IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS

9–13 June 2024 // Denver, CO, USA

Scaling the Peaks of Global Communications

IEEE INTERNATIONAL CONFERENCE ON
COMMUNICATIONS

DeepIDPS: An Adaptive DRL-based Intrusion Detection and Prevention System for SDN

Nadia Niknami (presenter), and Jie Wu

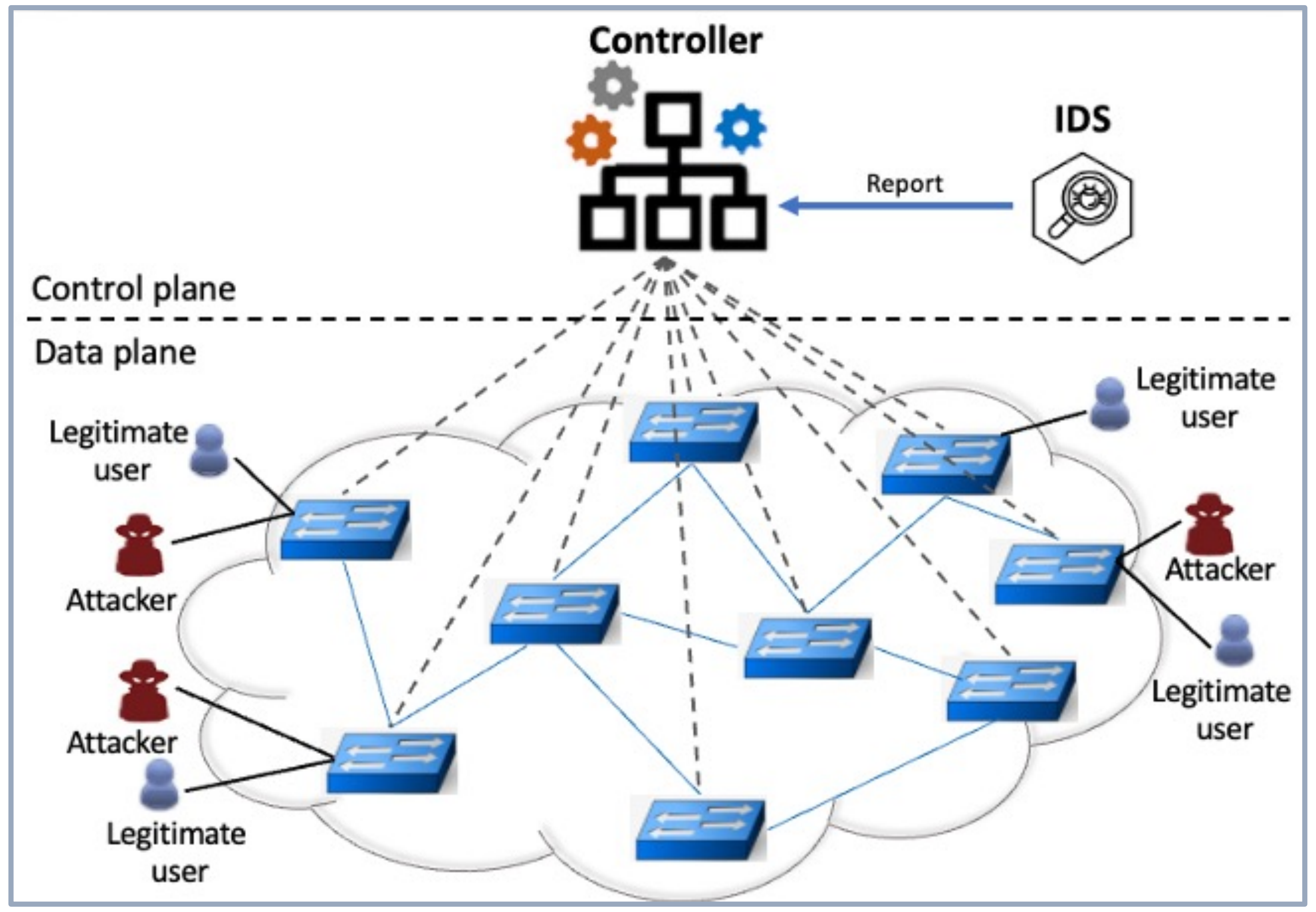
Dept. of Computer and Information Sciences
Temple University, Philadelphia, PA

Content

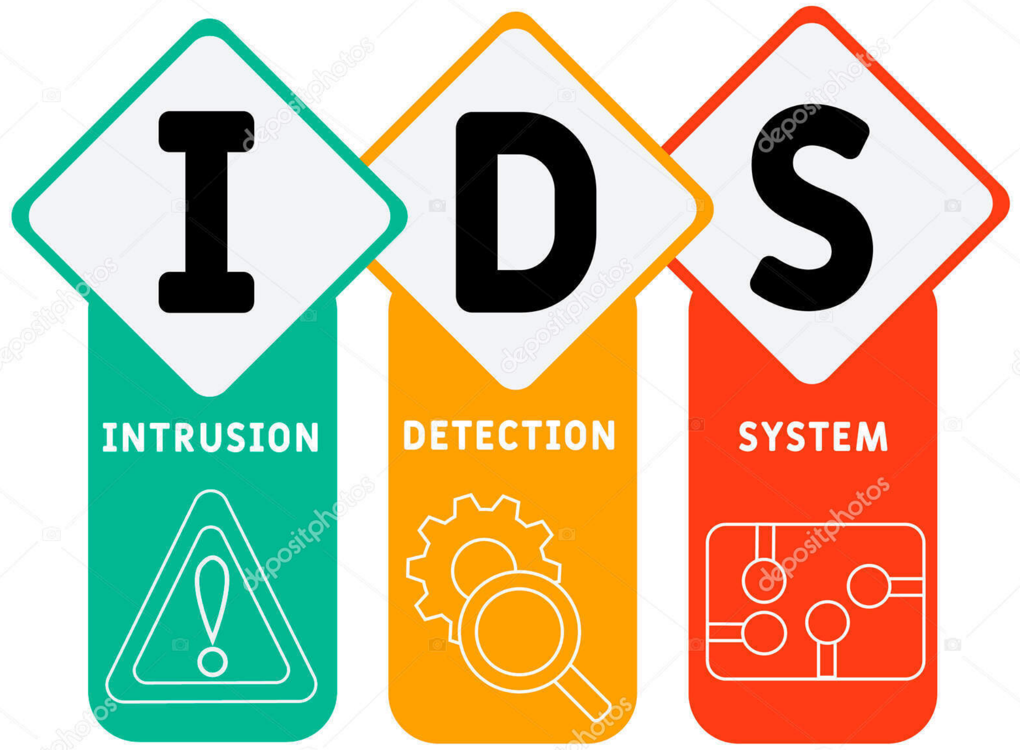
- Introduction
- Challenges and Motivation
- The proposed approach: DeepIDPS
- Experiment
- Conclusions

Introduction

- SDN technology
  - Data plane: processing and delivery of packets with local forwarding state
  - Control plane: computing the forwarding state in routers
  - A network in which the control plane is physically separate from the data plane.
  - A single (logically centralized) control plane controls several forwarding devices
  - Challenges: Vulnerability to attacks across various planes



- Intrusion Detection System
  - Monitoring network traffic and generating alerts
  - IDS inspects all network activity and identifies suspicious patterns that may indicate a network attack from someone attempting to compromise a system.



Content

- Introduction
- Challenges and Motivation
- The proposed approach: DeepIDPS
- Experiment
- Conclusions

Challenges and Motivation

- High accuracy (both correct positive and negative predictions)
- Low FPR (negative instances that are misclassified as positive)

- The effectiveness of an IDS heavily relies on the quality of the feature selection algorithms employed.

- IDSs provides some alerts and the admin must effectively perceive the current state of network and make the best decision for reacting to malicious traffic.

Solution:

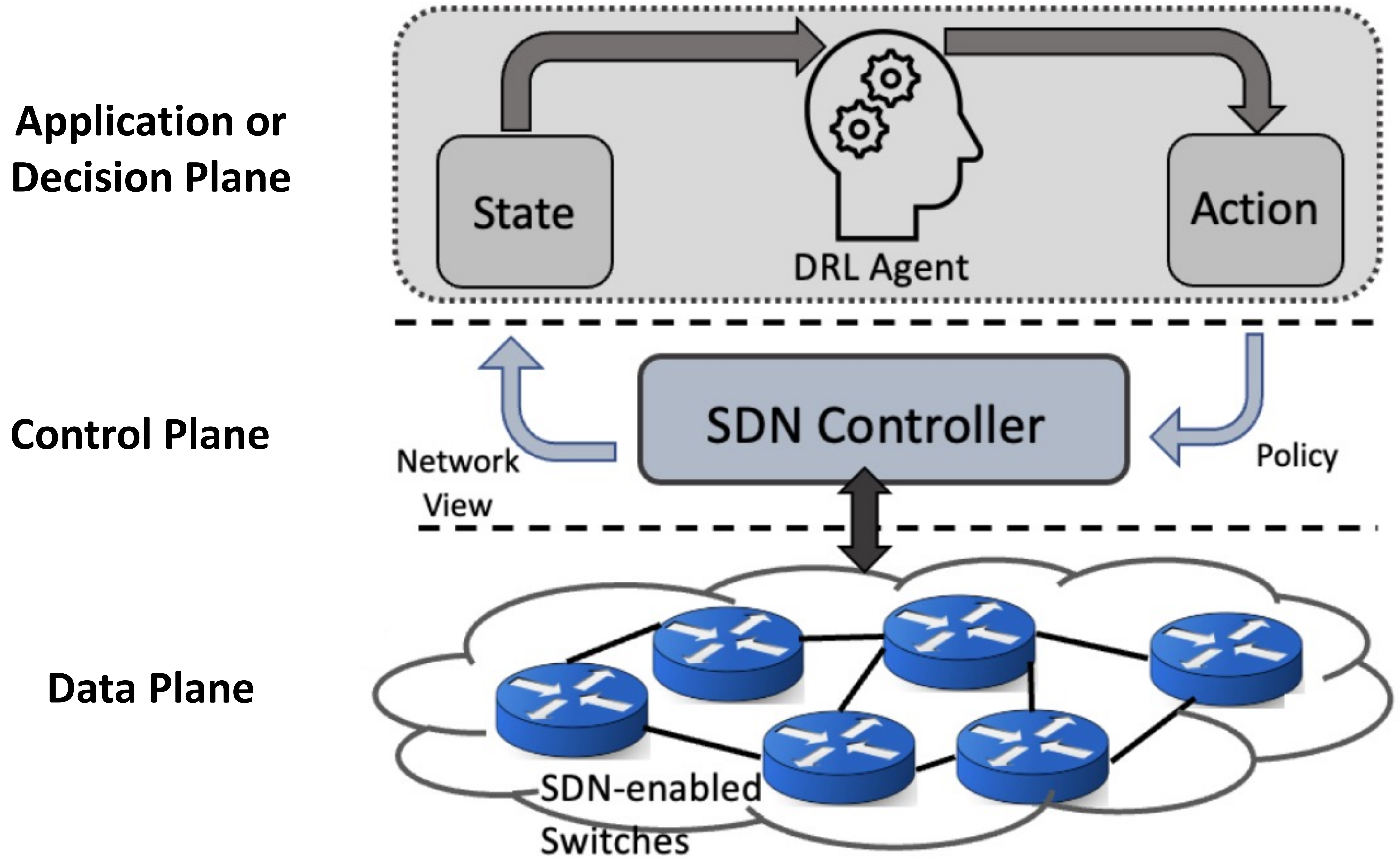
- The CNN part focuses on spatial features, essential for identifying <u>specific types of attacks</u>,
- The LSTM part captures time-related features in network traffic.
- RL with deep neural networks: improving efficiency and effectiveness of IDSs.

Content

- Introduction
- Challenges and Motivation
- The proposed approach: DeepIDPS
- Experiment
- Conclusions

The Proposed Approach: DeepIDPS

- CNN: Extracts spatial features from network traffic.
- LSTM: Captures temporal features.
- *Attention Mechanism:* Focuses on critical features.
- RL Agent: Interacts with the SDN environment to update security policies



DeepIDPS: CNN and LSTM

- **Parallel Architecture:**

CNN and LSTM operate independently on the input data.

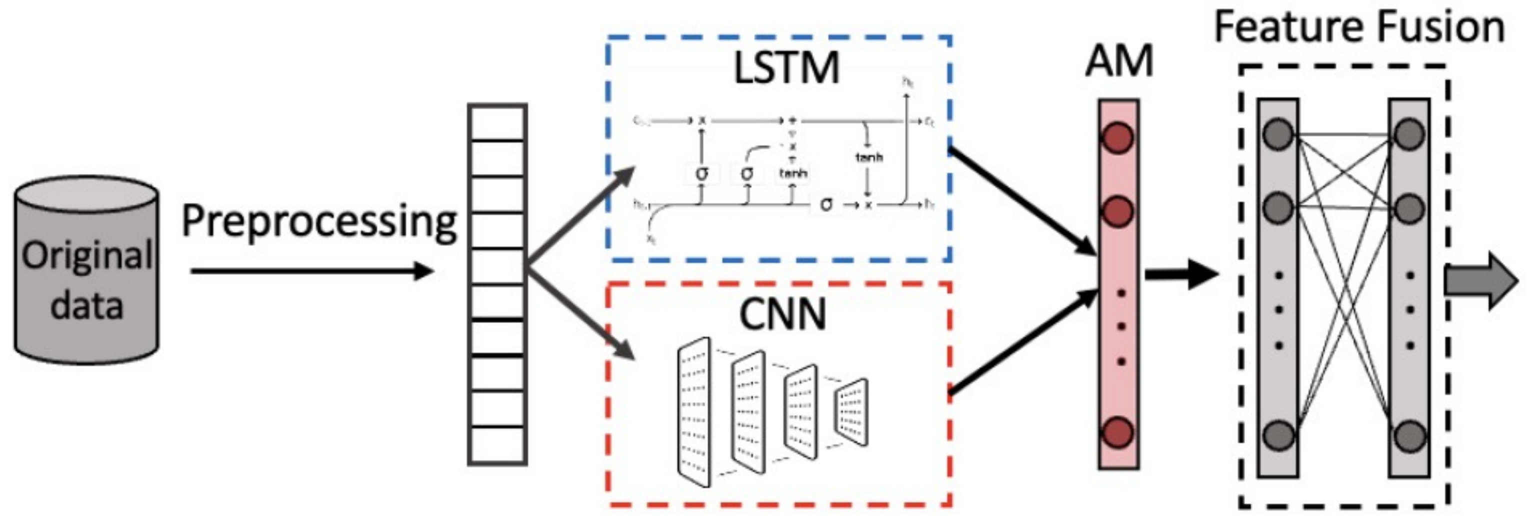


Fig. 2: Feature Fusion.

- **Sequence Architecture:**

CNN output is used as the input to the LSTM. This enables the LSTM to learn additional features from the input data that have already been extracted by the CNN

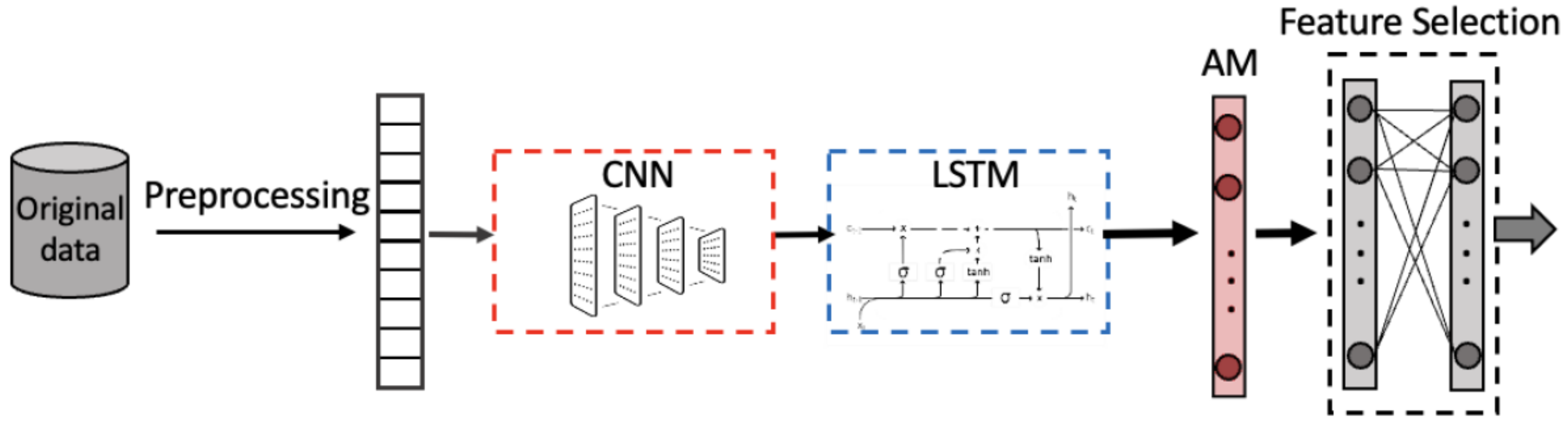


Fig. 3: Feature Selection.

- **Attention Mechanism (AM):**

Weigh the importance of different features extracted by the CNN and LSTM

DeepIDPS: Reinforcement Learning

State:

- D: the detection state of the existing traffic in the network
- M: the level of harm caused by malicious traffic.

$$S = (D, M)$$

Actions:

- a1: BlockIP-30secs (Drop all incoming packets with the attacker's IP address for 30 seconds)
- a2: LimitRate-25% (Reduce the rate of incoming packets from attacker's IP address by 25%)
- a3: ReRoute: Redirect the attack traffic flows,
- a4: DoNothing: No action.

Reward Function:

$$\mathcal{R}(s,a) = \alpha * D + \beta * U + \gamma * (1/T) + \omega * (1 - F) + \zeta * M,$$

- D is the detection accuracy
- U is the resource utilization
- T is the response time
- F is the false positive rate
- M is the attack mitigation

DeepIDPS: Transition States

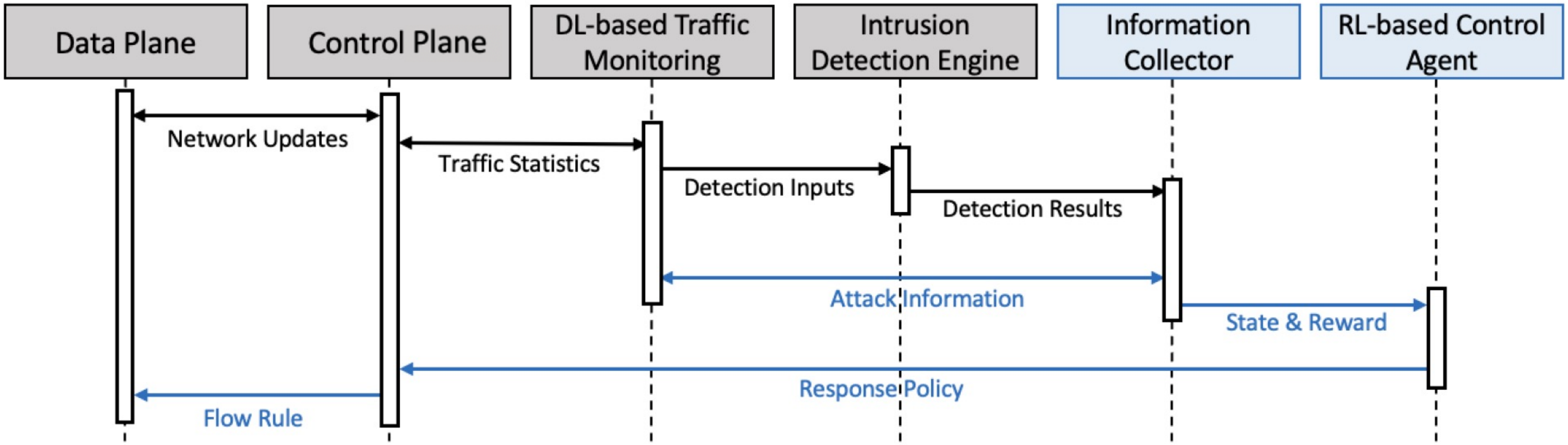


Fig. 4: Transition States.

Content

- Introduction
- Challenges and Motivation
- The proposed approach: DeepIDPS
- Experiment
- Conclusions

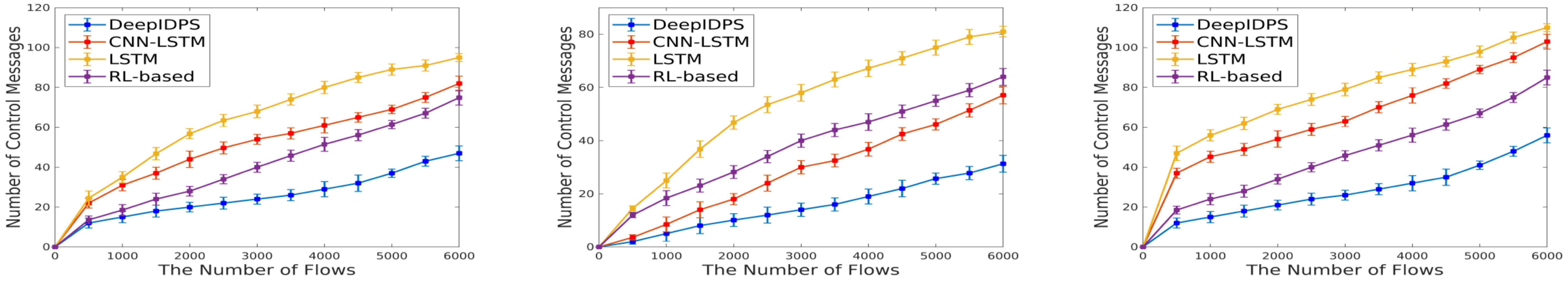
Experiment Results

Dataset:

- Traffic dataset containing a diverse range of normal and attack instances.

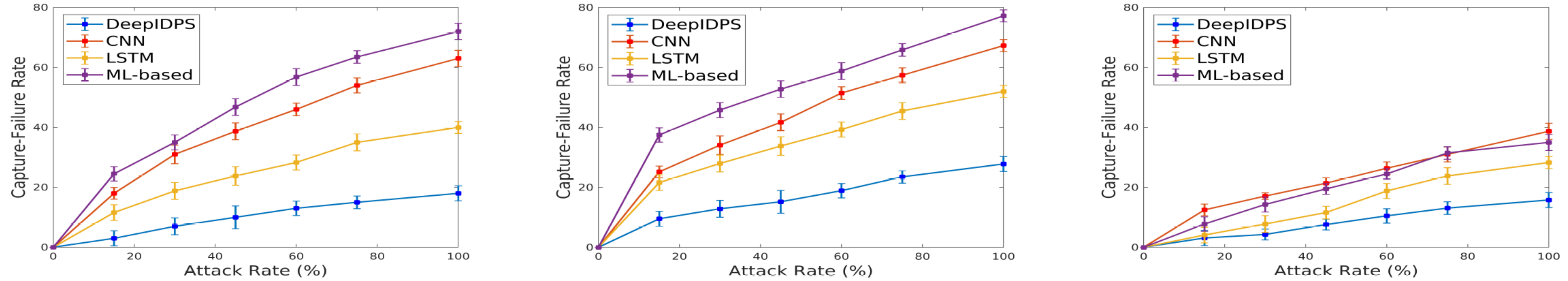
Control Messages and Overhead:

- Efficient in different traffic scales



(a) DDoS Attack (b) Port Scanning Attack (c) Zero-day Attack

Fig. 5: Total number of flow rules and control messages.



(a) DDoS Attack (b) Port Scanning Attack (c) Zero-day Attack

Fig. 6: Capture failure rate of malicious flows.

Attack Detection:

- Effective against DDoS, Port Scanning, Zero-day attacks.

Experiment Results

- **Accuracy**: Highest with 25 selected features.

TABLE I: Performance with different number of features

Performance Matrix	Number of Features				
	f=10	f=15	f=20	f=25	f=30
Accuracy (%)	97.70	98.24	98.60	98.64	97.81
Precision (%)	97.61	97.93	98.47	98.80	97.74
Recall (%)	97.56	97.86	98.41	98.77	98.71
F1-Score (%)	97.61	98.34	98.65	98.73	98.71
Loss	0.02	0.017	0.014	0.013	0.014
Time(ms)	10.3	12.3	18	23.9	26.5

- **Comparison**: CNN-LSTM outperforms CNN and LSTM.

TABLE III: Precision, Recall, and F1-score of the different methods.

Models	Precision(%)		Recall(%)		F1-score(%)	
	Normal	Attack	Normal	Attack	Normal	Attack
ML	81.19	97.86	95.17	85.21	85.74	94.23
CNN	83.43	97.55	93.62	91.85	91.17	94.56
LSTM	85.43	97.31	93.12	93.85	89.18	95.15
CNN-LSTM	94.28	98.14	93.11	96.25	94.43	97.52

Content

- Introduction
- Challenges and Motivation
- The proposed approach: DeepIDPS
- Experiment
- Conclusions

Conclusion

- ***Key Features*** *of this paper are Continuous Auto-learning and Efficient Detection and Prevention of network intrusions.*

- ***Performance:*** *DeepIDPS shows exceptional performance across multiple metrics and attack types.*

- ***Feature Selection:*** *Critical for optimizing detection accuracy and efficiency.*

- ***Model Efficiency:*** *CNN-LSTM model demonstrates superior capability in both spatial and temporal feature extraction.*

- ***Zero-day attack:*** *Deep learning models like this can adapt to new cyber threats over time.*

Conclusion

- ***Flexibility***:
  - *DeepIDPS is developed as an application for ease of deployment, configuration and interaction.*
  - *It can be implemented on top of any SDN controller.*
  - *Our DL models can be modified and optimized based on network requirements.*
  - *New threat models can also be easily added/updated.*

- ***Scalability***:
  - *DeepIDPS is designed with a goal to facilitate not only small scale networks, but also large scale networks.*
  - *The overhead of our approach does not degrade the performance of the whole network.*

Thank you!

Q & A

tun03933@temple.edu

CNN

- ***Initial Convolutional Layer:*** *Detects low-level features such as specific byte sequences in packet payloads or header fields.*

- ***Intermediate Convolutional Layers:*** *Capture more complex patterns, like sequences of packets that deviate from normal traffic.*

- ***Final Convolutional Layer:*** *Extracts high-level features representing overall traffic behavior.*

- ***Pooling Layers:*** *Reduce the feature maps' dimensions while retaining important features.*

- ***Fully Connected Layer:*** *Combines all the detected features to form a comprehensive understanding of the flow, enabling accurate classification*

Attention Mechanism

$$\mathbf{f}_{\text{CNN}} = [f_{\text{CNN1}}, f_{\text{CNN2}}, f_{\text{CNN3}}] = [0.2, 0.8, 0.5]$$

$$\mathbf{f}_{\text{LSTM}} = [f_{\text{LSTM1}}, f_{\text{LSTM2}}, f_{\text{LSTM3}}] = [0.4, 0.3, 0.9]$$

$$\mathbf{F} = [\mathbf{f}_{\text{CNN}}, \mathbf{f}_{\text{LSTM}}] = \begin{bmatrix} 0.2 & 0.8 & 0.5 \\ 0.4 & 0.3 & 0.9 \end{bmatrix}$$

$$\mathbf{W}_{\text{att}} = \begin{bmatrix} 0.1 & 0.2 \\ 0.2 & 0.1 \end{bmatrix}, \quad \mathbf{b}_{\text{att}} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$$

$$\mathbf{u}_1 = \tanh(\mathbf{W}_{\text{att}} \cdot \mathbf{f}_1 + \mathbf{b}_{\text{att}}) = \tanh(\begin{bmatrix} 0.1 & 0.2 \\ 0.2 & 0.1 \end{bmatrix} \cdot \begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}) = \tanh(\begin{bmatrix} 0.14 \\ 0.12 \end{bmatrix}) =$$

$$\alpha_i = \frac{\exp(\mathbf{u}_i^\top \mathbf{v}_{\text{att}})}{\sum_j \exp(\mathbf{u}_j^\top \mathbf{v}_{\text{att}})}$$