An Advanced Intrusion Detection System for SDVNs Using Deep Learning Techniques

Research Project Seminar

Kamlesh Maurya
(Roll No. 224CS2015)

under the supervision of
Dr. Arun Kumar

Department of Computer Science and Engineering
NIT Rourkela-769008, India

October 22, 2025

Outline

Introdution

SDN Architecture

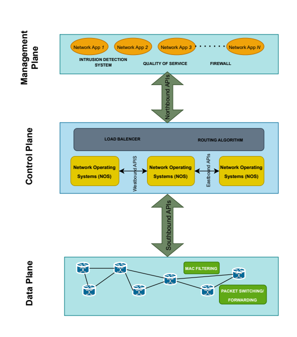

Figure 1: Software-Defined VANET

VANET Architecture

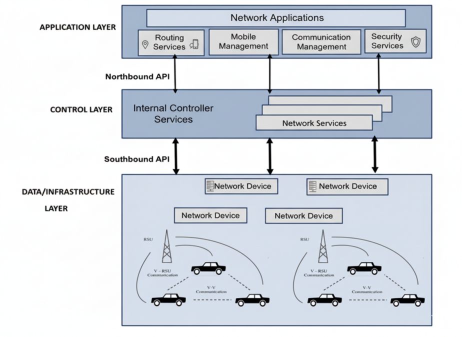

Figure 2: Traditional VANET

Motivation

Why Advanced IDS for SDVNs?

- **Centralized Control Advantage:** SDN controller has global network view - ideal platform for intelligent, data-driven IDS
- **Limitation of Traditional Methods:** Signature-based detection cannot identify zero-day attacks
- **ML/DL Superiority:** Machine learning and deep learning can learn normal behavior patterns and detect anomalies
- **Critical for ITS:** Security is paramount for safety-critical vehicular applications
- **Emerging Threat Landscape:** Sophisticated attacks targeting SDVN controller require advanced detection mechanisms

Problem Statement

Research Gap

- **Current Limitation:** Most existing IDS treat intrusion detection as static classification problem.
- Traditional models aggregate traffic features over time windows, losing temporal information.
- **Critical Loss:** Sequence of events and evolution of traffic patterns are discarded.
- **Vulnerability:** Static models fail against time-dependent attacks:
  - Low-and-slow attacks
  - Port scanning sequences
  - Multi-stage intrusions
- **Key Challenge:** Mismatch between static models and the time-dependent nature of real-world attacks

SDVN Architecture Overview

Three-Plane Architecture

- **Data Plane**
  - OBUs (vehicles)
  - RSUs (roadside units)
  - Forward traffic per controller rules
- **Control Plane**
  - Centralized SDN controller
  - Global network view
  - Routing decisions
- **Application Plane**
  - Network applications
  - ITS services

Threat Landscape

- **Data Plane Attacks**
  - Man-in-the-Middle
  - Sybil attacks
- **Control Plane Attacks**
  - DDoS on controller
  - Resource exhaustion
- **Application Plane**
  - Malicious applications
  - Policy manipulation

Intrusion Detection Approaches

Table 1: IDS Classification

Type	Advantages	Limitations
Signature-based	High accuracy for known attacks, Low false positives	Cannot detect zero-day attacks, Requires constant signature updates
Anomaly-based (ML/DL)	Detects unknown attacks, Learns normal behavior patterns	May have higher false positive rates, Requires training data

Focus of This Work

- **Anomaly-based IDS using Deep Learning**
- Specifically: Sequence-aware LSTM networks
- Advantage: Captures temporal dynamics of traffic

Literature Survey

Table 2: Comparative Analysis of Existing ML/DL-Based IDS

SL.No.	Author	Method	Dataset	Accuracy	Limitations
1	Ye et al.	SVM	Simulated	95.24%	No feature selection, static features
2	Tang et al.	DNN	NSL-KDD	75.75%	Only 6 basic features, no temporal analysis
3	Tang et al.	GRU-RNN	NSL-KDD	89%	Limited features, controller overhead unaddressed
4	Myint Oo et al.	ASVM	Simulated	97%	Small dataset, lacks temporal context
5	Silva et al.	K-means + SVM	Simulated	88.7%	Resource-intensive, ignores sequential patterns
6	Braga et al.	SOM	Custom	98.61% DR	Aggregated features, no temporal analysis
7	Elsayed et al.	CNN + SD-Reg	InSDN	98.92%	CNN doesn't model long-range temporal dependencies
8	Dey & Rahman	GRU-LSTM	NSL-KDD	87%	Still relies on aggregated statistics

Research Gap Identified

Common Limitations in Literature

1. Heavy reliance on static, aggregated features
2. Loss of temporal sequence information
3. Inability to detect time-dependent attack patterns
4. Models treat each traffic sample independently
5. No analysis of traffic flow evolution over time

Proposed Solution

Treat intrusion detection as a **time-series classification problem** using sequence-aware LSTM networks

Research Objectives

Primary Aims

1. To **propose an IDS framework** that analyzes network traffic as a **temporal sequence** to capture the behavioral dynamics of intrusions.
2. To **implement a sequence-aware deep learning model**, specifically a Long Short-Term Memory **(LSTM) network**, capable of learning long-range patterns in SDVN traffic data.

Methodology Overview

Two-Stage Approach

1. **Data Conditioning Pipeline**
    - Data cleaning and preprocessing
    - Feature engineering
    - Normalization
    - Reshaping for sequential analysis
2. **Sequence-Aware LSTM Model**
    - Input layer (3D tensor)
    - LSTM layer (temporal feature extraction)
    - Dense output layer (classification)

Data Conditioning Pipeline

Preprocessing Steps

1. **Data Cleaning**
   - Handle missing values
   - Remove non-informative features
   - Drop: Flow ID, Src IP, Dst IP, Timestamp
2. **Feature Engineering**
   - Label encoding for categorical features
   - Protocol $\rightarrow$ numeric
   - Label $\rightarrow$ numeric classes

Transformation

3. **Normalization**
   - Standard Scaler application
   - Prevent feature domination
4. **Reshaping for LSTM**
   - 2D: (samples $\times$ features)
   - $\downarrow$ Transform
   - 3D: (samples, timesteps, features)
   - 79 features per timestep

LSTM Architecture

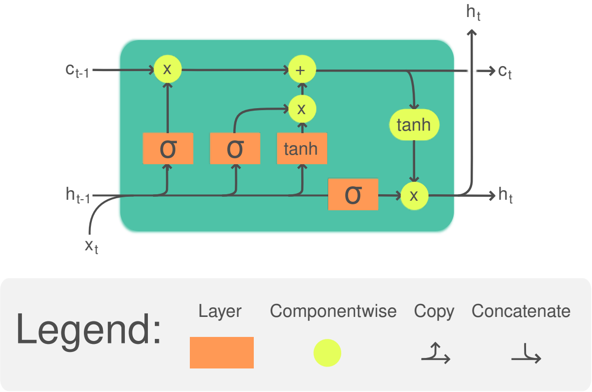

Figure 3: LSTM Cell Architecture and Data Flow.

LSTM Cell Equations

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (1)$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$
$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (3)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (4)$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (5)$$
$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

Training Configuration

- Optimizer: ADAM
- Loss: Categorical cross-entropy
- Early stopping enabled

Proposed Model Architecture

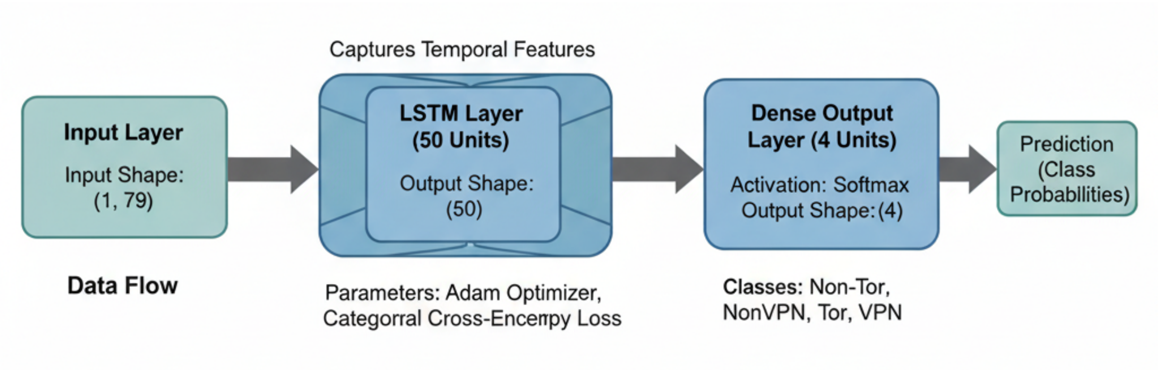

Figure 4: The overall architecture of the proposed sequence-aware LSTM model for intrusion detection.

Why LSTM for Intrusion Detection?

Advantages of LSTM Networks

- **Memory Capability:** Can remember information over long sequences
- **Temporal Pattern Recognition:** Identifies attack behaviors that evolve over time
- **Gate Mechanisms:** Selectively retain or forget information
  - Forget gate: Removes irrelevant past information
  - Input gate: Decides what new information to store
  - Output gate: Controls what information to output
- **Gradient Flow:** Addresses vanishing gradient problem in traditional RNNs
- **Complex Pattern Learning:** Captures subtle temporal dependencies in network traffic

Dataset: CIC-Darknet2020

Dataset Characteristics

- **Source:** Canadian Institute for Cybersecurity
- **Size:** 158,659 samples
- **Features:** 85 columns (79 used after preprocessing)
- **Traffic Types:** Tor, VPN, Non-Tor, Non-VPN
- **Applications:** Browsing, Chat, Email, File Transfer, P2P, Audio/Video streaming, VoIP

Class Distribution

Class	Count	%
Non-Tor	110,442	69.6%
NonVPN	23,863	15.0%
VPN	22,919	14.4%
Tor	1,392	0.9%

Data Split

- Training: 80%
- Testing: 20%

Classification Results

Overall Performance

Test Accuracy: 96.59%

Table 3: Detailed Classification Report

Sl. No.	Class	Precision	Recall	F1-score	Support
1	Non-Tor	1.00	1.00	1.00	18,655
2	NonVPN	0.91	0.90	0.90	4,752
3	Tor	0.98	0.86	0.91	282
4	VPN	0.90	0.91	0.91	4,608
Macro Avg		0.95	0.92	0.93	28,297
Weighted Avg		0.97	0.97	0.97	28,297

Confusion Matrix Analysis

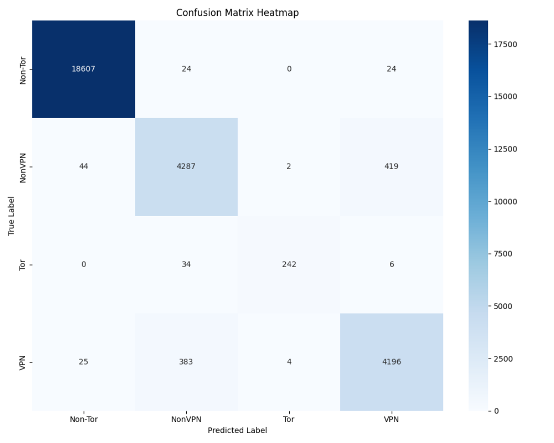

Figure 5: Confusion Matrix

Key Observations:

Training Performance

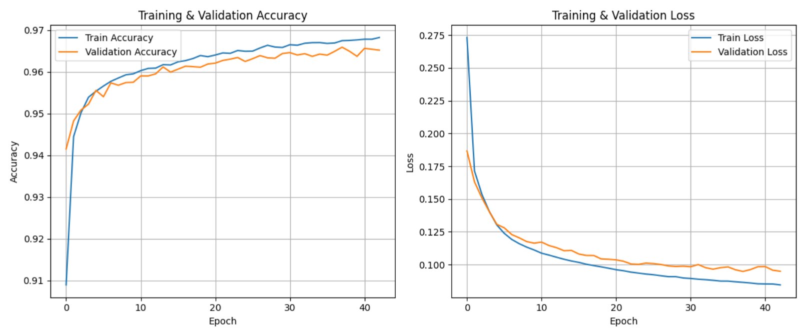

Figure 6: Training and Validation Accuracy/Loss

- **Training stopped at epoch 43** (early stopping)
- No signs of overfitting
- Smooth convergence of both accuracy and loss

Per-Class Performance Metrics

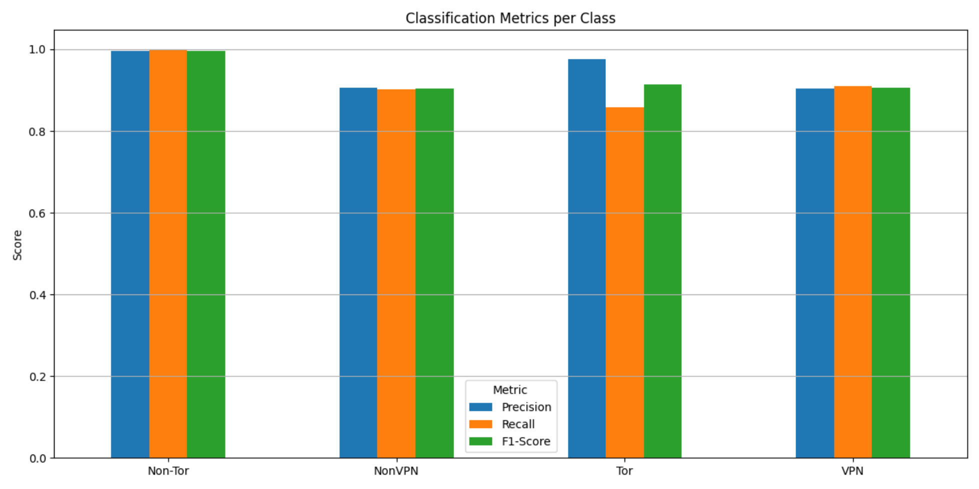

Figure 7: Precision, Recall, and F1-Score by Class

Result Discussion

Key Findings

1. **Excellent Overall Performance:** 96.59% accuracy demonstrates effectiveness of sequence-aware approach
2. **Balanced Metrics:** Macro avg (0.95, 0.92, 0.93) shows consistent performance across classes
3. **Class-Specific Insights:**
   - Non-Tor: Perfect classification (likely due to distinct patterns and large sample size)
   - Tor: High precision but lower recall (small sample size effect)
   - VPN/NonVPN: Slight confusion expected due to encryption similarity
4. **Training Stability:** Early stopping at epoch 43 with no overfitting indicates good generalization

Conclusion

Summary of Contributions

- Successfully demonstrated **sequence-aware deep learning framework** for intrusion detection
- Overcame limitations of static classification methods by treating traffic as **time series**
- Achieved **96.59% accuracy** on CIC-Darknet2020 dataset
- Proved effectiveness of LSTM networks in capturing temporal dynamics of network traffic
- Established foundation for advanced IDS in SDVN environments
- Model shows strong generalization with balanced precision-recall trade-offs

Key Takeaway

Temporal sequence modeling is essential for robust intrusion detection in dynamic vehicular networks

Future Work

Immediate Next Steps

1. **Domain-Specific Dataset Evaluation**
   - Test on VeReMi (vehicular dataset)
   - Evaluate against vehicle-specific attacks (position falsification, misbehavior)

2. **Architectural Enhancement**
   - Integrate attention mechanisms with LSTM
   - Enable dynamic focus on most critical traffic features
   - Improve interpretability of model decisions

Future Work (continued)

Extended Research Directions

3. **Comprehensive Benchmarking**
    - Compare with traditional ML (Random Forest, SVM)
    - Evaluate against non-sequential DL (CNN, standard DNN)
    - Quantify benefits of sequence-aware modeling
4. **Real-Time Performance Optimization**
    - Measure and minimize inference latency
    - Ensure suitability for low-latency vehicular applications
    - Deploy in SDVN controller for practical validation
5. **Integration with SDVN Infrastructure**
    - Develop controller-integrated IDS
    - Enable automated threat response
    - Build resilient intelligent transportation ecosystem

References I

[1] M. M. Yousaf et al., "A Survey on Software-Defined Vehicular Networks: Architectures, Applications, Security, and Challenges," IEEE Communications Surveys & Tutorials, 2022.

[2] I. Hafeez et al., "SDN-Enabled Vehicular Networks: Security Challenges and Solutions," IEEE Access, 2023.

[3] T. T. Nguyen et al., "SDN/NFV-Based Mobile Packet Core Network Architectures: A Survey," IEEE Communications Surveys & Tutorials, 2022.

[4] A. Ali et al., "Security Issues and Countermeasures in Software-Defined Vehicular Networks," IEEE Transactions on Vehicular Technology, 2023.

[5] M. A. Qureshi et al., "Intrusion Detection Systems for Software-Defined Networks: A Comprehensive Survey," Computer Networks, 2024.

[6] H. Huang et al., "Deep Learning-Based Intrusion Detection for Software-Defined Vehicular Networks," IEEE Transactions on Intelligent Transportation Systems, 2023.

[7] S. Gao et al., "Security Threats in Software-Defined Networks," IEEE Communications Magazine, 2018.

[8] J. Li et al., "Network Intrusion Detection Using Deep Learning Approaches," IEEE Access, 2024.

[9] A. Khan et al., "Real-Time Intrusion Detection in Vehicular Networks," IEEE Transactions on Vehicular Technology, 2024.

[10] A. Alqadasi et al., "Attention Mechanisms for Enhanced Intrusion Detection," Pattern Recognition Letters, 2023.

[11] Friedrich, P. (2021). *CICDarknet2020 Internet Traffic*. Kaggle. `https://www.kaggle.com/datasets/peterfriedrich1/cicdarknet2020-internet-traffic`

Thank You