VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

INTERNSHIP 1

Machine learning for internet of thing: Speech emotion recognition

Lecturer: **PhD. TBD**
Students: **2270375 - Nguyễn Trung Thuận**

Ho Chi Minh city - December 2023

Contents

I. Introduction

1. Machine Learning Introduction

Machine learning is an innovative application of artificial intelligence (AI) that empowers systems to learn and improve from experience autonomously without explicit programming.

The learning process begins by observing or obtaining data, such as examples, direct experiences, or instructions, to identify patterns within the data and make informed decisions based on the provided examples. The ultimate goal is to enable computers to learn automatically, free from human intervention, and adapt their actions accordingly.

1.1 Machine Learning algorithm

- **Supervised Machine Learning:** This approach involves leveraging labeled examples from past data to predict the outcome of future events [2]. By analyzing a known training dataset to extract patterns and learn from it, the learning algorithm creates a model to generate predictions about output values. With sufficient training, the system can provide accurate predictions for new inputs. It can also compare its outputs with the correct ones to identify errors and adjust the model accordingly.

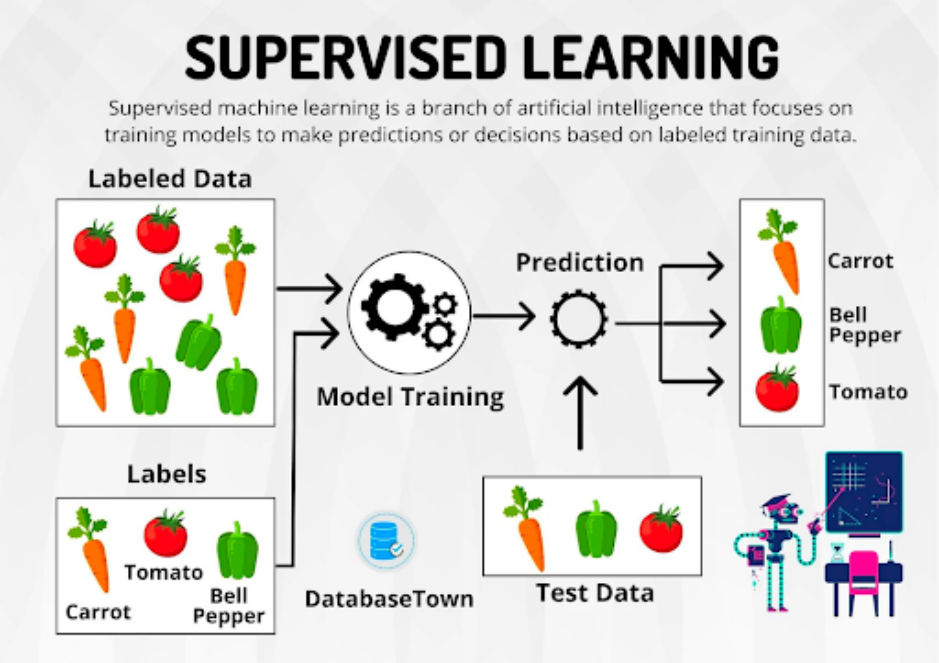

Figure 1: Supervised learning

- **Unsupervised Machine Learning:** In scenarios where training data lacks labels or classification, unsupervised learning comes into play [3]. This approach focuses on inferring hidden structures within unlabeled data to develop a function that describes the data. While this method does not aim to determine specific outputs, it explores the data to uncover valuable insights and patterns.
  For example (fig 1.2), the algorithm can extract the similar properties of fruits within a bag and group fruits with similar properties into different bags (same color or similar shape)

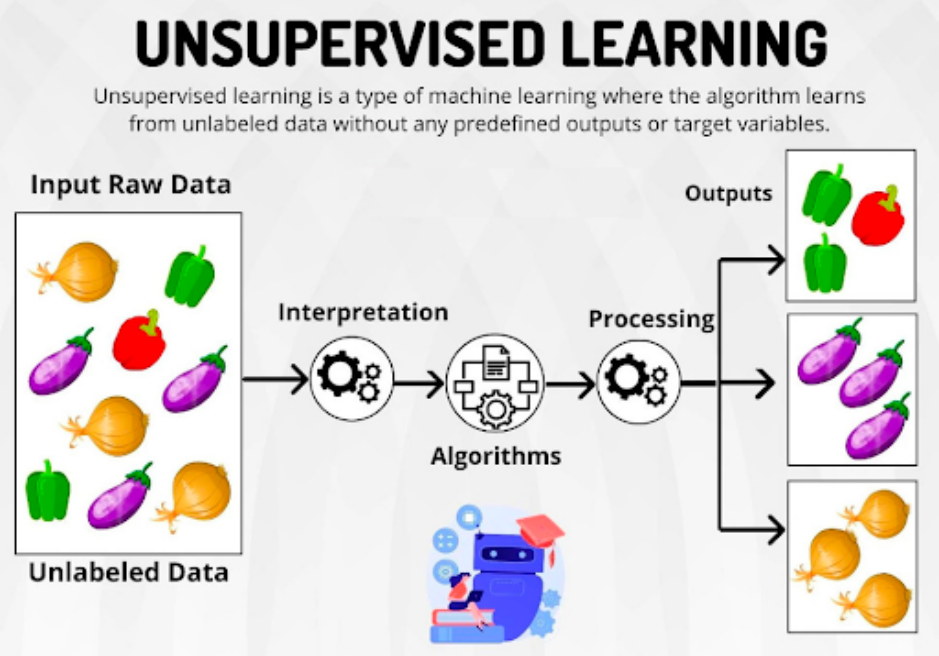

Figure 2: Unsupervised learning

- **Semi-supervised machine learning:** Combining aspects of both supervised and unsupervised learning. This approach utilizes a mix of labeled and unlabeled data for training [4]. Typically, an algorithm works on a small amount of labeled data and a larger pool of unlabeled data. By incorporating this hybrid approach, learning accuracy can significantly improve. Also, obtaining labeled data is expensive and time-consuming, as the model can leverage the large pool of unlabeled data to improve its performance

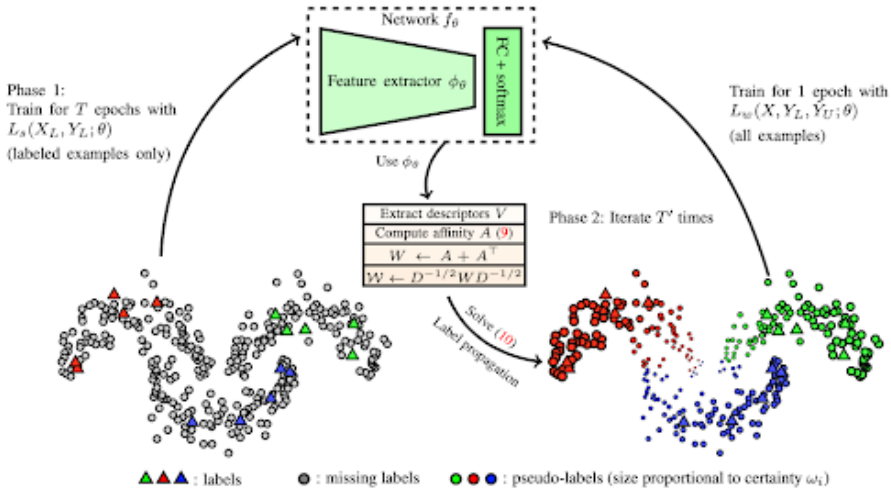

Figure 3: semi-supervised learning

- **Reinforcement learning:** This learning method involves an interactive process where an agent interacts with its environment, taking actions and receiving rewards [5]. Reinforcement learning is characterized by trial and error in making decisions and acquiring rewards. This approach enables machines and software agents to determine the optimal behavior within a specific context, maximizing their performance. Simple reward feedback guides the agent's learning process, known as the reinforcement signal. When combined with AI and cognitive technologies, machine learning becomes a powerful tool for processing vast amounts of information, yielding faster and more accurate results to identify the most profitable opportunities or potential risks. Reinforcement learning is commonly used in areas such as robotics, game-playing, and autonomous systems.

2. Machine Learning in Everyday Applications

2.1 Virtual Personal Assistants

Virtual personal assistants like Siri, Alexa, and Google Now have become popular examples of AI-driven applications. These assistants provide information and perform tasks when prompted using voice commands. By leveraging machine learning, they refine the information they provide based on user interactions and preferences. These assistants are integrated into platforms like Amazon Echo, Google Home, and smartphone software like Samsung Bixby.

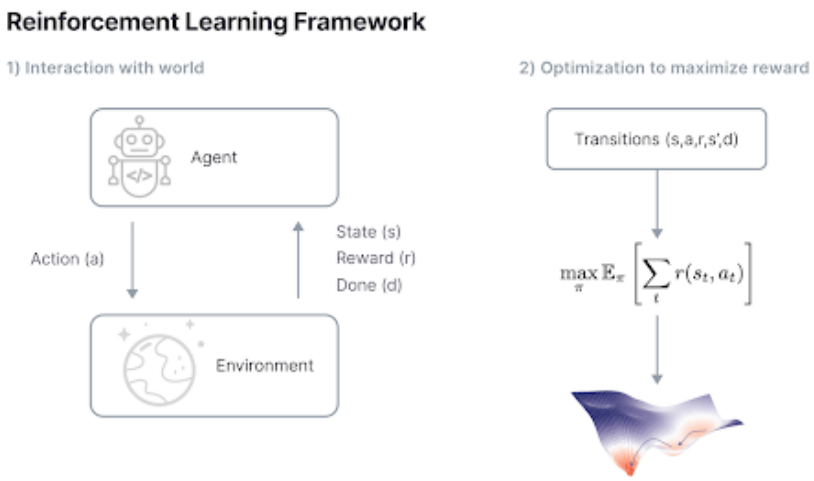

Figure 4: Reinforcement learning

2.2 Predictive in communication

Machine learning plays a crucial role in various aspects of communication. Traffic predictions rely on data from GPS navigation services to generate real-time traffic maps (for example, Google Maps) [6]. Machine learning algorithms can estimate areas prone to congestion by analyzing patterns and historical data. Online transportation networks also utilize machine learning to optimize routes, minimize detours, and predict demand, enhancing the efficiency and cost-effectiveness of shared mobility services [7].

2.3 Video surveillance

Machine learning revolutionizes video surveillance systems by enabling AI-powered detection of potential criminal activities. These systems can identify unusual behaviors, such as prolonged motionlessness or stumbling, and alert human attendants to prevent incidents. By continuously learning from data, machine learning enhances the accuracy and effectiveness of video surveillance, improving overall security [8].

2.4 Social media services

Machine learning underpins various features in social media platforms, enhancing user experiences and personalization. Examples include ”People You May Know” recommendations, where machine learning analyzes user connections, profile visits, and shared interests to suggest potential friends. Face recognition algorithms leverage machine learning to identify and tag individuals in

photos. Additionally, machine learning powers computer vision techniques that identify objects in images and recommend related content [9],

2.5 Recommendation System

Machine learning is instrumental in recommendation systems used by streaming platforms, e-commerce websites, and social media platforms. These systems analyze user preferences, browsing history, and behavior to suggest personalized content, products, or connections. By continuously learning from user feedback and interactions, recommendation systems improve their accuracy and provide tailored recommendations that cater to individual user preferences [10].

3. Internet of Things

3.1 Introduction

The Internet of Things (IoT) refers to a networked system where various physical objects are interconnected and accessible via the Internet. These objects, often called "things," can range from individuals with heart monitors to automobiles with sensors. Each object is assigned an IP address, enabling them to collect and transmit data over a network without manual intervention. By leveraging embedded technology, these objects can interact with their internal states or the surrounding environment, influencing their decisions.

Typically, IoT's primary components are devices capable of connecting to the Internet and are divided into Sensors and actuators. Sensors are devices capable of detecting or measuring specific phenomena, collecting related data, and transmitting them to other entities (other IoT devices or application servers in the cloud). Some examples of IoT sensors include GPS devices, thermostats, and temperature sensors. To meet the cost-effectiveness requirements of IoT solutions, sensor nodes typically employ small-scale embedded systems, often utilizing 8-bit microcontrollers with limited storage capacity. This design enables them to operate on battery power for extended periods, sometimes lasting years. Furthermore, a wide range of networking protocols allows sensor nodes to integrate seamlessly into existing infrastructures and diverse operational conditions. This flexibility greatly facilitates the deployment of IoT solutions across various domains [11].

On the other hand, Actuators are physical devices that can execute commands transmitted by a control center or act according to some programmed conditions to create a change in the surrounding environment [12].

3.2 Applications of IoT

IoT technology enhances mobility services, bolsters public safety, and automates city household systems. Within an area of smart city, one notable application is intelligent transportation, which focuses on optimizing road infrastructure and facilitating efficient route planning for drivers. It involves innovative solutions like smart traffic signals and sensors that monitor and manage traffic systems across the road network. By facilitating smoother traffic flow and reducing congestion, these technologies contribute to improved transportation experiences. However, the scope of smart city services extends beyond transportation. It encompasses various aspects of urban life, including public safety, environmental sustainability, efficient delivery of municipal services, smart grid systems, and integrating physical infrastructure with the digital realm. [13]

Figure 5: Applications of IoT

Home automation is another section (considered a sub-section of smart city), and control systems have significantly transformed our living environments.

They have diverse applications within homes, including entertainment and smart living, surveillance, and safety management. Home automation refers to integrating IoT technology into a standard home environment to provide a secure and comfortable lifestyle. These systems rely on intelligent, self-adaptive mechanisms that analyze and evaluate user behaviors, predict future actions, and interact accordingly. Home automation systems utilize image detection and facial recognition models embedded in an intelligent control system. This control system is connected to sensors such as light, motion, water leak, smoke, and CCTV. These devices communicate with each other through a gateway distributed across a home area network. The home control system connects different subsystems collaborating to model user actions and gather environmental information, such as temperature, humidity, noise, visibility, and light intensity, to enhance learning. For instance, lighting and AC temperature can be controlled and automated based on users' needs and movements within the home environment. Home automation research extends beyond energy optimization and encompasses health monitoring and security measures. By leveraging innovative IoT technologies, users can remotely access surveillance cameras within their homes through mobile devices. Additionally, stakeholders can employ door and window sensors to ensure home safety and security from a distance.

IoT has also expanded its application to the industrial sector. Industrial IoT harnesses the capabilities of IoT technology in the business and economic sectors to automate previously complex manual operations, meeting consumer demands while reducing production costs. Various industrial domains, including warehouse operations, logistics services, supply chain management, and agricultural breeding, can benefit from machine-to-machine (M2M) intercommunication, facilitating optimal industrial operations. For example, the application of communicating sensors in agricultural systems.

This system utilizes smart agriculture technology to monitor and analyze environmental parameters using sensors such as ZigBee, EnOcean, Z-wave, and ANT, which are specifically designed for soil moisture monitoring and harvesting. These sensors are automated to assess the plant's condition and gather relevant data through an IoT platform. Sensors utilize collected information to take appropriate actions, such as determining the optimal timing for irrigation by consulting a weather forecasting service available in the Cloud. It ensures

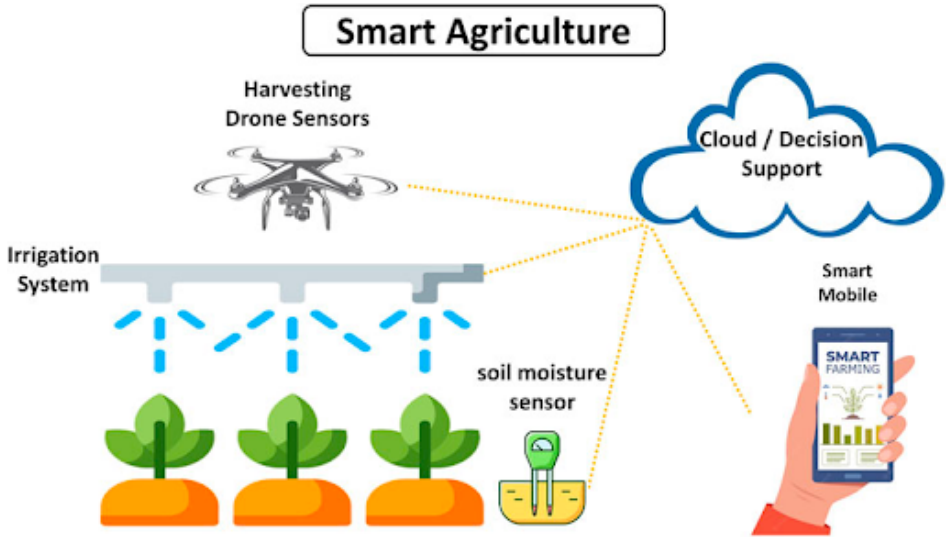

Figure 6: 2.2 Smart agricultural system

the efficient utilization of water resources while maintaining crop health. In the Healthcare domain, IoT also has a wide range of applications. IoT sensors and devices have transformed the landscape of portable and wearable medical devices, expanding their applications from fitness and wellness to medically qualified devices suitable for use in hospitals and healthcare facilities. This shift has facilitated the integration of remote patient monitoring (RPM) in healthcare settings, particularly for patients with chronic diseases. As a result, significant efforts are made to advance RPM systems by leveraging well-established IoT infrastructures and standards in the healthcare domain. These RPM systems aim to match or surpass the performance of existing monitoring and examination methods employed in hospitals and healthcare facilities. For instance, continuous heart rate monitoring and immediate detection of irregular heartbeats traditionally required patients to be hospitalized or connected to devices like Holter monitors for long-term cardiac diagnosis. However, these setups limited patient mobility due to device size and the number of connected wires.

Furthermore, hospitals expended substantial resources on providing long-term cardiac monitoring, which was often unavailable, particularly in low or middle-income countries. Remote patient monitoring systems effectively address these challenges and reduce mortality rates associated with chronic diseases such as heart disease and diabetes. IoT platforms and devices have played a significant role in accelerating the development and integration of RPM systems into existing healthcare infrastructures. Consequently, a typical RPM implementation encompasses various services, including but not limited to data acquisition,

tracking, communication, automated analysis, diagnoses, and notification systems.

4. Neural Networks Deep learning

4.1 Neural Networks Deep learning introduction

A neural network is a series of algorithms that aims to discover underlying relationships in a dataset by simulating the functioning of the human brain. Neural networks can adapt to changing input, generating optimal results without redesigning output criteria. This concept, rooted in artificial intelligence, is increasingly popular in developing intelligent systems.

In finance, neural networks are utilized in various processes such as time-series forecasting, algorithmic trading, securities classification, credit risk modeling, and constructing proprietary indicators and price derivatives.

Moreover, neural networks have been widely used for image recognition tasks such as object detection, image classification, and facial recognition. In natural language processing, it has shown remarkable performance in various tasks such as language translation, sentiment analysis, and text generation. Also, the application of neural networks has been demonstrated in speech recognition, autonomous vehicles, and medical diagnosis. The functioning of a neural network resembles that of the human brain's neural network. A "neuron" in a neural network is a mathematical function that collects and categorizes information based on a specific architecture. The network bears similarities to statistical methods like curve fitting and regression analysis [14].

A neural network consists of interconnected layers of nodes, where each node functions as a perceptron, similar to multiple linear regression. The perceptron passes the signal from multiple linear regression through an activation function, which can be nonlinear

In a multi-layered perceptron (MLP), perceptrons are organized into interconnected layers. The input layer receives input patterns, while the output layer provides classifications or output signals corresponding to the input patterns. For example, input patterns may consist of technical indicators for security, and potential outputs could be "buy," "hold," or "sell".

Hidden layers in the neural network adjust the input weightings until the network's margin of error is minimized. It is hypothesized that hidden layers extract significant features from the input data that have predictive power for

the outputs. This process, known as feature extraction, serves a purpose similar to statistical techniques like principal component analysis [15].

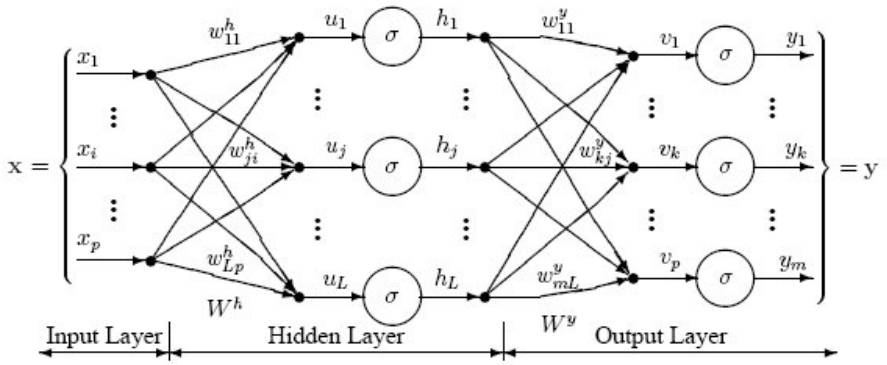

Figure 7: Multi-layer perceptron

A Deep Neural Network (DNN) is a type of artificial neural network (ANN) that consists of multiple layers between the input and output layers. The DNN employs various mathematical operations to transform the input into the desired output, accommodating both linear and non-linear relationships.

Deep learning is a specific function within the field of artificial intelligence (AI) that emulates the information processing and pattern recognition capabilities of the human brain. It falls under the umbrella of machine learning and involves the use of neural networks capable of unsupervised learning from unstructured or unlabeled data. It is also referred to as deep neural learning or deep neural network [15].

4.2 Machine learning vs Deep learning

Deep neural networks are characterized by their deeply nested network architectures, typically consisting of multiple hidden layers. These networks employ advanced neurons that utilize operations like convolutions and multiple activations within a single neuron, going beyond the simple activation functions used in traditional artificial neural networks (ANNs). It enables deep neural networks to process raw input data and automatically learn representations relevant to the learning task. This capability is commonly referred to as deep learning. In contrast, simple ANNs (such as shallow autoencoders) and other machine learning (ML) algorithms like decision trees are considered part of shallow machine

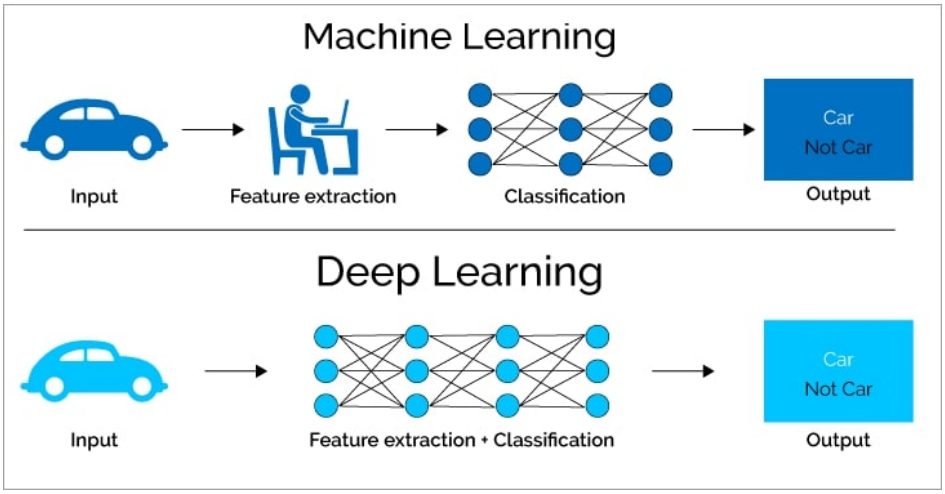

Figure 8: Deep learning vs Machine learning

learning as they lack these functionalities. While some shallow ML algorithms are inherently interpretable by humans and are considered as white boxes, the decision-making process of most advanced ML algorithms is inherently untraceable unless explicitly explained, making them black boxes.

Deep learning (DL) excels in domains with large and high-dimensional datasets, making deep neural networks outperform shallow ML algorithms in tasks involving text, image, video, speech, and audio data processing. However, when dealing with low-dimensional data inputs, especially in scenarios with limited training data availability, shallow ML approaches can still produce superior result, which are often more interpretable compared to those generated by deep neural networks [16].

4.3 Applications and challenges of AI in Internet of thing

Several case studies have showcased the successful integration of Internet of Things (IoT) and machine learning technologies in various fields including smart cities such as improving the efficiency of urban services, and enhancing citizens' overall quality of life. Deep learning algorithms combined with video analysis have been identified as practical applications in smart cities. In one study, researchers developed an IoT system based on deep learning for remote monitoring and early detection of health issues in real time. The system exhibited impressive accuracy in identifying heart conditions, achieving a remarkable accuracy rate of 0.982 [17, 18].

However, deploying deep learning methodologies within IoT frameworks presents challenges and limitations. These challenges can be broadly categorized into

ethical and privacy implications, scalability and resource constraints, and the ongoing need for research and development. Integrating deep learning into IoT has significantly advanced security and efficiency in surveillance applications. IoT devices with deep learning capabilities, such as convolutional neural networks, can effectively analyze video streams to detect threats and anomalies, improving real-time monitoring and predictive insights.

Nevertheless, these applications face significant challenges, such as the need for extensive datasets and substantial computational power. The scalability of deep learning models in the IoT poses a significant concern. IoT devices often have limited computational resources, making it challenging to deploy complex deep-learning models that require substantial data processing capabilities. It is crucial to optimize these models for deployment on resource-constrained devices, necessitating innovative solutions that strike a balance between the computational demands of deep learning algorithms and the inherent limitations of IoT hardware. [19]

II. TinyML and AI on the Edge

1. Edge computing and AIoT overview

In recent years, edge computing has emerged as a promising technology with significant potential in various fields. It offers advantages such as reduced latency and cost savings. Unlike traditional cloud computing, edge computing enables data processing at the edge of the network. This approach brings data computing closer to the data source, greatly benefiting the development of time-sensitive applications. Additionally, by performing processing locally, edge computing reduces network traffic and minimizes data transmission, resulting in substantial cost savings. [20]

From the advantages of edge computing, integrating edge devices with IoT ecosystems can facilitate the shift of computation from the cloud to the network edge through collaboration among sensors, edge devices, and cloud facilities. However, edge hardware has limited resources and is platform-dependent, which restricts its ability to support advanced and complex services such as machine learning applications. It obstructs the development of a standardized machine learning framework for all IoT-edge devices.

Furthermore, while existing embedded processors can handle generic sensor data processing and web-based applications, machine learning applications rely on sophisticated hardware chips such as graphics processing units (GPUs). These chips demand significant power and memory capacity to execute deep neural network models. Thus, the current landscape challenges achieving the envisioned "cloud-to-embedded" aspect. [21]

1.1 TinyML overview

Since 2019, a new technology called TinyML has been addressing the challenges of designing power-efficient deep learning models (in the milliwatt range and below) to be integrated into embedded systems like AIoT/IIoT/IoT devices. TinyML is defined as follows: "machine learning aware architectures, frameworks, techniques, tools, and approaches which are capable of performing on-device analytics for a variety of sensing modalities (vision, audio, speech, motion, chemical, physical, textual, cognitive) at mW (or below) power range setting, while targeting predominately battery-operated embedded edge devices suitable for implementation at large scale use cases preferable in the IoT or wireless sensor network domain" (TinyML, 2021a). Thus, TinyML can be envisaged

as the composition of three key elements (i) software, (ii) hardware, and (iii) algorithms. [25]

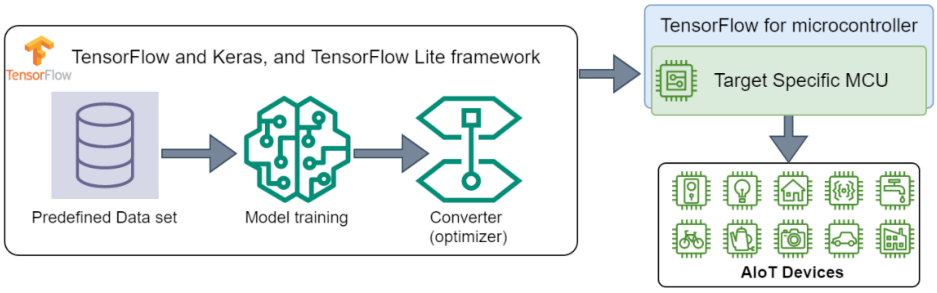

Figure 9: TensorFlow lite for microcontrollers

The above figure illustrates the workflow for developing deep learning algorithms. First, the dataset is created based on the chosen neural network model (CNN, RNN, etc.). In some cases, dataset preprocessing may be necessary to optimize training and improve performance. Training is typically conducted on a workstation or supercomputer server using frameworks like TensorFlow or PyTorch. The trained model's precision is usually float32, and its parameter memory footprint is significant (larger than the available flash and SRAM on a microcontroller unit).

TensorFlow Lite, within the TensorFlow framework, provides a converter to optimize the trained model by reducing the memory footprint and computational power requirement (using int8 instead of float32). [22] The TensorFlow Lite framework converter takes the trained model (saved in the xx.h5 file) and produces three optimized models: int8 quantization, integer dynamic quantization, and float16. One of these files, such as the int8 quantization tflite file, can be utilized in other tools like the X-CUBE-AI [23] extension pack of STM32CubeMx [24]. These tools can generate C codes that can be deployed and run on microcontrollers [22]

2. TinyML framework and software

TinyML requires several hardware specifications, libraries, and software platforms to leverage predictions. We only focus on software platforms and frameworks for this project's scope. TinyML frameworks and softwares are being developed to support the deployment of machine learning models on embedded devices. Some of examples are as follows: [23]

- TensorFlow Lite for Microcontrollers (TFLite Micro): TensorFlow Lite is a popular ML framework developed by Google. TFLite Micro is a specialized version of TensorFlow Lite designed for microcontrollers and other small devices. It provides a set of tools and libraries for training and deploying ML models on edge devices with limited resources. TFLite Micro supports various hardware platforms and offers optimizations for memory and computational efficiency.

- Edge Impulse: Edge Impulse is an end-to-end platform for building and deploying TinyML applications. It provides a comprehensive workflow for collecting, preprocessing, training, and deploying ML models on edge devices. Edge Impulse supports a wide range of sensors and development boards, making it accessible for developers without extensive ML expertise. It also offers integration with popular ML frameworks like TensorFlow and Keras.

- uTensor: uTensor is an open-source ML inference library specifically designed for microcontrollers. It allows developers to deploy trained ML models on resource-constrained devices with minimal code overhead. uTensor provides a C++ API for loading and executing models and supports quantization techniques to reduce model size and improve inference speed. It also offers compatibility with TensorFlow and supports various microcontroller platforms.

- TVM (pronounced "micro TVM") is a framework that extends the TVM (Tensor Virtual Machine) deep learning compiler and runtime to support edge devices. TVM supports quantization techniques, such as reduced-precision arithmetic, to reduce the memory footprint and improve inference speed while maintaining acceptable accuracy levels. It also offers support for other features, such as hardware abstraction (to support different microcontroller platforms) and model compression (to reduce the size of the model).

3. Conclusion

The Edge-IoT ecosystem holds immense potential in leveraging intelligent decision-making capabilities at the network edge. The integration of machine learning into resource-limited embedded devices has emerged as a crucial requirement for future-oriented applications. Therefore, a thorough exploration of the

TinyML paradigm is essential to advance the current landscape of edge-aware machine learning.

III. Project: Speech emotion recognition for edge devices

1. Motivation

Human vocal language is an interesting topic that has inspired researchers to explore speech communication with machines for a long time. Various applications, such as smartphone assistants, speech-to-text converters, and voice-operated machines, have been developed in this domain. However, catching up with human speech is not enough. By enabling machines to recognize and understand human emotions, we can enhance their ability to comprehend us more effectively.

Speech emotion recognition (SER) is a subfield of speech recognition that aims to classify a human speaker's emotional state. It is vital in this domain as it empowers machines to identify and respond to human emotions. Its applications span multiple fields, including safety, entertainment, and biomedicine. An exemplary use case is automatic call assistance systems, where understanding customer emotions is vital for providing appropriate service. By discerning emotions, such systems can redirect calls from angry or dissatisfied customers to human agents, ensuring personalized assistance. In web-based movies and computer tutorials, SER can enhance the viewer's experience by adapting content based on their emotional state [25]

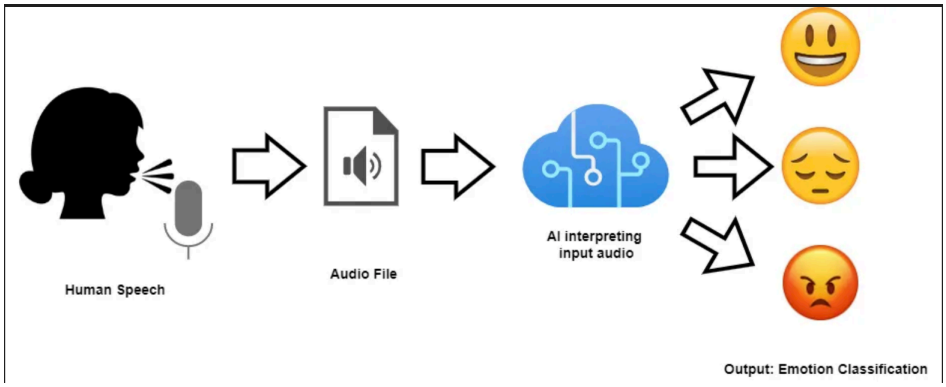

Figure 10: Speech recognition system

In the automotive industry, SER can serve as a safety feature. By monitoring the driver's emotional state, the system can take proactive measures if it detects signs of mental disturbance, ensuring a safer driving experience. Additionally, SER holds potential as a diagnostic tool in therapists' hands. Analyzing a

patient's speech patterns and emotional expressions can provide valuable insights for assessment and treatment.

In conclusion, integrating speech emotion recognition into machine communication opens up a world of possibilities. By recognizing and interpreting human emotions, machines can better comprehend and respond to our needs. Whether in customer service, entertainment, automotive safety, or healthcare, SER has the potential to elevate our interactions with machines to a more intuitive and emotionally aware level.

2. Project overview

2.1 Introduction

In this project, I aim to build a lightweight speech emotion recognition model and evaluate its expected performance and power consumption using Mel Frequency Cepstral Coefficients (MFCC) and Long Short-Term Memory (LSTM) neural networks. Besides utilizing the TensorFlow Lite framework to compress the model to reduce its memory footprint and computational power requirement, offering the ability to deploy the model on mobiles devices, IoT and wearable devices. The dataset used for training and evaluation comprises the RAVDESS Emotional speech audio, Toronto emotional speech set (TESS), and Surrey Audio-Visual Expressed Emotion (SAVEE) [1].

2.2 Dataset description

The RAVDESS dataset (created by Ryerson University's Sensory Communication Group), The TESS dataset (developed by the University of Toronto) and The SAVEE dataset (developed by the University of Surrey) are comprehensive and valuable collections of emotional speech and song recordings. It consists of audio samples performed by professional actors (male and female artist) who were instructed to portray various emotions, including neutral, happy, sad, angry, fearful, disgust, and surprised. These datasets serve as valuable resources for training and evaluating models to accurately recognize and classify emotions from speech. [1]

2.3 Preliminary Methodology

The method employed in this project is based on the work of Sheetal U. Bhandari and Harshawardhan S. Kumbhar. They selected the Long Short-Term Memory (LSTM) architecture and Mel-frequency cepstral coefficients (MFCCs)

for the speech emotion recognition (SER) task due to their proven effectiveness in learning from sequences. [26]

For feature extraction, I utilize Mel-Frequency Cepstral Coefficient (MFCC) as it is a popular feature extraction technique for speech recognition [25]. MFCC effectively reduces the computational complexity of the system while enhancing its ability to extract relevant features, including parameters like pitch and energy [3]. By converting the frequency information of the speech signal into a smaller set of coefficients, MFCC simplifies the feature extraction process [3]. It represents the short-term power spectrum of sound through a linear cosine transform of a logarithmic power spectrum on a non-linear Mel scale of frequency. [27]

LSTM is a type of recurrent neural network (RNN) that can learn long-term dependencies between time steps of sequence data. It is a popular choice for speech recognition tasks due to its ability to learn from sequences of data since in emotion detection, it is crucial to take into account the interdependence of each section with the preceding one [26].

Once the standard training and validation pipeline is completed, I convert the model into TensorFlow Lite (TFLite) format, optimizing its size through quantization. Furthermore, I compare the inference time of the quantized model with that of the original model, evaluating the trade-off between model size and inference speed.

IV. Project outcome

1. Preprocessing

The three datasets have the same format audio file with the same label categories. Thus, I combined them into one dataframe for simplicity.

Human emotion can be represented in terms of valence (pleasantness) and intensity [27]. For this implementation scenario, "Fear", "Disgusted", "Angry" and "Sad" are merged into a category as "Unpleasant" emotions, while "Happy", and "Surprised" are merged into another category as "Pleasant" emotions. Finally, we would have 3 classes: "Pleasant", "Unpleasant", "Neutral".

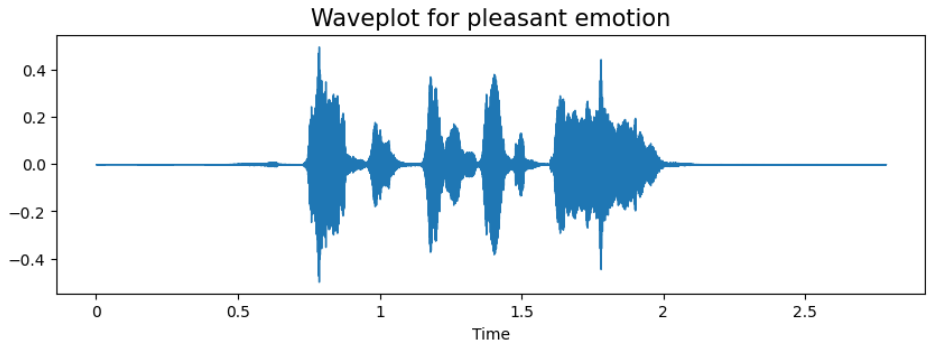

Figure 11: waveform of pleasant emotion

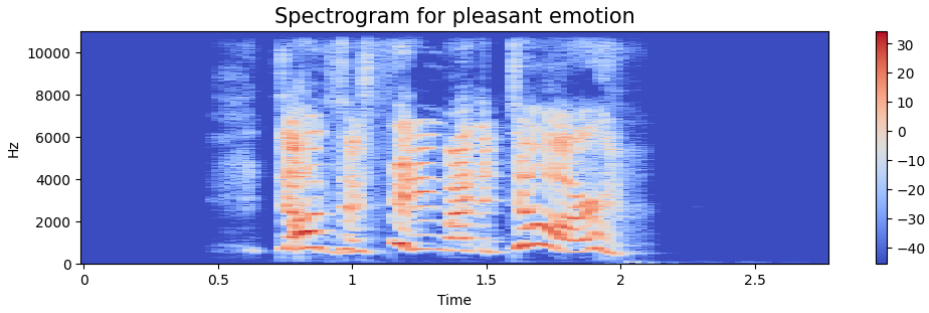

Figure 12: spectrogram of pleasant emotion

The detailed steps of preprocessing are as follows:

Emotion Label Encoding: The emotion labels are encoded to facilitate model training. This encoding step assigns numerical values to each unique emotion label.

- Feature Extraction using MFCC: The audio data is transformed into Mel Frequency Cepstral Coefficients (MFCC) features. MFCC is a commonly used feature representation for speech and audio processing tasks. The outcome is a features matrix with shape (47, 13).

- Data Augmentation with Audio Transformations: The audio data is augmented by applying various audio transformations such as noise addition, time stretching, and pitch shifting. This augmentation technique increases the diversity of the training data and helps the model generalize better.

- Data Storage in CSV: The preprocessed data, including the encoded emotion labels and extracted MFCC features, is stored in a CSV file for easy access and further analysis.

- Data Split into Training, Validation, and Testing Sets: The preprocessed data is split into training, validation, and testing sets. This division allows for model training using the training set, hyperparameter tuning using the validation set, and final evaluation using the testing set.

2. Model training evaluation

The model's structure is defined by adding layers sequentially. Two LSTM layers are incorporated, contributing to the model's ability to capture long-term dependencies. The first LSTM layer possesses 128 units and receives input with the specified $\text{input}_shape. Additionally, it returns sequences to allow for subsequent processing. The$

To introduce non-linearity and increase model capacity, a dense layer with 64 units and a rectified linear activation function (ReLU) is included. This layer enables the model to learn complex relationships and extract meaningful features from the input data. To mitigate overfitting, a dropout layer with a dropout rate of 0.3 is inserted, randomly disabling connections between neurons during training.

For multi-class classification, a dense layer with 3 units and a softmax activation function is appended as the output layer. The softmax activation ensures the production of class probabilities, facilitating the assignment of the input to the most suitable class.

Upon defining the model, the code proceeds to compile it. An Adam optimizer, a widely used optimization algorithm, is employed with a learning rate of 0.001. The loss function chosen is sparse categorical cross-entropy, which

is appropriate for multi-class classification tasks. The model's performance is evaluated using the accuracy metric.

After training, the model was evaluated with an accuracy of 76.15% on the test set. The confusion matrix shows that the model performs pretty fair for the 3 classes "Pleasant", "Unpleasant" and "Neutral", achieving precision of 77%, 77% and 71%, respectively.

This can be explained by the fact that the classes are merged from 6 classes to 3 classes, which makes the model easier to learn and predict. However, the accuracy is not high since I rescale the data samples, so the dataset is not large enough to train a good model.

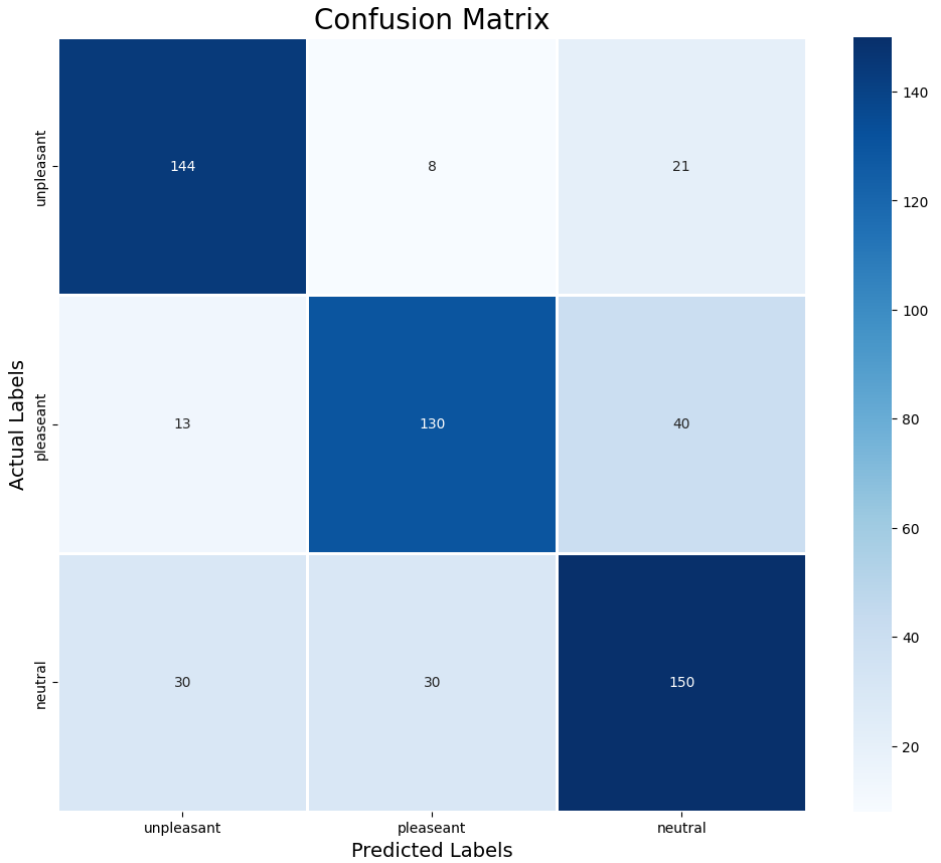

Figure 13: Confusion matrix

3. Model compression

The objective of employing TinyML techniques is to minimize both the model size and inference time, while preserving a comparable level of accuracy. Finally,

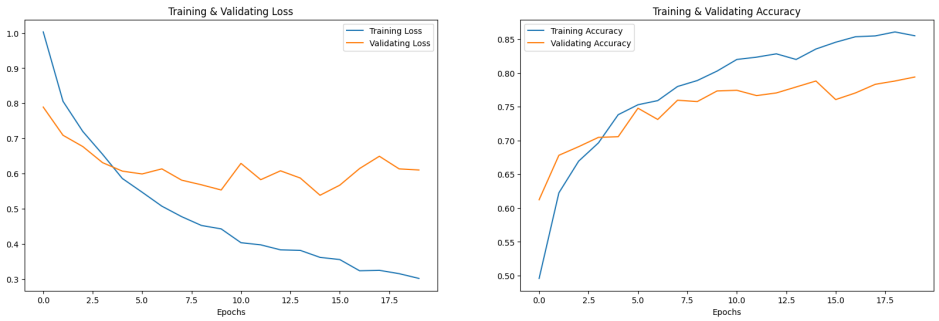

Figure 14: Training & validationresult

I conduct a comparison between the inference time and accuracy of the original TensorFlow model and the TensorFlow Lite model. In this section, our objects are as follow:

- Convert the model to tf lite model.
- Compare the size and the accuracy of models (before and after compress)
- Save the model in the format that deployable into suitable MCU

Reduce phase	Size (byte)	Accuracy	Inference Time
original model	1568912	74.911660%	16.890739s
tflite model	150032	74.558304%	0.413304

Table 1: Comparation table

In conclusion, the TensorFlow Lite (TFLite) model demonstrates comparable accuracy to the original model while exhibiting notable improvements in model size and inference time. Despite its smaller size, the TFLite model maintains similar accuracy levels, making it an efficient and practical choice for deployment on resource-constrained devices. Additionally, the reduced inference time of the TFLite model allows for faster predictions, enhancing real-time performance in various applications. These benefits make TFLite an appealing solution for scenarios where optimizing model size and inference speed are critical considerations.

References.

[1] Dataset sources: - Livingstone SR, Russo FA (2018) The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE 13(5): e0196391 https://doi.org/10.1371/journal.pone.0196391

-Toronto emotional speech set (TESS) Collection https://tspace.library.utoronto.ca/handle/1807/24487

-Surrey Audio-Visual Expressed Emotion (SAVEE) Database http://kahlan.eps.surrey.ac.uk/savee/Database.html

[2] Hastie, T., Tibshirani, R., Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science Business Media.

[3] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

[4] Zhu, X., Goldberg, A. B., Nowak, R. (2009). Introduction to semi-supervised learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 3(1), 1-130.

[5] Sutton, R. S., Barto, A. G. (2018). Reinforcement learning: An introduction. MIT Press.

[6] Chen, H., Zhang, J., Hsu, W. (2017). Data-driven optimization for ride-sharing systems: A reinforcement learning approach. Transportation Research Part C: Emerging Technologies, 80, 48-63.

[7] Li, Y., Zhu, J., Gong, Y. (2018). Deep reinforcement learning for person re-identification in video surveillance. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops (pp. 0-0).

[8] Pang, C., Li, Z. (2018). Deep learning based large scale visual recommendation and search for E-commerce. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI) (pp. 0-0).

[9] Domingos, P. (2012). A few useful things to know about machine learning. Communications of the ACM, 55(10), 78-87.

[10] Atzori, L., Iera, A., Morabito, G. (2010). The Internet of Things: A survey. Computer Networks, 54(15), 2787-2805.

[11] Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645-1660.

[12] Elgazzar, Khalid and Khalil, Haytham and Alghamdi, Taghreed and Badr, Ahmed and Abdelkader, Ghadeer and Elewah, Abdelrahman and Buyya, Rajkumar. (2022). Revisiting the internet of things: New trends, opportunities and grand challenges
http://dx.doi.org/10.3389/friot.2022.1073780

[13] Haykin, S. (1994). Neural networks: a comprehensive foundation. Prentice Hall.

[14] Hagan, M. T., Demuth, H. B., Beale, M. H. (2014). Neural network design. PWS Publishing Company.

[15] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MIT press.

[16] Janiesch, C., Zschech, P. Heinrich, K. Machine learning and deep learning. Electron Markets 31, 685–695 (2021).

[17] Ullah, A.; Anwar, S.M.; Li, J.; Nadeem, L.; Mahmood, T.; Rehman, A.; Saba, T. Smart cities: The role of Internet of Things and machine learning in realizing a data-centric smart environment. Complex Intell. Syst. 2023, 1–31

[18] Islam, M.R.; Kabir, M.M.; Mridha, M.F.; Alfarhood, S.; Safran, M.; Che, D. Deep Learning-Based IoT System for Remote Monitoring and Early Detection of Health Issues in Real-Time. Sensors 2023, 23, 5204.

[19] Elhanashi, A.; Dini, P.; Saponara, S.; Zheng, Q. Integration of Deep Learning into the IoT: A Survey of Techniques and Challenges for Real-World Applications. Electronics 2023, 12, 4925.

[20] W. Yu et al., "A Survey on the Edge Computing for the Internet of Things," in IEEE Access, vol. 6, pp. 6900-6919, 2018, doi: 10.1109/ACCESS.2017.2778504.

[21] F. Wang, M. Zhang, X. Wang, X. Ma and J. Liu, "Deep Learning for Edge Computing Applications: A State-of-the-Art Survey," in IEEE Access, vol. 8, pp. 58322-58336, 2020, doi: 10.1109/ACCESS.2020.2982411.

[22] Rong, G., Xu, Y., Tong, X., Fan, H. (2021). An edge-cloud collaborative computing platform for building AIoT applications efficiently. Journal of Cloud Computing, 10(1), 1-14. https://doi.org/10.1186/s13677-021-00250-w

[23] Ray, P. P. (2022). A review on TinyML: State-of-the-art and prospects. Journal of King Saud University - Computer and Information Sciences, 34(4), 1595-1623. https://doi.org/10.1016/j.jksuci.2021.11.019

[24] Hou, K. M., Diao, X., Shi, H., Ding, H., Zhou, H., De Vaulx, C. (2022). Trends and Challenges in AIoT/IIoT/IoT Implementation. Sensors, 23(11), 5074. https://doi.org/10.3390/s23115074

[25] ST. X-CUBE-AI Artificial Intelligence (AI) Software Expansion for STM32Cube; STMicroelectronics: Geneva, Switzerland, 2021; https://www.st.com/resource/en/data_brief/x-cube-ai.pdf

[26] ST. STM32CubeMX for STM32 Configuration and Initialization C Code Generation. June 2022 UM1718 Rev 38 https://www.st.com/resource/en/data_brief/stm32cubemx.pdf

[27] TinyML talk Felix Johnny and Fredrik Knutsson - Arm Sweden Area Group –February 8, 2021 https://cms.tinyml.org/wp-content/uploads/emea2021/tinyML_Talks_Felix_Johnny_Thomasmathibalan_and_Fredrik_Knutsson_210208.pdf

[28] Moataz El Ayadi, Mohamed S. Kamel, Fakhri Karray, Survey on speech emotion recognition: Features, classification schemes, and databases, Pattern Recognition, Vol. 44, Issue 3, 2011.Pages 572-587

[29] Kumbhar, Harshawardhan S., and Sheetal U. Bhandari. "Speech emotion recognition using MFCC features and LSTM network." 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA). IEEE, 2019

[30] Russell, James A. "A circumplex model of affect." Journal of personality and social psychology 39.6 (1980): 1161.

[31] Supriya B.Jagtap, Dr.K.R.Desai, Ms. J. K. Patil " A Survey on Speech Emotion Recognition Using MFCC and Different classifier" , 8th national conference on emerging trends in engg and technology, 10th march 2018

[32] github repository: `https://github.com/thuannt-se/academic_work/tree/main/internship1`