Electronical & Cyber Defence
Vol. 13, No. 3, Autumn 2025, Serial No. 51

Computer Networks traffic classification model based on DBScan clustering and gamma classification

S. Z. Majidian[1], Sh. TaghipourEivazi[2*], B. Arasteh[3], A. Ghaffari[4]

*Assistant professor. Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

Received:2024 /12/08, Revised: 2025/02/19, Accepted: 2025/03/13, Published: 2025/04/21

DOR:

ABSTRACT

Traffic classification is a crucial network monitoring process with wide applications in security, quality of service, and network management. With the increasing complexity and variety of network traffic, new challenges arise, including the lack of labeled training data. To address this challenge, this paper presents a traffic classification mechanism that combines unsupervised and semi-supervised machine learning algorithms. This mechanism uses a limited set of labeled training data to improve classification accuracy. The proposed method represents each traffic flow as a feature vector containing the statistical characteristics of that flow. The number of features generated for each sample is reduced using principal component analysis. DBScan clustering is employed to determine the correct traffic type for each untagged traffic stream. Finally, the gamma classifier model is used to separate the new traffic flows. The efficiency of the proposed method has been evaluated using real data sets. The results show that the proposed method can classify traffic flows with an average accuracy of 95.12%, representing at least a 7.03% improvement over previous approaches.

Keywords: Traffic classification, machine learning, DBScan clustering, gamma classification

. .

Cite this article: S. Z. Majidian, S. TaghipourEivazi, B. Arasteh, and A. Ghaffari, "Computer Networks traffic classification model based on DBScan clustering and gamma classification" Electronic and Cyber Defense, vol. 13(3), pp. 1-21, 2025.

http//doi.org/0000000000000000000

Publisher: Imam Hossein University

Corresponding Author Email: shiva.taghipour.e@gmail.com

"پدافند الکترونیکی و سایبری"
سال ۱۳، شماره ۳، پاییز ۱۴۰۴، ص۲۱-۱
شاپا الکترونیکی: ۸۹۷۹-۲۹۸۰ شاپا چاپی: ۴۳۴۷-۲۳۲۲

علمی-پژوهشی

تشخیص حملات در زیرساخت اینترنت اشیاء با استفاده از الگوریتم بهبودیافته شامپانزه و یادگیری عمیق

سیده زهره مجیدیان [۱]، شیوا تقی‌پور عیوضی*[۲]، بهمن آراسته[۳]، علی غفاری [۴]

۱- گروه مهندسی کامپیوتر، واحد بین المللی ارس، دانشگاه آزاد اسلامی، تبریز، ایران
۲-استادیار۳-دانشیار۴-دانشیار، گروه مهندسی کامپیوتر، واحد تبریز، دانشگاه آزاد اسلامی، تبریز، ایران
(دریافت: ۱۴۰۴/۰۴/۱۷، بازنگری: ۱۴۰۴/۰۵/۱۸، پذیرش: ۱۴۰۴/۰۶/۲۷، انتشار: ۱۴۰۴/۰۷/۱۶)

چکیده

افزایش تعداد دستگاه‌های اینترنت اشیا، سرعت بالا و حجم زیاد اطلاعات تولیدشده، باعث شده است که مسئله امنیت شبکه‌های اینترنت اشیا و شناسایی حملات سایبری در این شبکه‌ها، به یکی از چالش‌های مهم در این حوزه تبدیل شود. سیستم‌های تشخیص نفوذ به‌عنوان یکی از راهکارهای ارائه‌شده برای مقابله با این مشکل است. انتخاب صحیح ویژگی‌ها در ایجاد مدل‌های تشخیص نفوذ می‌تواند باعث افزایش چشمگیری در دقت تشخیص شود. در این مقاله الگوریتم دودویی و بهبودیافته شامپانزه‌ها برای انتخاب ویژگی طراحی‌شده است. الگوریتم شامپانزه برای حل مسائل پیوسته است و در حل مسائل دودویی نمی‌تواند کارآمد باشد. همچنین دارای مشکل افتادن در دام محلی است و اکتشاف و بهره‌وری و همگرایی در این الگوریتم کند است. بنابراین نیاز است تغییراتی در این الگوریتم برای حل مسائل دودویی انجام شود. ازاین‌رو در این مقاله یک نسخه بهبودیافته شامپانزه برای مسائل گسسته و انتخاب ویژگی و رفع موارد ذکرشده، در تشخیص نفوذ و حملات مبتنی بر شبکه‌های اینترنت اشیا طراحی و پیاده‌سازی شده است. روش پیشنهادی به‌طور میانگین ۶۰ درصد ویژگی‌ها را کاهش داده و به ترتیب با دقت‌های۹۹/۳، ۹۹/۶ و۹۹/۹درصد در مجموعه داده‌های ToN-IoT، UNSW-NB15 و IoTID20، موفق به تشخیص حملات شده است و به‌طور چشمگیری باعث کاهش نرخ هشدار کاذب حملات شده است. تحلیل آماری آزمون کراس کال واریس نشان داد که روش پیشنهادی نسبت به روش‌های مورد مقایسه با سرعت بیشتری همگرا می‌شود.

کلیدواژه‌ها: حملات، اینترنت اشیاء، انتخاب ویژگی، یادگیری عمیق

۱. مقدمه

امروزه اینترنت اشیاء به‌عنوان یک فناوری جدید و محبوب با دستگاه‌های کوچک و باقابلیت اتصال، به‌سرعت درحال‌توسعه و گسترش است و از پتانسیل بالایی در زمینه‌های مختلف برخوردار است. در یک شبکه اینترنت اشیا دستگاه‌ها از طریق اینترنت و به‌صورت بی‌سیم با یکدیگر ارتباط برقرار می‌کنند که به هر یک از این دستگاه‌های فیزیکی اشیاء می‌گویند. افزایش دستگاه‌های اینترنت اشیاء، وجود دستگاه‌های ناهمگن در کاربردهای مختلف، سرعت قابل‌توجه اینترنت و حجم بالای اطلاعات تولیدشده در شبکه، باعث شده است که مسئله حفاظت از شبکه‌های اینترنت اشیا و شناسایی انواع حملات سایبری در این شبکه‌ها به امر چالش‌برانگیز تبدیل شود[۱].

برای حل مشکل امنیت شبکه‌های اینترنت اشیاء محققان سیستم‌های تشخیص نفوذ را پیشنهاد دادند. سیستم‌های تشخیص نفوذ معمولاً به دو دسته کلی سیستم تشخیص نفوذ مبتنی بر امضاء [۱] و مبتنی بر ناهنجاری [۲] تقسیم کرد. در سیستم‌های مبتنی بر امضاء، رفتار و فعالیت‌های مخرب از قبل در پایگاه داده ذخیره‌شده است و با نظارت بر شبکه فعالیت‌های

[1] Signature-based
[2] anomaly-based

استناد: مجیدیان، سیده زهره، تقی پور عیوضی ، شیوا ، آراسته،بهمن و غفاری ، علی" تشخیص حملات در زیرساخت اینترنت اشیاء با استفاده از الگوریتم بهبودیافته شامپانزه....."، پدافند الکترونیک و سایبری، ۱۴ (۳)، ۲۱-۱، ۱۴۰۴.

http//doi.org/0000000000000000000000000

ناشر: دانشگاه جام امام حسین(ع).

* رایانامه نویسنده مسئول: shiva.taghipour.e@gmail.com

مشابه به رفتارهای ذخیره‌شده را به‌عنوان رفتارهای مخرب شناسایی می‌کند. در سیستم‌های مبتنی بر ناهنجاری، یک فعالیت که متفاوت‌تر از رفتار عادی سیستم در حالت عادی باشد، به‌عنوان مخرب تشخیص داده می‌شود[۲]. روش‌های تشخیص نفوذ مبتنی بر ناهنجاری و مبتنی بر قواعد سنتی، با توجه به پیچیدگی حملات، ماهیت پویا و ناهمگن شبکه‌های اینترنت اشیا ناکارآمد هستند. از طرف دیگر روش‌های سیستم تشخیص نفوذ مبتنی بر ناهنجاری به‌طور گسترده با توجه پیشرفت‌های یادگیری ماشین و عمیق در تشخیص نفوذ و انواع حملات سایبری در شبکه‌های اینترنت اشیا مورداستفاده قرار می‌گیرند. سیستم تشخیص نفوذ مبتنی بر ناهنجاری سرعت بالاتری در تشخیص حملات سایبری دارند و همچنین در تشخیص و شناسایی فعالیت‌های مخرب جدید کارآمدتر هستند[۳].

روش‌های یادگیری ماشین به‌عنوان یک روش مؤثر در زمینه‌های مختلف اینترنت اشیا و به‌خصوص امنیت شبکه‌های اینترنت اشیا نقش مهمی ایفا کرده‌اند. ولی مهاجمان و هکرها نیز می‌توانند برای انجام طراحی حملات پیچیده‌تر از همین روش‌های یادگیری استفاده کنند[۴]. بااین‌حال، به دلیل انتخاب نادرست و نامناسب ویژگی در برخی از مدل‌های یادگیری ماشین، طبقه‌بندی اشتباه، امری اجتناب‌ناپذیر است[۵]. بنابراین روش‌های یادگیری ماشین بیشتر مستعد خطا هستند و هزینه محاسباتی، به نسبت دقتی که ارائه می‌دهند بیشتر است. برای رفع این کاستی‌ها، سیستم‌های تشخیص مبتنی بر ناهنجاری با استفاده از روش‌های یادگیری عمیق می‌تواند فعالیت‌های مخرب و غیرعادی را با دقت بالاتری شناسایی کند. محققان بسیاری برای طراحی سیستم‌های تشخیص نفوذ با دقت بالا از الگوریتم‌های یادگیری عمیق استفاده کرده‌اند[۱, ۶-۸]. یک سیستم تشخیص نفوذ مبتنی بر یادگیری عمیق می‌تواند به‌طور خودکار ویژگی‌های پیچیده ترافیک شبکه را در سطح پایین و بالا استخراج کند و رفتارهای غیرعادی را با دقت و سرعت‌بالا شناسایی کند. همین امر موجب شده است که سیستم‌های تشخیص نفوذ مبتنی بر یادگیری عمیق، دقت و عملکرد بالاتری نسبت به روش‌های یادگیری ماشین داشته باشند[۹].

طبقه‌بندی یکی از تکنیک‌های به‌شدت استفاده‌شده درزمینهٔ داده‌کاوی و یادگیری ماشین است که نیازمند مجموعه‌ای از ویژگی‌ها برای فرآیند یادگیری است[۱۰]. بااین‌وجود، بهبود توانایی یادگیری الگوریتم‌های طبقه‌بندی، به‌ویژه برای مجموعه داده‌هایی که حاوی تعداد زیادی ویژگی هستند، کار پیچیده‌ای است[۱۱]. این امر منجر به طولانی شدن فرآیند طبقه‌بندی می‌شود و زمان بیشتری برای یادگیری هر ویژگی از داده‌های آموزشی نیاز است. ویژگی‌های اضافی و بی‌ربط در داده‌ها، پیچیدگی عملکرد الگوریتم‌های یادگیری را افزایش می‌دهند و درنتیجه زمان محاسبه را افزایش می‌دهند[۱]. این ویژگی‌های بی‌ربط بهتر است از مجموعه داده حذف شوند تا فرآیند یادگیری مؤثری حاصل شود. بنابراین، تکنیک های انتخاب ویژگی بهینه و وزن دهی ویژگی‌ها برای انتخاب ویژگی‌های مرتبط و بهبود دقت طبقه‌بندی موردنیاز است. این فرآیندها بُعد داده‌ها را کاهش می‌دهند و متقابلاً زمان یادگیری نیز کاهش می‌یابد[۳].

انتخاب ویژگی یک مسئله جستجوی ترکیبی است که ویژگی‌های اضافی و بی‌ربط را حذف می‌کند و ویژگی‌های مرتبط با مجموعه داده‌ها را حفظ می‌کند. بنابراین، فرایند انتخاب ویژگی‌ها تعداد ویژگی‌ها در مجموعه داده‌ها را کاهش می‌دهد و با کاهش پیچیدگی محاسباتی، فرایند یادگیری را سریع تر می‌کند. این کاهش در تعداد ویژگی‌ها، مجموعه داده‌ها را ساده‌تر و قابل مدیریت‌تر برای فرایند طبقه‌بندی می‌سازد.

تکنیک‌های انتخاب ویژگی به دو روش فیلتر[1] و بسته[2] طبقه‌بندی‌شده‌اند. از روش فیلتر برای فیلتر کردن ویژگی‌های ناچیز که گزینه‌های کمتری در تجزیه‌وتحلیل داده‌ها دارند، استفاده می‌شود. روش‌های فیلتر یک زیرمجموعه با تعداد زیادی ویژگی یا حتی انتخاب همه ویژگی ها در مجموعه داده را انتخاب می کنند. بنابراین، یک آستانه مناسب برای انتخاب زیرمجموعه ضروری است. ویژگی های انتخاب‌شده از روش فیلتر بر اساس خصوصیات داده نظیر اندازه‌های اطلاعاتی، همبستگی، سازگاری و فاصله در فضای ویژگی تحلیل می‌شوند. روش بسته‌بندی از دقت پیش‌بینی یک الگوریتم یادگیری از پیش تعیین‌شده برای تعیین کیفیت ویژگی‌های انتخابی استفاده می‌کند. الگوریتم‌های فرا ابتکاری به‌عنوان یکی از تکنیک‌های انتخاب ویژگی مبتنی بر بسته‌بندی هستند. این الگوریتم‌ها با توجه به عملگرهای تصادفی از اکتشاف و بهره‌وری خوبی برخوردار هستند و زمینه‌های مختلفی کاربرد دارند[۱۲-۱۴]. روش‌های مبتنی بر بسته‌بندی بیشتر برای وزن دهی و انتخاب ویژگی به کاربرده می‌شوند. یک الگوریتم بهینه‌ساز فرا ابتکار همیشه روی اکتشاف و بهره‌برداری از یک فضای جستجو تمرکز می‌کند تا تعادل خوبی بین آن‌ها حفظ شود. در فاز اکتشاف یک الگوریتم فرا ابتکاری، بهترین مکان‌ها در فضای جستجو کاوش می‌شوند. در فاز بهره‌برداری به دنبال راه‌حل‌های بهینه در بهترین مکان‌ها می‌گردد. الگوریتم‌های بهینه سازی بسیاری بر اساس فرآیند اکتشاف و بهره برداری وجود دارند. هر الگوریتم بهینه‌سازی طبیعت و پیچیدگی خاص خود را دارد که آن را در مسئله بهینه سازی خاصی مؤثر می سازد و

[1] filter
[2] wrapper

ممکن است در سایر مسائل بهینه‌سازی دیگر ناکارآمد باشد. بنابراین، همیشه نیاز به الگوریتم‌های بهینه‌سازی جدید وجود دارد.

این امر تحلیلگران را بر آن می‌دارد تا الگوریتم‌های بهینه‌سازی جدید و مؤثرتری را برای حل مشکلات خاص در زمینه های مختلف توسعه دهند. همچنین، هر الگوریتم فرا ابتکاری عملکرد مشابهی در تمام مسائل بهینه‌سازی دارد. بنابراین، مسائل حل‌نشده در الگوریتم‌های موجود می‌توانند با معرفی الگوریتم‌های فرا ابتکاری جدید حل شوند.

در این مقاله الگوریتم دودویی و بهبودیافته شامپانزه‌ها برای انتخاب ویژگی طراحی‌شده است. الگوریتم شامپانزه برای حل مسائل پیوسته طراحی‌شده است و در حل مسائل دودویی نمی‌تواند کارآمد باشد. همچنین دارای مشکل افتادن در دام محلی است و اکتشاف و بهره‌وری و همگرایی در این الگوریتم کند است. بنابراین نیاز است تغییراتی در این الگوریتم برای حل مسائل دودویی انجام شود. ازاین‌رو یک نسخه بهبودیافته الگوریتم دودویی شامپانزه برای مسئله انتخاب ویژگی و رفع موارد ذکرشده، در تشخیص نفوذ و حملات مبتنی بر شبکه‌های اینترنت اشیا در این مقاله طراحی و پیاده‌سازی شده است که اختصاراً BCHO[1] نام‌گذاری شده است. درروش پیشنهادی توابعی برای گذر از احتمال گیر افتادن در دام محلی ارائه می‌شود. BCHO برای کاهش مشکل سرعت همگرایی کند و گرفتار شدن در بهینه‌های محلی طراحی‌شده است.

نتایج ازنظر سرعت همگرایی، احتمال گیر افتادن در حداقل‌های محلی، و اکتشاف با چندین الگوریتم فرا ابتکاری جدید و مرسوم مقایسه شده است. برای ارزیابی مدل، از شبکه‌های عصبی کانولوشنی[2] استفاده‌شده است.

مدل پیشنهادیBCHO در این پژوهش دارای نوآوری‌های زیر است:

۱-گسسته سازی الگوریتم فرا ابتکاری شامپانزه‌ها
۲-کاهش مشکل همگرایی کند
۳-کاهش گرفتار شدن در بهینه‌های محلی
۴- افزایش دقت تشخیص و کاهش نرخ هشدار کاذب حملات سایبری

در بخش دوم این مقاله، به بررسی و مرور رویکردهای قبلی برای انتخاب ویژگی از الگوریتم‌های فرا ابتکاری پرداخته خواهد شد. در بخش سوم، روش پیشنهادی و عملکرد آن توضیح داده خواهد شد. بخش چهارم به توضیح مجموعه داده‌های استفاده‌شده، فرآیند پیش‌پردازش داده‌ها و مدل یادگیری عمیق مورداستفاده می‌پردازد. سپس، معیارهای ارزیابی ارائه‌شده و روش پیشنهادی با چندین الگوریتم فرا ابتکاری قدرتمند مقایسه خواهد شد. در پایان، یک خلاصه از یافته‌ها و پیشنهادهای برای کارهای آینده در این حوزه ارائه خواهد شد.

۲. پیشینه تحقیق

تحقیقات زیادی با هدف شناسایی حملات به زیرساخت‌های اینترنت اشیا، به‌خصوص حملات مبتنی بر بدافزار، با استفاده از تحلیل ترافیک اینترنت اشیا انجام‌شده است [۱۵, ۱۶]. در این تحقیقات چندین چالش همواره موردبحث بوده است. طراحی الگوریتم‌های یادگیری با دقت تشخیص بالا و طراحی الگوریتم‌های انتخاب ویژگی برای افزایش زمان تشخیص و دقت بیشتر، از پرچالش‌ترین موضوعات بحثی بوده است. یکی از روش‌های بسیار مؤثر برای حل مشکل انتخاب ویژگی، استفاده از روش‌های بهینه‌سازی فرا ابتکاری هست. تحقیقات اخیر نشان داده است که استفاده از الگوریتم‌های بهینه‌سازی فرا ابتکاری برای کاهش ابعاد و ویژگی‌های منحصربه‌فرد اثربخشی بیشتری دارد[۱۷]. در طی سال‌های اخیر، تحقیقات متعددی در زمینه سیستم‌های تشخیص نفوذ و حمله با استفاده از انتخاب ویژگی و الگوریتم‌های فرا ابتکاری انجام‌شده است. در این بخش به بررسی کارهای اخیر که از الگوریتم‌های فرا ابتکاری برای تشخیص نفوذ استفاده کرده‌اند، پرداخته‌شده است.

در مطالعه[۱۸]، یک الگوریتم انتخاب ویژگی برای سیستم تشخیص نفوذ با استفاده از بهینه‌سازی الهام گرفته از کبوتر دودویی معرفی‌شده است. الگوریتم بهینه‌ساز دودویی کبوتر با استفاده از سه دیتاست عمومی، شامل KDDCUP99، NLS-KDD و UNSW-NB15، ارزیابی شد. این الگوریتم با کاهش تعداد ویژگی‌های مورد نیاز برای ساخت یک سیستم تشخیص نفوذ قوی کمک کرد. نرخ تشخیص بالا و دقت با هشدارهای کاذب پایین را حفظ کرد.

در مطالعه [۱۹]، از الگوریتم بهینه‌سازی کلونی مورچه‌ها برای تشخیص نفوذ استفاده‌شده است. این روش برای انتخاب ویژگی‌های ارزشمندتر به‌منظور افزایش کارایی سیستم‌های تشخیص نفوذ استفاده شد. شبیه‌سازی‌ها روی دو مجموعه داده KDDCUP99و NSL-KDD نشان دادند که سرعت فرآیند تشخیص نفوذ با تمرکز بر ویژگی‌های مهم‌تر و کاربردی‌تر بدون تأثیر قابل‌توجهی بر دقت آن افزایش یافت.

همچنین، در مطالعه [۲۰]، یک رویکرد جدید چند هدفه مبتنی بر کلونی زنبور برای انتخاب ویژگی در سیستم‌های تشخیص نفوذ ارائه شد. هدف از این مطالعه نه‌تنها ارائه یک روش جدید برای انتخاب ویژگی بود، بلکه معرفی یک تابع مناسب برای دستیابی به اهداف انتخاب ویژگی، یعنی کاهش تعداد ویژگی‌های

[1] Binary Chimp Optimization algorithm(BCHO)
[2] Convolution Nural Networks(CNN)

انتخاب‌شده، کاهش نرخ هشدارهای کاذب، کاهش نرخ خطاهای طبقه‌بندی، و بهینه‌سازی دقت طبقه‌بندی در مقایسه با مواردی که تمام ویژگی‌ها استفاده می‌شوند، بود.

علاوه بر این، در[۲۱]، خلاصه‌ای از الگوریتم‌های دسته‌بندی سیستم تشخیص نفوذ بر اساس الگوریتم‌های یادگیری ماشین معتبر ارائه شد. در این مقاله، تکنیک‌های گروهی و هیبریدی بر اساس روش‌های گروهی همگن و ناهمگن موردبررسی قرارگرفته‌اند. همچنین، الگوریتم‌های بهینه‌سازی دسته ذرات، کلونی مورچگان و کلونی زنبور مصنوعی برای انتخاب ویژگی‌های ضروری در مجموعه داده NSL-KDD به‌منظور بهبود عملکرد سیستم تشخیص نفوذها به کار گرفته شدند. برای ارزیابی فرآیند انتخاب ویژگی، طبقه‌بندهای ماشین بردار پشتیبان و K نزدیک‌ترین همسایه استفاده شدند. مقایسه عملکرد نشان داد که زمان اجرا به‌طور قابل‌توجهی کاهش‌یافته، دقت و نرخ تشخیص افزایش‌یافته و نرخ هشدار کاذب کاهش‌یافته است. بین سه الگوریتم فرا ابتکاری، عملکرد کلونی زنبور بهتر از دو روش دیگر بود.

همچنین، در مطالعه [۲۲] از الگوریتم قطره‌های آب هوشمند برای افزایش دقت طبقه‌بندی ماشین بردار پشتیبان در سیستم تشخیص نفوذ استفاده شد. این اقدام با کاهش تعداد ویژگی‌های دیتاست KDDCUP 99 با استفاده از الگوریتم قطره‌های آب هوشمند انجام شد و نتایج نشان داد که ترکیب ماشین بردار پشتیبان و الگوریتم قطره‌های آب هوشمند باعث بهبود قابل‌توجهی در مقایسه با الگوریتم‌های ماشین بردار پشتیبان، بیزین و ترکیب الگوریتم ژنتیک و ماشین بردار پشتیبان شد.

در مطالعه [۲۳]، از الگوریتم کرم شب‌تاب برای انتخاب ویژگی در سیستم تشخیص نفوذ استفاده شد. از مجموعه ۴۱ ویژگی موجود در دیتاست KDDCUP99، ده ویژگی با استفاده از روش پیشنهادی استخراج شدند که بیشترین تأثیر را در تشخیص نفوذ داشتند. با کاهش اطلاعات موردنیاز، زمان اجرا کاهش یافت، ساختار ساده‌تر شد و عملکرد طبقه‌بندی بهبود یافت. نتایج نشان داد که، در مقایسه با استفاده از تمام ویژگی‌ها، عملکرد تشخیص بهبودیافته و هزینه محاسبات کاهش یافت. متأسفانه، در این تحقیق، خروجی الگوریتم کرم شب‌تاب فقط با طبقه‌بندهای C4.5 و شبکه بیزین مورد مقایسه قرار گرفت و هیچ مقایسه‌ای با عملکرد سایر الگوریتم‌های فرا ابتکاری انجام نشد.

در مطالعه [۲۴]، از الگوریتم گرگ خاکستری دودویی برای انتخاب ویژگی در دیتاست NSL-KDD استفاده شد و نتایج شبیه‌سازی نشان داد که این الگوریتم توانایی ایجاد تعادل مناسب بین افزایش دقت و نرخ تشخیص و کاهش تعداد ویژگی‌ها را دارد. روش پیشنهادی با استفاده از الگوریتم بهینه‌سازی ذرات و مورچگان مورد مقایسه قرار گرفت و نتایج نشان داد که روش پیشنهادی توانایی ایجاد تعادل مناسب بین پارامترهای موردنظر را دارد.

در سال ۲۰۲۳، یک الگوریتم بهبودیافته برای انتخاب ویژگی در سیستم تشخیص نفوذ به شبکه اینترنت اشیا با نام LS-PIO معرفی شد [۲۵]. این الگوریتم الهام گرفته از لانه یابی کبوترها بوده و با استفاده از یک روش جستجوی محلی بهبودیافته است. همچنین، برای بهبود عملکرد سیستم، از یک رویکرد یادگیری گروهی استفاده‌شده است که بر اساس چندین طبقه‌بندی کننده یک کلاسه عمل می‌کند. این الگوریتم بر روی چهار مجموعه داده معتبر BoT-IoT، UNSW-NB15، NLS-KDD و KDD99 ارزیابی شد و نتایج نشان داد که تعداد ویژگی‌های انتخاب‌شده برای هر یک از مجموعه داده‌ها بهبودیافته است. به‌عنوان مثال، تعداد ویژگی‌ها برای UNSW-NB15، KDD99 و NSL-KKD به ترتیب از ۴۹ به ۸، از ۴۲ به ۱۲ و از ۴۲ به ۱۰ ویژگی کاهش‌یافته است. همچنین، برای مجموعه داده BoT-IoT تنها سه ویژگی انتخاب‌شده است.

بسیاری از تحقیقات گذشته درزمینه ی یادگیری ماشین به تشخیص نفوذ در شبکه‌های کامپیوتری تمرکز داشته‌اند. یک سیستم تشخیص نفوذ با استفاده از طبقه‌بندی کننده‌های ترکیبی و الگوریتم‌های فرا ابتکاری برای بهینه‌سازی و انتخاب ویژگی با استفاده از الگوریتم ژنتیک کانولوشن در سال ۲۰۲۲ معرفی‌شده است[۲۶]. این مقاله یک روش جدید برای انتخاب ویژگی با استفاده از یک الگوریتم ژنتیک ارائه می‌دهد که زیرمجموعه‌های بهینه ویژگی را از مجموعه داده NSL-KDD تعیین می‌کند. همچنین، طبقه‌بندی ترکیبی با استفاده از رگرسیون لجستیک و درخت تصمیم‌گیری برای بهبود نرخ تشخیص و دقت استفاده می‌شود. نتایج تجربی نشان می‌دهد که الگوریتم بهینه‌سازی گرگ خاکستری با دقت ۹۹/۴۴ درصد و نرخ تشخیص ۹۹/۳۶ درصد و کاهش تعداد ویژگی‌ها از ۴۱ به ۲۱ بهترین عملکرد را ارائه می‌دهد.

یک روش یادگیری عمیق ترکیبی مبتنی بر بهینه‌سازی نهنگ چسبنده برای تشخیص نفوذ به شبکه با استفاده از ویژگی‌های شبکه عصبی کانولوشن در سال ۲۰۲۰ معرفی شد [۲۷]. مدل پیشنهادی بر روی مجموعه داده NSL KDD آزمایش شد و با استفاده از معیارهای ارزیابی مختلف مانند دقت، صحت، فراخوانی و معیار F1 به ترتیب با مقادیر ۸/ ۹۳،۹۲، ۹۳/۲ و ۹۲/۶ به عملکرد بهتری دست‌یافت.

یک سیستم تشخیص ناهنجاری و نفوذ در شبکه اینترنت اشیا مبتنی بر ترکیب الگوریتم گرگ خاکستری و الگوریتم

ازدحام ذرات به نام GWO_PSO در سال ۲۰۲۱ ارائه‌شده است[۲۸]. از ترکیبی از GWO و PSO برای استخراج ویژگی‌های مهم شبکه اینترنت اشیا استفاده شد و نتایج نشان داد که مدل پیشنهادی با دقت ۹۹/۶۶در مجموعه داده KDDCup99، ۹۹/۲۴ در NSL-KDD و ۹۹/۸۸ در CIC-2017 بهتر عمل کرده است. این مقایسه نشان می‌دهد که روش پیشنهادی بهتر از روش‌های LSTM-RNN، ANN، GRU-RNN و BRNN عمل کرده است.

بررسی کارهای قبلی مرتبط با سیستم تشخیص نفوذ نشان می‌دهد که حجم بالای داده‌ها باعث کاهش دقت و سرعت الگوریتم‌های دسته‌بندی در سیستم تشخیص نفوذ شده است. پژوهشگران از روش‌های فرا ابتکاری برای کاهش ویژگی‌ها در مجموعه داده‌های سیستم تشخیص نفوذ استفاده کرده‌اند. از طرفی، یادگیری عمیق نسبت به یادگیری ماشین از عملکرد بهتری برخوردار است. انتخاب ویژگی‌های ارزشمند و استفاده از مدل عمیق نقش مهمی در تشخیص نفوذ، بازی می‌کند[۱۱]. ازاین‌رو در این مقاله، یک روش مؤثر برای انتخاب ویژگی‌ها باهدف رفع دام محلی و استفاده از الگوریتم ژنتیک بر ای افزایش سرعت و افزایش اکتشاف و بهره‌وری و کاهش زمان جستجو ارائه‌شده است که به‌طور مؤثری ویژگی‌ها را کاهش داده و دقت را افزایش می‌دهد. جدول (۱) نتایج برخی الگوریتم‌های فراابتکاری که برای انتخاب ویژگی در تشخیص نفوذ استفاده‌شده اند را نشان می‌دهد.

جدول(۱): مروری بر کارهای پیشین انتخاب ویژگی با استفاده از الگوریتم‌های فراابتکاری

مرجع	سال	مدل	انتخاب ویژگی	نوع طبقه‌بندی	مجموعه داده	معیارها
[۲۹]	۲۰۲۰	SVM	double Particle swarm optimization(PSO)	Binary Multi	NSL-KDD, CICIDS2017	CICIDS2017: DR = 99.92%, FAR = 0.1%), NSL-KDD :DR = 99.81%,FAR = 0.23%
[۳۰]	۲۰۱۷	KNN	Binary PSO (Hybrid)	Multi	KDD Cup 99	Accuracy (DoS = 99.91%, R2L = 99.73% ,U2R = 99.77%, Probe = 94.41%, Normal= 99.57%)
[۳۱]	۲۰۱۶	DT	PSO+GA	Multi	KDD Cup 99	DR = 98.74%, FAR = 1.26%
[۳۲]	۲۰۱۶	SVM	Time-Varying Chaos PSO(TVCPSO)	Multi	NSL-KDD	DR (TVCPSO–MCLP = 97.23, TVCPSO–SVM = 97.03) and FAR (TVCPSO–MCLP = 2.41, TVCPSO–SVM = 0.87)
[۳۳]	۲۰۱۹		Firefly algorithm(FA)	Multi	KDD Cup 99	Accuracy (Normal = 97.93%, DoS = 87.25%, Probe = 86.73, U2R = 62.8%, R2L = 47.1%)
[۲۳]	۲۰۱۹	Beysian network	Firefly algorithm(FA)	Multi	KDD Cup 99	DR (DoS = 99.98%, Probe = 93.42%, R2L = 98.73%, U2R = 68.97%), FPR (DoS = 0.01%, Probe = 0.01%, R2L = 0.01%, U2R = 0.0%)
[۳٤]	۲۰۲۰	-------	CSO(Cuckoo search optimization)	Multi	ISCX 2017 NSL-KDD KDDCup 99	Accuracy (Botnet ISCX 2017 = 99.98%, NSL-KDD = 99.6%, and KDD-Cup 99 = 99.94%)
[۳٥]	۲۰۲۰	SVM	CSO(Cuckoo search optimization)	Binary	NSL-KDD	DoS = 91%, Probe = 93%, U2R = 78%, R2L = 83%
[۳٦]	۲۰۲۰	ENN	CSO(Cuckoo search optimization)	Binary	NSL-KDD	Accuracy = 98.81%, DR = 97.25%, and FAR = 0.022%
[۳۷]	۲۰۱۹	NN	Artificial bee colony(ABC)	Binary	KDDCup 99	Accuracy = 94.53%, FPR = 7.028%, and FNR = 5.087%
[۳۸]	۲۰۱۹	SVM	ABC	Binary	KDDCup 99	DR = 99.67%, FPR = 0.17%
[۳۹]	۲۰۱۹	-----	ABC	Multi	NSL-KDD ISCXIDS2012	Accuracy = 98.90%, DR = 99.61%, FPR = 0.01%

جدول(۱): مروری بر کارهای پیشین انتخاب ویژگی با استفاده از الگوریتم‌های فراابتکاری

مرجع	سال	مدل	انتخاب ویژگی	نوع طبقه‌بندی	مجموعه داده	معیارها
[40]	۲۰۲۰	BLSTM	ABC	Multi	NSL-KDD	Accuracy = 84.25%, FPR (Normal= 25.7%, DoS = 1.52%,Probe = 1.15%, U2R = 0.09%,R2L = 0.91%), and DR (Normal= 97.5%, DoS = 87.55%,Probe = 85.76%
[۴۱]	۲۰۲۰	CNN	Bat algorithm (BA)	Binary	KDDCup 99	Accuracy = 98.42%, Precision = 98.5%, and Recall = 97.9%
[۱۸]	۲۰۲۰	-----	Pigeon inspired optimizer (PIO)	Multi	KDDCup 99, NLS-KDD UNSW-NB15	NSL-KDD (TPR = 0.86, FPR = 0.088, accuracy = 88.3%, and F1-score = 0.882),
[۴۲]	۲۰۲۱	SVM	Grey wolf optimization(GWO)	Binary	NLS-KDD	DR = 96%, and FAR = 0.03
[۴۳]	۲۰۲۱	DL	GWO + Crow search algorithm (CSA)	Binary Multi	CICIDS 2017, UNSWNB15, NSL-KDD	of NSL-KDD(Accuracy = 97.8%, precision= 97.8%, recall = 98.2%, F1-sore = 97.8%)
[۴۴]	۲۰۲۰	SVM	GWO	Multi	NSL–KDD	Accuracy = 89.18%
[۴۵]	۲۰۱۹	SVM	Cuttle fish optimization (CFO)	Multi	KDD Cup 99	Accuracy = 95.03%, DR = 95.23%, and FPR = 1.65%
[۴۶]	۲۰۲۱	SVM	Grasshopper optimization algorithm (GOA)	Binary	NSL–KDD, KDDCup 99	NSL-KDD (DR = 99.26%, FAR = 0.07, and accuracy = 99.89%)
[۴۷]	۲۰۲۴	BILSTM	Golden jackal optimization algorithm	Multi	CICIDS-2017	Accuracy = 99.7%, precision = 98.95%, recall = 98.95%,,F1-sore = 98.95%
[۴۸]	۲۰۲۳	DNN	improved Harris Hawk	Binary Multi	NSL-KDD, KDD Cup 99, UNSW-NB15	NSL-KDD (Accuracy = 86.79%, precision = 89.46%,F1-sore = 87.56%), KDD Cup 99 (Accuracy = 94.03%, precision = 93.78%, F1-sore = 92.24%), UNSW-NB15 (Accuracy= 81.92%, precision = 82.75%, F1-sore = 81.63%)
[۴۹]	۲۰۲۴	BILSTM	honey badger	Binary	CICIDS2017 CIC DoS	Accuracy = 97%, precision = 94%, sensitivity = 95%%, specificity= 90%
[۵۰]	۲۰۲۴	DL	breeding	Multi	NSL-KDD	Accuracy = 85.23%, precision = 78.72%, recall = 98%%,F1-score = 85%

۳. روش پیشنهادی

در این بخش، ابتدا مفهوم و لـزوم الگـوریتم‌هـای فراابتکـاری در انتخاب ویژگـی موردبررسـی قرارگرفتـه اسـت. سـپس الگـوریتم بهبودیافته شامپانزه دودویـی پیشـنهادی بـرای انتخـاب ویژگـی توضیح داده‌شده است. الگوریتم پیشـنهادی در قسـمت ۴ در فـاز پیش آموزش به‌منظور افزایش عملکرد مدل‌های یادگیری عمیـق در تشخیص نفوذ شبکه استفاده‌شده است

۳-۱- انتخاب ویژگی

یکی از مسائل مهم در حوزه یادگیری ماشـین و تشـخیص الگـو، انتخاب ویژگی می‌باشد. ویژگی‌ها تأثیر بسیار زیادی بر عملکرد، دقت و کارایی سیستم دارند. ویژگی‌های غیر مرتبط یا تـا حـدی مرتبط، می‌توانند به عملکرد سیستم آسیب بزنند. انتخاب ویژگی، یکی از مراحل اصلی در سیستم‌های یادگیری هوشمند محسـوب می‌شود. به‌ویژه زمانی که ابعاد فضای ویژگی داده بسیار بالا است. استفاده از ویژگی‌های مناسب می‌تواند هزینه محاسباتی مورد نیاز برای آموزش بهینه سیستم را کاهش دهد.

روش‌های انتخاب ویژگی شامل روش‌های فیلتر و بسته‌بنـدی هستند که زیرمجموعه‌ای از داده‌ها را انتخاب می‌کنند که حـاوی بسیاری یا تمامی ویژگی‌هـا هسـتند. روش بسـته‌بنـدی از دقـت پیش‌بینی یک الگوریتم یادگیری مشخص بـرای ارزیـابی کیفیـت

ویژگی‌های انتخاب‌شده استفاده می‌کند. الگوریتم‌های فرا ابتکاری به طور گسترده در روش‌های بسته‌بندی برای انتخاب ویژگی استفاده می‌شوند، زیرا توانایی جستجوی پیشرفته‌شان موردتوجه قرار می‌گیرد. الگوریتم‌های فرا ابتکاری به دلیل توانایی جستجوی پیشرفته‌شان، به طور گسترده برای انتخاب ویژگی مورداستفاده قرار می‌گیرند[۵۱, ۵۲].

۲-۳ الگوریتم شامپانزه پیوسته

در مطالعه[۵۳] ، یک روش فرا ابتکاری تحت عنوان الگوریتم شامپانزه معرفی‌شده است. این الگوریتم با الهام از رفتار شکار و توانایی‌های ذهنی متفاوت شامپانزه‌ها، به‌خصوص تمایلات جنسی و هوش آن‌ها، عمل می‌کند. الگوریتم شامپانزه باهدف حل مشکلات مربوط به‌سرعت پایین همگرایی و مشکلات در یافتن بهینه‌های محلی در مسائل با ابعاد بزرگ طراحی‌شده است.

هر شامپانزه نقش خاصی دارد که بر اساس تمایز خود تعیین می‌شود. بنابراین، هر شامپانزه در گروه تلاش می‌کند تا با استفاده از مهارت‌های منحصربه‌فرد خود، فضای جستجو را به‌صورت فردی کشف کند. برای موفقیت در فرآیند شکار، هر شامپانزه وظیفه‌ای خاص مانند تحریک‌کننده ، ایجاد مانع کننده ، تعقیب‌کننده و حمله‌کننده را بر عهده می‌گیرد.

شامپانزه‌ای که تحریک‌کننده است، طعمه را به سمت منطقه جستجو هدایت می‌کند بدون اینکه تلاشی برای گرفتن آن انجام دهد. شامپانزه‌هایی که نقش ایجاد مانع را دارند، با ساخت موانع در اطراف مسیر فرار طعمه، امنیت اطراف را فراهم می‌کنند. در همین حال، شامپانزه‌های تعقیب گر به دنبال طعمه می‌روند و سعی در گرفتن آن دارند. شامپانزه حمله‌گر، که در پایان عملیات شکار قرار دارد، مسیر فرار شکار را پیش‌بینی کرده و به آن حمله می‌کند. این شامپانزه باید به‌اندازه‌ای زیرک باشد که بتواند حرکت‌های بعدی شکار را تخمین بزند و نقش کلیدی در موفقیت شکار ایفا کند. پس از به دست آوردن شکار موفق، آن‌ها با قطعات بزرگ‌تری از گوشت به‌عنوان پاداش مواجه می‌شوند. همچنین، شامپانزه‌ها می‌توانند در طول شکار نقش‌های خود را تغییر دهند یا طی کل فرآیند همان نقش را حفظ کنند.

گوشت به‌دست‌آمده از شکار معمولاً برای مواردی مانند اعتبار اجتماعی، جنسیت، یا حمایت در نظر گرفته می‌شود، که این امر می‌تواند انگیزه‌های اجتماعی را در پی داشته باشد. علاوه بر این، میل جنسی شامپانزه‌ها آن‌ها را به سمت رفتارهای غیرمنظم در مراحل نهایی شکار سوق می‌دهد. این میل جنسی باعث می‌شود که شامپانزه‌ها در مراحل پایانی شکار به‌طور غیر منظم عمل کنند و به‌شدت برای کسب گوشت تلاش کنند. در نتیجه، همه شامپانزه‌ها نقش‌های خود را رها کرده و تلاش می‌کنند گوشت را به‌صورت فردی به دست آورند. روند شکار توسط شامپانزه‌ها به دو مرحله اکتشاف و بهره‌برداری تقسیم می‌شود.

در مرحله اکتشاف، شامپانزه‌ها تحریک می‌کنند، مسدود می‌کنند و طعمه را طبق شکل ۱-الف تعقیب می‌کنند و در مرحله بهره‌برداری، مطابق شکل ۱-ب شامپانزه‌ها به طعمه حمله می‌کنند. فرآیند شکار شامپانزه توسط مرحله‌های اکتشاف و بهره‌برداری انجام می‌شود که در معادلات (۱) و (۲) مدل‌سازی شده است.

$$|(d=|c.x\_prey\ (t)-m\ ⟦.x⟧\ \_chimp\ (t \quad (۱)$$

$$x\_chimp\ (t+1)=x\_prey\ (t)-a.d \quad (۲)$$

مدل ریاضی برای تعقیب و تحریک به‌صورت d است. بردار موقعیت شامپانزه با x_chimp و بردار شکار با x_prey نشان داده می‌شود. شماره تکرار با t نشان داده‌شده است. a و c بردارهای ضریب تخمین هستند که از معادلات (۳) و (۴) به دست می‌آید:

$$a=2.f.r\_1-f \quad (۳)$$

$$c=2.r\_2 \quad (۴)$$

الف- فاز اکتشاف

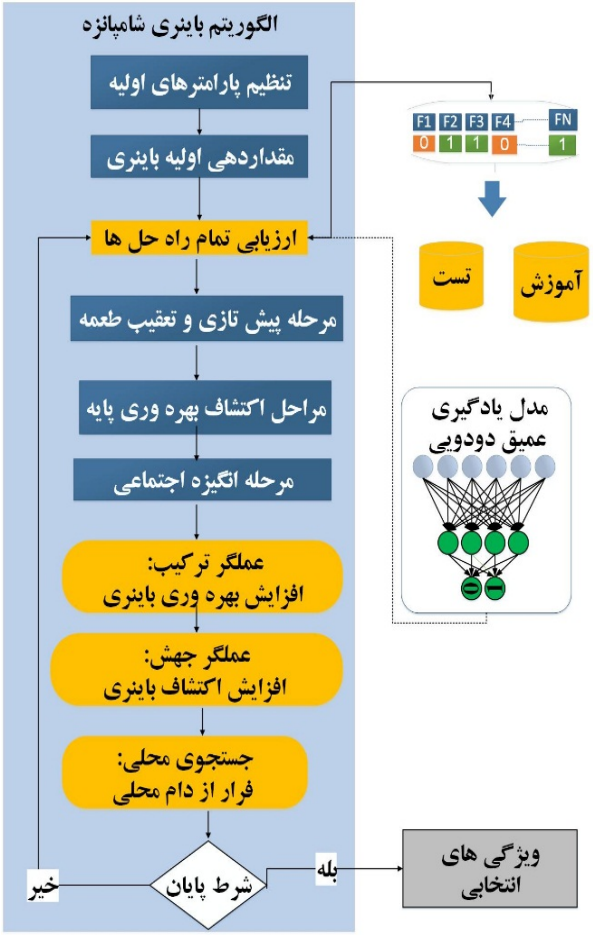
ب- فاز بهره‌برداری

شکل(۱): فاز اکتشاف و بهره‌برداری بهینه‌ساز شامپانزه[۵۳]

۳-۳ الگوریتم شامپانزه بهبودیافته

الگوریتم شامپانزه برای حل مسائل پیوسته طراحی‌شده است و در حل مسائل دودویی نمی‌تواند کارآمد باشد. بنابراین نیاز است تغییراتی در این الگوریتم برای حل مسائل دودویی انجام شود. ازاین‌رو یک نسخه بهبودیافته الگوریتم شامپانزه برای مسئله انتخاب ویژگی در تشخیص نفوذ مبتنی بر شبکه‌های اینترنت اشیا در این بخش طراحی و پیاده‌سازی شده است که BCHO نام‌گذاری شده است. فلوچارت روش پیشنهادی انتخاب ویژگی BCHO در شکل(۲) نشان داده‌شده است.

روش BCHO از پنج مرحله مختلف تشکیل‌شده است. در اولین مرحله جمعیت اولیه الگوریتم بهبودیافته شامپانزه به‌صورت تصادفی به‌صورت دودویی تولیدشده است. در دومین مرحله، بعد از اعمال عملگرهای مختلف بر روی جمعیت از توابع انتقال برای تبدیل راه‌حل‌های پیوسته به راه‌حل‌های دودویی استفاده‌شده است. در سومین مرحله از عملگر ترکیب الگوریتم ژنتیک برای افزایش همگرایی و بهره‌وری الگوریتم استفاده‌شده است. در چهارمین مرحله از عملگر جهش الگوریتم ژنتیک برای افزایش تنوع و همگرایی الگوریتم استفاده‌شده است. در پنجمین مرحله



شکل (۲): فلوچارت روش پیشنهادی انتخاب ویژگی BCHO

یک روش جدید جستجوی محلی برای فرار از دام محلی طراحی‌شده است. در ادامه مراحل بهبود و تغییرات الگوریتم شامپانزه برای حل مسئله دودویی گام‌به‌گام تشریح شده است.

۳-۳-۱تولید جمعیت اولیه دودویی

اولین گام در روش BCHO تولید جمعیت اولیه دودویی با توجه به تعداد ویژگی‌های مجموعه داده‌ها است. در واقع با توجه به ابعاد (dim) و تعداد جمعیت (N) باید راه‌حل دودویی تولید شود. در این گام با توجه به رابطه (۵) یک‌راه حل دودویی به‌صورت تصادفی تولید می‌شود.

$$X_{ij}=\begin{cases}0 & \text{if } r1<0.5\\ 1 & \text{if } r1\geq 0.5\end{cases} \qquad (۵)$$

در رابطه (۵)، r_1 یک عدد تصادفی بین صفر و یک است. متغیر i به راه‌حل i ام از جمعیت اشاره می‌کند و همچنین متغیر j به بُعد jام از راه‌حل i ام از جمعیت اشاره می‌کند. بدین ترتیب در اولین گام از BCHO تمام راه‌حل‌ها به‌صورت دودویی تولید و عملگرهای اصلی الگوریتم شامپانزه بر روی آن اعمال می‌شوند.

۳-۳-۲ عملگرهای پایه

در الگوریتم شامپانزه، چندین عملگر پایه تعریف‌شده است که درروش BCHO نیز بدون تغییر بر روی جمعیت اولیه اعمال می‌شوند و در مرحله بعد از تابع انتقال برای دودویی سازی راه‌حل‌ها، استفاده می‌شوند. در ادامه به‌طور خلاصه این عملگرها موردبررسی قرار می‌گیرند.

۳-۳-۳ توابع انتقال

راه‌حل‌های تولیدشده توسط عملگرهای اصلی الگوریتم شامپانزه ممکن است که به فضای پیوسته نگاشت داده شوند. از این رو برای تبدیل و نگاشت این راه‌حل در این مرحله از تابع انتقال استفاده‌شده است که به‌صورت رابطه (۶) قابل‌تعریف است.

$$XTF_{ij}=\begin{cases}0 & \text{if } TF(X_{ij})<r2\\ 1 & \text{if } TF(X_{ij})\geq r2\end{cases} \qquad (۶)$$

در رابطه(۶)، r_2 یک عدد تصادفی بین صفر و یک است. XTF_{ij} یک راه‌حل دودویی است که توسط تابع انتقال $TF(X_{ij})$ به صورت دودویی تبدیل‌شده است. تابع انتقال $TF(X_{ij})$ براساس رابطه (۷) به دست می‌آید:

$$TF(X_{ij})=\frac{1}{1+e^{-X_{ij}}} \qquad (۷)$$

بنابراین با توجه به روابط (۶) و (۷) همه راه‌حل‌های موجود در جمعیت به حالت دودویی تبدیل می‌شوند و قابل‌استفاده برای مسئله انتخاب ویژگی می‌شوند.

۳-۳-۴ عملگرهای ترکیب و جهش

ممکن است روش پیشنهادی BCHO از اکتشاف و بهره‌وری ضعیف‌تری برخوردار باشد. بنابراین، در این روش از عملگرهای ترکیب و جهش ژنتیک برای افزایش اکتشاف و بهره‌وری استفاده‌شده است. در اینجا، عملگر ترکیب تک نکته‌ای برای افزایش بهره‌وری بیشتر بر روی راه‌حل فعلی و بهترین راه‌حل استفاده می‌شود و سپس به‌صورت احتمالی عملگر جهش مبتنی بر مخالفت بر روی یکی از راه‌حل‌های عملگر ترکیب اعمال می‌شود. این عملیات در الگوریتم (۱) نشان داده‌شده است.

طبق الگوریتم(۱)، دو راه‌حل فعلی(X_{inew}) و بهترین راه‌حل (X_{best}) برای عملگر ترکیب نقطه‌ای در ابتدا به‌احتمال ۰/۵. انتخاب می‌شوند و سپس یک نقطه‌ای به‌صورت تصادفی $point1$ با توجه ابعاد dim مسئله انتخاب می‌شود و در گام بعدی دو راه‌حل جدید یا راه‌حل فرزند به نام‌های $Xnew_0$ و $Xnew_1$ تولید می‌شوند. در گام بعدی به‌احتمال۰/۱ عملگر جهش مبتنی بر مخالفت فقط بر روی راه‌حل $Xnew_0$ و بُعد $point2$ اعمال

می‌شود. در این عملگر جهش مبتنی بر مخالفت، اگر بُعد *point2* برابر با یک باشد، به صفر و اگر برابر با صفر باشد، به یک تبدیل می‌شود که باعث افزایش اکتشاف بیشتر بر روی راه‌حل $Xnew_0$ می‌شود. در نهایت، راه‌حلی که تابع هدف بهتری داشته باشد از بین راه‌حل فعلی X_{inew} و راه‌حل‌های جدید $Xnew_0$ و $Xnew_1$ انتخاب‌شده و به‌عنوان راه‌حل فعلی در نظر گرفته می‌شود.

الگوریتم ۱: شبه کد عملیات ترکیب و جهش مبتنی بر مخالفت در روش پیشنهادیBCHO

00:start
01: $Solution_1 = X_{inew}$
02: $Solution_2 = X_{best}$
03: point1 = one random rand of 1-dim
04: $Xnew_0$ = hstack($Solution_1$[1: point1]$Solution_2$ [point1: dim])
05: $Xnew_1$ = hstack($Solution_2$ [1: point1], $Solution_1$ [point1: dim]))
06: if(rand >0.1)
07: point2 = one random rand of 1-dim
08: $Xnew_0$[point] = 1 - $Xnew_0$ [point2]
09: Fit_0 = fitness($Xnew_0$)
11: Fit_1 = fitness ($Xnew_1$)
12: Fit_{inew}= fitness (X_{inew})
13: X_{inew}= min of $\{Fit_0, Fit_1, Fit_{\text{inew}}\}$
14: end

۳-۳-۵ عملگرهای جستجوی محلی

در زیر بخش قبل، از عملگرهای ژنتیک، ترکیب و جهش برای افزایش اکتشاف و بهره‌وری در جمعیت استفاده شد. اما یک نکته مهم در مسائل دودویی وجود دارد که فضای مسئله به‌صورت [۰و۱] است و این ممکن است منجر به گیر افتادن الگوریتم‌ها در دام بهینه محلی شود. در این بخش، یک روش جستجوی محلی جدید برای فرار از بهینه محلی در نظر گرفته‌شده است که با توجه به مسئله دودویی طراحی‌شده است و جزئیات آن در الگوریتم (۲) آمده است.

طبق الگوریتم (۲)، عملگر جدید جستجوی محلی برای فرار از بهینه محلی به اندازه $XMax_{LS}$ تکرار بر روی یک‌راه حل تصادفی از جمعیت اعمال می‌شود که در مقدار آن برابر با ۱۰ تنظیم‌شده است. قانون این عملگر جدید جستجوی محلی بدین‌صورت است که اندیکس‌هایی که مقدار آن صفر است را در idx_0 و اندیکس‌هایی که مقدار آن‌یک است را در idx_0 ذخیره می‌کند. در هر مرحله سعی می‌کند که از دسته‌ای از اندیکس‌ها یکی از آن‌ها را برعکس کند تا به نتایج بهتری دست پیدا کند. البته قانون جایگزین هم به این صورت است که راه‌حل جدید بهتر از راه‌حل قبلی باشد یا $r_3 > 0.95$ باشد. با این مقداردهی تا حدودی عملیات اکتشاف هم در این عملگر اعمال می‌شود.

الگوریتم(۲): شبه کد عملگر جدید جستجوی محلی برای فرار از بهینه محلی در روش پیشنهادیBCHO

00:start
01: $\text{Solution}_{\text{last}} = \text{X}_{\text{rand}}$
02: $\text{fit}_{\text{last}} = \text{fitness}(\text{X}_{\text{rand}})$
03: for it =1 to XMax_{LS}
04: idx_0=select dim from $\text{Solution}_{\text{last}}$ is 0
05: idx_1=select dim from $\text{Solution}_{\text{last}}$is 1
06: R_0= one rand from idx_0
07: R_1= one rand from idx_1
08: $\text{Solution}_{\text{new}} = \text{Solution}_{\text{last}}$
09: $\text{Solution}_{\text{new}}[\text{R}_0] = 1$
10: $\text{Solution}_{\text{new}}[\text{R}_1] = 0$
11: $\text{fit}_{\text{new}} = \text{fitness}(\text{Solution}_{\text{new}})$
12: if $\text{fit}_{\text{new}} < \text{fit}_{\text{last}}$ or r3 >0.95
13: $\text{Solution}_{\text{last}} = \text{Solution}_{\text{new}}$
14: $\text{fit}_{\text{last}} = \text{fit}_{\text{new}}$
15: **End**

۳-۳-۶ تابع هدف

فرایند انتخاب ویژگی، دو هدف اصلی را دنبال می‌کند. هدف اول کاهش یا حذف ویژگی‌های اضافی و انتخاب ویژگی‌های اساسی است که می‌توان آن‌ها را از زاویه‌ی یک تابع کمینه دید. هدف دوم افزایش دقت تشخیص حملات یا کاهش خطاهای تشخیص حملات است که می‌توان آن را نیز از زاویه‌ی یک تابع کمینه بررسی کرد. بنابراین، برای انتخاب ویژگی، این دو تابع باید به‌صورت وزنی تعریف شوند که در رابطه (۷) مشخص‌شده است.

$$\text{تابع هزینه} = \alpha \times (\text{Erorr}_{BCHO}) + (1-\alpha) \times \frac{\text{ویژگی تعداد انتخاب‌های شده با } bcho}{\text{تعداد کل ویژگی ها}} \quad (۷)$$

در رابطه (۷) α برابر با۹۹ /۰ تنظیم‌شده است که میزان اهمیت دقت مدل تشخیص را نشان می‌دهد و $(1-\alpha)$ نیز به اهمیت تعداد ویژگی‌های انتخابی از کل ویژگی‌های موجود اشاره دارد.

۳-۴ پیش‌پردازش

داده‌های اولیه ممکن است حاوی اطلاعات نامعلوم و اغتشاش بوده که بی‌تردید بر کارایی مدل تأثیرگذار خواهد بود. مدل ارائه‌شده نیازمند به‌کارگیری چندین مرحله پیش‌پردازش پیش از استفاده از تکنیک‌های یادگیری عمیق است. این فرآیندهای پیش‌پردازش به بهبود کیفیت داده‌های مربوط به اینترنت اشیا، ارتقای دقت مدل ارائه‌شده و کاهش پیچیدگی محاسبات کمک می‌کنند. سه مرحله کلیدی پیش‌پردازش شامل (۱) رفع مقادیر گمشده، (۲) رمزنگاری ویژگی‌های دسته‌ای و (۳) نرمال‌سازی MinMax انجام‌شده است. پس از اجرای این سه مرحله پیش‌پردازش، همه داده‌ها آماده برای به‌کارگیری در مدل پیشنهادی و دیگر روش‌های مقایسه‌ای می‌شوند. در ادامه، این سه فرآیند پیش‌پردازش به‌طور خلاصه شرح داده‌شده‌اند.

- **پاک‌سازی داده‌ها**

گام نخست در فرایند پیش‌پردازش داده‌ها معمولاً به پاک‌سازی داده‌ها اختصاص دارد. این مرحله شامل فرآیندهایی

نظیر حذف داده‌های تکراری، تکمیل کردن اطلاعات ناقص، دور ریختن داده‌های بی‌ربط، و مواردی ازاین‌دست است. در هر سه بخش از داده‌های شبکه اینترنت اشیا، دستکاری و مدیریت داده‌های گمشده حائز اهمیت است، چراکه مدل‌های پیشنهادی و الگوریتم‌های دیگر نمی‌توانند با وجود داده‌های گمشده به‌درستی آموزش ببینند. یکی از شیوه‌های متداول برای مقابله با این داده‌های ناقص، جایگزین کردن مقادیر NaN با مقداری است که بیشترین تکرار را در آن ستون دارد. در این مقاله، برای جایگزینی داده‌های گمشده از روش "mod" با بهره‌گیری از کتابخانه sklearn و تابع input در پایتون برای هر سه مجموعه داده استفاده‌شده است.

- **رمزنگاری دسته‌ای ویژگی**

همه ورودی‌های استفاده‌شده در متدهای یادگیری ماشین و مدل موردنظر باید در قالب آرایه‌هایی از اعداد صحیح یا اعداد ممیزی شناور باشند. البته، در داده‌های موجود ممکن است ویژگی‌هایی با نوع دسته‌بندی‌شده وجود داشته باشند که نیاز به تبدیل به نوع دیگری در این مرحله‌دارند. دو روش متداول برای این تبدیل، استفاده از روش‌های کدگذاری برچسب [1] و کدگذاری وان هات [2] می‌باشد. چون استفاده از روش کدگذاری وان هات می‌تواند به دلیل افزایش بیش‌ازحد تعداد ویژگی‌ها به نسبت تعداد دسته‌ها، کمتر کارآمد باشد. در این تحقیق از روش کدگذاری برچسب موجود در کتابخانه sklearn برای تغییر ویژگی‌های با نوع رشته‌ای یا دسته‌بندی‌شده به اعداد استفاده شده‌است.

- **نرمال‌سازی حداقل-حداکثر**

نرمال‌سازی یک فرایند است که در آن، داده‌های مربوط به هر ویژگی به‌گونه‌ای تغییر مقیاس یا تبدیل می‌شوند که در یک بازه خاص قرار بگیرند. روش حداقل-حداکثر یکی از روش‌های معمول برای تغییر داده‌ها به یک مقیاس و بازه نوین است که با استفاده از فرمول زیر انجام محاسبه می‌شود:

$$\boldsymbol{X_{norm}} = \frac{X_i - Min\,(X_i)}{Max(X_i) - Min\,(X_i)} \qquad (۸)$$

X_i نشان‌دهنده ارزش ویژگی عددی برای نمونه i است، Min و Max نشان‌دهنده حداقل و حداکثر ارزش‌های ویژگی‌های عددی می‌باشند. با استفاده از مقیاس‌بندی ، تمامی ویژگی‌ها در سه دیتاست به مقادیری در بازه صفرتا یک تبدیل می‌شوند.

۴. ارزیابی و نتایج

در این بخش ابتدا مجموعه داده‌های استفاده‌شده برای ارزیابی معرفی می‌شوند. سپس معیارهای ارزیابی الگوریتم‌های فرا ابتکاری برای مقایسه عملکرد آن‌ها تعریف می‌شود. در ادامه، از روش پیشنهادی برای انتخاب ویژگی در مجموعه داده‌های تعریف‌شده باهدف افزایش دقت استفاده می‌شود.

۴-۱ مجموعه داده‌ها

برای ارزیابی الگوریتم‌های انتخاب ویژگی، از مجموعه داده‌های با تعداد زیادی ویژگی استفاده می‌شود. در این مقاله، برای ارزیابی عملکرد مدل پیشنهادی و الگوریتم‌های دیگر در مسئله تشخیص نفوذ شبکه‌های اینترنت اشیا، سه مجموعه داده معتبر مورد استفاده قرار می‌گیرد. این مجموعه داده‌ها دارای ویژگی‌های منحصربه‌فرد برای تشخیص حملات سایبری در شبکه‌های اینترنت اشیا می‌باشند. این سه مجموعه داده شامل ToN-IoT، UNSW-NB 15 و IoTID20 هستند که تعداد نمونه‌های آن‌ها در جدول (۲) نشان داده‌شده است. در ادامه، توضیحات مختصری در مورد هر مجموعه داده ذکرشده است.

جدول (۲) مشخصات مجموعه داده‌ها

مجموعه داده	IoTID20	UNSW-NB15	ToN-IoT
تعداد رکوردهای نرمال	٤٠٠٧٣	١٦٤٦٧٣	٣٠٠٠٠٠
تعداد رکوردهای حمله	٥٨٥٧١٠	٩٣٠٠٠	١٦١٠١٤٣
تعداد کل ویژگی‌ها	٨٣	۴۵	٤٢
تعداد کلاس طبقه‌بندی	٢	٢	٢

۴-۱-۱مجموعه داده ToN-IoT

تمام نمونه‌های داده در مجموعه داده ToN-IoT از ۴۵ ویژگی تشکیل شده است. ۴۳ ویژگی اول برای صفت‌ها استفاده می‌شود و ویژگی ۴۴ ام با عنوان عادی بدون حمله و یا حمله، برچسب‌گذاری می‌شود و ویژگی آخر با عنوان بدون حمله یا نوع حمله‌ها برچسب‌گذاری شده است. مجموعه داده‌ای که برای آموزش و آزمون استفاده‌شده است شامل ۴۶۱۰۴۳ رکورد داده شبکه است. ۳۰۰۰۰۰ رکورد از آن ها عادی است و ۱۶۱۰۴۳ رکورد، بیانگر حمله است.

۴-۱-۲ مجموعه داده UNSW-NB15

مجموعه داده UNSW-NB15 که در آزمایشگاه امنیت سایبری استرالیا توسط IXIA Perfect-Storm ایجادشده، برای تولید و شبیه‌سازی داده‌های واقعی و به‌روز طراحی‌شده است. این مجموعه داده شامل ۴۹ ویژگی است که توسط دوازده الگوریتم برای ایجاد برچسب‌های کلاس تولیدشده‌اند. بخشی از این داده‌ها به دو بخش مجزا تقسیم می‌شود: یک بخش برای آموزش با ۱۷۵۳۴۱ رکورد و دیگری برای آزمون با ۸۲۳۳۲ رکورد، که هر دو شامل نمونه‌هایی از رفتارهای معمولی و انواع مختلف حملات هستند. هر مجموعه داده دارای ۴۵ ویژگی است. از این ۴۵ ویژگی، ۴۳ ویژگی برای توصیف صفات به کار می‌روند، ویژگی ۴۴ به‌عنوان نرمال یا حمله برچسب‌گذاری می‌شود و آخرین ویژگی نیز نوع نرمال یا حمله را مشخص می‌کند.

1 LabelEncoder
2 One Hot Encoding

۳-۱-۴مجموعه داده IoTID20

مجموعه داده IoTID20 با استفاده از دستگاه‌های هوشمند خانگی شامل SKT NGU و دوربین‌های وای فای EZVIZ طراحی و پیاده‌سازی شده‌است. یک محیط آزمایشی شامل دستگاه‌های اینترنت اشیاء و یک معماری اتصال برای این مجموعه‌داده تدارک دیده شده‌است. برای ساخت این مجموعه‌داده، از یک خانه هوشمند شامل دستگاه هوشمند SKTNGU و دوربین وای فای EZVIZ استفاده شده است. مجموعه داده شامل ۸۰ ویژگی مربوط به شبکه به همراه ویژگی‌هایی با برچسب‌های دودویی، دسته‌بندی و زیردسته می‌باشد. بزرگ‌ترین مزیت مجموعه داده IoTID20 این است که توانایی بازسازی یک‌روند پیشرفته در ارتباطات شبکه‌ی IoT را دارد. نسخه نهایی این مجموعه داده دارای ۸۳ ویژگی شبکه است و شامل ۴۰۰۷۳ نمونه نرمال و ۵۸۵۷۱۰ نمونه دارای حمله است.

۴-۲ معیارهای ارزیابی

در این بخش، معیارهای مختلف برای ارزیابی عملکرد مدل پیشنهادی و سایر الگوریتم‌های مقایسه‌ای معرفی می‌شود. با توجه به مسئله طبقه‌بندی، چهار معیار از رایج‌ترین معیارهای اعتبارسنجی شامل دقت[۱]، صحت[۲] بازخوانی[۳]، و امتیاز F1[۴] براساس ماتریس درهم‌ریختگی قابل‌تعریف است.

$$\text{Accurecy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (۹)$$

$$\text{precision} = \frac{TP}{TP + FP} \quad (۱۰)$$

$$\text{TPR or Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (۱۱)$$

$$\text{F1} - \text{score} = 2 * \frac{precision * recall}{precision + recall} \quad (۱۲)$$

مثبت واقعی[۵] نشان‌دهنده تعداد نمونه‌های ترافیک حملاتی است که به‌درستی توسط مدل پیشنهادی طبقه‌بندی‌شده‌اند .منفی واقعی[۶] نمونه ترافیک عادی را نشان می‌دهد که به‌درستی توسط مدل پیشنهادی طبقه‌بندی‌شده‌اند. مثبت کاذب[۷] تعداد نمونه ترافیک‌ها عادی هستند، اما به‌اشتباه توسط مدل پیشنهادی به‌عنوان نمونه حمله طبقه‌بندی‌شده‌اند. منفی کاذب[۸] تعداد نمونه ترافیک‌ها حملاتی هستند، اما به‌اشتباه توسط مدل پیشنهادی به‌عنوان نمونه ترافیک عادی طبقه‌بندی‌شده‌اند. اما علاوه بر این چهار معیار، معیارهای دیگر شامل نرخ مثبت کاذب[۹]، نرخ منفی کاذب[۱۰] برای ارزیابی عملکرد مدل پیشنهادی بر اساس ماتریس درهم‌ریختگی و همچنین برای ارزیابی الگوریتم BCHO در بحث انتخاب ویژگی معیار نرخ کاهش ویژگی استفاده‌شده است.

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (1)$$

$$\text{FNR} = \frac{\text{FN}}{\text{TP} + \text{FN}} \quad (2)$$

$$\text{FRR} = 1 - \frac{Number\ Of\ Selected\ Features}{Number\ Of\ All\ Features} \quad (3)$$

نرخ کاهش ویژگی به‌منظور ارزیابی کارایی الگوریتم‌ها با هدف کاهش ابعاد مورداستفاده قرار می‌گیرد. این معیار نشان می‌دهد که الگوریتم چقدر در کاهش تعداد ویژگی‌های داده موفق است

۴-۳ تنظیم پارامترها

برای تشخیص حمله از شبکه‌های عصبی کانولوشنی[۱۱] استفاده‌شده است.بر ای استفاده از مدل، پارامترهای شیکه کانولوشنی به صورت جدول (۳) در نظر گرفته شده است.

جدول (۳): تنظیم پرامترهای شبکه های کانولوشنی

پارامترها	مقادیر
Input size	(100,100,3)
Batch size	128
CONV2D filters	4
CONV2D kernel size	(1,1)
Permute(dimes)	(3,1,2)
CONV2D filters	128
CONV2D kernel size	(2,2)
CONV2D filters	32
CONV2D kernel size	(1,1)
Fattern dropout	0.2
Fattern dense	512
Fattern activation	RELU
Fully connected dense	2
Fully connected activation	SOFTMAX

1 accuracy
2 precision
3 recall
4 F1-score
5 True Possitive(TP)
6 True Negative(TN)
7 False Positive(FP)
8 False Negative(FN)
9 False Positive Rate (FPR)
10 False Negative Rate (FNR)

11 convolutional neural networks

برای اعتبارسنجی مجموعه داده‌ها، اعتبارسنجی متقابل k-fold انجام‌شده است.همانطور که در قسمت ارزیابی ثابت‌شده است. بهترین عملکرد در پنجاه‌درصد داده های آموزشی و پنجاه‌درصد داده های آزمایشی به‌دست‌آمده است. حداقل تعداد تکرار تنظیم‌شده در هر اجرا برابر با ۱۰۰ در نظر گرفته‌شده است.

در حین اجرای روشهای یادگیری، وجود پارامترهای تصادفی باعث تولید مقادیر متفاوت و متغیر شده است. بنابراین، برای هر مورد، اجراها به صورت ۱۰ بار (هر بار به مدت ۱۰۰ دور) انجام‌شده است و میانگین نتایج این اجراها به عنوان جواب نهایی در نظر گرفته می شود

۴-۴ نتایج تجربی

انتخاب وزن و ویژگی به طور مکرر در یادگیری ماشین برای پردازش داده‌های با ابعاد بالا استفاده می‌شود. این کار تعداد ویژگی‌ها را در مجموعه داده کاهش می‌دهد و فرآیند طبقه‌بندی را آسان‌تر می‌کند. ارزیابی‌های تجربی بر روی سه مجموعه داده IOTID20T،TONIOT, UNSWNB15 انجام‌شده است تا بهبودهای قابل‌توجهی که در فرایند طبقه‌بندی توسط تمام الگوریتم‌ها مشاهده می‌شود، نشان داده شود.

در این مقاله الگوریتم پیشنهادی BCHO با چهار الگوریتم انتخاب ویژگی بهینه‌سازی کبوتر[1] [۲۵] ، ازدحام ذرات بهینه‌شده[2][۲۹]، خفاش دودویی بهینه‌شده[3][۵۴] و شاهین هریس دودویی [4][۵۵]برای سیستم‌های تشخیص نفوذ مقایسه شده است. دلیل انتخاب الگوریتم های مورد مقایسه این است که این مقالات جدید هستند و نشان داده اند که عملکرد بهتری نسبت به روش‌های فراابتکاری قدیمی‌تر دارند. بنابراین، نیازی به مقایسه با روش های قدیمی‌تر نیست. از طرفی در مقاله [۵۶] الگوریتم شامپانزه پایه با روش های ابتکاری جدیدتر نیز مقایسه شده است و بعد از ارزیابی ثابت‌شده است که الگوریتم شامپانزه پایه، دارای دقت فراتری از آن‌ها است.

مدل پیشنهادی و سایر الگوریتم‌های مقایسه‌ای، با استفاده از فریمورک تنسور فلو و کراس، پیاده‌سازی شده و در محیط گوگل کولب با دارا بودن ۲۵ گیگابایت رم و پردازنده گرافیک T4 اجراشده‌اند. در این بخش، به ارزیابی عملکرد مدل پیشنهادی با انجام آزمایش‌های مختلف پرداخته‌شده است.

قبل از اجرای الگوریتم BCHO بر روی مجموعه داده های مورد آزمون، مراحل پیش‌پردازش اعمال‌شده‌اند. در این مرحله، از تکنیک‌های مختلف نرمال‌سازی استفاده شده است. جدول(۴) مقادیر نرمال‌سازی داده‌ها برای تکنیک‌های نرمال‌سازی مختلف را نمایش می‌دهد. تکنیک‌های نرمال‌سازی مورد استفاده شامل نرمال‌سازی حداقل-حداکثر، نرمال‌سازی z-score و نرمال‌سازی تانژانت هیپربولیک هستند. این جدول نشان می‌دهد که روش نرمال‌سازی حداقل-حداکثر دقت، حساسیت و ویژگی بهتری را برای تمام الگوریتم‌های بهینه‌سازی در مجموعه داده‌های داده‌شده به دست آورده است. بر همین اساس در مرحله پیش‌پردازش از روش حداقل- حداکثر برای نرمال‌سازی داده‌ها استفاده‌شده است. بعد از آماده‌سازی اولیه مجموعه داده و پس از انجام پیش‌پردازش، نوبت به انتخاب ویژگی با استفاده از الگوریتم پیشنهادی می‌رسد.

جدول(۴) نرمال‌سازی داده‌ها با تکنیک‌های نرمال‌سازی مختلف

معیار ارزابی	تکنیکهای نرمال‌سازی	مجموعه داده		
		امتیاز f1	صحت	دقت
TONIOT	Min-max	۹۳/۲	/۱۵ ۹۲	۹٤/۸
	z-score	۸۴/۱۲	/۹٦ ۸۳	۸٥/۸
	tanh	۹۲/۹	/۱۳ ۸۸	۹۳/۲
UNSWNB15	Min-max	۹۳/۱	/۱٥ ۹۳	/۹۷ ۹۲
	z-score	۹۳	۹۲/٤	/۲٥ ۹۱
	tanh	۹۳/٤	۹۰	۸۹/۳
IoTID20	Min-max	۹۸/٤	/٦۲ ۹۲	/۷۳ ۹٥
	z-score	۹۱	۹۰/۷	۹٦/۱
	tanh	۹۷/٥	/۱۸ ۹۱	۹٥/۲

همگرایی سریع به سمت تابع هدف نشان‌دهنده کارایی الگوریتم‌های فراابتکاری است. یافته‌های آزمایشی نشان می‌دهد که روش BCHO با افزایش تعداد تکرارها، توانسته در سه مجموعه داده مورد بررسی به همگرایی مطلوبی دست یابد و از الگوریتم‌های دیگری که مورد مقایسه قرارگرفته، پیشی بگیرد. همچنین، استفاده از عملگرهای ژنتیک و جستجوی محلی، تأثیر بلافصل و مثبتی بر عملکرد اولیه روش BCHO داشته و این روش از همان دورهای ابتدایی، نتایج بهتری را ارائه داده است. بر اساس نتایج به‌دست‌آمده در زمینه انتخاب ویژگی، مشخص‌شده که روش BCHO به طور متوسط توانسته است ۶۰ درصد از ویژگی‌ها را در سه مجموعه داده کاهش دهد و به همگرایی بهتری دست یابد.

در جدول (۵)، نتایج انتخاب ویژگی با استفاده از الگوریتم BCHO برای سه مجموعه داده ITon-IoT ، UNSWNB15، و IoTID20آورده شده است. این الگوریتم پیشنهادی به ترتیب ۱۹، ۱۶، و ۳۰ ویژگی را انتخاب کرده است. در همه‌ی سه مورد، الگوریتم شامپانزه پیشنهادی نه تنها از نظر تعداد ویژگی‌های انتخاب‌شده بلکه از نظر همگرایی نیز بهترین عملکرد را از خود نشان داده است. بعد از طی مراحل پیش‌پردازش و انتخاب ویژگی، دقت مدل نیز باید بررسی شود.

[1] Binary Pigeon Inspired(BPI)
[2] BinaryParticle swarm optimization(BPSO)
[3] Binary Bath(BBAT)
[4] Binary Harris Hawk Optimization(BHHO)

جدول (۵)نتایج انتخاب ویژگی BCHO و سایر الگوریتم ها

داده	الگوریتم	ویژگی‌های انتخاب‌شده	انتخابی
مجموعه داده ToN-IoT(۴۳ ویژگی)	BPI	۱،۲،۳،۴،۵،۷،۸،۹،۱۱،۱۲،۱۳،۱۵،۱۶،۱۸، ۲۰،۲۴،۲۶،۲۸،۳۰،۳۱،۳۳،۳۵،۳۶،۳۹،۴۰، ۴۱	۲۶
	BPSO	۱،۲،۳،۵،۹،۱۰،۱۳،۱۵،۱۶،۱۸،۲۰،۲۲،۲۳، ۲۴،۲۵،۲۸،۳۰،۳۴،۳۵،۳۶،۳۷،۳۹،۴۰،۴۲، ۴۳	۲۵
	BBATH	۱،۴،۶،۷،۹،۱۰،۱۳،۱۴،۱۵،۲۰،۲۲،۲۵،۲۹، ۳۱،۳۴،۳۵،۳۸،۴۱،۴۲،۴۳	۲۰
	BHHO	۱،۲،۳،۴،۵،۶،۷،۸،۹،۱۰،۱۴،۱۷،۱۹،۲۰،۲ ۲،۲۶،۲۷،۲۹،۳۲،۳۳،۳۵،۳۶،۳۷،۳۸،۳۹،۴ ۰،۴۱،۴۳	۲۸
	BCHO	**۱،۲،۴،۵،۶،۱۰،۱۱،۱۶،۱۷،۱۸،۲۰،۲۱،۲۲، ۲۳،۲۴،۲۵،۲۹،۳۷،۴۳**	**۱۹**
مجموعه داده UNSWNB-15(۴۳ ویژگی)	BPI	۱،۵،۹،۱۰،۱۱،۱۲،۱۳،۱۵،۱۷،۱۸،۲۰،۲۲، ۲۳،۳۰،۳۱،۳۳،۳۵،۳۶،۳۷،۳۸،۴۰،۴۳	۲۲
	BPSO	۲،۷،۸،۱۲،۱۳،۱۶،۱۷،۱۹،۲۰،۲۲،۲۳،۲۴، ۲۵،۲۷،۳۰،۳۳،۳۵،۳۶،۳۹،۴۲	۲۰
	BBAT H	۳،۴،۵،۶،۷،۸،۱۰،۱۱،۱۲،۱۴،۱۶،۱۷،۲۲،۲ ۳،۲۶،۲۷،۲۸،۳۱،۳۲،۳۶،۳۷،۳۹،۴۲	۲۳
	BHHO	۳،۴،۵،۶،۷،۹،۱۰،۱۱،۱۲،۱۳،۱۴،۱۶،۱۷،۱ ۸،۲۰،۲۱،۲۳،۲۴،۲۶،۲۷،۲۸،۲۹،۳۰،۳۲،۳ ۵،۳۷،۳۸،۳۹،۴۲،۴۳	۳۰
	BCHO	**۳،۹،۱۸،۲۰،۲۱،۲۶،۲۷،۳۰،۳۱،۳۳،۳۴، ۳۷،۴۰،۴۱،۴۲،۴۳**	**۱۶**
مجموعه داده IoTID20(۸۳ ویژگی)	BPI	۱،۲،۵،۶،۸،۱۰،۱۳،۱۴،۱۸،۲۲،۲۳،۲۵،۲۶، ۲۷،۲۸،۲۹،۳۳،۳۵،۳۶،۳۷،۴۱،۴۴،۴۷،۴۹، ۵۱،۵۲،۵۳،۵۴،۵۵،۵۷،۵۸،۶۰،۶۳،۶۴،۶۵، ۶۹،۷۰،۷۲،۷۴،۷۵،۷۶،۸۱،۸۲	۴۲
	BPSO	۴،۶،۷،۹،۱۰،۱۱،۱۲،۱۳،۱۴،۱۵،۱۶،۱۷،۲ ۲،۲۷،۳۱،۳۲،۳۸،۳۹،۴۱،۴۵،۴۶،۴۷،۴۸،۵ ۲،۵۷،۵۸،۶۲،۶۳،۶۵،۶۶،۶۹،۷۰،۷۲،۷۸،۸ ۰،۸۲	۳۵
	BBATH	۱،۴،۵،۶،۷،۱۰،۱۱،۱۲،۱۳،۱۴،۱۶،۱۷،۲۰، ۲۱،۲۲،۲۴،۲۵،۲۶،۲۷،۲۸،۲۹،۳۴،۳۵،۳۷، ۴۵،۴۸،۵۰،۵۳،۵۴،۵۹،۶۰،۶۲،۶۴،۶۶،۶۸، ۶۹،۷۰،۷۱،۷۳،۷۴،۷۷،۷۹،۸۲	۴۲
	BHHO	۱،۳،۴،۷،۱۰،۱۲،۱۴،۱۵،۱۶،۱۸،۱۹،۲۱،۲ ۲،۲۳،۲۵،۲۹،۳۱،۳۲،۳۳،۳۵،۳۷،۳۸،۳۹،۴ ۰،۴۱،۴۲،۴۴،۴۶،۴۷،۵۰،۵۲،۵۵،۵۶،۵۷،۶ ۰،۶۱،۶۲،۶۳،۶۴،۶۵،۶۶،۷۲،۷۴،۷۵،۷۶،۷ ۷،۷۸،۸۰،۸۱	۵۲
	BCHO	**۲،۶،۷،۸،۹،۱۰،۱۶،۱۹،۲۳،۲۵،۲۶،۲۷،۲ ۹،۳۱،۳۵،۳۸،۴۳،۴۶،۵۰،۵۵،۵۷،۵۸، ۶۵،۶۷،۷۱،۷۲،۷۴،۷۷،۷۹**	۳۰

نسبت به الگوریتم‌های دیگر دقت بیشتری داشته است. در K=۵ این الگوریتم به بیشترین دقت دست‌یافته است. همچنین در مورد مجموعه داده UNSWNB-15 در شکل (۴) نیز مدل پیشنهادی در بیشتر تکرارها دقت بیشتری نسبت به الگوریتم‌های دیگر از خود نشان داده است و در K=۵ به بیشترین دقت دست‌یافته است. این الگوریتم بدون انتخاب ویژگی به‌دقت ۹۹/۶۰ دست‌یافته است که نشان‌دهنده برتری مدل پیشنهادی همراه با ویژگی‌های انتخابی BCHO است. در معیارهای دیگر نیز مدل پیشنهادی همراه با ویژگی‌های انتخابی BCHO عملکرد بسیاری بالایی داشته است. در نهایت، در مجموعه داده IoTID20 در شکل(۵) نیز مشاهده می‌شود که مدل پیشنهادی در بیشتر تکرارها نسبت به الگوریتم‌های دیگر دقت بیشتری داشته است و در K=۵ و K=۷ بیشترین دقت را داشته است. در K=۷ این الگوریتم بدون انتخاب ویژگی به‌دقت ۹۸/۱۳ دست‌یافته است که نشان‌دهنده برتری مدل پیشنهادی همراه با ویژگی‌های انتخابی BCHO است. همچنین در معیارهای دیگر نیز مدل پیشنهادی همراه با ویژگی‌های انتخابی BCHO عملکردی بسیاری بالایی داشته است. نتایج k-Fold ثابت کرد که ویژگی‌های انتخابی BCHO دارای عملکرد بالا و پایدار خوبی نسبت به روش‌های دیگر مخصوصاً در K=۵ در هر سه مجموعه داده است.

نتایج با ویژگی‌های انتخابی BCHO بر روی‌داده‌های تست نشان می‌دهد که در مجموعه داده ToN-IoT به‌دقت ۹۹ درصد بدون انتخاب ویژگی و ۹۹/۳۰ درصد با انتخاب ویژگی، در مجموعه داده UNSWNB15 به‌دقت۹۷/۶درصد بدون انتخاب ویژگی و۹۹/۹۶ درصد با انتخاب ویژگی و در مجموعه داده IoTID20 به‌دقت۹۸ درصد بدون انتخاب ویژگی و۹۹/۹ درصد با انتخاب ویژگی دست‌یافته است. همچنین در معیار امتیاز f1 در هر سه مجموعه داده، مدل پیشنهادی با ویژگی های انتخابی BCHO به طور میانگین به ۹۹ درصد دست یافته است که در مقایسه با دیگر الگوریتم های انتخاب ویژگی مورد مقایسه بهترین عملکرد را دارد. نمودار شکل(۶)این مقایسه را گویاتر نشان می دهد.

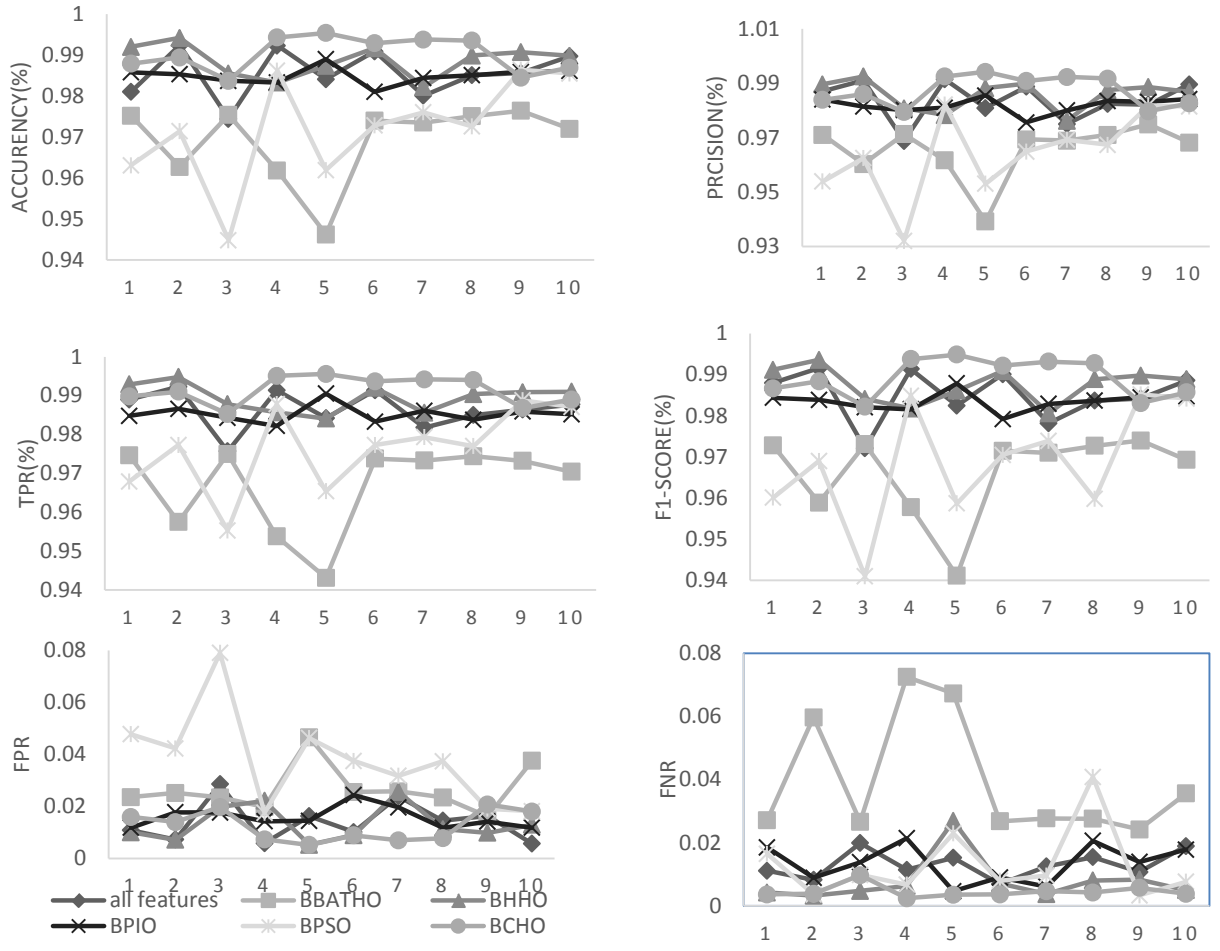
یکی از معیارهایی که در تشخیص حملات مهم است معیار تشخیص مثبت کاذب و نرخ هشدار کاذب است. با توجه به نتایج شکل (۷)روش پیشنهادی دارای کمترین مقدار نرخ هشدار کاذب در حملات است.

جدول (۶) نشان دهنده مقایسه زمانی بین الگوریتم پیشنهادی و الگوریتم های دیگر است. بر اساس نتایج این جدول، مشخص است که به دلیل تعداد کمتر ویژگی ها در هر سه مجموعه داده، الگوریتم bcho زمان کمتری برای آموزش و تست نیاز دارد.

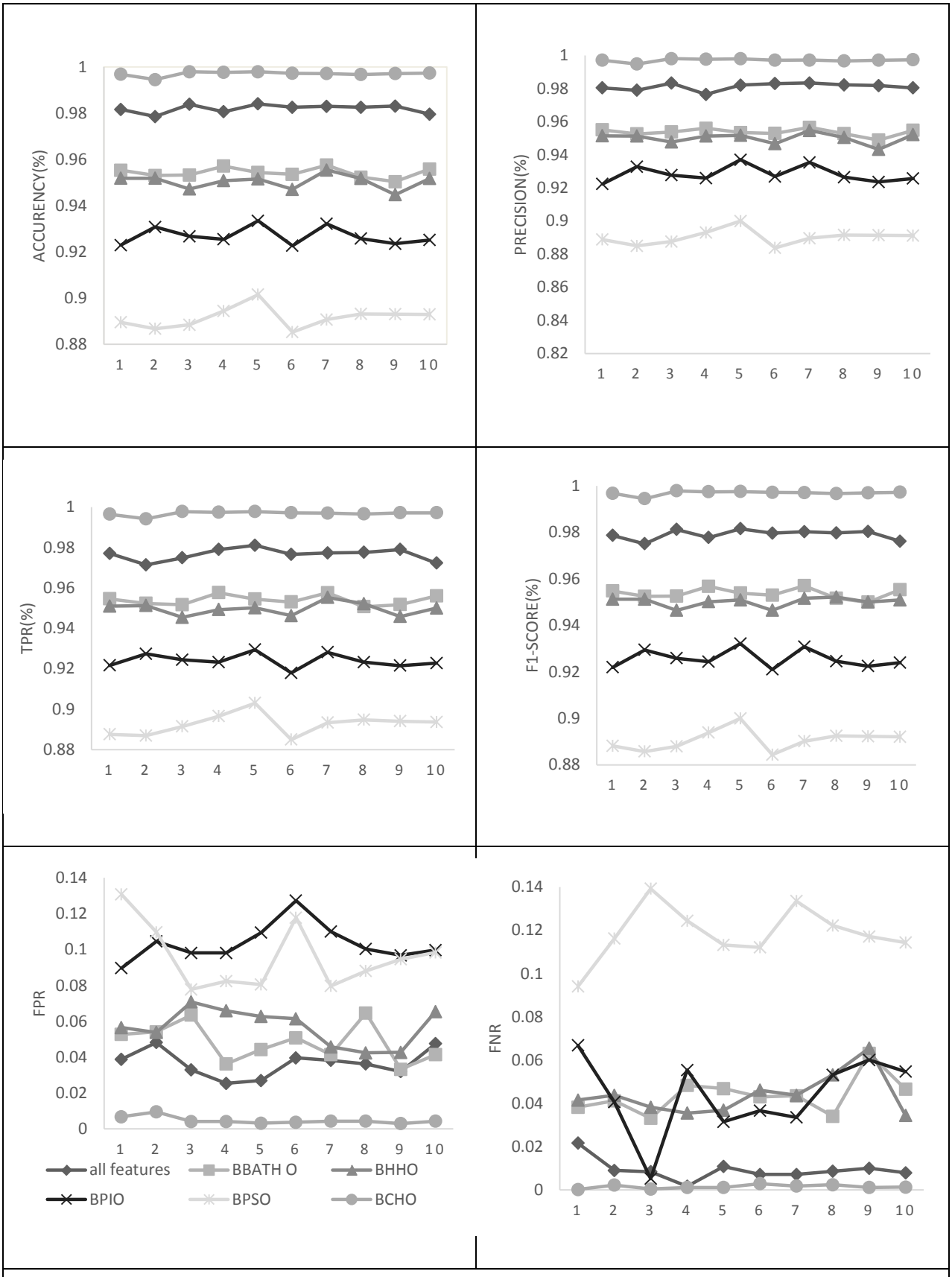
نتایج تکنیک K-Fold بر روی مجموعه داده ToN-IoT در شکل (۳) نشان می‌دهد که مدل پیشنهادی در بیشتر تکرارها

جدول(۶) مقایسه زمانی روش پیشنهادی در زمان آموزش و تست

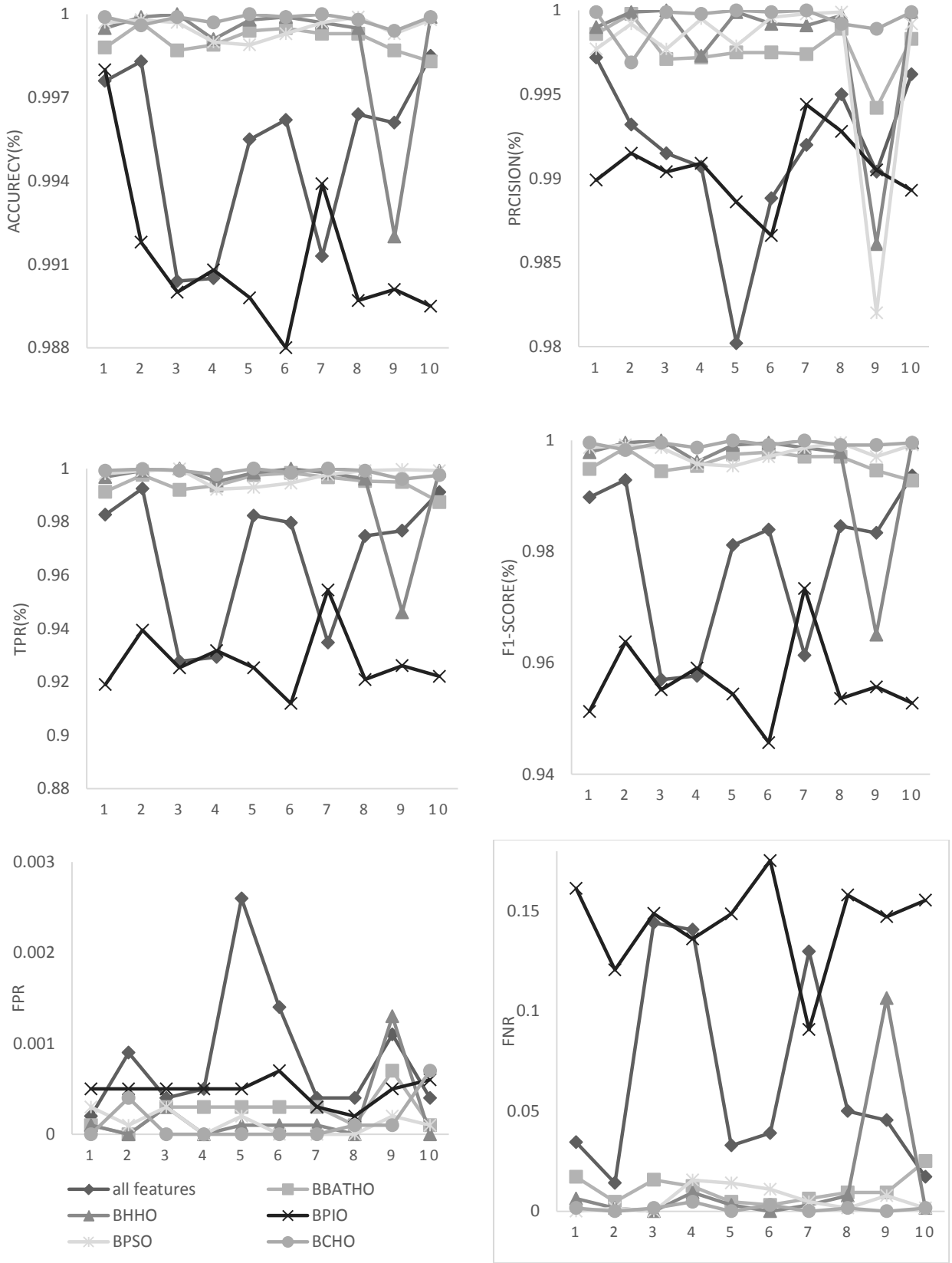
داده	الگوریتم	زمان آموزش(s)	زمان تست(ms)
مجموعه داده ToN-IoT(۴۳ ویژگی)	تمام ویژگی ها	۴۷/۴۹	۱۱
	BPI	۲۸/۳	۵
	BPSO	۴۰/۰۰	۹
	BBATH	۳۷/۵۴	۸
	BHHO	۲۹/۶۲	۵
	BCHO	**۲۷/۰۹**	**۴**
مجموعه داده UNSWNB-15(۴۳ ویژگی)	تمام ویژگی ها	۴۴/۲۹	۱۳
	BPI	۲۸/۹	۵
	BPSO	۴۰/۱۰	۹
	BBAT H	۳۸/۴	۹
	BHHO	۳۰/۲	۶
	BCHO	**۲۶/۹**	**۴**
مجموعه داده IoTID20(۸۳ ویژگی)	تمام ویژگی ها	۹۱	۲۵
مجموعه داده IoTID20(۸۳ ویژگی)	BPI	۵۸/۶	۱۰
	BPSO	۷۳/۲۰	۱۷
	BBATH	۶۹/۸۴	۱۶
	BHHO	۶۴/۶۲	۱۱
	BCHO	**۴۲/۶۹**	**۹**



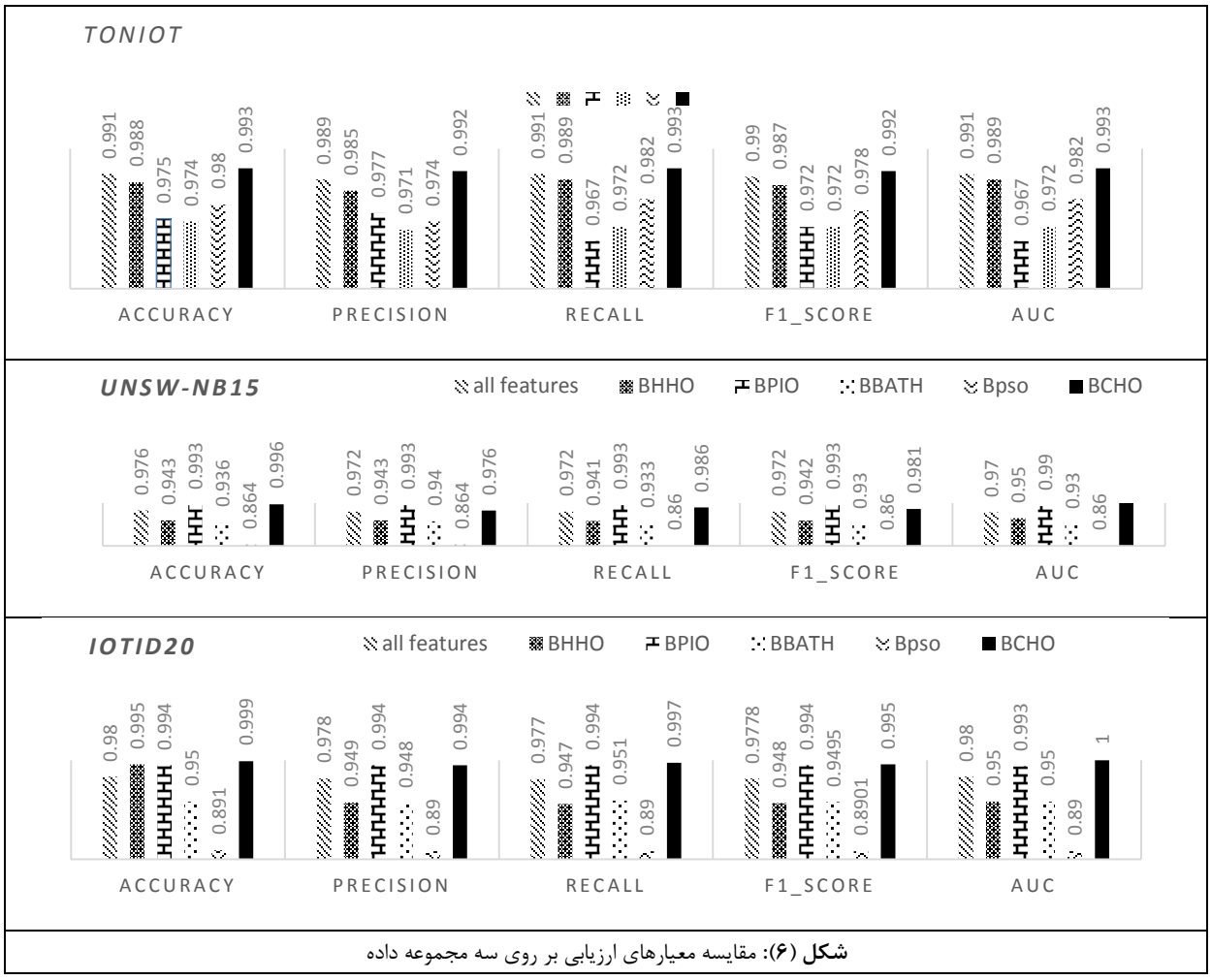
شکل (۳) نتیجهK-FOLD بر روی مجموعه داده TONIOT



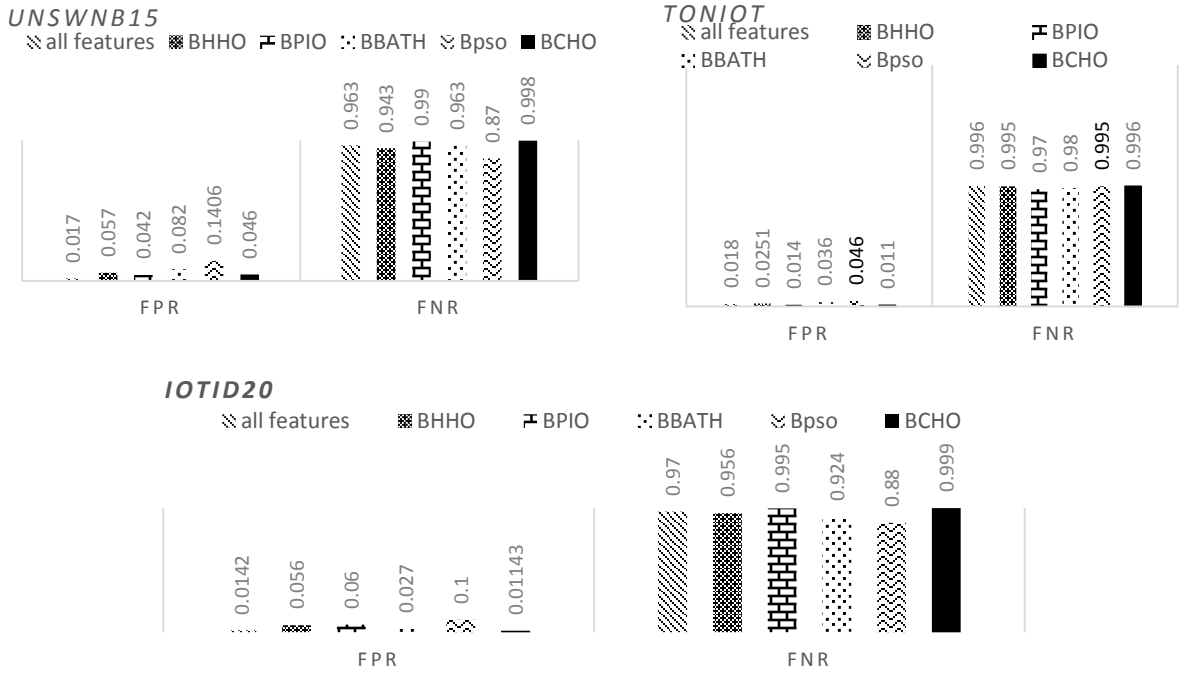
شکل (۴) نتیجهK-FOLD بر روی مجموعه داده UNSWNB15



شکل (۵) نتیجهK-FOLD بر روی مجموعه داده IOTID20



شکل (۶): مقایسه معیارهای ارزیابی بر روی سه مجموعه داده



شکل (۷): مقایسه معیارهای FPR,FNR بر روی سه مجموعه داده

۴ -۵ نتایج آماری

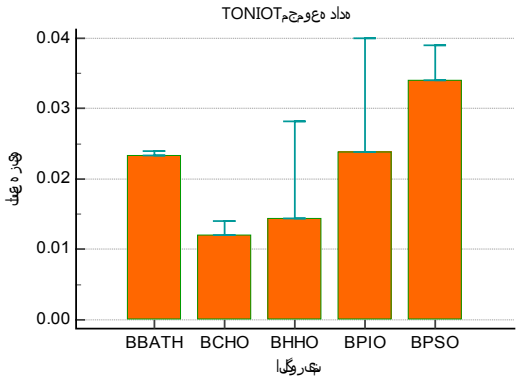
برای اثبات اینکه BCHO ازنظر همگرایی زودتر از سایر الگوریتم‌های مورد مقایسه همگرا شده است، تجزیه‌وتحلیل آماری انجام‌شده است. ابتدا بررسی شد که آیا توزیع داده‌های تابع هزینه توزیعی نرمال است یا نه. به ازای هر الگوریتم، تابع هزینه ۱۰۰ بار محاسبه شد. با استفاده از آزمون شاپیرو-ویلک مشخص شد که توزیع تابع هزینه توزیعی نرمال نیست. از مقدار p که کمتر از ۰/۰۰۰۱ است می‌توان نتیجه گرفت که داده‌ها نرمال نیستند و فرض صفر رد می‌شود. نتایج آزمون شاپیرو-ویلک بر روی مجموعه داده TONIOT در جدول (۷)نشان داده‌شده است

جدول (۷) نتایج آزمون شاپیرو-ویلک بر روی مجموعه داده TONIOT

Variable	fitness
size	500
Lowest value	0.01200
Highest value	0.04000
Arithmetic mean	0.02271
95% CI for the Arithmetic mean	0.02197 to 0.02345
Median	0.02340
95% CI for the median	0.02340 to 0.02370
Variance	0.0000716
Standard deviation	0.008461
Relative standard deviation	3726 (37.26%)
Standard error of the mean	0.0003784
Coefficient of Skewness	0.3038 (P=0.0060)
Coefficient of Kurtosis	-1.0070 (P<0.0001)
Shapiro-Wilk test for Normal distribution	W=0.8782 reject Normality (P<0.0001)

Percentiles		95% Confidence interval
2.5	0.01200	0.01200 to 0.01200
5	0.01200	0.01200 to 0.01200
10	0.01200	0.01200 to 0.01200
25	0.01440	0.01440 to 0.01440
75	0.02560	0.02390 to 0.03400
90	0.03400	0.03400 to 0.03400
95	0.03500	0.03400 to 0.04000
97.5	0.04000	0.03875 to 0.04000

با توجه به نرمال نبودن توزیع توابع هزینه، از تحلیل آماری کروسکال-والیس برای بررسی همگرایی الگوریتم‌های فرا ابتکاری استفاده‌شده است. نتایج تحلیل آماری در نمودار شکل(۸) نشان می‌دهد که به‌عنوان نمونه در مجموعه داده TON-IOT، مقدار تابع هدف در روش BCHO از بقیه روش‌ها کمتر است و زودتر همگرا می‌شود.



شکل(۸): نتایج آزمون کروسکال-والیس بر روی مجموعه داده TONIOT

۵. نتیجه‌گیری

در این مقاله، یک چهارچوب انتخاب ویژگی بر اساس یادگیری عمیق ارائه‌شده است که هشدارهای نادرست را تشخیص داده و کارایی سیستم را افزایش می‌دهد. نوآوری اصلی روش فرا ابتکاری پیشنهادی را می‌توان به‌صورت خلاصه در سه مرحله شرح داد. این مراحل شامل افزایش اکتشاف با استفاده از عملگرهای جهش در مرحله اول، افزایش بهره‌وری با استفاده از عملگرهای ترکیب در مرحله دوم و فرار از دام محلی با استفاده از توابع جستجوی محلی در مرحله سوم می‌شود. با استفاده از این روش، مشکل همگرایی کند حل‌شده و یک سیستم تشخیص نفوذ بر اساس کاهش ویژگی‌های مجموعه داده‌ها طراحی‌شده است. این سیستم با استفاده از مجموعه داده‌های IToN-IoT، UNSWNB15 و IoTID20 و مقایسه الگوریتم‌های بهینه‌سازی انتخاب ویژگی مختلف، ازجمله کبوتر، ازدحام ذرات بهینه‌شده، خفاش دودویی بهینه‌شده و شاهین هریس دودویی، پیاده‌سازی شده است. نتایج تجربی نشان می‌دهد که روش پیشنهادی دارای دقت بیشتر و هشدارهای نادرست کمتری برای تشخیص حملات سایبری است.

برای بررسی تفاوت آماری بین الگوریتم پیشنهادی و دیگر الگوریتم‌های قابل‌مقایسه، آزمون کروسکال-والیس استفاده‌شده نشان داد روش پیشنهادی دارای همگرایی سریع‌تر نسبت به روش‌های مورد مقایسه است. به‌عنوان راهکار آینده می‌توان مدل یادگیری عمیق را تقویت کرد و بر روی تشخیص دومرحله‌ای برای تشخیص نوع حملات نیز عملکرد را ارتقاء داد.

۶. مراجع

[1] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning based approach," Expert Systems with Applications, vol. 238, p. 121751, 2024.

[2] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Anomaly based network intrusion detection for IoT attacks using deep learning technique," Computers and Electrical Engineering, vol. 107, p. 108626, 2023.

[3] M. J. Idrissi, H. Alami, A. El Mahdaouy, A. El Mekki, S. Oualil, Z. Yartaoui, et al., "Fed-anids: Federated learning for anomaly-based network intrusion detection systems," Expert Systems with Applications, vol. 234, p. 121000, 2023.

[4] A. S. Dina, A. Siddique, and D. Manivannan, "A deep learning approach for intrusion detection in Internet of Things using focal loss function," Internet of Things, vol. 22, p. 100699, 2023.

[5] S. A. Khanday, H. Fatima, and N. Rakesh, "Implementation of intrusion detection model for DDoS attacks in Lightweight IoT Networks," Expert Systems with Applications, vol. 215, p. 119330, 2023.

[6] S. Khan and A. B. Mailewa, "Discover botnets in IoT sensor networks: A lightweight deep learning framework with hybrid self-organizing maps," Microprocessors and Microsystems, vol. 97, p. 104753, 2023.

[7] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," Computer Communications, vol. 199, pp. 113-125, 2023.

[8] H. Asgharzadeh, A. Ghaffari, M. Masdari, and F. S. Gharehchopogh, "Anomaly-based intrusion detection system in the Internet of Things using a convolutional neural network and multi-objective enhanced Capuchin Search Algorithm," Journal of Parallel and Distributed Computing, vol. 175, pp. 1-21, 2023.

[9] S. Wang, W. Xu, and Y. Liu, "Res-TranBiLSTM: An intelligent approach for intrusion detection in the Internet of Things," Computer Networks, vol. 235, p. 109982, 2023.

[10] R. Lazzarini, H. Tianfield, and V. Charissis, "A stacking ensemble of deep learning models for IoT intrusion detection," Knowledge-Based Systems, vol. 279, p. 110941, 2023.

[11] H. Asgharzadeh, A. Ghaffari, M. Masdari, and F. S. Gharehchopogh, "An Intrusion Detection System on The Internet of Things Using Deep Learning and Multi-objective Enhanced Gorilla Troops Optimizer, "Journal of Bionic Engineering, 2024/07/09 2024.

[12] !!!INVALID CITATION !!! [Khojand, 2024 #69].

[13] V. Hosseini, Y. Farhang, K. Majidzadeh, and C. Ghobadi, "Customized mutated PSO algorithm of isolation enhancement for printed MIMO antenna with ISM band applications," AEU-International Journal of Electronics and Communications, vol. 145, p. 154067, 2022.

[14] S. H. S. Ebrahimi, K. Majidzadeh, and F. S. Gharehchopogh, "A principal label space transformation and ridge regression-based hybrid gorilla troops optimization and jellyfish search algorithm for multi-label classification," Cluster Computing, pp. 1-45, 2024.

[15] H. Tanha and M. Abbasi, "Identify malicious traffic on IoT infrastructure using neural networks and deep learning," Electronic and Cyber Defense, vol. 11, pp. 1-13, 2023.

[16] J. Mazloum and H. Bigdeli, "An Optimized Compound Deep Neural Network Integrating With Feature Selection for Intrusion Detection System in Cyber Attacks," Electronic and Cyber Defense, vol. 10, pp. 41-51, 2023.

[17] A. Karimi, M. Irajimoghaddam, and E. Bastami, "Feature selection using combination of genetic-whale-ant colony algorithms for software fault prediction by machine learning," Electr Cyber Defense, vol. 10, 2022.

[18] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," Expert systems with applications, vol. 148, p. 113249, 2020.

[19] M. H. Aghdam and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization," Int. J. Netw. Secur., vol. 18, pp. 420-432, 2016.

[20] W. Ghanem and A. Jantan, "Novel multi-objective artificial bee colony optimization for wrapper based feature selection in intrusion detection," Int. J. Adv. Soft Comput. Appl, vol. 8, pp. 70-81, 2016.

[21] N. Farnaaz and M. Jabbar, "Random forest modeling for network intrusion detection system," Procedia Computer Science, vol. 89, pp. 213-217, 2016.

[22] N. Acharya and S. Singh, "An IWD-based feature selection method for intrusion detection system," Soft Computing, vol. 22, pp. 4407-4416, 2018.

[23] B. Selvakumar and K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," Computers & Security, vol. 81, pp. 148-155, 2019.

[24] Q. M. Alzubi ,M. Anbar, Z. N. Alqattan, M. A. Al-Betar, and R. Abdullah,

"Intrusion detection system based on a modified binary grey wolf optimisation," Neural computing and applications, vol. 32, pp. 6125-6137, 2020.

[25] O. A. Alghanam, W. Almobaideen, M. Saadeh, and O. Adwan, "An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning," Expert Systems with Applications, vol. 213, p. 118745, 2023.

[26] N. Kunhare, R. Tiwari, and J. Dhar, "Intrusion detection system using hybrid classifiers with meta-heuristic algorithms for the optimization and feature selection by genetic algorithm," Computers and Electrical Engineering, vol. 103, p. 108383, 2022.

[27] S. V. Pingale and S. R. Sutar, "Remora whale optimization-based hybrid deep learning for network intrusion detection using CNN features," Expert Systems with Applications, vol. 210, p. 118476, 2022.

[28] P. K. Keserwani, M. C. Govil, E. S. Pilli, and P. 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, vol. 7, pp. 3-21, 2021.

[29] W. Elmasry, A. Akbulut, and A. H. Zaim, "Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic," Computer Networks, vol. 168, p. 107042, 2020.

[30] A. Syarif and W. Gata, "Intrusion detection system using hybrid binary PSO and K-nearest neighborhood algorithm. 11 th Int Conf on Information & Communication Technology and System," ed, 2017.

[31] K. Anusha and E. Sathiyamoorthy, "A decision tree-based rule formation with combined PSO-GA algorithm for intrusion detection system," International Journal of Internet Technology and Secured Transactions, vol. 6, pp. 186-202, 2016.

[32] S. M. H. Bamakan, H. Wang, T. Yingjie, and Y. Shi, "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization," Neurocomputing, vol. 199, pp. 90-102, 2016.

[33] Z. Wang, M. Tang, J. Deng ,Y. Wang, J. Qian, and X. Chen, "A new feature selection method for intrusion detection," in 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS), 2019, pp. 298-304.

[34] I. Syarif, R. F. Afandi, and F. A. Saputra, "Feature selection algorithm for intrusion detection using cuckoo search algorithm," in 2020 International Electronics Symposium (IES), 2020, pp. 430-43.۵

[35] E. Roopa Devi and R. Suganthe, "Enhanced transductive support vector machine classification with grey wolf optimizer cuckoo search optimization for intrusion detection system," Concurrency and Computation: Practice and Experience, vol. 32, p. e499,2020. 9

[36] S. Sarvari, N. F. M. Sani, Z. M. Hanapi, and M. T. Abdullah, "An efficient anomaly intrusion detection method with feature selection and evolutionary neural network," IEEE Access, vol. 8, pp. 70651-70663, 2020.

[37] T. Gu, H. Chen, L. Chang, and L. Li, "Intrusion detection system based on improved abc algorithm with tabu search," IEEJ Transactions on Electrical and Electronic Engineering, vol. 14, pp. 1652-1660, 2019.

[38] L. Li, S. Zhang, Y. Zhang, L. Chang, and T. Gu, "The intrusion detection model based on parallel multi-artificial bee colony and support vector machine," in 2019 Eleventh International Conference on Advanced Computational Intelligence (ICACI), 2019, pp. 308-313.

[39] M. Mazini, B. Shirazi, and I. Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms," Journal of King Saud University-Computer and Information Sciences, vol. 31, pp. 541-553, 2019.

[40] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset," IEEE Access, vol. 8, pp. 29575-29585, 2020.

[41] G. M. Suresh and M. L. Madhavu, "AI based intrusion detection system using self-adaptive energy efficient BAT algorithm for software defined IoT networks," in 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-6.

[42] M. Safaldin, M. Otair, and L. Abualigah, "Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks," Journal of ambient intelligence and humanized computing, vol. 12, pp. 1559-1576, 2021.

[43] P. K. Keserwani, M. C. Govil, and E. S. Pilli, "An optimal intrusion detection system using GWO-CSA-DSAE model," Cyber-Physical Systems, vol. 7, pp. 197-220, 2021.

[44] T. A. Alamiedy, M. Anbar, Z. N. Alqattan, and Q. M. Alzubi, "Anomaly-based intrusion detection system using multi-objective grey wolf optimisation algorithm," Journal of Ambient Intelligence and Humanized Computing, vol. 11, pp. 3735-3756, 2020.

[45] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," Journal of information security and applications, vol. 44, pp. 80-88, 201.۹

[46] S. Dwivedi, M. Vardhan, and S. Tripathi, "Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection," Cluster Computing, pp. 1-20, 2021.

[47] N. O. Aljehane, H. A. Mengash, M. M. Eltahir, F. A. Alotaibi, S. S. Aljameel, A. Yafoz, et al., "Golden jackal optimization algorithm with deep learning assisted intrusion detection system for network security," Alexandria Engineering Journal, vol. 86, pp. 415-424, 2024.

[48] P. Zhou, H. Zhang, and W. Liang, "Research on hybrid intrusion detection based on improved Harris Hawk optimization algorithm," Connection Science, vol. 35, p. 2195595, 2023.

[49] O. Pandithurai, C. Venkataiah, S. Tiwari, and N. Ramanjaneyulu, "DDoS attack prediction using a honey badger optimization algorithm based feature selection and Bi-LSTM in cloud environment," Expert Systems with Applications, vol. 241, p. 122544, 2024.

[50] Z. Ye, J. Luo, W. Zhou, M. Wang, and Q. He, "An ensemble framework with improved hybrid breeding optimization-based feature selection for intrusion detection," Future Generation Computer Systems, vol. 151, pp. 124-136, 2024.

[51] M. Sharma and P. Kaur, "A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem," Archives of Computational Methods in Engineering, vol. 28, pp. 1103-1127, 2021.

[52] A. Shaddeli, F. S. Gharehchopogh, M. Masdari, and V. Solouk, "BFRA: a new binary hyper-heuristics feature ranks algorithm for feature selection in high-dimensional classification data," International Journal of Information Technology & Decision Making, vol. 22, pp. 471-536, 2023.

[53] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm," Expert systems with applications, vol. 149, p. 113338, 2020.

[54] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," Computer networks, vol. 174, p. 107247, 2020.

[55] J. Too, A. R. Abdullah, and N. Mohd Saad, "A new quadratic binary harris hawk optimization for feature selection," Electronics, vol. 8, p. 1130, 2019.

[56] P. Diaz and M. J. E. Jiju, "A comparative analysis of meta-heuristic optimization algorithms for feature selection and feature weighting in neural networks," Evolutionary Intelligence, vol. 15, pp. 2631-2650, 2022.

[57]