Model testing   
  
**LSTM-based architecture** – Captures time-series patterns in network traffic.

1. **Stacked Layers** – Deep model with 2 LSTM layers and dense layers for better feature learning.
2. **Regularization** – L2 and Dropout added to reduce overfitting.
3. **Balanced Training** – Used SMOTE and class weights to handle class imbalance.
4. **High Accuracy** – Achieved ~97.8% accuracy with strong precision/recall across all classes.
5. **Real-time Ready** – Model + scaler saved, ready to process live input.
6. **Supports 7 Attack Types** –
   * Normal Traffic
   * Bots
   * Brute Force
   * DDoS
   * DoS
   * Port Scanning
   * Web Attacks
7. **Training Optimizations** – Used EarlyStopping and ReduceLROnPlateau for smarter training.

**Attack Types Your Model Can Detect:**['Bots', 'Brute Force', 'DDoS', 'DoS', 'Normal Traffic', 'Port Scanning', 'Web Attacks']

**Feature Set**

[

'Destination Port', 'Flow Duration', 'Total Fwd Packets', 'Total Length of Fwd Packets',

'Fwd Packet Length Max', 'Fwd Packet Length Min', 'Fwd Packet Length Mean', 'Fwd Packet Length Std',

'Bwd Packet Length Max', 'Bwd Packet Length Min', 'Bwd Packet Length Mean', 'Bwd Packet Length Std',

'Flow Bytes/s', 'Flow Packets/s', 'Flow IAT Mean', 'Flow IAT Std', 'Flow IAT Max', 'Flow IAT Min',

'Fwd IAT Total', 'Fwd IAT Mean', 'Fwd IAT Std', 'Fwd IAT Max', 'Fwd IAT Min',

'Bwd IAT Total', 'Bwd IAT Mean', 'Bwd IAT Std', 'Bwd IAT Max', 'Bwd IAT Min',

'Fwd Header Length', 'Bwd Header Length', 'Fwd Packets/s', 'Bwd Packets/s',

'Min Packet Length', 'Max Packet Length', 'Packet Length Mean', 'Packet Length Std',

'Packet Length Variance', 'FIN Flag Count', 'PSH Flag Count', 'ACK Flag Count',

'Average Packet Size', 'Subflow Fwd Bytes', 'Init\_Win\_bytes\_forward', 'Init\_Win\_bytes\_backward',

'act\_data\_pkt\_fwd', 'min\_seg\_size\_forward', 'Active Mean', 'Active Max', 'Active Min',

'Idle Mean', 'Idle Max', 'Idle Min'

]

**CICFlowmeter-V4.0**

**Ethernet traffic Bi-flow generator and analyze**

**Should use above kinds of software to generate data into csv or json formats as needed**

**Real-World Data Format (from firewalls, IDS, NetFlow, etc.)**

1. Real-world network data is usually flow-based, where each record represents a communication session.
2. Each record will contain numerical metrics related to packet sizes, timings, flags, ports, etc.
3. The format is often CSV, JSON, or log-based.

Example flow record (JSON-like):

{

"Destination Port": 443,

"Flow Duration": 1030000,

"Total Fwd Packets": 12,

"Fwd Packet Length Max": 200,

...

}

**How to Use Real-World Data with Your Model**

1. **Collect real-time traffic**
   * Use tools like Zeek, Suricata, NetFlow, or Python with scapy or pyshark.
2. **Extract the 52 features**
   * Use custom scripts or CICFlowMeter to generate the same set of features.
3. **Preprocess the data**
   * Ensure the features are in the same order as used in training.
   * Load your saved scaler (scaler.pkl) and apply it to normalize the values.
   * Reshape the data to 3D: (1, 1, 52) for a single sample.
4. **Run inference**
   * Load the model using tf.keras.models.load\_model('model.h5').
   * Load the scaler and label encoder.
   * Predict and decode the predicted class.