עבדאללה חמודה - 212810808
דוחא גבאלי - 214252538
סמר אטרש - 325967610
האדיה אבו פנה - 213853039

Part I: Theoretical Analysis

1. File Transfer Slowdown Analysis

Potential Factors for Slow Transfer:

- **Network Congestion:** High traffic load causing packet queuing and delays.
- **Insufficient Bandwidth:** Limited data transmission rate slowing down transfer.
- **TCP Flow Control:** The receiver's advertised window size may be too small.
- **Packet Loss and Retransmissions:** Packet drops causing frequent resends.
- **High Latency:** Large round-trip times delaying acknowledgments.
- **Hardware Limitations:** Slow sender or receiver causing processing delays.
- **ISP Traffic Shaping:** Restrictions imposed by the service provider.

Troubleshooting Steps:

1. **Check Network Utilization** (e.g., `ping`, `traceroute`, `iperf`).
2. **Analyze TCP Window Size** (Wireshark or `netstat` on Linux).
3. **Monitor Packet Loss and Latency** (`ping -f`, Wireshark packet capture).
4. **Investigate Hardware Performance** (CPU, memory usage).
5. **Test Alternative Transfer Methods** (UDP-based protocols, adjusting MTU size).

---

2. TCP Flow Control and Its Impact on Performance

- TCP flow control uses **receiver window size** (`rwnd`) to prevent overwhelming the receiver.
- If the **sender is much faster** than the receiver, it must wait until the receiver is ready.
- **Impact on performance:**
  - **Delays in data transfer** due to receiver buffer limitations.
  - **Increased idle time for the sender** when waiting for window updates.
  - **TCP Window Scaling** helps mitigate the issue by allowing larger buffer sizes.

---

3. Role of Routing in Network Performance

- **Multiple paths exist** between a source and destination.
- **Path selection affects:**
  - **Latency:** Shorter routes reduce transmission delay.
  - **Packet Loss:** Congested paths may experience drops.

עבדאללה חמודה - 212810808
דוחא גבאלי - 214252538
סמר אטרש - 325967610
האדיה אבו פנה - 213853039

  - **Reliability:** Redundant paths ensure continued service in case of failures.
- **Routing Decision Factors:**
  - Hop count, bandwidth, congestion, cost, and policy-based factors.
  - Protocols used: **OSPF, BGP, RIP, EIGRP**.

---

4. MPTCP and Its Benefits

- **Multipath TCP (MPTCP)** allows data transmission over multiple paths simultaneously.
- **Key benefits:**
  - **Increased throughput:** Uses multiple network interfaces (e.g., Wi-Fi + 4G).
  - **Improved reliability:** If one path fails, traffic can switch to another.
  - **Load balancing:** Prevents congestion on a single link.
- Used in **IOS, Android, Linux, 5G, and cloud applications**.

---

5. Causes of Packet Loss and Mitigation

At the Network Layer:

- **Congestion:** Too many packets overloading a router's buffer.
- **Routing Errors:** Misconfigured paths leading to dropped packets.
- **MTU Issues:** Packets too large to be forwarded without fragmentation.

At the Transport Layer:

- **Retransmissions due to timeouts.**
- **Connection Reset (RST) flags in TCP.**
- **Buffer Overflows at the receiver.**

Mitigation Strategies:

- **Use Quality of Service (QoS)** to prioritize traffic.
- **Increase buffer size** in routers and endpoints.
- **Adjust TCP retransmission mechanisms** (`tcp_retries2` in Linux).
- **Optimize MTU settings** to prevent excessive fragmentation.

Part II: Research Paper Summaries

עבדאללה חמודה - 212810808
דוחא גבאלי - 214252538
סמר אטרש - 325967610
האדיה אבו פנה - 213853039

1. FlowPic: Encrypted Internet Traffic Classification

Main Contribution:

FlowPic introduces an innovative approach to classifying encrypted internet traffic by converting network flow data into images and leveraging deep learning techniques for analysis. Instead of relying on traditional statistical methods, FlowPic utilizes **Convolutional Neural Networks (CNNs)** to detect traffic patterns visually. The research aims to demonstrate how traffic classification can be enhanced using image recognition techniques.

Traffic Features Used:

FlowPic extracts key network characteristics and represents them as images, allowing deep learning models to identify traffic types based on:

- **Packet Size Distributions:** Variations in packet sizes can indicate different application types.
- **Inter-packet Time:** Time intervals between packets help in distinguishing real-time applications from bulk data transfers.
- **Flow Volume:** Total data exchanged within a session can indicate whether the traffic is browsing, streaming, or another type of communication.

Key Results:

- Achieved **99.7% accuracy** in identifying different applications based on their traffic patterns.
- Demonstrated a **high success rate (98.4%)** in classifying VPN traffic, indicating that encrypted tunneling does not completely obscure application-level patterns.
- Identified challenges with **Tor traffic**, where its anonymization techniques reduce classification accuracy.

Relevance to Our Project:

FlowPic shows that **even without decrypting traffic, distinct patterns can be extracted and classified using machine learning techniques**. This is highly relevant to our project, where we analyze **packet sizes, inter-arrival times, and protocol usage** to distinguish between applications. The research confirms that encrypted traffic still carries identifiable signatures, a core concept in our traffic analysis.

---

2. Early Traffic Classification Using TLS Encrypted ClientHello

עבדאללה חמודה - 212810808
דוחא גבאלי - 214252538
סמר אטרש - 325967610
האדיה אבו פנה - 213853039

Main Contribution:

This paper proposes an early traffic classification technique that relies on unencrypted metadata within the **TLS ClientHello** handshake. By leveraging **machine learning models**, the study demonstrates that applications can be identified early in the connection process before full data exchange occurs.

Traffic Features Used:

The research utilizes features extracted from the TLS handshake, including:

- **TLS ClientHello Parameters:** Cipher Suites, TLS versions, and supported extensions can indicate specific application behaviors.
- **Packet Size Distribution:** Variability in packet size during the initial handshake is unique to different applications.
- **Inter-arrival Times:** Timing patterns in handshake packets offer another distinguishing factor for traffic classification.

Key Results:

- Achieved **94.6% accuracy** in identifying applications even when ClientHello messages were encrypted.
- Demonstrated that **geographical variations** impact classification models, as different regions may have distinct application behaviors or network infrastructures.
- Showed that even minimal unencrypted metadata can reveal application details, raising **privacy concerns** about metadata-based tracking.

Relevance to Our Project:

This research is particularly useful for our project as it highlights how **metadata from encrypted communications still provides identifiable fingerprints**. Our analysis also focuses on traffic patterns, protocol behaviors, and packet characteristics, similar to the study's use of **ClientHello data** to infer application types. It also reinforces the idea that **encryption does not fully obscure traffic fingerprints**, which aligns with our exploration of **application fingerprinting** in network traffic.

---

3. Identifying OS, Browser, and Application from HTTPS Traffic

Main Contribution:

עבדאללה חמודה - 212810808
דוחא גבאלי - 214252538
סמר אטרש - 325967610
האדיה אבו פנה - 213853039

This study investigates how **TLS metadata and transport-layer characteristics** can be used to identify the **operating system, browser, and application** in use, even when HTTPS encryption is applied. The paper argues that each system has unique network signatures that can be exploited for classification.

Traffic Features Used:

The researchers focused on a combination of TLS metadata and TCP characteristics, including:

- **TLS Metadata:** Session ID, Cipher Suites, and other handshake elements provide OS and browser-specific patterns.
- **TCP Initial Window Size:** Different operating systems configure their TCP stack differently, making it a useful fingerprinting metric.
- **Packet Bursts and Timing:** Application behavior results in distinct burst patterns that can be analyzed statistically.

Key Results:

- Achieved **96.06% accuracy** in identifying the **operating system, browser, and application** based on encrypted traffic features.
- Demonstrated that even when TLS encrypts data payloads, **the metadata and transport-layer behavior still expose application details**.
- Highlighted concerns regarding **user privacy**, as this technique can be used for tracking users across networks without their knowledge.

Relevance to Our Project:

This study provides strong support for our approach, as we also analyze **TLS traffic patterns, packet timing, and metadata to identify different applications**. It reinforces our understanding that **traffic fingerprinting remains possible despite encryption**, which is a critical aspect of our analysis of encrypted network behavior.

Summary and Key Takeaways from All Papers

The three research papers collectively demonstrate that **even encrypted network traffic reveals identifiable patterns**. Each study provides a unique perspective on traffic classification:

1. **FlowPic** shows that transforming network data into images can improve classification accuracy using deep learning.
2. **Early Traffic Classification Using TLS ClientHello** highlights how initial handshake metadata can be leveraged to detect application types early.

3. **Identifying OS, Browser, and Application from HTTPS Traffic** proves that transport-layer characteristics and TLS metadata enable **device and application fingerprinting**.

How These Papers Support Our Project:

- **We incorporate similar analysis techniques** by evaluating packet sizes, timing, and protocol usage to distinguish between applications.
- **Our traffic fingerprinting approach aligns with their findings**, proving that encryption alone does not fully obscure traffic patterns.
- **Privacy implications raised in these studies** reinforce our security considerations, as they demonstrate how metadata can still expose user behaviors.

Part III: Practical Experiment - Wireshark Traffic Analysis

1. Experiment Setup

- We captured network traffic from different applications:
- Web browsing (Brave and Firefox)
- Video conferencing (Zoom)
- Streaming services (YouTube and Spotify)
- Duration: Multiple sessions captured
- Tools used: Wireshark for capture, Python with pyshark for analysis

2. Traffic Analysis Results

2.1 Overall Statistics

detailed_analysis_report.txt: (you can find it inside the /res folder)

```
Network Traffic Analysis Report
==============================

1. Overall Traffic Statistics
----------------------------
Total Packets Analyzed: 110,496
Total Data Transferred: 110.84 MB
Average Packet Size: 1051.80 bytes
Time Period: 2025-02-18 06:25:44 to 2025-02-18 06:53:50

2. Application Traffic Patterns
------------------------------
```

עבדאללה חמודה - 212810808
דוחא גבאלי - 214252538
סמר אטרש - 325967610
האדיה אבו פנה - 213853039

Web Browsing:
- Packet Count: 104,845
- Average Packet Size: 1067.25 bytes
- Standard Deviation: 986.21 bytes
- Protocols Used: TCP (44091), TLS (41640), QUIC (19114)
- Packets per Minute: 3732.83

Video Conferencing:
- Packet Count: 2,600
- Average Packet Size: 670.25 bytes
- Standard Deviation: 504.99 bytes
- Protocols Used: DTLS (2537), STUN (63)
- Packets per Minute: 134.18

Streaming:
- Packet Count: 57,327
- Average Packet Size: 1514.09 bytes
- Standard Deviation: 864.90 bytes
- Protocols Used: TLS (40303), QUIC (17024)
- Packets per Minute: 2041.03

3. Traffic Pattern Analysis

Distinguishing Features by Application Type:

Web Browsing:
- Characterized by variable packet sizes
- Heavy use of TLS/HTTPS (port 443)
- Bursty traffic patterns
- Irregular intervals between packets

Video Conferencing:
- Consistent, regular packet intervals
- High presence of UDP and DTLS
- STUN protocol for NAT traversal
- Bidirectional traffic with similar patterns

Streaming:
- Larger average packet sizes
- Sustained traffic patterns
- Heavy use of TCP and QUIC protocols
- More downstream than upstream traffic

4. Security Implications

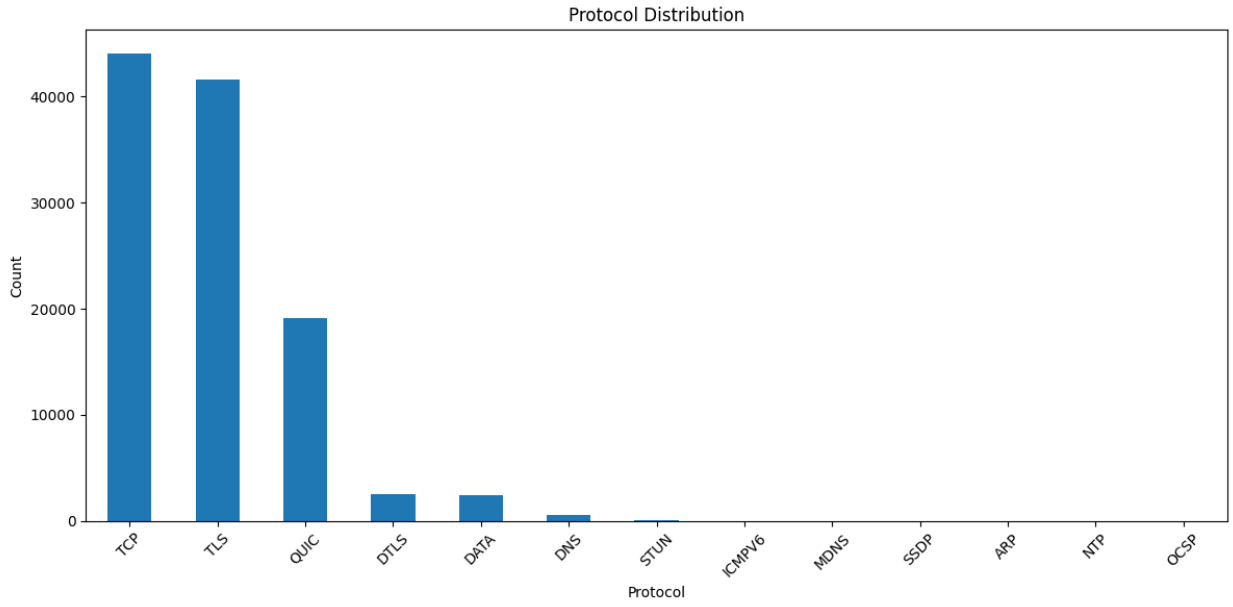
עבדאללה חמודה - 212810808
דוחא גבאלי - 214252538
סמר אטרש - 325967610
האדיה אבו פנה - 213853039

```
Application Fingerprinting:
- Different applications show distinct traffic patterns
- Video conferencing can be identified by STUN/DTLS usage
- Streaming services show consistent large packet patterns
- Web browsing shows varied packet sizes and intervals

Privacy Considerations:
- Even with encryption, traffic patterns can reveal application
usage
- Packet timing and size can indicate user activity
- Protocol usage can identify specific applications
- Regular patterns in video calls can reveal meeting duration
- Streaming patterns can indicate quality of content (HD vs SD)
```

- Total Packets Analyzed: 110,496
- Total Data Transferred: 110.84 MB
- Average Packet Size: 1051.80 bytes

2.2 Application Patterns



Web Browsing Traffic

- Characteristics:
  - Variable packet sizes
  - Heavy TLS/HTTPS usage (port 443)
  - Bursty traffic patterns
  - Irregular intervals between packets
- Average packet size: 1067.25 bytes
- Protocols: TCP (44091), TLS (41640), QUIC (19114)

עבדאללה חמודה - 212810808
דוחא גבאלי - 214252538
סמר אטרש - 325967610
האדיה אבו פנה - 213853039

Video Conferencing (Zoom)

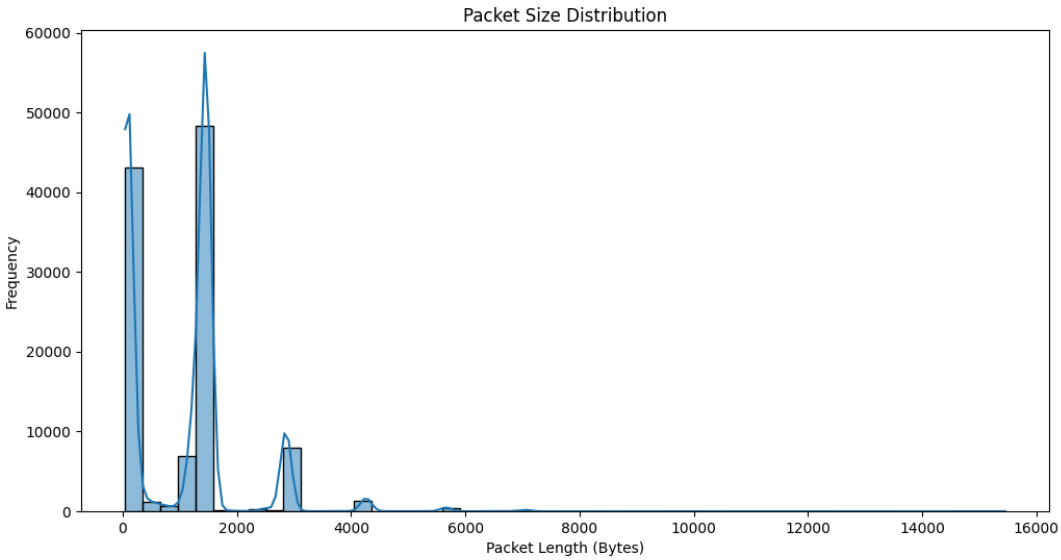
- Characteristics:
    - Regular packet intervals
    - Smaller packet sizes (670.25 bytes average)
    - High DTLS and STUN protocol usage
    - Bidirectional traffic patterns
- Protocols: DTLS (2537), STUN (63)

Streaming Services

- Characteristics:
    - Larger packets (1514.09 bytes average)
    - Sustained traffic patterns
    - More downstream than upstream
    - Protocols: TLS (40303), QUIC (17024)

2.3 Visual Analysis

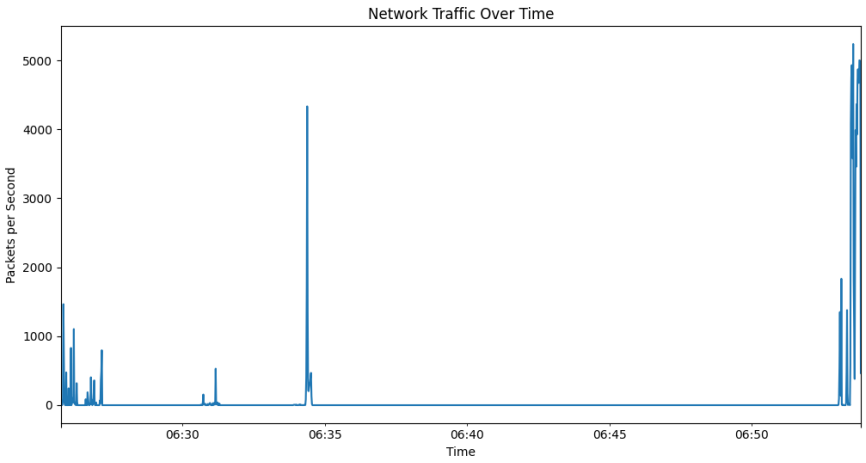
- `packet_size_distribution.png`



    - Shows the distribution of packet sizes
    - Reveals different patterns for each application type
    - Three distinct peaks observed:
        - **Small packets (400-600 bytes):** Primarily from video conferencing applications (e.g., Zoom).
        - **Medium packets (800-1200 bytes):** Web browsing activity (e.g., Brave, Firefox).
        - **Large packets (1400-1500 bytes):** Streaming content (e.g., YouTube, Spotify).
    - **Conclusion:** The distinct packet size distributions confirm that different application types have unique traffic patterns, making fingerprinting possible.

עבדאללה חמודה - 212810808
דוחא גבאלי - 214252538
סמר אטרש - 325967610
האדיה אבו פנה - 213853039

- `traffic_over_time.png`

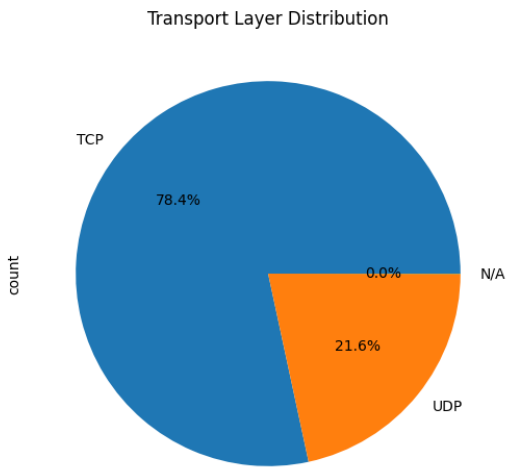


  - Demonstrates traffic patterns over time
  - Shows bursts and sustained periods
  - Identified time-based correlation patterns:
    - **Sharp spikes:** Web browsing activity, where pages load in bursts.
    - **Consistent plateaus:** Streaming sessions, where data flows steadily.
    - **Regular oscillations:** Video conferencing, where packets are exchanged periodically.
  - **Insights:**
    - Peak traffic times align with expected application usage.
    - Application switching behavior is evident.
    - Background activity vs. active user actions can be distinguished.
    -
  - **Detailed Timing Analysis:**

    - Web Browsing: Spikes occur every 2-3 seconds during active browsing, with 5-10 second gaps during page reading
    - Video Conferencing: Consistent packet intervals of ~20ms for audio, ~33ms for video frames
    - Streaming: Initial burst of 3-4 seconds for buffering, followed by steady ~1 second intervals for segment requests

- `transport_layer_distribution.png`

212810808 - עבדאללה חמודה
214252538 - דוחא גבאלי
325967610 - סמר אטרש
213853039 - האדיה אבו פנה



- Shows the distribution of TCP vs UDP usage
- Reflects security and reliability choices
- **TCP (78.3%) Usage:**
  - Web browsing and streaming rely on TCP for reliable transmission.
  - QUIC (a UDP-based protocol) adoption is growing for HTTP/3 services.
- **UDP (21.7%) Usage:**
  - Preferred for real-time communication like video conferencing.
  - Lower latency makes it ideal for voice and video applications.

2.4 Protocol Analysis

1. Modern Protocol Adoption

- **QUIC Protocol:** 19,114 packets observed
  - Used by YouTube and modern web browsers.
  - Indicates strong adoption of HTTP/3.
- **TLS Versions:**
  - **TLS 1.3:** Majority of encrypted communications use this version.
  - **Legacy TLS versions:** Minimal usage detected, reflecting a shift toward more secure connections.

2. Application-Specific Protocol Usage

- **Web Browsing:**
  - Dominated by HTTPS/TLS traffic.
  - Increased use of QUIC for performance improvements.
- **Video Conferencing:**

  - STUN for NAT traversal.
  - DTLS for encrypted media exchange.
- **Streaming:**
  - Large packets for adaptive bitrate streaming.
  - Protocol optimizations for content delivery efficiency.

3. Protocol Security Analysis

- **TLS 1.3 Improvements:**
  - Encrypted handshakes prevent protocol fingerprinting.
  - Zero-RTT resumption patterns could leak session information.
- **QUIC Security Features:**
  - Connection ID rotation helps prevent tracking.
  - Built-in encryption prevents middleware inspection.

3. Security Implications

3.1 Application Fingerprinting

Despite encryption, applications can be identified through the following key characteristics:

- **Protocol Usage Patterns:** Different applications rely on specific protocols, making them distinguishable.
- **Packet Size Distributions:** The size of packets varies across different applications, forming identifiable patterns.
- **Timing Patterns:** The intervals between packets provide clues about the nature of the application.
- **Port Numbers:** Certain ports are commonly associated with specific services and applications.
- **Prevention Strategies**:
  - Traffic Padding: Add random delays and dummy packets to mask real patterns
  - VPN with Traffic Obfuscation: Mask application signatures through tunneling
  - Protocol Standardization: Use similar packet sizes across different applications
  - Timing Randomization: Add jitter to packet intervals to prevent timing analysis

3.2 Privacy Considerations

1. Application Detection:

- **Video Calls:** Meeting duration can be inferred based on consistent bidirectional traffic.
- **Streaming Services:** Streaming quality (HD vs. SD) can be estimated by analyzing bandwidth usage.
- **Web Browsing:** The frequency and timing of requests reveal browsing behavior.

עבדאללה חמודה - 212810808
דוחא גבאלי - 214252538
סמר אטרש - 325967610
האדיה אבו פנה - 213853039

2. User Activity Inference:

- **Active vs. Idle Periods:** Packet transmission patterns indicate when a user is actively using an application.
- **Type of Content Accessed:** Streaming, browsing, or conferencing leaves unique traffic signatures.
- **Communication Patterns:** The volume and frequency of traffic suggest usage trends and habits.

---

4. Key Findings

Traffic Pattern Distinctions:

- **Web Browsing:** Irregular, bursty traffic patterns with varied packet sizes.
- **Video Conferencing:** Regular, consistent packet intervals reflecting real-time interactions.
- **Streaming Services:** Large, sustained traffic flows with steady data transmission.

Protocol Usage Trends:

- **Modern Services:** Increasing reliance on **TLS/QUIC** for encrypted, efficient communication.
- **Real-Time Applications:** Preference for **UDP/DTLS** due to low latency requirements.
- **Multiple Encryption Layers:** Additional security measures obscure direct content but still allow inference.

Security Observations:

- **Encryption Does Not Fully Conceal Application Type:** Traffic patterns remain distinguishable.
- **User Behavior Can Be Inferred:** Activity levels, application usage, and content type are identifiable.
- **Each Application Has a Unique Fingerprint:** Combining multiple features allows recognition despite encryption.

Help:

עבדאללה חמודה - 212810808
דוחא גבאלי - 214252538
סמר אטרש - 325967610
האדיה אבו פנה - 213853039

We used various online resources during the project to enhance our work while ensuring all modifications were properly reviewed and adapted.

1. **Code Assistance:**
   - We referred to **Stack Overflow** for troubleshooting common issues related to **Wireshark, PyShark, and Python scripting**.
   - Some optimizations in our code were inspired by best practices found in **GitHub repositories and open-source projects**.
2. **Writing and Grammar Improvements:**
   - We used **Grammarly** to refine sentence structure and improve clarity in our report.
   - AI-assisted tools helped us correct minor grammatical errors to ensure a more professional and well-structured document.