TEEGES

Technovations of Electrical Engineering in Green Energy System

Research Article (2025) 3(4):75-97

A New Approach to Detecting Intrusion and Malicious Behaviors in Big Data

Homa Movahednezhad[1,2], *Assistant Professor*, Mohsen Porshaban[1,2], *PhD Student*, Ehsan Yazdani Chamzini[1,2], *PhD Student*, Elahe Hemati Ashani[1,2], *PhD Student*, Mahdi Sharifi[1,2], *Assistant Professor*

[1] *Faculity of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran.*
[2] *Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad, Iran*

Abstract:

Today, maintaining information security and intrusion detection is very important to deal with malicious behaviors in massive data. In this article, a hybrid method for detecting malicious data is presented wherein three factors of time progress, history of users and scalability are taken into account. The proposed method utilizes storage and feature extraction techniques to increase the speed and reduce the amount of calculations. In addition, the support vector machine algorithm has been modified for classification, and the parallelized bacterial foraging optimization algorithm has been used for feature extraction. The results show that the proposed algorithm outperforms the existing methods in terms of detection rate by 21%, false positive rate by 62%, accuracy by 15% and execution time by 70%. The reduction in execution time indicates that less energy is needed to run the algorithm which results in saving energy and can be beneficial for use in green energy systems.

Keywords: Intrusion detection, Malicious behavior, Support vector machine algorithm, Big data, Bacterial foraging optimization algorithm.

Received: 03 January 2024
Revised: 03 April 2024
Accepted: 23 April 2024
Corresponding Author: Dr. Mahdi Sharifi, email: m.sharifi@pco.iaun.ac.ir
DOI: 10.30486/TEEGES.2024.904864

فناوری‌های نوین مهندسی برق در سیستم انرژی سبز

TEEGES

مقاله پژوهشی

ارائه یک روش جدید برای تشخیص نفوذ و رفتارهای مخرب در داده‌های حجیم

هما موحد نژاد[۱و۲]، *استادیار*، محسن پورشعبان[۱و۲]، *دانشجوی دکتری* ، احسان یزدانی چمزینی[۱و۲]، *دانشجوی دکتری*، الهه همتی اشنی[۱و۲]، *دانشجوی دکتری*، مهدی شریفی[۱و۲]، *استادیار*

۱- دانشکده مهندسی کامپیوتر، واحد نجف آباد، دانشگاه آزاد اسلامی، نجف آباد، ایران

۲- مرکز تحقیقات کلان داده، واحد نجف آباد، دانشگاه آزاد اسلامی، نجف آباد، ایران

چکیده : امروزه حفظ امنیت اطلاعات و تشخیص نفوذ به‌منظور مقابله با رفتارهای مخرب در داده‌های حجیم از اهمیت بسزایی برخوردار است. در این مقاله یک روش ترکیبی نرم‌افزاری سخت‌افزاری برای تشخیص داده‌های مخرب ارائه شده است. در این روش سه فاکتور پیشرفت زمانی، تاریخچه کاربران و مقیاس‌پذیری لحاظ شده است. در روش پیشنهادی از تکنیک‌های ذخیره‌سازی و استخراج ویژگی جهت افزایش سرعت و کاهش حجم محاسبات استفاده می‌شود. همچنین برای طبقه‌بندی از الگوریتم ماشین بردار پشتیبان تغییر یافته و برای عملیات استخراج ویژگی‌ها از الگوریتم غذایابی باکتری بصورت موازی‌سازی شده، بهره برده شده است. نتایج نشان می‌دهد که الگوریتم پیشنهادی نسبت به سایر روش‌های مشابه، از نظر نرخ تشخیص ۲۱٪، نرخ مثبت کاذب ۶۲٪، دقت ۱۵٪ و زمان اجرا ۷۰٪ بهتر عمل می‌کند. کاهش زمان اجرا بیانگر آن است که برای اجرای الگوریتم به انرژی مصرفی کمتری نیاز است که در نتیجه می‌تواند علاوه بر صرفه‌جویی انرژی؛ جهت بکارگیری در سیستم‌های انرژی سبز نیز سودمند باشد.

واژه های کلیدی: تشخیص نفوذ، رفتارهای مخرب، الگوریتم ماشین بردار پشتیبان، کلان داده‌ها، الگوریتم غذایابی باکتری.

تاریخ ارسال مقاله: ۱۴۰۲/۱۰/۱۳

تاریخ بازنگری مقاله: ۱۴۰۳/۰۱/۱۵

تاریخ پذیرش مقاله: ۱۴۰۳/۰۲/۰۴

نویسنده‌ی مسئول: دکتر مهدی شریفی، m.sharifi@pco.iaun.ac.ir

DOI: 10.30486/TEEGES.2024.904864

۱- مقدمه

به دلیل تنوع سرویس‌های دیجیتال و رشد تکنولوژی هر بخش از سیستم در معرض حمله داده‌های مخرب قرار دارد. با توجه به مقیاس، تنوع و سرعت داده‌های مخرب، نرم افزارهای دفاع کننده باید با استفاده از یادگیری ماشین قادر باشند تا حمله‌ها را تشخیص دهند. اولین تشخیص داده‌های مخرب برای تشخیص نفوذ حدود ۴۰ سال پیش توسط دنور انجام پذیرفت [۱]. امروزه تشخیص داده‌های مخرب شبکه سخت‌تر و پیچیده‌تر شده است. ولی مساله یافتن یک راه حل مناسب برای حجم بالای داده‌های شبکه هنوز به‌وجود نیامده است[۲].

تحقیقاتی که در این زمینه انجام گرفته است می‌توان به مطالعات موجود در[۵-۳] اشاره کرد که برای تشخیص داده‌های مخرب از یادگیری ماشین با زبان جاوا استفاده کرده‌اند و همچنین در تحقیقاتی دیگر می‌توان به داده‌های موبایل [۶] تشخیص داده‌های میزکار [۷] تشخیص نفوذ شبکه [۸] تشخیص اسپم [۹] و تشخیص آدرس‌های جعلی [۱۰] اشاره نمود.

بر خلاف کاربردهای دیگر، یادگیری ماشین که برای تشخیص استفاده می‌شوند مانند تشخیص متن یا چهره که در آنها شکل‌ها و کاراکترهای ثابتی وجود دارد و تشخیص بر اساس آنها انجام می‌شود، داده‌های مخرب الگوی ثابتی ندارد و نیاز به تلاش بیشتری برای شناسایی وجود دارد [۱۱]. در واقع این نوع جستجو باید به صورت آنلاین باشد و در هر لحظه به‌روزرسانی انجام دهد تا بتواند الگوهای خود را بسازد. این امر باعث ایجاد یک تاخیر می‌شود و همین تاخیر می‌تواند باعث شود دقت تشخیص داده‌های مخرب پایین بیاید. سیستم‌های تشخیص نفوذ برای کمک به مدیران امنیتی سیستم در جهت کشف نفوذ و حمله به کار گرفته شده است [۱۲]. هدف این سیستم‌ها تنها جلوگیری از حمله نمی‌باشد بلکه کشف و شناسایی حملات و تشخیص اشکالات امنیتی در سیستم یا شبکه‌های کامپیوتری و اعلام به مدیر است [۱۳]. سیستم‌های تشخیص نفوذ در کنار دیواره‌های آتش و به صورت مکمل امنیتی مورد استفاده قرار می‌گیرد. برخی از فواید این سیستم‌ها شامل کارایی بیشتر در تشخیص نفوذ، منبع دانش کاملی از حملات، توانایی رسیدگی به حجم زیادی از اطلاعات، توانایی هشدار نسبتاً بلادرنگ که باعث کاهش خسارت می‌شود، دادن پاسخ های خودکار مانند قطع ارتباط کاربر، افزایش میزان بازدارندگی، توانایی گزارش دهی می‌باشد [۱۴].

از تکنیک‌های دیگر می‌توان به معیارهای آماری اشاره نمود. در نوع پارامتریک مشخصات جمع شده بر اساس یک الگوی خاص در نظر گرفته می‌شود و بر اساس مقادیری که با تجربه حاصل شده الگو ساخته می‌شود، مقایسه صورت می‌گیرد. در این روش نیز به دلیل پارامتریک بودن بسیاری از داده‌های مخرب را نمی‌تواند شناسایی کند و دقت مورد نظر را نمی‌تواند داشته باشد [۱۵].

از معیارهای دیگر می‌توان به معیارهای آماری غیر پارامتریک اشاره نمود که داده‌های مشاهده شده را بر اساس الگوهای استفاده شده مشخصی به طور قابل قبول تعریف می‌کند. اما با الگوهایی که به عنوان قانون مشخص شده فرق دارد و به صورت شمارشی نمی‌باشد. متاسفانه در این تکنیک‌ها ایجاد تعداد زیادی هشدار نادرست می‌شود. زیرا الگوهای رفتاری از جانب استفاده‌کنندگان و سیستم بسیار متفاوت است [۱۶]. سیستم‌های تشخیص مبتنی بر امضا به میان آمدند که قادر به کشف حملات جدید هستند. در این تکنیک‌ها الگوهای نفوذ از پیش ساخته شده به صورت قانون نگهداری می‌شود به طوریکه هر الگو انواع مختلفی از یک نفوذ خاص را در بر گرفته است و در صورت بروز چنین الگویی در سیستم وقوع نفوذ اعلام می‌شود. در این روشها معمولا تشخیص‌دهنده دارای پایگاه داده‌ای از امضاها یا الگوهای حمله می‌باشند که سعی می‌کنند با بررسی ترافیک شبکه الگوهای مشابه با آنچه را در پایگاه داده خود نگهداری می‌کنند بیابد. این دسته از روش‌ها تنها قادر به تشخیص نفوذهای شناخته شده می‌باشند و در صورت بروز حملات جدید در سطح شبکه نمی‌توانند آنها را شناسایی کنند و مدیر شبکه باید همواره الگوی حملات جدید را به سیستم تشخیص نفوذ اضافه نماید [۱۷].

با گسترش شبکه‌های کامپیوتری، حفظ امنیت این شبکه‌ها نیز به یک اولویت بسیار مهم درآمده است. سیستم‌های تشخیص نفوذ به عنوان بخشی اساسی از زیرساخت‌های امنیتی شبکه‌ها شناخته می‌شوند. این سیستم‌ها به دو دسته تشخیص سوءاستفاده و تشخیص رفتار غیرعادی تقسیم می‌شوند. در دسته اول، تلاش می‌شود حملات با تشخیص الگوهای نفوذ شناسایی شوند، در حالی که رویکرد دوم توسعه‌ی روش‌های قبلی با استفاده از الگوهای رفتار عادی در شبکه را در برمی‌گیرد. در این روش، الگوهای رفتاری نرمال از پیش استخراج شده و نفوذ بر اساس انحراف از این الگوها تعریف می‌شود. در صورت مشاهده حتی کوچکترین انحراف، هشدار نفوذ فعال می‌گردد. از روش‌های مختلف در سیستم‌های تشخیص نفوذ استفاده می‌شود، که یکی از این روش‌ها، داده‌کاوی است. هرچند استفاده از سیستم‌های تشخیص نفوذ به عنوان یکی از راهکارهای اصلی برای افزایش امنیت در شبکه‌های کامپیوتری مورد استفاده قرار

می‌گیرد. با این حال، راهکارهای امروزی برای تشخیص نفوذ با چالش‌های انتخاب مدل‌هایی با قابلیت اطمینان بیشتر در تشخیص نفوذ مواجه شده‌اند. بر همین اساس موضوع تشخیص نفوذ بعنوان یکی از چالش‌های اساسی در حوزه امنیت مطرح شده است. به منظور حل این چالش، در این تحقیق یک راهکار تشـــخیص نفوذ ترکیبی با اســـتفاده از الگوریتم غذایابی باکتری (BFO) الگوریتم ماشـــین بردار پشتیبان (SVM) ارائه شده است. در واقع از این راهکار ترکیبی در جهت ارتقاء دقت تشخیص نفوذ در شبکه بهره برده شده است. با بهره‌گیری از این راهکار، ویژگی‌های مهم داده‌ها استخراج شده و از آن برای بهبود عملکرد کلی راهکار و طبقه‌بندی استفاده می‌شود. به کمک این رویکرد، کاهش ویژگی می‌تواند منجر به افزایش کارایی و دقت شود. در ادامه این بخش، جزئیات این روش تشریح شده است. ادامه مقاله به این صورت سازماندهی شده است. در بخش ۲ کارهای پیشین بحث شده است. بخش ۳ به مدل پیشنهادی پرداخته است. ارزیابی و کارایی در بخش ۴ آمده است و نهایتا در بخش ۵ نتیجه‌گیری و کارهای آینده مشخص شده است.

۲- کارهای پیشین

موضوع امنیت در داده های شبکه همواره یکی از موضوعات پیچیده و چالش برانگیز بوده است که به دلیل ماهیت آنها باید از ابزارها و روش های مناسب و کارآمد برای حفظ امنیت کاربران و اطلاعات آنها استفاده کرد. یکی از موثرترین روش‌هایی که در سال‌های اخیر بسیار مورد توجه فعالان امنیت شبکه قرار گرفته است، استفاده از سیستم‌های تشخیص نفوذ و ارتقای کارایی آن‌ها در مواجهه با حملات مزاحم است. برای داشتن یک IDS کارآمد، با توجه به ماهیت شبکه، یکی از اساسی‌ترین عناصر و ویژگی‌هایی که در تحلیل داده‌های شبکه باید مورد توجه قرار گیرد، داشتن ماهیت داده‌های حجیم این داده‌ها است. به طور دقیق‌تر، واضح است که ترافیک اینترنت به طور کلی در سال‌های اخیر در جامعه مدرن رشد کرده است و ما انتظار داریم این روند ادامه یابد بر همین اساس در [۱۴] یک چارچوب تشخیص نفوذ برای داده‌های حجیم ارایه شده است این راهکار بر مبنای شبکه‌ی بیزی به کمک روش بسته‌بندی[۱] می‌باشد. در این مقاله، چارچوبی، مبتنی بر دوفاز برای تشخیص نفوذ در شبکه، ارائه شده است. در فاز اول، یک روش بسته‌بندی برای انتخاب ویژگی مبتنی بر الگوریتم ژنتیک مورد استفاده قرار گرفته است. دلیل استفاده از روش بسته بندی در این پژوهش، دقت بالاتر آن نسبت به سایر روش‌های انتحاب ویژگی، نظیر روش فیلتر، بیان شده است. بعد از ساخت مدل، به کمک مجموعه داده تست، دقت مدل ساخته شده بر روی ویژگی‌های انتخابی مورد بررسی قرار گرفته است. این روند تا رسیدن به شرط توقف، ادامه می‌یابد. در [۱۵] از داده‌های فاقد برچسب به همراه روش‌های با نظارت، به منظور افزایش دقت سیستم‌های تشخیص نفوذ استفاده شده است. در این روش پیشنهادی، یک شبکه عصبی تک لایه، آموزش داده می‌شود تا توابع عضویت فازی و دسته بندی نمونه‌ها را بر روی داده‌های فاقد برچسب، ایجاد نمایند. در ادامه، هر یک از دسته‌های ایجاد شده با مجموعه داده اولیه، ترکیب شده و آموزش مجدد بر روی روش دسته بندی، اعمال می‌گردد. روش پیشنهادی این تحقیق بر روی مجموعه داده‌ی NSL-KDD اعمال گردیده است. نتایج حاصل از تحقیق نشان دهنده تأثیر مؤثر داده‌های فاقد برچسب متعلق به گروه‌های فازی اول و سوم بر روی دقت دسته بند می‌باشد. در روش پیشنهادی در [۱۶]، تأثیر PCA بر سیستم تشخیص نفوذ بررسی شده است. همچنین تعداد ایده آل مولفه های اصلی مورد نیاز برای تشخیص نفوذ و تأثیر داده‌های آلوده به نویز روی PCA نیز مورد توجه بوده است. مجموعه داده‌های با اندازه اصلی $d \times n$ به ساختاری با k مولفه اصلی معین، نگاشت شده‌اند و به مجموعه داده‌ای با اندازه‌ی $k \times n$ تبدیل یافته‌اند، که n تعداد نمونه‌ها و d تعداد ابعاد اصلی است. k تعداد مولفه های اصلی با محدوده‌ی تغییرات از ۲ تا ۲۰ است. نتایج انجام آزمایشات دقت طبقه بندی برای ۱۰ مولفه اصلی به ترتیب در حدود ۷.۹۹ ٪ و ۸.۹۸ ٪ می‌باشد که تقریباً همان دقت به دست آمده با استفاده از ۴۱ ویژگی از ۳۱۲۷۹ نمونه برای مجموعه داده‌ی KDD و ۲۸ ویژگی از ۳۳۷۴۶ نمونه برای مجموعه داده ISCX می‌باشد. در [۱۷] یک مدل تشخیص نفوذ بر مبنای کاوش قواعد انجمنی به وسیله برنامه‌ریزی ژنتیک ارائه شده است. در این تحقیق، یک روش کاوش قواعد انجمنی فازی بر مبنای برنامه ریزی شبکه ژنتیک (GNP[۲]) برای تشخیص نفوذ در شبکه‌های کامپیوتری پیشنهاد شده است. روش GNP یک روش بهینه‌سازی تکاملی است که به جای استفاده از رشته‌ها از گراف‌های جهت‌دار یا درخت در برنامه‌ریزی ژنتیک استفاده می‌کند که این امر منجر به بهبود قدرت نمایش راه حل با برنامه نویسی کمتر می‌شود. نتایج حاصل از پیاده‌سازی این تحقیق بر روی داده‌های KDD99Cup و DARPA98 مورد بررسی قرار گرفته است و نشان داده شده است که می‌تواند دقت تشخیص نفوذ را تا حد مناسبی بالا ببرد. در [۱۸] روشی تحت عنوان FC-ANN بر مبنای روش‌های شبکه‌ی عصبی مصنوعی[۳] و خوشه‌بندی فازی به منظور تشخیص نفوذ ارائه شده است تا از طریق آن بتوان به نرخ دقت بالاتری در سیستم‌های تشخیص نفوذ دست یافت. در این روش در قدم اول به کمک روش

خوشـــه بندی فازی، زیر مجموعه‌های آموزش مختلفی ایجاد می‌گردد. در ادامه، بر مبنای این زیر مجموعه‌های آموزشـــی ایجاد شـــده، شـــبکه عصـــبی مصـــنوعی مختلفی آموزش می‌بینند تا چندین مدل متفاوت ایجاد گردند. در انتها از یک روش تجمعی فازی برای جمع بندی نتایج حاصـــل از مدل‌های مختلف اســـتفاده می‌شـــود. نتایج این تحقیق بر روی مجموعه داده KDD99Cup مورد ارزیابی قرار گرفته است. در [۱۹] یک سیستم تشخیص نفوذ بر مبنای دسته بندی چندگانه‌ی نایو بیز مخفی (HNB[۴])، به منظور تشخیص نفوذ در شبکه‌های کامپیوتری ارائه شده است. محققان در این تحقیق بیان داشتند که روش HNB قابل اعمال بر روی مسائل تشخیص نفوذ با ابعاد[۵] بالا و ویژگی‌های وابسـته بهم[۶] می‌باشـد. روش HNB یک روش داده کاوی اسـت که فرض‌های موجود در روش نایوبیز[۷] را تعدیل می‌نماید. پیاده سـازی و ارزیابی روش مطرح شـده در این تحقیق برای مجموعه داده KDD نشـان می‌دهد که روش HNB نسـبت به روش نایوبیز برتری قابل توجه ای در زمینه بهبود نرخ دقت و کاهش نرخ خطا دارد. در تحقیق انجام شـــده دئدر [۲۰]، یک مدل تشخیص نفوذ با استفاده از ترکیب انتخاب ویژگی Square-Chi و SVM[۸] چندکلاسه ارائه شده است. بسیاری از سیستم‌های تشخیص نفوذ، تنها از یک الگوریتم طبقه بندی جهت دسـته بندی ترافیک شـبکه بعنوان نرمال و غیرنرمال اسـتفاده می‌کنند. با توجه به میزان زیاد داده‌ها، این مدل از طبقه بندها موفق به دسـتیابی با نرخ تشـخیص حمله بالا و کاهش نرخ هشـدار غلط نمی‌شـوند. با این حال، در این راهکار با اسـتفاده از کاهش ابعاد داده‌ها توانسـته‌اند به یک مجموعه بهینه از ویژگی‌ها بدون از دسـت دادن اطلاعات دسـت یابند و سـپس با اسـتفاده از روش مدل سـازی چندکلاسـه، شـناسـایی حملات شـبکه‌ای متفاوت را طبقه بندی کرده‌اند. در [۲۱] از طریق تلفیق روش‌های ماشـین بردار پشـتیبان، راهکاری برای دسـته‌بندی داده‌ها جهت تشـخیص نفوذ در شـبکه‌های کامپیوتری شـده اسـت. هدف از این روش، دسـته بندی داده‌های نرمال و غیرنرمال با دقت بالا و همچنین کاهش نرخ خطا بیان شـده اسـت. در این مقاله از تلفیق روش ماشـین بردار پشـتیبان و روش خوشـه بندی مبتنی بر شـبکه کلونی مورچه خود سـازمانده[۹] اسـتفاده شـده اسـت. روش مطرح شـده در این مقاله با مجموعه داده KDD99Cup مورد ارزیابی قرار گرفته است و نتایج حاصل از پیاده سازی روش بیانگر این موضوع است که روش تلفیقی مطرح شـده از هر یک از روش‌های ماشـین بردار پشـتیبان و روش شـبکه کلونی مورچه خود سـازمانده برتری دارد. در جدول ۱ خلاصه‌ای از راهکارهای پیشینهٔ تحقیق بیان شده است.

جدول (۱): خلاصه‌ای از راهکارهای پیشینهٔ تحقیق

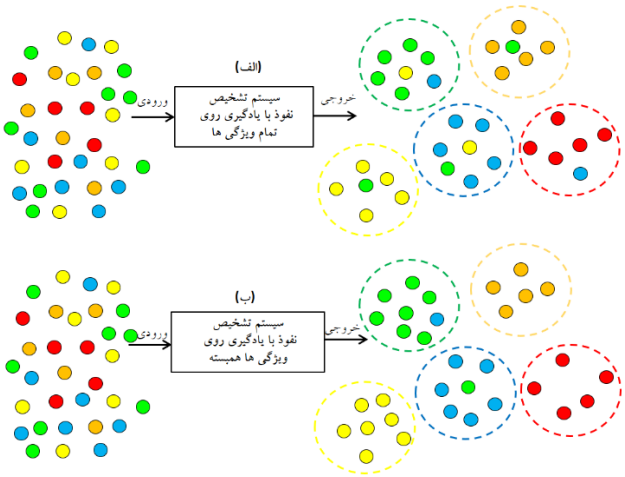
منبع	نوع تشخیص نفوذ	روش داده‌کاوی	نوع یادگیری	مجموعه داده
[۱۴]	مبتنی بر ناهنجاری	شبکه بیزی و ژنتیک	نظارت شده	KDD99Cup
[۱۵]	مبتنی بر امضا	فازی	نیمه‌نظارتی	NSL-KDD
[۱۶]	مبتنی بر امضا	PCA	بدون نظارت	KDDCUP
[۱۷]	مبتنی بر ناهنجاری	فازی	ژنتیک	KDD99Cup
[۱۸]	مبتنی بر امضا	شبکه عصبی و خوشه بندی فازی	نظارت شده	KDD99Cup
[۱۹]	مبتنی بر امضا	دسته بندی چندگانه‌ی نایو بیزی	بدون نظارت	KDD
[۲۰]	مبتنی بر ناهنجاری	دسته‌بندی ماشین بردار پشتیبان	نظارت شده	NSL-KDD
[۲۱]	مبتنی بر ناهنجاری	ماشین بردار پشتیبان و کلونی مورچه	نظارت شده	KDD99Cup

۳- روش پیشنهادی

در این بخش جزئیات مربوط به روش ترکیبی پیشنهادی ارائه می‌شود. جهت بکارگیری راهکار ارائه شده، در شبکه‌ها از مدل پیشنهادی در شـکل ۱ اسـتفاده می‌شـود. در ادامه به بیان جزئیات هر بخش پرداخته می‌شـود. در روش ارائه شـده، قبل از اعمال داده‌ها به مدل پیشنهادی ابتدا فرایند پیش‌پردازش داده‌ها به منظور نرمال‌سازی آنها انجام می‌گیرد. این بخش شامل مراحل زیر می‌باشد:

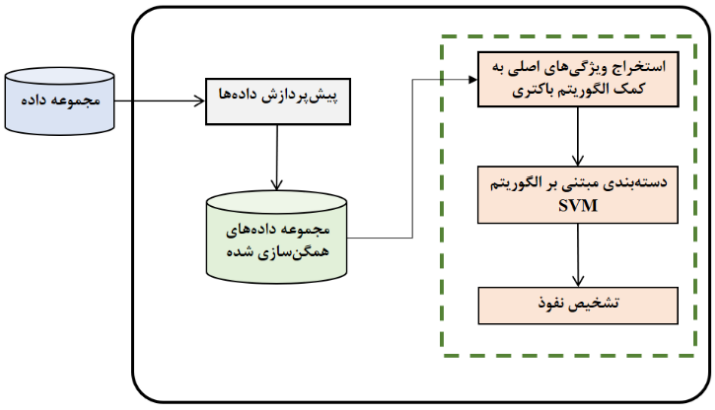
- **همگن‌ســـازی داده‌ها (داده‌های آموزش و آزمون):** در این بخش از طریق جایگذاری نویســـه‌های حرفی مجموعه داده با مقادیر عددی یک نوع همگن‌سازی در سطح داده‌ها انجام می‌گیرد.
- **هنجارســـازی داده‌ها (داده‌های آموزش و آزمون):** در این بخش عدم توازن بین داده‌ها از بین می‌رود. با توجه به آنکه از مجموعه داده KDDcup99 استفاده می‌شود [۲]. بعضی از ویژگی‌ها دارای مقادیر عددی بزرگ هستند که می‌توانند بر دیگر ویژگی‌ها چیره شوند. در این بخش این دسته از ناهنجاری‌ها حذف می‌شود.

در ماژول بعدی عملیات مربوط به الگوریتم باکتری انجام می‌گیرد و از طریق بهره‌گیری از آن، ویژگی‌های مهم داده‌ها استخراج می‌شود تا در نتیجه بتوان عملکرد کلی راهکار و همچنین کارایی کلی راهکار را افزایش یلبد زیرا کاهش ویژگی در نهایت می‌تولند باعث افزایش کارایی شود. در روش‌هایی که فاقد این قابلیت هستند، الگوریتم یادگیری مجبور به استفاده از ویژگی‌هایی است که ارتباط خاصی با نفوذ ندارند و بعبارتی در تعیین حمله نقشی ندارند. این نوع از یادگیری در واقع یادگیری با داده‌های پرت خواهد بود و تاثیر منفی بر روی راهکار تشخیص نفوذ می‌گذارد. در شکل ۱ نمایی از دو حالت بکارگیری فرایند استخراج ویژگی و عدم استفاده از آن نشان داده شده است.



شکل (۱): تفاوت حالت‌های مختلف یادگیری

همانطور که در شکل ۱ نشان داده شده است، در فرایند یادگیری راهکار پیشنهادی، بخش مربوط به طبقه‌بندی بجای استفاده از کلیه داده‌ها، فقط از ویژگی‌های مهمی که توسط الگوریتم غذایابی باکتری ارائه شده است، استفاده می‌کند. این اقدام منجر به افزایش دقت در فرآیند یادگیری می‌شود. از سوی دیگر، حذف ویژگی‌هایی که نقش مهمی در طبقه‌بندی ندارند، تشخیص نفوذ و سرعت یادگیری را نیز افزایش می‌دهد [۳]. در این روش، حذف داده‌های زائد بدون از دست دادن داده‌ها و ویژگی‌های مهم پایگاه داده انجام می‌گیرد. این کاهش داده‌ها، مجموعه‌ای از قواعد تلخیص شده و معنی‌دار را ایجاد می‌کند، که تصمیم‌گیری و یادگیری را بهتر و آسانتر می‌کند. به عبارت دیگر، الگوریتم غذایابی باکتری با کاهش اندازه داده‌ها و انتخاب ویژگی‌های مهم، یک نگاشت از فضای داده‌های خام به فضای مفاهیم ایجاد می‌کند. این رویکرد به این معناست که حمله به تعدادی از مهم‌ترین ویژگی‌های حملات تقسیم می‌شود و باعث افزایش کارایی و دقت تشخیص نفوذ می‌گردد. همچنین، در شکل ۲، شمای کلی هر یک از بخش‌های راهکار پیشنهادی نمایش داده شده است.



شکل (۲): مدل پیشنهادی

در ادامه جزئیات مربوط به نحوه بکارگیری الگوریتم‌های مورد استفاده در این پژوهش تشریح شده است.

۳-۱- الگوریتم باکتری

الگوریتم بهینه‌سازی تجمعی غذایابی باکتری‌ها (BFO) یکی از جدیدترین الگوریتم‌های بهینه‌سازی ایده گرفته شده از طبیعت است. ایده اولیه غذایابی باکتری بر این واقعیت استوار است که در طبیعت، جانداران با روش غذایابی ضعیف احتمال انقراض بیشتری نسبت به جاندارنی با استراتژی غذایابی موفق دارند. پس از نسل های زیاد؛ جاندارن با روش غذایابی ضعیف نابود شده و یا به حالت های بهتر تغییر شکل می‌دهند [۴, ۲۲].

ایده اصلی در طراحی این الگوریتم استفاده از استراتژی غذایابی باکتری ای کویل در بهینه‌سازی چند توابع با چند بهینه بوده است. غذایابی مساله به دو بخش تقسیم می‌شود. بر این اساس در ابتدا مهاجم(غذایابنده) بایستی منبع غذا را یافته و سپس آنرا تعقیب کرده و به گروه حمله می‌کند. اهمیت بخش‌های مختلف این روند به رابطه مهاجم و دسته، بستگی دارد. بعضی دسته‌ها بزرگ هستند پس مهاجم نیاز به انرژی بیشتری برای شکار دارند اما در عوض به سادگی پیدا می‌شوند. در این تحقیق از روش غذایابی گروهی در فرایند غذایابی استفاده شده است. غذایابی گروهی نسبت به غذایابی تکی دارای مزایایی است. در غذایابی گروهی وجود راه ارتباطی میان افراد ضروری است. مزایای غذایابی گروهی در زیر بیان می‌شود:

- عوامل بیشتری به جستجوی غذا هستند پس احتمال یافتن غذا افزایش می‌یابد. وقتی عاملی غذا می‌یابد، می‌تواند گروه را از محل این قضا آگاهی دهد. پیوستن به گروه در این حالت می‌تواند دسترسی به مرکز اطلاعاتی را تامین کرده و به بقای فرد کمک کند.
- امکان بیشتر برای برآمدن از پس گروه‌های بزرگتر غذایی.
- حفاظت در قبال مهاجمان ناخواسته می‌تواند توسط گروه تامین شود.

گاهی مناسب است گروه به عنوان مکانی برای بروز هوش تجمعی درنظر گرفته شود. این هوش تجمعی منجر به غذایابی بهتر برای هریک از اعضای گروه می‌شود زیرا دست آوردها نزاع‌های درون گروهی بر سر غذا را پوشش می‌دهد زیرا در گروه غذای بیشتری نسبت به حالتی که همکاری وجود ندارد بدست می‌آید. در هنگام غذایابی باکتری، حرکت توسط مجموعه از فلاژل‌ها با قابلیت کشش انجام می‌شود. فلاژل‌های این امکان را برای باکتری ای کویل فراهم می‌آورد که چرخیده یا شنا کند. این دو عمل در زمان غذایابی انجام می‌شود. وقتی فالاژل‌های به سمت عقربه‌های ساعت می‌گردند باعث چرخش سلول می‌شود. چرخش در محیط‌های سمی (عدم وجود غذا) و یا هنگامی که غذا یافت شده است به چشم می‌خورد کمتر است. با چرخش فلاژل‌ها عکس عقربه‌های ساعت، باکتری باسرعت قابل توجهی شنا می‌کند. با توجه به این اعمال باکتری علاقمند به یافتن غذای افزاینده و فرار از محیط‌های سمی است. به طور کلی باکتری در محیط‌های دوستانه طولانی تر عمل می‌کند [۵, ۲۳, ۲۴].

در هنگامی که غذای کافی وجود داشته باشد، طول این باکتری افزایش یافته و در دمای مناسب به دو کپی خود تبدیل می‌شود. این عمل باعث بوجود آمدن عمل تولید مجدد در الگوریتم می‌گردد. با توجه به وقوع تغییرات ناگهانی محیطی و یا حمله، پیشرفت چموماتیک ممکن است از بین رفته و یا اینکه گروهی از باکتری‌های به نقطه دیگری منتقل شوند. این اتفاق از بین رفتن و یا پخش شدن در باکتری واقعی اتفاق می‌افتد.

به صورت خلاصه می‌توان گفت جستجوگرهای ما برای داده‌های مخرب دارای رفتارهای زیر می‌باشند:

رفتار حرکتی باکتری‌ها: که از آن به‌عنوان دوره حیات باکتری‌ها یاد می‌شود. این رفتار شامل Nc تکرار (طول دوره حیات) بوده و در آن باکتری‌ها گام‌هایی برای جست و جوی مواد مغزی بر می‌دارند. اگر در حرکت اول که بصورت تصادفی انجام می‌شود تابع هزینه کمتر شده باشد، Ns گام دیگر می‌توان در همان جهت جلو رفت به شرط آنکه در هر گام کاهش هزینه داشته باشیم. برای حرکت باکتری‌ها از رابطه (۱) استفاده می‌شود:

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)dlt(i)/(dlt(i)^T\, dlt(i))^{0.5} \qquad (۱)$$

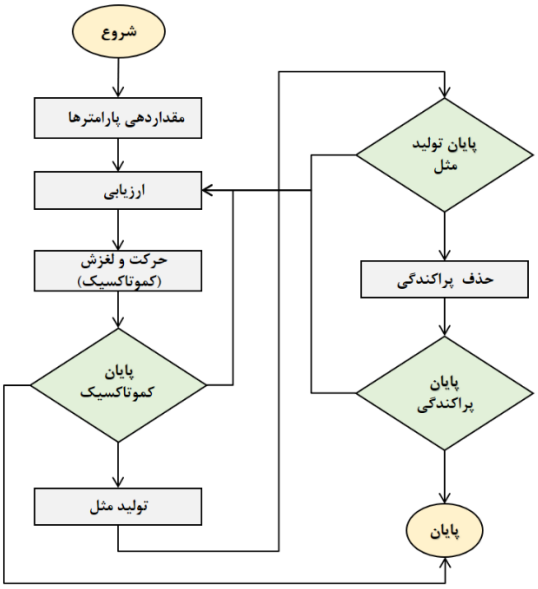
که در آن $\theta^{i}(j+1,k,l)$ موقعیت باکتری i در مرحله $j+1$ از رفتار حرکتی باکتری‌ها و kامین مرحله تولید مثل و یک امین مرحله حذف و پراکندگی است. C(i) گام حرکت و dlt(i) یک بردار تصادفی D بعدی در بازه برای تعیین جهت است. باکتری‌ها در شرایط خاصی ماده‌ای جاذب از خود ترشح می‌کنند که موجب جذب باکتری‌های دیگر به سمت یک ناحیه خاص می‌گردد. بر اساس این ارتباط در رابطه به‌روزرسانی تابع هزینه هر باکتری پس از حرکت آن طبق رابطه (۲) باید مقدار $J_{cc}(\theta, \theta^{i}(j,k,l))$ نیز به آن افزوده می‌شود که نماینده‌ای از میزان نیروهای جاذب و دافع بین باکتری‌ها در جمعیت است. (θ یک بردار در فضای D بعدی است.)

$$J(i,j,k,l) = J(i,j,k,l) + J_{cc}(\theta, \theta^{i}(j,k,l)) \qquad (۲)$$

پس از آنکه دوره حیات باکتری‌ها برای حرکت به پایان رسید میزان سلامت باکتری‌ها که متناسب با میزان مواد مغذی جمع‌آوری شده در طول دوره حیات است بر اساس رابطه (۳) برای همه باکتری‌ها محاسبه می‌گردد. سپس تعدادی از باکتری‌ها با بیشترین مجموع تابع هزینه می‌میرند (حذف می شوند) و به همان تعداد از بهترین باکتری‌ها تکثیر می‌شوند. (به دو باکتری تبدیل می‌شوند.)

$$J^{i}_{health} = \sum J(i,j,k,l) \qquad (۳)$$

در ازدحام واقعی باکتری‌ها اثر تغییرات محیطی مثل افزایش دما ممکن است خیلی از باکتری‌ها از بین بروند یا به نواحی دیگری بروند. با الهام از این رفتار بعد از تعداد تکرار خاصی از مرحله تولید مثل هریک از باکتری‌ها حذف شده و به مکان دیگری پرتاب می‌گردند (تبدیل به یک باکتری دیگر می‌شود.). فلوچارت این الگوریتم را در شکل ۳ می‌بینید:



شکل (۳): فلوچارت الگوریتم سیستم باکتری

۳-۱-۱- بهبود الگوریتم

الگوریتم جستجوی غذای باکتری، همچون سایر الگوریتم‌های مبتنی بر هوش جمعی، از ذرات مصنوعی به عنوان باکتری‌ها بهره می‌برد. در تمام این دسته از الگوریتم‌ها، ذرات مصنوعی مسئول جستجو در فضای مسئله هستند. در شرایط عادی، جستجو توسط این ذرات به صورت ترتیبی انجام می‌گیرد. زیرا هر باکتری، پردازنده را تا زمان اتمام پردازش مربوطه در اختیار می‌گیرد. در نتیجه در صورت امکانِ پردازش موازی، می‌توان چندین باکتری مصنوعی را به صورت همزمان و موازی اجرا کرد. در واقع، این ماهیت پردازش موازی الگوریتم‌های مبتنی بر هوش جمعی است که در محیط چندپردازنده، اجرای همزمان آنها ممکن است. الگوریتم جستجوی غذای باکتری نیز با توجه به وجود باکتری‌ها به عنوان ذرات مصنوعی، می‌تواند در یک محیط چندپردازنده و بصورت موازی اجرا شود. با ارایه یک الگوریتم کارا و توجه به محدودیت‌های موجود، می‌توان بهبود کارایی و سرعت الگوریتم را از طریق اجرای موازی تا چندین برابر افزایش داد. در ارائه

الگوریتم موازی غذایابی باکتری، از معماری CUDA بهره گرفته شده است. معماری CUDA امکانات مناسبی را برای برنامه‌نویسان موازی در اختیار قرار می‌دهد و در طراحی الگوریتم موازی غذایابی باکتری، از این امکانات بهره‌مند شده‌ایم. به این منظور، به هر یک از باکتری‌ها یک نخ پردازشی اختصاص داده شده است. نخ‌های موجود در برنامه‌های CUDA در قالب بلاک‌ها دسته‌بندی می‌شوند. این دسته‌بندی این امکان را فراهم می‌کند که نخ‌های موجود در یک بلاک با سرعت بیشتری اطلاعات را با یکدیگر تبادل کنند، و الگوریتم موازی غذایابی باکتری نیز از این امکان بهره‌مند شده است. اجرای الگوریتم از طریق بلاک باعث می‌شود که در فرایند پردازش الگوریتم، بجای بهره‌گیری از حافظه سراسری، از حافظه اشتراکی هر بلاک استفاده شود. با توجه به ماهیت موازی الگوریتم غذایابی باکتری و وجود امکانات سخت‌افزاری و نرم‌افزاری در معماری CUDA، تغییراتی در ساختار الگوریتم غذایابی باکتری ایجاد کرده‌ایم که اجرای الگوریتم موازی غذایابی باکتری را به صورت موازی و در معماری CUDA فراهم آورده است. در ادامه مهمترین این تغییرات بیان شده است.

- **مقداردهی اولیه بصورت موازی**: در مرحله مقداردهی اولیه الگوریتم موازی غذایابی باکتری، کلیهٔ باکتری‌ها به صورت همزمان و موازی مقداردهی اولیه می‌شوند. این عملیات با استفاده از تولید اعداد تصادفی در داخل پردازنده گرافیکی انجام می‌شود و موقعیت و پخش باکتری‌ها در فضای تعریف‌شده مسئله همزمان انجام می‌پذیرد.
- **محاسبه تابع برازش بصورت موازی**: یکی از محاسبات پیچیده در الگوریتم موازی غذایابی باکتری، محاسبه تابع برازش است. در این الگوریتم، این عملیات بصورت همزمان و موازی برای همه باکتری‌ها از طریق بکارگیری حافظه اشتراکی و توابع ریاضی تعریف‌شده در معماری CUDA انجام می‌شود.
- **اجرای فرایند مرتب‌سازی بصورت موازی**: در الگوریتم غذایابی باکتری، مرتب‌سازی بر اساس شاخص سلامت باکتری و استفاده از روش‌های ترتیبی صورت می‌گیرد. در راهکار پیشنهادی، از الگوریتم موازی مرتب‌سازی بایتونیک بهره گرفته شده است. این روش از دو فاز اصلی تشکیل گردیده است؛ در مرحلهٔ اول توالی اعداد به یک دنبالهٔ بایتونیک تبدیل می‌شود، سپس این دنباله مرتب می‌گردد. در این حالت از تعداد هسته‌های بالای پردازنده گرافیکی می‌توان برای انجام موازی این مرتب‌سازی استفاده کرد. در اینجا، شبه کد مرتب‌سازی بایتونیک در الگوریتم ۱ آمده است. در این شبه‌کد Array آرایه ورودی اعداد برای مرتب‌سازی می‌باشد، که تعداد اعداد برابر با n فرض شده است و n برابر با $2^{\wedge d}$ است که d اندازه توان عدد ۲ است.

```
parallel_bitonic_sort (array, direction):
For x in range (0 to direction-1)
  For y in range (x to 0)
     If (array[x+1] ≠ array[y]
        bitonic_compare max(y)
     Else
        bitonic_compare min(y)
End If
```

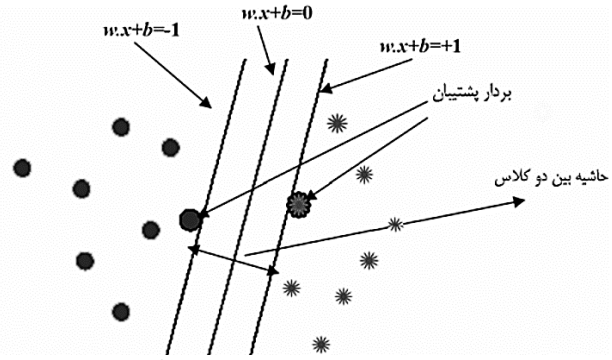
الگوریتم (۱): شبه کد مرتب‌سازی موازی بایتونیک

۳-۲- الگوریتم ماشین بردار پشتیبان (SVM)

الگوریتم SVM یک روش آماری غیر پارامتریک نظارت شده است و بر اساس این فرض عمل می‌کند که هیچ‌گونه اطلاعی از چگونگی توزیع مجموعه داده‌ها وجود نداشته باشد. ویژگی اصلی این روش توانایی بالا در استفاده از نمونه‌های تعلیمی کمتر و رسیدن به‌دقت بالاتر در مقایسه با سایر روش‌های طبقه‌بندی است. در شکل ۴ دو کلاس و بردارهای پشتیبان مربوط به آن‌ها نشان داده‌شده است. فرض می‌شود داده‌ها از دو کلاس تشکیل‌شده و کلاس‌ها درمجموع دارای($i=۱,\ldots,L$) x_i نقطه آموزشی باشند که x_i یک بردار است. این دو کلاس با $y_i = \pm ۱$ برچسب زده می‌شوند. با توجه به آنکه قصد داریم داده‌ها را به دو دسته داده‌ای تمیز و آلوده تقسیم نماییم در

نتیجه این دو کلاس کاملاً جدا ازهم درنظر گرفته می‌شوند و برای محاسبه مرز تصمیم‌گیری دو کلاس کاملاً جدا، از روش حاشیه بهینه استفاده می‌شود [۶]. در این روش مرز خطی بین دو کلاس به‌گونه‌ای محاسبه می‌شود که:

- تمام نمونه‌های کلاس +۱ در یک‌طرف مرز و تمام نمونه‌های کلاس -۱ در طرف دیگر مرز واقع شوند.
- مرز تصمیم‌گیری به‌گونه‌ای باشد که فاصله نزدیک‌ترین نمونه‌های آموزشی هر دو کلاس از یکدیگر در راستای عمود بر مرز تصمیم‌گیری تا جایی که ممکن است حداکثر شود.



شکل (۴): مرز خطی بهینه برای حالتی که دو کلاس کاملاً از یکدیگر جدا هستند

یک مرز تصمیم‌گیری خطی را در حالت کلی می‌توان به‌صورت (۴) نوشت:

$$w.x+b = 0 \qquad (۴)$$

در رابطه فوق x یک نقطه روی مرز تصمیم‌گیری و w یک بردار nبعدی بر مرز تصمیم‌گیری است. b نیز فاصله مبدا تا مرز تصمیم‌گیری و w. بیانگر ضرب داخلی دو بردار w و x است. از آنجاکه با ضرب یک ثابت در دو طرف رابطه فوق بازهم تساوی برقرار خواهد بود، برای تعریف یکتای مقدار b و w شرایط معادلات زیر بر روی آنها اعمال می شود.

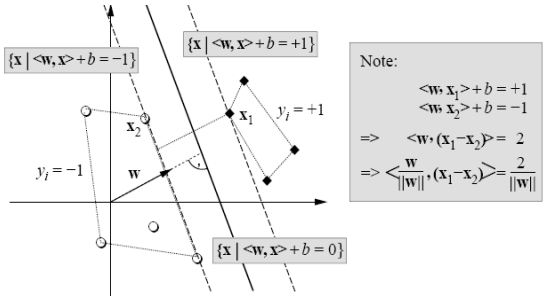
$y_i(w.X_1 + b) = 1$ اگر x_i یک بردار پشتیبان باشد (۵)

$y_i(w.X_1 + b) > 1$ اگر x_i یک بردار پشتیبان نباشد (۶)

اولین مرحله برای محاسبه مرز تصمیم‌گیری بهینه، پیدا کردن نزدیک‌ترین نمونه‌های آموزشی دو کلاس است. در مرحله بعد فاصله آن نقاط از هم در راستای عمود بر مرزهایی که دو کلاس را به طور کامل جدا می‌کنند، محاسبه می‌شود. مرز تصمیم‌گیری بهینه، مرزی است که حداکثر حاشیه را داشته باشد. در واقع روش‌هائی مانند SVM، سعی دارند که با ساختن یک ابرسطح (که عبارت است از یک معادله خطی)، داده‌ها را از هم تفکیک کنند. روش طبقه‌بندی ماشین بردار پشتیبان که یکی از روش‌های طبقه‌بندی خطی است، بهترین ابرسطحی را پیدا می‌کند که با حداکثر فاصله، داده‌های مربوط به دو طبقه را از هم تفکیک کند.

۳-۲-۱- نحوه تشکیل ابرسطح جداکننده

ابرسطحی را پیدا می‌کند که با حداکثر فاصله، داده‌های مربوط به دو طبقه را از هم تفکیک کند. در این بخش می‌خواهیم نحوه ساخت ابرسطح جداکننده را بر روی یک مثال با جزئیات شرح دهیم. تصویر دقیقی از نحوه تشکیل ابرسطح جداکننده توسط ماشین بردار پشتیبان در شکل ۵ نشان داده شده است.



شکل (۵): نحوه ساخت ابرسطح جداکننده بین دو طبقه داده در فضای دو بعدی [۵]

ابتدا یک پوسته محدب در اطراف نقاط هر کدام از کلاس‌ها در نظر بگیرید. در شکل ۵ در اطراف نقاط مربوط به کلاس -1 و نقاط مربوط به کلاس $+1$ پوسته محدب رسم شده است. خط $\boldsymbol{P}$ خطی است که نزدیکترین فاصله بین دو پوسته محدب را نشان می‌دهد. h که در واقع همان ابرسطح جداکننده است، خطی است که $\boldsymbol{P}$ را از وسط نصف کرده و بر آن عمود است. b عرض از مبدا برای ابرسطح با حداکثر مرز جداکننده است. اگر b صرف‌نظر شود، پاسخ تنها ابرسطح‌هایی هستند که از مبدا می‌گذرند. فاصله عمودی ابرسطح تا مبدا با تقسیم قدرمطلق مقدار پارامترb برطول w بدست می‌آید. ایده اصلی این است که یک جداکننده مناسب انتخاب شود. منظور، جداکننده‌ای است که بیشترین فاصله را با نقاط همسایه از هر دو طبقه دارد. این جواب درواقع بیشترین مرز را با نقاط مربوط به دو طبقه مختلف دارد و می‌تواند با دو ابرسطح موازی که حداقل از یکی از نقاط دو طبقه عبور می‌کنند، کران‌دار شود. این بردارها، بردارهای پشتیبان نام دارند. فرمول ریاضی این دو ابرسطح موازی که مرز جداکننده را تشکیل می‌دهند در عبارات زیر نشان داده شده است:

$$w.x - b = 1 \qquad (۷)$$

$$w.x - b = -1 \qquad (۸)$$

نکته قابل توجه این است که اگر داده‌های تعلیمی به صورت خطی تفکیک‌پذیر باشند، می‌توان دو ابرسطح مرزی را به گونه‌ای انتخاب کرد که هیچ داده‌ای بین آنها نباشد و سپس فاصله بین این دو ابرسطح موازی را به حداکثر رساند. با به‌کارگیری قضایای هندسی، فاصله این دو ابرسطح عبارت است از $2/|w|$، پس باید $|w|$ را به حداقل رساند. همچنین باید از قرار گرفتن نقاط داده در ناحیه درون مرز جلوگیری کرد، برای این کار یک محدودیت ریاضی به تعریف فرمال اضافه می‌شود. برای هر i، با اعمال محدودیت‌های زیر اِطمینان حاصل می‌شود که هیچ نقطه‌ای در مرز قرار نمی‌گیرد:

$w.x\_i - b \geq 1$ برای داده‌های مربوط به طبقه اول (۹)

$w.x\_i - b \leq -1$ برای داده‌های مربوط به طبقه دوم (۱۰)

می‌توان این محدودیت را به صورت رابطه زیر نشان داد:

$$c_i(w.x_i - b) \geq 1 \quad , 1 \leq i \leq n \qquad (۱۱)$$

لذا مسأله بهینه‌سازی بدین شکل تعریف می‌شود: به حداقل رساندن w، با درنظر گرفتن محدودیت زیر:

$$i \leq n \leq c\_i\,(w.x\_i - b) \geq 1\ ,\ 1 \qquad (۱۲)$$

انجام عملیات یادگیری

در این پژوهش از الگوریتم خودآموزی به‌منظور انجام علیات یادگیری استفاده می‌شود. نحوه عملکرد الگوریتم بدین صورت است که:

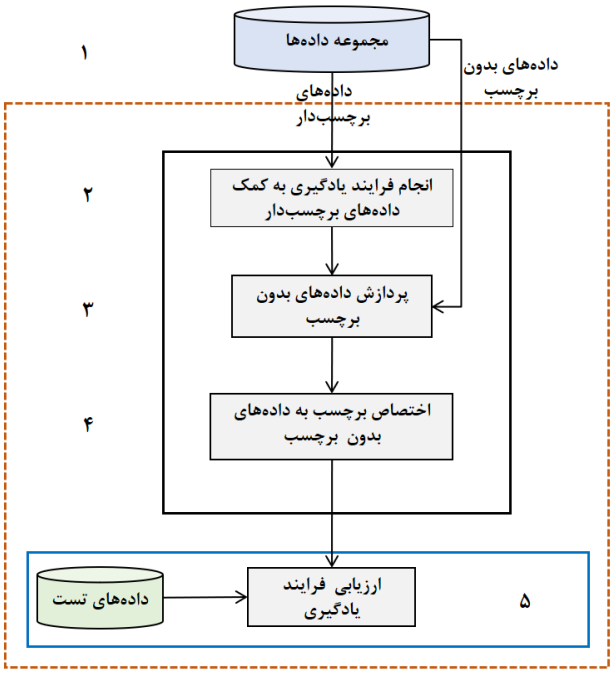
۱. یک رده‌بندی روی داده‌های برچسب دار ساخته می‌شود.
۲. از این رده برای طبقه بندی داده‌های بدون برچسب استفاده می‌شود.
۳. داده‌ها براساس بیشترین اطمینان برچسب دار می‌شوند.
٤. داده‌ها برچسب دار شده به مجموعه داده‌های آموزشی اضافه و رده‌بندی جدید تولید می‌شوند.

۵. به مرحله ۱ برمی‌گردد و تازمانی که هیچ داده‌ای باقی نمانده باشد این روند به صورت تکرار ادامه می‌یابد.

روش خودآموزی از قویترین الگوریتم‌های یادگیری نظارتی و نیمه نظارتی می‌باشد که به طور گسترده استفاده می‌شود. این راهکار بیشتر برای مسائل کشف دانش مختلف در مجموعه مختلفی از داده‌ها استفاده می‌شود. این روش، دقت خوبی را حتی زمانی که داده‌های برچسب‌دار کم باشند را ارائه خواهد داد. در واقع کار عمده‌ای که توسط راهکار پیشنهادی انجام گرفته است، ایجاد حداقل مقدار مجموع فواصل درون خوشه‌ای بوده است. طرح همراه با جزئیات مدل پیشنهادی در شکل ۶ نشان داده شده است. در اینجا با بهره‌گیری از الگوریتم پیشنهادی، داده‌های مربوط به تشخیص نفوذ، خوشه‌بندی می‌شوند که در مقایسه با روش‌هائی مانند K-Means خوشه‌بندی با دقت بیشتری صورت می‌گیرد. روند اجرای فرایند یادگیری در راهکار پیشنهادی که مبتنی بر تست خودکار می‌باشد به صورت زیر است:

۱. ترکیب داده‌های برچسب‌دار و غیر برچسب‌دار جهت ورود به مدل پیشنهادی.
۲. الگوریتم به کمک داده‌های برچسب‌دار آموزش داده می‌شود.
۳. فرایند پیش‌پردازش بر روی داده‌های به منظور همگن‌سازی آن‌ها انجام می‌گیرد.
٤. داده‌های برچسب‌دار به عنوان ورودی وارد الگوریتم می‌شوند، و عملیات برچسب گذاری به کمک داده‌های برچسب‌دار انجام می‌گیرد.

حال برای اینکه بدانیم مدل ما به واقعیت چقدر نزدیک است از روش‌های تست و ارزیابی استفاده می‌کنیم و دقت مدل پیشنهادی برای داده‌های غیر برچسب دار پیش بینی شده، ارزیابی و تست می‌شوند.



شکل (۶): طرح همراه با جزئیات مدل پیشنهادی در فرایند یادگیری

۳-۲-۲- ارزیابی و کارایی

برای مقایسه، ارزیابی و تجزیه و تحلیل نتایج، ما چهار معیار مهم و تاثیرگذار را برای ارزیابی سیستم‌های تشخیص نفوذ پیشنهادی انتخاب کردیم. همچنین برای ارزیابی مدل از دیتاست KDDcup99 استفاده شده است [۲۵]. که در آن حدود ۲.۵ میلیون اسکن در بین سالهای ۲۰۱۴ تا ۲۰۱۹ انجام گرفته است و دیتاست به صورت تصادفی از این دیتاست بزرگ تهیه شده است. برای ارزیابی چندین روش مختلف بررسی می‌شود که در ادامه آنها را بیان خواهیم کرد.

- **نرخ تشخیص:** نشان‌دهنده نسبت نمونه های نفوذی است که به درستی توسط مدل پیشنهادی شناسایی شده است [۲۶]. میزان تشخیص در فرمول ۱۳ نشان داده شده است.

$$DR = \frac{TP}{TP + FN} \quad (۱۳)$$

- **نرخ هشدار کاذب:** این معیار سطح هشدارهای کاذب را در سیستم زمانی که هیچ حمله ای رخ نمی‌دهد نشان می‌دهد، اما سیستم هشدار می‌دهد که نسبت نمونه‌هایی که عادی هستند به عنوان خطای حمله FP شناسایی می‌شوند [۲۴]. نرخ هشدار نادرست در فرمول ۱۴ نشان داده شده است.

$$FAR = \frac{TP}{TP + TN} \quad (۱۴)$$

دقت: یکی از مهمترین معیارهای ارزیابی است که دقت یک طرح را در تشخیص اتصالات و حمله معمولی توصیف می‌کند. نسبت حمله و نمونه‌های معمولی که به درستی تشخیص داده شده‌اند[۲۲]. دقت در فرمول ۱۵ نشان داده شده است.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (۱۵)$$

- **زمان آموزش (TT):** زمان آموزش عامل بسیار مهمی در توسعه سیستم‌های تشخیص نفوذ است و از این رو عملکرد سیستم‌های تشخیص نفوذ شبکه را افزایش می‌دهد. زیرا سیستم‌های شبکه برای استفاده مداوم طراحی شده‌اند و در هر ثانیه تعداد زیادی بسته از شبکه دریافت می‌کنند. یک سیستم‌های تشخیص نفوذ با قابلیت آموزش و ایجاد یک طبقه‌بندی کننده در زمان کوتاه را می‌توان یک سیستم‌های تشخیص نفوذ شبکه بسیار موثر در نظر گرفت. همچنین این طبقه‌بندی‌کننده باید بسیار مؤثر و کارآمد باشد و سیستم‌های تشخیص نفوذ می‌تواند آن طبقه‌بندی را همزمان به‌روزرسانی کند [۲۵].

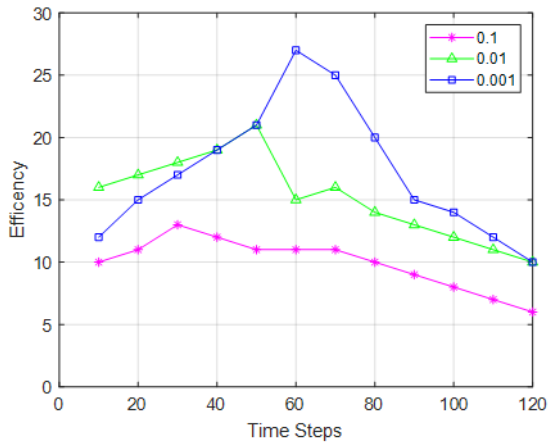
هدف ما ایجاد ساختاری است که بتواند حملات برنامه ریزی شده را با دقت در کمترین زمان برای آموزش تشخیص دهد. به طور خاص، تمرکز ما بر روی بررسی عملکرد ساختار پیشنهادی بر روی پارامترهای اصلی بود که می‌تواند عملکرد یک سیستمهای تشخیص نفوذ را در شبکه‌های کامپیوتری ارزیابی کند.

۳-۳- تجزیه و تحلیل عملکرد روش پیشنهادی

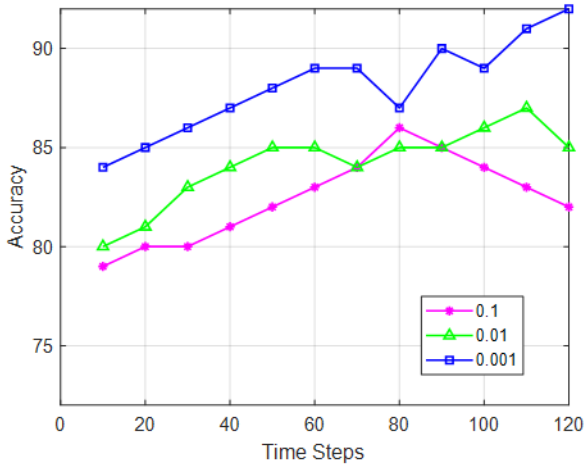
برای انجام نرخ یادگیری و تعداد مراحل زمانی ، ابتدا نرخ یادگیری در هر آزمایش را روی یکی از سه مقدار (۰.۱، ۰.۰۱، ۰.۰۰۱) قرار می‌دهیم و سپس مراحل زمانی را از ۱۰ به ۱۲۰ در مراحل ۱۰ افزایش می‌دهیم و در هر مورد چهار به DR نگاه می‌کنیم. FAR، زمان آموزش یا Training Time و معیار دیگری به نام Efficiency که نسبت نرخ تشخیص به تعداد هشدارهای های اشتباه است. با استفاده از این معیارها می‌توان به راحتی مدل IDS را ارزیابی کرد. هدف داشتن سیستم‌های تشخیص نفوذ DR بالاتر و FAR کمتر و در نهایت بالاترین دقت در کوتاه‌ترین زمان است. با محاسبه بازده ثابت می‌شود که با افزایش DR و کاهش FAR بازده افزایش می‌یابد و بر این اساس بهره‌وری با معادله ۱۷ محاسبه می‌شود.

$$Efficiency = \frac{DR}{FAR} \quad (۱۷)$$

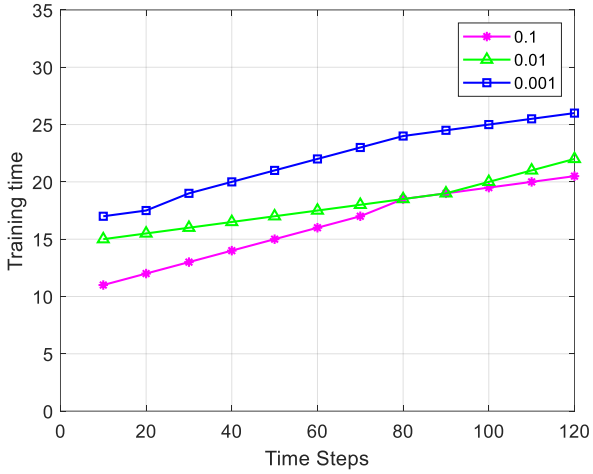
در این مدل، با توجه به نتایج به‌دست‌آمده در ایزوله‌ها و نمودارها، می‌توان گفت که بهترین نتایج با در نظر گرفتن تمامی معیارهای ارزیابی، با نرخ یادگیری ۰.۰۰۱ به دست می‌آید زیرا مطابق شکل۷ و ۸ ، با افزایش نرخ یادگیری، میانگین نرخ تشخیص افزایش می‌یابد و نرخ هشدار نادرست تا حدودی کاهش می‌یابد که منجر به افزایش بازده می‌شود. یادگیری دقت مدل نیز افزایش می‌یابد. مطابق شکل ۹، بیشترین بازده در اندازه گام زمانی ۶۰ و در نرخ یادگیری ۰.۰۰۱ است. همچنین مطابق شکل ۷ بیشترین دقت در این مورد اتفاق می‌افتد که برابر با ۹۲ درصد است. زمان آموزش این حالت ۲۴.۷۲ می‌باشد که مطابق شکل ۱۰ تا ۱۳ کمی بیشتر از دو نرخ یادگیری دیگر است که به دلیل کاهش سرعت تغییرات وزن با کاهش نرخ یادگیری برای افزایش دقت به دست آمده در مدل است.



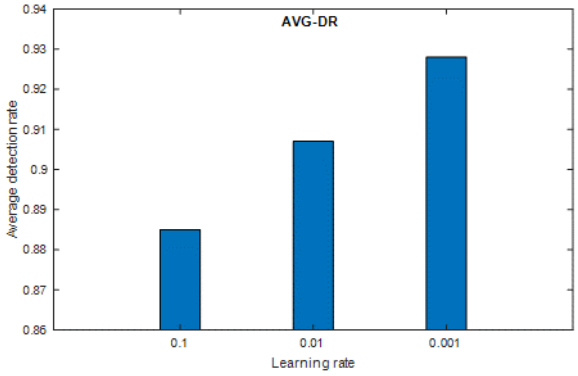
شکل (۷): مقایسه تغییرات نرخ عملکرد با تغییرات اندازه مرحله زمانی برای سه نرخ یادگیری استفاده شده



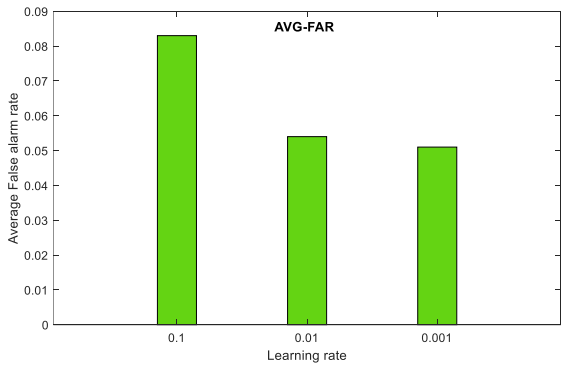
شکل (۸): مقایسه تغییرات نرخ دقت با تغییرات اندازه گام زمانی برای سه نرخ یادگیری استفاده شده



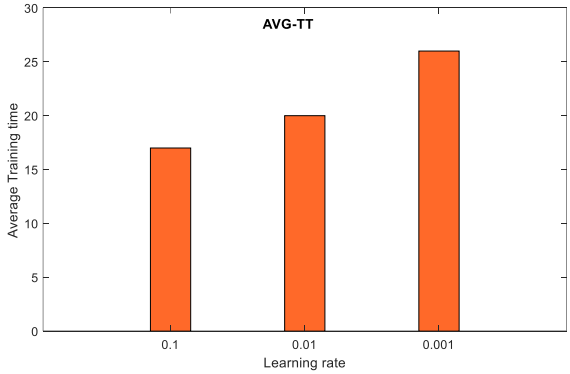
شکل (۹): مقایسه تغییرات زمان آموزش با تغییرات اندازه گام زمانی برای سه نرخ یادگیری استفاده شده



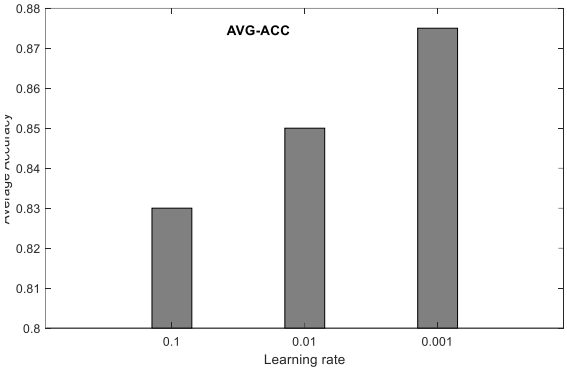
شکل (۱۰): مقایسه میانگین نرخ‌های تشخیص در سه نرخ یادگیری



شکل (۱۱): مقایسه میانگین نرخ هشدار نادرست در سه نرخ یادگیری



شکل (۱۲): مقایسه میانگین زمان آموزش AVG-TT در سه نرخ یادگیری



شکل (۱۳): مقایسه میانگین دقت سه نرخ یادگیری

۳-٤- تحلیل برچسب‌ها

در این ارزیابی برچسب‌ها را مورد ارزیابی قرار می‌دهیم . به این صورت که ۴ دسته مختلف از داده‌ها را به صورت تصادفی انتخاب می شود. رخدادهای تکراری نمونه ها در ترتیب اسکن اصلی بیانگر چندین اسکن نمونه‌های یکسان در نقاط مختلف در زمان می‌باشد.

در این مقاله پنج روش مختلف اسکن را برای تقسیم اسکن‌ها به زیربازه‌های زمانی در نظر گرفتیم که هرکدام حرکت مجزای خود را دارد. همه روش‌های اسکن ترتیب داده‌ها را اجرا می‌کنند و برای ایجاد هر زیربازه زمانی تقسیم‌بندی و ترتیب را رعایت می‌کنند. به دلیل اینکه هر نمونه می‌تواند در چندین بازه زمانی ظاهر شود می‌توان گفت که در زمان اجرا و هم در زمان تست یک داده یکسان می‌تواند حضور داشته باشد.

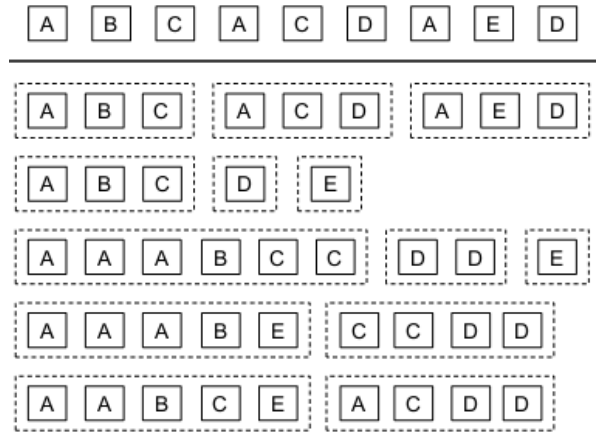
در جدول ۲ ارزیابی بین مدیریت برچسب داده‌های تست و آموزش برای انجام عملیات جستجو قابل مشاهده است. برچسب‌های نمونه‌های ۲ و ۳ در هر زمان ثابت هستند. و برچسب‌های ۱ و ۴ و ۵ با افزایش دانش داده‌ها در حال تغییر می‌باشد. استفاده از دانش برچسب‌ها می‌تواند کارایی روش را بالاتر ببرد. در اینجا داده‌های ۱ و ۴ بسیار آسان‌تر در بازه زمانی ۳ شناسایی می‌شوند و این نشان می‌دهد که در بازه زمانی ۱و ۲ آموزش دیده‌اند. این ارزیابی نشان می‌دهد که دانش برچسب‌ها تا چه اندازه می تواند در کیفیت آموزش داده‌ها و نمونه موثر واقع شود. برای مثال نمونه ۱ به عنوان داده مخرب در اینجا شناسایی شده است و در بازه زمانی ۳ به این شناسایی رسیده است. برای ارزیابی برچسب کردن را در پایان هر بازه زمانی انجام دادیم. داده‌های آموزش قبل از داده‌های تست پردازش می‌شوند و برچسب‌های ارزیابی بعد از پایان همه بازه‌ها جمع آوری می‌شوند.

جدول (۲): ارزیابی و مدیریت برچسب داده‌های تست و آموزش

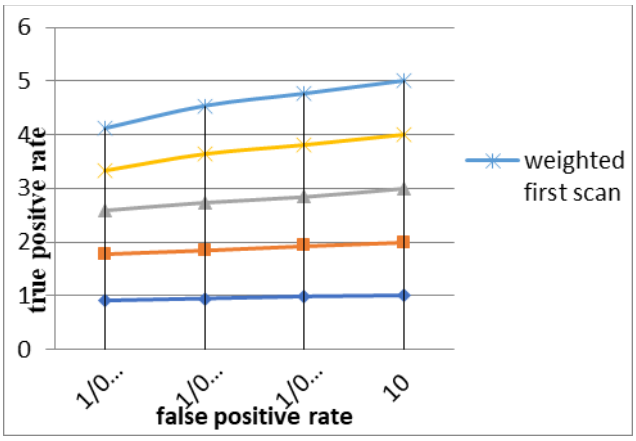
نمونه	بازه ۱			بازه ۲			بازه۳		
	۱	۲	۳	۱	۳	۴	۱	۵	۴
برچسب در بازه زمانی ۱	−	−	+						
برچسب در بازه زمانی ۲	−	−	+	−	+	−			
برچسب در بازه زمانی ۳	+	−	+	+	+	−	−	−	+
برچسب مربوط به ماه در بازه زمانی ۳	+	−	+	+	+	+	+	+	+

در اینجا از روش Cross- validation برای تقسیم داده‌ها به داده‌های تست و داده‌های آموزش استفاده شده است. زیرا در این روش توزیع داده‌ها یکسان می ماند. ما این روش را در مقایسه با روش‌های پیشین قرار می‌دهیم. در شکل ۱۴ مقایسه نوع توزیع داده‌ها را با روش‌های بیان شده برای مجموعه داده‌های A تا E نشان می‌دهد. در این شکل خط چین‌ها بیانگر بازه‌های زمانی می‌باشد و خط اول بیانگر ترتیب اسکن اصلی می‌باشد. و خطهای بعدی روش‌های اسکن به ترتیب روش all scans، روش first scan only، روش weighted first scan، روش sample cross validation و روش پیشنهادی ما روش scan cross validation می‌باشد.

روش‌های پارتیشن بندی در هر ۵ روش برای تست و آموزش متفاوت می‌باشد. و مدل پارتیشن بندی برای n بازه زمانی همان مدل برای بازه زمانی n+1 خواهد بود. در روش first scan only هر نمونه دقیقا در یک بازه زمانی اسکن می‌شود و برای هر نمونه اولین رخداد باقی می‌ماند. در روش weighted first scan اطلاعات نمونه های محلی را نگه می‌دارد ولی همه اسکن‌ها در اولین بازه زمانی جای می‌دهد. در روش cross validation داده‌های آموزش و تست با توجه به زمان چیده می‌شوند. این روش نسبت به روش‌های پیشین مناسب تر می‌باشد.



شکل (۱۴): مقایسه روش‌های پارتیشن‌بندی



شکل (۱۵): مقایسه بین روش‌های پارتیشن بندی

در جدول ۳ مقایسه نوع توزیع داده‌ها را با روش‌های بیان شده برای مجموعه داده‌های A تا E نشان می‌دهد.

جدول (۳): مقایسه نوع توزیع داده ها

	scan cross validation	all scans	simple cross validation	first scan	weighted first scan
۱.۰۰E-۰۲	۰.۹	۰.۸۸	۰.۸	۰.۷۵	۰.۷۸
۱.۰۰E-۰۱	۰.۹۴	۰.۹۱	۰.۸۸	۰.۹۱	۰.۸۹
۱.۰۰E+۰۰	۰.۹۸	۰.۹۵	۰.۹۱	۰.۹۷	۰.۹۵
۱۰	۱	۱	۱	۱	۱

در شکل ۱۶ کارایی روش پیشنهادی نمایش داده شده است به این صورت که نمودار آبی بیانگر ۸۰ در خواست در روز می‌باشد که کارایی روش پیشنهادی را ملاحظه می‌کنید.

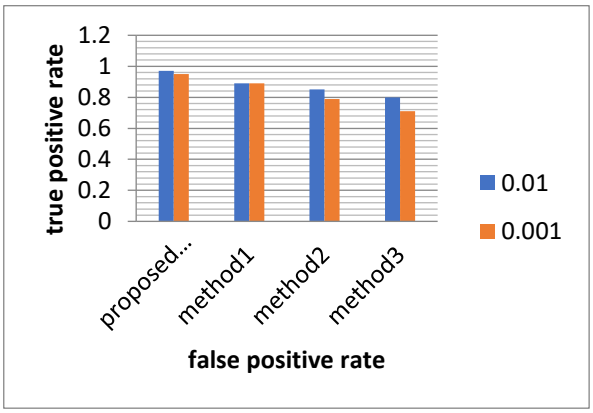
شکل (۱۶): کارایی روش پیشنهادی

جدول (۴): کارایی روش پیشنهادی

TPR	FPR
۰.۳۵	۱.۰۰E-۰۳
۰.۹۱	۱.۰۰E-۰۲
۰.۹۵	۱.۰۰E-۰۱
۰.۹۷	۱.۰۰E+۰۰
۱	۱۰

در جدول ۴ کارایی روش پیشنهادی نمایش داده شده است به این صورت که برای ۸۰ در خواست در روز مقادیر FPR, TPR نشان داده شده است که کارایی روش پیشنهادی را ملاحظه می‌کنید.

در این روش جستجو را طوری در نظر گرفتیم که به صورت آنلاین باشد یعنی بتواند تجربیات خود را بروز کند. تجربیاتی که از نمونه‌ها و مثالهای داده‌های مخرب بدست آورده است و آن تجربیات را در جستجوهای بعدی بتواند استفاده کند. و همچنین گرفتن تجربه از بیرون به این معنی که بتواند در هر بار از تجربیات مرور کاربران استفاده نماید و داده های مخرب جدیدی که به آن معرفی می‌گردد را به دیتابیس ویژگیهای خود اضافه کند. علاوه بر آن از تجربیات درونی خود نیز که پس از هر بار اجرا بدست می آید استفاده کند. در شکل ۱۷روشهای مختلف را مورد بررسی و مقایسه قرار دادیم . روش پیشنهادی را با روش اول یعنی روشی که از تجربیات عوامل خارجی و عوامل داخلی بدون در نظر گرفتن داده‌های مخرب و روش دوم یعنی روشهایی که فقط از کاربران داده های مخرب را شناسایی می‌کنند و روش سوم روشی که از هیچ کدام از این جستجوها استفاده نمی‌کند و صرفا بر اساس همان ماتریس ویژگی اولیه عملیات جستجو را انجام می‌دهد مورد مقایسه قرار می‌دهیم. نتایج نشان می دهد که روش پیشنهادی روش قابل قبولی نسبت به روشهای پیشین می‌باشد. شکل۱۷ مقایسه بین روش پیشنهادی و روش‌های پیشین نشان داده شده است. همانطور که نمودار نشان می‌دهد روش پیشنهادی نسبت به روش‌های پیشین نتایج بهتری داشته است.



شکل (۱۷): مقایسه روشها

جدول ۵ مقایسه بین روش پیشنهادی و روشهای پیشین نشان داده شده است. همانطور که جدول نشان می‌دهد روش پیشنهادی نسبت به روش‌های پیشین نتایج بهتری داشته است.

جدول (۵): مقایسه بین روشها

روش‌ها	FPR=0.001	FPR=0.01
روش پیشنهادی	۰.۹۵	۰.۹۷
روش ۱	۰.۸۹	۰.۸۹
روش ۲	۰.۷۹	۰.۸۵
روش ۳	۰.۷۱	۰.۸

۳-۵- ارزیابی کمی

در این بخش راهکار پیشنهادی با توجه به پارامترهای دقت، حساسیت و F۱ در مقایسه با راهکارهای NBFS، NBTree و C4.5 [۲۷] مورد مقایسه و ارزیابی قرار گرفته است. فرمول این توابع در روابط ۱۸ تا ۲۰ آمده است.

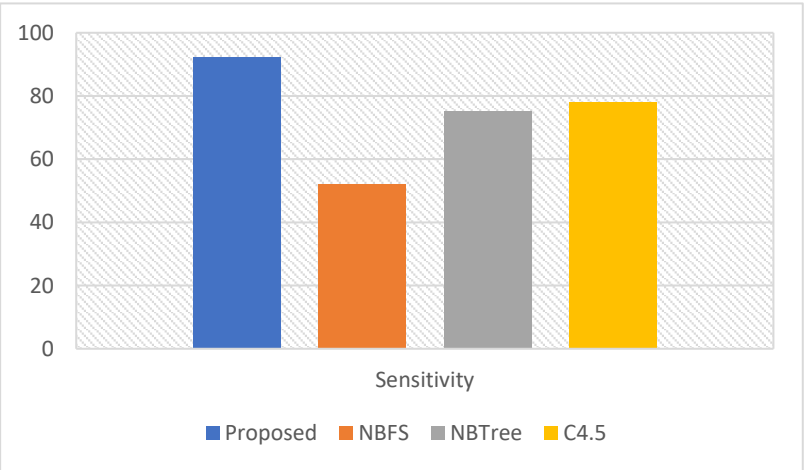
ابتدا و در شکل ۱۸ راهکارها با توجه به نرخ حساسیت بررسی و مقایسه شده‌اند. از این پارامتر به منظور بررسی اینکه تا چه میزان روش پیشنهادی،

$$Percesion = \frac{\mathrm{TP}}{TP + FP} \tag{۱۸}$$

$$F1 = \frac{2 \times Precision \times Sensitivity}{Precision + Sensitivity} \tag{۱۹}$$

$$Sensitivity = \frac{TN}{FP + TN} \tag{۲۰}$$

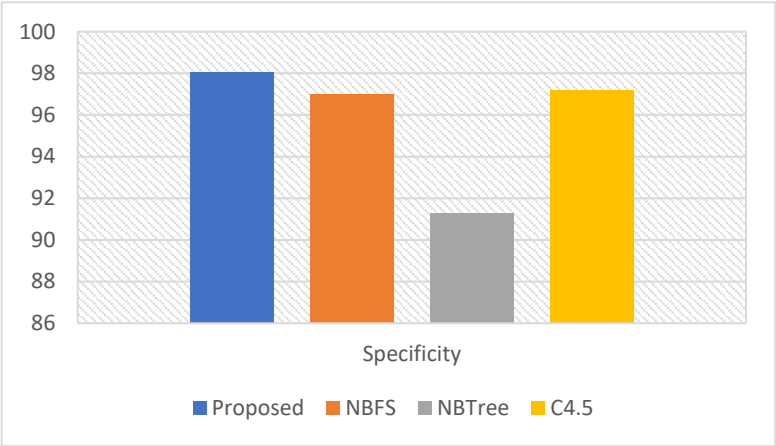
داده‌هایی که در کلاس موردنظر نبوده‌اند را به‌عنوان داده‌های درون کلاس دسته‌بندی کرده است. همانطورکه ملاحظه می‌شود راهکار ارائه شده از سه راهکار NBFS، C4.5 و NBTree بهتر عمل کرده است. این میزان از بهینگی در درست تشخیص داده شدن نفوذ، در بهره‌گیری از روش ترکیبی پیشنهادی می‌باشد که در واقع از طریق استفاده از الگوریتم باکتری توانسته‌ایم تا حد زیادی ویژگی‌هائی که جهت تشخیص نفوذ لازم نیست حذف نموده و در نتیجه دقت دسته‌بند در تشخیص موارد مثبت بخوبی بالا رفته است.



شکل (۱۸): مقایسه پارامتر حساسیت در بین راهکارهای مورد ارزیابی

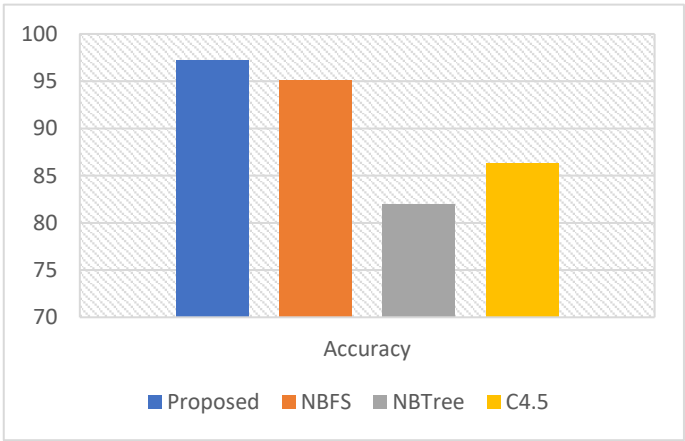
در ادامه و در شکل ۱۹ راهکارها با توجه به معیار F1 مورد ارزیابی قرار گرفته‌اند. از طریق این پارامتر می‌توان بررسی کرد که تا چه میزان روش‌های پیشنهادی، داده‌هایی که در کلاس موردنظر بوده‌اند را درست دسته‌بندی کرده‌اند. همانطورکه ملاحظه می‌شود در این

حالت نیز راهکار ارائه شده بخوبی توانسته داده‌ها را در کلاس‌های صحیح دسته‌بندی کند. بطوریکه به میزان دقت ۹۸٪ رسیده است. دلیل اصلی این میزان از کارایی در بهره‌گیری از الگوریتم BFO می‌باشد که بخوبی توانسته است از طریق انتخاب مجموعه ویژگی‌های مناسب برای الگوریتم SVM این امکان را فراهم کرده است که بهترین یادگیری از طریق ویژگی‌های اصلی، جهت تشخیص نفوذ انجام گیرد. در انتها و در شکل ۲۰ پارامتر نرخ صحت ارزیابی‌ها نشان داده شده است.



شکل (۱۹): مقایسه پارامتر اختصاصی بودن در بین راهکارهای مورد ارزیابی

همانطورکه در شکل ۲۰ ملاحظه می‌شود در این ارزیابی نیز راهکار پیشنهادی به نسبت سه راهکار دیگر بهتر عمل کرده است. با مقایسه این پارامترها می‌توان این نتیجه را گرفت که از طریق بهره‌گیری از راهکار ترکیبی پیشنهادی توانسته‌ایم بخوبی پارامترهای اصلی جهت تشخیص نفوذ در شبکه را بهینه نماییم بطوریکه با حداکثر دقت بتوان عملیات تشخیص نفوذ را به کمک الگوریتم ماشین‌بردار پشتیبان که توسط الگوریتم BFO بهینه شده است انجام داد. در کنار آن بهره‌گیری از دسته بندی مبتنی بر الگوریتم ماشین بردار پشتیبان نیز تأثیر بسزایی بر روی دقت و صحت عملیات داشته است زیرا خروجی مراحل قبلی بعنوان ورودی به الگوریتم ماشین بردار پشتیبان داده می‌شود تا به کمک آن بتوان نفوذهای تشخیص داده شده را در کلاس‌های مشخصی قرار داد و در نتیجه بر اساس آنها سیستم تشخیص نفوذ بتواند تصمیم‌گیری لازم را بگیرد. اگر در این مرحله، دسته‌بندی‌ها دقیق نباشد میزان کارایی راهکار نیز پایین خواهد آمد. در واقع همانطورکه در شکل ۲۰ نیز مشخص است راهکار پیشنهادی توانسته است به نرخ دقت بالای ۹۷ درصد برسد که در مقابل نرخ صحت ۸۲ و ۸۶ درصد راهکارهایی مانند NBTree و C4.5 دیگر بسیار بالاتر می‌باشد.



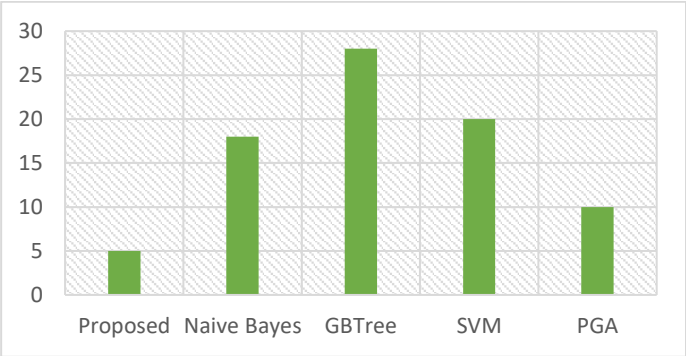
شکل (۲۰): مقایسه نرخ دقت در بین راهکارهای مورد ارزیابی

۳-۶- ارزیابی زمان اجرا

در این بخش زمان اجرای راهکار پیشنهادی در مقایسه با سایر راهکارها مورد ارزیابی قرار می‌گیرد. برای این منظور الگوریتم‌های زیر جهت مقایسه با روش پیشنهادی، انتخاب شده‌اند:

- ماشین بردار پشتیبان (SVM[۱۰])
- الگوریتم بیز ساده (Naive Bayes)
- الگوریتم مبتنی بر درخت (GB[۱۱])
- الگوریتم ژنتیک موازی‌سازی شده (PGA)

در ادامه و در شکل ۲۱ نتایج حاصل از ارزیابی نشان داده شده است. لازم به ذکر است در فرایند ارزیابی از دیتاست KDDcup۹۹ استفاده شده است.



شکل (۲۱): مقایسه زمان اجرای راهکارها

همانطورکه در شکل ۲۰ ملاحظه می‌شود، زمان اجرای روش پیشنهادی بسیار کمتر از چهار راهکار دیگر می‌باشد، بطوریکه در مقایسه با الگوریتم‌ها Navie Bayes، SVM و GBTree زمان اجرا کاهش چشم‌گیری داشته است. هر چند با توجه به آنکه در الگوریتم PGA نیز از موازی‌سازی‌شده استفاده شده است این میزان بهینگی کمتر است اما با توجه به آنکه PGA فاقد استخراج‌کننده ویژگیِ کارا می‌باشد در نتیجه به دلیل آنکه فرایند یادگیری هم بر روی داده‌های پرت و هم داده‌های اصلی صورت می‌گیرد، بر همین اساس زمان اجرای آن به نسبت راهکار پیشنهادی بالاتر رفته است. دلیل این میزان از بهینگی در راهکار پیشنهادی بکارگیری و اجرای الگوریتم غذایابی باکتری بصورت موازی است. در واقع در این مرحله با توجه به آنکه فرایند مربوط به مقداردهی اولی، توزیع باکتری‌ها و همچنین مرتب سازی آنها بصورت موازی انجام می‌گیرد، در نتیجه زمان اجرای الگوریتم به شدت پایین می‌اید. همین عمل باعث می‌شود که کارایی الگوریتم با توجه به زمان اجرا بالاتر رود که در نمودار نیز بخوبی مشخص است. از این طریق استخراج ویژگی‌ها از داده‌های حجیم در زمان کمتری صورت می‌گیرد و در نتیجه بکارگیری الگوریتم در داده‌های حجیم را بسیار بهینه می‌کند و استخراج ویژگی‌های اصلی در مدت زمان کمتری صورت می‌گیرد که همین امر نیز باعث افزایش کارایی و دقت الگوریتم می‌شود. زیرا کاهش ویژگی در نهایت می‌تواند باعث افزایش کارایی دسته‌بند نیز شود. در روشهایی که فاقد این قابلیت هستند، الگوریتم یادگیری مجبور به استفاده از ویژگی‌هایی است که ارتباط خاصی با نفوذ ندارند و بعبارتی در تعیین حمله نقشی ندارند. این نوع از یادگیری در واقع یادگیری با داده‌های پرت خواهد بود و تاثیر منفی بر روی راهکار تشخیص نفوذ می‌گذارد. همانطورکه در نتایج حاصل از ارزیابی نیز ملاحظه می‌شود روش‌های دیگر از جمله SVM، به تنهایی کارایی چندان مناسبی نخواهند داشت و در نتیجه زمان اجرای به نسبت بالاتری دارند.

٤- نتیجه گیری

در این مقاله، برای تشخیص داده‌های مخرب یک راهکار تشخیص نفوذ ترکیبی با استفاده از الگوریتم غذایابی باکتری (BFO) و الگوریتم ماشین بردار پشتیبان (SVM) ارائه شده است. در واقع از این راهکار ترکیبی در جهت ارتقاء دقت تشخیص نفوذ در شبکه بهره برده شده است. با بهره‌گیری از این راهکار، ویژگی‌های مهم داده‌ها استخراج شده و از آن برای بهبود عملکرد کلی راهکار و

طبقه‌بندی اســتفاده می‌شــود. به کمک این رویکرد، کاهش ویژگی می‌تواند منجر به افزایش کارایی و دقت شــود. در ادامه این بخش، جزئیات این روش تشریح شده است.

در چارچوب پیشنهادی زمان و برچسب‌ها و مقیاس‌پذیری همچنین مرور کاربران لحاظ گردید و نتایج نشان داد که توضیح قابل قبولی در مقایسه با روش‌های پیشین می‌تواند داشته باشد. این طراحی معماری چارچوب بر مبنای خط لوله انجام شده است و دارای دو خط لوله می‌باشد که پردازش را به صورت موازی انجام می‌دهد و همین امر باعث افزایش سرعت روش پیشنهادی می‌باشد.

این برنامه توســط نرم افزار آپاچی اســپارک، یک چارچوب محاســباتی و نرم افزار متلب انجام شــده اســت. همچنین از یک اینترفیس مجازی وب برای مشاهده نتایج استفاده شده است. این روش را برای آشکارسازی داده‌های مخرب استفاده شده است و دیتاست استفاده شده حاوی داده‌های ۳۰ ماه در سایت virus total می‌باشد که آنالیز استاتیکی و دینامیکی ویروسها را انجام می‌دهد و ما در هر لحظه تغییرات برچسب‌ها را اندازه می‌گرفتیم و به محض تبدیل شدن به داده‌های مخرب آنها را آشکار می‌ساختیم.

جهت ادامه کار به عنوان تحقیقات آینده می‌توان موارد زیر را پیشنهاد داد:

- توسعه و بهینه‌سازی الگوریتم‌ها و مدل‌ها برای افزایش دقت تشخیص نفوذ.
- امکان اضافه کردن ویژگی‌ها و پارامترهای جدید به مدل به منظور بهبود عملکرد.
- انجام تحقیقات بیشتر در زمینه بهبود روش‌های بهینه‌سازی برای کاهش زمان آموزش مدل و افزایش سرعت تشخیص.
- اســتفاده از تکنولوژی‌های نوین مانند یادگیری عمیق (Deep Learning) و شــبکه‌های عصــبی برای بهبود کارایی تشــخیص نفوذ.

مراجع

[1] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979-1993, 2018. doi: 10.1109/TPAMI.2018.2858821

[2] R. A. Dara, T. Khan, J. Azim, O. Cicchello, and G. Cort, "A semi-supervised approach to customer relationship management," in *Artificial Intelligence and Soft Computing*, 2006, pp. 58-64.

[3] A. Dutt, S. Aghabozrgi, M. A. B. Ismail, and H. Mahroeian, "Clustering algorithms applied in educational data mining," *International Journal of Information and Electronics Engineering*, vol. 5, no. 2, p. 112, 2015. doi: 10.7763/IJIEE.2015.V5.513

[4] C. Guo, H. Tang, B. Niu, and C. B. P. Lee, "A survey of bacterial foraging optimization," *Neurocomputing*, vol. 452, pp. 728-746, 2021. doi:10.1016/j.neucom.2020.06.142

[5] H. Chen, Q. Zhang, J. Luo, Y. Xu, and X. Zhang, "An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine," *Applied Soft Computing*, vol. 86, p. 105884, 2020. doi: 10.1016/j.asoc.2019.105884.

[6] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine learning*: Elsevier, 2020, pp. 101-121. doi: 10.1016/B978-0-12-815739-8.00006-7.

[7] C. Campbell and Y. Ying, *Learning with support vector machines*. Springer Nature, 2022. doi:10.1007/978-3-031-01552-6.

[8] N. A. Seresht, R. Azmi, and B. Pishgoo, "A new clonal selection algorithm based on radius regularization of anomaly detectors," in *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)*, 2012: IEEE, pp. 497-502. doi: 10.1109/AISP.2012.6313798.

[9] P. Rahul, S. Kedia, Sarangi, and Monika, "Analysis of machine learning models for malware detection," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 23, no. 2, pp. 395-407, 2020. doi:10.1080/09720529.2020.1721870.

[10] K. Asrigo, L. Litty, and D. Lie, "Using VMM-based sensors to monitor honeypots," in *Proceedings of the 2nd international conference on Virtual execution environments*, 2006, pp. 13-23. doi:10.1145/1134760.1134765.

[11] I. Bello *et al.*, "Detecting ransomware attacks using intelligent algorithms: Recent development and next direction from deep learning and big data perspectives," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 8699-8717, 2021. doi:10.1007/s12652-020-02630-7.

[12] P. Kumar, G. P. Gupta, and R. Tripathi, "Toward design of an intelligent cyber attack detection system using hybrid feature reduced approach for iot networks," *Arabian Journal for Science and Engineering*, vol. 46, pp. 3749-3778, 2021. doi:10.1007/s13369-020-05181-3.
[13] M. Rabbani, Y. L. Wang, R. Khoshkangini, H. Jelodar, R. Zhao, and P. Hu, "A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing," *Journal of Network and Computer Applications*, vol. 151, p. 102507, 2020. doi:10.1016/j.jnca.2019.102507.
[14] J. O. Onah, M. Abdullahi, I. H. Hassan, and A. Al-Ghusham, "Genetic Algorithm based feature selection and Naïve Bayes for anomaly detection in fog computing environment," *Machine Learning with Applications*, vol. 6, p. 100156, 2021. doi:10.1016/j.mlwa.2021.100156.
[15] F. Jemili, "Intelligent intrusion detection based on fuzzy Big Data classification," *Cluster Computing*, vol. 26, no. 6, pp. 3719-3736, 2023. doi:10.1007/s10586-022-03769-y.
[16] G. T. Reddy *et al.*, "Analysis of dimensionality reduction techniques on big data," *Ieee Access*, vol. 8, pp. 54776-54788, 2020. doi:10.1109/ACCESS.2020.2980942.
[17] Y. Xu, Y. Sun, Z. Ma, H. Zhao, Y. Wang, and N. Lu, "Attribute selection based genetic network programming for intrusion detection system," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 26, no. 5, pp. 671-683, 2022. doi:10.20965/jaciii.2022.p0671.
[18] D. A. Salih, Y. A. Mohamed, and M. Bashir, "Enhancing intrusion detection system performance against low frequent attacks using FC-ANN algorithem," *Journal of Engineering Science and Technology*, vol. 18, no. 5, pp. 2411-2431, 2023.
[19] H.-M. Lee and S.-J. Lee, "A Study on Security Event Detection in ESM Using Big Data and Deep Learning," *International Journal of Internet, Broadcasting and Communication*, vol. 13, no. 3, pp.-٤٢ .٢٠٢١ ,٤٩ , doi:10.7236/IJIBC.2021.13.3.42
[20] M. Naveed *et al.*, "A Deep Learning-Based Framework for Feature Extraction and Classification of Intrusion Detection in Networks," *Wireless Communications and Mobile Computing*, vol. 2022, 2022. doi:10.1155/2022/2215852.
[21] A. A. Alqarni, "Toward support-vector machine-based ant colony optimization algorithms for intrusion detection," *Soft Computing*, vol. 27, no. 10, pp. 6297-6305, 2023. doi:10.1007/s00500-023-07906-6.
[22] W. Mao, Z. Cai, Y. Yang, X. Shi, and X. Guan, "From big data to knowledge: A spatio-temporal approach to malware detection," Computers & Security, vol. 74, pp. 167-183, 2018. doi:10.1016/j.cose.2017.12.005.
[23] M. M. Shetty and D. Manjaiah, "Advanced Threat Detection Based on Big Data Technologies," in Cyber Warfare and Terrorism: Concepts, Methodologies, Tools, and Applications: IGI Global, 2020, pp. 80-٨ .٨٢٢ ,doi:10.4018/978-1-5225-3015-2.ch001.
[24] M. Rabbani et al., "A review on machine learning approaches for network malicious behavior detection in emerging technologies," Entropy, vol. 23, no. 5, p. 529, 2021. doi:10.3390/e23050529.
[25] C.-H. Liu and W.-H. Chen, "The Study of Using Big Data Analysis to Detecting APT Attack," Journal of Computers, vol. 30, no. 1, pp. 206-222, 2019. doi:10.3966/199115992019023001020.
[26] G. Xu, W. Su, and Z. He, "An Efficient implementation of Network Malicious Traffic Screening based on Big Data Analytics," in 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), 2021: IEEE, pp. 1274-1277.doi: 10.1109/ICOSEC51865.2021.9591700
[27] J. Kevric, S. Jukic, and A. Subasi, "An effective combining classifier approach using tree algorithms for network intrusion detection," Neural Computing and Applications, vol. 28, no. Suppl 1, pp. 1051-1058, 2017. doi:10.1007/s00521-016-2418-1.

زیرنویس ها

[1] Wrapper
[2] Genetic Network Programming
[3] Neural Network
[4] Hidden Naïve Bayes
[5] dimensionality
[6] correlated features
[7] Naive Bayes
[8] Support Vector Machine

[9] Self-Organized Ant Colony Network
[10] Support Vector Machine
[11] Gradient Boosting Tree