1. Introduction

The increasing complexity of cyberattacks demands predictive intelligence beyond conventional rule-based intrusion detection.
This research focuses on an **AI-driven cybersecurity threat prediction model** trained on **static datasets** (CICIDS-2018 / UNSW-NB15).
The model processes uploaded network data, performs feature engineering and ML/DL-based classification, and outputs detailed threat analysis via an interactive dashboard.

2. System Workflow Overview

🖥️ A. Frontend Module

- **User Uploads Data File:**
  Users provide static dataset files (e.g., CSV network traffic logs).
- **Input Validation:**
  System verifies file type, format, and schema consistency (checking headers, missing fields).
- **Send Data to Backend:**
  Upon validation, data is securely transmitted to the backend server for analysis via API (HTTP/Flask/Node).

⚙️ B. Backend Module

- **Data Preprocessing:**
  - Handle missing values, normalize numerical attributes.
  - Encode categorical values.
  - Remove duplicates and noise.
- **Database Interaction:**
  - Store cleaned data in SQL/NoSQL database (e.g., PostgreSQL, MongoDB).
  - Maintain logs for future retraining or auditing
- **Feature Extraction and Selection:**
  - Compute statistical flow features (e.g., packet rates, duration, flags).
  - Apply feature selection (Mutual Information, PCA, or Random Forest importance).
- **AI Model Training (Offline):**
  - Use ML/DL models (Random Forest, XGBoost, CNN-BiLSTM).

    - Train using CICIDS-2018 or UNSW-NB15 labeled data.
    - Evaluate on static validation/test splits.
- **AI Agent: Static Analysis and Classification:**
    - For uploaded data, the trained model classifies each record as *Normal* or *Attack Type*.
    - Although termed "real-time" in the diagram, here it performs **batch-based static analysis** (offline prediction).

🔍 C. Prediction and Decision Module

- **Model Predicts Potential Threats:**
  The trained model infers the class probabilities for each flow or session.
- **AI Agent Interprets Result:**
  Applies thresholding or rule logic to categorize severity (Low / Medium / High).
- **Decision Logic:**
    - If data is labeled as *Secure*, mark as benign.
    - If *Insecure*, flag for reporting and visualization.

📊 D. Output and Visualization Module

- **Dashboard Displays Results:**
  Graphical interface presents detection summary, accuracy scores, and attack distributions.
  (Built using tools like Dash, Streamlit, or Flask-Bootstrap.)
- **Threat Report Generation:**
  Automatically produces detailed CSV/PDF report summarizing attacks, IPs, timestamps, and confidence scores.
- **Final Status:**
  Displays final classification summary (Safe / Threat Detected) to the user.

Model Description

Model	Type	Reason for Use
Random Forest	Ensemble ML	Baseline interpretability and feature ranking
XGBoost	Gradient Boosting	High accuracy for tabular data
MLP	Deep Learning	Learns nonlinear feature relations
CNN-BiLSTM (Hybrid)	Deep Learning	Captures spatial + contextual dependencies for complex attacks

Model Architecture (CNN-BiLSTM Hybrid)

Input Layer: Normalized feature vector from preprocessed dataset
Convolutional Layers (CNN): Detect local correlations between traffic features
BiLSTM Layers: Capture sequential context of feature interactions
Dense Layer: Nonlinear transformations for classification
Output Layer: Softmax activation → multi-class attack probabilities

Training Setup:

- Optimizer: Adam
- Learning Rate: 0.001
- Batch Size: 64
- Epochs: 50–80
- Loss Function: Categorical Cross-Entropy

- Validation: 15% hold-out
- Regularization: Dropout (0.4), Early Stopping

Data Preprocessing & Feature Engineering

Step	Technique	Purpose
Missing Values	Mean/Median Imputation	Completeness
Normalization	Min-Max or Z-Score	Scale uniformity
Label Encoding	One-Hot / Ordinal	Numeric representation
Feature Selection	RF importance / PCA	Reduce noise & redundancy
Data Split	70/15/15 (Train/Val/Test)	Fair evaluation

Model Evaluation Metrics

- **Accuracy:** Correct predictions over total samples
- **Precision & Recall:** To balance false alarms and missed threats
- **F1-Score:** Combined metric for imbalanced data
- **ROC-AUC:** Overall classification robustness
- **Confusion Matrix:** Attack-wise performance visualization

Expected Results (Offline Static Data)

Model	Accuracy	F1-Score	ROC-AUC
Random Forest	95–96%	0.94	0.95
XGBoost	97%	0.96	0.97
CNN-BiLSTM	98–99%	0.98	0.99

Explainability and Threat Interpretation

- **SHAP (SHapley Additive exPlanations):** Identify top contributing features (e.g., Flow Bytes/s, Packet Length Mean).
- **Feature Importance Visualization:** Helps security analysts understand "why" an alert was triggered.
- **Confidence Scores:** Provide probability-based risk levels (e.g., 0.98 → High Threat).

CONCLUSION

The proposed AI-driven model effectively classifies cyber threats using static network data through a structured workflow comprising data ingestion, preprocessing, model training, and dashboard visualization.
With a hybrid CNN-BiLSTM model achieving near-99% accuracy, the system demonstrates that AI can significantly enhance cybersecurity prediction in static environments — forming a strong foundation for future real-time intelligent IDS systems.