
بسم الله الرحمن الرحيم

دانشکده مهندسی کامپیوتر

پایان‌نامه کارشناسی ارشد مهندسی هوش مصنوعی

یادگیری کمی افزایشی در جریان داده

نگارنده: الهام احمدی

استاد راهنما:

دکتر هدی مشایخی

استاد مشاور:

دکتر مرضیه رحیمی

مهرماه ۱۴۰۰

شماره: ۸۱۱۴/ف/ب
تاریخ: ۱۵/ ۱۱/ ۱۴۰۰

باسمه‌تعالی

دانشگاه صنعتی شاهرود
مدیریت تحصیلات تکمیلی

فرم شماره (۳) صورتجلسه نهایی دفاع از پایان نامه دوره کارشناسی ارشد

با نام و یاد خداوند متعال، ارزیابی جلسه دفاع از پایان نامه کارشناسی ارشد خانم **الهام احمدی** با شماره دانشجویی **۹۶۰۱۹۱۴** رشته **مهندسی کامپیوتر** گرایش **هوش مصنوعی و رباتیکز** تحت عنوان **یادگیری کمی در جریان داده** که در تاریخ **۱۴۰۰/۷/۲۶** با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح ذیل اعلام می گردد:

الف) درجه عالی: نمره ۲۰-۱۹ ☐ ب) درجه خیلی خوب: نمره ۱۸/۹۹ – ۱۸ ☐
ج) درجه خوب: نمره۱۷/۹۹– ۱۶ ☐ د) درجه متوسط: نمره ۱۵/۹۹-۱۴ ☑
ه) کمتر از ۱۴ غیر قابل قبول و نیاز به دفاع مجدد دارد ☐
نوع تحقیق: **نظری** ☐ **عملی** ☐

عضو هیأت داوران	نام ونام خانوادگی	مرتبهٔ علمی	امضاء
۱ـ استادراهنمای اول	**دکتر هدی مشایخی**	**استادیار**	
۲- استادراهنمای دوم	----------	-------	-------
۳- استاد مشاور	**دکتر مرضیه رحیمی**	**استادیار**	
۴- نماینده تحصیلات تکمیلی	**مهندس محسن فرهادی**	**مربی**	
۵- استاد ممتحن اول	**دکتر حمید حسن پور**	**استاد**	
۶ـ استاد ممتحن دوم	**دکتر مریم خدابخش**	**استادیار**	

نام و نام خانوادگی رئیس دانشکده: **دکتر علیرضا الفی** تاریخ و امضاء و مهر دانشکده:

وزارت علوم، تحقیقات و فناوری
دانشکده فناوری اطلاعات و کامپیوتر

تقدیم اول به

مادر عزیزتر از جانم کسی که از خواسته هایش گذشت

و سختی ها را به جان خرید تا من به جایگاهی که اکنون در آن ایستاده ام برسم

و تقدیم دوم به

تمامی کسانی که در این مسیر راهنمای من بودند.

با سپاس از آنان که

ناتوان شدند تا ما به توانایی برسیم

و عاشقانه سوختند

تا گرما بخش وجود ما

و روشنگر راهمان باشند...

تعهدنامه

اینجانب **الهام احمدی** دانشجوی دوره کارشناسی ارشد رشته **مهندسی کامپیوتر** دانشکده **مهندسی کامپیوتر** دانشگاه صنعتی شاهرود نویسنده پایان‌نامه **یادگیری کمی افزایشی در جریان داده** تحت راهنمائی **دکتر هدی مشایخی** متعهد می‌شوم.

- تحقیقات در این پایان‌نامه توسط اینجانب انجام‌شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های محققان دیگر به مرجع مورداستفاده استناد شده است.
- مطالب مندرج در پایان‌نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می‌باشد و مقالات مستخرج با نام «دانشگاه صنعتی شاهرود» و یا «Shahrood University of Technology» به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان‌نامه تأثیرگذار بوده‌اند در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافت‌های آن‌ها) استفاده‌شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده‌شده است اصل رازداری، ضوابط و اصول اخلاق انسانی رعایت شده است.

تاریخ

امضای دانشجو

چکیده

در سال‌های اخیر، یادگیری در حوزه جریان داده توجه بسیاری از محققان و متخصصان را به خود جلب کرده است ولی با این‌حال، یادگیری کمی تا حد زیادی ناشناخته باقی‌مانده است. در برخی از برنامه‌های کاربردی که باید توزیع نظرات مثبت و منفی را به دست آوریم، استفاده از یادگیری کمی بسیار مفید واقع‌شده است. همچنین با کمک این روش می‌توان خصوصیات عمومی خاصی را در مورد جمعیت یک شبکه به دست آورد و با تجزیه‌وتحلیل احساسات افراد، اطلاعات کاربردی مفیدی را استخراج کرد. یادگیری کمی شباهت زیادی با طبقه‌بندی دارد و هردو کار گروه‌بندی داده‌ها را انجام می‌دهند ولی اهداف و کاربرد آن‌ها متفاوت است. در مسائل یادگیری کمی به دنبال تعیین کلاس نمونه‌ها نیستیم و تنها آمار کلی داده‌ها مدنظر است و هدف این است که تخمینی از توزیع داده‌ها را ارائه دهد. الگوریتم‌های اخیر در حوزه‌ی یادگیری کمی در جریان داده با کمک تغییر مفهوم و با استفاده از درخواست برچسب برای بخش بزرگی از نمونه‌های جدید واردشده و با روش‌های انتخاب نمونه ارائه‌شده است. در این تحقیق ایده این است که برچسب زیرمجموعه کوچک‌تری از نمونه‌های اخیر را درخواست کنیم و مدل رده‌بند را با کمک چند کلاسه‌بند متفاوت به‌صورت افزایشی تشکیل دهیم. آزمایش‌های ما نشان می‌دهد که با وجود کاهش درخواست برچسب از نمونه‌های اخیر و حتی حذف آن، می‌توان دقت مدل را حفظ کرده یا بهبود بخشید.

کلمات کلیدی: یادگیری کمی، جریان داده، رده‌بندی، یادگیری افزایشی، درخواست برچسب

لیست مقالات مستخرج از پایان‌نامه

۱-

۲-

۳-

فهرست

فهرست اشکال

فهرست جداول

فصل۱: مقدمه

۱-۱ مقدمه

عصر اطلاعات، شیوه ایجاد داده‌ها را متحول کرده است. در این زمینه، با حجم عظیمی از داده‌ها برای استخراج دانش مواجه هستیم که باید به‌طور موثر مدیریت و پردازش شوند و با تحلیل داده‌ها اطلاعات مفیدی را از آن بدست آورد [۱]. در دنیای پیشرفته امروز کسب اطلاعات مفید از وضعیت جامعه از اهمیت بالایی برخوردار است [۲]. علاوه بر این یادگیری از جریان داده[۱] که در آن داده‌ها به‌طور مداوم وارد می‌شوند [۳] و در هر لحظه متغیر هستند، بسیار موردتوجه قرارگرفته است و می‌تواند بسیار مفید واقع شود و اطلاعات خوبی را در اختیار ما قرار دهد [۴].

یادگیری کمی[۲] یکی از روش‌های داده‌کاوی[۳] است که برای برآورد توزیع کلاس در یک مجموعه آزمایشی پیشنهاد شده است. در این روش، هدف طبقه‌بندی تک‌تک نمونه‌ها نیست، بلکه توزیع هر کلاس، در داده‌ها اهمیت دارد. بطورکلی یادگیری کمی به پیش‌بینی کلاس هر داده علاقه‌ای ندارد، بلکه سعی می‌کند توزیع کلی کلاس را در مجموعه آزمون اندازه‌گیری کند [۵].

یادگیری کمی کاربردهای زیادی دارد؛ به‌عنوان مثال، تجزیه و تحلیل احساسات زمانی که می‌خواهیم نسبت نظرات مثبت در مورد یک محصول خاص را برآورد کنیم [۶]؛ تخمین درصد بیکاران در دوره‌های زمانی مختلف؛ شمارش تعداد نظرات مثبت، منفی یا خنثی در مورد یک موضوع یا محصول خاص و بسیاری موارد دیگر. به دست آوردن تخمینی از توزیع کلاس، برای بررسی استراتژی‌ها یا تعیین سیاست‌های مناسب، کافی است؛ به‌ویژه زمانی که با حجم داده بالا مواجه هستیم، رده‌بندی تک‌به‌تک داده‌ها مناسب نخواهد بود و هزینه بالایی برای ما خواهد داشت. در مسائل دنیای واقعی، زمانی که توزیع کلاس‌ها به‌طور قابل‌توجهی تغییرپذیر است، رده‌بندی عملکرد مناسبی از خود نشان نمی‌دهد و کامل نخواهد بود؛ ازاین‌رو استفاده از یادگیری کمی، می‌تواند بسیار موثر واقع شود.

---

[۱] Data stream

[۲] Quantification learning

[۳] Data Mining

یادگیری کمی و طبقه‌بندی همچنین وظایف مرتبط باهم دارند و از یکدیگر بهره‌مند می‌شوند. کمی‌سازی می‌تواند به‌طور مستقیم توسط شمارش تعداد نمونه‌هایی که در کلاس طبقه‌بندی شده‌اند، محاسبه شود. این روش به‌عنوان طبقه‌بندی و شمارش شناخته می‌شود. ارزیابی آزمایش‌ها، در [۷]، نشان می‌دهد که طبقه‌بندی و شمارش معمولا به نتایج تقریبا بهینه منجر می‌شود. ازجمله اطلاعات اضافی، مانند برآورد خطا و نمرات طبقه‌بندی، به نتایج دقیق‌تر منجر می‌شود [۸].

۱-۲ شرح مسئله

با توجه به [۹]، کمی‌سازی یک روش یادگیری با نظارت است که اخیرا برای مسائل مربوط به یادگیری ماشین موردتوجه قرارگرفته است. همان‌طور که قبلا اشاره کردیم، روش یادگیری کمی، به پیش‌بینی برچسب برای هر نمونه علاقه‌مند نیست، بلکه علاقه‌مند است تعداد کلی عناصر یک کلاس خاص را در یک مجموعه داده مشخص کند. درنتیجه، یک مدل یادگیری کمی خروجی را برای مجموعه‌ای از نمونه‌ها به‌جای هر نمونه می‌نویسد. خروجی شامل یک دنباله از مقادیر واقعی است که برآورد توزیع کلاس داده‌ها است.

کمی‌سازی به‌طور عمده بر تنظیم دسته‌ای تمرکز دارد. در چنین مواردی، به‌استثنای تغییر در نسبت کلاس‌ها، فرض می‌شود که توزیع داده‌ها ثابت‌اند. در سال‌های اخیر با توجه به این‌که جریان داده موردتوجه قرارگرفته و روش‌های خاصی برای کلاس‌بندی و خوشه‌بندی داده‌ها بکار گرفته‌شده اما مسئله‌ی کمی‌سازی هنوز بطورکامل جا نیفتاده است و به همین دلیل می‌تواند حائز اهمیت باشد. با مسئله کمی‌سازی در دنیای واقعی زیاد برخورد کردیم به‌عنوان مثال برآورد درصد بیکاران در دوره‌های مختلف، شمارش نظرات مثبت و منفی یا خنثی راجع به یک موضوع یا محصول خاص و شمار نوع خاصی از حشرات [۱۰] [۱۱].

تبدیل داده به دانش یکی از اقدامات مهم در عصر امروز است که با توجه به حجم وسیع اطلاعات از اهمیت ویژه‌تری برخوردار می‌شود. ما به یافتن الگوهای خاص در داده‌ها علاقه‌مند هستیم. یادگیری ماشین، زیرشاخه‌ای از هوش مصنوعی، نوعی تجزیه‌وتحلیل داده است که با ایجاد مدل‌های تحلیلی، فرایند یافتن و توصیف الگوهای داده را خودکار می‌کند. یادگیری ماشین بر این فرض استوار است که ماشین‌ها می‌توانند از داده‌ها درس بگیرند، الگوها را شناسایی کرده و با حداقل دخالت انسان پیش‌بینی کنند. این امر برای داده‌های در جریان اهمیت بیشتری دارد.

۱-۳ اهمیت انجام پژوهش

در سال‌های اخیر روش‌های مختلفی برای یادگیری کمی به‌کاررفته است که هرکدام با چالش‌هایی روبه‌رو هستند. یکی از چالش‌هایی که در این مسئله با آن مواجه هستیم هزینه‌ی بالایی است که برای درخواست برچسب داده‌ها صرف می‌شود که در این پایان‌نامه با کاهش درخواست برچسب، این هزینه را کاهش می‌دهیم.

چالش دیگری که با آن مواجه هستیم تاخیر تایید یا اعتبار می‌باشد که برابر است با تاخیر قبل از دریافت برچسب واقعی. این مقدار در اکثر روش‌ها صفر در نظر درگرفته شده اما در دنیای واقعی مقدار بزرگ‌تر از صفر و حتی گاهی مقدار بی‌نهایت دارد. در این تحقیق با توجه به کاهش درخواست برچسب، مجموع تاخیر اعتبارها نیز کاهش می‌یابد.

در مسائل واقعی، داده‌ها بطور مداوم در حال تولید و تغییر هستند و تغییر مفهوم چالشی است که با آن روبرو هستیم. ما سعی کردیم با پیاده‌سازی روش این پایان‌نامه، تاثیر تغییر مفهوم را کاهش دهیم.

۱-۴ هدف پژوهش

هدف اصلی ما در این پایان‌نامه، این است که با استفاده از روش یادگیری کمی افزایشی برای داده‌های در جریان، مدل مناسب با حداکثر دقت و درعین‌حال با هزینه درخواست برچسب کم‌تر تشکیل دهیم و تا حدودی به حل چالش‌های این مسئله کمک کنیم. با تشکیل مدل یادگیری کمی اولیه، مدل با داده‌های جدید، به‌صورت افزایشی ساخته می‌شود و درنتیجه با حداقل تغییر مفهوم مواجه هستیم و مدل، تمامی داده‌ها را در نظر خواهد گرفت.

در نظر داشته باشید که در مسائل دنیای واقعی، برچسب یک سری داده‌ها از قبل مشخص نیستند و نمی‌توان آن را درخواست کرد. ما در این تحقیق فرض را بر این گذاشتیم که برچسب بخش کوچکی از داده‌ها مشخص است و برای مدل یادگیری کمی از آن استفاده کرده‌ایم.

با توجه به تحقیقات اخیر صورت گرفته و اهمیت بالای یادگیری افزایشی در جریان داده که داده‌ها به صورت لحظه‌ای به سیستم وارد می‌شوند، ما توانستیم با استفاده از ترکیب یادگیری افزایشی با یادگیری کمی مدل جدیدی را ارائه دهیم. ارزیابی‌های ما نشان می‌دهند که دقت مدل نسبت به قبل تا حد نسبتا مناسبی افزایش داشته است.

برای ارزیابی مدل ارائه‌شده در این پایان‌نامه از چندین معیار ارزیابی استفاده‌شده است که روی پایگاه داده خاص آزمایش کرده‌ایم. نتایج حاصل از آن نشان می‌دهد که با وجود کاهش هزینه‌های درخواست برچسب و حذف آن دقت به‌دست‌آمده تا حدی حفظ‌شده و در یک سری از مشاهدات، این دقت افزایش داشته است.

۱-۵ مروری بر فصل‌های دیگر

در ادامه این پایان‌نامه بعد از معرفی روش‌های انجام‌شده در سال‌های اخیر که در فصل دوم آمده، در فصل سوم به شرح راهکار پیشنهادی می‌پردازیم. بررسی نتایج و بیان عملی پژوهش در فصل چهارم و درنهایت در فصل پنجم نیز نتیجه‌گیری به عمل می‌آوریم.

فصل دوم: پیشینه پژوهش

۲-۱ جریان داده

یک جریان داده یک دنباله مرتب‌شده از نمونه‌هاست. E یک جریان داده است جایی که e_t یک نمونه در فضای ویژگی P بعدی است. در مسائل یادگیری با ناظر، هر نمونه‌ی e_t یک برچسب مرتبط دارد؛ بنابراین، یک جریان داده در روش یادگیری با نظارت می‌تواند با یک زوج نمونه مرتب‌شده تعریف شود.

$$E = (\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_t, \ldots) \qquad \vec{e}_t \in R^p$$

(رابطه ۲-۱)

مسائل دنیای واقعی به‌طور فزاینده به‌عنوان جریان داده مدل می‌شوند. این واقعیت موجب توسعه روش‌های جدیدی شده است که از نیازهای تحمیل‌شده توسط فرآیند تولید اطلاعات مانند حجم بالا، سرعت و نوسان استفاده می‌کند. همچنین کاوش جریان داده با چالش‌هایی روبرو است که یکی از این چالش‌ها، شناسایی و واکنش به تغییرات در فضای ویژگی و به‌روزرسانی مدل طبقه‌بند برای ترکیب این تغییرات است [۴]. این تغییرات در داده‌ها، همچنین به‌عنوان تغییر مفهوم شناخته می‌شود.

به‌طورکلی، برای کاهش تاثیر تغییر مفهوم، دو استراتژی استفاده می‌شود: یکی این‌که مدل‌ها را در فواصل منظم تطبیق دهیم و دوم استفاده از یک مدل برای حضور تغییر مفهوم و سپس تطبیق مدل با نمونه‌های متعدد است [۱۲]. در الگوریتم ما با توجه به این‌که مدل به‌صورت افزایشی ساخته خواهد شد و به دلیل این‌که نمونه‌های اخیر در تشکیل مدل نقش مهمی را ایفا می‌کنند، تغییر مفهوم به حداقل میزان خود می‌رسد.

یکی دیگر از چالش‌های مهم در کاوش جریان داده، وقوع تأخیر تأیید است [۱۳]. در اکثر الگوریتم‌های طبقه‌بندی جریان داده، فرض می‌شود که برچسب درستی از یک نمونه به‌محض طبقه‌بندی در دسترس قرار می‌گیرد. بااین‌حال، در برنامه‌های دنیای واقعی، این فرض به‌ندرت وجود دارد. تاخیر تأیید، تاخیر قبل از دریافت برچسب

صحیح است؛ بنابراین، معمولا یک تاخیر در واحد زمان T تا زمانی که برچسب واقعی در دسترس باشد، وجود دارد. در بسیاری از روش‌ها این زمان برابر صفر بوده و به همین ترتیب، می‌توانند مدل را با استفاده از رویدادهای برچسب شده سریع‌تر به‌روز کنند. متاسفانه، در شرایط واقعی، این زمان مخالف صفر و حتی گاهی بی‌نهایت خواهد بود که کاوش جریان داده را بیشتر به چالش می‌کشد. تأخیر تأیید در کاوش جریان داده در تحقیقات کمی مورد بررسی قرارگرفته است و تنها در سال‌های گذشته برخی از روش‌ها پیشنهادشده است [۱۴]. تأخیر تأیید در اغلب برنامه‌های کاربردی کمی‌سازی رخ می‌دهد و اگر برچسب‌های صحیح، سریع یا بعد از زمان کمی شناخته شوند، بهتر است به‌جای استفاده از یک روش طبقه‌بندی، صبر کنید و برچسب‌های واقعی را حساب کنید. در این تحقیق فرض شده است که برچسب داده‌ها از ابتدا در دسترس است و این تاخیر را برابر صفر در نظر گرفته‌ایم.

۲-۲ یادگیری کمی

در یادگیری ماشین و داده‌کاوی، کمی‌سازی با کمک یادگیری با نظارت برای آموزش مدل استفاده می‌شود که فرکانس‌های نسبی را در ورودی‌های داده تخمین می‌زند [۱۵]. بعنوان مثال در نمونه ای از ۱۰۰۰۰۰ داده از بیان عقاید افراد در مورد یک نامزد سیاسی خاص آمده است، ممکن است از یک کمیت خاص برای تخمین درصد نظرات استفاده شود که مربوط به کلاس "مثبت" باشند. برای کلاس‌های "خنثی" و "منفی" نیز می‌توان این کار را انجام داد.

اگرچه یادگیری کمی و طبقه‌بندی شباهت‌ها را به اشتراک می‌گذارند ولی اهداف آن‌ها متفاوت است؛ بنابراین، کمی‌سازی نیاز به ارزیابی خاص، مخصوصا در الگوریتم‌های تخصصی یادگیری ماشین، دارد. در رابطه ۲-۲ مشاهده می‌کنید مجموعه برچسب E، جایی که هر رویداد e_t دارای یک برچسب y_t است و y_t مجموعه‌ی کلاس داده‌ها می‌باشد. ما می‌خواهیم یک کلاس‌بند δ را یاد بگیریم که با تابع $\delta: E \rightarrow C$ تعریف می‌شود و یک برچسب C_i را

برای هر رویداد $e_t \in \mathrm{E}$ مشخص می‌کند. فرکانس واقعی کلاس C_i در E توسط $\mathrm{freq}_E(c_i)$ تعریف می‌شود و هدف یک مدل یادگیری کمی، تخمین $\mathrm{freq}_E(c_l) \approx \mathrm{freq}_E(c_i)$ است. این عناصر برای تعریف مسئله کمی‌سازی کافی هستند.

$$E = (\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_m) \qquad \text{(رابطه ۲-۲)}$$
$$\widehat{freq_E}(c_l) \approx freq_E(c_i)$$

پس در مسئله یادگیری کمی، هدف ما یافتن بهترین برآورد توزیع کلاس داده است، یعنی به ازای هر کلاس، ما می‌خواهیم اختلاف بین فرکانس واقعی و تخمینی را کاهش دهیم. به لحاظ علمی، مسئله جریان داده با تغییر مفهوم و تاخیر تایید حل نمی‌شوند. در ادامه روش‌های اساسی برای یادگیری کمی را ارائه می‌دهیم.

۲-۳ روش‌های اساسی یادگیری کمی در جریان داده

۲-۳-۱ طبقه‌بندی و شمارش (CC)

همان‌طور که قبلا ذکر شد، روش ساده‌ای برای یادگیری کمی به‌عنوان طبقه‌بندی و شمارش شناخته‌شده است و شامل برچسب‌گذاری هر نمونه با طبقه‌بند و سپس شمارش نمونه‌ها در هر کلاس است. به‌عبارت‌دیگر، با چندین مرتبه شمارش برای هر برچسب یک خروجی طبقه‌بند δ برای هر نمونه در مجموعه آزمون یعنی T_e، توزیع کلاس برآورد می‌شود. δ با یک مجموعه آموزشی T_r ایجادشده است. این رویکرد را در رابطه ۲-۳ بیان‌شده است؛ که C_i کلاسی است که توسط طبقه‌بند تعیین‌شده است و pT_e کسری از موارد در T_e است که به طبقه‌بند C_i اختصاص داده‌شده است.

$$\hat{p}_{T_e}^{CC} = pT_e(\hat{c}_i) = \frac{|\{\vec{e}_t \in T_e | \delta(\vec{e}_t) = c_i\}|}{|T_e|} \qquad \text{(رابطه ۲-۳)}$$

در [۹]، نویسندگان استدلال می‌کنند که این استراتژی بهینه نیست، زیرا به دیگر بخش‌های اطلاعات مانند خطای حاشیه‌ای در میان کلاس‌ها توجه نمی‌کند و به تعداد مثبت و منفی کاذب اهمیت نمی‌دهد. برای مثال، این دو دسته‌بند را در نظر بگیرید: $h_۱$ با FP = ۲۰ و FN = ۲۰ و $h_۲$ با FP = ۱۸ و FN = ۲۰ (FN و FP تعداد منفی کاذب و مثبت کاذب می‌باشند). در این سناریو، طبقه‌بند $h_۲$ بهتر از $h_۱$ است. بااین‌حال، برای یادگیری کمی، $h_۱$ مدل باینری بهتری است که FP و FN مساوی دارد با توجه به این‌که خطاهای یکدیگر را حذف می‌کنند و نتیجه کمی‌سازی کامل می‌شود؛ درنتیجه کلاس‌بند نادرست می‌تواند یک مدل یادگیری کمی دقیق باشد، اگر خطاهای حاشیه‌ای آن به‌طور مساوی در FP و FN گسترش یابد.

۲-۳-۲ طبقه‌بندی و شمارش احتمالاتی (PCC)

در این روش $pT_e(C_i)$ به‌عنوان کسر مورد انتظار از موارد پیش‌بینی‌شده برای تعلق به کلاس C_i می‌باشد. فرض کنید p احتمال تعلق e_j به کلاس C_i است که توسط طبقه‌بند برآورد می‌شود و $E[e_i]$ مقدار مورد انتظار e_j تعریف‌شده که در رابطه ۲-۴ آمده است.

$$\hat{p}_{T_e}^{PCC}(c_i) = E[PT_e(\hat{c}_i)] = \frac{۱}{|T_e|}\sum_{\vec{e}_t \in T_e} P(c_i|\vec{e}_t) \qquad \text{(رابطه ۲-۴)}$$

این روش واریانس مدل اول یعنی CC است و به‌طورکلی عملکرد خوبی را نسبت به سایر روش‌ها از خود نشان داده است [۹].

۲-۳-۳ طبقه‌بندی و شمارش تعدیل‌شده (ACC)

این روش نسبت واقعی کلاس داده‌شده و اصلاح‌شده بر اساس نرخ صحیح مثبت و غلط مثبت (tpr و fpr) را به دست می‌آورد. چنین مقادیری با مجموعه اعتبارسنجی یا یک روش اعتبارسنجی در T_r تخمین زده می‌شود. رابطه ۲-۵ محاسبه ACC را برای کلاس c_i ارائه می‌دهد.

$$\hat{p}_{T_e}^{ACC}(c_i) = \frac{\hat{p}_{T_e}^{CC}(c_i) - fpr_{T_r}}{tpr_{T_r} - fpr_{T_r}} \quad \text{(رابطه ۲-۵)}$$

روش ACC برای زمانی که توزیع داده ثابت نیست، سخت می‌شود. علاوه بر این، یک مسئله جزئی این است که این روش می‌تواند نتایج منفی یا نتایج بالاتر از ۱ را نیز تولید کند [۹].

یادگیری کمی همچنین مستلزم معیارهای ارزیابی خاصی است که در ادامه بحث به آن اشاره خواهیم کرد.

۲-۴ معیارهای ارزیابی پرکاربرد در یادگیری کمی

۲-۴-۱ خطای مطلق (AE)

معیارهای مختلف می‌تواند دقت مدل یادگیری کمی را مورد ارزیابی قرار دهد. اکثر آن‌ها بر اساس معیارهای خطا و آنتروپی است. رابطه ۲-۶ خطای مطلق (AE) را تعریف می‌کند. این رابطه مربوط به میانگین مطلق اختلاف بین توزیع کلاس $p(c_i)$ و توزیع کلاس واقعی است.

$$AE(p,\hat{p}) = \frac{۱}{|C|}\sum_{c_i \in C} |\hat{p}(c_i) - p(c_i)|$$ (رابطه ۲-۶)

۲-۴-۲ خطای مطلق نرمال شده (NAE)

در رابطه قبل، AE محدوده بین ۰ (بهترین) و ۱ (بدترین) دارد. رابطه ۲-۷ خطای مطلق نرمال شده (NAE)، نسخه نرمال شده بین ۰ و ۱ را برای خطای مطلق ارائه می‌دهد.

$$NAE(p,\hat{p}) = \frac{\sum_{c_i \in C} |\hat{p}(c_i) - p(c_i)|}{۲(۱ - \min_{c_i \in C} p(c_i))}$$ (رابطه ۲-۷)

۲-۴-۳ خطای نسبی نرمال (NRAE)

بااین‌حال، با توجه به [۷] AE و NAE زمانی که شیوع کلاس واقعی کوچک است از یک مسئله جدی رنج می‌برند. به‌عنوان مثال، اگر در هنگام پیش‌بینی $\hat{p}(c_i)$ = ۰,۱ و $p(c_i)$ = ۰,۰۱، یا در مورد دیگر $\hat{p}(c_i)$ = ۰,۵۰ و $p(c_i)$ = ۰,۴۱ با توجه به این‌که اختلاف این دو در هر دو مدل یکی است، طبق رابطه قبل خطاهای مشابهی تولید می‌شوند. برای جلوگیری از این موارد، نرخ خطای نسبی نرمال (NRAE) پیشنهاد شد.

$$NRAE(p,\hat{p}) = \frac{\sum_{c_i \in C} \frac{|\hat{p}(c_i) - p(c_i)|}{p(c_i)}}{|C| - ۱ + \frac{۱ - \min_{c_i \in C} p(c_i)}{\min_{c_i \in C} p(c_i)}}$$ (رابطه ۲-۸)

طبق رابطه فوق NRAE برای صفر تعریف‌نشده است؛ بنابراین با کمک لاپلاس رابطه ۲-۹ به دست می‌آید:

$$p_s(c) = \frac{\epsilon + p(c)}{\epsilon|C| + \sum_{c_i \in C} p(c_i)} \quad \text{(رابطه ۲-۹)}$$

$p_s(c)$ نسخه نرمال $p(c)$ و $\epsilon = \frac{1}{2|E|}$ فاکتور هموار[1] است.

۲-۵ انتخاب نمونه و یادگیری نیمه نظارتی

مالتزکه و همکاران در [۸] الگوریتم SQSI را ارائه کردند. روش کار این الگوریتم به این صورت هست که جریان داده را به تکه‌های با یک پنجره کشویی بدون هم‌پوشانی تقسیم می‌کند و هر قطعه را به ترتیب پردازش می‌کند. سپس به‌صورت نظارت‌شده برچسب واقعی داده‌ی موجود در استخر را درخواست می‌کند و مدل را تشکیل می‌دهد. هر زمان که SQSI یک تغییر مفهوم را گزارش کند، یک طبقه‌بند جدید را ایجاد می‌کند که با داده‌های موجود در استخر آموزش داده می‌شود و برچسب‌های درست برای تمام نمونه‌های آن را درخواست می‌کند. اگرچه SQSI تنها در صورت بروز تغییر مفهوم، هزینه درخواست برچسب را متحمل می‌شود، اما برچسب‌ها اغلب یک منبع محدود و گران‌قیمت هستند و این روش بهینه نخواهد بود. در ادامه تحقیقات خود در [۱۶] روش SQSI-IS یادگیری نیمه نظارتی را دنبال کردند. در این روش ابتدا یک نمونه انتخاب می‌کنیم و بجای درخواست برچسب کلاس برای کل نمونه‌های استخر، برچسب یک زیرمجموعه از آن را درخواست می‌کنیم و باقی نمونه‌های استخر را به کمک روش خودآموزی برچسب می‌دهیم و هزینه درخواست برچسب تا حدی کاهش خواهد داشت.

[1] smoothing factor

ازاین‌رو استفاده از روش انتخاب نمونه برای پیدا کردن یک زیرمجموعه از نمونه‌های مربوطه در استخر و درخواست برچسب‌های درست فقط برای آن‌ها حائز اهمیت است. روش‌های انتخاب نمونه، طبق گفته [۱۷]، سه روش اساسی وجود دارد:

- ترکیب پرس‌وجو برای عضویت: مجموعه‌ای از نمونه‌های مصنوعی را ایجاد و برچسب‌گذاری می‌کند.
- مبتنی بر استخر: برچسب‌های واقعی زیرمجموعه‌ای از نمونه‌های فاقد برچسب را درخواست می‌کند.
- نمونه‌گیری انتخابی مبتنی بر جریان: موارد مربوط به یک جریان است و تصمیم برای برچسب زدن نمونه‌ها به‌صورت آنلاین اتفاق می‌افتد.

با توجه به [۱۸] روش‌هایی که نمونه‌هایی را بر اساس اقدامات ارزیابی انتخاب می‌کنند در ادبیات گسترده است. این روش‌ها از روش‌هایی مانند عدم اطمینان، کاهش خطای مورد انتظار و پرس‌وجو بر اساس گروه برای برآورد بهره هر نمونه استفاده می‌کنند.

نمونه‌برداری عدم اطمینان مواردی را انتخاب می‌کند که طبقه‌بندی حداقل در مورد برچسب‌های کلاس خود اطمینان دارد. کاهش خطای مورد انتظار نمونه‌هایی را انتخاب می‌کند که به کاهش خطای مدل کمک می‌کند و پرس‌وجو بر اساس گروه از یک سری داده برای تصمیم‌گیری نمونه به‌عنوان برچسب استفاده می‌کند.

بااین‌حال، با توجه به سناریو غیرثابت که در آن تغییرات می‌تواند در توزیع کلاس و همچنین در فضای ویژگی رخ دهد، کاربرد روش‌های انتخاب نمونه بر اساس اقدامات ارزیابی ممکن است گمراه‌کننده باشد. به‌عنوان مثال، روش نمونه‌گیری عدم قطعیت مواردی را انتخاب می‌کند که در رده‌بندی آن حداقل اعتمادبه‌نفس را دارد؛ بنابراین، این روش غیرمستقیم فرض می‌کند که داده‌ها ثابت هستند. در ادامه روش‌های انتخاب نمونه را ارزیابی می‌کنیم.

۲-۵-۱ نمونه‌گیری تصادفی

نمونه‌گیری تصادفی شامل انتخاب یک زیرمجموعه از موارد به‌صورت تصادفی و با احتمال برابر می‌باشد [۱۹]. ازآنجایی‌که همه نمونه‌ها قابل اندازه‌گیری هستند، درنتیجه هیچ مکانیسم هوشمندانه‌ای تصمیم نمی‌گیرد که کدام‌یک بیشتر ارزشمند باشد؛ لذا این روش در عین سادگی نمی‌تواند نتایج مناسبی را برای ما به دنبال داشته باشد.

۲-۵-۲ دورترین در اولین گذر

این روش بدین‌صورت است که مجموعه‌ای از برچسب‌های نماینده از مجموعه‌ای از رویدادهای مختلف تشکیل می‌شود. برای رسیدن به چنین تنوعی، باید اولین و بزرگ‌ترین انتخاب k نمونه‌هایی که از یکدیگر دورترین هستند را انتخاب کنیم [۱۰].

۲-۵-۳ نمونه‌گیری مبتنی بر خوشه

تجزیه و تحلیل خوشه‌ای در زمینه‌های مختلف برای شناسایی گروه‌ها و مشاهده رابطه بین گروه‌ها به روش بدون نظارت استفاده‌شده است. روش‌های خوشه‌ای زیادی وجود دارد، اما در اصل، اکثر آن‌ها از ایده شباهت استفاده می‌کنند؛ بنابراین، هدف خوشه‌بندی شامل حداکثر سازی شباهت بین موارد در یک گروه و به حداقل رساندن شباهت بین نمونه‌هایی از گروه‌های مختلف است؛ بنابراین، مجموعه‌ای از حوادث نمایش داده‌شده توسط یک ساختار خوشه‌ای می‌تواند مفید باشد.

رویدادهای نمایشی انتخاب نمونه بر اساس الگوریتم‌های خوشه‌ای رویدادهای نزدیک به مراکز خوشه‌ای و نزدیک به مرزهای خوشه را انتخاب می‌کند تا مجموعه‌ای متنوعی از حوادثی را که برچسب‌های آن‌ها درخواست شده است، ایجاد کند. پیدا کردن خوشه‌ها در یک مجموعه‌ای از رویدادها معمولا شامل یک پارامتر k است که نشان‌دهنده تعداد خوشه‌ها است و گاهی اوقات این پارامتر می‌تواند بر اساس تعداد رویدادهایی که برچسب‌گذاری

می‌شوند تنظیم شود. استراتژی دیگری شامل تعریف تعدادی از حوادث است که برچسب‌های واقعی درخواست می‌شوند و تعداد رویدادهای مرزی از هر خوشه‌ای است. ازاین‌رو تعداد خوشه‌ها به‌وسیله $k = b/(b_e + ۱)$ به دست می‌آید.

ما به دنبال این هستیم که تعداد کمتری از برچسب‌ها را درخواست کنیم. بااین‌حال، مجموعه‌هایی با تعداد کمی از برچسب‌ها ممکن است منجر به مدل نامناسب شود، چرا که الگوریتم یادگیری، نمونه‌های کافی برای یادگیری درست ندارد و باید بدون توجه به آن به کار گرفته شود. در هر صورت، برای بهینه‌سازی استفاده از چنین مجموعه‌های برچسب دار کوچک، از روش نیمه نظارتی استفاده می‌کنیم تا برچسب‌های نمونه‌های باقیمانده در استخر را به دست آوریم.

روش یادگیری نیمه نظارت (SSL) برای یادگیری مفید است و در آن مقدار نمونه‌های برچسب شده بسیار محدود است که توانایی تعمیم الگوریتم یادگیری را به خطر می‌اندازد. زو و گلدبرگ در [۲۰] یک بررسی جامع از روش‌های یادگیری نیمه نظارتی را ارائه می‌دهند و در [۱۶]، یکی از ساده‌ترین روش‌های SSL در دسترس است که در ادامه الگوریتم آن را بیان می‌کنیم.

خودآموزی یک الگوریتم پیچیده است که تکرار یک روش یادگیری تحت نظارت را اعمال می‌کند. این الگوریتم یک مدل طبقه‌بندی اولیه با استفاده از بخش برچسب شده داده‌ها را ایجاد می‌کند و از این مدل برای طبقه‌بندی نمونه‌های بدون برچسب استفاده می‌کند. در هر تکرار، بخشی از موارد طبقه‌بندی‌شده به مجموعه برچسب منتقل می‌شود. رویکرد آن انتخاب نمونه‌هایی است که با بیشترین اعتماد به‌حساب می‌آیند. الگوریتم این نمونه‌ها را با برچسب پیش‌بینی‌شده تگ می‌کند. در این پیاده‌سازی یک نمونه را در هر تکرار با بالاترین نمره انتخاب می‌کنیم. الگوریتم ادامه می‌یابد تا همه نمونه‌ها را انتخاب کنیم. شکل ۲-۱ جزئیات استراتژی خودآموزی را نشان می‌دهد [۱۶].

Algorithm 1: Self-training

Input: Labeled instances L_s; Unlabeled instances U_s;
Supervised machine learning method ml_{alg}
Output: $\delta(U_s)$

```
1 begin
2     repeat
3         δ ← buildClassifier(L_s, ml_alg)
4         S ← SelectMostConfident(δ, U_s)
5         U_s ← U_s − S
6         L_s ← L_s ∪ S
7     until U_s = ∅;
8     return δ(U_s)
9 end
```

شکل ۲-۱ : الگوریتم خودآموزی

با توجه به [۲۰]، خودآموزی فرض می‌کند که پیش‌بینی‌های آن، زمانی که با اعتماد بالا انجام می‌شود، درست عمل می‌کنند. این فرض یک نقطه بحث‌برانگیز است که خطاهای اولیه ساخته‌شده توسط طبقه‌بندی δ می‌تواند به مجموعه‌ای بدون برچسب منتقل شود. به همین دلیل در [۱۶] اهمیت روش انتخاب نمونه تقویت شد تا نمونه‌هایی انتخاب شود که بتوان به‌صورت خارجی برچسب‌گذاری کرد.

۲-۶ روش‌های مرجع برای اندازه‌گیری جریان داده‌ها

اکنون دو رویکرد مرجع برای اندازه‌گیری جریان داده‌ها را ارائه می‌کنیم. اگرچه این رویکردها، فرضیه‌هایی دارند که در عمل نشدنی هستند؛ نظیر عدم در دسترس بودن لحظه‌ای همه برچسب‌ها در طول جریان؛ ولی از آن‌ها به‌عنوان روش‌های مرجع استفاده می‌کنند.

۲-۶-۱ روش استاتیک

این الگوریتم ناسازگار بودن در محیط‌های جریان داده را نادیده می‌گیرد. این روش، یک مدل را با نمونه‌های اولیه از جریان ایجاد می‌کند و در طول زمان آن را به‌روز نمی‌کند. این رویکرد مقدار هر یک از وقایع را تولید می‌کند؛ بنابراین، لازم است که بخش کوچکی از اطلاعات برچسب شده را از ابتدای جریان داشته باشیم. این نیز بسیار کارآمد است زیرا یک مدل را یک‌باره ایجاد می‌کند. ما آن را به‌عنوان پایه در نظر می‌گیریم.

۲-۶-۲ روش کشویی

در این رویکرد، ما به‌طور مرتب مدل را در هر رویداد به‌روز می‌کنیم و تلاش می‌کنیم تا آخرین تغییرات در جریان را پیگیری کنیم. پس از پیش‌بینی برچسب، برچسب‌های واقعی خود را در دسترس قرار می‌دهند که اجازه می‌دهد مقدار سنج به‌روز شود. ما این رویکرد را در نظر می‌گیریم.

در ادامه الگوریتم SQSI و نسخه پیشرفته‌تر آن را که بالا اشاره کردیم به تفضیل بیان می‌کنیم.

۲-۷ الگوریتم SQSI اصلی و پیشرفته

روش SQSI که قبلا به آن اشاره کردیم، به شرح زیر عمل می‌کند. در مرحله اول، یک مدل را از یک مجموعه آموزش برچسب‌گذاری یاد می‌گیرد. پس‌ازاین مقدار اولیه، روش مقدار سنجی را هر زمان که یک استخر به وقایع جدید دست می‌یابد، محاسبه کند. برای این منظور، SQSI نمرات طبقه‌بندی را برای هر رویداد در استخر با استفاده مدل تولید می‌کند. سپس، بررسی می‌کند که آیا این نمرات و برآوردهای موجود در مجموعه آموزشی (با اعتبار متقابل) از یک توزیع مشابه (با استفاده از یک آزمون آماری) حاصل می‌شود، یا نه. اگر فرضیه صفر رد نشده باشد (به‌عنوان مثال، هر دو نمونه از یک توزیع مشابه می‌آیند)، مدل را اعمال می‌کند و نتیجه صادر می‌شود.

بااین‌حال، اگر فرضیه صفر رد شود، در استخر اخیر از مدل به‌دست‌آمده مجددا استفاده می‌کنیم؛ بنابراین، SQSI در ابتدا تلاش می‌کند تا مقادیر ویژگی را در هر نمونه از مجموعه اخیر با میانگین و انحراف معیار مجموعه مرجع مقایسه کند [۸].

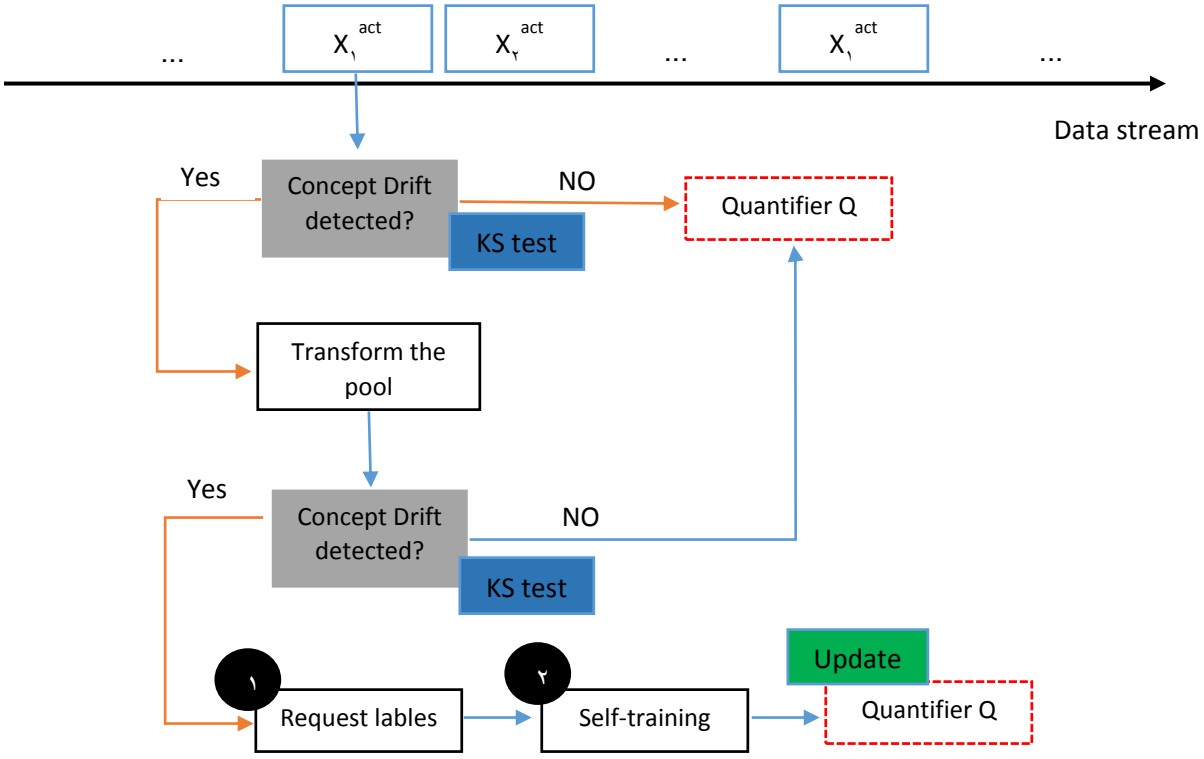
پس از اعمال مدل روی داده‌های استخر، SQSI نمرات جدیدی برای داده‌های جدید تولید می‌کند. پس‌ازآن آزمون آماری روی نمرات جدید اعمال می‌شود و اگر فرضیه صفر رد شود، الگوریتم، درخواست برچسب را برای داده‌های موجود در استخر ارسال می‌کند و طبقه‌بند δ به‌روزرسانی خواهد شد. در غیر این صورت، کلاس‌بندی با مدل قبلی انجام می‌شود و نتیجه صادر می‌شود. آزمون آماری مورداستفاده در این الگوریتم، آزمون کلموگروف اسمیرنف[۱] با سطح معنی‌داری ۰,۰۰۱ می‌باشد. این مقدار اندک برای به حداقل رساندن تعداد مثبت غلط به علت استفاده مجدد از آزمون تکراری لازم است.

در نسخه اصلی الگوریتم SQSI بسیاری از برچسب‌ها را برای یادگیری در طول جریان درخواست می‌کند. در موارد خاص، می‌تواند برچسب‌ها را برای هر نمونه‌ای درخواست کند. در [۱۶] یک مکانیسم برای درخواست تعداد کمتر از برچسب‌ها در هنگام تغییر ارائه‌شده است که با استفاده از یک روش انتخاب نمونه با آموزش نیمه نظارتی می‌تواند نتایج مشابهی را با برچسب کوچک‌تر ارائه دهد.

شکل ۲-۲ نشان‌دهنده الگوریتم SQSI با انتخاب نمونه (SQSI-IS) می‌باشد. به‌طور مشابه با SQSI، نیاز به یک گام اولیه دارد که یک مقدار سنج را با اولین تکه‌ای از جریان (مجموعه آموزش) ایجاد می‌کند؛ بنابراین، به‌عنوان نتیجه، ما اولین مدل را برای یادگیری داریم. بااین‌حال، متفاوت از SQSI، این الگوریتم یک روش انتخاب نمونه برای کاهش میزان برچسب‌های واقعی درخواست شده را تعریف می‌کند. به‌طور خلاصه، تفاوت اصلی بین SQSI و SQSI-IS شامل استفاده از انتخاب نمونه و خودآموزی است.

[۱] Kolmogorov- Smirnov

در روش SQSI-IS بخشی از برچسب‌ها را با استفاده از روش انتخاب نمونه‌ای درخواست می‌کند و الگوریتم خودآموزی، نمونه‌های باقی‌مانده را برچسب می‌کند. در الگوریتم SQSI-IS، نمونه‌های آزمایش‌شده مورداستفاده برای آموزش با استفاده از روش انتخاب نمونه، انتخاب می‌شوند. جزئیات تکمیل و مجموعه داده‌ها به‌صورت آزاد به‌عنوان مواد تکمیلی در دسترس هستند [۱۶].



شکل ۲-۲: الگوریتم SQSI پیشرفته

با توجه به [۸] یک سری محدودیت‌ها، برای این روش SQSI اصلی در نظر گرفته‌شده است که در روش پیشرفته آن که در [۱۶] آمده است این محدودیت‌ها تا حدی برطرف شده‌اند. این محدودیت‌ها را در ادامه متن بیان می‌کنیم:

اولین محدودیت این است که این روش فقط به مسائل یادگیری کمی باینری مربوط می‌شود و باید توجه داشته باشیم که اگرچه این محدودیت در این روش وجود دارد، اما در روش SQSI-IS به‌طور خاص به مسئله یادگیری باینری تکیه ندارد و روش‌های یادگیری کمی دقیق‌ترین و بهترین عملکرد برای این تنظیم به‌طور انحصاری از خود بروز می‌دهند.

محدودیت دوم این است که با تجزیه و تحلیل دسته‌های متوالی داده‌ها، آزمایش‌ها ساده‌تر شده‌اند، نمونه‌های اخیر یک عدد کوچک در مقایسه با حجم کل جریان داده است و این مسئله نباید عملکرد روش یادگیری را تغییر دهد. بااین‌حال، این محدودیت را می‌توان با استفاده از روش اسمیرنوف کولموگروف افزایشی[1] حذف کرد.

محدودیت سوم این است که برای هر تغییر مفهوم کشف‌شده که نمی‌تواند به‌طور موفقیت‌آمیز به‌عنوان یک تحول خطی از داده‌ها شناخته شود، هنوز باید برچسب‌های واقعی را جمع‌آوری کنیم و مدل را مجددا بسازیم. روش پیشرفته در جهت کاهش تعداد درخواست برچسب‌های واقعی موفق بوده است، اما می‌توانیم در یک سری شرایط خاص، هر نوع برچسب واقعی را درخواست بکنیم. به‌عنوان مثال، ما مطمئنا می‌دانیم که در برخی از مجموعه داده‌های آزمایشی، به‌عنوان مثال حروف زبان عربی، مفاهیم تکرار می‌شوند؛ درنتیجه به‌منظور جلوگیری از درخواست برچسب، یادگیری مفاهیم گذشته می‌توانند بسیار موثر واقع شوند.

چهارمین نکته قابل‌توجه این است که همان‌طور که آزمون کلموگروف-اسمیرنوف را با نمرات تولیدشده توسط طبقه‌بندی تغذیه می‌کنیم، یعنی اعتمادبه‌نفس آن بر هر نمونه مثبت، مثبت است، آزمون نسبت به تغییرات کلاس‌ها حساس است. ازآنجاکه تغییر در توزیع‌های قبلی می‌تواند طبقه‌بندی‌ها را به‌طور منفی تحت تأثیر قرار دهد، این نوع تغییر بدون تغییر در فضای ویژگی، نباید بر عملکرد یادگیری کمی تاثیر بگذارد؛ بنابراین، افزایش پرچم برای تغییر مفهوم و درخواست یک مدل به‌روز شده برای اهداف ارزیابی ضروری نخواهد بود.

[1] Incremental Kolmogorov-Smirnov

۲-۸ تاثیر اندازه مجموعه آزمون در یادگیری کمی

با توجه به [۱۶] یک متغیر مهم، یعنی اندازه مجموعه آزمون، نادیده گرفته‌شده است. این بی‌توجهی به‌اندازه مجموعه‌های آزمون، سه اثر مخرب عمده دارد:

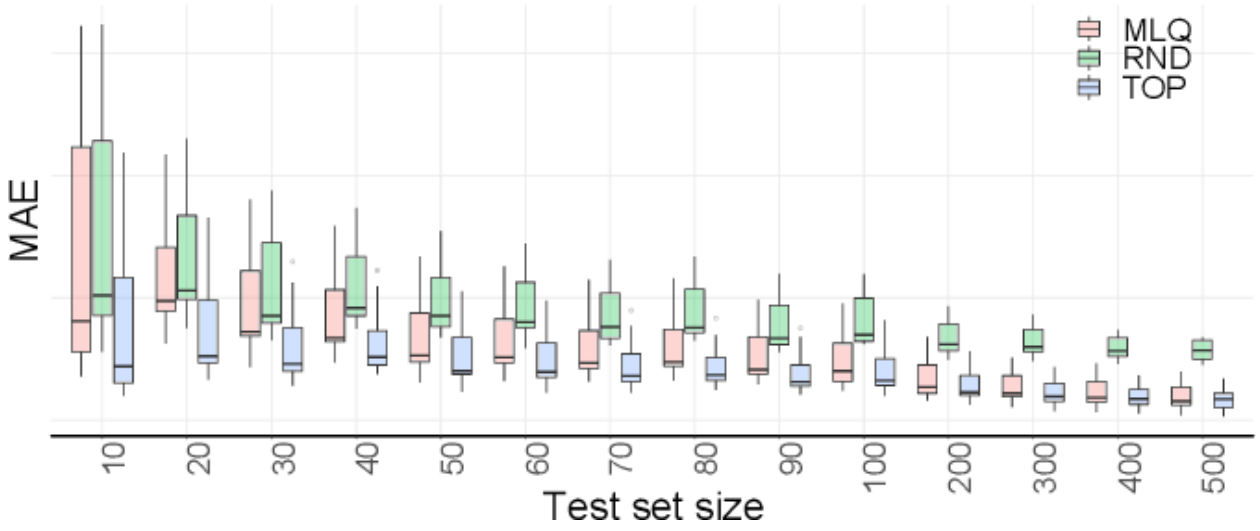
اول این‌که به‌طور ضمنی فرض می‌شود، دستگاه‌های اندازه‌گیری برای اندازه‌های مختلف مجموعه آزمایش به همان اندازه خوب عمل خواهند کرد درحالی‌که این‌گونه نیست. دوم، با انتخاب اندازه مجموعه آزمایشی، خطر برداشت گیلاس[۱] افزایش می‌یابد. درنهایت، اهمیت طراحی روش‌های مناسب، برای اندازه‌های مختلف مجموعه آزمایش نادیده می‌گیرد.

برای حل این مسئله، مالتزکه و همکاران مجددا در [۲۱] یک طرح فرا آموزشی برای انتخاب بهترین مدل یادگیری کمی بر اساس اندازه آزمون پیشنهاد کردند. آن‌ها مجموعه‌ای از آزمایش‌ها را با چندین الگوریتم کمی شناخته‌شده توصیف کردند، با این هدف که تأثیر اندازه‌های مختلف مجموعه آزمایش را بر عملکرد روش کمی نشان دهند.
در اکثر روش‌ها، تأثیر اندازه مجموعه آزمایش نادیده گرفته می‌شود. در روش [۲۱] آمده است که این موضوع، یک نقص جدی است زیرا عملکرد کمی‌سازها با توجه به این متغیر در نوسان است؛ به همین جهت یک گام اضافی درروش ارزیابی استاندارد وارد شد تا بتوانیم عملکرد را در اندازه‌های آزمایشی متمایز، علاوه بر توزیع‌های مختلف کلاس، ارزیابی کنیم.

بهترین الگوریتم کمی برای مجموعه‌های بزرگ ممکن است بهترین روش برای مجموعه‌های کوچک نباشد و برعکس. در روش MLQ[۲] ارزیابی می‌شود که کدام الگوریتم باید برای دستیابی به بهترین نتایج ممکن برای هر کار به صورت جداگانه، با توجه به ویژگی‌های آن استفاده شود. ابتدا، چندین مجموعه داده را جمع‌آوری کرده و بطور یکنواخت هر مجموعه داده را به دونیم تقسیم می‌کنیم (نیمه‌های آموزشی و آزمایشی). با بخش داده آموزش،

[۱] cherry-picking
[۲] Meta-Learning Quantification

ده مرتبه اعتبار سنجی انجام می‌گیرد تا بهترین دقت بدست آید.در شکل ۲-۳ مقایسه عملکرد بین این روش با خط پایه[۱] و خط بالا[۲] آمده است.



شکل ۲-۳: میانگین خطای کمی با روش‌های MLQ، TOP و RND با مجموعه تست متغیر از ۱۰ تا ۵۰۰

طبق این شکل با افزایش مجموعه آزمایش دقت مدل افزایش می‌یابد. در رویکرد خط پایه، مدل با کمک اولین نمونه‌ها ساخته می‌شود و متغیر بودن محیط‌های جریان داده نادیده گرفته می‌شود و در روش خط بالا بطور مرتب رویدادهای اخیر و آخرین تغییرات جریان داده ارزیابی می‌شود.

[۱] Baseline(RND)

[۲] topline

خلاصه پژوهش‌های این بخش را می‌توانید در جدول ۲-۱ مشاهده کنید.

جدول ۲-۱. خلاصه‌ای از پژوهش‌های انجام‌شده

شماره	نویسندگان	روش	شرح	معایب
۱	مالتزکه و همکاران (۲۰۱۷) [۸]	SQSI	روش یادگیری با نظارت ترکیب انتخاب نمونه و خودآموزی	درخواست برچسب برای کل داده‌های موجود در استخر در زمان تغییر مفهوم
۲	مالتزکه و همکاران (۲۰۱۸) [۱۶]	SQSI-IS	روش یادگیری با نظارت ترکیب انتخاب نمونه و خودآموزی همراه با انتخاب نمونه	درخواست برچسب برای یک سوم داده‌های موجود در استخر در زمان تغییر مفهوم و پیچیدگی زمانی بالا و مشکل اندازه آزمون
۳	مالتزکه و همکاران (۲۰۲۰) [۲۱]	MLQ	مجموعه‌ای از روش‌های کمی	پیچیدگی زمانی بالا

فصل سوم: روش پیشنهادی

در مسائلی که با داده‌های واقعی روبه‌رو هستیم، با چندین چالش مواجه می‌شویم؛ یکی از این چالش‌ها این است که اطلاعات ممکن است در طول زمان تغییر کند. دلیل این امر این است که کاربران رفتار خود را تغییر می‌دهند. به عنوان مثال با توجه به تحقیق انجام شده در [۲۲] حدود یک هفته طول کشید تا ده عبارت جستجوی آمازون در چندین کشور به مضامین مرتبط با کووید۱۹[۱] تغییر کرد.

از طرفی داده‌ها به‌طور مداوم توسط کاربران ساخته می‌شود. برای حل این چالش‌ها در برنامه‌های کاربردی از جریان داده استفاده می‌کنیم و فرض می‌کنیم که این داده‌ها تا بی‌نهایت ادامه خواهند داشت و داده‌ها را به‌طور آنلاین نگه‌داری می‌کنیم. درعین‌حال برای کارآمد بودن برنامه‌های کاربردی نیازمند این هستیم که داده‌ها را به‌صورت هم‌زمان در اختیار داشته باشیم و تغییرات داده‌ها را تشخیص دهیم و با آن‌ها سازگار شویم. پس وقتی در مورد یادگیری از جریان داده صحبت می‌کنیم به دلیل محدود بودن حافظه از بخش محدودی از داده‌ها استفاده می‌شود.

در ادامه به بیان یک سری مفاهیمی می‌پردازیم که درروش پیشنهادی این تحقیق به‌کاررفته است و سپس توضیح کاملی از روش معرفی‌شده ارائه خواهیم داد.

۳-۱ یادگیری افزایشی[۲]

در علوم کامپیوتر، یادگیری افزایشی روشی برای یادگیری ماشین است که در آن داده‌های ورودی به‌طور مداوم برای گسترش دانش مدل موجود و آموزش بیشتر مدل استفاده می‌شود. این نشان‌دهنده یک روش پویا از یادگیری تحت نظارت و یادگیری بدون نظارت است که می‌تواند هنگامی‌که داده‌های آموزش به‌تدریج با گذشت زمان در

[۱] covid-۱۹

[۲] incremental learning

دسترس قرار می‌گیرند یا اندازه آن‌ها از حد حافظه سیستم خارج می‌شود، استفاده شود. الگوریتم‌هایی که می‌توانند یادگیری افزایشی را تسهیل کنند به‌عنوان الگوریتم‌های یادگیری افزایشی شناخته می‌شوند.

هر سیستم یادگیری باید با محیط در حال تغییر، سازگار باشد. در بیشتر سناریوهای عملی، داده‌های آموزشی از بین می‌روند و دستگاه‌ها نمی‌توانند تمام سوابق را ذخیره کنند [۲۳] درنتیجه، داده‌ها و پارامترهای قدیمی به دلیل محدودیت ذخیره‌سازی و کاهش قدرت محاسباتی، حذف می‌شوند [۲۴].

هدف از یادگیری افزایشی این است که مدل یادگیری بدون اینکه دانش موجود خود را فراموش کند، با داده‌های جدید هم سازگار شود و مدل را دوباره آموزش ندهد [۲۵].

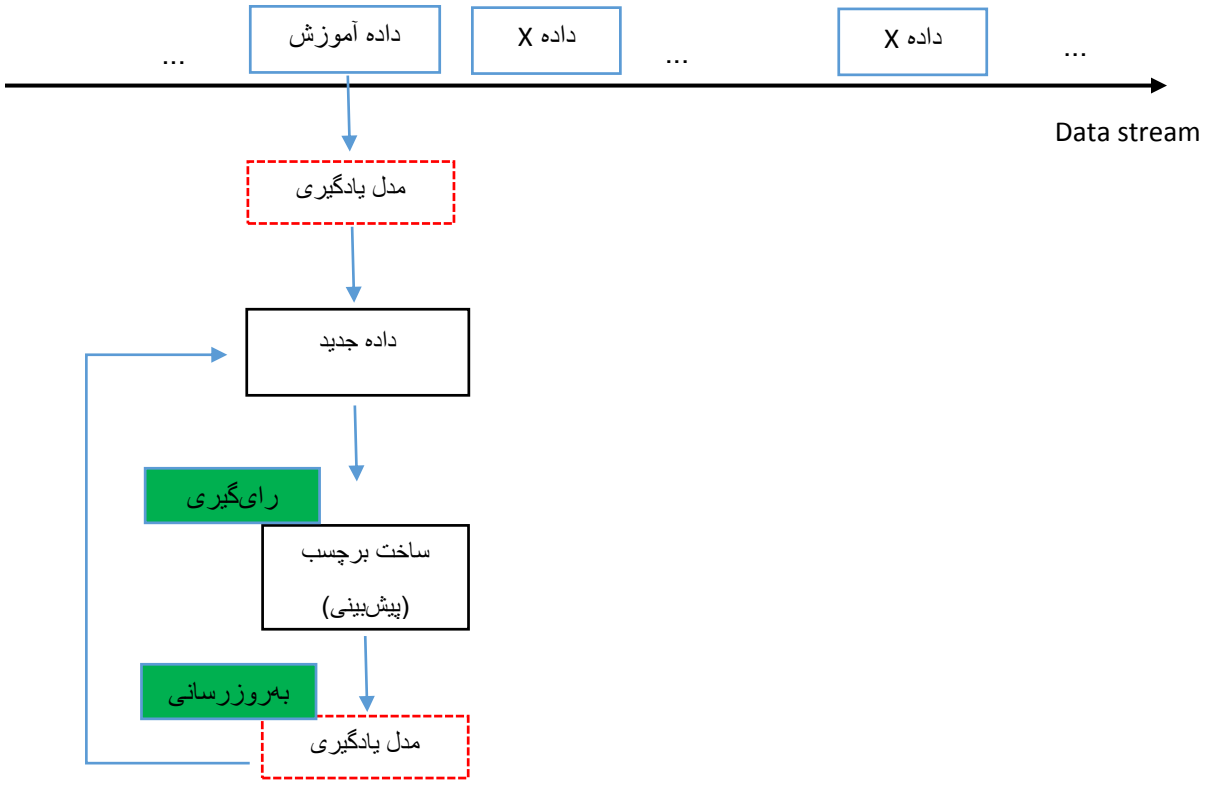
الگوریتم‌های افزایشی غالبا در جریان داده یا کلان داده اعمال می‌شوند، به ترتیب مشکلات موجود بودن داده‌ها و کمبود منابع را برطرف می‌کنند. پیش‌بینی روند برخی از نمونه‌های جریان داده این است که داده‌های جدید به‌طور مداوم در دسترس قرار می‌گیرند. با استفاده از یادگیری افزایشی در داده‌های بزرگ، طبقه‌بندی داده‌ها سریع‌تر انجام می‌شود [۲۶].

در طول روند یادگیری افزایشی، ویژگی‌ها و پیش‌بینی برچسب، مدل جدید را به مدل قبلی نزدیک می‌کند [۲۷] و هدف، یافتن یک‌راه حل برای تقلیل اثر داده‌های قدیمی بدون ذخیره آن‌ها است [۲۸].

۳-۲ فرایند مدل پیشنهادی

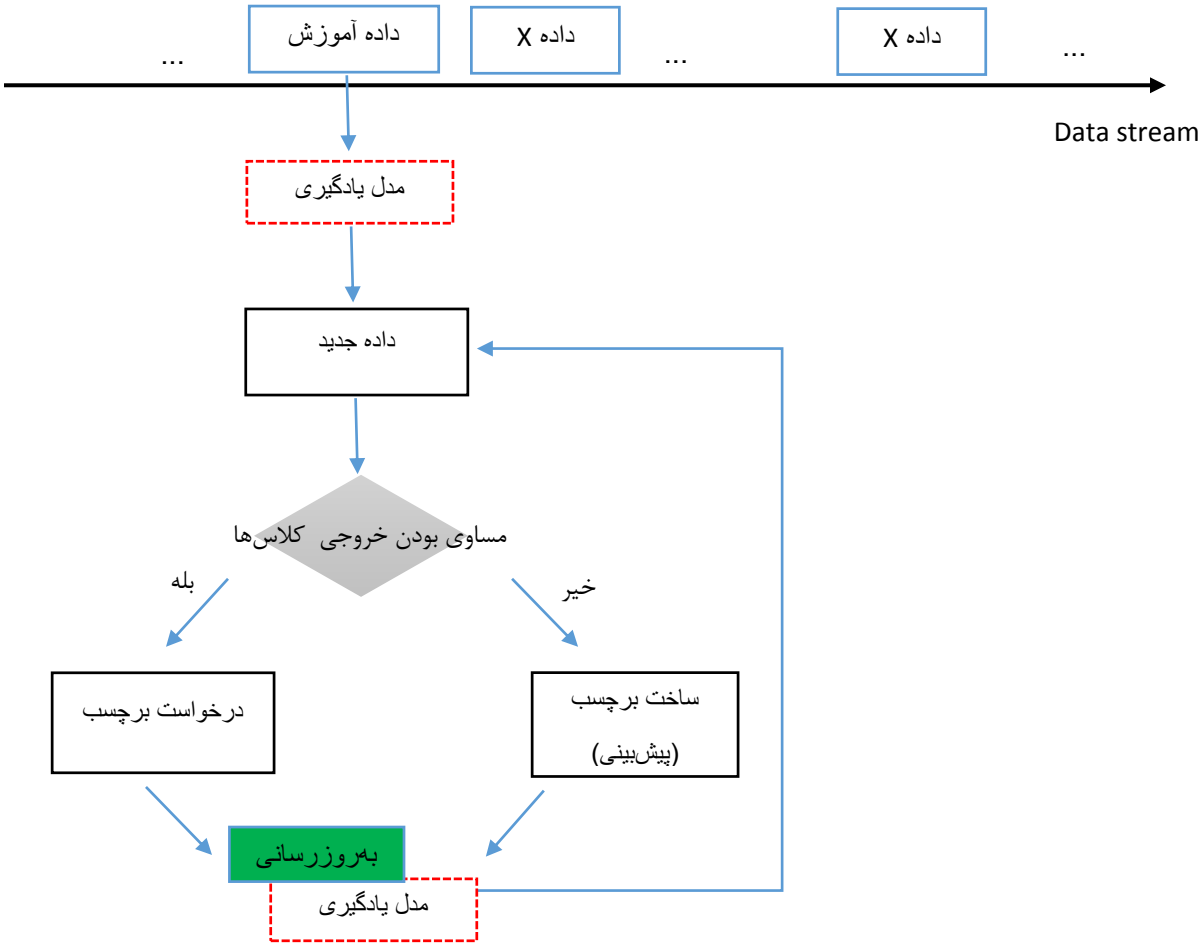
در این قسمت ابتدا مدل‌های پیشنهادی را به‌طور خلاصه شرح می‌دهیم و سپس به توضیح کامل هرکدام از بخش‌های آن‌ها خواهیم پرداخت. در روش اول، ابتدا جریانی از داده‌ها به‌عنوان ورودی سیستم، دریافت می‌شود؛ این داده‌ها شامل دو کلاس می‌باشند و هدف ما این است که نسبت دقیقی از توزیع هر کلاس داده را تخمین بزنیم. با کمک ۲۰۰۰ داده اولیه، مدل یادگیری کمی تشکیل می‌شود. برای ساخت مدل یادگیری کمی از ۵ مدل

متفاوت بهره گرفتیم و هرکدام به صورت جداگانه تعریف شدند. بعد از ساخت مدل اولیه در هر مرحله با ورود داده‌های جدید برچسب داده جدید با کمک ۵ مدل طبقه‌بند ساخته می‌شود و با کمک رای‌گیری از این ۵ مدل برچسب کلی بدست می‌آید و مدل های طبقه‌بند به‌روزرسانی خواهند شد. این روند تا اتمام داده‌ها، تکرار خواهد شد. در شکل ۳-۱ مراحل کار مدل اول را مشاهده می‌کنید.



شکل۳-۱: الگوریتم مدل پیشنهادی یک

تفاوت مدل پیشنهادی دوم با این مدل در مرحله درخواست برچسب است. برای درخواست برچسب از استراتژی خاصی استفاده شده که در ادامه توضیح خواهیم داد. شکل ۳-۲ روند کلی این مدل را نمایش می‌دهد.

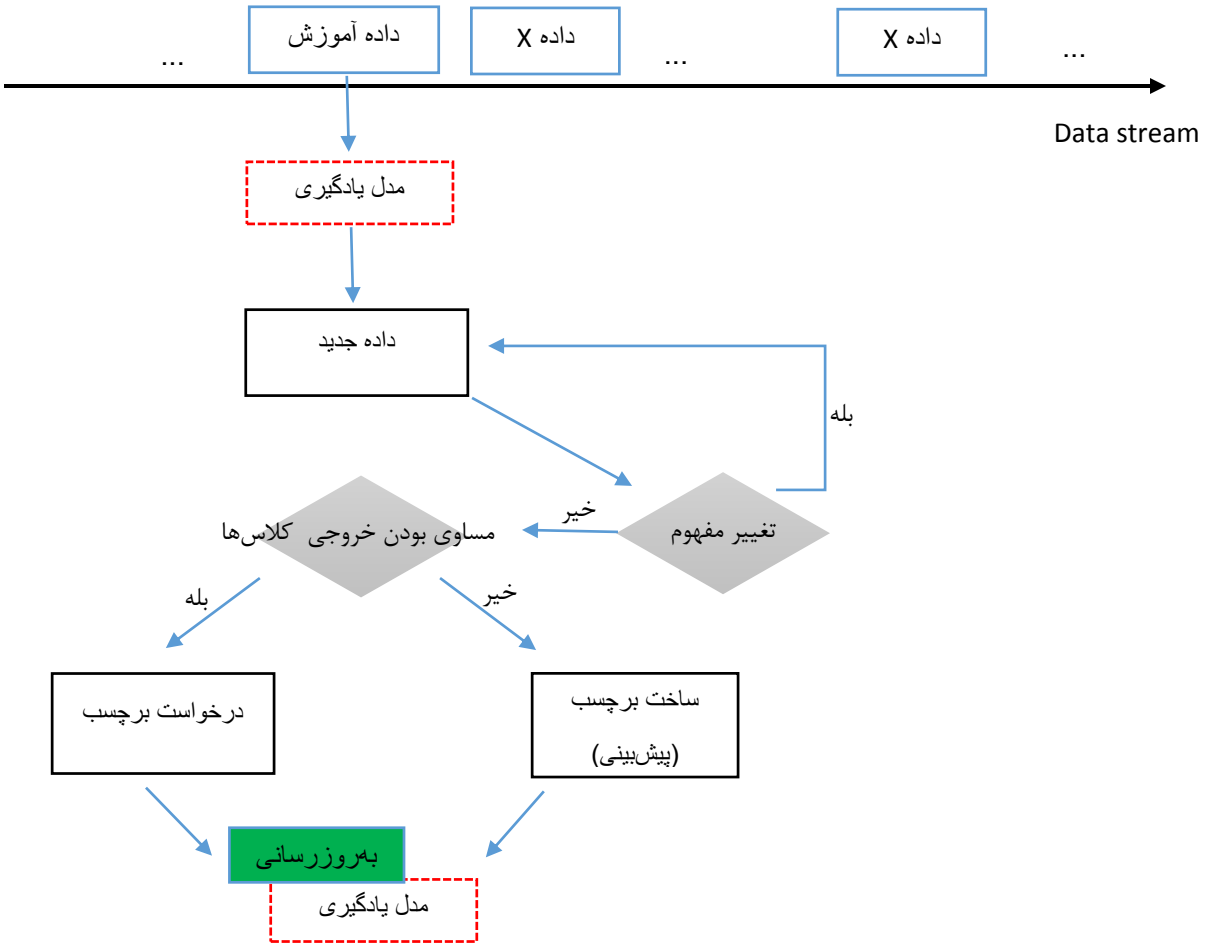


شکل۳-۲: الگوریتم مدل پیشنهادی دوم

برای مدل پیشنهادی دوم از تعداد زوج طبقه‌بند (۴ یا ۲) مدل متفاوت استفاده می‌کنیم و خروجی آن‌ها در صورتی که برای دو کلاس داده برابر باشند (از هر کلاس دو یا یک خروجی تولید شود) درخواست برچسب داده ارسال می‌شود و برای به‌روزرسانی مدل یادگیری کمی از برچسب اصلی داده استفاده می‌کنیم. برای دیگر حالات، خروجی کلی برای برچسب داده مانند روش اول با رای‌گیری از خروجی طبقه‌بندها بدست می‌آید.

۳-۲-۱ استفاده از تغییر مفهوم[۱] در روش دوم

از آنجایی که تغییر مفهوم برای داده‌های در جریان، بسیار حائز اهمیت است؛ لذا ما در این روش سعی می‌کنیم با بررسی تغییر مفهوم در هر مرحله و صرف‌نظر کردن از داده‌های دارای تغییر مفهوم ، به دقت مدل طبقه‌بند در روش دوم بیافزاییم. فرایند این روش را در شکل ۳-۳ مشاهده می‌کنید.



شکل۳-۳: استفاده از تغییر مفهوم برای مدل پیشنهادی دوم

[۱] concept drift

برای این مدل هم از تعداد فرد طبقه‌بند استفاده می‌کنیم و تفاوت آن با مدل دوم در بکارگیری تغییر مفهوم در جریان داده‌های ورودی است. در این روش، با ورود داده جدید، ابتدا شرط وجود تغییر مفهوم بررسی می‌شود و در صورتی‌که تغییر مفهوم برای داده رخ داده باشد، با صرف‌نظر ار آن داده از تغییر نامطلوب در مدل یادگیری جلوگیری می‌کنیم. برای حالتی که تغییر مفهوم نداریم مانند روش دوم به رای‌گیری طبقه‌بندها می‌پردازیم و باقی مراحل مانند قبل انجام می‌پذیرد.

۳-۳ الگوریتم‌های یادگیری

ما در این تحقیق، برای ساخت مدل یادگیری از چند مدل یادگیری استفاده کردیم که در ادامه آمده است:

۳-۳-۱ درخت هوفدینگ[۱]

طبق [۲۹] درخت هوفدینگ یا درخت تصمیم سریع، یک الگوریتم افزایشی از درخت تصمیم می‌باشد که در هر لحظه قادر به یادگیری از جریان داده‌های عظیم است و از این واقعیت استفاده می‌کند که یک نمونه کوچک اغلب می‌تواند برای انتخاب ویژگی و تقسیم‌بندی کلاس داده‌ها کافی باشد و همچنین می‌توان نشان داد که خروجی این مدل، با یک مدل غیر افزایشی با میزان داده بی‌نهایت تقریباً مشابه است. درخت هوفدینگ شامل چندین پارامتر است که به‌صورت پیش‌فرض مقداردهی شده است.

پارامتر اول حداکثر حافظه مصرفی درخت، پارامتر دوم فضای موردنیاز برای هر مرحله آموزش، پارامتر سوم تعداد کلاس‌ها برای مشاهده برگ‌ها، پارامتر چهارم معیار تقسیم هر مرحله، پارامتر پنجم میزان خطای مجاز، پارامتر ششم آستانه قطع، پارامتر هفتم مجوز تقسیم باینری، پارامتر هشتم مجوز توقف هنگام محدودیت حافظه، پارامتر نهم فعال یا غیرفعال کردن ویژگی‌های ضعیف، پارامتر دهم فعال یا غیرفعال کردن هرس اولیه، پارامتر یازدهم

[۱] Hoeffding Tree

روش پیش‌بینی مورداستفاده در برگ‌ها، پارامتر دوازدهم میزان تعداد برگ‌ها برای استفاده از کلاس‌بند بیز ساده و آخرین پارامتر که خالی بودن آن مشخص می‌کند که تمام ویژگی‌ها عددی هستند.

۳-۳-۲ حافظه خودتنظیم با طبقه‌بند k نزدیک‌ترین همسایه[۱]

با توجه به [۳۰] مدل حافظه خودتنظیم، با کمک مدل‌هایی که مفاهیم فعلی و قبلی را مورد هدف قرار می‌دهد، ساخته می‌شود. SAM از دو حافظه کمک می‌گیرد: STM برای مفهوم فعلی و LTM برای حفظ اطلاعات در مورد مفاهیم گذشته.

یک فرایند وظیفه کنترل اندازه STM را بر عهده دارد و همچنین اطلاعات LTM را با STM تطبیق می‌دهد. این ماژول‌ها از یک کتابخانه C ++ استفاده می‌کند که برای تسریع برخی از محاسبات الگوریتم استفاده می‌شود. هنگام فراخوانی توابع کتابخانه، مهم است که نوع آرگومان مناسب را ارسال کنید. عملکرد این چارچوب با انواع استاندارد پایتون و در C ++ هم با برچسب‌های ۸ بیتی کار می‌کند. این طبقه‌بند شامل چندین پارامتر است که در ادامه توضیح می‌دهیم:

پارامتر اول تعداد نزدیک‌ترین همسایگان، پارامتر دوم نوع وزن دهی نزدیک‌ترین همسایه‌ها، پارامتر سوم حداکثر نقاط داده‌ای ذخیره‌شده، پارامتر چهارم فضای مورداستفاده برای LTM، پارامتر پنجم مدل تطبیق حافظه STM، پارامتر ششم حداقل اندازه STM و پارامتر آخر استفاده یا عدم استفاده از LTM را مشخص می‌کند.

[۱] Self Adjusting Memory coupled with the kNN classifier (SAMKNN)

۳-۳-۳ k نزدیک‌ترین همسایگی[۱]

K نزدیک‌ترین همسایگی روشی است که در داده‌کاوی، یادگیری ماشین و تشخیص الگو مورداستفاده قرار می‌گیرد. این الگوریتم برای مسائل طبقه‌بندی و رگرسیون و نیز یادگیری کمی، قابل‌استفاده است و اغلب به دلیل سهولت تفسیر نتایج و زمان محاسبه پایین، مورداستفاده قرار می‌گیرد.

یک روش طبقه‌بند غیر پارامتری است که آخرین نمونه‌های آموزشی را ثبت می‌کند. در این مدل، برچسب کلاس در دو مرحله به دست می‌آید:

- در هر پنجره داده نزدیک‌ترین n همسایه به نمونه را تشخیص دهد.
- برای تعیین کلاس نمونه آزمایش، معدل برچسب کلاس‌های n نمونه که در نزدیک‌ترین همسایگی آن هستند را به دست آورد.

مولفه‌های این طبقه‌بند شامل تعداد همسایه برای جستجو، حداکثر تعداد برگ (هر چه این عدد بیشتر باشد زمان ساخت درخت سریع‌تر و زمان جست‌وجو کندتر خواهد بود) و تعیین معیار فاصله می‌باشد.

۴-۳-۳ بیز ساده[۲]

بیز ساده یک الگوریتم طبقه‌بندی است که به دلیل سادگی و هزینه محاسباتی پایین شهرت دارد. این طبقه‌بند با توجه به کلاس‌های مختلف آموزش دیده و برای هر نمونه بدون برچسب، کلاسی را که به آن تعلق دارد با دقت بالا پیش‌بینی می‌کند.

[۱] k-nearest neighbors

[۲] Naive Bayes

۳-۳-۵ جنگل تصادفی تطبیقی[۱]

جنگل های تصادفی تطبیقی روشی برای طبقه‌بندی جریان داده‌های در حال توسعه می‌باشد که شامل که استراتژی خاصی برای نظارت بر داده‌های ورودی و بررسی تغییر مفهوم دارد. در این روش روی داده‌های تحت نظر نرمال‌سازی صورت می‌پذیرد تا میانگین و انحراف معیار صفر باشد. فرض بر این است که داده‌های نرمال‌سازی شده در فاصله میانگین قرار دارند.

۳-۴ استراتژی انتخاب داده برای ارسال درخواست برچسب

با توجه به خروجی طبقه‌بندهای استفاده شده برای پیاده‌سازی، ۴ یا ۲ خروجی برای کلاس داده جدید بدست می‌آید. اگر در این خروجی‌ها مقدار مساوی از دو کلاس برای داده بدست آمده باشد این داده برای ارسال درخواست برچسب انتخاب می‌شود و مدل افزایشی با برچسب واقعی داده به‌روزرسانی می‌شود. در غیر این صورت یعنی حالتی که خروجی دو کلاس برابر نیستند از رأی اکثریت برای ساختن مدل افزایشی استفاده می‌کنیم.

۳-۵ روش‌های اندازه‌گیری کلاس جریان داده

در فصل دو برای اندازه‌گیری کلاس جریان داده، چندین روش معرفی شد. اولین و ساده‌ترین روش طبقه‌بندی و شمارش (CC) است که شامل برچسب‌گذاری هر نمونه با طبقه‌بند و سپس شمارش نمونه‌ها در هر کلاس است. روش دوم صورت احتمالاتی روش اول (PCC) است که احتمال تعلق نمونه را به هر کلاس مشخص می‌کند. روش آخر نیز مدل تعدیل‌شده (ACC) هست که کلاس داده را بر اساس نرخ صحیح مثبت و غلط مثبت به دست می‌آورد. ما در این تحقیق از ساده‌ترین روش یعنی طبقه‌بندی و شمارش بهره بردیم.

[۱] Adaptive Random Forest

۳-۶ تغییر مفهوم

تغییر مفهوم در جریان داده با روش‌های متعددی قابل محاسبه است. ما در این تحقیق از الگوریتم پنجره کشویی تطبیقی[۱] استفاده می‌کنیم. این الگوریتم روشی برای تشخیص تغییرات و نگه داشتن اطلاعات در مورد یک جریان داده است. در این روش، حداکثر میانگین داده‌های موجود در دو پنجره متوالی در جریان داده مشخص می‌شود و در صورتی‌که مقدار میانگین از این آستانه بیشتر شود، در آن نقطه تغییر تشخیص داده می‌شود. این تابع با تجزیه و تحلیل نقاط برش مختلف در پنجره کشویی وجود یا عدم وجود تغییر مفهوم را بررسی می‌کند [۳۱].

[۱] ADWIN (ADaptive WINdowing)

فصل چهارم: پیاده‌سازی و ارزیابی روش جدید

در این فصل به بیان پیاده‌سازی و بررسی راهکار پیشنهادی و بیان جزئیات مربوط به آن می‌پردازیم. ابتدا مجموعه داده مورداستفاده در آزمایش را تعریف می‌کنیم؛ سپس روش کار را به تفضیل بیان می‌کنیم و در پایان بخش، مقایسه‌ای با روش‌های ارائه‌شده در فصول ابتدایی خواهیم داشت.

۴-۱ مجموعه داده

ما از یک سری پایگاه داده برای ارزیابی این الگوریتم استفاده کردیم. این داده‌ها در [۱۶] نیز به‌کاررفته است. در ادامه توضیح هر یک از مجموعه داده‌ها آمده است:

- Bike: شامل سوابق ساعتی یک سیستم به اشتراک‌گذاری دوچرخه با اطلاعات مربوط به آب و هوا و فصلی بین سال های ۲۰۱۱ و ۲۰۱۲ می باشد. هدف این است که پیش بینی کنیم آیا تقاضای بالا یا کم وجود دارد. این مجموعه شامل ۱۷،۳۷۹ مورد و دارای ۴ ویژگی است.
- Mosquitoes: دارای داده های آزمایشگاهی با اطلاعات عبور پشه‌ها از یک سنسور حساس به نور است. داده های سنسور شامل هفت ویژگی برای هر رویداد است: فرکانس بالایی (WBF) و فرکانس‌های شش هارمونیک اول. دما در طول جریان تغییر می کند، که بر متابولیسم حشرات تاثیر می گذارد و در نتیجه تغییر فرکانس بالتیک خود را دارد. ما درجه حرارت را به عنوان متغیر پنهان برای وظایف اندازه گیری و طبقه بندی در نظر می گیریم. این مجموعه داده شامل ۱۳،۴۱۰ مورد است.
- Insects: شامل حوادثی است که توسط یک سنسور تولید می شود. هدف این است که نوعی پشه را از سایر پشه‌ها تفکیک کنند. این شامل ۸۳،۳۳۹ مورد و دارای ۹۳ ویژگی است.
- NOAA: از شرایط هواشناسی ثبت شده توسط اداره ملی اقیانوسی و جو زمین برای ۵۰ سال تشکیل شده است. این مجموعه داده شامل هشت ویژگی و ۱۸۱۵۹ ثبت روزانه است.
- Arabic-Digit: یک نسخه اصلاح‌شده از اعداد عربی شامل ۱۴،۳۸۰ داده و دارای ۲۶ ویژگی
- QG: یک نسخه از مجموعه داده Handwritten شامل ۱۳،۲۷۹ داده و دارای ۶۳ ویژگی

۴-۲ معیارهای ارزیابی

ما با استفاده از چند معیار ارزیابی در این پژوهش به بررسی عملکرد الگوریتم پیشنهادی پرداختیم. همان‌طور که در فصل دوم اشاره کردیم معیارهای خطای مطلق، خطای مطلق نسبی و خطای نسبی نرمال هرکدام دارای نقص‌هایی بودند؛ بنابراین از معیارهای دیگر برای ارزیابی استفاده می‌کنیم که توضیحات مربوط به هر یک از این معیارهای ارزیابی در ادامه آمده است.

۴-۲-۱ دقت[۱]

دقت رده‌بندی، بر اساس ماتریس درهم‌ریختگی[۲] محاسبه می‌شود که برابر با نسبت موارد حقیقی به کل حالات موجود می‌باشد. در جدول ۴-۱ توضیحات مربوط به ماتریس و در رابطه (۴-۱) تعریف این معیار آمده است.

$$Accuracy = \frac{TN+TP}{TN+FN+TP+FP} \quad \text{(رابطه ۴-۱)}$$

جدول ۴-۱ ماتریس درهم‌ریختگی

		کلاس تخصیص‌یافته	
		مثبت	منفی
کلاس واقعی	مثبت	مثبت حقیقی TP	منفی کاذب FN
	منفی	مثبت کاذب FP	منفی حقیقی TN

[۱] Accuracy

[۲] confusion matrix

۴-۲-۲ خطا

میزان خطای مدل یادگیری کمی، نسبت موارد کاذب به کل حالات موجود (رابطه ۴-۲) را مشخص می‌کند؛ به عبارتی این مقدار برابر میزان اختلاف دقت از مقدار واحد است.

$$Error = \frac{FN+FP}{TN+FN+TP+FP} = ۱ - Accuracy$$ (رابطه ۴-۲)

۴-۲-۳ صحت[۱]

این معیار نیز با کمک ماتریس درهم‌ریختگی به دست می‌آید و مقدار آن برابر است با تقسیم تعداد نمونه‌هایی که به‌درستی شناسایی‌شده‌اند به کل شناسایی‌های درست و نادرست مثبت. در رابطه ۴-۲ تعریف این معیار را مشاهده می‌کنید.

$$Precision = \frac{TP}{TN+TP}$$ (رابطه ۴-۳)

۴-۳ کتابخانه scikit-multiflow

در زبان برنامه‌نویسی پایتون چندین کتابخانه برای کار با داده‌ها و داده‌های در جریان وجود دارد. یکی از این کتابخانه‌ها که برای داده‌های در جریان مورداستفاده قرار می‌گیرد کتابخانه scikit-multiflow می‌باشد.

[۱] Precision

سال‌های اخیر شاهد گسترش نرم‌افزارهای رایگان و منبع باز[1] (FOSS) در جامعه تحقیقاتی بوده‌ایم. با پیروی از اصول FOSS، کتابخانه scikit-multiflow، یک چارچوب پایتون را برای پیاده‌سازی الگوریتم‌ها و انجام آزمایش‌ها در زمینه یادگیری ماشین بر روی جریان داده‌های درحال‌توسعه معرفی شد [۳۲].

scikit-multiflow یک کتابخانه منبع باز برای انجام یادگیری ماشین در تنظیم جریان است. در [۳۳] چندین کتابخانه پرکاربرد دیگر در حوزه داده‌های در جریان نظیر Creme و River آمده است. این کتابخانه‌ها عمدتا به زبان پایتون نوشته می‌شود و برای موارد مختلفی می‌توان از آن‌ها استفاده کرد. ازجمله: طبقه‌بندی، رگرسیون، خوشه‌بندی، یادگیری نمایشی، یادگیری چند برچسب و چند خروجی، پیش‌بینی و تشخیص ناهنجاری.

کتابخانه scikit-multiflow شامل مولدهای جریان، روش‌های یادگیری (طبقه‌بندی و رگرسیون)، آشکارسازهای تغییر مفهوم و روش‌های ارزیابی است [۳۴]. در scikit-multiflw، داده‌ها توسط کلاس Stream نمایش داده می‌شوند. مهم‌ترین قابلیت کلاس Stream ارائه نمونه‌های جدید داده در صورت تقاضا است. با کمک جریان سازها که یک منبع ارزان از داده‌ها هستند، نمونه‌های داده بر اساس تقاضا تولید می‌شوند تا از ذخیره فیزیکی داده‌ها جلوگیری شود. در scikit-multiflow چندین مولد جریان وجود دارد و همه آن‌ها به روشی مشابه کار می‌کنند [۳۵].

۴-۴ پیاده‌سازی روش پیشنهادی

در ابتدایی‌ترین مرحله و مرحله اول، چهار مدل طبقه‌بند اولیه با ۲۰۰۰ داده آموزشی تشکیل دهیم؛ در هر مرحله داده جدید دریافت می‌شود و بعد از بررسی خروجی‌های این چهار مدل روند ادامه می‌یابد. خروجی‌های بدست آمده

[1] Free and Open Source Software

از مدل‌ها را در یک لیست ذخیره می‌کنیم. با کمک تابع max مقدار بیشترین تکرار را در لیست بدست می‌آوریم. این تابع برای تک تک داده‌های آزمایش فراخوانی می‌شود.

برای روش اول که بدون درخواست برچسب پیاده سازی شده است با مشخص شدن کلاس غالب برچسب داده مورد نظر ساخته می‌شود و سپس به کمک آن، مدل را به صورت افزایشی می‌سازیم. در روش دوم برای این حالت، برای داده مورد نظر درخواست برچسب ارسال می‌شود و یک عدد به شمارنده خاص مربوط به درخواست برچسب افزوده خواهد شد. روش سوم مانند روش دوم می‌باشد اما قبل از مشخص کردن کلاس غالب، تغییر مفهوم برای داده بررسی می‌شود که در صورت وجود تغییر مفهوم، داده جدید وارد می‌شود. در تمامی روش‌ها، بعد از بدست آوردن مقدار برچسب داده (روش اول با استفاده از مدل ساخته شده و روش دوم و سوم در حالت تساوی خروجی‌ها با استفاده از مقدار برچسب داده و باقی حالات مانند روش اول) مدل طبقه‌بندها به صورت افزایشی به‌روزرسانی خواهند شد. سپس داده بعدی در جریان داده دریافت می‌شود و روند فوق را تکرار می‌کنیم.

در ادامه آمار کلی و پارامترهای استفاده‌شده در الگوریتم به‌طور خلاصه آمده است:

- تعداد داده برای طبقه‌بندی اولیه (آموزش): ۲۰۰۰ نمونه
- تعداد داده برای درخواست برچسب: قابل محاسبه برای هر مجموعه داده
- روش‌های انتخاب نمونه: استراتژی خاص بیان شده در بخش ۳-۴
- روش اندازه‌گیری کلاس جریان داده: CC
- مدل‌های طبقه‌بندی: Hoeffding Tree، SAM، KNN و NaiveBayes و Adaptive Random Forest
- معیارهای ارزیابی: دقت، صحت، خطا

۴-۵ ارزیابی دقت روش پیشنهادی

با اجرای برنامه دقت و خطای بدست آمده برای ۴ مدل طبقه‌بند و هر مجموعه داده به‌صورت زیر محاسبه شده‌است و میانگین دقت و خطا را در ادامه مشاهده خواهید کرد.

جدول۴-۲: دقت و خطای مدل‌های یادگیری برای مجموعه داده Bike

	خطا	صحت	دقت
روش اول	۰.۳۲۵۲	۰.۵۹۷۷	۰.۶۷۴۸
روش دوم	۰.۳۲۲۷	۰.۵۹۰۸	۰.۶۷۷۳
روش سوم	۰,۳۲۲۷	۰,۵۹۰۸	۰,۶۷۷۳
روشSQSI_IS (پایه)	میانگین خطا برای این پایگاه داده با ارسال ۵ درصد درخواست برچسب: ۰,۱۵۳۳		

تعداد ارسال برچسب برای روش دوم: ۳۹۱ معادل دو درصد

تعداد داده دارای تغییر مفهوم برای روش سوم: ۰

میانگین خطای کمی مدل پایه برای این پایگاه داده با ارسال درخواست ۵ درصد به میزان ۰,۱۵۳۳ گزارش شده است. با توجه به داده‌های بدست آمده، درخواست برچسب کمتر اعمال شده‌است. در این مجموعه داده به دلیل اینکه داده‌ها با تغییرات فصل تغییر می‌کنند مدل‌های پیشنهادی خوب عمل نکردند. برای این مجموعه داده، تغییر مفهومی شناسایی نشد و نتایج برای روش‌های دوم و سوم یکسان بدست آمد و به دقت مدل پیشنهادی کمکی نکرد.

جدول۴-۳: دقت و خطای مدل‌های یادگیری برای مجموعه داده *Mosquitoes*

	خطا	صحت	دقت
روش اول	۰.۳۲۷۳	۰.۹۳۷۵	۰.۶۷۲۷
روش دوم	۰,۲۵۷	۱,۰	۰,۷۴۲۳
روش سوم	۰,۲۴۳	۰,۸۹	۰,۷۵۷
روشSQSI_IS	میانگین خطا برای این پایگاه داده با ارسال ۵ درصد درخواست برچسب: ۰,۲۹		

تعداد ارسال برچسب برای روش دوم: ۵۶ معادل ۰,۴ درصد

تعداد داده دارای تغییر مفهوم برای روش سوم: ۲۰

میانگین خطای کمی مدل پایه برای این پایگاه داده با ارسال درخواست ۵ درصد به میزان ۰,۲۹ گزاش شده است. طبق مقادیر بدست آمده درخواست برچسب به شدت پایین آمده است. دقت تمامی مدل‌ها نزدیک به دقت مدل SQSI_IS می‌باشد که با توجه به کاهش شدید درخواست برچسب، نتیجه قابل قبول است.

جدول۴-۴: دقت و خطای مدل‌های یادگیری برای مجموعه داده *Insects*

	خطا	صحت	دقت
روش اول	۰,۲۴۷۶	۰,۶۵۴۴	۰,۷۵۲۳
روش دوم	۰,۰۸۳۲	۰,۸۷۱۳	۰,۹۱۶۷
روش سوم	۰,۰۷	۰,۸۹۹	۰,۹۳

روشSQSI_IS	میانگین خطا برای این پایگاه داده با ارسال ۵ درصد درخواست برچسب: ۰,۱۰۶۶

تعداد ارسال برچسب برای روش دوم: ۱۹۱۲ معادل دو درصد

تعداد داده دارای تغییر مفهوم برای روش سوم: ۱۹

میانگین خطای کمی مدل پایه برای این پایگاه داده با ارسال درخواست ۵ درصد به میزان ۰,۱۰۶۶ گزاش شده است. در روش های دوم و سوم، مدل یادگیری عملکرد مناسبی را از خود نشان دادهاست که با توجه به کاهش درخواست برچسب، بسیار ارزشمند است.

جدول۴-۵: دقت و خطای مدل‌های یادگیری برای مجموعه داده NOAA

	خطا	صحت	دقت
روش اول	۰,۳۱۲۴	۰,۶۸۷۴	۰,۶۸۷۶
روش دوم	۰.۲۷۵۳	۰,۷۴۸۸	۰.۷۲۴۷
روش سوم	۰,۲۶۴۵	۰,۷۴	۰,۷۳۵
روشSQSI_IS	میانگین خطا برای این پایگاه داده با ارسال ۵ درصد درخواست برچسب: ۰,۲۹		

تعداد ارسال برچسب برای روش دوم: ۳۲۴ معادل دو درصد

تعداد داده دارای تغییر مفهوم برای روش سوم: ۲۲۲

میانگین خطای کمی مدل پایه برای این پایگاه داده با ارسال درخواست ۵ درصد به میزان ۰,۲۹ گزاش شده است. داده‌های این مجموعه مرتبط با داده‌های هواشناسی هستند و به دلیل عدم پایداری جو، از ثبات کمی برخوردارند؛ لذا برای این مجموعه داده تغییر مفهوم زیادی داریم ولی مدل‌های پیشنهادی با درخواست برچسب کمتر، دقت را تا حدودی حفظ کردند و حتی به میزان اندکی بهبود داده‌اند.

جدول۴-۶: دقت و خطای مدل‌های یادگیری برای مجموعه داده Arabic-Digit

	خطا	صحت	دقت
روش اول	۰.۰۷۶	۰.۹۱۸۴	۰,۹۲۳۹
روش دوم	۰.۰۵۸	۰.۹۵۳۸	۰,۹۴۱۸
روش سوم	۰,۰۳۴	۰,۹۵	۰,۹۶۶
روشSQSI_IS	میانگین خطا برای این پایگاه داده با ارسال ۵ درصد درخواست برچسب: ۰,۱		

تعداد ارسال برچسب برای روش دوم: ۳۲۹ معادل دو درصد

تعداد داده دارای تغییر مفهوم برای روش سوم: ۳۸

میانگین خطای کمی مدل پایه برای این پایگاه داده با ارسال درخواست ۵ درصد به میزان ۰,۱ گزاش شده است. با توجه به پایین بودن تغییر مفهوم در این مجموعه داده با ارسال درخواست برچسب کمتر دقت نیز تا حد خوبی افزایش داشته است و خطای مدل به میزان اندکی کاهش داشته است.

جدول۴-۷: دقت و خطای مدل‌های یادگیری برای مجموعه داده QG

	خطا	صحت	دقت
روش اول	۰.۰۱۳۳	۰.۹۹۴۹	۰.۹۸۶۷
روش دوم	۰,۰۰۶۳	۰,۹۹۲۷	۰.۹۹۳۶
روش سوم	۰,۰۰۶۳	۰,۹۹۲۷	۰,۹۹۳۶
روشSQSI_IS	میانگین خطا برای این پایگاه داده با ارسال ۵ درصد درخواست برچسب: ۰,۳۳		

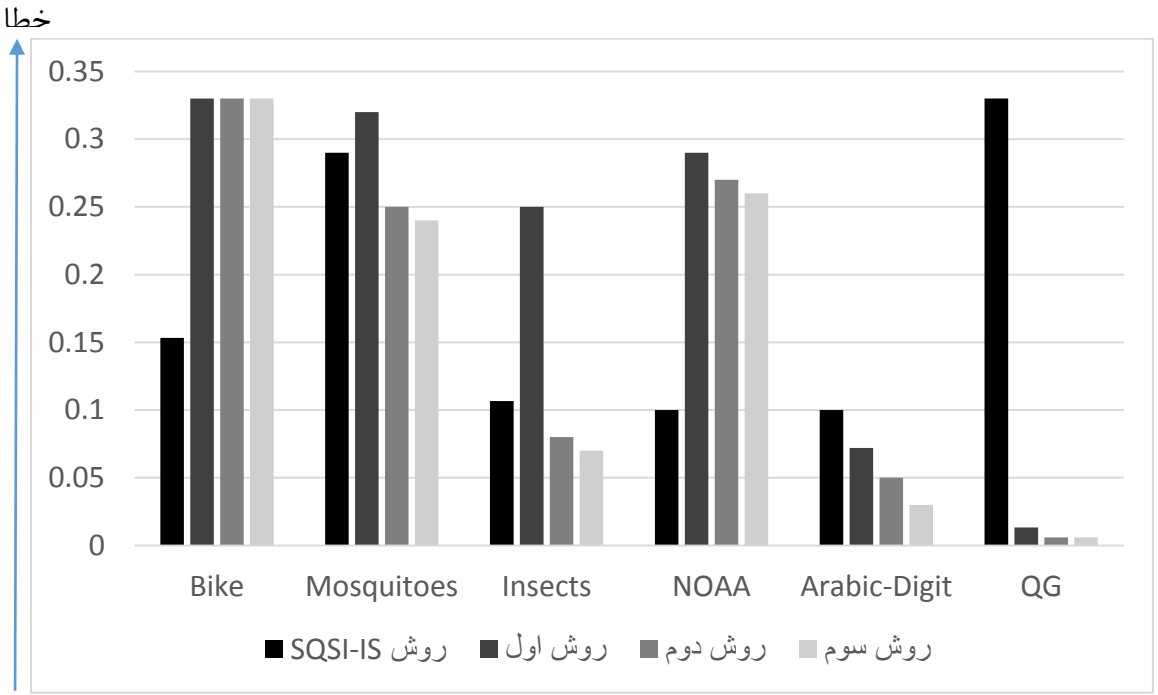
تعداد ارسال برچسب برای روش دوم: ۱۸۰ معادل یک درصد

تعداد داده دارای تغییر مفهوم برای روش سوم: ۰

میانگین خطای کمی مدل پایه برای این پایگاه داده با ارسال درخواست ۵ درصد به میزان ۰,۳۳ گزاش شده است یعنی دقتی حدود ۰,۶۷ را دارا است و طبق مقادیر بدست آمده، دقت روش‌های پیاده‌سازی شده نسبت به این مقدار افزایش خوبی داشته‌اند؛ همچنین درخواست برچسب نسبت به مدل پایه به شدت کاهش داشته است و حتی برای روش پیشنهادی اول بدون هیچ درخواست برچسبی دقت بالایی بدست آمده است. برای این مجموعه داده، تغییر مفهومی شناسایی نشد و نتایج برای روش‌های دوم و سوم یکسان بدست آمد.

بطور کلی برای تمامی روش‌های پیشنهادی، در تمام مجموعه داده‌ها درخواست برچسب کمتری داشتیم و همچنین برای روش‌های دوم و سوم بجز یک مورد، برای تمام مجموعه داده‌های موجود، دقت بالاتر از روش SQSI-IS بدست آمد که با وجود کاهش مناسب ارسال درخواست برچسب که منجر به کاهش هزینه‌های مربوط به آن خواهد شد، بسیار حائز اهمیت است. دقت برای روش اول، بدون ارسال درخواست برچسب، در دو

مجموعه‌داده اخر نسبت به قبل بهبود داشت. میانگین خطای محاسبه شده برای تمام مدل‌ها در مقایسه با هم در شکل ۴-۱ آمده‌است.



شکل۴-۱ میانگین خطای تمام مدل‌ها در مقایسه با هم

فصل پنجم: نتیجه‌گیری و پژوهش‌های آینده

۵-۱ نتیجه‌گیری

در این تحقیق سعی کردیم با کمک روش یادگیری افزایشی که برای ایجاد مدل داده‌های در جریان استفاده کردیم تا حد زیادی زمان کمتری را برای بررسی داده‌ها صرف کنیم. همچنین چالش‌های مطرح‌شده در بخش‌های ابتدایی را تا حد زیادی بهبود دادیم. یکی از این چالش‌ها هزینه بالای درخواست برچسب داده‌ها بود که در روش ما این هزینه به صورت قابل توجهی کاهش یافت.

در آزمایش‌ها نشان دادیم که با کمک روش‌های یادگیری افزایشی قادر هستیم با حفظ دقت، هزینه‌های زمان و درخواست برچسب را کاهش دهیم. با توجه به این‌که برچسب‌های داده در دنیای واقعی هزینه بالایی دارند این روش می‌تواند تا حدودی موثر واقع شود.

۵-۲ پژوهش‌های آینده

سعی ما این است که برای تحقیقات بعدی، با کمک حذف محدودیت‌هایی مانند اندازه مجموعه آموزشی که به‌عنوان فرضیه در آزمایش‌ها در نظر گرفتیم و با کمک روش‌های یادگیری از داده‌هایی که در لحظه تولید می‌شوند، مدل مناسبی برای استفاده بهتر در دنیای واقعی ارائه دهیم و آن را برای بهبود عملکرد افراد در مشاغل مختلف بکار بگیریم.

در [۳۶] با شکل دیگری از یادگیری کمی آشنا می‌شویم که با عدم قطعیت[۱] همراه است. این روش نقشی محوری در کاهش عدم قطعیت در حین بهینه‌سازی و تصمیم‌گیری دارد که برای برنامه‌های کاربردی در علوم و مهندسی مورد استفاده قرارگرفته است. در مطالعات بعد، پیشرفت‌های اخیر در این حوزه را مورد بررسی قرار خواهیم داد.

[۱] Uncertainty Quantification

همان‌طور که اشاره کردیم روش‌های انتخاب نمونه می‌توانند یک ابزار مفید برای مقابله با مشکلات یادگیری کمی در جریان داده باشند؛ مخصوصا زمانی که برچسب‌های کمی از داده‌ها در دسترس ما هستند؛ که این روش‌ها می‌توانند بسیار مفید واقع شوند.

در این تحقیق برای انتخاب نمونه، از روش ابداعی استفاده شد. روش‌های دیگر انتخاب نمونه از جمله انتخاب نمونه مبتنی بر خوشه را در تحقیقات بعدی در نظر خواهیم گرفت.

با توجه به تحولات علم داده‌کاوی، شناسایی داده‌های پرت موضوع بسیار مهمی است و در برنامه‌های کاربردی که داده‌ها در قالب جریان داده هستند، اهمیت به سزایی دارد [۳۷]. با تشخیص و حذف این داده‌ها می‌توان تاثیر منفی آن را در عملکرد مدل کاهش داد و دقت بالاتری برای مدل محاسبه کرد. در مطالعات بعدی از روش‌های تشخیص داده‌های پرت در جریان داده کمک خواهیم گرفت.

امروزه از یادگیری عمیق[۱] در انواع برنامه‌های کاربردی و تجزیه و تحلیل داده‌های بزرگ استفاده می‌شود. بسیاری از محققان علم داده به دلیل تفسیر برتر آن، یا توانایی در درک راه‌حل‌ها، از یادگیری عمیق برای یادگیری ماشین بهره می‌برند. ما نیز در آینده تلاش خواهیم کرد، یادگیری عمیق را به‌صورت تلفیقی با یادگیری کمی به کار بریم.

[۱] Deep Learning

مراجع

[۱] P González, A Castaño, NV Chawla, "A Review on Quantification Learning", in *CM Computing Surveys (CSUR)*, Vol: ۵۰, No: ۵, Article No: ۷۴., ۲۰۱۷.

[۲] Masud M, Gao J, Khan L, Han J, Thuraisingham BM, "Classification and novel class", in *IEEE*, Trans Knowl Data, ۲۰۱۱.

[۳] dos Reis DM, Flach P, Matwin S, Batista GEAPA, "Fast unsupervised online drift detection", in *ACM SIGKDD*, San Francisco, ۲۰۱۶.

[۴] Gama J, Medas P, Castillo G, Rodrigues P, "Learning with drift detection", in *Springer*, Berlin, Heidelberg, ۲۰۰۴.

[۵] Chen Y, Why A, Batista GEAPA, Mafra-Neto A, Keogh E, "Flying insect classification with inexpensive sensors", *J Insect Behav ۲۷(۵)*, p. ۶۵۷–۶۷۷, ۲۰۱۴.

[۶] Pablo Gonz´alez, Jorge D´ıez, Nitesh Chawla, and Juan Jos´e del Coz, "Why is quantification an interesting", in *Artificial Intelligence*, ۲۰۱۷.

[۷] G Forman, "Counting positives accurately despite inaccurate classification", in *ECML*, Springer. pp ۵۶۰–۵۷۵., ۲۰۰۵.

[۸] Maletzke A, Reis D, Batista G, "Quantification in data streams: initial results", in *BRACIS*, Uberlândia. pp ۰۳–۰۸., ۲۰۱۷.

[۹] Gao W, Sebastiani F, "From classification to quantification in tweet sentiment analysis", in *Soc Netw* , Anal Min, ۲۰۱۶.

[۱۰] Zliobaite I, Bifet A, Pfahringer B, Holmes G, "Active learning with drifting streaming data", in *IEEE*, Trans Neural Netw Learn Syst ۲۵(۱):۲۷–۳۹, ۲۰۱۴.

[۱۱] Souza VMA, Rossi RG, Batista GEAPA, Rezende SO, "Unsupervised active learning techniques for labeling training sets: an experimental evaluation on sequential data", in *Intell Data*, ۲۰۱۷.

[۱۲] os Reis DM, Flach P, Matwin S, Batista GEAPA, "Fast unsupervised online drift detection using incremental Kolmogorov-Smirnov test", in *ACM SIGKDD*, San Francisco. pp ۱۵۴۵–۱۵۵۴, ۲۰۱۶.

[۱۳] Masud M, Gao J, Khan L, Han J, Thuraisingham BM, "Classification and novel class detection in concept drifting data streams under time constraints", in *IEEE*, Trans Knowl Data Eng ۲۳(۶):۸۵۹–۸۷۴, ۲۰۱۱.

[۱۴] Souza VMA, Silva DF, Gama J, Batista GEAPA, "Data stream classification guided by clustering on nonstationary environments and extreme verification latency", in *SDM*, Vancouver. pp ۸۷۳–۸۸۱, ۲۰۱۵.

[۱۵] Milli L, Monreale A, Rossetti G, Giannotti F, Pedreschi D, Sebastiani F, "Quantification trees", in *ICDM*, Dallas. pp ۵۲۸–۵۳۶, ۲۰۱۳.

[۱۶] Maletzke A, dos Reis D, Batista G, "Combining instance selection and self-training to improve data stream quantification", *Journal of the Brazilian Computer Society*, ۲۰۱۸.

[۱۷] B Settles, "Active learning literature survey", *Univ Wis Madison ۵۲*, p. ۱۱.

[۱۸] Souza VMA, Rossi RG, Batista GEAPA, Rezende SO, "Unsupervised active learning techniques for labeling training sets: an experimental evaluation on sequential data", *Intell Data Anal*, p. ۱۰۶۱–۱۰۹۵, ۲۰۱۷.

[۱۹] Zliobaite I, Bifet A, Pfahringer B, Holmes G, "Active learning with drifting streaming data", in *IEEE*, Trans Neural Netw Learn Syst ۲۵(۱):۲۷–۳۹, ۲۰۱۴.

[۲۰] Zhu X, Goldberg AB, "Introduction to semi-supervised learning", in *Synth Lect Artif Intell Mach Learn*, ۲۰۰۹.

[۲۱] André Gustavo Maletzke, Waqar Hassan, "The Importance of the Test Set Size in Quantification Assessment", in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, july ۲۰۲۰.

[۲۲] heaven, Will Douglas, "Our weird behavior during the pandemic is messing with AI models", MIT Technology Review, ۲۰۲۰.

[۲۳] Anders Rasmussen, Riccardo Zucca, Fredrik Johansson, Dan-Anders Jirenhed, and Germund Hesslow, "Purkinje cell activity during classical conditioning with different conditional stimuli explains central tenet of Rescorla Wagner model", in *PNAS*, ۲۰۱۵.

[۲۴] German Parisi, Ronald Kemker, Jose Part, Christopher Kanan, and Stefan Wermter, "Continual Lifelong Learning with Neural Networks: A Review", in *Neural Networks*, ۲۰۱۸.

[۲۵] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu, "Few-Shot Incremental Learning with Continually Evolved Classifiers", in *CVPR*, ۲۰۲۱.

[۲۶] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, Hanwang Zhang, "Distilling Causal Effect of Data in Class-Incremental Learning", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, ۲۰۲۱.

[۲۷] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu, "Large Scale Incremental Learning", in *CVPR*, ۲۰۱۹.

[۲۸] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin, "Learning a Unified Classifier Incrementally via Rebalancing", in *CVPR*, ۲۰۱۹.

[۲۹] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams", in *KDD'۰۱*, *pages ۹۷–۱۰۶*, San Francisco, CA, ۲۰۰۱.

[۳۰] Losing, Viktor, Barbara Hammer, and Heiko Wersing, "Knn classifier with self adjusting memory for heterogeneous concept drift", in *In Data Mining (ICDM)*, ۲۰۱۶ IEEE ۱۶th International Conference on, pp. ۲۹۱-۳۰۰. IEEE, ۲۰۱۶.

[۳۱] A. Bifet, R. Gavalda, "Learning from time-changing data with adaptive windowing" , SIAM international conference on data mining, pp. ۴۴۳-۴۴۸, ۲۰۰۷.

[۳۲] Jacob Montiel, Jesse Read, Albert Bifet, Talel Abdessalem, "A multi-output streaming framework", in *The Journal of Machine Learning Research*, ۲۰۱۸.

[۳۳] J Montiel, M Halford, SM Mastelini, G Bolmier, R Sourty, "machine learning for streaming data in Python", *Journal of Machine Learning Research,* ۲۰۲۱.

[۳۴] Kim, W Nick Street and YongSeog, "A streaming ensemble algorithm (SEA) for large-scale classification", *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM,* vol. ۳۷۷–۳۸۲, ۲۰۰۱.

[۳۵] J. Montiel, J. Read, A. Bifet, T. Abdessalem, "Scikit-multiflow: A multioutput streaming framework", *Journal of Machine Learning ,* p. ۱ ,۵–Research ۱۹ (۷۲) (۲۰۱۸)

[۳۶] MAbdar, F Pourpanah, S Hussain, D Rezazadegan, L Liu, M Ghavamzadeh, P Fieguth, X Cao, A Khosravi, U R Acharya ,"A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges", *IEEE* .۲۰۲۱

[۳۷] فردین، سحر و مهدی هاشم زاده، ۱۳۹۹، "یک رویکرد جدید مبتنی بر یادگیری افزایشی برای تشخیص داده‌های پرت در جریان داده‌ها"، ششمین کنفرانس ملی محاسبات توزیعی و پردازش داده‌های بزرگ، تبریز.

Abstract

In recent years, learning from data streams has attracted the attention of researchers and specialists. However, quantification learning has remained mostly unexplored. In some applications where we need to distribute positive and negative feedback, the use of quantification learning has been very useful. Also, this method can also be used to obtain specific general characteristics about the population of a network, and extract useful practical information by analyzing people's emotions. Quantification learning is very similar to classification, and both of them do the grouping of the data, but their purpose is different; In quantification learning problems, we are not looking for to specify each class of samples, and only general data statistics are important, and the goal is to provide an estimate of the distribution of data. Recent algorithms in the field of quantification learning in data stream have been introduced with the help of changing the concept and using the label request for a large part of the new samples and have been presented with sample selection techniques. In this research, the idea is to request the label of a smaller subset of recent examples and we make the classification model incrementally with the help of several different classification classes. Our experiments show that despite reducing the label request from recent samples and even removing it, the accuracy of the model can be maintained or improved.

Keywords: Quantification learning, Data stream, Classification, Incremental learning, Label request

Shahrood University of Technology

Faculty of Computer Engineering

M.Sc. Thesis in Artificial Intelligence Engineering

Incremental Quantification Learning in Data Stream

By: Elham Ahmadi

Supervisor:

Dr. Hoda Mashayekhi

Advisor:

Dr. Marziea Rahimi

October ۲۰۲۱