انتخاب بردارهای پشتیبان مناسب در SVM افزایشی برای کشف نفوذ شبکه

چکیده

در یک دسته‌بندی SVM (ماشین بردار پشتیبان) افزایشی، داده‌های با بردارهای غیر پشتیبان عنوان بندی می-شوند(لیبل گذاری) که توسط دسته قبلی به عنوان داده‌های تمرینی در دسته بعدی همراه با داده‌های تایید شده توسط شرط KarusheKuhneTucker (شرایط کان تاکر) مورد استفاده قرار گرفته می‌شوند. این مقاله استراتژی نیمه تقسیم‌بندی انتخاب و حفظ بردارهای غیرپشتیبان از افزایش فعلی دسته بندی- که به عنوان بردارهای پشتیبان مناسب نامگذاری شدند(CSV)- را پیشنهاد می‌دهد که به احتمال زیاد به بردارهای پشتیبان در توسعه دسته بعدی تبدیل می‌شوند. همچنین این کار تحقیقاتی الگوریتمی را طراحی می‌کند که به عنوان بردار پشتیبان مناسب نامگذاری می‌شود که براساس الگوریتم SVM (CSV-ISVM) افزایشی می‌باشد به طوریکه استراتژی پیشنهادی را پیاده‌سازی می‌کند و کل فرایند دسته‌بندی SVM افزایشی تحقق می‌بخشد. این کار نیز تغییراتی را برای روش حلقه متمرکز پیشنهاد شده قبلی و استراتژی دسته معکوس پیشنهاد می‌دهد. عملکرد روش پیشنهادی با آزمایشات و همچنین مقایسه آن با تکنیکهای SVM دیگر ارزیابی می‌شود. نتایج تجربی و تحلیل عملکرد نشان می‌دهند که برای کشف نفوذ شبکه در زمان واقعی الگوریتم پیشنهادی CSV-ISVM از دسته‌بندی‌های کلی ISVM بهتر می‌باشد.

1- معرفی

تشخیص نفوذ شبکه به عنوان یک طرح تشخیص مشکل برای طبقه بندی الگوهای ترافیک شبکه در دو کلاس – نرمال و غیرنرمال- در نظر گرفته می‌شود؛ باتوجه به شباهت بین آنها. امروزه در زمینه تشخیص نفوذ، ماشین بردار پشتیبان (svm) تبدیل به یک ابزار طبقه‌بندی محبوب شده است که براساس فراگیری ماشین آماری می‌باشد (Mohammad et al., 2011). دو موضوع در فراگیری-تمرین ماشین برای مجموعه داده‌های بزرگ و در

دسترس‌پذیری یک مجموعه داده کامل وجود دارد. اگر مجموعه داده‌های تمرینی بسیار بزرگ باشند حافظه کامپیوتر کافی نخواهد بود و زمان تمرین بسیار طولانی خواهد بود. سپس، وقتی ما بسته‌های داده را از یک جریان شبکه می-گیریم، ما نمی‌توانیم اطلاعات کامل شبکه را در اولین فرصت به دست آوریم و بنابراین با افزایش تعداد نمونه‌ها، یک فراگیری آنلاین مستمر برای بالابردن دقت موردنیاز می‌باشد. چالش فراگیری افزایشی برای این است تا تصمیم بگیریم که چه چیزی و چه میزان اطلاعات از فراگیری قبلی باید برای تمرین در مرحله فراگیری بعدی انتخاب شود و چگونه با مجموعه داده‌های جدیدی که در آن مرحله اضافه می‌شوند باید رفتار کرد. بنابراین کلید فراگیری افزایشی این است که با مجموعه داده‌های اضافه شده مقابله کنیم بطوریکه اطلاعات نمونه داده اصلی ضمنا حفظ شود.

اکثر روشهای تشخیص نفوذ از الگوریتمهای فراگیری غیر افزایشی استفاده می‌کنند. با جمع‌آوری نمونه داده‌های جدید، زمان تمرین‌شان مداوم افزایش می‌یابد، و در همان زمان، آنها مشکلاتی در تنظیم خودشان با تغییرات محیط شبکه دارند. در این مورد، فراگیری افزایشی این قابلیت را دارد که از نمونه‌های جدید سریع یادبگیرد و مدل اصلی خودشان را اصلاح کنند. روشهای فراگیری افزایشی می‌توانند بهتر نیازهای تشخیص نفوذ در زمان واقعی را برآورده کرده، و همچنین دقت محاسباتی برنامه‌های زمان حقیقی را می‌توانند بهبود بخشند.

یک الگوریتم ماشین بردار پشتیبان افرایشی ساده بردارهای پشتیبان را با تمرین مجموعه داده اولیه به دست می-آورد. مجموعه داده‌های جدید و بردارهای پشتیبان قبلی درهم ادغام می‌شوند تا یک مجموعه نمونه جدید را بسازند، و آنها را تمرین می‌دهند تا بردارهای پشتیبان جدید تولید کنند. این فرایند تا مجموعه داده آخر تکرار می‌شود. تغییرات مجموعه داده‌های جدید که منجر به تولید بردارهای پشتیبان می‌شوند، بااستفاده از نظریه KKT (کان تاکر) تست می‌شوند. نمونه‌هایی که شرایط KKT را نقض می‌کنند ممکن است مجموعه بردار پشتیبان قبلی را تغییر دهند، از اینرو آنها به مجموعه داده‌های قبلی افزوده می‌شوند. نمونه‌هایی که در شرایط KKT صدق می‌کنند حذف می‌شوند چون آنها مجموعه بردار پشتیبان قبلی را تغییر نمی‌دهند.

این مقاله یک رویکرد فراگیری کارآمد و بهبود یافته از ماشین بردار پشتیبان را پیشنهاد می‌دهد که براساس ایده حفظ نمونه داده‌های اصلی و فعلی در کل فرایند فراگیری می‌باشد. در این رویکرد پیشنهادی، نقاط داده، که

بردارهای پشتیبان در افزایش دسته‌بندی فعلی نیستند اما دارای شانس تبدیل شدن به بردارهای پشتیبان در افزایش بعدی دسته‌بندی می‌باشند، انتخاب شده و به عنوان بردارهای پشتیبان مناسب (CSV) حفظ می‌شوند به طوریکه آنها با مجموعه داده تمرینی جدید ترکیب می‌شوند که به تمرین بعدی اضافه می‌شوند. این رویکرد همچنین استراتژی نیمه تقسیم بندی (پارتیشن) را به عنوان بخشی از روش انتخاب CSV معرفی می‌کند و الگوریتمی را طراحی می‌کند که CSV-ISVM نامیده شده و از این استراتژی استفاده می‌کند.

بقیه مقاله به صورت زیر سازماندهی می‌شود. بخش 2 برخی کارهای تحقیقاتی مربوط به این مقاله را ارائه می‌دهد. در بخش 3،CSV مبتنی بر SVM افزایشی را معرفی کرده و ادبیات موضوعی (یعنی شرایط KKT و گردشهای ابرصفحه SVM) را توصیف کرده است. بخش 4 تغییر و اصلاح روش حلقه متمرکز را پیشنهاد داده و بخش 5 کار اصلی این مقاله ارتوضیح می‌دهد یعنی روش نیمه پارتیشن. تمام آزمایشهای این تحقیق در بخش 6 همراه با آنالیزهای لازم نشان داده می‌شوند. در انتها، بخش 7 نتیجه‌گیری مقاله و پیشنهاداتی برای تحقیقات آتی را نیز مطرح می‌کند.

2- کارهای تحقیقاتی مرتبط

بسیاری از کارهای تحقیقاتی که انجام شده‌اند روشهای فراگیری افزایشی را درنظر می‌گرفتند که از دسته‌بندی SVM استفاده می‌کنند، البته تغییرات مختلف نیز پیشنهاد شده‌اند. اکثر کارهای تحقیقاتی به نظر می‌رسند که بهبودهایی را برای عملکرد تشخیص، دقت وغیره پیشنهاد داده‌اند.

یک رویکرد مبتنی بر SVM افزایشی بنیادی که فراگیری دسته افزایشی با SVM نامیده می‌شود، توسط Liu و همکاران در سال 2004 پیشنهاد شد، که در آن فقط بردارهای پشتیبان برای افزایش بعدی حفظ شدند، در حالیکه تمام نمونه داده‌های دیگر حذف می‌شوند. در واقع، نمونه‌های حذف شده مقدار اطلاعات کمی درمورد دسته‌بندی را حمل می‌کنند. با افزودن مجموعه داده‌های جدید به افزایش‌های متوالی فرایند فراگیری، این نمونه داده‌ها ممکن است بردارهای پشتیبان یا بالعکس باشند. بنابراین در SVMافزایشی ساده مشابه این، دقت دسته‌بندی در افزایشهای بعدی به شدت مورد تاثیر قرار گرفته می‌شود.

WenJian و Wang توضیح دادند که نمونه داده‌های متکی به ابرصفحه نزدیک بعد از افزودن مجموعه تمرینی جدید فرصتهای بیشتری برای تبدیل شدن به بردارهای پشتیبان را داشتند (Wang, 2008). بنابراین، آنها الگوریتم فراگیری افزایشی مازاد را پیشنهاد دادند که نمونه‌های مازاد متکی به ابرصفحه نزدیک را حفظ کرده و آنها را به مجموعه داده‌های جدید در افزایش بعدی اضافه کردند تا چک کنند که آیا آنها به بردارهای پشتیبان تبدیل شدند.

حفظ بردارهای پشتیبان در هر افزایش دسته‌بندی SVM زمان کلی دسته‌بندی را افزایش می‌دهد. بنابراین، کاهش زمان اجرا در ابتدا بسیار مهم می‌شود. یک راه‌حل معمولی برای تسریع فراگیری SVM این است که برنامه‌ریزی کوادراتیک(درجه دوم) را به تعدادی زیرمطئله تجزیه کنید به طوریکه پیچیدگی کلی فراگیری SVM می‌تواند از $O(N^3)$ به $O(N^2)$کاهش یابد (Joachims, 1998; Platt, 1998). بااین حال، مقتی نقاط داده N بسیار زیاد باشد، زمان پیچیدگی هنوز رضایت بخش نیست و فراگیری نیاز به بهبود و پیشرفت بیشتر دارد (Zhang et al., 2009).

تعدادی از کارهای تحقیقاتی که انجام شده‌اند مربوط به کاهش زمان اجرای دسته‌بندی افزایشی SVM می‌باشند. یک رویکرد فراگیری افزایشی با SVM برای دسته‌بندی جریان داده شبکه توسط Yao و همکاران(2012) پیشنهاد شد، یک روش فراگیری افزایشی برای طراحی ساختار چند مدلی توسط Sun و Guo (2012) پیشنهاد شد، یک الگوریتم فراگیری افزایشی سریع برای SVM که روش اصلاح مجموعه فعال نامیده شد (Tao, 2006) و غیره تعداد کمی در بین چندین کار هستند.

دوباره،یک الگوریتم فراگیری افزایشی از براساس خوشه‌بندیSVM توسط Du و همکاران (a,b2009) پیشنهاد شد که این واقعیت را درنظر می‌گیرد که حدود بردارهای پشتیبان ممکن است در بردارهای پشتیبان بعد از افزدون نمونه‌های جدید تغییر کند. این کار همچنین از نظریه KKT در فرایند فراگیری استفاده کرد. مراکز خوشه‌ای به دست آمده از فرایند خوشه‌بندی برای شرایط KKT مورد بررسی قرار گرفتند و مراکز خوشه‌ای که این شرایط را نقض می‌کنند به مجموعه بردار پشتیبان در تمرین بعدی اضافه شدند. شرایط کافی و لازم برای نقض KKT توسط Wang, Zheng, Wu, and Zhang (Wang et al., 2006) داده شدند. آنها نشان داده و ثابت کردند که آیا

نمونه جدیدی مخالف شرایط KKT وجود داشتند، سپس بردارهای غیرپشتیبان SVM اصلی این فرصت را داشتند تا به بردارهای پشتیبان تبدیل شوند.

یک الگوریتم فراگیری افزایشی مبتنی بر شرایط KKT در این ادبیات (Wen-hua & Jian, 2001) پیشنهاد شد. در این روش، کل فرایند فراگیری به دو فرایند اولیه و افزایشی تقسیم شد. راه‌حل بهینه QP به هر نمونه اجازه داد تا در شرایط KKT صدق کند.

اگرچه SVM افزایشی مبتنی بر KKT در سرعت‌ها و میزان‌های تشخیص بهتر است، ولی به زمان تمرین طولانی‌تری نیاز دارد چون آن باید خلاف قضاوت را انجام دهد و تمرین بیشتر را با SVM افزایشی ساده مقایسه کند. این در این مقاله ذکر شد که SVM افزایشی مبتنی بر مجموعه رزرو شده نامیده می‌شود و توسط Yi و همکاران (2011) پیشنهاد شد. برای کاهش بیستر زمان تمرین، آنها یک روش حلقه متمرکز را پیشنهاد دادند تا در انتخاب نمونه‌ها مورد استفاده قرار گرفتند تا یک مجموعه رزرو شده از بردارهای پشتیبان را بسازد که مورد استفاده قرار گرفت تا در افزایش بعدی همراه با مجموعه داده‌های جدید تمرین ببینند. در کارشان توضیح دادند که نمونه داده‌های متکی به ناحیه حلقه دو حلقه متمرکز بیشترین احتمال را داشتند تا به بردارهای پشتیبان تبدیل شوند، و بنابراین، به عنوان مجموعه رزرو شده حفظ شدند.

علاوه بر بردارهای پشتیبان حفظ شده و ابقا شده، تلاشهایی برای کاهش بردار پشتیبان نیز وجود داشته‌اند. برای کاهش هزینه محاسبات، Kobayashi و Otsu (2009) روشی را پیشنهاد دادند تا به ترتیب بردارهای پشتیبان مازاد را حذف کند. به طور مشابه، Habib و همکاران (2009) الگوریتمهای مختلفی برای کاهش مجموعه بردار پشتیبان پیشنهاد دادند تا فرایند دسته‌بندی را برای یک الگوریتم کلی تشخیص تغییر مبتنی بر SVM تسریع بخشند. پیشنهادهای این مقاله عبارتند از : 1) بهبودی برای روش حلقه متمرکز، 2) روش نیمه پارتیشن انتخاب بردارهای پشتیبان در دسته‌بندی افزایشی SVM، که به طورکلی زمان دسته‌بندی را همراه با تشخیص بالا و میزان هشدار اشتباه کم کاهش می‌دهد.

3- بردارهای پشتیبان مناسب مبتنی بر SVMافزایشی

کلید ماشین بردار پشتیبان افزایشی پیشنهادی انتخاب بردارهای پشتیبان مناسب است. بنابراین دو روش انتخاب CSV ها ارائه می‌شوند: 1) روش حلقه متمرکز بهبودیافته؛ 2) استرتژی نیمه پارتیشن.

ازآنجایی که ثابت شده است روش دوم بهتر است، این کار تحقیقاتی الگوریتم انتخاب CSV را طراحی می‌کند که از روش نیمه پارتیشن استفاده می‌کند. نمونه داده‌های جدید مورد استفاده برای تمرین در هر افزایش بازبینی می‌شوند که آیا آنها می‌توانند با استفاده از تابع تصمیم‌گیری پیشنهاد شده توسط نظریه KKT به بردارهای پشتیبان تبدیل شوند.

3-1- شرایط KKT برای ISVM

در دسته‌بندی SVM، یک تابع تصمیم با حل برنامه‌ریزی کودراتیک (QP) به دست آمده است. برای به دست آوردن راه‌حل بهینه، شرایط KKT زیر باید برآورده شود (باید در شرایط زیر صدق کند):

$$\begin{cases} \alpha_i = 0 \Rightarrow & f(x_i) > 1 \text{ or } f(x_i) < -1 \\ 0 < \alpha_i < C \Rightarrow & f(x_i) = 1 \text{ or } f(x_i) = -1 \\ \alpha_i = C \Rightarrow & -1 < f(x_i) < 1 \end{cases} \qquad (1)$$

که α_i ضریب لاگرانژ مربوط به نمونه‌ها است، و $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots)$ راه‌حل بهینه می‌باشد اگر و تنها اگر هر x_i نمونه در این شرایط صدق کند.

اینجا f(X)=0 ابرصفحه جداکننده بهینه است، $f(x) = \pm 1$ حدود مرزهای حاشیه‌ای جدا کننده هستند. بنابراین، برای یک نمونه تمرینی x_i، اگر $\alpha_i = 0$، آن خارج از حدود قرار می‌گیرد؛ $0 < \alpha_i < C$، سپس آن روی هر کدام از حدود می‌افتد؛ اگر $\alpha_i = C$، بنابراین آن داخل حدود حاشیه‌ای جداکننده قرار می‌گیرد.

3-2- چرخش ابرصفحه SV

ایده روش نیمه پارتیشن از پدیده چرخش‌های ابرصفحه SV ناشی می‌شود. بنابراین، چرخش‌های هندسی ممکن ابرصفحه SV اینجا نشان داده می‌شوند. همانطورکه در شکل 1 دیدید، ابرصفحه می‌تواند باتوجه به موقعیت فعلی ابرصفحه هم در جهت عقربه‌های ساعت و هم در خلاف جهت عقربه‌های ساعت بچرخد. هیچ تغییری در این گردش

دلالت براین ندارد که نمونه‌های جدید در شرایط KKT صدق کنند. برای لیست کردن گردش‌های ابرصفحه SV ممکن، موارد زیر را داریم:

1) خلاف جهت گردش عقربه‌های ساعت

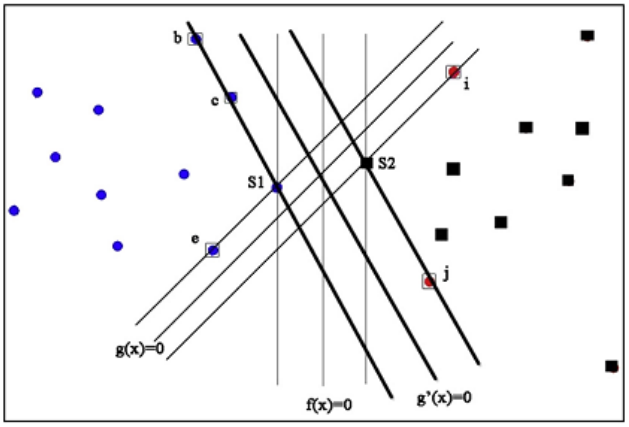
2) در جهت گردش عقربه‌های ساعت

3) بدون گردش

درشکل1، دایره‌ها و مربع‌های جامد نشان دهنده نمونه‌های اصلی می‌باشند، درحالیکه دایره‌های محصورشده توسط مربع‌های خالی نشان دهنده نمونه‌های افزایشی جدید هستند. برای ابرصفحه اصلی $f(X)=0$ می‌باشد. S_1 و S_2 بردارهای پشتیبان هستند. نمونه‌های جدید i و j برای هدف فراگیری افزایشی SVM معرفی می‌شوند. چون آنها شرایط KKT را نقض می‌کنند، در تکرارهای متوالی فرایند فراگیری SVM، آنها باعث می‌شوند گردش‌های ابرصفحه بهینه به ترتیب در جهت گردش عقربه‌های ساعت و خلاف آن باشند. سپس، بردارهای غیرپشتیبان {b,c,j} و {e,i} به بردارهای پشتیبان تغییر می‌یابند.

اگر ما فقط نمونه‌های جدید و بردارهای پشتیبان اصلی را درنظربگیریم، و بردارهای غیرپشتیبان اصلی را حذف کنیم، برخش اطلاعات ارزشمند را از دست خواهیم داد، که ممکن است باعث به دست آمدن یک رده‌بند بد شود.

اگر نمونه‌های اولیه کافی نباشند افزایش‌های فراگیری بعد منجر به ایجاد نوسان خواهند شد. بااین حال اگر تمام بردارهای غیرپشتیبان در فرایند فراگیری گنجانده شوند، سپس به جای بردارهای پشتیبان مناسب واقعی داده‌های نامطلوب بیشتری وجود خواهند داشت. درعین حال، با وارد شدن مداوم نمونه‌های جدید، اندازه مجموعه تمرینی درنهایت بسیار بزرگ می‌شود. بنابراین، یک روش برای انتخاب CSVها در اینجا توضیح داده می‌شود تا ISVM را کارامدتر کند.



شکل1- گردشهای ابرصفحه به خاطر مجموعه نمونه‌های جدید {b,c,j}&{e,i}

4- روش حلقه متمرکز بهبودیافته

تابع تصمیم SVM توسط بردارهای پشتیبان تعیین می‌شود. برای تعیین این که کدام نمونه‌ها بیشتر احتمال دارند تا بردارهای پشتیبان باشند و درنتیجه مجموعه CSV را بسازند، این مقاله روش حلقه متمرکز بهبود یافته را ارائه می-دهد، که واقعا تغییر و اصلاحی برای روش حلقه متمرکز پیشنهاد شده توسط Yi و همکاران (2011) می‌باشد. آن بیان کرد که اکثر نمونه‌های حاشیه‌ای در منطقه حلقه بین دو دایره متمرکز قرار می‌گیرند، همانطورکه در شکل 2 نشان داده شده است و به همین ترتیب روش حلقه متمرکز نمونه‌های قرار گرفته در منطقه حلقه را حفظ می‌کند. بنابراین فرایند ایجاد حلقه یعنی تعیین شعاع R_0 و R_i دایره داخلی و بیرونی به ترتیب نیاز به آزمایش های جداگانه دارد. بنابراین، روش پیشنهادی حلقه متمرکز بهبودیافته یک روش ثابت را برای تعیین R_0 و R_i پیشنهاد می‌دهد.

درابتدا، مراکز تمام کلاسها به صورت زیر محاسبه می‌شوند:

$$m_{ij} = \frac{1}{n_i} \sum_{k=1}^{n_i} x_{kj}$$

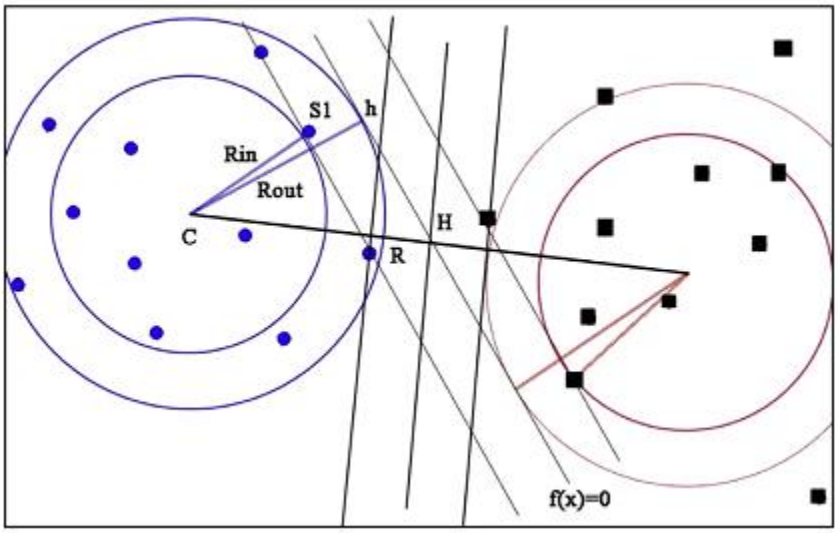
که j=1,2,.....,d؛ d بعد میانگین بردار می‌باشد. n_i تعداد نمونه‌های متعلق به دسته i در مجموعه نمونه است و x_{jk} jامین نسبت (ویژگی) نمونه k می‌باشد. i=1,2,... نشان دهنده دسته نمونه می‌باشد.

دوما، فاصله اقلیدسی R بین دو مرکز طبقه محاسبه کنید. برای مشکل دسته‌بندی باینری، می‌تواند به صورت زیر تعیین شود:

$$R = r(m_1, m_2) = \sqrt{\left(\sum_{k=1}^{d} |m_{1k} - m_{2k}|\right)}$$

که m_1 و m_2 میانگین بردارها، و m_{1k} و m_{2k} kامین بعد میانگین بردارها می‌باشند. فاصله اقلیدسی بین دو نمونه توسط رابطه زیر محاسبه می‌شود:

$$d_{ij} = \sqrt{\left(\sum_{k=1}^{m} |x_{ik} - x_{jk}|^2\right)}$$



شکل 2- روش حلقه متمرکز بهبودیافته برای انتخاب CSVها

4-1- انتخاب شعاع R_0 و R_i

انتخاب R_0 و R_i بسیار مهم و حساس است. این روش باید چنین باشد که نقاط داده قرار گرفته در داخل دایره داخلی دارای کمترین احتمال برای بودن بردار پشتیبان هستند و آنهایی که در بیرون دایره خارجی قرار دارند هیچ شانسی ندارند. بنابراین ما R_0 و R_i را به صورت مقیاسهای زیر درنظر می‌گیریم:

R_0= فاصله بین مرکز دسته C و نزدیکترین بردار پشتیبان S_1.

R_i= فاصله بین مرکز دسته C و نزدیکترین نقطه در ابرصفحه h.

اینجا R_i می‌تواند با استفاده از رابطه 4 محاسبه شود. ما می‌دانیم که نقطه میانی مرکز دو دسته، که H نام دارد، روی ابرصفحه قرار می‌گیرد. اگر R فاصله بین مرکز دو دسته باشد، بااستفاده از رابطه 3 به دست می‌آید. از قبل R معلوم می‌باشد، R_0 ممکن است بدین صورت محاسبه شود $R_0 = R/2$. اما این مقاله یک روش محاسباتی جایگزین را پیشنهاد می‌دهد به طوریکه در بخش بعدی توضیح داده شده است.

4-2- محاسبه r

فرض کنید، r فاصله اقلیدسی از نقطه $\vec{x}$ تا مرز تصمیم می‌باشد، همانطورکه در شکل 3 نشان داده شده است، مثلا نقطه $\acute{x}$. ما می‌دانیم که کوتاهترین فاصله بین یک نقطه داده و یک ابرصفحه عمود بر صفحه می‌باشد (Manning و همکاران، 2008)، و درنتیجه موازی با $\vec{w}$. یک بردار واحد در این مسیر برابر است $r\vec{w}/\vec{w}$. خط نقطه‌چین در نمودار تفسیری از بردار $r\vec{w}/\vec{w}$ می‌باشد. بگذارید این نقطه روی ابرصفحه که نزدیکترین به $\vec{x}$ است را $\acute{\vec{x}}$ بنامیم.

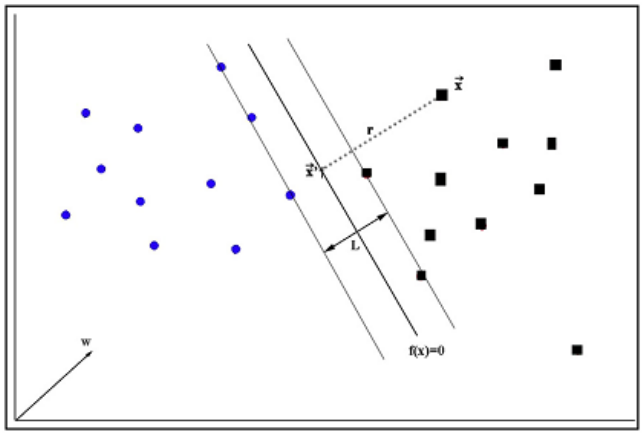
$$\vec{x}' = \vec{x} - yr\frac{\vec{w}}{|\vec{w}|}$$

که ضرب کردن در y فقط علامت را برای دو مورد $\vec{x}$ تغییر می‌دهد که در دو طرف صفحه تصمیم می‌باشد. علاوه‌براین، $\vec{x}$ در مرز تصمیم قرار دارد و در این رابطه صدق می‌کند $\vec{w}^T\acute{\vec{x}} + b = 0$. درنتیجه:

$$\vec{w}^T\left(\vec{x} - yr\frac{\vec{w}}{|\vec{w}|}\right)$$

حل براساس r نتیجه می‌دهد:

$$r = y\frac{\vec{w}^T\vec{x} + b}{|\vec{w}|}$$



شکل 3- فاصله (r) یک نقطه داده (x) از ابرصفحه

مزیت این روش محاسباتی این است که ما نیاز نداریم تا فاصله بین دو مرکز دسته، R، را محاسبه کنیم که محاسبات را فقط به دسته فعلی وابسته می‌کند.

5- استراتژی نیمه پارتیشن

در شکل2، نقاط داده حتی در خارج از منطقه حلقه وجود دارند. این نقاط داده با روشهای حلقه متمرکز به عنوان بردار پشتیبان مناسب درنظر گرفته نمی‌شوند. مگراینکه ما ثابت کنیم که آنها نقاط دورافتاده واقعی هستند (به عنوان مثال به همین دسته تعلق ندارند) یا در فرایند دسته‌بندی درنظرگرفته نمی‌شوند، انها نمی‌توانند از مجموعه CSV حذف شوند. بنابراین، این کار تحقیقاتی حذف دایره خارجی را درنظر می‌گیرد (همانطورکه در شکل 4 نشان داده شده است حلقه دیگر وجود ندارد) تا مجموعه CSV را ماکزیمم کند. باانجام این کار، تمام بردارهای غیرپشتیبان قرار گرفته در خارج دایره داخلی (دارای شعاع R_i) و همچنین نقاط دورافتاده به مجموعه CSV بزرگتر تعلق دارند درنتیجه آماده کردن آنها برای تبدیل شدن به یک CSV دارای شانس برابر می‌باشند.

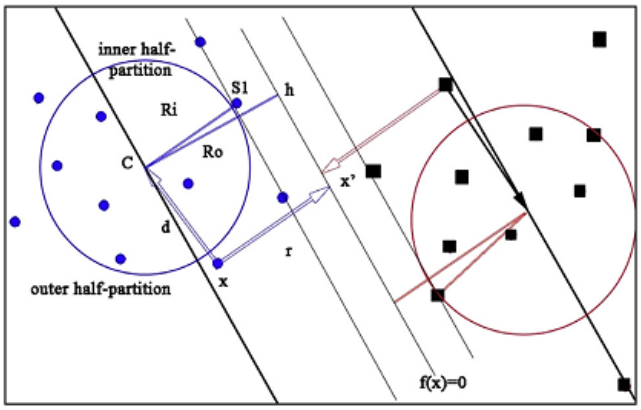
بنابراین با نگاهی به گردشهای ممکن ابرصفحه‌های SV در شکل 1، مشخص میشود که نیمی از کل فاصله داده‌های پوشش داده شده توسط هر دسته تحت تاثیر ابرصفحه چرخیده (مدور) قرار نمی‌گیرند، یعنی $g(x)=0$ و $\acute{g}(x) = 0$، که بدین معنی است که نیمی از دسته‌های خارجی حاوی نقاط داده نیستند که بتوانند به عنوان CSVها برای افزایش بعدی انتخاب شوند. این وضعیت درمورد طبقه‌بندی دو دسته‌ای بسیار درست می‌باشد. برای مساله تشخیص نفوذ شبکه در یک دسته باینری، تقسیم‌های خارجی دو دسته می‌توانند مانع انتخاب CSV شوند. بنابراین این مقاله

روش نیمه پارتیشن را همانطور که در شکل 4 نشان داده شده است پیشنهاد می‌دهد، که نقاط داده‌ای را درنظر می-گیرد که فقط در داخل نیمی از تقسیمات دو دسته قرار دارند.

اگر یک بردار غیرپشتیبان، یعنی نقطه داده x در شکل 4، شرایط زیر را ارضا می‌کند، درنتیجه به عنوان یک بردار پشتیبان مناسب درنظر گرفته خواهد شد:

$$d \geq R_i \quad \text{AND} \quad r \leq R_o$$

که d فاصله بین x و مرکز دسته C می‌باشد، و r فاصله بین x و ابرصفحه است.



شکل4- استراتژی نیمه پارتیشن و محاسبه وزن CSVها

5-1- محاسبه وزن CSV

احتمال شدن یک CSV به دو پارامتر بستگی دارد:

1) d- فاصله اقلیدسی بین بردار پشتیبان مناسب، x در شکل 3، و مرکز دسته C، و

2) r- فاصله بین بردار پشتیبان مناسب x و ابرصفحه SVM، ẋ. اینجه d وr می‌توانند به ترتیب با رابطه 4 و 7 محاسبه شوند.

نمونه داده‌هایی که دورتر از ابرصفحه قرار دارند دارای شانس کمتری برای تبدیل شدن به بردارهای پشتیبان می-باشد. به طور مشابه، نقاط داده‌ای هم که داخل دایره قرار دارند دارای شانس کمتری برای تبدیل شدن به بردارهای پشتیبان می‌باشد. بنابراین، نقاط داده‌ای که خارج دایره قرار دارند و به ابرصفحه نزدیکتر هستندشانس بالاتری دارند.

هرچه مقدار d بیشتر باشد، x شانس بیشتری برای تبدیل شدن به یک بردار پشتیبان دارد؛ و هر چه r کوچکتر باشد، احتمال دوباره بالا می‌رود. بنابراین $r \leq L$ و $d \geq R_i$ دو معیار برای فرمول بندی رابطه‌ای برای محاسبه وزن می‌باشند. اینجا، Lفاصله بین بردار پشتیبان S_1 و ابرصفحه می‌باشد، که توسط $\frac{1}{\vec{w}}$ داده می‌شود (ما می‌دانیم که $\frac{2}{\vec{w}}$ حاشیه مرز تصمیم‌گیری می‌باشد).

برای محاسبه وزن عبارت زیر پیشنهاد شده است:

$$W_C = \{d/(d + R_i) + L/(L + r)\}$$

که w_c فاصله وزنی کلی بردار پشتیبان مناسب می‌باشد. برای نزدیکترین بردار پشتیبان از مرکز دسته C، $d=R_i$ و $r=L$، بنابراین، $d/(R_i + d) = 0$ و $L/(L + r) = 0.5$ و W_c. بنابراین بردارهای غیرپشتیبانی که مقدار W_c نزدیک به 1 دارند به عنوان CSVها انتخاب می‌شوند.

5-2- تخصیص آستانه

از بخش قبلی، ما یاد گرفتیم که آستانه T ممکن است به صورت W_c-1 تعریف شود به طوریکه برای یک بردار پشتیبان معمولی T=0 می‌باشد. حالا، معادله برای آستانه T ممکن است به صورت زیر بیان شود:

$$T = W_C - 1 = \{d/(d + R_i) + L/(L + r)\} - 1$$

بستگی به توزیع نقاط داده و نوع رخ دادن نمونه داده‌های جدید، آستانه T ممکن است به طور متغیر تعیین شود.

5-3- انتخاب CSV و الگوریتم‌های CSV-ISVM

اولین الگوریتم انتخاب CSV به زودی راه‌اندازی می‌شود و اولین فراگیری SVM به پایان می‌رسد. بااستفاده از الگوریتم انتخاب CSV، که در الگوریتم 1 توصیف شد، CSVها انتخاب می‌شوند و برای تکرار بعدی دسته‌بندی SVM حفظ می‌شوند. طی فرایند انتخاب CSV، وزن‌های CSVها همراه با مجموعه‌های CSV حفظ می‌شوند. این وزن‌های CSV ممکن است با مقدار آستانه نیز تعیین شوند.

الگوریتم1 انتخاب CSV

ورودی: مجموعه نمونه X_0
خروجی: مجموعه CSV
//محاسبه بردار میانگین هر دسته//
برای هر x_i نمونه (i=0,1,....l) در X_0
اگر x_i به دسته نرمال تعلق دارد
برای k=1,.... ,d
$m_1(k)= m_1(k)+ x_i(k);$
ENDFOR
$n_1 = n_1 +1;$
دیگر // x_i به دسته حمله تعلق دارد//
برای k=1, ... ,d
$m_2(k)= m_2(k)+ x_i(k);$
ENDFOR
$n_2 = n_2 +1;$
ENDIF
ENDFOR
برای k=1, ... ,d
$m_1(k)= m_1(k)/n_1;$ $m_2(k)= m_2(k)/n_2;$
ENDFOR
محاسبه شعاع R_i بااستفاده از معادله 3؛
// نمونه‌ها را انتخاب کنید تا مجموعه CSV ساخته شود//
برای هر x_i نمونه (i=1,2, ... ,l)
فاصله d از x_i برای بردار میانگین دسته‌ای را محاسبه کنید که به x_i تعلق دارد؛
فاصله r از x_i را برای ابرصفحه در h محاسبه کنید؛
اگر $d \geq R_i$ and $r \leq R_o$
x_i را به مجموعه CSV اضافه کنید؛
ENDIF
ENDFOR

هروقت یک نمونه افزایشی جدید ظاهر می‌شود، SVM افزایشی ساده بررسی می‌شود که آیا تمام نمونه‌ها شرایط KKT را برآورده می‌کنند، اگر هیچ‌کدام وجود نداشته باشد، آن این نمونه‌ها را به بردارهای پشتیبان اضافه خواهد کرد تا یک مجموعه تمرینی جدید را ایجاد کنند. درهمین حال، نمونه‌ها از مجموعه CSV باتوجه به وزن‌شان انتخاب

می‌شوند و آنها با نمونه‌هایی از مجموعه نمونه جدید ترکیب می‌شوند که شرایط KKT را نقض می‌کنند و درنهایت آنها به بردارهای پشتیبان اصلی اضافه می‌شوند تا یک مجموعه تمرینی جدید را ایجاد کنند.

بعد از فراگیری افزایشی، مجموعه CSV و وزنهای نمونه‌ها در آن به روز می‌شوند. کل فرایند فراگیری افزایشی در الگوریتم 2 توصیف می‌شود.

الگوریتم 2: CSV-ISVM
ورودی: مجموعه نمونه X_0، مجموعه نمونه افزایشی جدید X_I
خروجی:مجموعه CSVبه روز شده و SVM رده‌بند
با تمرین دادن X_0 مجموعه بردار پشتیبان SV_0 و رده‌بند SVM_0 را بگیر؛
بااستفاده از رابطه 2 مراکز دسته $\mathbf{X_0^-}$ و $\mathbf{X_0^+}$ محاسبه کن؛
فاصله بین مراکز دسته و نمونه‌ها را با رابطه 4 محاسبه کن؛
مجموعه CSV، N_0 را باانتخاب نمونه‌ها ایجاد کن که در رابطه 8 صدق کند؛
وزنهای $\boldsymbol{\theta_i}$ و آستانه‌های T_i به ترتیب بااستفاده از رابطه 9 و 10 به دست آورید؛
مجموعه نمونه جدید X_I را بسازید؛
برای هر نمونه x_i در X_I
اگر x_i شرایط KKT، SVM_0 را نقض کند؛
x_i را به X_{Is} اضافه کن؛
ELSE
x_i را به X_{Id} اضافه کن؛
ENDIF
ENDFOR
اگر X_{Is} صفر نبود //نمونه‌ها را براساس آستانه‌شان انتخاب کن//
برای هر x_i نمونه در X_{Is}
اگر $T>$ مقدار آستانه داده شده باشد
x_i نمونه را به مجموعه NSV_0 اضافه کن؛
ENDIF
ENDFOR
مجموعه $N_I= SV_0 \cup NSV_0 \cup X_{Is}$؛
نمونه‌ها را از N_I انتخاب کنید تا N_I رابسازید؛
رده‌بند SVM_1 و SV_1 را با تمرین روی X_O به دست آورید؛
SVM_1 را به عنوان خروجی تنظیم کنید؛

مجموعه CSV و همچنین وزنهای $\boldsymbol{\theta_i}$ و آستانه‌های T_i به روز کنید؛
ELSE
مجموعه SVM_0 به عنوان خروجی تنظیم کنید؛
END

6- آزمایش

6-1 داده‌های تجربی

اغلب دو معیار مجموعه داده‌ها برای ارزیابی عملکرد هر آزمایش مربوط به تشخیص نفوذ استفاده می‌شوند. آنها عبارتند از: مجموعه داده KDD Cup 1999 (UCI, 1999) و مجموعه داده Kyoto 2006 + (Kyoto +, 2009). در این کار تحقیقاتی، مجموعه داده Kyoto 2006 + به عنوان مجموعه‌های داده برای آزمایشها استفاده می‌شود. بعد از آزمایش داده‌ها – خصوصا نسبت حمله- و آزمایشات اولیه، مجموعه‌های داده روزهای 1، 2 و 3 نوامبر 2007 برای آزمایش‌های نهایی انتخاب می‌شوند.

6-2 پیش پردازش داده‌ها

تکنیک‌های پیش پردازش داده‌ها برای اجرای راحت و ساده آزمایش‌ها اعمال می‌شوند. همانطورکه توسط Chitrakar and Huang (a,b2012) پیشنهاد شده، نمونه داده‌ها به گونه‌ای استخراج می‌شوند که هر مجموعه شامل 1 درصد از حملات می‌شود که حملات نامعلوم را نیز دربرمی‌گیرد. ویژگی‌های مجموعه داده‌ها به انواع مناسب تبدیل شده و باتوجه به نیاز هسته svm نرمال می‌شوند. نمونه‌هایی با حملات شناخته و ناشناخته به صورت یک دسته مهاجم واحد رفتار شده و به عنوان 1- برچسب‌گذاری می‌شوند. کل داده‌های انتخاب شده می‌توانند به صورت {1،-1} طبقه‌بندی شوند که 1 نشان دهنده دسته نرمال است. این تقریبا هر نوع دسته‌بندی SVM را قادر می-سازد تا از داده‌ها و فرایند دسته‌بندی راحتر و ساده‌تر استفاده کنند.

6-3- جزئیات تجربی

جزئیات تجربی به دست آمده از داده‌های + Kyoto 2006 به طور تصادفی به دو زیرمجموعه غیرهمپوش- مجموعه تمرینی و تست کردن- تقسیم می‌شوند. سپس 10 مجموعه داده جداگانه از زیرمجموعه تمرینی به عنوان مجموعه‌های تمرینی افزایشی استخراج می‌شوند که هر مجموعه شامل 1000 نمونه که دارای داده‌های نرمال و غیرنرمال می‌شوند. به طور مشابه، 10 مجموعه داده هم از مجموعه داده‌های آزمون استخراج می‌شوند.

بااستفاده از هر نمونه داده، آزمایشات برای ISVM ساده، KKT-ISVM، RS-ISVM و CSV-ISVM یک به یک انجام می‌شوند. در تمام چهار روش، هسته RBF استفاده می‌شود چون SVM با RBF عملکرد بهتری ایجاد می‌کند (Bhavsar et al., 2013).

تعداد مقدارهای مثبت واقعی، منفی واقعی، مثبت کاذب و منفی کاذب هر چهر روش ثبت می‌شوند و برای ارزیابی عملکرد مورد استفاده قرار می‌گیرند.

4-6- ارزیابی و آنالیز عملکرد

ارزیابی عملکرد آزمایش برحسب دقت (A)، سرعت تشخیص (DR) و میزان هشدار اشتباه (FAR) بااستفاده از معادلات زیر انجام می‌شود:

$$A = (TP + TN)/(TP + TN + FP + FN)$$

$$DR = (TP)/(TP + FP)$$

$$FAR = (FP)/(FP + TN)$$

که TP، TN، FP و FN به ترتیب نشان دهنده مثبت واقعی (حمله شناسایی شده به عنوان مهاجم)، منفی واقعی (نرمال شناسایی شده به عنوان نرمال)، مثبت کاذب (نرمال شناسایی شده به عنوان حمله) و منفی کاذب (حمله شناسایی شده به عنوان نرمال) هستند.

ارزیابی CSN-ISVM برای سرعت تشخیص و میزان هشدار اشتباه به وسیله مقایسه‌اش با ISVM ساده، KKT-ISVM و RS-ISVM انجام می‌شود. ارزیابی مدت آزمایشی و آزمون برای هر چهار روش انجام شده است.

جدول 1 نشان می‌دهد که مقادیر DR و FAR در ISVM ساده در یک توسعه افزایش می‌یابند درحالیکه در توسعه‌ای دیگر کاهش می‌یابند یعنی DR در تکرار دوم تا 86.826% افزایش یافته اما در تکرار سوم تا 81.435% کاهش یافته است؛ در مورد FAR هم به همین صورت است. این دلالت براین دارد که آنها به شدت وابسته به مجموعه داده‌های جدید در هر تکرار هستند. KKT-ISVM در مقایسه با افزایش‌های اولیه مقادیر DR بالاتری را (مقادیر FAR پایین‌تر) در دو افزایش آخر به وجود می‌آورد، اما مقادیر افزایشی و کاهشی را طی افزایش‌های میانی تولید می‌کند یعنی در تکرارهای 5،6 و 7 مقادیر DR به ترتیب برابر با 81.941%، 80.69% و 86.518٪ هستند و FARها به ترتیب 4.518٪، 5.153٪ و 4.327٪ می‌باشند. در شکل 5 و 6 دیده می‌شود که KKT-ISVM در مقایسه با ISVM ساده دارای روند رو به رشد بهتری از DR و روند نزولی بهتری از FAR می‌باشد، به طور کلی هنوز تناقضاتی در برخی موارد وجود دارد.

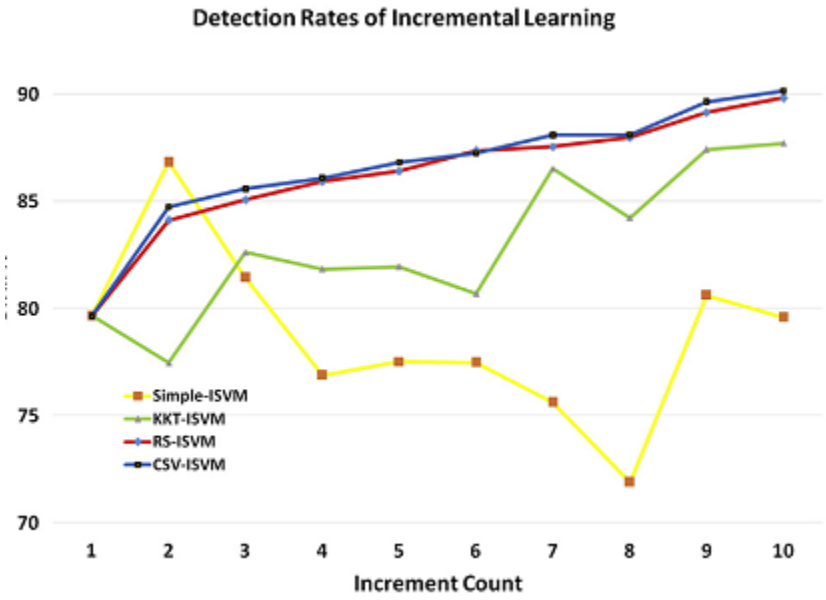
دوباره در جدول 1، دیده می‌شود که RS-ISVM و CSV-ISVM به طور مداوم DRهای افزایشی و FARهای کاهشی تولید می‌کنند و همچنین نسبت به ISVMساده و KKT-ISVM دارای سرعتهای بهتری می‌باشند.

Table 1 – Comparison of DR and FAR.

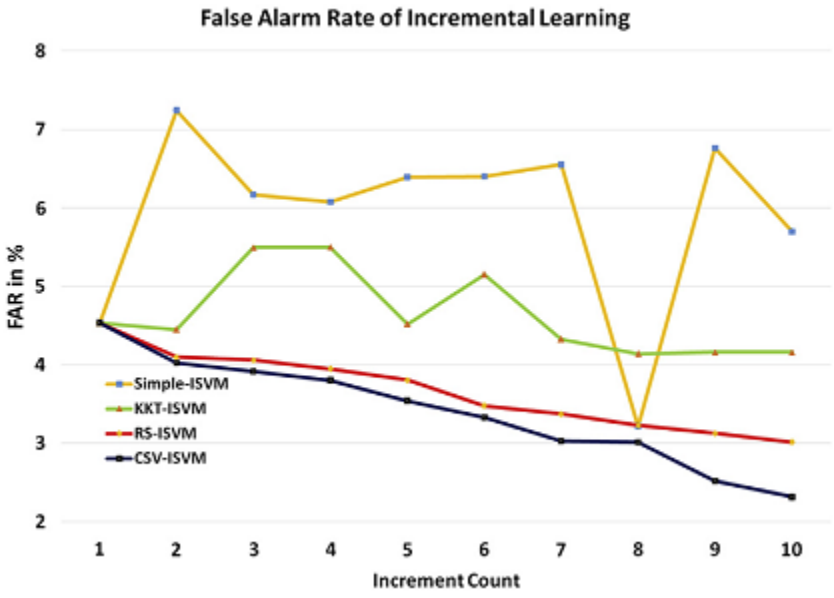
Iteration count	Simple-ISVM		KKT-ISVM		RS-ISVM		CSV-ISVM	
	DR (%)	FAR (%)	DR (%)	FAR (%)	DR (%)	FAR (%)	DR (%)	FAR (%)
1	79.646	4.535	79.646	4.535	79.646	4.535	79.646	4.535
2	86.826	7.245	77.451	4.449	84.113	4.099	84.724	4.025
3	81.435	6.169	82.608	5.494	85.069	4.059	85.576	3.915
4	76.871	6.076	81.829	5.501	85.917	3.948	86.082	3.802
5	77.513	6.393	81.941	4.518	86.407	3.805	86.814	3.54
6	77.474	6.403	80.69	5.153	87.352	3.476	87.238	3.33
7	75.618	6.552	86.518	4.327	87.54	3.372	88.082	3.027
8	71.882	3.216	84.221	4.141	87.968	3.225	88.095	3.012
9	80.606	6.762	87.419	4.162	89.144	3.124	89.643	2.52
10	79.576	5.699	87.706	4.16	89.817	3.015	90.147	2.314

در اولین تکرار نتیجه مشابهی برای هر چهار روش به دست آمد زیرا SVها فقط از تکرار دوم به بعد همراه با مجموعه نمونه‌های جدید تمرین می‌بینند؛ و نتیجه نهایی از تکرار ده‌ام به دست آمده است. در اینجا، RS-ISVM به ترتیب 3.015% , 89.817% از DR و FAR نهایی را داشته، در حالیکه CSV-ISVM به ترتیب 90.147% و 2.314٪ از DR و FAR نهایی را دارد.

همانطور که در شکل 5 و 6 دیده می‌شود، RS-ISVM و CSV-ISVM خطوط DR رو به رشد و خطوط FAR نزولی را به صورت توسعه‌های مداوم نشان می‌دهند. علاوه‌براین، روشهای RS-ISVM و CSV-ISVM، DR و FAR در CSV-ISVM به نظر می‌رسند که دارای مقادیر بهتر و روند کاهشی کاملتری را نسبت به RS-ISVM می‌باشند. این به وضوح نشان می‌دهد که ایده حفظ بردارهای پشتیبان به عنوان دانش پیشین نتایج بهتری را برای فراگیری افزایشی ارائه می‌دهد.



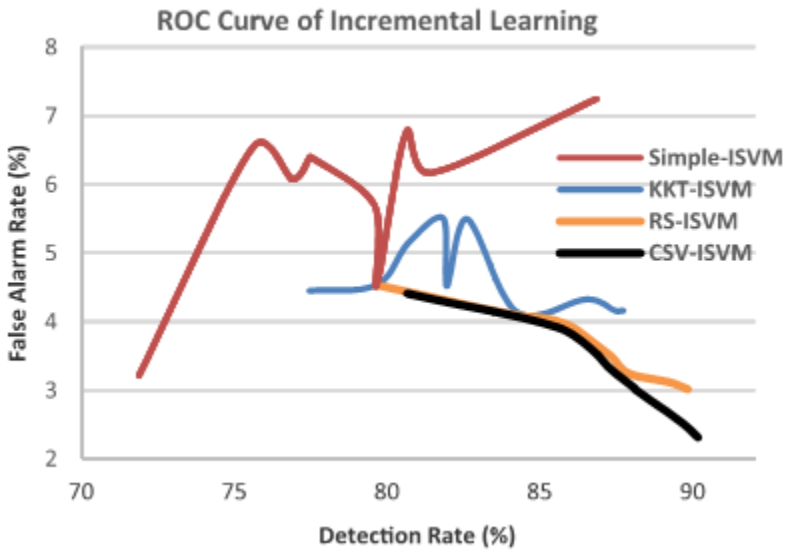
شکل5- مقایسه DR



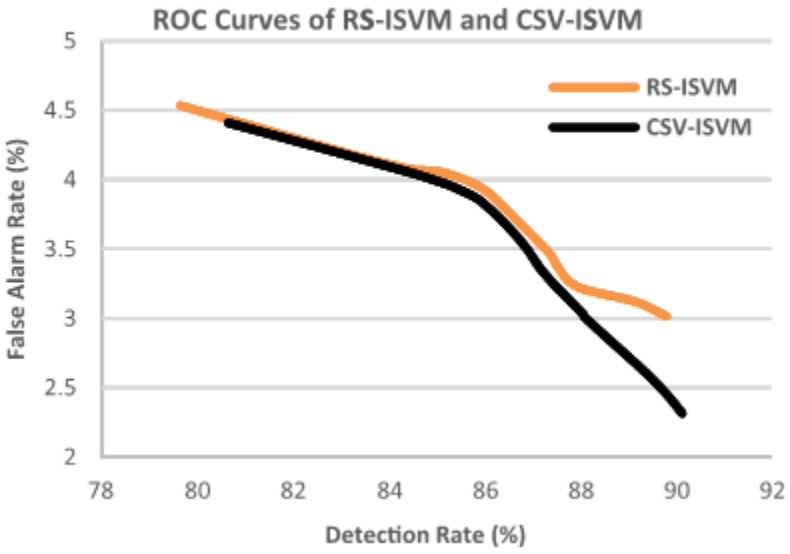
شکل 6- مقایسه FAR

منحنی‌های ROC، DR در برابر FAR در شکل 7 نشان داده شده‌اند، که ما می‌توانیم منحنی‌های ROC هموارتر و خمیده‌تری را در مقایسه با دیگر روشهای ISVM ببینیم که توسط CSV-ISVMتولید شده‌اند. شکل منحنی ISVM ROC ساده صاف نیست چون این روش نمونه داده‌های اولیه را حذف می‌کند (بردارهای پشتیبان بالقوه) و درنتیجه منجر به بی‌ثباتی می‌شود (Yi و همکاران،2011). KKT-ISVM، اگرچه منحنی ROC را در مقایسه با روش ISVM ساده بهبود داده است، اما هنوز نسبت DR/FAR را همیشه حفظ نمی‌کند. این شکل همچنین نشان می‌دهد که CSV-ISVM بالاترین DR ممکن را ارائه می‌دهد درحالیکه پایین‌ترین FARرا هم حفظ می-کند.

وقتی به طور خاص با RS-ISVM مقایسه می‌شود، CSV-ISVMبه نظر می‌رسد مقادیر کوچکتر FAR تولید شده حفظ شده‌اند وقتی DR در طول افزایش‌های بعدی بیش از 85 درصد تجاوز کند همانطور که توسط منحنی زیرین دیگر CSV-ISVM نشان داده شده در شکل 8 نشان داده شده است. این حاکی از این است که حتی با مجموعه نمونه‌های جدیدی که در هر افزایش جدید وجود دارند، هنوز DR CSV-ISVM افزایشی و FAR کاهشی را تولید می‌کند چون استراتژی نیمه پارتیشن مانع این می‌شود تا روش فراگیری به طور بالقوه نقاط داده‌های ناشناخته را حفظ کند.



شکل 7- مقایسه منحنی‌های ROC



شکل 8- مقایسه روش RS-ISVM و CSV-ISVM

جدول 2 مدت زمان را مقایسه می‌کند که توسط هر چهار روش در نمونه داده‌های تمرینی و آزمون در هر توسعه گرفته شده است. مدت زمان ثبت شده برای نمونه داده‌های تمرینی در افزایش اولیه تقریبا برای چهار روش برابر است یعنی 1.8xx ثانیه، درحالیکه برای نمونه داده‌های آزمون در روش CSV-ISVM از 9.4 ثانیه به 7.76 ثانیه کاهش یافته است و در روش ISVM ساده 1.64 ثانیه ذخیره شده است. به طور مشابه، در آخرین افزایش، 39.766 ثانیه از زمان تمرین و 39.881 ثانیه از زمان آزمون توسط CSV-ISVM ذخیره شده‌اند. برخلاف دیگر روشها، KKT-

ISVM، اگرچه DRها و FARهای بهتری را در مقایسه با ISVM ساده دارد، مدت زمان فراگیری طولانی‌تری را ثبت می‌کند، همچنان که در شکل 9 و 10 نشان داده شده است، که واضح است ان نیاز به قضاوت متقابل و تمرین بیشتری دارد. مشخص است که این زمان ثبت شده توسط CSV-ISVM برای تمرینی و آزمون در مقایسه با KKT-ISVM و ISVM ساده به نصف کاهش می‌یابند. علاوه‌براین به خاطر استراتژی نیمه پارتیشن، زمان ثبت شده توسط هر دو در مقایسه با روش حلقه متمرکز RS-ISVM به طور قابل توجهی کاهش می‌یابند.

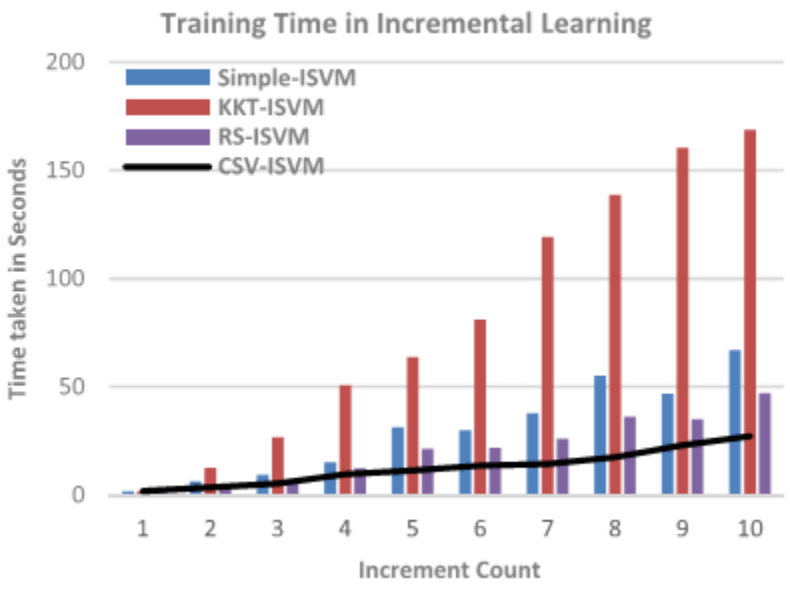
Table 2 – Comparison of training and testing time.

Iteration count	Simple-ISVM		KKT-ISVM		RS-ISVM		CSV-ISVM	
	Train (s)	Test (s)	Train (s)	Test (s)	Train (s)	Test (s)	Train (s)	Test (s)
1	1.843	9.4	1.842	9.48	1.833	8.58	1.823	7.76
2	6.407	24.688	12.578	27.005	4.935	17.5255	3.463	10.363
3	9.375	34.27	26.667	39.973	7.3175	24.8565	5.26	15.443
4	15.155	39.87	50.733	49.427	12.4085	29.701	9.662	19.532
5	31.433	44.427	63.833	57.162	21.3675	33.581	11.302	22.735
6	30.078	58.412	81.15	64.245	21.8355	42.1495	13.593	25.887
7	37.787	70.157	119.4	74.272	26.0675	49.1935	14.348	28.23
8	55.157	79.948	138.667	83.36	36.316	55.7815	17.475	31.615
9	46.9	73.022	160.45	88.412	34.96	54.2845	23.02	35.547
10	67.023	79.978	168.75	99.912	47.14	60.0375	27.257	40.097

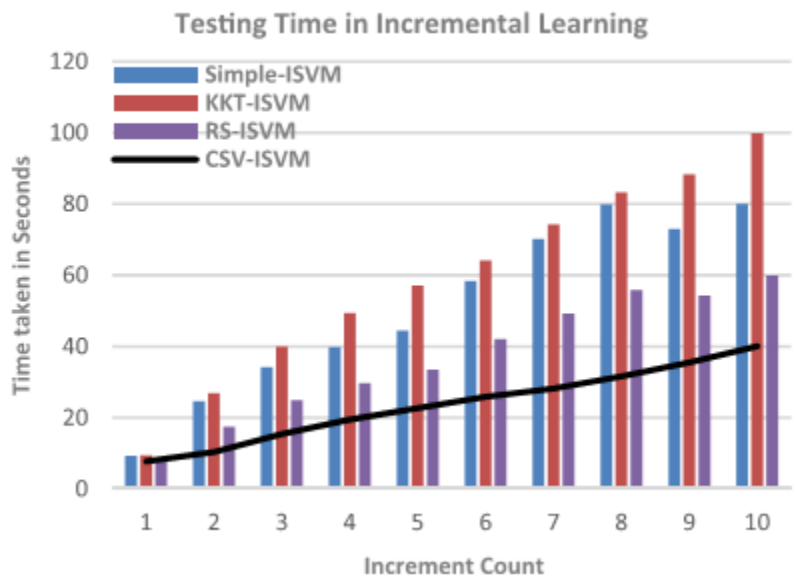
مقادیر زمان گرفته شده توسط هر چهار روش مشخص است که افزایش می‌یابند چون افزایشها ادامه دارند، اما RS-ISVM و CSV-ISVM در مقایسه با دو روش دیگر زمان کمتری را برای فراگیری می‌گیرند.

به عبارت دیگر، همانطور که در جدول 3 نشان داده شده است، این اختلاف زمان بین دو افزایش بعدی برای مجموعه داده‌های تمرینی و آزمون در یک روش به طور قابل ملاحظه‌ای با روش دیگر اختلاف دارد. میانگین تفاوت زمان یادگیری و آزمون در مورد ISVM ساده به ترتیب 7.242٪ و 7.842٪ محاسبه می‌شوند، درحالیکه آنها برای CSV-ISVM به ترتیب 2.826٪ و 3.593٪ می‌باشند. این اختلاف زمان بین دو افزایش ممکن است به عنوان زمان ثبت شده برای کنترل مجموعه داده‌های جدید اضافه شده در افزایش بعدی درنظر گرفته شود. این بدین معنی است که CSV-ISVM نسبت به چهار ISVM دیگر به طور موثرتری مجموعه داده‌های جدید را کنترل می‌کند.

تمام نتایج تجربی ذکرشده در بالا و آنالیزهای عمیق به طور واضح نشان می‌دهند که CSV-ISVM پیشنهادی از لحاظ DR و FAR و همچنین زمان تمرین و آزمایش از ISVM ساده، KKT-ISVM و RS-ISVM پیشی گرفته است.



شکل 9- مقایسه زمان تمرین



شکل 10- مقایسه زمان آزمون

7- نتیجه‌گیری و پیشنهادها

یک رویکرد بهبودیافته برای دسته‌بندی ماشین بردار پشتیبان افزایشی، که CSV-ISVM نامیده می‌شود، پیشنهاد شده است و برای یادگیری SVM افزایشی اجرا شده که براساس تشخیص نفوذ شبکه می‌باشد. این رویکرد در دو

روش کمک کرده است. اولا، آن اصلاحاتی را برای روش حلقه متمرکز پیشنهاد شده قبلی ارائه داده است. دوما، آن یک روش سریع‌تر و کاراتر برای انتخاب بردار پشتیبان پیشنهاد داده شده که استراتژی نیمه پارتیشن همراه با الگوریتم پیشنهاد داده است. آنالیزها و کارهای تجربی نشان داده‌اند که این روش پیشنهادی سرعت تشخیص و سرعت هشدار اشتباه بهتری و همچنین میزان زمان فراگیری قابل قبول‌تری را دارد، و می‌تواند برای تشخیص نفوذ شبکه در زمان واقعی استفاده شود.

در واقع استراتژی نیمه پارتیشن تقریبا نیمی از نقاط داده‌ای را حذف می‌کند که نسبت به مرکز دسته از صفحه دورتر می‌باشند. و چنین استراتژی نمی‌تواند در روشهای دیگری غیر از دسته‌بندی باینری اجرا شود. اصلاح کردن این استراتژی برای موثرتر کردن آن برای دسته‌بندی چند طبقه‌ای ممکن است به عنوان پیشنهاد (کار آتی) این تحقیق درنظر گرفته شود.

Table 3 – Comparison of time difference between two subsequent increments.

Iteration count	Simple-ISVM		KKT-ISVM		RS-ISVM		CSV-ISVM	
	Train (s)	Test (s)	Train (s)	Test (s)	Train (s)	Test (s)	Train (s)	Test (s)
1	*{This is the very first iteration, no incremental learning have been applied so far.}*							
2	4.564	15.288	10.736	17.525	3.102	8.9455	1.64	2.603
3	2.968	9.582	14.089	12.968	2.3825	7.331	1.797	5.08
4	5.78	5.6	24.066	9.454	5.091	4.8445	4.402	4.089
5	16.278	4.557	13.1	7.735	8.959	3.88	1.64	3.203
6	−1.355	13.985	17.317	7.083	0.468	8.5685	2.291	3.152
7	7.709	11.745	38.25	10.027	4.232	7.044	0.755	2.343
8	17.37	9.791	19.267	9.088	10.2485	6.588	3.127	3.385
9	−8.257	−6.926	21.783	5.052	−1.356	−1.497	5.545	3.932
10	20.123	6.956	8.3	11.5	12.18	5.753	4.237	4.55
Average	7.242222	7.842	18.54533	10.048	5.034111	5.7175	2.826	3.593

جدول3- مقایسه اختلاف زمان بین افزایش‌های بعدی

REFERENCES

Bhavsar Himani, Panchal Mahesh H, Patel Mittal. A comprehensive study on RBF kernel in SVM for multiclsss using OAA. Int J Comput Sci Manag Res 2013;2(5).

Chitrakar Roshan, Huang Chuanhe. Anomaly based intrusion detection using hybrid learning approach of combining k-medoids clustering and Naïve Bayes classification. In: Proceedings of the 8th international conference on wireless communication, networking and mobile computing; 2012.

Chitrakar Roshan, Huang Chuanhe. Anomaly detection using support vector machine classification with k-medoids clustering. In: Proceedings of the third Asian Himalayan international conference on internet (AH-ICI); 2012.

Du Hongle, Teng Shaohua, Fu Xiufen, Zhang Wei, Pu Yuanfang. A cooperative intrusion detection system based on improved parallel SVM. In: Proceedings of Joint conferences on pervasive computing (JCPC); 2009.

Du Hongle, Teng Shaohua, Yang Mei, Zhu Qingfang. Intrusion detection system based on improved SVM incremental learning. In: Proceedings of international conference on artificial intelligence and computational intelligence; 2009.

Habib Tarek, Inglada Jordi, Mercier Grégoire, Chanussot Jocelyn. Support vector reduction in SVM algorithm for abrupt change detection in remote sensing. Proc IEEE Geosci Remote Sens Lett 2009;6(3).

Joachims T. Making large-scale support vector machine learning practical. In: Smola AJ, Scholkopf B, Burges C, editors. Advances in kernel methods: support vector machines. Cambridge: MIT Press; 1998.

Karasuyama M, Takeuchi Ichiro. Multiple incremental decremental learning of support vector machines. IEEE Trans Neural Netw 2010;21(7).

Kobayashi Takumi, Otsu Nobuyuki. Efficient reduction of support vectors in kernel-based methods. In: Proceedings of IEEE international conference on image processing. ICIP; 2009. pp. 2077–80.

Kyoto+ Data. Traffic data from Kyoto University's Honeypots. Retrieved, http://www.takakura.com/Kyoto_data; 2009. on 16.10.12.

Le Giang, Nguyen Hoang. Machine learning with informative samples for large and imbalanced data sets [PhD thesis]. School of Electrical, Computer and Telecommunication Engineering, School of Wollongong; 2011.

Liu Yangguang, He Qinming, Chen Qi. Incremental batch learning with support vector machines. In: Proceedings of the 5th world congress on intelligent control and automation; 2004. pp. 1857–61.

Makili L, Vega J, Dormido-Canto S. Incremental support vector machines for fast reliable image recognition. Fusion Engineering and Design 2013;88:1170–3.

Manning Christopher D, Raghavan Prabhakar, Schütze Hinrich. Introduction to information retrieval. Cambridge University Press; 2008.

Mohammad Muamer N, Sulaiman Norrozila, Khalaf Emad T. A novel local network intrusion detection system based on support vector machine. J Comput Sci 2011;7(10):1560–4.

Platt J. Fast training of support vector machines using sequential minimal optimization. In: Smola AJ, et al., editors. Advances in kernel methods: support vector machines. Cambridge: MIT Press; 1998.

Sun Na, Guo Yanfeng. A modified incremental learning approach for data stream classification. In: Sixth international conference on internet computing for science and engineering; 2012.

Tao Liang. Fast incremental SVM learning algorithm based on active set iteration. Chin J Syst Simul 2006;18(11):3305–8.

UCI KDD. The third international knowledge discovery and data mining tools competition dataset KDD Cup 1999 data. Retrieved, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html; 1999. on October 16, 2012.

Wang Wen-Jian. A redundant incremental learning algorithm for SVM. In: Proceedings of the seventh international conference on machine learning and cybernetics; 2008. pp. 734–8.

Wang XD, Zheng CY, Wu CM, Zhang HD. New algorithm for SVM-based incremental learning. Comput Appl 2006;26(10):2440–3.

Wen-Hua Z, Jian M. An incremental learning algorithm for support vector machine and its application. Comput Integr Manuf Syst Spec Mag 2001;9:144–8.

Yao Yuan, Feng Lin, Jin Bo, Chen Feng. An incremental learning approach with support vector machine for network data stream classification problem. Proc Inf Technol J 2012;11:200–8.

Yi Yang, Wu Jiansheng, Xu Wei. Incremental SVM based on reserved set for network intrusion detection. Expert Syst Appl 2011:7698–707.

Zhang Ying, Wang Xizhao, Zhai Junhai. A fast support vector machine classification algorithm based on Karush–Kuhn–Tucker conditions. In: Rough sets, fuzzy sets, data mining and granular computing. Lecture notes in computer science, vol. 5908; 2009. pp. 382–9.