Leveraging Adaptive Deep Learning Model for High Precision Worm Detection

PENDYALA SATYA SAI CHARAN

ANOOP LASHIYAL

Agenda

- Abstract
- Introduction
- Problem Statement
- Data Collection and Preprocessing
- Feature Engineering and Selection
- Model Selection and Training
- Model Evaluation and Performance Metrics
- Comparative Analysis of Models
- Results
- Conclusion
- Future work

Abstarct

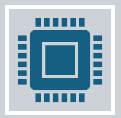
In the face of escalating cyber threats, network worms pose significant and rapidly evolving challenges to network security. Traditional detection methods often lag in adapting to new attack patterns, compromising their effectiveness against emerging threats. This project introduces an innovative approach that integrates adaptive deep learning models, specifically Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM) networks, to enhance the precision of network worm detection. By employing advanced machine learning techniques, robust preprocessing strategies, and real-time adaptive learning, our solution is designed to accurately identify both known and unknown threats. This adaptive framework not only improves anomaly detection but also ensures scalability for real-time deployment in dynamic environments. The result is a high-precision, resilient detection system capable of effectively defending against the evolving landscape of network security threats.

Introduction

Cybersecurity threats, including network worms, have become increasingly sophisticated, necessitating advanced detection systems.

Worms, as self-replicating malware, pose significant risks by spreading rapidly across interconnected systems.

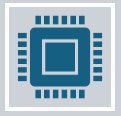
Machine learning models like Decision Trees, Random Forests, and Neural Networks enable the analysis of large datasets to identify malicious patterns.



Adaptive and real-time detection systems are the future, offering robust responses to both known and emerging threats.

Problem Statement

The growing complexity and sophistication of network threats, particularly worms, have highlighted the inadequacies of traditional detection systems.



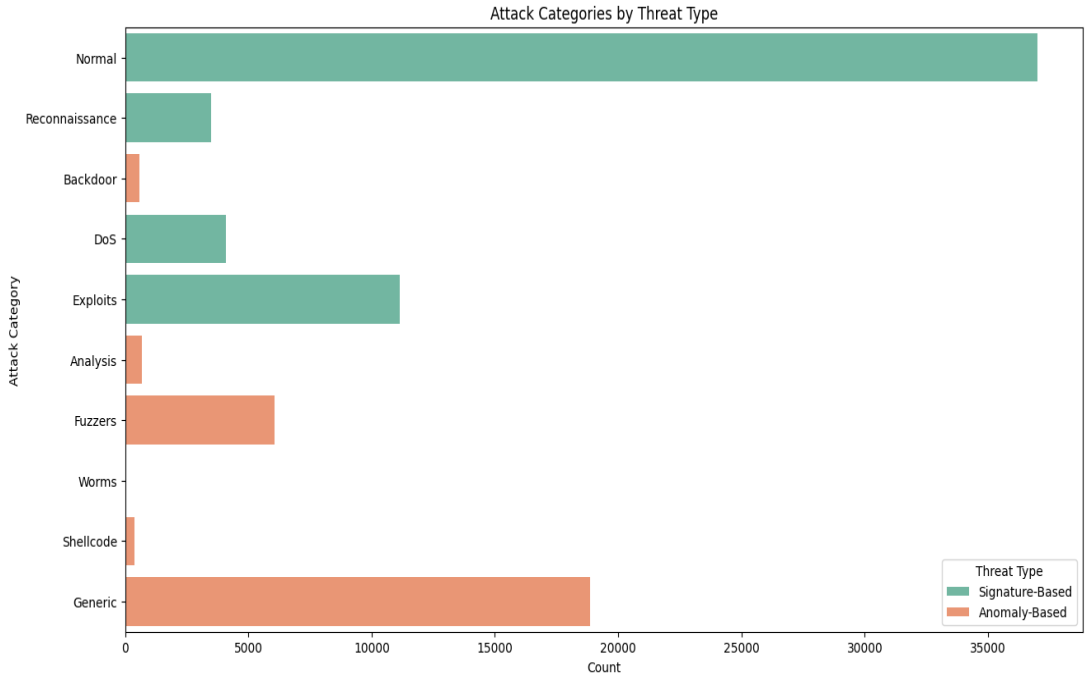
Existing models often struggle to detect zero-day vulnerabilities and adapt to evolving attack patterns, leading to significant security gaps.

The lack of real-time learning and adaptability in conventional systems hinders their effectiveness against dynamic and unknown threats.

This project aims to address these challenges by leveraging adaptive deep learning models for enhanced and scalable threat detection in real-time environments.

Dataset Information

- Dataset Name: UNSW Network Threat Dataset
- Rows: 50,000
- Columns: 44
- Target variables: Network Threat Type (Categorical)
- The data set consists of multiple features representing different aspects of network behavior such as packet size, frequency, connection duration, and various indicators of abnormal activity.
- The target variable indicates different types of network threats, allowing the models to classify different attacks types accurately.



Data Collection

- **Data Collection**
- Acquired the dataset containing network traffic features and labels, including attributes like protocols, packet sizes, and attack categories.
- Data represented both normal traffic and various types of attacks (signature-based and anomaly-based threats).
- **Exploratory Data Analysis (EDA)**
- Identified missing, irrelevant, or duplicate entries in the dataset.
- Analyzed data distributions, feature correlations, and the balance of attack categories to ensure the dataset's quality and usability.

Data Preprocessing

- **Feature Engineering:** Converted categorical variables (e.g., protocol types) into numerical format using encoding techniques.
- **Normalization:** Scaled numerical features to a range of 0 to 1 for consistency across algorithms.
- **Data Imputation:** Filled missing values using mean/mode imputation.
- **Categorization of Threats:** Mapped attack categories into two major groups: signature-based and anomaly-based, simplifying classification tasks.
- Synthetic Minority Over-sampling Techniques (SMOTE): Applied to address class imbalance in the dataset, allowing the model to learn better from minority classes

Dimensionality Reduction

- Applied **Principal Component Analysis (PCA)** to remove noise and reduce feature dimensions while preserving essential information, enhancing computational efficiency.

Feature Engineering and Selection

- **Feature Engineering:**
- **Encoding:** Converted categorical features (`proto`, `service`, `state`) to numerical using **One-Hot Encoding**.
- **Normalization:** Applied log transformation to skewed features like `sbytes` and `dbytes` to reduce outliers.
- **Threat Categorization:** Classified `attack_cat` into `Signature-Based` and `Anomaly-Based` threats for targeted analysis.
- **New Features:** Derived features like packet size and load ratios to enhance model insights.
- **Feature Selection:**
- **Correlation Analysis:** Removed highly correlated features to reduce redundancy and multicollinearity.
- **PCA (Principal Component Analysis):** Retained **15 components** to reduce dimensionality while preserving key information.
- **Feature Importance:** Selected top features (`rate`, `spkts`, `dpkts`, `sload`, `dload`) based on importance from Random Forests and Decision Trees.
- **Variance Thresholding:** Dropped low-variance features with minimal contribution to the models.

Models Selection

Traditional Models Used:

- Decision Tree
- Random Forest
- Gaussian Naive Bayes
- Gradient Boosting Classifier

Deep Learning Models Used:

- Artificial Neural Network (ANN)
- Long Short-Term Memory (LSTM)

Model Training

Logistic Regression

- **Binary and Multiclass Classification**: Logistic regression is leveraged for classifying signature-based threats due to its ability to model binary and multiclass problems. In this project, it forms the baseline for detecting malicious activities in a straightforward and interpretable manner.
- **Feature Contribution Analysis**: Logistic regression's coefficients help identify which features significantly impact the detection of signature-based threats, providing valuable insights into the nature of attacks.
- **Quick Prototyping**: It offers a fast and efficient way to establish a baseline model for identifying both malicious and normal activities, serving as a benchmark for more advanced techniques.

Support Vector Machine (SVM)

- **Margin Maximization for Classification**: SVM identifies signature-based threats by creating an optimal hyperplane, separating malicious activities from normal ones with maximum margin, ensuring robust classification.
- **Kernel Trick for Complex Patterns**: The kernel trick allows SVM to transform data into higher dimensions, effectively detecting non-linear patterns in signature-based and anomaly-based threats.
- **Anomaly Detection Capability**: SVM is effective in isolating anomalies in smaller datasets, which aligns with its ability to classify rare and evolving threat patterns.

Model Training

K-Nearest Neighbors (KNN)

- **Locality-Based Detection**: KNN identifies attack signatures by comparing an instance with its nearest neighbors, making it highly effective for localized and repetitive threat patterns.
- **Adaptive Detection**: By dynamically adjusting the number of neighbors (k), KNN is tuned for both signature-based and anomaly-based detection, depending on the data distribution.
- **Non-Parametric Learning**: KNN's non-parametric nature allows it to detect novel attack patterns without prior assumptions, aiding in the classification of evolving threats.

Gradient Boosting Classifier (GBC)

- **Ensemble Learning for Complex Patterns**: Gradient boosting excels in capturing complex relationships between features, making it highly suitable for detecting both signature-based and anomaly-based threats.
- **Iterative Refinement**: By iteratively improving weak learners, GBC ensures high accuracy in distinguishing between malicious and normal activities in network traffic.
- **Handling Imbalanced Data**: Gradient boosting effectively addresses class imbalance issues in the dataset, making it reliable for identifying minority attack categories.

Model Training

Artificial Neural Networks (ANN)

- **Multi-Layer Perceptrons for Feature Learning**: ANN utilizes dense layers to capture non-linear relationships between features, enhancing the detection of both signature-based and anomaly-based threats.
- **Scalability with Complex Data**: ANN's flexibility in handling high-dimensional data makes it ideal for processing large-scale network traffic datasets with diverse attack types.
- **Generalization to Unknown Threats**: By leveraging hidden layers, ANN generalizes well to detect zero-day vulnerabilities and evolving attack signatures.

Convolutional Neural Networks (CNN)

- **Hierarchical Feature Extraction**: CNN is employed for detecting structural patterns in numerical data, enabling efficient identification of both known and unknown threat signatures.
- **Reduced Feature Engineering**: CNN automatically learns essential features from raw data, reducing the need for manual preprocessing, crucial for dynamic threat detection.
- **Localized Threat Analysis**: By applying convolution operations, CNN excels in identifying localized attack patterns, making it highly effective in signature-based and anomaly-based classifications.

Model Training

Long Short-Term Memory Networks (LSTM)

- **Temporal Dependency Learning**: LSTM captures sequential patterns in network traffic, identifying time-dependent anomalies and evolving worm behaviors.
- **Memory Retention**: The ability to retain long-term dependencies makes LSTM suitable for anomaly-based detection by analyzing prolonged traffic trends.
- **Adaptive Anomaly Detection**: LSTM dynamically learns from past attack sequences, providing robust real-time detection of new and emerging threats.

Transformers

- **Attention Mechanisms**: Transformers focus on the most critical parts of the input data, ensuring accurate detection of anomalies and subtle threat patterns in real-time.
- **Scalability to High-Dimensional Data**: Transformers handle large datasets efficiently, making them ideal for detecting diverse attack types in extensive network traffic logs.
- **Real-Time Adaptive Learning**: By leveraging multi-head attention, Transformers quickly adapt to evolving threats, ensuring scalability and precision in dynamic environments.

Model Evaluation and Performance Metrics

Performance Highlights

- **Traditional Models:** Decision Trees and Random Forests delivered high accuracy (~98-99%) but showed signs of overfitting, limiting generalization.
- **Deep Learning Models:** ANN and LSTM exhibited superior recall and F1-scores, proving effective for adaptive threat detection.
- **Transformer Model:** Achieved the best balance across metrics, making it highly reliable for real-time detection of both signature-based and anomaly-based threats.

Key Insights from Metrics

- **Precision:** Ensures minimal false positives, reducing unnecessary alerts and enhancing system efficiency.
- **Recall:** High recall guarantees the detection of critical threats, crucial for addressing zero-day vulnerabilities.
- **F1-Score & ROC-AUC:** Demonstrated the robustness of the Transformer model in distinguishing between malicious and normal traffic, even in dynamic environments.

Model Comparison

- **Traditional Models:** Faster training times, suitable for smaller datasets, but less adaptive to evolving threats.
- **Deep Learning Models:** Scalable and adaptive, particularly effective for detecting complex patterns in dynamic network environments.
- **Transformer Model:** Best suited for real-time applications with high scalability, precision, and adaptability.

Comparative Analysis of Models

Detection Approach:

- **Traditional Models**: Perform exceptionally well in **signature-based detection**, where patterns of known threats are well-defined and labeled in the dataset.
- **Deep Learning Models**: Excel in **anomaly-based detection**, identifying novel or zero-day threats by learning complex patterns and deviations from normal behavior.

Accuracy and Generalization:

- **Traditional Models**: High accuracy for structured, labeled datasets with minimal noise; however, they may struggle with unseen or ambiguous threats, leading to potential overfitting.
- **Deep Learning Models**: Provide better generalization to unknown threats due to their ability to learn non-linear relationships, though they are sometimes sensitive to noisy data.

Performance Metrics:

- **Traditional Models**: Deliver faster execution times and are computationally less intensive, making them suitable for smaller datasets and real-time signature-based applications.
- **Deep Learning Models**: Require more computational resources and training time but are better at detecting complex anomalies, as reflected in higher **F1-scores** and **ROC-AUC** for anomaly detection.

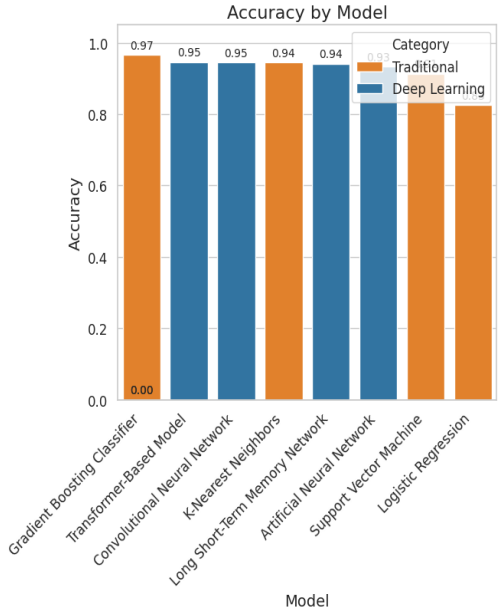
Real-Time Adaptability:

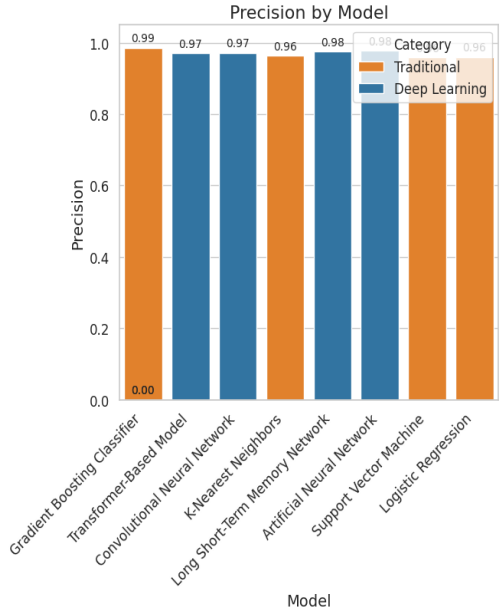
- **Traditional Models**: Limited in real-time adaptability; they rely heavily on predefined rules and patterns, which makes them less effective against evolving threats.
- **Deep Learning Models**: Adapt dynamically to new data trends, especially when using architectures like LSTMs and Transformers that excel in temporal and sequential learning.

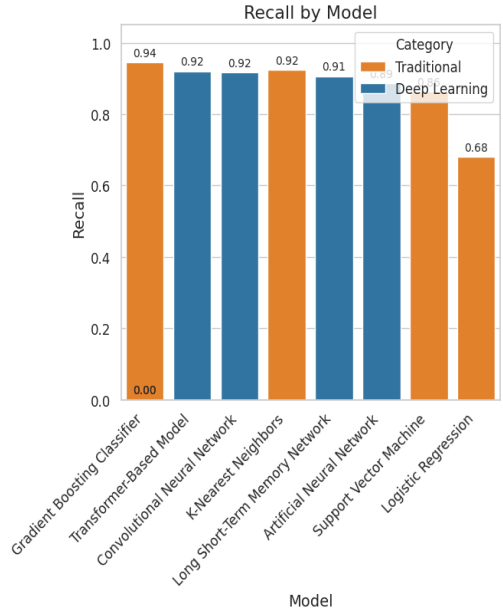
Scalability:

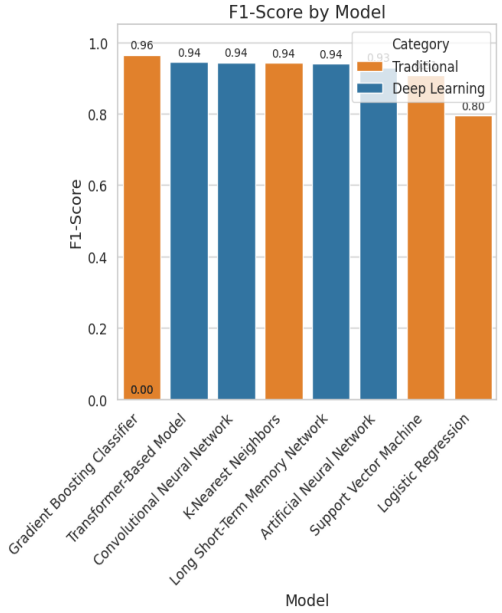
- **Traditional Models**: Less scalable for large-scale or high-dimensional datasets due to reliance on manual feature engineering.
- **Deep Learning Models**: Highly scalable, with the ability to automatically extract features from high-dimensional data, making them ideal for future-proofing detection systems.

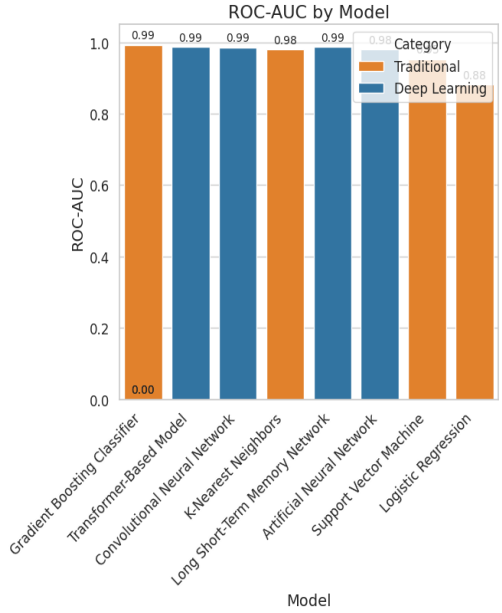
Results


Accuracy by Model
Accuracy
Model
Category
Traditional
Deep Learning
0.97
0.95
0.95
0.94
0.94
0.93
0.00
0.82
Gradient Boosting Classifier
Transformer-Based Model
Convolutional Neural Network
K-Nearest Neighbors
Long Short-Term Memory Network
Artificial Neural Network
Support Vector Machine
Logistic Regression

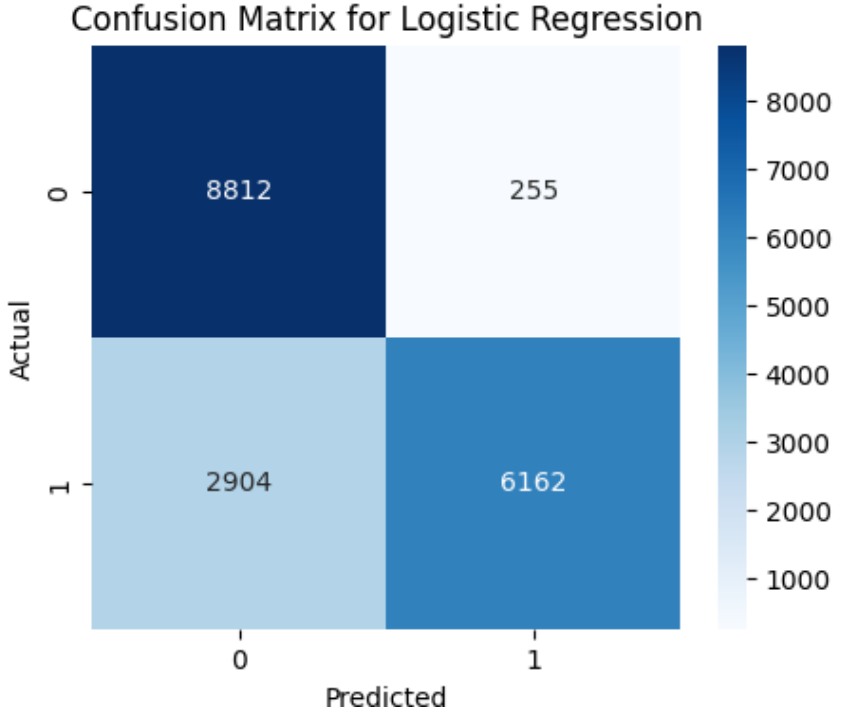

Precision by Model
Precision
Model
Category
Traditional
Deep Learning
0.99
0.97
0.97
0.96
0.98
0.98
0.96
0.00
0.96
Gradient Boosting Classifier
Transformer-Based Model
Convolutional Neural Network
K-Nearest Neighbors
Long Short-Term Memory Network
Artificial Neural Network
Support Vector Machine
Logistic Regression

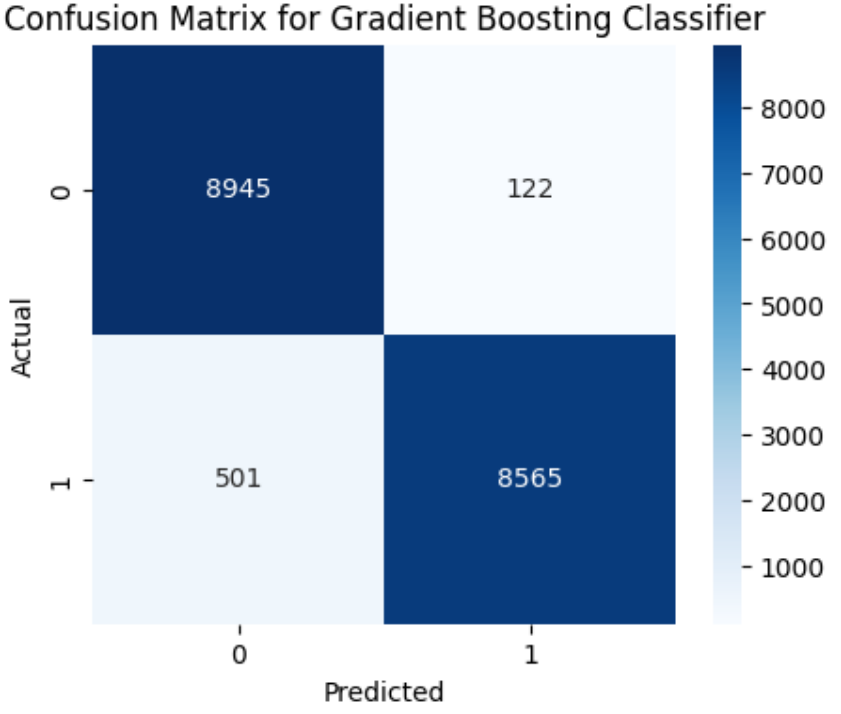

Recall by Model
Recall
Model
Category
Traditional
Deep Learning
0.94
0.92
0.92
0.92
0.91
0.88
0.89
0.68
0.00
Gradient Boosting Classifier
Transformer-Based Model
Convolutional Neural Network
K-Nearest Neighbors
Long Short-Term Memory Network
Artificial Neural Network
Support Vector Machine
Logistic Regression

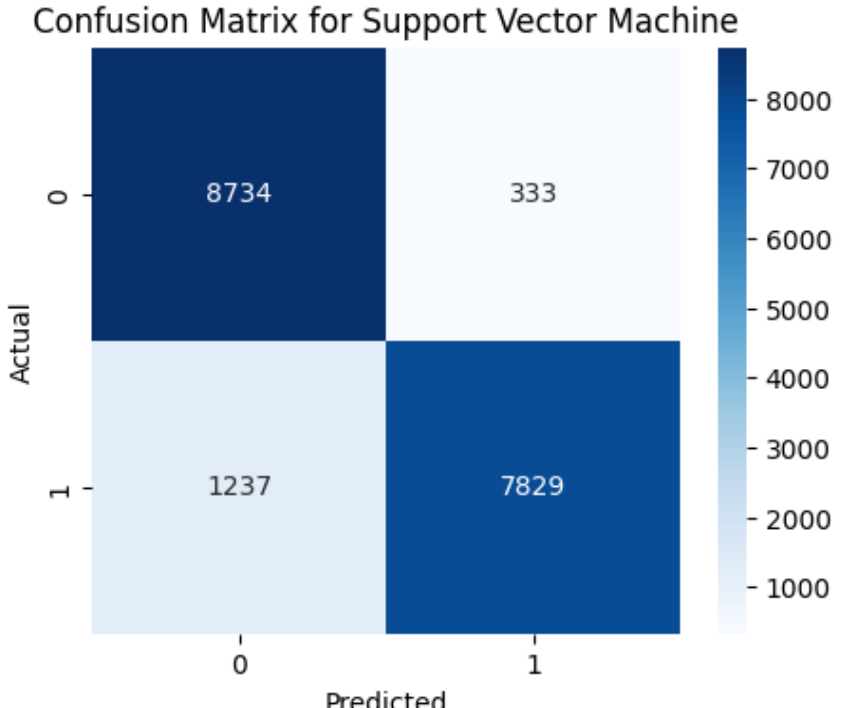

F1-Score by Model
F1-Score
Model
Category
Traditional
Deep Learning
0.96
0.94
0.94
0.94
0.94
0.91
0.80
0.00
Gradient Boosting Classifier
Transformer-Based Model
Convolutional Neural Network
K-Nearest Neighbors
Long Short-Term Memory Network
Artificial Neural Network
Support Vector Machine
Logistic Regression

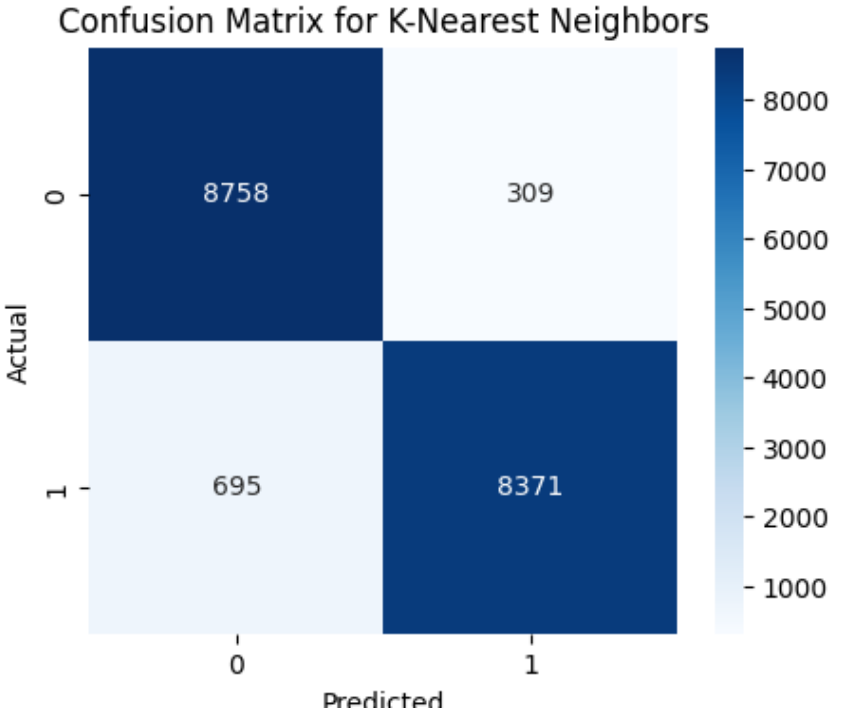

ROC-AUC by Model
ROC-AUC
Model
Category
Traditional
Deep Learning
0.99
0.99
0.99
0.98
0.99
0.98
0.88
0.00
Gradient Boosting Classifier
Transformer-Based Model
Convolutional Neural Network
K-Nearest Neighbors
Long Short-Term Memory Network
Artificial Neural Network
Support Vector Machine
Logistic Regression

Confusion Matrix

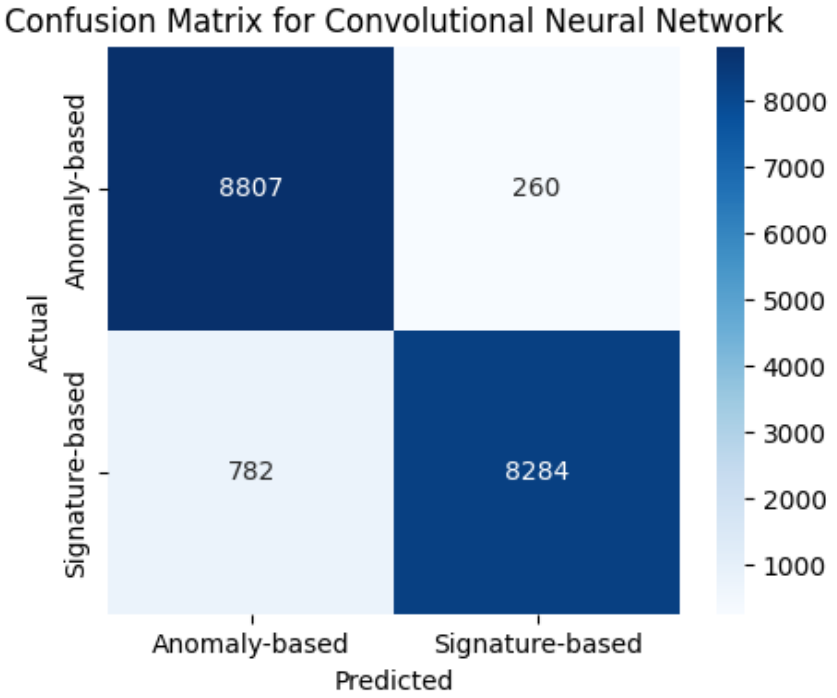


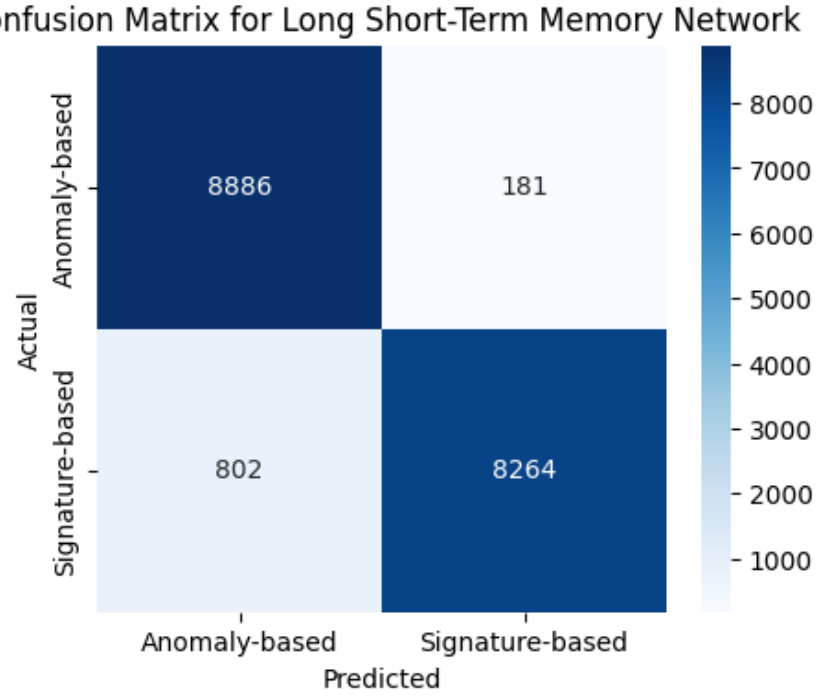


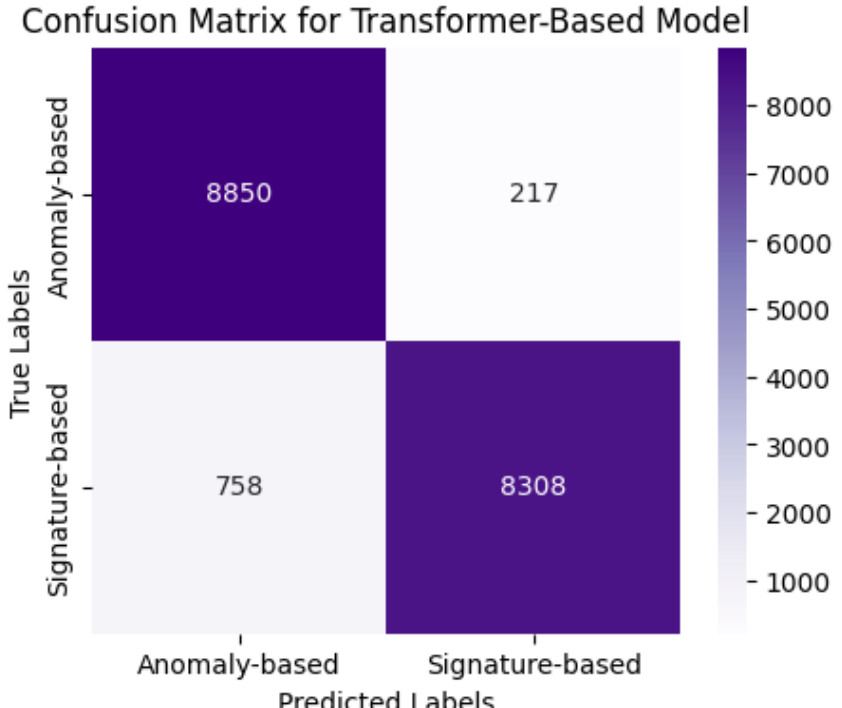


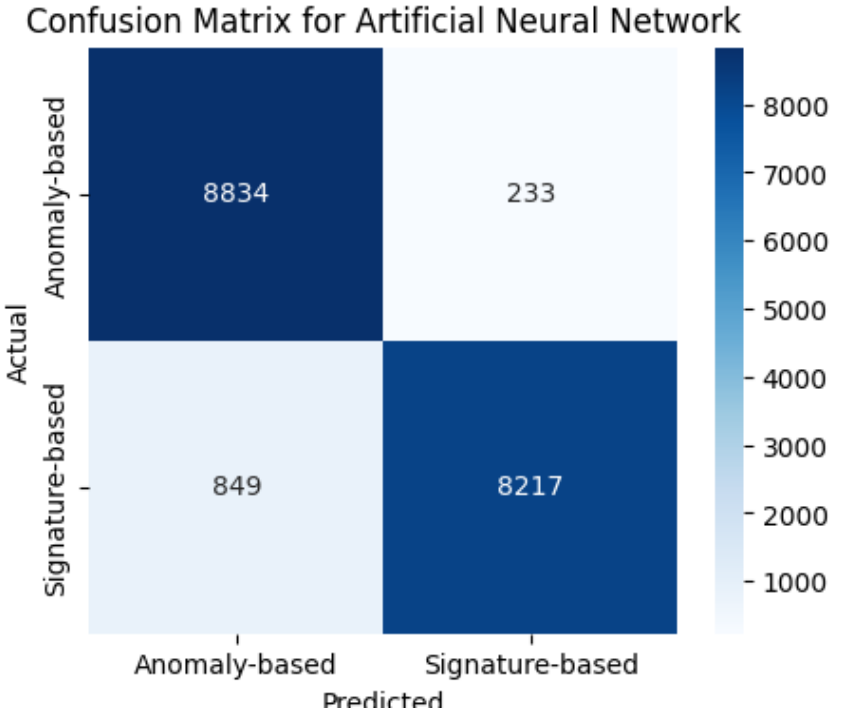


Confusion Matrix









Conclusion

Top Performers:

The **Transformer-Based Model** and **Gradient Boosting Classifier** were the most effective, achieving the highest accuracy, precision, recall, and F1-scores, ideal for real-world threat detection.

Threat Classification:

The models successfully categorized threats into **signature-based** and **anomaly-based** with balanced performance, leveraging efficient preprocessing to handle imbalanced data.

Insights from Visualizations:

Comprehensive visualizations provided clear comparisons of model metrics, aiding in understanding model performance and enabling informed decisions.

Scalability:

The **Transformer-Based Model** is well-suited for real-time detection due to its ability to capture complex patterns, while traditional models like **Gradient Boosting** offer computational efficiency for simpler environments.

Real-World Impact:

This project delivers a scalable and efficient pipeline for threat classification, combining accuracy, speed, and practical implementation potential.

Future Work

Enhancement of Transformer-Based Model:

- Expand the transformer model to handle **real-time streaming data** for proactive threat detection in dynamic environments.
- Incorporate pre-trained transformer architectures like **BERT** or **RoBERTa**, fine-tuned for cybersecurity tasks to improve performance further.

Development of API Interface:

- Build an **API service** to integrate the threat detection system into real-world applications, enabling seamless deployment across enterprise-level cybersecurity platforms.

Integration with Advanced Systems:

- Combine the model with **SIEM (Security Information and Event Management)** systems to enhance its real-time threat response capabilities.
- Incorporate anomaly detection from IoT devices and cloud platforms for broader application coverage.

Explainability and Interpretability:

- Implement **SHAP** or **LIME** for explainable AI (XAI), helping stakeholders understand model predictions and making the system more trustworthy in critical cybersecurity applications.

Thank You