سامانه‌های پردازشی و ارتباطی چندرسانه‌ای هوشمند
Intelligent Multimedia Processing and Communication Systems (IMPCS)

Fall 2023, 4 (3), 63-92
DOI: 10.71856/impcs.2023.1123897

Received: 22 July 2023
Accepted: 7 Sep 2023

مقاله مروری

Metaheuristic Algorithms for Feature Selection in Intrusion Detection Systems: A Systematic Review

Yashar Pourardbilkhah [1], Mirsaeid Hosseini Shirvani [2]*, Homayun Motameni [3]

1. PhD student, Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran. yashar.uni2010@gmail.com
2. Assistant Professor, Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran. mirsaeid_hosseini@iausari.ac.ir
3. Professor, Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran. motameni@iausari.ac.ir

Abstract

Introduction: Considering the increasing importance of network security in complex and dynamic environments, intrusion detection systems (IDS) play a very important role in identifying and dealing with security threats. However, the presence of a large amount of data in networks affects the efficiency of intrusion detection systems. Feature selection as a critical step in data preprocessing can help improve the accuracy, speed, and efficiency of these systems. This article deals with the systematic review of feature selection methods based on meta-heuristic algorithms in intrusion detection systems in the cloud computing environment.

Method: The methodology of this article includes a comprehensive review of the research conducted in the field of feature selection for IDS. In this review, meta-heuristic algorithms such as genetic algorithm, particle swarm optimization (PSO), bee colony optimization, bat algorithm, and other nature-inspired optimization methods are thoroughly reviewed. These algorithms are chosen due to their ability to search a large space of features and identify the best combinations to improve the performance of IDSs.

Evaluation: In this paper, detailed comparisons have been made between different algorithms in terms of performance criteria such as detection accuracy, false alarm rate, processing time and the number of selected features. Also, different data sets that have been used in this field have been discussed to evaluate the efficiency of each of the methods in different conditions.

Challenge: In addition, key challenges in this field have been identified and analyzed. These challenges include things such as high computational complexity, the problem of processing overhead in cloud environments, the balance between accuracy and detection speed, and the problem of feature interference. Also, research gaps in this field that require further research have been identified.

Keywords: Feature selection, meta-heuristic algorithm, intrusion detection system, cloud computing, optimization, network security.

سامانه‌های پردازشی و ارتباطی چندرسانه‌ای هوشمند
Intelligent Multimedia Processing and Communication Systems (IMPCS)
انجمن علمی تجارت الکترونیکی ایران
واحد زنجان

الگوریتم‌های فراابتکاری برای انتخاب ویژگی در سیستم‌های تشخیص نفوذ: یک بررسی سیستماتیک

تاریخ دریافت: ۳۱/ ۰٤/ ۱٤۰۲
تاریخ پذیرش: ۱٦/ ۰٦/ ۱٤۰۲

دورهٔ چهارم، پاییز ۱٤۰۲
شمارهٔ سوم، صص: ٦۳-۹۲

یاشار پوراردبیل‌خواه [۱]، میرسعید حسینی‌شیروانی [۲*]، همایون مؤتمنی [۳]

۱. دانشجوی دکتری، گروه مهندسی کامپیوتر، دانشکده فنی مهندسی، دانشگاه آزاد اسلامی، ساری، ایران
yashar.uni2010@gmail.com
۲. استادیار، گروه مهندسی کامپیوتر، واحد ساری، دانشگاه آزاد اسلامی، ساری، ایران
mirsaeid_hosseini@iausari.ac.ir
۳. استاد، گروه مهندسی کامپیوتر، واحد ساری، دانشگاه آزاد اسلامی، ساری، ایران
motameni@iausari.ac.ir

چکیده:

با توجه به اهمیت روزافزون امنیت شبکه در محیط‌های پیچیده و پویا، سیستم‌های تشخیص نفوذ (IDS) نقش بسیار مهمی در شناسایی و مقابله با تهدیدات امنیتی ایفامی‌کنند. با این حال، وجود حجم زیادی از داده‌ها در شبکه‌ها، کارایی سیستم‌های تشخیص نفوذ را تحت تأثیر قرارمی‌دهد. انتخاب ویژگی به عنوان یک مرحله حیاتی در پیش‌پردازش داده‌ها می‌تواند به بهبود دقت، سرعت و کارایی این سیستم‌ها کمک‌کند. این مقاله به بررسی سیستماتیک روش‌های انتخاب ویژگی مبتنی بر الگوریتم‌های فراابتکاری در سیستم‌های تشخیص نفوذ در محیط رایانش ابری می‌پردازد. روش کار شامل مرور جامعی بر تحقیقات انجام‌شده در حوزه انتخاب ویژگی برای IDS ها است. در این بررسی، الگوریتم‌های فراابتکاری مانند الگوریتم ژنتیک، بهینه سازی ازدحام ذرات (PSO)، بهینه سازی کلونی زنبور، الگوریتم خفاش، و سایر روش‌های بهینه سازی الهام‌گرفته از طبیعت، به‌طور کامل بررسی‌شده‌اند. این الگوریتم‌ها به‌دلیل توانایی‌شان در جستجوی فضای بزرگ ویژگی‌ها و شناسایی بهترین ترکیب‌ها برای بهبود عملکرد IDS ها، انتخاب‌شده‌اند. در این مقاله، مقایسه‌های دقیقی بین الگوریتم‌های مختلف از نظر معیارهای عملکردی همچون دقت تشخیص، نرخ هشدار غلط، زمان پردازش و تعداد ویژگی‌های انتخاب‌شده انجام‌شده‌است. همچنین، مجموعه داده‌های مختلفی در این زمینه، مورد بحث و بررسی قرار گرفته‌اند تا کارایی هر یک از روش‌ها در شرایط مختلف ارزیابی‌شود. علاوه بر این، چالش‌های کلیدی در این حوزه مورد شناسایی و تحلیل قرار گرفته‌اند. این چالش‌ها شامل مواردی همچون پیچیدگی محاسباتی بالا، مسئله سربار پردازشی در محیط‌های ابری، تعادل بین دقت و سرعت تشخیص، و مشکل تداخل ویژگی‌ها هستند. همچنین، شکاف‌های تحقیقاتی موجود در این زمینه که نیازمند تحقیقات بیشتر است، شناسایی شده‌اند. نتایج این بررسی نشان‌می‌دهد که هریک از الگوریتم‌های فراابتکاری دارای نقاط قوت و ضعف خاص خود هستند و انتخاب بهترین روش بستگی به نیازمندی‌ها و شرایط خاص سیستم دارد. این مقاله با ارائه توصیه‌هایی برای تحقیقات آینده، به‌عنوان یک راهنمای جامع برای پژوهشگران و توسعه‌دهندگان سیستم‌های تشخیص نفوذ در محیط‌های ابری مطرح‌می‌شود.

واژه‌های کلیدی: انتخاب ویژگی، الگوریتم فراابتکاری، سیستم تشخیص نفوذ، رایانش ابری، بهینه‌سازی، امنیت شبکه

۱. مقدمه

امروزه، سازمان‌ها به‌منظور کاهش هزینه‌های سراسری کسب و کار خود و دسترسی سریع به منابع ذخیره‌سازی مناسب برای داده‌ها، از سیاست برون‌سپاری منابع بهره‌می‌برند. در این راستا، ابرها به‌عنوان گزینه‌ای اصلی برای برون‌سپاری موردتوجه سازمان‌ها قرارمی‌گیرند. ابرها با ارائه ویژگی‌هایی چون ذخیره‌سازی اطلاعات و دسترسی از راه دور، برای استفاده از اینترنت طراحی‌شده‌اند. رایانش ابری از سه لایه انتزاعی تشکیل شده است که شامل لایه زیرساخت، لایه بستر، و لایه کاربردی می‌باشد. دو لایه اول مربوط به ماشین‌های مجازی (VM) و سیستم‌عامل است، درحالی‌که آخرین لایه شامل برنامه‌های کاربردی ارائه‌شده توسط ابر است [۱]. خدمات رایانش ابری به کاربران در سه مدل مختلف ارائه-می‌شود: زیرساخت به‌عنوان یک سرویس (IaaS)، بستر به‌عنوان یک سرویس (PaaS)، و نرم‌افزار به‌عنوان یک سرویس (SaaS). در مدل زیرساخت به‌عنوان یک سرویس، مدیران می‌توانند سیستم‌عامل، منابع سخت‌افزاری، و ذخیره‌سازی را کنترل یا حتی برنامه‌های کاربردی و اجزای شبکه را مستقرکنند. مدل PaaS هدف خود را در توسعه‌دهندگان قرارمی‌دهد تا برنامه‌های ایجادشده توسط کاربران را در فضای ابری مستقرکنند، درحالی‌که مدل SaaS کاربران را قادرمی‌سازد تا برنامه‌های ارائه‌دهنده را اجراکنند. معماری ابر از دو جزء متوالی تشکیل‌شده‌است: جزء اول که front-end نام‌دارد و توسط کاربر قابل‌مشاهده است (مثلاً شبکه یا کامپیوتر کاربر و برنامه‌ای که برای دسترسی به ابر از طریق رابط کاربری استفاده‌می‌کند)، و جزء دوم که backend نام‌دارد و خود ابر است و از سرورهای مختلفی تشکیل‌شده‌است [۲].

این اطلاعات نشان‌می‌دهد که استفاده از ابر به‌عنوان یک راهکار مناسب برای سازمان‌هاست، اما درعین‌حال، با چالش‌های امنیتی نیز روبرومی‌باشد که نیازمند توجه و پیشرفت در زمینه سیستم‌های امنیتی هوشمند می‌باشد. علی‌رغم مزایای بی‌شماری که ابر ارائه‌می‌دهد، موانع و تهدیداتی نیز در این حوزه وجوددارد که می‌تواند موجب نگرانی کسب و کارها و سازمان‌های مختلف شود. یکی از این موانع مسائل امنیتی است که به‌عنوان یکی از مهمترین مسائل در استفاده از ابر مطرح-می‌شود. بنابراین، ابرها نیازمند یک سیستم امنیتی هوشمند می‌باشند که به‌طور خودکار قادر به تشخیص نفوذ باشد و از آسیب‌پذیری‌های مختلف محافظت‌کند. این امر نیازمند توسعه و اجرای استانداردهای امنیتی فراگیر و مؤثر در سطح ابر و همچنین آموزش و آگاهی کاربران از روش‌های بهبود امنیت و حفاظت از داده‌ها در محیط ابر می‌باشد. همچنین، مدیران و اپراتورهای ابر باید به‌طور دائمی نگرانی‌ها و تهدیدات امنیتی را بررسی و برای پیشگیری از حملات و حفظ امنیت داده‌ها تدابیر مناسبی را اتخاذکنند.

تشخیص نفوذ (Intrusion Detection) به عنوان یکی از مهم‌ترین ابزارها در حفاظت از امنیت شبکه‌ها و سیستم‌های کامپیوتری مورد توجه قرار گرفته است. هدف اصلی این سیستم‌ها، تشخیص و پیش‌بینی حملات و نفوذهای ناخواسته به شبکه‌ها و سیستم‌های اطلاعاتی می‌باشد. تشخیص نفوذ به دو صورت نقش‌های مختلف انجام‌می‌پذیرد: تشخیص نفوذ مبتنی بر امضا (Signature-based IDS) و تشخیص نفوذ مبتنی بر رفتار (Behavior-based IDS) [۳]. در روش تشخیص نفوذ مبتنی بر امضا، الگوریتم‌ها و قوانینی براساس الگوهای مشخصی از حملات و نفوذهای قبلی تعیین‌می‌شوند و در صورت تطابق الگو، هشداری صادرمی‌شود. اما در روش تشخیص نفوذ مبتنی بر رفتار، سیستم بر اساس عملکرد و رفتار غیرمعمولی کاربران و شبکه، حملات را شناسایی‌می‌کند. سیستم‌های تشخیص نفوذ تاکنون توانسته‌اند به طور موفقیت‌آمیزی به تشخیص حملاتی چون حملات Denial of (DoS Service)، حملات با استفاده از نفوذگرهای مخرب (Malware)، حملات شنود (Sniffing) و حملات نفوذ به سیستم‌های کنترل صنعتی (ICS) پرداخته [۳] و مدیران شبکه را در جلوگیری از این حملات یاری-نموده‌اند. اما با پیشرفت فناوری و رویکردهای حمله‌کنندگان، نیاز به ارتقاء و بهبود مداوم سیستم‌های تشخیص نفوذ، احساس‌می‌شود.

سیستم تشخیص نفوذ در ابر شامل تعداد متغیرها و ویژگی‌های زیادی است که پردازش آن‌ها زمانبر است و ازآنجاکه سرعت تشخیص نفوذ در ابر بسیار بااهمیت است، نیاز به کاهش متغیرهای نامربوط و زائد است. متغیرها و ویژگی‌های نامربوط و غیرضروری، موجب کاهش سرعت و دقت سیستم تشخیص نفوذ می‌شوند بنابراین چندین تکنیک برای رسیدگی به مشکل کاهش متغیرها، توسعه داده‌شده‌است. کاهش متغیرها و ویژگی‌ها را انتخاب ویژگی گویند. انتخاب ویژگی به درک داده‌ها، کاهش نیاز محاسباتی، کاهش اثر بد ابعاد و بهبود عملکرد تشخیص کمک‌می‌کند [۳]. در این مقاله برخی از روش‌های موجود در تحقیقات گذشته بررسی‌خواهدشد که از تکنیک‌های مختلفی برای یافتن زیرمجموعه‌ای از ویژگی‌ها استفاده‌می‌کنند که موجب بهبود عملکرد کلی تشخیص نفوذمی‌شود.

روش انتخاب ویژگی، درواقع کاهش تعداد ویژگی‌های ورودی است. این روش، زیرمجموعه‌ای از متغیرهای مرتبط را انتخاب‌می‌کند تا یک مدل تشخیص صحیح با کمترین داده و بهترین دقت داشته‌باشیم. ویژگی‌های کمتر، پیچیدگی مدل را کاهش‌می‌دهند که موجب ارائه یک مدل ساده‌تر و قابل‌فهم‌تر می‌شود. روش‌های انتخاب ویژگی موجب شناسایی و حذف ویژگی‌های غیرضروری، غیرمرتبط و اضافی می‌شوند، این ویژگی‌ها تأثیری در دقت مدل پیش‌بینی ندارند و حتی ممکن است باعث کاهش دقت مدل شوند. بنابراین انتخاب ویژگی دارای سه هدف مهم است: بهبود کارایی سیستم تشخیص، افزایش سرعت تشخیص و ارائه درک بهتری از فرایند تولید داده .با استفاده از تکنیک‌های انتخاب ویژگی، می‌توان بینشی در مورد فرآیند داده‌ها به‌دست‌آورد و نیاز محاسباتی و دقت پیش‌بینی را بهبودبخشید [۳]. یکی از روش‌های انتخاب ویژگی، استفاده از الگوریتم‌های فراابتکاری است.

الگوریتم فراابتکاری (Metaheuristic Algorithm) یک روش محاسباتی است که برای حل مسائل بهینه‌سازی و تصمیم‌گیری استفاده-

می‌شود. این الگوریتم‌ها معمولاً از رویکردهای مبتنی بر طبیعت الهام-گرفته‌شده و قدرتمندترین روش‌های موجود برای حل مسائل پیچیده بهینه‌سازی محسوب‌می‌شوند [۴]. این الگوریتم‌ها برای حل مسائل به صورت تکاملی و هماهنگ استفاده می‌شوند. هدف اصلی این الگوریتم‌ها، یافتن راه‌حل‌های بهینه و تعاملی برای مسائل پیچیده است. الگوریتم‌های فراابتکاری از تکرار، تعامل بین عامل‌ها، و انطباق به محیط استفاده-می‌کنند تا به یافتن راه‌حل‌های بهتر و بهینه‌تر برای مسئله موردنظر بپردازند. این الگوریتم‌ها شامل مجموعه‌ای از قوانین و روش‌های مختلفی می‌باشند که با توجه به نوع مسئله و شرایط محیطی، مناسب‌ترین و بهینه‌ترین راه‌حل را ارائه‌می‌دهند [۵].

جستجوی تصادفی، یادگیری و بهبود ، قابلیت تطبیق از ویژگی‌های کلیدی الگوریتم‌های فراابتکاری هستند. از مزایای الگوریتم‌های فراابتکاری این است که می‌توانند مسائلی را حل‌کنند که برای الگوریتم‌های بهینه‌سازی سنتی دشوار یا غیرقابل حل هستند. همچنین این الگوریتم‌ها می‌توانند به‌طور کارآمد در فضای جستجوی بزرگ و نامحدود به‌دنبال راه‌حل بگردند و برای کارکردن نیازی به محاسبات پیچیده تابع هدف ندارند، که در مورد بسیاری از توابع غیرخطی، محاسبات، دشوار یا غیرممکن است. استفاده از الگوریتم‌های فراابتکاری در مسائل بهینه‌سازی، بهبود عملکرد سیستم‌ها و فرآیندها را به‌دنبال دارد. این الگوریتم‌ها علاوه‌بر کاربردهای بهینه‌سازی، در حوزه‌های مختلفی از جمله مسائل تصمیم‌گیری چندمعیاره (MCDM)، بهینه‌سازی مسائل شبکه‌های مخابراتی، و مسائل برنامه‌ریزی ریاضی نیز کاربرد دارند [۶].

الگوریتم‌های فراابتکاری می‌توانند به‌عنوان ابزارهای قدرتمندی برای انتخاب ویژگی در طیف گسترده‌ای از مسائل به‌کارگرفته‌شوند. الگوریتم‌های فراابتکاری می‌توانند مسائلی را حل‌کنند که برای روش‌های سنتی انتخاب ویژگی، مانند روش‌های فیلتر (filter methods) دشوار یا غیرقابل حل هستند. این الگوریتم‌ها می‌توانند با روش‌های سنتی انتخاب ویژگی ترکیب‌شوند تا نتایج بهتری حاصل‌شود [۷]. الگوریتم‌های فراابتکاری در طیف گسترده‌ای از مسائل انتخاب ویژگی به‌کارگرفته-می‌شوند، ازجمله: مسائل مربوط به پزشکی (تشخیص بیماری، پیش‌بینی بیماری، و انتخاب ژن‌های مرتبط با بیماری)، مسائل مربوط به مالی: (شخیص تقلب، پیش‌بینی بازار سهام، و مدیریت سبد سهام) و مسائل مربوط به مهندسی (طراحی سازه‌ها، فرآیندهای تولید، و سیستم‌های حمل و نقل).

۱،۱.خلاصه‌ای از مقالات مروری گذشته

به منظور آماده‌سازی این مطالعه، نتایج جستجوی اولیه شامل ۴۳۹ مقاله درباره تشخیص نفوذ از سال‌های ۲۰۱۸ تا ۲۰۲۴ می‌باشد. تعداد ۳۱۴ مقاله به‌دلیل عدم‌تناسخ با محیط ابری حذف‌شدند. ۸۰ مطالعه دیگر به‌دلیل عدم ارتباط قوی با روش‌های انتخاب ویژگی از لیست مقالات مورد مطالعه حذف‌شدند. در نهایت ۴۵ مقاله مناسب برای این پژوهش انتخاب‌شد. تعداد کمی از ادبیات فعلی به انتخاب ویژگی با استفاده از الگوریتم‌های فراابتکاری برای تشخیص نفوذ پرداخته‌اند [۸-۱۵]. ادبیات [۸، ۱۳] انوع تکنیک‌های انتخاب ویژگی برای سیستم‌های تشخیص نفوذ را بررسی‌کرده‌اند. برخی از ادبیات سیستم‌های تشخیص نفوذ مبتنی بر الگوریتم‌های فراابتکاری را مورد بررسی قراردادند [۹، ۱۲، ۱۰] همچنین ادبیات [۱۴ و ۱۶و۱۵] به بررسی الگوریتم‌های فراابتکاری در انتخاب ویژگی پرداختند.

Balasaraswathi و همکاران [۱۱] هر دو روش الگوریتم‌های فراابتکاری بهینه‌سازی زیستی و غیر زیستی را با استفاده از تکنیک‌های انتخاب ویژگی در تشخیص نفوذ شبکه‌های کامپیوتری به‌کارگرفتند.

Javier Maldonado و همکارانش [۸] مروری بر پیشرفت‌های اخیر در تکنیک‌های انتخاب ویژگی روش‌های بسته‌بند(Wrapper) برای تشخیص نفوذ ارائه‌می‌کند. هدف آن طبقه‌بندی و ارزیابی رویکردهای مختلف برای تشخیص و طبقه‌بندی حمله است. این مقاله چالش‌های تشخیص نفوذ را به‌دلیل پیچیدگی و تنوع حملات و همچنین اهمیت انتخاب ویژگی‌های کلیدی برای تشخیص مؤثر مورد بحث قرارمی‌دهد.

این بررسی یک طبقه‌بندی classification را پیشنهادمی‌کند، سناریوهای تجربی را تحلیل‌می‌کند، و چالش‌ها و مزایای تکنیک‌های موجود را برجسته‌می‌کند. همچنین جنبه‌های کلیدی خود را با سایر بررسی‌ها در ادبیات مقایسه‌می‌کند و یک نمای سازمان‌یافته از برنامه‌های انتخاب ویژگی روش‌های بسته‌بند(Wrapper) در تشخیص حمله ارائه-می‌دهد.

Shubhkirti Sharma و همکاران [۹] بررسی منظمی از الگوریتم‌های بهینه‌سازی چندهدفه برای تشخیص نفوذ در شبکه‌های اینترنت اشیاء (IoT) انجام‌داده‌اند. آن‌ها یک مقاله مروری سیستماتیک ارائه‌دادند که در آن با جمع‌آوری و تحلیل مقالات و تحقیقات پیشین در این حوزه، الگوریتم‌های بهینه‌سازی چندهدفه برای تشخیص نفوذ در شبکه‌های IoT را بررسی‌نمودند. یک بررسی جامع از پژوهش‌های پیشین در این زمینه ارائه‌شده‌است که شامل انواع الگوریتم‌های بهینه‌سازی چندهدفه، روش‌های ارزیابی، و نتایج حاصل از این تحقیقات می‌شود. اهداف متعدد برای تشخیص نفوذ در شبکه‌های IoT، مانند دقت طبقه‌بندی، نرخ تشخیص، دقت، نرخ هشدار نادرست، ویژگی و تعداد ویژگی‌ها، با فرمول‌بندی‌های ریاضی آن‌ها مورد بحث قرارمی‌گیرند. ، نقاط قوت و ضعف این الگوریتم‌ها و کاربردهای آن‌ها در شبکه‌های IoT بررسی‌می‌شود و پیشنهاداتی برای جهت‌گیری تحقیقات آینده در این زمینه ارائه‌می‌شود.

Dukka Karun Kumar Reddy و همکارانش [۱۰] از روش‌های سیستماتیک برای جمع‌آوری و تحلیل مقالات علمی و تحقیقات پیشین در زمینه استفاده از الگوریتم‌های هوش گروهی برای تشخیص نفوذ استفاده‌کرده‌اند. ابتدا یک بررسی جامع از پژوهش‌ها و مقالات پیشین در این زمینه ارائه‌شده‌است که شامل انواع الگوریتم‌های هوش گروهی مورد استفاده، روش‌های ارزیابی، و نتایج حاصل از این تحقیقات می‌شود. چالش‌ها شامل انتخاب ویژگی، انتخاب مدل ML و فقدان مجموعه

داده‌های برچسب‌گذاری شده‌است. علاوه‌بر این، پرداختن به مجموعه داده‌های نامتعادل و پیمایش بین رفتارهای عادی و غیرعادی چالش‌هایی را در ID ایجادمی‌کند .

Veeran Ranganathan Balasaraswathi و همکارانش [۱۱] تکنیک‌های انتخاب ویژگی برای سیستم‌های تشخیص نفوذ (IDS) را برای بهبود عملکرد با حذف ویژگی‌های نامربوط با استفاده از الگوریتم‌های بهینه‌سازی غیرالهام‌گرفته و زیستی مورد بحث قرارمی‌دهد. انواع مختلف IDS مانند NIDS و HIDS به همراه مراحل عملکرد IDS توضیح‌داده‌شده‌است. همچنین یک نمای کلی از FS، از جمله روش‌های بسته‌بند(Wrapper) ، فیلتر و ترکیبی ارائه‌می‌کند و نحوه استفاده از الگوریتم‌های الهام‌گرفته از زیستی مانند الگوریتم‌های ژنتیک و برنامه‌ریزی ژنتیک را برای انتخاب ویژگی مورد بحث قرارمی‌دهد. مراحل دخیل در FS با استفاده از الگوریتم‌های الهام‌گرفته از زیستی مشخص‌شده‌اند و مزایا و کاربردهای الگوریتم‌های تکاملی را در حل مؤثر مسائل پیچیده برجسته‌می‌کنند.

Rafika Saadouni و همکارانش [۱۲] مروری بر ادبیات سیستماتیک در مورد تکنیک‌های الهام‌گرفته از زیستی و یادگیری ماشین برای سیستم‌های تشخیص نفوذ در شبکه‌های IoT انجام‌دادند. آن‌ها دریافتند که ادغام الگوریتم‌های الهام‌گرفته از زیست با رویکردهای ML و DL می‌تواند عملکرد IDS را افزایش‌دهد و امنیت شبکه اینترنت اشیا را تقویت‌کند. این مطالعه ۲۵ مقاله انتخاب‌شده را مقایسه‌کرد و چالش‌ها و جهت‌گیری‌های آینده در این زمینه را شناسایی‌کرد. توصیه‌هایی برای تحریک تحقیقات و نوآوری بیشتر در امنیت اینترنت اشیا ارائه‌شد.

C. Kalimuthan و همکاران [۱۳] اهمیت امنیت شبکه و نیاز به تکنیک‌های جدید برای تشخیص نفوذ با استفاده از یادگیری ماشین را مورد بحث قرارمی‌دهند. انواع مختلف حملات شبکه، سیستم‌های تشخیص نفوذ، روش‌های انتخاب ویژگی و طبقه‌بندی‌کننده‌های یادگیری ماشین را تشریح‌می‌کند. موضوعات کلیدی شامل الگوریتم Naive Bayes، Support Vector Machine، Decision Tree و K-Nearest Neighbor است.

PRACHI AGRAWAL و همکارانش [۱۴] این مقاله مروری بر ادبیات انتخاب ویژگی با استفاده از الگوریتم‌های فراابتکاری از سال ۲۰۰۹ تا ۲۰۱۹ ارائه‌می‌کند. آن‌ها با استفاده از روش‌های سیستماتیک، ادبیات موجود در زمینه الگوریتم‌های فراابتکاری برای انتخاب ویژگی‌ها را بررسی و مورد ارزیابی قرارداده‌اند. مقاله شامل بررسی الگوریتم‌های مختلفی از جمله الگوریتم مورچه، الگوریتم ژنتیک، الگوریتم گرگ‌های خاکستری، و سایر الگوریتم‌های فراابتکاری است.

هدف اصلی این بررسی، بررسی عملکرد و کارایی این الگوریتم‌ها در فرایند انتخاب ویژگی‌ها، برای مسائل مختلف مانند تشخیص نفوذ و دیگر کاربردهای یادگیری ماشین است.

Mohamed Ziad Ali و همکاران [۱۵] چالش‌های انتخاب ویژگی در یادگیری ماشین و کاربردهای تشخیص الگو را مورد بحث قرارمی‌دهند و بر روش‌های بهینه‌سازی فراابتکاری الهام‌گرفته از طبیعت تمرکز می‌کنند. تجزیه و تحلیل دقیقی از رویکردها و استراتژی‌های مختلف در انتخاب ویژگی و همچنین تکنیک‌های مختلف بهینه‌سازی فراابتکاری پیشرفته ارائه و مزایا و معایب آن‌ها را برجسته‌می‌کنند. همچنین به مسائل و چالش‌های فعلی در انتخاب ویژگی و بهینه‌سازی فراابتکاری می‌پردازد و توصیه‌هایی برای تحقیقات آینده در این زمینه ارائه‌می‌دهد.

TIN H. PHAM و همکارانش [۱۶] الگوریتم‌های انتخاب ویژگی الهام‌گرفته از زیستی و کاربردهای آن‌ها در زمینه‌های مختلف را مورد بحث قرارمی‌دهند. آن‌ها ۳۸ مطالعه انتخاب‌شده را خلاصه‌می‌کنند و۲۱ الگوریتم الهام‌گرفته از زیست را بررسی‌می‌کنند و استفاده از الگوریتم‌های مبتنی بر ازدحام و روش‌های طبقه‌بندی نظارت‌شده را برجسته می‌کنند. آن‌ها پیشنهادمی‌کنند که تحقیقات آینده باید بر روی استفاده از انتخاب ویژگی‌های الهام‌گرفته از زیستی در زمینه‌های مختلف و کاوش تکنیک‌های بهبود، تمرکزکند. تکنیک‌های کاهش ابعاد، از جمله روش‌های خطی و غیرخطی، نیز مورد بحث قرارمی‌گیرند. در مجموع، تمام کارهای مروری گذشته، منابع خوبی هستند، اما جامع و کامل نیستند. مروری [۸] بر اساس مجموعه داده‌های استفاده‌شده، دسته-بندی‌شده‌است و معیارهای ارزیابی و توابع هدف مورد بحث قرارنگرفته-است. مطالعه [۹] به الگوریتم‌های بهینه‌سازی چندهدفه برای سیستم‌های تشخیص نفوذ توجه‌دارد و کمتر به انتخاب ویژگی پرداخته-است. توجه مطالعه [۱۰] بیشتر به الگوریتم‌های هوش جمعی برای سیستم‌های تشخیص نفوذ است و در بخش کوتاهی به انتخاب ویژگی توجه‌کرده‌است. مطالعه [۱۱] بر روی یک مجموعه داده تمرکز کرده‌است. مطالعه [۱۲] از تکنیک‌های انتخاب ویژگی مبتنی بر الگوریتم طبیعی و یادگیری ماشین برای استفاده در مسئله تشخیص نفوذ بهره‌برده‌اند. مطالعه [۱۳] به بررسی انتخاب ویژگی بر اساس یادگیری ماشین پرداخته‌است و الگوریتم‌های فراابتکاری را در نظر نگرفته‌است و فقط الگوریتم ژنتیک را بررسی‌کرده‌است. مطالعات [۱۴و ۱۵ و ۱۶] دسته-بندی خوبی درباره الگوریتم‌های فراابتکاری ارائه‌کرده‌است اما در مورد تشخیص نفوذ نیست.

۲,۱. دامنه بحث

تمرکز اصلی مرور ادبیات ما بر شناسایی تکنیک انتخاب ویژگی مبتنی بر فراابتکاری در سیستم‌های تشخیص نفوذ از سال ۲۰۱۹ تا ۲۰۲۴ است. هدف ما بررسی توابع هدف، مجموعه داده‌ها و معیارهای ارزیابی در ادبیات گذشته است. تا جایی که ما می‌دانیم هیچ مروری، توابع هدف و معیارهای ارزیابی در زمینه تشخیص نفوذ را به‌طور جامع بررسی‌نکرده-است. این کار بر روی انتخاب ویژگی مبتنی بر فراابتکاری تمرکز دارد و تأثیر آن را در حوزه تشخیص نفوذ بررسی‌می‌کند. ما در ادامه، مجموعه داده‌های مختلفی که در مطالعات آزمایش‌شده‌اند را معرفی‌می‌کنیم و همچنین تعداد ویژگی‌های انتخاب‌شده هر مجموعه داده را مشخص‌می-کنیم. سپس معیارهای ارزیابی مطالعه‌شده و فراوانی آن‌ها در مطالعات

گذشته را مرورمی‌کنیم. توابع هدف مطالعات و پارامترهای مختلف محبوب را شرح‌می‌دهیم.

۳،۱. نوآوری مقاله

این مطالعه مروری حاضر، الگوریتم‌های فراابتکاری که برای انتخاب ویژگی در سیستم‌های تشخیص نفوذ استفاده‌شده‌اند را شناسایی می‌کند. این پژوهش، برای تحقیقات آینده مفید خواهدبود زیرا تصویر کاملی از مطالعات قبلی را همراه نقاط قوت و ضعف ارائه‌می‌دهد. بنابراین، نوآوری مقاله به صورت زیر آورده‌شده‌است:

- بررسی جامع از الگوریتم‌های فراابتکاری را ارائه‌می‌کنیم که برای انتخاب ویژگی در سیستم‌های تشخیص نفوذ استفاده-شده‌اند.
- انواع مجموعه داده‌های به‌کاررفته را بررسی‌می‌کنیم تا متداول-ترین مجموعه داده‌ها مشخص‌شود.
- توابع هدف استفاده‌شده در کارهای گذشته را تجزیه و تحلیل و بررسی می‌کنیم تا بینش روشنی از کاربردهای آن ارائه‌کنیم.
- معیارهای ارزیابی مطالعات قبلی را ارائه‌می‌کنیم.
- چالش‌ها و پیشنهادهای آینده در حوزه انتخاب ویژگی در تشخیص نفوذ مورد بحث قرارمی‌گیرد.

مقاله به شرح زیر سازماندهی شده‌است: بخش ۲ به بررسی سیستم‌های رایانش ابری، سیستم تشخیص نفوذ و مسئله انتخاب ویژگی پرداخته-است. بخش ۳، روش بررسی ادبیات شرح داده‌می‌شود و بخش ۴، مسئله انتخاب ویژگی در سیستم‌های تشخیص نفوذ را طبقه‌بندی می‌کند. بخش ۵، تأثیر الگوریتم‌های فراابتکاری در مسئله انتخاب ویژگی برای تشخیص نفوذ را بررسی‌می‌کند و مجموعه‌داده‌ها، معیارهای ارزیابی و توابع هدف مطالعات را بررسی‌می‌کند. بخش ۶ چالش‌ها و شکاف‌های تحقیقاتی این حوزه بیان‌می‌شود. در نهایت، بخش ۷، نتیجه گیری و پیشنهادهای آینده برای مسئله انتخاب ویژگی در سیستم‌های تشخیص نفوذ آمده‌است.

۲. بررسی ادبیات

1.2. رایانش ابری

با پیشرفت فناوری اطلاعات نیاز به انجام کارهای محاسباتی در همه جا و همه زمان به وجود آمده‌است. همچنین نیاز به این هست که افراد بتوانند کارهای محاسباتی سنگین خود را بدون داشتن سخت‌افزارها و نرم‌افزارهای گران‌قیمت انجام‌دهند. رایانش ابری آخرین پاسخ فناوری به این نیازها بوده‌است. عموماً مصرف‌کننده‌های رایانش ابری مالک زیر ساخت فیزیکی ابر نیستند، بلکه برای اجتناب از هزینه سرمایه‌ای آن را از عرضه‌کنندگان شخص ثالث اجاره‌می‌کنند. آن‌ها منابع را در قالب سرویس مصرف‌کرده و تنها هزینه منابعی که به‌کارمی‌برند را می‌پردازند. این در حالی است که سایر گونه‌های عرضه‌کنندگان بر مبنای اشتراک سرویس‌های خود را عرضه‌می‌کنند. به اشتراک گذاردن قدرت رایانشی مصرف‌شدنی و ناملموس میان چند مستاجر می‌تواند باعث بهبود نرخ بهره‌وری شود؛ زیرا با این شیوه دیگر سرویس‌دهنده‌ها بدون دلیل بیکار نمی‌مانند و همین امر سبب‌می‌شود در عین حال که سرعت تولید و توسعه برنامه‌های کاربردی افزایش‌یابد، هزینه‌ها به میزان قابل‌توجهی کاهش‌یابند. اهدافی که تکنولوژی رایانش ابری در صنعت فناوری اطلاعات دنبال‌می‌کند به شرح زیر می‌باشد [۱۷]: مقیاس‌پذیری، در دسترس‌بودن، قابلیت اطمینان، امنیت، قابلیت انعطاف‌پذیری، قابلیت سرویس‌دهی و کارایی است. امنیت یکی از اهداف مهم رایانش ابری است که نگرانی بزرگی در تمام جنبه‌های ساخت رایانش ابری است. بیشتر این جنبه‌ها منحصرا برای تنظیمات ابر نیست بلکه داده‌ها بدون درنظرگرفتن محل ذخیره‌سازیشان در معرض خطر قراردارند. بنابراین امنیت رایانش ابری شامل تمام موضوعات امنیت محاسباتی از جمله طراحی معماری، به حداقل رساندن سطوح حمله و حفاظت در برابر ابزارهای مخرب و اعمال کنترل دسترسی می‌باشد. جنبه‌های خاص امنیت رایانش ابری شامل موارد ذیل است:

1. ابر به‌طور معمول منابع را به اشتراک می‌گذارد و شرکا ممکن است جزو هکرها باشند.
2. داده‌های مبتنی بر ابر عمدتاً توسط پروتکل‌های ناامن و رابط‌های برنامه‌های کاربردی (APL) در سراسر شبکه عمومی در دسترس هستند.
3. داده‌ها در ابر در معرض خطر از بین رفتن یا تغییر توسط ارائه-دهنده ابر قراردارد.
4. داده در ابر می‌تواند توسط ارائه‌دهنده ابر، پیمانکاران فرعی و کارکنان آن در دسترس قرارگیرند.

تضمین امنیت دسترسی به داده در شبکه عمومی اگرچه برای رایانش ابری ضروری است ولی درحقیقت یک چالش است و این چالش درواقع این است که داده مشتریان در ابر فاش‌شود و در دسترس عموم قرارگیرد.

۲،۲. سیستم تشخیص نفوذ

عملکردهای اصلی یک سیستم تشخیص نفوذ نظارت بر رویدادهای رخ-داده در یک سیستم یا شبکه رایانه‌ای، تجزیه و تحلیل رویدادهای سیستم، شناسایی فعالیت‌های مشکوک و اعلام هشدار در صورت شناسایی نفوذ است. یک سیستم تشخیص نفوذ معمولی را می‌توان به سه جزء عملکردی [۱۸] تقسیم‌کرد: منبع اطلاعات، موتور تجزیه و تحلیل و تصمیم‌گیرنده. این سه جزء را می‌توان بر روی یک کامپیوتر یا در چند کامپیوتر مختلف اعمال‌کرد. بنابراین کل سیستم تشخیص نفوذ می‌تواند یک سیستم میزبان در یک کامپیوتر یا یک سیستم توزیع شده در یک شبکه محلی یا حتی در سراسر اینترنت باشد.

اولین جزء از سیستم تشخیص نفوذ، منبع اطلاعات است که برای نظارت بر رویدادهای رخداده در یک سیستم یا شبکه کامپیوتری استفاده‌می‌شود. در طول مدت نظارت، منبع اطلاعات جریانی از سوابق رویداد را برای تجزیه و تحلیل فراهم‌می‌کند. این مؤلفه به‌عنوان یک مولد رویداد کارمی‌کند. منابع داده‌های مختلف را نظارت‌می‌کند و داده‌های

رویداد را تولیدمی‌کند که آیا به‌خوبی قالب‌بندی شده‌اند و برای یک موتور تجزیه و تحلیل برای انجام تجزیه و تحلیل بیشتر مناسب هستند. منابع داده را براساس اینکه داده‌های آن‌ها از کجا آمده‌است، می‌توان به سه دسته تقسیم‌کرد: دسته اول منبع داده مربوط به سیستم عامل‌ها مانند تماس‌های سیستم و گزارش‌های سیستم است. مانند مجموعه داده UNM [۱۹]. دسته دوم مانیتورهای ترافیک شبکه هستند که بسته‌های خام شبکه را تولیدمی‌کنند. یکی از نمونه‌های این نوع داده‌ها، مجموعه داده‌های KDD است [۱۹]. دسته سوم جمع‌آوری داده‌های برنامه‌های مختلف در حال اجرا در سیستم‌های شبکه یا محیط ابری است. جزء دوم، موتور تجزیه و تحلیل بخش کلیدی سیستم تشخیص نفوذ است. یک سیستم تشخیص نفوذ برای یافتن علائم نفوذ به موتور آنالیز متکی است. تمام تکنیک‌های هوش مصنوعی را می‌توان برای این جزء اعمال‌کرد. موتور تجزیه و تحلیل، اطلاعاتی را که از منبع اطلاعات به‌دست‌می آید، فیلتر می‌کند، و هرگونه داده نامربوط در اطلاعات را حذف می‌کند و در نتیجه فعالیت‌های مشکوک را شناسایی‌می‌کند. موتور تجزیه و تحلیل، معمولا از یک پایگاه داده سیاست تشخیص، برای تجزیه و تحلیل استفاده‌می‌کند. بسته به روش‌ها و تکنیک‌های مختلف تشخیص نفوذ، امضاهای حمله، پروفایل‌های رفتار عادی و پارامترهای لازم (مانند آستانه ها) در پایگاه داده سیاست تشخیص وجوددارد.

آخرین مؤلفه، تصمیم‌گیرنده است که قوانینی را برای نتایج موتور تحلیل اعمال‌می‌کند و تصمیم‌می‌گیرد که براساس نتایج موتور تجزیه و تحلیل چه واکنش‌هایی باید انجام‌شود. دلیل اصلی استفاده از تصمیم گیرنده افزایش قابلیت استفاده از سیستم تشخیص نفوذ است.

۳,۲ انتخاب ویژگی

یک ویژگی یک خاصیت قابل‌اندازه‌گیری منحصربه‌فرد از فرآیند مشاهده‌شده‌است [۲۰]. ویژگی‌ها پایه بسیاری از عملیات داده کاوی و یادگیری ماشین هستند. درواقع با استفاده از مجموعه‌ای از ویژگی‌ها، هر الگوریتم یادگیری ماشین می‌تواند طبقه‌بندی را انجام‌دهد. تعداد ویژگی‌ها در مسایل مختلف متفاوت است و در بعضی از مسایل تعداد آن‌ها از ده‌ها به صدها متغیر یا ویژگی گسترش یافته‌است بنابراین نیاز به کاهش متغیرهای نامربوط و زائد است. چندین تکنیک برای رسیدگی به این مشکل توسعه داده‌شده‌است.

انتخاب ویژگی (حذف متغیر) به درک داده‌ها، کاهش نیاز محاسباتی، کاهش اثر بد ابعاد و بهبود عملکرد پیش‌بینی کمک‌می‌کند [۲۰]. تمرکز انتخاب ویژگی انتخاب زیرمجموعه‌ای از متغیرها از ورودی است که می‌تواند به‌طور مؤثر داده‌های ورودی را توصیف‌کند و در عین حال اثرات ناشی از نویز یا متغیرهای نامربوط را کاهش‌دهد و همچنان نتایج پیش‌بینی خوبی ارائه‌دهد [۲۱]. انتخاب ویژگی تعداد کل ویژگی‌ها را به حداقل می‌رساند و فقط ویژگی‌های کارآمد را بر اساس ورودی ارائه‌شده با کاهش داده‌های نویز انتخاب‌می‌کند، که به شناسایی سریع نتیجه کمک‌می‌کند. در مجموعه داده‌هایی که تعداد ویژگی بسیار زیاد است تعدادی از ویژگی‌ها با دیگر ویژگی همبستگی بالایی دارند و وقتی دو ویژگی کاملاً همبسته هستند، وجود تنها یک ویژگی برای توصیف داده ها کافی است. بنابراین ویژگی‌های وابسته مانند یک نویز برای ماشین های یادگیری عمل‌می‌کنند. پس می‌توان با حذف آن‌ها، میزان داده‌ها را کاهش‌داد که منجر به بهبود عملکرد ماشین‌های یادگیری شود. در برخی برنامه‌ها، متغیرهایی که هیچ ارتباطی با کلاس‌ها ندارند، به عنوان نویز خالص عمل‌می‌کنند، ممکن است اختلالی در پیش‌بینی‌کننده ایجادکنند و عملکرد طبقه‌بندی را کاهش‌دهند. این امر می‌تواند زمانی اتفاق بیفتد که اطلاعات کافی در مورد فرآیند مورد مطالعه وجودنداشته‌باشد [۲۰]. با استفاده از تکنیک‌های انتخاب ویژگی، می‌توانیم بینشی در مورد فرآیند به‌دست‌آوریم و می‌توانیم نیاز محاسباتی و دقت پیش‌بینی را بهبودبخشیم [۲۰].

روش‌های مختلفی در انتخاب ویژگی وجوددارد. روش‌های اصلی در انتخاب ویژگی عبارتند از: انتخاب ویژگی مبتنی بر فیلتر، انتخاب ویژگی مبتنی بر روش‌های بسته‌بند(Wrapper) ، روش‌های تعبیه‌شده (Embedded methods).

فرآیند روش فیلتر معمولاً از تکنیک رتبه‌بندی استفاده‌می‌کند. برای امتیازدهی به متغیرها از معیار رتبه‌بندی مناسب و برای حذف متغیرهای زیر آستانه از آستانه استفاده‌می‌شود. روش‌های رتبه‌بندی، روش‌های فیلتر هستند، زیرا قبل از طبقه‌بندی برای فیلترکردن متغیرهای کمتر مرتبط، به‌کارمی‌روند [۲۰]. در فیلترها (Filters) میزان اهمیت و تفکیک‌پذیری ویژگی‌ها جداگانه بررسی‌می‌شوند و طبق رویکرد الگوریتم مورد نظر به تک‌تک ویژگی‌ها امتیازی داده‌می‌شود که این امتیاز میزان تفکیک‌پذیری و اهمیت ویژگی را مشخص‌می‌کند. در روش‌های فیلتر، مدل (طبقه‌بند) در جریان انتخاب ویژگی نیست که این امر هم مزیت و هم عیب این روش‌ها محسوب‌می‌شود. مزیت این رویکرد این است که زمان اجرای الگوریتم بسیار پایین است و در مدت زمان بسیار کوتاه بدون اینکه مدل در جریان کار باشد، به ویژگی‌ها امتیاز داده‌می‌شود و از بین ویژگی‌ها یک تعداد ویژگی بسته به نیاز مسأله، انتخاب‌می‌شوند. عیب این رویکرد این است که چون مدل در جریان کار نیست، ممکن است ویژگی‌های مناسب برای مدل انتخاب‌نشود. ایراد دوم روش‌های فیلتر این است که ارتباط بین ویژگی‌ها درنظرگرفته‌نمی‌شود، و همین ممکن است باعث انتخاب ویژگی‌هایی شود که به‌تنهایی خوب عمل‌کنند، ولی کنار هم عالی عمل‌نکنند. هدف اصلی ما این است که ویژگی‌های انتخاب‌شوند که کنار هم عملکرد طبقه‌بندی یا رگرسیون را بالا ببرند، ولی در این رویکرد ممکن است ویژگی‌های انتخاب‌شده ترکیب خوبی تشکیل‌ندهند. با این حال روش‌های فیلتر، روش‌های بسیار مهمی، مخصوصاً در تعیین اهمیت ویژگی‌ها، هستند، زیرا که با هزینه زمانی بسیار پایین می‌توانند مشخص‌کنند که یک ویژگی مناسب است یا نه.

روش‌های بسته‌بند(Wrapper) ، مدل را در جریان انتخاب ویژگی قرارمی‌دهند و ارتباط بین ویژگی‌ها را درنظرمی‌گیرند. این روش‌ها سعی بر آن دارند که بهترین ترکیب از بین ویژگی‌های موجود انتخاب کنند. مدل روش‌های بسته‌بند(Wrapper) از الگوریتم‌های یادگیری ماشین

برای تولید زیرمجموعه ویژگی به‌عنوان بخشی از تابع ارزیابی ویژگی استفاده‌می‌کند. یعنی از زیرمجموعه ویژگی‌ها استفاده‌می‌کند و مدل را آموزش‌می‌دهد [۲۱]. زیرمجموعه‌های ویژگی براساس الگوریتم‌های استقرایی انتخاب‌می‌شوند. این زیرمجموعه ویژگی انتخاب‌شده، دقت مدل آموزشی را تخمین‌می‌زند. بسته به دقت اندازه‌گیری در مرحله قبل، روش، تصمیم‌می‌گیردکه آیا یک ویژگی را از زیر مجموعه انتخاب‌شده، اضافه یا حذف کند. به همین دلیل، روش بسته‌بند(Wrapper) از نظر محاسباتی پیچیده‌تر است. روش بسته‌بند(Wrapper) مدل را در جریان انتخاب ویژگی قرار می‌دهند لذا زمان‌بر است و امکان دارد زمانی که تعداد ویژگی‌ها زیاد شود، هزینه زمانی بالایی داشته‌باشد. اما چون از لحاظ رویکرد انتخاب ویژگی بسیار توانمند است، در عمل بسیار استفاده-می‌شود.

روش‌های توکار (Embedded) انتخاب ویژگی را در فرآیند آموزش انجام‌می‌دهند و معمولاً برای ماشین‌های یادگیری خاصی مورد استفاده قرارمی‌گیرند. این روش به گونه‌ای عمل‌می‌کند که بهترین ویژگی‌ها در طول فرآیند یادگیری انتخاب‌می‌شوند. ترکیب انتخاب ویژگی در طول فرآیند یادگیری مزایایی در بهبود هزینه محاسباتی، دقت طبقه‌بندی دارد و همچنین از آموزش مدل در هر بار اضافه‌شدن ویژگی جدید جلوگیری‌می‌کند [۹]. روش Embedded بر پیچیدگی محاسباتی غلبه‌می کند. در این روش انتخاب ویژگی مناسب و یادگیری مدل به طور همزمان انجام‌می‌شود و ویژگی‌ها در مرحله آموزش مدل انتخاب می‌شوند. به همین دلیل هزینه محاسباتی این روش در مقایسه با روش بسته‌بند(Wrapper) به‌طور قطعی کمتر است. در این روش، هر بار که یک ویژگی جدید پیدامی‌شود، نیاز به آموزش مدل نیست.

جدول۳: نقش فیلتر، بسته‌بندی و مدل تعبیه‌شده در فرآیند انتخاب ویژگی

روش انتخاب ویژگی	نقش در فرآیند انتخاب ویژگی
مدل فیلتر	- این روش معیارهای آماری مختلفی را برای تخصیص رتبه یا امتیاز به هر ویژگی انجام می-دهد. - این روش زیر مجموعه‌های ویژگی را قبل از مدل یادگیری انتخاب‌می‌کند فرآیند انتخاب یک زیر مجموعه ویژگی تنها یک بار در مکانیسم کل مدل یادگیری ساختمان انجام‌می شود. - این روش برای ایجاد زیر مجموعه ویژگی از ویژگی مربوط به ویژگی پیروی می‌کند. - فرآیند تولید زیر مجموعه ویژگی مستقل از هر الگوریتم ML است. - این روش از ویژگی ذاتی ویژگی‌ها پیروی می کند. همچنین از وابستگی بین ویژگی ها برای تولید زیر مجموعه ویژگی ها غفلت می‌کند.
	- این روش خاصیت چند خطی را حذف نمی کند به طور متناوب باید با مکانیسم دیگری حذف شود.
مدل بسته‌بند	- این روش دقت مدل را برای تخصیص رتبه یا امتیاز به هر ویژگی نشان می‌دهد. - این روش از وابستگی بین ویژگی‌ها برای تولید زیر مجموعه ویژگی پیروی می‌کند. - برای تولید زیر مجموعه، ویژگی طبقه‌بندی کننده چندین بار اجرامی‌شود. - این روش برای تولید زیر مجموعه ویژگی‌ها با درنظرگرفتن مدل یادگیری استفاده‌می‌شود. - این روش مفیدترین ویژگی‌ها را برای تولید زیر مجموعه ویژگی انتخاب‌می‌کند. - از استنباط مدل یادگیری قبلی این روش تصمیم‌می‌گیرد که آیا ویژگی را از زیر مجموعه ویژگی اضافه یا حذف کند. - این روش از یک طبقه‌بندی خاص برای ارزیابی کیفیت ویژگی‌های انتخاب‌شده استفاده‌می‌کند.
مدل تعبیه‌شده	- فرآیند FS در مرحله آموزش ایجادمی‌شود و این روش زیر مجموعه ویژگی را برای الگوریتم در حال آموزش ارزیابی‌می‌کند. - این روش از ویژگی مفید بودن ویژگی‌ها برای تولید زیر مجموعه ویژگی‌ها استفاده‌می‌کند. - این روش از مکانیسم یادگیری برای فضای جستجو استفاده می‌کند. - در این روش از تجمیع مزایای روش فیلتر و لفاف استفاده‌می‌شود. - این رویکرد وابستگی بین ویژگی‌ها را بسیار موثر محاسبه می‌کند. با توجه به در نظر گرفتن طبقه‌بندی‌کننده، این روش ویژگی‌های مربوطه را انتخاب‌می‌کند.

بهترین روش (به صورت مطلق) برای انتخاب ویژگی وجود ندارد و از این رو تلاش‌های محققان بر انتخاب روشی متمرکز شده که برای یک مسأله مشخص بهتر عمل‌می‌کند. روش‌های گوناگونی برای مواجهه با مجموعه داده‌های کلان مقیاس وجوددارد که اهمیت انتخاب ویژگی در آن‌ها واقعیتی غیرقابل انکار است، زیرا منجر به کمینه‌کردن زمان آموزش و حافظه تخصیصی با حفظ صحت نتایج می‌شود.

۴،۲. الگوریتم‌های فراابتکاری

الگوریتم‌های فراابتکاری، مجموعه‌ای از روش‌های بهینه‌سازی تصادفی هستند که برای حل مسائل پیچیده که با روش‌های سنتی به‌سختی حل‌می‌شوند، به‌کارمی‌روند. این الگوریتم‌ها از الهام‌گیری از فرایندهای طبیعی مانند تکامل، رفتار اجتماعی حشرات و فیزیک ذرات برای یافتن راه‌حل‌های بهینه یا نزدیک بهینه استفاده‌می‌کنند [۲۲]. الگوریتم‌های

فراابتکاری از جستجوی تصادفی برای کاوش در فضای جستجو و یافتن راه‌حل‌های بالقوه استفاده‌می‌کنند. این الگوریتم‌ها مکرر راه‌حل‌های موجود را بهبودمی‌بخشند تا به راه حل بهینه ممکن برسند و البته الگوریتم‌های فراابتکاری تضمین‌نمی‌کنند که به بهترین راه‌حل ممکن دست پیدا کنند، اما می‌توانند راه‌حل‌های با کیفیت بالا را در زمان معقول پیداکنند.

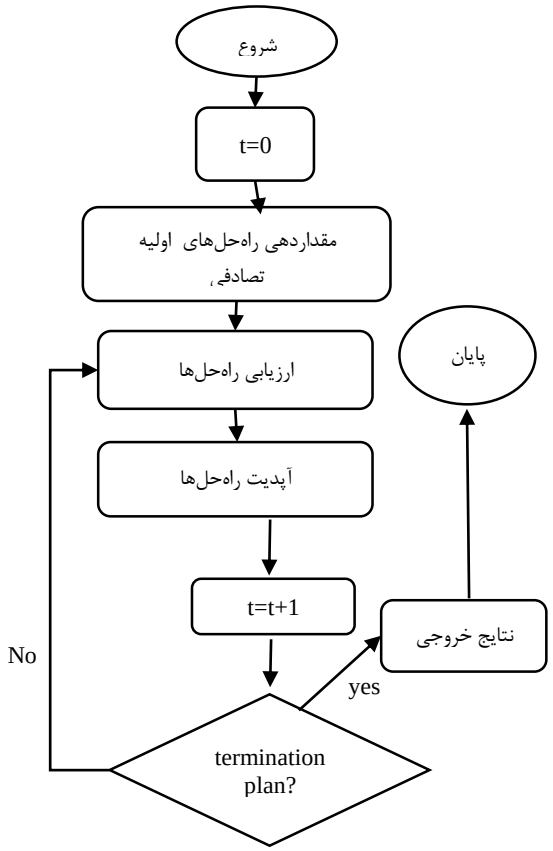
الگوریتم‌های فراابتکاری به سه دسته کلی تقسیم‌می‌شوند: الگوریتم‌های تکاملی: این الگوریتم‌ها از فرایند تکامل الهام‌می‌گیرند و از جهش، انتخاب و تقاطع برای ایجاد راه‌حل‌های جدید استفاده‌می‌کنند مانند الگوریتم ژنتیک (Genetic Algorithm) و استراتژی تکامل (Evolution Strategy) [۲۳]. الگوریتم‌های مبتنی بر جمعیت [۲۴]: این الگوریتم‌ها از یک جمعیت از راه‌حل‌ها استفاده‌می‌کنند و مکرر آن‌ها را بر اساس تابع هدفشان بهبودمی‌بخشند مانند بهینه‌سازی ازدحام ذرات (Particle Swarm Optimization)، بهینه‌سازی کلنی مورچه‌ها (Colony Optimization)، بهینه‌سازی کلنی زنبورها (Bee Colony Optimization). الگوریتم‌های مبتنی بر فیزیک: این الگوریتم‌ها از مفاهیم فیزیکی مانند شبیه‌سازی تبرید و جستجوی تصادفی شبیه‌سازی شده برای یافتن راه‌حل‌های بهینه استفاده می‌کنند [۲۵]. یک موضوع مشترک که در تمام فراابتکاری‌ها دیده می شود، تعادل بین اکتشاف (Exploration)و بهره برداری (Exploitation) است.

اکتشاف، به این معنی است که عامل‌ها باید در فضای مسئله جستجوکنند تا راه‌حل‌های ممکن مسئله را بیابند [۲۶]. این جنبه به الگوریتم فراابتکاری یک رفتار جستجوی جهانی می‌دهد. بهره‌برداری به این اشاره دارد که عامل‌ها چگونه می‌توانند از اطلاعات موجود از راه حل‌های تکرارهای قبلی، برای یافتن راه‌حل بهینه استفاده‌کنند [۲۶]. چنین تشدیدی به فراابتکاری ویژگی جستجوی محلی می‌دهد.

الگوریتم‌های فراابتکاری، در بسیاری از مسائل کاربرد دارند، از جمله مسائل بهینه‌سازی مانند زمان‌بندی، مسیریابی و غیره، مسائل پیش‌بینی یا طبقه‌بندی داده‌ها و مسائل مهندسی مانند طراحی سیستم‌های قدرت و شبکه‌های حمل و نقل. از دیگر کاربردهای الگوریتم‌های فراابتکاری، انتخاب ویژگی است [۲۷]. به‌عنوان مثال، می‌توان از آن‌ها برای جستجوی زیرمجموعه‌ای از ویژگی‌ها که منجر به بهترین دقت مدل می‌شوند، استفاده‌کرد. یکی از مزایای استفاده از الگوریتم‌های فراابتکاری برای انتخاب ویژگی این است که می‌توان از آن‌ها برای حل مسائل با تعداد زیادی ویژگی استفاده‌کرد. علاوه بر این، الگوریتم‌های فراابتکاری می‌توانند همزمان چندین ویژگی را انتخاب‌کنند، که می‌تواند به یافتن تعاملات بین ویژگی‌ها کمک‌کند.

شکل ۱ فلوچارت الگوریتم‌های فراابتکاری را نشان‌می‌دهد. اکثر الگوریتم‌های فراابتکاری، عملیات مشابهی را دنبال‌می‌کنند و بنابراین می‌توانند در یک چارچوب عمومی مشترک تعریف‌شوند که در زیر آمده‌است:

۱. شمارنده تکرار t را برابر با ۰ مقداردهی کنید.
۲. مجموعه راه‌حل‌ها را به‌صورت تصادفی مقداردهی اولیه کنید.
۳. هریک از راه‌حل‌ها را ارزیابی کنید.
۴. راه‌حل‌ها را آپدیت کنید.
۵. یک واحد به t اضافه کنید.
۶. مراحل ۳ تا ۵ را تا زمانی که یک طرح خاتمه (termination plan) برآورده‌شده‌است، ادامه‌دهید.



شکل۱: فلوچارت الگوریتم‌های فراابتکاری

هر الگوریتم فراابتکاری طی چندین مراحل به جواب بهینه می‌رسد. در مرحله اول، شمارنده t مقدار صفر می‌گیرد تا تعداد تکرار الگوریتم را بشمارد. در مرحله ۲، مجموعه راه‌حل‌های مسئله به‌طور تصادفی و دیگر پارامترهای الگوریتم فراابتکاری مقداردهی اولیه می‌شوند. روش تولید راه‌حل‌های اولیه، نقش مهمی در عملکرد الگوریتم‌های فراابتکاری دارد بنابراین روش‌های متفاوتی برای مقداردهی راه‌حل‌های اولیه وجود دارد. در مرحله ۳، با استفاده از تابع هدف، ارزش هر راه‌حل به‌دست‌می‌آید که با توجه به مقدار ارزش، بهترین راه‌حل مشخص‌می‌شود. ارزش تابع هدف، معیاری برای ارزیابی عملکرد خوب الگوریتم است [۲۷، ۲۵]. در مرحله ۴، با توجه به راه‌حل‌های قبلی بسته به نوع الگوریتم فراابتکاری، راه‌حل‌های جدید ارائه‌می‌شود. سپس در انتهای حلقه، شمارنده آن یک واحد اضافه‌می‌شود. در مرحله ۶ چنانچه شرط پایانی حلقه (که معمولاً آستانه دقت یا تعداد مراحل تکرار الگوریتم فراابتکاری است) برآورده شود نتایج به خروجی ارسال‌شده وگرنه دوباره باید الگوریتم تکرار شود. برای انتخاب ویژگی با الگوریتم‌های فراابتکاری باید برای هر عامل جستجو، به تعداد ویژگی‌های مجموعه داده، ویژگی درنظرگرفت.

۵،۲. انتخاب ویژگی مبتنی بر الگوریتم‌های فراابتکاری

مقادیر هر ویژگی نیز باینری می‌باشد یعنی هر ویژگی مقدار ۰ یا ۱ می‌گیرند. مثلاً اگر بخواهیم برای انتخاب ویژگی مجموعه داده‌ای با ۲۴ ویژگی از الگوریتم ژنتیک استفاده‌کنیم باید برای هر کروموزوم ۲۴ ژن درنظربگیریم که مقادیر هر یک از آن‌ها ۰ یا ۱ است. صفر نشان‌دهنده عدم انتخاب ویژگی متناظر با آن و یک نشان‌دهنده این است که ویژگی متناظر با آن انتخاب‌شده‌است.

3. روش بررسی ادبیات

این بخش نحوه انتخاب و بررسی تحقیقات مربوط به تکنیک انتخاب ویژگی مبتنی بر الگوریتم فراابتکاری در سیستم‌های تشخیص نفوذ را معرفی‌می‌کند. بر اساس سه سوال زیر، تحقیقات مرتبط انتخاب‌شده‌اند.

۱. چه الگوریتم‌های فراابتکاریی در انتخاب ویژگی برای سیستم‌های تشخیص نفوذ استفاده‌می‌شود؟

۲. انتخاب ویژگی مبتنی بر الگوریتم‌های فراابتکاری در سیستم‌های تشخیص نفوذ چقدر مؤثرند؟

۳. چالش‌ها و شکاف‌های تحقیقاتی در الگوریتم‌های فراابتکاری موجود برای انتخاب ویژگی در سیستم‌های تشخیص نفوذ چیست؟

سوال اول به ما کمک‌می‌کند تا الگوریتم‌های فراابتکاری جدیدی که در بحث انتخاب ویژگی برای سیستم‌های تشخیص نفوذ استفاده-می‌شوند، بشناسیم و تحقیقات مربوط به آن را جمع‌آوری و تجزیه و تحلیل کنیم. سوال دوم، به بررسی معیارهای ارزیابی و عملکرد آن الگوریتم‌ها می‌پردازد. و سوال اخر چالش‌های موجود در بحث را بررسی-می‌کند تا پیشنهادهای تحقیقات آینده ارائه‌کند.

کلمات کلیدی جستجوشده برای یافتن مقالات معتبر برای این مطالعه عبارتند از: "سیستم تشخیص نفوذ"، "سیستم تشخیص نفوذ در ابر"، و "انتخاب ویژگی در سیستم تشخیص نفوذ". این مقالات در مجلات معتبر مانند Springer، ScienceDirect، IEEE و سایر منابع علمی معتبر، در بازه زمانی سال‌های ۲۰۱۸ تا ۲۰۲۴ منتشرشده‌اند.

4 . مسئله انتخاب ویژگی در سیستم تشخیص نفوذ:

ترافیک داده در محیط ابری خیلی متنوع و دارای حجم زیادی از ویژگی هاست. سیستم‌های تشخیص نفوذ، با این ویژگی‌ها سروکار دارند. طبقه-بندی ترافیک ورودی مسأله سختی است زیرا: (۱) وجود بسیاری از ویژگی‌ها منجر به فرآیندهای آموزشی طولانی می‌شود (به ویژه زمانی که ویژگی ها بسیار همبسته هستند)، درحالی‌که دقت پیش‌بینی متناسب آن بهبود نمی‌یابد. (۲) برخی از ویژگی‌ها ممکن است در طول فرآیند طبقه‌بندی، سوگیری ایجادکنند، به‌ویژه آن‌هایی که ارتباط کمی با ترافیک داده‌ای مورد طبقه‌بندی، داشته‌باشند [۱۰]. برای این منظور، با کاهش فضای ویژگی و حفظ مهم‌ترین ویژگی‌ها، انتخاب ویژگی به یک مرحله پیش پردازش مهم در مدیریت شبکه و به‌ویژه برای اهداف تشخیص نفوذ شبکه تبدیل‌می‌شود[۱۰] و [۲۸].

انتخاب ویژگی فرآیندی است برای انتخاب زیرمجموعه بهینه ویژگی ها از یک مجموعه ویژگی بزرگ برای بهبود دقت طبقه‌بندی، عملکرد و هزینه استخراج ویژگی‌ها [۲۹]. در این مقاله در مورد روش‌های انتخاب ویژگی (فیلتر، روش بسته‌بند (Wrapper)، روش تعبیه‌شده (Embedded)) صحبت خواهدشد. در این بخش، الگوریتم‌های استفاده شده در مقالات انتخابی و میزان تأثیر هریک را بررسی و سعی می‌کنیم به سه سوال پیشتر مطرح شده پاسخ‌دهیم.

پاسخ سوال اول: الگوریتم‌های تکاملی (Evolutionary Algorithms) دسته‌ای از الگوریتم‌های بهینه‌سازی مبتنی بر اصول تکامل زیستی و ژنتیک هستند [۳۰]. این الگوریتم‌ها از مفاهیمی مانند انتخاب طبیعی، جهش، ترکیب و بازتولید برای پیدا کردن راه‌حل‌های بهینه استفاده-می‌کنند. مهمترین الگوریتم‌های تکاملی عبارتند از: الگوریتم ژنتیک و الگوریتم استراتژی‌های تکاملی. الگوریتم ژنتیک (Genetic Algorithm - GA -): یک روش بهینه‌سازی مبتنی بر فرآیندهای تکاملی زیستی است که از مفاهیم انتخاب طبیعی، جهش، ترکیب و بازتولید برای جستجوی راه‌حل‌های بهینه استفاده‌می‌کند. این الگوریتم با شروع از یک جمعیت اولیه از کروموزوم‌ها (راه‌حل‌های ممکن)، در هر نسل با استفاده از عملگرهای ژنتیکی مانند انتخاب (که به راه‌حل‌های بهتر شانس بیشتری برای بازتولید می‌دهد)، ترکیب (ادغام دو راه‌حل برای تولید راه‌حل‌های جدید) و جهش (ایجاد تغییرات تصادفی برای معرفی تنوع) به‌تدریج به سمت بهبود راه‌حل‌ها حرکت‌می‌کند. یکی از مزایای اصلی الگوریتم ژنتیک، توانایی آن در فرار از بهینه‌های محلی و جستجوی فضای گسترده‌ای از راه‌حل‌ها است. این الگوریتم به‌دلیل انعطاف‌پذیری بالا و قابلیت موازی‌سازی، در مسائل مختلفی مانند بهینه‌سازی توابع پیچیده، برنامه‌ریزی و طراحی مهندسی استفاده‌می‌شود .الگوریتم استراتژی‌های تکاملی (Evolution Strategies - ES) یکی از روش‌های قدرتمند بهینه‌سازی مبتنی بر اصول تکاملی زیستی است که به‌ویژه برای مسائل بهینه‌سازی پیوسته مناسب است. این الگوریتم با شروع از یک جمعیت اولیه از راه‌حل‌ها، از فرآیندهای انتخاب، جهش و بازتولید برای بهبود تدریجی راه‌حل‌ها استفاده‌می‌کند. در هر نسل، راه‌حل‌های بهتر برای تولید نسل بعد انتخاب می‌شوند و از توزیع‌های آماری برای ایجاد جهش‌های کوچک و بزرگ در پارامترهای راه‌حل‌ها استفاده‌می‌شود. برخلاف برخی دیگر از الگوریتم‌های تکاملی، ES معمولاً از عملگرهای ترکیب (جفت‌گیری) استفاده‌نمی‌کند و بیشتر بر روی تغییرات و جهش‌های تصادفی تمرکز دارد. یکی از مزایای اصلی استراتژی‌های تکاملی، کارایی بالا در بهینه‌سازی مسائل با پارامترهای واقعی و پیوسته است. این الگوریتم به دلیل انعطاف‌پذیری در انتخاب توزیع‌های جهش و توانایی تطبیق با شرایط مختلف، می‌تواند به راه‌حل‌های با کیفیت بالا دست‌یابد. همچنین، ES به‌دلیل ساختار ساده و قابلیت موازی‌سازی، در کاربردهای صنعتی و مهندسی مانند طراحی مکانیکی، بهینه‌سازی فرآیندها و تنظیم پارامترهای سیستم‌های پیچیده استفاده‌می‌شود.

الگوریتم‌های هوش جمعی (Swarm Intelligence Algorithms): الگوریتم‌های هوش جمعی از رفتار اجتماعی گروه‌های موجودات زنده مانند مورچه‌ها، پرندگان، و ماهی‌ها الهام گرفته‌شده‌اند. این الگوریتم‌ها به‌دنبال یافتن راه‌حل‌های بهینه از طریق تعاملات ساده بین اعضای جمعیت هستند [۳۱]. مهمترین الگوریتم‌های هوش جمعی عبارتند از: الگوریتم بهینه‌سازی ازدحام ذرات، الگوریتم بهینه‌سازی زنبور، الگوریتم کبوتر.

الگوریتم بهینه‌سازی ازدحام ذرات (Particle Swarm Optimization - PSO) الهام‌گرفته از رفتار جمعی ذرات در طبیعت است. در این الگوریتم، یک جمعیت از ذرات در یک فضای چندبعدی قراردارند که هر ذره یک حالت (یا راه‌حل) را نمایش‌می‌دهد. هر ذره به یک مکان در فضا نسبت داده‌می‌شود و در هر مرحله، بهترین حالت خود و بهترین حالت جمعیت را ذخیره‌می‌کند. ذرات در جستجوی بهینه‌سازی با تغییر مکان‌شان به سمت بهترین حالت‌ها حرکت‌می‌کنند. در هر مرحله، هر ذره با درنظرگرفتن مکان بهترین حالت خود و بهترین حالت جمعیت، سرعت و مکان خود را به‌روزرسانی می‌کند. این به‌روزرسانی از طریق محاسبه میزان تأثیرگذاری دو عامل اصلی، یعنی جهت بهترین حالت شخصی ذره و جهت بهترین حالت جمعیت انجام‌می‌شود. این تأثیرگذاری می‌تواند با استفاده از ضرایب وزنی مناسبی کنترل‌شود. یکی از مزایای اصلی PSO، سادگی پیاده‌سازی و انطباق آسان آن با مسائل مختلف است.

الگوریتم بهینه‌سازی زنبور (Bee Colony Optimization - BCO) الهام‌گرفته از رفتار جمعی زنبورها در جستجوی منابع غذایی است. در این الگوریتم، یک جمعیت از زنبورها وجود دارد که هر کدام از آن‌ها یک مسیر احتمالی برای یافتن بهینه‌سازی را نمایش‌می‌دهد. زنبورها در این الگوریتم به دو دسته تقسیم‌می‌شوند: زنبورهای کارگر و زنبورهای معمولی. زنبورهای کارگر به‌دنبال مکان‌های مختلف منابع غذایی (راه‌حل‌های پتانسیلی) در فضای جستجو هستند، درحالی‌که زنبورهای معمولی اطلاعات از زنبورهای کارگر را دریافت‌کرده و تصمیم‌گیری می‌کنند که کدام مکان‌ها را برای بررسی بیشتر انتخاب‌کنند. در هر مرحله، زنبورهای کارگر به مکان‌های مختلف در فضای جستجو حرکت‌می‌کنند و اطلاعات دریافتی را برای بهبود مکان‌های پتانسیلی به زنبورهای معمولی ارسال‌می‌کنند. زنبورهای معمولی سپس این اطلاعات را بررسی و مکان‌های مناسب برای جستجوی بیشتر تعیین‌می‌کنند. این فرآیند تکرار می‌شود تا بهینه‌سازی بهبود یابد. مزیت اصلی الگوریتم بهینه‌سازی زنبور، توانایی ترکیب جستجوی محلی و جهانی در یافتن بهینه‌سازی است. همچنین، قابلیت تطبیق با محیط‌های مختلف و ساختار ساده این الگوریتم آن را به یک روش مؤثر در حل مسائل بهینه‌سازی مختلف، از جمله مسائل مسیریابی و بهینه‌سازی سیستم‌های پیچیده می‌کند. الگوریتم‌های مبتنی بر مسیر جستجو به‌دنبال بهینه‌سازی با پیگیری و اصلاح مسیر حرکت یک راه‌حل در فضای جستجو هستند. این الگوریتم‌ها معمولاً از یک نقطه شروع و با دنبال-کردن مسیرهایی به سمت بهبود کیفیت راه‌حل حرکت‌می‌کنند [۳۲].

مهمترین الگوریتم مبتنی بر مسیر جستجو الگوریتم جستجوی تابو است. الگوریتم جستجوی تابو (Tabu Search) یک الگوریتم بهینه‌سازی است که برای حل مسائل بهینه‌سازی ترکیباتی استفاده می‌شود. این الگوریتم ابتدا از یک حالت اولیه شروع‌می‌کند و سپس با استفاده از جستجوی محلی، به بهبود آن حالت می‌پردازد. اما برخلاف الگوریتم‌های جستجوی محلی که ممکن است در یک بهینه محلی گیر کنند، الگوریتم جستجوی تابو از طریق استفاده از یک لیست تابو (Tabu List)، از بازگشت به حالت‌هایی که قبلاً بررسی‌شده‌اند و به‌عنوان محلی بهینه مشخص‌شده‌اند، جلوگیری‌می‌کند. علاوه‌بر استفاده از لیست تابو، الگوریتم جستجوی تابو از استراتژی‌های متنوعی مانند تغییرات جواری (Neighborhood Changes)، جستجوی محلی (Local Search) و معیارهای ارزیابی مختلف برای بهبود بهینه‌سازی استفاده‌می‌کند [۳۲].

این الگوریتم قابلیت پیشرفت پیش‌بینی را داراست، به این معنی که با گذر از هر مرحله، به یک بهینه بهتر نسبت به حالت قبلی دست‌می‌یابد. مزیت اصلی الگوریتم جستجوی تابو، توانایی پیش‌بینی و جلوگیری از گیر افتادن در بهینه‌های محلی است، همچنین این الگوریتم به‌دلیل ساختار انعطاف‌پذیری که دارد، می‌تواند در حل مسائل مختلفی از جمله مسائل ناپیوسته، مسائل جریان کار، و بهینه‌سازی ترکیبیاتی استفاده-شود. شکل ۲ طبقه‌بندی الگوریتم‌های فراابتکاری و فراوانی آن‌ها را در مطالعات بررسی‌شده، نشان‌می‌دهد. الگوریتم‌های فراابتکاری را طبق مقالات بررسی شده به سه دسته تقسیم‌کردیم. الگوریتم‌های تکاملی که شامل انواع الگوریتم‌های ژنتیک و تکاملی تفاضلی است [۴۸، ۵۷، ۶۰، ۶۵، ۵۰، ۵۱، ۵۹، ۶۲، ۶۴، ۶۶]. الگوریتم‌های مبتنی بر جستجو مانند الگوریتم جستجوی تابو [۳۶] و الگوریتم‌های مبتنی بر هوش جمعی که شامل الگوریتم‌های کبوتر [۴۶، ۷۰]، خفاش [۳۴، ۵۴، ۵۸، ۷۱]، گرگ خاکستری [۷۵و۷۶]، کلاغ [۴۱ و ۶۳]، ازدحام ذرات [۴۵، ۶۷، ۶۸]، کلونی زنبورها [۴۳، ۵۶،۶۹]، جستجوی گرانشی [۳۹]، کاتل فیش (cuttlefish) [۳۵، ۵۵]، کرم شبتاب [۳۳،۵۳]، الگوریتم نهنگ [۷۲، ۷۴،۷۳] و دیگر الگوریتم‌ها [۱۱۴، ۳۸، ۴۰، ۴۲، ۴۴، ۴۷، ۴۹، ۵۲، ۶۱] می‌باشد.

شکل۳ نشان‌می‌دهد که الگوریتم‌های فراابتکاری ۲۲٪، الگوریتم‌های مبتنی بر جستجو ۲٪ و الگوریتم‌های مبتنی بر هوش جمعی ۷۶٪ در انتخاب ویژگی برای تشخیص نفوذ استفاده‌شده‌اند.

الگوریتم‌های مبتنی بر هوش جمعی از رفتار جمعی سیستم‌های غیرمتمرکز استفاده‌می‌کنند و مقیاس‌پذیری، سازگاری و استحکام جمعی را در حل مسائل پیچیده فراهم‌می‌کنند [۱۵۴]. استفاده از الگوریتم‌های هوش جمعی مزایای زیادی دارد ازجمله:

۱. دسترسی به دانش جمعی: با استفاده از الگوریتم‌های هوش جمعی، می‌توان از دانش و تجربه یک گروه از افراد بهره‌برداری‌کرد. این

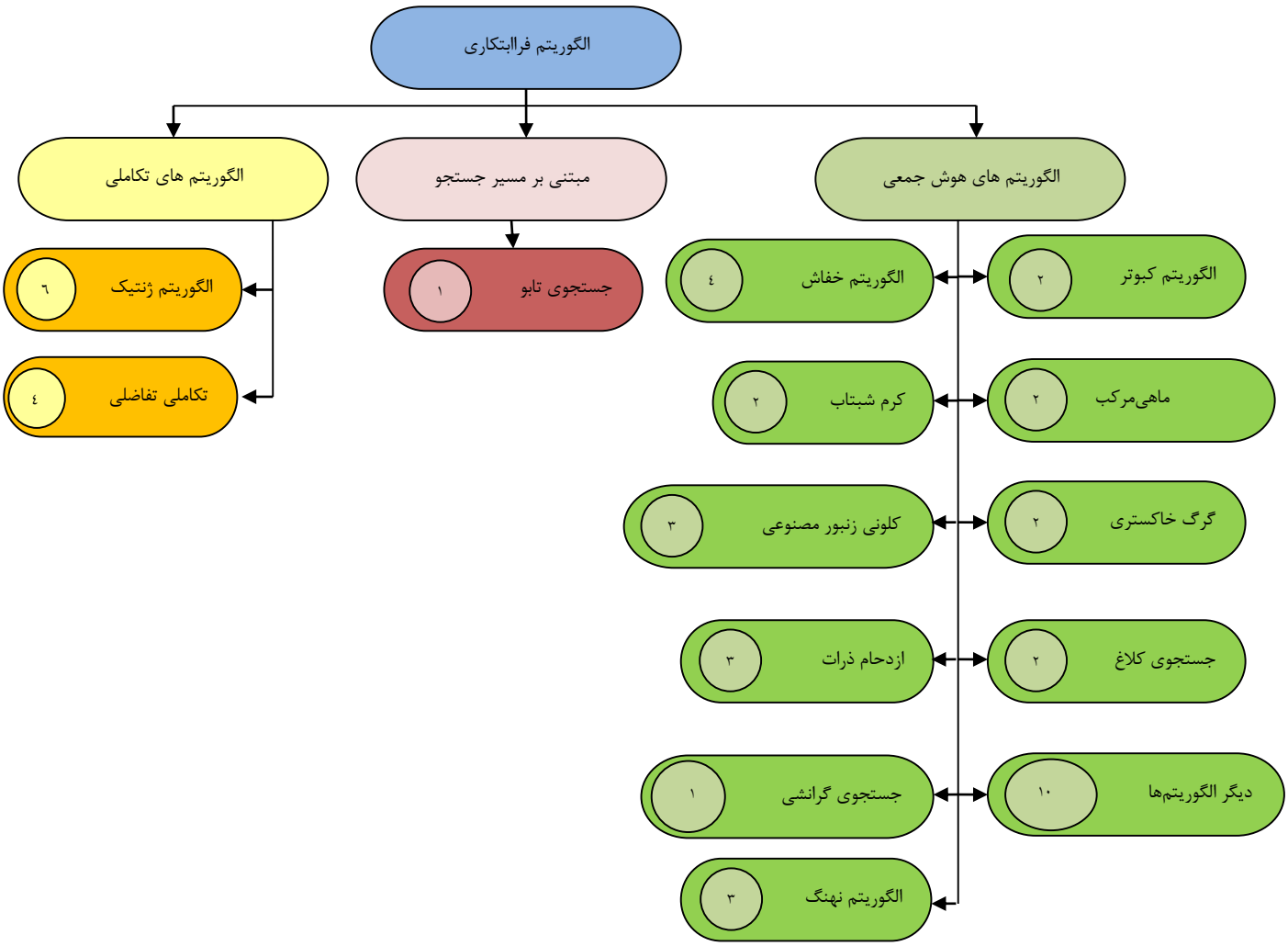
الگوریتم‌ها از تلاش و همکاری گروهی افراد برای حل مسئله استفاده‌می‌کنند و درنتیجه، دسترسی به دانش و تجربه بیشتری نسبت به روش‌های تک‌شخصی فراهم‌می‌کنند.

۲. بهبود کارایی و کیفیت: با استفاده از الگوریتم‌های هوش جمعی، می‌توان بهبود کارایی و کیفیت حل مسائل را به‌دست‌آورد. این الگوریتم‌ها با ترکیب ایده‌ها و راه‌حل‌های مختلف اعضای گروه، به یافتن راه‌حل بهینه و بهتری برای مسئله کمک‌می‌کنند.

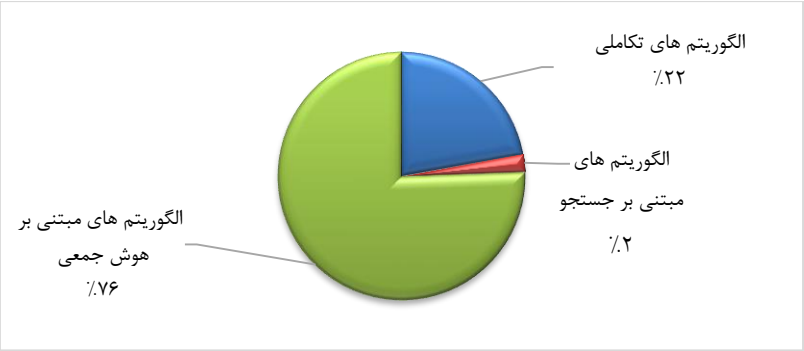
۳. مقاومت در برابر خطا: الگوریتم‌های هوش جمعی، به دلیل استفاده از چندین منبع دانش و تجربه، مقاومت بالاتری در برابر خطا دارند. اگر یکی از اعضای گروه اشتباه‌کند یا اطلاعات نادرستی ارائه‌دهد، سایر اعضا می‌توانند با ترکیب دانش خود، این خطا را تشخیص‌داده و راه‌حل درست را پیداکنند.

۴. انعطاف‌پذیری: الگوریتم‌های هوش جمعی از انعطاف‌پذیری بالایی برخوردار و قادرند با تغییرات در شرایط و متغیرهای ورودی، بهبود و تطبیق بهتری را ارائه‌دهند. انعطاف‌پذیری به این معناست که الگوریتم‌ها می‌توانند با تغییرات و اضافه‌شدن عناصر جدید، به نتیجه بهتری برسند.

شکل (۴) تعداد تحقیق بر اساس الگوریتم‌های فراابتکاری را نشان-می‌دهد. الگوریتم ژنتیک با ۱۳٪ بیشترین استفاده را در تحقیقات بررسی شده دارد (با ۶ مقاله). ا لگوریتم ژنتیک یکی از قدیمی‌ترین و معروف‌ترین الگوریتم‌های بهینه‌سازی فراابتکاری است که از رفتار و اصول انتخاب طبیعی در فرآیند تکاملی پیروی می‌کند. این الگوریتم از مکانیزم تکامل گیاهان و جانوران در طبیعت الهام گرفته‌است، یعنی در هر نسل، افراد با ویژگی‌های بهتر از نسل قبلی برای جمعیت انتخاب‌می‌شوند و از طریق ترکیب و جابجایی این ویژگی‌ها، نسل‌های بعدی به بهبود و بهینه‌سازی پیوسته می‌رسند. در این الگوریتم، هر فرد به‌عنوان یک کروموزوم نماینده از راه‌حل‌ها در فضای جستجو است و با استفاده از عملیات مانند انتخاب، ترکیب، و جابجایی، فرزندان جدیدی ایجادمی‌شوند که ویژگی‌های بهتری نسبت به والدین خود دارند. این فرآیند دائم تکرار می‌شود تا به یک حل بهینه یا حداقل بهینه‌تر برای مسئله مورد نظر برسیم. تحقیقات [۶۴،۵۱] از ژنتیک استاندارد استفاده‌می‌کنند و در [۶۶ ، ۵۰] از ژنتیک چندهدفه استفاده‌شده‌است.



شکل ۲: طبقه بندی الگوریتم‌های فراابتکاری و فراوانی آن‌ها در مقالات بررسی‌شده



شکل ۳: درصد استفاده از دسته‌های مختلف الگوریتم‌های فراابتکاری در مطالعات بررسی‌شده

Dogukan Aksu و همکارش [۵۹] ژنتیک استاندارد را تغییر دادند طوری که موجب افزایش تنوع جمعیت و سرعت رسیدن به راه‌حل بهینه می‌شود. Nilesh Kunhare و همکارانش [۶۲] از الگوریتم ژنتیک برای انتخاب ویژگی در سیستم تشخیص نفوذ و از الگوریتم گرگ خاکستری برای بهینه‌سازی زیرمجموعه انتخابی استفاده‌کردند.

الگوریتم بهینه‌سازی خفاش و الگوریتم تکاملی تفاضلی با ۹٪ دومین الگوریتم فراابتکاری است که بیشترین استفاده را در مقالات مورد بررسی داشته‌اند.

الگوریتم خفاش، الهام‌گرفته از رفتار و روش شکار خفاش‌ها در طبیعت است. در این الگوریتم، خفاش‌ها برای شکار طعمه از یک روش پیچیده و هوشمندانه استفاده‌می‌کنند که شامل حرکت‌های تصادفی و جستجوی هوشمندانه در فضای سه‌بعدی است. خفاش‌ها باید به‌دنبال طعمه در فضای سه‌بعدی حرکت‌کنند و با توجه به فاصله خود از طعمه و جذابیت طعمه، حرکات مناسبی را انجام‌دهند تا به بهترین نقطه در فضا برسند.

Yuyang Zhou و همکارانش [۳۴] ویژگی‌های با اهمیت را بر اساس ترکیب الگوریتم خفاش و همبستگی بین ویژگی‌ها انتخاب‌می‌کنند. آن‌ها از CFS به‌عنوان تابع هدف استفاده‌کردند و معتقدند CFS باعث بهبود دقت طبقه‌بندی می‌شود و برای بهینه‌سازی مراحل آموزش و آزمون مفید است.

Waheed Ali H. M. Ghanem و همکارانش [۵۸] یک چارچوب دو مرحله‌ای با استفاده از یک الگوریتم BAT چندهدفه برای انتخاب ویژگی و یک BAT پیشرفته برای آموزش شبکه عصبی mlp ارائه‌کرده‌اند.

Bukola Fatimah Balogun و همکارانش [۵۴] برای انتخاب ویژگی در سیستم تشخیص نفوذ ابتدا از الگوریتم خفاش، سپس RNS برای انتخاب ویژگی از باقی‌مانده ویژگی‌ها استفاده‌کردند و پس از آن PCA آخرین مرحله انتخاب ویژگی است.

Maheswari S و Arunesh K [۷۱] از الگوریتم خفاش باینری برای انتخاب ویژگی استفاده‌کردند.

الگوریتم تکاملی تفاضلی (DE) ابتدا یک جمعیت از بردارها را تصادفی ایجادمی‌کند و سپس با استفاده از عملیات تفاضلی و ترکیب این بردارها، به بهبود و بهینه‌سازی آن‌ها می‌پردازد. در هر مرحله از الگوریتم، بردارهای جدیدی با استفاده از تفاضل بین بردارهای موجود در جمعیت ایجادمی‌شوند که با اعمال عملیات تفاضلی و انتخاب بهترین بردارها، جمعیت در هر مرحله به سمت بهبود و بهینه‌سازی حرکت‌می‌کند. این الگوریتم از ویژگی‌هایی مانند سادگی پیاده‌سازی، قابلیت تنظیم پارامترها و قابلیت مقیاس‌پذیری برخوردار است که آن را به یکی از الگوریتم‌های پرکاربرد و مؤثر در حل مسائل بهینه‌سازی می‌سازد.

محمد فریس و همکارانش [۴۸] از الگوریتم تکاملی تفاضلی اصلاح‌شده و برای کاهش همگرایی زودرس این الگوریتم از عملیات جهش جدید استفاده‌کرده‌اند.

RUIZHE ZHAO و همکاران [۶۰] برای کاهش ابعاد مسئله تشخیص نفوذ ابتدا واریانس هر ویژگی را محاسبه‌کردند سپس ویژگی‌هایی که واریانس آن‌ها زیر مقدار آستانه باشد را حذف و در نهایت با استفاده از الگوریتم DE ویژگی‌های مناسب را از بین مجموعه ویژگی باقیمانده انتخاب‌کردند.

Ranjit Panigrahi و همکاران [۵۷] از تکاملی تفاضلی چندهدفه استفاده‌کرده‌اند.

Faezah Hamad Almasoudy و همکاران [۶۵] از DE استاندارد برای انتخاب ویژگی استفاده کرده‌اند.

الگوریتم بهینه‌سازی نهنگ، الگوریتم کلونی زنبور عسل و الگوریتم ازدحام ذرات با ۷٪ سومین الگوریتم فراابتکاری است

الگوریتم بهینه‌سازی نهنگ‌ها (Whale Optimization Algorithm - WOA) یک روش بهینه‌سازی فراابتکاری است که از رفتار شکار نهنگ‌های گوژپشت الهام‌گرفته‌است. این نهنگ‌ها از روش خاص "حباب شبکه‌ای" برای شکار طعمه استفاده‌می‌کنند. در این روش، نهنگ‌ها به‌صورت دایره‌ای دور طعمه می‌چرخند و حباب‌هایی را ایجاد می‌کنند تا طعمه را به سمت سطح آب ببرند. این الگوریتم به‌دلیل قابلیت تطبیق با مسائل پیچیده و فضاهای جستجوی بزرگ، در بسیاری از مسائل بهینه‌سازی استفاده‌می‌شود.

Vaishali Ravindranath و همکاران [۷۳] یک نسخه بهبود یافته از الگوریتم بهینه‌سازی نهنگ باینری را برای انتخاب ویژگی ارائه و برای به‌روزرسانی موقعیت نهنگ‌ها از همبستگی جدید و دقت تشخیص طبقه‌بندی استفاده‌کردند.

Riyadh Rahef Nuiaa Al Ogaili و همکاران [۷۴] از الگوریتم نهنگ تغییر یافته بر اساس میانگین وزنی برای انتخاب ویژگی جهت شناسایی حملات بدافزار استفاده‌کرده‌اند. ویژگی‌هایی که وزن بالاتری دارند، انتخاب و ویژگی‌هایی که وزن کمتری دارند حذف‌می‌شوند.

[۷۲] از ترکیب الگوریتم‌های ژنتیک و نهنگ استفاده‌کرده‌است. برای بهبود فضای جستجوی نهنگ از عملگر متقاطع و برای جلوگیری از گیر افتادن در بهینه محلی از عملگر جهش ژنتیک استفاده‌شده‌است.

الگوریتم کلونی زنبورها (Bee Colony Algorithm) از رفتار جستجوی غذا توسط زنبورهای عسل الهام گرفته‌است. این الگوریتم بر اساس سه نوع زنبور کارمی‌کند: زنبورهای جستجوگر، زنبورهای مشاهده‌گر و زنبورهای کارگر. زنبورهای جستجوگر به‌صورت تصادفی منابع غذایی جدید را در فضای جستجو پیدامی‌کنند و زنبورهای کارگر به منابع غذایی که قبلاً شناسایی‌شده‌اند، بازمی‌گردند و اطلاعات مربوط به کیفیت و موقعیت این منابع را به کندو گزارش‌می‌دهند. زنبورهای مشاهده‌گر بر اساس اطلاعات دریافتی از زنبورهای کارگر، تصمیم‌می‌گیرند که به کدام منبع غذایی بروند. این فرایند باعث‌می‌شود زنبورهای بیشتری به سمت منابع غذایی با کیفیت بالا هدایت‌شوند و به‌مرور زمان بهترین منابع غذایی (راه‌حل‌های بهینه) شناسایی‌شوند. توانایی الگوریتم زنبورها در کاوش کارآمد فضاهای راه‌حل پیچیده و

یافتن بهینه جهانی، آن را به ابزاری قدرتمند در تحقیقات و برنامه‌های کاربردی هوش محاسباتی مدرن تبدیل می‌کند.

Arun Kumar Sangaiah و همکاران [۴۳] یک انتخاب ویژگی ترکیبی ابتکاری برای تشخیص نفوذ در محیط‌های ابری با استفاده از الگوریتم‌های بهینه‌سازی کلونی مورچه‌ها و زنبورها پیشنهادکردند. آن‌ها ابتدا با استفاده از الگوریتم زنبور برای یافتن نگاشت عددی استفاده کرده، سپس با استفاده از ACO پاسخ‌های تقریبی را به‌دست‌می‌آورند.

Milos Stankovic و همکاران [۵۶] استفاده از ترکیب الگوریتم کلونی زنبورهای مصنوعی و کرم شبتاب را برای انتخاب ویژگی پیشنهادمی‌کنند. با توجه به یک عدد تصادفی، راه‌حل‌ها بر اساس الگوریتم‌های زنبور و کرم شبتاب به‌روزرسانی می‌شوند.

S. Velliangiri1 • P. Karthikeyan [۶۹] ترکیب کلونی زنبورهای مصنوعی تطبیقی با بهینه‌سازی ازدحام ذرات تطبیقی برای تشخیص فعالیت‌های نفوذی را طراحی‌کرده‌اند که در هر مرحله این دو الگوریتم نتایج خود را با یکدیگر به اشتراک می‌گذارند.

الگوریتم ازدحام ذرات (Particle Swarm Optimization - PSO) از رفتار اجتماعی گروهی از ذرات الهام گرفته‌است، مشابه رفتار پرندگان در هنگام پرواز دسته‌جمعی یا ماهی‌ها در شنا کردن گروهی. در این الگوریتم، هر ذره نماینده یک راه‌حل بالقوه است که در فضای جستجو حرکت‌می‌کند و موقعیت خود را بر اساس تجربه خود و همچنین بهترین موقعیتی که توسط دیگر ذرات مشاهده‌شده‌است، به‌روزرسانی می‌کند. PSO از دو مؤلفه اصلی، یعنی سرعت و موقعیت، برای هر ذره استفاده می‌کند و با به‌روزرسانی مداوم این مقادیر، به سمت بهترین راه‌حل همگرا می‌شود.

Shalini Subramani, M. Selvi [۴۵] ، از ازدحام ذرات چندهدفه استفاده‌کرده‌اند. Wisam Elmasry و دوستانش [۶۷] ازدحام ذرات دوگانه باینری را برای انتخاب زیرمجموعه ویژگی‌ها و فراپارامترهای مدل پیشنهادکردند.Roseline Oluwaseun Ogundokun و همکاران [۶۸] برای انتخاب ویژگی از ازدحام ذرات باینری، قبل از طبقه‌بندی ترافیک‌های نرمال و غیرنرمال استفاده‌کرده‌اند.

برای هر کدام از الگوریتم‌های کبوتر، کلاغ، گرگ خاکستری، کرم شبتاب و ماهی مرکب، ۲ مقاله بررسی‌شده‌است بنابراین ۴٪ از مقالات را شامل-می‌شود.

الگوریتم کبوتر، یک روش بهینه‌سازی مستقیم الهام‌گرفته از رفتار و روش جستجوی غذا در کبوترها در طبیعت می‌باشد. در این الگوریتم، کبوترها به‌دنبال یافتن منابع غذایی در محیط خود می‌گردند و با استفاده از روش‌های جستجویی مؤثر و هوشمند، به بهینه‌سازی جستجوی خود می‌پردازند. این الگوریتم بر اساس اصول رفتاری و جستجویی کبوترها طراحی شده‌است، هر کبوتر به‌عنوان یک نقطه در فضای جستجو در نظر گرفته‌می‌شود و با حرکت به سمت منابع غذایی و استفاده از اطلاعات محیط، به بهترین نقطه در فضا هدایت‌می‌شود.

Orieb Abu Alghanam و همکاران [۴۶] یک الگوریتم کبوتر جستجوی محلی پیشرفته پیشنهادداده‌اند.

Hadeel Alazzam و همکاران [۷۰] از یک روش جدید برای باینری کردن الگوریتم کبوتر استفاده‌کردند.

در الگوریتم کلاغ، کلاغ‌ها به‌دنبال یافتن منابع غذایی در محیط خود می‌گردند و با استفاده از روش‌های جستجویی خاص، به بهینه‌سازی جستجوی خود می‌پردازند.

D. Jayalatchumy و همکارانش [۴۱] الگوریتم جستجوی کلاغ بهبود یافته ارائه‌داده و با استفاده از تکنیک آشوب، جستجوی محلی الگوریتم کلاغ را بهبود دادند. Ashish Khanna و همکارانش [۶۳] نیز از الگوریتم کلاغ باینری برای انتخاب ویژگی استفاده‌کرده‌اند.

در الگوریتم گرگ خاکستری، گرگ‌ها به‌دنبال شکار طعمه در محیط خود می‌گردند و با استفاده از تاکتیک‌ها و استراتژی‌های مختلف، سعی در بهینه‌سازی شکار خود دارند. هر گرگ خاکستری در الگوریتم به‌عنوان یک نقطه در فضای جستجو درنظر گرفته‌می‌شود و با استفاده از قوانین و اصول رفتاری خود، به بهترین نقطه در فضا هدایت‌می‌شود. این الگوریتم بر اساس اصول تکاملی و انتخاب طبیعی طراحی‌شده‌است، گرگ‌های با ویژگی‌های بهتر و مطابق با محیط خود برای شکار، انتخاب-می‌شوند و با ترکیب و جابجایی ویژگی‌های مختلف، نسل‌های بهتری ایجادمی‌شود.

Qusay M. Alzubi و همکاران [۷۵] گرگ خاکستری باینری اصلاح-شده پیشنهاددادند. آن‌ها موقعیت گرگ‌ها را بر اساس چهار بهترین راه-حل α، β، δو ω آپدیت می‌کنند یعنی گرگ امگا نیز در تغییر موقعیت گرگ‌ها مؤثر است. Taief Alaa Alamiedy و همکارانش [۷۶] از الگوریتم بهینه‌سازی گرگ خاکستری چندهدفه برای انتخاب ویژگی استفاده‌کرده‌اند.

الگوریتم کرم شب‌تاب یک الگوریتم بهینه‌ساز فراابتکاری است که از رفتار چشمک‌زن کرم‌های شب‌تاب الهام گرفته و مبتنی بر جمعیت است.

Selvakumar B و دوستش [۳۳] از الگوریتم کرم شب‌تاب مبتنی بر Mutual Information (MI) استفاده‌کردند. Yakub Kayode Saheed [۵۳] الگوریتم کرم شبتاب را با استفاده از تابع سیگموئید باینری کرده و برای انتخاب ویژگی در سیستم تشخیص نفوذ استفاده‌کرده‌است.

الگوریتم ماهی مرکب(Cuttlefish Algorithm - CFA) یک الگوریتم بهینه‌سازی فراابتکاری است که از رفتار و ویژگی‌های ماهی مرکب الهام گرفته‌است. الگوریتم از توانایی ماهی مرکب در تغییر رنگ و الگوهای بدن خود برای به‌روزرسانی موقعیت راه‌حل‌ها استفاده‌می‌کند. این تغییر رنگ به معنای تغییر در موقعیت راه‌حل‌ها در فضای جستجو است تا به نقاط بهینه نزدیک‌تر شوند. ماهی‌های مرکب در طبیعت از استتار برای مخفی‌شدن از دید شکارچیان و جلب توجه طعمه‌ها استفاده‌می‌کنند. در الگوریتم، این رفتار به‌عنوان جستجوی محلی برای بهبود راه‌حل‌های فعلی مدل‌سازی می‌شود. ترکیب راه‌حل‌ها (مشابه جفت‌گیری در طبیعت) و جهش‌های تصادفی به‌منظور افزایش تنوع جمعیت و

جلوگیری از گیرافتادن در بهینه‌های محلی استفاده‌می‌شود. در هر تکرار، راه‌حل‌های جدید با راه‌حل‌های قبلی مقایسه‌می‌شوند و بهترین‌ها برای تکرارهای بعدی انتخاب‌می‌شوند.

سارا محمدی و همکاران [۳۵] ترکیبی از روش فیلتر و روش بسته‌بند به‌عنوان انتخاب ویژگی پیشنهادداده و ابتدا بر اساس ضریب همبستگی خطی، ویژگی‌ها را گروه‌بندی می‌کنند سپس با الگوریتم ماهی مرکب (CFA) ویژگی‌های نامربوط و اضافی را از مجموعه داده اصلی حذف‌می‌کنند.

Sharma و همکارانش [۵۵] االگوریتم ماهی مرکب اصلاح شده‌ای را پیشنهاد داده‌اند که از دو فرآیند اصلی تشکیل شده‌است: فرآیند دید و فرآیند بازتاب. این فرآیندها به یافتن راه‌حل بهینه جهانی کمک می‌کنند و راه‌حل جدید نتیجه ترکیب فرآیندهای دید و بازتاب است.

در این بین مقالاتی وجوددارند که از دیگر الگوریتم‌های فراابتکاری استفاده‌کرده‌اند. Anjum Nazir و همکارش [۳۶] از جستجوی تابو برای انتخاب ویژگی استفاده‌کردند. Nojood O. Aljehane و همکارانش [۳۹] جستجوی گرانشی باینری را برای انتخاب ویژگی به‌کاربردند.

Selva Rani و همکاران [۱۱۴] ترکیبی از الگوریتم بهینه‌سازی ارشمیدس (AOA) و بهینه‌سازی شاهین آتشین (FHO) ارائه‌کردند.

Anil V Turukmane و همکار [۳۸] ویژگی‌های استخراج‌شده را توسط الگوریتم بهینه‌سازی گوشاوک شمالی مبتنی بر مخالفت (ONgO) بهینه‌سازی‌کردند. Laith Abualigah و همکارانش [۴۰] از الگوریتم آکوآ اصلاح‌شده (mAO) برای سیستم‌های تشخیص نفوذ استفاده‌کردند.

Tingyao Jiang و همکارانش [۴۲] الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی را بهبوددادند. آن‌ها برای کنترل عملیات اپراتورهای مهاجرت و جهش از انتخاب رولت استفاده‌می‌کنند.

Zhiwei Ye و همکارانش [۴۴] از یادگیری مبتنی بر مخالفت نخبگان برای بهبود الگوریتم بهینه‌سازی پرورش ترکیبی (HBO) استفاده‌کردند تا جستجوی بهینه‌تری داشته‌باشد.

Malek Barhoushو همکارانش [۴۷] روش‌های یادگیری مبتنی بر مخالفت، یادگیری مبتنی بر مخالفت نخبگان و روش جستجوی همسایگی متغیر، الگوریتم ازدحام سالپ (SSA) را بهبوددادند.

K.G. Maheswari و همکاران [۴۹] از بهینه‌سازی تغذیه پرتوی اصلاح‌شده، استفاده‌کردند.

Silvio E. Quincozes و همکارانش [۵۲] از روش جستجوی تطبیقی تصادفی حریصانه (GRASP) برای انتخاب ویژگی استفاده‌کرده‌اند.

Junwen Chen و همکارانش [۶۱] از الگوریتم ترکیب شیر مورچه و محاسبات کوانتومی استفاده کرده‌اند.

5. تأثیر انتخاب ویژگی مبتنی بر الگوریتم‌های فرابتکاری در IDS

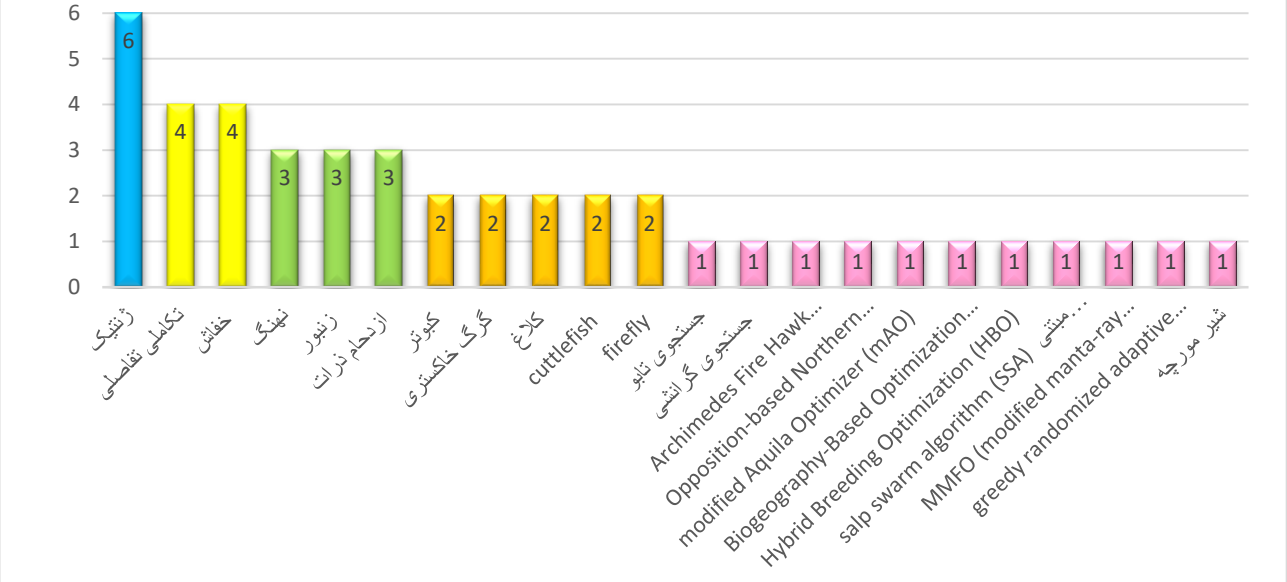
برای پاسخ به پرسش دوم ، نیاز است که معیارهای ارزیابی مقالات بررسی‌شود. ازآنجاکه در مقالات گذشته، روی مجموعه داده‌های مختلف کار شده‌است، ابتدا مجموعه داده‌های استفاده‌شده در مقالات را بررسی‌می‌کنیم سپس به معیارهای ارزیابی می‌پردازیم تا بتوانیم الگوریتم‌های فراابتکاری برای انتخاب ویژگی در تشخیص نفوذ را مقایسه‌کنیم.

1.5. مجموعه داده ها

یکی از مسائل مهم هنگام برخورد با رویکردهای انتخاب ویژگی نظارت‌شده در تحلیل ترافیک، یافتن مجموعه‌های آموزشی است که هم جدید باشد و هم برچسب‌گذاری شده‌باشد [۱۰]. در ادامه به مجموعه داده‌هایی که در تحقیقات گذشته استفاده‌شده‌است می‌پردازیم.

UNSW-NB : این مجموعه داده توسط چهار ابزار IXIA PerfectStorm، Tcpdump، Argus و Bro-IDS تولیدشده‌است. این ابزارها برای ایجاد برخی از انواع حملات از جمله DoS، Exploits، Shellcode، Reconnaissance، Generic و Worms استفاده‌می‌شوند [۲۱].KDD99 : این مجموعه داده براساس برنامه تخمین IDS DARPA’98 ایجاد شده‌است و ترافیک شبکه به مدت هفت هفته است. شامل ۴۹۰۰۰۰۰ رکورد است. حملات در گروه‌های زیر دسته‌بندی می شوند: حمله کاربر به ریشه (U2R)، حمله انکار سرویس (DoS)، حمله از راه دور به محلی (R2L)، و حمله کاوشگر. مجموعه داده KDD Cup 1999 شامل ۴۱ ویژگی است که به سه کلاس زیر طبقه‌بندی‌می‌شوند: ویژگی‌های اساسی، ویژگی‌های محتوا، و ویژگی‌های ترافیک [۲۱]-NSL KDD: این مورد توسط تولایی و همکاران پیشنهادشده‌است [۲۲] که منجر به حل‌شدن برخی از مشکلات مجموعه داده‌های KDD’99 شد. آن‌ها مجموعه داده‌های KDD را در موارد زیر بهبوددادند: بدون افزونگی، ورودی‌های تکراری وجودندارد، تعداد ورودی‌های انتخابی سازمان‌دهی شده‌است، و تعداد ورودی‌ها معقول است. مقالات زیادی در مورد تشخیص نفوذ وجوددارد که از هر دو مجموعه داده با هم در تکامل عملکرد استفاده‌می‌کنند و دریافتند که بهترین نتایج در NSL-KDD یافت‌می‌شود [۲۱].CDMC2012 : یکی دیگر از مجموعه داده‌های محبوب CDMC2012 [۲۳] است که با استفاده از چندین Honeypot از پنج شبکه مختلف ایجادشد. در میان نمونه‌های CDMC2012، مواردی با عنوان «ناشناخته» وجوددارد که صرفاً مواردی که به‌عنوان «حمله» یا «عادی» برچسب‌گذاری شده‌بودند، کنار گذاشته‌شدند. بدون از دست دادن کلیت، این مجموعه داده به دو مجموعه تقسیم‌شد، یک مجموعه آموزشی شامل ۴۸۳۵۷ نمونه و یک مجموعه تست با ۸۰۰۰۰. هر نمونه با ۱۴ ویژگی، از جمله کلاس، نشان‌داده‌می‌شود [۱۵]. CICIDS2017: این مجموعه داده حاوی داده‌هایی است که در همان دوره از مجموعه داده‌های CIC DoS گرفته‌شده‌است. CICIDS2017 توسط شرف‌الدین و همکاران پیشنهاد شده‌است [۲۴].

حملات شامل Brute Force SSH، DoS، Heartbleed، Web Attack، Infiltration، Botnet و DDoS، و Brute Force FTP هستند [۲۱].AWID : مجموعه داده AWID [۲۵] شامل گروهی از مجموعه داده‌های بزرگ مانند AWID-CLS-R-Trn، AWID-CLS-R-Tst، AWID-CLS-F-Trn، AWID-CLS-F-Tst است. در آزمایش تحقیق [۲۰]، مجموعه داده AWID-CLS-R-Trn انتخاب‌شده‌است که در آن "R" یک مجموعه داده کاهش یافته را نشان‌می‌دهد. AWID همچنین دارای مجموعه AWID-ATK-R-Trn است که در آن "ATK" شامل ۱۶ کلاس هدف است. ۱۶ نوع کلاس هدف حمله در



شکل ۴: تعداد تحقیق بر اساس الگوریتم‌های فراابتکاری در مطالعات بررسی‌شده

مجموعه داده "ATK" با جزئیات بیشتر مانند Evil Twins، Honeypot، EvilTwins و سایر حملات موجود است. داده‌های "CLS" دارای نسخه فشرده دسته‌های حمله هستند که عبارتند از تزریق، سیل، حمله جعل هویت و کلاس عادی که در مجموعه داده "ATK" فهرست شده اند [۲۰].

BoT-IoT: مجموعه داده BOT-IOT [۱۱۴] با طراحی یک محیط شبکه واقعی در آزمایشگاه Cyber Range در UNSW Canberra ایجادشد. محیط شبکه ترکیبی از ترافیک عادی و بات نت را در خود جای داده‌است. این مجموعه داده دارای ۴۵ ویژگی و 3668522 رکورد است. فایل‌ها بر اساس دسته و زیرمجموعه حمله جداشدند تا در فرآیند برچسب‌گذاری بهتر کمک‌شود. مجموعه داده شامل حملات DDoS، DoS، OS و Service Scan، Keylogging و Data Exfiltration است که حملات DDoS و DoS بر اساس پروتکل مورد استفاده بیشتر سازماندهی‌شده‌است.

CSE-CIC-IDS 2018: مجموعه داده CSE-CIC-IDS 2018 یک مجموعه داده معیار برای سیستم‌های تشخیص نفوذ (IDS) در اینترنت اشیا (IoT) است [۳۸]. این مجموعه داده شامل بیش از ۸۰ میلیون رکورد، با ترکیبی از ترافیک عادی و مخرب از انواع دستگاه‌ها و پروتکل‌های IoT است. مجموعه داده شامل ترافیک ناشی از حملات شبیه‌سازی شده، مانند حملات Botnet، Mirai و Denial-of-Service (DoS) است. مجموعه داده CSE-CIC-IDS 2018 شامل ۸۰ ویژگی است که از ضبط ترافیک شبکه استخراج شده‌است. SWat: داده‌های گرد-آوری‌شده از بستر آزمایش شامل ۱۱ روز کار مداوم است. داده‌های نرمال طی ۷ روز جمع‌آوری‌شد و داده با سناریوهای ۴۱ حمله ۴ روز جمع‌آوری-شد [۵۲]. در طول کار، کلیه داده‌های ترافیک شبکه، حسگر و محرک گردآوری‌شد. ترافیک شبکه و تمام مقادیر به‌دست‌آمده از ۵۱ حسگر و محرک جمع‌آوری‌شده‌است. داده‌ها بر اساس رفتارهای عادی و غیرعادی برچسبگذاری شده‌اند.ISCX2012 مرکز امنیت اطلاعات تعالی این مجموعه داده را ارائه‌کرده‌است. ISCX شامل ۲۰ ویژگی و ۱۵۱۲۰۰۰ رکورد است که فعالیت‌های شبکه را به مدت ۷ روز دربرمی‌گیرد. [۵۸]

HCRL-car hacking مجموعه داده‌های هک خودرو شامل حمله DoS، حمله فازی، جعل چرخ‌دنده درایو، و جعل RPM gauge است.

مجموعه داده‌ها با ثبت ترافیک CAN از طریق پورت OBD-II از یک وسیله نقلیه واقعی درحالی‌که حملات تزریق پیام انجام‌می‌شد، ساخته‌شدند. مجموعه داده‌ها شامل ۳۰۰ نفوذ از تزریق پیام‌ند. هر نفوذ به مدت ۳ تا ۵ ثانیه انجام‌می‌شود و هر مجموعه داده در کل ۳۰ تا ۴۰ دقیقه از ترافیک CAN را شامل‌می‌شود [۵۹]. CIRA-CIC-DOHBrw-2020مجموعه داده CIRA-CIC-DoHBrw-2020 ترافیک DoH عادی و مخرب را همراه با ترافیک غیر DoH ثبت‌می‌کند. این مجموعه داده دارای ۳۴ ویژگی و چهار کلاس است. تعداد نمونه‌های این مجموعه داده حدود ۱,۴ میلیون است [۶۴].

ADFA-LD مجموعه داده ADFA-LD دارای دو برچسب عادی و حمله است که حملاتش به گروه‌های adduser, Hydra-FTP,Hydra-

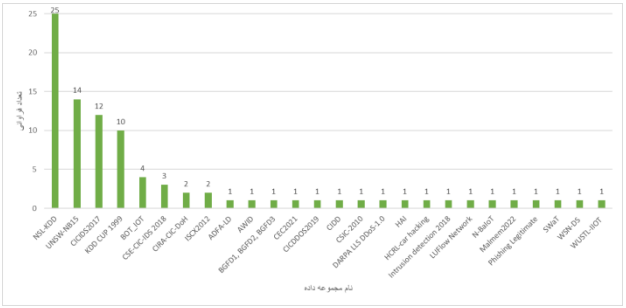
SSH, Java-metepreter, metepreter و Web-Shell دسته‌بندی شده‌اند. در مجموع ۴۴ ویژگی و ۱،۵۷۹ نمونه دارد [۷۲]. مجموعه داده‌های دیگری وجوددارد که مقالات کمتری استفاده‌شده‌اند مانند CEC2021, CIDD, WUSTL-IIOT and HAI که در [۴۴] و-CICDDOS2019, CIRA , CIC- CIRA-CIC-DoH, Intrusion detection 2018, Phishing Legitimate, Malmem2022, N-BaIoT, LUFlow Network در [۴۷] و DARPA LLS DDoS-1.0 و CSIC-2010 در [۴۹]، BGFD1و BGFD2 وBGFD3 در [۵۰] و ISCXURL-2016 در [۷۴]. جدول ۴ لیست مجموعه داده‌ها به همراه فراوانی آن‌ها در مقالات بررسی‌شده را نشان‌می‌دهد.

مجموعه داده NSL-KDD با ۲۵ فراوانی بیشترین استفاده را دارد و سپس مجموعه داده UNSW-NB15 با ۱۴ فراوانی در رتبه دوم بیشترین آزمایش را داراست.

جدول۴: مجموعه داده ها و فراوانی آن‌ها

Dataset Name	تعداد	مراجع
ADFA-LD	1	[72]
AWID	1	34
BGFD1, BGFD2, BGFD3	1	50
BOT_IOT	4	23,51,64,37
CEC2021	1	44
CICDDOS2019	1	47
CICIDS2017	12	34,42,47,49,55,56,57,58,59,66,67,72
CIDD	1	45
CIRA-CIC-DoH	1	47,64
CSE-CIC-IDS 2018	3	38, 47, 60
CSIC-2010	1	49
DARPA LLS DDoS-1.0	1	49
HAI	1	44
HCRL-car hacking	1	59
Intrusion detection 2018	1	47
ISCX2012	2	58, 74
KDD CUP 1999	10	33, 34,35, 45,46, 50, 58, 68, 70, 71
LUFlow Network	1	47
N-BaIoT	1	47
Malmem2022	1	47
NSL-KDD	25	37,40,41,42,43,44,45,46, 47,48,50,54,58,60,61,62, 63,65,66,67,69,70,73,75,76
Phishing Legitimate	1	47
SWaT	1	52
UNSW-NB15 UNSW-NB16	14	36,38,39,41,46,47,53,56, 58,59,63,64,66,70
WSN-DS	1	52
WUSTL-IIOT	1	44

شکل ۵ نمودار فراوانی مجموعه داده را نشان‌می‌دهد.



شکل ۵: فراوانی مجموعه داده

بنابراین مجموعه داده‌های محبوب به‌ترتیب NSL-KD، UNSW-NB15، CICIDS2017، KDD CUP 1999، BOT_IOT، CSE-CIC-IDS 2018 هستند و دیگر مجموعه داده‌ها شامل ISCX2012، ADFA-LD، AWID، BGFD1, BGFD2, BGFD3، CEC2021، CICDDOS2019، CIDD، CIRA-CIC-DoH، CSIC-2010، DARPA LLS DDoS-1.0، HAI، HCRL-car hacking، Intrusion detection 2018، LUFlow Network، N-BaIoT، Malmem2022، Phishing Legitimate، SWaT، WSN-DS و WUSTL-IIOT هستند.

۲.۵. معیارهای ارزیابی

در این بخش ابتدا معیارهای ارزیابی متداول را تعریف‌می‌کنیم سپس مقادیر به این معیارهای ارزیابی را در مقالات منتخب بررسی‌می‌کنیم.

مهم‌ترین معیار برای تعیین کارایی یک سیستم تشخیص نفوذ، معیار Accuracy می‌باشد. دقت کلی تشخیص نفوذ، به معنای میزان درستی سیستم تشخیص نفوذ (IDS) در تشخیص و شناسایی تلاش‌های نفوذ و حملات به شبکه یا سیستم است. این دقت معمولاً به عنوان نسبت تعداد حملات و نفوذهای صحیح شناسایی‌شده به مجموع حملات واقعی در سیستم تعریف‌می‌شود. به‌عبارتی، دقت تشخیص نفوذ نشان‌می‌دهد که چه تعداد از حملات واقعی توسط سیستم تشخیص نفوذ به‌درستی شناسایی‌شده‌اند. رابطه ۱ فرمول Accuracy را نشان‌می‌دهد.

$$Accuracy = \frac{\mathrm{TP+TN}}{\mathrm{TP+FP+FN+TN}} \qquad (۱)$$

Accuracy یک معیار کلی است که همه پیش‌بینی‌های صحیح را درنظرمی‌گیرد، چه حمله باشند و چه نباشند.

در سیستم‌های تشخیص نفوذ، تعداد بالای پیش‌بینی‌های نادرست حمله می‌تواند منجر به هشدارهای بی‌مورد شود که باعث ایجاد اختلال و مصرف منابع می‌شود. در اینجا Precision می‌تواند معیار بهتری برای ارزیابی کارایی سیستم باشد. Precision به نسبت تعداد پیش‌بینی‌های مثبت صحیح به تعداد کل پیش‌بینی‌های مثبت اشاره دارد. به‌عبارت دیگر، Precision نشان‌می‌دهد که از میان تمام نمونه‌هایی که به‌عنوان حمله پیش‌بینی شده‌اند، چند درصد واقعاً حمله بوده‌اند. رابطه ۲ فرمول این معیار را نشان‌می‌دهد. ازآنجاکه معیار Accuracy در شرایطی که کلاس‌ها به‌شدت نامتوازن هستند، می‌تواند گمراه‌کننده باشد بنابراین Precision می‌تواند دید بهتری از عملکرد مدل برای دسته‌بندی نمونه‌های مثبت ارائه‌دهد.

$$Precision = \frac{TP}{\mathrm{TP+FP}} \qquad (۲)$$

Recall یا Sensitivity یا نرخ تشخیص یا TPR (True Positive Rate): به نسبت تعداد نمونه‌های مثبت صحیح شناسایی‌شده به کل تعداد نمونه‌های مثبت واقعی اشاره‌دارد. به‌عبارت‌دیگر، Recall نشان‌می‌دهد که مدل چه تعداد از کل نمونه‌های مثبت را به‌درستی شناسایی‌کرده‌است. این معیار با رابطه (۳) محاسبه‌می‌شود.

$$Recall = \frac{TP}{TP+FN} \qquad (۳)$$

این معیار برای ارزیابی عملکرد مدل در شناسایی تمامی نمونه‌های مثبت اهمیت دارد. به عبارتی Recall یک معیار حیاتی برای ارزیابی کارایی مدل است زیرا نشان‌می‌دهد که مدل چقدر در شناسایی تمامی موارد مثبت، موفق است. شرایطی را در نظر بگیرید که حمله وجود دارد ولی مدل آن را شناسایی نمی‌کند در این صورت سیستم با عواقب جدی مواجه می شود. Recall به ما کمک می‌کند تا مطمئن شویم مدل تا چه حد، همه موارد مثبت را شناسایی می‌کند.

F-measure (که گاهی به عنوان F1-scoreشناخته می‌شود) یکی از معیارهای مهم در ارزیابی عملکرد مدل‌های یادگیری ماشین و به‌ویژه در مسائل تشخیص نفوذ است. این معیار ترکیبی از دو معیار Precision و Recall است و به منظور ایجاد تعادل بین این دو معیار مورد استفاده قرارمی‌گیرد که با رابطه (۴) محاسبه‌می‌شود. F-measure به تغییرات در هر دو معیار Precision و Recall حساس است و می‌تواند تصویری دقیق‌تر از عملکرد مدل در شناسایی صحیح نمونه‌های مثبت ارائه‌دهد.

$$\mathrm{F}-measure = \frac{2*Precision*Recall}{Precision+Recall} \qquad (۴)$$

F-measureبه‌خصوص زمانی مهم است که باید تعادلی بین Precision و Recall برقرار شود .این معیار برای مسائلی که نیازمند بالانس بین نرخ شناسایی صحیح (Recall) و نرخ مثبت‌های صحیح شناسایی شده (Precision)هستند، بسیار مناسب است. در مجموعه داده‌هایی که کلاس‌ها نامتوازن هستند، استفاده از F1-score می‌تواند معیار بهتری نسبت به Accuracy باشد، چرا که F1-scoreتأثیر خطاهای کلاسی که تعداد کمتری دارند را بهتر منعکس‌می‌کند.

نرخ مثبت کاذب (False Positive Rate - FPR) به نسبت تعداد نمونه‌هایی که به‌اشتباه، مثبت شناسایی‌شده‌اند به کل تعداد نمونه‌های واقعی منفی اشاره‌دارد (رابطه ۵). به‌عبارت‌دیگر، FPRنشان‌می‌دهد که چه درصدی از نمونه‌های منفی به‌غلط، مثبت تشخیص‌داده‌شده‌اند. نرخ مثبت کاذب بالا در سیستم‌های تشخیص نفوذ می‌تواند منجر به افزایش تعداد هشدارهای نادرست شود که باعث‌می‌شود اپراتورها زمان و منابع زیادی را صرف بررسی تهدیدات غیرواقعی کنند.

$$\mathrm{FPR}= \frac{FP}{\mathrm{FP+TN}} \qquad (۵)$$

Mathews Correlation Coefficient (MCC) یک معیار آماری است که برای ارزیابی کیفیت دسته‌بندی‌های باینری (دودسته‌ای) استفاده‌می‌شود. این معیار یک مقدار بین -۱ و ۱ تولیدمی‌کند، که ۱ نشان‌دهنده دسته‌بندی کامل، ۰ نشان‌دهنده دسته‌بندی تصادفی و -۱ نشان‌دهنده دسته‌بندی کاملاً اشتباه است. MCC در مقایسه با معیارهای دیگر مانند دقت (Accuracy) و F1-Score جامع‌تر است زیرا همزمان تأثیرات True Positives (TP)، True Negatives (TN)، False Positives (FP) و False Negatives (FN) را درنظرمی‌گیرد. MCC طبق رابطه (۶) به‌دست‌می‌آید.

$$\mathrm{MCC} = \frac{(\mathrm{TP}\times\mathrm{TN})-(\mathrm{FP}\times\mathrm{FN})}{\sqrt{(\mathrm{TP+FP})(\mathrm{TP+FN})(\mathrm{TN+FP})(\mathrm{TN+FN})}} \qquad (۶)$$

زمان اجرای الگوریتم یکی از معیارهای کلیدی برای ارزیابی کارایی و عملکرد یک الگوریتم است. این معیار نشان‌دهنده مدت زمانی است که یک الگوریتم برای حل یک مسئله خاص نیاز دارد.

در این معادلات:

- (TP) (True Positives) به تعداد حملات واقعی که به‌درستی شناسایی‌شده‌اند، اشاره دارد.
- (TN) (True Negatives) به تعداد دسترسی‌های مجاز که به‌درستی به‌عنوان غیرحمله شناسایی‌شده‌اند، اشاره دارد.
- (FP) (False Positives) به تعداد دسترسی‌های مجازی که به‌اشتباه به‌عنوان حمله شناسایی‌شده‌اند اشاره دارد.
- (FN) (False Negatives) به تعداد حملات واقعی که به‌اشتباه به‌عنوان دسترسی مجاز شناسایی‌شده‌اند، اشاره دارد.

این معیارها کمک‌می‌کنند تا کارایی یک سیستم تشخیص نفوذ را ارزیابی‌کرده و نقاط قوت و ضعف آن را مشخص‌کنیم. جدول ۵ معیارهای ارزیابی مختلف در سیستم تشخیص نفوذ را نشان‌می‌دهد.

همه مقالات معیار ارزیابی Accuracy را بررسی‌کرده‌اند و اکثر مقالات معیارهای ارزیابی Precision، Recall و F-measure را دارند. معیارهای دیگری مانند false positive rate (FPR)، specificity، تعداد ویژگی‌های انتخابی، false alarm rate (FAR) و غیره نیز وجوددارند که کمتر به آن‌ها پرداخته‌شده‌است. معیار Accuracy روش‌های بررسی شده از ۶۶٪ تا ۱۰۰٪ است. معیارهای ارزیابی برای هر مجموعه داده‌ها در جدول ۶ آمده‌است. مقادیر خالی در جدول ۶ نشان‌دهنده این است که یا مقدار دقیقی برای معیار ارزیابی مشخص‌نشده‌است یا کلاً آن معیار در مقاله مذکور ارزیابی‌نشده‌است.

جدول ۵: معیارهای ارزیابی مختلف در سیستم تشخیص نفوذ

نام معیار ارزیابی	مراجع	
Accuracy	[34,35,36,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,66,67,68,69,70,71,72,73,74,75,76]	
Precision	[34,114,38,39,40,42,44,45,47,48,49,50,52,53,57,58,59,60,65,67,72,74]	
TPR|recall|Sensitivity|DR	[34,114,38,41,42,44,45,47,46,47,48,49,50,51,52,53,54,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,72,74,75]	
F-measure | F1-score	[33,34,114,38,41,42,44,45,46,47,48,49,50,52,53,54,59,60,61,65,67,70,74]	
FPR|FAR	[33,34,35,36,38,40,45,46,48,49,51,58,61,62,63,65,66,67,68,69,70,75]	
time	زمان ساخت مدل	34
	training time	[38,45]
	testing time	35
	predicting time	51
	execute time	[33,40]
MCC	[38,39]	
number of selected features	[113,40,47,56,67,74,76]	
specificity |TNR	[38,39,48,49,58,69,72]	
AUC	[46,50,57]	
error rate|Missed Rate (MR)	[48,67]	
G-mean	50	
95% CI MST	59	

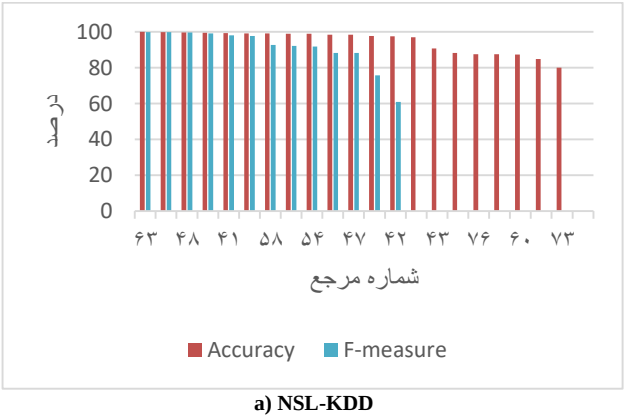
جدول۶: معیارهای ارزیابی مقالات منتخب به تفکیک مجموعه داده‌ها

Dataset Name	مرجع	Accuracy	Precision	TPR|recall| Sensitivity|DR	F-measure	FPR|FAR	Dataset Name	مرجع	Accuracy	Precision	TPR|recall|Sensitivity|DR	F-measure	FPR|FAR
NSL-KDD	63	99.98		97.29		0.003527	CICIDS2017	49	97.96	98.46	98.98	98.97	0.451
	67	99.79	99.83	99.81	99.82	0.23		57	96.8	97.4	96.7		
	48	99.66	99.59	99.59	99.59	0.464		66	95.16		85.32		0.35
	62	99.44		99.36		0.6		34	99.9	99.9	99.9	99.9	0.001
	41	99.4	98.38	98.02	98.14			47	98.51	91.77	0.97	93.9	
	75	99.22		99.1		0.0064		55					
	58	99.16		99.38		0.015		72			95.91		
	66	98.99		88.82		0.3	KDD CUP 1999	46	99.82		99.7	97.23	2.4
	54	98.95	96	99.4	97.7			34	99.8	99.8	99.8	99.8	0.001
	61	98.51		98.56	99.07	0.625		68	99.6	88.5	96.2	92.2	0.4
	47	98.34	98.51	99.93	99.87			58	98.05		99.59		0.029
	40	97.8		98.76				70	96		98.2		0.076
	42	97.5	94.5	94.8	92.1			35	95.03		95.23		1.65
	46	96.93		97.5	91.9	0.112		33	90.28			77	0.005
	43	90.73						45					
	70	88.3		86.6	88.2	0.088		50					
	76	87.59						71			83.49		
	65	87.53	87.68	67.2	75.74	0.05	BOT_IOT	51	99.99	99.99	99.99	99.99	
	60	87.34	89.09	87.34	88.21			64	98.9				
	44	84.8649	88.12	60.17	60.93			46	97.37		99.7	96.7	
	73	80						37		93.7	97.7	95.6	
	69		100				CSE-CIC-IDS 2018	38	99.89	99.914	99.125	99.214	0.013
	50							47	99.95	0.9982	0.9852	0.98516	
	45							60	99.87	99.88	99.87	99.88	
	37		90.8	94.8	92.7		CIRA-CIC-DoH	47	99.43	99.94	99.94	99.915	
UNSW-NB15	63	99.86		98.51		1.021394		64	98.94				
	39	99.79	98.99	98.49	99.88		ISCX2012	58	99.14		98.98		0.004
	53	99.72	99.27	99.84	99.56	0.16		74	99.66	99.8	99.57	99.69	
	41	99.2	83.57	83.29	81.66		ADFA-LD	72	99.5	99.5	99.5	99.5	0.001
	58	97.63		98.18		0.033	AWID	34			94.44		
	38	97.535	97.674	98.945	97.995	0.105	BGFD1, BGFD2, BGFD3	50					
	59	96.5	98.6	100	99.3		CEC2021	44					
	64	96.48					CICDDOS 2019	47	97.19	84.24	96.8	96.979	
	46	94.45		96.7	90.7	11.1	CIDD	45					
	47	93.32	92.31	100	100		CSIC-2010	49	97.56	97.458	98.789	98.369	50.254
	70	91.7		89.4	90.9	0.034	Intrusion detection 2018	59	98	96.1	100	98	
	36	83.12				3.7	LUFlow Network	47	99%	0.9806	0.9818	0.9812	
	56	71.5429		80.5779			N-BaIoT	47	99%	0.9959	0.9939	0.994	

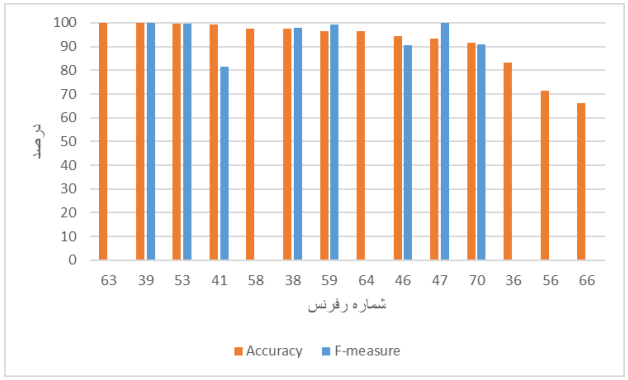
	66	66		64.9		3.85	Malmem2022	47	100%	1	0.937	0.92893	
CICIDS2017	67	99.91	99.99	99.92	99.95	0.1	Phishing Legitimate	47	96%	0.968	0.9958	0.9959	
	59	99.3	100	98.5	99.2		SWaT	52	99.65	97.16	96.78	96.97	
	58	99.23		99.26		0.013	WSN-DS	52	99%	92.87	96.44	94.62	
	42	99.1	97.2	94.1	94.5		WUSTL-IIOT	44		98.72	96.36	97.27	
	56	98.71		96.17									

در مجموعه داده NSL-KDD بیشترین دقت تشخیص ٪۹۹,۹۸ [۶۳] است که از الگوریتم کلاغ برای انتخاب ویژگی استفاده‌کرده‌است و کمترین مقدار دقت تشخیص ٪۸۰ [۷۳] است که الگوریتم نهنگ اصلاح‌شده را به‌عنوان انتخاب ویژگی به‌کاربرده‌اند. در حالی که [۶۹] با ترکیب دو الگوریتم زنبور و ازدحام ذرات تطبیقی توانسته‌است مقدار Precision را به ٪۱۰۰ برساند به‌عبارتی، روش پیشنهادی آن‌ها همه حملات را به‌درستی تشخیص‌می‌دهد. مقدار recall در بین مقالاتی که روی مجموعه داده NSL-KDD آزمایش انجام‌داده‌اند بین ٪۹۹,۹۳ [۴۷] و ٪۶۰,۱۷ [۴۴] است. بنابراین استفاده از الگوریتم ازدحام salp مبتنی بر مخالفت توانسته بهترین مقدار recall را داشته‌باشد یعنی تا حد زیادی در شناسایی تمامی موارد مثبت، موفق بوده‌است. همچنین مقدار معیار F-measure برای این روش برابر با ٪۹۹,۸۷ است [۴۷].

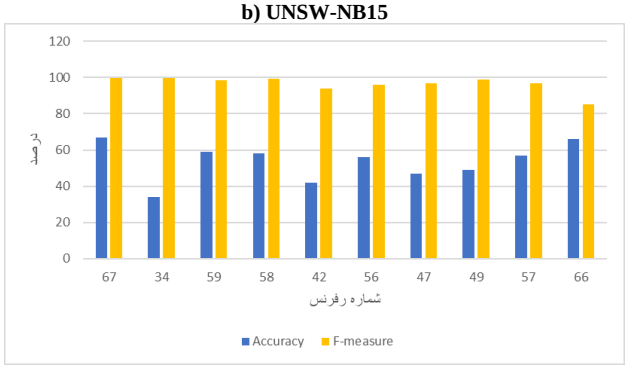
کمترین مقدار F-measure متعلق به Hybrid Breeding Optimization (HBO) است که برابر با ٪۶۰,۹۳ [۴۴] است. تعداد مقالاتی که FPR را بررسی‌کردند تقریباً کم بوده‌است. بهترین نتیجه این معیار برای الگوریتم کلاغ [۶۳] با مقدار ۰,۰۰۴ است و بدترین نتیجه را الگوریتم شیر مورچه [۶۱] با مقدار ۰,۶۲۵ داشته‌است. شکل ۶ نمودار درصد دقت تشخیص و F-measure را برای مجموعه داده‌هایی که تعداد فراوانی آن‌ها بالاتر از ۴ است، نشان‌می‌دهد. چنانچه مرجعی در این شکل نباشد به این دلیل است که درصد مشخصی از معیارها در مقاله مربوطه بیان نشده‌است و یا مقدار آن را به‌صورت نمودار نشان‌داده‌است که به‌دلیل عدم وجود مقدار دقیق از آن صرف‌نظر کردیم.



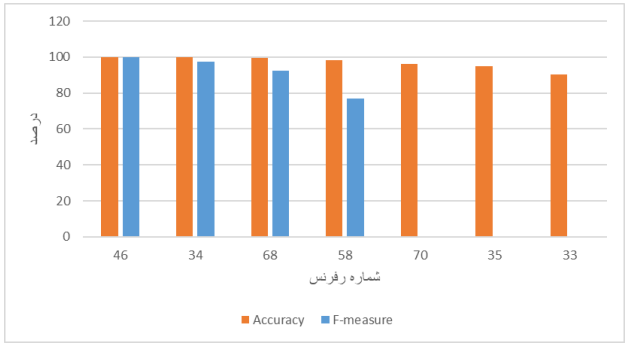
a) NSL-KDD



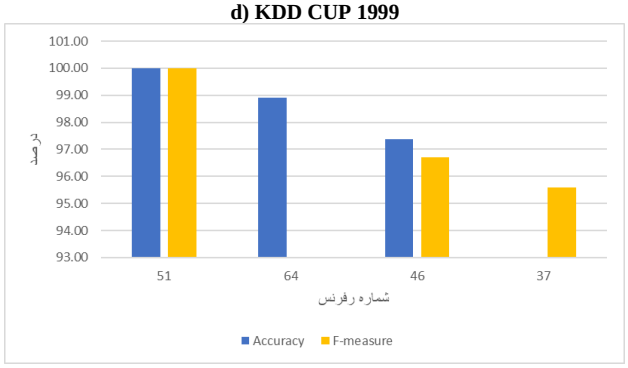
b) UNSW-NB15



c) CICIDS2017



d) KDD CUP 1999



e) BOT_IOT

شکل ۶:درصد دقت تشخیص و امتیاز F را برای مجموعه داده‌های a) NSL-KDD, b) UNSW-NB15, c) CICIDS2017, d) KDD CUP 1999, e) BOT_IOT

در مجموعه داده UNSW-NB15 بیشترین دقت تشخیص ۹۹٫۸۶٪ [۶۳] است که از الگوریتم کلاغ برای انتخاب ویژگی استفاده-کرده‌است و کمترین مقدار دقت تشخیص ۶۶٪ [۶۶] است که الگوریتم ژنتیک چندهدفه را به‌کاربرده‌اند. مقدار بیشترین Precision متعلق به الگوریتم کرم شب‌تاب باینری [۵۳] است و کمترین مقدار آن ۸۳٫۵۷٪ است که برای الگوریتم کلاغ [۴۱] است. درحالی‌که [۵۹] با الگوریتم ژنتیک اصلاح‌شده توانسته‌است مقدار recall را به ۱۰۰٪ برساند و الگوریتم ژنتیک چندهدفه [۶۶] کمترین مقدار recall را دارد (۶۶٪). و [۴۷] با الگوریتم ازدحام سالپ مبتنی بر مخالفت توانسته مقدار recall و F-measure را به ۱۰۰٪ برساند و [۴۱] کمترین مقدار F-measure را در مجموعه داده UNSW-NB15 دارد. بهترین نتیجه معیار FPR برای الگوریتم خفاش چندهدفه [۵۸] با مقدار ۰٫۰۳۳ و بدترین نتیجه برای الگوریتم کبوتر [۴۶] با ۱۱٫۱ است.

دقت در مجموعه داده CICIDS2017 به‌طورکلی بالا است رنج درصد دقت تشخیص در این مجموعه داده از ۹۵٫۱۶٪ تا ۹۹٫۹۱٪ است. بیشترین دقت مربوط به الگوریتم ازدحام ذرات [۶۷] با ۹۹٫۹۱٪ است البته این روش بیشترین مقدار recall را با ۹۹٫۹۲٪ از بین همه متدهای این قسمت داراست و مقدار Precision آن برابر با ۹۹٫۹۹٪ است. همچنین مقدار Precision در روش‌های مختلف برای CICIDS2017 بین ۹۱٫۷۷٪ تا ۱۰۰٪ است که [۵۹] با الگوریتم ژنتیک اصلاح‌شده بیشترین مقدار Precision را دارد. مقدار F-measure از ۹۳٫۹۰٪ تا ۹۹٫۵٪ است که درصد خوبی از تشخیص حملات و ترافیک نرمال را نشان‌می‌دهد. بهترین نتیجه معیار FPR برای الگوریتم خفاش [۳۴] با مقدار ۰٫۰۰۱ و پایین‌ترین نتیجه برای الگوریتم MMFO (modified manta-ray foraging optimization) [۴۹] با ۰٫۴۵۱ است.

دقت تشخیص نفوذ برای مجموعه داده KDD CUP 1999 از ۹۰٫۲۸٪ تا ۹۹٫۸۲٪ است. الگوریتم کبوتر [۴۶] بیشترین و الگوریتم کرم شب‌تاب [۳۳] کمترین مقدار دقت را دارد. مقدار معیار Precision برای الگوریتم خفاش [۳۴]، ۹۹٫۸٪ و برای ازدحام ذرات [۶۸] ۸۸٫۵٪ است. مقدار معیارهای recall و F-measure در الگوریتم خفاش [۳۴]، برابر با ۹۹٫۸٪ است که بیشترین مقدار این معیارها در مجموعه داده KDD CUP 1999 می‌باشد. کمترین مقدار معیار recall در این مجموعه داده ۸۳٫۴۹٪ [۷۱] است. الگوریتم کرم شب‌تاب [۳۳] کمترین مقدار معیار F-measure را دارد.

الگوریتم ژنتیک [۵۱] روی مجموعه داده BOT_IOT دارای بیشترین مقدار برای معیارهای دقت، صحت، فراخوانی و F-measure است و البته متدهای دیگر [۶۴، ۴۶ ، ۳۷] نیز درصدهای بالای ۹۳٪ را نشان‌می‌دهند. همچنین [۴۷، ۶۰ ، ۳۸] درصدهای بالای ۹۸٪ را برای معیارهای مختلف در مجموعه CSE-CIC-IDS 2018 اعلام‌کرده‌اند.

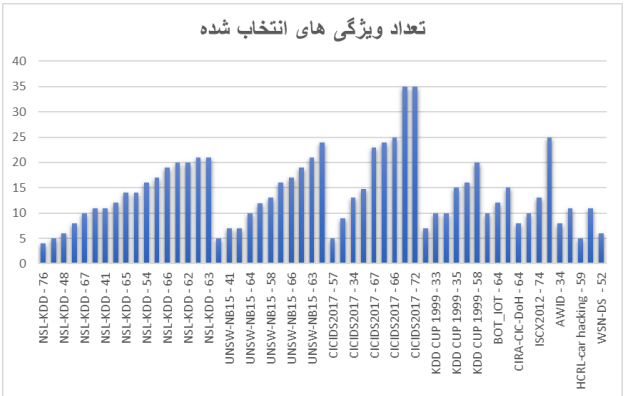
۵،۳. تعداد ویژگی‌های انتخاب‌شده

یکی از پارامترهایی که در انتخاب ویژگی اهمیت دارد تعداد ویژگی‌های انتخاب شده است. این تعداد به مجموعه‌ای از ویژگی‌ها اشاره دارد که از بین ویژگی‌های اصلی مجموعه داده انتخاب‌شده‌اند تا مدل یادگیری ماشین با استفاده از آن‌ها ساخته‌شود. به‌طورکلی، هیچ عدد خاصی برای تعداد بهینه ویژگی‌ها وجود ندارد. بهترین راه برای انتخاب تعداد ویژگی-ها، آزمایش با مقادیر مختلف و ارزیابی عملکرد مدل بر روی داده ها است. تعداد ویژگی‌های انتخاب‌شده بسته به روش انتخاب ویژگی می‌تواند متفاوت باشد. جدول ۷ و شکل ۷ تعداد ویژگی‌های انتخاب شده را بر اساس مجموعه داده‌ها در روش‌های مختلف ارائه‌می‌دهند.

جدول۷:تعداد ویژگی‌های انتخابی بر اساس مجموعه داده‌ها

تعداد ویژگی‌های انتخاب شده	مرجع	Dataset Name
11	40	NSL-KDD
11	41	
21	44	
17	45	
6	48	
16	54	
12	58	
20	60	
20	62	
21	63	
14	65	
19	66	
10	67	
5	70	
8	73	
14	75	
4	76	
16	36	UNSW-NB15
24	39	
7	41	
19	53	
11.9	56	
13	58	
7	59	
21	63	
10	64	
17	66	
5	70	
13	34	CICIDS2017

	55	24
	56	15
	57	5
	58	35
	59	9
	66	25
	67	23
	72	35
KDD CUP 1999	33	10
	34	10
	35	15
	45	16
	58	20
	70	7
BOT_IOT	51	10
	64	12
CSE-CIC-IDS 2018	60	15
CIRA-CIC-DoH	64	8
ISCX2012	58	10
	74	13
ADFA-LD	72	25
AWID	34	8
CIDD	45	11
HCRL-car hacking	59	5
SWaT	52	11
WSN-DS	52	6



شکل ۷ تعداد ویژگی‌های انتخابی را بر اساس مجموعه داده-ها در روش‌های مختلف

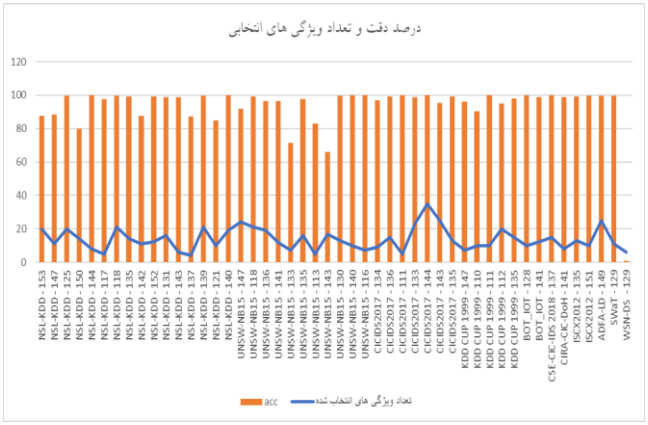
الگوریتم گرگ خاکستری چندهدفه [۷۶] توانسته‌است ۴ ویژگی از مجموعه داده NSL-KDD انتخاب کند

$$F_{i=}\frac{A_i+(1-M_i)}{2} \quad (7)$$

که کمترین تعداد ویژگی بین روش‌های بررسی‌شده است. بعد از آن الگوریتم کبوتر [۷۰] با ۵ ویژگی انتخابی است. سپس الگوریتم تکاملی تفاضلی [۴۸] با ۶ ویژگی و الگوریتم بهینه‌سازی نهنگ اصلاح‌شده [۷۳] با ۸ ویژگی انتخابی، هستند.

برای مجموعه داده UNSW-NB15، الگوریتم‌های کبوتر [۷۰]، الگوریتم‌های ژنتیک تغییریافته (MGA) و کلاغ [۴۱ ، ۵۹] به ترتیب ۵ و ۷ ویژگی را انتخاب‌کردند. Multi-Objective Evolutionary Feature Selection (MOEFS) [۵۷] و ژنتیک تغییر یافته [۵۹] به ترتیب ۵ و ۹ ویژگی از مجموعه داده CICIDS2017 را انتخاب‌کرده‌اند.

در مجموعه داده KDD CUP 1999 ابتدا الگوریتم کبوتر [۷۰] با ۷ ویژگی، سپس الگوریتم‌های کرم شب‌تاب و خفاش [۳۳ ، ۳۴] کمترین تعداد ویژگی را انتخاب‌کرده‌اند. بنابراین الگوریتم کبوتر [۷۰] برای سه مجموعه داده NSL-KDD، UNSW-NB15 و KDD CUP 1999 کمترین تعداد ویژگی‌ها را انتخاب‌کرده‌است. برای مقایسه بیشتر شکل ۸ تعداد ویژگی‌های انتخابی و دقت تشخیص را نشان‌می‌دهد. همان‌طور که مشخص است در مجموعه داده NSL-KDD الگوریتم NSGA-II [۶۶] هم تعداد ویژگی انتخابی کمتر و هم دقت تشخیص بالایی دارد.



شکل ۸: تعداد ویژگی‌های انتخابی و دقت تشخیص

۴،۵. توابع هدف

توابع هدف در الگوریتم‌های فراابتکاری نقش حیاتی در هدایت فرآیند بهینه‌سازی ایفامی‌کنند. این توابع، معیاری برای ارزیابی کیفیت هر راه‌حل پیشنهادی در فضای جستجو فراهم‌می‌کنند. انتخاب و طراحی مناسب تابع هدف می‌تواند تأثیر زیادی بر عملکرد و کارایی الگوریتم فراابتکاری داشته‌باشد.

معیار دقت نقش مهمی در تابع هدف مسأله تشخیص نفوذ دارد و مقالات زیادی [۳۳، ۳۸، ۴۰، ۴۲، ۴۳، ۴۹، ۵۱، ۵۳، ۵۴، ۵۵، ۵۷، ۵۹، ۶۱، ۶۵، ۶۸، ۷۰، ۷۱، ۷۲، ۷۳، ۷۶] از آن به‌تنهایی برای تابع هدف استفاده‌کرده‌اند. و تعدادی معیار دقت را با معیارهای دیگر ترکیب کرده-اند. مانند [۷۵،۴۱،۶۶،۷۴] که دقت و تعداد ویژگی انتخابی را ترکیب-کردند. [۶۴] از رابطه ۷ برای ترکیب دقت و ماتریس همبستگی بین ویژگی‌ها به‌عنوان تابع هدف استفاده‌کردند.

که Ai دقت و Mi , ماتریس همبستگی است. تابع هدف در [۶۲] ترکیبی از دقت و نرخ تشخیص است.

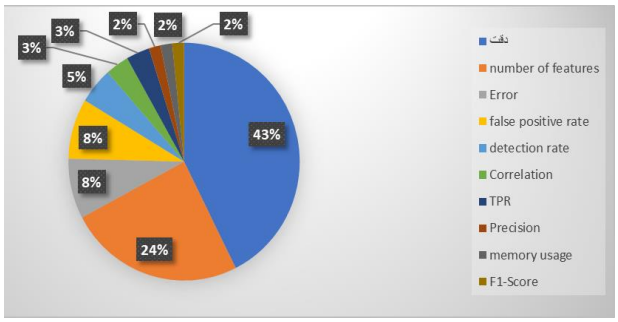
جدول ۸ خلاصه‌ای از پارامترهای توابع هدف را نشان‌می‌دهد. دقت بیشترین کاربرد را در تابع هدف تحقیقات گذشته دارد پس از آن تعداد ویژگی‌های انتخابی، میزان خطا و false positive rate هستند.

که Ai دقت و Mi , ماتریس همبستگی است. تابع هدف در [۶۲] ترکیبی از دقت و نرخ تشخیص است. جدول ۸ خلاصه‌ای از پارامترهای توابع هدف را نشان‌می‌دهد. دقت بیشترین کاربرد را در تابع هدف تحقیقات گذشته دارد، پس از آن تعداد ویژگی‌های انتخاب‌شده، میزان خطا و false positive rate هستند.

جدول ۸: توابع هدف مقالات بررسی‌شده

پارامترها	مراجع
دقت	[۳۳، ۳۸، ۴۰، ۴۲، ۴۳، ۴۵،۴۸، ۴۹، ۵۱، ۵۳، ۵۴، ۵۵، ۵۷، ۵۹، ۶۱،۶۴، ۶۵، ۶۸،۶۶، ۷۰، ۷۱، ۷۲، ۷۳،۷۴،۷۵، ۷۶]
Precision	39
Error	36, 47, 50, 56, 58
Correlation	34, 64
detection rate	35,62, 75
false positive rate	35,36,44, 46, 58
number of features	36,37,41,44,46,47, 48, 56, 58, 60, 63, 66, 67, 74,75
TPR	44, 46,
memory usage	50
F1-Score	52

شکل ۹ نشان‌می‌دهد که ۴۳٪ از مطالعات از پارامتر دقت برای تابع هدف (به صورت تک‌هدفه یا چندهدفه) استفاده‌کردند که ۷۷٪ آن فقط پارامتر دقت را در نظرگرفته‌اند و مابقی مطالعات ، ترکیب پارامتر دقت و دیگر پارامترها را به‌کار برده‌اند. دومین پارامتر پر استفاده، number of features است که ۲۴٪ از مطالعات به آن پرداخته‌اند. false positive rate و میزان خطا به‌طور مساوی ۸٪ و پارامتر detection rate ۵٪ مطالعات و پارامترهای Correlation و TPR به‌طور یکسان ۳٪ از توابع هدف مطالعات را به خود اختصاص‌داده‌اند. دیگر پارامترها هر کدام ۲٪ در توابع هدف مشارکت‌داشته‌اند.



شکل۹: درصد استفاده از پارامترهای مختلف در توابع هدف مطالعات انجام‌شده

۶. چالش‌ها و شکاف‌های تحقیقاتی انتخاب ویژگی مبتنی بر الگوریتم‌های فراابتکاری

در این بخش به سوال سوم «چالش‌ها و شکاف‌های تحقیقاتی در الگوریتم‌های فراابتکاری موجود برای انتخاب ویژگی در سیستم‌های تشخیص نفوذ چیست؟» پاسخ داده‌می‌شود.

انتخاب ویژگی یکی از مراحل حیاتی در پیش‌پردازش داده‌ها و ساخت مدل‌های یادگیری ماشین است. این فرآیند با هدف کاهش تعداد ویژگی‌ها، حذف ویژگی‌های زائد و بهبود دقت مدل انجام‌می‌شود. الگوریتم‌های فراابتکاری به‌دلیل توانایی‌شان در جستجوی فضای بزرگ و پیچیده ویژگی‌ها، به‌طور گسترده‌ای در انتخاب ویژگی استفاده می‌شوند. با این حال، این رویکردها با چالش‌ها و شکاف‌های تحقیقاتی متعددی مواجه هستند که در ادامه به برخی از آن‌ها اشاره می‌کنم:

پیچیدگی محاسباتی و زمان اجرا: الگوریتم‌های فراابتکاری معمولاً به زمان محاسباتی زیادی نیاز دارند، به‌خصوص زمانی که داده‌ها دارای تعداد زیادی ویژگی باشند. این مسئله می‌تواند کارایی را کاهش-دهد و در کاربردهای عملی محدودیت ایجادکند. با توجه به بررسی‌های انجام‌شده، تعداد مقالات کمی به زمان ساخت مدل، زمان اجرا، زمان آموزش و تست توجه‌کردند و این پارامترها را جهت ارزیابی نتایج درنظر نگرفته‌اند.

بهینه محلی: الگوریتم‌های فراابتکاری ممکن است در بهینه‌های محلی گیر کنند و نتوانند به بهینه جهانی دست‌یابند. این مشکل به‌ویژه در مسائل پیچیده و با فضای جستجوی بزرگ مشهود است. اکثر مقالات سعی‌داشتند جستجوی محلی را بهبوددهند و تعداد کمی از الگوریتم‌های پایه و استاندارد استفاده‌کردند. البته بهینه‌کردن جستجوی محلی الگوریتم‌های فراابتکاری همچنان موضوع باز تحقیقات امروزی است.

تعیین پارامترها: الگوریتم‌های فراابتکاری معمولاً دارای پارامترهای متعددی هستند که تنظیم دقیق آن‌ها برای دستیابی به عملکرد بهینه ضروری است. تنظیم نادرست پارامترها می‌تواند منجر به عملکرد ضعیف الگوریتم شود. تعداد کمی از تحقیقات مذکور، پارامترهای الگوریتم فراابتکاری خود را با روش‌های مختلف به‌دقت تنظیم‌کرده‌اند.

تعامد و همبستگی ویژگی‌ها: انتخاب ویژگی‌هایی که به شدت با هم همبسته هستند یا برعکس، تعاملات پیچیده‌ای با هم دارند، می‌تواند عملکرد مدل را تحت تأثیر قراردهد. الگوریتم‌های فراابتکاری به‌طور معمول نمی‌توانند این تعاملات پیچیده را به‌خوبی مدیریت کنند. بنابراین می‌توان از پارامتر همبستگی ویژگی‌ها در تابع هدف الگوریتم فراابتکاری مانند [۳۴ ، ۶۴] استفاده‌کرد.

ناتوانی در تضمین بهترین راه‌حل: الگوریتم‌های فراابتکاری تضمینی برای یافتن بهترین راه‌حل ندارند و تنها می‌توانند به یک راه‌حل تقریباً بهینه دست‌یابند. این مسئله می‌تواند در مسائل حساس به‌دقت مانند تشخیص نفوذ مشکل‌ساز شود. بنابراین بررسی تعداد معیارهای ارزیابی همزمان می‌تواند ما را به راه حل بهینه نزدیک‌تر کند.

پیچیدگی تابع هدف: تابع هدف باید به‌گونه‌ای طراحی‌شود که الگوریتم بتواند از بهینه‌های محلی فرار کرده و به سمت بهینه جهانی حرکت‌کند. از طرفی تابع هدف نباید بیش از حد پیچیده باشد، زیرا محاسبه مکرر آن می‌تواند زمان‌بر باشد. در مسائل چندهدفه، تعیین راه‌حل‌هایی که تعادل مناسبی بین اهداف مختلف برقرار کنند، یک چالش است

.

شکاف‌های تحقیقاتی:

توسعه الگوریتم‌های ترکیبی: یکی از زمینه‌های تحقیقاتی جذاب، ترکیب الگوریتم‌های فراابتکاری با دیگر تکنیک‌ها به‌منظور بهره‌برداری از مزایای هر دو روش است. توسعه الگوریتم‌های ترکیبی می‌تواند به بهبود کارایی و دقت انتخاب ویژگی کمک‌کند.

روش‌های تطبیقی و خودتنظیمی: تحقیقات بیشتری در زمینه توسعه روش‌های تطبیقی و خودتنظیمی برای تنظیم پارامترهای الگوریتم‌های فراابتکاری مورد نیاز است. این روش‌ها می‌توانند به صورت خودکار بهترین پارامترها را برای هر مسئله خاص تنظیم‌کنند.

الگوریتم‌های مقاوم در برابر بهینه‌های محلی: نیاز به توسعه الگوریتم‌هایی است که بتوانند از بهینه‌های محلی فرارکنند و به بهینه‌های جهانی نزدیک‌تر شوند. استفاده از تکنیک‌هایی مانند حافظه و مکانیزم‌های جستجوی چندگانه می‌تواند در این زمینه مؤثر باشد.

مدیریت تعاملات پیچیده ویژگی‌ها: الگوریتم‌های جدیدی که بتوانند تعاملات پیچیده بین ویژگی‌ها را به‌خوبی مدیریت‌کنند و از اطلاعات تعاملی برای انتخاب ویژگی‌ها بهره‌ببرند، یک شکاف تحقیقاتی مهم است.

7. نتیجه‌گیری و راهکارهای آینده

سیستم‌های تشخیص نفوذ موجود، می‌تواند به‌طور مؤثر رفتارهای ناهنجاری را در شبکه شناسایی‌کند. با این حال، هنوز هم دارای نرخ تشخیص پایین و نرخ هشدار کاذب بالا به‌خصوص برای حملاتی با رکوردهای کمتر است و البته شکاف‌های تحقیقاتی وجود دارد که بررسی‌نکردیم. ما فقط ۴۳ مقاله انتخاب ویژگی مبتنی بر الگوریتم‌های فراابتکاری مختلف در تشخیص نفوذ را بررسی‌کردیم. الگوریتم‌های فراابتکاری مبتنی بر هوش جمعی پراستفاده‌ترین نوع الگوریتم فراابتکاری است. ۷۶٪ از مطالعات از الگوریتم‌های هوش جمعی استفاده‌کردند.

تقریباً ۲۸٪ مطالعات به مجموعه داده KDD99، که حدود ۲۰ سال پیش ایجادشده‌است، وابسته هستند. این مجموعه داده به‌طور گسترده برای ارزیابی الگوریتم‌های یادگیری ماشین، به‌ویژه در تشخیص ترافیک مخرب از ترافیک عادی، استفاده‌شده‌است. با این حال، KDD99 دیگر نمی‌تواند ویژگی‌های ترافیک داده‌های مدرن را به‌خوبی منعکس‌کند. در حال حاضر، ترافیک چندرسانه‌ای مانند صدا و ویدیو اصلی‌ترین منبع درآمد ارائه‌دهندگان خدمات است و همچنین وسیله‌ای برای پوشش ترافیک مخرب محسوب‌می‌شود. این نوع ترافیک در مجموعه داده KDD99 گنجانده‌نشده‌است [۱۰].

تقریباً ۲۴٪ از مطالعات معیار دقت را برای ارزیابی نتایج متدشان انتخاب‌کردند و ۲۳٪ از آن‌ها TPR|recall|Sensitivity|DR را بررسی کرده‌اند. درحالی‌که معیارهای F1-score | F-measure، FPR|FAR و Precision فقط ۱۳٪ بررسی نتایج مطالعات را به خود اختصاص داده اند. در صورتی که برای بررسی کامل نتایج باید این معیارها نیز تجزیه و تحلیل شوند. همچنین زمان اجرای الگوریتم در ۴٪ مطالعات آمده‌است که بسیار کم می‌باشد درحالی‌که پیچیدگی زمان اجرای الگوریتم یک چالش مهم برای مسأله تشخیص نفوذ است.

معیار دقت برای توابع هدف الگوریتم‌های مختلف، زیاد به‌کار برده شده‌است به طوری که ۴۳٪ از توابع هدف مطالعات شامل دقت بوده است که ۳۳٪ فقط معیار دقت را برای انتخاب ویژگی انتخاب‌کردند و ۱۰٪ از ترکیب دقت با دیگر پارامترها تشکیل‌شده‌است. البته بهتر است پارامترهای مهمی مانند RECALL, PRECISION, FPR و F1-Score نیز در تابع هدف مشارکت داشته‌باشند. توابع هدف در الگوریتم‌های فراابتکاری نقش کلیدی در هدایت فرآیند بهینه‌سازی دارند. طراحی مناسب و انتخاب صحیح تابع هدف می‌تواند تأثیر زیادی بر عملکرد و کارایی الگوریتم داشته باشد. با توجه به نوع مسئله و ویژگی‌های فضای جستجو، می‌توان تابع هدف مناسبی را انتخاب و بهینه‌سازی را به نحو مؤثری انجام‌داد .

برای کارهای آینده بررسی الگوریتم‌های طبقه‌بندی جهت تشخیص نفوذ نیاز است. مطالعه الگوریتم‌های یادگیری ماشین و طبقه‌بندی آن‌ها در سیستم‌های تشخیص نفوذ مؤثر خواهدبود. برای ادامه کار مجموعه داده‌هایی که شامل ترافیک چندرسانه‌ای و انواع جدید حملات مانند حملات مبتنی بر هوش مصنوعی هستند را بررسی خواهیم‌کرد. همچنین بررسی و ارزیابی الگوریتم‌های فراابتکاری جدید و نوآورانه می‌توانند به بهبود انتخاب ویژگی کمک‌کنند. بررسی تکنیک‌های ترکیبی روش‌های فیلتر و لفاف انتخاب ویژگی نیز می‌تواند مؤثر باشد.

انجام تحلیل‌های مقایسه‌ای بین روش‌های مختلف انتخاب ویژگی و تشخیص نفوذ در محیط‌های ابری مختلف و بررسی عملکرد آن‌ها در

شرایط مختلف و با داده‌های مختلف نیز جهت کار آینده پیشنهاد می‌شود. البته تحقیق در مورد روش‌های بهینه‌سازی پارامترها برای الگوریتم‌های انتخاب ویژگی و تشخیص نفوذ، به‌ویژه در محیط‌های ابری با مقیاس بزرگ مورد نیاز است.

منابع

1. Thakkar, A., & Lohiya, R. (2022). A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. Artificial Intelligence Review, 55(1), 453-563. https://doi.org/10.1007/S10462-021-10037-9.
2. Ghaffari, A., & Hossinnezhad, R. (2022).Intrusions detection system in the cloud computing using heterogeneity detection technique.Intelligent Multimedia Processing and Communication Systems (IMPCS),3(1),37-46. https://srb.sanad.iau.ir/en/Article/903516
3. Roknaldini, M., & Noroozi, E. (1402). Presenting A Hybrid Method of Deep Neural Networks to Prevent Intrusion in Computer Networks. Intelligent Multimedia Processing and Communication Systems (IMPCS), 4(4), 57-65. http://sanad.iau.ir/fa/Article/903465
4. Kaya, İ. (2009). RETRACTED: A genetic algorithm approach to determine the sample size for control charts with variables and attributes. https://doi.org/10.1016/j.eswa.2008.12.011
5. Mohammad Akhlaghpour. (2023,apr). Using the Modified Colonial Competition Algorithm to Increase the Speed and Accuracy of the Intelligent Intrusion Detection System. (IMPCS), (pp. 1-10). https://doi.org/10.1016/j.cie.2019.106040
6. Khanduja, N., & Bhushan, B. (2021). Recent advances and application of metaheuristic algorithms: A survey (2014–2020). Metaheuristic and evolutionary computation: algorithms and applications, 207-228. https://doi.org/10.1007/978-981-15-7571-6_10.
7. Dokeroglu, T., Deniz, A., & Kiziloz, H. E. (2022). A comprehensive survey on recent metaheuristics for feature selection. Neurocomputing, 494, 269-296. https://doi.org/10.1016/j.neucom.2022.04.083.
8. Maldonado, J., Riff, M. C., & Neveu, B. (2022). A review of recent approaches on wrapper feature selection for intrusion detection. Expert Systems with Applications, 198, 116822. https://doi.org/10.1016/j.eswa.2022.116822.
9. Sharma, S., Kumar, V., & Dutta, K. (2024). Multi-objective optimization algorithms for intrusion detection in IoT networks: A systematic review. Internet of Things and Cyber-Physical Systems. https://doi.org/10.1016/j.iotcps.2024.01.003.
10. Reddy, D. K. K., Nayak, J., Behera, H. S., Shanmuganathan, V., Viriyasitavat, W., & Dhiman, G. (2024). A Systematic Literature Review on Swarm Intelligence Based Intrusion Detection System: Past, Present and Future. Archives of Computational Methods in Engineering, 1-68. https://doi.org/10.1007/s11831-023-10059-2.
11. Balasaraswathi, V. R., Sugumaran, M., & Hamid, Y. (2017). Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms. Journal of Communications and Information Networks, 2, 107-119. https://doi.org/10.1007/s41650-017-0033-7.
12. Saadouni, R., Gherbi, C., Aliouat, Z., Harbi, Y., & Khacha, A. (2024). Intrusion detection systems for IoT based on bio-inspired and machine learning techniques: a systematic review of the literature. Cluster Computing, 1-27. https://doi.org/10.1007/s10586-024-04388-5.
13. Kalimuthan, C., & Renjit, J. A. (2020). Review on intrusion detection using feature selection with machine learning techniques. Materials Today: Proceedings, 33, 3794-3802. https://doi.org/10.1016/j.matpr.2020.06.218.
14. Agrawal, P., Abutarboush, H. F., Ganesh, T., & Mohamed, A. W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). Ieee Access, 9, 26766-26791. https://doi.org/10.1109/ACCESS.2021.3056407.
15. Nssibi, M., Manita, G., & Korbaa, O. (2023). Advances in nature-inspired metaheuristic optimization for feature selection problem: A comprehensive survey. Computer Science Review, 49, 100559. https://doi.org/10.1016/j.cosrev.2023.100559.
16. Pham, T. H., & Raahemi, B. (2023). Bio-inspired feature selection algorithms with their applications: a systematic literature review. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3272556.
17. Bello, S. A., Oyedele, L. O., Akinade, O. O., Bilal, M., Delgado, J. M. D., Akanbi, L. A., ... & Owolabi, H. A. (2021). Cloud computing in construction industry: Use cases, benefits and challenges. Automation in Construction, 122, 103441. https://doi.org/10.1016/j.autcon.2020.103441.
18. Lansky, J., Ali, S., Mohammadi, M., Majeed, M. K., Karim, S. H. T., Rashidi, S., ... & Rahmani, A. M. (2021). Deep learning-based intrusion detection systems: a systematic review. IEEE Access, 9, 101574-101599. https://doi.org/10.1109/ACCESS.2021.3097247
19. Melvin, A. A. R., Kathrine, G. J. W., Ilango, S. S., Vimal, S., Rho, S., Xiong, N. N., & Nam, Y. (2022). Dynamic malware attack dataset leveraging virtual machine monitor audit data for the detection of intrusions in cloud. Transactions on Emerging Telecommunications Technologies, 33(4), e4287. https://doi.org/10.1002/ett.4287.
20. Rostami, M., Berahmand, K., Nasiri, E., & Forouzandeh, S. (2021). Review of swarm intelligence-based feature selection methods. Engineering Applications of Artificial Intelligence, 100, 104210. https://doi.org/10.1016/j.engappai.2021.104210.
21. Solorio-Fernández, S., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2020). A review of unsupervised feature selection methods. Artificial Intelligence Review, 53(2), 907-948. .https://doi.org/10.1007/s10462-019-09682-y
22. Abdel-Basset, M., Abdel-Fatah, L., & Sangaiah, A. K. (2018). Metaheuristic algorithms: A comprehensive *review*. *Computational intelligence for multimedia* big data on the cloud with engineering applications, 185-231. https://doi.org/10.1016/B978-0-12-813314-9.00010-4.
23. Osaba, E., Villar-Rodriguez, E., Del Ser, J., Nebro, A. J., Molina, D., LaTorre, A., ... & Herrera, F. (2021). A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization

problems. Swarm and Evolutionary Computation, 64, 100888. https://doi.org/10.1016/j.swevo.2021.100888.
24. Dehghani, M., Montazeri, Z., Trojovská, E., & Trojovský, P. (2023). Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. Knowledge-Based Systems, 259, 110011. https://doi.org/10.1016/j.knosys.2022.110011.
25. Priyadarshini, J., Premalatha, M., Čep, R., Jayasudha, M., & Kalita, K. (2023). Analyzing physics-inspired metaheuristic algorithms in feature selection with K-nearest-neighbor. Applied Sciences, 13(2), 906. https://doi.org/10.3390/app13020906.
26. Jia, H., & Lu, C. (2024). Guided learning strategy: A novel update mechanism for metaheuristic algorithms design and improvement. Knowledge-Based Systems, 286, 111402. https://doi.org/10.1016/j.knosys.2024.111402.
27. Kwakye, B. D., Li, Y., Mohamed, H. H., Baidoo, E., & Asenso, T. Q. (2024). Particle guided metaheuristic algorithm for global optimization and feature selection problems. Expert Systems with Applications, 248, 123362. https://doi.org/10.1016/j.eswa.2024.123362.
28. Di Mauro, M., Galatro, G., Fortino, G., & Liotta, A. (2021). Supervised feature selection techniques in network intrusion detection: A critical review. Engineering Applications of Artificial Intelligence, 101, 104216. https://doi.org/10.1016/j.engappai.2021.104216
29. Nazir, A., & Khan, R. A. (2021). A novel combinatorial optimization based feature selection method for network intrusion detection. Computers & Security, 102, 102164. https://doi.org/10.1016/j.cose.2020.102164.
30. Slowik, A., & Kwasnicka, H. (2020). Evolutionary algorithms and their applications to engineering problems. Neural Computing and Applications, 32, 12363-12379. https://doi.org/10.1007/s00521-020-04832-8.
31. Beni, G. (2020). Swarm intelligence. Complex Social and Behavioral Systems: Game Theory and Agent-Based Models, 791-818. https://doi.org/10.1007/978-1-0716-0368-0_530.
32. Prajapati, V. K., Jain, M., & Chouhan, L. (2020, February). Tabu search algorithm (TSA): A comprehensive survey. In 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE) (pp. 1-8). IEEE. https://doi.org/10.1109/ICETCE48199.2020.9091743.
33. Selvakumar, B., & Muneeswaran, K. (2019). Firefly algorithm based feature selection for network intrusion detection. Computers & Security, 81, 71-155. https://doi.org/10.1016/j.cose.2018.11.005
34. Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. Computer networks, 174, 107247. https://doi.org/10.1016/j.comnet.2020.107247
35. Mohammadi, S., Mirvaziri, H., Ghazizadeh-Ahsaee, M., & Karimipour, H. (2019). Cyber intrusion detection by combined feature selection algorithm. Journal of information security and applications, 44, 80-88. https://doi.org/10.1016/j.jisa.2018.11.007
36. Nazir, A., & Khan, R. A. (2021). A novel combinatorial optimization based feature selection method for network intrusion detection. Computers & Security, 102, 102164. https://doi.org/10.1016/j.cose.2020.102164
37. Rani, B. S., Vairamuthu, S., & Subramanian, S. (2024). Archimedes Fire Hawk Optimization Enabled Feature Selection with Deep Maxout for Network Intrusion Detection. Computers & Security, 103751. https://doi.org/10.1016/j.cose.2024.103751
38. Turukmane, A. V., & Devendiran, R. (2024). M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning. Computers & Security, 137, 103587. https://doi.org/10.1016/j.cose.2023.103587
39. Aljehane, N. O., Mengash, H. A., Hassine, S. B., Alotaibi, F. A., Salama, A. S., & Abdelbagi, S. (2024). Optimizing intrusion detection using intelligent feature selection with machine learning model. Alexandria Engineering Journal, 91, 39-49. https://doi.org/10.1016/j.aej.2024.01.073
40. Abualigah, L., Ahmed, S. H., Almomani, M. H., Zitar, R. A., Alsoud, A. R., Abuhaija, B., ... & Elaziz, M. A. (2024). Modified Aquila Optimizer Feature Selection Approach and Support Vector Machine Classifier for Intrusion Detection System. Multimedia Tools and Applications, 1-27. https://doi.org/10.1007/s11042-023-17886-2
41. Jayalatchumy, D., Ramalingam, R., Balakrishnan, A., Safran, M., & Alfarhood, S. (2024). Improved Crow Search-based Feature Selection and Ensemble Learning for IoT Intrusion Detection. IEEE Access. https://doi.org/10.1109/ACCESS.2024.3372859
42. Jiang, T., Fu, X., & Wang, M. (2024). BBO-CFAT: Network intrusion detection model based on BBO algorithm and hierarchical transformer. IEEE Access. https://doi.org/10.1109/ACCESS.2024.3386405
43. Sangaiah, A. K., Javadpour, A., Ja'fari, F., Pinto, P., Zhang, W., & Balasubramanian, S. (2023). A hybrid heuristics artificial intelligence feature selection for intrusion detection classifiers in cloud of things. Cluster Computing, 26(1), 599-612. https://doi.org /10.1007/s10586-022-03629-9.
44. Ye, Z., Luo, J., Zhou, W., Wang, M., & He, Q. (2023). An ensemble framework with improved hybrid breeding optimization-based feature selection for intrusion detection. Future Generation Computer Systems. https://doi.org/10.1016/j.future.2023.09.035
45. Subramani, S., & Selvi, M. (2023). Multi-objective PSO based feature selection for intrusion detection in IoT based wireless sensor networks. Optik, 273, 170419. https://doi.org/10.1016/j.ijleo.2022.170419
46. Alghanam, O. A., Almobaideen, W., Saadeh, M., & Adwan, O. (2023). An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning. Expert Systems with Applications, 213, 118745. https://doi.org/10.1016/j.eswa.2022.118745
47. Barhoush, M., Abed-alguni, B. H., & Al-qudah, N. E. A. (2023). Improved discrete salp swarm algorithm using exploration and exploitation techniques for feature selection in intrusion detection systems. The Journal of Supercomputing, 79(18), 21265-21309. https://doi.org/10.1007/s11227-023-05444-4

using hybrid Naïve Bayes—Decision table and multi-objective evolutionary feature selection. Computer Communications, 188, 133-144. https://doi.org/10.1016/j.comcom.2022.03.009.

58. Ghanem, W. A. H., Ghaleb, S. A. A., Jantan, A., Nasser, A. B., Saleh, S. A. M., Ngah, A., ... & Abiodun, O. I. (2022). Cyber intrusion detection system based on a multiobjective binary bat algorithm for feature selection and enhanced bat algorithm for parameter optimization in neural networks. IEEE Access, 10, 76318-76339. https://doi.org/10.1109/ACCESS.2022.3192472.
59. Aksu, D., & Aydin, M. A. (2022). MGA-IDS: Optimal feature subset selection for anomaly detection framework on in-vehicle networks-CAN bus based on genetic algorithm and intrusion detection approach. Computers & Security, 118, 102717. https://doi.org/10.1016/j.cose.2022.102717
60. Zhao, R., Mu, Y., Zou, L., & Wen, X. (2022). A hybrid intrusion detection system based on feature selection and weighted stacking classifier. IEEE Access, 10, 71414-71426. https://doi.org/10.1109/ACCESS.2022.3186975.
61. Chen, J., Qi, X., Chen, L., Chen, F., & Cheng, G. (2020). Quantum-inspired ant lion optimized hybrid k-means for cluster analysis and intrusion detection. Knowledge-Based Systems, 203, 106167. https://doi.org/10.1016/j.knosys.2020.106167.
62. Kunhare, N., Tiwari, R., & Dhar, J. (2022). Intrusion detection system using hybrid classifiers with meta-heuristic algorithms for the optimization and feature selection by genetic algorithm. Computers and Electrical Engineering, 103, 108383. https://doi.org/10.1016/j.compeleceng.2022.108383.
63. Khanna, A., Rani, P., Garg, P., Singh, P. K., & Khamparia, A. (2022). An enhanced crow search inspired feature selection technique for intrusion detection based wireless network system. Wireless Personal Communications, 127(3), 2021-2038. https://doi.org/10.1007/s11277-021-08766-9.
64. Halim, Z., Yousaf, M. N., Waqas, M., Sulaiman, M., Abbas, G., Hussain, M., ... & Hanif, M. (2021). An effective genetic algorithm-based feature selection method for intrusion detection systems. Computers & Security, 110, 102448. https://doi.org/10.1016/j.cose.2021.102448.
65. Almasoudy, F. H., Al-Yaseen, W. L., & Idrees, A. K. (2020). Differential evolution wrapper feature selection for intrusion detection system. Procedia Computer Science, 167, 1230-1239. https://doi.org/10.1016/j.procs.2020.03.438.
66. Khammassi, C., & Krichen, S. (2020). A NSGA2-LR wrapper approach for feature selection in network intrusion detection. Computer Networks, 172, 107183. https://doi.org/10.1016/j.comnet.2020.107183
67. Elmasry, W., Akbulut, A., & Zaim, A. H. (2020). Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. Computer Networks, 168, 107042. https://doi.org/10.1016/j.comnet.2019.107042
68. Ogundokun, R. O., Awotunde, J. B., Sadiku, P., Adeniyi, E. A., Abiodun, M., & Dauda, O. I. (2021).
48. Faris, M., Mahmud, M. N., Salleh, M. F. M., & Alsharaa, B. (2023). A differential evolution-based algorithm with maturity extension for feature selection in intrusion detection system. Alexandria Engineering Journal, 81, 178-192. https://doi.org/10.1016/j.aej.2023.09.032
49. Maheswari, K. G., Siva, C., & Nalinipriya, G. (2023). Optimal cluster based feature selection for intrusion detection system in web and cloud computing environment using hybrid teacher learning optimization enables deep recurrent neural network. Computer Communications, 202, 145-153. https://doi.org/10.1016/j.comcom.2023.02.003
50. Rabash, A. J., Nazri, M. Z. A., Shapii, A., & Hasan, M. K. (2023). Non-Dominated Sorting Genetic Algorithm based Dynamic Feature Selection for Intrusion Detection System. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3328395
51. Mohy-Eddine, M., Guezzaz, A., Benkirane, S., & Azrour, M. (2023). An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection. Multimedia Tools and Applications, 82(15), 23615-23633. https://doi.org/10.1007/s11042-023-14795-2
52. Quincozes, S. E., Passos, D., Albuquerque, C., Mossé, D., & Ochi, L. S. (2022). An extended assessment of metaheuristics-based feature selection for intrusion detection in CPS perception layer. Annals of Telecommunications, 77(7), 457-471. https://doi.org/10.1007/s12243-022-00912-z.
53. Saheed, Y. K. (2022). A binary firefly algorithm-based feature selection method on high dimensional intrusion detection data. In Illumination of Artificial Intelligence in Cybersecurity and Forensics (pp. 273-288). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-93453-8_12.
54. Balogun, B. F., Gbolagade, K. A., Arowolo, M. O., & Saheed, Y. K. (2021). A hybrid metaheuristic algorithm for features dimensionality reduction in network intrusion detection system. In Computational Science and Its Applications–ICCSA 2021: 21st International Conference, Cagliari, Italy, September 13–16, 2021, Proceedings, Part IX 21 (pp. 101-114). Springer International Publishing. https://doi.org/10.1007/978-3-030-87013-3_8.
55. Sharma, M., Saini, S., Bahl, S., Goyal, R., & Deswal, S. (2021). Modified bio-inspired algorithms for intrusion detection system. In International Conference on Innovative Computing and Communications: Proceedings of ICICC 2020, Volume 1 (pp. 185-201). Springer Singapore. https://doi.org /10.1007/978-981-15-5113-0_14 .
56. Stankovic, M., Antonijevic, M., Bacanin, N., Zivkovic, M., Tanaskovic, M., & Jovanovic, D. (2022, October). Feature selection by hybrid artificial bee colony algorithm for intrusion detection. In 2022 International Conference on Edge Computing and Applications (ICECAA) (pp. 500-505). IEEE. https://doi.org/10.1109/ICECAA55415.2022.9936116
57. Panigrahi, R., Borah, S., Pramanik, M., Bhoi, A. K., Barsocchi, P., Nayak, S. R., & Alnumay, W. (2022). Intrusion detection in cyber–physical environment

An enhanced intrusion detection system using particle swarm optimization feature extraction technique. Procedia Computer Science, 193, 504-512. https://doi.org/10.1016/j.procs.2021.10.052

69. Velliangiri, S., & Karthikeyan, P. (2020). Hybrid optimization scheme for intrusion detection using considerable feature selection. Neural Computing and Applications, 32(12), 7925-7939. https://doi.org/10.1007/s00521-019-04477-2
70. Alazzam, H., Sharieh, A., & Sabri, K. E. (2020). A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. Expert systems with applications, 148, 113249. https://doi.org/10.1016/j.eswa.2020.113249
71. Maheswari, S., & Arunesh, K. (2020, September). Unsupervised Binary BAT algorithm based Network Intrusion Detection System using enhanced multiple classifiers. In 2020 International Conference on Smart Electronics and Communication (ICOSEC) (pp. 885-889). IEEE. https://doi.org/10.1109/ICOSEC49089.2020.9215453
72. Vijayanand, R., & Devaraj, D. (2020). A novel feature selection method using whale optimization algorithm and genetic operators for intrusion detection system in wireless mesh network. IEEE Access, 8, 56847-56854. https://doi.org/10.1109/ACCESS.2020.2978035
73. Ravindranath, V., Ramasamy, S., Somula, R., Sahoo, K. S., & Gandomi, A. H. (2020, July). Swarm intelligence based feature selection for intrusion and detection system in cloud infrastructure. In 2020 IEEE congress on evolutionary computation (CEC) (pp. 1-6). IEEE. https://doi.org/10.1109/CEC48606.2020.9185887
74. Al Ogaili, R. R. N., Alomari, E. S., Alkorani, M. B. M., Alyasseri, Z. A. A., Mohammed, M. A., Dhanaraj, R. K., ... & Karuppayah, S. (2023). Malware cyberattacks detection using a novel feature selection method based on a modified whale optimization algorithm. Wireless Networks, 1-17. https://doi.org/10.1007/s11276-023-03606-z
75. Alzubi, Q. M., Anbar, M., Alqattan, Z. N., Al-Betar, M. A., & Abdullah, R. (2020). Intrusion detection system based on a modified binary grey wolf optimisation. Neural computing and applications, 32, 6125-6137. https://doi.org/10.1007/s00521-019-04103-1.
76. Alamiedy, T. A., Anbar, M., Alqattan, Z. N., & Alzubi, Q. M. (2020). Anomaly-based intrusion detection system using multi-objective grey wolf optimisation algorithm. Journal of Ambient Intelligence and Humanized Computing, 11(9), 3735-3756. https://doi.org /10.1007/s12652-019-01569-8 .