CISC 499: Undergraduate Projects
2020

Project 1: IoT-Enabled Cyber-Physical Systems

Scope
The migration towards the era of smart cities is heavily dependent on connecting physical things to the internet (a.k.a. the internet of things, IoT) and building environments that enable users to actively interact with these physical things (e.g., monitoring/tracking, control, reservation, etc.) through these connections. Systems with this description are usually known as cyber-physical systems (CPSs).

Project Description
In this project, interested groups of one or two students will develop a **multi-purpose** IoT-enabled CPS (see examples below). Any system must consist of more than two sides, one of which is a user app/website and the others are "things". The user side should be able to establish connections with the thing sides, read monitoring information from them, and send control commands and/or requests to them. The thing sides will be required to maintain the established connections with the user side, send monitoring information, and receive and execute a variety of controls and/or requests from the user side. Applications of the system are left open to students' innovation, and I will specify requirements for each system based on its chosen applications. Since some CPSs cannot be fully deployed in real life for beyond-student reasons (e.g., examples below), a full functional system should still be built and tested in an emulated environment for demonstration (details to be discussed with me as per the system's idea and applications).

An example of a multi-purpose CPS is a smart home environment with multiple controls and interactions (control light/heating, open/close door locks remotely, send and receive intercom-like voice messages, etc.). Another example is a smart bus interaction system, enabling users to monitor a bus position, send pickup requests to the bus at any given station, provide options to pay ride fees online, receive alerts to start moving toward the station when necessary, and transfer pickup requests to other buses if late.

Required Expertise
In addition to strong programming skills, the project requires basic knowledge on app/web development, interfacing with hardware, and connection establishment between end devices.

Deliverables:
1. Codes for the entire system
2. Prototype of a fully functional system in a real or emulated environment
3. Project report
4. Video recording illustrating all the functions of the prototype

Learning Outcomes
Learn how to select, build, and program CPS applications, and interface them with physical hardware.

Project 2: Federated Learning on Wireless Edge Devices

Scope

Federated learning is a new distributed learning paradigm, recently developed by Google, which enables edge devices to collaboratively learn from their local data without sharing this data to a central node. Federated learning thus decouples the ability to perform machine learning from the need and burden of transferring and storing this data centrally.

Project Description

In this project, a federated learning task is to be fully implemented on a group of (possibly heterogeneous) wireless edge devices (e.g., laptops are fine but preferably on smart phones and/or wirelessly connected raspberry pies, etc.). The project consists of two sub-projects. The first sub-project will be concerned with building local training processes on a set of wireless edge devices using a neural network (e.g., CNN). The training process on each device will be triggered by a central orchestrator to:

1. Perform training rounds of its neural network using samples from the devices' stored data sets
2. Periodically send the trained local model parameters to the orchestrator.
3. Replace the local parameters with the globally updated ones once received from the orchestrator.

The second sub-project will focus on developing an orchestrator application that:

1. Initiates the training process on each of the edge devices.
2. Determines the number of samples employed by each of them for training.
3. Calculates the global model parameters from the periodically sent local ones by all devices.
4. Sends the updated global model parameters to the devices for use in their subsequent training rounds.

Required Expertise

In addition to strong programming skills, the project requires basic knowledge on building and training neural networks, and establishing wireless connections between edge nodes.

Deliverables:

1. Codes for the device training process and orchestrator application
2. Prototype of a fully deployed federated learning system on a group of wireless edge devices.
3. Project report
4. Video recording of an annotated run of a federated learning task.

Learning Outcomes

Learn about distributed federated learning approaches and networking/computing resource optimization for distributed computing and learning.

Project 3: Intention Prediction in Vehicular Environments

Scope
Some of the accidents recently endured by autonomous vehicle (AV) prototypes were mainly caused by their inability to predict the behavior of other vehicles and pedestrians in near future times. This shortcoming will affect not only the safety of AVs and pedestrians but also the efficiency of AVs driving decisions. For example, it can result in unnecessary stops or slowdowns by the AV.

Project Description
In this project, a prediction engine for vehicle (and possibly pedestrian) intentions will be both developed using machine learning techniques and tested through simulations. The project consists of two sub-projects. The first sub-project will be concerned with training a machine learning model to predict the intentions of objects (e.g., other vehicles, pedestrians) observed by the AVs for up to few seconds in the near-future. Several driving sequences of AV sensory data (e.g., KITTI benchmark suite) can be used in training this model. The second sub-project will focus on building a simulation environment to test both the accuracy of the trained model in predicting objects intentions and the efficiency of the taken driving decisions accordingly. Several simulators can be used for this task such as Carla, VISSIM, or INTEGRATION.

Required Expertise
In addition to strong programming skills, the project requires basic knowledge on building and training neural networks, and implementing prediction techniques.

Deliverables:
1. Codes for the behavior prediction training process
2. The parameters of the trained model
3. Testing results in different traffic environments and weather conditions
4. Project report
5. Video recording of a simulation showing the predicted behavior of detected objects

Learning Outcomes
Learn how to handle driving sequences of AV sensory data, detect objects from this data, train learning models on predicting the future behavior of these objects, and test learning models in simulation environments.