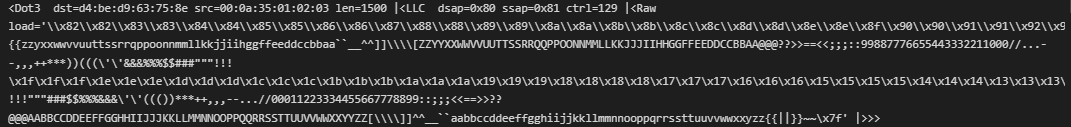
**After Receiving Data through Ethernet**

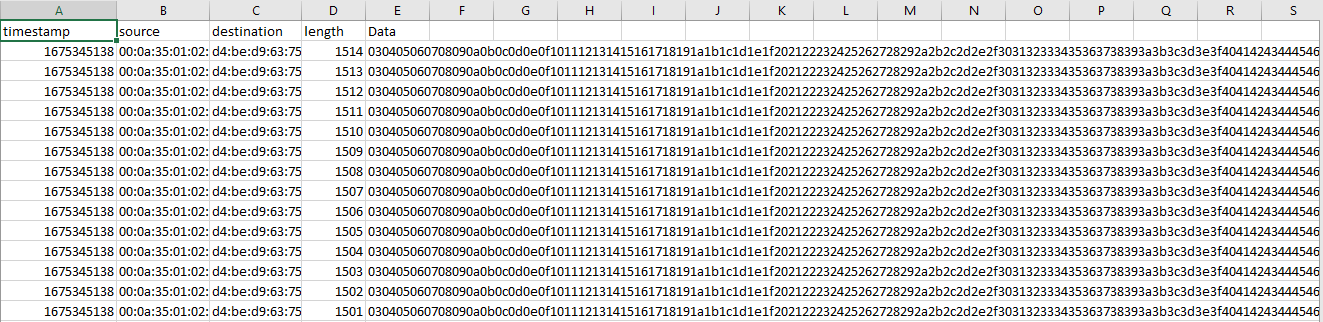
**Flow diagram**



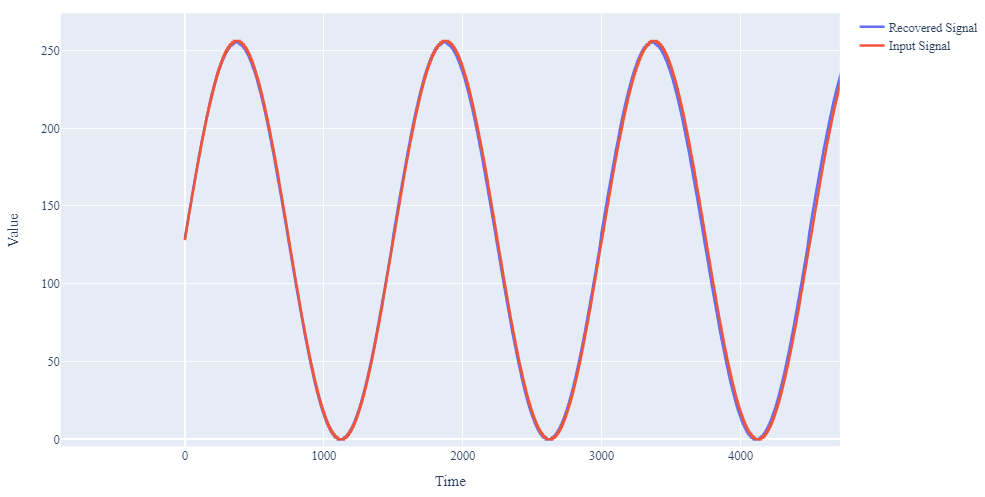
**Experimental Results**

**Received Format of Ethernet Packet into PC:**

**Preprocess Format of Received Ethernet Data (Can be used to train supervised machine learning models):**



**Plotting Recovered Data:**



**Supported Python Code**

**Importing Modules:**

from scapy.all import \* # Packet manipulation

import pandas as pd # Pandas - Create and Manipulate DataFrames

import numpy as np # Math Stuff (don't worry only used for one line :] )

import binascii # Binary to Ascii

import sys  # This module provides access to some objects used or maintained by the

import pandas as pd

from scapy.layers.l2 import Ether

from scapy.layers.inet import TCP, UDP

**Reading and Data Decryption:**

pcap = rdpcap("sinusoidal\_waveform.pcapng")   # Insert File here

df = pd.DataFrame()

b=[]

for packet in pcap:

    hexvalue = binascii.hexlify(packet.load).decode()

    for i in range(0, len(hexvalue), 2):

            b.append(int(hexvalue[i:i+2], 16))

df['data'] = b

**Data Labelling for machine learning:**

import csv

df = pd.DataFrame()

def pcap\_to\_csv(pcap\_file, csv\_file):

    # Load the pcap file

    pcap = rdpcap(pcap\_file)

    # Open the csv file for writing

    with open(csv\_file, 'w', newline='') as f:

        writer = csv.writer(f)

        # Write the header row

        writer.writerow(['timestamp', 'source', 'destination', 'length','Data'])

        # Iterate through each packet in the pcap

        for packet in pcap:

 # Get the timestamp, source and destination addresses, protocol, and packet length

            timestamp = packet.time

            source = packet.src

            destination = packet.dst

            length = len(packet)

            Data = packet.load

            b = []

            hexvalue = binascii.hexlify(Data).decode()

            # Write a row to the csv file

            writer.writerow([timestamp, source, destination, length,hexvalue])

pcap\_to\_csv("mutiple\_frame.pcap", "sample.csv")

**Plotting Data:**

from plotly.subplots import make\_subplots

import plotly.graph\_objects as go

fig = make\_subplots(rows=1, cols=1)

fig.add\_trace(

    # Aroon indicator

    go.Scatter(

    y=df['data'],

    mode='lines',

    name='Data',

    line\_width=3

    ),

row=1, col=1

)

fig.update\_xaxes(title\_text="Time", row=1, col=1)

fig.update\_yaxes(title\_text="Value", row=1, col=1)

fig.update\_layout(height=700, width=1200,font\_family='Times New Roman',font\_size=15)

fig.show()

fig.write\_image("figure.svg", engine="kaleido")