مجله علمی پژوهشی «پدافند الکترونیکی و سایبری»
سال ششم، شماره ۴، زمستان ۱۳۹۷؛ ص ۱۵۲-۱۴۳

ارایه روشی نوین برای انتخاب ویژگی داده‌های ترافیک شبکه به منظور بهبود عملکرد سامانه‌های تشخیص نفوذ

زهرا جعفرپور[۱]، فرهاد راد[۲*]، حمید پروین[۳]

۱- دانشجوی کارشناسی ارشد، گروه مهندسی کامپیوتر، واحد یاسوج، دانشگاه آزاد اسلامی، یاسوج، ایران

۲- دانشجوی دکتری گروه مهندسی کامپیوتر، واحد یاسوج، دانشگاه آزاد اسلامی، یاسوج، ایران و باشگاه پژوهشگران جوان و نخبگان، واحد یاسوج، دانشگاه آزاد اسلامی، یاسوج، ایران

۳- استادیار گروه مهندسی کامپیوتر، واحد یاسوج، دانشگاه آزاد اسلامی، یاسوج، ایران

(دریافت: ۱۳۹۶/۱۰/۲۴، پذیرش: ۱۳۹۷/۰۳/۰۶)

چکیده

تشخیص نفوذ در فضای سایبری زمینه مهمی برای تحقیقات امروزی در حوزه امنیت شبکه‌های کامپیوتری است. هدف از طراحی و پیاده‌سازی سامانه‌های تشخیص نفوذ، دسته‌بندی دقیق کاربران مجاز، هکرها و نفوذکنندگان به شبکه براساس رفتار طبیعی و غیرطبیعی آنها است. با توجه به افزایش چشمگیر حجم داده‌های رد و بدل شده در فضای سایبری، شناسایی و کاهش ویژگی‌های نامناسب داده‌ها نقش مهمی در افزایش دقت و سرعت سامانه‌های تشخیص نفوذ خواهد داشت. در این مقاله، روشی نوین برای انتخاب ویژگی داده‌های شبکه به نام ادغام ویژگی افزایشی پیشنهاد شده است. روش پیشنهادی، با ادغام سطح به سطح و گام به گام ویژگی‌ها، زیر مجموعه‌ای از ویژگی‌های مناسب را به‌گونه‌ای انتخاب می‌نماید تا در نهایت سامانه تشخیص نفوذ بتواند با دقت و سرعت بیشتری شناسایی نفوذها را انجام دهد. هدف از ارایه روش پیشنهادی، به‌کارگیری آن در سامانه‌های تشخیص نفوذ جهت شناسایی یک اتصال عادی از یک اتصال حمله و خراب‌کارانه به شبکه است. نتایج آزمایش‌های انجام شده بر روی مجموعه داده NSL-KDD نشان داده است که روش پیشنهادی در مقایسه با دیگر روش‌ها، از میان ۴۱ ویژگی موجود در پایگاه مورد بررسی، ۶ ویژگی مهم را انتخاب و تنها با تکیه بر همین شش ویژگی قادر است نفوذ را با دقت بالای ۹۹/۵۸ درصد تشخیص دهد. به عبارت دیگر، روش پیشنهادی به‌ازای هر ۱۰۰۰۰ اتصالی که به شبکه انجام شده است، تنها در شناسایی ۴۲ مورد ناکام مانده و حمله یا عادی بودن ۹۹۵۸ اتصال دیگر را به درستی تشخیص داده است. در پایان، مدت زمان اجرای الگوریتم و درصد دقت روش پیشنهادی در مقایسه با دیگر روش‌ها بررسی و بهبود نتایج به‌دست‌آمده گزارش شده است.

واژه‌های کلیدی: امنیت در فضای سایبری، تشخیص نفوذ، رده‌بندی، انتخاب ویژگی، حمله

۱- مقدمه

توسعه روزافزون اینترنت چه به لحاظ زیرساخت و چه از جنبه نرم‌افزاری، سبب افزایش تعداد کاربران شبکه و نیز برنامه‌های کاربردی آنها شده است. امروزه، بسیاری از خدمات بخش دولتی و خصوصی به‌طور مجازی در بستر اینترنت صورت می‌گیرد. توسعه این فضای مجازی سبب شده است تا تشخیص نفوذ به یکی از موضوعات پراهمیت در حوزه امنیت شبکه‌های کامپیوتری تبدیل شود. سامانه‌های تشخیص نفوذ سعی دارند فعالیت اتصال‌های انجام شده توسط کاربران را در دو مقوله هنجار و ناهنجار دسته‌بندی کنند. در سامانه‌های پیشرفته‌تر گاهی نوع رفتار ناهنجار که حمله نیز نامیده می‌شود، مشخص می‌گردد. هر اتصال در شبکه بر مبنای مجموعه‌ای از ویژگی‌ها توصیف می‌شود که تصمیم‌گیری در مورد هنجار یا ناهنجار بودن آن اتصال به کمک همین ویژگی‌ها قابل بررسی است. در اکثر زمان‌ها، تعداد ویژگی‌ها بسیار زیاد است. از این‌رو، تعیین تاثیر آنها در تشخیص نفوذ به‌صورت دستی کار دشواری خواهد بود. روش‌های کاهش بعد (کاهش ویژگی) این امکان را فراهم می‌سازند که ویژگی‌هایی به صورت خودکار انتخاب و پالایش شوند. به بیان دیگر، کاهش ویژگی، نگاشت تبدیل یک فضای بزرگ ویژگی به فضایی با تعداد ویژگی کمتری است. در یک فضای بزرگ ویژگی، برخی از ویژگی‌ها می‌توانند نویزی، نامناسب یا حتی گمراه‌کننده باشند

*رایانامه نویسنده پاسخگو: Rad@iauyasooj.ac.ir

(این ویژگی‌ها، نه تنها کمکی به توصیف بهتر الگوها نمی‌کنند، بلکه گاهی موجب کاهش دقت شناسایی و تخمین نیز می‌شوند) از این‌رو، تلاش برای کم کردن یا حذف این ویژگی‌ها از اهمیت بالایی برخوردار است.

دقت تشخیص سامانه‌های تشخیص نفوذ، مهم‌ترین شاخص کارایی این سامانه‌هاست. افزایش دقت در تشخیص نفوذ، مانع اثرگذاری حمله‌های بیشتر به سامانه می‌شود. بی‌اثر کردن حمله‌ها نقش تعیین کننده‌ای در کاهش هزینه‌های ناشی از حمله‌ها به منابع با ارزش شبکه خواهند داشت. در این مقاله، تلاش شده است، با ارایه یک رویکرد جدید جهت انتخاب ویژگی‌های پراهمیت، ضمن کاهش فضای ویژگی، دقت تشخیص سامانه‌های تشخیص نفوذ افزایش داده شود. این کار گامی اثربخش در جهت توسعه و بهبود ابزارها و مولفه‌هایی است که در سامانه‌های تشخیص نفوذ به کار گماشته می‌شوند.

در این مقاله، اهداف زیر دنبال شده است:

- اهمیت انتخاب ویژگی در شناسایی و تشخیص نفوذ با توجه به نتایج تجربی سنجیده شده است.
- ارزیابی عملکرد الگوریتم پیشنهادی روی مجموعه داده استاندارد NSL-KDD مورد سنجش قرار گرفته است.
- پارامترهای مهم و تاثیرگذار در روش پیشنهادی، مورد آزمایش و بررسی قرار گرفته است.

برای حصول نتایج درست، فرضیه‌های زیر نیز در نظر گرفته شده است:

الف) روش رده‌بندی در این مقاله، محدود به یک یا چند روش متداول رده‌بندی است و بهبود رده‌بندی جزء اهداف مقاله نیست.

ب) فرض شده است که مجموعه داده‌ها مقادیر از دست رفته را شامل نشده است.

این مقاله از بخش‌های زیر تشکیل شده است: در بخش ۱، مروری بر کارهای گذشته در زمینه شناسایی نفوذ و راه‌حل‌های انتخاب ویژگی در سامانه‌های تشخیص نفوذ آمده است. در بخش ۲، روش نوین پیشنهادی ابتدا ارایه شده و در خصوص نحوه تعیین هر یک از پارامترهای موثر در روش پیشنهادی نکاتی بیان شده است. نتایج پیاده‌سازی روش پیشنهادی و مقایسه آن با روش‌های دیگر در بخش ۳ آمده است. در پایان، نتیجه‌گیری و پیشنهادها برای ادامه کارهای آینده نیز بیان شده است.

۲- مروری بر ادبیات موضوع

سامانه تشخیص نفوذ در شبکه، جهت مانیتور و تحلیل ترافیک شبکه در راستای حفاظت سامانه از تهدیدهای شبکه‌ای مورد استفاده قرار می‌گیرد. این سامانه برای یافتن فعالیت‌های مخرب نظیر حملات رد خدمت، حملاتی که ترافیک شبکه را مانیتور می‌کنند و حملات پویش درگاه تلاش می‌کند [۱]. به صورت کلی دو راه‌کار برای کشف نفوذ در شبکه مورد استفاده قرار می‌گیرد:

الف) تشخیص ناهنجاری: زمانی که رفتار مشاهده شده‌ی کاربر، از یک رفتار مورد انتظار پیروی نکند [۳-۱]. در شبکه تشخیص ناهنجاری، فعالیت‌های نرمال سامانه مانند پهنای‌باند شبکه، درگاه ها، قوانین و ارتباط دستگاه‌ها مورد بررسی قرار می‌گیرد. کشف نفوذ ناهنجاری، به چند دلیل یک مساله مشکل است. اول این‌که کاربرد سامانه و رفتار کاربر دائم تکامل می‌یابند، لذا کشف‌کننده نفوذ نیز باید تکامل یابد. بدون اجازه برای چنین تغییراتی در رفتار، به‌زودی مدیر شبکه در هشدارهای نادرست غوطه‌ور شده و اطمینان خود به سامانه را به سرعت از دست خواهد داد. دومین عامل مهم در رابطه با تشخیص ناهنجاری این است که یک هشدار رفتار غیرنرمال، ممکن است هیچ اطلاعات خاص مفیدی را برای مدیر شبکه فراهم نکند. هشداری مبهم مبنی‌بر اینکه سامانه ممکن است تحت حمله باشد اتخاذ اقدام محکم و سنجیده را مشکل می‌کند. همچنین گاهی ممکن است حمله‌ای شناخته شده، توسط تشخیص ناهنجاری پیدا نشود. این یک مشکل اساسی است.

ب) تشخیص مبتنی بر امضا: زمانی است که رفتار مشاهده شده یک تعمد برای سوء استفاده از منابع کامپیوتری شبکه را نشان دهد. مزایای تشخیص مبتنی بر امضا شامل سادگی، کارایی (به شرطی که تعداد امضاها بیش از اندازه نباشد) است و نیز یک قابلیت فوق‌العاده برای یافتن حملات شناخته شده است. فایده مهم دیگر آن این است که هشداری که منتشر می‌شود ویژه است زیرا امضا، یک الگوی حمله خاص را تطبیق می‌دهد. با یک هشدار ویژه، مدیر شبکه می‌تواند به سرعت مشخص کند که حمله‌ی مشکوک، واقعی است یا یک هشدار نادرست است. اگر واقعی شناسایی شد، مدیر شبکه می‌تواند پاسخ مناسبی را اتخاذ نماید. از دیگر مزایای این رویکرد، قابلیت تولید نتیجه دقیق و کاهش هشدارهای نادرست است. با این وجود، از معایب سامانه تشخیص مبتنی بر امضا این است که فایل امضا باید به‌روز رسانی گردد. همچنین، با زیاد شدن تعداد امضاها کارایی سامانه کاهش می‌یابد. از همه مهم‌تر اینکه، سیستم تنها توانایی کشف حملات شناخته شده را خواهد داشت. کوچکترین تغیر در حملات شناخته‌شده، باعث از دست دادن احتمالی تشخیص حمله به‌وسیله سامانه می‌شود [۶-۴].

از آنجایی‌که هدف اصلی از ارایه این مقاله، ارایه روشی جدید برای انتخاب ویژگی داده‌های ترافیک شبکه است در ادامه این

بخش، به معرفی روش‌های انتخاب ویژگی پرداخته و تعدادی از آنها معرفی شده است.

یک روش انتخاب ویژگی، به‌طور تکراری زیر مجموعه‌هایی را تولید و آنها را مورد ارزیابی قرار می‌دهد. در صورتی‌که شرط معینی برآورده شود، این فرایند متوقف می‌شود. پس از توقف فرایند، ویژگی‌های انتخاب شده برای تخمین تابع استفاده شده و در واقع اعتبار پاسخ روش انتخاب ویژگی مورد سنجش قرار می‌گیرد. روش‌های انتخاب ویژگی به دو دسته اصلی تقسیم شده است: روش‌های پالایشی و روش‌های درهم‌تنیده. وجه تمایز این دو دسته روش، نحوه ارزیابی زیر مجموعه‌های تولید شده است. در روش‌های در هم تنیده، برای ارزیابی زیر مجموعه‌ها، همان معیاری که برای اعتبارسنجی زیر مجموعه نهایی استفاده شده است به‌کار گرفته می‌شود. به عبارت دیگر، هر زیر مجموعه تولیدی به طور مستقیم برای رده‌بندی یا رگرسیون استفاده شده و دقت بدست آمده مبنای انتخاب قرار خواهد گرفت. از سوی دیگر، در روش‌های پالایشی از معیاری به لحاظ محاسباتی کم هزینه برای سنجش شایستگی زیر مجموعه‌ها استفاده شده است. به طور کلی، روش‌های پالایشی سریع هستند اما دقت کمی دارند. در مقابل روش‌های در هم تنیده کندتر، اما از دقت بالاتری برخوردار هستند.

روش RELIEF، یک روش انتخاب ویژگی پالایشی مبتنی بر نمونه است [۷]. شالوده این روش وزن‌دهی به ویژگی‌ها با توجه به توانایی آنها در جداسازی کلاس‌ها از هم است. روش پالایشی دیگری، انتخاب ویژگی مبتنی بر همبستگی است [۸]. ایده محوری در این روش، ارزیابی زیرمجموعه‌های ویژگی برحسب همبستگی‌های دوتایی بین ویژگی‌ها است. به بیان دیگر، ارزش یک ویژگی یا یک مجموعه ویژگی بر پایه همبستگی و افزونگی آن با دیگر ویژگی‌ها مشخص می‌شود. اشکالی که معیار ارزیابی در انتخاب ویژگی مبتنی بر همبستگی دارد، بررسی همبستگی دوگانه بین ویژگی‌ها و چشم‌پوشی از همبستگی‌های چندگانه است. در [۱۰] معیار بهبود یافته‌ای برای ارزیابی زیر مجموعه‌ها پیشنهاد شده است. این پژوهش از کارهای انجام شده در خوشه‌بندی ابرگراف و نظریه اطلاعات الهام گرفته شده است. معیار پیشنهادی، تعامل اطلاعات چند بعدی نام دارد که به‌طور همزمان ارزش اطلاعات چند ویژگی را برای تخمین ویژگی تصمیم مشخص می‌کند. در کاری مشابه [۱۱]، روش دیگری برای وزن‌دهی همزمان به‌ویژگی‌های منفرد و همچنین گروه‌های ویژگی پیشنهاد شده است. نویسندگان مقاله یک روش خوشه‌بندی به نام FG-k-Means پیشنهاد کرده‌اند که با توجه به اهمیت تکی و گروهی ویژگی، عمل خوشه‌بندی را انجام داده است. نویسندگان مدعی شده‌اند که این روش در فضاهایی با ابعاد زیاد، بهتر از روش‌های مشابه عمل کرده و همچنین نسبت به مقدارهای از دست رفته نیز مقاوم است.

در روش‌های در هم تنیده، معیار ارزیابی مشخص است و تمرکز اصلی بر روی روش جستجو است. در [۹] یک روش ترکیبی دو مرحله‌ای برای انتخاب ویژگی پیشنهاد شده است. در این روش، برای سرعت بخشیدن به فرآیند انتخاب در هر گام، نخست معیار ارزیابی پالایشی روی ویژگی‌ها اعمال شده است و در ادامه معیار ارزیابی در هم تنیده به صورت بسیار گزیده‌تر برای ارزیابی زیر مجموعه‌های با کیفیت‌تر استفاده شده است.

انتخاب زیر مجموعه در هم تنیده افزایشی یا به طور خلاصه IWSS رهیافتی ساده و موثر برای انتخاب ویژگی در فضاهایی با بعدهای زیاد است. این رهیافت با زیر مجموعه تهی $S = \phi$ آغاز شده و به صورت گام به گام، افزودن ویژگی‌های جدید به زیر مجموعه را بررسی می‌کند [۱۲-۱۱]. پیش از آغاز جستجو، نخست ویژگی‌ها یک به یک بررسی شده و فهرست مرتبی از آنها ساخته می‌شود. در این فهرست، مهم‌ترین ویژگی بعنوان نخستین عنصر فهرست و کم اهمیت‌ترین ویژگی به‌عنوان آخرین عنصر فهرست جای‌گذاری شده است. در ادامه، جستجو آغاز و در هر گام افزودن یکی از ویژگی‌ها به زیر مجموعه بررسی می‌شود. اگر زیر مجموعه $S \cup A_i$ از زیر مجموعه S بهتر باشد، ویژگی جدید به زیر مجموعه اضافه می‌شود. تعداد گام‌های جستجو در این رهیافت برابر با تعداد ویژگی‌ها و برابر n است. ارزیابی زیر مجموعه‌ها با یک معیار درهم‌تنیده و به کمک درخت تصمیم، شبکه عصبی مصنوعی و یا هر رده‌بند دیگری انجام می‌گیرد. در این روش، معیار درهم‌تنیده n بار استفاده می‌شود. اشکال این روش در این است که همه‌ی ویژگی‌های مجموعه داده بررسی می‌شوند. این در حالی است که معمولا بسیاری از این ویژگی‌ها اهمیت بسیار ناچیزی داشته و می‌توان از آنها صرف‌نظر کرد.

در روش جستجو در بهترین بلوک k تایی، پس از مرتب‌سازی ویژگی‌ها بر پایه اهمیت آنها، بلوکی از k ویژگی برتر جدا شده و دیگر ویژگی‌ها حذف می‌شوند. در گام دوم، جستجو آغاز شده و هر بار یکی از k ویژگی برای افزوده شدن به زیر مجموعه انتخابی مورد بررسی قرار می‌گیرد. تفاوت دیگری که این روش نسبت به روش IWSS دارد، بررسی چند باره ویژگی‌ها است. در IWSS هر ویژگی تنها یک بار مورد بررسی قرار می‌گیرد در حالی‌که در روش بهترین بلوک k تایی پس از افزودن ویژگی، ویژگی‌های افزوده نشده از میان A_1 تا A_{i-1} مجددا مورد بررسی قرار خواهند گرفت. بررسی چند باره ویژگی‌ها به این دلیل انجام

شده است که مفید بودن افزودن یک ویژگی به یک زیر مجموعه به اعضای آن زیر مجموعه وابسته است. به بیان دیگر، گاهی افزودن ویژگی به زیر مجموعه در ابتدا مفید نیست، اما پس از افزودن چند ویژگی دیگر مفید خواهد شد. در این رهیافت، معیار در هم تنیده در بدترین حالت می‌تواند $k \times (k+1)/2$ مرتبه به‌کار گرفته شود.

انتخاب ویژگی ممکن است به عنوان یک مساله بهینه‌سازی نیز فرمول‌بندی شود. در این دیدگاه انتخاب ویژگی، یک روش برای یافتن یک زیر مجموعه از ویژگی‌های شرطی با بیشترین ارتباط با ویژگی هدف است. GRASP یک الگوریتم بهینه‌سازی دو فازی است که در [۱۳] معرفی شده است. GRASP یک الگوریتم تکراری است که در هر تکرار، بعد از تکمیل فازها، راه‌حل بهبود یافته و ساخته شده به راه‌حل‌های غیر مسلط افزوده می‌شوند (مجموعه راه‌حل‌های غیر مسلط، یک مجموعه از راه‌حل‌هایی است که هیچ موردی از آن به دیگری برتری کاملی ندارد).

FCGRASP در [۱۲] پیشنهاد شده است. در این کار، یک روش جستجوی جدید برای انتخاب ویژگی در داده‌های ابعاد بالا پیشنهاد شده است که تعداد ارزیابی‌های پوشاننده را به‌وسیله تعویض تناوبی بین ارزیابی‌های فیلتر و پوشاننده کاهش داده است. روش پیشنهادی در [۱۲]، بر اساس یک الگوریتمی فرا اکتشافی عمل می‌کند. این الگوریتم یک الگوریتم تکراری دو مرحله‌ای است که در هر تکرار یک راه‌حل ساخته شده و سپس بهبود می‌یابد.

روش پیشنهاد شده در بخش بعدی، در گروه روش‌های انتخاب ویژگی در هم تنیده طبقه‌بندی شده است که با انتخاب ویژگی های مناسب، مصالحه‌ای درست بین دقت و زمان اجرای الگوریتم را نیز به‌همراه داشته است. روش پیشنهادی ضمن داشتن سرعت مناسب، دقت بالایی در تشخیص حملات را نیز داشته است.

۳- روش پیشنهادی

در این بخش، برای افزایش دقت سامانه‌های تشخیص نفود، روش جدیدی برای انتخاب ویژگی پیشنهاد شده است. این روش، ادغام ویژگی افزایشی نامیده شده است. ایده روش پیشنهادی تا حدودی الهام گرفته از الگوریتم آپریوری است که برای استخراج قوانین وابستگی از آن استفاده شده است. گام‌های روش پیشنهادی به صورت زیر است:

۱) در آغاز همه ویژگی‌ها به صورت تک تک به مجموعه زیر مجموعه‌های ویژگی با اندازه یک افزوده می‌شوند. ارزش هر ویژگی برابر با دقتی است که به‌کارگیری آن ویژگی در تشخیص نفوذ بدست آمده است.

۲) سپس به طور تکراری در هر مرحله، هر یک از ویژگی‌ها با هر یک از زیر مجموعه‌های ویژگی با اندازه i ادغام شده و چنانچه بهبود دقت حاصل بیشتر از پارامتر *MinInc* شود و نیز فاصله دقت آن از بهترین پاسخ موجود بیشتر از پارامتر *DistFromBest* نباشد، به زیر مجموعه‌ی زیر مجموعه‌های ویژگی با اندازه $i+1$ اضافه می‌شود. پس از ایجاد کلیه زیر مجموعه‌ها با اندازه $i+1$، اگر تعداد زیر مجموعه‌های ویژگی با اندازه $i+1$ بیشتر از پارامتر *MaxRemainingSubsets* شود، بهترین *MaxRemainingSubsets* حفظ شده و بقیه زیر مجوعه‌ها حذف می‌شوند.

۳) بهترین زیر مجموعه ویژگی به عنوان راه‌حل نهایی در نظر گرفته می‌شود.

شبه‌کد روش پیشنهادی در شکل (۱) نمایش داده شده است. روش پیشنهادی دارای پنج ورودی است. ماتریس X، ماتریسی با n سطر و f ستون است که n مشخص کننده تعداد نمونه‌ها یا همان اتصال‌ها است و f مشخص کننده تعداد ویژگی‌هایی است که بر پایه‌ی آنها در مورد هنجار یا حمله بودن اتصال داوری شده است. ماتریس Y به عنوان دومین ورودی دارای n سطر و تنها یک ستون است و در آن نوع هر اتصال مشخص شده است. سومین ورودی پارامتر *MinInc* است که مشخص می‌کند کمترین میزان افزایش دقت قابل قبول با افزوده شدن یک ویژگی جدید چقدر است. چهارمین ورودی، دومین پارامتر روش پیشنهادی یعنی *DistanceFromBest* است که مشخص می‌کند کدام زیر مجموعه‌ها می‌توانند در ادامه اجرای روش حفظ شوند. شرط باقی ماندن یک زیر مجموعه جدید آن است که اختلاف دقت به‌دست‌آمده برای آن زیر مجموعه بیشتر از *DistanceFromBest* نباشد. پنجمین ورودی نیز سومین پارامتر روش پیشنهادی یا همان *MaxRemainingSubsets* است که مشخص می‌کند بیشترین تعداد زیر مجموعه با یک اندازه مشخص چقدر است.

در خطوط ۳ تا ۸ شبه‌کد الگوریتم پیشنهادی، همه زیر مجموعه‌ها با اندازه یک به مجموعه *Subset*(1) افزوده می‌شوند. به عبارت دیگر، هر یک از ویژگی‌ها به عنوان یک زیر مجموعه در نظر گرفته شده و همراه با دقت به‌دست‌آمده از به‌کارگیری آن ویژگی در *Subset*(1) ثبت شده است. در خطوط ۹ تا ۲۸، روال تکراری الگوریتم در قالب یک حلقه، زیر مجموعه‌هایی با اندازه بزرگ‌تر از یک را به صورت گام به‌گام می‌سازد. خط ۱۰ شبه‌کد، شرط خروج زود هنگام از روال تکرار را نمایندگی می‌کند. این شرط دلالت بر آن دارد که چنانچه هیچ زیر مجموعه‌ای با اندازه $i-1$ وجود نداشته باشد، نمی‌توان به ساختن زیر مجموعه‌هایی با اندازه i یا بزرگتر از آن پرداخت. در خطوط ۱۲ تا ۳۱، همه مجموعه‌های ممکن حاصل از اضافه کردن هر یک از ویژگی‌های

موجود به هر یک از زیر مجموعه‌های موجود با اندازه $i-1$ تولید می‌شود. چنانچه زیر مجموعه تولید شده با اندازه‌ی i دقت بیشتری حداقل به اندازه *MinInc* نسبت به زیر مجموعه قبلی که اندازه $i-1$ داشته باشد و همچنین فاصله آن با بهترین جواب موجود بیشتر از *DistanceFromBest* نباشد به زیر مجموعه‌های با اندازه i اضافه می‌شود. در انتهای هر تکرار، چنانچه تعداد زیر مجموعه‌هایی با اندازه i بیشتر از *MaxRemainingSubsets* شود، تنها بهترین زیر مجموعه‌ها تا سقف *MaxRemainingSubsets* زیر مجموعه، حفظ شده و بقیه حذف می‌شوند.

Function **IncrementalFeatureJoining**

inputs:
- **X:** a n*f matrix of values (n samples with f features)
- **Y:** a n*1 matrix of target values (labels of n samples)
- **MinInc:** first parameter of the proposed method
- **DistanceFromBest:** second parameter of the proposed method
- **MaxRemainingSubsets:** third parameter of the proposed method

Outputs:
- **BestSubset:** the output of proposed method

```
1     BestPrecision = 0;
2     BestSubset = [];

3     for a= 1: NumOfFeatures
4       FeatureSubset = {a} ;
5       XX = X(FeatureSubset);
6       Precision = EvaluateSubset(XX, Y);
7      Add (FeatureSubset, Precision) to Subsets (1)
8     end

9     for i=2: NumOfFeatures
10      if (isempty(Subsets(i-1)))
11        break;

12      for a=1: NumberOfSubsets in Subsets(i-1)
13    CurrentSubset = Subsets (i-1, a). FeatureSubset;
14     CurrentPrecision = Subsets (i-1, a). Precision;
15     for b=1: NumOfFeatures
16          NewSubset = [CurrentSubsets, b];
17          XX = X(NewSubset);
18          Precision = EvaluateSubset(XX, Y);

19          if (Precision>BestPrecision)
20            BestPrecision = Precision;
21            BestSubset = NewSubset;
22          end

23    if (Precision >= CurrentPrecision + MinInc &&
      Precision >= BestPrecision - DistanceFromBest)
24     Add (NewSubset, Precision) to Subsets(i);
25    end
26    end
27    end
28      if(|Subsets(i)|)
29        Keep best MaxRemainingSubsets subsets
      and ignore others
30      end
31  end
```

شکل(۱): شبه‌کد روش پیشنهادی (ادغام ویژگی افزایشی)

به‌طور خلاصه می‌توان گفت که برای ایجاد زیر مجموعه‌هایی با اندازه بیشتر باید به هر یک از زیر مجموعه‌های موجود، ویژگی‌های مختلف را اضافه کرد. (به بیان دیگر، گاهی افزودن ویژگی به زیر مجموعه در ابتدا مفید نیست، اما پس از افزودن چند ویژگی دیگر مفید خواهد شد.) این کار با اضافه کردن یک به یک ویژگی‌ها به زیر مجموعه ابتدایی آغاز می‌شود. با توجه به مقدار پارامتر *MinInc* (یکی از پارامترهای مهم در الگوریتم پیشنهادی)، افزودن یک ویژگی حداقل باید ارتقای دقت به اندازه این پارامتر را داشته باشد در غیراینصورت این ویژگی حذف خواهد شد.

نکته بسیار مهم در مورد الگوریتم پیشنهادی و آپریوری این است که اگر چه شباهت قابل توجهی بین آنها وجود دارد، اما این دو الگوریتم را نمی‌توان با یکدیگر مقایسه کرد زیرا برای دو کاربرد مختلف پیاده‌سازی شده‌اند. از آپریوری برای استخراج قوانین تداعی در داده‌کاوی و از روش پیشنهادی برای انتخاب ویژگی استفاده شده است. نسخه‌ای از الگوریتم آپریوری برای انتخاب ویژگی وجود ندارد که با روش پیشنهادی مقایسه شود. وجه مشترک بین دو الگوریتم شیوه جستجو به دنبال جواب است به نحوی که هر دو سعی می‌کنند دسته‌ای از راه‌حل‌های احتمالی را به صورت گام به گام بهبود داده و این کار را با کنترل هوشمندانه تعداد راه‌حل‌ها انجام می‌دهند. هر دو الگوریتم روی راه‌حل‌هایی تمرکز می‌کنند که احتمال رسیدن به جواب در آنها بسیار زیاد است.

مقادیر پارامترهای کنترلی الگوریتم پیشنهادی، بر پایه نتایج متعدد تجربی، در جهت دستیابی به حداکثر کارایی و بهترین مصالحه بین سرعت و دقت تنظیم شده است. از این‌رو، نتایج آزمایش‌ها نشان داده است که هر چقدر مقدار *MinInc* کمتر و مقادیر *DistanceFromBest* و *MaxRemainingSubsets* بیشتر باشد (سه پارامتر کنترلی تاثیرگذار بر الگوریتم پیشنهادی)، روش پیشنهادی جستجوی وسیع‌تری را انجام داده و تعداد زیر مجموعه‌های بیشتری را مورد بررسی قرار خواهد داد. از این‌رو، شانس پیدا کردن جواب بهینه افزایش خواهد یافت.

در ادامه نحوه کار روش پیشنهادی را با ذکر یک مثال نشان خواهیم داد.

۳-۱- مثالی از نحوه کار روش پیشنهادی برای استخراج ویژگی

یک مجموعه داده با هفت ویژگی شامل ویژگی‌های F_1 تا F_7 را در نظر بگیرید.

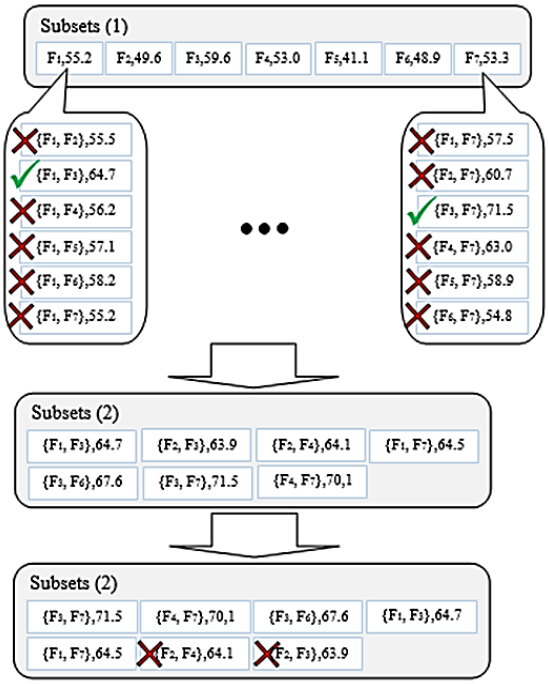
پارامترهای الگوریتم	همه ویژگی‌ها
MinINC: 0.5 *DistanceFromBest:2* *MaxRemainingSubsets :5*	$\{F_1, F_2, F_3, F_4, F_5, F_6, F_7\}$

با آغاز الگوریتم، هر یک از این ویژگی‌ها به صورت یک زیر مجموعه از ویژگی با اندازه یک در نظر گرفته شده است و دقت حاصل از به کارگیری آن ویژگی محاسبه شده است.

Subset(1):

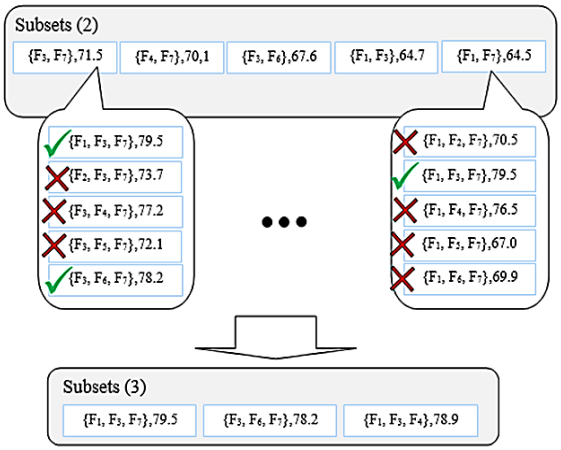
F1, 55.2 -- F2, 49.6 --F3, 59.6 --F4, 53.0 --F5, 41.1 --F6, 48.9 --F7, 53.3

اکنون *Subset(1)* شامل هفت زیر مجموعه با اندازه یک است. در ادامه می‌توان وارد روال تکراری الگوریتم شد و اقدام به ایجاد زیر مجموعه‌هایی با اندازه بیشتر کرد. در این مرحله از اجرای الگوریتم، مقدار *BestPrecision* برابر با ۵۹/۶ است. برای ایجاد زیر مجموعه‌هایی با اندازه دو نیز باید به هر یک از زیر مجموعه‌های موجود، ویژگی‌های مختلف را اضافه نمود. این کار با اضافه نمودن تک تک ویژگی‌ها به زیر مجموعه $\{F_1\}$ آغاز می‌شود. با توجه به اینکه مقدار پارامتر *MinInc* برابر با ۰/۵ است، افزودن یک ویژگی حداقل باید ارتقای دقت به میزان ۰/۵ را داشته باشد. با افزودن F_2 به $\{F_1\}$ زیر مجموعه $\{F_1, F_2\}$ ایجاد شده و دقت از ۵۵/۲ به ۵۵/۵ می‌رسد. میزان افزایش دقت در این حالت ۰/۳ است، بنابراین زیر مجموعه $\{F_1, F_2\}$ کنار گذاشته می‌شود. سپس F_3 به $\{F_1\}$ افزوده شده و زیر مجموعه $\{F_1, F_3\}$ با دقت ۶۴/۷ بدست می‌آید. میزان افزایش دقت بیش از ۰/۵ است و همچنین عدد ۶۴/۷ کمتر از *BestPrecision-DistanceFromBest* نیست (۵۳/۵=۲-۵۵/۵) بنابراین، این زیر مجموعه قابل قبول است و به *Subset(2)* اضافه می‌شود. مقدار *BestPrecision* هم به‌روزرسانی شده است. این کار به همین ترتیب ادامه پیدا می‌کند. هفت زیر مجموعه با اندازه یک در مجموعه *Subset(1)* وجود دارد که به هر کدام از شش ویژگی می‌تواند افزوده شود. به این ترتیب، از هفت زیر مجموعه موجود در *Subset(1)* می‌توان به ۴۲ زیر مجموعه با اندازه دو رسید که تنها بعضی از آنها قابل قبول است. همانطور که در شکل (۲) نشان داده شده است، از مجموعه ۴۲ زیر مجموعه، تنها هفت مجموعه قابل قبول است. با توجه به این‌که پارامتر *MaxRemainingSubsets* برابر با ۵ در نظر گرفته شده است، پنج زیر مجموعه‌ای که دارای بیشترین دقت هستند حفظ شده و دو زیر مجموعه دیگر حذف می‌شوند.



شکل(۲): نخستین تکرار روش پیشنهادی برای مثال

اکنون با پنج زیر مجموعه‌ای که در *Subset(2)* وجود دارد، می‌توان به ساخت زیر مجموعه‌هایی با اندازه سه پرداخت. شکل (۳) نشان می‌دهد، چگونه زیر مجموعه‌های *Subset(3)* بر پایه *Subset(2)* ساخته شده است. در این تکرار، می‌توان به پنج زیر مجموعه با انداز دو، به پنج شکل مختلف ویژگی افزود تا ۲۵ زیر مجموعه با اندازه سه به‌دست آید. همانطور که در شکل (۳) نشان داده شده است، فقط سه زیر مجموعه شرایط لازم برای اضافه شدن به *Subset(3)* را خواهند داشت. این روال همچنان ادامه پیدا می‌کند تا این‌که به زیر مجموعه‌ای با حداکثر ممکن یعنی هفت برسیم یا این‌که در مرحله i، تعداد زیر مجموعه‌های عضو *Subset(i-1)* برابر با صفر شود.



شکل(۳): دومین تکرار روش پیشنهادی در مثال

۴- نتایج آزمایش‌ها

در این بخش، توصیف مجموعه داده ترافیک شبکه مورد استفاده، پیاده‌سازی الگوریتم پیشنهادی و بررسی نتایج بیان شده است. پیاده‌سازی شامل همه مراحل از پیش‌پردازش داده تا آزمایش رده‌بند را شامل شده است. مجموعه داده KDD یک مجموعه داده مناسب برای بررسی مجموعه‌ای از مسایل در حوزه سامانه‌های کشف نفوذ است. یکی از مهمترین مشکلات مجموعه داده KDD، تعداد بسیار زیاد نمونه‌های تکراری آن است. در جدول‌های (۲-۱) نشان داده شده است که میزان تکراری بودن رکوردها در دو مجموعه آموزشی و آزمایشی KDD تا چه حد بالاست. در این مقاله، از مجموعه داده‌ی ترافیک شبکه بهبود یافته KDD به نام NSL-KDD [۱۴] استفاده شده است. این مجموعه داده چند مزیت مهم نسبت به مجموعه داده اصلی دارد که عبارتند از:

- رکوردهای تکراری در مجموعه آموزشی حذف شده است. این ویژگی مانع از آن شده است که رده‌بندها به رکوردهای با فراوانی بالا گرایش پیدا کنند.
- رکوردهای تکراری در مجموعه داده آزمایشی نیز حذف شده‌اند. بنابراین، دقت رده‌بندها به طور واقعی سنجیده شده است. این امر باعث شده است که رده‌بندهایی با دقت بیشتر روی نمونه‌های تکراری کارآمدتر به نظر نرسند.
- تعداد رکوردها در هر دو مجموعه آموزشی و آزمایشی به حد معقولی رسیده است. از این رو، می‌توان از آنها به طور مستقیم برای آموزش و آزمایش رده‌بندها استفاده کرد. در مجموعه داده قبلی، زیاد بودن تعداد نمونه‌ها ایجاب می‌کرد که پیش از به‌کارگیری مجموعه داده‌ها نمونه‌گیری انجام شده تا تعداد رکوردها به اندازه کافی کاهش یابد.

همان‌طور که در جدول (۳) نشان داده شده است در مجموعه داده NSL-KDD پس از ویرایش مجموعه داده و حذف رکوردهای تکراری، تعداد رکوردهای مجموعه داده آموزش و آزمایش به ترتیب به ۱۲۵۹۷۳ رکورد و ۲۲۵۴۴ رکورد رسید. هر رکورد شامل ۴۱ ویژگی است که یک اتصال را توصیف می‌کنند و یک برچسب دارد که نوع حمله را مشخص می‌نماید. در مجموع اتصال‌ها به ۴۰ دسته مختلف تقسیم می‌شوند که شامل اتصال‌های عادی و ۳۹ نوع اتصال از حمله‌های گوناگون است که در جدول (۴) آمده است.

جدول (۱): آمار رکوردهای افزونه در مجموعه داده آموزشی KDD

تعداد داده‌ها	Original Records	Distinct Records	Reduction Rate
حمله	۳۹۲۵۶۵۰	۲۶۲۱۷۸	٪۹۳/۳۲
نرمال	۹۷۲۷۸۱	۸۱۲۸۱۴	٪۱۶/۴۴
مجموع	۴۸۹۸۴۳۱	۱۰۷۴۹۹۲	٪۷۸/۰۵

جدول (۲): آمار رکوردهای افزونه در مجموعه داده آزمایشی KDD

تعداد داده‌ها	Original Records	Distinct Records	Reduction Rate
حمله	۲۵۰۴۳۶	۲۹۳۷۸	٪۸۸/۲۶
نرمال	۶۰۵۹۱	۴۷۹۱۱	٪۲۰/۹۲
مجموع	۳۱۱۰۲۷	۷۷۲۸۹	٪۷۵/۱۵

جدول (۳): اطلاعات مجموعه داده NSL-KDD

تعداد داده‌ها	مجموعه داده آموزشی	مجموعه داده آزمایشی
حمله	۵۸۶۳۰	۹۷۱۱
نرمال	۶۷۳۴۳	۱۲۸۳۳
مجموع	۱۲۵۹۷۳	۲۲۵۴۴

رده‌بندهای مورد استفاده در پیاده‌سازی انجام شده، الگوریتم‌های درخت تصمیم و نزدیک‌ترین K همسایه است. برای استفاده از رده‌بند نزدیک‌ترین همسایه باید همه ویژگی‌ها از نوع عددی باشند. در مجموعه داده NSL-KDD چهار ویژگی از نوع غیرعددی است که شامل برچسب نیز می‌شود. تبدیل ویژگی‌های غیرعددی به عددی مستلزم آن است که ابتدا همه مقادیر غیرعددی لیست شده و به هر کدام یک عدد نسبت داده شود. پس از نسبت دادن یک عدد به هر مقدار غیرعددی یکتا، می‌توان مقادیر عددی را جایگزین نمود. الگوریتم جدول (۵) نحوه انجام این مرحله را نشان داده است. ورودی تابع یک ستون از ویژگی‌های غیرعددی به نام *SymbolicColumn* است. در آغاز، خروجی تابع یعنی *NumericalColumn* را برابر با *SymbolicColumn* است. سپس به‌طور تکراری در یک حلقه هر بار یک برچسب عددی جایگزین یکی از نمادهای غیرعددی می‌شود.

جدول (۴): انواع حمله در مجموعه دادهNSL-KDD

نوع حمله	برچسب	نوع حمله	برچسب	نوع حمله	برچسب	نوع حمله	برچسب
normal	1	Back	11	loadmodule	21	snmpguess	31
neptune	2	guess_passwd	12	Spy	22	mailbomb	32
warezclient	3	ftp_write	13	Perl	23	named	33
ipsweep	4	multihop	14	Saint	24	sendmail	34
portsweep	5	rootkit	15	Mscan	25	xterm	35
teardrop	6	buffer_overflow	16	apache2	26	worm	36
nmap	7	imap	17	snmpgetattack	27	xlock	37
satan	8	warezmaster	18	processtable	28	xsnoop	38
smurf	9	Phf	19	httptunnel	29	sqlattack	39
pod	10	Land	20	Ps	30	Udpstorm	40

جدول(۵): الگوریتم تبدیل یک‌ستون از ویژگی‌های غیرعددی به عددی

```
NumericalColumn SymolicToNumerical (SymbolicColumn)

Current Label = 1;
NumericalColumn = SymbolicColumn

While there are some Symbolic values in NumericalColumn Do
Symbolic Value = find first symbolic value in NumericalColumn
Change all positions of NumericalColumn with value of
Symbolic Value by Current Label
Current Label = Current Label + 1
End
Return NumericalColumn
```

نتایج اجرای الگوریتم‌ها برای روش نزدیک‌ترین K همسایه به‌ازای پنج اجرای مستقل از هم گزارش شده است. در ادمه، نتایج پیاده‌سازی و ارزیابی روش پیشنهادی و سه روش شناخته شده دیگر بر روی مجموعه داده‌های NSL-KDD گزارش شده است. این‌سه روش عبارتند از: روش انتخاب زیر مجموعه در هم تنیده افزایشی، روش جستجو در بهترین بلوک Kتایی و روش GRASP برای انتخاب ویژگی. با توجه به این‌که روش پیشنهادی و سه روش رقیب در دسته‌ی روش‌های در هم تنیده به حساب می‌آیند، باید از یک رده‌بند مناسب برای ارزیابی زیر مجموعه ویژگی‌های انتخابی بهره گرفت. نتایج به‌ازای دو رده‌بند نزدیک‌ترین K همسایه و درخت تصمیم گزارش شده است. دقت روش پیشنهادی و سه روش دیگر در جدول (۶) نشان داده شده است.

جدول (۶): مقایسه دقت روش پیشنهادی با سه روش رقیب

		روش انتخاب ویژگی			
		ادغام ویژگی افزایشی (روش پیشنهادی)	انتخاب زیر مجموعه درهم‌تنیده افزایشی	جستجو در بهترین بلوک Kتایی	GRASP برای انتخاب ویژگی
رده‌بند	درخت تصمیم	**۹۹/۵۸**	۹۸/۹۵	۹۸/۸۹	۹۹/۲۸
	نزدیک‌ترین k همسایه	**۹۹/۳۷**	۹۸/۶۹	۹۸/۷۴	۹۹/۱۵

دقت گزارش شده برای روش پیشنهادی، کمترین مقداری است که به ازای ترکیب‌های مختلف پارامترهای ورودی به‌دست آمده است. نتایج نشان داده است که دقت روش پیشنهادی قابل قبول است و این تنها در شرایطی به‌دست آمده است که دقت روش پیشنهادی به ازای بدترین تنظیم پارامترها گزارش شده است (به این جهت، سطر دوم جدول (۸) انتخاب شده است که در آن کمترین تعداد ویژگی انتخابی مشخص شده است. با این‌حال همچنان روش پیشنهادی از دقت بالایی برخوردار است). نتایج همچنین نشان داده است که نزدیک‌ترین رقیب به‌روش پیشنهادی، روش GRASP برای انتخاب ویژگی است. همچنین عملکرد رده‌بندها روی مجموعه داده مورد استفاده نشان داده است. نتایج نشان داده است که در کاربرد تشخیص نفوذ، رده‌بند درخت تصمیم مناسب‌تر از رده‌بند نزدیک‌ترین k همسایه است. ویژگی‌های انتخاب شده در هر یک از چهار روش در جدول (۷) نشان داده شده است. نتایج این جدول برتری روش پیشنهادی را نیز مشخص کرده است. روش پیشنهادی با انتخاب تنها شش ویژگی به دقت ۹۹/۵۸ رسیده است. این در حالی است که نزدیک‌ترین رقیب آن یعنی روش GRASP برای انتخاب ویژگی با یازده ویژگی به دقت ۹۹/۲۸رسیده است.

همان‌طور که در بخش ۳ بیان شده است، روش پیشنهادی یا همان روش ادغام ویژگی افزایشی، دارای سه پارامتر کلیدی است که روی ویژگی‌هایی که توسط روش پیشنهادی انتخاب

شده‌اند، تاثیر به‌سزایی داشته است. این پارامترها عبارتند از *MinInc*، *DistanceFromBest* و *MaxRemainingSubsets* که در بخش قبل معرفی شده است. نتایجی که در جدول (۸) نشان داده شد، مشخص کرده است که هر چقدر مقدار *MinInc* کمتر و مقادیر *DistanceFromBest* و *MaxRemainingSubsets* بیشتر باشند، روش پیشنهادی می‌تواند جستجوی وسیع‌تری را انجام داده و تعداد زیر مجموعه‌های بیشتری را مورد ارزیابی قرار دهد. این فرایند، احتمال بیشتری برای رسیدن به جواب بهینه را به همراه خواهد داشت.

جدول (۷): ویژگی‌های انتخاب روش پیشنهادی و سه روش رقیب

روش	ویژگی‌ها
ادغام ویژگی افزایشی **(روش پیشنهادی)**	۳۷، ۳۰، ۲۷، ۵، ۳، ۲
انتخاب زیر مجموعه درهم‌تنیده افزایشی	۳۸، ۳۷، ۳۰، ۲۹، ۲۸، ۲۶، ۱۳، ۵، ۴، ۳، ۲
جستجو در بهترین بلوک تایی k	۳۹، ۳۷، ۳۶، ۳۵، ۳۴، ۲۷، ۱۳، ۱۰، ۵، ۲
GRASP برای انتخاب ویژگی	۳۹، ۳۷، ۳۰، ۲۹، ۲۶، ۲۵، ۲۳، ۵، ۴، ۳، ۲

نتایج الگوریتم پیشنهادی در اجرای شماره نهم یعنی سطر آخر جدول (۸) به‌وضوح این مساله را نشان داده است. در این اجرا مقدار *MinInc* برابر با ۰/۰۱ و مقادیر *DistanceFromBest* و *MaxRemainingSubsets* به ترتیب برابر با ۱۰ و ۱۰۰۰۰۰ است. نتایج جدول (۸) نشان داده است که دقت در اجرای مرحله نهم روش پیشنهادی، توانسته است به مقدار ۹۹/۹۲ درصد برسد. این نتیجه برتری مطلق روش پیشنهادی را نسبت به سایر اجراها نشان داده است. البته باید توجه داشت که مصالحه بین دقت و سرعت اقتضا می‌کند که به ازای افزایش دقت، سرعت اجرای الگوریتم کاهش یابد. از همین‌رو به زمان بیشتری جهت اجرا نیاز است.

از آنجا که هدف از ارایه الگوریتم پیشنهادی، پیدا کردن بهترین حالت جواب با توجه به، الف) کمترین تعداد ویژگی، ب) دقت مناسب و ج) زمان اجرای کمتر بوده است لذا بین تعداد اجراهای انجام شده روی مجموعه داده مورد آزمایش، اجرای دوم با شرایط مورد اشاره توانسته است هر سه شرط لازم برای مساله را برطرف نماید. از این‌رو، این مرحله به عنوان راه‌حل قابل قبول و نهایی انتخاب شده است.

جدول (۸): نتایج ۹ اجرای روش پیشنهادی با تنظیم پارامترهای مختلف

شماره اجرا	*MinInc*	*DistanceFromBest*	*MaxRemainingSubsets*	دقت	ویژگی‌های انتخاب شده	زمان اجرا (ثانیه)
۱	۰/۱	۳	۵۰	۹۹/۶۷	۳۷، ۳۶، ۳۵، ۲۸، ۲۶، ۶، ۵	۵۶۲
۲	۰/۱	۰/۵	۵۰	۹۹/۵۸	۳۷، ۳۰، ۲۷، ۵، ۳، ۲	۲۴۱
۳	۰/۰۱	۳	۵۰	۹۹/۶۷	۳۹، ۳۷، ۳۶، ۳۵، ۳۲، ۲۷، ۲۱، ۱۰، ۵، ۲	۱۴۱۱
۴	۰/۱	۳	۱۰۰	۹۹/۵۸	۳۷، ۳۶، ۳۵، ۲۷، ۱۱، ۶، ۵	۱۱۶۱
۵	۰/۱	۳	۲۰	۹۹/۵۸	۴۰، ۳۷، ۳۴، ۳۰، ۲۷، ۵، ۲، ۱	۱۰۹۷
۶	۰/۱	۳	۱۰	۹۹/۶۷	۳۷، ۳۱، ۳۰، ۲۵، ۷، ۶، ۵، ۴، ۳	۲۸۶
۷	۰/۱	۳	۱۰۰۰	۹۹/۶۷	۳۹، ۳۷، ۳۶، ۳۵، ۲۷، ۶، ۵	۱۰۶۴
۸	۰/۱	۳	۱۰۰۰۰	۹۹/۶۷	۳۷، ۳۵، ۳۴، ۲۳، ۲۱، ۵، ۲	۹۲۰
۹	۰/۰۱	۱۰	۱۰۰۰۰۰	۹۹/۹۲	۳۷، ۳۵، ۳۴، ۲۳، ۱۸، ۱۰، ۵، ۲، ۱	۸۶۷۷

۵- نتیجه‌گیری

در این مقاله، روش نوینی برای انتخاب ویژگی پیشنهاد شده است که در آن با ادغام سطح به سطح و گام به گام ویژگی‌ها سعی شده است تا جواب نهایی بهبود یابد. در روش پیشنهادی زیر مجموعه‌ای از ویژگی‌ها به‌گونه‌ای انتخاب شده است که بتوان به دقت بیشتری نیز دست یافت. از روش پیشنهادی برای انتخاب ویژگی مجموعه داده ترافیک شبکه جهت تشخیص نفوذ در شبکه‌های کامپیوتری استفاده شده است. روش پیشنهادی از بین ۴۱ ویژگی موجود، شش ویژگی را انتخاب کرده است و تنها با تکیه بر همان شش ویژگی توانسته است نفوذ را با دقت بالای ۹۹/۵۸درصد تشخیص دهد. در آزمایش‌های تجربی انجام شده، عملکرد روش پیشنهادی با به‌کار بردن دو رده‌بند مختلف (درخت تصمیم و نزدیک‌ترین K همسایه) با سه روش دیگر انتخاب ویژگی مورد بررسی و مقایسه قرار گرفته است. نتایج آزمایش‌ها نشان

داده است که روش پیشنهادی از نظر دقت و تعداد ویژگی‌های انتخاب شده (انتخاب ویژگی کمتر) توانسته است کارایی بالاتری نسبت به دیگر روش‌ها داشته باشد. همچنین، روش پیشنهادی از کیفیت بالایی برای انتخاب ویژگی برخوردار است به این شرط که تعداد ویژگی‌ها از چند ده مورد تجاوز نکند. در صورتی‌که تعداد ویژگی‌ها افزایش یابد روش کار سخت‌تر شده به نحوی که به نظر می‌رسد روش پیشنهادی برای تعداد ویژگی‌هایی در مقیاس ۱۰۰۰ و ۱۰۰۰۰ مورد نامناسب باشد. برای ادامه کار در آینده، می‌توان به دنبال توسعه نسخه‌های سریع‌تری از روش پیشنهادی بود تا برای تعداد ویژگی‌های بالا نیز به درستی عمل نماید. همچنین، تعداد پارامترها در روش پیشنهادی قابل توجه است. ارائه روشی در ادامه کار که بتواند با توجه به نوع مجموعه داده، مشخص نماید که چه مقداری برای هر یک از پارامترهای الگوریتم پیشنهادی مناسب‌تر است، کمک موثری به عملکرد و استفاده مناسب‌تر از روش پیشنهادی بر روی مجموعه داده‌های استاندارد مختلف خواهد داشت.

۶- منابع

[1] O. Joldzic, Z. Djuric, and P. Vuletic, "A transparent and scalable anomaly-based DoS detection method," Computer Networks, vol. 104, pp. 27-42, 2016.

[2] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," Journal of Network and Computer Applications, vol. 60, pp. 19-31, 2016.

[3] T. F. Ghanem, W. S. Elkilani, and H. M. Abdul-Kader, "A hybrid approach for efficient anomaly detection using metaheuristic methods," Journal of advanced research, vol. 6, no. 4, pp. 609-619, 2015.

[4] C. N. Modi, D. R. Patel, A. Patel, and M. Rajarajan, "Integrating signature apriori based network intrusion detection system (NIDS) in cloud computing," Procedia Technology, vol. 6, pp. 905-912, 2012.

[5] K. Shafi and H. A. Abbass, "An adaptive genetic-based signature learning system for intrusion detection," Expert Systems with Applications, vol. 36, no. 10, pp. 12036-12043, 2009.

[6] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," Expert Systems with Applications, vol. 39, no. 1, pp. 424-430, 2012.

[7] U. Stanczyk, "RELIEF-based selection of decision rules," Procedia Computer Science, vol. 35, pp. 299-308, 2014.

[8] P. Bermejo, L. de la Ossa, J. A. Gámez, and J. M. Puerta, "Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking," Knowledge-Based Systems, vol. 25, no. 1, pp. 35-44, 2012.

[9] P. Bermejo, J. A. Gámez, and J. M. Puerta, "Speeding up incremental wrapper feature subset selection with Naive Bayes classifier," Knowledge-Based Systems, vol. 55, pp. 140-147, 2014.

[10] T. A. Feo and M. G. Resende, "A probabilistic heuristic for a computationally difficult set covering problem," Operations research letters, vol. 8, no. 2, pp. 67-71, 1989.

[11] X. Chen, Y. Ye, X. Xu, and J. Z. Huang, "A feature group weighting method for subspace clustering of high-dimensional data Pattern Recognition," vol. 45, no. 1, pp. 434-446, 2012.

[12] P. Bermejo, J. A. Gámez, and J. M. Puerta, "A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets," Pattern Recognition Letters, vol. 32, no. 5, pp. 701-711, 2011.

[13] P. Festa and M. G. Resende, "An annotated bibliography of GRASP–Part I: Algorithms," International Transactions in Operational Research, vol. 16, no. 1, pp. 1-24, 2009.

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," In Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009, IEEE Symposium on, pp. 1-6, July 2009.

Journal of Electronical & Cyber Defence
Vol. 6, No. 4, 2019, Serial No. 24

Improving Intrusion Detection System Using a New Feature Selection Technique

Z. Jafarpour, F. Rad*, H. parvin

Department of Computer Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran and

Young Researchers and Elite Club, Yasooj Branch, Islamic Azad University, Yasooj, Iran

(Received: 14/01/2018, Accepted: 27/05/2018)

ABSTRACT

Intrusion detection is an important subject of research in the cyberspace field. In an Intrusion DetectionSystem (IDS), redundant and irrelevant features have a negative impact on the IDS performance. Therefore, an appropriate feature selection method is an important part of IDSs for eliminating unrelated and redundant features. In this paper, a new feature selection method is proposed that joins features level to level and step by step to select a subset of proper features in order to finally detect intrusion more accurately and speedily. The purpose of the proposed method is applying it in intrusion detection systems to distinguish a normal the connection from an intruding connection to the network. The experiments on the NSL-KDD dataset show that the proposed method in comparison with other methods selects only six important features among the 41 features in the baseline, and can detect an intrusion with precision above 99.58% by relying only on these six features. In other words, the proposed method's failure has been 42 in 10,000 connections of the network and has correctly identified other 9958 regular connections and labeled them as normal. Finally, improvement in the algorithm runtime and the percentage accuracy of the proposed method in comparison with other methods has been verified and reported.

Keywords: Security in cyberspace, Intrusion detection, Classification, Feature selection, anomaly, attack

* Corresponding Author Email: Rad@iauyasooj.ac.ir