بهینه‌سازی هسته‌های چندگانه در ماشین‌بردارپشتیبان جفتی برای کاهش شکاف معنایی تشخیص صفحات فریب‌آمیز

محمدعلی زارع چاهوکی[۱]، استادیار، سیدحمیدرضامحمدی[۲]، دانشجوی کارشناسی ارشد

۱- دانشکده مهندسی برق و کامپیوتر - دانشگاه یزد - یزد - ایران – chahooki@yazd.ac.ir

۲- دانشکده مهندسی برق و کامپیوتر - دانشگاه یزد - یزد - ایران – mohammadi_6468@stu.yazd.ac.ir

چکیده: موتورهای جستجو با خزش صفحات موجود در اینترنت و شاخص‌گذاری آن‌ها، قابلیت جستجوی سریع اطلاعات را به کاربران می‌دهند. یکی از چالش‌های مهم در استفاده از این ابزار، صفحاتی هستند که از آن‌ها به‌عنوان صفحات فریب‌آمیز نام برده می‌شود. رویکردهای مختلفی جهت تشخیص صفحات فریب ابداع شده است که می‌توان به روش‌هایی مانند سنجش میزان شباهت سبک کدهای صفحات، تحلیل الگوی زبانی صفحات و همچنین استفاده از روش‌های یادگیری ماشین بر اساس ویژگی‌های صفحات اشاره کرد. ازجمله الگوریتم‌های یادگیری ماشین که در این حوزه استفاده شده است ولی نتایج قابل‌توجهی را ارائه نکرده، الگوریتم ماشین‌بردارپشتیبان[۱] (SVM) است. استفاده از هسته در ساختار طبقه‌بند SVM باعث می‌شود که داده‌هایی که دارای الگوی غیرخطی هستند با نگاشت به فضایی با ابعاد بیش‌تر بتوانند با مدل خطی تفکیک‌پذیر شوند. این کار باعث افزایش دقت تفکیک‌کنندگی مدل یادگیری ماشین می‌شود. اخیراً توسعه‌ایی از SVM با نام SVM جفتی[۲] (TSVM) ارائه شده است که با تغییر در فرضیه اولیه آن، از دو اَبَرصفحه[۳] برای تفکیک نمونه‌های هر کلاس استفاده می‌کند و توانسته نتایج بهتری در طبقه‌بندی ارائه کند. به دلیل استفاده از دو ابرصفحه در TSVM، لذا بهتر است تا از هسته‌های چندگانه در ساختار آن استفاده شود. به دلیل اینکه توابع هسته در هر کاربرد اختصاصی هستند لذا نمی‌توان از یک هسته عمومی برای همه کاربردها استفاده کرد. در این مقاله برای بهینه‌سازی ترکیب‌های بهینه توابع هسته پایه، از روشی تکاملی مبتنی بر الگوریتم ژنتیک (GA) استفاده شده است که با بهره‌گیری از آن در فرآیند تصمیم‌گیری هر اَبَرصفحه TSVM، بهبود در تشخیص صفحات فریب حاصل گردیده است. برای پیاده‌سازی و ارزیابی روش پیشنهادی، از مجموعه دادگان UK-2006 و UK-2007 استفاده شده است که نتایج حاصل بیانگر مؤثر بودن ایده پیشنهادی در این پژوهش است.

واژه‌های کلیدی: موتور جستجو، صفحات وب‌فریب، رتبه‌بندی، یادگیری ماشین، ماشین‌بردارپشتیبان جفتی، هسته‌های چندگانه، الگوریتم ژنتیک.

Optimization of Multiple Kernels in Twin SVM for Decreasing Web Spam Page Detection Semantic Gap

M.A.Zare Chahooki, Assistant Professor[1], S.H.R. Mohammadi, MSc Student[2],

1- Faculty of Electrical and Computer Engineering, Yazd University, Yazd, Iran, Email: chahooki@yazd.ac.ir

2- Faculty of Electrical and Computer Engineering, Yazd University, Yazd, Iran, Email: mohammadi_6468@stu.yazd.ac.ir

Abstract: Web pages are crawled and indexed by search engines for fast accessing data on the web. One of the challenges in the search engines is web spam pages. There are many approaches to web spam pages detection such as measurement of HTML code style similarity, pages linguistic pattern analysis and machine learning algorithm on page content features. One of the famous algorithms has been used in machine learning approach is Support Vector Machine (SVM) classifier. Unfortunately SVM could not achieve a reasonable accuracy in this scope. In order to classify non-linear data in a linear manner, the SVM needs to use the idea of the kernel, which leads to enhanced classification capabilities. A kernel, implicitly maps the data to a higher-dimensional space. Recently basic structure of SVM has been changed by new extensions called Twin SVM (TSVM) to increase robustness and classification accuracy using two separate hyperplanes. Because of using two separate hyperplanes in TSVM, it is better to use multiple kernels in it. Kernel functions are designed based on specific data sample. Therefore they cannot use for general purpose. In this paper we improved accuracy of web spam detection by using two nonlinear kernels into TSVM as an improved extension of SVM. These two kernels have been created based on genetic algorithm. The classifier ability to data separation has been increased by using two separated kernels for each class of data. Effectiveness of new proposed method has been experimented with two publicly used spam datasets called UK-2007 and UK-2006.

Keywords: Search engine, web spam page, ranking, machine learning, twin support vector machine (TSVM), multiple kernels, genetic algorithm (GA).

تاریخ ارسال مقاله: ۹۴/۷/۲۱

تاریخ اصلاح مقاله: ۹۴/۱۰/۱۵

تاریخ پذیرش مقاله: ۹۵/۲/۹

نام نویسنده مسئول: محمدعلی زارع چاهوکی

نشانی نویسنده مسئول: ایران - یزد - صفائیه - دانشگاه یزد - پردیس فنی- گروه مهندسی کامپیوتر

۱- مقدمه

با گسترده شدن روزافزون حجم اطلاعات در اینترنت، موتورهای جستجو به مهم‌ترین ابزار برای استخراج اطلاعات تبدیل‌شده‌اند. از نگاه نظری، موتورهای جستجو یکی از ابزارهای بازیابی اطلاعات تلقی می‌شوند که وظیفه جمع‌آوری، شاخص گذاری، پالایش، بازیابی و رتبه‌بندی اطلاعات را به عهده دارند [۱].

یکی از معضلات موجود در موتورهای جستجو که در سایر سامانه‌های بازیابی اطلاعات کم‌تر با آن مواجه هستیم، وجود محتوای مخرب در شبکه اینترنت است. از منظر موتور جستجو، داده‌های مخرب به داده‌هایی گفته می‌شوند که باهدف فریب موتورهای جستجو و به‌تبع آن، دست‌کاری نتایج آن‌ها ایجاد شده‌اند. به دلیل ماهیت کاملاً باز شبکه اینترنت که به‌موجب آن هر شخصی می‌تواند محتوای دلخواه خود را تولید و در شبکه عرضه کند، وجود داده‌های مخرب امری اجتناب‌ناپذیر است. مهم‌ترین نوع داده مخرب در وب، صفحاتی هستند که به‌منظور دست‌کاری نتایج موتور جستجو ایجاد می‌شوند. به این صفحات اصطلاحاً صفحات فریب گفته می‌شود. از انواع دیگر داده‌های مخرب می‌توان به نظرات هرز و پرس‌وجوهای فریب و صفحات فیشینگ[۴] اشاره کرد.

اکثر دارندگان سایت‌های اینترنتی مایل هستند تا سایت آن‌ها در رده‌های بالای نتایج موتورهای جستجو قرار گیرد. علت این موضوع را می‌توان در این مورد مشاهده کرد که تنها ۱۵٪ از کاربران به صفحات دوم به بعد در نتایج موتورهای جستجو مراجعه می‌کنند و در اکثر مواقع اقدام به تغییر پرس‌وجوی خود می‌کنند. به همین دلیل و در راستای افزایش کارآمدی و درآمد، موتورهای جستجو همواره در تلاش هستند تا سایت‌های مرتبط با پرس‌وجو کاربر را در رتبه‌های بالاتر قرار دهند. همین تلاش موتورهای جستجو می‌تواند توسط سودجویان مورد سوءاستفاده قرار گرفته و تولید صفحات فریب را سبب شود [۲].

هدف اصلی صفحات فریب، افزایش رتبه کاذب صفحات اینترنتی از طریق فریب الگوریتم‌های موتورهای جستجو است. در این راستا آنچه اهمیت بیش‌تری دارد دستیابی هرچه سریع‌تر به جایگاه بالاتر در میان ده سایت اول نتایج جستجو با پرس‌وجوهای متفاوت است [۳]. از منظر موتور جستجو، مهم‌ترین اثر تخریبی صفحه فریب، کاهش اعتماد کاربر به موتور جستجو است. همچنین با توجه به تعداد بسیار زیاد این صفحات، منابع زیادی از موتور جستجو (پهنای باند، فضای ذخیره‌سازی و منابع نرم‌افزاری) صرف خزش، شاخص‌گذاری، ذخیره‌سازی و پردازش این صفحات می‌شود.

محوریت موضوع صفحات فریب به‌منظور خدشه‌دار کردن الگوریتم‌های رتبه‌بندی در موتورهای جستجو است. رتبه‌بندی فرآیندی است که در آن کیفیت یك صفحه از جنبه ارتباط با پرس‌وجوی کاربر، توسط موتور جست‌وجو تخمین زده می‌شود. به‌عبارت‌دیگر به هرکدام از اسناد بازیابی‌شده، امتیازی نسبت داده شده و اسناد بر طبق ترتیب نزولی (و یا صعودی) این امتیاز مرتب می‌شوند. موتورهای جستجو معمولاً صفحات وب را بر اساس دو فاکتور زیر رتبه‌بندی می‌کنند:

- ارتباط متن درخواست و صفحه وب: این ارتباط معمولاً از طریق اندازه‌گیری شباهت درخواست ورودی و متن صفحات وب به دست می‌آید.
- اهمیت صفحه وب: منظور از اهمیت، میزان اهمیت آن صفحه بدون در نظر گرفتن درخواست ورودی است. این اهمیت ممکن است با توجه به تعداد ارجاعاتی که از صفحات دیگر به این صفحه شده است اندازه‌گیری شود.

همان‌طور که قبلاً گفته شد، صفحات فریب به‌منظور به دست آوردن رتبه‌های بالا در نتایج موتور جستجو ایجاد می‌شوند. بر همین اساس تولید صفحات فریب بر اساس الگوریتم‌های رتبه‌بندی صفحات در موتورهای جستجو انجام می‌گیرد [۱].

در تولید صفحات فریب بر اساس محتوا، محتوای صفحه هدف به‌گونه‌ای سازمان‌دهی می‌شود که با درخواست‌های بیش‌تری منطبق شود. بنابراین کلمات متنوع و پرکاربرد در تولید صفحه تکرار می‌شوند. درنتیجه این صفحه با تعداد بیش‌تری از درخواست‌های کاربران مطابقت می‌کند. تغییر محتوا با توجه به ساختارهای صفحات وب انجام می‌گیرد. الگوریتم‌های رتبه‌بندی بر اساس محتوا نیز امتیاز بیش‌تری به این صفحات می‌دهند. از طرفی اگر هدف این باشد که صفحه هدف با درخواست‌های خاصی منطبق شود کلمات کلیدی موردنظر باید تکرار بیش‌تری داشته باشند. در ۲۰۰۶ ناجورک و همکاران در [۴] رویکردی مبتنی بر تحلیل محتوایی صفحات وب ارائه کردند که در آن با بررسی ویژگی‌های محتوایی صفحات فریب و غیرفریب به نتایج جالبی در این مورد دست یافتند که بعدها این ویژگی‌ها مبنایی بر تشخیص صفحات فریب شد.

ایجاد پیوندهای زیاد به‌منظور بالا بردن امتیاز صفحه هدف نیز یکی دیگر از روش‌های تولید صفحات فریب است. در [۵-۷] رویکرد تشخیص مبتنی بر پیوند میان صفحات در گراف وب تغییر کرده است. در این رویکرد بر اساس پیوندهای میان صفحات در گراف وب و الگویی که در این گراف بین صفحات است تمایز بین صفحات فریب و غیرفریب ایجاد شده است.

روش‌هایی نیز به وجود آمدند که بر اساس ترکیب روش‌های محتوایی و پیوند به نتایج خوبی دست یافتند. در [۸] با ترکیب ویژگی‌های روش‌های مبتنی بر محتوا و پیوند صفحات، عمل تشخیص صفحات فریب صورت گرفته است. در سال‌های بعد رفتار کاربران در مواجهه با صفحات نیز عاملی تعیین‌کننده در تشخیص صفحات فریب به‌حساب آورده شد از همین رو لیو و همکاران در [۹] روشی مبتنی بر رفتار کاربر ارائه کردند؛ اما از روش‌های نوینی که در ایجاد صفحات فریب استفاده می‌شود تکنیک‌های مخفی‌کاری اطلاعات است در این روش از تولید صفحات فریب، فریب‌گرها با استفاده از نسخه‌های

متفاوتی که در اختیار موتورهای جستجو مرورگر کاربران قرار می‌دهند صفحات فریب خود را ارائه می‌دهند [۱۰].

تاکنون رویکردهای مختلفی جهت تشخیص صفحات فریب ابداع‌شده است که می‌توان به روش‌هایی مثل تحلیل الگوی زبانی صفحات [۱۱]، استفاده از روش‌های یادگیری ماشین بر اساس ویژگی‌های صفحات [۱۶-۱۲] و همچنین سنجش میزان شباهت سبک کدهای صفحات [۱۷] اشاره کرد.

عمده روش‌های تولید صفحات فریب، بر اساس روش‌های مبتنی بر محتوا می‌باشند و تحلیل ویژگی‌های استخراج‌شده از صفحات در این روش به‌منظور تعیین فریب یا غیرفریب بودن، به‌مراتب سریع‌تر و کارآمدتر از روش‌های دیگر است. صفحات در وب، دارای ویژگی‌هایی هستند که با تحلیل محتوایی آن‌ها قابل‌استخراج می‌باشند. این ویژگی‌ها در مواردی که صفحه موردنظر به‌عنوان فریب تلقی می‌شود دارای مقادیری می‌باشند که تفاوت‌های زیادی با نمونه‌های معمولی صفحات دارند. ازآنجایی‌که مسئله تشخیص صفحات فریب، یک مسئله با دو کلاس «فریب» و «نرمال» است به همین دلیل با داشتن یک مجموعه از ویژگی‌های استخراج‌شده از صفحات می‌توان مسئله را تبدیل به یک مسئله طبقه‌بندی دو کلاسه کرده و آن را حل نمود؛ بنابراین با داشتن ویژگی‌های استخراج‌شده از محتوای صفحات و تعیین برچسب هریک از صفحات (فریب و نرمال) می‌توان توسط یک طبقه‌بند مناسب، الگوی موجود بین این صفحات را استخراج کرده و از آن برای تشخیص صفحات بدون برچسب که توسط موتور جستجو جمع‌آوری‌شده‌اند استفاده کرد.

روش‌های مبتنی بر یادگیری ماشین از الگوریتم‌های طبقه‌بندی به‌منظور جداسازی و تشخیص الگوهای موجود میان صفحات عادی و فریب استفاده می‌شود. هرچه الگوریتم مورداستفاده قدرت بیش‌تری در جداسازی صفحات داشته باشد می‌توان به‌دقت بالاتری در زمینه تشخیص دست یافت. در [۱۲] از الگوریتم مدل مخفی مارکوف به‌عنوان طبقه‌بند استفاده شده است. در [۱۳] از درخت تصمیم‌گیری و در [۱۴] از شبکه بیزین که مبتنی بر محاسبات آماری است استفاده شده است. در [۱۵] سیلوا و همکاران از الگوریتم شبکه عصبی مصنوعی در جداسازی نمونه‌های فریب و غیر فریب استفاده شده است.

تاکنون الگوریتم‌های مختلفی در حوزه یادگیری ماشین برای طبقه‌بندی دو کلاسه ارائه شده است. یکی از الگوریتم‌هایی که در این حوزه استفاده شده و نتایج مناسبی را دربر داشته است، طبقه‌بند ماشین‌بردارپشتیبان (SVM) است. این طبقه‌بند هرچند در کاربردهای دیگر نتایج مناسبی در تمیز نمونه‌های مثبت و منفی داشته است ولی در تشخیص صفحات فریب تاکنون دقت مناسبی از آن گزارش نشده است. در همین‌حال پژوهش‌های انجام‌گرفته در سایر کاربردها نشان می‌دهد که استفاده از نمونه‌های توسعه داده شده ماشین‌بردارپشتیبان، دارای عملکرد به‌مراتب بهتری بوده است [۱۸]. نمونه‌های توسعه داده شده ماشین‌بردارپشتیبان با تغییر در ساختار آن، سعی در بهبود عملکرد طبقه‌بند در برخورد با داده‌های مختلف دارند.

ماشین‌بردارپشتیبان، طبقه‌بندی دو کلاسه است که با ایجاد ابرصفحه‌ای[۵] میان نمونه‌های هر کلاس و حداکثر کردن فاصله نمونه‌ها از این صفحه، عمل طبقه‌بندی را انجام می‌دهد. در نمونه‌های توسعه داده شده آن موسوم به ماشین‌بردارپشتیبان جفتی (TSVM) از دو ابرصفحه‌ای مجزا برای هرکدام از انواع نمونه‌ها استفاده می‌شود. با توجه به ماهیت داده‌های صفحات فریب و تعداد ویژگی‌های آن می‌توان از این نمونه توسعه داده شده به‌منظور طبقه‌بندی استفاده کرد.

در برخی مواقع ویژگی‌های نمونه‌ها دارای مقادیری می‌باشند که نمی‌توان آن‌ها را توسط یک صفحه خطی تمیز داد. ایده استفاده از هسته[۶] در روش‌های یادگیری ماشین برای حل این مشکل مطرح گردیده است. در این رویکرد، ابتدا داده‌ها به فضای جدیدی با ابعاد بیش‌تر برده می‌شود. ابعاد جدید داده‌ها به‌گونه‌ای ایجاد می‌شود که بتوان در فضای جدید آن‌ها را با دقت بیش‌تری با ابرصفحه‌ها از هم جدا کرد [۱۹].

در این مقاله از ایده مؤثر بودن استفاده از ماشین‌های بردارپشتیبان جفتی در کاربرد تشخیص صفحات وب فریب استفاده شده است. استفاده از هسته‌های غیرخطی در ساختار ماشین‌بردارپشتیبان باعث افزایش دقت طبقه‌بند و همچنین کاهش میزان پیچیدگی آن‌ها خواهد شد. در این مقاله برای اولین بار از ایده به‌کارگیری دو هسته مجزا برای هریک از ابرصفحاتی که مربوط به یک نمونه می‌باشند استفاده شده است. روش‌های مختلفی برای ایجاد توابع هسته وجود دارد که با توجه به کاربرد و همچنین ساختار داده‌ها، بایستی از آن‌ها استفاده کرد. یکی از روش‌های ایجاد توابع هسته، ترکیب توابع هسته پایه و انتخاب پارامترهای مناسب برای آن است. در این روش، توابع هسته می‌توانند به‌صورت خطی یا غیرخطی با یکدیگر ترکیب شوند. در این مقاله، روشی بر مبنای الگوریتم ژنتیک ارائه شده است که با توجه به داده‌های مسئله و طبقه‌بند TSVM، توابع هسته بهینه را ایجاد می‌کند. نتایج تجربی که به‌واسطه آزمایش‌ها انجام شده بر دادگان UK-2006 و UK-2007 انجام پذیرفته است بیانگر مؤثر بودن رویکرد پیشنهادی در تشخیص صفحات فریب است.

در ادامه این مقاله در بخش ۲ چگونگی عملکرد الگوریتم ماشین‌بردارپشتیبان و نمونه توسعه داده شده آن آورده شده است. به جهت استفاده از الگوریتم ژنتیک در روش پیشنهادی، مروری اجمالی بر این الگوریتم در بخش ۳ بیان شده است. در بخش ۴ به‌تفصیل، توضیح روش پیشنهادی در این مقاله آورده شده است. بیان نتایج تجربی آزمایش‌ها در بخش ۵ و در بخش ۶ نتیجه‌گیری و پژوهش‌های آینده آورده شده است.

۲- توسعه ماشین‌بردارپشتیبان به نسخه جفتی

در این بخش ابتدا به تشریح عملکرد ماشین‌بردارپشتیبان پرداخته و سپس نسخه توسعه‌یافته آن با نام ماشین‌بردارپشتیبان جفتی را توضیح می‌دهیم.

۲-۱- ماشین‌بردارپشتیبان استاندارد

ماشین بردار پشتیبان یك طبقه‌بند دودویی غیر آماری است که در سال‌های اخیر بسیار موردتوجه قرار گرفته است. در این روش با استفاده از تمامی نمونه‌ها و یك الگوریتم بهینه‌سازی، نمونه‌هایی که مرزهای کلاس‌ها را تشکیل می‌دهند به دست می‌آید. با استفاده از این نمونه‌ها یك مرز تصمیم‌گیری خطی بهینه برای جدا کردن کلاس‌ها محاسبه می‌شود و توسط مسائل برنامه‌نویسی مربعی[۷] (QPP) حل می‌شود [۲۰]. فرض کنید داده‌هایی به شکل $D = \{(x_i, y_i) \mid x_i \in \Re^n, y_i \in \{-1,+1\}\}_{i=1}^{n}$ داریم به‌طوری‌که x نمونه دلخواه و y مقدار برچسب هر نمونه است. می‌خواهیم آن‌ها را توسط یک صفحه به‌گونه‌ای طبقه‌بندی کنیم که تمام نمونه‌های + در یک‌طرف صفحه و نمونه‌های − در طرف دیگر صفحه با بیش‌ترین‌حاشیه[۸] نسبت به صفحه قرار گیرند. درواقع مرز تصمیم‌گیری بین دو نمونه، باید به‌گونه‌ای باشد که فاصله نزدیک‌ترین نمونه‌های آموزشی هر دو کلاس از یکدیگر در راستای عمود بر مرز تصمیم‌گیری تا جایی که ممکن است حداکثر شود. مرز تصمیم‌گیری با این شرایط را مرز تصمیم‌گیر یا تفکیک‌کننده بهینه[۹] می‌گویند. برای به دست آوردن یک مرز تصمیم گیر، بایستی معادله آن را نوشت. تفکیک‌کننده‌های خطی مربوط به دو معادله را در حالت کلی می‌توان به شکل زیر نوشت:

$$w.x + b = +1 \quad , w.x + b = -1 \qquad (۱)$$

x یک نقطه بر روی مرز تصمیم‌گیری و w یک بردار نرمال nبعدی عمود بر مرز تصمیم‌گیری است. همان‌طور که قبلاً گفته شد بایستی فاصله بین این دو ابرصفحه بیشینه شود. فاصله بین دو صفحه توسط روابط هندسی و با استفاده از بردار نرمال w که عمود بر ابرصفحه است به‌صورت $\frac{2}{\|w\|}$ به دست می‌آید. اگر بخواهیم رابطه $\frac{2}{\|w\|}$ را بیشینه کنیم کافی است که $\|w\|$ کمینه شود؛ بنابراین تابع هدف ما در این مسئله به‌صورت زیر تبدیل می‌شود که یک مسئله اصلی است.

$$\min \frac{\|w\|^2}{2} \qquad (۲)$$

$$\text{st.} \quad y_i.(w^T.x_i + b) - 1 \geq 0 \quad \forall i\,.$$

برای حل تابع بالا از ضرایب لاگرانژ استفاده کرده و به رابطه زیر می‌رسیم:

$$Lp = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N}\alpha_i\left[y_i(w^T x_i + b) - 1\right] \qquad (۳)$$

$$= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N}\alpha_i y_i(w^T x_i + b) + \sum_i \alpha_i \,.$$

پس از یافتن ضرایب لاگرانژ و حذف w و b از رابطه، به فرم دوگان مسئله می‌رسیم.

$$Ld = -\frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j x_j^T x_i + \sum_i \alpha_i \qquad (۴)$$

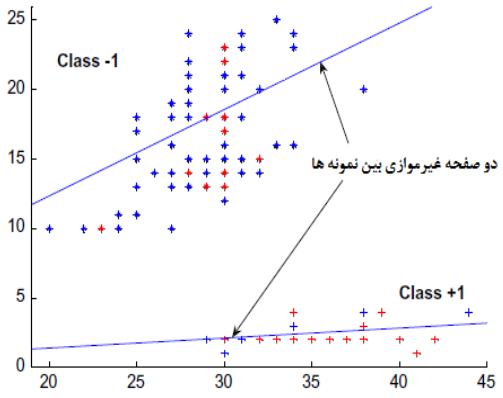
با حل رابطه بالا ضرایب b و w به دست می‌آید. با به دست آوردن مقادیر b و w، طبقه‌بند نهایی به شکل زیر به دست می‌آید. با ورود داده جدید x، کلاس مربوط به آن توسط رابطه زیر محاسبه می‌شود:

$$f(\mathbf{x}, w, b) = w^* x + b^* \qquad (۵)$$

$$= \sum_{i=1}^{N}\alpha_i^* y_i x_i x + b^* = \sum_{i\in SV}\alpha_i^* y_i x_i x + b^*$$

۲-۲- ماشین‌بردارپشتیبان جفتی

این الگوریتم بر مبنای ماشین‌بردارپشتیبان عادی ارائه شده است. در این الگوریتم به‌جای استفاده از یک صفحه واحد و افزایش حاشیه‌های این صفحه به‌سمت نمونه‌ها، برای جدا کردن نمونه‌های دو کلاس، از دو صفحه مجزای غیرموازی استفاده شده است. هر یک از ابرصفحات در نزدیکی نمونه‌های یک کلاس قرار می‌گیرد. ابرصفحات طوری قرار می‌گیرند که در مجاورت یکی از نمونه‌های هر کلاس باشد. حالت بهینه در این ساختار، نزدیک شدن هرچه‌بیش‌تر هر ابرصفحه به یکی از نمونه‌های مربوطه و دور شدن از نمونه‌های کلاس دیگر است [۲۱]. شکل ۱ نحوه عملکرد این طبقه‌بند را نشان می‌دهد.



شکل ۱: نحوه عملکرد ماشین‌بردارپشتیبان جفتی [۲۱]

فرض کنید داده‌هایی با نمونه‌های مثبت و نمونه‌های منفی در اختیارداریم. برای جدا کردن نمونه‌های مثبت از نمونه‌های منفی ابتدا دو معادله به ازای هر ابرصفحه تشکیل می‌دهیم که می‌تواند داده‌ها را به‌صورت خطی در TSVM تمیز دهد.

$$F_+(X) = W_+^T X + b_+ \,,\; F_-(X) = W_-^T X + b_- \qquad (۶)$$

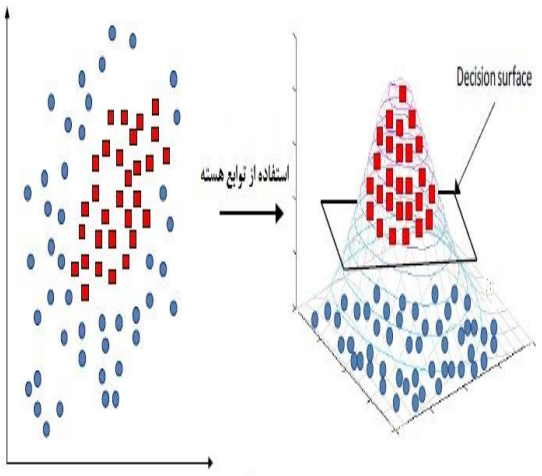
که در این رابطه $W_+, W_- \in \Re^n, b_+, b_- \in \Re$ است. در مرحله بعد برای به دست آوردن ابرصفحات بهینه، بایستی که دو مسئله QPP متناسب با آن حل شود. ازآنجایی‌که حل مسائل QPP مشکل است، می‌توان آن‌ها را از طریق به دست آوردن ضرایب نامعین لاگرانژ، به فرم ساده‌تر تبدیل کرده و حل نمود. پس از حل مسائل QPP، در آخرین مرحله، معادله نهایی TSVM برای جداسازی نمونه‌های هر کلاس به‌صورت زیر به دست می‌آید:

$$Class = \arg \min_{i=1,2} \left|x^T w_i + b_i\right| \qquad (۷)$$

با ورود هر نمونه جدید می‌توان از طریق حل رابطه ۷، کلاس آن نمونه را تشخیص داد.

۲-۳- بهره‌گیری از هسته‌های غیرخطی

استفاده از مدل‌های خطی برای طبقه‌بندی داده‌ها بسیار سریع‌تر و آسان‌تر از مدل‌های غیرخطی است. مدل‌های خطی بر روی داده‌هایی که توسط الگوهای خطی، جداپذیر هستند کار می‌کند. اغلب استفاده از مدل‌های خطی مناسب است چراکه هزینه محاسبات را کاهش و سرعت آن را افزایش می‌دهد؛ اما در برخی مواقع نمونه‌های موجود به‌گونه‌ای نیستند که بتوان از یک مدل خطی برای تمییز آن‌ها استفاده نمود. در چنین مواقعی نیاز است تا داده‌ها به فضای جدید نگاشت[۱۰] شوند تا بتوان در فضای جدید از مدل خطی استفاده کرد. در شکل ۲ نحوه نگاشت داده‌ها به فضای جدید آورده شده است. نگاشت داده‌ها به فضای جدید همواره به‌آسانی انجام نمی‌پذیرد و معمولاً هزینه بالایی را تحمیل می‌کند. ایده هسته، این نقیصه را به‌خوبی پوشش داده و راهکاری را ارائه می‌دهد تا بتوان با هزینه کم این کار را انجام داد. هسته به‌صورت ضمنی عمل نگاشت داده‌ها به فضای بالا را انجام می‌دهد. ماشین‌بردارپشتیبان نیز برای اینکه بتواند به‌صورت خطی بروی داده‌های غیرخطی جداسازی را انجام دهد نیاز دارد تا از ایده هسته استفاده کند. استفاده از ایده هسته، منجر به افزایش قدرت طبقه‌بندی ماشین‌بردارپشتیبان می‌شود [۱۹]. شکل ۲ نحوه عملکرد تابع هسته بر روی داده‌ها را نشان می‌دهد.



شکل ۲: ایده استفاده از هسته [۱۹]

به‌طورکلی در الگوریتم‌هایی که داده‌ها به‌صورت ضرب نقطه‌ای ظاهر می‌شوند می‌توان از ایده هسته استفاده کرد. متناسب با کاربرد مسئله، انواع مختلفی از توابع هسته ایجاد شده‌اند. از مهم‌ترین این هسته‌ها که عمومیت بیش‌تری نیز دارند می‌توان به هسته‌های چندجمله‌ای، گوسی، خطی، فازی و موارد دیگر اشاره کرد [۱۹].

در زیر رابطه نهایی طبقه‌بند TSVM با استفاده از ایده هسته بازنویسی شده است.

$$Class = \arg \min_{i=1,2} \left| x^T w_i + b_i \right| \qquad (۸)$$

$$= \arg \min_{i=1,2} \left| k(x^T, c^T) w_i + b_i \right|$$

۳- الگوریتم ژنتیک

الگوریتم ژنتیک یک روش جستجوی مؤثر در فضاهای بسیار وسیع و بزرگ است که درنهایت منجر به جهت‌گیری به‌سمت پیدا کردن یک جواب می‌گردد. الگوریتم ژنتیک با یک سری متغیرهای کدشده کار می‌کنند. مزیت کار با متغیرهای کدشده در این است که اصولاً کدها قابلیت تبدیل فضای پیوسته به فضای گسسته را دارند. به‌طورکلی، الگوریتم ژنتیک از اجزاء زیر تشکیل می‌شوند:

کروموزوم: در الگوریتم ژنتیکی، هر کروموزوم، نشان‌دهنده یک نقطه در فضای جستجو و یک راه‌حل ممکن برای مسئله موردنظر است. خود کروموزوم‌ها (راه‌حل‌ها) از تعداد ثابتی ژن (متغیر) تشکیل می‌شوند. برای نمایش کروموزوم‌ها، معمولاً از کدگذاری‌های دودویی (رشته‌های بیتی) استفاده می‌شود.

جمعیت: مجموعه‌ای از کروموزوم‌ها یک جمعیت را تشکیل می‌دهند. با تأثیر عملگرهای ژنتیکی بر روی هر جمعیت، جمعیت جدیدی با همان تعداد کروموزوم تشکیل می‌شود.

تابع برازندگی: به‌منظور حل هر مسئله با استفاده از الگوریتم‌های ژنتیکی، ابتدا باید یک تابع برازندگی برای آن مسئله ایجاد شود. برای هر کروموزوم، این تابع عددی غیرمنفی را برمی‌گرداند که نشان‌دهنده شایستگی یا توانایی فردی آن کروموزوم است [۲۲].

الگوریتم ژنتیک توسط عملگرهای تعریف‌شده آن، مجموعه‌ای از پاسخ‌های مسئله را طی فرآیندی تکراری بهینه می‌کند. عملگرهای مهم الگوریتم ژنتیک، عملگر تقاطع[۱۱] و عملگر جهش[۱۲] است. این عملگرها باعث می‌شود تا جواب‌های بهینه، شکل بگیرد.

۴- روش پیشنهادی

در این مقاله چارچوبی ارائه شده است که با ترکیب الگوریتم ژنتیک و طبقه‌بند TSVM، دقت طبقه‌بندی داده‌ها افزایش یافته است. در این رویکرد بر مبنای داده‌های مسئله، تابع هسته بهینه مرتبط با نمونه‌های کاربرد موردنظر ایجاد می‌شود. در این چارچوب، هدف، پیدا کردن تابع هسته مناسب با نمونه‌ها و استفاده از آن در ساختار طبقه‌بند TSVM به‌منظور انجام عملیات طبقه‌بندی است.

۴-۱- هسته بهینه بر مبنای الگوریتم ژنتیک

در بخش قبل به طبقه‌بندهایی اشاره شد که در ساختار آن‌ها می‌توان از توابع هسته بهره گرفت. توابع هسته به‌منظور نگاشت ضمنی داده‌ها به عمقی جدید برای استفاده از تفکیک‌کننده‌های خطی ایجاد می‌گردند. ازآنجایی‌که ساختار برخی داده‌ها به‌گونه‌ای است که نمی‌توان از تفکیک‌کننده‌های خطی استفاده کرد لذا می‌توان با استفاده از ایده هسته ابتدا داده‌ها را به عمقی جدید تبدیل کرد و سپس تفکیک‌کننده‌های خطی را به کار گرفت.

همان‌طور که اشاره شد یکی از روش‌های ایجاد توابع هسته، ترکیب توابع هسته پایه است. در این روش، توابع هسته پایه به‌صورت خطی و یا غیرخطی با یکدیگر ترکیب شده و پارامترهای موردنیاز آن‌ها

نیز تنظیم می‌گردد؛ اما اینکه چه نوع هسته‌ای انتخاب شود، چگونه ترکیب شوند و پارامترهای آن با چه مقادیری تنظیم شوند، یکی از چالش‌های طراحان توابع هسته است. توابع هسته بر مبنای داده‌ها شکل می‌گیرد؛ بنابراین در هر کاربرد، هسته استفاده‌شده برای داده‌های مربوطه، اختصاصی است. توابع هسته را نمی‌توان برای نگاشت هر نوع داده‌ای به کار گرفت. به همین دلیل، بایستی رویکردی ارائه دهیم تا بتواند متناسب با نمونه‌های هر مسئله، مناسب‌ترین هسته ممکن را ایجاد کند. مقادیر مناسب برای پارامترها و نیز ضرایب تأثیرگذاری هر تابع در هسته نهایی ازجمله پارامترهایی است که بایستی مقادیر بهینه برای آن‌ها تنظیم شود. ازهمین‌رو می‌توان یافتن توابع هسته مناسب برای هر کاربرد را به‌نوعی یک مسئله بهینه‌سازی دانست [۱۹].

همان‌طور که در بخش قبل اشاره شد الگوریتم ژنتیک یکی از الگوریتم‌های بهینه‌سازی است که مقادیر بهینه مسئله را با تکرارهای متوالی به دست می‌آورد. در روش ارائه‌شده در این مقاله از الگوریتم ژنتیک برای پیدا کردن توابع هسته مناسب برای داده‌های صفحات فریب استفاده شده است. در شکل ۳ گام‌های اصلی روش پیشنهادی برای ایجاد هسته بهینه غیرخطی، ارائه شده است.

۱- انتخاب توابع هسته پایه برای ترکیب و مشخص کردن روش ترکیب آن‌ها.
۲- استخراج پارامترهای موردنیاز برای بهینه‌سازی.
۳- ایجاد جمعیت اولیه از پارامترها.
۴-ارزیابی جمعیت ایجادشده بر مبنای تابع برازندگی (در این مرحله هسته موردنظر توسط پارامترهای ایجادشده تشکیل می‌گردد و در ساختار TSVM استفاده می‌گردد).
۵- انجام عملگرهای الگوریتم ژنتیک بر روی جمعیت اولیه و تولید جمعیت جدید.
۶- تکرار مراحل ۴ و ۵.
۷- ساخت طبقه‌بند TSVM نهایی توسط هسته بهینه.

شکل ۳: گام‌های اصلی روش پیشنهادی

۴-۲- انتخاب و ترکیب هسته‌های بهینه

در ابتدا تعدادی تابع هسته پایه انتخاب می‌شوند. انتخاب توابع هسته پایه می‌تواند تأثیر به سزایی در ایجاد هسته نهایی داشته باشد. در این روش، توابع پایه‌ای انتخاب‌شده که بیش‌ترین کاربرد و دقت را در حل مسائل مختلف داشته‌اند. لیست توابع انتخابی برای ساخت هسته نهایی در جدول ۱ آورده شده است.

جدول۱: توابع هسته پایه

ردیف	رابطه	نام تابع
۱	$K_{Lin}(x_i, x_j) = x_i x_j + b$	Linear(K_{Lin})
۲	$K_{RBF}(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$	RBF(K_{RBF})
۳	$K_{Poly}(x_i, x_j) = (\alpha(x_i..x_j) + \beta)^d$	Polynomial(K_{poly})
۴	$K_{Sig}(x_i, x_j) = \tanh(\alpha(x_i..x_j) + \beta)$	Sigmoidal(K_{sig})

پس از اینکه توابع هسته انتخاب شد بایستی که نحوه ترکیب آن‌ها به‌منظور ساخت هسته نهایی مشخص شود. در اینجا روشی ارائه می‌شود که بر مبنای آن، هسته‌های ذکرشده با یکدیگر ترکیب می‌شوند. در رابطه ارائه‌شده سعی بر این بوده تا بتوان پارامترهای کنترلی بیش‌تری را به‌صورت متغیر تعریف کرد. چگونگی ترکیب هسته‌ها، میزان تأثیرگذاری هر هسته در رابطه نهایی و پارامترهای دیگر بیانگر منعطف بودن رابطه است. رابطه ۹ نحوه ترکیب توابع هسته پایه را بیان می‌کند.

$$K_{Final} = w_1 K_{Lin}^{e_1} + w_2 K_{Poly}^{e_2} + w_3 K_{RBF}^{e_3} + w_4 K_{Sig}^{e_4} \quad (۹)$$

در این رابطه K_{Final} هسته نهایی است که از ترکیب چهار هسته پایه به دست می‌آید. w_i ضرایب وزنی هر تابع است. درواقع این ضریب بیان می‌کند هر هسته به چه میزان در رابطه نهایی تأثیرگذار باشد. e_i توان تأثیرگذار هر هسته در هسته نهایی است.

۴-۳- ایجاد کروموزوم‌ها

پس از آنکه رابطه هسته نهایی مشخص شد نوبت به آن می‌رسد تا رشته کروموزومی را ایجاد کنیم. همان‌طور که قبلاً گفته شد یافتن مقادیر بهینه برای پارامترهای تابع هسته نهایی، توسط الگوریتم ژنتیک انجام می‌گیرد. برای استفاده از الگوریتم ژنتیک ابتدا بایستی ورودی‌های الگوریتم را فراهم کنیم. ورودی الگوریتم در این مسئله، پارامترهای استخراج‌شده از رابطه هسته نهایی است که در قسمت قبل گفته شد. پارامترهای استخراج‌شده از مسئله، در قالب رشته کروموزومی بیان می‌شود. درواقع نگاشت مسئله به رشته کروموزومی در این مرحله انجام می‌گیرد. در این مرحله بایستی با انتخاب ژن‌های مربوط به رشته کروموزوم، جمعیت اولیه از پاسخ را ایجاد کنیم. ما در این مسئله به دنبال پیدا کردن مقادیر مناسب برای پارامترهای w، e و همچنین پارامترهای α_i و β_i به‌علاوه پارامترهای d و b هستیم؛ بنابراین رشته کروموزومی ما در اینجا که مجموعه‌ای از جواب ممکن را در بردارد بایستی شامل متغیرهای گفته‌شده باشد. پارامترهای e و w که قبلاً توضیح داده شد. پارامترهای α_i و β_i به ترتیب شامل ضریب ثابت و مقدار بایاس در توابع هسته مربوطه است. پارامترهای b و d نیز به ترتیب مربوط ضریب بایاس در هسته خطی و مقدار توان در هسته چندجمله‌ای است. لیست کلیه پارامترهای موردنیاز برای ایجاد هسته نهایی در جدول ۲ آورده شده است.

جدول۲: پارامترهای موجود در رشته کروموزوم

w_1	e_1	w_2	e_2	w_3	e_3	w_4	e_4	α_1	α_2	β_1	β_2	b	d

در تمامی مراحل الگوریتم ژنتیک سعی داریم تا این رشته از پارامترها را بهینه کرده و بهترین جواب ممکن را به دست آوریم.

جدول۲ شامل تمامی متغیرهای حاضر در رابطه ۹ است که می‌بایست مقدار بهینه برای آن‌ها محاسبه شود. دامنه مقادیر ممکن برای هر پارامتر بر اساس هر تابع، در جدول ۳ آورده شده است.

جدول۳: دامنه مقادیر مجاز هر پارامتر در هسته نهایی

ردیف	دامنه	پارامتر
۱	محدوده بین ۰ تا ۱	w_1, w_2, w_3, w_4
۲	محدوده بین ۰ تا ۳	d
۳	محدوده اعداد حقیقی	b
۴	محدوده اعداد حقیقی	α_1, α_2
۵	محدوده اعداد حقیقی	β_1, β_2

۴-۴- اعمال عملگرهای الگوریتم ژنتیک

پس از ایجاد جمعیت اولیه، عملگرهای الگوریتم ژنتیک را برای ارزیابی و همچنین بازتولید جمعیت اولیه به جهت تولید جمعیتی بهینه به کار می‌گیریم. اولین عملی که باید انجام پذیرد ارزیابی است. در فرایند ارزیابی بررسی می‌شود که با توجه به هدف مسئله آیا جواب کنونی به‌دست‌آمده، جواب بهینه‌ای برای مسئله است یا خیر؟ این کار با بررسی معیارهای مربوط به سنجش بهینگی پاسخ یا همان تابع برازندگی انجام می‌گیرد. درصورتی‌که جواب به‌دست‌آمده جواب بهینه‌ای نباشد، الگوریتم ژنتیک توسط عملگرهای تعریف‌شده، جمعیت جدید را بازتولید می‌کند؛ و این کار تا جایی ادامه پیدا می‌کند تا جواب بهینه حاصل شود و یا اینکه تعداد تکرارهای الگوریتم ژنتیک به اتمام برسد. عملگرها و همچنین تابع برازندگی الگوریتم ژنتیک در روش پیشنهادی به شکل زیر است:

تابع برازندگی: تابع برازندگی در این روش، میزان دقت طبقه‌بندی الگوریتم TSVM است که به‌صورت رابطه ۱۰ تعریف می‌شود.

$$Accuracy = \frac{TP + TN}{P + N} \qquad (۱۰)$$

P تعداد کل نمونه‌های مثبت، N تعداد کل نمونه‌های منفی، TP تعداد نمونه‌های مثبت درست تشخیص داده شده و TN تعداد نمونه‌های منفی درست تشخیص داده شده است.

عملگر انتخاب: عملگر انتخاب در این مسئله کروموزوم‌هایی را انتخاب می‌کند که مقدار تابع برازندگی آن‌ها بیش‌تر باشد. مقادیر مربوط به ارزیابی کروموزوم‌ها به‌صورت نزولی مرتب شده و کروموزومی انتخاب می‌شود که بیش‌ترین مقدار برازندگی را داشته باشد.

عملگر جفت‌یابی: بعد از انتخاب کروموزوم‌های مناسب برای تولید نسل جدید، هم‌اکنون نوبت به‌کارگیری سیاستی جهت انتخاب زوج‌های مناسب است. در اینجا ما از روش تورنمنت برای انتخاب زوج کروموزوم‌ها از بین کروموزوم‌ها استفاده می‌کنیم. این روش شانس بیش‌تری در اختیار همه کروموزوم‌ها قرار می‌دهد.

عملگر آمیزش: توسط این عملگر، دو کروموزوم جدید از آمیزش هر جفت پدر و مادر انتخابی در مرحله قبل تولید می‌شود. معمول‌ترین شکل آمیزش و تولید نسل، استفاده از نقطه تقاطع است. فرمول تقاطع دو کروموزوم در اینجا به شکل زیر است.

$$P_{Offspring} = \omega Parent_1 + (1-\omega) Parent_2 \qquad (۱۱)$$

$P_{Offspring}$ فرزند تولیدشده از تقاطع، $Parent_1$ و $Parent_2$ والدین شرکت‌کننده در عملگر تقاطع و ω مقدار ثابت در بازه (۰ و ۱) است. در این مسئله مقدار ω برابر با ۰/۵ است. مقدار ۰/۵ برای امکا این امکان را فراهم می‌کند تا سهم هر یک از والدین در تولید فرزند جدید به‌صورت مساوی باشد. در این روش چون هیچ اولویت و امتیاز خاصی برای والدین در نظر گرفته نمی‌شود لذا سهم هریک، برابر در نظر گرفته می‌شود.

عملگر جهش: ژن‌هایی که دچار جهش ژنی می‌شوند به‌صورت اتفاقی انتخاب می‌شوند. مقدار انتخاب‌شده برای نرخ جهش در این مسئله ۰/۱۵ است. این مقدار بیان می‌کند هر یک از ژن‌ها با احتمال ۱۵ درصد دچار تغییر می‌شوند.

۴-۵- ارائه TSVM برمبنای هسته‌های چندگانه بهینه

در بخش قبل مشاهده شد که با ترکیب دو تکنیک الگوریتم ژنتیک و طبقه‌بند TSVM توانستیم هسته‌ای بهینه برای داده‌های مسئله موردمطالعه ایجاد کرده و دقت طبقه‌بندی را به‌واسطه استفاده از هسته ایجادشده در ساختار TSVM افزایش دهیم.

همان‌طور که در بخش دو توضیح داده شد طبقه‌بند TSVM از دو ابر صفحه به ازای هر یک از نمونه‌ها استفاده می‌کند. در این مقاله ما از همین نکته استفاده کرده و به ازای هر یک از ابرصفحات، هسته‌ای جداگانه طراحی می‌کنیم. درواقع نمونه‌های مربوط به هر کلاس، با یک تابع هسته مناسب، به عمق جدید نگاشت می‌شود.

انتخاب هسته بهینه به ازای هر یک از نمونه‌ها توسط رویکرد پیشنهادی انجام می‌گیرد. در این رویکرد دو مجموعه جواب به طبقه‌بند داده می‌شود که هریک مخصوص یک تابع هسته است. طبقه‌بند با استفاده از این دو هسته، ابتدا داده‌ها را نگاشت کرده و سپس طبقه‌بندی داده‌ها را انجام می‌دهد. رابطه ۱۲ با استفاده از دو هسته مجزا در ساختار TSVM بازنویسی شده است.

$$Class = \arg \min_{i=1,2} \left| x^{\mathrm{T}} w_i + b_i \right| \qquad (۱۲)$$
$$= \arg \min_{i=1,2} \left| k_i(x^{\mathrm{T}}, c^T) w_i + b_i \right|$$

در این رابطه به ازای ابرصفحه i تابع هسته k_i استفاده شده است

۴-۶- تحلیل پیچیدگی زمانی

پیچیدگی محاسباتی رویکرد پیشنهادی را می‌توان بر اساس پیچیدگی الگوریتم‌های ژنتیک و TSVM محاسبه کرد. در رابطه ۱۳ نحوه محاسبه پیچیدگی رویکرد پیشنهادی آورده شده است [۲۲].

$$O(O(\ fitness) \times \left(O(Mutation) + O(CrossOver)\right) \qquad (۱۳)$$

ازآنجایی‌که مقادیر عملگرهای جهش و ترکیب به صورت ثابت درنظر گرفته می‌شود لذا پیچیدگی نهایی الگوریتم، برابر با پیچیدگی تابع برازندگی می‌شود. همان‌طور که قبلاً بحث شد تابع برازندگی، تابع محاسبه میزان دقت در الگوریتم TSVM می‌باشد. این مقدار در

الگوریتم TSVM در حالت آموزش برابر است با $\#iteration \times O(2l)$ و در حالت آزمون برابر با $O(2l)$ می‌باشد. در این رابطه، l برابر با اندازه ویژگی‌های موجود می‌باشد. *iteration* نیز تعداد تکرارهای الگوریتم ژنتیک می‌باشد. رابطه نهایی رویکرد پیشنهادی به شکل زیر بیان شده است:

$$O(\ fitness) = O(2l) \qquad (۱۴)$$

پیچیدگی زمانی مدل برای طبقه‌بندی نمونه جدید نیز در رابطه ۱۶ محاسبه می‌شود [۲۱].

$$O(\mathrm{mod}\ el) = \frac{n^3}{4} \qquad (۱۶)$$

۵- نتایج آزمایش‌های تجربی

در این بخش به بیان چگونگی پیاده‌سازی و ارزیابی رویکرد پیشنهادی می‌پردازیم.

۵-۱- دادگان

به‌منظور آموزش الگوریتم پیشنهادی و همچنین اجرای آن و مشاهده دقت نتایج از دو مجموعه دادگان UK-2006 و UK-2007 استفاده شده است. این دو مجموعه دادگان در حوزه ارزیابی روش‌های تشخیص صفحات فریب، در پژوهش‌های متعددی استفاده شده‌اند. مجموعه دادگان UK-2006 شامل حدود ۱۱۴۰۰ هاست از دامنه uk. است. اسناد دادگان مورداستفاده در این مجموعه ۶۱/۷۵ درصد با برچسب نرمال، ۲۲/۰۸درصد برچسب فریب و ۱۶/۱۶درصد از صفحات نیز با برچسب نامشخص، مشخص شده است [۲۳]. مجموعه دادگان UK-2007 از ۱۱۴۵۲۹ هاست از دامنه uk. جمع‌آوری شده است. ۹۴درصد از صفحات این مجموعه برچسب نرمال و ۶ درصد آن نیز برچسب فریب دارا می‌باشند. این دادگان شامل دو دسته ویژگی پیوندی و محتوایی است. در این پژوهش از ویژگی‌های محتوایی دادگان استفاده شده است. ویژگی‌های محتوایی مانند تعداد کل کلمات صفحه، تعداد کلمات عنوان، تعداد تصاویر، میانگین طول کلمات، تعداد کلمات یکتا و ویژگی‌های دیگر مرتبط با محتوای صفحات می‌باشند.

به‌منظور انجام آزمایش‌های، از نمونه‌های با برچسب نامشخص صرف‌نظر شده است. همچنین به‌صورت تصادفی ۷۵ درصد از داده‌ها به‌عنوان داده‌های آموزش برای ساخت مدل و ۲۵ درصد از آن به‌عنوان داده‌های آزمون در نظر گرفته شده است.

۵-۲- معیار و روش ارزیابی

روش پیشنهادی در محیط متلب پیاده‌سازی شده و بر روی رایانه‌ای با دو پردازنده ۱/۲ گیگاهرتزی و حافظه موقت ۲ گیگابایت بر روی بستر سیستم‌عامل ویندوز ۸ اجرا شده است.

به‌منظور نشان دادن نتایج الگوریتم پیشنهادی و همچنین مقایسه آن با سایر الگوریتم‌ها بایستی که از یک معیار ارزیابی استاندارد استفاده کرد. معیار ارزیابی استفاده‌شده، میزان دقت است؛ که میزان مواردی که به‌درستی تشخیص داده شده است را نشان می‌دهد. این معیار که کاربرد زیادی در حوزه وب‌فریب دارد در مقایسه با سایر معیارها از پایداری بیش‌تری برخوردار است (رابطه ۱۷).

$$Accuracy = \frac{TP + TN}{P + N} \qquad (۱۷)$$

P تعداد کل نمونه‌های مثبت،N تعداد کل نمونه‌های منفی،TP تعداد نمونه‌های مثبت درست تشخیص داده شده و TN تعداد نمونه‌های منفی درست تشخیص داده شده است.

دادگان به دو بخش آموزش و آزمون تقسیم می‌شود. قسمت آزمون که به‌صورت تصادفی ۲۵٪ از کل دادگان است برای ارزیابی و قسمت آموزش که به‌صورت تصادفی ۷۵٪ از کل دادگان است برای استفاده در الگوریتم بهینه‌سازی استفاده می‌شود. قسمت دادهای آموزش نیز به دو بخش آموزش و اعتبارسنجی تقسیم می‌شوند. در هر مرحله از اجرای الگوریتم ژنتیک، هسته به‌وسیله ضرایب به‌دست‌آمده تشکیل شده و پس‌ازآن طبقه‌بند با استفاده از داده‌های قسمت آموزش مدل را آموزش می‌دهد. پس از آموزش مدل، دقت آن توسط قسمت اعتبارسنجی ارزیابی می‌شود.

برای اینکه ضرایب به‌دست‌آمده به قسمتی از داده‌ها بایاس نشود از روش ده-بخشی[۱۳] استفاده می‌شود. به این صورت که قسمت آموزش به ده قسمت تقسیم شده و هر بار یک قسمت به‌عنوان اعتبارسنجی و نه قسمت باقی‌مانده به‌عنوان آموزش استفاده می‌شود. پس از اینکه ضرایب نهایی به دست آمد توسط داده‌های آزمون ارزیابی می‌شوند.

انتخاب تصادفی نمونه‌های آموزشی و داده‌های آزمایش باعث تغییر در دقت نتایج در هر آزمایش می‌گردد. بنابراین انتخاب تصادفی داده‌های آموزش وآزمون ۱۰۰ بار صورت پذیرفته است و نتایج با مدل آماری گاوسی مبتنی بر قضیه حد مرکزی[۱۴] تخمین زده شده است [۲۴]. در این روش به‌جای استفاده از مقدار میزان دقت از مقادیر میانگین دقت‌ها به همراه انحراف معیار به شکل $\mu \pm 2\sigma$ با فاصله اطمینان ۹۵٪ استفاده شده است که در آن μ مقدار میانگین و σ مقدار انحراف معیار است. در مقایسه‌هایی که در دو دادگان UK-2006 و UK-2007 صورت پذیرفته است، دیگر پژوهش‌ها تنها نتایج یک انتخاب تصادفی را گزارش کرده‌اند.

۵-۳- مقایسه الگوریتم پیشنهادی با نمونه‌های SVM

الگوریتم SVM استاندارد یکی از طبقه‌بندهایی است که علی‌رغم کارکرد خوب در حوزه طبقه‌بندی داده‌ها، به لحاظ نوع داده‌های موجود در حوزه صفحات فریب، نتایج خوبی را در این حوزه ارائه نداده است. ما ابتدا الگوریتم SVM استاندارد را با بهره‌گیری از هسته‌های مختلف پیاده کردیم و نتایج حاصل از آن را با نتایج الگوریتم پیشنهادی در جدول۴ آورده‌ایم. نتایج نشان می‌دهد که در SVM استاندارد تأثیر استفاده از هسته چندان قابل‌توجه نیست که می‌توان دلیل آن را در ساختار خود طبقه‌بند جستجو کرد. در طبقه‌بند SVM استاندارد به

دلیل استفاده از فقط یک صفحه متمایزکننده در ساختار آن، قدرت جداسازی نمونه‌های فریب و نرمال که نسبتاً شبیه به هم هستند کاهش می‌یابد. در داده‌های صفحات فریب، نمونه‌های شبیه به هم زیادی با برچسب‌های متفاوت وجود دارد. ازاین‌رو توانایی طبقه‌بند SVM در این حوزه کاهش می‌یابد. در جدول ۴ رویکرد پیشنهادی با الگوریتم استاندارد SVM به همراه هسته‌های مختلف مورد مقایسه قرار گرفته است.

جدول ۴: مقایسه دقت طبقه‌بندی الگوریتم پیشنهادی با SVM و TSVM با هسته‌های مختلف

ردیف	الگوریتم	نوع هسته استفاده شده	دقت(٪) UK-2007	دقت(٪) UK-2006
۱	SVM	Linear	۸۰	۸۲
		RBF	۸۱/۴	۸۴
		Hyperbolic	۸۴	۸۴
۲	TSVM (One Kernel)	Linear	۹۲	۸۸
		RBF	۹۲	۸۸/۳
		Custom Kernel	۹۴	۹۰
۳	TSVM (Two Kernel)	Kernel1 = Linear Kernel2 = RBF	۹۲	۹۰
		Kernel1 = RBF Kernel2 = CK	۹۴	۹۲
		Kernel1 = Linear Kernel2 = CK	۹۵/۶	۹۳
۴	TSVM (Optimized Kernel)	Kernel1 = Optimized Kernel Kernel2 = Optimized Kernel	± ۱/۷۲ ۹۶/۴۲	± ۱/۶۴ ۹۵/۳۴

در جدول بالا همان‌طور که مشاهده می‌شود استفاده از هسته در ساختار SVM باعث بهبود نتایج و افزایش دقت طبقه‌بندی شده است. استفاده از هسته‌های غیرخطی باعث می‌شود تا نگاشت داده‌ها به عمق بالاتر بهتر انجام گیرد که نتیجه آن افزایش دقت طبقه‌بند می‌شود. ازآنجایی‌که طبقه‌بند TSVM به لحاظ ساختاری از دو ابرصفحه استفاده می‌کند باعث می‌شود که قدرت تفکیک‌کنندگی طبقه‌بند بالا رود. به‌منظور افزایش دقت طبقه‌بندی داده‌های صفحات فریب از این طبقه‌بند استفاده شد و باعث بهبود نتایج گردید.

اما همان‌طور که در جدول۴ مشاهده می‌گردد استفاده از هسته در ساختار TSVM باعث افزایش دقت طبقه‌بندی گردیده است. قبلاً اشاره شد که به دلیل استفاده TSVM از دو ابرصفحه به ازای هر یک از نمونه‌ها، می‌توان از دو هسته مجزا برای هریک از ابرصفحات استفاده کرد. در رویکرد پیشنهادی با ایجاد هسته‌های بهینه برای هر یک از ابرصفحات باعث افزایش دقت طبقه‌بندی شده است.

۵-۴- مقایسه الگوریتم پیشنهادی با الگوریتم‌های دیگر

همان‌طور که در بخش دو اشاره شد از طبقه‌بندهای مختلفی به‌منظور تشخیص صفحات فریب استفاده شده است. در [۲۵] از درخت تصمیم‌گیری و در [۲۶] سیلوا و همکاران از الگوریتم شبکه عصبی مصنوعی در جداسازی نمونه‌های فریب و غیرفریب استفاده کرده‌اند. از مدل مخفی مارکوف نیز در تشخیص صفحات فریب استفاده شده است [۲۷]. در جدول ۵ نتایج مقایسه رویکرد پیشنهادی با سایر روش‌ها بر روی دادگان UK-2006 آورده شده است و همچنین در جدول۶ نیز رویکرد پیشنهادی با روش‌های دیگر بر روی دادگان UK-2007 بیان شده است. در این جداول، نتایج رویکرد پیشنهادی به همراه نتایج برخی از الگوریتم‌هایی که بهترین دقت را در نتایج حاصل از طبقه‌بندی بر روی دادگان موردنظر ارائه کرده‌اند آورده شده است. این الگوریتم‌ها در حوزه تشخیص صفحات فریب و بر مبنای محتوای صفحات عمل طبقه‌بندی را انجام داده‌اند.

جدول ۵: مقایسه دقت الگوریتم پیشنهادی با الگوریتم‌های دیگر بر روی دادگان UK-2006

UK-2006			
ردیف	الگوریتم	میزان دقت (٪)	F-M
۱	Neural Network[۲۶]	۸۷/۶	-
۲	HMM[۲۷]	-	۰/۸۵۹
۳	Decision Tree[۲۵]	-	۰/۹۲
۴	TSVM	۹۰	۰/۹۱۸
۵	MKTSVM	۹۳	۰/۹۴۱
۶	OPTKTSVM[۱۵]	۹۵/۳۴ ±۱/۶۴	-

استفاده از هسته غیرخطی در ساختار TSVM باعث افزایش دقت نسبت سایر روش‌های ذکرشده است. نکته قابل‌تأمل در این روش تبدیل فضای ورودی داده‌ها به‌گونه‌ای است که بتوان با استفاده از یک مدل خطی، آن‌ها را تمیز داد. انتخاب و طراحی هسته متناسب با داده‌ها یکی از نکات مهم مسئله است. با توجه به اثربخشی هسته‌های غیرخطی در ساختار TSVM مشاهده می‌شود که استفاده از دو هسته غیرخطی در ساختار آن بهبود قابل‌توجهی نسبت به استفاده از یک هسته، ایجاد کرده است. همچنین استفاده از رویکرد OPTKTSVM به‌واسطه عملکردش که توسط الگوریتم ژنتیک، دو هسته بهینه برای داده‌ها ایجاد کرده، توانسته دقت را افزایش دهد.

جدول ٦: مقایسه دقت الگوریتم پیشنهادی با الگوریتم‌های دیگر بر روی دادگان UK-2007

UK-2007		
ردیف	الگوریتم	میزان دقت (٪)
۱	Genetic Algorithm[۲۶]	۹۲/۴۱
۲	Supervised Neural Network[۲۹]	۹۳/۶۶
۳	HTSVM	۹۴
۴	MKTSVM	۹۵/۶
۵	OPTKTSVM	۹۶/۴۲± ۱/۷۲

۶- نتیجه‌گیری و پژوهش‌های آینده

تاکنون الگوریتم‌های یادگیری ماشین زیادی در حوزه تشخیص صفحات فریب استفاده شده است که هر یک با توجه به توانایی ساختاری خود توانسته‌اند بهبودهایی را در حوزه تشخیص صفحات فریب ارائه دهند؛

اما یکی از نکاتی که در این مقاله نیز به آن اشاره کردیم الگوریتم‌های استفاده‌شده در این حوزه اغلب تکنیک‌های رایج یادگیری ماشین بوده که به‌صورت عمومی در سایر کاربردها نیز استفاده شده است. در بخش دو با برخی از تکنیک‌های استفاده‌شده در این حوزه آشنا شدیم.

در این مقاله با بهره‌گیری و ترکیب تکنیک‌های یادگیری ماشین سعی شده تا رویکردی ارائه شود تا بتواند به کاهش شکاف معنایی تشخیص صفحات فریب کمک کند. در رویکرد پیشنهادی با ترکیب الگوریتم ژنتیک و الگوریتم TSVM از مزیت‌های دو الگوریتم در این حوزه استفاده شده است.

در این مقاله سعی شد تا با طراحی هسته‌ای کارا و متناسب با داده‌های فریب قدرت تفکیک‌کنندگی طبقه‌بند را افزایش دهیم. برای این کار از الگوریتم ژنتیک استفاده شد. در این مسئله پارامترهای تشکیل‌دهنده هسته نهایی، استخراج گردید و توسط الگوریتم ژنتیک مقادیر بهینه آن‌ها بر مبنای طبقه‌بند TSVM محاسبه گردید.

مؤثر بودن این روش در تشخیص صفحات فریب بر روی دو دادگان مطرح در زمینه صفحات فریب مورد آزمایش قرار گرفت و نتایج آن در جدول ۴ و ۵ و ۶ بیان شده است.

پیشنهاد می‌شود در ادامه، پژوهش‌هایی در حوزه‌های زیر برای افزایش دقت در تشخیص صفحات فریب صورت پذیرد:

۱-در برخی مواقع مانند مرحله خزش در موتور جستجو احتیاج است که با صرف حداقل زمان ممکن بتوان برحسب ویژگی‌های استخراج‌شده از یک صفحه، فریب بودن آن را تشخیص داد. یکی از عوامل تأثیرگذار بر سرعت اجرای الگوریتم کاهش تعداد ویژگی‌ها است. برای این کار می‌توان با استفاده از تکنیک‌های کاهش بعد ابتدا تعداد ویژگی‌های موردبررسی را کاهش داد تا بتوان با تعداد کمی ویژگی، به‌سرعت اجرای مناسب دست پیدا کرد.

۲- می‌توان از روش‌هایی استفاده کرد که به‌صورت صفر و یکی عمل تشخیص صفحات را انجام ندهد. در این روش‌ها می‌توان احتمال فریب بودن یک صفحه را محاسبه کرد. از این مقدار درجاهایی که نیاز است موتور جستجو بتواند با ایجاد یک مقدار آستانه میزان حذف صفحات از منابع موتور جستجو را کاهش یا افزایش دهد می‌توان استفاده کرد.

۳- تلفیق روش‌های تشخیص صفحات فریب می‌تواند منجر به بهبود تشخیص آن‌ها شود. همان‌طور که در بخش یک اشاره شد روش‌هایی مانند روش‌های تحلیل الگوهای زبانی برای تشخیص در این حوزه به کار گرفته می‌شوند. استخراج ویژگی‌های مربوط به مشخصات زبانی صفحات و ترکیب آن‌ها با ویژگی‌های محتوایی و پیوندی میان صفحات می‌تواند به افزایش دقت در تشخیص صفحات فریب منجر شود.

مراجع

[1] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke, "Efficient and effective spam filtering and re-ranking for large web datasets," *Proceedings of the Information Retrieval Conference*, pp. 1-25, 2010.

[2] P. T. Metaxas, and J. DeStefano, "Web spam, propaganda and trust," *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web*, pp. 60-69, 2005.

[3] D. Fetterly, M. Manasse, and M. Najork, "Spam, damn spam and statistics," *Proceedings of the 7th International Workshop on the Web and Databases*, pp. 210-223, 2004.

[4] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, "Detecting spam web pages through content analysis," *Proceedings of the 15th International Conference on World Wide Web*, China, Beijin University, pp. 83-92, 2006.

[5] D. Zhou, J. Huang, and B. Schölkopf, "Learning from labeled and unlabeled data on a directed graph," *Proceedings of the 22nd International Conference on Machine Learning*, Brazil, Pugn University, pp. 1036-1043, 2007.

[6] L. Becchetti, C. Castillo, D. Donato, R. Baeza-Yates, and S. Leonardi, "Link analysis for web spam detection," *ACM Transactions on the Web (TWEB)*, vol. 2, no. 2, pp. 1-42, 2008.

[7] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri, "Know your neighbors: web spam detection using the web topology," *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 8-17, 2007.

[8] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates, "Web spam detection: link-based and content-based techniques," *The European Integrated Project Dynamically Evolving Large Scale Information Systems (DELIS): Proceedings of the Final Workshop*, Paderborn University, pp. 99-113, 2008.

[9] Y .Liu, R .Cen, M .Zhang, S .Ma, and L .Ru, "Identifying web spam with user behavior analysis," *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, pp. 9-16, 2009.

[10] B. Wu, and B. D. Davison, "Cloaking and redirection: A preliminary study," *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 7-16, 2005.

[11] K. Chellapilla, and A. Maykov, "Cross-Lingual web spam classification," *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web*, pp. 81-88, 2007.

[12] H .Najadat, and I. Hmeidi, "Web spam detection using machine learning in specific domain features*," Journal of Information Assurance and Security*, vol. 38, no. 4, pp. 2117-2123, 2009.

[13] A. Torabi, K. Taghipour, and S. Khadivi, "Web spam detection: new approach with hidden markov models," *Information Retrieval Technology*, vol. 13, no. 2, pp. 230-239, 2013.

[14] B. Tundalwar, R. Rashmi, and M. Kulkarni, "New classification method based on decision tree for web spam detection," *International Journal of Current Engineering and Technology*, vol. 8, no. 9, pp 929-940, 2014.

[15] A. A. Soni, and A. Mathur, "Content based web spam detection using naive bayes with different feature representation technique," *Journal of Engineering Research and Applications*, vol. 3, no. 5, pp. 198-205, 2013.

[16] M. Silva, M. Renato, T. A. Almeida, and A. Yamakami, "Artificial neural networks for content-based web spam detection," *Proceedings of the 14th International*

Conference on Artificial Intelligence (ICAI'12), pp. 1-7. 2012.

[17] T. Urvoy, T. Lavergne, and P. Filoche, "Tracking web spam with hidden style similarity," *Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 25-34, 2006.

[18] S. Bernhard, A. Smola, C. Williamson, and L. Bartlett, "New support vector algorithms," *Journal of Neural Computation*, vol. 4, no. 7, pp. 1207-1227, 2000.

[19] J. S. Taylor, and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Wiley Publishing, 2004.

[20] J. S. Taylor, and N. Cristianini, "Support vector machines and kernel method," *Journal of Artificial Intelligence Review*, vol. 12, no. 5, 2005.

[21] J. R. Khemchandani, "Twin support vector machines for pattern classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, 2007.

[22] D. E. Goldberg, E. David, and J. Holland. "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2, pp. 95-99, 1988.

[23] H. Castillo, D. Donato, L. Becchetti, P. Boldi, S. Leonardi, M. Santini, and S. Vigna, "A reference collection for web spam," *ACM Sigir Forum*, vol. 40, no. 2, pp. 11-24, 2006.

[24] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley Press, 2004.

[25] M. Tundalwar, R. Rashmi, and M. Kulkarni, "New classification method based on decision tree for web spam detection," *International Journal of Current Engineering and Eechnology*, vol. 4, no. 1, pp 112-117, 2014.

[26] M. Silva, M. Renato, T. A. Almeida, and A. Yamakami. "Artificial neural networks for content-based web spam detection," *Proceedings of the 14th International Conference on Artificial Intelligence (ICAI'12)*, pp. 1-7. 2012.

[27] A. Torabi, K. Taghipour, and S. Khadivi, "Web spam detection: new approach with hidden markov models," *Information Retrieval Technology*, vol. 3, no. 7, pp. 239-250, 2013.

[28] A. Keyhanipour, and B. Moshiri, "Designing a web spam classifier based on feature fusion in the layered multi-population genetic programming framework," *Proceedings of 16th International Conference on Information Fusion*, pp. 53-60, 2013.

[29] C .Ashish, M. Suaib ,and D. Beg, "Web spam classification using supervised artificial neural network algorithms," *Advanced Computational Intelligence: An International Journal*, vol. 2, no. 1, pp. 45-55, 2015.

زیرنویس‌ها

۱ Support Vector Machine (SVM)

۲ Twin Support Vector Machine (TSVM)

۳ Hyper-plane

۴ Fishing

۵ Hyper-plane

۶ Kernel

۷ Quadratic Programming Problem

۸ Margin

۹ Optimized Discriminant

۱۰ Map

۱۱ Crossover

۱۲ Mutation

۱۳ 10-Fold

۱۴ Central Limited Theorem

۱۵ Optimized Kernel TSVM (OPTKTSVM)