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

In [ ]:
```
pip install tensorflow keras-tuner scikit-learn pandas numpy matplotlib
```

In [37]:
```
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 [38]:
```python
df = pd.read_csv("/kaggle/input/cloud-computing-performance-metrics/vmCloud_dat

df.head()
```

```
/usr/local/lib/python3.10/dist-packages/pandas/io/formats/format.py:1458: Run
timeWarning: invalid value encountered in greater
  has_large_values = (abs_vals > 1e6).any()
/usr/local/lib/python3.10/dist-packages/pandas/io/formats/format.py:1459: Run
timeWarning: invalid value encountered in less
  has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals > 0)).any
()
/usr/local/lib/python3.10/dist-packages/pandas/io/formats/format.py:1459: Run
timeWarning: invalid value encountered in greater
  has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals > 0)).any
()
```

Out[38]:

	vm_id	timestamp	cpu_usage	memory_usage	network_traffic	power_consumption	nu
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 [39]:
```python
df.shape
```

Out[39]: (2000000, 12)

In [40]:
```python
# Handle missing values
df = df.fillna(method='ffill')


df.shape
```

```
<ipython-input-40-1c3ca3487565>:2: FutureWarning: DataFrame.fillna with 'meth
od' is deprecated and will raise in a future version. Use obj.ffill() or obj.
bfill() instead.
  df = df.fillna(method='ffill')
```

Out[40]: (2000000, 12)

In [41]:

```python
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[41]:

	cpu_usage	memory_usage	network_traffic	power_consumption	num_executed_instructions	e
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 [59]:

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

Out[59]:

	cpu_usage	memory_usage	network_traffic	power_consumption	num_executed_instructions
0	0.169442	1.004010	-1.160942	0.260706	0.874592
6	-0.215845	-0.954949	-0.245324	0.157970	1.387868
9	-0.403406	-1.163505	0.969151	0.917638	-0.698161
10	1.010887	-1.630568	1.476837	-0.529774	1.261715
12	0.236059	-1.651950	0.770904	-0.738856	1.658195

In [42]:

```python
# Convert target column to binary classification
def convert_target_column(df):
    df['task_status'] = df['task_status'].apply(lambda x: 0 if x in ['completed

    return df

cleaned_df = convert_target_column(cleaned_df)
cleaned_df.head()
```

Out[42]:

	cpu_usage	memory_usage	network_traffic	power_consumption	num_executed_instructions	e
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 [107]:

```python
cleaned_df = cleaned_df.drop(columns=['task_type','task_priority'])
cleaned_df.head()
```

Out[107]:

	cpu_usage	memory_usage	network_traffic	power_consumption	num_executed_instructions	e
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	

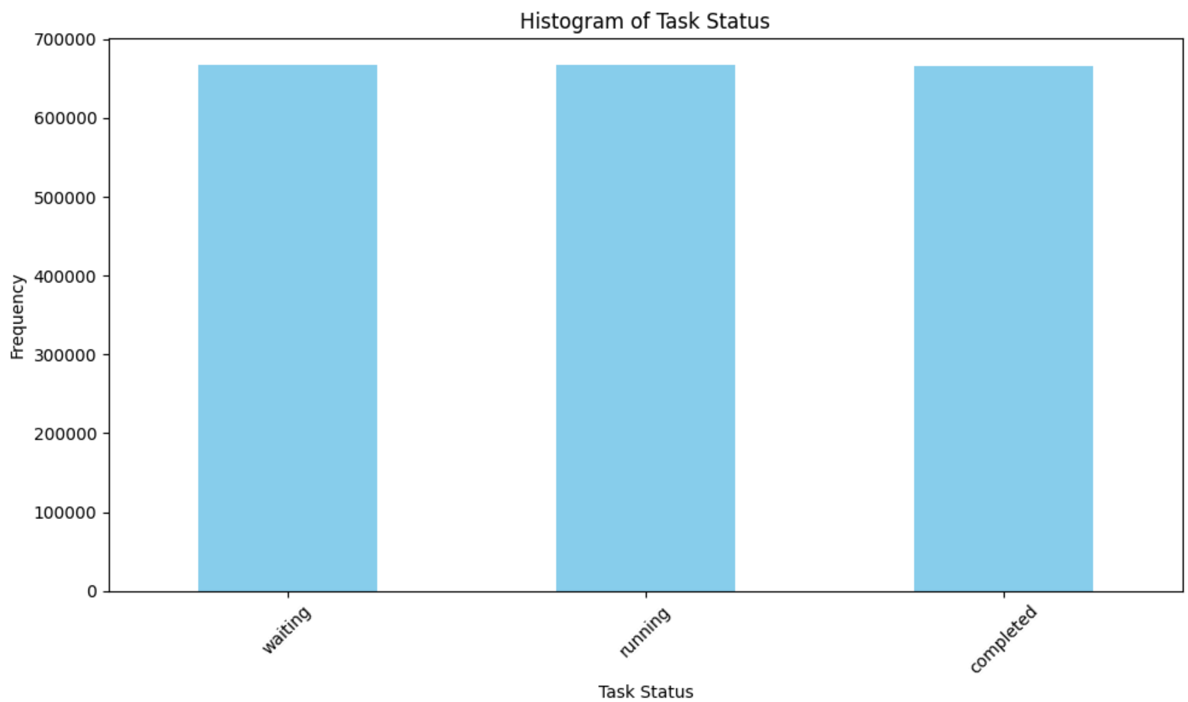
In [108]:
```
# Create a histogram for the 'task_status' column
plt.figure(figsize=(10, 6))  # Set the figure size

# Plot the histogram
df['task_status'].value_counts().plot(kind='bar', color='skyblue')

# Add title and labels
plt.title('Histogram of Task Status')
plt.xlabel('Task Status')
plt.ylabel('Frequency')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()  # Adjust layout to prevent label cutoff
plt.show()
```



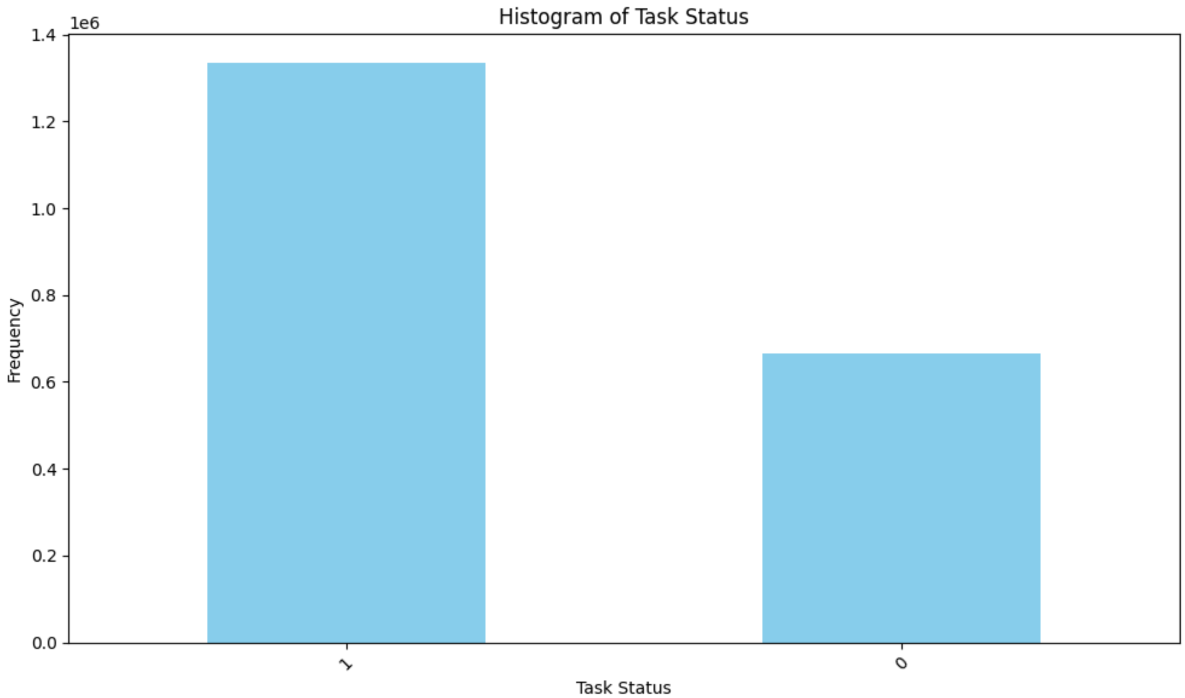
In [43]:
```
# Create a histogram for the 'task_status' column
plt.figure(figsize=(10, 6))  # Set the figure size

# Plot the histogram
cleaned_df['task_status'].value_counts().plot(kind='bar', color='skyblue')

# Add title and labels
plt.title('Histogram of Task Status')
plt.xlabel('Task Status')
plt.ylabel('Frequency')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()  # Adjust layout to prevent label cutoff
plt.show()
```



In [44]:
```
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_feature
        ])

    # 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, ra

    # 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 [47]:
```
# Enhanced LSTM Model
def create_lstm_model(input_shape):
    model = Sequential([
        LSTM(128, input_shape=input_shape, return_sequences=True),
        BatchNormalization(),
        Dropout(0.3),
        LSTM(64),
        BatchNormalization(),
        Dropout(0.2),
        Dense(1, activation='sigmoid')
    ])

    model.compile(
        optimizer=tf.keras.optimizers.Adam(learning_rate=0.0005),
        loss='binary_crossentropy',
        metrics=['accuracy', tf.keras.metrics.Precision(name='precision'), tf.k
    )
    return model
# Train the model
model = create_lstm_model((X_train.shape[1], X_train.shape[2]))
history = model.fit(
    X_train, y_train,
    validation_split=0.2,
    epochs=100,
    batch_size=64,

    verbose=1
)
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/rnn/rnn.py:204: User
Warning: Do not pass an `input_shape`/`input_dim` argument to a layer. When u
sing Sequential models, prefer using an `Input(shape)` object as the first la
yer in the model instead.
  super().__init__(**kwargs)
```

```
Epoch 1/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 125s 6ms/step - accuracy: 0.6519 - loss: 0.6
541 - precision: 0.6671 - recall: 0.9542 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 2/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6677 - loss: 0.6
361 - precision: 0.6677 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 3/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 119s 6ms/step - accuracy: 0.6671 - loss: 0.6
364 - precision: 0.6671 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 4/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 119s 6ms/step - accuracy: 0.6668 - loss: 0.6
365 - precision: 0.6668 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 5/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 120s 6ms/step - accuracy: 0.6672 - loss: 0.6
362 - precision: 0.6672 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 6/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 119s 6ms/step - accuracy: 0.6669 - loss: 0.6
364 - precision: 0.6669 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 7/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 120s 6ms/step - accuracy: 0.6676 - loss: 0.6
359 - precision: 0.6676 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 8/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6671 - loss: 0.6
362 - precision: 0.6671 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6364 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 9/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 119s 6ms/step - accuracy: 0.6672 - loss: 0.6
361 - precision: 0.6672 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6364 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 10/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 120s 6ms/step - accuracy: 0.6676 - loss: 0.6
359 - precision: 0.6676 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 11/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6678 - loss: 0.6
358 - precision: 0.6678 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6364 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 12/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 119s 6ms/step - accuracy: 0.6674 - loss: 0.6
360 - precision: 0.6674 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 13/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6673 - loss: 0.6
360 - precision: 0.6673 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 14/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6668 - loss: 0.6
364 - precision: 0.6668 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6364 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 15/100
```

```
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 119s 6ms/step - accuracy: 0.6675 - loss: 0.6
359 - precision: 0.6675 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 16/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6679 - loss: 0.6
356 - precision: 0.6679 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 17/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6668 - loss: 0.6
364 - precision: 0.6668 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 18/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6669 - loss: 0.6
364 - precision: 0.6669 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 19/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 119s 6ms/step - accuracy: 0.6669 - loss: 0.6
363 - precision: 0.6669 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 20/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 121s 6ms/step - accuracy: 0.6676 - loss: 0.6
359 - precision: 0.6676 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 21/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 119s 6ms/step - accuracy: 0.6679 - loss: 0.6
356 - precision: 0.6679 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 22/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6668 - loss: 0.6
363 - precision: 0.6668 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 23/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 119s 6ms/step - accuracy: 0.6663 - loss: 0.6
367 - precision: 0.6663 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 24/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6676 - loss: 0.6
359 - precision: 0.6676 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 27/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6673 - loss: 0.6
360 - precision: 0.6673 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 29/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6674 - loss: 0.6
359 - precision: 0.6674 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 31/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 119s 6ms/step - accuracy: 0.6670 - loss: 0.6
362 - precision: 0.6670 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 33/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6675 - loss: 0.6
359 - precision: 0.6675 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 34/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6672 - loss: 0.6
```

```
360 - precision: 0.6672 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 35/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6676 - loss: 0.6358 - precision: 0.6676 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 36/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6673 - loss: 0.6360 - precision: 0.6673 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 37/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6670 - loss: 0.6362 - precision: 0.6670 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 38/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6671 - loss: 0.6361 - precision: 0.6671 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 39/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6671 - loss: 0.6361 - precision: 0.6671 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 40/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6672 - loss: 0.6360 - precision: 0.6672 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 41/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6672 - loss: 0.6360 - precision: 0.6672 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 42/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6675 - loss: 0.6358 - precision: 0.6675 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6365 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 43/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6672 - loss: 0.6360 - precision: 0.6672 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 44/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6671 - loss: 0.6361 - precision: 0.6671 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 45/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6669 - loss: 0.6362 - precision: 0.6669 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 46/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6675 - loss: 0.6358 - precision: 0.6675 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 47/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6669 - loss: 0.6362 - precision: 0.6669 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss: 0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 48/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6664 - loss: 0.6365 - precision: 0.6664 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
```

```
0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 49/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6666 - loss: 0.6
364 - precision: 0.6666 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 50/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6675 - loss: 0.6
358 - precision: 0.6675 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 51/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6669 - loss: 0.6
362 - precision: 0.6669 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 52/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6667 - loss: 0.6
363 - precision: 0.6667 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 53/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6668 - loss: 0.6
362 - precision: 0.6668 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 54/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6672 - loss: 0.6
360 - precision: 0.6672 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 55/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6676 - loss: 0.6
357 - precision: 0.6676 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 56/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6669 - loss: 0.6
362 - precision: 0.6669 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 57/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6676 - loss: 0.6
357 - precision: 0.6676 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 58/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6673 - loss: 0.6
359 - precision: 0.6673 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 59/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6663 - loss: 0.6
366 - precision: 0.6663 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 60/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6671 - loss: 0.6
360 - precision: 0.6671 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 61/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6669 - loss: 0.6
361 - precision: 0.6669 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 62/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6674 - loss: 0.6
358 - precision: 0.6674 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6366 - val_precision: 0.6668 - val_recall: 1.0000
```

```
Epoch 63/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6669 - loss: 0.6
362 - precision: 0.6669 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 64/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6672 - loss: 0.6
359 - precision: 0.6672 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 65/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6667 - loss: 0.6
363 - precision: 0.6667 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 66/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6670 - loss: 0.6
361 - precision: 0.6670 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 67/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6670 - loss: 0.6
361 - precision: 0.6670 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 68/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6675 - loss: 0.6
357 - precision: 0.6675 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 69/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6672 - loss: 0.6
359 - precision: 0.6672 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 70/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6665 - loss: 0.6
364 - precision: 0.6665 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 71/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6673 - loss: 0.6
358 - precision: 0.6673 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 72/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6673 - loss: 0.6
359 - precision: 0.6673 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 73/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6671 - loss: 0.6
360 - precision: 0.6671 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 74/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6665 - loss: 0.6
364 - precision: 0.6665 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6366 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 75/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6678 - loss: 0.6
355 - precision: 0.6678 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 76/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6673 - loss: 0.6
358 - precision: 0.6673 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 77/100
```

```
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6667 - loss: 0.6
363 - precision: 0.6667 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 78/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6671 - loss: 0.6
359 - precision: 0.6671 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 79/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6671 - loss: 0.6
360 - precision: 0.6671 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 80/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6674 - loss: 0.6
357 - precision: 0.6674 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 81/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6665 - loss: 0.6
364 - precision: 0.6665 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6369 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 82/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6676 - loss: 0.6
356 - precision: 0.6676 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 83/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6672 - loss: 0.6
359 - precision: 0.6672 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 84/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6668 - loss: 0.6
362 - precision: 0.6668 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 85/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6674 - loss: 0.6
357 - precision: 0.6674 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 86/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6670 - loss: 0.6
360 - precision: 0.6670 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 87/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6676 - loss: 0.6
356 - precision: 0.6676 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 88/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6675 - loss: 0.6
356 - precision: 0.6675 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 89/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6679 - loss: 0.6
354 - precision: 0.6679 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6369 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 90/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6677 - loss: 0.6
354 - precision: 0.6677 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 91/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6679 - loss: 0.6
```

```
353 - precision: 0.6679 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 92/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 117s 6ms/step - accuracy: 0.6668 - loss: 0.6
361 - precision: 0.6668 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 93/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6679 - loss: 0.6
354 - precision: 0.6679 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 94/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6663 - loss: 0.6
364 - precision: 0.6663 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6369 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 95/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6676 - loss: 0.6
356 - precision: 0.6676 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6370 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 96/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6670 - loss: 0.6
360 - precision: 0.6670 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6367 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 97/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6668 - loss: 0.6
361 - precision: 0.6668 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 98/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6673 - loss: 0.6
358 - precision: 0.6673 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 99/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6668 - loss: 0.6
361 - precision: 0.6668 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6369 - val_precision: 0.6668 - val_recall: 1.0000
Epoch 100/100
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 118s 6ms/step - accuracy: 0.6678 - loss: 0.6
354 - precision: 0.6678 - recall: 1.0000 - val_accuracy: 0.6668 - val_loss:
0.6368 - val_precision: 0.6668 - val_recall: 1.0000
```

In [48]:
```python
# Predictions on test set
y_pred = (model.predict(X_test) > 0.5).astype(int)

print("Classification Report:")
print(classification_report(y_test, y_pred))

print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

```
12500/12500 ━━━━━━━━━━━━━━━━━━━━ 19s 2ms/step
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00    133160
           1       0.67      1.00      0.80    266840

    accuracy                           0.67    400000
   macro avg       0.33      0.50      0.40    400000
weighted avg       0.45      0.67      0.53    400000


Confusion Matrix:
[[     0 133160]
 [     0 266840]]

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:13
44: UndefinedMetricWarning: Precision and F-score are ill-defined and being s
et to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:13
44: UndefinedMetricWarning: Precision and F-score are ill-defined and being s
et to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:13
44: UndefinedMetricWarning: Precision and F-score are ill-defined and being s
et to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [50]:
```python
def plot_metrics():
    plt.figure(figsize=(15, 5))

    # Accuracy plot
    plt.subplot(1, 2, 1)
    plt.plot(history.history['accuracy'], label='Train')
    plt.plot(history.history['val_accuracy'], label='Validation')
    plt.title('Model Accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend()

    # Loss plot
    plt.subplot(1, 2, 2)
    plt.plot(history.history['loss'], label='Train')
    plt.plot(history.history['val_loss'], label='Validation')
    plt.title('Model Loss')
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend()

    plt.tight_layout()
    plt.show()

plot_metrics()
```