

# **רשותות תקשורת - פרויקט גמר**

**קבצי הקלטות נמצאים כאן:**

[link](https://drive.google.com/drive/folders/1mz_QXR5Qlmiu4UvDlwEPCSz7vvq817_4?usp=drive_link)

תאריך הגשתה: 05.03.2025

## **תוכן עניינים:**

חלק:	עמודים:
חלק 1	<a href="#"><u>3-9</u></a>
חלק 2	<a href="#"><u>10-22</u></a>
חלק 3:	<a href="#"><u>23-45</u></a>
בONO:	<a href="#"><u>46</u></a>
נוף 1:	<a href="#"><u>47-48</u></a>

## חלק 1: מענה על שאלות

.1.  **משתמש מדויק שהעברת הקבצים שלו איטית, ועליך לנתח את שכבת התעבורה כדי לזהות את הסיבות האפשריות. אילו גורמים עשויים לגרום להעברה האיטית, וכייזה הייתה פוטר את הבעיה?**

**בעיה:** שימוש בפרוטוקול UDP להעברת חבילות עלול לגרום לאובדן נתונים, שכן UDP אינו כולל מגנון שידור חוזר כמו TCP ואינו מספק בקרת זרימה לשילטה על קצב השידור. כתוצאה לכך, אם חבילות אובדות או מגיעות בסדר שגוי, אין תיקון אוטומטי, מה שעולול להוביל להאטה או לתקלות בהעברת הנתונים.

**פתרון:** יש לבדוק האם ניתן להמיר את ההעברה ל-TCP, שմבטייח אמינות באמצעות מנוגנוני אישור (ACKs) ושידור חוזר במקרה של אובדן חבילות. TCP גם כולל בקרת זרימה שמנועת הצפת הרשות ושומרת על יציבות הביצועים. לחילופין, אם השימוש ב-UDP נדרש לצורך מהירות והשהייה נמוכה, ניתן לעבור ל-QUIC, שמבוסס על UDP אך מכיל תיקון שגיאות ובקרת זרימה, ובכך משלב ביצועים גבוהים עם אמינות טובה יותר.

.1.2.  **בעיה:** כאשר TCP מזהה שאיבד חבילות (Packet Loss), הוא משדר אותן מחדש, מה שעולול להאט את קצב העברה. ככל שאובדן החבילות גבוהה יותר, כך יהיה יותר שידורים חוזרים, מה שוביל לעיכובים נוספים ופגע ביעילות העברה.

**פתרון:** אחת הדרכים לצמצם איבוד חבילות היא להשתמש בחיבור קווי (Ethernet) במקום Wi-Fi, שכן חיבור קווי יציב יותר, מספק זמן השהייה (Latency) נמוך יותר, מפחית משמעותית את הסיכוי לאובדן חבילות ואניון מושפע מרפרעות חיצונית כמו אותות אלחוטיים מתנגשים בנוסף, ניתן להגדיל את גודל חלון TCP (Window Scaling) כדי לשפר את קצב העברה. מגנון זה מאפשר לשולח לשלוח יותר נתונים לפני המתנה לאישור (ACK), ובכך מפחית את כמות ההפסקות בשידור ומיעל את ניצול רוחב הפס.

.1.3.  **בעיה:** שליחת הודעות רבות: שליחת הודעות רבות ללא שליטה, מה שיכול להוביל לאיבוד חבילות ובזבוז זמן עיבוד.

**פתרון:** נגביל את קצב השילחה בפרוטוקול TCP או באפליקציה עצמה, מה שמנוע עמוס ומגביל את קצב העברה, כל פעם שנגיע למכסת ההודעות שניתנת לשולח (לדוגמא בשנייה), פרוטוקול TCP ימתין עד לקבלת כל ACKs ולאחר מכן ניתן להמשיך.

#### 1.4.

**בעיה:** גודל חלון TCP : בTCP ישנו שימוש ב Sliding Window בצד שבעת איבוד נתונים לא ישלחו יותר מדי נתונים מל' לקלט ACK. אך , אם נגידיר את גודל החלון להיות קטן מדי נצטרך לחכות לעתים תכופות מדי לאישור מה שייצור העברה איטית גם אם אכן יש רווח פס גבוה.

**פתרון:** קודם כל יש לזהות את מקור הבעיה על ידי ניתוח החבילות ב Wireshark , לאחר שהבנו את מקור הבעיה נוכל להגדיל את חלון ההעברה מהצד של השולח ואם אפשרי גם המקלט בצד שנוכל לשЛОח יותר נתונים טרם קבלת האישור.

#### 1.5.

**בעיה:** עיכוב בהגעת הקבצים עקב זמן תגובה גבוהה מצד השירות (RTT) - במידה והRTT יהיה ארוך , המשמש יצרך לחכות הרבה זמן בכל פעם שיקבל ACK מהשירות , זה יגרום לקצב השידור להיות איטי יותר.

**פתרון:** נבדוק את נתיב הרשת בעזרת traceroute או ping כדי לנתח את זמן התגובה של שרת ואופן ניתוב המידע בראשת . אם מקבל זמן תגובה גבוהה, נוכל לנתק מחדש את נתיב הרשת כך שהמידע תעבור דרך שרת חלופי ומהיר יותר. זו יכולה להיות גם בעיה גיאוגרפית, لكن יהיה علينا לעכור לשרת קרוב יותר. בנוסף נוכל לשלב שימוש בפרוקס' עם cache (למנוע בקשות חוזרות).

- .2. נתח את ההשפעות של מגנון בקרת הזרימה של TCP על העברת נתונים. כיצד הדבר ישפיע על הביצועים כאשר למקור יש כוח עיבוד גבוה משמעותית מזה של המקלט?
- 2.1. בקרת הזרימה (Flow Control) בפרוטוקול TCP מיועדת למנוע מהשולח להציף את המקלט בכמות נתונים שהוא אינו מסוגל לעמוד בזמן אמת, ע"י שימוש במנגנונים הבאים:  
 .Sliding window, Receiving window, ACK send/receive
- הבעיה:** כאשר המקור בעל יכולת גבוהה מיכולת העיבוד של היעד, יוצר חוסר איזון המוביל למספר דברים:  
 כאשר המקור בעל כוח עיבוד גבוה מכך העיבוד של היעד, הוא ישלח הודעות בקצב גבוה, מה שימלא את קיבולת הbuffern receiving window בצד המקובל.  
 כאשר buffern מתמלא, גודל הwindow יקטן עד כדי "0", מה שמוביל לכך שהשולח ימתין עד לפינוי buffern, ויגרום להאטה בקצב העברת הנתונים.  
 הצד מקבל, שלא מצליח לעמוד בהירות את הנתונים המתקבלים, מתקשה לשЛОח ACKs בתדרות גבוהה.  
 במקרים מסוימים, אם העיכוב נמשך, ה-Window Sliding אף עלול להצטמצם, מה שמאגביר עוד יותר את קצב העברה.  
 אם השולח אינו מקבל ACK בזמן, הוא עלול לפרש זאת כהודה שאבדה או כעומס בראשת ובוצע שליחה חוזרת (Retransmission), וכתוצאה לכך, כמות הנתונים המתתקבלת אצל מקבל עולה עוד יותר, מה שמכביד על הבادر ומחמיר את הבעיה.

.3. בחירת המסלול משפיעה על ביצועי הרשת בכמה אופנים , נסביר את העיקריים שבהם ונגידיר

גורמים אוטם יש לנקח בחשבון בעת קבלת החלטות נি�וטוב:

.3.1. **עומס ברשת - congestion:**

אם ישנו מספר גדול של חברים משתמשים באותו מסלול , יכולם להיגרם עומסים, בעקבות עומסים אלו יכולם להיווצר איבודים של חבילות ובעקבותיהם גם שידורים חוזרים שייאטו עוד יותר את החיבור בעקבות בחירת מסלול זה.

.3.2. **אורך המסלול , כמות הנתבים בדרך זמן השהיה - latency:**

נגידיר זמן השהיה - הזמן שעובר מהרגע שבו חבילה נשלחת ממקורה ועד לרגע שבו היא מגיעה ליעדה. כאשר משתמש במסלול ארוך יותר עם הרבה hops (נתבים) בדרך החבילות יגיעו באיחור מה שיפגע בביצועים של הרשת. אך , כאשר משתמש במסלול עם כמה שפותה נתבים ונוכל לבצע קישורים מהירים זמן השהיה יהיה נמוך יותר , لكن נשתדל לבחור עוד שיקולים בהמשך)

.3.3. **רוחב פס זמן - bandwidth:**

רוחב הפס גם הוא משפיע על זמן העברה, כל מסלול עשוי לכלול קישורים עם רוחב פס שונה , לדוגמה, אם מסלול מסוים עובר דרך נתבים בעלי חיבור איטי או מהירות איטית מה שעלול לגרום בדרך הוא מעין "צואר בקבוק" שיגרום להאטת תהליך העברת החבילות.

.3.4. **אמינות - reliability:**

לכל מסלול יש עמידות שונה לכשל. במידה נתב או קישור נזילם, המסלול צריך להיות מוחלף במסלול גיבוי ב מהירות האפשרית. פרוטוקולים כמו OSPF ו-BGP יכולים להבטיח החלפת מסלול אוטומטית במקרה של כשל.

#### **4. כיצד פרוטוקול MPTCP משפר את ביצועי הרשת?**

TCP Multipath מאפשר העברת נתונים דרך מספר מסלולים במקביל, במקום העברתם במסלול יחיד. ע"י חלוקת הנתונים דרך *flows-subs* (תת חיבורים), מונצל רוחב הפס הזמן באופן מיטבי.

##### **4.1. הגדלת קצב העברה**

בעת שימוש ב-TCP, כל החבילות עוברות דרך נתיב אחד בלבד, כך שמהירות ההעברה מוגבלת לרוחב הפס של אותו נתיב.

MPTCP מאפשר שימוש במספר חיבורים במקביל כך שההעברה מתחולקת בין כמה ערכאים באותו הזמן, ובכך גם מעלה את קצב ההעברה הכלול.

לדוגמא: ע"י חיבור ב WiFi ורשת סלולרית (4G או 5G) במקביל, או חיבורים דרך מספר נתבים או שרתיים במקביל.

כתוצאה לכך ישנו יותר רוחב פס זמן, הנתונים יכולים לעבור דרך מספר מסלולים בו זמנית ובכך הנתונים שייעברו ביחד ביחידת זמן מסוימת יהיו גדולים יותר MCP.

##### **4.2. שיפור אמינות וחוסן הרשת**

בניגוד לTCP שבו קיים רק מסלול אחד להעברת הנתונים, שבמקרה ומשהו נכשל באותו המסלול נדרש להפסיק את ההעברה עד למציאת מסלול חדש, ב TCP MPTCP ההעברה מבוצעת ע"י מספר מסלולים באותו הזמן, כך שגם אם יש בעיה באחת הנתיבים או שההעברה בהם נכשלה, ההעברה תמשיך להתבצע בשאר הנתיבים ללא עיכוב או הפסקה. בכך בעת כשל ברשות לא תהיה השפעה משמעותית כמו בTCP על ביצועי הרשת והמידע יוכל להמשיך לעבור.

##### **4.3. זמן שהוא קטן יותר**

בעת שימוש ב-MPTCP תבוצע בחירה של המסלול האופטימלי באופן דינמי, כאשר ישנה התחשבות בזמן השהייה, בעומסם, וביכולת של ה *hop* הנוכחי והחיבור של המשתמש. אם אחד הנתיבים עמוס או איטי, התעבורה מנוטבת לנכדים מהירים יותר בזמן אמיתי.

.5. אתה מנטר תעבורת רשת ושם לב לאובדן חבילות גבוהה בין שני נתבים. נתח את הסיבות

האפשריות לאובדן חבילות בשכבות הרשת והתעבורה, והצע צעדים לפתרון הבעיה.

### 5.1. סיבות לאובדן חבילות בשכבות הרשת:

#### 5.1.1. עומס ברשת:

כאשר אחד הנתבים נמצא בעומס תעבורה, הוא מתחילה להיפטר מחבילות קיימות עקב מהסור בזיכרון של buffer, בעיה זו מתרחשת כאשר קיבלת הרשת נמוכה או ברשותות שלא מקיימות ניתוב אופטימלי.

פתרון: נזהה את עומס הרשת ומהין הם נגרמים, ניתן עדיפות לתעבורה הקritisית ונשתמש בפרוטוקולים המספרים עומסים כמו: load balancing, הבודק האם יש מספר נתבים אפשריים מאותו מקור לאותו יעד אז יבדוק אם ניתן פזר את העומס כדי למנוע ריכוז חבילות במסלול ייחודי לא אופטימלי. בנוסף אם העומס גבוה ניתן לשדרג את הנתב או את רוחב הפוף.

#### 5.1.2. בעיות ניתוב (Routing Issues):

יש כמה גורמים לבעיית הניתוב:

##### 5.1.2.1. כתובת יעד שגוי

פתרון: לוודא שהכתובת תקינה לפני שאנחנו מתקשרים אליו או לעדכן דרך נתיב אופטימלי או באמצעות ping או traceroute.

##### 5.1.2.2. הבעיה שנכנסים ללולאות ניתוב(Routing Loop) זהה קורה שהחbillה

עוברת בין שני נתבים ללא כתובת יעד.

פתרון: להשתמש במנגנים למניעת הבעיה הזאת כמו OSPF

##### 5.1.2.3. נתיב לא עדכני(Stale Route) אשר יש קשר עדין קיימ בטהלה אבל

הקשר כבר נתק זה גורם לשילוח חבילות לנטייב שאינו פעיל

פתרון: להשתמש ב프וטוקול ניתוב דינמי כמו : OSPF , BGP שיעדכן את הטבלה באופן אוטומטי או בהגדרת timeout עבור נתיבים ישנים הקיימים בטהלה.

#### 5.1.3. TTL נ苟ך (Time to Live Expired)

היא תימחק לפני שתגיע ליעד.

פתרון: הגדלת ערך TTL של החבילות אם הבעיה מתרחשת בתחנות קצה (כמו מחשבים או שרתים), יש לוודא שה-TTL של החבילות היוצאות גבוהה מספיק כדי להגיע ליעד.

בדוק כמה קפיצות (Hops) יש בין המקור ליעד. אם הרשת כוללת מספר רב של נתבים, יש להגדיל את ה-TTL בהתאם.

5.1.4. שרת DNS לא מעודכן (בעיות בטבלה הניתוב) – אם טבלאות הניתוב אינן מעודכנות, הנתבים עלולים לנתק חבילות לכתובות שגויות או להשתמש במסלולים עמוסים ולא אופטימליים.

פתרון: עדכן טבלה DNS ע"י בדיקת הניתוב כדי לראות אם הכתובת אכן נגישה, ניקוי המטען לאחר תקופה מסוימת של זמן או שימוש בשרת DNS אמין כמו google cloudflare.

## 5.2 סיבות לאובדן חבילות בשכבת התעבורה:

### 5.2.1 נתק חיבור TCP - אם חיבור ה-TCP מתנתק עקב Timeout או עומס יתר, הנטב

השולח עשוי שלא לקבל אישור קבלה בזמן ולהפעיל מגנון ניתוק. במקרה זה, החיבור יכול ייסגר, וכל הנתונים שלא הועברו בהצלחה יאבדו, מה שעלול לגרום להאטה וביצועים ירודים ברשת.

**פתרון:** יש להגדיל את זמן ה-timeout בפרמטרים של tcp כדי לאפשר התואששות טובה יותר מעומס זמן ולהשתמש במנגנוןים כמו `keep-alive` ו-`fast retransmit` כדי למנוע ניתוקים לא רצויים. בנוסף, ניתן לישם QoS כדי לתת עדיפות לחיבורים חשובים ולהפחית עומס על הרשת.

### 5.2.2 שימוש ב프וטוקול UDP - UDP אינו מספק מגנון אישור קבלה, ולכן חבילות

עלולות לאלת לאיבוד מבלי שהשולח ידע זאת.

**פתרון:** שימוש ב프וטוקול QUIC אשר מספק מגנון ACK כמו לTCP.

### 5.2.3 פרוגמנטציה ו恢復 חבילות - אם חבילת TCP גודלה מגודל MSS של המקלט,

היא עשויה להידחות, מה שייגרם לאובדן נתונים. בנוסף, אם חבילת אחת מתוך מקטעים נאבדת, TCP ישלח מחדש את כל הזרם, מה שMOVIL לעומס והשיות ברשת.

**פתרון:** יש להגדיר MSS מתאים בהתאם ל-MTU של הרשת ולהפעיל Path MTU Discovery למניעת פרוגמנטציה. בנוסף, שימוש ב-selective acknowledgment (SA) מאפשר ל-TCP לשלח מחדש רק את החבילות שאבדו במקום מחדש את כל הזרם.

### 5.2.4 חלון קבלה קטן מדי – אם הנטב מקבל איינו מעבד נתונים מספיק מהר, הוא עשוי

לשנות עדכון לחילון הקבלה עם גודל קטן מאוד או אפילו אפס. מצב זה יכול לגרום לנטב השולח לעצור זמנית את שליחת הנתונים, ובמקרים של עיכובים או שגיאות ברשת, חבילות עלולות להישטט או להתיישן לפני שהן נשלחות מחדש.

**פתרון:** הגדלת גודל חילון הקבלה באופן דינמי ושימוש במנגנון window scaling מאפשר ניצול טוב יותר של רוחב הפס ומנעים השהיות ואובדן חבילות.

## חלק 2: קריית שלושה מאמרים ומענה על השאלות:

יש לענות על השאלות הבאות עבורה כל מאמר :

**מהי התרומה המרכזית של המאמר?**

- איזה חידוש או ערך עיקרי הוא מציג?

**בailo מאפייני תעבורת המאמר משתמש, ואילו מהם הם חדשניים?**

- האם יש תכונות שלא הופיעו במאמרים קודמים?

**מהן התוצאות המרכזיות (ניתן להעתיק את התרשימים מהמאמר), ומה המסקנות העיקריות מהתוצאות הללו?**

- אילו תובנות עולות מהתוצאות, ואיך הן משפרות את ההבנה בתחום?

## **מאמר מספר 1 :**

"FlowPic: Encrypted Internet Traffic Classification is as Easy as Image Recognition"

### **מהי התרומה המרכזית של המאמר?**

המאמר בראשיתו דן בסוגיית החיפוש אחר פתרון של ניתוח נכון ומדויק של תעבורה מוצפנת ברשת. הוא מציג מספר שיטות המשמשות כיום, וסביר מדויק כל אחת מהן אינה רלוונטיית עוד עקב השתכללות ההצפנה ההולכת ומשתפרת של תעבורה מידע (במאמר מדובר בעיקר עלי VPN ו-Tor).

המאמר מציע פתרון של גישה ייעילה (מאוד) וחדשנית הנקראת FlowPics, המבוססת על המרת זרם נתונים לתמונה ולאחר מכן שימוש בטכניקות של עיבוד תמונה Tor שילוב עם למידה عمוקה CNN (בעברית - רשת נירונים קונבנציונלית) על מנת לסוג את התעבורה.

כותבי המאמר ביצעו מחקר על מנת לבדוק כמה ייעלה שיטת ניתוח התעבורה באמצעות גישת ה FlowPics, הם מתארים כיצד ניתן ליצור גוף ויזואלי שמייצג את גודל וכמות החבילות שהגיעו כתלות בזמן שבו אנו מסתכלים על תעבורת המידע באופן שיתן לנו להפוך אותו למשתמונה ומדגים כמה גדול ההבדל בין תמונה של תעבורה מוצפנת לubyteורה רגילה, וכמה שונות יכולות היראות תעבורות של אפליקציות שונות, מסקנת החוקרים היא ששיטות סיוג המבוססות על מאפיינים ידניים (כמו סטטיסטיות בסיסיות) לא תמיד מצליחות לכוד את כל הדפוסים המורכבים הללו. אך, הם עוברים לגישה של המרת כל הזרימה לתמונה (FlowPic) ושימוש ב CNN ש"ילמד" את הדפוסים הללו אוטומטית.

המאמר תורם בתחום הסיוג של תעבורה אינטרנט בכך שהוא מציג פתרון ייעיל וחדשי לאתגרים הקיימים בסיווג תעבורה מוצפנת. הגישה המוצעת יכולה לשמש כבסיס לפיתוחים עתידיים בתחום ניהול רשתות ובטחת מידע.

להלן כמה יתרונות עיקריים של הגישה המוצגים במאמר:

1. דיק גבואה בסיווג תעבורה מוצפנת: השיטה מצליחה לסוג בדיק גבואה תעבורה מוצפנת, כולל תעבורה שעוברת דרך VPN ו-Tor, שהיא מאיגרת מאוד לסיוג בשיטות מסורתיות.
2. זיהוי יישומים חדשים: הגישה מאפשרת זיהוי יישומים חדשים שלא היו חלק משלב האימון, מה שמראה על יכולת הכללה טובה.
3. שמירה על פרטיות: השיטה לא מסתמכת על תוכן המטען של המנות, ולכן לא פוגעת בפרטיות של המשתמשים.
4. יעילות חישובית: הגישה מאפשרת לעבוד עם חלון זמן קצר של זרימה חד-כיוונית, מה שמקטין את הצורך בחישובים מורכבים ומספר את יעילות השיטה.

**מאפייני התעבורה בהם המאמר משתמש ואלו מהם מחדשים ממשו?**

### 1. גודל החבילות (Packet Size)

**מה זה?**

כמות הנתונים (בBITS) שנשלחים בכל מנה בודדת במהלך החיבור.

**דוגמה:**

בעת שידור וידאו, המנות יהיו גדולות יחסית ויכילו הרבה נתונים. בשונה מזה, הودעת צ'אט פשוטה תוכל מנות קטנות בהרבה.

**שימוש:**

המאפיין הזה נפוץ בניתוח תעבורת רשת ונועד לזהות את סוג השירות (VIDAO, הורדה, גלישה).

**החידוש במאמר:**

השימוש במאפיין זה כחלק מייצוג היסטוגרמָה דו-ממדית המmirה את גודל המנות לתמונה. בציר ה-X של **FlowPic** (היצוג הויזואלי) גודל המנות מוצג בטווח של **0-1500** BITS.

### 2. זמן הגעת המנות (Packet Arrival Time)

**מה זה?**

זמן החולף בין הגעת שתי מנות עוקבות בפרק זמן מסוים.

**דוגמה:**

בחיבור לאישרו שירות של סטרימינג (כמו YouTube) נראה מנות הגיעו בפרק זמן אחדים, בעוד שבתעborת גלישה באינטרנט נראה זמני הגעה משתנים.

**שימוש:**

משמש בניתוח תעborה כדי לזהות דפוסים של שירותים שונים (סטרימינג, הורדה, גלישה).

**החידוש במאמר:**

שילוב זמן הגיעו עם גודל החבילה מייצוג דו-ממדי חדשני. בציר ה-Y של **FlowPic** זמן הגיעו מוצג בדلتא של **0.01** שניות.

### 3. תדירות המנות (Packet Frequency)

**מה זה?**

מספר המנות שנשלחות או מתתקבלות בפרק זמן מסוים (מנות לשניה).

**דוגמה:**

חיבור סטרימינג שלוחים מנות בתדרות גבוהה וקבועה, בעוד שמערכות Gibi שלוחו מנות בקצב איטי יותר.

**שימוש:**

שימוש בעבר לזיהוי התקפות (כגון DDoS) או להבדיל בין סוגי שירותים שונים.

**החידוש במאמר:**

המרת תדרות המנות לתמונה באמצעות יצירת **FlowPic**. בכל פיקסל בתמונה מופיעה עצמת תדרות המנות בפרק זמן מסוים.

## **החדשון העיקרי במאמר:**

### **1. ייצוג ויזואלי (FlowPic)**

**מה זה?**

המאמר מציע שיטה חדשנית הממירת נתוניםTeVורה להיסטוגרמאות דו-ממדיות שמייצגות את דפוסי המנות בתמונה בגווני אפור.

**למה זה חדש?**

מאפשר להפעיל רשותות נירונים קובולוציוניות (CNN), טכנולוגיה מתקדמת מאוד התמונה, לשוווגTeVורה מוצפנת.

### **2. שימוש בחילון זמן קצר**

**מה זה?**

המאמר משתמש בחילון זמן של **60 שניות** בלבד כדי לאסוף ולנתח נתונים.

**למה זה חדש?**

מאפשר ביצוע סיווג מהיר וחסכני במשאים לעומת שיטות אחרות המשתמשות בפרק זמן ארוכים יותר.

### **3. אי-שימוש בתוכן המטען (Payload)**

**מה זה?**

המאמר לא מנתח את תוכן המנות עצמן אלא רק את המאפיינים החיצוניים שלהם.

**למה זה חדש?**

מאפשר ניתוחTeVורה מוצפנת **ambilatively** לפגוע בפרטיות המשתמש, מה שהופך את השיטה לעיליה יותר מבחינה אתית ובטיחותית.

## **לseiכום:**

החדשונות המרכזי במאמר אינו במאייניTeVורה עצם, אלא באופן שבו הם משלבים יחד ומיצגים בצורה ויזואלית. השיטה החדשה משתמשת בטכניקות עיבוד תמונה ולמידה عمוקה, מה שmobiel לשיפור ממשמעותי בדיקת הסיווג של סוגTeVורה בראשת.

## תוצאות המרכזיות ומסקנות מתוך המאמר:

### 1. השוואה לשיטות קודמות (טבלה 3):

הטבלה מראה את דיק סיוג המוצע שהושג על ידי שיטות שונות על מערך הנתונים ISCX VPN-nonVPN.

כפי שמצוין במאמר, קשה להשוות ישירות בין התוצאות עקב הבדלים בהגדרות הבעה ובקטגוריות התבורה.

למרות זאת, הטבלה מראה שהגישה המוצעת (FlowPic + CNN) משיגה דיק סיוג ממוצע גבוה יותר מרוב השיטות האחרות, במיוחד שלא משתמשות נתונים מטען (payload).

Problem	FlowPic Acc. (%)	Best Previous Result	Remark
<i>Non-VPN Traffic Categorization</i>	85.0	84.0 % Pr., Gil et al. [15]	Different categories. [15] used unbalanced dataset
<i>VPN Traffic Categorization</i>	98.4	98.6 % Acc., Wang et al. [7]	[7] Classify raw packets data. Not including browsing category
<i>Tor Traffic Categorization</i>	67.8	84.3 % Pr., Gil et al. [15]	Different categories. [15] used unbalanced dataset
<i>Non-VPN Class vs. All</i>	97.0 (Average)	No previous results	
<i>VPN Class vs. All</i>	99.7 (Average)	No previous results	
<i>Tor Class vs. All</i>	85.7 (Average)	No previous results	
<i>Encryption Techniques</i>	88.4	99. % Acc., Wang et al. [7]	[7] Classify raw packets data, not including Tor category
<i>Applications Identification</i>	99.7	93.9 % Acc., Yamanasavascilar et al. [10]	Different classes

### 2. דיק סיוג גבוה לקטגוריות תעבורת (טבלה 4):

דיק סיוג עבור קטגוריות התעborות (VoIP, וידאו, העברת קבצים, צ'אט, גלישה) הוא גבוה מאוד. במיוחד עבור תעבורת Non-VPN ו-VPN.

עבור Non-VPN, הדיק נع בין 93.3% (צ'אט) ל-99.6% (וידאו).

עבור VPN, הדיק נע בין 83.6% (צ'אט) ל-99.9% (וידאו).

סיוג תעבורת Tor מציג דיק נמוך יותר, אך עדין סביר (בין 57.2% לגלישה ל-90.6% לוידאו).

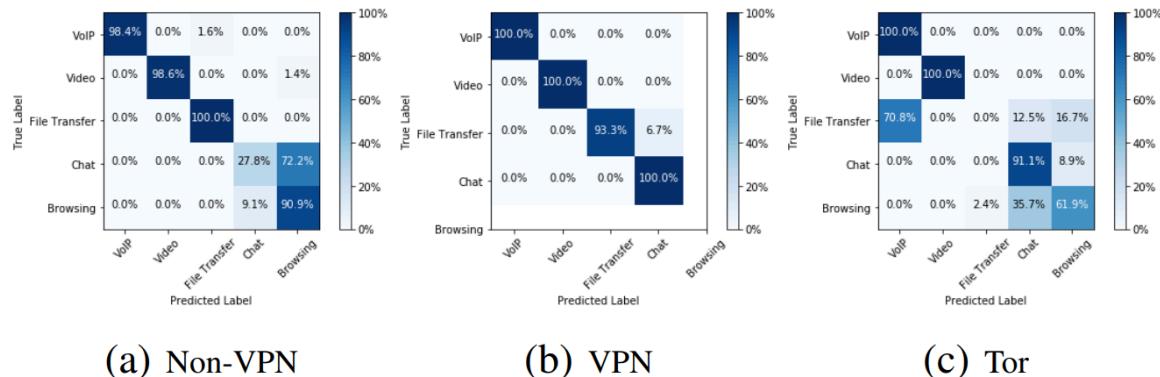
Class	Accuracy (%)			
	Training/Test	Non-VPN	VPN	Tor
<i>VoIP</i>	Non-VPN	<b>99.6</b>	99.4	48.2
	VPN	95.8	<b>99.9</b>	58.1
	Tor	52.1	35.8	<b>93.3</b>
<i>Video</i>	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	<b>99.9</b>	98.8	83.8
	VPN	54.0	<b>99.9</b>	57.8
<i>File Transfer</i>	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	<b>98.8</b>	79.9	60.6
	VPN	65.1	<b>99.9</b>	54.5
<i>Chat</i>	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	<b>96.2</b>	78.9	70.3
	VPN	71.7	<b>99.2</b>	69.4
<i>Browsing</i>	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	<b>90.6</b>	-	57.2
	VPN	-	-	-
	Tor	76.1	-	<b>90.6</b>

3. מטריצות בלבול לסיווג קטגוריות (איור 4):

איור 4 מציג מטריצות בלבול עבור סיווג קטגוריות התעבורה עבור VPN ו-Non-VPN.

מטריצות אלו מאפשרות לראות אילו קטגוריות מתבלבלות זו בזו.

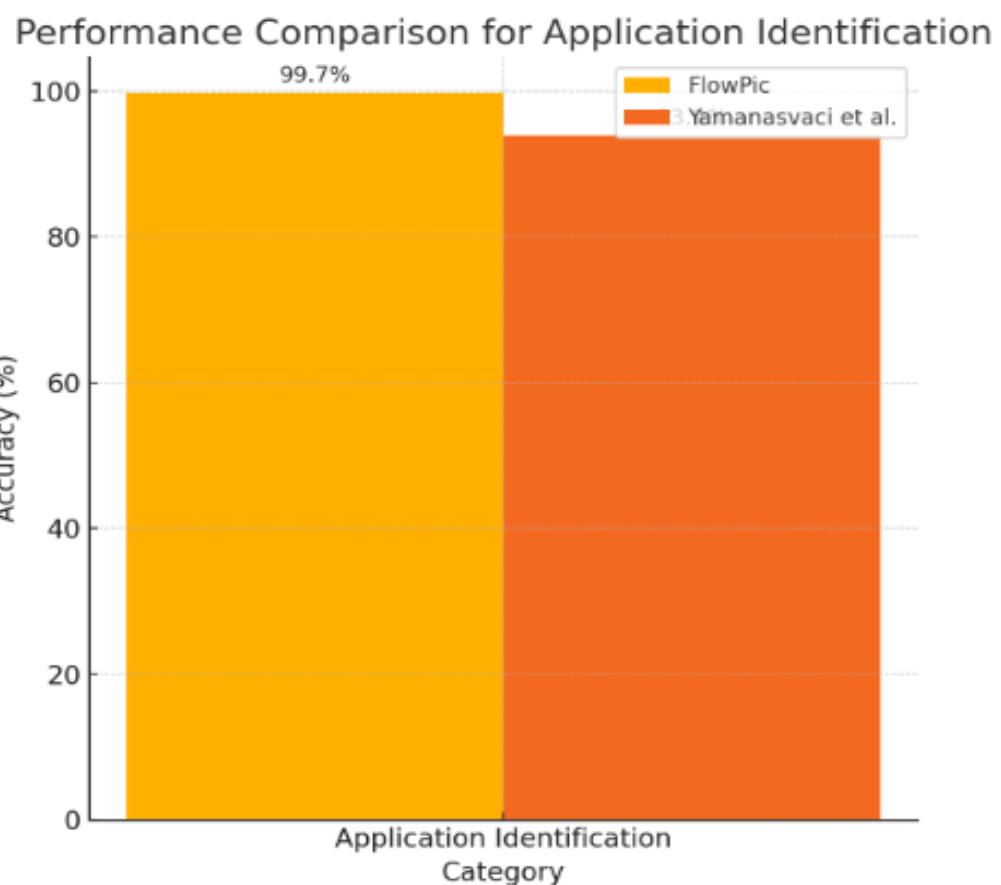
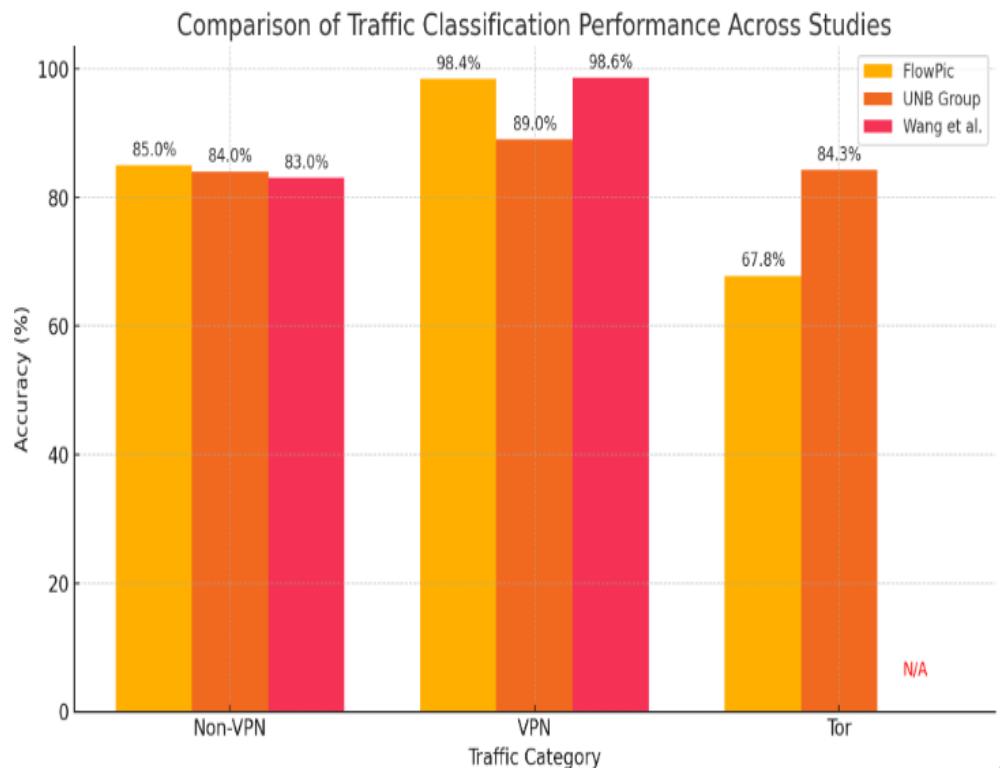
ניתן לראות שה הבלבול העיקרי בתעבורה Non-VPN ו-VPN הוא בין צ'אט לגילשה, בעוד שבתעבורה Tor הבלבול גדול יותר בין כל הקטגוריות.



### מסקנות עיקריות:

1. הגישה משפרת את ביצועי הסיווג בהשוואה לשיטות קודמות: טבלה III מספקת אינדיקציה לכך שהגישה המוצעת משפרת את ביצועי הסיווג בהשוואה לשיטות קודמות, במיוחד, במקרים של שימוש בנתוני מטען.
2. הגישה יعلاה לסיווג תעבורת רשת מוצפנת (במיוחד VPN): הדיק הგבה בסיווג תעborת VPN (טבלה VII) מצביע על כך שהגישה יعلاה במיוחד לסיווג תעborת מוצפנת, וזהו יתרון משמעותי.
3. סיווג תעborת Tor מأتגר יותר: הדיק הנמוך יותר בסיווג תעborת Tor מצביע על כך שסוג זה של תעborה קשה יותר לסיווג, כנראה בגלל המאפיינים הייחודיים של Tor (שכבות הצפנה רבות, שינוי נתיבים תדיר).
4. קיימים הבדלים במאפיינים של קטגוריות תעborה שונות: ההבדלים בדיק הסיווג בין הקטגוריות השונות (טבלה VII) מצביעים על כך שלכל קטgorיה יש מאפיינים ייחודיים המקלים או מתקשים על הסיווג. לדוגמה, תעborת וידאו קלה יותר לסיווג מתעborות צ'אט.

על מנת להמחיש את התוצאות بصورة מיטבית יצרנו גרפ' שמשווה את תוצאות הניסוי של סיווג התעבורה של FlowPic לעומת מחקרים אחרים שעשו UNB Group ו- Wang et al, ובנוסף, גרפ' המשווה בין דוחי האפקציות בהן השתמשו במחקר לעומת המחבר השני שלו בוצאה ההשווואה:



## מאמר מספר 2 :

"Early Traffic Classification With Encrypted ClientHello A Multi Country Study"

### מהי התרומה המרכזית של המאמר?

המאמר עוסק בסיווג תעבורה מוקדם בסביבת Encrypted ClientHello, המנסה על דיזיין סוג התעבורה בשלב הצפנה מידע רגיש-ב-TLS handshake. כתרומה מרכזית, המאמר מציג אלגוריתם חדש בשם hRFTC TLS (hybrid Random Forest Traffic Classifier) עם מאפיינים סטטיסטיים של זרימות התעבורה, כמו גודל מנוט וסדרות זמן מבוססות זרימה. בנוסף, המאמר מציג מערך נתונים מקיף ועכני שאוסף מדיניות שונות בצוות ארה"ב, אירופה ואסיה, הכלול למעלה מ-600,000 זרימות TLS המחלקות ל-19 סוגים שונים, ומאפשר מחקר מעמיק של ביצועי אלגוריתמים שונים בסביבת ECH. המאמר מנהת את יכולת של אלגוריתמים קיימים להתמודד עם אטגרי-h-ECH ומדגים את החשיבות של התאמת מודלים של סיווג תעבורה למיקומים גיאוגרפיים שונים עקב הבדלים בדףי התעבורה. מחקר זה מראה כי אלגוריתם hRFTC עולה בביצועו על אלגוריתמים חדשים אחרים, ומספק פתרון יעיל לסיווג תעבורה מוקדם בסביבות רשת מודרניות.

### באילו מאפייני תעבורה המאמר משתמש, ואילו מהם הם חדשים?

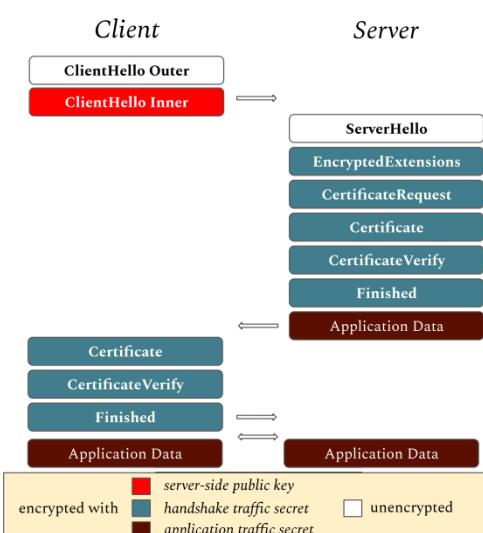
#### סיווג תעבורה (TC-TRAFFIC CLASSIFICATION)

הוא תהליך לזהותoSיווג של תעבורה רשות על פי מאפיינים שונים ומטרתו היא לשפר את אינטואטיבית השירות ברשות על ידי התאמת אוטומטית של משאבי כוח אדם רוחב פוטו.

#### סיווג תעבורה (TLS)

בתחילת תהליך handshake, הלוקו שולח לשרת הודעה ClientHello, והשרת משיב בהודעת ServerHello. הודעה ClientHello מכילה מידע רגיש, כולל Client Name Indication (SNI), אשר חושף את שם הדומיין אליו הלוקו מנסה להתחבר.

בגרסת TLS 1.3, כל השלבים הקרייטיים בתהליך handshake מוצפנים, וכן התקשרות המאובטחת בין הצדדים. לאחר השלמת handshake, כל המידע המועבר בין הלוקו לשרת מוצפן, כך שרק הצדדים המשתתפים בתקשורת יכולים לגשת אליו.



## :ECH

ECH מהו שיפור ממשוני לTLS 1.3, הוא מגן על פרטיות המשתמשים בשלב קרייטי של הקמת החיבור המאובטח (SNI).

יש כמה תרומות מרכזיות של המאמר על ECH:

1. ייצור מאגר נתונים מהגדולים והמעודכנים ביותר לשילוב מוקדם של תעבורת TLS, הכול כולל מעל 600,000 זרים מ-19 קטגוריות שונות שנאספו מאזרחים אגראפיים מגוונים, ומספק תמונה ריאלית של תעבורת האינטרנט המודרנית.
2. פיתוח אלגוריתם חדש (hRFTC) - אלגוריתם היבידי חדש לשילוג תעבורת המשתמש במאפייני TLS לא מוצפנים ובמאפייני זרימה סטטיסטיים.
3. השוואת ביצועים - הוכחת עליונות האלגוריתם החדש על פני אלגוריתמים מתקדמים אחרים עם דיוק סיווג של עד 94.6%.
4. בדיקת הטורוגניות בתנאים - ניתוח כיצד אלגוריתמי סיווג מתמודדים עם שונות אגראפית ושירותים שונים.

יש שני סוגים מאפיינים עיקריים לשילוב מוקדם של תעבורת רשת:

מאפיינים מבוססי חבילות (Packet-based Features): שילוב של מידע מתוך הودעות ClientHello ו-ServerHello שאין מוצפנות, כיל גדול ותוכן של הודעות TLS, תכניות בהודעות התחלה והעדפות הצפנה, המאפשרות>Zרימה ראשוני של סוג התעבורת.  
מאפיינים מבוססי זרימה (Flow-based Features): שימוש בסטטיסטיות של גדי חבילות (Packet Size - PS), זמן הגעה בין חבילות (IPT - Inter-Packet Time), והבחנה בין תעבורת העלה (uplink) להורדת (downlink), מה שמאפשר ניתוח עמוק של התנהגות התעborה לארוך זמן.

המאמר מציע אלגוריתם חדש, Hybrid Random Forest Traffic Classifier (hRFTC), המיועד לשילוג מוקדם של תעבורת רשת מוצפנת, במיוחד בתחום השימוש בתפקידים שביהם נעשה שימוש ב-*Encrypted ClientHello*-ב-*TLS* (ECH) בפרוטוקול *TLS*. האלגוריתם משלב בין מאפייני חבילות *TLS* שאין מוצפנות לבין מאפיינים סטטיסטיים מבוססי זרימה, כגון גודל חבילות, זמן הגעה ופיזור תעבורת העלה והורדת, במטרה לשפר את דיוק הסיווג גם כאשר מידע קרייטי כמו IP/MNS מוצפן.

החדשניים המרכזיים של hRFTC כוללים:

- שימוש משולב במאפייני חבילות וזרימה: שילוב מידע מתוך ה-*TLS handshake* הבלתי מוצפן יחד עם נתוני זרימה סטטיסטיים משפר את יכולת הסיווג גם במצבים בהם חלק גדול מהמידע מוסתר.
- תמייהה בתעborה מבוססת QUIC: האלגוריתם מתרחב מעבר ל-TLS-over-TCP ותומך גם בפרוטוקול QUIC, באמצעות עיבוד המידע הבלתי מוצפן של TLS ושימוש בסטטיסטיות זרימה מתקדמות.
- קרייטריון חדש לבחירת חבילות לשילוג: מקום לנתח מספר קבוע של חבילות, האלגוריתם מבצע ניתוח עד לחבילה הראשונה המכילה נתוני אפליקציה, מה שמאפשר סיווג מהיר ומדויק יותר.

אלגוריתם hRFTC מבוסס על Random Forest, אשר מאפשר סיווג יעיל ומדויק של תעborת רשת מוצפנת, גם כאשר מידע קרייטי כמו IP/MNS מוסתר באמצעות ECH. תוצאות המחקר מצביעות על כך שהאלגוריתם החדש משיג דיוק של עד 94.6%, מה שמצויב אותו מעל האלגוריתמים המובילים ביום בתחום סיווג התעborה המוצפנת.

## מהם הממצאים המרכזיים, ואילו תובנות עלות מהם?

1. **דיק גבוח בסיווג התעבורה – האלגוריתם החדש hRFTC** משיג דיק של עד 94.6% מה שמעלה אותו מעל האלגוריתמים המובילים הקיימים, במיוחד בתרחישים שבהם נעשה שימוש ב-ECH.
2. **שיפור משמעות שיטות קיימות – אלגוריתמים המבוססים רק על מאפייני TLS בלבד** מוצפנים הגיעו לדיק של 38.4% בלבד, בעוד שהשילוב של מאפייני חבילות וזרימה ב-hRFTC שיפור משמעותי את ביצועי הסיווג.
3. **שונות גיאוגרפית – נמצא כי אלגוריתמים המאומנים במדינה מסוימת אינם בהכרח יעילים במדינות אחרות**, מה שנדגיש את הצורך באימון מחדש בהתאם למיקום הגיאוגרפי.
4. **תרומת מאפייני זרימה – מאפייני זרימה, כמו גודל חבילות וזמן הגעה בינהן, תרמו תרומה משמעותית לסיווג התעבורה, והוכחו חשובים לא פחות ממאפייני TLS.**
5. **התמודדות עם הצפנה ECH – למרות שהצפנה-hello מקשה על הסיווג המסורתי,** השיטה ההיברידית של hRFTC מצליחה להתגבר על האתגר באמצעות שילוב חכם של נתונים סטטיסטיים ומידע בלתי מוצפן.

Test Country	Share in Dataset	Training Country	Classifier Macro F-score [%]		
			hRFTC	hC4.5	UW
Germany	18.8%	Others	38.4	26.9	19.5
Kazakhstan	3.0%	Others	57.3	32.3	27.5
Russia	29.2%	Others	49.8	35.6	20.9
Spain	16.3%	Others	38.5	34.4	12.6
Turkey	25.2%	Others	35.1	26.0	16.4
USA	7.5%	Others	49.2	41.4	21.3

### תובנות:

1. **השפעת ECH על סיווג תעבורה:**  
השימוש ב-ECH (*Encrypted ClientHello*) מצריך משמעותית את יכולת הסיווג של שיטות מסורתיות, לאחר שהוא מציג מידע קרטיס כמו *N*.SN. כתוצאה לכך, יש צורך בגישה היברידית המשלבת מאפייני חבילות וזרימה כדי לשמר את דיק הסיווג.
2. **תרומת מאפייני זרימה לסיווג מדויק:**  
פרמטרים כגון גודל חבילות, זמן הגעה והבדלים בין תעבורת העלה (uplink) להורדת (downlink) מספקים מידע חיוני לדחייה סוגית שירותים שונים גם בתעבורה מוצפנת.
3. **התמודדות עם QUIC:**  
השימוש ב프וטוקול QUIC מוסיף מורכבות בשל הצפנה-TLS handshake וקיים מתקדמות מואפשר עדין סיווג מדויק של התקשורת. עם זאת, שילוב סטטיסטיות זרימה מתקדמות מואפשר עדין סיווג מדויק של התעבורה.
4. **צורך באימון מותאם גיאוגרפית:**  
בשל הבדלים בתשתיות רשת, ספקי CDN, והגדירות השירותים במדינות שונות, נדרש לבצע אימון חדש של המודל לכל אזור גיאוגרפי כדי לשפר את ביצועי הסיווג ולהתאים אותו לתנאי הרשות המקומית.

## מאמר מס' 3 :

"Analyzing HTTPS Encrypted Traffic to Identify User's Operating System, Browser and Application"

### מהי התרומה המרכזית של המאמר?

התרומה המרכזית של המאמר היא הצגת גישה תקיפה 'פאסיבית' המנצלת נקודות תורפה באפליקציה HTTPS מוצפנת, ובכך מאפשרת זיהוי מדויק של מערכת הפעלה, הדפדפן והאפליקציה קבוצה שבהם המשמש משתמש. המחברים מראים כי למטרות השימוש בפרוטוקול הגנה כמו SSL/TLS, ניתן לנצל את המאפיינים החיצוניים של התעבורה – הן באמצעות תכונות סטטיסטיות בסיסיות והן באמצעות תכונות חדשות המבוססות על דפוסי ההתנהגות המתפרצת (bursty) של הדפסנים והתנהגות-SSL – כדי להגיע לדיווק סיווג גבוה (93.51% בתכונות בסיסיות, עד 96.06% בשילוב עם התכונות החדשניות). בנוסף, המאמר מלווה ב-data set שנאסף באופן אוטומטי, המאפשר אימון ובדיקה מדויקת של המודל.

כך, תרומתו המרכזית טמונה בכך שהמחקר מוכיח את האפשרות לזהות פרטן מערכת, דפדפן ואפליקציה באמצעות שהתעבורה מוצפנת, דבר המצביע על נקודות תורפה בשיטות ההצפנה הקיימות כיום.

### באילו מאפייני תעבורת המאמר משתמש, ואילו מהם הם חדשניים?

המאמר עושה שימוש בשתי קבוצות מאפיינים: קבוצת תכונות בסיסיות הקשורות לתעבורת הרשת, וקבוצת תכונות חדשות שהוצעו על ידי החוקרים לשיפור הדיווק. השילוב בין שתי הקבוצות מאפשר לזהות באופן מדויק את מערכת הפעלה, סוג הדפדפן והאפליקציה קבוצה של המשתמש, באמצעות קריאת הפקות בתעבורת HTTPS.

התכונות החדשניות בהם החוקרים הוסיפו לטובת דיווק הממצאים

TCP initial window size
TCP window scaling factor
# SSL compression methods
# SSL extension count
# SSL cipher methods
SSL session ID len
Forward peak MAX throughput
Mean throughput of backward peaks
Max throughput of backward peaks
Backward min peak throughput
Backward STD peak throughput
Forward number of bursts
Backward number of bursts
Forward min peak throughput
Mean throughput of forward peaks
Forward STD peak throughput
Mean backward peak inter arrival time diff
Minimum backward peak inter arrival time diff
Maximum backward peak inter arrival time diff
STD backward peak inter arrival time diff
Mean forward peak inter arrival time diff
Minimum forward peak inter arrival time diff
Maximum forward peak inter arrival time diff
STD forward peak inter arrival time diff
# Keep alive packets
TCP Maximum Segment Size
Forward SSL Version

(b) new features

התכונות הבסיסיות משמשות להרבה סיווגי תעבורת

# Forward packets
# Forward total Bytes
Min forward inter arrival time difference
Max forward inter arrival time difference
Mean forward inter arrival time difference
STD forward inter arrival time difference
Mean forward packets
STD forward packets
# Backward packets
# Backward total Bytes
Min backward inter arrival time difference
Max backward inter arrival time difference
Mean backward inter arrival time difference
STD backward inter arrival time difference
Mean backward packets
STD backward packets
Mean forward TTL value
Minimum forward packet
Minimum backward packet
Maximum forward packet
Maximum backward packet
# Total packets
Minimum packet size
Maximum packet size
Mean packet size
Packet size variance

(a) base features

## מהם הממצאים המרכזיים, ואילו תובנות עלות מהם?

**הממצאים** מסתמכים על **data set** שנagar ע"י אוטומטית רשת בשם 'סלניום' אשר מגיעה עם **crawlers** (תוכנה הסורקת אוטומטית דף אחר דף, אינדקסים וקישורים) וכל התעבורה הוקלט דרך פорт 443 (TLS) ולאחר מכן קתlico את התעבורה לקטעים של פעילות. **data set** מכיל יותר מ 20 אלף פעילות.

לטובת האנליה השתמשו במבנה מלכוטית מבוקרת אשר מקבלת דוגמית של פעילות מסוימת ומחזירה קטלוג של פעילות ע"י טפל. למידת המכונה התחילה ב- 70 אחוז אימון ו 30 אחוז בדיקות.

הממצאים מוצגים בתור "מטריצת בלבול" אשר מציגה כיצד ה-Y את הפעולות האמיתית שמתקיימת אשר מתייחסת בתור >OS, Browser, Application< וביציר ה-X מופיעות פעילות שהאלגוריתם ניבא באותו קטגוריות. כל תא בטבלה מציג את ההסתברות שבה האלגוריתם הצליח לזהות נכון את סוג הפעולות בהתאם על הפרמטרים והתכונות שנבחרו לחילוץ המידע.

		Predicted labels																				
		Real labels																				
		Windows IExplorer Twitter	Ubuntu Firefox Google-Background	Windows Non-Browser Microsoft-Background	Windows Chrome Twitter	Ubuntu Chrome Unknown	OSX Safari Google-Background	OSX Safari Unknown	Ubuntu Firefox Twitter	Ubuntu Chrome Google-Background	Ubuntu Firefox Twitter	Ubuntu Chrome Unknown	Windows Firefox Google-Background	OSX Safari Unknown	Ubuntu Firefox Unknown	Ubuntu Chrome Unknown	Windows Firefox Unknown	Ubuntu Chrome Unknown	Windows Non-Browser Teamviewer	Ubuntu Chrome Unknown	Windows Non-Browser Dropbox	Ubuntu Chrome Unknown
Windows IExplorer Twitter	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Ubuntu Firefox Google-Background	0	.97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Windows Non-Browser Microsoft-Background	0	0	.99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Windows Chrome Twitter	0	0	0	.99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Windows Firefox Twitter	0	0	0	0	.98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
OSX Safari Google-Background	0	0	0	0	0	.92	.04	0	0	0	.02	0	0	0	0	0	0	0	0	0	0	
OSX Safari Youtube	0	0	0	0	0	0	.97	.01	0	0	0	0	0	0	0	0	0	0	0	0	0	
Ubuntu Chrome Unknown	0	0	0	0	0	0	.84	0	0	0	.07	.04	0	0	0	0	.01	0	.03	0	0	
Windows Chrome Google-Background	0	0	.01	.03	0	0	0	.94	0	0	0	0	.02	0	0	0	0	.01	0	0	0	
Ubuntu Firefox Twitter	0	0	0	0	0	0	0	.95	0	.03	0	0	0	.01	0	0	0	0	0	0	0	
OSX Safari Unknown	0	0	0	0	0	.06	.01	0	0	.91	0	0	0	.01	0	0	0	0	0	0	0	
Ubuntu Firefox Unknown	0	0	0	0	0	0	0	0	.05	.87	0	0	0	0	.01	0	0	.02	0	0	0	
Ubuntu Chrome Google-Background	0	.07	0	0	0	0	0	.18	0	0	0	.73	0	0	0	0	0	0	.02	0	0	
Ubuntu Chrome Twitter	0	.02	0	0	0	0	0	.08	0	0	0	.03	.84	0	0	0	.01	0	.01	0	0	
Windows Firefox Google-Background	0	0	0	.01	0	0	0	0	.01	0	0	0	.97	0	0	0	0	.01	0	0	0	
OSX Safari Twitter	0	0	0	0	0	0	.06	0	0	0	.03	0	0	.91	0	0	0	0	0	0	0	
Ubuntu Firefox Youtube	0	.02	0	0	0	0	0	0	.02	0	.02	0	0	.93	0	0	0	0	0	0	0	
Windows Non-Browser Teamviewer	0	0	0	0	0	0	0	0	0	0	0	0	0	.01	0	0	0	0	0	0	0	
Ubuntu Chrome Youtube	0	0	0	0	0	0	.07	0	0	0	.13	.04	0	0	.02	.74	0	.02	0	0	0	
Windows Non-Browser Dropbox	0	0	0	0	0	0	0	0	0	0	.02	0	0	0	0	0	.01	0	0	0	0	
Windows Chrome Unknown	0	0	.02	.09	0	0	0	0	.02	0	0	0	0	0	0	0	.03	0	0	0	0	
Ubuntu Chrome Facebook	0	0	0	0	0	0	.3	0	0	0	.04	0	0	0	0	0	.01	0	0	0	0	
Windows Firefox Unknown	0	0	.06	0	0	0	0	0	0	0	0	0	0	0	0	0	.02	0	0	0	0	
Ubuntu Firefox Facebook	0	.06	0	0	0	0	0	0	.11	0	.28	0	0	0	0	0	.05	.56	0	0	0	
OSX Chrome Twitter	0	0	0	0	0	0	.13	0	0	0	0	0	0	0	0	0	.75	0	0	.29	.06	
Windows IExplorer Unknown	.71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Ubuntu Non-Browser Microsoft-Background	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
Windows IExplorer Google-Background	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
OSX Chrome Google-Background	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
OSX Chrome Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

(a) Tuple Confusion Matrix

Predicted labels

Real labels	Windows	Ubuntu	OSX
Windows	1	0	0
Ubuntu	0	1	0
OSX	0	0	1

(b) OS Confusion Matrix

Predicted labels

Real labels	Chrome	Firefox	IExolorer	Safari	Non-Browser
Chrome	.97	.02	0	0	0
Firefox	.01	.98	0	0	0
IExolorer	0	0	1	0	0
Safari	.01	0	0	.99	0
Non-Browser	.03	0	0	0	.96

(c) Browser Confusion Matrix

Predicted labels

Real labels	Dropbox	Facebook	Google-background	Microsoft-Background	Teamviewer	Twitter	Youtube	Unknown
Dropbox	.98	0	.02	0	0	0	0	0
Facebook	0	.62	.04	0	.04	.29	0	.03
Google-background	0	0	.95	0	0	.01	.01	.04
Microsoft-Background	0	0	0	.96	0	0	0	.04
Teamviewer	0	0	0	0	1	0	0	0
Twitter	0	0	0	0	0	.98	0	.01
Youtube	0	0	.03	0	0	.02	.93	.01
Unknown	0	.02	.04	.01	0	.05	.01	.86

(d) Application Confusion Matrix

## **תובנות:**

הסיווג לרוב הפעולות היה כמעט מושלם, אם דיק ש-ל כמעט 1 לרבות טופלים עם שם זהה וגם טאפלים שלא ניתן לזהות. סיווג מערכות הפעלה הוא מושלם ללא טעויות, סיווג סוג הדדרפנום הוא כמעט מושלם, אך סיווג סוג האלפיקציות גם כמעט מושלם למעט פיסבוק, האלגוריתם טעה בו ב-29 אחוז מזמן הלמידה וקיטל-אotto בתור אפליקציה לא ידועה.

לסיכום, ניתן לזהות תעבורת מוצפנת ולהוציא מהם מידע כגון איזה סוג מערכת הפעלה, דפדפן ואפליקציה המשמש משתמש דרך המחשב האישי או הנייד שלו. ניתן לראות, למרות שימוש בפרוטוקול הגנה כגון SSL/TLS, שניצול נקודות התורפה באפליקציית HTTPS מאפשר לתקוף לזהות דפוסים התנהגותיים ולהסיק מידע רגיש אודiot המستخدم, מה שעלול לסכן את פרטיותו ואף לאפשר מתקפות ממוקדות.

## חלק 3: ניתוח תעבורת רשת ויזיהו יישומים באמצעות Wireshark: אפיון יישומים וסימולציה תקיפה

### 1. הקלטת תעבורת רשת ואיסוף נתונים

- הגדרת מתודולוגיית הניסוי והגבלה "רעש" תעבורתי
- אפיון סוג היישומים הנבדקים
- שמירת מפתחות TLS לצורך ניתוח נתונים מוצפנים

### 2. ניתוח והשוואת מאפייני התעבורה בין יישומים

- מאפייני התעבורה הנבדקים (שדות כתובת IP, שדות כותרת TCP, שדות TLS, גודל חבילות, זמני הגעה, נפח התעבורה)
- הצגת גרפים וניתוח השוואתי של סוג התעבורה
- דפואים שימושיים ייחודיים לכל יישום

### 3. הסבר מפורש של הגרפים והמצאים

- השוואת גודל חבילות בין היישומים
- השפעת זמני הגעה בין חבילות על זיהוי סוג השירות
- זיהוי דפואים תעבורה ייחודיים

### 4. סימולציה של תקיפה לזיהוי אפליקציות על בסיס תעבורה מוצפנת

- תרחיש 1: התקוף מכיר את גודל חבילות, חוותות זמן, ו-*hash* של מזחה הזרימה
- תרחיש 2: התקוף מכיר רק את גודל חבילות חוותות הזמן
- ניתוח יכולת של התקוף לזהות את האפליקציה לפי הנתונים הזמןניים
- שיטות למזער הסיכון ודריכים למניעת זיהוי

### 5. ניסוי בונוס: תעבורת רקע והשפעת שימוש משולב ביישומים

- השפעת יישומים שונים על ניתוח תעבורה (למשל, גלישה ברקע תוך כדי האזנה לספוטיפי)
- השפעת תעבורה מעורבת על יכולת הזיהוי של התקוף
- הצעת ניסויים חלופיים והשלכותיהם

### **3.1. הקלטת תעבורת רשות אויסוף נתונים**

#### **תיאור תהליך ההקלטה והלכידה של התעבורה**

במהלך העבודה, השתמשנו בתוכנת **Wireshark** כדי ללכוד ולנתח את תעבורת הרשות של מספר אפליקציות נפוצות בהתאם להנחיות. לצורך הבדיקה, כל אפליקציה נבדקה בנפרד על מנת למנוע "רעש" (תעבורה לא רצiosa) ולאפשר ניתוח מדויק של תוכנות התעבורה. בנוסף, מפתחות ההצפנה נשמרו מראש כדי לאפשר פענוח של התעבורה המוצפנת לאחר מכן.

**אנו מודגשים כי ביטלנו את פרוטוקול QUIC על מנת שנוכל לפתח את ההצפנה של המפתחות בצורה מיטבית וולכן אנו מתייחסים לתוצאות בצורה כזו שהוא לא מופיע!**

פירוט על כל אחת מההקלטות השונות:

#### **1. הקלטת תעבורת גלישה בדף Chrome**

- נכנסנו לאתר **Mako** לרשימת הפרויקטים של הסדרה הפרלמנט דרך דף דפן.
- המתנו לטעינה מלאה של העמוד כדי להבטיח שכל הבקשות נטעוו במלואן.
- לאחר סיום הטעינה, עצרנו את ההקלטה ושמרנו את מפתחות ההצפנה (SSL/TLS) על מנת לפענוח את המידע המוצפן בניתוח מאוחר יותר.

#### **2. הקלטת תעבורת גלישה בדף Firefox**

- חזרנו על התהליך עם אותו עמוד באתר **Mako**, הפעם באמצעות דף דפן **Firefox**.
- שוב, המתנו לטעינה מלאה לפני עצירת ההקלטה ושמרנו את מפתחות ההצפנה לצורך פענוח.

#### **3. הקלטת תעבורת הזרמת שמע (Audio Streaming) ב- Spotify**

- נכנסנו ל-**Spotify** והפעילנו הזרמת שמע.
- על מנת לנתח את הבקשות המוצפנות, ביטלנו את פרוטוקול **QUIC** המוגדר כברירת מחדל.
- פעולה זו אילצה את התקשרות לעברו דרך פרוטוקול **TCP**, מה שאפשר ל透סוס את הבקשות באמצעות Wireshark ולפענוח את הנתונים המוצפנים באמצעות מפתחות ההצפנה.

#### **4. הקלטת תעבורת הזרמת וידאו (Video Streaming) ב- YouTube**

- גלשנו ל-**YouTube** דרך דף דפן **Google Chrome** והפעילנו סרטון וידאו.
- במהלך ההקלטה סיננו את התעבורה כך שתציג רק התעבורה הרלוונטיות לשרת הווידאו שמבצע את הסטרימינג.
- פעולה זו אפשרה לנתח את תוכנות התעבורה הרלוונטיות בלבד, כמו גודל הפקודות והפרוטוקולים המעורבים.

#### **5. הקלטת תעבורת שיחת וידאו (Video Conference) ב- Zoom**

- פתחנו פגישה ב-**Zoom** והוספנו מספר משתתפים נוספים.
- במהלך הפגישה, דיברנו במסך מספר שניות כדי לייצר תעבורת וידאו ושמע אמיתית.
- עם סיום השיחה, סגרנו את הפגישה ואת ההקלטה במקביל.

## 3.2. ניתוח והשוואת מאפייני התעבורה בין יישומים

על מנת ליצור את הגրפים השתמשנו בחבילות הבאות שעל הבודק להתקין כאשר הוא מሪץ את הקוד:

### :Matplotlib

למה זה משמש?

ספרייה גרפים שמאפשרת ליצור **גרפים דו-ממדיים** במגוון פורמטים – עמודות, עוגות, קווי מגמה ועוד.

### שימוש עיקרי בפרויקט:

הציג השוואות בין אפליקציות שונות (לפי פרוטוקולים, גודל פקודות וכו').  
הציג תיאורים ויזואליים של התעבורה ברשת.

### :Pyshark

למה זה משמש :

ספריה לניתוח קבצי תעבורת רשת , המאפשרת גישה נוחה לניתוח חבילות.

### שימוש עיקרי בפרויקט :

קריאה וניתוח של קבצי הקלטה של ה - Wireshark .  
חילוץ מידע מהחבילות כמו כתובות IP, פרוטוקולים, פורטים ונתוני TLS להשוואה בין אפליקציות.

### :Pandas

למה זה משמש :

ספריה לניהול וניתוח נתונים בטבלאות.  
גם מכללה כלים מתאימים לשינון, קיבוץ וניתוח סטטיסטי של נתונים.

### שימוש עיקרי בפרויקט :

לשמר את הנתונים שהופקו מקובצי ה- PCAP בטבלאות מסודרות.  
ניתוח נתונים , הכנה להציג גרפית של התוצאות.

### :Os

למה זה משמש :

ספריה מובנית ב Python המאפשרת אינטראקציה עם מערכת הפעלה.  
משמשת בעיקר לניהול קבצים ותיקיות, גישה לנתיבי קבצים ועוד.

### שימוש עיקרי בפרויקט :

איתור כל קבצי ה הקלטה בתיקיה אחת.  
טעינה אוטומטית של הקבצים לניתוח ללא צורך בהזנה ידנית.  
עבודה דינמית עם נתיבים עבור פונקציית הניתוח.

### :Numpy

למה זה משמש?

ספריית Python שמיועדת לעובדה עם מערכים, וקטורים ופעולות מתמטיות מתקדמות.

### שימוש עיקרי בפרויקט:

יצירת מערכים לניהול נתונים גרפים (למשל, מיקומים של עמודות בגרפים).  
חישובים מתמטיים כמו ממוצעים, סטיית תקן ועוד.

### 3.3. הסבר מפורט של הגרפים והמצאים

#### פירוט גרפ A: התפלגות פרוטוקולי IP

##### מה מייצג הגרף?

הגרף מציג את **התפלגות הפרוטוקולים** שנמצאו בטעורות הרשות **כנגד** כמות החבילות שהגיעה בחמש Web-surfing 1, Web-surfing 2, Audio Streaming, Video Streaming, Video (Conference).

##### ציר ה-X (פרוטוקול):

מציג את סוגי הפרוטוקולים ששימושו להעברת נתונים בין השרת ללקוח. הפרוטוקולים המוצגים כוללים:

**UDP:** פרוטוקול מהיר אך לא אמין (לא תיקוני שגיאות), נפוץ לשיחות וידאו ואודיו

**DATA:** תעבורת נתונים גולמית (בדרכ כל עبور שירות סטרימינג או שיחות וידאו).

**DNS:** בקשות לתרגום כתובות דומיין לכתובות IP.

**DTLS (Datagram Transport Layer Security):** גרסה של TLS מעל UDP, משמש לאבטחת תעborות (Zoom-UDP).

**STUN (Session Traversal Utilities for NAT):** פרוטוקול ליזיה כתובות IP חיונית עבור חיבורים.

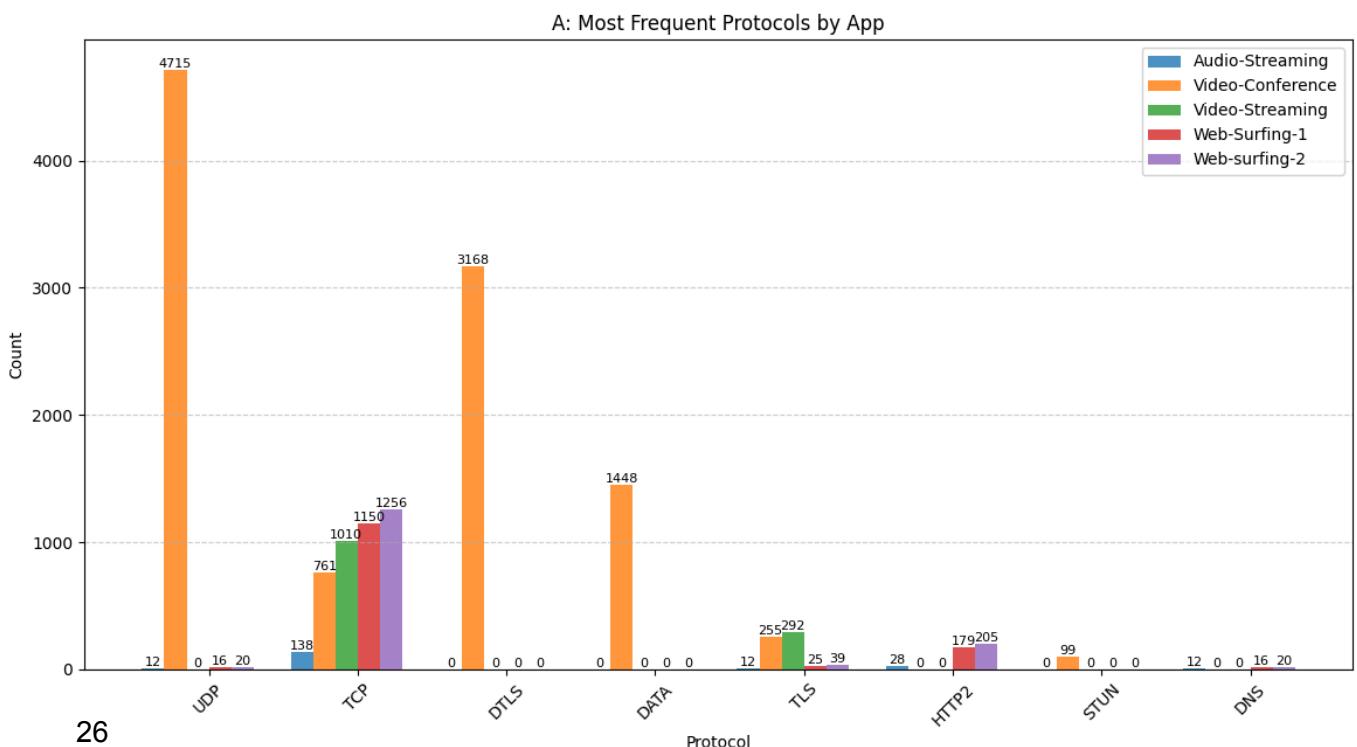
**TCP (Transmission Control Protocol):** פרוטוקול אמין לתקשורת שמבצע בדיקות שגיאות.

**TLS (Transport Layer Security):** שכבה אבטחה על גבי TCP.

##### ציר ה-Y (מספר החבילות):

מצין את מספר החבילות שנפתחו עבור כל פרוטוקול, בכל אפליקציה.

משקף את רמת השימוש של כל פרוטוקול בכלל אחת מהאפליקציות.



## פירוט גרפּ B : עשרת ה포רטים הנפוצים ביותר ב-**TCP**:

### מה מייצג הגרף?

הגרף מציג את 10 ה포רטים הנפוצים ביותר ב-プロトוקול TCP **בנגד** כמות החבילות שהגיעה בתעבורת הרשות שנכלדה ממחש אפליקציות שונות:

**.Web-surfing 1, Web-surfing 2, Audio Streaming, Video Streaming, Video Conference**

### ציר ה-X (פורט המקורי):

מציג את מספר הפורטים ששימשו כנקודת התחלה (Source Port) במחשב הלוקו לשילוח בקשות אל השירות.

כל מספר פורט מייצג "דלת" בתקשרות, שדרכה נשלחים ונקלטים נתונים.

### המשמעות של הפורטים הנפוצים:

**.443**: פורט בירית מיחל עبور תעבורת HTTPS המאובטחת (TCP/TLS)

**80**: פורט בירית מיחל עبور תעבורת HTTP לא מאובטחת.

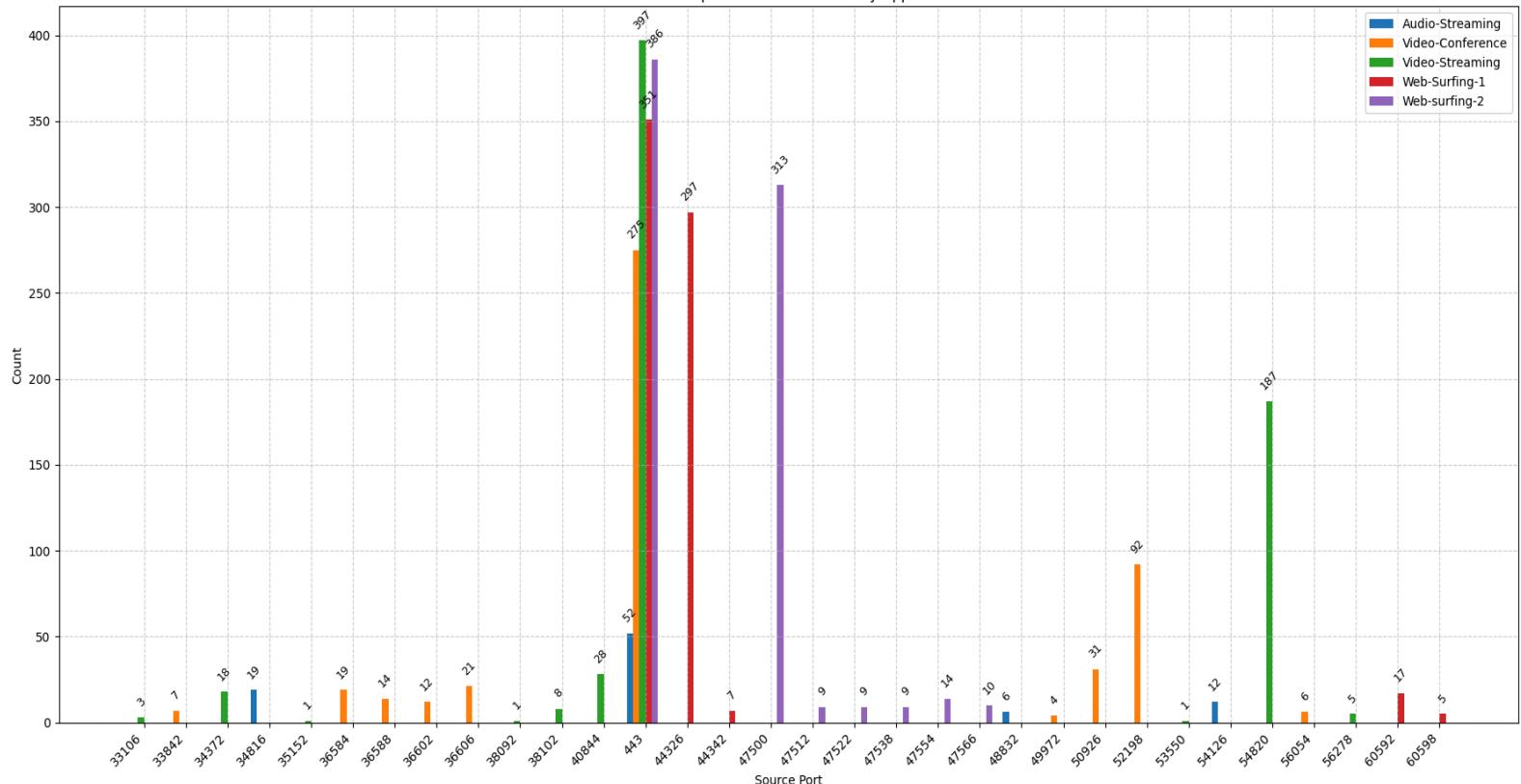
**36584, 57760, 53684, 50026, 52198, 41940, 44326**: פורטים זמינים (Ephemeral Ports) – נוצרים באופן דינמי עבור תקשורת זמנית.

### ציר ה-Y (מספר החבילות):

מצין את מספר הפקודות (packets) שנשלחו דרך כל פורט עבור כל אפליקציה.

ספירה זו מאפשרת לראות אילו פורטים היו בשימוש הכי תדיר בתקשרות בין הלוקו לשרת.

B: Top 10 TCP Source Ports by App



## פירוט גרפּ C: התפלגות זמני הגעה בין-פקטוט:

### מה מייצג הגרף?

הגרף מציג את הזמנים בין הגעת שתי פקודות עוקבות (Inter-arrival time) נגדי הצפיפות של הגעתן בחמשת האפליקציות שנבדקו:

.Web-surfing 1, Web-surfing 2, Audio Streaming, Video Streaming, Video Conference

זמן הגעת הפקודות משקף את דפוסי השימוש של כל אפליקציה ואת סוג התעבורה שלה.

### ציר ה-X (זמן הגעה בין-פקטוטות בשניות)

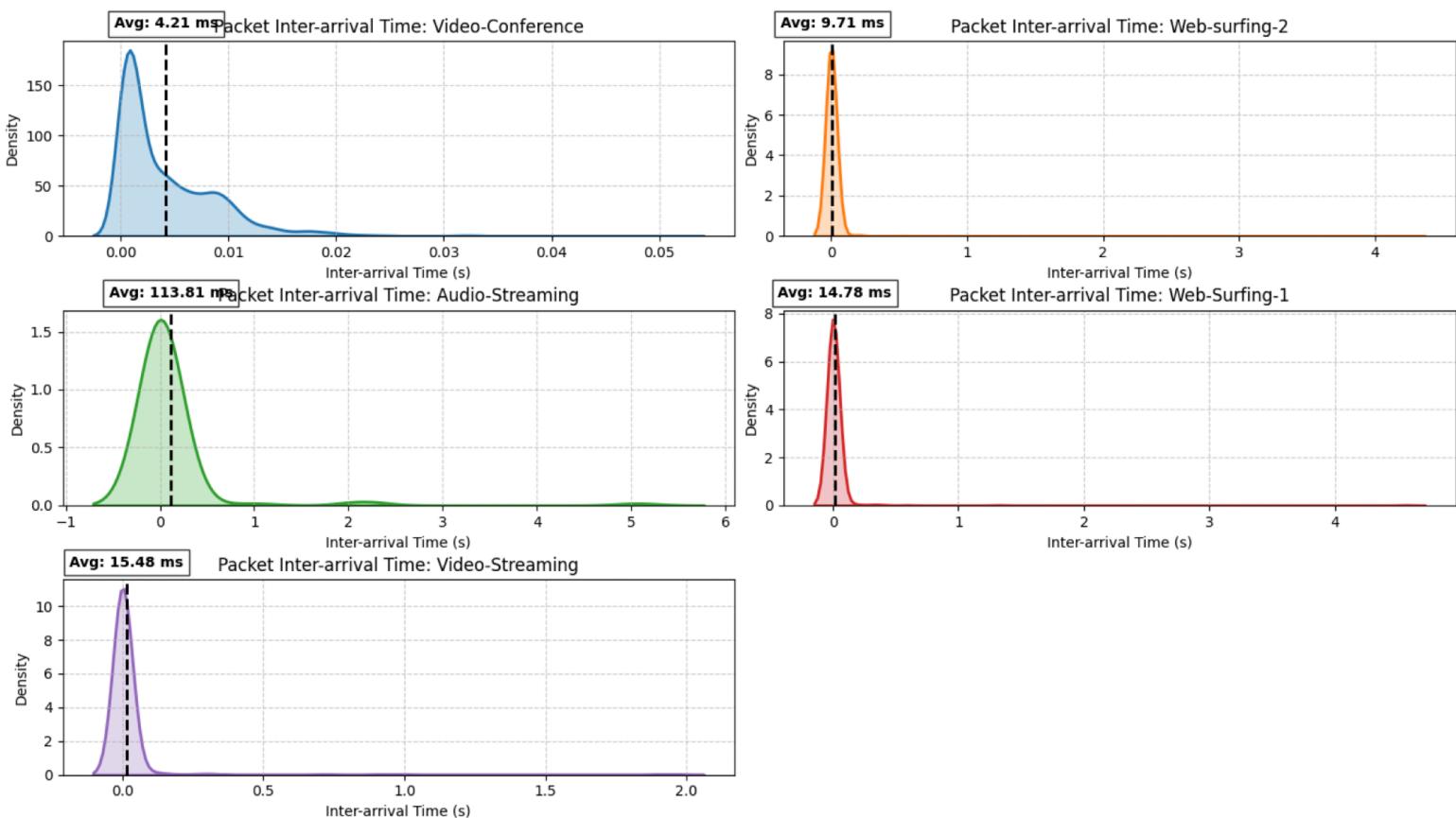
ציר ה-X מציין את הזמן בין כל שתי פקודות רצויות שנשלחו מהרשות ללקוח. הנתונים מוצגים בסולם לוגריתמי (Log Scale), כיוון שההפרטים יכולים לנוע בין מיקרו-שניות (זם) לשניות (ס).

ערכים קטנים יותר ← פקודות מגיעות בקצב מהיר יותר.ערכים גדולים יותר ← פקודות מגיעות בפער זמן גדולים יותר (למשל, כאשר יש המתנה להורדת תוכן נוסף).

### ציר ה-Y (צפיפות):

מייצג את רמת השכיחות של זמנים מסוימים בין חבילות. קווים הצפיפות הגבוהים מצביעים על פרקי זמן בין חבילות שבהם התקשרות מתרכזת.

### C: Packet Inter-arrival Time Distribution by App



## פירוט גרפּ F: התפלגות גודל החבילות:

### מה הגרף מציג?

הגרף מתאר את התפלגות גודל החבילות (Packet Size Distribution) כנגד תדרות הגעתן של החבילות ביחס אפליקציות שונות:

Video-Conference , Video-Streaming , Web-Surfing-1, Web-Surfing-2, Audio-Streaming המטרה של הגרף היא להראות **יכד כל אפליקציה משתמשת בגודל חבילות שונה**, בהתאם לצורכי התעבורה שלה (רכישות, מהירות, אבטחה ויכולות שירות).

### ציר ה-X: גודל החבילה (Packet Size - Bytes):

מתאר את הגודל של כל חבילה מידע שנשלחה בתעבורה של האפליקציה.

החבילות נמדדות ב-Bytes (בתים) וכוללות:

כותרות (Headers) – מידע פרוטוקולי כמו IP, TCP, TLS.

נתוני תוכן (Payload) – המידע עצמו (שמע, וידאו, תוכן אינטרנט וכו').

קיימים טווח רחב של גדלים – החל מחבילות קטנות מאוד (כמה עשרות Bytes) ועד חבילות גדולות מאוד (אלפי Bytes):

**חבילות קטנות (0–500 Bytes)**: נפוצות בשיחות בזמן אמיתי, בהן המהירות קריטית.

**חבילות בינוניות (500–2000 Bytes)**: אופייניות לשירותים שדורשים שילוב של מהירות ואמינות.

**חבילות גדולות (2000–10,000 Bytes ומעלה)**: משמשות להורדת תוכן כבד כמו סרטונים או קבצים גדולים.

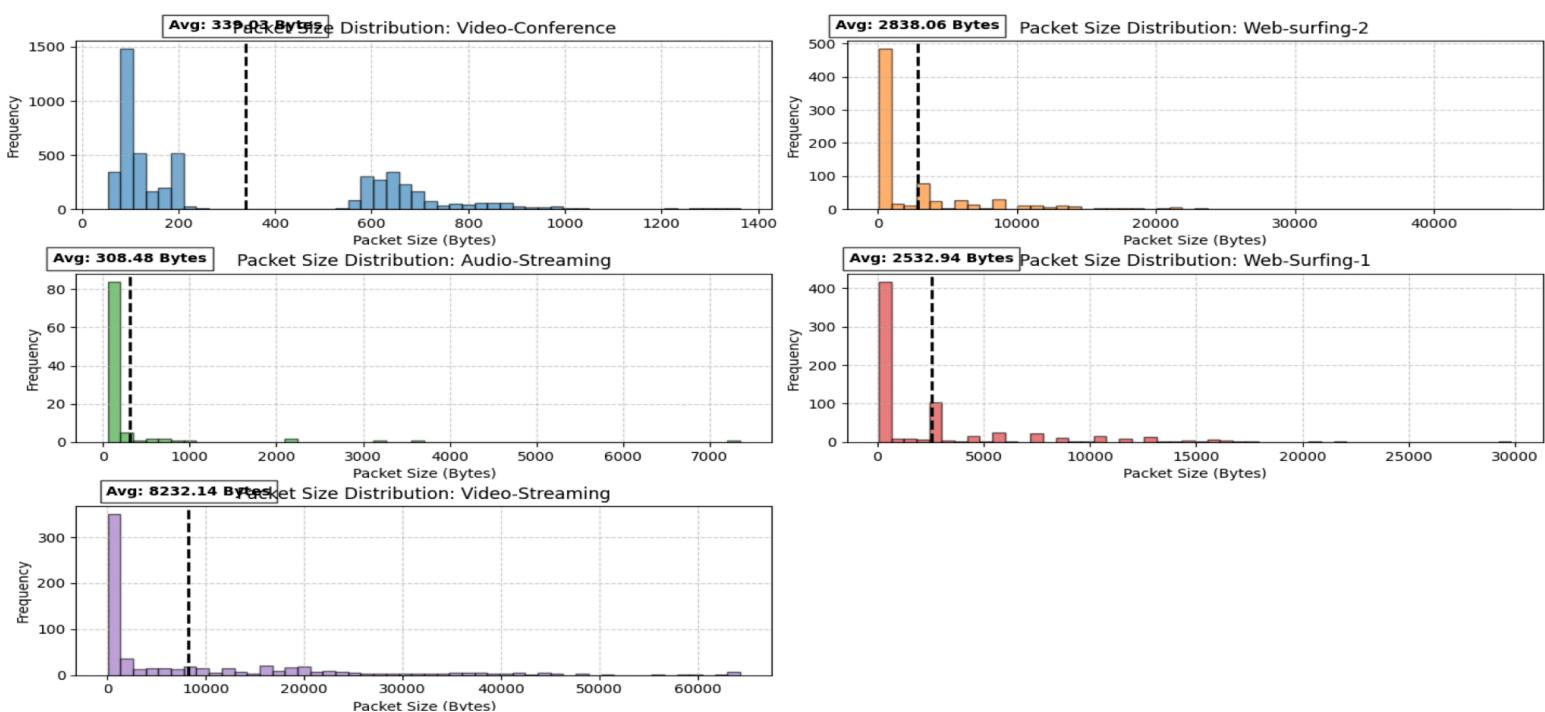
### ציר ה-Y: תדרות הופעת החבילה (Frequency):

מייצג את מספר הפעמים שבהם חבילה בגודל מסוים הופעה במהלך הקלטה.

כל שהעודה גבוהה יותר – כך החבילה בגודל זה הייתה נפוצה יותר.

תדרות גבוהות של חבילות קטנות מעידה על צורך מינימלי (Latency), בעוד תדרות גבוהות של חבילות גדולות מצביעה על אופטימיזציה של רוחב פס להעברת כמויות מידע גדולות בפרק זמן קצר.

### D: Packet Size Distribution by App



## תיאור הגרפים והמסקנות:

### הסבר גרפ A:

1. **שיחות וידאו (Video Conference)** מtabססota בעיקר על **UDP** ובנוסף **DTLS** מופיע באופן משמעוני

**Zoom** משתמש בפרוטוקול UDP לניהול השיחה, אך בנוסף ניתן לראות שימוש משמעוני בפרוטוקול DTLS, מה מעיד על הצפנה חזקה ברמת התüberה. הופעת **DTLS** מעידה על מגנון אבטחה מוגבר, שבו הנתונים המועברים בזמן אמת מוצפנים, מה שמוסיף שכבת הגנה על המידע המשודר.

### 2. גישה באינטרנט מתבצעת באמצעות **TLS 2/2/HTTP**

**Firefox** ו-**Chrome** משתמשים בפרוטוקולים TLS 2/2/HTTP, המעידים על הצפנה מתקדמת ושימוש בטכנולוגיות מודרניות להעברת נתונים מאובטחת ויעילה יותר. **Firefox** מציג תעבורת גבולה יותר בפרוטוקול TLS בהשוואה ל-**Chrome**, מה שעשוי להעיד על מגנון הצפנה שונים בין הדפדפנים. **HTTP 2** מופיע בתüberה של הדפדפנים, דבר שמצוין על אופטימיזציה בביצועי טעינת האתרים ושיפור מהירות ההעברה.

### 3. וידאו סטרימינג (**YouTube**) מתבצע באמצעות **TLS 2/2/HTTP**

**TLS** משמש להצפת התקשרות, בעוד **HTTP 2** אחראי לניהול הזרמת הווידאו, דבר המעיד על שיפור ביצועים ייועל תעבורת הנתונים. **UDP** לא מופיע בתüberה YouTube, מה שמצוין על קר שהזרמה מבוססת על **TLS + TCP** ללא שימוש ב-**QUIC**. (ביטלנו אותו לצורך הניסוי)

**אודיו סטרימינג (Spotify)** מתבצע על גבי **TLS** ולא שימוש ב-**UDP**. ספוטיפי משתמש ב-**TLS + TCP**, מה שמעיד על הצפנה מלאה של תעבורת השמע. העדר **UDP** מעיד על קר שהזרמת האודיו מבוצעת בתקשר של תקשורת מבוססת חיבור (TCP), ככל הנראה בשל מגנון שליטה טובים יותר באיכות ההשמעה. (וגם כי ביטלנו את QUIC).

### 4. **שיחות וידאו (Zoom)** עושות שימוש בפרוטוקול DTLS, המעיד על הצפנה חזקה יותר בתקשר של השמע והוידאו

**DTLS** נוכח בתüberה, דבר המצביע על שימוש במגנון הצפנה חזק המגן על תקשורת בזמן אמת. בנוסף, Zoom עושה שימוש משמעוני ב-**UDP**, מה שמאפשר לו לספק תקשורת וידאו ושמע עם השהייה מינימלית.

**5. STUN נמצא בשימוש במגנון NAT traversal**

**STUN** מופיע בתüberה, ככל הנראה לצורך מעבר דרך נתבים וחומות אש. הופעת **STUN** בהקשר של Zoom ושירותים נוספים מראה שהפרוטוקול מס' 5 בהקמת חיבורים ישירים בין משתמשים.

### 6. כמות קטנה של תעבורת **DNS**, מכיוון שלא הייתה חשיבות לבקשת **DNS** רבות בתüberה הנמدادת

ניתן לראות כי ישן מעט מאוד בבקשת DNS, דבר שמצוין על קר שרוב התקשרות בתüberה שנבדקה לא דרש ריבוי שאילתות **DNS**.

מרבית התקשרות היא הזרמת נתונים (Streaming), גישה ושיחות ידאו, וכך אין צורך בבקשת **DNS** רבות, מאחר והחיבורים מtbodyים ישירות לשרתים הרלוונטיים לאחר שהדף או היחסום פתר את כתובת ה-IP בתחילת התהילן.

## **הסבר גרפ' B:**

1. **שיחות וידאו (Video Conference) משתמשת ב포רטים גבוהים (49152 ומעלה) ובפורטים "יעודיים אחרים"**

Zoom משתמש בעיקר בפורטים 52198 ו-50926, שהם פורטים זמינים. זה מראה שהאפליקציהutzia משתמשת בפורטים דינמיים לחיבור TCP, חלק ממנגנון ה-NAT-Traversal. כמו כן, ניתן לראות שימוש נמוך בפורטים 36588 ו-36606, מה שעשוי להיות קשור לממשק הרשת של Zoom. (לאחר בדיקה בגוגל וחיפוש)

### **2. הזרמת וידאו (YouTube) משתמשת בפורטים הקלאסיים של HTTP ו-HTTPS**

ניתן לראות שימוש ממשוני בפורטים 443 ו-40844, מה שמצוין על שימוש בפרוטוקול HTTPS. פорт 443 הוא הסטנדרטי עבור TLS, ולכן השימוש הנרחב בו הגיוני. בנוסף, ניתן לראות שימוש בפורט 54820, שהוא פорт זמני שמקורו על ידי מערכת הפעלה עבור חיבור TCP. (גם לאחר חיפוש על הפורט)

### **3. גלישה באינטרנט (Web-Surfing) מציגה שימוש דומה בין Chrome ו-Firefox, עם הבדל קילובייטים המשנים**

גם Chrome ו-Firefox משתמשים באופן ממשוני בפורט 443, שכן רוב האתרי האינטרנט כיוון משתמשים HTTPS. עם זאת, Firefox נוטה להשתמש גם בפורט 47500, בעוד ש-Chrome משתמש יותר בפורט 44326. ההבדל הזה נובע ממדיניות ההקצאה של פורטים זמינים בדפדפניים או מהשירותים אליהם התחברים.

### **4. הזרמת אודיו (Spotify) משתמשת בעיקר בפורטים של HTTPS ופורטים זמינים**

ניתן לראות שימוש בפורטים 443, 44126 ו-34816. פорт 443 מעיד על שימוש בפרוטוקול TLS עבור אבטחת הזרמת הנתונים. בנוסף, השימוש בפורטים זמינים מראה לנו ש-Spotify מקצה חיבורים דינמיים להזרמה ולא משתמשת בפורטים קבועים.

### **5. נראה שימוש ב-STUN באמצעות הפורטים 3478, 5349 או 44326**

פורטים אלו משמשים לרוחב לפרוטוקול STUN, שמטרתו היא לאפשר חיבור תקשורת בין ללקוחות מאחור NAT. השימוש בפורטים זמינים נוספים כגון 52198, 50926, 50925 ו-56054 עשוי להעיד על כך שהאפליקציהutzia מבעלות חיבור P2P מסוימים.

## **הסבר גרפ' C:**

1. **שיחות וידאו (Video Conference) מציגות זמני הגעה קצרים מאוד של חבילות, מה שמעיד על שימוש ב프וטוקול מבוסס UDP.**

ניתן לראות שבשילוב וידאו (Zoom) زمنי ההגעה בין החבילות הם הקצרים ביותר (כ-21 מילישניות בממוצע).

הדבר מצביע על כך שהחbillות נשלחות באופן כמעט מיידי זו אחר זו, מה שמאפיין תקשורת אינטראקטיבית הדורשת השהיות נמוכות. השימוש ב-UDP מאפשר העברת חbillות מהירה ללא הצורך אישור קבלה, וכן הפרוטוקול מתאים במיוחד לשיחות וידאו.

2. **הזרמת וידאו (YouTube)** מציגה זמני הגעה ממוצעים נמוכים יחסית, אך ארכויים יותר מאשר בשיחות וידאו.

זמן הגעה של חbillות בהזרמת וידאו (~48.15 מילישניות בממוצע) גבוהים מallow של שיחות וידאו אך עדין נמוכים יחסית. הדבר מעיד על כך שהוידאו מוזרם בחbillות בצורה שוטפת אך עם מעט יותר השהיות בין החbillות, מה שעשוי להעיד על מנגנון אגירה (buffering) בתקשות. מכיוון שההעבורה אינה חיבת להגיע בזמן אמת כמו בשיחות וידאו, ניתן שימוש ב-CPTCP לצד UDP כדי להבטיח איזוטר טובה יותר.

3. **גלישה באינטרנט (Web-Surfing)** מציגה שני כלשים בין הדפדףים, כאשר Firefox מציג זמני הגעה קצרים יותר מאשר Chrome. בזמן הגעה הממוצעים בין החbillות של פיירפוקס (9.71 ms) ניתן לראות כי הדפדף משדר בקשות ווחbillות בצורה עלייה יותר יחסית. לעומת זאת, Chrome מציג זמני הגעה ממוצעים מעט גבוהים יותר (14.78 ms), מה שעשו להעיד על מנגנוני ניהול חיבור שונים או שיטת אופטימיזציה שונה בשכבות הпрוטוקולים של הדפדף. ניתן להניח שההבדלים בין הדפדףים נובעים מהתנהגות ניהול הקשרים ומהשימוש בפרוטוקולים שונים. (או גם תוספים שונים שקיימים בדפדף השונים).

4. **הזרמת אודיו (Spotify)** מציגה זמני הגעה גבוהים באופן משמעותי מאשר לכל שאר היישומים, מה שמעיד על אופי תעבורת TCP. בזמן הגעה הממוצעים בין החbillות ב-Spotify עומדים על 113.81 מילישניות, גבוהים בהרבה מכל שאר האפליקציות. הדבר מצביע על כך שההעבורה מושפעת מהתבססת בעיקר על TCP, שבו יש צורך באישורים חוזרים על כל חבילה שנשלחת. התנהגות זו הגיונית בהזרמת אודיו, מכיוון שהיא שומרת זמן איסוף נתונים (buffering) ואינו דורש תעבורת מיידית כמו בשיחות וידאו. ניתן לראות שהפרש זמני הגעה מציבים על כך שספוטיפי' בהשוואה להזרמת וידאו (YouTube), פחות רגיש לאיבוד חbillות וכן לא מחזיב להשתמש ב-UDP.

5. **באופן כללי, ניתן לראות כי אפליקציות מבוססות UDP מציגות זמני הגעה נמוכים יותר, בעוד אפליקציות מבוססות TCP מציגות זמני הגעה גבוהים יותר.** השוואה בין Zoom (שימוש נרחב ב-UDP) לבין Spotify (שימוש ב-CPTCP) מדגימה את ההבדל המרכדי בין פרוטוקולי תקשורת בזמן אמת לעומת תקשורת מבוססת אמינות. ניתן להסביר כי ישומים הדורשים אינטראקטיביות גבוהה יעדיפו להשתמש בפרוטוקול UDP על פני TCP כדי למנוע השהיות.

#### D.3.3. הסבר גרף D:

1. **שיחות וידאו (Video Conference)** מציגות חbillות קטנות באופן יחסית, מה שמעיד על שימוש בפרוטוקול UDP.

ניתן לראות כי החbillות הנפוצות ביותר בשיחות וידאו (Zoom) הן בגודל ממוצע של 339.03 בתים.

השימוש בחבילות קטנות נובע מכך ששיחות ידאו מחייבות תעבורת רציפה ומהירה, ולכן עדיף לשולח חבילות קטנות ובתדריות גבוהה כדי להימנע מההיאות. הדבר תואם את העבודה שהשירות משתמש ב프וטוקול UDP, שמאפשר שליחת רציפה של מידע ללא צורך בהמתנה לאישור קבלת.

2. **בザרמת וידאו (YouTube) נקבל חבילות גדולות יותר, מה שמעיד על השימוש ב-TCP ובפרוטוקולי הצפנה כגון TLS.**

החבילות הנפוצות ביותר בהזרמת וידאו הן בגודל ממוצע של **14.8232 بتים**, מה שמרמז על שליחה אופטימלית של נתונים וידאו דחוסים בפרוטוקול אמין. הזרמת וידאו אינה דורשת זמן אמת כמו שיחות וידאו, ולכן ניתן לשולח חבילות גדולות יותר כדי לנצל בצורה מיטבית את רוחב הפס. עובדה זו מעידה על שימוש ב-TCP עם TLS להבטחת אמינות העברת והצפת הנתונים.

3. **גישה באינטרנט (Web-Surfing) מציגה הבדלים בין דפדפניים, כאשר Firefox משתמש בחבילות מעט גדולות יותר מאשר Chrome.**

בגישה לאתר מקו, החבילות הממוצעות ב-Firefox הן בגודל **2838.06 بتים**, בעוד שב-Chrome החבילות הממוצעות קטנות יותר ועומדות על **2532.94 بتים**. מה שיכל לנבוע מהבדלים בניהול החיבורים של הדפדפניים / הבדלים בכיווץ תעבורת / מנגנון טעינה מוקדמת.. ניתן לראות כי בשני הדפדণים, גודל החבילות הממוצע קטן מהותית מזה של הזרמת וידאו, מה שמעיד על כך שתעבורת S/HTTP מורכבת מבקשות קצרות ותשבות מותאמות.

4. **הזרמת אודיו (Spotify) מציגה חבילות קטנות יחסית, אך גדולות יותר מאשר שיחות וידאו.**

החבילות הנפוצות ביותר בהזרמת אודיו עומדות על ממוצע של **308.48 بتים** לחבילה, גבוות מעלו של שיחות וידאו, אך קטנות משמעותית מעלו של הזרמת וידאו (Youtube). הדבר נובע מכך שהזרמת אודיו משתמשים ב-TCP עם מנגנון אגירה (buffering), המאפשרים טעינה מוקדמת של תוכן כדי למנוע השהיות בשידור. בגיןוד לשיחות וידאו, ההזרמה אינה דורשת עדכונים מיידיים ולכן ניתן להעביר את המידע בחבילות מעט גדולות יותר.

5. **באופן כללי, ניתן לראות כי אפליקציות מבוססות UDP משתמשות בחבילות קטנות יותר, בעוד אפליקציות מבוססות TCP משתמשות בחבילות גדולות יותר.**

ההשוואה בין Zoom (שימוש ב-UDP) לבין YouTube (שימוש ב-TCP) ממחישה את השוני בין פרוטוקולי התקשרות. שיחות וידאו דורשות שליחת חבילות קטנות בתדריות גבוהה, בעוד שהזרמות מדיה (אודיו/וידאו) משתמשות על חבילות גדולות יותר כדי לנצל את רוחב הפס בצורה מיטבית. עובדה זו מחזקת את ההנחה כי ישומים אינטראקטיביים יעדיפו UDP, בעוד שישומים המבוססים על הזרמה אמינה יעדיפו TCP.

### 4.3. סימולציה של תקיפה לזיהוי אפליקציות על בסיס תעבורה מוצפנת

הקדמה והסביר:

בסעיף זהה מבקשים מאיתנו לשחק תפקיד של תוקף שמנסה לזהות באילו אפליקציות או אתרים המשמש ביך, גם אם התעבורה מוצפנת. יש שתי אפשרויות שאלהן נחשף התוקף:

1. **אפשרות א:** התוקף יודע את גודל הפקודות, חותמת הזמן שלהן, ואתה **hash** של ה-4-tuple (כתובת IP מקור, כתובת IP יעד, פורט מקור, פורט יעד):

במקרה זה, התוקף יכול **לקבץ** את התעבורה **לזרמים** (Flows) לפי שילוב כתובת ה-IP והפורט.

אם התוקף מזהה תבנית גודל חבילות מסוימת או דפוס של תדיירות זמינים קבועה שנייה לאפיין בין חבילות – ניתן להסיק **איזו אפליקציה או שירות נמצאים בשימוש**.

לדוגמה:

**Zoom** ישלח הרבה פקודות קטנות בתדיירות גבוהה (שיעור וידאו).

**YouTube** ישלח פקודות גדולות בהפרש זמן קבועים (אפרינג של וידאו).

**Spotify** ישלח פקודות בגודל בינוני עם הפרשי זמן שונים. (גם מבצע אפרינג אך פחות משמעותי)

2. **אפשרות ב:** התוקף יודע רק את גודל הפקודות ואת חותמות הזמן שלהן:

כאן, התוקף לא יכול **לקבץ** זרים בקלות כי אין לו מידע על ה-IP והפורט.

עם זאת, ניתן לנסות לבצע  **ניתוח סטטיסטי** על דפוסי הגודל והזמן ולזהות אם הם מתאימים לאפליקציה מסוימת.

לדוגמה, תוקף יכול לגלוות **שימוש ב-YouTube** אם הוא רואה חבילות גדולות כל כמה שניות – גם בלי לדעת לאן הן נשלחות.

לעומת זאת, **שימוש ב-Zoom** יופיע כחבילות קטנות עם תדיירות גבוהה, גם אם התוקף לא יודע מה היעד.

**נסביר איך כל קרייטריון יתרום לתוקף בעת מסוימת לזהות את התעבורה:**

### 1. חותמת זמן (Timestamp)

חותמת זמן היא הרגע שבו הפקטה נשלחה או התקבלה, והוא משמש כדי להבין את סדר הפקטות ואות המרוחקים ביניהם.  
לדוגמה:

Timestamp : 1711782923.451

ניתן לחשב מרוחקים בין פקודות עוקבות כדי להבין תדריות התקשרות של אפליקציה מסוימת.

**איך חותמת זמן עוזרת לתוקף?**

אם פקודות נשלחות מרוחקים קבועים (כמו 0.3 שניות), זה עשוי להעיד על הזרמת ידאו (buffering).  
אם יש פרצ'י חבילות (bursts) בזמן קצר מאוד, ניתן שמדובר בשיחת ידאו (Zoom).  
חבילות בודדות עם הפרשים גדולים יכולות להיות טעינה אתר בדף (כמו שכבר רأינו)

### Tuple-4 של Hash .2

ה-4-Tuple (רביעייה) מתאר מאפיינים ייחודיים לכל זרם תקשורת (Flow) והוא מורכב מ: Source IP , Destination IP, Source Port , Destination Port.

לדוגמה, אם משתמש פותח חיבור HTTPS ל-Youtube, ניתן שנראה שהוא צזה:

(Source IP: 192.168.1.100, Destination IP: 142.250.187.46, Source Port: 52000, Destination Port: 443)

כדי לטשטש את הכתובות, ניתן להמיר את הריביעייה ל-hash כך שהותוקף לא יידע את הכתובות בפועל, אך עדין יוכל לזהות פקודות שישיכות לאותו חיבור.

לדוגמה:

hash(192.168.1.100, 142.250.187.46, 52000, 443) → "A7F9D8C3B2"

כל הפקטות שימושות באותו חיבור יקבלו את אותו hash, וכך התוקף יוכל לזהות זרמים (Flows) מבלי לדעת בפועל את כתובות ה-IP.

**איך ה-Hash של Tuple-4 עוזר לתוקף?**

הוא מזהה חיבורים שונים גם אם ה-IP מוצפן או מוסתר.

הוא מבידיל בין תקשורת מקבילה, למשל:

אחד לשיחת Zoom

אחד להזרמת מוזיקה בסופוטיפי

אחד לגלישה ב-Chrome

הוא מאפשר ניתוח של זרמים שונים (Flows) ולראות אילו פקודות שייכות אותה פעילות.

### **:Flow ID .3**

נתחיל בלהסביר מה הכוונה ב Flow:

#### **Flow (זרם תקשורת):**

כל חיבור נפרד בין שני מכשירים נחשב Flow.

**לדוגמה**, אם אתה צופה ב-YouTube בזמן שאתה משתמש ב-Zoom, אז יש לפחות שני Flows שונים: Zoom: תקשורת אינטרנטית עם הרבה חבילות UDP קטנות.

YouTube: הורדת חבילות גדולות מהשרת של YouTube, את זה ניתן לעשות בגלל השימוש ב Buffering.

### **:Flow ID**

ה-ID Flow הוא מזהה ייחודי לכל זרם תקשורת והוא לרוב מחושב על בסיס ה-4-Tuple:

$\text{Flow ID} = \text{hash}(\text{Source IP}, \text{Destination IP}, \text{Source Port}, \text{Destination Port})$

כך שאמם יש לדוגמה חיבור למשתמש עם YouTube:

142.250.187.46 :443 → 192.168.1.10

ה-ID Flow בעברו יהיה שונה מה-ID Flow של שיחת Zoom שמתרכשת באותו זמן.

#### **איך Flows עוזרים לתוקף?**

אם התוקף מזהה מספר רב של Flows קצרים מאוד (Zoom משתמש בהרבה חיבורים קצרים טוווח), הוא יכול לשער שמדובר בשיחת וידאו.

אם יש Flow אחד עיקרי עם חבילות גדולות, יתכן שמדובר בהזרמת וידאו (YouTube, Netflix).

אם Flows נמשכים זמן רב אך עם תעבורת מעטה, מדובר כנראה בגלישה באינטרנט (Web-Surfing).

כל זה כמובן מבוסס על מה שהראנו בשאלת 3 בגרפים.

(\*על גודל הפקות לא צריך להסביר לכך נסביר רק איך הוא תורם לתוקף)

### **4. גודל הפקות איך גודל הפקות עוזר לתוקף?**

הגודל של הפקות מספק מידע חשוב, גם אם התוכן מוצפן. מהמסקנות שהסקנו בחילוק הקודם נובע:

#### **1. פרוטוקולים שונים שלחים פקטות בגדים אופייניים**

פקחות **VoIP (שיחות קוליות)** נוטות להיות קטנות (100-200 בתים).

פקחות **הזרמת וידאו** נוטות להיות גדולות (1000-1500 בתים) כדי לשדר מידע רב.

תעבורה Web מכילה פקודות קצרות (בREQUEST HTTP) ופקודות גדולות (תשובות עם תוכן האתר).

## 2. תכניות תעבורה לפי סוג היישום

גילשה באתר חדש תראה גודל פקודות מגוון מאוד.

יוטיוב או נטפליקס יראו גודל פקודות גדול באופן אחד.

שירותי הודעות מיידיות (WhatsApp, Telegram) ישלו פקודות קצרות עם מרוחקי זמן גדולים ביןיהן.

## 3. ניתוח יחס בקשות-תשובות

בקשות S/HTTP ישנה בקשה קצרה מאוד (Client Request) ותשובה גדולה יחסית (Response).

בפרוטוקולים כמו VoIP, גודל הפקודות נותר קבוע וקטן.

### מדוע זה שימושי לתוקף?

תיקוף יכול להשתמש באותו ניתוח כדי לזהות חתימה סטטיסטית של כל אפליקציה.

גם אם התעבורה מוצפנת, גודל הפקודות וזמן מספקים מודיע חשוב לזהות השירותים בהם המשתמש משתמש.

ניתן לשפר את ההתקפה עם Machine Learning מתמקד כדי לזהות אפליקציות בדיק גובה.

**פיתוח אלגוריתם לפי תכונות באפשרות א' - סיווג אפליקציות בהתבסס על גודל חבילות,  
חוותמאות זמן וזיהוי זרמים:**

דרישות וחבילות שלל הבודק להתקין בשבייל שיוכל להשתמש בקוד בצורה תקינה :

Pandas  
Os  
Numpy  
MatPlotLib  
Seaborn  
Sklearn (scikit-learn)  
Ae-tools

## תיאור האלגוריתם:

### 1. טיענת הנתונים מקובצי ה-CSV

הקוד טוען את נתוני התעבורה מקובצי ה-CSV שנמצאים בתיקייה **csv-files**.  
לכל קובץ, אנו שולפים:

- חותמות זמן (Timestamp) – לтиיעוד מתי נשלחה כל חבילה.
- גודל הפקחות (Packet Size) – כדי להבין את נפח התעבורה.
- מזהה זרם (Flow ID) – ליהוי הזרם אליו שייכת כל חבילה.

הנתונים עוברים ניקוי ראשוני, הכלול:

- הרמת נתונים זמן לערכים מספוריים.
- הסרת נתונים חסרים כדי להבטיח תקינות.

### 2. סיווג תעבורה לפי דפוסים

בשלב זה נזהה התבניות בתעבורה, שייעזרו לנו לשיך זרמים לסוגי אפליקציות:

- מרוחחים בין פקחות קבועים → שימוש VoIP.  
כasher מרוחח הזמן בין חבילות עקבים ונמכרים, זה עשוי להעיד על תעבורה של שיחות וידאו (Video Calls).
- גודל פקחות אחד וגדול → הזרמת וידאו (Streaming).  
סטרימינג מתאפיין בחבילות יחסית, עם שונות נמוכה בגודל החבילות.
- שילוב של בקשות קטנות ותשבות גדולות → גלישה באינטרנט (Web Browsing).  
גלישה בדפדפן כוללת בקשות קטנות (GET, POST) עם-tagות גדולות יותר שמכילות תוכן.

### 3. חישוב מאפיינים סטטיסטיים לכל זרם

לאחר עיבוד הנתונים, מחשבים סט מאפיינים לכל זרם תעבורה, הכלול:  
**גודל חבילה ממוצע וסטיית תקן**

מאפשר זיהוי אפליקציות עם תנועה קבועה (כגון שיחות קוויות, שבחן הפקחות בגודל אחיד). גלישה באינטרנט מציגה שונות גובהה בגודל החבילות עקב שילוב של בקשות קטנות ותשבות גדולות.

#### 3.2. מרוחך זמן ממוצע וסטיית תקן בין חבילות

שיעור VoIP וסטרימינג אודיו יציגו מרוחחים קבועים וצפופים.  
galisha באינטרנט וישוםים אחרים יציגו שונות גדולה יותר בזמן ההגעה בין חבילות.

#### 3.3. מדד "התפרצויות" התעבורה (Burstiness Measure)

מוגדר כיחס בין סטיית התקן של זמן ההגעה לבין המרוחך שלהם.  
תעבורה יציבה כמו סטרימינג תציג ערך נמוך, ואילו גלישה או תעבורה אקראית תציג ערכים גבוהים יותר.

#### 3.4. גודל חבילה מקסימלי

סטרימינג של וידאו ואודיו נטה לכלול חבילות גדולות מאוד.  
galisha ושירותי תקשורת (כגון צ'אטים) מכילים חבילות קטנות יותר.

#### 3.5. מספר הפקחות הכלול בזמן קבוע

זרמים אינטנסיביים כמו סטרימינג מכילים הרבה חבילות בפרק זמן נתון.  
galisha באינטרנט כוללת פחות חבילות במשך זמן יחסית.

#### **4. שימוש ב-K-Means Clustering ליזיהו אפליקציות**

לאחר חישוב כל המאפיינים, אנו מפעילים אלגוריתם K-Means Clustering על הנתונים. המטרה: לבדוק אם ניתן לשיר זרמי תעבורת לאפליקציות שונות על סמך דמיון סטטיסטי.

1. מחלקים את הנתונים לקבוצות - Clusters
2. מציגים את התוצאות על ציר דו-ממדי, מוקדדות בצבעים שונים

## תוצאות:

כasher nnshe libzcu at hsiogg ul chmsht hchbilot shakltno basuf' a' caser hn mclot at cl hmidu, npfta otun mhatikia .csv\_files

laachr mcn b'amzutot algoritm shizcrno wlaachr shaimno at model shizcrno nkbl at htzotot bavot:

==== CLASSIFICATION REPORT ====				
	precision	recall	f1-score	support
Audio Streaming	0.90	1.00	0.95	9
Video Calls	1.00	1.00	1.00	12
Video Streaming	1.00	0.50	0.67	2
Web Browsing	0.00	0.00	0.00	1
accuracy			0.92	24
Accuracy:	0.92			

bdoh hsiogg mzgutot tzotot ubor arbu katgoriot yikriot (Audio Streaming, Video Calls, Video Streaming, Web Browsing, Streaming).

- Precision (דיק): mturk clil dgimot shsogg lo katgoria zo, cmha b'amt shiyicot alia b'po'ul.
- Recall (rgishot): mturk clil dgimot shbamt shiyicot lkatgoria, cmha zoho ncuna.
- Support: msfur dgimot b'po'ul b'cll katgoria.
- F1 : zua mmutz hrmoni shma'zn bn d'ik (Precision) lbvn zihoi ncun (Recall), wmspk mdd ychid hmskaf at b'izui hsiogg.

bsof hdoh mzui hsiogg (d'ik clili) shl 0.92, cilomr 92% mhdgimot sogg ncuna b'cll katgoriot. ureci hsiogg (Recall) bgohim myidim ul model shiodu gm l'mnuu siogg-iyter (Precision) bgoh) wgm lzahot at mrabit dgimot hrloontiot (Recall).

clomr, hzlnu lsogg at cl hmidu lfi sog tutbora shl caser anchnu yoduyim at cl kritriyonim hnndrim wcaser ain pfrutot tutbora brku cmo shasuf b'iksh ma'itnu.

## סקירה כללית

בתרחיש זה אנו מודמים מתקפה שבה לתקוף יש גישה מוגבלת מאוד לנוטרי התעבורה. התקוף ידע רק שני פרמטרים על כל חבילת:

1. גודל החבילות (Packet Size)
2. חותמת זמן (Timestamp)

המטרה היא לבדוק עד כמה תוקף יכול לזהות אילו אפליקציות או אתרים המשמש מבקר, גם כאשר התעבורה מוצפנת או אNONYMITY.

## מגבילות התקוף

מכיוון שהתקוף ידע רק את גודל החבילות ואת חותמת הזמן, יש מידע רב שהוא אינו יכול לגשת אליו:

1. אין לו מידע על כתובות ה-IP של המקור והיעד.
2. אין לו מידע על מזהה זרם (Flow ID) המבוסס על ה-4-Tuple (source IP, dest IP, source port, dest port).
3. אין לו גישה לתוכן החבילות, שכן התעבורה מוצפנת.

עם זאת, התקוף יכול להשתמש בדפוסים מסוימים בגודל החבילות ובמרווחי הזמן ביןיה כדי לנחש את סוג האפליקציה.

## אופן הביצוע

בסעיף ב ניצור אלגוריתם דומה לסעיף א שמנתח את אותה התעבורה שניתנו בסעיף א, אך, הוא משתמש הפעם בתעבורה הלא מוצפנת ומונתח אותה כאשר לא נעשה שימוש תעבורה המפוענחת, אלא רק בתעבורה המוצפנת, ובנוסף יש לו גישה רק לגודל החבילות וחותמת הזמן של כל חבילה, ניצור בדיקת אוטם קבצי CSV אך, ללא שימוש בחבילות המפוענחות ולא העמודות המתאימות (נזכר דוגמא להבדל).

להלןקובץ CSV המתאר את הפעולות לאחר פתיחת ההצפנה):

No.	A	B	C	D	E	F	G
	No.	Time	Source	Destination	Protocol	Length	Info
1		0	192.168.12.192.168.12DNS			74	Standard query 0x6bcb A www.mako.co.il
2	0.000604	192.168.12.192.168.12DNS				74	Standard query 0x85c1 HTTPS www.mako.co.il
3	0.005976	192.168.12.192.168.12DNS				211	Standard query response 0x85c1 HTTPS www.mako.co.il CNAME wilcard.mako.co.il.edgekey.net CNAME e974.b.akamaiedge.net SOA n0b.akamaiedge.net
4	0.006312	192.168.12.192.168.12DNS				166	Standard query response 0x6bcb A www.mako.co.il CNAME wilcard.mako.co.il.edgekey.net CNAME e974.b.akamaiedge.net A 23.56.225.242
5	0.012359	192.168.1223.56.225.TCP				74	44326 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSeq=3058958283 TSecr=0 WS=128
6	0.01295	192.168.1223.56.225.TCP				74	44342 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSeq=3058958283 TSecr=0 WS=128
7	0.059978	23.56.225.192.168.12TCP				60	443 > 44342 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
8	0.059978	23.56.225.192.168.12TCP				60	443 > 44326 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
9	0.060146	192.168.1223.56.225.TCP				54	44342 > 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
10	0.060275	192.168.1223.56.225.TCP				54	44326 > 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
11	0.06116	192.168.1223.56.225.TLSv1.3				2081	Client Hello (SNI=www.mako.co.il)
12	0.061508	23.56.225.192.168.12TCP				60	443 > 44342 [ACK] Seq=1 Ack=1461 Win=64240 Len=0
13	0.061508	23.56.225.192.168.12TCP				60	443 > 44342 [ACK] Seq=1 Ack=2028 Win=64240 Len=0
14	0.061817	192.168.1223.56.225.TLSv1.3				2081	Client Hello (SNI=www.mako.co.il)
15	0.06202	23.56.225.192.168.12TCP				60	443 > 44326 [ACK] Seq=1 Ack=1461 Win=64240 Len=0

מנגד קובץ ה-.csv שמתאר את 1-web-browsing בסעיף ב' (שמכיל תעבורת מוצפנת בלבד):

A	B	C
No.	Time	Length
1	0	74
2	0.000604	74
3	0.005976	211
4	0.006312	166
5	0.012359	74
6	0.01295	74
7	0.059978	60
8	0.059978	60
9	0.060146	54
10	0.060275	54
11	0.06116	2081
12	0.061508	60
13	0.061508	60
14	0.061817	2081
15	0.06202	60

## תוצאות:

==== CLASSIFICATION REPORT (LIMITED DATA) ====			
Class	Precision	Recall	Support
-----			
Audio Streaming	1.00	0.00	1
Video Calls	1.00	0.00	1
Video Streaming	1.00	0.00	1
Web Browsing	0.40	1.00	2
 Accuracy: 0.40			

בדוח הנוכחי ניתן לראות ארבע קטגוריות (Audio Streaming, Video Calls, Video Streaming, Web Browsing), כאשר מספר הדגימות (Support) בכל קטgorיה נמוך מאוד: קטגוריות מסוימות מופיעות פעמיות אחת בלבד, ו-בודדים Web Browsing פעמיים. כתוצאה לכך, דיקן המדדים (Precision, Recall) מושפע מאוד מכל טעות בודדת, ומתקבלות תוצאות כמו  $Recall=0$  בקטגוריות עם דגימה בודדת שלא זוהתה נכון. כך, למרות שבחלק מהקטגוריות המدد Precision הוא 1.00 (כ倘igma אחת סוגה נכון), במקרים נוספים מצלח לנבא נכון רק כ-40% מהדגימות הכוללות, ניתן להסיק שהיעדר מידע נוסף (כמו כתובות מקור/יעד או פרוטוקול) מקשה על המודל להבדיל בין סוג התעborah, במיוחד כאשר יש מעט דגימות בכל קטgorיה.

## **מסקנה מהתוצאות:**

במודל הקיים (شنער ב-ID Flow ובארבעה שדות: כתובות IP מקור/יעד ופורטים), כל CSV יכול היה להכיל מספר "זרמים" (Flows) שונים. בכל פעם שהקוד איתר הבדל ב-4 Tuple (מקור, פорт מקור, יעד, פортיעד) או בפרוטוקול, נוצר Flow חדש. כך, גם אם עבדנו על אותם קבצי CSV, הקוד "פיזל" את המידע למספר דוגימות רב יותר — אחת לכל זרם.

לעומת זאת, בקוד ש商量סס רק על שני עמדות (Time, Length), אנו מתייחסים לכל CSV כאלו "דגם אחד" יחידה, ללא חלוקה פנימית לזרמים נפרדים. כתוצאה לכך, מספר הדוגימות הכלול קטן יותר (ולכן גם ערכי Support בכל קטגוריה נמוכים יותר). לעומת זאת, במודל הקיים "נוצרו" יותר דוגימות מפני שכל CSV פועל למספר Flows; במודל הנוכחי (רק שני שדות), אין מנגנון פיצול, ולכן כל CSV הוא דגם אחד.

## **מסקנות לגבי מידע שהtopic יכול לפענן:**

.1.1. **גודל החבילה:** פרוטוקולים ושירותים שונים משתמשים בתבניות גודל' חבילות אופייניות.  
לדוגמה:

.1.1.1. **שירותי סטרימינג כמו YouTube או Netflix** גוטים לשילוח חבילות גדולות באופן עיקבי.

.1.1.2. **שירות VoIP כמו Zoom או Skype** משתמשות בחבילות קטנות מאוד הנשלחות בקצב קבוע.

.1.2. **שירותי גישה באינטרנט** יכולים לשילוב של חבילות קטנות (בקשות S/HTTP) וחבילות גדולות יותר (תשיבות מהשרת).

.1.3. **סטטיסטיקות מתקדמות** (Flow Analysis, Burstiness)

מודדים מתקדמים כמו Flow Entropy לאפשר אילו שירותים פועלים במערכת  
שלר.

### **דוגמאות:**

.1. **גישה באינטרנט** → דפו לא סדר עם חבילות גדולות וקטנות.

.2. **משחקים מקוונים** → זרימה קבועה עם מעט נתונים.

.3. **הברת קבצים (BitTorrent, Google Drive)** → חבילות גדולות מאוד  
שנשלחות בקצב יציב.

.1.4. **חותמות זמן (Timestamps):** ניתוח זמני ההגעה של החבילות מאפשר לזהות דפוסים אופייניים:

.1.4.1. **וידאו סטרימינג** ישלח חבילות בגודלים אחידים בהפרש זמן יציבים.

.1.4.2. **שירותי מסרים מיידיים (WhatsApp, Telegram)** יכולו מרוחחים בלתי סדרים עם  
חבילות קטנות יותר.

## עד כמה ניתן לזהות את האתר/אפליקציה?

### תרחיש 1: התקף חשוף לגודל החבילה ולחומרת הזמן בלבד .1.5

רמת הדיק עדין גבואה יחסית, שכן תכניות התעבורה ייחודיות לכל שירות.

בעזרת למידת מכונה, תוקפים יכולים לזהות שירותי ספציפיים עבור שירותי כמו Spotify ו-YouTube, Netflix, Zoom

מחוקרים הראו שניתן לזהות **מערכת הפעלה ודפדפן** על בסיס מאפייני חבילות בלבד.

### תרחיש 2: התעבורה עברת אונומיזציה (VPN, TOR) .1.6

**VPN:** עדין ניתן לזהות את סוג השירות (למשל, סטרימינג מול שיחת וידאו), אך קשה יותר להבחין בין שירותי דומים.

**TOR:** ההצפנה והטיול דרך מספר שירותי מטאטשים חלק מהדפוסים, אך התקף יכול עדין לשער באיזה סוג שירות מדובר, במיחוד אם נעשה שימוש בחבילות בגודל קבוע או אם יש דפוסים עקובים בין החבילות.

## מה נוכל לעשות כדי להקשות על התוקף?

### 1. שימוש בפaddirינג (Padding)

הוספת נתונים רנדומליים לחבילות, כך שכלן יהיו בגודל אחיד (לדוגמה, 1500 בתים), מה שימנע זיהוי של סוג היישום לפי גודל החבילות.

זה מונע מצב לדוגמא שבו תוקף מזיהה שירותי סטרימינג משתמש בחבילות גדולות, בעוד שירות VoIP משתמש בחבילות קטנות.

### 2. טשטוש תעבורה (Traffic Obfuscation)

שינוי הדפסים של שליחת הנתונים, כך שלא יוכל לזהות אפליקציות על בסיס חותמות זמן.  
לדוגמה, שירותי VPN מסוימים ששולחים פקודות דמה או דוחים חבילות בכונה, כדי לשבש את הדפסים.

### 3. VPN / Proxy

VPN מצפין את כל התקשרות ושולח את הנתונים דרך שרת ביןיהם, כך שהtokף לא יוכל לזהות את ה-4-tuple.

Proxy עובד בצורה דומה אך רק על תעבורה מסוימת (למשל דפדפן).

### 4. שימוש בפרוטוקולים כמו TOR

TOR שולח את התקשרות דרך מספר שרתים שונים בעולם כדי לטשטש את הנתיב.  
מכיוון שהחבילות מוצפנות ונשלחות דרך מספר צמתים, תוקף מקומי לא יוכל לראותו لأن המשתמש מתחבר.

### 5. הוספה רעש לתעבורה (Traffic Shaping)

שליחת חבילות דמה (Dummy Packets) או עיכוב מכוען של חבילות כדי לבלב את התוקף.  
שימוש בפרוטוקולים כמו FPE (Format-Preserving Encryption) לטשטוש דפסים.

## **בונוס:**

שמרנו את ה הקלטה ל Seite הבונוס תחת התיקייה \res\pcapfiles\Bonus-Lecture-Mix .  
ובנוסף ניתן לגשת גם למפתחות הרצפנה שלה במידת הצורך שנמצאים באותה התיקייה.

כאשר ביצענו את הניסוי מחדש - csv הראשון שמכיל את כל המאפיינים נשמר ביחד עם המפתחות ולאחר הפענוח בתיקייה :

\res\csv-files\

וכאשר ביצענו את החלק השני של הניסוי , כלומר התקף מקבל לידי רק את גודל הפקחות וחوتמת הזמן , ה- csv נשמר כאשר נמצא בו רק המידע הלא מוצפן (למרות שאין גישה לתוךו لكن זה לא משנה). ניתן למצוא אותו בתיקייה :

\res\csv-files-encrypted\

**תוצאות מחלק א':**

==== CLASSIFICATION REPORT ===				
	precision	recall	f1-score	support
Audio Streaming	0.90	0.95	0.92	19
Video Calls	0.95	0.90	0.92	20
Video Streaming	1.00	0.50	0.67	2
Web Browsing	0.67	1.00	0.80	2
accuracy			0.91	43
Accuracy:	0.91			

**תוצאות מחלק ב':**

==== CLASSIFICATION REPORT (LIMITED DATA) ===			
Class	Precision	Recall	Support
-----			
Audio Streaming	1.00	0.00	1
Unknown	0.00	0.00	1
Video Calls	0.00	0.00	1
Video Streaming	1.00	0.00	1
Web Browsing	0.50	1.00	2
Accuracy:	0.33		

## השוואה בין שתי טבלאות הסיווג

### 1. גודל מערך הדגימות והקטגוריות

בטבלה הראשונה (43 דוגמאות סך-הכל) מופיעות ארבע קטגוריות (Calls, Video Streaming, Web Browsing, Audio Streaming, Video). עם מספר דוגמאות גדול יחסית בכל קטgorיה. זה מאפשר למודל ללמידה טוב יותר את התכונות הסטטיסטיות של כל סוג תעבורת, וכן דיוק הסיווג (Accuracy=0.91) לעומת זאת, בטבלה השנייה (5 דוגמאות סך-הכל) רואים קטגוריות דוגמת "Web Browsing", "Video Calls", "Video Streaming" ו-"Unknown", "Video" — חלקן עם דוגמה אחת בלבד. כאשר יש רק דוגמה אחת בקטgorיה, דיוק המדדים (Precision, Recall) הגיע לכל טעות בודדת, והתוצאה הסופית נמוכה (Accuracy=0.33).

### 2. השפעת מספר הדגימות על ביצועי המודול

כמות גדולה יותר של דוגמאות בכל קטgorיה (כמו בטבלה הראשונה) מאפשרת למודל להבחן טוב יותר בין סוגי התעborות. בטבלה השנייה, כל קטgorיה כמעט לא מכילה דוגמאות, מה שambil לא-יציבות במידדים (Precision ו-Recall יכולים להיות 0 או 1 במקרה של הצלחה או כישלון בודד). כתוצאה לכך, הדיוק הכללי נפגע ממשמעותית.

### 3. הבדלי ביצועים

- בטבלה הראשונה רואים Precision גבוהה (0.90+) ו-Recall גבוהה (1.00, 0.95) ברוב הקטגוריות, משקף יכולת טוביה זו לזרות נכון דוגמאות שייכות לקטgorיה (Recall) והן להימנע מלסוג בטעות דוגמאות לקטgorיה לא נכון (Precision).
- בטבלה השנייה, חלק מהקטגוריות מקבלות Recall=0, מה שמצביע על כך שדגימה בודדת בקטgorיה לא זהותה כלל.

### מסקנה מרכזית

כאשר המודול מקבל ורק מעט מידע (כמו זמן ואורך החבילה) וכמות דוגמאות קטנה לכל קטgorיה, הוא מתבקש מאוד להבחן בין פעילויות שונות (כגון ניגון בסופוטיפי' לעומת שליחת מייל) ולכן קיטלג אותו בתור *Unknown* ולא ידע לסוג אותו במקום הנכון ללא מידע נוסף. לעומת זאת, כשהוא מקבל יותר פרטים (כתובות IP, פורטים, מספר זרים נפרדים בכל CSV) יוכל לדגום כמות דוגמאות גדולה יותר לכל סוג תעborת (בגלל *ID flow*), והוא מסוגל ללמידה חתימה סטטיסטית טובה יותר של כל פעילות ולהגיע לדיוק סיווג גבוה בהרבה.

## נספח 1 : פקודות שנערכנו בAI

תסביר לי בקצרה על פרוטוקול  
mptcp

**MPTCP (Multipath TCP)** הוא הרחבה ל-TCP שמאפשר שימוש בכמה חיבורים בו-זמנית כדי לשפר מהירות, (סלולר + Wi-Fi למושל) יציבות ועמידות בפני ניתוקים. אם חיבור אחד נופל, הנתונים המשיכים לזרום דרך האחרים.

כיצד אני בונה בפייתון אגרפים?

בפייתון ניתן לבנות אגרפים באמצעות הספרייה **Matplotlib**, ולעיתים גם **Seaborn** להדימות מתקרדות יותר.

### דוגמה בסיסית עם Matplotlib

Copy code

Python

```
import matplotlib.pyplot as plt

# נתונים לדוגמה
x = [1, 2, 3, 4, 5]
y = [10, 20, 15, 25, 30]

# ייצרת אגרף קווי
x, y, marker=plt.plot(x, y, 'o', linestyle='--', color='b', label="נתונים")

# הוספת כותרת לציר X/Y
plt.xlabel("ציר X")
plt.ylabel("ציר Y")
plt.title("דוגמה לארפ בפייתון")
plt.legend() # הציגת מקרא

# הציגת האגרף
plt.show()
```

תכתב לי בקצרה - איזה מידע חותמת זמן יכולה לתת לתוקף?

חותמת זמן יכולה לחסוף לתוקף:

- **זהוי אפליקציה** – תבניות זמן אופייניות לכל שירות (למשל, סטרימינג מול צ'אט).
- **מדידת זמן תגובה (RTT)** – מאפשר להסיק על מיקום השירות.
- **שחזור פעולות משתמש** – מעבר בין עמודים, הפעלת שירותים וכו'.
- **התקפות הזמן על הצפנה** – ניתוח זמני שליחה עשויה להסיגר תוכן תקשורת מוצפן.
- **גילוי זהות משתמש אונונימי** – הצלבה עם תעבורת אחרת עלולה לחסוף משתמשי VPN .Tor

תסביר לי כיצד שומרים מפתחות TLS בקצרה

שמירת מפתחות TLS נעשית בדרכים שונות בהתאם לסוג המפתח ומשימוש בו:

1. **מפתחות זמינים (Session Keys)** – נוצרים במהלך TLS Handshake ומשמשים להצפנת התקשרות. לרוב נשמרם בזיכרון (RAM) (ונמחקים כשהחיבור נסגר).
  2. **מפתחות פרטיים (Private Keys)** – משמשים לזהוי השירות ונשמרים באופן מאובטח בקובץ מערכת (.key, .pem). ב-HSM (חומרה ייעודית), או ב-TPM (רכיב הצפנה במחשב) כדי למנוע אישה לא מושricht.
  3. **מפתחות ציבוריים (Public Keys)** – נמצאים בתעודות TLS (כגון .crt) וניתנים לכל לקוח שמחבר.
- בקצרה, מפתחות TLS זמינים נשמרם בזיכרון ומוחלפים בכל חיבור, בעוד שמפתחות פרטיים מאוחסנים באופן מאובטח להגנה על זהות השירות.

מסביר ל- בקירה כיצד ניתן לבטל שימוש ב프וטוקול QUIC בלבד כרגע  
chromium

כדי לבטל את השימוש ב프וטוקול QUIC-ב-[Google Chrome](#), בצע את השלבים הבאים:

1. פתח את chromium ו הקלד בשורת הכתובת:

Edit ⌂ Copy ↗

chrome://flags/#enable-quic

2. אחרר את האפשרות "Experimental QUIC protocol".

3. שנה את הערך מ- "Disabled" ל- "Enabled" או "Default".

4. הפעיל מחדש את הדפדפן.

זה ימנע מהדף להשתמש ב-QUIC וכיירח אותו להשתמש ב-TCP במקום.