Passive Defence
Vol. 15, No. 4, Winter 2024, Serial No. 60
ISSN: 2980-8030 , E-ISSN: 2008-6849

Feature Selection and Intrusion Detection in Wireless Sensor Networks with Unsupervised Extreme Learning Machine (UELM)

Hamid Tabatabaee* , Samira Hadavi

*Associate Professor, Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

(Received: 08/01/2024, Revised: 03/04/2024, Accepted: 22/09/2024, Published: 15/12/2024)

DOR: 20.1001.1.20086849.1403.15.4.3.6

ABSTRACT

Nowadays, network-based computer systems play a vital role in today's modern society, and for this reason, they may be the target of hostility or infiltration. In order to ensure complete security in a computer system connected to the network, using a firewall and other intrusion prevention mechanisms is not always enough. This need has led to the use of other systems called intrusion detection systems. An intrusion detection system can be considered a set of tools, methods, and documents that help identify, determine, and report unauthorized or unapproved activities on the network. Intrusion detection systems are created in the form of software and hardware systems, each with its own advantages and disadvantages. Due to the presence of many features in the data related to intrusion detection systems, this thesis focuses on selecting the desired and effective features using Unsupervised Extreme Learning Machine. A model for data classification is then presented using UELM. To evaluate the performance of the proposed method, the NSL-KDD database is used because it contains more realistic records than other intrusion detection datasets. The test results show that UELM achieves an accuracy of 98.38%, compared to GWO's accuracy of 93.74%. The superiority of UELM in classification and intrusion detection problems is attributed to its robust and generalizable structure as an unsupervised neural network.

Keywords: Feature Selection, Artificial Neural Networks, Unsupervised Extreme Learning Machine, Intrusion Detection

* Corresponding Author Email: h_tabatabaee@mshdiau.ac.ir

پدافند غیرعامل

سال پانزدهم، شماره ۴، زمستان ۱۴۰۳، (پیاپی ۶۰): صص ۴۰-۲۵

شاپای چاپی: ۶۹۴۹-۲۰۰۸ | **شاپای الکترونیکی:** ۸۰۳۰-۲۹۸۰

علمی – پژوهشی

انتخاب ویژگی و تشخیص نفوذ در شبکه‌های حسگر بی‌سیم با استفاده از یادگیری ماشین مفرط بدون نظارت (UELM)

حمید طباطبایی[1*]، سمیرا هادوی[2]

DOR: 20.1001.1.20086849.1403.15.4.3.6

تاریخ دریافت: ۱۴۰۲/۱۰/۱۸ تاریخ پذیرش: ۱۴۰۳/۰۷/۰۱

تاریخ بازنگری: ۱۴۰۳/۰۱/۱۵ تاریخ انتشار: ۱۴۰۳/۰۹/۲۵

چکیده

امروزه سیستم‌های کامپیوتری مبتنی بر شبکه، نقش حیاتی در جامعه مدرن امروزی دارند و به همین علت ممکن است هدف دشمن و یا نفوذ قرار گیرند. به‌منظور ایجاد امنیت کامل در یک سیستم کامپیوتری متصل به شبکه، استفاده از دیوار آتش و سایر مکانیزم‌های جلوگیری از نفوذ همیشه کافی نیست و باید از سیستم‌های دیگری به نام سیستم‌های تشخیص نفوذ استفاده شود. به‌دلیل وجود مشخصه‌های زیاد در داده‌های مربوط به سیستم‌های تشخیص نفوذ، جهت استفاده از مشخصه‌های مطلوب و موثر از الگوریتم یادگیری ماشین مفرط بدون نظارت استفاده می‌شود. جهت طبقه‌بندی داده‌ها از مدل UELM و ارزیابی عملکرد روش پیشنهادی، از پایگاه داده با رکوردهای واقعی تر NSL-KDD نسبت به سایر مجموعه دادگان تشخیص نفوذ، استفاده می‌گردد. نتایج آزمایش‌ها نشان‌دهنده صحت ۹۸/۳۸ UELM در مقایسه با صحت ۹۳/۷۴ GWO است. دلیل این برتری، استفاده ازمدل مناسب در مسئله دسته‌بندی، تشخیص نفوذ، ساختار مستحکم و تعمیم‌پذیر شبکه عصبی بدون نظارت می باشد.

کلیدواژه‌ها: انتخاب ویژگی، شبکه‌های عصبی مصنوعی، ماشین یادگیری مفرط بدون نظارت، تشخیص نفوذ

[1] دانشیار گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران (h_tabatabaee@mshdiau.ac.ir)- نویسنده مسئول

[2] کارشناسی ارشد گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران

ناشر: دانشگاه جامع امام حسین (ع)

۱- مقدمه

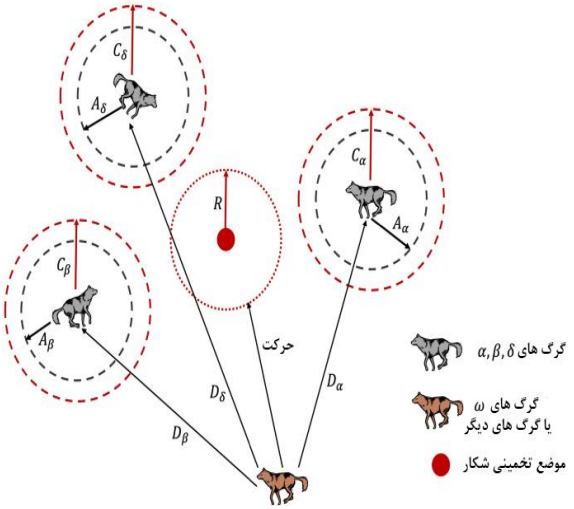
با پیشرفت بیش از پیش فناوری و افزایش روز افزون بهره‌گیری از شبکه‌های کامپیوتری، خطرات حمله به این شبکه‌ها را افزایش داده است[۱]. نفوذ به مجموعه اقدامات غیرقانونی که محرمانگی و یا دسترسی به یک منبع را به خطر می‌اندازد اطلاق می‌گردد. سیستم تشخیص نفوذ را می‌توان مجموعه‌ای از ابزارها، روش‌ها و مدارکی در نظر گرفت که به شناسایی، تعیین و گزارش فعالیت‌های غیرمجاز یا تائید نشده تحت شبکه، کمک می‌کند.افزایش دسترسی به داده‌ها و پردازش سریع‌تر آن‌ها و در عین حال افزایش حجم داده‌ها و نیاز به فراهم آوردن داده‌ها از منابع مختلف از طریق این شبکه‌ها، منجر به پدید آمدن منابع تهدیدآمیزی می‌گردد که از طریق نقاط ضعف موجود در سیستم‌ها، به استثمار سیستم‌ها و ایجاد اختلال در آن‌ها می‌پردازد. تشخیص و جلوگیری از نفوذ (IDS) امروزه به عنوان یکی از مکانیزم‌های اصلی در برآوردن امنیت شبکه‌ها و سیستم‌های کامپیوتری مطرح است[۲]. بیشتر سیستم‌های تشخیص نفوذ کنونی از تمامی پارامترهای موجود در بسته‌های شبکه برای ارزیابی و کشف الگوهای حملات استفاده می‌نمایند، در صورتی که برخی از این پارامترها غیرمرتبط و زائد می‌باشند. استفاده از تمامی پارامترها باعث می‌شود که فرآیند تشخیص طولانی و کارایی سیستم تشخیص نفوذ تنزل یابد. در واقع چالش اساسی در سیستم تشخیص نفوذ حجم عظیم داده‌هاست. از طرفی با توجه به ترافیک بالا کاهش نرخ هشدار غلط در سیستم تشخیص نفوذ نیز از اهمیت خاصی برخوردار است. تمام سیستم‌های تشخیص نفوذ قادر به تولید هشدار در مورد وقوع نفوذ در شبکه می‌باشند. ولی به علت حجم بالای هشدارهای تولید شده توسط این سیستم‌ها و همچنین تولید هشدارهای اشتباه، این سیستم‌ها قادر به مدیریت و آنالیز هشدارهای تولید شده نمی‌باشند.

امروزه بیشتر رویکردها در تشخیص نفوذ مربوط به مساله انتخاب و یا استخراج ویژگی‌های مهم متمرکز شده است. تعداد ویژگی‌هایی که بهترین دقت را در تشخیص نفوذ دارند، در مجموعه داده‌های مختلف بیشتر به صورت تجربی بدست می‌آید. اما انتخاب ویژگی‌ها ممکن است باعث از دست دادن قسمتی از داده‌ها شود.

با توجه به اینکه پردازش داده‌ها با اعداد کار می‌کنند و برخی از مشخصه‌های این مجموعه‌داده دارای مقادیر رشته‌ای هستند، لذا تغییر آن‌ها به مقادیر عددی برای انجام پردازش الزامی می‌باشد. برای تبدیل نوع رشته‌ای به عدد نیز به این صورت عمل می‌شود که به ازای هر نوع یک عدد منحصر به فرد در نظر گرفته می‌شود. بنابراین با توجه به تفاوت داده‌های مربوط به شبکه، تخمین تعداد بهینه ویژگی‌ها یکی از چالش‌های مهم می‌باشد[۳]. در این مقاله از ویژگی‌هایی به عنوان صفات اساسی استفاده می شود که دقت تشخیص نفوذ را افزایش دهد و همچنین تاثیر ویژگی‌هایی که انتخاب نشده‌اند را نیز در سیستم تشخیص نفوذ در نظر گرفته می شود. با استفاده از ماشین یادگیری مفرط بدون نظارت مدلی برای تشخیص نفوذ در شبکه‌های حسگر بی‌سیم ارائه می‌گردد. همچنین برای کاهش ضعف‌های سیستم‌های تشخیص نفوذ اعم از (افتادن در دام بهینه محلی و همگرایی زودرس الگوریتم‌های فراابتکاری) یک رویکرد جدید برای انتخاب ویژگی و تشخیص هر نوع حمله، پیشنهاد می‌گردد که توسط یادگیری ماشین مفرط بدون نظارت انجام می‌شود. از یادگیری ماشین مفرط بدون نظارت با ضریب تنظیم (رگولاریزیشن) در تشخیص نفوذ با اعمال پارامترهای بهینه به منظور کاهش اثرات منفی پارامترهای غیر بهینه استفاده شده است.

۱-۱- الگوریتم بهینه‌سازی گرگ خاکستری

الگوریتم بهینه‌سازی گرگ خاکستری[1]، الگوریتم فرا ابتکاری الهام گرفته شده از طبیعت است.این الگوریتم از تعدادی مشاهدات درباره زندگی گرگ خاکستری در طبیعت الهام گرفته شده است. در اصل، ایده اصلی الگوریتم رفتار اجتماعی و مکانیزم شکار این گرگ را شبیه‌سازی می‌نماید. در GWO، افراد به چهارگونه به نام‌های آلفا (α)، بتا (β)، دلتا (δ) و امگا (ω) تقسیم می‌شوند. α راه‌حل با بالاترین برازندگی را نشان می‌دهد یا به عبارتی بهترین راه‌حل یافت شده تاکنون. بطور مشابه، β و δ به ترتیب دومین و سومین بهترین فرد را نشان می‌دهند. ω به باقیمانده افراد موجود در جمعیت اشاره دارد[۴].



شکل (۱): مثالی از مکانیزم بهنگام‌سازی موقعیت گرگ‌های ω با توجه به گرگ‌های α، β و δ [۵]

[1]Gray Wolf Optimization Algorithm

۱-۲- انتخاب ویژگی

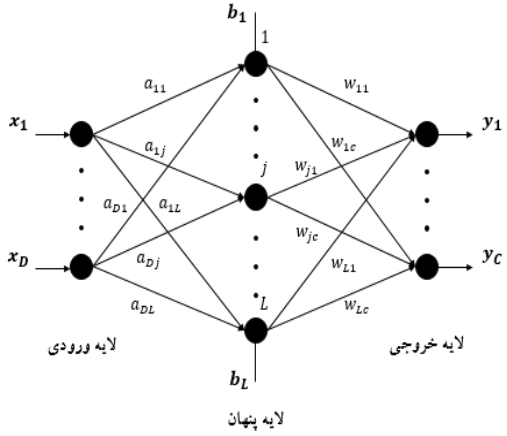
انتخاب ویژگی (با نام انتخاب متغیر[1] نیز شناخته می‌شود) رویکردی است که بر اساس یکسری از معیارها به انتخاب زیرمجموعه‌ای مناسب از ویژگی‌ها از میان مجموعه‌ی تمام ویژگی‌ها می‌پردازد[۶].

۱-۳- شبکه‌های عصبی

امروزه استفاده از هوش مصنوعی جهت تشخیص نفوذ از اهمیت بالایی برخوردار شده است[۲۹]. یکی از مهم‌ترین ویژگی‌های شبکه‌های عصبی مصنوعی که عملکرد آن را به انسان نزدیک‌تر می‌نماید، قدرت یادگیری است. فرآیند یادگیری در شبکه‌های عصبی مصنوعی عبارت است از بهنگام‌سازی معماری شبکه و وزن‌های ارتباطی آن به نحوی است که یک شبکه بتواند یک وظیفه خاص را به‌صورت کارا انجام دهد.[۷] .

۱-۴- ماشین یادگیری مفرط

شبکه‌های عصبی پیشخور با یک لایه مخفی یکی از محبوب‌ترین مدل‌های شبکه عصبی هستند که ساختار ساده شامل یک لایه ورودی، یک لایه مخفی و یک لایه خروجی دارند. اثبات شده است که چنین شبکه ای با یک لایه مخفی با تعداد دلخواه پارامتر و توابع فعال‌سازی دلخواه می‌توانند تقریب زننده جهانی برای هر تابع پیوسته باشند، این مشخصه در سال ۱۹۸۹ توسط سیبنکو[2] و فوناشی[3] تایید شد و اخیرا هوآنگ [4] و بابری[5] توانایی‌های بالای یادگیریشان را نشان دادند[۸]. با این حال، این نوع از شبکه‌های عصبی از مشکل زمان زیاد فرآیند آموزش رنج می‌برند زیرا الگوریتم یادگیری‌شان براساس روش‌های گرادیانت نزولی[6] مانند پس‌انتشار خطا[7] بوده و این الگوریتم‌های یادگیری در مینیمم محلی[8] گیر می‌افتند. به منظور غلبه بر این مشکل، هوآنگ و همکارانش در سال ۲۰۰۴ تکنیک یادگیری ماشین یادگیری مفرط (*ELM*)[9] با هدف کاهش هزینه محاسباتی متحمل شده توسط روش پس انتشار خطا در طول فرآیند آموزش را برای شبکه‌های عصبی پیشخور با یک لایه مخفی پیشنهاد کرده اند که این شبکه‌ها ELM نامیده می‌شوند. ELM بطور گسترده‌ای برای حل مسائل دسته‌بندی و رگرسیون بکار برده می‌شود[۹] . شکل (۲) ساختار شبکه ELM را نشان می‌دهد که لایه ورودی آن تعداد D نورون و لایه مخفی آن تعداد L نورون دارد. تعداد نورون‌های لایه خروجی در مسائل طبقه‌بندی برابر با تعداد کلاس‌ها و در مسائل رگرسیون برابر با تعداد متغیرهای خروجی است که در اینجا شبکه دارای C نورون خروجی است[۱۰] .



شکل (۲): نمای کلی از ماشین یادگیری مفرط[۱۱]

۱-۵- همه برای برنده[10]

سیستم‌هایی که دارای رویکرد همه برای برنده هستند، به کمک اتصال ماژول‌ها (در محیط‌هایی با طراحی وظیفه محور) کار می‌کنند؛ به‌گونه‌ای که وقتی یک عمل انجام می‌شود، انجام تمام اقدامات دیگر متوقف می‌شود، بنابراین در یک زمان تنها یک عمل انجام می‌شود. در این رویکرد، عمل برنده تمام قدرت سیستم موتور را می‌گیرد[۱۲].

۲- پیشینه پژوهش

اغلب مقالات ارائه شده درخصوص رویکردهای انتخاب ویژگی در زمینه‌های تشخیص نفوذ می باشند که بررسی و در قالب جدول (۱) مطرح گردیده است.

[1] Variable Selectio
[2] Cybenco
[3] Funahashi
[4] Huang
[5] Babri
[6] Gradient Descent Methods
[7] Error Back-Propagation
[8] Local Minima
[9] Extreme Learning Machine (ELM)
[10] winner-Take-All

جدول (۱): خلاصه مقالات بررسی شده در سال‌های اخیر

ردیف	سال انتشار	موضوع مورد بررسی	روش مورد استفاده	مزایا و معایب
۱	۲۰۲۲	مقاله‌ای مروری در خصوص رویکردهای اخیر انتخاب ویژگی برای تشخیص نفوذ[۱۳]	مروری بر پیشرفت‌های اخیر در تکنیک‌های انتخاب ویژگی برای تشخیص و طبقه‌بندی حمله، اعمال شده در ناحیه تشخیص نفوذ با در نظر گرفتن طراحی، منطق، ویژگی‌های فنی و معیارهای ارزیابی رایج ارائه شده است.	بررسی چالش‌های موجود و ارائه رویکرد جدید برای تشخیص نفوذ
۲	۲۰۲۲	ارائه یک مدل انتخاب ویژگی موثر با استفاده از الگوریتم‌های فراابتکاری ترکیبی برای تشخیص نفوذ در اینترنت اشیا[۱۴]	انتخاب ویژگی کارآمد از طریق ترکیبی از انتخاب ویژگی همبستگی همراه با ویژگی‌های پانل‌شده جنگل (CFS-FPA) به دست می‌آید. تشخیص نفوذ بهبودیافته شامل بهره‌برداری از الگوریتم‌های یادگیری AdaBoosting و بسته‌بندی برای اصلاح چهار طبقه‌بندی‌کننده ماشین بردار پشتیبان، جنگل تصادفی، بیز ساده و k نزدیکترین همسایه است.	+ افزایش دقت و کاهش خطا - پیچیدگی محاسباتی
۳	۲۰۲۲	ارائه یک مدل انتخاب ویژگی موثر با استفاده از الگوریتم‌های فراابتکاری ترکیبی برای تشخیص نفوذ در اینترنت اشیا[۱۵]	روش انتخاب ویژگی جدید را از طریق افزایش عملکرد بهینه ساز Gorilla Troops (GTO) بر اساس الگوریتم برای ازدحام پرندگان (BSA) ارائه می‌کند. این BSA برای تقویت بهره برداری از عملکرد GTO در GTO-BSA جدید توسعه یافته استفاده می‌شود	+ نرخ هم‌گرایی بهتر و راه‌حل با کیفیت بالاتر - پیچیدگی محاسباتی
۴	۲۰۲۲	ارائه یک نسخه باینری از الگوریتم باروری زمین‌های کشاورزی (*FFA*) به نام*BFFA* برای انتخاب ویژگی در بهبود سیستم‌های تشخیص نفوذ[۱۶]	از تابع V شکل برای جابجایی فرآیندهای FFA در فضای باینری استفاده می‌شود روش پیشنهادی بر روی دو مجموعه داده معتبر IDS یعنی NSL-KDD و UNSW-NB15 آزمایش شده و در معیارهای دقت، صحت، Recallو امتیاز F1 با k نزدیکترین همسایه(KNN) ، ماشین بردار پشتیبان(SVM) ، درخت تصمیم(DT) ، جنگل تصادفی (RF)و بیز ساده (NB) مقایسه شده است.[22]. روش [24] پیشنهادی مبتنی بر ترکیب انتخاب ویژگی بر اساس الگوریتم بهینه‌سازی نهنگ و الگوریتم ژنتیک با طبقه‌بندی KNN از نظر معیارهای دقت، نتایج بهتری نسبت به سایر روش‌های قبلی دارد.[23] در سال ۲۰۲۲ اوتیر و همکارانش، رویکردی جدید برای تشخیص نفوذ با استفاده از بهینه‌سازی گرگ خاکستری (GWO) برای حل مشکلات انتخاب ویژگی و ترکیب آن با بهینه‌سازی ازدحام ذرات (PSO) برای استفاده از بهترین مقدار برای به‌روزرسانی اطلاعات هر موقعیت گرگ خاکستری ارائه کرده اند.[24].	+ افزایش دقت و کاهش خطا - پیچیدگی محاسباتی
۵	۲۰۲۲	ارائه یک سیستم تشخیص نفوذ مبتنی بر انتخاب ویژگی با استفاده از الگوریتم بهینه‌سازی نهنگ ژنتیکی و طبقه‌بندی مبتنی بر نمونه[۱۷]	در این تحقیق از مجموعه داده های استاندارد KDDCUP1999 استفاده شده است که در آن مشخصات مربوط به گره‌های سالم و انواع گره‌های مخرب بر اساس نوع حملات در شبکه ذخیره می‌شود. روش پیشنهادی مبتنی بر ترکیب انتخاب ویژگی بر اساس الگوریتم بهینه‌سازی نهنگ و الگوریتم ژنتیک با طبقه‌بندی KNN از نظر معیارهای دقت، نتایج بهتری نسبت به سایر روش‌های قبلی دارد.	+ افزایش دقت و کاهش خطا - پیچیدگی محاسباتی
۶	۲۰۲۲	ارائه رویکرد بهینه‌سازی برای تشخیص نفوذ با استفاده از بهینه‌سازی گرگ خاکستری (*GWO*) و ترکیب آن با بهینه‌سازی ازدحام ذرات (*PSO*) [۱۸]	انتخاب ویژگی و ترکیب آن با بهینه‌سازی ازدحام ذرات (PSO) برای استفاده از بهترین مقدار برای به‌روزرسانی اطلاعات هر موقعیت گرگ خاکستری ارائه کردند. این تکنیک بهترین اطلاعات موقعیت فرد را توسط الگوریتم PSO حفظ می‌کند، که از سقوط الگوریتم GWO به یک بهینه محلی جلوگیری می کند.	+ افزایش دقت و کاهش خطا - پیچیدگی محاسباتی
۷	۲۰۲۱	ازائه یک سیستم تشخیص نفوذ پیشرفته (*IDS*) با استفاده از بهینه‌ساز گرگ خاکستری باینری اصلاح شده با ماشین بردار پشتیبان[۱۹]	در روش پیشنهادی از ۳ گرگ، ۵ گرگ و ۷ گرگ برای یافتن بهترین تعداد گرگ استفاده شد. هدف روش پیشنهادی افزایش دقت تشخیص نفوذ و نرخ تشخیص و کاهش زمان پردازش در محیط WSN از طریق کاهش نرخ آلارم‌های کاذب و تعداد ویژگی‌های حاصل از IDSها در محیط WSN است. مجموعه داده NSL KDD'99 برای نشان دادن عملکرد روش پیشنهادی و مقایسه آن با سایر روش‌های موجود استفاده شده است. روش‌های پیشنهادی از نظر دقت، تعداد ویژگی‌ها، زمان اجرا، نرخ هشدار نادرست و نرخ تشخیص ارزیابی می‌شوند.	+ افزایش دقت و کاهش خطا - پیچیدگی محاسباتی
۸	۲۰۲۱	ارائه یک روش انتخاب ویژگی مبتنی بر بهبود الگوریتم جستجوی کلاغ که در سیستم تشخیص نفوذ برای محدود کردن اندازه مجموعه‌داده[۲۰]	از الگوریتم جستجوی کلاغ در سیستم تشخیص نفوذ به عنوان مدلی برای یافتن زیرمجموعه ویژگی بهینه و جنگل تصادفی به عنوان قضاوت در مورد ویژگی‌هایی که توسط CSA-IDS تولید می‌شوند، استفاده می کند. مجموعه داده‌های KDD و UNSW برای ارزیابی مدل پیشنهادی قبلی استفاده می‌شوند. مدل پیشنهادی به دقت ۹۹,۸۴ درصد برای تشخیص حمله با استفاده از مجموعه داده های UNSW دست یافت. به طور مشابه، حملات R2L و U2R دقت ۹۹,۹۷٪ را برای مجموعه داده NSL-KDD شناسایی کرده اند.	+ افزایش دقت و کاهش خطا - پیچیدگی محاسباتی

جدول (۱): خلاصه مقالات بررسی شده در سال‌های اخیر

ردیف	سال انتشار	موضوع مورد بررسی	روش مورد استفاده	مزایا و معایب
۹	۲۰۲۱	ارائه یک سیستم تشخیص نفوذ هوشمند مبتنی بر ناهنجاری برای شبکه اینترنت اشیا با استفاده از الگوریتم‌های گرگ خاکستری، بهینه‌سازی ازدحام ذرات و جنگل تصادفی[۲۱]	از داده‌های آموزشی و برچسب داده‌ها KDD-Cup'99 با الگوریتم جنگل تصادفی استفاده شده است. پس از آن از داده‌های آزمایشی استفاده می شود. از داده‌های ذخیره شده در مرحله آموزش استفاده می‌شود که در واقع یک کپی از داده‌ها است، به طوری که در هنگام انجام مرحله آزمون، می‌توان همان داده‌های آموزشی را با طبقه‌بندی با استفاده از الگوریتم PSO مقایسه کرد. دقت روش پیشنهادی در این مقاله به ۹۷ درصد رسید. با روش پیشنهادی سرعت یادگیری بسیار افزایش یافته و دقت قابل قبولی دارد	+ افزایش دقت و کاهش خطا - پیچیدگی محاسباتی
۱۰	۲۰۲۱	ارائه یک رویکرد جدید تشخیص نفوذ شبکه اینترنت اشیا مبتنی بر شبکه عصبی کانولوشن[1] و بهینه‌سازی ازدحام ذارت تطبیقی[۲۲]	الگوریتم PSO با تغییر وزن اینرسی برای بهینه سازی تطبیقی پارامترهای ساختار CNN یک بعدی استفاده می شود. مقدار تابع زیان متقاطع آنتروپی، که از اولین آموزش CNN به دست می‌آید، به عنوان مقدار برازندگی PSO در نظر گرفته می‌شود. نتایج شبیه‌سازی نشان می‌دهد که تکلیف چند نوع تشخیص حمله نفوذ شبکه اینترنت اشیا بر اساس الگوریتم APSO-CNN مؤثر و قابل اعتماد است	+ افزایش دقت و اثربخشی - پیچیدگی محاسباتی
۱۱	۲۰۲۱	ارائه یک روش موثر انتخاب ویژگی مبتنی بر الگوریتم ژنتیک برای سیستم‌های تشخیص نفوذ[۲۳]	روش انتخاب ویژگی مبتنی بر الگوریتم ژنتیک (GA) بهبود یافته، به نام انتخاب ویژگی مبتنی بر GA (GbFS)، برای افزایش دقت طبقه‌بندی‌کننده ارائه می‌دهد. ایمن سازی شبکه از حملات سایبری یک وظیفه حیاتی است و باید تقویت شود. مقایسه‌ای نیز با روش های استاندارد انتخاب ویژگی انجام می‌شود. نتایج نشان می‌دهد که دقت با استفاده از GbFS به حداکثر دقت ۹۹/۸۰ درصد دست می‌یابد	+ افزایش دقت و کاهش خطا - پیچیدگی محاسباتی
۱۲	۲۰۲۱	ارائه یک سیستم تشخیص نفوذ مبتنی بر طبقه‌بندی‌کننده ژنتیک – فازی برای تشخیص حملات مخرب[۲۴]	یک مدل جدید به نام سیستم تشخیص نفوذ مبتنی بر رویکرد ترکیبی (GA-Fuzzy) برای مدیریت مجموعه داده‌های NSL-KDD با حجم زیاد برای تشخیص حملات موثر و کاهش نرخ هشدار طبقه‌بندی اشتباه پیشنهاد شده است. که کارایی را بهبود می بخشد و به دقت تشخیص ۹۹/۹۶٪ و نرخ هشدار نادرست ۰/۰۴٪ می‌رسد	+ افزایش دقت - پیچیدگی محاسباتی
۱۳	۲۰۲۱	یک سیستم تشخیص نفوذ اصلاح‌شده با استفاده از الگوریتم کرم شب‌تاب در محیط ابری[۲۵]	IDS پیشنهادی امنیت کار بر روی ایده انتخاب ویژگی را فراهم می‌کند. نویسندگان یک الگوریتم کرم شب‌تاب اصلاح شده تهیه کرده‌اند که به عنوان یک روش انتخاب ویژگی ماهر عمل می‌کند و مجموعه داده NSL-KDD را قادر می‌سازد تا فضای ذخیره‌سازی کمتری را با کاهش ابعاد و همچنین زمان آموزش کمتر با دقت طبقه بندی بیشتر مصرف کند	+ افزایش دقت و کارایی - پیچیدگی محاسباتی
۱۴	۲۰۲۱	ارائه یک سیستم تشخیص نفوذ برای پیشرفت الگوریتم‌های یادگیری عمیق مبتنی بر زیرساخت اینترنت اشیا[۲۶]	رویکرد پیشنهادی، توسعه یک سیستم چارچوب قوی برای تشخیص نفوذ بر اساس محیط اینترنت اشیا است. مجموعه‌داده IoTID20 برای توسعه سیستم پیشنهادی استفاده شد. این مجموعه‌داده جدید از زیرساخت اینترنت اشیا ایجاد شده است. در این چارچوب، سه الگوریتم یادگیری عمیق پیشرفته برای طبقه‌بندی نفوذ استفاده شد: شبکه عصبی کانولوشن (CNN)، حافظه کوتاه‌مدت بلند مدت (LSTM) و ترکیب شبکه عصبی کانولوشن با حافظه کوتاه‌مدت (CNN-LSTM). برای بهبود سیستم پیشنهادی، از روش بهینه‌سازی ازدحام ذرات (PSO) برای انتخاب ویژگی‌های مرتبط از مجموعه‌داده شبکه استفاده شد.	+ افزایش دقت و اثربخشی - پیچیدگی محاسباتی
۱۵	۲۰۲۰	ارائه یادگیری ماشین مفرط مبتنی بر انتخاب ویژگی بدون نظارت را برای خوشه‌بندی[۲۷]	با به حداقل رساندن نُرم $l_{1,2}$ وزن خروجی، انتخاب ویژگی را انجام می‌دهد و نتایج خوشه‌بندی توسط ترکیب ویژه محاسبه می‌شود. با حل مسئله بهینه‌سازی فرموله شده به روشی تکراری، دقت خوشه‌بندی را بهبود می‌بخشد.	+ افزایش دقت و اثربخشی - بیش برازش
۱۶	۲۰۲۰	ارائه یک سیستم تشخیص نفوذ مشارکتی بهینه شده (*OCIDS*) برای شبکه‌های حسگر بی سیم[۲۸]	از الگوریتم بهینه‌سازی کلونی زنبورهای مصنوعی بهبود یافته برای بهینه‌سازی IDS سلسله مراتبی اعمال شده در WSN ها با توجه به دقت تشخیص نفوذ و همچنین مصرف منابع محدود استفاده می‌کند. علاوه بر این، سیستم پیشنهادی الگوریتم ماشین بردار پشتیبانی وزنی را برای بهبود دقت تشخیص و کاهش نرخ هشدار کاذب بهینه می‌کند.	+ افزایش دقت و اثربخشی - پیچیدگی محاسباتی
۱۷	۲۰۲۴	تشخیص نفوذ با نزول شیب تصادفی برای سیستم تشخیص حمله در	این تحقیق رویکرد جدیدی از سیستم تشخیص نفوذ (IDS) به نام SG-IDS را پیشنهاد می‌کند که از الگوریتم‌های نزول شیب تصادفی	+ دقت بهبود یافته نسبت به روش‌های پیشرفته موجود.

[1] Convolutional Neural Network (CNN)

جدول (۱): خلاصه مقالات بررسی شده در سال‌های اخیر

ردیف	سال انتشار	موضوع مورد بررسی	روش مورد استفاده	مزایا و معایب
		شبکه‌های حسگر بی‌سیم با استفاده از یادگیری ماشین[۳۰]	(SGD) و بیز ساده گاوسی (GNB) برای بهبود تشخیص نفوذ در شبکه‌های حسگر بی‌سیم (WSN) استفاده می‌کند.	+ کاهش پیچیدگی محاسباتی به دلیل انتخاب ویژگی با استفاده از PCA و SVD. + قابل‌اعمال بر روی هر دو شبکه WSN و شبکه‌های IoMT. - نیاز به تنظیم دقیق پارامترها برای عملکرد بهینه. - احتمال overfitting (برازش بیش از حد) در صورتی که داده‌های آموزشی نماینده سناریوهای دنیای واقعی نباشند.
۱۸	۲۰۲۳	سیستم تشخیص نفوذ با استفاده از الگوریتم‌های طبقه‌بندی ژنتیکی عصبی فازی (FNGCA) و بهینه‌سازی اجتماع ذرات مبتنی بر قانون (RBPSO) در شبکه‌های حسگر بی‌سیم[۳۱]	بهینه‌سازی اجتماع ذرات مبتنی بر قانون (RBPSO) برای انتخاب ویژگی و الگوریتم طبقه‌بندی ژنتیکی عصبی فازی (FNGCA) برای طبقه‌بندی	+ کاهش مصرف انرژی، بهبود امنیت ارتباطات، افزایش نرخ تشخیص، ارتباط قابل اعتماد - افزایش پیچیدگی، هزینه محاسباتی بالاتر
۱۹	۲۰۲۳	تشخیص نفوذ مبتنی بر انتخاب ویژگی با بهینه‌سازی چندهدفه اجتماع ذرات (PSO) برای شبکه‌های حسگر بی‌سیم مبتنی بر اینترنت اشیا (IoT)[۳۲]	بهینه‌سازی چندهدفه اجتماع ذرات هوشمند (IMOPSO) و ماشین بردار پشتیبان چند کلاسه مبتنی بر قانون (rule-based MSVM)	+ دقت و نرخ تشخیص بالاتر. + نرخ مثبت کاذب پایین‌تر. + انتخاب ویژگی کارآمدتر. - الگوریتم پیچیده‌تر. - نیاز به منابع محاسباتی بیشتر.
۲۰	۲۰۲۴	یک رویکرد پیش‌بینی مبتنی بر موقعیت برای کشف و پیشگیری از نفوذ امنیت سایبری با استفاده از الگوریتم‌های یادگیری ماشین و یادگیری عمیق در شبکه‌های حسگر بی‌سیم مبتنی بر صنعت ۴,۰[۳۳]	پرسپترون چند لایه، رمزگذار خودکار، درخت تصمیم	+ اولویت‌بندی هوشمند، سیستم پیشگیری پیش‌فعال، رویکرد چند معیاره - نیاز به داده آموزشی، ممکن است برای همه انواع حملات مناسب نباشد
۲۱	۲۰۲۳	الگوریتم بهینه‌سازی باینری شامپانزه با تشخیص نفوذ مبتنی بر یادگیری ماشین برای ایمن‌سازی شبکه‌های حسگر بی‌سیم تحت پوشش اینترنت اشیا[۳۴]	الگوریتم باینری بهینه‌سازی شامپانزه (BCOA) برای انتخاب ویژگی و ماشین یادگیری شدید با تنظیم هزینه بر اساس کلاس (CCR-ELM) برای طبقه‌بندی	+ BCOA الگوریتمی ساده، قوی و با سرعت همگرایی بالا است. + CCR-ELM در رسیدگی به مسائل عدم تعادل کلاس (class imbalance) مؤثر است. + رویکرد پیشنهادی به دقت بالایی در تشخیص نفوذ دست می‌یابد. - BCOA ممکن است مستعد بهینه‌های محلی (local optima) باشد. - CCR-ELM نیازمند تنظیم دقیق پارامترها است.
۲۲	۲۰۲۴	تشخیص نفوذ سایبری در شبکه‌های حسگر بی‌سیم با استفاده از تکنیک ترکیبی کاهش ویژگی با روش‌های هوش مصنوعی و یادگیری ماشین[۳۵]	خوشه‌بندی K-means، انتخاب ویژگی بر اساس gain اطلاعات، متوازن‌سازی داده با SMOTE-based ENN، شبکه عصبی عمیق پیش‌رو (DFFNN)	+ دقت بالا، عملکرد بهبود یافته نسبت به روش‌های سنتی یادگیری ماشین - نیاز به داده برچسب‌گذاری شده، پرهزینه از نظر محاسباتی

۳- روش پیشنهادی

با توجه به اینکه پردازش داده‌ها با اعداد کار می‌کنند و برخی از مشخصه‌های این مجموعه‌داده دارای مقادیر رشته‌ای هستند، لذا تغییر آن‌ها به مقادیر عددی برای انجام پردازش الزامی می‌باشد. برای تبدیل نوع‌های رشته‌ای به عدد نیز به این صورت عمل می‌شود که به ازای هر نوع یک عدد منحصربه‌فرد در نظر گرفته می‌شود. گام اول در ایجاد هر مدلی مبتنی بر تکنیک‌های داده‌کاوی، مرحله پیش‌پردازش داده‌ها می‌باشد. این مرحله شامل نرمال‌سازی و درهم‌ریختن داده‌ها می‌باشد. در مرحله بعد با استفاده از ماشین یادگیری مفرط بدون نظارت ویژگی‌های مطلوب استخراج می‌شود. در برخی از مجموعه داده‌ها تعداد ویژگی‌ها زیاد می‌باشند و ممکن است برخی از این ویژگی‌ها در طبقه‌بندی داده‌ها نقشی نداشته باشند. بنابراین نیاز است که زیرمجموعه‌ای از بهترین ویژگی‌ها انتخاب شوند. روشی که برای انتخاب ویژگی‌ها در این پژوهش درنظر گرفته شده است، براساس اندازه‌گیری توزیع ویژگی‌ها عمل می‌کند. به این صورت که در تکرارهای متوالی ویژگی‌های سنجیده شده توسط یادگیری ماشین مفرط بدون نظارت شناسایی شده و از آن‌ها در ساخت راه‌حل‌های جدید استفاده می‌شود. بعد از انتخاب ویژگی‌ها دادگان خوشه‌بندی می‌شوند. در مرحله آخر، خوشه‌بندی داده‌ها با ویژگی‌های استخراج شده محاسبه می‌شود. منظور از تعداد ویژگی‌ها مشخص‌کردن تعداد ویژگی‌های انتخابی از کل ویژگی‌ها در مجموعه‌داده می‌باشد. بعد از تعیین ویژگی‌های مطلوب نهایی، صحت خوشه‌بندی ویژگی‌های انتخاب شده توسط یادگیری ماشین مفرط بدون نظارت نشان داده می‌شود.

روش پیشنهادی و مقاله[۳۶] هر دو در مورد تشخیص نفوذ در شبکه‌های حسگر بی‌سیم می باشند. اما این مقاله رویکردی جدید برای انتخاب ویژگی و تشخیص نفوذ با استفاده از ماشین یادگیری افراطی بدون نظارت (UELM) ارائه می‌دهد در حالی که مقاله[۳۶] از الگوریتم بهینه‌سازی گرگ خاکستری (GWO) برای انتخاب ویژگی در سیستم‌های تشخیص نفوذ و از رویکرد یادگیری با نظارت تمرکز دارد.

داده‌های عمومی توزیع‌های پیچیده متفاوتی در فضاهای اصلی خود دارند، خواص مشابه نمونه‌های داده را نمی توان به راحتی آشکار کرد. بنابراین، معمولاً خوشه‌بندی داده‌های عمومی به شدت به ویژگی‌های اصلی آنها بستگی دارد. لذا ابتدا از نگاشت غیرخطی برای تبدیل داده‌ها به یک فضای پنهان استفاده می‌گردد که در آن ویژگی‌های مشابه نمونه‌ها را می‌توان با ویژگی‌ها نشان داد. سپس، ویژگی‌های انتخاب شده می‌توانند نشانگر خوشه را در فضای خروجی تقریب بزنند. علاوه بر این، از منظم‌سازی منیفولد برای حفظ هندسه ذاتی فضای خروجی به‌صورت مشابه در فضای اصلی استفاده می‌شود. در نهایت، شاخص خوشه را از خروجی با هر دو محدودیت غیر منفی و متعامد محاسبه می‌گردد. گام های روش پیشنهادی به صورت زیر شرح داده می شود:

جدول (۲): گام های روش پیشنهادی

ردیف	گام	ردیف	گام
۱	ورود دادگان	۴	دسته بندی و تشخیص نفوذ
۲	بهبود تابع هدف یادگیری ماشین مفرط بدون نظارت	۵	محاسبه معیار ارزیابی
۳	انتخاب ویژگی		

۳-۱- ورود داده

مجموعه‌داده مورد استفاده در این پژوهش مجموعه داده *NSL-KDD* است

۳-۲ -تابع هدف (انتخاب ویژگی)

یادگیری بی‌نظارت مبتنی بر این فرضیه است: اگر دو نقطه ی x_1 و x_2 نزدیک هم باشند، احتمال‌های شرطی آن‌ها $P(y|x_1)$ و $P(y|x_2)$ باید مشابه باشند. برای اجرای این فرض بر داده‌ها، رابطه‌ی (۱) بدست آورده می‌شود:

$$L_m = \frac{1}{2}\sum_{i,j} w_{ij}\left\|P(y|x_i) - P(y|x_j)\right\|^2 \quad (۱)$$

در فرآیند خوشه‌بندی دو معیار مورد استفاده قرار می‌گیرد شباهت و فاصله در ماشین یادگیری مفرط بدون نظارت و رابطه (۱) رابطه نشان می‌دهد براساس احتمالات کدام ورودی به کدام خروجی تعلق خواهد داشت به عبارتی عمل خوشه بندی صورت می‌پذیرد.

که در آن w_{ij} شباهت بین x_1 و x_2 است، که می‌توان آن را با تابع گوسین $exp\left(-\left\|x_i - x_j\right\|^2/2\sigma^2\right)$ حساب کرد. به دلیل اینکه محاسبه‌ی احتمال‌های شرطی سخت است، رابطه‌ی (۲) می‌تواند رابطه‌ی فوق را تقریب بزند.

$$\hat{L}_m = Tr\left(\hat{Y}^T L\hat{Y}\right) \quad (۲)$$

که در آن $Tr(\cdot)$ اثر ماتریس را نشان می‌دهد، $\hat{Y}$ پیش‌بینی‌های مجموعه‌داده بدون برچسب است، $L = D - W$ گراف لاپلاسین است و D ماتریس قطری با عناصر قطری $D_{ii} = \sum_{j=1}^{u} w_{ij}$ است. در یادگیری بی‌نظارت، مجموعه‌داده $X = \{x_i\}_{i=1}^{N}$ بدون برچسب است. مسئله بهینه‌سازی ماشین یادگیری مفرط بدون نظارت برای انتخاب ویژگی به صورت (۳) فرموله می‌شود:

$$\min_{\beta,\mathbf{F}} \|\bar{\boldsymbol{H}}\beta - \boldsymbol{F}\|_F^2 + \lambda_1\|\beta\|_{2,1} + \lambda_2\frac{R}{2}\mathrm{Tr}[\beta^T\bar{\boldsymbol{H}}^T\boldsymbol{L}\bar{\boldsymbol{H}}\beta] \quad (۳)$$
$$s.t \qquad \boldsymbol{F}\epsilon[0,1]^{n\times c}$$

که در آن $\overline{\boldsymbol{H}}$ و β به ترتیب نورون های پنهان نرمال شده و وزن های خروجی ماشین یادگیری مفرط بدون نظارت هستند، R ضریب تنظیم (رگولاریزیشن) خطاهای آموزش، $\boldsymbol{L}$ ماتریس لاپلاسین از پیش تعریف شده و $\boldsymbol{F}$ ماتریس نشانگر خوشه است. ابتدا، نورون‌های تصادفی در لایه پنهان، داده‌های اصلی را با نگاشت غیرخطی گسترش می‌دهند و مجموعه ویژگی‌های بزرگ $\boldsymbol{H}$را تولید می‌کنند که سپس به میانگین صفر نرمال می‌شود.

پارامترهای نورون های اولیه به طور تصادفی نیازی به تغییر ندارند. ماتریس داده اصلی $\boldsymbol{X}$ فقط برای ساختن نرمال شده $\overline{\boldsymbol{H}}$ و ماتریس لاپلاسین $\boldsymbol{L}$ استفاده می شود که در طول فرآیند آموزش ثابت هستند. ایده کلیدی روش پیشنهادی یادگیری وزن های خروجی بین لایه پنهان و لایه خروجی است، به طوری که خروجی $\overline{\boldsymbol{H}}\beta$ می تواند نسخه ماتریس نشانگر خوشه $\boldsymbol{F}$ را تقریب بزند زیرا میانگین مربعات خطای بین آن‌ها به حداقل می رسد.

جدول (۳): شبه کد روش پیشنهادی

The UFSELM algorithm.

Input: Normalized data $\boldsymbol{X}\epsilon\mathbb{R}^{n\times d}$,
Laplacian matrix $\boldsymbol{L}\ \epsilon\mathbb{R}^{n\times n}$,
Number of clusters $\boldsymbol{c}$,
Number of neurons $\boldsymbol{m}$,
Trade-off parameters $\boldsymbol{\lambda_1 and\ \lambda_2}$,
Number of iterations $\boldsymbol{t}$.
Output: Cluster indicator matrix $\boldsymbol{F}\ \epsilon\mathbb{R}^{n\times c}$.

1. **Begin**
2. **Initialize the cluster indicator matrix F by k-Means clustering on X.**
3. **Generate input weights and biases of hidden neurons randomly and calculate the centered hidden layer output matrix $\widehat{\boldsymbol{H}}$.**
4. **Initialize $\boldsymbol{B}\ \epsilon\mathbb{R}^{m\times m}$ as an identity matrix.**
5. **Compute $\boldsymbol{Q}$ by $\boldsymbol{Q}=\widehat{\boldsymbol{H}}^{\boldsymbol{T}}\widehat{\boldsymbol{H}}+\boldsymbol{\lambda_1 B}+\boldsymbol{\lambda_2}\widehat{\boldsymbol{H}}^{\boldsymbol{T}}\boldsymbol{L}\widehat{\boldsymbol{H}}$**
6. **Repeat**
7. **Update $\boldsymbol{\beta}=\boldsymbol{Q}^{-1}\widehat{\boldsymbol{H}}\boldsymbol{F}$**
8. **foreach *diagonal element* of $\boldsymbol{B}$ do**
9. **Update $\boldsymbol{B_{ii}}=\frac{1}{\sqrt{\beta_i^T\beta_i+\epsilon}}$**
10. **end**
11. **Update by $\boldsymbol{Q}=\widehat{\boldsymbol{H}}^{\boldsymbol{T}}\widehat{\boldsymbol{H}}+\boldsymbol{\lambda_1 B}+\boldsymbol{\lambda_2}\widehat{\boldsymbol{H}}^{\boldsymbol{T}}\boldsymbol{L}\widehat{\boldsymbol{H}}$**
12. **Update F by solving problem**
13. **Until *Reaching t-th iteration*.;**
14. **End**

الگوریتم UFSELM

ورودی: ماتریس داده ورودی $\boldsymbol{X}\epsilon\mathbb{R}^{n\times d}$
ماتریس لاپلاسین $\boldsymbol{L}\ \epsilon\mathbb{R}^{n\times n}$
تعداد کلاس / خوشه $\boldsymbol{c}$
ابعاد لایه پنهان $\boldsymbol{m}$
پارامترهای مصالحه $\boldsymbol{\lambda_1 and\ \lambda_2}$,
تعداد تکرار $\boldsymbol{t}$
Output: ماتریس خوشه شاخص (ویژگی) $\boldsymbol{F}\ \epsilon\mathbb{R}^{n\times c}$.

۱۵. **شروع**
۱۶. **ماتریس نشانگر خوشه F را با خوشه بندی k-Means روی X راه اندازی کنید..**
۱۷. **وزن های ورودی و بایاس های نورون های پنهان را به طور تصادفی ایجاد کنید و ماتریس خروجی لایه پنهان مرکزی H را محاسبه کنید.**
۱۸. **Bبه عنوان یک ماتریس مورب مقداردهی اولیه می‌شود.**
۱۹. **محاسبه کنید $\boldsymbol{Q}$ بوسیله $\boldsymbol{Q}=\widehat{\boldsymbol{H}}^{\boldsymbol{T}}\widehat{\boldsymbol{H}}+\boldsymbol{\lambda_1 B}+\boldsymbol{\lambda_2}\widehat{\boldsymbol{H}}^{\boldsymbol{T}}\boldsymbol{L}\widehat{\boldsymbol{H}}$**
۲۰. **تکرار کردن**
۲۱. **بروزرسانی $\boldsymbol{\beta}=\boldsymbol{Q}^{-1}\widehat{\boldsymbol{H}}\boldsymbol{F}$**
۲۲. **برای هر عنصر مورب $\boldsymbol{B}$**
۲۳. **بروزرسانی کنید $\boldsymbol{B_{ii}}=\frac{1}{\sqrt{\beta_i^T\beta_i+\epsilon}}$**
۲۴. **پایان**
۲۵. **بروزرسانی بوسیله $\boldsymbol{Q}=\widehat{\boldsymbol{H}}^{\boldsymbol{T}}\widehat{\boldsymbol{H}}+\boldsymbol{\lambda_1 B}+\boldsymbol{\lambda_2}\widehat{\boldsymbol{H}}^{\boldsymbol{T}}\boldsymbol{L}\widehat{\boldsymbol{H}}$**
۲۶. **بروزرسانی F**
۲۷. **رسیدن تا تکرار نهایی ;t.**
۲۸. **پایان**

در ماشین یادگیری بدون نظارت هنجار وزن‌های خروجی β به حداقل می‌رسد تا حاشیه‌های جداکننده دو کلاس مختلف در فضای ویژگی به حداکثر برسد. علاوه بر این، مجموعه ویژگی‌های بزرگ تولید شده توسط لایه پنهان احتمالاً از ویژگی‌های بی‌ارزش تشکیل شده است. برای گرفتن مجموعه ویژگی‌هایی که عملکرد بهتری را به دست می‌آورند، هنجار $L_{2,1}$ را اتخاذ می‌کنیم. هنجار $L_{2,1}$یک ماتریس به عنوان هنجار L_1 هنجارهای L_2 هر ردیف از ماتریس تعریف می‌شود. برای بسیاری از ویژگی‌های بی‌اهمیت اتفاق می‌افتد که تمام وزن‌های مربوط به یک ویژگی فردی به طور همزمان به صفر کاهش می‌یابد، زمانی که هنجار $L_{2,1}$ ماتریس وزن‌های خروجی به حداقل برسد. با باقی گذاشتن تنها تعداد کمی از ردیف‌های غیرصفر، مهم‌ترین ویژگی‌ها انتخاب می‌شوند. بنابراین، برای جاسازی انتخاب ویژگی در فرمول مسئله و افزایش حاشیه‌های جداکننده، ماتریس وزن خروجی $L_{2,1}$ را به تابع بهینه‌سازی اضافه می‌گردد. دو عبارت اول با هم یک مدل رگرسیون پراکنده را تشکیل می‌دهند. پراکندگی توسط پارامتر مبادله λ_1 کنترل می‌شود، اگر مقدار λ_1 بسیار بزرگ بود، تمام عناصر β به صفر نزدیک می‌شوند.

جدول (۴): معرفی نمادها

نماد	تعریف	نماد	تعریف
n	تعداد نمونه‌ها	$H \in \mathbb{R}^{n\times m}$	خروجی ماتریس مرکزی لایه پنهان
c	تعداد کلاس / خوشه	$\beta \in \mathbb{R}^{m\times c}$	وزن لایه خروجی
d	ابعاد لایه ورودی	$L \in \mathbb{R}^{n\times n}$	ماتریس لاپلاسین
m	ابعاد لایه پنهان	I	ماتریس همانی
$X \in \mathbb{R}^{n\times d}$	ماتریس داده ورودی	λ_1, λ_2	پارامترهای مصالحه
$Y \in \mathbb{R}^{n\times c}$	ماتریس برچسب کلاسی	$H \in \mathbb{R}^{n\times c}$	خروجی ماتریس لایه پنهان
$F \in \mathbb{R}^{n\times c}$	ماتریس خوشه شاخص (ویژگی)		

اگر دو نمونه داده ورودی نزدیک به یکدیگر باشند، برچسب‌های مربوطه آنها به نسبت نزدیک هستند. بنابراین، منظم‌سازی منیفولد در اینجا برای حفظ هندسه ذاتی توزیع داده استفاده می‌گردد. برای به‌حداقل‌رساندن پارامتر تنظیم، ماتریس وزن‌های خروجی باید به‌گونه‌ای تنظیم شود که خروجی $\overline{\boldsymbol{H}}\beta$ دارای هندسه ذاتی مشابه با ماتریس ورودی $\boldsymbol{X}$ باشد. همانطور که مسئله فرموله شده است، هدف ماشین یادگیری بدون نظارت برای انتخاب ویژگی، یادگیری خوشه‌بندی با تنظیم وزن‌های خروجی برای انتخاب ویژگی‌هاست تا خروجی ماشین یادگیری بدون نظارت بتواند ماتریس نشانگر خوشه را با ساختار هندسی مشابه داده‌های اصلی تقریب کند.

۳-۳- بهینه‌سازی متناوب (انتخاب ویژگی و دسته‌بندی)

بهینه‌سازی متناوب برای حل مسئله بهینه‌سازی فرمول‌بندی شده اتخاذ شده است. هنجار $L_{2,1}$ را می‌توان به شکل $\mathrm{Tr}[\beta^T B\beta]$ بیان کرد که در آن B یک ماتریس مورب با $B_{ii} = \frac{1}{\sqrt{\beta_i^T\beta_i+\epsilon}}$ است. سپس مسئله بهینه‌سازی به رابطه (۴) تبدیل می‌شود:

$$\min_{\beta,\mathbf{F}} \|\overline{\boldsymbol{H}}\beta - \boldsymbol{F}\|_F^2 + \lambda_1 \mathrm{Tr}[\beta^T B\beta] + \lambda_2 \mathrm{Tr}[\beta^T \overline{\boldsymbol{H}}^T \boldsymbol{L}\overline{\boldsymbol{H}}\beta] \quad (۴)$$
$$s.t \quad \boldsymbol{F}\epsilon[0,1]^{n\times c}$$

برای حل مسئله فرمول‌بندی‌شده، ابتدا نشانگر خوشه‌ای $\boldsymbol{F}$ اولیه از خوشه‌بندی k-$Means$ بر روی داده‌های اصلی به‌دست می‌آید. با ثابت بودن $\boldsymbol{F}$، می‌توان با تنظیم مشتق رابطه (۳-۲) برابر با صفر، جواب بهینه را به دست آورد.

$$2\overline{\boldsymbol{H}}^T(\overline{\boldsymbol{H}}\beta - \boldsymbol{F}) + 2\lambda_1 B\beta + 2\lambda_2 \overline{\boldsymbol{H}}^T \boldsymbol{L}\overline{\boldsymbol{H}}\beta = 0 \quad (۵)$$

بدین ترتیب،

$$\beta = Q^{-1}\overline{\boldsymbol{H}}^T\boldsymbol{F} \quad (۶)$$

که در آن $Q = \overline{\boldsymbol{H}}^T\overline{\boldsymbol{H}} + \lambda_1 B + \lambda_2 \overline{\boldsymbol{H}}^T \boldsymbol{L}\overline{\boldsymbol{H}}$یک ماتریس متقارن است. اشاره شد که B به عنوان یک ماتریس مورب مقداردهی اولیه می‌شود و در هر تکرار پس از به‌روزرسانی β به روز می‌شود. تابع زیان (۳-۴) را می‌توان به صورت بازنویسی کرد:

$$\begin{aligned}&\mathrm{Tr}[(\beta^T\bar{\boldsymbol{H}}^T-\boldsymbol{F}^T)(\bar{\boldsymbol{H}}\beta-\boldsymbol{F})+\lambda_1\beta^T B\beta\\&\quad+\lambda_2\beta^T\bar{\boldsymbol{H}}^T\boldsymbol{L}\bar{\boldsymbol{H}}\beta]\\&=\mathrm{Tr}[\beta^T\bar{\boldsymbol{H}}^T\bar{\boldsymbol{H}}\beta-\beta^T\bar{\boldsymbol{H}}^T\boldsymbol{F}\\&\quad-\boldsymbol{F}^T\bar{\boldsymbol{H}}\beta+\boldsymbol{F}^T\boldsymbol{F}+\lambda_1\beta^T B\beta\\&\quad+\lambda_2\beta^T\bar{\boldsymbol{H}}^T\boldsymbol{L}\bar{\boldsymbol{H}}\beta]\\&=\mathrm{Tr}[\beta^T(\bar{\boldsymbol{H}}^T\bar{\boldsymbol{H}}+\lambda_1 B\\&\quad+\lambda_2\bar{\boldsymbol{H}}^T\boldsymbol{L}\bar{\boldsymbol{H}})\beta-\beta^T\bar{\boldsymbol{H}}^T\boldsymbol{F}\\&\quad-\boldsymbol{F}^T\bar{\boldsymbol{H}}\beta+\boldsymbol{F}^T\boldsymbol{F}]\end{aligned}\quad(۷)$$

با جایگزینی رابطه (۶) و $Q=\bar{\boldsymbol{H}}^T\bar{\boldsymbol{H}}+\lambda_1 B+\lambda_2\bar{\boldsymbol{H}}^T\boldsymbol{L}\bar{\boldsymbol{H}}$ در رابطه (۷):

$$\begin{aligned}&\mathrm{Tr}[\boldsymbol{F}^T\bar{\boldsymbol{H}}Q^{-1}(\bar{\boldsymbol{H}}^T\bar{\boldsymbol{H}}+\lambda_1 B\\&+\lambda_2\bar{\boldsymbol{H}}^T\boldsymbol{L}\bar{\boldsymbol{H}})Q^{-1}\bar{\boldsymbol{H}}^T\boldsymbol{F}\\&-2\boldsymbol{F}^T\bar{\boldsymbol{H}}Q^{-1}\bar{\boldsymbol{H}}^T\boldsymbol{F}+\boldsymbol{F}^T\boldsymbol{F}]\\&=\mathrm{Tr}[\boldsymbol{F}^T(\bar{\boldsymbol{H}}Q^{-1}QQ^{-1}\bar{\boldsymbol{H}}^T\\&-2\bar{\boldsymbol{H}}Q^{-1}\bar{\boldsymbol{H}}^T+\boldsymbol{I_c})\boldsymbol{F}]\\&=\mathrm{Tr}[\boldsymbol{F}^T(\bar{\boldsymbol{H}}Q^{-1}\bar{\boldsymbol{H}}^T-2\bar{\boldsymbol{H}}Q^{-1}\bar{\boldsymbol{H}}^T\\&+\boldsymbol{I_c})\boldsymbol{F}]\\&=\mathrm{Tr}[\boldsymbol{F}^T(\boldsymbol{I_c}\\&-\bar{\boldsymbol{H}}Q^{-1}\bar{\boldsymbol{H}}^T)\boldsymbol{F}]\end{aligned}\quad(۸)$$

بنابراین، مسئله بهینه‌سازی با توجه به $\boldsymbol{F}$ تبدیل می‌شود:

$$\min_{\mathbf{F}}\mathrm{Tr}[\boldsymbol{F}^T(\boldsymbol{I_c}-\bar{\boldsymbol{H}}Q^{-1}\bar{\boldsymbol{H}}^T)\boldsymbol{F}]\quad(۹)$$
$$s.t\quad \boldsymbol{F}\epsilon[0,1]^{n\times c}$$

که می‌توان با تجزیه ویژه ماتریس $\boldsymbol{I_c}-\bar{\boldsymbol{H}}Q^{-1}\bar{\boldsymbol{H}}^T$ را حل کرد. بردارهای ویژه $\boldsymbol{c}$ مربوط به کوچکترین مقادیر ویژه $\boldsymbol{c}$ ستون‌های $\boldsymbol{F}$هستند، جایی که $\boldsymbol{c}$ تعداد خوشه‌هاست.

$$(\boldsymbol{I_c}-\bar{\boldsymbol{H}}Q^{-1}\bar{\boldsymbol{H}}^T)\boldsymbol{v}=\boldsymbol{\mu v}\quad(۱۰)$$

فرض کنید $v_1,v_2,\cdots,v_c$ بردارهای ویژه نرمال شده مربوط به کوچکترین مقادیر ویژه $\boldsymbol{c}$ از رابطه (۱۰) باشد. راه‌حل رابطه (۹) بصورت (۱۱) خواهد بود:

$$\boldsymbol{F}=[v_1^T,v_2^T,\cdots,v_c^T]^T\quad(۱۱)$$

برای برآورده کردن محدودیت $\boldsymbol{F}\epsilon[0,1]^{n\times c}$، *winner-take-all* برای هر ردیف از $\boldsymbol{F}$ اعمال می‌شود. به طور خاص، تمام عناصر $\boldsymbol{F}$ روی صفر تنظیم می‌شوند به جز حداکثر عنصر در هر ردیف که یا مقدار یک تنظیم شده است.

۴- محاسبه معیار ارزیابی

معیارهای ارزیابی مورد استفاده در این پژوهش معیار صحت، نرخ صحیح مثبت است.

۴-۱- مجموعه‌داده

مجموعه‌داده مورد استفاده در این پژوهش مجموعه‌داده *NSL-KDD* است که جزئیات آن به شرح زیر است:

مجموعه‌داده *NSL-KDD* نسبت به سایر مجموعه داده‌ها در سیستم تشخیص نفوذ از رکوردهای واقعی‌تری برخوردار است. در واقع، *NSL-KDD* برخی مشکلات مجموعه‌داده اصلی *KDD99* مثل رکوردهای افزونه و تکراری در مجموعه آموزش و آزمون را که باعث انحراف طبقه‌بندی به سمت نمونه‌های تکراری می‌شود را رفع کرده است. مجموعه‌داده *NSL-KDD* شامل ۴۱ ویژگی و ۵ کلاس است که یک کلاس نرمال و ۴ نوع کلاس حمله در آن وجود دارد که کلاس‌های حمله عبارت هستند از: *Dos*، *U2R*، *R2L*، *Probing* در ادامه این چهار کلاس حمله معرفی می‌شوند.

حملات انکار سرویس[1]: در این حملات منابع سیستم بیش از حد مورد استفاده قرار می‌گیرد و باعث می‌شود که درخواست‌های نرمال برای در اختیارگرفتن منابع رد شود. مهاجمان برخی محاسبات را که وظیفه رسیدگی به درخواست‌های قانونی را دارند، از کاربران سلب می‌کنند. یعنی به عبارتی سرویس‌دهنده را از پاس‌گویی خارج می‌کنند.

جدول (۵): جزئیات مجموعه‌داده

کلاس *NSL-KDD*	تعداد داده آموزش	تعداد داده آزمون
نرمال	۶۷۳۴۳	۹۷۱۱
DoS	۴۵۹۲۷	۷۴۵۸
PRB	۱۱۶۵۶	۲۴۲۱
R2L	۹۹۵	۲۷۵۴
U2R	۵۲	۲۰۰

حملات کاربر به ریشه[2]: در این حملات مهاجم به حساب کاربری برروی سیستم دسترسی پیدا می‌کند و به سیستم آسیب می‌رساند. در این حمله مهاجم با استفاده از روش‌هایی مانند مهندسی اجتماعی به یک حساب کاربری عادی دسترسی پیدا می‌کند و سپس با استفاده از آسیب‌پذیری‌های موجود، امتیاز یک کاربر ارشد را به دست می‌آورد.

حملات حمله راه دور به نزدیک[3]: در این حملات مهاجم با نفوذ غیرمجاز از راه دور به ماشین قربانی شروع به سوءاستفاده از حساب قانونی کاربر کرده و اقدام به ارسال بسته برروی شبکه می‌کند. مهاجم با استفاده از روش آزمون و خطا و توسط اسکریپت‌های خودکار یا روش‌های دیگر اقدام به حدس‌زدن کلمه عبور قربانی می‌کند.

حملات حملات پویشی[4]: در این حملات شبکه و یا میزبان برای جمع‌آوری اطلاعات و یافتن آسیب‌پذیری‌های شناخته شده پویش می‌شود. یک حمله کاوش به عنوان نخستین گام یک

[1] DOS
[2] U2R
[3] R2L
[4] Probing

حمله واقعی به شبکه یا میزبان درنظر گرفته می‌شود. هرچند به وسیله این حملات هیچ آسیب مستقیمی به سیستم وارد نمی‌شود، اما این حملات باید جدی گرفته شود زیرا ممکن است مهاجم به وسیله این حملات پویشی اطلاعات مهمی برای انجام حملات بسیارخطرناک در سیستم به دست آورد.

۴-۲- معیار ارزیابی

به منظور ارزیابی روش پیشنهادی و مقایسه با سایر روش‌ها از دو معیار متداول استفاده شد که هر کدام به شرح زیر است:

معیار اول: صحت[1]، عبارت از میزان نزدیکی مقدار اندازه‌گیری شده به مقدار صحیح است؛ به‌عبارت‌دیگر از همخوانی یک مقدار آزمایشی یا میانگین چند سنجش با یک مقدار نظری (حقیقی) یک کمیت، صحت به دست می‌آید. دو مقدار TP و TN مهم‌ترین مقادیری هستند که در یک مسئله دو دسته‌ای باید بیشینه شوند:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (۱۲)$$

معیار دوم: معیار حساسیت[2] به عنوان (نرخ صحیح مثبت) در نظر گرفته می‌شود. معیار حساسیت ارزیابی می‌کند که چقدر مدل توانایی شناسایی سوابق غیرطبیعی را دارا می‌باشد. حساسیت بالاتر یک دسته‌بندی بهتر را نشان می‌دهد. این عبارت معمولا از نسبت نمونه‌های صحیح دسته‌بندی شده به کل نمونه‌ها حاصل می‌شود. میزان صحیح مثبت بیشتر یعنی میزان حساسیت بیشتر که منجر به دقت بیشتر هم خواهد شد. معادله حساسیت به شرح (۱۳) خواهد بود:

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

۴-۳- محیط شبیه‌سازی

روش پیشنهادی و الگوریتم گرگ خاکستری در نرم‌افزار متلب ۲۰۱۴ در سیستم عامل ویندوز ۱۱ - ۶۴ بیت با پردازنده ® *Intel Core ™ i7 2.8 GHz* با رَم ۱۶ *GB* پیاده‌سازی و اجرا گردیده است.

۴-۴- ارزیابی نتایج

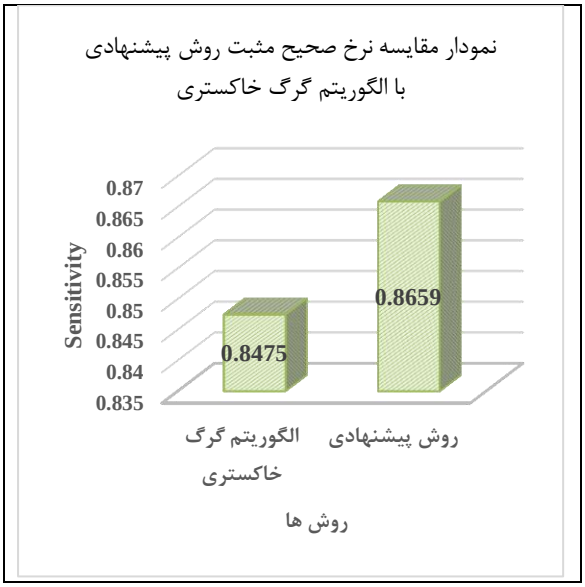
در این بخش به ارزیابی روش پیشنهادی به منظور بدست‌آوردن بیشتر میزان صحت، بر روی دادگان مورد نظر، پرداخته می شود. نتایج بدست‌آمده از روش‌های پیشنهادی و الگوریتم بهینه‌سازی گرگ خاکستری توسط معیار ارزیابی در جدول (۶) نمایش داده شده است.

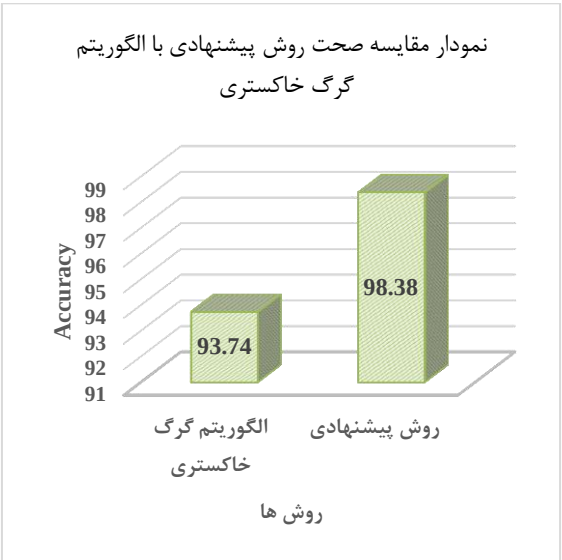
[1] Accuracy
[2] Sensitivity

جدول (۶): مقایسه صحت نتایج روش پیشنهادی و الگوریتم بهینه‌سازی گرگ خاکستری

روش پیشنهادی	الگوریتم گرگ خاکستری	معیار ارزیابی	مجموعه داده
۹۸/۳۸	۹۳/۷۴	Accuracy	*NSL-KDD*
۰/۸۶۵۹	۰/۸۴۷۵	Sensitivity	

نمودار ستونی مقایسه صحت، نرخ صحیح مثبت و نرخ صحیح منفی روش پیشنهادی با الگوریتم گرگ خاکستری بر روی مجموعه‌داده *NSL-KDD* در شکل (۳) نشان داده شده است.





شکل (۳): نمودار مقایسه صحت و نرخ صحیح مثبت روش پیشنهادی با الگوریتم گرگ خاکستری

عملکرد روش پیشنهادی با الگوریتم بهینه‌سازی گرگ خاکستری برای مسئله انتخاب ویژگی و دسته‌بندی بر روی مجموعه‌داده *NSL-KDD* مورد بررسی و مقایسه قرار گرفت. نتایج آزمایش‌ها در جدول (۶) نشان‌دهنده صحت ۹۸/۳۸ و نرخ صحیح‌مثبت ۰/۸۶۵۹ روش پیشنهادی در مقایسه با صحت ۹۳/۷۴ و نرخ صحیح‌مثبت ۰/۸۴۷۵ الگوریتم بهینه‌سازی گرگ خاکستری است. دلیل این برتری استفاده از ماشین یادگیری مفرط بدون نظارت و ساختار تعمیم‌پذیر این شبکه عصبی بدون نظارت است.

در خصوص مقادیر بدست آمده از معیارهای ارزیابی می‌توان اثبات نمود که پدیده بیش‌برازش رخ نداده است؛ علت این امر را می‌توان اینگونه توجیه کرد ابتدا اینکه بیش‌برازش به پدیده نامطلوبی در آمار گفته می‌شود که در آن درجه آزادی مدل بسیار بیشتر از درجه آزادی واقعی انتخاب شده و در نتیجه اگرچه مدل روی داده استفاده شده برای یادگیری بسیار خوب نتیجه می‌دهد، اما بر روی داده جدید دارای خطای زیاد است و اینکه از جمله دلایل اصلی ایجاد بیش‌برازش می‌توان به پیچیدگی بیش از حد مدل، بیش از حد بودن تعداد ویژگی‌های مجموعه‌داده و برابربودن ویژگی‌ها با نمونه‌ها، حجم بسیار کم مجموعه‌داده و وجود نویز اشاره کرد. رویکرد ارائه شده از پیچیدگی کمتری برخوردار بوده و به دلیل استفاده از ضریب تنظیم مشکل بیش‌برازش حل شده و با توجه شرح جزئیات مجموعه‌داده می‌توان دریافت که مجموعه‌داده از حجم کم و تعداد ویژگی برابر با نمونه‌های مجموعه‌داده برخوردار نبوده و از طرفی چون رویکرد انتخاب ویژگی استفاده شده است وجود نویز در دادگان هم بطور قطع رد می‌شود. دلیل این امر این است که روش پیشنهادی می‌تواند تعمیم‌پذیری بهتری برای مسئله انتخاب ویژگی، دسته‌بندی و تشخیص نفوذ تولید نماید و این امسئله سبب شده است تا در مسئله دسته‌بندی مکانیسم استفاده از الگوریتم ماشین یادگیری مفرط بدون نظارت برای انتخاب ویژگی و آموزش شبکه و روش رتبه‌بندی عملکرد بهتر و استحکام بیشتر نتایج روش پیشنهادی را نشان می‌دهد.

۴-۵- جمع‌بندی و کارهای آتی

نتایج آزمایش‌ها نشان‌دهنده صحت ۹۸/۳۸ و نرخ صحیح‌مثبت ۰/۸۶۵۹ و نرخ صحیح‌منفی ۰/۷۰۷۹ روش پیشنهادی در مقایسه با صحت ۹۳/۷۴ و نرخ صحیح‌مثبت ۰/۸۴۷۵ و نرخ صحیح‌منفی ۰/۶۲۶۷ الگوریتم بهینه‌سازی گرگ خاکستری است. دلیل این برتری استفاده از UELM در مسئله دسته‌بندی و تشخیص نفوذ و ساختار مستحکم و تعمیم‌پذیر این شبکه عصبی بدون نظارت است. در ارزیابی با مجموعه‌داده NSL-KDD برای دسته‌بندی، روش پیشنهادی عملکرد تعمیم پذیری بهتری در مقایسه با روش الگوریتم بهینه‌سازی گرگ خاکستری داشته و همچنین از نظر استحکام بهتر می‌باشد.به عنوان کارهای آتی پیشنهاد می گردد در صورت زیاد بودن تعداد ویژگی ها و ناهمگن بودن داده ها و لزوم شناسائی برخط، از روش های طبقه بندهای ترکیبی به همراه الگوریتم های بهینه سازی پارامترهای اولیه استفاده کرد و همچنین شبکه عصبی ماشین یادگیری مفرط بدون نظارت به دلیل تعیین تصادفی پارامترهای گره‌های لایه پنهان و وزن لایه ورودی ممکن است بعضی از پارامترهای غیر بهینه تولید کند که در عملکرد و ثبات شبکه تاثیرگذار است. بدین منظور استفاده از رویکردهایی نظیر آشوب و کوانتوم جهت مقداردهی اولیه این پارامترها می‌تواند یکی از راه‌های مقابله با این مشکل باشد.در نهایت، اعمال روش پیشنهادی بر روی مسائل علمی- مهندسی و واقعی برای بررسی هر چه بیشتر روش ارائه شده و همچنین بررسی تعمیم‌پذیری هر چه بیشتر روش ارائه شده در حل سایر مسائل می‌تواند چالش‌های پیش رو را نمایان و برطرف سازد.

۵- مراجع

[1] T. Saranya, S. Sridevi, C. Deisy, Tran Duc Chung, and MKA Ahamed Khan, "Performance analysis of machine learning algorithms in intrusion detection system: A review," Procedia Computer Science, vol. 171, pp. 1251-1260, 2020. DOI:10.1016/j.procs.2020.04.133.

[2] Lansky, Jan, Saqib Ali, Mokhtar Mohammadi, Mohammed Kamal Majeed, Sarkhel H. Taher Karim, Shima Rashidi, Mehdi Hosseinzadeh, and Amir Masoud Rahmani. "Deep learning-based intrusion detection systems: a systematic review." IEEE Access 9 (2021): 101574-101599, doi.org/10.1109/ACCESS.2021.3097247.

[3] Al-Daweri, Muataz Salam, Khairul Akram Zainol Ariffin, Salwani Abdullah, and Mohamad Firham Efendy Md. Senan. "An analysis of the KDD99 and UNSW-NB15 datasets for the intrusion detection system."Symmetry12,no.10(2020): 1666, doi.org/10.3390/sym12101666.

[4] Al-Tashi, Qasem, Helmi Md Rais, Said Jadid Abdulkadir, Seyedali Mirjalili, and Hitham Alhussian. "A review of grey wolf optimizer-based feature selection methods for classification." Evolutionary Machine Learning Techniques: Algorithms and Applications (2020): 273-286, DOI:10.1007/978-981-32-9990-0_13.

[5] Mirjalili, Seyedali, Ibrahim Aljarah, Majdi Mafarja, Ali Asghar Heidari, and Hossam Faris. "Grey wolf optimizer: theory, literature review, and application in computational fluid dynamics problems." Nature-inspired optimizers: Theories, literature reviews and applications (2020): 87-105, https://doi.org/10.1007/978-3-030-12127-3_6.

[6] Hancer, Emrah, Bing Xue, and Mengjie Zhang. "A survey on feature selection approaches for clustering." Artificial Intelligence Review 53 (2020): 4519-4545, https://doi.org/10.1007/s10462-019-09800-w.

[7] Gawlikowski, Jakob, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe et al. "A survey of uncertainty in deep neural networks." arXiv preprint arXiv:2107.03342 (2021), https://doi.org/10.48550/arXiv.2107.03342.

[8] Ding, Shifei, Xinzheng Xu, and Ru Nie. "Extreme learning machine and its applications." Neural Computing and Applications 25 (2014): 549-556, DOI:10.1007/s00521-013-1522-8.

[9] Huang, Guang-Bin, Qin-Yu Zhu, and Chee-Kheong Siew. "Extreme learning machine: theory and applications." Neurocomputing 70, no. 1-3 (2006): 489-501, https://doi.org/10.1016/j.neucom.2005.12.126

[10] Huang, Gao, Guang-Bin Huang, Shiji Song, and Keyou You. "Trends in extreme learning machines: A review." Neural Networks 61, pp. 32-48, 2015.

[23] Halim, Zahid, Muhammad Nadeem Yousaf, Muhammad Waqas, Muhammad Sulaiman, Ghulam Abbas, Masroor Hussain, Iftekhar Ahmad, and Muhammad Hanif. "An effective genetic algorithm-based feature selection method for intrusion detection systems." Computers & Security 110 (2021): 102448 , https://doi.org/10.1016/j.cose.2021.102448.
[24] Pradeep Mohan Kumar, K., M. Saravanan, M. Thenmozhi, and K. Vijayakumar. "Intrusion detection system based on GA-fuzzy classifier for detecting malicious attacks." Concurrency and Computation: Practice and Experience 33, no. 3 (2021): e5242 , https://doi.org/10.1002/cpe.5242.
[25] Ghosh, Partha, Dipankar Sarkar, Joy Sharma, and Santanu Phadikar. "An intrusion detection system using modified-firefly algorithm in cloud environment." International Journal of Digital Crime and Forensics (IJDCF) 13, no. 2 (2021): 77-93 , DOI:10.4018/IJDCF.2021030105.
[26] Alkahtani, Hasan, and Theyazn HH Aldhyani. "Intrusion detection system to advance internet of things infrastructure-based deep learning algorithms." Complexity 2021 (2021) , DOI:10.1155/2021/5579851.
[27] Chen, Jichao, Yijie Zeng, Yue Li, and Guang-Bin Huang. "Unsupervised feature selection based extreme learning machine for clustering." Neurocomputing 386 (2020): 198-207 , doi.org/10.1016/j.neucom.2019.12.065.
[28] Elsaid, Shaimaa Ahmed, and Nouf Saleh Albatati. "An optimized collaborative intrusion detection system for wireless sensor networks." Soft Computing 24, no. 16 (2020): 12553-12567 , https://doi.org/10.1007/s00500-020-04695-0.
[29] همایون، حامد، دهقانی، مهدی، اکبری، حمید، مروری تحلیل ترافیک شبکه گمنام‌ساز پارس با استفاده از یادگیری ماشین .پدافند غیرعامل،، صص ۱۷-۱، شماره ۱۲(۲)، ۱۴۰۰.
[30] H. M. Saleh, H. Marouane, and A. Fakhfakh, "Stochastic Gradient Descent Intrusions Detection for Wireless Sensor Network Attack Detection System Using Machine Learning," IEEE Access, vol. 12, no. December 2023, pp. 3825–3836, 2024, doi: 10.1109/ACCESS.2023.3349248.
[31] S. Subramani and M. Selvi, "Intrusion detection system using RBPSO and fuzzy neuro-genetic classification algorithms in wireless sensor networks," Int. J. Inf. Comput. Secur., vol. 20, no. 3–4, pp. 439–461, 2023, doi: 10.1504/IJICS.2023.128857.
[32] S. Subramani and M. Selvi, "Multi-objective PSO based feature selection for intrusion detection in IoT based wireless sensor networks," Optik (Stuttg)., vol. 273, no. December 2022, p. 170419, 2023, doi: 10.1016/j.ijleo.2022.170419.
[33] F. Al-Quayed, Z. Ahmad, and M. Humayun, "A Situation Based Predictive Approach for Cybersecurity Intrusion Detection and Prevention Using Machine Learning and Deep Learning Algorithms in Wireless Sensor Networks of Industry 4.0," IEEE Access, vol. 12, no. February, pp. 34800–34819, 2024, doi: 10.1109/ACCESS.2024.3372187.
[34] M. Aljebreen et al., "Binary Chimp Optimization Algorithm with ML Based Intrusion Detection for Secure IoT-Assisted Wireless Sensor Networks," Sensors, vol. 23, no. 8, 2023, doi: 10.3390/s23084073.
[35] M. H. Behiry and M. Aly, "Cyberattack detection in wireless sensor networks using a hybrid feature reduction technique with AI and machine learning methods," J. Big Data, vol. 11, no. 1, 2024, doi: 10.1186/s40537-023-00870-w.
[36] L. Dhanabal and S. P. Shantharajah, "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms," Int. J. Adv. Res. Comput. Commun. Eng., vol. 4, no. 6, pp. 446–452, 2015, doi: 10.17148/IJARCCE.2015.4696.
https://doi.org/10.1016/j.neunet.2014.10.001
[11] Wang, Jian, Siyuan Lu, Shui-Hua Wang, and Yu-Dong Zhang. "A review on extreme learning machine." Multimedia Tools and Applications 81, no. 29 (2022): 41611-41660 , https://doi.org/10.1007/s11042-021-11007-7
[12] Schilling, M., Paskarbeit, J., Hoinville, T., Hüffmeier, A., Schneider, A., Schmitz, J., Cruse, H. (Sept. 17 2013). A hexapod walker using a heterarchical structure for action selection. Frontiers in Computational Neuroscience, 7. doi:10.3389/fncom.2013.00126
[13] Maldonado, Javier, María Cristina Riff, and Bertrand Neveu. "A review of recent approaches on wrapper feature selection for intrusion detection." Expert Systems with Applications (2022): 116822 , https://doi.org/10.1016/j.eswa.2022.116822.
[14] Mhawi, Doaa
N., Ammar Aldallal, and Soukeana Hassan. "Advanced feature-selection-based hybrid ensemble learning algorithms for network intrusion detection systems." Symmetry 14, no. 7 (2022): 1461 , https://doi.org/10.3390/sym14071461.
[15] Kareem, Saif S., Reham R. Mostafa, Fatma A. Hashim, and Hazem M. El-Bakry. "An effective feature selection model using hybrid metaheuristic algorithms for iot intrusion detection." Sensors 22, no. 4 (2022): 1396 , https://doi.org/10.3390/s22041396.
[16]Naseri, Touraj Sattari, and Farhad Soleimanian Gharehchopogh. "A Feature Selection Based on the Farmland Fertility Algorithm for Improved Intrusion Detection Systems." Journal of Network and Systems Management 30, no. 3 (2022): 1-27 , DOI:10.3390/math10152675
[17] Mojtahedi, Amir, Farid Sorouri, Alireza Najafi Souha, Aidin Molazadeh, and Saeedeh Shafaei Mehr. "Feature Selection-based Intrusion Detection System Using Genetic Whale Optimization Algorithm and Sample-based Classification." arXiv preprint arXiv:2201.00584 (2022).
[18] Otair, Mohammed, Osama Talab Ibrahim, Laith Abualigah, Maryam Altalhi, and Putra Sumari. "An enhanced grey wolf optimizer based particle swarm optimizer for intrusion detection system in wireless sensor networks." Wireless Networks 28, no. 2 (2022): 721-744 , DOI:10.1007/s11276-021-02866-x.
[19] Safaldin, Mukaram, Mohammed Otair, and Laith Abualigah. "Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks." Journal of ambient intelligence and humanized computing 12, no. 2 (2021): 1559-1576 , DOI:10.1007/s12652-020-02228-z.
[20] Khanna, Ashish, Poonam Rani, Puneet Garg, Prakash Kumar Singh, and Aditya Khamparia. An Enhanced Crow Search Inspired Feature Selection Technique for Intrusion Detection Based Wireless Network System. Wireless Personal Communications (2021): 1-18 , DOI:10.1007/s11277-021-08766-9.
[21] Keserwani, Pankaj Kumar, Mahesh Chandra Govil, Emmanuel S. Pilli, and Prajjval Govil. "A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model." Journal of Reliable Intelligent Environments 7, no. 1 (2021): 3-21 , DOI: https://doi.org/10.58325/ijisct.003.01.0073.
[22] Kan, Xiu, Yixuan Fan, Zhijun Fang, Le Cao, Neal N. Xiong, Dan Yang, and Xuan Li. "A novel IoT network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network." Information Sciences 568 (2021): 147-162 , https://doi.org/10.1016/j.ins.2021.03.060.

پیوست

کدهای شبیه سازی

```
clc
clear
close all

[data, lbl] = PrepareData('KDDData.txt');
rmv_id = [];
for i = 1:size(data,2)
   if numel(unique(data(:,i)))==1
      rmv_id = [rmv_id i];
   end
end
data(:,rmv_id) = [];

data = mapminmax(data',0,1)';

[nD,nFea] = size(data);

% initialize parameters
nPop = 20;
lb = 0;
ub = 1;
Max_iter = 10;

%Create data test and train
cv = cvpartition(lbl,'holdout',0.2);

dataTrain = data(cv.training,:);
lblTrain = lbl(cv.training);

dataTest = data(cv.test,:);
CODE-PAPER
lblTest = lbl(cv.test);

% initialize alpha, beta, delta, gama, teta
Alpha.pos = zeros(1,nFea);
Alpha.score = -inf;

Beta.pos = zeros(1,nFea);
Beta.score = -inf;

Delta.pos = zeros(1,nFea);
Delta.score = -inf;

Gamma.pos = zeros(1,nFea);
Gamma.score = -inf;

Teta.pos = zeros(1,nFea);
Teta.score = -inf;

%Initialize the positions of search agents
indiv.pos = [];
indiv.score = [];
Pop = repmat(indiv,1,nPop);
for i=1:nPop
   while(1)
      sol = round(lb+(ub-lb)*rand(1,nFea));
      if sum(sol)>0
         break
      end
   end
   Pop(i).pos = sol;
end
```

کدهای شبیه سازی (ادامه)

```
      rnd = rand;
      if rnd<(1/5)
         Pop(i).pos(j) = X1(j);
      elseif rnd>=(1/5) && rnd<(2/5)
         Pop(i).pos(j) = X2(j);
      elseif rnd>=(2/5) && rnd<(3/5)
         Pop(i).pos(j) = X3(j);
      elseif rnd>=(3/5) && rnd<(4/5)
         Pop(i).pos(j) = X4(j);
      else
         Pop(i).pos(j) = X5(j);
      end

   end

   while(sum(Pop(i).pos)==0)
      sol = round(lb+(ub-lb)*rand(1,nFea));
      Pop(i).pos = sol;
      if sum(sol)>0
         break
      end
   end
 end
 iter = iter+1;
end

sol = Alpha.pos;
nf_sel = sum(sol==1);

dataTrain = dataTrain(:, sol==1);
dataTest = dataTest(:, sol==1);

SVMModel = fitcsvm(dataTrain,lblTrain);

prlbl_Tr = predict(SVMModel,dataTrain);
Cmat_train = confusionmat(lblTrain,prlbl_Tr);
TP = Cmat_train(1,1);
TN = Cmat_train(2,2);
FP = Cmat_train(2,1);
FN = Cmat_train(1,2);
Accuracy_train = (TP+TN)/(FP+FN+TP+TN);
DetectionRate_train = TP/(TP+FN);
FalseAlarm_train = FP/(TN+FP);


prlbl_te = predict(SVMModel,dataTest);
Cmat_test = confusionmat(lblTest,prlbl_te);
TP = Cmat_test(1,1);
TN = Cmat_test(2,2);
FP = Cmat_test(2,1);
FN = Cmat_test(1,2);
Accuracy_test = (TP+TN)/(FP+FN+TP+TN);
DetectionRate_test = TP/(TP+FN);
FalseAlarm_test = FP/(TN+FP);

clc
disp('Result of KDDData')
disp('=========================================
======')

Acc = mean(Accuracy_test)*100;
fprintf('Accuracy = %2.4f \n', Acc)
```

کدهای شبیه سازی

```
iter = 0; % Loop counter
% Main loop
while iter<Max_iter
   for i =1:nPop
      % Calculate objective function for each search
agent
      Pop(i).score =
CalFitness(Pop(i).pos,dataTrain,lblTrain);

      % Update Alpha, Beta, Delta, Gamma and Teta
      if  Pop(i).score>Alpha.score
         Alpha = Pop(i); % Update alpha
      end

      if  Pop(i).score<Alpha.score &&
Pop(i).score>Beta.score
         Beta = Pop(i); % Update beta
      end

      if  Pop(i).score<Alpha.score &&
Pop(i).score<Beta.score &&  Pop(i).score>Delta.score
         Delta = Pop(i); % Update delta
      end

      if  Pop(i).score<Alpha.score &&
Pop(i).score<Beta.score &&  Pop(i).score<Delta.score
&&  Pop(i).score>Gamma.score
         Gamma = Pop(i); % Update gamma
      end

      if  Pop(i).score<Alpha.score &&
Pop(i).score<Beta.score &&  Pop(i).score<Delta.score
&&  Pop(i).score<Gamma.score &&
Pop(i).score>Teta.score
         Teta = Pop(i); % Update teta
      end
   end


   a = 2-iter*((2)/Max_iter); % a decreases linearly fron
2 to 0

   % Update the Position of search agents including
omegas
   for i=1:nPop
      X1 = zeros(1,nFea);
      X2 = zeros(1,nFea);
      X3 = zeros(1,nFea);
      X4 = zeros(1,nFea);
      X5 = zeros(1,nFea);
      for j=1:nFea

         r1 = rand(); % r1 is a random number in [0,1]
         r2 = rand(); % r2 is a random number in [0,1]
         A1 = 2*a*r1-a;   %eq3
         C1 = 2*r2;       %eq4
         D_alpha = abs(C1*Alpha.pos(j)-Pop(i).pos(j));
%eq9
         cstep_alpha = 1/(1+exp(-10*(A1*D_alpha)));
         bstep_alpha = 0;
         if cstep_alpha>=rand
            bstep_alpha = 1;
         end
         if (Alpha.pos(j)+bstep_alpha)>=1
            X1(j) = 1;
         end
```

کدهای شبیه سازی (ادامه)

```
Sensitivity = mean(DetectionRate_test);
fprintf('Sensitivity = %2.4f \n', Sensitivity)

Specificity = mean(FalseAlarm_test);
fprintf('Specificity = %2.4f \n', Specificity)
```

CODE-PROPOSED

```
clc
clear
close all

load('KDDData.mat')

lambda1 = 0.01;           % trade-off parameter
lambda2 = 0.001;          % trade-off parameter
m = 30;                   % number of neurons
E = 0.001;
tmax = 10;                % number of iterations


nC = numel(unique(lbl));   % number of clusters
ub = 1;
lb = -ub;

grp = unique(lbl);
data = mapminmax(data', -1, 1)';  %Normalized data
[n, d] = size(data);

Q = 10^-2;  % param of the pair-wise similarity: W

dis = pdist2(data, data);
W = exp(-dis.^2./(2*Q^2));
W = W-diag(diag(W));
D = diag(sum(W));
L = D-W;

for nEpoch = 1:2

   % Initialize the cluster indicator matrix F by K-means
clustering on X
   predLbl = kmeans(data, nC);
   F = zeros(n, nC);
   for i = 1:nC
      F(predLbl==i, i) = 1;
   end

   % Generate input weights and biases of hidden neurons
randomly
   W = lb+(ub-lb)*rand(m,d);
   B = lb+(ub-lb)*rand(m,1);

   % Calculate the centered hidden layer output matrix H'
   H = calculate_H(B, W, data, n);
   Ht = (eye(n)-1/n*ones(n)*ones(n))*H;

   % Initialize B as an identity matrix.
   B = eye(m);

   % Compute Q
   Q = Ht'*Ht+lambda1*B+lambda2*Ht'*L*Ht;

   for t = 1:tmax


      % Update beta
      beta = Q^-1*Ht'*F;
```

کدهای شبیه سازی

```
        r1 = rand();
        r2 = rand();
        A2 = 2*a*r1-a;  %eq3
        C2 = 2*r2;      %eq4
        D_beta = abs(C2*Beta.pos(j)-Pop(i).pos(j));
%eq10
        cstep_beta = 1/(1+exp(-10*(A2*D_beta)));
        bstep_beta = 0;
        if cstep_beta>=rand
           bstep_beta = 1;
        end
        if (Beta.pos(j)+bstep_beta)>=1
           X2(j) = 1;
        end

        r1 = rand();
        r2 = rand();
        A3 = 2*a*r1-a;  %eq3
        C3 = 2*r2;      %eq4
        D_delta = abs(C3*Delta.pos(j)-Pop(i).pos(j));
%eq11
        cstep_delta = 1/(1+exp(-10*(A3*D_delta)));
        bstep_delta = 0;
        if cstep_delta>=rand
           bstep_delta = 1;
        end
        if (Delta.pos(j)+bstep_delta)>=1
           X3(j) = 1;
        end

        r1 = rand(); % r1 is a random number in [0,1]
        r2 = rand(); % r2 is a random number in [0,1]
        A4 = 2*a*r1-a;   %eq3
        C4 = 2*r2;       %eq4
        D_gamma = abs(C4*Gamma.pos(j)-
Pop(i).pos(j));  %eq9
        cstep_gamma = 1/(1+exp(-
10*(A4*D_gamma)));
        bstep_gamma = 0;
        if cstep_gamma>=rand
           bstep_gamma = 1;
        end
        if (Gamma.pos(j)+bstep_gamma)>=1
           X4(j) = 1;
        end

        r1 = rand(); % r1 is a random number in [0,1]
        r2 = rand(); % r2 is a random number in [0,1]
        A5 = 2*a*r1-a;   %eq3
        C5 = 2*r2;       %eq4
        D_teta = abs(C5*Gamma.pos(j)-Pop(i).pos(j));
%eq9
        cstep_teta = 1/(1+exp(-10*(A5*D_teta)));
        bstep_teta = 0;
        if cstep_teta>=rand
           bstep_teta = 1;
        end
        if (Teta.pos(j)+bstep_teta)>=1
           X5(j) = 1;
        end
```

کدهای شبیه سازی (ادامه)

```
    % foreach diagonal element of B do
    for i = 1: m
       B(i, i) = 1/sqrt(beta(i,:)*beta(i,:)'+E);
    end

    % Compute Q
    Q = Ht'*Ht+lambda1*B+lambda2*Ht'*L*Ht;


    % Update F by solving problem (14)
    tmp = eye(n)-Ht*Q^-1*Ht';
    [eigVec, eigVal]= eig(tmp);
    eigVal = real(eigVal);
    eigVec = real(eigVec);

    [eigVal_sor, idSort] = sort(diag(eigVal));
    V = eigVec(:, idSort(1:nC));
    [~, predLbl] = max(V, [], 2);
    F = zeros(n, nC);
    for i = 1:nC
       F(predLbl==i, i) = 1;
    end
  end

  Cmat = confusionmat(lbl,predLbl);
  TP = Cmat(1,1);
  TN = Cmat(2,2);
  FP = Cmat(2,1);
  FN = Cmat(1,2);

  Sensitivity(nEpoch) = TP/(TP+FN);
  Specificity(nEpoch) = FP/(TN+FP);
  Acc(nEpoch) = cluster_acc(lbl,predLbl);
end

clc
disp('Result of KDDData')
disp('=========================================
======')

Acc = mean(Acc);
fprintf('Accuracy = %2.4f \n', Acc)

Sensitivity = mean(Sensitivity);
fprintf('Sensitivity = %2.4f \n', Sensitivity)

Specificity = mean(Specificity);
fprintf('Specificity = %2.4f \n', Specificity)
```