פרויקט סופי רשתות

מגישים: 325942688_329091953_318817137_329079834

link to GitHub repository:

https://github.com/Shganga/Communication-Networks-Final-Project

קישור להקלטת הpcapng שלfirefox:

https://drive.google.com/file/d/1BRIkkZ22800wxJ-B0lN-8DMxZDCYcDCy/view?usp=sharing

חלק 1:

שאלה 1:

כאשר משתמש מדווח על העברה איטית של קובץ, יש מספר גורמים ברמת שכבת ההעברה שיכולים להשפיע על הביצועים. כדי לזהות את הסיבה לאיטיות, ניתן לבחון את הגורמים הבאים:

1. **עומס על הרשת**: כאשר יש עומס ברשת, יכול להיווצר עיכוב בהעברת המידע. עומס יכול להיגרם על ידי שימוש נרחב ברוחב הפס או שקיימות בעיות בחיבורים בין המכשירים.
2. **בעיות בפרוטוקול ה-TCP/UDP**: אם ההעברה מבוססת על TCP, ייתכן שהעיכוב נגרם כתוצאה מבעיות בתקשורת, כמו אובדן פאקטות או הצורך לשלוח מחדש פאקטות חסרות (retransmission), מה שגורם להאטה משמעותית בהעברה.
3. **גודל חלון ה- TCP (TCP Window Size)**: אם גודל חלון ה-TCP קטן מדי, זה יכול להגביל את כמות הנתונים שנשלחים בבת אחת ולגרום להאטה בקצב ההעברה.
4. **הגדרות MTU(Maximum Transmission Unit)**: גודל ה-MTU של הרשת עשוי להיות מוגדר בצורה שאינה אופטימלית, דבר שגורם לפיצול פאקטות (fragmentation), שמאט את ההעברה.
5. **בעיות עם פרוטוקולים בעלי overhead גבוה**: פרוטוקולים כמו HTTP עשויים להוסיף עומס נוסף על ההעברה בהשוואה לפרוטוקולים יעילים יותר כמו FTP או SFTP.

כדי לפתור את הבעיה, מומלץ לבצע ניתוח בעזרת כלים כמו Wireshark או tcpdump כדי לזהות פאקטות חסרות, בעיות בזמני ה-ACK או עומסים ברשת. כמו כן, יש לבדוק את גודל חלון ה-TCP, הגדרות ה-MTU, ולוודא שאין תהליכים נוספים שמשתמשים ברוחב הפס במקביל.

שאלה 2:

מנגנון בקרת הזרימה (Flow Control) ב-TCP נועד למנוע מצב שבו השולח שולח נתונים מהר מדי לעומת יכולת העיבוד של המקבל, דבר שעלול לגרום לאובדן נתונים או לעומס יתר. המנגנון מבוסס על גודל חלון ה- TCP (TCP Window Size), שבו המקבל מציין לשולח כמה נתונים הוא יכול לקלוט ולהעביר מבלי לגרום לעומס.

כאשר לשולח יש כוח עיבוד משמעותי יותר מזה של המקבל, מנגנון בקרת הזרימה יכול להשפיע באופן משמעותי על ביצועי ההעברה:

הגבלת מהירות השידור ב-TCP נובעת מהתאמת קצב השידור ליכולת העיבוד של המקבל, שכן חלון ה-TCP מתעדכן באופן דינמי ומגביל את כמות הנתונים שהשולח יכול לשלוח בכל רגע נתון. כאשר המקבל אינו מסוגל לעבד נתונים במהירות מספקת, השולח ייאלץ להמתין לעדכון מחלון ה-TCP לפני שיוכל להמשיך לשלוח, גם אם יש לו די משאבים להאיץ את ההעברה. כתוצאה מכך, עלול להיווצר מצב של ניצול לא אופטימלי של רוחב הפס (underutilization), שבו השולח מסוגל לשלוח בקצב גבוה, אך בשל מגבלות המקבל, חלק מרוחב הפס אינו מנוצל במלואו, מה שמוביל לבזבוז משאבים ברשת.

כלומר, כאשר לשולח יש כוח עיבוד גבוה משמעותית מזה של המקבל, מנגנון בקרת הזרימה של TCP עשוי להאט את הביצועים ולהוביל לניצול פחות אופטימלי של רוחב הפס הזמין.

שאלה 3:

במערכת רשת בה קיימות מספר דרכים בין המקור ליעד, לניהול המסלולים (Routing) יש תפקיד חשוב בהשפעה על ביצועי הרשת. הבחירה במסלול המתאים משפיעה על מהירות העברת הנתונים, יציבות החיבור, וה- QoS.

ההשפעה של בחירת המסלול על ביצועי הרשת תלויה במגוון גורמים:

1. **עומס על המסלול**: כל מסלול ברשת עשוי להיות בעומס שונה, כלומר יש מסלולים שמתפקדים טוב יותר בזמן נתון (למשל, פחות פקקים/עומס), בעוד אחרים עשויים להיות עמוסים יותר, מה שגורם לעיכובים ולירידה בביצועים.
   - **השפעה על ביצועים**: מסלול עמוס מדי יכול להוביל לעיכוב בהעברת הנתונים, אובדן פאקטות, ולירידה בקצב השידור. המערכת תעדיף מסלולים פחות עמוסים כדי לשמור על ביצועים אופטימליים.
2. **זמני השהייה - Latency**: זמני השהייה יכולים להשתנות בין המסלולים השונים, ויכולים להשפיע על זמן התגובה של היישומים. מסלול עם זמני השהייה גבוהים יוביל לעיכוב גדול יותר בהעברת הנתונים.
   - **השפעה על ביצועים**: אם האפליקציה רגישה לזמן (כמו שיחות VoIP או משחקים), בחירה במסלול עם זמני הגעה נמוכים יותר תהיה קריטית לשמירה על ביצועים גבוהים.
3. **רוחב פס - Bandwidth**: לכל מסלול יכול להיות רוחב פס שונה, כלומר יכולת העברת נתונים בקצב שונה. בחירה במסלול עם רוחב פס גבוה יותר תאפשר העברת נתונים מהירה יותר.
   - **השפעה על ביצועים**: רוחב פס מוגבל יכול להאט את העברת הנתונים ולגרום לעומס. המערכת תבחר מסלול עם רוחב פס אופטימלי עבור כמות הנתונים שברצונה להעביר.
4. **יציבות המסלול - Path Stability**: ישנם מסלולים שיכולים להיות יציבים יותר ועמידים בפני שיבושים או חיתוכים ברשת, בעוד אחרים עשויים להיות פחות יציבים ולעיתים לא זמינים.
   - **השפעה על ביצועים**: מסלול פחות יציב עשוי לגרום לאיבוד פאקטות או להפסקות בחיבור, ולכן יש להעדיף מסלול יציב כאשר יש יותר מסלול אחד.
5. **ניהול תורים ומסלולים Quality of Service - QoS**: כאשר ישנן דרישות שונות לאיכות השירות (כגון סדר עדיפות לפאקטות של יישומים קריטיים), המסלול שנבחר יכול להתחשב גם בקביעת עדיפויות אלו.
   - **השפעה על ביצועים**: במקרים בהם יש צורך בהעדפה של תעבורה מסוימת (כגון תעבורה של VoIP או ווידאו), ייתכן שהמערכת תבחר במסלול עם תעדוף גבוה יותר ליישומים אלו, כדי להבטיח חוויית שימוש חלקה עבורם.

כלומר, בחירת המסלול ברשת משפיעה ישירות על ביצועי הרשת, ויש לקחת בחשבון את העומס, זמני ההגעה, רוחב הפס, יציבות המסלול ודרישות איכות השירות על מנת להבטיח ביצועים אופטימליים.

שאלה 4:

MPTCP (Multipath TCP) - היא הרחבה של TCP שמאפשרת שימוש במספר נתיבים (Paths) במקביל להעברת חיבור רשת אחד. כלומר, במקום שחיבור TCP יזרום דרך נתיב יחיד, כמו ב-TCP רגיל, הוא יכול להתפצל למספר חיבורים משניים, וכך לשפר את הביצועים, האמינות ואת הניצול של רוחב הפס.

הגורמים המשפרים את הביצועים:

1. **ניצול מרבי של רוחב הפס** – כאשר למכשיר יש כמה חיבורים זמינים, הפרוטוקול מאפשר להשתמש בהם בו זמנית, ובכך להגדיל את קצב ההעברה ולהפחית זמני השהיה.

2. **עמידות גבוהה בפני שיבושים** – אם אחד המסלולים נקטע או חווה עומס, התקשורת יכולה להמשיך במסלול חלופי ללא הפרעה משמעותית, דבר המשפר את יציבות החיבור.

3. **פיזור עומסים ברשת** – היכולת לשלוח פאקטות נתונים במסלולים שונים מפחיתה את העומס על כל מסלול בודד, משפרת איזון עומסים ומונעת צווארי בקבוק.

שאלה 5:

במקרה שאנחנו מבחינים באובדן פאקטות גדול בין שני נתבים נפעל בשלבים אלה:

1. זיהוי הבעיה ואיסוף נתונים
נתחיל בבדיקת היקף הבעיה – האם היא מתרחשת באופן עקבי או לסירוגין אנחנו מחפשים דפוסים שיכולים לרמז על מקור הבעיה, כמו עומסים חריגים או צווארי בקבוק. בנוסף, אנחנו בודקים את מדדי הביצועים ברשת כדי לזהות אם יש שינויים חריגים בקצב התעבורה או בתגובת הנתבים.

2. בדיקת שכבת הרשת - Network Layer
ראשית נבדוק את מצב החומרה, נראה שאין כבלים פגומים, נתבים לא מחוברים היטב וכדומה. כעת, נמשיך בבדיקת עומס על הנתבים ונבדוק אם יש שגיאות או תורים ארוכים שמונעים מעבר חלק של נתונים. אנחנו גם בודקים את מסלול הניתוב כדי לזהות אם הנתבים מנתבים את התעבורה במסלול עמוס או לא יעיל, מה שעלול לגרום לאובדן פאקטות.

3. בדיקת שכבת התעבורה - Transport Layer
כאשר סיימנו לבדוק את שכבת הרשת נעבור לבדוק את שכבת התעבורה. אנחנו מוודאים שגודל חלון ה-TCP אינו קטן מדי, דבר שיכול לגרום להאטה בהעברת הנתונים (כמו שפירטנו בשאלה 1 סעיף 3). אנחנו מחפשים מקרים של שליחה חוזרת של נתונים, מה שיכול להצביע על בעיה בקצב השידור או על עיכובים ברשת. אם התעבורה מבוססת על UDP, אנחנו בודקים אם יש צורך בשיפור תהליכי תיקון שגיאות או התאמות אחרות.

4. יישום פתרונות בהתאם לממצאים
לאחר שאיתרנו את מקור הבעיה, נפעל בהתאם. אם מדובר בעומס יתר, נגדיר מדיניות לניהול עדיפויות בתעבורה או שנפצל עומסים בין חיבורים שונים. אם הבעיה נובעת מכשל

פיזי, נחליף את החומרה הפגומה. במקרים שבהם TCP גורם לבעיות, אנו נתאים את הפרמטרים הרלוונטיים כדי לשפר את ביצועי הפרוטוקול.

5. ניטור ובקרה מתמשכת

לאחר יישום הפתרונות, נעקוב אחר ביצועי הרשת כדי לוודא שהבעיה נפתרה. נבצע בדיקות עומס נוספות כדי לזהות אם עדיין מתרחש אובדן פאקטות, ונגדיר התראות שיתריעו על חריגות עתידיות (במידה ויהיו). אם נזהה סימנים לכך שהבעיה עלולה לחזור,נשקול לבצע אופטימיזציות נוספות ומעקב צמוד כדי לשמור על יציבות הרשת.

חלק 2

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

1. התרומה העיקרית של המאמר היא הצגת גישה חדשנית לסיווג תעבורת אינטרנט מוצפנת על ידי המרת נתוני זרימה בסיסיים לתמונות הנקראות "FlowPics". שיטה זו עושה שימוש רשתות עצביות קונבולוציוניות (CNNs) כדי לסווג קטגוריות של זרימה (כגון גלישה ברשת, צ'אט, וידאו וכו') ולזהות יישומים ספציפיים בדיוק גבוה. השיטה מאפשרת לכידת מידע לפי פרמטרים של זמן וגודל מתוך זרימות רשת, ללא תלות בתוכן הפאקטות, מה שהופך אותה ליעילה ומאובטחת יותר. המחברים מראים כי השיטה שלהם משיגה ביצועים טובים יותר בהשוואה לשיטות קודמות, במיוחד בסיווג תעבורה שעוברת דרך VPN או Tor, ובנוסף מפגינה עמידות בזיהוי יישומים חדשים שלא נכללו בשלב האימון.

2. פירוט המאפיינים המשמשים במאמר:

מאפיינים לא חדשניים

- **מאפיינים סטטיסטיים:** המאמר מציין שימוש במאפיינים סטטיסטיים הקשורים לגדלי חבילות ולזמני הגעה בין חבילות, אשר נפוצים בשיטות קודמות לסיווג תעבורה. אלה כוללים:
  - סטטיסטיקות של RTT.
  - סטטיסטיקות המבוססות על גדלים.
  - סטטיסטיקות של זמני הגעה בין חבילות.
  - תדירויות של גדלי חבילות.

מאפיינים חדשניים

- **ייצוג FlowPic:** החידוש המרכזי במאמר הוא יצירת FlowPics, שהן דיאגרמות דו-ממדיות הנבנות על בסיס גדלי החבילות וזמני ההגעה שלהן. טרנספורמציה זו מאפשרת למודל הסיווג לנצל טכניקות למידת עומק, ובפרט CNNs, כדי לסווג את התעבורה על פי הדפוסים הוויזואליים בתמונות אלו.
- **שימוש בחלון זמן קצר:** השיטה מסוגלת לסווג תעבורה על בסיס חלון זמן קצר של זרימה חד-כיוונית, במקום להסתמך על כל המפגש הדו-כיווני, מה שהופך אותה ליעילה יותר.
- **עצמאות מתוכן המטען:** שיטת הסיווג אינה תלויה בתוכן המטען של החבילות, ובכך שומרת על פרטיות המשתמשים ומפחיתה את דרישות האחסון.
- **יכולת זיהוי יישומים:** הגישה מפגינה יכולת לזהות יישומים שלא היו חלק ממערך האימון, מה שמצביע על יכולת הכללה שאינה אופיינית לשיטות מסורתיות של חילוץ מאפיינים.

לסיכום, המאפיינים הלא-חדשניים מבוססים בעיקר על ניתוח סטטיסטי של זרימות תעבורה, בעוד שהמאפיינים החדשניים כוללים את ייצוג ה-FlowPic, ההתמקדות בחלונות זמן קצרים, העצמאות מתוכן המטען, והיכולת לזהות יישומים שלא הופיעו בשלבי האימון.

3. התוצאות העיקריות של המחקר המוצג במאמר כוללות דיוקי סיווג שונים עבור קטגוריות תעבורת אינטרנט ויישומים שונים, כאשר נעשה שימוש בשיטת FlowPic המוצעת יחד עם CNNs. להלן כמה מהתוצאות המרכזיות כפי שסוכמו בטבלאות III ו-IV במאמר

סיכום תוצאות טבלה שלוש:

- Non-VPN Traffic Categorization: 85.0% accuracy
- VPN Traffic Categorization: 98.4% accuracy
- Tor Traffic Categorization: 67.8% accuracy
- Non-VPN Class vs. All: 97.0% accuracy (בממוצע)
- VPN Class vs. All: 99.7% accuracy (בממוצע)
- Tor Class vs. All: 85.7% accuracy (בממוצע)
- Encryption Techniques Classification: 88.4% accuracy
- Applications Identification: 99.7% accuracy

דיוק (accuracy) הסיווג לפי סוגי תעבורה (טבלה 4):

בטבלה הזאת רשומים אחוזי הדיוק של כל מחלקה אחרי הרצת מבחנים על דרכי הצפנה שונים

דרך הצפנה\מחלקה	Non-VPN	VPN	Tor
VoIP	99.6%	95.8%	52.1%
Video	99.9%	54%	55.3%
File Transfer	98.8%	65.1%	63.1%
Chat	96.2%	71.7%	85.8%
Browsing	90.6%	—	76.1%

תובנות מהתוצאות

- **דיוק גבוה עבור תעבורת VPN ותעבורת Non–VPN:** השיטה השיגה דיוקים גבוהים מאוד (מעל 98%) בסיווג קטגוריות תעבורה ובזיהוי יישומים בתעבורה מוצפנת ב-VPN, מה שמעיד על כך ששיטת FlowPic עמידה יחסית לאתגרי ההצפנה שמציבים שירותים כמו VPN.
- **אתגרים בסיווג תעבורת Tor:** דיוק הסיווג עבור תעבורת Tor היה נמוך משמעותית (67.8% בממוצע), מה שמצביע על כך ששיטות ההסוואה של Tor והמאפיינים הייחודיים של התעבורה בו מקשים על הסיווג.
- **יכולת הכללה ליישומים שלא נראו באימון:** היכולת של רשת ה-CNN לזהות יישומים שלא הופיעו בשלבי האימון (למשל, דיוק של 83.1% עבור יישומי וידאו לא מוכרים) מעידה על כך ששיטת FlowPic קולטת מאפיינים מהותיים של התעבורה, במקום "לזכור" חתימות של יישומים ספציפיים.
- **הסתמכות על מאפייני תעבורה:** השיטה מתבסס על מאפיינים הקשורים לזמן ולגודל של חבילות, ולא על תוכן המטען שלהן, מה שמעניק יתרון משמעותי בשמירה על פרטיות ובאפשרות לסיווג בזמן אמת.
- **השפעת טכניקות הצפנה:** התוצאות מראות כי לטכניקות הצפנה שונות יש השפעה ייחודית על מאפייני התעבורה, המשפיעה על דיוק הסיווג. לדוגמה, לתעבורה שאינה מוצפנת ב-VPN היה ביצוע סיווג טוב יותר בהשוואה ל-VPN ול-Tor, מה שמדגיש את ההשפעה של טכנולוגיות אלו על התנהגות התעבורה.

בסך הכול, המחקר מדגיש את היעילות של המרת נתוני זרימת הרשת לתמונות (FlowPics) לצורך סיווג תעבורה, תוך שימוש בטכניקות למידה לעומק להשגת דיוק גבוה עבור מגוון רחב של קטגוריות תעבורה ויישומים.

לצורך כתיבת סיכום הנתונים מטבלה 3 השתמשנו ב-CoralAI עם ההנחיה: "תסכם את תוצאות הטבלה 3". לאחר מכן, בדקנו את אמינות התשובה בכך שווידאנו שהנתונים שקיבלנו היו זהים למה שקראנו במאמר.

לצורך כתיבת סיכום הנתונים מטבלה 4 השתמשנו ב-CoralAI עם ההנחיה: "תסכם את תוצאות הטבלה 4". לאחר מכן, בדקנו את אמינות התשובה בכך שווידאנו שהנתונים שקיבלנו היו זהים למה שקראנו במאמר ולאחר קבלת הסיכום ווידוא האמינות של המידע הכנסנו את המידע לטבלה מסודרת לצורך סדר ואסתטיקה.

Early Traffic Classification With Encrypted ClientHello A Multi-Country Study

1. התרומות העיקריות של המאמר "Early Traffic Classification With Encrypted ClientHello: A Multi-Country Study" הן:

איסוף נתונים: החוקרים אספו מערך נתונים מגוון על גבי מספר רב של מדינות, המכיל יותר מ-600,000 זרמי TLS המחולקים ל-19 מחלקות תנועה. מערך נתונים זה בולט במגוון הפרוטוקולים, הזמן, הגיאוגרפיה והמכשירים שיוצרים את זרמי ה-TLS, מה שהופך אותו לאחד ממערכי הנתונים הפתוחים הגדולים והמפורטים ביותר לסיווג תנועה.

פיתוח hRFTC: החוקרים פיתחו אלגוריתם חדש לסיווג תנועה מוקדמת בשם hybrid Random Forest Traffic Classifier. אלגוריתם זה עושה שימוש במטענים בלתי מוצפנים של TLS handshake , בסדרות זמן מבוססות זרם ובסטטיסטיקות של גודל פאקטות כתכונות עצמאיות לסיווג, ומציג ביצועים עדיפים בהשוואה לאלגוריתמי סיווג מתקדמים שקיימים.

הערכת אלגוריתמים קיימים: המאמר מעריך את ביצועי אלגוריתמי סיווג תנועה מתקדמים על מערך הנתונים שנאסף, וחושף כי אלגוריתמים אחרים שנמצאים בשימוש רב יותר המתבססים אך ורק על תכונות TLS מציגים ביצועים נמוכים יותר בהקשר של Encrypted ClientHello - ECH.

תובנות לגבי הכללה: החוקרים בוחנים את יכולות ההכללה של האלגוריתם שלהם, ומראים כי גם hRFTC מסוגל להציג ביצועים טובים כאשר הוא מאומן על תת-קבוצה של מערך הנתונים. הם גם מציינים כי יש צורך לאמן מחדש את המסווגים כאשר הם מיושמים באזורים גיאוגרפיים שונים, בשל שונות בתבניות התנועה.

בגדול המאמר מתמודד עם אתגרים משמעותיים בסיווג תנועה מוקדם תחת מגבלות ECH ומציג פתרון טוב עם hRFTC.

2. המאמר דן בתכונות תנועה שונות המשמשות לסיווג תנועה מוקדם (eTC) עם התמקדות בזרמי TLS, במיוחד בהקשר של ECH. התכונות מתחלקות לשני סוגים עיקריים: תכונות מבוססות חבילה ותכונות מבוססות זרם.

תכונות מבוססות חבילה:

תכונות אלו כוללות את המטען של הודעות לחיצת היד TLS, המשמש כתכונות סיווג עצמאיות. המאמר מציין במיוחד את השימוש ב:

- פרמטרי TLS בלתי מוצפנים, כגון Cipher Suite ו-Key Share Group.
- הגדלים והסוגים של הרחבות בהודעות לחיצת היד של TLS.

תכונות מבוססות זרם:

תכונות אלו נגזרות מרצף גדלי החבילות (PSs) וזמני ההפרש בין חבילות (IPTs) וכוללות:

- מאפיינים סטטיסטיים של הזרם, כגון גדלי חבילות ועיכובים בין חבילות.
- סטטיסטיקות שונות על פני החבילות הראשונות עד לחבילת ה-Downlink הראשונה המכילה נתוני יישום. זה כולל מאפיינים כמו מספר החבילות, גודל החבילה הממוצע והשונות בגדלי החבילות.

התכונות החדשניות שהוצגו במאמר, במיוחד דרך האלגוריתם hRFTC כוללות:

- וקטור מטען ייחודי המשולב מחדש, אשר משלב את פרמטרי לחיצת היד הבלתי מוצפנים של TLS לתוך וקטור בייטים בגודל קבוע. הדבר מאפשר לclassifier לנצל בצורה יעילה יותר את מבנה הודעות ה-handshake.
- התחשבות בסדרות זמן מבוססות זרם ובסטטיסטיקות של גדלי חבילות כתכונות סיווג עצמאיות, אשר לא נחקרו (בהרחבה) בעבר בהקשר של סיווג תנועה עם ECH.

השילוב של תכונות אלו, ובעיקר הגישה החדשנית לניצול מאפיינים מבוססי חבילה ומבוססי זרם באופן היברידי, מהווה תרומה משמעותית לתחום סיווג התנועה בסביבות מוצפנות.

3. התוצאות המרכזיות של המאמר מתמקדות בביצועי האלגוריתם hRFTC בהשוואה לאלגוריתמי סיווג אחרים בהקשר של סיווג תנועה מוקדם (eTC) עם ECH.

תוצאות עיקריות:

ביצועי הסיווג:

- hRFTC השיג *ציון F-מקרו של עד 94.6% על מערך הנתונים שנאסף, והציג ביצועים עדיפים משמעותית בהשוואה לאלגוריתמים מתקדמים אחרים, אשר הגיעו לציון F של 38.4% בלבד כאשר הסתמכו אך ורק על תכונות TLS.
- הגישה ההיברידית שמשלבת בין נתוני TLS לא מוצפנים לבין תכונות סטטיסטיות מתוך זרמי תעבורה הובילה לשיפור בביצועים.

***ציון F-מקרו** (Macro F-score) הוא מדד ביצועים המשמש להערכת המודלים של למידת מכונה, במיוחד בסיווג עם מספר קטגוריות. הציון מחושב על ידי חישוב ציון ה- F לכל אחת מהקטגוריות בנפרד, ולאחר מכן חישוב ממוצע הציונים הללו.

השפעת תכונות TLS ותכונות סטטיסטיות:

- המחקר הראה כי בעוד שתכונות מבוססות-חבילה (כגון פרמטרי TLS) עדיין בעלות חשיבות מסוימת, תכונות מבוססות-זרם (כגון גדלי חבילות וזמני הפרש בין חבילות) היו בעלות השפעה משמעותית יותר על איכות הסיווג.
- ערך מדד ההבחנה הממוצע על פני כל מערך הנתונים עמד על 0.1753 בלבד, מה שמעיד על הבחנה מוגבלת בין שירותים שונים בהתבסס על הגדרות TLS.

יכולת הכללה:

- ה- hRFTC הפגין יכולות הכללה יוצאות דופן: גם כאשר אומן על 10% בלבד ממערך הנתונים, הוא השיג ביצועים דומים לאלגוריתמים אחרים שאומנו על מערכי נתונים גדולים יותר.
- עם זאת, ביצועי האלגוריתם ירדו כאשר נבדק על תנועה מאזורים גיאוגרפיים שלא נראו במהלך האימון, מה שמדגיש את הצורך באימון מותאם למיקום.

תובנות מהתוצאות:

- הממצאים מצביעים על כך ששיטות סיווג מסורתיות המבוססות-חבילה מתקשות בסביבות בהן נעשה שימוש ב-ECH, בעיקר בשל הדמיון בהגדרות TLS בין שירותים שונים.
- הגישה ההיברידית שננקטה ב-hRFTC מנצלת בצורה יעילה הן נתונים בלתי מוצפנים והן נתונים מוצפנים, ומאפשרת סיווג חזק יותר גם בתנאים בהם תעבורה מוצפנת היא הדומיננטית.
- הממצאים מדגישים את החשיבות של שילוב תכונות מבוססות-זרם לצד תכונות מבוססות-חבילה כדי לשפר את האפקטיביות של eTC, במיוחד ברשתות מודרניות בהן ההצפנה נפוצה.
- הירידה המשמעותית בדיוק הסיווג כאשר נתקלים בתנועה מאזורים גיאוגרפיים חדשים מצביעה על כך שתבניות תנועה יכולות להשתנות באופן משמעותי בהתאם לתנאי הרשת המקומיים ולשיטות הנהוגות, מה שמחייב מודלים אדפטיביים לסיווג תנועה.

תובנות אלו מדגישות את האתגרים המתפתחים בסיווג תנועה בשל ההצפנה ואת הצורך בגישות חדשניות המשלבות מקורות נתונים מרובים לניהול רשתות יעיל.

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

1. התוצאות המרכזיות של המאמר הן:

זיהוי מערכת ההפעלה, הדפדפן והיישום של המשתמש: זהו המחקר הראשון שמדגים כיצד ניתן לזהות את מערכת ההפעלה, הדפדפן והאפליקציה מתוך תעבורת HTTPS. החוקרים מנצלים דפוסי תנועה ומציגים תכונות חדשות המדגישות את ההתנהגות המקוטעת של דפדפנים ואת מאפייני SSL. שימוש בתכונות הבסיס בלבד מוביל לדיוק סיווג של 93.51%, בעוד ששילוב שלהן עם התכונות החדשות משיג דיוק של 96.06%.

מערך נתונים מקיף: המאמר מספק מערך נתונים מקיף הכולל יותר מ-20,000 סשנים מתויגים. מערך הנתונים כולל מערכות הפעלה שונות (Windows, Linux-Ubuntu ו-OSX), דפדפנים (Chrome, Internet Explorer, Firefox ו-Safari) ויישומים (YouTube, Facebook ו-Twitter).

2. המאמר עושה שימוש בשני סוגים של תכונות תעבורה לזיהוי מערכת ההפעלה, הדפדפן והאפליקציה מתוך תעבורת HTTPS.

תכונות בסיסיות: אלו תכונות נפוצות המשמשות בשיטות סיווג תעבורה רבות וכוללות:

- סך כל הבייטים שנשלחו קדימה
- סטטיסטיקות גודל חבילה (מינימום, מקסימום, ממוצע, שונות)
- הפרשי זמני הגעה בין חבילות עבור זרמים קדמיים ואחוריים
- תכונות TCP כגון גודל חלון התחלתי ופקטור שינוי קנה המידה של החלון
- תכונות SSL כולל שיטות דחיסה ושיטות הצפנה
- מספר חבילות והתפרצויות עבור הזרמים הקדמיים והאחוריים

תכונות חדשות: המאמר מציג תכונות חדשות המנצלות את ההתנהגות המקוטעת של דפדפנים ואת מאפייני SSL. תכונות אלו כוללות:

- אורך מזהה סשן (SSL Session ID)
- שיא תפוקה מרבי קדמי
- ממוצע, סטיית תקן והפרש של זמני הגעה מינימליים ומקסימליים
- מספר חבילות keep-alive
- גרסת SSL קדמית

תכונות חדשניות אלו נועדו לשפר את היכולת להבחין בין מערכות הפעלה ודפדפנים שונים על ידי ניתוח דפוסי התעבורה הייחודיים שהם מייצרים.

3. תוצאות עיקריות:

דיוק בהכרת סוגי התעבורה:
דיוק ההכרה עבור הצירוף <מערכת הפעלה, דפדפן, יישום> היה גבוה, עם דיוק של 96.06% כאשר משלבים את התכונות הבסיסיות והחדשות. (התכונות הבסיסיות בלבד נתנו דיוק של 93.51%).

קבוצות תכונות:
המחקר השווה בין שלוש קבוצות תכונות שונות:

- תכונות בסיסיות
- תכונות חדשות
- שילוב של תכונות בסיסיות וחדשות

מטריצות בלבול:
מטריצות הבלבול המוצגות במאמר מראות שהמיון כמעט מושלם עבור רוב הצמדים, כאשר רוב הניבויים תואמים באופן קרוב לתוויות האמיתיות. עם זאת, היו כמה יוצאים מן הכלל, במיוחד עבור הקטגוריות "לא ידועות", שעשויות להיות מיון תקני שהמודל לא הצליח לאמת.

הנה הדימויים הרלוונטיים הממחישים את התוצאות הללו:

- **תוצאות דיוק הצירוף:** שילוב של תכונות בסיסיות וחדשות השיג דיוק של 96.06%. דיוק עם תכונות בסיסיות בלבד היה 93.51%.

	Windows IExplorer Twitter	Ubuntu Firefox Google-Background	Windows Non-Browser Microsoft-Background	Windows Chrome Twitter	Windows Firefox Twitter	OSX Safari Google-Background	OSX Safari Youtube	Ubuntu Chrome Unknown	Windows Chrome Google-Background	Ubuntu Firefox Twitter	OSX Safari Unknown	Ubuntu Firefox Unknown	Ubuntu Chrome Google-Background	Ubuntu Chrome Twitter	Windows Firefox Google-Background	OSX Safari Twitter	Ubuntu Firefox Youtube	Windows Non-Browser Teamviewer	Ubuntu Chrome Youtube	Windows Non-Browser Dropbox	Windows Chrome Unknown	Ubuntu Chrome Facebook	Windows Firefox Unknown	Ubuntu Firefox Facebook	OSX Chrome Twitter	Windows IExplorer Unknown	Ubuntu Non-Browser Microsoft-Background	Windows IExplorer Google-Background	OSX Chrome Google-Background	OSX 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	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	.01	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	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	.01	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	0	.02	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	.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OSX Safari Youtube	0	0	0	0	0	.02	.97	.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Unknown	0	0	0	0	0	0	0	.84	0	0	0	0	.07	.04	0	0	0	0	.01	0	0	.03	0	0	0	0	0	0	0	0
Windows Chrome Google-Background	0	0	.01	.03	0	0	0	0	.94	0	0	0	0	0	.02	0	0	0	0	0	.01	0	0	0	0	0	0	0	0	0
Ubuntu Firefox Twitter	0	0	0	0	0	0	0	0	0	.95	0	.03	0	0	0	0	.01	0	0	0	0	0	0	0	0	0	0	0	0	0
OSX Safari Unknown	0	0	0	0	0	.06	.01	0	0	0	.91	0	0	0	0	.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Firefox Unknown	0	.02	0	0	0	0	0	0	0	.08	0	.87	0	0	0	0	.01	0	0	0	0	0	0	.03	0	0	0	0	0	0
Ubuntu Chrome Google-Background	0	.07	0	0	0	0	0	.18	0	0	0	0	.73	0	0	0	0	0	.02	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Twitter	0	.02	0	0	0	0	0	.08	0	0	0	0	.03	.84	0	0	0	0	.01	0	0	.01	0	0	0	0	0	0	0	0
Windows Firefox Google-Background	0	0	0	.01	0	0	0	0	.01	0	0	0	0	0	.97	0	0	0	0	0	0	0	.01	0	0	0	0	0	0	0
OSX Safari Twitter	0	0	0	0	0	0	.06	0	0	0	.03	0	0	0	0	.91	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Firefox Youtube	0	.02	0	0	0	0	0	0	0	.02	0	.02	0	0	0	0	.93	0	0	0	0	0	0	0	0	0	0	0	0	0
Windows Non-Browser Teamviewer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Youtube	0	0	0	0	0	0	0	.07	0	0	0	0	.13	.04	0	0	0	0	.74	0	0	.02	0	0	0	0	0	0	0	0
Windows Non-Browser Dropbox	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Windows Chrome Unknown	0	0	.02	.09	0	0	0	0	.02	0	0	0	0	0	0	0	0	0	0	0	.86	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Facebook	0	0	0	0	0	0	0	.3	0	0	0	0	.04	0	0	0	0	0	0	0	0	.67	0	0	0	0	0	0	0	0
Windows Firefox Unknown	0	0	.06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.94	0	0	0	0	0	0	0
Ubuntu Firefox Facebook	0	.06	0	0	0	0	0	0	0	.11	0	.28	0	0	0	0	0	0	0	0	0	0	0	.56	0	0	0	0	0	0
OSX Chrome Twitter	0	0	0	0	0	0	0	.13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.75	0	0	0	.06	.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	0	0	0	0	.29	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	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	0	0	0	0	0	0	0	0	1	0	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	0	0	0	0	0	0	0	0	1	0
OSX Chrome Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

(a) Tuple Confusion Matrix

- **תוצאות דיוק מערכת הפעלה:** הדיוק בהכרת מערכת ההפעלה בעזרת קבוצות תכונות שונות היה גם הוא גבוה, כאשר הצירוף של התכונות נתן את התוצאות הטובות ביותר.

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

b) OS Confusion Matrix

- **תוצאות דיוק דפדפן:** בדומה לתוצאות מערכת ההפעלה, דיוק בהכרת הדפדפן השתפר כאשר השתמשו בתכונות החדשות, עם הצירוף של התכונות המשיג את הדיוק הגבוה ביותר.

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

- **תוצאות דיוק יישום:** תוצאות ההכרה ביישום הלכו באותו כיוון, עם דיוק גבוה שהושג באמצעות שילוב של תכונות בסיסיות וחדשות.

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	0	.04	0	.29
Google-background	0	0	.95	0	0	.01	.01	.03
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 מתוך התוצאות:

- **פוטנציאל גבוה בהכרת סוגי התעבורה:**
התוצאות מאשרות את הפוטנציאל של טכניקות ניתוח תעבורה בהכרת תעבורת רשת מוצפנת למרות האתגרים שמציבה ההצפנה. הדבר מצביע על כך שגם נתונים מוצפנים יכולים לחשוף מידע חשוב אודות המערכת של המשתמש והשימוש ביישומים.
- **חשיבות התכונות:**
השיפור המשמעותי בדיוק כאשר נוספו תכונות חדשות מדגיש את חשיבות בחירת התכונות במשימות למידת מכונה. התכונות החדשות, במיוחד אלו שמנצלות את ההתנהגות הפורצת של דפדפנים ומאפייני SSL, משחקות תפקיד קריטי בשיפור ביצועי המודל.
- **השלכות מעשיות:**
היכולת להכיר מערכות הפעלה, דפדפנים ויישומים מתוך תעבורה מוצפנת יש לה השלכות

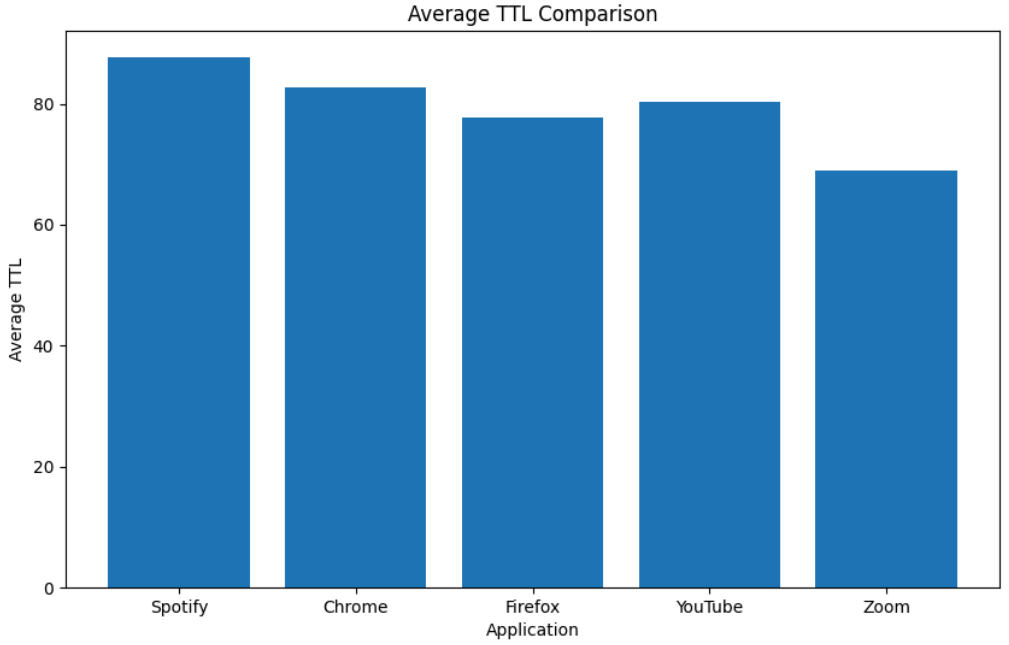
על פרטיות ואבטחה. הדבר מציע כי אויבים פאסיביים עשויים לנצל את המידע הזה למטרות זדוניות, ומדגיש את הצורך באמצעי פרטיות משופרים.

ממצאים אלה תורמים לשיח המתמשך בנושא פרטיות בהקשר של תקשורת מוצפנת ומדגישים תחומים למחקר נוסף, במיוחד במטרת פיתוח אמצעי נגד נגד טכניקות זיהוי כאלו.

חלק 3

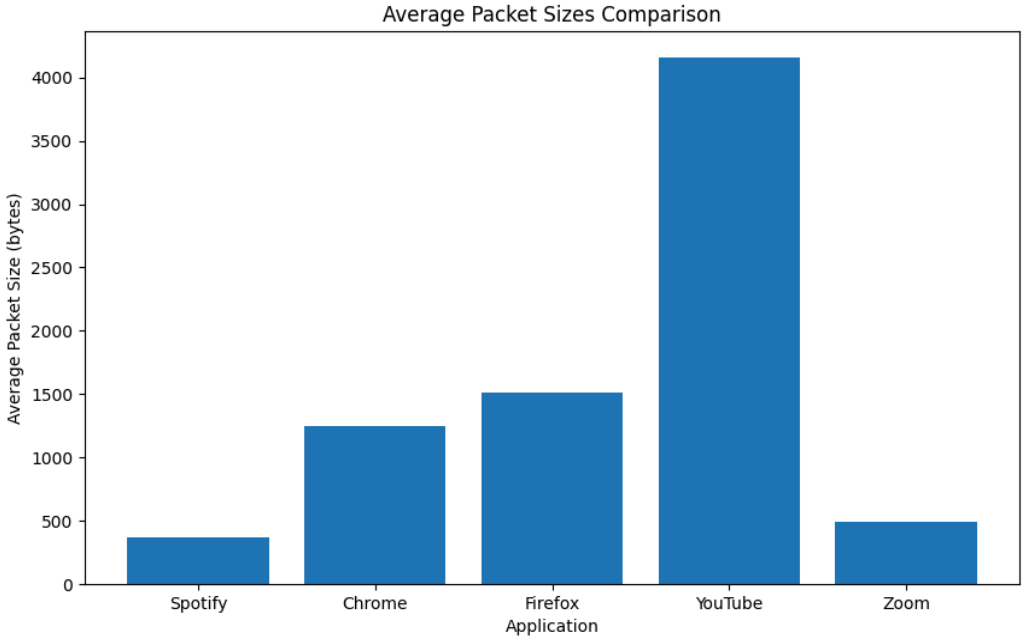
(סעיף 1 ו-2 שהם הקלטת התעבורה וכתיבת קוד ניתוח בתעבורה נמצאים ב-Git repository)
(בסוף חלק 3 מצורפות הספריות בהן השתמשנו לחלק זה)

3. השוואת זמן החיים (time to live) של הפאקטות באפליקציות השונות:



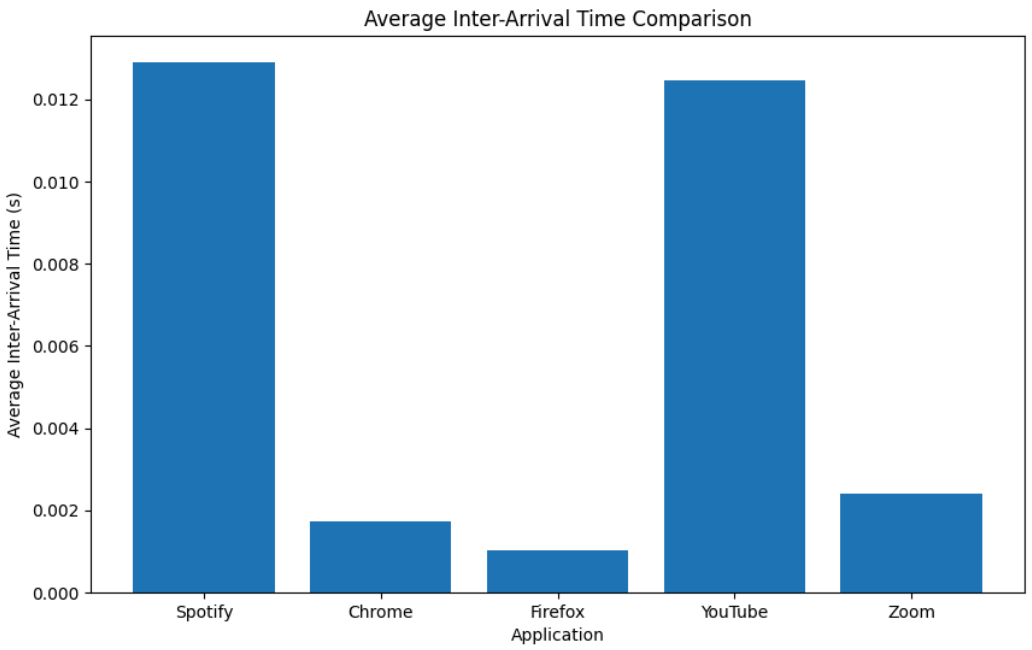
לפי הגרף ניתן לראות כי זמן החיים של כל האפליקציות יחסית דומה. לכל האפליקציות יש בין 70-80 קפיצות לפני שימחקו, הכמות הזו נחשבת טווח חיים יחסית קצר. TTL נמוך מאפשר עדכניות ומהירות תגובה. (ככל שהזמן חיים יהיה יותר נמוך כך נקבל פידבק על חבילות שלא הגיעו מהר יותר).
לכל אפליקציה יש סיבה מדוע היא משתמשת בזמן חיים קצר:
בגלל שיוטיוב מבוססת על רשת שרתים גלובלית, הוא משתמש בגישה זו. כך הוא מבטיח שהלקוח יקבל את בקשתו מהשרת הקרוב והמהיר ביותר (מסלול מהיר).
לספוטיפיי והזום יש סיבה דומה. שתי האפליקציות הללו רוצות חיבור מהיר ולהימנע מניתוקים. זמן חיים קצר מסייע להשיג מטרה זו.
הדפדפנים מפעילים שאילתות DNS רבות, כלומר הם ממירים שמות טקסט לכתובת IP. זמן חיים נמוך מוודא שהלקוח יקבל את הכתובת העדכנית ביותר של האתרים אותו חיפש.

השוואת גודל חבילה ממוצע של האפליקציות:



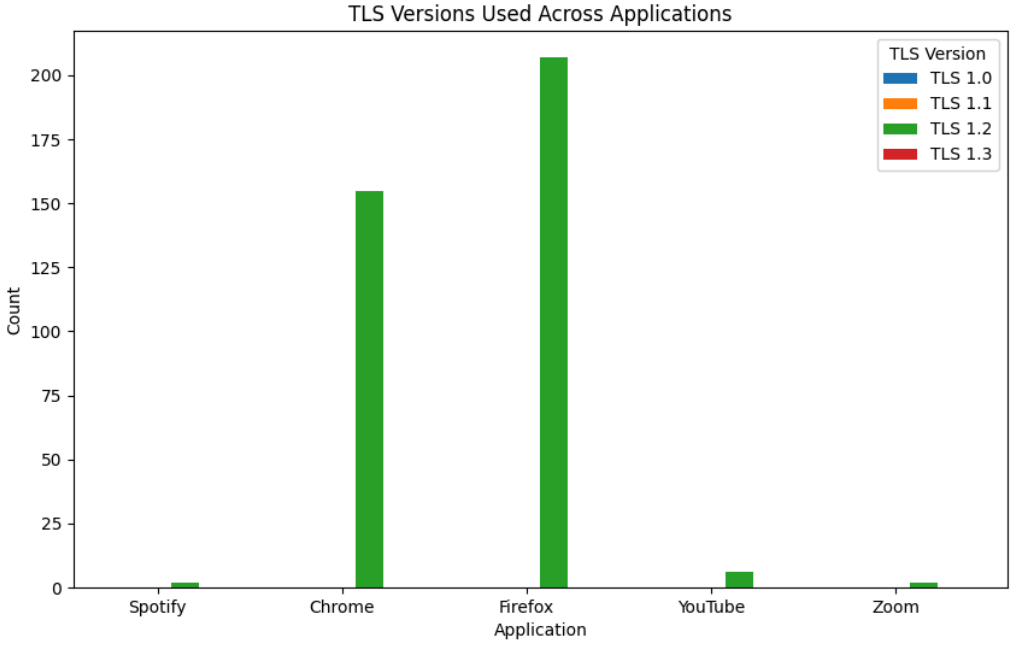
בהיבט של גודל החבילות ניתן לראות שיש הבדלים בין האפלקציות.
היוטיוב משתמש בגודל חבילה גדול יחסית. כל חבילה מכילה חלקי וידאו המועברים ללקוח בשידור חי. הגודל של החבילות תלוי ברזולוציה של הסרטון. ככל שהרזולוציה גבוהה כך גם גודל החבילה.
הסיבה היא שכדי לשמור על איכות הסרטון יש להעביר יותר נתונים.
ניתן לראות שגודל החבילות של הדפדפנים קטן משמעותית ביחס ליוטיוב. החבילות של הדפדפנים מכילות טקסט, תמונות, קבצי CSS ועוד. גודל החבילה כמובן תלוי בכמות האלמנטים שהדף משתמש ומשתנה בהתאם.
שתי האפליקציות הבאות (זום וספוטיפיי) משתמשות בחבילות קטנות אף יותר. הזום, בדומה ליוטיוב, מעביר חבילות בזמן אמת. חבילות אלו מורכבות בעיקר מוידאו ואודיו. גם כאן ככל שהאיכות גבוהה יותר החבילות יגדלו.
החבילה הקטנה ביותר היא של הספוטפיי. אפליקציה זו משתמשת בחבילות אודיו. חבילות אלו קטנות יותר מחבילות וידאו.

השוואת average inter-arrival time:



average inter-arrival time הוא זמן הממוצע בין הגעת חבילה אחת לאחרת.
ניתן לראות הבדל בין האפליקציות השונות. ביוטיוב, זמן ההגעה של החבילות תלוי באיכות הוידאו ובהזרמתו. ככל שהרזולוציה גבוהה יותר, למשל ב4K, החבילות יגיעו בתדירות גבוהה יותר כדי לשמור על איכות חלקה של הוידאו.ביוטיוב יש שימוש בBuffer ולכן זמן הממוצע בין החבילות הוא גבוה. באפלקציה זו הדפדפן מחלק את הסרטון למקטעים, שומר אותם בBuffer ומעביר אותם ללקוח. הזמן הממוצע ארוך כיוון שזמן ההגעה בין המקטעים גדול יחסית. בספוטיפיי, הזמן בין החבילות שונה כי מדובר בהזרמת אודיו. כאן נראה כי יש שימוש שונה בBuffer ולכן זמן הממוצע בין החבילות הוא קצת גבוה יותר. דרך הפעולה של הBuffer בספוטפיי הוא לשמור את כל השיר ולהעביר את אותו ללקוח. כך הממוצע אמור לרדת, אבל כנראה שבגלל שהספוטיפיי צורך פחות נתונים הוא מוריד מקטעים גדולים יותר בפחות בקשות ולכן הזמן הממוצע גדל.
בזום זמן ההגעה של החבילות תלוי באיכות השיחה, במיוחד כשמדובר בשיחות וידאו, כך שככל שהאיכות גבוהה יותר, החבילות יגיעו בתדירות גבוהה יותר לשמירה על שיחה רציפה ללא הפרעות.
בדפדפנים כמו כרום ופיירפוקס, הזמן בין החבילות משתנה לפי התוכן שבדף. הזמן הממוצע בין החבילות נמוך מכמה סיבות. הדפדפן צריך להוריד הרבה נתונים כמו טקסט, תמונות ופרסומות. כל נתון כזה מכיל חבילות והדפדפן מקבל ושולח אותן במהירות, דבר שמקטין את הממוצע. בנוסף, הדפדפנים משתמשים בחיבורים רבים כדי להוריד כמה קבצים בו זמנית. וככה הצפיפות בין החבילות גדלה. כלומר, סיבות אלו מקטינות את המרווח בין הגעת החבילות.

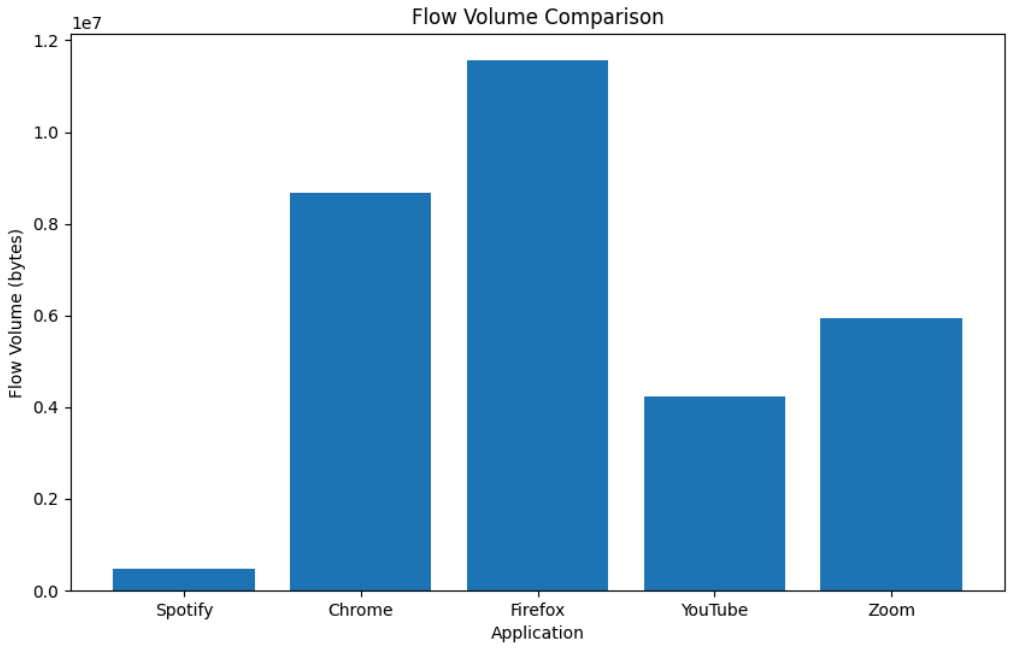
השוואת גרסאות TLS:



הגרף מציג את התפלגות גרסאות(TLS (1.0, 1.1, 1.2, 1.3 בקרב האפליקציות. מהנתונים עולה כי TLS 1.2 היא הגרסה הדומיננטית, שכן כמעט כל האפליקציות משתמשות בה, ובכמות גבוהה יחסית. ניתן לראות כי כרום ופיירפוקס הן הצרכניות העיקריות של TLS 1.2, כאשר כרום מציגה כ-150 חיבורים, ואילו Firefox מובילה עם יותר מ-200 חיבורים. לעומת זאת, ספוטיפיי, יוטיוב וזום כמעט ואינן משתמשות בפרוטוקול TLS, שכן היקף החיבורים שלהן קטן מאוד ביחס לשאר האפליקציות. מעניין לציין כי TLS 1.0, TLS 1.1, ו-TLS 1.3 אינן מופיעות כלל בגרף, מה שעשוי להעיד על כך שלא נמצאו חיבורים בגרסאות אלה בקבצי התעבורה שנבדקו, או שהכמות שלהן זניחה. באופן כללי, הנתונים אינם מפתיעים, שכן רוב האתרים והאפליקציות המודרניות כבר עברו ל-TLS 1.2 או TLS 1.3, בעוד ש-TLS 1.0 ו-1.1 יצאו משימוש באופן נרחב. עם זאת, היעדר מוחלט של TLS 1.3 הוא טעות בניתוח הנתונים, מכיוון שלאחר בדיקה מעמיקה לסיבה שבעקבותיה הקוד לא מצליח למצוא את הTLS 1.3 המשומש במספר מהאפליקציות. לדוגמה, בכרום הסיבה לכך ששני הערכים 0.0303 (TLS 1.2) ו- 0.0304 (TLS 1.3) עשויים להופיע יחד בתהליך handshake של TLS היא שבמהלך ההחלפה בין הלקוח לשרת, הלקוח שולח את רשימת הגרסאות שהוא תומך בהן. הלקוח יכול לכלול יותר מגרסה אחת ברשימה הזו, כמו TLS 1.2 ו-TLS 1.3, על מנת לוודא שהשרת יוכל לבחור את הגרסה המתאימה לו. השרת בודק את הרשימה הזו ומבצע חיבור בגרסה שהוא תומך בה, וזה יכול להוביל לכך שבפועל תמצא חבילות עם תמיכה

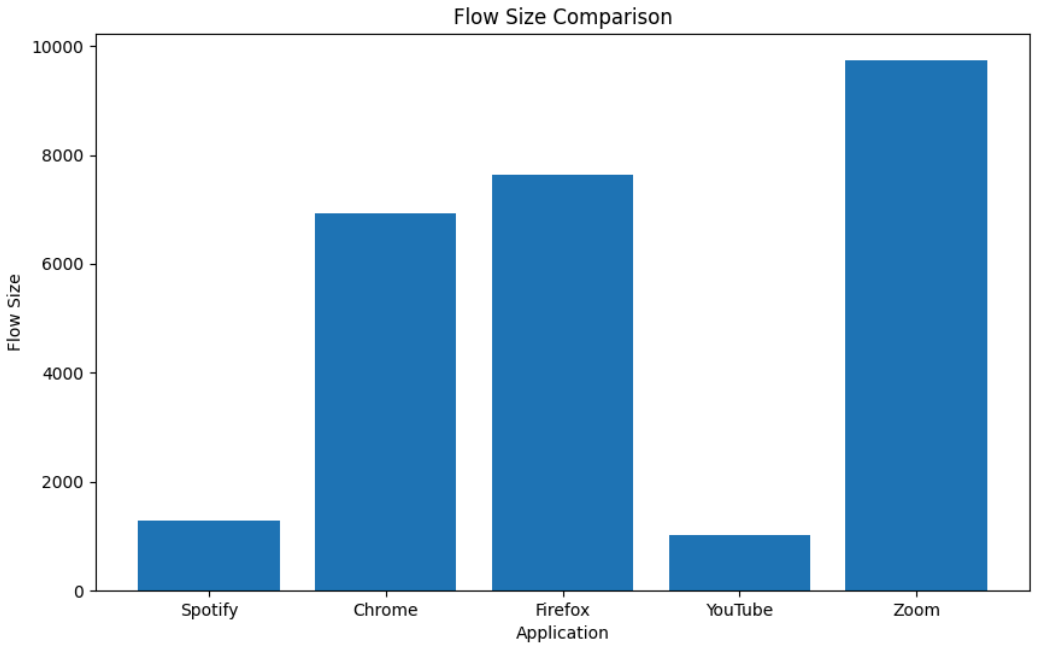
בגרסאות שונות (למשל, 1.2 ו-1.3) יחד באותו חיבור. כתוצאה מכך, הלקוח יכול להודיע על תמיכתו בגרסאות TLS 1.2 ו-TLS 1.3, מה שגורם לכך ששני הערכים, 0.0303 (TLS 1.2) ו-0.0304 (TLS 1.3), יופיעו יחד בתהליך החיבור ויפוענחו כגרסאות נתמכות, ובסופו של דבר הם עשויים להופיע כ-0.0303 (TLS 1.2) , אם השרת בוחר את הגרסה TLS 1.2 למרות שהתמיכה ב-1.3 TLS נמסרה גם כן.

<u>השוואה על פי flow volume:</u>



הגרף מציג השוואת נפחי תעבורה (Flow Volume) בין האפליקציות כאשר ציר ה-Y מייצג את כמות הנתונים (בבייטים). מהנתונים ניתן לראות כי פיירפוקס היא האפליקציה עם נפח התעבורה הגבוה ביותר, המגיע לכ-12 מיליון בתים, ואחריה כרום עם נפח של כ-9 מיליון בתים. זום ו-יוטיוב מציגות נפח תעבורה נמוך יותר, כאשר ל-זום יש כ-6 מיליון בתים ול-יוטיוב כ-4 מיליון בתים. ספוטיפי מציגה את נפח התעבורה הנמוך ביותר מבין האפליקציות, בהפרש משמעותי מהאחרות. הנתונים עשויים להעיד על דפוסי השימוש השונים של כל אפליקציה – פיירפוקס ו-כרום, כדפדפנים, צורכים נפח גדול של נתונים, ככל הנראה עקב טעינת דפי אינטרנט ותעבורה מוצפנת. זום, למרות היותו יישום תקשורת וידאו, מציג נפח נמוך יחסית, מה שעשוי להעיד על דגימה מסוימת של שימוש או דחיסת נתונים יעילה. יוטיוב, למרות היותו יישום הזרמת וידאו, מציג תעבורה נמוכה יחסית, ייתכן בשל אופטימיזציה של רוחב פס או ניתוח של קטעים קצרים.

השוואה על פי flow size:

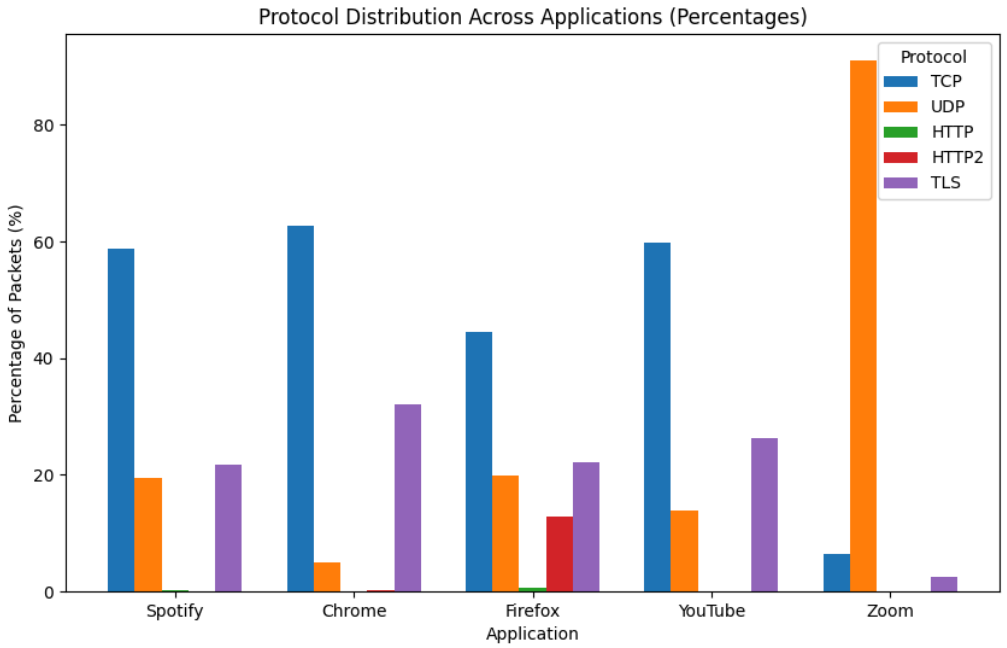


הגרף מציג השוואת גודל הזרימה בין האפליקציות ספוטיפיי, כרום, פיירפוקס, יוטיוב וזום, כאשר ציר ה-Y מייצג את גודל הזרימה. ניתן לראות כי Zoom מציגה את גודל הזרימה הגבוה ביותר, המגיע לכ-10,000 יחידות, מה שמעיד על כך שחיבורי הרשת של Zoom כוללים תעבורה רבה בכל זרימה בודדת. פיירפוקס וכרום מציגות ערכים גבוהים גם כן, עם גודל זרימה של כ-7,500 ו-7,000 בהתאמה, מה שיכול להעיד על כך שדפדפנים אלו מבצעים העברות נתונים רציפות ומשמעותיות במסגרת חיבורי רשת בודדים.

מנגד, יוטיוב וספוטיפיי מציגות גודל זרימה נמוך משמעותית, עם ערכים שנעים סביב 1,000-1,500. המשמעות היא שהזרימות שלהן קצרות יותר או שהנתונים מחולקים ליחידות קטנות יותר לאורך זמן. עובדה זו עשויה להיגרם ממנגנוני הזרמת נתונים אופטימליים, אשר מחלקים את המידע לחבילות קטנות יותר כדי לשפר את חוויית המשתמש ולצמצם שיהוי .

באופן כללי, הנתונים מעידים כי יישומים שונים מתאפיינים בפרופילי זרימה מגוונים: זום, כרום ופיירפוקס נוטים להשתמש בזרימות ארוכות וכבדות יותר, בעוד יוטיוב וספוטיפיי משתמשים בזרימות קטנות ומבוקרות יותר, ככל הנראה כדי להתאים לאופי השימוש שלהם.

<u>השוואה על פי התפלגות פרוטוקולים (לפי אחוזים):</u>



הגרף מציג את התפלגות הפרוטוקולים באחוזים על פני האפליקציות, ניתן לראות ש-TCP הוא הפרוטוקול הנפוץ ביותר ברוב האפליקציות, פרט לזום, שבו UDP שולט באופן מוחלט עם אחוזים גבוהים במיוחד. TLS נפוץ גם כן, בעיקר בדפדפנים כמו Chrome וFirefox, אך נוכח גם ספוטיפיי ויוטיוב (שאותם פתחנו דרך הדפדפן פיירפוקס). הפרוטוקול HTTP2 מופיע בעיקר בפיירפוקס, אך בשאר האפליקציות כמעט שאינו קיים. HTTP כמעט שאינו נראה על הגרף, מה שמעיד על השימוש הנמוך בו ביחס לשאר הפרוטוקולים. ההתפלגות מציגה הבדלים מעניינים בין האפליקציות, כאשר אפליקציות מדיה כמו זום מסתמכות על UDP, בעוד שדפדפנים נשענים במידה רבה על TCP ו-TLS.

הספריות בהן השתמשנו:

```
import os
import pyshark
import matplotlib.pyplot as plt
import numpy as np
from collections import defaultdict, Counter
```

חלק 4

(בסוף חלק 4 מצורפות הספריות בהן השתמשנו לחלק זה)
בחלק זה התבקשנו להציג שני תרחישים ובהם תוקף מנסה לזהות את האפליקציות שהמשתמש מפעיל. בשני התרחישים התוקף יודע נתונים שונים תרחיש ראשון התוקף יודע גודל של כל חבילה, חותמת זמן וFlowID בעוד שבתרחיש השני יודע התוקף רק את גודל החבילה וחותמת הזמן שלה.

תרחיש 1: כשהתוקף יודע את גודל החבילה, הזמן המוערך (timestamp), ואת הFlowID הסבר למה התוקף יכול/לא יכול לזהות את האתר:
בתרחיש זה, התוקף יודע את כל הפרטים המהותיים על כל חבילה: גודל, זמן מוערך וFlowID. בעזרת המידע הזה, התוקף יכול לעקוב אחרי כל זרם תעבורה ולבצע התאמות לזיהוי של האפליקציות או האתרים שבהם השתמש המשתמש.

- **זיהוי אפליקציות:** המידע על גודל החבילה והזמן המוערך יכולים להיות מספיקים כדי לזהות את האפליקציות בהן המשתמש עושה שימוש, אפילו אם התעבורה מוצפנת. לדוגמה, אפליקציות כמו YouTube, Zoom או Spotify יש להן חתימות תעבורה ייחודיות שיכולות להיות מזוהות לפי גודל החבילות ופעילותן בזמן.
- **הזרמים של האפליקציות:** המידע על הFlowID מאפשר לתוקף לעקוב אחרי זרם התעבורה באופן יותר מדויק. התוקף יוכל גם להשוות בין החבילות ולחזות את היישום בהתאם לדפוסים הקיימים, גם אם התעבורה מוצפנת.

מניעה של התקפה כזו:

- **שימוש בהסוואת תעבורה (Traffic Obfuscation):** ניתן להטמיע מנגנוני הסוואה שמערבבים את סדר החבילות או מייצרים חבילות בגודל אחיד כדי להקשות על זיהוי דפוסים.
- **הצפנת התעבורה בשכבות גבוהות יותר (End-to-End Encryption):** השימוש בהצפנה שמבוססת על פרוטוקולים חזקים (כמו TLS 1.3) יכול להקשות על זיהוי האפליקציות, כיוון שהתוקף לא יוכל לזהות את התוכן של התעבורה.
- **שימוש ב-VPN:** שימוש ב-VPN יכול להסתיר את כתובת הIP של המשתמש ולהקשות על המעקב אחרי זרם התעבורה, מה שיכול למנוע זיהוי מדויק של האפליקציות.

תרחיש 2: כשהתוקף יודע רק את גודל החבילה וזמן המוערך (timestamp)

הסבר למה התוקף יכול/לא יכול לזהות את האתר:

בתרחיש זה, התוקף יודע רק את גודל החבילה וזמן המוערך של כל חבילה, ללא מידע על הFlowID של הזרם. למרות שהתוקף יכול עדיין להשתמש במידע הזה לזיהוי הכללי של התעבורה, הוא יפגוש במגבלות רבות:

- **זיהוי אפליקציות:** למרות שהתקפות כמו אלו יכולות לזהות את דפוסי התעבורה הכלליים של אפליקציות מסוימות, המידע הנחוץ לעיתים לזהות אפליקציות באופן מדויק הוא מוגבל יותר. התוקף לא יודע את הזרם המדויק, וזה מקשה על זיהוי האפליקציה.
- **הבדלים בין אפליקציות:** אפליקציות שונות יוצרות תעבורה ברמות שונות של גודל חבילות ועיכוב בין החבילות, אך בגלל חוסר מידע על הFlowID, קשה יותר לתוקף להפריד ביניהם בדיוק.

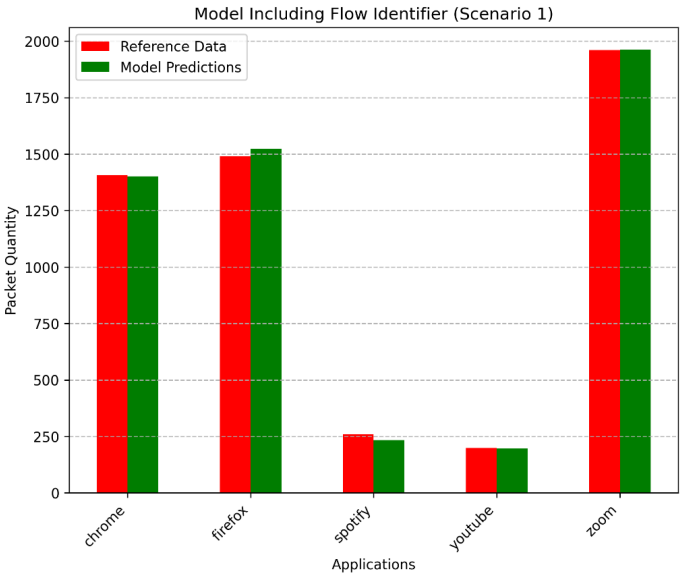
מניעה של התקפה כזו:

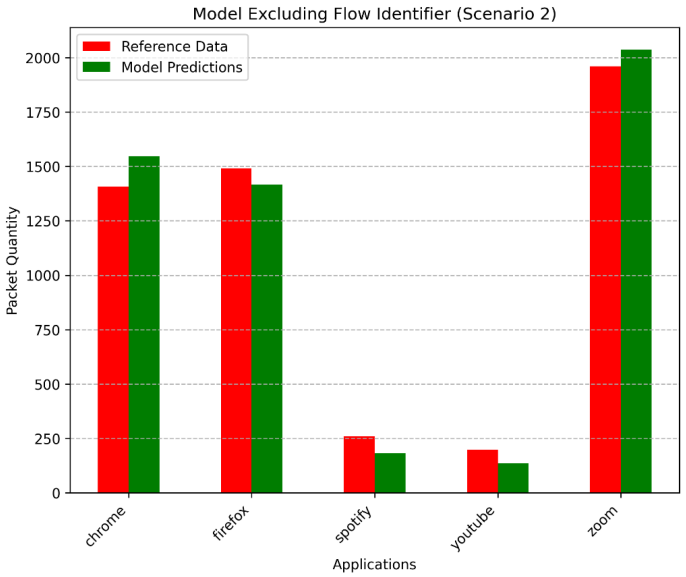
- **הצפנה של כל המידע:** הצפנה טובה, כמו TLS 1.3, יכולה להסתיר לא רק את התוכן של החבילות אלא גם את דפוסי התעבורה. התוקף יתקשה מאוד לזהות אפליקציות רק על בסיס גודל החבילות וזמן הגעתן.
- **שימוש באסטרטגיות כמו padding:** הוספת פדינג (ריפוד) לחבילות או קביעת גודל קבוע לחבילות עשוי להקשות על זיהוי דפוסים בעזרת גודל החבילות בלבד.
- **שימוש בפרוטוקולים המתמודדים עם דליפות מידע:** לדוגמה, פרוטוקולים שמפיצים את התעבורה או שמבצעים ערבוב בין החבילות, כך שהתוקף לא יכול להבחין בקלות בין תעבורה של אפליקציות שונות.

לסיכום, בתרחיש הראשון התוקף יכול לזהות את האתר או האפליקציה בקלות יחסית, בעוד שבתרחיש השני, המידע המצוי בידו מוגבל יותר, מה שמקשה על זיהוי מדויק של האפליקציה או האתר, אך לא מונע לגמרי את האפשרות לזהות דפוסים כלליים.

בנוסף כתבנו קוד המנתח את הנתונים ומציג גרף, הגרף מציג השוואה בין שני התרחישים של למידת מכונה שנועדו לחזות את מספר חבילות המידע שמיוצרות על ידי אפליקציות שונות. המודל הראשון משתמש במידע שיש בתרחיש מספר 1 והמודל השני משתמש במידע שיש בתרחיש מספר 2. "Reference Data" מייצגת את המספר האמיתי של חבילות המידע שנרשמו, ו"Model Predictions" מייצגת את התחזית של המודל. השוואה בין התוצאות מאפשרת לראות שהמודל שמשתמש בFlowID מדויק יותר, מה שמצביע על חשיבות המידע הזה לזיהוי אפליקציות.

כדי להגיע לתוצאות אלו, השתמשנו באלגוריתם Random Forest, שהוא אלגוריתם למידה מונחית ומיועד למשימות סיווג. לפני אימון המודל, ביצענו סוג של ניתוח נתונים על נתוני הרשת. כלומר, חילצנו את התכונות הרלוונטיות מכל חבילת מידע: גודל החבילה, זמן בין חבילות, פרוטוקול התקשורת, ובמודל הראשון גם את הFlowID. תכונות אלו, לאחר עיבוד מתאים, שימשו כקלט לאימון המודל.





הניתוח של הגרפים חושף מספר תובנות משמעותיות. ראשית, ניתן לראות בבירור כי התרחיש הראשון, שבו נעשה שימוש ב-FlowID, מספק דיוק חיזוי גבוה יותר של 95.39%. זהו שיפור משמעותי לעומת התרחיש השני, בו המודל פשוט יותר ומניב דיוק של 63.71%. נתון זה מצביע על כך ש-FlowID, כמרכיב מרכזי במודל, משפר בצורה דרמטית את יכולת החיזוי של המודל ומעלה את ביצועיו. זה מדגיש את היתרון של עבודה עם נתונים מפורטים יותר, כמו FlowID, שמסייעים לדייק יותר בניתוח התנועה של חבילות הנתונים ברשת.

שנית, הגרפים מראים הבדל ניכר ביכולת החיזוי בין היישומים השונים. דוגמה בולטת לכך היא היישום Zoom, שבו המודל הראשון מצליח לנבא את כמות החבילות בצורה מדויקת מאוד. נתון זה יכול להעיד על כך שZoom מציג מאפיינים ייחודיים בתעבורת הרשת שלו, שמאפשרים למודל לזהות ולהתאים בצורה טובה יותר את החבילות והזרמים הקשורים אליו. ייתכן כי מאפיינים אלו קשורים לאופי השיחות הווידאו, למידת הצפיפות של התעבורה, או לאסטרטגיות ניהול החיבורים שZoom משתמש בהן.

לסיכום, ניתן ללמוד מהגרפים של שני המודלים את הדגש ביתרון של המודל הראשון, המבוסס על FlowID. הדבר מצביע על החשיבות הרבה של הזרם המלא והפרטים המובילים לדיוק חיזוי גבוה יותר, במיוחד כאשר מדובר ביישומים עם מאפיינים ייחודיים כמו Zoom.

```
import pyshark
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
```

בדיקת בונוס:

במהלך הבדיקה שמענו שירים בספוטיפי באמצעות כרום ושלחנו 2 מיילים בזמן הזה, נבהיר שגם ספוטיפיי וגם Gmail משתמשים ב-TLSv1.3 בין היתר.

<u>ניתוח הבדיקה:</u>

- ניתן לראות כבר כי בעקבות שליחת המיילים(בפקטות Client Hello) נגרם איבוד פקטות של ספוטיפיי מה שגרם לשליחה חוזרת של ack (TCP Dup ACK), לכן ניתן להסיק כי ספוטיפיי משתמש בשיטת Chunking של שליחת פקטות, הוא שולח כמות מידע בגודל מסוים מחולקת לפקטות קטנות יותר ומשתמש ב-ack.
- בנוסף, ניתן לראות כי נגרם עיכוב בין הפקטות של Gmail המסמלות את שליחת האימייל.
- ניתן להסיק כי דברים אלו נגרמו מכיוון שגם Gmail וגם ספוטיפיי משתמשים בפרוטוקול TCP על אותו חיבור רשת מה שיכל לגרום לכמה סוגי "התנגשויות":
  1. <u>עומס רשת(Network Congestion):</u> יכול להיות ששליחת האימייל באותו הרגע גרמה לעלייה מעבר לרוחב הפס, מה שגרם לעיכוב חבילות ואולי להגעת timeout מצד ספוטיפיי.
  2. <u>TCP Queueing:</u> עוד אפשרות שיכלה לגרום לעיכוב חבילות היא packet scheduling, שנכנס לפעולה כאשר מגיעות מספר פקטות ממקורות שונים המשתמשים באותו פרוטוקול אינטרנט, מה שקורה זה שפקטות מסוימות יעובדו לפני אחרות למרות הזמן שהן התקבלו בו(כמובן תלוי במספר דברים כגון אלגוריתם המיון, מערכת ההפעלה וכו').
  3. <u>Router Buffer:</u> סיבה דומה לקודמת שגם משפיעה על זמן הגעת החבילות אך קשורה ל-router שלנו היא ה-Buffering של החבילות שמבצע גם הוא מיון, שקשור למקור החבילות ולהעדפה מסיבות כגון גודל וסוג, פחות סביר שזו הסיבה שגרמה לעיכוב כיוון שלאימייל יש העדפה נמוכה ב-Buffering.

1.255890857	10.0.2.15	142.250.75.106	TLSv1.3	2334	Client Hello
1.363370530	10.0.2.15	142.250.75.170	TLSv1.3	2304	Client Hello
5.055879386	10.0.2.15	35.186.224.24	TCP	56	[TCP Dup ACK 2#1] 44380 → 443 [ACK] Seq=1 Ack=1 Win=65535 Len=
5.060739034	10.0.2.15	35.186.224.24	TCP	56	[TCP Dup ACK 1#1] 60654 → 443 [ACK] Seq=1 Ack=1 Win=61320 Len=
5.060923235	35.186.224.24	10.0.2.15	TCP	62	[TCP Dup ACK 4#1] [TCP ACKed unseen segment] 443 → 44380 [ACK]
5.061232770	35.186.224.24	10.0.2.15	TCP	62	[TCP Dup ACK 3#1] [TCP ACKed unseen segment] 443 → 60654 [ACK]
8.715922119	10.0.2.15	142.250.75.132	TLSv1.3	1812	Client Hello

ניתן לראות בצילום כי לאחר שליחת האימייל התקבלו פקטות שנשלחו מצדנו עבור Spotify, שני פקטות המודיעות על ack חוזר, ולאחר מכן קבלת הack על מנת להמשיך את ה-sequence.

ניתן לראות בנוסף את העיכוב שנוצר בין הפקטות של Gmail ממש בין החבילה האחרונה שמתקבלת לפני שליחת הack והחבילה שנשלחת ישר אחרי(הבדל של 17 שניות).

*בקובץ pcapng. המסונן(רק חבילות שקשורות ל-ספוטיפיי או Gmail) מופיע כי החבילות של Gmail משתמשות ב-TLSv1 אבל נציין כי זוהי שגיאה ויזואלית וכי בהקלטה בזמן אמת הפקטות שהתקבלו השתמשו אכן ב-TLSv1.3.

Linkedin accounts:
https://www.linkedin.com/in/ori-hamou-a63826354/
https://www.linkedin.com/in/yaniv-greenberg-82330533a/
https://www.linkedin.com/in/eyal-sheffer-7337b1354/
https://www.linkedin.com/in/yuvaliloo-%D7%A4%D7%9C%D7%92-7a92b8354/