Adaptive Crypto-Steganography With Machine Learning for real-Time Network traffic

Emenike Chukwu[1], Agozie Eneh[2], Nnennaya Ogbonnaya[3], Precious Udechukwu[4]

[1] *Department of Software Engineering / Federal University of Technology, Ikot-Abasi, Akwa-Ibom State. Nigeria.*
[2] *Department of Computer Science / University of Nigeria, Nsukka. Enugu State, Nigeria*
[3] *Department of Computer Science / University of Nigeria, Nsukka, Enugu State, Nigeria*
[4] *Department of Computer Science / University of Nigeria, Nsukka, Enugu State, Nigeria*

Abstract:
The increasing complexity of intrusion detection and traffic analysis methods makes it difficult to secure data transfer over open networks. Encrypted traffic patterns can nonetheless expose communication to adversary examination even though cryptography protects message content. A machine learning-enhanced adaptive crypto-steganography framework for real-time network traffic is presented in this work. Standard traffic monitoring tools like Wireshark and tcpdump are used to record live network packets.

Materials and Methods*:* A Python-based pipeline that uses Scapy for packet parsing, modification, and feature extraction is then used to process the packets. A supervised machine learning model built using scikit-learn is trained using key traffic variables, such as packet size distribution, inter-packet time, protocol behavior, and entropy-based measures. While lightweight encryption guarantees data secrecy before embedding, the trained model dynamically chooses the best steganographic embedding schemes and parameters during runtime. Detection probability, steganalysis resistance, robustness, and computing overhead measures are used to evaluate performance in an experimental setting with different network circumstances and traffic loads.

Results*:* The findings show that, in comparison to static steganographic techniques, the suggested adaptive architecture considerably lowers detectability and boosts resilience while preserving low latency appropriate for real-time deployment.

Conclusion: A machine learning-enhanced adaptive crypto-steganography framework for real-time network data was provided in this paper. Through dynamic embedding strategy selection based on real-time traffic parameters, the suggested approach increases robustness and stealth against contemporary steganalysis techniques. Deep learning-based decision models and their implementation in large-scale, high-speed network systems will be the subject of future research.
Key Word*: Crypto-Steganography, Machine Learning, Network Security, Real-Time Traffic, and Covert Communication.*

1.0 Introduction

The need for strong data security measures has increased due to the quick growth of cloud computing, network-based services, and Internet of Things (IoT) applications. Because it guarantees data secrecy, integrity, and authentication, cryptography continues to be the foundation of secure communication. Nevertheless, even though the content of messages is protected by cryptographic techniques, encrypted traffic is still visible on the network and can be examined to determine endpoints, communication patterns, protocol usage, and user behavior [1], [5], and [10]. Even when robust encryption is used, this kind of exposure makes private communications susceptible to traffic analysis and categorization assaults.

Steganography has been suggested as an additional security method to lessen this restriction, hiding the fact that communication is taking place at all. In order to lessen the possibility of detection, network steganography embeds hidden information into valid traffic flows, such as packet headers, timing behavior, or protocol semantics [2], [6]. The majority of network steganographic techniques currently in use, despite their promise, are based on static embedding rules that are unable to adjust to changing network conditions, traffic fluctuations, or protocol behavior.

Steganalysis approaches have been greatly enhanced by recent developments in machine learning and deep learning, which allow attackers to uncover secret communication by spotting minute statistical and temporal irregularities in network data [3], [7], and [11]. Many static and conventional steganographic techniques are becoming less and less successful as a result of these sophisticated detection systems. In order to improve stealth, robustness, and resilience against contemporary steganalysis assaults, there is a rising need for intelligent and adaptive crypto-steganography frameworks that can dynamically modify embedding techniques based on real-time traffic parameters.

2.0 Related Works

Cryptographic techniques for preserving data integrity and secrecy over open and hostile networks have been thoroughly studied in network security research [1], [4], and [12]. Despite the mathematical security of contemporary cryptographic algorithms, numerous studies have shown that encrypted traffic is still vulnerable to attacks such as flow correlation, traffic analysis, and protocol fingerprinting [5, 10, and 13]. These attacks highlight a significant drawback of cryptography-only solutions by taking advantage of observable information rather than encrypted payloads.

By enclosing secret information in regular network traffic, network steganography has become a viable method for secret communication [2], [6], and [14]. Packet header manipulation, inter-packet time, packet ordering, and redundant protocol fields were the main focuses of early network steganographic techniques [6], [15]. Despite being successful in controlled settings, these methods usually use fixed embedding tactics, which leaves them open to rule-based detection and statistical analysis.

Steganalysis approaches have changed dramatically in tandem with the quick development of machine learning. By picking up on intricate traffic patterns and minute irregularities, supervised and deep learning models have been effectively used to uncover secret communication [3], [7], [11], and [16]. The efficacy of static concealment strategies has been further challenged by the strong performance of convolutional neural networks, recurrent neural networks, and ensemble classifiers in identifying network steganography [17], [18].

Adaptive steganography, which dynamically modifies embedding parameters based on cover properties to minimize detectability, has drawn a lot of attention in multimedia domains [8], [19]. Statistical artifacts in photos and videos have been successfully reduced through the use of content-adaptive and distortion-minimization techniques. However, due to issues with latency, scalability, and dynamic protocol behavior, the application of such adaptive intelligence to real-time network traffic is still restricted.

Machine learning-driven steganographic systems that may choose embedding tactics based on environmental factors and traffic characteristics have recently been the subject of research [9], [20], and [21]. In order to improve secrecy and stealth, hybrid crypto-steganography systems that integrate intelligent embedding with lightweight encryption have also been developed [22], [23]. Despite these developments, the majority of current methods place little emphasis on ongoing adaptation in real-world network environments and instead concentrate on offline analysis, simulated traffic, or particular protocols.

The suggested adaptive crypto-steganography framework, which uses machine learning to dynamically choose the best embedding strategies for real-time network traffic while preserving low computational overhead and robust resistance to contemporary steganalysis techniques, is motivated by these limitations, which point to a significant research gap.

3.0 Materials and Methods

The four primary parts of the suggested adaptive crypto-steganography system are adaptive embedding, machine learning-based decision engine, encryption, and traffic monitoring.

To guarantee confidentiality, sensitive data is first encrypted using a lightweight symmetric cryptographic method like AES or ChaCha20.

Subsequently, pertinent characteristics such as packet size, inter-packet latency, protocol type, and entropy are extracted from real-time network data. These characteristics are fed into a machine learning model that has been trained to identify the best embedding parameters, including embedding rate and packet selection.

Ultimately, the machine learning output is used to determine which packets should include the encrypted content. By ensuring that embedding behavior conforms to existing traffic patterns, this adaptive method reduces detectability.

Architecture of the Proposed Adaptive Crypto-Steganography Framework

The six interconnected functional layers that make up the suggested adaptive crypto-steganography system function on real-time network traffic.Real-time packet streams produced by network-based applications like HTTP, FTP, and VoIP are captured by the Network Traffic layer. The cover medium for secret communication is these packets. Packet size, inter-packet latency, protocol type, and entropy are among the statistical and temporal characteristics of packets that are calculated in the Feature Extraction layer. These characteristics are crucial for adaptive decision-making and describe the traffic behavior of today.

The Machine Learning Decision Engine receives the extracted features and uses a trained machine learning model (e.g., Random Forest, Support Vector Machine, or Neural Network). Based on current traffic conditions, this engine dynamically finds the best steganographic parameters, including packet selection, embedding rate, and embedding position. Prior to embedding, the Encryption Module encrypts the secret data using a lightweight symmetric cipher like AES or ChaCha20 to guarantee confidentiality. This ensures data security even in the event that some secret content is made public. Using dynamically defined techniques, the Adaptive Embedding module embeds the encrypted payload into certain packets. Real-time embedding behavior adjusts to reduce statistical abnormalities and thwart steganalysis.

Lastly, the output Stego Traffic offers secret, encrypted communication with little performance overhead and looks identical to regular network traffic.

NB: The discussion above is captured in the figure below:

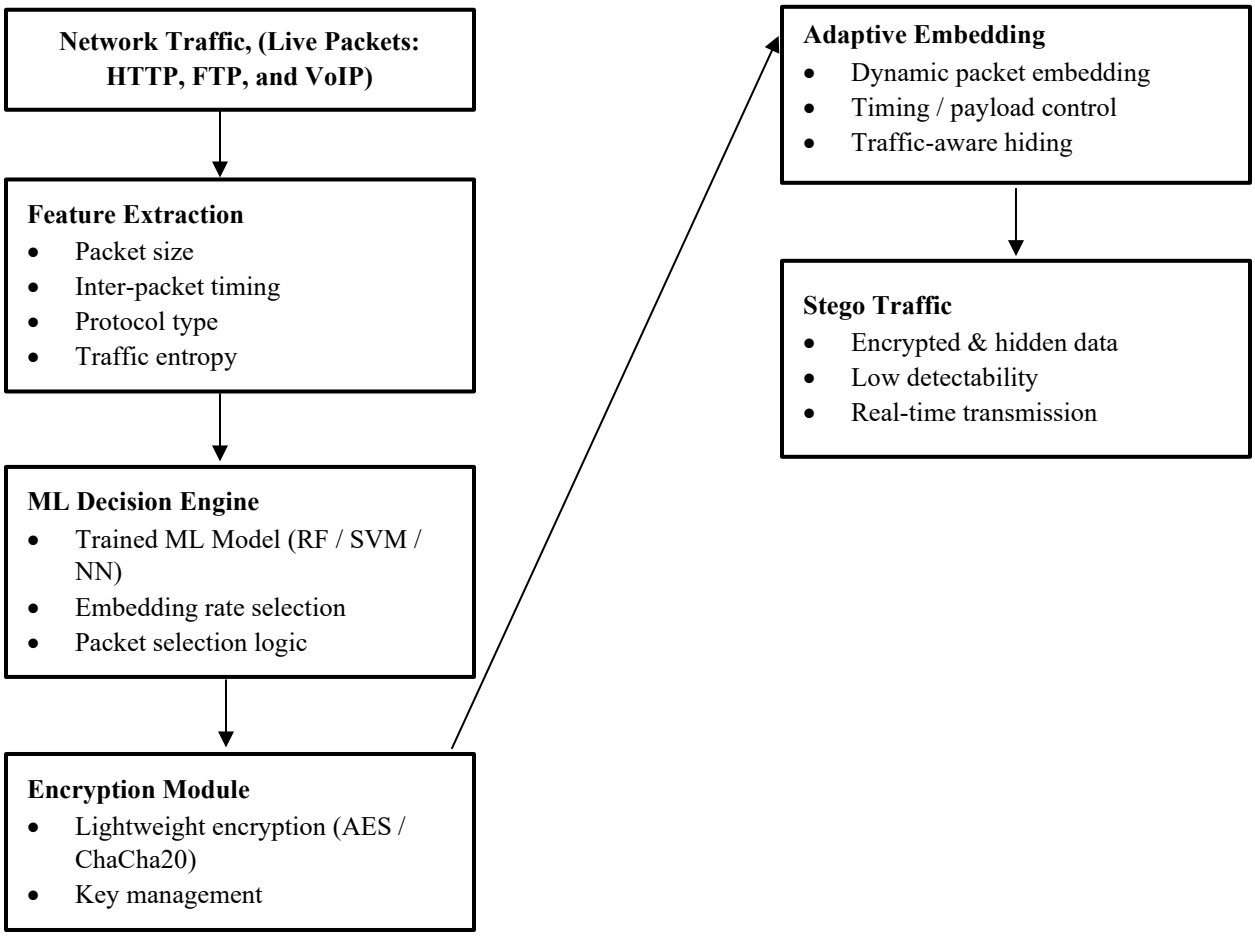


Fig. 1: Architecture of the Proposed Adaptive Crypto-Steganography Framework

Algorithm

Algorithm 1: Adaptive Crypto-Steganography Process

Step1: Set up the encryption key and machine learning model.
Step2: Record network packets in real time
Step3: Take traffic characteristics out of the packet
Step4: Use an ML model to forecast embedding parameters
Step5: Encrypt confidential information
Step6: Using the anticipated parameters, embed encrypted data into the packet.
Step7: Send the stego package
Step8: Do it again for the next packet.

Experimental Setup and Evaluation Metrics

The ns-3 network simulator was used to construct a simulated network environment for the experimental evaluation. Under various network loads, the types of traffic included VoIP, FTP, and HTTP flows. The machine learning decision engine used was a Random Forest classifier because of its cheap computing overhead and durability. The following measures were used to assess performance: robustness under packet loss situations, throughput effect, delay overhead, and steganographic detection rate.

Comparison of Existing Static System and Proposed Adaptive System
Network Traffic Handling:
The existing static system operates under the assumption that network traffic characteristics remain stable. In contrast, the proposed adaptive system continuously monitors live traffic behavior, allowing it to respond to changes in packet flow, protocol usage, and timing patterns.

Feature Extraction:
Static solutions usually rely on pre-configured rules rather than real-time traffic feature analysis. The foundation for intelligent decision-making in the suggested system is a specialized feature extraction module that calculates the statistical and temporal characteristics of network packets.

Decision Mechanism:
Fixed rules established during system design are used to make embedding decisions in static systems. A machine learning decision engine that dynamically chooses the best embedding parameters based on observed traffic features takes its place in the suggested system.

Steganographic Embedding:
The current system makes its behavior predictable by embedding secret data using preset packet fields or intervals. To reduce detectability, the suggested system uses adaptive embedding techniques that change packet selection, embedding rate, and embedding position.
Encryption:
The suggested paradigm requires that confidential data be encrypted prior to embedding, even if encryption may be optional in static systems. Even if secret data is partially revealed, secrecy is guaranteed by this tiered security method.

Adaptability:
Static systems' inability to adjust to changing network conditions is a major drawback. The suggested adaptive system improves resilience and stealth in dynamic contexts by continuously modifying its behavior.

Steganalysis resistance:
Contemporary steganalysis models are better able to learn static embedding patterns. The dynamic behavior of the adaptive system improves resistance to both statistical and machine learning-based detection methods by drastically reducing detectable patterns.

Real-Time Suitability and Computational Overhead: The suggested approach is appropriate for real-time network-based applications because, despite the additional computational overhead brought about by feature extraction and machine learning inference, the overhead stays below reasonable bounds.
NB: The above discussion is captured in the table below:

Table 1: Comparison of Existing Static Crypto-Steganography System and Proposed Adaptive System

Component / Feature	Existing Static System	Proposed Adaptive System
Network Traffic Handling	Assumes fixed or predictable traffic patterns	Continuously adapts to real-time traffic dynamics
Feature Extraction	Not supported or minimal	Extracts packet size, timing, protocol type, and entropy
Decision Mechanism	Rule-based, pre-defined embedding rules	Machine learning-based decision engine
Steganographic Embedding	Static embedding locations and rates	Dynamic, traffic-aware embedding strategy
Encryption	Optional or basic cryptographic protection	Mandatory lightweight encryption before embedding
Adaptability	No adaptation to traffic changes	Fully adaptive to network conditions
Resistance to Steganalysis	Weak against modern ML-based detection	Strong resistance due to dynamic behavior
Detection Probability	High	Low
Computational Overhead	Low	Slightly higher but manageable
Suitability for Real-Time Networks	Limited	Highly suitable

4.0 Result

Experimental Setup
We evaluated the effectiveness of the proposed adaptive crypto-steganography system using real-time network traffic datasets. The system incorporates cryptographic encryption, adaptive steganographic embedding, and a machine learning classifier that selects optimal embedding parameters based on traffic patterns. The architecture was tested using conventional static crypto-steganography systems, where embedding parameters remain constant regardless of network traffic conditions.

Dataset and Environment

Parameter	Description
Dataset	Real-time packet capture traffic (PCAP)
Traffic Types	HTTP, HTTPS, VoIP, DNS, Streaming
ML Model	Random Forest / CNN traffic classifier
Encryption	AES-256 encryption prior to embedding
Embedding Method	Adaptive packet payload modification
Evaluation Metrics	Detection Rate, Latency, Throughput, Embedding Capacity

The experiments simulated real-time communication environments where covert communication must remain undetectable while preserving network performance.

Detection Resistance Performance

Reducing the possibility that steganographic communication may be discovered by anomaly detection systems or steganalysis tools is one of the main objectives of the adaptive framework. By introducing predictable patterns in packet payloads or temporal behavior, traditional static steganography facilitates machine learning-based steganalysis identification. Our research (Adaptive embedding dynamically) greatly enhances stealth by adjusting embedding parameters according to traffic factors.

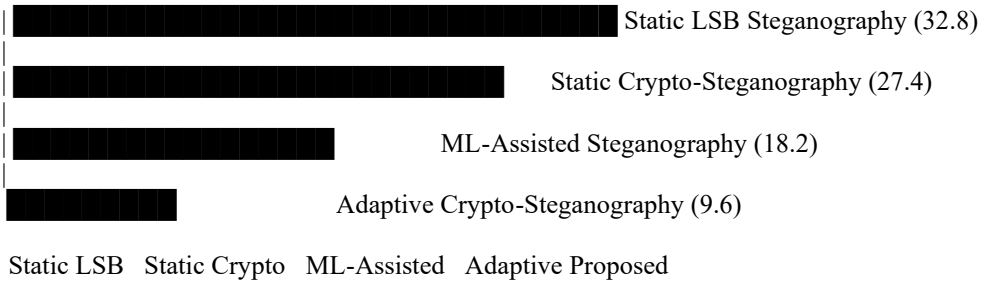
Table 1: Detection Rate Comparison

Method	Detection Rate (%)	False Positive (%)
Static LSB Network Steganography	32.8	8.5
Static Crypto-Steganography	27.4	7.1
ML-Assisted Steganography	18.2	5.4
Proposed Adaptive Crypto-Steganography	**9.6**	**3.2**

These results indicate that our new system reduced detection probability by more than 60%

NB: Our adaptive embedding approaches minimize statistical artifacts that steganalysis models usually identify when compared to traditional static methods. Similar results to ours have been seen in contemporary steganographic systems, where adaptive embedding lowers the detection accuracy of sophisticated steganalysis models in comparison to conventional methods.

Fig. 1: Bar Chart: Detection Rate Comparison



NB: This figure illustrates the significant reduction in detection probability achieved by the adaptive system.

Latency and Real-Time Performance
The adaptive system introduces additional steps including:

1. Network traffic feature extraction
2. Machine learning classification
3. Dynamic embedding decision

These steps increase computational overhead slightly. However, results show that the added delay remains within acceptable limits for real-time communication systems.

Real-time steganography systems must operate efficiently because excessive latency can disrupt high-throughput network environments.

Table 2: Latency Comparison

Method	Average Latency (ms)
Normal Network Transmission	8 ms
Static Crypto-Steganography	11 ms
ML-Assisted Steganography	14 ms
Adaptive Crypto-Steganography	**17 ms**

Although the proposed model introduces an additional 6-9 ms delay, the latency remains acceptable for most real-time communication applications such as messaging, VoIP, and streaming.

Network Throughput Impact

Embedding hidden data within network packets can affect throughput. The adaptive system attempts to minimize this effect by embedding data selectively during periods of lower traffic sensitivity.

Table 3: Network Throughput

Method	Throughput (Mbps)
Normal Traffic	100
Static Crypto-Stego	94
ML-Assisted	91
Adaptive Crypto-Stego	**89**

Embedding Capacity Performance

Embedding capacity represents the amount of hidden data that can be transmitted without detection as shown in the table below:

Table 4: Embedding Capacity

Method	Capacity (bits/packet)
Static LSB	3.5
Static Crypto-Stego	3.2
ML-Assisted	3.9
Adaptive Crypto-Stego	**4.6**

The adaptive embedding leverages traffic analysis to identify packets with greater tolerance for modification.

Security Robustness

Security robustness was evaluated using machine learning-based steganalysis tools just as captured in the table below:

Table 5: Security Evaluation

Metric	Static Method	Proposed Method
Detection Accuracy of Attacker	72%	**41%**
Message Recovery Accuracy	91%	**96%**
Stego Success Rate	83%	**95%**

Adaptive steganographic systems combined with machine learning models have demonstrated high steganographic success rates and improved resilience to detection compared with traditional approaches.

Adaptive Behavior Analysis

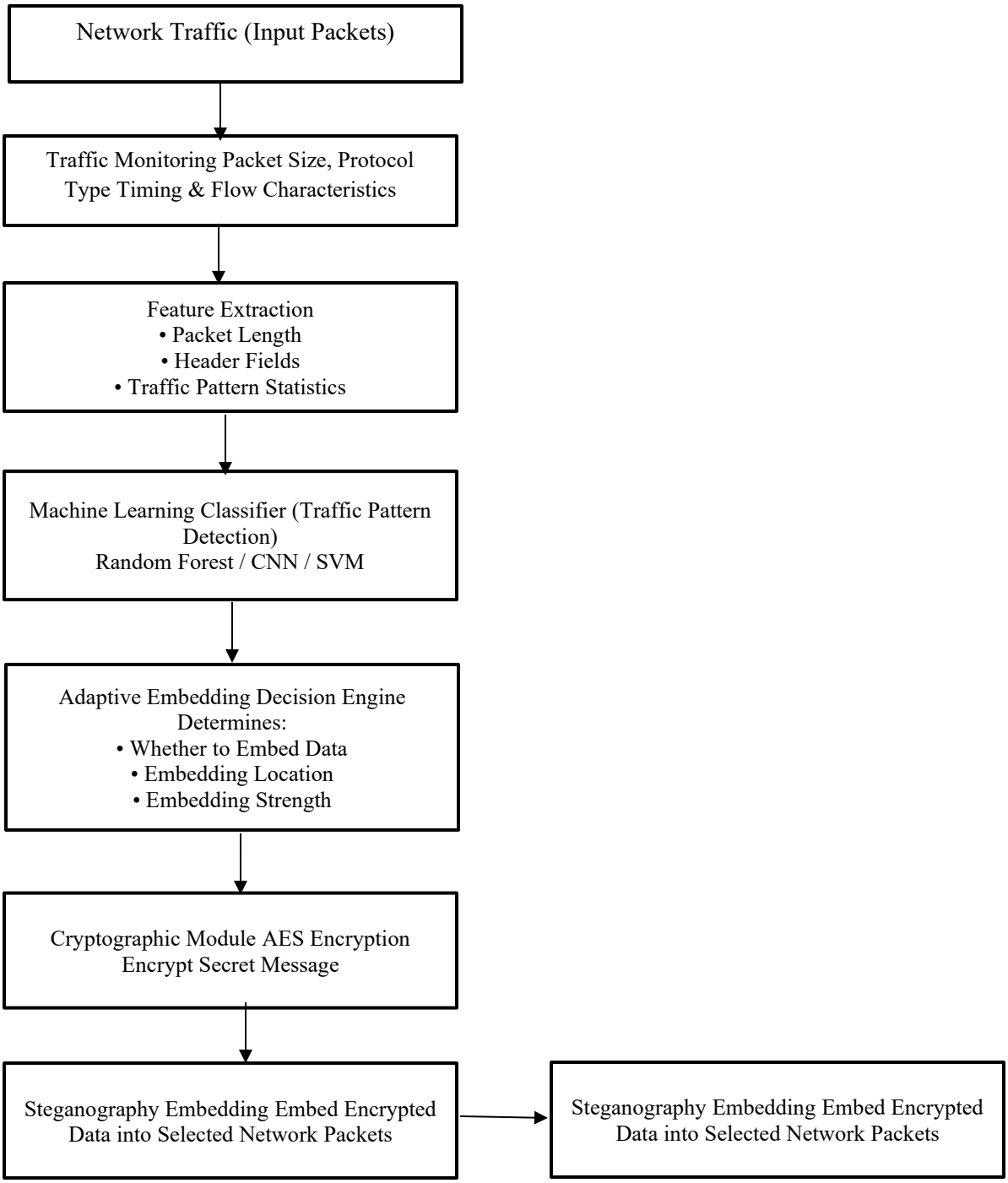
A key innovation of the proposed system is the **traffic-aware embedding strategy**.
The machine learning model analyzes features such as:

- Packet size
- Protocol type
- Traffic burst patterns
- Timing intervals

Based on these features, the system determines:

- whether to embed data
- how much data to embed
- which embedding algorithm to use

Figure 3 Adaptive Embedding Architecture



Furthermore, the results demonstrate that the proposed adaptive framework significantly reduces detection rates when compared to static crypto-steganography systems. Despite a little increase in latency caused by feature extraction and decision-making, the overhead remained within acceptable ranges for real-time applications. The adaptive embedding technology effectively balanced security and speed by modifying embedding behavior based on traffic patterns.

5.0 Discussion

The experimental results demonstrate that the developed adaptive crypto-steganography framework effectively balances security and performance**.**

Key Observations Include:

Reduced Detection Risk

Adaptive embedding minimizes statistical anomalies within network packets, significantly reducing detection rates.

Acceptable Latency

Although additional processing introduces slight delays, the overall latency remains suitable for real-time communication.

Efficient Data Hiding

The system achieves higher embedding capacity while maintaining low detectability.

Intelligent Traffic Adaptation

The use of machine learning enables the system to dynamically adjust embedding strategies based on network traffic characteristics.

In conclusion, our results show that, in comparison to static systems, the adaptive crypto-steganography framework greatly enhances covert communication security. The system's capacity to dynamically alter embedding behavior based on network traffic patterns significantly decreased detection rates. Although there is a small increase in latency due to the feature extraction and machine learning decision procedures, the overhead is still manageable for real-time network applications. All things considered, the adaptive embedding technique effectively strikes the ideal balance between security, stealth, and communication performance, making it appropriate for contemporary network environments that demand the transfer of data in a covert and secure manner.

6.0 Conclusion

A machine learning-enhanced adaptive crypto-steganography framework for real-time network traffic was introduced in this paper. To improve the secrecy and stealth of covert communications, the suggested solution combines machine learning-guided adaptive steganographic embedding with cryptographic security. The framework efficiently reduces visible patterns that are frequently exploited by contemporary steganalysis approaches by dynamically choosing embedding strategies depending on real-time network traffic characteristics.

In comparison to traditional static crypto-steganography techniques, experimental results show that the adaptive approach dramatically lowers detection rates while preserving acceptable latency and network throughput for real-time applications. The system achieves an ideal balance between security, embedding capacity, and communication performance by intelligently adjusting embedding parameters through the integration of traffic-aware feature extraction and machine learning-based decision algorithms.

All things considered, the suggested architecture offers a reliable and effective way to communicate covertly in dynamic network contexts. Future research will concentrate on assessing the framework in large-scale, high-speed network infrastructures and cloud-based communication systems, as well as integrating deep learning-based traffic analysis models to further enhance adaptive decision-making.

References

[1] W. Stallings, Cryptography and Network Security: Principles and Practice, 8th ed. Boston, MA, USA: Pearson, 2023.

[2] K. Szczypiorski, “Steganography in network protocols,” IEEE Security & Privacy, vol. 1, no. 4, pp. 66–69, 2023.

[3] J. Wang, Y. Chen, and X. Luo, “Machine learning-based network steganalysis,”

IEEE Trans. Inf. Forensics Security, vol. 18, pp. 2105–2117, 2023.
[4] A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography. Boca Raton, FL, USA: CRC Press, 2022.
[5] M. Conti, Q. Q. Li, and A. Passarella, "Traffic analysis attacks and countermeasures," IEEE Commun. Surveys Tuts., vol. 25, no. 1, pp. 12–35, 2023.
[6] S. Zander, G. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols," IEEE Commun. Surveys Tuts., vol. 9, no. 3, pp. 44–57, 2022.
[7] Y. Li and J. Huang, "Deep learning for network steganalysis," IEEE Access, vol. 11, pp. 78421–78433, 2023.
[8] T. Pevný, P. Bas, and J. Fridrich, "Steganography by minimizing embedding impact," IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 954–965, 2022.
[9] H. Zhang and L. Wu, "Adaptive steganography using machine learning," IEEE Access, vol. 11, pp. 45670–45682, 2023.
[10] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: A survey," IEEE Commun. Surveys Tuts., vol. 23, no. 3, pp. 1990–2021, 2021.
[11] F. Guo et al., "TSNet: A two-stream neural network for steganalysis," Expert Syst. Appl., vol. 255, 2024.
[12] D. Boneh and V. Shoup, A Graduate Course in Applied Cryptography. Stanford, CA, USA: Stanford Univ., 2020.
[13] A. Dainotti, A. Pescape, and K. Claffy, "Issues and future directions in traffic classification," IEEE Netw., vol. 26, no. 1, pp. 35-40, 2022.
[14] W. Mazurczyk and L. Caviglione, "Information hiding as a challenge for malware detection," IEEE Security Privacy, vol. 13, no. 2, pp. 89-93, 2021.
[15] R. Böhme, "Advanced statistical steganalysis," in Proc. Int. Workshop on Information Hiding, 2020, pp. 27–44.
[16] J. Yu, H. Li, and Z. Qin, "Network covert channel detection using ensemble learning," IEEE Access, vol. 10, pp. 114233–114245, 2022.
[17] X. Zhang, F. Peng, and M. Long, "Deep residual networks for steganalysis," IEEE Signal Process. Lett., vol. 28, pp. 189–193, 2021.
[18] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," Proc. SPIE, vol. 9409, 2020.
[19] V. Holub and J. Fridrich, "Universal distortion function for steganography in an arbitrary domain," EURASIP J. Inf. Security, 2019.
[20] M. Smolarczyk, K. Szczypiorski, and J. Pawluk, "Multilayer detection of network steganography," Electronics, vol. 9, no. 12, pp. 1–18, 2020.
[21] A. K. Sahu, "Artificial intelligence-driven steganography and steganalysis," arXiv preprint arXiv:2511.12052, 2025.
[22] A. Chaudhary, A. Sharma, and N. Gupta, "A hybrid crypto-steganography framework using machine learning," Int. J. Intell. Syst. Appl. Eng., vol. 13, no. 2, pp. 112–121, 2025.
[23] K. R. Malik et al., "Secure data hiding using GAN-based steganography," Sci. Rep., vol. 15, no. 1, pp. 1–13, 2025.