

HUMAN ACTIVITY RECOGNITION

BY TANARAT SAEHIA 663380035-4 & PHINATKANIN PHISITKUL 663380528-1

INTRODUCTION

เพื่อให้สามารถกำกิจกรรมต่างๆ ตามจำนวนที่แพทแบงค์ทำ
เราจึงพัฒนา Application กี่ช่วยในการติดตามกิจวัตร
ประจำวันว่ากำลังทำอะไร เช่น เดิน, เดินขึ้นบันได, เดินลง
บันได, นั่ง และลุก เป็นต้น เพื่อเป็นผู้ช่วยจำว่าในวันนี้ๆ เรา
กำกิจกรรมใด ไปจำนวนเท่าไหร่แล้วบ้าง

PROCESS

01

Collect Data

02

Data Preprocessing

03

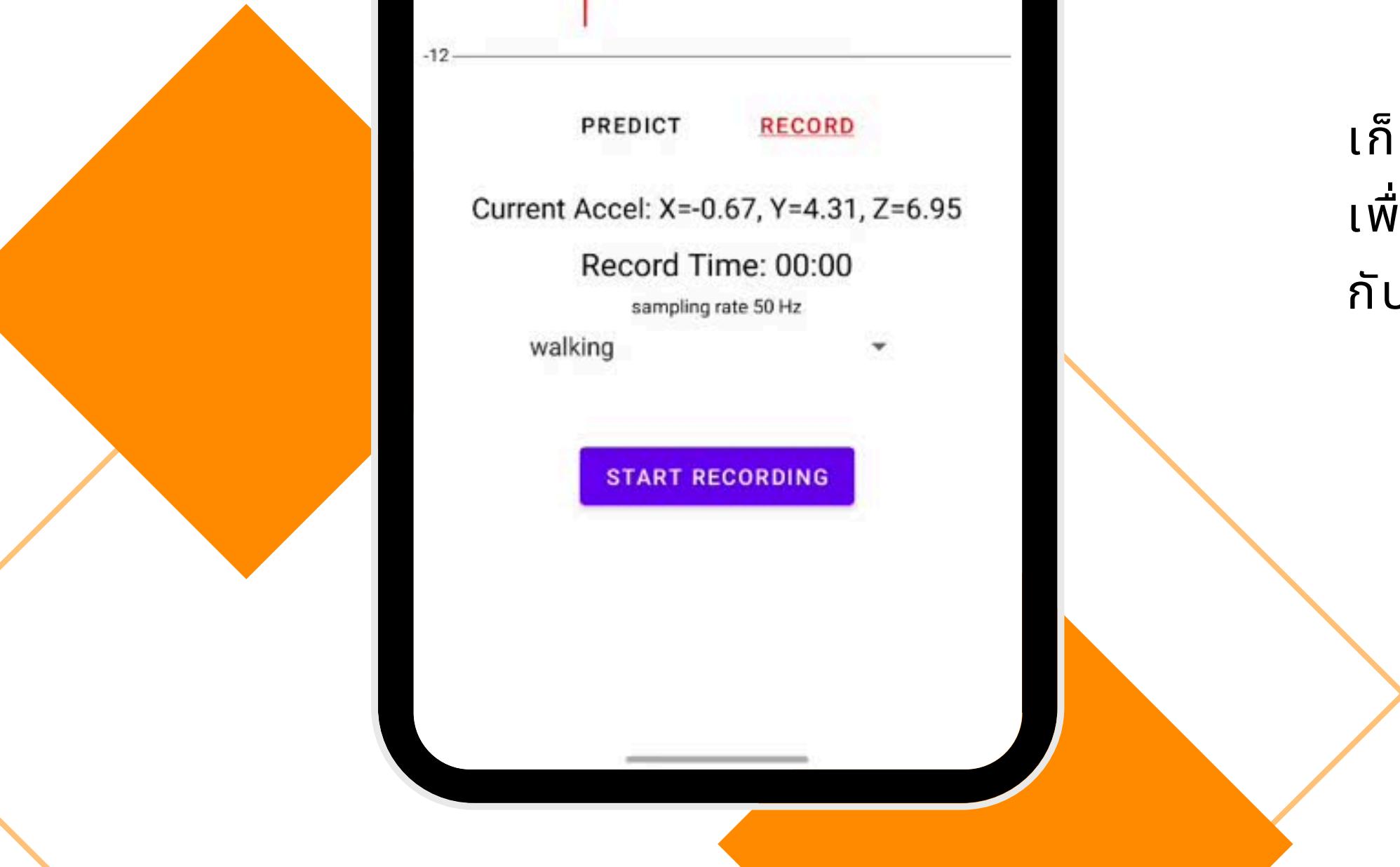
Model Training

04

Evaluate Model

05

Deploy



COLLECT DATA

เก็บข้อมูลโดยสร้าง Android Application
เพื่ออ่าน Accelerometer โดยใช้ Android Studio
กับภาษา Kotlin

DATA PREPROCESSING



COMBINE SIGNAL

รวมสัญญาณจากทั้ง 3 แกน (x,y,z) โดยใช้ Magnitude ในการรวม

Pros : ลด cost การคำนวณใน process อีก

Cons : สูญเสียข้อมูลทิศทาง

$$\text{Magnitude} = \sqrt{x^2 + y^2 + z^2}$$

ภาพสมการ magnitude



REMOVE NOISE

โดยทั่วไป noise ใน accelerometer เป็น electronic noise (high-frequency noise)

Human activity โดยส่วนมากจะมีความถี่ไม่เกินช่วง 10-20 Hz

วิธีจัดการ : ใช้ high-pass filter โดย cut-off frequency = 15 Hz

DATA PREPROCESSING



REMOVE GRAVITY

gravity ใน context ของ accelerometer คือ DC component(0 Hz)

หรือ ถ้าความถี่ต่ำมากๆ ไม่เกิน 0.5 Hz

วิธีจัดการ : ใช้ low-pass filter โดย cut-off frequency = 0.5 Hz



WINDOWING

เพื่อให้โน้ตเดลได้เห็นความสำคัญของข้อมูลมากขึ้น จึงจำเป็นต้องให้ window size มีขนาดที่ใหญ่คือ

แบ่งข้อมูลไปประมวลผลทุกๆ 5 วินาที โดยที่จะให้แต่ละข้อมูล overlap กันที่ 50%

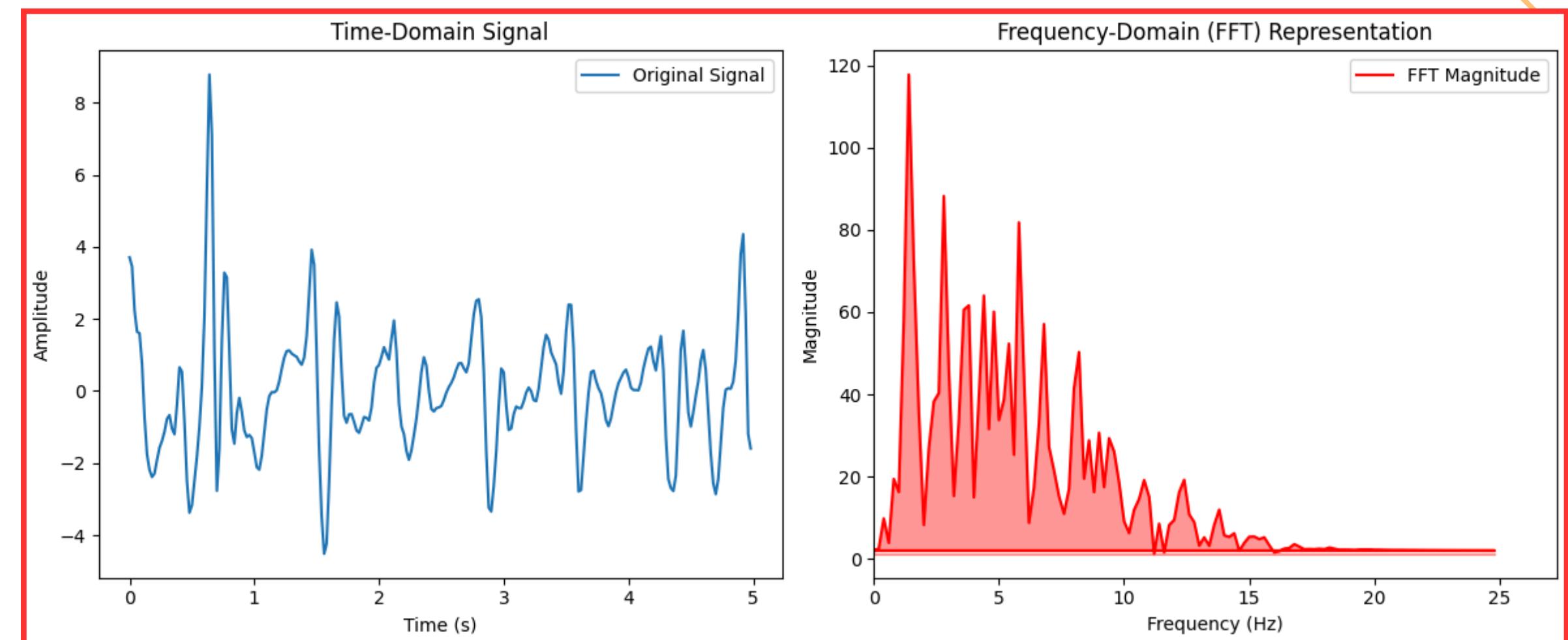
เนื่องจากมีการ overlap ซึ่งทำให้การประมวลผลเรียงกันทุก 2.5 วินาที

DATA PREPROCESSING



FAST FOURIER TRANSFORM

การแปลงรูปแบบสัญญาณจาก Time domain ให้เป็น Frequency domain เพื่อทำให้โมเดลได้เห็นภาพรวมความถี่ในแต่ละ Activity เนื่องจากแต่ละ Activity มีลักษณะที่แตกต่าง กันพอสมควร และยังช่วยลด Dimension ของข้อมูลเนื่องจากความถี่ครึ่งหลังจะเป็นความถี่ซ้ำซ้อนสามารถลบออกก่อนนำไปให้โมเดลทำนายได้



DATA PREPROCESSING

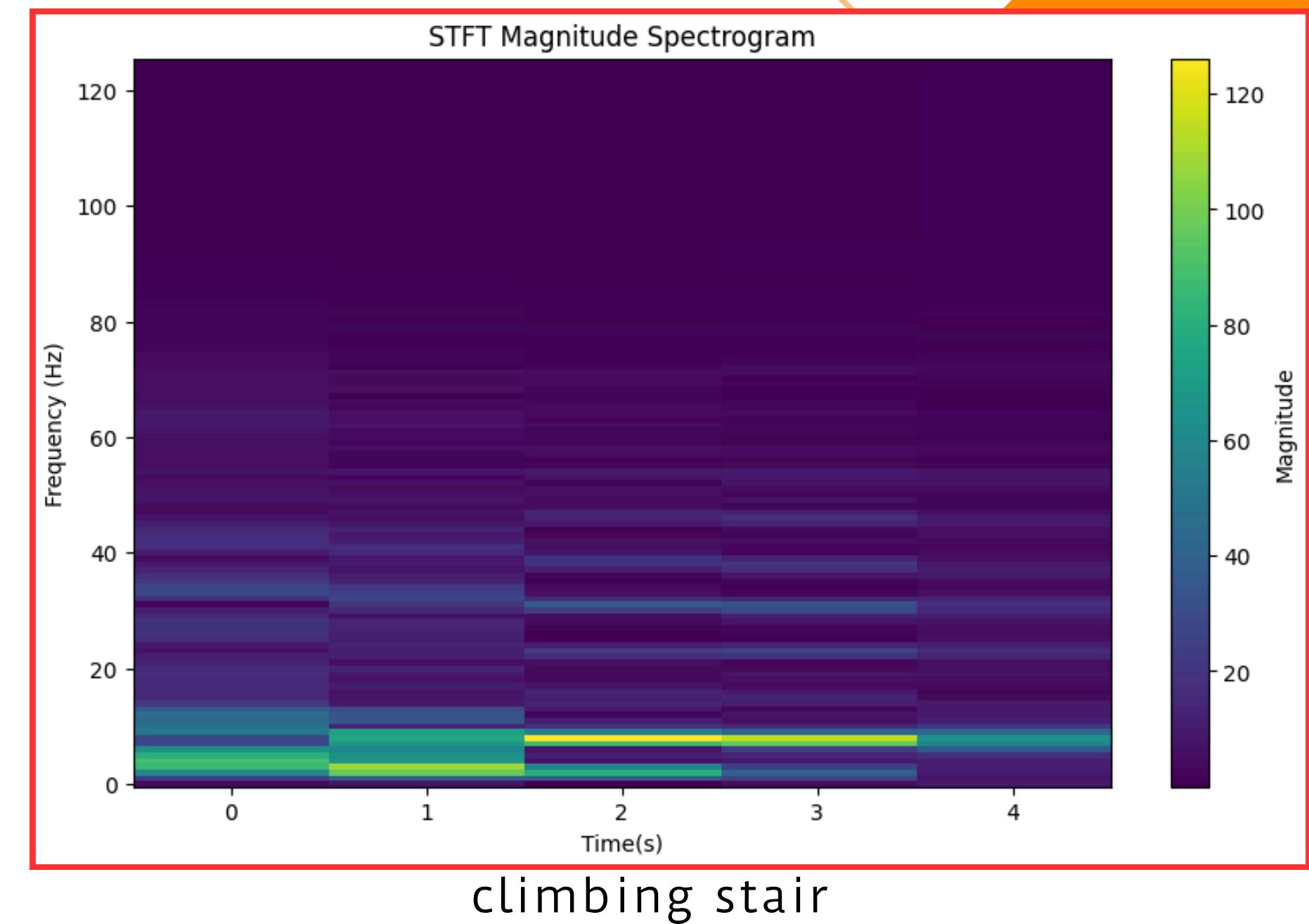


SHORT TIME FOURIER TRANSFORM

การแปลงสัญญาณให้อยู่ใน time-frequency domain จะได้ข้อมูลภาพการเปลี่ยนแปลงความถี่ตามเวลา (spectrogram) เพื่อให้เห็นภาพรวมของการเปลี่ยนแปลงความถี่ของสัญญาณตามเวลา

window ที่ใช้คือ Hann window เนื่องจากลดการรั่วไหลของความถี่ (spectral leakage) ได้ดี และยังคงความละเอียดความถี่ที่ดี

window size เป็น wide window เพื่อความละเอียดความถี่ที่ดี เนื่องจาก sample rate ที่ต่ำ

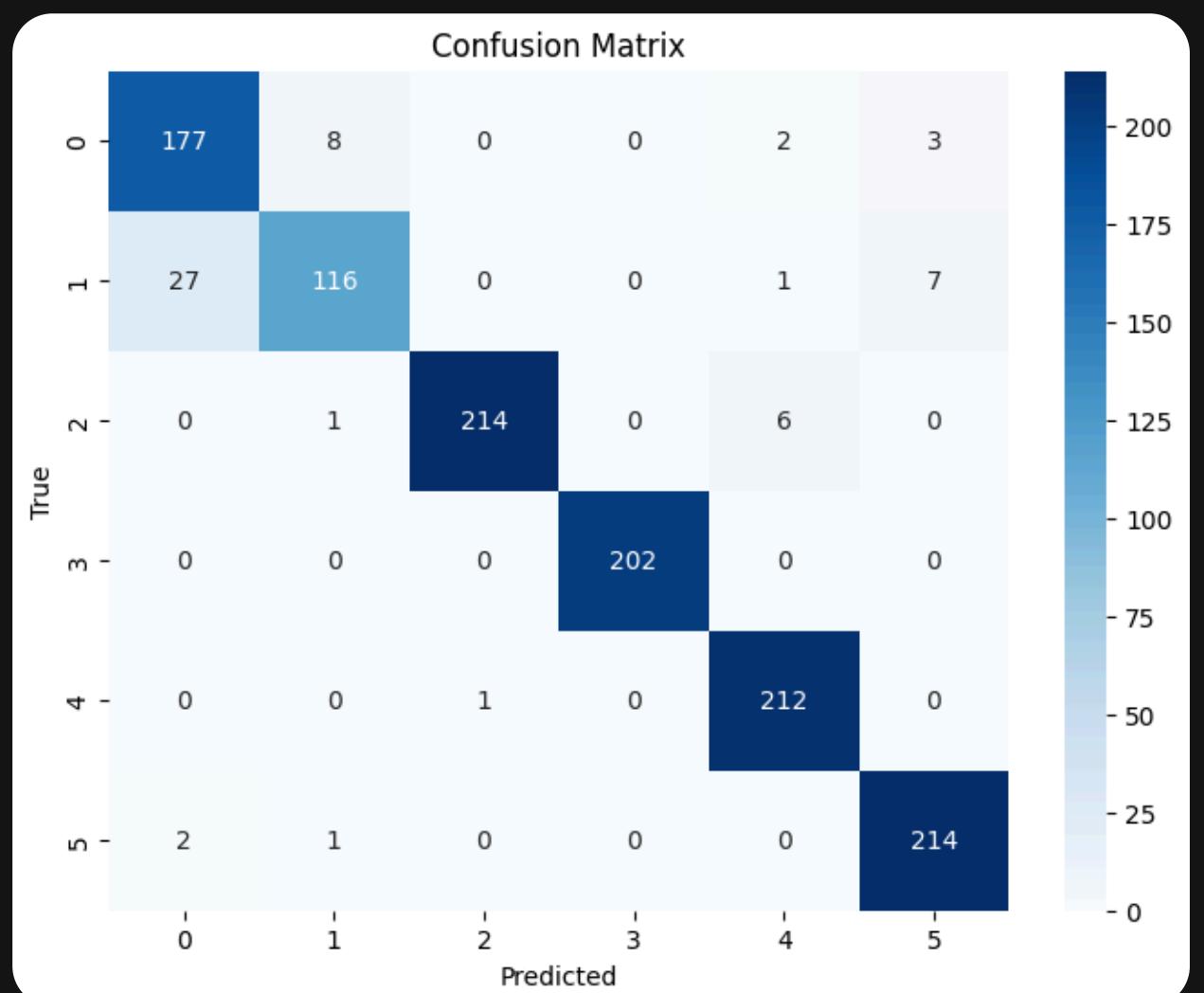


MODELS

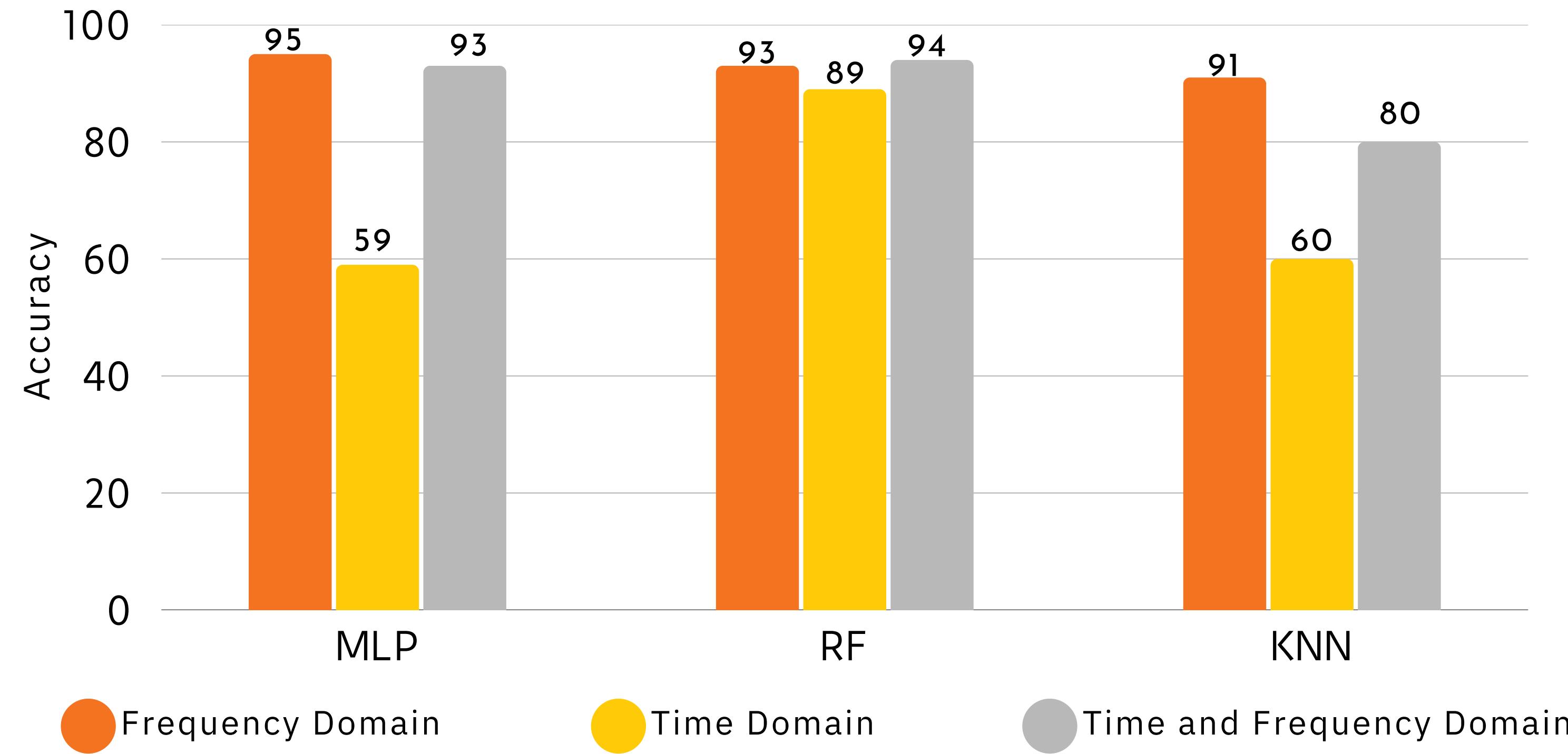
- **MODEL 1 CNN**
Image -> Time-Frequency domain
- **MODEL 2 MULTILAYER PERCEPTRON**
Tabular -> Time domain, Frequency domain and combine.
- **MODEL 3 RANDOM FOREST**
Tabular -> Time domain, Frequency domain and combine.
- **MODEL 4 K-NEAREST NEIGHBORS**
Tabular -> Time domain, Frequency domain and combine.

EVALUATE

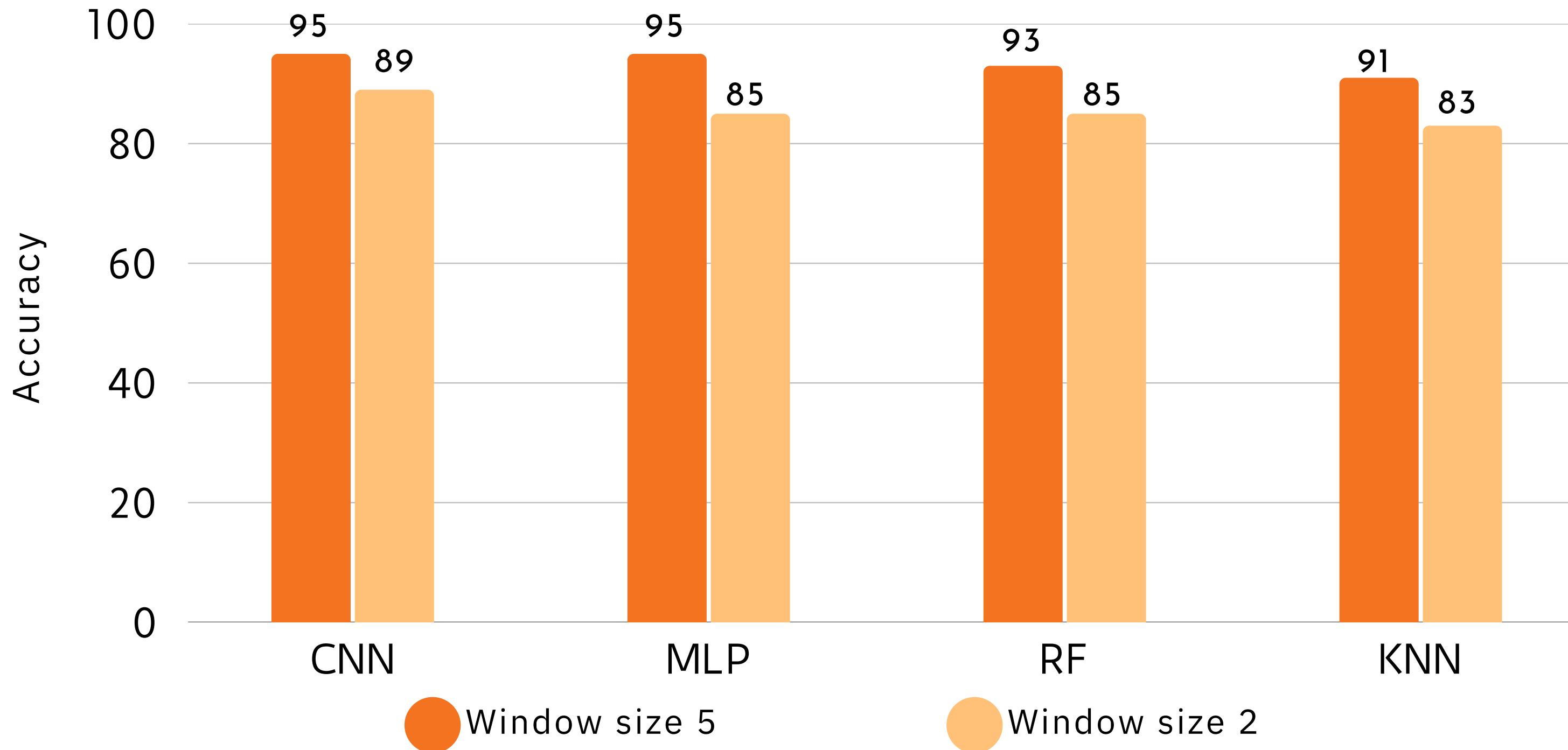
confusion metric & accuracy



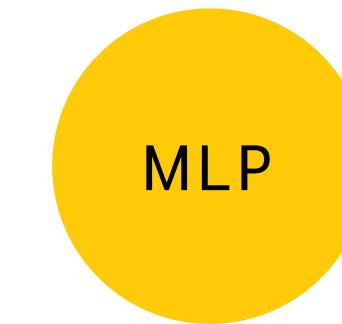
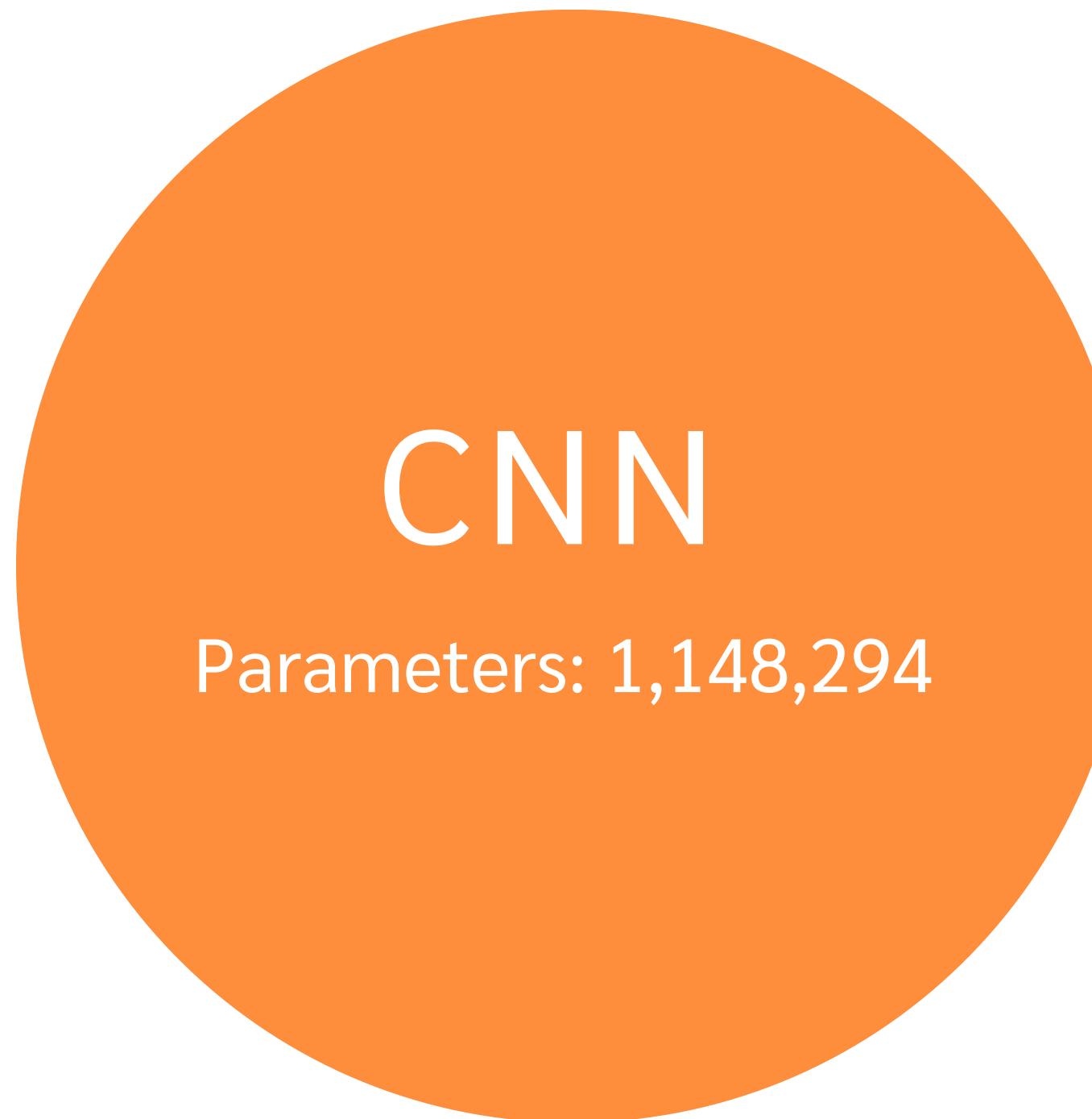
RESULTS (TABULAR)



RESULTS (TABULAR VS IMAGE)

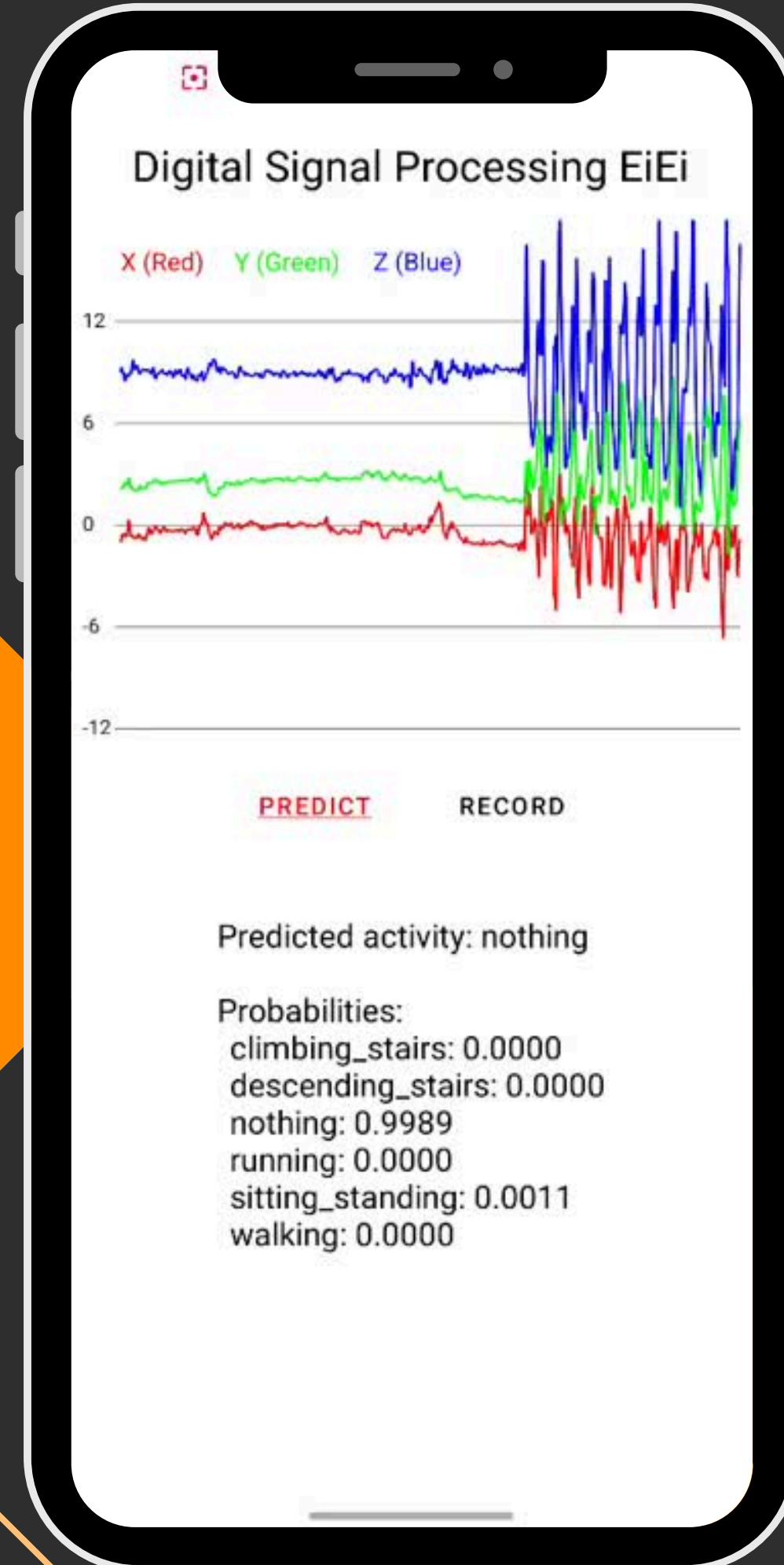


BEST MODEL FOR DEPLOY



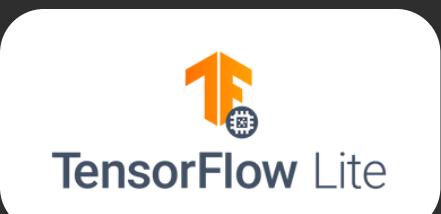
Parameters: 27,256

Same 95% accuracy at window 5 second



DEPLOY

Export MLP Model to TFLite format
for deploy on Android Application
using Android Studio with Kotlin.





THANK YOU