Downloaded from adst.ir at 14:58 +0430 on Monday July 24th 2017

مجله علمی پژوهشی «علوم و فناوری‌های پدافند نوین»
سال هشتم، شماره ۳، پاییز ۱۳۹۶؛ ص ۲۶۳-۲۵۱

سامانه تشخیص نفوذ بلادرنگ با استفاده از ترکیب گسسته‌سازی و انتخاب ویژگی‌های مهم

رحیم طاهری[۱]، محمدرضا پارسائی[۲*]، رضا جاویدان[۳]
۱ و ۲- دانشجوی دکتری، ۳- دانشیار، دانشگاه صنعتی شیراز
(دریافت: ۹۵/۰۸/۱۷، پذیرش: ۹۵/۱۱/۲۶)

چکیده

سامانه‌های تشخیص نفوذ در یک شبکه سایبری، یکی از خطوط دفاعی مهم در مقابل تهدیدات است. دو چالش اصلی در حوزه سامانه‌های تشخیص نفوذ، بلادرنگ بودن و دقت تشخیص حملات است که حذف ویژگی‌های غیر مهم و گسسته‌سازی، روش‌های اصلی برای کاهش زمان پردازش بلادرنگ و افزایش دقت مدل هستند. نوآوری این مقاله استفاده از دو روش حذف ویژگی‌های غیر مهم و گسسته‌سازی به صورت هم‌زمان است. در روش پیشنهادی از الگوریتم درخت تصمیم هرس شده C4.5 به عنوان الگوریتم انتخاب ویژگی و گسسته‌سازی در فاز پیش‌پردازش استفاده شده است. نتایج آزمایش‌های انجام شده بر روی مجموعه داده KDD cup 99 نشان می‌دهد که دقت پیش‌بینی مدل در الگوریتم‌های SVM، CART و Naïve Bayes پس از به‌کارگیری روش پیشنهادی در فاز پیش‌پردازش، به ترتیب به ٪۹۹/۳، ٪۹۷/۷ و ٪۹۹/۵ افزایش پیدا می‌کند. همچنین زمان ساخت مدل به ترتیب از ۳۵/۹، ۰/۱ و ۶/۶ ثانیه به ۲/۱، ۰/۰۱ و ۶/۳ ثانیه کاهش می‌یابد. به طور مشابه بر روی مجموعه داده NSL-KDD دقت پیش‌بینی با الگوریتم‌های فوق به ترتیب به ٪۹۹/۳ و ٪۹۹/۵و ۹۶/۶ افزایش پیدا می‌کند و زمان ساخت مدل به ترتیب از ۳۵/۹، ۰/۱ و ۶/۷ ثانیه به ۲/۱، ۰/۰۱ و ۶/۲ ثانیه کاهش می‌یابد. این نتایج نشان می‌دهد که سامانه پیشنهادی می‌تواند به عنوان یک ابزار پدافندی مناسب جهت تشخیص نفوذ در برابر حملات سایبری مورد استفاده مؤثر قرار گیرد.

کلیدواژه‌ها: سامانه تشخیص نفوذ بلادرنگ، گسسته‌سازی، انتخاب ویژگی، درخت تصمیم، داده‌کاوی

Real-Time Intrusion Detection System Using a Combination of Discretization and Feature Selection

R. Taheri, M. R. Parsaei*, R. Javidan
Shiraz University of Technology
(Received: 07/11/2016; Accepted: 14/02/2017)

Abstract

An intrusion detection system in the cyber-networks is one of the most important lines of defense against the threats. Two main challenges in the field of intrusion detection systems are their ability to work in real-time domain and their attack detection accuracy. Elimination of non-critical features and discretization are two systematic ways to reduce the period of real-time processing and to increase the accuracy of the model. The main innovation of this paper is that eliminating of non-critical features and discretization are used simultaneously. In the proposed method, the pruned C4.5 algorithm is used as feature selection together with discretization algorithm in pre-processing phase. Experimental results on KDD cup 99 and NSL-KDD data sets, repectively showed that prediction accuracy of model in SVM, CART and Naïve Bayes algorithms after using the proposed method in the pre-processing phase, increases as 99.25% and 99.26%, 97.66% and 99.52%, 99.46% and 96.62% in that order. Also model construction time are reduced from 35.88, 0.08 and 6.64 seconds to 2.13 and 2.09, 0.01 and 0.01, 6.29 and 6.20 seconds, respectively. The results showed that the proposed system can effectively be used as a modern defense intrusion detection tool against cyber-attacks.

Keywords: Real-Time Intrusion Detection, Discretization, Feature Selection, Decision Tree, Data Mining, SVM

*Corresponding Author E-mail: mr.parsaei@sutech.ac.ir

Downloaded from adst.ir at 14:58 +0430 on Monday July 24th 2017

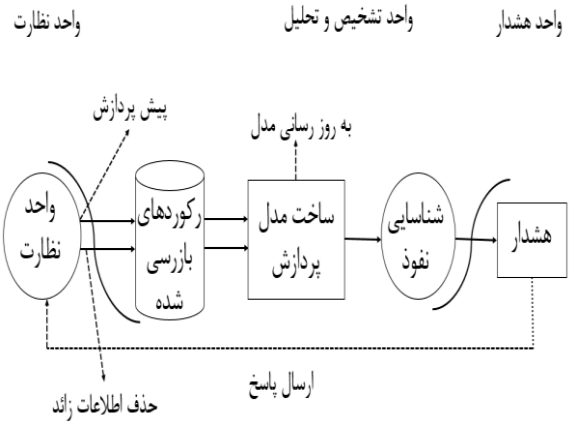
۱. مقدمه

نفوذ فعالیتی است که توسط آن محرمانگی، صحت یا دسترسی‌پذیری به منابع دچار اختلال می‌شود. تشخیص نفوذ در واقع شناسایی دستیابی‌های غیر مجاز به حملات انجام شده به شبکه است. سامانه تشخیص نفوذ وظیفه نظارت بر فعالیت سامانه، تجزیه و تحلیل بسته‌های شبکه، تعیین الگوی حملات و ارزیابی صحت و یکپارچگی فایل‌ها را بر عهده دارد. در تشخیص نفوذ رویدادهای یک سامانه پایش شده و بر اساس این پایش‌ها وقوع نفوذ در آن سامانه مشخص می‌شود [۱]. سامانه‌های تشخیص نفوذ را می‌توان از نظر تحلیلی که روی داده‌های ورودی انجام می‌دهند، در دو دسته مبتنی بر ناهنجاری و مبتنی بر امضاء قرار داد [۲].

در روش مبتنی بر ناهنجاری، مسئله اصلی تعریف پروفایل نرمال برای کاربران، برنامه‌های کاربردی یا ترافیک شبکه است. چالش اصلی در این روش مشخص کردن یک حد آستانه برای تمایز دادن رفتار نرمال از رفتار غیر نرمال است. برای این منظور از مفاهیم آماری و خوشه‌بندی استفاده می‌شود. روش‌های مبتنی بر ناهنجاری دقت کمتری نسبت به روش‌های مبتنی بر امضاء دارند. اما توانایی آن‌ها در شناسایی حملات جدید بیشتر است.

روش مبتنی بر امضاء، معمولاً هدف تعیین دنباله‌ای از اعمال است که در هنگام وقوع یک حمله رخ می‌دهد. این روش‌ها الگوهای حملات مختلف را در اختیار دارند و عمل تطبیق الگو را برای پیدا کردن حملات انجام می‌دهند و در مقایسه با روش‌های مبتنی بر ناهنجاری دقت بیشتری دارند. محدودیت روش‌های مبتنی بر امضاء آن است که توانایی تشخیص حملات جدید و حملاتی که الگوی آن‌ها در دسترس نیست را ندارند [۳].

معمولاً سامانه‌های تشخیص نفوذ با یک ساختار سه سطحی توصیف می‌شوند [۴] (شکل (۱)).



شکل ۱. ساختار سه بخشی و عمومی سامانه‌های تشخیص نفوذ

واحد نظارت، وظیفه ثبت رخدادها را بر عهده دارد. رخدادهای سامانه توسط این واحد جهت تشخیص و تحلیل ثبت می‌شوند. همچنین نمایش هشدار به مدیر شبکه یا مسئول نظارت و پیگیری رویدادهای شبکه بر عهده این واحد است.

واحد تشخیص و تحلیل، اطلاعات ثبت شده توسط واحد نظارت را به عنوان ورودی دریافت می‌کند. سپس بر اساس این اطلاعات به یکی از روش‌های مبتنی بر امضاء یا مبتنی بر ناهنجاری، مدل ساخته می‌شود. حملات و تهدیدهای امنیتی به کمک مدل ساخته شده تمایز داده می‌شوند. درنهایت واحد هشدار وظیفه دارد که متناسب با نوع حمله رفتار مناسب را انجام دهد. این واحد در صورت نیاز اخطارهای مناسب را برای ثبت یا نمایش به واحد نظارت ارسال می‌کند [۴ و ۵].

داده کاوی، به عنوان یکی از مهم‌ترین روش‌های تشخصی نفوذ، به فرآیند استخراج خودکار مدل‌ها از میان انبوه داده‌ها اطلاق می‌شود. از انواع الگوریتم‌های داده کاوی می‌توان به منظور تشخیص حملات و نفوذها در سامانه‌های رایانه‌ای استفاده نمود. یکی از فازهایی که در الگوریتم‌های داده کاوی به‌کار می‌رود، پیش‌پردازش است. روش‌هایی برای پیش‌پردازش داده‌ها وجود دارد که همگی قابل به‌کارگیری در سامانه‌های تشخیص نفوذ هستند. این روش‌ها شامل تمیز کردن داده‌ها، نمونه‌گیری، تغییر شکل دادن داده‌ها، انتخاب ویژگی و گسسته‌سازی هستند که قبل از تحلیل و کاوش در داده‌ها مورد استفاده قرار می‌گیرند.

گسسته‌سازی یک روش پیش‌پردازش داده‌ها است که در آن عملیات تبدیل داده‌های پیوسته به بازه‌هایی جدا از هم صورت می‌گیرد و در افزایش دقت مدل ساخته شده و مسئله بلادرنگ بودن سامانه‌های تشخیص نفوذ مؤثر است [۶]. بنابراین با توجه به اهمیت بلادرنگ بودن در سامانه‌های تشخیص نفوذ رکوردهای ورودی گسسته‌سازی می‌شوند که علاوه بر بهبود دقت، در کاهش زمان مورد نیاز برای ساخت مدل و بازرسی رکوردهای ورودی نیز مؤثر باشد [۷ و ۸].

انتخاب ویژگی یکی از روش‌های پیش‌پردازش داده‌ها است که زیرمجموعه‌ای از ویژگی‌های مجموعه داده را انتخاب می‌کند؛ به گونه‌ای که ویژگی‌های انتخاب شده فاقد ویژگی‌های نامربوط و افزونه باشند. ویژگی‌های نامربوط ویژگی‌هایی هستند که در ساخت مدل تأثیری ندارند. ویژگی‌های افزونه ویژگی‌هایی هستند که اطلاعات جدیدی برای ساخت مدل ندارند و این اطلاعات را می‌توان از دیگر ویژگی‌ها به‌دست آورد [۹]. معمولاً فرآیند انتخاب ویژگی شامل چهار قدم اصلی است. تولید زیرمجموعه‌ای از ویژگی‌ها، ارزیابی زیرمجموعه تولید شده، معیار توقف و معتبرسازی نتایج. هدف اصلی الگوریتم‌های انتخاب ویژگی،

Downloaded from adst.ir at 14:58 +0430 on Monday July 24th 2017

انتخاب ویژگی‌های مربوط است. روش‌های انتخاب ویژگی مزیت‌های دیگری هم دارند. این مزیت‌ها عبارتند از: بهبود کارایی و دقت مدل، ساخت سریع‌تر و کاهش هزینه زمانی ساخت مدل، درک بهتر از عملیاتی که داده را تولید می‌کند، نمایش و بصری کردن ساده‌تر داده‌ها و کاهش فضای مورد نیاز برای ذخیره کردن داده‌ها [۱۰].

همان‌طور که در بخش تحقیقات پیشین اشاره شده است هیچ کدام از روش‌های گسسته‌سازی و انتخاب ویژگی به تنهایی سرعت مناسب را برای یک سامانه بلادرنگ فراهم نمی‌کنند و بنابراین در این مقاله روشی ارائه شده است که با ترکیب انتخاب ویژگی و گسسته‌سازی داده‌ها در الگوریتم C4.5، برای سامانه‌های تشخیص نفوذ بلادرنگ مناسب است.

روش پیشنهادی در این پژوهش بر مبنای دو محور است. ابتدا استفاده از درخت تصمیم هرس شده C4.5 به عنوان الگوریتم انتخاب ویژگی است که با حذف ویژگی‌های افزونه و نامربوط می‌تواند باعث کاهش زمان ساخت مدل می‌شود. این مسئله در سامانه‌های تشخیص نفوذ که بلادرنگ بودن در آن‌ها یک پارامتر اساسی است، بسیار مفید است. علاوه بر این انتخاب ویژگی‌ها می‌تواند دقت را نیز افزایش دهد. پس از انتخاب ویژگی از الگوریتم C4.5، برای گسسته‌سازی متغیرهای پیوسته استفاده شده است. الگوریتم C4.5، در زمان ساخت مدل با پیدا کردن یک نقطه برش، متغیرهای پیوسته را به گسسته تبدیل می‌کند. از این نقطه برش برای گسسته‌سازی متغیرهای پیوسته در فاز پیش‌پردازش استفاده شده است. گسسته‌سازی نیز مانند انتخاب ویژگی می‌تواند باعث افزایش دقت و کاهش زمان مورد نیاز برای ساخت مدل شود. تجمیع گسسته‌سازی و انتخاب ویژگی می‌تواند تأثیر بیشتری در پارامترهای دقت و زمان داشته باشد که تاکنون در پژوهش‌ها انجام نشده است و نوآوری اصلی این مقاله است. نتایج نیز بیانگر مؤثر بودن روش پیشنهادی است.

۲. پیشینه تحقیق

روش پیشنهادی در این پژوهش استفاده از گسسته‌سازی و انتخاب ویژگی‌ها به صورت هم‌زمان است. به همین دلیل در بخش تحقیقات پیشین ابتدا تحقیقات در زمینه روش‌های انتخاب ویژگی‌ها و سپس روش‌های گسسته‌سازی مورد استفاده در حوزه سامانه‌های تشخیص نفوذ بررسی می‌شوند.

هر رخداد یا ارتباط در شبکه می‌تواند به یک رکورد نگاشت شود. مجموعه ویژگی‌های یک رکورد مشخصات رخداد یا ارتباط را توصیف می‌کنند. با توجه به تعداد زیاد ویژگی‌ها برخی از ویژگی‌ها شامل اطلاعات مفید هستند که به صورت مستقیم در شناسایی نفوذ مؤثر هستند. برخی دیگر شامل اطلاعاتی هستند که تأثیر کمتری در فرآیند تحلیل و تشخیص نفوذ دارند و برخی دیگر از ویژگی‌ها نیز در فرآیند تشخیص نفوذ، بی‌تأثیر می‌باشند [۱۱]. در نظریه با افزایش تعداد ویژگی‌ها، بهتر می‌توان رکوردهای نرمال و غیر نرمال را از هم جدا کرد ولی گاهی اوقات این عمل با توجه به وجود ویژگی‌های نامربوط و افزونه درست نیست [۱۲].

ماندیانی و همکاران [۱۳] با استفاده از الگوریتم خوشه‌بندی k-means و طبقه‌بندی کننده C4.5 روشی برای تشخیص نفوذ در سامانه‌های رایانه‌ای ارائه کرده‌اند. در مقاله آن‌ها برای طبقه‌بندی از روش فازبندی سه مرحله‌ای استفاده شده است که باعث افزایش زمان می‌شود و برای سامانه‌های بلادرنگ مناسب نیست.

هدایتی و همکاران [۱۴] با استفاده از الگوریتم درخت تصمیم و الگوریتم خوشه‌بندی k-means تشخیص نفوذ در شبکه را انجام داده‌اند. این آزمایش بر روی دو مجموعه داده KDD CUP99 و NSL-KDD آموزش و آزمایش شده است و به دلیل اینکه از روش‌های ترکیبی استفاده شده است، زمان تشخیص افزایش یافته است.

روش دیگر، حذف ویژگی‌ها به صورت یکی یکی است. بدین صورت که یک ویژگی از مجموعه داده حذف می‌شود و پس از آن مجموعه داده جدید به الگوریتم القاء می‌شود. اگر پس از حذف ویژگی متریک‌های کارایی مدل بهبود یافت، می‌توان آن ویژگی را غیر مهم در نظر گرفت [۱۵ و ۱۶]. این روش نیز به دلیل لزوم کنترل کردن کارایی پس از حذف تک تک ویژگی‌ها زمان زیادی لازم دارد و مناسب سامانه‌های بلادرنگ نیست.

الگوریتم درخت تصمیم CART، اولین بار توسط بریمن و همکارانش مطرح شد [۱۷]. جداسازی در درخت CART به صورت باینری است. در این الگوریتم بهترین متغیر که کمترین ناخالصی را دارد، به عنوان متغیر جدا کننده اصلی در نظر گرفته می‌شود. این الگوریتم برای هر ویژگی جدا کننده یک ویژگی جانشین نیز پیدا می‌کند. ویژگی جانشین، ویژگی است که رفتاری شبیه به رفتار ویژگی جدا کننده اصلی دارد. به بیان دیگر تا حد ممکن مانند ویژگی اصلی رکوردها را جدا کند. الگوریتم CART اهمیت ویژگی‌ها را نیز محاسبه می‌کند. اهمیت یک ویژگی از جمع بهبود مربوط به همان ویژگی در تمامی گره‌ها محاسبه می‌شود. پیچیدگی فرآیند به‌کار رفته در این الگوریتم باعث شده است که زمان زیادی از زمان پردازنده صرف محاسبه اهمیت ویژگی‌ها شود و به همین دلیل کارایی چندانی ندارد.

یک روش دیگر برای انتخاب ویژگی‌ها، حذف ویژگی‌هایی است که با هم همبستگی دارند [۱۸]. این کار باعث حذف

Downloaded from adst.ir at 14:58 +0430 on Monday July 24th 2017

ویژگی‌هایی می‌شود که دارای اطلاعات افزونه هستند. مشکل این الگوریتم آن است که به طور کلی ویژگی‌های حاوی اطلاعات افزونه را حذف می‌کند، در حالی که گاهی این افزونگی به افزایش دقت روش کمک می‌کند.

بیدگلی و همکاران [۱۹] فاز پیش‌پردازش داده‌ها را به اجرای روش‌های انتخاب ویژگی شامل الگوریتم CART، انتخاب ویژگی‌ها توسط Markov Blanket، الگوریتم CLIQUE (که یک الگوریتم خوشه‌بندی بر اساس چگالی است) و الگوریتم شبکه عصبی ارتجاعی، اختصاص داده‌اند. پس از آن الگوریتم درخت تصمیم C4.5 بر روی مجموعه داده کاهش یافته اجرا شده است. نتایج بر اساس پارامترهای زمان آموزش، زمان آزمون و دقت با هم مقایسه شده‌اند. نتیجه نهایی نشان دهنده برتری هر مجموعه ویژگی انتخاب شده در برخی از حملات است. به عنوان مثال اگر چه ویژگی‌های انتخاب شده توسط الگوریتم CART بهترین عملکرد را در تشخیص داده‌های نرمال دارد ولی هیچ کدام از الگوریتم‌های معرفی شده در حملات مختلف کارایی بهتری نسبت به بقیه نداشته است.

هانچوان و همکاران [۲۰] یک معیار حداقل افزونگی و حداکثر ارتباط (mRMR) برای انتخاب ویژگی‌ها به صورت مرحله‌ای ارائه کرده‌اند. این معیار امکان انتخاب ویژگی‌هایی با هزینه بسیار پایین را می‌دهد. این روش در مقایسه با معیار حداکثر ارتباط نشان می‌دهد که انتخاب ویژگی mRMR می‌تواند نرخ دسته‌بندی را بهبود بخشد. اما این تحقیق بر روی داده‌های پیوسته انجام شده است.

آگروال و آمریتا [۲۱] شش روش مختلف انتخاب ویژگی که در نرم‌افزار WEKA موجود هستند را با یکدیگر ترکیب کرده‌اند. پس از انتخاب ویژگی‌ها مدل‌ها بر اساس الگوریتم‌های Naïve Bayes و C4.5 ساخته شده‌اند. نتایج از نظر پارامترهای زمان آموزش، زمان آزمون و نرخ تشخیص با هم مقایسه شده‌اند که نتایج حاصل مقایسه‌های ذکر شده در فوق را تأیید می‌کند.

فیاد و ایرانی [۲۲] یک روش گسسته‌سازی با چند بازه، که از کمینه کردن اطلاعات آنتروپی استفاده می‌کند، پیشنهاد کردند. همچنین در نظریه نشان دادند، گسسته‌سازی متغیرهای پیوسته توسط این روش عملکرد بهتری نسبت به روش گسسته‌سازی پیش‌فرض مورد استفاده در الگوریتم‌هایCART، ID3 و C4.5 دارد. این الگوریتم‌ها به صورت پیش‌فرض گسسته‌سازی یک متغیر پیوسته را به صورت باینری انجام می‌دهند (هر گره فقط دو فرزند دارد). استفاده از این روش گسسته‌سازی در سامانه‌های تشخیص نفوذ علاوه بر کاهش زمان، دقت را نیز با توجه به الگوریتم خوشه‌بندی مورد استفاده افزایش می‌دهد. ترکیب این روش گسسته‌سازی و انتخاب ویژگی با الگوریتم ژنتیک، دقت را در بعضی از الگوریتم‌های خوشه‌بندی افزایش می‌دهد [۲۳] اما به دلیل زمان زیاد مورد نیاز برای اجرای الگوریتم ژنتیک مناسب کاربردهای بلادرنگ نیست.

الگوریتم Equal-width Binning، ساده‌ترین روش گسسته‌سازی است [۲۴]. این روش نیاز به مشخص کردن مقدار k دارد که عبارت است از تعداد بازه‌هایی که توسط کاربر مشخص می‌شود. هر ویژگی پیوسته با توجه به مقدار مشخص شده به k بازه مساوی تقسیم می‌شود. نتایج نشان می‌دهد که استفاده از این الگوریتم ساده و الگوریتم ژنتیک برای استخراج قوانین، خیلی مؤثر نیست [۲۵].

۳. روش تحقیق

روش پیشنهادی در این مقاله عملیات انتخاب ویژگی جدا کننده و گسسته‌سازی ویژگی‌های پیوسته را بر اساس الگوریتم C4.5 انجام می‌دهد. اما چون در روش گسسته‌سازی پیشنهادی بعد از اجرای الگوریتم C4.5، از نقاط برش انتخاب شده توسط این الگوریتم به عنوان نقاط برش برای گسسته‌سازی استفاده می‌شود، گسسته‌سازی به صورت با ناظر خواهد بود.

۳-۱. نحوه انتخاب ویژگی جدا کننده در الگوریتم C4.5

C4.5 یک الگوریتم درخت تصمیم است. این الگوریتم بر اساس روش تقسیم و غلبه کار می‌کند. وظیفه الگوریتم‌های درخت تصمیم، پیش‌بینی کردن متغیر کلاس برای رکوردهایی است که قبلاً دیده نشده‌اند. در ساخت درخت به یک معیار اندازه‌گیری مناسب نیاز است که هر بار مشخص کند کدام جداسازی مناسب است. به بیان دیگر، جداسازی مناسب است که اطلاعات بیشتری را به ما بدهد. با در نظر داشتن رابطه (۱)، اگر $info(T)$ مقدار اطلاعات در گره پدر باشد (قبل از جداسازی) و $info_x(T)$ مقدار اطلاعات به‌دست آمده پس از جداسازی باشد، $Gain(X)$ مقدار اطلاعات به‌دست آمده از جداسازی را مشخص می‌کند. در C4.5 از سنجش آنتروپی برای محاسبه میزان اطلاعات به‌دست آمده استفاده می‌شود.

$$Gain(X) = info(T) - info_x(T) \qquad (۱)$$

۳-۲. روش گسسته‌سازی ویژگی‌های پیوسته در الگوریتم C4.5

با توجه به اینکه برای پیش‌بینی متغیر کلاس در درخت‌های تصمیم، لازم است که مسیر شاخه‌ها طی شود تا درنهایت به برگ رسیده و متغیر کلاس پیش‌بینی شوند، ویژگی‌های پیوسته در هنگام ساخت درخت معمولاً به صورت پویا به گسسته تبدیل می‌شوند. بنابراین اگر A نام ویژگی باشد، باید یک مقدار برای T

Downloaded from adst.ir at 14:58 +0430 on Monday July 24th 2017

پیدا شود. به گونه‌ای که رکوردها به صورت A≤T و A>T جداسازی شوند. T همان نقطه برش است. الگوریتم C4.5 برای پیدا کردن مقدار T، ابتدا تمام مقادیر ویژگی پیوسته A را مرتب می‌کند. اگر مقادیر به صورت $\{v_1, v_2, \dots v_i\}$ باشند. الگوریتم هر بار یک مقدار (که برابر با $\frac{v_i+v_{i+1}}{2}$) بین دو مقدار همسایه v_i و v_{i+1} انتخاب می‌کند و اطلاعات به‌دست آمده از این جداسازی را بر اساس رابطه (۱) محاسبه می‌کند. درنهایت، جداسازی که بیشترین اطلاعات را داشته باشد انتخاب می‌شود. اگر i تعداد مقدارها برای ویژگی پیوسته A باشد، $i-1$ بار (به ازای هر دو مقدار همسایه) باید رابطه (۱) محاسبه شود تا بهترین مقدار برش برای گسسته‌سازی ویژگی A پیدا شود. بنابراین روش گسسته‌سازی مورد استفاده در الگوریتم C4.5 یک روش با ناظر، پویا و محلی است.

۳-۳. انواع هرس و هرس خطا-محور در الگوریتم C4.5

به طور کلی می‌توان روش‌های هرس در درخت تصمیم را به دو دسته پیش-هرس و پسین-هرس تقسیم کرد. روش‌های پیش-هرس در هنگام ساخت درخت بر اساس قوانین توقف از پیش مشخص شده رشد درخت را متوقف می‌کنند. برخی از این قوانین در مرجع [۲۶] ذکر شده‌اند. پسین هرس به معنی ساخت درخت به صورت کامل و انجام عمل هرس پس از ساخت درخت است.

این روش از نظر محاسباتی پیچیده‌تر است. به عبارت دیگر در این روش درختی ساخته می‌شود که در مرحله بعدی ممکن است بخشی از آن هرس شود. اما پرداخت هزینه بیشتر محاسباتی در مقابل دریافت سودی که ممکن است از جداسازی‌های بیشتر به‌دست آورده شود، قابل توجیه است [۲۷]. مرسوم‌ترین روش‌های پسین هرس عبارتند از: هرس کاهش خطا [۲۸ و ۲۹]، هرس بدبینانه [۳۰]، هرس کمینه خطا [۳۱]، هرس مقدار بحرانی [۳۲]، هرس خطا محور [۲۷] و هرس هزینه پیچیدگی [۳۰]. برای مطالعه جزئیات بیشتر، مقایسه، ارزیابی دقت و پیچیدگی روش‌های مختلف پسین هرس بر روی مجموعه داده‌های مختلف، این منابع مناسب هستند [۳۱ و ۳۲].

ایده در روش هرس خطا محور استفاده از مجموعه آموزش برای پیش‌بینی کردن خطا در داده‌های آزمایش (داده‌های دیده نشده) است. بر این اساس فرض شده است که در فاز آموزش درخت ساخته شده با مجموعه داده‌ای آزمون خواهد شد که اندازه آن برابر با داده‌های فاز آموزش است؛ اگر چه این فرض لزوماً همیشه برقرار نیست. در این روش از راهبرد پایین به بالا برای هرس کردن استفاده می‌شود. ریشه در یک زیردرخت هنگامی به برگ تبدیل می‌شود که نرخ خطا برای تمامی گره‌های آن زیردرخت محاسبه شده باشد. نرخ خطا برای هر برگ محاسبه می‌شود. پس از محاسبه نرخ خطا برای هر برگ، نرخ خطا برای گره والد برگ‌ها با ترکیب برگ‌ها در گره والد قبل از اینکه جداسازی شوند نیز محاسبه می‌شود. اگر نرخ خطای گره والد کمتر بود فرزندان گره والد هرس می‌شوند.

در هرس خطا محور فرض شده است که خطای داده‌ها داری توزیع دوجمله‌ای است. همچنین از Confidence Factor (CF) برای تخمین احتمال حد بالای خطا در یک گره استفاده می‌شود. این کار با در نظر گرفتن CF به عنوان سطح اطمینان در رابطه (۲) انجام می‌شود. الگوریتم C4.5 به صورت پیش‌فرض از سطح اطمینان ۲۵٪ استفاده می‌کند. به بیان دیگر اگر CF=٪۲۵ باشد، با ۷۵٪ بازه اطمینان، احتمال نرخ خطا در داده‌های دیده نشده آزمایش برابر با [0,p] است. p مقدار مجهول در رابطه است [۳۳].

این عدد به عنوان یک پارامتر توسط کاربر می‌تواند تغییر کند. هر چه این عدد بزرگ‌تر باشد هرس کمتری انجام می‌شود. در رابطه (۲)، E برابر با تعداد خطای دسته‌بندی در گره است. N تعداد رکوردهای آموزش در گره است. با داشتن این سه مقدار می‌توان احتمال خطا یا همان p را در داده‌های آزمایش پیش‌بینی کرد.

$$if\ \mathrm{E} > 0\ \ then\ \ CF = \sum_{x=0}^{E} \binom{N}{x} p^x\ (1-p)^{N-x}$$

$$Else\ if\ E = 0\ then\ CF = (1-p)^N \qquad (۲)$$

۴-۳. فاز پیش‌پردازش روش پیشنهادی

شکل (۲) ترتیب اجرای مراحل روش پیشنهادی را نشان می‌دهد. روش پیشنهادی یک روش در فاز پیش‌پردازش داده‌ها است. به بیان دیگر قبل از اجرای الگوریتم‌های دسته‌بندی ویژگی‌های مهم انتخاب می‌شوند، پس از آن داده‌ها گسسته‌سازی می‌شوند و درنهایت الگوریتم‌های دسته‌بندی اجرا می‌شوند.

در روش پیشنهادی از الگوریتم C4.5 در فاز پیش‌پردازش داده‌ها به منظور انتخاب ویژگی‌های مهم و گسسته‌سازی استفاده می‌شود، اما از این الگوریتم برای دسته‌بندی استفاده نمی‌شود. با این هدف که بتوان بهترین ویژگی‌ها را انتخاب کرد و بهترین بازه‌ها را برای گسسته‌سازی پیدا نمود، از همه داده‌ها برای آموزش الگوریتم C4.5 استفاده می‌شود.

در درخت ساخته شده نهایی توسط الگوریتم C4.5 به دو دلیل برخی از ویژگی‌ها در درخت استفاده نمی‌شوند.

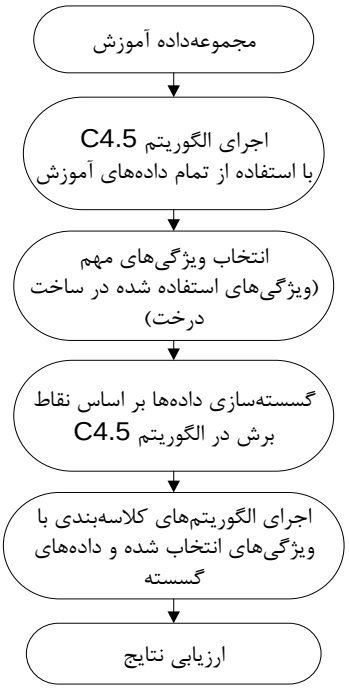
دلیل اول این است که ممکن است در هر بار آزمایش برای پیدا کردن بهترین ویژگی به منظور ساخت درخت (ویژگی با ناخالصی کمتر)، برخی از ویژگی‌ها در مقایسه با بقیه ویژگی‌ها

Downloaded from adst.ir at 14:58 +0430 on Monday July 24th 2017

همیشه حاوی اطلاعات کمتری باشند. به بیان دیگر این ویژگی‌ها در تمام مراحل ساخت درخت اطلاعات کمتری نسبت به بقیه ویژگی‌ها دارند و هرگز در ساخت درخت استفاده نمی‌شوند. علاوه بر این دلیل دومی که برخی از ویژگی‌ها در درخت نهایی وجود ندارند این است که ممکن است برخی از ویژگی‌ها پس از ساخت درخت هرس شوند (روش پسین هرس). روش پیشنهادی ویژگی‌های استفاده شده برای ساخت درخت تصمیم را به عنوان ویژگی‌های مهم انتخاب می‌کند.

الگوریتم C4.5 از روش جاسازی شده برای انتخاب ویژگی‌ها استفاده می‌کند. اما چون از این الگوریتم در فاز پیش‌پردازش استفاده شده است و پس از ساخت درخت ویژگی‌های غیر مهم حذف شده‌اند، روش انتخاب ویژگی پیشنهاد شده یک روش انتخاب ویژگی فیلتر خواهد بود.

در الگوریتم C4.5 برای پیدا کردن ویژگی مناسب در زمان ساخت درخت و محاسبه اطلاعات ویژگی‌ها نیاز است که ویژگی‌های پیوسته گسسته‌سازی شوند. پس از گسسته‌سازی ویژگی‌های پیوسته، ویژگی‌های مناسب برای ساخت درخت انتخاب می‌شوند (بر اساس اندازه‌گیری ناخالصی).



شکل ۲. مراحل روش پیشنهادی

در روش پیشنهادی بر اساس مقادیری که برای گسسته‌سازی ویژگی‌های پیوسته در ساخت درخت استفاده می‌شوند، داده‌ها گسسته‌سازی می‌شوند. به بیان دیگر روش پیشنهادی از مقادیر برش ویژگی‌های پیوسته در درخت نهایی به عنوان بازه‌های گسسته‌سازی ویژگی‌های پیوسته استفاده می‌کند و قبل از اجرای الگوریتم‌های کلاسه‌بندی ویژگی‌های پیوسته بر اساس این بازه‌ها گسسته‌سازی می‌شوند.

روش گسسته‌سازی مورد استفاده در الگوریتم C4.5 یک روش با ناظر، پویا و محلی است. اما به دلیل اینکه در روش گسسته‌سازی پیشنهادی بعد از اجرای الگوریتم C4.5 از نقاط برش انتخاب شده توسط این الگوریتم به عنوان نقاط برش برای گسسته‌سازی استفاده می‌شود، روش گسسته‌سازی استفاده شده در این پژوهش به صورت با ناظر، استاتیک، سراسری و مستقیم است. به این دلیل با ناظر است چون از اطلاعات متغیر کلاس برای به‌دست آوردن نقاط برش گسسته‌سازی استفاده می‌شود. همچنین استاتیک است چون در مرحله پیش‌پردازش گسسته‌سازی بر روی داده‌ها انجام می‌شود و سپس الگوریتم‌های دسته‌بند اعمال می‌شوند. روش پیشنهادی به صورت سراسری است چون بر روی تمام رکوردها اجرا می‌شود و در آخر روش مورد استفاده را می‌توان مستقیم نامید چون تعداد بازه‌ها مشخص است.

۴. نتایج و بحث

جهت بررسی سامانه‌های تشخیص نفوذ معمولاً از مجموعه داده‌های KDD cup99 و NSL-KDD استفاده می‌گردد، بنابراین در این بخش از مقاله ابتدا این مجموعه داده‌ها به طور مختصر معرفی شده است. سپس چگونگی انجام آزمایش‌ها تشریح می‌شود. با توجه به کاربرد گسترده‌تر مجموعه داده KDD cup99 در تحقیقات پیشین سامانه‌های تشخیص نفوذ و امکان مقایسه روش پیشنهادی با طیف گسترده‌ای از پژوهش‌ها، آزمایش اصلی با این مجموعه داده صورت گرفته است. همچنین با توجه به پیچیده و مبهم شدن رفتارهای مخرب از سوی مهاجمین و لزوم استفاده از مجموعه داده جدیدتر، آزمایش‌ها با مجموعه داده NSL-KDD نیز انجام شده است که نتایج حاصل بر روی هر دو مجموعه داده بهبود رضایت بخشی را نشان می‌دهد.

۴-۱. مجموعه داده‌های مورد استفاده

از سال ۱۹۹۹ مجموعه داده KDD cup99، پر کاربردترین مجموعه داده مورد استفاده در زمینه سامانه‌های تشخیص نفوذ مبتنی بر شبکه بوده است. این مجموعه داده شامل حدود پنج میلیون رکورد است. هر رکورد نشان دهنده یک اتصال شبکه است [۳۴]. این رکوردها در طول ۷ هفته نظارت بر ترافیک شبکه ثبت شده‌اند. هر رکورد شامل ۴۱ ویژگی اصلی به علاوه یک ویژگی برای متغیر کلاس است. این ۴۱ ویژگی نشان دهنده مشخصات ارتباط می‌باشند. ویژگی آخر نوع حمله یا نرمال بودن ارتباط را مشخص می‌کند. حملات این مجموعه داده به چهار دسته زیر

Downloaded from adst.ir at 14:58 +0430 on Monday July 24th 2017

گروه‌بندی می‌شوند [۳۵]:

- *حملات DoS (Denial of Service)*: در این نوع حمله، حمله کننده سعی دارد که از منابع قربانی به حدی استفاده کند که قربانی از نظر حافظه و توان محاسباتی، قادر به پاسخگویی به سایر درخواست‌ها نباشد.

- *حملات U2R (User to Root)*: در این نوع حمله، حمله کننده در ابتدا به عنوان یک کاربر نرمال قصد دسترسی به شبکه را دارد. پس از متصل شدن با حساب کاربری نرمال، حمله کننده از نقاط ضعف سامانه برای به‌دست آوردن دسترسی ریشه استفاده می‌کند.

- *حملات R2L (Remote to Local)*: این نوع از حملات زمانی اتفاق می‌افتند که حمله کننده توانایی ارسال بسته به یک ماشین روی شبکه را دارد اما حمله کننده روی ماشین قربانی حساب کاربری ندارد. حمله کننده سعی دارد از نقاط ضعف سامانه برای دسترسی به عنوان کاربر محلی به ماشین قربانی استفاده کند.

- *حملات Probe*: در واقع Probe حمله نیست. بلکه حمله کننده سعی دارد اطلاعات مفید و نقاط ضعف سامانه را برای سامان‌دهی یک حمله به‌دست آورد.

در مجموعه داده KDD cup99 به منظور رعایت محرمانگی در آزمایش‌های گوناگون، داده‌های اصلی با داده‌های حملات مختلف ترکیب شده است. به همین خاطر مجموعه داده NSL-KDD معرفی شد که شامل زیرمجموعه کوچکی از KDD cup99 است. در این مجموعه داده رکوردهای تکراری حذف شده است. بنابراین جواب‌ها به سمت این رکوردهای تکراری متمایل نیست. در جدول (۱) تعداد رکوردهای دو مجموعه داده ذکر شده است.

جدول ۱. تعداد رکوردهای دو مجموعه داده مورد استفاده

تعداد رکورد NSL-KDD	تعداد رکورد KDD cup99
۱۲۵۹۷۳	۴۹۴۰۲۱

۴- ۲. طراحی آزمایش

مشابه همه تحقیقات انجام شده در این زمینه، در این مقاله از مجموعه داده KDD cup 99 به تعداد ۱۳۴۹۹ رکورد و از مجموعه داده NSL-KDD به تعداد ۲۲۵۴۴ رکورد به صورت تصادفی انتخاب و استفاده شده است. متغیر کلاس برای هر رکورد شامل یکی از مقادیر Normal، Probe، R2L، U2R و DoS است. چون هدف استفاده از درخت تصمیم به عنوان الگوریتم انتخاب ویژگی و گسسته‌سازی است، از همه رکوردهای انتخاب شده برای ساخت مدل استفاده شده است. لازم به ذکر است که فقط برای ساخت درخت بر اساس الگوریتم C4.5 از تمام رکوردها استفاده شده است و برای ترکیب روش پیشنهاد شده با دیگر الگوریتم‌های کلاسه‌بندی، روش 10-fold cross validation مورد استفاده قرار گرفته است. درخت بر اساس روش خطا محور به عنوان روش پیش‌فرض در الگوریتم C4.5 هرس شده است و مقدار پیش‌فرض CF=0.25 در نظر گرفته شده است. تعداد ویژگی‌های استفاده شده در ساخت درخت، ۲۱ ویژگی به علاوه متغیر کلاس از ۴۱ ویژگی است. لازم به ذکر است که این تعداد ویژگی توسط الگوریتم پیشنهادی انتخاب ویژگی تعیین شده است. ۴۱ ویژگی مجموعه داده‌های KDD cup99 و NSL-KDD بر اساس جدول (۲) نام‌گذاری شده است.

جدول ۲. نام‌گذاری ویژگی‌های مجموعه داده‌های مورد استفاده

نام ویژگی	نام انتخاب شده	نام ویژگی	نام انتخاب شده
duration	A	is_guest_login	V
protocol_type	B	count	W
service	C	srv_count	X
flag	D	serror_rate	Y
src_bytes	E	srv_serror_rate	Z
dst_bytes	F	rerror_rate	AA
land	G	srv_rerror_rate	AB
wrong_fragment	H	same_srv_rate	AC
urgent	I	diff_srv_rate	AD
hot	J	srv_diff_host_rate	AE
num_failed_logins	K	dst_host_count	AF
logged_in	L	dst_host_srv_count	AG
num_compromised	M	dst_host_same_srv_rate	AH
root_shell	N	dst_host_diff_srv_rate	AI
su_attempted	O	dst_host_same_src_port_rate	AJ
num_root	P	dst_host_srv_diff_host_rate	AK
num_file_creations	Q	dst_host_serror_rate	AL
num_shells	R	dst_host_srv_serror_rate	AM
num_access_files	S	dst_host_rerror_rate	AN
num_outbound_cmds	T	dst_host_srv_rerror_rate	AO
is_host_login	U		

ویژگی‌های مجموعه داده KDD cup99 استفاده شده در درخت عبارتند از:

{B,C,D,E,F,H,J,L,M,Q,V,W,AD,AF,AG,AH,AI,AJ,AK,AL,AO}

Downloaded from adst.ir at 14:58 +0430 on Monday July 24th 2017

لازم به ذکر است که برخی از ویژگی‌هایی که در درخت بدون هرس موجود هستند در درخت هرس شده موجود نیستند. واضح است که این ویژگی‌ها هرس شده‌اند. ویژگی‌های هرس شده عبارتند از: { N,P,AE,AM}

جهت انتخاب این ویژگی‌ها الگوریتم C4.5 بر روی تمام داده‌های آموزش اجرا شده و آن ویژگی‌هایی که در ساخت درخت مورد استفاده قرار گرفته‌اند، به عنوان ویژگی‌های مهم انتخاب شده است. سپس عمل گسسته‌سازی و اجرای الگوریتم‌های کلاسه بند بر روی این ویژگی‌های مهم صورت گرفته است. ویژگی‌های دیگری وجود دارند که در هر دو درخت موجود نیستند. دلیل انتخاب نشدن این ویژگی‌ها نداشتن اطلاعات کافی در هر بار جداسازی در زمان ساخت درخت است. در مورد مجموعه داده NSL-KDD نیز ویژگی‌های زیر انتخاب شده است:

{B,C,D,E,H,J,L,M,Q,R,V,X,AD,AF,AE,AG,AH,AI,AJ,AK,AO}

نتایج حاصل از این روش پیشنهادی بر روی الگوریتم‌های کلاسه‌بندی Naïve Bayes، SVM و CART بررسی شده‌اند. برای اجرای همگی این الگوریتم‌ها از روش 10-fold cross validation استفاده شده است. هر الگوریتم سه بار اجرا شده است. یک بار با مجموعه داده با ۴۱ ویژگی و بدون انجام پیش‌پردازش. بار دوم با حذف ویژگی‌های غیر مهم و استفاده از ۲۱ ویژگی مهم استفاده شده در ساخت درخت (روش پیشنهادی انتخاب ویژگی). بار سوم با ۲۱ ویژگی انتخاب شده توسط درخت تصمیم و گسسته‌سازی ویژگی‌های پیوسته بر اساس نقاط برش انتخاب شده در درخت تصمیم (مجتمع‌سازی انتخاب ویژگی و گسسته‌سازی). نتایج بر اساس پارامترهای زمان ساخت مدل و دقت با هم مقایسه شده‌اند. جدول‌های (۵-۳) نشان دهنده عملکرد روش پیشنهادی با الگوریتم‌های کلاسه‌بندی مختلف هستند.

لازم به ذکر است که رایانه مورد استفاده در تحلیل راهکار این پژوهش دارای مشخصات CPU: Core i5، Ram: 6 GB و Cache: 2GB بوده است.

۴-۳. تحلیل نتایج

ابتدا نتایج اجرای روش پیشنهادی بر روی مجموعه داده KDD cup99 بررسی می‌شود. جدول (۳) نشان دهنده دقت به‌دست آمده برای هر نوع حمله در الگوریتم SVM است. دقت این الگوریتم در تشخیص حمله نوع U2R برابر ٪۰/۰۷ است که نرخ بسیار پایینی است. در واقع این الگوریتم تقریباً قادر نیست حملات از این نوع را پیش‌بینی کند. بنابراین افزایش دقت تشخیص این الگوریتم برای این نوع حمله، مهم و ضروری است. پس از انتخاب ویژگی‌های مهم توسط روش پیشنهادی دقت الگوریتم SVM در حملات نوع U2R تغییر زیادی نمی‌کند. این مسئله نشان می‌دهد که روش پیشنهادی توانسته است ویژگی‌های افزونه را در تشخیص این حمله حذف کند. پس از مجتمع‌سازی روش گسسته‌سازی و انتخاب ویژگی، دقت الگوریتم SVM برای تشخیص حمله U2R به ٪۲۳/۷ افزایش پیدا می‌کند.

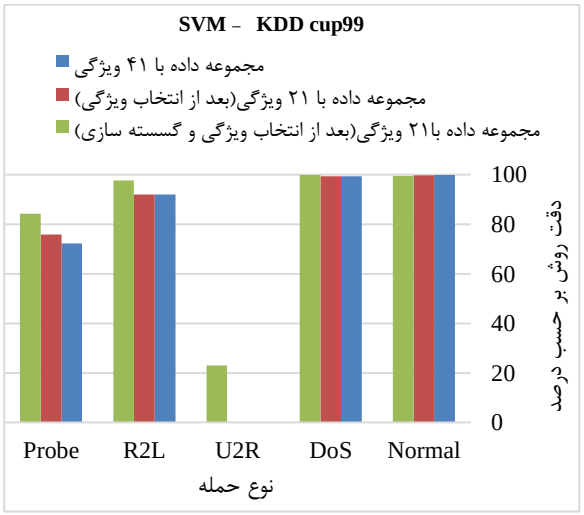
به طور کلی دقت کلی الگوریتم SVM با ۴۱ ویژگی و ۲۱ ویژگی انتخاب شده و بدون گسسته‌سازی تقریباً یکسان هستند و تفاوت قابل توجهی ندارند. این بدین معنی است که ویژگی‌های حذف شده ویژگی‌های افزونه بوده‌اند. همان‌طور که در شکل (۳) هم می‌توان دید این ویژگی‌ها در دقت الگوریتم تأثیری نداشته‌اند و حذف آن‌ها دقت مدل را کاهش نمی‌دهد، اما باعث می‌شود که زمان ساخت مدل کاهش قابل توجهی داشته باشد. همچنین پس از تجمیع هر دو روش گسسته‌سازی و انتخاب ویژگی، دقت پیش‌بینی تمام حملات افزایش پیدا کرده است؛ به غیر از دقت پیش‌بینی داده‌های نرمال که کاهش اندکی داشته است و از ٪۹۹/۹۱ به ٪۹۹/۵۳ کاهش یافته است. بنابراین روش پیشنهادی دقت الگوریتم SVM را در پیش‌بینی همه حملات افزایش می‌دهد.

نتایج نشان می‌دهد که مجتمع کردن روش گسسته‌سازی و انتخاب ویژگی می‌تواند تأثیر قابل توجهی در زمان مورد نیاز برای ساخت مدل در الگوریتم SVM داشته باشد. به طور کلی انجام پیش‌پردازش به روش پیشنهادی و استفاده از الگوریتم SVM به عنوان کلاسه بند، دو پارامتر اصلی در سامانه‌های تشخیص نفوذ را تحت تأثیر قرار می‌دهد. یکی دقت پیش‌بینی حملات را افزایش می‌دهد و دیگری زمان ساخت مدل کاهش می‌یابد.

جدول ۳. نتایج الگوریتم SVM، مجموعه داده KDD cup

نوع حمله	مجموعه داده با ۴۱ ویژگی	مجموعه داده با ۲۱ ویژگی (بعد از انتخاب ویژگی)	مجموعه داده با ۲۱ ویژگی (بعد از انتخاب ویژگی و گسسته‌سازی)
Normal	۹۸/۷۹	۹۸/۷۶	۹۸/۹۱
DoS	۹۹/۳۵	۹۹/۸۸	۹۹/۹۲
U2R	۶۶/۱۱	۷۳/۷۷	۷۸/۸۶
R2L	۹۷/۳۵	۹۷/۹۱	۹۹/۲۵
Probe	۹۴/۷۱	۹۴/۸۷	۹۳/۶۹
دقت کل (درصد)	۹۸/۹۴	۹۸/۶۲	۹۹/۵۲
زمان ساخت مدل (ثانیه)	۶/۷	۵/۵۸	۶/۲

Downloaded from adst.ir at 14:58 +0430 on Monday July 24th 2017



شکل ۳. نتایج الگوریتم SVM، مجموعه داده KDD cup99

نتایج استفاده از الگوریتم Naïve Bayes، به عنوان الگوریتم کلاسه بند با روش پیشنهادی در فاز پیش‌پردازش، در جدول (۴) نشان داده شده است. کلاسه بندی داده‌ها با مجموعه داده با ۲۱ ویژگی باعث می‌شود که دقت پیش‌بینی داده‌های نرمال و همه حملات افزایش پیدا کند. تنها دقت پیش‌بینی حملات U2R کاهش پیدا می‌کند و از ٪۸۸/۴۶ به ٪۷۳/۰۷ کاهش می‌یابد. این کاهش دقت نشان می‌دهد که برخی از ویژگی‌های حذف شده ویژگی‌های مهم در پیش‌بینی این نوع حمله بوده‌اند، که حذف آن‌ها موجب کاهش دقت این الگوریتم در تشخیص حملات U2R شده است.

جدول ۴. نتایج الگوریتم Naïve Bayes، مجموعه داده KDD cup

نوع حمله	مجموعه داده با ۴۱ ویژگی	مجموعه داده با ۲۱ ویژگی (بعد از انتخاب ویژگی)	مجموعه داده با۲۱ ویژگی (بعد از انتخاب ویژگی و گسسته‌سازی)
Normal	۸۶/۶۴	۸۹/۳۷	۹۳/۸۲
DoS	۹۳/۶۵	۹۷/۳	۹۸/۸۵
U2R	۸۸/۴۶	۷۳/۰۷	۷۸/۸۴
R2L	۲۲/۶۶	۵۹/۸۴	۹۶/۰۹
Probe	۸۴/۲۵	۹۵/۳۷	۱۰۰
دقت کل	۸۶/۹۸	۹۲/۰۸	۹۷/۶۶
زمان ساخت مدل (ثانیه)	۰/۰۸	۰/۰۲	۰/۰۱

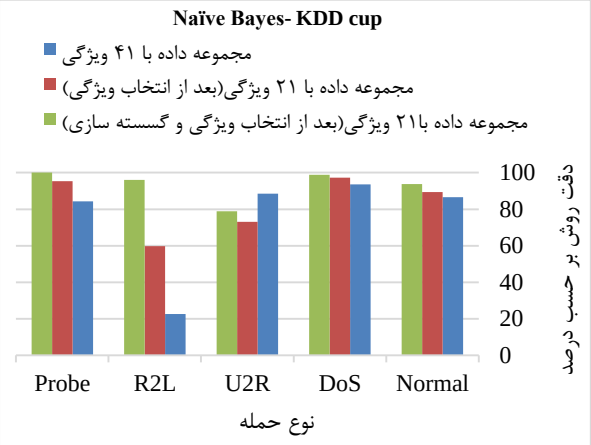
بیشترین افزایش دقت پس از انتخاب ویژگی‌های مهم، مربوط به حملات از نوع R2L و Probe است. دقت پیش‌بینی حملات R2L از ٪۲۲/۶۶ به ٪۵۹/۸۴ می‌رسد. همچنین دقت پیش‌بینی حملات Probe از ٪۸۴/۲۵ به ٪۹۵/۳۷ افزایش پیدا می‌کند. این نتایج نشان می‌دهد که ویژگی‌های نامربوط در پیش‌بینی حملات (به جز حملات از نوع U2R) و داده‌های نرمال حذف شده‌اند که همین مسئله باعث افزایش دقت الگوریتم شده است.

پس از مجتمع سازی روش گسسته‌سازی و انتخاب ویژگی، دقت پیش‌بینی تمام حملات و داده‌های نرمال نسبت به مجموعه داده با ۲۱ ویژگی افزایش پیدا می‌کند. با اینکه مجتمع سازی دو روش پیش‌پردازش باعث می‌شود که دقت پیش‌بینی حملات U2R نسبت به مجموعه داده با ۲۱ ویژگی از ٪۷۳/۰۷ به ٪۷۸/۸۴ افزایش پیدا کند. اما باز هم دقت تشخیص این حملات در مقایسه با مجموعه داده بدون پیش‌پردازش با ۴۱ ویژگی کمتر است.

پس از مجتمع کردن گسسته‌سازی و انتخاب ویژگی دقت پیش‌بینی حملات Probe به ٪۱۰۰ می‌رسد. بنابراین تمامی حملات Probe توسط روش پیشنهاد شده و استفاده از الگوریتم Naïve Bayes به عنوان کلاسه بند قابل‌شناسایی هستند. همچنین دقت پیش‌بینی حملات R2L نسبت به مجموعه داده با ۴۱ ویژگی افزایش قابل توجهی داشته است و از ٪۲۲/۶۶ به ٪۹۶/۰۹ رسیده است. دقت کلی الگوریتم Naïve Bayes بدون پیش‌پردازش برابر با ٪۸۶/۹۸ است. پس از انتخاب ویژگی‌های مهم دقت کلی این الگوریتم به ٪۹۲/۰۸ می‌رسد و درنهایت با مجتمع کردن گسسته‌سازی و انتخاب ویژگی‌ها، دقت کلی الگوریتم به ٪۹۷/۶۶ افزایش پیدا می‌کند. برای ساخت مدل با الگوریتم Naïve Bayes در مجموعه داده با ۴۱ ویژگی به ۰/۰۸ ثانیه نیاز است که زمان بسیار کمی است. اما با کاهش ویژگی‌ها به ۲۱ ویژگی زمان مورد نیاز برای ساخت مدل نیز کاهش پیدا می‌کند و به ۰/۰۲ ثانیه می‌رسد. پس از مجتمع سازی دو روش پیش‌پردازش زمان به ۰/۰۱ ثانیه کاهش پیدا می‌کند. نتایج استفاده از روش پیشنهادی با الگوریتم درخت تصمیم CART به عنوان کلاسه بند در جدول (۵) نمایش داده شده است. زمانی که از مجموعه داده با ۲۱ ویژگی استفاده می‌شود، دقت حملات از نوع U2R و Probe به ترتیب از ٪۶۵/۵۸ و ٪۹۳/۵۱ به ٪۷۳/۰۷ و ٪۹۵/۳۷ افزایش پیدا می‌کند.

دقت داده‌های نرمال و بقیه حملات تقریباً مشابه مجموعه داده با ۴۱ ویژگی است. این نتایج تصدیق می‌کند که ویژگی‌های افزونه که در ساخت مدل تأثیری نداشته‌اند، حذف شده‌اند. چون پس از حذف ویژگی‌ها دقت کلی الگوریتم همان‌گونه که در شکل (۴) ملاحظه می‌شود تغییر قابل توجهی نداشته است و از ٪۹۹/۳۴ به ٪۹۹/۳۹ رسیده است.

Downloaded from adst.ir at 14:58 +0430 on Monday July 24th 2017



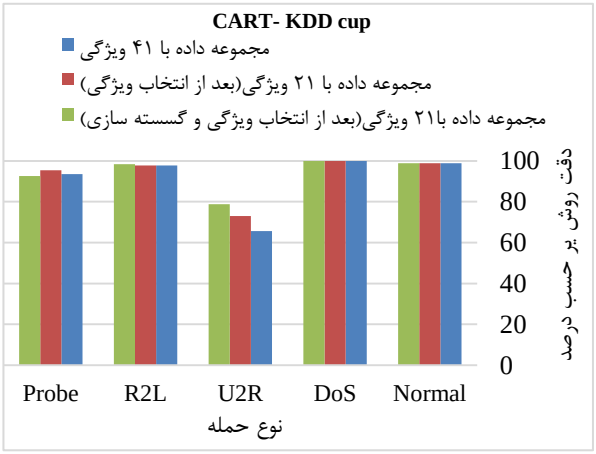
شکل ۴. نتایج الگوریتم Naïve Bayes، مجموعه داده KDD cup99

پس از مجتمع سازی گسسته‌سازی و انتخاب ویژگی، دقت حملات U2R و R2L نسبت به مجموعه داده بدون پیش‌پردازش افزایش پیدا کرده است و به ترتیب از ٪۶۵/۵۸ و ٪۹۶/۶۹ به ٪۷۸/۸۴ و ٪۹۸/۳۱ رسیده است. همچنین دقت تشخیص حملات Probe کاهش اندکی داشته است و از ٪۹۳/۵۱ به ٪۹۲/۵۹ رسیده است. دقت حملات DoS و نرمال بهبود قابل توجهی نداشته است.

به طور کلی اگر دقت کلی الگوریتم مد نظر قرار داده شود، پس از مجتمع سازی دو روش پیش‌پردازش، دقت از ٪۹۹/۳۴ به ٪۹۹/۴۶ افزایش پیدا می‌کند. از نظر پارامتر زمان، کاهش ویژگی‌های به ۲۱ ویژگی باعث می‌شود که زمان ساخت مدل نسبت به حالت با ۴۱ ویژگی از ۶/۶۴ ثانیه به ۵/۲۸ ثانیه کاهش پیدا کند. مدت زمان مورد نیاز برای ساخت مدل پس از مجتمع سازی دو روش برابر با ۶/۲۹ ثانیه است که از مجموعه داده با ۴۱ ویژگی اندکی کمتر است و از مجموعه داده با ۲۱ ویژگی بیشتر است. با توجه به اینکه پس از مجتمع سازی دو روش، دقت الگوریتم CART، نسبت به مجموعه داده با ۲۱ ویژگی که گسسته‌سازی روی آن انجام نشده است بهبود قابل توجهی نداشته است و زمان ساخت مدل نیز افزایش پیدا کرده است، می‌توان بیان کرد که گسسته‌سازی با روش پیشنهادی تأثیر قابل توجهی روی بهبود الگوریتم CART در هر دو پارامتر دقت و زمان ندارد. اما استفاده از روش انتخاب ویژگی پیشنهاد شده با الگوریتم CART باعث بهبود پارامتر زمان می‌شود و پارامتر دقت نیز بهبود اندکی داشته است. بنابراین ویژگی‌های افزونه توسط روش پیشنهاد شده حذف شده‌اند. این نتایج در جدول (۵) نشان داده شده است. نتایج اجرای الگوریتم CART بر روی مجموعه داده KDD cup در شکل (۵) نشان می‌دهد که روش پیشنهادی بهبود بسیار ناچیزی در دقت ایجاد کرده است. به طور مشابه الگوریتم بر روی مجموعه داده NSL-KDD اجرا شد که نتایج در جدول‌های (۶-۸) و شکل‌های (۶-۸) ارائه شده است.

جدول ۵. نتایج الگوریتم CART، مجموعه داده KDD cup

نوع حمله	مجموعه داده با ۴۱ ویژگی	مجموعه داده با ۲۱ ویژگی (بعد از انتخاب ویژگی)	مجموعه داده با ۲۱ ویژگی (بعد از انتخاب ویژگی و گسسته‌سازی)
Normal	۹۸/۸۲	۹۸/۸۲	۹۸/۸۶
DoS	۹۹/۹	۹۹/۹۱	۹۹/۹۲
U2R	۶۵/۵۸	۷۳/۰۷	۷۸/۸۴
R2L	۹۷/۶۹	۹۷/۶۹	۹۸/۳۱
Probe	۹۳/۵۱	۹۵/۳۷	۹۲/۵۹
دقت کل (درصد)	۹۹/۳۴	۹۹/۳۹	۹۹/۴۶
زمان ساخت مدل (ثانیه)	۶/۶۴	۵/۲۸	۶/۲۹

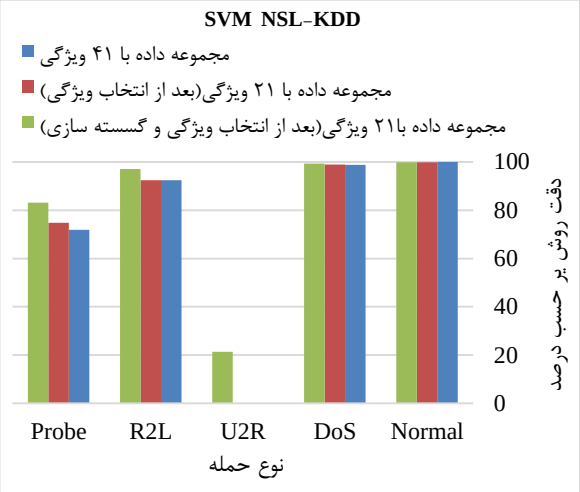


شکل ۵. نتایج الگوریتم CART، مجموعه داده KDD cup99

جدول ۶. نتایج الگوریتم SVM، مجموعه داده NSL-KDD

نوع حمله	مجموعه داده با ۴۱ ویژگی	مجموعه داده با ۲۱ ویژگی (بعد از انتخاب ویژگی)	مجموعه داده با۲۱ ویژگی (بعد از انتخاب ویژگی و گسسته‌سازی)
Normal	۹۹/۹	۹۹/۸۳	۹۹/۷۹
DoS	۹۸/۷۶	۹۸/۹۳	۹۹/۲۶
U2R	۰/۰۷	۰/۱۱	۲۱/۳۵
R2L	۹۲/۳۶	۹۲/۳۵	۹۷/۰۹
Probe	۷۱/۸۵	۷۴/۷۳	۸۳/۱۱
دقت کل (درصد)	۹۸/۳۳	۹۸/۳۴	۹۹/۲۶
زمان ساخت مدل (ثانیه)	۳۵/۸۸	۱۶/۴	۲/۰۹

Downloaded from adst.ir at 14:58 +0430 on Monday July 24th 2017

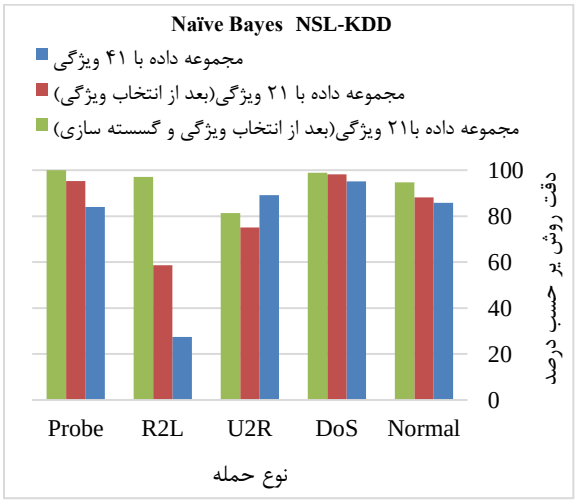


شکل ۶. نتایج الگوریتم SVM، مجموعه داده NSL-KDD

با ملاحظه جدول (۶) و شکل (۶)، به طور مشابه با مجموعه داده KDD cup ملاحظه می‌شود که در مجموعه داده NSL-KDD روش پیشنهادی با استفاده از الگوریتم SVM بهبود نسبی داشته است. با مشاهده جدول (۷) و شکل (۷) مشاهده می‌شود که اجرای روش پیشنهادی بر روی مجموعه داده NSL-KDD بهبود مناسبی را به خاطر استفاده از الگوریتم Naïve Bayes ایجاد کرده است. شکل (۷) داده‌های جدول (۷) را با وضوح بیشتری نشان می‌دهد. جدول (۸) و شکل (۸) نتیجه اجرای روش پیشنهادی به کمک الگوریتم CART بر روی مجموعه داده NSL-KDD است که بهبود ناچیزی را نشان می‌دهد.

جدول ۷. نتایج الگوریتم Naïve Bayes، مجموعه داده NSL-KDD

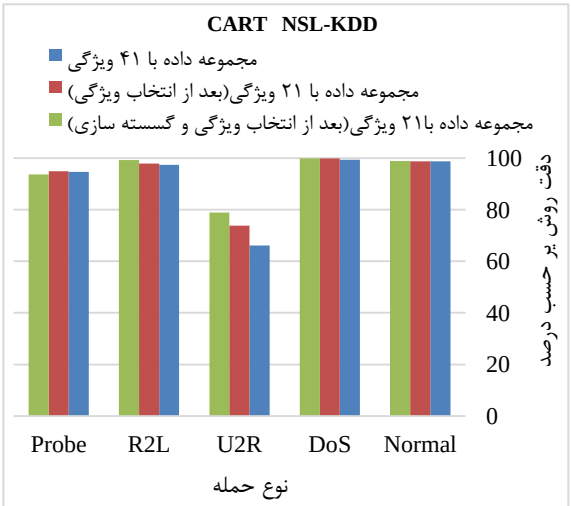
نوع حمله	مجموعه داده با ۴۱ ویژگی	مجموعه داده با ۲۱ ویژگی (بعد از انتخاب ویژگی)	مجموعه داده با۲۱ ویژگی (بعد از انتخاب ویژگی و گسسته‌سازی)
Normal	۸۵/۸۳	۸۸/۲۶	۹۴/۷۵
DoS	۹۵/۱۱	۹۸/۲۱	۹۸/۹۳
U2R	۸۹/۲۳	۷۵/۰۸	۸۱/۳۴
R2L	۲۷/۴۶	۵۸/۷۱	۹۷/۱۱
Probe	۸۴/۰۶	۹۵/۳۱	۹۹/۹۸
دقت کل (درصد)	۸۵/۳۷	۹۴/۱۷	۹۶/۶۲
زمان ساخت مدل (ثانیه)	۰/۰۸	۰/۰۳	۰/۰۱



شکل ۷. نتایج الگوریتم Naïve Bayes، مجموعه داده NSL-KDD

جدول ۸. نتایج الگوریتم CART، مجموعه داده NSL-KDD

نوع حمله	مجموعه داده با ۴۱ ویژگی	مجموعه داده با ۲۱ ویژگی (بعد از انتخاب ویژگی)	مجموعه داده با۲۱ ویژگی (بعد از انتخاب ویژگی و گسسته‌سازی)
Normal	۹۸/۷۹	۹۸/۷۶	۹۸/۹۱
DoS	۹۹/۳۵	۹۹/۸۸	۹۹/۹۲
U2R	۶۶/۱۱	۷۳/۷۷	۷۸/۸۶
R2L	۹۷/۳۵	۹۷/۹۱	۹۹/۲۵
Probe	۹۴/۷۱	۹۴/۸۷	۹۳/۶۹
دقت کل (درصد)	۹۸/۹۴	۹۸/۶۲	۹۹/۵۲
زمان ساخت مدل (ثانیه)	۶/۷	۵/۵۸	۶/۲



شکل ۸. نتایج اجرای الگوریتم CART، مجموعه داده NSL-KDD

Downloaded from adst.ir at 14:58 +0430 on Monday July 24th 2017

با مقایسه نتایج اجرای الگوریتم پیشنهادی بر روی دو مجموعه داده KDD cup و NSL-KDD ملاحظه می‌شود که هر چند دقت نتایج اجرای روش پیشنهادی بر روی دو مجموعه داده تا حدودی تفاوت دارد اما دقت بالاتری را نسبت به زمانی که ۴۱ ویژگی و یا ۲۱ ویژگی بدون عمل گسسته‌سازی مورد استفاده قرار گیرد ایجاد کرده است. در جدول (۹) نتایج حاصل از این پژوهش را با پژوهش‌های مشابه مقایسه شده است. همچنین به توجه به جدول‌های (۵ و ۸) مشخص است که روش پیشنهادی نسبت به الگوریتم مینایی و همکاران [۱۹]، از نظر دقت الگوریتم و زمان اجرا نتایج مطلوب‌تری دارد. نتایج روش پیشنهادی با نتایج پژوهش هانچوان و همکاران [۲۰] که محدود به داده‌های پیوسته است شباهت زیادی دارد اما روی داده‌های گسسته و پیوسته قابل‌اجرا است.

جدول ۹. مقایسه نتایج روش پیشنهادی از نظر دقت و زمان اجرا

الگوریتم	زمان اجرای الگوریتم (ثانیه)	دقت الگوریتم (درصد)
روش پیشنهادی با الگوریتم SVM و مجموعه داده KDD cup99	۲/۱۳	۹۹/۲۵
روش پیشنهادی با الگوریتم Naïve Bayes و مجموعه داده KDD cup99	۰/۰۱	۹۷/۶۶
روش پیشنهادی با الگوریتم CART و مجموعه داده KDD cup99	۶/۲۹	۹۹/۴۶
روش پیشنهادی با الگوریتم SVM و مجموعه داده NSL-KDD	۲/۰۹	۹۹/۲۶
روش پیشنهادی با الگوریتم Naïve Bayes و مجموعه داده NSL-KDD	۰/۰۱	۹۶/۶۲
روش پیشنهادی با الگوریتم CART و مجموعه داده NSL-KDD	۶/۲	۹۹/۵۲
الگوریتم پیشنهادی ماندیانی و همکاران (بر روی مجموعه داده KDD cup99) [۱۳]	۱۱/۳	حداکثر دقت ۹۵/۸
الگوریتم پیشنهادی آمریتا (مجموعه داده KDD cup99) [۲۱]	۰/۰۴	حداکثر دقت ۹۷/۳۵

۵. نتیجه‌گیری

در این پژوهش یک روش پیش‌پردازش برای سامانه‌های تشخیص نفوذ ارائه شده است. تا کنون در تحقیقات پیشین از یکی از روش‌های پیش‌پردازش مثل انتخاب ویژگی‌های یا گسسته‌سازی استفاده شده است. روش پیشنهادی، استفاده از گسسته‌سازی و انتخاب ویژگی‌ها به صورت هم‌زمان است. نتایج آزمایش بر روی مجموعه داده‌های استاندارد KDD cup99، NSL-KDD و سه الگوریتم کلاسه بند، کارایی روش پیشنهادی را نشان می‌دهد. به بیان دیگر بهبود دقت و کاهش زمان، دو چالش مهم در سامانه‌های تشخیص نفوذ هستند که بسیاری از محققین در این حوزه به دنبال ارائه راه‌حلی برای آن‌ها هستند. نتایج تجربی نشان می‌دهد که روش پیشنهادی توانسته است پارامترهای دقت و زمان را بهبود بدهد. استفاده از روش پیشنهادی باعث افزایش دقت در کلاسه بندهای SVM، Naïve Bayes و CART می‌شود. بهبود پارامتر زمان نیز در سامانه‌های تشخیص نفوذ بسیار مهم و اساسی است. چون سامانه‌های تشخیص نفوذ به صورت آنلاین کار می‌کنند، تشخیص حملات باید به صورت بلادرنگ و سریع باشد. همین مسئله اهمیت کاهش زمان برای پیش‌بینی حمله یا نرمال بودن ارتباطات را مشخص می‌کند. پارامتر زمان نیز با انجام پیش‌پردازش بر اساس روش پیشنهادی بهبود می‌یابد. پیشنهاد می‌شود که در پژوهش‌های آینده ترکیب روش انتخاب ویژگی پیشنهادی با دیگر روش‌های گسسته‌سازی بررسی شود. همچنین می‌توان از کلاسه بندهای دیگر در پژوهش‌های آینده برای پیش‌بینی حملات استفاده کرد و پارامترهای ارزیابی مدل را تحلیل کرد.

۶. مراجع

[1] Shirazi, H.; Jamalyfard, A.; Farshchi, S. M. R. "Detection of Attacks against Web Applications Using Combination of One-Class Classifiers"; Advanced Defence Sci. & Tech. 2014, 5, 107-119 (In Persian).

[2] Folino, G.; Sabatino, P. "Ensemble Based Collaborative and Distributed Intrusion Detection Systems: A Survey"; J. Network Comput. Appl. 2016, 66, 1-16.

[3] Garcia-Teodoroa, P.; Diaz-Verdejoa, J.; Macia´-Ferna´ndeza, G.; Va´zquezb, E. "Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges"; Computers & Security 2009, 28, 18-28.

[4] Al-Nashif, Y.; Kumar, A. A.; Hariri, S.; Luo, Y.; Szidarovsky, F.; Qu, G. "Multi-level Intrusion Detection System (ml-ids)"; Proc. Autonomic Computing 2008, 131-140.

[5] Ahmed, M.; Mahmood, A. N.; Hu, J. "A Survey of Network Anomaly Detection Techniques"; J. Network Comput. Appl. 2016, 60, 19-31.

[6] Berka, P.; Bruha, I. "Discretization and Grouping: Preprocessing Steps for Data Mining"; Proc. Principles of Data Mining and Knowledge Discovery 1998, 239-245.

[7] Yang, Y.; Webb, G. I.; Wu, X. "Discretization Methods"; Data Mining and Knowledge Discovery Handbook, 2005, 113-130.

[8] Garcia, S.; Luengo, J.; Sáez, J. A.; Lopez, V.; Herrera, F. "A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning"; IEEE Trans. Knowledge and Data Eng. 2013, 25, 734-750.

[9] Piramuthu, S. "Evaluating Feature Selection Methods for Learning in Data Mining Applications"; European J. Operational Res. 2004, 156, 483-494.

[10] Dash, M.; Liu, H. "Feature Selection for Classification"; Intelligent Data Anal. 1997, 1, 131-156.

Downloaded from adst.ir at 14:58 +0430 on Monday July 24th 2017

[11] Mukkamala, S.; Sung, A. H. “Significant Feature Selection using Computational Intelligent Techniques for Intrusion Detection”; Advanced Methods for Knowledge Discovery from Complex Data 2005, 285-306.

[12] Varma, P. R.; Kumari, V. V.; Kumar, S. S. “Feature Selection Using Relative Fuzzy Entropy and Ant Colony Optimization Applied to Real-Time Intrusion Detection System”; Procedia Computer Sci. 2016, 85, 503-510.

[13] Muniyandi, A. P.; Rajeswari, R.; Rajaram, R. “Network Anomaly Ddetection by Cascading k-Means Clustering and C4. 5 Decision Tree Algorithm”; Procedia Eng. 2012, 30, 174-182.

[14] Hidayati, R.; Kanamori, K.; Feng, L.; Ohwada, H. “Combining Feature Selection with Decision Tree Criteria and Neural Network for Corporate Value Classification”; Proc. Pacific Rim Knowledge Acquisition 2016, 31-42.

[15] Sung, A. H.; Mukkamala, S. “Identifying Important Features for Intrusion Detection using Support Vector Machines and Neural Networks”; Proc. Applications and the Internet 2003, 209-216.

[16] Namazi, M.; Shokrolahi, A.; Maharluie, MS. “Detecting and Ranking Cash Flow Risk Factors via Artificial Neural Networks Technique”; J. Business Res. 2016, 69, 1801-1806.

[17] Speybroeck, N. “Classification and Regression Trees”; Int. J. Public Health 2012, 57, 243-246.

[18] Ektefa, M.; Memar, S.; Sidi, F.; Affendey, L. S. “Intrusion Detection using Data Mining Techniques”; Proc. Information Retrieval & Knowledge Management 2010, 200-203.

[19] Bidgoli, B. M.; Analoui, M.; Rezvani, M. H.; Shahhoseini, H. S. “Performance Evaluation of Decision Tree for Intrusion Detection Using Reduced Feature Spaces”; Proc. Trends in Intelligent Syst. and Computer Eng. 2008, 273-284.

[20] Peng, H.; Long, F.; Ding, C. “Feature Selection Based on Mutual Information Criteria of Max-dependency, Max-relevance, and Min-Redundancy”; IEEE Trans. Pattern Analysis and Machine Intelligence 2005, 27, 1226-1238.

[21] Aggarwal, M. A. “Performance Analysis of Different Feature Selection Methods in Intrusion Detection”; Int. J. Scientific & Tech. Res. 2013, 2, 20-30.

[22] Fayyad, U.; Irani, K. “Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning”; Proc. International Joint Conference on Artificial Intelligence 1993, 1022-1029.

[23] Eid, H.F.; Azar, A. T.; Hassanien, A. E. “Improved Real-Time Discretize Network Intrusion Detection System”; Proc. Bio-Inspired Computing: Theories and Applications, 2012, 99-109.

[24] Ramírez-Gallego, S.; García, S.; Benítez, J. M.; Herrera, F. “A Wrapper Evolutionary Approach for Supervised Multivariate Discretization: A Case Study on Decision Trees”; Proc. Computer Recognition Syst. 2016, 47-58.

[25] Aziz, A. S.; Azar, A. T.; Hassanien, A. E.; Hanafy, S. E. “Continuous Features Discretization for Anomaly Intrusion Detectors Generation”; Soft Computing in Industrial Applications 2014, 209-221.

[26] Esposito, F.; Malerba, D.; Semeraro, G.; Kay, J. “A Comparative Analysis of Methods for Pruning Decision Trees”; IEEE Transactions Pattern Analysis and Machine Intelligence 1997, 19, 476-491.

[27] Quinlan, J. R. “Simplifying Decision Trees”; Int. J. Man-Machine Studies 1987, 27, 221-234.

[28] Zimmerman, R. K.; Balasubramani, G. K.; Nowalk, M. P.; Eng, H.; Urbanski, L.; Jackson, M. L.; Jackson, L. A.; McLean, H.Q.; Belongia, E. A.; Monto, A. S.; Malosh, R. E. “Classification and Regression Tree (CART) Analysis to Predict Influenza in Primary Care Patients”; BMC Infectious Diseases 2016, 16, 503-515.

[29] Mingers, J. “Expert Systems-Rule Induction with Statistical Data”; J. Oper. Res. Soc. 1987, 38, 39-47.

[30] Malerba, D.; Esposito, F.; Semeraro, G. “A further Comparison of Simplification Methods for Decision-tree Induction”; Learning from Data 1996, 365-374.

[31] Mingers, J. “An Empirical Comparison of Pruning Methods for Decision Tree Induction”; Machine Learning 1989, 4, 227-243.

[32] Windeatt, T.; Ardeshir, G. “An empirical Comparison of Pruning Methods for Ensemble Classifiers”; Proc. Intelligent Data Anal. 2001, 208-217.

[33] Beck, J. R.; Garcia, M. E.; Zhong, M.; Georgiopoulos, M.; Anagnostopoulos, G. “A Backward Adjusting Strategy for the C4. 5 Decision Tree Classifier”; Technical Report, 2007.

[34] “KDD cup 99 intrusion Detection Data Set”.

[35] Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A. A. “A Detailed Analysis of the KDD CUP 99 Data Set”; Proc. Computational Intelligence for Security and Defense Applications 2009, 1-6.