Scientific Journal of Electronic & Cyber Defense
Vol. 11, No. 3, Autumn 2023, Serial No. 43
ISSN: 2322-4347 , E-ISSN: 2980-8979

DDoS Attack Detection System Using Ensemble Method Classification and Active Learning Approach

Mohammad Rahmani Manesh[1], **Masoud Khorram**[2]

((Received: 2023/04/05, Revised: 2023/07/17, Accepted: 2023/07/22, Published: 2023/07/06))

DOR: https://dorl.net/dor/ 20.1001.1.23224347.1402.11.3.10.5

Abstract

Distributed Denial of Service (DDoS) attack is the widespread sending of valid or invalid packets to a server on the Internet, occupying its bandwidth and preventing execute legitimate requests of other users. The best approach to secure the network from such attacks is to exploit security controls such as intrusion detection and prevention systems. Cyber security researchers have significantly focused on identifying and counteracting this attack and have increased the accuracy and performance of security systems by providing various artificial intelligence solutions. The purpose of this paper is also to provide a solution for detecting DDoS attack, where, decision tree, multi-layer perceptron and random forest classifiers have been utilized in an ensemble method to mitigate the over-fitting problem. Also, two approache, i.e., batch learning and active learning have been implemented and evaluated in the classification phase of the proposed method. The evaluation results show that the mean value of accuracy in DDoS attack detection is 99.81%.

Keywords: DDoS attack detection, Network traffic classification, Network security, Ensemble method, Active learning, Flow-level features, CICIDS2017 dataset

نشریه علمی "پدافند الکترونیکی و سایبری"
سال یازدهم، شماره ۳، پاییز ۱۴۰۲، ص ۱۱۸-۱۰۱

علمی – پژوهشی

سیستم تشخیص حملات DDOS با استفاده از روش دسته‌بندی گروهی و رویکرد یادگیری فعّال

مسعود خرّم[۱]، محمد رحمانی منش[۲]

کارشناس ارشد هوش مصنوعی دکتر محمد رحمانی منش استادیار دانشگاه سراسری سمنان، سمنان، ایران.

(دریافت: ۱۴۰۲/۰۱/۱۶، بازنگری: ۱۴۰۲/۰۴/۲۶،پذیرش: ۱۴۰۲/۰۴/۳۱، انتشار: ۱۴۰۲/۰۷/۰۶)

DOR: https://dorl.net/dor/ 20.1001.1.23224347.1402.11.3.10.5

چکیده

حمله منع سرویس توزیع شده (**DDoS**) ارسال گسترده‌ای از بسته‌های معتبر یا نامعتبر به یک سرویس‌دهنده[۱] در اینترنت است که از این طریق پهنای باند آن را اشغال کرده و مانع از اجرای درخواست‌های قانونی سایر کاربران می‌شود. بهترین رویکرد برای امن‌سازی شبکه از چنین حملاتی، داشتن کنترل‌های امنیتی از قبیل سامانه‌های تشخیص و پیشگیری از نفوذ و شناسایی حملات با دقت بالا است. محققان امنیت سایبری به طور قابل توجهی بر روی شناسایی و مقابله با این حمله تمرکز کرده و با ارائه راهکارهای مختلف هوش مصنوعی، دقت و عملکرد سامانه‌های امنیتی را افزایش داده‌اند. هدف از این مقاله ارائه راهکاری برای تشخیص حمله **DDoS** است. در روش پیشنهادی از الگوریتم‌های درخت تصمیم، پرسپترون چندلایه و جنگل تصادفی به روش گروهی برای افزایش اطمینان از عدم ایجاد مشکل برازش بیش‌ازحد استفاده شده است. همچنین دو رویکرد یادگیری دسته‌ای و یادگیری فعّال در بخش دسته‌بندی طرح پیشنهادی، پیاده‌سازی و ارزیابی شده است. نتایج ارزیابی نشان می‌دهد دقت معماری پیشنهادی جهت شناسایی حمله **DDoS**، ۹۹/۸۱ درصد شده است.

کلیدواژه‌ها: شناسایی حمله DDoS، دسته‌بندی ترافیک شبکه، امنیت شبکه، روش گروهی، یادگیری فعّال، ویژگی‌های سطح جریان، مجموعه داده CICIDS2017.

[۱] Server

۱- مقدمه

مشاغل و جامعه مدرن به دلیل نیازهای ارتباطی و اطلاعاتی خود به خدمات اینترنتی متکی هستند. اطمینان از در دسترس بودن این خدمات به دلیل افزایش حجم ترافیک اینترنت و استانداردهای ارتباطی مختلفی که از آن پشتیبانی می‌کند یک کار چالش‌برانگیز است [۱،۲]. حملات از کار اندازی سرویس (DoS[۲]) به شبکه و سرویس‌دهنده وب یکی از انواع رایج تهدیدات سایبری است. حمله DoS، تلاش برای خارج کردن ماشین و منابع شبکه از دسترس کاربران مجاز، از طریق ارسال انبوهی از بسته‌های معتبر یا نامعتبر و اشغال پهنای باند شبکه یا منابع فیزیکی و نرم‌افزاری سرویس‌دهنده‌ها است. این حمله از طریق یک سیستم آلوده به بدافزار صورت می‌گیرد، درحالی‌که در حمله DoS توزیع‌شده (DDoS[۳])، از طریق مجموعه‌ای از کامپیوترهای آلوده که در شبکه‌ای از روبات‌ها به نام باتنت[۴] سازمان‌دهی شده‌اند، به قربانی حمله می‌شود.

در حقیقت، حملات سیل آسا[۵] گزارش‌شده پس از سال ۱۹۹۹، بیشتر به‌صورت DDoS بوده است [۳]. حملات DDoS از مکان‌های مختلف جغرافیایی با هدف مشترک برای از کار انداختن سرویس انجام می‌شود. اهداف مشترک مهاجمین DDoS / DoS برای انجام اخاذی، باج‌گیری، انتقام‌گیری، رقابت تجاری، مسائل سیاسی، منافع اقتصادی و بعضی اوقات سرگرم‌کننده است. یکی از دلایل اصلی فراوانی بالای آن، تعداد راه‌های ایجاد و راه‌اندازی این حمله است [۴].

انجام حمله DDoS در هر لایه از مدل OSI امکان‌پذیر است. ازاین‌رو، مهاجم روش‌های مختلفی را برای انکار موفقیت‌آمیز سرویس بر روی یک قربانی هدفمند پیدا می‌کند. اگرچه این حمله سال‌هاست که وجود دارد، اما فن‌های مورداستفاده برای شروع موفقیت‌آمیز حمله به قربانیان در حال تغییر بوده‌اند. در شکل (۱) حملات مختلف DDoS که در لایه‌های مختلف شبکه انجام می‌شود، نشان داده شده است.

امروزه نیاز به امنیت سایبری برای ایجاد حس اعتماد کاربران به کسب‌وکارهای برخط[۶] به موضوعی مهم تبدیل شده است. وجود دسترسی[۷] به سامانه‌های شبکه به‌عنوان یکی از ضلع‌های مثلث امنیت است که توسط حملات DDoS تهدید می‌شود. جامعه پژوهش به‌طور قابل توجهی در ارائه راه‌حل‌های خنثی‌سازی حملات DDoS که شامل تجزیه‌وتحلیل سطح بسته، تجزیه‌وتحلیل سطح جریان، آنالیز رفتاری[۸]، کاوش ترافیک[۹] و بازرسی بسته‌ها به‌صورت عمیق است، نقش داشته است. به‌طور مشابه با توسعه مدل‌های یادگیری ماشینی و اجرای موفقیت‌آمیز آن‌ها برای رفع مشکلات مهم حوزه‌های مختلف، محافظت از شبکه در برابر حملات DDoS از طریق یادگیری ماشین و رویکردهای مبتنی بر هوش مصنوعی نیز مورد توجه محققان قرارگرفته است [۵]. نیاز به یادگیری ماشین در این زمینه به این واقعیت منجر می‌شود که حملات DDoS اکنون پیچیده‌تر و فریبنده‌تر از گذشته است.

این مقاله الگوریتم‌های درخت تصمیم[۱۰]، پرسپترون چندلایه[۱۱] و جنگل تصادفی[۱۲] را به روشی گروهی و با دو رویکرد یادگیری دسته‌ای و یادگیری فعّال ترکیب می‌کند و بدین ترتیب دقت سیستم تشخیص حملات DDoS در ترافیک شبکه را بالا می‌برد. روش پیشنهادی ارائه‌شده در این مقاله، نمونه بهبودیافته روشی است که در مقاله مرجع [۵] ارائه‌شده است.

نام حملات	لایه‌های شبکه
GET flood, Slow POST, Slowloris, SQL injection, INVITE flood, Slow read	لایه کاربرد
SYN flood, UDP flood, DNS query flood, SSL MiM attack, LAND attack	لایه انتقال
Smurf attack, Teardrop, ICMP flood, Ping flood	لایه شبکه
Generating forged frames, Repeated frame header flood	لایه پیوند داده
Disrupting or breaking physical media, Signal jamming, Backhoe fade	لایه فیزیکی

شکل (۱). حملات *DDoS* در لایه‌های مختلف مدل *OSI*

ادامه این مقاله بر این اساس سازمان‌دهی شده است. در بخش ۲ به‌طور خلاصه مفاهیم پایه موردنیاز و در بخش ۳ کارهای مرتبط ارائه شده است. سپس روش پیشنهادی در بخش ۴ توضیح داده شده و در بخش ۵ پیاده‌سازی طرح پیشنهادی به‌صورت مفصل بیان گردیده است و به ارزیابی نتایج پرداخته شده است. درنهایت در بخش ۶ پژوهش صورت گرفته

* رایانامه نویسنده پاسخگو:

[2] Denial of Service
[3] Distributed Denial of Service
[4] Botnet
[5] Flooding
[6] Online
[7] Availability
[8] Behavioral analyse
[9] Traffic mining
[10] Decision tree (DT)
[11] Multilayer perceptron (MLP)
[12] Random forest (RF)

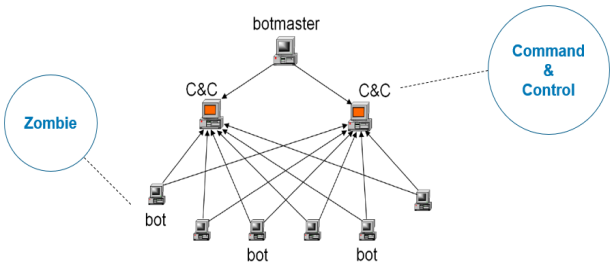
نتیجه‌گیری شده است.

۲. مفاهیم پایه

۲-۱. حمله منع سرویس توزیع‌شده

حملات DDoS از مکان‌های مختلف جغرافیایی با هدف مشترک برای از کار انداختن سرویس انجام می‌شود [۶]؛ به‌عبارت‌دیگر حملات DDoS با استفاده از تعداد زیادی دستگاه که در جهان پراکنده شده‌اند انجام می‌شود. هر دستگاه به صورت جداگانه به‌عنوان عامل DoS عمل می‌کند [۷].

در یک سناریوی معمولی DDoS، مهاجم قدرت محاسباتی بالایی را برای اجرای عملکرد فرمان و کنترل [۱] دارد که برای انتقال دستورالعمل به لایه بعدی از ماشین‌هایی به نام کنترل‌کننده‌ها [۲] استفاده می‌شود. این ماشین‌ها برای پویش [۳] سرویس‌دهنده‌ها [۴] و میزبان‌های آسیب‌پذیر در اینترنت و نصب بدافزارها برای کنترل آن دستگاه‌های آسیب‌پذیر استفاده می‌شوند. ماشین‌های تسخیرشده، زامبی نامیده می‌شوند و به‌کل این شبکه باتنت گفته می‌شود. زامبی‌های موجود در باتنت برای حمله مستقیم به هدف نهایی و ایجاد انکار سرویس استفاده می‌شوند. همچنین زامبی‌ها اطلاعات را از قربانی جمع‌آوری می‌کنند و آن را برای ارتباطات روبه‌جلو با $C\&C$ و مهاجم به کنترل‌کننده‌ها ارسال می‌کنند. در شکل (۲) یک طرح مفهومی از ساختار شبکه باتنت نشان داده‌شده است.



شکل (۲). طرح مفهومی از ساختار شبکه باتنت

۲-۲. یادگیری فعّال

روش‌های یادگیری ماشین نیاز به کاهش یا محدود کردن زمان تأخیر دسته‌بندی در هنگام اجرا دارند. برای کاهش زمان آموزش مدل یادگیری ماشین، روش‌های مختلفی ازجمله کاهش تعداد نمونه‌ها در یک جریان و استخراج ویژگی‌های مناسب وجود دارند. همچنین کاهش سربار محاسباتی می‌تواند با انتخاب‌های بهتر معماری یادگیری ماشین انجام شود. نتایج دسته‌بندی به‌دست‌آمده از روش‌های نظارت‌شده به کیفیت نمونه‌های برچسب‌دار برای یادگیری تکیه دارند. در روش‌های یادگیری ماشین نظارت‌شده، برچسب زدن مناسب نمونه‌ها در زمان واقعی معمولاً دشوار و پرهزینه است زیرا نیازمند آن است که نمونه‌های آموزشی را همان‌طور که می‌آیند، انتخاب کرده و برچسب بزنند.

برای مقابله با مشکلاتی که بیان شد، روش یادگیری فعّال معرفی شده است. در یادگیری فعّال که گاهی اوقات در ادبیات آماری یادگیری پرس‌وجو [۵] نامیده می‌شود، فرضیه اصلی این است که اگر الگوریتم یادگیری مجاز باشد داده‌هایی را که از آن یاد می‌گیرد انتخاب کند، با آموزش کمتر عملکرد بهتری خواهد داشت [۸]. یادگیری فعّال یک مورد خاص از یادگیری ماشین نیمه نظارتی و روشی برای کاهش هزینه برچسب زدن است. برای جلوگیری از افزونگی داده‌ها، مجموعه آموزش تا حد امکان کوچک نگه‌داشته می‌شود. در فرآیند یادگیری، نمونه‌های بدون برچسبی که آموزنده‌تر هستند انتخاب‌شده و مجموعه آموزش به‌طور مکرر و با توجه به قضاوت یک ناظر که برچسب‌های صحیح نمونه‌ها را می‌داند، به‌روزرسانی می‌شود. سپس مدل با نمونه‌های جدید آموزش داده می‌شود [۹]. این روند یادگیری در طول زمان ادامه می‌یابد؛ بنابراین، می‌توانیم یک مجموعه آموزش کوچک برچسب‌دار را با هزینه کم به دست آورده و مدل یادگیری ماشین را به مرور زمان تقویت کرده و دقت آن را بهبود ببخشیم.

۲-۳. روش‌های یادگیری فعّال

به‌طورکلی، سه سناریو مختلف از یادگیری فعّال وجود دارد که در ادامه به توضیح هر یک پرداخته شده است:

۲-۳-۱. سناریو نمونه‌برداری انتخابی مبتنی بر جریان [۶]

این رویکرد را یادگیری فعّال مبتنی بر جریان یا ترتیبی [۷] می‌نامند، زیرا نمونه داده‌های بدون برچسب به‌طور مداوم از منبع داده به یادگیرنده فعّال فرستاده می‌شود و هر نمونه بدون برچسب معمولاً یک‌بار از منبع داده گرفته می‌شود. یادگیرنده [۸] باید تصمیم بگیرد که آیا برچسب صحیح آن را از ناظر استعلام کرده یا از آن صرف‌نظر کند. لازم به ذکر است که در این روش، توزیع داده‌های حاصل از منبع داده ممکن است در طول زمان تغییر کند.

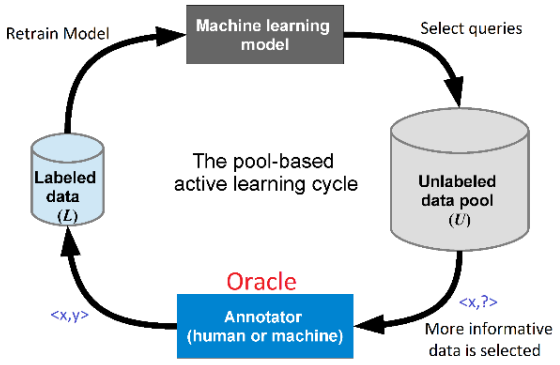
[1] Command and Control (C&C)
[2] Handlers
[3] Scan
[4] Server
[5] Query learning
[6] Stream-based
[7] Sequential
[8] Learner

۲-۳-۲. سناریو نمونه‌برداری مبتنی بر استخر[1]

در این سناریو، نمونه‌های داده برای برچسب‌گذاری توسط پیش‌گو[2] از استخر[3] نمونه داده‌های بدون برچسب انتخاب می‌شوند. به ناظری که برچسب صحیح داده‌ها را می‌داند پیش‌گو گفته می‌شود که می‌تواند انسان یا ماشین باشد. برای بسیاری از مسائل یادگیری ماشین در دنیای واقعی می‌توان مجموعه بزرگی از داده‌های بدون برچسب را به یک‌باره جمع‌آوری کرد. این انگیزه نمونه‌گیری مبتنی بر استخر است [۱۰] که فرض می‌کند مجموعه کوچکی از داده‌های دارای برچسب L و مجموعه بزرگی از داده‌های بدون برچسب موجود است. پرس‌وجوها به‌صورت انتخابی از استخر داده‌ها انجام می‌شوند.

باید به این نکته توجه داشت که معمولاً بسته بودن استخر داده‌های بدون برچسب مفروض است؛ به عبارت دیگر استخر داده‌ها ساکن و بدون تغییر است و داده‌ای به آن اضافه نشده و از آن کم نمی‌شود (اگرچه این موضوع ضروری نیست). در این سناریو برخلاف نمونه‌برداری انتخابی مبتنی بر جریان می‌توانیم به‌طور هم‌زمان از بیش از یک نمونه داده برای آموزش مدل استفاده کنیم. اکثر نمونه داده‌های اطلاعاتی از استخر داده‌های بدون برچسب و بر اساس برخی رویکردهای نمونه‌برداری یا معیار اطلاع‌رسانی[4] انتخاب‌شده‌اند. به‌طورمعمول و با توجه به معیار اطلاعاتی که برای ارزیابی همه موارد موجود در استخر داده بدون برچسب مورد استفاده قرار می‌گیرد، نمونه‌ها به‌طور حریصانه[5] مورد پرسش[6] قرار می‌گیرند؛ به‌عبارت‌دیگر با زیرمجموعه زیادی از نمونه‌ها برای پیش‌بینی به مدل یادگیری ماشین فرستاده می‌شوند تا مواردی که به اشتباه پیش‌بینی می‌شود را برای آموزش دوباره مدل شناسایی نماید.

سناریوی مبتنی بر استخر در بین مقالات پژوهشی بسیار رایج است، زیرا یک سناریو کاربردی بوده و می‌توان تنظیماتی را که برای هر مسئله مناسب‌تر است را انتخاب کرد. به عنوان مثال هنگامی که حافظه یا قدرت پردازش ممکن است محدود باشد، مانند دستگاه‌های تلفن همراه یا دستگاه‌های نهفته[7]، می‌توان تنظیمات یادگیری ماشین را متناسب با آن تنظیم کرد. سناریوی مبتنی بر استخر در بسیاری از مسائل واقعی یادگیری ماشین مانند دسته‌بندی متن، استخراج اطلاعات از متن، دسته‌بندی و بازیابی تصاویر، دسته‌بندی و بازیابی تصاویر ویدیویی و... مورد استفاده قرار گرفته است. تفاوت اصلی بین یادگیری فعّال مبتنی بر جریان و مبتنی بر استخر در این است که اولی داده‌ها را پی‌درپی پویش کرده و برای هر داده به‌صورت جداگانه تصمیم‌گیری می‌کند درحالی‌که مورد دوم قبل از انتخاب بهترین پرس‌وجو، کل مجموعه را ارزیابی و رتبه‌بندی می‌کند. در شکل (۳) چرخه یادگیری فعّال با سناریو مبتنی بر استخر داده نشان داده شده است.



شکل (۳). چرخه یادگیری فعّال با سناریو مبتنی بر استخر [۲۸]

۲-۳-۳. سناریو ترکیب پرس‌وجو عضویت[8]

در این سناریو، یادگیرنده یک نمونه جدید را با توجه به فضای ورودی تولید کرده و یک پرس‌وجو ایجاد می‌کند تا پیش‌گو به آن برچسب صحیح بزند و یادگیرنده آن را آموزش می‌بینند. این سناریو برای بسیاری از مسائل منطقی کاربرد دارد، زیرا اگر پیش‌گو یک حاشیه‌نویس انسانی[9] باشد، برچسب زدن به چنین موارد دلخواهی کار سختی است. به‌عنوان‌مثال در [۱۱]، برای آموزش یک شبکه عصبی برای دسته‌بندی نویسه‌های دست‌نویس، پرس‌وجو عضویت را با پیش‌گو انسانی انجام داده و با مشکل غیرمنتظره‌ای روبرو شدند: بسیاری از تصاویر تولید شده توسط یادگیرنده، هیچ علامت قابل تشخیصی نداشتند و فقط نویسه‌های دوگانه[10] مصنوعی بودند که معنایی نداشتند.

۲-۴. چارچوب‌های راهبرد پرس‌وجو[11]

یادگیری فعّال از راهبردهای پرس‌وجو که به طور خاص برای انتخاب مفیدترین[12] و آموزنده‌ترین[13] نمونه‌های داده طراحی شده‌اند، استفاده می‌کند. در این بخش چند نمونه از این راهبردها به‌صورت مختصر توضیح داده‌شده است.

[1] Pool-based
[2] Oracle
[3] Pool
[4] Informativeness
[5] Greedy fashion
[6] Queried
[7] Embedded Device
[8] Membership Query
[9] Human Annotator
[10] Hybrid character
[11] Query Strategy Frameworks
[12] Most useful
[13] Informative

۲-۴-۱. نمونه‌برداری عدم قطعیت[1]

ساده‌ترین و متداول‌ترین چارچوب پرس‌وجو، نمونه‌برداری عدم قطعیت است. در این چارچوب، یک یادگیرنده فعّال مواردی را جستجو می‌کند که نحوه برچسب زدن در آن‌ها قطعیت کمی دارد. این رویکرد اغلب برای مدل‌های یادگیری احتمالی مورد استفاده قرار می‌گیرد. به عنوان مثال، هنگام استفاده از یک مدل احتمالی برای دسته‌بندی دودویی[2]، نمونه‌گیری عدم قطعیت به سادگی نمونه‌ای را که احتمال مثبت بودن آن نزدیک به ۰/۵ است را پرس‌وجو می‌کند [۸].

ساده‌ترین معیار عدم قطعیت دسته‌بندی توسط رابطه (۱) تعریف می‌شود:

$$U(x) = 1 - P(\hat{x}|\, x) \qquad (۱)$$

(در رابطه فوق، x نمونه مورد پیش‌بینی و $\hat{x}$ محتمل‌ترین پیش‌بینی است.)

۲-۴-۲. پرس‌وجو با استفاده از کمیته[3]

در این روش چندین مدل مشخص با مجموعه داده ایجاد می‌شود و به جمع این مدل‌ها کمیته گفته می‌شود. مدل‌های مختلف می‌توانند از ساختارهای مختلف، ابر پارامترهای مختلف یا الگوریتم‌های مختلف باشند. به عنوان مثال، یک مدل می‌تواند SVM باشد، مدل دوم می‌تواند درخت تصمیم و مدل سوم رگرسیون لجستیک و... باشد. اکنون در میان این کمیته از مدل‌های مختلف، اختلاف‌نظر در پیش‌بینی‌ها را برای یک نمونه داده خاص اندازه‌گیری می‌کنیم. آموزنده‌ترین پرسش نمونه‌ای است که در آن بیشتر مدل‌ها موافق نیستند. یادگیرنده فعّال تصمیم می‌گیرد که در صورت ایجاد اختلاف برچسب از برچسب واقعی، پیش‌گو را برای برچسب‌گذاری صحیح فراخوانی کند. اختلاف نظر در کمیته را می‌توان با دو روش آنتروپی و *KL*-Divergence محاسبه کرد.

۳. کارهای مرتبط

در سال‌های اخیر پیشرفت چشمگیری در دقت تشخیص حملات DDoS در سامانه‌های تشخیص نفوذ[4] با استفاده از الگوریتم‌های یادگیری ماشین صورت گرفته است. ازجمله این تحقیقات می‌توان به تشخیص حملات DDoS از ترافیک وب و شبکه با دسته‌بندی ترافیک نرمال و ترافیک حمله DDoS اشاره کرد. همچنین تحقیقات زیادی برای شناسایی وجود بات‌نت از طریق تحلیل و دسته‌بندی ترافیک شبکه صورت گرفته است. در ادامه چند نمونه از تحقیقات انجام‌شده در حوزه وب و شبکه مورد بررسی قرارگرفته است.

۳-۱. تشخیص حملات DDoS از ترافیک شبکه

سوده حسینی و مهرداد عزیزی [۱۲] یک چارچوب ترکیبی جدید مبتنی بر رویکرد جریان داده برای تشخیص حمله *DDoS* با یادگیری افزایشی[5] پیشنهاد داده‌اند. در روش پیشنهادی از فنی که بار محاسباتی بین سرویس‌گیرنده و پروکسی را بر اساس میزان منبعشان تقسیم می‌کند تا کار را با سرعت بالا سازمان‌دهی کند، استفاده شده است. در این چارچوب سعی شده است تا از الگوریتم‌های چندگانه با هم استفاده کرده تا از تمام ویژگی‌های الگوریتم‌ها به‌طور هم‌زمان بهره برده شود. همچنین یک تعیین‌کننده[6] استفاده می‌شود که به بهبود نتایج کمک می‌کند. چارچوب ارائه شده مبتنی بر دو مجموعه داده مجزای *nsl-KDD* و مجموعه داده ارائه شده در [۱۳] است. در روش ارائه شده در سمت پروکسی از بیز ساده، جنگل تصادفی، درخت تصمیم، پرسپترون چندلایه و k نزدیک‌ترین همسایه (*KNN*) برای ایجاد نتایج بهتر استفاده شده است. نتایج نشان می‌دهد که جنگل تصادفی نتایج بهتری را در میان سایر الگوریتم‌های ذکر شده ایجاد می‌کند اما در یک موقعیت خاص، هرکدام از الگوریتم‌های دیگر ممکن است بهتر کار کنند. این یکی از دلایل استفاده از چندین دسته‌بندی‌کننده است.

وحید یادگاری و احمدرضا متین‌فر [۱۴]، یک سیستم شناسایی حملات منع سرویس وب با استفاده از آنتروپی و الگوریتم ماشین بردار پشتیبان پیشنهاد داده‌اند. در این تحقیق برای شناسایی این دسته از حملات، رخداد نگاشت[7]های وب سرور با ایجاد پنجره‌های زمانی ۲۰ ثانیه‌ای و محاسبه میزان فعالیت هر آی‌پی دسته‌بندی گردیده و سپس آنتروپی مربوط به هر *IP* در پنجره زمانی محاسبه و از طریق واریانس آنتروپی پنجره‌های زمانی دارای پیوستگی تعیین و در مرحله بعد از طریق الگوریتم ماشین بردار پشتیبان، شبکه آموزش داده می‌شود تا پنجره‌های زمانی ناهنجار و درنهایت IP آدرس‌هایی که منجر به حملات منع سرویس و یا منع سرویس توزیع شده‌اند دسته‌بندی و برچسب‌گذاری شوند. مدل پیشنهادی بر روی مجموعه داده

[1] Uncertainty Sampling
[2] Binary
[3] Query-By-Committee (QBC)
[4] Intrusion detection systems
[5] Incremental learning
[6] Determiner
[7] Log

استاندارد EPA-HTTP پیاده‌سازی و نتایج آن با سایر روش‌ها مقایسه گردید که بیانگر بهبود نتایج نسبت به نتایج سایر تحقیق‌های قبل است.

راج کومار باتوچا، هاری سیتا [۱۵] یک روش تشخیص خودکار حمله منع سرویس توزیع شده با استفاده از دسته‌بندی ترافیک شبکه به دو حالت ترافیک نرمال و ترافیک DDoS توسعه داده‌اند که به نوبه خود باعث کاهش بیش‌برازش و زمان محاسباتی مدل می‌شود. در این روش ابتدا پیش‌پردازش داده‌ها برای بهبود تعمیم‌پذیری مدل انجام می‌شود. در مرحله بعد انتخاب ویژگی برای انتخاب مناسب‌ترین ویژگی‌ها اعمال می‌شود که به بهبود دقت دسته‌بندی کمک می‌کند. علاوه بر این، عملکرد مدل با استفاده از تنظیم فراپارامتر با انتخاب پارامترهای مناسب برای رویکردهای یادگیری افزایش می‌یابد. در نهایت هم ویژگی‌ها و هم فراپارامترهای بهینه به رویکردهای یادگیری نظارت شده مختلف از جمله رگرسیون لجستیک (LR)، درخت تصمیم (DT)، تقویت گرادیان (GB)، K نزدیک‌ترین همسایه (KNN) و ماشین بردار پشتیبان (SVM) داده می‌شوند. همه این آزمایش‌ها بر روی مجموعه داده CICDDoS2019 ارزیابی می‌شوند. نتایج تجربی نشان می‌دهد که مدل GB در مقایسه با روش‌های پیشرفته با دقت ۹۹,۹۷ درصد عملکرد خوبی داشته است.

۲-۳. تشخیص حملات DDoS از ترافیک وب

۳-۲-۱. تشخیص حمله DDoS در پروتکل HTTP

ایاکووس یوآنو و همکاران [۱۶] با تکرار یک فرانامه در دنیای واقعی، حملات منع سرویس توزیع شده Slowloris را در یک شبکه ارتباطی دستگاه به دستگاه (D2D) شبیه‌سازی کرده‌اند. در ارتباطات دستگاه به دستگاه، حملات منع سرویس توزیع شده بسیار مضر است زیرا می‌تواند منجر به تخریب ساختار شبکه شود. در روش پیشنهادی از الگوریتم‌های یادگیری ماشین ارتقای گرادیان Light GBM ، جنگل تصادفی، ادا بوست و XGBoost استفاده شده است. همچنین از یک مجموعه داده ویژه شبکه *D2D* که محققان این مقاله بدست آورده‌اند استفاده شده است سپس از این مجموعه داده به همراه مجموعه داده CICDDoS2019 برای آموزش یک مدل برای شناسایی حمله منع سرویس توزیع شده استفاده کرده‌اند که به شناسایی و پیشگیری از حملات DDoS (Slowloris و SYN) در چارچوب D2D در نظر گرفته شده کمک می‌کند. نتایج جمع‌آوری‌شده این روش نشان می‌دهد که هر دو مجموعه داده Slowloris و CICDDoS2019 با جنگل تصادفی به دقت بیشتری دست می‌یابند. راهکار پیشنهادی با سایر رویکردهای مرتبط مورد ارزیابی و مقایسه قرار گرفته است که عملکرد قابل توجهی را از نظر زمان شناسایی، پیشگیری، میزان پردازش، منابع حافظه مورد نیاز و مصرف باتری دستگاه نشان می‌دهد، بدون اینکه بر دقت شناسایی حمله تأثیر بگذارد.

۳-۲-۲. تشخیص حمله DDoS در پروتکل‌های TCP/UDP

آنتونی یاسچ و دیوید پولپ [۱۷] چارچوبی به نام AIMM (روش های ادغام شده هوش مصنوعی) را برای شناسایی حملات منع سرویس توزیع شده در وب پیشنهاد کرده‌اند. راه حل ارائه شده بر اساس سه ماژول است: پیش‌پردازش داده‌های ورودی به سرویس‌دهنده، دسته‌بندی و تصمیم‌گیری. آخرین مرحله، ماژول تصمیم‌گیری است که احتمال را از تمام روش‌های هوش مصنوعی پیاده‌سازی شده دریافت می‌کند و آنها را برای تصمیم‌گیری نهایی در مورد حمله، تجزیه و تحلیل و سپس تجمیع می‌کند. این ایده مبتنی بر تجزیه و تحلیل اطلاعات پروتکل‌های TCP/UDP است که به سرویس‌دهنده مورد نظر می‌رسد. در این روش از دو الگوریتم متفاوت شبکه‌های عصبی و k نزدیک‌ترین همسایه استفاده شده است. همچنین برای تجمیع خروجی‌ها روش‌های استنتاج مجموعه‌های نرم[1] و میانگین‌گیری وزنی مورد استفاده قرار گرفته است. چارچوب ارائه شده با مجموعه داده در دسترس عموم به نام BOUN DDoS تحت آزمایش عملکرد قرار گرفت و مدل آموزش داده شده به دقت ۹۹/۵ ٪ رسید.

۳-۲-۳. تشخیص حمله DDoS در پروتکل DNS

لیگو چنا و همکاران [۱۸]، شناسایی حمله DNS DDoS Water Torture را به عنوان یک مشکل دسته‌بندی شناخته و از یک مدل مبتنی بر جنگل تصادفی برای دسته‌بندی ترافیک در اسپارک[2] استفاده کرده‌اند. هدف مدل پیشنهادی این نیست که مشخص شود که آیا سرویس‌دهنده DNS تحت حمله است یا خیر، بلکه پرس‌وجوهای منظم را از پرس‌وجوهای غیرعادی متمایز می‌کند. نتایج نشان می‌دهد که این مدل دقت ۹۹/۰۲ ٪ و ۰٪ FPR و ۴/۳۶ ٪ FNR را به دست آورده است. این موضوع به این معنی است که در عمل می‌تواند برای مقابله با جریان پرس‌وجو بزرگ DNS استفاده شود.

[1] Soft sets inference
[2] Spark

۳-۳. تشخیص بات‌نت

توسعه و انطباق سریع اینترنت اشیا[1] مشکلات جدیدی را برای امن‌سازی این دستگاه‌ها و شبکه‌های متصل به هم ایجاد کرده است. صدها هزار دستگاه اینترنت اشیا با آسیب‌پذیری‌های امنیتی اساسی وجود دارد، مانند احراز هویت و سطح دسترسی نادرست که آن‌ها را در برابر آلودگی به بدافزار آسیب‌پذیر می‌کند. بات‌نت‌های اینترنت اشیا برای رشد و رقابت با یکدیگر بر سر دستگاه‌ها و شبکه‌های ناامن طراحی شده‌اند. پس از آلوده شدن دستگاه به بدافزار، آن دستگاه تبدیل به یک سرویس‌دهنده فرمان و کنترل[2] که قربانی را از طریق سایر روبات‌های شبکه مورد حمله منع سرویس توزیع شده قرار می‌دهد تبدیل می‌شود.

میکال موتیلینسکی و همکاران [۱۹] روشی برای پیش‌پردازش مجموعه داده‌های *IoT-Bot* و دسته‌بندی انواع حملات مختلف منع سرویس توزیع شده ارائه داده‌اند. در روش ارائه شده در این مقاله به منظور دسته‌بندی ترافیک شبکه دستگاه‌های اینترنت اشیاء از الگوریتم‌های یادگیری ماشین k نزدیک‌ترین همسایه، نسخه تسریع شده جنگل تصادفی برای واحد پردازش گرافیکی، ماشین بردار پشتیبان و طبقه‌بندی کننده‌های رگرسیون لجستیک از کتابخانه cuML استفاده شده است. در حالی که پیاده‌سازی‌های کتابخانه *Scikit-Learn* پیشرفته در نظر گرفته می‌شوند و در اکثر کارهای تحقیقاتی در مورد شناسایی ربات *IoT* مورد استفاده قرار می‌گیرند، اما فقط می‌توانند از واحد پردازنده مرکزی استفاده کنند که زمان آموزش و مرحله پیش‌بینی طولانی دارند و این یک اشکال بزرگ است. نوآوری مقاله در روش ارائه شده در این مقاله، استفاده از الگوریتم‌های شتاب‌دهنده واحد پردازش گرافیکی می‌باشد. یکی از مهم‌ترین بخش‌های روش پیشنهاد شده، بخش مهندسی ویژگی‌ها برای تشخیص بهترین ویژگی‌ها برای ایجاد مجموعه‌ای کوچک‌تر از داده‌ها برای مراحل آموزش مدل می‌باشد. در روش ارائه شده در این مقاله از الگوریتم‌های یادگیری ماشین ذکر شده، هم با استفاده از پردازش واحد پردازنده مرکزی و هم پردازش با واحد پردازش گرافیکی مورد بررسی قرار گرفتند. نتایج بررسی به وضوح نشان داده است که عملکرد جنگل تصادفی از سایر الگوریتم‌ها در شناسایی تمام انواع حملات منع سرویس توزیع شده بهتر بوده است.

۳-۴. تشخیص حملات DDoS با رویکرد یادگیری فعّال

راپ کومار دکا و همکاران [۹]، انتخاب ویژگی توزیع‌شده در سامانه‌های تشخیص نفوذ را با استفاده از محاسبات موازی مورد بحث قرار داده و یک الگوریتم موازی تجمیعی برای رتبه‌بندی ویژگی‌های[3] مجموعه داده برای دسته‌بندی مقرون‌به‌صرفه ترافیک شبکه ارائه داده‌اند. همچنین از مجموعه داده‌های DARPA, CAIDA, ISCX-IDS و TU-datasets برای اعتبارسنجی روش ارائه‌شده، استفاده کرده‌اند.

الگوریتم رتبه‌بندی ویژگی پیشنهاد شده بر روی مجموعه داده‌های بزرگ (۱۰۰۰۰۰۰-۵۰۰۰۰ مورد) بهترین ویژگی‌های ممکن از مجموعه داده‌های ذکرشده فوق را پیدا کرده و دقت بالایی (۹۲٪-۹۷٪) را در یک محیط موازی ایجاد می‌کند که به‌طور قابل توجهی زمان کمتری (۷۱ درصد کم‌تر از یک محیط متوالی) را برای آموزش به خود اختصاص می‌دهد. برای ارزیابی سیستم پیشنهادی از ۵ دسته‌بندی کننده معروف k نزدیک‌ترین همسایه، دسته‌بندی کننده خطی[4]، رگرسیون لجستیک، Boosting و درخت تصمیم پیچیده استفاده شده است. همچنین در مورد اهمیت یادگیری فعّال برای انتخاب نمونه‌های مناسب با یک ماژول خبره[5] در یک روش بدون نظارت برای آموزش دسته‌بندی‌کننده دودویی ماشین بردار پشتیبان برای تشخیص ترافیک حملات DDoS بحث شده است. رویکرد یادگیری فعّال استفاده شده در این مقاله، دسته‌های کوچکی از نمونه‌های آموزشی را از مجموعه داده انتخاب می‌کند تا دسته‌بندی ترافیک شبکه را با دقت بالا به دست آورد. روش پیشنهادی بر روی داده‌های بزرگ، دقت بهتری در دسته‌بندی را با نمونه‌های آموزشی کمتر ارائه می‌دهد.

۴. روش پیشنهادی

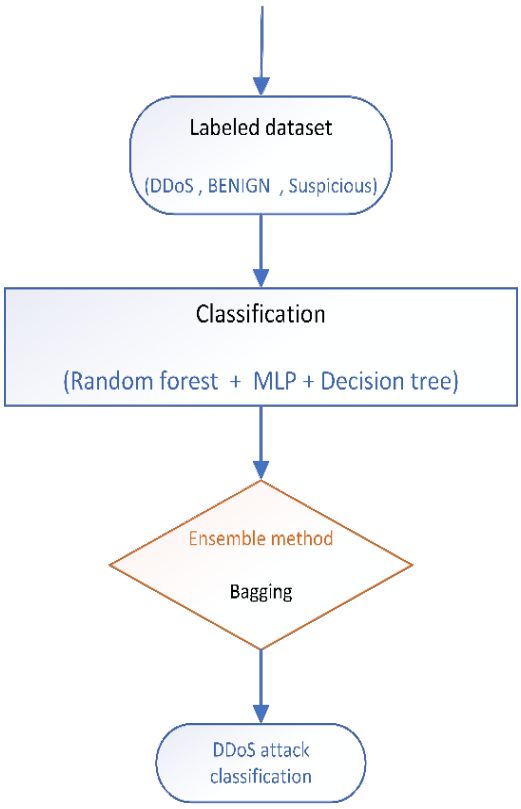
با توجه به حجم بالای رخداد نگاشت‌های تولیدشده توسط دستگاه‌های نظارت بر شبکه مانند دیوار آتش[6] و سامانه‌های تشخیص و جلوگیری از نفوذ، دسته‌بندی ترافیک در زمان واقعی در چنین ترافیک گسترده‌ای از شبکه نیاز به سامانه‌ای دارد که سربار زمانی کمی برای آموزش آن وجود داشته و بتواند عملکرد خود را با شناسایی حملات جدید بهبود ببخشد [۹]. به همین دلیل در روش ارائه شده این مقاله، مدل‌های دسته‌بندی درخت تصمیم، پرسپترون چندلایه و جنگل تصادفی را به روش گروهی و با دو رویکرد مختلف یادگیری دسته‌ای و یادگیری فعّال مورد

[1] IOT
[2] Command-and-Control (C&C)
[3] Parallel cumulative ranker algorithm
[4] Linear Discriminant
[5] Expert module
[6] Firewall

بررسی و مقایسه قرار داده‌ایم که در ادامه به‌صورت تفصیلی بیان گردیده است.

۴-۱. یادگیری دسته‌ای

در این روش مجموعه داده را با استفاده از الگوریتم‌های یادگیری ماشین درخت تصمیم، پرسپترون چندلایه و جنگل تصادفی با دو روش گروهی تجمیع‌پذیری و رای‌گیری[1] به‌صورت جدا دسته‌بندی می‌کنیم. در شکل (۴) رویکرد یادگیری دسته‌ای طرح پیشنهادی نشان داده‌شده است.



شکل (۴). سیستم پیشنهادی مقاله - یادگیری دسته‌ای

۴-۲. یادگیری فعّال

اگر یک روش دسته‌بندی بتواند به‌طور مداوم ترافیک *DDoS* را با دقت بالا از ترافیک نرمال در تمام حالات ممکن متمایز کند، ایده‌آل خواهد بود؛ اما برای شناسایی ترافیک *DDoS* که با ترافیک نرمال شبکه ادغام شده است مشکلات فراوانی نظیر تغییر الگوی حمله در طول زمان، حمله به پورت‌های غیراستاندارد، پورت‌های پنهان و... وجود دارند که کار شناسایی حملات *DDoS* به شبکه را سخت‌تر می‌کنند.

هنگامی‌که ترافیک شبکه حاوی پروتکل‌های مختلفی باشد، یافتن تحلیل دقیق ترافیک دشوار شده و نظارت تهاجمی‌تری توسط دسته‌بندی کننده مورد نیاز است. هم‌زمان با رشد سریع اینترنت دسته‌بندی ترافیک باید به‌صورت برخط کار کند تا بتواند اطلاعات زنده را ارائه دهد یا مطابق حمله و در زمان واقعی واکنش نشان دهد. دسته‌بندی ترافیک در زمان واقعی در چنین ترافیک گسترده‌ای از شبکه نیاز به سامانه‌ای دارد که سربار زمانی کمی برای آموزش آن وجود داشته و بتواند عملکرد خود را با شناسایی حملات جدید بهبود ببخشد [۹]. با استفاده از رویکرد یادگیری فعّال، می‌توان مشکلاتی که در پاراگراف قبل بیان شد را برطرف کرده و با دقت بیشتر دسته‌بندی ترافیک نرمال و ترافیک حمله *DDoS* را انجام داد.

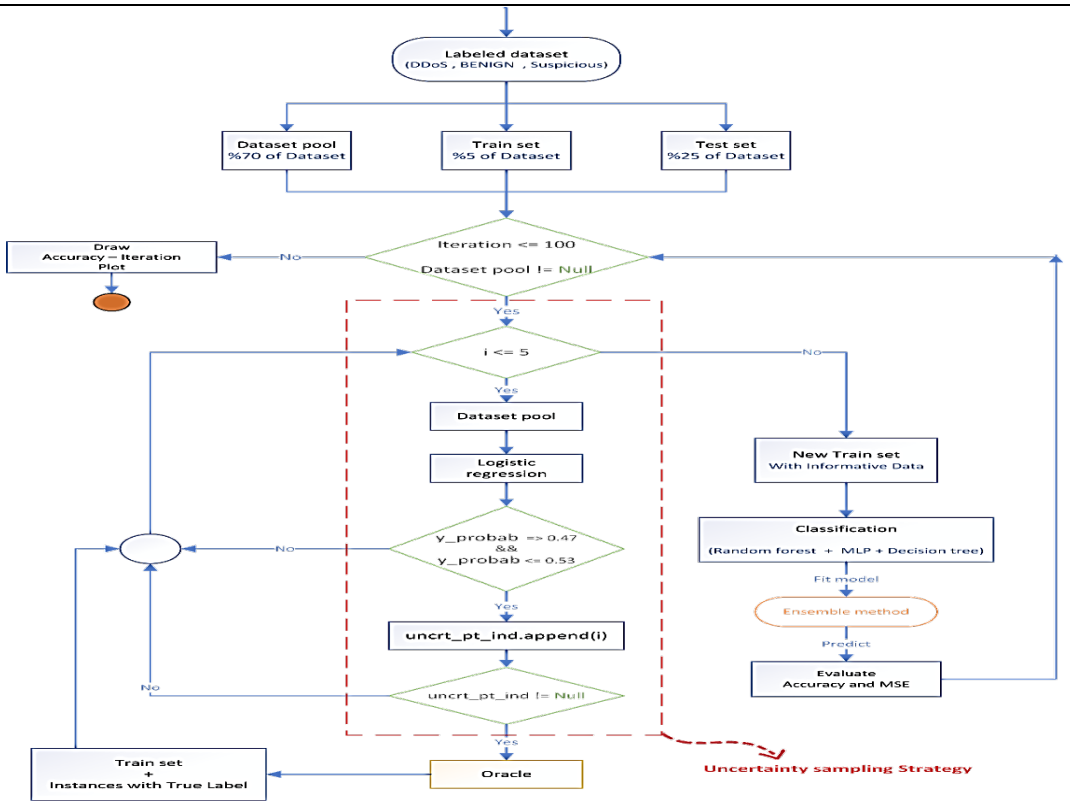
در طرح پیشنهادی از رویکرد یادگیری فعّال با سناریو نمونه‌برداری مبتنی بر استخر[2] استفاده شده است. در این سناریو کل مجموعه داده بدون برچسب را به عنوان مجموعه داده استخر در نظر می‌گیریم. در هنگام مقداردهی اولیه ۵ درصد از مجموعه داده را به همراه برچسب آن‌ها به‌عنوان مجموعه داده آموزش، ۲۵ درصد از مجموعه داده را به همراه برچسب به‌عنوان مجموعه آزمایش و بقیه مجموعه داده را به عنوان مجموعه داده استخر در نظر گرفته و مدل اولیه را ایجاد می‌کنیم. سپس دسته‌ای از داده‌های بدون برچسب را با استفاده از راهبرد عدم قطعیت، به مدل می‌دهیم تا داده‌هایی که قطعیت کمی در برچسب‌گذاری صحیح دارند مشخص شوند، سپس مدل برچسب صحیح را از پیش‌گو پرس‌وجو می‌کند. لازم به ذکر است که در سیستم پیشنهادی از پیش‌گو ماشین استفاده شده است. دلیل انتخاب پیش‌گو ماشین این است که با استفاده از مجموعه داده معیار *CICIDS2017* برچسب‌های صحیح مشخص می‌باشند.

داده‌هایی که قطعیت کمی در برچسب‌گذاری صحیح دارند به عنوان داده‌های آموزنده در نظر گرفته می‌شوند. داده‌های آموزنده شناسایی شده و توسط پیش‌گو برچسب صحیح را دریافت می‌کنند، سپس همراه با مجموعه آموزش قبل که با آن مدل را ایجاد کرده بودیم مجموعه داده آموزش جدید را ساخته و مجدداً مدل را آموزش می‌دهیم.

در شکل (۵) رویکرد یادگیری فعّال طرح پیشنهادی نشان داده شده است.

[1] Voting

[2] Pool-based



شکل (۵). سیستم پیشنهادی مقاله - دسته‌بندی - یادگیری فعّال

شبه کد الگوریتم شکل ۵ به صورت زیر است:

START
GET dataset
Create variable x_train**,** y_train**,** unlabel**,** label
Create variable LR_classifier, y_probab
CALL split_dataset _Function (dataset, 0.05, 0.25)
For Iteration = 1 to 100 and Dataset pool not equal with Null
For i Less than or equal to '5'
LR_classifier = ***CALL*** *Logistic Regression Function ()*
CALL *LR_classifier_fit model_Function (x_train, y_train)*
y_probab = ***CALL*** *LR_classifier_predict_proba_Function (unlabel) [: ,0]*
Declare an Array *uncertainty_index = []*
If *y_probab* ***greater than or equal to*** *0.47* ***And*** *y_probab* ***less than or equal to*** *0.53* ***THEN***
uncertainty_index [] = y_probab
If *uncertainty_index[i] not equal with Null* ***THEN***
y with true label = Oracle Function (x train, y train)
SET TrainSet to TrainSet + y with true label
End If
End If
End For
NewTrainSet = TrainSet
CALL *Random_forest_Classification_Function (NewTrainSet)*
CALL *Decision _tree _Classification_Function (NewTrainSet)*
CALL *Multilayer_perceptron_Classification_Function (NewTrainSet)*
CALL *Ensemble_method_Function ()*
Evaluate Accuracy and MSE (Mean squared error)
End For
END

توضیح شبه کد:

۱- ابتدا کل مجموعه داده را در یک متغیر به عنوان مجموعه داده استخر نگهداری می‌کنیم، همچنین به این دلیل که در یادگیری فعال، یادگیرنده به تدریج و با داده‌های جدید که به‌طور جریانی وارد سیستم می‌شوند آموزش داده می‌شود؛ مجموعه داده آموزش را کوچک در نظر گرفته‌ایم. بخش کوچکی از داده‌های دارای برچسب (۵ درصد به‌طور دلخواه انتخاب شده است) را به عنوان مجموعه آموزش و ۲۵ درصد را به عنوان مجموعه آزمایش در نظر می‌گیریم و رکوردهای مجموعه‌های آموزش و آزمایش را از مجموعه داده استخر حذف می‌کنیم. با استفاده از تابع *split* این کار را انجام می‌شود.

۲- روند آموزش مدل را با ۱۰۰ بار تکرار ادامه می‌دهیم تا به دقت مطلوبی برسیم. دومین شرط ادامه تکرار حلقه خالی نبودن استخر است. در هر بار آموزش مدل تعدادی از داده‌های مجموعه آموزش نادرست و در برخی موارد به درستی دسته‌بندی می‌شوند، به همین دلیل با یک حلقه ۵ بار داده‌های آزمایش را بررسی کرده و اندیس داده‌هایی که اشتباه دسته‌بندی می‌شوند را در آرایه uncertainty_index نگه‌داری می‌کنیم. (Uncertain به معنای نامشخص است.) همچنین از پیش‌گو ماشین برای زدن برچسب صحیح به نمونه‌های مشخص‌شده در آرایه uncertainty_index استفاده‌شده است.

۳- درصورتی‌که آرایه *uncertainty_index* خالی از عضو نباشد به این معنی است که داده مورد نظر قطعیت کمی دارد و آموزنده تشخیص داده می‌شود بنابراین باید برچسب صحیح توسط پیش‌گو تعیین شود. سپس مجموعه داده آموزش را با داده جدید به‌روزرسانی می‌کنیم.

۴- پس از به‌روزرسانی مجموعه داده آموزش، مدل را دوباره ایجاد می‌کنیم. لازم به ذکر است که از سه دسته‌بند ذکرشده در بخش‌های قبل به‌صورت روش گروهی با رویکرد رأی اکثریت استفاده شده است. دقت هر مرحله محاسبه شده و این روند تا مشاهده کامل مجموعه داده استخر ادامه می‌یابد.

در این مقاله با رویکرد یادگیری فعّال و با استفاده از آموزنده‌ترین داده‌هایی که از طریق راهبرد عدم قطعیت انتخاب شده‌اند و برچسب صحیح آن‌ها از پیش‌گو پرس‌وجو شده است، می‌توان با مجموعه داده کوچک‌تر به‌دقت بالایی دست یافت.

۵. پیاده‌سازی

پیاده‌سازی طرح پیشنهادی در ژوپیتر نوت‌بوک [1] و با کتابخانه‌های Scikit-learn، Pandas و Numpy انجام‌شده و توسعه نهایی کدها در محیط pycharm صورت گرفته است. *Scikit-Learn* یک کتابخانه یادگیری ماشین رایگان برای پایتون است [۲۶].

۵-۱. مجموعه داده

مجموعه داده تشخیص نفوذ معیار CICIDS2017، در سال ۲۰۱۷ توسط CIC[2] که یک موسسه کانادایی برای امنیت سایبری است، ارائه شده است [۲۲]. حملات DDoS قرار داده شده در مجموعه داده CICIDS2017، در ابتدا شامل ۲۲۵،۷۴۵ رکورد است که از آن‌ها ۹۷،۷۱۸ به‌عنوان «Benign» به معنای ترافیک بی‌خطر برچسب‌گذاری می‌شوند درحالی‌که ۱۲۸،۰۲۷ رکورد باقیمانده «DDoS» هستند.

۵-۲. انتخاب ویژگی‌ها

با توجه به [۲۳]، برای شناسایی حملات DDoS دو دسته ویژگی زیر مورد استفاده قرار می‌گیرند.

- ویژگی‌های سطح بسته[3]: ویژگی‌های خاص بسته مانند آدرس IP، پرچم‌ها، طول بار مفید و مؤثر و... می‌باشند. این ویژگی‌ها حاوی اطلاعات بسیار کمتری برای تشخیص حملات اخیر هستند زیرا وضعیت واقعی بسته‌ها به‌راحتی پنهان می‌شوند. به این ویژگی‌ها، ویژگی‌های درگیر با بسته‌های منفرد نیز گفته می‌شود.

- ویژگی‌های سطح جریان[4]: این ویژگی‌ها به محتوای بسته‌ها وابستگی ندارد، درنتیجه نسبت به آخرین شکل رمزنگاری و مبهم سازی[5] که توسط مهاجمان بکار می‌رود تا روبات‌های خود را پنهان کنند مصون است. این ویژگی‌ها شامل تأخیر، نرخ و ویژگی‌های بهره‌برداری[6] می‌شوند.

اگرچه تقریباً تمام مجموعه داده‌های معیار شامل هر دو نوع ویژگی‌ها می‌شوند، ثابت شده است که ویژگی‌های سطح جریان دسته‌بندی کننده‌های بهتری هستند و مجموعه داده‌های کوچک‌تر متشکل از تنها ویژگی‌های سطح جریان می‌توانند به طور مؤثر حملات DDoS را تشخیص دهند [۲۳]. با توجه به نکته ذکر شده و با انتخاب ویژگی‌های درست، می‌توانیم حتی با مجموعه کوچک‌تری از ویژگی‌ها، حملات DDoS را با دقت بالا شناسایی نماییم [۲۴].

[1] Jupyter notebook

[2] Canadian Institute for Cybersecurity
[3] Packet level
[4] Flow level
[5] Obfuscation
[6] Utilization

مجموعه داده CICIDS2017 شامل ۷۸ ویژگی کلی سطح جریان و سطح بسته از رخداد نگاشت‌های ترافیک شبکه نرمال به همراه ترافیک حمله DDoS است. با بررسی‌های اولیه در سایت مرجع این مجموعه داده دریافتیم که سازندگان مجموعه داده CICIDS2017 ویژگی‌ها را به ترتیب میزان مؤثر بودن پیشنهاد داده‌اند. [۲۵] به همین دلیل در این مقاله از میان ۷۸ متغیر موجود در این مجموعه داده، ۴ ویژگی مهم را مطابق جدول (۱) انتخاب کرده و فضای ویژگی را کاهش می‌دهیم.

جدول (۱). ویژگی‌های انتخاب‌شده

نام ویژگی	توضیح
Bwd Packet Length Std	طول بسته در مدت‌زمان اتصال (بایت در ثانیه)
Average Packet Size	متوسط طول بسته
Flow Duration	مدت‌زمان اتصال (به ثانیه)
Flow IAT Std	بسته‌هایی که خارج از زمان ارسال‌شده است. (انحراف استاندارد زمان)

۵-۳. پیش‌پردازش مجموعه داده

به منظور کاهش سربار فضای حافظه کامپیوتر در هنگام پیاده‌سازی سیستم پیشنهادی از ۱۰،۰۰۰ نمونه داده (از مجموع ۲۲۵،۷۴۵) که با روش برهم‌زدن و تصادفی‌سازی [1] استخراج کرده‌ایم استفاده شده است. این زیرمجموعه از مجموعه داده اصلی دارای سهم مشابهی از ترافیک نرمال و ترافیک DDoS است. بدین ترتیب ۴۳۴۵ نمونه نرمال و ۵۶۵۵ نمونه DDoS به دست آمده است. برای پر کردن مقادیر گمشده [2] از راهبرد پر کردن با مقدار میانگین مقادیر موجود در همان ستون یک ویژگی استفاده شده است. همچنین با استفاده از نرمال‌سازی، مقادیر پراکنده ویژگی‌های مختلف را به یک مقیاس مشخص می‌آوریم تا بزرگ یا کوچک بودن یک ویژگی تأثیر زیادی بر مدل ایجادشده نگذارد. لازم به ذکر است که در سیستم پیشنهادی از رویکرد *min-max* در دامنه $\{۱، ۰\}$ که بسیار متداول است، استفاده شده است.

[1] Shuffling and randomization
[2] Missing values

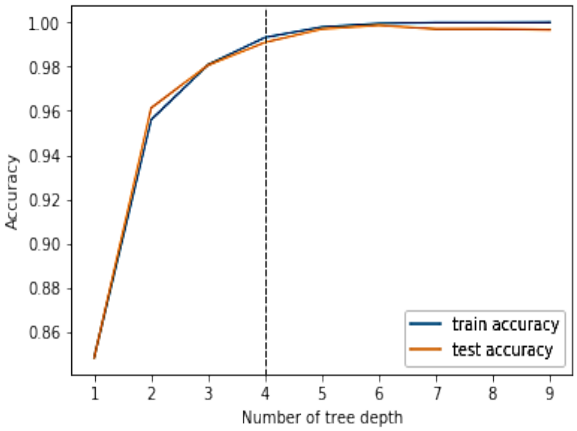
۵-۴. تنظیم پارامترها در مدل‌های دسته‌بندی

در سیستم پیشنهادی از مدل‌های یادگیری ماشین درخت تصمیم، پرسپترون چندلایه و جنگل تصادفی با دو رویکرد گروهی تجمیع‌پذیری [3] و رأی اکثریت [4] برای ایجاد مدل و دسته‌بندی داده‌های جدید استفاده شده است. الگوریتم‌های یادگیری ماشین دارای ابر پارامترهایی [5] هستند که به ما امکان می‌دهند رفتار الگوریتم را با مجموعه داده خاص خود تنظیم کنیم. به‌طور معمول، چگونگی استفاده از مقادیر برای ابر پارامترهای یک الگوریتم داده شده در یک مجموعه داده خاص چالش‌برانگیز است [۲۷]. در سیستم پیشنهادی، با تنظیم ابر پارامترهای بهینه عملکرد و دقت مدل‌ها را افزایش داده‌ایم.

یکی از ابر پارامترهای مهم در الگوریتم درخت تصمیم، عمق درخت است. عمق درخت تعیین می‌کند که مدل ایجاد شده چه میزان انعطاف‌پذیر است. هر اندازه درخت عمیق‌تر شود دقت افزایش می‌یابد اما از یک عمقی به بعد دقت مجموعه آزمایش خراب می‌شود زیرا برازش بیش‌ازحد [6] اتفاق افتاده و مدل ایجاد شده تمام داده‌های مجموعه آموزش را حفظ می‌کند. همچنین ممکن است از یک عمقی دقت مجموعه آزمایش تغییر زیادی نداشته و در صورتی که به عمق بیشتری برویم، صرفاً مدل خود را پیچیده و زمان یادگیری را طولانی‌تر نماییم. به همین دلیل تعیین ابر پارامتر عمق درخت یکی از معیارهای اساسی در تنظیم کردن [7] درخت تصمیم و دستیابی به نتایج مطلوب این الگوریتم یادگیری ماشین است.

همان‌طور که در طرح [8] رسم شده شکل (۶) نشان داده شده است، میزان دقت مرحله آموزش و آزمایش را با افزایش عمق درخت به دست آورده‌ایم. درنتیجه عمق ۴ بهینه‌ترین مقدار برای ابر پارامتر max_depth است؛ زیرا در این عمق به بیشترین دقت دست‌یافته‌ایم و از آن عمق به بعد تغییر زیادی در دقت حاصل نمی‌شود.

[3] Bagging
[4] Voting Majority
[5] Hyperparameter
[6] Over fitting
[7] Tune
[8] Plot



شکل (۶). تعیین عمق بهینه برای درخت تصمیم

مقادیر بهینه به‌دست‌آمده ابر پارامترهای بهینه پرسپترون چندلایه در جدول (۲) نشان داده‌شده است.

جدول (۲). ابر پارامترهای بهینه پرسپترون چندلایه

Best Parameters:	Value
activation	tanh
Alpha	0.05
hidden_layer_sizes	(4,4,4)
learning_rate	adaptive
solver	adam
Best Score found: 99.68%	

مزیت عمده استفاده از الگوریتم جنگل تصادفی، رویکرد تصمیم‌گیری گروهی آن بوده که باعث جلوگیری از مشکل برازش بیش‌ازحد [1] می‌شود و به‌عنوان روشی دقیق و قوی در زمینه دسته‌بندی شناخته شده است. به دلیل مزایای این الگوریتم در سیستم پیشنهادی مورد استفاده قرارگرفته است. مقادیر بهینه به‌دست‌آمده ابر پارامترهای جنگل تصادفی در جدول (۳) نشان داده شده است.

[1] Over fitting

جدول (۳). ابر پارامترهای بهینه جنگل تصادفی

Best Parameters:	Value
criterion	entropy
n_estimators	300
Best Score found: 99.79%	

۵-۵. رویکردهای دسته‌بندی

برای بخش دسته‌بندی از سیستم پیشنهادی دو رویکرد یادگیری دسته‌ای و یادگیری فعّال را با استفاده از ابر پارامترهای بهینه‌ای که برای هر الگوریتم یادگیری ماشین در بخش قبل یافت شد پیاده‌سازی کرده‌ایم.

۵-۵-۱. یادگیری دسته‌ای

در رویکرد یادگیری دسته‌ای، دو روش تجمیع‌پذیری و دسته‌بند مبتنی بر رأی‌گیری [2] به‌صورت جدا انجام گرفته است که در ادامه به توضیح هر یک پرداخته شده است.

تجمیع‌پذیری [3]

در این روش نمی‌توان از دسته‌بندهای مختلف به‌طور هم‌زمان استفاده کرد، به همین دلیل با استفاده از یک حلقه و برای هر الگوریتم یادگیری ماشین پیشنهاد شده، یک‌بار روش تجمیع‌پذیری را اجرا کرده و خروجی را با هم مورد مقایسه قرار داده‌ایم. پس از اجرای کد برای الگوریتم‌های *DT*، *RF* و *MLP* به ترتیب ۹۸/۵۰ درصد، ۹۹/۰۱ درصد و ۹۹/۸۱ درصد دقت به دست‌آمده است.

دسته‌بند مبتنی بر رأی‌گیری

برای پیاده‌سازی این روش ابتدا یک نسخه از مجموعه داده آموزش را در اختیار هر الگوریتم یادگیری ماشین قرار داده و پس از ایجاد و آموزش مدل با رأی اکثریت نتیجه نهایی مشخص می‌شود. پس از اجرای کد، دقت ۹۸/۳۱ درصد به دست آمده است.

۵-۵-۲. یادگیری فعّال

از سناریوهای مختلف یادگیری فعّال، در سیستم پیشنهادی از سناریو نمونه‌برداری مبتنی بر استخر استفاده شده است. این روش یکی از بهترین سناریوهای شناخته‌شده در زمینه یادگیری

[2] Voting Classifier
[3] Bagging

فعّال است. ابتدا کل مجموعه داده را در یک متغیر به عنوان مجموعه داده استخر نگهداری می‌کنیم، سپس بخش کوچکی از داده‌های دارای برچسب (۵ درصد) را به عنوان مجموعه آموزش و ۲۵ درصد را به‌عنوان مجموعه آزمایش در نظر می‌گیریم و رکوردهای مجموعه‌های آموزش و آزمایش را از مجموعه داده استخر حذف می‌کنیم.

با استفاده از مجموعه داده آموزش، یک مدل اولیه ایجاد کرده و با داده‌های مجموعه آزمایش مدل را اعتبارسنجی می‌کنیم. همچنین در این مقاله از چارچوب راهبرد پرس‌وجو عدم قطعیت استفاده شده است. در این راهبرد بر اساس احتمال وجود نمونه داده در یک کلاس، نقاط ارزشمند برای یادگیری تعیین می‌شود. در دسته‌بند رگرسیون لجستیک نزدیک‌ترین نقطه به حد آستانه (احتمال = ۰/۵) نامشخص‌ترین نقطه است؛ به عبارت دیگر نقطه‌ای است که قطعیت کمی دارد؛ بنابراین، به دلیل یافتن تمام نقاطی که قطعیت کمی دارند یک بازه با ۰/۳ بیشتر و کمتر (احتمال بین ۰/۴۷ تا ۰/۵۳) به عنوان دامنه عدم اطمینان انتخاب شده است. این مقدار تخمین زده شده است. لازم به ذکر است، برای استفاده از دسته‌بند رگرسیون لجستیک که یک دسته‌بند دودویی است از مجموعه داده *CICIDS2017*، داده‌های مشکوک را برای انجام بررسی بیشتر از مجموعه داده استخر حذف می‌کنیم؛ بنابراین مسئله دسته‌بندی به‌صورت دودویی (دسته‌بندی ترافیک نرمال و ترافیک *DDoS*) خواهد شد.

پس از اجرای حلقه، نمونه‌هایی از مجموعه داده که قطعیت کمی دارند شناسایی شده و به عنوان داده‌های جدید به همراه مجموعه آموزش قبلی، یک مجموعه آموزش جدید ایجاد کرده و مدل را مجدداً آموزش می‌دهیم. همچنین روند آموزش مدل را با ۱۰۰ بار تکرار ادامه می‌دهیم تا به دقت مطلوبی برسیم. به دلیل این که در هر بار آموزش مدل تعدادی از داده‌های مجموعه آموزش نادرست و در برخی موارد به‌درستی دسته‌بندی می‌شوند، به همین دلیل با یک حلقه ۵ بار داده‌های آزمایش را بررسی کرده و اندیس داده‌هایی که اشتباه دسته‌بندی می‌شوند را در آرایه *uncrt_ind* نگه‌داری می‌کنیم.

لازم به ذکر است که از پیش‌گو ماشین برای زدن برچسب صحیح به نمونه‌های مشخص‌شده در آرایه *uncrt_ind* استفاده شده است.

```
for i in range(100):
    # split dataset into train (%5), test (%25), unlabel (%70)
    x_train, y_train, x_test, y_test, unlabel, label = split (dataset, 0.05, 0.25)

    # Query Strategy Framework -->  Uncertainty Sampling
    for i in range(5):
      LR_classifier = LogisticRegression()
      LR_classifier.fit(x_train, y_train)
      y_probab = LR_classifier.predict_proba(unlabel)[:,0]
      p = 0.47 # range of uncertanity 0.47 to 0.53
      uncrt_ind = []
      for i in range(unlabel.shape[0]):
        if (y_probab[i] >= p and y_probab[i] <= 1-p):
          uncrt_ind.append(i)
      x_train = np.append (unlabel[uncrt_ind, :], x_train, axis = 0)
      y_train = np.append (label[uncrt_ind], y_train)
      unlabel = np.delete (unlabel, uncrt_ind, axis = 0)
      label = np.delete (label, uncrt_ind)
```

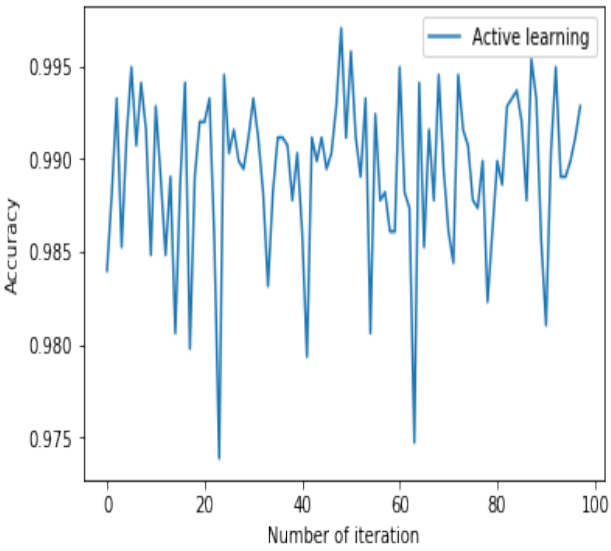
پس از به‌روزرسانی مجموعه داده آموزش، مدل را دوباره ایجاد می‌کنیم.

```
# Train model by Active learning
tr = DecisionTreeClassifier(max_depth = 4 ,
max_features = 4 ,min_samples_leaf = 5)

mlp = MLPClassifier(hidden_layer_sizes=(4,4,4),
 max_iter=200, alpha=0.05, solver='adam', random_state=1)

rf = RandomForestClassifier (criterion='entropy',
 n_estimators=300, max_depth=3)
```

لازم به ذکر است که از سه دسته‌بند ذکرشده در بخش قبل به‌صورت روش گروهی با رویکرد رأی اکثریت استفاده شده است. دقت هر مرحله محاسبه شده و در آرایه *acc_active* ذخیره می‌گردد.

```
evc = VotingClassifier(estimators= [('tr', tr), ('rf', rf), ('mlp', mlp)],voting='hard')
evc.fit(x_train, y_train)
acc_active.append(evc.score(x_test, y_test))
```

این روند تا اتمام رکوردهای مجموعه استخر تکرار می‌شود. در طرح رسم شده در شکل (۷) دقت یادگیرنده فعّال در هر تکرار نشان داده شده است.



شکل (۷). دقت دسته‌بندی با رویکرد یادگیری فعّال

۵-۶. ارزیابی نتایج

برای افزایش دقت طرح پیشنهادی نسبت به روش ارائه شده در مقاله مرجع از دسته‌بندهای پرسپترون چندلایه، درخت تصمیم و جنگل تصادفی با روش گروهی و دو رویکرد یادگیری دسته‌ای و یادگیری فعّال استفاده شد. همچنین برای اطمینان از عدم ایجاد مشکل برازش بیش‌ازحد، روش گروهی با روند اعتبارسنجی متقابل[۱] *10-fold* به کار گرفته شد؛ بنابراین دقت به دست آمده بدون مشکل برازش بیش‌ازحد صحت دارد.

با یافتن ابر پارامترهای بهینه برای مدل‌های دسته‌بندی انتخاب شده در این مقاله سعی در بهبود دقت شناسایی و دسته‌بندی ترافیک حمله منع سرویس توزیع شده از ترافیک نرمال شبکه را داشته‌ایم. در جدول (۴) دقت متوسط هر مدل یادگیری ماشین با اعتبارسنجی متقابل *10-fold* با استفاده از ابر پارامترهای بهینه برای هر مدل نشان داده شده است. لازم به ذکر است که از این ابر پارامترهای بهینه در رویکردهای یادگیری دسته‌ای و یادگیری فعّال برای افزایش دقت استفاده شده است.

جدول (۴). نتایج حاصل از مدل‌های دسته‌بندی با ابر پارامترهای بهینه

مدل یادگیری	پارامترهای بهینه‌شده	دقت متوسط با اعتبارسنجی متقابل	تشخیص نادرست
DT	max_depth: 4, max_features: 3, min_samples_leaf: 7	٪ ۹۸/۹۰	۱۶
MLP	activation: 'tanh', alpha: 0.05, hidden_layer_sizes: (4, 4, 4), learning_rate: 'adaptive', solver: 'adam'	٪ ۹۹/۶۱	۱۰
RF	criterion: 'entropy', max_depth: 3, n_estimators: 200	٪ ۹۹/۰۵	۱۵

[1] Cross Validation (CV)

۵-۶-۱. نتایج یادگیری دسته‌ای

تعداد نمونه مجموعه آموزش ۷۰۰۰ عدد و تعداد نمونه مجموعه آزمایش ۳۰۰۰ عدد انتخاب شده است. نتایج به دست آمده از یادگیری دسته‌ای با روش گروهی تجمیع‌پذیری و رأی‌گیری اکثریت در جدول (۵) نشان داده شده است.

جدول (۵). نتایج به‌دست‌آمده از یادگیری دسته‌ای با روش گروهی

مدل یادگیری	روش گروهی	دقت با اعتبارسنجی متقابل
DT	تجمیع‌پذیری	٪ ۹۸/۵۰
MLP		٪ ۹۹/۸۱
RF		٪ ۹۹/۰۱
DT	رأی‌گیری اکثریت	٪ ۹۹/۳۱
MLP		
RF		

جدول (۶). ماتریس درهم‌ریختگی برای DT با روش گروهی تجمیع‌پذیری

	BENIGN	DDoS	Suspicious
BENIGN	۱۶۸۹	۰	۱۱
DDoS	۰	۸۵۹	۳
Suspicious	۱	۰	۴۳۷

جدول (۷). ماتریس درهم‌ریختگی برای RF با روش گروهی تجمیع‌پذیری

	BENIGN	DDoS	Suspicious
BENIGN	۱۶۸۴	۰	۱۶
DDoS	۰	۸۵۹	۳
Suspicious	۷	۰	۴۳۱

جدول (۸). ماتریس درهم‌ریختگی برای MLP با روش گروهی تجمیع‌پذیری

	BENIGN	DDoS	Suspicious
BENIGN	۱۶۹۰	۱۰	۰
DDoS	۰	۸۵۳	۰
Suspicious	۶	۰	۴۴۱

۵-۶-۲. نتایج یادگیری فعّال

یکی از معایب جنگل تصادفی زمان‌بر بودن آموزش و پیش‌بینی داده‌های جدید است که این مشکل زمانی که از روش یادگیری فعّال استفاده شود برطرف می‌شود. در رویکرد یادگیری فعّال، هنگام مشاهده داده جدید درصورتی که مدل نتواند به‌درستی آن را دسته‌بندی کند دقت مدل افت پیدا کرده و در تکرار بعد به دلیل آموزش موارد اشتباه به مدل دقت افزایش می‌یابد. در برخی تکرارها نیز به دلیل عدم وجود دسته‌بندی اشتباه، میزان دقت ثابت مانده است. دقت مدل با ۱۰۰ بار تکرار و آموزش ۷۰۰۰ داده به مدل، افزایش‌یافته و دقت میانگین ۹۹/۲۰ درصد به دست آمده است.

۵-۷. مقایسه نتایج با مقاله مرجع

در مقاله مرجع از الگوریتم‌های یادگیری ماشین، k نزدیک‌ترین همسایه، ماشین بردار پشتیبان و جنگل تصادفی استفاده‌شده است. در نتایج تجربی و با تنظیم پارامترهای بهینه شده در مجموعه مقادیر معین به ترتیب ۹۵ درصد، ۹۲ درصد و ۹۹/۶۶ درصد نمرات دقت را ارائه داده‌اند. در روش پیشنهادی مقاله از الگوریتم‌های درخت تصمیم، پرسپترون چندلایه و جنگل تصادفی به روش‌های گروهی تجمیع‌پذیری و رأی‌گیری اکثریت استفاده شده است و بالاترین دقت مربوط به پرسپترون چندلایه با روش گروهی تجمیع‌پذیری و به میزان ۹۹/۸۱ درصد است. نتایج به‌دست‌آمده نشان می‌دهد که میزان دقت دسته‌بندی ترافیک نرمال از ترافیک حملات *DDoS* نسبت به مقاله مرجع به میزان قابل‌توجهی بهبود داشته است و دلیل آن استفاده از الگوریتم‌های قدرتمندتر و استفاده از روش دسته‌بندی گروهی است. همچنین استفاده از رویکرد یادگیری فعّال در سیستم پیشنهادی موجب آموزش برخط مدل یادگیری ماشین شده و نسبت به حملات روز صفر عملکرد بهتری خواهد داشت. نتایج مقایسه شده با مقاله مرجع در جدول (۹) نشان داده شده است.

جدول (۹). مقایسه نتایج با مقاله مرجع

	مدل یادگیری	نوع یادگیری	روش گروهی	دقت با اعتبارسنجی متقابل
مقاله مرجع	KNN	یادگیری دسته‌ای	استفاده نشده است.	٪ ۹۵
	SVM			٪ ۹۲
	RF			٪ ۹۹/۶۶
طرح پیشنهادی	DT	یادگیری فعّال	رأی‌گیری اکثریت و تجمیع‌پذیری	٪ ۹۹/۸۱
	MLP			
	RF			

۵-۸. مقایسه نتایج روش پیشنهادی با سایر مقاله‌ها

جدول (۱۰). مقایسه نتایج روش پیشنهادی با سایر مقاله‌ها

ردیف	عنوان پژوهش	سال انتشار	حوزه فعالیت	دسته‌بندها	بهترین دقت
۱	یک مدل یادگیری ماشین تعمیم یافته برای تشخیص حملات *DDoS* با استفاده از انتخاب ویژگی ترکیبی و تنظیم فراپارامتر *[۱۵]*	۲۰۲۱	آنالیز ترافیک شبکه	LR DT GB KNN SVM	٪ ۹۹,۹۷
۲	روش پیشنهادی ارائه شده در مقاله: سیستم تشخیص حملات *DDOS* با استفاده از روش دسته‌بندی گروهی و رویکرد یادگیری فعّال	-	آنالیز ترافیک شبکه	DT MLP RF	٪ ۹۹,۸۱
۳	تشخیص حملات *DDoS* در ارتباطات دستگاه به دستگاه با استفاده از رویکرد یادگیری ماشین *[۱۶]*	۲۰۲۳	آنالیز ترافیک وب	Light GBM RF Adaboost XGBoost	٪ ۹۹,۱
۴	روش‌های ادغام شده هوش مصنوعی برای تشخیص حملات سیل‌آسای *DDoS [۱۷]*	۲۰۲۲	آنالیز ترافیک وب	KNN Neural Networks	٪ ۹۹,۵۰
۵	یک رویکرد یادگیری ماشین مبتنی بر *GPU* برای تشخیص حملات بات‌نت *[۱۹]*	۲۰۲۲	تشخیص بات‌نت	KNN *RF* *SVM* *LR*	٪ ۹۹

۶. نتیجه‌گیری

الگوریتم‌های یادگیری ماشین برای دستیابی به بیشترین دقت و بهبود عملکرد خود نیاز به نمونه آموزشی بالایی دارند درصورتی که مجموعه داده بدون برچسب داشته باشیم و برچسب‌گذاری مجموعه داده توسط کارشناس انسانی دارای هزینه زمانی و مالی زیادی باشد مجبور به استفاده از مجموعه داده کوچک‌تر و بدون برچسب هستیم. در مسئله شناسایی حملات DDoS، به دلیل زیاد بودن روش‌ها و ابزارهای حمله مدل‌های سنتی در مقابل حملات روز صفر[1]، دچار خطا و حتی دور خوردن[2] می‌شوند. همچنین به دلیل حجم بالای رخداد نگاشت‌های تولید شده توسط دیوارهای آتش و ابزارهای شناسایی و پیشگیری از نفوذ، امکان برچسب‌گذاری داده‌های حجیم[3] شامل رخداد نگاشت دستگاه‌های نظارت بر شبکه وجود ندارد. به همین دلیل استفاده از روش یادگیری فعّال در زمینه تشخیص حمله DDoS موجب می‌شود تا با استفاده از مجموعه داده کوچک برچسب‌گذاری شده بتوان به‌دقت بالایی دست یافت.

به‌طورکلی، هرچه داده‌های بیشتری برای آموزش یک مدل در دسترس باشد، صرف نظر از این‌که از روش یادگیری فعّال استفاده می‌شود یا خیر دقت بالاتر است. با این‌حال با استفاده از رویکرد یادگیری فعّال، می‌توان با مجموعه داده‌های بسیار کمتری به یک دقت مطلوب دست یافت.

۷. مراجع

[1] Choo Kim-Kwang Raymond (2011). "The cyber threat landscape: challenges and future research directions." Compute Secure 2011;30(8):719–31.

[2] Choo Kim-Kwang Raymond (2011). "Cyber threat landscape faced by financial and insurance industry." Trends Issues Crime Criminal Justice 2011; 408:1–6.

[3] Stevanovic D, Vlajic N (2014). "Next generation application-layer DDoS defences: ap-plying the concepts of outlier detection in data streams with concept drift." In: Machine learning and applications (ICMLA), 2014 13th international conference on, Detroit, MI; 2014. p. 456–62.

[4] Aamir, M., Zaidi, S.M.A., (2015). "Denial-of-service in content centric (named data) networking: a tutorial and state-of-the-art survey." Security Commun. Networks8 (11), 2037–2059.

[1] Zero-day attacks
[2] Bypass
[3] Big Data

[5] Muhammad Aamir, Syed Mustafa Ali Zaidi (2019), "Clustering based semi-supervised machine learning for DDoS attack classification", Journal of King Saud University - Computer and Information Sciences, 3 February 2019.

[6] McGregory Steve (2013). "Preparing for the next DDoS attack." Netw Secur 2013;2013(5):5–6.

[7] Moustis D, Kotzanikolaou P (2013). "Evaluating security controls against HTTP-based DDoS attacks." In: Information, intelligence, systems and applications (IISA), 2013 fourth international conference on; 2013. p. 1–6.

[8] Burr Settles, "Active Learning", Morgan & Claypool Publishers (October 10, 2017).

[9] Rup Kumar Deka, Dhruba Kumar Bhattacharyya, Jugal Kumar Kalita (2019). "Active learning to detect DDoS attack using ranked features" In: Computer Communications 145 (2019); 203–222.

[10] D. Lewis and W. Gale. "A sequential algorithm for training text classifiers." In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, pages 3–12. ACM/Springer, 1994.

[11] K. Lang and E. Baum. Query learning can work poorly when a human oracle is used. In Proceedings of the IEEE International Joint Conference on Neural Networks, pages 335–340. IEEE Press, 1992.

[12] Soodeh Hosseini, Mehrdad Azizi (2019), "The Hybrid Technique for DDoS Detection with Supervised Learning Algorithms", Computer Networks (2019), 25 April 2019.

[13] M. Alkasassbeh, G. Al-Naymat, A. Hassanat, M. Almseidin, "Detecting distributed denial of service attacks using data mining techniques", Int. J. Adv. Comput. Sci. Appl. 7 (1) (2016).

[14] V. Yadegari, A. R. Matinfar, "Detect Web Denial of Service Attacks Using Entropy and Support Vector Machine Algorithm" Electronic and Cyber Defense Magazine, vol. 6, no. 4, 2019. (In Persian)

[15] Raj Kumar Batchu a, Hari Seetha (2021), "A generalized machine learning model for DDoS attacks detection using hybrid feature selection and hyperparameter tuning", Journal of Computer Networks, 2021.

[16] S.V. Jansi Rani a, Iacovos Ioannou b c, Prabagarane Nagaradjane d, Christophoros Christophorou c b, Vasos Vassiliou b c, Sai Charan a, Sai Prakash a, Niel Parekh a, Andreas Pitsillides b e (2023), "Detection of DDoS attacks in D2D communications using machine learning approach", Journal of Computer Communications, 2023.

[17] Antoni Jaszcz, Dawid Połap (2022), "AIMM: Artificial Intelligence Merged Methods for flood DDoS attacks detection", Journal of King Saud University - Computer and Information Sciences, 2022.

[18] Liguo Chena b, Yuedong Zhangb, Qi Zhaob, Guanggang Gengb, ZhiWei Yanb (2018), "Detection of DNS DDoS Attacks with Random Forest Algorithm on Spark", The 2nd International Workshop on Big Data and Networks Technologies,2018.

[19] Michal Motylinski a, Áine MacDermott a, Farkhund Iqbal b, Babar Shah b (2022), "A GPU-based machine learning approach for detection of botnet attacks", Journal of Computers & Security (2022).

[20] Fitriani S. Mandala S. Murti M.A. (2016) "Review of semi-supervised method for Intrusion Detection System" in Multimedia and Broadcasting (APMediaCast) Asia Pacific Conference on, 2016, 36-41.

[21] Berkhin, P. (2006). "A survey of clustering data mining techniques." In: Grouping Multidimensional Data. Springer, pp. 25–71.

[22] "Intrusion Detection Evaluation Dataset (CICIDS2017)." Available on:

https://www.unb.ca/cic/datasets/ids-2017.html

[23] Kirubavathi, G. Anitha, R. (2016). "Botnet detection via mining of traffic flow characteristics." Comput. Electr. Eng. 50, 91–101.

[24] Miller S. and Busby-Earle C. ‘‘The role of machine learning in botnet detection,” in Internet Technology and Secured Transactions (ICITST), 2016 11th International Conference for, 2016, pp. 359–364.

[25] Sharafaldin, I. Lashkari, A.H. Ghorbani, A.A. (2018). "Toward generating a new intrusion detection dataset and intrusion traffic characterization." ICISSP, 108–116.

[26] Scikit Learn (Sklearn) - Machine learning in Python:

URL: https://scikit-learn.org

[27] "Hyperparameter Tuning the Random Forest in Python" https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

[28] Liping Yang, Alan M. MacEachren, Prasenjit Mitra and Teresa Onorati. (2018). " Visually-Enabled Active Deep Learning for (Geo) Text and Image Classification: A Review" International Journal of Geo-Information. (2018).