ZERO-DAY ATTACK DETECTION USING DEEP LEARNING

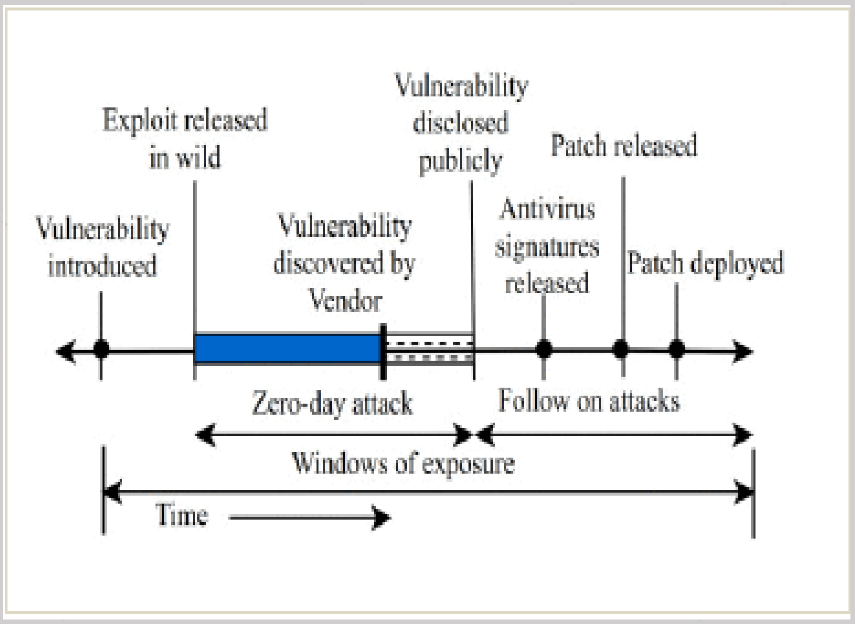
Presenter: Manish Acharya
Department: Computer Science (Cyber Security Analytics)

INTRODUCTION

- Cybercrime and attack methods have been steadily increasing since 2019 (pandemic).
- In the years following 2019, the number of victims and attacks per hour has rapidly increased
- Threat landscape has grown rapidly as more technologies are introduced and businesses continue to go digital
- Zero-day exploits have skyrocketed across all industries with increasing internet of things (IoT), cloud hosting, and more advanced mobile technologies
- State-sponsored actors, led by Chinese groups, are the primary attackers of zero-days.
- Zero-days bypass the traditional signature and anomaly-based detections and antivirus software
- Frameworks incorporating AI, such as machine learning and deep learning along with traditional techniques are more effective at detecting zero-days

WHAT IS ZERO DAY ATTACK AND WHY IT IS DANGEROUS.

- Unpredictable and Severe Impact : Zero-day attacks exploit unknown vulnerabilities, leading to potentially catastrophic consequences such as data breaches, system disruptions, and financial losses.
- No Immediate Fixes Available: Since the vulnerabilities are not yet known, there are no patches or updates available at the time of the attack, leaving systems exposed and vulnerable.
- High Risk of Exploitation: Attackers can leverage zero-day vulnerabilities to gain unauthorized access or control over systems, making it easier to execute malicious activities before defenses can be updated.
- Extended Exposure Time: The period between the discovery of a zero-day vulnerability and the development of a patch can be long, during which systems remain at high risk of exploitation.
- Proactive Defense Required: Organizations must implement advanced threat detection and response strategies, as well as maintain up-to-date security practices, to mitigate the risk and impact of zero-day attacks.



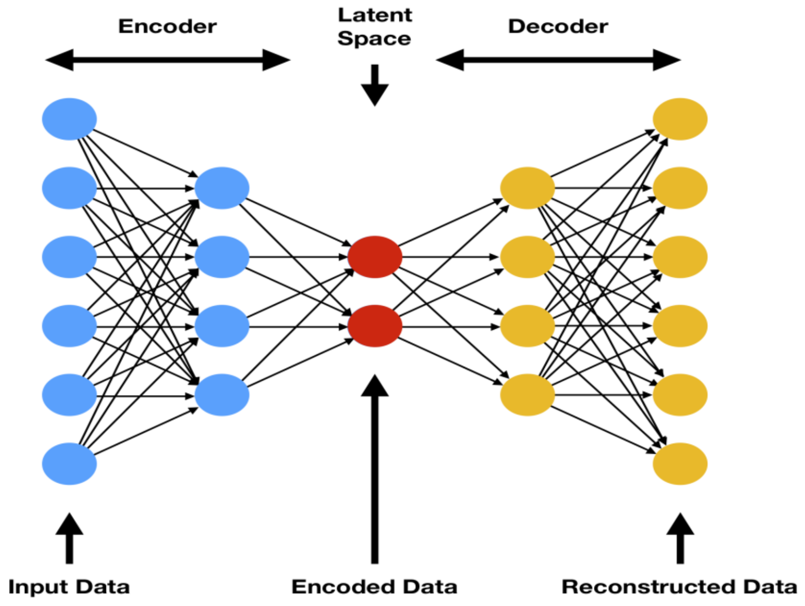
TECHNICAL BACKGROUND

•An **autoencoder** is a type of **neural network** that is trained to learn a compressed representation of the input data.

•The autoencoder consists of an **encoder** network that compresses the input data into a lower-dimensional latent space, and a **decoder** network that **reconstructs** the original input from the latent representation.

•During training, the autoencoder learns to minimize the **reconstruction error**, which helps it capture the essential features and patterns in the data.

•In the context of cyber security, the autoencoder is trained on normal network traffic data, allowing it to learn a model of "normal" behavior.



AIMS & OBJECTIVES

- **Aim:** To develop and optimize an autoencoder-based model for accurate and reliable detection of cyber attacks.

- **Objectives:**
  - **Optimize Model Architecture:** Refine the autoencoder design to improve detection capabilities.
  - **Reduce False Positives:** Implement strategies to lower false positive rates in anomaly detection.
  - **Evaluate on Real-World Data:** Test the model using the CICIDS2017 dataset to ensure robustness and effectiveness in practical scenarios.

- **Challenges:**
  - High variability in attack patterns.
  - Imbalanced datasets (normal vs. attack traffic).

DATASET

- **The Data**: CICIDS2017 Dataset
- **Dataset Description:**
  - Created by capturing network traffic data from July 3-7, 2017.
  - Emulated environment with both packet-based and flow-based formats.
- **Attack Types:**
  - Contains a wide range of inside and outside attacks: DDoS, Infiltration, Brute Force SSH, Heartbleed, etc.

Category	Class Labels	Number of instances
Attack/Test Data	Web Brute Force	1507
	Botnet	1966
	DoS-Slowhttptest	5499
	DoS-slowloris	5796
	SSH-Patator	5897
	FTP-Patator	7938
	DoS-GoldenEye	10293
	DDoS	41835
	PortScan	158930
	DoS Hulk	231072
Normal/Training Data	BENIGN	146170
Total		616903

EXPERIMENT DESIGN & METHODS

•**Model Design:** A deep autoencoder with multiple hidden layers, optimized using random search to select the best architecture, learning rate, and activation functions.

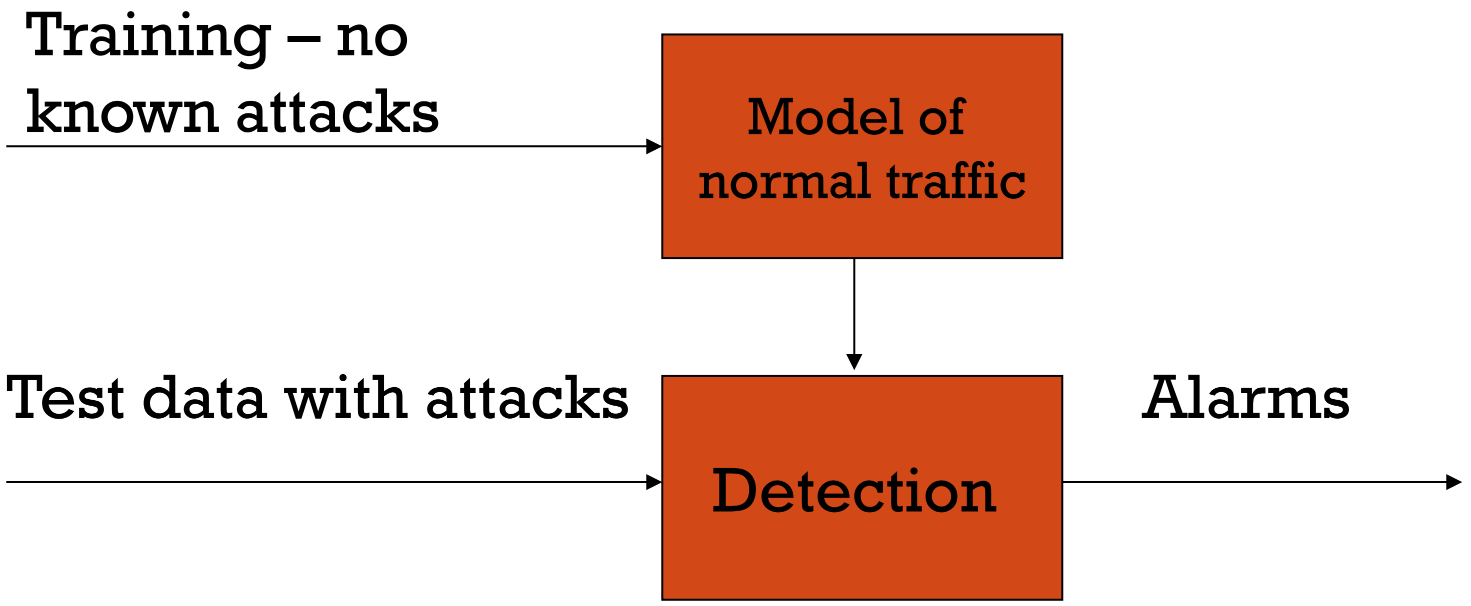
•**Training Strategy:**

- •**Data Split:** 80% of the data for training and 20% for validation.
- •**Regularization:**L2 regularization to prevent overfitting.
- •**Optimizer:** Adam optimizer with a learning rate of 1.00E-05.

•**Evaluation Metrics:** Mean Squared Error (MSE) for anomaly detection, accuracy for evaluating detection performance.

EXPERIMENT DESIGN & METHODS (CONTINUED…)

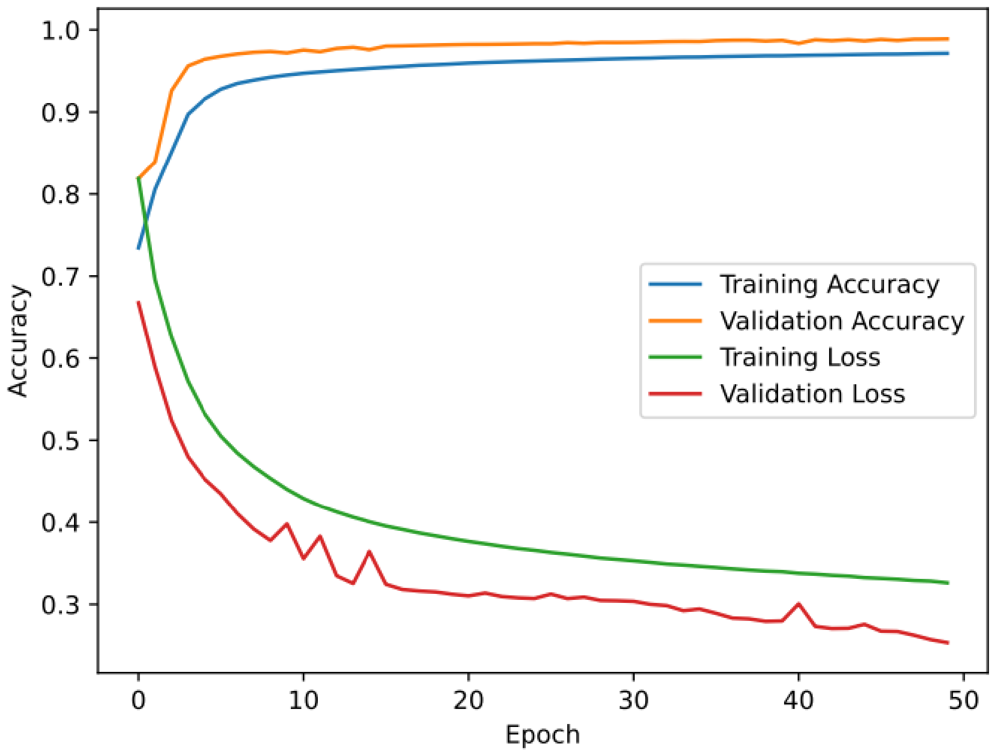
- Detect (not prevent) attacks in network traffic
- No prior knowledge of attack characteristics



EXPERIMENTS

- **Tests Conducted:**
- Various architectures and hyperparameters.
- Different MSE thresholds for anomaly detection.

- **Success Measure:**
- Accuracy in detecting attacks.
- Mean Squared Error (MSE) as the primary metric.



RESULTS

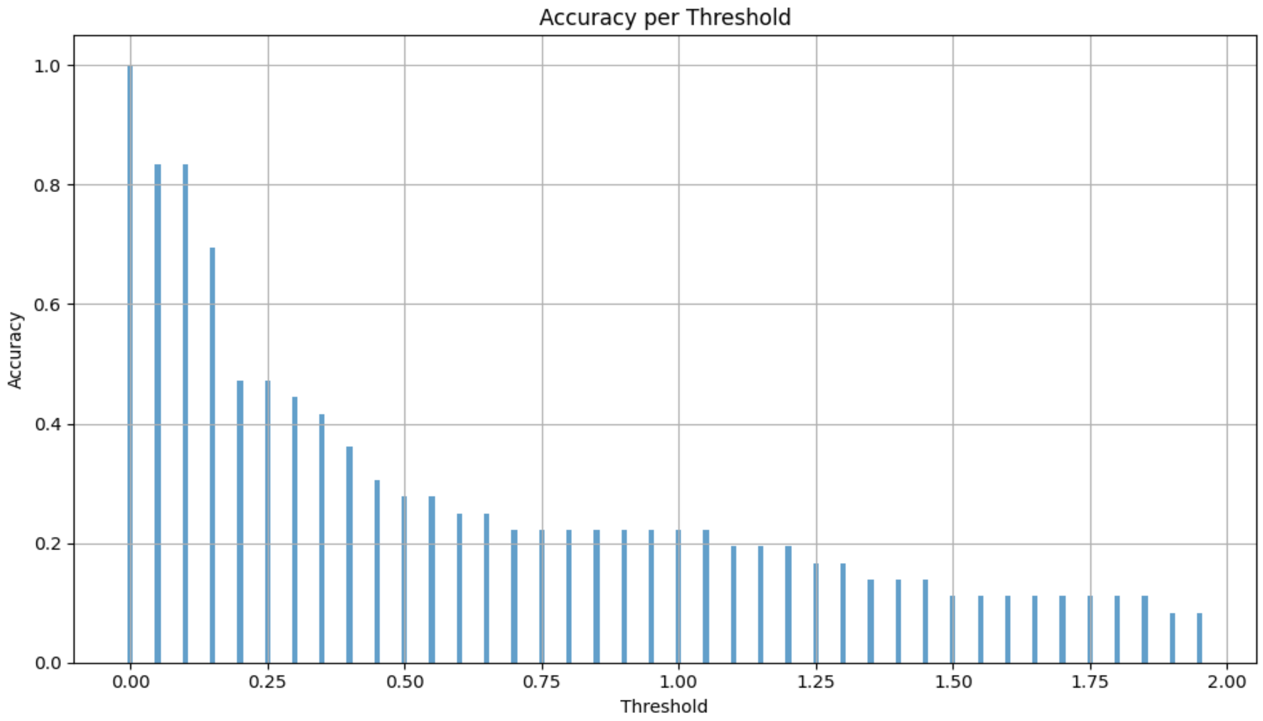
•Graph shows accuracy vs. detection threshold

•Lower thresholds: Higher accuracy, more false positives

•Higher thresholds: Lower accuracy, fewer false positives

•However, the model exhibited higher false positives, particularly for certain attack types, indicating the need for further refinement.

•**Performance Metrics:** Accuracy scores for different thresholds, with a focus on reducing false positives without compromising



CONCLUSIONS

- **Key Takeaways:**
  - The optimized autoencoder shows promise in detecting cyber attacks, but further work is needed to reduce false positives.
  - Challenges include handling the variability in normal traffic and improving the model's ability to generalize.

- **Future Directions:**
  - Focus on reducing false positives through better model calibration and feature engineering.
  - Explore the integration of ensemble methods for more robust anomaly detection.

Thank you! Below is recording link

Recording-20240809_095654.webm