ITRC
ICT Research Institute
(Iran Telecommunication Research Center)

Journal of Information and Communication Technology

Iranian Society of Information and Communication Technology

Volume 16, Issue 59-60, Spring and Summer 2024, pp. 264-278

Application Identification Through Intelligent Traffic Classification

Shaghayegh Naderi [1*]

[1] Assistant Professor, ICT Research Institute, Tehran, Iran

Received: 12 May 2023, Revised: 27 August 2023, Accepted: 17 September 2023
Paper type: Research

Abstract

Traffic classification and analysis is one of the big challenges in the field of data mining and machine learning, which plays an important role in providing security, quality assurance and network management. Today, a large amount of transmission traffic in the network is encrypted by secure communication protocols such as HTTPS. Encrypted traffic reduces the possibility of monitoring and detecting suspicious and malicious traffic in communication infrastructures (instead of increased security and privacy of the user) and its classification is a difficult task without decoding network communications, because the payload information is lost, and only the header information (which is encrypted too in new versions of network communication protocols such as TLS1.03) is accessible. Therefore, the old approaches of traffic analysis, such as various methods based on port and payload, have lost their efficiency, and new approaches based on artificial intelligence and machine learning are used in cryptographic traffic analysis. In this article, after reviewing the traffic analysis methods, an operational architectural framework for intelligent traffic analysis and classification has been designed. Then, an intelligent model for Traffic Classification and Application Identification is presented and evaluated using machine learning methods on Kaggle141. The obtained results show that the random forest model, in addition to high interpretability compared to deep learning methods, has been able to provide high accuracy in traffic classification (95% and 97%) compared to other machine learning methods. Finally, tips and suggestions about using machine learning methods in the operational field of traffic classification have been provided.

Keywords: Encrypted traffic classification, Operational architectural framework, Statistical features, Application Identification, Machine learning.

[*] Corresponding Author's email: naderi@itrc.ac.ir

دوفصلنامه
فناوری اطلاعات و ارتباطات ایران

سال شانزدهم، شماره های ۵۹ و ۶۰، بهار و تابستان ۱۴۰۳، صفحه ۲۶۴ الی ۲۷۸

شناسایی برنامه با طبقه‌بندی هوشمند ترافیک شبکه

شقایق نادری[1*]

[1] استادیار پژوهشگاه ارتباطات و فناوری اطلاعات، تهران، ایران

تاریخ دریافت: ۱۴۰۲/۰۲/۲۲ تاریخ بازبینی: ۱۴۰۲/۰۶/۰۵ تاریخ پذیرش: ۱۴۰۲/۰۶/۲۶

نوع مقاله: پژوهشی

چکیده

طبقه‌بندی و تحلیل ترافیک، یکی از چالش‌های بزرگ در حوزه داده کاوی و یادگیری ماشین است که نقش مهمی در تأمین امنیت، تضمین کیفیت و مدیریت شبکه دارد. امروزه حجم زیادی از ترافیک انتقالی در بستر شبکه توسط پروتکلهای ارتباطی امن مانند HTTPS رمز می‌شوند. ترافیک رمز، امکان نظارت و تشخیص ترافیک مشکوک و مخرب در زیرساخت‌های ارتباطی را (در قبال افزایش امنیت و حریم خصوصی کاربر) کاهش می‌دهد و طبقه‌بندی آن بدون رمزگشایی ارتباطات شبکه‌ای کار دشواری است، چرا که اطلاعات payload از دست می‌رود و تنها اطلاعات سرآیند که بخشی از آن هم در نسخه‌های جدید پروتکلهای ارتباطی شبکه (نظیرTLS1.03) رمز می‌شود، قابل دسترس است. از اینرو رویکردهای قدیمی تحلیل ترافیک مانند روش‌های مختلف مبتنی بر پورت و Payload کارآمدی خود را از دست داده، و رویکردهای جدید مبتنی بر هوش مصنوعی و یادگیری ماشین در تحلیل ترافیک رمز مورد استفاده قرار می‌گیرند. در این مقاله پس از بررسی روش‌های تحلیل ترافیک، چارچوب معماری عملیاتی برای تحلیل و طبقه‌بندی هوشمند ترافیک طراحی شده است. سپس یک مدل هوشمند با رویکرد شناسایی ترافیک برنامه‌ها مبتنی بر معماری پیشنهادی ارائه گردیده و با استفاده از روش‌های یادگیری ماشین روی مجموعه داده ترافیکی Kaggle141 و مجموعه داده محلی مورد ارزیابی قرار گرفته است. نتایج بدست آمده نشان می‌دهد که مدل مبتنی بر جنگل تصادفی، علاوه بر قابلیت تفسیرپذیری بالا در مقایسه با روش‌های یادگیری عمیق، توانسته است دقت بالایی در طبقه‌بندی هوشمند ترافیک (به ترتیب ۹۵٪ و ۹۷٪) نسبت به سایر روش‌های یادگیری ماشین روی مجموعه داده Kaggle141 و ترافیک محلی ارائه دهد.

کلیدواژگان: طبقه‌بندی ترافیک رمز، معماری عملیاتی، ویژگی‌های آماری، شناسایی برنامه، یادگیری ماشین.

* رایانامهٔ نویسنده مسؤول: naderi@itrc.ac.ir

۱- مقدمه

امروزه شاهد افزایش روزافزون تعداد کاربران اینترنت و ورود برنامه‌های متنوع تحت شبکه و موبایل هستیم که به مرور سهم بیشتری از ظرفیت خطوط ارتباطی شبکه اینترنت را به خود اختصاص می‌دهند و به تبع آن انواع ترافیک مخرب و تهدیدات امنیتی که امنیت کاربران را به خطر می‌اندازد نیز رو به افزایش است.

درخواست روزافزون استفاده از پهنای باند بیشتر از یک سو و محدودیت ظرفیت فیزیکی خطوط ارتباطی شبکه از سوی دیگر، سبب می‌شود که ارائه‌دهندگان خدمات اینترنت در پی یافتن راه‌کارهایی جهت بهبود کیفیت بهره‌برداری کاربران از منابع شبکه باشند. یکی از این راهکارها، طبقه‌بندی ترافیک و تخصیص پهنای باند مشخص به هر یک از برنامه‌ها و سرویس‌های موجود در شبکه و اولویت‌دهی به آن‌ها می‌باشد.

همچنین، با توجه به رشد روزافزون بدافزارها و تلاش آن‌ها برای پنهان‌سازی ترافیک خود و دور زدن سیستم‌های امنیتی، شناسایی و طبقه‌بندی ترافیک به عنوان گامی موثر و مقدمه‌ای لازم برای بسیاری از وظایف امنیتی، مدیریتی و کنترلی در شبکه دارای اهمیت است [1] و [2].

ازینرو در حال حاضر شناسایی ترافیک با اهداف مختلفی مانند: تامین کارایی شبکه، ارتقا کیفیت سرویس، تشخیص نفوذگران، مدیریت منابع، و کنترل دسترسی‌ها صورت می‌پذیرد [3].

یکی از چالش‌های اصلی در تحلیل ترافیک مواجهه با ترافیک رمز است. با افزایش استفاده از ترافیک رمز شده در لایه انتقال، رمزنگاری محتوا با استفاده از استانداردها بسیار رایج شده که این امر تحلیل و شناسایی ترافیک رمز را به یکی از چالش‌های این نوع ارتباطات تبدیل کرده است. بنابراین امروزه شناسایی ترافیک در بستر رمز ارتباطات اینترنتی، به عنوان یکی از رویکردهای اصلی تحلیل ترافیک مطرح است [4].

با بررسی مقالات متعدد، نکات زیر در خصوص تحلیل و تشخیص ترافیک رمز شده قابل ارائه است [5] تا [9]:

- امکان استفاده از اطلاعات ترافیک در فاز برقراری ارتباط برای تشخیص ترافیک رمز وجود دارد.
- ساختار پروتکل رمزنگاری می‌تواند در تشخیص نوع پروتکل ارتباطی شبکه موثر باشد.
- تشخیص پروتکل و نوع ترافیک عبوری می‌تواند مبتنی بر اندازه بسته و داده‌های مربوط به ساختار بسته در بخش‌هایی که رمز نشده‌اند (نظیر محتوای بسته، نسخه پروتکل و طول بسته ارسالی در زمان شروع ارتباط) صورت گیرد.
- مشخص نمودن ساختار بسته‌ها و استانداردهای ارتباطی در تشخیص و شناسایی رفتار آن‌ها جهت استخراج الگوی ارتباطی پروتکل‌ها مفید است.
- اگر چه بسته‌های ابتدایی در فاز برقراری ارتباط شامل سرآیندهای لایه ۲، ۳ و ۴ و در زمان دستداد[1]، پروتکل و شناسه نام سرور (SNI)، معمولا بصورت رمز نشده در دسترس هستند، اما در TLS1.3 این اطلاعات هم رمز می‌شود و در صورت استفاده از این نسخه، کارایی اغلب الگوریتم‌ها و روش‌های تحلیل ترافیک باید مورد بازبینی قرار گیرد.

از آنجائیکه در ترافیک رمزشده، امکان تحلیل و شناخت به دلیل رمزنگاری ارتباط وجود ندارد. تحلیل ترافیکی بهتر است بدون رمزگشایی الگوریتم‌های رمزنگاری و مبتنی بر ویژگیهای آماری بسته‌ها انجام شود تا زمان صرف شده برای تحلیل و طبقه‌بندی ترافیک کاهش یافته و قابل استفاده در TLS 1.3 نیز باشد.

در این مقاله پس از بررسی روش‌های تحلیل ترافیک در بخش ۲، چارچوب یک معماری عملیاتی برای تحلیل ترافیک برنامه‌های کاربردی به همراه ماژول‌ها و توابع اصلی و ارتباطات بین آنها در بخش ۳ ارائه شده است. در ادامه در بخش ۴ مدل هوشمندی با هدف شناسایی ترافیک برنامه‌ها در بستر شبکه، مبتنی بر معماری پیشنهادی، ارائه گردیده و توسط روش‌های یادگیری مختلف پیاده‌سازی و مورد ارزیابی قرار گرفته است. در نهایت نکات و پیشنهاداتی جهت بکارگیری مؤثر روش‌های یادگیری ماشین در کاربردهای عملی تحلیل ترافیک در بخش ۵ ارائه شده است.

نوآوری‌های فعالیت پیش رو شامل موارد زیر است:

- *ارزیابی قابلیت استفاده از روش‌های مبتنی بر هوش مصنوعی به جای روش‌های موجود مبتنی بر قانون:* با توجه به اینکه در حال حاضر تمام روش‌های مورد استفاده در سامانه‌های مختلف شناسایی ترافیک داخلی مبتنی بر امضاهای خریداری شده عمل می‌کنند، مهمترین هدف و نوآوری این پروژه امکان‌سنجی استفاده از هوش مصنوعی برای استخراج الگوی ترافیکی برنامه‌های جدید در راستای کاهش وابستگی به نرم‌افزارهای خارجی (مبتنی بر قانون و بالطبع نیازمند خرید امضاهای جدید) در این حوزه است.

[1] Handshake

- *طراحی معماری عملیاتی سامانه شناسایی ترافیک:* در گام اول پروژه مطالعه و بررسی جامعی در حوزه تحلیل ترافیک روی مقالات و مراجع معتبر انجام گرفت (که خلاصه آن در جدول ۱ مقاله [20] قابل دسترس است). در این مقاله مبنی بر دانش اخذ شده و تحلیل نیازمندی‌ها، توابع اصلی مورد نیاز در شناسایی ترافیک شناسایی و در قالب پنج ماژول اصلی در طراحی معماری سامانه شناسایی ترافیک بکار گرفته شد.
- *سفارشی‌سازی مجموعه داده* Kaggle141: مشتمل بر نرمال‌سازی، انتخاب دسته‌ها و ویژگی‌های مناسب
- *تولید نمونه آزمایشگاهی ترافیک واقعی:* (شامل بررسی روال‌های تولید مجموعه داده، استخراج ویژگی و نرمال‌سازی آن، در راستای استفاده کاربردی از مدل پیشنهادی در کاربردهای بومی و سیستم‌های موجود)
- پیاده‌سازی نمونه عملیاتی سامانه تشخیص ترافیک: با قابلیت تست روی داده واقعی و آنلاین
- قابلیت بکارگیری روش پیشنهادی در کنار سامانه‌های مبتنی بر قانون/امضاء: جهت استخراج امضای برنامه‌های جدید به صورت آفلاین

با توجه به اینکه سیستم‌های تشخیص ترافیک موجود در کشور تماما مبتنی بر قانون هستند و با توجه به اینکه ترافیک برنامه‌ها مدام در حال تغییر است، هدف اصلی از این پژوهش ارزیابی این موضوع بوده که آیا روش‌های هوشمند می‌توانند در کاربردهای واقعی جایگزین روش‌های مبتنی بر قانون شده و یا به صورت موازی برای تولید قوانین جدید در کنار این سامانه‌ها قرار بگیرند یا خیر؟ که با توجه به ارزیابی صورت گرفته و سرعت پاسخ مدل پیشنهادی نتایج امیدوارکننده بوده است.

۲- روش‌های تحلیل ترافیک

برای تحلیل و طبقه‌بندی ترافیک شبکه با اهداف مختلف از جمله: شناسایی بدافزار [10] و نوع سیستم عامل، شناسایی ردپا[1] [11] و تاثیر فعالیت کاربر [2]، شناسایی موقعیت جغرافیایی، شناسایی برنامه‌های جعل و فریب [12]، شناسایی برنامه کاربردی خاص و یا تشخیص علاقه‌مندی کاربر، روش‌های متفاوتی وجود دارد. به طور کلی روش‌های تحلیل و طبقه‌بندی ترافیک در سه دسته کلی ارائه می‌شوند [13] تا [16]:

- روش‌های مبتنی بر پورت[2]
- روش‌های مبتنی بر بار/ محتوای بسته[3]
- روش‌های مبتنی بر ویژگی‌های آماری[4]/ رفتاری[5]

در جدول ۱ روش‌های تحلیل ترافیک همراه با مزایا و معایب هر یک ارائه شده است.

جدول ۱. مقایسه روش‌های مختلف تحلیل ترافیک

روش	مزایا و قابلیت‌ها	معایب و کاستی‌ها
مبتنی بر پورت	نسبتا ساده و سریع (بدلیل استفاده از پورت‌های معروف برای تشخیص)	ناکارامدی هنگام استفاده از پورت‌های آزاد (پورت پویا در برنامه‌های P2P) ظهور تکنیک و پروتکل تونل‌سازی (ناکارامدی روش‌های مبتنی بر پورت)
مبتنی بر محتوا (بار) بسته	تحلیل محتوای بسته و تشخیص مبتنی بر الگو صحت تشخیص بالا تشخیص ترافیک رمز با بررسی بایت‌های آغازین ارتباط استفاده از ابزار DPI	سرعت پایین به علت محاسبات بالا نیاز به سخت‌افزار قوی برای جستجوی الگوهای از پیش تعیین شده لزوم بروزرسانی قوانین و الگوها عدم دسترسی به محتوا بسته‌های رمز محدودیت تشخیص در ترافیکهای جدید (بدون الگو و امضا)
مبتنی بر ویژگی آماری/ رفتاری	دقت بالا و پوشش گسترده امکان تفسیر تلفیق ویژگی آماری و ML در تشخیص ترافیک رمز استفاده از ویژگی‌های رفتاری در تشخیص ترافیک خاص	سرریزهای بزرگ محاسباتی و حافظه مربوط به استخراج و انتخاب ویژگی تاخیر در تحلیل ترافیک در زمان واقعی (با استفاده از کل دنباله بسته‌ها) نیاز به کاهش ابعاد و انتخاب ویژگی‌ها

همانطور که در جدول اشاره شده با توجه به ظهور تکنیک‌های جدید روش‌های مبتنی بر پورت دیگر کارایی لازم را ندارند. در تکنیک مبتنی بر محتوا، شناسایی بر اساس بازرسی عمیق [14] و [17] محتوای بسته‌ها صورت گرفته و با استفاده از الگوها و ساختارهای از پیش تعیین شده، طبقه‌بندی و شناسایی نوع ترافیک انجام می‌شود. این روش‌ها از صحت تشخیص بالایی برخوردار هستند. با این حال، بازدهی پایینی دارند که علت آن نیاز به محاسبات زیاد و سخت‌افزار قوی برای جست‌وجوی الگوها، به روزرسانی مکرر قوانین تطبیق و مشکلات حریم خصوصی است. علاوه بر این، تعداد فزاینده‌ای از برنامه‌ها، محتوای بسته‌های ارسالی خود را رمزگذاری می‌کنند و باعث ناکارامدی این روش در شناسایی ترافیک می‌شوند. از اینرو روش‌های مبتنی بر یادگیری ماشین با ارائه مدل‌های هوشمند مبتنی بر ویژگی‌های آماری/ رفتاری به عنوان روش‌های پرکاربرد و موثر برای طبقه‌بندی و تحلیل ترافیک در بستر رمز شبکه مطرح

[1] Foot Print
[2] Port-Based
[3] Payload-Based
[4] Statistical-Based
[5] Behavioral-Based

شدند [18] و [19]. این روش‌ها علاوه بر دقت بالا در تشخیص الگوها و تعیین رابطه‌های میان ورودی‌ها و خروجی‌ها، می‌توانند الگوهای پیچیده و مختلف را تشخیص دهند. همچنین روش‌های یادگیری ماشین مبتنی بر ویژگیهای آماری/ رفتاری می‌توانند به صورت آنلاین و در زمان واقعی برای طبقه‌بندی و تحلیل داده‌ها مورد استفاده قرار گرفته و قابلیت تفسیر داده‌ها و مدل‌ها را برای کاربران و مدیران فراهم سازند.

۳- مدل پیشنهادی

شناسایی ترافیک با رویکردهای مختلفی صورت می‌گیرد که در دسته‌های کاربردی زیر قابل تفکیک هستند:

- طبقه‌بندی ترافیک
- شناسایی برنامه‌های کاربردی
- شناسایی گروه خدماتی
- تفکیک دو گروه از هم
- شناسایی رفتارهای خاص کاربر
- شناسایی پروتکل‌ها

در این مقاله، کاربرد مدنظر شناسایی برنامه‌های کاربردی است.

اولین و مؤثرترین گام در پیاده‌سازی موفق یک مدل، طراحی دقیق معماری آن است. انتخاب نحوه جمع‌آوری داده به صورت آنلاین/ آفلاین، تعیین ابزار و تجهیزات انتخابی، سطح و لایه جمع‌آوری ترافیک، تعیین نوع و فرمت ویژگیهای جریان ترافیکی از جمله چالشهای مطرح در حوزه تحلیل ترافیک است [13] که بسته به نوع مسأله، حوزه کاربرد آن، میزان هزینه‌کرد و سایر پارامترها باید قبل از هر اقدامی در مورد آن تصمیم‌گیری کرد.

هدف از ارائه معماری، در واقع ارائه توابع و ماژول‌های اصلی یک سامانه و تعیین ارتباطات بین آنها مستقل از جزئیاتی نظیر نوع فناوری، چگونگی اجرا، و توسعه کد است. به این ترتیب هر نوع تغییری که بر اساس توسعه و تغییرات فناوری‌ها و نیازمندی‌ها و تغییرات نرم‌افزار و برنامه‌ها در آینده وجود داشته باشد، درون ماژول‌های اصلی قابل انجام است.

برای طراحی معماری، پس از مطالعه و بررسی معماری‌های ارائه شده در بیش از ۱۲۰ مرجع و مقاله معتبر (که جزئیات آن به صورت جامع در مقاله [20] قابل دسترس است) نیازمندیهای یک سامانه تحلیل ترافیک از جنبه‌های مختلف مورد بررسی قرار گرفته و بر این اساس ماژول‌های پنجگانه زیر برای آن پیشنهاد شد:

- ماژول ورودی
- ماژول پیش پردازش و استخراج ویژگی
- ماژول تحلیل مبتنی بر یادگیری ماشین
- ماژول مجموعه داده (دیتاست) و مخزن داده/ دانش
- ماژول خروجی

در ادامه توابع و جزئیات هر ماژول و نحوه ارتباط آنها در قالب طرح معماری ارائه شده است.

۳-۱- ماژول ورودی / خروجی

ماژول دریافت داده می‌تواند از طریق دو رویکرد انجام پذیرد.

- دریافت آنلاین داده با استقرار بر ترافیک عبوری
- دریافت آفلاین داده با دریافت کپی ترافیک عبوری

داده ورودی بسته‌های عبوری جریان ترافیک شبکه هستند که در فرمت **pcap.** یا **csv.** دریافت شده و قابلیت تبدیل به سایر فرمت‌ها جهت استخراج ویژگی را دارند.

ماژول خروجی شامل نوع برنامه کاربردی تشخیص داده شده توسط الگوریتم یادگیری ماشین می‌باشد. این خروجی‌ها در فرمت‌ها و روش‌های مختلف قابل ثبت و نمایش هستند. همچنان که بسته فایل‌های ورودی در فرمت **pcap** است، خروجی‌ها نیز می‌تواند بسته به نیاز جهت ارزیابی و محاسبه پارامترهای کارایی در فرمت‌های مختلف متنی یا اکسل ذخیره شود.

۳-۲- ماژول داده و دانش

در این بخش معماری عملیاتی مجموعه داده بررسی می‌شود. این ماژول شامل سه بخش زیر است که در بخش اول و دوم، داده ترافیکی متناسب با نیازمندی‌های تحلیل پردازش و آماده می‌شود.

- پایگاه داده خام: که یکی از ملزومات اصلی در کاربردهای واقعی تحلیل ترافیک محسوب می‌شود و شامل مراحل ضبط ترافیک[1] و برچسب‌گذاری[2] آن است.
- پایگاه داده پردازش شده: که شامل مجموعه داده آماده شده برای آموزش، اعتبارسنجی و تست مدل است.
- پایگاه دانش: که در برگیرنده دانش مدلهای آموزش یافته و پارامترهای نهایی آنها می‌باشد.

هر یک از پایگاه‌های داده/ دانش بسته به مدل انتخابی و روند تحلیل در جای خود مورد استفاده قرار خواهد گرفت.

[1] Capture

[2] Labeling

۳-۳- ماژول پیش پردازش

ماژول پیش پردازش وظیفه تنظیم و آماده‌سازی داده ترافیکی جهت استفاده در ماژول تحلیل را بر عهده دارد. وظایف این ماژول بشرح زیر است:

- ارزیابی کیفی داده: بطوریکه داده‌های معیوب، ناشناس و غیرنرمال به درون سامانه پردازشی هدایت نشوند و داده‌هایی که معرف ویژگی‌های مرتبط با برنامه‌ها است و متناسب با استانداردها تولید شده اند، برای تحلیل ارسال شوند.
- حذف داده بدون ارزش: حذف داده‌های ترافیکی که ارزش افزوده دانشی و اطلاعاتی از منظر تحلیل ندارند.
- حذف داده نویزی: حذف داده‌های نویزی که بر روند تحلیل تاثیر مخرب دارند.
- نرمال‌سازی: با توجه به تفاوت فرمت و بازه متغیرها در نرم‌افزارهای مختلف جمع‌آوری ترافیک، لازم است کلیه داده‌های دریافتی (از انواع منابع و برنامه‌های کاربردی) در یک بازه تعریف شده نرمال و فرمت‌ها یکسان شود.
- کاهش ابعاد (**Reduction**): گاهی اطلاعاتی درون فایل ورودی هست که فاقد اطلاعات مفید بوده و یا بدلیل مباحث حفظ حریم خصوصی باید حذف شود. با حذف این اطلاعات (نظیر **IP masking**) حجم داده/فایل‌های ورودی کاهش یافته و در نهایت داده جهت ورود به ماژول تحلیل آماده می‌شود.

۳-۴- ماژول تحلیل

این ماژول، بسته به رویکرد تحلیل ترافیک می‌تواند توابع مختلفی از موارد زیر را در برگیرد:

- استخراج ویژگی‌ها: ابتدا نیازمند استخراج ویژگی‌هایی هستیم که به هر دسته ترافیکی تعلق دارد و از این ویژگی‌ها می‌توان با قطعیت بالا، ترافیک مربوطه را تشخیص داد. جدول ۲ نمونه‌هایی از ویژگیهای آماری مورد استفاده جهت شناسایی ترافیک را نشان می‌دهد.
- کاهش ویژگی‌ها: حذف داده‌ها و ویژگی‌هایی که در کارایی روش‌های تشخیص و طبقه‌بندی مبتنی بر یادگیری ماشین تاثیر چندانی ندارند و گاها منجر به ایجاد خطا در شناسایی می‌شوند (مانند حذف آدرس IP مبدأ و مواردی که مربوط به حریم خصوصی کاربر می‌شود)، در این مرحله انجام می گیرد.
- تبدیل ویژگی‌ها[1]: تبدیل فضای ویژگی‌ها بسته به روش یادگیری ماشین انتخابی و روش کاهش ویژگی‌ها ممکن است مورد نیاز باشد.
- آموزش مدل: در این بخش بسته به هدف تحلیل ترافیک، نوع و مدل روش هوشمند انتخاب خواهد شد که می‌تواند روش‌های یادگیری ماشین یا یادگیری عمیق باشد. برای انجام این کار، بخشی از داده ورودی را برای تنظیمات مربوط به آموزش مدل استفاده خواهیم کرد (فاز یادگیری / آموزش) که در آن سامانه ویژگی‌ها، فضا و تنظیماتی را که برای تشخیص هر کدام از خروجیها نیاز دارد یاد می‌گیرد، سپس با بخش دوم داده‌ها مدل آموزش یافته را مورد ارزیابی قرار می‌دهیم (فاز تست) و در نهایت بر اساس نتایج ارزیابی مدل را بهبود می‌دهیم.
- به عبارتی آموزش مدل شامل سه مرحله: آماده‌سازی داده آموزشی، پیاده‌سازی مدل یادگیری و تصحیح خطا و بهینه‌سازی مدل می‌باشد.
- شکل ۱ معماری پیشنهادی برای تحلیل ترافیک را نشان می‌دهد که ماژول تحلیل با دو رویکرد استفاده از روش‌های یادگیری ماشین و یادگیری عمیق ارائه شده است.
- با توجه به قدرت تفسیرپذیری روش‌های یادگیری ماشین نسبت به به روش‌های یادگیری عمیق [20] و [21]، در این مقاله رویکرد یادگیری ماشین برای پیاده‌سازی مدل استفاده شده است.

۴- پیاده‌سازی و ارزیابی روش پیشنهادی

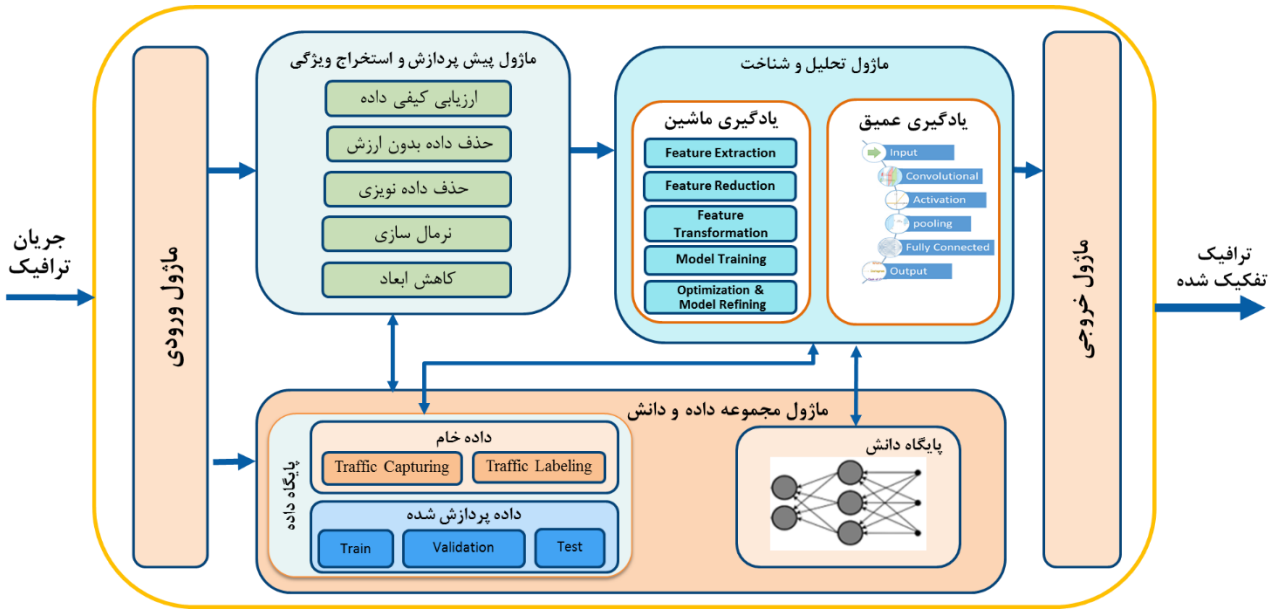
در این بخش به بیان مسأله، مدل پیشنهادی، معرفی پایگاه داده، روش‌های یادگیری ماشین مورد استفاده پرداخته می‌شود.

در معماری شبیه‌سازی شده دراین مقاله ماژول ورودی، ترافیک رمز برنامه‌های مختلف در قالب یک فایل csv است که دربرگیرنده ویژگی‌های جریان ترافیک است.

در ماژول پیش‌پردازش یکسان‌سازی پارامترها و حذف داده‌های نویزی صورت گرفته و اطلاعات پس از نرمال‌سازی در اختیار ماژول تحلیل قرار می‌گیرند. مبتنی بر مطالعات و تحقیقات قبلی انجام گرفته [20]، تعدادی از روش‌های یادگیری ماشین متناسب با مسأله برای ماژول تحلیل کاندید شده و توسط معیارهای ارزیابی روی ترافیک ورودی مورد ارزیابی قرار می‌گیرند.

در ادامه جزئیات مجموعه داده و مدل‌های یادگیری انتخابی و نحوه ارزیابی آنها تشریح شده است.

[1] Feature Transformation



شکل ۱. معماری عملیاتی پیشنهادی برای سامانه شناسایی ترافیک

۴-۱- مجموعه داده

پیاده‌سازی مدل نیاز به داشتن مجموعه کاملی از داده‌ها با ویژگی‌های تعیین‌کننده دارد. برای انتخاب مجموعه داده، انواع مجموعه داده ترافیکی موجود از جنبه‌های مختلف مانند: نوع ترافیک و انواع دسته‌بندی ترافیک موجود، تنوع و حجم ترافیک به ازای هر نرم‌افزار، امکان دسترسی به مجموعه داده و ... مورد بررسی قرار گرفته و مجموعه داده Kaggle141 برای آموزش و ارزیابی عملکرد روش‌های مختلف یادگیری ماشین (طبقه‌بندهای منتخب) با هدف شناسایی ترافیک رمز برنامه‌ها انتخاب شد [22].

البته به صورت موازی و با هدف کاربردی نمودن طرح در گام بعدی و ایجاد مجموعه داده ترافیکی بومی، پروسه ایجاد یک مجموعه داده محلی شامل ترافیک چند نمونه برنامه کاربردی در محیط آزمایشگاهی (مبتنی بر بررسی ابزارها و روالهای ایجاد مجموعه داده‌های استاندارد) آغاز گردید که پس از نرمال‌سازی، استخراج ویژگی و پیش پردازش در کنار مجموعه Kaggle برای ارزیابی روش پیشنهادی مورد استفاده قرار گرفته است.

Kaggle141 مشتمل بر **۲۳۴۴۵۳۴** رکورد و **۴۶** ویژگی پایه و آماری از جریان‌های ترافیکی جمع‌آوری شده در لایه شبکه در قالب فایل

جدول ۲. ویژگی‌های آماری پکت‌ها

• طول یک بسته	• تعداد بایت جریان ارسالی/دریافتی
• تعداد پکت/ بایت در ثانیه	• نرخ بایت‌های جریان دریافت شده
• انحراف از مقدار پرتکرار[1]	• میانگین طول بسته
• اختلاف زمان[2] درخواست و پاسخ	• میانه طول بسته
• واریانس زمان درخواست و پاسخ	• مقدار پرتکرار طول بسته
• انحراف معیار[3] اختلاف زمان درخواست و پاسخ	• واریانس طول بسته
• ضریب تغییر[4] اختلاف زمان درخواست و پاسخ	• انحراف معیار طول بسته
• انحراف از میانه[5] اختلاف زمان درخواست و پاسخ	• ضریب تغییر طول بسته
• انحراف از میانه زمان بسته	• انحراف از میانه طول بسته
• انحراف از مقدار پرتکرار زمان بسته	• انحراف پرتکرار طول بسته
• میانگین اختلاف زمان درخواست و پاسخ	• میانگین زمان بسته
• میانه اختلاف زمان درخواست و پاسخ	• میانه زمان بسته
• مقدار پرتکرار اختلاف زمان درخواست و پاسخ	• مقدار پرتکرار زمان بسته
• نرخ[6] بایت جریان ارسالی	• ضریب تغییرات زمان بسته
• واریانس زمان بسته	• میانگین فاصله زمانی[7] ارسال بسته‌ها
• حداقل/حداکثر طول بسته‌های ارسالی	• انحراف معیار زمان بسته
• حداقل/اکثر فاصله زمانی ارسال بسته‌ها	• میانگین طول بسته‌های ارسالی

[1] Mode
[2] Request-response time difference
[3] Standard deviation
[4] Coefficient of variation
[5] Skew from median
[6] Rate
[7] Inter-arrival time

CSV است که در یک شبکه دانشگاهی در مدتی معین و در ساعات مشخص و خاص از یک روز (برای سطح سازمانی یا کوچکتر) جمع‌آوری شده است. این مجموعه داده یکی از مجموعه‌های معتبر جهت تحلیل ترافیک با هدف شناسایی پروتکل و دسته‌بندی سرویس‌های تحت وب است که برای هر رکورد از داده‌های ترافیکی دو یا سه سطح برچسب به نحو زیر ارائه می‌دهد:

- سرویس مبتنی بر وب: نظیر آمازون، گوگل، msn، DNS، واتساپ، NetBIOS، windows update، Facebook، yahoo، http، مایکروسافت و دسته نامعین
- گروه: شامل وب، شبکه، سیستم عامل، بروزرسانی نرم‌افزار و دسته نامعین
- پروتکل: شامل Http، TLS، DNS و غیره

که در این مقاله (با توجه به هدف و تعریف مسأله) از برچسب‌های دسته اول استفاده شده است.

به منظور بررسی داده‌های مجموعه داده فوق، ابتدا فایل‌های csv مجموعه داده مورد بررسی قرار گرفت. با توجه به امکانات محدود excel برای گزارش‌گیری و تعداد رکورد، مجموعه داده‌های یاد شده در سامانه مدیریت پایگاه داده اوراکل بارگذاری و از امکانات گزارش‌گیری آن جهت تحلیل مجموعه داده استفاده شد.

پس از بارگذاری مجموعه داده روی پایگاه داده اوراکل، گزارش گروه‌بندی چندگانه توسط برآیند سه برچسب گرفته شد و جدولی با ۳۲۴ ردیف (دسته) بدست آمد که می‌توان جهت درک چگونگی جداسازی و شناسایی ترافیک از آن بهره برد. همچنین بررسی دسته‌های مختلف ترافیکی و پروتکل‌های هر دسته نشان دهنده میزان ترافیک شناسایی نشده و ترافیک‌هایی است که علی رغم آن که در یک پروتکل هستند می‌توانند در وب سرویس‌های متفاوتی دسته‌بندی شوند. جدول ۳ نمونه‌ای از ردیفهای مربوط به نرم‌افزار Xbox را نشان می‌دهد. شماره ردیف‌ها از جدول اصلی برایند برچسب‌ها آمده تا قابل رهگیری باشند.

جدول ۳. نمونه برآیند برچسب‌های سه گانه Kaggle برای برنامه Xbox

ردیف	دسته	پروتکل	وب سرویس	تعداد
80	Game	DNS	Xbox	576
83	Game	HTTP	Xbox	102
87	Game	TLS	Xbox	438
91	Game	Unknown	Xbox	3

همانطور که مشاهده می‌شود ۵۷۶ جریان داده در دسته بازی با پروتکل DNS، ۱۰۲ جریان داده در دسته بازی با پروتکل HTTP، ۴۳۸ جریان داده در دسته بازی با پروتکل TLS، و در نهایت ۳ جریان داده در دسته بازی با پروتکل ناشناخته مربوط به ترافیک Xbox تفیک شده است. این تفکیک به خوبی نشان از تفکیک ترافیک در ابعاد مختلف داشته و همچنین تعداد جریان‌های داده در هر بخش می‌تواند راهنمای خوبی برای نوع ترافیک، رمزنگاری و سایر ابعاد ترافیکی باشد.

جدول ۴ برنامه‌های انتخابی از مجموعه Kaggle 141 با تمرکز بر وب سرویس‌های بازی و شبکه اجتماعی را به همراه تعداد رکوردهای هر وب سرویس نشان می‌دهد که در این مقاله به عنوان مجموعه هدف انتخاب و برای ارزیابی عملکرد مدلهای مختلف یادگیری ماشین در رویکرد شناسایی برنامه‌ها مورد استفاده قرار گرفته‌اند.

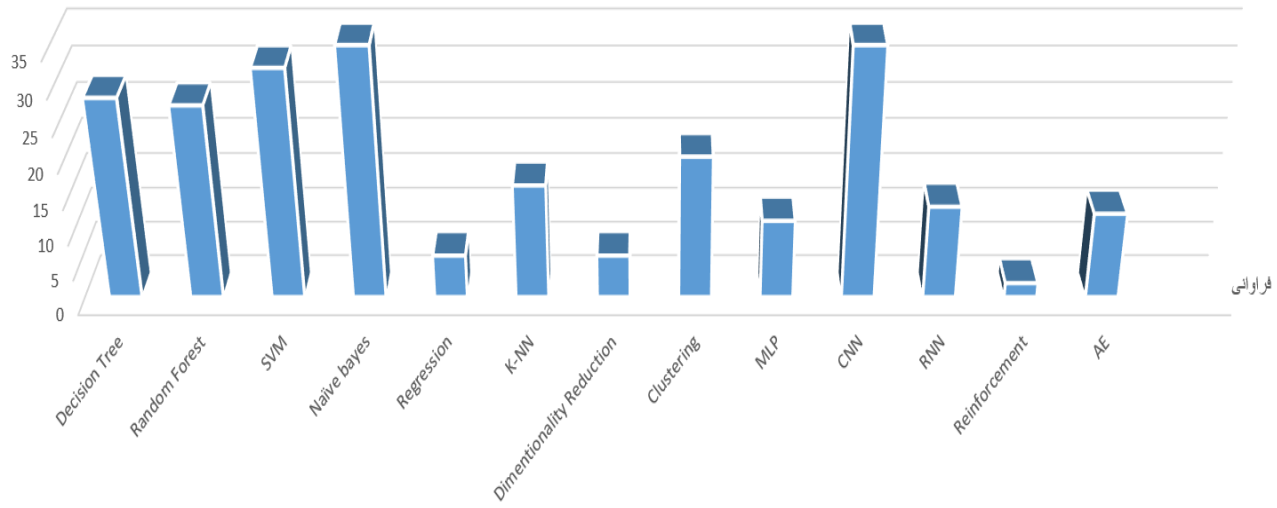
۴-۲- انتخاب مدل یادگیری

همانطور که اشاره شد، تحلیل ترافیک رمز بیشتر توسط روش‌های یادگیری ماشین و مبتنی بر ویژگی‌های آماری صورت می‌گیرد و در این مقاله هدف از تحلیل ترافیک، شناسایی برنامه از روی ویژگی‌های آماری ترافیک مربوطه در بستر شبکه است.

بر اساس مطالعات انجام گرفته [23] تا [25] ، در گام اول پروژه بررسی جامعی روی روش‌های هوش مصنوعی و یادگیری عمیق جهت انتخاب روش‌های پیشنهادی برای تحلیل ترافیک برنامه‌های کاربردی انجام گرفت. توزیع فراوانی استفاده از روش‌های مختلف یادگیری ماشین در حوزه‌های مرتبط با تحلیل ترافیک به عنوان یک خروجی آماری (مستخرج از مطالعه ۱۲۰ مقاله معتبر در حوزه تحلیل ترافیک در که پشتوانه علمی و تحلیلی آن در مرجع [20] ارائه شده است) در شکل ۲ نشان داده شده است.

جدول ۴. برنامه‌های انتخابی مجموعه داده

برچسب وب سرویس	تعداد رکوردها
Playstation	261
Steam	796
Xbox	1119
Starcraft	680
Telegram	832
WhatsApp	23816
Instagram	15661
Snapchat	94
Twitter	10628
WhatsAppCall	1013



شکل ۲. فراوانی روش‌های یادگیری ماشین و یادگیری ژرف مطالعه شده در حوزه تحلیل ترافیک [20]

بر اساس مطالعات تطبیقی و میدانی و تحلیل انجام گرفته منتج به مقاله [۲۰] و با توجه به تفسیر پذیری روش‌های یادگیری ماشین نسبت به روش‌های یادگیری عمیق، روش‌های زیر برای پیاده‌سازی مدل در این مقاله انتخاب شدند :

- روش‌های نایو بیز
- بردار پشتیبان ماشین (SVM)
- درخت تصمیم
- جنگل تصادفی

جدول ۵ پیچیدگی زمانی چهار الگوریتم پایه انتخابی و پارامترهای آنها را نشان می‌دهد، و پیچیدگی تست به ازای یک نمونه تست (تشخیص یک نمونه جریان ترافیکی) محاسبه شده است.

جدول ۵. پیچیدگی زمانی الگوریتم‌های پایه

مدل	پیچیدگی آموزش	پیچیدگی تست	پارامترها
SVM	$O(n^2)$	$O(k.d)$	K:تعدادبردارهای‌پشتیبان d: تعداد ویژگی‌ها n: تعداد نمونه‌ها
درخت تصمیم	$O(n.\log n.d)$	$O(p)$	n: تعداد نمونه‌ها p: حداکثر عمق درخت d: تعداد ویژگی‌ها
جنگل تصادفی	$O(n.\log n.d.k)$	$O(p.k)$	n: تعداد نمونه‌ها k: تعداددرختهای‌تصمیم d: تعداد ویژگی‌ها p: حداکثر عمق درخت
نایو بیز	$O(Nd)$	$O(cd)$	n: تعداد نمونه‌ها d: تعداد ویژگی‌ها c: تعداد کلاس‌ها

همانطور که در این جدول مشخص است الگوریتم درخت تصمیم دارای پیچیدگی زمانی مناسبی در مقایسه با سایر الگوریتم‌ها است. بنابراین از نظر سرعت یادگیری و پیش بینی نیز الگوریتم جنگل تصادفی (تجمیع درخت‌های تصمیم) انتخاب مناسبی به نظر می‌رسد.

با توجه به مسئله‌ی نامتوازن بودن بسیاری از مجموعه داده‌ها (از جمله داده‌های ترافیک شبکه)، دو مدل تجمیع بوت استرپ متوازن[1] (با الگوریتم پایه درخت تصمیم) و جنگل تصادفی متوازن[2] نیز به عنوان روش‌های مقاوم در برابر داده‌های نامتوازن، مورد ارزیابی قرار گرفته‌اند که هر کدام روشی را برای انتخاب داده‌های آموزشی از مجموعه داده نامتوازن دارد. پیچیدگی محاسباتی این دو الگوریتم مشابه جنگل تصادفی است.

۴-۳- پیاده‌سازی و اعتبارسنجی مدل

شکل ۳ نمودار گردش کار ترافیک در مدل پیشنهادی را نشان می‌دهد. در این مقاله واحد طبقه‌بندی ترافیک شبکه، جریان در نظر گرفته شده است به این معنی که هر جریان ترافیکی به عنوان یک ورودی برای مدل در نظر گرفته شده (شکل ۳ قسمت A) و پس از تبدیل به یک بردار ویژگی در یکی از کلاس‌ها طبقه‌بندی می‌شود (مجموعه‌ای از بسته‌های متوالی یک جریان را تشکیل می‌دهد). جهت پیاده‌سازی الگوریتم‌های یادگیری ماشین، ابتدا با جمع‌آوری ترافیک برنامه‌های کاربردی مختلف از طریق موبایل یا اینترنت، یک مجموعه داده‌ی خام (که در زمان جمع‌آوری ترافیک بانظارت

[1] Balanced Bagging Classifier

[2] Balanced Random Forest

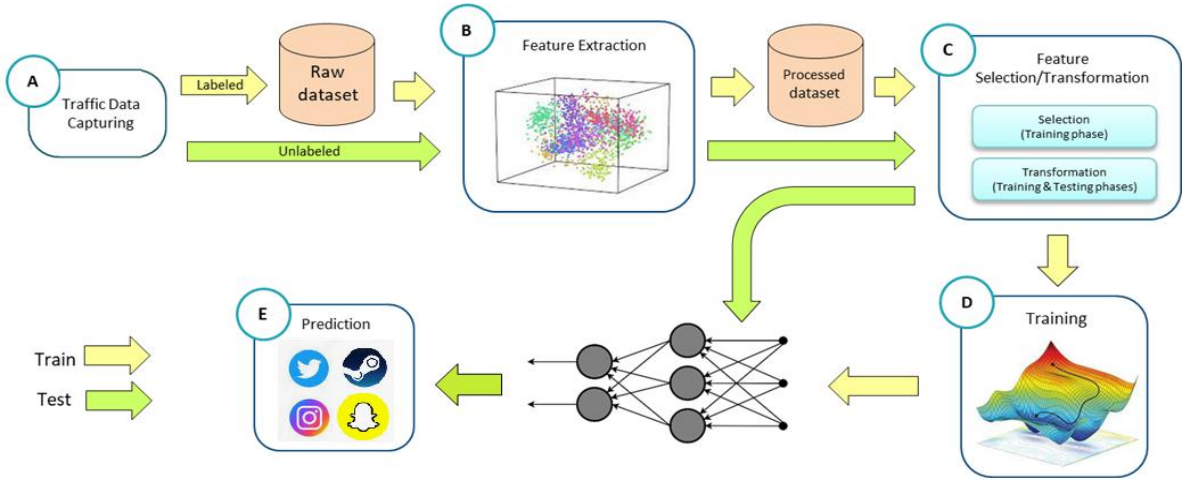
برچسب آن مشخص است) بدست می‌آید (شکل ۳، خروجی قسمت A). سپس، نرمال‌سازی و استخراج ویژگی‌های آماری از داده‌های خام ترافیک با استفاده از ابزارهای DPI منتخب شامل: NFstream و CICFlowmeter صورت می‌گیرد. در این مرحله حذف ویژگیهایی مانند آدرس IP و پورت مبدا و تبدیل ویژگیهایی غیر عددی به عددی با روش binarization (با استفاده از متد توابع کتابخانه scikit-learn مربوط به پکیج pandas) صورت می‌گیرد و مقادیر عددی در بازه صفر و ۱ نرمال می‌شوند (شکل ۳ قسمت B). در مرحله بعد، از طریق روش‌های تبدیل فضای ویژگی و نیز روش‌های کاهش ویژگی متناسب با هر الگوریتم یادگیری ماشین و با استفاده از رتبه‌بندی ویژگی‌ها با روش‌های مختلف نظیر آنتروپی متقابل به انتخاب ویژگی‌های مناسب برای طبقه‌بندی داده‌های ترافیک پرداخته می‌شود (شکل ۳ قسمت C و درنهایت الگوریتم‌های یادگیری ماشین بر روی داده‌های آماری به‌دست‌آمده اعمال می‌شوند. خروجی این مرحله یک مدل یادگیری ماشین آموزش یافته است (شکل ۳ قسمت D). در نهایت از مدل آموزش دیده شده برای طبقه‌بندی داده‌های ترافیک شبکه در مرحله تست استفاده می‌شود (شکل ۳ قسمت E). در این نمودار دو خط انتقال آموزش و تست به‌طور مجزا با رنگ‌های زرد و سبز نشان داده شده‌اند.

تکنیک‌های متفاوتی برای بخش‌بندی مجموعه دادگان وجود دارد که در این مقاله برای ارزیابی توانایی الگوریتم‌های یادگیری ماشین از اعتبارسنجی معروف 10-fold cross-validation استفاده شده است. به این معنی که مجموعه داده به ۱۰ قسمت تقسیم شده و هر بار ۹ قسمت برای آموزش مدل یادگیری ماشین و یک قسمت برای تست آن استفاده می‌شود. در نهایت میانگین کارایی بدست آمده در این ۱۰ تکرار به عنوان کارایی الگوریتم ثبت می‌شود. معیارهای مقایسه صحت[1]، حساسیت[2]، دقت[3] و ضریب متئوس[4] که تعریف و نحوه محاسبه آنها در ادامه آمده، برای سنجش کارایی روش‌های یادگیری ماشین انتخابی مورد استفاده قرار گرفته‌اند.

$$recall = \frac{TP}{TP+FN} \qquad precision = \frac{TP}{TP+FP} \quad (1)$$

برای درک بهتر پارامترهای TP، TN، FP، و FN مفهوم آنها برای شناسایی جریان ترافیک توئیتر در پاورقی آورده شده است.[5] Recall یا حساسیت نشان می‌دهد دسته‌بند به چه اندازه در تشخیص تمام نمونه‌های متعلق به یک کلاس موفق بوده است. این معیار در مواردی که احتمال false negatives بالا باشد، معیار مناسبی محسوب می‌شود.

Precision یا صحت بیان کننده این است که وقتی مدل نتیجه را مثبت پیش‌بینی می‌کند، این نتیجه تا چه اندازه درست است؟ زمانی که ارزش false positives بالا باشد، معیار صحت، معیار مناسبی خواهد بود. با توجه به توضیحات داده شده و اهمیت False Positive در این پروژه، این معیار می‌تواند بیشتر مد نظر قرار گیرد.

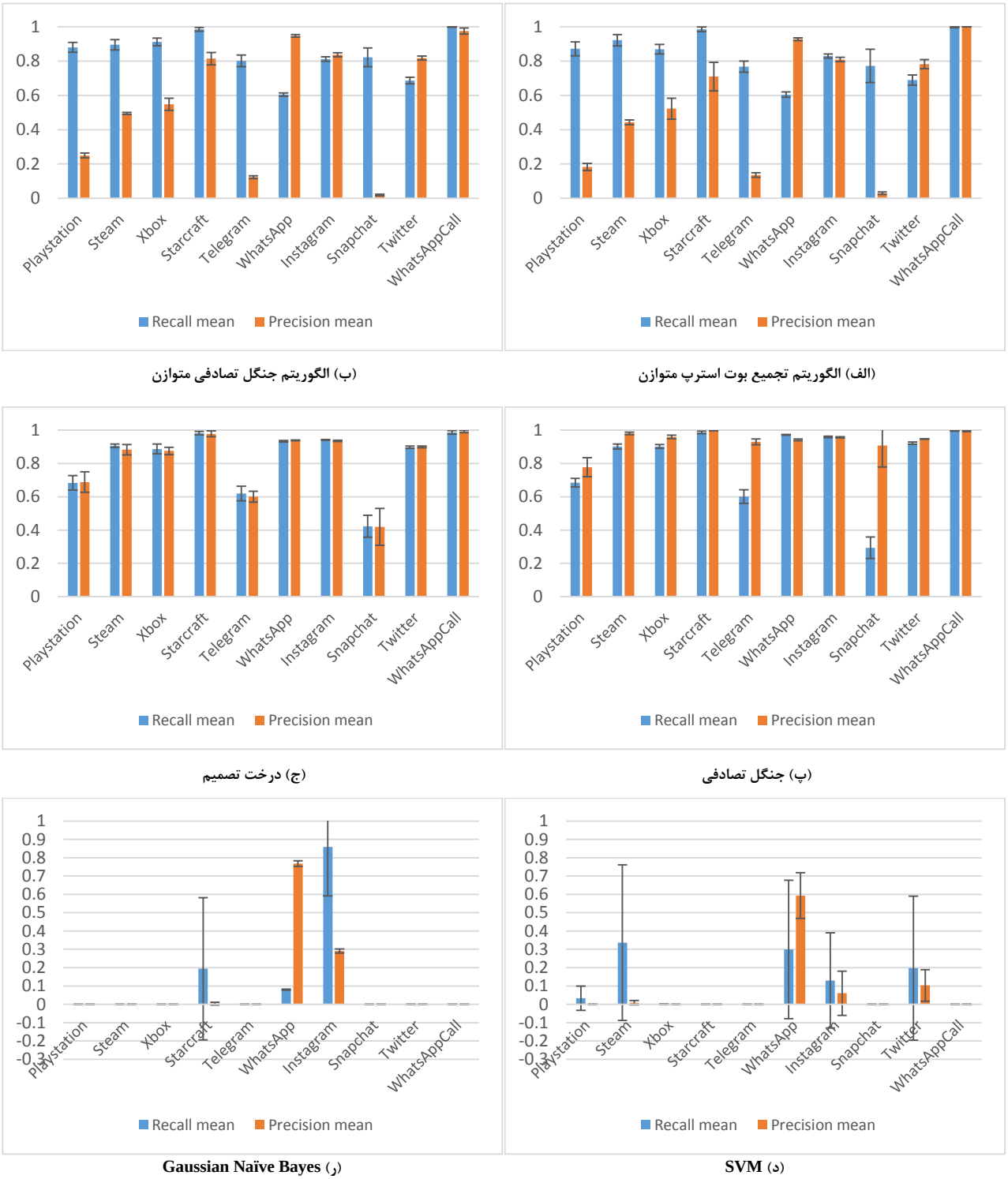


شکل ۳. نمودار گردش کار در مرحله آموزش و تست

[1] Precision
[2] Recall
[3] Accuracy
[4] Matthews correlation coefficient (MCC)
[5] TP: زمانی که ترافیک ورودی توئیتر بوده و سامانه آن را توئیتر تشخیص دهد (True Positive). یعنی سامانه این جریان را برای توئیتر، بدرستی (True) مثبت (Positive) ارزیابی کرده است.

FP: زمانی که ترافیک ورودی توئیتر نیست، اما سامانه آن را توئیتر تشخیص دهد (False Positive). یعنی سامانه این جریان را برای توئیتر، به غلط (False)، مثبت (Positive) ارزیابی کرده است.
TN: زمانی که ترافیک ورودی توئیتر نیست، و سامانه هم آن را توئیتر تشخیص نمی‌دهد (True Negative). یعنی سامانه جریان را، بدرستی (True) منفی (Negative) ارزیابی کرده است.
FN: زمانی که ترافیک ورودی توئیتر بوده، اما سامانه آن را توئیتر تشخیص ندهد. (False Negative). یعنی سامانه ترافیک توئیتر را، به غلط (False)، منفی (Negative) ارزیابی کرده است.

میزان کارایی روش‌های یادگیری مورد مطالعه به تفکیک هر یک از کلاس‌ها در قالب precision و recall در شکل ۴ نشان داده شده است. نوار مشکی رنگ در شکل‌ها پراکندگی دقت بدست آمده در تکرارهای مختلف 10-fold cross-validation را نشان می‌دهد.



(الف) الگوریتم تجمیع بوت استرپ متوازن

(ب) الگوریتم جنگل تصادفی متوازن

(پ) جنگل تصادفی

(ج) درخت تصمیم

SVM (د)

Gaussian Naïve Bayes (ر)

شکل ۴.

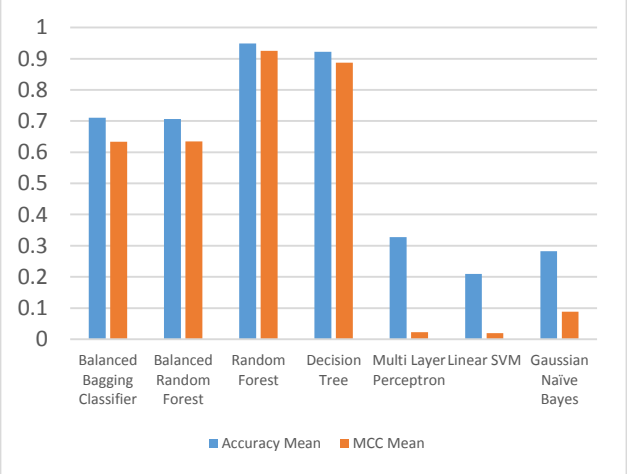
شکل ۵. کارایی الگوریتم‌های مورد مطالعه بر اساس معیارهای Recall و Precision به تفکیک هر یک از کلاس‌ها (الف – ر) محور عمودی مقدار معیار ارزیابی محاسبه شده را نشان می‌دهد.

با توجه به عدم توازن داده‌ها در کلاسهای مختلف پایگاه داده Kaggle141، برای مقایسه عملکرد الگوریتمهای مختلف در تشخیص نوع برنامه (شکل ۵) از معیار دقت (Accuracy) و ضریب همبستگی متئوس (MCC) استفاده کرده‌ایم.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$MCC = \frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (3)$$

Accuracy یا دقت نشان می‌دهد که مدل آموزش یافته تا چه اندازه خروجی را درست پیش‌بینی می‌کند و مقدار آن بین صفر و ۱ است. MCC به عنوان یک معیار ارزیابی متوازن در کلاس‌هایی با اندازه متفاوت مورد استفاده قرار می‌گیرد و نشان‌دهنده کیفیت کلاس‌بندی برای یک مجموعه باینری است. مقدار ۱+ یک پیش‌بینی کامل و ۱- عدم تطابق کامل بین پیش‌بینی و مشاهده را نشان می‌دهد. با استفاده از روش‌های Micro-averaging و Macro-averaging امکان استفاده از این معیار برای مسائل چند کلاسه (بیش از دو کلاس) وجود دارد. اما در این مقاله مسأله به صورت دو کلاسه مطرح شده و یک مدل جنگل تصادفی برای هر برنامه کاربردی آموزش داده می‌شود و در مرحله تشخیص، مبتنی بر رأی‌گیری (بر اساس تعداد پکتهای متعلق به هر برنامه کاربردی در یک جریان ترافیکی) نوع جریان ترافیک تشخیص داده می‌شود.



شکل ۶. مقایسه عملکرد الگوریتم‌های مختلف

بر اساس نتایج پیاده‌سازی همانطور که در شکل ۵ نشان داده شده است، الگوریتم‌های تجمیع بوت استرپ متوازن، جنگل تصادفی متوازن، درخت تصمیم و جنگل تصادفی برتری قابل ملاحظه‌ای نسبت به سایر روش‌ها داشتند، و روش جنگل تصادفی با دقت ٪۹۵ بهترین عملکرد را در تشخیص ترافیک برنامه‌ها ارائه داده است.

زمان مورد نیاز برای آموزش مدل بسته به توان سخت‌افزاری و پردازشی متفاوت است و بین ۱۵ دقیقه تا چند ساعت برای الگوریتمهای مختلف، متفاوت است. مرحله خواندن فایل‌های pcap و پیش پردازش آنها روی گوگل کولب ۲۶,۴۹ ثانیه زمان می‌برد.

در جدول ۶ میانگین زمانی سرعت مدل در مواجهه با انواع ترافیک مورد آزمون قرار گرفته و میانگین زمان پاسخ مدل به هر جریان (flow) ورودی گزارش شده است.

جدول ۶. نتایج تست سرعت بر روی گوگل کولب (بر حسب ثانیه)

نوع فرایند	زمان اجرا
پیش‌بینی برچسب	0.299185
محاسبه معیارهای ارزشیابی	0.030947

همانطور که در جدول بالا مشاهده می‌شود، متوسط زمان لازم برای شناسایی برچسب برنامه کاربردی مربوط به یک جریان کمتر از ۰/۳ ثانیه می‌باشد که زمان قابل قبولی برای کاربردهای بلادرنگ است.

با استفاده از اکثر مدل‌های یادگیری ماشین، به ازای هر ویژگی یک مقدار به‌عنوان درجه‌ی اهمیت آن ویژگی قابل محاسبه است. که علاوه بر قابلیت استفاده در کاهش ویژگی، می‌توان از آن در راستای اولویت‌بندی ویژگیها و استخراج قوانین تشخیص ترافیک مرتبط با هر برنامه استفاده نمود. این قوانین می‌توانند در سیستم‌های تشخیص ترافیک مبتنی بر قانون جهت بروزرسانی قوانین مورد استفاده قرار گیرند. جدول ۷ لیست ویژگی‌های ترافیکی به‌ترتیب درجه‌ی اهمیت آنها در روش جنگل تصادفی را نشان می‌دهد.

از آنجائیکه مدلهای تحلیل ترافیک عموما با اهدافی خاص و روی مجموعه داده‌های محلی متفاوتی انجام می‌گیرند، نتایج حاصل از آنها قابلیت مقایسه با یکدیگر را ندارد، با اینحال در جدول ۸ برخی از نتایج ارائه شده در مقالات مختلف، در کنار نتایج بدست آمده در مقاله نشان داده شده است.

جدول ۷. لیست ویژگی‌ها و ترتیب درجه اهمیت آنها در جنگل تصادفی

Rank	Random Forest	Rank	Random Forest	Rank	Random Forest	Rank	Random Forest
1	src2dst_first_seen_ms	44	dst2src_rst_packets	87	application_category_name_Cloud	130	application_name_DNS.GoogleDocs
2	bidirectional_last_seen_ms	45	src2dst_stddev_ps	88	application_name_Dropbox	131	application_name_NTP.Apple
3	src2dst_last_seen_ms	46	src2dst_syn_packets	89	application_name_QUIC.Facebook	132	application_name_Apple
4	bidirectional_first_seen_ms	47	dst2src_ack_packets	90	application_name_QUIC.Google	133	application_category_name_ConnCheck
5	application category name SocialNetwork	48	application_name_TLS.WhatsApp	91	application_name_NTP.Google	134	application_name_ICMP.Apple
6	dst2src_last_seen_ms	49	src2dst_mean_piat_ms	92	application_name_WhatsAppFiles	135	application_name_RX.Facebook
7	dst2src_first_seen_ms	50	dst2src_stddev_piat_ms	93	application_name_Instagram	136	application_name_RTP.Facebook
8	dst_ip	51	src2dst_packets	94	application_name_DNS.GoogleServices	137	application_name_DNS.Microsoft
9	bidirectional_min_ps	52	bidirectional_stddev_piat_ms	95	application_name_DNS.Apple	138	application_name_TLS.GoogleDocs
10	application_name_TLS	53	dst2src_packets	96	application_name_NetBIOS	139	application_name_DNS.QQ
11	src2dst_min_ps	54	dst2src_duration_ms	97	application_name_RTP	140	application_name_Z39.50.Apple
12	src2dst_max_ps	55	application_category_name_Network	98	application_name_TLS.Amazon	141	application_name_QUIC.GoogleDrive
13	dst_port	56	bidirectional_rst_packets	99	application_name_ICMP	142	application_name_HTTP.GoogleServices
14	src2dst_mean_ps	57	src2dst_stddev_piat_ms	100	application_name_MDNS	143	application_name_DNS.GoogleDrive
15	dst2src_min_ps	58	bidirectional_psh_packets	101	application category name SoftwareUpdate	144	application_name_TLS.QQ
16	bidirectional_mean_ps	59	src2dst_psh_packets	102	application_category_name_Media	145	application_name_DNS.AppleiTunes
17	src2dst_bytes	60	application_category_name_Download	103	application name DNS WhatsAppFiles	146	application_name_TLS.AppleSiri
18	bidirectional_max_ps	61	dst2src_mean_piat_ms	104	ip_version	147	application_name_DNS.Amazon
19	application_name_TLS.Google	62	application_category_name_VoIP	105	application_name_SSDP	148	application_name_TLS.Microsoft365
20	bidirectional_stddev_ps	63	protocol	106	application name QUIC WhatsAppFiles	149	application_category_name_VirtAssistant
21	bidirectional_bytes	64	dst2src_psh_packets	107	application_name_TLS.Apple	150	application_name_DNS.Activision
22	bidirectional_duration_ms	65	application_name_QUIC.Instagram	108	application category name Collaborative	151	application_name_DNS.Ookla
23	bidirectional_max_piat_ms	66	application_is_guessed	109	application_name_Facebook	152	application_name_HTTP_Proxy.Facebook
24	application_category_name_Web	67	src2dst_min_piat_ms	110	application_name_DHCPV6	153	src2dst_urg_packets
25	bidirectional_mean_piat_ms	68	application_name_TLS.GoogleServices	111	application_name_STUN	154	application_name_TLS.GoogleDrive
26	dst2src_mean_ps	69	application_name_DNS.Messenger	112	application_category_name_Streaming	155	bidirectional_urg_packets
27	dst2src_max_ps	70	bidirectional_fin_packets	113	application_name_LLMNR	156	vlan_id
28	application_name_TLS.Instagram	71	application_name_DNS.Facebook	114	application_name_QUIC	157	tunnel_id
29	dst2src_bytes	72	src2dst_rst_packets	115	application_name_TLS.Activision	158	src2dst_cwr_packets
30	bidirectional_ack_packets	73	dst2src_min_piat_ms	116	application_name_STUN.Apple	159	src2dst_ece_packets
31	application_category_name_Chat	74	src2dst_fin_packets	117	application_name_TLS.Cloudflare	160	dst2src_cwr_packets
32	application_name_DNS.Instagram	75	application_name_TLS.Messenger	118	application name QUIC GoogleServices	161	bidirectional_ece_packets
33	src2dst_ack_packets	76	application name TLS WhatsAppFiles	119	application_name_QUIC.WhatsApp	162	application_name_SNMP
34	application_name_Unknown	77	application_name_DNS	120	application_name_STUN.Facebook	163	application_category_name_RPC
35	application_name_TLS.Facebook	78	application_name_DNS.WhatsApp	121	application_name_DHCP	164	dst2src_ece_packets
36	application category name Unspecified	79	application_name_STUN.Messenger	122	application_category_name_Game	165	dst2src_urg_packets
37	bidirectional_syn_packets	80	dst2src_syn_packets	123	application_name_ApplePush.Apple	166	application_name_DNS.Microsoft365
38	src2dst_max_piat_ms	81	application name STUN WhatsAppCall	124	application_name_IGMP	167	application_name_HTTP
39	bidirectional_min_piat_ms	82	dst2src_fin_packets	125	application name QUIC GoogleDocs	168	bidirectional_cwr_packets
40	src2dst_duration_ms	83	application category name System	126	application_name_ICMPV6		
41	dst2src_max_piat_ms	84	application_name_WhatsApp	127	application_name_HTTP.Google		
42	dst2src_stddev_ps	85	application_name_Google	128	application_name_TLS.AppleiCloud		
43	bidirectional_packets	86	application_name_DNS.Google	129	application_name_TLS.AppleiTunes		

جدول ۸. دقت گزارش شده برای جنگل تصادفی و درخت تصمیم

مرجع	درخت‌تصمیم	جنگل‌تصادفی	مجموعه داده
[26]	%88	٪90	Self-Collected
[27]	%98	٪99	Self-Collected
[28]	%95	%94	Kaggle
[29]	%81	%82	UNB Dataset NIMS Dataset
[30]	%96	%97	Self-Collected
روش پیشنهادی	%92	%95	Kaggle141
روش پیشنهادی	-	%97	ترافیک محلی

در ادامه روش پیشنهادی روی ترافیک واقعی نیز مورد ارزیابی قرار گرفت. به این صورت که پس از نرمال‌سازی داده‌های واقعی (ترافیک سه برنامه کاربردی در محیط آزمایشگاهی) و آموزش مدل جنگل تصادفی، عملکرد آن جهت تشخیص ترافیک آنلاین برنامه‌های هدف مورد ارزیابی قرار گرفت. نتایج اولیه نشان دهنده عملکرد قابل قبول روش پیشنهادی (دقت متوسط ٪۹۷) در تشخیص ترافیک واقعی برنامه‌های کاربردی است.

۵- جمع‌بندی و پیشنهادات

شناسایی ترافیک با اهداف متفاوتی نظیر تامین کارایی شبکه، تامین کیفیت سرویس، تشخیص نفوذگران، مدیریت منابع و گاهی با هدف تنظیم کنترل دسترسی ترافیک انجام می‌شود.

روش‌های مختلف مبتنی بر پورت، مبتنی بر دنباله بسته، مبتنی بر گراف، مبتنی بر تحلیل‌های آماری و مبتنی بر یادگیری ماشین وجود دارد. روش یادگیری ماشین، رویکرد مناسبی جهت استفاده از ویژگی‌ها در شناسایی ترافیک رمز با اهداف مختلف است. در این مقاله پس از بررسی چالش‌های تحلیل ترافیک و رویکردهای آن، مدل هوشمندی برای تحلیل و طبقه‌بندی ترافیک رمز مبتنی بر ویژگی‌های آماری و روش‌های یادگیری ماشین ارائه شده و توسط روش‌های یادگیری منتخب و متناسب با رویکرد تشخیص ترافیک برنامه‌ها روی وب سرویس‌های مجموعه داده Kaggle مورد ارزیابی قرار گرفت.

دستاوردها و پیشنهادات زیر جهت ادامه کار توصیه می‌شود:

- ایجاد مجموعه داده ترافیکی با استفاده از روش‌ها و ابزارهای تولید و جمع‌آوری ترافیک جهت تامین QOS و سایر کاربردهای تحلیل ترافیک یکی از ملزومات اصلی جهت بکارگیری روش‌های هوشمند تحلیل ترافیک در صنعت و کاربردهای واقعی است.
- با توجه به اینکه امضای نرم‌افزارهای کاربردی مدام در حال تغییر است، نیاز به جمع‌آوری پایگاه داده بروز و آموزش به صورت مستمر برای الگوریتم‌های یادگیری ماشین یک نیاز اساسی است که باید به آن پرداخته شود.
- انتخاب هوشمند ویژگیهای ترافیکی مبتنی بر هوش مصنوعی و یادگیری عمیق می‌تواند بسته به کاربرد در راستای حفظ حریم خصوصی کاربران، کاهش ترافیک ورودی و افزایش سرعت مدل بکارگرفته شود.
- یکی از مهمترین نیازمندیها در تحلیل ترافیک مبتنی بر روش‌های هوشمند، بروزرسانی داده‌های ترافیکی و آموزش مستمر مدل است که مستلزم پشتیبانی دائمی مدلهای هوشمند است.
- با استفاده از روش‌های مبتنی بر یادگیری ماشین، می‌توان ویژگیهای ترافیکی مؤثر و امضاهای جدید نرم‌افزارها، پروتکل‌ها و تهدیدات جدید را استخراج کرد که یکی از الزامات در سیستم‌های مبتنی بر قانون جهت پاسخگویی مستمر مدل است و می‌تواند وابستگی به نرم‌افزارهای خارجی این حوزه را کاهش دهد.
- همچنین می‌توان طرح معماری هوشمند پیشنهادی را در کنار مدلهای متداول مبتنی بر قانون، جهت تشخیص هوشمند رفتار غیر نرمال و شناسایی تهدیدات zero-day بکار گرفت.

در حال حاضر پروسه ایجاد یک مجموعه داده محلی از ترافیک برنامه‌های کاربردی مختلف در دست انجام است، که امکان تحلیل ترافیک و ارزیابی عملکرد آن به صورت بومی و آنلاین را فراهم ساخته و قابلیت استفاده در کاربردهای واقعی را فراهم می‌سازد.

مراجع

[1] D. Gibert, C. Mateu, and J. Planes, “The rise of machine learning for detection and classification of malware: Research developments, trends and challenges,” J. Netw. Comput. Appl., vol. 153, p. 102526, 2020.

[2] J. Pluskal, O. Lichtner, and O. Rysavy, “Traffic Classification and Application Identification in Network Forensics,” 2018, pp. 161–181.

[3] J. Zhao, X. Jing, Z. Yan, W. Pedrycz, Network traffic classification for data fusion: A survey, Information Fusion 72 (2021) 22–47. doi:https://doi.org/10.1016/j.inffus.2021. 02.009

[4] Zihao Wang, Kar-Wai Fok, Vrizlynn L. L. Thing, "Machine Learning for Encrypted Malicious Traffic Detection: Approaches, Datasets and Comparative Study", Computers & Security, Volume 113, 2022.

[5] M. Conti, Q. Q. Li, A. Maragno, and R. Spolaor, “The dark side(-Channel) of Mobile Devices: A survey on network traffic analysis,” IEEE Commun. Surv. Tutorials, 2018.

[6] Z. Liu, R. Wang, N. Japkowicz, Y. Cai, D. Tang, and X. Cai, “Mobile app traffic flow feature extraction and selection for

2nd IEEE International Conference on Computer and Communications, ICCC 2016 - Proceedings, 2017.

[19] O. Salman, I. H. Elhajj, A. Kayssi, A. Chehab, A review on machine learning–based approaches for internet traffic classification, Annals of Telecommunications 75 (11) (2020) 673–710.

[20] Mohammad Pooya Malek, Shaghayegh Naderi, Hossein Gharaee Garakani, "A Review on Internet Traffic Classification Based on Artificial Intelligence Techniques", International Journal of Information & Communication Technology Research (IJICTR), Vol. 14, N. 2, pp. 1-13, 2022.

[21] Madushi H. Pathmaperuma,Yogachandran Rahulamathavan,Safak Dogan and Ahmet M. Kondoz, "Deep Learning for Encrypted Traffic Classification and Unknown Data Detection", Sensors 2022, 22(19), 7643; https://doi.org/10.3390/s22197643.

[22] "Labeled Network Traffic flows - 141 Applications", Kaggle.com. [Online]. Available: https://www.kaggle.com/jsrojas/ labeled-network-traffic-flows-114-applications. [Accessed: 07- Jan- 2022].

[23] S. Dong, Multi class SVM algorithm with active learning for network traffic classification, Expert Systems with Applications 176 (2021) 114885. doi:https://doi.org/10.1016/j.eswa.2021.114885.

[24] A. A. Afuwape, Y. Xu, J. H. Anajemba, G.Srivastava, Performance evaluation of secured network traffic classification using a machine learning approach, Computer Standards & Interfaces 78 (2021) 103545. doi:https://doi.org/10.1016/j.csi.2021.103545

[25] Bei Lu, Nurbol Luktarhan, Chao Ding and Wenhui Zhang, "ICLSTM: Encrypted Traffic Service Identification Based on Inception-LSTM Neural Network", Symmetry 2021, 13(6), 1080; https://doi.org/10.3390 /sym13061080.

[26] Fathi-Kazerooni, Sina, Yagiz Kaymak, and Roberto Rojas-Cessa. "Identification of user application by an external eavesdropper using machine learning analysis on network traffic." In 2019 IEEE International Conference on Communications Workshops (ICC Works.

[27] Perera, Menuka, Kandaraj Piamrat, and Salima Hamma. "Network Traffic Classification using Machine Learning for Software Defined Networks." In Journées non thématiques GDR-RSD 2020. 2020..

[28] Peng, Lizhi, Bo Yang, and Yuehui Chen. "Effective packet number for early stage internet traffic identification." Neurocomputing 156 (2015): 252-267..

[29] Dong, Yu-ning, Jia-jie Zhao, and Jiong Jin. "Novel feature selection and classification of Internet video traffic based on a hierarchical scheme." Computer Networks 119 (2017): 102-111..

[30] Khatouni, Ali Safari, and Nur Zincir Heywood. "How much training data is enough to move a ML-based classifier to a different network?." Procedia Computer Science 155 (2019): 378-385.

improving classification robustness," J. Netw. Comput. Appl., 2019.

[7] Eva Papadogiannaki, Sotiris IoannidisSotiris Ioannidis, "A Survey on Encrypted Network Traffic Analysis Applications, Techniques, and Countermeasures", ACM Computing Surveys 54(6):1-35, July 2021.

[8] D. S. Mohamad Amar, "A Survey on Tor Encrypted Traffic Monitoring", International Journal of Advanced Computer Science and Applications 9(8), 2018.

[9] A. Bhatiaa, A. A. Bahugunaa, K. Tiwaria, K. Haribabua, and D. Vishwakarmab, "A Survey on Analyzing Encrypted Network Traffic of Mobile Devices," Jun. 2020.

[10] Adi Lichy, Ofek Bader, Ran Dubin, Amit Dvir, Chen Hajaj, "When a RF Beats a CNN and GRU, Together - A Comparison of Deep Learning and Classical Machine Learning Approaches for Encrypted Malware Traffic Classification", arXiv:2206.08004v1 [cs.CR] 16 Jun 2022.

[11] T. van Ede, R. Bortolameotti, A. Continella, J. Ren, D. J. Dubois, M. Lindorfer, D. Choffnes, M. van Steen, and A. Peter, "Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic," in Network and Distributed System Security Symposium (NDSS), vol. 27, 2020.

[12] R.-H. Hwang, M.-C. Peng, C.-W. Huang, P.-C. Lin, and V.-L. Nguyen, "An unsupervised deep learning model for early network traffic anomaly detection," IEEE Access, vol. 8, pp. 30 387–30 399, 2020.

[13] A. Azab, M. Khasawneh, S. Alrabaee, K.-K. Raymond Choo, M. Sarsour, "Network traffic classification: Techniques, datasets, and challenges", Digital Communications and Networks (2022), doi: https://doi.org/10.1016/j.dcan.2022.09.009.

[14] P. Khandait, N. Hubballi, B. Mazumdar, Efficient keyword matching for deep packet inspection based network traffic classification, in: 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), IEEE, 2020, pp. 567–570.

[15] S. Rezaei and X. Liu, "Multitask Learning for Network Traffic Classification," in Proceedings - International Conference on Computer Communications and Networks, ICCCN, 2020.

[16] M. Perera Jayasuriya Kuranage, K. Piamrat, and S. Hamma, "Network Traffic Classification Using Machine Learning for Software Defined Networks," 2020, pp. 28–39.

[17] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, "Deep packet: a novel approach for encrypted traffic classification using deep learning," Soft Comput., vol. 24, no. 3, pp. 1999–2012, 2020.

[18] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn, and F. Abdessamia, "Network Traffic Classification techniques and comparative analysis using Machine Learning algorithms," in 2016