Transformer-based Network Anomaly Detection

Afia Darko Asante
Department of Computer Science
University of Wisconsin-Milwaukee
Wisconsin, USA
adasante@uwm.edu

Amogha Manjunatha Kuruvadi
Department of Computer Science
University of Wisconsin-Milwaukee
Wisconsin, USA
kuruvadi@uwm.edu

Indrasena Kalyanam
Department of Computer Science
University of Wisconsin-Milwaukee
Wisconsin, USA
kalyana3@uwm.edu

Introduction

In the constantly evolving world of cybersecurity, anomaly detection has become increasingly critical in safeguarding networks from many threats. Traditional methods often struggle with the complex and nuanced nature of modern network traffic, necessitating advanced techniques. Transformers, a groundbreaking neural network architecture initially popularized in natural language processing (NLP), have emerged as a powerful tool for addressing these challenges. This report explores the application of transformers in anomaly detection within network environments, offering insights into the advantages and technical underpinnings of this approach.

Problem Definition

The problem addressed by this project is the detection of anomalies in network traffic, a crucial aspect of cybersecurity. Anomalies represent deviations from normal behavior and can indicate security threats, such as unauthorized access or system faults. As networks grow and complex, traditional anomaly detection methods struggle to keep pace with the evolving threats and increasing traffic volumes.

Anomaly Types: Understanding Deviations from Normality

Anomalies in networks typically fall into two broad categories: intrusions and faults. Intrusions involve unauthorized access to networks or their resources, usually with malicious intent, encompassing threats like denial-of-service attacks, data theft, and malware propagation. Faults, on the other hand, result from internal issues such as equipment malfunctions, configuration errors, and software bugs, leading to disruptions and data loss without malicious motives.

Challenges

Several challenges are inherent in network anomaly detection:

High Complexity: Modern networks generate enormous amounts of data with complex patterns, making it difficult to distinguish between normal and anomalous behavior.

Dynamic Nature: Network traffic patterns change frequently due to varying user activities, making it hard to maintain a consistent baseline for anomaly detection.

Noise and Imbalance: Network data often contains noise and imbalanced distribution of normal and anomalous instances, complicating the detection process.

Detection Accuracy: False positives and negatives can lead to security risks and wasted resources. Anomaly detection methods must balance sensitivity with accuracy.

Importance of the Problem

The importance of effective anomaly detection in networks cannot be overstated:

Cybersecurity: Anomalies often indicate potential security breaches, such as malware infections, denial-of-service attacks, or data exfiltration. Early detection is crucial to prevent damage.

System Reliability: Anomalies related to system faults, like equipment malfunctions or software bugs, can cause downtime and disrupt business operations.

Compliance and Regulations: Many industries have strict compliance requirements for data security and network integrity. Effective anomaly detection ensures compliance with these regulations.

Existing Solutions

Traditional solutions for network anomaly detection include:

Statistical Methods: Techniques like standard deviation and mean-based thresholds are used to identify deviations from expected patterns.

Machine Learning Models: Supervised and unsupervised learning algorithms are applied to detect anomalies based on labeled or unlabeled data.

Signature-Based Detection: This approach uses predefined patterns or signatures to identify known threats.

Limitations of Existing Solutions

Despite their widespread use, existing solutions have notable limitations:

Inability to Handle Complexity: Statistical methods struggle with high-dimensional data and complex patterns in modern network traffic.

Static Nature: Signature-based detection relies on predefined patterns, making it ineffective against novel threats.

High False Positives/Negatives: Machine learning models can produce inaccurate results, especially with noisy or imbalanced data.

Lack of Contextual Understanding: Many traditional methods fail to capture contextual information, reducing their ability to detect subtle anomalies.

System Design

Transformer-based Anomaly Detection:

The anomaly detection process using transformers involves two primary stages:

Feature Extraction: Network traffic data, in the form of packets, undergoes preprocessing to extract key features from headers, payloads, and timestamps. These features are transformed into a format suitable for analysis by transformer-based models.

Anomaly Detection: The transformer model uses these extracted features to identify patterns that deviate from normal network behavior. It assigns a score to each packet, indicating its likelihood of being an anomaly. Higher scores signal potential anomalies, prompting further investigation.

Technical Nuances: Understanding the Transformer Architecture

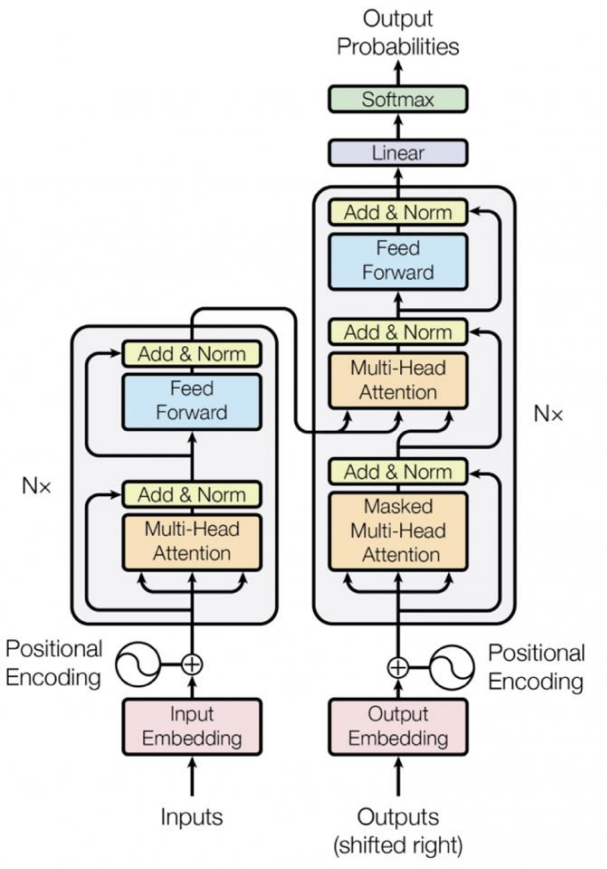


Figure 1: Transformer Architecture Source: *From a slide of Prof. Rohit Kate (NLP)*

The transformer architecture has several key components that contribute to its efficacy in anomaly detection:

Encoder-Decoder Structure: This structure allows the transformer to capture long-range dependencies and extract contextual information from sequences of features.

Self-Attention Mechanism: The self-attention mechanism focuses on the most relevant parts of the input sequence, identifying relationships between features and highlighting patterns that suggest anomalies.

Positional Encoding: Since network traffic data is sequential, positional encoding provides information about feature order, enabling the model to detect temporal dependencies and anomalies evolving over time.

Advantages of Transformer-based Anomaly Detection

Transformers offer several distinct advantages for anomaly detection in networks:

Robustness: Transformers are adept at handling noisy and imbalanced data, allowing them to discern anomalies even in complex network traffic.

Adaptability: These models can adapt to changing network patterns and evolving threats, reducing the need for constant retraining.

Explainability: Unlike many traditional black-box models, transformers can provide insights into the factors contributing to anomaly detection, aiding in root cause analysis and incident response.

Implementation

Dataset 1: NSL-KDD

Data Preprocessing

Loading Data: The dataset is loaded into a Pandas Data Frame. The features are extracted from all but the last column, which is used as the label.

Handling Non-numeric Data: The code checks for non-numeric values in the feature columns, labeling any problematic columns. If any are found, they are converted to numeric using Label Encoder.

Normalizing Numerical Features: The StandardScaler is used to normalize the numerical features to ensure they're on a similar scale.

Label Encoding: Labels are encoded to convert them into a format suitable for training a neural network.

Model Definition

A simple transformer-based model is defined with the following components:

Linear Embedding Layer: Converts input features into a 256-dimensional space suitable for processing by the transformer.

Transformer Encoder: Consisting of three layers with an attention mechanism, it analyzes the transformed input to identify patterns.

Classifier: A linear layer that classifies the transformed data into the expected output categories.

Training the Model

The model is trained using a standard Py Torch Data Loader for handling batches of data during training. The following aspects were critical during training:

Optimizer: AdamW was used as the optimizer with a learning rate of 1e-4.

Criterion: Cross-Entropy Loss was used to calculate the loss during training.

Training Loop: The model was trained for four epochs, with each epoch consisting of multiple iterations through the training data. At each iteration, the loss was calculated, backpropagation was performed, and model parameters were updated.

***Description**:* The dataset includes several attributes representing various aspects of network traffic:

Protocol Type: Refers to the communication protocol, including 'tcp', 'udp', and 'icmp'.

Service: Denotes the service involved in the traffic, such as 'http', 'ftp', or 'smtp'.

Flag: Represents the status of the network connection, with values like 'SF', 'REJ', or 'S0'.

Source and Destination Bytes: Indicates the number of bytes sent from the source to the destination and vice versa.

Numerical Metrics: Includes various metrics, such as 'duration', 'land', 'wrong_fragment', and 'urgent', reflecting aspects of network behavior.

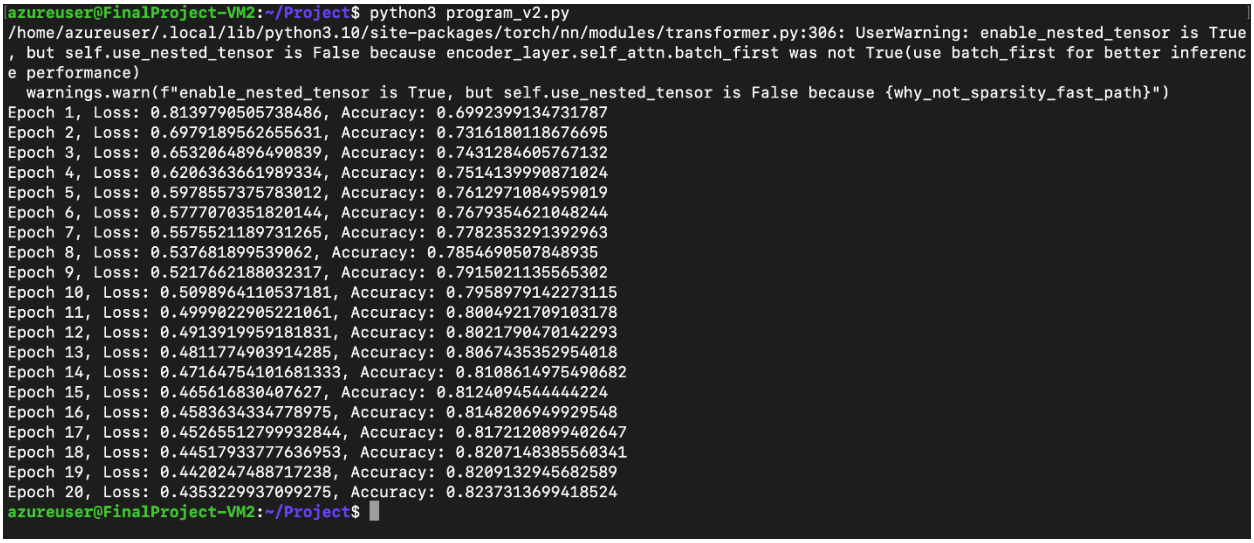
Boolean Flags: Contains binary indicators for conditions like 'logged_in', 'is_host_login', and 'is_guest_login'.

Derived Metrics: Consists of calculated values derived from other attributes, such as 'serror_rate', 'srv_serror_rate', and 'rerror_rate'.

Label: Defines the classification of the traffic as either 'normal' or 'anomaly'.

Results

The following loss values were recorded during training:



Screenshot 1*: NSL-KDD Results*

Consistent Reduction in Loss: Over the four epochs, the loss steadily decreased from 0.8139 to 0.4353. This indicates that the model is effectively learning to classify the data and is minimizing the error with each epoch.

Increase in Accuracy: Accuracy also improved across the epochs, starting at 69% and rising to 82%. This consistent increase suggests that the model is becoming better at correctly predicting whether a given network traffic pattern is normal or an anomaly.

Model Learning and Adaptation: The trend of decreasing loss and increasing accuracy reflects successful learning. As the model is exposed to more training data, it refines its internal parameters to better classify the network traffic. This indicates that the transformer-based approach is suitable for this type of anomaly detection task.

Dataset 2: KDDCUP99

Description:

KDDCUP99 is a classic dataset used for benchmarking anomaly detection systems, specifically network intrusion detection systems (NIDS). It was derived from a dataset built for the Third International Knowledge Discovery and Data Mining Tools Competition, held with KDD-99, The Fifth International Conference on Knowledge Discovery, and Data Mining. Here is a more detailed description of the dataset:

Origin: Created by processing the tcpdump portions of the 1998 DARPA Intrusion Detection System (IDS) Evaluation dataset, created by MIT Lincoln Labs.

Content: The dataset contains a wide variety of intrusions simulated in a military network environment.

Size: Contains about 4.9 million instances, each described by 41 features and labeled as either normal or an attack, with one specific attack type.

Features: These include basic features of individual TCP connections (like duration and protocol type), content features within a connection (like number of failed login attempts), and traffic features computed over a window interval (like the number of connections to the same host).

Attack Types: Includes a variety of attacks categorized into four main groups—Denial of Service (DoS), Unauthorized Access from a Remote Machine (R2L), Unauthorized Access to Local Superuser (Root) Privileges (U2R), and Probing.

How the Model Works

The implemented model uses a **Transformer architecture** tailored for tabular data, particularly effective for tasks where relationships within the data at different time steps (or sequence of events) are crucial for prediction. Here is a breakdown of its operation:

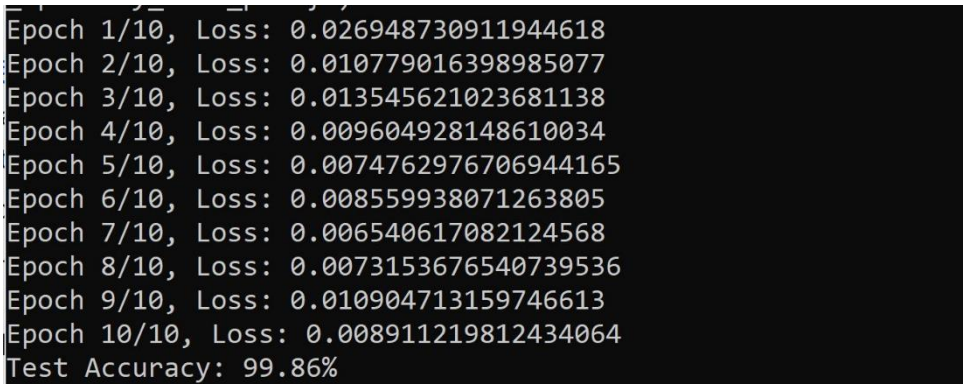
Embedding Layer: Transforms each feature into a higher-dimensional space, enabling the model to better capture interactions between distinctive features.

Transformer Encoder: Applies self-attention mechanisms to weigh the importance of unique features dynamically. It allows the model to focus on more relevant features for making predictions.

Classifier: A final dense layer that uses the output of the transformer to predict whether a network connection is normal or an attack, identifying the specific type of attack.

Training and Evaluation Process

The training process involves several epochs where the model learns to minimize the difference between its predictions and the actual classifications of the data (loss). Optimizations are made using the Adam optimizer, a popular choice for deep learning tasks due to its efficient handling of sparse gradients and adaptive learning rate capabilities.



Screenshot 2*: KDDCUP99 Results*

Loss Reduction: The above screenshot shows the model's loss decreasing significantly across epochs, indicating effective learning and adaptation to the training data.

Test Accuracy: Achieving a 99.86% accuracy on the test set, the model demonstrates high efficacy in classifying network connections correctly, discerning between normal operations and several types of attacks.

Dataset 3: UNSW-NB15

Description:

The UNSW-NB15 dataset is a comprehensive dataset designed for training and testing network intrusion detection systems (NIDS). It was developed by the Australian Centre for Cyber Security (ACCS) using the IXIA PerfectStorm tool to generate a mixture of modern malicious and benign network traffic.

Creation: The dataset was created using commercial penetration testing software in a controlled environment to simulate real-world traffic scenarios.

Composition: It consists of several types of attacks and normal traffic flows, providing a diverse range of challenges for intrusion detection models.

Features: The dataset includes a mix of flow features (like source and destination IPs, ports, and protocol type), basic features (such as duration and byte counts), content features (including HTTP methods and status codes), time features (like inter-arrival times and jitter), and additional generated features (such as Boolean flags to indicate the presence of FTP login attempts and successful outcomes).

How the Model Works

The model is built using a Transformer architecture, specifically adapted for handling tabular data, which is common in network traffic analysis:

Data Preprocessing:

***Categorical and Numerical Data*:** Initial data preprocessing involves encoding categorical features and scaling numerical features to ensure the model receives well-formatted input.

***Imputation*:** Missing values are handled through imputation, ensuring the model does not encounter gaps in the data.

Model Architecture:

***Embedding Layer*:** Maps numerical input into a higher-dimensional space to better capture the relationships between different inputs.

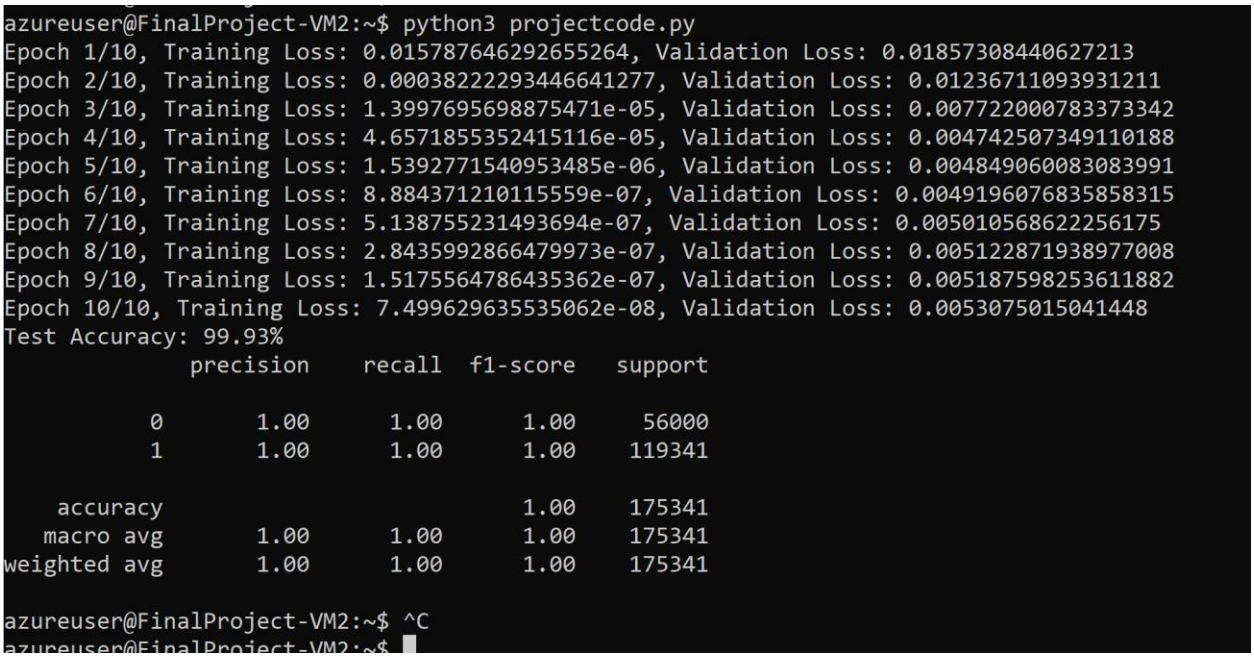
***Transformer Encoder*:** Utilizes self-attention mechanisms to dynamically weigh the relevance of distinctive features from the input data.

***Output Layer*:** A linear layer that outputs the probabilities of different classes, which in this case are types of network traffic including various attack types.

Training and Validation:

***Epoch-based Training*:** The model is trained over multiple epochs, improving its predictions by minimizing a loss function that measures the difference between predicted and actual values.

***Validation*:** At each epoch, the model's performance is also tested on a separate validation set to monitor its ability to generalize to unseen data.



Screenshot 3*: UNSW-NB15 Results*

***Loss and Accuracy*:** Throughout the training process, the model shows a decreasing trend in loss, indicating effective learning. The final recorded accuracy test is exceptionally high, suggesting that the model is highly effective at classifying network traffic correctly.

***Precision, Recall, and F1-Score*:** The model achieves perfect scores across these metrics for both classes (normal and attack), which implies excellent performance in not only identifying attack instances but also in minimizing false positives and false negatives.

Dataset 4: CIC (Canadian Institute for Cybersecurity)

Dataset Description: The CICIDS2017 dataset was created by the Canadian Institute for Cybersecurity at the University of New Brunswick in 2017 specifically for intrusion detection research and development purposes. This dataset is notable for its comprehensive representation of a typical modern network environment, which was designed to provide a balanced and realistic platform for evaluating intrusion detection systems.

Key Characteristics:

Realistic Traffic: The dataset consists of benign and the most up-to-date common attack scenarios, which were generated in a controlled environment to simulate authentic interactions in a typical enterprise network. This realism aids in training more effective intrusion detection systems that can perform well in real-world settings.

Comprehensive Features: A total of 80 features were extracted from the network traffic captured in pcap files. These features include but are not limited to SourceIP, SourcePort, DestinationIP, DestinationPort, and Protocol. Such detailed feature extraction helps in creating a detailed context around each network event, facilitating more nuanced analysis and detection capabilities.

Attack Scenarios: The dataset includes a variety of attack types such as DDoS, PortScan, Botnet, Web attacks, and more, providing a broad spectrum of test cases to evaluate the resilience and accuracy of intrusion detection models.

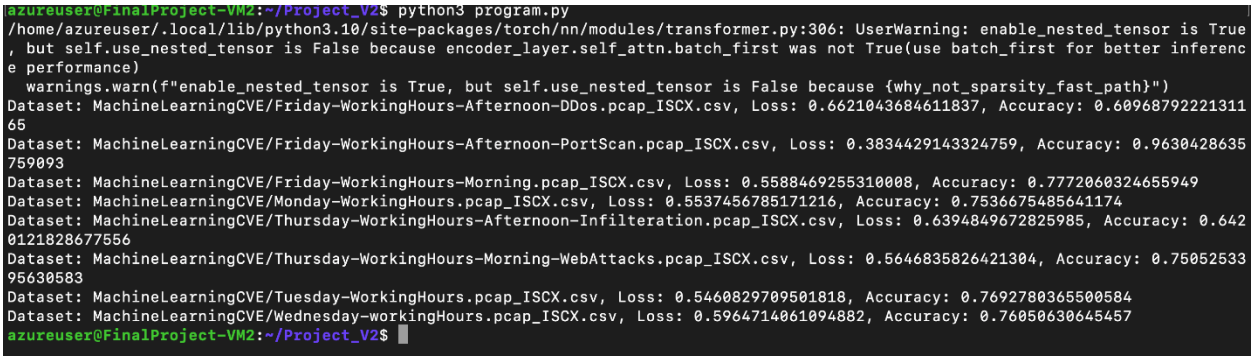
Generated using B-Profile System: The background traffic and the attack scenarios were generated using the B-Profile system, which is designed to mimic user behaviors and traffic patterns in a way that closely resembles a real-world network. This system ensures that the dataset is not only diverse but also includes minor subtleties that typically occur in actual network environments.

Model Overview and Performance:

Model Architecture: The model utilized is a Transformer-based architecture adapted for handling tabular data, which includes an embedding layer, a transformer encoder with self-attention

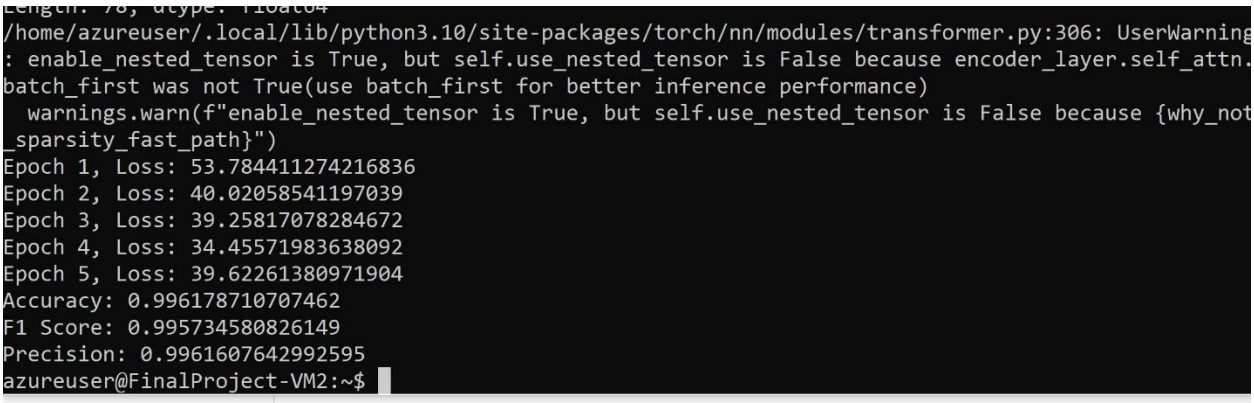
mechanisms, and a linear classifier. This setup is particularly effective in discerning patterns and anomalies in network traffic data.

Preprocessing and Training Approach:



Screenshot 4*: CIC Dataset initial accuracy results*

Initial testing on multiple datasets revealed variability in performance as you can see in above screenshot, leading to a strategic shift to focus solely on the "**Friday-WorkingHours.pcap_ISCX.csv**" for detailed optimization. Preprocessing steps were introduced to manage missing values, infinite values, and normalize data to ensure quality input into the model.



Screenshot 5: *CIC Results*

Focusing on a single, well-preprocessed dataset and leveraging the Transformer architecture yielded a significant improvement in model performance, achieving an accuracy of approximately 99%. Precision and F1 scores were also exceptionally high, indicating excellent model reliability and generalization capability across different types of network traffic.

Conclusion

This project explored the application of transformer-based models for anomaly detection within network environments, addressing significant challenges in cybersecurity. Our findings demonstrate the transformer's ability to handle the complex and dynamic nature of network traffic, making it an excellent candidate for detecting network anomalies effectively.

High Performance Across Datasets: The transformer model achieved high accuracy levels on multiple benchmark datasets, including NSL-KDD, KDDCUP99, UNSW-NB15, and CICIDS2017, with accuracy figures often exceeding 99%. This indicates a strong generalization capability across different types of network environments.

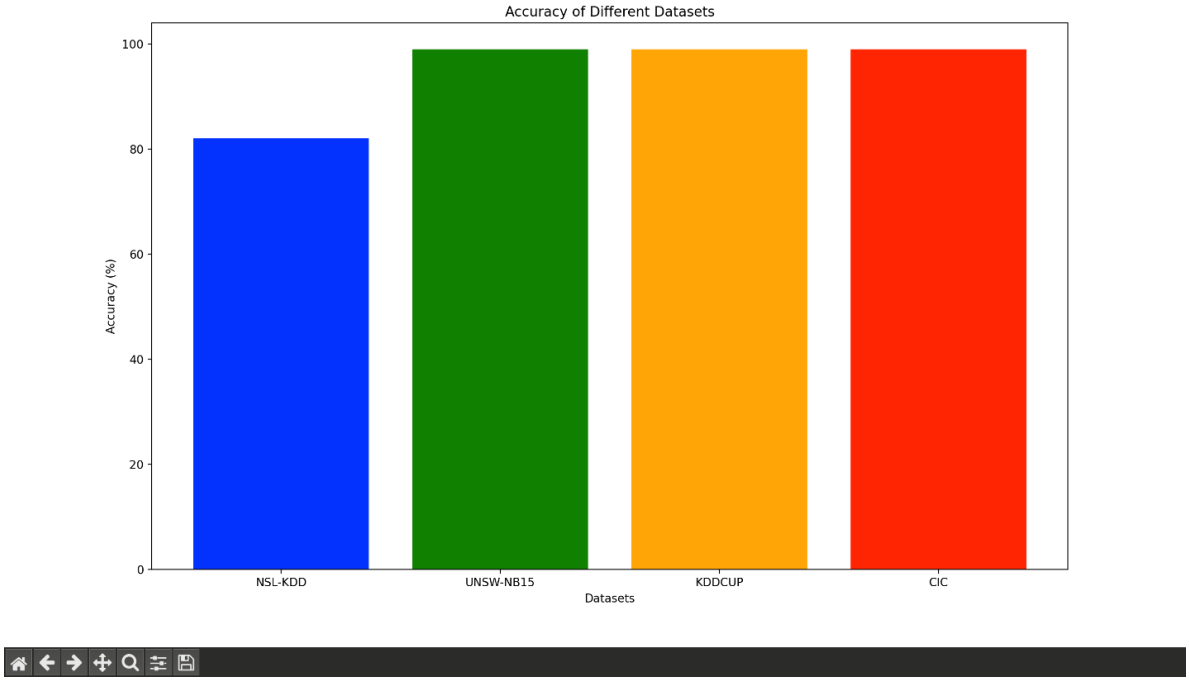
Robustness to Complex Patterns: Transformers proved highly effective in managing the high complexity and volume of network data. The self-attention mechanism was crucial in discerning nuanced patterns that signify anomalies, a task traditional models often struggle with due to noise and data imbalance.

Adaptability to Evolving Threats: The adaptability of transformers to evolving network patterns and new types of attacks reduces the need for frequent retraining, a significant advantage in the dynamic field of cybersecurity.

Operational Insight and Explainability: Beyond mere detection, transformers provide valuable insights into the nature of detected anomalies. This aspect is vital for cybersecurity teams in understanding and mitigating threats effectively.

Superior Handling of Data Challenges: The training and optimization processes addressed several data-specific challenges, such as handling missing values and infinite values, demonstrating the model's robustness in practical scenarios.

The successful implementation of transformer-based models for network anomaly detection marks a significant advancement in cybersecurity. It not only enhances the detection of potential security breaches and system faults but also supports compliance with stringent data security and network integrity regulations.



Screenshot 6*: Results for different datasets*

References

[1] Let's Code AI. (n.d.). Anomaly Detection in Networks Using Transformers: A Comprehensive Guide. Medium. Retrieved from https://medium.com/@letscodeai/anomaly-detection-in-networks-using-transformers-a-comprehensive-guide-36783c95ad73

[2] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018.

[3] Ferriyan, A., Thamrin, A., Takeda, K., & Murai, J. (2021). Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic. Applied Sciences, 11(17), 7868. https://www.mdpi.com/2076-3417/11/17/7868

[4] Tavallaee, M., Stakhanova, N., & Ghorbani, A. A. (2010). Toward Credible Evaluation of Anomaly-based Intrusion-detection Methods. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 40, 516–524. https://doi.org/10.1109/TSMCC.2010.2041647

[5] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2(1), 20. https://doi.org/10.1186/s42400-019-0038-7