Computational Intelligence in Electrical Engineering, Vol. 11, No. 1, 2020

Improving Network Intrusion Detection by Identifying Effective Features using Evolutionary Algorithms based on Support Vector Machine

Masoud Sharifian [1], Hossein Karshenas [2], Saeid Sharifian[3]

[1] Dept. of Computer Engineering, University of Shahid Ashrafi Isfahani, Isfahan, Iran
h.karshenas@eng.ui.ac.ir
[2] Dept. of Computer Engineering, University of Isfahan, Isfahan, Iran
sharifianmasoud@yahoo.com
[3] Dept. of Computer Engineering, University of Isfahan, Isfahan, Iran
s.sharifian@eng.ui.ac.ir

Abstract:
The growing use of internet and the existence of vulnerable points in networks have made the use of intrusion detection systems as one of the most important security elements. Intrusion detection is essentially a classification problem and it is the identification of effective features such as important issues in the classification This paper presents a novel method for selecting effective features in network intrusion detection based on an estimation of distribution algorithm that uses a probabilistic dependency tree to identify important interactions between features. To evaluate the performance of the proposed method, the NSL- KDD dataset is used, in which the packets are divided into five normal types and intrusive types of DOS, U2R, R2L and Prob. The performance of the proposed algorithm has been compared alone and in combination with other feature selection algorithms such as forward selection, backward selection and genetic algorithm. Moreover, the effect of algorithm parameters like population size on intrusion detection accuracy is tested. Based on this analysis and also considering the intra-class accuracy of different feature selection methods studied in this paper, an effective subset of features for intrusion detection is identified.

Keywords: Intrusion Detection, Feature Selection, Estimation of Distribution Algorithm (EDA), Dependency Tree, Genetic Algorithms, Support Vector Machine (SVM).

بهبود تشخیص نفوذ در شبکه با شناسایی ویژگی‌های مؤثر بر پایهٔ الگوریتم‌های تکاملی و دسته‌بند ماشین بردار پشتیبان

مسعود شریفیان [۱]، حسین کارشناس [۲]، سعید شریفیان [۳]

۱- دانشکده فنی و مهندسی - دانشگاه شهید اشرفی اصفهانی - اصفهان - ایران
m.sharifian@ashrafi.ac.ir
۲- استادیار، دانشکده مهندسی کامپیوتر - دانشگاه اصفهان - اصفهان - ایران
h.karshenas@eng.ui.ac.ir
۳- دانشکده مهندسی کامپیوتر - دانشگاه اصفهان - اصفهان - ایران
s.sharifian@eng.ui.ac.ir

چکیده: روند رو به رشد استفاده از اینترنت و وجود نقاط آسیب‌پذیر در شبکه، استفاده از سیستم‌های تشخیص نفوذ را به‌عنوان یکی از مهم‌ترین عناصر برقراری امنیت درخور توجه قرار داده است. تشخیص نفوذ در اصل مسئلهٔ دسته‌بندی است و شناسایی ویژگی‌های مؤثر ازجمله موضوعات با اهمیت در دسته‌بندی است. در این مقاله یک روش جدید برای انتخاب ویژگی‌های مؤثر در تشخیص نفوذ در شبکه، مبتنی بر الگوریتم تخمین توزیع ارائه شده است که از درخت وابستگی احتمالاتی برای شناسایی تعاملات بین ویژگی‌ها استفاده می‌کند. به‌منظور ارزیابی عملکرد این الگوریتم از مجموعه دادهٔ NSL-KDD استفاده شده است که در آن، بسته‌ها به پنج دستهٔ نرمال و نفوذهای نوع DOS، U2R، R2L و Prob تقسیم شده‌اند. عملکرد الگوریتم ارائه‌شده به تنهایی و به‌صورت ترکیبی با سایر الگوریتم‌های انتخاب ویژگی، مانند انتخاب پیشرو، انتخاب پسرو و الگوریتم ژنتیک، مقایسه و تأثیر پارامترهای الگوریتم، مانند اندازهٔ جمعیت بر میزان دقت تشخیص نفوذ بررسی شده است. براساس نتایج حاصل از این تحلیل و نیز ترکیب نتایج بررسی میزان دقت درون دسته‌ای حاصل از به‌کارگیری الگوریتم‌های انتخاب ویژگی متفاوت، زیرمجموعه‌ای از ویژگی‌های مؤثر در تشخیص نفوذ شناسایی شده است.

واژه‌های کلیدی: تشخیص نفوذ، انتخاب ویژگی، الگوریتم تخمین توزیع، درخت وابستگی، الگوریتم ژنتیک، ماشین بردار پشتیبان.

۱- مقدمه

هدف از تشخیص نفوذ اجتناب از استفادهٔ غیرمجاز، سوءاستفاده از بانک‌های اطلاعاتی و آسیب‌رسیدن به منابع در شبکه توسط هر دو دستهٔ کاربران داخلی و نفوذگران خارجی است. سیستم‌های تشخیص نفوذ، یکی از عناصر اصلی زیرساخت امنیت، در بسیاری از سازمان‌ها استفاده می‌شوند. این سیستم‌ها شامل مدل‌ها و الگوهای سخت‌افزاری و نرم‌افزاری‌اند که به خودکارکردن فرایند پایش وقایع در شبکه به‌منظور حل مسئلهٔ امنیت می‌پردازد.

هدف از داده‌کاوی، کشف یا تولید روابط موجود میان مشاهدات اولیه و همچنین، پیش بینی مشاهدات به کمک الگوهای به‌دست‌آمده است. چهار مرحلهٔ اصلی داده‌کاوی

[۱] تاریخ ارسال مقاله: ۱۳۹۷/۱۱/۱۷
تاریخ پذیرش مقاله: ۱۳۹۸/۰۷/۱٤
نام نویسندهٔ مسئول: حسین کارشناس
نشانی نویسندهٔ مسئول: ایران، اصفهان، دانشگاه اصفهان، دانشکدهٔ مهندسی کامپیوتر

برای تشخیص نفوذ عبارت‌اند از جمع‌آوری داده‌ها از شبکه با حسگرهای سیستم‌های مانیتورینگ، تبدیل داده‌های خام به داده‌های قابل استفاده در مدل‌های داده‌کاوی، ایجاد مدل داده‌کاوی و تحلیل نتایج. داده‌کاوی بدون نظارت[۱] و با نظارت[۲]، دو روش مرسوم داده‌کاوی است. در روش بدون نظارت، پاسخ کشف می‌شود؛ اما در روش با نظارت، پاسخ مشخص است و باید پاسخ مشاهدات آینده پیش بینی شود. روش داده‌کاوی در این مقاله، در دستهٔ الگوریتم‌های با نظارت قرار می‌گیرد [۱].

پس از جمع‌آوری داده‌ها از شبکه، مجموعهٔ گسترده‌ای از نمونه‌ها با مدل‌های داده‌کاوی، بررسی و به کمک آن، مجموعه آموزشی ایجاد می‌شود. سپس دقت این مدل با یک مجموعه آزمایشی ارزیابی می‌شود. روش‌های متعددی برای دسته‌بندی مطرح شده‌اند؛ نظیر k نزدیک‌ترین همسایه[۳]، درخت تصمیم[٤]، ماشین بردار پشتیبان[٥]، شبکه‌های بیزی و شبکه های عصبی [٤-۲]. در این پژوهش از یک مجموعه دادهٔ استاندارد در زمینهٔ تشخیص نفوذ به نام NSL-KDD استفاده شده است. این مجموعه داده شامل ٤١ ویژگی و ٥ کلاس متفاوت برای مشخص کردن رفتار بسته‌ها در شبکه است که این کلاس‌ها دربرگیرندهٔ یک کلاس نرمال و ٤ کلاس نفوذ شامل حملات DoS، U2R، R2L و Prob هستند.

هدف از انتخاب ویژگی ساده‌سازی داده‌ها، شناسایی و استفاده از ویژگی‌های اساسی است. انتخاب ویژگی در بسیاری از زمینه‌ها ازجمله طبقه‌بندی متن، کاوش داده، شناخت الگو، پردازش سیگنال، تشخیص نفوذ و ... استفاده می‌شود. اهمیت انتخاب ویژگی در دو جنبه بررسی می‌شود. جنبهٔ اول، حذف ویژگی‌های نامناسب و غیرمؤثر و جنبهٔ دوم به‌عنوان مسئلهٔ بهینه‌سازی برای به دست آوردن زیرمجموعهٔ بهینه از ویژگی‌هاست که هدف مدنظر را بهتر برآورده می‌کند [٥].

در این مقاله از یک ماشین بردار پشتیبان چند کلاسه برای تشخیص نفوذ به کمک داده‌های جمع‌آوری‌شدهٔ قبلی استفاده شده است. به‌منظور شناسایی ویژگی‌های مؤثر در تشخیص نفوذ، یک نوع از الگوریتم‌های تخمین توزیع[٦] استفاده شده است که اقدام به مدل‌سازی روابط بین ویژگی‌ها می‌کند. عملکرد این الگوریتم در انتخاب ویژگی برای این مسئله، با الگوریتم‌های پایه‌ای مانند انتخاب پیشرو، انتخاب پسرو و همچنین، الگوریتم ژنتیک استاندارد مقایسه شده است. در ادامه و برای بهبود عملکرد الگوریتم استفاده‌شده، یک الگوریتم ممتیک تخمین توزیع پیشنهاد شده است که از جستجوی تصادفی محلی بهره می‌برد. نتایج بررسی عملکرد الگوریتم‌ها براساس کارایی کلی و جزئی (درون دسته‌ای)، نشان‌دهندهٔ قابلیت خوب این روش در شناسایی ویژگی‌های مؤثر برای تشخیص نفوذ است. در پایان، با جمع‌آوری نتایج به‌دست‌آمده از الگوریتم‌های انتخاب ویژگی مختلف، زیرمجموعه‌ای از مهم‌ترین ویژگی‌ها برای تشخیص نفوذ شناسایی و معرفی شده است.

در بخش دوم این نوشتار، پیشینهٔ پژوهش و کارهای انجام‌شده بررسی شده‌اند. در بخش سوم، نحوهٔ به‌کارگیری الگوریتم تخمین توزیع درخت وابستگی[۷] برای شناسایی ویژگی‌های مؤثر تشریح شده است. در بخش چهارم، نتایج آزمایش‌های انجام‌شده ارائه و توضیح داده شده‌اند. در پایان، نتیجه‌گیری و پیشنهادهایی برای انجام کارهای آتی مطرح شده‌اند.

۲- پیشینهٔ پژوهش

۲-۱- انتخاب ویژگی در دسته‌بندی

تشخیص نفوذ در اصل یک مسئلهٔ دسته بندی است. انتخاب ویژگی ازجمله موضوعاتی است که در دسته بندی شایان توجه قرار می‌گیرد. ارتباط خطی بین تعداد ویژگی‌ها و عملکرد یک دسته‌بند وجود ندارد؛ اما با تجاوز تعداد ویژگی‌ها از یک مقدار مشخص در عملکرد دسته بند تغییر ایجاد خواهد شد. انتخاب ویژگی برای داده‌های با ابعاد زیاد علاوه بر کاهش زمان تشخیص و هزینه، کارایی دسته بند را بهبود می‌دهد [٦ و٧].

در مسائلی که با تعداد زیادی از ویژگی‌ها مواجه می‌شویم، انتخاب ویژگی[۸] یک گام معمول در روش های یادگیری ماشین است. در یکی از روش های متداول انتخاب ویژگی مراحل شامل تولید زیرمجموعه‌ای از ویژگی‌ها، ارزیابی زیرمجموعه، معیار خاتمه و اعتبارسنجی نتایج است.

به‌منظور انتخاب زیرمجموعه‌ای از ویژگی‌ها، فرایند انتخاب و ارزیابی زیرمجموعه‌ها تکرار می‌شود تا شرط خاتمه تحقق یابد.

تولید زیرمجموعه‌ای از ویژگی‌ها اساساً فرایند جستجوی اکتشافی در یک فضای جستجو است. ماهیت این فرایند با دو موضوع اساسی تعیین می شود. ابتدا باید نقطه یا نقاط شروع جستجو تعیین شود که بر جهت جستجو تأثیر می‌گذارد. موضوع دوم، تعیین یک استراتژی جستجو است. استراتژی‌های مختلفی ازجمله کامل، دنباله‌ای و تصادفی برای جستجوی زیرمجموعهٔ بهینه استفاده می‌شوند.

هر زیرمجموعهٔ جدید تولیدشده باید با یک معیار ارزیابی شود. معیار ارزیابی، براساس وابستگی به الگوریتم‌های یادگیری، به دو گروه مستقل و وابسته طبقه بندی می شود. در مدل مستقل، انتخاب زیرمجموعهٔ ویژگی‌ها به طور مستقل از الگوریتم یادگیری انجام می‌شود. در مدل وابسته، از یک الگوریتم یادگیری، به عنوان تابع ارزیاب[9] برای انتخاب زیرمجموعهٔ مناسب استفاده می‌شود.

معیار توقف تعیین می‌کند چه زمانی فرایند انتخاب ویژگی باید متوقف شود. بعضی معیارهای توقف عبارت‌اند از تعیین برخی محدودیت‌ها که نباید نقض شوند، تعیین حداکثر تعداد تکرار، توقف فرایند انتخاب ویژگی در زمانی که اضافه‌کردن ویژگی‌ها منجر به تولید زیرمجموعهٔ بهتر نشود یا توقف در صورتی که یک زیرمجموعه به اندازهٔ کافی خوب انتخاب شده باشد.

روشی ساده برای اعتبارسنجی نتایج، نتیجه گیری با استفاده از دانش قبلی است؛ اما در برنامه های دنیای واقعی، معمولاً چنین دانشی وجود ندارد؛ ازاین رو، باید از برخی روش‌های غیرمستقیم با نظارت بر تغییر عملکرد استفاده کنیم.

دو الگوریتم پایه برای انتخاب ویژگی، انتخاب پیشرو و پسرو هستند. فرایند انتخاب ویژگی در روش انتخاب پیشرو با یک مجموعهٔ خالی شروع می‌شود و در هر مرتبه تکرار الگوریتم یک ویژگی به مجموعه جواب، اضافه و با استفاده از تابع ارزیاب ارزیابی می‌شود. این کار تا انتخاب تعداد ویژگی‌های لازم تکرار می‌شود. مشکلی که الگوریتم انتخاب پیشرو با آن روبه‌رو است، حذف‌نشدن ویژگی اضافه‌شده در صورت نامناسب‌بودن از مجموعه جواب است.

الگوریتم انتخاب پسرو برخلاف الگوریتم انتخاب پیشرو کارش را با مجموعه‌ای شامل تمام ویژگی‌ها آغاز می‌کند و در هر تکرار الگوریتم، ویژگی انتخاب‌شده با تابع ارزیاب، از مجموعه ویژگی‌ها حذف می‌شود. این عمل تا زمانی ادامه می یابد که حذف هیچ ویژگی بهبودی حاصل نکند. ویژگی های حذف‌شده از مجموعه در این روش، حتی درصورت مناسب‌بودن، دیگر به مجموعه اضافه نمی‌شوند.

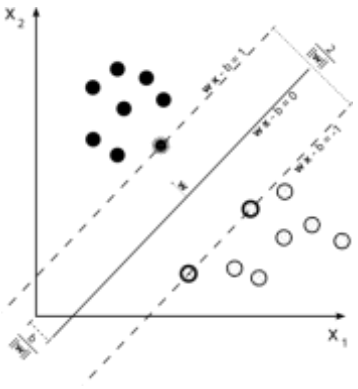
۲-۲- دسته‌بند ماشین بردار پشتیبان

ماشین بردار پشتیبان، یک دسته‌بند دودویی است که دو کلاس را با استفاده از یک مرز خطی از هم جدا می‌کند. هدف در تقسیم خطی داده‌ها، دست‌یابی به تابعی است که تعیین‌کنندهٔ ابرصفحه‌ای با بیشترین حاشیه باشد.

فرض کنید مجموعه دادهٔ آموزشی شامل n نمونه به‌صورت زیر باشد:

$$D=\{(x_i,y_i)\,|\,x_i\in R^p, y_i\in\{-1,1\}\}_{i=1}^{n} \qquad (۱)$$

مقدار y برابر ۱ یا ۱- و هر x_i یک بردار حقیقی p بعدی است. نزدیک‌ترین داده‌های آموزشی به ابرصفحه‌های جداکننده، بردارهای پشتیبان نامیده می‌شوند.



شکل (۱): ابرصفحهٔ جداکنندهٔ دو کلاس ۱+ و ۱-

براساس این، هدف، پیداکردن ابرصفحهٔ جداکننده ای با بیشترین فاصله از بردارهای پشتیبان است که نقاط با $y_i = 1$ نقاط با $y_i = -1$ جدا کند. مطابق با شکل (۱)، با حداکثرشدن حاشیهٔ ابرصفحه، تفکیک بین دسته ها حداکثر می‌شود.

هر ابرصفحه را به‌صورت مجموعه‌ای از نقاط x می‌توان نوشت که شرط $w.x-b=0$ را برآورده می‌کند؛ w بردار نرمالی است که بر ابرصفحه عمود است. باید w و b طوری انتخاب شوند که بیشترین فاصله بین ابرصفحه‌های موازی ایجاد شود که داده‌ها را از هم جدا می‌کنند. این ابرصفحه‌ها با استفاده از رابطهٔ (۲) توصیف می‌شوند:

$$w.x-b=1 \quad w.x-b=-1 \qquad (۲)$$

w بردار وزن متعامد بر ابرصفحه‌های جداکننده و b بردار عرض از مبدأ هر ابرصفحه است. معادلهٔ اول، ابرصفحهٔ جداکنندهٔ نمونه های مثبت و معادلهٔ دوم، ابرصفحهٔ جداکنندهٔ نمونه‌های منفی است. در ماشین بردار پشتیبان به دو روش خطی و غیرخطی می توان مجموعه نقاط را از یکدیگر جدا کرد. اگر داده‌های آموزشی جدایی‌پذیر خطی باشند، می توان دو ابرصفحه در حاشیهٔ نقاط را طوری در نظر گرفت که هیچ نقطهٔ مشترکی نداشته باشند؛ سپس سعی شود فاصلهٔ آن ابرصفحه‌ها بیشینه شود. زمانی که داده‌ها را بتوان به صورت خطی از هم جدا کرد، ماشین بردار پشتیبان با در نظر گرفتن مجموعه داده های آموزشی، با استفاده از حل یک مسئلهٔ بهینه سازی، ابرصفحهٔ بهینه با حاشیهٔ حداکثر را به دست می‌آورد.

در صورتی که داده‌ها جداناپذیر خطی باشند و کلاس‌ها همپوشانی داشته باشند، جداسازی کلاس‌ها با مرز خطی همواره با بروز خطا همراه می‌شود. به‌منظور حل این مسئله ابتدا داده ها با استفاده از یک تبدیل غیرخطی، از فضای اولیه به فضایی با ابعاد بالاتر منتقل می‌شوند؛ با این هدف که در فضای جدید، کلاس‌ها تداخل کمتری با یکدیگر داشته باشند. انتقال به ابعاد بالاتر با توابع هسته انجام می‌شود. توابع هستهٔ متفاوتی به این منظور معرفی شده‌اند؛ مانند تابع چندجمله‌ای، تابع پایه شعاعی و ... [۸ و ۹].

۳-۲- الگوریتم‌های تکاملی

الگوریتم های تکاملی یک رویکرد تصادفی و مبتنی بر تولید و آزمایش برای حل مسائل بهینه‌سازی‌اند. الگوریتم ژنتیک از پایه‌ای‌ترین انواع الگوریتم‌های تکاملی است که پژوهشگران به آن توجه می‌کنند. این الگوریتم از نظریهٔ انتخاب طبیعی[۱۰] و بازترکیب ژنتیک[۱۱] الهام گرفته است. در این الگوریتم با بازترکیب جواب های امیدبخش[۱۲]، سعی می شود جواب بهینهٔ مسئله پیدا شود. استفاده از این روش در دامنهٔ متنوعی از مسائل به نتایج خوبی منجر شده است؛ اما در برخی مواقع، انتخاب و بازترکیب ساده برای رسیدن به پاسخ بهینه مؤثر نیست. این مورد بیشتر در مواقعی رخ می‌دهد که بلوک های ساختاری[۱۳] پاسخ بهینه در فضای جستجو به سستی توزیع شده باشند. این موضوع به علت حفظ نشدن مؤثر بلوک های ساختاری یا جواب های جزئی است که در راه حل ها به وجود می‌آید. در اصطلاح به جواب های زیرمسئله ها که نمایان‌کنندهٔ دانش و روابط حاکم بر ابعاد مسئله است، بلوک های ساختاری می‌گویند. جستجو برای یافتن تکنیکی که از بلوک های ساختاری محافظت بیشتری کند، به ظهور کلاس جدیدی از الگوریتم های تکاملی به نام الگوریتم های تخمین توزیع منجر شده است [۱۰]. در بخش سوم، این نوع از الگوریتم‌ها بیشتر توضیح داده خواهند شد.

۱-۳-۲- الگوریتم ژنتیک

الگوریتم ژنتیک، در طی مرحلهٔ تولیدمثل از عملگرهای ژنتیک استفاده می‌کند. عملگرهای انتخاب[۱۴]، ترکیب[۱۵] و جهش[۱۶] بیشترین کاربرد را در الگوریتم های ژنتیک دارند. استفاده از این عملگرها روی یک جمعیت، از بین رفتن پراکندگی یا تنوع ژنتیک جمعیت را موجب می‌شود. روند کلی الگوریتم ژنتیک به شرح زیر است:

در مرحلهٔ اول، یک جمعیت اولیه از راه‌حل‌های کاندید تولید می‌شود که کروموزوم نامیده می‌شوند. سپس انتخاب جمعیت والدین از جمعیت اولیه انجام می‌شود که از یک تابع ارزیاب به این منظور بهره گرفته می‌شود. پس از آن، اعمال عملگرهای ترکیب و جهش روی جمعیت والدین و تولید جمعیتی از راه حل های جدید به نام جمعیت فرزندان است. در پایان، جمعیت راه حل های جدید با جمعیت اولیه ترکیب می‌شود و جمعیت اولیهٔ نسل بعد ایجاد می‌شود [۲٤]. این روند تا محقق‌شدن یکی از شرایط توقف ادامه می‌یابد.

۲-۳-۲- الگوریتم‌های تخمین توزیع

در الگوریتم های تخمین توزیع، از حذف جواب های جزئی موجود در کروموزوم در حد امکان پیشگیری می‌شود. درواقع، با دادن احتمال زیاد به بلوک های ساختاری سعی می‌شود این بلوک ها در نسل فرزندان ظاهر شوند. به این منظور، به جای استفاده از عملگرهای استاندارد ژنتیک، از تخمین توزیع احتمال جواب‌های امیدبخش برای تولید جواب‌های کاندیدا استفاده می‌شود. در هر مرحله از الگوریتم، یک مدل احتمالی براساس جواب های برگزیده جمعیت ساخته می‌شود و نسل بعد از راه‌حل‌های کاندیدا[۱۷] با نمونه‌گیری از این مدل تولید می‌شود؛ بنابراین، الگوریتم‌های تخمین توزیع، الگوریتم های ژنتیک مبتنی بر مدل احتمالی نامیده می شوند که در آن، دو عملگر ترکیب و جهش با ساخت مدل احتمالی و نمونه‌گیری از مدل ساخته‌شده جایگزین شده‌اند [۱۱].

در حالت کلی، الگوریتم‌های تخمین توزیع براساس مدل احتمالی استفاده‌شده و تعداد وابستگی بین ژن‌ها به سه دستهٔ یک متغیره[۱۸]، دو متغیره[۱۹] و چند متغیره[۲۰] تقسیم می شوند. تفاوت در این سه مدل براساس تعداد وابستگی هر متغیر به متغیرهای دیگر است. الگوریتم های این دسته هیچ وابستگی بین ژن‌ها در نظر نمی‌گیرند؛ درواقع، بلوک های ساختاری از مرتبه اول‌اند و توزیع احتمال آنها از ضرب احتمالات حاشیه‌ای تمام متغیر ها در هر فرد محاسبه می‌شود؛ ازجمله معروف‌ترین این الگوریتم‌ها، به الگوریتم توزیع حاشیه‌ای یک متغیره[۲۱]، الگوریتم جمعیتی براساس یادگیری افزایشی[۲۲] و الگوریتم ژنتیک متراکم[۲۳] اشاره می‌شود.

در بسیاری از مسائل، بیشتر متغیرها به نحوی با یکدیگر مرتبط‌اند. در مدل دو متغیره، الگوریتم قادر به ضبط برخی از تعاملات دوتایی بین متغیرها با استفاده از ساختارهایی مانند درخت است. در مدل های مبتنی بر درخت، یک متغیر ممکن است با بیش از یک متغیر دیگر ارتباط داشته باشد که به‌صورت فرزندان آن در یک ساختار درختی قرار می گیرند. این الگوریتم ها قادر به مدل سازی ارتباطات با درجهٔ دو بین ژن‌های مسئله‌اند؛ بنابراین، مدل توزیع احتمال نسبت به مدل یک متغیره قدری پیچیده‌تر خواهد شد و فرمی شبیه شبکه احتمالاتی را بین متغیرها به وجود می‌آورد. الگوریتم بیشینه‌سازی اطلاعات دوطرفه برای خوشه بندی ورودی[۲۴] (*MIMIC*) و الگوریتم ترکیب بهینه‌سازها با درخت اطلاعات دوطرفه[۲۵] (*COMIT*) نمونه‌ای از این الگوریتم ها هستند. نتایج تجربی نشان دهندهٔ کارآیی بهتر الگوریتم *COMIT* در مقایسه با *MIMIC*، *PBIL* و *GA* بوده‌اند. الگوریتم تخمین توزیع درخت وابستگی استفاده شده در این پژوهش، شبیه به این الگوریتم است.

در الگوریتم‌های تخمین توزیع مبتنی بر مدل‌های چند متغیره امکان مدل سازی درجات بالاتری از ارتباطات بین متغیرها وجود دارد. الگوریتم ژنتیک متراکم توسعه یافته[۲۶] و الگوریتم بهینه‌سازی بیزی نمونه‌ای از این نوع الگوریتم‌ها هستند [۱۱ و ۱۲].

بزرگ‌ترین مشکل این الگوریتم‌ها پیچیدگی بالا و زمان‌بربودن فرایند مدل‌سازی به‌علت پیچیدگی مدل‌های احتمالی به کار گرفته شده است. با توجه به اینکه روند ارزیابی زیرمجموعه‌های ویژگی انتخاب‌شده با روش دسته‌بندی خود، پیچیدگی زیادی دارد، در این پژوهش از الگوریتم دو متغیره مبتنی بر درخت وابستگی برای جستجوی فضای زیرمجموعه های ممکن استفاده شده است که در ادامه توضیح داده خواهد شد.

۳-۳-۲- مدل احتمالی درخت وابستگی

در یک مدل احتمالی، احتمال مقادیر مختلف برای متغیرهای مسئله به دست می‌آید. هر مدل احتمالی شامل یک ساختار و تعدادی پارامتر است. در ساختار مدل احتمالی، وابستگی متغیرها و در پارامترهای مدل احتمالی، مقدار احتمال این وابستگی‌ها مشخص می‌شود. در صورتی که متغیرها باینری باشند، مدل احتمالی برای هر متغیر احتمال مقادیر صفر و یک را نشان می‌دهد. نحوهٔ محاسبهٔ مقدار احتمال ها از روی راه‌حل های موجود در جمعیت والدین است.

درخت وابستگی یک مدل دو متغیره از نوع درختی است و ارتباط هر متغیر به شرط یک متغیر دیگر را به دست می‌آورد. این درخت، درختی جهت دار است که گره ها در آن، متغیرهای مسئله‌اند. توزیع احتمال تعریف‌شده با ساختار درختی مطابق رابطهٔ (۳) است:

$$P(\mathrm{x}) = \prod_{i=1}^{n} P(\mathrm{x}_i \mid \mathrm{x}_j) \qquad (۳)$$

در رابطهٔ بالا x_j به پدر x_i اشاره دارد و در زمانی کـه i ریشه است، مقدار $\mathrm{p}(x_i \mid x_j)$ معادل $\mathrm{p}(x_i)$ خواهد بود.

برای یـادگیری درخـت وابسـتگی از روی جمعیـت در ابتدا آنتروپی تک‌تک متغیرها به دست می‌آید و بی‌نظم‌تـرین متغیر به‌عنوان ریشه مشخص می‌شود. سپس براساس معیـار اطلاعات متقابل[۲۷] یـک مـاتریس وابسـتگی بـین متغیرهـای مسئله تشکیل می‌شود که به تعداد متغیرهای مسئله، سـطر و ستون دارد و در هر خانهٔ آن، اطلاعات متقابل بین دو متغیـر محاسبه شده است. در مرحلهٔ بعد با یک الگـوریتم سـاخت درخت پوشای ماکسیمم، بار متغیری انتخاب می‌شود کـه بـا متغیرهای اضافه‌شده به درخت بیشترین ارتباط را دارد و بـا یک یال به درخت وابستگی اضافه می شـود؛ ایـن عمـل تـا افزودن تمام متغیرها به درخت ادامـه مـی یابـد. در انتهـا بـا استفاده از یک روش تخمین مونت کارلو احتمال حاشـیه ای متغیـر ریشـه و احتمـال شـرطی سـایر متغیرهـا بـه شـرط والدینشان با توجه به جمعیت راه‌حل‌های امیدبخش محاسبه می‌شود.

راه‌حل‌های جدید (فرزندان) با استفاده از مدل احتمـالی فرا گرفته شده به‌صورت مستقل از هم تولید مـی‌شـوند؛ بـه این منظور، از ساختار مدل احتمالی استفاده می‌شود. ریشـه، نخستین متغیری است که برای آن مقدار تولید می‌شود؛ زیرا ریشه به هیچ متغیری وابسته نیست. احتمال مقادیر مختلـف متغیر ریشه در قسمت پارامترهای مـدل احتمـالی مشـخص شده است که با توجه به آن، به‌صورت تصادفی یـک مقـدار تولید می شود و در جـای آن متغیـر در نمونـهٔ جدیـد قـرار می‌گیرد. این روند برای تمام متغیرهای دیگر تکرار می‌شـود تا به‌ازای تمام متغیرها یک مقدار تولیـد شـود. بـرای تولیـد راه‌حل بیشتر باید روند بالا با شـروع از ریشـه تکـرار شـود [۱۱].

٤-۲- کارهای انجام‌شده

بیشتر سیستم های تشخیص نفوذ عمدتاً از یک الگوریتم دسته‌بندی برای تشخیص نفوذ اسـتفاده مـی کننـد؛ امـا ایـن سیستم ها تنها موفق به ارائه احتمال بهترین تشخیص نفوذ با نرخ هشدار غلط کم‌اند. پژوهش‌های متنـوعی روی اسـتفاده از تکنیک‌های داده‌کاوی به‌منظور تشخیص نفوذ انجام شـده است. هریک از این پژوهش ها به دنبال ارائه نتـایج بهتـر در دستیابی به الگوهای مفید در سیستم‌های تشخیص نفوذند. از تکنیک‌های داده‌کاوی به‌کاررفته در این خصوص بـه مـوارد زیر اشاره می‌شود:

وانگ و همکاران [۱۳] به‌منظور افزایش دقت و پایداری در تشخیص حملات کم تکرار که با پایین‌آوردن نرخ مثبـت غلط[۲۸] محقق شده است، روش ترکیب شبکه های عصـبی و خوشه بندی فازی را ارائه دادند. ابتدا کل مجموعه یـادگیری با استفاده از روش خوشه‌بندی فازی به زیرمجموعه‌هـای بـا تعداد کمتر، شکسته و روی هریک از زیرمجموعـه هـا یـک شبکهٔ عصبی مناسـب اعمـال مـی‌شـود. هـر شـبکهٔ عصـبی می‌تواند هریک از زیرمجموعه‌ها را سریع‌تر و دقیـق‌تـر یـاد بگیـرد و درنهایـت، بـا اسـتفاده از روش تجمـع فـازی[۲۹]، خروجی اصلی را از خروجی همـهٔ شـبکه هـای عصـبی بـه دست می‌آورند.

چن و ابراهام [۱٤] به‌منظور آمـوزش دسـته‌بنـد شـبکهٔ عصبی پیشرو[۳۰] برای تشـخیص نفـوذ از الگـوریتم تخمـین توزیع اسـتفاده کـرده‌انـد؛ بـه‌طـوری‌کـه وزن‌هـا، بایـاس و پارامترهای تابع استفاده‌شـده در شـبکهٔ عصـبی،ماننـد تـابع گاوسی یا سیگموید، با الگوریتم تخمین توزیع بهینـه سـازی می‌شوند. در این مقاله نیز دسته‌بند شبکهٔ عصبی با الگـوریتم بهینه‌سازی ازدحام ذرات[۳۱] آموزش داده شده و مقایسهٔ نتایج نشان‌دهنـدهٔ دقـت بـالا و نـرخ مثبـت غلـط بهتـر در روش آموزش شبکهٔ عصبی با الگوریتم تخمین توزیع است.

سوناوان و همکاران [۱۵] برای تشخیص نفوذ مبتنی بـر سوءاستفاده، دو روش بر پایهٔ شـبکهٔ عصـبی ارائـه داده انـد. نخستین روش، استفاده از شبکهٔ عصبی با داده هـای کمتـر و استفاده از تکنیـک آنـالیز اجـزای اصـلی[۳۲] و دومـین روش، استفاده از شبکهٔ عصبی با همهٔ ویژگی‌های پایگاه داده است. بر طبق نتایج به‌دست‌آمده، به‌کارگیری ویژگی‌هـای کمتـر در پایگاه دادهٔ *KDDCUP99* پارامترهای زمـان و حافظـهٔ لازم برای تشخیص نفوذ را بهبود می‌بخشد.

در کار مشابه دیگر [۱٦] یـک سیسـتم کشـف نفـوذ بـا استفاده از الگوریتم آنالیز اجزای اصلی بـرای کـاهش تعـداد

ویژگی‌ها به منظور پایین‌آوردن پیچیدگی سیستم و استفاده از ماشین بردار پشتیبان برای دسته‌بندی‌کردن نمونه‌ها معرفی شده است. سیستم پیشنهادی، سرعت پردازش کشف نفوذ را بالا برده و فضای حافظهٔ لازم را به مراتب کاهش داده است.

به منظور شناسایی رفتار نرمال در [۱۷] از خوشه بندی استفاده شده است. رفتارهای نرمال به‌صورت خوشهٔ نرمال گروه‌بندی می‌شوند، از خوشه های نرمال به‌عنوان امضا برای تشخیص نفوذ استفاده می‌شود و هرگونه انحراف از آن، نفوذ در نظر گرفته می‌شود.

در [۱۸]روش ترکیبی به نام *FWP-SVM-GA* پیشنهاد شده است. در این الگوریتم ابتدا احتمال عملگرهای ترکیب و جهش در الگوریتم ژنتیک با توجه به وضعیت تکاملی جمعیت و ارزش برازندگی بهینه، محاسبه و از آن برای انتخاب ویژگی استفاده می شود. نوآوری این روش نحوهٔ محاسبهٔ تابع برازندگی است که برای هر زیرمجموعه از ویژگی‌ها سه پارامتر نرخ مثبت صحیح (*TPR*)، نرخ خطا (*Error*) و تعداد ویژگی‌های انتخاب شده (NumF(S)) را با هم ترکیب می کند؛ درنهایت، با توجه به زیرمجموعه ویژگی‌های بهینه، وزن‌های ویژگی و پارامترهای *SVM* به‌طور هم‌زمان بهینه می‌شوند.

در کار مشابهی، جاناتان و ماندالا [۱۹] از یک الگوریتم بازگشتی حذف ویژگی ها، مانند روش انتخاب پسرو برای شناسایی ویژگی های مرتبط در تشخیص نفوذ استفاده کردند. آنها دو دسته بند ماشین بردار پشتیبان با هستهٔ گاوسی و نزدیک‌ترین همسایه را برای تصمیم‌گیری دربارهٔ حذف متغیرها استفاده کرده‌اند. برای بهبود دقت دسته بندی با هر زیرمجموعه از ویژگی ها، با روش تنظیم پارامتر، مقادیر مناسب برای پارامترهای الگوریتم‌های دسته‌بندی نیز به دست آمده‌اند.

در کار دیگری [۲۰] ابتدا با روش های انتخاب ویژگی مبتنی بر همبستگی و Chi-Square به‌ترتیب ۱۷ و ۳۵ ویژگی مرتبط از مجموع ۴۱ ویژگی موجود در پایگاه دادهٔ NSL-KDD انتخاب شده‌اند؛ در ادامه، از مدل‌های دسته‌بندی ماشین بردار پشتیبان و شبکه‌های عصبی تشخیص نفوذ براساس ویژگی‌های انتخاب‌شده استفاده شده است.

مانکار و واگمار در [۲۱]، با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات، مقادیر مناسب برای پارامتر های C (هزینه) و g (گاما) در دسته بند ماشین بردار پشتیبان را پیدا کرده و هم‌زمان از همان الگوریتم برای انتخاب زیرمجموعه ویژگی‌های مرتبط در مسئلهٔ تشخیص نفوذ نیز استفاده کرده‌اند.

۳- شناسایی ویژگی‌های مؤثر

در این پژوهش از الگوریتم تخمین توزیع درخت وابستگی برای انتخاب ویژگی ها استفاده شده است. در این الگوریتم هر فرد در جمعیت راه‌حل‌ها نشان‌دهندهٔ یک زیرمجموعه از ویژگی‌هاست. برای ارزش‌گذاری هر راه‌حل از جمعیت از دسته‌بند ماشین بردار پشتیبان استفاده می‌شود. پس از انتخاب راه‌حل‌های امیدبخش جمعیت، از آنها به‌عنوان یک مجموعه داده برای آموزش مدل احتمالی درخت وابستگی با روش توضیح داده شده در بخش ۲-۳-۳ استفاده می‌شود. راه‌حل‌های جدید تولیدشده، پس از ارزش‌گذاری در جمعیت اصلی، جایگزین راه‌حل‌های بدتر قبلی می شوند. جزئیات نحوهٔ نمایش زیرمجموعه ویژگی‌ها در الگوریتم و نحوهٔ استفاده از ماشین بردار پشتیبان برای ارزیابی راه‌حل‌ها در ادامه توضیح داده شده‌اند.

۱-۳- تابع ارزیاب و کدگذاری راه‌حل‌ها

ارزیابی هر زیرمجموعه از ویژگی ها براساس یک دسته‌بند ماشین بردار پشتیبان چند کلاسه انجام گرفته است. به این منظور، در ابتدا این دسته بند با یک مجموعه داده‌های آموزشی آموزش داده می شود که براساس زیرمجموعه ویژگی های داده‌شده فیلتر شده است. سپس با مجموعه داده های آزمایشی ارزیابی می‌شود که به‌طور مشابه فیلتر شده اند. برای هر کلاس یک دسته بندی جداگانه آموزش داده می‌شود (رویکرد one-vs-all). درنهایت، میانگین عملکرد دسته بندی ماشین های بردار پشتیبان مختلف روی مجموعه داده‌های آزمایشی، ملاک ارزش‌گذاری زیرمجموعه ویژگی‌ها است.

برای کدگذاری هر زیرمجموعه از ویژگی ها، مانند کارهای مشابه قبلی از یک رشته دودویی به طول تعداد

ویژگی‌ها استفاده می‌شود که در آن، عدد صفر به معنی انتخاب‌نشدن ویژگی مربوطه و عدد یک به معنی انتخاب ویژگی در زیرمجموعه است.

۲-۳- ترکیب با جستجوی محلی

الگوریتم‌های ممتیک دسته‌ای از الگوریتم‌های فراابتکاری‌اند که از ترکیب روش‌های ابتکاری مانند جستجوهای محلی با جستجوگرهای پایه مانند الگوریتم‌های تکاملی به دست می‌آیند و به بهبود عملکرد الگوریتم جستجوی پایه مانند کاهش زمان دستیابی به پاسخ بهینه منجر می‌شوند. [۲۲]. معمولاً الگوریتم‌های تکاملی برای جستجوی سراسر فضای جستجو ایجاد می‌شوند؛ درحالی که جستجوی محلی حوزهٔ همسایگی، هر پاسخ یافته‌شده با الگوریتم تکاملی را برای یافتن پاسخ های بهتر جستجو می کند. انتخاب عملگرهای تولید نسل در یک الگوریتم ممتیک و نوع و روش جستجوی محلی استفاده شده در آن، به نتایج اجرای بسیار متفاوت منجر خواهد شد. به این منظور، در این مقاله یک الگوریتم جستجوی محلی استفاده شده است که با دریافت راه‌حل به‌دست‌آمده با الگوریتم تخمین توزیع، مجاورت آن را بررسی می‌کند. این الگوریتم با یافتن زیرمجموعه مجاوری که برازندگی بیشتری دارد، آن را انتخاب می‌کند و این کار را تا جای ممکن ادامه می‌دهد؛ درنهایت، بهترین راه‌حل پیداشده را جایگزین راه‌حل فعلی می‌کند.

٤- نتایج شبیه‌سازی

۱-٤- مجموعه داده‌های *NSL-KDD*

در مجموعه دادهٔ *NSL-KDD* هر رکورد شامل ٤۳ فیلد است. ٤۱ ویژگی، یک فیلد رفتاربسته که مشخص‌کنندهٔ رفتار نرمال یا نوع نفوذ است و فیلد آخر نمایش‌دهندهٔ درجهٔ سختی تشخیص نفوذ است. ستون برچسب، ۵ دسته دارد که یک کلاس نرمال و ٤ کلاس نفوذ شامل *DoS*، *U2R*، *R2L* و *Prob* است. در حملهٔ انکار سرویس با اشباع کردن ماشین هدف با درخواست ارتباط، سربار زیاد روی سرور، ایجاد و مانع از پاسخگویی سرور به ترافیک قانونی در شبکه می‌شود. در حملهٔ کاربر به ریشه، مهاجم با یک حساب کاربری عادی برای سوءاستفاده از آسیب‌پذیری سیستم به‌منظور به دست آوردن امتیازات ریشه تلاش می‌کند. در نفوذ خارج به داخل، مهاجم قابلیت ارسال بسته‌ها به یک ماشین را دارد؛ اما هیچ شناسه ای روی ماشین ندارد و مانند یک کاربر نمی‌تواند از دسترسی بر سیستم بهره‌برداری کند. در نفوذ پویشی مهاجم، ماشین را به‌منظور تعیین نقاط ضعف یا آسیب پذیری که ممکن است بعدها بهره‌برداری شود، پویش می‌کند. به این صورت، فهرستی از قابلیت‌های آسیب پذیری بالقوهٔ یک ماشین به دست می‌آید که برای انجام یک حمله می‌تواند استفاده شود.

ویژگی‌ها در این مجموعه داده به‌صورت داده‌های عددی و متنی در سه دستهٔ پایه ای، محتوایی و ترافیکی تقسیم‌بندی شده‌اند[۲۳].

ویژگی‌های پایه ای شامل ویژگی‌هایی است که از یک ارتباط پروتکل TCP/IP استخراج می‌شود. این ویژگی‌ها باعث تأخیر در فرایند تشخیص نفوذ می‌شوند. نمونه‌هایی از این ویژگی‌ها، مدت زمان اتصال، نوع پروتکل و سرویس استفاده شده و بایت های ارسالی و دریافتی در یک اتصال است.

ویژگی های محتوایی: برخلاف بسیاری از حملات جلوگیری از سرویس و پویشی، حملات خارج به داخل و کاربر به ریشه، الگوی ترتیبی تکرار ناهنجاری ندارند؛ به این علت که برخلاف حملات جلوگیری از سرویس و پویشی که اتصالات بسیاری به میزبان ها طی دورهٔ زمانی کوتاه دارند، نفوذ خارج به داخل و کاربر به ریشه در بخش دادهٔ بسته‌های شبکه تعبیه می‌شوند و عموماً یک تک اتصال دارند. برای تشخیص این نوع حملات، به ویژگی هایی نیاز است که قادر باشند در بخش دادهٔ بسته ها رفتار نفوذ را جستجو کنند، مانند تعداد تلاش هایی که به شکست منجر شده‌اند؛ این ویژگی‌ها، ویژگی‌های محتوایی نامیده می‌شوند. نمونه‌هایی از این ویژگی‌ها شامل مجموع عملیات انجام‌شده در یک اتصال، تعداد ورودهای ناموفق در یک اتصال، دستیابی کاربر به‌عنوان مدیر به سیستم و ... است.

ویژگی‌های ترافیکی شامل ویژگی‌هایی‌اند که با توجه به اندازهٔ پنجره محاسبه شده‌اند و به دو گروه تقسیم می‌شوند. یک گروه اتصالاتی‌اند که در دو ثانیه گذشته با

اتصال فعلی دارای سرویس و میزبان مشابه بوده اند و مبتنی بر زمان نامیده می شوند. گروه دیگر برای ارزیابی حملاتی در نظر گرفته شده‌اند که در بازهٔ بیشتر از دو ثانیه رخ می دهند. اینها ویژگی‌هایی اند که در آنها درصد اتصال گذشته نسبت به اتصال فعلی تعیین می شوند که سرویس و میزبان مشابه داشته اند و مبتنی بر ماشین نام دارند. این گروه برای ارزیابی حملاتی در نظر گرفته شده‌اند که در بازهٔ بیشتر از دو ثانیه رخ می‌دهند.

۲-٤- مفروضات شبیه‌سازی

پیاده‌سازی الگوریتم تخمین توزیع درخت وابستگی به زبان C انجام شده است و سپس به‌صورت یک تابع Mex، امکان اجرای آن در محیط متلب (MATLAB) فراهم شده است. همچنین، قسمت های مربوطه برای فراخوانی تابع ارزیاب به آن افزوده شده است. سایر مقادیر پیش فرض شبیه‌سازی در جدول (۱) نشان داده شده‌اند.

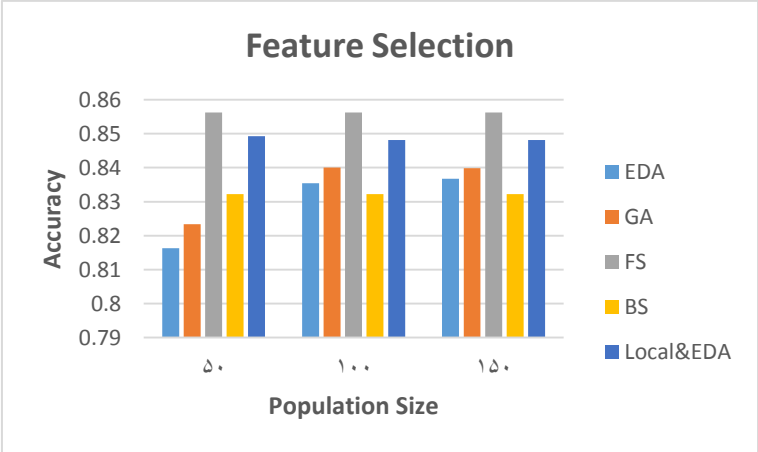
قبل از انجام آزمایش ها مجموعه دادهٔ NSL-KDD پیش‌پردازش شده است و داده‌های آن نرمالیزه شده‌اند. همچنین، برای آموزش دسته‌بند ماشین بردار پشتیبان لازم است داده های غیرعددی به داده‌های عددی تبدیل شوند. سپس الگوریتم های انتخاب ویژگی ژنتیک، تخمین توزیع، تخمین توزیع به همراه جستجوی محلی، انتخاب پیشرو و انتخاب پسرو اجرا شده‌اند.

جدول (۱): مفروضات شبیه‌سازی

Feature Selection Strategy: DT-EDA, GA, Forward and Backward Selection, DT-EDA and Local Search	
Feature Fitness strategy: SVM	Data Set: NSL-KDD
Standardize :0-1	SVM type: one vs. All
population_size=50,100,150	Kernel Function: RBF
problem_size=42	Number of Packets in train Data Set:25192
max_generations=10	
Selection Operator : Binary Tournament selection	Number of Packets in test Data Set: 4507

٤-۳- نتایج به‌دست‌آمده

در این بخش، نتایج حاصل از آزمایش ها روی پایگاه دادهٔ NSL-KDD آمده است. در شکل (۲)، عملکرد پنج روش انتخاب ویژگی با استفاده از ماشین بردار پشتیبان به‌عنوان تابع ارزیاب در اندازه های جمعیت ۵۰، ۱۰۰ و ۱۵۰ مقایسه شده است. همانطور که مشاهده می‌شود، الگوریتم های انتخاب پیشرو و انتخاب پسرو بدون جمعیت است و افزایش جمعیت در عملکرد آنها تأثیر ندارد. مطابق نتایج به دست‌آمده، در اندازه های جمعیت کوچک‌تر، الگوریتم ژنتیک عملکرد بهتری نسبت به الگوریتم تخمین توزیع داشته است و این تفاوت میزان دقت با افزایش اندازهٔ جمعیت کاهش می یابد. به کارگیری ترکیب الگوریتم تخمین توزیع با جستجوی محلی، عملکرد آن را در اندازه های جمعیت کوچک نیز به‌طور چشمگیری بهبود بخشیده است.



شکل (۲): مقایسهٔ میزان دقت تشخیص نفوذ با اجرای الگوریتم‌های انتخاب ویژگی

با توجه به اینکه بسته‌ها در پایگاه دادهٔ NSL-KDD به ۵ کلاس مختلف تقسیم شده‌اند، میزان دقت درون دسته‌ای به‌دست‌آمده با به‌کارگیری الگوریتم‌های انتخاب ویژگی متفاوت و با اندازهٔ جمعیت‌های مختلف در شکل (۳) نشان داده شده است. دسته نفوذهایی که در پایگاه دادهٔ آموزشی، تعداد کمی نمونه برای یادگیری دارند، با دقت به مراتب میزان کمتری تشخیص داده می‌شوند و همین موضوع به کاهش میزان دقت تشخیص کل منجر می‌شود.

شکل (۳): مقایسهٔ دقت درون دسته‌ای حاصل از اجرای الگوریتم‌های انتخاب ویژگی

۱-۳-٤- مقایسه با سایر روش‌ها

نتایج به‌دست‌آمده از روش پیشنهادی برای انتخاب ویژگی‌های مؤثر در تشخیص نفوذ در جدول (۲) با کارهای مشابه دیگر روی پایگاه دادهٔ NSL-KDD، با استفاده از دسته‌بند ماشین بردار پشتیبان مقایسه شده‌اند.

همان‌طور که مشاهده می‌شود، میانگین دقت به‌دست‌آمده با روش پیشنهادی مقایسه‌شده با روش‌های قبلی و در برخی موارد بهتر بوده است؛ برای مثال، در مقایسهٔ روش پیشنهادی با الگوریتم بهینه‌سازی ازدحام ذرات [۲۱] برای انتخاب ویژگی نتایج دقت به دست‌آمده بهتر بوده است؛ درحالی‌که در روش‌های استفاده‌شده در مراجع [۱۹] و [۲۱] علاوه بر انتخاب ویژگی، مقادیر مناسب پارامترهای دسته‌بندی ماشین بردار پشتیبان نیز با استفاده از بهینه‌سازی یا به‌صورت دستی به دست آمده اند. در پژوهش حاضر فقط روی انتخاب ویژگی تمرکز شده است و پارامترهای متداول برای دسته‌بندی در نظر گرفته شده‌اند.

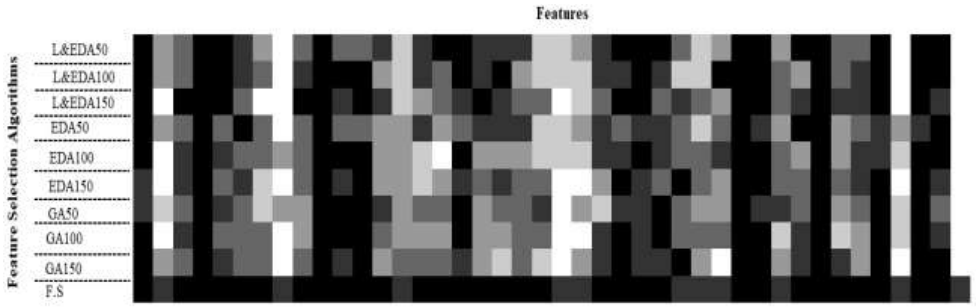
نکتهٔ درخور توجه، گزارش میانگین عملکرد دربارهٔ الگوریتم پیشنهادی در مقایسه با سایر روش‌ها است؛ درحالی‌که دربارهٔ سایر روش ها یا بهترین عملکرد گزارش شده یا در این زمینه اظهار نظری نشده است.

جدول (۲): مقایسهٔ روش پیشنهادی با کارهای مرتبط

Algorithm	Accuracy
PSO-SVM (parameter tuning and feature selection) [21]	81.8
PSO-SVM (parameter tuning) [21]	47.99
Filter (35 features)+SVM [20]	82.34
RFE-SVM [19]	81.27
Local&EDA50-SVM	84.93
GA100-SVM	84
FS-SVM	85.62

۲-۳-٤- ویژگی‌های مؤثر در تشخیص نفوذ

در شکل (٤)، نتایج الگوریتم های متفاوت بررسی شده با هم ترکیب شده‌اند و فرکانس ویژگی‌های انتخاب‌شدهٔ حاصل از ۵ مرتبه اجرای مختلف آنها نشان داده شده است. هر ستون نشان‌دهندهٔ یک ویژگی است و شدت روشنایی هر خانه از تصویر، تعداد دفعات انتخاب‌شدن آن ویژگی در ۵ اجرای مختلف از هر الگوریتم (سطرها) را نشان می‌دهد. مربع های سفید رنگ نشان‌دهندهٔ انتخاب ویژگی مربوطه در هر ۵ مرتبه اجرای الگوریتم‌اند.



شکل (٤): ویژگی‌های انتخاب‌شدهٔ حاصل از ۵ مرتبه اجرای هر الگوریتم

براساس نتایج ترکیبی، ویژگی های زیر در تشخیص نفوذ مؤثرند و با بیشتر الگوریتم های انتخاب ویژگی انتخاب شده‌اند:

- Protocol_type (نوع پروتکل استفاده شده برای اتصال)
- Wrong_fragment (مجموع بسته‌های با کد Checksum اشتباه در یک اتصال)
- Count (تعداد اتصالاتی که آدرس IP مقصد یکسانی دارند)
- Is_urgent_login(اگر کاربر به‌عنوان کاربر مهمان یا ناظر به سیستم دست یابد)
- dst_host_srv_serroe_rate (درصدی از اتصالاتی که شماره پورت مقصد یکسانی دارند و مقدار ویژگی flag آنها S0، S1، S2 یا S3 هستند)
- سختی تشخیص نفوذ

همچنین، برخی ویژگی ها کمک چندانی به تشخیص نفوذ نکرده و با بیشتر الگوریتم‌های انتخاب ویژگی، ویژگی‌های نامربوط یا زائد تلقی شده‌اند:

- Duration (مدت زمان اتصال)
- Flag (وضعیت اتصال)
- Hot (مجموع عملیات انجام‌شده در یک اتصال)
- num_faild_login (تعداد loginهای ناموفق در یک اتصال)
- Logged_in (اگر login صحیح باشد، مقدار یک می‌گیرد)
- srv_diff_host_rate (درصدی از اتصالات ویژگی ٢٤ که ماشین مقصد متفاوتی دارند)
- dst_host_count (مجموع اتصالاتی که آدرس IP مقصد یکسان دارند)
- dst_host_diff_srv_count (درصدی از اتصالات ویژگی ۳۲ که سرویس متفاوت دارند)
- dst_host_serroe_rate (درصدی از اتصالات ویژگی ۳۲ که مقدار ویژگی flag آنها S0، S1، S2 یا S3 هستند)
- dst_host_rerroe_rate (درصدی از اتصالات ویژگی ۳۲ که مقدار ویژگی flag آنها REJ هستند)

۵ – نتیجه‌گیری و کارهای آینده

تشخیص نفوذ در اصل یک مسئلهٔ دسته‌بندی است و انتخاب ویژگی ازجمله موضوعاتی است که در دسته بندی به آن توجه می‌شود. برای داده‌های با ابعاد زیاد، انتخاب ویژگی، زمان تشخیص و هزینه را کاهش می دهد و کارآیی دسته‌بند را بهبود می‌بخشد. در این مقاله، عملکرد الگوریتم‌های انتخاب ویژگی ژنتیک، تخمین توزیع، تخمین توزیع ترکیبی با جستجوی محلی، انتخاب پیشرو و انتخاب پسرو مقایسه شده‌اند و دسته‌بند ماشین بردار پشتیبان به‌عنوان تابع برازندگی این الگوریتم‌ها استفاده شده است. مطابق نتایج به‌دست‌آمده، الگوریتم ژنتیک با اندازهٔ جمعیت ۱۰۰، دقت تشخیص بسته های نرمال را به حداکثر رسانده است. همچنین، الگوریتم تخمین توزیع با جستجوی محلی با اندازهٔ جمعیت ۵۰، منجر به بیشترین دقت در تشخیص حملات نوع DOS شده است و برای تشخیص حملات U2R که کمترین تعداد نمونه در پایگاه دادهٔ آموزشی را دارند، الگوریتم انتخاب پیشرو بهتر از بقیهٔ الگوریتم ها به

تشخیص نفوذ منجر شده است. حملات نوع R2L از الگوریتم انتخاب پسرو و حملات نوع Prob از الگوریتم تخمین توزیع با جستجوی محلی که از جمعیتی با اندازهٔ ۱۵۰ استفاده می کند، با دقت بیشتری به تشخیص نفوذ منجر شده‌اند.

به‌کارگیری سایر الگوریتم‌های تخمین توزیع، ازجمله الگوریتم بهینه‌سازی بیزی و مقایسهٔ آن با نتایج به‌دست‌آمده در این مقاله و ترسیم نقشهٔ پارامتری الگوریتم‌های استفاده‌شده را برای توسعهٔ پژوهش فعلی در آینده می‌توان انجام داد.

مراجع

[1] S. M. Lee, D.S. Kim, and J.S. Park, "A survey and taxonomy of lightweight intrusion detection systems", Journal of Internet Services and Information Security, pp.119–131, 2012.

[2] S.Mukkamala and A.H.Sung, "Identifying Significant Features for Network Forensic Analysis Using Artificial Intelligent Techniques", International Journal of Digital Evidence, Vol.1, pp. 1-17, 2003.

[3] S. M. Tidke, and S.Vishnu, "Intrusion Detection System using Genetic Algorithm and Data Mining":An Overview, International Journal of Computer Science and Informatics, Vol.1, pp. 91-95, 2012.

[4] H.A.Sonawane and T.M Pattewar, "A Comparative Performance Evaluation of Intrusion Detection based on Neural Network and PCA" , presented at the IEEE ICCSP conference ,pp.841-845, 2015

[5] S.Oh, J.S.Lee and B. R Moon, "Hybrid Genetic Algorithms for Feature Selection", IEEE transactions on pattern analysis and machine intelligence, Vol. 26, NO. 11, pp. 1424-1437, 2004

[6] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review, "Data Classification: Algorithms and Applications". Editor: Charu Aggarwal, CRC Press In Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2014.

[7] G.Chandrashekar and F.Sahin, "A survey on feature selection methods," Computers & Electrical Engineering, vol. 40, no. 1, pp. 16-28, 2014.

[8] R.Ravinder, R.B.Kavya and Y.Ramadevi, "A Survey on SVM Classifiers for Intrusion Detection", International Journal of Computer Applications, Vol. 98– No.19, pp.38-44, 2014.

[9] Y.B.Bhavsar and K.C.Waghmare, "Intrusion Detection System Using Data Mining Technique: Support Vector Machine" International Journal of Emerging Technology and Advanced Engineering, Vol. 3, pp.581-586, 2013.

[10] P.Larranaga and J.A.Lozano, "Estimation of Distribution Algorithms": A New Tool for Evolutionary Computation, Kluwer Academic Publishers, 2001.

[11] M. Hauschild and M. Pelikan "An Introduction and Survey of Estimation of Distribution Algorithms", Missouri Estimation of Distribution Algorithms Laboratory Report No. 2011004, Department of mathematics and Computer Science University of Missouri–St. Louis, 2011.

[12] M. Pelikan, "Probabilistic model building Genetic Algorithms" University of Missouri at St. Louis. July 2008.

[13] G.Wang, J.Hao, J.Ma and J.Huang, "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering", Expert Systems with Applications, vol. 37, pp. 6225–6232, 2010.

[14] Y.Chen and A.Abraham, "Estimation of Distribution Algorithm for Optimization of Neural networks for Intrusion Detection System", International Conference on Artificial Intelligence and Soft Computing ICAISC, pp. 9-18, 2006.

[15] M.Sheikhan, Z. Jadidi and A.Farrokhi, "Intrusion detection using reduced-size RNN based on feature grouping", Neural Comput & Applic, No.25, pp. 1185–1190. 2010.

[16] P.NSKH, N.M.Varma and N.R.Ramakrishna, "Principle Component Analysis based Intrusion Detection System Using Support Vector Machine", IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May 20-21, pp.1344-1350, 2016.

[17] G.Kumar, K.Kumar and M.Sachdeva, "The use of artificial intelligence based techniques for intrusion detection: a review", An International Science and Engineering Journal of Artificial Intelligence Review, Vol. 34, pp. 369–387, 2010.

[18] P.Tao, Z.Sun and Z.Sun, "An improved intrusion detection algorithm based on GA and SVM", Human-Centered Smart Systems and Technologies IEEE Access, Vol. 6, March 2018.

[19] A.Jonathan and S.Mandala, "Increasing Feature Selection Accuracy through Recursive Method in Intrusion Detection System", Vol. 4, Issue. 2, pp. 43-50, December 2018.

[20] K.Taher, B.Jisan and M.Rahman, "Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection", International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 10-12 Jan, 2019.

[21] V.Manekar and K.Waghmare, "Intrusion Detection System using Support Vector Machine (SVM) and Particle Swarm Optimization (PSO)", International Journal of Advanced Computer Research, Vol.4 Number-3, Issue-16, September 2014.

[22] N.Krasnogor and J.Smith, "A Tutorial for Competent Memetic Algorithms": Model, Taxonomy, and Design Issues, IEEE Transactions on Evolutionary Computation, Vol. 9, No. 5, October 2005.

[23] M.Tavallaee, N.Stakhanova and A.A Ghorbani, "Towards credible evaluation of anomaly based intrusion detection methods", IEEE Transaction on

System, Man and Cybernetics, Part-c, Applications and Reviews; 40(5):516-524, 2010.
[24] M.Pelikan "Genetic Algorithms", Missouri Estimation of Distribution Algorithms Laboratory Report No. 2010007, Department of mathematics and Computer Science University of Missouri–St. Louis , 2010.

[1] Unsupervised
[2] Supervised
[3] K-Nearest Neighbor
[4] Decision Tree
[5] Support Vector Machine (SVM)
[6] Estimation of Distribution Algorithm (EDA)
[7] Dependency Tree
[8] Feature Selection
[9] Evaluation Function
[10] Natural Selection
[11] Genetic Recombination
[12] Promising Solution
[13] Building Blocks (BBs)
[14] Selection
[15] Crossover
[16] Mutation
[17] New Candidate Solution
[18] Univariate Models
[19] Bivariate Models
[20] Multivariate Models
[21] Univariate Marginal Distribution Algorithm
[22] Population Increamental Learning (PBIL)
[23] Compact Genetic Algorithm (CGA)
[24] Mutual Information Maximizing Input Clustering
[25] Combining Optimizers with Mutual Inf.Trees
[26] Extended Compact Genetic Algorithm (ECGA)
[27] Mutual Information
[28] False Positive Rate
[29] Fuzzy Aggregation
[30] Feed Forward
[31] Particle Swarm Optimization
[32] Principle Component Analysis