

AMATH 582 Final Project

Tyrone DeSilva

February 21, 2020

Abstract

- 1 Introduction and Overview
- 2 Theoretical Background
- 3 Algorithm Implementation and Development
- 4 Computational Results
- 5 Summary and Conclusions

Appendix A Python Functions

- Placeholder

Appendix B Python Code

B.1 preprocess.py

```
from scipy import signal
import load
import numpy as np
from tqdm import tqdm
import argparse

Fs = 40000000
n = int(Fs * 20e-3)
k = 24

def wavelet(s):
    downsampled = signal.resample(s, n // 2 // 1600)
    widths = [2 ** (j / k) for j in range(1, 101)]
    z = signal.cwt(downsampled, signal.ricker, widths)
    return z
```

```

def preprocess(loader, output):
    signals, meta = loader()
    result = np.zeros((int(25000), signals.shape[1]), dtype=np.float32)
    for i in tqdm(range(signals.shape[1])):
        z = wavelet(signals.iloc[:, i])
        z = np.ravel(z)
        result[:, i] = z.astype(np.float32)
    np.save(output, result)

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument("--train", action="store_true", default=False)
    parser.add_argument("--test", action="store_true", default=False)
    args = parser.parse_args()
    if args.train:
        preprocess(load.load_train, "data/preprocessed/train.npy")
    if args.test:
        preprocess(load.load_test, "data/preprocessed/test.npy")

```

B.2 load.py

```

import pandas as pd
import pyarrow.parquet as pq
import os

```

```

THIS_FILE_DIR = os.path.dirname(os.path.abspath(__file__))
DATA_DIR = os.path.join(THIS_FILE_DIR, "data/vsb-power-line-fault-detection")

```

```

def load_train(n_columns: int = None) -> (pd.DataFrame, pd.DataFrame):
    columns = None
    if n_columns:
        columns = [str(i) for i in range(n_columns)]
    train_data = pq.read_pandas(
        os.path.join(DATA_DIR, "train.parquet"), columns=columns
    ).to_pandas()
    train_meta = pd.read_csv(
        os.path.join(DATA_DIR, "metadata_train.csv"),
        index_col="signal_id",
        nrows=n_columns,
    )
    return train_data, train_meta

```

```

def load_test(n_columns: int = None) -> (pd.DataFrame, pd.DataFrame):
    columns = None
    if n_columns:
        columns = [str(i) for i in range(n_columns)]
    test_data = pq.read_pandas(
        os.path.join(DATA_DIR, "test.parquet"), columns=columns
    ).to_pandas()
    test_meta = pd.read_csv(

```

```
    os.path.join(DATA_DIR, "metadata_test.csv"),
    index_col="signal_id",
    nrows=n_columns,
)
return test_data, test_meta
```