest on Static and Dynamic Features**: The method under the name $RF_{s,d}$ transforms dynamic data to static, concatenates it with the original static features and employs Random Forest on the resulting feature set.
- **Hidden Markov Model on Static and Dynamic Features**: The method implemented in $HMM_{s,d}$ transforms static features into sequences.

Stand-alone Models on Bimodal data

Long Short-Term Memory on Static and Dynamic Features



- **Long Short-Term Memory on Static and Dynamic Features**: Here we also obtain the dynamic features from the static features, concatenate them with the original dynamic features and train an LSTM classifier on the combined feature set. This approach is referred to as $LSTM_{s,d}$.

- **Ensemble Models**: With an ensemble approach one can train different models for different data modalities (for example, Random Forest for static features and LSTM for dynamic features) and combine their predictions using a linear model or another layer of Random Forest (or any other discriminative method).

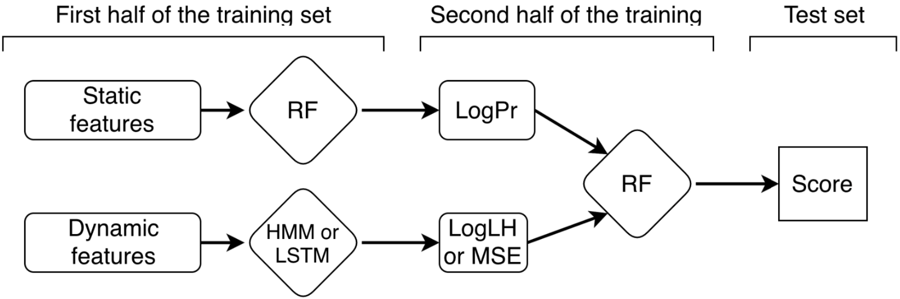


Figure: Architecture of an ensemble

- **Hybrid Models**: The general idea of the hybrid approach is to employ generative models such as HMM or LSTM to act as feature extractors from dynamic data.
- As generative models are able to generate sequences from the training data distribution, it is reasonable to assume that these models can capture temporal dynamics in the data.
- Therefore, the features extracted using these models can act as an approximation for temporal information contained in the data.
- These features are concatenated with the static features and a discriminative classifier (Random Forest) is used to build the final predictor.

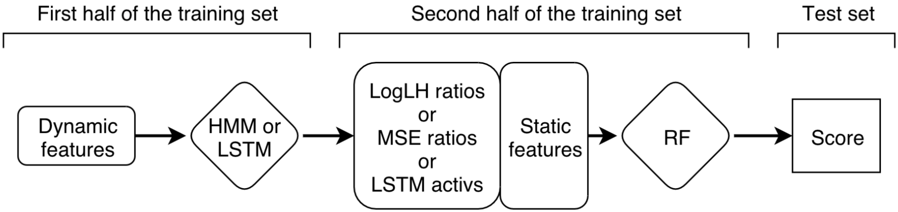


Figure: Architecture of the hybrid model. First half of the training set is used to create a model which will act as a feature extractor. Feature extractor is applied to enrich the feature set of the second half of the training set and the test set with additional features. Random Forest is trained on the second half of the training set to create a final classifier, which is evaluated on the test set.

Synthetic Datasets

- The authors compared the described approaches on several datasets from different domains as well as on simulated data.
- **Synthetic ARMA Dataset**: In order to be able to compare results with the ground truth and form the intuition how much information can be extracted from different types of features, the authors generated a synthetic dataset with specific properties.

Synthetic Datasets: An Overview

SYNTHETIC DATASET IS DESIGNED IN A SPECIFIC WAY. EACH BLOCK CONTAINS STATIC AND DYNAMIC DATA, HOWEVER THE DYNAMIC DATA IN BLOCK 1 AND STATIC DATA IN BLOCK 2 ARE USELESS, MAKING IT IMPOSSIBLE FOR A MODEL THAT OPERATES ONLY ON ONE DATA MODALITY TO CLASSIFY THE WHOLE DATASET CORRECTLY.

Block	Classifiability	Label	Static	Dynamic
1	Classifiable by discriminative model as T, but generative model will confuse it for F	T	$\sim \mathcal{N}_{\mathtt{T}}$	$\sim \mathrm{ARMA}_{\mathtt{F}}$
2	Classifiable by generative model as T, but discriminative model will confuse it for F	T	$\sim \mathcal{N}_{\mathtt{F}}$	$\sim \mathrm{ARMA}_{\mathtt{T}}$
3, 4	Classifiable by both as F	F	$\sim \mathcal{N}_{\mathtt{F}}$	$\sim \mathrm{ARMA}_{\mathtt{F}}$

Figure: Overview of the synthetic dataset

Real-life Datasets

- **Real-life Dataset**: The authors use datasets with different aspects: few univariate time-series widely used in the literature FordA and FordB and a multivariate dataset from a particular domain classification of electrocorticography (ECoG) recordings.

Real-life Datasets: An Overview

DESCRIPTIONS OF THE REAL-LIFE DATASETS.

Datasets	Samples	Train set	Test set	Static features	Dynamic features	Sequence length
ECoG	10584	5-fold CV		320	64	300
FordA	4291	1320	3601	500	1	500
FordB	4446	810	3636	500	1	500
Phalanges	2658	1800	858	80	1	80
Yoga	3300	300	3000	426	1	426

Figure: Overview of the Real-life datasets

Hyper-parameter Optimization

ESTIMATED HYPERPARAMETERS. COLUMN NAMES STAND FOR: LSTM SIZE, DROPOUT, OPTIMIZATION METHOD, BATCH SIZE, NUMBER OF EPOCHS; NUMBER OF HMM STATES, NUMBER OF ITERATIONS; NUMBER OF RF TREES.

	LSTM					HMM		RF
Dataset	S	D	O	B	E	S	I	T
ECoG	2000	0.5	rmsprop	32	50	6	50	500
FordA	512	0.0	rmsprop	1	20	2	50	500
FordB	512	0.0	rmsprop	1	20	2	50	500
Phalanges	128	0.0	rmsprop	1	10	2	50	500
Yoga	256	0.0	rmsprop	1	10	2	50	500

Figure: Hyperparameters

Performance on the synthetic dataset

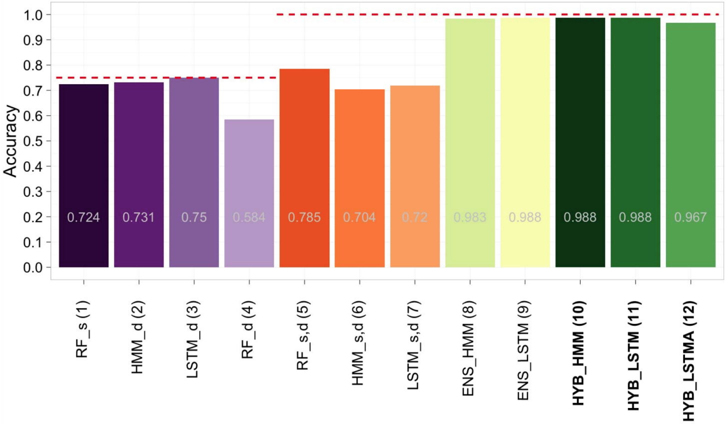


Figure: Performance on the synthetic dataset. The vertical dashed lines show the maximal achievable level of performance.

Performance on the real-life datasets

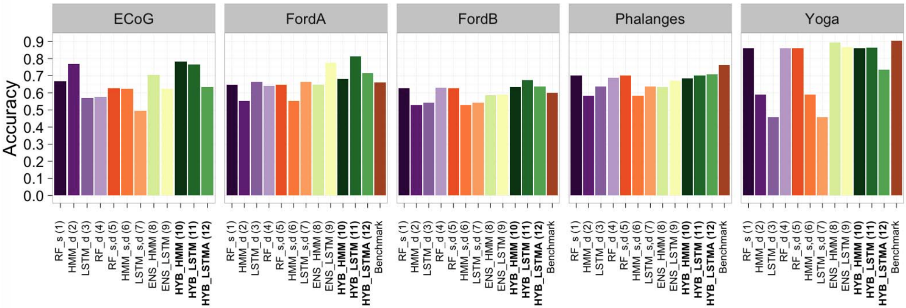


Figure: Performance on the real-life datasets.

Summary

- This is a well researched work.
- RF and LSTM are current state-of-the-art methods for predictive tasks using static and dynamic features respectively.
- The No Free Hunch Theorem shows us that an independent dataset with its own tweak of the hyperparameter space can thrown up interesting result.
- However, the take away from this work is the idea of extracting information from a dynamic feature set by using a generative model and then combining the newly extracted feature space with the existing static feature set for classification.

The End

Thank You for listening.