****

**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 می­شوند( 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 می­گیریم، 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 می­آورد. 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.

1. **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(N3) به O(N2)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 (2009a,b) 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.

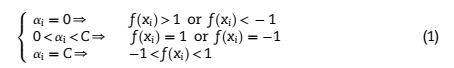
1. **بردارهای پشتیبان مناسب مبتنی بر SVM افزایشی**

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):



که ضریب لاگرانژ مربوط به نمونه­ها است، و راه­حل بهینه می­باشد اگر و تنها اگر هر xi نمونه در این شرایط صدق کند.

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

3-2- چرخش ابرصفحه SV

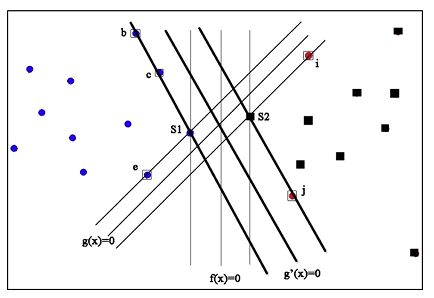
ایده روش نیمه پارتیشن از پدیده چرخش­های ابرصفحه SV ناشی می­شود. بنابراین، چرخش­های هندسی ممکن ابرصفحه SV اینجا نشان داده می­شوند. همانطورکه در شکل 1 دیدید، ابرصفحه می­تواند باتوجه به موقعیت فعلی ابرصفحه هم در جهت عقربه­های ساعت و هم در خلاف جهت عقربه­های ساعت بچرخد. هیچ تغییری در این گردش دلالت براین 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. S1 و S2 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.

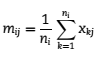


شکل1- گردشهای ابرصفحه به خاطر مجموعه نمونه­های جدید {b,c,j}&{e,i}

1. **روش حلقه متمرکز بهبودیافته**

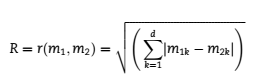
تابع تصمیم SVM توسط بردارهای پشتیبان تعیین می­شود. برای تعیین این که کدام نمونه­ها بیشتر احتمال دارند تا بردارهای پشتیبان باشند و درنتیجه مجموعه CSV را بسازند، این مقاله روش حلقه متمرکز بهبود یافته را ارائه می­دهد، که واقعا تغییر و اصلاحی برای روش حلقه متمرکز پیشنهاد شده توسط Yi و همکاران (2011) می­باشد. 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 R0 و Ri دایره داخلی و بیرونی به ترتیب نیاز به آزمایش های جداگانه دارد. بنابراین، روش پیشنهادی حلقه متمرکز بهبودیافته یک روش ثابت را برای تعیین R0 و Ri پیشنهاد می­دهد.

درابتدا، مراکز تمام کلاسها به صورت زیر محاسبه می­شوند:

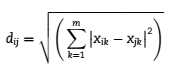


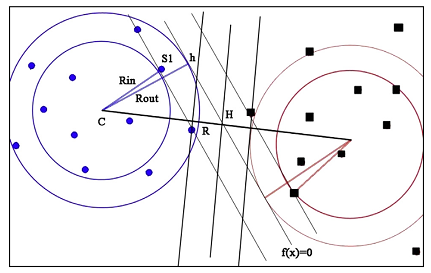
که j=1,2,……,d؛ d بعد میانگین بردار می­باشد. ni  تعداد نمونه­های متعلق به دسته i در مجموعه نمونه است و xjk jامین نسبت (ویژگی) نمونه k می­باشد. i=1,2,… نشان دهنده دسته نمونه می­باشد.

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



که m1 و m2 میانگین بردارها، و m1k و m2k kامین بعد میانگین بردارها می­باشند. فاصله اقلیدسی بین دو نمونه توسط رابطه زیر محاسبه می­شود:





شکل 2- روش حلقه متمرکز بهبودیافته برای انتخاب CSVها

**1-4- انتخاب شعاع R0 و Ri**

انتخاب R0 و Ri 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 R0 و Ri را به صورت مقیاسهای زیر درنظر می­گیریم:

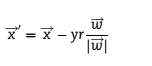
R0= فاصله بین مرکز دسته C و نزدیکترین بردار پشتیبان S1.

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

اینجا Ri 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، R0 ممکن است بدین صورت محاسبه شود . اما این مقاله یک روش محاسباتی جایگزین را پیشنهاد می­دهد به طوریکه در بخش بعدی توضیح داده شده است.

**2-4- محاسبه r**

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

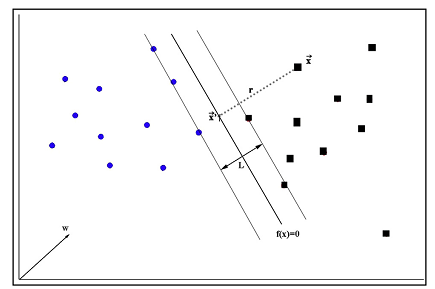


که ضرب کردن در y فقط علامت را برای دو مورد تغییر می­دهد که در دو طرف صفحه تصمیم می­باشد. علاوه­براین، در مرز تصمیم قرار دارد و در این رابطه صدق می­کند . درنتیجه:



حل براساس r نتیجه می­دهد:





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.

1. **استراتژی نیمه پارتیشن**

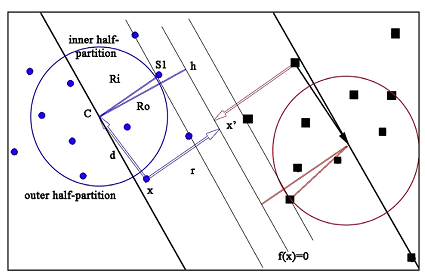
در شکل2، نقاط داده حتی در خارج از منطقه حلقه وجود دارند. این نقاط داده با روشهای حلقه متمرکز به عنوان بردار پشتیبان مناسب درنظر گرفته نمی­شوند. مگراینکه ما ثابت کنیم که آنها نقاط دورافتاده واقعی هستند (به عنوان مثال به همین دسته تعلق ندارند) یا در فرایند دسته­بندی درنظرگرفته نمی­شوند، انها نمی­توانند 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 Ri) 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 ، 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 می­گیرد 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:



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

**1-5- 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، . 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 می­باشد. 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 دو معیار برای فرمول بندی رابطه­ای برای محاسبه وزن می­باشند. اینجا، Lفاصله بین بردار پشتیبان S1 و ابرصفحه می­باشد، که توسط داده می­شود (ما می­دانیم که حاشیه مرز تصمیم­گیری می­باشد).

برای محاسبه وزن عبارت زیر پیشنهاد شده است:



که wc فاصله وزنی کلی بردار پشتیبان مناسب می­باشد. برای نزدیکترین بردار پشتیبان از مرکز دسته C، d=Ri و r=L، بنابراین، و و Wc . بنابراین بردارهای غیرپشتیبانی که مقدار Wc نزدیک به 1 دارند به عنوان CSV­ها انتخاب می­شوند.

**2-5- تخصیص آستانه**

از بخش قبلی، ما یاد گرفتیم که آستانه T ممکن است به صورت Wc-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:



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

**3-5- 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 X0 |
| خروجی: مجموعه CSV |
| //محاسبه بردار میانگین هر دسته// |
| برای هر xi نمونه (i=0,1,….l) در X0 |
| اگر xiبه دسته نرمال تعلق دارد |
| برای k=1,…. ,d |
|  |
| ENDFOR |
|  |
| دیگر // xi به دسته حمله تعلق دارد// |
| برای k=1, …. ,d |
|  |
| ENDFOR |
|  |
| ENDIF |
| ENDFOR |
| برای k=1, …. ,d |
|  |
| ENDFOR |
| محاسبه شعاع Ri بااستفاده از معادله 3؛ |
| // نمونه­ها را انتخاب کنید تا مجموعه CSV ساخته شود// |
| برای هر xi نمونه (i=1,2, … ,l) |
| فاصله d از xi برای بردار میانگین دسته­ای را محاسبه کنید که به xi تعلق دارد؛ |
| فاصله r از xi  را برای ابرصفحه در h محاسبه کنید؛ |
| اگر |
| xi 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 X0، مجموعه نمونه افزایشی جدید Xl |
| خروجی:مجموعه CSVبه روز شده و SVM رده­بند |
| با تمرین دادن X0 مجموعه بردار پشتیبان SV0 و رده­بند SVM0  را بگیر؛ |
| بااستفاده از رابطه 2 مراکز دسته محاسبه کن؛ |
| فاصله بین مراکز دسته و نمونه­ها را با رابطه 4 محاسبه کن؛ |
| مجموعه CSV، N0 را باانتخاب نمونه­ها ایجاد کن که در رابطه 8 صدق کند؛ |
| وزنهای و آستانه­های Ti به ترتیب بااستفاده از رابطه 9 و 10 به دست آورید؛ |
| مجموعه نمونه جدید Xl را بسازید؛ |
| برای هر نمونه xi در Xl |
| اگر xi شرایط KKT، SVM0 را نقض کند؛ |
| xi را به Xls اضافه کن؛ |
| ELSE |
| xi را به Xld اضافه کن؛ |
| ENDIF |
| ENDFOR |
| اگر Xls صفر نبود //نمونه­ها را براساس آستانه­شان انتخاب کن// |
| برای هر xi نمونه در Xls |
| اگر T> مقدار آستانه داده شده باشد |
| xi نمونه را به مجموعه NSV0 اضافه کن؛ |
| ENDIF |
| ENDFOR |
| مجموعه Nl= SV0 ꓴ NSV0 ꓴ Xls؛ |
| نمونه­ها را از NI انتخاب کنید تا Nl رابسازید؛ |
| رده­بند SVM1 و SV1 را با تمرین روی XO به دست آورید؛ |
| SVM1 را به عنوان خروجی تنظیم کنید؛ |
| مجموعه CSV و همچنین وزنهای و آستانه­های Ti به روز کنید؛ |
| ELSE |
| مجموعه SVM0 به عنوان خروجی تنظیم کنید؛ |
| END |

1. **آزمایش**

**1-6 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.

**2-6 پیش پردازش داده­ها**

تکنیک­های پیش پردازش داده­ها برای اجرای راحت و ساده آزمایش­ها اعمال می­شوند. همانطورکه توسط Chitrakar and Huang (2012a,b) 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 می­سازد تا از داده­ها و فرایند دسته­بندی راحتر و ساده­تر استفاده کنند.

**3-6- جزئیات تجربی**

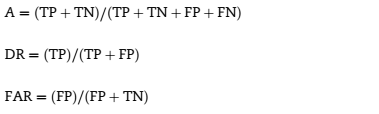
جزئیات تجربی به دست آمده از داده­های Kyoto 2006 + به طور تصادفی به دو زیرمجموعه غیرهمپوش- مجموعه تمرینی و تست کردن- تقسیم می­شوند. سپس 10 مجموعه داده جداگانه از زیرمجموعه تمرینی به عنوان مجموعه­های تمرینی افزایشی استخراج می­شوند که هر مجموعه شامل 1000 نمونه که دارای داده­های نرمال و غیرنرمال می­شوند.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

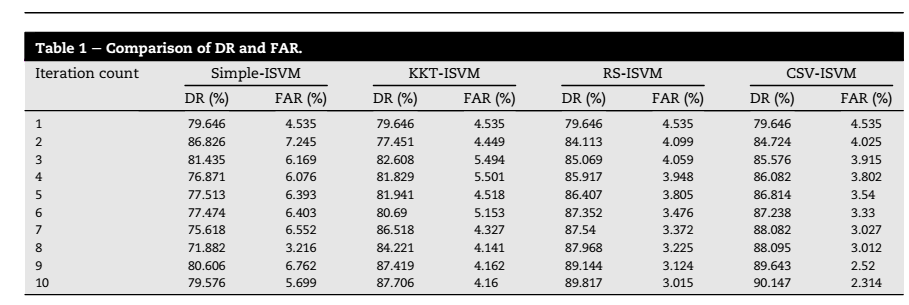


که TP، TN، FP و FN به ترتیب نشان دهنده مثبت واقعی ( حمله شناسایی شده به عنوان مهاجم)، منفی واقعی (نرمال شناسایی شده به عنوان نرمال)، مثبت کاذب (نرمال شناسایی شده به عنوان حمله) و منفی کاذب (حمله شناسایی شده به عنوان نرمال) هستند.

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.

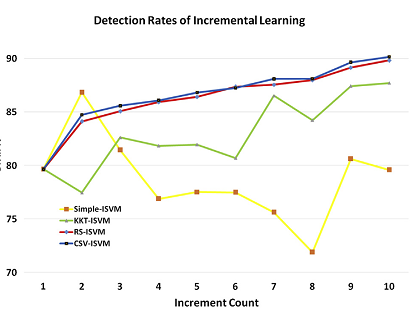
جدول 1 نشان می­دهد که مقادیر DR و FAR در ISVM ساده در یک توسعه افزایش می­یابند درحالیکه در توسعه­ای دیگر کاهش می­یابند یعنی DR در تکرار دوم تا 86.826% افزایش یافته اما در تکرار سوم تا 81.435% کاهش یافته است؛ در مورد FAR هم به همین صورت است. این دلالت براین دارد که آنها به شدت وابسته به مجموعه داده­های 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 ISVM ساده و KKT-ISVM دارای سرعتهای بهتری می­باشند.

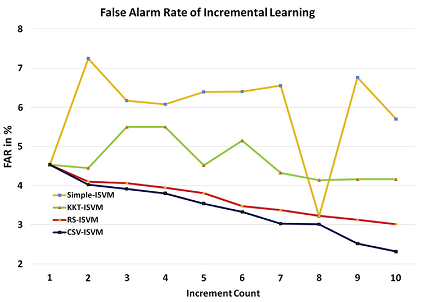


در اولین تکرار نتیجه مشابهی برای هر چهار روش به دست آمد زیرا SVها فقط از تکرار دوم به بعد همراه با مجموعه نمونه­های جدید تمرین می­بینند؛ و نتیجه نهایی از تکرار ده­ام به دست آمده است. در اینجا، RS-ISVM به ترتیب 89.817% , 3.015% از DR و FAR نهایی را داشته، در حالیکه CSV-ISVM به ترتیب 90.147% و 2.314% از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



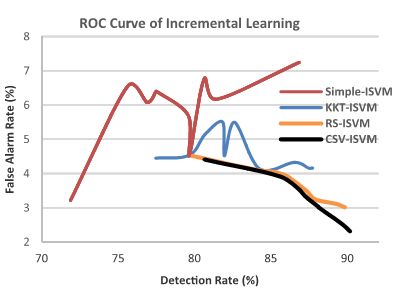
شکل5- مقایسه DR



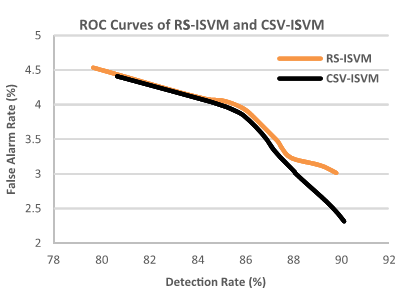
شکل 6- مقایسه ­­FAR

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 ROC ISVM 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 می­کند.

وقتی به طور خاص با RS-ISVM مقایسه می­شود، CSV-ISVM به 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 CSV-ISVM DR 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.

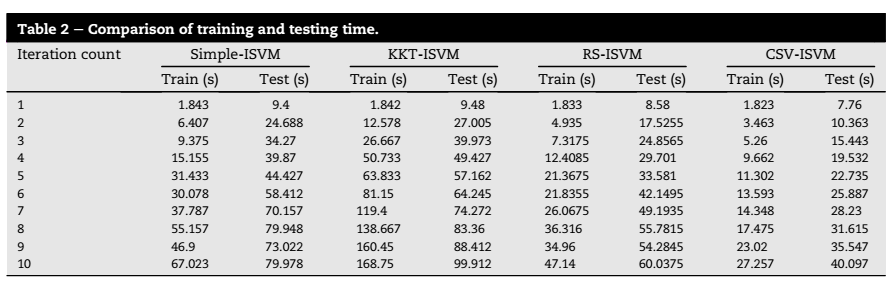


شکل 7- مقایسه منحنی­های ROC



شکل 8- مقایسه روش RS-ISVM و CSV-ISVM

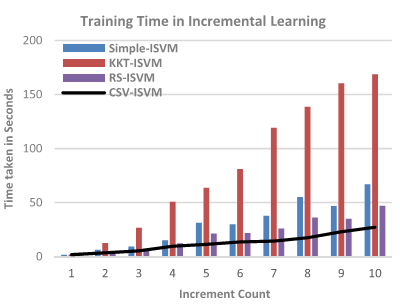
جدول 2 مدت زمان را مقایسه می­کند که توسط هر چهار روش در نمونه داده­های تمرینی و آزمون در هر توسعه گرفته شده است. مدت زمان ثبت شده برای نمونه داده­های تمرینی در افزایش اولیه تقریبا برای چهار روش برابر است یعنی 1.8xx ثانیه، درحالیکه برای نمونه داده­های آزمون در روش CSV-ISVM از 9.4 ثانیه به 7.76 ثانیه کاهش 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.



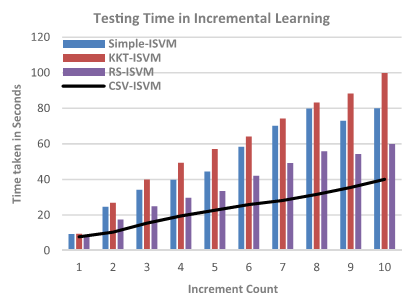
مقادیر زمان گرفته شده توسط هر چهار روش مشخص است که افزایش می­یابند چون افزایشها ادامه دارند، اما RS-ISVM و CSV-ISVM در مقایسه با دو روش دیگر زمان کمتری را برای فراگیری می­گیرند.

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

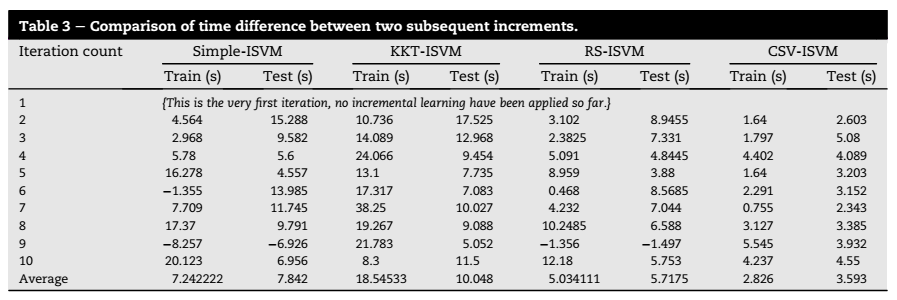


شکل 10- مقایسه زمان آزمون

1. **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.



جدول3- مقایسه اختلاف زمان بین افزایش­های بعدی



