Journal of Electronical & Cyber Defence
Vol. 9, No. 1, 2021, Serial No. 33

Distributed Denial of Service Attacks Detection in Software Defined Networks

A. Banitalebi-Dehkordi, M. Soltanaghaei*, F. Zamani-Boroujeni

*Department of Computer Engineering, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran

(Received: 15/03/2020, Accepted: 05/08/2020)

ABSTRACT

The software defined network (SDN) is a new computer architecture, where the central controller is applied. These networks rely on software and consequently, their security is exposed to different attacks through different components therein. One type of these attacks, which is the latest threat in computer network realm and the efficiency therein, is called the distributed denial of services (DDoS). An attempt is made to develop an attack- detector, through a combined statistical and machine learning method. In the statistical method, the entropy, based on destination IP and normal distribution in addition to dynamic threshold are applied to detect attacks. Normal distribution is one of the most important distributions in the theory of probability. In this distribution the entropy average and standard deviation are effective in attack detection. As for the learning algorithm, by applying the extracted features from the flows and supervised classification algorithms, the accuracy of attack detection increases in such networks. The applied datasets in this study consist of: ISCX-SlowDDoS2016 ،ISCX-IDS2012, CTU-13 and ISOT. This method outperforms its counterparts with an accuracy of 99.65% and 0.12 false positive rate (FPR) for the UNB-ISCX dataset, and with an accuracy of 99.84% and 0.25 FPR for CTU-13 dataset.

Keywords: Distributed Denial of Service, Software Defined Network,Entropy, Normal Distribution, Classification Algorithm

* Corresponding Author Email: soltan@khuisf.ac.ir

نشریه علمی "پدافند الکترونیکی و سایبری"
سال نهم، شماره ۱، بهار ۱۴۰۰، ص ۵۹-۴۳

علمی – پژوهشی

تشخیص حملات منع سرویس توزیع‌شده در شبکه‌های نرم‌افزارمحور

افسانه بنی‌طالبی دهکردی[۱]، محمدرضا سلطان آقایی[۲*]، فرساد زمانی بروجنی[۳]

۱- دانشجوی دکتری کامپیوتر، ۲ و ۳- استادیار، دانشگاه آزاد اسلامی، واحد اصفهان (خوراسگان)، اصفهان، ایران

(دریافت: ۱۳۹۸/۱۲/۲۵، پذیرش: ۱۳۹۹/۰۵/۱۵)

چکیده

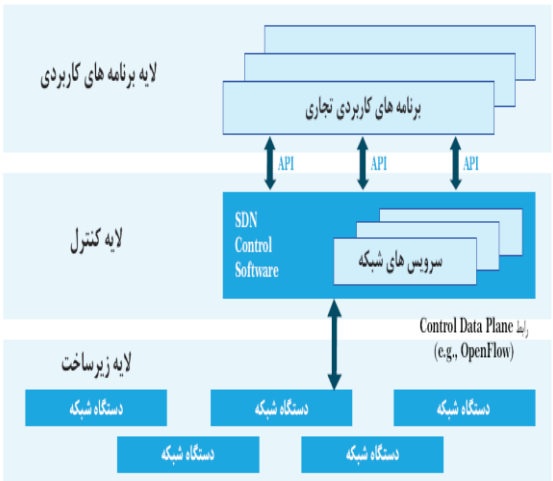
شبکه‌های نرم‌افزارمحور، معماری جدیدی از شبکه‌های کامپیوتری بوده که از هدایت‌کننده مرکزی استفاده می‌کنند. این شبکه‌ها متکی بر نرم‌افزار هستند و از این‌رو، حملات امنیتی گوناگونی می‌تواند از طریق اجزای مختلف شبکه بر ضد آن‌ها صورت گیرد. یکی از این نوع حملات، حمله منع سرویس توزیع‌شده است. این حمله یکی از جدی‌ترین تهدیدات در دنیای شبکه‌های کامپیوتری است و بر روی کارایی شبکه، تاثیرمی‌گذارد. در این پژوهش یک روش تشخیص حملات منع‌سرویس توزیع‌شده به نام «حمله‌یاب» در شبکه‌های نرم‌افزارمحور ارائه شده است. این سامانه مبتنی بر ترکیب روش‌های آماری و یادگیری ماشین است. در روش آماری از آنتروپی مبتنی بر آی پی مقصد و توزیع نرمال با استفاده از حد آستانه انعطاف‌پذیر، برای تشخیص حملات استفاده شده است، توزیع نرمال، یکی از مهم‌ترین توزیع‌های احتمال پیوسته در نظریه احتمالات است. در این توزیع، میانگین آنتروپی و انحراف استاندارد در تشخیص حملات تأثیر دارند. در بخش یادگیری ماشین، با استخراج ویژگی‌های مناسب و استفاده از الگوریتم‌های کلاس‌بندی نظارت‌شده، دقت تشخیص حملات منع‌سرویس توزیع‌شده بالا می‌رود. مجموعه داده‌های مورد استفاده در این پژوهش، ISCX-SlowDDoS2016، ISCX-IDS2012، CTU-13 و ISOT هستند. روش پیشنهادی حمله‌یاب با چند روش دیگر مقایسه شده است که نتیجه مقایسه نشان می‌دهد که روش حمله‌یاب با دقت ۹۹/۶۵ و نرخ هشدار غلط، ۰/۱۲ برای مجموعه داده UNB-ISCX و دقت تشخیص ۹۹/۸۴ و نرخ هشدار غلط ۰/۲۵ برای مجموعه داده CTU-13 دقت و کارایی بالایی نسبت به سایر روش‌های دیگر دارد.

کلیدواژه‌ها: حملات منع‌سرویس توزیع‌شده، شبکه‌های نرم‌افزارمحور، آنتروپی، توزیع نرمال، الگوریتم‌های کلاس‌بندی

۱- مقدمه

شبکه نرم‌افزارمحور[۱]، یک معماری نوین در شبکه‌های کامپیوتری است که کنترل شبکه در آن، از انتقال ترافیک مجزا بوده و به‌طور مستقیم توسط کنترلر برنامه‌ریزی می‌شود. کنترلر با جمع‌آوری اطلاعات آماری از همه سوییچ‌ها و میزبان‌ها، شبکه را به آسانی مدیریت می‌کند [۱].

شکل (۱) نمایی از معماری شبکه‌های نرم‌افزارمحور است. شبکه‌های نرم‌افزارمحور از سه لایه زیرساخت، لایه کنترل و لایه برنامه‌های کاربردی تشکیل شده‌اند. وظیفه هر دستگاه، در لایه زیرساخت، تنها انتقال داده است. هوشمندی در شبکه‌های نرم‌افزارمحور به کنترلر منتقل شده است و مجموعه‌ای از برنامه‌های تجاری در لایه برنامه‌های کاربردی، پشتیبانی می‌شود. شبکه‌های نرم‌افزارمحور در معرض حملات کامپیوتری مختلفی قرار می‌گیرند. حمله منع‌سرویس توزیع‌شده[۲] یکی از حملات رایج در شبکه‌های رایانه‌ای است که سرور را مورد هدف قرار می‌دهید.



شکل (۱): معماری شبکه‌های نرم‌افزارمحور

*رایانامه نویسنده مسئول: soltan@khuisf.ac.ir

[1] Software Defined Network

[2] Disributed Denial Of Service(DDoS)

ناشر: دانشگاه جامع امام حسین (ع)

این حملات با افزایش تأخیر در شبکه و کاهش پهنای باند باعث کاهش کارایی در شبکه می‌شوند [۲]. برای شبکه‌های نرم‌افزارمحور، حمله منع سرویس توزیع‌شده مخرب است چرا که در این شبکه‌ها یک جریان مداوم بین کنترل‌کننده و سوئیچ‌ها وجود دارد [۳]. این حملات روز به روز زیادتر می‌شوند. یکی از حملات منع سرویس توزیع‌شده گسترده در ۲۱ اکتبر سال ۲۰۱۶ رخ داد. شرکت دین[1] یک شرکت مدیریت اینترنت است. یکی از خدمات این شرکت، کنترل و نظارت بر زیرساخت‌های برخط است که مورد حمله منع‌سرویس توزیع‌شده قرار گرفت. این حمله تقریباً یک روز ادامه داشت و اندازه آن به ۱/۲ ترابیت بر ثانیه رسید. تعداد زیادی از کاربران در آمریکای شمالی، اروپا و همچنین چندین وب‌سایت بزرگ از جمله آمازون[2]، فاکس‌نیوز[3] تحت تأثیر این حمله قرار گرفتند [۴].

در حملاتی که در سال‌های مختلف به سایت‌های متعددی صورت گرفت تعداد زیادی از آن‌ها دچار مشکل شدند ازجمله حمله‌ای که به سایت گیت‌هاب که یکی از وب‌سایت‌های اشتراک کد برای برنامه‌نویسان است، شد حجم زیادی از ترافیک آن مورد دسترسی قرار گرفت. این حمله در مدت کوتاهی مشکلات زیادی برای آن سایت ایجاد کرد [۵].

موارد ذکرشده نمونه‌هایی از حملات منع‌سرویس توزیع‌شده بودند. به دلیل اهمیت موضوع امنیت در شبکه‌های نرم‌افزارمحور، تاکنون روش‌های مختلفی برای تشخیص این حملات مطرح‌شده است که اکثر این روش‌ها تکیه بر راه‌حل‌های آماری یا روش‌های یادگیری ماشین[4] دارند. در بعضی از مراجع که روش‌های آماری برای تشخیص حمله مورد استفاده قرارگرفته‌اند، از روش آنتروپی برای تشخیص استفاده شده است [۶]. اگرچه آنتروپی یک روش تشخیص مناسب است اما به تنهایی قادر به تشخیص بعضی از حملات نیست و ممکن است در زمان اوج هشدار اشتباه حمله بدهد، از این‌رو بهتر است با روش‌های دیگر ترکیب شود. زمان اوج، زمانی است که ترافیک‌های قانونی در شبکه به دلایلی مانند خدمت‌رسانی مجاز به تعداد زیادی از کاربران افزایش یافته است. در این زمان، حمله‌ای رخ نداده است. در مراجعی که از روش‌های مربوط به گراف استفاده شده است، اگر تعداد پیام‌های نامطمئن زیاد شود، جریان‌های نرمال به‌عنوان مخرب شناسایی می‌شوند [۷]. در این روش، اگر هم‌بندی شبکه تغییر کند یک سری از آموزش‌های قبلی مختل می‌شود.

در این پژوهش، روش حمله‌یاب[5] با دقت بالا برای تشخیص حملات منع‌سرویس توزیع‌شده در شبکه‌های نرم‌افزار محور ارائه شده است. حملات مورد بررسی در این پژوهش، حملات منع سرویس توزیع‌شده لایه کاربردی هستند. این حملات به دو حالت حمله سطح بالا[6] و حمله سطح پایین[7] تقسیم می‌شوند. حملات سطح بالا با درخواست‌های حجم بالای لایه کاربردی، به قربانی حمله می‌کنند [۸]، حملات سطح پایین حجم ترافیک ورودی کمی دارند و می‌توانند توسط حمله‌کنندگان فریبنده و خبره ایجاد شوند [۹].

هدف از روش حمله‌یاب، تشخیص حملات سطح بالا، با استفاده از ترکیب روش‌های آماری و روش یادگیری ماشین است. در روش آماری از راه‌حل آنتروپی و حد آستانه انعطاف‌پذیر با به‌روزرسانی مقدار α در کنار توزیع نرمال[8]، برای تشخیص حملات استفاده شده است، در روش یادگیری ماشین از الگوریتم‌های کلاس‌بندی برای ساخت مدل یادگیری استفاده می‌شود.

الگوریتم‌های کلاس‌بندی مورد استفاده، الگوریتم‌های J۴۸، RandomTree، RepTree، BayesNet و Logistic هستند. نتایج کار بر روی مجموعه داده‌های ISCX-SlowDDoS2016[9]، ISCX-IDS2012[10]، CTU-13[11] و ISOT[12] بررسی شده است.

ادامه پژوهش به ترتیب زیرنگارش شده است: در بخش ۲ پژوهش‌های مرتبط در رابطه با تشخیص حملات منع‌سرویس توزیع‌شده و روش‌های آن‌ها مطرح‌شده است. در بخش ۳ روش پیشنهادی حمله‌یاب در این پژوهش معرفی شده است. در بخش ۴ مجموعه داده‌هایی برای ارزیابی روش حمله‌یاب معرفی می‌شوند. در بخش ۵، معیارهای ارزیابی راه‌حل مطرح شده بیان شده‌اند و در بخش ۶ پیاده‌سازی و محیط شبیه‌سازی معرفی می‌شود. نتایج در دو بخش آماری و یادگیری ماشین و زمان اجرای الگوریتم روی مجموعه داده‌های موردنظر در بخش ۷، مورد ارزیابی قرار می‌گیرند، در بخش ۸ مقایسه روش حمله‌یاب

[1] www.dyn.com
[2] https://amazon.com
[3] www.foxnews.com
[4] Machine-Learning Method

[5] Attack-Detector Method
[6] High-Volume DDoS Attack
[7] Low-Volume DDoS Attack
[8] Normal Distribution
[9] http://www.unb.ca/cic/datasets/ids-2017.html
[10] https://www.unb.ca/cic/datasets/ids.html
[11] https://www.stratosphereips.org/datasets-ctu13
[12] https:// www.uvic.ca/engineering/ece/isot/contact/index.php

یک‌بار با روش‌های یادگیری ماشین به‌صورت مستقل و یک‌بار با سایر روش‌های مطرح‌شده در کارهای پژوهشی پیشین انجام شده است. نهایتاً، در بخش ۹ نتیجه‌گیری و پیشنهادهایی برای کارهای آینده ارائه می‌شود.

۲- پژوهش‌های مرتبط

در پژوهش‌های انجام‌شده، روش‌های مختلفی برای تشخیص حملات منع‌سرویس توزیع‌شده معرفی شده است. در این بخش نمونه‌ای از آن‌ها مطرح می‌شود:

واروسیا و همکاران در [۱۰] از روش‌های یادگیری ماشین برای تشخیص حملات استفاده کرده‌اند. در این روش هم از خوشه‌بندی K-Means و هم طبقه‌بندی Naive Bayes برای تشخیص حمله منع‌سرویس توزیع‌شده در شبکه‌های کامپیوتری استفاده شده است. در این روش پس از اینکه داده‌های مشابه از نظر رفتار به کمک روش K-Means، به چندین خوشه گروه‌بندی می‌شوند، سپس کل داده‌ها بر اساس این k خوشه برچسب می‌خورند. در مرحله بعد داده‌های برچسب خورده توسط روش طبقه‌بندی بیز ساده به دو دسته حمله و نرمال دسته‌بندی می‌شوند. مجموعه‌داده مورد استفاده در این پژوهش ISCX IDS 2012 بوده و دقت تشخیص حملات، ۹۸٫۸ و در بعضی موارد به ۹۹ نیز می‌رسد.

در مقاله [۱۱]، وانگ و همکاران در ابتدا به شمارش تعداد بسته‌های جریان در جدول‌های اپن فلو[1] پرداختند، سپس مدلی برای تشخیص حمله سبک‌وزن مبتنی بر آنتروپی پیشنهاد دادند. در این روش، تشخیص حمله به سوییچ لبه، انتقال داده شده است و موجب هوشمندی سوییچ‌ها شده است. پیچیدگی زمانی الگوریتم ارائه‌شده $O(n)$ بوده که n اندازه جدول جریان است. در این پژوهش، روش تشخیص ارائه‌شده موجب تشخیص حملات با دقت بالا و نرخ هشدارمثبت غلط[2] پایین می‌شود.

روش ذکرشده در پژوهش [۱۲] از تحلیل همبستگی برای تشخیص حملات منع‌سرویس توزیع‌شده استفاده‌کرده است. در این روش تشخیص حملات منع سرویس از همبستگی ماشین و رکوردهای ترافیک به‌عنوان هدف استفاده کرده است. در این کار یک روش تحلیلی همبستگی چند متغیره[3] برای تشخیص دقیق سوابق ترافیک شبکه و تبدیل سوابق به تصاویر مربوطه معرفی شده است. از این تصاویر سوابق شبکه به‌عنوان اهداف مشاهده شده برای سامانه تشخیص حمله استفاده می‌شود. این روش که فاصله زمین متحرک[4] نامیده می‌شود، بر اساس اندازه‌گیری فاصله بین دو توزیع احتمال محاسبه می‌شود. در محاسبه نیاز هست تا رکوردها و سوابق ترافیک شبکه که به‌صورت یک بردار ویژگی تک‌بعدی هستند، به یک ماتریس ویژگی‌های دوبعدی تبدیل شوند. از طریق این تبدیل، سابقه ترافیک شبکه به‌عنوان یک تصویر شناخته می‌شود. از این الگوریتم می‌توان برای اندازه‌گیری عدم شباهت بین سوابق ترافیک شبکه استفاده کرد.

ساید و همکاران در پژوهش [۱۳] با بررسی حملات منع‌سرویس توزیع‌شده شناخته‌شده و ناشناخته به این نتیجه رسیدند که دقت روش مربوطه، به میزانی که الگوریتم توسط مجموعه داده آموزش می‌بینند ارتباط دارد. در این روش، سه ساختار هم‌بندی مختلف که هرکدام سه لایه دارند مورد استفاده قرار گرفته است. تعداد گره‌های هر هم‌بندی با یکدیگر متفاوت است. شبکه عصبی مربوط به ICMP[5]، شامل سه ورودی و چهار گره مخفی است. شبکه عصبی مربوط به TCP[6]، شامل پنج ورودی و چهار گره مخفی و شبکه عصبی مربوط به UDP[7]، شامل چهارگره ورودی و سه گره مخفی هستند. لایه خروجی برای هر سه حالت یک گره دارد که از دو حالت حمله یا نرمال تشکیل شده است. روش حمله‌یاب با روش‌های دیگری مانند شبکه عصبی انتشار روبه‌عقب[8]، خی‌دو[9] و ماشین بردار پشتیبان[10] مقایسه شده است. دقت در روش حمله‌یاب، ۹۹٫۶۴ درصد بوده است، در حالی‌که در روش خی دو، ۹۲ و در روش ماشین بردار پشتیبان ۹۶ درصد بوده است.

بررسی و تشخیص حملات منع‌سرویس توزیع‌شده در شبکه‌های نرم‌افزار محور و محاسبات ابری توسط وانگ‌بینگ و همکاران در [۱۴] انجام و تأثیر ترکیب این فناوری بر تشخیص حملات منع‌سرویس توزیع‌شده بررسی شده است. مدل‌های مختلفی از ویژگی‌های مجموعه‌داده، بر اساس مقدار درصدهای مختلفی از آن برای بخش آموزش و بخش آزمون داده‌ها ایجاد شده است. نویسنده معتقد است که برای بالا رفتن کارایی، مدل

[1] Openflow
[2] False Positive Rate(FPR)
[3] Multivariate Correlation

[4] Distance Mover's Distance (EMD)
[5] Internet Control Message Protocol
[6] Transmission Control Protocol
[7] User Datagram Protocol
[8] Back Propagation
[9] Chi-square
[10] Support Vector Machine

تشخیص نیاز به به‌روزرسانی در زمان واقعی دارد. برای سه مدل ارائه‌شده در این پژوهش، در بهترین حالت، دقت، ۸۹٫۳۰ به‌دست‌آمد.

در پژوهش [۱۵] فلاحی و همکاران با بررسی روش‌های مختلف از روش تولید قوانین جریانی خودکار برای سامانه تشخیص حملات منع‌سرویس استفاده کردند. برای پیاده‌سازی کار از دو الگوریتم داده‌کاوی استفاده شده است. روش حمله‌یاب روی مجموعه‌داده ISCX آزمون شده است. دقت تشخیص حملات، برای الگوریتم Ripper مقدار ۹۸٫۳۳ و برای الگوریتم C5.0 مقدار ۹۴٫۵۴ است.

D-Face، روشی برای تشخیص سریع حملات منع سرویس توزیع‌شده است که توسط سانی‌بال و همکاران در [۱۶] ارائه شده است. این سامانه انعطاف‌پذیر و توزیع‌شده بوده و از آنتروپی عمومی برای تشخیص حملات منع‌سرویس استفاده می‌کند. در این روش، مقدار اختلاف آنتروپی مبتنی بر آی.پی منبع بین وضعیت جاری و وضعیت نرمال ترافیک شبکه محاسبه شده است. با مقایسه آن با حد آستانه، ترافیک‌های مختلف شبکه ازجمله حملات منع‌سرویس شناسایی می‌شوند. پیاده‌سازی این سامانه هزینه‌بر است و برای بالابردن امنیت سامانه‌ها استفاده می‌شود.

کومار و همکاران در [۱۷] یک سامانه تشخیص حملات منع سرویس توزیع‌شده بر اساس روش ماشین بردار پشتیبان و الگوریتم رتبه‌بندی تجمعی موازی[1] ارائه داده‌اند. افزایش توان مصرفی برای رسیدن به سرعت بالاتر و گرمای ناشی از توان مصرفی مشکلاتی را در این سامانه به‌وجود می‌آورد. دقت تشخیص حملات در این سامانه بین ۹۲ تا ۹۷ درصد گزارش شده است.

روجالینا و همکاران در [۱۸] یک روش دفاعی و تشخیص‌دهنده حمله مبتنی بر یادگیری عمیق[2] طراحی کردند. این روش برای آموزش مدل یادگیری، از داده‌های تاریخی استفاده می‌کند. در این روش از امکانات شبکه‌های نرم‌افزار محور برای طراحی الگوریتم مربوطه استفاده‌شده است. زیرساخت‌های قدرتمند و هزینه‌بر ازجمله مشکلات مربوط به این روش است. دقت سامانه در تشخیص حملات ۹۸٫۸۸ گزارش شده است.

در این بخش، در جدول (۱) مقایسه‌ای از پژوهش‌های بررسی‌شده مطرح شده است:

جدول (۱): مقایسه پژوهش‌های انجام‌شده

روش‌های تشخیص	مزایا	معایب
روش خوشه‌بندی-k means و طبقه‌بندی Naive Bayes [۱۰]	ارائه یک متد سبک وزن برای تشخیص حملات	دقت پایین تشخیص حملات
تشخیص حمله مبتنی برآنتروپی [۱۱]	تشخیص سریع حملات راحتی محاسبه آنتروپی بارکاری کم آنتروپی	کم رنگ شدن مفهوم شبکه‌های نرم‌افزارمحور انتقال تشخیص حمله به سوییچ به‌جای کنترلر
استفاده از روش بینایی کامپیوتر [۱۲]	دقت بالای تشخیص استفاده از تصویر سوابق بررسی شده به‌عنوان اهداف پیش‌بینی‌شده	عدم تشخیص انواع بسته‌های حمله
روش شبکه عصبی مصنوعی [۱۳]	کارایی بالاتر نسبت به روش خی دو بررسی حملات شناخته و ناشناخته	انتخاب سخت تشخیص دوره زمانی بهینه
تشخیص حملات در محاسبات ابری [۱۴]	بررسی تأثیر ترکیب این فنّاوری روی تشخیص حملات	دقت بالایی برای تشخیص حملات ندارد نیاز به به‌روزرسانی مدل در زمان واقعی.
استفاده از C5.0 Ripper+ [۱۵]	کارایی بهتر نسبت به سامانه‌های مشابه	عدم تشخیص انواع بسته‌های حمله
روش آنتروپی تجمیع شده [۱۶]	تشخیص سریع حملات انعطاف‌پذیر	هزینه زیادی دارد بالابردن امنیت بین ارتباطات
الگوریتم رتبه‌بندی تجمعی موازی و روش ماشین بردار پشتیبان [۱۷]	دقت بین ۹۲ تا ۹۷ درصد دسته‌بندی با دقت بالاتر	افزایش توان مصرفی در سرعت بالا گرمای ایجادشده ناشی از توان مصرفی بالا محدودیت سرعت دسترسی به حافظه
روش یادگیری عمیق [۱۸]	استفاده از امکانات شبکه‌های نرم‌افزار محور برای طراحی ماژول مدافع منع سرویس در کنترلر شبکه‌های نرم‌افزارمحور	نیاز به زیرساخت‌های سخت افزاری قدرتمند هزینه بالا برای پردازش داده‌ها. دقت سامانه ۹۸/۸۸

در این جدول، مطالعات پژوهش‌های مختلف مورد بررسی قرار گرفت. عدم تشخیص انواع حملات منع سرویس توزیع‌شده [۱۰]، کاهش کارایی شبکه نرم‌افزارمحور به خاطر انتقال تشخیص حمله از کنترلر به بخش‌های دیگر [۱۱]، عدم تشخیص انواع بسته‌های حمله [۱۲]، عدم تشخیص دوره‌های زمانی مناسب برای تشخیص حملات [۱۳]، دقت پایین مدل و نیاز به بروز رسانی آن در زمان واقعی [۱۴]، دقت پایین در تشخیص بسته‌های حمله در شبکه [۱۵]، نیاز به روش تشخیصی دیگر در کنار روش آنتروپی برای افزایش دقت تشخیص [۱۶]، افزایش توان مصرفی برای رسیدن به سرعت بالاتر و ایجاد گرمای زیاد و محدودیت در دسترسی به حافظه [۱۷]، استفاده از مجموعه داده‌های قدیمی و نامعتبر،

[1] Parallel Cumulative Ranker Algorithm
[2] Deep Learning

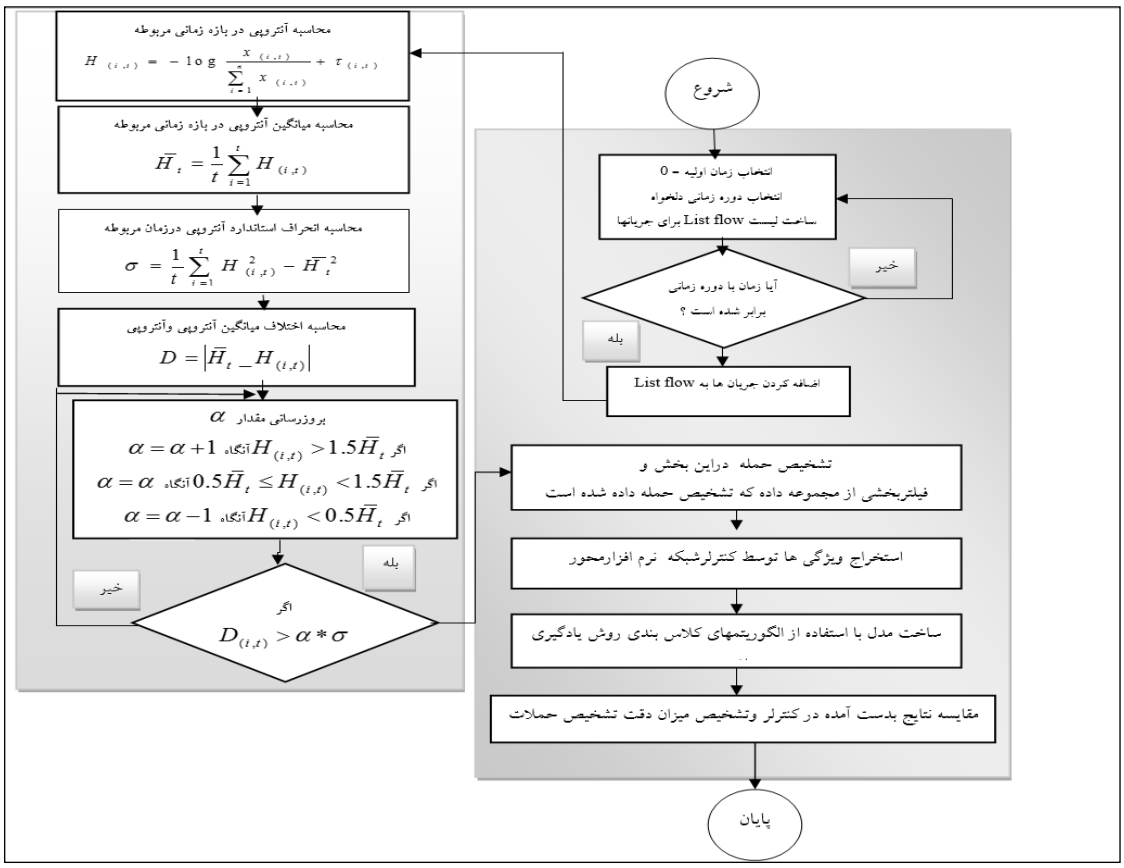
هزینه‌های بالا برای پردازش داده‌ها و نیاز به زیرساخت‌های قدرتمند [۱۸] از جمله مشکلاتی بود که در سایر پژوهش‌ها به‌چشم می‌خورد. در این پژوهش، روش حمله‌یاب در شبکه‌های نرم‌افزارمحور معرفی شده است. در این روش سعی شده است که مشکلات قبلی برطرف شوند. بررسی حالت‌های مختلف حملات منع سرویس توزیع‌شده، یافتن بهترین دوره زمانی بین دوره‌های زمانی مختلف، استفاده از روش یادگیری ماشین در کنار روش آنتروپی، دقت تشخیص بالا، استفاده از مجموعه داده‌های معتبر، عدم نیاز به هزینه زیاد برای زیرساخت سامانه تشخیص ازجمله مواردی است که در این روش ارائه شده است. در بخش ۸، نتایج روش حمله‌یاب با سایر روش‌های پیشین مقایسه شده است. نتایج ارزیابی نشان می‌دهد که روش حمله‌یاب دقت بالایی دارد.

۳- روش پیشنهادی حمله‌یاب

در روش حمله‌یاب برای اولین بار از ترکیب روش‌های آماری و یادگیری ماشین با استفاده از الگوریتم‌های کلاس‌بندی نظارت‌شده استفاده شده است. این روش، از دو کلاس جریان حمله و نرمال تشکیل شده است و به تشخیص ذکرشده می‌پردازد. چارچوب پیشنهادی حمله‌یاب، در شکل (۲) نشان داده شده است:

شکل (۲): چارچوب پیشنهادی حمله‌یاب

این چارچوب از چهار مرحله جمع‌آوری جریان، روش آماری، روش یادگیری ماشین و تشخیص حمله تشکیل شده است. جزییات هر مرحله در فلوچارت شکل (۳) بیان شده است.

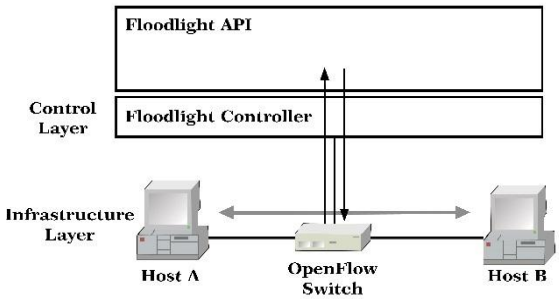


شکل (۳): فلوچارت روش حمله‌یاب

در این شکل، ابتدا در لایه زیرساخت شبکه نرم‌افزار محور، سوییچ‌ها به جمع‌آوری اطلاعات بسته‌ها و جریان‌های واردشده در دوره زمانی مورد نظر می‌پردازند، سپس در لایه کنترلر، با استفاده از اطلاعات جمع‌آوری‌شده، کنترلر مربوطه به تشخیص حملات یا جریان‌های نرمال با استفاده از روش‌های آماری و یادگیری ماشین می‌پردازد. روش‌های ذکرشده در هر بخش در ادامه به تفصیل شرح داده می‌شوند.

۳-۱- جمع‌آوری جریان

برای برقراری یک جریان در شبکه نرم‌افزارمحور، میزبان A یک بسته به میزبان B ارسال می‌کند. این بسته به سوئیچ می‌رسد، سوئیچ جداول خود را به دنبال یافتن قانونی برای این جریان بررسی می‌کند، اگر قانونی وجود نداشت، یک پیام ورود بسته به کنترلر می‌فرستد، کنترلر با توجه به برنامه‌های کاربردی در مورد این بسته تصمیم می‌گیرد، اگر فرض این باشد که تصمیم کنترلر بر ارسال بسته به سمت میزبان B باشد، پاسخ را در قالب یک پیام خروج بسته به سوئیچ می‌فرستد. سوئیچ یک قانون جدید در جدول خود ثبت می‌کند که بسته‌هایی با میزبان A به سمت میزبان B از درگاه خروجی تعیین‌شده ارسال شوند. از این به بعد بسته‌هایی که با این مشخصات به سوئیچ برسد، سوئیچ بدون اینکه بسته را به کنترلر بفرستد طبق قانونی که در جدول دارد عمل می‌کند. این روند تا زمانی که بسته‌ای به سمت سوئیچ بیاید ادامه خواهد داشت، اگر از زمان مشخصی به بعد بسته‌ای به سمت سوئیچ با این مشخصات وارد نشود، سوئیچ این قانون را در جدول خود حذف خواهد کرد. در شکل (۴) نحوه برقراری جریان‌های بین دو میزبان A و B نشان داده شده است.



شکل (۴): نحوه تشکیل جریان بین دو میزبان

در این معماری، کنترلر فلودلایت[1]، برنامه‌های کاربردی کنترلر و سوییچ‌های مجازی دیده می‌شود. هر سوئیچ شامل آمارهای هر جریان مانند مجموع بایت‌های ارسال‌شده، تعداد بسته‌های ارسال شده و همچنین زمان جریان است. برای جمع‌آوری جریان‌ها و استخراج ویژگی‌های مورد نظر، یک برنامه کاربردی در کنترلر ساخته شده و پیام‌های ورود بسته و حذف جریان به آن فرستاده می‌شود. پس از جمع‌آوری اطلاعات، کنترلر روش‌های آماری و یادگیری ماشین را برای تشخیص جریان‌های حمله و نرمال در نظر می‌گیرد.

۳-۲- بخش آماری در روش حمله‌یاب

حملات منع‌سرویس توزیع‌شده با ایجاد اختلال در شبکه، باعث متوقف شدن فعالیت‌های وب شده و سامانه را دچار بی‌نظمی می‌کنند. بی‌نظمی روی پارامترهای مختلفی مانند تعداد بایت‌های هر جریان، تعداد بسته‌های موجود در هر ثانیه، تعداد جریان‌هایی که میزبان مورد نظر گیرنده بوده است و یا روی آی.پی مقصد[2] تعریف می‌شود. در حالت جریان‌های حمله و نرمال مقدار این پارامتر می‌تواند کم و زیاد شده و در تشخیص وضعیت شبکه کمک کند. میزان این بی‌نظمی را می‌توان با محاسبه آنتروپی محاسبه نمود [۱۹]. در پژوهش [۲۰]، جیس و همکاران، از آنتروپی سریع و حد آستانه انعطاف‌پذیر، برای تشخیص حملات منع سرویس توزیع‌شده در شبکه‌های سنتی استفاده کردند. معیار مورد ارزیابی آنتروپی در این کار، بر اساس تعداد جریان‌ها در هر دوره زمانی بود. برای هر اتصال در یک دوره زمانی خاص تعداد کل جریان‌ها محاسبه و آنتروپی مربوط به آن محاسبه شد. در بخش کارهای مربوط به آینده در این پژوهش، ضرورت بررسی روش پیشنهادی بر اساس پارامتر آی.پی ذکر شده است. در پژوهش جاری این موضوع مورد بررسی قرار خواهد گرفت. در این روش، ردیابی آی. پی‌های مقصد در کنترلر فلودلایت بررسی و تأثیر آن در افزایش دقت تشخیص حملات در شبکه‌های نرم‌افزارمحور بررسی شده است. از آنجا که اندازه دوره‌های زمانی و حد آستانه در نظر گرفته شده در تشخیص حملات موثر است. با ارزیابی دوره‌های زمانی مختلف سعی می‌شود تا آنتروپی بهینه برای دوره زمانی مربوطه تعیین شود. در کنترلر شبکه‌های نرم‌افزارمحور، عوض کردن اندازه دوره‌های زمانی به‌صورت نرم‌افزاری آسان است. این انعطاف‌پذیری یکی از ویژگی‌های شبکه‌های نرم‌افزارمحور است. برای محاسبه آنتروپی فرض می‌شود، یک مجموعه از داده W با n عنصر مجزا طبق رابطه (1) وجود دارد. اندازه W، اندازه پنجره دوره زمانی خوانده می‌شود.

$$W = \{X_{(1,t)}, X_{(2,t)}, ..., X_{(n,t)}\} \qquad (۱)$$

[1] Floodlight Controller

[2] Destination IP

در این رابطه، $X_{(i,t)}$، یک رخداد در مجموعه داده W است و تعداد جریان‌های ارتباط i ام را در دوره زمانی t محاسبه می‌کند. آنتروپی جریان i در دوره زمانی t، در رابطه (۲) محاسبه شده است

$$H_{(i,t)} = -\log \frac{x_{(i,t)}}{\sum_{i=1}^{n} x_{(i,t)}} + \tau_{(i,t)} \quad (۲)$$

در این رابطه، $\tau_{(i,t)}$، طبق رابطه (۳) در محاسبه آنتروپی تخمین زده می‌شود.

$$\tau_{(i,t)} = \left| \log \frac{x_{(i,t+1)}}{x_{(i,t)}} \right|, x_{(i,t)} \geq x_{(i,t+1)} \quad (۳)$$

$$\tau_{(i,t)} = \left| \log \frac{x_{(i,t)}}{x_{(i,t+1)}} \right|, x_{(i,t)} < x_{(i,t+1)}$$

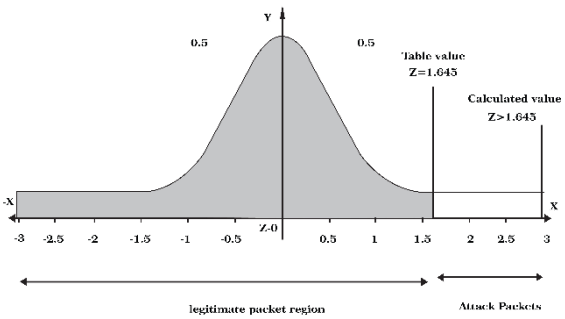
در رابطه (۳)، مقدار خطا در دو حالت متفاوت صعودی و نزولی محاسبه شده است.

پس از محاسبه آنتروپی، میانگین آنتروپی‌های به‌دست‌آمده در هر دوره زمانی طبق رابطه (۴) به‌دست می‌آید.

$$\bar{H}_t = \frac{1}{t} \sum_{i=1}^{t} H_{(i,t)} \quad (۴)$$

در این رابطه، $H_{(i,t)}$، آنتروپی جریان i در دوره زمانی t است.

تحقیقات در [۲۱] نشان می‌دهد که حمله منع‌سرویس توزیع‌شده، از توزیع نرمال تبعیت می‌کند. توزیع نرمال، یکی از مهم‌ترین توزیع‌های احتمال پیوسته در نظریه احتمالات است. در این توزیع، میانگین آنتروپی و انحراف معیار تأثیر دارند. نمودار مربوط به توزیع نرمال در حملات منع سرویس توزیع‌شده در شکل (۵) نشان داده شده است.



شکل (۵): توزیع نرمال در حملات منع سرویس توزیع‌شده [۲۱]

در این شکل وضعیت بسته‌های نرمال و بسته‌های حمله را در نمودار نشان داده است.

در این پژوهش ابتدا در رابطه (۵)، $D_{(i,t)}$ که اختلاف آنتروپی در هر لحظه با مقدار میانگین آنتروپی‌ها محاسبه می‌شود، سپس مقدار آن با حد آستانه Thr، در رابطه (۷) مقایسه می‌شود.

$$D_{(i,t)} = \left| \overline{H}_t - H_{(i,t)} \right| \quad (۵)$$

از آنجا که در حالت حمله طبق شکل (۶)، این پارامتر افزایش می‌یابد، با درنظر گرفتن حد آستانه، بررسی شده و اگر رابطه (۷) برقرار باشد حمله تشخیص داده می‌شود.

$$D_{(i,t)} > Thr \quad (۶)$$

در این رابطه، Thr، حد آستانه انعطاف‌پذیر است که در ادامه محاسبه شده است.

۳-۲-۱- محاسبه حد آستانه

در تشخیص حملات منع‌سرویس توزیع‌شده، مقدار حد آستانه مهم است. برای محاسبه حد آستانه انعطاف‌پذیر از یک روش محاسباتی مبتنی بر ترتیب زمانی استفاده شده است. یکی از اهداف این حد آستانه این است که در پنجره‌های زمانی با عرض کم و سریع به تشخیص حملات منع‌سرویس توزیع‌شده بپردازد. در رابطه (۷) یک حد آستانه منعطف طراحی شده است که بتواند حملات منع‌سرویس توزیع‌شده را تشخیص دهد.

$$Th = \alpha * \sigma \quad (۷)$$

در این رابطه، σ انحراف معیار[1] آنتروپی بوده و α، مقدار تجربی است که متناسب با ترافیک شبکه تغییر می‌کند. انحراف‌معیار طبق رابطه (۸) به‌دست می‌آید.

$$\sigma = \frac{1}{t} \sum_{i=1}^{t} H_{(i,t)}^2 - \overline{H_t^2} \quad (۸)$$

در این رابطه، $H_{(i,t)}$، آنتروپی جریان i در دوره زمانی t و $\bar{H}_t$ میانگین آنتروپی جریان‌ها است.

برای تطبیق‌پذیرکردن حد آستانه با شرایط شبکه‌های کامپیوتری در حالت حمله و نرمال، از یک مقدار تجربی α استفاده می‌شود که متناسب با ترافیک شبکه تغییر می‌کند. برای این مقدار، با تحلیل‌های ارائه شده در [۲۰] سه حالت در نظر گرفته می‌شود:

[1] Standard Deviation

۱) در شرایط حمله حجم بالا، با بررسی جریان‌ها، ازنظر تعداد جریان‌های اضافه‌شده به شبکه بی‌نظمی افزایش می‌یابد. مقدار آنتروپی در این حالت به معیار مربوط به آن وابسته است. آنتروپی در نظر گرفته‌شده در این پژوهش، بر اساس معیار آی.پی مقصد است. در این حالت از آنجا که اکثر جریان‌ها به سمت یک آی.پی خاص که قربانی مورد نظر است حمله می‌کنند لذا در این حالت بی‌نظمی از نظر تعداد آی.پی‌ها کاهش یافته و مقدار آنتروپی بر اساس معیار مورد بررسی در حالت حمله نسبت به حالت قبلی خود کاهش می‌یابد، در این حالت رابطه (۹) برقراراست:

$$H_{(i,t)} > 1.5\bar{H}_t \qquad (۹)$$

حال در حالت حمله سطح بالا، اگر مقدار حد آستانه ثابت باشد خیلی از حالت‌های نرمال به اشتباه حمله تشخیص داده می‌شوند. برای افزایش دقت تشخیص، با افزایش α و کاستن از تعداد اشتباه‌های تشخیصی، مقدار حد آستانه افزایش یافته تا نرخ هشدار مثبت غلط پایین بیاید. در این حالت طبق رابطه (۱۰)، مقدار α یک مقدار افزایش پیدا می‌کند.

$$\alpha = \alpha + 1 \qquad (۱۰)$$

۲) در شرایط نرمال، مقدار آنتروپی در حالت ایستا بوده و مقدار آن طبق رابطه (11)، بین ضریب ۰٫۵ و ۱٫۵ برابر میانگین آنتروپی قرارداد.

$$0.5\bar{H}_t < H_{(i,t)} < 1.5\bar{H}_t \qquad (۱۱)$$

در این حالت، در رابطه (۱۲) به دلیل ثابت بودن آنتروپی، مقدار α در این حالت ثابت می‌ماند.

$$\alpha = \alpha \qquad (۱۲)$$

۳) در شرایط حمله سطح پایین، زمانی که الگوی حمله به‌صورت آرام و مخفی است، در این حالت مقدار آنتروپی طبق رابطه (۱۳) کمتر از میانگین آنتروپی‌ها است،

$$H_{(i,t)} < 0.5\bar{H}_t \qquad (۱۳)$$

در این شرایط، اگر شبکه در حالت پایدار باشد و مهاجم یک حمله سطح پایین به شبکه داشته باشد، مقدار حد آستانه باید کاهش یابد تا حساسیت تشخیص بالاتر رود، از آنجا که بی‌نظمی زیادی در شبکه دیده نمی‌شود مقدار زیاد حد آستانه باعث افزایش مقدار هشدارهای منفی غلط[1] می‌شود. در این حالت، بسیاری از موارد حمله، نرمال تشخیص داده می‌شوند به همین خاطر برای بالاتر بردن دقت تشخیص نیاز به به‌روزرسانی مقدار α بوده و طبق رابطه (۱۴) مقدار آن یک واحد کم می‌شود.

$$\alpha = \alpha - 1 \qquad (۱۴)$$

پس از تعیین مقادیر α، حد آستانه انعطاف‌پذیر به‌روزرسانی شده و با روش مطرح‌شده به تشخیص حملات منع سرویس توزیع‌شده پرداخته می‌شود. در ادامه روابط مطرح‌شده اثبات می‌شوند.

۳-۲-۲- اثبات روابط مطرح‌شده

از آنجا که انحراف استاندارد یکی از شاخص‌های پراکندگی است، نشان می‌دهد که به‌طور میانگین داده‌ها چه مقدار از مقدار میانگین فاصله دارند. در تحلیل‌های آماری، بر اساس منطق انحراف استاندارد، داده‌های با اختلاف بیشتر از دو انحراف استاندارد از مقدار میانگین طبق رابطه (۱۵)، به‌عنوان داده‌های پرت و غیرنرمال شناخته می‌شوند [۲۲].

$$\left|H_{(i,t)} - \bar{H}_{(i,t)}\right| > 2\sigma \qquad (۱۵)$$

در این حالت، $Thr = 2\sigma$ بوده است و به این معنا است که در این شرایط در حالتی که مقدار $\alpha = 2$ باشد می‌توان بین حالت‌های حمله و نرمال تمایز ایجاد کرد، لذا مقدار اولیه در ماژول مورد نظر در کنترلر فلودلایت برای تشخیص حملات و جریان‌های نرمال ۲ در نظر گرفته می‌شود. در رابطه (۱۵) مقدارهای مختلف انحراف استاندارد، شرایط مرزی مختلفی را برای سه حالت حمله سطح بالا، شرایط نرمال و حمله سطح پایین در شبکه فراهم می‌کند. در این پژوهش حالت‌های مختلف مورد بررسی و تحلیل قرارگرفته است. از بین این حالت‌ها در این بخش شرایط وضعیت شبکه در حالت مقدار پیش‌فرض $\sigma = \frac{1}{4}\bar{H}_t$ مورد بررسی قرار می‌گیرد.

الف) اثبات $H_{(i,t)} > 1.5\bar{H}_{(i,t)}$

برای اثبات این رابطه، از رابطه (۱۵) استفاده می‌شود. در این رابطه مقدارهای مختلف انحراف استاندارد شرایط مرزی مختلفی را برای سه حالت شبکه فراهم می‌کند. با قرار دادن و جایگذاری مقدار پیش‌فرض $\sigma = \frac{1}{4}\bar{H}_t$ در رابطه (۱۵)، رابطه (۱۶) اثبات می‌شود.

[1] False Negative Rate(FNR)

$$H_{(i,t)} > \bar{H}_{(i,t)} + 2\sigma \quad (۱۶)$$

$$\sigma = \frac{1}{4}\bar{H}_t$$

$$H_{(i,t)} > 1.5\bar{H}_{(i,t)}$$

ب) اثبات رابطه $0.5\bar{H}_t < H_{(i,t)} < 1.5\bar{H}_t$

ازآنجاکه رابطه (۱۵) شرایط پیش‌بینی جریان‌های حمله را معرفی می‌کند، منطقی است که اگر آن شرایط برقرار نباشد در شرایط نرمال و یا حملات سطح پایین قرارگرفته است و رابطه (۱۷) برقراراست. برای اثبات رابطه (۱۱) از رابطه (۱۷) در دو حالت ۱ و ۲ استفاده می‌شود.

حالت ۱) در این حالت ابتدا نیمه سمت راست نامساوی اثبات می‌شود.

$$\left|H_{(i,t)} - \bar{H}_{(i,t)}\right| < 2\sigma \quad (۱۷)$$

$$-2\sigma < H_{(i,t)} - \bar{H}_{(i,t)} < 2\sigma$$

حال برای بخش سمت راست عبارت زیر بررسی می‌شود:

$$H_{(i,t)} - \bar{H} < 2\sigma \quad (۱۸)$$

با فرض در نظر گرفته‌شده $\sigma = \frac{1}{4}\bar{H}_t$ و جایگذاری آن در رابطه (۱۸)، حالت ۱ طبق رابطه (۱۹) ثابت می‌شود.

$$H_{(i,t)} < 1.5\bar{H}_t \quad (۱۹)$$

حالت ۲) در این حالت ابتدا نیمه سمت چپ نامساوی اثبات می‌شود.

$$-2\sigma < H_{(i,t)} - \bar{H}_{(i,t)} < 2\sigma \quad (۲۰)$$

$$-2\sigma < H_{(i,t)} - \bar{H}_t$$

با فرض در نظر گرفته‌شده $\sigma = \frac{1}{4}\bar{H}_t$ و جایگذاری در رابطه (۲۰)، می‌توان حالت ۲ را طبق رابطه (۲۱) ثابت کرد.

$$0.5\bar{H}_t < H_{(i,t)} \quad (۲۱)$$

از ترکیب دو رابطه (۱۹) و (۲۱) در دو حالت ۱ و ۲، رابطه (۱۱) اثبات می‌شود.

ج) اثبات رابطه $H_{(i,t)} < 0.5\bar{H}_t$

اگر دو حالت روابط (الف) و (ب) برقرار باشند حالت (ج) اثبات می‌شود.

۳-۳- بخش یادگیری ماشین در روش حمله‌یاب

یکی از چالش‌های مهم در روش یادگیری ماشین، یافتن ویژگی‌های مؤثر برای بالابردن دقت نتایج است. در این بخش با توجه به داده‌های ورودی که از مرحله روش آماری دریافت شده است ویژگی‌های موردنظر استخراج می‌شوند. برای این کار از جریان‌هایی که برچسب‌گذاری شده است و گراف در این پژوهش استفاده می‌شود. یک گراف از مجموعه‌ای ناتهی از اشیاء به نام رأس و یال تشکیل شده است. رأس‌ها با V و یال‌ها که رأس‌ها را به هم وصل می‌کنند با E نشان داده شده‌اند. یک گراف با تعداد رأس V و یال‌های E با G=(V,E) نشان داده می‌شود. هر جریان در شبکه نرم‌افزارمحور به‌صورت یک یال در گراف در نظر گرفته می‌شود که میزبان‌های دو سر این جریان رأس‌های گراف هستند.

برای استخراج این ویژگی‌ها، ابتدا هر آی.پی به‌صورت یک رأس در نظر گرفته می‌شود، سپس همه ارتباط‌هایی که آن دو رأس با سایر رأس‌ها دارند برای به دست آوردن ویژگی‌ها استفاده می‌شوند. در انتها با توجه به جریان‌های موجود، یک گراف جهت‌دار وزن‌دار ساخته می‌شود. در این پژوهش، ۹ ویژگی برای هرکدام از میزبان‌های یک جریان استخراج می‌شود. جدول (۲) به معرفی این ویژگی‌ها پرداخته است.

جدول (۲): ویژگی‌های استخراج‌شده

ویژگی‌ها	توضیح
APS[1]	میانگین اندازه بسته‌های هر جریان
FRR[2]	تعداد بسته‌هایی که در هر ثانیه به کنترلر وارد می‌شود
NPF[3]	تعداد جریان‌های دوطرفه
Packet In	اولین بسته‌ای که برای شروع یک جریان می‌فرستد
CountBytes	مجموع بایت‌هایی که برای آن جریان مشخص وجود دارد
CountedSentSource	تعداد جریان‌هایی که میزبان موردنظر فرستنده بوده است
CountedRecieveSource	تعداد جریان‌هایی که میزبان مورد نظر گیرنده بوده است
InputDegree	نسبت تعداد ارتباط‌هایی که میزبان گیرنده بوده است به کل ارتباط‌های گره
OutputDegree	نسبت تعداد ارتباط‌های یک طرفه‌ای که میزبان فرستنده بوده است به تعداد کل ارتباط‌های گره مورد نظر

رکوردهای ایجاد شده در مرحله استخراج ویژگی‌ها در جدول (۲)، به‌عنوان ورودی به الگوریتم‌های کلاس‌بندی داده می‌شود. مدل ساخته‌شده توسط الگوریتم‌های کلاس‌بندی برای تشخیص حملات منع‌سرویس استفاده می‌شود. انواع الگوریتم‌های کلاس‌بندی [۲۳] مانند الگوریتم BayesNet، Randomtree، Logistic Regression، Reptree و J۴۸، برای یادگیری و ساخت

[1] Average Packets Size
[2] Flow Request Rate
[3] Number of Pair-Flows

مدل مورد استفاده قرار می‌گیرند. در شکل (۶) شبه کد مربوط به روش حمله‌یاب ارائه شده است.

Algorithm 1 The DDoS Detection Algorithm

1: Procedure DDoS Detection
2:Input: α **, P, T,** σ
3:Output: Detection-Result
4: Set
$P \leftarrow$ **Sampling Period**
$T \leftarrow$ **Sampling interval**
$Thr \leftarrow$ **Threshold**
$\alpha \leftarrow$ **Threshold Multiplication Factor =2**
$H_t \leftarrow$ **Average Entropy**
$\sigma \leftarrow$ **standard Deviation**
5: while T !==P do
Analyze the network Traffic and extract packet header features and aggregate every T seconds in ListFlow
end while;
6: FlowAdd to ListFlow
7: Compute the Entropy ($H_{(i,t)}$) between network flows using Eq. (2).
8: Calculate the Average Entropy (H_t) for a particular time interval using Eq. (4).
9: Compute the standard deviation (σ) for a particular time interval using Eq. (8).
10: Compute Differences between $H_{(i,t)}$ **and** $1.5\bar{H}_t$ **using Eq. (6).**
11: while $D_{(i,t)} <= Thr$ **do**
12: if $H_{(i,t)} > 1.5\bar{H}_t$ **then**
13: $\alpha = \alpha + 1$
14: $Thr = \alpha * \sigma$
15: end if;
16: if $(0.5\bar{H}_t < H_{(i,t)})$ **and** $(H_{(i,t)} < 1.5\bar{H}_t)$ **then**
17: $\alpha = \alpha$
18: $Thr = \alpha * \sigma$
19: end if;
20: if $H_{(i,t)} < 0.5\bar{H}_t$ **then**
21: $\alpha = \alpha - 1$
22: $Thr = \alpha * \sigma$
23: end if;
24: end while;
25: Detect Result= DDoS attack
26: Filter part of the dataset which detect as attack and TPR =100
27: Extracts 9 features : APS, FRR, NPF, Packet In, Count Bytes , CountSentSrc, CountReceiveSrc, InputDegree, OutputDegree
28: Build training model for each classification algorithm (J48, Random Tree, Logistic Regression, RepTree, Bayes Net)
29: Compute Accuracy, TPR, FPR, F-Measure, Precision
30: end Procedure

شکل (۶): الگوریتم روش حمله‌یاب

پس از اجرای شبه‌کد در کنترلر فلودلایت، جریان‌های حمله و نرمال شناسایی شده و دقت روش پیشنهادی مشخص می‌شود.

۴- مجموعه داده

در این پژوهش از مجموعه داده‌های ISCX-SlowDDoS2016، ISCX-IDS2012، CTU-13 و ISOT استفاده شده است.

در مجموعه داده ISCX-SlowDDos2016 که حاوی حملات منع‌سرویس توزیع‌شده است، از ابزارهای مختلفی برای تولید حملات استفاده شده است. مجموعه داده ISCX-IDS2012 حاوی انواع مختلفی از حملات منع‌سرویس توزیع‌شده است. هر سناریو در این مجموعه داده، حاوی جریان‌های حمله و جریان‌های نرمال است. این مجموعه داده یک فایل ۸۴ گیگابایتی ضبط‌شده از جریان‌های مختلف مربوط به یک هفته است [۲۴]. مجموعه داده CTU-13 که یکی از بزرگ‌ترین مجموعه داده‌های برچسب‌دار موجود است، مجموعه‌ای از ترافیک‌های حمله بوده که در دانشگاه چک در سال ۲۰۱۱ ضبط شده است. این مجموعه داده از ۱۳ نمونه از سناریوهای متفاوت تشکیل شده است. در این پژوهش از سناریو شماره ۱۰ و ۱۱ برای تشخیص حملات منع‌سرویس توزیع‌شده استفاده شده است. مجموعه داده CTU-10 مربوط به حمله منع‌سرویس با ترافیک UDP و مجموعه داده CTU-11 مربوط به حمله منع‌سرویس با ترافیک ICMP است. هر دو حمله از نوع حملات منع‌سرویس سطح بالا هستند [۲۵]. مجموعه داده ISOT توسط مرکز تحقیقاتی فن‌آوری شیء و امنیت اطلاعاتی در دانشگاه ویکتوریا ایجاد شد. در این مجموعه داده، ترافیک‌های نرمال از دو منبع مختلف مرکز تحقیقاتی اریکسون و آزمایشگاه ملی برکلی با یکدیگر ترکیب شده‌اند. این مجموعه داده شامل ۱۴/۱ گیگابایت فایل ضبط شده است. در این پژوهش از بخش نرمال این مجموعه داده در ترکیب با سایر مجموعه داده‌ها استفاده شده است [۲۶].

۵- معیارهای ارزیابی

برای آزمون و ارزیابی روش ارائه‌شده در این پژوهش، از روش K-Fold استفاده شده است. در این روش، همه ورودی‌ها در مجموعه داده آموزشی، هم برای آموزش و هم برای ارزیابی به‌کار می‌روند. در این راه‌حل، مجموعه‌داده آموزشی بر اساس روش k-Fold به‌طور تصادفی به ده زیرنمونه، با حجم یکسان تفکیک می‌شوند. در هر مرحله، نه زیرنمونه به‌عنوان مجموعه داده آموزشی و یک نمونه به‌عنوان مجموعه داده اعتبارسنجی در نظر گرفته می‌شود. تعداد تکرارهای فرآیند برابربا ده خواهد بود و دستیابی به مدل مناسب به سرعت امکان‌پذیر است [۲۷]. معیارهای ارزیابی کارایی در روش حمله‌یاب، صحت[1]، دقت[2]، فراخوانی هشدارمجدد[3]، معیارترکیبی F[4]، نرخ هشدار مثبت درست[5] و نرخ هشدارمثبت غلط هستند که در جدول (۳) روابط مربوط به آن‌ها معرفی شده است [۲۸]. در این جدول، نمونه‌های کلاس نرمال یا منفی، معرف ترافیک عادی است و نمونه‌های کلاس غیرنرمال یا مثبت، معرف ترافیک حمله است.

[1] Accuracy
[2] Precision
[3] Recall
[4] F-Measure
[5] True Positive Rate(TPR)

جدول (۳): پارامترهای تحلیل

پارامتر	رابطه	توضیح
صحت	$\frac{TP+TN}{TP+FN+TN+FP}$	معمول‌ترین مقیاس برای ارزیابی عملکرد مدل‌های پیش‌بینی، دقت پیش‌بینی کننده است که نسبت تعداد کلاس‌های به‌درستی پیش‌بینی‌شده (ازجمله ترافیک نرمال و غیرنرمال)، به تعداد کل کلاس‌ها، تعریف‌شده است
دقت	$\frac{TP}{TP+FP}$	دقت پیش‌بینی به‌صورت نسبت نمونه‌هایی که به‌درستی دسته‌بندی‌شده‌اند به تعداد کل نمونه جریان‌ها در نظر گرفته‌شده توسط یادگیرنده به دست می‌آید
فراخوانی مجدد	$\frac{TP}{TP+FP}$	نسبت نمونه‌هایی که به‌درستی دسته‌بندی‌شده‌اند به تعداد کل نمونه‌های نرمال واقعی
معیارترکیبیF	$\frac{2 \times \Pr ecision \times \mathrm{Re}\, call}{\Pr ecision + \mathrm{Re}\, call}$	یک معیار وزن‌دار که ترکیبی از صحت و فراخوانی مجدد تفسیر شود.
نرخ تشخیص صحیح دسته مثبت	$\frac{TP}{FN+TP}$	نرخ نسبت حملاتی که به‌درستی حمله بوده و حمله تشخیص داده شده است
نرخ تشخیص غلط دسته منفی	$\frac{FP}{FP+TN}$	نرخ نسبت نرمال‌هایی که به اشتباه حمله تشخیص داده شده است.

در این جدول، شش معیار ارزیابی برای بررسی نتایج روش حمله‌یاب ارائه شده است. در این جدول، TN[1]، بیانگر تعداد رکوردهایی است که دسته واقعی آن‌ها منفی بوده و الگوریتم دسته‌بندی نیز دسته آن‌ها را به‌درستی منفی تشخیص داده است.

TP[2]، بیانگر تعداد رکوردهایی است که دسته واقعی آن‌ها مثبت بوده و الگوریتم دسته‌بندی نیز دسته آن‌ها را به‌درستی مثبت تشخیص داده است.

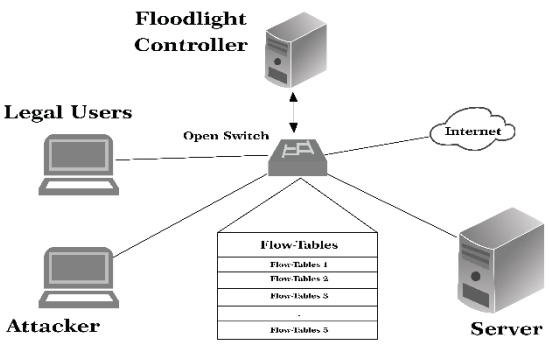
FP[3]، بیانگر تعداد رکوردهایی است که دسته واقعی آن‌ها منفی بوده و الگوریتم دسته‌بندی آن‌ها را به اشتباه مثبت تشخیص داده است.

FN[4]، بیانگر تعداد رکوردهایی است که دسته واقعی آن‌ها مثبت بوده و الگوریتم دسته‌بندی، آن‌ها را به اشتباه منفی تشخیص داده است.

با استفاده از این اطلاعات، مقادیر شش پارامتر محاسبه می‌شود و برای ارزیابی روش حمله‌یاب از آن‌ها استفاده می‌شود.

۶- پیاده‌سازی و محیط شبیه‌سازی

برای پیاده‌سازی روش حمله‌یاب از سامانه عامل لینوکس توزیع اوبنتو با استفاده از ماشین مجازی واز امولاتورمینینت [۲۹] و کنترلر فلودلایت [۳۰] استفاده شده است. در محیط نرم‌افزار ایکلیپس، کدهای پیاده‌سازی مربوط به ماژول حمله‌یاب نوشته شده و در کنترلر اجرا می‌شود. نمایش بسته و جریان‌ها با استفاده از نرم‌افزار وایرشارک انجام می‌شود. مشخصات سخت‌افزاری سامانه اجراکننده ماشین‌مجازی، حافظه اصلی ۸ گیگابایت، پردازنده اینتل و سامانه عامل ویندوز ۱۰ هست. در شکل (۷)، نمایی از هم‌بندی شبیه‌سازی حمله در این مقاله در این شبکه دیده می‌شود.



شکل (۷): هم‌بندی شبیه‌سازی

هم‌بندی مربوطه از کنترلر فلودلایت، سوییچ مجازی، وب‌سرور، کاربران قانونی، مهاجم و اتصال اینترنت تشکیل شده است. پروتکل اپن‌فلو، یک پروتکل شبکه‌ای قابل برنامه‌ریزی برای محیط شبکه‌های نرم‌افزارمحور است که برای برقراری ارتباط میان سوئیچ‌های مجازی و کنترلر استفاده می‌شود.

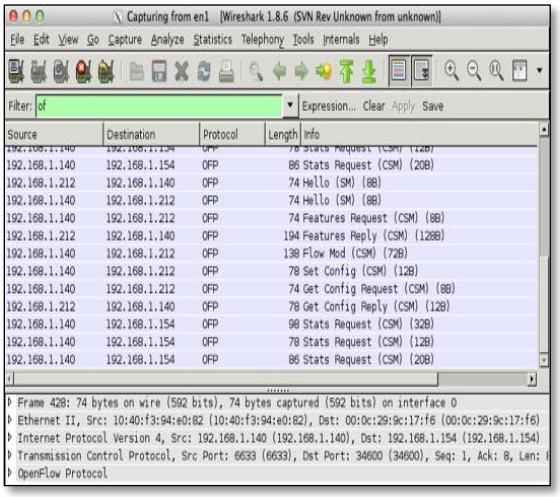
برای شبیه‌سازی حمله، در ابتدا مجموعه داده AppDDoS.pcap که حاوی مجموعه داده ISCX-SlowDDoS2016 هست، از طریق میزبان مهاجم h1 در مینینت با ابزار Tcpreplay به شبکه تزریق می‌شود:

```
Tcpreplay –i h1-eth0 ~/Desktop/AppDDoS.pcap
```

این دستور، یک ابزار متن‌باز[5] بوده و برای ویرایش و به‌کارگیری ترافیک ضبط‌شده در شبکه استفاده می‌شود.

برای تشخیص حملات منع‌سرویس در این هم‌بندی توسط کنترلر ابتدا باید اطلاعات مورد نیاز از سوییچ، میزبان و ارتباطات موجود جمع‌آوری شده تا از آن‌ها برای تشخیص کمک گرفته شود. حملات منع‌سرویس توزیع‌شده با استفاده از ویژگی‌های مختلف ترافیک قابل شناسایی هستند. ویژگی‌های هدر بسته‌ها با استفاده از ابزار وایرشارک استخراج می‌شود. این ویژگی‌ها هر چند ثانیه یک‌بار، جمع‌آوری شده و ویژگی‌های مربوطه با استفاده از این آمار ترافیکی استخراج شده، محاسبه می‌شوند. شکل (۸) نمایی از نرم‌افزار وایرشارک است که به کمک آن اطلاعات شبکه جمع‌آوری شده است.

[1] True Negative
[2] True Positive
[3] False Positive
[4] False Negative
[5] Open Source



شکل (۸): جریان‌های تحلیلی در نرم‌افزار وایرشارک با فیلتر اپن‌فلو

پس از آنکه اطلاعات استخراج‌شده از مجموعه‌داده به کنترلر فرستاده شد، کنترلر فلودلایت با استفاده از ماژول نوشته‌شده حمله یاب به بررسی و تشخیص حملات منع‌سرویس توزیع‌شده می‌پردازد. شکل (۹)، بخشی از این ماژول که در محیط نرم‌افزار ایکلیپس، در کنترلر فلودلایت با استفاده از زبان جاوا نوشته شده است، مشاهده می‌شود.

```
if (Sent.containsKey(ListFlowClass.get(i).ipD))
   APS = Sent.get(ListFlowClass.get(i).ipD).size() / (double)(Sent.get( ListFlowClass.get(i).ipD).size()
else
   APS = 0;
if (Sent.containsKey(ListFlowClass.get(i).ipD))
   NPF = Receive.get(ListFlowClass.get(i).ipD).size() / (double)(Sent.get( ListFlowClass.get(i).ipD).siz
else
   NPF = 1;

if (Sent.containsKey(ListFlowClass.get(i).ipS))
   CountBytes = EntropyFlow(Sent.get(ListFlowClass.get(i).ipS));
else
   CountBytes = -1;

if (Receive.containsKey(ListFlowClass.get(i).ipS))
   CountedSentSource = EntropyFlow(Receive.get(ListFlowClass.get(i).ipS));
else
   CountedSentSource = -1;

if (Sent.containsKey(ListFlowClass.get(i).ipD))
   CountedReceiveSource = EntropyFlow(Sent.get(ListFlowClass.get(i).ipD));
else
   CountedReceiveSource = -1;
```

شکل (۹). بخشی از پیاده‌سازی ماژول حمله یاب برای تشخیص حمله در کنترلر فلودلایت

کد پیاده‌سازی نوشته‌شده در شکل (۹) بخشی از ماژول حمله یاب برای تشخیص حملات در کنترلر فلودلایت را نشان می‌دهد. پس از اینکه، نتایج روش‌های آماری با مشخص کردن حد آستانه انعطاف‌پذیر به‌دست آمد، بخشی از نتایج، به بخش یادگیری ماشین ارسال می‌شود. در این بخش با استفاده از الگوریتم‌های کلاس‌بندی در نرم‌افزار وکا به تشخیص حملات پرداخته می‌شود.

۷- نتایج

در این بخش نتایج روش ترکیبی ارائه شده با هدف شناسایی حملات منع‌سرویس توزیع‌شده ارائه می‌شود. در این پژوهش، دوره‌های زمانی بین ۵ s تا ۲۵۰ s درنظر گرفته شده است. از بین این دوره‌ها، دوره‌های ۵ s تا ۴۵ s به‌عنوان دوره زمانی منتخب برای تشخیص حملات انتخاب شده‌اند. علت این انتخاب این است که در این دوره‌های زمانی، حملات بیشتر، در زمان کمتری تشخیص داده می‌شوند. اگر دوره زمانی بالا در نظر گرفته شود، تشخیص حملات به تأخیر می‌افتد. این موضوع می‌تواند باعث افزایش زمان پاسخ[1] سامانه شده و به کنترلر و سوییچ‌ها آسیب وارد کند. اگر دوره زمانی پایین در نظر گرفته شود، عمل تشخیص حمله سریع آغازشده و منابع زیادی مانند سی‌پی‌یو و پهنای باند شبکه توسط کنترلر از دست داده می‌شود. با توجه به این دو نکته از بین دوره‌های زمانی مختلف دوره‌های زمانی بین s ۱۰ تا ۴۵ s به‌عنوان بهترین دوره‌های زمانی انتخاب می‌شوند. مقدار بهینه α مقداری می‌باشد که دقت تشخیص حمله را افزایش داده و نرخ هشدار غلط پایین داشته باشد.

این مقدار در ماژول حمله یاب در کنترلر فلودلایت به‌دست می‌آید. نتایج محاسبه‌شده در این بخش، برای مجموعه داده‌های مختلف در جداول (۴) تا (۷) بیان شده است.

جدول (۴): نتایج تحلیل روش آماری بر روی دوره‌های زمانی مختلف روی مجموعه‌داده UNB-ISCX2016

دوره‌های زمانی (ثانیه)	تعداد بسته	میانگین منتخب جریان	α	σ	D(i,t)	TPR (%)	FPR (%)
۵	۳۶۹/۰	۳۳/۷۷	۰/۵	۰/۷۴	۱/۶۵	۸۵/۳۲	۱۵/۱۸
۱۰	۷۳۷/۹	۶۷/۵۲	۱	۰/۸۲	۱/۳۹	۹۹/۰۶	۴۱/۱۱
۱۵	۱۱۰۶/۵	۱۰۱/۲۴	۲/۵	۰/۹۳	۱/۳۶	۹۹/۱۳	۳۴/۲۱
۲۰	۱۴۷۵/۰	۱۳۴/۹۳	۱	۰/۶۴	۱/۴۸	۱۰۰	۳۴/۳۶
۲۵	۱۱۴۳/۴	۱۶۸/۵۹	۱	۱/۲۰	۱/۵۴	۱۰۰	۵۲/۶۰
۳۰	۲۲۱۱/۶	۲۰۲/۲۴	۰/۵	۰/۸۴	۱/۳۹	۱۰۰	۸۳/۶۴
۳۵	۲۵۸۱/۴	۲۳۶/۱۳	۲	۰/۵۴	۱/۴۷	۱۰۰	۳۲/۸۲
۴۰	۲۹۴۷/۴	۲۶۹/۴۵	۲	۰/۶۵	۲/۲۹	۱۰۰	۸/۹۰
۴۵	۳۳۱۵/۳	۳۰۳/۰۵	۱	۰/۵۶	۲/۲۲	۱۰۰	۱۷/۶۵

[1] Response Time

جدول (۵): نتایج تحلیل روش آماری بر روی دوره‌های زمانی مختلف روی مجموعه‌داده UNB-ISCX2012

FPR (%)	TPR (%)	D(i,t)	σ	α	میانگین منتخب جریان	تعداد بسته	دوره زمانی (ثانیه)
۷٫۳۴	۵۸٫۳۴	۰٫۹۱	۰٫۷۸	۱	۲٫۰۶	۵۸٫۰۸	۵
۱۳٫۳۴	۷۵٫۷۹	۱٫۴۴	۰٫۹۳	۱٫۵	۲٫۷۵	۷۷٫۴۶	۱۰
۱۴٫۸۲	۱۰۰	۱٫۸۹	۰٫۸۴	۱	۳٫۰۹	۸۷٫۱۵	۱۵
۸٫۷۵	۴۴٫۳۵	۱٫۴۵	۰٫۳۵	۲	۳٫۳۰	۹۲٫۹۷	۲۰
۱۳٫۲۳	۷۵٫۴۲	۱٫۲۲	۰٫۵۹	۲	۳٫۴۴	۹۶٫۸۵	۲۵
۱۵٫۱۱	۹۵٫۶۵	۱٫۸۳	۰٫۹۲	۰٫۵	۳٫۵۳	۹۹٫۶۲	۳۰
۸٫۷۴	۷۲٫۴۵	۲٫۰۴	۱٫۰۵	۰٫۵	۳٫۶۱	۱۰۱٫۷۰	۳۵
۱۳٫۹۲	۷۹٫۸۴	۲٫۰۶	۰٫۸۳	۲	۳٫۶۷	۱۰۳٫۳۱	۴۰
۱۱٫۹۵	۹۹٫۶۴	۲٫۲۲	۰٫۷۳	۱٫۵	۳٫۷۱	۱۰۴٫۶۰	۴۵

جدول (۶): نتایج تحلیل روش آماری بر روی دوره‌های زمانی مختلف روی مجموعه‌داده CTU-10

FPR (%)	TPR (%)	D(i,t)	σ	α	میانگین منتخب جریان	تعداد بسته	دوره زمانی (ثانیه)
۱۶٫۲	۸۸٫۲	۱٫۱۲	۰٫۳۴	۱٫۵	۱۹۰٫۹۵	۳۳۴۹٫۹	۵
۳۰٫۰	۹۰٫۸	۱٫۱۱	۰٫۲۰	۳	۳۷۸٫۳۵	۶۶۳۵٫۸	۱۰
۸۳٫۵	۹۶٫۲	۱٫۷۹	۰٫۴۳	۲	۵۶۵٫۵۵	۹۹۳۰٫۲	۱۵
۱۳٫۸	۹٫۳۲	۱٫۸۵	۰٫۳۲	۳	۷۵۲٫۰۸	۱۳۲۱۱٫۸	۲۰
۳۲٫۴	۹۶٫۴	۱٫۹۸	۰٫۹	۱٫۵	۹۳۹٫۱۵	۱۶۵۰۵٫۱	۲۵
۹۱٫۶	۹۸٫۸	۲٫۱۳	۰٫۴۳	۰٫۵	۱۱۲۴٫۸	۱۹۷۸۲٫۱۳	۳۰
۱۴٫۷	۸۸٫۸	۱٫۷۵	۰٫۸۳	۲	۱۳۱۳٫۲	۲۳۰۴۷٫۳	۳۵
۲۹٫۸	۹۳٫۲	۱٫۸۵	۰٫۳۸	۲٫۵	۱۴۹۹٫۳	۲۶۳۳۳٫۱	۴۰
۹۲٫۶	۹۸٫۵	۲٫۰۱	۰٫۹۳	۱٫۵	۱۶۸۰٫۹	۲۹۶۵۴٫۰	۴۵

جدول (۷): نتایج تحلیل روش آماری بر روی دوره‌های زمانی مختلف روی مجموعه‌داده CTU-11

FPR	TPR	D(i,t)	σ	α	میانگین منتخب جریان	تعداد بسته	دوره زمانی (ثانیه)
۴۶٫۷۲	۱۰۰	۱٫۶۵	۰٫۵۶	۲	۴۳۷۵٫۵	۳۱۷۵۸٫۵	۵
۴۵٫۲۷	۱۰۰	۱٫۳۰	۰٫۸۷	۱٫۵	۸۰۰۱٫۰	۵۸۰۷۴٫۴	۱۰
۷۳٫۰۹	۱۰۰	۱٫۳۶	۰٫۳۴	۳	۹۳۳۷٫۸	۶۷۷۵۷٫۷	۱۵
۵۱٫۱۳	۱۰۰	۱٫۴۸	۰٫۱۲	۱	۹۶۶۱٫۹	۷۰۰۹۸٫۲	۲۰
۵۴٫۰۹	۱۰۰	۱٫۵۴	۰٫۵۶	۲٫۵	۱۰۰۰۵٫۹	۷۲۶۰۱٫۷	۲۵
۶۲٫۴۸	۱۰۰	۱٫۶۵	۰٫۷۲	۲	۹۶۶۵٫۱	۷۰۱۰۴٫۶	۳۰
۴۴٫۳۲	۱۰۰	۱٫۹۶	۰٫۵۳	۳	۱۰۳۷۸٫۰	۷۵۲۹۴٫۶	۳۵
۴۸٫۴۵	۱۰۰	۲٫۰۰	۰٫۶۸	۳	۱۱۶۷۹٫۸	۸۴۷۱۱٫۷	۴۰
۵۲٫۵۴	۱۰۰	۲٫۲۹	۰٫۷۰	۲	۱۳۳۴۳٫۱	۹۶۸۰۷٫۴	۴۵

در حالت کلی ارزیابی‌ها در جداول ذکرشده نشان می‌دهد که برای حد آستانه موردنظر نتایج تشخیص حملات خوب، اما با نرخ هشدار بالا به‌دست آمد. این حالتی است که همه حملات به‌درستی شناسایی شده‌اند اما بخشی از جریان‌های نرمال به اشتباه حمله تشخیص داده شده‌اند. این بخش برای افزایش دقت تشخیص به بخش یادگیری ماشین رفته و بخشی که درست تشخیص داده شده بود برای متوازن‌سازی جریان‌های نرمال و حمله حذف می‌شود. در این مرحله، مجموعه‌داده منتخب ارسالی از بخش آماری با استفاده از روش K-Fold، به ده بخش تقسیم می‌شود. از این ده بخش، نه بخش برای آموزش و یک بخش برای آزمون مدل در نظر گرفته شده و با استفاده از الگوریتم‌های کلاس‌بندی به بررسی هر روش پرداخته شده است. در این ارزیابی، دسته‌بندی جریان‌های ارسال شده در کنترلر انجام شده و تعداد طبقات پیش‌بینی‌شده درست و غلط برای جریان‌های نرمال و حمله برای این حالت به‌دست آمده است. پس از بررسی هر ۱۰ حالت موجود برای هر الگوریتم با جمع‌بندی نتایج، نتیجه کلی برای آموزش و آزمون مجموعه‌داده برای هر الگوریتم به‌دست می‌آید. نتایج کلی در جدول (۸) مشخص شده است.

جدول (۸): نتایج ۱۰ بار عملیات آموزش و آزمون با الگوریتم J48

FPR	TPR	تعداد طبقات پیش‌بینی شده (غلط)	تعداد طبقات پیش‌بینی شده (درست)	خطای مطلق نسبی	ریشه میانگین خطای مطلق	J48
۰٫۰۱	۰٫۹۸	۱۹۶	۵۲۴۶۲۱	۱٫۲۴	۰٫۰۱۲	Sw۰
۰٫۰۳	۰٫۹۹	۹۴۳	۵۵۹۳۳۷	۲٫۳۱	۰٫۰۲۱	Sw۱
۰٫۰۰	۰٫۹۹	۳۶۲	۵۴۰۵۱۱	۱٫۴۰	۰٫۰۳۵	Sw۲
۰٫۰۱	۱٫۰	۲۴۰	۵۲۴۷۵۳	۰٫۹۴	۰٫۰۲۱	Sw۳
۰٫۰۰	۱٫۰	۲۱۶	۵۸۷۷۹۶	۰٫۶۵	۰٫۰۱۸	Sw۴
۰٫۰۲	۰٫۹۸	۲۲۸۷	۵۱۲۷۱	۷٫۸۴	۰٫۰۷۸	Sw۵
۰٫۰۶	۰٫۹۸	۱۰۴۶	۵۳۱۴۶۱	۳٫۲۹	۰٫۰۵۶	Sw۶
۰٫۰۱	۰٫۹۷	۵۹۰	۵۵۰۳۳۵	۲٫۰۹	۰٫۰۱۱	Sw۷
۰٫۰۰	۰٫۹۹	۵۳۶	۵۳۵۶۹۷	۱٫۵۱	۰٫۰۷۰	Sw۸
۰٫۰۰	۱٫۰	۱۴۷	۵۱۱۲۱۵	۰٫۷۱	۰٫۰۴۸	Sw۹

در این جدول، اطلاعات به‌دست‌آمده در هر ده مرحله Sw۰ تا Sw۹ از آموزش و آزمون برای الگوریتم J۴۸ و برای مجموعه‌داده ISCX-SlowDDoS2016 توسط نرم‌افزار وکا محاسبه شده است. اطلاعات آماری مانند ریشه میانگین‌خطای مطلق[1]، خطای مطلق نسبی[2] [۳۱]، تعداد طبقات پیش‌بینی‌شده درست و تعداد طبقات پیش‌بینی‌شده غلط در مورد درخت پیش‌بینی در هر مرحله توسط نرم‌افزار وکا محاسبه می‌شود. این کار برای الگوریتم‌های کلاس‌بندی Bayesnet، Logistic Regression، RandomTree، NaiveBayes و Reptree و برای مجموعه داده‌های ISCX-SlowDDoS2016، ISCX-IDS-2012، CTU-10 و CTU-11 در جدول (۹) نشان داده شده است.

[1] Root mean squared error
[2] Relative absolute error

جدول (۹): نتایج الگوریتم‌های کلاس‌بندی مختلف برای مجموعه داده‌های مختلف

	الگوریتم	TPR	FPR	ACC	Precision	F-Measure
ISCX-SlowDos-2016	BayesNet	۸۶٫۵۰	۱٫۶۷	۹۶٫۹۲	۸۷٫۴۶	۸۶٫۹۷
	J۴۸	۹۷٫۱۵	۱٫۵۰	۹۷٫۷۳	۹۸٫۰۹	۹۶٫۸۸
	Logistic regression	۹۵٫۰۴	۲٫۴۷	۹۶٫۰۲	۹۸٫۳۵	۹۶٫۶۷
	RandomTree	۹۹٫۶۵	۰٫۱۲	۹۹٫۸۲	۹۹٫۶۰	۹۹٫۶۲
	RepTree	۹۳٫۹۹	۴٫۰۰	۹۵٫۶۴	۸۳٫۷۳	۸۸٫۵۶
ISCX-IDS-2012	J۴۸	۹۷٫۴۱	۰٫۶۲	۹۸٫۳۹	۹۹٫۳۸	۹۷٫۸۹
	BayesNet	۹۶٫۹۵	۸٫۲۶	۹۶٫۱۵	۹۸٫۴۷	۹۷٫۷۱
	Logistic regression	۹۹٫۰۸	۱۴٫۴۹	۹۸٫۵۷	۹۹٫۴۲	۹۹٫۲۴
	Naive Bayes	۹۶٫۲۹	۶٫۷۰	۹۴٫۱۴	۸۴٫۶۹	۹۰٫۱۹
	RandomTree	۹۹٫۶۴	۰٫۱۰	۹۹٫۸۳	۹۹٫۶۶	۹۹٫۶۵
	RepTree	۹۷٫۳۵	۳٫۹۶	۹۷٫۳۲	۹۹٫۵۶	۹۸٫۴۵
CTU-10	J۴۸	۹۹٫۱۴	۰٫۲۵	۹۸٫۷۲	۹۹٫۱۳	۹۹٫۱۴
	BayesNet	۹۵٫۵۵	۳٫۹۰	۹۵٫۸۴	۹۵٫۵۳	۹۵٫۵۴
	Logistic regression	۹۶٫۷۰	۰٫۵۴	۹۸٫۸۰	۹۸٫۲۲	۹۷٫۴۶
	Naive Bayes	۹۴٫۴۹	۴٫۳۳	۹۴٫۹۸	۹۶٫۸۷	۹۵٫۶۷
	RandomTree	۹۷٫۴۸	۱٫۸۴	۹۷٫۹۸	۹۴٫۷۷	۹۶٫۱۰
	RepTree	۹۵٫۵۹	۴٫۸۷	۹۵٫۴۷	۹۸٫۲۸	۹۶٫۹۲
CTU-۱۱	J۴۸	۹۸٫۱۰	۰٫۳۴	۹۸٫۸۷	۹۹٫۶۴	۹۸٫۸۶
	BayesNet	۳۸٫۹۱	۱۰٫۳۷	۳۶٫۷۹	۳۹٫۴۲	۵۶٫۱۹
	Logistic regression	۹۹٫۶۲	۱۲٫۰۴	۹۶٫۸۴	۹۶٫۳۵	۹۷٫۹۵
	Naive Bayes	۹۸٫۰۶	۱۵٫۱۵	۹۴٫۷۲	۹۵٫۰۳	۹۶٫۵۲
	RandomTree	۹۷٫۹۵	۲٫۳۶	۹۷٫۸۴	۹۸٫۸۰	۹۸٫۳۷
	RepTree	۹۹٫۶۸	۰٫۱۰	۹۹٫۸۴	۹۹٫۶۶	۹۹٫۶۷

نتایج نشان می‌دهد که برای مجموعه‌داده ISCX-Slow DDoS 2016 بهترین الگوریتم برای شناسایی حملات، الگوریتم Randomtree با دقت ۹۹٫۸۲ و مقدار FPR، ۰٫۱۲ است. نتایج ارزیابی مجموعه‌داده ISCX-IDS2012 در بخش یادگیری ماشین و الگوریتم‌های کلاس‌بندی نشان می‌دهد که بهترین الگوریتم برای شناسایی حملات، الگوریتم Randomtree با دقت ۹۹٫۸۳ و مقدار FPR، ۰٫۱۰ است. برای مجموعه‌داده CTU-10 بهترین الگوریتم برای شناسایی حملات، الگوریتم J۴۸ با دقت ۹۸٫۷۲ و مقدار FPR، ۰٫۲۵ است. برای مجموعه‌داده CTU-11 بهترین الگوریتم برای شناسایی حملات، الگوریتم Reptree با دقت ۹۹٫۸۴ و مقدار FPR، ۰٫۱ است.

با توجه به نتایج به‌دست‌آمده، می‌توان نتیجه گرفت که برای این مجموعه داده‌ها، الگوریتم‌های درخت نتایج بهتری داشته‌اند.

۷-۱- محاسبه زمان اجرای الگوریتم‌ها

در این بخش زمان اجرا برای هر الگوریتم در هر مجموعه داده محاسبه می‌شود. نتایج برای مجموعه داده UNB-ISCX و CTU-13 در جدول‌های (۱۰) و (۱۱) نشان داده شده است.

جدول (۱۰): زمان اجرا در مجموعه داده UNB-ISCX

مجموعه داده	الگوریتم کلاس‌بندی	میانگین زمان اجرا (S)
ISCX-SlowDDoS2016	J۴۸	۸۷٫۵
	BayesNet	۸۹٫۶۸
	Logistic regression	۸۹٫۶۴
	RandomTree	۸۴٫۳۳
	RepTree	۸۴٫۳۱
ISCX-IDS2012	J۴۸	۸۲٫۷۱
	BayesNet	۸۸٫۰۹
	Logistic regression	۸۱٫۰۶
	RandomTree	۸۱٫۳۸
	RepTree	۸۰٫۹۲

نتایج جدول (۱۰)، ناشی از محاسبه زمان اجرا برای الگوریتم‌های مختلف کلاس‌بندی در مجموعه داده‌های ISCX-SlowDoS2016 و ISCX-IDS-2012 است.

در میان الگوریتم‌های بررسی شده، الگوریتم RepTree با ۸۴٫۳۱s و الگوریتم RandomTree با ۸۴٫۳۳s سریع‌ترین الگوریتم و الگوریتم BayesNet با ۸۹٫۶۸s کندترین الگوریتم برای مجموعه داده ISCX-SlowDDoS2016 است. همچنین، برای مجموعه داده ISCX-IDS-2012، در میان الگوریتم‌های بررسی شده، الگوریتم RepTree با ۸۰٫۹۲s سریع‌ترین الگوریتم و الگوریتم BayesNet با ۸۸٫۰۹s، کندترین الگوریتم برای مجموعه داده ISCX-IDS-2012 است.

جدول (۱۱): میانگین زمان اجرا در مجموعه داده CTU-13

مجموعه داده	الگوریتم کلاس‌بندی	میانگین زمان اجرا (ثانیه)
CTU-10	J۴۸	۵٫۳۱
	BayesNet	۵٫۵۲
	Logistic regression	۵٫۲۷
	RandomTree	۵٫۳۱
	RepTree	۵٫۳۴
CTU-11	J۴۸	۴٫۴۳
	BayesNet	۴٫۵۸
	Logistic regression	۴٫۵۱
	RandomTree	۴٫۴۴
	RepTree	۴٫۳۷

برای مجموعه داده CTU-10، الگوریتم Logistic با ۵٫۲۷s سریع‌ترین و الگوریتم BayesNet با ۵۲٫۵s کندترین الگوریتم است. برای مجموعه داده CTU-11، الگوریتم Reptree با زمان اجرای ۴٫۳۷s سریع‌ترین و الگوریتم BayesNet با ۴٫۵۸s کندترین الگوریتم است.

مقایسه و ارزیابی نتایج دو جدول (۱۰) و (۱۱) نشان می‌دهد که الگویتم Reptree در سه دیتاست ISCX-SlowDDoS2016،ISCX-IDS-2012 و CTU-10 سریع‌ترین الگوریتم کلاس‌بندی بوده است، کندترین الگوریتم در هر ۴ مجموعه داده، الگوریتم Bayesnet بوده است.

۸- مقایسه روش پیشنهادی با سایر روش‌ها

در این پژوهش، علاوه بر ارائه روش حمله‌یاب، روش یادگیری ماشین به‌صورت مستقل و بدون ترکیب با روش آماری، با استفاده از الگوریتم‌های کلاس‌بندی برای مجموعه داده‌های یکسان بررسی شد. در این روش برای آزمون و ارزیابی روش ارائه‌شده، از روش K-Fold، استفاده شده است. هدف از این کار بررسی روش آماری در این روش حمله‌یاب است. نتایج مربوط به این روش در جدول (۱۲) ارائه شده‌اند.

جدول (۱۲): نتایج الگوریتم‌های کلاس‌بندی برای مجموعه‌داده‌های مختلف در روش یادگیری ماشین

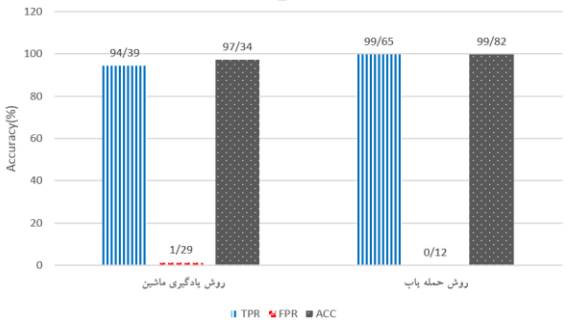
	الگوریتم	TPR	FPR	ACC	Precision	F-Measure
ISCX-SlowDos-2016	BayesNet	۹۳٫۲۵	۴٫۷۴	۹۴٫۸۴	۸۳٫۴۷	۸۸٫۰۹
	J۴۸	۹۴٫۷۰	۳٫۸۶	۹۵٫۵۱	۹۴٫۸۳	۹۴٫۷۶
	Logistic regression	۹۶٫۰۶	۳٫۱۹	۹۶٫۶۸	۸۵٫۱۶	۹۰٫۲۸
	RandomTree	۹۴٫۳۹	۱٫۲۹	۹۷٫۳۴	۹۷٫۱۱	۹۵٫۷۳
	RepTree	۹۳٫۴۹	۱٫۳۲	۹۷٫۶۱	۹۴٫۸۳	۹۴٫۱۶
ISCX-IDS-2012	J۴۸	۹۷٫۲۸	۴٫۴۵	۹۷٫۰۶	۹۹٫۳۵	۹۸٫۳۰
	BayesNet	۸۷٫۲۵	۳٫۰۳	۹۴٫۸۰	۸۹٫۲۰	۸۸٫۲۱
	Logistic regression	۹۵٫۸۴	۵٫۲۹	۹۴٫۹۶	۸۳٫۹۶	۸۹٫۵۱
	Naive Bayes	۸۶٫۷۷	۲٫۹۷	۹۵٫۱۳	۸۶٫۷۹	۸۶٫۷۸
	RandomTree	۹۴٫۶۹	۲٫۶۲	۹۶٫۲۲	۹۶٫۴۴	۹۵٫۵۶
	RepTree	۹۶٫۱۴	۰٫۱۱	۹۸٫۰۱	۹۹٫۸۸	۹۷٫۹۷
CTU-10	J۴۸	۹۶٫۱۷	۳٫۲۷	۹۶٫۵۱	۹۴٫۹۳	۹۵٫۵۵
	BayesNet	۹۴٫۷۳	۲٫۱۰	۹۷٫۰۲	۹۴٫۵۸	۹۴٫۶۵
	Logistic regression	۹۲٫۴۶	۳٫۷۲	۹۴٫۴۲	۹۵٫۹۱	۹۴٫۱۶
	Naive Bayes	۹۲٫۳۲	۲٫۰۲	۹۶٫۲۲	۹۵٫۳۳	۹۳٫۸۰
	RandomTree	۹۵٫۶۱	۲٫۴۷	۹۶٫۷۲	۹۶٫۶۱	۹۶٫۱۱
	RepTree	۹۴٫۹۱	۴٫۳۸	۹۵٫۰۹	۹۸٫۴۳	۹۶٫۶۳
CTU-11	J۴۸	۹۱٫۵۴	۴٫۸۲	۹۲٫۹۰	۹۶٫۹۳	۹۴٫۱۶
	BayesNet	۳۵٫۴۵	۳٫۳۱	۳۶٫۸۲	۹۹٫۷۹	۵۲٫۳۱
	Logistic regression	۹۵٫۶۹	۴٫۰۲	۹۶٫۱۳	۸۶٫۴۸	۹۱٫۳۹
	Naive Bayes	۹۳٫۹۹	۴٫۰۰	۹۵٫۶۴	۸۳٫۷۳	۸۸٫۵۶
	RandomTree	۹۱٫۶۷	۲٫۳۱	۹۴٫۹۷	۹۷٫۰۱	۹۴٫۲۷
	RepTree	۹۸٫۴۸	۳٫۲۵	۹۷٫۵۶	۹۴٫۴۶	۹۷٫۴۶

در این جدول، بهترین نتایج به‌دست‌آمده از روش یادگیری ماشین به‌صورت مستقل با نتایج روش حمله‌یاب در این پژوهش با یکدیگر مقایسه شده‌اند. در جدول (۱۳) مقایسه این دو روش ارائه شده است.

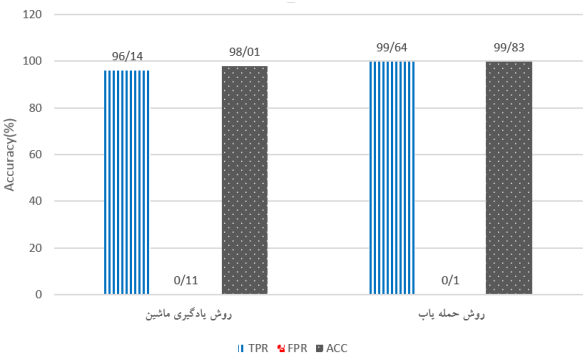
جدول (۱۳): مقایسه نتایج دو روش حمله یاب و یادگیری ماشین

مجموعه‌داده	روش مربوطه	TPR	FPR	ACC
ISCX-SlowDDoS2016	یادگیری ماشین	۹۴٫۳۹	۱٫۲۹	۹۷٫۳۴
	روش حمله‌یاب	۹۹٫۶۵	۰٫۱۲	۹۹٫۸۲
ISCX-IDS2012	یادگیری ماشین	۹۶٫۱۴	۰٫۱۱	۹۸٫۰۱
	روش حمله‌یاب	۹۹٫۶۴	۰٫۱۰	۹۹٫۸۳
CTU-10	یادگیری ماشین	۹۴٫۷۳	۲٫۱۰	۹۷٫۰۲
	روش حمله‌یاب	۹۹٫۱۴	۰٫۲۵	۹۸٫۷۲
CTU-11	یادگیری ماشین	۹۸٫۴۸	۳٫۲۵	۹۷٫۵۶
	روش حمله‌یاب	۹۹٫۶۸	۰٫۱۰	۹۹٫۸۴

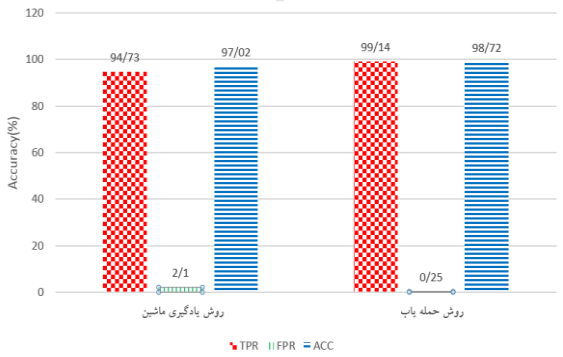
نتایج حاکی از آن است که روش حمله‌یاب در این پژوهش دقت تشخیص بالا و نرخ هشدار پایین‌تری برای تشخیص حملات دارد.



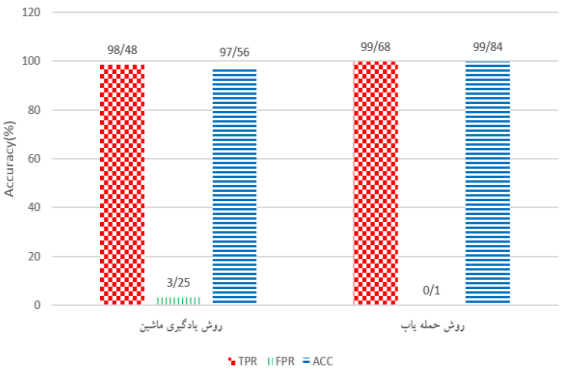
شکل (۱۰): مقایسه روش حمله یاب با روش یادگیری ماشین برای مجموعه‌داده ISCX-SlowDos-2016



شکل (۱۱): مقایسه روش حمله یاب با روش یادگیری ماشین برای مجموعه‌داده ISCX-IDS-2012



شکل (۱۲): مقایسه روش حمله یاب با روش یادگیری ماشین برای مجموعه‌داده CTU-10



شکل (۱۳): مقایسه روش حمله یاب با روش یادگیری ماشین برای مجموعه‌داده CTU-11

نتایج به‌دست‌آمده در شکل‌های (۱۰) تا (۱۳) نشان‌دهنده دقت بالای روش حمله یاب برای تشخیص حملات نسبت به روش یادگیری ماشین به‌صورت مستقل است.

در ادامه این بخش به مقایسه روش حمله‌یاب این پژوهش با پژوهش‌های دیگر پرداخته شده و نتایج آن‌ها با یکدیگر مقایسه شده است. لازم به ذکر است همه این پژوهش‌ها، از مجموعه‌داده UNB ISCX و CTU-13 برای تحلیل روش خود استفاده کرده‌اند. نتایج بررسی در جداول (۱۴) و (۱۵) به‌طور خلاصه بیان شده است.

جدول (۱۴): مقایسه روش حمله‌یاب با سایر روش‌ها برای مجموعه‌داده UNB-ISCX

مرجع	FPR	ACC
[۱۰]	۲۲٪	۹۹٪
[۱۲]	٪۷٫۹۲	۹۰٫۱۲٪
[۱۳]	ذکر نشده است	۹۸٪
[۱۴]	ذکر نشده است	۸۰٫۳۰٪
[۱۵]	۲٪	۹۹٪
روش حمله‌یاب	۰٫۱٪	۹۹٫۶۵٪

جدول (۱۵): مقایسه روش حمله‌یاب با سایر روش‌ها برای مجموعه‌داده CTU-13

مرجع	TPR	ACC
[۳۲]	۹۹٫۱۰٪	۹۸٫۴۰٪
[۳۳]	۸۴٫۴۷٪	۹۸٫۹۳٪
[۳۴]	ذکر نشده است	۹۳٫۶۱٪
روش حمله‌یاب	۹۹٫۶۸٪	۹۹٫۸۴٪

نتایج روش حمله‌یاب این پژوهش بر مجموعه داده‌های UNB-ISCX و CTU-13 در جدول (۱۴) و (۱۵)، با مراجعی که از این مجموعه داده‌ها برای پیاده‌سازی استفاده کرده‌اند، مقایسه شده‌اند. نتایج در هردو مجموعه داده حاکی از بالاتر بودن دقت روش حمله‌یاب است.

۹- نتیجه‌گیری

ترکیب روش‌های آماری و یادگیری ماشین برای تشخیص حملات منع‌سرویس توزیع‌شده در شبکه‌های نرم‌افزار محور و ارائه روش حمله‌یاب، پیشنهاد جدیدی است که در پژوهش‌های قبلی دیده نشده بود. در این پژوهش از روش آنتروپی که یک روش کارا و مناسب برای تشخیص حملات است، در کنار توزیع نرمال، استفاده شد. این روش، بار کاری کمی به CPU وارد می‌کند و به‌راحتی توسط کنترلر در شبکه‌های نرم‌افزارمحور قابل پیاده‌سازی است. در بخش یادگیری ماشین با استخراج ویژگی‌های مناسب و استفاده از روش K-Fold و الگوریتم‌های کلاس‌بندی دقت تشخیص حملات بالاتر می‌رود. روش حمله‌یاب در این پژوهش، با روش یادگیری ماشین به‌صورت مستقل و بدون ترکیب با روش آماری و همچنین با روش‌های سایر پژوهش‌های انجام‌شده در حوزه تشخیص حملات منع سرویس توزیع‌شده، مقایسه شد. نتایج نشان می‌دهد که روش حمله‌یاب در هر دو حالت برای مجموعه داده‌های یکسان، بهتر عمل کرده و دقت تشخیص بالاتری دارد.

کنترلر در شبکه‌های نرم‌افزارمحور برای مدیریت شبکه نیاز به ماژول‌هایی دارد تا بتواند وظایف خود را انجام دهد. عملکرد این ماژول‌ها در زمان تشخیص حملات می‌توانند بارکاری CPU را افزایش دهد. بررسی این موضوع و تأثیر عملکرد ماژول‌ها بر بارکاری CPU می‌تواند به‌عنوان کار آینده برای این تحقیق در نظر گرفته شود.

روشی که در این پژوهش ارائه شد، از یک کنترلر در شبکه‌های نرم‌افزار محور برای تشخیص حملات استفاده می‌کند. در شبکه‌های نرم‌افزارمحور کنترلرهای بیشتری می‌توانند به یکدیگر وصل شده و موجب شناسایی منبع حمله شوند. این روش نیاز به ارتباط بین کنترلرها دارد که هشدار تهدید را برای همه کنترل‌کننده‌ها ارسال می‌کند. افزودن این فرایند ارتباطی به شبکه‌های نرم‌افزارمحور می‌تواند به‌عنوان یک کار آینده برای این پژوهش در نظر گرفته شود.

۱۰- مراجع

[1] J. Cui, M. Wang, Y. Luo, and H. Zhong, "DDoS detection and defense mechanism based on cognitive-inspired computing in SDN," Future generation computer systems, vol. 97, pp. 275-283, 2019.

[2] M. S. Mahmoud and Y. Xia, "Cloud Control Systems: Analysis," Design and Estimation, Academic Press, 2020.

[3] Q. Yan, Q. Gong, and F.-A. Deng, "Detection of DDoS Attacks Against Wireless SDN Controllers Based on the Fuzzy Synthetic Evaluation Decision-making Model," Adhoc & Sensor Wireless Networks, vol. 33, 2016.

[4] S. Hilton, "Dyn analysis summary of friday october 21 attack," Dyn blog https://dyn. com/blog/dyn-analysis-summary-of-friday-october-21-attack, 2016.

[5] L. H. Newman, "Github survived the biggest DDoS attack ever recorded," Wired, vol. 1, 2018.

[6] Y. Dai, J. He, Y. Wu, S. Chen, and P. Shang, "Generalized entropy plane based on permutation entropy and distribution entropy analysis for complex time series," Physica A: Statistical Mechanics and its Applications, vol. 520, pp. 217-231, 2019.

[7] S. Lim, J. Ha, H. Kim, Y. Kim, and S. Yang, "A SDN-oriented DDoS blocking scheme for botnet-based attacks," In 2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN), IEEE, pp. 63-68, 2014.

[8] K. M. Prasad, A. R. M. Reddy, and K. V. Rao, "Anomaly based Real Time Prevention of under rated App-DDOS attacks on web: An experiential metrics based machine learning approach," Indian Journal of Science and Technology, vol. 9, p. 27, 2016.

[9] M. H. Bhuyan, D. Bhattacharyya, and J. K. Kalita, "An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection," Pattern Recognition Letters, vol. 51, pp. 1-7, 2015.

[10] W. Yassin, N. I. Udzir, A. Abdullah, M. T. Abdullah, H. Zulzalil, and Z. Muda, "Signature-Based Anomaly intrusion detection using Integrated data mining classifiers," In 2014 International Symposium on Biometrics and Security Technologies (ISBAST), IEEE, pp. 232-237, 2014.

[11] R. Wang, Z. Jia, and L. Ju, "An entropy-based distributed DDoS detection mechanism in software-defined networking," In 2015 IEEE Trustcom/BigDataSE/ISPA, IEEE, vol. 1, pp. 310-317, 2015.

[12] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, "Detection of denial-of-service attacks based on computer vision techniques," IEEE transactions on computers, vol. 64, no. 9, pp. 2519-2533, 2014.

[13] A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using Artificial Neural Networks," Neurocomputing, vol. 172, pp. 385-393, 2016.

[14] B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, "DDoS attack protection in the era of cloud computing and software-defined networking," Computer Networks, vol. 81, pp. 308-319, 2015.

[15] N. Fallahi, A. Sami, and M. Tajbakhsh, "Automated flow-based rule generation for network intrusion detection systems," In 2016 24th Iranian Conference on Electrical Engineering (ICEE), IEEE, pp. 1948-1953, 2016.

[16] S. Behal, K. Kumar, and M. Sachdeva, "D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events," Journal of Network and Computer Applications, vol. 111, pp. 49-63, 2018.

[17] R. K. Deka, D. K. Bhattacharyya, and J. K. Kalita, "Active learning to detect DDoS attack using ranked features," Computer Communications, vol. 145, pp. 203-222, 2019.

[18] R. M. A. Ujjan, Z. Pervez, K. Dahal, A. K. Bashir, R. Mumtaz, and J. González, "Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN," Future Generation Computer Systems, vol. 111, pp. 763-779, 2020.

[19] V. Yadegari and A. Matinfar, "Detect Web Denial of Service Attacks Using Entropy and Support Vector Machine Algorithm," 2019. (In Persian)

[20] J. David and C. Thomas, "DDoS attack detection using fast entropy approach on flow-based network traffic," Procedia Computer Science, vol. 50, pp. 30-36, 2015.

[21] V. Shyamaladevi and R. Umarani, "Thwarting Distributed Denial of Service Attacks Using Normal Distribution and Weibull Theorem,"

[22] F. E. Harris, "Mathematics for physical science and engineering: symbolic computing applications in Maple and Mathematica," Academic Press, 2014.

[23] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Communications surveys & tutorials, vol. 18, no. 2, pp. 1153-1176, 2015.

[24] H. H. Jazi, H. Gonzalez, N. Stakhanova, and A. A. Ghorbani, "Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling," Computer Networks, vol. 121, pp. 25-36, 2017.

[25] O. Yavanoglu and M. Aydos, "A review on cyber security datasets for machine learning algorithms," In 2017 IEEE International Conference on Big Data (Big Data), IEEE, pp. 2186-2193, 2017.

[26] D. Bhamare, T. Salman, M. Samaka, A. Erbad, and R. Jain, "Feasibility of supervised machine learning for cloud security," In 2016 International Conference on Information Science and Security (ICISS), IEEE, pp. 1-5, 2016.

[27] T.-T. Wong, "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation," Pattern Recognition, vol. 48, no. 9, pp. 2839-2846, 2015.

[28] E. Adi, Z. Baig, and P. Hingston, "Stealthy Denial of Service (DoS) attack modelling and detection for HTTP/2 services," Journal of Network and Computer Applications, vol. 91, pp. 1-13, 2017.

[29] R. L. S. De Oliveira, C. M. Schweitzer, A. A. Shinoda, and L. R. Prete, "Using mininet for emulation and prototyping software-defined networks," In 2014 IEEE Colombian Conference on Communications and Computing (COLCOM), IEEE, pp. 1-6, 2014.

[30] S. Asadollahi and B. Goswami, "Experimenting with scalability of floodlight controller in software defined networks," In 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), IEEE, pp. 288-292, 2017.

[31] V. N. Maiorov and G. M. Crippen, "Significance of root-mean-square deviation in comparing three-dimensional structures of globular proteins," Journal of molecular biology, vol. 235, no. 2, pp. 625-634, 1994.

[32] P. Kalaivani and M. Vijaya, "Mining based detection of botnet traffic in network flow," Int. J. Comput. Sci. Inf. Technol. Secur., vol. 6, pp. 535-540, 2016.

[33] A. Bansal and S. Mahapatra, "A comparative analysis of machine learning techniques for botnet detection," In Proceedings of the 10th International Conference on Security of Information and Networks, pp. 91-98, 2017.

[34] R. Chen, W. Niu, X. Zhang, Z. Zhuo, and F. Lv, "An effective conversation-based botnet detection method," Mathematical Problems in Engineering, vol. 2017, 2017.

Journal of Electronical & Cyber Defence
Vol. 9, No. 1, 2021, Serial No. 33

Distributed Denial of Service Attacks Detection in Software Defined Networks

A. Banitalebi-Dehkordi, M. Soltanaghaei*, F. Zamani-Boroujeni

*Department of Computer Engineering, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran

(Received: 15/03/2020, Accepted: 05/08/2020)

ABSTRACT

The software defined network (SDN) is a new computer architecture, where the central controller is applied. These networks rely on software and consequently, their security is exposed to different attacks through different components therein. One type of these attacks, which is the latest threat in computer network realm and the efficiency therein, is called the distributed denial of services (DDoS). An attempt is made to develop an attack- detector, through a combined statistical and machine learning method. In the statistical method, the entropy, based on destination IP and normal distribution in addition to dynamic threshold are applied to detect attacks. Normal distribution is one of the most important distributions in the theory of probability. In this distribution the entropy average and standard deviation are effective in attack detection. As for the learning algorithm, by applying the extracted features from the flows and supervised classification algorithms, the accuracy of attack detection increases in such networks. The applied datasets in this study consist of: ISCX-SlowDDoS2016 ،ISCX-IDS2012, CTU-13 and ISOT. This method outperforms its counterparts with an accuracy of 99.65% and 0.12 false positive rate (FPR) for the UNB-ISCX dataset, and with an accuracy of 99.84% and 0.25 FPR for CTU-13 dataset.

Keywords: Distributed Denial of Service, Software Defined Network,Entropy, Normal Distribution, Classification Algorithm

* Corresponding Author Email: soltan@khuisf.ac.ir