In [3]:
```
!pip install pyswarms
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting pyswarms
  Downloading pyswarms-1.3.0-py2.py3-none-any.whl (104 kB)
     ------------------------------------ 104.1/104.1 kB 601.6 kB/s eta 0:00:00
Requirement already satisfied: scipy in c:\users\hp\appdata\roaming\python\pyth
on39\site-packages (from pyswarms) (1.10.1)
Requirement already satisfied: matplotlib>=1.3.1 in c:\programdata\anaconda3\li
b\site-packages (from pyswarms) (3.5.2)
Requirement already satisfied: numpy in c:\users\hp\appdata\roaming\python\pyth
on39\site-packages (from pyswarms) (1.23.5)
Requirement already satisfied: attrs in c:\users\hp\appdata\roaming\python\pyth
on39\site-packages (from pyswarms) (23.1.0)
Requirement already satisfied: future in c:\programdata\anaconda3\lib\site-pack
ages (from pyswarms) (0.18.2)
Requirement already satisfied: tqdm in c:\programdata\anaconda3\lib\site-packag
es (from pyswarms) (4.64.1)
Requirement already satisfied: pyyaml in c:\programdata\anaconda3\lib\site-pack
ages (from pyswarms) (6.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\programdata\anaconda3\li
b\site-packages (from matplotlib>=1.3.1->pyswarms) (1.4.2)
Requirement already satisfied: packaging>=20.0 in c:\users\hp\appdata\roaming\p
ython\python39\site-packages (from matplotlib>=1.3.1->pyswarms) (23.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\programdata\anaconda3
\lib\site-packages (from matplotlib>=1.3.1->pyswarms) (2.8.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\programdata\anaconda3\li
b\site-packages (from matplotlib>=1.3.1->pyswarms) (4.25.0)
Requirement already satisfied: cycler>=0.10 in c:\programdata\anaconda3\lib\sit
e-packages (from matplotlib>=1.3.1->pyswarms) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in c:\programdata\anaconda3\lib\si
te-packages (from matplotlib>=1.3.1->pyswarms) (9.2.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\programdata\anaconda3\lib
\site-packages (from matplotlib>=1.3.1->pyswarms) (3.0.9)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-pa
ckages (from tqdm->pyswarms) (0.4.5)
Requirement already satisfied: six>=1.5 in c:\users\hp\appdata\roaming\python\p
ython39\site-packages (from python-dateutil>=2.7->matplotlib>=1.3.1->pyswarms)
(1.16.0)
Installing collected packages: pyswarms
Successfully installed pyswarms-1.3.0
```

In [2]:

```
pip install tensorflow keras-tuner scikit-learn pandas numpy matplotlib
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: tensorflow in c:\users\hp\appdata\roaming\python
\python39\site-packages (1.14.0)
Collecting keras-tuner
  Downloading keras_tuner-1.4.7-py3-none-any.whl (129 kB)
     ------------------------------------ 129.1/129.1 kB 582.9 kB/s eta 0:00:00
Requirement already satisfied: scikit-learn in c:\programdata\anaconda3\lib\sit
e-packages (1.0.2)
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-pack
ages (1.4.4)
Requirement already satisfied: numpy in c:\users\hp\appdata\roaming\python\pyth
on39\site-packages (1.23.5)
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib\site-
packages (3.5.2)
Requirement already satisfied: keras-preprocessing>=1.0.5 in c:\users\hp\appdat
a\roaming\python\python39\site-packages (from tensorflow) (1.1.2)
Requirement already satisfied: wrapt>=1.11.1 in c:\users\hp\appdata\roaming\pyt
hon\python39\site-packages (from tensorflow) (1.14.1)
Collecting tensorflow-estimator<1.15.0rc0,>=1.14.0rc0
  Using cached tensorflow_estimator-1.14.0-py2.py3-none-any.whl (488 kB)
Collecting tensorboard<1.15.0,>=1.14.0
  Using cached tensorboard-1.14.0-py3-none-any.whl (3.1 MB)
Requirement already satisfied: absl-py>=0.7.0 in c:\users\hp\appdata\roaming\py
thon\python39\site-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astor>=0.6.0 in c:\users\hp\appdata\roaming\pyth
on\python39\site-packages (from tensorflow) (0.8.1)
Requirement already satisfied: google-pasta>=0.1.6 in c:\users\hp\appdata\roami
ng\python\python39\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: wheel>=0.26 in c:\users\hp\appdata\roaming\pytho
n\python39\site-packages (from tensorflow) (0.40.0)
Requirement already satisfied: six>=1.10.0 in c:\users\hp\appdata\roaming\pytho
n\python39\site-packages (from tensorflow) (1.16.0)
Requirement already satisfied: gast>=0.2.0 in c:\users\hp\appdata\roaming\pytho
n\python39\site-packages (from tensorflow) (0.4.0)
Requirement already satisfied: protobuf>=3.6.1 in c:\users\hp\appdata\roaming\p
ython\python39\site-packages (from tensorflow) (4.23.0)
Requirement already satisfied: keras-applications>=1.0.6 in c:\users\hp\appdata
\roaming\python\python39\site-packages (from tensorflow) (1.0.8)
Requirement already satisfied: grpcio>=1.8.6 in c:\users\hp\appdata\roaming\pyt
hon\python39\site-packages (from tensorflow) (1.54.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\hp\appdata\roaming
\python\python39\site-packages (from tensorflow) (2.3.0)
Requirement already satisfied: packaging in c:\users\hp\appdata\roaming\python
\python39\site-packages (from keras-tuner) (23.1)
Requirement already satisfied: requests in c:\users\hp\appdata\roaming\python\p
ython39\site-packages (from keras-tuner) (2.30.0)
Requirement already satisfied: keras in c:\users\hp\appdata\roaming\python\pyth
on39\site-packages (from keras-tuner) (2.15.0)
Collecting kt-legacy
  Downloading kt_legacy-1.0.5-py3-none-any.whl (9.6 kB)
Requirement already satisfied: joblib>=0.11 in c:\users\hp\appdata\roaming\pyth
on\python39\site-packages (from scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda3
\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: scipy>=1.1.0 in c:\users\hp\appdata\roaming\pyth
on\python39\site-packages (from scikit-learn) (1.10.1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\programdata\anacond
```

```
a3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\lib\sit
e-packages (from pandas) (2022.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\programdata\anaconda3\li
b\site-packages (from matplotlib) (1.4.2)
Requirement already satisfied: pillow>=6.2.0 in c:\programdata\anaconda3\lib\si
te-packages (from matplotlib) (9.2.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\programdata\anaconda3\li
b\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\programdata\anaconda3\lib
\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: cycler>=0.10 in c:\programdata\anaconda3\lib\sit
e-packages (from matplotlib) (0.11.0)
Requirement already satisfied: h5py in c:\programdata\anaconda3\lib\site-packag
es (from keras-applications>=1.0.6->tensorflow) (3.10.0)
Requirement already satisfied: markdown>=2.6.8 in c:\users\hp\appdata\roaming\p
ython\python39\site-packages (from tensorboard<1.15.0,>=1.14.0->tensorflow) (3.
4.3)
Requirement already satisfied: werkzeug>=0.11.15 in c:\users\hp\appdata\roaming
\python\python39\site-packages (from tensorboard<1.15.0,>=1.14.0->tensorflow)
(2.3.4)
Requirement already satisfied: setuptools>=41.0.0 in c:\users\hp\appdata\roamin
g\python\python39\site-packages (from tensorboard<1.15.0,>=1.14.0->tensorflow)
(67.7.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\hp\appdata\roamin
g\python\python39\site-packages (from requests->keras-tuner) (1.26.15)
Requirement already satisfied: idna<4,>=2.5 in c:\users\hp\appdata\roaming\pyth
on\python39\site-packages (from requests->keras-tuner) (3.4)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\hp\appdata
\roaming\python\python39\site-packages (from requests->keras-tuner) (3.1.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\hp\appdata\roamin
g\python\python39\site-packages (from requests->keras-tuner) (2023.5.7)
Requirement already satisfied: importlib-metadata>=4.4 in c:\users\hp\appdata\r
oaming\python\python39\site-packages (from markdown>=2.6.8->tensorboard<1.15.0,
>=1.14.0->tensorflow) (6.6.0)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\hp\appdata\roaming
\python\python39\site-packages (from werkzeug>=0.11.15->tensorboard<1.15.0,>=1.
14.0->tensorflow) (2.1.2)
Requirement already satisfied: zipp>=0.5 in c:\users\hp\appdata\roaming\python
\python39\site-packages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorb
oard<1.15.0,>=1.14.0->tensorflow) (3.15.0)
Installing collected packages: tensorflow-estimator, kt-legacy, keras-tuner, te
nsorboard
  Attempting uninstall: tensorflow-estimator
    Found existing installation: tensorflow-estimator 2.15.0
    Uninstalling tensorflow-estimator-2.15.0:
      Successfully uninstalled tensorflow-estimator-2.15.0
  Attempting uninstall: tensorboard
    Found existing installation: tensorboard 2.15.2
    Uninstalling tensorboard-2.15.2:
      Successfully uninstalled tensorboard-2.15.2
Successfully installed keras-tuner-1.4.7 kt-legacy-1.0.5 tensorboard-1.14.0 ten
sorflow-estimator-1.14.0
Note: you may need to restart the kernel to use updated packages.
```

```
ERROR: pip's dependency resolver does not currently take into account all the p
ackages that are installed. This behaviour is the source of the following depen
dency conflicts.
tensorflow-intel 2.12.0 requires keras<2.13,>=2.12.0, but you have keras 2.15.0
which is incompatible.
tensorflow-intel 2.12.0 requires tensorboard<2.13,>=2.12, but you have tensorbo
ard 1.14.0 which is incompatible.
tensorflow-intel 2.12.0 requires tensorflow-estimator<2.13,>=2.12.0, but you ha
ve tensorflow-estimator 1.14.0 which is incompatible.
```

In [3]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout, BatchNormalization
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
```

In [4]:

```python
df = pd.read_csv("vmCloud_data.csv")

df.head()
```

Out[4]:

	vm_id	timestamp	cpu_usage	memory_usage	network_traffic	power_consumption	num_
0	c5215826-6237-4a33-9312-72c1df909881	2023-01-25 09:10:54	54.881350	78.950861	164.775973	287.808986	
1	29690bc6-1f34-403b-b509-a1ecb1834fb8	2023-01-26 04:46:34	71.518937	29.901883	NaN	362.273569	
2	2e55abc3-5bad-46cb-b445-a577f5e9bf2a	2023-01-13 23:39:47	NaN	92.709195	203.674847	231.467903	
3	e672e32f-c134-4fbc-992b-34eb63bef6bf	2023-02-09 11:45:49	54.488318	88.100960	NaN	195.639954	
4	f38b8b50-6926-4533-be4f-89ad11624071	2023-06-14 08:27:26	42.365480	NaN	NaN	359.451537	

In [5]:

```
# Handle missing values
df = df.fillna(method='ffill')


df.shape
```

Out[5]:

```
(2000000, 12)
```

In [6]:

```
def clean_data(df):
    # Drop irrelevant columns
    df = df.drop(columns=['vm_id', 'timestamp'])

    # Handle missing values
    # Numerical columns: impute with median
    num_cols = df.select_dtypes(include=np.number).columns
    num_imputer = SimpleImputer(strategy='median')
    df[num_cols] = num_imputer.fit_transform(df[num_cols])

    # Categorical columns: impute with mode (excluding target column)
    cat_cols = df.select_dtypes(include='object').columns.drop('task_status')
    cat_imputer = SimpleImputer(strategy='most_frequent')
    df[cat_cols] = cat_imputer.fit_transform(df[cat_cols])

    # Remove duplicates
    df = df.drop_duplicates()

    return df

cleaned_df = clean_data(df)
cleaned_df.head()
```

Out[6]:

	cpu_usage	memory_usage	network_traffic	power_consumption	num_executed_instructions	exe
0	54.881350	78.950861	164.775973	287.808986	7527.0	
1	71.518937	29.901883	164.775973	362.273569	5348.0	
2	71.518937	92.709195	203.674847	231.467903	5483.0	
3	54.488318	88.100960	203.674847	195.639954	5876.0	
4	42.365480	88.100960	203.674847	359.451537	3361.0	

In [7]:

```python
scaler = StandardScaler()
numerical_columns = cleaned_df.select_dtypes(include=[np.number]).columns
cleaned_df[numerical_columns] = scaler.fit_transform(cleaned_df[numerical_columns
cleaned_df.head()
```

Out[7]:

	cpu_usage	memory_usage	network_traffic	power_consumption	num_executed_instructions	exe
0	0.168796	1.004291	-1.161974	0.261818	0.875470	
1	0.744872	-0.695900	-1.161974	0.777600	0.120372	
2	0.744872	1.481198	-1.027167	-0.128431	0.167154	
3	0.155187	1.321462	-1.027167	-0.376594	0.303342	
4	-0.264566	1.321462	-1.027167	0.758053	-0.568191	

In [8]:

```python
def preprocess_data(df, target='task_status'):
    # Separate features and target variable
    y = df[target]
    X = df.drop(columns=[target])

    # Create preprocessing pipeline
    numeric_features = X.select_dtypes(include=np.number).columns
    categorical_features = X.select_dtypes(include='object').columns

    preprocessor = ColumnTransformer(
        transformers=[
            ('num', StandardScaler(), numeric_features),
            ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features)
        ])

    # Train-test split first to prevent data leakage
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ranc

    # Apply preprocessing pipeline
    X_train = preprocessor.fit_transform(X_train)
    X_test = preprocessor.transform(X_test)

    return X_train, X_test, y_train, y_test

X_train, X_test, y_train, y_test = preprocess_data(cleaned_df)

# Reshape for LSTM [samples, timesteps, features]
X_train = X_train.reshape((X_train.shape[0], 1, X_train.shape[1]))
X_test = X_test.reshape((X_test.shape[0], 1, X_test.shape[1]))
```

In [7]:

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from pyswarm import pso  # PSO implementation
import numpy as np

# Example data shape (replace with your actual data loaders)
X_train = np.random.rand(100, 64, 128)  # e.g., (samples, timesteps, features)
y_train = np.random.randint(0, 2, 100)

X_val = np.random.rand(20, 64, 128)
y_val = np.random.randint(0, 2, 20)

# Objective function for PSO: return validation loss
def objective(params):
    lstm_units = int(params[0])
    dropout_rate = params[1]

    model = Sequential([
        LSTM(lstm_units, input_shape=(X_train.shape[1], X_train.shape[2])),
        Dropout(dropout_rate),
        Dense(1, activation='sigmoid')
    ])

    model.compile(optimizer=Adam(1e-4), loss='binary_crossentropy', metrics=['acc

    history = model.fit(X_train, y_train, epochs=3, batch_size=32, verbose=0, val
    val_loss = history.history['val_loss'][-1]

    return val_loss

# Bounds: [lstm_units, dropout_rate]
lb = [32, 0.1]
ub = [512, 0.5]

# Run PSO
best_params, _ = pso(objective, lb, ub, swarmsize=10, maxiter=5)
best_units, best_dropout = int(best_params[0]), best_params[1]


# Final model with best parameters
final_model = Sequential([
    LSTM(best_units, input_shape=(X_train.shape[1], X_train.shape[2])),
    Dropout(best_dropout),
    Dense(1, activation='sigmoid')
])

final_model.compile(optimizer=Adam(1e-4), loss='binary_crossentropy', metrics=['a

final_model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_va
/usr/local/lib/python3.10/dist-packages/keras/src/trainers/data_adapters/py_dat
aset_adapter.py:122: UserWarning: Your `PyDataset` class should call `super()._
_init__(**kwargs)` in its constructor. `**kwargs` can include `workers`, `use_m
ultiprocessing`, `max_queue_size`. Do not pass these arguments to `fit()`, as t
hey will be ignored.
  self._warn_if_super_not_called()
```

```
Epoch 1/5
4376/4376 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.9409 - auc: 0.9805 - lo
ss: 0.1692
Epoch 1: val_accuracy improved from -inf to 0.94618, saving model to /kaggle/wo
rking/140K_efficientnetb7_model.keras
4376/4376 ━━━━━━━━━━━━━━━━━━━━ 7290s 2s/step - accuracy: 0.9409 - auc: 0.9805 -
loss: 0.1692 - val_accuracy: 0.9462 - val_auc: 0.9894 - val_loss: 0.1571
Epoch 2/5
4376/4376 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.9700 - auc: 0.9944 - lo
ss: 0.0835
Epoch 2: val_accuracy did not improve from 0.94618
4376/4376 ━━━━━━━━━━━━━━━━━━━━ 7150s 2s/step - accuracy: 0.9701 - auc: 0.9944 -
loss: 0.0835 - val_accuracy: 0.9413 - val_auc: 0.9864 - val_loss: 0.1802
Epoch 3/5
4376/4376 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.9799 - auc: 0.9977 - lo
ss: 0.0538
Epoch 3: val_accuracy improved from 0.94618 to 0.96059, saving model to /kaggl
e/working/140K_efficientnetb7_model.keras
4376/4376 ━━━━━━━━━━━━━━━━━━━━ 7136s 2s/step - accuracy: 0.9799 - auc: 0.9977 -
loss: 0.0538 - val_accuracy: 0.9606 - val_auc: 0.9962 - val_loss: 0.1034
Epoch 4/5
4376/4376 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.9835 - auc: 0.9984 - lo
ss: 0.0439
Epoch 4: val_accuracy improved from 0.96059 to 0.97662, saving model to /kaggl
e/working/140K_efficientnetb7_model.keras
4376/4376 ━━━━━━━━━━━━━━━━━━━━ 7156s 2s/step - accuracy: 0.9835 - auc: 0.9984 -
loss: 0.0439 - val_accuracy: 0.9766 - val_auc: 0.9977 - val_loss: 0.0592
Epoch 5/5
4376/4376 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.9847 - auc: 0.9986 - lo
ss: 0.0416
Epoch 5: val_accuracy did not improve from 0.97662
4376/4376 ━━━━━━━━━━━━━━━━━━━━ 7154s 2s/step - accuracy: 0.9847 - auc: 0.9986 -
loss: 0.0416 - val_accuracy: 0.9725 - val_auc: 0.9961 - val_loss: 0.0733
```

In [8]:

```python
test_loss, test_accuracy, test_auc = model.evaluate(test_generator)

print(f"Test Loss: {test_loss}")
print(f"Test Accuracy: {test_accuracy:.2f}")
print(f"Test AUC: {test_auc:.2f}")
```

```
341/341 ━━━━━━━━━━━━━━━━━━━━ 109s 320ms/step - accuracy: 0.9263 - auc: 0.9823 -
loss: 0.2009
Test Loss: 0.21513307094573975
Test Accuracy: 0.92
Test AUC: 0.98
```