EE P 596 - TinyML - Assignment 2

Total Points: 100
Spring Quarter, 2024
Department of Electrical and Computer Engineering
University of Washington, Seattle, WA 98195

Due: 11:59 pm (PST) on May 19 (Sun), 2024 via Canvas

Note:

- This homework contains both programming questions (marked as [**Pro**]) which are required to write Python codes and discussion questions (marked as [**Dis**]) which are required to write answers and provide detailed explanations for your answers.
- You can use and modify the Python functions and codes provided in the Labs or file section of the EEP 596 canvas page when coding the [**Pro**] questions.
- Your homework submissions must be uploaded to ***Canvas*** in the following formats: (i) a `.ipynb` file containing all Python code for the programming questions. If you have multiple `.ipynb` files, submit a `.zip` file with all the Jupyter notebooks. (ii) a `.pdf` file with either scanned handwritten or typed responses to discussion questions, and the final results for the programming questions. (iii) C++ files (`.ino` and `.h` files) which you compiled and uploaded to the Arduino Nano BLE.
- Name of your submission files should follow the following format:
"*#_$_EEP596_HW2.ipynb*" where "#" and "$" should be replaced with your first name and last name, respectively. Use the same format for the `.pdf` writeup file.

In this assignment, you will make use of the following dataset and files. You will find these files when you unzip the **EE596_HW2_Files.zip**:

- **Network Anomaly Dataset:** This dataset consists of 125,973 data samples, each with 42 features corresponding to normal and attack network connections. The **network_anomaly_data.txt** file included with this homework assignment provides this dataset.
- **FirstName_LastName_EEP596_HW2.ipynb:** You will use this Python notebook file to answer Questions 2, 3, 4, and 5. When submitting your homework, **please replace FirstName_LastName with your first and last name**.
- **c_writer.py:** This script will be called in **FirstName_LastName_EEP596_HW2.ipynb** to automatically convert your tflite model to C and create the header file (**network_model.h**) for Arduino Nano BLE deployment.
- **network_data.ino:** You will use this Arduino Sketch file to answer Question 6. You will find this file inside the **network_data** folder.

1. **[Pro]** (Data Collection for TinyML Lab 6, "MagicWand," 10 points) The deadline was May 1st.

2. **[Pro]** (Data Preprocessing, 5 pts x 3 tasks = 15 pts)
   Complete the following tasks in the provided Python notebook (`.ipynb`) file:

   (a) Drop the *'land', 'urgent', 'numfailedlogins', 'numoutboundcmds'* columns from the dataframe *data*.

   (b) Change any label that is not named **normal** to **attack** in the {'attack'} column of the dataframe *data*.

   (c) Use **LabelEncoder()** function from the **sklearn.preprocessing** library to convert non-numerical attributes in the *'protocoltype', 'service', 'flag', 'attack'* columns of the dataframe *data* to numerical values.

3. **[Pro]** (Dimensionality Reduction for Visualization, 5 pts x 3 tasks = 15 pts)
   Complete the following tasks in the provided Python notebook (`.ipynb`) file:

   (a) Use **TSNE** from the **sklearn.manifold** library to visualize the data in the **test set (X_test)** in 2D. In your figure, use color **red** to mark ***attack*** data points and color **blue** to mark ***normal*** data points.

   (b) Use **PCA** from the **sklearn.decomposition** library to visualize the data in the **test set (X_test)** in 2D. In your figure, use color **red** to mark ***attack*** data points and color **blue** to mark ***normal*** data points.

   (c) Use **KernelPCA** from the **sklearn.decomposition** library to visualize the data in the **test set (X_test)** in 2D. Use **radial basis function (rbf)** as the kernel. In your figure, use color **red** to mark ***attack*** data points and color **blue** to mark ***normal*** data points.

4. **[Pro]** (Implementing a DNN on the dataset, 5 pts x 3 tasks = 15 pts)
   Complete the following tasks in the provided Python notebook (`.ipynb`) file:

   (a) Implement a **deep neural network (DNN)** on the **Network Anomaly Dataset**. Ensure to include **two neurons** and **softmax activation** in the output layer of your DNN.

   (b) Compile and train your DNN model on the **training set (X_train)**. Denote the trained model as **base_model**.

   (c) Evaluate the **base_model** on the **test set (X_test)** using **classification_report** and **confusion_matrix** from the **sklearn.metrics** library. **Report these numbers in your** `.pdf` **writeup file using screenshots**.

5. **[Pro]** (Implementing Quantized Model, 7.5 pts x 2 tasks = 15 pts)
   Complete the following tasks in the provided Python notebook (`.ipynb`) file:

   (a) Implement **Dynamic Range Quantization** on the **base_model**. Designate the resulting quantized ML model as **tflite_quant_model**.

   (b) Evaluate the **tflite_quant_model** on the **test set (X_test)** using **classification_report** and **confusion_matrix** from the **sklearn.metrics** library. **Report these numbers in your** `.pdf` **writeup file using screenshots**.

6. **[Pro]** (Deploying the Quantized Model, 5 pts x 4 tasks = 20 pts)
   Complete the following tasks in the provided Arduino sketch (`.ino`) file:

   (a) Implement code to obtain the **prediction** from the **output tensor** and determine the **predicted class label**.

   (b) Implement code to output **Sample #, Predicted Class**, and **Actual Class** for each sample to the serial monitor using **Serial.print** function.

   (c) Add the **network_model.h** file, generated at the end of the Python notebook, to the **network_data folder** where **network_data.ino** is kept. Upload the **network_data.ino** to the Arduino Nano BLE. **Obtain screenshots of the printed Serial Monitor outputs for five samples and report these in your `.pdf` writeup.**

   (d) Use the code at the end of the Python notebook to obtain **10 features and actual labels of the test set excluding the first five samples**. Add these new data at the appropriate places in the **network_data.ino**. Upload the **network_data.ino** to the Arduino Nano BLE. **Obtain screenshots of the printed Serial Monitor outputs for these ten samples and report these in your `.pdf` writeup.**

7. **[Dis]** (Open-ended Discussion Questions, 5 pts x 2 tasks = 10 pts)

   (a) In this homework, we hard code the test samples that need to be inferred using the TinyML model deployed on the Arduino Nano BLE. Is it possible to **stream network data** to the Arduino Nano BLE? If your answer is "YES," describe what **functionality within the Arduino Nano BLE** allows for this task.

   (b) If we have high-dimensional nonlinear network data, which dimensionality reduction method among **TSNE, PCA, and KernelPCA is most suitable to use and why**? What **parameters** need to be passed to the Arduino Nano BLE to accomplish this?