مجله علمی پژوهشی «علوم و فناوری‌های پدافند نوین»
سال هشتم، شماره ۳، پاییز ۱۳۹۶؛ ص ۲۰۰-۱۹۱

یک روش جدید تشخیص نفوذ سبک وزن برای شبکه‌های رایانه‌ای

مهدی نجفی[۱]، رضا رافع[۲*]
۱- کارشناسی ارشد، ۲- استادیار، دانشگاه اراک
(دریافت: ۹۴/۰۱/۲۸، پذیرش: ۹۵/۰۳/۱۴)

چکیده

انتخاب ویژگی یکی از موضوعات کلیدی در سامانه‌های کشف نفوذ است. یکی از مشکلات طبقه‌بندی در سامانه‌های کشف نفوذ وجود تعداد زیادی ویژگی است که باعث بزرگ شدن فضای حالات می‌شود. بسیاری از این ویژگی‌ها ممکن است نامرتبط یا تکراری باشند که حذف آن‌ها تأثیر قابل توجهی در عملکرد طبقه‌بندی خواهد داشت. الگوریتم رقابت استعماری دارای سرعت همگرایی بالایی برای انتخاب ویژگی‌ها بوده ولی مشکل آن گیر افتادن در بهینه محلی هست. الگوریتم ژنتیک دارای قدرت جستجوی بالا جهت پیدا کردن جواب‌ها هست ولی مشکل آن عدم توانایی در مدیریت جواب‌های یافت شده جهت همگرایی است. بنابراین ترکیب این دو الگوریتم می‌تواند از یک سو سرعت همگرایی و از سوی دیگر دقت در انتخاب ویژگی را به همراه داشته باشد. در این مقاله با اعمال عملگر جذب الگوریتم رقابت استعماری به الگوریتم ژنتیک، روش جدیدی برای انتخاب ویژگی‌های بهینه در سامانه تشخیص نفوذ ارائه می‌شود. روش پیشنهادی با روش طبقه‌بندی درخت تصمیم روی مجموعه داده KDD99 آزمایش شده که نشان دهنده افزایش نرخ تشخیص (٪۹۵/۰۳)، کاهش نرخ هشدار غلط (٪۱/۴۶) و همچنین افزایش سرعت همگرایی (۳/۸۲ ثانیه) است.

کلیدواژه‌ها: طبقه‌بندی، انتخاب ویژگی، تشخیص ناهنجاری، الگوریتم ژنتیک، سامانه‌های تشخیص نفوذ

A New Light Weight Intrusion Detection Algorithm for Computer Networks

M. Najafi, R. Rafeh*
Arak University
(Received: 17/04/2015; Accepted: 03/06/2016)

Abstract

Feature selection is one of the key challenges in developing intrusion detection systems. Classification algorithms in intrusion detection systems may be inconvenient for problems having so many features, because the size of the search space grows exponentially in terms of the number of features. This is while most of the features may be either irrelevant or redundant. Therefore, considering only relevant features (i.e. feature selection) may have a significant impact on the performance of the classification algorithms. The Imperialist Competitive Algorithm (ICA) can be used as a feature selection method with a high convergence, but it sometimes gets trapped in a local optimum. On the contrary, the Genetic Algorithm (GA) is powerful enough in terms of search for solutions, but it suffers from late convergence. Therefore, using a combination of both algorithms for feature selection may result in a rapid convergence as well as in a high precision. In this paper, by applying the Assimilate operator of the ICA to the GA, we propose a new feature selection algorithm for intrusion detection systems. The proposed algorithm has been tested on the KDD99 dataset using the decision tree classification. The experimental results show that the proposed algorithm has improved the detection rate (95.03%), false alarm rate (1.46) and the speed of convergence (3.82 second).

Keywords: Feature Selection, Anomaly Detection, Genetic Algorithm, Intrusion Detection System

* Corresponding Author E-mail: r-rafeh@araku.ac.ir

۱. مقدمه

سامانه‌های تشخیص نفوذ (IDS)[1] یک سازوکار دفاعی بسیار مهم برای نقاط آسیب‌پذیر شبکه‌های رایانه‌ای است. از آنجایی که از نظر فنی ایجاد سامانه‌های رایانه‌ای بدون نقاط ضعف و شکست امنیتی عملاً غیرممکن است؛ تشخیص نفوذ در تحقیقات سامانه‌های رایانه‌ای با اهمیت خاصی دنبال می‌شود. سامانه کشف نفوذ نقش حیاتی در کشف انواع مختلفی از حمله‌ها را ایفاء می‌کند و همچنین ابزار با ارزشی برای دفاع در شبکه‌های رایانه‌ای است. عموماً سامانه‌های تشخیص نفوذ در کنار دیواره‌های آتش و به صورت مکمل امنیتی برای آن‌ها مورد استفاده قرار می‌گیرند. با توجه به حجم زیاد داده‌ای که در ترافیک شبکه وجود دارد استخراج الگوهایی برای رفتارهای نادرست کاربران و پیش‌بینی رفتارهای آنان کاری زمان‌بر و دشوار است. ممکن است بین ویژگی‌های مختلف داده‌ها، ارتباط اشتباهی وجود داشته باشد که مانع از تشخیص درست نفوذ شود. بعضی ویژگی‌ها ممکن است از ویژگی‌های دیگر قابل استنتاج باشند. بنابراین انتخاب ویژگی‌های مناسب، می‌تواند سرعت طبقه‌بندی را افزایش دهد و موجب افزایش کارایی و سرعت سامانه‌های تشخیص نفوذ شود.

هدف اصلی IDS شناسایی استفاده‌های غیرمجاز و سوء استفاده از رایانه‌های خودی توسط کاربران داخلی و مهاجمین است. به دلیل پیشرفت فناوری اینترنت و رشد پیوسته حمله‌های شبکه‌ای، شناسایی نفوذ شبکه یکی از موضوعات مهم برای تحقیق شده است.

در سال‌های گذشته تلاش‌های زیادی برای طبقه‌بندی حمله‌ها انجام گرفته است. یکی از طبقه‌بندی‌هایی که مورد قبول اکثریت واقع شده حمله‌ها را به چهار گروه زیر طبقه‌بندی می‌کند [۱]:

- حمله‌های پویش پورت[2]: حمله‌هایی که بر اساس به‌دست آوردن اطلاعات در مورد سامانه برای پیش بردن نفوذ است.
- حمله‌های از کار اندازی سرویس (Dos[3]): حمله‌هایی که تلاش می‌کنند با ایجاد وقفه کم یا وقفه کامل استفاده از یک سامانه یا سرویس‌ها را برای کاربران قانونی دچار اختلال نمایند.
- حمله‌های کاربر به ریشه (U2R[4]): حمله‌هایی که هدف آن‌ها به‌دست آوردن سطح دسترسی مدیر اصلی[5] سامانه به وسیله استفاده از آسیب‌پذیری‌های موجود در سامانه عامل یا برنامه‌های کاربردی است.
- حمله‌های از راه دور (R2L[6]): حمله‌هایی که بر اساس به‌دست آوردن دسترسی محلی، در خارج از شبکه داخلی است.

روش‌هایی که IDS بر اساس آن کار می‌کند به دو دسته تقسیم می‌شوند [۲]:

- تشخیص الگو[7] (مبتنی بر امضاء)
- تشخیص ناهنجاری[8]

اکثر IDSها از روش تشخیص نفوذ مبتنی بر امضاء استفاده می‌کنند، این نوع سامانه‌های تشخیص نفوذ اکثراً محصول تجاری هستند. سامانه‌های تشخیص نفوذ مبتنی بر ناهنجاری معمولاً برای اهداف تحقیق، طراحی و ساخته می‌شوند و در آینده سامانه‌های تشخیص نفوذ ایده‌آل خواهند بود و جایگزین سامانه‌های تشخیص نفوذ مبتنی بر امضاء خواهند شد [۳ و ۴].

تشخیص نفوذ فرایند نظارت و تحلیل رخدادهای اتفاق افتاده در یک سامانه رایانه‌ای جهت شناسایی کردن نشانه‌هایی از مشکلات امنیتی است [۵]. به عبارت دیگر تشخیص نفوذ مجموعه‌ای از فن‌ها و روش‌هایی است که برای تشخیص فعالیت‌های مشکوک در هر دو سطح شبکه[9] و میزبان[10] استفاده می‌شود [۶]. تشخیص نفوذ مبتنی بر میزبان بر روی یک سرور، دیده‌بانی یک کاربر خاص، عمل برنامه‌های کاربردی و ثبت وقایع[11] متمرکز می‌شود ولی تشخیص نفوذ مبتنی بر شبکه، ترافیک شبکه سیمی را برای فعالیت‌های خاص یا امضاءهایی که نشان‌دهنده یک حمله هستند مورد نظارت قرار می‌دهد [۷].

یک سامانه تشخیص نفوذ را می‌توان مجموعه‌ای از ابزارها، روش‌ها و مدارکی در نظر گرفت که به شناسایی، تعیین و گزارش فعالیت‌های غیرمجاز یا تأیید نشده تحت شبکه، کمک می‌کند.

سامانه‌های تشخیص نفوذ (IDS) برای کمک به مدیران امنیتی سامانه جهت کشف نفوذ و حمله به‌کار گرفته شده‌اند. هدف یک سامانه تشخیص نفوذ جلوگیری از حمله نیست و تنها کشف و احتمالاً شناسایی حمله‌ها و تشخیص اشکالات امنیتی در سامانه یا شبکه‌های رایانه‌ای و اعلام آن به مدیر سامانه است. عموماً سامانه‌های تشخیص نفوذ در کنار دیواره آتش[12] و به صورت مکمل امنیتی برای آن مورد استفاده قرار می‌گیرند.

[1] Intrusion Detection System
[2] Probing
[3] Denial of Service
[4] User to Root
[5] Super User
[6] Remote to Local
[7] Misuse Detection
[8] Anomaly Detection
[9] Network-Based
[10] Host-Based
[11] Log
[12] Firewall

مجموعه داده KDD99، مجموعه داده استاندارد برای ارزیابی سامانه‌های تشخیص نفوذ است. این مجموعه داده اعتبار خود را از سومین مسابقه بین‌المللی کشف دانش و داده کاوی کسب کرده است. بهترین دلیل انتخاب مجموعه داده KDD99، استفاده زیاد و جامع از این مجموعه داده است که به وسیله تعداد زیادی محقق به اشتراک گذاشته می‌شود.

ویژگی‌های ارائه شده در KDD99 به سه دسته ویژگی‌های پایه، ویژگی‌های محتوایی و ویژگی‌های ترافیکی گروه‌بندی می‌شوند.

ویژگی‌های پایه شامل تمام ویژگی‌هایی است که از اتصال TCP/IP استخراج می‌شوند. این ویژگی‌ها از سرآیند بسته خارج می‌شوند (Src, Dst, Protocol,…).

ویژگی‌های محتوایی برای ارزیابی داده مفید (payload) بسته‌های TCP/IP و جستجو برای رفتارهای مشکوک در آن است. حمله‌های R2L و U2R توسط این ویژگی‌ها شناسایی می‌شوند.

ویژگی‌های ترافیکی شامل ویژگی‌هایی می‌باشند که میزبان مقصد مشابه یا سرویس مشابه داشته باشند که با بازرسی آن‌ها و محاسبه آمار آن‌ها بر اساس زمان می‌باشند.

سامانه تشخیص نفوذ با استفاده از خوشه‌بندی فازی جهت تشخیص ناهنجاری پیشنهاد شده است [۸] و ترکیب الگوریتم CBHFS و ماشین‌بردار پشتیبان برای انتخاب ویژگی‌ها و الگوریتم ژنتیک به عنوان راهبرد جستجو استفاده شده است [۹]. از فن‌های متفاوت داده کاوی مثل دسته‌بندی، خوشه‌بندی و رویکردهای ترکیبی یادگیری از قبیل فن‌های ترکیب دسته‌بندی و خوشه‌بندی برای شناسایی نفوذ استفاده شده است [۵]. برای طراحی سامانه تشخیص نفوذ سه مرحله در نظر گرفته شده است [۱۰]. مرحله اول داده‌های تکراری را حذف کرده است. مرحله دوم یک زیرمجموعه بهینه از ویژگی‌ها را توسط الگوریتم ژنتیک انتخاب کرده و سپس در مرحله سوم از درخت تصمیم برای به‌دست آوردن دقت بالاتر استفاده شده است. از الگوریتم ژنتیک و ماشین‌بردار پشتیبان برای انتخاب ژن برای طبقه‌بندی سرطان استفاده شده است [۱۱]. در گزارشی، ترکیب الگوریتم ژنتیک و K نزدیک‌ترین همسایه برای انتخاب ویژگی‌ها ارائه شده است [۱۲]، در این گزارش، از ۳۵ ویژگی در مرحله آموزش استفاده شده است. همه ۳۵ ویژگی که در شروع فاز آموزش بودند را ابتدا وزن‌بندی کرده است، سپس بهترین آن‌ها برای پیاده‌سازی و آزمون انتخاب شدند. با استفاده از الگوریتم ژنتیک و روش طبقه‌بندی ماشین‌بردار پشتیبان انتخاب ویژگی‌های بهینه انجام شده است [۱۳ و ۱۴]. از نسخه تغییر یافته تپه‌نوردی و ماشین‌بردار پشتیبان برای انتخاب ویژگی استفاده شده است [۱۵]. از ماشین‌بردار پشتیبان، درخت تصمیم و الگوریتم ذوب فلزات استفاده شده است که ماشین‌بردار پشتیبان و ذوب فلزات، بهترین ویژگی‌ها را جهت بالا بردن نرخ تشخیص نفوذ پیدا می‌کنند و درخت تصمیم و ذوب فلزات قوانین تصمیم را برای حمله‌های جدید ایجاد می‌کنند [۱۶]. زیرمجموعه‌ای از ویژگی های بهینه توسط الگوریتم ده‌پا[1] (CFA) و روش دسته‌بندی درخت تصمیم روی مجموعه داده KDD99 انتخاب شده است [۱۷]. از منطق فازی، محاسبات نرم و فن‌های هوش مصنوعی جهت کاهش نرخ هشدار غلط استفاده شده است [۱۸]. از الگوریتم ژنتیک خطی برای بهبود نرخ تشخیص و از الگوریتم زنبور عسل برای کاهش نرخ هشدار غلط استفاده شده و سپس نتیجه کار به الگوریتم دسته‌بندی ماشین‌بردار پشتیبان جهت دسته‌بندی با دقت بالا داده شده است [۱۹]. با انتخاب ویژگی‌های مهم و مؤثر می‌توان طبقه‌بندی و عملکرد آن را بهبود بخشید. انتخاب‌گر ویژگی وظیفه پیدا کردن زیرمجموعه‌ای از ویژگی‌ها برای بهبود دقت پیش‌بینی را دارد. به طور کلی خلاصه مراحلی که در سامانه‌های تشخیص نفوذ انجام می‌شود شامل شش مرحله است. مرحله اول شامل جمع‌آوری داده‌هاست. در مرحله دوم یک پیش‌پردازش روی داده‌های جمع‌آوری شده انجام می‌شود و طی مرحله سوم انتخاب ویژگی‌های مهم و ضروری توسط الگوریتم هوشمند انجام می‌شود. در مرحله چهارم پس پردازش ویژگی‌های ضروری جهت استفاده در الگوریتم‌های طبقه‌بندی با نرمال‌سازی و تبدیل به فرمت مناسب انجام می‌شود. مرحله پنجم شامل اعمال روش طبقه‌بندی روی ویژگی‌های انتخاب شده است. درنهایت، مرحله ششم ناهنجاری موجود در شبکه برای انواع حمله‌ها را گزارش می‌دهد.

انتخاب ویژگی یکی از مهم‌ترین مراحل در سامانه‌های تشخیص نفوذ است. انتخاب ویژگی‌های بهینه کاری زمان‌بر است ولی با توجه به اینکه در حالت برون خطی[2] اجرا می‌شود و ویژگی‌های انتخاب شده برای سامانه‌های تشخیص نفوذ واقعی و بر خط[3] مورد استفاده قرار می‌گیرد، بنابراین زمان‌بر بودن آن مشکلی ایجاد نمی‌کند.

هدف این مقاله ارائه یک سامانه تشخیص نفوذ سبک وزن با افزایش نرخ تشخیص و کاهش نرخ هشدار اشتباه است. با توجه به اینکه الگوریتم ژنتیک دارای قدرت بالایی جهت انتخاب ویژگی‌های بهینه است، در این مقاله با اعمال عملگر جذب[4] الگوریتم رقابت استعماری به الگوریتم ژنتیک، روش جدیدی برای انتخاب ویژگی‌های بهینه در سامانه تشخیص نفوذ ارائه شده است.

[1] Cuttlefish Optimization Algorithm
[2] Offline
[3] Online
[4] Assimilation

ادامه مقاله بدین ترتیب سازمان‌دهی می‌شود: الگوریتم‌های بهینه‌سازی هوشمند در بخش ۲ آمده است. در بخش ۳ ارزیابی روش پیشنهادی و در بخش ۴ نتیجه‌گیری شرح داده می‌شود.

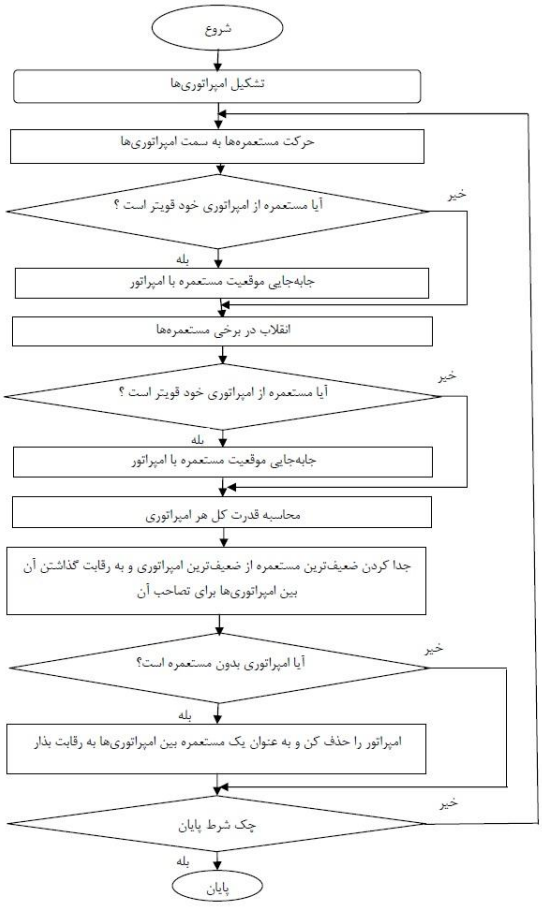
۲. الگوریتم‌های بهینه‌سازی هوشمند

الگوریتم‌های بهینه‌سازی الهام گرفته از طبیعت به عنوان روش‌های هوشمند بهینه‌سازی در کنار روش‌های کلاسیک موفقیت فراوانی از خود نشان داده‌اند. در این بخش شرح مختصری از دو الگوریتم ژنتیک و رقابت استعماری داده شده است.

۲-۱. الگوریتم رقابت استعماری

استعمار یک پدیده ذاتی در تاریخ بوده است. کشورهای استعمارگر رقابت شدیدی را برای به استعمار کشیدن مستعمرات همدیگر نشان می‌دادند. این رقابت به نوبه خود باعث رشد و توسعه کشورهای استعمارگر از لحاظ سیاسی، نظامی و اقتصادی شد، زیرا کشورها برای داشتن امکان رقابت، مجبور به توسعه بودند.

روند نمای الگوریتم رقابت استعماری که در [۲۰ و ۲۱] استفاده شده، در شکل (۱) نشان داده شده است.

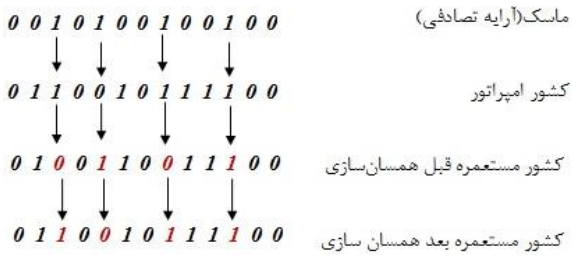


شکل ۱. روند نمای الگوریتم رقابت استعماری [۲۰ و ۲۱]

در الگوریتم رقابت استعماری بعد از تولید جمعیت اولیه و تشکیل امپراتوری‌ها، دو عملگر جذب و انقلاب روی مستعمره‌ها اعمال می‌شود. سپس به رقابت درون امپراتوری بین مستعمره‌ها و امپراتور مربوطه برای تصاحب امپراتوری و سپس رقابت برون امپراتوری بین امپراتورها با هم جهت گرفتن ضعیف‌ترین مستعمره از ضعیف‌ترین امپراتوری انجام می‌گیرد. در ادامه عملگرها توضیح داده خواهند شد.

با توجه به اینکه هر رکورد از مجموعه داده KDD99 دارای ۴۱ ویژگی است، بنابراین هر رکورد در الگوریتم رقابت استعماری به صورت یک کشور شامل ۴۱ ویژگی نمایش داده می‌شود. هر ویژگی به صورت یک عدد باینری (۰ یا ۱) نشان داده می‌شود. وجود یک در هر ویژگی یعنی ویژگی متناظر با آن انتخاب شده و وجود صفر به معنای عدم انتخاب آن ویژگی است.

جذب: در این عملگر سعی می‌شود مستعمره‌ها به سمت امپراتوری خود حرکت نمایند و باعث می‌شود مستعمره‌ها خیلی شبیه و نزدیک به امپراتوری خود شوند. در شکل (۲) عملگر جذب نشان داده شده است، برای پیاده‌سازی آن یک آرایه به تعداد ویژگی‌های موجود در هر کشور به صورت تصادفی بین صفر و یک ایجاد می‌شود در مکان‌هایی که این آرایه دارای عدد یک است مستعمره ویژگی امپراتوری خود را جایگزین ویژگی‌های خود می‌کند که درصدی از ویژگی‌های مستعمره از امپراتوری گرفته شده که باعث می‌شود مستعمره هر بار کمی به سمت امپراتوری حرکت نماید و به آن شبیه‌تر شود.

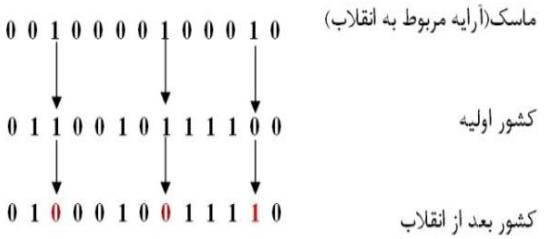


شکل ۲. عملگر جذب

انقلاب[1]: عملگر انقلاب بعضی از ویژگی‌های یک کشور را تغییر می‌دهد. با توجه به اینکه هر کشور دارای ۴۱ ویژگی است و به صورت صفر و یک نشان داده می‌شود، عملگر انقلاب باعث می‌شود به صورت تصادفی بعضی از ویژگی‌های مربوط به یک کشور حذف و یا اضافه گردد. برای حذف یک ویژگی مکان مربوط به آن ویژگی از یک به صفر تغییر پیدا خواهد کرد و برای اضافه کردن یک ویژگی مکان مربوط به آن ویژگی از صفر به یک تغییر می‌کند.

[1] Revolution

در شکل (۳) عملگر انقلاب نشان داده شده است، برای پیاده‌سازی آن در این مسئله ابتدا به تعداد انقلاب (احتمال انقلاب * ۴۱)، عدد از ۱ تا ۴۱ تولید می‌شود و مکان مربوط به آن ویژگی از یک به صفر یا از صفر به یک تغییر داده می‌شود؛ که این عملگر به فرار کردن از بهینه محلی کمک می‌کند.



شکل ۳. عملگر انقلاب الگوریتم رقابت استعماری

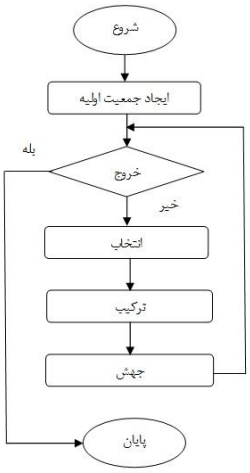
در حین اجرای عملگرهای جذب و انقلاب روی مستعمره‌ها ممکن است قدرت مستعمره‌ها بیشتر از امپراتور خود گردد. در صورتی که قدرت مستعمره بیشتر از امپراتوری بود جای مستعمره با امپراتوری خود جابه‌جا می‌شود.

رقابت بین امپراتوری: در هر بار اجرا، امپراتورها برای به‌دست آوردن ضعیف‌ترین مستعمره از ضعیف‌ترین امپراتوری رقابت می‌کنند. برای این کار با چرخ رولت یکی از امپراتورها انتخاب می‌شود و مستعمره ضعیف مربوط به ضعیف‌ترین امپراتوری به آن اختصاص داده می‌شود.

حذف امپراتوری ضعیف: امپراتوری که هیچ مستعمره‌ای ندارد حذف شده و به صورت یک مستعمره بین امپراتورها به رقابت گذاشته می‌شود.

۲-۲. الگوریتم ژنتیک

روند نمای الگوریتم ژنتیک که در [۲۲] استفاده شده، در شکل (۴) نشان داده شده است. در الگوریتم ژنتیک بعد از تولید جمعیت اولیه سه عملگر انتخاب، ترکیب و جهش روی جمعیت اعمال می‌گردد.



شکل ۴. روند نمای الگوریتم ژنتیک [۲۲]

عملگر انتخاب[1]: برای انتخاب کروموزوم‌ها از چرخ رولت استفاده شده است.

عملگر ترکیب[2]: برای اعمال عملگر ترکیب از یک ماسک استفاده می‌شود، این ماسک به صورت تصادفی ایجاد می‌گردد. برای تولید فرزند اول در مکان‌هایی از ماسک که دارای عدد ۱ است از والد ۱ و در مکان‌هایی از ماسک که دارای عدد ۰ است از والد ۲ به فرزند اول کپی می‌شود. برای تولید فرزند دوم برعکس این عمل می‌شود یعنی در مکان‌هایی از ماسک که دارای عدد ۰ است از والد ۱ و در مکان‌هایی از ماسک که دارای عدد ۱ است از والد ۲ به فرزند دوم کپی می‌شود.

عملگر جهش[3]: ابتدا یک آرایه تصادفی به اندازه تعداد ویژگی‌ها (اندازه کروموزوم) به طوری که به اندازه احتمال جهش از آن دارای عدد ۱ و بقیه ۰ باشند ایجاد می‌شود و سپس در مکان‌هایی از آرایه جدید که دارای عدد ۱ هست، بیت متناظر در کروموزوم برعکس می‌شود (صفر به یک و یک به صفر تغییر داده می‌شود).

۳. الگوریتم ژنتیک بهبود یافته (روش پیشنهادی)

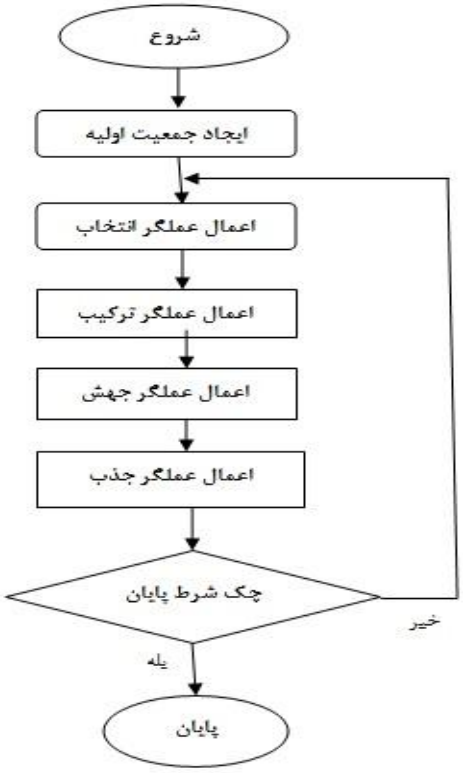
افزایش کارایی الگوریتم کاهش ویژگی درنهایت می‌تواند باعث افزایش کارایی الگوریتم طبقه‌بندی شود. در این بخش با اعمال عملگر جذب الگوریتم رقابت استعماری به الگوریتم ژنتیک در عملیات انتخاب ویژگی، بهترین ویژگی‌ها انتخاب و کارایی بهبود داده می‌شود. در ادامه قسمت‌های مهم الگوریتم ژنتیک بهبود یافته شرح داده خواهند شد.

هر رکورد از مجموعه داده KDD99 دارای ۴۱ ویژگی است بنابراین هر رکورد به صورت یک کروموزوم شامل ۴۱ ژن نمایش داده می‌شود که هر ژن به صورت یک عدد دودویی (۰ یا ۱) می‌تواند باشد. کشور در الگوریتم رقابت استعماری معادل با کروموزوم در الگوریتم ژنتیک است. وجود ۱ در هر ژن یعنی ویژگی متناظر با آن انتخاب شده و وجود ۰ به معنای عدم انتخاب آن ویژگی است. از ۴۱ ویژگی موجود در مجموعه داده KDD99، ۳۸ ویژگی عددی و ۳ ویژگی نمادی[4] است. برای استفاده از این مجموعه داده ابتدا پیش‌پردازشی روی این مجموعه داده انجام شده است و سه ویژگی نمادی به عددی تبدیل شده است.

روند نمای الگوریتم ژنتیک بهبود یافته در شکل (۵) نشان داده شده است. در الگوریتم ژنتیک بعد از تولید جمعیت اولیه سه عملگر انتخاب، ترکیب و جهش روی جمعیت اعمال می‌گردد.

[1] Selection
[2] Crossover
[3] Mutation
[4] Symbol

سپس عملگر جذب الگوریتم رقابت استعماری به جمعیت اعمال می‌شود.



شکل ۵. روند نمای الگوریتم ژنتیک بهبود یافته (روش پیشنهادی)

عملگر انتخاب: برای انتخاب کروموزوم‌ها از چرخ رولت استفاده شده است. این عملگر ۲۰ درصد از جمعیت را انتخاب می‌کند و سپس عملگر ترکیب به آن‌ها اعمال می‌شود. ۱۰ درصد از این جمعیت با جمعیت نخبه (۱۰ درصد از بهترین جمعیت) به صورت جفت انتخاب می‌شوند. ۱۰ درصد دیگر به صورت تصادفی انتخاب می‌شوند. انتخاب ۱۰ درصد اول برای ترکیب، امکان تولید جمعیتی قوی‌تر را به دلیل ترکیب نخبه با جمعیت دیگر به وجود خواهد آورد.

عملگر ترکیب: تغییری که در عملگر ترکیب الگوریتم پیشنهادی، نسبت به الگوریتم ژنتیک اولیه داده شده این است که در الگوریتم پیشنهادی نسل جدید به وجود آمده از عملگر ترکیب، جایگزین ضعیف‌ترین فرزندهای موجود در جمعیت می‌شوند در حالی که در الگوریتم ژنتیک پایه نسل جدید به وجود آمده از عملگر ترکیب جایگزین پدران خود که انتخاب شده بودند می‌شوند که ممکن است نسل جدید تولید شده ضعیف‌تر از نسل قبل بوده و موجب از بین رفتن بهترین جواب‌های به‌دست آمده گردند.

عملگر جهش: تغییری که در عملگر جهش الگوریتم پیشنهادی داده شده به این صورت است که عملگر جهش روی ۱۰ درصد نخبه اعمال می‌شود و وقتی کروموزوم ضعیف‌تری تولید شود این عملگر نادیده گرفته می‌شود؛ به عبارت دیگر در صورتی که حاصل اعمال عملگر جهش روی جمعیت نخبه بهتر از حالت اولیه آن باشد پذیرفته می‌شود، در غیر این صورت نادیده گرفته خواهد شد.

عملگر جذب: عملگر جذب الگوریتم رقابت استعماری به الگوریتم ژنتیک اعمال شده است. در الگوریتم ژنتیک برای اعمال عملگر جذب، تعداد ۲۰ درصد از جمعیت به سمت جمعیت نخبه حرکت می‌کنند که باعث می‌شود جمعیت انتخابی خیلی شبیه و نزدیک به جمعیت نخبه شوند. این عملگر موجب می‌شود که جمعیتی قوی‌تر از جمعیت موجود ایجاد گردد و قدرت الگوریتم ژنتیک را افزایش می‌دهد.

۴. ارزیابی روش پیشنهادی

انتخاب ویژگی‌های بهینه برای طراحی سامانه تشخیص نفوذ سبک وزن با سرعت و دقت بالاتر استفاده می‌شود. در این مقاله انتخاب ویژگی‌های بهینه توسط الگوریتم ژنتیک بهبود یافته روی مجموعه داده KDD99 انجام شده است. در ادامه مجموعه داده KDD99 و معیارهای مورد بررسی شرح داده شده و سپس نتایج و تفسیر آن‌ها ارائه شده است.

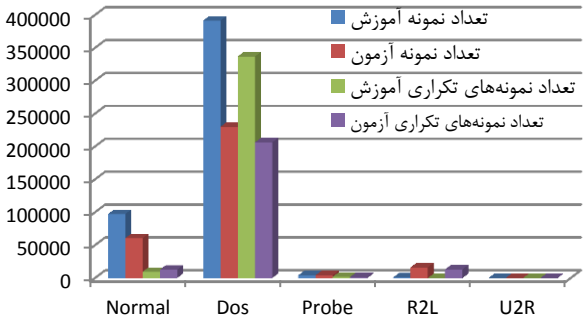
۴-۱. مجموعه داده KDD99

مجموعه داده KDD99 دارای پنج کلاس Normal، Dos، Probe، R2L و U2R است که هر کلاس شامل چندین زیر کلاس است. کلاس‌ها و زیر کلاس‌های موجود در KDD99 در جدول (۱) نشان داده شده است.

برای ارزیابی از ۱۰٪ مجموعه داده KDD99 استفاده شده است. اطلاعات مربوط به ۱۰٪ مجموعه داده KDD99 در شکل (۶) نشان داده شده است.

جدول ۱. کلاس‌ها و زیرکلاس‌های KDD99

ردیف	کلاس	زیرکلاس‌ها
۱	Normal	Normal
۲	Dos	Smurf, Neptune, Back, Teardrop, Pod, Land
۳	Probe	Satan, Ipsweep, Portsweep, Nmap
۴	R2L	Warezclient, Guess_Passwd, Warezmaster, Imap, Ftp_Write, Multihop, Phf, Spy
۵	U2R	Buffer_Overflow, Rootkit, Oadmodule, Perl



شکل ۶. ۱۰٪ مجموعه داده KDD99

مجموعه kddcup.data_10_percent به عنوان مجموعه آموزش[1] و corrected به عنوان مجموعه آزمون در نظر گرفته شده است. مجموعه آزمون از مجموعه داده KDD99 دارای ۱۹/۴۸٪ نمونه Normal، ۷۳/۹۰٪ نمونه Dos، ۱/۳۴٪ نمونه Probe، ۵/۲۰٪ نمونه R2L، ۰/۰۷٪ نمونه U2R است. با توجه به وجود نمونه‌های تکراری برای کلاس Dos معمولاً طبقه‌بندی به سمت این کلاس متمایل می‌شود.

۴-۲. معیارهای ارزیابی

نتایج به‌دست آمده از الگوریتم‌های فوق بر اساس ماتریس اغتشاش[2] که در جدول (۲) نشان داده شده، معیار نرخ تشخیص[3] (DR)، معیار نرخ هشدار اشتباه[4] (FAR)، معیار دقت[5] و معیار تابع ارزیابی مورد بررسی قرار خواهد گرفت. در ادامه پارامترها شرح داده شده است.

جدول ۲. ماتریس اغتشاش [۲]

Predict Class			
Attack	Normal	Data Class	
False Positive (FP)	True Negative (TN)	Normal	Actual class
True Positive (TP)	False Negative (FN)	Attack	

TP: نمونه‌ای ناهنجار باشد و ناهنجار تشخیص داده شود.

FP: نمونه‌ای هنجار باشد و ناهنجار تشخیص داده شود.

TN: نمونه‌ای هنجار باشد و هنجار تشخیص داده شود.

FN: نمونه‌ای ناهنجار باشد و هنجار تشخیص داده شود.

نرخ تشخیص (DR) تعداد نمونه‌های ناهنجار (نفوذ) شناسایی شده توسط سامانه تشخیص نفوذ تقسیم بر تعداد کل نمونه‌های ناهنجار (نفوذ) موجود در مجموعه آزمون است. نرخ تشخیص با معیارهای Recall، True Positive Rate و Sensitivity معادل است. برای محاسبه نرخ تشخیص در طبقه‌بندی چند کلاسه از رابطه (۱ و ۲) استفاده می‌شود. Ni تعداد نمونه‌های کلاس i است.

$$DR(Class_i) := \frac{TP(Class_i)}{TP(Class_i) + FN(Class_i)} \quad (۱)$$

$$Weighted\ DR := \frac{\sum_{i=1}^{5} DR(Class_i) * N_i}{\sum_{i=1}^{5} N_i} \quad (۲)$$

نرخ هشدار اشتباه (FAR) تعداد نمونه‌های هنجار که به اشتباه به عنوان نفوذ تشخیص داده شده تقسیم بر تعداد کل نمونه‌های هنجار موجود در مجموعه آزمون است. برای محاسبه نرخ هشدار اشتباه در طبقه‌بندی چند کلاسه از رابطه (۳ و ۴) استفاده می‌شود.

$$FAR(Class_i) := \frac{FP(Class_i)}{FP(Class_i) + TN(Class_i)} \quad (۳)$$

$$Weighted\ FAR := \frac{\sum_{i=1}^{5} FAR(Class_i) * N_i}{\sum_{i=1}^{5} N_i} \quad (۴)$$

تابع ارزیابی شامل درصدی از میزان دقت و درصدی از معکوس تعداد ویژگی‌های (تعداد یک‌های موجود در کروموزوم) استفاده شده است. با این تابع ارزیابی کروموزومی که دقت بالاتر و تعداد ویژگی‌های کمتری داشته باشد انتخاب می‌گردد. برای محاسبه تابع ارزیابی از رابطه (۵) استفاده می‌شود.

$$Fitness := W1 * Accuracy + W2 * (1/Sum\ Of\ Ones) \quad (۵)$$

W1 ضریب دقت طبقه‌بندی، W2 ضریب معکوس تعداد یک‌های استفاده شده و Accuracy دقت الگوریتم طبقه‌بندی است.

معیار دقت نسبت داده‌های از یک کلاس که به درستی پیش‌بینی شده‌اند به تمام داده‌های پیش‌بینی شده برای آن کلاس است. برای محاسبه دقت در طبقه‌بندی چند کلاسه از رابطه (۳ و ۴) استفاده می‌شود.

$$Precision(Class_i) := \frac{TP(Class_i)}{TP(Class_i) + FP(Class_i)} \quad (۶)$$

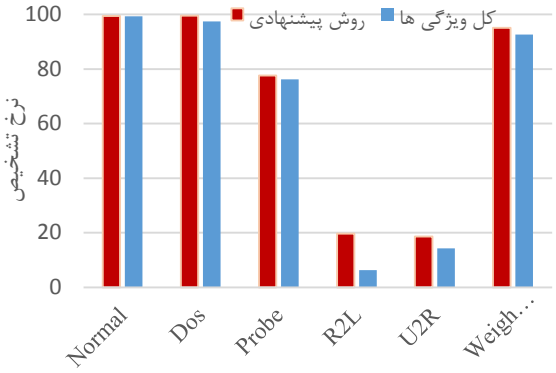
$$Weighted\ Precision := \frac{\sum_{i=1}^{5} Precision(Class_i) *}{\sum_{i=1}^{5} N_i} \quad (۷)$$

۵. نتایج و بحث

روش پیشنهادی برای به‌دست آوردن زیرمجموعه‌ای بهینه از ویژگی‌ها روی مجموعه داده KDD99 در نرم‌افزار متلب پیاده‌سازی و اجرا شد. نتایج حاصل از اجرای آن با توجه به معیارهای ذکر شده به شرح ذیل است:

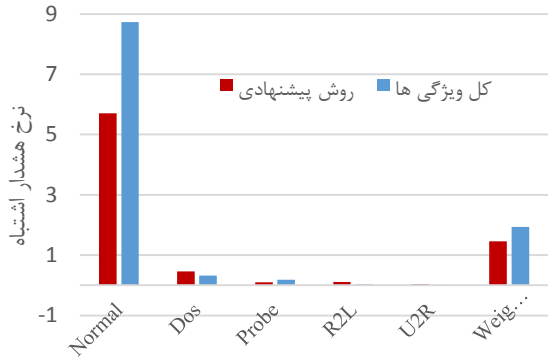
[1] Train
[2] Confusion Matrix
[3] Detection Rate
[4] False Alarm Rate
[5] Precision

نرخ تشخیص میزان حمله‌هایی که به درستی توسط سامانه تشخیص داده شده را نشان می‌دهد. شکل (۷) مقایسه نرخ تشخیص ویژگی‌های انتخاب شده توسط روش پیشنهادی با کل ویژگی‌ها (بدون انتخاب ویژگی) به تفکیک کلاس‌های Normal، Dos، Probe، R2l، U2R و نرخ تشخیص وزنی[1] را نمایش می‌دهد. همان‌طور که در شکل (۷) نشان داده شده است با ویژگی‌های انتخاب شده توسط الگوریتم ژنتیک بهبود یافته، نرخ تشخیص برای هر چهار کلاس حمله افزایش پیدا کرده است.



شکل ۷. مقایسه نرخ تشخیص الگوریتم پیشنهادی بر اساس حملات مختلف

نرخ هشدار اشتباه میزان پیغام‌هایی هست که اشتباهی به مدیر داده می‌شود. وقتی سامانه تشخیص نفوذ یک بسته نرمال را حمله در نظر می‌گیرد یک پیغام به مدیر سامانه ارسال می‌گردد. شکل (۸) مقایسه نرخ هشدار اشتباه ویژگی‌های انتخاب شده توسط الگوریتم ژنتیک بهبود یافته با کل ویژگی‌ها به تفکیک کلاس‌های Normal، Dos، Probe، R2l، U2R و نرخ هشدار اشتباه وزنی را نمایش می‌دهد.



شکل ۸. مقایسه نرخ هشدار اشتباه الگوریتم پیشنهادی بر اساس حملات مختلف

با ویژگی‌های انتخاب شده توسط الگوریتم پیشنهادی، نرخ هشدار اشتباه برای هر پنج کلاس (چهار کلاس حمله و کلاس Normal) به جز کلاس Dos و R2L کاهش پیدا کرده است. ویژگی‌های انتخاب شده برای کل کلاس‌ها بر اساس نرخ تشخیص شامل ۲۲ ویژگی به صورت زیر است:

{۱،۲،۴،۵،۷،۹،۱۱،۱۳،۱۶،۲۱،۲۳،۲۵،۲۶،۳۰،۳۲،۳۳،۳۶،۳۷،۳۸،۳۹،۴۰،۴۱}

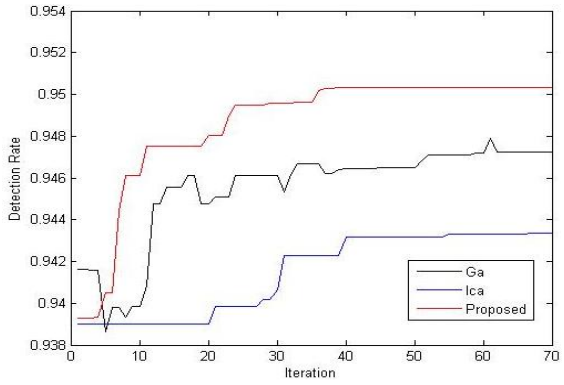
مقایسه سرعت اجرای طبقه‌بندی درخت تصمیم روی ویژگی‌های انتخاب شده توسط روش پیشنهادی و کل ویژگی‌ها در جدول (۳) نمایش داده شده است.

جدول ۳. مقایسه سرعت ویژگی‌های انتخابی

	تعداد ویژگی	مدت زمان اجرا
الگوریتم پیشنهادی	۲۲	۳/۸۲ ثانیه
کل ویژگی‌ها (بدون انتخاب ویژگی)	۴۱	۷/۶۸ ثانیه

همان‌طور که در جدول (۳) مشاهده می‌شود، سرعت اجرای طبقه‌بندی درخت تصمیم روی ویژگی‌های انتخاب شده توسط روش پیشنهادی نسبت به حالت بدون انتخاب ویژگی بهبود پیدا کرده است.

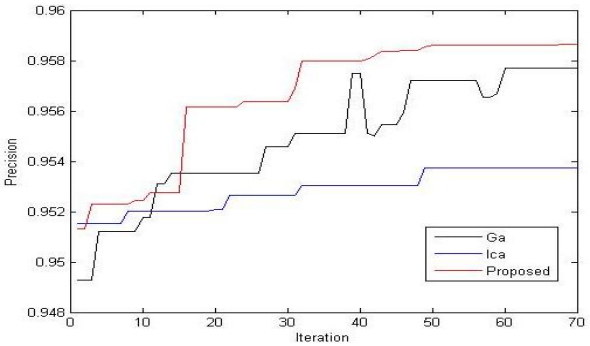
شکل (۹) مقایسه نرخ تشخیص الگوریتم پیشنهادی با الگوریتم ژنتیک و الگوریتم رقابت استعماری را نمایش می‌دهد. در شکل (۹) مشاهده می‌شود الگوریتم پیشنهادی دارای سرعت همگرا شدن خیلی سریع‌تر و همچنین نرخ تشخیص بالاتری نسبت به دو الگوریتم ژنتیک و رقابت استعماری است. الگوریتم پیشنهادی در تکرار ۱۱ به نرخ تشخیص مطلوب ۹۴/۷۵ رسیده است که الگوریتم ژنتیک و الگوریتم رقابت استعماری بعد از تکرار ۷۰ امین بار هم نتوانسته‌اند به این نرخ تشخیص دست پیدا نمایند که این نشان دهنده سرعت همگرایی و دقت بالای الگوریتم پیشنهادی است. دلیل همگرایی سریع الگوریتم پیشنهادی اعمال عملگر جذب الگوریتم رقابت استعماری به الگوریتم ژنتیک است. الگوریتم ژنتیک دارای قدرت جستجوی بالا جهت پیدا کردن جواب‌ها هست ولی مشکل آن عدم توانایی در مدیریت جواب‌های یافت شده جهت همگرایی است. استفاده از عملگر جذب در الگوریتم ژنتیک موجب می‌شود که نمونه‌ها با سرعت بالاتری شبیه به نخبه‌ها شده و سریع‌تر به همگرایی برسند.



شکل ۹. مقایسه نرخ تشخیص الگوریتم پیشنهادی با روش‌های GA و ICA

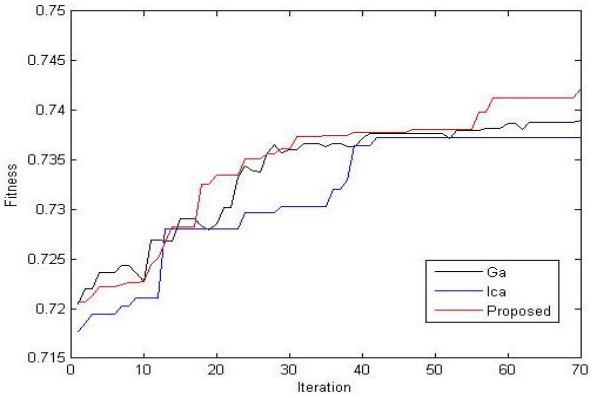
[1] Weighted

شکل (۱۰) مقایسه دقت الگوریتم پیشنهادی با الگوریتم ژنتیک و الگوریتم رقابت استعماری را نمایش می‌دهد. در شکل (۱۰) مشاهده می‌شود الگوریتم پیشنهادی دارای سرعت همگرا شدن سریع‌تر و همچنین دارای مقدار Precision بالاتری نسبت به دو الگوریتم ژنتیک و رقابت استعماری است.



شکل ۱۰. مقایسه دقت الگوریتم پیشنهادی با روش‌های GA و ICA

شکل (۱۱) مقایسه تابع ارزیابی الگوریتم پیشنهادی با الگوریتم ژنتیک و الگوریتم رقابت استعماری را نمایش می‌دهد. در شکل (۱۱) مشاهده می‌شود الگوریتم پیشنهادی دارای مقدار تابع ارزیابی بالاتری نسبت به دو الگوریتم ژنتیک و رقابت استعماری است.



شکل ۱۱. مقایسه تابع ارزیابی الگوریتم پیشنهادی با روش‌های GA و ICA

۶. نتیجه‌گیری

در این مقاله با اعمال عملگر جذب الگوریتم رقابت استعماری به الگوریتم ژنتیک در عملیات انتخاب ویژگی، بهترین ویژگی‌ها انتخاب و کارایی بهبود داده شد. روش پیشنهادی بر روی مجموعه داده KDD99 برای به‌دست آوردن زیرمجموعه‌ای بهینه از ویژگی‌ها اجرا شد. نتایج حاصل از اجرای آن نشان دهنده آن است که در روش درخت تصمیم با ویژگی‌های انتخاب شده توسط روش پیشنهادی در مقایسه با الگوریتم ژنتیک پایه، الگوریتم رقابت استعماری و حالت بدون انتخاب ویژگی، نرخ تشخیص بالاتر (٪۹۵/۰۳) و نرخ هشدار اشتباه کمتر (٪۱/۴۶) می‌شود. با مقایسه سرعت، مشاهده شد روش پیشنهادی تنها از ۲۲ ویژگی بهینه جهت تشخیص نفوذ استفاده می‌کند که دارای سرعت بالاتری نسبت به حالت بدون انتخاب ویژگی است. در نتیجه روش پیشنهادی در این مقاله می‌تواند روش مناسبی برای انتخاب ویژگی‌های بهینه محسوب شود. از جمله کارهایی که در ادامه این تحقیق می‌توان انجام داد، استفاده از ترکیب الگوریتم ژنتیک و الگوریتم ازدحام ذرات جهت انتخاب ویژگی‌های بهینه است. ترکیب دو الگوریتم می‌تواند به طور مؤثری یک مجموعه بهینه از ویژگی‌ها را با سرعت همگرایی بالا انتخاب نماید.

۷. مراجع

[1] Catania, C. A.; Garino, C. G. "Automatic Network Intrusion Detection: Current Techniques and Open Issues"; Computers & Electrical Eng. 2012, 38, 1062-1072.

[2] Balajinath, B.; Raghavan, S. V. "Intrusion Detection through Learning Behavior Model"; Computer Communications 2001, 24, 1202-1212.

[3] Wu, S. X.; Banzhaf, W. "The Use of Computational Intelligence in Intrusion Detection Systems: A Review"; Applied Soft Computing 2010, 10, 1-35.

[4] Poston, H. E. "A Brief Taxonomy of Intrusion Detection Strategies"; Proc. of the Aerospace and Electronics Conf. 2012, 255-263.

[5] Wankhade, K.; Patka, S.; Thool, R. "An Overview of Intrusion Detection Based on Data Mining Techniques"; Proc. of the Communication Syst. and Network Tech. 2013, 626-629.

[6] Duanyang, Z.; Qingxiang, X.; Zhilin, F. "Analysis and Design For Intrusion Detection System Based on Data Mining"; Proc. of the Education Tech. and Computer Sci. 2010, 339-342.

[7] Richards, K. "Network Based Intrusion Detection: A Review of Technologies"; Computers & Security 1999, 18, 671-682.

[8] Gao, X.; Wang, M.; Rongchun, Z. "Applying Fuzzy Data Mining to Network Unsupervised Anomaly Detection"; Proc. of the Communications and Information Tech., IEEE Int. Symposium 2005, 1296-1300.

[9] Park, J.; Shazzad, K.; Kim, D. "Toward Modeling Lightweight Intrusion Detection System Through Correlation-Based Hybrid Feature Selection"; Proc. of Information Security and Cryptology 2005, 279-289.

[10] Sivatha Sindhu, S. S.; Geetha, S.; Kannan, A. "Decision Tree Based Light Weight Intrusion Detection Using a Wrapper Approach"; Expert Systems with Applications 2012, 39, 129-141.

[11] Chen, X. W. "Gene Selection for Cancer Classification Using Bootstrapped Genetic Algorithms and Support Vector Machines"; Proc. of Bioinformatics Conf. 2003, 504-505.

[12] Su, M.-Y. "Real-Time Anomaly Detection Systems for Denial-Of-Service Attacks by Weighted K-Nearest-Neighbor Classifiers"; Expert Systems with Applications 2011, 38, 3492-3498.

[13] Frohlich, H.; Chapelle, O.; Scholkopf, B. "Feature Selection for Support Vector Machines by Means of Genetic Algorithm"; Proc. of the Tools with Artificial Intelligence, 15th IEEE Int. Conf. 2003, 142-148.

[14] Kim, D.; Nguyen, H. N.; Ohn, S. Y.; Park, J. "Fusions of GA and SVM for Anomaly Detection in Intrusion Detection System"; Proc. Advances in Neural Networks 2005, 415-420.

[15] Li,Y.; Wang, J. L.; Tian, Z. H.; Lu, T. B.; Young, C. "Building Lightweight Intrusion Detection System Using Wrapper-Based Feature Selection Mechanisms"; Computers & Security 2009, 28, 466-475.

[16] Lin, S. W.; Ying, K. C.; Lee, C. Y.; Lee, Z. J. "An Intelligent Algorithm with Feature Selection and Decision Rules Applied to Anomaly Intrusion Detection"; Applied Soft Computing 2012, 12, 3285-3290.

[17] Eesa, A. S.; Orman, Z.; Brifcani A. M. A. "A Novel Feature-Selection Approach Based on the Cuttlefish Optimization Algorithm for Intrusion Detection Systems"; Expert Systems with Applications 2015, 42, 2670-2679.

[18] Elshoush, H. T.; Osman, I. M. "Alert Correlation in Collaborative Intelligent Intrusion Detection Systems-A Survey"; Applied Soft Computing 2011, 11, 4349-4365.

[19] Li, Y.; Tian, J. L. Z. H.; Lu T. B.; Young, C. "Building Lightweight Intrusion Detection System Using Wrapper-Based Feature Selection Mechanisms"; Computers & Security 2009, 28, 466-475.

[20] Atashpaz-Gargari, E.; Lucas, C. "Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition"; Proc. of the Evolutionary Computation, IEEE Congress 2007, 4661-4667.

[21] Khorani, V.; Forouzideh, N.; Nasrabadi, A. M. "Artificial Neural Network Weights Optimization Using ICA, GA, ICA-GA and R-ICA-GA: Comparing Performances"; Proc. of the Hybrid Intelligent Models and Applications, IEEE Workshop 2011, 61-67.

[22] Pillai, M. M.; Eloff, J. H. P.; Venter, H. S. "An Approach to Implement a Network Intrusion Detection System Using Genetic Algorithms"; Proc. of the SAICSIT 2004, 221-228.