AI & Machine Learning in Network Security

CYB 260: Network Defenses and Countermeasures

Christina Alli, Samantha Betancourt, Jordyn Bostick, & Vanessa Reino

1865 RIDER UNIVERSITY

Why This Matters

- Cyberattacks are increasingly leveraging AI and automation
- 5G networks introduce higher speeds and larger data volumes
- Increased connectivity expands the overall attack surface
- Traditional security systems struggle with real-time analysis
- Organizations need scalable and adaptive security solutions
- AI-based intrusion detection systems are emerging as a solution for modern threats

Problem Statement

- Signature-based detection relies on known attack patterns
- New or modified attacks can bypass traditional systems
- Large-scale network traffic creates detection challenges
- High false negatives in dynamic environments
- Limited ability to detect zero-day attacks
- Traditional systems fail to detect anomaly-based attacks in real-time environments

Why AI/ML?

- Identifies patterns in large-scale network data
- Detects anomalies using learned patterns in network behavior
- Adapts to new and evolving threats over time
- Enables faster and more automated threat detection
- Scales more effectively than manual or rule-based systems

Project Objectives

1. Developed an AI-based intrusion detection system (IDS)
2. Simulated network traffic in a controlled environment
3. Analyzed differences between normal and malicious behavior
4. Applied machine learning to classify network activity
5. Evaluated performance using detection metrics

RIDER UNIVERSITY

Research Questions

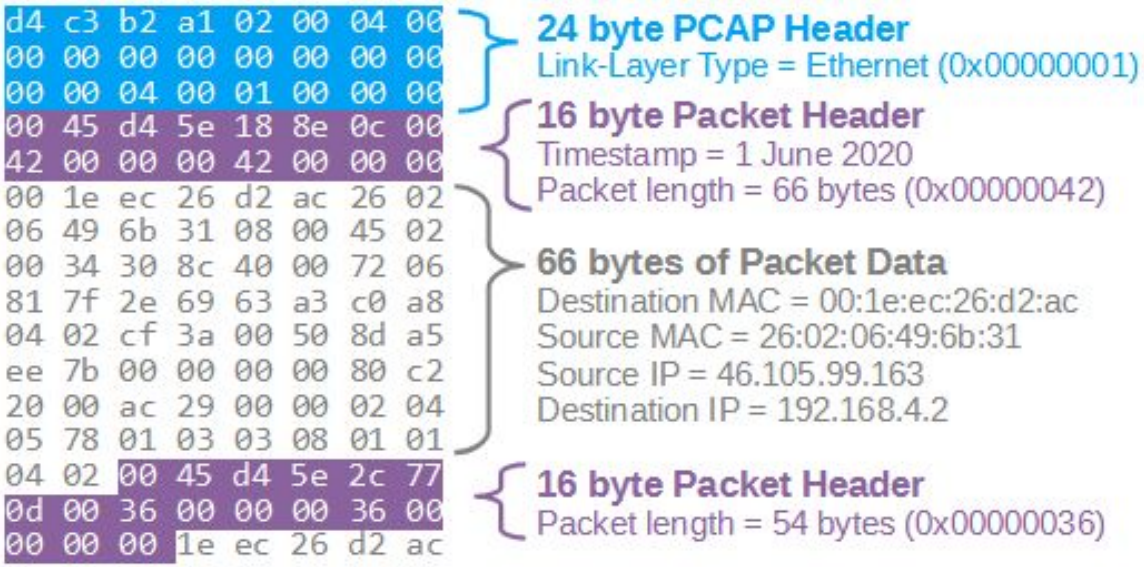
1. How effectively can an AI-based IDS detect anomalies in network traffic?
2. Can machine learning maintain accuracy when analyzing high-volume data?
3. How does anomaly detection improve network security (CIA triad)?

Lab Environment Architecture

- Isolated virtual environment
- Scapy (custom packet generation)
- Wireshark (packet capture)
- Python (model + preprocessing)

Data Collection Process

- Network traffic captured using packet analysis tools (Wireshark)
- ~1100 packets generated using Scapy (benign + malicious traffic)
- Includes simulated attacks (SYN flood, UDP flood, scans)
- Data stored in PCAP format for analysis
- Data prepared and converted for machine learning model input



Feature Extraction

- Source and destination IP addresses
- Network protocol types (TCP, UDP, etc.)
- Packet size and transmission patterns
- TCP flags and session behavior
- Port numbers and timestamps
- Data normalized and converted to CSV
- Extracted features used to train detection models

Real-Life Example

CrowdStrike Falcon — AI/ML in Modern Network Security

- Uses cloud-native AI/ML for real-time threat detection
- Analyzes trillions of events per day across global endpoints
- Detects behavioral anomalies, lateral movement, and zero-day attacks

Machine Learning Approach

- Autoencoder neural network used for anomaly detection
- Model trained on normal network traffic only
- Detects attacks using reconstruction error
- Learns patterns of normal vs abnormal behavior

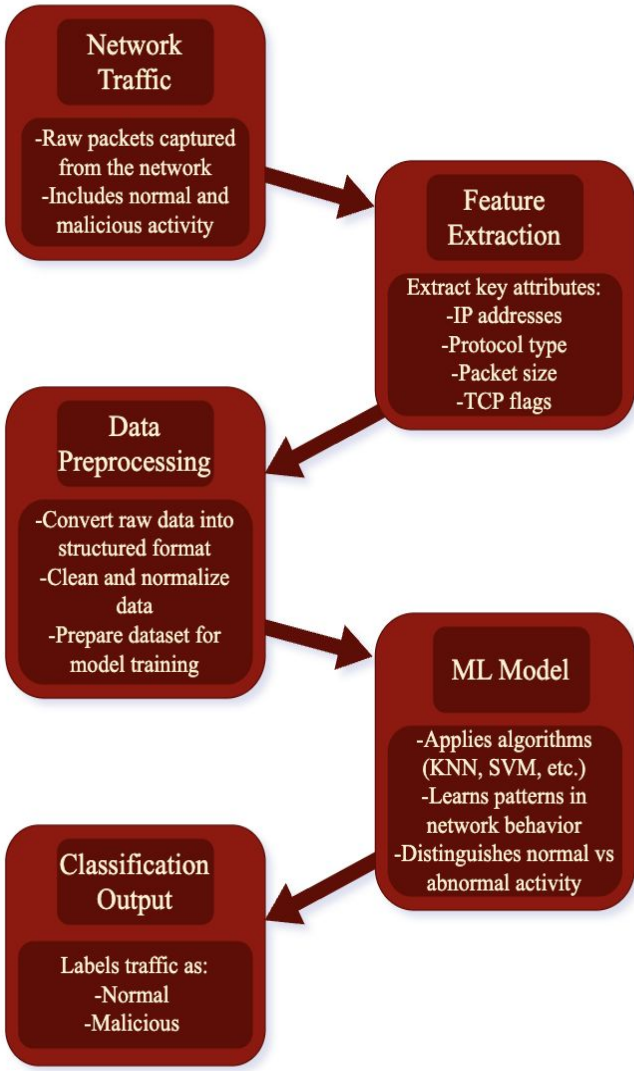
```python
def create_model(input_size, hyper_param):

    act_func = 'elu'
    model = Sequential()

    # ENCODER
    model.add(Dropout(hyper_param.drop_out[0], input_shape=(input_size,)))

    model.add(Dense(
        hyper_param.layers[0],
        activation=act_func,
        kernel_initializer=initializers.glorot_uniform(seed=hyper_param.random_seed),
        activity_regularizer=regularizers.l1(hyper_param.kernel_reg[0])
    ))

    model.add(Dropout(hyper_param.drop_out[1]))

    for i in range(1, len(hyper_param.layers) - 1):
        model.add(Dense(
            hyper_param.layers[i],
            activation=act_func,
            kernel_initializer=initializers.glorot_uniform(seed=hyper_param.random_seed),
            activity_regularizer=regularizers.l1(hyper_param.kernel_reg[i])
        ))
        model.add(Dropout(hyper_param.drop_out[i+1]))

    # BOTTLENECK
    model.add(Dense(
        hyper_param.layers[-1],
        activation=act_func,
        kernel_initializer=initializers.glorot_uniform(seed=hyper_param.random_seed)
    ))

    # DECODER
    for i in range(len(hyper_param.layers) - 2, -1, -1):
        model.add(Dense(
            hyper_param.layers[i],
            activation=act_func,
            kernel_initializer=initializers.glorot_uniform(seed=hyper_param.random_seed),
            activity_regularizer=regularizers.l1(hyper_param.kernel_reg[i])
        ))

    model.add(Dense(
        input_size,
        kernel_initializer=initializers.glorot_uniform(seed=hyper_param.random_seed)
    ))

    model.compile(loss='mse', optimizer='adam')

    print("\nModel Summary...")
    print(model.summary())

    return model
```

Learning Approaches

- Unsupervised learning used for anomaly detection (autoencoder)
- Reinforcement learning adapts based on system feedback
- Combination of approaches improves detection flexibility
- Allows identification of both known and unknown threats

Model Workflow

- Network traffic is collected and preprocessed
- Data is converted into structured feature sets
- Machine learning model is trained on labeled data
- Model analyzes incoming traffic patterns
- Output classifies activity as normal or malicious
  - Classification based on reconstruction error



Training & Validation

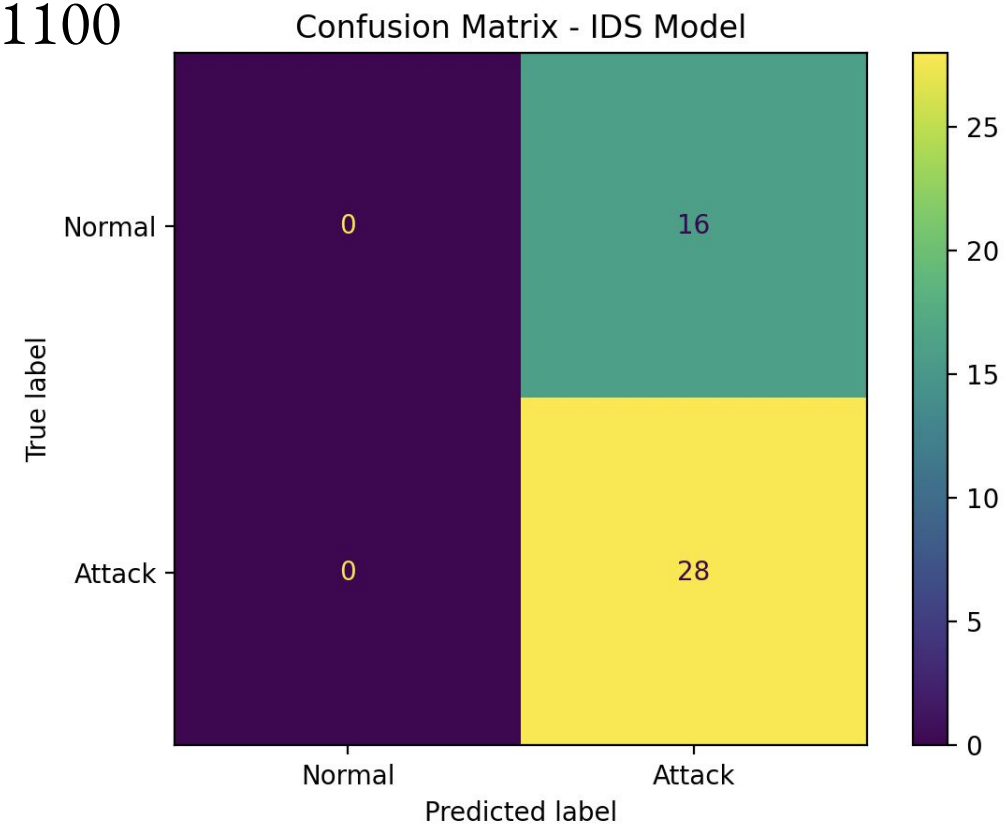
- Dataset split into 95% training and 5% validation
- Model trained only on normal traffic
- Early stopping used to prevent overfitting
- Performance monitored using reconstruction error

Example Detection

- Normal traffic:
  - Stable packet patterns
  - Consistent protocol behavior
  - Successful data reconstruction
- Malicious traffic:
  - SYN flood / abnormal packet spikes
  - Unusual flag patterns
  - Errors in data reconstruction

Results and Findings

- Total packets analyzed: ~1100
- Anomalies detected: 800
- Benign packets: 279
- Detected attacks: 44
- True Positives: 28
- False Positives: 16
- False Negatives: 0
- True Negatives: 0
- High detection rate with increased false positives



AI vs Traditional IDS Comparison

Feature	Traditional IDS	AI-Based IDS
Detection	Signature-based	Behavior-based
Unknown Threats	Incapable	Capable
Scalability	Limited	High

AI-based IDS uses anomaly detection via autoencoder.

Strengths of AI Detection

- Successfully detected all attack samples (FN = 0)
  - Very sensitive to threats
- Processes large volumes of data efficiently
- Reduces reliance on manual analysis
- Detects previously unknown attack patterns
- Enables real-time monitoring and response
- Improves scalability in complex network environments

Limitations and Risks

- High false positive rate observed in results
- Limited dataset size (~1100 packets)
  - Requires large, high-quality datasets for training
- High computational and resource requirements
- Potential for false positives and misclassification
- Model performance depends on training data quality

Real-World Implications

- Enhances enterprise network security systems
- Supports protection of critical infrastructure
- Applicable to cloud and large-scale environments
- Improves defense against modern cyber threats
- Plays a role in national cybersecurity strategies

Future Improvements

- Incorporate more diverse and complex datasets
- Combine AI with traditional detection systems
- Integrate with SIEM and monitoring platforms
- Improve real-time deployment capabilities
- Reduce false positives through model tuning

Conclusion

- Model achieved high detection but with false positives
- Demonstrated effectiveness of anomaly-based detection
- AI enhances intrusion detection capabilities
- Provides scalable and adaptive security solutions
- Effectiveness depends on data quality and implementation
- Represents a key direction for future cybersecurity defense
- Not a replacement, but a complement to traditional systems

References

A. S. Jacobs, R. Beltiukov, W. Willinger, R. A. Ferreira, A. Gupta, and L. Z. Granville, "AI/ML for Network Security," Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, Nov. 2022, doi: https://doi.org/10.1145/3548606.3560609.

M. A. Goffer et al., "AI-Enhanced Cyber Threat Detection and Response Advancing National Security in Critical Infrastructure," Journal of Posthumanism, vol. 5, no. 3, Apr. 2025, doi: https://doi.org/10.63332/joph.v5i3.965.

A. Perez Veiga, "Applications of Artificial Intelligence (AI) to Network Security," 2018. Available: https://arxiv.org/pdf/1803.09992

M. Sumathi, J Shruthi, V. Jain, G Kalyan Kumar, and Z. Z. Khan, "Using Artificial Intelligence (AI) and Internet of Things (IoT) for Improving Network Security by Hybrid Cryptography Approach," Evergreen, vol. 10, no. 2, pp. 1133–1139, Jun. 2023, doi: https://doi.org/10.5109/6793674.

Y. Tan, Q. Zhang, Y. Li, and X. Yu, "AI-Driven Network Security and Privacy," Electronics, vol. 13, no. 12, pp. 2311–2311, Jun. 2024, doi: https://doi.org/10.3390/electronics13122311.

N. Haider, M. Z. Baig, and M. Imran, "Artificial Intelligence and Machine Learning in 5G Network Security: Opportunities, advantages, and future research trends," arXiv.org, Jul. 08, 2020. https://arxiv.org/abs/2007.04490