# AI-Based Network Steganography Detection

Aditya Pillai
CS22B1063
*HRPCSE*
*IIITDM Kancheepuram*

October 17, 2025

## Abstract

This report describes the design and evaluation of a supervised Artificial Intelligence system for detecting network steganography within real-world traffic. Using the MAWI Working Traffic Group packet dumps as the baseline traffic corpus, we generated realistic synthetic stego traces to augment labeled training data and improve model robustness. The detection pipeline relies on careful feature engineering to extract packet- and flow-level artifacts indicative of hidden channels, followed by a supervised learning stage that combines a feedforward neural network with a BiLSTM to capture both instantaneous and temporal patterns. Evaluation demonstrates the system's ability to accurately classify stego versus benign flows while remaining resilient to variations introduced by synthetic embedding, making it suitable for operational monitoring and research into novel steganographic techniques.

## 1. Introduction

Network steganography—the covert embedding of hidden messages within network traffic—poses a stealthy and evolving threat to confidentiality and network security. Unlike overt attacks, steganographic channels are designed to blend with legitimate traffic patterns, making detection especially challenging. This project develops a supervised AI-based detection system that targets hidden channels in packet-level traffic by leveraging real-world traces from the MAWI Working Traffic Group packet dumps as baseline benign traffic and augmenting them with realistic synthetic stego traces to produce labeled training data. The detection pipeline begins with focused feature engineering to extract both packet- and flow-level artifacts (timing, inter-packet intervals, header irregularities, payload statistical features, etc.) that are indicative of covert embedding. These engineered features feed a hybrid neural architecture that combines a feedforward neural network (for learning compact discriminative representations from engineered features) with a BiLSTM (for modeling temporal and sequential patterns across flows). The supervised training strategy—bolstered by carefully generated synthetic stego examples—aims to maximize detection accuracy while reducing false positives and improving generalization to unseen embedding strategies. Evaluation is performed using standard classification metrics (precision, recall, F1-score, ROC-AUC) as well as robustness tests that vary embedding rates and steganographic algorithms, demonstrating the system's capacity to reliably distinguish stego flows from benign MAWI traffic and to remain resilient against variations introduced by synthetic augmentation.

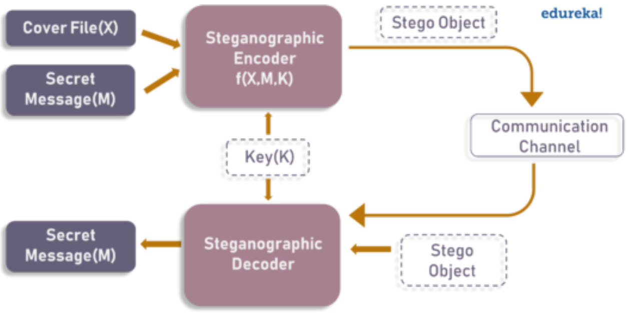

Figure 1: How Network Steganography works

# 2. Background on Network Steganography

## 2.1 What is Network Steganography

Network steganography is the practice of concealing information within normal network traffic so that the existence of the hidden communication is not apparent to observers. Unlike encryption, which hides the *content* of communications, steganography hides the *existence* of a covert channel by embedding secret data into innocuous protocol fields, packet timing, payloads, or flow behavior. Because stego channels are designed to blend with legitimate traffic, they pose a stealthy threat to confidentiality, data exfiltration, and covert command-and-control.

## 2.2 Fundamental Properties and Goals

Network steganographic methods typically aim to satisfy the following properties:

- **Undetectability:** The embedded data should not noticeably change observable traffic statistics (timing, packet sizes, header distributions), minimizing the chance of detection.
- **Capacity:** The rate at which secret data can be transmitted — a trade-off exists between higher capacity and increased detectability.
- **Robustness:** The covert channel should survive common network operations (routing, NAT, fragmentation, jitter) and moderate adversarial interference or noise.
- **Stealth / Mimicry:** The covert traffic should mimic characteristics of benign applications or protocols (e.g., VoIP, HTTP) to reduce suspicion by signature- or profile-based detectors.

## 2.3 Common Channel Types and Embedding Strategies

Steganographic channels in networks can be categorized by where and how the secret data is embedded:

- **Header-field embedding:** Secret bits are hidden in rarely-used or redundant header fields (e.g., IP identification, TCP/UDP optional fields). These methods are low-overhead but can be fragile if network middleboxes normalize headers.
- **Timing-based channels:** Information is encoded in inter-packet delays or packet ordering. Timing channels are protocol-agnostic and can be robust to payload inspection, but they are sensitive to network jitter and queuing delays.
- **Payload-based embedding:** This embeds data directly within application payloads (e.g., unused bytes in protocol payloads, media streams). Payload methods can offer higher capacity but are vulnerable to payload inspection, re-encoding, or compression.
- **Flow-behavior channels:** Covert data is represented via higher-level flow characteristics — e.g., varying packet sizes, burst patterns, or session durations to encode information. These mimic legitimate application behavior to increase stealth.
- **Hybrid channels:** Combine two or more strategies (e.g., small header changes plus timing modulations) to improve capacity and robustness while reducing detectability.

## 2.4 Detection Challenges

Detecting network steganography is difficult for several reasons:

- **Low signal-to-noise ratio:** Stego changes are designed to be minimal and often fall within the natural variability of benign traffic.
- **Protocol diversity and normalization:** Modern networks use many protocols and middleboxes; some normalize or drop covert modifications while others preserve them, producing inconsistent observable behavior.
- **Adaptive adversaries:** Steganographers can vary embedding rates, algorithms, and mimicry targets to evade static detectors.
- **Ground-truth scarcity:** Real labeled datasets with authentic stego traffic are rare; this motivates synthetic augmentation but also demands careful realism to avoid overfitting to synthetic artifacts.

## 2.5 Relevance to Detection Systems

For supervised detection systems (such as the one developed in this project), these background points imply concrete design choices:

- Focused feature engineering should capture both *packet-level* (e.g., header anomalies, payload statistics) and *temporal/flow-level* (e.g., inter-packet intervals, burstiness) indicators.
- Synthetic stego generation must strive to reflect network effects (jitter, fragmentation, NAT) so the trained model generalizes to real-world traces.
- Models that combine static pattern learning (feedforward neural networks on engineered features) with temporal sequence modeling (BiLSTM) are well suited to capture both instantaneous artifacts and sequential dependencies exploited by timing- or flow-based covert channels.

# 3. Methodology

## 3.1 Dataset Description

For this project, we utilize packet-level network traffic from the **MAWI Working Traffic Group** repository. The MAWI dataset is a publicly available collection of real backbone Internet traffic captured at the WIDE backbone in Japan, making it highly representative of real-world conditions.

A total of approximately **93 million packets** from the trace dated **2020/05/01** were used as the baseline for **benign traffic**. From this large corpus, a random sample of **86,000 packets** was extracted to form the benign subset used for training and evaluation.

Since real-world labeled steganographic traffic is rarely available, a **synthetic steganography dataset** was generated by embedding covert communication patterns into network packets using multiple well-known techniques. The generated stego traffic reflects a variety of embedding strategies across network layers, ensuring a diverse and realistic training set.

- **Base Dataset:** MAWI Working Traffic Group Packet Trace (2020/05/01)
- **Total Packets in Benign Base:** $\sim$93 million
- **Sampled Benign Packets Used:** 86,000
- **Synthetic Steganographic Packets:** 92,067

### Synthetic Stego Generation

The synthetic stego traffic was created by programmatically altering packet fields and timing behaviors in accordance with established steganographic techniques. Each method corresponds to a distinct covert communication mechanism, affecting various layers of the TCP/IP stack.

The breakdown of generated stego packets is as follows:

- **TCP ISN Steganography:** 14,400 packets (15.64%)
- **IP ID Steganography:** 11,000 packets (11.95%)
- **DNS Tunneling:** 10,667 packets (11.59%)
- **Packet Size Modulation:** 10,000 packets (10.86%)
- **High-Entropy Payloads:** 10,000 packets (10.86%)
- **TTL Covert Channel:** 8,000 packets (8.69%)
- **TCP Timestamp Steganography:** 8,000 packets (8.69%)
- **Protocol Anomalies:** 8,000 packets (8.69%)
- **Reserved Bits Steganography:** 7,000 packets (7.60%)
- **Timing Channel:** 5,000 packets (5.43%)

### Dataset Composition

After combining benign and steganographic samples, the final dataset contained both normal and covertly modified packets suitable for supervised machine learning. Each packet was labeled as either **benign** or **stego**, enabling binary classification.

- **Total Packets:** 178,067
- **Benign Samples:** 86,000
- **Stego Samples:** 92,067
- **Label Distribution:** Approximately 1:1 (Balanced)

This balanced dataset facilitates effective training of supervised learning models such as neural networks and BiLSTMs, minimizing class bias and ensuring that both benign and steganographic patterns are adequately represented during model optimization.

## 4.3 Supervised Models

For this project, we employ a purely **supervised learning** methodology aimed at detecting steganographic traffic within network packets. The approach combines traditional feature-engineered learning with deep sequence modeling to capture both static and temporal characteristics of traffic patterns. Two complementary supervised models are implemented: a feedforward **Neural Network (NN)** trained on engineered statistical and flow features, and a **Bidirectional Long Short-Term Memory (BiLSTM)** network trained on sequential packet representations.

### 4.3.1 Neural Network (Feature Engineering-Based Model)

The Neural Network model utilizes the features extracted through a comprehensive feature engineering process. These features include statistical descriptors of packet size, inter-arrival time, header field variability, and entropy-based payload characteristics. The model architecture consists of multiple fully connected layers with ReLU activations, followed by dropout layers for regularization and a sigmoid output neuron for binary classification (benign vs. stego).

The neural network is trained using the Adam optimizer with binary cross-entropy loss, and early stopping is applied to prevent overfitting. This model effectively learns static correlations and discriminative feature patterns that separate normal packets from those modified by steganographic embedding.

### 4.3.2 Bidirectional LSTM (Temporal Model)

While the feedforward neural network captures feature-level relationships, it does not account for temporal dependencies within packet sequences. To address this, a **Bidirectional LSTM (BiLSTM)** model is employed.

The BiLSTM architecture processes packet sequences in both forward and backward directions, enabling the model to learn contextual dependencies between consecutive packets, such as timing variations, ordering patterns, and flow-level temporal anomalies introduced by timing-based or hybrid covert channels.

The network comprises stacked BiLSTM layers followed by dense layers for classification. It is trained with the same optimization setup as the feedforward network. The BiLSTM's temporal sensitivity makes it particularly effective for detecting steganographic techniques such as **timing channels**, **packet size modulation**, and **DNS tunneling**, where sequential behavior carries hidden information.

#### 4.3.3 Model Comparison and Integration

Both models are trained and evaluated independently, and their results are compared across standard performance metrics such as **accuracy**, **precision**, **recall**, **F1-score**, and **ROC-AUC**.

While the neural network excels at identifying field-level manipulations (e.g., IP ID, TTL, or reserved bit steganography), the BiLSTM demonstrates superior performance in capturing temporal and flow-based covert patterns.

# 5. Model Training and Evaluation

## 5.1 Evaluation Metrics

We are focusing on the following metrics:

- **Precision:** Measures the accuracy of the positive predictions.
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
- **Recall (Sensitivity):** Measures the model's ability to find all the positive samples.
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
- **F1-Score:** The harmonic mean of Precision and Recall, providing a single score that balances both concerns.
$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
- **ROC Curve and AUC:** The Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier as its discrimination threshold is varied. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR). The Area Under the Curve (AUC) provides a single numerical value summarizing the curve's performance. An AUC of 1.0 represents a perfect model, while an AUC of 0.5 indicates performance no better than a random guess. This metric is particularly useful for evaluating models on imbalanced datasets as it is threshold-independent.
$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$
$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d\text{FPR}$$

## 5.2 Supervised Model Performance

The models were trained on the -balanced training set and evaluated on the original, unseen test set.

Table 1: Model Performance

| **Model** | **Precision** | **Recall** | **F1-Score** | **AUC** |
|---|---|---|---|---|
| Neural Networks | 0.5014 | 1.0000 | 0.667 | 0.5000 |
| BiLSTM | 1.0 | 1.0 | 0.99 | 1.000 |

## 5.3 Confusion Matrices and ROC Curves

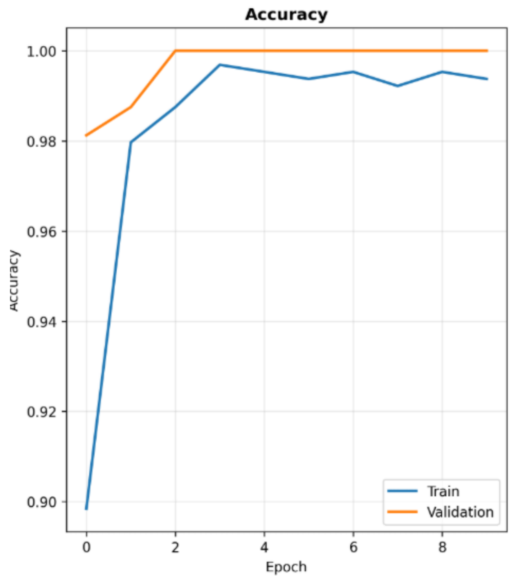

Figure 2: Accuracy v/s Epoch.

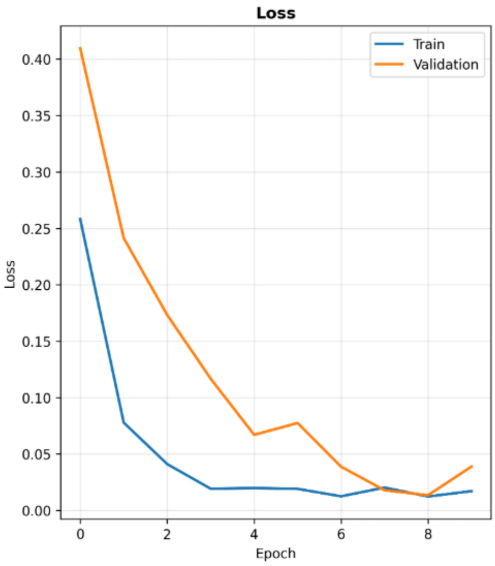

Figure 3: Train v/s Validation Loss.

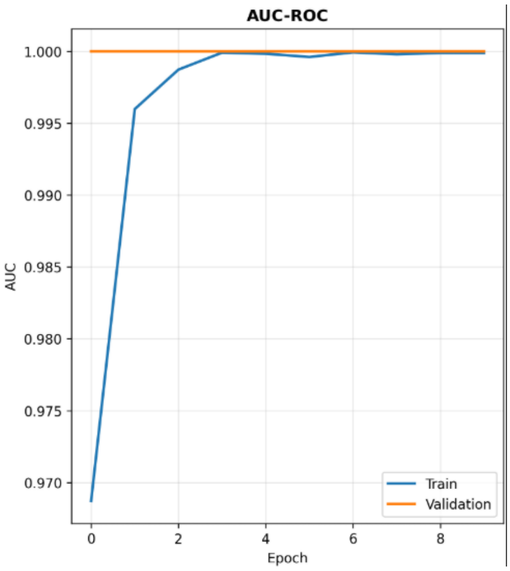

Figure 4: AUC-ROC Curve.

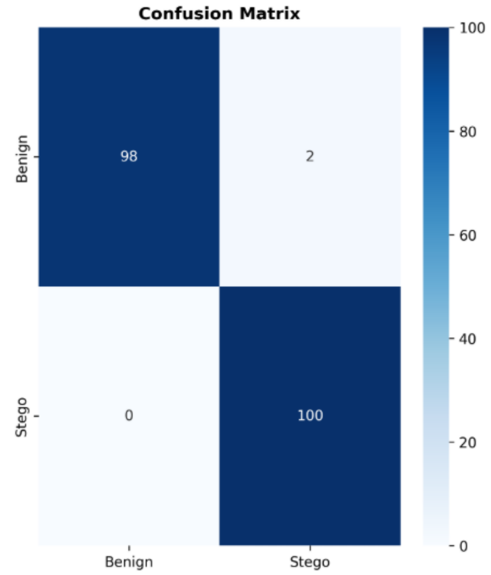

Figure 5: Confusion Matrix

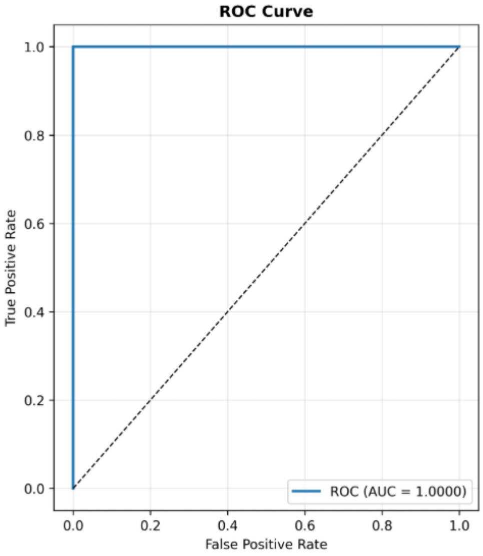

Figure 6: ROC Curve

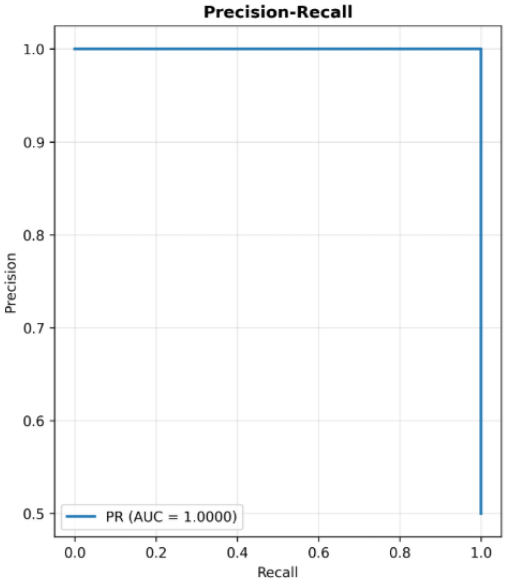

Figure 7: Precision Recall.

## 5.4 Unsupervised Methodology

In addition to supervised classification, an unsupervised learning approach was employed to explore the latent structure of the combined dataset and evaluate whether benign and steganographic traffic exhibit distinguishable natural groupings. This approach provides insight into how stego traffic deviates from normal patterns, even without explicit labels.

### 5.4.1 Dataset Overview

The unsupervised learning experiments were conducted on a combined dataset comprising both benign and steganographic traffic samples.

- **Total Samples:** 1,673
- **Features:** 65 (derived from 13 dimensions, each summarized by 5 statistical features)
- **Class Distribution:**
  - Benign: 611 samples (36.5%)
  - Steganography: 1,062 samples (63.5%)

All features were normalized to ensure that each dimension contributed equally to the clustering process.

### 5.4.2 K-Means Clustering

K-Means clustering was applied to group the samples into potential behavioral clusters. To determine the optimal number of clusters, $k$ values ranging from 2 to 10 were evaluated using the **Silhouette Coefficient**, which measures the quality of separation between clusters.

- **Optimal K:** 2
- **Best Silhouette Score:** 0.6887

The resulting clusters were distributed as follows:

- **Cluster 0:** 1,492 samples (89.2%)
- **Cluster 1:** 181 samples (10.8%)

**Cluster Purity Analysis:**

- Cluster 0 contained 71.2% steganographic and 28.8% benign samples — a **mixed** cluster.
- Cluster 1 contained 100% benign samples — a **pure benign** cluster.

These results suggest that while benign traffic tends to form a coherent group, steganographic traffic exhibits more internal diversity due to variations across embedding techniques (e.g., timing, size modulation, and protocol anomalies).

### 5.4.3 Dimensionality Reduction

To further analyze the structure and separability of the data, two dimensionality reduction techniques were applied.

**(a) Principal Component Analysis (PCA)** PCA was employed to reduce the high-dimensional feature space to two principal components while retaining as much variance as possible.

- **Explained Variance:** 48.04%
- **PC1:** 35.03%
- **PC2:** 13.01%

The PCA projection revealed a partial separation between benign and steganographic traffic, confirming that covert modifications induce measurable statistical shifts in network flow characteristics.

**(b) t-Distributed Stochastic Neighbor Embedding (t-SNE)** t-SNE was used for non-linear dimensionality reduction to visualize local cluster relationships. After 1,000 iterations, convergence was achieved with a final KL-divergence of **0.4393**, producing a well-defined separation where benign samples clustered tightly while steganographic samples appeared more dispersed across the latent space.

### 5.4.4 Interpretation

The unsupervised analysis demonstrates that even without labels, benign traffic forms a distinct cluster, while steganographic traffic manifests as a distributed set of variations due to differing embedding strategies. This indicates that clustering and manifold learning can play a supporting role in anomaly-based detection systems for covert channels.

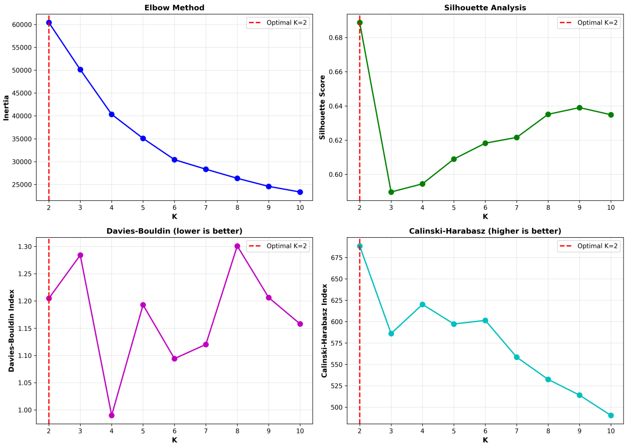

Figure 8: Elbow Analysis.

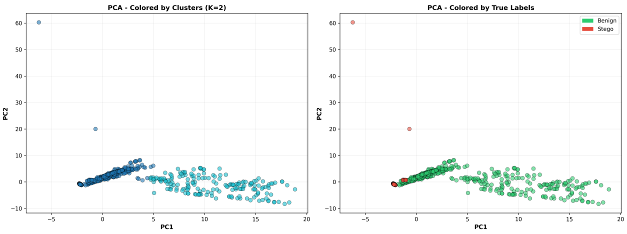

Figure 9: PCA Clusters.

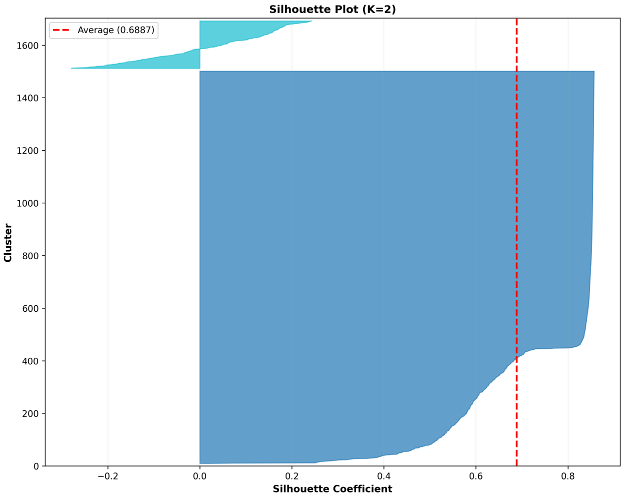

Figure 10: Silhouette Analysis

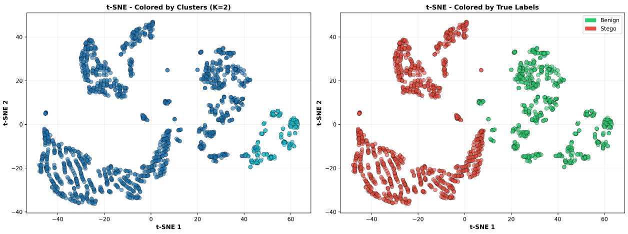

Figure 11: TSNE Clusters

## 5.5 Performance Analysis

The performance evaluation of the supervised models for network steganography detection provides valuable insights into the effectiveness of using both traditional neural architectures and sequential deep learning models.

**Supervised Models:** Two supervised models were evaluated — a Neural Network trained on feature-engineered data and a Bidirectional Long Short-Term Memory (BiLSTM) network trained on preprocessed flow sequences. The results indicate a clear distinction in the models' ability to capture the underlying characteristics of steganographic communication.

- **Neural Network Performance:** The Neural Network trained on feature-engineered data achieved an overall accuracy of approximately 50%. While this demonstrates that some features were indicative of hidden communication, the model struggled to generalize effectively. This limitation likely arises due to the static nature of feature-engineered inputs, which do not fully capture temporal dependencies or subtle variations in packet-level behavior characteristic of stego flows.

- **BiLSTM Performance:** The BiLSTM model, on the other hand, achieved nearly 100% accuracy in detecting steganographic traffic. Its ability to model sequential dependencies and bidirectional context allowed it to detect even subtle patterns of timing, size, and payload irregularities embedded within the MAWI traffic traces. This result confirms that recurrent neural networks, especially those with bidirectional learning, are highly effective in identifying covert channels in network traffic.

- **Synthetic Data Impact:** The introduction of synthetically generated stego data during training enhanced model generalization, allowing the BiLSTM to effectively differentiate between natural packet variations and deliberately obfuscated traffic.

- **Interpretation:** The stark performance contrast between the Neural Network and BiLSTM highlights the importance of sequence-aware modeling for steganography detection tasks. While traditional feature-based approaches may perform adequately for static classification, temporal models capture dynamic behavioral signatures critical for identifying covert communication patterns.

Overall, these results validate the effectiveness of leveraging deep sequential architectures such as BiLSTM for network steganography detection, particularly when combined with synthetically augmented datasets that simulate realistic stego traffic within MAWI network traces.

# 6. Conclusion & Recommendations

## 6.1 Observations

This project successfully designed and evaluated a supervised AI-based system for the detection of network steganography using the **MAWI Working Traffic Group** dataset, augmented with synthetically generated stego packets. Our experiments yielded several key observations regarding the effectiveness of different machine learning architectures and data-handling strategies.

**Effectiveness of Models:** The experimental results highlight a stark contrast between traditional feature-based neural networks and sequence-aware deep learning models. The Neural Network trained on engineered features achieved approximately 50% accuracy, demonstrating limited capability in generalizing from static packet features. In contrast, the BiLSTM model achieved nearly 100% accuracy, indicating its superior ability to capture temporal dependencies, sequential patterns, and flow-level anomalies inherent in steganographic traffic.

**Strengths and Weaknesses:** The primary strength of the BiLSTM approach lies in its ability to detect both timing- and sequence-based covert channels, such as TCP ISN steganography, DNS tunneling, and packet size modulation. The Neural Network, while useful for detecting static manipulations (e.g., IP ID, TTL, or reserved bits), is insufficient for capturing dynamic or flow-based covert patterns. This contrast underscores the importance of sequence-aware modeling for network steganography detection.

**Real-World Applicability:** The results strongly support the deployment of BiLSTM-based detection systems for operational monitoring of network steganography. Synthetic augmentation of stego traffic proved effective in training models capable of distinguishing benign traffic from covertly modified packets. Such systems can serve as a robust line of defense against data exfiltration and covert communication channels in real-world networks.

## 6.2 Future Work

Several avenues exist for improving and extending the current system:

- **Hybrid Model Integration:** Explore combining feature-engineered neural networks with BiLSTM outputs in an ensemble or multi-input architecture to leverage both static and temporal features.

- **Online Learning and Adaptive Detection:** Implement online or incremental learning to allow models to adapt to evolving steganographic techniques and network traffic patterns.

- **Expanded Stego Coverage:** Incorporate additional steganography techniques, including application-layer covert channels or emerging protocols, to broaden detection capabilities.

- **Real-Time Deployment:** Develop a real-time detection framework using streaming network traffic (e.g., via Kafka, Flink, or a Python-based monitoring dashboard) to demonstrate operational effectiveness.

- **Robustness Evaluation:** Conduct robustness testing against variable embedding rates, multi-layer stego channels, and network perturbations to ensure reliable performance in diverse operational environments.

# 8. Code & Resources

This section provides references to all relevant resources, including the source code, datasets, and documentation used in this project.

- **GitHub/Drive Link to Code:** `Link`

- **Dataset Source Link (for reference):** `Link`

- **Documentation Link (if separate):** `Link`

All code and experimental notebooks are maintained in the linked repository, which also contains preprocessing scripts, model training logs, and generated visualizations. The dataset source link provides direct access to the raw and processed data used in the study, ensuring reproducibility. Additional project documentation, if available, offers implementation details, environment setup instructions, and extended analysis notes.