Electronical & Cyber Defence
Vol. 12, No. 4, Winter 2024, Serial No. 48, pp. 89–102

Improving the speed of the intrusion detection system and maintaining its performance by reducing the data volume using kernel-based DBSCAN.

S.A. Shahzadeh Fazeli* , A. Ghoveh Nodoushan, J. Zarepour Ahmadabadi

*Associate Professor, Yazd University, Yazd, Iran

Received:2024 /10/05, Revised: 2024 /12 /12, Accepted: 2024/01/12, Published: 2024/02 /01

DOR: https://dor.isc.ac/dor/20.1001.1.23224347.1403.12.4.8.2

ABSTRACT

The Internet of Things (IoT) is a rapidly evolving technology that connects physical devices through networked systems. However, as IoT continues to expand, it poses various security challenges that require appropriate solutions to protect sensitive information and user privacy. This paper focuses on improving the speed of intrusion detection systems (IDS) as a critical solution for IoT security. In IDS, the large volume of data can slow down the learning process. In this paper, the DBSCAN clustering algorithm is modified by adding a minimum neighborhood parameter to reduce data samples in a targeted manner, aiming to enhance the speed of IDS and reduce learning time and costs. The parameters of the modified DBSCAN are tuned using a genetic algorithm. Experimental results on the Kaggle and NSL_KDD datasets demonstrate that the proposed model can maintain classification accuracy above 96% for the Kaggle dataset and above 92.51% for the NSL_KDD dataset, even with up to an 80% reduction in data volume. Additionally, computation time for the Kaggle dataset decreased from 458.09 ms to 47.21 ms, and for the NSL_KDD dataset from 995.2 ms to 223.60 ms. Thus, despite improvements in speed and reductions in time and cost, the model's optimal performance is maintained.

Keywords: : Internet of Things, Intrusion detection system, Clustering, Classification and regression tree algorithm, DBSCAN, Data reduction, RN_DBSCAN, Genetic algorithm, Kaggle dataset, NSL_KDD dataset.

* Corresponding Author Email: fazeli@yazd.ac.ir

«پدافند الکترونیکی و سایبری»
سال سیزدهم، شماره ۴، فصل زمستان ۱۴۰۳؛ ص۱۰۲-۸۹
شاپا الکترونیکی: ۸۹۷۹-۲۹۸۰ شاپا چاپی: ۴۳۴۷-۲۳۲۲

علمی – پژوهشی

بهبود سرعت سیستم تشخیص نفوذ و حفظ عملکرد آن از طریق کاهش حجم داده‌ها با استفاده از DBSCAN مبتنی بر هسته

سید ابوالفضل شاهزاده فاضلی [۱*]، اعظم قوه ندوشن [۲]، جمال زارع پور احمدآبادی [۳]

۱–دانشیار، ۲– دانشجوی کارشناسی ، ۳– استادیار، دانشگاه یزد، یزد، ایران

(دریافت: ۱۴۰۳/۰۷/۱۴، بازنگری: ۱۴۰۳/۰۹/۲۲، پذیرش: ۱۴۰۳/۱۰/۲۳، انتشار: ۱۴۰۳/۱۱/۱۳)

DOR: https://dor.isc.ac/dor/20.1001.1.23224347.1403.12.4.8.2

چکیده

اینترنت اشیاء یک فناوری به‌سرعت در حال تکامل است که دستگاه‌های فیزیکی را از طریق سیستم‌های شبکه‌ای به هم متصل می‌کند. بااین‌حال، همان‌طور که اینترنت اشیاء به گسترش خود ادامه می‌دهد، چالش‌های امنیتی مختلفی را ایجاد می‌کند که نیازمند راه‌حل‌های مناسب برای محافظت از اطلاعات حساس و حریم خصوصی کاربران است. این مقاله بر روی بهبود سرعت سیستم تشخیص نفوذ به‌عنوان یک راه‌حل حیاتی برای امنیت اینترنت اشیاء تمرکز دارد. در سیستم‌های تشخیص نفوذ، وجود حجم زیاد داده موجب کاهش سرعت یادگیری می‌شود. در این مقاله، الگوریتم خوشه‌بندی DBSCAN با افزودن پارامتر حداقل همسایگی جهت کاهش هدفمند نمونه‌ها اصلاح شده است که سعی در افزایش سرعت سیستم تشخیص نفوذ و کاهش زمان و هزینه یادگیری دارد. تنظیم پارامترهای DBSCAN اصلاح‌شده با الگوریتم ژنتیک انجام می‌شود. نتایج آزمایش‌ها بر روی مجموعه‌داده Kaggle و NSL_KDD نشان می‌دهد که مدل پیشنهادی قادر است با کاهش تا ۸۰٪ از حجم داده‌ها، دقت طبقه‌بندی را برای مجموعه‌داده Kaggle بالای ۹۶٪ و برای مجموعه‌داده NSL_KDD بالای ۹۲/۵۱٪ حفظ نماید. همچنین، زمان محاسبات برای مجموعه‌داده Kaggle از ۴۵۸/۰۹ ms به ۴۷/۲۱ ms و برای مجموعه‌داده NSL_KDD از ۹۹۵/۲ ms به ۲۲۳/۶۰ ms، کاهش‌یافته است. به‌این‌ترتیب، باوجود بهبود در سرعت و کاهش زمان و هزینه، عملکرد مطلوب مدل حفظ شده است.

کلیدواژه‌ها: اینترنت اشیاء، سیستم تشخیص نفوذ، الگوریتم درخت طبقه‌بندی و رگرسیون، DBSCAN، کاهش داده، RN_DBSCAN، مجموعه‌داده Kaggle، مجموعه‌داده NSL_KDD.

۱. مقدمه

اینترنت اشیاء[1] به شبکه‌ای از اشیاء فیزیکی اطلاق می‌شود که از طریق اینترنت به یکدیگر متصل می‌شوند [۱]. امنیت در اینترنت اشیاء بسیار مهم است؛ زیرا اینترنت اشیاء شامل دستگاه‌هایی است که به شبکه اینترنت متصل می‌شوند و اطلاعات حساس را به اشتراک می‌گذارند. اگر این دستگاه‌ها مورد حملات نفوذ قرار گیرند، می‌تواند منجر به دسترسی غیرمجاز به اطلاعات شخصی، کنترل غیرمجاز بر روی دستگاه‌ها و حتی خطرات فیزیکی مانند حمله به خودروهای هوشمند یا سیستم‌های خانه هوشمند شود. باتوجه‌به تعداد روزافزون دستگاه‌های متصل به اینترنت، تهدیدات امنیتی نیز روبه‌افزایش است. برای مثال، دستگاه‌های متصل به IoT اطلاعات حساسی را دربارهٔ کاربران، رفتارها، و محیط‌ها جمع‌آوری می‌کنند. حفظ حریم خصوصی این اطلاعات برای جلوگیری از سوءاستفاده و نفوذ به اطلاعات شخصی افراد بسیار حائز اهمیت است. همچنین سیستم‌های IoT به طور مداوم در معرض حملات امنیتی مانند حمله توزیع شده انکار از سرویس[2]، نفوذ به شبکه و سرقت اطلاعات هستند. بدون توجه به امنیت مناسب، دستگاه‌های IoT می‌توانند به دروازه‌های ورودی برای حملات به سایر سیستم‌ها تبدیل شوند. از طرفی در بعضی موارد مانند صنایع نفت و گاز، حمل‌ونقل هوایی و صنایع برق، سیستم‌های IoT نقش حیاتی در عملکرد و امنیت این صنایع را ایفا می‌کنند. نقص امنیتی در این دستگاه‌ها می‌تواند منجر به تخریب سیستم‌های حیاتی، ایجاد خسارت‌های جدی و حتی تهدید جان افراد شود [۲، ۳].

به‌طورکلی، امنیت در اینترنت اشیاء باید در همه جوانب توسعه و استفاده از دستگاه‌های متصل به اینترنت مدنظر قرار

[1] Internet of things (IOT)

[2] distributed denial-of-service (DDOS)

* رایانامه نویسنده مسئول: fazeli@yazd.ac.ir

گیرد. استفاده از سیستم‌های تشخیص نفوذ[1] مبتنی بر یادگیری ماشین می‌تواند به‌عنوان یک راه‌حل قوی و کارآمد برای شناسایی و پیشگیری از حملات نفوذ در اینترنت اشیاء مورداستفاده قرار گیرد.

سیستم تشخیص نفوذ، یک سیستم امنیتی هست که طراحی شده است تا به طور خودکار فعالیت‌های ناهنجار و هجمه‌های امنیتی را در یک شبکه یا سیستم کامپیوتری تشخیص دهد و آن‌ها را گزارش کند. هدف اصلی سیستم تشخیص نفوذ، شناسایی هجمه‌های امنیتی و محافظت از سیستم‌ها و شبکه‌ها در برابر تهدیدهای امنیتی است [۴، ۵].

یادگیری ماشین و سیستم تشخیص نفوذ دو حوزه مرتبط هستند که در کنار یکدیگر استفاده می‌شوند. یادگیری ماشین، به‌عنوان یک فرایند که به‌وسیلهٔ آن ماشین‌ها و سیستم‌ها قادر به یادگیری الگوها و رفتارهای مشکوک و استنتاج از داده‌ها هستند، در سیستم‌های تشخیص نفوذ به کار می‌رود [۶، ۷].

در سیستم‌های تشخیص نفوذ، کاهش حجم داده‌ها یکی از موارد مهم برای بهبود عملکرد و کارایی سیستم است. دلیل اصلی این امر، مربوط به حجم بسیار زیاد داده‌های شبکه و سیستم است که باید تحلیل شوند تا فعالیت‌های مشکوک و نفوذ را تشخیص دهند. زمانی که یک IDS فعال است، در هر لحظه داده‌های مختلفی از سیستم‌ها و شبکه‌ها را بررسی می‌کند. این داده‌ها شامل پروتکل‌های شبکه، لاگ‌ها، رویدادها و اطلاعات مختلف دیگر هستند. باتوجه‌به حجم بالای این داده‌ها، تحلیل و پردازش آن‌ها ممکن است زمان‌بر و پرهزینه باشد؛ بنابراین، کاهش حجم داده‌ها به‌منظور تسهیل فرایند تحلیل و تشخیص نفوذ اهمیت دارد. به‌طورکلی، با کاهش حجم داده‌ها در سیستم‌های تشخیص نفوذ، مزایای زیر به دست می‌آید:

۱. کارایی بالا: با کاهش حجم داده‌ها، فرایند تحلیل و تشخیص نفوذ سریع‌تر انجام می‌شود. زمان واکنش به فعالیت‌های مشکوک کاهش می‌یابد و امکان تشخیص به‌موقع تهدیدات امنیتی افزایش می‌یابد.
۲. صرفه‌جویی در منابع: با کاهش حجم داده‌ها، نیاز به منابع سخت‌افزاری و نرم‌افزاری کم‌تر می‌شود. این به معنی کاهش هزینه‌ها و استفاده بهینه از ظرفیت سیستم است.
۳. کاهش اشتباهات: با انتقال و تحلیل حجم کمتری از داده‌ها، احتمال بروز خطاها و اشتباهات انسانی کاهش می‌یابد. تمرکز بر داده‌های مهم‌تر و مشکوک‌تر، دقت و قابلیت تشخیص بیشتری را فراهم می‌کند.
۴. حفظ حریم خصوصی: با کاهش حجم داده‌ها، میزان اطلاعات حساسی که باید در دسترس سیستم‌های تشخیص نفوذ قرار گیرد کم‌تر می‌شود. این امر به حفظ حریم خصوصی کاربران و موجودیت‌های سازمان کمک می‌کند.

به طور خلاصه، کاهش حجم داده‌ها در سیستم‌های تشخیص نفوذ بهبود کارایی، کاربردی‌تر شدن و افزایش دقت و قابلیت تشخیص سیستم را به همراه دارد [۸]. در ادامه، کارهای مرتبط انجام شده توسط محققان دیگر در بخش ۲ شرح داده می‌شود و در بخش ۳ پیش‌نیازهای تحقیق بیان می‌شود. در بخش ۴ روش کاهش پیشنهادی به همراه شبه کد آن توضیح داده می‌شود و تنظیم پارامترهای آن توسط الگوریتم ژنتیک، انجام می‌شود و همچنین مدل استفاده شده به همراه روش‌های هرس و بهبود آن، ارائه می‌شود. در بخش ۵ نتایج عملی روش پیشنهادی به همراه نمودارهای بررسی عملکرد و همچنین نمودار مقایسه روش پیشنهادی با کارهای مشابه قبل، قرار داده شده است و در نهایت، در بخش ۶، نتایج کلی بیان می‌شود.

۲. مروری بر کارهای مرتبط

فرایند کاهش داده‌ها را می‌توان با استفاده از تکنیک‌های خوشه‌بندی انجام داد که هدف آن تقسیم داده‌ها به خوشه‌های کوچک برای یادگیری است. ویهارتو و همکاران [۹]، خوشه‌بندی را انجام داده‌اند و سپس از هر خوشه برای الگوریتم یادگیری لونبرگ - مارکوارت و الگوریتم شبه نیوتن استفاده نموده‌اند تا برای نتیجه‌گیری با الگوریتم بیز ساده، ترکیب شود. استفاده از خوشه‌بندی برای کاهش داده‌ها نیز توسط جان و همکاران انجام شد [۱۰] که ترکیبی از کاهش داده و کاهش ویژگی است. در تحقیقات فوق، با استفاده از خوشه‌بندی قصد دارند داده‌ها را به تعدادی خوشه تقسیم کنند، به‌طوری که مقدار داده برای هر خوشه کم‌تر از کل داده‌ها باشد. این رویکرد، خوشه‌بندی کل داده‌ها را کاهش نمی‌دهد، بلکه داده‌ها را به تعداد ثابتی از خوشه‌ها تقسیم می‌کند و اگر همه داده‌های هر خوشه جمع شوند، مانند داده‌های اصلی باقی می‌مانند و در نتیجه پیچیدگی محاسبات را افزایش می‌دهد. استفاده از خوشه‌بندی، علاوه بر تقسیم داده‌ها، می‌تواند مبنایی برای کاهش داده‌ها باشد تا از حجم داده‌ها کاسته شود.

برخی از مدل‌های کاهش داده با استفاده از الگوریتم‌های خوشه‌بندی، در ادامه توصیف می‌شوند. لی‌خاک و همکاران [۱۱]، یک مدل کاهش داده را با ترکیب الگوریتم شباهت نزدیک‌ترین همسایه مشترک[2] و الگوریتم خوشه‌بندی بر اساس

[1] Intrusion Detection System (IDS)

[2] Sharing Nearest Neighbor (SNN)

تراکم فاصله‌ها برای برنامه‌های کاربردی نویزدار[1] ایجاد کردند. نقطه‌ضعف این مدل این است که در صورت وجود حجم زیادی از داده‌ها، وجود دو فرایند یعنی SNN و DBSCAN، سرعت محاسبات را به‌شدت تحت‌تأثیر قرار می‌دهند. توسعه انجام شده توسط وانگ و همکاران [۱۲]، کاهش در مقدار داده‌ها را با کاهش نمونه تصادفی ترکیب می‌کند و از روش تحلیل مؤلفه اصلی برای خوشه‌بندی بیشتر با استفاده از الگوریتم خوشه‌بندی c-میانگین استفاده می‌کند. با اشاره به الگوریتم خوشه‌بندی مورداستفاده، نتیجه حاصل از تحقیق [۱۱] بهتر از [۱۲] بوده است؛ به دلیل اینکه عملکرد الگوریتم خوشه‌بندی DBSCAN نسبت به c_میانگین، به‌ویژه برای داده‌های نویزدار بهتر است [۱۳، ۱۴].

الگوریتم کاهش مبتنی بر خوشه‌بندی دیگری وجود دارد که از طریق خوشه‌های همگن[2] کاهش می‌یابد. RHC، بر اساس مفهوم همگنی است، اما از الگوریتم خوشه‌بندی k_میانگین استفاده می‌کند [۱۷-15]. الگوریتم برای داده‌های غیرهمگن خوشه‌بندی را انجام می‌دهد به‌طوری که همه داده‌ها همگن می‌شوند. نقطه ضعف این الگوریتم این است که استفاده از k_میانگین با مجموعه داده‌های پرت و نویز به خوبی کار نمی‌کند. DBSCAN در مقایسه با k_میانگین از قابلیت حساسیت خوبی برخوردار است [۱۴]. بر اساس تحقیقات انجام شده توسط اوجیاروگلو وهمکاران [۱۷]، کاهش داده‌ها با استفاده از الگوریتم کاهش خوشه‌های همگن برای تقریباً ۸۰٪ کاهش داده‌ها انجام شده است و از الگوریتم‌های طبقه‌بندی شبکه عصبی، ماشین بردار پشتیبان و k_نزدیک‌ترین همسایگان استفاده شده است که دقت حاصل زیر ۹۰٪ می‌باشد.

الگوریتم RHC، روی داده‌های نویزی ضعیف عمل می‌کند. باتوجه‌به این ضعف، الگوریتم کاهش خوشه‌های همگن با عنوان «ویرایش و کاهش از طریق خوشه‌های همگن[3]»، توسط اوجیاروگلو و همکاران، توسعه یافت [۱۸]. در الگوریتم eRHC، وقتی یک نقطه داده تشکیل یک خوشه می‌دهد، به‌عنوان نویز در نظر گرفته شده و دور انداخته می‌شود. در مطالعه [۱۸]، نشان می‌دهد که کاهش داده با الگوریتم eRHC می‌تواند عملکردی بادقت کمتر از ۹۰٪، زمانی که با الگوریتم طبقه‌بندی ماشین بردار پشتیبان ترکیب شود، به دست آورد.

ویهارتو و همکاران [۱۹]، جهت کاهش داده، از الگوریتم‌های نمونه‌گیری[4] و برش داده[5] استفاده نمودند. نمونه‌گیری یک روش ناپارامتری کاهش تعداد است که با نمایش مجموعه‌داده‌های بزرگ به‌عنوان زیر مجموعه‌داده‌های تصادفی کوچک‌تر، داده‌ها را کاهش می‌دهد. بااین‌حال تکنیک نمونه‌گیری هنوز مشکلات متعددی در نمایش داده‌ها دارد، یعنی نمونه‌گیری می‌تواند برخی از داده‌هایی را که ممکن است با داده‌های گرفته شده همگن نباشد را حذف کند. این بر میزان دقت در نتایج طبقه‌بندی تأثیر می‌گذارد.

در سیستم تشخیص نفوذ که حاوی داده‌های پرت و نویز زیادی هست، انتخاب الگوریتم برای خوشه‌بندی بسیار مهم است. از بین روش‌های خوشه‌بندی، خوشه‌بندی DBSCAN به دلیل توانایی در تشخیص و حذف داده‌های پرت و کشف خوشه‌های پیچیده، برای سیستم تشخیص نفوذ که بسیار حجیم و نویزی می‌باشد، مناسب‌تر است. در [۱۹] کاهش داده با استفاده از الگوریتم m_DBSCAN انجام شده است این کاهش با اصلاح الگوریتم DBSCAN بوده است که با درنظرگرفتن نقاط همگن برای هر نقطهٔ بازدید نشده و حذف این نقاط انجام می‌شود. این روش نسبت به روش‌های کاهش Slice Data و Sampling بهتر است؛ اما به دلیل اینکه نقاط همگن با هر نقطه را کاهش می‌دهد، باعث ایجاد نقاط نویزی می‌شود.

۳. پیش نیاز تحقیق

داده‌های مورداستفاده: این تحقیق بر اساس داده‌های ارزیابی تشخیص نفوذ که نتیجه شبیه‌سازی محیط شبکه نظامی LAN نیروی هوایی ایالات متحده است، انجام شده است. شرکت دارپا که متعلق به وزارت دفاع ایالات متحده می‌باشد، اولین داده‌های استاندارد را برای بررسی و ارزیابی سیستم‌های تشخیص نفوذ، جمع‌آوری نمودند. در این پژوهش از مجموعه‌داده‌های Kaggle و NSL_KDD استفاده شده است. مجموعه‌داده Kaggle شامل ۲۵،۱۹۲ نمونه برای داده‌های آموزش و ۲۲،۵۴۴ نمونه برای داده‌های آزمون هست. همچنین مجموعه‌داده NSL_KDD نسخه جدید از مجموعه‌داده KDD-99 است که برای حل برخی از مشکلات ذاتی مجموعه‌داده KDD-99 پیشنهاد شده است و از توزیع مناسبی برخوردار است.

الگوریتم DBSCAN: یکی از الگوریتم‌های بسیار قدرتمند در زمینه خوشه‌بندی، الگوریتم «خوشه‌بندی بر اساس تراکم

[1] Density Based Spatial Clustering algorithm for Applications with Noise (DBSCAN)
[2] Reduced Homogeneous Clusters (RHC)
[3] Editing and Reduction through Homogeneous Clusters (eRHC)
[4] Sampling
[5] Slice Data

فاصله‌ها برای برنامه‌های کاربردی نویزدار[1]» است. این الگوریتم بر اساس چگالی داده‌ها، خوشه‌بندی را انجام می‌دهد [۲۰]. الگوریتم DBSCAN نیاز به دو پارامتر نقطه حداقل[2] و اپسیلون[3] دارد. هر نقطه از داده با نقاط دیگر، فاصله‌ای دارد؛ اگر این فاصله بین دو نقطه مفروض، کم‌تر از اپسیلون باشد، به‌عنوان همسایه آن نقطه در نظر گرفته می‌شود. هر نقطه مفروض که به تعداد نقطه حداقل یا بیشتر، همسایه داشته باشد، یک نقطه مرکزی محسوب می‌شود. در واقع با استفاده از این دو پارامتر، حداقل چگالی یک خوشه تعیین می‌شود [۲۱].

الگوریتم ژنتیک: الگوریتم ژنتیک یک الگوریتم بهینه‌سازی است که از انتخاب طبیعی و از مفهوم تکامل الهام‌گرفته شده است. این الگوریتم یک جستجو مبتنی بر جمعیت هست و از نظریه داروینی در مورد بقای بهترین‌ها در طبیعت تقلید می‌کند [۲۲].

الگوریتم CART: یکی از محبوب‌ترین و درعین‌حال ساده‌ترین الگوریتم‌های درخت‌های تصمیم، درخت طبقه‌بندی و رگرسیون[4] است که کاربردهای فراوانی در طبقه‌بندی و رگرسیون دارد. این الگوریتم بر اساس درخت‌های دودویی (باینری) بنا نهاده شده است. الگوریتم درخت طبقه‌بندی و رگرسیون، برای ساخت درخت تصمیم، داده‌ها را به قسمت‌های دوتایی تقسیم کرده و بر اساس آن‌ها درخت دودویی را می‌سازد. این درخت (و البته درخت‌های دیگر) می‌تواند پایه‌ای برای الگوریتم‌های پیچیده‌تر مانند جنگل تصادفی[5] باشد [۲۳].

۴. روش پیشنهادی

مسئله موردنظر در این پژوهش، بهینه‌سازی سیستم‌های تشخیص نفوذ است. در واقع هدف تعیین راهبردی برای حداکثرسازی عملکرد سیستم تشخیص نفوذ می‌باشد؛ به‌طوری که بتوان زمان و هزینه را تاحدامکان کاهش داد.

برای بهبود الگوریتم یادگیری ماشینی که قادر به کاوش اطلاعات و در پی آن تعیین الگویی برای تشخیص نفوذ باشد، نیازمند به انتخاب برخی نمونه‌هایی که بتوانند الگوریتم طبقه‌بندی را به‌خوبی آموزش داده و از حجم مناسبی برخوردار باشند، حس می‌شود؛ بنابراین در این مقاله از بهبود خوشه‌بندی DBSCAN جهت حذف سروصداهای موجود و همچنین کاهش حجم داده‌ها به‌منظور بهبود زمان و فضا و همچنین کاهش هزینه‌های موجود، استفاده می‌گردد.

روند کلی کار انجام شده، در شکل (۱) نشان داده شده است. همان‌طور که در قسمت (الف) نمودار گردشی شکل (۱) مشخص است، بعد از جمع‌آوری داده‌ها و انجام عملیات پیش‌پردازش، داده‌ها براساس کلاس تفکیک شده و سپس هر گروه داده به‌صورت مجزا کاهش می‌یابد. بعد از کاهش بر روی هر گروه، داده‌های کاهش یافته، ادغام می‌شوند و به دو بخش برای یادگیری و آزمون روش پیشنهادی، تقسیم می‌شوند. سپس داده‌های نرمال شده با الگوریتم CART طبقه‌بندی می‌شوند. در نهایت، عملکرد سیستم تشخیص نفوذ به‌وسیله معیارهای ارزیابی مختلف، مورد بررسی و آزمایش قرار می‌گیرد و نتیجه نهایی به‌دست می‌آید.

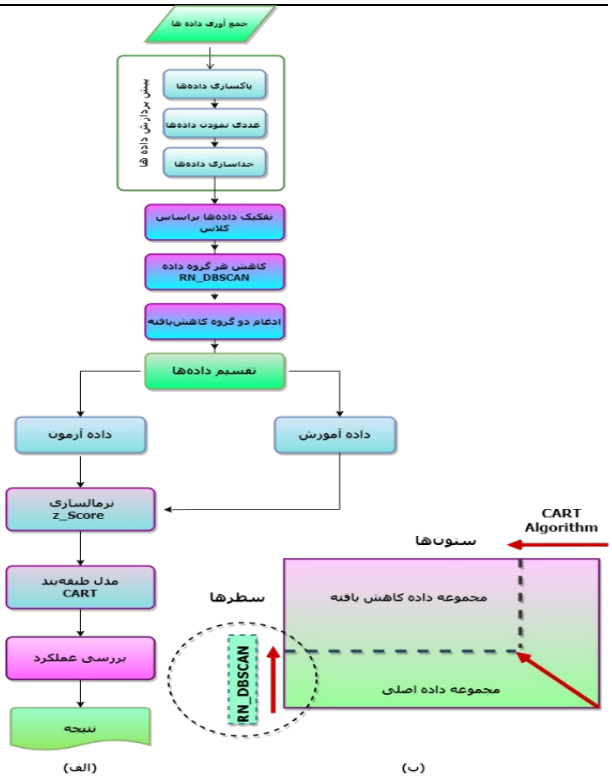
مرحله پیش‌پردازش به سه بخش؛ پاک‌سازی داده‌ها، عددی‌سازی داده‌ها و جداسازی داده‌ها تقسیم می‌شود. عددی‌سازی داده‌ها با استفاده از روش «وان هات[6]» انجام شده است. در روش کدبندی وان‌هات، به‌ازای هر مقدار از ویژگی غیرعددی، ستونی به مجموعه‌داده اضافه می‌گردد، درصورتی‌که نمونه، آن ویژگی را داشته باشد، مقدار یک و در غیر این صورت، مقدار صفر قرار می‌گیرد. همچنین از روش نرمال‌سازی «نمره استاندارد[7]» استفاده شده است. این روش سعی دارد، برای یک مجموعه‌داده، مقدارهایی را به دست آورد که دارای میانگین صفر و واریانس یا انحراف معیار یک باشد؛ بنابراین اگر میانگین داده‌های اصلی برابر با μ و انحراف معیار آن‌ها نیز σ باشد، مقدار Z را براساس رابطه (۱) می‌توان به‌دست آورد.

$$z = \frac{x - \mu}{\sigma} \qquad (۱)$$

به‌این‌ترتیب مشخص است که داده‌های تبدیل یافته Z دارای میانگین صفر و واریانس یک هستند.

باتوجه‌به شکل (۱) قسمت (ب)، مرحله کاهش داده‌ها از دو بخش تشکیل می‌شود، بخشی از کاهش ابعادی از نظر مقدار داده می‌باشد که با استفاده از الگوریتم اصلاح شده DBSCAN انجام می‌شود و بخشی دیگر از کاهش ابعاد، مربوط به الگوریتم CART است. این الگوریتم علاوه بر آموزش و طبقه‌بندی داده‌ها، کاهش ویژگی نیز انجام می‌دهد.

[1] Density Based Spatial Clustering algorithm for Applications with Noise (DBSCAN)
[2] Min Point (MinPts)
[3] Epsilon (Eps)
[4] Classification And Regression Tree (CART)
[5] Random forest
[6] One hot encoding
[7] Z-score



شکل (۱): نمودار گردشی روش پیشنهادی تحقیق

۱-۴. کاهش ابعاد با الگوریتم RN_DBSCAN

الگوریتم DBSCAN برای انجام فرایند کاهش، مورد اصلاح قرار می‌گیرد و با عنوان «کاهش به‌وسیلهٔ همسایگان در الگوریتم خوشه‌بندی بر اساس تراکم فاصله‌ها برای برنامه‌های کاربردی نویزدار[1]» مطرح می‌شود.

اصلاح الگوریتم DBSCAN با افزودن پارامتری به نام حداقل همسایگی[2] انجام شده است که این پارامتر برای تعیین فاصله چگالی تا نقطه مرکزی مورداستفاده قرار می‌گیرد و سپس برای حذف علامت‌گذاری می‌شود. در واقع سطح چگالی اطراف نقطه مرکزی به‌صراحت به‌عنوان مقدار حداقل همسایگی تعریف می‌شود. کاهش داده‌ها در این روش، با حذف برخی از داده‌ها که تراکم بیشتری نسبت به داده‌های دیگر دارند، انجام می‌شود. این تراکم باتوجه‌به توزیع داده‌ها تعیین می‌شود.

جهت تعیین پارامترهای Eps، $MinPts$ و $MinNr$ محاسبه ماتریس فاصله طبق رابطه (۲) برای هر داده انجام می‌گیرد، که در آن d_{ij} فاصله بین نقطه i و نقطه j، x_{vi} مختصات نقطه i برای بعد v و x_{vj} مختصات نقطه j برای بعد v می‌باشد [۲۴].

$$d_{ij} = \sqrt{\sum_{v=1}^{n} (x_{vi} - x_{vj})^2} \qquad (۲)$$

شبه کد مربوط به فرایند کاهش داده‌ها با روش RN_DBSCAN در الگوریتم‌های ۱ و ۲ نشان داده شده است. در این الگوریتم‌ها، مواردی که پررنگ‌تر نوشته شده‌اند، تفاوت الگوریتم پیشنهادی این مقاله و الگوریتم DBSCAN است.

الگوریتم ۱: RN-DBSCAN

ورودی: پارامترهای $D, Eps, MinPts,$ **MinNr** را وارد کنید.
خروجی: تشکیل خوشه و یافتن نقاط مرکزی.
$C = 0$ قرار دهید.
برای هر نقطه p ملاقات نشده در مجموعه داده D مراحل زیر را انجام دهید.
  p را به‌عنوان بازدید شده علامت‌گذاری کنید.
  همسایه‌های p به شعاع Eps را در N قرار دهید.
  اگر تعداد N کوچک‌تر از $MinPts$ **آنگاه**
    p را به‌عنوان $Noise$ علامت‌گذاری کنید.
  درغیراینصورت
    p را در خوشه C جدید قرار دهید.
    همسایه‌های p به شعاع MinNr را در X قرار دهید و حذف نمایید.
    خوشه را با ExpandCluster$(P, N, C, Eps, MinPts,$ **MinNr**$, X)$ گسترش دهید.
  پایان اگر
پایان حلقه

الگوریتم ۲: ExpandCluster

ورودی: $P, N, C, Eps, MinPts,$ **MinNr**$, X$ را وارد کنید.
خروجی: گسترش خوشه‌ها و یافتن نقاط همگن با نقاط مرکزی جهت کاهش.
p را به خوشه C اضافه کنید.
برای هر نقطه p' در N
  اگر p' ملاقات نشده **آنگاه**
    p' را به‌عنوان ملاقات شده علامت‌گذاری کنید.
    همسایه‌های p' به شعاع Eps را در N' قرار دهید.
    اگر تعداد N' بزرگتر مساوی با $MinPts$ **آنگاه**
      N' را با N ترکیب کرده در N قرار دهید.
      همسایه‌های 'p به شعاع MinNr را در deleteNeighbor قرار دهید و حذف نمایید.
    پایان اگر
  پایان اگر
  اگر p' در هیچ خوشه‌ای نیست **آنگاه**
    p' را به خوشه C اضافه کنید.
  پایان اگر
پایان حلقه

سازوکار این الگوریتم مشابه الگوریتم خوشه‌بندی DBSCAN است، با این تفاوت که نقطه P علاوه بر شعاع Eps، شعاع $MinNr$ را جهت یافتن نقاط همگن با نقطه مرکزی، یعنی نقاطی که بسیار نزدیک و مشابه نقاط مرکزی هستند و به اصطلاح هم‌پوشانی دارند، محاسبه می‌نماید، تا در نهایت از این نقاط برای کاهش استفاده کند. در واقع کاری که انجام شده است، استفاده از الگوریتم DBSCAN، جهت خوشه‌بندی و یافتن نقاط مرکزی می‌باشد، تا بتوان از این نقاط برای کاهش بهره‌مند شد، به این صورت که نقاطی که در شعاع $MinNr$ به مرکز نقاط مرکزی

[1] Reduction by Neighboring in DBSCAN (RN_DBSCAN)
[2] MinNeighborhood (MinNr)

هستند، کاهش یابد.

دلیل استفاده از نقاط مرکزی جهت کاهش، تراکمی هست که در اطراف این نقاط وجود دارد؛ بنابراین حذف نقاط همگن به شعاع $MinNr$ از نقاط مرکزی، اطلاعاتی را از دست نداده و می‌توان مدل یادگیر را به‌وسیله نقاط باقی‌مانده به‌طور مناسبی آموزش داد. همچنین به دلیل داشتن تراکمی که در اطراف نقاط مرکزی وجود دارد، می‌توان مطمئن بود که با حذف نقاط اطراف نقاط مرکزی، آن نقطه تبدیل به نویز نخواهد شد.

۲-۴. تنظیم پارامترهای RN_DBSCAN با استفاده از الگوریتم ژنتیک

همواره یکی از چالش‌های موجود در الگوریتم DBSCAN، تنظیم پارامترهای آن است. حال با افزودن پارامتر جدید به این الگوریتم جهت کاهش، این چالش بیش‌ازپیش مشهود است؛ بنابراین به‌منظور استفاده از روش کاهش RN_DBSCAN و به‌دست‌آوردن نتیجه مطلوب، تنظیم پارامترهای آن از اهمیت ویژه‌ای برخوردار است.

در این پژوهش برای تنظیم پارامترهای الگوریتم RN_DBSCAN از الگوریتم ژنتیک بهره گرفته شده است. الگوریتم ژنتیک علاوه بر تنظیم خودکار پارامترها قادر است پارامترهای بهینه را به دست آورد. در واقع با استفاده از این الگوریتم، برای هر درصد کاهش، نمونه‌هایی که بار اطلاعاتی کمتری دارند حذف می‌شوند؛ بنابراین، می‌توان گفت کارایی و دقت طبقه‌بند نیز علاوه بر تعداد داده‌های باقی‌مانده از کاهش، مدنظر قرار گرفته می‌شود.

برای تعیین اثر کاهش داده‌ها، فرایند کاهش داده‌ها، با ایجاد دو رویکرد، یعنی کاهش داده‌ها بدون درنظرگرفتن برچسب داده‌ها و کاهش داده‌ها با درنظرگرفتن برچسب‌های داده، انجام می‌شود. کاهش با درنظرگرفتن برچسب‌های داده با گروه‌بندی داده‌ها بر اساس برچسب‌ها انجام می‌گیرد؛ به این منظور که درصد کاهش مدنظر، بر روی هر گروه از کلاس‌ها به‌صورت مجزا انجام می‌شود. باتوجه‌به این دو حالت، برای الگوریتم ژنتیک، توابع برازش متفاوتی تعریف می‌شود که در ادامه توضیح داده می‌شود.

۱. بدون درنظرگرفتن برچسب داده: تابع برازش در این حالت، طبق رابطه (۳) محاسبه می‌شود. این تابع برازش کلی، از مجموع دو تابع برازش وزن‌دار به‌دست می‌آید. در رابطه (۳)، $W1$ و $W2$ وزن‌هایی هستند که به هر یک از توابع برازش $fitness1$ و $fitness2$ نسبت داده می‌شوند. $W1$ وزن تعداد نمونه‌های کاهش یافته و $W2$ وزن دقت می‌باشد.

$$TotalFitness = w_1 fitness_1 + w_2 fitness_2 \quad (۳)$$

توابع برازش $fitness1$ و $fitness2$ طبق رابطه (۴) تعریف شده اند:

$$fitness_1 = |N_{main} - N_{remain}|, \quad fitness_2 = |1 - accuracy| \quad (۴)$$

که در آن، $fitness1$ تفاضل N_{main} تعداد مطلوب داده‌های کاهش یافته و N_{remain} تعداد داده‌های کاهش یافته با RN_DBSCAN است و $fitness2$ از تفاضل دقت به‌دست آمده با دقت کل، که یک درنظر گرفته شده است، به‌دست می‌آید.

۲. با درنظرگرفتن برچسب داده: در این حالت تابع برازش طبق رابطه (۵) محاسبه می‌شود:

$$TotalFitness = w_1 fitness_1 + w_2 fitness_2 + w_3 fitness_3 \quad (۵)$$

که در آن $W1$ و $W2$ و $W3$ به ترتیب وزن‌های $fitness1$ و $fitness2$ و $fitness3$ هستند.

مقادیر $fitness1$ و $fitness2$ و $fitness3$ از روابط ۶ تا ۸ بدست می‌آیند.

$$fitness_1 = |N_{main_n} - N_{remain_n}|, \quad (۶)$$

$$fitness_2 = |N_{main_{an}} - N_{remain_{an}}|, \quad (۷)$$

$$fitness_3 = |1 - accuracy|. \quad (۸)$$

در رابطه(۶)، $fitness1$ از تفاضل مقدار داده‌های کاهش یافته با روش RN_DBSCAN از کلاس نرمال (N_{remain_n})، با تعداد کاهش مطلوب از این گروه (N_{main_n}) به‌دست می‌آید. همچنین با توجه به رابطه (۷)، $fitness2$ از تفاضل مقدار داده‌های کاهش یافته با روش RN_DBSCAN از کلاس ناهنجار ($N_{remain_{an}}$) با تعداد کاهش مطلوب از این گروه ($N_{main_{an}}$) حاصل می‌شود. $fitness3$ در رابطه (۸)، نیز بهترین دقت حاصل از درصد کاهش مدنظر را بررسی می‌کند.

نتایج تنظیم پارامترهای RN_DBSCAN با استفاده از الگوریتم ژنتیک، جهت کاهش داده‌ها برای مجموعه‌داده Kaggle، بدون درنظرگرفتن برچسب داده در جدول (۱) و با درنظرگرفتن برچسب داده در جدول (۲)، قرار گرفته است. همچنین تنظیم پارامترهای RN_DBSCAN با استفاده از الگوریتم ژنتیک، جهت کاهش داده‌ها برای مجموعه‌داده NSL-KDD، با درنظرگرفتن برچسب داده انجام گرفت که نتایج آن در جدول (۳) نشان‌داده‌شده است.

جدول (۱). نتایج کاهش داده‌ها بدون درنظر گرفتن کلاس (Kaggle)

پارامترهای RN_DBSCAN				
تعداد داده‌ها	*MinNr*	*MinPts*	*Eps*	کاهش داده (٪)
۲۵۱۹۲	-	-	-	۰
۲۳۹۳۲	۲/۴۵	۲۱	۲/۷	۵
۲۰۱۵۴	۳/۶	۵۹	۵/۵۹	۲۰
۱۷۶۳۴	۱۰/۲۷	۹۷	۱۱/۳۵	۳۰
۱۲۸۴۸	۱۱/۱۷	۱۲	۲۰/۰۲	۴۹
۱۰۰۷۷	۱۴/۹۰۵	۲	۲۰	۶۰
۵۰۳۸	۴۲/۶۲	۲	۷۶/۰۴	۸۰

جدول (۲). نتایج کاهش داده‌ها با درنظر گرفتن کلاس (Kaggle)

	ناهنجار				طبیعی			
تعداد کلی داده‌ها	تعداد داده‌ها	*MinNr*	*MinPts*	*Eps*	تعداد داده‌ها	*MinNr*	*MinPts*	*Eps*	کاهش داده (٪)
25192	13000	-	-	-	12192	-	-	-	0
23932	12350	1/32	12	2	11582	5/75	23	12/55	5
20154	10400	2/02	58	5	9754	13/55	4	15/3	20
17634	9100	1/94	27	8/01	8534	20/79	4	22/4	30
12848	6630	3/01	21	9/03	6218	29/19	1	56/62	49
10077	5200	4/11	40	23/7	4877	38/9	2	56/09	60
5038	2600	10/23	34	45/05	2438	74/87	2	85/07	80

جدول (۳). نتایج کاهش داده‌ها با درنظر گرفتن کلاس (NSL_KDD)

	ناهنجار				طبیعی			
تعداد کلی داده‌ها	تعداد داده‌ها	*MinNr*	*MinPts*	*Eps*	تعداد داده‌ها	*MinNr*	*Min Pts*	*Eps*	کاهش داده (٪)
125973	58630	-	-	-	67343	-	-	-	0
119675	55699	3/27	490	4/84	63976	3/25	490	36/83	5
100778	46904	2	296	4/85	۳۸۷۴۵	36/91	208	37/48	20
94044	41041	4	296	48/86	۴۷۱۴۰	10/8	143	70/0	30
64246	29901	2/81	150	5/55	۳۴۳۴۵	24/51	132	78/75	49
50389	23452	3/31	490	12/11	۲۶۹۳۷	71/06	208	90/93	60
25195	11726	5/25	172	19/01	۱۳۴۶۹	68/48	531	435/08	80

باتوجه‌به پارامترهای به‌دست‌آمده برای هر مجموعه‌داده در جداول (۱) تا (۳)، می‌توان مشاهده نمود، هراندازه مقدار پارامترهای Eps و $MinNr$ افزایش یابد، کاهش بیشتر می‌شود و هر اندازه مقدار پارامتر $MinPts$ افزایش یابد، کاهش کم‌تری رخ می‌دهد. دلیل این امر به این خاطر است که با افزایش Eps، شعاع بزرگ‌تر شده، بنابراین تعداد همسایه‌های اطراف نقطه مورد نظر بیشتر

* رایانامه نویسنده مسئول: fazeli@yazd.ac.ir

خواهد شد و در نتیجه با فرض ثابت بودن مقدار $MinPts$، نقاط بیشتری به‌عنوان نقاط مرکزی، درنظر گرفته می‌شوند و با افزایش $MinNr$ شعاع اطراف نقطه‌ی مرکزی بزرگ‌تر شده بنابراین نقاط بیشتری حذف خواهند شد. در نتیجه با افزایش مقدار این دو پارامتر، میزان کاهش داده‌ها افزایش می‌یابد. اما با افزایش پارامتر $MinPts$، به دلیل اعمال سخت‌گیری بیشتر در انتخاب نقاط به‌عنوان نقطه‌ی مرکزی، تعداد کم‌تری نقطه مرکزی به‌دست می‌آید، بنابراین با افزایش پارامتر $MinPts$ با فرض ثابت بودن دو پارامتر دیگر، نقاط کمتری کاهش می‌یابند.

هدف اصلی، کاهش نقاطی است که بسیار به نقطه مرکزی نزدیک بوده و در واقع مشابه و همگن با آن است؛ بنابراین به مرکز نقطهٔ مرکزی و به شعاع کوچک‌ترین همسایگی، نقاط کاهش می‌یابند. نقاط موجود در این شعاع همان نقاط همگن با نقطه مرکزی بوده و باید از مجموعه‌داده کاهش یابند. پر واضح است که مقدار اپسیلون می‌بایست از کوچک‌ترین همسایگی، بیشتر باشد. همچنین مقدار نقطه حداقل که تعداد همسایه‌ها است صحیح در نظر گرفته می‌شود.

۴-۳ مدل CART

در این پژوهش از مدل CART جهت آموزش و آزمون عملکرد سیستم تشخیص نفوذ استفاده شده است. الگوریتم CART یک الگوریتم طبقه‌بندی است که از رویکرد طبقه‌بندی درخت تصمیم استفاده می‌کند. الگوریتم CART برای سیستم‌های تشخیص نفوذ، دارای عملکرد بسیار مناسبی است

[۲۵]. در کل می‌توان گفت این الگوریتم دارای برتری‌هایی نسبت به الگوریتم‌های دیگر در این مسئله می‌باشد [۲۸- ۲۶]. برای استفاده از مدل CART، داده‌ها با نسبت ۷۰٪ برای آموزش و ۳۰٪ برای اعتبارسنجی به اشتراک گذاشته می‌شوند و سپس با استفاده از الگوریتم CART طبقه‌بندی انجام می‌شود.

۴-۳-۱ تشکیل درخت طبقه‌بندی

داده‌های نمونه برای یادگیری ناهمگن به‌عنوان مبنایی برای تشکیل یک درخت طبقه‌بندی استفاده می‌شوند. در این مرحله، با رعایت قوانین مرتب‌سازی و معیارهای تقسیم‌بندی، مرتب‌سازی از داده‌هایی برای یادگیری انتخاب می‌شوند که منجر به بیش‌ترین کاهش در سطوح ناهمگونی شوند.

تعیین مناسب‌بودن تقسیم $\emptyset(s,t)$ شاخه کاندید s در نقطه تصمیم t به‌صورت معادلات (۹) تا (۱۱) محاسبه می‌شود [19]. که در آن t_{left} کاندید شاخه چپ از نقطه تصمیم t، t_{right} کاندید شاخه سمت راست از نقطه تصمیم t، $p(j|t_{left})$ احتمال j در گره شاخه سمت چپ، $p(j|t_{right})$ احتمال j در شاخه سمت راست است.

$$N\emptyset(s,t) = 2P_L P_R Q(s|t)\,, \qquad (۹)$$

$$\emptyset(s,t) = 2P_L P_R Q(s|t)\,, \qquad (۱۰)$$

$$\emptyset(s|t) = \sum_{j=1}^{The number of catgorics} \left|p(j|t_{left}) - p(j|t_{right})\right|. \qquad (۱۱)$$

گره شاخه مرتب‌کننده‌ای که به مقادیر $\emptyset(s,t)$ بالاتر منجر می‌شود، مرتب‌سازی بهتری است و به‌عنوان تقسیم‌گر/شاخه انتخاب می‌شود. گره t مانند تنها یک مشاهده در هر گره فرزند، تبدیل به یک گره پایانی می‌شود، اگر کاهش قابل توجهی در ناهمگنی/ناخالصی یا حداقل حد n، وجود نداشته باشد. حداقل تعداد محدودیت‌ها در ترمینال انتهایی به‌طور کلی، پنج است. هنگامی که گره‌های ترمینال پیدا می‌شوند، تشکیل درخت تصمیم خاتمه می‌یابد.

۴-۳-۲ هرس درخت تصمیم

هدف اصلی از ساختن درخت طبقه‌بندی، ایجاد یک مدل طبقه‌بندی‌کننده به شکل درخت تصمیم است که کم‌ترین مقدار خطای طبقه‌بندی یا نزدیک به صفر را داشته باشد. بااین‌حال، درخت تصمیم در توصیف ساختار داده، بسیار پیچیده است. یکی از راه‌های تعیین درخت طبقه‌بندی بهینه، بدون کاهش دقت مدل طبقه‌بندی‌کننده، هرس نمودن درخت طبقه‌بندی است. سطح اهمیت درخت یا بهره آن، بر اساس حداقل پیچیدگی هزینه فرموله شده در معادله (۱۲) اندازه‌گیری می‌شود [19].

$$D_\alpha(T_K) + \alpha\left|\tilde{T}_K\right| \qquad (۱۲)$$

که در آن:

$D_\alpha(T_K)$ = انحراف از درخت جزئی (T_K)

$\tilde{T}_K$ = مجموعه‌ای از گروه‌های ترمینال در (T_K)

$\left|\tilde{T}_K\right|$ = تعداد گره‌های ترمینال به $\tilde{T}_K$

α = پارامتر هزینه - پیچیدگی

هستند. هرس حداقل هزینه پیچیدگی، یکی از انواع هرس درختان تصمیم است.

به‌طورکلی می‌توان گفت، هرس نمودن درخت طبقه‌بندی، باعث جلوگیری از خطای بیش برازش شده و عملکرد و اعتبار سیستم را افزایش می‌دهد و موجب کاهش هزینه‌های محاسباتی می‌شود. همچنین هرس درخت طبقه‌بندی، برای نمونه‌های دیده نشده، دقت بالاتری را رقم می‌زند.

۴-۳-۳ معیارهای ارزیابی عملکرد

عملکرد مدل سیستم IDS توسعه‌یافته با اصلاح DBSCAN، با استفاده از ماتریس ابهام، اندازه‌گیری شد. پارامترهای عملکرد مورداستفاده شامل دقت[1]، صحت[2]، حساسیت[3] و خاصیت[4] و امتیاز F [5] بود. پارامترهای عملکرد را می‌توان با استفاده از معادلات (۱۳) تا (17) محاسبه کرد.

ماتریس ابهام

	منفی	مثبت
منفی	TN	FP
مثبت	FN	TP

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}, \quad (۱۳)$$

$$Sensitivity = TP_{rate} = \frac{TP}{TP + FN}, \quad (۱۴)$$

$$Specificity = TN_{rate} = \frac{TN}{TN + FP}, \quad (۱۵)$$

$$Precision = \frac{TP}{TP + Fp}, \quad (۱۶)$$

$$F\_Score = \frac{2 * TP}{2 * TP + Fp + FN}. \quad (۱۷)$$

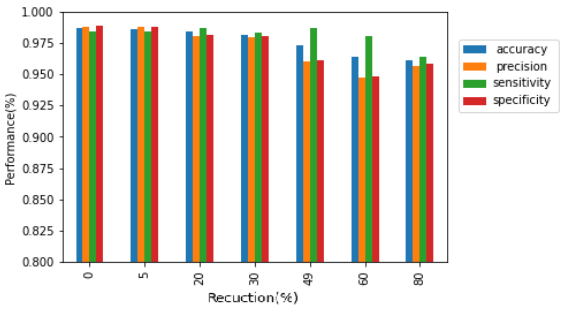
۵. نتایج عملی

نتایج حاصل از ترکیب کاهش RN-DBSCAN و مدل CART بر روی دو مجموعه‌داده NSL-KDD و Kaggle با استفاده از نمودارهای عملکرد و زمان برای درصد کاهش‌های ۵٪، ۲۰٪، ۳۰٪، ۴۹٪، ۶۰٪ و در نهایت ۸۰٪، مورد بررسی و تحلیل قرار گرفته است. برای بررسی این نتایج به دلیل نیاز به محیطی یکسان جهت بررسی زمان حاصل شده، از محیط colab، بهره گرفته شده است.

در این مقاله، دو مجموعه‌داده به‌صورت مجزا مورد بررسی جهت بررسی تعیین اثر فرایند Kaggleقرار گرفتند. داده‌های کاهش با دو سناریو، بدون درنظرگرفتن برچسب داده و با درنظرگرفتن برچسب داده و گروه‌بندی داده‌ها بر اساس برچسب، برای انجام NSL-KDD بررسی شدند. همچنین از مجموعه‌دادهٔ فرایند کاهش از رویکرد با درنظرگرفتن برچسب داده و گروه‌بندی داده‌ها بر اساس برچسب، استفاده نموده است. زمانی که برچسب داده در نظر گرفته نشود، عمل کاهش بر روی تمامی داده‌ها بدون توجه به برچسب آن‌ها، انجام می‌شود و دراین‌بین، امکان دارد تنها داده‌هایی از یک کلاس حذف گردند و یا به‌صورت غیرمساوی از هر کلاس کاهش اعمال شود که این امر موجب عدم تعادل در مجموعه‌داده خواهد شد. اما با درنظرگرفتن برچسب داده‌ها طی فرایند کاهش، داده‌ها بر اساس برچسب به دو گروه (باتوجه‌به اینکه دو کلاس داده وجود دارد) تقسیم شده و برای هر گروه، درصد کاهش به‌صورت جدا انجام می‌شود. جهت بررسی این‌که آیا تغییرات در تعادل داده‌ها بر عملکرد نتایج طبقه‌بندی تأثیر به‌صورت Kaggleمی‌گذارد یا خیر، این مورد بر روی داده‌های مجزا انجام‌گرفته است که در ادامه به بررسی نتایج پرداخته می‌شود.

۵-۱ نتایج بر روی مجموعه Kaggle بدون تفکیک کلاس

شکل (۲) نشان می‌دهد که درصدهای مختلف کاهش، بر عملکرد سیستم تشخیص نفوذ اثر می‌گذارد. با توجه به شکل، دقت به‌طور یکنواخت با افزایش درصد کاهش، کم شده است. معیارهای خاصیت و صحت با افزایش درصد کاهش داده‌ها، سیر تغییر مشخص و یکنواخت نداشته‌اند. همچنین معیار حساسیت نسبت به بقیه معیارها از عملکرد بهتری برخوردار بوده است. با توجه به تمامی پارامترها، هر چه درصد کاهش داده‌ها افزایش می‌یابد، عملکرد سیستم تشخیص نفوذ، کاهش می‌یابد. اما این کاهش عملکرد نسبت به درصد کاهشی که اعمال می‌شود، کم است و همان‌طور که مشخص است، برای ۸۰٪ کاهش، دقت ۹۶/۱۲٪ حفظ شده است.



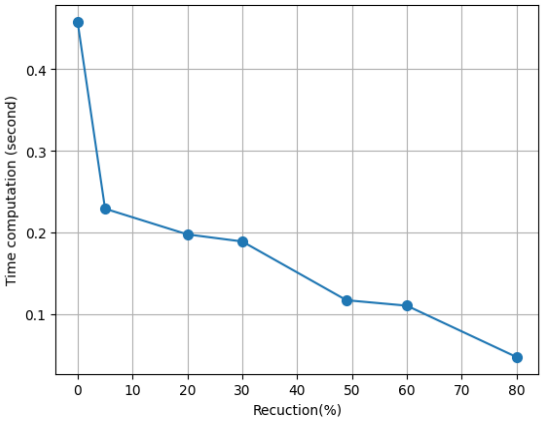
شکل (۲): تأثیر کاهش داده‌ها بر عملکرد سیستم تشخیص نفوذ، بدون درنظر گرفتن برچسب‌ها، در مجموعه داده Kaggle.

نمودار بعدی که مورد ارزیابی قرار می‌گیرد، بررسی و مقایسهٔ زمان با عملکرد سیستم تشخیص نفوذ است.

محاسبه زمان برای اندازه‌گیری اثر کاهش درصد داده‌ها بر

[1] Accuracy
[2] Precision
[3] Sensitivity
[4] Specificity
[5] F_Score

زمان انجام می‌شود. نتایج ارزیابی پارامترهای عملکرد محاسبات زمانی، برای طبقه‌بندی در سیستم تشخیص نفوذ، در شکل (۳) نشان‌داده‌شده است. همان‌طور که از شکل مشخص است؛ هر چه درصد کاهش داده‌ها بیشتر باشد، زمان محاسبه کم‌تر است. از منظر عملکرد، کاهش نسبتاً کم است، اما کاهش زمان محاسبات بسیار قابل‌توجه است. برای کاهش ۸۰ درصدی داده، زمان محاسبه از ۴۵۸/۰۹ ms به ۴۷/۲۱ ms کاهش یافت.



شکل (۳): زمان محاسبات سیستم تشخیص نفوذ، بدون درنظر گرفتن برچسب‌ها، در مجموعه داده Kaggle

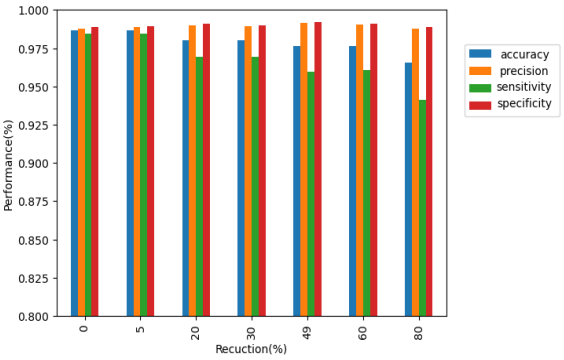
مقدار دقیق نتایج به‌دست‌آمده در جدول (۴) قرار داده شده است.

جدول (۴): مقادیر دقیق عملکرد سیستم تشخیص نفوذ، مجموعه داده Kaggle بدون درنظر گرفتن برچسب‌ها

کاهش داده (٪)	دقت	خاصیت	حساسیت	صحت	زمان (ms)
۰	۹۸/۶۶	۹۸/۸۶	۹۸/۴۴	۹۸/۸۰	۴۵۸/۰۹
۵	۹۸/۶۳	۹۸/۸۳	۹۸/۴۲	۹۸/۷۵	۲۲۹/۰۹
۲۰	۹۸/۳۷	۹۸/۰۹	۹۸/۶۶	۹۷/۹۹	۱۹۷/۶۱
۳۰	۹۸/۱۸	۹۸/۰۴	۹۸/۳۳	۹۷/۹۴	۱۸۸/۹۱
۴۹	۹۷/۳۴	۹۶/۰۹	۹۸/۶۶	۹۵/۹۸	۱۱۶/۹۲
۶۰	۹۶/۳۸	۹۴/۸۰	۹۸/۰۶	۹۴/۷۰	۱۱۰/۲۳
۸۰	۹۶/۱۲	۹۵/۸۸	۹۶/۳۷	۹۵/۶۸	۴۷/۲۱

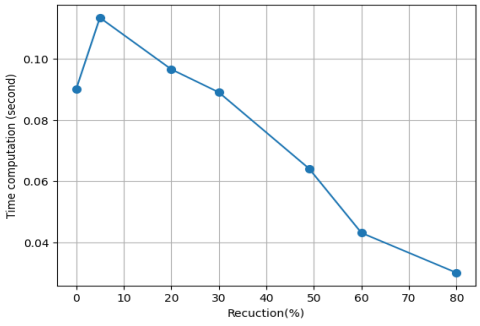
۵-۲ نتایج بر روی مجموعه Kaggle با تفکیک کلاس

در شکل (۴)، مشخص است که افزایش درصد کاهش داده‌ها برعملکرد سیستم تشخیص نفوذ تأثیر می‌گذارد. کاهش در عملکرد سیستم تشخیص نفوذ، به‌ویژه برای پارامتر خاصیت نسبتاً کم بود. پارامتر حساسیت، بیشترین کاهش را دارد، اما به‌طور کلی، دقت عملکرد سیستم تشخیص نفوذ با نرخ کاهش داده تا ۸۰٪ هنوز نسبتاً خوب است. پارامتر صحت، که توانایی سیستم تشخیص نفوذ در انتقال حمله را می‌سنجد [۲۶]، درصد بالایی داشت. به‌طور کلی با افزایش میزان درصد کاهش داده‌ها، الگوی عملکرد به صورت خطی کاهش یافته است.



شکل (۴): تأثیر کاهش داده‌ها بر عملکرد سیستم تشخیص نفوذ با تفکیک کلاس، در مجموعه داده Kaggle

آزمون بعدی شامل محاسبه زمان، با درنظرگرفتن برچسب‌های داده‌ها می‌باشد. نتایج آزمون در شکل (۵) ارائه شده است.



شکل (۵): زمان محاسبات سیستم تشخیص نفوذ، با درنظر گرفتن برچسب‌ها، در مجموعه داده Kaggle.

استفاده از کاهش داده‌ها با RN_DBSCAN می‌تواند زمان محاسبات را به طور قابل‌توجهی کاهش دهد و درعین‌حال عملکرد سیستم تشخیص نفوذ را حفظ کند. باتوجه‌به شکل (۵) مشاهده می‌شود که با ۸۰٪ کاهش داده، زمان از ۹۰ ms به ۳۰ ms کاهش می‌یابد، درصورتی‌که دقت از ۹۸/۶۶٪ به ۹۶/۵۶٪ کاهش داشته است که نسبت به کاهش زمان، کم بوده و همچنان بالای ۹۵ درصد حفظ شده است. مقدار دقیق نتایج به‌دست‌آمده در جدول (۵) قرار داده شده است.

جدول (۵): مقادیر دقیق عملکرد سیستم تشخیص نفوذ، مجموعه داده Kaggle با تفکیک کلاس

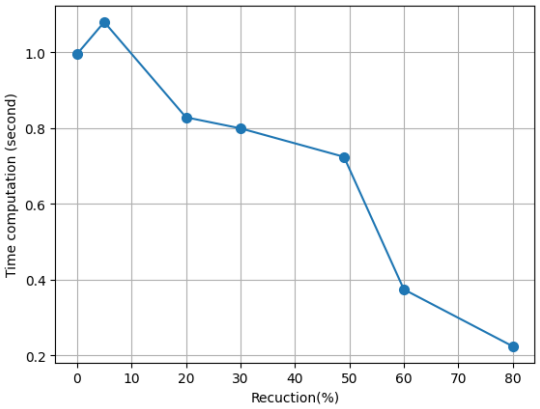
کاهش داده (٪)	دقت	خاصیت	حساسیت	صحت	زمان (ms)
۰	۹۸/۶۶	۹۸/۸۶	۹۸/۴۴	۹۸/۷۹	۹۰
۵	۹۸/۶۹	۹۸/۹۴	۹۸/۴۳	۹۸/۸۷	۱۱۳/۴۳
۲۰	۹۸/۰۴	۹۹/۰۸	۹۶/۹۵	۹۹	۹۶/۵۶
۳۰	۹۸	۹۹/۰۲	۹۶/۹۴	۹۸/۹۴	۸۹/۰۳
۴۹	۹۷/۶۴	۹۹/۲۳	۹۵/۹۶	۹۹/۱۶	۶۴/۰۶
۶۰	۹۷/۶۳	۹۹/۱۰	۹۶/۰۸	۹۹/۲	۴۳/۰۶
۸۰	۹۶/۵۶	۹۸/۹۰	۹۴/۱۱	۹۸/۷۷	۳۰

کاهش داده‌ها، با درنظرگرفتن تفکیک کلاس یا بدون درنظرگرفتن آن، تأثیری بر عملکرد سیستم ندارد، اما برای پیش‌بینی وقوع داده‌های نامتعادل تولید شده در فرایند کاهش، تفکیک کلاس بهتر است انجام شود. همچنین کاهش با درنظرگرفتن برچسب داده‌ها، مشکل عدم چگالی یکسان در سطح داده [29] که یکی از موانع تنظیم صحیح پارامترهای الگوریتم DBSCAN می‌باشد را تا حدی حل می‌نماید.

۵-۳ نتایج بر روی مجموعه NSL_KDD با تفکیک کلاس

باتوجه‌به نمودار عملکرد سیستم تشخیص نفوذ در شکل (۶)، پارامتر دقت که مهم‌ترین معیار ارزیابی عملکرد سیستم می‌باشد، برای درصد کاهش‌های مختلف تا ۸۰٪ کاهش، بالای ۹۰٪ حفظ شده است. همچنین معیار حساسیت، از صفر تا چهل و نه درصد کاهش، بالای ۹۸٪ بوده و عملکرد بسیار مناسبی دارد. معیار صحت نسبت به پارامترهای دیگر ارزیابی، از عملکرد ضعیف‌تری برخوردار بوده است، اما به‌طورکلی می‌توان گفت، در این مجموعه‌داده نیز با کاهش ۸۰ درصدی داده‌ها، عملکرد سیستم به‌خوبی حفظ شده است.

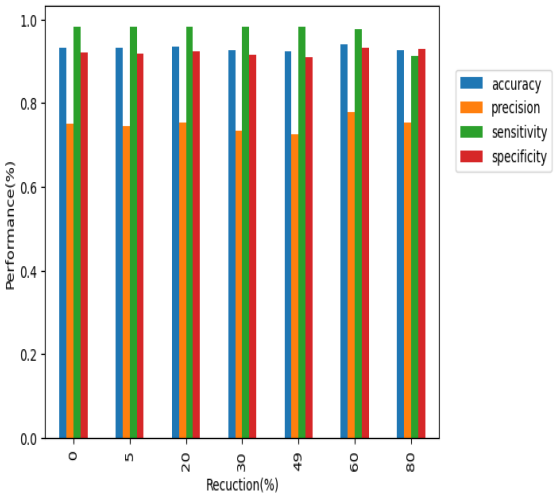
معیار مورد ارزیابی بعدی برای این مجموعه‌داده، بررسی زمان، برای درصدهای مختلف کاهش می‌باشد. محاسبهٔ زمان برای این مجموعه‌داده در شکل (۷) نشان‌داده‌شده است.



شکل (۷): زمان محاسبات سیستم تشخیص نفوذ، با درنظر گرفتن برچسب‌ها، در مجموعه داده NSL_KDD

همان‌طور که در شکل (۷) مشخص است؛ هر چه درصد کاهش داده‌ها بیشتر شود، پارامترهای عملکرد کاهش یافته و زمان محاسبات نیز کاهش می‌یابد. با توجه به جدول (۶) که مقادیر دقیق به‌دست آمده از ترکیب کاهش RN_DBSCAN و اعمال مدل CART برای مجموعه داده NSL_KDD می‌باشد، مشاهده می‌شود که زمان محاسبات در ۰٪ کاهش، ۹۹۵/۲ ms بوده است که این مقدار برای ۸۰٪ کاهش به ۲۲۳/۶۰ ms، رسیده است. همان‌طور که مشخص است کاهش در زمان محاسبات بسیار قابل توجه می‌باشد، درصورتی که کاهش در نتیجه عملکرد نسبتاً کم است و دقت بالای ۹۲/۳۶٪ حفظ شده است

شکل (۶): تأثیر کاهش داده‌ها بر عملکرد سیستم تشخیص نفوذ، در مجموعه داده مجموعه دادهNSL-KDD



جدول (۶): مقادیر دقیق عملکرد سیستم تشخیص نفوذ، مجموعه داده NSL_KDD با تفکیک کلاس

زمان (ms)	صحت	حساسیت	خاصیت	دقت	کاهش داده (٪)
۹۹۵/۲	۷۹/۹۵	۹۸/۲۲	۹۲/۰۶	۹۳/۲۶	۰
۱۰۸۰/۹	۷۴/۵۲	۹۸/۲۱	۹۱/۸۸	۹۳/۱۱	۵
۸۲۸/۳۱	۷۵/۳۲	۹۸/۱۵	۹۲/۲۲	۹۳/۳۷	۲۰
۷۹۹/۱۸	۷۳/۴۷	۹۸/۱۶	۹۱/۴۲	۹۲/۷۴	۳۰
۷۲۳/۹۶	۷۲/۴۰	۹۸/۱۹	۹۰/۹۵	۹۲/۳۶	۴۹
۳۷۳/۹۳	۷۷/۷۴	۹۷/۶۴	۹۳/۲۴	۹۴/۱	۶۰
۲۲۳/۶۰	۷۵/۴۲	۹۱/۲۱	۹۲/۸۰	۹۲/۵۱	۸۰

۴-۵-کارهای مشابه در سال‌های اخیر

در سال‌های اخیر بسیاری از تحقیقات و مطالعات در حوزه سیستم‌های تشخیص نفوذ و کاهش داده‌ها انجام شده است. هدف این تحقیقات بررسی روش‌های مختلف برای کاهش داده‌ها در سیستم تشخیص نفوذ بوده است.

بر اساس نتایج جدول (۷)، روش پیشنهادی مورد بررسی در این مقاله نشان داده است که در مقایسه با روش‌های قبلی، عملکرد بهتری در کاهش داده‌ها در سیستم تشخیص نفوذ داشته است.

باتوجه‌به جدول ارائه شده، این روش علاوه بر افزایش سرعت، توانسته است دقت و عملکرد را بهبود بخشد و نتایج بهتری را به دست آورد. با استناد به این مقاله و تحقیقات گذشته، می‌توان این نتیجه را گرفت که روش پیشنهادی در این مقاله، به‌عنوان یک جایگزین قوی برای روش‌های قبلی در حوزه کاهش داده‌ها در سیستم تشخیص نفوذ می‌تواند مورداستفاده قرار گیرد.

جدول (۷): مقایسه با تحقیقات قبلی

مطالعه	روش	رویکرد کاهش داده‌ها	مجموعه‌داده	امتیاز اف	دقت
باتاچار و همکاران [30]	SVM	PCA	Kaggle	-	95/2
	NB	PCA	Kaggle	-	75/3
	NB	PCA+Firefly	Kaggle	-	84/2
سار کر و همکاران [31]	NB	-	Kaggle	90/0	90/0
ویهارتو و همکاران [19]	CART	m_DBSCAN	Kaggle	89/43	92/3
شهباز و همکاران [32]	J48	CFS	NSL_KDD	-	86/1
عبدالله و همکاران[33]	Voting classifier	IG-Filters	NSL_KDD	82/3	86/67
صبری و همکاران[34]	DT	PIO	NSL_KDD	88/2	88/3
کاسونگو [35]	RNN+ LSTM+ GRU	XGBoost	NSL_KDD	99/58	88/13
پیشنهاد شده	CART	RN_DBSCAN	Kaggle	96/39	**96/56**
			NSL_KDD	82/60	**92/51**

۶. نتیجه‌گیری

پژوهش‌های گذشته نشان داده است که استفاده از روش‌های خوشه‌بندی می‌تواند به کاهش داده‌ها کمک کند و به‌عنوان پایه‌ای برای کاهش حجم داده‌ها عمل کند. در این پژوهش به‌منظور بهبود سیستم تشخیص نفوذ و در نتیجهٔ آن بهبود امنیت شبکه‌های کامپیوتری، کاهش حجم داده‌ها با استفاده از روش RN_DBSCAN مطرح گردید. این روش بر اساس رویکرد چگالی کار می‌کند و با یافتن نقاط اطراف نقاط مرکزی که بسیار مشابه و نزدیک نقاط مرکزی هستند، این نقاط را کاهش می‌دهد. با این روش، داده‌های مهم برای آموزش مدل حفظ می‌شوند و داده‌های یکنواخت و همگن حذف می‌شوند.

برای تنظیم پارامترهای الگوریتم RN_DBSCAN، از الگوریتم ژنتیک بهره برده شده است. همچنین برای طبقه‌بندی داده‌های کاهش‌یافته، از الگوریتم طبقه‌بند CART استفاده شده است. نتایج حاصل از پژوهش نشان می‌دهد که سیستم تشخیص نفوذ با استفاده از الگوریتم RN_DBSCAN قادر است حجم داده‌ها را تا ۸۰٪ کاهش دهد، درحالی‌که عملکرد سیستم بادقت بالای ۹۰٪ حفظ می‌شود. علاوه بر این، استفاده از این الگوریتم

منجر به کاهش زمان محاسباتی و کاهش پیچیدگی و هزینه می‌شود. همچنین، مقایسه الگوریتم‌های مختلف نشان داده است که RN_DBSCAN نسبت به الگوریتم‌های کاهش سطری دیگر مانند Sampling و Slice و حتی الگوریتم m_DBSCAN کارایی بهتری دارد. این روش منجر به بهبود عملکرد و سرعت سیستم تشخیص نفوذ می‌شود و درعین‌حال حجم داده‌ها را به طور آگاهانه کاهش می‌دهد تا داده‌های مهم برای آموزش مدل حفظ شوند. این پژوهش می‌تواند به توسعه و پیشرفت در حوزه امنیت شبکه‌های کامپیوتری کمک کند.

۷. مراجع

[1] E. A. Shammar and A. T. Zahary, "The Internet of Things (IoT): a survey of techniques, operating systems, and trends," Library Hi Tech, vol. 38, no. 1, pp. 5-66, 2020.

[2] H. Tanha and M. Abbasi, "Traffic identification in Internet of Things infrastructure using neural networks and deep learning," Scientific Journal of Electronic and Cyber Defense, vol. 11, no. 2, pp. 1-13, Jul. 2023. (in Persian).DOR:20.1001.1.23224347.1402.11.2.1.4

[3] Y. Zhang, Y. Zhang, T. Chen, and B. Xia, "Internet of Things (IoT) security: A survey," Journal of Information Security and Applications, vol. 50, p. 102419, 2020.

[4] J. Mazloom and H. Bigdeli, "Optimized combination deep neural network integrated with feature selection for intrusion detection system in cyber-attacks," Journal of Information Security and Applications, vol. 10, no. 4, pp. 41-51, Feb. 2022. . (in Persian). DOR:20.1001.1.23224347.1401.10.4.5.5

[5] W. Zhang and S. Li, "A Deep Learning Approach for Intrusion Detection System," IEEE Access, vol. 9, pp. 35470-35479, 2021.

[6] M. H. Nataj Salhaddar, "Investigating a New Hybrid Approach for Intrusion Detection System on Various Datasets," Journal of Information Security and Applications, vol. 10, no. 3, pp. 43-57, Dec. 2022. (in Persian).DOR:20.1001.1.23224347.1401.10.3.5.3

[7] I. Ahmed, M. Mahfuzul Islam, and A. A. Adewole, "A survey of intrusion detection techniques in cloud computing," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 42-57, 2013.

[8] M. S. Farash and S. Samet, "Feature Selection for Intrusion Detection Systems: A Comprehensive Review," Computer Networks, vol. 74, pp. 443-460, 2014.

[9] A. A. Wiharto and U. Permana, "Improvement of performance intrusion detection system (IDS) using artificial neural network ensemble," Journal of Theoretical and Applied Information Technology, vol. 80, no. 2, pp. 191-201, 2015.

[10] D. Uhm, S. H. Jun, and S. J. Lee, "A classification method using data reduction," International Journal of Fuzzy Logic and Intelligent Systems, vol. 12, no. 1, pp. 1-5, 2012.

[11] N. A. Le-Khac, M. Bue, M. Whelan, and M. T. Kechadi, "A clustering-based data reduction for very large spatiotemporal datasets," in Advanced Data Mining and Applications, 2010, pp. 43-54.

[12] J. Wang, S. Yue, X. Yu, and Y. Wang, "An efficient data reduction method and its application to cluster analysis," Neurocomputing, vol. 238, pp. 234-244, 2017.

[13] A. C. Benabdellah, A. Benghabrit, and I. Bouhaddou, "A survey of clustering algorithms for an industrial context," Procedia Computer Science, vol. 148, pp. 291-302, 2019.

[14] J. M. Dudik, A. Kurosu, J. L. Coyle, and E. Sejdic, "A comparative analysis of DBSCAN, K-means, and quadratic variation algorithms for automatic identification of swallows from swallowing accelerometry signals," Computers in Biology and Medicine, vol. 59, pp. 10-18, 2015.

[15] S. Ougiaroglou and G. Evangelidis, "Efficient dataset size reduction by finding homogeneous clusters," in Proceedings of the 5th Balkan Conference in Informatics, 2012, pp. 168-173.

[16] S. Ougiaroglou and G. Evangelidis, "RHC: a nonparametric cluster-based data reduction for efficient k-NN classification," Pattern Analysis and Applications, vol. 19, no. 1, pp. 93-109, 2016.

[17] S. Ougiaroglou, K. I. Diamantaras, and G. Evangelidis, "Exploring the effect of data reduction on neural network and support vector machine classification," Neurocomputing, vol. 208, pp. 101-110, 2018.

[18] S. Ougiaroglou and G. Evangelidis, "Efficient editing and data abstraction by finding homogeneous clusters," Annals of Mathematics and Artificial Intelligence, vol. 76, no. 3-4, pp. 327-349, 2016.

[19] A. K. Wicaksana and D. E. Cahyani, "Modification of a density-based spatial clustering algorithm for applications with noise for data reduction in intrusion detection systems," International Journal of Fuzzy Logic and Intelligent Systems, vol. 21, no. 2, pp. 189-203, 2021.

[20] F. O. Ozkok and M. Celik, "A new approach to determine Eps parameter of DBSCAN algorithm," International Journal of Intelligent Systems and Applications Engineering, vol. 5, no. 4, pp. 247-251, 2017.

[21] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 1996, pp. 226-231.

[22] M. Zbigniew, "Genetic algorithms + data structures = evolution programs," Comput Stat, 1996.

[23] L. Rutkowskia, M. Jaworskia, L. Pietruczuka, and P. Duda, "The CART Decision Tree for Mining Data Streams," Information Sciences, vol. 266, pp. 1-15, 2014.

[24] R. A. Johnson and D. W. Wichern, Applied Multivariate Statistical Analysis. Upper Saddle River, NJ: Pearson Prentice Hall, 2007.

[25] P. I. Radoglou-Grammatikis and P. G. Sarigiannidis, "An anomaly-based intrusion detection system for the smart grid based on CART decision tree," in Proceedings of the Global Information Infrastructure and Networking Symposium (GIIS), 2018.

[26] H. H. Patel and P. Prajapati, "Study and analysis of decision tree based classification algorithms," International Journal of Computer Sciences and Engineering, vol. 6, no. 10, pp. 74-78, 2018.

[27] A. Priyam, G. R. Abhijeeta, A. Rathee, and S. Srivastava, "Comparative analysis of decision tree classification algorithms," International Journal of Current Engineering and Technology, vol. 3, no. 2, pp. 334-337, 2013.

[28] H. M. Sani, C. Lei, and D. Neagu, "Computational complexity analysis of decision tree algorithms," in Proceedings of the Artificial Intelligence XXXV, 2018, pp. 191-197.

[29] A. Zadedehbalaei, A. Bagheri, and H. Afshar, "A study on DBSCAN Clustering algorithm issues and a survey on its improvements," Soft Computing Journal, vol. 6, 2021, pp. 2322-3707.

[30] S. Bhattacharya, P.K.R. Maddikunta, R. Kaluri, S. Singh, T.R. Gadekallu, M. Alazab, U. Tariq, "A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU," Electronics, vol. 9, no. 2, 2020.

[31] I.H. Sarker, Y.B. Abushark, F. Alsolami, A.I. Khan, "IntruDTree: a machine learning based cyber security intrusion detection model," Symmetry, vol. 12, no. 5, 2020.

[32] M.B. Shahbaz, X. Wang, A. Behnad, J. Samarabandu, "On efficiency enhancement of the correlation-based feature selection

for intrusion detection systems," in Proceedings of the 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2016, pp. 1–7.
[33] M. Abdullah, A. Balamash, A. Alshannaq, S. Almabdy, "Enhanced Intrusion Detection System using Feature Selection Method and Ensemble Learning Algorithms," International Journal of Computer Science and Information Security (IJCSIS), vol. 16, no. 2, 2018, pp. 48–55.
[34] H. Alazzam, A. Sharieh, K.E. Sabri, "A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer," Expert Systems with Applications, vol. 148, 2020, 113249.
[35] S. M. Kasongo, "A deep learning technique for intrusion detection system using a recurrent neural networks based framework," Computer Communications, vol. 199, no. 1, pp. 113–125, 2023