EnergySniffer: Home Energy Monitoring System using Smart Phones

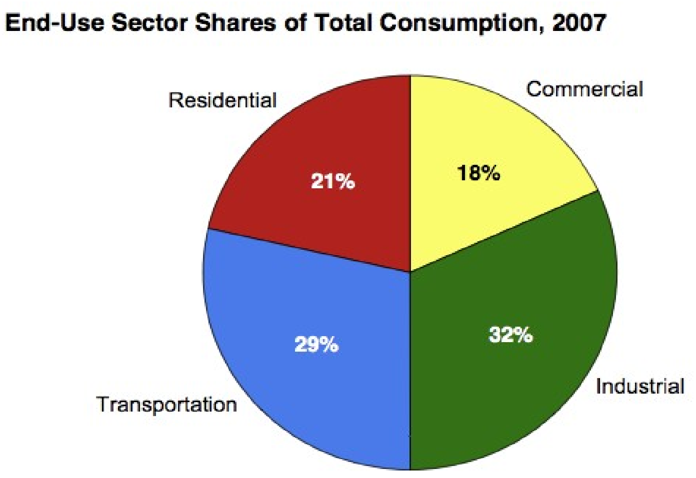
Mostafa Uddin, Tamer Nadeem

OLD DOMINION UNIVERSITY

Why Home Energy Monitoring?

Home energy consumption (21%) is a great concern in modern days.

- Studies shows detailed energy monitoring works as a prerequisite for the conservation of home energy consumption.
  - Detailed energy monitoring: Monitoring energy consumption for each individual machines at home



We use term ”machine” to refer any type of machines at home including home appliances, computing machines, non-computing machines etc.

Industrial Product

- Industries have developed numerous energy monitoring system for household users:
  - TED
  - Control 4
  - GE Nucleus
  - Watts up
  - Alert Me
  - Energy Hub

Researchers Contribution

- At the flick of a switch: S.N Patel etl.
  - Require custom plug-in sensor to be attached with the main power line to detect electric event.
- Viridiscope: Y. kim etl.
  - Require number of sensors to be installed close to the targeted machine.

Why EnergySniffer?

- *TED*:
  - Does not provide energy consumption for each individual machine.
- *Watts Up*:
  - Requires additional inline installation between AP plug and outlets.
- Many such energy monitoring system requires additional invasive and expensive installation of sensor devices.
- Smartphone comes with useful sensors that can be utilized.

What is EnergySniffer?

- EnergySniffer is a simple and flexible energy monitoring system that utilizes smartphone sensors.

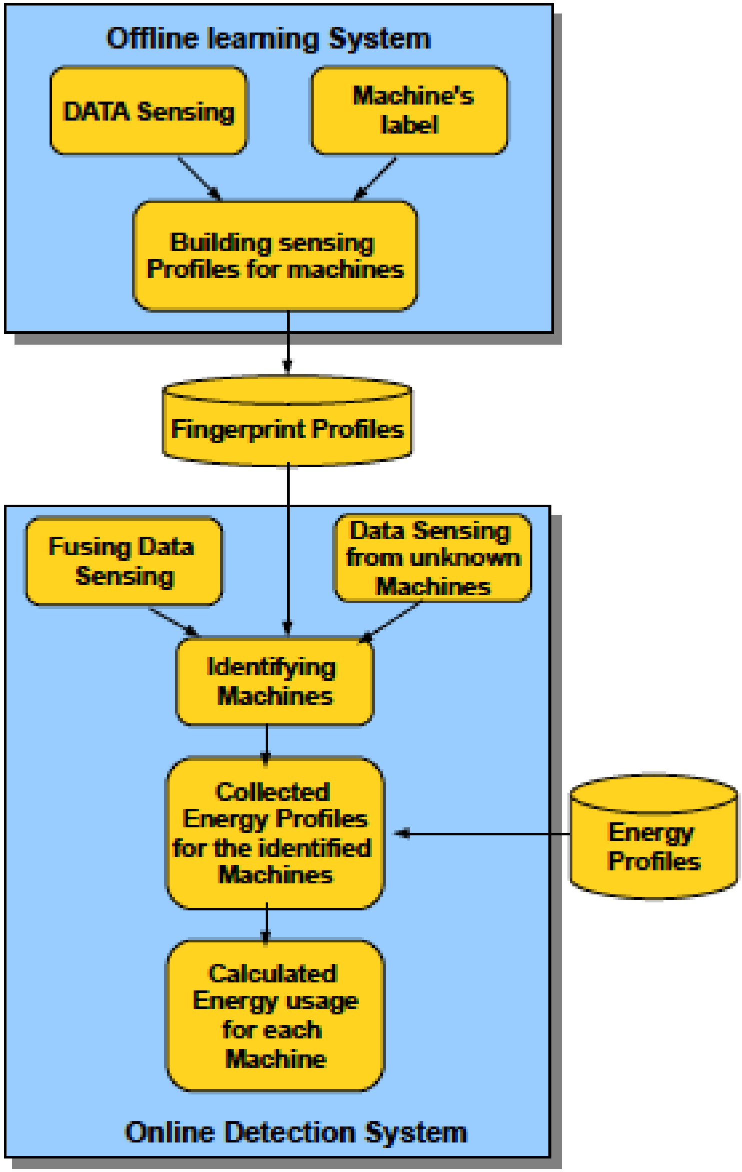
- EnergySniffer exploit sensor such as magnetic sensor, light, microphone, camera, WiFi in smartphone to detect and monitor each operating machines in its vicinity.

EnergySniffer: Advantage

- First, it monitors energy consumption for each individual machines.
- Second, it has very low overhead and also no new hardware is needed to install or maintain.
- Third, it is very flexible in updating software and deploying new services using smartphone application updating service.

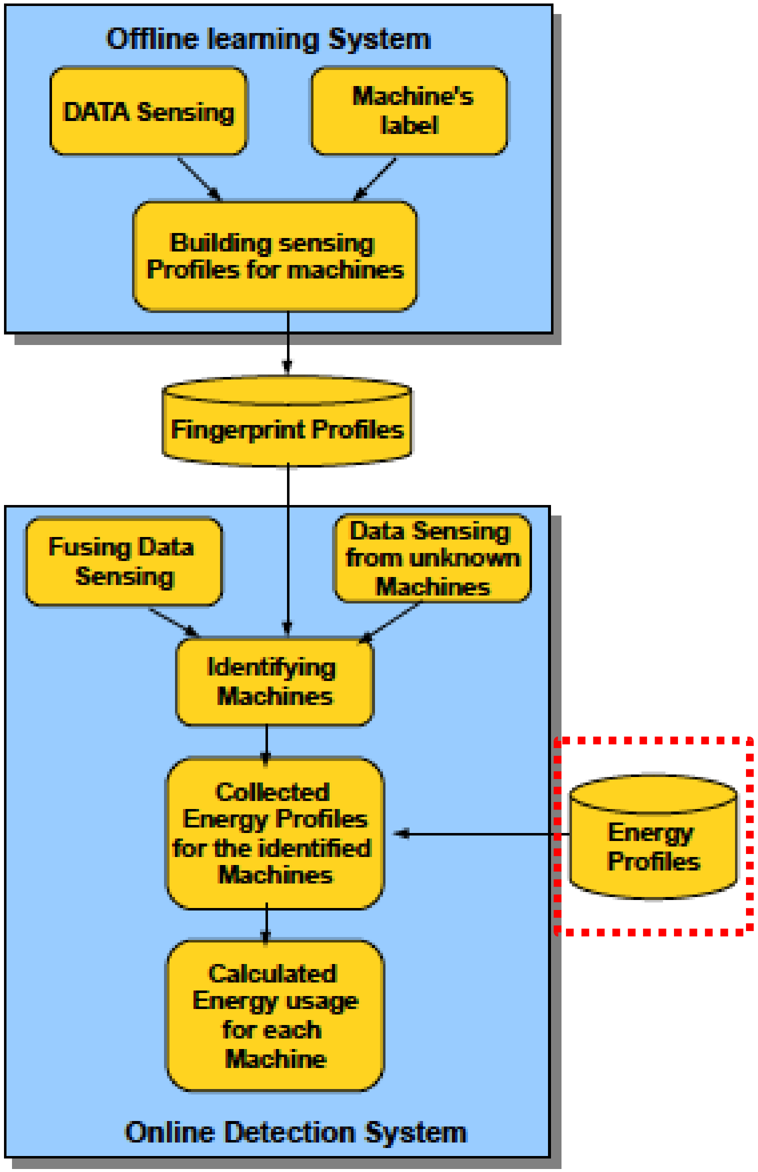
“EnergySniffer” System

- Two components:
  1. Energy Profile
  2. Multi Sensing Framework(MSF)



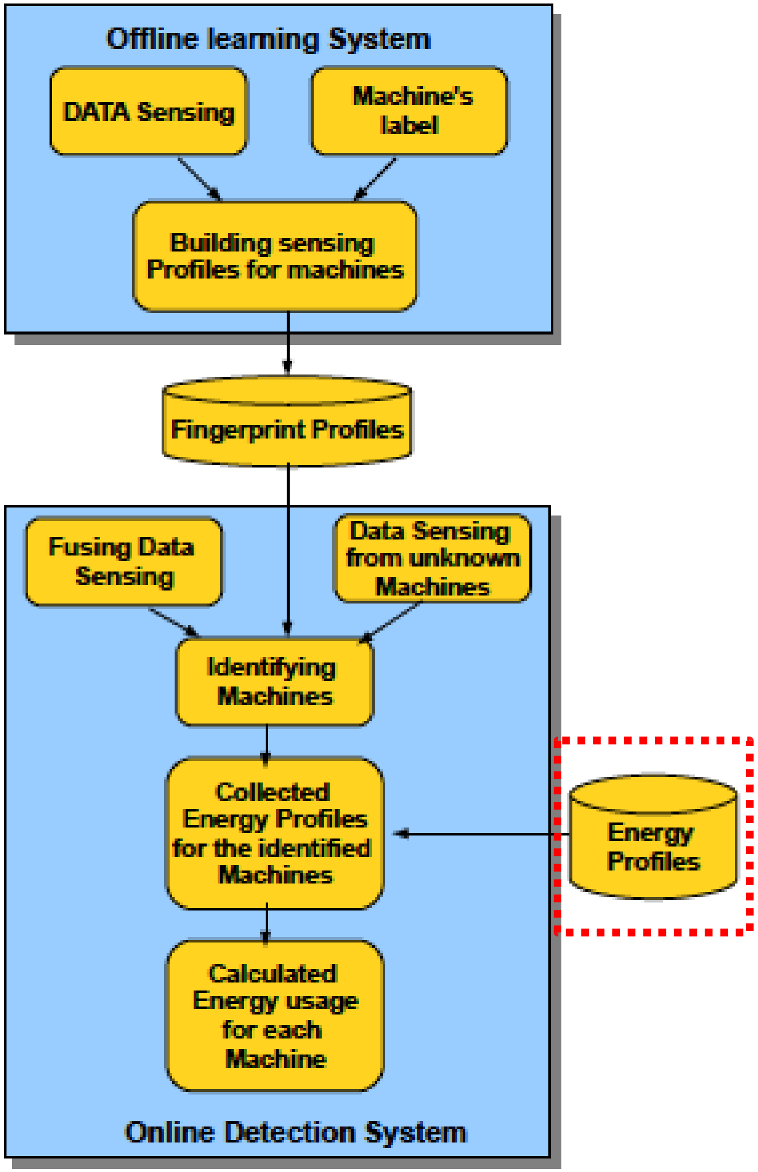
EnergySniffer: Energy Profile

- Energy Profile is a database of list of machines with their corresponding energy consumption profiles.
- Energy Profile is maintained in as a web service instead of storing it locally in phone.



EnergySniffer: Energy Profile

- Initially, Energy Profile database is built from the information provided by the machines's manufacturer.



- The user can download or upload the energy consumption profiles of the machines through mobile application.

Old Dominion University

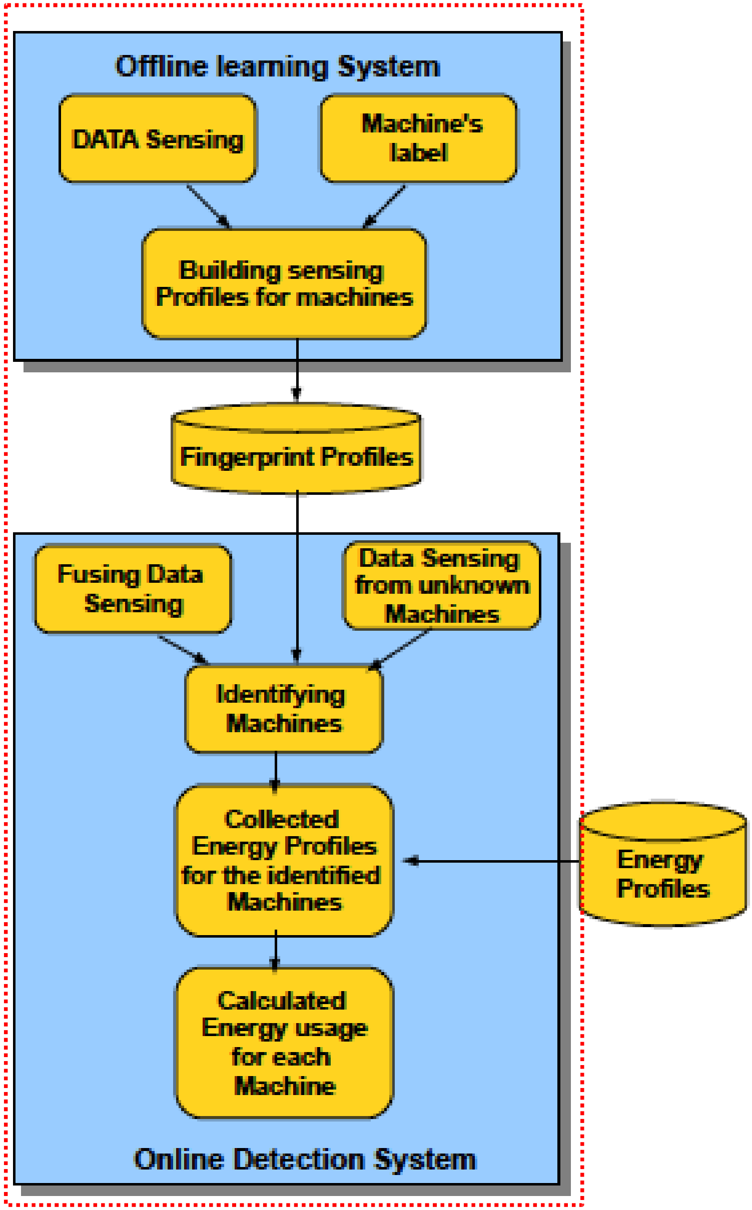
EnergySniffer: Energy Profile

- Example: Energy Profile of a microwave oven

```
<microwave oven>
 <brand>Emerson</brand>
 <model_number>MW8784</model_number>
 <capacity unit="cubic␣feet">0.7</oven>
 ...
 <energy consumption="W/hr">1050</energy>
</microwave oven>
```

EnergySniffer:Multi Sensing Framework

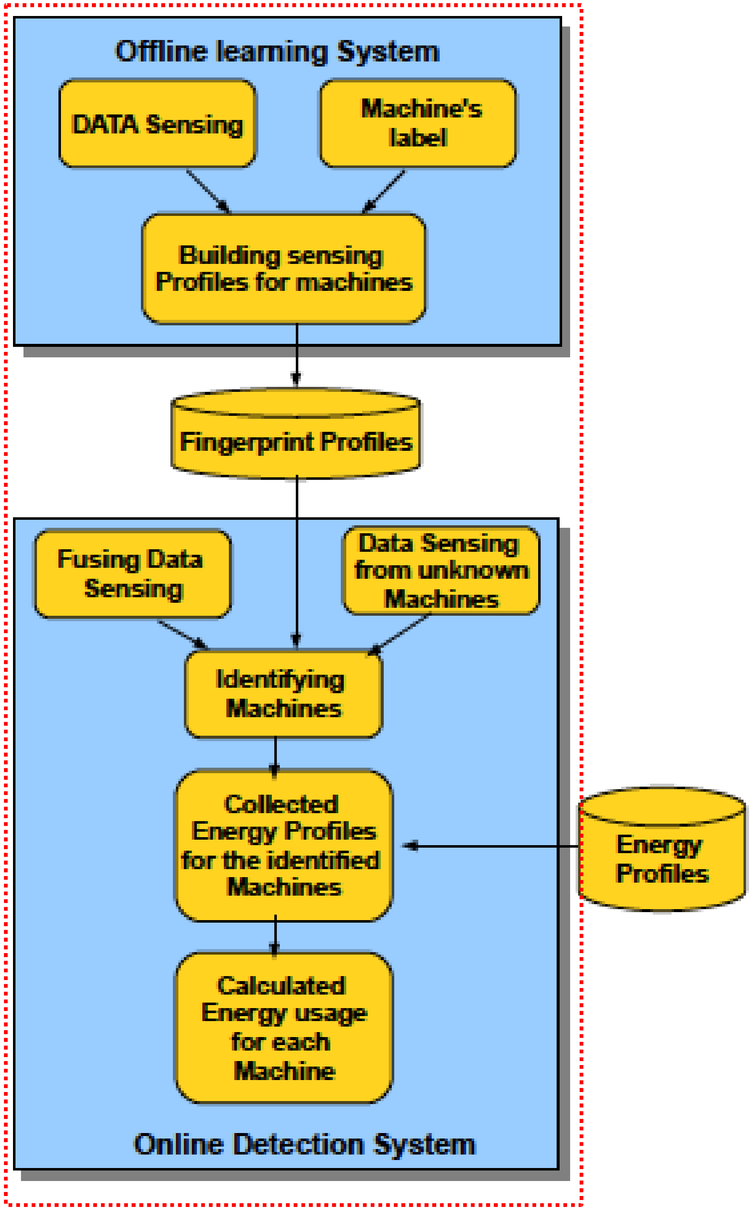
- Consists of Offline Learning and Online Detection Phases.
- Offline learning is responsible to build fingerprint profile for each individual machine.
- Online detection uses the fingerprint profiles to detect and monitor operating machines



EnergySniffer:Multi Sensing Framework

<u>Building fingerprint profile:</u>

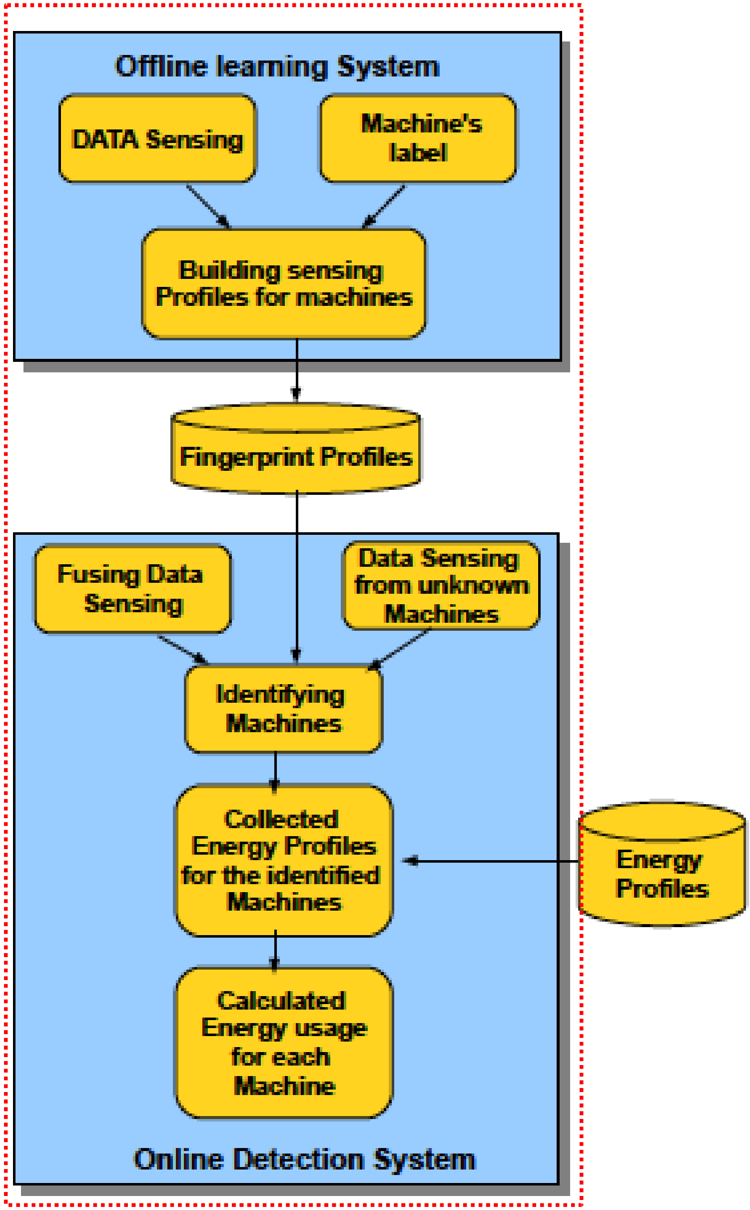
- Identify the sensors that are relevant to the machine.
- Building sensing profile of a machine from the collected sensing data of the identified sensors.
- Combining such multiple sensing profile to build the fingerprint.



EnergySniffer:Multi Sensing Framework

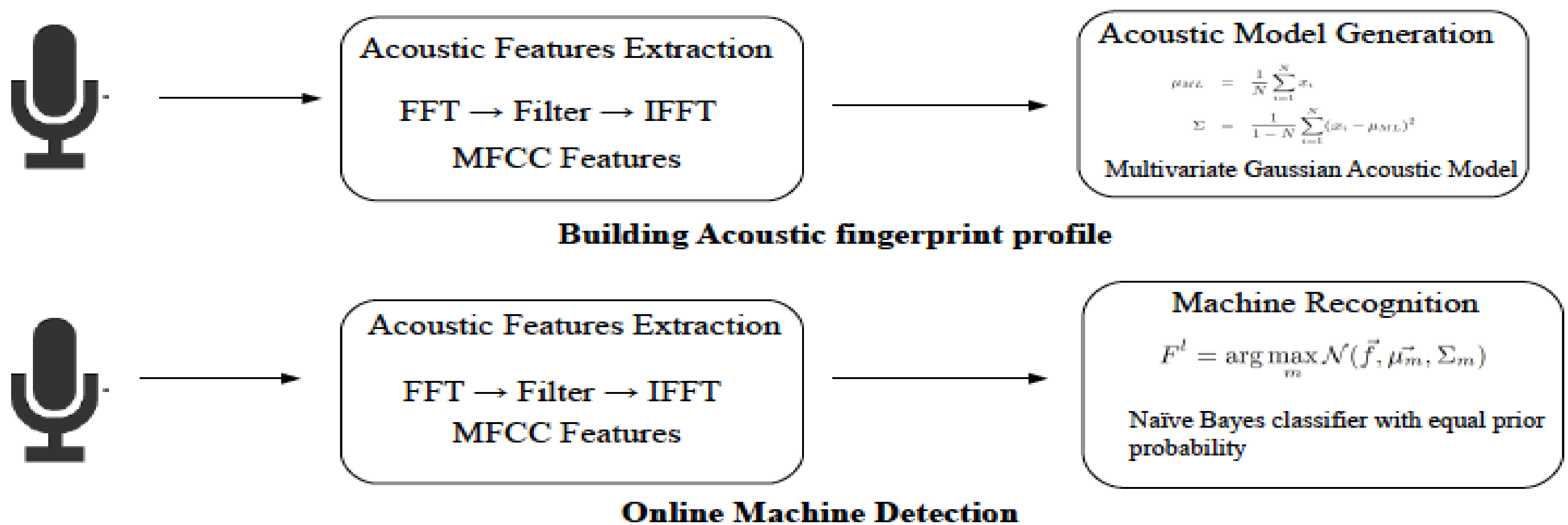
<u>Online Detection and Monitoring Phase:</u>

- Utilizing the machine learning algorithm and fingerprint profiles to detect and monitor running machines.
- Once the system detect a machine, it uses Energy Profile database to track the energy consumption of the machine.



EnergySniffer: Prototype

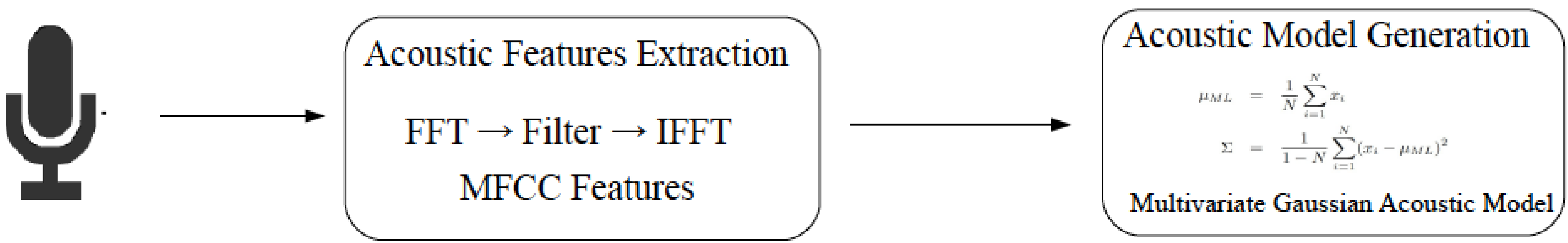
- Implementing and evaluation a prototype of EnergySniffer which only uses sound sensing: Sound Sensing Framework.
- We only use the microphone sensor of the smartphone to build fingerprint profile for each individual machine.



Sound Sensing Framework

Building fingerprint profile

- Acoustic Feature Extraction: Extracts the acoustic features from the raw audio through a features extraction procedure.
- Acoustic Model Generation: Use supervised machine algorithm to generate the multivariate gaussian model from the extracted features.



Building Acoustic fingerprint profile

Acoustic Feature Extraction

- Machines have consistent ambient sound
  - We use larger Frame (fixed size of raw audio data) length to reduce computation.
- Machines show distinguish characteristic in lower frequency range
  - Filters with high weight values at lower frequency.

Acoustic Features Extraction

FFT → Filter → IFFT

MFCC Features

Acoustic Model Generation

- Each model is represented by a Multivariate Gaussian function with certain mean and variance.
- Mean and variance are calculated from the collected training features using Maximum Likelihood algorithm as follows:

Acoustic Model Generation

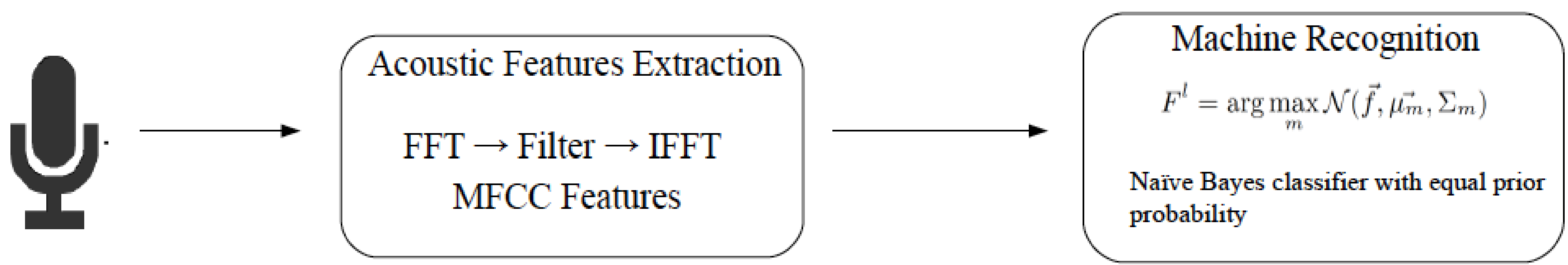
$$\mu_{ML} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\Sigma = \frac{1}{1-N}\sum_{i=1}^{N}(x_i - \mu_{ML})^2$$

Multivariate Gaussian Acoustic Model

Machine Recognition

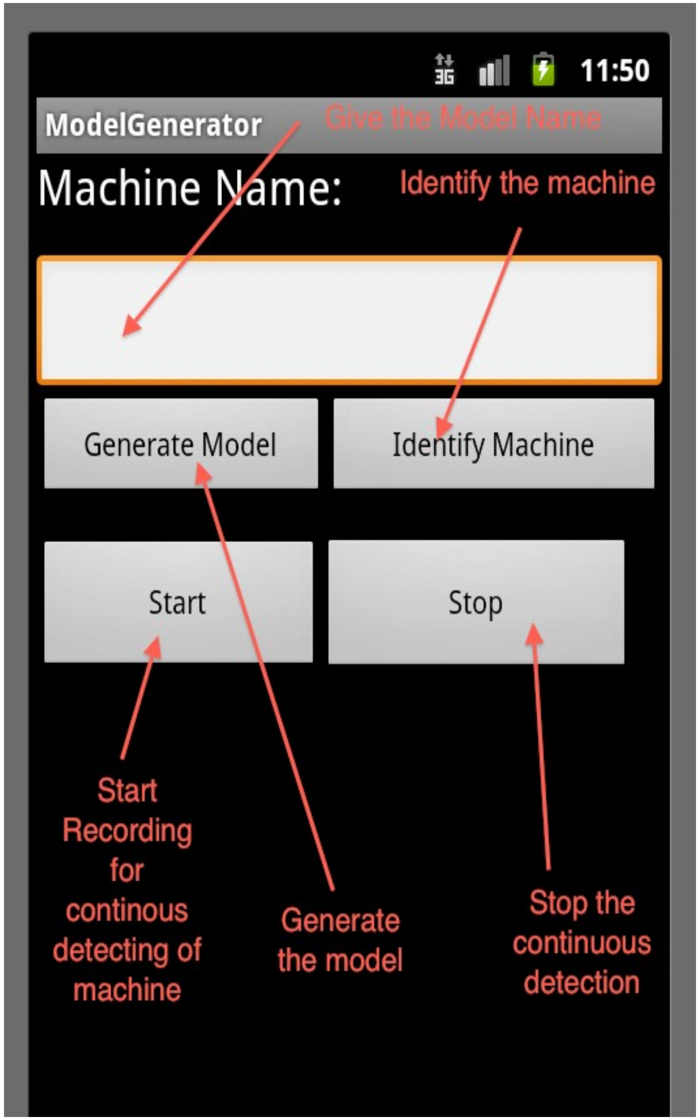
- Collect features from a frame of testing sound data.
- Calculate likelihood values of the features for each Multivariate Gaussian Model of a machine.
- Machine's model with maximum likelihood value represents the ultimate label of the frame.



Online Machine Detection

Implementation

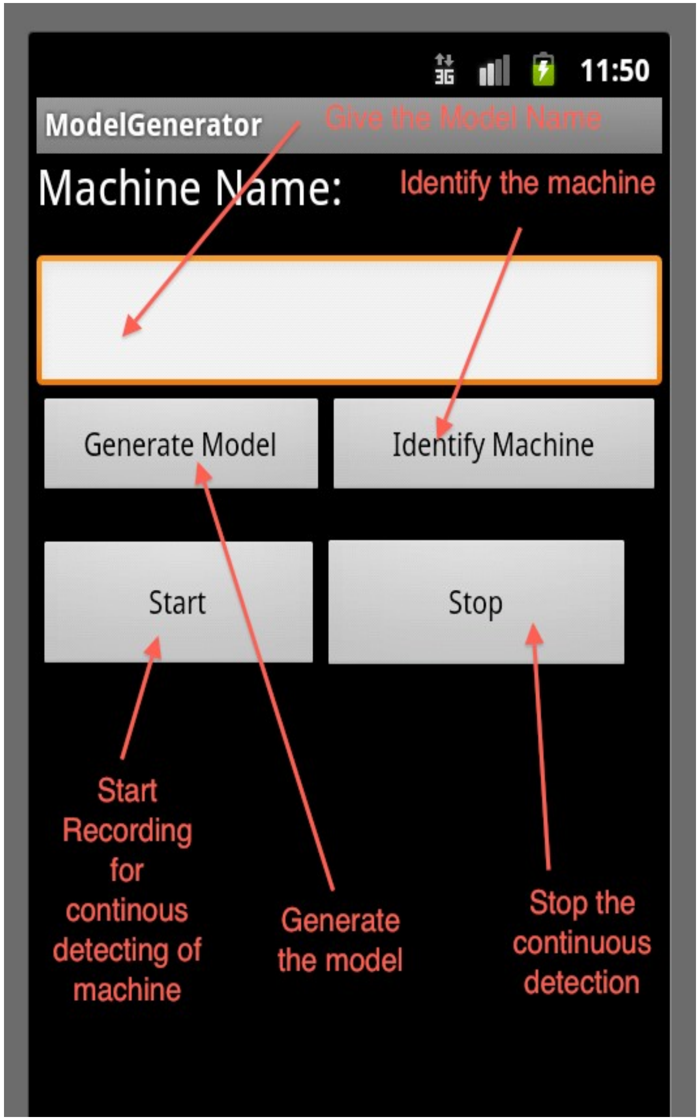
- We implemented the sound sensing framework in Nexus S Android Phone.
- Action:Building a machine's acoustic model.
    - Record 3 sec of sound data of the running machine to generate the machine's acoustic model.



Implementation

- Click “Identify Machine” button to detect what machine is running at smartphone's vicinity.

- Click “Start” button to begin continuous monitoring of the running machines at smartphone's vicinity.

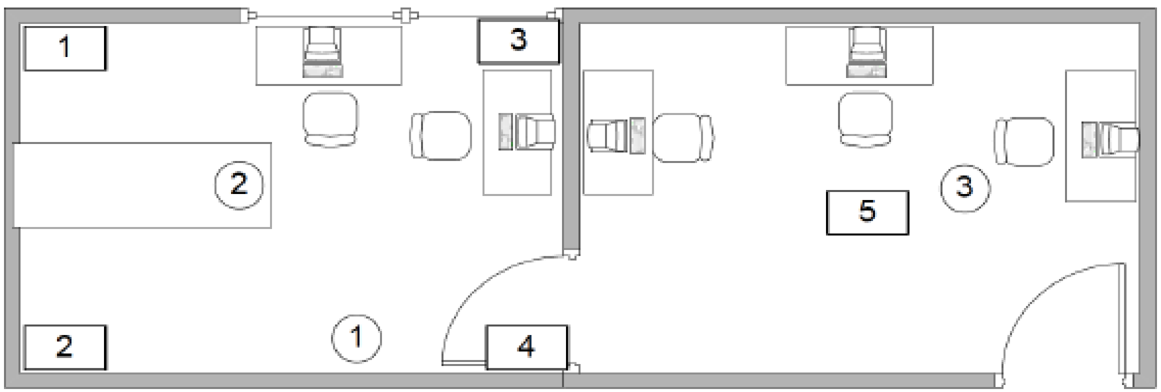


Experiments and Evaluation

- We use the android based implementation to conduct the following three experiments:
  - First experiment evaluates how correctly it identify the operational machine.
  - Second experiment evaluates the performance of continuous real-time monitoring.
  - Third experiment evaluates the accuracy of detecting different operational mode of an operating machine.

Experiments and Evaluation: First Experiment

- We build fingerprint profile for three machines: a microwave oven, a table fan and a vacuum cleaner.



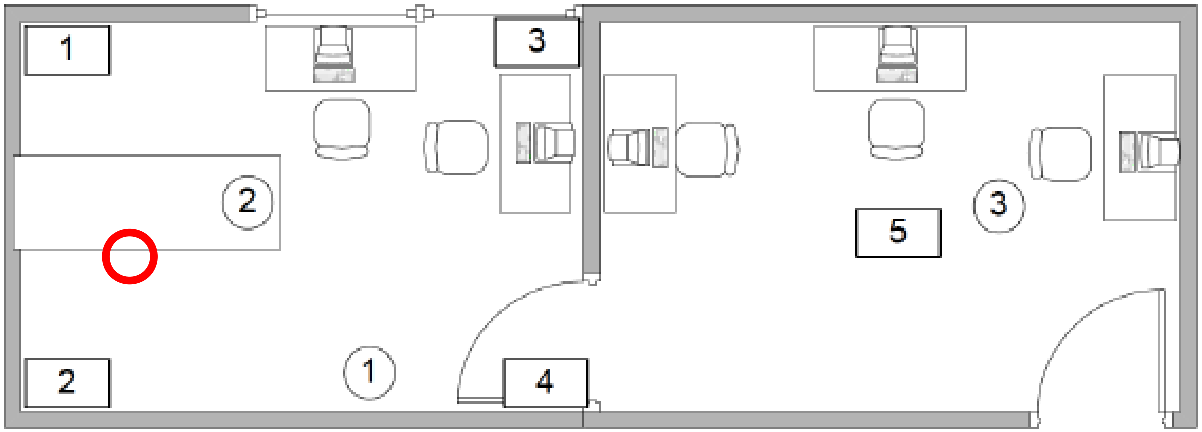
- Circle 1,2,3 shows the position of a microwave, a fan and a vacuum cleaner respectively at our lab office. Square 1,2,3,4 and 5 shows the position from we have identified the current running machine using our prototype application.

Experiments and Evaluation: First Experiment

Machine	Position	microwave	fan	vacuum	none
Microwave	1	**64.06%**	23.43%	3.12%	9.37%
	2	**71.87%**	9.37%	0.00%	18.75%
	3	**65.62%**	28.12%	0.00%	6.25%
	4	**60.93%**	20.31%	0.00%	18.75%
	5	**53.12%**	12.5%	0.00%	34.37%
Fan	1	32.81%	**67.19%**	0.00%	0.00%
	2	42.19%	**54.68%**	0.00%	3.12%
	3	43.75%	**56.25%**	0.00%	0.00%
	4	32.81%	**57.81%**	0.00%	9.37%
	5	**48.44%**	29.69%	0.00%	21.87%
Vacuum	1	3.12%	0.00%	**96.88%**	0.00%
	2	0.00%	0.00%	**100.00%**	0.00%
	3	15.66%	0.00%	**84.34%**	0.00%
	4	0.00%	0.00%	**100.00%**	0.00%
	5	0.00%	0.00%	**100.00%**	0.00%

- Above table shows the accuracy of detecting running machines from different positions.
- In the table “none” is a sound profile that represents wher none of the machine is running.
- From each location we identified the machine at different orientation of the mobile phone.

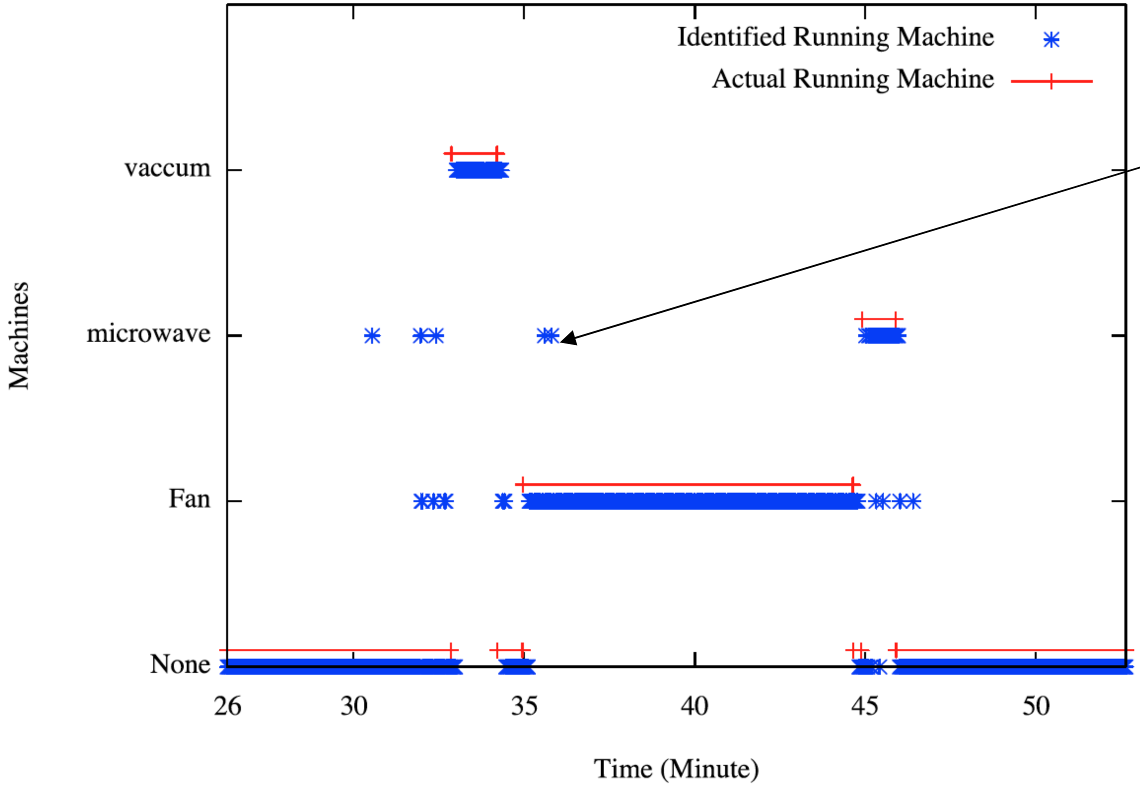
Experiments and Evaluation: Second Experiment



Experiment setup:

- We use same set of machines as first experiment.
- We place the machines at same location as first experiment
- We place the android phone on the center of the table(red circle).
- We run the prototype application 105 minutes for continuous monitoring.

Experiments and Evaluation: Second Experiment



The plot shows some outliers that happens separately. Therefore, these outliers could be easily removed using the smoothing techniques.

Above plot shows the comparison of actual running machine and recognized running machine using our prototype system for 25 minutes.

Experiments and Evaluation: Third Experiment

Experiment setup:

- We use two machines: table fan and dishwasher.
- We build acoustic model for two operational modes of dishwasher: water filling and washing.
- Three acoustic model for three operational modes for the fan: slow speed, medium speed and high speed.
- Testing and training data were taken at different days.

Experiments and Evaluation: Third Experiment

Machine,Mode	Fan1	Fan2	Fan3	DF	DW
Fan slow speed (Fan1)	59	8	9	9	15
Fan medium speed (Fan2)	19	65	6	8	2
Fan fast speed(Fan3)	9	25	59	3	4
Dishwasher waterfilling (DF)	0	1	1	96	2
Dishwasher washing (DW)	3	0	4	4	89

Above table shows the confusion matrix of detecting different operational states of the machines from the testing dataset.

Future Challenges

- Some smartphone sensors are limited in Functionality.
- Smartphone sensors show differences in sensitivity among different devices and Platforms.
- Detecting multiple machines at a time and recognizing running machines from different positions.
- The orientation of the smart phone.

Future Work

- Making the challenges more approachable by knowing the layout/position of the machines in addition with the smart phone location.
  - Extensive experiment on using smart phones location in addition with layout information of the machines, to detect multiple machine.
- Leveraging multiple smart phone with wireless communication for improvising the detection of multiple machines.
- Interfacing additional sensors with the smart phone to create sophisticated fingerprints for the machine.

Thank you

Email: muddin@cs.odu.edu

Web: http://cs.odu.edu/~muddin