نشریه علمی «علوم و فناوری‌های پدافند نوین»
سال دهم، شماره ۳، پاییز ۱۳۹۸؛ ص ۲۹۶-۲۸۷

ارائه روشی جدید جهت بهبود تشخیص نفوذ با استفاده از ترکیب الگوریتم جنگل تصادفی و الگوریتم ژنتیک

سید جواد کاظمی‌تبار[1*]، ریحانه طاهری امیری[2]، قربان خردمندیان[3]

۱- استادیار دانشگاه صنعتی نوشیروانی بابل، ۲- کارشناس ارشد دانشگاه علوم و فنون مازندران، ۳- دکتری، داده‌کاوان هوشمند توسن

(دریافت: ۹۷/۰۵/۰۱، پذیرش: ۹۷/۱۰/۰۲)

چکیده

همگام با گسترش شبکه‌های کامپیوتری، حملات و نفوذها به این شبکه‌ها نیز افزایش یافته است. برای داشتن امنیت کامل در یک سامانه کامپیوتری، علاوه بر فایروال‌ها و دیگر تجهیزات جلوگیری از نفوذ، سامانه‌های دیگری به نام سامانه‌های تشخیص نفوذ (IDS) مورد نیاز هستند. هدف از یک سامانه تشخیص نفوذ نظارت بر فعالیت‌های غیرعادی و افتراق بین رفتارهای طبیعی و غیرطبیعی (نفوذ) در یک سامانه میزبان و یا در یک شبکه است. یک سامانه تشخیص نفوذ را زمانی می‌توان کارا دانست که نرخ تشخیص نفوذ بالا و به‌صورت هم‌زمان نرخ هشدار اشتباه کمی را دارا باشد. در این مقاله روشی جدید جهت طبقه‌بندی مجموعه داده KDD-Cup-99 معرفی شده است که از ترکیب الگوریتم جنگل تصادفی و الگوریتم ژنتیک حاصل شده است و هدف آن افزایش سرعت فاز یادگیری و آزمون و همچنین دقت روش جنگل تصادفی است. از جنگل تصادفی به دلیل ساختار ساده و کارایی بالای آن در بسیاری از محصولات مبتنی بر یادگیری ماشین استفاده می‌شود. ولی مانند دیگر الگوریتم‌های مبتنی بر درخت تصمیم، وجود تعداد زیادی متغیر غیرعددی (نوعی) می‌تواند برای دقت و سرعت برنامه مشکل ایجاد کند. در مسئله تشخیص نفوذ دقیقاً ما با چنین سناریویی مواجه هستیم. نوآوری این مقاله، حل این معضل با استفاده از الگوریتم ژنتیک است. در این مقاله با تعریف کردن معیاری با نام بهره اطلاعات، تعداد ویژگی‌ها کاهش یافته است.

کلیدواژه‌ها: تشخیص نفوذ مبتنی بر الگوی رفتاری، داده‌کاوی، الگوریتم ژنتیک، الگوریتم جنگل تصادفی

A Novel Technique for Improvement of Intrusion Detection *via* Combining Random Forrest and Genetic Algorithm

J. Kazemitabar*, R. Taheri, Gh. Kheradmandian
Babol Noshirvani University of Technology
(Received: 23/07/2018; Accepted: 23/12/2018)

Abstract

As computer networks grow, so attacks and intrusions to these networks are increased. In order to have a fully secure computer network, one needs 'intrusion detection systems' (IDS) on top of firewalls. The goal of using an IDS is to supervise the abnormal activities and differentiate between normal and abnormal activities in a host system or in a network. An efficient IDS has high detection rate while keeping a low false alarm rate. In this paper, a new approach to classify KDD-Cup-99 data set using a combination of random forest method and genetic algorithm is presented. The purpose is to increase the speed of learning and test phases while improving the accuracy. Random forest is an ensemble learning method based on decision trees. Due to its relatively simple structure and good performance, it is used in many supervised learning applications. However, like all tree based machine learning algorithms, having too many categorical features, can be a problem both for the speed and accuracy. This is exactly the case with the problem in hand, i.e. intrusion detection; many of the features are in the form of categorical data. For example, in R language, the maximum number of definable categorical features for random forest is 53. The contribution of this work is resolving this issue with the aid of Genetic Algorithm (GA). In this research information gain as a measure of importance is defined and the number of features is reduced based on genetic algorithm.

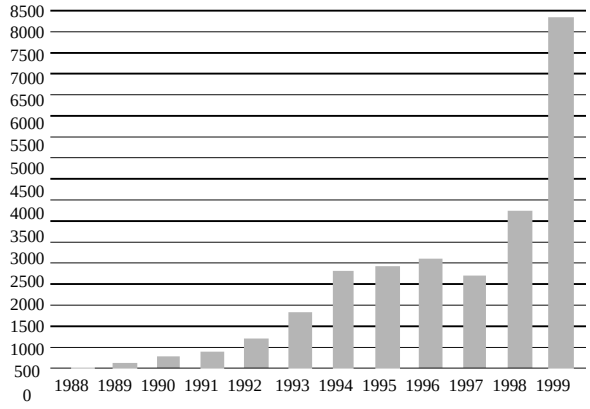
Keywords: Signature-Based Intrusion Detection, Data Mining, Genetic Algorithm, Random Forest Algorithm

*Corresponding Author E-mail: j.kazemitabar@nit.ac.ir

۱. مقدمه

امروزه با رشد استفاده و افزایش اهمیت شبکه‌های کامپیوتری، حملات و نفوذها به این شبکه‌ها نیز به‌صورت فزاینده گسترش یافته و به شکل‌های متعددی انجام می‌گیرد. نفوذ، تمامی اقدامات غیرقانونی را که صحت، محرمانگی و یا دسترسی به یک منبع را به خطر اندازد را شامل می‌شود [۱]. نفوذی‌ها به دو دسته خارجی و داخلی تقسیم می‌شوند. نفوذی‌های خارجی کسانی هستند که بدون اجازه استفاده از منابع سامانه، جهت دست‌یابی به آن‌ها تلاش می‌کنند و نفوذی‌های داخلی کسانی هستند که با دارا بودن اختیارات محدودی در سامانه سعی در دست‌یابی به منابعی خارج از حوزه دسترسی خود را دارند. در گذشته، نفوذها بیشتر از ناحیه افرادی انجام می‌شد که علاقه‌مند بودند تا مهارت‌ها و توانایی‌های خود را آزمایش کنند، اما در حال حاضر، تمایل به نفوذ با انگیزه‌های مالی، سیاسی و نظامی بیشتر شده است. اینترنت و فرایندهای برخط یکی از ابزارهای ضروری زندگی روزمره امروز هستند و به‌عنوان یک جز مهم از عملیات کسب‌وکار مورد استفاده قرار می‌گیرند. در دنیای مدرن امروز، نفوذ در کسری از ثانیه صورت می‌پذیرد. نفوذگران به‌صورت کاملاً هوشمندانه با استفاده از نسخه اصلاح‌شده از فرمان‌ها اقدام به نفوذ نموده و سپس ردپای خود را از گزارش‌ها پاک می‌کنند، بنابراین، امنیت شبکه باید به‌دقت مورد توجه قرار گیرد تا کانال‌های اطلاعاتی امنی را فراهم آورد. تشخیص نفوذ فرآیند مانیتور کردن رویدادهای رخ‌داده در یک سامانه کامپیوتری و یا یک شبکه و تحلیل آن‌ها برای پیدا کردن نشانه‌هایی از حوادث است. این حوادث می‌توانند تخلف یا تهدیدات قریب‌الوقوع نسبت به سیاست‌های کاری امنیتی، سیاست‌های موردقبول در استفاده از سامانه و یا استانداردهای امنیتی باشد. سامانه‌های تشخیص نفوذ[1] بر روی شناسایی حوادث ممکن، اطلاعات واقعه‌نگاری در مورد آن‌ها، تلاش برای متوقف کردن آن‌ها و گزارش آن‌ها به مدیران امنیتی تمرکز دارد. به‌علاوه، سازمان‌ها سامانه‌های تشخیص نفوذ را برای هدف‌های دیگری هم استفاده می‌کنند. مواردی از قبیل شناسایی مشکلات در سیاست‌های امنیتی، مستندسازی تهدیدات موجود و بازداشتن افراد از تخلف از سیاست‌های امنیتی، جزو این هدف‌ها است [۲].

این بحث با یک حقیقت آشکار دیگر ترکیب می شود. افزایش این گونه حملات در اینترنت خود بازگوکننده رشد همه‌گیرشده بیشتر اینترنت در سال‌های گذشته است. در شکل (۱) تعداد حوادثی که به CERT/CC گزارش شده‌اند در یک بازه ۱۲ ساله نشان داده شده است. تجارت الکترونیک به‌تنهایی توانسته شدت زیادی را به این روند ببخشد [۲].



شکل ۱. نرخ رشد حملات ثبت‌شده به‌وسیله CERT/CC [۲]

درحالی‌که در سال‌های گذشته حملات خارجی اغلب برای گشت‌وگذارهای شخصی و آزمایش مقدار مهارت هکرها انجام می‌شد؛ این روزها هدف‌های نظامی، اقتصادی و سیاسی بیش‌ترین نقش را در تحریک هکرها داشته است. هدف سامانه‌های تشخیص نفوذ، تشخیص دسترسی‌های غیرمعمول و یا حملات به یک شبکه داخلی است. سامانه تشخیص نفوذ مبتنی بر شبکه، یک ابزار ارزشمند برای دفاع در عمق شبکه‌های کامپیوتری هستند. سامانه تشخیص نفوذ، به دنبال فعالیت‌های شناخته‌شده و یا بالقوه مخرب در ترافیک شبکه می‌گردد و هر زمان یک فعالیت مشکوک را شناسایی کند زنگ خطری را به صدا درمی‌آورد.

به‌طورکلی سامانه‌های تشخیص نفوذ به دو روش تقسیم می‌شوند: تشخیص مبتنی بر الگو[2] و تشخیص ناهنجاری[3]. روش تشخیص ناهنجاری این مزیت را نسبت به روش تشخیص مبتنی بر الگو دارد که می‌تواند حملات ناشناخته را نیز شناسایی کند. روش مبتنی بر الگو نیازمند یادگیری الگوی کاربر معمولی جهت تشخیص آن از الگوی رفتاری حمله‌کنندگان و نفوذگران است. این امر نیازمند داده‌های نمونه رفتاری هر یک از این دو گروه است.

با تشخیص به‌موقع اقدامات انجام‌گرفته در راستای نفوذ به سامانه می‌توان امیدوار بود که امنیت تا حدودی در شبکه برقرار گردد. جهت دست‌یابی به این هدف یکی از ابزارهایی که می‌تواند مورد استفاده قرار گیرد، داده‌کاوی است. تاکنون از روش‌های مختلف داده‌کاوی برای یادگیری الگوی رفتاری کاربران معمولی و کاربران اخلال‌گر استفاده شده است. این مقاله با هدف بهبود سرعت و دقت تشخیص نفوذ با استفاده از روش‌های داده‌کاوی انجام گرفته که بر روی داده‌های نمونه KDD-Cup-99 اجرا شده است.

[1] Intrusion Detection System (IDS)

[2] Misuse Detection

[3] Anomaly Detection

۱-۱. جنگل تصادفی

الگوریتم جنگل تصادفی اولین بار در سال ۲۰۰۱ توسط بریمن معرفی گردید. جنگل تصادفی یک الگوریتم طبقه‌بندی نظارت شده است که شامل مجموعه‌ای از درخت‌های تصمیم‌گیر است. الگوریتم درخت تصمیم‌گیر یکی از الگوریتم‌های طبقه‌بندی و از رایج‌ترین روش‌های داده‌کاوی است که سادگی و کارآمدی آن باعث شده تا علی‌رغم مشکلاتی که در اجرای الگوریتم با متغیرهای دارای نویز یا صفات فاقد مقدار وجود دارد، به شکل گسترده‌ای در کاربردهای مختلف و مسائل مربوط به یادگیری ماشین استفاده شود. در درخت تصمیم با دنبال کردن مجموعه‌ای از سؤالات مرتبط با خصوصیات داده‌ها و نگاه به داده جاری برای اتخاذ تصمیم، طبقه یا دسته آن تعیین می‌شود. CART یک الگوریتم درخت باینری در الگوریتم درخت تصمیم است. جنگل تصادفی مجموعه‌ای از درختان CART است و تحت چهار مرحله بیان می‌شود.

۱. K زیرمجموعه از نمونه‌های آموزش $(D_1, D_2 \dots . D_k)$ در میان مجموعه کل نمونه‌های موجود در بخش آموزش (D) توسط روش نمونه‌برداری Bootstrap انتخاب می‌شوند. درنهایت K درخت تصمیم‌گیر ایجاد خواهد شد.

۲. در N شاخص گره درخت طبقه‌بندی، m مشخصه به‌طور تصادفی انتخاب می‌شود و مطابق با اصل حداقل خلوص گره، بهترین مشخصه در بین M شاخص کاندید انتخاب خواهد شد. به این‌ترتیب درختان رشد خواهند کرد.

۳. این مرحله تکرار گام دوم است. K درخت تصمیم‌گیر تولید می‌شود.

K درخت تصمیم‌گیر که به‌خوبی رشد پیدا کرده‌اند جنگل تصادفی طبقه‌بند ترکیبی را تشکیل می‌دهند. نمونه واقع در طبقه نهایی جنگل تصادفی منتظر رأی اکثریت می‌ماند.

۱-۲. الگوریتم ژنتیک

الگوریتم ژنتیک یکی از زیرمجموعه‌های محاسبات تکامل‌یافته است که رابطه مستقیمی با مبحث هوش مصنوعی دارد در واقع الگوریتم ژنتیک یکی از زیرمجموعه‌های هوش مصنوعی است. الگوریتم ژنتیک را می‌توان یک روش جستجوی کلی نامید که از قوانین تکامل بیولوژیک طبیعی تقلید می‌کند. الگوریتم ژنتیک بر روی یک سری از جواب‌های مسئله به امید به‌دست آوردن جواب‌های بهتر قانون بقای بهترین را اعمال می‌کند. در هر نسل به کمک فرآیند انتخابی متناسب باارزش جواب‌ها و تولیدمثل جواب‌های انتخاب‌شده به کمک عملگرهایی که از ژنتیک طبیعی تقلید شده‌اند، تقریب‌های بهتری از جواب نهایی به دست می‌آید. این فرایند باعث می‌شود که نسل‌های جدید با شرایط مسئله سازگارتر باشد.

هرکدام از افراد جمعیت، که تقریب‌هایی از جواب نهایی‌اند، به‌صورت رشته‌هایی از حروف یا ارقام کدگذاری می‌شوند. این رشته‌ها را کروموزوم می‌نامند. متداول‌ترین حالت، نمایش با ارقام صفر و یک است. حالت‌های دیگر مثل استفاده از سه رقم، اعداد حقیقی و اعداد صحیح هم مورد استفاده قرار می‌گیرند. برای مثال، یک کروموزوم با دو متغیر a و b با ساختار 1001100101100101011110100 نمایش داده می‌شود. متغیر a با ۱۰ خانه اول سمت راست و b با ۱۵ خانه باقیمانده نمایش داده شده است. این می‌تواند به علت سطح دقت و یا محدوده متغیر تصمیم‌گیری باشد. مقادیر موجود بر روی کروموزوم‌ها به‌تنهایی معنی خاصی ندارد بلکه باید از حالت کدشده خارج شوند تا به‌عنوان متغیرهای تصمیم‌گیری دارای معنی و نتیجه باشند باید توجه داشت که فرآیند جستجو بر روی اطلاعات کدشده انجام می‌گیرد مگر درصورتی‌که از ژن‌هایی با مقادیر حقیقی استفاده شود. بعد از اینکه کروموزوم‌ها از حالت کدگذاری‌شده خارج شدند می‌توان کارایی یا برازش هر فرد از جمعیت را محاسبه کرد. برازش مقیاسی نسبی است که شایستگی افراد برای تولید نسل بعد را نشان می‌دهد. در طبیعت برازش معادل توانایی فرد برای بقا است. تابع هدف در تعیین برازش افراد نقش تعیین‌کننده دارد.

در هنگام تکثیر به کمک اطلاعات اولیه‌ای که از تابع هدف به دست می‌آید برازش هر فرد مشخص می‌گردد. از این مقادیر در فرآیند انتخاب استفاده می‌شود تا آن را به سمت انتخاب افراد مناسب سوق دهد.

۱-۳. پایگاه داده KDD

از سال ۱۹۹۹ داده‌های KDD به‌عنوان یکی از پرکاربردترین مجموعه داده‌های پیشنهادی جهت ارزیابی روش‌های مختلف تشخیص ناهنجاری مورد استفاده بوده است. این مجموعه داده توسط استولفو و همکاران [۳] بر اساس داده‌های برنامه ارزیابی تشخیص نفوذ DARPA98 تولید شده است. DARPA98 حاوی ۴ GB اطلاعات خام هفت هفته ترافیک شبکه است. اطلاعات دو هفته آن نزدیک به دو میلیون رکورد ارتباطی را شامل می‌شود. داده‌های آزمون KDD-Cup-99 درمجموع حاوی ۳۱۱۰۲۷ رکورد اطلاعاتی است که تعداد ۶۰۵۹۱ رکورد آن نمایش‌دهنده فعالیت نرمال و تعداد ۲۵۰۴۳۶ رکورد حمله را شامل می‌گردد. هر رکورد از داده‌ها دارای ۴۱ ویژگی هستند و هر ردیف در ستون ۴۲ دارای برچسب مشخص‌کننده نرمال یا حمله هستند، درواقع هر

رفتاری غیر از نرمال به‌عنوان نوع خاصی از حمله برچسب‌گذاری شده است [۴] . انواع کلاس‌های حمله موجود در KDD-Cup-99 به چهار گروه زیر تقسیم می‌گردد [۶- ۵].

محرومیت از سرویس[1]: نوعی از حمله را شامل می‌شود که منابع قربانی را با انجام برخی رایانش‌ها و یا اشغال‌نمودن منابع حافظه به حدی درگیر می‌نماید که مانع از سرویس‌دهی به درخواست‌های قانونی قربانی می‌گردد. حملات محرومیت از سرویس شامل Neptune, Back, Smurf, Pod, Land و Teardrop است.

کاوش[2]: در این حالت مهاجم اقدام به جمع‌آوری اطلاعاتی از راه دور در رابطه با شبکه می‌نماید تا هدف مورد نظر که دور زدن اقدامات امنیتی است ممکن گردد. کاوش شامل حملات Portsweep, Satan, Nmap و Ipsweep است.

دسترسی غیرمجاز کاربر محلی به مدیر[3]: نوعی از حمله را شامل می‌شود که مهاجم به‌وسیله یک حساب کاربری عادی به سامانه قربانی وارد شده و با استفاده از یک سری آسیب‌پذیری‌ها تلاش می‌نماید تا دسترسی مدیر را به دست آورد. U2R شامل حملات Buffer_overflow, Loadmodule, Rootkit و Perl است.

دسترسی غیرمجاز از یک ماشین از راه دور[4]: مهاجم به یک ماشین از راه دور نفوذ نموده و دسترسی محلی به ماشین قربانی پیدا می‌کند. R2L شامل حملات Warzclient, Multishop, Phf و ftp_write, Imap, Guess_password, Warezmaster, Spy است.

اگرچه، این نسخه از مجموعه KDD-Cup-99 از برخی مشکلات مطرح‌شده توسط مک‌هاف [۷] رنج می‌برد و به دلیل دارا نبودن مجموعه داده‌های عمومی برای سامانه‌های تشخیص نفوذ مبتنی بر شبکه، نماینده کاملی از شبکه‌های واقعی موجود نیست. اما عقیده بر آن است که باوجود تمام این کمبودها می‌تواند به‌عنوان داده معیار مرجع اثرگذاری برای کمک به محققان جهت مقایسه روش‌های مختلف تشخیص نفوذ مورد استفاده قرار گیرد.

۱-۴. تابع آنتروپی

نویسندگان این مقاله به دنبال به حداکثر رسانیدن محتوای اطلاعات از تمامی پارامترها هستند. یک روش جهت دست‌یابی به هدف فوق بهینه‌سازی آنتروپی است. در این روش به حداقل رسانیدن آنتروپی منجر به حداکثر رسانیدن میزان پشتیبانی هر پارامتر از شواهد می‌گردد، درحالی‌که منجر به حداقل رسانیدن عدم قطعیت در آمار و آنتروپی متقاطع بین الگوریتم و داده می‌شود.

آنتروپی می‌تواند به‌عنوان یک معیار اندازه‌گیری از محتویات اطلاعات مورد انتظار و یا عدم اطمینان از توزیع احتمال تعریف شود. همان‌طور می‌تواند به‌عنوان میزان بی‌نظمی درون یک سامانه و یا میزان عدم قطعیت یک پارتیشن را شامل شود.

فلسفه کاهش آنتروپی در زمینه تشخیص الگو را می‌توان برای طبقه‌بندی، تجزیه تحلیل داده‌ها و داده‌کاوی که یکی از وظایف مطرح‌شده در آن کشف الگو یا نظم در مجموعه داده‌های بزرگ است، مورد استفاده قرار داد. قواعد ساختار داده‌ها توسط مقادیر آنتروپی کوچک و در مقابل مقادیر تصادفی با آنتروپی‌های بزرگ مشخص می‌گردند. در زمینه داده‌کاوی شناخته‌شده‌ترین برنامه کاربردی از آنتروپی، جمع‌آوری داده‌های درخت تصمیم است.

در نظریه اطلاعات و یادگیری ماشین، از مفهومی به نام بهره اطلاعات یاد می‌شود. امید ریاضی بهره اطلاعات همانا تابع اطلاعات متقابل یا همان میزان کاهش آنتروپی در صورت به دست آوردن اطلاعات است. در یادگیری ماشین و هوش مصنوعی از این مفهوم برای تعین ویژگی استفاده می‌شود [۱۴]. بهره اطلاعات[5] یک ویژگی عبارت است از مقدار کاهش آنتروپی که به‌واسطه جداسازی مثال‌ها از طریق این ویژگی حاصل می‌شود. به‌عبارت‌دیگر بهره اطلاعات Gain(S,A) برای یک ویژگی نظیر A نسبت به مجموعه مثال‌های S به‌صورت رابطه (۶) تعریف می‌شود:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (۶)$$

که در آن Values(A) مجموعه همه مقدار ویژگی‌های A بوده و S_V زیرمجموعه‌ای از S است که برای آن A دارای مقدار V است. در تعریف فوق، عبارت اول مقدار آنتروپی داده‌ها و عبارت دوم مقدار آنتروپی مورد انتظار بعد از جداسازی داده‌هاست.

۱-۵. معیارهای ارزیابی

عملکرد روش‌ها را می‌توان برمبنای صحت (دقت)[6]، حساسیت[7]، ویژگی[8]، دقت[9] و زمان مقایسه نمود. محاسبه معیارهای فوق به‌وسیله تشخیص موفق، نادیده‌گرفتن یک رفتار قابل‌قبول اخطار اشتباه و عدم موفقیت در تشخیص یک حمله (که در جدول (۱) به‌صورت خلاصه توضیح داده شده است) امکان‌پذیر است. هیچ‌یک از موارد نادیده گرفتن یک رفتار قابل‌قبول و تشخیص موفق تا زمانی که سامانه تشخیص نفوذ مطابق انتظار عمل می‌کند خطرناک نیستند. در واقع عدم موفقیت در تشخیص

[1] DOS: Denial of Service
[2] Probing
[3] U2R
[4] R2L
[6] Accuracy
[7] Sensitivity
[8] Specificity
[9] Precision

یک حمله به‌عنوان جدی‌ترین و خطرناک‌ترین حالت ممکن ایفای نقش می‌نماید که زمانی است که سامانه تشخیص نفوذ یک حمله را به‌عنوان یک رفتار قابل‌قبول قلمداد می‌نماید. این حالت در واقع شکست سامانه در کشف یک حمله است. خروجی‌های FP،FN،TP،TN حاصل از اجرای الگوریتم‌های پیشنهادی پژوهش جاری جهت محاسبه دقت مشاهده است.

جدول ۱. حالات مختلف ممکن بین واقعیت و تشخیص سیستم

True	تشخیص: حمله واقعیت: حمله نتیجه: تشخیص موفق یک حمله	تشخیص: رفتار قابل‌قبول واقعیت: رفتار قابل‌قبول نتیجه: نادیده گرفتن یک رفتار قابل‌قبول
False	تشخیص: حمله واقعیت: رفتار قابل‌قبول نتیجه: اخطار اشتباه	تشخیص: رفتار قابل‌قبول واقعیت: حمله نتیجه: عدم موفقیت در تشخیص یک حمله

۱-۶. کاربرد هوش مصنوعی در سامانه‌های تشخیص نفوذ

هوش مصنوعی می‌تواند کاربری تشخیص نفوذ را بسیار ساده‌تر نماید. هوش مصنوعی می‌توانند اولویت‌های کارمندان امنیتی را یاد گرفته و هشدارهای حیطه هر کارمند را ابتدا نمایش دهد. همان‌طور که قابل پیش‌بینی است، سخت‌ترین قسمت آموزش، آموختن چیزهای درست به سامانه است. هوش مصنوعی می‌تواند با مشاهده کارهای یک مأمور امنیتی همان کارها را به‌عنوان یک سامانه مبتنی بر قانون یاد بگیرد. هوش مصنوعی می‌تواند اتفاقاتی را که به‌خودی‌خود ناچیز به نظر می‌آیند اما پس از ترکیب با هم ممکن است نمایانگر جریان داشتن یک تهاجم باشند را نیز تشخیص دهد.

بیان این موضوع که چگونه یک محیط سامانه تشخیص نفوذ، اولویت‌ها و روش‌های کاری کارمندان امنیتی را خواهد دانست، و چگونه این محیط را با ارائه ناهنجاری‌هایی که بیشتر اوقات دیده و کشف‌شده کاربردی‌تر باشد هدفی واقعاً ساده است، اما برنامه‌نویسی سامانه‌ای که قادر به انجام این کارها باشد به‌هیچ‌عنوان ساده نیست. نامرئی و استمرارپذیر کردن پروسه آموزش نیز بر سختی این کار می‌افزاید.

مشکل‌ترین کار برای هوش مصنوعی کشف ارتباط بین رویدادهای متفاوت است. راه‌های گوناگونی برای دست‌یابی به این هدف وجود دارد، که شبکه‌های عصبی برجسته‌ترین آن‌ها است. در ادامه به‌مرور پژوهش‌های متکی بر هوش مصنوعی سامانه‌های تشخیص نفوذ پرداخته می‌شود.

گروهی از روش‌ها از رأی اکثریت استفاده نموده‌اند، لین و همکاران [۸]، یک روش طبقه‌بندی SVM بر مبنای جنگل دوار را پیشنهاد دادند. نتایج حاصل از طبقه‌بندی کنندگان با استفاده از رأی اکثریت ترکیب شد. در این تحقیق از مجموعه داده KDD-Cup-99 جهت آزمون بازدهی روش استفاده شد. نتایج نشان‌دهنده آن بود که استفاده از یک SVM دولایه مبتنی بر جنگل دوار، دقت بهتری را در حملات R2L و Probe داشته است.

مالک و همکاران [۹] یک طبقه‌بندی‌کننده بر اساس BPSO[1] و جنگل تصادفی جهت طبقه‌بندی و تشخیص حملات Probe در شبکه‌ها را ارائه نمودند. عملکرد با استفاده از مجموعه داده KDD-Cup-99 اعتبارسنجی شد. عملکرد این روش در مقابل حملات Probe خوب بود اما ضعف آن توزیع یکسان داده‌های آموزش و آزمون بود.

بوختویوروف و ژوکوف [۱۰] یک رهیافت احتمالی جهت طراحی طبقه‌بندی‌کننده‌های مبتنی بر شبکه عصبی را به نام PGNS[2] طراحی نمودند. تجمیع طبقه‌بندی‌کنندگان شبکه عصبی GPEN[3] انجام شد. GPEN از اپراتورهای برنامه‌نویسی ژنتیک برای یافتن یک تابع بهینه جهت ترکیب طبقه‌بندی‌کننده‌های پایه به یک گروه استفاده نمود. این پژوهش بر روی مجموعه داده KDD-Cup-99 صورت گرفت که هدف از آن تمایز بین حملات Probe و غیر Probe بر اساس ۹ ویژگی از ۴۱ ویژگی مجموعه داده مذکور است. پژوهشگران در این تحقیق نتایج حاصل را با نتایج تحقیق دیگری [۱۲] مقایسه نمودند. نتایج حاصل از اجرای روش پیشنهادی بوختویوروف و ژوکوف دقت تشخیص بهتری را در تشخیص حملات Probe به نسبت تقریباً تمامی روش‌های مقایسه شده [۹] نمایش داد. تنها روشی که دقت تشخیص بهتر و اخطار اشتباه کمتری را ارائه داد، روش PSO-RF بود. این روش به‌صورت خاص در تشخیص حملات Probe بسیار خوب عمل می‌نماید اما بر روی حملات ناشناخته آزمون نگردیده است، یکی از معایب دیگر این روش آن است که دقت آن به‌اندازه روش‌های دیگر نیست. در میان روش‌هایی که از ترکیب روش‌ها استفاده نموده‌اند. بحری و همکاران [۱۱]، یک روش ترکیبی بر اساس یک روش گروهی به نام Greedy-Boost را معرفی نمودند. آن‌ها به مقایسه دقت و به یادآوری روش‌های AdaBoost, C4.5 و Greedy-Boost جهت طبقه‌بندی پایگاه داده KDD-Cup-99 اقدام نمودند. نتایج نشان داد که روش Greedy-Boost عملکرد بهتری نسبت به سایر الگوریتم‌ها ازنظر دقت برای حملاتی همچون Probe, U2R و R2L را نمایش داد. این روش در

[1] Binary Particle Swarm Optimization
[2] Probability Based Generator of Neural Network Structures
[3] Genetic Programming Based Ensembling

تشخیص حملات نادر و کاهش هزینه میانگین نتایج خوبی را بروز داد اما بر روی حملات ناشناخته آزمون نگردید.

پرویز و فرید [۱۲]، یک روش ترکیبی بر اساس انتخاب ویژگی و طبقه‌بندی بعدازآن را با استفاده از پایگاه داده KDD-Cup-99 معرفی نمودند. انتخاب ویژگی از طریق روش LOO اجرا شد و نویسندگان به‌عنوان طبقه‌بندی‌کننده از ماشین‌های OVAR-SVM استفاده نمودند. آزمایش‌ها نشان داد که بیش‌ترین دقت در طبقه‌بندی با انتخاب ۱۴ ویژگی حاصل شد. نجفی و رافع [۱۳]، از الگوریتم ژنتیک برای کاهش ویژگی هنگام استفاده از درخت تصمیم برای تشخیص نفوذ استفاده نمودند.

در این تحقیق، یک الگوریتم تشخیص نفوذ با استفاده از الگوریتم ژنتیک و جنگل تصادفی با هدف بهبود سرعت و دقت تشخیص ارائه گردیده است. جهت انجام این امر با استفاده از ابزارهای داده‌کاوی، نیاز به یک سری داده نمونه وجود داشت که از داده‌های KDD-Cup-99 جهت آموزش و آزمون روش پیشنهادی استفاده شده است.

۲. الگوریتم‌های پیشنهادی

سامانه پیشنهادی، با تحلیل رفتار کاربران خود اقدام به شناسایی رفتارهای مخرب با توجه به آموزش‌های داده شده می‌نماید. سامانه پس از دریافت داده‌های برچسب دار در مرحله پیش‌پردازش با ترکیب روش‌های انتخاب ویژگی و اعمال تبدیل بر روی ویژگی‌ها اقدام به کاهش تعداد ویژگی‌ها به حداقل مقدار ممکن می‌نماید. در فاز دوم با اعمال الگوریتم جنگل تصادفی به فراگیری این الگوها اقدام می‌نماید (شکل ۲).

در این سامانه پس از انتخاب مجموعه داده‌های حاوی رفتار کاربران، ابتدا اقدام به انتخاب ویژگی‌هایی با بیش‌ترین اثرگذاری از میان ویژگی‌های موجود شده که در اینجا ویژگی‌های منتخب شامل ویژگی‌های اول تا ششم از دیتاست KDD-Cup-99 است.

در الگوریتم‌های مبتنی بر درخت، اگر ویژگی‌هایی که نوع آن‌ها غیرعددی[1] است، تعداد مقادیر منحصربه‌فرد زیادی داشته باشند، در این صورت اعمال روش‌های مبتنی بر درخت با چالش کاهش سرعت (در هر دو فاز آموزش و آزمون) همچنین کاهش دقت مواجه خواهد شد. و ازآنجاکه در package الگوریتم جنگل تصادفی در زبان R حداکثر تعداد مقادیر یک ویژگی غیرعددی محدودیت داشته و در بیش‌ترین حالت مقداری برابر ۵۳ را می‌تواند دارا باشد، در مرحله بعد، اقدام به شناسایی ویژگی یا ویژگی‌هایی با تعداد مقدار غیرعددی بیش از ۵۳ می‌شود. در بررسی‌های انجام‌شده مشخص شد که ویژگی سوم جزو این دسته بوده و نیازمند اعمال تغییراتی در راستای کاهش تعداد مقادیر تا حدنصاب قابل‌قبول توسط الگوریتم جنگل تصادفی است.

در الگوریتم ژنتیک، هر کروموزوم (جواب) یک زیرمجموعه از مقادیر برای یک ویژگی و تابع برازش نیز تابع آنتروپی است که برای هر کروموزوم (جواب)، مشخص می‌کند که برای یک گروه بندی از مقادیر، احتمال وقوع هر کلاس چقدر است و زمانی کمینه می‌شود که کلاس‌های متناظر با این مقادیر برای ویژگی انتخاب شده فقط به یک کلاس تعلق داشته باشند. به‌عبارت‌دیگر مقدار تابع آنتروپی برای یک گروه از مقادیر، زمانی کمینه است که احتمال وقوع یک کلاس برای آن گروه از مقادیر نزدیک به ۱ باشد و درعین‌حال احتمال وقوع بقیه کلاس‌ها برای آن گروه از مقادیر نزدیک به صفر باشد.

شکل ۲. شمای الگوریتم پیشنهادی در این پژوهش

[1] Categorical

جدول ۲. یک نمونه از ماتریس نهایی پس از اجرای جنگل تصادفی

برچسب داده

پیش‌گویی	نوع حمله	Apache2	Back	Buffer_overflow	Ftp_write	Guess_passwd	Httptunnel	Imap	Ipsweep	Land	Loadmodule	Mailbomb	Mscan	Multihop	Named	Neptune	Nmap	Normal	Perl	Phf	Pod	Portsweep	Processtable	Ps	Rootkit	Saint	Satan	Sendmail	Smurf	Snmpgetattack	Snmpguess	Sqlattack	Teardrop	Warezmaster	Worm	Xlock	Xsnoop	Xterm
	Apache2	70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Back	0	174	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Buffer Over Flow	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Ftp Write	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Guess Passwd	0	0	0	0	554	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Http Tunnel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	IMap	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Ip Sweep	0	0	0	0	0	0	0	58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Land	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Load Module	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Mail Bomb	0	0	0	0	0	0	0	0	0	0	783	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Mscan	0	0	0	0	0	0	0	0	0	0	0	74	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Multi Hop	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Named	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Neptune	45	0	0	0	0	20	0	0	2	0	0	71	0	2	9071	14	8	0	1	0	15	0	0	0	98	196	0	0	0	0	0	0	0	0	0	0	0
	Nmap	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Normal	1	0	5	0	131	3	0	0	0	0	5	19	5	2	0	0	9622	0	0	0	0	0	0	1	19	2	3	0	1219	389	1	3	186	0	0	0	3
	Perl	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Phf	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Pod	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Port Sweep	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Process Table	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	121	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Ps	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Rootkit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Saint	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Satan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	2	56	0	0	0	0	0	0	0	0	0	0	0
	Send Mail	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Smurf	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	11	0	0	0	0	0	0	0	0	26353	0	0	0	0	0	0	0	0
	Snmp Get Attack	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Snmp Guess	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sql Attack	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Tear Drop	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Warez Master	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Worm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Xlock	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Xsnoop	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Xterm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

در مرحله بعد اقدام به گروه‌بندی مقادیر ویژگی شد، در اینجا هدف گروه‌بندی خودکار مقادیر ویژگی‌هایی بود که از نوع غیرعددی بوده، به‌نحوی‌که تعداد مقادیر منحصربه‌فرد کم شده و درعین‌حال منجر به کاهش دقت نگردد (یک گروه از مقادیر به‌عنوان یک مقدار در نظر گرفته شدند) و پس از آماده‌سازی داده با ویژگی یا مقادیر جدید، مقادیر حاصل در اختیار فاز بعدی که همان روش جنگل تصادفی است قرار داده شده است.

فاز آخر این روش شامل دو مرحله یادگیری و آزمون است که در مرحله یادگیری سامانه را توسط بخشی از داده‌هایی که در اختیار داشت آموزش داده تا برای مرحله بعد که همان مرحله آزمون است آماده گردد. داده‌های آماده‌سازی شده در مرحله قبل را در اختیار جنگل قرار داده و با اجرای جنگل تصادفی با تعداد درختان متفاوت اقدام به بررسی میزان اثربخشی روش پیشنهادی در پیش‌بینی رفتار احتمالی کاربران بر اساس آموخته‌های پیشین نمودیم. با استفاده از این سامانه و بر مبنای پیش‌بینی انجام‌شده در مرحله آزمون می‌توان در زمانی کوتاه از رفتارهای خطرناک موجود در شبکه اطلاع یافته و جهت برخورد با آن تصمیم مناسب را اتخاذ نمود. در ادامه و در فصل آینده به تشریح نحوه پیاده‌سازی روش پیشنهادی در محیط R خواهیم پرداخت.

۳. نتایج و بحث

همان‌طور که قبلاً توضیح داده شد از میان داده‌های آزمون استاندارد موجود، در این پژوهش داده‌های KDD-Cup-99 با تمام فرضیات مسئله جهت بررسی انتخاب شده است. یکی از مراحل پیاده سازی الگوریتم تشخیص نفوذ پیشنهادی انتخاب ویژگی است. پس از بارگذاری داده‌ها، ساختار دیتا ست را استخراج نموده و مشاهده می‌نماییم که از ۴۲ ویژگی Dataset موردنظر تعداد ۳۸ ویژگی عددی و ۴ ویژگی غیرعددی است و تمامی

ویژگی‌ها به‌جز ویژگی‌های ردیف B و C و D و AP عددی هستند.

- ویژگی غیرعددی اول دارای ۳ مقدار[1]
- ویژگی غیرعددی دوم دارای ۶۵ مقدار
- ویژگی غیرعددی سوم دارای ۱۱ مقدار
- ویژگی غیرعددی آخر که در واقع label داده‌ها است دارای ۳۸ مقدار است. شمایی از داده‌های موجود در Dataset در جدول (۳) قابل مشاهده است.

شبه کد ۱: تابع برازش

```
Algorithm: Fitness Function

Input :  x, train

Values <- get_non_zero_indices(x)

Group <- concat(values)

Group_target <- get_target_for_group(train)

Return (entropy(Group_target))
```

همان‌طور که در شبه کد ۱ مشاهده می‌شود، تابع برازش در ورودی کروموزم‌ها یا همان راه‌حل‌های نامزد را می‌گیرد که آرایه‌ای از صفر و یک هستند. طول این آرایه به اندازه خود ویژگی‌ها (۶۵) است. ورودی دیگر تابع برازش داده‌های آموزش است. تابع برازش ابتدا اندیس و مقدار خانه‌های غیر صفر را می‌گیرد و کنار هم قرار می‌دهد. درنهایت آنتروپی نتیجه را به دست می‌آورد. بدیهی است هرچه مقدار آنتروپی کمتر باشد آن نامزد بهتری خواهد بود.

جهت اجرای فرایند انتخاب ویژگی‌ها در R بدین‌صورت عمل می‌شود که ابتدا ویژگی‌های ستون یک تا ۶ در یک متغیر ذخیره شد و سپس تعداد تکرار هر مقدار ستون سوم محاسبه و ذخیره شده است. پس از آن مقادیری از ستون سوم را که تعداد تکرارشان کمتر از ۵۴ بود را در یک متغیر دیگر ذخیره نموده و سطرهایی که ستون سوم آن‌ها حاوی مقادیر یافته شده در بالا (تعداد تکرار کمتر از ۵۴) بود از مجموعه داده‌های تحت بررسی حذف شد درنتیجه از ۶۵ مقدار ممکن ویژگی سوم تنها ۵۰ مقدار باقی ماند. با این اوصاف دیتاست تولیدی جدید را جایگزین دیتاست قبلی نموده و تعداد سطرها و ستون‌های آن محاسبه و ذخیره شد درنتیجه ۳۱۰۵۰۱ سطر از داده‌های اولی در ۷ ستون از ویژگی‌ها جهت انجام آزمون در مراحل بعدی باقی ماند.

در این مرحله داده‌های آموزش و داده‌های برچسب را به الگوریتم جنگل تصادفی ارائه داده و جنگل‌هایی متشکل از ۱ تا ۳۰ درخت را تشکیل می‌دهیم. پس از ساخت یک الگوریتم از جنگل تصادفی، داده‌های آزمون را که در مرحله آموزش در اختیار الگوریتم قرار نداده بودیم بدون مقادیر برچسب به جنگل تصادفی ارائه می‌نماییم تا پیش‌بینی توسط الگوریتم در رابطه با نوع حمله انجام گیرد.

در این قسمت از کد از داده‌های ذخیره‌شده در train2 استفاده می‌نماییم که در واقع یک کپی از داده‌های ذخیره‌شده در train است که در ابتدای فرایند آن را ذخیره نموده‌ایم تا در زمان انجام مرحله آزمون داده‌های آموزشی یکسانی را جهت مقایسه با حالت دسته‌بندی با استفاده از الگوریتم ژنتیک داشته باشیم.

شبه کد ۲: الگوریتم ژنتیک

```
Algorithm: GA

Input : fitness_function, train

Initialize(population)

While (not_stop_condition)  do:

        Pop_fitness <-  fitness_function(population)

        Cell_parent <- select_parent(pop_fitness)

        Child <-cross_over(cell_parent)

        Child <- mutation(child)

        Population <- survival_select(population,  child)

End

Return( max_fitness(population))
```

همان‌طور که در شبه کد ۲ دیده می‌شود ورودی الگوریتم ژنتیک تابع برازش و داده‌های آموزش هستند. ابتدا یک جمعیت اولیه به‌صورت تصادفی از کروموزم‌ها یا همان راه‌حل‌های نامزد تولید می‌کنیم. سپس از روی یک جمعیت با ترکیب راه‌حل‌ها یک راه‌حل بهتر به دست می‌آوریم. درنهایت صدتا از نامزدهای برتر را انتخاب کرده و به مرحله بعد می‌فرستیم. این عملیات آن‌قدر تکرار می‌شود تا شرط توقف صادق باشد

در انتهای جنگل تصادفی یک ماتریس که سطرهای آن نمایش‌دهنده پیشگویی الگوریتم و ستون‌های آن نمایش‌دهنده مقدار واقعی برچسب داده است تشکیل می‌شود تا نتیجه پیشگویی الگوریتم با واقعیت مقایسه شود. در واقع ماتریس آشفتگی جدولی است که به تشریح عملکرد طبقه‌بندی‌کنندگان[2] می‌پردازد. منطق انجام این امر به این علت که اعداد حاضر در قطر اصلی این ماتریس نمایش‌دهنده تعداد پیش‌بینی‌های صحیح

[1] Level

[2] Classifier

الگوریتم و هر مقدار غیرصفری خارج از قطر ماتریس نمایشگر تعداد پیش‌بینی اشتباه الگوریتم و تعداد رخ داد آن به انضمام نوع صحیح حمله و نوع تشخیص اشتباه جنگل است. در ادامه جهت نمایش دقت روش پیشنهادی یک نمونه از ماتریس آشفتگی تشکیل‌شده در کد در جدول (۴) ارائه شده است.

جدول ۳. نمایش برشی از دیتا ست Kdd-Cup-99

Protocol Type	Service	Flag	Src_bytes	...	Label
tcp	http	SF	223	...	Normal.
udp	private	SF	105	...	Snmpget Attack.
tcp	http	SF	230	...	Normal.
udp	private	SF	105	...	Normal.
udp	private	SF	105	...	Snmpget Attack.
tcp	smtp	SF	3170	...	Normal.
tcp	http	SF	297	...	Normal.

شبه کد ۳: جنگل تصادفی ترکیب‌شده با الگوریتم ژنتیک

```
Algorithm: GA aided random forest

Input : GA_best_solution,  train, test

Values <-  non_zero_indices(x)

Group <- union(values)

Group_idx = get_group_idx(group,  train)

Train[group_idx] <- new_value

Rfmodel <- randomForrest(train,  train_label)

Test_label <- predict (rfmodel, test)
```

همان‌طور که در شبه کد ۳ دیده می‌شود، در این مرحله یک کروموزوم داریم که تعدادی صفر و یک دارد. معنی اینکه یک مقدار در آرایه یک است این است که ویژگی متناظر وجود دارد. مقادیری از یک ویژگی که اگر با هم ترکیب کنیم و یک مقدار جدید بگذاریم چیزی از دست نمی‌رود (طبق شهادت آنتروپی). از این رو اجتماع مقادیر را می‌گیریم. سپس شماره سطرهایشان را در داده‌های آموزش پیدا می‌کنیم و مقدار ویژگی موردنظر را به یک مقدار ثابت تغییر می‌دهیم. داده آموزش جدید را به الگوریتم جنگل تصادفی می‌دهیم و الگوریتم پیش‌بینی‌کننده از روی آن می‌سازیم. الگوریتم ساخته‌شده را روی داده آزمون هم امتحان می‌کنیم.

عملکرد تمامی روش‌ها و حالات مورد بررسی بر مبنای دقت و زمان در جدول (۴) مورد مقایسه قرار گرفته است. همان‌طور که مشاهده می‌شود، اعمال الگوریتم ژنتیک به‌صورت معناداری بر بهبود تمامی فاکتورهای مقایسه یعنی دقت و زمان تأثیرگذار بوده و در حالت استفاده از یک درخت تصمیم، در حالتی که گروه‌بندی توسط الگوریتم ژنتیک صورت گرفته است، تفاوت معناداری بین تشخیص‌های انجام‌شده در دو حالت استفاده و عدم استفاده از الگوریتم ژنتیک وجود دارد. در واقع در تمامی دفعات اجرا، همواره روش الگوریتم ژنتیک توانسته زمان کمتری را به خود اختصاص دهد که کاملاً قابل‌قبول و مطابق هدف پیش‌بینی شده است.

جدول ۴. مقایسه نتایج حاصله

		درخت تصمیم	۱۰ درخت تصمیم	۲۰ درخت تصمیم	۲۵ درخت تصمیم	۳۰ درخت تصمیم
دقت	-	۰/۶۶۵۴۸	۰/۹۰۵۹	۰/۹۳۱۴	۰/۹۴۹۰	۰/۹۴۷۰
	GA	۰/۹۷۰۲	۰/۹۷۰۵	۰/۹۷۰۶	۰/۹۷۰۵	۰/۹۷۰۵
زمان آموزش (ثانیه)	-	۰/۲۳	۰/۲۸	۰/۳۱	۰/۳۹	۰/۴۲
	GA	۰/۲۲	۰/۲۷	۰/۳۰	۰/۳۳	۰/۳۹
زمان آزمون (ثانیه)	-	۲/۳۷	۰/۶۳	۱۵/۶۵	۱۸/۱۰	۲۲/۵۰
	GA	۲/۶۳	۶/۹۴	۱۲/۸۴	۱۴/۵۰	۱۸/۰۱

در مقایسه با الگوریتم قبلاً گزارش‌شده [۱۳] که از مفهوم بهره اطلاعات استفاده ننموده بود و مبتنی بر درخت تصمیم معمولی بود، روش استفاده شده در این تحقیق حدود ۲/۲۵٪ بهبود در تشخیص درست را نشان می‌دهد. نرخ تشخیص درست در آن گزارش برابر ۹۴/۷۵ درصد بود و در روش پیشنهادی در این مقاله به رقم ۹۷ درصد رسید. در مقایسه با مقاله دیگری [۱۵] که حدود ۲/۲۵ درصد خطا گزارش کرده‌اند نتایج این تحقیق حدود ۰/۷۴ درصد کاهش کیفیت نشان می‌دهد (جدول ۵). در عوض روش پیشنهادی در این تحقیق به دلیل بهره بردن از استخراج ویژگی (به‌جای انتخاب ویژگی)، امکان پیاده‌سازی سریع‌تر را به دست می‌دهد.

جدول ۵. مقایسه با مقالات مشابه

روش پیشنهادی	الگوریتم مقاله [۱۳]	الگوریتم مقاله [۱۵] ۱۰۰ نسل و ۲۰ بار اجرا
٪ ۹۷	٪ ۹۴/۷۵	٪ ۹۷/۷۴

۴. نتیجه‌گیری

ایده اصلی مقاله، گروه‌بندی خودکار مقادیر یک ویژگی (با نوع اسمی و ۶۵ حالت منحصربه‌فرد) و تولید مقادیر جدید با تعداد مقادیر منحصربه‌فرد خیلی کمتر است. این ایده بیش‌ترین کاربرد را در افزایش سرعت روش‌های مبتنی بر درخت دارد. پس اولین نکته سرعت روش پیشنهادی برای هر روش مبتنی بر درخت تصمیم افزایش می‌یابد. در مقایسه دقت دو روش باید داده‌های train و test یکسانی برای هر دو روش به‌کار برده شود و حتی باید تعداد آزمایش‌ها ۱۰، ۲۰ بار تکرار شود تا بتوان دقت دو روش را مقایسه کرد؛ بنابراین، نمی‌توان دقت اشاره‌شده در مقاله را در نظر گرفت. نکته‌ای که باید تأکید شود این است که برای روش‌های مبتنی بر درخت تصمیم، وقتی یک ویژگی اسمی (nominal) تعداد مقادیر منحصربه‌فرد زیادی دارد، زمان یادگیری آن خیلی زیاد خواهد شد و یادگیری کند خواهد بود. با روش پیشنهادی سرعت یادگیری بسیار افزایش یافته است و دقت نیز قابل‌قبول است.

۵. مرجع‌ها

[1] Cumming, I. G.; Wong, F. H. "Digital Processing of Synthetic Aperture Radar Data"; Artech House: London, 2005.

[2] Brown, J.; Anwar, M.; Dozier, G. "An Evolutionary General Regression Neural Network Classifier for Intrusion Detection"; Proc. 25th Int. Conf. Comput. Commun. and Network, Waikoloa, USA, 2016.

[3] Stolfo, S. J.; Fan, W.; Lee, W.; Prodromidis, A.; Chan, P. K. "Cost-Based Modeling for Fraud and Intrusion Detection: Results from the JAM Project"; Proc. Int. DARPA Information Survivability Conference and Exposition 2000, 2, 1130.

[4] Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A. "A Detailed Analysis of the KDD CUP 99 Data Set"; Second IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009.

[5] Kaushik, S. S.; Deshmukh, P. R. "Detection of Attacks in an Intrusion Detection System"; Int. J. Comput. Sci. Information Technol. 2011, 2982-986.

[6] McHugh, J. "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 Darpa Intrusion Detection System Evaluations as Performed by Lincoln Laboratory"; ACM Trans. Inform. Syst. Secur. 2000, 3, 262-294.

[7] Okafor, A. "Entropy Based Techniques with Applications in Data Mining"; Dissertation, University of Florida, 2005.

[8] Lin, L.; Zuo, R.; Yang, S.; Zhang, Z. "SVM Ensemble for Anomaly Detection Based on Rotation Forest"; IEEE 3th Int. Conf. Intelligent Control and Information Processing, 2012.

[9] Malik, A.; Shahzad, W.; Khan, F. "Binary PSO and Random Forests Algorithm for Probe Attacks Detection in a Network"; IEEE Congress on Evolutionary Computation 2011, 662-668.

[10] Bukhtoyarov, V.; Zhukov, V. "Ensemble-Distributed Approach in Classification Problem Solution for Intrusion Detection Systems"; Proc. Int. Conf. Intelligent Data Engineering and Automated Learning 2014, 255-265.

[11] Bahri, E.; Harbi, N.; Huu, H. "Approach Based Ensemble Methods for Better and Faster Intrusion Detection"; Comput. Intell. Secur. Inform. Syst. 2011, 17-24.

[12] Pervez, M.; Farid, D. "Feature Selection and Intrusion Classification in NSL-KDD Cup 99 Dataset Employing SVMS"; 8th Int. Conf. Software, Knowledge Information Management and Applications 2014, 1-6.

[13] Najafi, M.; Rafeh, R. "A New Light Weight Intrusion Detection Algorithm for Computer Networks"; Adv. Defence Sci. Technol. 2017, 10, 191-200.

[14] Mitchell, T. M. "Machine Learning"; Mc-Graw-Hill Companies, Inc. ISBN 0070428077, 1997.

[15] Stein, G.; Chen, B.; Wu, A.; Hua, K. "Decision Tree Classifier for Network Intrusion Detection with GA-Based Feature Selection"; Proceedings of the 43rd annual Southeast regional conference 2005, 2, 136-141.