To Supervise or not - ML for UWB Close Range Obstacle Detection

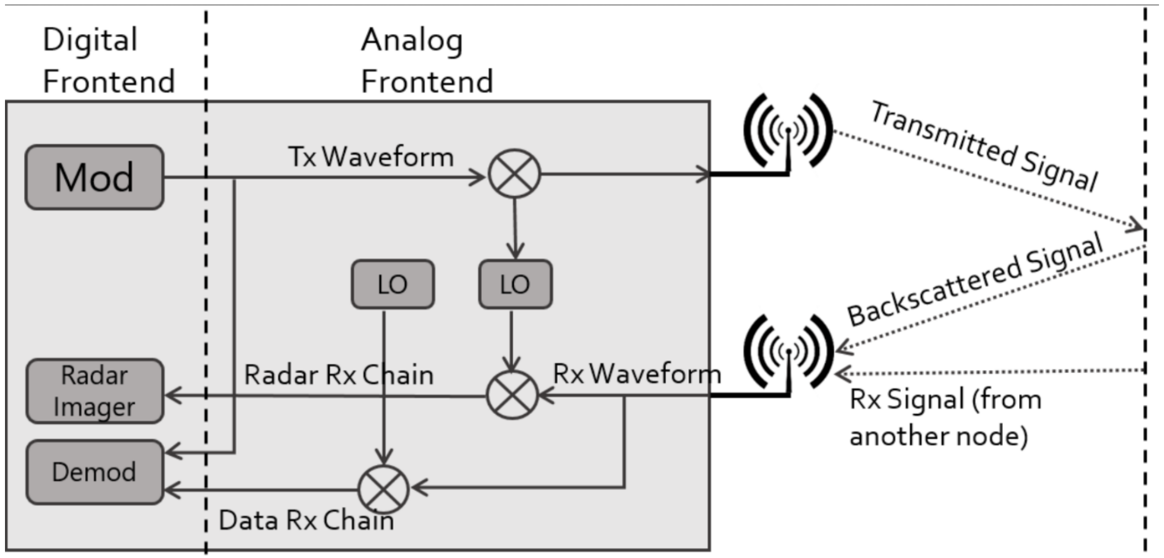
Raja Sattiraju, Jacob Kochems and Hans D. Schotten

Chair for Wireless Communications and Navigation

Technische Universität Kaiserslautern

Contents

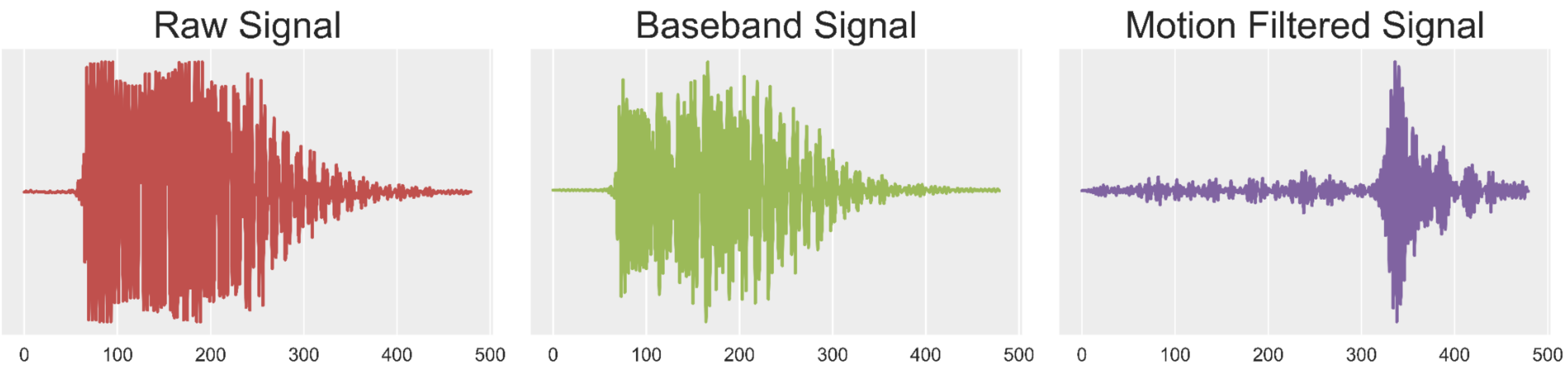
Introduction & Motivation



- Joint Communication & Obstacle Detection (JCOD) has applications in
  - Automatic Train Pairing (ATP)
  - Automatic Guided Vehicles (AGVs)

- UWB-like waveforms at 60GHz band offer the needed granularity for realizing JCOD

Monostatic Radar / UWB Pulses



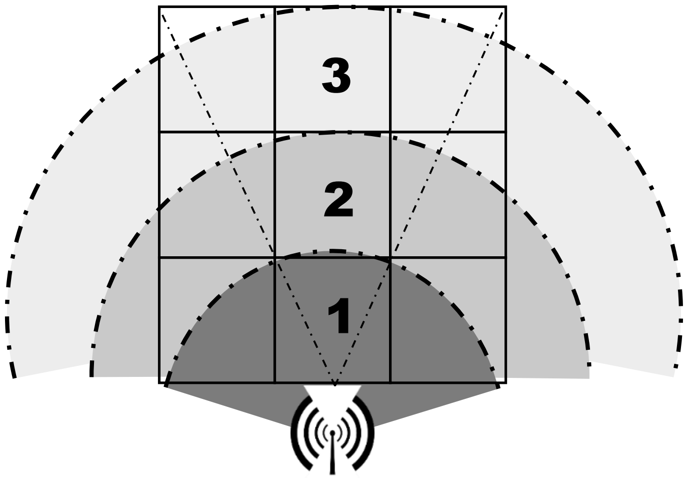
Scans collected using TimeDomain P440

Use the Time domain resolution of UWB Signals to construct a ML hypothesis

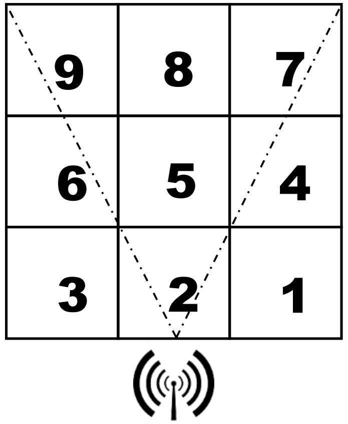
Construction of Hypothesis

Binary Hypothesis is not sufficient for the considered applications.

Hence Binary hypothesis is extended to multi-class labeling problem.



$$H = \begin{cases} 0, & \text{If there is no person} \\ 1, & \text{High Risk person} \\ 2, & \text{Medium Risk person} \\ 3, & \text{Low Risk person} \end{cases}$$



$$H = \begin{cases} 0, & \text{If there is no person} \\ 1, & \text{Person in Grid 1} \\ .. & .... \\ 9, & \text{Person in Grid 9} \end{cases}$$

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Input Data Collection and Preparation

- 2 Scenarios
  - Indoor
  - Outdoor

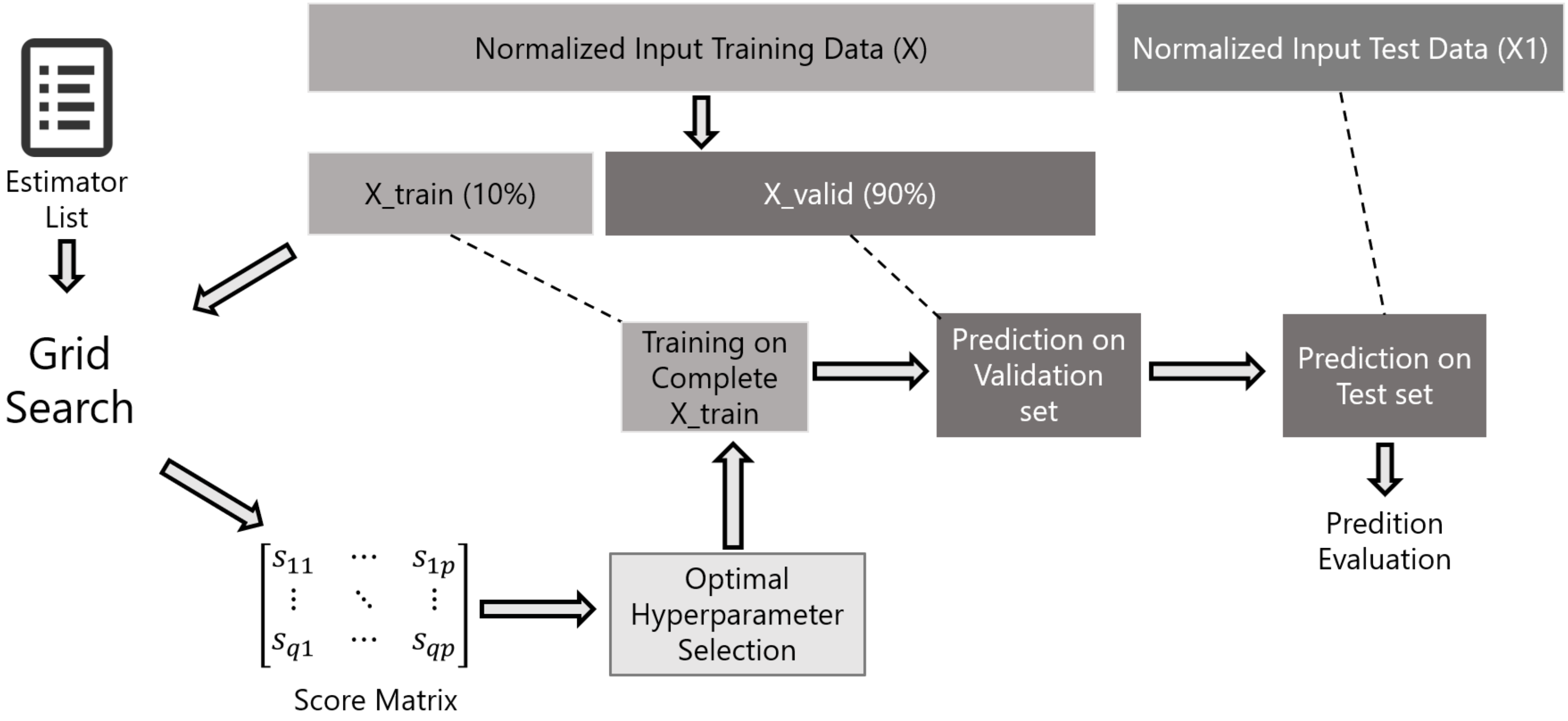
The Collected scans are of the size (N_s, N_f) where N_s is the number of samples and N_f is the number of features.

- Data Preprocessing
  - Zero centering of data and normalizing it with standard deviation (Feature Wise)

$$X_n = \sum_{i=1}^{N} \left(\frac{X_i - \frac{1}{N}\sum_{i=1}^{N} X_i}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)^2}} \right)$$

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Training Methodology



Estimators Used

Mixture Models – Gaussian Mixture Model (GMM) and Bayesian Gaussian Mixture Model (BGMM)

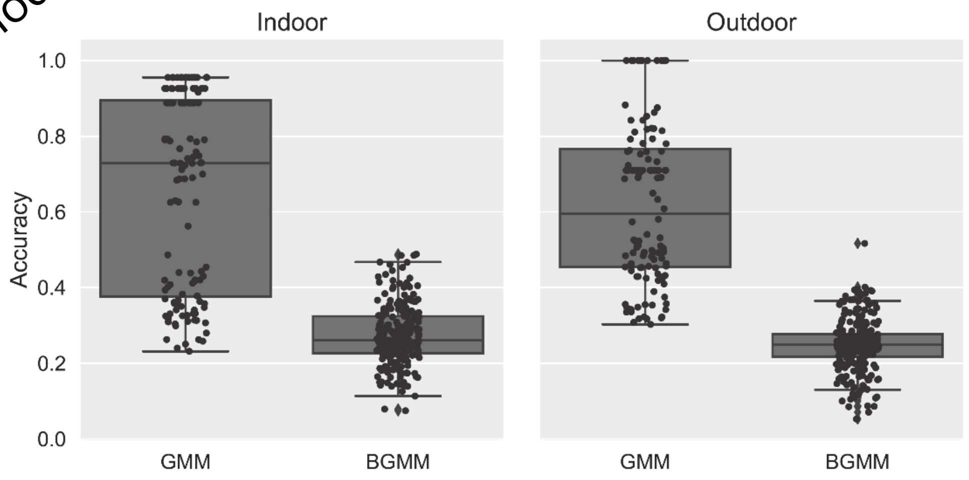
Clustering Models – KMeans, Mini-batch KMeans, Agglomerative Clustering and Birch Model

Manifold Learning – Locally Linear Embedding (LLE), Isometric Mapping, Multi-Dimensional Scaling (MDS), Spectral Embedding (SE) and t-Distributed Stochastic neighbour Embedding (tSNE)

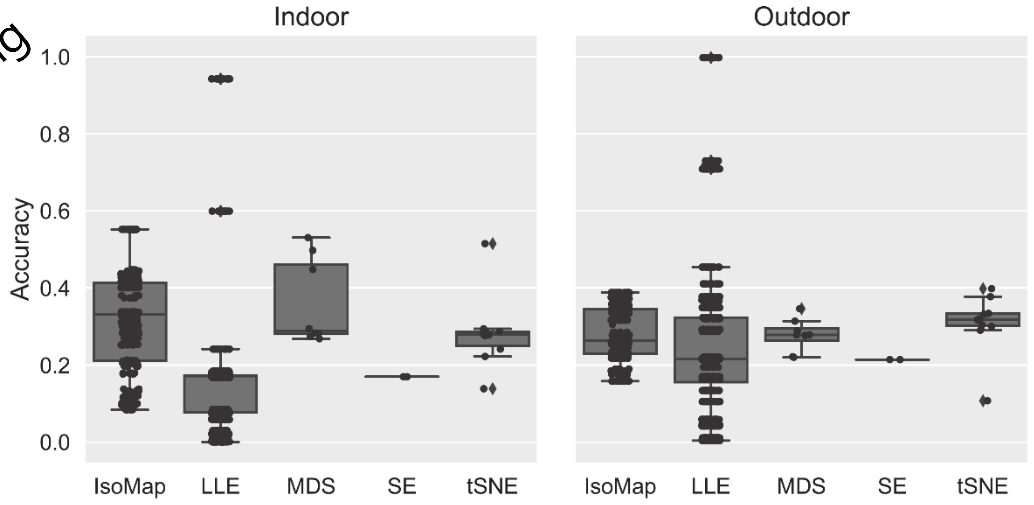
Name	Underlying Model	Parameters	Values
Gaussian Mixture Model Bayesian Gaussian Mixture Model	Mixture Models	Regularization covariance	[1e-6, 1e-5]
		tolerance	[1e-4, 1e-5]
		Number initializations	[1,2]
		Maximum iterations	[100,200]
		Covariance type	[diagonal, spherical, tied, full]
KMeans	Clustering	initialization	[random, k-means++]
		Number initializations	[10,15,20]
		tolerance	[1e-4, 5e-4, 1e-3, 5e-3]
Mini-batch KMeans	Clustering	initialization	[random, k-means++]
		Batch Size	[100,200,300,400,500]
		Reassignment ratio	[0.01, 0.05, 0.1, 0.5]
Agglomerative Clustering	Clustering	Affinity	[euclidean, l1, l2, manhattan, cosine]
		linkage	[ward, complete, average]
Birch	Clustering	Threshold	[0.2, 0.5, 0.7, 0.9]
		Branching Factor	[25,50,100,200]
		Compute Labels	[True, False]
Locally Linear Embedding	Manifold Learning	Number of neighbors	[1,2,3,4,5,6,7,8,9,10]
		Method	[standard, ltsa, hessian, modified]
		Neighbors algorithm	[auto, brute, kd_tree, ball_tree]
Isometric Mapping	Manifold Learning	Number of neighbors	[1,2,3,4,5,6,7,8,9,10]
		Eigen solver	[auto, arpack, dense]
		Path method	[auto, FW, D]
		Neighbors algorithm	[auto, brute, kd_tree, ball_tree]
Multi Dimensional Scaling	Manifold Learning	Metric	[True, False]
		EPS	[0.001, 0.005, 0.01, 0.05]
Spectral Embedding	Manifold Learning	Affinity	['nearest_neighbors', 'rbf'
t-Distributed Stochastic Neighbor Embedding	Manifold Learning	Learning rate	[10.0, 50.0, 100.0, 250.0, 500.0]
		Initialization	[Random, PCA]
		Method	[barnes_hut, exact]

Results

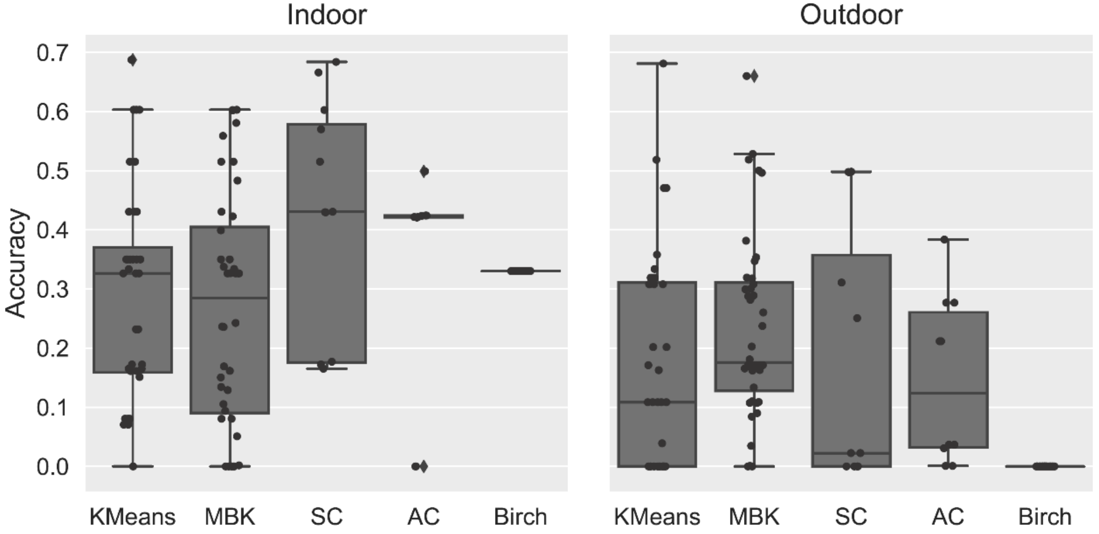
Mixture Models



Manifold Learning



Clustering



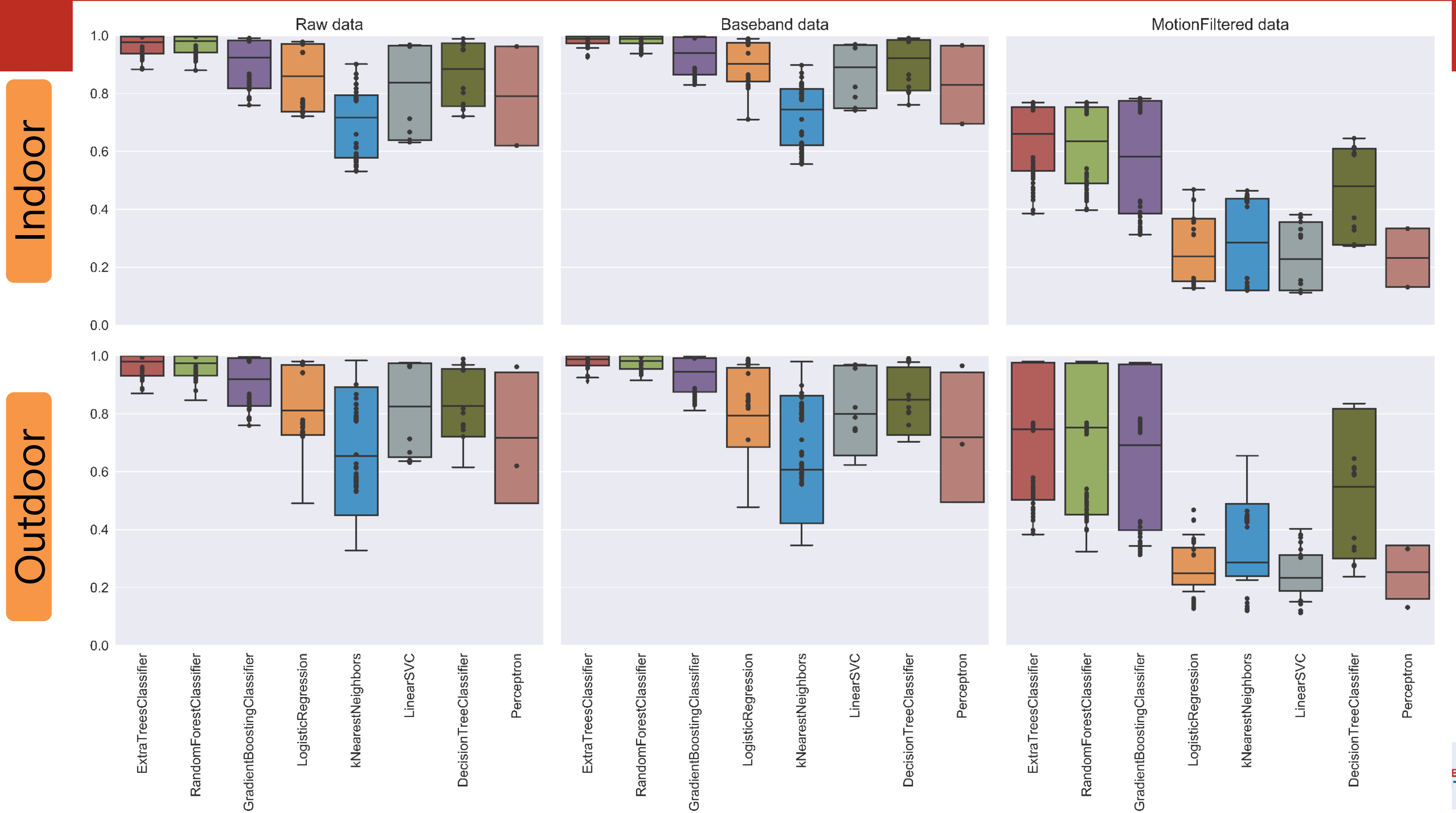
Estimator	Accuracy Indoor	Accuracy Outdoor
GMM	65%	42%
BGMM	52%	51%
KMeans	54%	33%
MBK	53%	34%
SC	97%	64%
AC	18%	18%
Birch	29%	10%
IsoMap	78%	80%
LLE	68%	88%
MDS	44%	32%
SE	52%	26%
t-SNE	15%	12%

Test Scores

Comparison with Supervised Learning Algorithms

Name	Underlying Model	Parameters	Values
Logistic Regression	Linear	Regularization parameter (C)	[0.001, 0.01, 0.1, 1, 10, 100, 1000]
		Solver	[lbfgs, sag, newton-cg]
Perceptron	Linear	Regularization parameter (Alpha)	[0.0001, 0.001, 0.01, 0.1, 1]
K-Nearest neighbors	Nearest neighbors	Number of neighbors to consider (N)	[1, 2 ,....., 30]
Linear SVC	Support Vector Machine	Regularization parameter (C)	[0.001, 0.01, 0.1, 1, 10, 100, 1000]
Decision Tree	Tree Based	Splitting Quality Measure	[gini, entropy]
		Max_features to consider for splitting	[auto, sqrt, log2]
Random Forest Classifier Extra Trees Classifier	Tree based ensemble	Number of estimators, n	[16, 32, 64, 128, 25 6]
		Splitting Quality Measure	[gini, entropy]
		Max_features to consider for splitting	[auto, sqrt, log2]
Gradient Boosting Classifier	Tree based boosting	Number of estimators, n	[16, 32, 64, 128, 256]
		Learning Rate	[0.2, 0.5, 0.8, 1.0]

Grid Scores – Supervised Learning



Conclusions

- Supervised Learning Methods outperform the unsupervised methods
- However, they needed predetermined number of labels and labeled data
- A hybrid model using a combination of both supervised and unsupervised models
  - First check the data distinguishability by means of unsupervised learning models (Mixture / Clustering models)
  - Second, fit the data using any of the supervised learning models and obtain the best model
  - Third, use manifold learning to reduce the dimensions. The number of output dimensions can be chosen so as not to affect the models score
  - Re-fit the best supervised learning model with the reduced data to obtain the final model

Thank You

Raja Sattiraju, M.Sc.
Chair for Wireless Communication & Navigation
Technical University of Kaiserslautern

Ph: +49 6312053680
Email: sattiraju@eit.uni-kl.de

TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN