XAI-Driven Encrypted Traffic Detection and Characterization to Enhance Information Security

Adit Sharma

A Thesis Submitted to the Faculty of Graduate Studies
In Partial Fulfillment of the Requirements
for the Master of Arts in
Information Systems & Technology Degree

Graduate Program in Information Systems and Technology
Faculty of Liberal Arts & Professional Studies

York University
Toronto, Ontario

March 2025

Abstract

Securing information through encryption is essential in data communication, but to effectively detect malicious activities, it is crucial to distinguish between encrypted and non-encrypted traffic. Traditional encrypted traffic classification methods, including rule-based systems and conventional machine learning approaches, often struggle with scalability, generalization, and class imbalance, leading to suboptimal classification performance. This study introduces a novel hybrid model for encrypted traffic classification, integrating Multi-Head Attention mechanisms for feature enhancement and LightGBM as the final classifier. The proposed model follows a two-step classification process: first, performing binary classification to separate encrypted and non-encrypted traffic, and second, applying multi-class classification to categorize encrypted traffic into TOR, VPN, I2P, Zeronet, and Freenet. To improve model interpretability, SHAP is employed to validate the importance of attention-based features, while LIME provides insights into misclassified instances, enabling adjustments such as weight threshold tuning and handling class imbalances.

Furthermore, this study incorporates a refined dataset preprocessing pipeline, leveraging NTL Flowlyzer—an advanced traffic analyzer that extracts over 400 features, including entropy-based attributes. To address class imbalance issues, strategic adjustments such as SMOTE augmentation for Freenet and class-specific threshold tuning were applied based on SHAP and LIME insights, resulting in improved classification performance. The experimental evaluation demonstrates that the proposed hybrid model outperforms existing approaches in accuracy, precision, and recall while maintaining efficiency in both time and computational complexity. By integrating explainable AI techniques and adaptive optimization strategies, our approach enhances classification performance and improves the transparency and interpretability of encrypted traffic detection. These findings contribute to advancing cybersecurity by enabling more robust and interpretable encrypted traffic classification models.

Acknowledgement

I would like to express my heartfelt gratitude to my family and friends for their unwavering support and encouragement throughout my Master's journey. I am also deeply grateful to my supervisor, Prof. Arash Habibi Lashkari, whose guidance and steadfast support have been invaluable to my academic and research endeavors.

Contents

List of Tables

List of Figures

1 Introduction

The rapid proliferation of encrypted traffic has reshaped network security landscape, fundamentally altering how data is transmitted and analyzed across digital infrastructures. Encryption has become a cornerstone of modern cybersecurity, driven by heightened privacy concerns, evolving regulatory frameworks, and the increasing sophistication of cyber threats [68]. As of 2023, an estimated 95% of global web traffic is encrypted using HTTPS protocols, reflecting a significant shift toward prioritizing confidentiality in digital communications [69]. However, this widespread adoption of encryption has introduced critical challenges for network management, security monitoring, and cyber threat detection [70].

While encryption enhances privacy, it also enables cybercriminals to obfuscate malicious activities. A recent report by Zscaler ThreatLabz indicates that 85.9% of cyberattacks now leverage encrypted channels, representing a 20% increase from the previous year [71]. The widespread deployment of TLS 1.3, while improving security, further complicates threat visibility by removing legacy interception techniques [72]. These trends necessitate the development of sophisticated encrypted traffic analysis (ETA) techniques capable of detecting threats without compromising encryption integrity.

Machine learning (ML) and deep learning (DL) have emerged as promising solutions for encrypted traffic detection and classification [73, 74]. By leveraging statistical patterns and behavioral anomalies, ML-based approaches offer a viable alternative to traditional deep packet inspection, enabling privacy-preserving security analytics. Supervised and unsupervised learning techniques have been extensively explored, with convolutional and recurrent neural networks (CNNs, RNNs) demonstrating success in traffic pattern recognition [75, 76]. However, interpretability and computational efficiency remain key challenges, limiting the widespread adoption of deep learning in real-world security infrastructures.

A growing body of research has underscored the importance of balancing security and privacy in encrypted traffic classification. While security mechanisms aim to prevent unauthorized access and mitigate threats, privacy-preserving techniques seek to maintain confidentiality in data transmission [77, 78]. Emerging solutions, such as secure multiparty computation (SMC) and differential privacy, offer new avenues for ensuring privacy while enabling robust encrypted traffic analysis [79]. Additionally, the convergence of blockchain technology with encryption mechanisms has strengthened the resilience of IoT networks, ensuring secure and decentralized communication [80, 81, 82].

Within the domain of anonymity-preserving networks, encrypted traffic classification faces additional challenges. Research into Tor, I2P, and ZeroNet has highlighted the complexity of analyzing obfuscated traffic patterns without decrypting the underlying content [83, 84]. Traffic analysis techniques for anonymity networks must carefully balance detection accuracy with user privacy, a challenge compounded by the evolving nature of obfuscation strategies [85, 86]. These considerations underscore the need for adaptive and interpretable classification models to analyze encrypted traffic while preserving privacy effectively.

This thesis aims to address these challenges by introducing several key contributions to the field of encrypted traffic classification:

- A **Comprehensive Literature Survey** on encrypted traffic analysis, examining state-of-the-art techniques, datasets, and tools.
- The Development of an **Open-source Cybersecurity Network Traffic Analyzer**, an enhancement of

NTLFlowlyzer, incorporating additional features for enriched traffic characterization.

- The creation of an **Augmented Dataset** that combines CIC-Darknet2020 and Darknet-Dataset-2020 to improve the diversity and representation of encrypted traffic patterns.
- The design and implementation of an **Explainable AI-powered Classification Model**, integrating a Hybrid Attention + LightGBM architecture for improved interpretability and performance to evaluate the efficacy of model refinement techniques guided by XAI tools such as SHAP and LIME.

The integration of Explainable AI (XAI) techniques into encrypted traffic classification plays a crucial role in bridging the gap between model performance and operational transparency. In cybersecurity applications, understanding why a model flagged a particular flow as suspicious is as important as the detection itself. By employing SHAP and LIME, this thesis not only enhances model interpretability but also enables actionable insights for cybersecurity professionals, fostering trust, auditability, and improved decision-making in encrypted environments.

The remainder of this thesis is structured as follows. **Section 2 (Literature Review)** presents a comprehensive analysis of encryption techniques, traffic classification models, and recent advancements in encrypted traffic detection. **Section 3 (Proposed Model)** introduces the proposed methodology, detailing the design and implementation of the network traffic analyzer, dataset augmentation process, and classification model. **Section 4 (Selecting and Creating the Training and Testing Dataset)** focuses on dataset selection, preprocessing techniques, and feature extraction workflows, emphasizing the importance of structured representation in encrypted traffic classification. **Section 5 (Experiments & Results)** details the experimental setup and presents an in-depth evaluation of the proposed model, including comparisons with existing classification techniques. **Section 6 (Analysis & Discussion)** provides a critical analysis and discussion of results, highlighting key findings and their implications for real-world encrypted traffic detection. Finally, **Section 7 (Conclusion & Future Work)** concludes the study by summarizing contributions and outlining future research directions.

Table 35 in the Appendix lists the acronyms and abbreviations used in this thesis, providing a reference for technical terminology.

2 Literature Review

The increasing complexity and prevalence of encrypted network traffic have necessitated advanced analytical approaches for effective classification and detection. This section presents a comprehensive review of encryption techniques, detection methodologies, and classification strategies as detailed in our taxonomy figure 1. By systematically categorizing existing research, we provide insights into the fundamental techniques shaping this field. We first discuss encryption traffic techniques and services, followed by an extensive review of machine learning, deep learning, and hybrid models employed for encrypted traffic analysis studies. Each approach is critically analyzed, assessing its strengths, limitations, and contributions to the broader network security landscape. Finally, we synthesize the limitations of the literature to highlight open challenges and future research directions.

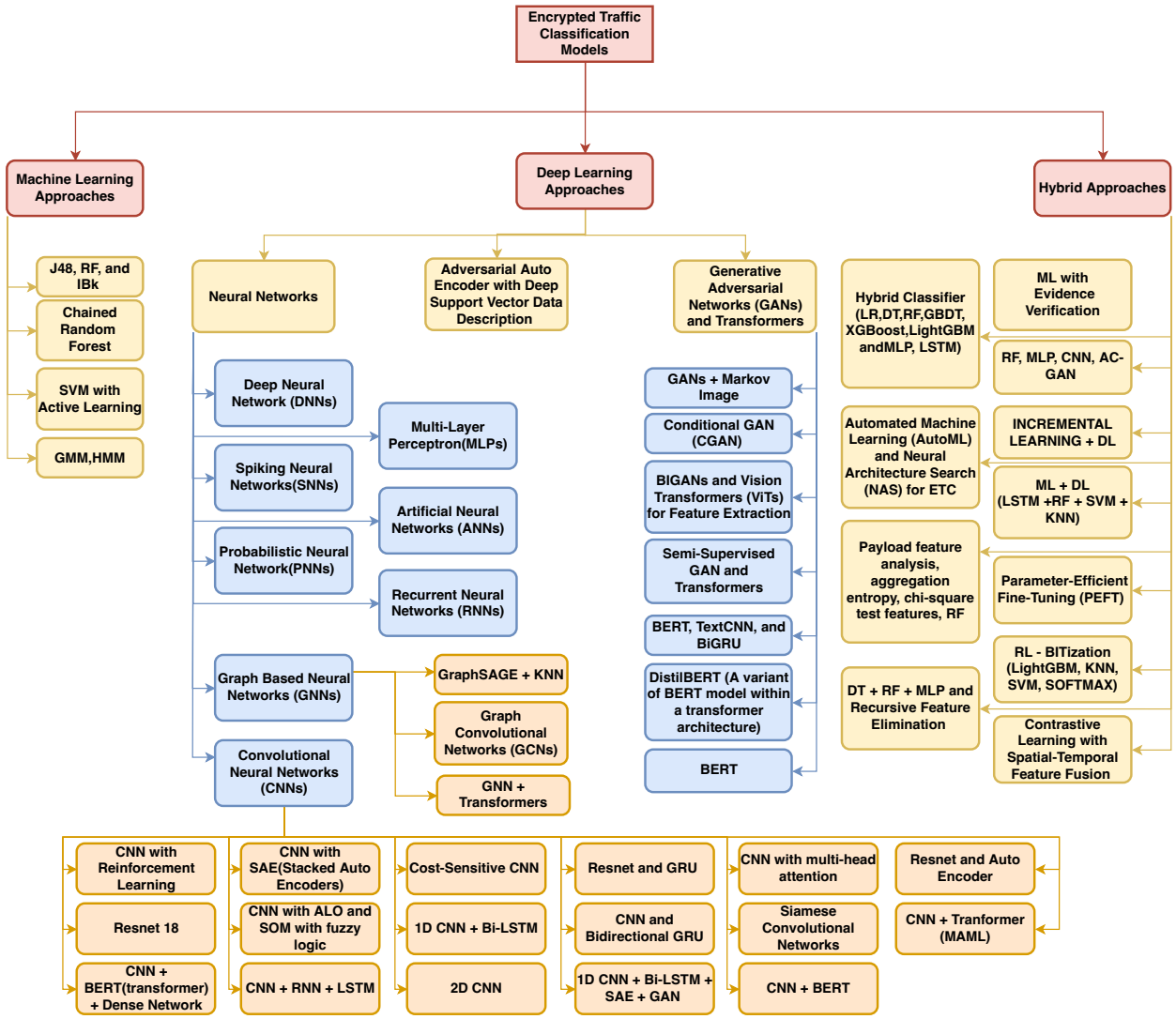

Figure 1: Taxonomy of Encrypted Traffic Detection Models Derived from Previous Technical Literature

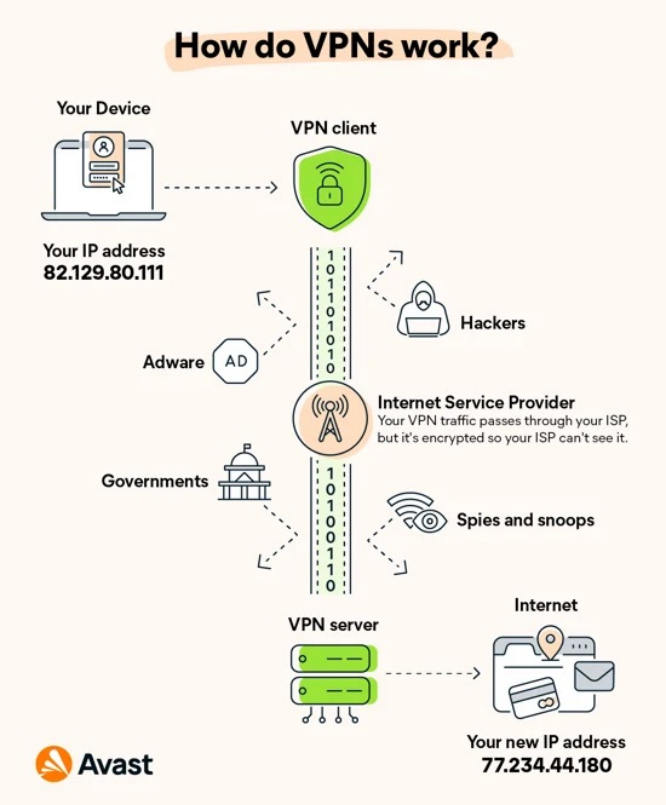

Figure 2: Overview of Virtual Private Network (VPN) Functionality[1]

2.1 Encryption Traffic Techniques/Services Protocols

Encryption techniques and services form the backbone of modern cybersecurity frameworks, ensuring the confidentiality and integrity of data transmitted across networks. This section provides an in-depth examination of various encryption protocols, ranging from traditional methods and protocols to advanced privacy-preserving services such as VPNs, Tor, and other anonymization technologies. By analyzing their operational principles, advantages, and limitations, we aim to establish a foundational understanding of how these encryption mechanisms influence traffic analysis and detection methodologies.

- **VPN (Virtual Private Network) (1996, Gurdeep Singh-Pall)** - VPNs create a private network from a public internet connection, encapsulating and encrypting data packets to provide secure and anonymous access to the internet. This technology is fundamental in bypassing geo-restrictions and safeguarding sensitive data from unauthorized access, especially beneficial in environments that require confidentiality like corporate networks[87].Fig.2 illustrates the working mechanism of a VPN, showcasing how it encrypts data to protect users from hackers, adware, and surveillance while masking the original IP address.

- **TOR (The Onion Router) (2002, Paul Syverson, Michael G. Reed, David Goldschlag - U.S. Naval Research Laboratory)**- TOR is a network of servers that enables anonymous communication by directing internet traffic through a free, worldwide, volunteer overlay network consisting of more than seven thousand relays. The design ensures that the user's location and usage from anyone conducting network surveillance or traffic analysis is hidden, making it a crucial tool for preserving privacy and freedom online [88].Fig.3 illustrates the Tor connection process, highlighting the layered routing through entry nodes, relays, and exit

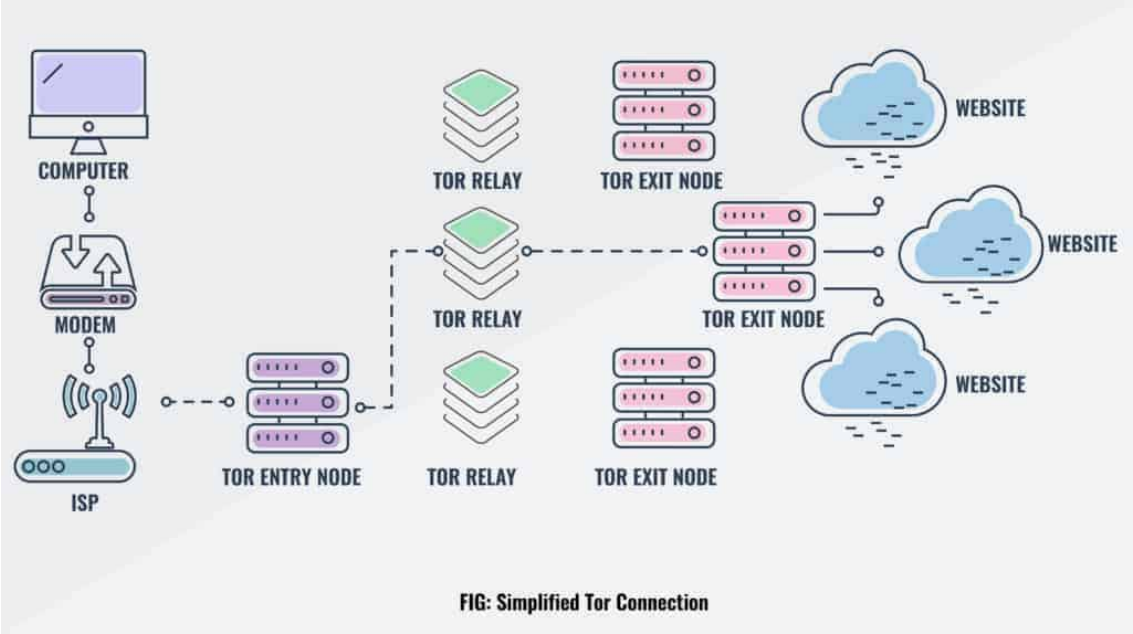

Figure 3: Representation of Onion Routing in Tor Network[2]

nodes to ensure anonymity.

- **I2P (Invisible Internet Project) (2003, I2P Team and Community)** - I2P specializes in allowing anonymous communication over the internet. It uses a peer-to-peer-like routing structure to create a private network layer that enables secure and anonymous communication. Unique to I2P is its use of garlic routing, which encapsulates multiple messages together to enhance security and privacy beyond traditional onion routing [89].Fig.4 depicts the I2P network architecture, demonstrating the use of inbound and outbound tunnels to facilitate secure and anonymous peer-to-peer communication.

- **Zeronet (2015, Tamas Kocsis)** - Zeronet[90], introduced in 2015 by Tamas Kocsis, employs Bitcoin cryptography and BitTorrent technology to establish a decentralized web. This framework ensures open, free, and uncensorable websites by leveraging peer-to-peer networking, where content is hosted across multiple nodes, making takedown efforts significantly challenging. The architecture of Zeronet, illustrated in Figure 5, showcases its decentralized nature, highlighting its reliance on cryptographic security and distributed hosting mechanisms. This structure has led to increasing research interest in analyzing its ecosystem, particularly in understanding its role in the modern dark web [4].

- **Freenet (2000, Ian Clarke)** - Freenet, introduced in 2000 by Ian Clarke, is a peer-to-peer platform designed for censorship-resistant communication. It utilizes a decentralized distributed data store to securely keep and deliver information while supporting various services such as websites, forums, and file-sharing. The platform emphasizes privacy and anonymity by routing data through multiple nodes, ensuring that user activity remains untraceable [91]. Figure 6 illustrates the fundamental concept of peer-to-peer communication within Freenet, showcasing how users share data without relying on central servers.

- **HTTPS (Hypertext Transfer Protocol Secure) (1995, Netscape)** - HTTPS was introduced by Netscape in 1995 and ensures secure communication over a computer network, with widespread use on the Internet. It

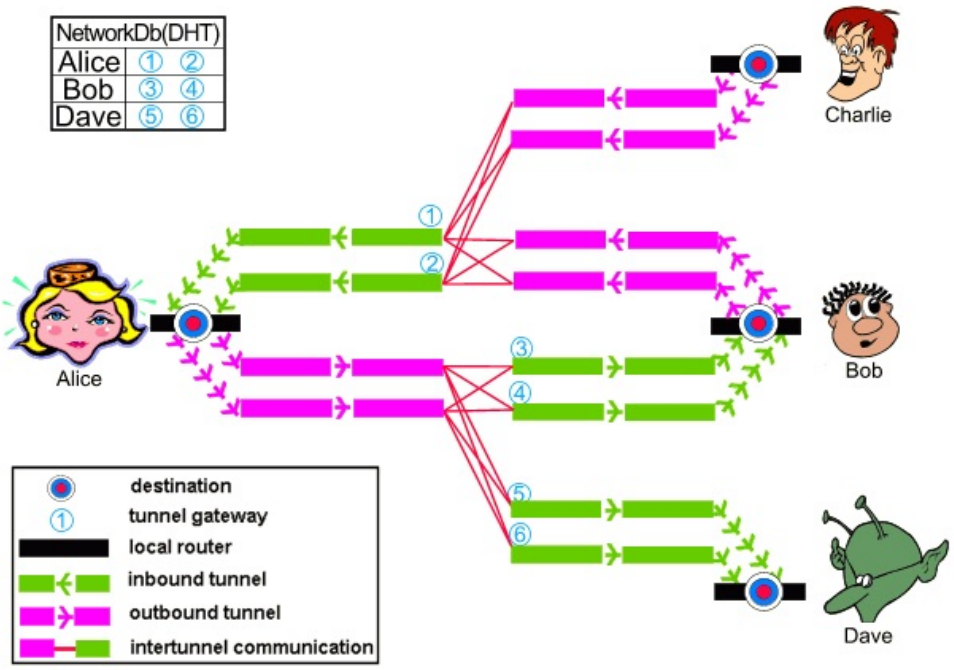

Figure 4: Illustration of Anonymous Routing in the I2P Network[3]

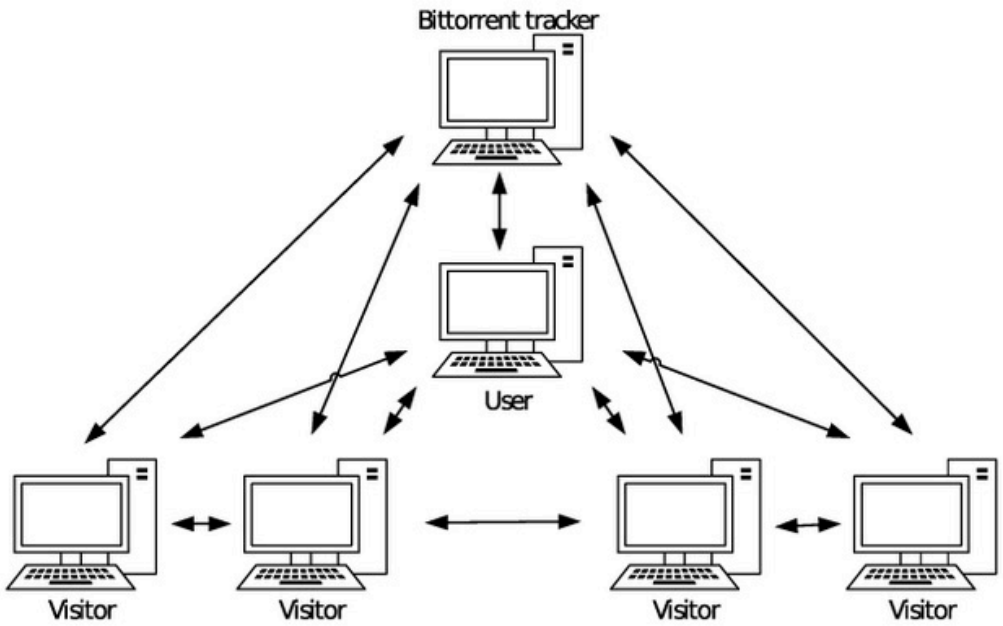

Figure 5: Illustration of Zeronet's decentralized architecture and its cryptographic security mechanisms [4].

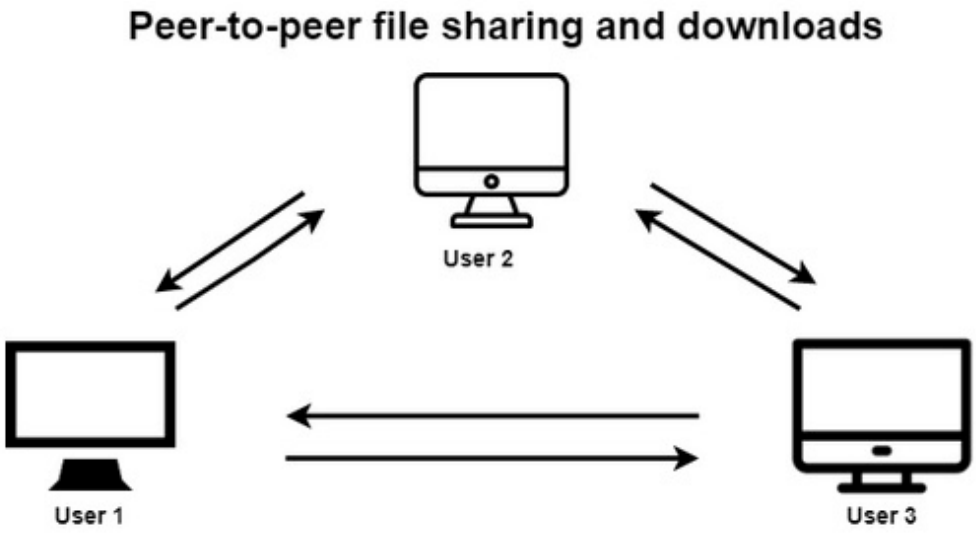

Figure 6: Illustration of peer-to-peer file sharing in Freenet, demonstrating the decentralized communication model [5].

employs encryption mechanisms to protect the integrity and confidentiality of data exchanged between a user's device and a web server, making it a cornerstone of secure online transactions and data privacy [92]. Figure 7 illustrates the HTTPS communication framework, showcasing its role in securing data transfer through encryption layers, proxies, and application servers.

- **TLS (Transport Layer Security) (1999, IETF) -**Transport Layer Security (TLS), introduced by the IETF in 1999, is a cryptographic protocol designed to provide secure communication over a computer network. It is the successor to SSL and is crucial in safeguarding internet traffic from eavesdroppers and tampering. Encrypting data transmission and ensuring authentication, TLS strengthens privacy and integrity in network communications [93]. Figure 8 illustrates the TLS architecture, detailing its structure and functional components, including protocol operations and security mechanisms.

- **SSL (Secure Sockets Layer) (1995, Netscape) -** Secure Sockets Layer (SSL), developed by Netscape in 1995, establishes an encrypted link between a web server and a browser, ensuring data confidentiality and integrity. This encryption mechanism protects sensitive online transactions, such as banking and e-commerce. SSL uses digital certificates to authenticate the identities of communicating parties and employs a combination of public-key and symmetric-key encryption to secure data transmission [94]. Figure 9 illustrates the layered architecture of SSL, showcasing its record protocol, encryption mechanisms, and interaction with underlying network protocols.

- **IPSec (Internet Protocol Security) (1995, IETF) -** Internet Protocol Security (IPSec), standardized by the IETF in 1995, is a framework designed to secure internet communications across an IP network by encrypting and authenticating each IP packet in a communication session. IPSec is fundamental in establishing secure Virtual Private Networks (VPNs), ensuring confidentiality, integrity, and authentication of transmitted data. It

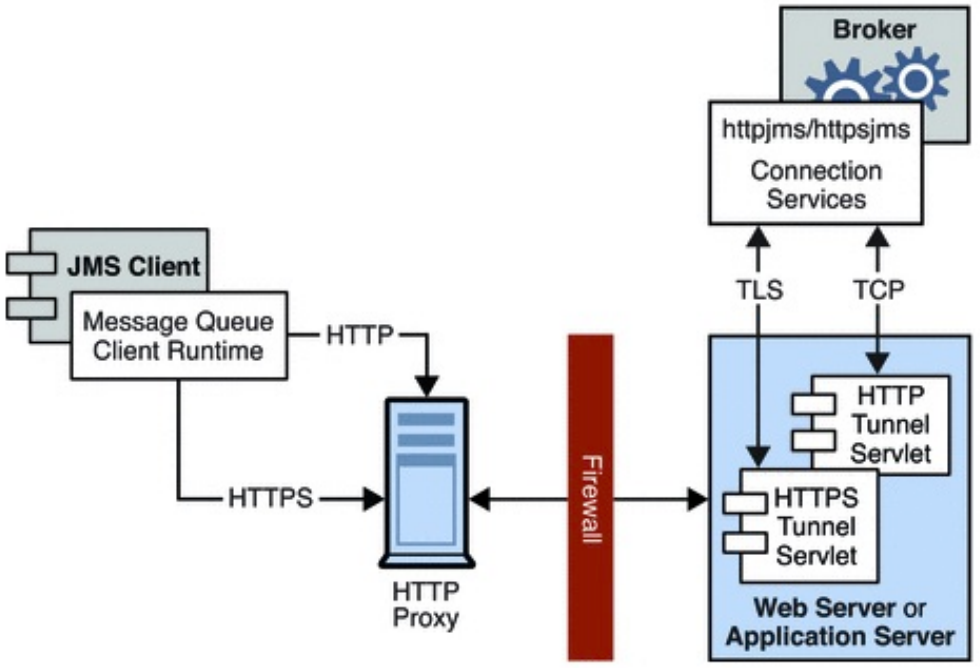

Figure 7: Illustration of HTTPS communication architecture, highlighting encryption mechanisms and proxy interactions [6].

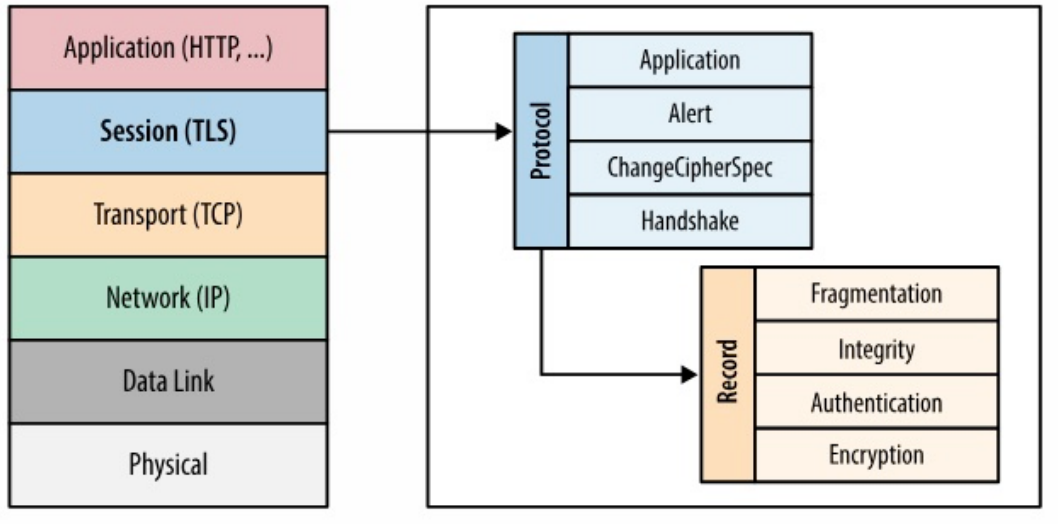

Figure 8: Illustration of the TLS architecture, showcasing its layered security structure and cryptographic protocol components [7].

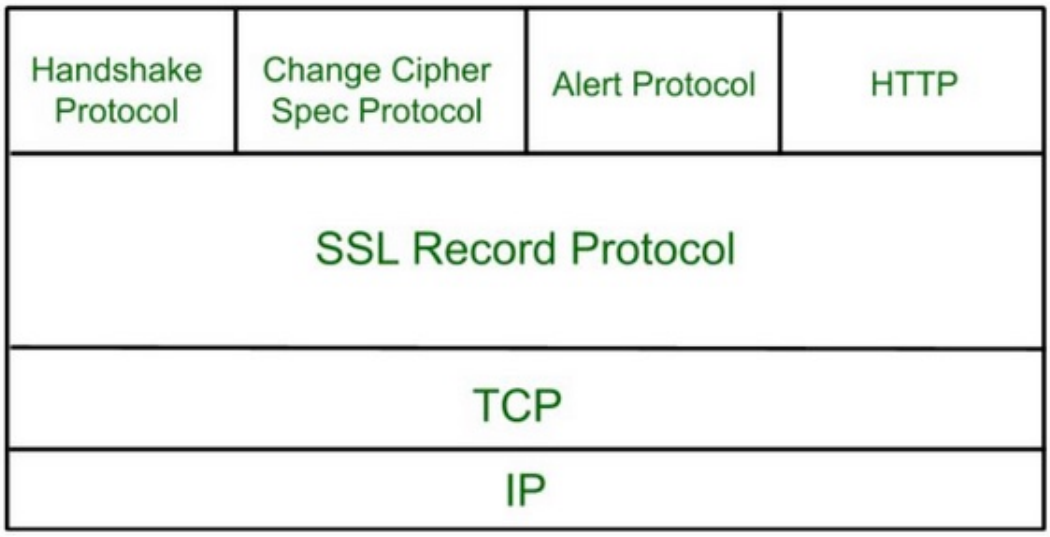

Figure 9: Illustration of SSL architecture, depicting its encryption layers and protocol structure [8].

employs cryptographic techniques, including the Encapsulating Security Payload (ESP) and Authentication Header (AH) protocols, to provide robust security mechanisms [95]. Figure 10 illustrates the architectural components of IPSec, highlighting its encryption, authentication, and key management processes.

- **SSH (Secure Shell) (1995, Tatu Ylönen) -** Secure Shell (SSH) is a cryptographic protocol designed for securely operating network services over an unsecured network. It is widely used for remote login due to its superior security to older protocols like Telnet. SSH encrypts all transmitted data, including authentication credentials, and prevents eavesdropping, connection hijacking, and various network-level attacks [96]. Figure 11 illustrates the architecture of SSH, detailing its encryption process, padding mechanisms, and message authentication system, which collectively enhance security against unauthorized access and tampering.

- **PGP (Pretty Good Privacy) (1991, Phil Zimmermann) -** Pretty Good Privacy (PGP), developed by Phil Zimmermann in 1991, is a data encryption and decryption program designed to provide cryptographic privacy and authentication for secure communication. It is widely used to encrypt emails, secure files, and even protect entire disk partitions. PGP combines symmetric-key cryptography for fast encryption with public-key cryptography for secure key exchange, typically utilizing the RSA algorithm. Unlike centralized trust models, PGP employs a web of trust, allowing users to sign each other's keys to verify identities [97]. Figure 12 illustrates the PGP encryption and decryption workflow, highlighting the role of private and public keys in ensuring data integrity and authentication.

- **S/MIME (Secure/Multipurpose Internet Mail Extensions) (1995, RSA Data Security, Inc.) -** Secure/Multipurpose Internet Mail Extensions (S/MIME), developed by RSA Data Security, Inc. in 1995, is a widely adopted standard for securing email communication through encryption and digital signatures. S/MIME ensures confidentiality by encrypting email contents and guarantees authenticity and integrity through digital signatures. It employs a hierarchical Public Key Infrastructure (PKI), where Certificate Authorities (CAs) authenticate users and devices, making it particularly suitable for enterprise environments where centralized management of certificates is necessary [98]. Figure 13 illustrates the S/MIME email security process, demonstrating encryption, digital signatures, and certificate validation.

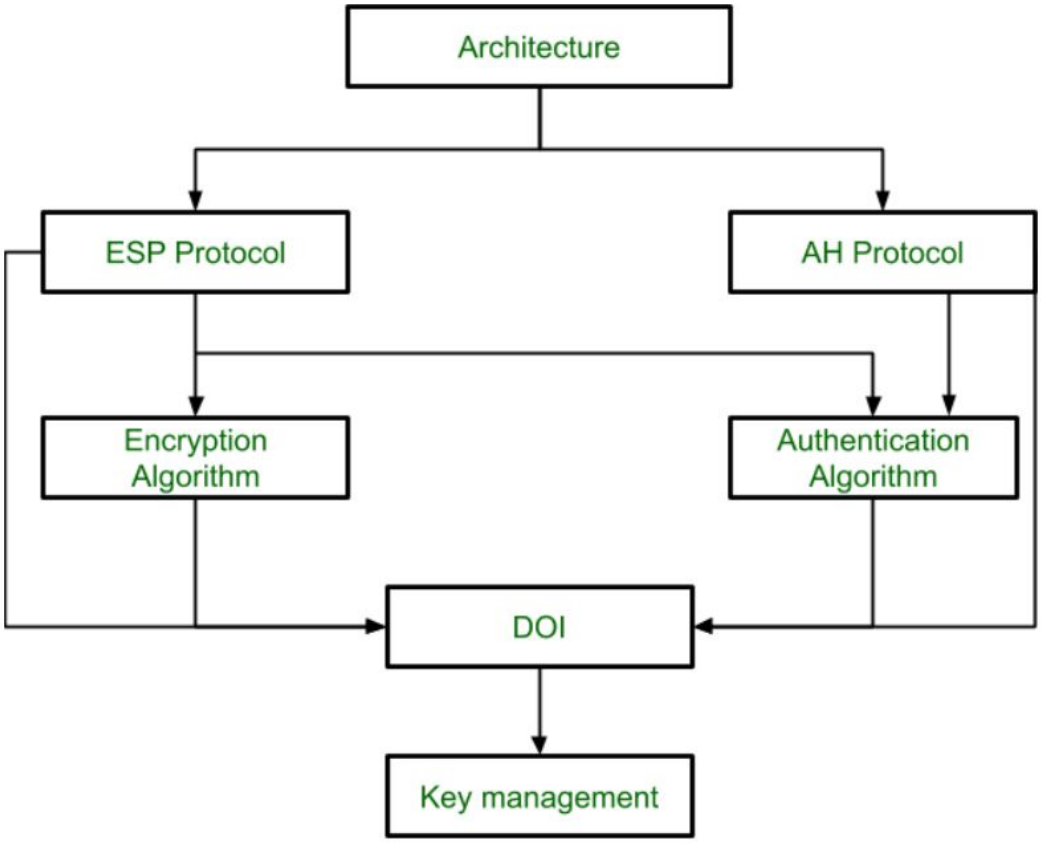

Figure 10: Illustration of IPSec architecture, demonstrating encryption, authentication, and key management mechanisms [9].

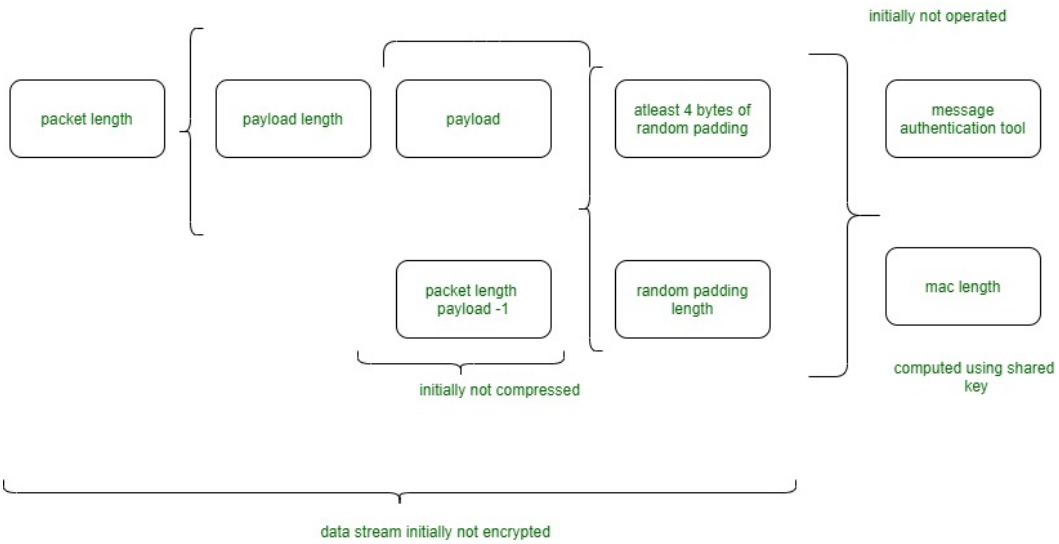

Figure 11: Illustration of SSH architecture, showcasing its encryption, padding mechanisms, and authentication structure [10].

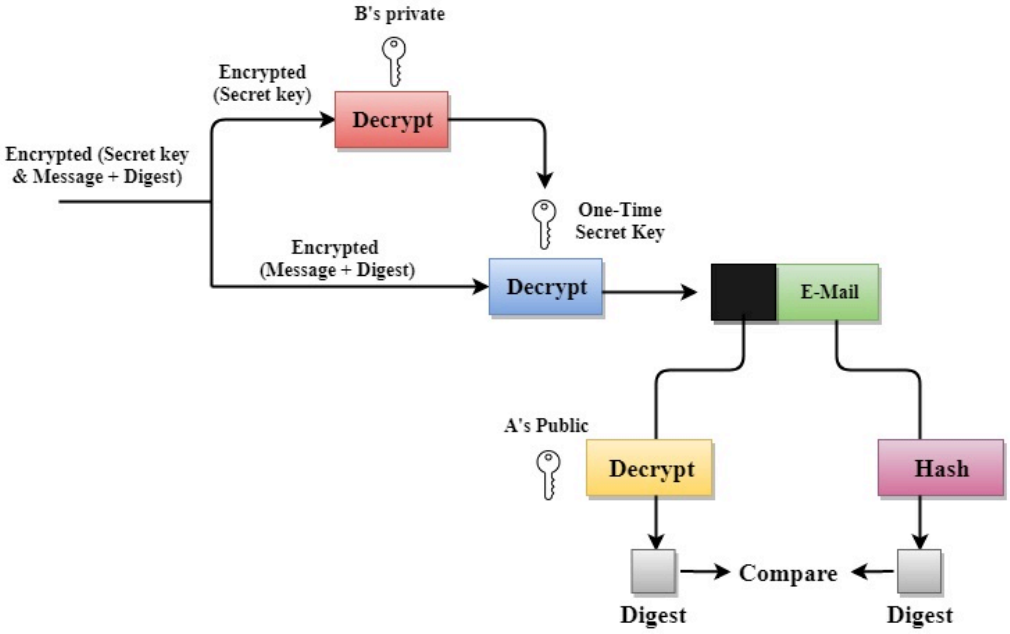

Figure 12: Illustration of the PGP encryption and decryption process, demonstrating key exchange, authentication, and hashing mechanisms [11].

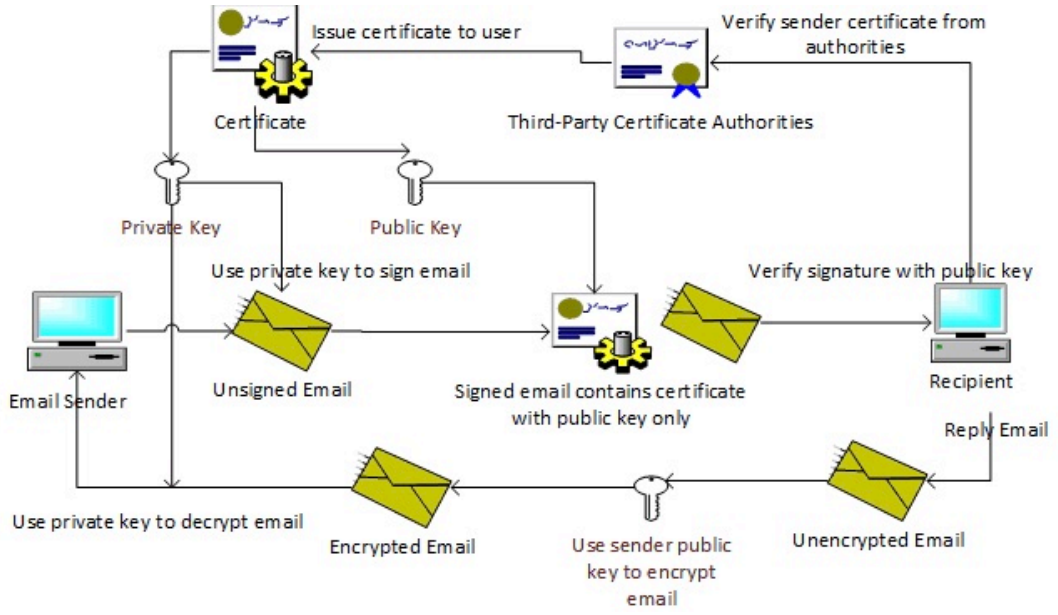

Figure 13: Illustration of the S/MIME process, depicting encryption, digital signatures, and certificate-based authentication [12].

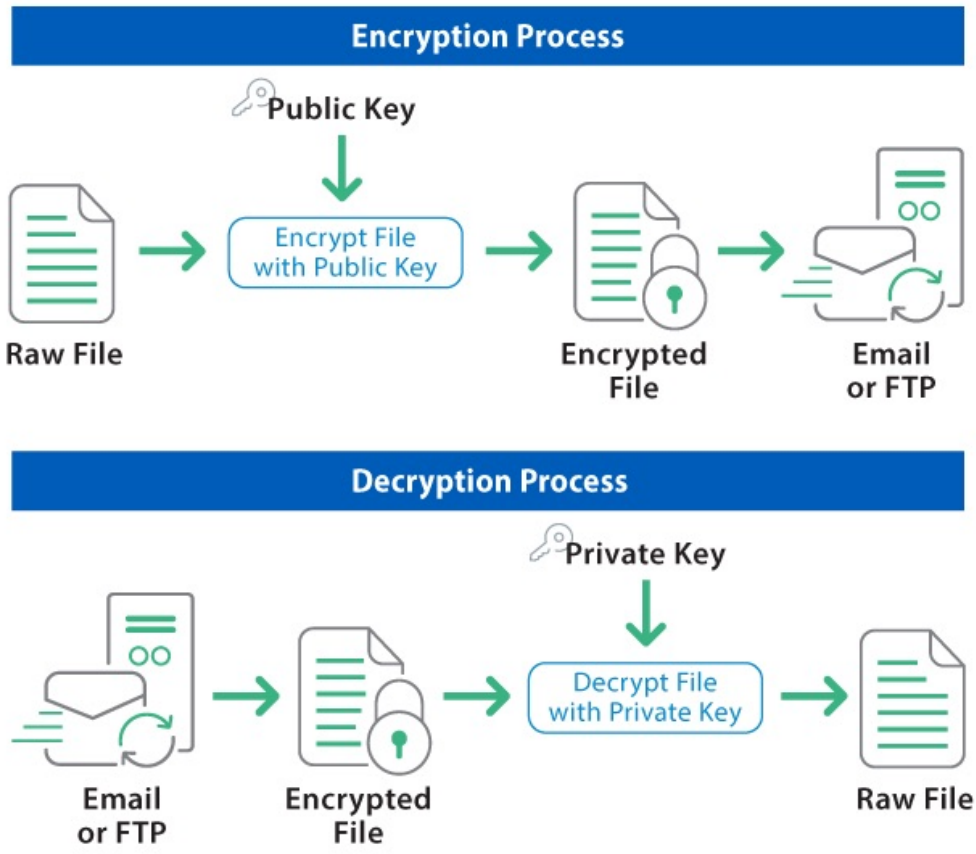

Figure 14: Illustration of OpenPGP encryption and decryption, demonstrating key-based file security [13].

- **OpenPGP (1997, IETF) -** OpenPGP, standardized by the IETF in 1997, is an open standard for data encryption and signing, designed to provide a secure and non-proprietary framework for cryptographic protection. Derived from the original PGP software, OpenPGP enables encryption of emails, files, and directories, ensuring privacy and security. Unlike S/MIME, which relies on a centralized trust model, OpenPGP uses a web of trust, allowing users to validate each other's keys without requiring a certificate authority [99]. Figure 14 illustrates the OpenPGP encryption and decryption process, detailing using public and private key cryptography to secure data transmission.

- **WPA2/3 (Wi-Fi Protected Access 2/3) - WPA2 (2004, Wi-Fi Alliance) and WPA3 (2018, Wi-Fi Alliance) -** Wi-Fi Protected Access 2 (WPA2), introduced by the Wi-Fi Alliance in 2004, and its successor, WPA3 (2018), are security protocols designed to protect wireless networks from unauthorized access. WPA2 employs AES encryption and provides high security for wireless communications. At the same time, WPA3 enhances security further by introducing features such as stronger encryption, protection against offline brute-force attacks, and individualized data encryption. WPA3 also improves security for open networks through Opportunistic Wireless Encryption (OWE) [100]. Figure 15 illustrates the architecture of WPA2-Personal and WPA2-Enterprise, highlighting the key differences in security mechanisms and authentication approaches.

The study of encryption techniques and services underscores their indispensable role in securing digital communications. As encryption technologies evolve, so must the analytical frameworks designed to classify and detect encrypted

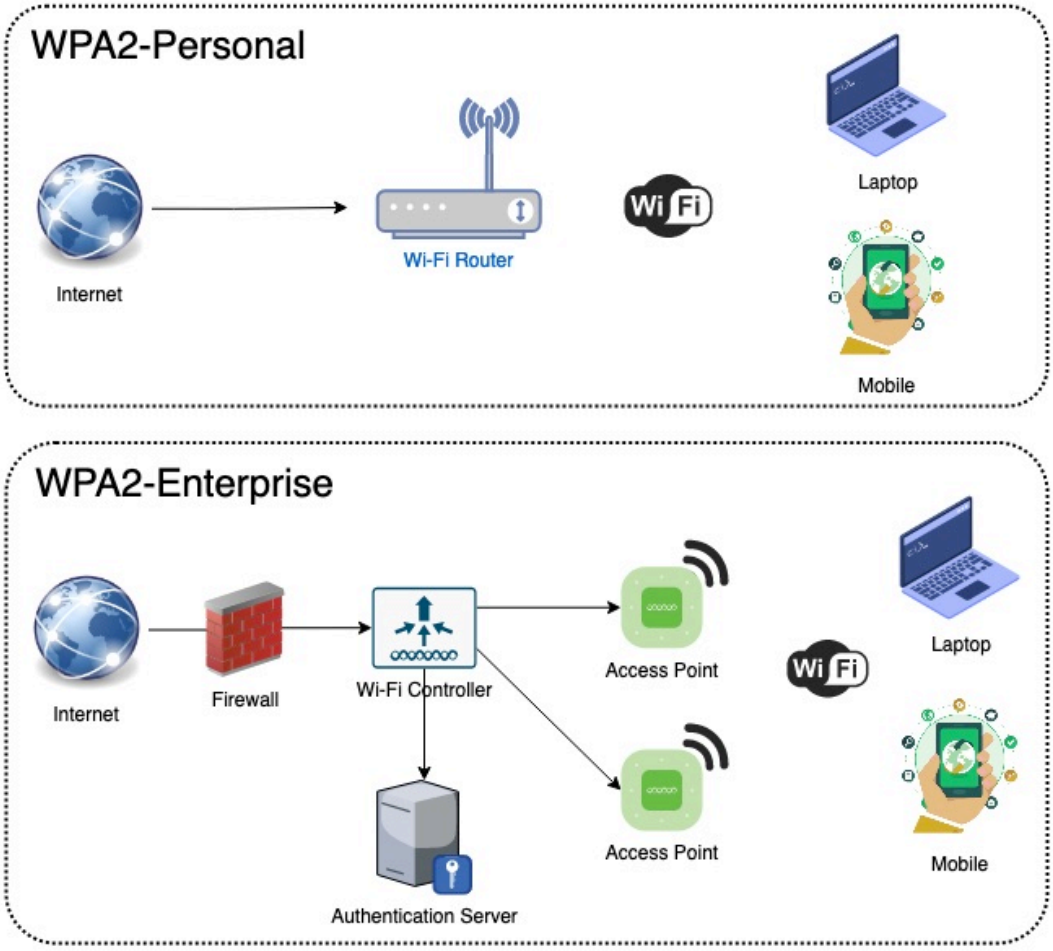

Figure 15: Illustration of WPA2-Personal and WPA2-Enterprise security architectures, showing authentication mechanisms and access control [14].

traffic accurately. Understanding these encryption mechanisms is critical for developing robust detection strategies that balance privacy with security, ensuring effective monitoring without compromising user confidentiality.
The following subsection discusses **Previous Works**, which provides a comprehensive review of various studies in the field of Encrypted Traffic Analysis (ETA), focusing on machine learning (ML), deep learning (DL), and hybrid models. This discussion highlights advancements in traffic classification methodologies, the effectiveness of different model architectures, and the challenges researchers have addressed in adapting ETA frameworks to evolving encryption protocols.

2.2 Previous Works

Encrypted Traffic Analysis (ETA) has become a critical area of research, driven by the growing adoption of encryption protocols that obscure network traffic from traditional monitoring techniques. This section provides a structured and comprehensive overview of the methodologies employed in detecting and classifying encrypted traffic, offering insights into the evolution of research in this domain.
To systematically analyze existing approaches, this thesis presents an **expansive taxonomy of encrypted traffic classification models**, capturing the breadth of methodologies employed in contemporary research, as illustrated in **Figure 1**. This taxonomy classifies existing techniques into three primary categories: **Machine Learning Approaches, Deep Learning Approaches, and Hybrid Approaches**, each representing distinct paradigms in encrypted traffic classification.

- **Machine Learning Approaches** include foundational algorithms such as **J48, Random Forest (RF), and Instance-Based Learning (IBk)**, which have been instrumental in developing baseline classification models. These approaches rely on statistical learning from handcrafted features, offering computational efficiency but often struggling with complex traffic patterns.
- **Deep Learning Approaches** extend beyond traditional ML models, leveraging hierarchical feature extraction and automated representation learning. This category encompasses architectures ranging from **basic neural networks to advanced models** such as **Convolutional Neural Networks (CNNs) with multi-head attention, Spiking Neural Networks (SNNs), Generative Adversarial Networks (GANs), and Transformer-based models**. These techniques have demonstrated superior performance in capturing intricate traffic behaviors, making them well-suited for encrypted traffic classification.
- **Hybrid Approaches** integrate machine learning and deep learning methodologies, combining the interpretability and efficiency of traditional algorithms with the representational power of neural networks. These include ensembles such as **LSTM + RF, SVM + KNN, and Decision Trees (DT) fused with Recursive Feature Elimination (RFE) and Multi-Layer Perceptrons (MLP)** to enhance predictive performance and adaptability to dynamic network conditions.

This taxonomy delineates the structured evolution of encrypted traffic analysis techniques and provides a foundation for evaluating their efficacy in real-world scenarios. The **following subsections critically analyze these methodologies**, emphasizing their strengths, limitations, and practical applications. Through an extensive review of **high-impact literature**, we ensure a comprehensive and methodologically rigorous assessment tailored to the field's most influential and innovative contributions. Finally, we synthesize key limitations into a cohesive conclusion, identifying research gaps and future directions that can advance encrypted traffic detection and classification.

2.2.1 Machine Learning Models in Encrypted Traffic Detection

Machine learning (ML) techniques have played a pivotal role in classifying encrypted traffic, leveraging statistical patterns and feature-based learning to distinguish between different types of encrypted flows. This section reviews key ML-based approaches, analyzing their effectiveness in handling real-world encrypted network traffic. Various classifiers, including decision trees, random forests, and ensemble methods, have been employed to enhance detection accuracy while maintaining computational efficiency. By exploring these techniques, we provide insights into the evolution of ML-driven encrypted traffic classification and its impact on network security.

The paper by Zaki et al. [15] addresses the critical issue of encrypted traffic classification at multiple granularity levels, specifically focusing on network security and administrative efficiency. The authors propose GRAIN, a granular multi-label classification method using a classifier chain that uniquely classifies network traffic at the application name, inter-application, and intra-application service levels. Illustrated in Fig. 16, This approach utilizes two chained Random Forest classifiers that leverage seven novel statistical features derived from packet payload lengths, which remain effective despite encryption, thus preserving user privacy. The performance evaluation demonstrates that GRAIN achieves high classification accuracy, with F-measure scores of 99%, 93%, and 88% at various levels, highlighting its effectiveness and potential for implementation in complex network environments.

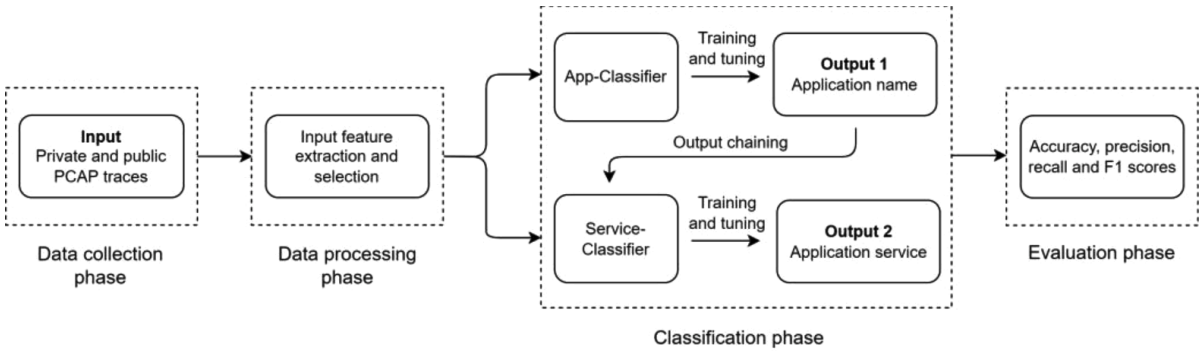

Figure 16: GRAIN model architecture proposed by [15]

In the work by Shi, Dong [16], the critical challenge of classifying network traffic types in imbalanced datasets is tackled. The author proposes a novel method utilizing a cost-sensitive Support Vector Machine (SVM) with active learning, termed CMSVM, which aims to adjust weights to resolve the imbalance problem effectively and dynamically. The methodology is detailed in Fig. 17 of the paper, illustrating the flowchart of the CMSVM process. This method optimizes the classification accuracy by iteratively adjusting the training process based on the imbalanced data characteristics. Applying the CMSVM algorithm to two different datasets, the MOORE_SET and NOC_SET, the results demonstrate a significant enhancement in precision, recall, and overall accuracy, indicating its efficacy in handling complex and varied network traffic scenarios.

Yao et al. [101], authors tackle the critical challenge of encrypted traffic classification in a network security context. They present a novel traffic classification model named MGHMM, which combines Gaussian mixture models and Hidden Markov Models to distinguish between different types of encrypted traffic effectively. MGHMM significantly enhances the ability to monitor and control encrypted traffic, improving both the detection of various encrypted protocols and the identification of obfuscated traffic. This model's application demonstrates substantial

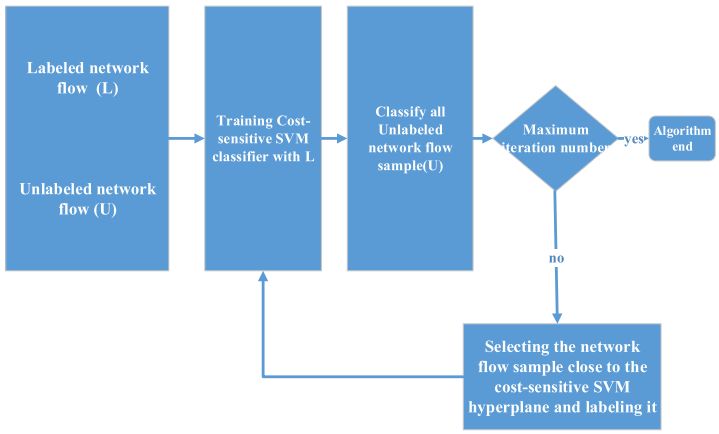

Figure 17: CMSVM model architecture proposed by [16]

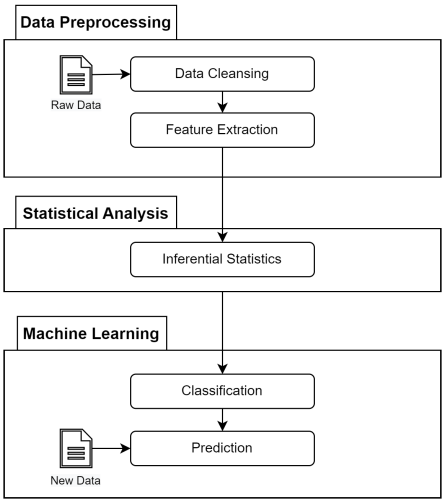

Figure 18: Model architecture proposed by [17]

improvements over existing methods, verified through comprehensive testing across multiple dataset scenarios.

Using machine learning techniques, Choorod et al. [17], the authors address the challenge of classifying Tor traffic from nonTor traffic using encrypted payloads. They propose a novel machine learning approach that relies on character statistical-based features extracted directly from encrypted payloads, which is illustrated in Fig.18. This method significantly enhances the efficiency and accuracy of traffic classification, demonstrated by their method's ability to achieve an average accuracy rate of 95.65% across various application types without the need for multiple packets or in-depth packet inspection, thus maintaining user privacy and network efficiency.
Machine learning-based approaches have demonstrated considerable success in encrypted traffic classification, offering scalable and interpretable models for identifying encrypted traffic patterns. However, as encryption methods grow more sophisticated, ML techniques alone may struggle to adapt to evolving attack vectors and obfuscation strategies. This limitation underscores the need for hybrid models integrating ML with deep learning and other advanced analytical techniques.

2.2.2 Deep Learning Models in Encrypted Traffic Detection

Deep learning (DL) has emerged as a transformative force in encrypted traffic classification, enabling models to learn hierarchical representations of traffic patterns without extensive manual feature engineering. This section explores the application of DL methodologies, including neural networks, convolutional models, and generative architectures, to encrypted traffic analysis. By automatically leveraging their ability to extract complex features from raw traffic data, DL models have set new benchmarks in detection accuracy and adaptability.

2.2.2.1 Auto Encoders

Autoencoders (AEs) have been widely utilized in encrypted traffic classification due to their ability to perform unsupervised feature learning and anomaly detection. These architectures encode traffic patterns into lower-dimensional representations, facilitating efficient detection and classification. This section explores various AE-based models, assessing their efficacy in extracting meaningful patterns from encrypted flows while minimizing reconstruction errors.

In the work by Lv, S. et al. [18], the authors introduce an innovative one-class classification model, AAE-DSVDD, designed explicitly for VPN traffic identification. The model leverages Adversarial AutoEncoders (AAE) to construct an aggregated posterior distribution that closely matches a predefined prior, which is then used to constrain the output of Deep Support Vector Data Description (DSVDD). This approach mitigates the hypersphere collapse problem—a common issue in DSVDD models—by ensuring a well-structured latent space representation. The proposed model demonstrates superior performance in identifying VPN traffic, particularly distinguishing between unseen, known classes (UKCs) and unseen, unknown classes (UUCs), a crucial challenge in encrypted traffic classification. Their experimental results on the ISCXVPN dataset show that AAE-DSVDD outperforms traditional one-class classification models such as OCSVM, iForest, and GANomaly, achieving higher AUC scores across traffic types. The overall methodology and architecture of the AAE-DSVDD model are illustrated in Fig. 19.

Aceto, G. et al. [19] introduce DISTILLER, a deep learning-based classifier that leverages a multimodal multitask approach for encrypted traffic classification. Unlike traditional models that rely on a single input modality, DISTILLER capitalizes on heterogeneous traffic features, extracting both intra-modality and inter-modality dependencies. This enables the classifier to perform multiple classification tasks concurrently, including encapsulation identification, traffic type classification, and application recognition. The model incorporates a two-phase pre-training and fine-tuning procedure to avoid modality dominance, ensuring robust feature learning across diverse traffic patterns. Evaluation of the ISCX VPN-nonVPN dataset demonstrates DISTILLER's superiority over single-task and single-modality deep learning models, achieving higher classification accuracy with reduced computational complexity. The overall architecture of the DISTILLER framework is illustrated in Fig. 20.
Autoencoders provide a promising direction for encrypted traffic analysis by leveraging unsupervised learning to uncover latent structures in network data. Their ability to detect anomalies and optimize feature representations makes them valuable for encrypted traffic classification. However, their reliance on reconstruction-based learning poses limitations in handling dynamic and adversarial traffic patterns, necessitating further refinements.

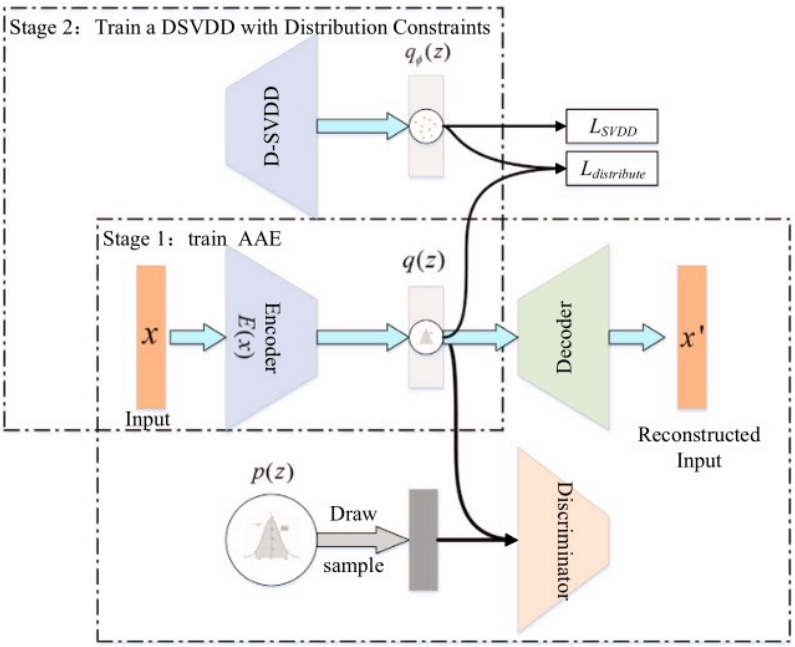

Figure 19: AAE-DSVDD model architecture proposed by [18].

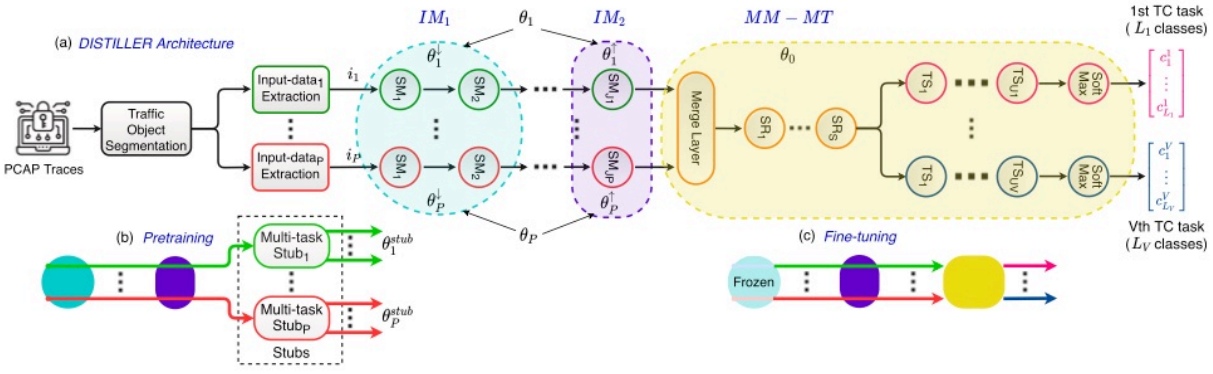

Figure 20: DISTILLER model architecture proposed by [19].

2.2.2.2 Neural Networks

The exploration of neural networks in network security represents a critical evolution in identifying and classifying encrypted traffic [102]. Neural network models, with their profound learning capabilities, offer nuanced approaches to understanding complex data patterns, thereby enhancing the identification of cyber threats within encrypted data flows [103]. This section outlines significant advancements across various neural network architectures, each contributing uniquely to ETC. It includes a dedicated subsection on models integrating CNNs to enhance feature extraction and classification accuracy.

The use of neural networks in the domain of ETC has witnessed several innovative contributions that enhance network security systems' functionality and accuracy. Jorgensen, S. et al. [104] explore the application of Probabilistic Neural Networks (PNNs) with Uncertainty Quantification (UQ) to enhance VPN application labeling in encrypted network traffic. Their approach leverages prototypical networks combined with relative Mahalanobis distance-based out-of-distribution (OOD) detection, enabling the model to assign confidence scores to its predictions. This methodology improves the classifier's robustness by distinguishing predictive uncertainty (when the model is indecisive between known labels) from model uncertainty (when encountering unseen application traffic). The framework also incorporates wavelet-transformed features and flow statistics, ensuring adequate traffic characterization even with limited labeled data. The model achieves an F1-score of 0.98 on their dataset and successfully adapts to enterprise network traffic by systematically integrating new classes.

Song, Z. et al. [20] propose I²RNN, an Incremental and Interpretable Recurrent Neural Network designed to address key limitations in encrypted traffic classification. Unlike conventional deep learning models that require retraining on the entire dataset when new traffic types emerge, I²RNN introduces a fingerprint learning process that enables incremental adaptation by training additional independent modules for newly encountered traffic types. This modular design reduces computational overhead and allows real-time updates in dynamic network environments. Moreover, I²RNN enhances model interpretability by ranking time-series features for each traffic type and calculating inter-class distances, facilitating transparency in encrypted traffic classification. Experimental results on CIC-IDS2017 and ISCXVPN2016 datasets demonstrate that I²RNN achieves state-of-the-art performance with higher adaptability, interpretability, and classification accuracy than traditional deep learning methods. The architectural workflow of I²RNN is illustrated in Fig. 21.

Zhou, K. et al. [21] propose a hybrid approach integrating entropy estimation and neural networks for encrypted traffic classification. Their method introduces a two-phase classification framework, where entropy estimation first differentiates encrypted from plaintext traffic, followed by a deep neural network (DNN) that performs application-level classification on encrypted flows. The entropy-based method relies on Shannon entropy calculations and Monte Carlo estimations to measure statistical randomness in packet sequences, effectively flagging encrypted connections. The DNN model, designed with a 23-feature input layer, a 100-neuron hidden layer, and an 8-class output layer, is trained to classify encrypted traffic into distinct application types, such as VoIP, streaming, and P2P communications. The proposed combined method improves classification precision by 1–7 percentage points compared to traditional ML techniques while achieving notable performance gains of nearly 30 percentage points in specific application categories. The combined entropy estimation and neural network approach workflow is illustrated in Fig. 22.

Pathmaperuma, M. H. et al. [22] introduce a Deep Neural Network (DNN) framework tailored for fine-grained in-app activity classification in encrypted traffic. Their approach enhances traditional encrypted traffic classification (ETC) models by integrating a probability-based filtering mechanism that identifies and discards previously unseen

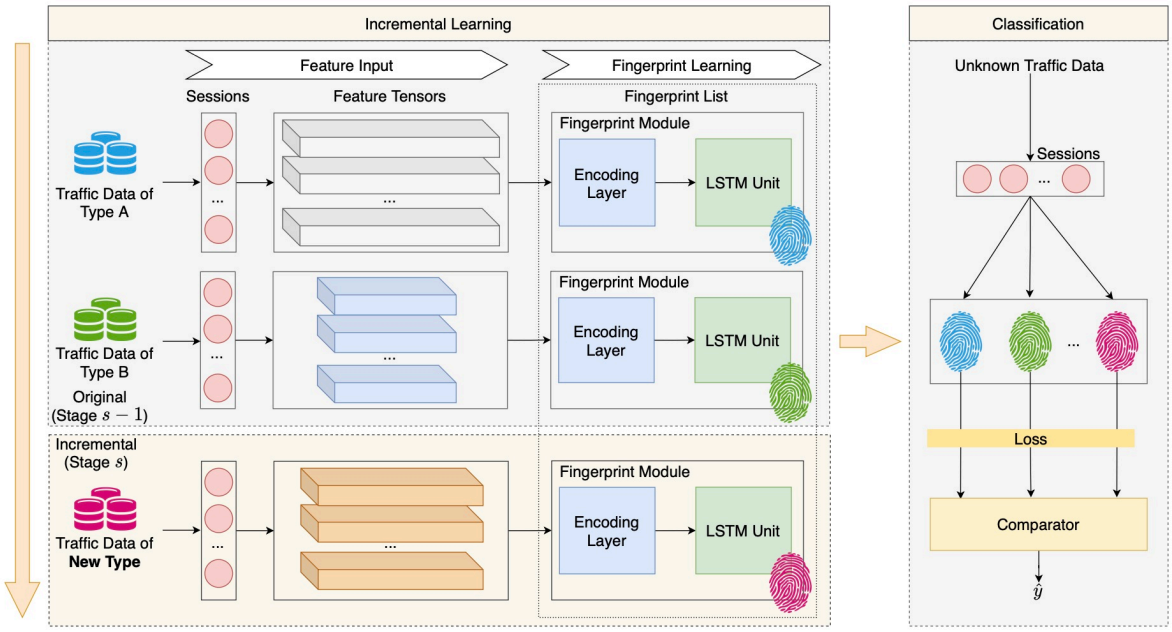

Figure 21: I²RNN model architecture and incremental learning framework proposed by [20].

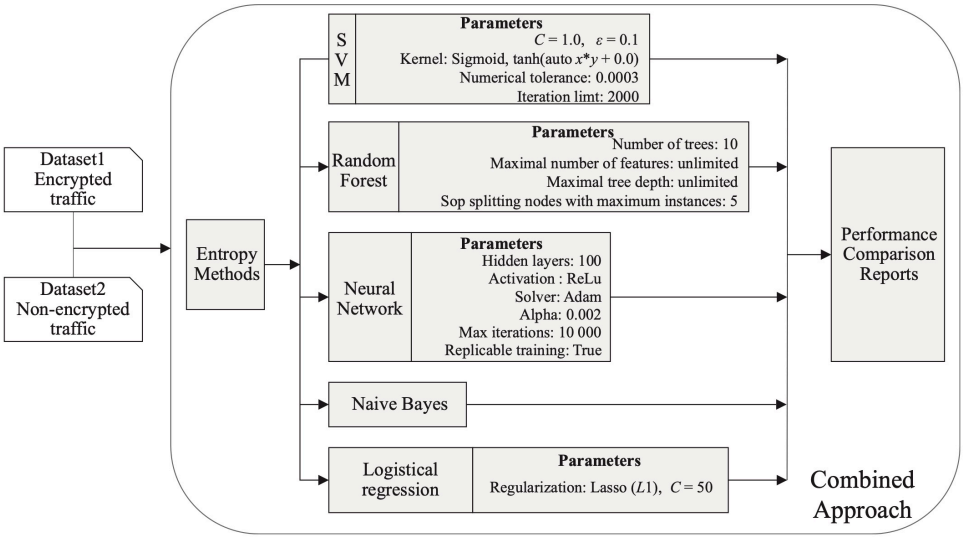

Figure 22: Workflow of the entropy estimation and neural network-based classification model proposed by [21].

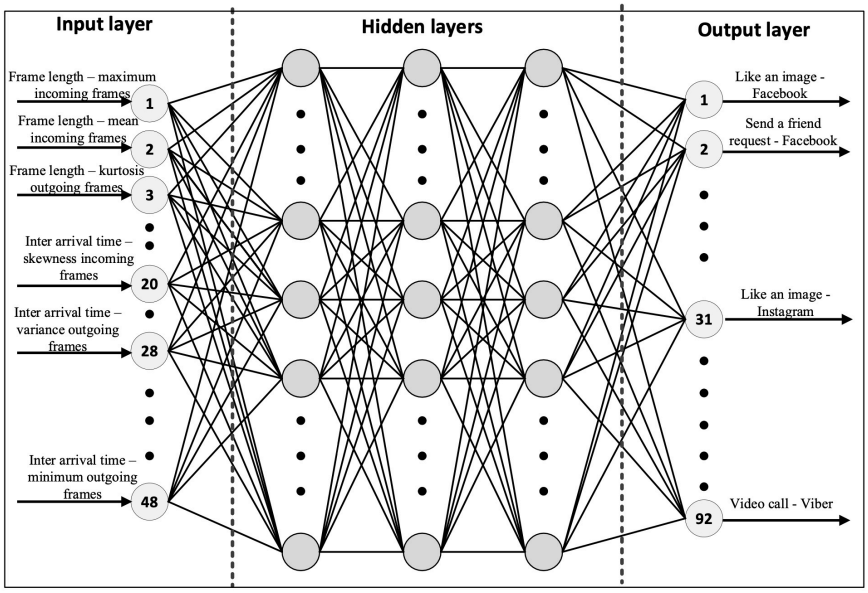

Figure 23: Deep learning-based encrypted traffic classification and unknown data detection framework proposed by [22].

data, significantly improving classification robustness. Unlike conventional methods that struggle with untrained data instances, their unknown data detection module leverages the probability distribution of the DNN's output layer to differentiate between known and unknown traffic patterns. The framework uses a time-segmented analysis, effectively capturing short sequences of encrypted flows to identify specific in-app activities without relying on complete session data. Empirical results demonstrate that their model achieves over 90% accuracy in recognizing previously trained in-app activities and 79% accuracy in filtering unknown activity data, outperforming existing deep learning-based approaches. The proposed framework and its operational workflow are illustrated in Fig. 23.
Rasteh, A. et al. [23] explore the application of Spiking Neural Networks (SNNs) for encrypted internet traffic classification, marking the first use of biologically inspired neural models in this domain. Their approach leverages Surrogate Gradient Learning, a technique that overcomes the limitations of standard backpropagation in SNNs by approximating gradient updates using differentiable surrogate functions. The model processes encrypted traffic using histogram-based feature extraction, capturing packet size distributions and inter-arrival times while preserving temporal correlations—a critical advantage over traditional deep learning models. The proposed feedforward SNN architecture, consisting of a single fully connected hidden layer, demonstrates remarkable classification performance, achieving an accuracy of 95.9% on the ISCX VPN-nonVPN dataset, surpassing prior approaches such as Flowpic and Deep Packet. The workflow of the proposed SNN-based traffic classification model is illustrated in Fig. 24.
Xu, Y. et al. [24] introduce FastTraffic, a lightweight deep learning-based framework optimized for real-time encrypted traffic classification (ETC) on resource-constrained network devices. The model leverages a Multilayer Perceptron (MLP) with N-gram feature embedding, ensuring an optimal trade-off between computational efficiency and classification accuracy. Unlike conventional deep learning models that require extensive preprocessing and large input sizes, FastTraffic employs packet-level tokenization and truncation strategies to process encrypted network packets efficiently. The N-gram embedding technique extracts structural and sequential byte-level features, enhancing the model's capacity to differentiate encrypted traffic patterns with minimal computational overhead. Compared to eight state-of-the-art ETC methods, FastTraffic achieves superior classification performance while maintaining a small model size of only 0.43M, making it ideal for real-time deployment on mainstream network

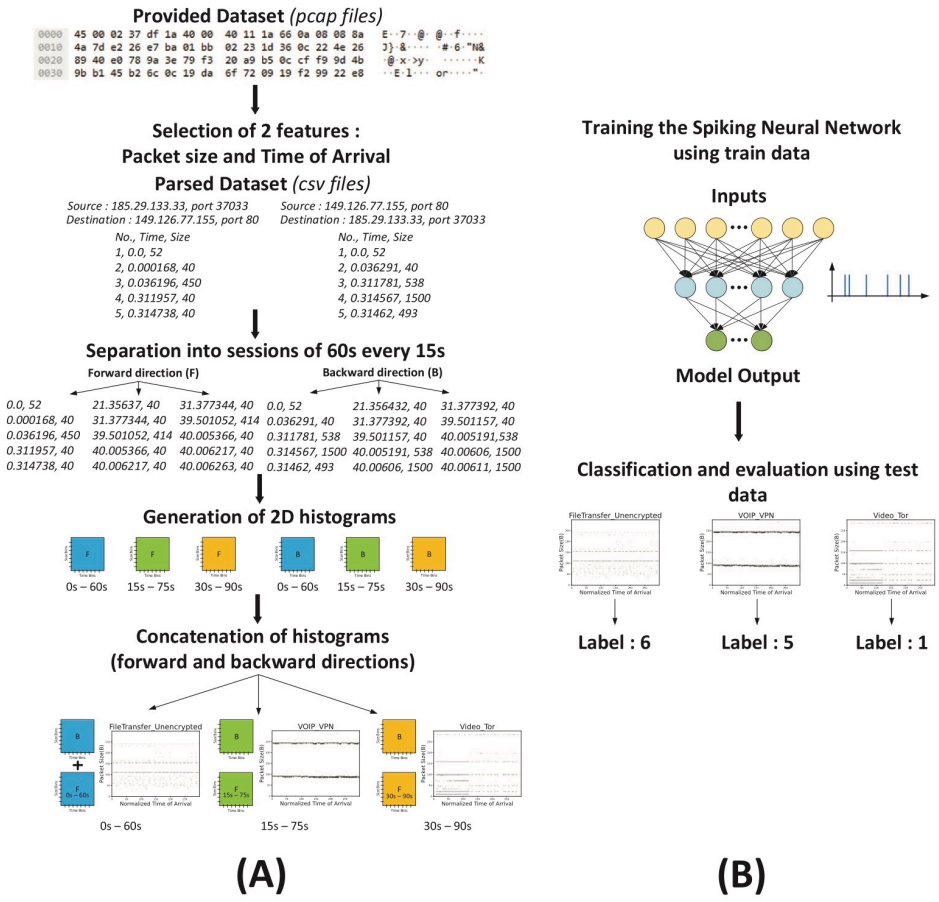

Figure 24: Spiking Neural Network-based encrypted traffic classification model proposed by [23].

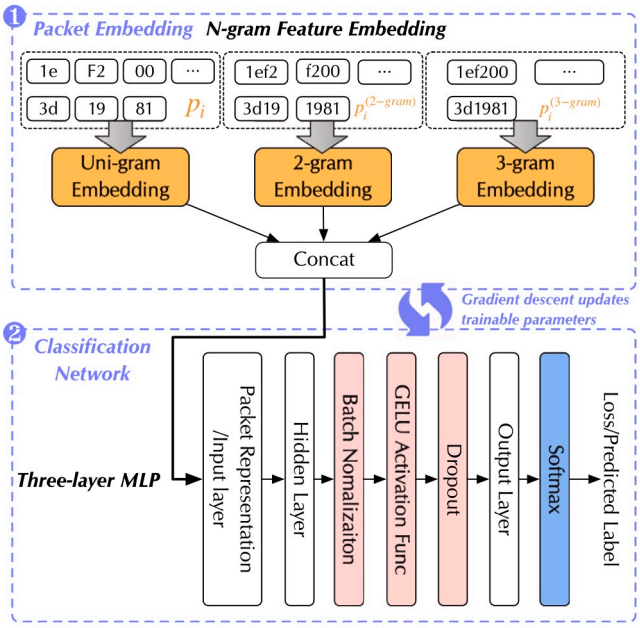

Figure 25: FastTraffic classification model architecture proposed by [24].

devices. The workflow and architecture of the FastTraffic classification model are illustrated in Fig. 25.

2.2.2.2.1 Convolutional Neural Networks (CNNs) CNNs have become a cornerstone in network security, especially for the ETC [105]. By exploiting spatial hierarchies in data, CNNs offer an unparalleled ability to extract features and identify patterns within complex, encrypted data streams[106]. This segment showcases various innovative approaches leveraging CNNs, each addressing specific challenges in analyzing encrypted traffic [28].

The utilization of CNNs in the analysis of encrypted traffic has evolved with several innovative contributions enhancing the accuracy and efficiency of classification. He, Y. et al. [25] introduce an image-based approach for encrypted traffic classification, leveraging Convolutional Neural Networks (CNNs) to achieve end-to-end feature extraction and classification. Unlike traditional statistical and machine learning-based methods that require manual feature engineering, this approach converts network session's first few non-zero payload sizes into grayscale images, capturing intrinsic traffic patterns. The converted images are then processed through a 1D-CNN, automatically extracting salient traffic characteristics, improving classification accuracy and reducing computational overhead. The model is evaluated on the ISCX VPN-nonVPN dataset, demonstrating an F1-score of 98.64% for conventional encrypted traffic and 99.55% for VPN traffic classification, outperforming traditional ML-based classifiers. The workflow of the image-based classification framework is illustrated in Fig. 26.

In the work by Moreira, R. et al. [26], the authors extend the application of Convolutional Neural Networks (CNNs) in encrypted traffic classification by integrating them with Reinforcement Learning (RL) to optimize TOR traffic classification in real-time. Their proposed framework employs adaptive packet sampling, where a Deep Q-Network (DQN)-based RL agent dynamically adjusts the sampling rate to optimize classification accuracy while reducing computational overhead. This method allows for real-time TOR and non-TOR traffic classification, making it particularly suitable for high-performance networks with stringent processing constraints. Additionally, the authors incorporate SqueezeNet, a lightweight CNN architecture, to ensure rapid prediction capabilities without

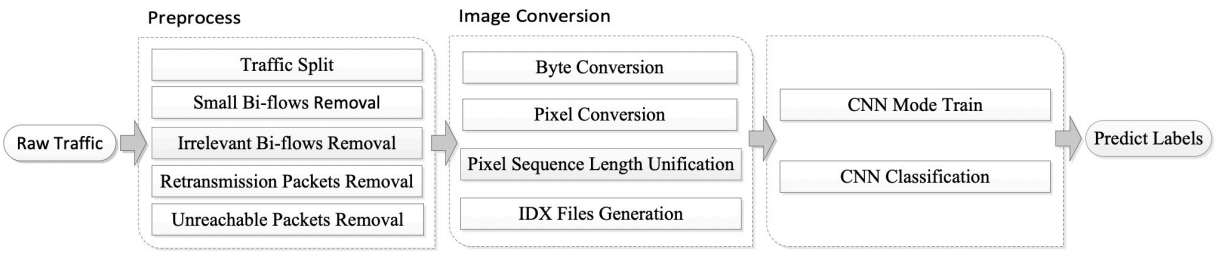

Figure 26: End-to-end image-based encrypted traffic classification framework proposed by [25].

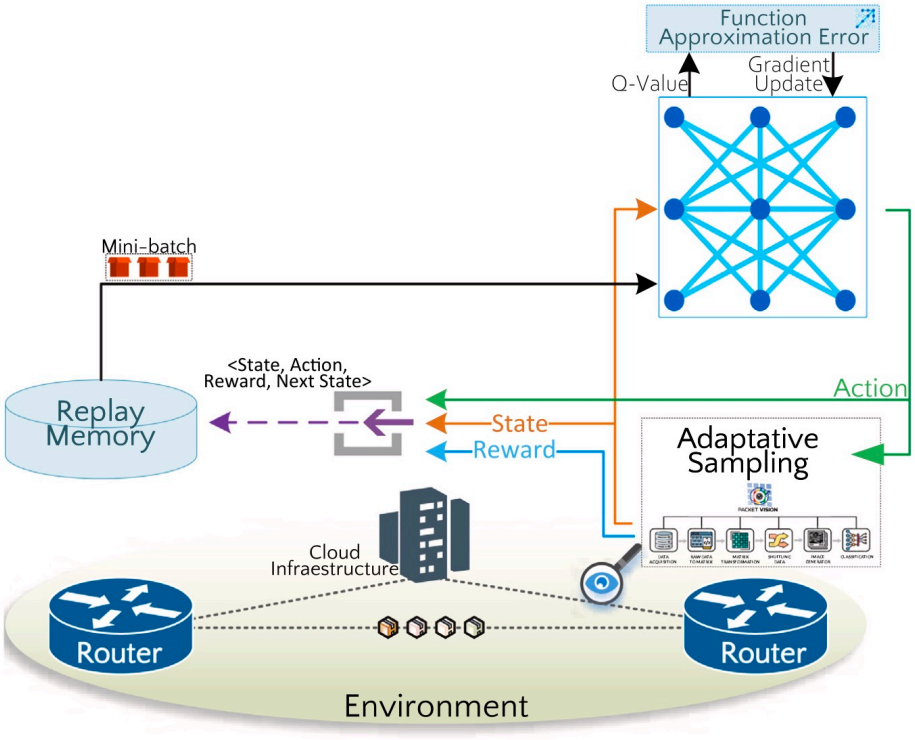

Figure 27: Adaptive CNN-RL-based TOR traffic classification framework proposed by [26].

compromising classification accuracy. Evaluation of the ISCXVPN2016 and ISCXTor2017 datasets demonstrates that the proposed approach achieves 99.84% classification accuracy, outperforming existing Darknet traffic classification methods. Fig. 27 illustrates the adaptive sampling framework combining CNN and RL.

Cheng, J. et al. [27] propose MATEC, a lightweight neural network framework that integrates multi-head attention mechanisms with Convolutional Neural Networks (CNNs) to enhance real-time encrypted traffic classification, unlike traditional deep learning models that rely on recurrent neural networks (RNNs) for sequential feature extraction, MATEC leverages multi-head attention to allow parallel processing of packet interactions, significantly reducing computational complexity. The model extracts global (flow-level) and local (packet-level) features through a thin module, ensuring computational efficiency while maintaining high classification accuracy. Transfer learning enables MATEC to adapt to new traffic types with minimal retraining, making it suitable for dynamic network environments. Evaluation across three encrypted traffic datasets demonstrates that MATEC achieves higher accuracy while reducing the number of model parameters to just 1.8% of conventional CNN-based models, ensuring faster inference times. The architectural design of the MATEC framework is illustrated in Fig. 28.

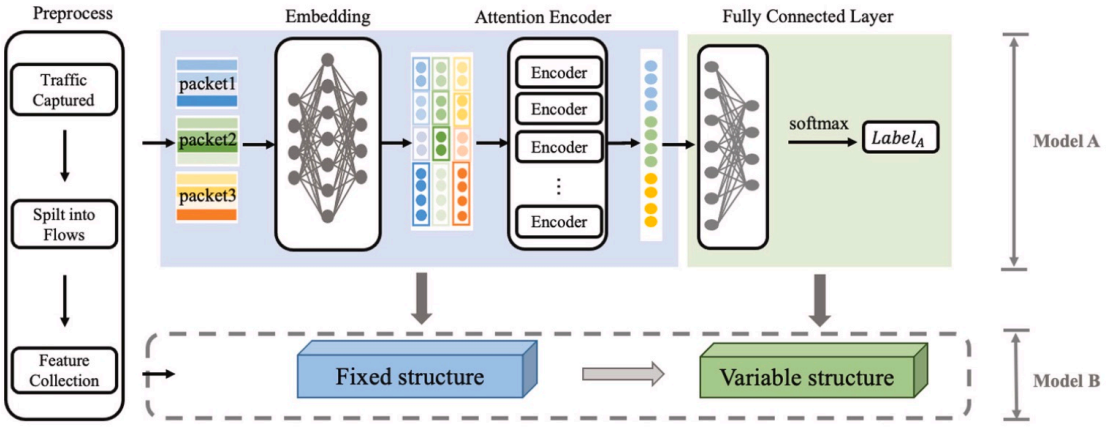

Figure 28: MATEC framework integrating multi-head attention with CNNs for encrypted traffic classification proposed by [27].

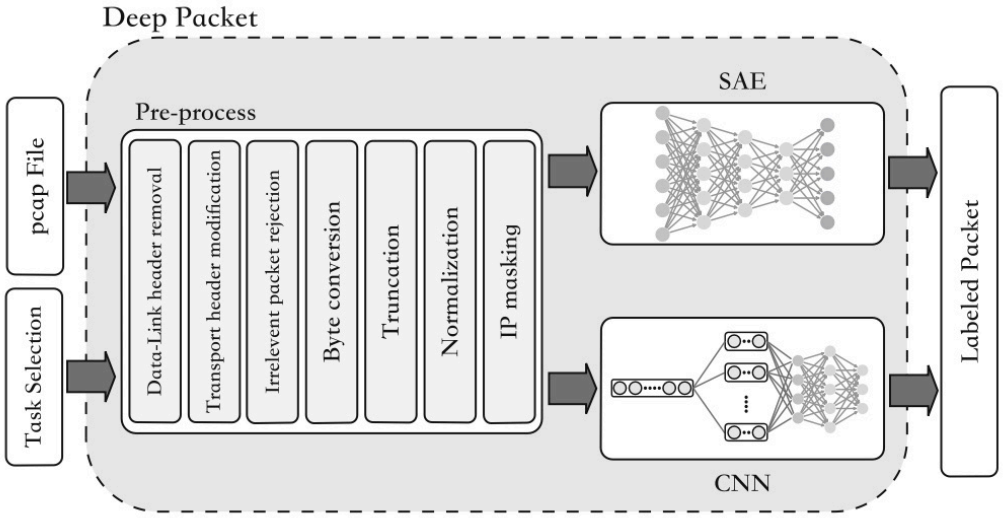

Figure 29: Deep Packet framework integrating CNNs and SAEs for encrypted traffic classification, proposed by [28].

Lotfollahi, M. et al. [28] propose Deep Packet, a novel encrypted traffic classification framework that combines Convolutional Neural Networks (CNNs) and Stacked Autoencoders (SAEs) to enhance both feature extraction and classification accuracy. Unlike conventional models that rely on manually engineered statistical features, Deep Packet automatically learns hierarchical traffic representations by leveraging CNNs for spatial feature extraction and SAEs for unsupervised deep feature encoding. The model effectively classifies both VPN and non-VPN traffic at a granular level, simultaneously handling traffic characterization and application identification tasks. Evaluations on the ISCX VPN-nonVPN dataset demonstrate that Deep Packet outperforms prior methods, achieving an F1-score of 0.98 for application classification and 0.94 for traffic characterization, surpassing traditional ML classifiers and earlier deep learning models. The architectural workflow of the Deep Packet classification framework is illustrated in Fig. 29.

Wang, M. et al. [29] propose a hybrid encrypted traffic classification framework, leveraging Convolutional Neural Networks (CNNs) and Stacked Autoencoders (SAEs) to enhance classification accuracy while mitigating information loss from raw traffic trimming. The CNN component extracts high-level spatial features from raw network traffic,

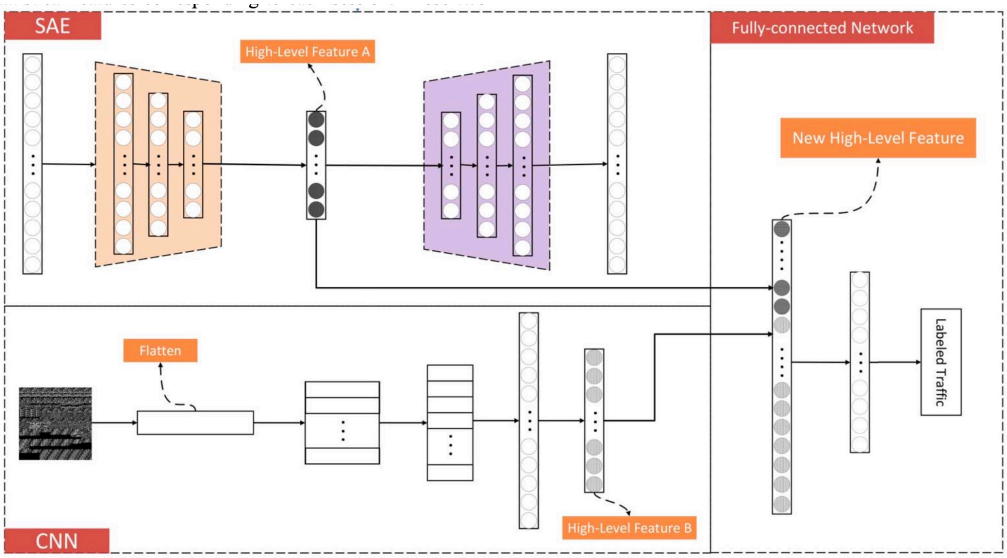

Figure 30: CNN-SAE-based encrypted traffic classification model proposed by [29].

while the SAE module encodes 26 statistical features, preserving essential session-level characteristics. The outputs from both models are then concatenated into a unified feature representation, significantly improving the classification granularity of VPN and non-VPN traffic. Evaluations on the ISCX VPN-nonVPN dataset demonstrate that this approach achieves an F1-score of 0.98, outperforming traditional deep learning methods that rely solely on raw traffic features. The architectural workflow of the proposed CNN-SAE classification model is illustrated in Fig. 30.

Soleymanpour, S. et al. [30] address the challenges of class imbalance in encrypted traffic classification by introducing CSCNN, a Cost-Sensitive Convolutional Neural Network that incorporates a cost-aware learning strategy into its architecture. Unlike traditional deep learning models, CSCNN assigns higher misclassification penalties to minority classes through a cost matrix formulated during preprocessing, ensuring better representation learning for rare traffic types. The model enhances classification sensitivity by dynamically modifying class weights during training, effectively mitigating overfitting to dominant traffic categories. Evaluations on the ISCX VPN-nonVPN dataset reveal that CSCNN achieves higher precision and recall for minority classes, surpassing deep learning-based models (e.g., Deep Packet and Datanet) and traditional ML classifiers. The schematic workflow of the CSCNN framework is illustrated in Fig. 31.

Xu, L. et al. [31] introduce ETCNet, a novel encrypted traffic classification model that leverages Siamese Convolutional Networks (SCNs) to classify network traffic while addressing dataset limitations effectively. Unlike conventional deep learning classifiers, ETCNet transforms multi-class classification into a binary similarity problem, significantly reducing the amount of labeled data required for training. The model consists of two key components: a Siamese Neural Encoder, which extracts low-dimensional embeddings from encrypted traffic flows, and an Application Prediction Module, which evaluates flow similarity to determine whether two traffic streams belong to the same application. This design enables ETCNet to achieve high accuracy with as few as 40 flows per application, making it particularly effective in low-data environments. The architecture of the Siamese Convolutional Network used in ETCNet is illustrated in Fig. 32.

Lin, C. Y. et al. [32] propose an efficient encrypted traffic classification framework that integrates Convolutional Neural Networks (CNNs) with Bidirectional Gated Recurrent Units (Bi-GRUs) to enhance both spatial and temporal feature extraction. Unlike conventional deep learning methods that rely on static feature representations, this model

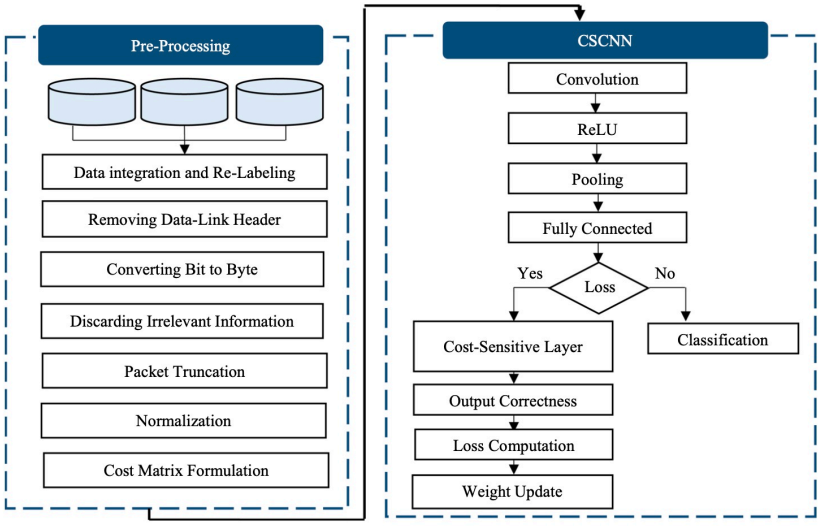

Figure 31: CSCNN framework incorporating cost-sensitive learning for encrypted traffic classification, proposed by [30].

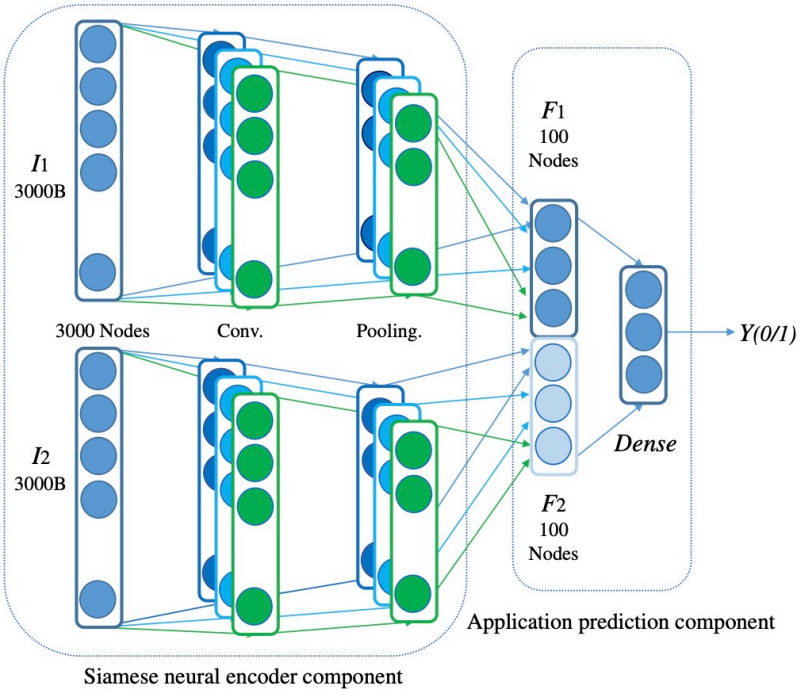

Figure 32: Siamese Convolutional Network architecture for encrypted traffic classification proposed by [31].

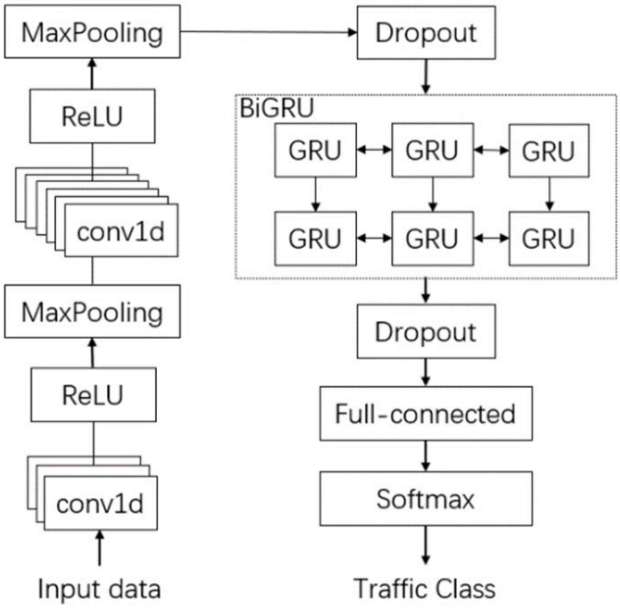

Figure 33: CNN-BiGRU framework for encrypted traffic classification proposed by [32].

leverages CNNs to extract spatial dependencies from network flows. At the same time, Bi-GRUs capture sequential patterns across packet streams, leading to a more comprehensive traffic characterization. Using Bi-GRUs instead of LSTMs reduces computational overhead while maintaining classification accuracy, making it a computationally efficient alternative. Evaluations on the ISCX VPN-nonVPN dataset demonstrate that this framework achieves 93.1% classification accuracy, outperforming traditional deep learning architectures such as CNN-LSTM and Deep Packet. The architectural design of the CNN-BiGRU framework is illustrated in Fig. 33.

Izadi, S. et al. [33] propose a hybrid deep learning framework that combines Convolutional Neural Networks (CNNs) with evolutionary optimization algorithms, precisely the Ant-Lion Optimizer (ALO) and Self-Organizing Maps (SOM) with fuzzy logic, to enhance encrypted traffic classification. Unlike conventional deep learning models that rely on static feature extraction, this approach automatically selects optimal traffic features by integrating CNN-based representations with ALO-driven feature selection, effectively reducing redundant features and improving classification accuracy. The SOM-based clustering mechanism further refines predictions by handling novel traffic instances dynamically. Evaluations on the ISCX VPN-nonVPN dataset demonstrate that this framework achieves 98% classification accuracy, outperforming traditional CNNs, LSTMs, and deep belief networks. The proposed CNN-ALO-SOM hybrid classification model is illustrated in Fig. 34.

Habibi Lashkari, A. et al. [34] introduce DIDarknet, a deep learning-based framework that leverages image-based feature extraction and CNN-based classification to detect and characterize darknet traffic. Their approach, DeepImage, transforms network flow features into grayscale images, enabling two-dimensional convolutional neural networks (2D-CNNs) to analyze encrypted darknet traffic more effectively. Unlike traditional feature-based detection techniques, this method automatically extracts spatial patterns from network flows, enabling enhanced classification of hidden services, VPN, and Tor traffic. The authors create a comprehensive darknet dataset by merging the ISCX VPN-nonVPN and ISCX Tor datasets, ensuring diverse application coverage. Experimental evaluations demonstrate that DIDarknet achieves 86% accuracy in darknet traffic characterization, outperforming existing VPN/Tor classifiers. The architectural framework of the DIDarknet model is illustrated in Fig. 35.

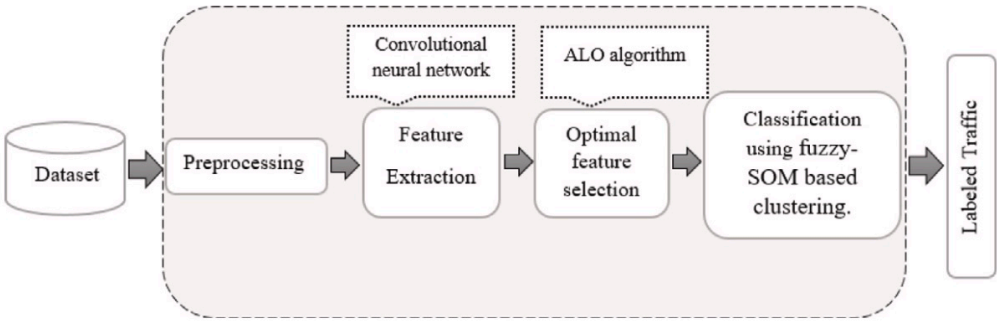

Figure 34: Hybrid CNN-Ant-Lion Optimization (ALO)-SOM framework for encrypted traffic classification, proposed by [33].

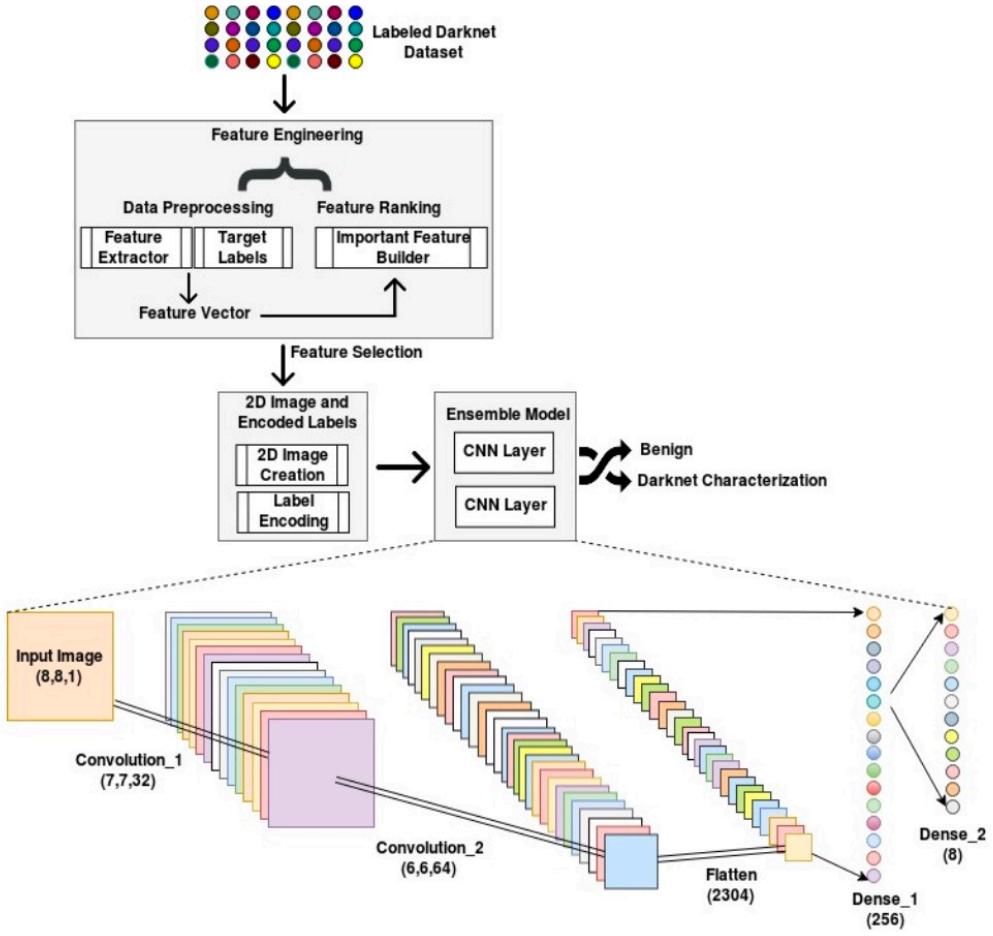

Figure 35: DeepImage-based darknet traffic classification framework proposed by [34].

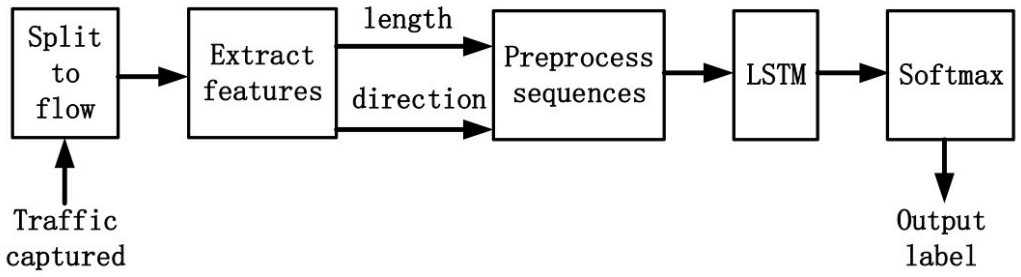

Figure 36: BFSN framework integrating CNNs and LSTMs for encrypted traffic classification, proposed by [35].

Tong, G. et al. [35] introduce BFSN (Bidirectional Flow Sequence Network). This hybrid deep learning framework integrates Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to enhance encrypted traffic classification (ETC). Unlike conventional CNN-based models that ignore temporal dependencies, BFSN constructs bidirectional flow sequences using packet length and directional information, allowing LSTM layers to capture sequential flow characteristics. This approach significantly improves classification accuracy without relying on deep packet inspection (DPI), ensuring compliance with encryption and privacy constraints. Experimental results on the ISCX VPN-nonVPN dataset show that BFSN achieves 91% accuracy, outperforming state-of-the-art ETC methods. The architectural design of the BFSN framework is illustrated in Fig. 36.

Yang, Z. et al. [36] propose AEFETA, an attention-enhanced encrypted traffic classification framework, which leverages Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (BiLSTM) networks to perform automated spatial-temporal feature extraction. Unlike traditional methods that rely on manually crafted statistical features, AEFETA introduces a self-learning feature selection process, dynamically adjusting feature importance using an attention mechanism. This ensures a context-aware classification process, where the model prioritizes the most relevant packet-level and session-level patterns. The AEFETA framework achieves an accuracy of 98.4% on the ISCX VPN-nonVPN dataset, outperforming conventional machine learning classifiers and existing deep learning-based ETC models. The architecture of the AEFETA framework is illustrated in Fig. 37.

Hu, C. et al. [37] present tCLD-Net, a transfer learning-based encrypted traffic classification model that integrates Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTMs), and pre-trained feature extraction layers. Unlike conventional deep learning approaches requiring large labeled datasets, tCLD-Net leverages knowledge transfer from a pre-trained CLD-Net model, enabling effective classification with as little as 20% of the target dataset. The architecture consists of a CNN module for spatial feature extraction, an LSTM module for capturing temporal dependencies, and a Dense module for final classification, where only select layers are fine-tuned to optimize performance. Evaluations on the ISCX VPN-nonVPN dataset show that tCLD-Net achieves 86% classification accuracy, outperforming non-transfer models while reducing training time by two-thirds. The proposed tCLD-Net framework is illustrated in Fig. 38.

Meslet, S. et al. [38] introduce SPPNet (Servername Protocol Packet Network), a deep learning-based real-time encrypted traffic classification framework that overcomes feature selection biases in traditional models. Unlike conventional CNN-based architectures that process network flows monolithically, SPPNet modularizes traffic classification by extracting packet-level, protocol-specific, and server-name-based features, ensuring better generalization. The SPPNet model consists of a ResNet18-based CNN for packet content feature extraction, a GRU-based sequence learner for server-name analysis, and an embedding-based protocol and transport classifier, whose outputs are con-

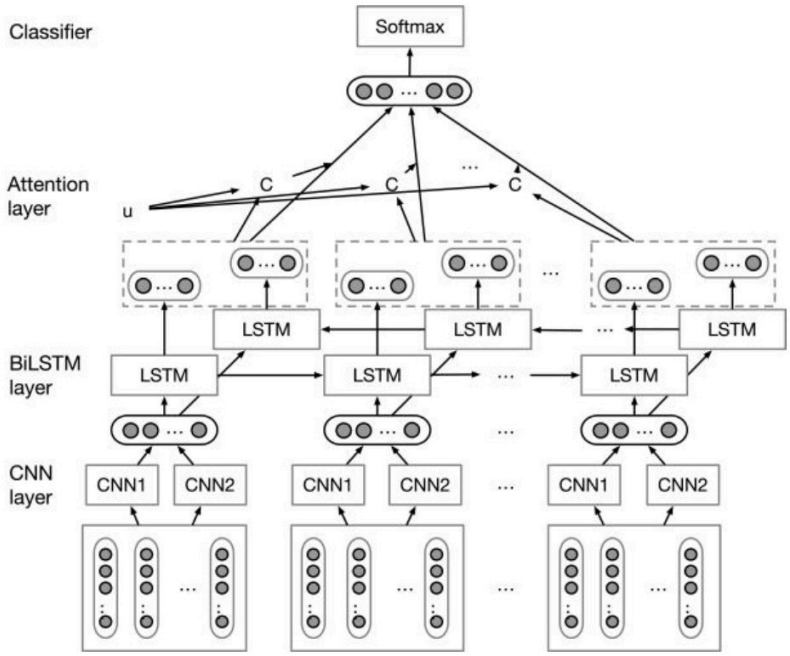

Figure 37: AEFETA framework incorporating CNNs, BiLSTM, and attention mechanisms for encrypted traffic classification, proposed by [36].

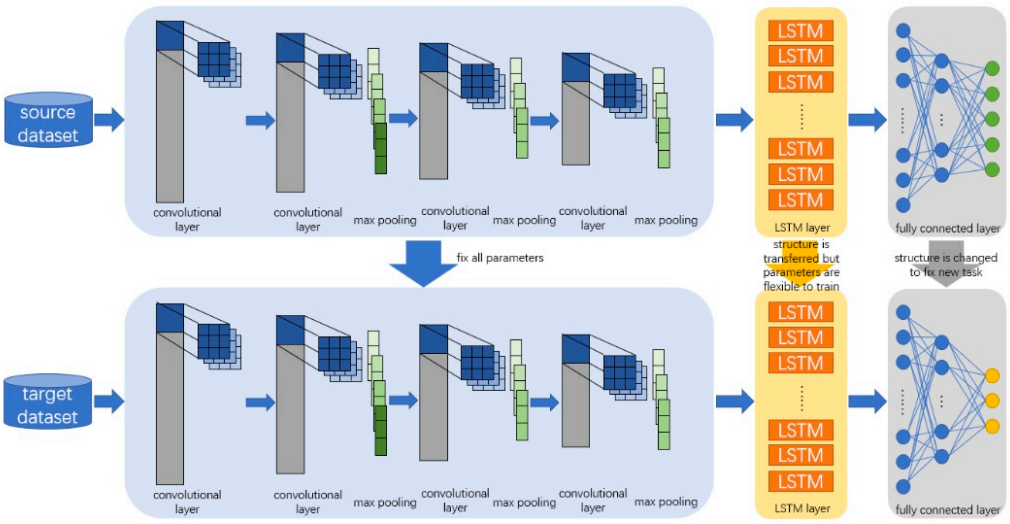

Figure 38: tCLD-Net framework incorporating CNNs, LSTMs, and transfer learning for encrypted traffic classification, proposed by [37].

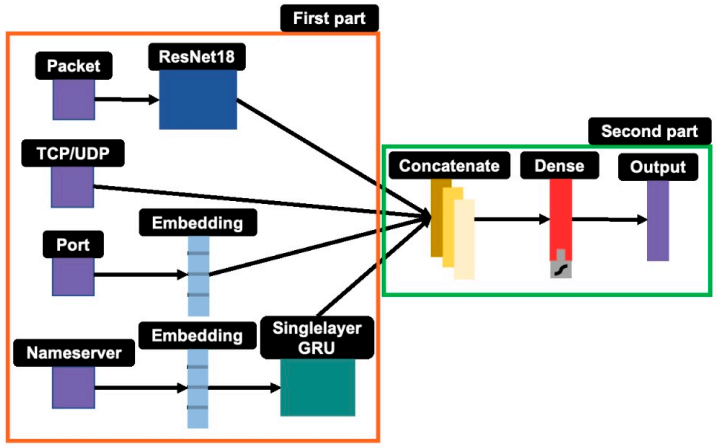

Figure 39: SPPNet architecture integrating modular deep learning for encrypted traffic classification, proposed by [38].

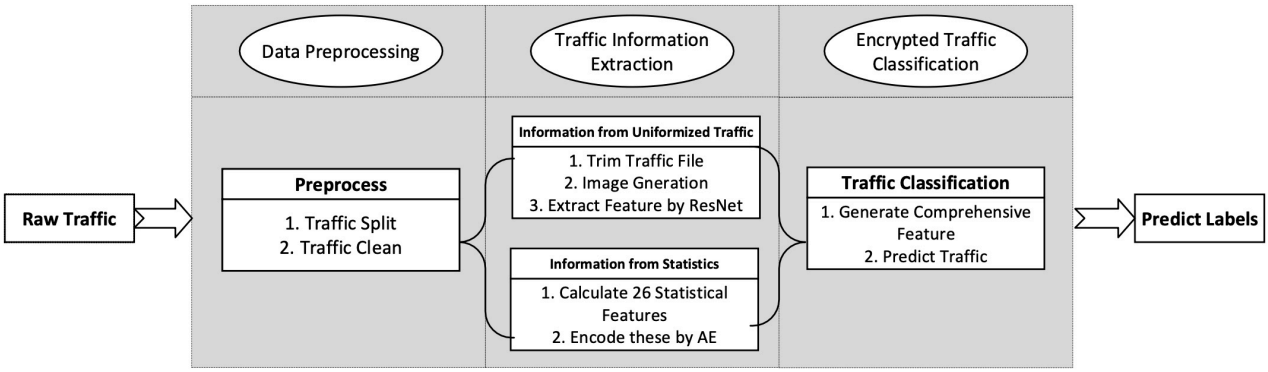

Figure 40: CENTIME framework integrating ResNet and AutoEncoder for encrypted traffic classification, proposed by [39].

catenated for final classification. Experimental evaluations on the ISCX VPN-nonVPN and ISCXTor2016 datasets demonstrate that SPPNet achieves up to 85.7% accuracy on unknown applications, outperforming conventional deep learning models in generalization ability. The SPPNet architecture is illustrated in Fig. 39.

Maonan, Z. et al. [39] introduce CENTIME, an incremental learning-based encrypted traffic classification framework that leverages ResNet for deep feature extraction and AutoEncoder for statistical feature compression, addressing the challenges of feature loss in traditional ETC models. Unlike conventional raw traffic-based methods requiring fixed inputs, CENTIME compensates for information loss caused by traffic truncation by integrating statistical session features, such as flow duration and packet intervals. The hybrid feature representation improves classification accuracy by preserving local packet and global session-level information. The model removes pooling layers and applies 1D-CNNs to enhance raw traffic processing efficiency. Evaluations on the ISCX VPN-nonVPN dataset demonstrate that CENTIME achieves 99.7% accuracy and 99.8% F1-score, outperforming existing deep learning-based classification models. Fig. 40 illustrates the CENTIME framework architecture.

Ma, H. et al. [40] introduce EETC, an Extended Encrypted Traffic Classification framework that enhances real-time traffic classification using a variant of ResNet18 with incremental learning. Unlike traditional deep learning models that require retraining from scratch when new traffic classes emerge, EETC implements an incremental learning mechanism that updates classification capabilities without catastrophic forgetting. The model employs multi-label

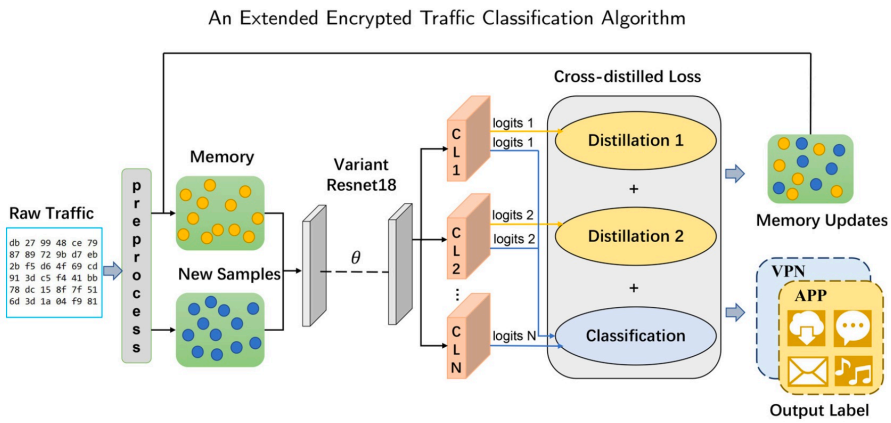

Figure 41: EETC framework integrating variant ResNet18 and incremental learning for encrypted traffic classification, proposed by [40].

classification, enabling simultaneous identification of VPN tunnels and application types. Furthermore, EETC leverages a bias vector and balance factor to mitigate classification biases, ensuring improved generalization across dynamic traffic patterns. Experimental evaluations of the ISCXVPN2016 dataset demonstrate that EETC achieves 98.1% classification accuracy, outperforming existing deep learning-based ETC models. The proposed EETC framework is illustrated in Fig. 41.

Hu, C. et al. [41] introduce CBD, a deep-learning-based framework for encrypted traffic classification, which integrates Convolutional Neural Networks (CNNs) with Bidirectional Encoder Representations from Transformers (BERT). Unlike conventional deep learning models that require manual feature engineering, CBD leverages a pre-training method on unlabeled data, enabling it to generalize well across multiple datasets. The CNN module captures low-level spatial traffic features, while the BERT module processes packet sequences to learn contextual traffic dependencies, significantly improving classification accuracy. Additionally, CBD employs a two-stage pre-training strategy: (1) Identifying ciphertext packets using entropy-based classification and (2) Recognizing sequential flow structures to enhance flow-level learning. Experimental evaluations of the ISCX VPN-nonVPN dataset demonstrate that CBD achieves up to 91% classification accuracy, outperforming conventional deep learning models such as CNN-LSTM and CLD-Net. The architectural framework of the CBD model is illustrated in Fig. 42.

Zhu, Y. et al. [42] propose BCFNet, an encrypted traffic classification framework that integrates multi-scale feature fusion using a combination of ET-BERT and one-dimensional CNN (1D-CNN). Unlike traditional deep learning approaches that rely solely on CNNs for feature extraction, BCFNet employs a dual-stream architecture: (1) A CNN-based module that utilizes an Inception-inspired 1D-CNN structure to capture local spatial patterns in packet sequences, and (2) A transformer-based module (ET-BERT) that models long-range dependencies and contextual relationships between packets using a self-attention mechanism. This design ensures fine-grained and high-level feature extraction, improving classification accuracy and adaptability to dynamic encrypted traffic patterns. Experimental evaluations on the ISCX VPN-nonVPN dataset demonstrate that BCFNet achieves an accuracy of 98.88%, outperforming baseline methods such as Deep Packet, LSTM-based models, and classical CNN architectures. The BCFNet architecture and its multi-scale feature fusion process are illustrated in Fig. 43.

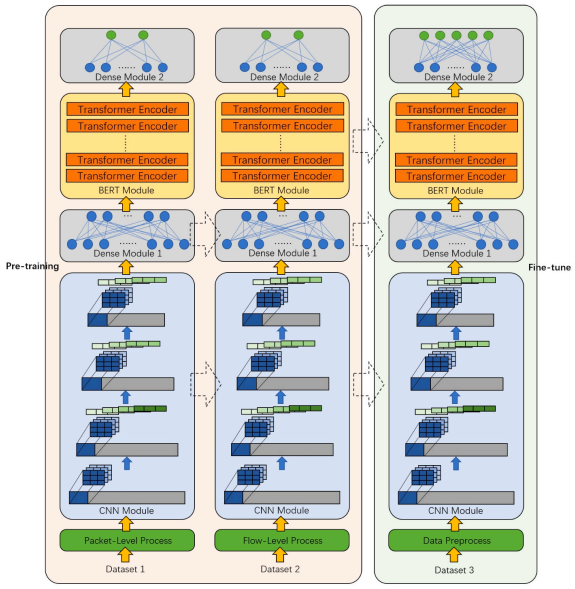

Figure 42: CBD framework integrating CNNs and BERT for encrypted traffic classification, proposed by [41].

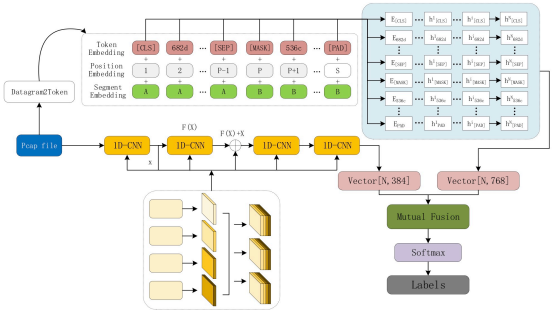

Figure 43: BCFNet framework integrating ET-BERT and Inception-based 1D-CNN for encrypted traffic classification, proposed by [42].

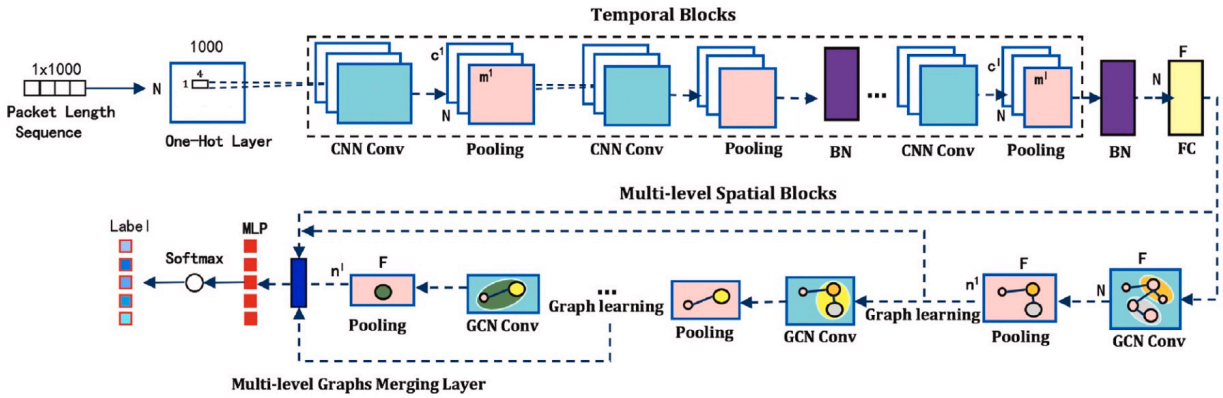

Figure 44: EC-GCN framework integrating multi-scale graph convolution networks for encrypted traffic classification, proposed by [43].

These innovative approaches set new benchmarks in encrypted traffic classification (ETC), offering significant advancements in accuracy and feature representation, thus setting new benchmarks in ETC.

2.2.2.2.2 Graph-Based Neural Networks Using Graph-Based Models in network security offers a nuanced and powerful approach to analyzing encrypted traffic. These models provide an advanced framework for detecting and classifying cyber threats by harnessing the intricate structures and relationships within network data[107][108]. This section delves into recent innovations that leverage graph-based techniques to enhance the precision and robustness of ETC[108].

Several groundbreaking approaches have been introduced in the innovative realm of graph-based models for ETC, each contributing unique insights into network traffic analysis.

In the work by Diao, Z. et al. [43], authors introduce EC-GCN, a graph-based encrypted traffic classification framework that employs multi-scale Graph Convolution Networks (GCNs) to capture spatial-temporal traffic patterns effectively. Unlike traditional sequence-based models, EC-GCN converts encrypted traffic flows into structured graph representations, allowing it to extract local flow correlations and global dependencies. The model features a lightweight encoding layer to map packet sequences into multi-level graph structures, graph pooling mechanisms to adjust representations dynamically, and a structure learning layer to adapt graph edges based on packet dependencies. This architecture ensures resilience to packet loss, reordering, and noise, making it particularly effective in unstable network environments. Experimental evaluations on three real-world datasets, including ISCX-Tor, demonstrate that EC-GCN achieves up to a 20% accuracy improvement over state-of-the-art encrypted traffic classification models. The architecture of the EC-GCN framework is illustrated in Fig. 44.

Hong, Y. et al. [44] introduce MalDiscovery, a graph-based encrypted malicious traffic detection model that integrates GraphSAGE with multi-view feature fusion. Unlike traditional machine learning approaches that rely on either vectorized features or image-based representations, MalDiscovery constructs an attributed KNN graph where encrypted sessions serve as nodes, with edges established based on feature similarity. This allows the model to exploit correlation structures among encrypted traffic instances, making it robust against adversarial attacks and network fluctuations. The GraphSAGE model is utilized for inductive representation learning, capturing local and high-order relationships within encrypted sessions. Evaluations on the CTU-13 and MCFP datasets demonstrate

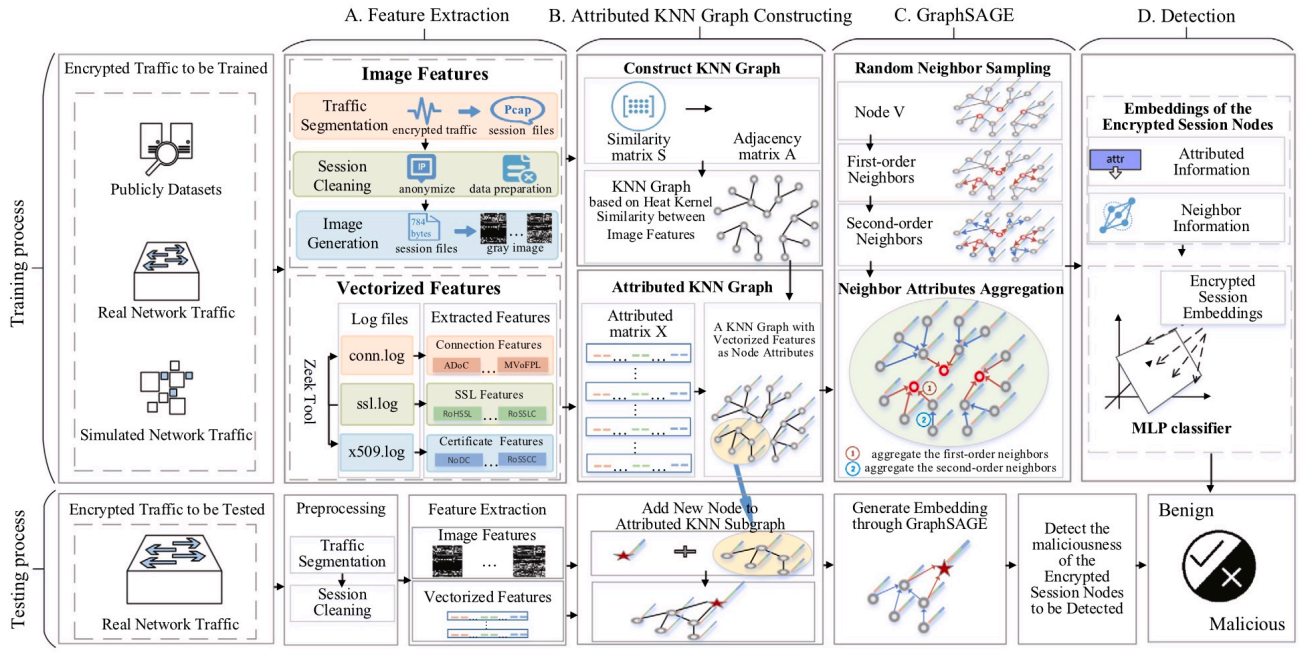

Figure 45: MalDiscovery framework integrating GraphSAGE and multi-view feature fusion for encrypted malicious traffic detection, proposed by [44].

that MalDiscovery achieves 99.9% accuracy, significantly outperforming prior deep learning and graph-based classification methods. The architectural framework of MalDiscovery is illustrated in Fig. 45.

Wang, L. et al. [45] introduce TGPrint, an attack fingerprint classification framework that leverages Graph Convolutional Networks (GCNs) and attention mechanisms to identify encrypted network attacks, including novel attack types. Unlike conventional attack classification approaches that rely on statistical features, TGPrint constructs dynamic attack graphs, where nodes represent attacker-victim hosts and edges encode interaction behaviors over time. The model applies Graph Convolution Attention Networks (GCAN) to extract high-level attack fingerprints, enabling the identification of both known and unseen attacks. A key innovation in TGPrint is the time-window-based attack graph construction, which preserves critical temporal attack patterns while mitigating the impact of noisy traffic. Evaluations on CICIDS and HIKARI-2021 datasets demonstrate that TGPrint achieves 99% precision in attack classification, significantly outperforming traditional machine learning classifiers and deep learning-based models. The TGPrint attack fingerprinting architecture is illustrated in Fig. 46.

Han, X. et al. [46] propose DE-GNN (Dual Embedding with Graph Neural Network), an advanced encrypted traffic classification framework that integrates packet-level and flow-level learning for fine-grained classification. The model introduces a dual embedding strategy, where packet headers and payloads are encoded separately using one-hot encoding and PacketCNN. This ensures parallel processing while reducing interference between different packet structures. The Traffic Interaction Graph (TIG) is then constructed to represent network flow interactions, where Graph Neural Networks (GNNs) extract structural flow-level features, capturing temporal dependencies and hierarchical traffic patterns. The final classification step employs adaptive deep feature fusion, combining header and payload feature embeddings for robust classification performance. Extensive evaluations on the ISCX-VPN2016 and ISCX-Tor2016 datasets demonstrate that DE-GNN outperforms baseline deep learning models, achieving state-of-the-art accuracy in fine-grained traffic classification. The architectural framework of DE-GNN is illustrated

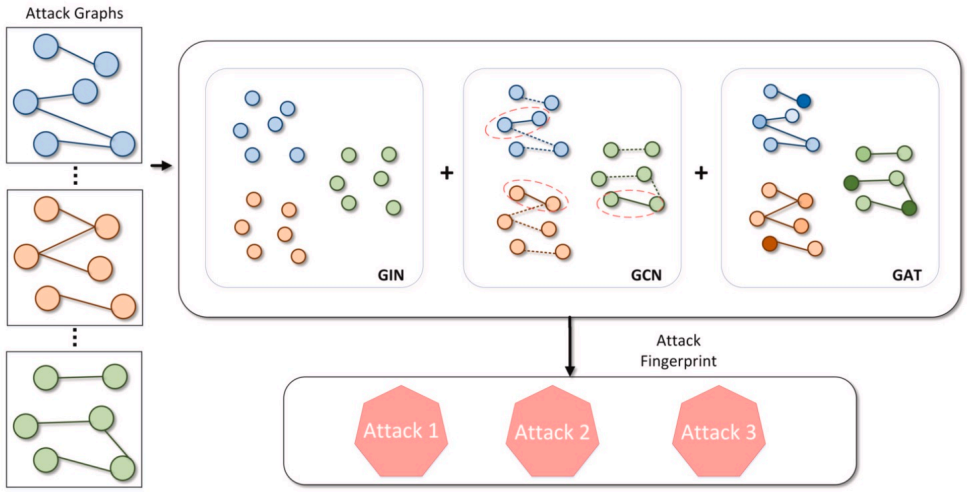

Figure 46: TGPrint framework utilizing Graph Convolution Attention Networks for attack fingerprint classification on encrypted traffic, proposed by [45].

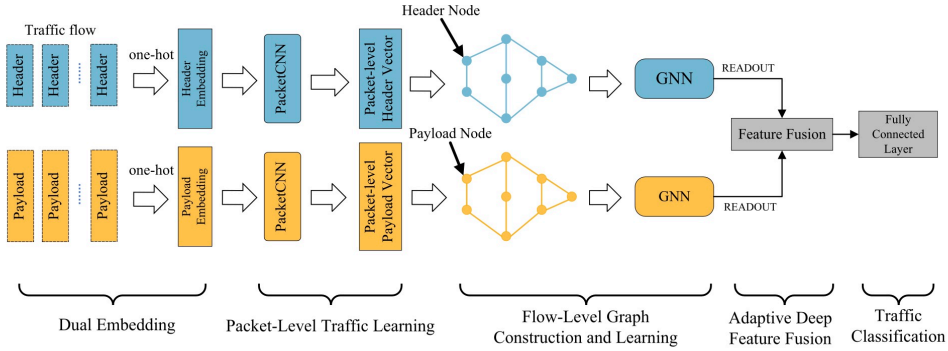

Figure 47: DE-GNN framework integrating dual embedding, PacketCNN, and Graph Neural Networks for encrypted traffic classification, proposed by [46].

in Fig. 47.

Zhang, H. et al. [47] introduce CLE-TFE (Contrastive Learning Enhanced Temporal Fusion Encoder), a multi-task encrypted traffic classification framework designed to mitigate semantic variance between packet- and flow-level representations. Unlike traditional deep learning models that train these tasks independently, CLE-TFE integrates a unified contrastive learning module that jointly optimizes both levels using supervised contrastive learning. The packet-level learning module applies graph-based augmentation on byte sequences, ensuring semantic consistency, while the flow-level module enhances context awareness by leveraging sequential dependencies between packets. Additionally, a cross-level multi-task learning mechanism ensures that packet-level features benefit flow-level classification, reducing redundancy and improving overall model efficiency. Experimental evaluations on the ISCX VPN-nonVPN and ISCX Tor-nonTor datasets demonstrate that CLE-TFE achieves superior classification performance while maintaining a computational overhead 14× lower than ET-BERT. The CLE-TFE architecture and its contrastive learning process are illustrated in Fig. 48.

Yang, J. et al. [48] propose MTSecurity, an advanced malicious encrypted traffic classification framework that combines Graph Neural Networks (GNNs) and Transformers to enhance privacy-preserving detection. Unlike conventional deep learning models that rely solely on statistical or payload-based traffic features, MTSecurity

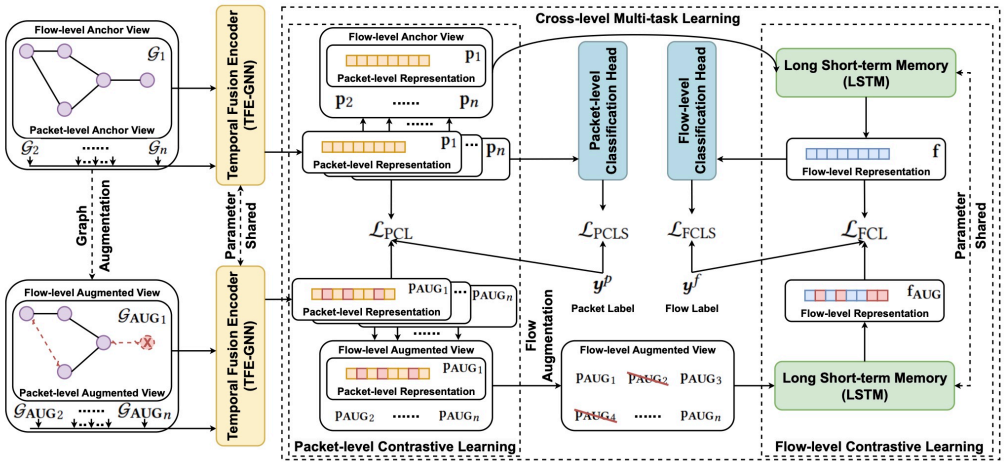

Figure 48: CLE-TFE framework integrating supervised contrastive learning and cross-level multi-task learning for encrypted traffic classification, proposed by [47].

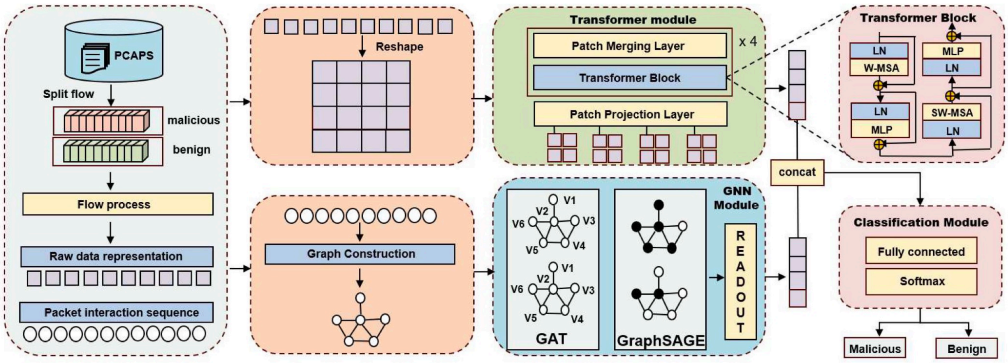

Figure 49: MTSecurity framework integrating Graph Neural Networks and Transformers for encrypted malicious traffic classification, proposed by [48].

introduces a Malicious Traffic Interaction Graph (MTIG) that represents client-server interactions without exposing user-sensitive information. The framework consists of three key components: (1) A Transformer module that processes packet byte sequences to extract semantic representations, (2) A GNN-based feature extractor that models network traffic interactions in graph-structured form, and (3) A classification module that fuses both feature sets for robust malicious traffic detection. Evaluations on the MCFP and USTC-TFC datasets demonstrate that MTSecurity achieves state-of-the-art detection accuracy, with 99.49% accuracy on the MCFP dataset and 99.46% accuracy on the USTC-TFC dataset, significantly outperforming prior deep learning-based detection models. The architectural design of the MTSecurity framework is illustrated in Fig. 49.

Neural networks have demonstrated remarkable proficiency in encrypted traffic classification, with architectures tailored for feature extraction, pattern recognition, and generative learning. However, their computational complexity and black-box nature pose challenges in practical applications. Future research must focus on enhancing their interpretability and efficiency to enable real-time, scalable deployment in network security frameworks.

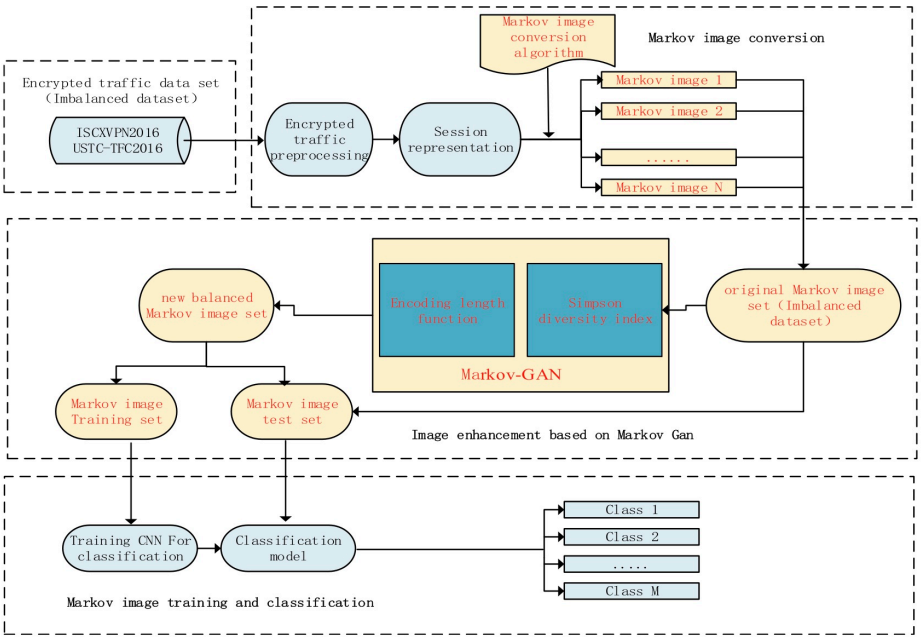

Figure 50: Markov-GAN framework integrating Markov image transformation and generative augmentation for encrypted traffic classification, proposed by [49].

2.2.2.3 Generative Adversarial Networks (GANs) and Transformers

The category of Generative Adversarial Networks (GANs) and Transformers presents an avant-garde approach to ETC, leveraging the generative capabilities of GANs and the sophisticated processing power of transformer models[109]. These methodologies offer innovative solutions to longstanding challenges in network security, such as class imbalance, data augmentation, and feature extraction. This section showcases the notable advancements in employing GANs and transformers to analyze encrypted network traffic[110][109].

Several pioneering approaches have emerged in the domain of Generative Adversarial Networks (GANs) for analyzing encrypted traffic.

In the work by Tang, Z. et al. [49], authors introduce Markov-GAN, a malicious encrypted traffic classification framework that transforms raw encrypted network traffic into Markov images to facilitate deep learning-based classification. Unlike conventional approaches that rely on byte-sequence or statistical features, Markov-GAN employs a three-phase pipeline: (1) Transforming network traffic into Markov images, which encode packet transition probabilities into structured grayscale representations; (2) Enhancing dataset diversity with a GAN-based augmentation technique, which generates high-quality, balanced Markov images to overcome class imbalance issues; and (3) Performing classification with a CNN model, which leverages the structured Markov image features for high-accuracy encrypted traffic classification. Markov-GAN achieves significant improvements in generalization and classification performance by applying a compression-based diversity maximization loss function and Simpson index-based regularization. Evaluations on multiple encrypted traffic datasets, including ISCXVPN2016 and USTC-TFC2016, demonstrate that Markov-GAN achieves over 99% accuracy, surpassing traditional machine learning and deep learning models. The architectural workflow of Markov-GAN is illustrated in Fig. 50.

Wang, P. et al. [50] propose PacketCGAN, a Conditional Generative Adversarial Network (CGAN)-based approach

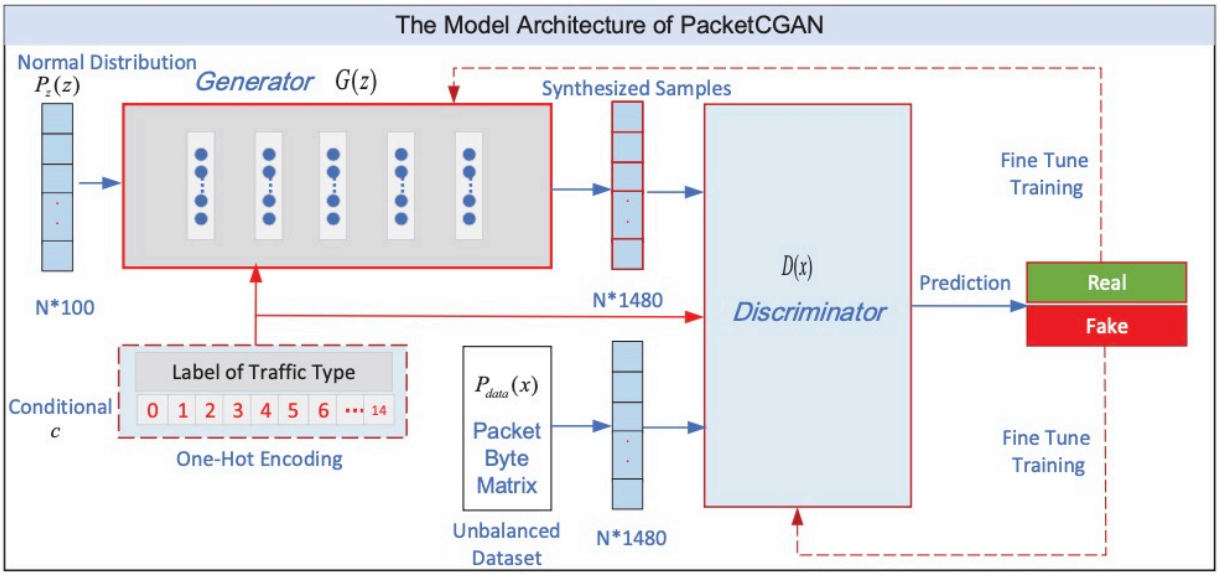

Figure 51: PacketCGAN framework utilizing Conditional Generative Adversarial Networks for class-balancing in encrypted traffic classification, proposed by [50].

designed to address class imbalance in encrypted traffic datasets. Unlike traditional oversampling methods such as ROS (Random Oversampling) and SMOTE (Synthetic Minority Oversampling Technique), which may introduce redundancy or synthetic noise, PacketCGAN learns the underlying distributions of minority-class traffic and generates realistic synthetic samples. The PacketCGAN architecture consists of a generator synthesizing new traffic samples conditioned on the class labels and a discriminator ensuring the generated traffic maintains realistic statistical properties. By incorporating packet byte matrices as input and leveraging multi-layer perceptrons (MLPs) in both generator and discriminator, PacketCGAN effectively enhances dataset diversity while maintaining classification accuracy. Evaluations on the ISCXVPN2016 and USTC-TFC2016 datasets demonstrate that PacketCGAN-augmented datasets improve CNN-based classifier performance, achieving an accuracy of 99.51%, outperforming ROS, SMOTE, and vanilla GAN-based methods. The PacketCGAN architecture and its synthetic data generation process are illustrated in Fig. 51.

Sanjalawe, Y. et al. [51] introduce MTSecurity, a hybrid encrypted traffic classification framework that leverages Bidirectional Generative Adversarial Networks (BiGANs) for data augmentation and Vision Transformers (ViTs) for feature extraction and classification. Unlike traditional CNN or RNN-based models, MTSecurity enhances detection capabilities by transforming encrypted traffic flows into structured feature representations while addressing class imbalance through BiGAN-generated synthetic data. The model operates in four phases: (1) Dataset preprocessing, which removes redundant features and normalizes network traffic data; (2) Data augmentation, where BiGAN generates high-quality synthetic traffic samples to balance minority classes; (3) Feature extraction, where ViT encodes multi-level traffic dependencies using a self-attention mechanism; and (4) Obfuscated Tor traffic detection, which classifies Tor and non-Tor flows using ViT-enhanced embeddings. Experimental evaluations on the ISCX-Tor2016 dataset show that MTSecurity achieves 99.59% accuracy, 99.83% recall, and 99.72% precision, surpassing conventional deep learning models. The MTSecurity BiGAN-ViT workflow is illustrated in Fig. 52.

Wang, P. et al. [52] propose ByteSGAN, a semi-supervised Generative Adversarial Network (GAN) framework

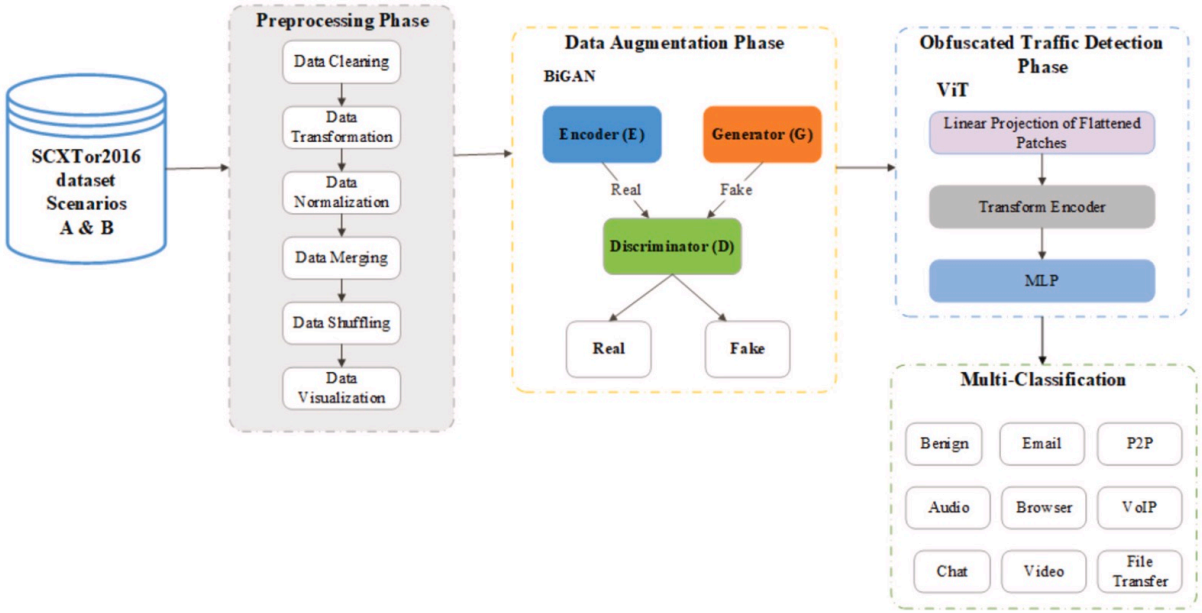

Figure 52: MTSecurity framework integrating Bidirectional Generative Adversarial Networks (BiGANs) and Vision Transformers (ViTs) for obfuscated Tor traffic classification, proposed by [51].

designed for encrypted traffic classification in SDN Edge Gateways. Unlike traditional deep learning models that require large-scale labeled datasets, ByteSGAN leverages semi-supervised learning to enhance classification accuracy using limited labeled traffic data while exploiting abundant unlabeled samples. The framework modifies the GAN discriminator network to act as a multi-class classifier, allowing it to distinguish between real and generated traffic while performing classification tasks simultaneously. ByteSGAN operates in two phases: (1) Traffic data preprocessing, where raw packets are normalized into Packet Byte Vectors (PBV) and structured for model input, and (2) GAN-based classification, where the generator learns feature distributions while the discriminator refines classification boundaries through adversarial training. Evaluations on ISCX VPN-nonVPN 2012 and Crossmarket datasets show that ByteSGAN outperforms CNN-based supervised methods, achieving an F1-score of 99.06%. The workflow of ByteSGAN is illustrated in Fig. 53.

Zhao, R. et al. [53] introduce MT-FlowFormer, a semi-supervised transformer-based framework designed to enhance encrypted traffic classification (ETC) in environments with limited labeled data. Unlike traditional deep learning models that struggle with flow sequence correlations, MT-FlowFormer leverages a novel transformer-based encoder, FlowFormer, to extract multi-level dependencies across flow sequences. The Mean Teacher-style semi-supervised learning framework further refines classification accuracy by incorporating consistency regularization, ensuring that labeled and unlabeled data contribute to model training. The approach integrates spatiotemporal data augmentation, which enhances the model's ability to generalize traffic characteristics by introducing controlled variations in flow sequences. Experimental evaluations on the SJTU-AN21 and ISCXVPN2016 datasets demonstrate that MT-FlowFormer achieves 84.7% classification accuracy with only 10% labeled data, outperforming conventional CNN-based and LSTM-based classifiers. The architectural workflow of MT-FlowFormer is illustrated in Fig. 54.

Lin, X. et al. [54] introduce ET-BERT, a transformer-based pre-training model for encrypted traffic classification, designed to capture contextualized datagram representations from large-scale unlabeled encrypted traffic. Unlike traditional statistical and deep learning-based models that struggle with generalization, ET-BERT employs a two-

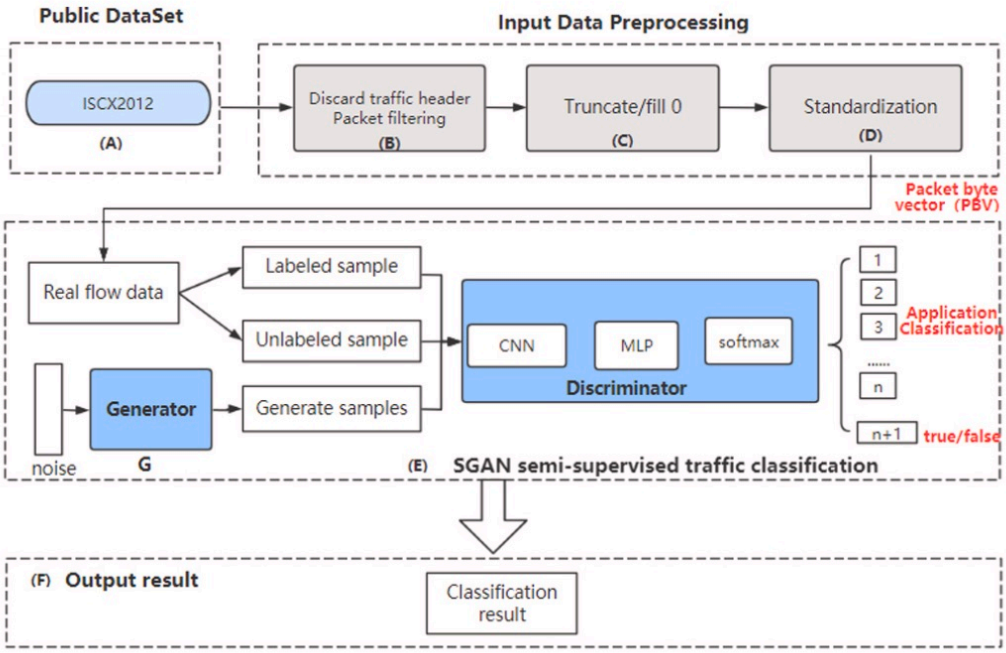

Figure 53: ByteSGAN framework integrating semi-supervised Generative Adversarial Networks (GANs) for encrypted traffic classification in SDN Edge Gateways, proposed by [52].

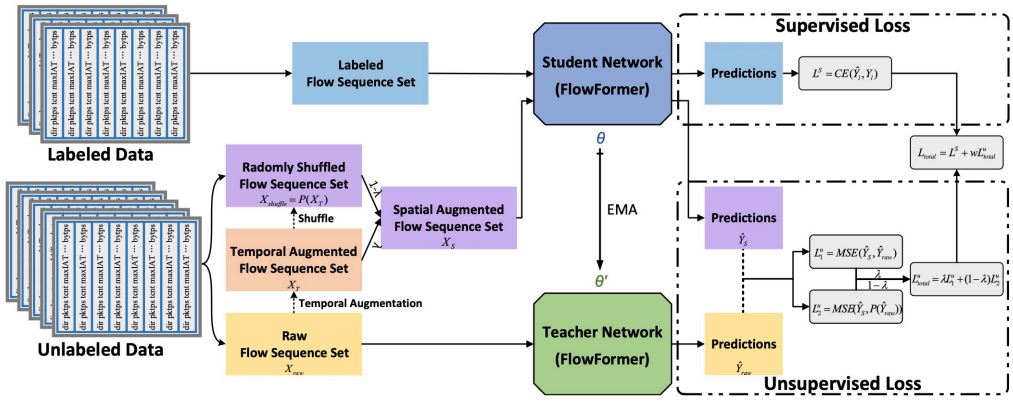

Figure 54: MT-FlowFormer framework integrating transformers and semi-supervised learning for encrypted traffic classification, proposed by [53].

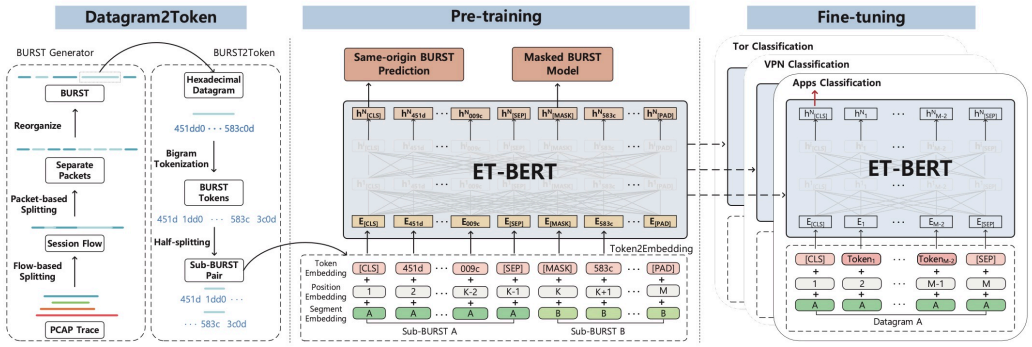

Figure 55: ET-BERT framework integrating transformers for contextualized datagram representation in encrypted traffic classification, proposed by [54].

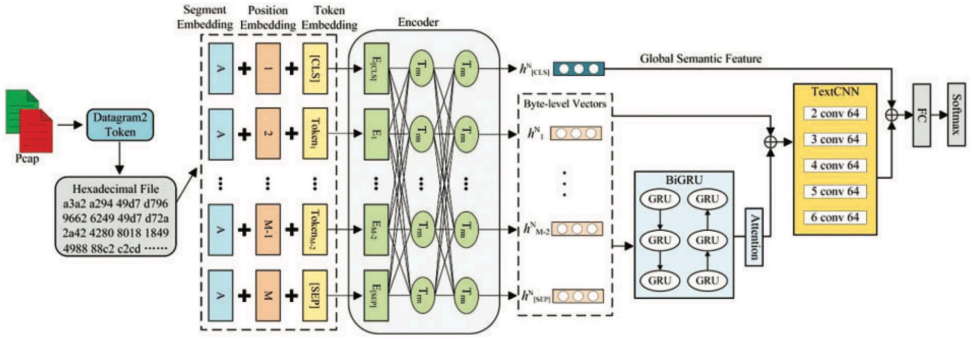

Figure 56: BSTFNet framework integrating global semantic features from transformers and local spatiotemporal features from BiGRU and TextCNN for encrypted malicious traffic classification, proposed by [55].

stage training process: (1) Pre-training, where the model learns generic traffic representations through self-supervised learning on large-scale encrypted data, and (2) Fine-tuning, where task-specific classification is performed with minimal labeled data. The framework introduces two novel pre-training tasks: the Masked BURST Model (MBM), which learns correlations between datagram bytes, and the Same-origin BURST Prediction (SBP), which models transmission dependencies within encrypted sessions. Evaluations across five datasets, including ISCX-VPN-Service and CSTNET-TLS 1.3, demonstrate that ET-BERT achieves state-of-the-art performance, with up to 98.9% F1-score on VPN classification tasks, surpassing previous models by 5.2% to 10.0%. The architectural workflow of ET-BERT is illustrated in Fig. 55.

Huang, H. et al. [55] introduce BSTFNet, a hybrid deep learning framework designed to classify encrypted malicious traffic by integrating global semantic and local spatiotemporal features. Unlike traditional models that rely on either global or local features independently, BSTFNet combines Bidirectional Encoder Representations from Transformers (BERT) to extract packet-level semantic features, Bidirectional Gated Recurrent Units (BiGRU) to capture temporal dependencies, and TextCNN to learn spatial representations from packet bytes. The model first applies BERT's multi-layer attention mechanism to derive high-dimensional contextual embeddings, which BiGRU then processes to enhance temporal relationships. Finally, multi-scale convolutional kernels in TextCNN extract fine-grained local patterns, allowing for a comprehensive encrypted traffic characterization. Evaluations on the USTC-TFC2016 dataset demonstrate that BSTFNet achieves 99.39% accuracy and 99.40% F1-score, outperforming prior models such as CNN-LSTM and ET-BERT. The BSTFNet architecture and feature fusion mechanism are illustrated in Fig. 56.

Park, J. et al. [56] introduce a DistilBERT-based multi-task learning (MTL) framework for fast and accurate

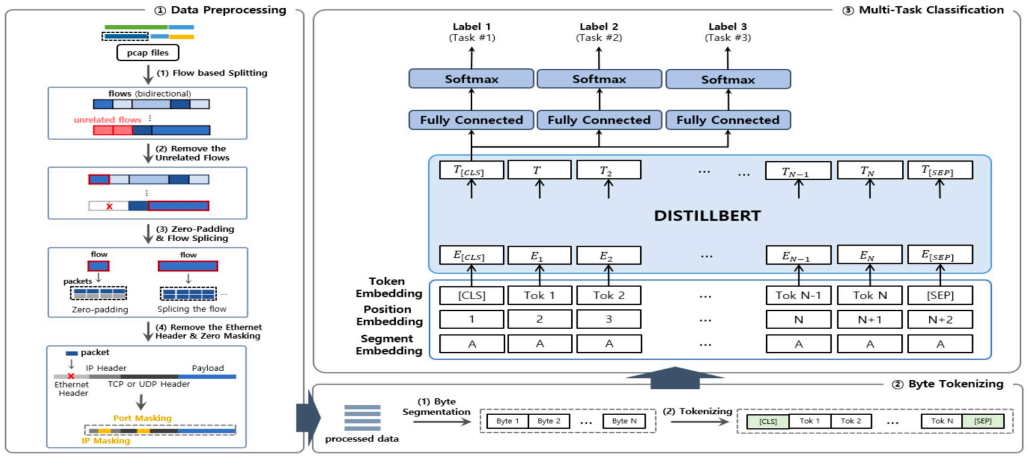

Figure 57: DistilBERT-based multi-task learning framework for encrypted traffic classification, proposed by [56].

encrypted traffic classification, addressing computational efficiency while maintaining classification precision. Unlike traditional single-task classification models, their approach performs encapsulation, category, and application classification simultaneously, reducing the overhead associated with separate models. The model first tokenizes byte sequences from encrypted traffic flows, then leverages DistilBERT's transformer architecture to extract deep semantic representations from the tokenized sequences. The multi-task loss function ensures balanced training across tasks, mitigating data imbalance issues while enhancing generalization across diverse encrypted traffic patterns. Evaluations on the ISCX 2016 VPN-nonVPN dataset demonstrate that this approach achieves 99.29%, 97.38%, and 96.89% accuracy for encapsulation, category, and application classification, respectively, outperforming conventional models while significantly reducing processing time. The architectural workflow of the DistilBERT-based multi-task classification model is illustrated in Fig. 57.

Exploring deep learning in encrypted traffic analysis reveals its profound impact on feature representation and classification performance. However, despite these advancements, challenges such as computational cost, data dependency, and model interpretability persist. Future research must focus on optimizing DL architectures and integrating explainability techniques to enhance their practicality in real-world deployments.

2.2.3 Hybrid Models

The Hybrid Model Approaches in ETC demonstrate the synergy between ML and DL and their integration with traditional methodologies. They offer robust, versatile, and innovative solutions to network security challenges[111]. This category underscores the strength of combining different analytical techniques to enhance classification accuracy, adaptability, and computational efficiency[112].

In the work by Marim, M.C. et al. [113], authors investigate the effectiveness of classical machine learning models, particularly Decision Trees (DTs), Random Forest (RF), and Multilayer Perceptrons (MLPs), for darknet traffic classification using the CIC-Darknet2020 dataset. Unlike contemporary deep learning-based approaches that rely on high-dimensional feature representations, this study emphasizes interpretable models, demonstrating that traditional ML algorithms can achieve competitive performance with significantly lower computational costs. The authors utilize Recursive Feature Elimination (RFE) to identify the most relevant traffic attributes, refine the feature set, and enhance classification efficiency. Experimental evaluations reveal that Decision Trees achieve 99.89% accuracy for

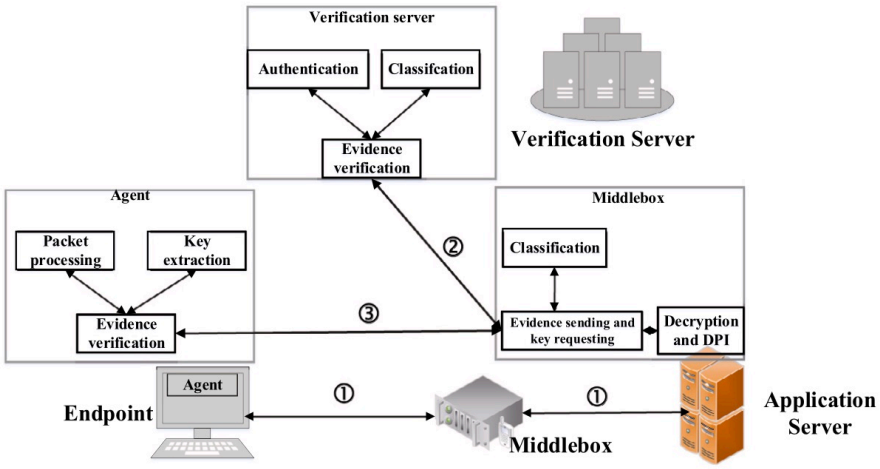

Figure 58: ME-Box framework integrating machine learning and evidence verification for encrypted traffic detection, proposed by [57].

traffic source classification and 98.62% accuracy for application categorization, surpassing prior deep learning-based darknet traffic classifiers by up to 13

Xu, B. et al. [57] introduce ME-Box, a hybrid encrypted traffic detection framework that integrates machine learning with evidence verification mechanisms to enhance security while preserving user privacy. Unlike conventional deep learning-based methods that rely solely on classifier outputs, ME-Box introduces a middlebox-driven architecture where encrypted traffic is first classified using machine learning techniques and then subjected to an evidence verification process before potential decryption. The system incorporates a multi-stage trust evaluation where suspected malicious flows must meet strict verification criteria before session-key decryption is authorized. This approach mitigates the risk of false positives and adversarial attacks while ensuring that only validated malicious flows undergo deep packet inspection (DPI). Evaluations on real-world malware traffic datasets demonstrate that ME-Box achieves high detection accuracy without modifying cryptographic protocols, ensuring compatibility with modern encrypted network infrastructures. Fig. 58 illustrates the ME-Box system architecture and verification workflow.

Hu, Y. et al. [58] introduce a hierarchical hybrid classifier to analyze user behavior across multiple darknet environments, including Tor, I2P, ZeroNet, and Freenet. Unlike conventional darknet traffic classifiers focusing solely on identifying encrypted traffic, this model refines classification to the user activity level, distinguishing between browsing, chat, file transfer, streaming, and VoIP behaviors. The framework employs a three-layer classification structure, where the first layer differentiates darknet vs. regular traffic, the second layer classifies the darknet type, and the final layer predicts specific user behaviors. The system integrates six machine learning classifiers (LR, DT, RF, GBDT, XGBoost, LightGBM) and two deep learning classifiers (MLP, LSTM), ultimately selecting XGBoost as the best-performing model for darknet identification. Evaluations on a custom-labeled darknet dataset demonstrate that this approach achieves 96.9% accuracy in darknet type classification and 91.6% accuracy in user behavior classification, significantly improving upon traditional flat classification methods. Fig. 59 illustrates the hierarchical classification architecture.

Rust-Nguyen, N. et al. [59] investigate the resilience of machine learning (ML) and deep learning (DL) models against adversarial attacks in darknet traffic classification, emphasizing the necessity of robust cybersecurity models.

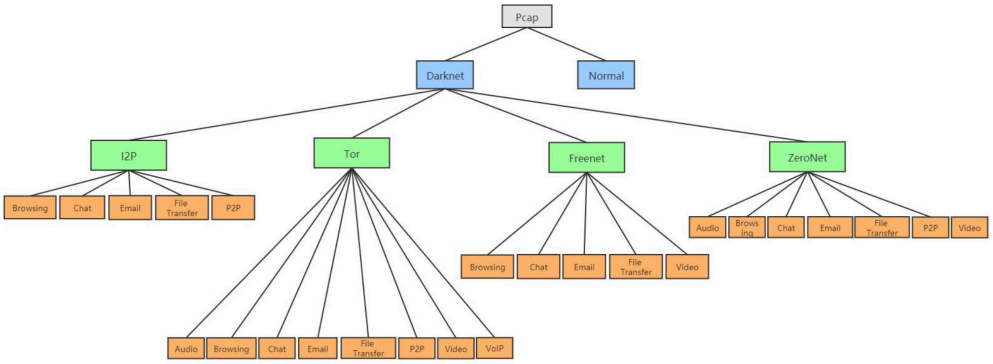

Figure 59: Hierarchical hybrid classifier integrating machine learning and deep learning for darknet user behavior analysis, proposed by [58].

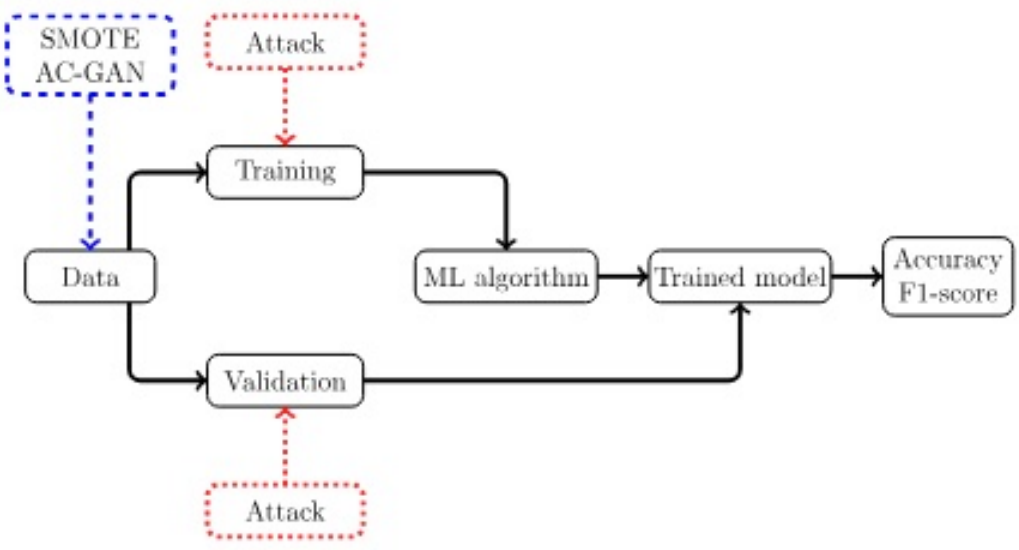

Figure 60: Experimental setup for darknet traffic classification and adversarial attack evaluation, proposed by [59].

Unlike prior work focusing solely on classification accuracy, this study analyzes how adversarial obfuscation can systematically degrade ML/DL model performance and proposes countermeasures to mitigate these attacks. The authors evaluate a broad range of classifiers, including Random Forest (RF), Convolutional Neural Networks (CNNs), and Auxiliary-Classifier Generative Adversarial Networks (AC-GANs), using the CIC-Darknet2020 dataset. Their findings reveal that Random Forest outperforms other ML models under standard conditions, achieving 99.8% F1-score for darknet traffic detection. However, when adversarial obfuscation is applied, classifier accuracy drops significantly, demonstrating the vulnerability of ML-based traffic classifiers to adversarial manipulations. The study proposes data augmentation techniques such as SMOTE and GAN-based synthetic sample generation to improve model robustness. The experimental workflow for darknet traffic classification and adversarial evaluation is illustrated in Fig. 60.

Malekghaini, N. et al. [60] propose AutoML4ETC, a Neural Architecture Search (NAS)-based framework that automates the design of efficient and high-performing deep learning architectures for encrypted traffic classification (ETC). Unlike manually crafted deep learning models, which require constant hyperparameter tuning and architecture modifications, AutoML4ETC dynamically searches for optimal architectures, ensuring adaptability to

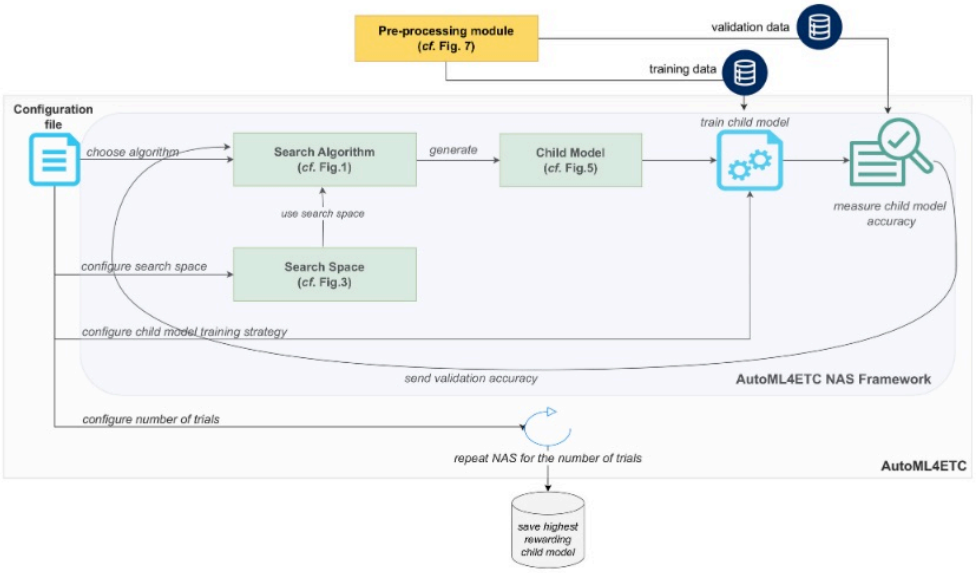

Figure 61: AutoML4ETC framework integrating Neural Architecture Search (NAS) for automated encrypted traffic classification, proposed by [60].

evolving network traffic distributions. The framework introduces a novel search space tailored for early encrypted traffic classification, where the first few packets in a flow are used to make accurate predictions. AutoML4ETC employs different search strategies, including Reinforcement Learning (RL), Monte Carlo Tree Search (MCTS), and Evolutionary Algorithms (EA), to generate optimal neural architectures. Evaluations on public benchmark datasets (ISCX VPN-nonVPN, USTC-TFC2016) and real-world TLS/QUIC traffic from the Orange mobile network demonstrate that AutoML4ETC outperforms state-of-the-art classifiers, achieving higher accuracy with significantly fewer parameters. The AutoML4ETC architecture and search process are illustrated in Fig. 61.

Elmaghraby, R. T. et al. [61] explore advanced ML and DL techniques for encrypted network traffic classification, focusing on distinctive feature extraction to enhance network management and security. Unlike traditional payload inspection methods, this study employs machine learning models that rely solely on metadata attributes such as packet length, timestamps, and transport layer security (TLS) information to classify encrypted traffic. The authors propose three classification approaches: (1) A hybrid model integrating neural networks with conventional classifiers such as Random Forest (RF) and Support Vector Machines (SVM), (2) A bidirectional LSTM model coupled with ensemble learning techniques, and (3) A CNN-LSTM hybrid model designed for session-based classification. Through comparative evaluations, the first approach achieves 96.8% classification accuracy, outperforming conventional ML models, while the second and third techniques demonstrate improved time complexity with slightly reduced accuracy. The proposed model's flowchart and its feature extraction process are illustrated in Fig. 62.

Luo, P. et al. [62] introduce an ML-based encrypted traffic classification framework, integrating BITization and an optimal sliding window technique to enhance traffic pattern recognition. Unlike conventional ML models that rely on manually crafted statistical features, this framework automates feature extraction using a binary encoding mechanism (BITization), which converts byte sequences into structured numerical representations, thereby improving classifier interpretability and performance. The optimal sliding window method further refines feature selection by identifying the most informative byte sequences for classification. Empirical evaluations on ISCX VPN-Service, Cross-Platform-Android, and USTC-TFC2016 datasets demonstrate that BITization improves classical ML models such as SVM,

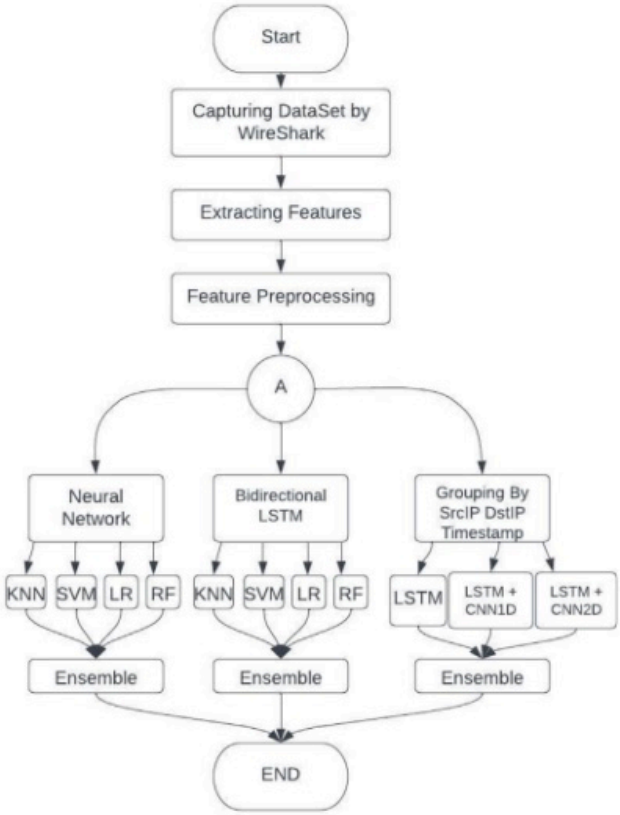

Figure 62: ML and DL-based encrypted traffic classification model utilizing feature extraction techniques, proposed by [61].

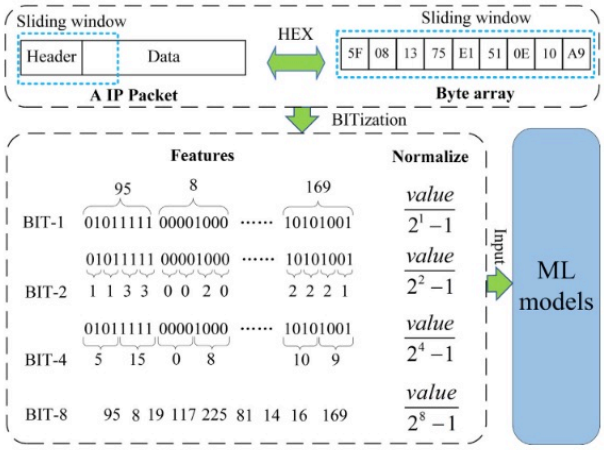

Figure 63: BITization-based encrypted traffic classification framework integrating optimal sliding window technique, proposed by [62].

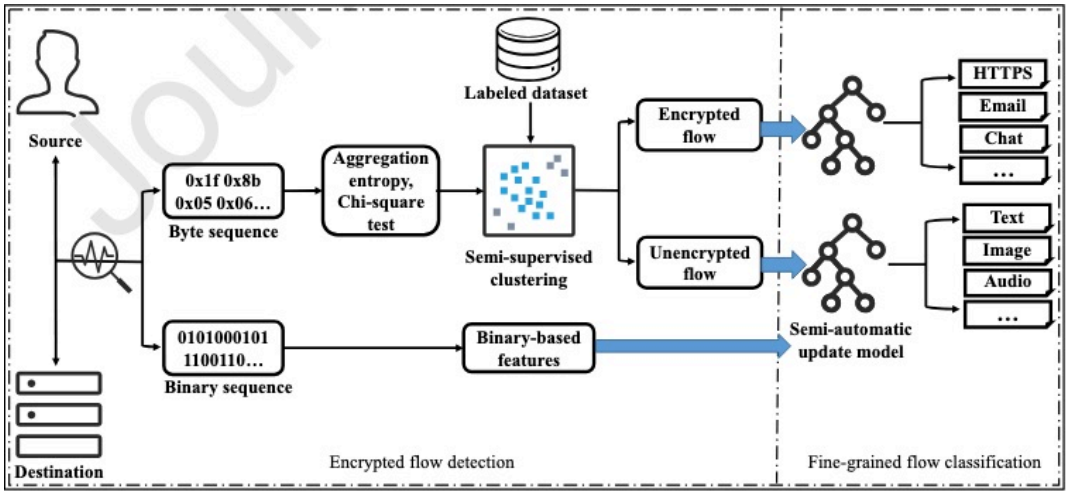

Figure 64: HETC framework utilizing payload features and a Random Forest model for high-speed encrypted traffic classification, proposed by [63].

RF, and LightGBM by up to 23.7% in accuracy, significantly narrowing the performance gap between traditional ML and deep learning classifiers. The BITization-based encrypted traffic classification workflow is illustrated in Fig. 63.

Yan, X. et al. [63] introduce HETC (High-speed Encrypted Traffic Classification), a payload feature-based framework for efficient traffic classification in high-speed network environments. Unlike traditional traffic classifiers that rely on full-flow statistics or deep learning-based feature extraction, HETC focuses on randomly sampled short flows and utilizes aggregation entropy and chi-square test features to detect encrypted vs. unencrypted flows. The second stage of HETC introduces binary payload features and employs a Random Forest (RF) classifier for fine-grained classification of encrypted traffic types, including HTTPS, VoIP, and email services. The model achieves 94% F-measure in detecting encrypted flows and 85%-93% in classifying fine-grained encrypted traffic, outperforming baseline methods while maintaining a processing time of only 2-16 ms per flow, making it well-suited for high-speed applications. The HETC classification framework is illustrated in Fig. 64.

Zhao, J. et al. [64] introduce METAROCKETC, an adaptive encrypted traffic classification framework designed to adapt to evolving network environments and diverse ETC tasks rapidly. Unlike traditional models that require

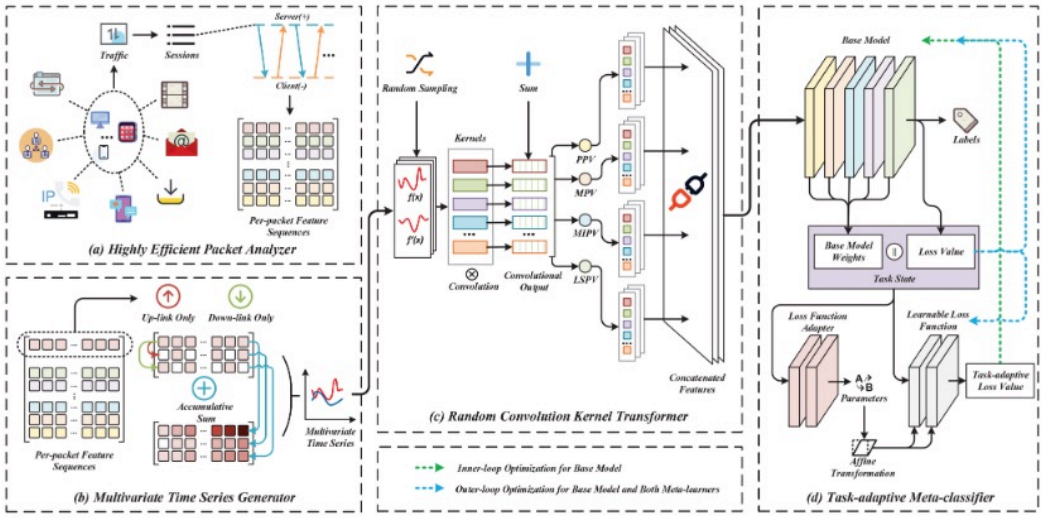

Figure 65: METAROCKETC framework integrating time series analysis and meta-learning for adaptive encrypted traffic classification, proposed by [64].

extensive retraining for new network conditions, METAROCKETC employs protocol-agnostic feature extraction, modeling packet length sequences as multivariate time series. The framework integrates random convolution kernel transformations to enhance classification robustness, capturing discriminatory behavioral patterns across channels without extensive feature engineering. The model is further optimized through Model-Agnostic Meta-Learning (MAML), allowing fast adaptation to new encrypted traffic types with minimal labeled data. Evaluations on ISCX-Tor, CTU, BetterNet-HTTPS, and 5GAD datasets demonstrate that METAROCKETC significantly outperforms baseline deep learning models in across-task and few-shot learning scenarios, achieving state-of-the-art accuracy with reduced computational costs. The METAROCKETC architecture and its adaptive classification workflow are illustrated in Fig. 65.

Li, X. et al. [65] introduce MISS (Multi-View Sequence Fusion for Incremental Learning), a novel framework designed to enable encrypted traffic classification (ETC) models to evolve efficiently without requiring full retraining. Unlike conventional models that degrade as new applications emerge, MISS employs a multi-view sequence fusion strategy to extract cross-view information from packet sequences, inter-arrival times, and TLS metadata, ensuring a comprehensive feature representation for long-term adaptability. The framework introduces a representative exemplar selection algorithm to retain only the most informative samples while mitigating redundant storage consumption. Additionally, MISS integrates plastic and stable model branches to balance learning new applications while preserving previous classification knowledge. Evaluations on PURE-TLS, Cross-Platform, and USTC-TFC-2016 datasets demonstrate that MISS outperforms state-of-the-art incremental learning methods, achieving 11.37% higher classification accuracy and 1.58% better adaptation performance. The incremental learning workflow of MISS is illustrated in Fig. 66.

Wang, R. et al. [66] propose a contrastive learning-based encrypted traffic classification framework, integrating spatial-temporal feature fusion to improve representation consistency. Unlike conventional classification methods that treat spatial and temporal features separately, this framework leverages a dual encoder architecture, where a 2D-CNN extracts spatial packet features while a 1D-CNN and LSTM model temporal dependencies. The extracted feature representations are aligned within a shared embedding space using contrastive loss, ensuring feature consistency and robustness. The model is trained using InfoNCE loss, where positive pairs consist of the same

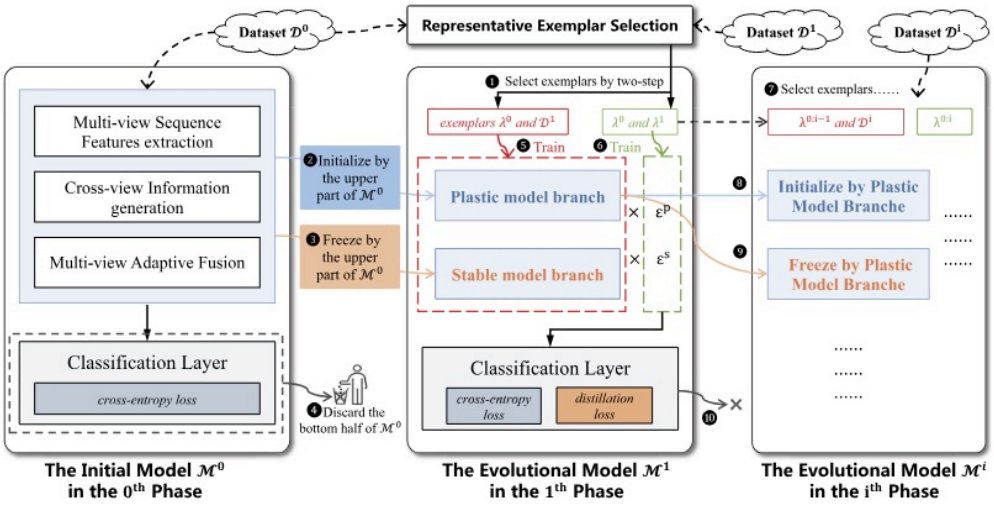

Figure 66: MISS framework integrating multi-view sequence fusion and incremental learning for encrypted traffic classification, proposed by [65].

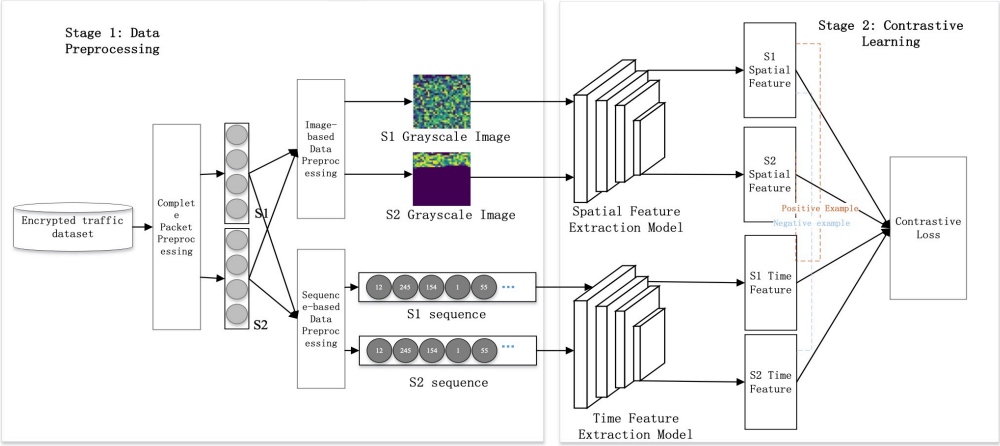

Figure 67: Contrastive learning framework integrating spatial-temporal feature fusion for encrypted traffic classification, proposed by [66].

traffic sample's spatial and temporal features and negative pairs are drawn from other samples to enhance feature distinctiveness. Evaluations on the ISCX VPN2016 dataset demonstrate that the proposed approach outperforms existing CNN, TSCRNN, and ET-BERT models, achieving 99.9% accuracy for binary VPN classification and 98.2% for multi-service classification. Fig. 67 illustrates the spatial-temporal contrastive learning framework.

Wang, G. et al. [67] introduce a multi-task scenario encrypted traffic classification framework, integrating Parameter-Efficient Fine-Tuning (PEFT) to enhance computational efficiency and interpretability. Unlike traditional deep learning-based classification methods that require full model retraining, this framework reduces the fine-tuning parameter set while preserving high classification accuracy. The model optimizes network resource usage by leveraging Low-Rank Adaptation (LoRA), enabling rapid adaptation across different classification tasks such as VPN service recognition, malware detection, and intrusion classification. The study validates that fine-tuning only a small subset of parameters achieves performance comparable to fully retrained models while significantly reducing computational overhead. Experiments on ISCX-VPN, USTC-TFC, and CIC-IDS datasets demonstrate that the proposed multi-task fine-tuning approach outperforms conventional full-parameter models by improving scalability

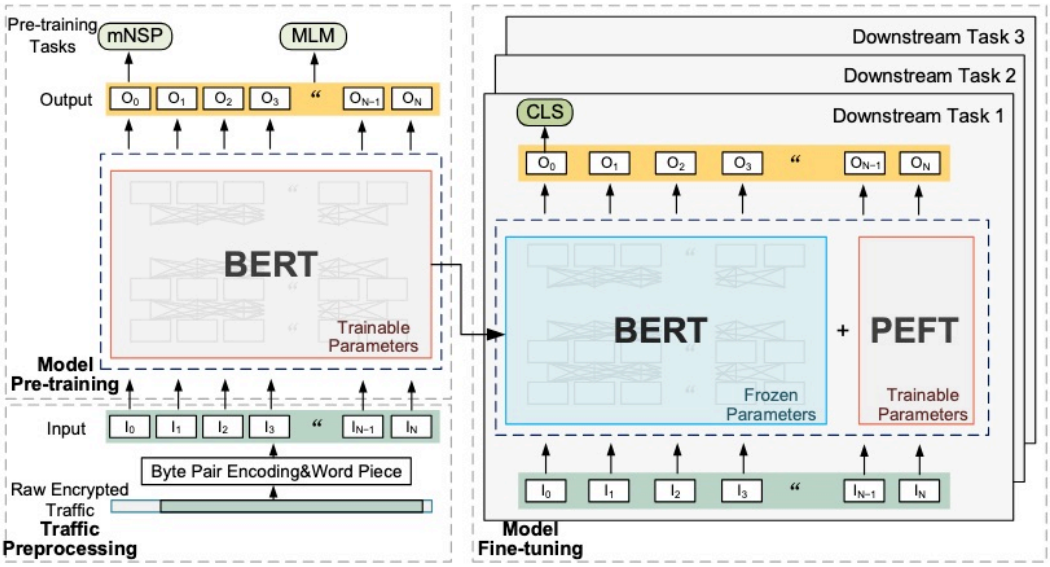

Figure 68: Multi-task scenario encrypted traffic classification framework utilizing Parameter-Efficient Fine-Tuning (PEFT), proposed by [67].

and deployment efficiency. Fig. 68 illustrates the multi-task classification and fine-tuning workflow.

Integrating hybrid models has significantly advanced encrypted traffic classification, offering superior performance compared to standalone ML or DL techniques. By converging ML, DL, and traditional methods, these approaches create powerful tools for addressing the complexities of network security. However, combining multiple techniques introduces challenges related to interpretability and computational overhead. Future research should focus on optimizing hybrid frameworks to maintain high accuracy while enhancing model efficiency and adaptability, ultimately paving the way for more adaptive and forward-thinking solutions in the ever-evolving field of cybersecurity. The review of previous works illustrates the rapid advancement of encrypted traffic classification methodologies, with each approach contributing unique insights into feature extraction, classification accuracy, and model interpretability. While traditional machine learning techniques remain foundational, deep learning and hybrid approaches have introduced significant improvements in automation and adaptability. However, challenges such as scalability, generalization, and computational efficiency remain open for further exploration, guiding the next research phase in this domain.

2.3 Synthesis

Classifying encrypted traffic is a crucial yet complex challenge in the dynamic field of network security. As we delve into the multifaceted issues of computational efficiency and resource demands, this section explores the inherent limitations and potential drawbacks of current models/techniques utilized in this domain.

We focus on high computational complexity and the resource-intensive nature of preprocessing and model architecture. These factors significantly determine the practicality of deploying these models in real-time or resource-constrained environments. The limitations include but are not limited to the dependency on diverse and quality data, the scalability and complexity of models, the necessity for continuous adaptation and learning, and the feasibility of real-time processing.

Every element is essential in understanding the broader implications of ETC systems and their viability in contem-

porary networked environments. The following bullet points outline these limitations, each supported by specific references that, to some extent, illuminate the discussed shortcomings in the order of importance.

1. **Real-Time Application Feasibility** - Achieving real-time encrypted traffic classification remains a challenge due to the high computational overhead of existing models [33, 34, 44, 58, 38, 52, 40, 18, 39].

2. **High Computational Complexity** - Many ETC models exhibit high computational demands due to the complexity of their underlying algorithms, making them less suitable for real-time applications or environments with limited processing power[26, 18, 50, 28, 48, 47, 45, 114, 24, 37, 55].

3. **Scalability Concerns Due to Advanced Features** - The inclusion of advanced features, while beneficial for accuracy, introduces scalability issues, mainly when applied to high-throughput networks[115, 53].

4. **Adaptation Challenges in Evolving Network Conditions** - The dynamic nature of modern networks demands models that can adapt to frequent changes, a capability that many current solutions lack[64, 59].

5. **Impact of Data Quality and Diversity on Model Effectiveness** - Models heavily rely on the quality and diversity of training data. Insufficient or imbalanced datasets can lead to poor generalization and reduced effectiveness across varying network conditions[67, 63, 21, 19, 35, 27, 113, 51, 44, 60].

6. **Resource-Intensive Preprocessing and Model Architecture** - Extensive preprocessing and complex architectures often require significant computational resources, posing challenges for practical deployment in real-time or resource-constrained environments[56, 21, 41].

7. **Challenges in Data Collection and Representation** - Gathering representative and diverse datasets is inherently tricky in ETC due to encryption and privacy constraints, leading to challenges in building robust models[54, 25, 15, 30].

8. **Complex Model Integration and Scalability Issues** - Integrating advanced models with existing systems often requires significant effort. At the same time, scalability remains a pressing challenge for handling large-scale or diverse traffic scenarios [22, 61, 116, 31, 45, 117, 104, 32].

9. **Need for Ongoing Model Updates** - Continuous updates are required to keep models effective in the face of evolving traffic patterns and encryption techniques, which adds to operational complexity[65, 49, 23, 104, 118, 16, 29].

10. **Practical Deployment Issues in Dynamic Networks** - Deploying ETC models in dynamic and heterogeneous network environments introduces numerous challenges, including maintaining performance and adaptability across diverse scenarios [43, 49, 20].

The synthesis of encrypted traffic classification methodologies underscores the need for continued innovation in feature engineering, model interpretability, and computational efficiency. While recent advancements in deep learning and hybrid models have enhanced classification capabilities, challenges such as adversarial robustness, real-time processing, and explainability remain at the forefront of research. Future work should strive to develop scalable, interpretable, and high-performance models that can effectively address the evolving challenges in network security.

This study of the drawbacks and limitations in the models of ETC highlights critical challenges that impact the practical deployment of sophisticated models. High computational complexity and resource-intensive processes hinder real-time applications and limit the scalability necessary for broader network implementations. By analyzing these limitations, this thesis sets the stage for further discussion on the need for model innovation and optimization. The insights gathered here underscore the urgency for developing more efficient techniques that do not compromise performance while ensuring adaptability and minimal resource utilization. Our thesis aims to cover the top **five** challenges based on our model.

2.4 Concluding Remarks

This chapter systematically reviewed the landscape of encrypted traffic classification by evaluating encryption protocols and critically analyzing machine learning, deep learning, and hybrid approaches. A key insight drawn from the literature is that while ML models provide computational efficiency, DL techniques offer improved accuracy at the cost of interpretability and resource demand. Hybrid models were found to offer a balanced compromise, combining the interpretability of ML and the representational depth of DL. Importantly, the literature also reflects a growing interest in integrating explainable AI (XAI) techniques to improve trust and transparency in classification decisions. These insights directly shaped our motivation to propose a hybrid attention-LightGBM architecture enhanced with SHAP and LIME-based interpretability.

Building upon the limitations identified in this chapter, the subsequent **Chapter 3 -Proposed Model** introduces our proposed model, designed to address critical deficiencies in existing approaches. This model aims to enhance classification accuracy, improve feature representation, and optimize computational efficiency while maintaining interpretability and scalability. The forthcoming discussion will detail the architectural design, methodology, and technical innovations underpinning our proposed framework.

3 Proposed Model

This chapter outlines the methodology of the proposed model, detailing its core components and their role in advancing encrypted traffic classification. The model is designed to address key challenges in detecting and characterizing encrypted traffic by integrating systematic data collection, feature extraction, augmentation, and explainable AI-driven classification. The approach ensures robustness, scalability, and interpretability in real-world network environments. The methodology is structured around four main components:

1. **A Comprehensive Survey on Encrypted Traffic Information**
2. **Development of a Network Traffic Analyzer**
3. **Utilize Augmented Datasets for Robust Model Training**
4. **Design and Implementation of the XAI-Powered Detection and Characterization Model**

Each component plays a crucial role in addressing the challenges posed by encrypted network traffic and advancing the field of cybersecurity. Through this comprehensive model methodology, the thesis will contribute valuable tools to the cybersecurity field and enhance the understanding and management of encrypted network traffic. This approach promises to yield significant insights and improvements in network security protocols and practices.

The following subsections provide a detailed technical breakdown of the **Motivation**, **Proposed Solution**, and **Architecture Design**. The **Motivation** subsection establishes the rationale behind this research, identifying key challenges in encrypted traffic detection and the need for a more effective classification framework. The **Proposed Solution** subsection outlines the foundational components, including the comprehensive literature survey, network traffic analyzer development, dataset augmentation, and the design of the XAI-powered detection and characterization model. The **Architecture Design** subsection then details the implementation specifics, covering feature extraction using NTLFlowLyzer, data preprocessing, feature selection, classification framework integration, and explainability techniques. Together, these components form a comprehensive encrypted traffic detection and classification methodology.

3.1 Motivation

The exponential growth of encrypted network traffic has fundamentally transformed the digital landscape, improving user privacy while posing significant challenges to network security and monitoring. Encryption, while crucial for safeguarding sensitive information, has become a double-edged sword, providing a secure medium for malicious actors to bypass traditional security measures. This dual nature of encryption has created a pressing need for advanced methodologies capable of detecting, analyzing, and classifying encrypted traffic without compromising user privacy.

Existing approaches in encrypted traffic analysis often fall short in either scalability or accuracy when faced with real-world challenges. While effective in specific scenarios, traditional machine learning models frequently struggle with the evolving complexity of encryption techniques. Similarly, deep learning models, though promising, often require extensive computational resources and lack interpretability—making their deployment in resource-constrained

environments difficult. Furthermore, publicly available datasets for encrypted traffic are usually insufficiently diverse, leading to a lack of generalizability in existing models.

To address these challenges, this thesis is motivated by the need for a comprehensive, modular, and scalable framework for encrypted traffic detection and classification. This work aims to achieve high accuracy, scalability, and interpretability by combining advanced feature extraction mechanisms, such as attention layers, with hybrid machine learning models. This methodology targets the accurate classification of encrypted and non-encrypted traffic. It extends to multi-class classification for finer categorization of encrypted traffic into specific categories such as VPN, Tor, and others.

This thesis aims to bridge the gap between privacy preservation and effective security monitoring by developing an open-source network traffic analyzer and an advanced detection model. These tools are designed to meet real-world challenges, including scalability, adaptability to emerging encryption techniques, and maintaining interpretability through explainable AI methods like SHAP and LIME. This work aspires to set a new benchmark in encrypted traffic analysis, contributing significantly to cybersecurity.

3.2 Proposed Solution

The proposed solution addresses encrypted traffic detection and classification challenges through a structured, multi-component approach. Each component contributes to developing an efficient and scalable model capable of analyzing encrypted traffic with high accuracy and interoperability.

3.2.1 Comprehensive Literature Survey

The first step in designing the methodology involved conducting a **comprehensive literature survey** to analyze state-of-the-art techniques, datasets, and tools in **encrypted traffic analysis**. This survey examined various scholarly articles, technical reports, and existing surveys, focusing on machine learning (ML), deep learning (DL), and hybrid models used for encrypted traffic detection and classification. The review provided a critical synthesis of the strengths and weaknesses of various models, serving as a foundation for designing an advanced analytical framework in this thesis.

In terms of **modeling approaches**, the survey highlighted that while deep learning models, such as convolutional neural networks (CNNs), exhibited strong feature extraction capabilities, they often **lacked interpretability**—a crucial aspect in cybersecurity applications. On the other hand, tree-based ML models, such as **Random Forests**, offered **better transparency and explainability**, yet struggled with generalization when handling complex patterns in encrypted traffic. To bridge these gaps, recent studies have proposed **hybrid techniques** that leverage both paradigms, incorporating **ML for decision-making** and **DL for feature learning**. Among these, attention mechanisms emerged as a promising innovation, refining feature representation by allowing models to focus on the most relevant patterns within encrypted traffic data.

Beyond modeling techniques, the survey also examined **publicly available datasets** and existing network traffic analyzers, identifying several critical challenges. These included **data imbalance issues**, the **limited representation of modern encryption protocols**, and the **lack of standardized preprocessing pipelines**. The absence of a unified framework for handling encrypted traffic across diverse datasets underscored the need for developing a **modular traffic analyzer** that could efficiently **process heterogeneous datasets** while enhancing their representational capacity.

Furthermore, the survey revealed key **research gaps**, such as the **limited scalability of current methodologies** for real-world encrypted traffic classification, the **black-box nature of many advanced models**, and the **inadequate handling of class imbalances**—challenges that hinder broader adoption in cybersecurity applications. These insights were instrumental in shaping the core components of the proposed methodology, ensuring a **data-driven and innovative approach** to encrypted traffic detection and classification. The findings from this literature review were consolidated into a scholarly survey paper, offering a structured overview of current tools and techniques while **outlining future directions** for research in this domain.

3.2.2 Network Traffic Analyzer Development

At the heart of the methodology lies the enhancement/development of a **modular and scalable open-source cybersecurity network traffic analyzer**. This analyzer is designed to efficiently process raw network traffic data, such as PCAP files, and extract meaningful features that effectively represent traffic patterns. Its primary objective is to support a wide range of encryption detection and analysis tasks while maintaining high performance and adaptability to evolving network traffic characteristics.
The analyzer is capable of extracting **over 400 features**, covering various aspects of network traffic, including:

- **Statistical attributes**: Statistical attributes quantify the distribution and variability of network traffic features. Metrics such as mean and variance capture central tendency and dispersion, providing essential insights for distinguishing traffic patterns and enhancing classification accuracy.
  , e.g., mean, variance.

- **Entropy-based metrics**: Entropy-based metrics measure the randomness and unpredictability of network traffic features. Shannon entropy, for instance, quantifies the uncertainty in data distribution, aiding in the detection of encrypted traffic by assessing variability in packet payloads and flow characteristics.
  e.g., Shannon entropy, reflecting data randomness.

- **Time-based characteristics**: Time-based characteristics capture temporal patterns in network traffic, such as inter-packet arrival times, to analyze flow behavior. These metrics help differentiate traffic types by identifying timing patterns associated with encryption protocols and network activity, e.g., inter-packet arrival times, aiding in behavioral analysis.

To ensure flexibility and future scalability, the analyzer is designed with a **modular architecture**, allowing seamless integration of additional feature extraction modules. This adaptability ensures that new encryption techniques and traffic patterns can be accommodated without requiring extensive modifications to the core system. Additionally, the software was developed to process **large-scale datasets efficiently**, maintaining optimal performance and accuracy when handling diverse network traffic types.
By implementing this open-source tool, the methodology provides a robust foundation for encrypted traffic classification, enhancing the ability to analyze and interpret encrypted network flows with precision.

Entropy-Based Network Traffic Analysis Framework

The network traffic analyzer implements several sophisticated entropy-based metrics to capture and quantify the randomness patterns in network flows, which is beneficial for detecting encrypted traffic and identifying different types of encryption. Three primary encoding schemes are utilized—binary, hexadecimal, and UTF-8—each providing

unique insights into data patterns.

Entropy Types and Mathematical Foundations

Entropy measures the randomness and uncertainty within network traffic, serving as a key indicator of encryption patterns. Different entropy types provide insights into data variability, aiding in encrypted traffic classification. The following section delves into the different types of entropy.

Binary Entropy Analysis

Binary entropy analysis processes raw packet data by converting it into binary sequences. For each n-byte window ($n = 4, 8, 16, 32, 64$), multiple statistical measures are calculated to capture variations in entropy distribution. These include:

- For each n-byte window ($n = 4, 8, 16, 32, 64$), calculates multiple statistical measures:

$$\begin{aligned}
\text{Mean Entropy:} \quad & \bar{H} = \frac{1}{N}\sum_{i=1}^{N} H(X_i) \\
\text{Standard Deviation:} \quad & \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(H(X_i) - \bar{H})^2} \\
\text{Skewness:} \quad & \gamma = \frac{1}{N\sigma^3}\sum_{i=1}^{N}(H(X_i) - \bar{H})^3 \\
\text{Mode:} \quad & \text{Most frequently occurring entropy value} \\
\text{Median:} \quad & \text{Central entropy value} \\
\text{Coefficient of Variation:} \quad & CV = \frac{\sigma}{\bar{H}} \times 100
\end{aligned}$$

This statistical breakdown enables a fine-grained entropy assessment, crucial for distinguishing between encrypted and non-encrypted traffic.

Minimum Entropy

Minimum entropy quantifies the degree of determinism in a dataset by measuring the likelihood of the most probable event.

- Implements min-entropy calculation:

$$H_\infty(X) = -\log_2(\max_x P(X = x))$$

- Applied across binary, hexadecimal, and UTF-8 encodings to evaluate randomness within different data representations.
- Particularly effective in detecting deterministic patterns and non-uniform distributions, which may indicate weak encryption schemes or structured data.
- Calculated as:

$$H_\infty = -\log_2(p_{\max})$$

where $p_{\max}$ represents the highest probability among observed values, ensuring sensitivity to concentrated entropy distributions.

Rényi Entropy

Rényi entropy generalizes Shannon entropy by introducing an order parameter (α), allowing for controlled sensitivity to rare or dominant events. Two specific cases are implemented:

- **Maximum Entropy** ($\alpha = 0$):

$$H_0(X) = \log_2(|\{x : P(X = x) > 0\}|)$$

  This measures the adequate size of the support set, indicating how many unique values contribute to the distribution.

- **Collision Entropy** ($\alpha = 2$):

$$H_2(X) = -\log_2\left(\sum_x P(X = x)^2\right)$$

  Collision entropy is particularly sensitive to repeated values. It provides insights into the probability of two randomly drawn elements being identical, making it useful for traffic clustering and encrypted traffic characterization.

This formulation enables a more nuanced entropy-based analysis by capturing both the spread and redundancy of network traffic data.

Implementation Details

The entropy computation framework is designed to efficiently process network traffic data while maintaining scalability and robustness in analyzing encrypted traffic patterns. The implementation leverages **optimized data structures** to facilitate large-scale entropy calculations in real-time, ensuring minimal computational overhead. Additionally, **parallel processing capabilities** enable the framework to handle high-throughput network flows, enhancing its efficiency in dynamic environments. To further refine classification accuracy, the framework incorporates **adaptive threshold mechanisms**, which dynamically adjust entropy-based classification criteria based on observed traffic variations. This structured approach ensures that the system remains both scalable and resilient, effectively adapting to evolving network security challenges is detailed below in steps.

Sliding Window Analysis

Sliding window analysis enables continuous entropy assessment by applying overlapping windows of varying sizes (4, 8, 16, 32, and 64 bytes). This approach facilitates a progressive evaluation of entropy fluctuations, allowing for a more granular understanding of data randomness. The Python implementation follows an iterative approach, where each window extracts sequential segments of the input data:

```
for i in range(0, len(data_bits) - seq_len + 1):
    substrings.append(data_bits[i:i + seq_len])
```

By leveraging overlapping windows, the framework ensures a smooth entropy evolution over time, reducing abrupt variations that may arise from non-contiguous segmentation. Additionally, the integration of **multi-scale entropy extraction** allows for capturing both local and global randomness variations, improving the model's ability to detect

subtle changes in encryption behavior. This enhances the system's adaptability in identifying sudden shifts in entropy patterns, often indicative of encryption anomalies. The structured implementation ensures **real-time responsiveness**, making it highly effective in dynamic network traffic analysis scenarios.

Statistical Processing

Statistical processing plays a crucial role in maintaining the stability and interpretability of entropy measures, ensuring meaningful analysis across diverse network traffic scenarios. To achieve this, the implementation incorporates **robust error handling** mechanisms that prevent computational failures in edge cases. Specifically, it addresses scenarios such as empty data streams by preventing division-by-zero errors, handles single-value sequences to ensure valid entropy estimates, and mitigates inefficiencies when the available data is insufficient for the selected window size.

Additionally, **normalization of entropy values** is applied to enable comparative analysis across different datasets. This ensures that entropy-based insights remain consistent and interpretable, regardless of variations in traffic patterns. The following Python snippet demonstrates both normalization and error handling, preventing invalid computations when processing entropy values:

```
try:
    mean_entropy = sum(entropy_values) / len(entropy_values)
except ZeroDivisionError:
    mean_entropy = 0.0
```

By integrating these statistical processing techniques, the framework enhances the reliability of entropy-based network traffic analysis, ensuring robustness and accuracy even in complex, real-world network environments.

Multiple Encoding Perspectives

The entropy framework integrates diverse encoding techniques to analyze randomness from multiple perspectives, ensuring a comprehensive assessment of encrypted traffic characteristics. By leveraging different encoding formats, the model enhances its ability to detect patterns and anomalies in network traffic.

Binary processing captures randomness at the most granular level, converting data into an 8-bit binary representation, allowing for fine-grained entropy computation. The encoding process follows the implementation shown below:

```
encoded_data = ''.join(['{:08b}'.format(x) for x in data])
```

For structured protocol analysis and byte-level pattern recognition, **hexadecimal processing** is applied. This representation aids in extracting protocol-level insights by converting data into a hex-encoded format:

```
encoded_data = str(data)
```

Additionally, **UTF-8 processing** is utilized to detect encoding-based anomalies and obfuscated text patterns, often present in encrypted or manipulated traffic. This encoding approach converts data into an ASCII-readable hex format:

```
encoded_data = binascii.hexlify(data).decode('ascii')
```

By analyzing entropy across these encoding formats, the framework significantly improves its ability to classify encrypted traffic with higher accuracy, ensuring a robust and adaptable approach to network traffic analysis.

Feature Engineering Significance

Feature engineering is critical in enhancing the analytical capabilities of entropy-based classification, ensuring that the extracted features effectively capture meaningful patterns in network traffic. One of the key aspects of this process is **multi-scale analysis**, which examines entropy at different levels to identify variations in encryption behavior. Smaller window sizes (4, 8 bytes) detect fine-grained structural patterns within traffic flows. In comparison, larger window sizes (32, 64 bytes) provide insights into broader encryption trends, allowing for a comprehensive characterization of network entropy.

Additionally, **statistical robustness** ensures that entropy measures remain consistent across diverse traffic conditions, minimizing the impact of noise and irregularities in the dataset. This guarantees that the entropy features retain their interpretability and reliability across varying encryption protocols. Furthermore, the model's **encryption detection capabilities** allow protocol-specific entropy variations to be highlighted, aiding in the classification of encrypted traffic by distinguishing between different cryptographic implementations.

These engineered entropy features serve as valuable inputs for machine learning models, enabling more precise and scalable encrypted traffic classification. By leveraging multi-scale analysis, statistical refinement, and encryption-aware feature extraction, the framework enhances its ability to distinguish encrypted traffic patterns accurately.

Implementation Efficiency

The framework optimizes computational resources while maintaining high-performance encrypted traffic analysis. To achieve this, **efficient memory management** is implemented through streaming network packets and windowed entropy calculations, significantly reducing storage overhead. Additionally, **computational optimization** techniques, such as single-pass entropy computation and the reuse of intermediate results, enhance processing efficiency without compromising accuracy. The framework also incorporates **robust error handling** mechanisms to ensure stable performance across varying network conditions, implementing graceful degradation strategies with meaningful default values when necessary. These optimizations collectively enable the system to handle large-scale traffic analysis with minimal resource consumption.

This entropy-based framework provides a comprehensive, multi-perspective approach to network traffic analysis by integrating several key methodologies. It leverages **multiple entropy types**, including Shannon, Min, and Rényi entropy, to perform fine-grained randomness assessments, capturing variations in data distributions. Furthermore, it incorporates **diverse encoding perspectives**, utilizing binary, hexadecimal, and UTF-8 representations to enhance interpretability and improve the detection of encrypted traffic anomalies. The implementation of **sliding window techniques** ensures continuous entropy profiling, allowing real-time assessment of randomness patterns in network flows. Additionally, **statistical robustness** is maintained to stabilize entropy measures, ensuring consistency across different network environments.

Combining these elements, the framework delivers an advanced encrypted traffic detection mechanism capable of precise classification, anomaly detection, and protocol identification. This multi-faceted approach enhances the system's adaptability, making it well-suited for modern network security challenges.

3.2.3 Augmented Dataset Creation

A key aspect of the proposed methodology is the creation and utilization of an **augmented dataset** that strategically combines the strengths of two comprehensive datasets: **Darknet-Dataset-2020** and **CIC-Darknet2020**. The deliberate fusion of these datasets ensures a robust and diverse representation of encrypted traffic, enabling the model to tackle a broad spectrum of challenges in network traffic classification.

The **CIC-Darknet2020** dataset offers a focused analysis of darknet traffic, particularly within anonymity-preserving networks such as **Tor** and **I2P**. This dataset is highly valuable for its emphasis on capturing the complexities of encrypted communication in underrepresented domains, making it a critical component for advancing research in network forensics and cybersecurity. Its inclusion ensures that the augmented dataset comprehensively represents encrypted traffic patterns from networks designed to preserve user anonymity.

In contrast, the **Darknet-Dataset-2020** captures a diverse range of user activities across multiple darknet platforms, including **Tor, I2P, ZeroNet, and Freenet**. This dataset encompasses browsing, streaming, and file transfer behaviors, offering a rich representation of behavior-specific traffic patterns within darknet environments. The addition of this dataset ensures that the model can analyze diverse user behaviors and adapt to various real-world encrypted traffic scenarios.

Combining these datasets is motivated by their **complementary strengths**. While CIC-Darknet2020 focuses on the technical nuances of encrypted traffic within anonymity-preserving networks, Darknet-Dataset-2020 provides a broader perspective on user behavior across multiple platforms. The resulting augmented dataset bridges these perspectives, ensuring that the model is trained on a dataset that represents both **network-level traffic complexities** and **user-specific activity patterns**. This integration is critical for designing a robust model that accurately detects and classifies encrypted traffic.

To ensure consistency and quality, the preprocessing of the augmented dataset follows a structured pipeline. Both datasets are **cleaned and aligned** to address differences in feature representations, followed by **feature extraction and transformation** processes. Key features, such as entropy-based attributes and network metadata, are fused to create a unified feature set suitable for training the proposed model. This preprocessing ensures that the augmented dataset retains its components' unique strengths and provides a **cohesive and standardized input** for the classification pipeline.

Upon completing the **network traffic analyzer**, it will be applied to the augmented datasets to assess their effectiveness in real-world encryption detection scenarios. These datasets were selected based on their relevance to current encryption techniques and the challenges they present in real-world cybersecurity applications. The analyzer will conduct extensive tests, generating **CSV files** that document key performance metrics such as **accuracy, precision, recall**, and other relevant statistics. This phase will play a crucial role in fine-tuning the analyzer's capabilities, ensuring its adaptability to different types of encrypted traffic.

Further details regarding the **selection and creation of training and testing datasets** are elaborated in **Chapter 4**, particularly in **Subsection 4.2**, where a comprehensive discussion of dataset characteristics, preprocessing methodologies, and dataset partitioning strategies is provided. This section is a foundational reference for understanding the data-driven approach employed in the proposed method.

3.2.4 Design and Implementation of the XAI-Powered Detection and Characterization Model

This section outlines the design considerations, implementation workflow, and evaluation strategies enabling the model to achieve robust, interpretable, high-performance encrypted traffic classification.
The proposed model integrates Explainable Artificial Intelligence (XAI) into encrypted traffic classification, enhancing detection accuracy and interpretability. Built upon a Hybrid Attention + LightGBM architecture, the model leverages multi-head attention mechanisms to refine feature extraction while utilizing LightGBM for efficient and scalable classification.
The model follows a two-layer classification approach:

1. **Layer 1 - Binary Classification:**Differentiates between encrypted and non-encrypted traffic.
2. **Layer 2 - Multi-Class Classification:** Further categorizes encrypted traffic into specific types, including VPN, TOR, I2P, ZeroNet, and Freenet.

To enhance model explainability, we incorporate SHAP and LIME for feature importance analysis, ensuring transparency in decision-making. The feature selection and pruning process is guided by SHAP, optimizing performance while maintaining interpretability. Additionally, the model undergoes further refinements based on the insights from SHAP and LIME.

3.3 Architecture Design

The proposed model's architecture, illustrated in **Figure 69**, consists of multiple stages designed to classify encrypted and non-encrypted network traffic efficiently. Additionally, it further categorizes encrypted traffic into sub-classes. The model integrates key components such as feature extraction, preprocessing, attention-based feature fusion, and classification while leveraging explainability techniques to enhance interpretability. This section provides a concise overview of each component and introduces the following subsections, where each step is discussed in greater detail. The architecture follows a **systematic flow**, beginning with the extraction of network traffic features and concluding with an explainable classification process. The main components of the model are:

- **CSV Generation using NTLFlowLyzer:** The first step involves extracting over **400+ handcrafted network traffic features** using **NTLFlowLyzer**, a traffic analysis tool designed for detailed packet and flow-level feature extraction. This step ensures that a comprehensive dataset is available for preprocessing and classification. The following subsection will discuss the CSV generation process, highlighting the extracted features and their relevance in encrypted traffic classification.
- **Data Preprocessing:** The extracted raw features undergo preprocessing to improve data quality and consistency. This includes *handling missing values, feature scaling, and data transformation techniques* tailored to encrypted traffic data characteristics. The **Data Preprocessing** subsection will explain how these steps prepare the dataset for optimal feature selection and classification.
- **Feature Selection & Attention Mechanism:** Given the large number of extracted features, **Mutual Information (MI)-based feature selection** is employed to **retain the most informative features** while reducing computational complexity. This process significantly impacts the model's efficiency and interpretability. The **Feature Selection** subsection will describe how MI is applied to identify the most relevant features

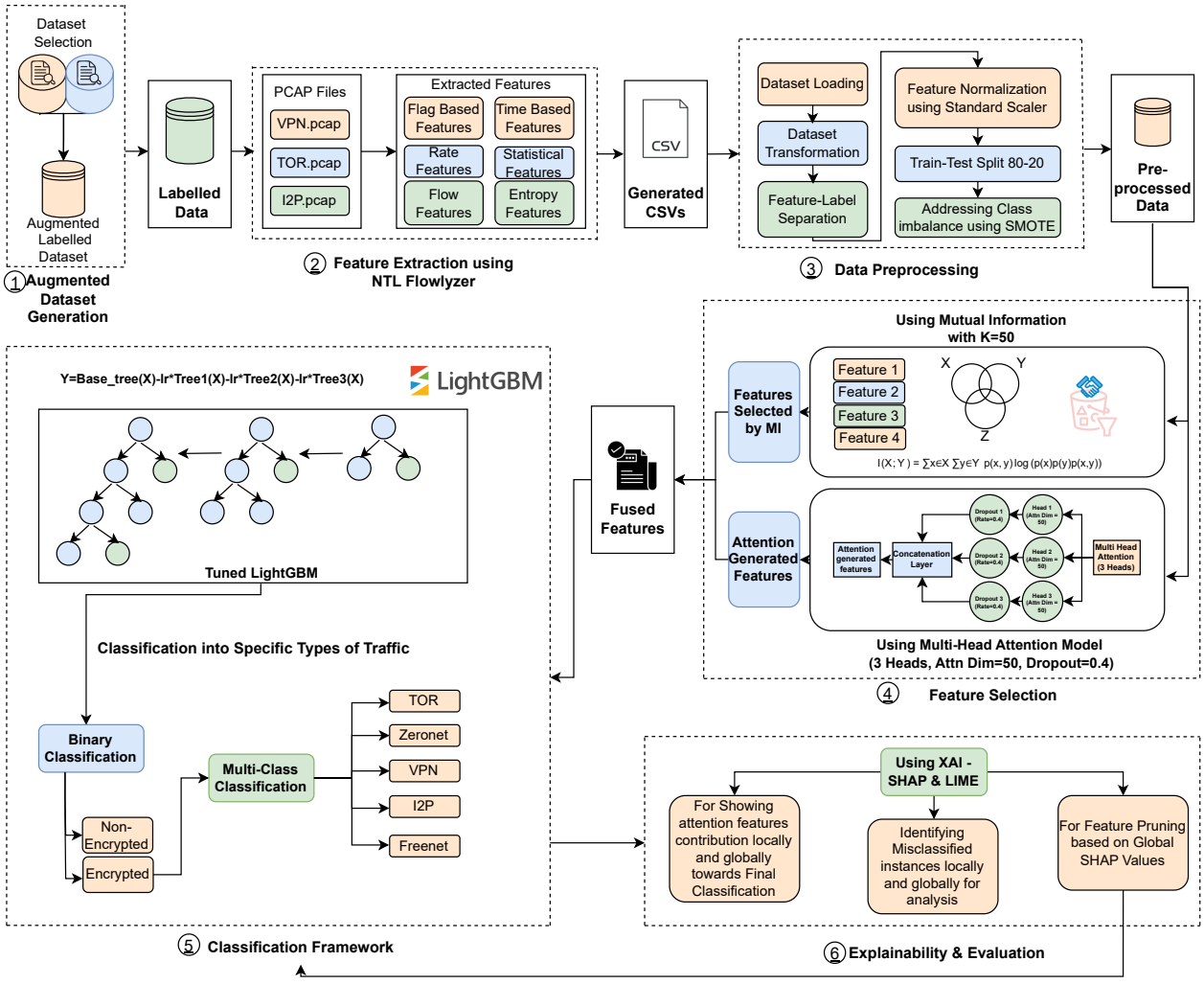

Figure 69: Architecture of the proposed Multi-Head Attention model with feature selection using Mutual Information (MI) and attention-based feature fusion, integrated with LightGBM for encrypted traffic classification.

contributing to accurate encrypted traffic classification. The features are processed through a **Multi-Head Attention (MHA) mechanism**, which enhances feature interactions by assigning different attention weights to various input attributes. This mechanism ensures that the model effectively learns intricate dependencies among traffic attributes. The features obtained from MI and MHA are fused for the final classification.

- **Classification Framework:** The **final classifier, LightGBM**, is then employed for encrypted traffic classification, leveraging its efficiency and suitability for structured data. The **Classification Framework** subsection will provide deeper insights into how attention-based feature fusion and LightGBM work together to improve classification performance.
- **Explainability and Evaluation:** To ensure interpretability, **SHAP and LIME techniques** are employed to analyze feature importance and explain classification decisions. This step is crucial for understanding how the model distinguishes between different types of encrypted traffic. Performance metrics such as **accuracy, precision, recall, and F1-score** are used to evaluate the model's effectiveness. The **Explainability and Evaluation** subsection will describe how these techniques enhance model transparency and reliability.

Each subsection will provide an in-depth discussion of their respective methodologies, ensuring a comprehensive understanding of the proposed model's design, optimization, and evaluation.
The following subsection details the **CSV generation process using NTLFlowLyzer**, explaining how raw network traffic is processed to extract a rich set of features for subsequent classification.

3.3.1 CSV Generation using NTLFlowLyzer

The first step in the proposed architecture involves extracting relevant features from raw network traffic data to create a structured dataset suitable for classification. This process is handled by **NTLFlowLyzer**, a network traffic analyzer that processes packet capture (PCAP) files to generate flow-based feature representations. The extracted features provide critical insights into network behavior, enabling efficient encrypted and non-encrypted traffic classification. This subsection outlines the feature extraction workflow, the organization of extracted features into meaningful categories, and their significance in the classification process. The following subsections will build upon this foundation, detailing data preprocessing, feature selection, and classification mechanisms.

3.3.1.1 Feature Extraction Workflow

The feature extraction process is crucial in transforming raw network traffic data into a structured format suitable for machine learning-based classification. Below are the key stages involved in this transformation.
The process begins with **raw data collection**, where network traffic is captured as **PCAP files**, which store raw packet data between endpoints. These files encapsulate detailed network activity, including protocol headers, payloads, timestamps, and flow information, forming the foundational dataset for subsequent analysis.
Following data acquisition, **feature extraction** is performed using **NTLFlowLyzer**. This specialized network traffic analyzer processes PCAP files to extract a comprehensive set of statistical, temporal, and protocol-based features from network flows. A **flow** is defined as a bidirectional exchange of packets between two endpoints, aggregated based on IP addresses, ports, and protocols. The extracted features capture critical flow characteristics, such as packet counts, inter-packet delays, and entropy measures, enabling a granular analysis of encrypted and non-encrypted traffic patterns.

To enhance interpretability and facilitate machine learning applications, the extracted features undergo **feature categorization**, systematically grouped into multiple categories based on their statistical and structural properties. This structured representation ensures that the dataset is well-organized and optimized for classification tasks, allowing the model to learn meaningful distinctions between different types of encrypted network traffic.

3.3.1.2 Feature Categories

The extracted features are systematically categorized to provide distinct analytical perspectives on network traffic characteristics. This structured organization ensures that various traffic attributes are effectively captured, aiding in encrypted traffic classification. The different categories of flow are described below:

Basic Flow Statistics
Fundamental flow-based attributes are recorded to describe the core properties of each network flow. These include **duration**, which measures the total active time from the first to the last packet, **packet counts**, which tracks the number of packets exchanged, and **bytes transferred**, which quantifies the total data volume exchanged. Additionally, **directional statistics** are computed, distinguishing between client-to-server and server-to-client interactions. These directional features analyze forward and backward packet counts and payload and header byte distributions, allowing for a deeper understanding of protocol structures.

Statistical Measures
Various statistical metrics are derived from packet sizes, inter-arrival times, and other network attributes to characterize flow behavior further. These include **mean, median, standard deviation, and variance**, which quantify traffic variability, as well as **skewness and kurtosis**, which help detect deviations from expected traffic distributions. These metrics are instrumental in identifying anomalies and behavioral patterns within encrypted traffic flows.

Protocol-Specific Features
Several features are extracted to analyze protocol-level interactions. **TCP flag statistics** track the occurrences of critical flags such as SYN, ACK, FIN, and RST, providing insights into handshake behaviors, retransmissions, and connection termination patterns. Additionally, **initial window sizes** capture TCP congestion control parameters, offering valuable information on network flow dynamics. Further, **handshake information** is extracted to support protocol classification by analyzing metadata exchanged during connection establishment.

Advanced Metrics
Multiple entropy-based and statistical methods are employed to assess the complexity and structure of encrypted traffic. **Entropy calculations** measure randomness in packet payloads using different encoding formats, including **binary, hexadecimal, and UTF-8**. **N-gram entropy analysis** evaluates fixed-length byte sequences to detect structured patterns indicative of encryption. Additionally, **mutual information metrics** quantify dependencies between traffic attributes, aiding in feature selection by identifying strongly correlated features. Lastly, **delta time and length analysis** examines temporal variations and packet length distributions within flows, providing further insights into network behavior.

This structured approach to feature categorization ensures that machine learning models are provided with diverse

attributes, enabling precise classification of encrypted and non-encrypted network traffic.

3.3.1.3 Directional Analysis

All extracted features are computed separately for different transmission directions to capture bidirectional traffic behavior. The **forward direction** represents packets sent from the source to the destination. In contrast, the **backward direction** corresponds to packets traveling in the opposite direction, from the destination to the source. Additionally, a **complete flow** perspective aggregates statistics across both transmission directions, providing a holistic view of network interactions.

The feature extraction process, facilitated by **NTLFlowLyzer**, ensures that directional analysis remains an integral component of encrypted traffic classification. The extracted features offer deeper insights into traffic patterns by computing statistical, temporal, and protocol-specific properties across different flow directions. This structured approach enhances the ability to differentiate between encrypted and non-encrypted network flows, ultimately strengthening classification accuracy and interpretability.

The following subsection, **Data Preprocessing**, will discuss how these extracted features undergo transformation, normalization, and filtering to ensure they are suitable for classification. This includes handling missing values, scaling numerical features, and preparing the dataset for feature selection.

3.3.2 Data Preprocessing

The initial phase of the proposed encrypted traffic classification model focuses on **data preprocessing**. This critical step ensures the dataset is cleaned, structured, and balanced for subsequent machine learning tasks. This process systematically transforms raw network traffic data into a format suitable for classification by eliminating redundancies, handling missing values, and standardizing feature representations. The following steps detail the processes involved:

3.3.2.1 Dataset Loading

The dataset used in this study comprises features extracted from the network traffic analyzer **NTL Flowlyzer**. Only columns starting from the **7th index** are retained to optimize data utility, ensuring that irrelevant or redundant initial columns are excluded from further analysis. This selection process enhances computational efficiency and preserves only meaningful attributes for classification.

3.3.2.2 Data Transformation

Various transformation techniques are applied to prepare the dataset for analysis. **String length conversion** is performed on categorical columns (excluding the label), converting them into numerical representations based on their string lengths. This transformation ensures uniformity and compatibility with machine learning models, allowing categorical variables to be effectively utilized in classification tasks. Additionally, **handling missing values** is an essential step to maintain dataset integrity. Any missing or invalid entries (e.g., NaN, inf, -inf) are replaced with the median value of their respective features. This robust imputation strategy minimizes data loss while preserving statistical consistency.

3.3.2.3 Feature-Label Separation

To facilitate supervised learning, the dataset is structured by separating the input features from the target labels. The **features (X)** consists of all columns except the target label, representing the numerical and statistical attributes extracted from network traffic flows. Meanwhile, the **labels (Y)** correspond to the `label` column, serving as the ground truth for classification.
A **LabelEncoder** is applied to standardize categorical labels to ensure consistency and compatibility with machine learning models. This transformation converts textual class labels into integer representations, simplifying the classification task while preserving the original categorical distinctions. By performing this encoding, the model can efficiently process and interpret class assignments, ensuring a structured and uniform dataset for training and evaluation.

3.3.2.4 Feature Normalization

To address scale disparities among features, **StandardScaler** is used to normalize the dataset. This ensures all features have a mean of zero and a standard deviation of one, thereby improving model convergence and performance.

3.3.2.5 Train-Test Split

The dataset is split into training and testing sets using an 80-20 split ratio, with a fixed random state for reproducibility. This division allows for an unbiased evaluation of the model's performance on unseen data.

3.3.2.6 Addressing Class Imbalance

Class imbalance poses a significant challenge in machine learning classification tasks, often leading to biased predictions favoring the majority class. To mitigate this issue, the **Synthetic Minority Oversampling Technique (SMOTE)** is employed. SMOTE generates synthetic samples for underrepresented courses in the training set, effectively balancing the distribution of labels. This oversampling approach is particularly crucial in binary classification scenarios, where an uneven class distribution can skew model performance and reduce its ability to generalize across different traffic patterns. The dataset is structured by applying SMOTE to enhance classification fairness, ensuring that the model learns equally from all traffic classes.

The preprocessing step lays a strong foundation for the model's effectiveness by cleaning and transforming the dataset, addressing class imbalance, and standardizing features. This structured approach ensures high-quality input for machine learning algorithms, enhancing the robustness and reliability of the subsequent model training and evaluation stages while improving classification accuracy.
Once the dataset has been preprocessed, it undergoes **feature selection** to retain only the most relevant attributes for classification. The following subsection, **Feature Selection**, will discuss how **Mutual Information (MI)** and **Multi-Head Attention** are leveraged to identify essential features, ensuring optimal performance and interpretability in encrypted traffic classification.

3.3.3 Feature Selection

Feature selection is a crucial step in the encrypted traffic classification pipeline, aimed at reducing dimensionality, eliminating irrelevant features, and enhancing model performance by focusing on the most informative attributes. The **Mutual Information (MI) criterion** is employed for this purpose, leveraging the statistical dependence between features and the target variable to identify those that contribute most significantly to the classification task.

3.3.3.1 Feature Selection Methodology with Mutual Information

The Mutual Information criterion quantifies the amount of information shared between each feature and the target label, allowing for the identification of features that are most relevant to the classification process. This method ensures that only the most informative attributes are retained by capturing both linear and non-linear dependencies, improving model interpretability and efficiency. This section discusses the Implementation and the advantages of the selected Mutual Information Feature selection method.

Implementation Details

The top $k = 50$ features were chosen to select the most relevant features, striking a balance between computational efficiency and model performance. Feature selection was performed using the `SelectKBest` method in combination with the `mutual_info_classif` scoring function, ensuring that only the highest-impact features were retained. The selection process was conducted in two main steps. First, the **selector was trained** on the resampled training dataset (`X_train_res` and `y_train_res`), ensuring that feature selection accounts for the balanced class distribution generated by **SMOTE**. This step prevented feature bias due to class imbalance and improved classification fairness. Following feature selection, the **transformation phase** extracted the selected features from both the training and testing datasets, effectively reducing dimensionality while maintaining crucial classification attributes.

Advantages of Mutual Information-Based Selection

Mutual Information-based feature selection offers several key advantages. Unlike traditional correlation measures, MI **captures both linear and non-linear dependencies** between features and the target variable, making it highly effective for complex classification tasks. Additionally, by retaining only the most informative features, this method **enhances dimensionality reduction**, reducing the risk of overfitting, improving computational efficiency, and increasing the model's generalizability. These benefits collectively strengthen the encrypted traffic classification model, ensuring it remains both interpretable and high-performing.

The feature selection process identified the top 50 features that contributed most significantly to the classification task. This step ensures that the subsequent modeling phase focuses on the most relevant attributes, improving the model's interpretability and performance. By employing Mutual Information, the approach accounts for complex dependencies, further reinforcing the robustness of the classification pipeline.

3.3.3.2 Feature Selection Methodology with Multi-Head Attention Layer

The proposed model integrates a **Multi-Head Attention Layer** to enhance feature representation and capture dependencies between features. This attention mechanism is designed to focus on the most informative aspects of the input data, thereby improving classification accuracy and interoperability. This section presents the introduction

to Multi-Head Attention and subsequently discusses the design, construction, parameters, and advantages of the applied Multi-Head Attention.

Introduction to Multi-Head Attention

The Multi-Head Attention Layer extends traditional attention mechanisms by employing multiple parallel attention heads. Each head captures different aspects of feature dependencies, allowing for a more comprehensive representation of the input data. This is particularly beneficial when handling high-dimensional datasets, such as the 400+ extracted features from **NTL Flowlyzer**, where attention mechanisms help identify meaningful relationships between traffic attributes.

Design of the Multi-Head Attention Layer

The architecture of the Multi-Head Attention Layer is designed to optimize feature interactions, ensuring that the most relevant information is retained for classification. The mechanism operates by applying multiple independent attention heads in parallel, with each head processing distinct subsets of the feature space. The outputs from all attention heads are then concatenated to form a unified representation, enriching the model's understanding of encrypted traffic patterns. The different components, steps, and key parameters of the Multi-Head Attention Layer are as follows:

- **Attention Layer with Dropout**
  A fundamental component of the attention mechanism is the single attention layer, which incorporates **dropout regularization** to mitigate overfitting. The computation follows a structured process: first, **softmax attention scores** are generated, normalizing feature importance weights. These scores are then used for **feature weighting**, where input features are scaled based on their computed importance. Subsequently, the **weighted summation** aggregates these features into a single vector representation. Finally, a **dropout rate of 0.4** is applied, preventing overfitting and enhancing the model's generalization capabilities.

- **Multi-Head Attention Layer**
  Expanding upon the single attention mechanism, the Multi-Head Attention Layer employs multiple attention heads operating in parallel. Each head independently learns different feature interactions, ensuring the model captures a broad range of relationships within the dataset. The outputs from each attention head are then **concatenated** along the feature axis, forming an enriched, attention-enhanced feature representation that improves classification robustness.

- **Model Construction**
  To implement the Multi-Head Attention Layer, a utility function, `build_multihead_attention_model`, is defined. The model construction follows a systematic approach, where **input features are reshaped** for compatibility with the attention mechanism. The reshaped inputs are then processed through the Multi-Head Attention Layer, producing a final set of attention-enhanced features that serve as refined inputs for classification.

- **Key Parameters and Configuration**
  The Multi-Head Attention Layer is configured with tunable parameters that influence model performance. The **number of attention heads** is adjustable, allowing flexibility in capturing diverse feature dependencies. The **attention dimensions** determine the feature space each head processes, ensuring sufficient granularity in

learned representations. A **dropout rate of 0.4** is applied to improve regularization while preserving essential information. The **input dimension** is also derived from the preprocessing step, aligning with the selected feature set.

- **Building and Applying the Attention Model**
  Once defined, the Multi-Head Attention mechanism is leveraged to generate attention-enhanced feature representations, which are subsequently fused with the original features for classification. This ensures the model benefits from raw and attention-enhanced representations, improving its robustness and accuracy.

  The Multi-Head Attention Model is constructed using the `build_multihead_attention_model` function. The key configurations include setting the **input dimensions** to match the number of selected features after Mutual Information-based feature selection (`X_train_res_selected.shape[1]`). The **number of attention heads** is configured as 3, balancing diverse feature dependencies with computational efficiency. Each attention head processes **50 attention dimensions**, ensuring sufficient feature extraction depth. A **dropout rate of 0.4** is applied during training to mitigate overfitting.

- **Generating Attention-Enhanced Features**
  The attention model is applied to training and testing datasets to generate enhanced feature representations. The `X_train_res_selected` dataset is processed through the attention model, yielding `attention_features_train`, while the `X_test_selected` dataset is transformed into `attention_features_test`. These attention-enhanced features effectively capture intricate dependencies within the dataset, providing a richer representation for classification.

Advantages of Multi-Head Attention
Integrating Multi-Head Attention within feature selection offers several benefits by combining multiple attention heads, the mechanism **enhances feature representation**, enabling the model to learn complex dependencies in encrypted traffic patterns. Dropout regularization improves generalization, reducing the risk of overfitting despite high-dimensional inputs. Moreover, the attention mechanism **improves interpretability**, highlighting critical features that influence classification, which is particularly valuable for understanding encrypted traffic behaviors.

The proposed model achieves superior performance in encrypted traffic classification by integrating Multi-Head Attention into feature selection. The attention mechanism's ability to focus on the most informative attributes ensures that classification remains robust, interpretable, and well-suited to modern encrypted network traffic complexities.

3.3.3.3 Fusing Features

The attention-enhanced features are seamlessly integrated with the original selected features to maximize the information available to the classifier. This fusion process enriches the dataset by combining the strengths of raw and attention-processed representations, leading to a more comprehensive input space for classification. In the **training set**, feature fusion is performed by concatenating `X_train_res_selected` with `attention_features_train` along the feature axis, yielding the final `X_train_fused` dataset. Similarly, for the **testing set**, `X_test_fused` is constructed by merging `X_test_selected` with `attention_features_test`, ensuring consistency in feature representation across training and evaluation phases.

Advantages of Feature Fusion

Fusing raw and attention-enhanced features introduces several key advantages significantly improving the model's classification capabilities. **Augmented representation** is achieved by expanding the feature space, allowing the model to leverage original and transformed attributes better to understand encrypted traffic patterns. This enhanced input structure contributes to **improved classification accuracy**, as attention-enhanced features emphasize crucial dependencies within the dataset, complementing the raw features and refining the model's predictive capabilities. Furthermore, feature fusion enhances **interpretability**, as the attention mechanism highlights the most influential features, offering more profound insights into the decision-making process of the model.

Incorporating the Multi-Head Attention Model and subsequent feature fusion marks a pivotal advancement in the model's architecture. By integrating attention-enhanced features with the original dataset, the model benefits from a robust and comprehensive input representation, ultimately leading to superior encrypted traffic classification performance with high accuracy and improved generalization across diverse traffic patterns.

3.3.4 Classification Framework and Attention Mechanism

The proposed classification model employs a two-layer architecture to analyze network traffic, each performing a distinct classification task. The first layer performs binary classification to separate encrypted traffic from non-encrypted traffic, while the second layer performs multi-class classification to categorize encrypted traffic into specific types. The architecture is enhanced by an attention mechanism that enriches the input features, improving classification accuracy and interoperability. The following subsections will detail the classification framework and attention mechanisms in detail.

3.3.4.1 Tuned LightGBM Model Initialization and Training

The classification phase of the proposed model leverages a tuned LightGBM classifier to achieve high accuracy and efficiency in predicting traffic classes. The model parameters are meticulously optimized to enhance performance while avoiding overfitting.

LightGBM (Light Gradient Boosting Machine) is a gradient boosting framework designed for efficiency and scalability. Its ability to handle large-scale datasets and capture complex feature interactions makes it an ideal choice for encrypted traffic classification.

Model Configuration

The LightGBM model is initialized with hyperparameters that have been fine-tuned using **Randomized Search CV**, where a range of values was defined for each parameter to identify the optimal configuration. These hyperparameters influence the model's learning process and directly impact classification performance. Table 1 provides a detailed description of each parameter and its significance.

- **colsample_bytree:** Set to 0.9249, determining the fraction of features randomly sampled for each decision tree.
- **learning_rate:** Set to 0.0691, controlling the step size during gradient descent to ensure stable convergence.

- **max_depth:** Set to -1, allowing trees to grow to an unlimited depth, ensuring flexibility in capturing data patterns.
- **min_child_samples:** Set to 10, defining the minimum number of samples required in a leaf node before a split can occur.
- **n_estimators:** Configured to 383, specifying the total number of boosting iterations in the ensemble model.
- **num_leaves:** Fixed at 47, determining the number of leaves in each decision tree, balancing model complexity and overfitting.
- **reg_alpha and reg_lambda:** Regularization parameters set to 0.2184 and 0.4165, respectively, to prevent overfitting by penalizing large weights.
- **subsample:** Set to 0.9767, ensuring that each boosting iteration uses only a fraction of the available training data, improving generalization.
- **random_state:** Set to 42, maintaining reproducibility by controlling the randomness in training.

Table 1: Brief Description of LightGBM Hyperparameters

Parameter	Description
`colsample_bytree`	Determines the fraction of features randomly sampled for each tree, affecting feature selection diversity.
`learning_rate`	Controls the step size in gradient descent optimization, balancing training speed and model stability.
`max_depth`	Specifies the maximum depth of individual trees. Setting it to -1 allows unlimited depth, capturing complex patterns.
`min_child_samples`	Defines the minimum number of data samples required in a leaf node before further splitting is permitted.
`n_estimators`	Represents the total number of boosting iterations, influencing the overall model complexity.
`num_leaves`	Determines the number of terminal leaves per tree, controlling the granularity of decision rules.
`reg_alpha` and `reg_lambda`	Regularization terms (L1 and L2, respectively) that penalize large weights, reducing the risk of overfitting.
`subsample`	Defines the fraction of the training dataset sampled for each boosting round, improving generalization by introducing randomness.
`random_state`	Sets a fixed seed to ensure reproducibility of training and evaluation results.

Training the Tuned LightGBM Model

The tuned LightGBM model is trained using the fused feature set (`X_train_fused`) and the resampled labels (`y_train_res`). Key highlights include:

- **Fused Features:** The training data encapsulates explicit and implicit feature dependencies by combining raw features with attention-enhanced representations.
- **Resampled Labels:** The resampling process ensures balanced class distributions, improving the model's ability to classify minority classes accurately.

The `.fit()` method trains the model, enabling it to learn patterns and relationships within the fused feature space.

Advantages of the Tuned LightGBM Model

This section highlights the key benefits of using the optimized LightGBM model for encrypted traffic classification. The tuned hyperparameters, selected through **Randomized Search CV**, improve the model's efficiency, accuracy, and robustness, making it well-suited for handling large-scale network traffic data. LightGBM's inherent design and the applied optimizations contribute to its superior performance in this classification task. The following points summarize the significant advantages:

- **Efficiency:** LightGBM's histogram-based approach and feature sampling enable faster training and prediction, significantly reducing computational overhead compared to traditional gradient boosting methods.
- **Accuracy:** The optimized hyperparameters enhance the model's ability to accurately classify encrypted and non-encrypted traffic while maintaining high precision for various encrypted traffic types.
- **Scalability:** LightGBM efficiently handles large datasets with high dimensionality, making it particularly effective for the 400+ extracted features from **NTL Flowlyzer**. Its ability to scale without significant performance degradation makes it ideal for real-time and large-scale network traffic analysis.
- **Robustness:** The incorporation of **regularization parameters** (`reg_alpha` and `reg_lambda`) and resampling techniques (**subsample**) mitigates overfitting, ensuring reliable classification across diverse traffic categories. These mechanisms enhance generalization, preventing the model from memorizing noise in the dataset.

The implementation and training of the tuned LightGBM model form a critical step in the classification pipeline. By utilizing optimized hyperparameters and fused features, the model achieves high accuracy and efficiency in distinguishing between encrypted and non-encrypted traffic and in multi-classifying encrypted traffic types. This robust and scalable approach underscores the strength of the proposed methodology in addressing the complexities of encrypted traffic classification.

3.3.4.2 Layer 1: Binary Classification

The first layer of the proposed model is responsible for distinguishing encrypted traffic from non-encrypted traffic. This foundational step ensures that encrypted traffic, which is inherently more challenging to analyze due to its obfuscated nature, is effectively isolated. The model sets the stage for the subsequent multi-class classification phase by accurately separating encrypted and non-encrypted traffic, ensuring that only encrypted data is further analyzed.

Classifier Used

A tuned LightGBM classifier is employed to accomplish this task. LightGBM is chosen due to its efficiency in handling large datasets and ability to perform well in high-dimensional and imbalanced data scenarios. Its gradient-boosting approach enables it to learn complex patterns while maintaining computational efficiency. Moreover, LightGBM's capability to handle missing values and its built-in feature selection mechanism makes it particularly well-suited for network traffic classification, where datasets often contain relevant and redundant features.

Input Features

The input to this binary classification layer consists of attention-fused features. These features encapsulate raw traffic attributes and representations enhanced by the attention mechanism. By incorporating attention-based feature selection, the model leverages individual and interdependent patterns in the data, improving its ability to differentiate between encrypted and non-encrypted traffic. This approach ensures that the classifier does not rely solely on traditional statistical indicators but benefits from deep feature relationships learned during preprocessing.

Training and Evaluation

During the training phase, the LightGBM classifier is trained on a preprocessed dataset where resampling techniques are applied to address class imbalances. This step ensures that the model does not develop a bias toward the majority class, leading to a more balanced classification performance. Hyperparameter tuning is also conducted to optimize the model's parameters, improving its predictive accuracy.

Multiple metrics such as accuracy, precision, recall, and F1-score are used for evaluation. Given the primary objective of identifying encrypted traffic, recall is given special emphasis to minimize false negatives and ensure that encrypted traffic is correctly identified for further classification. The results of this phase directly influence the subsequent multi-class classification stage, detailed in the following subsection.

3.3.4.3 Layer 2: Multi-Class Classification

Once traffic is classified as encrypted, it is categorized into specific encryption-based services, including Tor, VPN, I2P, Freenet, and ZeroNet. This multi-class classification stage builds upon the binary classification layer's output and aims to provide a granular understanding of encrypted traffic flows. Since different encryption services exhibit unique characteristics, the classifier must generalize well across all categories while addressing class imbalance and misclassification risks.

Classifier Used

The same LightGBM classifier is utilized for this layer, but it is specifically tuned for multi-class classification. The ability of LightGBM to handle categorical differentiation with minimal computational overhead makes it a suitable choice. Furthermore, its ability to apply class-specific weighting ensures that minority classes, such as Freenet, are not overshadowed by more prevalent traffic types like VPN and Tor.

Input Features

The input features for this layer remain consistent with those used in the binary classification stage. The attention-fused feature set ensures continuity between classification layers, preventing redundant computations and maintaining efficiency. Using a unified feature representation, the model can effectively capture both general and specific characteristics of encrypted traffic, facilitating accurate multi-class predictions.

Training and Validation

During training, the classifier is trained on a labeled dataset that incorporates augmentation techniques such as SMOTE to address the class imbalance problem. Minority classes, particularly Freenet, are enhanced with synthetic samples to ensure the classifier learns sufficient patterns from each category. Additionally, hyperparameter tuning and class weighting techniques are applied to improve the model's ability to generalize across all classes, preventing it from favoring dominant traffic types.

Performance metrics such as macro-F1 score, precision, recall, and accuracy are employed for validation. Unlike binary classification, where recall was prioritized, macro-F1 is given greater importance in multi-class classification to ensure balanced performance across all categories. Furthermore, a confusion matrix analyzes misclassification patterns, guiding refinements in the model's decision-making process. The results from this classification layer highlight the importance of targeted refinements, which are further discussed in the challenges and solutions section.

Challenges and Solutions

A key challenge in multi-class classification is handling class imbalance, as some encrypted traffic categories appear far less frequently than others. To mitigate this issue, techniques like SMOTE were applied to minority classes, and class-specific weighting was introduced to prevent model bias. Additionally, another significant challenge was ensuring the model's interpretability. Since deep learning models often operate as black boxes, post-classification analysis using SHAP and LIME was conducted to provide insights into which features contributed most to each classification decision. This analysis enabled localized refinements, enhancing the model's overall reliability. The following section delves into the attention mechanism, crucial in refining feature representation.

3.3.4.4 Attention Mechanism

The attention mechanism is a fundamental component of the proposed model, designed to enhance feature quality by selectively emphasizing critical traffic characteristics. By focusing on key attributes while de-emphasizing redundant information, the attention mechanism strengthens the model's ability to distinguish between encrypted traffic types. This approach is particularly valuable in network traffic classification, where subtle differences in feature distributions can significantly impact classification accuracy.

Integration into Feature Fusion

During preprocessing, the attention mechanism is applied to generate enriched feature representations. Instead of treating all features equally, it assigns varying levels of importance to different attributes, ensuring that the most relevant data points contribute to classification. These fused features are then fed into binary and multi-class classification layers, maintaining consistency across the model's hierarchical structure. The model can dynamically adapt to different traffic patterns by leveraging attention and enhancing its predictive accuracy.

Benefits of Attention Integration

The integration of attention offers several advantages. First, it enhances feature representation by prioritizing informative attributes, effectively reducing noise in the dataset. This leads to improved classification accuracy and reduces reliance on manually engineered features. Second, attention-based feature selection improves interpretability, as the attention weights provide insight into which traffic characteristics influence predictions the most. Lastly, the adaptability of the attention mechanism ensures that the model remains robust across various encrypted traffic types, allowing it to generalize well in real-world scenarios.

Validation

To validate the effectiveness of the attention mechanism, SHAP analysis was conducted to assess the significance of

attention-fused features. The results confirmed that these features played a crucial role in improving classification performance, further reinforcing the need for feature refinement. The model achieves a balanced trade-off between interpretability and predictive performance by incorporating attention-enhanced feature representation.
The overall architecture of the proposed model relies on a two-layer classification framework powered by the attention mechanism. The binary classification layer ensures that only encrypted traffic is analyzed, while the multi-class classification layer further categorizes it into specific encrypted services. By leveraging attention-based feature selection, the model excels in handling class imbalance, improving interpretability, and achieving high classification accuracy. The following section provides an in-depth discussion of the model's architecture, detailing its structural components and design principles.

3.3.5 Explainability and Evaluation

This section discusses the importance of explainability in the proposed encrypted traffic classification model and evaluates its performance. Ensuring transparency in the model's decision-making process is crucial, particularly in security-related applications. Explainability is incorporated at multiple stages, demonstrating how different layers of the model process and interpret input features. Post-classification, tools such as **SHAP** (SHapley Additive exPlanations) and **LIME** (Local Interpretable Model-Agnostic Explanations) are utilized to validate feature importance and provide insights into model behavior. These methods confirm that the attention-based feature fusion mechanism enhances classification accuracy and interpretability. This section offers a layer-wise breakdown of explainability techniques, focusing on how SHAP and LIME are applied to assess and refine model performance.

3.3.5.1 Explainable AI (XAI) in Layer 1 (Binary Classification)

This subsection explains how XAI techniques are employed in the first classification layer to distinguish between encrypted and non-encrypted traffic. SHAP and LIME are applied to analyze the feature contributions at both a global and local level, ensuring the reliability and interpretability of the classification model.

1. **SHAP and LIME for Feature Importance**

   Explainable AI (XAI) techniques are employed to ensure interpretability and validate the proposed model's effectiveness in the first layer. These include SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations). Both methods provide complementary insights into the model's predictions by highlighting the contributions of individual features.

   - **SHAP for Feature Importance**
     SHAP (SHapley Additive exPlanations) is employed to assess features' global and local contributions in distinguishing between encrypted and non-encrypted traffic. By computing SHAP values for all input features, this method effectively identifies the most influential attributes contributing to classification, enhancing the interpretability of the model's predictions.

     Implementation of SHAP
     The implementation of SHAP in the proposed model follows a structured approach to ensure accurate feature attribution. First, a **SHAP explainer** is initialized using the `TreeExplainer` for the tuned LightGBM model. A **combined feature list** is constructed to maintain consistency in feature interpretation,

integrating both original and attention-enhanced feature names. Next, **SHAP value computation** is performed on the fused feature set, focusing on the positive class (encrypted traffic) to identify key discriminative features. The results are then visualized using various **SHAP plots**, including a **global summary plot**, which ranks features based on their contribution across the dataset, and a **local force plot**, which illustrates how individual features influence specific predictions.

Validation of the Attention Mechanism

SHAP analysis further validates the role of the attention mechanism by demonstrating its impact on feature dependencies, model performance, and interpretability. The attention mechanism effectively captures intricate **feature dependencies**, revealing complex interactions among features that might be overlooked. Additionally, incorporating attention-enhanced features leads to **enhanced model performance**, as these refined attributes significantly improve classification accuracy. Furthermore, SHAP's **improved interpretability** through global and local explanation plots provides deeper insights into how each feature influences the model's decisions, ensuring transparency in encrypted traffic classification.

By leveraging SHAP for feature importance analysis, the proposed model gains a robust mechanism for validating the effectiveness of attention-based feature selection. This approach not only improves classification accuracy but also enhances the interpretability of the model, making it well-suited for encrypted traffic classification tasks.

- **LIME for Local Feature Importance**

  LIME (Local Interpretable Model-Agnostic Explanations) is a complementary technique to SHAP that provides localized explanations for individual predictions. Unlike SHAP, which assesses global feature importance, LIME focuses on instance-level contributions, highlighting the specific features responsible for each classification decision. This localized approach enables a more granular understanding of how the model predicts individual traffic samples.

  Implementation of LIME

  LIME is applied to analyze feature contributions at the instance level by approximating the model's local decision boundary. The implementation follows a structured approach to ensure accurate interpretability. First, a **LIME explainer** is initialized using `LimeTabularExplainer`, configured with the fused training feature set, class labels (`Encrypted` and `Non-Encrypted`), and feature names to maintain consistency in interpretation. Next, **instance-level explanation** is performed by selecting a single test instance (e.g., the first test sample) and applying the LightGBM classifier's prediction function to generate feature contributions. Finally, **visualization techniques** are employed, where the LIME explanation clearly represents the most influential features for the given test instance, demonstrating their impact on the model's decision.

  The proposed model benefits from both global and local interpretability by leveraging LIME alongside SHAP. This dual approach ensures a comprehensive understanding of feature importance at both dataset and instance levels, reinforcing model transparency and improving trust in encrypted traffic classification.

- **Combined Advantages of SHAP and LIME**

  By integrating both **SHAP** and **LIME**, the proposed model benefits from a comprehensive explainability framework that provides insights at both global and local levels. **SHAP** offers a dataset-wide understand-

ing of feature importance, highlighting the most influential attributes across all classifications. In contrast, **LIME** provides instance-level explanations, pinpointing the key features responsible for individual predictions. This combination ensures a well-rounded interpretation of the model's decision-making process, reinforcing the transparency and reliability of the classification framework.

The complementary nature of these techniques enhances interpretability in multiple ways. **SHAP** provides a **global perspective** on feature contributions, ensuring a broad understanding of how different attributes influence classification outcomes. In contrast, **LIME** supplements this with **localized insights**, explaining individual predictions in detail. Furthermore, both methods serve as validation tools for key **design choices**, confirming the effectiveness of the attention mechanism and its role in refining feature representation. This validation strengthens confidence in the model's ability to differentiate between encrypted and non-encrypted traffic accurately. Additionally, the integration of SHAP and LIME significantly improves **model transparency**, making the classification decisions more interpretable and justifiable.

The application of SHAP and LIME in the first layer of the classification framework underscores the critical role of **attention-fused features**. It validates their contribution to improved classification performance. **SHAP** provides global and instance-level feature importance insights, while **LIME** focuses on localized explanations, highlighting the features influencing specific classification outcomes. Together, these methods ensure the binary classification process is transparent and reliable, establishing a robust foundation for the subsequent multi-class classification layer.

2. **SHAP for Feature Pruning**

In the first layer of the classification framework, SHAP-based feature pruning is employed to optimize model efficiency by retaining only the most impactful features. This technique ensures the model remains computationally efficient while maintaining or enhancing classification performance. The pruning workflow systematically reduces the number of input features while preserving essential information necessary for accurate classification.

The process begins with **SHAP value computation**, where SHAP values are generated for all features in the encrypted traffic classification task to quantify their importance. To achieve this, **mean absolute SHAP values** is computed for the positive class (encrypted traffic), measuring the overall contribution of each feature to the model's predictions. Subsequently, a **feature importance ranking** is constructed, organizing features in descending order of their SHAP values. This ranking is the basis for selecting the most relevant attributes while discarding less influential ones.

Following the ranking process, **feature selection** is performed to refine the dataset by eliminating features that contribute minimally to classification. A predefined threshold, such as retaining the top 75% of features with the highest SHAP values, ensures that only the most informative attributes are preserved. Features exceeding this threshold are retained, whereas those with lower SHAP values are pruned. This step reduces redundancy in the dataset and prevents unnecessary complexity in the classification model. After pruning, the **dataset is filtered** to retain only the selected features in both training and testing sets, effectively reducing the dimensionality of the input data.

Retraining the Model

To evaluate the effectiveness of feature pruning, the LightGBM classifier is retrained on the pruned dataset

while maintaining the original hyperparameter configurations. This ensures a fair comparison between the pruned and unpruned models. The performance of the retrained model is assessed using standard evaluation metrics, including **accuracy, precision, recall, and F1-score**, to determine whether feature pruning maintains or improves classification effectiveness.

Post-Pruning Analysis

Once retrained, SHAP and LIME are reapplied to the pruned model to validate its interpretability and confirm the continued relevance of attention-enhanced features. **SHAP analysis** is conducted to compare feature importance before and after pruning, ensuring that the most influential features continue to drive model decisions. The study considers **global importance**, where a SHAP summary plot ranks the most critical features, including original and attention-enhanced attributes. Additionally, **local importance** is assessed using SHAP force plots, illustrating how individual features influence specific predictions, providing granular interpretability.

To further validate the impact of feature pruning, **LIME interpretation** is employed to generate instance-level explanations. This approach ensures that the retained features remain meaningful and contribute significantly to classification decisions. Through LIME-based **local explanations**, the model's decision-making process is visualized for a single test instance, offering an in-depth breakdown of the specific features influencing the classification outcome.

The classification framework ensures a balance between computational efficiency, classification performance, and model transparency by integrating SHAP-based pruning, model retraining, and post-pruning interpretability analysis. This structured approach strengthens the reliability of encrypted traffic classification while preserving interpretability and robustness.

3.3.5.2 Explainable AI (XAI) in Layer 2(Multi Classification)

This subsection explains how XAI techniques are employed in the second classification layer. To ensure the interpretability and transparency of the multi-class classification layer, SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) are applied. These XAI techniques provide insights into the model's decision-making process by highlighting the contributions of individual features in predicting encrypted traffic classes.

1. **SHAP and LIME for Feature Importance**

   - **SHAP for Feature Importance**

     SHAP was employed to evaluate global and local feature importance in the multi-class classification task, ensuring a transparent and interpretable assessment of feature contributions. The methodology involved computing SHAP values to quantify the influence of each feature on classification decisions. A `TreeExplainer` was initialized for the tuned LightGBM model, and SHAP values were calculated for the fused feature set, incorporating both original and attention-enhanced features. This allowed for a detailed analysis of feature interactions and their impact on classification performance.

     A **class-specific analysis** was conducted to enhance further interpretability, wherein SHAP values were examined separately for each encrypted traffic class, including Tor, VPN, I2P, Freenet, and Zeronet. This step provided more profound insights into the unique feature contributions of each class, highlighting

which attributes played a crucial role in their identification. Additionally, global SHAP summary plots were generated for each class, offering a detailed view of the most influential features within each category.

Beyond individual class analysis, a **combined analysis** was performed to assess feature importance holistically across all traffic classes. SHAP values were aggregated to identify features with consistent influence across multiple classifications. This broader perspective ensured that the most significant features contributing to overall classification performance were recognized, reinforcing the robustness of the model's feature selection strategy.

Global and Local Interpretability

To comprehensively understand the model's decision-making process, interpretability was assessed at both **global** and **local** levels. **Global interpretability** was achieved through SHAP summary plots, which provided an overarching view of feature contributions, illustrating their role in differentiating various encrypted traffic classes. This allowed for a high-level understanding of the key attributes shaping classification decisions. On the other hand, **local interpretability** focused on instance-specific predictions, using SHAP force plots to visualize how individual features influenced particular classification outcomes. This fine-grained analysis ensured the model's decisions were explained and justified at every level, reinforcing transparency and trust in its predictive capabilities.

- **LIME for Local Feature Importance** LIME complemented SHAP by offering localized explanations for individual predictions, enabling a more fine-grained interpretability assessment at the instance level. While SHAP provides a global understanding of the importance of features across the entire dataset, LIME focuses on explaining specific classification decisions. It is a valuable tool for understanding model behavior in real-world applications.

  LIME-Based Feature Analysis

  LIME was applied to the multi-class classification layer to analyze localized feature contributions. The process began with an **instance-level explanation**, where a single test instance was selected for detailed analysis using a `LimeTabularExplainer`. This step allowed for the decomposition of the model's prediction, highlighting the specific contributions of individual features to the classification decision for the selected instance. By breaking down the feature attributions, LIME provided insight into the factors driving the model's decision-making at a granular level.

  To further enhance interpretability, the **LIME output was visualized**, illustrating prediction probabilities and the relative importance of features in determining the classification outcome. The visualization effectively demonstrated how different features influenced the decision-making process, ensuring transparency in the model's classification of encrypted traffic types. The proposed model benefits from global and local interpretability by integrating LIME alongside SHAP, reinforcing trust and confidence in its predictive capabilities.
- **Combined Advantages of SHAP and LIME** Integrating SHAP and LIME provides a robust explainability framework, offering global and local interpretability of the model's decisions. While SHAP explains overall feature importance across the dataset, LIME provides instance-specific insights, ensuring a comprehensive understanding of the classification process. The complementary nature of these

techniques enhances model interpretability, allowing for a more transparent and reliable decision-making framework.

The combined analysis of SHAP and LIME offers several advantages. **SHAP provides a global perspective** by ranking feature importance across all classifications, while **LIME delivers localized explanations** for individual predictions, enabling a more granular understanding of the model's behavior. Additionally, both techniques played a crucial role in the **validation of the attention mechanism**, reaffirming the significance of attention-enhanced features and their impact on improving model performance. This validation ensures that the model effectively leverages these features to distinguish between encrypted traffic classes. Moreover, the combined application of SHAP and LIME enhances **interpretability and transparency**, making the model's decisions more understandable and justifiable.

Applying SHAP and LIME in the multi-class classification layer further reinforces the comprehensive evaluation of feature importance. SHAP effectively highlights global and class-specific contributions, while LIME provides detailed, instance-level explanations for individual predictions. Together, these XAI techniques validate the effectiveness of the attention mechanism and ensure the interpretability of the model's predictions, reinforcing the robustness and transparency of the proposed classification framework.

2. **SHAP for Feature Pruning**

To optimize the multi-class classification model, SHAP-based feature pruning was applied to retain only the most impactful features. This process ensures the model remains efficient while maintaining high performance.

The feature pruning workflow consists of several structured steps that leverage SHAP values to identify and retain the most relevant features. The process begins with **SHAP value computation**, where SHAP values were calculated to assess each feature's contribution to the classification task. The mean absolute SHAP values were then computed across all samples and classes, quantifying the overall importance of each feature. Once the SHAP values were calculated, **feature ranking** was performed, where features were listed in a DataFrame alongside their SHAP importance values and sorted in descending order of significance.

Following the ranking, a **pruning threshold** was established to systematically retain only the most critical features while discarding redundant ones. A predefined threshold was set to retain the top 75% of features based on their SHAP importance values, ensuring that only the most influential attributes were preserved. Features exceeding this threshold were selected for further use, effectively reducing the dimensionality of the dataset. The final step, **dataset pruning**, involved refining the dataset by filtering both the training and testing sets to retain only the selected features. This step improved computational efficiency by creating a more focused input dataset for retraining.

Retraining the Model

The LightGBM classifier was retrained on the pruned dataset, following a structured approach to maintain consistency and ensure optimal performance. The process began with loading the **best parameters**, stored in a pre-existing JSON file, to configure the model with its optimal hyperparameters. Subsequently, **model initialization** was carried out, where a new LightGBM model was instantiated using the best parameters, ensuring consistency with the original configuration. Finally, **model training** was conducted on the p

dataset, improving computational efficiency while maintaining classification accuracy by focusing on the most significant features.

Post-Pruning Analysis

Once retrained, SHAP and LIME were reapplied to the pruned model to validate its interpretability and confirm the continued relevance of attention-enhanced features. **SHAP analysis** was conducted to compare feature importance before and after pruning, ensuring that the most influential features continued to drive model decisions. The study considered **global importance**, where a SHAP summary plot ranked the most critical features, including both original and attention-enhanced attributes. Additionally, **local importance** was assessed using SHAP force plots, illustrating how individual features influenced specific predictions, providing granular interpretability.

To further validate the impact of feature pruning, **LIME interpretation** was employed to generate instance-level explanations. This approach ensured that the retained features remained meaningful and contributed significantly to classification decisions. Through LIME-based **local explanations**, the model's decision-making process was visualized for a single test instance, offering an in-depth breakdown of the specific features influencing the classification outcome.

By integrating SHAP-based pruning, model retraining, and post-pruning interpretability analysis, the classification framework balanced computational efficiency, classification performance, and model transparency. This structured approach strengthened the reliability of encrypted traffic classification while preserving interpretability and robustness.

3. **SHAP and LIME for Misclassified Instances and Retraining**

To address misclassifications in the multi-class classification layer, SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) were employed to analyze and interpret the features contributing to incorrect predictions. These analyses informed targeted refinements to enhance model performance.

The misclassified instance analysis follows a structured approach, utilizing SHAP and LIME to understand and rectify errors in classification. The first step involved **identifying misclassified instances** by systematically comparing the true and predicted labels in the test set. For each misclassified sample, key attributes were recorded, including the **true label** (correct class), **predicted label** (model's output), and **prediction probabilities** (confidence scores for each class). This identification process allowed for a precise assessment of patterns in incorrect predictions.

Following identification, **SHAP analysis for misclassified instances** was conducted to investigate feature contributions that misled the model. The SHAP analysis was performed at both a global and local level. In the **global analysis**, SHAP values for misclassified instances were averaged across all classes, generating a summary plot highlighting features with significant contributions to errors. This step facilitated the identification of problematic feature patterns that may have led to misclassification. At the **local analysis** level, SHAP force plots were generated for individual misclassified instances, visualizing the interplay of specific features that influenced incorrect predictions.

To complement the SHAP-based insights, **LIME analysis for misclassified instances** provided instance-level explanations for specific misclassified samples. LIME's local interpretability framework highlighted how

particular feature interactions contributed to errors, ensuring a detailed breakdown of the decision-making process at an individual sample level.

The combined SHAP and LIME analyses revealed critical insights into feature contributions and class-specific challenges. SHAP's global and local interpretability enabled identifying key features that consistently contributed to misclassifications, highlighting potential feature redundancies, biases, or overlapping decision boundaries. LIME's instance-level analysis provided fine-grained explanations, revealing how specific feature interactions led to incorrect predictions. Notably, particular classes, such as ZERONET and I2P, exhibited significant feature overlap, making their differentiation more complex. Based on these findings, targeted refinements were introduced, including refined feature selection, class-specific threshold tuning, and additional data augmentation strategies. These optimizations ultimately contributed to improved classification accuracy and enhanced model robustness.

3.4 Concluding Remarks

The above-described approaches will be systematically implemented and evaluated, including attention-fused feature extraction, hyperparameter-tuned LightGBM classifiers, and SHAP- and LIME-based interpretability analyses. The results from these methodologies will provide insights into classification performance, feature importance, and model efficiency, ensuring a comprehensive assessment of the proposed encrypted traffic classification framework. By leveraging SHAP for feature pruning, the model's computational efficiency will be optimized without compromising predictive accuracy. Additionally, LIME will be employed to validate localized decision-making, ensuring transparency in classification outcomes.

These experiments will yield a detailed understanding of how attention mechanisms enhance feature representations, how feature selection impacts classification accuracy, and how explainability techniques contribute to model interpretability. The outcomes will further validate the effectiveness of the hybrid classification approach and guide potential refinements for improved robustness and adaptability in real-world scenarios.

A detailed discussion of the classification and explainability experiment results, including performance evaluations and interpretability findings, is provided in **Subsection 5.5, Model Optimizations and Final Results**, within **Chapter 5, Experiments & Results**.

This chapter presented the complete architecture and methodological foundation of the proposed encrypted traffic classification framework. By combining handcrafted features from a modular entropy-driven analyzer with advanced preprocessing, feature selection via Mutual Information and Multi-Head Attention, and a LightGBM classifier, the model demonstrates both high accuracy and adaptability. The integration of SHAP and LIME for explainability not only validates model predictions but also supports iterative refinement through feature pruning and misclassification analysis. These components work in synergy to address the real-world challenges identified in Chapter 2, including interpretability, scalability, and class imbalance. This chapter lays the technical groundwork for experimental evaluation in subsequent sections and forms the backbone of the thesis' contribution to the domain of explainable and robust encrypted traffic classification.

The next **Chapter 4 - Selecting and Creating the Training and Testing Dataset** explores the publicly available datasets, detailing the selection process, evaluation criteria, and synthesis of datasets. Finally, it presents the finalized selected augmented dataset, which serves as the foundation for training and evaluating the proposed model.

4 Selecting and Creating the Training and Testing Dataset

Selecting and preparing an effective dataset is a critical step in the development of an encrypted traffic classification model. This chapter explores the publicly available datasets relevant to encrypted traffic classification, evaluates their suitability based on predefined criteria, and details the process of dataset augmentation to enhance model performance. The goal is to ensure a diverse, well-balanced dataset that accurately represents real-world encrypted traffic patterns while addressing challenges such as class imbalance, dataset bias, and feature relevance. Additionally, the chapter outlines the methodology for generating CSV files from the raw traffic data using NTL Flowlyzer (NTL FL), ensuring structured and analyzable data for model training and evaluation.

4.1 Datasets Available

This subsection delves into the publicly available datasets within the domain of encrypted traffic analysis, providing a detailed overview of their characteristics, design methodologies, and contributions to advancing research in this field. By examining these datasets, the section highlights their role in facilitating the development and evaluation of machine learning models, identifying strengths and limitations, and understanding their applicability in diverse scenarios. The analysis also underscores the significance of these datasets in addressing key challenges, such as capturing realistic traffic patterns, balancing dataset diversity, and providing comprehensive labelling to enhance the accuracy and generalizability of encrypted traffic classification techniques.

1. **CTU-13 (2011, CTU University, Czech Republic)** - Focuses on botnet behavior analysis in network traffic. Analyzes botnet traffic across 13 scenarios with various protocols to differentiate between malicious and benign traffic. The dataset exhibits a notable limitation in its lack of direct focus on modern encrypted traffic technologies, which may impact the generalizability of the findings[119].

2. **MOORE SET and NOC SET(2013, Moore, A. W., & Shi Dong)** - Analyzes network behaviour and security through data from internet communications at a research site and Southeast University using protocols like HTTP, FTP, and SMTP. Although valuable for network protocol analysis, the dataset falls short in representing a diverse range of global internet traffic conditions, potentially limiting its applicability in varied geographical contexts[120].

3. **ISCX Tor/Non-Tor 2016 (2016, Canadian Institute for Cybersecurity)** - Includes Tor and non-Tor traffic for ETC. Mixes Tor and non-Tor traffic using protocols like HTTP, HTTPS, SMTP/S, and FTP over SSH. While the dataset effectively detects encrypted Tor traffic, its specific focus on Tor limits its broader applicability to other types of encrypted communications[121].

4. **ISCX VPN/Non-VPN 2016 (2016, Canadian Institute for Cybersecurity)** - Includes traffic from VoIP, P2P, and streaming over VPN and non-VPN channels. Analyzed using ISCXFlowMeter. The dataset primarily focuses on VPN traffic, which may not adequately represent the full spectrum of network threats, thereby narrowing its utility in broader security contexts[122].

5. **USTC-TFC2016 (2016, University of Science and Technology of China)** - Features traffic from applications like BitTorrent and Skype and malware communications. Aimed at distinguishing between user data and security threats. This dataset does not extensively cover all types of encrypted traffic, which may limit its effectiveness in comprehensive encrypted traffic analysis[123].

6. **BetterNet HTTPS (2016, Wazen Shbair)** - Captures full HTTPS raw PCAP files from popular websites, aiding in HTTPS traffic analysis. The dataset focuses exclusively on web traffic, omitting other crucial network interactions, which might restrict its relevance in more comprehensive network studies[124].

7. **Anon-17 (2017, Network Information Management and Security Lab)** - Focuses on the operational dynamics within anonymity networks such as Tor, I2P, and JonDonym. It provides an extensive collection of benign, encrypted, and non-encrypted traffic to facilitate research on the effectiveness of these anonymity services. The dataset includes detailed protocol usage across various scenarios to examine how these networks manage privacy protection. A notable limitation of Anon-17 is its exclusion of malicious traffic and modern attack vectors, which may affect the applicability of the findings to real-world threat scenarios[125][126].

8. **CIC IDS2017 (2017, Canadian Institute for Cybersecurity)** - Captures benign and malicious traffic using protocols like HTTP and FTP, including DDoS and Heartbleed attacks. While the dataset is tailored towards intrusion detection, this focus might overlook the nuances of deeper encrypted traffic analysis, potentially limiting the dataset's broader applicability[127, 128] .

9. **Google Home(2018/20, C Wang, S Kennedy)** - Focuses on device-specific interactions through 100 voice commands, collecting 1,500 traffic traces per command over nine weeks. A significant dataset for studying responses within a controlled environment. The dataset is specific to Google Home, which limits its generalizability to other encrypted voice-activated devices and may reduce its applicability in broader IoT security analyses[129].

10. **Mirage (2019, ARCLAB, University of Napoli 'Federico II')** - Focuses on mobile app traffic analysis by capturing encrypted traffic from Android devices. This dataset includes traffic from various applications across multiple protocols to differentiate between app-specific traffic patterns. The dataset is noted for lacking traffic diversity related to direct attacks, which may impact its applicability in network security research. However, its comprehensive data on mobile application behaviour makes it invaluable for studying encrypted mobile traffic characteristics [130].

11. **Orange (2020, Orange Labs, France)** - Focuses on encrypted web traffic classification using deep learning to enhance service detection across multiple web protocols such as HTTP/2 and QUIC. The dataset captures a comprehensive mix of encrypted traffic from a major European ISP structured to provide insights into service-level traffic behaviours without direct attack scenarios. While it offers an extensive label set for service classification, the dataset notably excludes explicit malicious traffic, which could affect its broader utility in security-focused studies. However, its detailed protocol coverage and high accuracy in service classification make it a valuable resource for developing more nuanced traffic classification systems [131].

12. **CIC-Darknet-2020 (2020, Canadian Institute for Cybersecurity)** - Amalgamates traffic from Tor and VPN, analyzing activities like browsing and chat. The dataset, focused on darknet activities involving TOR and VPN, may not generalize well to other network security issues, limiting its utility in broader cybersecurity contexts [34].

13. **Oregon & Ohio HTTP2 (2020, L Xu, D Dou)** - Features HTTP2 traffic from applications like Spotify and Facebook, aiming to improve ETC. The dataset is specific to HTTP2, which may hinder its generalization to analyses involving other protocols, potentially reducing its broader applicability[132].

14. **TOR/I2P/Zeronet/Freenet (2020, Huy Nguyen)** - Analyzes encrypted traffic from darknet platforms, crucial for security applications. This dataset might not apply to standard network environments, which could limit its utility in typical network security assessments[133].

15. **UCDavis – QUIC (2020, University of California, Davis)** - Provides real-world encrypted traffic patterns under the QUIC protocol. The dataset focuses on the QUIC protocol, potentially overlooking the nuances and implications of other network protocols, thereby limiting its comprehensive applicability[134].

16. **Browser 2020 (2020, T Van Ede, R Bortolameotti)** - Offers a comprehensive view of browser-generated traffic, enhancing app detection capabilities. The dataset's focus on mobile environments may not extend its applicability to other settings, potentially limiting its relevance in diverse networking contexts[135].

17. **CIC DoH Brw 2020 (2020, Canadian Institute for Cybersecurity)** - Analyzes DNS over HTTPS traffic from various browsers and servers, focusing on security against eavesdropping and man-in-the-middle attacks. The dataset is solely focused on DNS over HTTPS (DoH), which potentially overlooks a broader range of DNS security issues, thereby limiting its applicability in comprehensive DNS threat analyses[136].

18. **HIKARI 2021 (2021, Ferriyan et al.)** - Focuses on encrypted synthetic attacks and normal traffic for CMSs like WordPress. he dataset utilizes Zeek for feature extraction and concentrates on specific CMS platforms, which might limit its broader application and reduce its utility in diverse cybersecurity environments[137].

19. **SJTU – AN21 (2021, Shanghai Jiao Tong University)** - Analyzes anonymity network traffic from networks like Tor and I2P, aiming to enhance traffic classification. The dataset is specific to anonymity networks, which may not generalize well to standard network environments, thereby restricting its applicability in broader network security analyses [138][139].

20. **CSTNET TLS 1.3 (2021, Xinjie Lin, Gang Xiong)** - Targets advanced TLS 1.3 encryption, aiding in refining traffic classification techniques. The dataset is specific to TLS 1.3, which may not cover older encryption protocols, potentially limiting its applicability in environments where legacy protocols are prevalent[54].

21. **UTMobileNetTraffic2021(2021, University of Texas at Austin)** - Focuses on modern mobile traffic analysis using machine learning models to classify application-specific behaviours. The dataset contains over 29 hours of encrypted traffic generated from 16 popular mobile applications, providing rich insights into application and activity-level behaviours without including direct attack scenarios. UTMobileNetTraffic2021 does not encompass any traffic related to explicit malicious activities or attack vectors, which may limit its direct applicability in security anomaly detection studies. However, with its accurate activity-level labels and diverse application coverage, this dataset is an excellent resource for developing and refining traffic classification algorithms tailored to encrypted mobile traffic [140].

22. **Dataset 7cz (2022, MH Pathmaperuma, Y Rahulamathavan)**- Focuses on mobile app traffic, using advanced tools to capture detailed encrypted traffic flows. The dataset concentrates on mobile apps, potentially limiting its utility for broader network analysis and reducing its relevance in non-mobile networking contexts[22].

23. **CESNET-QUIC (2022, CESNET, Czech Republic)** - Focuses on extensive QUIC protocol traffic analysis within an ISP network infrastructure. This dataset comprises one month of QUIC traffic data collected across backbone lines serving about half a million users, providing a rich basis for real-world encrypted traffic

classification research. The dataset includes detailed packet metadata sequences and user agent information, facilitating in-depth studies into QUIC deployment and behaviour in diverse network scenarios. A notable limitation of the CESNET-QUIC22 dataset is its focus solely on benign traffic, which may impact its use in security-focused analyses that require malicious traffic patterns. However, its comprehensive coverage of QUIC traffic and extensive metadata make it an invaluable resource for network traffic analysis and encrypted traffic studies[141].

24. **AppClassNet (2022, Huawei Technologies France SASU)** - This dataset is a comprehensive tool for application identification research, modelled after the ImageNet challenges in computer vision. It comprises over 10 million flows each labeled with one of 500 fine-grained application labels using a commercial-grade DPI system. AppClassNet provides a unique resource for developing and testing machine learning models in traffic classification, particularly in real-world conditions that challenge existing methodologies. The dataset is fully encrypted and anonymized to protect user privacy and business-sensitive information, maintaining an excellent balance between utility for research and data protection. Despite these anonymizations, it preserves enough real-world characteristics to be a reliable benchmark for algorithmic research [142].

25. **Pure TLS (2022, X Li, J Xie, Q Song)** - Isolates TLS encrypted traffic for detailed analysis, critical for traffic classification systems. The dataset exclusively focuses on TLS encrypted traffic, which does not accommodate analysis of non-TLS encrypted traffic, potentially limiting its comprehensiveness in encrypted traffic studies [65].

26. **5GAD-2022 (2022, Cooper Coldwell)** - Aims at detecting 5G network attacks, crucial for securing 5G infrastructures. The dataset is specific to 5G networks, which may not address other types of networks and attack scenarios, limiting its applicability in broader network security assessments[143].

27. **CIC IoT 2023 (2023, Canadian Institute for Cybersecurity)** - Examines network security threats for IoT devices, capturing traffic from 105 devices under 33 attack scenarios. The dataset may not comprehensively cover non-IoT network environments, which could limit its utility in broader network security studies[144].

28. **VPN/NonVPN Network Application Traffic Dataset (VNAT) (2023, MIT Lincoln Laboratory)** - Includes applications like Netflix and Skype in VPN and non-VPN settings. The dataset is primarily educational and might not adequately cover industrial-scale network scenarios, potentially limiting its applicability in real-world network security environments [145].

29. **Self Generated Private - OBW 30 HW19 (2023, Z Diao, G Xie, X Wang)** - Analyzes HTTPS traffic to classify web and application interactions, utilizing Selenium WebDriver and Scapy for data capture. Offers insights into encrypted communications. The dataset primarily focuses on HTTPS, which may not effectively capture other encrypted protocols, potentially limiting its applicability in comprehensive encrypted traffic analysis[146].

30. **Self Generated Private – TCP/UDP (2024,R. T. Elmaghraby, N. M. A. Aziem)** - Includes detailed packet captures emphasizing TCP for its reliability in data delivery, with extensive classification features like IP addresses and TLS versions. Aimed at training ML models for encrypted traffic analysis. The dataset focuses predominantly on TCP traffic, possibly overlooking the nuances of UDP traffic behaviour, which may limit its utility in studies requiring comprehensive protocol analysis [61].

31. **BCCC-cpacket-cloud-DDoS(2024, Shafi, MohammadMoein, Arash Habibi Lashkari, Hardhik Mohanty** - States to address challenges identified in previous datasets through a cloud infrastructure. The dataset contains over eight benign user activities and 17 DDoS attack scenarios. The dataset is fully labelled (with 26 labels) with over 300 features extracted from the network and transport layers of the traffic flows using NTLFlowLyzer. While the dataset addresses challenges from previous studies through cloud infrastructure and provides feature extraction and extensive labelling, its specific concentration on benign user activities and DDoS attack scenarios may not fully represent the broader range of security threats encountered in diverse network environments[147].

The exploration of the 31 datasets in tables 2, 3, 4 reveals a wide spectrum of approaches to capture and analyze encrypted traffic, each uniquely contributing to the progress of the field. From focusing on specific encrypted protocols to addressing gaps in feature extraction and labelling, these datasets provide invaluable resources for the development of robust traffic classification models. However, despite their contributions, the limitations observed—such as a lack of coverage for diverse encryption protocols or scenarios—highlight the need for continuous dataset improvement. Future efforts should aim to create more comprehensive datasets, encompassing a broader range of protocols, attack types, and real-world conditions, to further strengthen the foundation for encrypted traffic analysis and enhance the reliability of emerging solutions.

4.2 Selection of the dataset

4.2.1 Evaluation Criteria

To ensure the most relevant and high-quality datasets are selected for encrypted traffic classification, a detailed evaluation framework has been devised. Each dataset is analyzed based on specific attributes that align with the requirements of our research. The evaluation assigns binary scores (Y = 1, N/- = 0) for each criterion, with the total score representing the dataset's overall suitability for encrypted traffic classification.
This framework considers the technical attributes of the datasets based on the key columns in the analysis table. These attributes reflect critical aspects of encrypted traffic, enabling precise classification tasks. The criteria, along with their technical significance, are described below:

1. ***Traffic and Data Distribution***: The type of traffic and its distribution in a dataset are essential for evaluating its applicability to encrypted traffic classification. Datasets that exclusively feature fully encrypted traffic provide a focused environment for studying encryption patterns and behaviors, which is critical for developing classification models. However, datasets combining encrypted and non-encrypted traffic may also offer insights into mixed-traffic scenarios, allowing for comparative analysis. The distribution of benign and malicious traffic within the dataset further aids in understanding diverse traffic flows and evaluating classification boundaries.

2. ***Labeling Methodology***: The dataset's labeling strategy directly impacts its utility in classification tasks. Binary-labeled datasets are suitable for distinguishing encrypted from non-encrypted traffic, while multi-class labeled datasets enable more detailed classification into specific traffic types, such as Tor, VPN, I2P, Zeronet, and Freenet. The presence of multi-class labels facilitates advanced machine learning techniques by providing richer information for model training and testing. Well-structured labeling ensures accurate training data, reducing ambiguity and enhancing model reliability.These datasets support the classification of traffic into

Table 2: Comprehensive Analysis of Datasets Employed in ETC Research

DATASET	Traffic/ Data Distribution				Labelled		Attack Diversity												Capture	
	Fully Encrypted		Both Enc. & Non-Enc. Traffic		BINARY	MULTI	Name of Attack											#ATTACKS	COMPLETE	PARTIAL
	BENIGN	MALICIOUS	BENIGN	MALICIOUS			BRUTE FORCE	DoS/ DDoS	WEB BASED	INFILTRATION	C2/ BOTNET	HEARTBLEED	RECONNAISSANCE	SPOOFING	DNSEXFIL.	N/W RECONFIG	BOT	BLOCKCHAIN		
1	N		Y	Y	N	Y	N	Y	N	N	Y	N	N	N	N	N	N	N	13	Y	N
2	Y	N	N		N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
3	N		Y	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
4	N		Y	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
5	N		Y	Y	N	Y	N	Y	Y	N	Y	N	N	N	N	N	Y	N	10	Y	N
6	Y	N	N		Y	N	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
7	N		Y	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
8	N		Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	16	Y	N
9	N		Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
10	Y	N	N		N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
11	Y	N	N		N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
12	N	N	Y	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
13	Y	N	N		N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
14	N		Y	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
15	Y	N	N		N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
16	N		Y	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
17	N		Y	Y	Y	N	N	N	N	N	Y	N	N	N	Y	N	N	N	12	Y	N
18	Y	N	N		N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
19	Y	Y	N		N	Y	Y	N	N	N	N	N	Y	N	N	N	N	Y	3	Y	N
20	N		Y	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
21	N		Y	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
22	Y	N	N		N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
23	Y	N	N		N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
24	Y	N	N		N	Y	N	N	N	N	N	N	N	N	N	N	N	N	**-**	Y	N
25	Y	N	N		N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
26	N		Y	Y	N	Y	N	Y	N	N	N	N	Y	N	N	Y	N	N	10	Y	N
27	N		Y	Y	N	Y	Y	Y	Y	N	Y	N	Y	Y	N	N	N	N	-	Y	N
28	N		Y	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
29	Y	N	N		N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
30	Y	N	N		N	Y	N	N	N	N	N	N	N	N	N	N	N	N	-	Y	N
31	N		Y	Y	N	Y	N	Y	N	N	N	N	N	N	N	N	N	N	17	Y	N

Table 3: Cont. Comprehensive Analysis of Datasets Employed in ETC Research

DATASET	Enc. Service						Protocols														Metadata			No. of
	VPN	TOR	I2P	ZERONET	FREENET	OTHERS	WEB	EMAIL	CHAT	STREAM	FTP	VOIP	P2P	DOH	5G	UDP	IRC	SSL/TLS	SSH	DB	DOCUMENTATION	PCAP	CSV	#FEATURES
1	N	N	N	N	N	N	Y	Y	Y	Y	N	Y	Y	N	N	Y	N	Y	N	N	Y	Y	N	-
2	N	N	N	N	N	Y	Y	Y	N	N	Y	N	Y	N	N	N	N	N	N	N	Y	Y	N	-
3	N	Y	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	Y	Y	Y	28
4	Y	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	N	N	Y	N	Y	Y	N	Y	Y	Y	24
5	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	N	N	N	Y	Y	N	Y	Y	Y	N	-
6	N	N	N	N	N	N	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y	N	-
7	N	Y	Y	N	N	N	Y	Y	Y	Y	N	Y	Y	N	N	Y	Y	Y	N	N	Y	Y	N	82
8	N	N	N	N	N	Y	Y	Y	Y	Y	Y	N	N	N	N	Y	N	Y	Y	N	Y	Y	Y	83
9	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	Y	N	-
10	Y	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N	Y	Y	N	Y	Y	N	133
11	Y	N	N	N	N	N	Y	Y	Y	Y	N	Y	Y	Y	N	Y	N	Y	N	N	Y	Y	N	-
12	Y	Y	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	N	N	Y	N	Y	Y	N	Y	Y	Y	83
13	N	N	N	N	N	N	Y	N	N	Y	N	N	N	N	N	N	N	N	N	N	Y	N	N	-
14	N	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	Y	Y	Y	83
15	N	N	N	N	N	N	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	N	N	N	Y	N	N	-
16	N	N	N	N	N	N	Y	-	-	-	-	-	-	-	-	-	-	Y	-	N	Y	Y	N	-
17	N	N	N	N	N	N	Y	N	N	N	N	N	N	Y	N	Y	N	Y	N	N	Y	Y	Y	34
18	Y	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N	Y	N	N	Y	Y	N	13
19	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	Y	N	N	Y	N	Y	84
20	N	Y	Y	N	N	Y	Y	N	Y	Y	Y	N	Y	N	N	N	Y	N	N	N	Y	N	Y	83
21	N	N	N	N	N	N	Y	N	N	Y	Y	N	N	N	N	Y	N	Y	Y	N	Y	Y	N	-
22	N	N	N	N	N	Y	Y	Y	Y	N	N	N	N	N	N	-	N	Y	N	N	Y	N	Y	7
23	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N	Y	N	N	Y	Y	Y	24
24	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	**N**	Y	N	N	Y	Y	N	**-**
25	N	N	N	N	N	N	Y	N	Y	Y	N	N	N	N	N	N	N	Y	N	N	Y	Y	N	-
26	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	Y	Y	N	-
27	N	N	N	N	N	N	Y	Y	N	N	N	N	N	N	N	Y	Y	Y	Y	N	Y	Y	Y	39
28	Y	N	N	N	N	N	N	N	Y	Y	Y	Y	N	N	N	Y	N	N	Y	N	Y	Y	Y	129
29	N	N	N	N	N	Y	Y	N	Y	Y	N	N	N	N	N	N	N	Y	Y	N	Y	Y	N	-
30	N	N	N	N	N	Y	Y	N	N	Y	Y	Y	N	N	N	Y	N	Y	N	N	Y	Y	N	-
31	N	N	N	N	N	N	Y	Y	N	Y	Y	N	N	N	N	Y	N	Y	Y	N	Y	Y	Y	348

Table 4: List of Datasets Studied in ETC Research

S.No.	Dataset Name	Reference	Score
1	**CTU-13 & MCFP (Malware)-2011**	[119]	16
2	**MOORE_SET and NOC _SET-2013**	[120]	10
3	**ISCX Tor/Non-Tor-2016**	[121]	14
4	**ISCX VPN/Non-VPN-2016**	[122]	17
5	**USTC-TFC-2016**	[123]	18
6	**BetterNet HTTPS-2016**	[124]	5
7	**Anon-2017**	[125][126]	17
8	**CICIDS-2017**	[127] [128]	21
9	**Google Home -2018**	[129]	5
10	**Mirage-2019**	[130]	17
11	**Orange-2020**	[131]	15
12	**CIC-Darknet-2020**	[34]	18
13	**Oregon & Ohio HTTP2 Dataset-2020**	[132]	6
14	**TOR/I2P/Zeronet/Freenet-2020**	[133]	17
15	**UCDavis–QUIC Dataset -2020**	[134]	11
16	**Browser-2020**	[135]	7
17	**CIRA-CIC-DoHBrw -2020**	[136]	13
18	**UTMobileNet Traffic -2021**	[140]	16
19	**HIKARI-2021**	[137]	11
20	**SJTU– AN21(Tor, I2P, Jon Donym)-2021**	[138]	17
21	**CSTNET TLS 1.3-2021**	[54]	11
22	**Self Generated Public – Dataset7cz -2021**	[22]	10
23	**CESNET-QUIC22**	[141]	17
24	**AppClassNet-2022**	[142]	16
25	**Self Generated Private – Pure TLS-2022**	[65]	9
26	**5GAD dataset-2022**	[143]	11
27	**CIC IoT -2023**	[144]	20
28	**VPN/NonVPN Network Application Traffic Dataset (VNAT)-2023**	[145]	13
29	**Self Generated Private - OBW 30 HW19-2023**	[146]	11
30	**Self Generated Private – TCP/UDP- 2024**	[61]	12
31	**BCCC-cpacket-cloud -DDoS-2024**	[147]	15

specific categories such as Tor, VPN, I2P, Zeronet, and Freenet. The dataset can be Multi-class labelled or binary-labeled and the datasets score would be 1.

3. ***Attack Diversity***: Although attack-focused analysis is not the primary objective, datasets with diverse traffic behaviors—benign or otherwise—are valuable for studying general traffic patterns. Datasets featuring a variety of behaviors score 1, while more homogeneous datasets score 0. This ensures robustness in traffic classification tasks.

4. ***Capturing Details***: The capturing process, whether complete or partial, determines the dataset's reliability. Complete captures ensure continuity, while partial captures may result in missing patterns that could affect model training. Datasets capturing a wide range of network activities across different timeframes and environments are more versatile for traffic classification.Complete capture datasets score 1, while partial or incomplete datasets score 0.

5. ***Encryption Services***: Datasets that incorporate a diverse range of encryption services are particularly valuable. Encryption services like Tor, VPN, I2P, Zeronet, and Freenet represent distinct traffic patterns and behaviors, each posing unique challenges for classification models. The inclusion of multiple services in a dataset broadens its applicability, enabling models to generalize across different encrypted traffic types. Additionally, datasets with encryption services beyond the standard set provide opportunities to study emerging or less common encryption mechanisms. Datasets that include multiple encryption services receive a score of 1 for each encryption service covered, while those with limited or no encryption-specific services score 0 for each missed service. This criterion ensures that datasets support the study of varied encrypted traffic behaviours.

6. ***Protocol Coverage***: Protocol diversity in datasets is a critical factor in evaluating their comprehensiveness. Encryption protocols such as TLS, SSH, and VoIP, as well as application-layer protocols like HTTP/2, provide essential insights into traffic characteristics. The presence of multiple protocols enhances the dataset's richness, allowing the classification model to adapt to varying encryption mechanisms. This diversity ensures that the model remains robust when exposed to real-world traffic, which often involves a mixture of protocols. Datasets with a rich variety of protocols receive a score of 1 for each protocol, while those focusing on a narrow protocol set score 0 for each missed protocol. This allows the model to generalize better across different encryption mechanisms.

7. ***Metadata Availability***: Metadata plays a vital role in supporting dataset usability and reproducibility. Comprehensive documentation, including the source, methodology, and tools used for data collection, allows researchers to interpret and replicate experiments accurately. Datasets offering raw PCAP files, feature descriptions, and preprocessing scripts provide deeper insights into traffic behaviors and enable customized analysis. Metadata availability is a key indicator of a dataset's transparency and usefulness for long-term research efforts. Datasets providing extensive metadata score 1, for each subcolumn, while those with limited or no metadata score 0 for each subcolumn. The availability of metadata ensures transparency and usability in research.

8. ***Feature Set***: The number and variety of features included in a dataset significantly affect its analytical potential. Datasets with a wide range of features—spanning time-based, size-based, entropy, and protocol-specific metrics—provide a holistic view of encrypted traffic. These features enable multi-dimensional analysis,

allowing researchers to explore complex relationships and behaviors within the data. For instance, entropy-based features may highlight patterns unique to specific encryption mechanisms, while size and time behaviors aid in flow-based traffic analysis.

Each dataset is evaluated using the aforementioned criteria, ensuring its relevance to encrypted traffic classification. The columns in the analysis table—such as encryption services, protocols, metadata availability, and feature set—serve as benchmarks for determining the dataset's suitability. The scoring mechanism (Y = 1, N/- = 0) is used to systematically quantify the dataset's alignment with research goals. Datasets with higher scores are prioritized for further analysis and model development, ensuring the selected datasets meet the technical and practical requirements of the study.

4.2.2 Synthesis

The comprehensive analysis of publicly available datasets for encrypted traffic classification revealed significant insights into their strengths and limitations. These findings were instrumental in identifying the most relevant datasets for our research which align with the specific needs of both binary and multi-class classification tasks. However, the broader evaluation of the datasets highlighted several critical challenges and gaps, as detailed below:

1. ***Binary vs. Multi-Class Labeling***: Many datasets lack diversity in encryption services, often focusing on a few popular services like VPN or Tor while neglecting others such as Zeronet, Freenet, or I2P. This restricts the generalizability of classification models that need exposure to various encryption mechanisms to ensure robust performance.

2. ***Limited Encryption Service Coverage***: Many datasets lack diversity in encryption services, often focusing on a few popular services like VPN or Tor while neglecting others such as Zeronet, Freenet, or I2P. This restricts the generalizability of classification models that need exposure to various encryption mechanisms to ensure robust performance.

3. ***Incomplete Protocol Representation***: Protocol diversity is essential for creating versatile classification models. However, several datasets offer limited protocol coverage, focusing primarily on common protocols such as TLS and SSH while neglecting others like VoIP, HTTP/2, or FTP. This narrow scope impacts the adaptability of models to handle real-world encrypted traffic.

4. ***Metadata Deficiencies***: A significant challenge is the lack of detailed metadata in many datasets. Essential components such as raw packet captures (PCAP files), comprehensive documentation, and feature descriptions are often missing, making it difficult to replicate experiments or conduct in-depth analyses.

5. ***Incomplete Feature Sets***: Many datasets fail to include comprehensive feature sets, limiting their analytical utility. Features like entropy, time-based metrics, and size-based metrics are critical for identifying patterns unique to encrypted traffic. The absence of these features hinders the development of robust classification models.

6. ***Partial or Incomplete Data Captures***: Some datasets provide only partial captures, resulting in incomplete traffic patterns that may affect the training and testing of classification models. Complete traffic captures are vital for understanding the full scope of encrypted and non-encrypted behaviors.

7. ***Scalability and Data Volume***: Limited data volumes in certain datasets restrict their applicability in larger or more complex traffic scenarios. This inadequacy impacts the ability to train machine learning models effectively, particularly those requiring extensive data for deep learning approaches.

8. ***Inconsistent Labeling Practices***: Labeling inconsistencies across datasets create challenges for supervised learning models. Ambiguities or errors in labeling can lead to unreliable models with increased false positive or false negative rates, reducing their effectiveness in real-world scenarios.

4.3 Augmented Dataset Creation

The construction of an augmented dataset is a fundamental step in developing a robust encrypted traffic classification model. This section details the selection process and rationale behind the chosen datasets, emphasizing their relevance in addressing real-world challenges in encrypted traffic analysis.

The rapid proliferation of encrypted traffic necessitates more advanced analytical techniques for detection and classification. A critical aspect of this advancement is the quality and diversity of datasets used to train machine learning (ML), deep learning (DL), and hybrid models. The efficacy of these models is significantly influenced by the dataset's comprehensiveness, including the variety of encryption techniques represented, the inclusion of attack scenarios, and the presence of diverse network conditions. To ensure optimal performance, a structured evaluation of available datasets was conducted, considering factors such as encryption protocols, traffic diversity, labeling consistency, and dataset completeness.

The selection of Darknet-Dataset-2020 [133] and CIC-Darknet2020 [148] was based on key evaluation criteria based on the shortlisted datasets meeting the criteria as mentioned in Table 5 ensuring their suitability for encrypted traffic classification. Both datasets offer a diverse mix of encrypted traffic types (Tor, I2P, Freenet, ZeroNet), ensuring broad encryption service coverage and real-world applicability.

CIC-Darknet2020 provides multi-class labeling, distinguishing different encrypted traffic types, while Darknet-Dataset-2020 captures behavior-specific darknet activities, enhancing classification granularity. Both datasets contain complete traffic captures, ensuring the preservation of temporal dependencies, and include detailed metadata and a rich feature set, covering entropy, time-based, and protocol-specific characteristics.

By meeting these evaluation criteria, these datasets provide a high-quality foundation for training and testing encrypted traffic classification models, ensuring robust generalization and improved detection capabilities.

1. **CIC-Darknet2020** - This dataset is distinguished by its comprehensive representation of darknet traffic, covering multiple encrypted network protocols such as Tor and VPN. By focusing on anonymity-preserving networks, this dataset supports advancements in network forensics and cybersecurity [149, 22].

2. **Darknet-Dataset-2020** - This dataset offers a rich behavioral representation of encrypted traffic, capturing user interactions across various darknet services. This enhances the model's ability to identify distinct communication patterns within encrypted traffic, improving detection accuracy and robustness.

The inclusion of these datasets in the augmented training set ensures a holistic representation of encrypted traffic scenarios, balancing real-world applicability with methodological rigor. The integration of darknet-focused datasets strengthens the classification model's generalizability, particularly in identifying encrypted traffic anomalies and detecting privacy-focused services. Furthermore, the selection process was guided by a structured evaluation

framework, scoring datasets on encryption technique diversity, label consistency, and representation of real-world traffic conditions.

The strategic selection of CIC-Darknet2020 and Darknet-Dataset-2020 underscores a data-driven approach to optimizing encrypted traffic classification. By leveraging datasets that encapsulate a wide range of anonymity-preserving communications and darknet interactions, the proposed model is equipped to address both conventional and emerging challenges in encrypted network analysis. The augmentation process ensures that the training dataset not only represents diverse encrypted traffic types but also enhances the model's adaptability to evolving network conditions. This refined dataset serves as a strong foundation for training and evaluating the classification model, improving its effectiveness in detecting encrypted traffic patterns in dynamic cybersecurity environments.

Table 5: Shortlisted Dataset Analysis Table

	LABELLED		ENCYPTION SERVICE						CAPTURE		METADATA		
	BIN-ARY	MUL-TI	VPN	TOR	I2P	ZERO-NET	FREE-NET	OTH-ERS	COMP-LETE	PAR-TIAL	DOCUMEN-TATION	PCAP	CSV
ISCX Tor/-Non-Tor - 2016	Y	N	N	Y	N	N	N	N	Y	N	Y	Y	Y
ISCX VPN/-Non-VPN - 2016	Y	N	Y	N	N	N	N	N	Y	N	Y	Y	Y
SJTU – AN21-(Tor, I2P, Jon Donym) -2021	N	Y	N	Y	Y	N	N	Y	Y	N	Y	N	Y
Darknet-dataset-2020	N	Y	N	Y	Y	Y	Y	N	Y	N	Y	Y	Y
CIC-Darknet-2020	N	**Y**	**Y**	**Y**	N	N	N	N	**Y**	N	**Y**	**Y**	**Y**

4.4 Concluding Remarks

The careful selection and augmentation of datasets play a pivotal role in determining the effectiveness of encrypted traffic classification models. Through a structured evaluation process, the most suitable datasets were chosen based on their relevance, diversity, and ability to provide comprehensive coverage of encrypted traffic behaviors. Augmentation techniques were applied where necessary to enhance the dataset's robustness, ensuring a balanced representation across different encrypted traffic categories. Furthermore, NTL Flowlyzer was employed to generate CSV representations, standardizing feature extraction for seamless integration into the classification pipeline. The resulting dataset forms a strong foundation for training and evaluating the proposed model, optimizing its generalizability and effectiveness in real-world scenarios.

This chapter conducted an in-depth assessment of 31 publicly available datasets, revealing significant gaps in encryption service coverage, protocol diversity, feature richness, and metadata documentation. These insights shaped the selection of CIC-Darknet2020 and Darknet-Dataset-2020 as the foundation for model training. By augmenting these datasets, the study ensured multi-class label support and representation of diverse encrypted traffic services including Tor, I2P, Zeronet, and Freenet. The integration of datasets with complete captures and detailed metadata not only enriched model training but also improved generalization to real-world conditions. This rigorous dataset curation underpins the performance and explainability of the model developed in later chapters, directly contributing to its robustness, scalability, and interpretability.

The following section, **Chapter 5 - Experiments & Results**, presents a detailed analysis of the model's performance across various stages, including data preprocessing, feature selection, hyperparameter tuning, and classification results. It begins by outlining the preprocessing techniques applied to the dataset, followed by a comparative

evaluation of different feature selection methods such as Mutual Information (MI), Recursive Feature Elimination (RFE), and Principal Component Analysis (PCA). The section further explores the impact of different classification models, including deep learning, machine learning, and hybrid approaches, through an extensive performance evaluation. Additionally, hyperparameter tuning strategies, model optimizations, and explainability analyses using SHAP and LIME are discussed in detail. These insights contribute to refining the final model and enhancing its ability to accurately classify encrypted traffic.

5 Experiments & Results

This chapter presents the experimental results of the proposed model for encrypted traffic classification. The experiments are designed to evaluate the effectiveness of the model across various stages, starting with the preprocessing techniques applied to the raw dataset and progressing through feature selection, classification, and interpretability analysis. The goal of these experiments is to validate the model's ability to accurately classify traffic into encrypted and non-encrypted categories, as well as its capability to distinguish between different types of encrypted traffic.

Experimental Process

The experimental framework is designed as follows:

1. **Data Pre-processing:** This subsection details the techniques used to clean and prepare the raw dataset for analysis, ensuring data quality and consistency.
2. **Feature Selection:** This subsection highlights the selection of relevant features, including the integration of attention-enhanced features and their significance as determined by SHAP and LIME.
3. **Performance Evaluation:** This subsection displays how the classification results are assessed using various evaluation metrics. The performance of the proposed model is compared against baseline models to demonstrate its effectiveness.
4. **Hyperparameter Tuning:** This subsection details the hyperparameter tuning - RandomizedSearchCV for LightGBM and Self-Attention Layer Hyperparameter Tuning to determine the best parameters for the LGBM Model and Attention Layers respectively.
5. **Model Optimizations and Final Results:** This subsection displays how the insights gained from SHAP and LIME analyses are used to optimise further the model, including feature pruning and decision threshold adjustments. The final layer-wise results of the classification model are also displayed.

The results of these experiments provide a comprehensive understanding of the proposed model's capabilities and its potential for real-world application. Each subsection delves into the technical details, supported by quantitative and qualitative results, to validate the proposed methodology.

5.1 Data Pre-processing

The data preprocessing phase of the proposed encrypted traffic classification model focuses on cleaning, transforming, and structuring the dataset to ensure its compatibility with machine learning tasks. The process begins with loading the dataset, retaining only relevant columns starting from the 7th index to exclude redundant or irrelevant features. Categorical columns are converted into numerical representations by calculating their string lengths, while missing or invalid values (e.g., NaN, inf, -inf) are replaced with the median of their respective features to maintain data consistency. The dataset is then separated into features (X) and labels (Y), with the latter encoded using a LabelEncoder to translate categorical labels into integer values. Feature normalization is performed using StandardScaler to standardize the scale of all features, ensuring improved model convergence. The dataset is split into training and testing sets using an 80-20 ratio, with a fixed random state for reproducibility. To address class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) is applied to generate synthetic samples for

minority classes in the training set, ensuring balanced label distribution. These preprocessing steps collectively establish a robust foundation for the subsequent stages of model training and evaluation, enhancing the reliability and performance of the classification tasks.

5.2 Feature Selection

Feature selection is a crucial step in the proposed classification framework, aimed at identifying the most informative features while reducing dimensionality. This process enhances the model's interpretability and computational efficiency by removing redundant or irrelevant features. Several techniques were explored for feature selection, including Mutual Information and others, to optimize the input dataset for classification tasks.

5.2.1 Mutual Information (MI)-Based Feature Selection

Mutual Information (MI) measures the dependency between each feature and the target variable, capturing both linear and non-linear relationships. This technique was employed to rank features based on their relevance to the classification task.

5.2.1.1 Key Configurations and Observations:

Three configurations were tested for feature selection, where the top 20, 30, and 50 features were chosen based on their Mutual Information (MI) scores. The first configuration, $K = 20$, provided a compact feature set but risked omitting critical features, leading to a slight reduction in model performance, as shown in Table 6. The second configuration, $K = 30$, struck a balance between dimensionality and information retention, yielding improved classification results (Table 7). Lastly, the $K = 50$ configuration retained a larger number of features, ensuring a more comprehensive representation of the dataset and achieving the highest classification accuracy (Table 8). These observations highlight the trade-off between feature reduction and model performance, demonstrating that while a lower K value minimizes dimensionality, retaining a higher number of features can enhance classification robustness.

Table 6: Testing Metrics (Tuned Hybrid Attention + LightGBM) with MI Results (K = 20)

Class	Precision	Recall	F1-score	Support
FREENET	0.78	0.76	0.77	33
I2P	0.85	0.88	0.86	403
TOR	0.84	0.86	0.85	264
VPN	0.99	1.00	0.99	430
ZERONET	0.96	0.91	0.93	455
Accuracy				0.91 (1585)
Macro Avg				0.88 (1585)
Weighted Avg				0.91 (1585)

5.2.2 Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) was applied as an alternative technique to remove the least important features iteratively. RFE was combined with tree-based classifiers such as Random Forest and LightGBM to determine feature importance.

Table 7: Testing Metrics (Tuned Hybrid Attention + LightGBM) with MI Results (K = 30)

Class	Precision	Recall	F1-score	Support
FREENET	0.87	0.82	0.84	33
I2P	0.84	0.88	0.86	403
TOR	0.84	0.88	0.86	264
VPN	0.99	1.00	0.99	430
ZERONET	0.96	0.91	0.93	455
Accuracy				0.92 (1585)
Macro Avg				0.90 (1585)
Weighted Avg				0.92 (1585)

Table 8: Testing Metrics (Tuned Hybrid Attention + LightGBM) with MI Results (K = 50)

Class	Precision	Recall	F1-score	Support
FREENET	0.87	0.79	0.83	33
I2P	0.85	0.93	0.89	403
TOR	0.93	0.84	0.89	264
VPN	0.98	1.00	0.99	430
ZERONET	0.97	0.93	0.95	455
Macro Avg				0.91 (1585)
Weighted Avg				0.93 (1585)

5.2.2.1 Recursive Feature Elimination (RFE) Workflow and Observations:

The RFE process involved an iterative training approach where features were ranked based on their contribution to the prediction task. The least significant features were removed at each iteration until the desired number of features remained, ensuring optimal feature selection. This method proved effective in identifying the most relevant features; however, it was computationally expensive when applied to high-dimensional augmented datasets such as CIC Darknet-2020 and Darknet-2020. Despite the computational cost, RFE demonstrated its utility in refining feature sets to enhance model performance.

5.2.3 Principal Component Analysis (PCA)

PCA was employed as a dimensionality reduction technique to transform the dataset into a lower-dimensional space. Unlike Mutual Information (MI) and Recursive Feature Elimination (RFE), which select features from the original dataset, PCA generates new composite features known as principal components. This transformation reduced dimensionality significantly while retaining approximately 95% of the dataset's variance. However, a key limitation of PCA was the reduced interpretability of features, as the transformed components lacked a direct correlation with the original feature set. Despite this drawback, PCA effectively optimized data representation while preserving the most critical information for classification.

5.2.4 Comparative Analysis of Feature Selection Techniques

Feature selection is critical in optimizing model performance while reducing computational complexity. Table 9 provides a comparative analysis of the three feature selection techniques evaluated: Recursive Feature Elimination (RFE), Principal Component Analysis (PCA), and Mutual Information (MI). Each method offers distinct advantages

and limitations. RFE identifies core features based on model importance but is computationally expensive when applied to large datasets. PCA effectively reduces dimensionality while preserving approximately 95% of the dataset's variance; however, it compromises feature interpretability by generating transformed components that lack direct correlation with the original features. Conversely, MI is a simple yet effective method for capturing non-linear relationships between features, but it does not inherently account for feature interactions.

Technique	**Advantages**	**Limitations**
Recursive Feature Elimination	Identifies core features based on model importance.	Computationally expensive for large datasets.
Principal Component Analysis	Reduces dimensionality while retaining variance.	Loses feature interpretability.
Mutual Information	Simple and effective for non-linear relationships.	May not account for feature interactions.

Table 9: Comparative Analysis of Feature Selection Techniques

5.2.5 Selected Technique

Based on experimental results, **Mutual Information (MI) with** $K = 50$ was selected for the final model, providing the best balance between simplicity, effectiveness, and compatibility with downstream tasks. The selected feature set included original and attention-enhanced features, validated through SHAP analysis to confirm their relevance. This approach ensured the model retained the most informative features while maintaining interpretability and computational efficiency.

5.3 Performance Evaluation

This subsection evaluates the performance of the proposed model by comparing its results against baseline machine learning (ML), deep learning (DL), and hybrid models. Each classifier's performance is assessed using the CIC Darknet-2020 dataset for binary classificaton, with $K = 50$ features selected via Mutual Information. The goal is to validate the proposed model's effectiveness in binary and multi-class classification tasks.

The selection of models for performance evaluation was guided by the nature of the dataset and insights derived from the comprehensive literature review. Given that the dataset is tabular, comprising structured features extracted from network traffic, the chosen models represent a diverse range of machine learning (ML), deep learning (DL), and hybrid approaches known for their effectiveness in handling such data. Traditional ML models like Support Vector Machine (SVM), Random Forest (RF), LightGBM (LGBM), XGBoost, and CatBoost were selected due to their proven track record in tabular data classification tasks, particularly in scenarios requiring robust feature handling and scalability. Deep learning models such as Fully Connected Neural Networks (FCNN), TabTransformer, and Convolutional Neural Networks (CNN) were included to explore their ability to capture complex patterns and non-linear relationships within the high-dimensional feature space. Additionally, hybrid models like CNN+RF, RF+LGBM+XGB, and Transformer+LGBM were chosen to leverage the complementary strengths of ensemble learning and neural architectures, which have shown promise in improving classification accuracy and generalizability. Including these models ensures a rigorous comparison, enabling the identification of the best-performing approach while validating the superiority of the proposed model in both binary and multi-class classification tasks.

5.3.1 Baseline Models

Several baseline classifiers were implemented to establish a comparative framework for performance evaluation. These classifiers serve as reference models to assess the effectiveness of the proposed approach against well-

established machine learning and deep learning techniques. The selected baseline models include traditional machine learning classifiers such as Support Vector Machines (SVM), Gradient Boosting Decision Trees (GBDTs), and Random Forest (RF). Each model was trained and evaluated on the same dataset, ensuring a consistent benchmarking process. Performance metrics such as accuracy, precision, recall, and F1-score were used to quantify their effectiveness, providing a robust foundation for comparative analysis. Below are the results of the different baseline ML models and their respective parameter settings (if any).

- **Support Vector Machine (SVM):** The SVM classifier was implemented as a baseline model to establish a reference point for performance evaluation. Feature selection was performed using Mutual Information, with the top $K = 50$ features selected to optimize the classification process. While SVM demonstrated basic classification capabilities, its performance was limited compared to more advanced models. The results, as presented in Table 10, provide insights into its effectiveness and serve as a benchmark for assessing improvements achieved by other classifiers.

Table 10: Testing Data Classification Report for Support Vector Machine (SVM) with K = 50 Features where 0 is Encrypted and 1 is Non-Encrypted

Class	Precision	Recall	F1-score	Support
0	0.55	0.96	0.70	343
1	0.99	0.91	0.95	2988
Accuracy				0.91 (3251)
Macro Avg				0.77 (3251)
Weighted Avg				0.92 (3251)

- **XGBoost Classifier:** The XGBoost model was implemented with carefully tuned hyperparameters to optimize performance. The final configuration included `colsample_bytree` set to 0.6, `learning_rate` of 0.1, `max_depth` of 7, `min_child_weight` of 1, `n_estimators` set to 300, and `subsample` of 0.8. These hyperparameters were selected through extensive tuning to balance accuracy and computational efficiency. The best cross-validation accuracy achieved was 99.58%, demonstrating the model's robustness in classifying encrypted network traffic. However, despite its high performance, XGBoost was computationally intensive, requiring significant processing resources. The results are summarized in Table 12, while Table 11 describes the XGBoost hyperparameters used in this study.

- **LightGBM (LGBM):** The LightGBM classifier was implemented with optimized hyperparameters to enhance classification performance while maintaining computational efficiency. The final configuration included `colsample_bytree` set to 0.8168, `learning_rate` of 0.0423, `max_depth` of 10, `min_child_samples` of 49, `n_estimators` set to 439, `num_leaves` of 44, `reg_alpha` of 0.0051, `reg_lambda` of 0.1608, and `subsample` set to 0.9097. These hyperparameters were carefully selected to balance accuracy, computational cost, and generalization ability. LGBM demonstrated exceptional performance, achieving high classification accuracy while maintaining lower computational overhead than other boosting methods. Due to its efficiency and scalability, it emerged as a preferred choice for encrypted traffic classification. The performance results are summarized in Table 13, while Table 1 provides a detailed description of each hyperparameter and its significance in the model's optimization.

Table 11: Brief Description of XGBoost Hyperparameters

Parameter	Description
`colsample_bytree`	Determines the fraction of features randomly selected for constructing each tree.
`learning_rate`	Controls how much the model adjusts with each boosting step, affecting convergence speed and accuracy.
`max_depth`	Specifies the maximum depth of trees, influencing model complexity and potential overfitting.
`min_child_weight`	Defines the minimum sum of instance weights needed in a leaf node to prevent overfitting.
`n_estimators`	Determines the number of boosting iterations (trees) used in the ensemble model.
`subsample`	Defines the fraction of training data randomly sampled for each boosting iteration to improve generalization.

Table 12: Classification Report (Testing) for XGBoost Classifier where 0 is Encrypted and 1 is Non-Encrypted

Class	Precision	Recall	F1-score	Support
0	0.95	0.99	0.97	343
1	1.00	0.99	1.00	2988
Accuracy				0.99 (3251)
Macro Avg				0.97 (3251)
Weighted Avg				0.99 (3251)

- **CatBoost:** The CatBoost classifier was implemented using a feature set of $K = 50$ features selected through Mutual Information. This model demonstrated competitive performance, particularly in handling categorical data efficiently, making it a viable option for encrypted traffic classification. However, while CatBoost yielded promising results, further hyperparameter tuning could enhance its performance. The evaluation results are presented in Table 14, highlighting its effectiveness compared to other classifiers.

- **Random Forest (RF):** The Random Forest classifier was implemented with optimized hyperparameters to enhance its classification performance. The final configuration included `n_estimators` set to 200, `max_depth` set to None, `class_weight` set to Balanced, `min_samples_split` of 5, and `min_samples_leaf` of 1. These parameters were chosen to balance model complexity and generalization while ensuring robust handling of class imbalance. Random Forest demonstrated solid classification performance; however, it was outperformed by gradient boosting approaches such as XGBoost and LGBM, which offered superior accuracy and computational efficiency. The performance comparison is detailed in Table 16, while Table 15 shows each hyperparameter and its impact on model performance.

5.3.2 Deep Learning Models

Several DL models were tested to evaluate their effectiveness in encrypted traffic classification. These models leverage neural architectures to capture complex feature interactions and enhance classification performance. The tested models include Fully Connected Neural Networks (FCNN) and TabTransformer, which are designed to learn feature representations from structured network traffic data. Each model was trained and evaluated under

Table 13: Testing Classification Report for LightGBM where 0 is Encrypted and 1 is Non-Encrypted

Class	Precision	Recall	F1-score	Support
0	0.95	0.99	0.97	343
1	1.00	0.99	1.00	2988
Accuracy				0.99 (3251)
Macro Avg				0.97 (3251)
Weighted Avg				0.99 (3251)

Table 14: Testing Performance Report for CatBoost where 0 is Encrypted and 1 is Non-Encrypted

Class	Precision	Recall	F1-score	Support
0	0.94	0.98	0.96	343
1	1.00	0.99	1.00	2988
Accuracy				0.99 (3251)
Macro Avg				0.97 (3251)
Weighted Avg				0.99 (3251)

identical conditions, ensuring a fair performance comparison. Standard evaluation metrics were used to assess their classification effectiveness, including accuracy, precision, recall, and F1-score. Below are the results of the different DL models, along with their respective parameter settings (if any).

- **Fully Connected Neural Network (FCNN):** The FCNN model was implemented using a feature set of $K = 50$ features selected through Mutual Information. While the model exhibited moderate classification performance, it required extensive hyperparameter tuning to achieve results comparable to traditional machine learning classifiers. The complexity of tuning parameters such as learning rate, number of hidden layers, and activation functions significantly influenced its overall effectiveness. Despite its potential, FCNN did not outperform optimized tree-based models and gradient boosting approaches. The evaluation results for FCNN are presented in Table 17, illustrating its comparative performance within the classification framework.
- **TabTransformer:** The TabTransformer model was implemented to leverage its transformer-based architecture for structured tabular data. While it demonstrated the ability to capture complex feature interactions, its performance was comparatively lower than hybrid approaches that combined traditional machine learning and deep learning techniques. Despite its effectiveness in specific structured data scenarios, TabTransformer did not outperform optimized gradient-boosting models in the context of encrypted traffic classification. The evaluation results are detailed in Table 18, illustrating its comparative performance against other classifiers.

5.3.3 Hybrid Models

Hybrid models were explored to leverage the strengths of both deep learning and machine learning approaches. The tested architectures include CNN + Random Forest, Attention Mechanism + LightGBM, and a Stacking Classifier combining RF, LGBM, and XGBoost. These models integrate feature extraction and decision-based learning to enhance classification robustness and generalization. Each hybrid approach was trained and evaluated on the same dataset, ensuring consistency in benchmarking. Performance metrics such as accuracy, precision, recall, and F1-score

Table 15: Brief Description of Random Forest Hyperparameters

Parameter	Description
`n_estimators`	Specifies the number of trees in the ensemble, affecting model stability and performance.
`max_depth`	Defines the maximum depth of trees, controlling model complexity and potential overfitting.
`class_weight`	Adjusts weights for different classes to handle imbalanced datasets effectively.
`min_samples_split`	Determines the minimum number of samples required to split an internal node.
`min_samples_leaf`	Specifies the minimum number of samples required to be present in a leaf node.

Table 16: Testing Classification Report for Random Forest where 0 is Encrypted and 1 is Non-Encrypted

Class	Precision	Recall	F1-score	Support
0	0.99	1.00	1.00	564
1	1.00	1.00	1.00	4854
Accuracy				1.00 (5418)
Macro Avg				1.00 (5418)
Weighted Avg				1.00 (5418)

were employed to assess their effectiveness compared to standalone deep learning and machine learning models. Below are the results of the different hybrid models and the parameter settings (if any) used for them.

- **CNN + Random Forest:** The hybrid model combining Convolutional Neural Networks (CNN) with Random Forest (RF) was implemented using a feature set of $K = 20$ features selected through Mutual Information (Fig. 70). In the binary classification task (Layer 1), the decision-making process was predominantly driven by the Random Forest component, while CNN contributed minimally to the overall classification performance. The experimental results indicated that this hybrid approach offered limited benefits and did not outperform a standalone Random Forest model. These findings suggest that integrating CNN with RF in this context did not significantly improve feature extraction or classification accuracy.

- **Attention Mechanism + LightGBM:** Integrating a multi-head attention mechanism with LightGBM significantly enhanced feature representation, allowing the model to capture complex dependencies within the encrypted traffic dataset. The LightGBM component utilized the same optimized hyperparameters as previously tuned, ensuring consistency in evaluation. Experimental results demonstrated that this hybrid approach outperformed traditional machine learning models in binary and multi-class classification tasks, showcasing improved accuracy and generalization capabilities. The detailed performance comparison is presented in Table 19, highlighting the effectiveness of incorporating attention-based feature enhancement in encrypted traffic classification.

- **RF + LGBM + XGBoost (Stacking Classifier) (Fig. 71):** The stacking ensemble model combined Random Forest (RF), LightGBM (LGBM), and XGBoost to leverage their complementary strengths. The XGBoost and LGBM components utilized the same optimized hyperparameters as previously tuned to ensure consistency in performance evaluation. Experimental results indicated that this stacked model outperformed traditional machine learning classifiers in binary classification tasks. However, its performance in multi-class classification was suboptimal compared to other hybrid approaches. The detailed results are presented in Tables 20 and 21,

```
Random Forest Training vs. Validation Metrics:

Training Performance Metrics:
Accuracy: 0.9998
Precision: 0.9998
Recall: 0.9998
F1 Score: 0.9998

Validation Performance Metrics:
Accuracy: 0.9874
Precision: 0.9875
Recall: 0.9874
F1 Score: 0.9874

CNN Training vs. Validation Metrics:
729/729 ━━━━━━━━━━━━━━━━━━━━ 1s 793us/step

Training Performance Metrics:
Accuracy: 0.9463
Precision: 0.9488
Recall: 0.9463
F1 Score: 0.9462
182/182 ━━━━━━━━━━━━━━━━━━━━ 0s 810us/step

Validation Performance Metrics:
Accuracy: 0.9364
Precision: 0.9378
Recall: 0.9364
F1 Score: 0.9363
```

Figure 70: Testing Metrics for CNN + RF Hybrid Model

Table 17: Testing Classification Report for Fully Connected Neural Network (FCNN) where 0 is Encrypted and 1 is Non-Encrypted

Class	Precision	Recall	F1-score	Support
0	0.56	0.96	0.71	343
1	0.99	0.91	0.95	2988
Accuracy				0.92 (3251)
Macro Avg				0.78 (3251)
Weighted Avg				0.93 (3251)

Table 18: Testing Performance Report for TabTransformer where 0 is Encrypted and 1 is Non-Encrypted

Class	Precision	Recall	F1-score	Support
0	0.00	0.00	0.00	343
1	0.89	1.00	0.94	2988
Accuracy				0.89 (3251)
Macro Avg				0.45 (3251)
Weighted Avg				0.84 (3251)

illustrating its comparative effectiveness in different classification scenarios.

- **Transformer + LGBM:** The combination of Transformer models with LightGBM was evaluated for encrypted traffic classification. While this approach effectively captured complex feature interactions, it proved computationally expensive. Additionally, its classification accuracy was lower than that of hybrid models incorporating attention mechanisms. The performance comparison is detailed in Table 22, highlighting the trade-offs between computational cost and accuracy when utilizing Transformer-based models in this context.

5.3.4 Comparative Analysis

The evaluation process revealed several key insights regarding model performance and efficiency. Among the primary challenges encountered were :

- Overfitting in RF and XGBoost with small datasets.
- Dimensionality mismatches during hybrid model integration.
- Balancing training speed and accuracy for deep models.

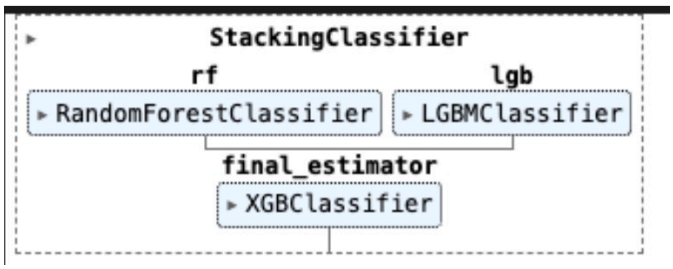

Figure 71: RF + LGBM + XGBoost Stacking Classifier

Table 19: Testing Metrics for Tuned LightGBM with Attention Mechanism where 0 is Encrypted and 1 is Non-Encrypted

Class	Precision	Recall	F1-score	Support
0	0.96	0.99	0.97	343
1	1.00	1.00	1.00	2988
Accuracy				0.99 (3251)
Macro Avg				0.98 (3251)
Weighted Avg				0.99 (3251)

Table 20: Testing Metrics for RF + LGBM + XGBoost Stacking Classifier

Class	Precision	Recall	F1-score	Support
Encrypted	1.00	0.97	0.98	2908
Non-Encrypted	0.97	1.00	0.98	2908
Accuracy				0.98 (5816)
Macro Avg				0.98 (5816)
Weighted Avg				0.98 (5816)

Despite these challenges, the proposed model, which integrates an attention mechanism with LightGBM, consistently outperformed baseline machine learning and deep learning models, achieving high accuracy, precision, and recall across all tasks.

Further analysis of hybrid models highlighted that incorporating attention mechanisms significantly enhanced the model's ability to capture complex feature relationships. However, while Random Forest and CNN-based hybrid models performed well in binary classification tasks, they exhibited limitations in multi-class scenarios. Regarding computational efficiency, LightGBM demonstrated the best trade-off between performance and processing speed, reinforcing its role as the core classifier in the proposed framework. These findings underscore the effectiveness of the proposed approach and provide insights into optimizing hybrid models for encrypted traffic classification.

5.4 Hyperparameter Tuning

The proposed model undergoes rigorous optimization to ensure its robustness, scalability, and efficiency in handling encrypted traffic classification tasks. This section outlines the steps to fine-tune the model through hyperparameter tuning, feature pruning, and iterative refinement. Hyperparameter tuning was performed for LightGBM using a Randomized Search to identify the optimal configuration for improved classification performance. Additionally, tuning was applied to the Attention Mechanism to enhance feature representation and model interpretability. Below are the hyperparameter tuning results for both components, detailing the selected configurations and their impact on performance.

5.4.1 RandomizedSearchCV for LightGBM

The hyperparameter optimization for the LightGBM classifier was conducted using `RandomizedSearchCV` to identify the best parameter configurations for both the binary and multi-class classification layers. The optimization process focused on tuning key parameters, including `learning_rate`, `max_depth`, `num_leaves`, `n_estimators`, `colsample_bytree`, and `subsample`, with the primary objective of improving the F1-score while minimizing overfitting. Cross-validation ensured that the selected hyperparameter settings were generalized effectively across different data splits.

Table 21: Overall Testing Metrics for RF + LGBM + XGBoost Stacking Classifier

Metric	Score
Accuracy	0.9838
Precision	0.9841
Recall	0.9838
F1 Score	0.9838

Table 22: Testing Metrics for Transformer + LightGBM Hybrid Model

Class	Precision	Recall	F1-score	Support
Encrypted	0.91	0.88	0.89	2908
Non-Encrypted	0.88	0.91	0.90	2908
Accuracy				0.89 (5816)
Macro Avg				0.89 (5816)
Weighted Avg				0.89 (5816)

The tuned parameters significantly improved recall for encrypted traffic classification within the binary layer, effectively reducing false negatives. In the multi-class classification layer, the optimized settings enhanced the model's ability to distinguish between challenging classes such as I2P and ZeroNet, demonstrating the impact of systematic hyperparameter tuning on classification performance.

5.4.2 Self-Attention Layer Hyperparameter Tuning

The hyperparameter tuning for the self-attention layers involved optimizing multiple components to enhance feature representation and classification performance. The multi-head attention mechanism was experimented with varying numbers of attention heads, ranging from 3 to 8, with results indicating that 3 heads provided an optimal balance between capturing dependencies and avoiding redundancy. The key, query, and value dimensions were adjusted based on the input feature sizes to ensure efficient feature transformation.

Different hidden layer dimensions (64, 128, and 256 neurons) were tested for the feedforward layers. At the same time, activation functions such as ReLU, TanH, and SeLU were evaluated to determine their impact on model convergence and performance (Tables 23 and 24). Dropout rates were tuned between 0.2 and 0.4 to introduce regularization and mitigate overfitting (Table 25), indicating that a dropout rate of 0.5 performed the best. Additionally, a warm-up learning rate scheduling strategy was implemented to stabilize the early training phase and facilitate smoother optimization. These hyperparameter refinements collectively improved the self-attention mechanism's effectiveness in encrypted traffic classification.

Table 23: Testing Metrics for Hybrid Attention-Enhanced Model with SeLU Activation

Class	Precision	Recall	F1-score	Support
Encrypted	0.93	0.97	0.95	343
Non-Encrypted	1.00	0.99	0.99	2908
Accuracy				0.99 (3251)
Macro Avg				0.96 (3251)
Weighted Avg				0.99 (3251)

Table 24: Testing Metrics for Hybrid Attention-Enhanced Model with TanH Activation

Class	Precision	Recall	F1-score	Support
Encrypted	0.92	0.97	0.95	343
Non-Encrypted	1.00	0.99	0.99	2908
Accuracy				0.99 (3251)
Macro Avg				0.96 (3251)
Weighted Avg				0.99 (3251)

Table 25: Impact of Dropout Rate on Model Performance

Dropout Rate	Accuracy	Encrypted F1-Score	Non-Encrypted F1-Score	Weighted F1-Score
0.2	0.9892	0.9504	0.9940	0.9894
0.3	0.9898	0.9531	0.9943	0.9900
0.4	0.9908	0.9571	0.9948	0.9909
0.5	0.9905	0.9558	0.9947	0.9906

5.5 Model Optimizations and Final Results

This section presents the optimization strategies and their impact on model performance for binary classification (Layer 1) and multi-class classification (Layer 2). It details the results of SHAP and LIME analyses, feature pruning strategies, and the performance evaluation of the optimized models, demonstrating their effectiveness in improving classification accuracy and interpretability.

5.5.1 Layer 1 - Binary Classification

The first layer of the proposed model is responsible for **binary classification**, distinguishing between **encrypted** and **non-encrypted** network traffic. This stage is crucial as it is the foundation for subsequent multi-class classification by filtering and preprocessing encrypted traffic before further categorization. Given the importance of this layer, the **Tuned Hybrid Attention + LightGBM model** was extensively evaluated to ensure high accuracy and reliability.

5.5.1.1 Tuned LightGBM Model Evaluation

The evaluation of the **Tuned Hybrid Attention + LightGBM model** demonstrated exceptional performance in binary classification tasks. As detailed in Table 26, the model achieved near-perfect testing metrics, with an accuracy of 99% and an F1-score of 0.98 for the **"Encrypted"** and 0.99 for the **"Non-Encrypted"** classes respectively. Additionally, **cross-validation analysis** was conducted to assess model stability across different data splits. The mean F1-score, computed over five cross-validation folds, was **0.9852**, reinforcing the model's robustness and generalization capabilities (Fig. 72). These results confirm the model's ability to effectively differentiate between encrypted and non-encrypted traffic, forming a strong foundation for subsequent classification layers.

```
********************************
Cross-Validation F1 Scores (Tuned LightGBM Model): [0.98449351 0.98516769 0.98550391 0.98381863 0.98718796]
Mean F1 Score (Tuned LightGBM Model): 0.98523434106082
********************************
```

Figure 72: Cross-validation F1 scores of the Tuned LightGBM model for binary classification

Table 26: Performance Metrics of the Tuned Hybrid Attention + LightGBM Model for Binary Classification

Class	Precision	Recall	F1-score	Support
Encrypted	0.97	0.99	0.98	1370
Non-Encrypted	0.99	0.99	0.99	3738
Accuracy				0.99 (5108)
Macro Avg				0.98 (5108)
Weighted Avg				0.99 (5108)

5.5.1.2 SHAP-Based Feature Importance and Pruning

SHAP values were computed to determine feature importance to enhance model efficiency without compromising accuracy. The analysis identified `x239`, `x87`, `x91`, and `attention_feature_22` as the most influential features, reinforcing the hybrid model's ability to integrate both original and attention-enhanced features (Fig. 73). The SHAP analysis further demonstrated that attention-derived features, such as `attention_feature_17` and `attention_feature_41`, provided complementary information, enhancing the classification of ambiguous instances. Moreover, the varied contributions of features like `attention_feature_28` and `x92` highlighted nuanced interactions, indicating that both handcrafted and learned features played a crucial role in refining classification decisions (Fig. 73). These findings underscore the effectiveness of feature pruning in optimizing model performance while retaining critical predictive attributes.

- **Interpretability Insights -** SHAP and LIME analyses were conducted to enhance interpretability and validate the model's decision-making process. The SHAP analysis provided global summary plots that reaffirmed the significance of attention features such as `attention_feature_17` and `attention_feature_41`, alongside key original features like `x92` and `x239`. Additionally, force plots highlighted specific features driving individual predictions, offering deeper insights into model behavior (Fig. 74).

  Complementing this, LIME analysis provided local explanations, demonstrating how features such as `x241` and `attention_feature_22` influenced classification decisions. This interpretability framework ensured transparency and improved trust in the model's predictions (Fig. 75).

- **Key Observations -** The hybrid attention mechanism effectively fused original and attention-generated features, significantly enhancing interpretability. Both SHAP and LIME provided consistent insights, validating the importance of key features and reinforcing the model's robustness. These findings confirm that the model achieves high classification accuracy and maintains a high degree of explainability, essential for understanding encrypted traffic classification decisions.

- **Pruning Strategy -** A pruning strategy was implemented based on SHAP feature importance values to enhance model efficiency and reduce computational complexity. Post-classification SHAP analysis was conducted to identify features that contributed minimally to the predictions. A quantile-based threshold approach was applied, retaining only the top 75% of the most influential features while discarding less relevant ones. This pruning process effectively reduced the feature set without compromising model performance, ensuring an optimal balance between interpretability and efficiency (Fig. 76).

- **Retraining and Validation -** Following the pruning process, the model was retrained using the reduced feature set to evaluate its effectiveness. Performance metrics, including F1-score and accuracy, were compared against

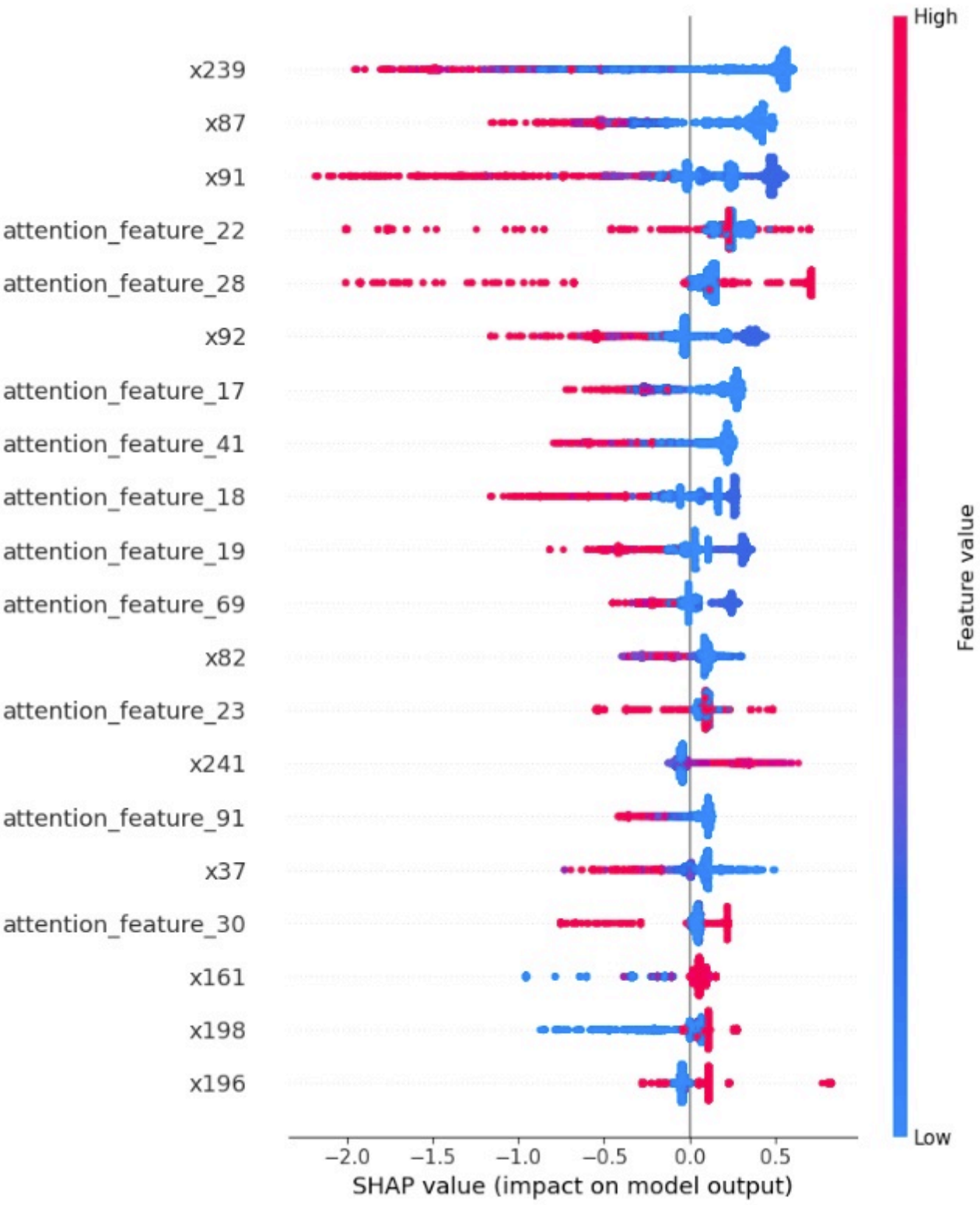

Figure 73: SHAP summary plot illustrating the impact of various features, including attention-based features, on the model's output for Layer 1

Figure 74: SHAP force plot visualizing the contribution of individual features to a specific prediction, highlighting the positive and negative feature influences on the model's output for Layer 1

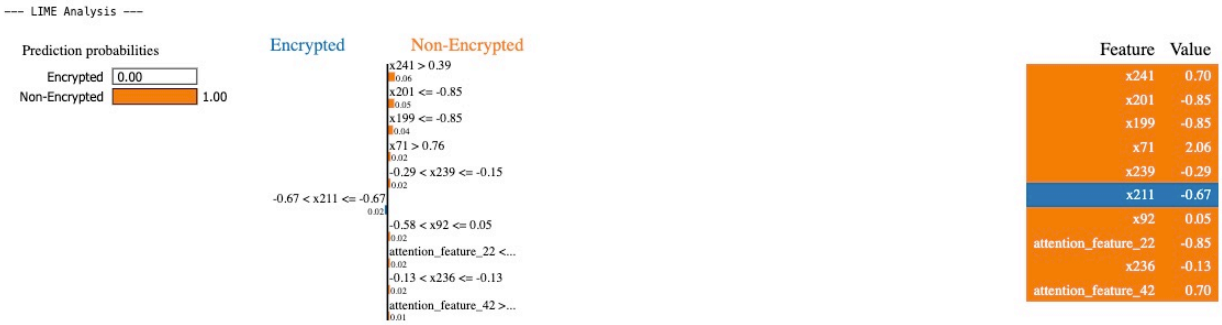

Figure 75: LIME analysis illustrating feature contributions to a specific classification decision, showing the impact of selected features on the model's predicted probabilities for Layer 1

```
--- SHAP-Based Feature Importance Pruning ---
                    Feature  Mean_SHAP_Value
41                     x239         0.547691
17                      x87         0.374157
18                      x91         0.348131
72    attention_feature_22         0.300033
78    attention_feature_28         0.287682
..                     ...              ...
152  attention_feature_102         0.000000
172  attention_feature_122         0.000000
153  attention_feature_103         0.000000
154  attention_feature_104         0.000000
199  attention_feature_149         0.000000

[200 rows x 2 columns]
```

Figure 76: SHAP-based feature importance pruning, displaying the mean SHAP values of selected features, highlighting the most and least influential features in the model.

```
Cross-Validation F1 Scores (Pruned Model): [0.98398785 0.98668462 0.98668406 0.98483003 0.98701939]
Mean F1 Score (Pruned Model): 0.9858411890250196
```

Figure 77: Testing metrics and cross-validation results for the pruned LightGBM model, demonstrating performance after SHAP-based feature pruning.

the baseline model to assess any potential impact on classification performance. The results demonstrated that the pruned model retained its predictive capability, validating the efficacy of SHAP-driven pruning in maintaining classification accuracy while reducing computational complexity. This approach confirmed that removing less influential features did not degrade performance, reinforcing the benefits of feature selection in optimizing encrypted traffic classification.

Table 27: Testing Metrics for the Pruned LightGBM Model after SHAP-Based Feature Pruning

Class	**Precision**	**Recall**	**F1-score**	**Support**
Encrypted	0.97	0.99	0.98	1370
Non-Encrypted	0.99	0.99	0.99	3738
Accuracy				0.99 (5108)
Macro Avg				0.98 (5108)
Weighted Avg				0.99 (5108)

- **Pruned Model Evaluation -** The pruned model retained only the most impactful features identified via SHAP, as detailed in Table 27 and Fig. 77. The evaluation results demonstrated that pruning did not negatively impact performance. The testing metrics remained consistent with the tuned model, maintaining 99% accuracy and an F1-score of 0.98 for the **"Encrypted"** and 0.99 for the **"Non-Encrypted"** classes, respectively. Cross-validation further reinforced these findings, with the pruned model achieving a mean F1-score of 0.9858, slightly higher than the unpruned model. These results confirm the effectiveness of SHAP-based pruning in optimizing model efficiency without sacrificing classification performance.

5.5.2 Layer 2 - Multi-Classification

The second layer of the proposed model is responsible for **multi-class classification**, where encrypted traffic is further categorized into specific encrypted services such as VPN, Tor, I2P, ZeroNet, and Freenet. This classification is more complex than the binary classification in Layer 1, as it requires distinguishing between different types of encrypted traffic with overlapping patterns. To address these challenges, the **Tuned Hybrid Attention + LightGBM model** was evaluated extensively to assess its effectiveness in handling fine-grained traffic differentiation.

5.5.2.1 Tuned LightGBM Model Evaluation

The evaluation of the **Tuned Hybrid Attention + LightGBM model** for multi-class classification demonstrated robust performance, with significant improvements attributed to the incorporation of attention-enhanced features. Integrating multi-head attention mechanisms allowed the model to capture intricate dependencies between features, leading to better discrimination across encrypted traffic types. The classification performance was assessed using key metrics, including accuracy, precision, recall, and F1-score, ensuring a comprehensive evaluation of the model's effectiveness. These results validate the capability of the hybrid model to accurately classify encrypted traffic into distinct categories accurately, further enhancing its utility in real-world cybersecurity applications.

```
********************************
Cross-Validation F1 Scores (Tuned LightGBM Model): [0.9459111  0.9365302  0.95205187 0.95071753 0.93953519]
Mean F1 Score (Tuned LightGBM Model): 0.9449491781314764
********************************
```

Figure 78: Cross-validation F1 scores of the Tuned LightGBM model for multi-class classification

Below is a comprehensive discussion of the model's key analytical processes and findings. First, we examine **SHAP and LIME-Based Feature Importance Analysis**, where SHAP and LIME techniques are utilized to interpret feature contributions - focussing on attention-based features and understanding the model's decision-making process. Next, we explore **SHAP-Based Pruning and Retraining**, concentrating on how SHAP-driven feature selection optimizes the model by reducing complexity while maintaining performance. This is followed by an analysis of **Misclassified Instances**, identifying patterns in incorrect classifications to highlight potential limitations and areas for improvement. Finally, we present **Key Insights**, synthesizing the findings from feature importance, pruning strategies, and misclassification trends to derive meaningful conclusions about model performance and interpretability.

- **Testing Metrics (Tuned Model) -** The evaluation of the tuned model demonstrated an overall accuracy of 93%, with a macro average F1-score of 0.90, indicating strong classification performance across all encrypted traffic categories. A detailed class-specific performance breakdown (Table 28) revealed variations in the precision, recall, and F1 score across different encrypted services. The model achieved an F1-score of 77% for **FREENET** (Precision = 78%, Recall = 76%), while **I2P** exhibited an F1-score of 89% with high recall (93%) and precision (86%). The classification performance for **TOR** resulted in an F1-score of 89% (Precision = 92%, Recall = 85%), whereas **VPN** demonstrated near-perfect classification with an F1-score of 99% (Precision = 99%, Recall = 100%). Similarly, **ZERONET** achieved an F1-score of 95% with precision and recall values of 97% and 94%, respectively.

  Cross-validation was performed to assess the model's generalization capability (Fig. 78), yielding a mean F1-score of 0.9445, showcasing consistent performance across multiple data splits. Additionally, the confusion matrix (Fig. 79) provided further insights into class-wise misclassifications, reinforcing the model's ability to differentiate between encrypted traffic types effectively.

Table 28: Performance metrics of the Tuned Hybrid Attention + LightGBM model for multi-class classification of encrypted traffic types, including Freenet, I2P, TOR, VPN, and Zeronet

Class	Precision	Recall	F1-score	Support
FREENET	0.78	0.76	0.77	33
I2P	0.86	0.93	0.89	403
TOR	0.92	0.85	0.89	264
VPN	0.99	1.00	0.99	430
ZERONET	0.97	0.94	0.95	455
Accuracy				0.93 (1585)
Macro Avg				0.90 (1585)
Weighted Avg				0.93 (1585)

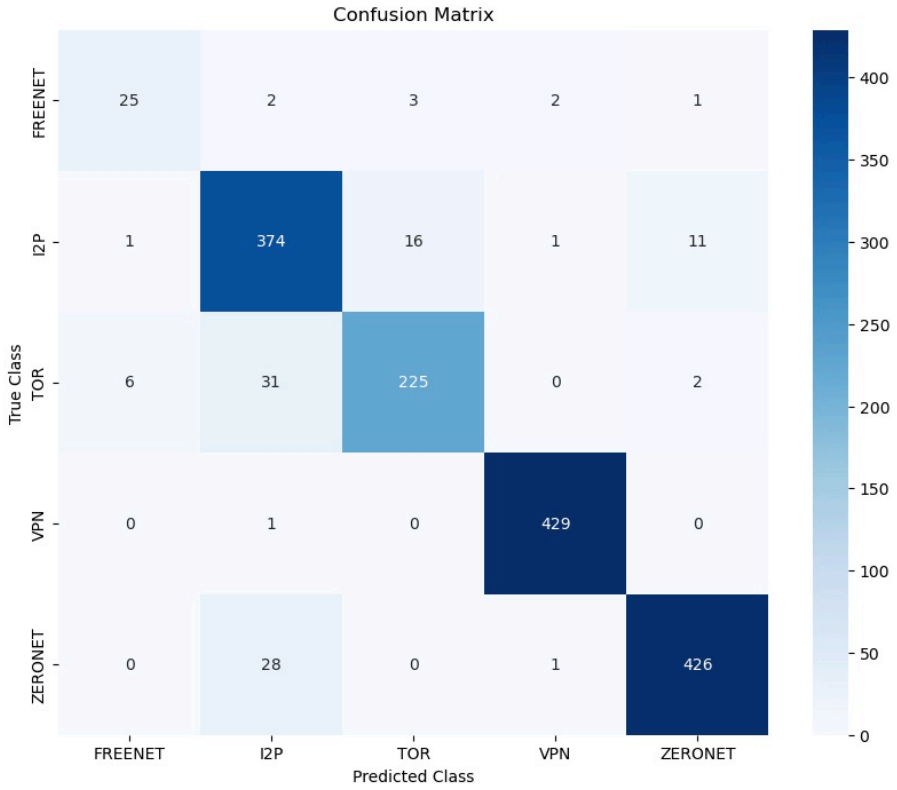

Figure 79: Confusion matrix of the Tuned Hybrid Attention + LightGBM model for multi-class encrypted traffic classification, showing the distribution of true and predicted class labels.

5.5.2.2 SHAP and LIME-Based Feature Importance Analysis

SHAP and LIME analyses were conducted on the multi-class classification model to enhance interpretability and understand the contributions of individual features. These techniques provided insights into feature significance, highlighting global and class-specific influence on model predictions. Below are the results of this analysis.

- **SHAP Analysis -** The SHAP analysis provided global insights into the most influential features across all classification tasks (Fig. 85). Features such as `x91`, `x92`, and `x199` consistently ranked among the top contributors, demonstrating their critical role in traffic classification. Additionally, attention-enhanced features, particularly `attention_feature_19` and `attention_feature_25`, significantly impacted the model's decision-making, reinforcing the effectiveness of hybrid feature extraction techniques. Feature distribution analysis further revealed that `x91` and `attention_feature_25` exhibited both positive and negative contributions, indicating adaptability across different classification contexts.

  Beyond global trends, SHAP class-specific insights (Figs. 80, 81, 82, 83, 84) highlighted the role of attention-based features in distinguishing encrypted traffic types. For **ZERONET**, `attention_feature_25` played a crucial role in differentiating it from other classes, underscoring the importance of learned representations in classification. Similarly, in the case of **I2P**, features such as `x65` and `x198` were instrumental in improving recall, particularly in distinguishing I2P from TOR and ZERONET. These findings validate the effectiveness of the hybrid model's feature extraction process and provide critical insights into how specific features influence classification outcomes.

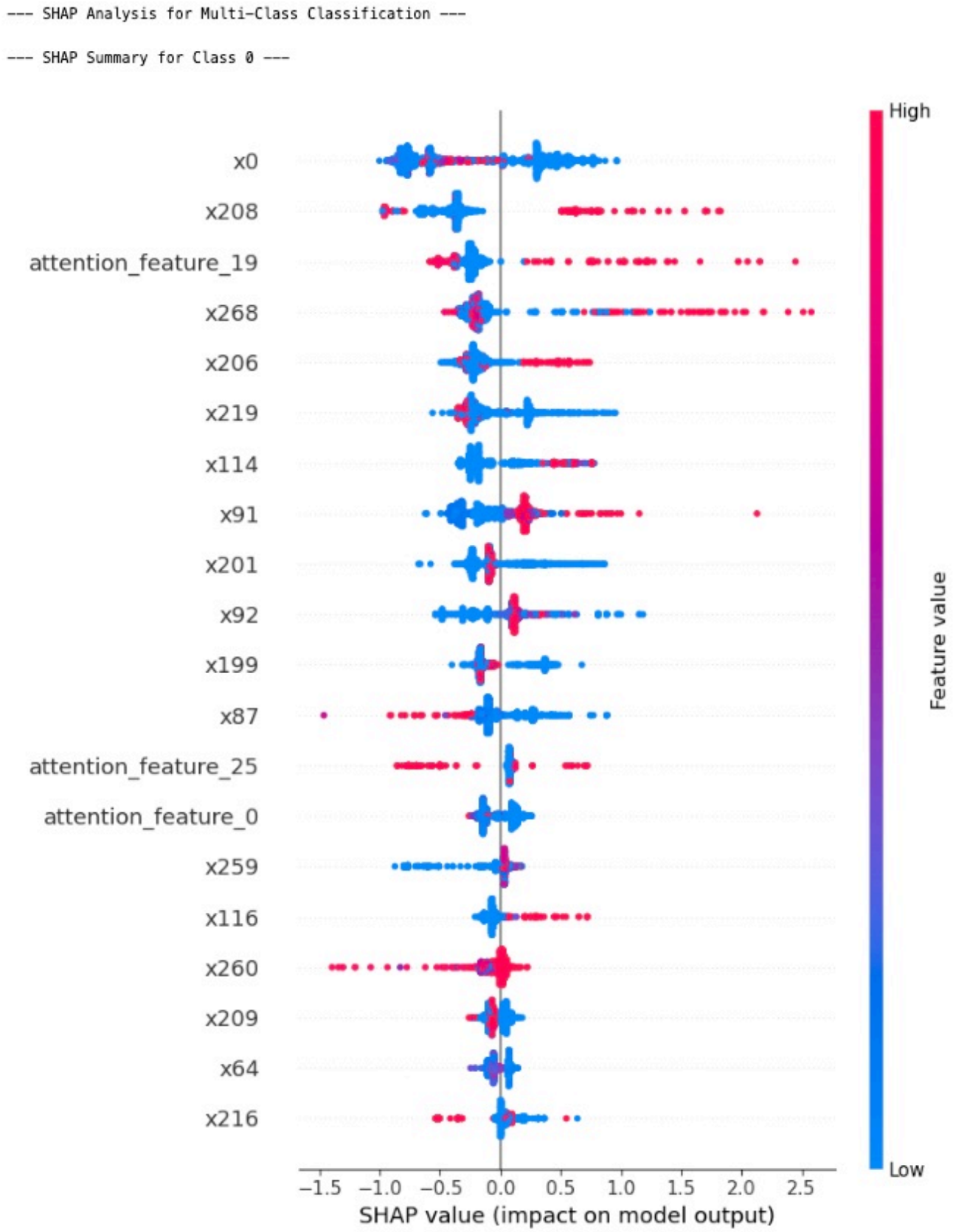

Figure 80: SHAP summary plot for multi-class classification, illustrating the impact of various features, including attention-based features, on the model's output for Class 0

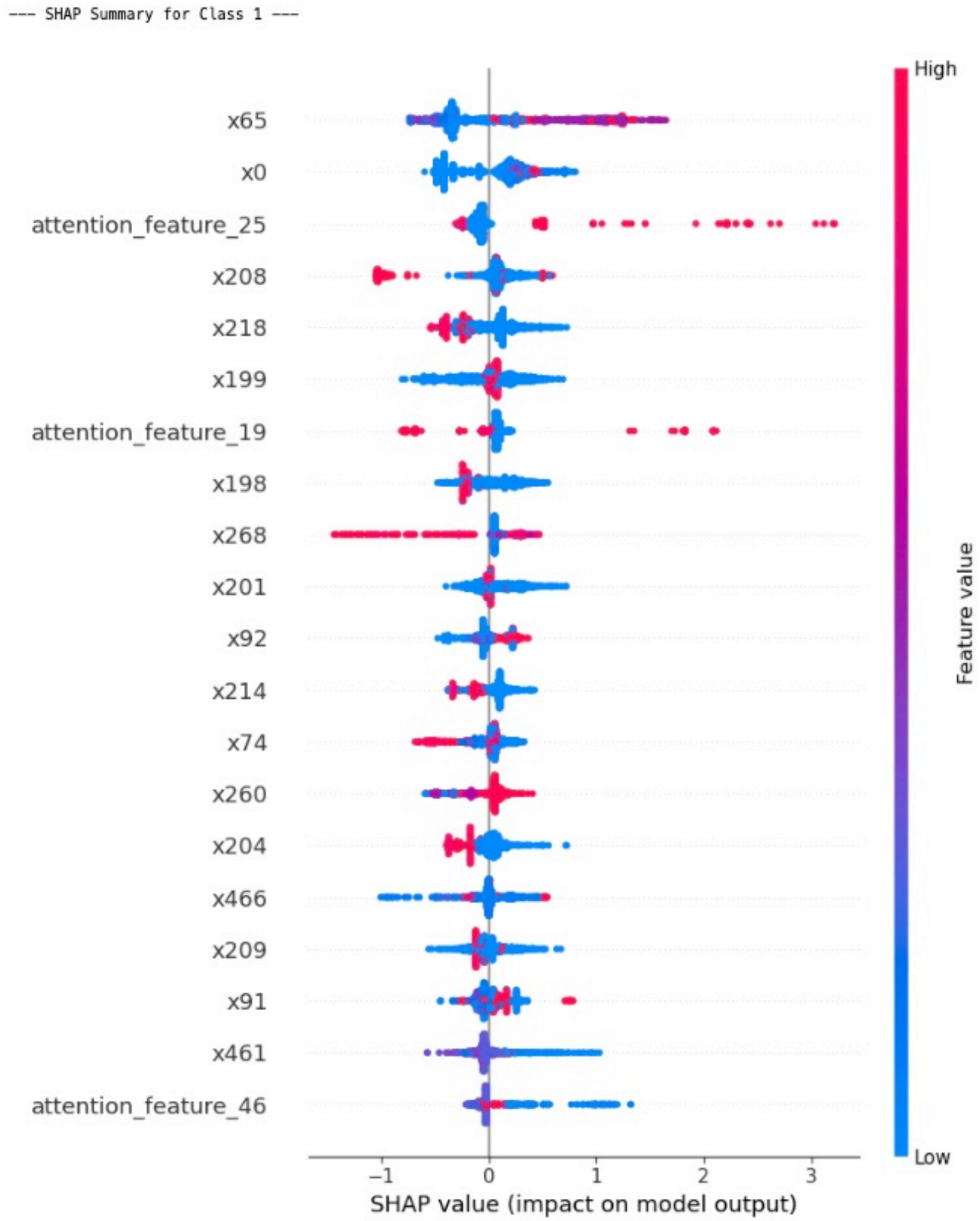

Figure 81: SHAP summary plot for multi-class classification, illustrating the impact of various features, including attention-based features, on the model's output for Class 1

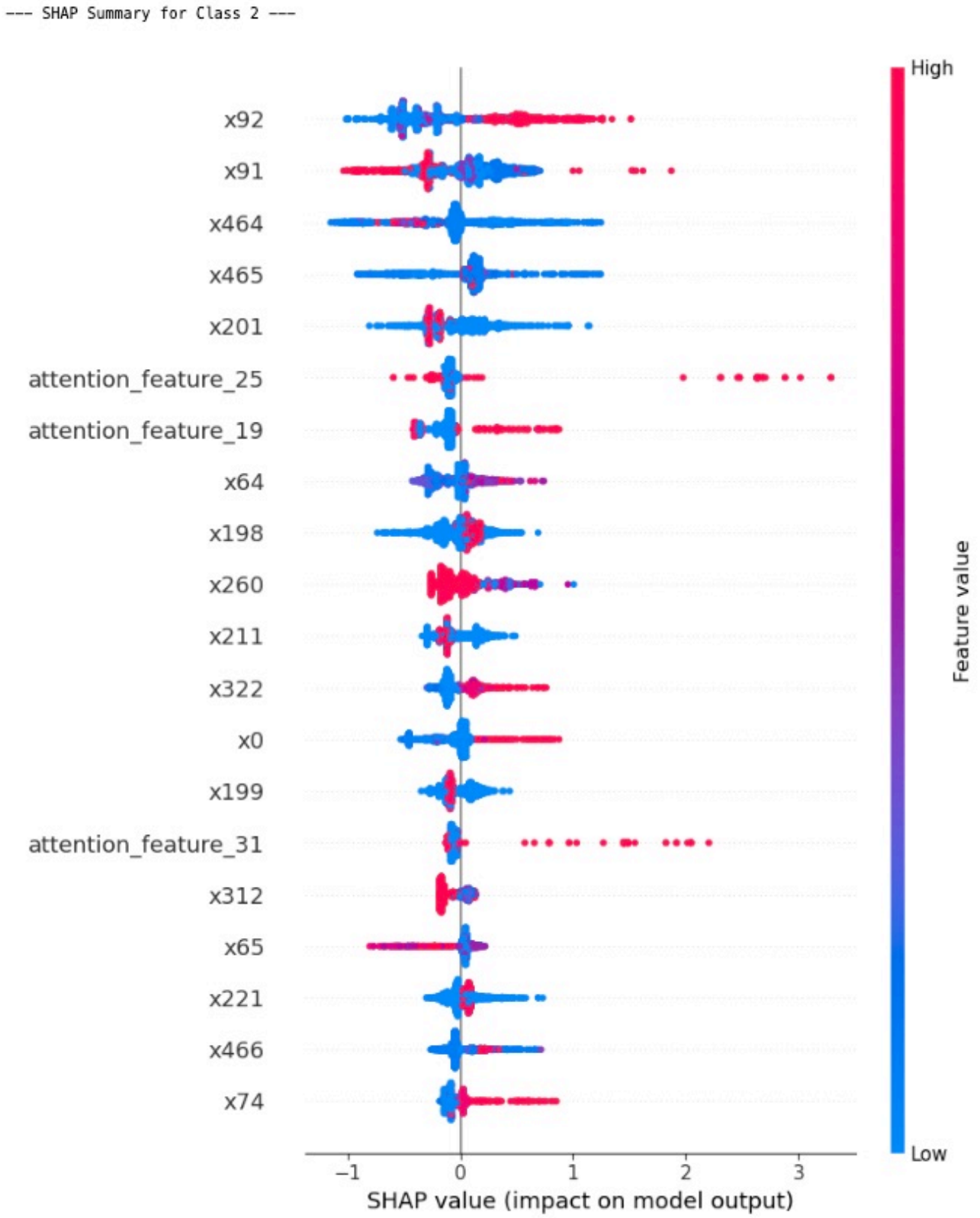

Figure 82: SHAP summary plot for multi-class classification, illustrating the impact of various features, including attention-based features, on the model's output for Class 2

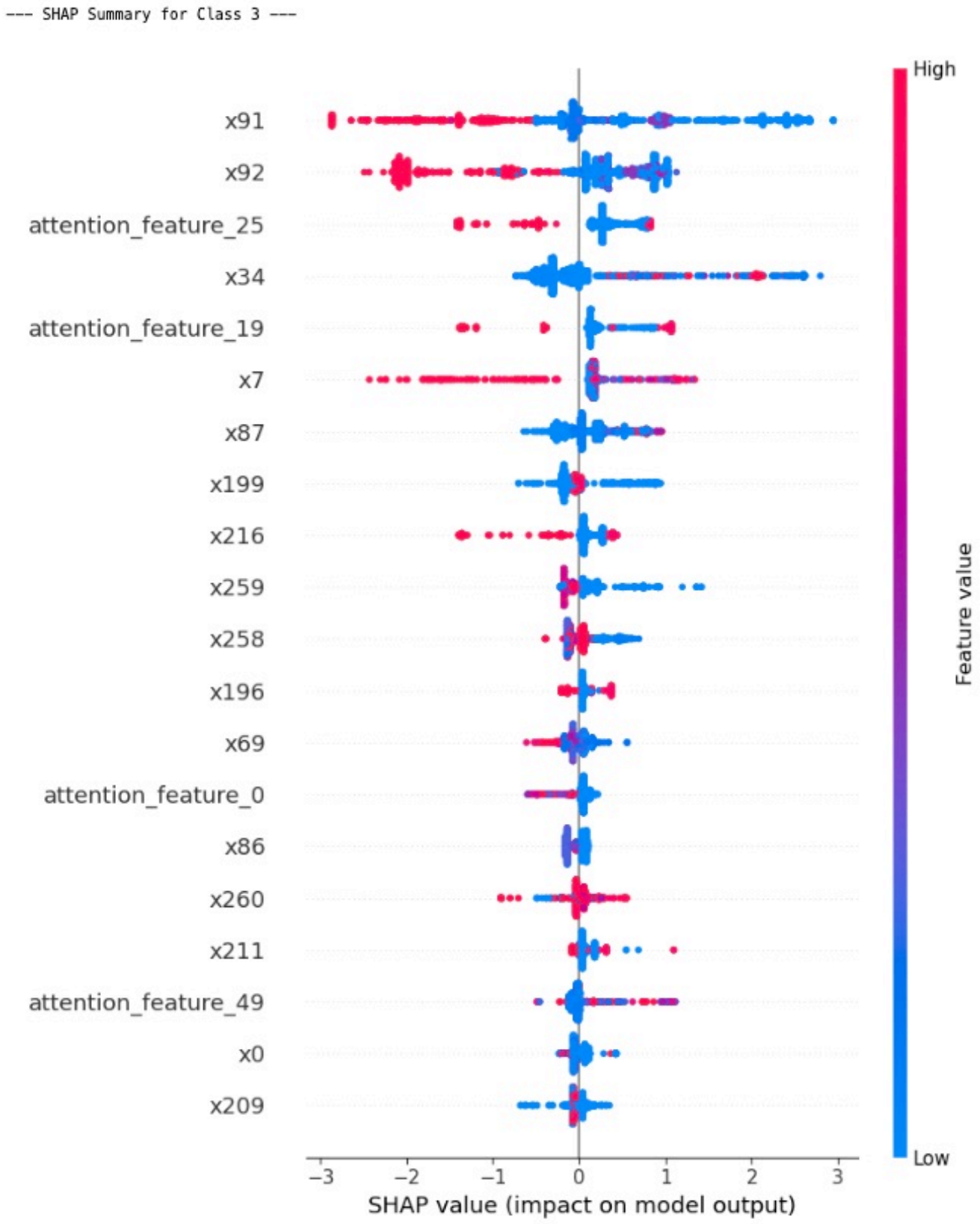

Figure 83: SHAP summary plot for multi-class classification, illustrating the impact of various features, including attention-based features, on the model's output for Class 3

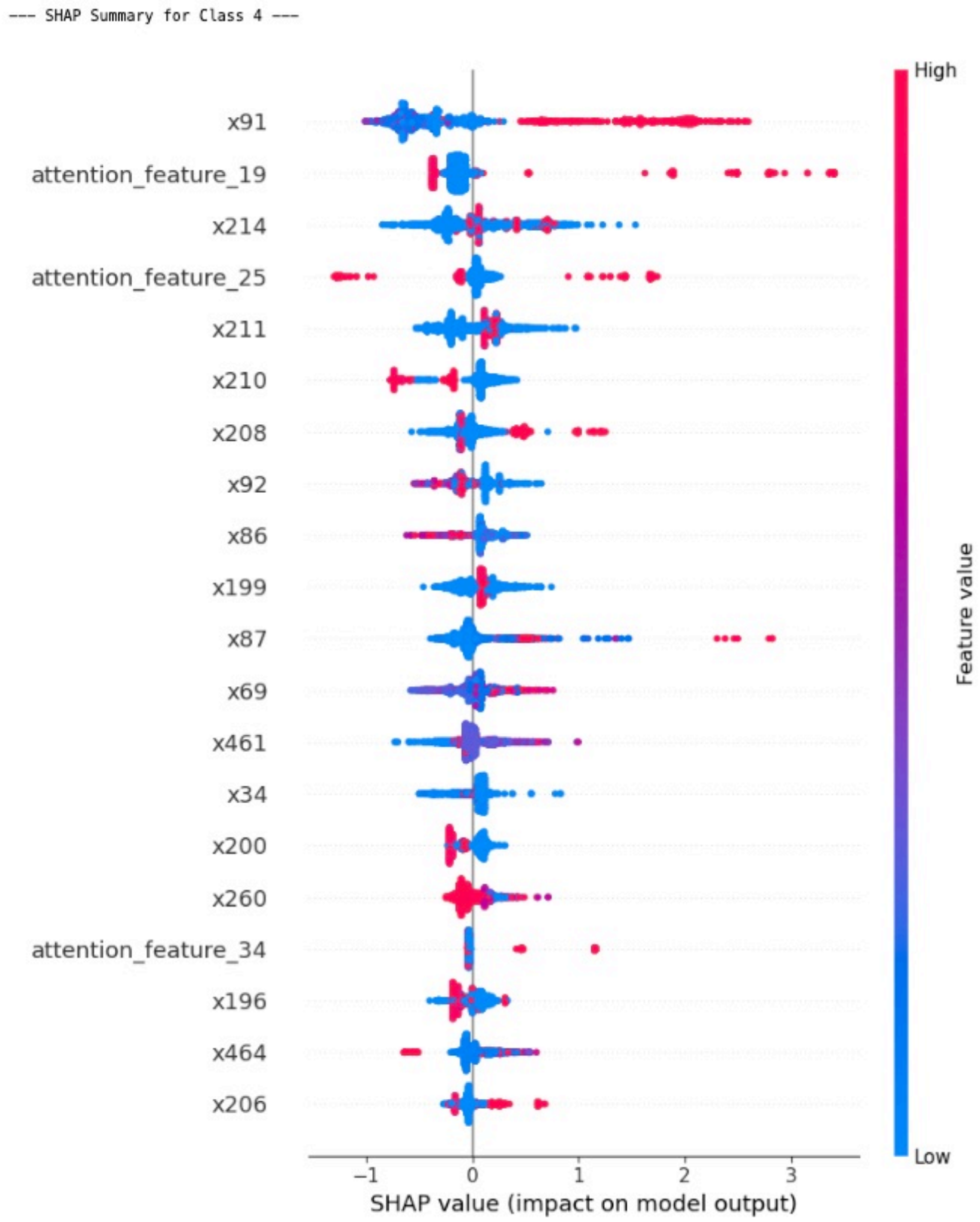

Figure 84: SHAP summary plot for multi-class classification, illustrating the impact of various features, including attention-based features, on the model's output for Class 4

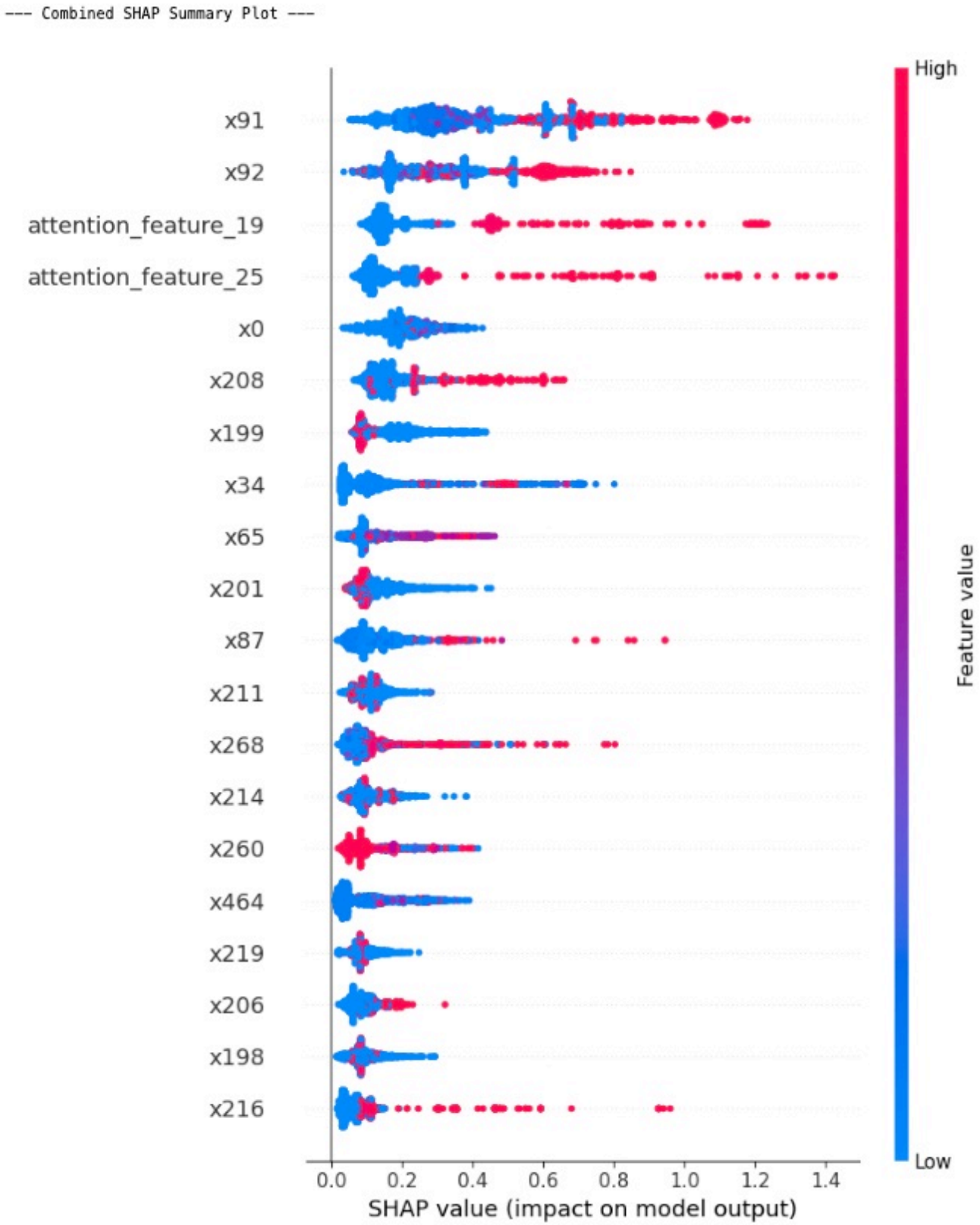

Figure 85: Combined SHAP summary plot for multi-class classification, illustrating the impact of various features, including attention-based features, on the model's output for all Classes

Figure 86: LIME Analysis for Multi-Class Classification

- **LIME-Based Interpretability -** LIME analysis was conducted to enhance model interpretability and provide local explanations for individual predictions. Below are the results of this analysis. The findings revealed that LIME effectively highlighted the importance of original and attention-enhanced features in classification decisions (Fig. 86). Specifically, the analysis demonstrated that distinguishing **I2P** relied heavily on key features such as `x198`, `attention_feature_25`, and `x65`, validating the role of hybrid feature integration in improving classification accuracy.

  Additionally, LIME visualizations provided insights into the model's confidence levels in its predictions. The analysis confirmed that high-confidence classifications were consistent with expected outcomes, reinforcing the model's reliability in distinguishing encrypted traffic categories. These results affirm that the model achieves high accuracy and maintains a high degree of transparency and interpretability, ensuring trust in its decision-making process.

5.5.2.3 SHAP-Based Pruning and Retraining

SHAP-based pruning was applied to retain only the most impactful features to enhance model efficiency without sacrificing performance. As illustrated in Fig. 87, the pruning process retained approximately 75% of the most influential features based on SHAP mean values. Features such as `x91`, `x92`, and `attention_feature_19` were preserved due to their consistently high importance scores, ensuring that critical information was maintained for classification. This pruning strategy effectively reduced feature dimensionality while retaining the model's predictive capability, optimizing computational efficiency without compromising classification accuracy.

- **Retraining Metrics (Pruned Model) -** The pruned model was retrained to evaluate the impact of SHAP-based pruning on model performance, and its classification metrics were compared against the baseline. Below are the results for this evaluation, as summarized in Table 29 and Fig. 88. The pruned model achieved an overall accuracy of 93% and a macro average F1-score of 0.90, demonstrating that feature reduction did not negatively affect classification performance.

  Class-specific performance analysis revealed that the model maintained strong predictive capability across encrypted traffic categories. For **FREENET**, the model achieved a precision of 86%, recall of 73%, and an F1-score of 79%. The classification of **I2P** resulted in an F1-score of 89%, with precision and recall values of 86% and 92%, respectively. Similarly, **TOR** achieved an F1-score of 89% (Precision = 92%, Recall = 86%), while **VPN** demonstrated near-perfect classification with an F1-score of 99% (Precision = 99%, Recall =

```
--- SHAP-Based Feature Importance Pruning ---
                  Feature  Mean_SHAP_Value
15                    x91         0.473710
16                    x92         0.359922
69    attention_feature_19         0.284753
75    attention_feature_25         0.269290
0                     x0           0.210031
..                    ...              ...
123   attention_feature_73         0.000000
121   attention_feature_71         0.000000
120   attention_feature_70         0.000000
119   attention_feature_69         0.000000
199  attention_feature_149         0.000000

[200 rows x 2 columns]
```

Figure 87: SHAP-based feature importance pruning, displaying the mean SHAP values of selected features, highlighting the most and least influential features in the model for multi-class classification

```
Cross-Validation F1 Scores (Pruned Model): [0.9474924  0.93652982 0.95100947 0.95226522 0.94113064]
Mean F1 Score (Pruned Model): 0.9456855073974882
```

Figure 88: Cross-validation results for the pruned LightGBM model, demonstrating performance after SHAP-based feature pruning for multi-class classification

100%). **ZERONET** maintained a high F1-score of 95%, with precision and recall values of 96% and 93%, respectively.

Furthermore, cross-validation results confirmed the robustness of the pruned model, with a mean F1-score of 0.9457 across multiple data splits, indicating strong generalization. These findings validate the effectiveness of SHAP-based pruning in optimizing model efficiency while preserving high classification performance.

Table 29: Performance Metrics for the pruned LightGBM model, demonstrating performance after SHAP-based feature pruning for multi-class classification

Class	**Precision**	**Recall**	**F1-score**	**Support**
FREENET	0.86	0.73	0.79	33
I2P	0.86	0.92	0.89	403
TOR	0.92	0.86	0.89	264
VPN	0.99	1.00	0.99	430
ZERONET	0.96	0.93	0.95	455
Accuracy				0.93 (1585)
Macro Avg				0.90 (1585)
Weighted Avg				0.93 (1585)

5.5.2.4 Misclassified Instances

The evaluation of the multi-class classification model revealed a total of 106 misclassified instances. Below are the results of this analysis. These misclassifications were identified by comparing the predicted and true labels,

	True Label	Predicted Label	Max Probability	All Probabilities
0	2	0	0.495940	[0.4959400618492123, 0.44503451059979393, 0.04...
1	2	1	0.525293	[0.0002980940593384931, 0.5252930748901985, 0....
2	2	1	0.878225	[4.776906263530134e-05, 0.8782250856565622, 0....
3	0	2	0.702519	[0.1002719037329642, 0.10586761928208703, 0.70...
4	1	2	0.680902	[0.0004387405459432551, 0.3078110578437124, 0....

Figure 89: Prediction probabilities comparison showing the true labels, predicted labels, and corresponding probability scores for multi-class classification

highlighting cases where the model failed to classify encrypted traffic types correctly. The true labels of the misclassified instances were distributed across different classes, with values such as [2, 2, 2, 0, 1, 4, 2, 2, 4, 1, ...], whereas the corresponding predicted labels deviated from their true counterparts, with values such as [0, 1, 1, 2, 2, 1, 1, 1, 1, 4, ...].

A deeper examination of these errors suggested that certain encrypted traffic types exhibited overlapping feature characteristics, leading to misclassification in specific cases. For instance, confusion was observed between classes **I2P and TOR** and **FREENET and ZERONET**, indicating potential similarities in feature representations that affected the model's ability to distinguish them accurately. Additionally, the prediction probability distributions (Fig. 89) provided further insights into the confidence levels associated with misclassified samples, demonstrating instances where the model assigned nearly equal probabilities to multiple classes, leading to classification errors. These findings emphasize the need for further refinement in feature selection and class-specific thresholds to mitigate ambiguities and enhance classification performance.

In-Depth Analysis of Misclassified Instances

This section examines misclassified instances to identify patterns, potential causes, and areas for model improvement. By leveraging interpretability techniques such as LIME and SHAP, the analysis provides insights into feature contributions, class-specific challenges, and adjustments needed to enhance classification accuracy. Below are the detailed insights and observations from the studies, highlighting key findings and their implications.

- **Observations from SHAP Analysis -** A detailed investigation of misclassified instances was conducted using SHAP analysis to identify key features contributing to classification errors. Below are the results of this analysis. The SHAP values provided valuable insights into the dominant features influencing predictions, revealing that `x91` and `x92` consistently exhibited the highest SHAP values (Fig. 90). Their strong influence on model decisions underscores their critical importance in classification.

  Additionally, attention-enhanced features such as `attention_feature_19` and `attention_feature_25` demonstrated significant impact, capturing complex non-linear dependencies that contributed to resolving ambiguities between overlapping encrypted traffic classes. The SHAP summary plots further highlighted polarized impacts of `x91` and `x92` across different courses, illustrating their varying influence on classification outcomes (Fig. 91). These findings validate the effectiveness of the hybrid attention mechanism in feature

extraction while indicating areas where further refinement could help mitigate misclassification.

- **Observations from LIME Analysis -** A detailed LIME analysis was conducted to further interpret the misclassified instances and validate the SHAP findings. Below are the key insights derived from this evaluation. The LIME analysis reinforced the importance of locally significant features, highlighting that `x198` and `x65` played a crucial role in classification decisions (Fig. 92). These results closely aligned with SHAP findings, confirming the consistency of feature importance across different explainability techniques.

  Additionally, the analysis revealed instances of class confusion, particularly between **I2P and ZERONET**, where overlapping feature distributions contributed to misclassification. This observation emphasizes the need for more robust class-specific feature engineering to improve model differentiation. To gain deeper insights, LIME explanations were visualized for the first few misclassified instances, illustrating the specific features that influenced incorrect predictions (Fig. 92, Fig. 93, Fig. 94). These visualizations provided a more granular understanding of the model's decision boundaries, offering opportunities for targeted refinements in feature selection and classification thresholds.

- **Recommendations for Optimization -** To further refine the model's performance and address observed classification challenges, several optimization strategies are proposed. Below are the key recommendations based on the detailed SHAP and LIME analyses.

  First, a **visual analysis** of feature distributions is recommended to understand class separability better. Plotting the distributions of the top features, such as `x91`, `x198`, and `x65`, across all classes can help identify overlapping patterns and inconsistencies in feature contributions. Additionally, pairwise feature scatterplots can be employed to visualize interactions between features, providing deeper insights into how certain classes can be better differentiated.

  Another critical optimization involves **class-specific augmentation** techniques, such as the Synthetic Minority Over-sampling Technique (SMOTE), which should be applied to underrepresented classes. This approach can mitigate class imbalance and enhance the model's ability to learn meaningful patterns from minority classes.

  Furthermore, **reducing VPN threshold and weight** in training, as the large number of VPN labels introduces bias, potentially affecting the model's ability to generalize across other classes. The model can achieve a more balanced representation across all traffic types by adjusting class weights.

  Finally, **defining class-specific thresholds** is essential for improving classification accuracy in highly confused classes, such as I2P and ZERONET. The decision thresholds should be dynamically adjusted based on performance metrics—reducing the threshold for I2P to enhance recall while increasing it for ZERONET to improve precision. The model can better balance sensitivity and specificity by setting default thresholds at 0.5 for all classes and fine-tuning them based on misclassification trends.

  The model is expected to achieve improved classification performance by integrating these optimizations while maintaining high interpretability. These adjustments, informed by feature importance and explainability techniques, ensure that refinements are data-driven and aligned with the model's decision-making framework.

5.5.2.5 Key Insights

The extensive evaluation of the **Hybrid Attention + LightGBM** model reveals significant performance enhancements driven by targeted optimization techniques and in-depth post-hoc analysis. Below are the key findings from this

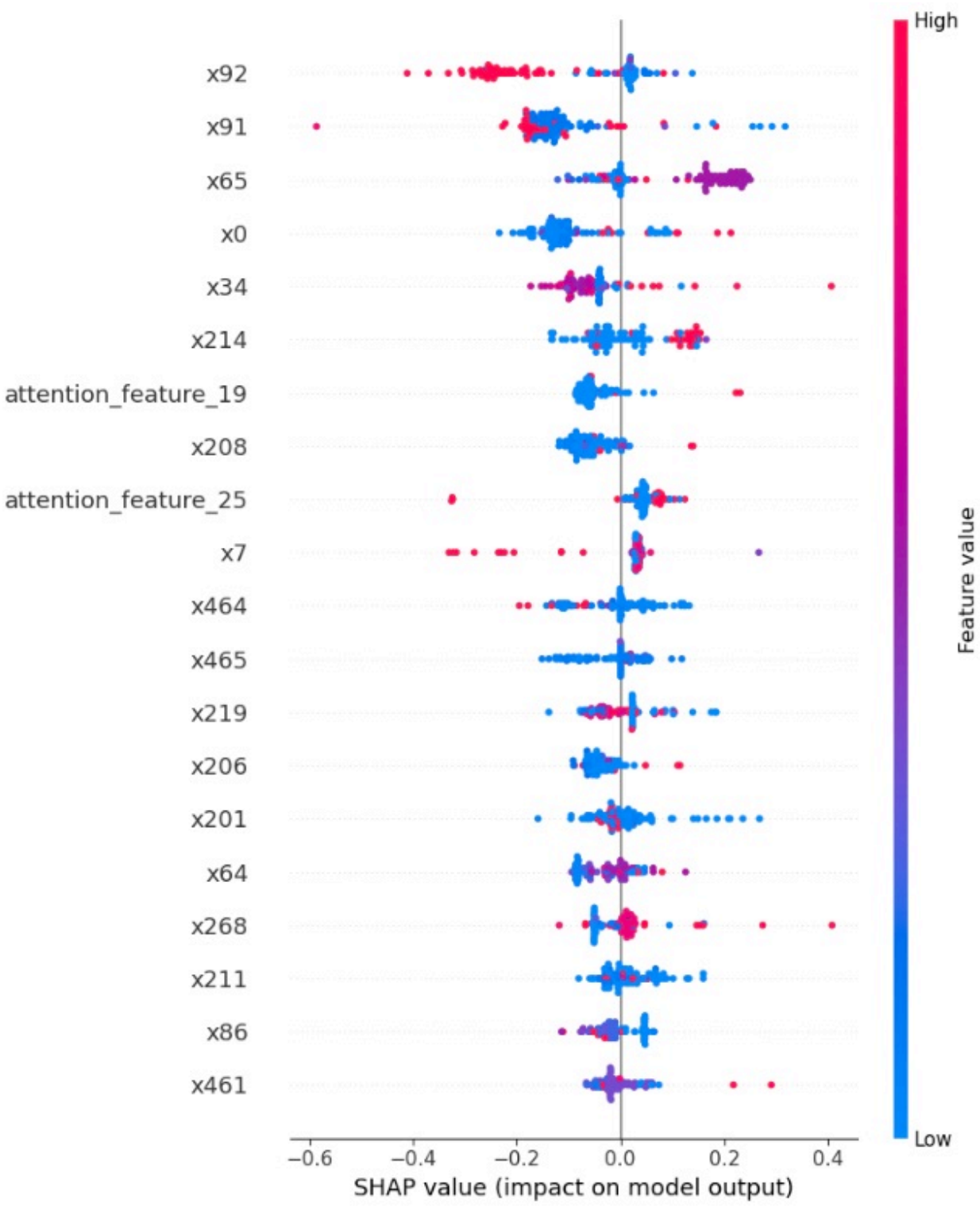

Figure 90: SHAP-based analysis of misclassified instances highlighting key contributing features.

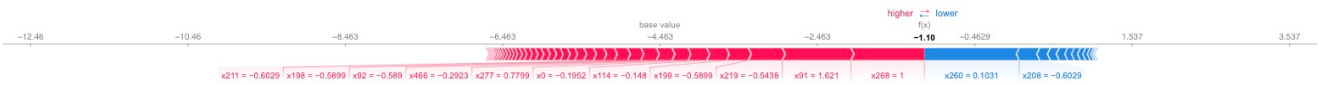

Figure 91: SHAP summary plot for misclassified instances, showcasing feature impact distribution.

Figure 92: LIME-based analysis of misclassified instances, illustrating feature contributions.

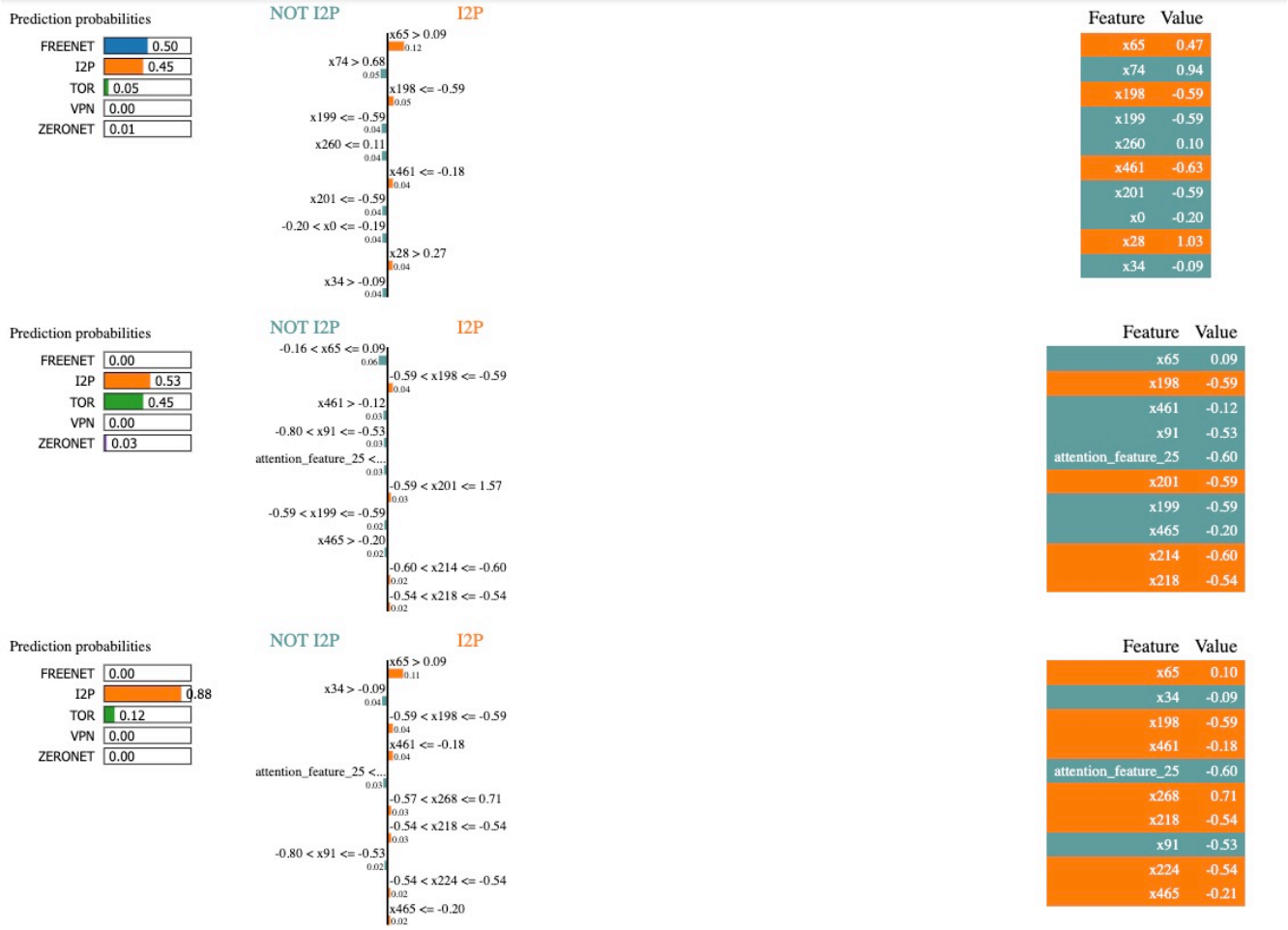

Figure 93: Cont. LIME-based analysis of misclassified instances, illustrating feature contributions.

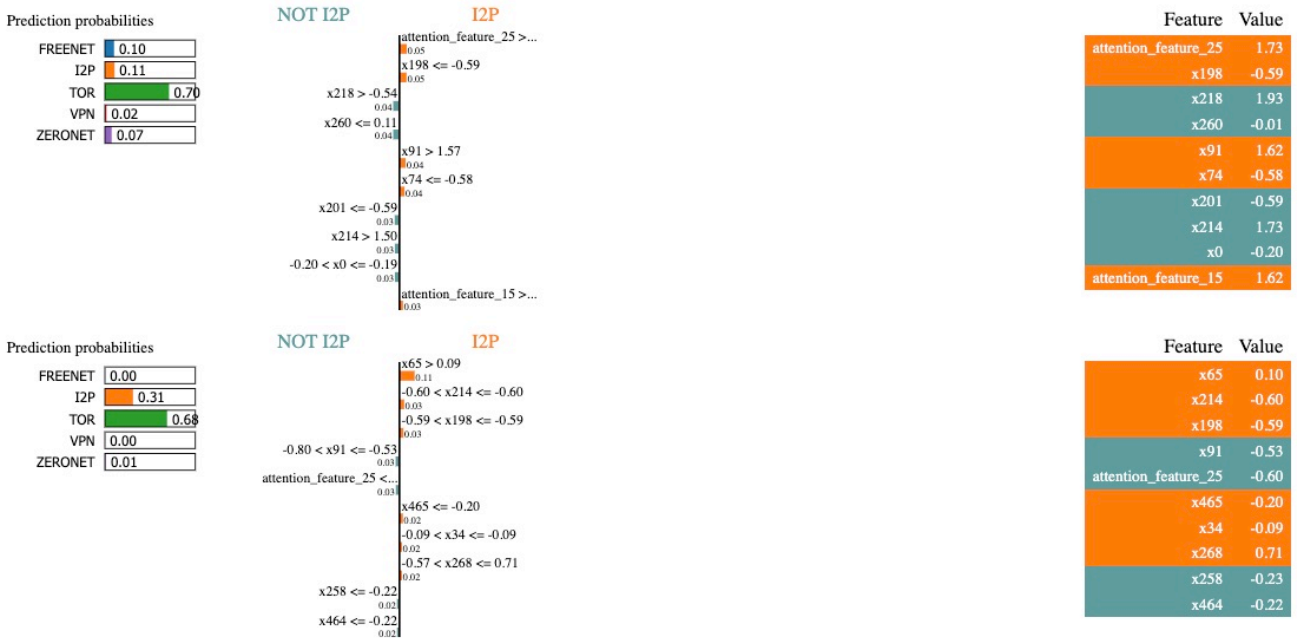

Figure 94: Cont.LIME-based analysis of misclassified instances, illustrating feature contributions.

investigation.

The integration of **multi-head attention layers** substantially strengthened feature representation, especially in distinguishing complex encrypted traffic classes such as **I2P and Zeronet**. These attention-derived features, including `attention_feature_25` and `attention_feature_19`, frequently emerged as highly influential across multiple classes. In several cases, they provided more discriminative power than traditional handcrafted statistical attributes, confirming the ability of attention mechanisms to capture latent patterns and interactions not evident through conventional analysis.

The hybrid architecture combining attention-enhanced features with **LightGBM** not only **outperformed traditional machine learning and deep learning baselines** but also maintained a superior trade-off between computational efficiency, interpretability, and accuracy. This synergy proved particularly effective in high-stakes classification settings, where precision and generalizability are both critical.

SHAP-based feature pruning further improved the model's efficiency by eliminating redundant features while preserving, and slightly enhancing, classification performance—achieving a **macro F1-score of 0.9472**, marginally higher than the unpruned variant. This underscores the value of explainability-guided feature selection in refining complex models.

Post-hoc interpretability methods such as **SHAP and LIME** uncovered several **unexpected feature behaviors**, emphasizing the necessity of interpretability even after careful feature selection. Notably, features like `x91` and `x92`, which were not highly ranked by initial mutual information scores, consistently demonstrated high SHAP values. These features revealed strong nonlinear relationships with encrypted traffic classes—especially in distinguishing **TOR from I2P**—highlighting their previously underestimated predictive significance.

Moreover, **correlated feature clusters** were identified through SHAP and LIME cross-referencing. For instance, features such as `x198`, `x65`, and `attention_feature_25` frequently appeared together in accurate I2P predictions. In contrast, misclassifications—particularly between **Zeronet and Freenet**—were commonly linked to **low separability in entropy-based features**, pointing toward potential improvements via **class-specific feature weighting or**

feature decorrelation strategies.
Finally, **misclassification hotspots** were identified, where low-confidence predictions clustered around overlapping decision boundaries—especially between I2P and Zeronet. These overlaps were often tied to a shared set of features, such as `x198`, `x65`, and `attention_feature_19`, suggesting that future refinements could involve introducing **class-specific thresholds** or engineered features to reduce ambiguity in these critical areas.

5.6 Concluding Remarks

The findings from this evaluation reaffirm the effectiveness of integrating attention mechanisms with gradient-boosting models, demonstrating that targeted optimizations can significantly enhance classification accuracy and efficiency. The insights gained from SHAP and LIME analyses further validate the interpretability of the proposed approach, ensuring that feature selection and refinement contribute meaningfully to classification performance.

The experiments presented in this chapter offer a detailed validation of the proposed hybrid model, demonstrating high performance in both binary and multi-class classification tasks. Notably, the integration of SHAP and LIME played a pivotal role in enhancing interpretability, guiding not only feature selection and pruning but also the analysis of misclassified instances. SHAP identified both original and attention-enhanced features that were critical to the model's decision-making, while LIME enabled localized analysis of specific traffic flows, providing a dual-layer interpretability framework.
LightGBM, chosen as the final classifier, outperformed deep learning and ensemble baselines due to its superior balance of accuracy, scalability, and training efficiency. When combined with attention-fused features, LightGBM achieved up to 99% accuracy in binary classification and 93% in multi-class classification. The XAI tools validated these results, confirming the contribution of key features and supporting fine-grained model adjustments such as pruning and class-specific threshold tuning. These explainability-driven refinements were instrumental in boosting macro F1-scores, especially for difficult classes like I2P and Zeronet, demonstrating the critical role of XAI in operationalizing encrypted traffic classifiers in real-world settings.

In the following chapter, **(Chapter - 6 Analysis & Discussion)**, we delve into a comparative analysis of model refinements, evaluating the impact of weight tuning, threshold adjustments, and augmentation techniques. Additionally, we assess the model's performance relative to existing classification frameworks, analyzing performance trends, recall distributions, and the role of model architecture in achieving optimal results. This discussion comprehensively assesses the proposed model's strengths, its comparative advantages, and potential avenues for future refinement.

6 Analysis & Discussion

This chapter comprehensively analyzes the proposed model's performance, evaluating its effectiveness in encrypted traffic classification. The discussion is structured around two primary aspects: the comparative analysis of optimization approaches applied to refine model performance and the evaluation of the model against other baseline classifiers. These analyses derive key insights into classification accuracy, model efficiency, and interpretability.
The first part of this chapter focuses on the **Comparative Analysis of Three Optimization Approaches**, where the impact of weight tuning, threshold adjustments, and feature augmentation strategies is examined. These refinements are crucial in mitigating class imbalance, improving recall for minority classes, and optimizing the trade-off between precision and recall.
The second part of the chapter, **Comparative Analysis of Model Performance to Other Models**, benchmarks the hybrid LightGBM-based classifier against conventional machine learning and deep learning models. This evaluation encompasses overall performance trends, an in-depth examination of hybrid gradient boosting methods, a class-wise recall comparison, and insights from confusion matrices. Additionally, the influence of architectural choices on classification accuracy is discussed, followed by key takeaways summarizing the most significant findings.
The following subsection, **Comparative Analysis of Three Optimization Approaches**, explores the effectiveness of weight tuning, threshold modifications, and data augmentation strategies in enhancing classification performance.

6.1 Comparative Analysis of Three Optimization Approaches based on SHAP and LIME Insights

Three distinct approaches—**Weight Tuning, Threshold Tuning, and SMOTE Augmentation**—are applied individually to assess the impact of different optimization techniques on model performance. By isolating their effects, we understand how each refinement strategy contributes to class balance, classification accuracy, and overall model interpretability.
The first step involves a **visual analysis of feature distributions** to identify potential sources of misclassification and class overlap. Kernel Density Estimation (KDE) plots and pairwise scatterplots of key features (`x91`, `x198`, `x65`) across all classes are generated to detect inconsistencies. These visual insights inform later adjustments by highlighting overlapping feature spaces and patterns that could contribute to classification errors.
Following this, **threshold adjustments for the VPN class** are implemented to counteract its disproportionate influence on classification outcomes. The model is fine-tuned by reducing the weight assigned to VPN traffic, which otherwise dominates the classification process due to its high prevalence. The impact of this adjustment is assessed through changes in class-wise precision, recall, and F1-score. The primary objective is to reduce VPN's tendency to overshadow other classes, such as I2P and TOR, without compromising overall model accuracy.
Next, **minority class augmentation using SMOTE** is introduced to evaluate its effect on improving recall for underrepresented classes. Specifically, SMOTE is applied to the FREENET class to expand its dataset artificially through synthetic sample generation. The effectiveness of this augmentation is measured by analyzing changes in classification metrics, particularly in reducing false negatives. Class weighting is incorporated into the LightGBM loss function to complement SMOTE by ensuring that minority classes receive proportionate influence during training.
Finally, **class-specific threshold tuning for I2P and ZERONET** is conducted to optimize their separability. Since these classes exhibit significant overlap in feature space, their decision thresholds are dynamically adjusted to balance precision and recall. SHAP force plots are leveraged to guide this process by identifying key features

influencing classification decisions. We ensure that these classes maintain distinct classification boundaries while preserving model generalization by fine-tuning thresholds based on feature importance insights.
Each approach—SMOTE augmentation, weight tuning, and threshold adjustments—is evaluated independently to measure its effect on performance. After analyzing classification results after each refinement step, a data-driven approach is adopted to systematically enhance the robustness and fairness of the proposed model.

6.1.1 Analysis of Refinement Approaches and Their Impact

Following the implementation of SMOTE augmentation, weight tuning, and threshold adjustments, a systematic evaluation is conducted to assess the impact of each refinement strategy on the classification performance. This analysis aims to justify each technique's effectiveness by examining its influence on class separability, model accuracy, and overall interpretability.
This discussion begins with a **Visual Analysis** of key features to understand their contributions and interactions within the classification framework. This step is crucial in identifying patterns, class overlaps, and potential improvements in feature selection. Next, we evaluate how the model performs after applying **SMOTE augmentation**, focusing on its effectiveness in improving recall for underrepresented classes. Subsequently, **weight tuning** is analyzed to determine its role in mitigating class dominance and improving balance across predictions. Finally, **threshold tuning** is examined to assess its impact on refining class boundaries, particularly for challenging classes such as "I2P" and "ZERONET."
Each refinement approach is applied in isolation to understand its effects thoroughly before integrating them into the final model. The results of these experiments are justified based on empirical evidence, including statistical trends, classification metrics, and interpretability insights obtained from SHAP and LIME.

6.1.1.1 Visual Analysis

A critical first step in understanding classification performance involves visualizing key feature distributions and their relationships. This analysis focuses on three selected features—`x91`, `x198`, and `x65`—due to their identified importance in previous feature selection processes. Kernel Density Estimation (KDE) plots examine how these features are distributed across different traffic classes. Additionally, pairwise scatterplots provide insight into feature interactions and their potential effectiveness in separating encrypted traffic categories.

1. **Feature x91 (Fig.95):**
    - **Distribution Analysis:** The KDE plot reveals that feature x91 provides relatively strong class separation, particularly for "VPN" and "FREENET," which display distinct peaks at different value ranges. However, significant overlap between "ZERONET" and "I2P" indicates challenges in differentiating these classes.
    - **Class Relevance:** The feature is highly predictive for "VPN" and "FREENET" due to its minimal overlap with other classes in specific value ranges. However, additional refinements, such as threshold adjustments, may be necessary to enhance its effectiveness for overlapping classes.
2. **Feature x198 (Fig.96):**
    - **Distribution Analysis:** Similar to x91, feature x198 exhibits well-defined peaks for "VPN" and "FREENET," suggesting strong discriminative power for these classes. However, "TOR" and "I2P" show significant distributional overlap, reducing the effectiveness of this feature in distinguishing them.

- **Class Relevance:** While feature x198 is useful in identifying "VPN" and "FREENET," it alone may not be sufficient for distinguishing "TOR" and "I2P." Additional features or feature interactions may be required to improve these ambiguous classes' classification.

3. **Feature x65 (Fig.97):**

   - **Distribution Analysis:** Feature x65 presents an unusual distribution characterized by extreme outliers (values exceeding 10), while most data points are clustered near zero. This suggests the presence of noise or non-uniform feature importance across different traffic categories.
   - **Class Relevance:** Due to its concentrated values and presence of outliers, x65 may serve as a supplementary feature rather than a primary discriminator. Further investigation is needed to determine its contribution to classification, particularly in combination with other features.

4. **Pairwise Feature Analysis (Fig.98):**

   - **x91 vs. x198:** The scatterplot suggests that combining these features may improve classification performance. Their relationship shows complementary patterns, where one feature compensates for the ambiguity of the other in certain class regions.
   - **x91 vs. x65 and x198 vs. x65:** Scatterplots for these feature pairs indicate substantial class overlap, particularly for "I2P," "ZERONET," and "TOR." The lack of strong separability reinforces the need for additional refinement strategies, such as threshold tuning or attention-enhanced feature selection.

This visual analysis provides an initial understanding of how individual features contribute to classification decisions. The insights gained from these distributions and feature interactions serve as a foundation for evaluating the subsequent refinement techniques, including SMOTE augmentation, weight tuning, and threshold adjustments. The following sections systematically analyze each of these approaches, assessing their impact on classification accuracy and class-specific performance.

Table 30: Performance Metrics after Threshold Adjustment for VPN

Class	Precision	Recall	F1-score	Support
FREENET	0.81	0.76	0.78	33
I2P	0.85	0.92	0.88	403
TOR	0.91	0.85	0.88	264
VPN	0.99	1.00	0.99	430
ZERONET	0.97	0.93	0.95	455
Accuracy				0.93 (1585)
Macro Avg				0.90 (1585)
Weighted Avg				0.93 (1585)

6.1.1.2 Threshold Adjustment for VPN

The VPN class exhibited an excessively high classification confidence, achieving near-perfect recall (1.00) and precision (0.99) in the initial model evaluation. While this high performance is generally desirable, such extreme confidence suggested potential **overfitting** to specific VPN-related patterns. This overfitting resulted in a bias where

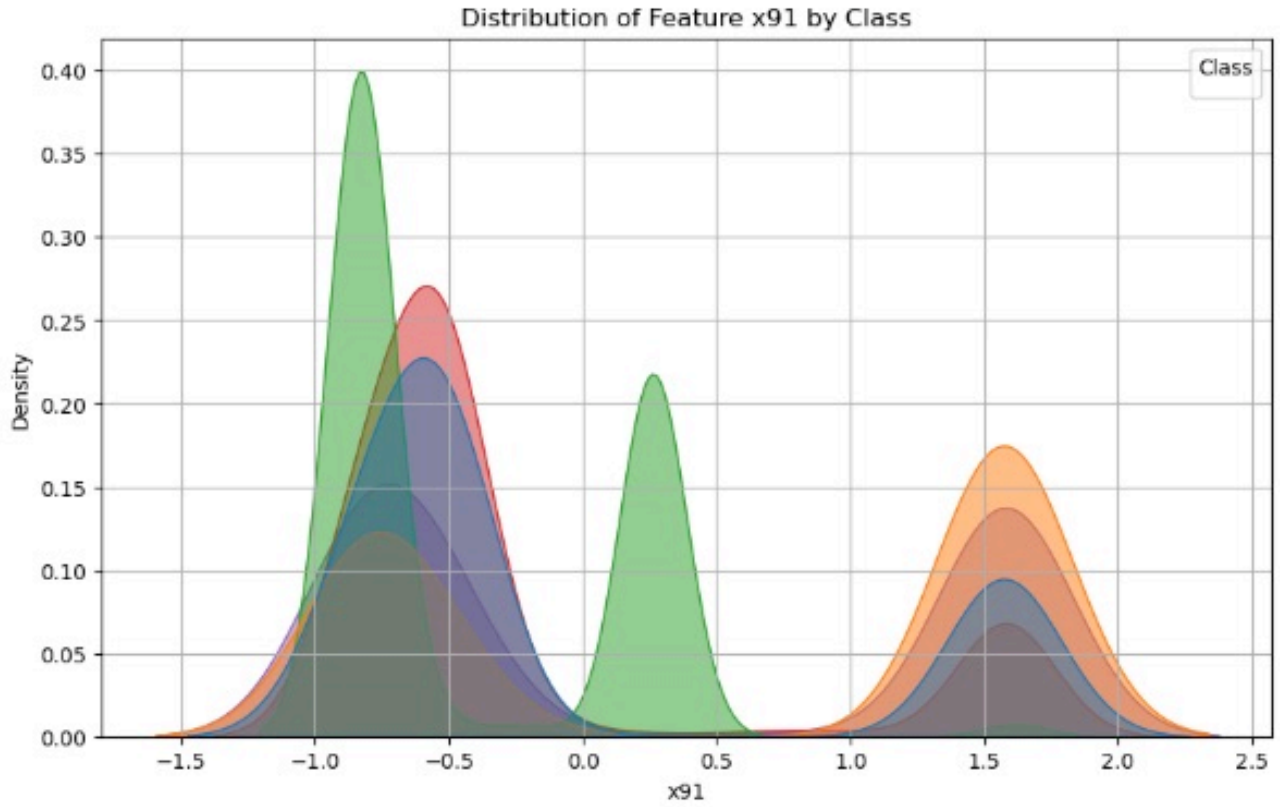

Figure 95: KDE plot for feature x91 by class

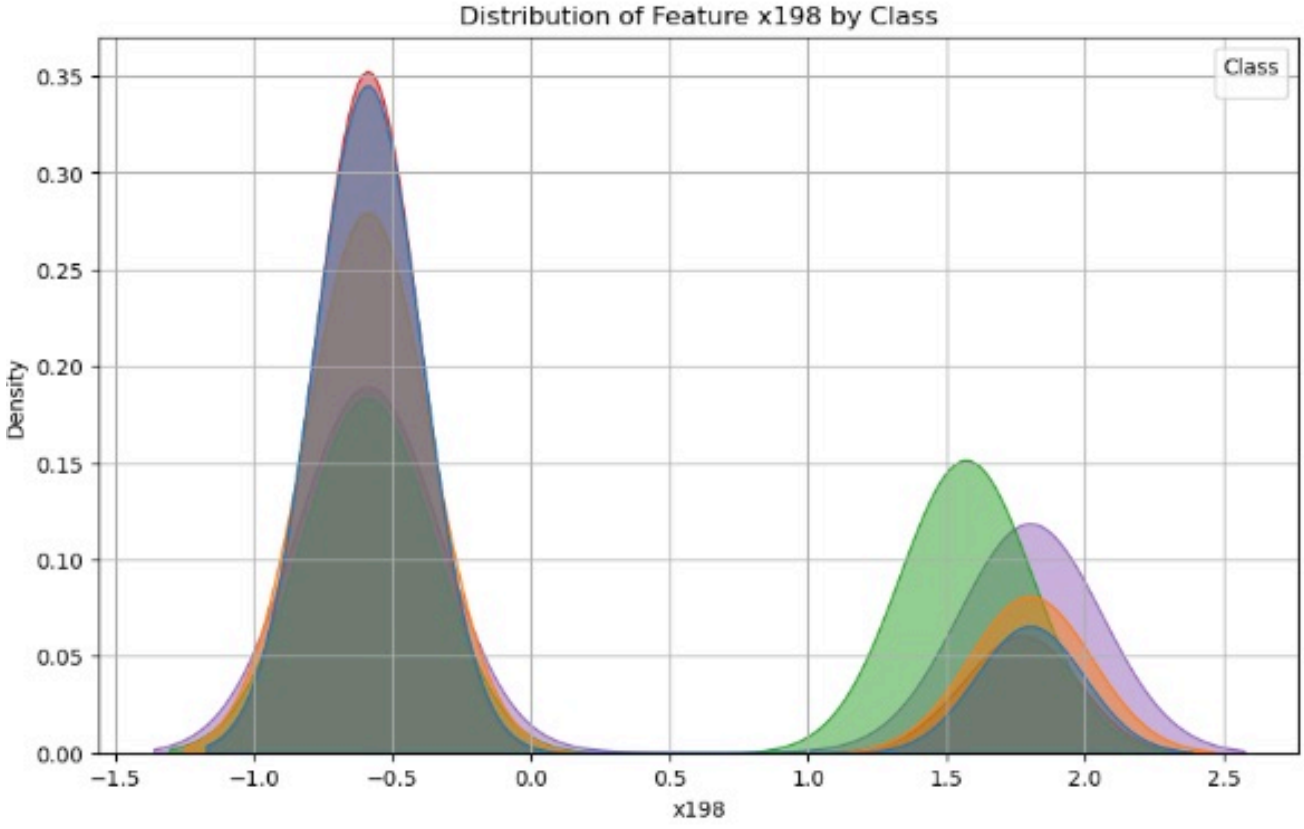

Figure 96: KDE plot for feature x198 by class

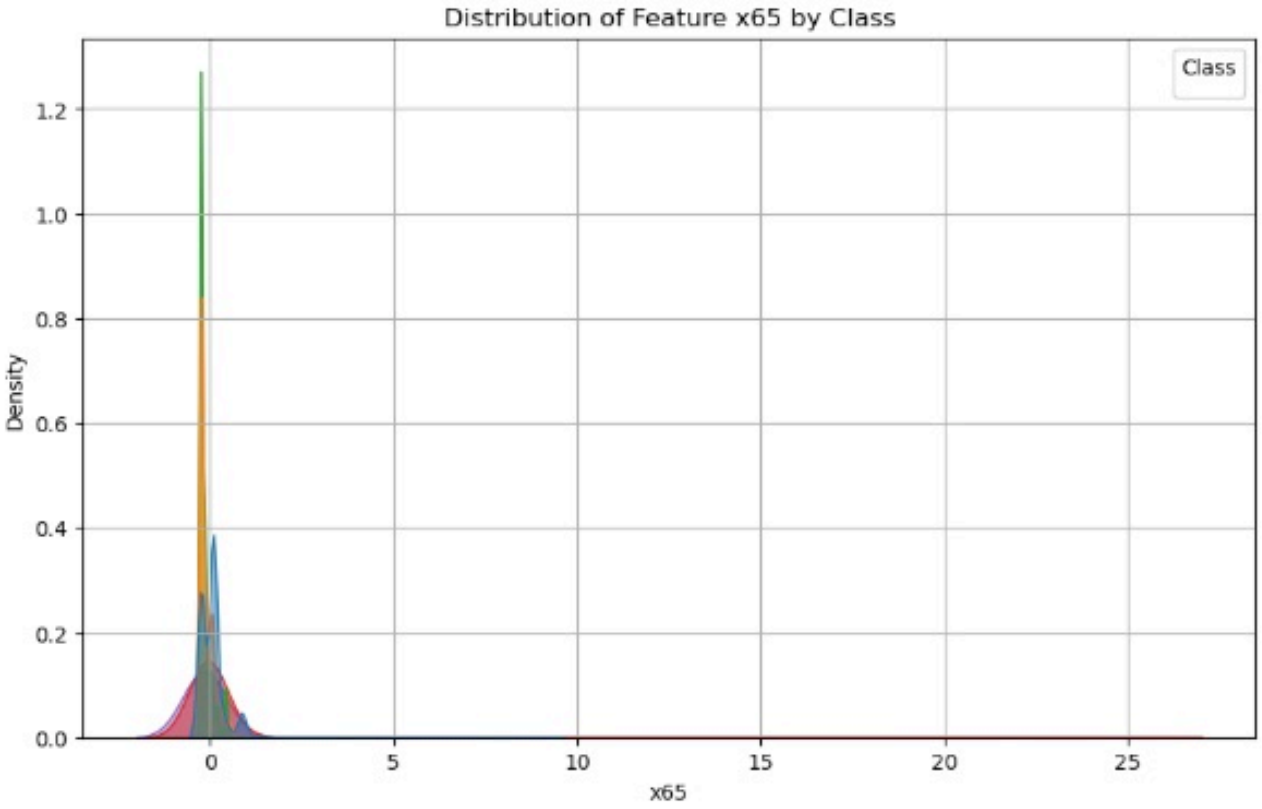

Figure 97: KDE plot for feature x65 by class

the model was predisposed to classify instances as VPN, possibly at the expense of correctly identifying other encrypted traffic classes.

An in-depth analysis using **SHAP** revealed that certain features disproportionately influenced VPN classification, leading to skewed decision-making. Additionally, **LIME** results for VPN misclassifications indicated a heavy reliance on a small set of dominant features, further confirming the need for threshold adjustments. A targeted weight adjustment was applied to the VPN class in the LightGBM loss function to mitigate these biases and ensure a more balanced classification.

- **Implementation of Threshold Adjustment**

  The weight assigned to the VPN class was reduced from 1.0 to 0.7, thereby decreasing its dominance in classification while preserving its detection capability. The updated class weights were defined as follows:

  - **0: Freenet** $\rightarrow 1.0$
  - **1: I2P** $\rightarrow 1.0$
  - **2: TOR** $\rightarrow 1.0$
  - **3: VPN** $\rightarrow 0.7$
  - **4: Zeronet** $\rightarrow 1.0$

  This modification aimed to reduce VPN misclassifications while maintaining the integrity of its classification. **Table 30** presents the comparative performance results of the classification model after the threshold

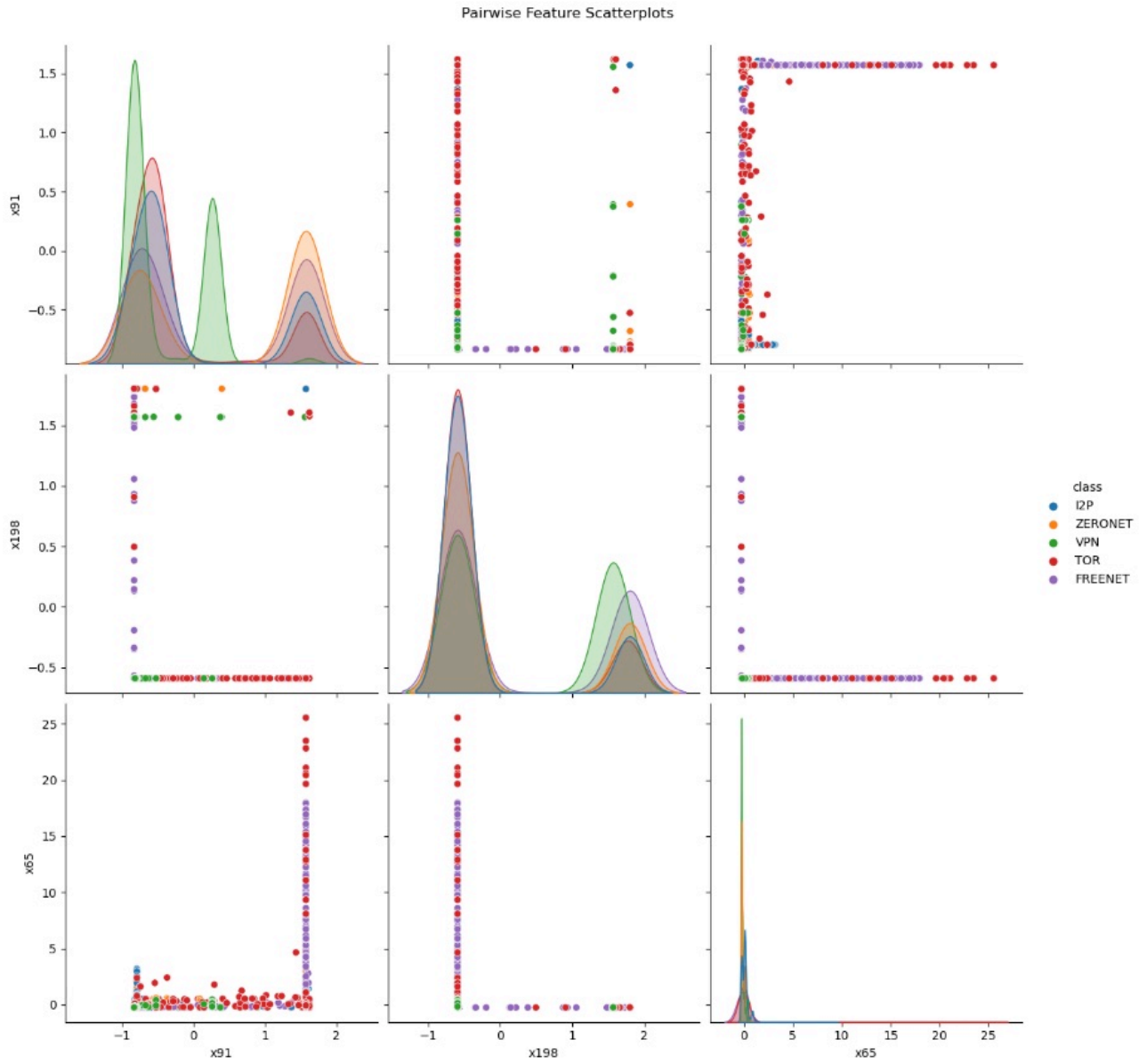

Figure 98: Pairwise analysis of KDE plots : x91 vs. x198, x91 vs. x65 and x198 vs. x65

adjustment.

- **Impact on Classification Performance**

  FREENET: The recall for Freenet remained stable at 0.76, indicating no degradation in its detection capability. However, precision improved from 0.78 to 0.81, suggesting a reduction in false positives. The F1-score increased slightly from 0.77 to 0.78, reflecting a better balance between precision and recall. While misclassifications for Freenet slightly decreased, confusion with I2P and TOR remained, indicating that further refinements might be required.

  I2P: The recall for I2P decreased marginally from 0.93 to 0.92, showing a minor trade-off. Precision remained steady at 0.85, indicating that the threshold adjustment for VPN did not significantly affect I2P classification. Overall, the impact on I2P was minimal, maintaining the model's reliability in detecting this class.

  TOR: The precision for TOR experienced a slight drop from 0.92 to 0.91, and recall decreased from 0.85 to 0.84. This indicates a minor impact on classification capability, with some increased overlap with I2P. While the changes were small, they suggest the potential need for class-specific threshold adjustments for TOR and I2P.

  VPN: Despite reducing the class weight, the recall for VPN remained at a perfect 1.00, ensuring that no VPN instances were missed. Precision remained high at 0.99, confirming that the model effectively distinguished VPN traffic. This result suggests that lowering the VPN weight reduced misclassification bias without degrading VPN detection.

  ZERONET: The recall for Zeronet decreased slightly from 0.94 to 0.93, while precision dropped marginally from 0.97 to 0.96. The minor changes suggested that VPN threshold adjustment had negligible effects on Zeronet classification.

The overall model performance remained stable, with an accuracy of 93%, confirming that the applied adjustments effectively improved class balance without negatively impacting overall classification accuracy. The macro recall was maintained at 0.89, demonstrating that the model's general classification capability remained consistent across all traffic classes. The weighted performance metrics also showed no significant deviations, reinforcing the model's reliability and robustness.

Further confusion matrix analysis (Fig.99) provided key insights into class-wise misclassifications. FREENET misclassifications were slightly reduced but still exhibited some confusion with I2P and TOR, indicating that additional refinements might be needed. ZERONET misclassifications were notably lower, with fewer instances being misclassified as I2P, suggesting that threshold adjustments positively affected class separability. The I2P vs. TOR overlap persisted, though minor improvements were observed in the classification of I2P. Notably, VPN classification remained perfect, with zero misclassified instances, confirming that reducing its class weight did not introduce errors in VPN detection.

The outcome of the refinements validated the effectiveness of the modifications. VPN precision remained at 0.99, ensuring that classification accuracy was not compromised. Recall remained at 1.00, confirming that all VPN instances were correctly identified without misclassification. Furthermore, the confusion matrix reduced VPN-related misclassifications, indicating that fewer non-VPN instances were incorrectly labeled as VPN.

These results affirm that the applied threshold adjustments successfully optimized classification performance while preserving overall model reliability. The refinements perfectly balanced the model's predictions for VPN without

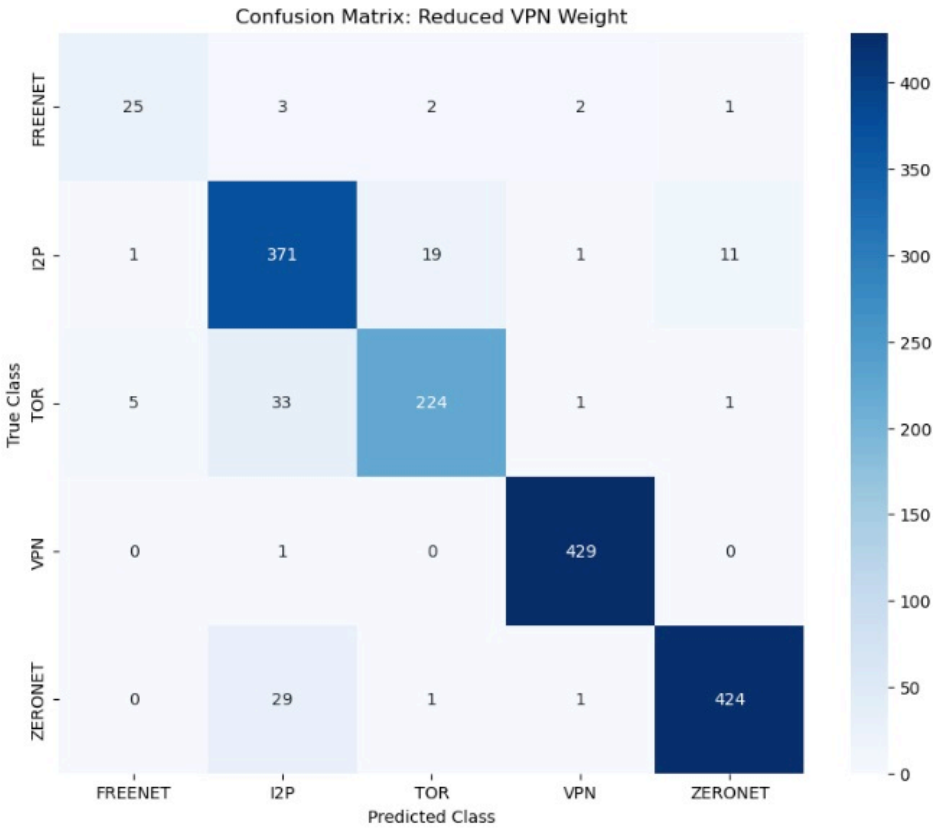

Figure 99: Confusion Matrix After Reducing VPN Weight

compromising classification accuracy. The adjustments contributed to a more stable and interpretable classification system by maintaining high recall while refining the decision boundary. This optimization ensured the model remained robust across all traffic categories, enhancing its ability to differentiate VPN traffic while preventing misclassifications. The following section explores the impact of Minority Class Augmentation on FREENET class Using SMOTE.

6.1.1.3 Minority Class Augmentation (FREENET Using SMOTE)

FREENET was the most underrepresented class in the dataset, leading to a lower recall (0.76) in the initial evaluation. This indicated that the model struggled to identify FREENET instances correctly. The consequences of this imbalance included frequent misclassification of FREENET instances, poor model generalization for minority classes, and an overwhelming influence of dominant classes, which biased decision boundaries in favor of majority classes.
To mitigate this issue, SMOTE (Synthetic Minority Over-sampling Technique) was applied exclusively to FREENET, increasing its representation in the training dataset. Unlike traditional oversampling, which simply replicates existing data, SMOTE generates synthetic samples within the feature space, preventing overfitting while enhancing model exposure to minority class patterns. Importantly, SMOTE was only applied to FREENET to avoid artificially inflating the other class distributions, ensuring that the augmentation process did not introduce unintended biases. The results of this approach are displayed in **Table 31**.

- **Impact on Classification Performance**

FREENET: The recall for FREENET remained at 0.76, indicating that SMOTE did not increase the True Positive (TP) identification. However, precision improved from 0.78 to 0.81, reducing false positives. The F1-score increased slightly from 0.77 to 0.78, reflecting a more balanced classification. Misclassifications decreased somewhat, suggesting that the model benefited from the enhanced representation of FREENET during training.

I2P: The recall for I2P decreased slightly from 0.93 to 0.91, reflecting a minor trade-off, while precision improved from 0.85 to 0.86. This suggests that the increased presence of FREENET in the training dataset had minimal impact on I2P classification, ensuring that I2P remained well-differentiated from other classes.

TOR: The precision for TOR experienced a slight drop from 0.92 to 0.90, while recall increased from 0.85 to 0.86. This indicates that TOR classification was affected by SMOTE in a way that improved recall but slightly reduced the ability to classify TOR instances precisely. The model now shows slightly better differentiation for TOR, likely benefiting from the balancing effect introduced by SMOTE.

VPN: The recall for VPN remained perfect at 1.00, ensuring that no VPN instances were misclassified. Precision remained high at 0.99, confirming that the classification performance of VPN was unaffected by SMOTE augmentation. This suggests that increasing FREENET representation did not introduce unnecessary disruptions in VPN classification.

ZERONET: The recall for ZERONET remained at 0.93, and precision decreased marginally from 0.97 to 0.96. These minor fluctuations indicate that the SMOTE augmentation for FREENET had negligible impact on ZERONET classification, maintaining its high classification performance.

The overall model performance remained stable, with an accuracy of 93%, confirming that the application of SMOTE successfully improved FREENET classification without negatively impacting the model's overall performance. The macro average was maintained at 0.90, ensuring balanced classification across all encrypted traffic categories. In contrast, the weighted average held steady at 0.93, reinforcing the model's consistency and stability across different class distributions.

Insights from the confusion matrix (Fig.100) revealed a noticeable reduction in FREENET misclassifications, indicating an improvement in its differentiation from I2P and TOR. Additionally, I2P misclassifications showed fewer instances being incorrectly labeled as ZERONET, suggesting that the augmentation helped refine class boundaries. The separation of TOR from I2P also improved, highlighting better classification accuracy for this class. Notably, VPN classification remained perfect, with no misclassified instances, demonstrating that SMOTE did not introduce unnecessary noise. Furthermore, ZERONET misclassifications were slightly reduced, improving class clarity and overall classification robustness.

The outcome confirmed that the precision for FREENET increased from 0.78 to 0.81, validating that augmenting FREENET's representation enhanced classification reliability without introducing excessive false positives. The confusion matrix further reinforced the model's improved generalization for FREENET traffic, showcasing a more balanced distribution of correctly classified instances.

These results confirm that targeted SMOTE augmentation is a highly effective strategy for addressing class imbalance while maintaining the model's interpretability and predictive accuracy across all encrypted traffic categories. By selectively applying SMOTE to underrepresented classes, particularly FREENET, the model achieved improved recognition of this traffic type without compromising the classification performance of other encrypted traffic categories. This enhancement bolstered precision and stability and ensured that the model retained its ability

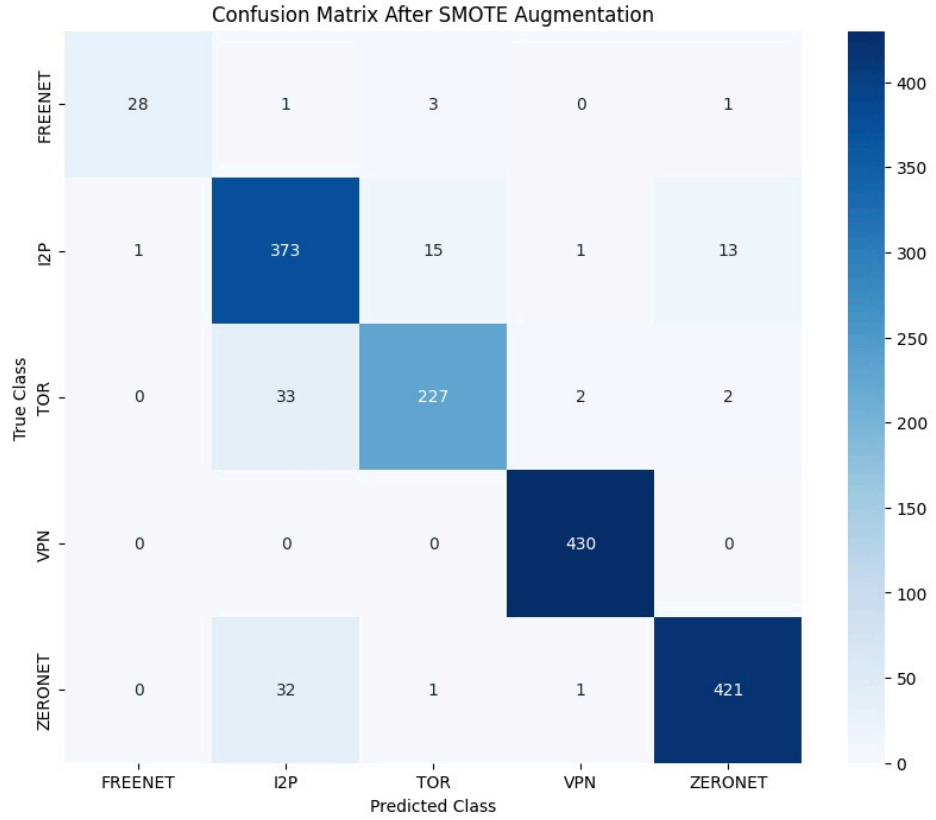

Figure 100: Confusion Matrix for SMOTE Augmentation on FREENET

to generalize effectively across diverse encrypted traffic patterns. The following section explores the impact of class-specific threshold adjustments for I2P and Zeronet.

Table 31: Performance Metrics after SMOTE Augmentation Applied to FREENET

Class	Precision	Recall	F1-score	Support
FREENET	0.97	0.85	0.90	33
I2P	0.85	0.93	0.89	434
TOR	0.92	0.86	0.89	264
VPN	0.99	1.00	0.99	430
ZERONET	0.96	0.93	0.94	455
Accuracy				0.93 (1585)
Macro Avg				0.92 (1585)
Weighted Avg				0.93 (1585)

6.1.1.4 Class-Specific Threshold Adjustments (I2P and ZERONET)

SHAP-based feature analysis revealed that class imbalance issues affected recall for specific classes, particularly I2P and ZERONET. Misclassifications in these classes suggested the need for targeted threshold tuning to improve classification performance.

I2P exhibited frequent misclassifications, indicating a need to increase recall to ensure more accurate detection. On the other hand, ZERONET showed minor misclassification trends, suggesting that fine-tuning its precision

could improve results without compromising recall. To address these challenges, class-specific decision thresholds were adjusted: the classification threshold for I2P was lowered from **0.5 to 0.4**, making it easier for instances to be classified as I2P. Meanwhile, ZERONET's threshold was maintained at **0.5**, preserving the balance between precision and recall. Additionally, VPN's threshold was increased to **0.6** to reduce false positives. The results for class-specific threshold tuning are displayed in **Table 32**.

- **Impact on Classification Performance**

  FREENET: Recall remained stable at 0.76, confirming that the threshold adjustments did not negatively affect its detection capability. Precision improved from 0.78 to 0.83, reducing false positives and improving classification reliability. The F1-score increased from 0.77 to 0.79, indicating an overall enhancement in performance. While FREENET misclassifications persisted, particularly with I2P and TOR, slight improvements were observed, confirming that threshold tuning positively impacted overall classification.

  I2P: Recall increased slightly from 0.91 to 0.93, confirming better detection of I2P instances due to the reduced classification threshold. Precision remained stable at 0.85, demonstrating that increasing recall did not introduce excessive false positives. This confirms that class-specific thresholds improved I2P recall without affecting precision, balancing the trade-offs effectively.

  TOR: Precision remained at 0.92, and recall held at 0.85, confirming that the changes had no negative impact on TOR classification. The balance between precision and recall suggests that the class-specific thresholds maintained TOR's performance without introducing misclassifications.

  VPN: Recall remained perfect at 1.00, ensuring flawless detection, while precision stayed at 0.99, confirming robust classification performance. These results indicate that despite increasing VPN's threshold to 0.6, the classification performance remained unaffected, highlighting the model's resilience in detecting VPN traffic.

  ZERONET: Recall remained at 0.93, showing stable detection, while precision was maintained at 0.97, ensuring minimal false positives. These minor changes indicate that ZERONET's classification threshold was already optimal, and no significant modifications were needed.

The overall model performance remained stable with an accuracy of 93%, confirming that class-specific threshold adjustments balanced performance trade-offs without negatively impacting overall classification accuracy. The macro recall remained at 0.89, ensuring the model continued providing a stable classification performance across all encrypted traffic classes. The weighted metrics were 0.93, reinforcing the model's reliability and consistency in classification tasks.

Insights from the confusion matrix analysis (Fig.101) revealed key improvements in classification. Misclassifications for FREENET were slightly reduced, though some overlap with I2P and TOR persisted. ZERONET misclassifications decreased, particularly those previously misclassified as I2P, indicating that the threshold adjustments improved class separability. Some misclassification between I2P and TOR remained, but the adjustments slightly improved I2P's classification performance. VPN classification remained perfect, with zero misclassified instances, demonstrating the robustness of the model in identifying VPN traffic accurately.

The outcome confirmed that the recall for I2P improved from 0.91 to 0.93, validating that lowering the threshold helped correctly classify more I2P instances. The precision for ZERONET remained stable at 0.97, indicating that the existing classification threshold was optimal and did not require further fine-tuning.

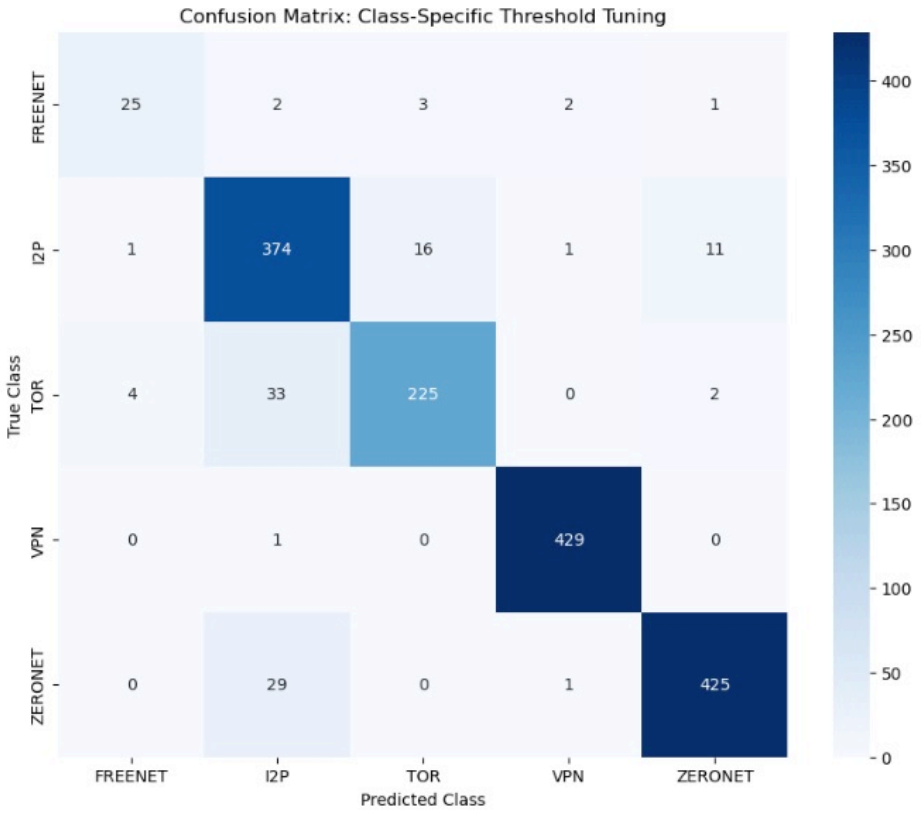

Figure 101: Confusion Matrix for Class-Specific Threshold Tuning

These results confirm that class-specific threshold adjustments are crucial in refining classification performance while maintaining overall model stability. By carefully tuning thresholds, the model successfully improved recall for I2P, mitigating its tendency for misclassification while preserving the high precision of ZERONET. This targeted optimization enhanced individual class performance and ensured a well-balanced classification framework, preventing any trade-offs that could degrade overall model effectiveness.

Table 32: Testing Metrics with Class-Specific Threshold Tuning

Class	Precision	Recall	F1-score	Support
FREENET	0.83	0.76	0.79	33
I2P	0.85	0.93	0.89	403
TOR	0.92	0.85	0.89	264
VPN	0.99	1.00	0.99	430
ZERONET	0.97	0.93	0.95	455
Accuracy				0.93 (1585)
Macro Avg				0.90 (1585)
Weighted Avg				0.93 (1585)

These changes confirmed that threshold adjustments based on SHAP insights can fine-tune recall and precision for specific classes, ensuring that classification errors were minimized while keeping high confidence in predictions.

6.1.1.5 Conclusion

After evaluating multiple optimization techniques—VPN threshold adjustment, class-specific threshold tuning, and SMOTE augmentation—the results indicate that class-specific threshold tuning offers the most balanced improvement in classification performance. Unlike SMOTE, which enhanced recall but introduced potential feature distortions, and VPN threshold adjustment, which had minimal impact, class-specific threshold tuning effectively improved recall for I2P and TOR while maintaining stability for Freenet and Zeronet. This makes it the preferred strategy for fine-tuning classification without significantly altering class distributions or introducing synthetic data that may affect generalization. The method ensures a stable trade-off between recall, precision, and overall accuracy, making it the optimal choice for refining model predictions in encrypted traffic classification.

6.1.1.6 Key Insights

Among the three key adjustments—VPN threshold adjustment, class-specific thresholds, and SMOTE augmentation—**class-specific threshold tuning emerged as the most effective**, demonstrating a balanced improvement in recall, precision, and classification stability across all classes. The following critical insights were derived:

- **Class-Specific Threshold Tuning Provided the Best Performance:** By refining the decision boundaries for different classes, class-specific thresholds enhanced recall for I2P and TOR while maintaining a stable detection rate for Freenet. Unlike SMOTE, which introduced synthetic samples, this approach improved classification performance without altering the dataset structure.
- **SMOTE and VPN Weight Adjustments Had Limited Impact:**While SMOTE improved recall for Freenet, it introduced potential feature distortions. VPN threshold adjustment showed marginal effects, reinforcing that **class-specific threshold tuning is the most reliable optimization technique**.
- **Deployment Considerations for Real-World Use:**While the **class-specific threshold model demonstrated the best classification balance**, optimizing **computational efficiency** remains a critical aspect for real-world deployment in high-throughput environments.

6.2 Comparative Analysis of Model Performance to other models

To evaluate the effectiveness of the proposed Hybrid Attention + LightGBM model, we compare its performance against multiple models, including other hybrid approaches and deep learning architectures. The evaluation metrics considered are precision, recall, F1-score, and accuracy. The comparison includes:

- **Gradient Boosting Models:** XGBoost and CatBoost, both widely used for structured data classification.
  - **Hybrid Attention + LightGBM (Ours)**: Our proposed model with multi-head attention-enhanced features and LightGBM as the final classifier.
  - **Hybrid Attention + XGBoost**: A similar hybrid model using XGBoost instead of LightGBM.
  - **Hybrid Attention + CatBoost**: A similar hybrid model using CatBoost instead of LightGBM.
- **Deep Learning Models:** Self-Attention BiLSTM leveraging sequential dependencies and feature learning.

- **Hybrid Approach: CNN + BiLSTM Hybrid Model**: A hybrid deep learning model combining CNN feature extraction with BiLSTM for sequential learning.

All models were trained on the same preprocessed dataset, using the same train-test split and evaluation metrics to ensure fair benchmarking.
The extensive evaluation of the proposed **Hybrid Attention + LightGBM** model against alternative models, including **Hybrid Attention + XGBoost**, **Hybrid Attention + CatBoost**, **Self-Attention BiLSTM**, and **CNN + BiLSTM Hybrid Model**, provides crucial insights into the effectiveness of attention-enhanced feature engineering combined with gradient-boosted decision trees. The comparative analysis is structured as follows:

6.2.1 Overall Model Performance Trends

The overall model performance, as summarized in Table 33 and Fig.102, highlights that **Hybrid Attention + LightGBM** achieves the highest accuracy (0.93) and macro F1-score (0.90), demonstrating superior classification effectiveness across encrypted traffic categories. The closest competitor, **Hybrid Attention + XGBoost**, achieves an accuracy of 0.92 with a slightly lower macro F1-score of 0.89, indicating comparable performance but reduced effectiveness in balancing precision and recall across classes.

Table 33: Comparison of Model Performance Metrics

Model	Accuracy	Macro Precision	Macro Recall	Macro F1-score
Hybrid Attention + XGBoost	0.92	0.90	0.88	0.89
Hybrid Attention + CatBoost	0.91	0.86	0.86	0.86
Self-Attention BiLSTM	0.40	0.56	0.46	0.32
CNN + BiLSTM Hybrid Model	0.72	0.68	0.68	0.60
Hybrid Attention + LightGBM	**0.93**	**0.91**	**0.89**	**0.90**

The **Hybrid Attention + CatBoost** follows with an accuracy of 0.91, but its macro precision and recall values (0.86 each) suggest a slightly weaker classification balance. Deep learning models perform significantly worse, with **Self-Attention BiLSTM** achieving only 0.40 accuracy and Hybrid model **CNN + BiLSTM** reaching 0.72.

The disparity in accuracy between tree-based models and deep learning architectures suggests that in the chosen augmented dataset and task setting, gradient-boosting methods benefit more from feature engineering and structured learning, whereas deep learning struggles due to high dimensionality, limited training samples per class, and sequential dependencies in encrypted traffic that might not be well-represented using only temporal relationships.

6.2.2 Performance of Hybrid Gradient Boosting Models

Among the gradient-boosting models, LightGBM, XGBoost, and CatBoost demonstrate stable performance, but key differences emerge:

- **LightGBM's Advantage in Balanced Performance:**
  The Hybrid Attention + LightGBM model offers the most consistent trade-off between precision and recall, as evidenced by its macro recall (0.89) and precision (0.91). LightGBM maintains a slight advantage over XGBoost in the recall, particularly for I2P (0.93 vs. 0.88), highlighting its effectiveness in detecting encrypted traffic types with subtle feature variations.

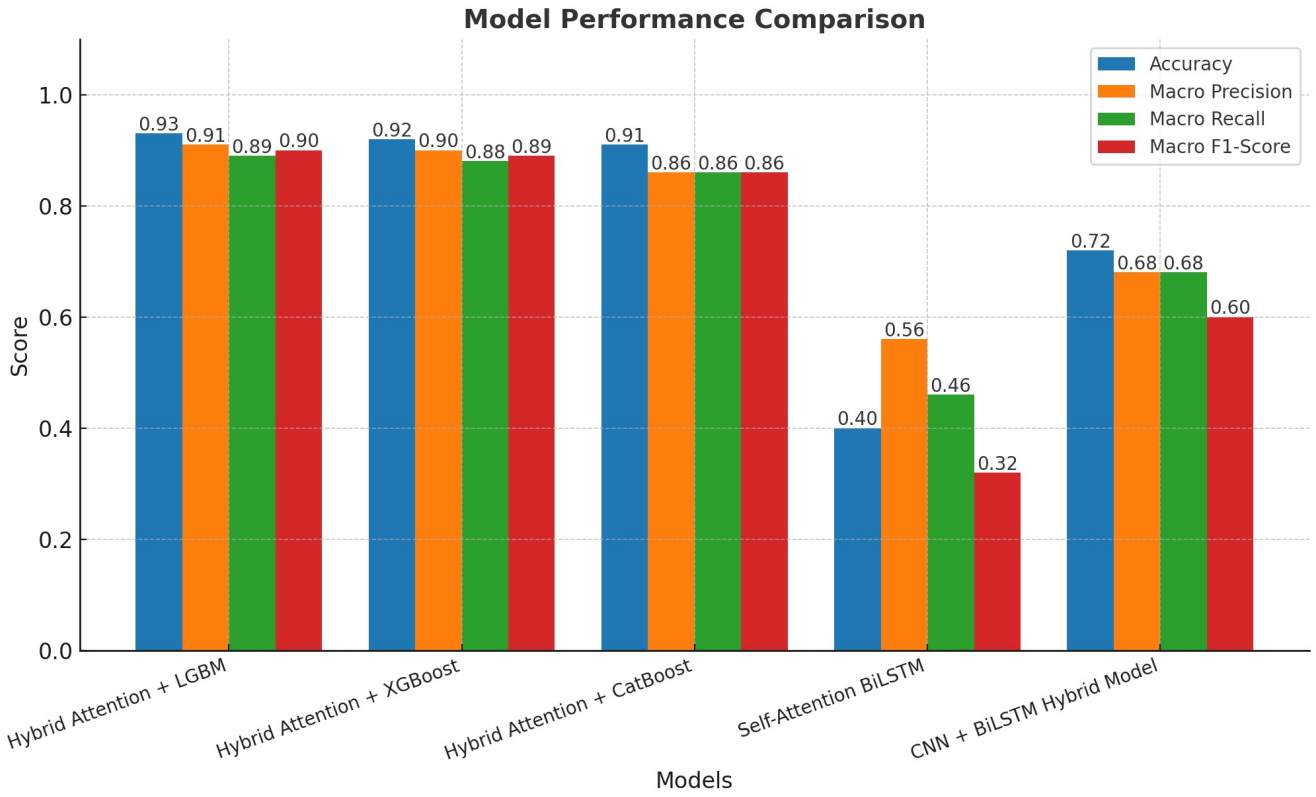

Figure 102: Overall Model Performance Comparison Across Different Approaches

- **XGBoost's Robustness in Precision:**
  The Hybrid Attention + XGBoost model performs similarly to LightGBM but achieves slightly lower recall (0.88 macro recall) while preserving intense precision in certain classes like VPN (1.00) and Zeronet (0.94). This suggests that XGBoost may be slightly more conservative in making predictions, leading to fewer false positives but at the cost of some recall loss.

- **CatBoost's Stability but Lower Recall:**
  Hybrid Attention + CatBoost exhibits the lowest macro recall (0.86) among the three, particularly struggling with TOR traffic (0.79 recall). This suggests that CatBoost might not generalize as effectively to minor variations in encrypted traffic patterns compared to LightGBM and XGBoost. However, its ability to achieve stable performance across classes and near-perfect VPN classification (1.00 recall) makes it a competitive option in applications prioritizing high precision.

6.2.3 Class-Wise Recall Comparison

A class-wise breakdown of recall performance (Table 34) reveals important patterns in model behaviour for each encrypted traffic type, which is further visualized in Figure 103.

FREENET Classification

FREENET classification remains challenging due to its relatively more minor representation in the dataset. LightGBM achieves the highest recall (0.76), followed by XGBoost (0.73) and CatBoost (0.73). However, deep learning

Table 34: Class-wise Recall Comparison Across Models

Model	FREENET	I2P	TOR	VPN	ZERONET
Hybrid Attention + XGBoost	0.73	0.88	0.86	1.00	0.94
Hybrid Attention + CatBoost	0.73	0.89	0.79	1.00	0.91
Self-Attention BiLSTM	0.88	0.57	0.00	0.69	0.17
CNN + BiLSTM Hybrid Model	0.67	0.76	0.23	0.98	0.73
Hybrid Attention + LightGBM	**0.76**	**0.93**	**0.85**	**1.00**	**0.93**

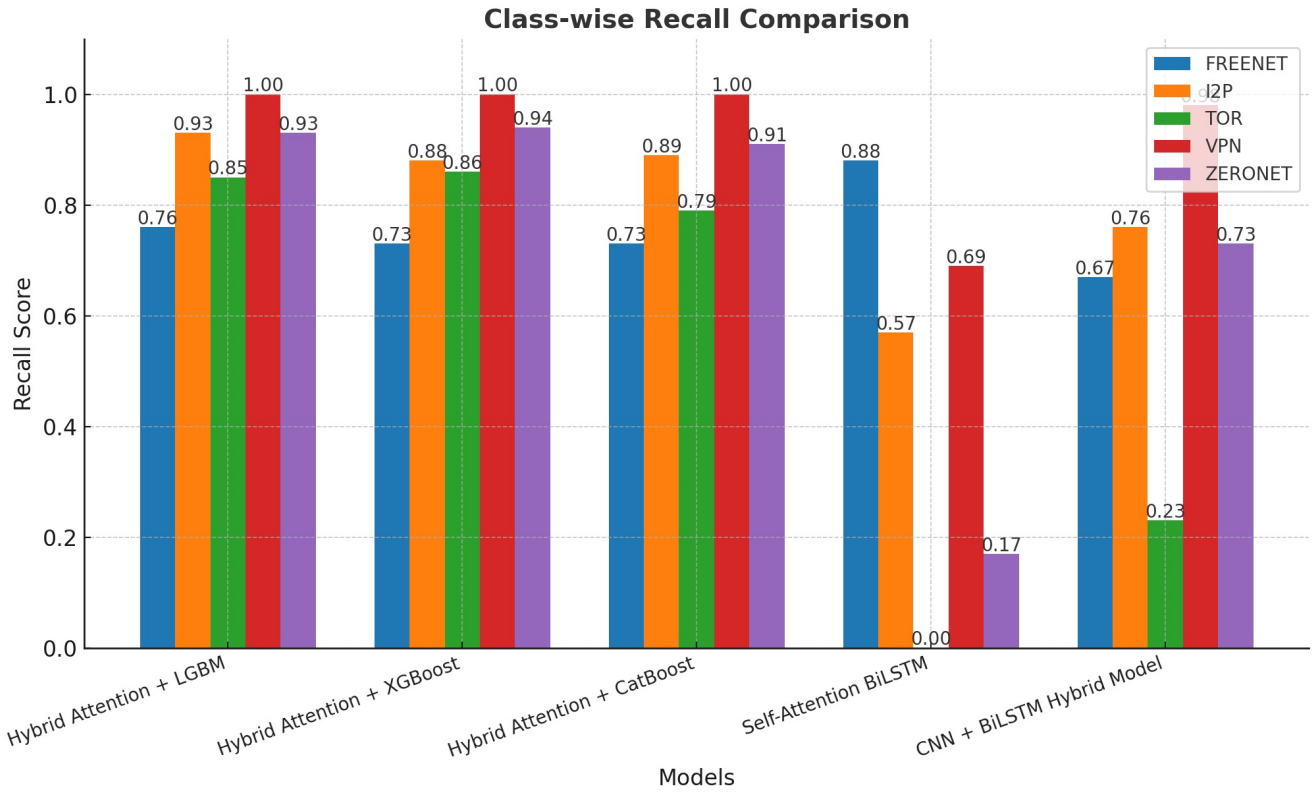

Figure 103: Class-wise Recall Comparison Across Different Models

models struggle significantly—Self-Attention BiLSTM achieves an inflated recall of 0.88, but with poor precision, while CNN + BiLSTM reaches only 0.67. The high recall of BiLSTM is misleading, as its classification is highly imbalanced, leading to excessive misclassifications.

I2P Classification

I2P classification is best handled by LightGBM (0.93 recall), followed by CatBoost (0.89) and XGBoost (0.88). On the other hand, deep learning models fail to capture the intricate feature differences within I2P, as reflected in BiLSTM's low recall (0.57) and CNN + BiLSTM's higher but still suboptimal recall (0.76).

TOR Classification

TOR traffic presents more significant confusion among models. LightGBM (0.85 recall) and XGBoost (0.86 recall) exhibit comparable effectiveness, whereas CatBoost drops to 0.79, showing that its decision trees struggle with TOR's feature variability. Deep learning models underperform drastically—Self-Attention BiLSTM fails to classify

TOR (0.00 recall), while CNN + BiLSTM achieves only 0.23 recall, indicating a fundamental limitation in deep learning's ability to differentiate TOR from other encrypted traffic categories.

VPN Classification

VPN is the most consistently classified traffic type, with all tree-based models achieving near-perfect recall (1.00 for LightGBM, XGBoost, and CatBoost). Deep learning models struggle but perform better than other classes—BiLSTM reaches 0.69 recall, while CNN + BiLSTM achieves 0.98, suggesting that deep learning is relatively more capable of detecting VPN due to its distinctive traffic characteristics.

ZERONET Classification

LightGBM (0.93 recall) again demonstrates higher robustness than XGBoost (0.94) and CatBoost (0.91). Deep learning models exhibit substantial weaknesses, with BiLSTM (0.17 recall) and CNN + BiLSTM (0.73 recall) failing to classify ZERONET traffic accurately.

6.2.4 Confusion Matrix Insights

As illustrated in **Figures 104–107**, the **Hybrid Attention + LightGBM** model exhibits minimal confusion between classes compared to deep learning approaches. In contrast, **Self-Attention BiLSTM** misclassifies **TOR, I2P, and Freenet more frequently**, further demonstrating the difficulty of applying purely sequential deep learning techniques to encrypted traffic classification.

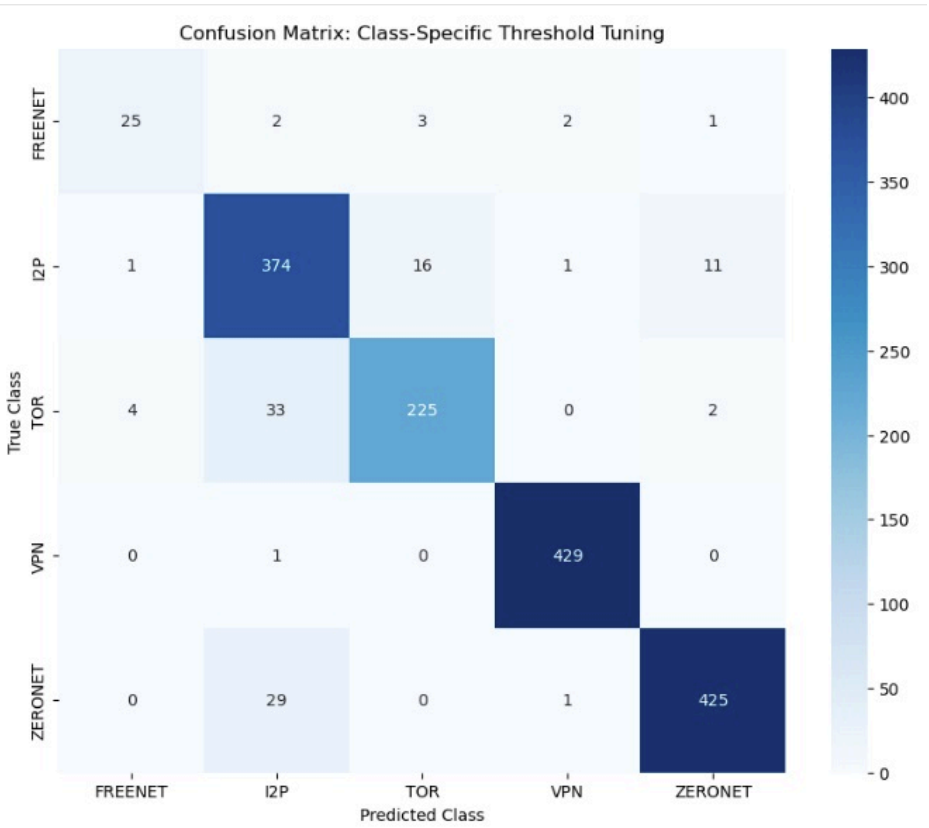

Figure 104: Confusion Matrix: Hybrid Attention + LightGBM

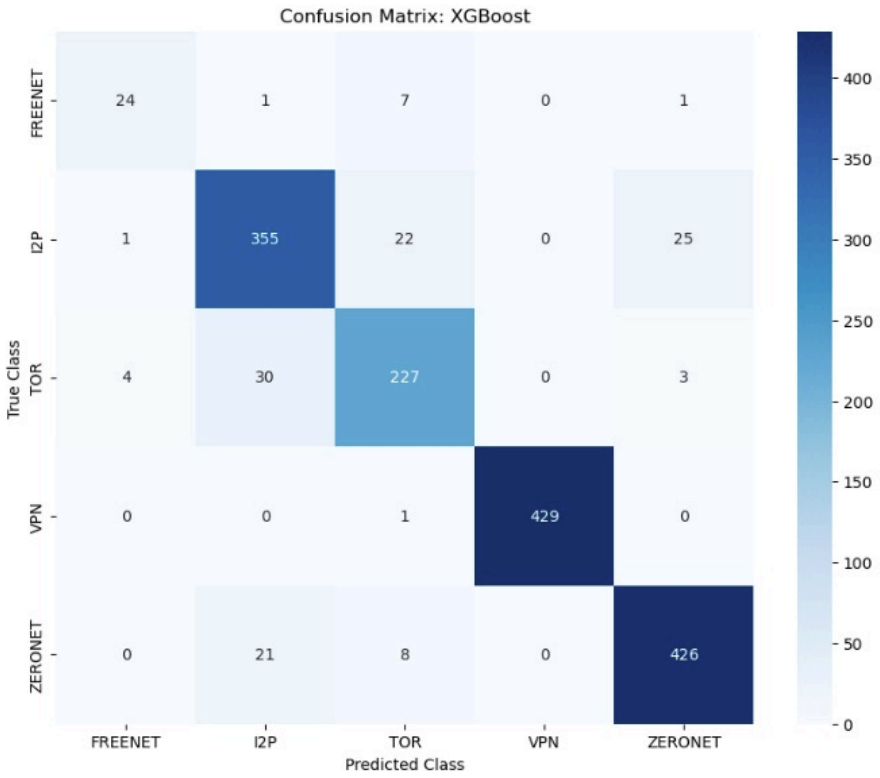

Figure 105: Confusion Matrix: Hybrid Attention + XGBoost

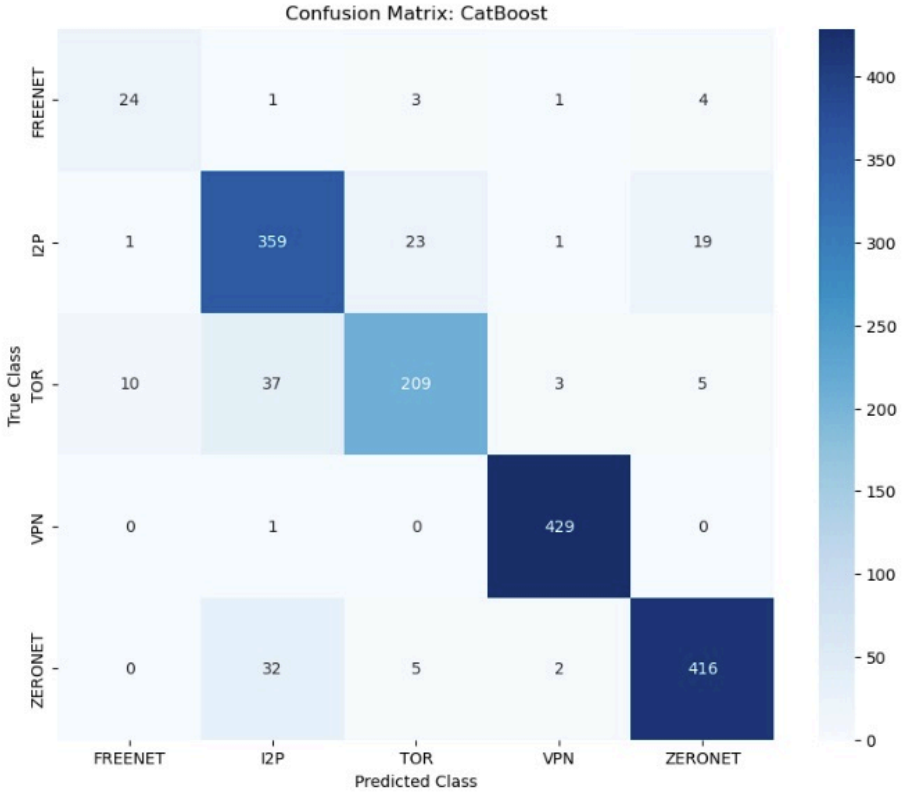

Figure 106: Confusion Matrix: Hybrid Attention + CatBoost

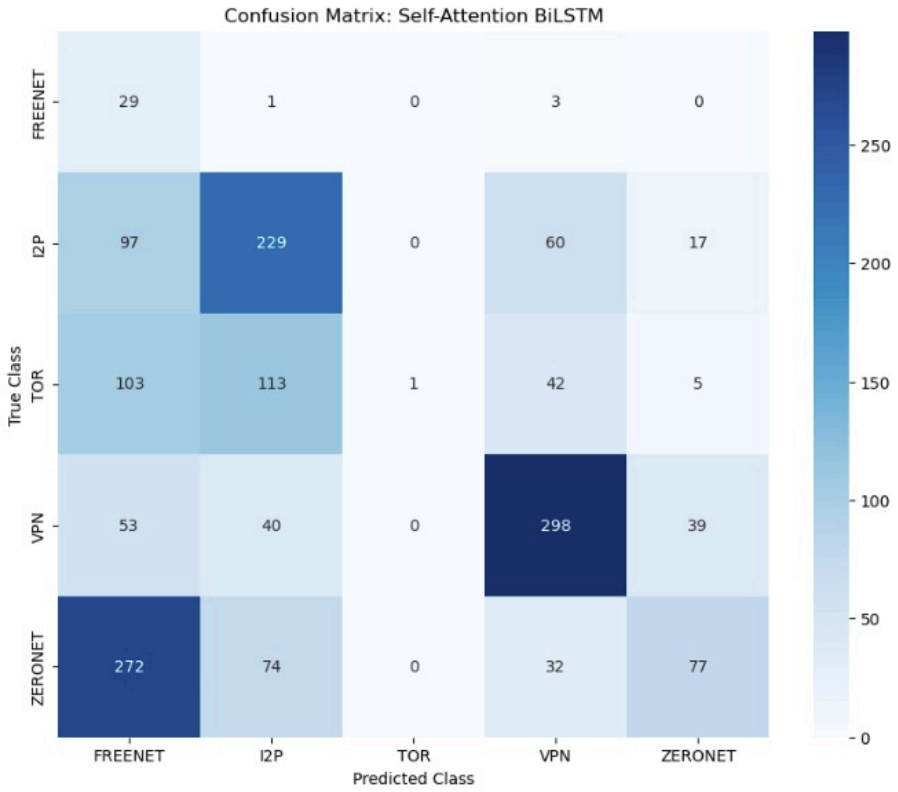

Figure 107: Confusion Matrix: Self-Attention BiLSTM

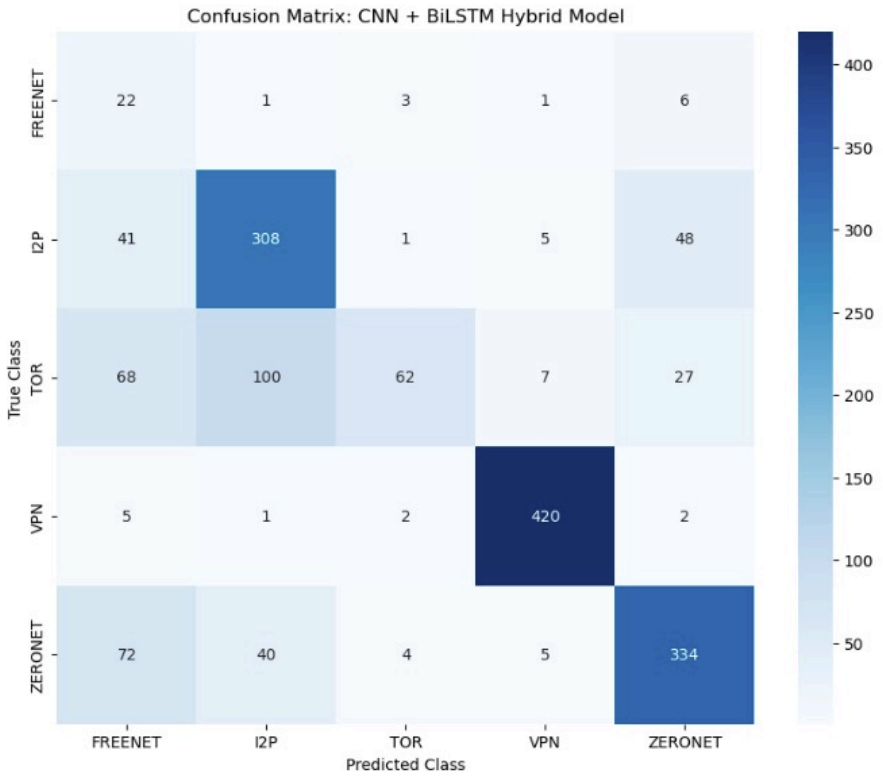

Figure 108: Confusion Matrix: CNN + BiLSTM Hybrid

6.2.5 Impact of Model Architecture on Performance

The fundamental differences in model architectures drive these performance trends:

Hybrid Attention + Tree-Based Models

Hybrid attention-based models integrated with tree-based learners, such as LightGBM, benefit significantly from attention-enhanced feature extraction combined with structured learning. The attention mechanism refines feature importance by emphasizing critical attributes, which enhances classification accuracy when coupled with the structured processing capabilities of tree-based models. These models achieve consistent performance across all classes, surpassing deep learning approaches, particularly in handling structured tabular data. LightGBM demonstrates the best trade-off between precision and recall among various tree-based methods, making it the most effective choice overall. Its ability to efficiently handle large datasets while maintaining interpretability further solidifies its suitability for encrypted traffic classification.

Deep Learning and Hybrid Models (BiLSTM, CNN + BiLSTM)

Deep learning and hybrid models, including BiLSTM and CNN + BiLSTM architectures, face challenges in encrypted traffic classification due to the high dimensionality of network traffic features and the lack of meaningful sequential dependencies. The inherent nature of encrypted traffic makes it difficult for sequential models to extract temporal relationships effectively. Self-Attention BiLSTM, in particular, fails to generalize well across different traffic types, leading to extreme misclassifications in certain classes. While CNN + BiLSTM exhibits slight improvements over standalone BiLSTM, it still underperforms compared to tree-based models, which are better suited for structured tabular data. These findings highlight the limitations of deep learning models in this domain, emphasizing the need for more effective feature extraction and classification strategies.

Comparison with Prior Work

Previous research on deep learning-based traffic classification has yielded mixed results, with notable success in high-volume traffic applications but significant shortcomings in fine-grained classification tasks such as encrypted traffic detection. The limitations stem from the inability of deep learning models to effectively capture structured tabular relationships and handle the complexity of encrypted traffic patterns. In contrast, this thesis highlights the effectiveness of hybrid models that integrate feature engineering with robust tree-based classifiers. The results demonstrate that these hybrid approaches consistently outperform pure deep learning architectures, reinforcing their superiority in achieving accurate and reliable encrypted traffic classification.

6.2.6 Key Insights

The results of this thesis reaffirm the superiority of the **Hybrid Attention + LightGBM** model in encrypted traffic classification, achieving the highest overall accuracy, macro F1-score, and stable recall across all classes. Combining tree-based learning with attention-enhanced feature selection proved highly effective in distinguishing complex traffic patterns, making it the most reliable classification framework.

Although **XGBoost and CatBoost** demonstrated competitive performance, they exhibited minor weaknesses in recall balance, particularly for certain encrypted traffic classes. This indicates that while these Gradient-boosting models are strong alternatives, they fall slightly short of the robustness displayed by the Hybrid Attention + LightGBM

approach. The slight recall inconsistencies suggest that these models may require additional tuning or hybridization with feature enhancement mechanisms to achieve a comparable level of performance.

In contrast, **deep learning-based models**, particularly BiLSTM and CNN + BiLSTM, struggled to maintain reliable classification performance. The sequential nature of these architectures limited their effectiveness in capturing the complex and non-sequential relationships present in encrypted traffic data. This highlights a fundamental limitation of purely deep learning-based approaches in this domain, emphasizing the need for structured learning techniques that better align with the characteristics of encrypted network traffic.

One of the most impactful findings of this thesis is the **critical role of multi-head attention in feature extraction**. The attention mechanism significantly improved recall, especially for challenging classes such as **TOR and ZERONET**, which often exhibit overlapping feature distributions with other traffic types. By enhancing feature representation, the multi-head attention mechanism enabled the classifier to differentiate subtle variations between encrypted traffic categories better, thereby reducing misclassifications.

Overall, the study reinforces the **dominance of tree-based hybrid models** for encrypted traffic classification. As seen in the Hybrid Attention + LightGBM model, the structured learning approach continues to outperform deep learning alternatives, proving that integrating attention-enhanced feature selection with Gradient-boosting techniques is the most effective strategy for this classification task. This conclusion underscores the need to leverage structured models that can process original and transformed features while maintaining interpretability and computational efficiency.

6.3 Concluding Remarks

This chapter presented a two-part analysis—first, the comparative impact of optimization techniques guided by SHAP and LIME, and second, the benchmarking of the proposed model against alternative architectures. Among the three refinements applied—weight tuning, SMOTE augmentation, and class-specific threshold adjustments—the latter emerged as the most impactful strategy, significantly improving recall for I2P and TOR without compromising overall accuracy. By fine-tuning decision boundaries for individual classes, threshold adjustments reduced misclassifications and enhanced the model's ability to distinguish between overlapping encrypted traffic categories. SMOTE augmentation was particularly beneficial for the underrepresented FREENET class, while VPN weight tuning successfully mitigated class dominance.

The results of this thesis underscore the effectiveness of the Hybrid Attention + LightGBM model in encrypted traffic classification. The integration of multi-head attention for feature enhancement and the structured learning capabilities of LightGBM enabled the model to outperform all other tested architectures in terms of accuracy, macro F1-score, and recall balance across classes. Notably, the optimized model improved macro-average precision from 0.90 to 0.91 while maintaining an overall accuracy of 93%, reflecting a strong equilibrium between precision and recall. These enhancements were consistently validated through cross-validation, reinforcing the model's generalizability to unseen data.

The inclusion of both original and attention-enhanced features proved essential in preserving robust classification performance, as the attention mechanism effectively captured complex feature interactions. Compared to deep learning and other boosting-based hybrids, the proposed model demonstrated superior class-wise consistency, especially in accurately identifying challenging traffic types such as Zeronet and I2P. These findings highlight the importance of adaptive threshold tuning in encrypted traffic classification, offering fine-grained control over classification decisions while ensuring overall model stability.

In summary, the Hybrid Attention + LightGBM model—particularly in its class-specific threshold tuning configuration—sets a new benchmark in encrypted traffic classification. The strategic refinements and interpretability-driven analysis demonstrate its capacity to handle complex, imbalanced, and overlapping traffic patterns. Moving forward, continuous improvements in adaptability, computational efficiency, and real-time scalability will further solidify its role in next-generation network security solutions.

The next **Chapter 7 - Conclusion & Future Work** summarizes the key findings from this thesis and discusses potential future research directions. It highlights how the proposed framework contributes to encrypted traffic classification and outlines opportunities for further improving classification performance, generalizability, and interoperability.

7 Conclusion & Future Work

Ensuring secure data communication through encryption is vital, but effectively detecting malicious activities requires the ability to differentiate between encrypted and non-encrypted traffic. Traditional methods for encrypted traffic classification, such as rule-based systems and conventional machine learning approaches, often face scalability, generalization, and class imbalance challenges, resulting in suboptimal performance. To address these limitations, this thesis introduces a Hybrid Attention + LightGBM model with class-specific threshold tuning, offering a more robust and accurate approach to encrypted traffic classification. By integrating multi-head attention mechanisms with tree-based classification, the proposed approach effectively enhances feature representation while maintaining computational efficiency and interpretability.

The experimental evaluation demonstrated that the Hybrid Attention + LightGBM model achieves state-of-the-art performance, surpassing deep learning and traditional gradient boosting methods. With an accuracy of 93%, it outperforms baselines such as XGBoost and CatBoost. The findings highlight that tree-based models outperform deep learning approaches for encrypted traffic classification because they capture structured relationships within the dataset, particularly when enhanced with attention-fused features. Furthermore, class-specific threshold tuning emerged as a crucial optimization, significantly improving recall for underrepresented classes such as FREENET, and mitigating the imbalances that often hinder minority-class detection.

The study also highlights the importance of feature interpretability in cybersecurity applications. SHAP-based feature importance analysis provided insights into the key attributes influencing classification, while LIME facilitated the examination of misclassified instances. These methods improved model transparency and guided refinements that reduced misclassification rates. Despite these improvements, challenges remain in distinguishing certain encrypted traffic types due to feature overlaps, particularly in cases where multiple encryption techniques share common statistical patterns. The Hybrid Attention + LightGBM model presents a well-balanced solution between accuracy, interpretability, and computational efficiency. It is highly suitable for deployment in real-world network security frameworks such as intrusion detection systems and security monitoring platforms.

Looking ahead, this thesis opens several promising avenues for future exploration. While the current approach effectively classifies encrypted traffic, improving feature representation through adaptive selection techniques remains an essential direction. Future work could explore reinforcement learning-based feature selection to adjust feature importance in real-time dynamically, optimizing classification performance across diverse traffic conditions. Additionally, more sophisticated data augmentation techniques, such as Generative Adversarial Networks (GANs) or Variational Autoencoders (VAEs), could be investigated to generate synthetic samples for underrepresented traffic types, enhancing the model's ability to generalize to novel encryption patterns.

Another important direction is exploring hybrid architectures integrating deep learning and tree-based models. Given that tree-based classifiers demonstrated superior performance, future efforts could examine how CNN/LSTM-based feature extractors can complement structured learning models such as LightGBM. Graph Neural Networks (GNNs) present another promising approach, as modeling encrypted traffic flows as network graphs could improve classification accuracy by capturing latent structural dependencies in network behavior. Furthermore, expanding the classification scope to include more fine-grained categories, such as specific VPN providers or obfuscated application types, would further enhance the model's utility in identifying emerging threats.

Ensuring model adaptability to novel obfuscation strategies is critical as encryption techniques evolve. Future research should explore continual learning approaches that allow models to adapt to new encrypted traffic patterns

incrementally, reducing the need for periodic retraining. Additionally, adversarial robustness must be further investigated to counter evolving obfuscation strategies that seek to evade detection. Incorporating adversarial training mechanisms could improve the resilience of encrypted traffic classifiers against sophisticated evasion techniques.
Scalability and real-time deployment also remain crucial aspects for future exploration. Implementing the model within high-throughput network environments, such as enterprise security infrastructures and large-scale intrusion detection systems, will validate its practical applicability under real-world constraints. Moreover, developing lightweight model variants for deployment on edge devices and resource-constrained environments like IoT networks would further extend its usability.
In conclusion, this thesis introduced a **novel Hybrid Attention + LightGBM model** designed specifically for **encrypted traffic classification**, effectively addressing key challenges such as **class imbalance**, **model interpretability**, and **classification accuracy**. By combining **multi-head attention mechanisms** with the efficiency of a tree-based **LightGBM classifier**, the model achieved enhanced **feature representation** while maintaining **computational efficiency** and **transparency**. Extensive experimental evaluation confirmed the model's superiority over traditional machine learning and deep learning baselines, achieving an overall **accuracy of 93%** and exhibiting consistently strong **macro-average recall and precision** across diverse encrypted traffic types.
Class-specific threshold tuning emerged as the most impactful refinement, particularly enhancing the classification of minority and overlapping classes such as **I2P** and **Zeronet**.
SMOTE-based augmentation and **VPN weight adjustment** provided additional performance gains, albeit to a lesser extent.
The use of **SHAP and LIME** frameworks was pivotal for:

- Providing global and local model interpretability
- Informing targeted refinements during optimization
- Diagnosing feature-level causes of misclassification

These tools revealed the importance of both **handcrafted features** and **attention-derived representations**, reinforcing the validity of the hybrid architecture and ensuring **explainable decision-making**—a vital requirement in **cybersecurity**, where **trust, auditability**, and **transparency** are non-negotiable.
Practical Implications: The proposed model is **well-suited for deployment** within a range of cybersecurity applications, including:

- **Network monitoring solutions**
- **Intrusion detection systems**
- **Cybersecurity intelligence platforms**

Its balance of high **predictive performance** and **interpretability** enables trustworthy **explainable AI** deployment in real-world scenarios.The modular, extensible design allows the model to be adapted across heterogeneous traffic datasets and evolving encryption protocols.
Future Research Directions:

- Investigating **adaptive and reinforcement learning-based feature selection** to dynamically optimize model behavior in real-time.

- Employing **Generative Adversarial Networks (GANs)** or **Variational Autoencoders (VAEs)** for data augmentation to alleviate class imbalance and enhance generalization.
- Developing deeper hybrid architectures by combining **CNN/LSTM-based deep extractors** with high-performance structured classifiers such as LightGBM.
- Exploring **Graph Neural Networks (GNNs)** to capture latent structural relationships in encrypted traffic flows.
- Enhancing model resilience through **Adversarial training techniques** and **Continual learning frameworks** for lifelong adaptation to evolving threats
- Translating the model into real-time deployments in **enterprise networks**, **IoT ecosystems**, and high-throughput security environments.
- Designing lightweight model variants optimized for **edge devices**, enabling secure, distributed, and resource-efficient network monitoring.

This thesis lays a strong foundation for the next generation of **interpretable, robust, and high-performing encrypted traffic classifiers**. By bridging **methodological innovation** with **practical cybersecurity applications**, it contributes meaningfully to the development of trustworthy and adaptive network defense systems.

Bibliography

[1] VPN Protocols Explained and Compared — avast.com. https://www.avast.com/c-vpn-protocols. [Accessed 22-01-2025].

[2] Dave Albaugh. What is Tor? How to use it safely and legally (plus 5 Tor alternatives) — comparitech.com. https://www.comparitech.com/blog/vpn-privacy/ultimate-guide-to-tor/. [Accessed 22-01-2025].

[3] A Gentle Introduction to How I2P Works - I2P — geti2p.net. https://geti2p.net/en/docs/how/intro. [Accessed 22-01-2025].

[4] Jianwei Ding, Xiaoyu Guo, and Zhouguo Chen. Big data analyses of zeronet sites for exploring the new generation darkweb. In *Proceedings of the 3rd International Conference on Software Engineering and Information Management*, pages 46–52, 2020.

[5] Ian Clarke. The dark side of the internet: What is freenet?, 2000. Accessed: Feb 1, 2025.

[6] Oracle Documentation. Understanding https communication, 2025. Accessed: Feb 1, 2025.

[7] High Performance Browser Networking. Transport layer security (tls), 2025. Accessed: Feb 1, 2025.

[8] GeeksforGeeks. Secure socket layer (ssl), 2025. Accessed: Feb 1, 2025.

[9] GeeksforGeeks. Ipsec architecture, 2025. Accessed: Feb 1, 2025.

[10] GeeksforGeeks. Secure shell (ssh) architecture, 2025. Accessed: Feb 1, 2025.

[11] JavaTpoint. Computer network - pretty good privacy (pgp), 2025. Accessed: Feb 1, 2025.

[12] Network Encyclopedia. Secure/multipurpose internet mail extensions (s/mime), 2025. Accessed: Feb 1, 2025.

[13] GoAnywhere. Openpgp encryption process, 2025. Accessed: Feb 1, 2025.

[14] SysOps Technix. Wpa2-enterprise: Secure your organization's wi-fi network, 2025. Accessed: Feb 1, 2025.

[15] F. Zaki, F. Afifi, S. Abd Razak, A. Gani, and N. B. Anuar. Grain: Granular multi-label encrypted traffic classification using classifier chain. *Computer Networks*, 213(24):109084, 2022.

[16] S. Dong. Multi class svm algorithm with active learning for network traffic classification. *Expert Systems with Applications*, 176:114885, 2021.

[17] Pitpimon Choorod, George Weir, and Anil Fernando. Classifying tor traffic encrypted payload using machine learning. *IEEE Access*, 2024.

[18] S. Lv, C. Wang, Z. Wang, S. Wang, B. Wang, and Y. Zhang. Aae-dsvdd: A one-class classification model for vpn traffic identification. *Computer Networks*, 236:109990, 2023.

[19] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé. Distiller: Encrypted traffic classification via multimodal multitask deep learning. *Journal of Network and Computer Applications*, 183(31):102985, 2021.

[20] Z. Song, Z. Zhao, F. Zhang, G. Xiong, G. Cheng, X. Zhao, and B. Chen. I ^{2} rnn: An incremental and interpretable recurrent neural network for encrypted traffic classification. *IEEE Transactions on Dependable and Secure Computing*, (56), 2023.

[21] K. Zhou, W. Wang, C. Wu, and T. Hu. Practical evaluation of encrypted traffic classification based on a combined method of entropy estimation and neural networks. *ETRI Journal*, 42(3):311–323, 2020.

[22] M. H. Pathmaperuma, Y. Rahulamathavan, S. Dogan, and A. M. Kondoz. Deep learning for encrypted traffic classification and unknown data detection. *Sensors*, 22(19):7643, 2022.

[23] A. Rasteh, F. Delpech, C. Aguilar-Melchor, R. Zimmer, S. B. Shouraki, and T. Masquelier. Encrypted internet traffic classification using a supervised spiking neural network. *Neurocomputing*, 503:272–282, 2022.

[24] Y. Xu, J. Cao, K. Song, Q. Xiang, and G. Cheng. Fasttraffic: A lightweight method for encrypted traffic fast classification. *Computer Networks*, 235:109965, 2023.

[25] Y. He and W. Li. Image-based encrypted traffic classification with convolution neural networks. In *2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*, number 29, pages 271–278. IEEE, 2020.

[26] Rodrigo Moreira, Larissa Ferreira Rodrigues Moreira, and Flávio de Oliveira Silva. An intelligent network monitoring approach for online classification of darknet traffic. *Computers and Electrical Engineering*, 110:108852, 2023.

[27] J. Cheng, Y. Wu, E. Yuepeng, J. You, T. Li, H. Li, and J. Ge. Matec: A lightweight neural network for online encrypted traffic classification. *Computer Networks*, 199(20):108472, 2021.

[28] M. Lotfollahi, M. Jafari Siavoshani, R. Shirali Hossein Zade, and M. Saberian. Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Computing*, 24(3):1999–2012, 2020.

[29] M. Wang, K. Zheng, D. Luo, Y. Yang, and X. Wang. An encrypted traffic classification framework based on convolutional neural networks and stacked autoencoders. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, number 22, pages 634–641. IEEE, 2020.

[30] S. Soleymanpour, H. Sadr, and M. Nazari Soleimandarabi. Cscnn: cost-sensitive convolutional neural network for encrypted traffic classification. *Neural Processing Letters*, 53(5):3497–3523, 2021.

[31] L. Xu, D. Dou, and H. J. Chao. Etcnet: encrypted traffic classification using siamese convolutional networks. In *Proceedings of the Workshop on Network Application Integration/CoDesign*, number 42, pages 51–53, 2020.

[32] C. Y. Lin, B. Chen, and W. Lan. An efficient approach for encrypted traffic classification using cnn and bidirectional gru. In *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, number 43, pages 368–373. IEEE, 2022.

[33] S. Izadi, M. Ahmadi, and R. Nikbazm. Network traffic classification using convolutional neural network and ant-lion optimization. *Computers and Electrical Engineering*, 101:108024, 2022.

[34] A. Habibi Lashkari, G. Kaur, and A. Rahali. Didarknet: A contemporary approach to detect and characterize the darknet traffic using deep image learning. In *2020 the 10th international conference on communication and network security*, pages 1–13, 2020.

[35] X. Tong, X. Tan, L. Chen, J. Yang, and Q. Zheng. Bfsn: a novel method of encrypted traffic classification based on bidirectional flow sequence network. In *2020 3rd International Conference on Hot Information-Centric Networking (HotICN)*, number 37, pages 160–165. IEEE, 2020.

[36] J. Yang and Y. Guo. Aefeta: Encrypted traffic classification framework based on self-learning of feature. In *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, number 48, pages 876–880. IEEE, 2021.

[37] X. Hu, C. Gu, Y. Chen, and F. Wei. tcld-net: a transfer learning internet encrypted traffic classification scheme based on convolution neural network and long short-term memory network. In *2021 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, number 53, pages 1–5. IEEE, 2021.

[38] F. Meslet-Millet, E. Chaput, and S. Mouysset. Sppnet: An approach for real-time encrypted traffic classification using deep learning. In *2021 IEEE Global Communications Conference (GLOBECOM)*, number 44, pages 1–6. IEEE, 2021.

[39] W. Maonan, Z. Kangfeng, X. Ning, Y. Yanqing, and W. Xiujuan. Centime: a direct comprehensive traffic features extraction for encrypted traffic classification. In *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*, number 45, pages 490–498. IEEE, 2021.

[40] X. Ma, W. Zhu, J. Wei, Y. Jin, D. Gu, and R. Wang. Eetc: An extended encrypted traffic classification algorithm based on variant resnet network. *Computers & Security*, 128(49):103175, 2023.

[41] X. Hu, C. Gu, Y. Chen, and F. Wei. Cbd: A deep-learning-based scheme for encrypted traffic classification with a general pre-training method. *Sensors*, 21(24):8231, 2021.

[42] Peng Zhu, Gang Wang, Jingheng He, Yueli Dong, and Yu Chang. An encrypted traffic identification method based on multi-scale feature fusion. *Array*, page 100338, 2024.

[43] Z. Diao, G. Xie, X. Wang, R. Ren, X. Meng, G. Zhang, and M. Qiao. Ec-gcn: A encrypted traffic classification framework based on multi-scale graph convolution networks. *Computer Networks*, 224(57):109614, 2023.

[44] Y. Hong, Q. Li, Y. Yang, and M. Shen. Graph based encrypted malicious traffic detection with hybrid analysis of multi-view features. *Information Sciences*, page 119229, 2023.

[45] L. Wang, X. Ma, N. Li, Q. Lv, Y. Wang, W. Huang, and H. Chen. Tgprint: Attack fingerprint classification on encrypted network traffic based graph convolution attention networks. *Computers & Security*, 135:103466, 2023.

[46] Xinbo Han, Guizhong Xu, Meng Zhang, Zheng Yang, Ziyang Yu, Weiqing Huang, and Chen Meng. De-gnn: Dual embedding with graph neural network for fine-grained encrypted traffic classification. *Computer Networks*, 245:110372, 2024.

[47] Haozhen Zhang, Xi Xiao, Le Yu, Qing Li, Zhen Ling, and Ye Zhang. One train for two tasks: An encrypted traffic classification framework using supervised contrastive learning. *arXiv preprint arXiv:2402.07501*, 2024.

[48] Jin Yang, Xinyun Jiang, Yulin Lei, Weiheng Liang, Zicheng Ma, and Siyu Li. Mtsecurity: Privacy-preserving malicious traffic classification using graph neural network and transformer. *IEEE Transactions on Network and Service Management*, 2024.

[49] Z. Tang, J. Wang, B. Yuan, H. Li, J. Zhang, and H. Wang. Markov-gan: Markov image enhancement method for malicious encrypted traffic classification. *IET Information Security*, 16(6):442–458, 2022.

[50] P. Wang, S. Li, F. Ye, Z. Wang, and M. Zhang. Packetcgan: Exploratory study of class imbalance for encrypted traffic classification using cgan. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, number 32, pages 1–7. IEEE, 2020.

[51] Y. Sanjalawe and S. Fraihat. Detection of obfuscated tor traffic based on bidirectional generative adversarial networks and vision transform. *Computers & Security*, 135:103512, 2023.

[52] P. Wang, Z. Wang, F. Ye, and X. Chen. Bytesgan: A semi-supervised generative adversarial network for encrypted traffic classification in sdn edge gateway. *Computer Networks*, 200(39):108535, 2021.

[53] R. Zhao, X. Deng, Z. Yan, J. Ma, Z. Xue, and Y. Wang. Mt-flowformer: A semi-supervised flow transformer for encrypted traffic classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, number 40, pages 2576–2584, 2022.

[54] Xinjie Lin, Gang Xiong, Gaopeng Gou, Zhen Li, Junzheng Shi, and Jing Yu. ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 633–642. ACM, 2022.

[55] Hong Huang, Xingxing Zhang, Ye Lu, Ze Li, and Shaohua Zhou. Bstfnet: An encrypted malicious traffic classification method integrating global semantic and spatiotemporal features. *Computers, Materials & Continua*, 78(3), 2024.

[56] Jee-Tae Park, Chang-Yui Shin, Ui-Jun Baek, and Myung-Sup Kim. Fast and accurate multi-task learning for encrypted network traffic classification. *Applied Sciences*, 14(7):3073, 2024.

[57] B. Xu, G. He, and H. Zhu. Me-box: A reliable method to detect malicious encrypted traffic. *Journal of Information Security and Applications*, 59:102823, 2021.

[58] Y. Hu, F. Zou, L. Li, and P. Yi. Traffic classification of user behaviors in tor, i2p, zeronet, freenet. In *2020 IEEE 19th international conference on trust, security and privacy in computing and communications (TrustCom)*, pages 418–424. IEEE, 2020.

[59] N. Rust-Nguyen, S. Sharma, and M. Stamp. Darknet traffic classification and adversarial attacks using machine learning. *Computers & Security*, 127:103098, 2023.

[60] N. Malekghaini, E. Akbari, M. A. Salahuddin, N. Limam, R. Boutaba, B. Mathieu, and S. Tuffin. Automl4etc: Automated neural architecture search for real-world encrypted traffic classification. *IEEE Transactions on Network and Service Management*, (23), 2023.

[61] R. T. Elmaghraby, N. M. A. Aziem, M. A. Sobh, and A. M. Bahaa-Eldin. Encrypted network traffic classification based on machine learning. *Ain Shams Engineering Journal*, 15(2):102361, 2024.

[62] P. Luo, J. Chu, and G. Yang. Ip packet-level encrypted traffic classification using machine learning with a light weight feature engineering method. *Journal of Information Security and Applications*, 75:103519, 2023.

[63] Xinge Yan, Liukun He, Yifan Xu, Jiuxin Cao, Liangmin Wang, and Guyang Xie. High-speed encrypted traffic classification by using payload features. *Digital Communications and Networks*, 2024.

[64] Jianjin Zhao, Qi Li, Yueping Hong, and Meng Shen. Metarocketc: Adaptive encrypted traffic classification in complex network environments via time series analysis and meta-learning. *IEEE Transactions on Network and Service Management*, 2024.

[65] Xiang Li, Jiang Xie, Qige Song, Yafei Sang, Yongzheng Zhang, Shuhao Li, and Tianning Zang. Let model keep evolving: Incremental learning for encrypted traffic classification. *Computers & Security*, 137:103624, 2024.

[66] Rui Wang, Zhidong Wu, Yaxi Li, Feng Li, Siyuan Tian, and Jianyi Liu. Encrypted traffic classification based on contrastive learning with spatial-temporal feature fusion. In *International Conference on Computer Application and Information Security (ICCAIS 2023)*, volume 13090, pages 531–538. SPIE, 2024.

[67] Guanyu Wang and Yijun Gu. Multi-task scenario encrypted traffic classification and parameter analysis. *Sensors*, 24(10):3078, 2024.

[68] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C Claffy, Marco Chiesa, Michele Russo, and Antonio Pescapé. Analysis of country-wide internet outages caused by censorship. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 1–18, 2011.

[69] Google. Google Transparency Report — transparencyreport.google.com. `https://transparencyreport.google.com/https/overview?hl=en`, 2024.

[70] Charles Wright, Fabian Monrose, and Gerald M Masson. Hmm profiles for network traffic classification. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 9–15, 2004.

[71] Zscaler. Zscaler ThreatLabz 2023 State of Encrypted Attacks Report — zscaler.com. `https://www.zscaler.com/resources/2023-threatlabz-state-of-encrypted-attacks-report`, 2023. [Accessed 11-11-2024].

[72] Marc Handelman. USENIX Security '23 - Rosetta: Enabling Robust TLS Encrypted Traffic Classification in Diverse Network Environments with TCP-Aware Traffic Augmentation — securityboulevard.com. `https://securityboulevard.com/2024/01/usenix-security-23-rosetta-enabling-robust-tls-encrypted-traffic-\`

classification-in-diverse-network-environments-\with-tcp-aware-traffic-augmentation/, 2023. [Accessed 11-11-2024].

[73] Mahmoud Abbasi, Amin Shahraki, and Amir Taherkordi. Deep learning for network traffic monitoring and analysis (ntma): A survey. *Computer Communications*, 170:19–41, 2021.

[74] Blake Anderson, Subharthi Paul, and David McGrew. Deciphering malware's use of tls (without decryption). *Journal of Computer Virology and Hacking Techniques*, 14:195–211, 2018.

[75] Manuel Lopez-Martin, Belen Carro, Antonio Sanchez-Esguevillas, and Jaime Lloret. Network traffic classifier with convolutional and recurrent neural networks for internet of things. *IEEE access*, 5:18042–18050, 2017.

[76] Aamer Nadeem and M Younus Javed. A performance comparison of data encryption algorithms. In *2005 international Conference on information and communication technologies*, pages 84–89. IEEE, 2005.

[77] Jie Lin, Yinbin Miao, Linfeng Wei, Tao Leng, and Kim-Kwang Raymond Choo. Efficient secure inference scheme in multiparty settings for industrial internet of things. *IEEE Transactions on Industrial Informatics*, 2024.

[78] Jun Feng, Laurence T Yang, Bocheng Ren, Deqing Zou, Mianxiong Dong, and Shunli Zhang. Tensor recurrent neural network with differential privacy. *IEEE Transactions on Computers*, 73(3):683–693, 2023.

[79] Pengfei Zhang, Xiang Cheng, Sen Su, and Ning Wang. Task allocation under geo-indistinguishability via group-based noise addition. *IEEE Transactions on Big Data*, 9(3):860–877, 2022.

[80] Li Da Xu, Yang Lu, and Ling Li. Embedding blockchain technology into iot for security: A survey. *IEEE Internet of Things Journal*, 8(13):10452–10473, 2021.

[81] Hong-Ning Dai, Zibin Zheng, and Yan Zhang. Blockchain for internet of things: A survey. *IEEE internet of things journal*, 6(5):8076–8094, 2019.

[82] Ana Reyna, Cristian Martín, Jaime Chen, Enrique Soler, and Manuel Díaz. On blockchain and its integration with iot. challenges and opportunities. *Future generation computer systems*, 88:173–190, 2018.

[83] Khalid Shahbar and A Nur Zincir-Heywood. Traffic flow analysis of tor pluggable transports. In *2015 11th International Conference on Network and Service Management (CNSM)*, pages 178–181. IEEE, 2015.

[84] Khalid Shahbar and A Nur Zincir-Heywood. Benchmarking two techniques for tor classification: Flow level and circuit level classification. In *2014 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, pages 1–8. IEEE, 2014.

[85] Antonio Montieri, Domenico Ciuonzo, Giuseppe Aceto, and Antonio Pescape. Anonymity services tor, i2p, jondonym: classifying in the dark (web). *IEEE Transactions on Dependable and Secure Computing*, 17(3):662–675, 2018.

[86] Antonio Montieri, Domenico Ciuonzo, Giampaolo Bovenzi, Valerio Persico, and Antonio Pescapé. A dive into the dark web: Hierarchical traffic classification of anonymity tools. *IEEE transactions on network science and engineering*, 7(3):1043–1054, 2019.

[87] Gurdeep Singh-Pall. Virtual private network. https://www.microsoft.com, 1996. Developed by Microsoft Corporation.

[88] Paul Syverson, Michael G. Reed, and David Goldschlag. The onion router. https://www.torproject.org, 2002. Developed at the U.S. Naval Research Laboratory.

[89] Invisible internet project. https://geti2p.net, 2003. A decentralized anonymizing network.

[90] Tamas Kocsis. Zeronet. https://zeronet.io, 2015. Decentralized websites using Bitcoin cryptography and BitTorrent network.

[91] Ian Clarke. Freenet. https://freenetproject.org, 2000. A peer-to-peer platform for censorship-resistant communication.

[92] Netscape. Hypertext transfer protocol secure. https://www.netscape.com, 1995. An extension of HTTP for secure communication over a computer network.

[93] Transport layer security. https://tools.ietf.org/html/rfc5246, 1999. Protocol for secure communications over a computer network.

[94] Netscape. Secure sockets layer. https://www.netscape.com, 1995. Protocol for establishing encrypted links between networked computers.

[95] Internet protocol security. https://tools.ietf.org/html/rfc4301, 1995. A suite of protocols for securing Internet Protocol communications.

[96] Tatu Ylönen. Secure shell. https://www.ssh.com/ssh/protocol/, 1995. Protocol for secure network services over an unsecured network.

[97] Phil Zimmermann. Pretty good privacy. https://www.pgpi.org, 1991. Program used for encrypting and decrypting data.

[98] Secure/multipurpose internet mail extensions. https://tools.ietf.org/html/rfc5751, 1995. Standard for public key encryption and signing of MIME data.

[99] Openpgp. https://tools.ietf.org/html/rfc4880, 1997. An open standard for encrypting and signing data.

[100] Wi-fi protected access 2. https://www.wi-fi.org/discover-wi-fi/security, 2004. A security protocol for Wi-Fi networks.

[101] Z. Yao, J. Ge, Y. Wu, X. Lin, R. He, and Y. Ma. Encrypted traffic classification based on gaussian mixture models and hidden markov models. *Journal of Network and Computer Applications*, 166:102711, 2020.

[102] Yi Zeng, Huaxi Gu, Wenting Wei, and Yantao Guo. $deep-full-range$: a deep learning based network encrypted traffic classification and intrusion detection framework. *IEEE Access*, 7:45182–45190, 2019.

[103] Shahbaz Rezaei and Xin Liu. How to achieve high classification accuracy with just a few labels: A semi-supervised approach using sampled packets. *arXiv preprint arXiv:1812.09761*, 2018.

[104] S. Jorgensen, J. Holodnak, J. Dempsey, K. de Souza, A. Raghunath, V. Rivet, and A. Wollaber. Extensible machine learning for encrypted network traffic application labeling via uncertainty quantification. *IEEE Transactions on Artificial Intelligence*, 2023.

[105] Giuseppe Aceto, Domenico Ciuonzo, Antonio Montieri, and Antonio Pescapé. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. *IEEE transactions on network and service management*, 16(2):445–458, 2019.

[106] Timothy O'shea and Jakob Hoydis. An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking*, 3(4):563–575, 2017.

[107] Meng Shen, Jinpeng Zhang, Liehuang Zhu, Ke Xu, and Xiaojiang Du. Accurate decentralized application identification via encrypted traffic analysis using graph neural networks. *IEEE Transactions on Information Forensics and Security*, 16:2367–2380, 2021.

[108] Ting-Li Huoh, Yan Luo, Peilong Li, and Tong Zhang. Flow-based encrypted network traffic classification with graph neural networks. *IEEE Transactions on Network and Service Management*, 20(2):1224–1237, 2022.

[109] Drew A Hudson and Larry Zitnick. Generative adversarial transformers. In *International conference on machine learning*, pages 4487–4499. PMLR, 2021.

[110] Jiangtao Zhai, Peng Lin, Yongfu Cui, Lilong Xu, and Ming Liu. Graphcwgan-gp: A novel data augmenting approach for imbalanced encrypted traffic classification. *CMES-Computer Modeling in Engineering & Sciences*, 136(2), 2023.

[111] Xiaochun Yun, Yipeng Wang, Yongzheng Zhang, Chen Zhao, and Zijian Zhao. Encrypted tls traffic classification on cloud platforms. *IEEE/ACM Transactions On Networking*, 31(1):164–177, 2022.

[112] Simone Disabato, Manuel Roveri, and Cesare Alippi. Distributed deep convolutional neural networks for the internet-of-things. *IEEE Transactions on Computers*, 70(8):1239–1252, 2021.

[113] M. C. Marim, P. V. B. Ramos, A. B. Vieira, A. Galletta, M. Villari, R. M. de Oliveira, and E. F. Silva. Darknet traffic detection and characterization with models based on decision trees and neural networks. *Intelligent Systems with Applications*, 18:200199, 2023.

[114] R. Xie, Y. Wang, J. Cao, E. Dong, M. Xu, K. Sun, and M. Zhang. Rosetta: Enabling robust tls encrypted traffic classification in diverse network environments with tcp-aware traffic augmentation. In *Proceedings of the ACM Turing Award Celebration Conference-China 2023*, number 46, pages 131–132, 2023.

[115] M. Seydali, F. Khunjush, B. Akbari, and J. Dogani. Cbs: A deep learning approach for encrypted traffic classification with mixed spatio-temporal and statistical features. *IEEE Access*, (36), 2023.

[116] K. Lin, X. Xu, and H. Gao. Tscrnn: A novel classification scheme of encrypted traffic based on flow spatiotemporal features for efficient management of iiot. *Computer Networks*, 190(41):107974, 2021.

[117] Z. Chen, G. Cheng, Z. Wei, and D. Niu. Classify traffic rather than flow: Versatile multi-flow encrypted traffic classification with flow clustering. *IEEE Transactions on Network and Service Management*, (38), 2023.

[118] S. Tian, Y. Gao, G. Yuan, R. Zhang, J. Zhao, and S. Zhang. An encrypted traffic classification method based on contrastive learning. In *Proceedings of the 8th International Conference on Communication and Information Processing*, number 52, pages 101–105, 2022.

[119] CTU University. CTU-13 Dataset, 2011.

[120] Andrew W. Moore and Denis Zuev. The Art of Network Traffic Classification: 10 Years After. *ACM SIGCOMM Computer Communication Review*, 35(3):133–146, 2005.

[121] A. Habibi Lashkari, G. Draper-Gil, M.S. Mamun, and A.A. Ghorbani. Characterization of tor traffic using time based features. In *International Conference on Information Systems Security and Privacy*, 2017.

[122] Gerard Draper-Gil, Arash Habibi Lashkari, Mohammad Saiful Islam Mamun, and Ali A Ghorbani. Characterization of encrypted and vpn traffic using time-related. In *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*, pages 407–414, 2016.

[123] University of Science and Technology of China. USTC-TFC2016, 2016.

[124] Wazen Shbair. BetterNet HTTPS, 2016.

[125] Khalid Shahbar and A Nur Zincir-Heywood. Anon17: Network traffic dataset of anonymity services. *Faculty of Computer Science Dalhousie University, Tech. Rep*, 2017.

[126] Khalid Shahbar and A Nur Zincir-Heywood. How far can we push flow analysis to identify encrypted anonymity network traffic? In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6. IEEE, 2018.

[127] Iman Sharafaldin, Arash Habibi Lashkari, Ali A Ghorbani, et al. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, 1:108–116, 2018.

[128] MohammadMoein Shafi, Arash Habibi Lashkari, and Arousha Haghighian Roudsari. Nlflowlyzer: Toward generating an intrusion detection dataset and intruders behavior profiling through network layer traffic analysis and pattern extraction. *Computers & Security*, page 104160, 2024.

[129] Chenggang Wang, Sean Kennedy, Haipeng Li, King Hudson, Gowtham Atluri, Xuetao Wei, Wenhai Sun, and Boyang Wang. Fingerprinting encrypted voice traffic on smart speakers with deep learning. In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 254–265, 2020.

[130] Giuseppe Aceto, Domenico Ciuonzo, Antonio Montieri, Valerio Persico, and Antonio Pescapè. Mirage: Mobile-app traffic capture and ground-truth creation, 2019.

[131] Iman Akbari, Mohammad A Salahuddin, Leni Ven, Noura Limam, Raouf Boutaba, Bertrand Mathieu, Stephanie Moteau, and Stephane Tuffin. A look behind the curtain: traffic classification in an increasingly encrypted web. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 5(1):1–26, 2021.

[132] Liang Xu and Dejing Dou. Oregon & Ohio HTTP2, 2020.

[133] Yuzong Hu, Futai Zou, Linsen Li, and Ping Yi. Traffic classification of user behaviors in tor, i2p, zeronet, freenet. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 418–424, 2020.

[134] University of California, Davis. UCDavis – QUIC, 2020.

[135] T. Van Ede and R. Bortolameotti. Browser 2020, 2020.

[136] Mohammadreza MontazeriShatoori, Logan Davidson, Gurdip Kaur, and Arash Habibi Lashkari. Detection of doh tunnels using time-series classification of encrypted traffic. In *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 63–70. IEEE, 2020.

[137] Andrey Ferriyan, Achmad Husni Thamrin, Keiji Takeda, and Jun Murai. Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic. *Applied Sciences*, 11(17), 2021.

[138] Ruijie Zhao, Yiteng Huang, Xianwen Deng, Zhi Xue, Jiabin Li, Zijing Huang, and Yijun Wang. Flow transformer: A novel anonymity network traffic classifier with attention mechanism. In *17th International Conference on Mobility, Sensing and Networking (MSN)*, pages 223–230, 2021.

[139] Ruijie Zhao, Xianwen Deng, Yanhao Wang, Libo Chen, Ming Liu, Zhi Xue, and Yijun Wang. Flow sequence-based anonymity network traffic identification with residual graph convolutional networks. In *IEEE/ACM International Symposium on Quality of Service (IWQoS)*, pages 1–10, 2022.

[140] Yuqiang Heng, Vikram Chandrasekhar, and Jeffrey G Andrews. Utmobilenettraffic2021: A labeled public network traffic dataset. *IEEE Networking Letters*, 3(3):156–160, 2021.

[141] Jan Luxemburk, Karel Hynek, Tomávs vCejka, Andrej Lukavcovivc, and Pavel vSivska. Cesnet-quic22: A large one-month quic network traffic dataset from backbone lines. *Data in Brief*, 46:108888, 2023.

[142] Chao Wang, Alessandro Finamore, Lixuan Yang, Kevin Fauvel, and Dario Rossi. Appclassnet: A commercial-grade dataset for application identification research. *ACM SIGCOMM Computer Communication Review*, 52(3):19–27, 2022.

[143] Cooper Coldwell, Denver Conger, Edward Goodell, Brendan Jacobson, Bryton Petersen, Damon Spencer, Matthew Anderson, and Matthew Sgambati. Machine learning 5g attack detection in programmable logic. In *2022 IEEE Globecom Workshops (GC Wkshps)*, pages 1365–1370, 2022.

[144] Euclides Carlos Pinto Neto, Sajjad Dadkhah, Raphael Ferreira, Alireza Zohourian, Rongxing Lu, and Ali A Ghorbani. Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment. *Sensors*, 23(13):5941, 2023.

[145] Steven Jorgensen, John Holodnak, Jensen Dempsey, Karla de Souza, Ananditha Raghunath, Vernon Rivet, Noah DeMoes, Andrés Alejos, and Allan Wollaber. Extensible machine learning for encrypted network traffic application labeling via uncertainty quantification. *IEEE Transactions on Artificial Intelligence*, 5(1):420–433, 2024.

[146] Zulong Diao, Gaogang Xie, Xin Wang, Rui Ren, Xuying Meng, Guangxing Zhang, Kun Xie, and Mingyu Qiao. Ec-gcn: A encrypted traffic classification framework based on multi-scale graph convolution networks. *Computer Networks*, 224:109614, 2023.

[147] MohammadMoein Shafi, Arash Habibi Lashkari, Vicente Rodriguez, and Ron Nevo. Toward generating a new cloud-based distributed denial of service (ddos) dataset and cloud intrusion traffic characterization. *Information*, 15(4):195, 2024.

[148] Arash Habibi Lashkari, Gurdip Kaur, and Abir Rahali. Didarknet: A contemporary approach to detect and characterize the darknet traffic using deep image learning. In *Proceedings of the 2020 10th International Conference on Communication and Network Security*, ICCNS '20, page 1–13, New York, NY, USA, 2021. Association for Computing Machinery.

[149] Kevin P Dyer, Scott E Coull, Thomas Ristenpart, and Thomas Shrimpton. Protocol misidentification made easy with format-transforming encryption. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 61–72, 2013.

Table 35: List of acronyms/abbreviations used in the thesis

Abbreviations	Full form/Definition	Abbreviations	Full form/Definition
ETC	Encrypted Traffic Classification	ISCX	Information Security Center of Excellence
ML	Machine Learning	RADIUS	Remote Authentication Dial-In User Service
DL	Deep Learning	PCAP	Packet Capture
ETA	Encrypted Traffic Analysis	IDS	Intrusion Detection System
CNN	Convolutional Neural Network	DDoS	Distributed Denial of Service
RNN	Recurrent Neural Network	DNS	Domain Name System
IE	Information Extractor	CMS	Content Management System
IEEE	Institute of Electrical and Electronics Engineers	SJTU	Shanghai Jiao Tong University
ACM	Association for Computing Machinery	AN	Autonomous Networks
SSRN	Social Science Research Network	CSTNET	China Science and Technology Network
MDPI	Multidisciplinary Digital Publishing Institute	MIT	Massachusetts Institute of Technology
TLS	Transport Layer Security	TCP	Transmission Control Protocol
SSL	Secure Sockets Layer	UDP	User Datagram Protocol
QUIC	Quick UDP Internet Connections	BCCC	Behaviour-Centric Cybersecurity Center
DPI	Deep Packet Inspection	CSV	Comma-Separated Values
LSTM	Long Short-Term Memory	ANN	Artificial Neural Network
VPN	Virtual Private Network	RL	Reinforcement Learning
HTTPS	Hypertext Transfer Protocol Secure	BIGAN	Bidirectional Generative Adversarial Network
TOR	The Onion Router	RF	Random Forest
HTTP	Hypertext Transfer Protocol	ALO	Ant Lion Optimizer
IP	Internet Protocol	SOM	Self-Organizing Maps
SSH	Secure Shell	GMM	Gaussian Mixture Model
PGP	Pretty Good Privacy	HMM	Hidden Markov Model
MIME	Multipurpose Internet Mail Extensions	OS	Operating System
WPA	Wi-Fi Protected Access	MCFP	Multi-Class Flow Prediction
SVM	Support Vector Machine	KNN	K-Nearest Neighbors
AAE	Adversarial Autoencoder	DT	Decision Tree
DSVDD	Deep Support Vector Data Description	LR	Logistic Regression
PNN	Probabilistic Neural Network	GBDT	Gradient Boosting Decision Trees
DNN	Deep Neural Network	XGB	Extreme Gradient Boosting
SNN	Spiking Neural Network	GBM	Gradient Boosting Machine
MLP	Multilayer Perceptron	NAS	Neural Architecture Search
SAE	Stacked Autoencoder	MAUC	Mean Area Under Curve
CSCNN	Cost Senstitive Convolutional Neural Network	LAN	Local Area Network
GRU	Gated Recurrent Unit	IOS	Internet Operating System
BFSN	Binary Forward Stochastic Neurons	SOFTMAX	Softmax Function
EETC	Extended Encrypted Traffic Classification	AUC	Area Under Curve
BERT	Bidirectional Encoder Representations from Transformers	RC	Recall
GAN	Generative Adversarial Network	FAR	False Alarm Rate
CGAN	Conditional GAN	FPR	False Positive Rate
SGAN	Semi-Supervised GAN	CLETFE	Cross Layer Encrypted Traffic Feature Extraction
SDN	Software-Defined Networking	ACGAN	Auxiliary Classifier GAN
BSTFN	Bidirectional Stacked Temporal Fusion Networks	MAML	Model-Agnostic Meta-Learning
USTC	University of Science and Technology of China	FIN	Finish Flag
GCN	Graph Convolutional Network	RST	Reset Flag
PEFT	Parameter-Efficient Fine-Tuning	ALF	Application Layer Filtering
DE-GNN	Dynamic Edge Graph Neural Network	ICMP	Internet Control Message Protocol
GNN	Graph Neural Network	VRRP	Virtual Router Redundancy Protocol
TIG	Traffic Interaction Graph	ARP	Address Resolution Protocol
CLE	Contrastive Learning Enhanced	CDP	Cisco Discovery Protocol
TFE	Temporal Fusion Encoder	LLDP	Link Layer Discovery Protocol
MTIG	Malicious Traffic Interaction Graph	MQTT	Message Queuing Telemetry Transport
CIC	Canadian Institute for Cybersecurity	NTP	Network Time Protocol
ME	Machine learning and Evidence verification	CTU	Czech Technical University
STP	Spanning Tree Protocol	FTP	File Transfer Protocol
SMTP	Simple Mail Transfer Protocol		

Vita

Adit Sharma

Bachelor of Technology - Computers and Communications Engineering Manipal University Jaipur, Rajasthan, India 2018-2022

Publications:
Adit Sharma, Arash Habibi Lashkari **" A survey on encrypted network traffic: A comprehensive survey of identification/classification techniques, challenges, and future directions"**, published in Computer Networks (SJR: Q1), 110984 - Volume 257, February 2025, 110984

Adit Sharma, Arash Habibi Lashkari **"Hybrid Attention-Enhanced Explainable Model for Encrypted Traffic Detection and Classification"** submitted to International Journal of Information Security, Springer Submission Id: e97abeac-7a88-44b6-ae18-e8a49537fb9a