פרוייקט סיום

רשתות תקשורות

מגישות- אגסי נועה (209280635), נדב רוני (325730190) ואיזביצקי בר (212138457)

Link to the drive (pcapng)-
https://drive.google.com/drive/folders/1WNNaHGG2pDRnILCBVo5OqqCd8e36atGM?usp=drive_link

link to the git-

https://github.com/barIzbizki/networkpro/tree/main/%D7%A4%D7%A8%D7%95%D7%99%D7%99%D7%A7%D7%98%20%D7%A1%D7%99%D7%95%D7%9D/finelprog

תוכן עניינים

חלק א-

שאלה 1-

אלו הגורמים לאיטיות בשכבת התעבורה-

עומס ברשת (Network Congestion)-
עומס ברשת עלול לגרום לאבדה של חבילות ולשליחת חבילות חוזרת , מה שמאט את התעבורה
בנוסף, עומס כזה עלול להקטין את חלון ה-(TCP Window Size) TCP בשל חוסר קבלה של אישורים (ACK), מה שמגביל עוד יותר את זרימת התעבורה.
פתרון: הגדלת רוחב הפס ושימוש במסלולים חלופיים

רוחב פס- מייצג את כמות הנתונים שיכולה לעבור דרך חיבור רשת בפרק זמן מסוים.
כאשר יש עומס על הרשת, כל משתמש או אפליקציה מקבלים חלק קטן יותר מרוחב הפס, מה שמוביל להאטת התעבורה.
הגדלת רוחב הפס מאפשרת יותר מקום להעברת הנתונים, כך שתוכל לעבור יותר תעבורה במקביל.
לדוגמא ב-TCP, החלון הזזה יגדל - מה שמאפשר לשדר יותר נתונים לפני שמקבלים אישור (ACK).
פתרון: דרכים להגדלת רוחב הפס- שדרוג חיבור הרשת, שימוש בכבלים מהירים יותר, שדרוג רשת ה-Wi-Fi .

גודל החלון בTCP-
חלון קטן מגביל את כמות המנות שניתן לשלוח ללא אישור, מה שמקטין את ניצול רוחב הפס.
פתרון: הגדלת גודל החלון.
לעומת זאת, חלון גדול מדי עלול לגרום לשליחת כמות רבה של מנות חוזרות במקרה של בעיה ברשת, מה שעלול להחמיר את העומס ברשת.
פתרון: הקטנת גודל החלון.

Timeout קצר-
אם הtimeout קצר מדי, השולח עלול לשלוח מנות מחדש לפני קבלת אישור (ACK) מהמקבל, מה שגורם לשליחת חבילות כפולות-מה שיגרום לעומס נוסף על הרשת, ולעיתים גם להפעלת בקרת הגודש של TCP.
פתרון: הגדרת הtimeout בהתאם לתנאי הרשת כדי למנוע שליחות חוזרות מיותרות.

שאלה 2-

בקרת זרימה ב-TCP הוא מנגנון שמטרתו למנוע מצב שבו השולח שולח בבת אחת כמות נתונים גדולה יותר ממה שהמקבל מסוגל לעבד. זה נעשה כדי למנוע עומס על המחשב שמקבל את הנתונים, ובכך לשמור על יציבות העברת המידע.
בכל חיבור TCP ישנו חלון הזזה שמציין את מספר החבילות שהשולח יכול לשלוח מבלי לקבל אישור מהמקבל (Ack). המחשב שמקבל את הנתונים מציין בשדה ה-flow control את הגודל המרבי של החבילות שהוא יכול לקבל, וכך מווסת הפרוטוקול את קצב השליחה.
אם השולח הרבה יותר חזק (במובן של עיבוד נתונים) מהמקבל, הוא יכול לשלוח הרבה יותר נתונים ממה שהמקבל יכול לעבד בפעם אחת. במקרה כזה, המחשב שמקבל את הנתונים לא יספיק לעבד את כל המידע שהוא מקבל, מה שיגרום לו לאבד חלק מהחבילות, או שהוא יצטרך להשהות את

קבלת החבילות עד שהוא יוכל לעבד אותן.
לכן, כאשר משתמשים בTCP –כמו שציינו קודם לכן הפרוטוקול בודק כמה חבילות ניתן לשלוח עד להמתנה של אישור ובהתאם לכך מתאים את קצב השליחה מכאן נובע שהוא יאיט את קצב השליחה ולכן קצב התעבורה הכללי ירד.

שאלה 3-

ניתוב הוא התהליך שבו נתונים עוברים ממחשב אחד לאחר דרך הרשת. כאשר קיימים מספר מסלולים אפשריים בין המקור ליעד, הנתב (Router) בוחר את המסלול הטוב ביותר על סמך קריטריונים שונים, כמו מהירות, עומס ברשת ויציבות. בחירה נכונה של מסלול ניתוב משפיעה על ביצועי הרשת ויכולה לגרום לתקשורת להיות מהירה ויעילה יותר, או להפך — איטית ומלאה בשגיאות.

כיצד בחירת המסלול משפיעה על ביצועי הרשת?
א. Round Trip Time – RTT
RTT הוא משך הזמן שלוקח לשלוח הודעה מהמקור ליעד ולקבל תגובה חזרה. אם הנתונים עוברים דרך פחות נתבים (Hops), זמן הסבב קצר יותר, מה שמשפר את הביצועים. אם המסלול כולל חיבורים לא יציבים או עיכובים בתקשורת, זמן הסבב יגדל, מה שיגרום להאטה בתגובת הרשת.
ב. רוחב פס- קובע כמה נתונים ניתן להעביר בפרק זמן מסוים.
מסלול עם קיבולת גבוהה מאפשר העברת יותר מידע במקביל, מה שמאפשר מהירות גבוהה יותר.
אם ישנו נתב במסלול שתומך בגודל קטן יותר של חבילות מידע ,ייתכן שיהיה צורך לפצל את המידע מה שיכול לגרום לעיכובים בתקשורת.
ג. אובדן חבילות – דבר זה קורה כאשר חלק מהמידע שאנו שולחים לא מגיע ליעדו וצריך לשלוח אותו שוב. אם המסלול עמוס מאוד בתנועה, עלולות להיות שגיאות בתקשורת והחבילות לא יגיעו ליעדן.
במקרים כאלה, עדיף לבחור מסלול קצת יותר ארוך, אך כזה שכולל פחות עומס ולכן יהיה אמין יותר.

הגורמים שיש לקחת בחשבון כשבוחרים מסלול ניתוב-
א. מדיניות הניתוב - קובעת האם הרשת תעדיף מסלול מהיר יותר או מסלול יציב יותר. לדוגמה, ברשתות של סטרימינג או גיימינג, העדיפות תהיה למהירות נמוכה יותר של RTT לעומת זאת, ברשתות שיש בהן מאגרי מידע ישנה עדיפות ליציבות/אמינות על פני מהירות כדי למנוע אובדן מידע קריטי.
ב. איכות החיבור - כדי לבחור מסלול נכון, יש לבדוק כמה פרמטרים:
רוחב הפס — כמה נתונים ניתן להעביר דרך המסלול.
עומס – כמה משתמשים מחוברים לאותו נתיב ברגע נתון.
אובדן מנות – כמה מהחבילות שאנו שולחים מגיעות ליעדן ללא בעיות.
ג. איזון עומסים במקום להעביר את כל התנועה דרך נתיב אחד, אפשר לפצל אותה בין כמה נתיבים כדי למנוע עומס יתר. לדוגמה, אם ישנם שני נתיבים עם זמני תגובה דומים, הרשת יכולה לחלק את התעבורה ביניהם ולמנוע עיכובים. איזון עומסים עוזר במיוחד ברשתות גדולות עם הרבה תנועה בו-זמנית.
ד. איכות השירות - לא כל המסלולים ברשת מתאימים לכל סוגי השירותים. לדוגמה, בשיחות וידאו חשוב לבחור נתיב עם זמני תגובה נמוכים. לעומת זאת, בהעברת קבצים גדולים (כמו גיבוי נתונים), עדיף לבחור נתיב עם רוחב פס גבוה גם אם הוא קצת יותר איטי.

שאלה 4-

MPTCP -Multipath TCP - מאפשר להשתמש במספר נתבים בו-זמנית בתוך אותו חיבור

TCP. כיצד הוא משפר את ביצועי הרשת?
א. מהירות גבוהה יותר ושימוש יעיל ברוחב הפס-
MPTCP מאפשר להשתמש בכמה נתיבים במקביל, מה שמאפשר למידע לעבור מהר יותר. לדוגמה, אם יש לך חיבור גם ל-Wi-Fi וגם לרשת סלולרית, MPTCP יכול להשתמש בשניהם יחד כדי להעביר נתונים מהר יותר.
ב. חיבור יציב יותר — התעבורה לא תפסק במידה וישנה תקלה באחד הנתיבים- אם הנתיב שבו המידע עובר נתקע או מפסיק לעבוד, TCP רגיל היה גורם לניתוק עד שיימצא פתרון.
עם MPTCP, המידע יכול לעבור לנתיב אחר בלי שתהיה הפסקה בחיבור, מה שמבטיח שהתקשורת תמשיך לפעול.
ג. איזון עומסים — פחות עומס על הרשת- רשת האינטרנט לפעמים עמוסה, וזה גורם להאטה בתקשורת.

MPTCP יודע לפצל את המידע לכמה נתיבים, כך שאם נתיב אחד עמוס, המידע יעבור דרך נתיב אחר שבו התנועה זורמת מהר יותר.

שאלה 5-

אובדן חבילות ברשת יכול להתרחש כאשר יש עומס על הנתבים, בעיות בהגדרות תקשורת או תקלות חומרה, והוא יכול לקרות הן בשכבת הרשת (IP) והן בשכבת התעבורה (TCP/UDP). אחת הסיבות המרכזיות לאובדן חבילות היא עומס על הרשת – כאשר נתב מקבל יותר חבילות ממה שהוא מסוגל לעבד, הוא מתחיל לזרוק חבילות כדי למנוע קריסה. ניתן לפתור בעיה זו על ידי שימוש בנתבים עם יותר זיכרון ועיבוד מהיר יותר, איזון עומסים בין נתיבים שונים והרחבת הרשת עם נתבים נוספים שיפחיתו את הלחץ. גורם נוסף הוא בעיה בגודל ה-MTU — כאשר חבילות גדולות מדי דורשות פיצול (Fragmentation), דבר שעלול לגרום לאובדן חבילות. ניתן לפתור זאת על ידי קביעת גודל אחיד של MTU בכל הרשת או שליחת חבילות קטנות יותר מההתחלה. כמו כן, חבילות יכולות להימחק אם ערך ה-TTL שלהן נמוך מדי, ולכן יש להגדיר TTL גבוה מספיק כדי שיגיעו ליעדן ולבחון אם ניתן להשתמש בנתיב עם פחות תחנות בדרך.

גם בשכבת התעבורה TCP/UDP יש גורמים שיכולים להוביל לאובדן חבילות. למשל, חוסר התאמה בין חלון השולח לחלון הקבלה של המקבל עלול לגרום לכך שהשולח שולח יותר מידע ממה שהמקבל יכול לעבד, ולכן חבילות ייזרקו. פתרון אפשרי הוא להתאים את גודל החלון כך שיתאים לקיבולת של המקבל ולהשתמש במנגנוני בקרת עומס. בעיות בתזמון הן גורם נוסף — אם השולח לא מקבל אישור קבלה (ACK) מהמקבל בזמן, הוא עלול להניח שהחבילה אבדה ולשלוח אותה שוב, אף על פי שהמקבל פשוט היה איטי במתן האישור. ניתן לפתור זאת על ידי התאמת משך ההמתנה (TimeOut) לפני שליחה מחדש, כך שהוא יתאים לתנאי הרשת.

חלק ב-

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

שאלה 1-

המאמר מציג שיטה חדשנית לזיהוי מערכת ההפעלה, הדפדפן והאפליקציה של המשתמש באמצעות ניתוח מאפייני תעבורה חיצוניים (Metadata) של חבילות HTTPS מוצפנות, ללא צורך בפענוח ההצפנה עצמה.

עד כה, מרבית המחקרים שהתמקדו בזיהוי תעבורה פעלו בשתי גישות עיקריות:

1. ניתוח תעבורה לא מוצפנת –שיטה זו מאפשרת גישה ישירה לתוכן החבילות, אך אינה ישימה כאשר נעשה שימוש ב HTTPS-ובפרוטוקולי הצפנה מודרניים.
2. זיהוי יישומים בלבד –מתמקדת בזיהוי השירות שבו נעשה שימוש למשל Facebook , או YouTubeאך אינה מאפשרת קביעת מערכת ההפעלה או הדפדפן של המשתמש.

המאמר מציע גישה חדשה המאפשרת לזהות לא רק את האפליקציה, אלא גם את מערכת ההפעלה והדפדפן ,תוך הישענות על מאפייני תעבורה חיצוניים. שיטה זו מתבססת על ניתוח פרמטרים כגון:
גודל ותזמון החבילות (Packet Size & Timing) , שלב ה Handshake-שלTLS/SSL ,
מספר הרחבות SSL ואורך מזהה Session ID , דפוסי "התפרצות" של
דפדפנים Bursty Behavior , ניתוח חלונותTCP
באמצעות גישה זו, ניתן למפות במדויק את סביבת המשתמש גם כאשר כל התקשורת מוצפנת, תוך השגת רמת דיוק של 96.06%.
בנוסף, המחקר מתמקד בזיהוי תעבורה ממחשבים שולחניים וניידים (Desktops & Laptops) בלבד , ואינו עוסק בזיהוי במכשירים סלולריים.

תרומה נוספת של המאמר היא הנגשת מאגר נתונים נרחב (מעל 20,000 דוגמאות מסומנות) הכולל מידע על מערכות ההפעלה, הדפדפנים והאפליקציות שנבדקו. נתון זה מאפשר לחוקרים אחרים להשתמש בנתונים לצורך פיתוח שיטות מתקדמות לזיהוי תעבורה מוצפנת.

שאלה 2-

המאמר מציע תכונות חדשות שלא נעשה בהן שימוש קודם לצורך זיהוי מערכת ההפעלה והדפדפן:

מאפייני SSL/TLS-
כאשר משתמש ניגש לאתר באמצעות HTTPS, מתרחש תהליך Handshake שבו הדפדפן שולח מידע על פרוטוקולי ההצפנה שהוא תומך בהם. המאמר גילה כי מערכות הפעלה ודפדפנים שונים שולחים מזהי SSL ייחודיים, ולכן ניתן לזהות אותם מבלי לקרוא את המידע המוצפן עצמו.
המאפיינים שנבדקו כוללים:

א. SSL Extension Count – כמות הרחבות ההצפנה הנשלחות במהלך תהליך ה-Handshake.
ב. SSL Session ID Length — משתנה בין דפדפנים שונים.
ג. SSL Cipher Methods Count — לכל מערכת הפעלה ודפדפן יש שיטות הצפנה מוגדרות שהוא תומך בהן.

מאפייני TCP-
פרוטוקול TCP מנוהל בצורה שונה בכל מערכת הפעלה, ולכן ניתן להשתמש בהתנהגות הפרוטוקול לצורך זיהוי מערכת ההפעלה:

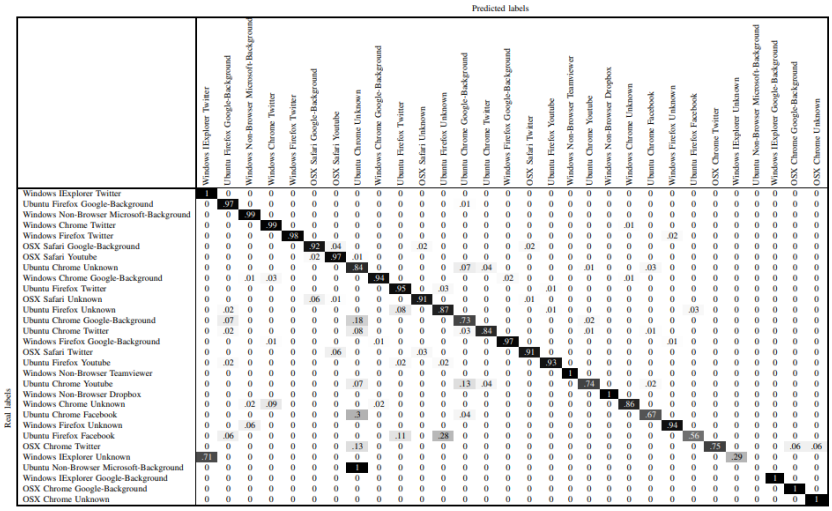
א. TCP Initial Window Size — כמות הנתונים שניתן לשלוח לפני קבלת אישור (ACK). ב. TCP Window Scaling Factor — מנגנון להגדלת חלון הנתונים בהתאם למצב הרשת.
ג.
קצב ורציפות שליחת החבילות -

המאמר מצא כי דפדפנים שונים שולחים חבילות בקצב שונה ובצורה לא רציפה. כל דפדפן נוטה לשלוח כמות של מידע, ולאחר מכן לעצור לזמן קצר.
כתוצאה מכך, ניתן להשתמש בתבניות של קצב ורציפות התעבורה כדי לזהות באיזה דפדפן המשתמש עושה שימוש.
המאפיינים שנבדקו:

א. מספר הפעמים שהדפדפן שולח כמות גדולה של מידע ברצף ולאחר מכן עוצר. ב. מהירות המקסימום שבה הדפדפן שולח מידע בפעם אחת.

שאלה 3-

3. המאמר מראה ששימוש במאפיינים החדשים הוביל לשיפור בדיוק הסיווג של מערכת ההפעלה, הדפדפן והאפליקציה.



(a) Tuple Confusion Matrix

תוצאות המודל-

זיהוי מערכת ההפעלה –דיוק של 100%
זיהוי דפדפן -דיוק של 99%-97%
זיהוי אפליקציה – דיוק של 96%

המטריצה שנלקחה מהמאמר מראה את הביצועים של המודל-
השורות מייצגות את המערכת שהמשתמש באמת השתמש בה.
העמודות מייצגות את התחזיות של המערכת.
האלכסון הראשי של המטריצה מייצג את מספר הסיווגים שנעשו נכון- המקרים שבהם המערכת זיהתה בדיוק את מערכת ההפעלה, הדפדפן והאפליקציה של המשתמש.

ערכים גבוהים באלכסון מעידים על כך שהשיטה שהמאמר הציע אכן אפקטיבית בזיהוי. אם מערכת ההפעלה לא זוהתה נכון, היא קוטלגה כ"לא ידועה" (Unknown).

Early Traffic Classification With Encrypted ClientHello: A Multi-Country Study

שאלה 1-

המאמר עוסק בפיתוח שיטה חדשה לסיווג מוקדם של תעבורת רשת מוצפנת, תוך התמקדות בפרוטוקול TLS 1.3 עם מנגנון Encrypted Client Hello (ECH). השינוי שמבצע ECH נועד להסתיר מידע קריטי על החיבורים ברשת, כמו ה-SNI (Server Name Indication), ובכך מונע את השימוש בשיטות סיווג מסורתיות שהתבססו על מידע זה. עד כה, נחשב כי הצפנה זו הופכת את סיווג התעבורה לבלתי אפשרי, אך המאמר מראה כי עדיין ניתן לבצע סיווג אפקטיבי גם בסביבה מוצפנת.

כדי להתמודד עם אתגר זה, המאמר מציג אלגוריתם חדש בשם hRFTC (hybrid Random Forest Traffic Classifier). אלגוריתם זה משלב נתונים סטטיסטיים של זרמי תעבורה עם מאפיינים בלתי מוצפנים מתוך פרוטוקול TLS, כדי לזהות סוגים שונים של שירותים ותעבורה. בשונה מאלגוריתמים קודמים שהתבססו על מידע חשוף כמו SNI, האלגוריתם החדש משתמש בדפוסים סמויים בתוך המבנה של זרימת התעבורה ובכך מאפשר סיווג ברמת דיוק גבוהה, גם כאשר המידע המזהה מוסתר.

המחקר מנתח את מגבלות השיטות הקיימות, מראה את הפגיעות שלהן בעקבות יישום ECH, ומציג את היתרונות של הגישה ההיברידית החדשה. הוא כולל בדיקות על מערכי נתונים מגוונים ממדינות שונות כדי להוכיח את יכולת ההכללה של האלגוריתם, ומראה כיצד ניתן להשתמש בו כדי לשפר את יכולות סיווג התעבורה מבלי לפגוע בפרטיות ובאבטחת המידע של המשתמשים.

בסופו של דבר, המאמר מדגיש את ההשפעה הרחבה של שינויי ההצפנה החדשים, בוחן את האיזון בין פרטיות לסיווג תעבורה, ומציע פתרון חדשני שמאפשר לסווג תעבורה באופן יעיל גם בעידן שבו יותר ויותר מידע ברשת מוצפן לחלוטין.

שאלה 2-

המאמר מתבסס על שילוב של מאפיינים בסיסיים ו-חדשים, כדי לסווג סוגי שירותים ברשת.
מאפיינים בסיסיים-
מאפיינים אלו נעשה בהם שימוש במחקרים קודמים לסיווג תעבורה, כמו-
גודל חבילות – ניתוח הערכים המינימליים, המקסימליים והממוצעים של חבילות הרשת.
תזמון בין חבילות (Inter-Packet Timing) – בדיקה של זמני ההגעה והשליחה של חבילות בין הלקוח לשרת, שמסייעת בזיהוי דפוסים ייחודיים לכל שירות.
יחס תעבורה קדימה/אחורה — השוואת מספר החבילות שנשלחו מהלקוח לשרת מול מספר החבילות שהתקבלו, כדי לזהות שירותים בעלי תבניות העברת מידע ייחודיות.

מאפיינים חדשים - בנוסף למאפיינים המוכרים והסטנדרטיים, המאמר מראה שיטות חדשות המתמקדות בפרוטוקול TLS 1.3 ו-QUIC, המאפשרות זיהוי שירותים גם בתנאים של הצפנה מתקדמת.
TLS Cipher Suites — ניתוח האלגוריתמים הנתמכים על ידי הלקוח והשרת, שמספקים רמזים לגבי סוג היישום.
Session Length & Extensions — התבוננות במספר תוספי TLS שבהם נעשה שימוש ואורך הסשן, אשר מספקים מאפיינים ספציפיים לכל שירות.
ניתוח חבילות QUIC — הבנת אופן פעולתו של פרוטוקול QUIC, תוך התמקדות במנגנון Padding QUIC, שמטרתו לטשטש את מאפייני התעבורה ולהקשות על הסיווג שלה.

שאלה 3-

תוצאות הניסוי ומסקנות:

א. שיפור דיוק הסיווג-
האלגוריתם החדש ,hRFTC, מצליח לסווג תעבורה מוצפנת בדיוק של ,94.6%בהשוואה לאלגוריתמים מבוססי TLS שהגיעו רק ל .38.4%-נתון זה מוכיח כי גם בתנאים של TLS 1.3עם ,ECH בהם מידע קריטי כמו ה SNI-מוסתר, ניתן עדיין לבצע סיווג תעבורה ברמת דיוק גבוהה.

ב. השפעת כמות נתוני האימון-
גם כאשר האלגוריתם אומן על 10% בלבד מהנתונים ,הוא הצליח להגיע לדיוק של ,87.2%מה שמצביע על כך שהוא כלי למידה חזק ויעיל שיכול לבצע סיווג גם עם כמות נתונים מצומצמת.

ג. השפעת מיקום גיאוגרפי על הביצועים-
התוצאות מצביעות על כך שכאשר האלגוריתם אומן במדינה אחת, הוא לא בהכרח מניב ביצועים טובים במדינה אחרת. בהתאם לכך, יש צורך להתאים את מודל הסיווג לכל מדינה בנפרד ,כפי שניתן לראות בטבלה המצורפת, המפרטת את ההבדלים בדיוק הסיווג בין מדינות שונות.

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

ד. מאפייני חבילות והשפעתם על הדיוק-
נמצא כי גודל החבילות הוא הפרמטר בעל ההשפעה הגבוהה ביותר על יכולת הסיווג. עם זאת, גם תזמון החבילות ויחס כמות החבילות (uplink/downlink) היו בעלי משמעות בסיווג התעבורה.

ה. השפעת פרוטוקול QUIC על דיוק הסיווג-
נמצא כי פרוטוקול QUIC מקשה על זיהוי תעבורה, שכן הוא כולל מנגנון ,QUIC Paddingאשר ממלא את החבילות בנתונים אקראיים כדי לטשטש את מאפייני התעבורה. עם זאת, ניתן להתגבר על כך באמצעות ניתוח חכם של גודל החבילות ופרמטרים נוספים שנשארים גלויים גם תחת הצפנה.

ו. השפעות ההצפנה והפרטיות-
למרות שהצפנת ECH נועדה להסתיר מידע על סוג השירותים שבהם משתמשים גולשים, האלגוריתם הצליח לשחזר מידע שימושי גם ללא .SNI ממצא זה מדגיש את החשיבות של שימוש בטכניקות סיווג מתקדמות ,אשר מאפשרות לגורמים מסוימים, כמו ספקיות אינטרנט וארגוני אבטחה ,לעקוב אחר סוג הפעילות המתבצעת ,גם אם לא ניתן לזהות במדויק את האתר או השירות.

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

שאלה 1-

המאמר מציג גישה חדשה לסיווג תעבורת אינטרנט מוצפנת, הנקראת FlowPic, אשר ממירה flow data לתמונות ומעבירה אותן לרשתות נוירונים (CNN) לצורך סיווג. בניגוד לשיטות קודמות שהתבססו על תכונות שנבחרו ידנית ומודלים של למידת מכונה רדודה, FlowPic משתמש בגישה חדשנית שבה גודל החבילה וזמני הגעת החבילות מועברים לייצוג תמונתי דו-ממדי , אותו מעבדים באמצעות טכניקות לזיהוי תמונות.
שיטה זו מאפשרת סיווג מהיר ומדויק של סוגי תעבורה, כגון VoIP, וידאו, צ'אט, גלישה והעברת קבצים, גם כאשר נעשה שימוש בפרוטוקולי הצפנה מתקדמים כמו VPN או Tor. יתרון נוסף הוא שהשיטה מתמודדת היטב עם זרימה חד-כיוונית של נתונים בחלון קצר, מה שהופך אותה יעילה במגוון תרחישים.
בנוסף, FlowPic אינו נשען על תוכן החבילות אלא רק על המאפיינים הסטטיסטיים שלהן, ולכן יש לו יתרון משמעותי מבחינת פרטיות – הוא מסוגל לסווג תעבורה בצורה אפקטיבית מבלי לחשוף מידע רגיש של המשתמשים.

שאלה 2-

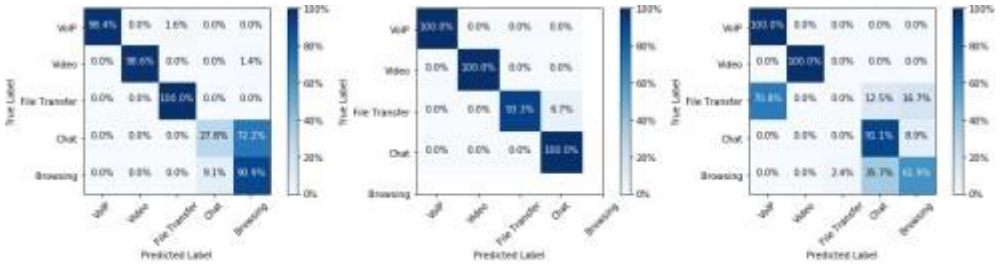
המאמר משתמש במאפייני תעבורה בסיסיים כגון גודל החבילה, זמני הגעה בין חבילות וכיוון הזרימה, אך מחדש בכך שהוא מייצג את הנתונים כהיסטוגרמות דו-ממדיות, מה שמאפשר שמירה על מידע ללמידת עומק. בנוסף, בשונה משיטות קודמות המסתמכות על אלגוריתמים פשוטים כמו KNN ועצי החלטה, FlowPic משתמשת בלמידה עמוקה לצורך סיווג מדויק ואפקטיבי.
יתרון משמעותי נוסף של השיטה הוא ההתמודדות שלה עם הצפנה — מכיוון שאינה דורשת גישה לתוכן החבילות, היא מסוגלת לסווג תעבורה גם כאשר נעשה שימוש בפרוטוקולים מאובטחים כמו VPN ו-Tor. כמו כן, המודל אינו מזהה רק פרוטוקולים, אלא מסוגל לסווג את התעבורה לפי קטגוריות שימוש כגון VoIP, וידאו, צ'אט, גלישה והעברת קבצים. גישה זו מהווה פתרון שאינו פוגע בפרטיות, ומתמודד טוב גם עם זרימות נתונים קצרות וחד-כיווניות.

שאלה 3-

תוצאות המאמר —

Class	Accuracy (%)			
VoIP	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	**99.6**	99.4	48.2
	VPN	95.8	**99.9**	58.1
	Tor	52.1	35.8	**93.3**
Video	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	**99.9**	98.8	83.8
	VPN	54.0	**99.9**	57.8
	Tor	55.3	86.1	**99.9**
File Transfer	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	**98.8**	79.9	60.6
	VPN	65.1	**99.9**	54.5
	Tor	63.1	35.8	**55.8**
Chat	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	**96.2**	78.9	70.3
	VPN	71.7	**99.2**	69.4
	Tor	85.8	93.1	**89.0**
Browsing	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	**90.6**	-	57.2
	VPN	-	-	-
	Tor	76.1	-	**90.6**

FlowPic השיג דיוק של 85.0% עבור תעבורה שאינה מוצפנת ב-98.4% ,VPN עבור VPN, ו-67.8% בלבד עבור Tor, מה שמעיד על כך שהצפנת Tor מצליחה להסתיר טוב יותר את דפוסי הזרימה. בנוסף, זיהוי היישומים התבצע בדיוק של 99.7%, ובכך עקף את השיטות הקודמות כמו KNN
(93.9%).
כמו כן, FlowPic הצליח לזהות תעבורה מוצפנת גם כאשר לא אומן על תעבורה כזו מראש, עם 99.4% דיוק בזיהוי VPN, מה שמעיד על כך שהצפנת VPN אינה מעלימה לחלוטין את דפוסי התעבורה.



(a) Non-VPN (b) VPN (c) Tor

בנוסף, מטריצות הסיווג מצביעות על כך שמרבית הקטגוריות מסווגות נכון, אך קיים בלבול מסוים בין צ'אט לגלישה, ככל הנראה בשל הדמיון ביניהן. עוד נמצא כי דיוק הזיהוי של VPN גבוה יחסית, בעוד שדיוק הסיווג תחת Tor נמוך יותר, בשל מנגנוני ההצפנה המתקדמים שלו.
לבסוף, אחת התובנות המרכזיות היא היכולת של FlowPic לבצע הכללה רחבה, כך שגם אם הוא לא אומן על יישום מסוים, הוא עדיין מצליח לסווג אותו לקטגוריה הנכונה. למשל, המודל זיהה תעבורת Facebook Video בדיוק של 99.9%, מה שמעיד על כך שהוא לומד דפוסי זרימה כלליים ולא מאפיינים ספציפיים של אפליקציות בודדות.

חלק ג -

התפלגויות פרוטוקולים-

בגרפים אלו ניתן לראות עבור כל הקלטה (יוטיוב, ספוטיפיי וכדומה) באילו פרוטוקולים היה שימוש וכמה חבילות השתמשו בכל פרוטוקול, גרפים אלו מבהירים את התעבורה ברשת.
מעבר לכך, הפרוטוקולים מלמדים אותנו רבות על מטרת התקשורת, אופי הנתונים שמועברים, רמת האמינות ורמת המהירות.

פרוטוקול TCP- שימוש בTCP מעיד על אמינות, הוא מבטיח שכל החבילות יגיעו (במידה ולא הגיעו הוא שולח שוב) וכתוצאה מכך עובד בצורה איטית יותר.

פרוטוקול UDP- נשתמש כאשר נרצה שהחבילות יגיעו בצורה מהירה ללא אימות להגעתן ושליחה מחדש ולכן הוא לא אמין. משתמשים כאר יש צורך בתגובה מיידית.

פרוטוקול QUIC- משלב את UDP,TCP ע"י חיבור אמין ומהיר.

פרוטוקול DNS- ממיר שמות דומיין לכתובות IP. מאפשר למשתמשים לגשת לאתרים באמצעות שמות דומיינים קריאים במקום כתובות מספריות.

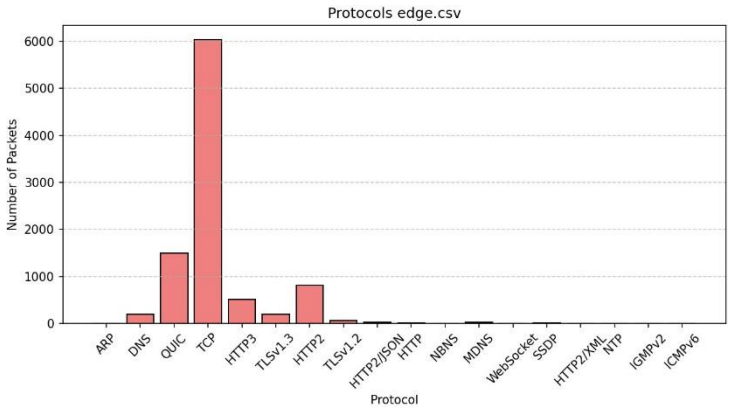
פרוטוקול HTTP- פרוטוקול התקשורת הבסיסי של האינטרנט, שתפקידו להעביר בקשות ותשובות בין הלקוח (דפדפן אינטרנט) לשרת (אתר אינטרנט)

פרוטוקול HTTP/3- גרסה של HTTP שמבוססת QUICK. משפר את ביצועי התקשורת באינטרנט אמין ומפחית את זמן השהיה.

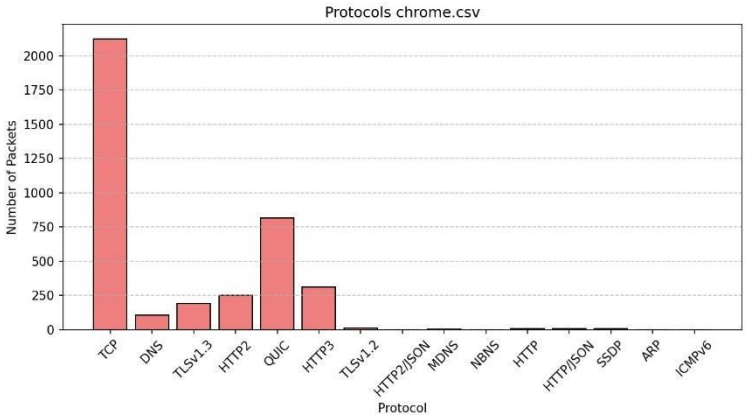
פרוטוקול TLS - תפקידו להגן על התקשורת ברשת, במיוחד על ידי הצפנת הנתונים שנשלחים בין שני מחשבים.

מכיוון שלכל פרוטוקול יש מאפיינים משלו נראה ששימוש רב בפרוטוקול מעיד על מאפייני הרשת ודרך התקשורת.

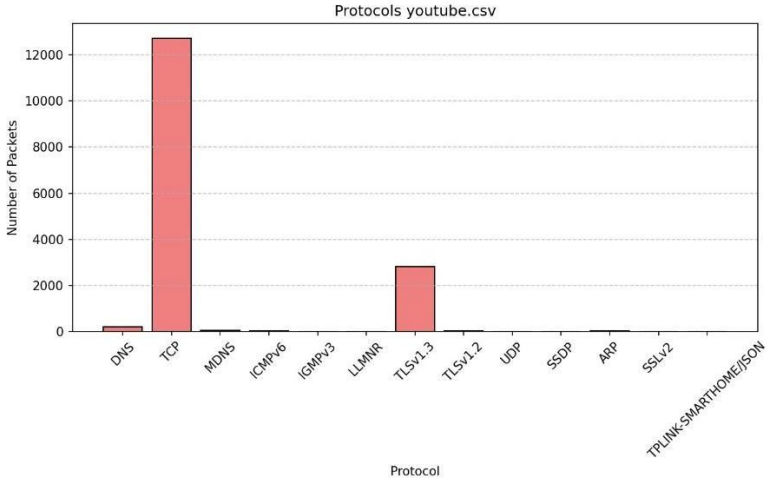
היינו מצפות שבאתרי האינטרנט שליחת החבילות תתבצע באופן אמין ולכן נראה מס רב של חבילות וTCP http (מכל הגרסאות). בזום היינו מצפות לUDP רב כיוון שפחות חשוב האמינות אלא יותר המהירות של שליחת החבילות בזמן אמת. ביוטיוב נצפה לפרוטוקולים רבים של הצפנה



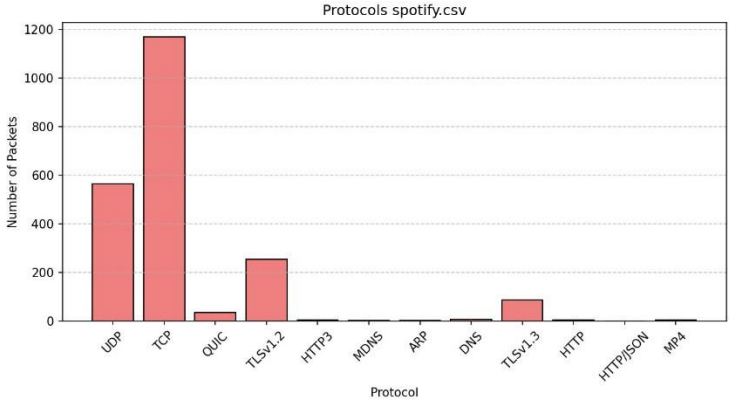
ניתן להבין מהגרף **edge** שהיה שימוש רב בפרוטוקולי TCP, QUIC HTTP.
משימוש בפרוטוקולים אלו ניתן ללמוד שEDGE מעדיף אמינות, משפר ביצועי מהירות ומבוסס על אינטרנט



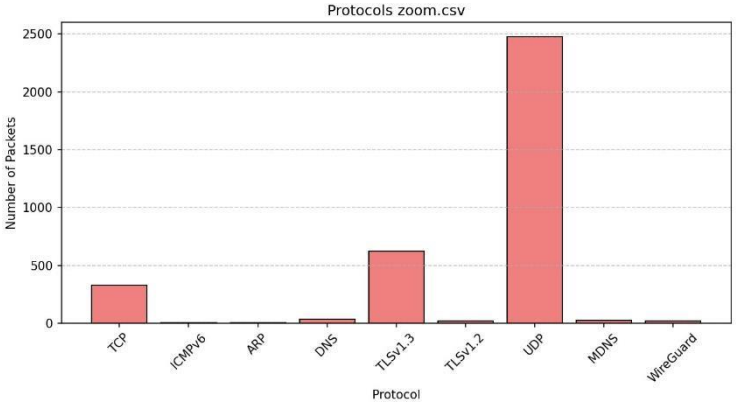
ניתן להבחין כי בגרף **chrome** יש שימוש רב בTCP ושימוש ב- QUICK, HTTP3, tlsv1.3, dns השירותים בchrome דורשים אמינות יחד עם שירות מהיר של quic.



ניתן להבחין כי בגרף **youtube** יש שימוש רב בTCP, tlsv1.3
בהקלטה המקורית היה שימוש רב בquic (שסונן בשביל הניתוח) שמראה על מהירות יחד עם אמינות ואיכות גבוהה. בעקבות הביטול ייתכן תוצאות איטיות יותר בעקבות הסינון נשארנו עם אמינות בנוסף הtlsv1.3 מעיד על הצפנה חזקה.



ניתן להבחין כי בגרף **spotify** יש שימוש רב בtcp, udp
udp — השמיעה היא בזמן אמת ולכן יש צורך לשלוח במהירות
tcp - מעיד על אמינות השליחה של החבילות
שימוש בשניהם ביחד בדר"כ מעיד על איכות גבוהה שמצריכה יותר רוחב פס.



ניתן להבחין כי בגרף **zoom** יש שימוש רב ב -udp מכיוון שזה וידיאו בזמן אמת וחשוב יותר המהירות מהאמינות פרוטוקול זה מאפשר איכות גבוהה של וידיאו עם עיכוב מינימלי.

tlvs1.3 מעיד על אבטחה ברמה גבוהה אך רק על חלק מהנתונים כי הוא לא מופיע הרבה.

tcp במקרה שלנו כנראה מחפה על שגיאות

דגלי TCP–

הדגלים הם חלק מהכותרת של חבילות TCP שמשמשים לניהול תקשורת ולשליטה בהתחלה, בסיום ובמהלך חיבור. כל דגל מייצג פעולה או מצב שקשור לחיבור בין שני שרתים. הדגלים תורמים רבות בניתוח תעבורת הרשת ומראים מידע חיוני על האופן שבו הרשת פועלת.

SYN- דגל להקמת חיבור ברשת, מראה כוונה לפתוח חיבור TCP חדש. ריבוי חבילות SYN מעיד על חיבורי רשת חדשים או ניסיונות חיבור שנכשלו.

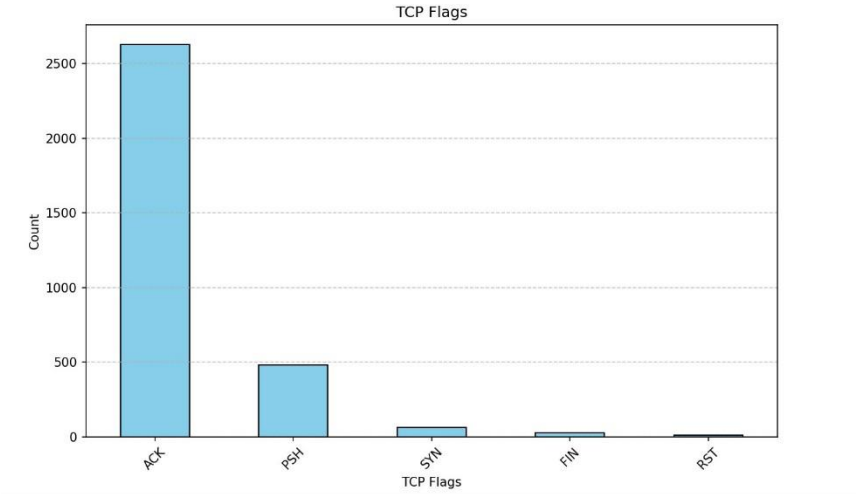
ACK- מאשר את קבלת החבילות שהתקבלו. כל חבילה TCP כוללת מספר רציף שמציין את מיקומה של החבילה והחבילה מכילה אישור על חבילה שנשלחה קודם. ריבוי חבילות ACK מעיד על חיבור פעיל ותקשורת דו-כיוונית.

FIN- משמש לסיום חיבור כאשר צד אחד רוצה לסיים את החיבור, הוא שולח חבילה עם דגל FIN. ריבוי חבילות FIN יכול להצביע על סיום חיבור, כך ניתן לזהות מתי יישום מסיים את השיחה.

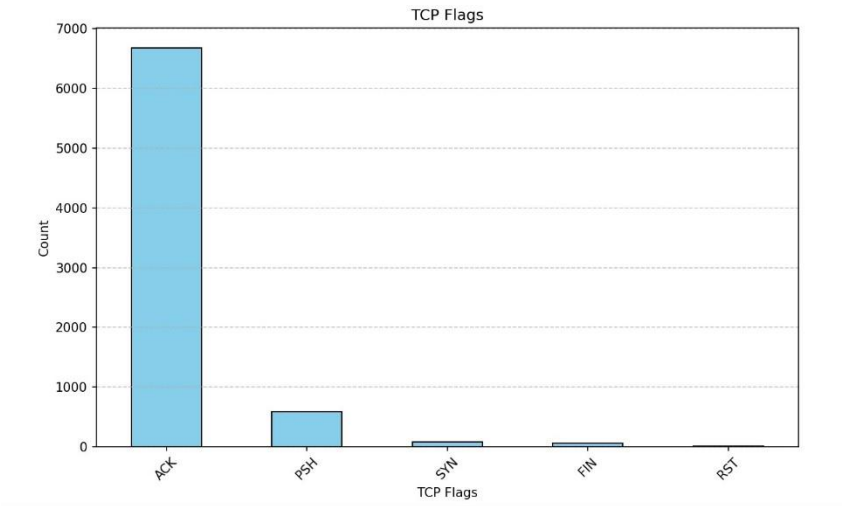
RST- משמש להפסיק חיבור TCP באופן מידי, במקרים של בעיות או כאשר יש צורך לסגור את החיבור בצורה מהירה ומידית. בדר"כ נשלחת במקרים בהם יש בעיות. ריבוי בRST מעיד על תקלות ברשת.

PSH- משמש כאשר הנתונים צריכים להיות מועברים באופן מידי. ברגע שהחבילה הגיעה, הנתונים צריכים להיבדק ולהיות מועברים מיד ליישום. ריבוי חבילות עם דגל זה יכול להעיד על אפליקציות שדורשות תגובה מיידית, כמו שיחות וידאו.

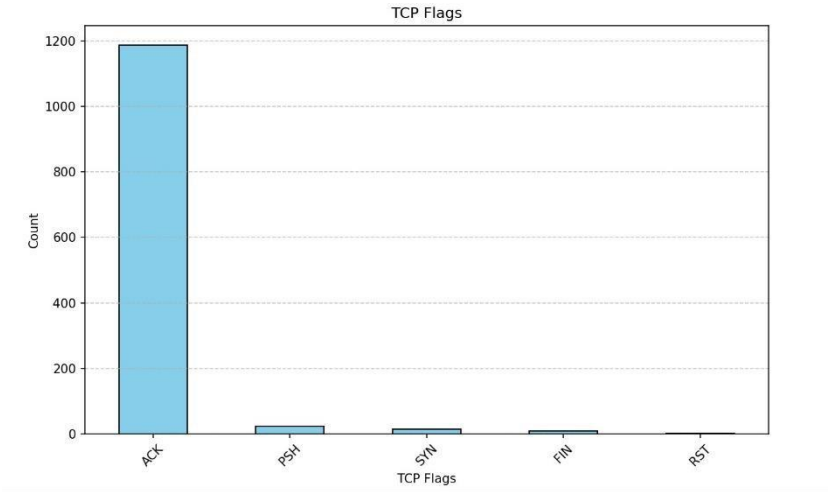
בגרפים הבאים נראה את הדגלים וכמה חבילות השתמשו בדגל (התפלגות הדגלים).



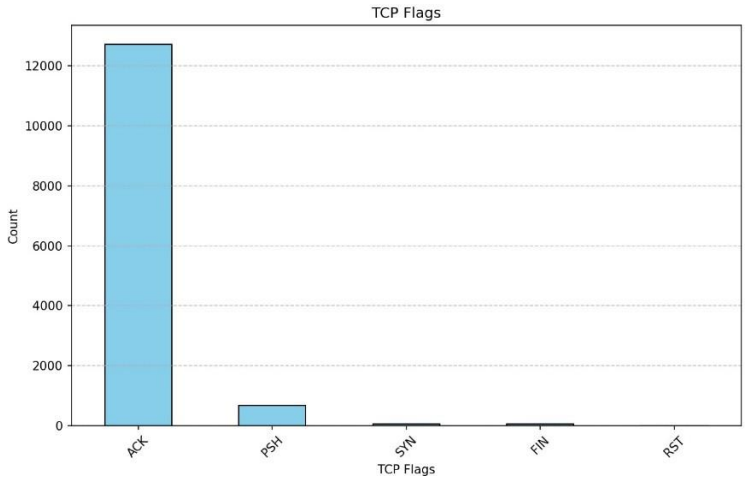
ניתן להבחין כי בגרף **chrome** יש שימוש רב ב- ack מעיד על חיבור יציב ומראה שאין הרבה נתונים שצריכים טיפול מידי



ניתן להבחין כי בגרף **edge** בדיוק כמו chrome יש שימוש רב ב- ack בהתאמה
psh- יש נתונים שצריכים טיפול מידי מעיד על שליחת בקשות לקבלת נתונים ואישורם
מראה על תעבורה אחידה כמו העלאת דפדפנים.

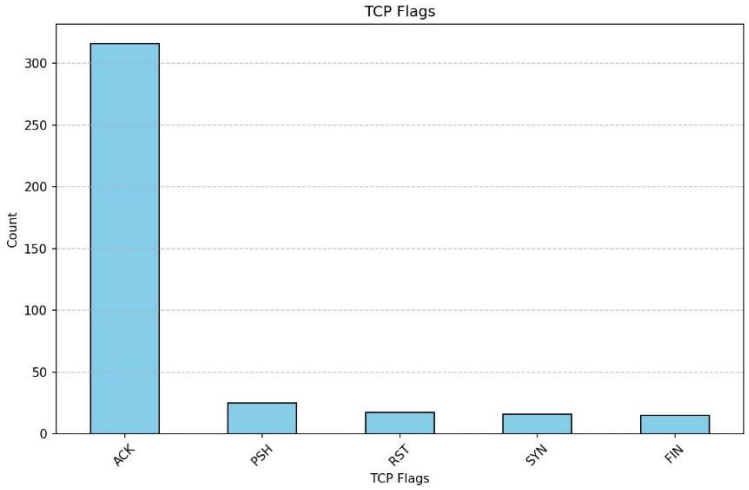


ניתן להבחין כי בגרף **spotify** יש שימוש רב ב- ack מעיד על חיבור יציב ומראה שאין הרבה נתונים שצריכים טיפול מידי
מעיד על שליחת בקשות לקבלת נתונים ואישורם
השילוב של udp עם tcp גורם לכך שיהיו הרבה אישורים ללא הרבה בקשות דחופות



ניתן להבחין כי בגרף **youtube** יש שימוש רב ב- ack ביוטיוב מעיד על זרימה רציפה של נתונים ויש הרבה חבילות שנשלחות.

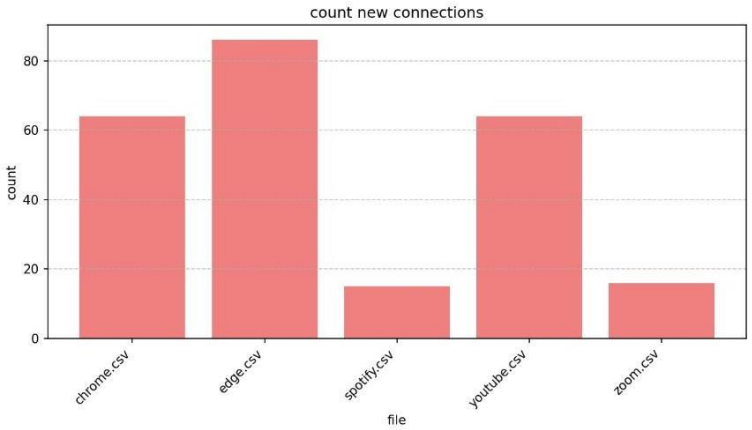
ניתן להבחין כי בגרף **zoom** יש שימוש ב- ack מעיד על חיבור יציב ואישור של tcp. (אך קצת ביחס לאחרים כי הוא מבוסס udp)
rst – ברוב המקרים החיבורים יישארו פתוחים.
psh- העברת נתונים דחופים כמו אודיו ווידאו בזמן אמת



<u>כמות חיבורים חדשים-</u>

תהליך בו שני שרתים או מערכות מתחברות ומבצעות תחילתו של חיבור תקשורת. במקרה שלנו, כל "לחיצת יד" מתארת חיבור חדש בין מקור ליעד.
כמות החיבורים מעידה על רציפות התעבורה.
ניתן להסיק כי כאשר רואים יותר חיבורים לשירות מסוים זה עשוי להעיד על האם הקשר רציף או לא.
כמות "לחיצות הידיים" עוזרת לנו להבין את דפוסי השימוש ברשת.

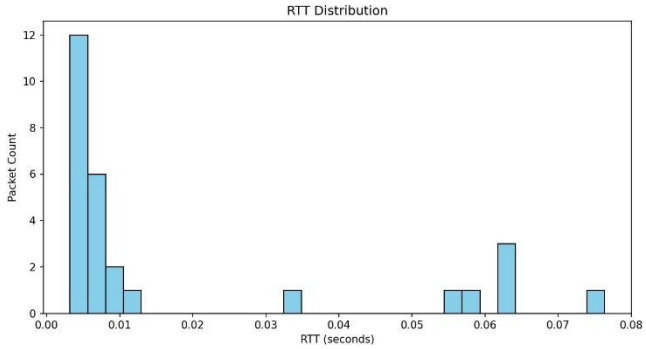
הגרף הבא מייצג את מספר החיבורים החדשים לכל יישום-



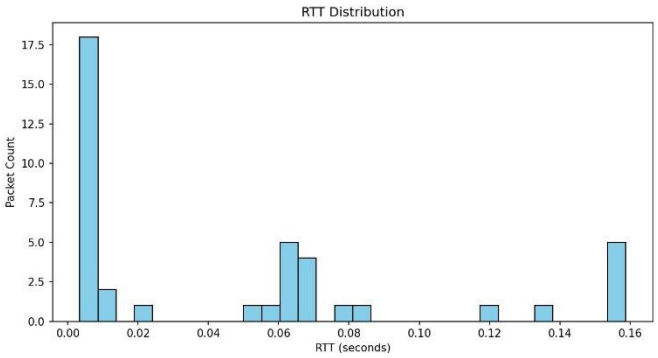
בגרף ניתן לראות כי edge הוא הכי גבוה — זאת מכיוון שהוא דפדפן וכל פעם שמשתמש פותח דף אינטרנט חדש או טוען תמונה וידיאו וכו' חדש ייווצר חיבור TCP חדש.
youtube הוא הבא אחריו כיוון שהוא גם כן שירות שמבוסס דפדפן .צפייה בסרטונים דורשת יצירת חיבורים חדשים כדי לשלוף את הנתונים מהשרתים.
chrome לאחר מכן, גם הוא כמו edge, כנראה שטוען פחות אובייקטים ולכן נמוך יותר מיוטיוב ואדג'.
אחריו בפער גדול מהקודם זה הזום, כל שיחה יש חיבורי tcp בהתחלה אך ברוב השיחות ישנה עדיפות להעברת נתונים באמצעות udp ולכן יש מספר קטן מאוד ביחס לקודמים.
לבסוף, יש את spotify שגם אצלו הtcp היא בעיקר לחיבור הראשוני או שליחה של פקודות או בקשות לשליפת מוזיקה, אך רוב התעבורה שלה נעשית באמצעות udp.

RTT-הוא הזמן שעובר מרגע שליחת חבילה SYN לשרת ועד שהתגובה ACK-SYN חוזרת מהשרת ללקוח .הוא משמש להערכת ביצועי הרשת והשהיות בתקשורת.

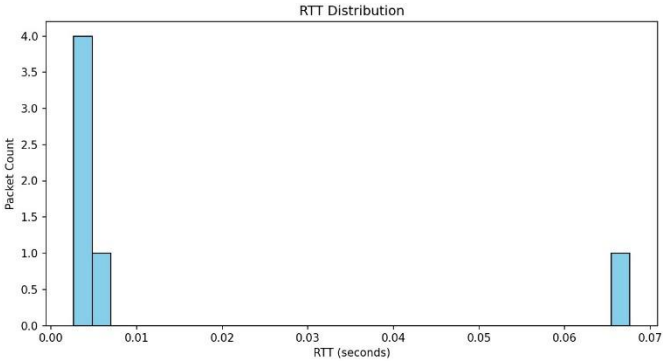
בגרפים הבאים נראה כמה חבילות היו בכל rtt שהופיעו כלומר, התפלגות הrtt-



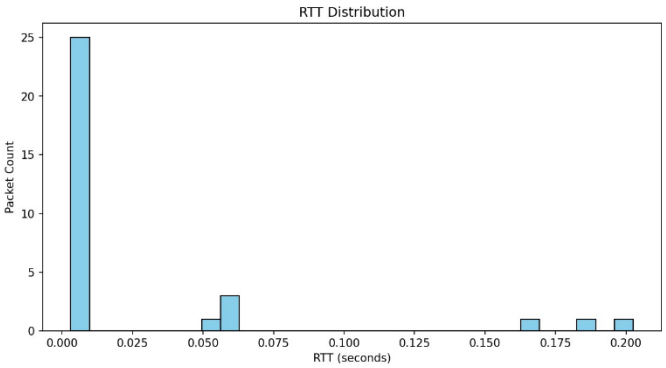
בגרף ניתן לראות **chrome** שבחבילות רבות זמני תגובה נמוכים מאוד, בסביבות 0.01 שניות ואף פחות. זה מעיד על חיבור רשת מהיר ויעיל ברוב המקרים. ישנן חבילות נוספות שנמצאות בערך ב0.06 שמעיד על שינויים ברשת. האטה יכולה לנבוע מחיבורי רשת רחוקים פיזית או הפרעות ברשת.



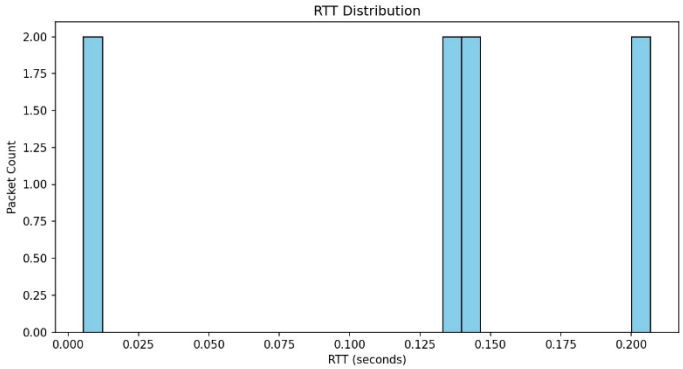
בגרף ניתן לראות **edge** שבחבילות רבות זמני תגובה נמוכים מאוד, בסביבות 0.01 שניות ואף פחות. זה מעיד על חיבור רשת מהיר ויעיל ברוב המקרים. ישנן חבילות נוספות שנמצאות בטווח בין 0.00 ל0.16 שמעיד על שינויים ברשת. האטה יכולה לנבוע מחיבורי רשת רחוקים פיזית או הפרעות ברשת.



בגרף ניתן לראות **spotify** שבחבילות רבות זמני תגובה נמוכים מאוד, בסביבות 0.00 שניות ואף פחות. זה מעיד על חיבור רשת מהיר ויעיל ברוב המקרים. יש את 0.07 שמעיד על שינוי או השהייה נקודתית ברשת. האטה יכולה לנבוע מחיבורי רשת רחוקים פיזית או הפרעות ברשת.



בגרף ניתן לראות **youtube** שבחבילות רבות זמני תגובה נמוכים מאוד, בסביבות 0.00 שניות ואף פחות. זה מעיד על חיבור רשת מהיר ויעיל ברוב המקרים. יש את 0.05 שמעיד על שינוי או השהייה נקודתית ברשת. האטה יכולה לנבוע מחיבורי רשת רחוקים פיזית או הפרעות ברשת. ישנה התפלגות לא אחידה עם הרבה חבילות סביב 0.00 וחריגות.



בגרף ניתן לראות **zoom** שבחבילות רבות זמני תגובה נמוכים מאוד, בסביבות 0.00 שניות. זה מעיד על חיבור רשת מהיר ויעיל ברוב המקרים אך, יש חריגות גדולות וזמני תגובה יותר גדולים בחבילות רבות מה שמעיד על עומס ושינויים ברשת.

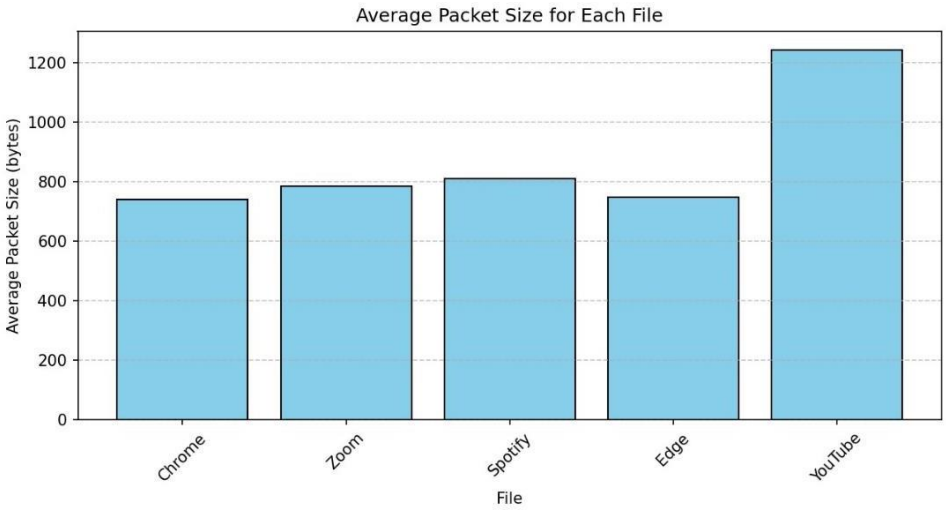
גודל חבילות ממוצע-

חבילה היא יחידת נתונים שנשלחת ברשת, והיא כוללת בתוכה מידע כמו כתובת יעד, מידע על ניהול החיבור, ונתונים שנמסרים.

גודל החבילה משפיע ברשת על יעילות העברת הנתונים- חבילות גדולות מידי עשויות לקחת יותר זמן להעברה בעיקר באיבוד חבילות בעוד שחבילות קטנות מידי יהיה צורך בדר"כ להעביר יותר חבילות וגורמות לניהול של יותר חיבורים ותהליכים וגורם להאטה ברשת.

בנוסף, ישנם פרוטוקולים שעשויים לשלוח חבילות בגודל שונה, והם עשויים לעזור להבין את סוגי התקשורת המתרחשים ברשת.

בגרף הבא נראה את התפלגות גודל ממוצע החבילות, הגרף תורם על הבנת פעילות הרשת



ניתן להבחין לפי הגרף youtube הוא עם גודל חבילות ממוצע הכי גדול מכיוון שוידאו דורש באפר (שדורש חבילות גדולות), איכות גבוהה של סרטונים ומכיל המון פיקסלים בכל פריים(סרטוני וידאו). הבא אחריו הוא ה spotify הוא משתמש בקבצי אודיו שהם קטנים יותר מוידאו אבל עדיין החבילות גדולות כיוון שיש לו באפר ובנוסף שולחים מקטעים של אודיו בבת אחת.

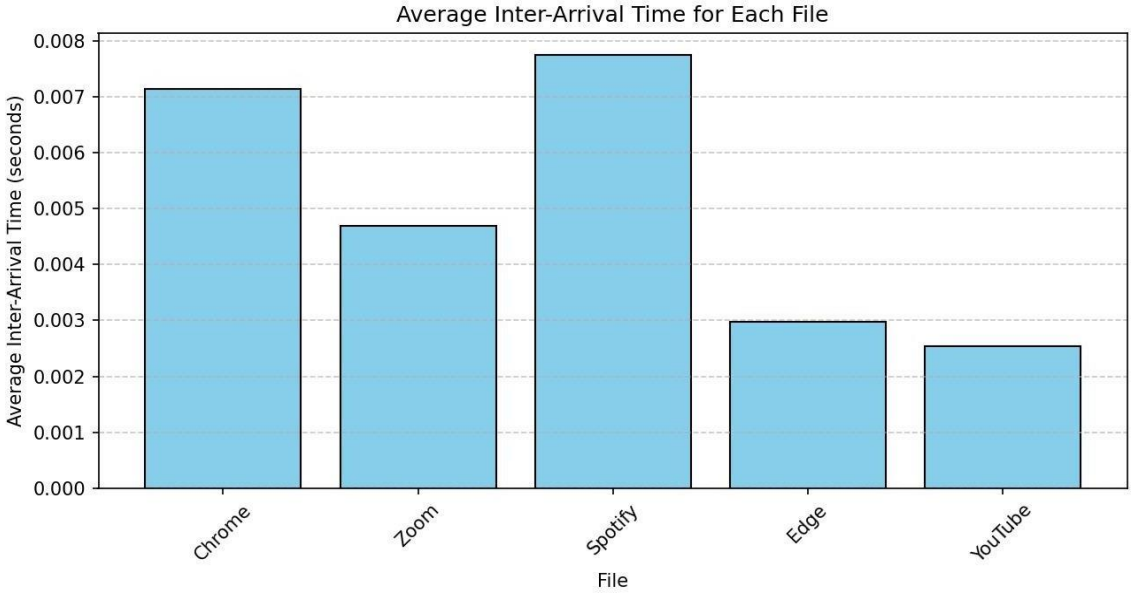
לאחר מכן, edge וchrome בדפדפנים שיש בהם תמונות אך גם קטעי וידאו (אך לא קטעים גדולים), כאשר יש הרבה אובייקטים וקישורים ייתכן ויהיו חבילות גדולות.

צמוד אליו יש את הzoom מכיוון שהוא משתמש בחבילות אודיו ווידאו שנשלחות בזמן אמת הוא לא שולח הכל ביחד, הוא מחלק אותם לקטעים קטנים כדי למנוע עיכובים כי כאשר החבילות קטנות יותר ניתן להעביר מידע במהירות גבוהה יותר ומונעים עיכובים בשיחה.

IPT (הזמן הממוצע בין שתי חבילות)-

הזמן שעובר בין חבילה אחת לשליחת החבילה הבאה ברשת
הוא תורם לנו בהבנת העומס ברשת — כאשר זמן ארוך מעיד על רשת ריקה או בעיות בקצב העברת נתונים בעוד שזמן קצר בין החבילות מעיד על רשת פעילה והעברה של הרבה נתונים בזמן קצר.
בנוסף, הזמן עלול להעיד על סוג התקשורת כאשר TCP היא תקשורת איטית בעוד שUDP היא תקשורת מהירה.
זמן יציב בין החבילות מעיד על תעבורה יציבה ורגילה אבל אם יש נבין כי יש בעיה ברשת. יתר על כן, אם יש זמן קצר מאוד בין חבילות זה עלול להעיד על התקפות.

בגרף הבא נראה את הזמן הממוצע שעובר בין 2 חבילות –



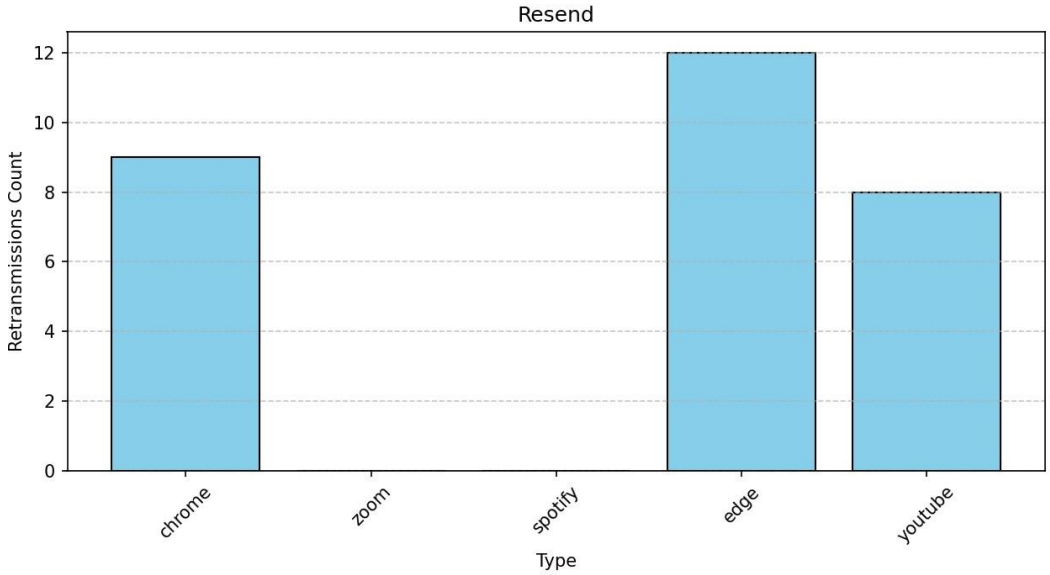
לspotify יש את הזמן הממוצע לשליחת 2 חבילות הכי גבוה מכיוון שהוא שולח חבילות גדולות ולכן הוא צריך לשלוח בפחות תדירות. יש לנו באפר שטוען את כל השיר מראש ולכן יש פחות חבילות שנשלחות.
הבא אחריו הוא chrome, הדפדפן שולח הרבה בקשות HTTP קטנות אבל לא זורמת מדיה רציפה. הוא תלוי בסוג הדף והאובייקטים שיש בו.
לאחר מכן יש את הzoom, מכיוון שהוא שולח חבילות בזמן אמת אז הוא שולח חבילות בתדירות גבוהה יחסית.
בedge פחתנו המון אובייקטים של תמונות וקישורים במקביל ולכן זה מוריד את זמן הממוצע בין שליחת החבילות. היינו מצפים שאם היו מתבצעות אותן פעולות בכרום הם היו בממוצע קרוב.
לבסוף עם זמן הממוצע הקצר ביותר זה youtube, הוא שולח וידאו רציף, עם חבילות שנשלחות בתדירות גבוהה מאוד. כאשר בכל חבילה שנשלחת (של וידיאו) ישנם נתונים רבים.

כמות חבילות חוזרות-

כאשר יש עיכוב ברשת או חבילות שנאבדו בדרך- כלומר חבילות שלא הגיעו ליעד בזמן בפרוטוקולים מסוימים כמו TCP שמבטיח אמינות נרצה לשדר מחדש את חבילות אלו.

כמות גבוהה של חבילות חוזרות יכולה להעיד על בעיות ברשת, כמו אובדן חבילות או בעיות בתקשורת. בעיות אלו עשויות להיגרם ע"י עומס יתר ברשת, או בעיות בשרתי הביניים.
מספר רב של חבילות חוזרות מסמן שאיכות החיבור לא טובה. בנוסף, החבילות החוזרות גורמות לעומס ברשת מכיוון שיש לשלוח מחדש את כל הנתונים.
אובדן החבילות בדר"כ נובע מבעיות תקשורת בין מחשבים ושיבושים או רעש ברשת.
גרף של חבילות חוזרות מאפשר להבין אם הרשת פועלת בצורה אופטימלית או אם יש בעיות שדורשות תיקון.

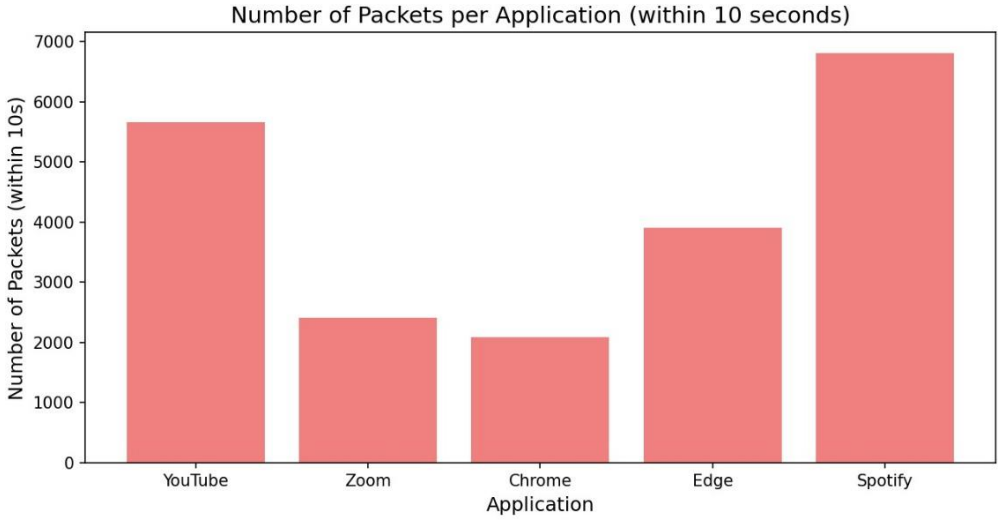
בגרף הבא נראה כמה חבילות חזרו בכל אפליקציה-



בגרף ניתן לראות כי edge הוא הכי גבוה — זאת מכיוון שהוא דפדפן וכל פעם שמשתמש פותח דף אינטרנט חדש או טוען תמונה וידיאו וכו' חדש ייווצר חיבור TCP חדש. כלומר, אם יש עיכוב ברשת או איבוד חבילות, ישלח מחדש חבילות שלא קיבלו אישור בזמן.
לאחר מכן chrome הוא גם כמו edge. יכול להיות שיש פחות חבילות חוזרות בגלל מנגנוני הזרימה של גוגל.
youtube הוא הבא אחריו, הוא מבוסס על סטרימינג ולכן פחות רגיש לאיבוד חבילות אך גם בו יש אובייקטים לטעינה ועוד.
ב spotify, zoom אין בכלל חבילות חוזרות כי הם משתמשים בעיקר בudp השימוש בtcp הוא רק בחיבור הראשוני ובדרכ לא נראה שם חבילות חוזרות.

גודל הזרימה (כמות חבילות עד 10 שניות) -

גודל הזרימה הוא מספר החבילות שהועברו עבור אפליקציה מסוימת, זרימה מתארת את התקשורת הכוללת שמתבצעת בין שני התקנים (כלומר כל חבילה נשלחת דרך הרשת כחלק מהזרימה).
הרבה חבילות- ככל הנראה מדובר בסרטונים או שליחה של חבילות בשידור חי וזה גורם לכך שיישלחו הרבה חבילות קטנות.
נוסף על כך, פרוטוקול TCP שולח הרבה חבילות קטנות כדי לשמור על אמינות.
מעט חבילות- תקשורת מבוססת פרוטוקול UDP יש בו זרימה רציפה של נתונים עם פחות חבילות, אבל כל חבילה מכילה הרבה נתונים. סיבה נוספת היא תקשורת קצרה (נגיד הקלטת התעבורה לזמן קצר).
רצינו להראות עד 10 שניות מכיוון שכל הקלטה הקלטנו כמות שונה של זמן וזה משפיע על כמות החבילות.



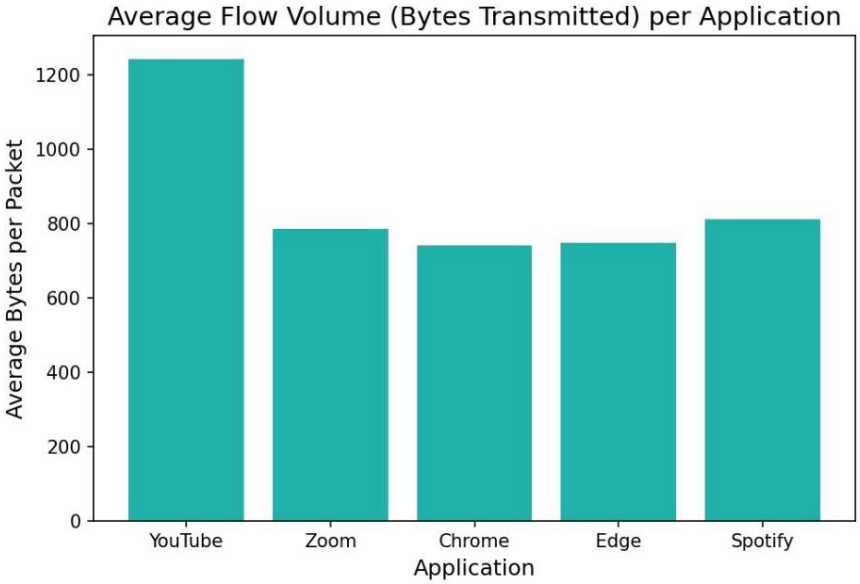
בגרף ניתן לראות כי spotify הוא הכי גבוה והבא אחריו youtube - מכיוון ששירותי אודיו ווידיאו שולחים ומקבלים כמויות גדולות של נתונים,אודיו נשלח בהרבה חבילות קטנות ווידאו מחייב העברה רציפה של חבילות נתונים, ולכן נרשמת כמות גדולה של חבילות בפרק זמן קצר.
הבא אחריו הוא edge, דפדפן שנכנסנו אליו להרבה קישורים ותמונות מעבר לכך שהוא משתמש בתדירות גבוהה בtcp.
אחריו הzoom, היינו מצפות שהוא יהיה גבוה מכיוון שהוא עובד על שידור חבילות בזמן אמת אבל שיחות וידאו מצריכות הרבה נתונים בזמן אמת, אך יש להם גם דחיסה טובה יחסית וההבדל בין זמני ההקלטה השפיעו על התוצאות. בנוסף יש לו שימוש רב בUDP ששולח בזרימה רציפה עם פחות חבילות ולכן זה מסביר את כמות החבילות שלו.
לאחר מכן chrome, היינו מצפוץ שיהיה כמו edge אבל להבנתנו הוא מבצע טעינה מוקדמת של דפים יותר מאשר Chrome מה שיכול לגרום להגדלה במספר החבילות הנשלחות.

נפח הזרימה (סך כל ממוצע הבתים שהועברו ב10 שניות הראשונות)-

נפח הזרימה מלמד את כמות הנתונים שהועברו. כאשר אנחנו מחשבים את נפח הזרימה, אנחנו מודדים את כמות המידע הכוללת שהועבר – כלומר, כמה בתים הועברו בפועל. ז"א שיכול להיות מצב שיש הרבה חבילות אבל החבילות קטנות ולכן הנפח יהיה קטן יותר.

מס רב של בתים שהועברו- יכול להצביע על סטרימינג וידאו שמחייב העברת נתונים רבים כדי לשלוח את הסרטונים והאודיו באיכות גבוהה. או חבילות שנשלחות בשידור חיי.

מס קטן של בתים שהועברו- מעיד על תעבורה מבוססת אודיו שבו הזרימה היא רציפה אך נתוני האודיו קטנים יותר בהשוואה לוידיאו. בנוסף יכול גם להיות קשור ל-גלישה כאשר אין תוכן כבד בעמודים הנגללים.



בגרף ניתן לראות כי youtube הוא הכי גבוה – זאת מכיוון שהוא סטרימינג וידאו וזה דורש הרבה נתונים כדי לשלוח את הוידאו באיכות גבוהה והאודיו, הזרימה מייצגת שידור רציף ונתונים רבים.

אחריו הzoom, מכיוון שהוא עובד על שידור חבילות בזמן אמת. שולח הרבה חבילות שכוללות וידאו דחוס, במקום שליחה של קובץ גדול בפחות חבילות. ולכן יש לו פחות חבילות אבל יש לו יחסית הרבה בתים בכל חבילה שהוא שולח כי זה אודיו וסטרימינג.

לאחר מכן יש את spotify, מכיוון שהוא טוען את הנתונים והבאפר אז הוא שולח קצת חבילות אבל בכל חבילה יש הרבה נתונים ומידע ולכן בכל חבילה הוא שולח הרבה בתים.

הבאים אחריו הוא chromeו edge, אתרים שנכנסנו אליהם להרבה קישורים ותמונות ולכן גם שולחים הרבה חבילות וגם עם נפח גדול כי יש כניסה להרבה קישורים וטעינת אובייקטים.

שאלה 4-

בחלק זה התבקשנו לכתוב מודל שאיתו התוקף יכול לסווג את התעבורה.
עבור כך השתמשנו ברעיון של המודל ניבוי KNN- שבו אנו מחשבים את ה"מרחק" מההקלטה שאנו רוצה לנבא מול ההקלטות שיש לנו באימון.
לשם כך, הקלטנו 15 הקלטות מכל סוג (chrome, edge, zoom, Spotify, youtube) , לקחנו לקחנו 25 הקלטות למבחן ואת השאר לאימון.
המודל עובד בצורה הבאה-
תחילה אנו בונים את האימון שאצלנו הוא בעצם קובץ csv שבו יש את כלל הקלטות האימון ועבורן את הממוצעים של כל קטגוריה (לפי הפרמטרים שהתוקף יכול לגשת אליהם).
בניית הcsv- עוברים על כל אחת מההקלטות בתקייה – ובכל הקלטה עוברים על כל החבילות ומחשבים את הממוצע עבור הקטגוריות הרצויות או את ההכי נפוצות (אם זה לא מספרי).

Label	MostCommonFlowID	MeanInterarrivalTime	MeanPacketSize
Chrome	61bbc5dbf72003f8f7f78729338d4ed0	0.004714038	656.5662372
Chrome	ecc0ffa70e625a96d7a72779af9443aa	0.000742827	910.8411077
Chrome	5403b060f73ac87b15ee62d67eca49ae	0.00293624	771.2704671
Chrome	d466afa21d931e46d8ccd12e905ceb28	0.00185843	886.4726137
Chrome	19e512881cb605bcbade20e596891abc	0.013156381	488.5
Chrome	c5bb29e60a106a3cc11f14ba78fc0674	0.001841469	716.8718822
Chrome	3c5c6fb3bdb27ee1a5877ef72330d13d	0.001828875	848.1445618
Chrome	687d6079472f5d2806a26bceafbaa098	0.001369215	792.4208397
Chrome	7436fb2fd391acf47855dcfb3f768905	0.000754616	630.3260372
Chrome	ecc0ffa70e625a96d7a72779af9443aa	0.000542293	707.7441539
Edge	1ccc948c0d090deb926424ef187ecb43	0.004256324	679.5771268
Edge	02b64aebe9f7cbd7078bd81bbb4d3985	0.000446575	599.8951464
Edge	ff61d5ed98ff5958ec92704c59616df0	0.001248877	787.4604995
Edge	886edf96fe660926ea710160ac98ab33	0.001228022	756.28259
Edge	886edf96fe660926ea710160ac98ab33	0.000811182	790.1691268
Edge	886edf96fe660926ea710160ac98ab33	0.001584317	898.0961937
Edge	886edf96fe660926ea710160ac98ab33	0.000897725	887.9993327
Edge	e36f812ec33443e1fe8fe1f94f422db3	0.002331735	754.4671666
Edge	329e9e6abdd8cda5c72be3a33ca2dae9	0.002649716	750.8143887

(תמונה להמחשה של הcsv)

שלב הטסט-
אנחנו ביצענו את הטסט עבור 100 דגימות (100 פעמים), בכל פעם נבחרת הקלטה רנדומלית מתיקיית ההקלטות של המבחן ואת התווית שלה אנו מנסים לנבא.
עבור כל הקלטה אנו עוברים על כל הקלטות האימון ומחשבים את "המרחק" שלה מכל אחת מהקלטות.
חישוב המרחק- עבור ערכים מספריים זהו ההפרש בין הערכים בערך מוחלט ועבור הflowid אנו ממירים אותו למספר (מהקסה דצימלי) ואז עובדים איתם כמספרים רגילים.
לאחר חישוב ההפרשים אנחנו מנרמלים את המרחקים כך-
מקובץ ה csv של האימון אנחנו לוקחות עבור כל קריטריון את המקסימלי והמינימלי. וכך עבור כל חישוב מרחק אנו עושות את הנרמול הבא-
(חישוב ההפרש – הגודל המינילמלי עבור אותו קריטריון) חלקי (גודל המקסימלי עבור אותו קרטיון- הגודל המינילמלי עבור אותו קריטריון)
לאחר מכן סוכמות את כל הקריטריונים יחד למספק שייצג לנו את ה"מרחק" של ההקלטת האימון מההקלטה לניבוי.
אחרי שיש לנו את כל המרחקים של הקלטות האימון מהקלטת הניבוי, אנו נבחר את הk (במקרה שלנו k=3) הכי קטנים (מה שמעיד על כך שהיו הכי קרובים להקלטה הרצויה) ונעשה "הצבעת רוב" ונחזיר את התווית השכיחה בין הנבחרים.

חלק א-

בחלק זה לתוקף הייתה גישה לשדות הבאים- גודל החבילה, חותמת זמן, וטאפל (כתובת IP מקורית, כתובת IP יעד, פורט מקור, פורט יעד).

ניתן לראות שהצלחת הניבוי עומדת על 87% ,

בצילום מסך זה ישנן רק חלק מ100 הניבויים שהיו.

```
File: edd12.pcapng -> Predicted Label: Edge
File: yoo11.pcapng -> Predicted Label: YouTube
File: s14.pcapng -> Predicted Label: Spotify
File: edd15.pcapng -> Predicted Label: Edge
File: yoo11.pcapng -> Predicted Label: YouTube
File: edd11.pcapng -> Predicted Label: Edge
File: edd14.pcapng -> Predicted Label: Edge
File: yoo15.pcapng -> Predicted Label: YouTube
File: zoo13.pcapng -> Predicted Label: Zoom
File: yoo12.pcapng -> Predicted Label: YouTube
File: s15.pcapng -> Predicted Label: Spotify
File: zoo13.pcapng -> Predicted Label: Zoom
File: yoo11.pcapng -> Predicted Label: YouTube
File: zoo15.pcapng -> Predicted Label: Zoom
File: yoo12.pcapng -> Predicted Label: YouTube
File: zoo12.pcapng -> Predicted Label: Zoom
File: chroo15.pcapng -> Predicted Label: Spotify
File: edd14.pcapng -> Predicted Label: Edge
correct predictions: 87 %

Process finished with exit code 0
```

חלק ב-

בחלק זה לתוקף הייתה גישה לשדות הבאים- גודל החבילה וחותמת זמן.

ניתן לראות שתוצאות הניבוי ירדו משמעותית מהחלק הקודם וזאת בגלל שהורדנו את נתוני flowid.

מה שמקשה יותר על זיהוי התווית.

הצלחת הניבוי עומדת על 57% .

```
File: chroo13.pcapng -> Predicted Label: Zoom
File: s13.pcapng -> Predicted Label: Zoom
File: edd15.pcapng -> Predicted Label: Chrome
File: s11.pcapng -> Predicted Label: Zoom
File: edd15.pcapng -> Predicted Label: Chrome
File: yoo13.pcapng -> Predicted Label: YouTube
File: s14.pcapng -> Predicted Label: Spotify
File: yoo15.pcapng -> Predicted Label: YouTube
File: edd11.pcapng -> Predicted Label: Edge
File: chroo13.pcapng -> Predicted Label: Zoom
File: yoo14.pcapng -> Predicted Label: YouTube
File: yoo12.pcapng -> Predicted Label: YouTube
File: chroo11.pcapng -> Predicted Label: Edge
File: zoo13.pcapng -> Predicted Label: Zoom
File: yoo13.pcapng -> Predicted Label: YouTube
File: s15.pcapng -> Predicted Label: Zoom
File: zoo13.pcapng -> Predicted Label: Zoom
File: s11.pcapng -> Predicted Label: Zoom
File: chroo14.pcapng -> Predicted Label: Chrome
Number of correct predictions: 57%

Process finished with exit code 0
```