TinyML Based Self Diagnostic Kit for Respiratory Diseases

Samson Otieno Ooko

Supervisors: Dr. Jimmy Nsenga
Dr. Didacienne Mukanyiligira

Research Motivation

Increasing deaths and disabilities:

Respiratory disease kill over 4 million people yearly (WHO).

Diagnosis involves an expensive process with few experts:

Expensive lab equipment and reagents used thus high cost of diagnosis

Lack of data to enable complete reliance on AI solutions:

There is inadequate dataset to develop an independent AI algorithm

Aims of the Study

Non-technical aim; Enable day to day cheap or free respiratory disease self-diagnostic kit for the African population.	**Technical aim;** Rely on open source IoT and AI technologies to prototype a rapid test kit for respiratory diseases.

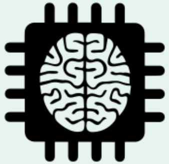
Basis of Integrating IoT, ML and Edge Computing to Predict Respiratory Diseases

Breathomics

The amount of VOCs varies with the health status of an individual, can be used for non invasive diagnosis [10]

Transforming exhaled breaths into data:

Breath samples can be directly sampled into the analytical hardware [14]. Commercial gas sensors applicable in detection breath VOCs [3-8, 15,17]



Using ML to predict respiratory diseases:

Deep learning models are becoming more popular lately due to the availability of large labelled datasets [4]

State of the Art

Commercial Solutions

SpiroNose integrates electronic nose technology and spirometry

Trio-smart a breath test commercial solution.

BreathTracker Analyzer to guide doctors on diagnosis.

Opens source prototypes

18 recent studies that attempt to diagnose respiratory disease also reviewed. Propose use of sensors to collect VOC signatures with data being processed on local computers or cloud platforms.

Gaps Identified

Commercial solutions expensive

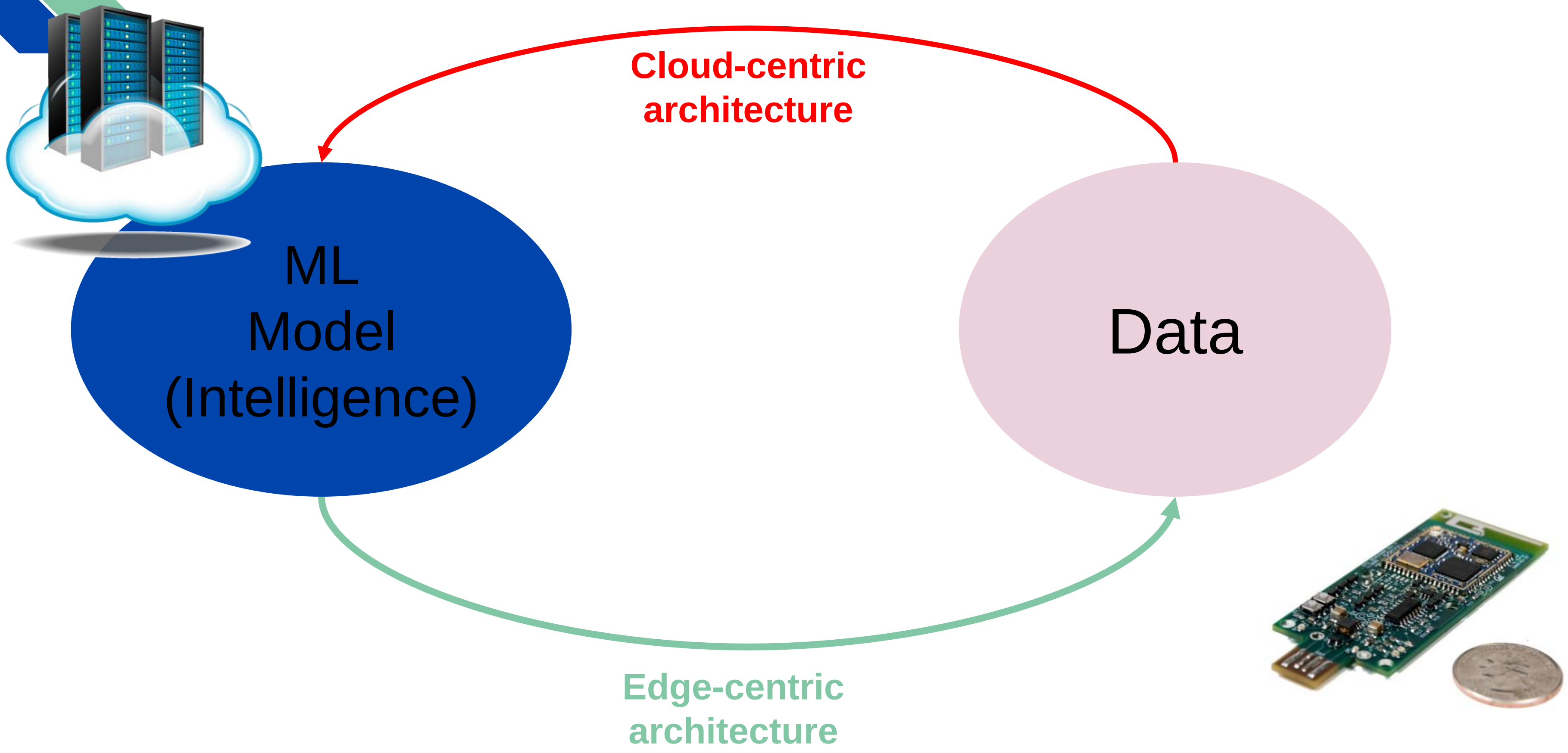
Cloud/Local desktop based AI not portable in African setting

Lack of datasets hence the need for solutions to enhance collection

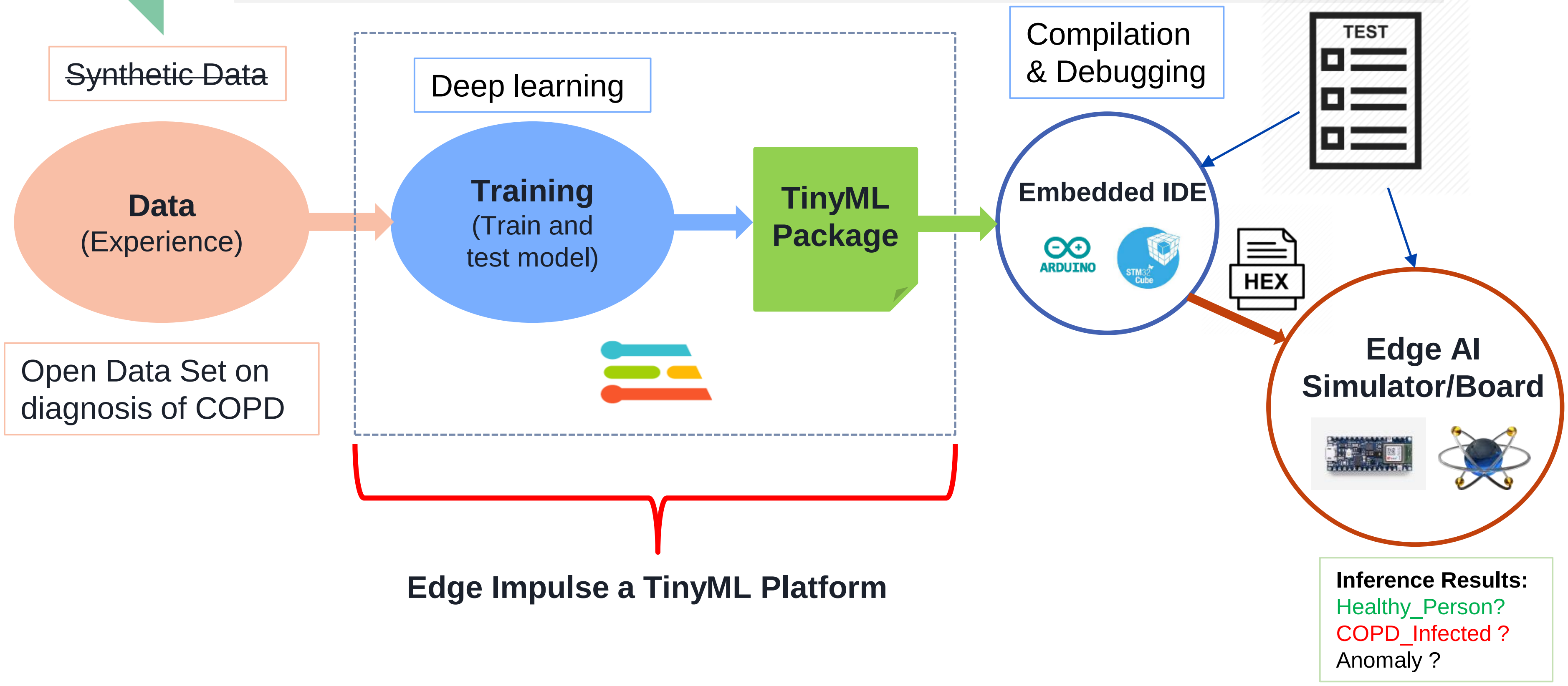
Challenges with cloud AI driven IoT-based applications

Connectivity problems	Privacy Concerns	Energy	Cost Concerns
Enhance portability to areas with connection problems	Concerns about health information leaks	Enhance energy savings for constrained devices	Reduced costs of diagnosis and reusability

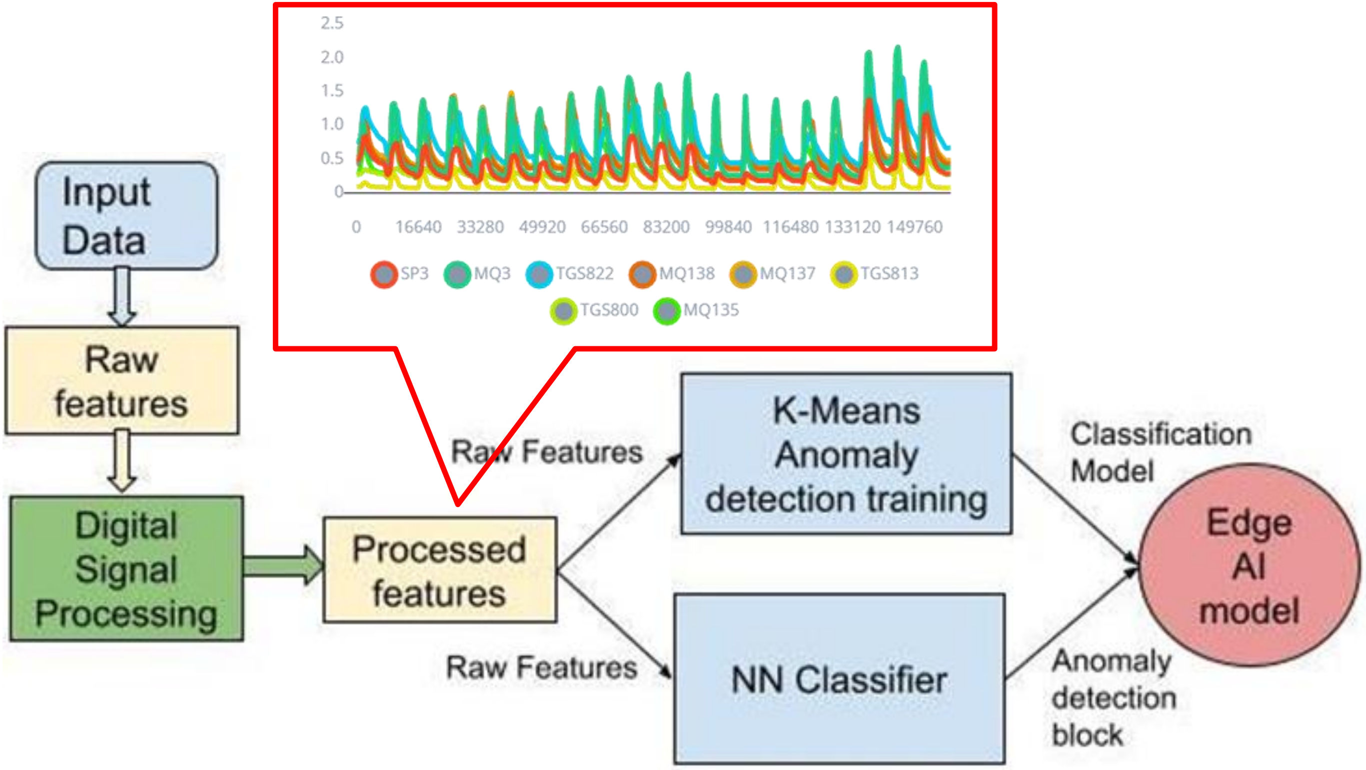
Interestingly, TinyML solves those challenges by moving the AI model to the data source



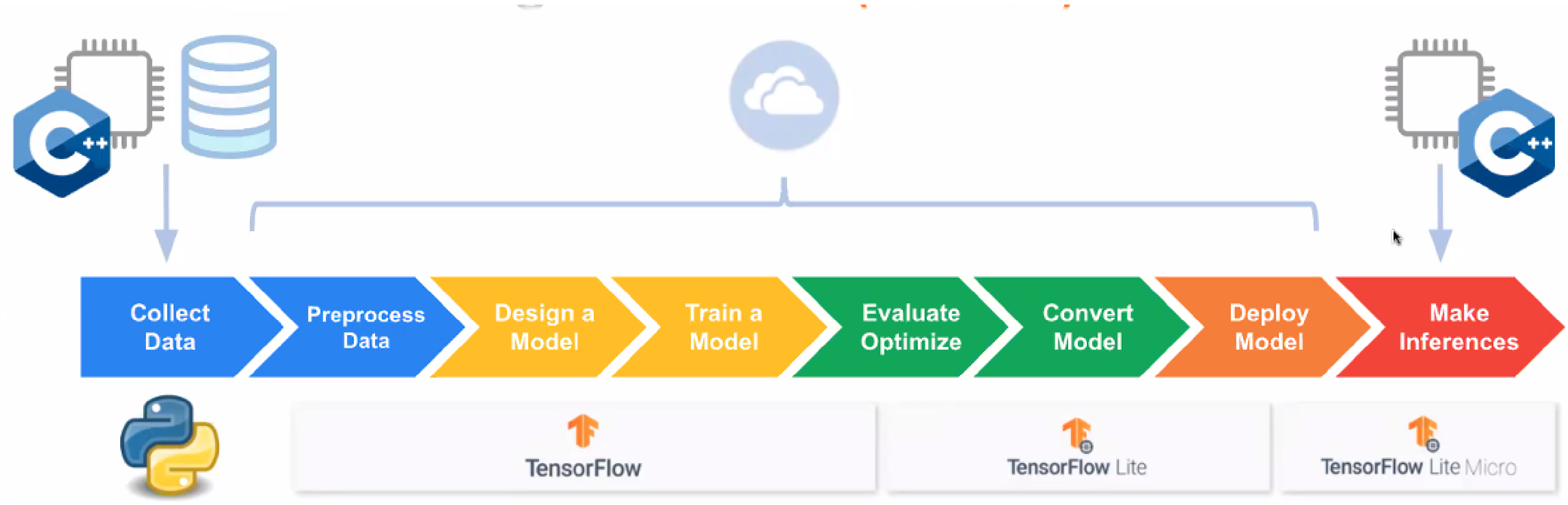
We Developed an Edge AI Tool stack for TinyML applications



Edge AI model training architecture

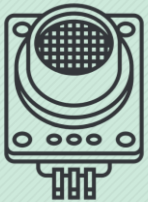


Our machine learning work flow



COPD Open Dataset

Collect Data → Preprocess Data → Design a Model → Train a Model → Evaluate Optimize → Convert Model → Deploy Model → Make Inferences



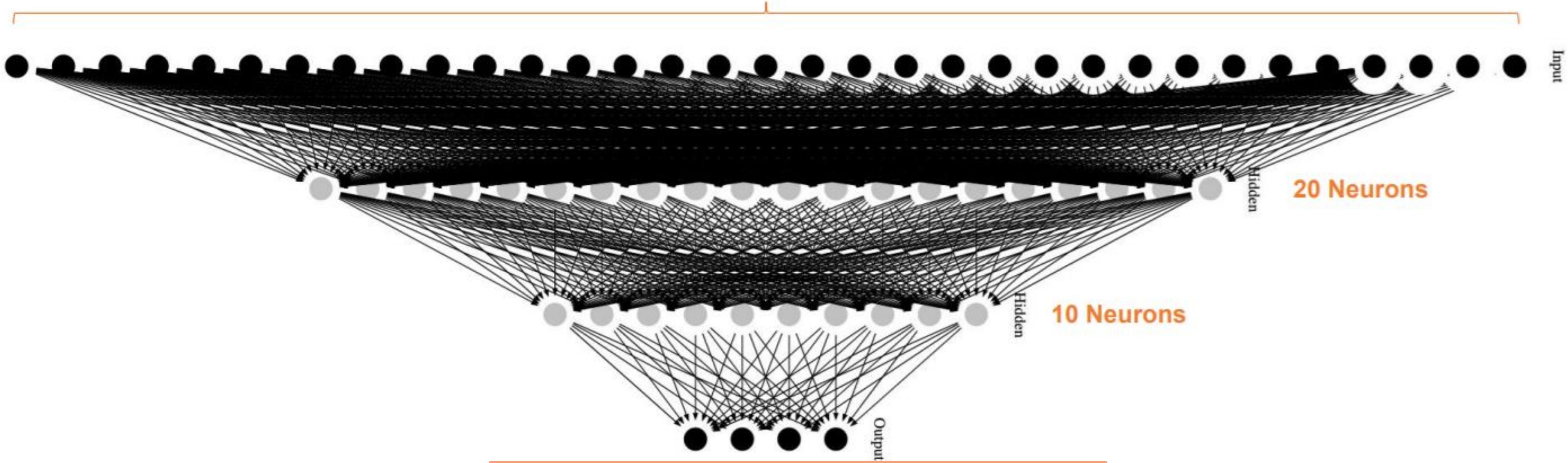
8 gas sensors for: TVOC, Alcohol, Organic Solvent, Formaldehyde, Ammonia, Combustible Gases, CO ,Air Quality

20 Healthy People
40 COPD Infected
20 From Air

4000 samples per person
320,000 Samples
2ms Sampling rate

Model Design

4000 Features



2 Features (Healthy, COPD_Infected)

K-Means Anomaly Detection: model cluster count to 32 with the threshold value of 0.30

50 training cycles at a learning rate of 0.0005

Training Performance

Last training performance (validation set)

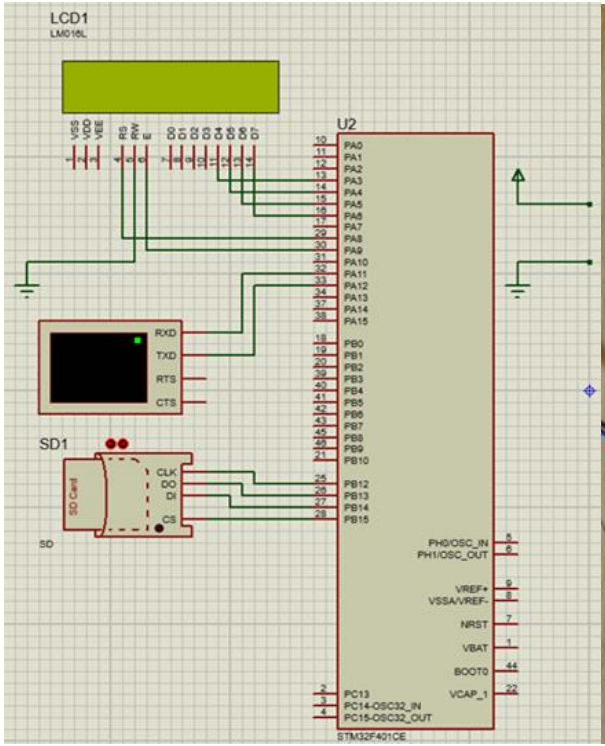
ACCURACY
96.9%

LOSS
0.14

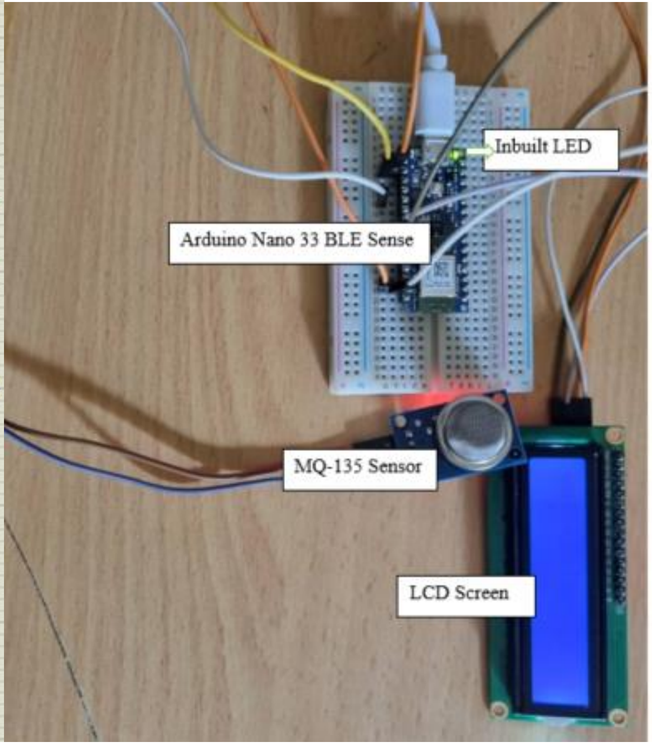
Confusion matrix (validation set)

	COPD_INFECTED	HEALTHY_PERSON
COPD_INFECTED	100%	0%
HEALTHY_PERSON	5.9%	94.1%
F1 SCORE	0.97	0.97

Simulation and Board Tests



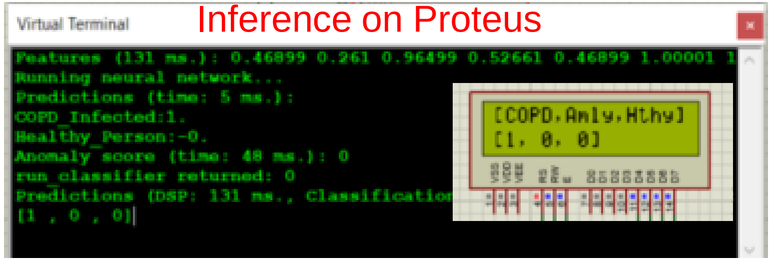
Simulation on proteus, Sensor data through SD Card

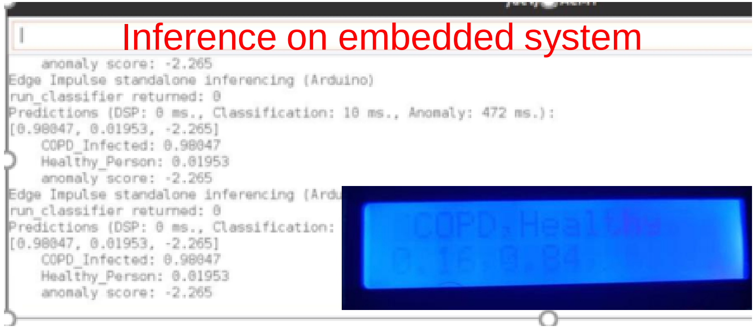


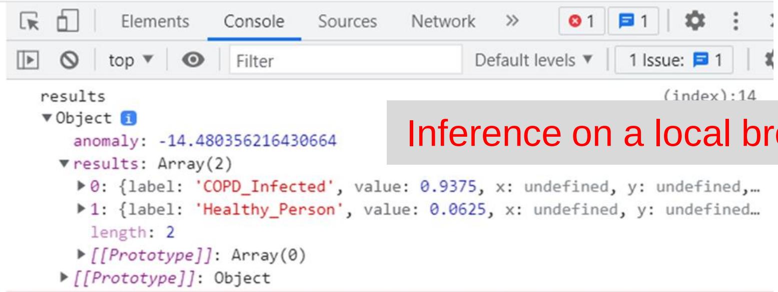
Inference done with real-time data collected by sensors

Initial Results (Inference at the Edge)

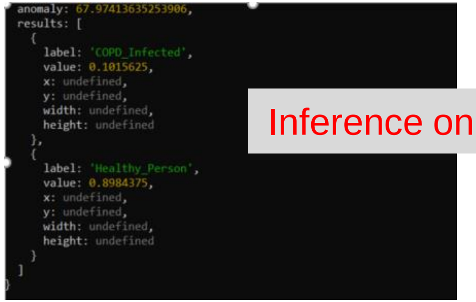
Edge AI model achieves similar inference accuracy in all platforms.





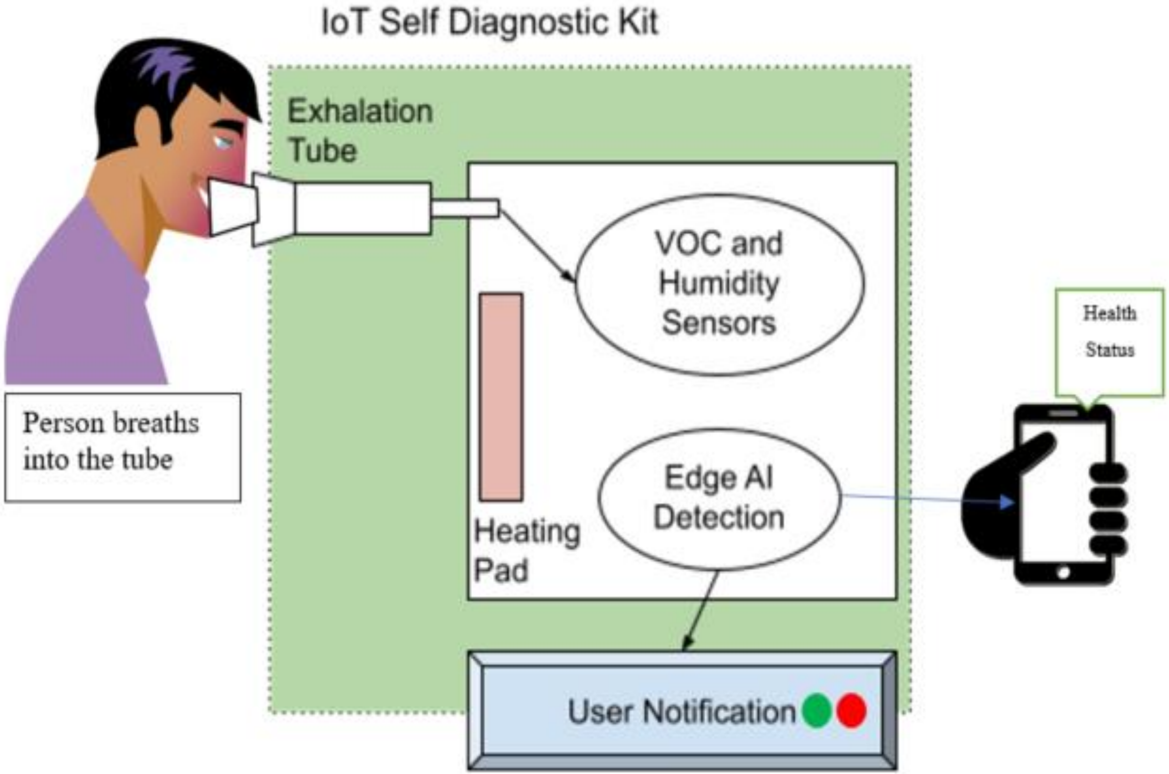


Inference on a local browser



Inference on Node.js application

Ultimate Goal



Technical/Research challenges

Breath Collection

Design of the breath collection tube and sensing unit

Breath VOC Sensors

No commercial sensors for breath VOCs

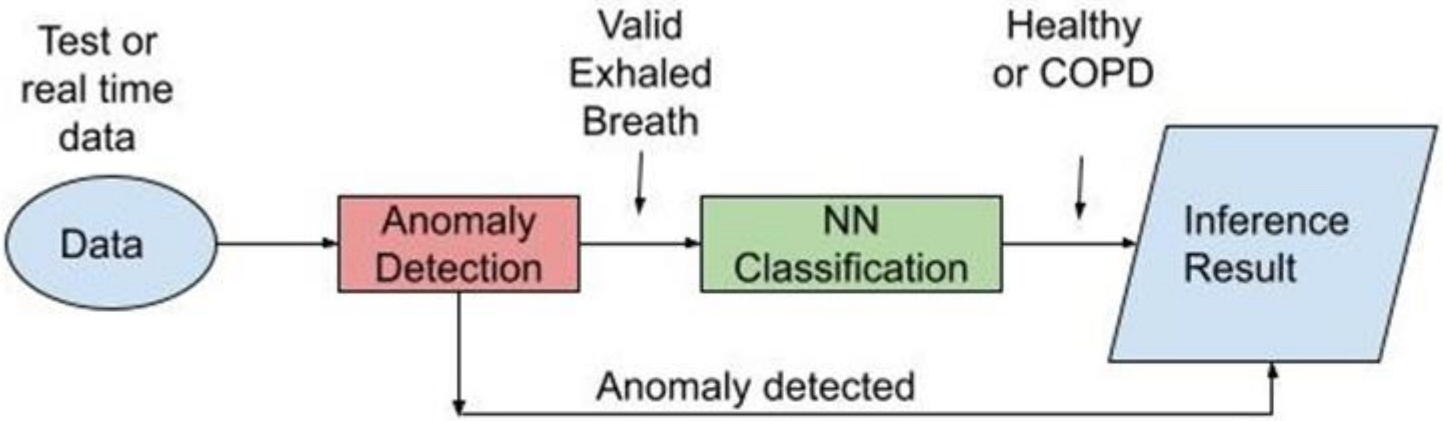
Dataset/AI Algorithm

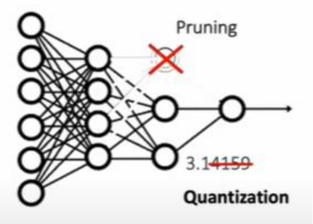
Limited datasets and AI algorithms for diagnosis

Embedded AI Diagnosis

Moving AI from the cloud to edge to enable autonomous system

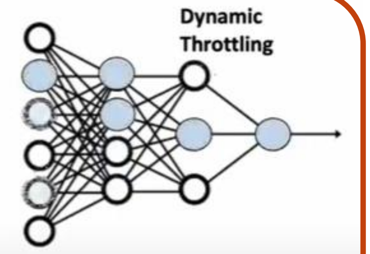
Dynamic Inference process???





Current approach

Statically defined during training → *Dynamically adjust at runtime*



Ultimate goal

On going works

- System level design with available devices
- Finding appropriate alternative VOC sensors
- Design and fabrication of kit
- Use kit to collect new data for training
- Dynamic inference research
- Testing of device functionality