نوع مقاله: پژوهشی

مدلی برای تشخیص نفوذ چند کلاسه با داده‌های نامتوازن مجموعه‌داده CICIDS-2017

محمود نیائی[۱]، دانشجوی دکتری مدیریت فناوری اطلاعات ، جعفر تنها[۲]، دانشیار، غلامرضا شاهمحمدی[۳]، دانشیار، علیرضا پورابراهیمی[۴]، استادیار

۱- دانشکده مدیریت و اقتصاد - دانشگاه آزاد علوم تحقیقات - تهران - ایران -mahmoud.niaie@srbiau.ac.ir

۲- دانشکده مهندسی برق و کامپیوتر - دانشگاه تبریز - تبریز - ایران -tanha @tabrizu.ac.ir

۳- دانشکده مهندسی برق و کامپیوتر - دانشگاه ایوان کی - سمنان - ایران - Shah_mohammadi@yahoo.co.uk

۴- دانشکده مدیریت و حسابداری دانشگاه آزاد اسلامی کرج - کرج - ایران- poorebrahimi@gmail.com

چکیده: امروزه بخش عمده‌ای از فعالیت‌ها و تعاملات اقتصادی، تجاری، فرهنگی، اجتماعی و حاکمیتی در تمام کشورها، از طریق فضای سایبر انجام می‌گیرد. باتوجه‌به آسیب‌پذیری‌های ذاتی موجود در این فضا، مخاطرات سامانه‌های مبتنی بر آن نیز در حال افزایش می‌باشند؛ بنابراین، امنیت شبکه‌ها و سیستم‌ها در مقابل انواع نفوذ، به یکی از مهم‌ترین چالش‌های عصر حاضر تبدیل شده است. در این پژوهش، یک مدل برای تشخیص نفوذ در شبکه، بررسی و پیشنهاد شده است. در روش پیشنهادی که یک روش چند کلاسه می‌باشد، از الگوریتم سنجاقک برای انتخاب ویژگی و از جنگل تصادفی به‌منظور دسته‌بندی استفاده شده است. داده‌های بکار رفته در پژوهش، مجموعه‌داده نامتوازن CICIDS-2017 بوده است لذا عملیات متوازن‌سازی در آن استفاده شده است. مسئله با الگوریتم‌های مختلف مورد آزمون قرار گرفته و بهترین الگوریتم انتخاب شده است. مقدار صحت در روش پیشنهادی برابر با ۰/۹۹۸۵ به‌دست‌آمده است. همچنین، نتایج پژوهش با چندین روش دیگر که توسط محققان قبلی پیشنهاد شده مورد مقایسه قرار گرفته است و این مقایسه نشان می‌دهد که روش پیشنهادی نسبت به اکثر پژوهش‌هایی که در مقاله معرفی شده‌اند، دارای معیارهای ارزیابی بالاتری بوده است.

واژه‌های کلیدی: تشخیص نفوذ، انتخاب ویژگی، الگوریتم سنجاقک، داده‌های نامتوازن، CICIDS-2017

A Model for Multi-Class Intrusion Detection with Imbalanced Data in the CICIDS-2017 Dataset

Mahmoud Niaei, PhD student[1], Jafar Tanha, Associate Professor[2], Gholamreza Shahmohammadi, Associate Professor[3], Alireza Poorebrahimi, Assistant Professor[4]

1- Faculty of Management and Accounting, Azad University, Research Sciences, Tehran, Iran, Email: Mahmoud.niaie@srbiau.ac.ir
2- Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran, Email: tanha @tabrizu.ac.ir
3-Faculty of Electrical and Computer Engineering, Ivan Key University, Semnan, Iran, Email: shah_mohammadi@yahoo.co.uk
4- Faculty of Management and Accounting, Islamic Azad University, Karaj, Iran, Email: poorebrahimi@gmail.com

Abstract: Today, most economic, commercial, cultural, social and governmental activities and interactions in all countries are carried out through cyberspace. Due to the inherent vulnerabilities in cyberspace, the risks of systems are increasing. Therefore, the security of networks and systems against various types of intrusion has become one of the most important challenges of the present age. In this research, a model for detecting network intrusion has been reviewed and proposed. The proposed method is a multi-class method and the dragonfly algorithm is used for feature selection and the Random forest algorithm is used for classification. For analysis, the CICIDS-2017 unbalanced data set has been used, so the balancing operation has been used. To select the method, different algorithms are tested and the best algorithm is selected. The value of accuracy in the proposed method is 0.9985. In addition, the research results have been compared with several other methods proposed by previous researchers, and this comparison shows that the proposed method were better than most of the researches presented in the article.

Keywords: Intrusion detection, feature selection, dragonfly algorithm , imbalanced data , CICIDS-2017.

تاریخ ارسال مقاله: ۱۴۰۰/۰۷/۱۹
تاریخ اصلاح مقاله: ۱۴۰۰/۱۱/۲۹
تاریخ پذیرش مقاله: ۱۴۰۰/۱۲/۱۳
نام نویسنده مسئول: جعفر تنها
نشانی نویسنده مسئول: ایران - تبریز - دانشگاه تبریز - دانشکده برق و کامپیوتر - گروه کامپیوتر.

۱- مقدمه

اصطلاح امنیت شبکه روشی برای تأمین امنیت اطلاعات شبکه می‌باشد که محرمانه بودن، یکپارچگی و دردسترس‌بودن سیستم را بررسی می نماید. یک سیستم تشخیص نفوذ، نوعی سرویس امنیتی می‌باشد که وظیفه شناسایی و تشخیص هرگونه استفاده غیرمجاز به سیستم، سوءاستفاده یا آسیب‌رسانی توسط کاربران داخلی و خارجی را بر عهده دارد. عموماً سیستم‌های تشخیص نفوذ در کنار دیواره‌های آتش[۱] و به‌صورت مکمل امنیتی برای آن‌ها مورداستفاده قرار می‌گیرند.

در سال ۲۰۲۰ در مقاله‌ای، پژوهشگران از روش شبکه عصبی عمیق به‌صورت باینری استفاده کرده‌اند. این پژوهش، توانایی تشخیص نفوذ و عدم نفوذ را در سیستم داشته و نوع حمله در آن تشخیص داده نمی‌شود. در این پژوهش مجموعه‌داده KDD99 و دو مجموعه‌داده دیگر مورد بررسی قرار گرفته و مقدار صحت نهایی در این روش ۰/۹۶۰۳ درصد بوده است [۱].

در پژوهش[۲] از یک روش یادگیری با عنوان STL[۲] بهره گرفته شده است. این بررسی که در سال ۲۰۱۶ به‌صورت باینری و چند کلاس انجام و نتایج آن با روش SMR[۳]مقایسه شده است، حداکثر صحت مربوط به STL ، ۰/۹۸۸۴درصد گزارش شده است. همچنین در سال ۲۰۲۰ ، روش یادگیری CNN و DNN برای تشخیص نفوذ بررسی شده است. این دو روش بر روی مجموعه‌داده NSL-KDD به‌صورت چند کلاس بررسی شده و مقدار بازخوانی ۰/۹۷۴۸ درصد برای CNN و مقدار دقت۰/۹۷۸۹ درصد برای CNN محاسبه شده است [۳].

در منبع [۴] از روش یادگیری SVM برای تشخیص حمله و نفوذ به‌صورت باینری استفاده شده است. برای شبیه‌سازی و ارزیابی از مجموعه‌داده KDD99 استفاده شده و درصد صحت به‌دست‌آمده ۰/۹۸۵۷بوده است.

در پژوهش حاضر، یک روش تشخیص نفوذ با عنوان ID2F[۴]پیشنهاد شده است. این روش از یک شیوه جدید انتخاب ویژگی با عنوان الگوریتم سنجاقک استفاده می‌کند، الگوریتم فرابتکاری سنجاقک که الهام گرفته از رفتار سنجاقک طبیعی می‌باشد، برای اولین‌بار در سال ۲۰۱۵ ارائه شده و در مسائل ان‌پی - سخت[۵] (مسائلی که تاکنون راه‌حلی سریع و قابل‌حل در زمان معقول، پیدا نشده است) به کار گرفته می‌شود [۵].

همچنین، در این پژوهش به‌منظور تشخیص و دسته‌بندی حملات ترافیک شبکه، از الگوریتم جنگل تصادفی استفاده شده است. انتخاب این روش، پس از اعمال روش‌های مختلف و آزمون آنها، به دلیل عملکرد مناسب و میزان صحت و دقت بالا و زمان کمتر، بوده است. در این تحقیق از مجموعه‌داده‌های نامتوازن CICIDS-2017 که یک مجموعه‌داده جدید به شمار می‌آید استفاده شده است و عملیات متوازن‌سازی بر روی آن انجام شده است. سیستم مورد تحقیق، علاوه بر تشخیص نرمال یا حمله بودن ترافیک وارده، توانایی تشخیص نوع حملات را نیز دارد؛ لذا نوآوری و مزایای روش پیشنهاد شده در پژوهش را می‌توان به‌صورت زیر عنوان نمود:

- استفاده از یک شیوه جدید انتخاب ویژگی در تشخیص نفوذ
- چند کلاسه بودن روش پیشنهادی
- استفاده از مجموعه‌داده جدید CICIDS-2017

ساختار مقاله در ادامه به شرح ذیل می‌باشد، ابتدا در بخش دوم به معرفی پژوهش‌های پیشین پرداخته می‌شود، سپس در بخش سوم مجموعه‌های داده و الگوریتم‌های روش پیشنهادی معرفی شده‌اند. در بخش چهارم نتایج پژوهش بیان می‌گردد و در نهایت بخش پنجم به جمع‌بندی روش پیشنهادی می‌پردازد.

۲- پژوهش‌های پیشین

سیستم‌های تشخیص نفوذ از نظر نوع فناوری تشخیص، به سیستم‌های تشخیص نفوذ از طریق روش‌های آماری، روش‌های مبتنی بر دانش و روش‌های یادگیری ماشین قابل تقسیم می‌باشند. یادگیری ماشین یکی از شاخه‌های پرکاربرد هوش مصنوعی است که به مطالعه الگوریتم‌هایی می‌پردازد که بر اساس آن رایانه‌ها و سامانه‌ها توانایی یادگیری پیدا می کنند [۶]. از انواع روش‌های یادگیری ماشین می‌توان به الگوریتم‌های درخت تصمیم، ماشین‌های بردار پشتیبان، شبکه‌های بیزی، شبکه‌های عصبی مصنوعی، یادگیری عمیق و همچنین جنگل تصادفی اشاره نمود که در پژوهش‌های مختلف از آنها استفاده شده است [۷].

بدی و همکارانش [۸]، یک سیستم تشخیص نفوذ به نام Siam-IDS را معرفی نموده است. این سیستم با استفاده از شبکه عصبی سیامی‌برای رفع مشکل عدم تعادل کلاس در مجموعه‌داده NSL-KDD پیشنهاد شده است. نتایج صحت و بازخوانی مربوط به هر کلاس به‌صورت جداگانه محاسبه و بیشترین مقدار صحت ۰/۶۱۱۴ و بیشترین مقدار بازخوانی، ۰/۶۳۴ گزارش شده است.

در پژوهشی دیگر [۹] ، محقق سه روش NB ،SVM و J48 را مقایسه کرده و مقادیر صحت هر کلاس را محاسبه نموده است. پژوهشگران در این مقاله که در سال ۲۰۱۵ منتشر شده از مجموعه‌داده NSL-KDD استفاده نموده و میانگین صحت کلاس‌ها را به ترتیب ۰/۹۵۲، ۰/۷۳۳ و ۰/۹۸۸۸ گزارش نموده‌اند.

در سال ۲۰۲۱ مقاله دیگری [۱۰] منتشر شده که محقق در آن از یک شبکه عصبی عمیق مبتنی بر پشته به‌منظور کشف نفوذ استفاده نموده است. در این مقاله، محقق برای تحلیل روش خود از مجموعه‌داده‌های CICIDS 2017، NSL-KDD و UNSW-NB15 استفاده نموده که نتیجه صحت به‌دست‌آمده برای مجموعه‌داده NSL-KDD مقدار ۰/۸۹۹۷ گزارش شده است.

ویناياکمار و همکارانش [۱۱]، روش یادگیری شبکه‌های عصبی عمیق را پیشنهاد داده است. این پژوهش که از چند مجموعه به همراه مجموعه‌داده CICIDS2017 استفاده کرده بهترین نتایج را در روش DNN با ۵ لایه یافته است. در پژوهش دیگر، ریوس و همکارانش

[۱۲]، از مجموعه‌داده CICIDS2017 و چند مجموعه دیگر استفاده کرده و روش BLS را بکار برده و بهترین نتیجه صحت مقدار ۰/۹۶۶۳ بر روی مجموعه‌داده فوق به‌دست‌آمده است.

در مقاله دیگری، احمد آهمین و همکارانش [۱۳]، از الگوریتم درخت تصمیم و مدل‌های مبتنی بر قوانین در طراحی سیستم تشخیص نفوذ سلسله‌مراتبی استفاده نموده‌اند. این پژوهش که در سال ۲۰۱۹ به چاپ رسیده بر روی مجموعه‌داده CICIDS2017 انجام شده است. پژوهشگران در این تحقیق، برای طبقه‌بندی از ترکیب سه الگوریتم REP Tree، JRip و Forest PA استفاده کرده و مقدار صحت را برابر ۰/۹۶۶۶۵ و مقدار بازخوانی را به میزان ۰/۹۴۴۵۷ گزارش نموده‌اند.

در پژوهش [۱۴] در سال ۲۰۲۰ از روشی به نام HADMLP-IDS برای تشخیص نفوذ استفاده شده است. این روش مبتنی بر شبکه عصبی و ترکیب الگوریتم کلونی زنبور مصنوعی و الگوریتم سنجاقک می‌باشد. در این روش که از مجموعه‌داده‌های KDD 99، NSL-KDD، UNSW-NB15 و ISCX2012 برای تحلیل استفاده شده، با چندین روش مختلف مقایسه شده است. مقدار صحت برای مجموعه‌داده‌های ذکر شده، به ترتیب ۰/۸۸۷، ۰/۹۱۷، ۰/۹۴۴و ۰/۹۱۲۴ گزارش شده است. همچنین در مقاله [۱۵] که از مجموعه‌داده CICIDS2017 استفاده شده، الگوریتم DNN بکار گرفته شده و نتایج آن عبارت‌اند از: حداکثر صحت ۹۹/۹۵ و دقت ۹۴/۳۱ و بازخوانی ۹۵/۶۲و F1-Score برابر ۹۴/۱بوده است.

چنانچه از موارد فوق برمی‌آید، در پژوهش‌های مختلف از روش‌های متفاوت برای تشخیص نفوذ استفاده می‌شود. در ضمن برخی مقالات، تشخیص نفوذ را به‌صورت باینری بررسی نموده‌اند و توانایی تشخیص نوع حمله را نداشته‌اند. همچنین در اکثر مقالات فقط معیار صحت محاسبه شده است.

براین‌اساس، برای رفع موارد فوق در پژوهش حاضر سعی شده است روشی مورد بررسی قرار گیرد که بتواند نوع حملات را نیز مشخص نماید. همچنین معیارهای دقت، بازخوانی و F1-Score نیز محاسبه شده و با چندین الگوریتم یادگیری ماشین و عمیق مورد مقایسه قرار گیرد. در این پژوهش از یک روش جدید انتخاب ویژگی استفاده شده تا حجم محاسبات و زمان اجرای آن تا اندازه زیادی کاهش یابد. در انتهای بخش ۴، مقادیر معیارهای ارزیابی پژوهش حاضر با مقادیر مطالعاتی که از مجموعه‌داده CICIDS2017 استفاده نموده‌اند مقایسه خواهد شد.

۳- روش پیشنهادی (ID2F)

در این فصل روش ID2F تشریح شده است. این روش مبتنی بر انتخاب ویژگی با استفاده از یک الگوریتم فرابتکاری[۶] می‌باشد. همچنین از الگوریتم جنگل تصادفی برای دسته‌بندی حملات استفاده شده است. در ادامه این بخش، ابتدا مجموعه‌داده‌ها معرفی می‌شوند و سپس به تشریح الگوریتم سنجاقک و جنگل تصادفی پرداخته خواهد شد و در نهایت الگوریتم مدل IDF2 معرفی می‌شود.

۳-۱- معرفی مجموعه‌داده

در این پژوهش، از مجموعه‌داده CICIDS-2017 به‌عنوان یکی از جدیدترین مجموعه‌داده‌های این حوزه استفاده شده است. این مجموعه‌داده به‌منظور طراحی و ساخت IPS و IDS توسط مؤسسه امنیت سایبری کانادا طراحی شده است. مجموعه‌داده CICIDS-2017 حاوی داده‌های اصلی ترافیک شبکه بوده و برای رفع مشکل کمبود مجموعه‌داده به‌روز و معتبر برای تشخیص نفوذ طراحی شده است [۱۶].

۳-۲- انواع حملات در مجموعه‌داده CICIDS-2017

مجموعه‌داده CICIDS-2017 شامل پنج‌روز ترافیک شبکه (از دوشنبه ۳ جولای ۲۰۱۷ تا جمعه ۷ جولای ۲۰۱۷) می‌باشد. روز اول فقط شامل ترافیک عادی است و ۴ روز بعد شامل ترافیک عادی و ۱۴ نوع حمله است که با دستگاه فلومتر[۴]اخذ شده‌اند. این مجموعه‌داده شامل ۲۸۳۰۷۴۳ سطر با برچسب و ۷۸ متغیر می‌باشد، انواع حملات این مجموعه‌داده به‌صورت ذیل می‌باشد [۱۷]:

- جستجوی پورت [۸] : این نوع حملات که غالباً در مرحله ابتدایی حملات دیگر صورت می‌پذیرد، برای بررسی پورت‌های باز، پورت‌های مهم و یافتن نقاط ضعف سیستم‌های هدف استفاده می‌شود.
- محروم‌سازی از سرویس[۹]: تلاش برای خارج کردن موقت یا دائمی ماشین و منابع شبکه از دسترس کاربران مجاز می‌باشد.
- تزریق SQL [۱۰]: نوعی حمله که مهاجم با استفاده از دستورات SQL، عملیاتی را در پایگاه‌داده وب‌سایت آسیب‌پذیر انجام می‌دهد.
- حملات HULK [۱۱] : یک ابزار انکار سرویس‌دهنده وب است. این حمله که برای اهداف تحقیق طراحی شده است، حجم زیادی از ترافیک منحصربه‌فرد و مبهم را در یک سرور وب ایجاد می‌کند.
- حملات چشم طلایی [۱۲] : نوعی از حملات انکار سرویس است که مهاجم با درخواست مداوم URL های منفرد یا چندگانه و نگه‌داشتن ارتباطات، برای سرریز کردن منابع سرورهای وب استفاده می‌کند.
- نفوذ از داخل شبکه[۱۳] : در این نوع حملات، از یک کاربر آسیب‌پذیر در داخل شبکه استفاده می‌شود و پس از نفوذ به سیستم کاربر، اسکن شبکه داخلی و اجرای حملات دیگر در شبکه فراهم می‌گردد.
- بات نت[۱۴] : شبکه‌ای از چندین کامپیوتر است که مخفیانه و بدون اطلاع کاربران واقعی، توسط مهاجم، برای انجام فعالیت‌های مخرب و حملات مختلف تحت کنترل گرفته شده‌اند. این حملات، به‌صورت جدول شماره (۱) به شش دسته حمله و یک دسته نرمال دسته‌بندی شده‌اند [۱۷].

جدول ۱: انواع حملات در مجموعه‌داده CICIDS-2017

شماره برچسب	برچسب	انواع ترافیک	تعداد مشاهدات
۰	ترافیک نرمال	Benign	۲۲۷۳۰۹۷
۱	بروت فورس	FTP-Patator SSH-Patator	۱۳۸۳۵
۲	محروم‌سازی از سرویس	DDoS-GoldenEye Hulk-Slow-httptest Slowloris-Heartbleed	۳۸۰۶۹۹
۳	حملات وب	Web Attack – Brute Force Web Attack – SQL Injection Web Attack – XSS	۲۱۸۰
۴	فیلتر	infiltration of the network from inside	۳۶
۵	بات	Botnet	۱۹۶۶
۶	جستجوی پورت	PortScan	۱۵۸۹۳۰

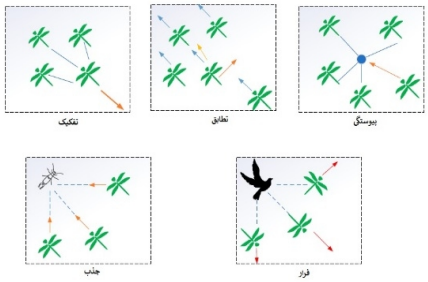
۳-۳- عدم توازن در مجموعه‌داده CICIDS-2017

چنانچه در جدول شماره (۱) مشاهده می‌شود تعداد مشاهدات ترافیک نرمال در مجموعه‌داده بسیار بیشتر از تعداد برخی حملات می‌باشد، علی‌الخصوص برچسب حمله Infiltration که مقدار کل ترافیک مشاهده شده از آن ۳۶ حمله می‌باشد که در مقابل تعداد کل ترافیک نرمال، بسیار کم می‌باشد. این عدم توازن مجموعه‌داده، موجب می‌شود یادگیری مناسبی از داده‌ها به دست نیامده و پیش‌بینی حملات جدید، به‌درستی انجام نشود. برای حل این مشکل در این پژوهش از روش نمونه‌گیری افزایشی و کاهشی استفاده شده است. در این روش‌ها تعداد نمونه‌های مشابه از نوع سطرهایی که تعداد آنها کمتر می‌باشد، به مجموعه آموزش اضافه و یا تعدادی از سطرها که مشاهدات بیشتری از آنها دیده شده است کاهش داده می‌شود. این تغییرات به نحوی انجام می‌شود که داده‌های شامل برچسب‌های مختلف، سهم مساوی در آموزش سیستم ایفا نمایند. همچنین در این پژوهش از روش وزن‌دهی کلاس‌ها استفاده شده است تا وزن کلاس‌های شامل برچسب بسیار پایین با بقیه کلاس‌ها متناسب گردد. با اعمال این تغییرات بر روی مجموعه‌داده، یادگیری به طور عادلانه بین کلاس‌ها تقسیم می‌شود و میزان دقت کاذبی که به‌خاطر یادگیری از یک کلاس با تعداد ترافیک بیشتر دیده می‌شود، اصلاح می‌گردد.

۳-۴- الگوریتم انتخاب ویژگی سنجاقک

معمولاً در مجموعه‌های داده، تعدادی از ویژگی‌ها، بار اطلاعاتی چندانی ندارند. حذف نکردن این ویژگی‌ها مشکلی ازلحاظ اطلاعاتی ایجاد نمی‌کند ولی بار محاسباتی را برای کاربرد موردنظر بالا می‌برد. علاوه بر این باعث می‌شود که اطلاعات غیرمفید زیادی به همراه داده‌های مفید ذخیره شود. در حالت کلی، الگوریتم‌های انتخاب ویژگی، یک فضای چندبعدی را به فضایی با ابعاد کمتر نگاشت می‌دهند.

از انواع الگوریتم‌های انتخاب ویژگی که در پژوهش‌ها از آنها استفاده می‌شود، می‌توان به انتخاب ویژگی با استفاده از الگوریتم ژنتیک، الگوریتم کرم شب‌تاب، الگوریتم کلونی زنبورعسل، الگوریتم فاخته و



شکل ۱: قوانین پنج‌گانه در زندگی سنجاقک

انتخاب ویژگی با استفاده از الگوریتم رقابت استعماری اشاره کرد [۱۸]. در این پژوهش از الگوریتم فراابتکاری سنجاقک به‌عنوان انتخاب ویژگی استفاده شده است. این الگوریتم بر اساس رفتار سنجاقک طبیعی طراحی شده است. رفتار هوشمندانه یک سنجاقک، مطابق شکل (۱) بر پنج اصل "اجتناب از برخورد با سایر افراد همسایه"، "تنظیم سرعت باتوجه‌به موقعیت سایر افراد همسایه"، "تمایل سنجاقک به سمت مرکز ثقل همسایه‌ها"، "جذب به سمت منبع غذایی" و "فرار از دشمن" استوار است. براین‌اساس در الگوریتم سنجاقک پنج تابع تعریف و مورداستفاده قرار گرفته است [۵].

- تابع تفکیک: سنجاقک‌ها در حالت عادی، در گروه‌های کوچکی تجمع می‌کنند و در نواحی مختلف به‌صورت یک ازدحام ایستا پرواز می‌کنند. تابع تفکیک به شیوه‌ای اطلاق می‌شود که هر سنجاقک، موقعیت خود را برای جلوگیری از برخورد با همسایگان تنظیم می‌کند. برای این منظور، سنجاقک‌ها از رابطه (۱) پیروی می‌کنند.

$$Si = - \sum_{j=1}^{N} X - X_j \qquad (۱)$$

که در آن X موقعیت فعلی سنجاقک را نشان می‌دهد، Xj نماینده موقعیت همسایه X بوده و N اندازه فضای موجود است.

- تابع سرعت تطابق: تنظیم سرعت سنجاقک‌ها باتوجه‌به سنجاقک‌های نزدیک که به‌صورت ریاضی با استفاده از رابطه (۲) نشان‌داده‌شده است.

$$Ai = \frac{\sum_{j=1}^{N} V_j}{N} \qquad (۲)$$

که در آن Vj سرعت jامین همسایه می‌باشد.

- تابع پیوستگی: تمایل هر سنجاقک به سمت مرکز ثقل همسایه‌ها مطابق تابع پیوستگی می‌باشد که با استفاده از فرمول ریاضی (۳) قابل محاسبه است.

$$Ci = \frac{\sum_{j=1}^{N} X_j}{N} - X \qquad (۳)$$

که در آن Xj نماینده موقعیت سنجاقک همسایه X را نشان می‌دهد.

- تابع جذب: در هنگام شکار یا فرار از دشمن، سنجاقک‌ها در گروه‌های بزرگ و در امتداد یک‌جهت (ازدحام پویا)، پرواز می‌کنند. متمایل شدن سنجاقک‌ها به سمت منبع غذایی با تابع جذب قابل محاسبه می‌باشد که با رابطه (۴) محاسبه می‌شود.

$$Fi = X+ - X \qquad (۴)$$

که در آن X+ موقعیت منبع غذایی را بیان می‌کند.

- تابع فرار از دشمن: رفتاری که هر سنجاقک در هنگام حمله دشمن، برای زنده ماندن خود انجام می‌دهد. این تابع را در فرمول (۵) می‌توان مشاهده کرد.

$$Ei = X- + X \quad (۵)$$

که X- موقعیت دشمن را بیان می‌کند.

در الگوریتم سنجاقک، فرایند بهینه‌سازی، با ایجاد یک مجموعه از راه‌حل‌های تصادفی برای مسئله داده شده شروع می‌شود. در حقیقت، بردارهای موقعیت و گام مربوط به سنجاقک‌ها به‌صورت تصادفی و باتوجه‌به حد پایین و بالای متغیرها مقداردهی اولیه می‌شوند و در هر تکرار، بردار بهترین موقعیت و گام هر سنجاقک، پی‌درپی به‌روزرسانی می‌شود.

برای به‌روزرسانی بردار موقعیت سنجاقک، از بردار گام (مرحله) (ΔX) و بردار موقعیت فعلی، استفاده می‌شود. بردار گام جهت حرکت سنجاقک را نشان می‌دهد و بر اساس فرمول (۶) محاسبه می‌شود:

$$\Delta Xt+1 = (sSi + aAi + cCi + fFi + eEi) + wXt \quad (۶)$$

که بردارهای وزن در رابطه (۶) با استفاده از پارامترهای e,w,a,c,s,f نشان‌داده‌شده‌اند.

بدین ترتیب بردار موقعیت سنجاقک با استفاده از فرمول (۷) به‌روزرسانی می‌شود.

$$Xt+1 = Xt + \Delta Xt+1 \quad (۷)$$

که پارامتر t تعداد تکرارها را نشان می‌دهد.

شبه‌کد الگوریتم سنجاقک به‌صورت شکل (۲) می‌باشد [۵]. به‌منظور انتخاب ویژگی، یک سنجاقک ویژگی‌هایی را به‌صورت تصادفی انتخاب نموده و با استفاده از الگوریتم KNN، برای هر نمونه با ویژگی‌های جدید، دسته موردنظر را پیش‌بینی می‌کند. در این پیش‌بینی، شبیه‌ترین نمونه‌ها از طریق محاسبه فاصله منهتن (بین نمونه برچسب نزده و نمونه‌های برچسب زده) انجام می‌گردد.

Initialize the dragonflies population Xi (i = 1, 2, ..., n)
Initialize step vectors ΔXi (i = 1, 2, ..., n)
while the end condition is not satisfied
Calculate the objective values of all dragonflies
Update the food source and enemy
Update w, s, a, c, f, and e
Calculate S, A, C, F, and E مطابق روابط(۱)تا(۵)
Update neighbouring radius
If a dragonfly has at least one neighbouring dragonfly
Update velocity vector مطابق رابطه(۶)
Update position vector مطابق رابطه(۷)
Else
Update the position vector using L'evy flight
End if
Check and correct the new positions based on the boundaries of variables
End while.

شکل ۲: شبه‌کد سنجاقک[۵]

سپس باتوجه‌به وضعیت همسایه‌ها، در خصوص دسته موردنظر تصمیم گیری می‌کند. این پیش‌بینی، با ویژگی‌های انتخاب شده برای تمام نمونه‌ها انجام می‌شود سپس مقدار نمونه‌های پیش‌بینی شده، با مقدار برچسب واقعی مقایسه می‌شود اگر برابر بودند، مقدار (۱) و در غیر این صورت (۰) برمی‌گرداند. این مقایسه برای تمام نمونه‌ها صورت می‌گیرد و مجموع آنها محاسبه و سنجاقکی که مجموعه نمونه‌های درست بیشتری را پیش‌بینی کرده باشد به‌عنوان بهترین گزینه انتخاب می‌شود.

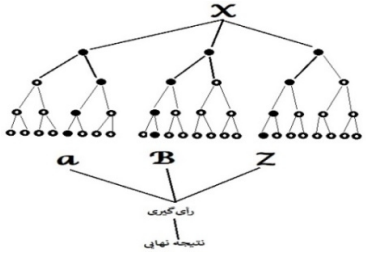
حال مطابق روابط (۶) و (۷) سنجاقک‌ها جابه‌جا شده و مجدداً مقادیر فوق محاسبه شده و با بهترین گزینه قبلی مقایسه می‌گردد. الگوریتم تا تعداد تکرار مشخص (که توسط محقق انتخاب می‌گردد) ادامه پیدا می‌کند و سنجاقک دارای بهترین گزینه انتخاب می‌شود و ویژگی‌هایی که آن سنجاقک انتخاب کرده ویژگی‌های بهینه خواهند بود [۱۹].

۳-۵- الگوریتم جنگل تصادفی

این روش یک الگوریتم ترکیبی می‌باشد که از درخت‌های تصمیم، برای الگوریتم‌های ساده و ضعیف خود استفاده می‌کند. در الگوریتم جنگل تصادفی به هرکدام از درخت‌ها، یک زیرمجموعه‌ای از داده‌ها اختصاص می‌یابد این درخت‌ها با داده‌های زیرمجموعه، می‌توانند تصمیم بگیرند و مدل طبقه‌بند خود را بسازند.

در هنگام پیش‌بینی، هرکدام از الگوریتم‌های یاد گرفته‌شده، یک نتیجه را جهت پیش‌بینی برمی‌گردانند. الگوریتم جنگل تصادفی در نهایت، می‌تواند با استفاده از رأی‌گیری، طبقه‌ای را که بیشترین رأی را کسب کرده است انتخاب کند و به‌عنوان طبقه نهایی جهت انجام عملیات طبقه‌بندی قرار دهد. عملکرد یک الگوریتم جنگل تصادفی در شکل (۳) مشاهده می‌شود [۲۰].

در جنگل تصادفی، برای آموزش داده‌ها چندین درخت ساخته می‌شود که در هرکدام از آنها ترتیب قرارگرفتن ویژگی‌ها در اعماق درخت، به‌صورت تصادفی تعیین می‌شود. سپس داده‌های یادگیری به‌تمامی این درختان تصمیم داده می‌شود و هر یک به‌صورت جداگانه آموزش می‌بینند. در مرحله بعد متغیر هدف توسط تمامی درختان پیش‌بینی می‌شود. سپس نتیجه نهایی، با رأی اکثریت درخت‌ها تعیین می‌گردد. از مزایای الگوریتم جنگل تصادفی می‌توان به پاسخگویی بهتر در داده‌های با حجم بالا، پایداری الگوریتم، قابل‌استفاده برای رگرسیون



شکل ۳: نحوه عملکرد جنگل تصادفی

شکل ۴: مدل ID2F

و دسته‌بندی، محدود بودن تعداد پارامترها و آسان بودن استفاده از آن نام برد [۲۱]. لازم به ذکر است در این پژوهش پیش از انتخاب الگوریتم جنگل تصادفی، اکثر الگوریتم‌های یادگیری ماشین و یادگیری عمیق مورد آزمایش قرار گرفته و الگوریتم جنگل تصادفی در مقایسه با آنها توانسته است مقادیر بهتری از معیارهای ارزیابی را کسب نماید. مدل ID2F در شکل شماره (۴) نشان‌داده‌شده است.

چنانچه در مدل فوق مشاهده می‌شود، داده‌های خام پس از پیش‌پردازش و یکسان کردن مقیاس داده‌ها از طریق روش کمترین - بیشترین، با استفاده از الگوریتم سنجاقک انتخاب ویژگی شده و سپس نتایج از طریق الگوریتم جنگل تصادفی دسته‌بندی می‌شوند.

در این پژوهش از معیارهای ارزیابی صحت[۱۵]، دقت[۱۶]، میزان بازخوانی، F1-Score و همچنین معیار زمان استفاده شده است.

۴- آزمایش‌ها و نتایج

در این بخش، ابتدا نتایج پیش‌پردازش داده‌ها و انتخاب ویژگی‌ها بیان می‌گردد و پس از آن به عملیات دسته‌بندی و مقایسه نتایج ID2F با سایر روش‌های موجود در این زمینه پرداخته خواهد شد. لازم به ذکر است سیستم رایانه مورداستفاده در این پژوهش، دارای پردازنده Core i5 و مقدار ۸ گیگابایت RAM بوده و از Matlab2020b و پایتون در محیط Colab استفاده شده است.

۴-۱- پیش‌پردازش داده‌ها

در مجموعه‌داده CICIDS-2017، ابتدا ویژگی‌هایی که مقدار صفر داشتند کنار گذاشته شدند. همچنین داده‌های گم شده و داده‌هایی که مقداری برای آنها ثبت نشده بود حذف گردیدند. در ادامه برای مجموعه‌داده CICIDS-2017، حملات به‌صورت شش گروه حمله و یک گروه نرمال عددی شدند. پس از آن برای یکسان کردن مقیاس داده‌ها و نرمال‌سازی آنها، از روش کمترین - بیشترین استفاده شد.

۴-۲- نتایج انتخاب ویژگی سنجاقک

مجموعه‌داده CICIDS-2017 پس از پیش‌پردازش، دارای ۷۰ ویژگی می‌باشد. عملیات انتخاب ویژگی با الگوریتم سنجاقک، با اعمال مقادیر مختلف برای متغیرهای تعداد مشاهدات، تعداد سنجاقک‌ها، تعداد تکرارها و تعداد همسایگی‌های الگوریتم KNN بررسی گردید و مشاهده شد که با اعمال کل مشاهدات و تعداد ۲ تکرار و تعداد k برابر ۵ و تعداد ۱۰ سنجاقک، موجب انتخاب تعداد ۲۸ ویژگی بادقت ۹۹/۸۴ شد. شماره ویژگی‌های انتخاب شده در جدول (۲) مشاهده می‌شوند. لازم به ذکر می‌باشد این مجموعه ویژگی‌ها پس از آزمون‌وخطا بر روی معیارهای مختلف به مقدار دقت موردنظر رسیده و انتخاب شده است.

پس از انجام مراحل پیش‌پردازش، داده‌ها به دودسته ویژگی‌ها (X) و برچسب‌ها (Y) تقسیم شدند و سپس تعداد ۲۰درصد از سطرهای مجموعه‌داده به طور تصادفی انتخاب و با نام X_test و y_test (داده‌های آزمایشی) نام‌گذاری و بقیه داده‌ها (۸۰ درصد) به نام X_train، y_train (داده‌های آموزشی) نام‌گذاری شدند. این روش انتخاب به شکلی بوده که به نسبتِ تعداد هر کلاس در مجموعه‌داده، ترکیبی از تمام انواع کلاس‌ها در هر دو گروه آموزشی و آزمایشی موجود باشند. به‌منظور مقایسه صحیح نتایج، X_test و y_test را به‌صورت ثابت نگهداری نموده و تمامی تغییرات، صرفاً بر روی داده‌های آموزشی اعمال شده‌اند.

در مرحله بعد الگوریتم‌های یادگیری رگرسیون لجستیک (LR)، درخت تصمیم (DT)، نزدیک‌ترین همسایگی (KNN)، جنگل تصادفی (RF)، گرادیان بوستینگ (GB) و شبکه عصبی (NN) و عمیق بازگشتی (LSTM-GRU-RNN) بر روی X_train و y_train اعمال و نتایج دقت، صحت، میانگین تقریبی، بازخوانی و زمان اجرا در جداول شماره (۳) تا (۷)ثبت شده‌اند. در روش‌های عمیق نیز آرگومان‌های مختلف آزمایش و بهترین مقادیر به‌صورت جدول شماره (۴) به‌دست‌آمده‌اند. چنانچه در جدول شماره (۳) و (۴) ملاحظه می‌شود، بهترین مقدار صحت از بین الگوریتم‌های یادگیری ماشین و روش‌های عمیق، مربوط به الگوریتم جنگل تصادفی بوده است که دارای میزان صحت ۰/۹۹۸۰ می‌باشد. جدول شماره ۵ مدت‌زمان اجرای الگوریتم‌ها

جدول ۲: گزارش ویژگی‌های انتخاب شده از مجموعه‌داده CICIDS-2017 توسط الگوریتم سنجاقک

ویژگی های انتخاب شده	F1	F2	F3	F4	F5
	۱	۲	۵	۷	۱۲
F6	**F7**	**F8**	**F9**	**F10**	**F11**
۱۵	۱۸	۲۴	۲۶	۲۷	۳۰
F12	**F13**	**F14**	**F15**	**F16**	**F17**
۳۳	۳۴	۳۷	۳۹	۴۱	۴۴
F18	**F19**	**F20**	**F21**	**F22**	**F23**
۴۷	۴۸	۴۹	۵۲	۵۳	۵۵
F24	**F25**	**F26**	**F27**	**F28**	
۵۶	۵۷	۵۸	۵۹	۶۲	

جدول ۳: نتایج صحت الگوریتم‌های یادگیری‌ماشین با داده‌های اولیه(پس از مراحل پیش‌پردازش)

الگوریتم	LR	DT	KNN	RF	GB
ACC	۰/۹۰۱۹	۰/۵۰۲۷	۰/۹۹۴۴	۰/۹۹۸۰	۰/۹۹۷۴

جدول ۴: نتایج صحت روش‌های عمیق بازگشتی با داده‌های اولیه(پس از مراحل پیش‌پردازش)

الگوریتم	NN	RNN	GRU	LSTM
لایه پنهان	۲	۲	۲	۲
تعداد گره‌ها	۵۱۲ – ۲۵۶	۵۱۲- ۵۱۲	۵۱۲-۵۱۲	۵۱۲-۵۱۲
نرخ یادگیری	۰/۰۰۱	3e-4	3e-4	3e-4
تعداد تکرار	۵	۳	۳	۳
فعال‌سازی	Relu – SoftMax	Relu – SoftMax	Relu – SoftMax	Relu – SoftMax
Batch	۳۲	۶۴	۶۴	۶۴
Verbose	۲	۲	۲	۲
ACC	۰/۸۳۱۱	۰/۸۵۱۷	۰/۸۳۰۹	۰/۸۳۳۸

جدول ۵: زمان اجرای الگوریتم‌ها با داده‌های اولیه(پس از مراحل پیش‌پردازش)-(بر حسب ثانیه)

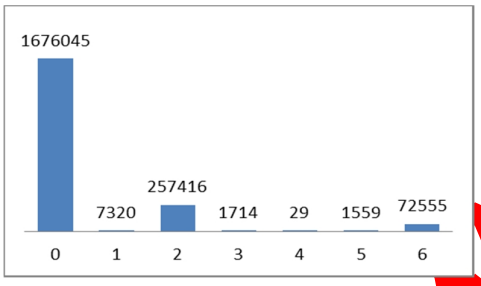
الگوریتم	LR	DT	KNN	RF	GB	NN	RNN	GRU	LSTM
زمان آموزش	۳۱۶	۵۰	۴۲۶	۱۴۷	۱۶۰۵	۴۱۸	۱۰۰۵	۱۸۶۵	۲۸۷۸
زمان آزمایش	۰/۳	۰/۶۷	۳۹۳	۰/۲۹	۱۳/۶	۲۰	۳۳	۴۷	۴۸

را نشان می‌دهد. مقادیر این جدول زمان آموزش و آزمایش هر الگوریتم را بدون انجام انتخاب ویژگی و متوازن‌سازی مشان می‌دهد و صرفاً عملیات پیش‌پردازش بر روی داده‌ها اعمال شده است.

مطابق جدول شماره (۵)، زمان آموزش جنگل تصادفی ۱۴/۷ ثانیه و زمان آزمایش آن ۰/۲۹ ثانیه می‌باشد. لازم به ذکر است که در هر الگوریتم، برای رسیدن به بالاترین مقدار، آرگومان‌ها به‌صورت آزمون‌وخطا بارها تکرار و بررسی شده و نتیجهٔ مشاهده شده، مربوط به بهترین آرگومان بوده است. جدول شماره (۶) میانگین مقادیر دقت، بازخوانی و F1-Score الگوریتم‌های یادگیری را نشان می‌دهد. همان گونه که در جدول شماره (۶) مشاهده می‌شود، برخی الگوریتم‌ها مقدار دقت، بازخوانی و F1-Score بسیار کمتری داشته‌اند. این امر به دلیل نامتوازن بودن مجموعه‌داده CICIDS-2017 می‌باشد. تعداد مشاهدات

جدول ۶: مقادیر میانگین F1-Score ، بازخوانی و دقت الگوریتم‌ها با داده‌های اولیه(پس از مراحل پیش‌پردازش)

الگوریتم	میانگین F1-Score	میانگین بازخوانی	میانگین دقت
LR	۰/۲۴۷	۰/۲۵۳	۰/۲۴
DT	۰/۵۳۷	۰/۴۴۷	۰/۵۵۷
Knn	۰/۸۸	۰/۸۲۵	۰/۹۴۳
RF	۰/۸۶۸	۰/۸۵۷	۰/۸۸۸
GB	۰/۷۵	۰/۷۵۶	۰/۷۸۱
NN	۰/۱۳	۰/۱۴۳	۰/۱۱۸
RNN	۰/۲۳۸	۰/۲۳	۰/۳۷۴
GRU	۰/۱۹	۰/۱۷۷	۰/۴۱۳
LSTM	۰/۱۳۶	۰/۱۳۶	۰/۱۱۹

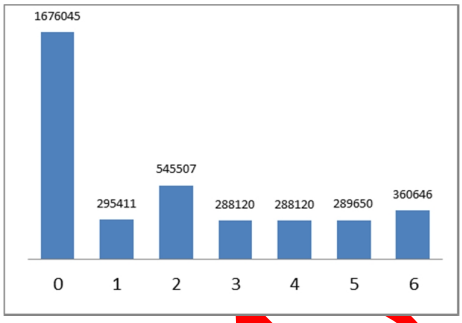


نمودار ۱: مقایسه تعداد مشاهدات هر کلاس در مجموعه y_train مربوط به داده‌های اولیه

در هر کلاس در نمودار شماره (۱) مشاهده می‌شود.

به‌طوری‌که در نمودار شماره (۱) مشاهده می‌شود تعداد کلاس‌ها، اختلاف بسیار زیادی باهم دارند. به‌عنوان‌مثال برچسب شماره چهار تعداد ترافیک مشاهده شده از آن در مجموعه y_train بالغ بر ۲۹ حمله بوده است که در مقابل تعداد ترافیک نرمال، ناچیز به‌حساب می‌آید؛ لذا نتیجه گرفته می‌شود که مقدار صحت مربوطه در جدول شماره (۴)، بیشتر از یادگیری ترافیک نرمال حاصل شده است. به‌منظور منطقی کردن مقادیر نتایج جهت انتخاب بهترین روش، از نمونه‌گیری افزایشی و وزن‌دهی کلاس‌ها استفاده شده است.

در روش نمونه‌گیری افزایشی، تعداد نمونه‌های مشابه از کلاس‌های اقلیت تولید و به مجموعه آموزش y_train اضافه می‌شود تا کلاس‌های اقلیت نیز بتوانند در آموزش سیستم سهم ایفا نمایند. در این پژوهش نتایج الگوریتم‌های یادگیری عمیق که مشخصات آنها کاملاً با به میزان میانگین کل مشاهدات، به کلاس‌های اقلیت اضافه شده است. عملیات وزن‌دهی به کلاس‌ها نیز، در هنگام آموزش و بر اساس تعداد کلاس‌ها انجام می‌شود تا سهم کلاس‌های اکثریت از آموزش سیستم، به‌اندازه کلاس‌های اقلیت کاهش پیدا کند. نمودار شماره (۲)، تعداد مجموعه y_train را پس از نمونه‌گیری افزایشی نشان می‌دهد. از سوی دیگر و در ادامه، الگوریتم انتخاب ویژگی سنجاقک نیز، بر روی داده‌ها اعمال



نمودار ۲: مقایسه تعداد مشاهدات هر کلاس در مجموعه y_train پس از نمونه‌گیری افزایشی

گردید. این الگوریتم چنانچه در جدول شماره (۲) مشاهده شد تعداد ویژگی‌ها را از ۷۰ ویژگی به ۳۰ ویژگی، کاهش داده است، نتیجه این تغییرات در جداول شماره (۷) تا (۱۱) ثبت گردیده است.

جدول شماره (۳) مطابقت دارد، با داده‌های جدید در جدول شماره (۸)نشان‌داده‌شده است.

مقایسه نتایج صحت الگوریتم‌ها با داده‌های اولیه (پس از مراحل پیش‌پردازش) و داده‌های نهایی (پس از مراحل انتخاب ویژگی و متوازن‌سازی) در نمودار شماره (۳) نشان‌داده‌شده است. چنانچه ملاحظه می‌شود، نتایج مربوط به صحت در ID2F از مقدار ۹۹/۸۰ به ۹۹/۸۳ افزایش‌یافته است.

همچنین مقادیر زمان اجرای الگوریتم‌ها در جدول شماره (۹) ثبت شده است.

نتایج F1-Score، دقت و بازخوانی الگوریتم‌ها پس از اعمال تغییرات بر روی مجموعه‌داده به‌صورت جدول شماره (۱۰) بوده است.

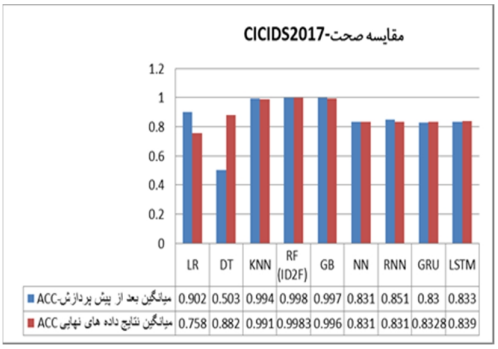
چنانچه در جدول شماره (۱۰) ملاحظه می‌شود، مقادیر F1-Score، بازخوانی و دقت الگوریتم‌ها مقادیر بالاتری را نسبت به جدول شماره (۶) به خود اختصاص داده‌اند. نمودار شماره (۴) مقایسه نتایج به‌دست‌آمده از مقادیر دقت الگوریتم‌های اجرا شده با داده‌های اولیه(پس از مراحل پیش‌پردازش) و داده‌های نهایی (پس از مراحل انتخاب ویژگی و متوازن‌سازی) را نشان می‌دهد. چنانچه از نمودار شماره (۴) بر می‌آید، مقدار دقت در روش ID2F نسبت به داده‌های قبل از متوازن‌سازی و انتخاب ویژگی بهبودیافته و نسبت به سایر

جدول ۷: نتایج صحت الگوریتم‌های یادگیری پس از متوازن‌سازی و انتخاب ویژگی

الگوریتم	LR	DT	KNN	**RF (ID2F)**	GB
ACC	۰/۷۵۸۰	۰/۸۸۲۵	۰/۹۹۱۹	**۰/۹۹۸۳**	۰/۹۹۵۷

جدول ۸: نتایج صحت الگوریتم‌های عمیق پس از متوازن‌سازی و انتخاب ویژگی

الگوریتم	NN	RNN	GRU	LSTM
ACC	۰/۸۳۱۱	۰/۸۳۱۱	۰/۸۳۲۸	۰/۸۳۹۷



نمودار ۳: مقایسه نتایج صحت الگوریتم‌ها با داده‌های اولیه (پس از مراحل پیش‌پردازش) و داده‌های نهایی

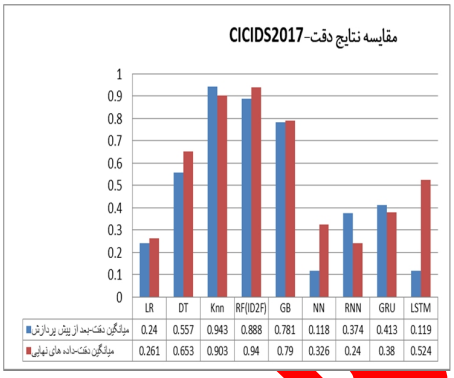
روش‌ها نیز مقدار بالاتری را به خود اختصاص داده است. نمودار شماره (۵) نتایج به‌دست‌آمده از مقادیر بازخوانی الگوریتم‌ها را قبل و بعد از مراحل متوازن‌سازی و انتخاب ویژگی نشان می‌دهد. چنانچه در نمودار شماره (۵) دیده می‌شود، مقدار بازخوانی نیز در روش ID2F نسبت به سایر روش‌ها بالاتر بوده است. در ضمن مقدار بازخوانی در این روش

جدول ۹: نتایج زمان یادگیری پس از متوازن‌سازی و انتخاب ویژگی(بر حسب ثانیه)

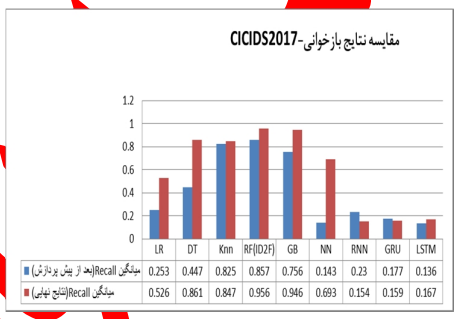
الگوریتم	LR	DT	KNN	RF (ID2F)	GB	NN	RNN	GRU	LSTM
زمان آموزش	۲۰۶	۴۷/۶	۱۱۴۷	۹/۹۵	۸۶۸۷	۹۱۵	۱۱۰۸	۱۸۳۰	۳۰۶۲
زمان آزمایش	۰/۱۲	۰/۷	۳۰۴	۰/۱۴	۸/۴	۱۸	۱۶	۱۹	۳۲

جدول ۱۰: میانگین نتایج F1-Score، بازخوانی و دقت الگوریتم‌های یادگیری با داده‌های نهایی

الگوریتم	میانگین F1-Score	میانگین بازخوانی	میانگین دقت
LR	۰/۲۲۷	۰/۵۲۶	۰/۲۶۱
DT	۰/۶۴۹	۰/۸۶۱	۰/۶۵۳
KNN	۰/۸۷	۰/۸۴۷	۰/۹۰۳
ID2F	**۰/۹۴۶**	**۰/۹۵۶**	**۰/۹۴**
GD	۰/۸۳۶	۰/۹۴۶	۰/۷۹
NN	۰/۳۵۹	۰/۶۹۳	۰/۳۲۶
RNN	۰/۱۴۹	۰/۱۵۴	۰/۲۴
GRU	۰/۱۵۷	۰/۱۵۹	۰/۳۸
LSTM	۰/۱۶۷	۰/۱۶۷	۰/۵۲۴



نمودار ۴: مقایسه نتایج دقت الگوریتم‌ها با داده‌های اولیه (پس مراحل از پیش‌پردازش) و داده‌های نهایی



نمودار ۵: مقایسه نتایج بازخوانی الگوریتم‌ها با داده‌های اولیه (پس مراحل از پیش‌پردازش) و داده‌های نهایی

نسبت به داده‌های اولیه رشد داشته است. در نمودار شماره (۶) مقادیر F1-Score الگوریتم‌ها قبل و بعد از مراحل متوازن‌سازی و انتخاب ویژگی مشاهده می‌شود. همان گونه که دیده می‌شود، مقدار F1-Score نیز در روش ID2F نسبت به سایر روش‌ها بالاتر می‌باشد و این روش نسبت به داده‌های اولیه، افزایش داشته است. زمان اجرای الگوریتم‌ها در نمودار شماره (۷) مورد مقایسه قرار گرفته است.

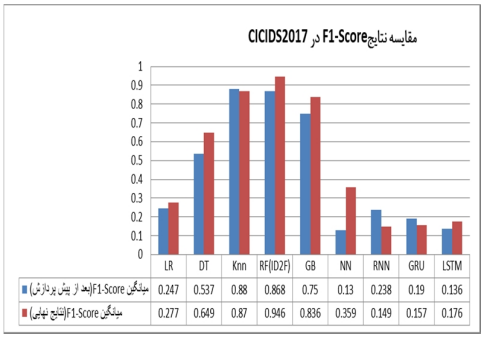
چنانچه در نمودار شماره (۷) مشخص می‌باشد، زمان آموزش روش ID2F نسبت به بقیه الگوریتم‌ها کمتر بوده و نسبت به قبل کاهش‌یافته است.

۴-۳- مقایسه نتایج ID2F با پژوهش‌های پیشین

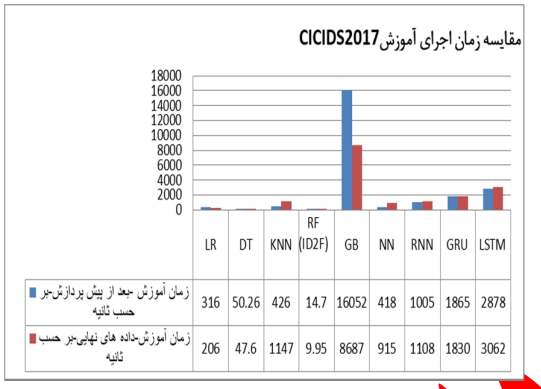
چنانچه در بخش دوم ذکر شد، در مطالعات مختلف از روش‌های متفاوتی برای کشف نفوذ استفاده شده است. به‌منظور مقایسه این پژوهش و روش ID2F با روش‌هایی که محققین پیشین آزمایش نموده‌اند، برخی از این پژوهش‌ها انتخاب و مورد مقایسه قرار گرفته‌اند.

در این انتخاب، سعی شده برخی معیارها مانند چاپ در نشریات معتبر (حتی‌الامکان نشریات مرتبط با امنیت)، استفاده از مجموعه‌داده‌های مشابه و جدید و به‌روز بودن مطالعه، رعایت شود. براین‌اساس، از بخش دوم این پژوهش، مقالاتی که دارای مجموعه داده مشابه بوده‌اند انتخاب و مورد بررسی قرار گرفته‌اند.

در منبع [۱۰] پژوهشگران از الگوریتم KNN به‌صورت ۷ کلاس استفاده کرده و نتایج را با روش‌های مختلف مقایسه کرده‌اند



نمودار ۶: مقایسه نتایج F1-Score الگوریتم‌ها با داده‌های اولیه (پس مراحل از پیش‌پردازش) و داده‌های نهایی



نمودار ۷: مقایسه زمان اجرای الگوریتم‌ها با داده‌های اولیه (پس مراحل از پیش‌پردازش) و داده‌های نهایی

به‌طوری‌که حداکثر درصد صحت ۹۹/۸۲، بازخوانی ۹۵/۷۰، دقت ۹۴/۸۹ و F1-Score برابر با ۹۵/۳۳ بوده است. چنانچه دیده می‌شود، مقدار صحت در مقایسه با نتایج ID2F، کمتر و سایر موارد مقادیر بیشتر بوده است.

در مقاله [۱۱] نیز پژوهشگران روش یادگیری شبکه‌های عصبی عمیق را پیشنهاد داده‌اند. این پژوهش که از چند مجموعه به همراه مجموعه‌داده CICIDS2017 استفاده کرده بهترین نتایج را در روش DNN با ۵ لایه یافته است. نتیجه این پژوهش حداکثر درصد صحت ۹۸/۸برای حملات وب اعلام شده که نسبت به صحت روش پیشنهادی، مقدار پایین‌تری می‌باشد. همچنین در مقاله [۳] پژوهشگران از روش شبکه عصبی عمیق بر روی مجموعه‌داده CICIDS2017 استفاده کرده‌اند. نتیجه پژوهش نشان می‌دهد حداکثر درصد صحت در حالت هفت کلاس با انتخاب ویژگی ۹۹/۵۷و بدون انتخاب ویژگی ۹۹/۵۵ بوده است. این مقدار نیز نسبت به مقدار مربوطه در ID2F کمتر می‌باشد.

در منبع [۱۲] پژوهشگران از مجموعه‌داده CICIDS2017 و چند مجموعه دیگر استفاده کرده و روش BLS [۱۷] را بکار برده‌اند. بهترین نتیجه پژوهش بر روی مجموعه‌داده فوق‌الذکر، صحت ۹۶/۶۳ به‌دست‌آمده که نسبت به مقدار صحت در روش پیشنهادی پایین‌تر می‌باشد.

در مقاله [۱۳] که از الگوریتم درخت تصمیم و مدل‌های مبتنی بر قوانین استفاده شده، از ترکیب سه الگوریتم REP Tree، JRip و Forest PA استفاده و درصد صحت را برابر ۹۶/۶۶۵ و مقدار بازخوانی ۹۴/۴۵۷ گزارش گردیده است. مقاله [۱۵] که محقق از مجموعه‌داده CICIDS2017 استفاده نموده، الگوریتم DNN بکار گرفته شده و نتایج آن عبارتند از: حداکثر صحت ۹۹/۹۵ و دقت ۹۴/۳۱ و بازخوانی ۹۵/۶۲و F1-Score برابر ۹۴/۱بوده است. در مقایسه با روش پیشنهادی در این مقاله، مقدار F1-Score از مقدار ID2F کمتر و در مورد بقیه موارد مقدار بیشتری داشته است.

در [۱۵] که در ۲۰۲۱ منتشر شده است، محقق از روش شبکه عصبی عمیق مبتنی بر پشته بر روی مجموعه‌دادهNSL-KDD CICIDS2017 و یک مجموعه دیگر استفاده نموده و به ترتیب مقادیر ۸۹/۹۷ و ۹۹/۶۵ را کسب نموده است. چنانچه می‌توان مشاهده کرد، مقادیر ID2F در هر دو مجموعه‌داده بیشتر بوده است.

همچنین در مقاله [۲۳] برای تشخیص نفوذ از GRU استفاده شده است. در این مقاله که در سال ۲۰۲۰ انجام گردیده، از مجموعه‌داده CICIDS2017 برای بررسی بهره‌برداری‌شده و نتایج آن به ترتیب ۸۲/۲۸، ۸۵/۲۸ و ۷۸/۱۴ را برای دقت، بازخوانی و F1-Score بوده است که باتوجه‌به مقادیر پژوهش حاضر، مقادیر پایین‌تری می‌باشند.

در منبع [۲۴] از روش شبکه باور عمیق و SVM استفاده نموده و درصد نتایج به‌دست‌آمده در آن برای صحت، دقت، بازخوانی و F1-Score به ترتیب مقادیر ۹۶/۴۹، ۹۰/۴۰، ۹۵/۶۵ و ۹۲/۹۵ می‌باشد. در این پژوهش از مجموعه‌داده CICIDS2017 برای تحلیل استفاده شده است. چنانچه مشاهده می‌شود، مقدار بازخوانی به‌دست‌آمده نسبت به ID2F بیشتر و در خصوص سایر موارد، پایین‌تر می‌باشد.

چنانچه در جدول (۱۱) مشاهده می‌شود، از بین نه مقاله معتبر که در سال‌های اخیر چاپ شده‌اند، در تعداد هشت مقاله، مقدار صحت به‌دست‌آمده، پایین‌تر از روش ID2F بوده و فقط در مقاله [۱۵] مقدار بالاتری بدست آمده است.

۵- نتیجه‌گیری و پیشنهادها

مدل ID2F (پیشنهادشده در این پژوهش)، به‌عنوان یک مدل تشخیص نفوذ با استفاده از الگوریتم سنجاقک به‌عنوان انتخاب ویژگی و الگوریتم جنگل تصادفی به‌منظور دسته‌بندی می‌باشد. این روش بر روی دو مجموعه‌داده معتبر و متفاوت اعمال شده و با چندین روش شاخص یادگیری ماشین و یادگیری عمیق مقایسه گردیده است.

نتایج معیارهای ارزیابی‌ها نشان می‌دهند که سیستم پیشنهادی، معیارهای صحت، دقت، بازخوانی و F1-Score بالاتری نسبت به سایر الگوریتم‌های اجرا شده در این پژوهش داشته و زمان کمتری نیز در اجرای سیستم صرف نموده است. همچنین مطابق جداول شماره (۲) تا(۱۰)، پیش از اعمال انتخاب ویژگی و متوازن‌سازی، معیار صحت برابر

جدول ۱۱: مقایسه نتایج پژوهش با نتایج پژوهش‌های پیشین بر روی مجموعه‌داده‌های مشابه

شماره ارجاع	روش یادگیری	سال ارائه	مجموعه‌داده	معیار ارزیابی	مقدار (درصد)
[۱۰]	KNN	۲۰۱۹	CICIDS2017	صحت	۹۹/۸۲
				دقت	۹۴/۸۹
				بازخوانی	۹۵/۷۰
				F1-Score	۹۵/۳۳
[۱۱]	DNN	۲۰۱۹	CICIDS2017	صحت	۹۶/۲
[۳]	DNN و انتخاب ویژگی AE	۲۰۱۹	CICIDS2017	صحت	۹۹/۵۷
[۱۲]	BLS	۲۰۲۰	CICIDS2017	صحت	۹۶/۶۳
[۱۳]	DT	۲۰۲۰	CICIDS2017	صحت	۹۶/۶۶۵
				بازخوانی	۹۴/۴۵۷
[۱۵]	DNN	۲۰۱۹	CICIDS2017	صحت	۹۹/۹۵
				دقت	۹۴/۳۱
				بازخوانی	۹۵/۶۲
				F1-Score	۹۴/۱
[۲۲]	Stacking Based DNN	۲۰۲۱	NSL-KDD	صحت	۸۹/۹۷
			CICIDS2017	صحت	۹۹/۶۵
[۲۳]	GRU	۲۰۲۰	CICIDS2017	دقت	۸۲/۲۸
				بازخوانی	۸۵/۲۸
				F1-Score	۷۸/۱۴
[۲۴]	DBN و SVM	۲۰۱۸	CICIDS2017	صحت	۹۶/۴۹
				دقت	۹۰/۴۰
				بازخوانی	۹۵/۶۵
				F1-Score	۹۲/۹۵
ID2F (مدل پیشنهادی)	**جنگل تصادفی و انتخاب ویژگی سنجاقک**		**CICIDS2017**	**صحت**	**۹۹/۸۳**
				دقت	**۹۴**
				بازخوانی	**۹۵/۶**
				F1-Score	**۹۴/۶**
			KDD99	**صحت**	**۹۹/۸۵**
				دقت	**۹۰**
				بازخوانی	**۸۷/۶**
				F1-Score	**۸۸/۸**

با ۰/۹۹۸بوده و معیارهای دقت، بازخوانی و F1-Score، به ترتیب دارای میانگین ۰/۸۸۸ و ۰/۸۵۷ و ۰/۸۶۸ بوده‌اند که در روش پیشنهادی ID2F با اعمال انتخاب ویژگی و متوازن‌سازی، معیار صحت به مقدار ۰/۹۹۸۳ارتقاء یافته و مقادیر میانگین دقت، بازخوانی وF1-Score نیز به ترتیب مقادیر ۰/۹۴و ۰/۹۵۶و ۰/۹۴۶ را کسب نموده‌اند. چنانچه ملاحظه می‌شود تمامی مقادیر معیارها نسبت به مقادیر قبل از اعمال روش، ارتقاء یافته‌اند و سرعت اجرای الگوریتم نیز در روش پیشنهادی بهبود داشته است. همچنین باتوجه‌به جدول شماره (۱۱)، صحت در

روش پیشنهاد شده، نسبت به اکثر مطالعات پیشین که در این پژوهش انتخاب شده‌اند، بهبود یافته است و صرفاً مقدار صحت منبع [۱۵] نسبت به پژوهش پیشنهادی بالاتر بوده است. در نهایت، باتوجه‌به مقادیر مناسب کسب شده در روش پیشنهادی نسبت به اکثر پژوهش‌های دیگر، پیشنهاد می‌شود برای تحقیقات آینده، روش‌های پیش‌بینی اهداف مهاجم و پیشنهاد یا اعمال تغییرات لازم در شبکه به‌منظور کاهش صدمات، موردمطالعه قرار گرفته و به این روش اضافه گردد.

مراجع:

[1] S. Choudhary and N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT". Procedia Computer Science, 2020. 167: p. 1561-1573.

[2] A. Javaid, et al., "A deep learning approach for network intrusion detection system". Eai Endorsed Transactions on Security and Safety, 2016. 3(9): p. e2.

[3] O. Faker, and E. Dogdu, "Intrusion detection using big data and deep learning techniques", Proceedings of the 2019 ACM Southeast Conference. 2019.

[4] Y.B. Bhavsar, and K.C. Waghmare, "Intrusion detection system using data mining technique: Support vector machine", International Journal of Emerging Technology and Advanced Engineering, 2013. 3(3): p. 581-586.

[5] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems",Neural Computing and Applications, 2016. 27(4): p. 1053-1073.

[6] H. Hindy, et al., "A taxonomy and survey of intrusion detection system design techniques", network threats and datasets. 2018.

[7] R. Yahalom, et al., "Improving the effectiveness of intrusion detection systems for hierarchical data". Knowledge-Based Systems, 2019. 168: p. 59-69.

[8] P. Bedi, N. Gupta, and V. Jindal, "Siam-IDS: Handling class imbalance problem in intrusion detection systems using siamese neural network". Procedia Computer Science, 2020. 171: p. 780-789.

[9] L. Dhanabal, and S. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms". International journal of advanced research in computer and communication engineering, 2015. 4(6): p. 446-452.

[10] K.A. Tait, , et al., "Intrusion Detection using Machine Learning Techniques: An Experimental Comparison". arXiv preprint arXiv:2105.13435, 2021.

[11] R. Vinayakumar, et al., "Deep learning approach for intelligent intrusion detection system". IEEE Access, 2019. 7: p. 41525-41550.

[12] A.L.G. Rios, et al. "Detection of denial of service attacks in communication networks". in 2020 IEEE International Symposium on Circuits and Systems (ISCAS). 2020. IEEE.

[13] A. Ahmim, et al. "A novel hierarchical intrusion detection system based on decision tree and rules-based models". in 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS). 2019. IEEE.

[14] W.A.H. Ghanem, et al., "An efficient intrusion detection model based on hybridization of artificial bee colony and dragonfly algorithms for training multilayer perceptrons". IEEE Access, 2020. 8: p. 130452-130475.

[15] P. Toupas, et al. "An intrusion detection system for multi-class classification based on deep neural networks". in 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). 2019. IEEE.

[16] A. Boukhamla ,and J.C. Gaviro, "CICIDS2017 dataset: performance improvements and validation as a robust intrusion detection system testbed". International Journal of Information and Computer Security, 2021. 16(1-2): p. 20-32.

[17] R. Panigrahi, and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems". International Journal of Engineering & Technology, 2018. 7(3.24): p. 479-482.

[18] S. Panwar, Y. Raiwani, and L.S. Panwar. "Evaluation of network intrusion detection with features selection and machine learning algorithms on CICIDS-2017 dataset". in International Conference on Advances in Engineering Science Management & Technology (ICAESMT)-2019, Uttaranchal University, Dehradun, India. 2019.

[19] M.M. Mafarja, et al. "Binary dragonfly algorithm for feature selection". in 2017 International conference on new trends in computing sciences (ICTCS). 2017. IEEE.

[20] T. Bhavani, M.K. Rao, and A.M. Reddy. "Network intrusion detection system using random forest and decision tree machine learning techniques". in First international conference on sustainable technologies for computational intelligence. 2020. Springer.

[21] M. Pal, "Random forest classifier for remote sensing classification". International journal of remote sensing, 2005. 26(1): p. 217-222.

[22] L. Nkenyereye, B.A. Tama, and S. Lim. "A stacking-based deep neural network approach for effective network anomaly detection". CMC-Computers Materials & Continua, 2021. 66(2): p. 2217-2227.

[23] I. Kurochkin, and S. Volkov, "Using GRU based deep neural network for intrusion detection in software-defined networks". in IOP Conference Series: Materials Science and Engineering. 2020. IOP Publishing.

[24] N. Marir, et al., "Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark". IEEE Access, 2018. 6: p. 59657-59671.

زیرنویس‌ها

[1] Firewall
[2] Self-taught learning
[3] Soft-max Regression
[4] Intrusion Detection based on Dragonfly
[5] NP-Hard
[6] Meta-heuristic Algorithms
[7] CIC Flow Meter
[8] Port scan
[9] DOS
[10] SQL injection
[11] HTTP Unbearable Load King
[12] GoldenEye
[13] Infiltration of the network from inside
[14] Botnet
[15] Accuracy
[16] Precision
[17] Broad learning system