

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import confusion_matrix, classification_report

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

/Users/hassankhalid/tf_env/lib/python3.12/site-packages/keras/src/export/tf2onnx_lib.py:
8: FutureWarning: In the future `np.object` will be defined as the corresponding NumPy scalar.

    if not hasattr(np, "object"):
```

In [2]:

```
data = pd.read_csv("/Users/hassankhalid/Downloads/motor_current_data-2.csv")
data.head()
```

Out[2]:

	RMS	Peak	DominantFreq	PeakMag	SpectralEnergy	SidebandEnergy	Label
0	0.655311	0.975335		50	0.437776	0.429432	0.209267
1	0.585150	0.875086		50	0.404756	0.342400	0.169323
2	0.700628	1.027384		50	0.494618	0.490880	0.245114
3	0.544065	0.810040		50	0.380513	0.296007	0.147083
4	0.578313	0.849470		50	0.394034	0.334446	0.164377

In [3]:

```
print(data.shape)
print(data.columns)
print(data.info())

(800, 7)
Index(['RMS', 'Peak', 'DominantFreq', 'PeakMag', 'SpectralEnergy',
       'SidebandEnergy', 'Label'],
      dtype='object')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 800 entries, 0 to 799
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype  
 --- 
 0   RMS              800 non-null    float64
 1   Peak             800 non-null    float64
 2   DominantFreq    800 non-null    int64  
 3   PeakMag          800 non-null    float64
 4   SpectralEnergy  800 non-null    float64
 5   SidebandEnergy  800 non-null    float64
 6   Label            800 non-null    int64  
dtypes: float64(5), int64(2)
memory usage: 43.9 KB
None
```

In [4]:

```
X = data.drop(columns=["Label"])
y = data["Label"]
```

In [5]:

```
encoder = LabelEncoder()
y = encoder.fit_transform(y)
```

In [6]:

```
X_train, X_temp, y_train, y_temp = train_test_split(
    X, y,
    test_size=0.30,
    random_state=42,
    stratify=y
)

X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp,
    test_size=0.50,
    random_state=42,
    stratify=y_temp
)
```

In [7]:

```
scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)
X_val = scaler.transform(X_val)
X_test = scaler.transform(X_test)
```

In [26]:

```
model = Sequential()

model.add(Dense(
    64,
    activation="relu",
    input_shape=(X_train.shape[1],)
))
model.add(Dense(
    16,
    activation="relu",
    input_shape=(X_train.shape[1],)
))

model.add(Dense(
    len(np.unique(y)),
    activation="softmax"
))
```

In [27]:

```
model.compile(
    optimizer="adam",
    loss="sparse_categorical_crossentropy",
    metrics=["accuracy"]
)
```

In [28]:

```
history = model.fit(  
    X_train, y_train,  
    validation_data=(X_val, y_val),  
    epochs=120,  
    batch_size=32  
)
```

```
Epoch 1/120  
18/18 ━━━━━━━━ 1s 20ms/step - accuracy: 0.2482 - loss: 1.4349 - val_accuracy: 0.3583 - val_loss: 1.3465  
Epoch 2/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.3839 - loss: 1.2589 - val_accuracy: 0.4417 - val_loss: 1.2128  
Epoch 3/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.4482 - loss: 1.1501 - val_accuracy: 0.4833 - val_loss: 1.1260  
Epoch 4/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.4946 - loss: 1.0680 - val_accuracy: 0.5083 - val_loss: 1.0499  
Epoch 5/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.6107 - loss: 0.9956 - val_accuracy: 0.5667 - val_loss: 0.9869  
Epoch 6/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.6393 - loss: 0.9314 - val_accuracy: 0.5667 - val_loss: 0.9302  
Epoch 7/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.6482 - loss: 0.8750 - val_accuracy: 0.5833 - val_loss: 0.8825  
Epoch 8/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.6786 - loss: 0.8238 - val_accuracy: 0.5833 - val_loss: 0.8290  
Epoch 9/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.6964 - loss: 0.7838 - val_accuracy: 0.6083 - val_loss: 0.7973  
Epoch 10/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.7232 - loss: 0.7390 - val_accuracy: 0.6667 - val_loss: 0.7482  
Epoch 11/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.7286 - loss: 0.7133 - val_accuracy: 0.6583 - val_loss: 0.7237  
Epoch 12/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.7536 - loss: 0.6768 - val_accuracy: 0.6833 - val_loss: 0.6869  
Epoch 13/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.7571 - loss: 0.6471 - val_accuracy: 0.7250 - val_loss: 0.6614  
Epoch 14/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.7661 - loss: 0.6224 - val_accuracy: 0.7500 - val_loss: 0.6357  
Epoch 15/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.7839 - loss: 0.5950 - val_accuracy: 0.7500 - val_loss: 0.6118  
Epoch 16/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.7964 - loss: 0.5754 - val_accuracy: 0.7750 - val_loss: 0.5878  
Epoch 17/120  
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.7804 - loss: 0.5597 - val_accuracy:
```

```
y: 0.7917 - val_loss: 0.5661
Epoch 18/120
18/18 0s 10ms/step - accuracy: 0.8054 - loss: 0.5474 - val_accurac
y: 0.7667 - val_loss: 0.5589
Epoch 19/120
18/18 0s 10ms/step - accuracy: 0.7839 - loss: 0.5337 - val_accurac
y: 0.8167 - val_loss: 0.5330
Epoch 20/120
18/18 0s 10ms/step - accuracy: 0.7946 - loss: 0.5171 - val_accurac
y: 0.8167 - val_loss: 0.5188
Epoch 21/120
18/18 0s 10ms/step - accuracy: 0.8107 - loss: 0.5045 - val_accurac
y: 0.7833 - val_loss: 0.5176
Epoch 22/120
18/18 0s 10ms/step - accuracy: 0.8054 - loss: 0.4921 - val_accurac
y: 0.8000 - val_loss: 0.4960
Epoch 23/120
18/18 0s 10ms/step - accuracy: 0.8143 - loss: 0.4778 - val_accurac
y: 0.8417 - val_loss: 0.4856
Epoch 24/120
18/18 0s 10ms/step - accuracy: 0.8000 - loss: 0.4751 - val_accurac
y: 0.8167 - val_loss: 0.4695
Epoch 25/120
18/18 0s 10ms/step - accuracy: 0.8214 - loss: 0.4610 - val_accurac
y: 0.8083 - val_loss: 0.4695
Epoch 26/120
18/18 0s 10ms/step - accuracy: 0.8196 - loss: 0.4520 - val_accurac
y: 0.8583 - val_loss: 0.4517
Epoch 27/120
18/18 0s 10ms/step - accuracy: 0.8161 - loss: 0.4522 - val_accurac
y: 0.8333 - val_loss: 0.4414
Epoch 28/120
18/18 0s 10ms/step - accuracy: 0.8071 - loss: 0.4459 - val_accurac
y: 0.8250 - val_loss: 0.4464
Epoch 29/120
18/18 0s 10ms/step - accuracy: 0.8482 - loss: 0.4350 - val_accurac
y: 0.8583 - val_loss: 0.4260
Epoch 30/120
18/18 0s 10ms/step - accuracy: 0.8196 - loss: 0.4297 - val_accurac
y: 0.8500 - val_loss: 0.4184
Epoch 31/120
18/18 0s 10ms/step - accuracy: 0.8357 - loss: 0.4190 - val_accurac
y: 0.8583 - val_loss: 0.4085
Epoch 32/120
18/18 0s 10ms/step - accuracy: 0.8446 - loss: 0.4085 - val_accurac
y: 0.8417 - val_loss: 0.4251
Epoch 33/120
18/18 0s 10ms/step - accuracy: 0.8429 - loss: 0.4167 - val_accurac
y: 0.8917 - val_loss: 0.4087
Epoch 34/120
18/18 0s 10ms/step - accuracy: 0.8393 - loss: 0.4052 - val_accurac
y: 0.8667 - val_loss: 0.4023
Epoch 35/120
18/18 0s 10ms/step - accuracy: 0.8357 - loss: 0.3970 - val_accurac
y: 0.8250 - val_loss: 0.3992
Epoch 36/120
18/18 0s 11ms/step - accuracy: 0.8393 - loss: 0.3937 - val_accurac
y: 0.8583 - val_loss: 0.4024
Epoch 37/120
```

```
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8518 - loss: 0.3862 - val_accurac
y: 0.8250 - val_loss: 0.3894
Epoch 38/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8500 - loss: 0.3821 - val_accurac
y: 0.8750 - val_loss: 0.3836
Epoch 39/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8429 - loss: 0.3738 - val_accurac
y: 0.8583 - val_loss: 0.3722
Epoch 40/120
18/18 ━━━━━━━━ 0s 11ms/step - accuracy: 0.8464 - loss: 0.3828 - val_accurac
y: 0.8250 - val_loss: 0.3955
Epoch 41/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8393 - loss: 0.3727 - val_accurac
y: 0.8500 - val_loss: 0.3738
Epoch 42/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8464 - loss: 0.3678 - val_accurac
y: 0.8417 - val_loss: 0.3823
Epoch 43/120
18/18 ━━━━━━━━ 0s 11ms/step - accuracy: 0.8589 - loss: 0.3685 - val_accurac
y: 0.8750 - val_loss: 0.3646
Epoch 44/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8518 - loss: 0.3690 - val_accurac
y: 0.8667 - val_loss: 0.3748
Epoch 45/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8643 - loss: 0.3471 - val_accurac
y: 0.8750 - val_loss: 0.3406
Epoch 46/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8589 - loss: 0.3514 - val_accurac
y: 0.8833 - val_loss: 0.3318
Epoch 47/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8625 - loss: 0.3458 - val_accurac
y: 0.8917 - val_loss: 0.3289
Epoch 48/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8589 - loss: 0.3364 - val_accurac
y: 0.8167 - val_loss: 0.3621
Epoch 49/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8464 - loss: 0.3510 - val_accurac
y: 0.8833 - val_loss: 0.3229
Epoch 50/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8714 - loss: 0.3324 - val_accurac
y: 0.8833 - val_loss: 0.3281
Epoch 51/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8607 - loss: 0.3471 - val_accurac
y: 0.8917 - val_loss: 0.3159
Epoch 52/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8571 - loss: 0.3430 - val_accurac
y: 0.8833 - val_loss: 0.3437
Epoch 53/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8518 - loss: 0.3458 - val_accurac
y: 0.9000 - val_loss: 0.3094
Epoch 54/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8393 - loss: 0.3427 - val_accurac
y: 0.9000 - val_loss: 0.3055
Epoch 55/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8589 - loss: 0.3229 - val_accurac
y: 0.8917 - val_loss: 0.2966
Epoch 56/120
18/18 ━━━━━━━━ 0s 10ms/step - accuracy: 0.8554 - loss: 0.3378 - val_accurac
y: 0.8917 - val_loss: 0.3058
```

```
Epoch 57/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8982 - loss: 0.3075 - val_accurac
y: 0.9250 - val_loss: 0.3015
Epoch 58/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8661 - loss: 0.3205 - val_accurac
y: 0.9000 - val_loss: 0.3327
Epoch 59/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8625 - loss: 0.3184 - val_accurac
y: 0.9250 - val_loss: 0.2876
Epoch 60/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8768 - loss: 0.3063 - val_accurac
y: 0.9083 - val_loss: 0.2988
Epoch 61/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8929 - loss: 0.3009 - val_accurac
y: 0.9083 - val_loss: 0.2964
Epoch 62/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8625 - loss: 0.3190 - val_accurac
y: 0.8083 - val_loss: 0.3481
Epoch 63/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8750 - loss: 0.3049 - val_accurac
y: 0.8583 - val_loss: 0.3194
Epoch 64/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8607 - loss: 0.3118 - val_accurac
y: 0.8167 - val_loss: 0.3732
Epoch 65/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8786 - loss: 0.3086 - val_accurac
y: 0.8917 - val_loss: 0.2925
Epoch 66/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8982 - loss: 0.2854 - val_accurac
y: 0.8750 - val_loss: 0.3063
Epoch 67/120
18/18 ██████████ 0s 11ms/step - accuracy: 0.8679 - loss: 0.3061 - val_accurac
y: 0.8583 - val_loss: 0.3138
Epoch 68/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8821 - loss: 0.3072 - val_accurac
y: 0.8917 - val_loss: 0.3100
Epoch 69/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8750 - loss: 0.2909 - val_accurac
y: 0.9250 - val_loss: 0.2568
Epoch 70/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8911 - loss: 0.2820 - val_accurac
y: 0.9000 - val_loss: 0.2935
Epoch 71/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8786 - loss: 0.2866 - val_accurac
y: 0.9333 - val_loss: 0.2641
Epoch 72/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8857 - loss: 0.2861 - val_accurac
y: 0.9083 - val_loss: 0.2533
Epoch 73/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8839 - loss: 0.2945 - val_accurac
y: 0.9000 - val_loss: 0.2976
Epoch 74/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.8696 - loss: 0.3100 - val_accurac
y: 0.9000 - val_loss: 0.2714
Epoch 75/120
18/18 ██████████ 0s 11ms/step - accuracy: 0.8911 - loss: 0.2896 - val_accurac
y: 0.9000 - val_loss: 0.2754
Epoch 76/120
18/18 ██████████ 0s 10ms/step - accuracy: 0.9000 - loss: 0.2681 - val_accurac
```

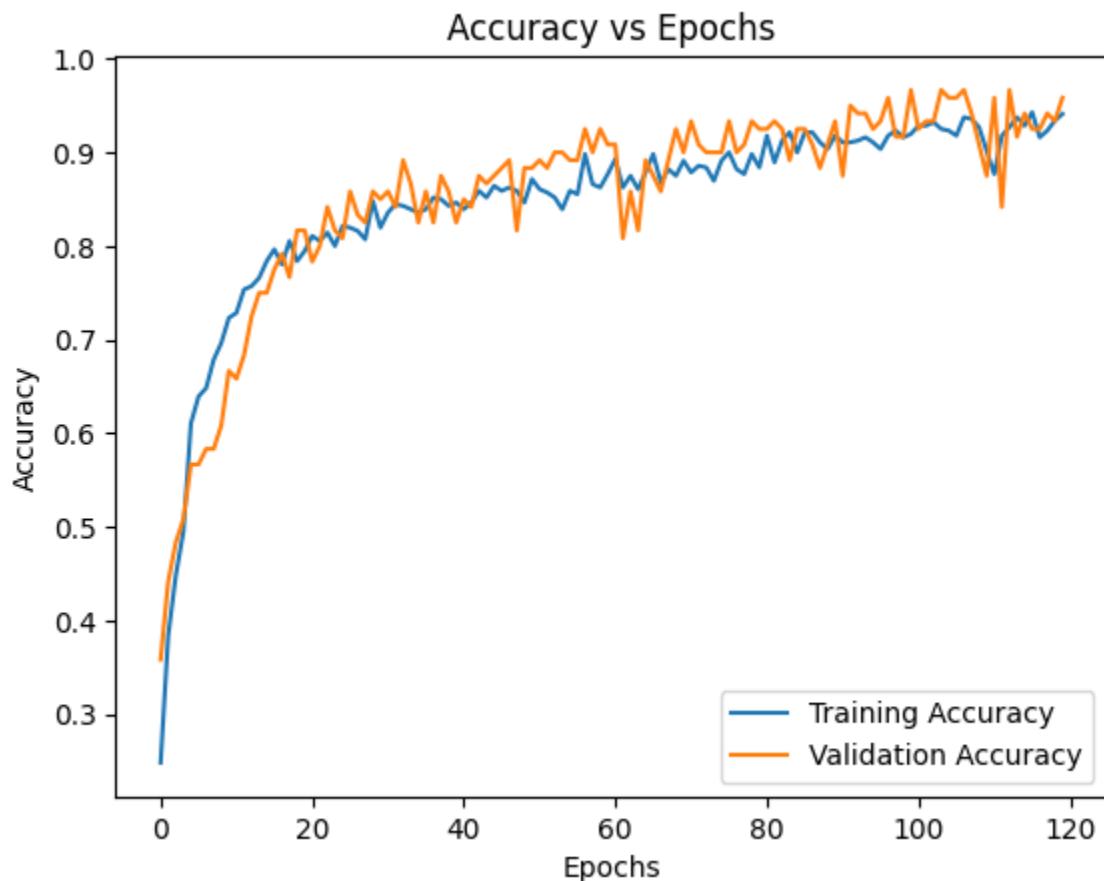
```
y: 0.9333 - val_loss: 0.2397
Epoch 77/120
18/18 0s 10ms/step - accuracy: 0.8821 - loss: 0.2812 - val_accurac
y: 0.9000 - val_loss: 0.2787
Epoch 78/120
18/18 0s 10ms/step - accuracy: 0.8768 - loss: 0.2895 - val_accurac
y: 0.9083 - val_loss: 0.2543
Epoch 79/120
18/18 0s 10ms/step - accuracy: 0.8982 - loss: 0.2782 - val_accurac
y: 0.9333 - val_loss: 0.2479
Epoch 80/120
18/18 0s 10ms/step - accuracy: 0.8839 - loss: 0.2699 - val_accurac
y: 0.9250 - val_loss: 0.2385
Epoch 81/120
18/18 0s 10ms/step - accuracy: 0.9179 - loss: 0.2498 - val_accurac
y: 0.9250 - val_loss: 0.2559
Epoch 82/120
18/18 0s 11ms/step - accuracy: 0.8893 - loss: 0.2689 - val_accurac
y: 0.9333 - val_loss: 0.2273
Epoch 83/120
18/18 0s 10ms/step - accuracy: 0.9125 - loss: 0.2504 - val_accurac
y: 0.9250 - val_loss: 0.2485
Epoch 84/120
18/18 0s 10ms/step - accuracy: 0.9214 - loss: 0.2434 - val_accurac
y: 0.8917 - val_loss: 0.2590
Epoch 85/120
18/18 0s 10ms/step - accuracy: 0.9000 - loss: 0.2702 - val_accurac
y: 0.9250 - val_loss: 0.2278
Epoch 86/120
18/18 0s 10ms/step - accuracy: 0.9214 - loss: 0.2452 - val_accurac
y: 0.9250 - val_loss: 0.2394
Epoch 87/120
18/18 0s 10ms/step - accuracy: 0.9214 - loss: 0.2473 - val_accurac
y: 0.9083 - val_loss: 0.2432
Epoch 88/120
18/18 0s 10ms/step - accuracy: 0.9107 - loss: 0.2322 - val_accurac
y: 0.8833 - val_loss: 0.2559
Epoch 89/120
18/18 0s 11ms/step - accuracy: 0.9036 - loss: 0.2520 - val_accurac
y: 0.9083 - val_loss: 0.2494
Epoch 90/120
18/18 0s 10ms/step - accuracy: 0.9179 - loss: 0.2390 - val_accurac
y: 0.9333 - val_loss: 0.2224
Epoch 91/120
18/18 0s 10ms/step - accuracy: 0.9107 - loss: 0.2494 - val_accurac
y: 0.8750 - val_loss: 0.2811
Epoch 92/120
18/18 0s 10ms/step - accuracy: 0.9107 - loss: 0.2400 - val_accurac
y: 0.9500 - val_loss: 0.2190
Epoch 93/120
18/18 0s 10ms/step - accuracy: 0.9125 - loss: 0.2347 - val_accurac
y: 0.9417 - val_loss: 0.2229
Epoch 94/120
18/18 0s 10ms/step - accuracy: 0.9161 - loss: 0.2321 - val_accurac
y: 0.9417 - val_loss: 0.2113
Epoch 95/120
18/18 0s 10ms/step - accuracy: 0.9107 - loss: 0.2292 - val_accurac
y: 0.9250 - val_loss: 0.2282
Epoch 96/120
```

```
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9036 - loss: 0.2382 - val_accurac
y: 0.9333 - val_loss: 0.2133
Epoch 97/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9179 - loss: 0.2373 - val_accurac
y: 0.9583 - val_loss: 0.1993
Epoch 98/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9232 - loss: 0.2219 - val_accurac
y: 0.9167 - val_loss: 0.2491
Epoch 99/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9161 - loss: 0.2400 - val_accurac
y: 0.9167 - val_loss: 0.2306
Epoch 100/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9196 - loss: 0.2261 - val_accurac
y: 0.9667 - val_loss: 0.1957
Epoch 101/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9286 - loss: 0.2182 - val_accurac
y: 0.9250 - val_loss: 0.2165
Epoch 102/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9286 - loss: 0.2135 - val_accurac
y: 0.9333 - val_loss: 0.2115
Epoch 103/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9321 - loss: 0.2150 - val_accurac
y: 0.9333 - val_loss: 0.2014
Epoch 104/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9250 - loss: 0.2092 - val_accurac
y: 0.9667 - val_loss: 0.1922
Epoch 105/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9232 - loss: 0.2276 - val_accurac
y: 0.9583 - val_loss: 0.1926
Epoch 106/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9179 - loss: 0.2270 - val_accurac
y: 0.9583 - val_loss: 0.2053
Epoch 107/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9375 - loss: 0.2099 - val_accurac
y: 0.9667 - val_loss: 0.1989
Epoch 108/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9357 - loss: 0.2157 - val_accurac
y: 0.9417 - val_loss: 0.2200
Epoch 109/120
18/18 ━━━━━━ 0s 11ms/step - accuracy: 0.9268 - loss: 0.2339 - val_accurac
y: 0.9083 - val_loss: 0.2355
Epoch 110/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9018 - loss: 0.2835 - val_accurac
y: 0.8750 - val_loss: 0.3471
Epoch 111/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.8768 - loss: 0.3071 - val_accurac
y: 0.9583 - val_loss: 0.2092
Epoch 112/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9179 - loss: 0.2277 - val_accurac
y: 0.8417 - val_loss: 0.3628
Epoch 113/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9268 - loss: 0.2089 - val_accurac
y: 0.9667 - val_loss: 0.1774
Epoch 114/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9375 - loss: 0.2045 - val_accurac
y: 0.9167 - val_loss: 0.2302
Epoch 115/120
18/18 ━━━━━━ 0s 10ms/step - accuracy: 0.9286 - loss: 0.2091 - val_accurac
y: 0.9417 - val_loss: 0.2008
```

```
Epoch 116/120
18/18 - 0s 11ms/step - accuracy: 0.9429 - loss: 0.2076 - val_accuracy: 0.9250 - val_loss: 0.2078
Epoch 117/120
18/18 - 0s 10ms/step - accuracy: 0.9161 - loss: 0.2661 - val_accuracy: 0.9250 - val_loss: 0.3315
Epoch 118/120
18/18 - 0s 10ms/step - accuracy: 0.9232 - loss: 0.2432 - val_accuracy: 0.9417 - val_loss: 0.2031
Epoch 119/120
18/18 - 0s 10ms/step - accuracy: 0.9339 - loss: 0.1943 - val_accuracy: 0.9333 - val_loss: 0.1758
Epoch 120/120
18/18 - 0s 10ms/step - accuracy: 0.9411 - loss: 0.1864 - val_accuracy: 0.9583 - val_loss: 0.1877
```

In [29]:

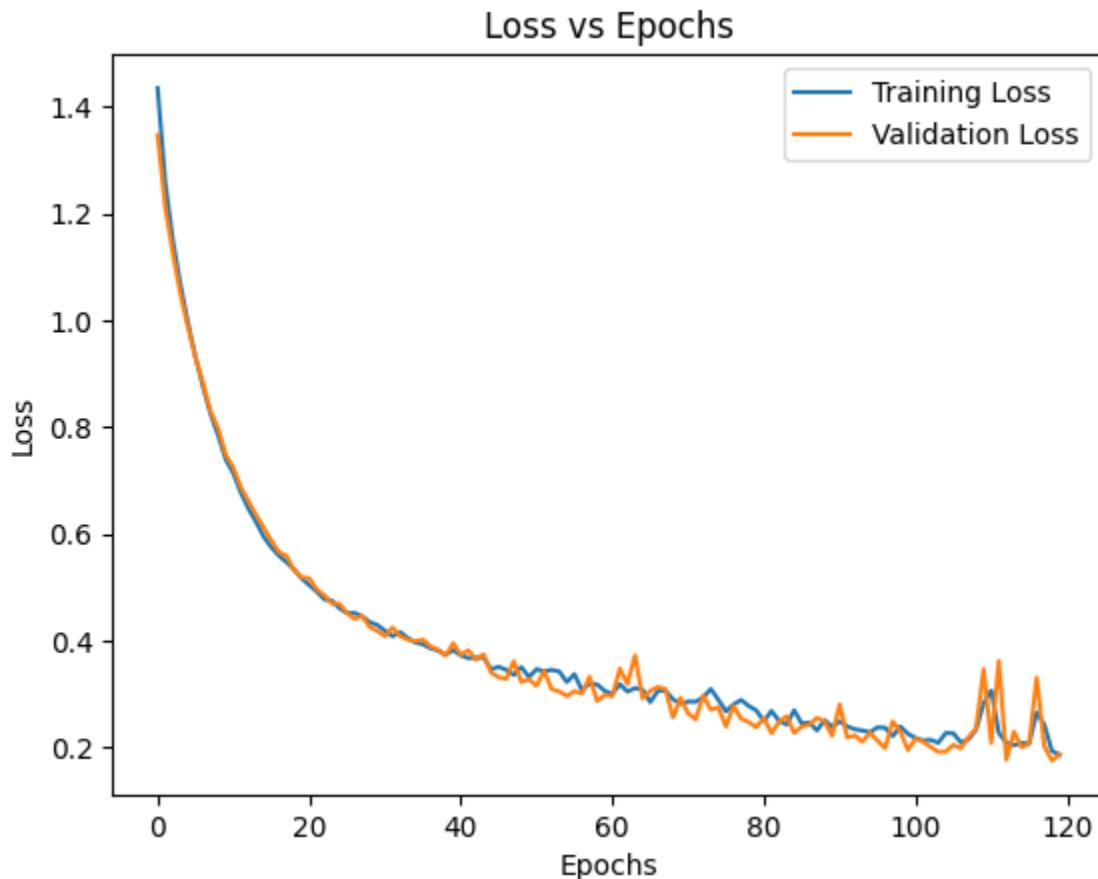
```
plt.figure()
plt.plot(history.history["accuracy"], label="Training Accuracy")
plt.plot(history.history["val_accuracy"], label="Validation Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.title("Accuracy vs Epochs")
plt.legend()
plt.show()
```



In [30]:

```
plt.figure()
plt.plot(history.history["loss"], label="Training Loss")
plt.plot(history.history["val_loss"], label="Validation Loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
```

```
plt.title("Loss vs Epochs")
plt.legend()
plt.show()
```



In [31]:

```
test_loss, test_accuracy = model.evaluate(X_test, y_test)

print("Unseen Test Accuracy:", test_accuracy)
print("Unseen Test Loss:", test_loss)
```

```
4/4 ━━━━━━━━ 0s 10ms/step - accuracy: 0.9417 - loss: 0.2066
Unseen Test Accuracy: 0.9416666626930237
Unseen Test Loss: 0.2065727859735489
```

In [32]:

```
y_pred = model.predict(X_test)
y_pred_classes = np.argmax(y_pred, axis=1)

print(confusion_matrix(y_test, y_pred_classes))
print(classification_report(y_test, y_pred_classes))
```

```
4/4 ━━━━━━━━ 0s 10ms/step
[[30  0  0  0]
 [ 0 25  5  0]
 [ 0  0 30  0]
 [ 2  0  0 28]]
          precision    recall  f1-score   support
          0       0.94      1.00     0.97      30
          1       1.00      0.83     0.91      30
          2       0.86      1.00     0.92      30
          3       1.00      0.93     0.97      30
```

```
accuracy           0.94      120
macro avg         0.94      120
weighted avg     0.94      120
```

In [33]:

```
model.save("motor_fault_model-3.keras")

import joblib
joblib.dump(scaler, "scaler3.pkl")
joblib.dump(encoder, "label_encoder3.pkl")
```

Out[33]:

```
['label_encoder3.pkl']
```

In [34]:

```
#ADDING NOISE
noise = np.random.normal(0, 0.05, X_test.shape)
X_test_noisy = X_test + noise

model.evaluate(X_test_noisy, y_test)
```

4/4 ————— 0s 10ms/step - accuracy: 0.8917 - loss: 0.2719

Out[34]:

```
[0.27190497517585754, 0.8916666507720947]
```

In [35]:

```
#DOING IT WITH MORE NOISE
noise = np.random.normal(0, 0.5, X_test.shape)
X_test_noisy = X_test + noise

model.evaluate(X_test_noisy, y_test)
```

4/4 ————— 0s 10ms/step - accuracy: 0.4250 - loss: 6.2994

Out[35]:

```
[6.299350261688232, 0.42500001192092896]
```

In [20]:

```
#FINALLY ACCURACY DROPS AND NOW IT FEELS NO OVERFITTING HAPPENED
```

In [36]:

```
model.save("motor_fault_model-3.keras")

import joblib
joblib.dump(scaler, "scaler-3.pkl")
joblib.dump(encoder, "label_encoder-3.pkl")
```

Out[36]:

```
['label_encoder-3.pkl']
```

In []: