פרויקט גמר- רשתות תקשורת

עליזה לזר-336392899
ספיר בקשי אטיאס- 212125660

חלק 1:

1.

גורמים להעברת קבצים איטית:

- בקרת גודש ב TCP- אם הרשת חווה עומס TCP ,מפחית את קצב השידור כדי למנוע איבוד נתונים.
- בקרת זרימה ב TCP-אם חלון הקבלה של המקבל קטן מדי, זה מאט את קצב השידור של השולח.
- לטנציה גבוהה –עיכובים בהגעת אישורי קבלה (ACK) מאריכים את זמן ההעברה.
- איבוד מנות ושידורים חוזרים –ככל שאחוז איבוד המנות גבוה יותר, יש יותר שידורים חוזרים, מה שמאט את ההעברה.
- מגבלות רוחב פס –הרשת עשויה שלא לתמוך בקצב השידור הרצוי.
- גודל Maximum Segment Size ו – Maximum Transmission Unit -אם חבילות מפוצלות ,(fragmentation) זמן השידור מתארך.

שלבי פתרון תקלה:

1. בדיקת תעבורת TCP עם – Wireshark יש לחפש שידורים חוזרים, חלונות קטנים או גודש.
2. בדיקת גודל חלון ה – TCP-אם קטן מדי, ניתן להגדילו.
3. בדיקת רוחב הפס הזמין –ניתן להשתמש ב iperf-כדי למדוד תפוקה בפועל.
4. כוונון הגדרות – TCP יש להתאים גדלי באפרים ולהפעיל הרחבת חלון TCP (Window Scaling).
5. שימוש בהעברות מקבילות או – UDP אם אפשר, להשתמש ב MPTCP-(נלמד בשיעור 9) או UDP כדי לעקוף את המגבלות של.TCP

.2

בקרת זרימה ב TCP:

בקרת הזרימה של TCP מבטיחה שהשולח לא יציף את המקבל בנתונים שהוא אינו יכול לעבד. היא מבוססת על חלון הקבלה של.TCP

כאשר לשולח יש כוח עיבוד גבוה יותר מהמקבל :

- השולח יכול לשלוח נתונים במהירות גבוהה, אך אם גודל הבאפר של המקבל קטן, זה יגביל את קצב ההעברה.
- גודל חלון ה TCP-קובע כמה נתונים ניתן לשלוח לפני שמתקבל אישור.
- אם החלון קטן מדי, זה גורם לבזבוז של רוחב פס.

השפעה על הביצועים:

- השולח נאלץ להמתין עד שהמקבל יעבד את הנתונים.
- אם הבאפר של המקבל מלא ,הוא שולח אישור חלון בגודל 0 ,מה שגורם לעצירת ההעברה.
- פתרון אפשרי :הגדלת הבאפרים או הפעלת window scalingלשיפור היעילות.

.3

תפקיד הניתוב :

כאשר קיימים מספר נתיבים אפשריים, פרוטוקולי ניתוב כגון OSPFאו RIP קובעים את הנתיב האופטימלי בהתבסס על :

- מספר הקפיצות.(RIP)
- עלות הנתיב.(OSPF)
- לטנציה ועומס ברשת (ניתוב דינמי).

גורמים המשפיעים על הביצועים:

1. לטנציה –נתיב קצר יותר מפחית את זמן העברת הנתונים.
2. עומס ברשת –אם נתיב מסוים עמוס, הוא עלול לגרום לאיבוד מנות.
3. אמינות –חלק מהנתיבים עשויים להיות לא יציבים, מה שגורם לשינויים תכופים.
4. רוחב פס –נתיבים מסוימים מספקים קיבולת גבוהה יותר.
5. איזון עומסים –חלק מהנתבים משתמשים ב ECMP (Equal-Cost Multi-Path)-לפיזור העומס.

פתרונות לשיפור ביצועים:

- שימוש בניתוב דינמי (OSPF) כדי להתאים את הנתיב למצבי הרשת.
- ניטור איבוד מנות וג'יטר .(Jitter)
- תעדוף תנועה קריטית בעזרת QoS (Quality of Service

.4

ב MPTCP (Multipath TCP)-והוא משפר את הביצועים בדרך הבאה:

- <u>שימוש במספר נתיבים בו זמנית</u> –בניגוד ל TCP-רגיל שמשתמש רק בנתיב אחד , MPTCPיכול לפצל את התעבורה על פני כמה רשתות.
- <u>הגדלת רוחב הפס-</u> שילוב של מספר חיבורים מאפשר תפוקה גבוהה יותר.
- <u>הפחתת גודש-</u> אם נתיב אחד עמוס, התעבורה תעבור לנתיב אחר.
- <u>שיפור האמינות-</u> אם חיבור אחד כושל MPTCP ,ממשיך להשתמש בשאר הנתיבים ללא ניתוק.

שימושים נפוצים:

- ברשתות סלולריות (4G + Wi-Fi)לחיבור יציב.
- העברות קבצים מהירות יותר בענן באמצעות .Multipath Transfers

.5

סיבות אפשריות לאיבוד מנות:

<u>שכבת הרשת:(Network Layer - IP)</u>

- עומס ברשת –אם יש יותר מדי תעבורה, חלק מהמנות נמחקות.
- לולאות ניתוב – (Routing Loops) טבלאות ניתוב לא נכונות עלולות לגרום למנות להיתקע בלולאה.
- MTU לא מתאים –אם ה MTU-גדול מדי ויש פירוק מנות ,(fragmentation) חלקן עשויות ללכת לאיבוד.

<u>שכבת התעבורה:(Transport Layer - TCP/UDP)</u>

- שידורים חוזרים מרובים ב – TCP-אם ACKs הולכים לאיבוד TCP ,ישלח מחדש נתונים.
- UDP לא אמין –UDP אינו מבצע שידורים חוזרים, כך שאיבוד מנות משפיע ישירות.
- חומת אש או – NAT מדיניות שגויה עלולה להפיל מנות.

<u>דרכי פתרון:</u>

1. שימוש ב ping-ו traceroute- כדי לבדוק היכן מתרחשים האיבודים.
2. ניתוח התעבורה ב Wireshark- כדי לזהות שידורים חוזרים.
3. בדיקת עומס על הנתבים- אם הנתב עמוס, הוא עלול להפיל מנות.
4. התאמת הגדרות MTU כדי למנוע פירוק מנות.
5. הפעלת QoS (Quality of Service) לתעדוף חיבורים חשובים.

חלק 2:

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

1. מהי התרומה המרכזית של המאמר?

המאמר מציג שיטה חדשה לזיהוי מערכת ההפעלה, הדפדפן והאפליקציה של משתמשים דרך ניתוח **תעבורת HTTPS מוצפנת** .למרות ש HTTPS-אמורה להגן על פרטיות המשתמשים, החוקרים הראו שאפשר עדיין לחשוף מידע חשוב על המכשיר והשימוש בו בעזרת ניתוח דפוסים חיצוניים.

- **זיהוי מתעבורה מוצפנת:**
  השיטה מאפשרת לזהות מידע על מערכת ההפעלה והאפליקציות של המשתמשים בלי לפתוח את ההצפנה, אלא דרך ניתוח **זמני ההגעה של חבילות** ,**הגודל שלהן** ,ומאפיינים מסוימים בפרוטוקול **SSL**שלא מוצפנים.
- **מאגר נתונים גדול ומקיף:**
  כדי לבדוק את השיטה, החוקרים יצרו מאגר של **יותר מ-20,000 דוגמאות** ,שכללו:
  - **מערכות הפעלה** כמו Windows, Ubuntu ו.OSX-
  - **דפדפנים** כמו Chrome, Firefox ו.Safari-
  - **אפליקציות** כמו YouTube, Facebook ו.Twitter-
- **מאפיינים חדשים לניתוח התעבורה:**
  מעבר למאפיינים הסטנדרטיים, המאמר מציע תכונות חדשות:
  מאפייני :SSL למשל, ניתוח מספר הרחבות SSL ואורך מזהי הסשן.
  התנהגות "מתפרצת" של הדפדפנים :(Bursty Behavior) דפוסי שימוש כמו באפליקציות YouTube, שבהן יש קטעי שקט בתעבורה ואז פרצי מידע גדולים.
- **תוצאות הניסוי:**
  השיטה הראתה **דיוק מרשים של 96.06%** בזיהוי המערכת, הדפדפן והאפליקציה. לשם השוואה, שימוש במאפיינים בסיסיים בלבד הניב דיוק של **93.51%**, מה שמדגיש עד כמה המאפיינים החדשים משפרים את התוצאה.
- **השלכות על אבטחת מידע:**
  המאמר מדגים שגם אם משתמשים בחיבורים מוצפנים, **לא תמיד הפרטיות נשמרת** .תוקפים יכולים להשתמש במידע הזה כדי לעקוב אחרי משתמשים או לבצע מתקפות ממוקדות.

2. אילו תכונות תעבורה המאמר משתמש בהן, ואילו מהן חדשות?

המאמר משתמש בשתי קבוצות עיקריות של תכונות לניתוח תעבורת:HTTPS

<u>מאפיינים בסיסיים (base features)</u>

- מספר חבילות יוצאות (Forward packets)
- סה"כ בתים יוצאים (Forward total Bytes)
- הפרש זמן הגעה מינימלי בין חבילות יוצאות (Min forward inter arrival time difference)
- הפרש זמן הגעה מקסימלי בין חבילות יוצאות (Max forward inter arrival time difference)
- הפרש זמן הגעה ממוצע בין חבילות יוצאות (Mean forward inter arrival time difference)
- סטיית תקן של הפרש זמן הגעה בין חבילות יוצאות (STD forward inter arrival time difference)
- מינימום חבילות יוצאות (Min forward packets)
- סטיית תקן של מספר חבילות יוצאות (STD forward packets)

- מספר חבילות נכנסות (Backward packets)
- סה"כ בתים נכנסים (Backward total Bytes)
- הפרש זמן הגעה מינימלי בין חבילות נכנסות (Min backward inter arrival time difference)
- הפרש זמן הגעה מקסימלי בין חבילות נכנסות (Max backward inter arrival time difference)
- הפרש זמן הגעה ממוצע בין חבילות נכנסות (Mean backward inter arrival time difference)
- סטיית תקן של הפרש זמן הגעה בין חבילות נכנסות (STD backward inter arrival time difference)
- מינימום חבילות נכנסות (Min backward packets)
- סטיית תקן של מספר חבילות נכנסות (STD backward packets)
- ערך TTL ממוצע עבור חבילות יוצאות (Mean forward TTL value)
- גודל מינימלי של חבילה יוצאת (Minimum forward packet)
- גודל מינימלי של חבילה נכנסת (Minimum backward packet)
- גודל מקסימלי של חבילה יוצאת (Maximum forward packet)
- גודל מקסימלי של חבילה נכנסת (Maximum backward packet)
- סה"כ חבילות (Total packets)
- גודל מינימלי של חבילה (Minimum packet size)
- גודל מקסימלי של חבילה (Maximum packet size)
- גודל חבילה ממוצע (Mean packet size)
- שונות בגודל החבילות (Packet size variance)

<u>מאפיינים חדשים (new features)</u>

- גודל חלון התחלתי של TCP (TCP initial window size)
- גורם שינוי חלון TCP (TCP window scaling factor)
- מספר שיטות דחיסת SSL (# SSL compression methods)
- מספר הרחבות SSL (# SSL extension count)
- מספר שיטות הצפנה של SSL (# SSL cipher methods)
- אורך מזהה הפעלת SSL (SSL session ID len)
- תפוקה מקסימלית של פסגות יוצאות (Forward peak MAX throughput)
- תפוקה ממוצעת של פסגות נכנסות (Mean throughput of backward peaks)
- תפוקה מקסימלית של פסגות נכנסות (Max throughput of backward peaks)
- תפוקה מינימלית של פסגות נכנסות (Backward min peak throughput)
- סטיית תקן של תפוקה מרבית של פסגות נכנסות (Backward STD peak throughput)
- מספר פרצי מידע יוצאים (Forward number of bursts)
- מספר פרצי מידע נכנסים (Backward number of bursts)
- תפוקה מינימלית של פסגות יוצאות (Forward min peak throughput)
- תפוקה ממוצעת של פסגות יוצאות (Mean throughput of forward peaks)
- סטיית תקן של תפוקה מרבית של פסגות יוצאות (Forward STD peak throughput)
- הפרש זמן הגעה מינימלי בין פסגות נכנסות (Mean backward peak inter arrival time diff)
- הפרש זמן הגעה מינימלי בין פסגות יוצאות (Minimum backward peak inter arrival time diff)
- הפרש זמן הגעה מקסימלי בין פסגות יוצאות (Maximum backward peak inter arrival time diff)
- סטיית תקן של הפרש זמן הגעה בין פסגות נכנסות (STD backward peak inter arrival time diff)
- הפרש זמן הגעה מינימלי בין פסגות יוצאות (Minimum forward peak inter arrival time diff)
- הפרש זמן הגעה מקסימלי בין פסגות יוצאות (Maximum forward peak inter arrival time diff)
- סטיית תקן של הפרש זמן הגעה בין פסגות יוצאות (STD forward peak inter arrival time diff)

- מספר חבילות שמורות (Keep alive packets)
- גודל חלון המקטעים של (TCP Maximum Segment Size) TCP
- גרסת SSL עבור חבילות יוצאות (Forward SSL Version)

3. מהם התוצאות העיקריות ומהן התובנות מהן?

תוצאות עיקריות:

- השיטה שפיתחו החוקרים הצליחה לזהות את מערכת ההפעלה, הדפדפן והאפליקציה בדיוק של **96.06%**כאשר השתמשו בשילוב של התכונות הבסיסיות והתכונות החדשות.
- כאשר נעשה שימוש רק בתכונות הבסיסיות, הדיוק היה .**93.51%**הוספת התכונות החדשות הביאה לשיפור של כ-3% בדיוק.
- זיהוי מערכת ההפעלה היה **מושלם כמעט לחלוטין** ,בעוד שזיהוי הדפדפנים והאפליקציות היה **קרוב לשלמות** ,עם טעויות בעיקר כאשר התוויות סומנו כ-"לא ידוע."

תובנות מרכזיות:

1. ניתן להסיק שתעבורת ,HTTPS אף שהיא מוצפנת, אינה מספקת הגנה מוחלטת מפני ניתוח תעבורת רשת.
2. היכולת לזהות את סוג המכשיר, הדפדפן והאפליקציה יכולה לשמש תוקפים בבניית אסטרטגיות התקפה ממוקדות, ואף לאיסוף מידע סטטיסטי לצרכי פרסום ושיווק.
3. התכונות החדשות שמבוססות על התנהגות תעבורתית של דפדפנים (burst behavior) מהוות פריצת דרך בתחום ניתוח התעבורה המוצפנת.

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

1. מהי התרומה המרכזית של המאמר?

המאמר מציע פתרון חדשני לסיווג מוקדם של תעבורת אינטרנט מוצפנת, במיוחד תחת TLS 1.3 ו-Encrypted ClientHello (ECH), שמסתיר מידע קריטי ומקשה על זיהוי תעבורה בשיטות מסורתיות. התרומה המרכזית היא פיתוח אלגוריתם hRFTC, שמשלב מטא-נתונים של TLS (שאינם מוצפנים) עם מאפיינים סטטיסטיים של זרימה, מה שמאפשר סיווג מדויק ומהיר יותר בהשוואה לשיטות קיימות.

המאמר מציג גם מספר חידושים נוספים. ראשית, הוא מראה כי שיטות סיווג מבוססות TLS בלבד מאבדות מיעילותן תחת ECH, כשהדיוק שלהן יורד ל 38.4%, בעוד ש hRFTC מצליח להגיע ל 94.6%. בנוסף, האלגוריתם תומך גם בתעבורה שאינה מבוססת TCP, כמו QUIC (HTTP/3), מה שהופך אותו לגמיש יותר לשימושים עתידיים. החוקרים גם אספו מסד נתונים נרחב, הכולל מעל 600,000 זרימות TLS מ 19 סוגי תעבורה שונים מאזורים שונים בעולם, והפכו אותו למשאב פתוח למחקר. כמו כן, המאמר מראה כי hRFTC יעיל יותר מאלגוריתמים קודמים, במיוחד תחת הצפנת ECH, אך מציין כי הביצועים שלו משתנים בהתאם למיקום גיאוגרפי, ולכן נדרש אימון מחדש בכל אזור כדי לשמור על דיוק מרבי.

2. אילו תכונות תעבורה המאמר משתמש בהן, ואילו מהן חדשות?

המאמר משתמש במאפיינים שמחולקים לשלוש קבוצות עיקריות:

<u>מאפייני TLS :</u> מבוססים על TLS Handshake. חלק מהשדות עדיין לא מוצפנים גם תחת ECH ויכולים לשמש לסיווג.

- רשימת שיטות ההצפנה הנתמכות בין הלקוח לשרת (Cipher Suites)
- המבנה של ההרחבות בTLS (TLS Extensions)
- קבוצת שיתוף מפתחות ומפתחות משותפים מראש (Key Share Group ו-Pre-Shared Keys)
- אורך הודעות (ClientHello/ServerHello Length)
- גרסת TLS בשימוש (TLS Version)
- אורך וסדר ההרחבות ב (TLS (Extension Length & Ordering-
- מספר האישורים בשרשרת (TLS (Certificate Chain Length
- מידע על גודל ההודעות והמבנה שלהן ב (TLS (TLS Record Layer Fields-
- הרחבות אקראיות שנועדו להקשות על זיהוי (GREASE Extensions)
- חישוב רמת הרנדומליות של הרחבות TLS כדי להבדיל בין שירותים שונים (Entropy of TLS Extensions)
- נוכחות של הרחבות מסוימות שיכולות לרמוז אם השירות משתמש ב HTTP/2-או QUIC (Presence of Specific Extensions).

<u>מאפיינים סטטיסטיים של זרימה:</u> מבוססים על ההתנהגות הכללית של הזרימה.

- חישובים של גדלי חבילות – ממוצע, חציון, סטיית תקן ועוד (Packet Size Statistics)
- הזמן שעובר בין חבילה אחת לשנייה (Inter-Packet Times - IPTs)
- האם יש דפוס מסוים בסדר שבו נשלחות החבילות (Packet Ordering)
- מספר החבילות שנשלחות לכל כיוון (העלאה/הורדה (Uplink/Downlink Packet Count)
- שידור נתונים בקצב קבוע (Recurring Timing Patterns)
- משך הזמן שעובר עד שהשרת שולח נתונים ראשונים (Flow Duration)
- יחס בין תעבורת העלאה לתעבורת הורדה (Ratio of Uplink/Downlink Data)
- האלגוריתם אינו בודק מספר חבילות קבוע מראש, אלא מפסיק את ניתוח הזרימה ברגע שמתקבלת

- חבילת הנתונים הראשונה מהשרת (Dynamic Packet Selection)

מאפיינים של QUIC (HTTP/3)

- מזהה חיבור ב QUIC- (QUIC Connection ID).
- פרמטרים שונים של QUIC (QUIC Transport Parameters).
- שדות מחבילת ה Initial-הראשונה של QUIC ,כולל גרסת הפרוטוקול והגדרות הצפנה (QUIC Initial Packet Fields)
- מספר זרמים במקביל על אותו חיבור QUIC (QUIC Stream Multiplexing Patterns)
- שליחת אישורי ACK בקצב קבוע (QUIC Acknowledgment Timing)
- . QUIC מוסיף Padding כדי להסתיר את גודל ההודעות, ולכן משתמשים בזיהוי דפוסים סטטיסטיים (QUIC Padding) והשפעתו על הסיווג

3. מהם התוצאות העיקריות ומהן התובנות מהן?

תוצאות עיקריות:

- האלגוריתם hRFTC הצליח לזהות תעבורה מוצפנת בדיוק של 94.6% בהשוואה לדיוק של 38.4% בלבד בשיטות מסורתיות המבוססות רק על TLS .
- גם כאשר נעשה שימוש רק ב-10% מנתוני האימון האלגוריתם שמר על ביצועים גבוהים, מה שמעיד על יכולת הכללה טובה גם עם כמות נתונים מוגבלת.
- hRFTC פועל מהר יותר משיטות אחרות כיוון שהוא מסתמך על ניתוח של החבילות הראשונות בלבד ,ולא על כל הזרימה.
- האלגוריתם טוב יותר משיטות קיימות כמו UW, CESNET ו hC4.5- שלא מצליחות להתמודד ביעילות עם הצפנת ECH.

תובנות מרכזיות:

1. שיטות סיווג שמתבססות רק על TLS הופכות לפחות יעילות כאשר ClientHello מוצפן ,מה שמדגיש את הצורך לשלב מאפיינים סטטיסטיים של זרימה.
2. האלגוריתם מושפע מהאזור הגיאוגרפי שבו הוא אומן – דיוק הסיווג יורד כשהוא מיושם במדינה אחרת ,ולכן נדרש אימון מחדש לפי מיקום גיאוגרפי.
3. תוספת Paddingב QUIC- מקשה על זיהוי גדלי החבילות, אך האלגוריתם עדיין מצליח לזהות דפוסים סטטיסטיים שמאפשרים סיווג מדויק.
4. היכולת לזהות סוג תעבורה מוצפנת יכולה לשמש לשיפור אבטחת מידע ,למשל בזיהוי Botnets, תקשורת C&C של תוכנות כופר וניסיונות לעקוף חומות אש.
5. ניתן ליישם את השיטה גם בניהול איכות שירות (QoS), על ידי זיהוי תעבורה של סטרימינג, משחקי רשת ושיחות ,VoIP וכך לאפשר ניהול משאבי רשת יעיל יותר.
6. למרות היתרונות, היכולת לסווג תעבורה מוצפנת מעלה שאלות של פרטיות ,שכן ניתן לזהות אילו שירותים מוצפנים בשימוש מבלי לפענח את הנתונים עצמם.

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

1. מהי התרומה המרכזית של המאמר?

המאמר מציג שיטה מרשימה לזיהוי סוגי תעבורת אינטרנט מוצפנת על ידי המרת זרמי תעבורה לתמונות (*FlowPic*), ושימוש בטכניקות למידת עומק מתחום זיהוי התמונות, הם השתמשו ב- *Convolutional Neural Networks* (CNNs), לסיווג סוגי התעבורה והאפליקציות.

התרומות העיקריות כוללות:

- הפיכה נתונים לתמונות: במקום לנתח מספרים יבשים, לוקחים את **גודל החבילות וזמן ההגעה שלהן**, ממקמים אותם על גרף, ומקבלים תמונה שאפשר ללמוד ממנה דפוסים שלא היה אפשר לזהות קודם.
- דיוק בזיהוי תעבורה מוצפנת: השיטה מצליחה לזהות תעבורה בדיוק של מעל 96%, ותעבורה מוצפנת ב VPN-בדיוק של 99.2%, והצלחה בזיהוי תעבורה דרך Tor ברמת דיוק מעל 89%.
- יכולת לזהות אפליקציות חדשות: גם אם מופיעה אפליקציה שלא הייתה חלק מהאימון, המערכת עדיין מצליחה לשייך אותה לקטגוריה הנכונה.

2. אילו מאפייני תעבורה המאמר משתמש, ואילו מהם חדשים?

במקום לעבוד עם רשימות של נתונים סטטיסטיים, כמו בשיטות הישנות, המאמר מציע דרך חדשה להסתכל על תעבורה ברשת:

- גדל וזמן הגעה של החבילה: כל זרימה ברשת מנותחת לפי **גודל החבילות** והזמן שבו הן הגיעו, והם מתורגמים ל .FlowPic-ציר ה X-מייצג את זמן ההגעה של החבילות, וציר ה Y-את הגודל של החבילות .
- הפיכת הנתונים לתמונה דו-ממדית: המאפיין החדש המרכזי הוא המרת התעבורה לתמונה ,וכך לזהות דפוסים מורכבים באמצעות רשתות .CNN זה שונה משיטות אחרות שמסתמכים רק על מאפיינים סטטיסטיים .
- הפרדה לפי הצפנה: המערכת הצליחה להבחין בין סוגם שונים של תעבורה גם כאשר הם עוברים הצפנה באמצעות VPN או ,Tor וזה מאפשר לאפליקציות לתחת הצפנה כבדה.

3. מהם התוצאות העיקריות ומהן התובנות מהן?

תוצאות עיקריות:

- המערכת הצליחה לסווג תעבורה ב-דיוק של מעל 96% עבור תעבורת אינטרנט רגילה.
- הצליחה לזיהוי תעבורה מוצפנת ב VPN- ברמת דיוק של 99.2% ,ודיוק של מעל 89% לתעבורה שעוברת דרך .Tor
- שהרשת ניסה רק על תעבורה שאינה מוצפנת, הצליחה עדיין להשיג דיוק של בין 78.9%ל-99.4% בזיהוי תעבורת.VPN
- הצליחה לזיהוי אפליקציות כמו Skype, YouTube הצליח להשיג דיוק של .99.7%
הנתונים האלה גבוהים מדברים שהצליחו בעבר ומאוד מרשימים.

תובנות מרכזיות:

- השיטה טובה ומצליחה גם בתנאים קשים :השיטה מצליחה לסווג תעבורה מוצפנת ומורכבת, שזה שמצביע על הצלחה לעמוד מול הצפנה כמו VPN ו.Tor-
- הצליח לזהות אפליקציות חדשות :גם כשהאפליקציות חדשות ולא נכללו באימון והוסיפו אותם לבדיקה, המערכת הצליחה לזהות את סוג התעבורה במדויק.

- יתרון של הגישה החזותית :במקום לנסות לחפש דפוסים במספרים יבשים, השיטה **הופכת את הנתונים לתמונה ,**וזה מאפשר **לראות דברים שאי אפשר היה לגלות עם ניתוח מספרי רגיל**.

חלק 3:

הסבר על 7 הגרפים שפועלים על כל אפליקציה

הגרפים האלו מנתחים את התעבורה של כל אפליקציה בנפרד. המטרה היא לזהות את דפוסי השימוש ולסווג את האפליקציה לפי סוג התעבורה שהיא מייצרת.

1.**התפלגות גודל חבילות(Packet Size Distribution)**

מה הגרף מציג?

גרף זה מציג היסטוגרמה של גדלי החבילות (Packet Size) שנשלחו או התקבלו במהלך השימוש באפליקציה. הציר האופקי (X) מייצג את גדלי החבילות) בבתים ,(Bytes - והציר האנכי (Y) מציג את כמות החבילות שהיו בטווחי הגדלים השונים.

איך זה עוזר לסיווג האפליקציה?

- **אפליקציות צ'אט, HTTP, Web Browsing ,(כגון – Chrome, WhatsApp)**ישלחו חבילות קטנות יותר לרוב, כי הן שולחות בעיקר הודעות טקסט וקבצים קטנים.
- **אפליקציות סטרימינג ווידאו (כגון –YouTube, Zoom, Netflix)** יראו חבילות גדולות יותר באופן עקבי, כי הן מעבירות מדיה בפורמטים כבדים.
- **אפליקציות העברת קבצים (כגון – Google Drive, FTP)** יכולות להראות חבילות גדולות מאוד, אך גם קטנות בזמן תחילת החיבור.

2. התפלגות דגלי(TCP Flags Distribution) TCP

מה הגרף מציג?

גרף עמודות המציג את מספר הפעמים שכל דגל (Flag) של TCP הופיע בתעבורה של האפליקציה. הציר האופקי (X) מכיל את שמות הדגלים (SYN, ACK, FIN וכו), 'והציר האנכי (Y) מראה את מספר הפעמים שכל דגל הופיע.

איך זה עוזר לסיווג האפליקציה?

- **אם יש הרבה דגלי – SYN** האפליקציה מייצרת הרבה חיבורים חדשים, אופייני לגלישה באינטרנט (Chrome, Edge).
- **אם יש הרבה דגלי FIN ו – RST-**האפליקציה מבצעת סגירה של חיבורים לעיתים קרובות (יכול להצביע על בעיות חיבור או גישה מהירה לשרתים שונים).
- **אם יש יותר ACK מאחרים** –האפליקציה מבוססת על חיבורים מתמשכים כמו סטרימינג או שיחות וידאו.(Zoom, WhatsApp Calls)

3. סוגי לחיצות יד(TLS Handshake Type Distribution) TLS

מה הגרף מציג?

רף עמודות המציג את כמות כל סוג של Handshake בפרוטוקול TLS שבו השתמשה האפליקציה. הציר האופקי מציין את סוגי ה Handshakes (ClientHello, ServerHello -וכו),והציר האנכי מציג את מספר המופעים של כל סוג.

איך זה עוזר לסיווג האפליקציה?

- **אפליקציות מבוססות HTTP מאובטח – (Chrome, Edge, WhatsApp Web, YouTube)** יבצעו TLS Handshake רבים בתחילת החיבורים.
- **אפליקציות עם חיבורים ארוכים – (Zoom, Spotify, Google Meet)** יראו פחות TLS Handshakesכי הן שומרות על חיבור קיים לאורך זמן.
- **אם יש יותר מדי – TLS Handshake** יכול להצביע על בעיית חיבור מתחדשת או על שימוש מוגבר בפרוטוקולים מאובטחים.

4. התפלגות גרסאות(TLS Version Distribution) TLS

מה הגרף מציג?

גרף עמודות המציג את התפלגות גרסאות ה TLS-שבהן נעשה שימוש. הציר האופקי מציג גרסאות (TLS 1.0, 1.2, 1.3) והציר האנכי את מספר המופעים של כל גרסה.

איך זה עוזר לסיווג האפליקציה?

- **אם האפליקציה משתמשת ב TLS 1.3-בלבד** –מדובר באפליקציה מודרנית (לדוגמה Chrome, Edge, WhatsApp, Google Services)
- **אם יש שימוש ב – TLS 1.0/1.1-**מדובר באפליקציה ישנה יותר או שאינה מעודכנת (או שהאפליקציה מחוברת לשירות ישן).
- **שירותי סטרימינג ותקשורת רגישים כמו בנקאות** –ישתמשו תמיד בפרוטוקולים החדשים והבטוחים ביותר.

5. גרף סדרת זמן של גדלי החבילות(Time Series of Packet Sizes)

מה הגרף מציג?

גרף קו שמראה כיצד גודל החבילות משתנה לאורך זמן. הציר האופקי מציין את הזמן (Timestamp), והציר האנכי את גודל החבילות.

איך זה עוזר לסיווג האפליקציה?

- **אפליקציות עם קצב תנועה יציב כמו סטרימינג ושיחות וידאו (YouTube, Zoom, Google Meet)**–יראו דפוס עקבי של גודל חבילות דומה.
- **אפליקציות דינמיות כמו גלישה ודוא"ל – (Chrome, Gmail)** יראו שינויים חדים בהתאם לפעולות המשתמש.
- **אם יש ירידה פתאומית בגודל החבילות** –זה יכול להצביע על האטה ברשת או אובדן חבילות.

6. התפלגות זמני ביניים בין חבילות(Inter-Packet Time Distribution)

מה הגרף מציג?

גרף Boxplot שמראה את ההפצה של זמני השהייה בין חבילות ,(Inter-Packet Time) כלומר כמה זמן עבר בין שליחת חבילה אחת לשנייה.

איך זה עוזר לסיווג האפליקציה?

- **אפליקציות עם זמן קצר בין חבילות ,(Zoom, WhatsApp) סטרימינג** – (משדרות נתונים רציפים בזמן אמת.
- **אפליקציות עם פערים גדולים בין חבילות (דוא"ל, העברת קבצים)** –מעבירות מידע בפולסים גדולים יותר.
- **אם הפערים אינם עקביים** –יכול להצביע על בעיות רשת או שיבושים בחיבור.

7. יחס(TCP/UDP Ratio) TCP/UDP

מה הגרף מציג?

גרף עמודות המציג את כמות חבילות ה TCP-לעומת UDP שנשלחו בתעבורה של האפליקציה.

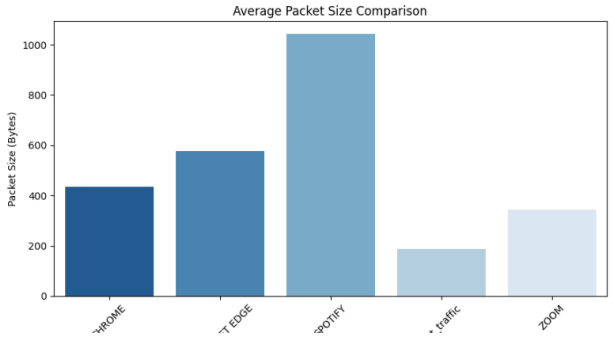
איך זה עוזר לסיווג האפליקציה?

- **אפליקציות עם יותר – (Zoom, WhatsApp Calls, Google Meet) UDP** פועלות על תקשורת מהירה ללא בדיקות תקינות קפדניות.
- **אפליקציות עם יותר ,(Chrome) TCP דוא"ל, העברת קבצים** – (דורשות מהימנות גבוהה ולכן שולחות חבילות.TCP
- **אם יש יחס גבוה של – UDP** סביר להניח שמדובר באפליקציה שמשדרת מידע בזמן אמת כמו שיחות וידאו או משחקים.

גרפי ההשוואה בין אפליקציות:

גרפים אלו מציגים נתונים השוואתיים בין אפליקציות שונות על פי מאפיינים מרכזיים של תעבורת הרשת שלהן. מטרתם היא לאפשר זיהוי דפוסי תקשורת ייחודיים לכל אפליקציה ולסווג אותן בהתאם.

1. השוואת גודל חבילה ממוצע(Average Packet Size Comparison)



מה מוצג בגרף?

גרף זה מציג את גודל החבילה הממוצע שנשלח בכל אפליקציה. לכל אפליקציה יש עמודה משלה, והגובה של כל עמודה מציין את הגודל הממוצע של החבילות) בבתים.(Bytes -

כיצד הנתונים מחושבים ?

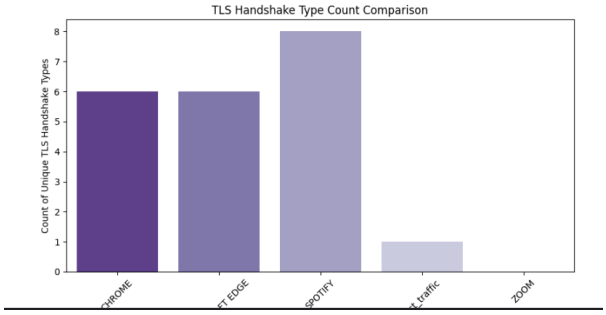
- לכל אפליקציה נאספים כל גדלי החבילות במהלך ההקלטה.
- מחשבים את הממוצע של גדלי החבילות.
- הנתונים מוצגים בגרף, בו ניתן לראות איזה אפליקציות מעבירות חבילות גדולות יותר לעומת כאלה שמעבירות חבילות קטנות יותר.

מה ניתן ללמוד מהגרף?

- אפליקציות שמעבירות בעיקר טקסט (כמו הודעות) יראו חבילות קטנות.
- אפליקציות שדורשות העברת קבצים או וידאו (כגון סטרימינג) יראו חבילות גדולות יותר.
- השוואה בין דפדפנים) כמו Chrome ו (Edge-יכולה להראות אם יש הבדלים ביעילות ניהול החבילות שלהן.

2. השוואת מספר TLS Handshake בכל אפליקציה(TLS Handshake Type Count Comparison)

מה מוצג בגרף?



גרף זה מציג את כמות ה TLS Handshakes-שהתבצעו בכל אפליקציה TLS Handshake .הוא תהליך יצירת חיבור מאובטח בין הלקוח (האפליקציה) לשרת.

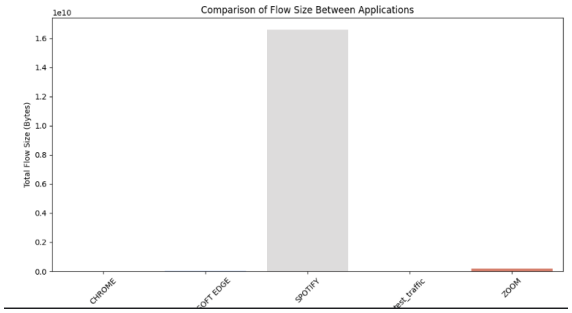
כיצד הנתונים מחושבים?

- לכל אפליקציה נספרים כל ה TLS Handshakes-שבוצעו.
- מוצגים בגרף העמודות, כך שניתן לראות איזה אפליקציות ביצעו יותר חיבורים מאובטחים.

מה ניתן ללמוד מהגרף?

- אפליקציות שמבצעות הרבה Handshakes הן לרוב אפליקציות גלישה כמו דפדפנים, אשר פותחות חיבורים רבים לאתרים שונים.
- אפליקציות סטרימינג כמו YouTube או Spotify יראו פחות Handshakes משום שהן מקימות חיבור ארוך ומתמשך ולא מתחברות מחדש בכל רגע.
- אפליקציות שדורשות אבטחה גבוהה, כמו אפליקציות פיננסיות, אמורות להראות מספר גבוה של חיבורים מאובטחים.

3. השוואת גודל הזרימה(Flow Size Comparison)



מה מוצג בגרף?

גרף זה משווה את כמות הנתונים הכוללת (Bytes - בבתים) שכל אפליקציה העבירה במהלך השימוש בה. כל עמודה מייצגת את כמות הנתונים שנשלחו והתקבלו על ידי האפליקציה.

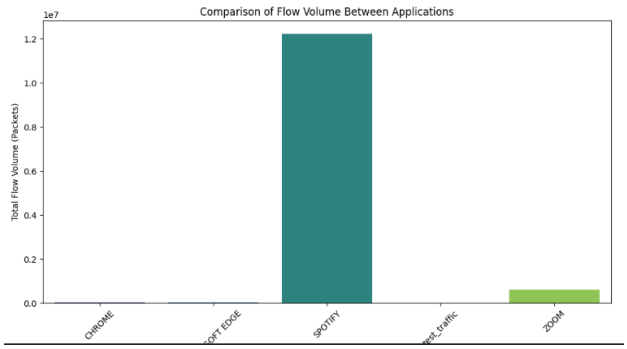
כיצד הנתונים מחושבים?

- לכל אפליקציה נסכמות כל החבילות שנשלחו ונמדד גודלן הכולל.
- הנתונים מוצגים בגרף כך שניתן לראות איזה אפליקציה צורכת יותר נתונים.

מה ניתן ללמוד מהגרף?

- אפליקציות עם תעבורת נתונים גבוהה הן לרוב אפליקציות סטרימינג (כגון YouTube ו.(Netflix-
- אפליקציות שיתוף קבצים או העברת וידאו כמו Zoom עשויות להראות ערכים גבוהים מאוד.
- אפליקציות קלות כמו דפדפנים יציגו זרימה נמוכה יחסית, אלא אם הן מבצעות הורדות כבדות.

4. השוואת נפח תעבורה(Flow Volume Comparison)



מה מוצג בגרף?

גרף זה מציג את כמות החבילות הכוללת שנשלחו והתקבלו על ידי כל אפליקציה. בניגוד לגרף הקודם, שבו נבדק גודל הנתונים הכללי, כאן בודקים את מספר החבילות ללא תלות בגודל שלהן .

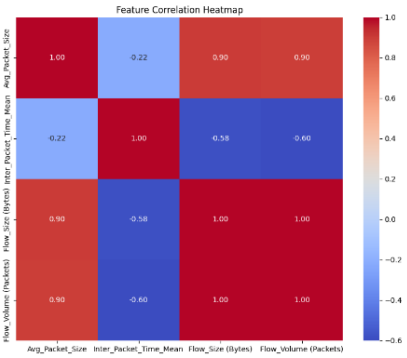
כיצד הנתונים מחושבים?

- לכל אפליקציה נספרות כל החבילות שהועברו במהלך השימוש בה.
- הנתונים מוצגים בגרף, בו ניתן לראות איזה אפליקציות יוצרות הכי הרבה חבילות בתקשורת.

מה ניתן ללמוד מהגרף?

- אפליקציות שמעבירות כמויות קטנות של נתונים בתדירות גבוהה) כגון WhatsApp או (Zoom יראו מספר גבוה של חבילות קטנות.
- סטרימינג ווידאו בדרך כלל משתמשים בפחות חבילות, אבל כל חבילה תהיה גדולה יותר.
- ניתן להבין אילו אפליקציות יוצרות עומס גבוה על הרשת מבחינת מספר החבילות ולא רק מבחינת נפח הנתונים.

5. מטריצת מתאם בין מאפייני תעבורה (Feature Correlation Heatmap)



מה מוצג בגרף ?

גרף זה מציג טבלת מתאם (Correlation Matrix) בין מאפיינים שונים של התעבורה (כגון גודל חבילה, זמן בין חבילות, גודל זרימה ונפח תעבורה). כל תא במטריצה מציג את רמת הקשר בין שני מאפיינים שונים.

כיצד הנתונים מחושבים?

- לכל זוג משתנים מחושב מתאם סטטיסטי (למשל, קורלציה של פירסון).
- הערכים מוצגים בצבעים שונים – צבעים חמים (אדום) מציינים קשר חזק, צבעים קרים (כחול) מציינים קשר חלש או הפוך.

מה ניתן ללמוד מהגרף?

- האם יש קשר בין גודל החבילות לבין זמן השהייה בין חבילות.
- האם מספר החבילות המועברות משפיע על גודל הזרימה.
- מאפשר לזהות מגמות חשובות בין מאפייני הרשת, ולסייע בסיווג אפליקציות על סמך מאפיינים משותפים.

סיכום

1. הגרפים האישיים של כל אפליקציה **עוזרים להבין איך היא פועלת ומה דפוסי השימוש שלה**.
2. הגרפים להשוואה **מאפשרים להבדיל בין אפליקציות שונות ולזהות סוגים שונים של תעבורה**.
3. אפשר להשתמש בנתונים **כדי לקבוע האם אפליקציה מסווגת כאפליקציה לצ'אט, סטרימינג, גלישה, משחקים וכו.'**

<u>הסבר מילולי של הגרפים:</u>

1. **גודל הזרימה(Flow Size)**
   - Spotify **ש**ולחת הרבה יותר מידע מאשר כל שאר האפליקציות**.**
   - Zoomשולחת מידע בכמות בינונית ,אך הרבה פחות מ.Spotify-
   - Chromeו Microsoft Edge-שולחים מעט מאוד נתונים בהשוואה ל Spotify-ול.Zoom-
2. **נפח הזרימה - (Flow Volume) מספר המנות שנשלחו**
   - Spotify שולחת את מספר המנות הגדול **ביותר** בהשוואה לכל שאר האפליקציות.
   - Zoom שולחת מספר מנות גדול יותר מהקלטה הקודמת, אך עדיין פחות מ**.**Spotify-
   - Chromeו Microsoft Edge-שולחים מעט מאוד מנות ,מה שמצביע על תעבורה נמוכה יותר.
3. **גודל ממוצע של מנות(Average Packet Size)**
   - Spotifyמשתמשת במנות גדולות בהרבה מאשר**.**Zoom
   - Microsoft Edgeו Chrome-משתמשים במנות בינוניות בגודלן.
   - Zoomשולחת מנות קטנות יחסית.
4. **שימוש בהצפנה(TLS Handshake)**
   - כל האפליקציות משתמשות בכמות דומה של חיבורים מוצפנים **,**כלומר כולן עומדות בסטנדרטים דומים של אבטחה.
5. **קשרים בין מאפיינים מתאם(Feature Correlation)–**
   - אפליקציות ששולחות יותר נתונים נוטות לשלוח יותר מנות**.**

- יישומים כמו Spotify שולחים מנות גדולות יותר בקצב יציב **,**בעוד Zoom שולחת הרבה מנות קטנות בפרקי זמן קצרים כדי לשמור על שידור חי וחלק.

מסקנות

- Spotify משתמשת בכמות הנתונים הגבוהה ביותר מכל היישומים שנבדקו.
- Zoom שולחת יותר מנות בהשוואה להקלטה הקודמת ,מה שמעיד על פעילות אינטנסיבית יותר בשיחה.
- דפדפנים כמו Chrome ו Edge-שולחים נתונים רק כאשר יש צורך בכך ,ולכן התעבורה שלהם קטנה בהרבה.
- כל האפליקציות משתמשות בפרוטוקולי אבטחה דומים ,ואין הבדלים משמעותיים בכמות ההצפנה שלהן.

לסיכום Spotify שולחת הרבה יותר נתונים מאשר ,Zoom בעוד ש Zoom-שולחת יותר מנות, אך קטנות יותר. דפדפנים משתמשים בפחות נתונים ומשדרים מידע רק כאשר יש בכך צורך.

<u>הסבר על המודל ותהליך הלמידה</u>

database בו השתמשנו בפרויקט נלקח מאתר ,**Kaggle** כפי שנכתב במייל. בחרנו להשתמש בנתונים אלו כי הם מכילים פרמטרים חשובים שמסייעים לסווג אפליקציות תעבורה רשת בצורה מדויקת. הנתונים מכילים מידע על תעבורת רשת שנעשתה דרך פרוטוקולים שונים, ובפרט נתונים שמסייעים להבחין בין סוגי אפליקציות שונים על פי התנהגות התעבורה שלהן ברשת.

הנתונים והפיצ'רים:

הנתונים שבחרנו כוללים פיצ'רים שיכולים להעיד על התנהגות של אפליקציות שונות ברשת, כגון אפליקציות סטרימינג, גלישה באתרים ועוד. כל אחד מהפיצ'רים שנבחרו נבחר בקפידה כדי לייצג את התעבורה ברשת בצורה שתאפשר למודל להבין ולהבדיל בין סוגי האפליקציות השונות.

המודל שבו השתמשנו בפרויקט הוא ,**Random Forest** שהוא אלגוריתם סיווג המשלב מספר עצי החלטה במודל אחד. כל עץ במודל לומד באופן עצמאי על חלק שונה מהנתונים ומחזיר תחזית. התוצאה הסופית של המודל מתקבלת על ידי חיבור התחזיות של כל העצים במודל, ובכך המודל מצליח למזער טעויות ומפחית את הסיכון להיתקל בבעיות של overfitting . הוא מצוין בזיהוי דפוסים בתעבורת רשת, ולכן בחרנו להשתמש בו בפרויקט שלנו, שכן הוא מבצע סיווג מדויק ומהיר, גם עם כמות גדולה של נתונים ומאפיינים. המודל מתאים במיוחד למשימות סיווג.

הסיווג של סוגי האפליקציות התבצע לפי אותיות מייצגות, וכל אחת מהן מתייחסת לסוג אפליקציה שונה, כפי שמצוין בנתונים:

- **D:** אפליקציות הקשורות לתעבורת דואר אלקטרוני .
- **L**: אפליקציות גלישה באתרים .
- M: אפליקציות לצפייה בסטרימינג וידאו.
- **P**: אפליקציות שיתוף קבצים .
- **U**: אפליקציות גיימינג אונליין .
- W: אפליקציות רשתות חברתיות .

כדי להתאים את הנתונים למודל ,Random Forest שדורש נתונים נומריים, היינו צריכים להמיר את כל אחת מהתוויות האלו למספרים. למשל 'D' ,הומרה ל-0 'L' ,ל-1 'M' ,ל-2 'P' ,ל-3 'U' ,ל-4 ו 'W'-ל-5.

הפיצ'רים שנבחרו ואיך הם תורמים למודל:

1. **Flow_Size:** גודל הזרם מציין את כמות המידע הכוללת שנשלחה בין המכשירים בשלב אחד של תקשורת. הוא חשוב כי הוא משקף את כמות המידע שנשלח ברשת. אפליקציות שונות מייצרות זרמים בגודל שונה.
2. **Flow_Volume:** מספר החבילות שנשלחו או התקבלו בזרם תעבורה נתון. נתון זה חשוב כי הוא מעיד על אינטנסיביות התקשורת. אפליקציות כמו סטרימינג, משחקים או פגישות וידאו ישלחו יותר חבילות, בעוד אפליקציות אחרות, כמו גלישה או שליחה של הודעות, ישלחו פחות חבילות. הפיצ'ר הזה מאפשר למודל להבחין בין תעבורה אינטנסיבית לבין תעבורה רגילה.
3. **Avg_Packet_Size:** גודל החבילה הממוצע שנשלחה ברשת. חשוב כי הוא יכול להעיד על סוג התעבורה. לדוגמה, שירותי סטרימינג (כגון וידאו או מוזיקה) ישלחו חבילות גדולות יותר בהשוואה לאפליקציות שיחה שבהן גודל החבילה יהיה קטן יותר. כך, המודל יכול לזהות סוגי תעבורה לפי גודל החבילות.
4. **Inter_Packet_Time_Mean:** הזמן הממוצע שעובר בין שליחת חבילה אחת לשנייה בזרם תעבורה. זה מציין את קצב התקשורת. אפליקציות שונות דורשות קצב שונה של תעבורה. לדוגמה, סטרימינג של וידאו או משחקים מקוונים דורשים קצב גבוה של חבילות שנשלחות בזה אחר זה, בעוד אפליקציות כמו גלישה באתרים עשויות לשלוח חבילות בקצב איטי יותר. הפיצ'ר הזה חשוב כדי להבחין בין אפליקציות שדורשות קצב גבוה ואפליקציות אחרות.

בהתאם למאפיינים בהם השתמשנו, היינו מצפים לראות את מאפייני האפליקציות המתאימים לכל סוג אפליקציה בצורה הבאה:

סוג אפליקציה	Flow_Size	Flow_Volume	Avg_Packet_Size	Inter_Packet_Time_Mean
D (דואר אלקטרוני)	נמוך	נמוך	קטן	גבוה
L (גלישה באתרים)	בינוני	בינוני	בינוני	בינוני
M (סטרימינג וידאו)	גבוה מאוד	גבוה מאוד	גדול	נמוך מאוד
P (שיתוף קבצים)	בינוני עד גבוה	בינוני עד גבוה	גדול	בינוני
U (גיימינג אונליין)	בינוני עד גבוה	משתנה	קטן עד בינוני	נמוך
W (רשתות חברתיות)	גבוה	גבוה	גבוה	נמוך מאוד

סטטיסטיקה:

לאחר שהמודל אומן על הנתונים שבחרנו, ביצענו סיווג של סוגי האפליקציות השונות על פי תעבורת הרשת. להלן התוצאות שהתקבלו:

המודל השיג דיוק כולל של 96% עם תוצאות טובות ברוב הקטגוריות. המדדים לכל קטגוריה, כולל דיוק, זיכרון ו F1-Score, מראים ביצועים חזקים.

מטרה ותוצאה:

במהלך הפרויקט, מטרתנו הייתה להכניס נתוני אפליקציות שהקלטנו ב Wireshark-ולראות אם המודל מצליח לסווג אותם כראוי. למרבה הצער, לא הצלחנו להטעין את נתוני ההקלטות לתוך המודל בצורה תקינה, עקב בעיות טכניות שקשורות למודל (המודל לא נשמר בקובץ). הבעיות הללו נבעו מקשיים הקשורים ללמידת מכונה. ביצענו בצורה מלאה את השלבים המודל מוכן לקבל נתונים משלנו, אך זה חלק שלא הצליח. החלטנו להתמקד במענה מקסימלי על המטלה בדרך מילולית וגרפית- כמו שהוסבר בהרחבה בקובץ זה.

באופן כללי, בסיכום המודל שלנו הוא מודל סיווג רשת שמבוסס על תכונות תעבורת הרשת - גודל הזרם, נפח הזרם, גודל החבילה הממוצע והזמן בין חבילות. כל אחד מהפיצ'רים נבחר בקפידה כדי לממש את המטרה של זיהוי סוגי האפליקציות השונות בצורה מדויקת.

<u>הסבר מתקפה על תעבורה מוצפנת או אנונימית</u>

נתמקד בזיהוי האפליקציות דרך ניתוח של דפוסים בתעבורה, גם כשהתעבורה מוצפנת או אנונימית. יש שתי אפשרויות לתוקף: האחת בה הוא יודע את כל פרטי החבילה והשנייה בה הוא יודע רק חלק מהמידע.

1- המתקפה עם ידע מלא על החבילה(Flow ID)

הידע שיש לתוקף:

- גודל החבילה (Packet Size)
- זמן הגעת החבילה (Timestamp)
- 4-Tupleשל ה – Flow ID-המזהה הייחודי של כל חיבור, הכולל את כתובת ה IP-של המקור והיעד, הפורטים של המקור והיעד, והפרוטוקול .

איך התוקף יכול לזהות את האפליקציה?

- סטרימינג של וידאו או אודיו :אפליקציות כמו Youtubeאו Spotifyשולחות חבילות גדולות ועם פחות הפסקות ביניהן. תוקף יכול לזהות דפוסים כאלה, כמו גודל החבילות) בין 1 MBל-5 (MBוהזמן הקצר בין החבילות, שמצביעים על סטרימינג.
- גלישה באתרים :דפדפנים כמו Chromeאו Firefoxשולחים חבילות קטנות יותר) בין 500 KBל-1 ,(MBעם יותר הפסקות ביניהן. תוקף יכול לזהות את הדפוס הזה ולנחש שמדובר בגלישה.
- שיחות וידאו :אפליקציות כמו Zoomשולחות חבילות בגודל משתנה, לפעמים חבילות קטנות אם מדובר באודיו בלבד, ולעיתים חבילות גדולות יותר אם יש גם וידאו. תוקף יכול לזהות את השוני בגודל החבילות ואת הגידול בהן כאשר הוידאו מופעל.
- חיבורים אחרים :תעבורה של HTTPו DNSלמשל תייצר חבילות קטנות עם זמני הגעה משתנים.

הסבר למה התוקף יכול להצליח בזיהוי:

- לכל אפליקציה יש דפוסים ייחודיים בתעבורה – גדלים שונים של חבילות, תדירותן והזמנים בין החבילות. אם התוקף יודע את פרטי ה ,Flow ID-הוא יכול בקלות להצליב את המידע הזה עם מאפיינים של אפליקציות ידועות ולזהות את הפעילות.

2- המתקפה עם מידע מוגבל (גודל החבילה וזמן הגעתה)

המידע שברשות התוקף:

- גודל החבילה
- זמן הגעת החבילה

איך התוקף יכול לזהות את האפליקציה?

- סטרימינג של וידאו/אודיו :חבילות גדולות, במיוחד כאלו שמגיעות בקצב גבוה ועם הפסקות קצרות, יכולות להעיד על סטרימינג.

- גלישה באתרים :חבילות קטנות יותר, עם הפסקות גדולות יותר ביניהן, עשויות להעיד על גלישה בדפדפן.
- שיחות וידאו :חבילות בגודל משתנה שיכולות להעיד על שיחות עם וידאו או אודיו בלבד.

הסבר למה התוקף יכול או לא יכול לזהות את האפליקציה:

- גם אם התוקף יודע רק את גודל החבילות והזמן בין כל חבילה, הוא יכול עדיין לזהות את האפליקציה ברמה גבוהה על פי הדפוסים, אבל לא באותה רמת דיוק כמו במתקפה עם כל המידע. כאשר יש הרבה חבילות עם גדלים קבועים ופערי זמן קבועים, התוקף יכול להסיק די בקלות מהן האפליקציות המעורבות, במיוחד אם הוא מכיר דפוסים של אפליקציות נפוצות.

האם התוקף תמיד יצליח לזהות את האפליקציה?

התוקף עשוי להיתקל בקשיים בזיהוי:

- אם האפליקציה משתמשת בהסוואה :חלק מהאפליקציות מבצעות שינויים דינמיים בתדירות ובגודל החבילות כדי להסתיר את פעילותן.
- הצפנה חזקה :אם התעבורה מוצפנת בצורה חכמה (לדוגמה, אם האפליקציה משנה את הפרוטוקולים באופן אוטומטי או מבצעת הצפנה חדשה כל הזמן), התוקף ייתקל בקושי רב יותר בזיהוי.
- שימוש ב VPN-או פרוקסי :במקרים אלו, המידע על כתובת ה IP-משתנה, מה שמקשה על התוקף לזהות את המשתמש.

דרכים להקטין את הסיכון לזיהוי

כדי להקטין את הסיכון לזיהוי של האפליקציות בשני המקרים, יש כמה אמצעים:

1. **שימוש בVPN-**: מסתיר את ה IP-של המשתמש ומפנה את התעבורה דרך שרת חיצוני. זה מקשה על התוקף לקשר את התעבורה לאפליקציה מסוימת.
2. **הצפנה מתקדמת** :**(TLS/SSL)** הצפנה מונעת מהתוקף לקרוא את תוכן החבילות.
3. **הוספת רעש:** הוספת חבילות ריקות או חבילות "רעש" לתעבורה יכולה להסתיר את הדפוסים האמיתיים ולהקשות על זיהוי האפליקציה.
4. **שימוש בפרוטוקולים משתנים** :אפליקציות יכולות לשנות את פרוטוקולי ההצפנה או את הפורטים של התעבורה מדי פעם, כך שהתוקף לא יכול לזהות דפוסים עקביים שמצביעים על אפליקציה מסוימת.

לסיכום, תוקף יכול לזהות את האפליקציות בהן השתמש המשתמש בתעבורה מוצפנת או אנונימית על ידי ניתוח המאפיינים של החבילות: גודלן, הזמן שעובר בין כל אחת, ודפוסים אחרים. גם אם התוקף יודע רק חלק מהמידע על החבילות, הוא עדיין יכול לזהות את האפליקציות ברוב המקרים. עם זאת, ניתן להקטין את הסיכון לזיהוי על ידי שימוש ב VPN, הצפנה מתקדמת, הוספת רעש לתעבורה ושימוש בטכנולוגיות כמו Tor. כל אלה מקשים על התוקף לזהות את האפליקציות ומבטיחים פרטיות גבוהה יותר.

מקרה קצה וטסטים:

בהרצת הבדיקות שלנו, בדקנו מגוון **מצבי קיצון** כדי להבטיח שהקוד מתמודד בצורה נכונה עם תרחישים שונים.

-**בדיקת קובץ חסר :**בדקנו מה קורה כאשר הקובץ pcapng. לא קיים בתיקיית הנתונים. המערכת מזהה את המקרה הזה ונותנת שגיאה מתאימה במקום לקרוס.

-**בדיקת קובץ ריק :**בדקנו איך הקוד מתמודד עם קובץ pcapng. ריק, כלומר כזה שאין בו כלל חבילות רשת. המערכת מזהה זאת ומדלגת על עיבוד הנתונים בצורה מסודרת.

- **בדיקת ניתוח חבילות תקינות :**הרצנו קובץ pcapng. עם תעבורת רשת אמיתית כדי לוודא שהקוד מנתח נכון את החבילות, מוציא את התכונות הרלוונטיות ושומר את הנתונים כראוי.

-**מניעת חלוקה באפס :**דאגנו לטפל במקרים שבהם חישובי **packet loss rate**עשויים להוביל לחלוקה באפס, כך שהקוד לא יתרסק במקרים כאלה.

- **בדיקות ניקוי נתונים :**ווידאנו שהמערכת יודעת להתמודד עם נתונים חסרים בעמודות שונות, ולבצע השלמות או הסרות לפי הצורך.

באמצעות בדיקות אלו, אנחנו מבטיחים **שהקוד יציב, עמיד למצבי קיצון, ומסוגל להתמודד עם נתוני רשת**

מקורות מידע:

במהלך הפרויקט נעזרנו במקורות מידע שונים לצורך איסוף נתונים, לימוד עקרונות עבודה ויישום שיטות ניתוח. בין המקורות בהם השתמשנו:

- **חומרי הקורס** –מצגות, הרצאות ומסמכים שסופקו במסגרת הפרויקט.
- **חיפושים באינטרנט** –מחקר עצמאי והעמקה בנושאים רלוונטיים.
- **מאמרים שסופקו לנו** –קריאה ויישום רעיונות מהמאמרים שהיו חלק מהפרויקט.
- **עזרה מה** – **ChatGPT -**שימוש בהכוונה טכנית ובהסברים תיאורטיים.
- **מאגרי מידע חיצוניים** – **(Kaggle)** במיוחד מתוך מאגר HTTPS Traffic Classification, ממנו השתמשנו בהקלטות לצורך אימון המודל והזנת נתונים.

Python packages שהשתמשנו בהם:

pyshark

pandas

numpy

matplotlib

seaborn

scikit-learn

tensorflow

joblib

os

sys

argparse

pathlib

logging

collections

קישור לREPO של ה GITHUB:

https://github.com/Alizanoa1234/Communication-Networks

דרך הרצת המודל:

כדי להריץ את המודל:

יש להוריד את הdataset מהאתר KAGGLE בקישור
https://www.kaggle.com/datasets/inhngcn/https-traffic-classification?resource=download

יש לשמור את הקובץ בשם .dataset.csv, בתיקיית processed_data

הוספת הקלטה לקוד:

כדי להריץ הקלטה על הקוד צריך להוסיף את הקלטת ה pcang לתיקיה של ה DATA.