نشریه علمی «علوم و فناوری‌های پدافند نوین»

سال چهاردهم، شماره ۴، زمستان ۱۴۰۲؛ ص ۷۶-۶۷

علمی - پژوهشی

تحلیل ترافیک شبکه با یادگیری ماشین برای تشخیص سریع‌تر حمله قطع سرویس توزیع یافته

محمد ظهیری[۱]، کیمیا شیرینی[۲]، سینا صمدی قره ورن[۳*]

۱ -دانشجوی کارشناسی ارشد، دانشگاه شهید مدنی آذربایجان۲- دانشجوی دکتری ، دانشگاه تبریز۳-استاد، دانشگاه تبریز-تبریز-ایران

(دریافت: ۱۴۰۲/۰۷/۲۶، بازنگری: ۱۴۰۲/۰۹/۲۰، پذیرش: ۱۴۰۲/۱۰/۱۲، انتشار: ۱۴۰۲/۱۰/۲۶)

DOR: https://dorl.net/dor/20.1001.1.26762935.1402.14.3.1.5

چکیده

با افزایش استفاده از خدمات آنلاین، حملات DDoS به‌عنوان یکی از تهدیدات جدی برای سرویس‌های اینترنتی شناخته شده‌اند. این حملات قادرند به‌سرعت سیستم‌ها و سرویس‌های آنلاین را مختل کنند. در این تحقیق، به بررسی و تحلیل چهار الگوریتم یادگیری ماشین برای شناسایی حملات DDoS پرداخته شده است. برای این منظور، از پایگاه‌داده Intrusion Detection Evaluation Dataset (CIC-IDS2017) استفاده شده است که شامل نمونه‌های ترافیک بات نت است. الگوریتم‌های KNN، RF، Naive Bayes و J48 با استفاده از ویژگی‌های انتخاب‌شده با تابع SelectKBest و کتابخانه scikit-learn آموزش داده شدند. نتایج نشان می‌دهند که الگوریتم‌های RF، KNN و J48 از نظر دقت بسیار به هم نزدیک بوده و عملکرد خوبی در شناسایی ترافیک بات نت و ترافیک معمولی داشته‌اند. الگوریتم RF با F1-score بالاتر نسبت به KNN، دقت بیشتری در شناسایی ترافیک بات نت ارائه داده است. از سوی دیگر، الگوریتم Naive Bayes، با وجود دقت کلی بالا، در شناسایی ترافیک بات نت عملکرد ضعیفی داشته و Precision و Recall آن برای دسته‌بندی بات نت بسیار پایین است. الگوریتم J48 نیز عملکرد نسبتاً خوبی داشته، ولی به دلیل مقدار پایین Recall، بخش قابل‌توجهی از ترافیک حمله به‌اشتباه به‌عنوان ترافیک معمولی شناسایی شده است. این تحقیق تأکید دارد که برای مقابله با حملات DDoS، استفاده از الگوریتم‌های مدرن یادگیری ماشین می‌تواند دقت و سرعت شناسایی را بهبود بخشد. در آینده، آزمایش مدل‌ها در شرایط واقعی و با داده‌های متنوع‌تر، به‌منظور افزایش دقت و قابلیت‌های مدل‌های شناسایی حملات اینترنتی، ضروری است.

کلیدواژه‌ها: یادگیری ماشین، ترافیک شبکه، حمله DDos، درخت تصمیم، نزدیک‌ترین همسایه.

Network traffic analysis with machine learning for faster detection of distributed denial of service attack

M. zahiri , K. shirini , S. Samadi Gharehveran *

Tabriz University

(Received:2023/10/18, Revised: 2023/12/11, Accepted: 2024/01/02, Published: 2024/01/16)

Abstract

With the increasing use of online services, DDoS attacks have been recognized as one of the most serious threats to Internet services. These attacks are able to quickly disrupt online systems and services. In this research, four machine learning algorithms for detecting DDoS attacks have been investigated and analyzed. For this purpose, the Intrusion Detection Evaluation Dataset (CIC-IDS2017) database, which includes botnet traffic samples, has been used. KNN, RF, Naive Bayes, and J48 algorithms were trained using the selected features with the SelectKBest function and the scikit-learn library. The results show that the RF, KNN, and J48 algorithms are very close in terms of accuracy and have performed well in identifying botnet traffic and normal traffic. The RF algorithm with a higher F1-score compared to KNN has provided more accuracy in identifying botnet traffic. On the other hand, the Naive Bayes algorithm, despite its high overall accuracy, has performed poorly in identifying botnet traffic, and its precision and recall are very low for botnet classification. The J48 algorithm has also performed relatively well, but due to the low recall value, a significant part of the attack traffic has been mistakenly identified as normal traffic. This research emphasizes that to deal with DDoS attacks, the use of modern machine learning algorithms can improve the accuracy and speed of identification. In the future, it will be necessary to test the models in real-world conditions with more diverse data in order to increase the accuracy and capabilities of Internet attack detection models.

Keywords: DDos attack, Decision tree, Machine learning, Nearest neighbor, Network traffic.

*Corresponding Author E-mail: s.samadi@tabrizu.ac.ir

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Publisher: Imam Hussein University © Authors

۱. مقدمه

تا قبل از دهه ۱۹۸۰، مفهوم امنیت شبکه به شکل امروزی آن مورد توجه نبود؛ چون بیشتر رایانه‌ها به اینترنت متصل نبودند. با افزایش تعداد رایانه‌های متصل به اینترنت، کمبود امنیت شبکه و مفاهیم آن مورد توجه قرارگرفت. حمله‌های اینترنتی یکی از مهم‌ترین موضوعات در شبکه است. حمله‌های اینترنتی می‌توانند در هر زمان و مکانی در سطح‌ها و انواع مختلفی اتفاق بیفتد. در سال‌های اخیر، پرتکرارترین حمله اینترنتی، حمله DDoS بوده‌است. زیرا بقیه حمله‌ها تقریباً یک راه‌حل مؤثر دارند یا حداقل راحت‌تر قابل رفع هستند. اما حمله DDoS تا الان هیچ راه حل تضمینی برای پیشگیری یا حتی تشخیص نداشته است. ترسناک‌ترین موضوع در مورد DDoS ، حجم آن است. در ۵ سال گذشته شاهد بزرگ‌ترین حمله‌ها در طول تاریخ اینترنت بوده‌ایم که هر کدام از آن‌ها چند گیگابایت ترافیک و بیش از ۴۰ میلیون درخواست در ثانیه را به سرور می‌فرستد. از طرف دیگر اخیراً حمله DDoS در دستگاه‌های IoT و شبکه‌های SDN افزایش چشمگیری داشته است[۱].

تحلیل ترافیک شبکه نقش بسیار حیاتی در مدیریت و امنیت شبکه‌های مدرن دارد. باتوجه‌به افزایش حملات سایبری و پیچیدگی‌های آن‌ها، تحلیل ترافیک شبکه به‌عنوان ابزاری برای شناسایی الگوهای غیرعادی و حملات به کار می‌رود. در مقاله حاضر، تمرکز اصلی بر روی تحلیل ترافیک برای شناسایی حملات DDoSاست. این نوع حملات به دلیل توزیع‌شدگی منابع حمله و حجم بالای ترافیک، به راحتی از سیستم‌های سنتی شناسایی عبور می‌کند. لذا استفاده از روش‌های پیشرفته تحلیل ترافیک، مانند یادگیری ماشین، می‌تواند به شناسایی به‌موقع و کاهش اثرات این حملات کمک کند.

حمله DDoS در مقایسه با نوع اولیه خودش (DoS) یک تفاوت کوچک ولی مهم دارد؛ این حمله منبع مشخصی ندارد و این موضوع کار را سخت می‌کند؛ برای همین دیوارهای آتش[1] و IDSها به‌سختی می‌توانند آن را تشخیص دهند. از طرفی، DDoS انواع مختلفی دارد و در لایه‌های مختلفی از معماری شبکه می‌تواند اتفاق بیفتد. حمله‌های لایه ۷- آسیب‌پذیری اپلیکیشن را هدف قرار می‌دهد - حمله پروتکل SYN Flood و حمله‌های حجمی - پهنای باند شبکه را اشغال می‌کند. متأسفانه ایمن کردن شبکه نسبت به‌تمامی این موارد کار دشواری است[۲].

در بسیاری از موارد، استفاده از سخت‌افزارهای امنیتی مانند Firewall، IPS، IDS و Load Balancer برای حمله‌های کوچک مناسب هستند. این سخت‌افزارها در حمله‌های بزرگ به دلیل مصرف ram و cpu بالا ممکن است از کار بیفتند. گاهی هم هدف حمله، خود این‌ها هستند. در این مقاله، تلاش می‌کنیم تا با استفاده از روش‌های یادگیری ماشین ترافیک شبکه - ترافیک جمع‌آوری شده در طول زمان - را بررسی و تحلیل کنیم تا در نهایت الگویی از ترافیک شبکه پیدا کنیم و با این الگو راه‌حلی برای تشخیص سریع‌تر حمله یا حتی رفع آن پیدا کنیم.

در اواخر ماه می سال ۲۰۲۲، یکی از مشتریان سرویس رایگان Cloudflare مورد حمله DDoS قرارگرفت. این حمله با استفاده از بات‌نت Mantis انجام‌شد. حجم این حمله ۲۶ میلیون درخواست HTTPS در ثانیه گزارش شد که تا آن زمان سابقه نداشت[۳]. چند روز بعد، حمله‌ای ۷۶٪ بزرگ‌تر، برای یکی از مشتریان Google Cloud رخ داد. حجم این حمله ۴۶ میلیون درخواست HTTPS در ثانیه گزارش شد. برای درک بهتر، این عدد معادل آن است که ترافیک ۲۴ساعت از wikipedia را در ده ثانیه جای دهیم[۴]. در فوریه سال ۲۰۲۱ هم صرافی رمزارز EXMO مورد حمله‌ای با ۳۰ گیگابایت ترافیک بر ثانیه قرارگرفت[۵, ۶].

این مقاله به بررسی و تحلیل حملات DDoS با استفاده از روش‌های یادگیری ماشین می‌پردازد. هدف اصلی تحقیق، تحلیل ترافیک شبکه و شناسایی الگوهای مؤثر در شناسایی حملات DDoS است. با استفاده از داده‌های جمع‌آوری‌شده و روش‌های پیشرفته یادگیری ماشین، ما به دنبال ایجاد مدلی برای تشخیص سریع‌تر و مؤثرتر حملات DDoS خواهیم بود. به‌ویژه، تلاش خواهیم کرد تا الگوریتم‌های مختلف یادگیری ماشین را بررسی و مقایسه کنیم تا بهترین راه‌حل‌ها را برای مقابله با این تهدیدات ارائه دهیم.

این مقاله از تکنیک‌های پیشرفته انتخاب ویژگی، از جمله `SelectKBest` با تابع `chi2`، برای شناسایی مؤثرترین ویژگی‌ها در تشخیص حملات DDoS استفاده کرده است. این رویکرد به بهبود دقت و کارایی الگوریتم‌های یادگیری ماشین در این زمینه کمک کرده است. همچنین عملکرد چهار الگوریتم مختلف یادگیری ماشین (KNN، RF، J48، و Naive Bayes) را در تشخیص حملات DDoS به طور جامع مقایسه کرده و با استفاده از معیارهای مختلف از جمله دقت، recall، و F1-score، به شناسایی الگوریتم‌های برتر پرداخته است. برای اولین بار، این تحقیق پیشنهاد می‌دهد که الگوریتم‌های یادگیری ماشین بررسی‌شده در محیط‌های واقعی و عملیاتی تست شوند تا عملکرد آن‌ها در شرایط واقعی ارزیابی شده و نقاط ضعف و قوت آن‌ها شناسایی گردد.

[1] Firewall

۱-۱. حمله DDoS

حمله قطع سرویس[1] زمانی اتفاق می‌افتد که سرویس، دستگاه یا شبکه قابلیت ارائه سرویس به کاربران واقعی خود را نداشته باشد. حمله قطع سرویس توزیع شده زیرمجموعه DoS است و زمانی اتفاق می‌افتد که حمله‌کننده از چندین دستگاه متخلف با IPهای مختلف برای اختلال در ترافیک قربانی استفاده کند[۷].

حمله DDoS انواع مختلفی دارد که از معروف‌ترین آن‌ها می‌توان به HTTP-flood، Smurf، UDP-flood و SIDDoSSYN-flood است. پایه تمام این روش‌ها فرستادن حجم زیادی از ترافیک به شبکه قربانی است [۸]. حمله DDoS ابزارهای مختلفی دارد که تعداد آن‌ها به بیش از ۲۰ مورد می‌رسد. هر کدام از این ابزارها، برای یک یا چند نوع خاص از حمله ساخته شده‌اند. چند مورد از پراستفاده‌ترین آن‌ها به طور خلاصه در جدول (۱) آمده است[۹].

جدول ۱. ابزارهای حمله DoS و DDoS

نام	DoS /DDoS	نوع حمله	سیستم‌عامل	ایجاد بات‌نت
SlowLoris	DoS	http	windows, linux	no
LOIC	DoS, DDoS	tcp, udp, icmp, http	windows, linux, mac, android	yes
Hping	DoS	icmp, udp, tcp	linux, windows	no
Aldi bot-net	DDoS	http, tcp	windows	yes
HOIC	DDoS	http	windows	yes
HULK	DoS, DDoS	http	linux, windows	no

۲-۱. روش‌های تشخیص حمله

تکنیک‌های زیادی برای تشخیص DDoS وجود دارد مانند: Shallow Machine Learning، Machine Learning، Statistical و غیره. از بین تمام تکنیک‌ها، Machine Learning از همه مناسب‌تر است و روش‌های دیگر هر کدام محدودیت‌هایی دارند [۹].

- **روش Statistical:** از محدودیت‌های تکنیک Statistical این بود که بیشتر روش‌های آن مبتنی بر آستانه‌های تعریف شده توسط برنامه‌نویس‌ها و وابسته به اطلاعات قبلی بود. این روش با تغییر روش حمله کارایی آن به‌شدت کاهش پیدا می‌کرد و سیستم پویایی برای تغییر در سطح آستانه‌های حمله نداشت. از طرفی انتخاب ویژگی مناسب برای تحلیل ترافیک در این روش کار دشواری بود و احتیاج به حجم بالایی از ترافیک گذشته داشت [۷].
- **روش Shallow Machine Learning:** مشکل تکنیک Shallow Mechine Learning این است که با مقادیر کم‌ترافیک خوب کار می‌کند؛ اما در حجم‌های بالا با مشکل روبه‌رو می‌شود. مشکل دیگر نیاز به به‌روزرسانی‌های درست متناسب با نوع حمله است [۸]. این روش مسئله را به زیر مسئله‌های کوچک‌تر می‌شکند و در نهایت با حل زیر مسئله‌ها، مسئله نهایی را حل می‌کند. این روش از لحاظ سرعت یادگیری بسیار سریع‌تر از یادگیری ماشین است؛ اما در نهایت روش مناسبی برای حمله‌های امروزی نیست.
- **روش Machine Learning:** دلیل مناسب‌بودن روش‌های Machine Learning این است که می‌توانند از اطلاعات، ویژگی‌های مناسب برای تحلیل را استخراج کنند. در بحث DDoS ترافیک عادی قابل‌تشخیص است؛ اما برچسب‌زدن به ترافیک مشکوک کار سختی است و Machine Learning در این زمینه به ما کمک می‌کند. همچنین نتایج نشان داده که تشخیص حمله‌های خفیف‌تر با Machine Learning بسیار بهتر عمل می‌کند [۸].

ابزارهای زیادی برای ذخیره ترافیک استفاده می‌شوند. مثلاً IDSها که کل ترافیک شبکه از آن‌ها عبور می‌کند. TCPDump یکی دیگر از این ابزارهاست که می‌تواند ترافیک شبکه را به قالب pcap ذخیره کند. ابزار دیگر که برای این کار استفاده می‌شود WireShark است که قابلیت جمع‌آوری ترافیک و نمایش لحظه‌ای اطلاعات ترافیک و جداسازی برخی از ویژگی‌های ترافیک را نیز دارد.

ترافیک خام برای ما اهمیتی ندارد، زیرا نمی‌توان از آن برای مدل‌های Machine Learning استفاده کرد؛ باید اطلاعات مفید ترافیک از آن جدا شود. برای این کار می‌توان از ابزار CICFlowMeter استفاده کرد. این ابزار قابلیت استخراج بیش از ۷۰ ویژگی از ترافیک را دارد. طبقه‌بندی ترافیک شبکه با استفاده ابزارهای یادگیری ماشین توسط ویژگی‌های رایجی مانند متوسط سایز پکت‌ها، بیت ریت، بازه زمانی، نوع پکت، پورت مقصد، حجم آن و غیره انجام می‌شود. از این ویژگی‌ها برای تشخیص نوع ترافیک و در نهایت طبقه‌بندی آن‌ها به دودسته حمله یا معمولی استفاده می‌شود [۱۰]. این ویژگی‌ها برای هر نوع از DDoS متفاوت هستند. در این مقاله ما ویژگی‌های حجم کامل پکت درخواست[2]، حجم کامل پکت پاسخ[3]، آهنگ ارسال پکت‌های هر درخواست بر ثانیه[4]، پورت مقصد[5]، طول هدر پاسخ[6]، آهنگ ارسال

[1] Denial-of-Service (DoS)

[2] Total Length of Fwd Packets
[3] Total Length of Bwd Packets
[4] Flow Packets/s
[5] Destination Port
[6] Bwd Header Length

بیت درخواست بر ثانیه[1] است. در انتخاب ویژگی‌های موردنیاز، علاوه بر استفاده از مقالات پیشین، از هوش مصنوعی ChatGPT نیز کمک گرفته شده است.

تمام اطلاعات به‌دست‌آمده از ترافیک قابل‌استفاده و درست نیستند؛ گاهی حتی دارای اهمیت نیز نیستند. باید اطلاعات به‌دست‌آمده را قبل از به‌کارگیری در یادگیری ماشین پیش‌پردازش کنیم [۷]. یعنی حذف‌کردن مقادیر خالی و بی-اهمیت. ما برای تحلیل میزان تأثیر هر ویژگی انتخاب شده، از ابزار SelectBest با ورودی‌های chi2 و k=4 استفاده کرده‌ایم. تابع chi2 برای مقادیر غیرمنفی کار می‌کند، اما در ترافیک شبکه همه مقادیر مثبت هستند، پس به‌راحتی می‌توان مقادیر منفی را حذف کرد و در نهایت داده‌های متنی را به عددی تبدیل کرد. پیش‌پردازش اطلاعات برای جلوگیری از خطا و افزایش عملکرد الگوریتم ضروری است [۱۱].

بعد از انجام پیش‌پردازش، باید مؤثرترین ویژگی‌های دیتاست خود را پیدا کنیم؛ تعداد زیاد ویژگی باعث سخت‌تر شدن کار با داده، کاهش سرعت آموزش و پیش‌بینی و بیش برازش در مدل آموزشی می‌شود. کاهش ابعاد باعث سرعت بیشتر در حین آموزش و نتایج دقیق‌تر می‌شود. از جمله روش‌های رایج کاهش ابعاد SelectBest با ورودی chi2 در پکیج scikit-learn است. برای نتیجه‌گیری بهتر و دقیق‌تر در Machine Learning باید پارامترهای ورودی مدل در بهترین حالت ممکن باشند. برای این کار می‌توان از روش Hyperparameter Tuning استفاده کرد. برای جستجوی ابر پارامترها الگوریتم‌های متفاوتی وجود دارند؛ مانند GridSearchCV.

۳-۱. الگوریتم‌های یادگیری ماشین

پس از بررسی مقالات و جستجو منابع، از بین تمامی الگوریتم‌های یادگیری ماشین، تعدادی از آن‌ها که برای این کار مناسب‌ترند را انتخاب کردیم؛ بعضی از این الگوریتم‌ها برای آموزش نیاز به داده آموزشی کمتری نسبت به سایر الگوریتم‌ها دارند [۱۱] و برخی دیگر هم قدرت بالایی در دسته‌بندی دارند[۱۲]. چند نمونه از الگوریتم‌های رایج به شرح زیر هستند:

- **Random Forest**: این الگوریتم بر اساس تعداد زیادی از درخت تصمیم کار می‌کند و می‌تواند بر اساس تعداد مشخصی از پارامترها تصمیم‌گیری کند. ساختار درخت شامل والد و فرزند است که ریشه اولین گره آن و برگ‌ها آخرین گره‌ها هستند. برگ‌ها نتیجه تصمیم‌گیری الگوریتم بر اساس شرایط هستند [۱۱]. از ویژگی‌های مثبت Random Forest این است که اطلاعات ورودی نیاز به پیش‌پردازش زیاد ندارند و هم با داده‌های عددی و هم غیرعددی به‌راحتی کار می‌کند. دیگر اینکه این الگوریتم برای داده‌های با ابعاد بالا بسیار مناسب است و می‌تواند هم روابط خطی و هم غیرخطی را کنترل کند. از معایب آن هم این است که برای پایگاه‌داده‌های بزرگ، در قسمت آموزش بسیار سنگین عمل می‌کند و پیداکردن ابر پارامتر را دشوارتر می‌کند.
- **KNN**: این مدل بر اساس نزدیک بودن همسایه‌ها کار می‌کند و مبتنی بر این است که اشیای مشابه نزدیک به هم قرار می‌گیرند. این مدل بیشتر برای تشخیص الگو کاربرد دارد. در این روش، مقدار K باید بادقت انتخاب شود. با تغییرات K مقدار پایداری[2] مدل می‌تواند افزایش پیدا کند؛ اما اگر از مقدار مشخصی بیشتر شود مقدار خطاها هم‌افزایش پیدا می‌کند [۱۱,۱۴]. از مزایای این الگوریتم این است که نیاز به آموزش‌های دوره‌ای ندارد، برای پیاده‌سازی بسیار ساده است و داده آموزشی جدید در هر زمانی می‌تواند به مدل اضافه شود. معایب آن نیز این است که با پایگاه‌داده‌های خیلی بزرگ و با ابعاد بالا به‌خوبی کار نمی‌کند؛ زیرا محاسبه فاصله برای تمامی رکوردها بسیار زمان‌بر است، دیگر اینکه باید ویژگی‌های پایگاه‌داده مورداستفاده حتماً از نوع عددی باشند و از قبل نرمال‌سازی شده و مرتب باشند.
- **Naive Bayesian**: این مدل یک مدل بر پایه احتمال است که بر اساس نظریه Bayes طراحی شده است (قانون بیز). این الگوریتم از یک الگوریتم تشکیل نشده، بلکه مجموعه‌ای از الگوریتم‌ها است که بر پایه نظریه Bayes کار می‌کنند. این الگوریتم احتمال وقوع یک احتمال را به‌گونه‌ای محاسبه می‌کند که انگار اتفاق افتاده است. از مزایای این الگوریتم این است که می‌تواند با داده‌هایی با ویژگی‌های زیاد هم به‌خوبی کار کند [۸،۱۲]. دیگر مزایای آن، راحتی پیاده‌سازی، احتیاج نداشتن به داده آموزشی زیاد است و اینکه سرعت بالایی دارد تاحدی‌که می‌تواند برای پیش‌بینی‌های در لحظه هم استفاده شود. از ضعف‌های آن این است که تمامی ویژگی‌های دیتا را مستقل فرض می‌کند که کارایی این الگوریتم را در دنیای واقعی محدود می‌کند.
- **J48**: این الگوریتم یک الگوریتم دسته‌بندی است که درخت تصمیم‌هایی بر اساس نظریه اطلاعات تولید می‌کند. این الگوریتم توسعه‌یافته الگوریتم ID3 رأس کوینلان است. در این الگوریتم درخت تصمیم‌های تولید شده توسط C4.5 برای دسته‌بندی استفاده می‌شوند. در پیاده‌سازی C4.5، به این

[1] Flow Bytes/s

[2] Stablitiy

الگوریتم ویژگی‌های دیگری مانند محاسبه برای مقادیر گم شده، بازه‌بندی مقادیر پیوسته اضافه شده است.

کارهای مرتبط بسیاری در سال‌های اخیر در زمینه پیاده‌سازی تشخیص DDoS به کمک یادگیری ماشین انجام شده و هر کدام از الگوریتم‌ها، ویژگی‌ها و پایگاه‌داده‌های خاصی استفاده کرده‌اند. اکثر الگوریتم‌های مطرح شده در این مقالات شامل Decision Tree، Naive bayesian ، Random Forest وKNN هستند [۸,۱۴]. در بسیاری از کارهای گذشته همچنان از پیاده‌سازی‌های قدیمی‌تر مانند روش Statistical و Shallow Machine Learning استفاده شده است که همان‌طور که پیش‌ازاین گفته شد، امروزه دیگر روش‌های مؤثری در تشخیص DDoS نیستند. همچنین بخشی از این کارها از مفهوم مشابهت ترافیک استفاده کرده‌اند تا DDoS را تشخیص دهند.

بهال و همکاران [۸] از پایگاه‌داده جمع‌آوری شده توسط Mouhammd Alkasassbe, Ahmad B.A Hassant, Hazi Al-Naymat و Mohammd Almsidin استفاده کردند که شامل چهار نوع حمله DDoS می‌شود اما از لحاظ تعداد رکوردها بسیار کمتر از پایگاه داده مورد استفاده در این مقاله است. Naive Bayesian، RF، MLP و J48 چهار الگوریتم استفاده شده در این تحقیق هستند که نتایج آن‌ها شرح داده شده است. در پایگاه داده این مقاله اشاره‌ای به حمله‌های بات نت نشده است.

سه ویژگی delta time، packet length و protocol باعث تسریع و ساده‌تر کردن بخش‌های preproccesing و feature extraction می‌شود [۱۳]. کاهش تعداد ویژگی‌ها باعث می‌شود مدل آموزشی بتواند سریعتر آموزش ببیند و از طرفی طیف وسیع‌تری از حملات را با آموزش روی پایگاه داده مربوطه تشخیص دهد. الگوریتم‌های مورد استفاده Naive bayesian، KNN وRF هستند. همچنین این بحث مطرح شده که سیستم پیاده‌سازی‌شده باید به گونه‌ای باشد که روی هر سیستمی با هر توان سخت افزاری اجرا شود و یکی از دلایل انتخاب تعداد کم ویژگی هم این موضوع است. در این مقاله از ابزار hping3 برای تولید ترافیک استفاد شده که این احتمال را به وجود می‌آورد که مدل آموزشی با ابر پارامترهای پیدا شده، نتواند روی دیگر ترافیک‌های DDoS که با دیگر ابزارها تولید شده، به خوبی عمل کند.

کرامر [۱۴]، روش جدیدتری برای تشخیص DDoS مطرح شده به گونه‌ای که از چهار الگوریتم Machine Learning به طور همزمان در کنار مدل‌های خودشان را برای تشخیص تولید میکنند و نتیجه نهایی حاصل majority voting از نتیجه به دست آمده از هر چهار مدل است. این نوآوری دقت تشخیص را تا ۲٪ افزایش داده است. اما تعداد ویژگی‌های انتخاب شده برای آموزش این مدل ها بیش از ۲۰ عدد است که امر آموزش مدل‌ها و پیدا کردن ابر پارامتر را کمی دشوار میکند.

داس و همکاران [۱۶]، موضوع حلمه DDoS به وسیله بات‌نت به طور مفصل مورد بحث و ارزیابی قرار گرفته و کمک شایانی در زمینه شناخت این نوع از حمله و درک بهتر آن به ما کرد. در نهایت هم با بررسی دقیق الگوریتم‌های Decision Tree، USML، Naive bayes، SVM و ANN و مقایسه نتایج آن‌ها به انتخاب الگوریتم‌ها در این مقاله کمک کرده است. اسدی و همکاران [۱۷]، روش پیشنهادی از روش یادگیری جمعی رأی‌گیری برای بهبود دقت و صحت در تشخیص حملات انکار سرویس استفاده می‌کند. روش پیشنهادی سه مرحله پیش‌پردازش داده‌ها، مرحله انتخاب و کاهش ویژگی و در نهایت مرحله پیش‌بینی حمله را شامل می‌شود.

جدول ۲. بررسی اجمالی کارهای پیشین

مرجع	سال انتشار	DDoS type	Algorithm – Accuracy%	معایب
[۶]	۲۰۲۰	UDP-Flood Smurf SIDDOS HTTP-Flood	J48 - 98.64% MLP - 98.63% RandomForest – 98.10% NaiveBayes – 96.93%	نبودن حمله بات نت در پایگاه‌داده
[۹]	۲۰۲۰	SYN-Flood ICMP-Flood	NaiveBayes - 97.65% KNN – 99.88% RandomForest – 100%	استفاده از ابزار hping3 برای تولید پایگاه‌داده که احتمال دارد مدل نهایی، بعد از آموزش روی دیگر پایگاه‌داده‌ها، نتواند به‌خوبی عمل کند. استفاده‌کردن از ۳ ویژگی و دقت ۱۰۰٪ برای RandomForest نیازمند بررسی بیشتر است.
[۱۰]	۲۰۱۹	Botnet	SVM – 84.32% DecisionTree – 94.43% NaiveBayes – 74.63% ANN – 63.97% USML – 94.67%	دو مورد از الگوریتم-های انتخابی ما در بین الگوریتم‌های این مقاله وجود نداشت که نیازمند پیاده‌سازی بود.
[۱۱]	۲۰۱۹	Land Smurf Neptune Teardrop Back	MLP – 96.5% SVM – 95.73% KNN – 97.83% J48 – 97.89% ensemble – 99.1%	استفاده از بیش از ۲۰ ویژگی برای آموزش مدل‌ها

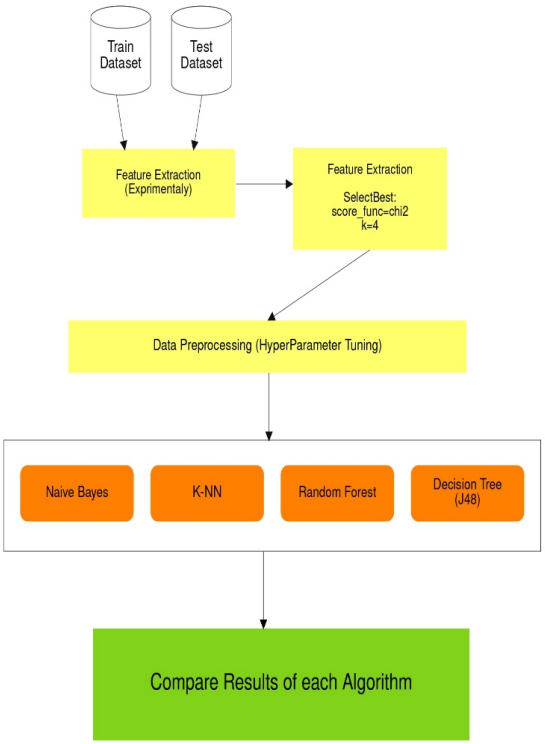
۲. روش تحقیق

روش انجام این تحقیق برای تحلیل ترافیک شبکه، در ابتدا با تقسیم پایگاه‌داده به دو بخش آموزش و آزمایش شروع می‌شود. بعد ویژگی‌های انتخاب شده باتوجه‌به مقالات پیشین و ChatGPT، از پایگاه‌داده جدا می‌شوند. در مرحله بعدی، به کمک کتابخانه Scikit-learn و با تابع SelectBest با score_func=chi2 و k=4 ویژگی‌های مؤثرتر انتخاب می‌شوند و هر مدل با ویژگی-های جدید آموزش داده می‌شود و نتایج آن ذخیره می‌شود. در نهایت، در مرحله بررسی و مقایسه نتایج هر الگوریتم با دیگر الگوریتم‌ها مقایسه می‌شود.

کتابخانه scikit-learn یک کتابخانه متن‌باز نوشته‌شده با Python است. این کتابخانه به ما کمک می‌کند تا به‌راحتی Machine Learning را با کاربردهای Python ترکیب کنیم. Scikit-learn شامل مجموعهٔ گسترده‌ای از روش‌های رگرسیون، دسته‌بندی، ماتریس واریانس، کاهش ابعاد، پیش‌پردازش اطلاعات، آزمایش و مقایسه می‌شود. این کتابخانه در لینک http://scikit-learn.org در دسترس است [۱۳].

به دلیل در دسترس نبودن تجهیزات لازم برای تهیه پایگاه‌داده حمله، از داده‌های آماده در منابع اینترنتی معتبر استفاده‌شده است. پایگاه‌داده مورداستفاده Intrusion Detection Evaluation Dataset (CIC-IDS2017) است. این دیتاست توسط UNB جمع‌آوری شده است. پایگاه‌داده اصلی شامل انواع حمله‌های اینترنتی مانند SQL Injection، SSH-Patator، Port Scan و XSS است. برای این مقاله فقط بخش Bot این پایگاه‌داده مورداستفاده قرار گرفته است. بخش Bot شامل حمله بات نت ARES یکی از بات نت‌های شناخته شده در اینترنت است که دارای قابلیت‌های زیادی از جمله حمله DDoS، گرفتن نماگرفت از صفحه قربانی، اجرای دستور، انتقال فایل از سیستم به سیستم قربانی و تکثیر خودش است. نصب و راه‌اندازی ARES بسیار راحت و سریع است و تقریباً هر کسی با کمترین دانش برنامه‌نویسی می‌تواند این بات نت را اجرا کند که این امر حمله‌های بات نت را بسیار جدی‌تر و خطرناک‌تر از قبل می‌کند.

پایگاه‌داده مورداستفاده از لحاظ کیفیت ترافیک جمع‌آوری‌شده داری رتبه بالایی بین محققان است؛ و برای راحتی کار محققان آینده، سازندگان آن، بخش‌های مفید اطلاعات آن را جدا کرده و تا حد ممکن اطلاعات آن پیش‌پردازش شده است [۷]. این دیتاست در صبح روز جمعه، ۷ جولای سال ۲۰۱۷ از ساعت ۱۰:۰۲ تا ۱۱:۰۲ جمع‌آوری شده و حمله آن توسط BotNet ARES انجام‌شده است. در تقسیم پایگاه‌داده، ۲۰٪ از پایگاه‌داده به بخش آموزش و ۸۰٪ آن برای آزمایش اختصاص داده شده است.

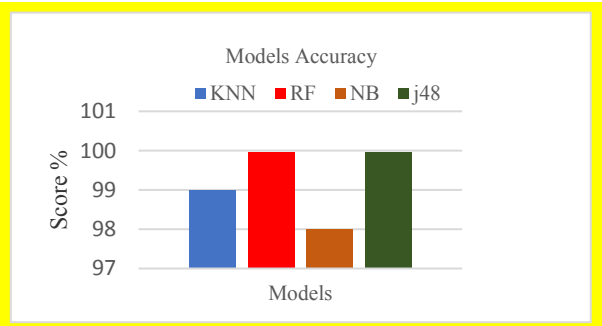


شکل ۱. فلوچارت روش ارائه شده

جدول (۳) ویژگی‌های انتخاب شده به روش تجربی را نشان می‌دهد.

جدول ۳. ویژگی‌های انتخاب شده در روش

ردیف	ویژگی
۱	Total Length of Fwd Packets
۲	Total Length of Bwd Packets
۳	Destination Port
۴	Flow Packets
۵	Bwd Header Length
۶	Flow Bytes



شکل ۲. تفاوت انتخاب درگاه مقصد در دقت مدل‌ها

به کمک تابع SelectBest از کتابخانه scikit-learn و با score_func=chi2 و k=4 ویژگی‌های انتخابی در مرحله تجربی را بررسی کردیم. جدول (۴) نتایج بررسی را نشان می‌دهد.

جدول ۴. موثرترین ویژگی در تشخیص باتنت

ویژگی	امتیاز
Total Length of Fwd Packets	1.31e7
Total Length of Bwd Packets	5.31e7
Destination Port	3.9e4
Flow Packets	2.52e7
Bwd Header Length	4.58e5
Flow Bytes	1.7e9

باتوجه‌به جدول (۴)، ویژگی Backward Bytes/s موثرترین ویژگی در تشخیص بات نت است. در این مرحله چهار ویژگی که بیشترین مقدار تأثیر را داشتند انتخاب شده‌اند. پس از استاندارد کردن مقادیر پایگاه‌داده، این مرحله باتوجه‌به الگوریتم این متفاوت است. برای هر الگوریتم باید پارامترهای مختلفی آزمایش تا در نهایت ابر پارامتر آن پیدا شود. برای پیداکردن ابر پارامتر از کتابخانه scikit-learn و تابع GridSearchCV استفاده شده است. در طول بررسی‌ها برای پیداکردن ابر پارامتر، الگوریتم Random Forest به‌شدت کندتر از دیگر الگوریتم‌ها عمل کرد؛ اما دیگر الگوریتم‌ها نسبتاً سرعت بیشتری داشته‌اند. ابر پارامترهای هر الگوریتم در جدول (۵) آمده است.

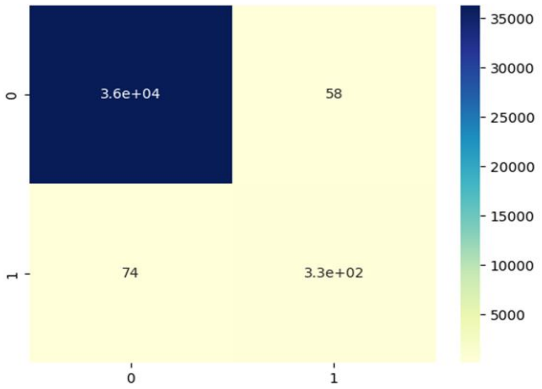
جدول ۵. ابر پارامترهای در نظر گرفته شده در هر روش

الگوریتم	ابر پارامتر
Naive Bayes	var_smoothing: 1.0
K-NN	Metric: manhattan
Random Forest	n_neighbors: 2
Decision Tree (J48)	weights: uniform

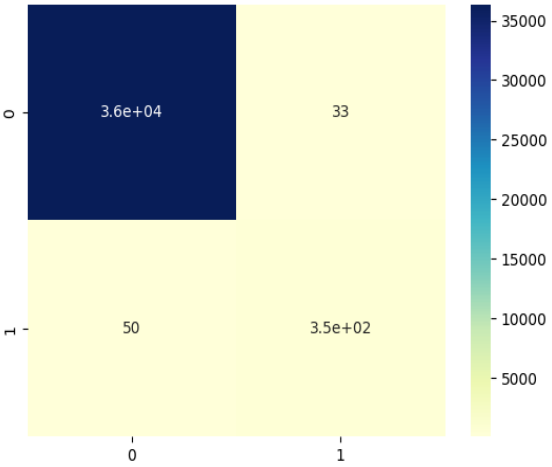
۳. نتایج و بحث

در این مرحله هر مدل با ویژگی‌های انتخاب شده با Select Best آموزش داده می‌شوند. گزارش کامل دقت، ماتریس درهم‌ریختگی، بازیابی،FPR، TPR، ROC Area، F-measure در ادامه آمده است. در جدول‌های گزارش دسته‌بندی، مقدار ۱ یعنی DDoS و مقدار صفر یعنی ترافیک معمولی.

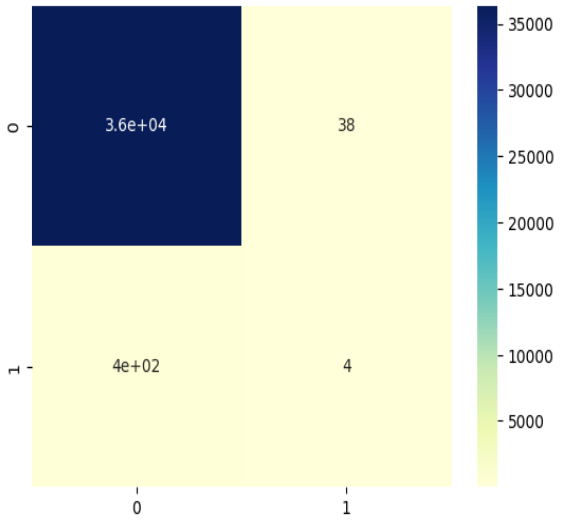
شکل‌های (۳)، (۴)، (۵) و (۶) به ترتیب نشانگر ماتریس درهم‌ریختگی روش‌های ارائه شده می‌باشند. در شکل (۷) نتایج مدل‌ها به صورت جداگانه مورد بررسی قرار گرفته اند. پس از بررسی نتایج به دست آمده از آزمایش‌های انجام شده، سه الگوریتم RF، KNN و J48 از لحاظ دقت بسیار به هم نزدیک بوده و الگوریتم Naive Bayes از همه ضعیف‌تر عمل کرده است. با اینکه دقت این الگوریتم حدود ٪۹۸ بوده ولی دیگر شاخص‌ها مانند بازیابی، دقت و f1-score اختلاف زیادی با دیگر مدل‌ها داشته است. شکل (۵) نشان می‌دهد که خطای مدل KNN در تشخیص ترافیک بسیار پایین بوده و عمل‌کرد بالایی دارد.



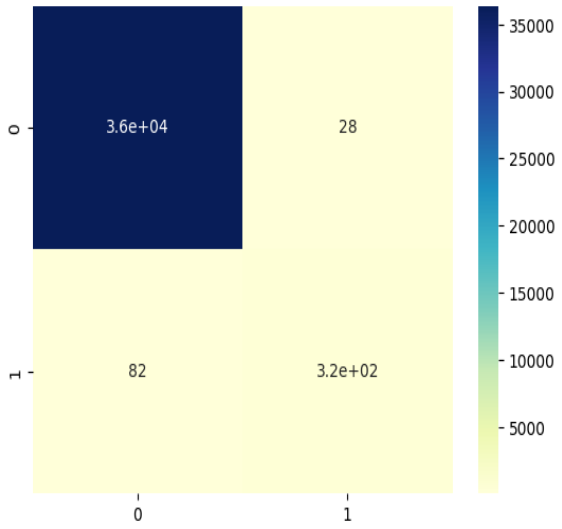
شکل ۳. ماتریس درهم‌ریختگی KNN



شکل ۴. ماتریس درهم‌ریختگی Random Forest



شکل ۵. ماتریس درهم‌ریختگی Naive Bayes



شکل ۶. ماتریس درهم‌ریختگی J48

در جدول (۶)، precision نشان می‌دهد که نسبت پیش‌بینی‌های درست نسبت به تمام پیش‌بینی‌های Bot حدود ۸۵٪ است که دقت بالایی است. Recall نشان میدهد که ۸۲٪ از مقدار ترافیک Bot به درستی دسته‌بندی شده اند. f1-score نزدیک بودن مقادیر recall و precision و f1-score نشان میدهد که مدل هم در تشخیص ترافیک معمولی و هم در تشخیص ترافیک بات نت به خوبی عمل کرده است.

فاصله نسبتاً زیاد بین مقدار accuracy و f1-score کمی نگران‌کننده است. زیرا الگوریتم دقت بالایی در تشخیص دارد؛ اما برای تشخیص ترافیک باتنت دقت کمتری دارد که این می‌تواند در عمل باعث شود که بخشی از ترافیک حمله به سمت سرورها روانه شود.

جدول ۶. عملکرد روش **KNN**

	precision	recall	f1-score	support
Benign	1.00	1.00	1.00	36406
Bot	0.85	0.82	0.83	401

جدول ۷. عملکرد روش **RF**

	precision	recall	f1-score	support
Benign	1.00	1.00	1.00	36406
Bot	0.85	0.82	0.83	401

جدول ۸. عملکرد روش **Naive base**

	precision	recall	f1-score	support
Benign	0.99	1	0.99	36406
Bot	0.10	0.01	0.02	401

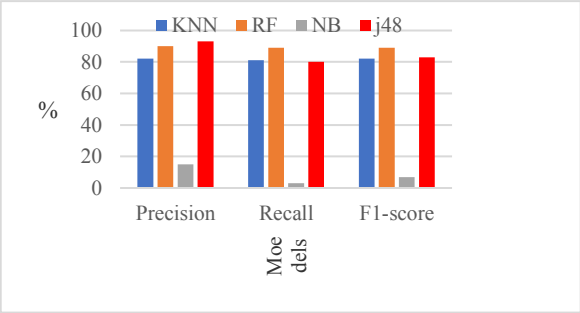
جدول ۹. عملکرد روش **J48**

	precision	recall	f1-score	support
Benign	1	1	1	36406
Bot	0.92	0.80	0.85	401

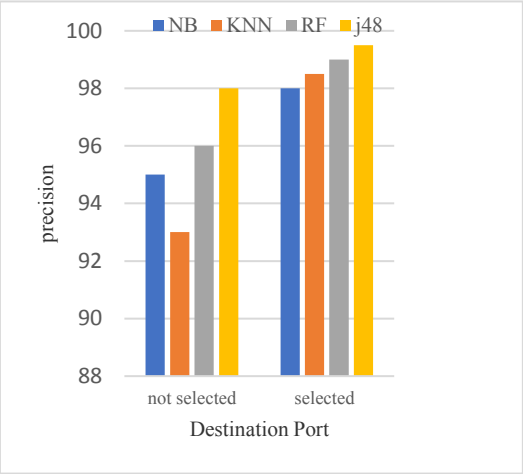
شکل (۷) نشان میدهد که خطای RF نسبت به KNN در دسته-بندی ترافیک کمتر است. در جدول (۱۰)، percision نشان میدهد که نسبت ترافیک Bot تشخیص داده شده درست، به کل تشخیص‌های Bot برابر ۹۱٪ است که نسبت به KNN پیشرفت بزرگی است. همچنین recall نشان میدهد که ۸۸٪ از مقدار ترافیک Bot به درستی دسته‌بندی شده‌اند. f1-score مانند KNN نزدیک‌بودن مقادیر recall و percision و f1-score را نشان میدهد که مدل هم در تشخیص ترافیک معمولی و هم در تشخیص ترافیک بات نت به خوبی عمل کرده است.

جدول ۱۰. مقایسه روش‌های ارائه شده

الگوریتم	دقت
KNN	٪۹۹/۴۵
RF	٪۹۹/۶۷
Naive Bayes	٪۹۸/۸
J48	٪۹۹/۶۴



شکل ۷. مقایسه دقت روش‌های ارائه شده



شکل ۸. مقایسه F1-score، precision و recall مدل‌های ارائه شده

در RF و KNN مقدار دقت بسیار نزدیک به هم است (هر دو حدود ۹۹٪) اما f1-score بالاتر در RF نشان میدهد که دسته-بندی ترافیک حمله با RF دقیق‌تر از KNN انجام شده و در عمل قابل اعتمادتر است. شکل (۸) نشان می‌دهد که الگوریتم Naive Bayes در طبقه‌بندی ترافیک حمله اصلاً خوب عمل نکرده اما ترافیک معمولی را به خوبی تشخیص داده است. با توجه به جدول (۱۰) مقادیر percision و recall و f1-score برای Bot نشان میدهد که مدل Naive Bayes در دسته‌بندی ترافیک حمله اصلاً خوب عمل نکرده است. تقریباً میتوان گفت که این الگوریتم نمیتواند ترافیک حمله را تشخیص دهد؛ این مورد باعث می‌شود در عمل، بخش اعظم ترافیک حمله روانه سرور شود که اصلاً قابل قبول نیست. شکل (۸) نشان میدهد که J48 نسبت سه الگوریتم دیگر، در تشخیص‌های حمله خطای کمتری داشته است. اما کمتربودن خطای تشخیص تنها عامل مؤثر نیست. مقدار recall هم در میزان دقت الگوریتم تأثیر گذار است. با توجه به جدول (۱۰)، precisio نشان‌دهنده این است که ترافیک حمله با دقت ۹۲٪ درست تشخیص داده شده است. Recall نشان میدهد که بخشی از ترافیک حمله را به اشتباه به عنوان ترافیک معمولی در نظر گرفته است. با توجه به F1-scoreمیتوان گفت که J48 نسبت به KNN بهتر عمل کرده اما باز هم به دلیل کمتر بودن recall تا حدود ۲۰٪ از ترافیک حمله روانه سرور می‌شود. به طور کلی الگورتیم J48 عمل‌کرد نسبتاً بهتری از بین سایر الگوریتم‌ها داشت و بسیار به RF نزدیک است.

۴. نتیجه‌گیری

حمله درگذشته به کمک روش‌های قدیمی‌تر، تلاش‌های زیاد برای تشخیص DDoS شده است و هر کدام از آن‌ها در زمان خود موفق بوده‌اند؛ اما امروزه به دلیل پیچیدگی حمله‌ها و افزایش حجم و تعداد آن‌ها و از جمله راحتی و سرعت انتشار بات نت‌ها، بهتر است که از روش‌های مدرن‌تری مانند Machine Learning برای تشخیص استفاده کنیم که هم از لحاظ دقت و هم سرعت قابل‌اعتماد هستند. در این مقاله، ما از ۴ الگوریتم Machine Learning برای تشخیص حمله استفاده کرده‌ایم. در پایان پس از بررسی نتایج به‌دست‌آمده از آزمایش‌های انجام شده، سه الگوریتم RF، KNN و J48 از لحاظ دقت بسیار به هم نزدیک بوده و الگوریتم Naive Bayes از همه ضعیف‌تر عمل کرده است. با اینکه دقت این الگوریتم حدود ۹۸٪ بود؛ ولی دیگر شاخص‌ها مانند precision، f1-score و recall اختلاف زیادی با دیگر مدل‌ها داشته است.

نتایج نشان داد که الگوریتم‌های RF، KNN و J48 از نظر دقت به طور قابل‌توجهی نزدیک به یکدیگر عمل کرده‌اند، درحالی‌که Naive Bayes عملکرد ضعیف‌تری نسبت به دیگر مدل‌ها داشته است. دقت بالای Naive Bayes حدود ۹۸٪، با این حال، شاخص‌های دیگری مانند precision، recall و f1-score آن اختلاف قابل توجهی نسبت به سایر الگوریتم‌ها نشان می‌دهد. به‌طور خاص، Naive Bayes در تشخیص ترافیک Bot به‌طور معناداری ضعیف عمل کرده و این مسئله می‌تواند منجر به از دست رفتن بخش قابل توجهی از ترافیک حمله شود.

در مقایسه با KNN و RF که دقت بالایی حدود ۹۹٪ دارند، RF به‌ویژه در مورد f1-score عملکرد بهتری داشته و نشان‌دهنده توانایی بالاتر آن در دسته‌بندی دقیق‌تر ترافیک‌های حمله است. همچنین، J48 نیز عملکرد نسبتاً خوبی از خود نشان داده است، هرچند که همچنان در شناسایی ترافیک حمله نسبت به RF کمی ضعیف‌تر عمل کرده است. تحلیل‌های بیشتر شامل ماتریس‌های درهم‌ریختگی، FPR، TPR و نمودار ROC نیز نشان‌دهنده نقاط قوت و ضعف هر الگوریتم در شناسایی انواع ترافیک بودند. به‌ویژه، ماتریس‌های درهم‌ریختگی نشان می‌دهند که مدل‌های RF و KNN در تشخیص دقیق ترافیک‌های عادی و حمله موفق‌تر بوده‌اند. برای تشخیص مؤثر حملات DDoS، استفاده از الگوریتم‌های یادگیری ماشین می‌تواند بهبود قابل توجهی در دقت و سرعت شناسایی داشته باشد. با این حال، برای کاربردهای عملی و واقعی، نیاز به بررسی و آزمایش این مدل‌ها در محیط‌های عملیاتی وجود دارد. علاوه بر این، هدف آینده تحقیق توسعه و آموزش مدل‌هایی است که قادر به شناسایی و طبقه‌بندی انواع مختلف حملات اینترنتی به‌طور جامع‌تری باشند.

ازآنجایی‌که این کار یک تحقیق بر روی یک پایگاه‌داده بوده، باید این مدل‌ها در دنیای واقعی آزمایش شوند و نتایج آن‌ها به طور کامل و دقیق بررسی شود. در آینده ما این الگوریتم‌ها را روی

سرور اجرا می‌کنیم و آن‌ها را بررسی می‌کنیم. این مدل‌های فقط برای تشخیص DDoS هستند و دیگر حمله‌ها را پشتیبانی نمی‌کنند. یکی از اهداف آینده آموزش مدلی است که بتواند انواع حمله‌های اینترنتی را تشخیص و طبقه‌بندی کند.

۵. مرجع‌ها

[1] Asadi, M.; BagherI, Z. "Detection of Denial of Service Attacks by Ensemble Learning Method"; Journal of Advanced Defense Science & Technology 14, no. 1. 2023. 51-68 (In Persian). Dor: 20.1001.1.26762935.1402.14.1.5.5

[2] Wang, B.; He, Y.; Shui, Z.; Xin, Q.; Lei, H. "Predictive Optimization of DDoS Attack Mitigation in Distributed Systems using Machine Learning"; Applied and Computational Engineering. 2024, 64, 95-100. DOI:10.54254/2755 2721/64/20241350

[3] Mittal, M.; Kumar, K.; Behal, S. "Deep Learning Approaches for Detecting DDoS Attacks: A Systematic Review"; Soft computing. 2023, 27(18), 13039-13075. DOI:10.1007/s00500-021-06608-1

[4] Kumari, P.; Jain, A. K. "A Comprehensive Study of DDoS Attacks Over IoT Network and Their Countermeasures"; Computers & Security. 2023, 127, 103096. DOI:10.1016/j.cose.2023.103096

[5] Mittal, M.; Kumar, K.; Behal, S. "Deep Learning Approaches for Detecting DDoS Attacks: A Systematic Review"; Soft computing. 2023, 27(18). DOI:10.1007/s00500-021-06608-1

[6] Sattari, M. T.; Bagheri, R.; Shirini, K.; Allahverdipour, P. "Modeling Daily and Monthly Rainfall in Tabriz using Ensemble Learning Models and Decision Tree Regression"; Scientific Journal of Golestan University, 2024, 5(18). Doi: 10.30488/CCR.2024.433394.1192

[7] Saleh Esfehani, M.; Abo Ali, M. "An IDS for Detection of Active Attacks Against Routing in Mobile Ad Hoc Networks"; Journal of Advanced Defense Science & Technology 1, no. 1. 2010, 15-22 (In Persian). Dor: 20.1001.1.26762935.1389.1.1.2.1

[8] Behal, S.; Kumar, K. "Characterization and Comparison of DDoS Attack Tools and Traffic Generators: A Review"; Int. J. Netw. Secur. 2017, 19(3). DOI: 10.6633/IJNS.201703.19(3).07)

[9] Michelena, Á,; Aveleira-Mata, J.; Jove, E.; Bayón-Gutiérrez, M.; Novais, P.; Romero, O. F.; Aláiz-Moretón, H. "A novel intelligent approach for man-in- the- middle attacks detection over internet of things environments based on message queuing telemetry transport"; Expert Systems. 2024, 41(2), e13263. DOI:10.1111/exsy.13263.

[10] Choorod, P.; George, W.; Anil Fernando. "Classifying tor traffic encrypted payload using machine learning"; IEEE Access. 2024. DOI: 10.1109/ACCESS.2024.3356073

[11] Priya, S. S.; Sivaram, M.; Yuvaraj, D.; Jayanthiladevi, A. "Machine Learning Based DDoS Detection"; In international conference on emerging smart computing and informatics. 2020,234-237. DOI: 10.1109/ESCI48226.2020.9167642

[12] Kazemitabar, J.; Taheri, R.; Kheradmandian, H. "A Novel Technique for Improvement of Intrusion Detection via Combining Random Forrest and Genetic Algorithm"; 2019, 287-296 (In Persian). Dor: 20.1001.1.26762935.1398.10.3.9.5

[13] Das, S.; Mahfouz, A. M.; Venugopal, D.; Shiva, S. "DDoS Intrusion Detection Through Machine Learning Ensemble"; IEEE 19th international conference on software Quality, Reliability and Security Companion. 2019, 471-477. DOI: 10.1109/QRS-C.2019.00090

[14] Asgharian, H.; Ahmad A.; Raahemi, B. "Engineered Feature Set to Detect Flooding Attacks in SIP Based VoIP'; Journal of Advanced Defense Science & Technology 8, no. 1. 2019: 61-69 (In Persian). Dor: 20.1001.1.26762935.1396.8.1.7.5

[15] Pande, S.; Khamparia, A.; Gupta, D.; Thanh, D. N. "DDOS Detection Using Machine Learning Technique"; Recent Studies on Computational Intelligence: Doctoral Symposium on Computational Intelligence, Springer Singapore. 2021, 59-68. DOI:10.1007/978-981-15-8469-5_5

[16] Abdulla, N. N.; Hasoun, R. K. "Review of Detection Denial of Service Attacks Using Machine Learning Through Ensemble Learning"; Iraqi Journal for Computers and Informatics. 2022, 48(1), 13-20. DOI:10.25195/ijci.v48i1.349

[17] Sattari, M. T.; Shirini, K.; Javidan, S." Evaluating the efficiency of dimensionality reduction methods in improving the accuracy of water quality index modeling using machine learning algorithms"; Water and Soil Management and Modelling .2024,4(2), 89-104. DOI: 10.22098/mmws.2023.12434.1241