Load Detection Techniques using High Frequency Features on Embedded Systems

Master Thesis Presentation
Hugo Wöhrl

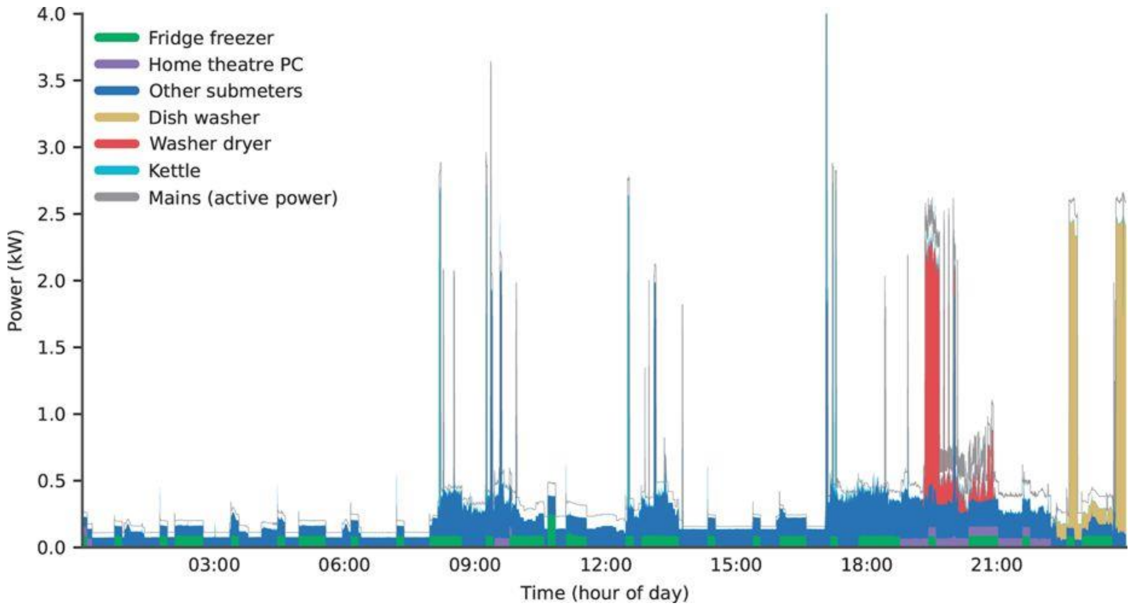
University of Bologna
Department of Electrical, Electronic and Information Engineering
Supervisor: Prof. Luca Benini, Dr. Davide Brunelli

Table Of Contents

Introduction to NILM

- NILM is the task of extracting energy consumption from an agglomerated power signal
- NILM founded in the mid 80s by G.W. Hart at MIT
- Smart Meter Roll out in several countries increased interest massively
- Most research focuses on Smart Meter generated Data (1 Hz Power Measurements)
- In older publications mostly statistical methods as Hidden Markov Models
- Kelly published in 2015 "*Neural NILM*", which was a kickoff for using Neural Networks in the Field of NILM

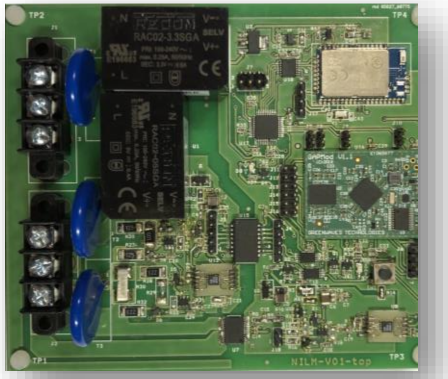
- Introduction of Smart Meters experienced a Backlash due to Privacy Concerns in the UK



Power Plot over one day from the open UK-Dale Dataset
[The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes]

The Smart Measurement Node

Development Kit for Non-Intrusive Load Monitoring Applications

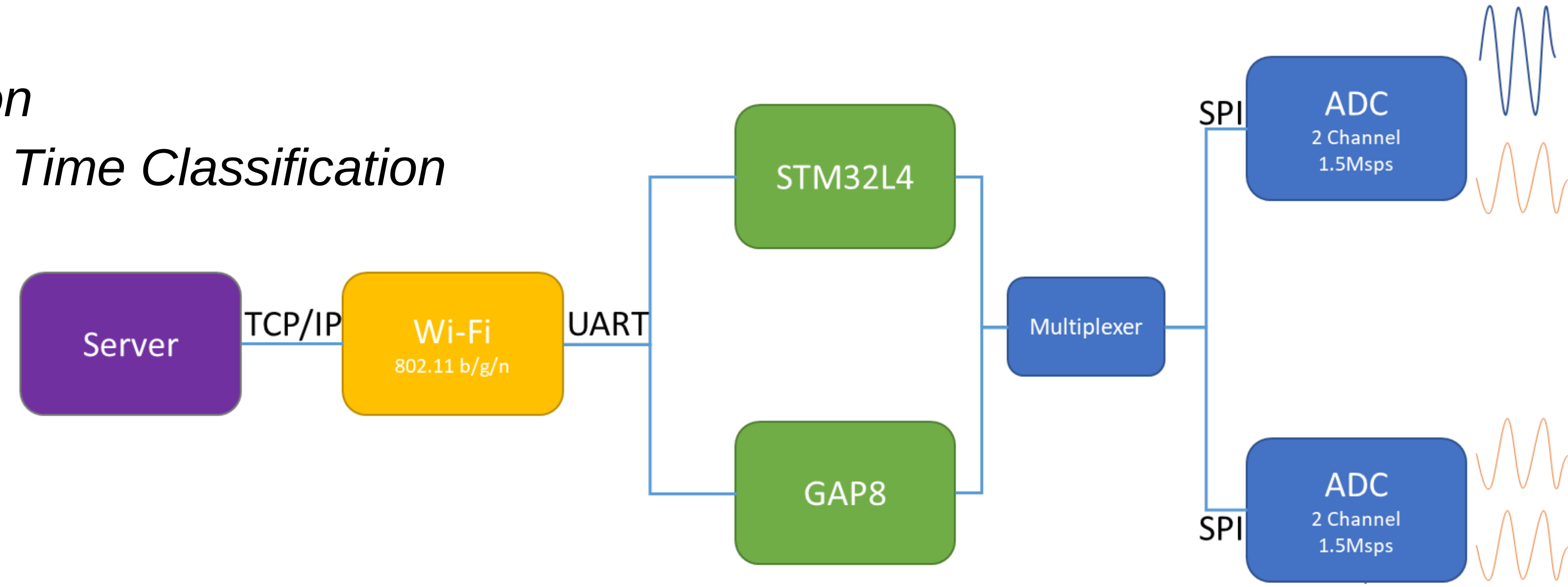


- Two Analog Interfaces
  - Simultaneous Sampling @ 1.5MSps
  - 14Bit Resolution
  - Invasive and Non-Invasive Measurements possible

- Two MCUs
  - STM32F410x8 → *Data Acquisition*
  - GreenWaves GAP8 → *Offline Real Time Classification*

- Wi-Fi Module
  - Tested Bandwidth: 100kB/s

  → Data Transmission Rate of 25kHz for 2 Channels

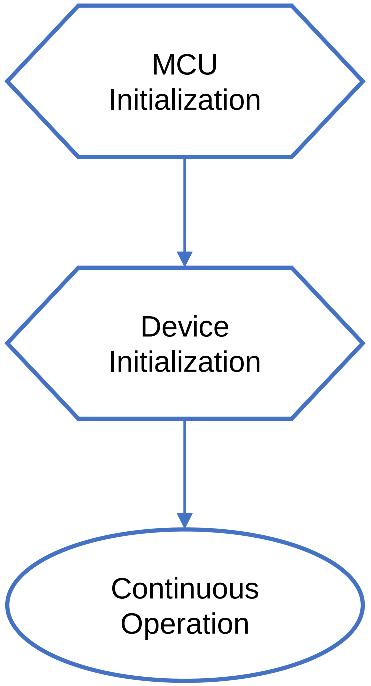


Firmware

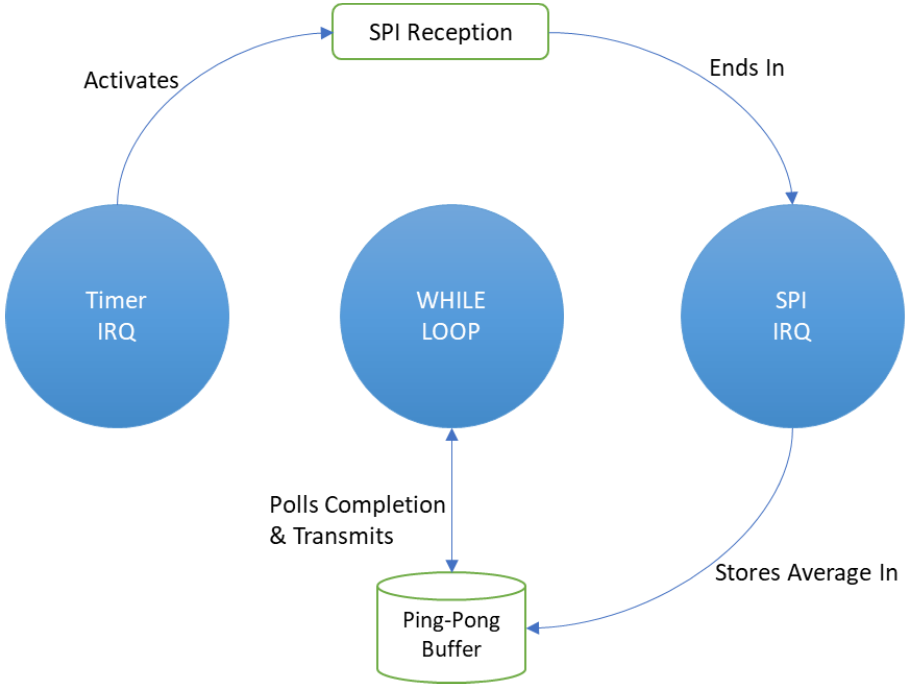
Configure:
- Clocks, GPIOs, Timers
- DMAs, Interrupts

- Initialize Wi-Fi Module
- Establish Connection
- Instantiate Measurement

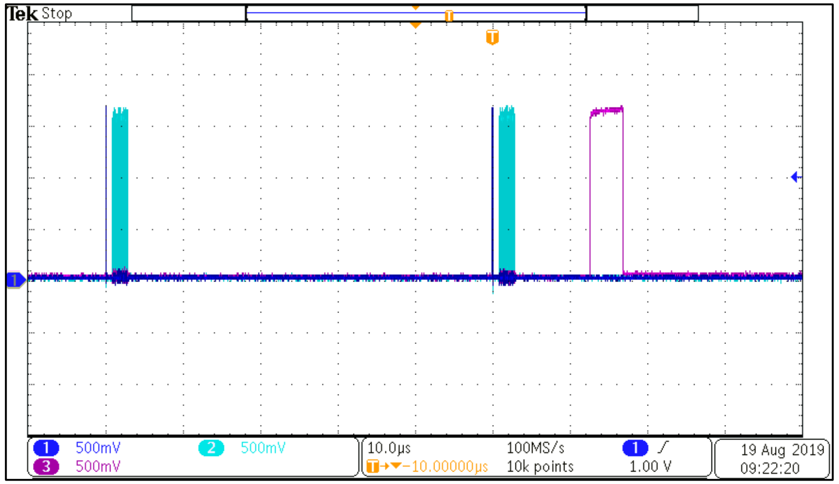
- Perform Measurement
- Form Average
- Fill Ping-Pong Buffer
- Send Full Ping-Pong Buffer



Flow Chart of Firmware



Processes of Continuous Operation

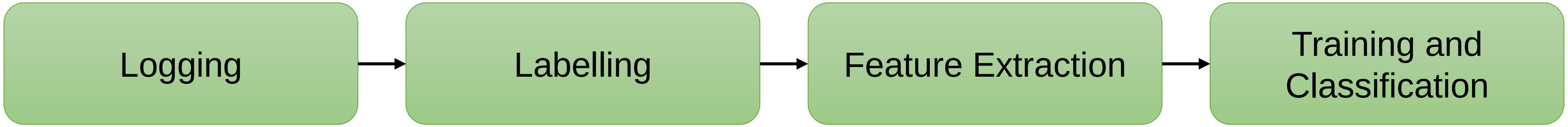


Oscilloscope Plot of Conversion Signals: (1) = Timer1; (2) SPI Clock; 3 Storage Complete

Source Code online available:
https://github.com/fabianhugo/nilm-stm32f4-daq

Software Tools

Tool Pipeline:



- Data Set Recorded Via TCP/IP Socket
- Manually Labelled using Plots of Current Traces
- Features are extracted from Time Series and stored to .csv
- Classifiers: Support Vector Machine Classifier (SVC), k-Nearest Neighbor (kNN), Multi Layer Perceptron (MLP), Random Forest Trees (RFT)

All tools written in Python3, based on Scikit-Learn and Pandas

Data Acquisition

- **8 Appliances of 3 different types**
  - On/Off Appliances: Incandescent Light Bulb, Electric Kettle, Fan(3 States), Fluorescent Light =6
  - Finite-State-Machine Appliances: Microwave (3 States) =3
  - Continuously Variable Loads: Laptop, Monitor, Cellphone Charger =3

  → **12 Classes + 1 'No Load' Class**

- **5 Different Scenarios:**
  1. Single Appliances without Switching Events
  2. Single Appliances with Switching Events
  3. Multi Appliance Recording
     1. Without Superposition of Loads
     2. With Superposition of Loads
  4. Extended Single Appliance Detection
  5. Switching Event Collection

Dataset available online:
https://github.com/fabianhugo/nilmdata

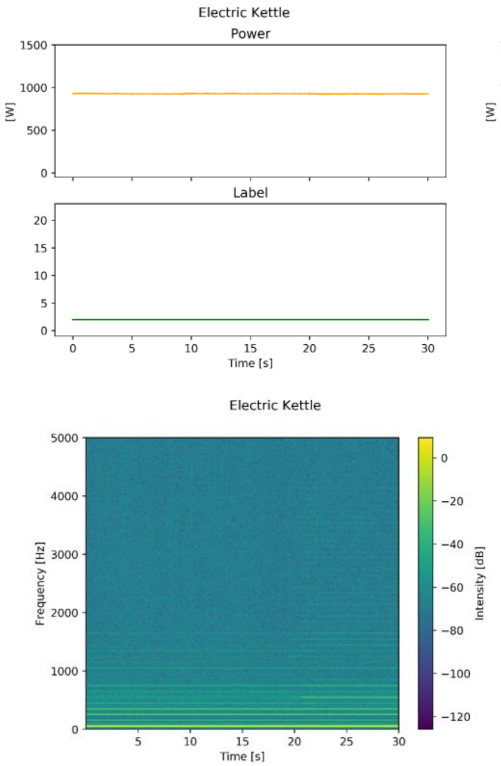
- **For Extended Single Appliance Detection 10 Laptop States were added → 23 Classes**

Recordings were repeated to create a robust dataset and to cover different appliance states and states of the grid

Data Acquisition: Recorded Scenarios

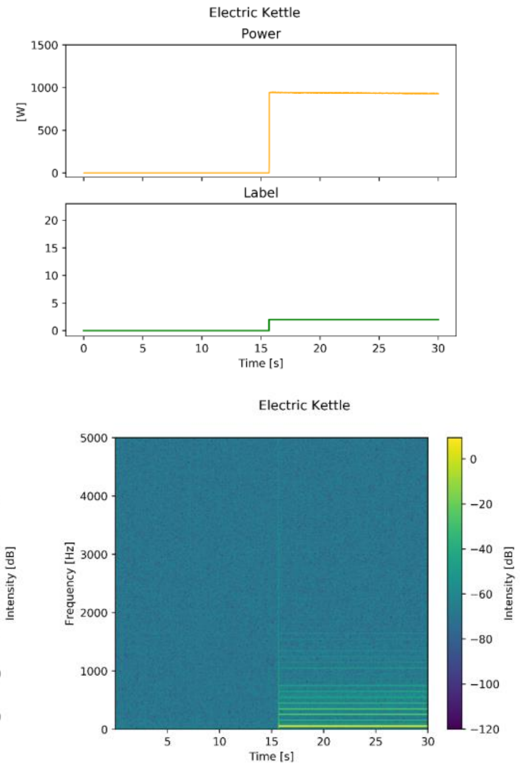
1 Single Appliances without Switching Events

30s



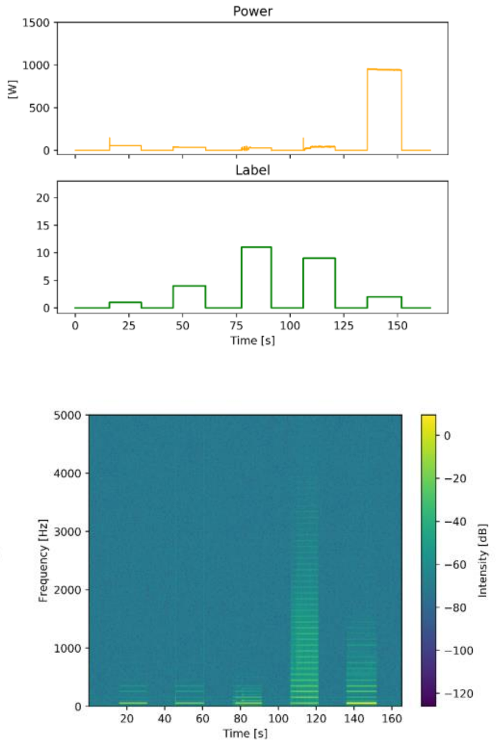
2 Single Appliances with Switching Events

30s



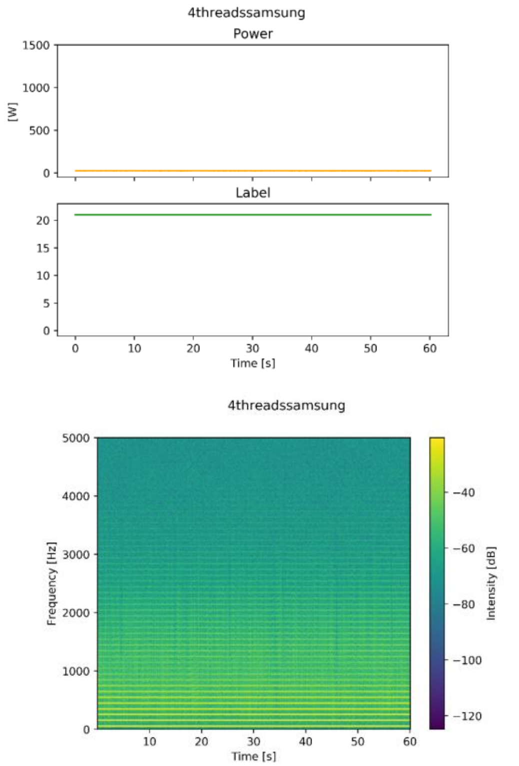
3 Multi Appliance Recording (with and without Superposition)

150s



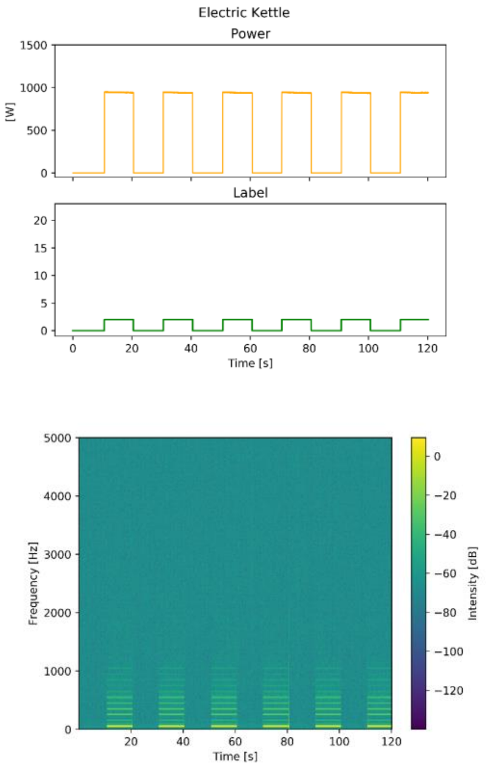
Extended Single Appliance Detection

60s



Switching Event Collection

30s



Features

- Measurement Sampling Rate: fs $= 10kHz$
- Feature Fusion Approach from Bernard:
  - Low Frequency Power Features
  - Odd Harmonics of Current $[50Hz, 150Hz, 250Hz..]$
- Features are extracted from windows of 100ms
  - Windows contain 1000 Samples
  - Grid Frequency: 50Hz → Averaging over 5 Periods
- Nyquist Frequency: $f_{nyquist} = \frac{fs}{2} = 5kH$

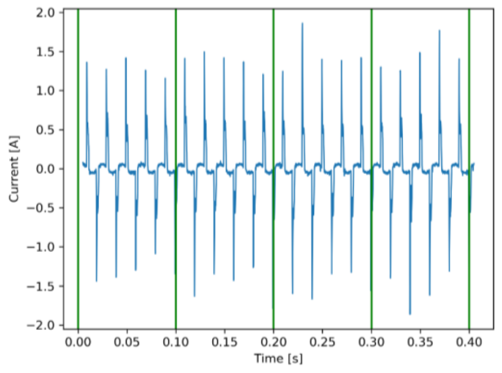
→ Odd Harmonics till 99th (4950Hz) can be extracted

1. $Apparent\ Power\ S = U_{RMS} \cdot I_{RMS} = \sqrt{\frac{1}{N}\sum_0^N u^2(n\tau)} \cdot \sqrt{\frac{1}{N}\sum_0^N u^2(n\tau)}$

2. $Active\ Power\ R = \frac{1}{T}\int_0^T u(t) \cdot i(t)dt = \frac{1}{N}\sum_0^N (u(n\tau) \cdot i(n\tau))$

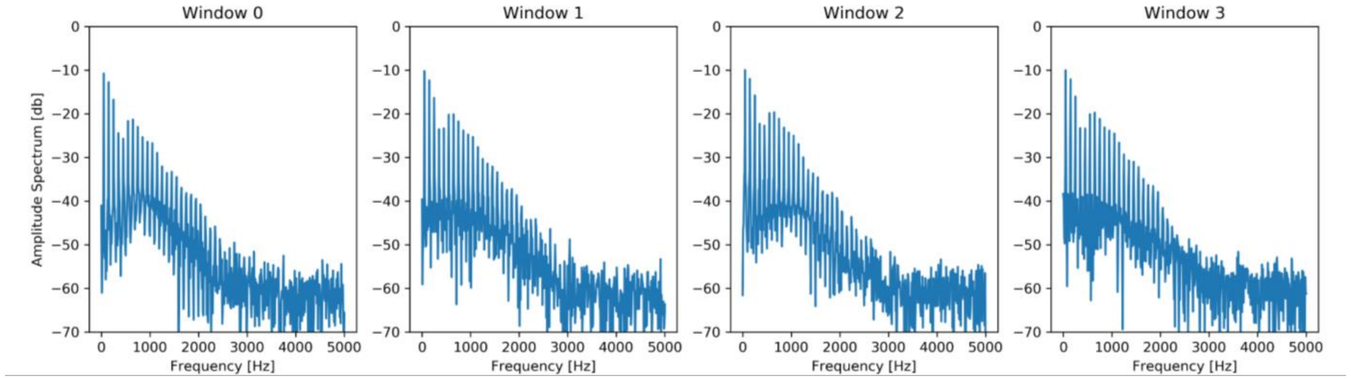
3. $Reactive\ Power\ Q = \sqrt{S^2 - P^2}$

$4.–103.\ Harmonics\ of\ Current: I_1..I_m with\ I_{1-m} = DFT\ (i(n\tau))$
Real and Imaginary Part of Complex Harmonics are stored seperately



Current Trace of Laptop without Switching Event

DFT

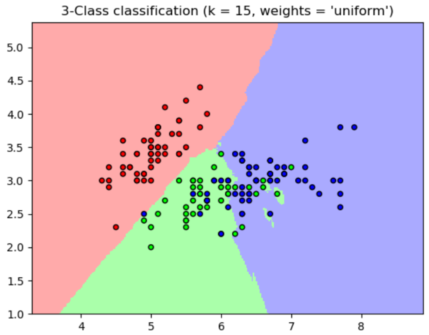


Transformed Current Spectrum of 100ms windows

Training and Classification Procedure

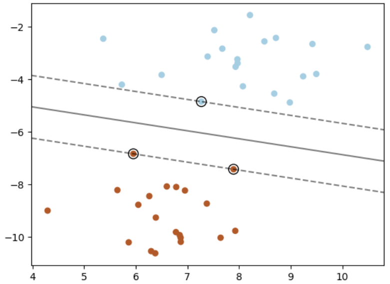
1. Load features and labels from dataset(s)
2. Perform grid search to find best algorithm and hyper parameters on training dataset
3. Split dataset into test- and training dataset
4. Use 10-fold Cross validation to affirm classification result
5. Train Model on best algorithm and best hyper parameters
6. Load unseen dataset and validate performance

k Nearest-Neighbor Classifier



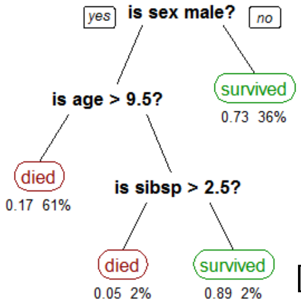
[scikit-learn,kNN]

Support Vector Machine Classifier



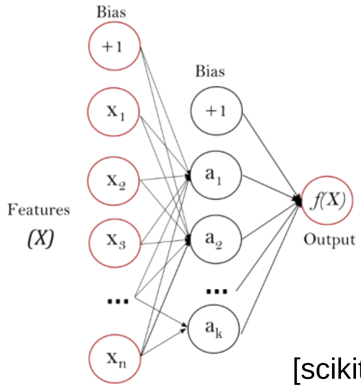
[scikit-learn, SVM]

Random Forest Tree Classifier



[wikipedia, Decision Tree]

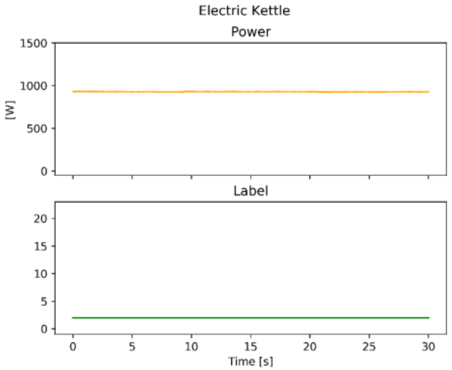
Multi-Layer Perceptron Classifier

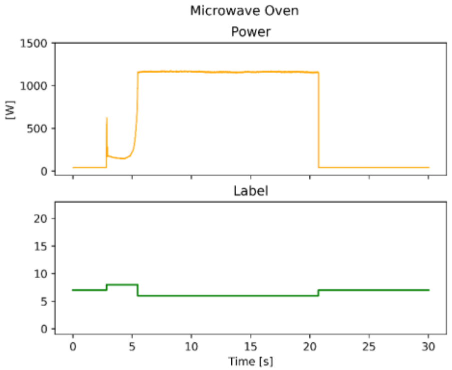


[scikit-learn, Neural Networks]

1 Eventless Classification

Training and Validation on Scenario 1 Dataset:
Single Appliances Without Switching Events

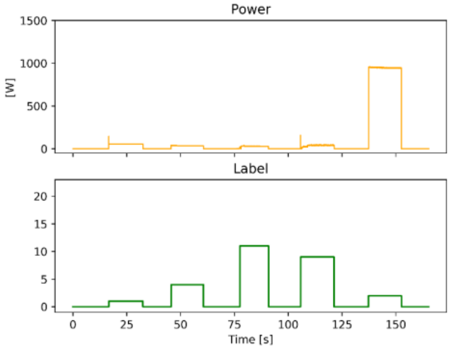




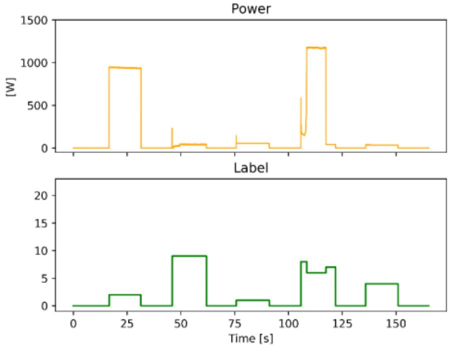
- 5 Repetitions of 30s-Recordings of 13 Classes
- Window size: 100ms

→ Dataset contains ca. 20 000 labelled Instances

Evaluation on Dataset 3:
Multi-Appliance Recordings Without Switching Events



Switching Order00:
Heatbulb, Fan2, Fluorescent Light, Laptop, Kettle



Switching Order02:
Kettle, Laptop, Heatbulb, Microwave, Fan2,

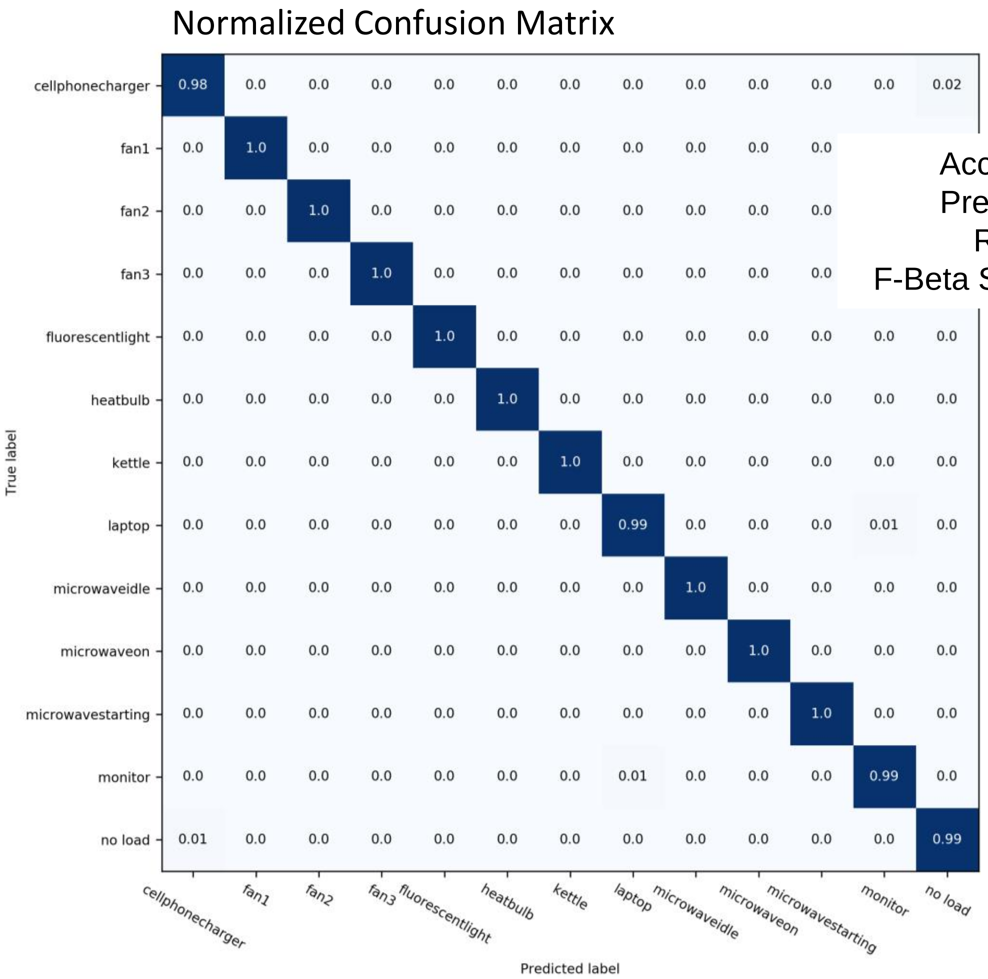
Dataset contains 150s-recordings of different combinations of appliances switching on and off consecutively.
Dataset is not used during training

Evaluation 1: Eventless Classification

Best Classification of13 Classes with SVC @ C=1, gamma = 0.0001 and a linear kernel.

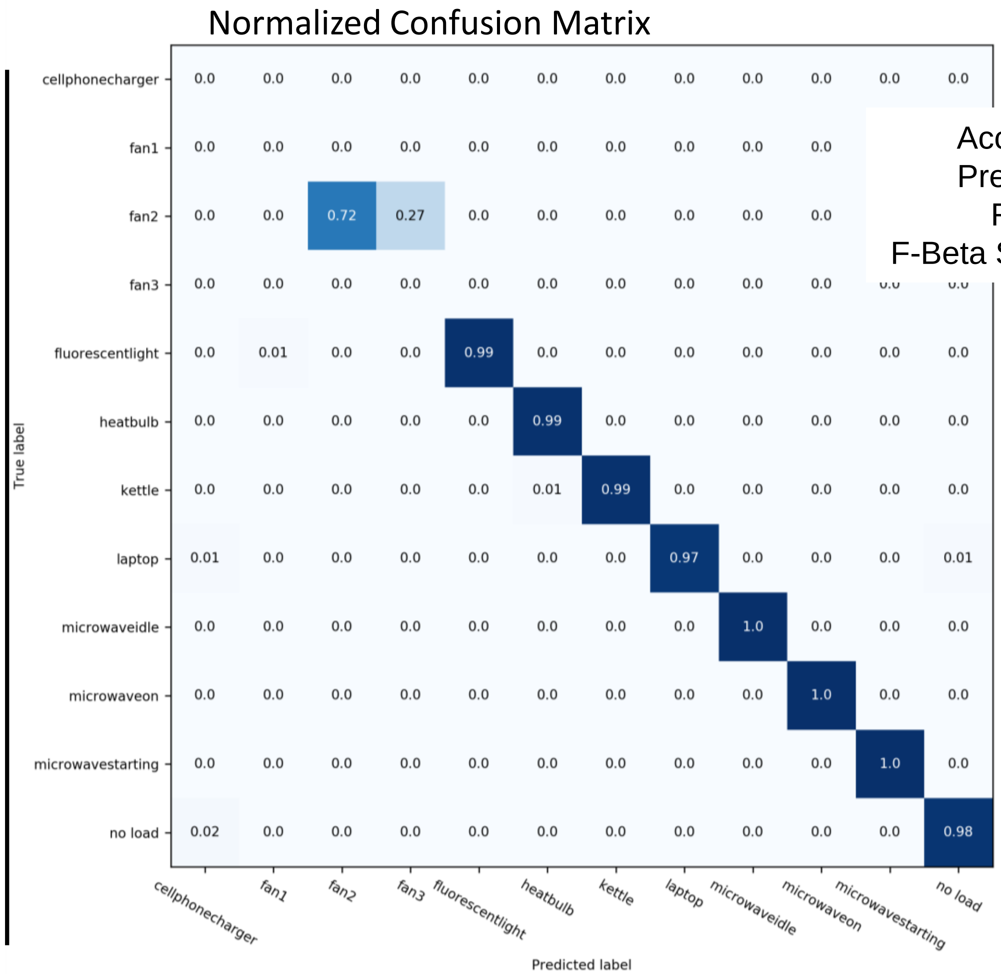
MLP-Classifier, Random Forest and k-NN perform only slightly worse

On Test Dataset of Model



Accuracy= 99.33%
Precision= 99.33%
Recall = 99.33%
F-Beta Score = 99.32%

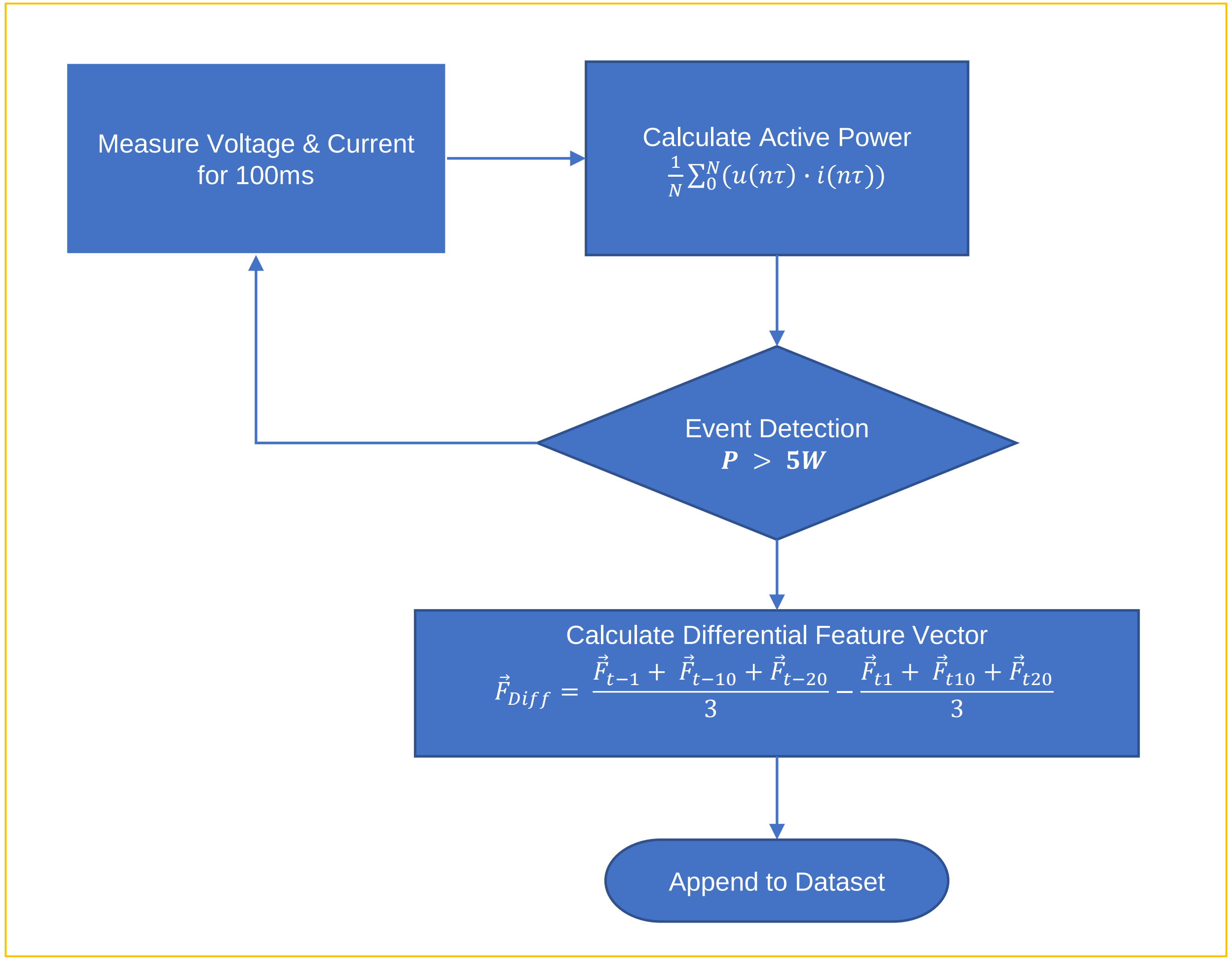
On Scenario 3.1



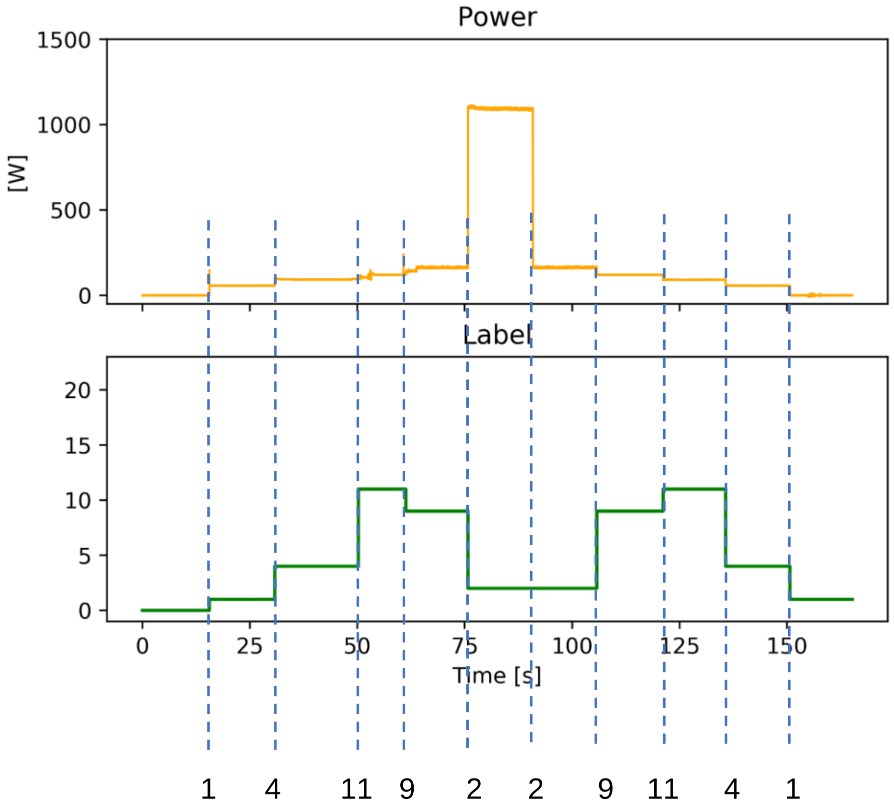
Accuracy= 95.83%
Precision= 99.55%
Recall = 95.83%
F-Beta Score = 97.47%

Event-based Classification

Event Detection and Feature Extraction Algorithm



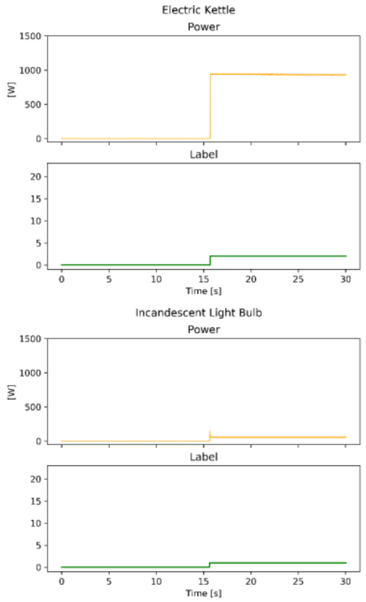
Dataset is labelled differently here
The labels here signify load state change.



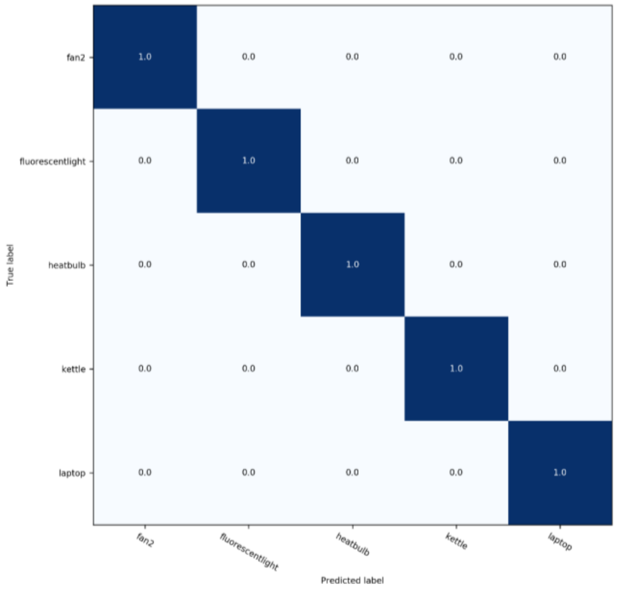
Evaluation 2: Eventless Classification

Best Classification with Random Forest Tree @ Max Leaf Nodes: 50, Min Samples Leaf: 1, ‘N Estimators': 1000 with 5 Classes

On Test Dataset of Model

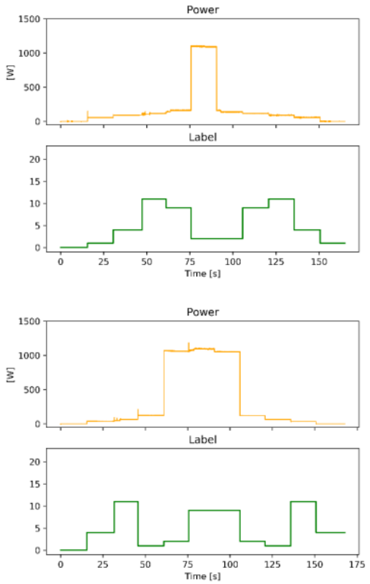


Confusion Matrix

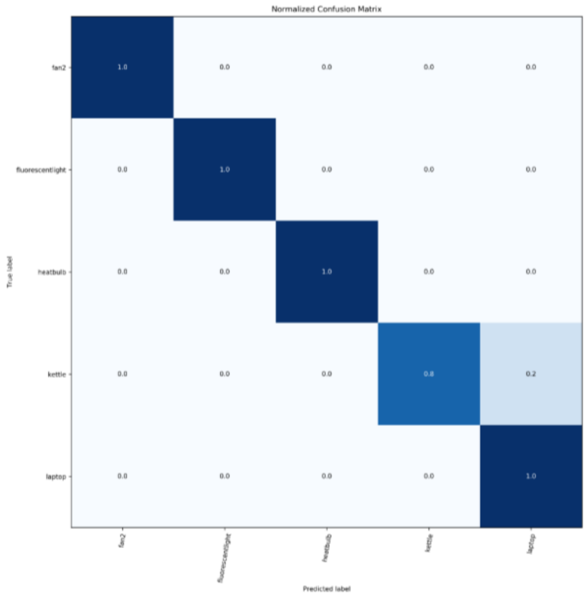


Accuracy= 100.00%
Precision= 100.00%
Recall = 100.00%
F-Beta Score = 100.00%

On Scenario 3



Confusion Matrix



Accuracy= 90.91%
Precision= 93.94%
Recall = 90.91%
F-Beta Score = 90.30%

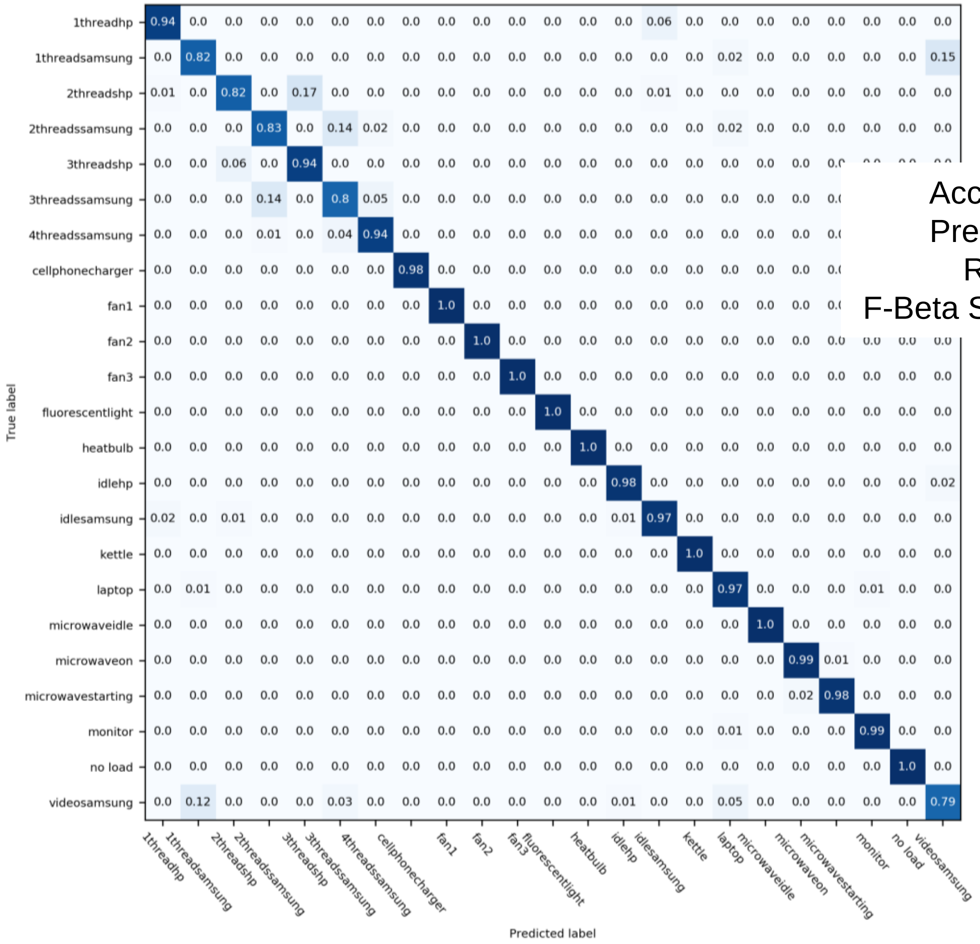
Evaluation 3: Extended Eventless Classification

Best Classification of 23 Classes with SVC @ C=1, gamma = 0.0001 and a linear kernel.

MLP-Classifier, Random Forest Trees and k-NN perform also similar

On Test Dataset of Model

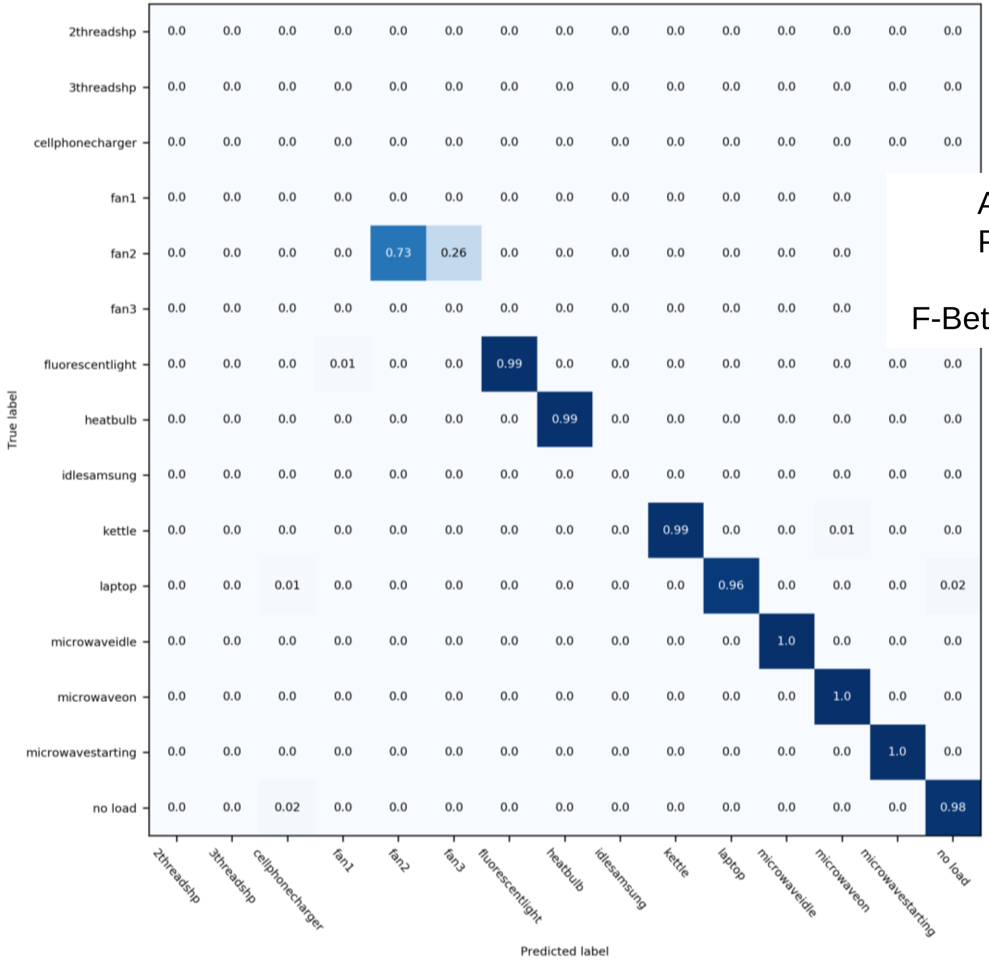
Normalized Confusion Matrix



Accuracy= 97.01%
Precision= 97.03%
Recall = 97.01%
F-Beta Score = 97.01%

On Scenario 3.1

Normalized Confusion Matrix



Accuracy= 96.01%
Precision= 99.63%
Recall = 96.01%
F-Beta Score = 97.62%

Evaluation 3: Feature Weights for different Appliances