# نظرکاوی افزایشی با استفاده از یادگیری فعال بر روی جریان متون

سیدفخرالدین نوربهبهانی

**چکیده: نظرکاوی امروزه به عنوان یکی از کاربردهای پراهمیت پردازش زبان طبیعی مطرح است که به دلیل بالابودن حجم و نرخ نظرات تولیدشده نیاز به روش‌های ویژه‌ای برای پردازش دارد. امروزه با توجه به ماهیت جریان داده‌ای نظرات کاربران در شبکه‌های اجتماعی و سایت‌های تجارت الکترونیکی، استفاده از الگوریتم‌های دسته‌بندی غیر افزایشی باعث می‌گردد به مرور زمان کارایی مدل یادگرفته‌شده برای کاوش نظرات کاهش یافته و عملاً غیر قابل استفاده شود. علاوه بر این به دلیل نامحدودبودن تعداد نظرات، امکان برچسب‌گذاری تمام نظرات برای ایجاد نمونه‌های آموزشی جدید و به روزرسانی مدل یادگرفته‌شده وجود ندارد. از آنجا که ممکن است نظرات جدید دارای واژگان جدید بوده و یا توزیع دسته‌های قطبیت تغییر کند، رانش مفهوم نیز می‌بایست در نظرکاوی افزایشی پشتیبانی گردد.**

**در این مقاله یک روش جدید برای یادگیری قطبیت متون به صورت افزایشی ارائه می‌گردد که با استفاده از یادگیری فعال جریان داده‌ای، متون ارزشمند برای به‌روز رسانی مدل دسته‌بندی را انتخاب می‌کند و پس از تعیین برچسب آنها توسط متخصص انسانی، از آنها برای بهبود مدل دسته‌بندی بهره می‌گیرد. روش پیشنهادی به صورت برخط و بدون نیاز به ذخیره متون، با استفاده از تعداد محدودی متون برچسب‌خورده آموزش می‌بیند و قادر به تشخیص و پشتیبانی از رانش مفهوم می‌باشد. روش پیشنهادی با روش‌های شاخص افزایشی و غیر افزایشی، با استفاده از مجموعه داده‌های معتبر و معیارهای ارزیابی استاندارد مقایسه و ارزیابی می‌شود.**

***کلیدواژه:*** **جریان داده‌ها، رانش مفهوم، نظرکاوی، یادگیری افزایشی، یادگیری فعال.**

## ۱- مقدمه

با رشد روزافزون اطلاعات متنی تولیدشده توسط کاربران در اینترنت، تجزیه و تحلیل احساسات در متون، زمینه کاری جذابی در بین محققان علوم داده‌کاوی و پردازش زبان طبیعی شده است. یکی از کاربردی‌ترین موضوعات در این زمینه تعیین قطبیت متون است که هدف آن تشخیص خودکار نظر نویسنده می‌باشد.

تعیین قطبیت می‌تواند توسط روش‌های یادگیری ماشین یا روش‌های مبتنی بر فرهنگ لغت[1] انجام شود. روش‌های مبتنی بر یادگیری ماشین می‌توانند نظارتی، نیمه نظارتی یا غیر نظارتی باشند. در روش‌های نظارتی به دو مجموعه داده‌های برچسب‌خورده برای آموزش و آزمایش دسته‌بند نیاز است. روش‌های مبتنی بر فرهنگ لغت به دو گروه مبتنی بر واژه‌نامه[2] و مبتنی بر مجموعه‌ای از متون[3] تقسیم می‌شوند. در روش مبتنی بر واژه‌نامه، از واژه‌نامه موجود که شامل مجموعه‌ای از واژگان نظر به همراه میزان منفی یا مثبت‌بودن احساس آنها است استفاده می‌گردد که این واژه‌نامه‌ها می‌توانند با یا بدون استفاده از هستی‌شناسی[4] ایجاد شوند. روش مبتنی بر مجموعه‌ای از متون، بر احتمال وقوع یک کلمه احساس به همراه مجموعه‌ای از واژگان مثبت یا منفی استوار است که بر اساس جستجو در حجم زیادی از متون به دست می‌آید [۱]. مزیت روش‌های مبتنی بر فرهنگ لغت این است که نیازمند حجم زیادی از داده برای آموزش نیستند. با این حال در این روش‌ها می‌بایست واژه‌نامه احساس ساخته شود. روش‌های مبتنی بر فرهنگ لغت استقلال بیشتری از دامنه دارند و این در حالی است که روش‌های مبتنی بر یادگیری ماشین وابسته به دامنه بوده و دقت بالاتری دارند [۲].

تکنیک‌های یادگیری ماشین به دو نوع غیر افزایشی و افزایشی تقسیم می‌شوند. در نوع غیر افزایشی الگوریتم یادگیری ماشین مدلی را با در نظر گرفتن کل مجموعه داده آموزشی یاد می‌گیرد. در این نوع یادگیری هنگامی که مجموعه داده جدید آموزشی در اختیار باشد، مدل جدید یاد گرفته می‌شود. در نوع افزایشی هر نمونه آموزشی جدید می‌تواند مدل یاد گرفته شده را به‌روز کند. در حوزه یادگیری ماشین واژه دیگری که نباید با روش‌های افزایشی و غیر افزایشی اشتباه گرفته شود، عبارت "جریان داده" است که به معنای تولید سریع حجم زیادی از داده‌ها به‌صورت پیوسته می‌باشد. مثال‌هایی از جریان داده، ترافیک شبکه، جستجوهای وب و داده‌های حسگر است. کاوش جریان داده به دلیل حجم زیاد داده‌ها و ماهیت متغیر آنها مشکل است و می‌تواند به‌صورت افزایشی یا غیر افزایشی انجام شود. در حالت غیر افزایشی، جریان داده‌ها به تعدادی تکه تقسیم شده که الگوریتم یادگیری روی هر تکه اجرا می‌گردد اما در حالت افزایشی هر نمونه مدل یاد گرفته شده را به روز می‌کند و سپس دور انداخته می‌شود.

با توجه به ماهیت جریان داده‌ای نظرات کاربران در شبکه‌های اجتماعی و سایت‌های تجارت الکترونیکی، استفاده از الگوریتم‌های یادگیری ماشین غیر افزایشی برای دسته‌بندی نظرات باعث می‌گردد به مرور زمان کارایی مدل یاد گرفته شده برای کاوش نظرات کاهش یافته و عملاً غیر قابل استفاده شود. علاوه بر این به دلیل نامحدودبودن تعداد نظرات، امکان برچسب‌گذاری تمام نظرات برای ایجاد نمونه‌های آموزشی جدید و به روز رسانی مدل یاد گرفته شده وجود ندارد. از آنجا که ممکن است نظرات جدید دارای واژگان جدید بوده و یا توزیع دسته‌های قطبیت تغییر کند رانش مفهوم نیز می‌بایست در نظرکاوی افزایشی پشتیبانی گردد.

## ۲- مرور کارهای گذشته

آقای Pang و همکارشان [۳] از روش‌های یادگیری ماشین بیزین ساده (NB)، بیشترین آنتروپی (ME) و ماشین بردار پشتیبان (SVM) برای دسته‌بندی دوتایی مرورهای فیلم‌ها[5] استفاده نموده‌اند. آنها برای این منظور

این مقاله در تاریخ ۱۳ خرداد ماه ۱۳۹۷ دریافت و در تاریخ ۲۹ مهر ماه ۱۳۹۷ بازنگری شد.

سیدفخرالدین نوربهبهانی (نویسنده مسئول)، گروه کامپیوتر، دانشگاه خوانسار، خوانسار، ایران، (email: f.noorbehbahani@khansar-cmc.ac.ir).

1. Lexicon-Based
2. Dictionary-Based
3. Corpus-Based
4. Ontology
5. Movies' Reviews

مرورهای فیلم‌ها را از IMDb.com جمع‌آوردی کرده و الگوریتم‌های مختلفی را آزمایش کردند که بیشترین دقت توسط ماشین بردار پشتیبان به دست آمده است. ادعای نویسندگان این مقاله آن است که تحلیل سخن، تشخیص تمرکز و حل ارجاع مشترک می‌تواند باعث بهبود دقت تعیین قطبیت شود.

آقای McDonald و همکارانشان [۴] یک مدل ساختاریافته برای دسته‌بندی احساس در سطح جمله و سطح سند به صورت مشترک ارائه نموده‌اند. فرض آنها این بوده که برچسب جملات و اسناد، وابسته به هم هستند. آنها برای استنتاج از الگوریتم Viterbi که به عنوان مدل زنجیره خطی نیز شناخته شده استفاده نموده‌اند. برای دسته‌بندی از الگوریتم MIRA استفاده نموده و برای انتخاب ویژگی از تک‌واژه، دوواژه و سه‌واژه به همراه علامت‌گذاری ادات سخن (POS)[1] استفاده شده است. در این مقاله آزمایش‌ها بر روی ۶۰۰ مرور محصولات از سایت Amazon.com در حوزه‌های "صندلی ماشین برای کودکان"، "تجهیزات تناسب اندام" و "پخش‌کننده‌های Mp۳" انجام شده است.

آقای Dang و همکارانشان [۵] احساسات را با استفاده از SVM و روش‌های مختلف انتخاب ویژگی دسته‌بندی کردند. آزمایش‌های آنها بر روی مجموعه متون شامل ۳۰۵ مرور مثبت و ۳۰۷ مرور منفی مربوط به دوربین‌های دیجیتالی و مجموعه داده Biltzer [۶] انجام شده است. الگوریتم SVM بر روی سه مجموعه شامل ویژگی‌های مستقل از دامنه، وابسته به دامنه و ویژگی‌های احساس انجام و از بهره اطلاعات برای کاهش مجموعه ویژگی‌ها استفاده شده است.

آقای Saleh و همکارانشان [۷] ۲۷ آزمایش با استفاده از SVM و روش‌های مختلف انتخاب ویژگی انجام دادند. آزمایش‌های آنها بر روی مجموعه داده‌های Pang [۸]، Taboada [۹] و SINAI انجام شده است. در [۱۰] از روش TS-MBC برای یادگیری وابستگی‌های شرطی بین کلمات استفاده شده که این وابستگی‌ها را در یک گراف بدون دور مارکف از کلمات احساس کدگذاری می‌کند. در مرحله بعد نویسنده از روش جستجوی Tabu برای تنظیم دقیق گراف برای بهبود دقت بهره می‌جوید.

در [۱۱] از روش‌های یادگیری ماشین (NB و SVM) برای دسته‌بندی مرورهای رستوران استفاده شده است. در این مقاله تأثیر نمایش‌های ویژگی‌ها و اندازه ویژگی‌ها بر کارایی دسته‌بندی، مطالعه و آزمایش‌هایی روی ۱۵۰۰ مرور مثبت و ۱۵۰۰ مرور منفی انجام شده است. نمایش‌های مختلف آزمایش‌شده شامل تک‌واژه‌ها، تک‌واژه- فرکانس، دوواژه‌ها، دوواژه‌ها- فرکانس، سه‌واژه‌ها، سه‌واژه‌ها- فرکانس و تعداد متغیری از ویژگی‌ها بین ۵۰ تا ۱۶۰۰ ویژگی بوده است.

آقای Tan و همکارانشان [۱۲] یک روش خودکار برای استخراج الگوهای تعیین قطبیت در سطح عبارت در قالب قوانین ارائه نمودند. در این مقاله، قوانین ترتیبی دسته برای یادگیری خودکار الگوهای وابستگی استفاده شده و به نظر می‌رسد در آینده باید روابط پیچیده‌تری بین عبارات در نظر گرفته شود و به تحلیل عمیق‌تری برای تأثیرات در سطح عبارات بر روی دقت کلی تعیین قطبیت نیاز است.

آقای Wang و همکارانشان [۱۳] کارایی سه روش معروف یادگیری گروهی یعنی روش Bagging، Boosting و زیرفضای تصادفی را بر اساس پنج یادگیرنده NB، ME، DT، KNN و SVM برای دسته‌بندی احساس مقایسه نموده‌اند. آزمایش‌های انجام‌شده نشانگر دقت بیشتر نسبت به یادگیرنده‌های پایه به بهای زمان محاسباتی بیشتر بوده است. در [۱۴] روش‌های SVM، NB و روش‌های مبتنی بر شبکه‌های عصبی برای دسته‌بندی احساس مقایسه شده‌اند و آزمایش‌ها بر روی مجموعه داده‌های متعادل و نامتعادل انجام شده است. چهار مجموعه داده برای این منظور انتخاب شده که شامل مرورهای فیلم، کتاب، دوربین و محصولات GPS بوده است. برای مجموعه داده‌های نامتعادل عملکرد دسته‌بندهای ANN و SVM تحت تأثیر قرار گرفته است. استفاده از بهره اطلاعات برای انتخاب ویژگی نتایج خوبی برای ویژگی‌های بیش از ۱۰۰۰ عدد ندارد، به همین دلیل باید از روش‌های دیگر انتخاب ویژگی برای آزمایش دسته‌بندها استفاده نمود.

آقای Basari و همکارانشان [۱۵] از ترکیب روش‌های PSO و SVM برای دسته‌بندی احساس مرورهای فیلم استفاده نموده‌اند. PSO برای انتخاب بهترین پارامترها جهت حل مسأله بهینه‌سازی دوگانه مورد استفاده قرار گرفته و آزمایش‌ها روی مجموعه داده EMOT انجام شده است. در آینده دسته‌بندی چندتایی احساس با استفاده از ترکیب وزن‌دهی ویژگی‌ها و $n$ واژه‌ای‌ها می‌تواند برای بهبود دسته‌بندی انجام شود. در [۱۶] از روش کاهش ویژگی نظارت‌شده با استفاده از $n$ واژه‌ای‌ها و تحلیل آماری برای ساخت واژه‌نامه جهت تحلیل احساس استفاده شده است. در این مقاله شش ویژگی اتلاف اطلاعات، پیش‌قدر، اختلال، تصادم، تغییر مقیاس و بیش‌پوشش در نظر گرفته و از شبکه عصبی مصنوعی پویا (DAN۲ و SVM) برای دسته‌بندی چندتایی استفاده شده است. DAN۲ شامل چندین لایه مخفی با چهار گره در هر لایه است و دقت بهتری نسبت به SVM به دست آورده است. روش پیشنهادی این مقاله می‌تواند با استفاده از روش‌های مبتنی بر هستی‌شناسی برای ساختن واژه‌نامه بهبود یابد.

در زمینه یادگیری فعال روی جریان نظرات پژوهش‌های محدودی انجام شده است. Samilovic و همکاران [۱۷] یک روش یادگیری فعال جریانی برای تحلیل نظرات در حوزه مالی پیشنهاد داده‌اند که با استفاده از آن تغییرات قیمت سهام پیش‌بینی می‌شود. روش پیشنهادی این مقاله استراتژی‌های مختلفی برای یادگیری فعال پیشنهاد می‌کند که دقت به دست آمده آنها پایین است (بیشترین دقت ۰/۵۴ است). علاوه بر این، روش پیشنهادی آنها فاقد مکانیزم تشخیص رانش مفهوم و تعیین بودجه برچسب‌گذاری دلخواه است.

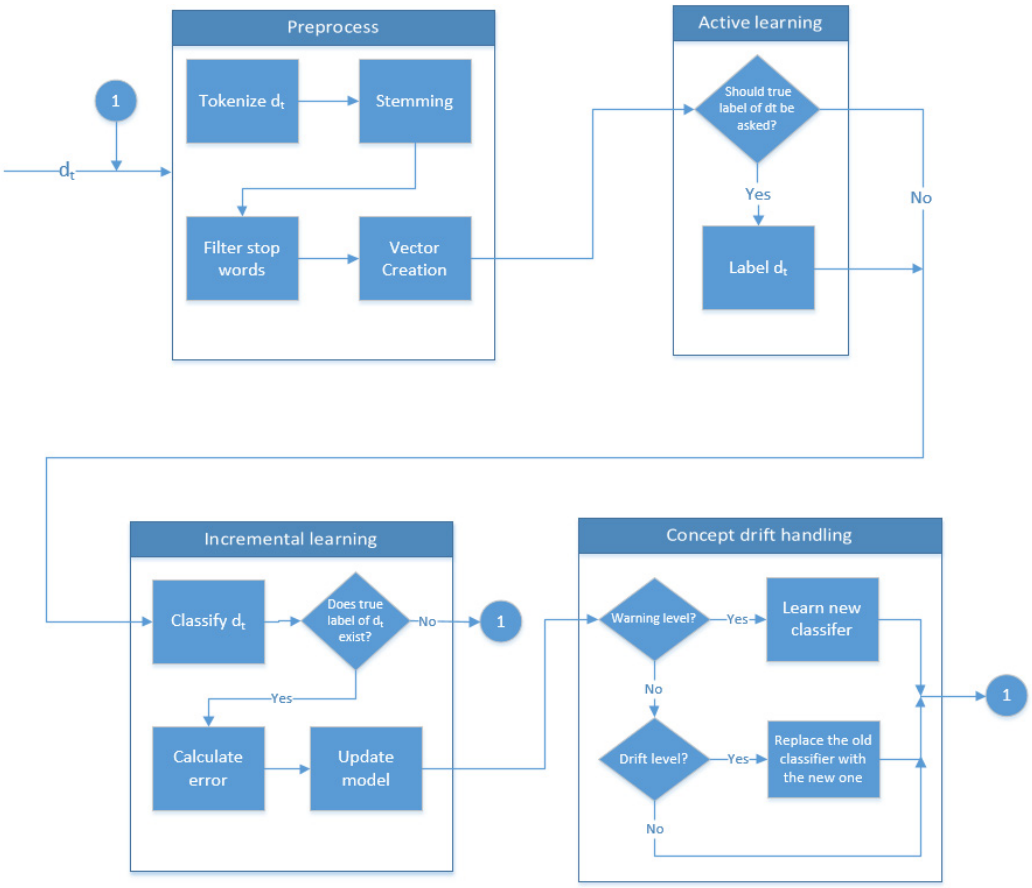
در [۱۸] دو استراتژی بهره اطلاعات و عدم قطعیت برای یادگیری فعال روی جریان نظرات، ارائه و روش پیشنهادی آنها با روش نمونه‌برداری تصادفی مقایسه شده است. گذشته از عدم انتخاب روش یادگیری فعال مناسب برای مقایسه و ارزیابی، دقت به دست آمده روش پیشنهادی آنها پایین است. در [۱۹] نیز تأثیر در دسترس بودن یا نبودن فرد خبره برای آزمون یادگیری فعال جریانی نظرات در شرایط واقعی بررسی شده و به این نتیجه رسیده‌اند زمانی که خبره به طور مداوم در دسترس نیست، استراتژی پرسش تصادفی قابل رقابت با پرسش از خبره است. همان طور که مشخص است موضوع پژوهش این مقاله با پژوهش پیش رو کاملاً متفاوت می‌باشد.

## ۳- روش پیشنهادی

در شکل ۱ روش پیشنهادی تعیین قطبیت نشان داده شده که شامل ۴ مرحله پیش‌پردازش[2]، یادگیری فعال[3]، یادگیری افزایشی[4] و مدیریت رانش

1. Part of Speech

2. Preprocess
3. Active Learning
4. Incremental Learning




شکل ۱: روندنمای روش پیشنهادی تعیین قطبیت.

مفهوم[1] می‌باشد. روش پیشنهادی افزایشی بوده و بدون ذخیره داده‌های ورودی مدل دسته‌بندی خود را به روز رسانی می‌نماید. مرحله پیش‌پردازش متن ورودی را برای پردازش‌های بعدی آماده ساخته و مرحله یادگیری فعال وظیفه تصمیم‌گیری در ارتباط با پرسش برچسب واقعی متن را بر عهده دارد. در این مرحله برچسب متونی که برای به روز رسانی مدل دسته‌بندی مفید هستند پرسیده شده و به این ترتیب در برچسب‌گذاری صرفه‌جویی می‌شود. پس از آن در مرحله یادگیری افزایشی مدل دسته‌بندی به روز شده و مرحله مدیریت رانش مفهوم وظیفه تشخیص و رسیدگی به رانش مفهوم را دارد.

در ابتدا متن مورد نظر در زمان $t$ که آن را با $d_t$ نشان می‌دهیم برای تعیین قطبیت وارد مرحله پیش‌پردازش می‌شود. در این مرحله ابتدا متن مورد نظر به واژه‌ها و عبارات تقسیم و شکسته می‌شود که این مرحله Tokenization نام دارد. پس از آن مرحله ریشه‌یابی انجام می‌شود که در این مرحله ریشه واژه‌ها به جای هر واژه قرار می‌گیرد. در قسمت بعدی مرحله پیش‌پردازش، واژه‌های کم‌ارزش و زاید که به آنها ایست‌واژه می‌گویند، نظیر حروف تعریف حذف می‌شوند. پس از آن متن به برداری از واژه‌ها تبدیل شده و مرحله پیش‌پردازش به اتمام رسیده و بردار تولیدشده به مرحله یادگیری فعال وارد می‌شود.

در مرحله یادگیری فعال در مورد این که آیا برچسب واقعی نمونه واردشده پرسیده شود یا خیر تصمیم‌گیری می‌گردد. در صورتی که برچسب واقعی نمونه پرسیده شود توسط متخصص انسانی برچسب واقعی، تعیین و پس از آن نمونه وارد مرحله یادگیری افزایشی می‌شود.

در مرحله یادگیری افزایشی ابتدا با مدل دسته‌بندی یاد گرفته شده برچسب نمونه تعیین می‌گردد. سپس در صورتی که برچسب واقعی نمونه موجود نباشد فرایند به اتمام رسیده و نمونه بعدی از ورودی گرفته می‌شود. در صورتی که برچسب واقعی نمونه موجود باشد (یعنی در مرحله یادگیری فعال پرسیده شده باشد) از مقایسه برچسب واقعی و برچسب پیش‌بینی شده خطای دسته‌بندی محاسبه می‌گردد. پس از محاسبه خطا از نمونه جدید دارای برچسب واقعی برای به روز رسانی و یادگیری مدل دسته‌بندی استفاده می‌گردد و پس از آن وارد مرحله تشخیص رانش مفهوم می‌شویم.

در مرحله تشخیص رانش مفهوم از خطای محاسبه‌شده برای تشخیص این که در سطح هشدار یا سطح رانش مفهوم هستیم استفاده می‌شود. در صورتی که به سطح هشدار رسیده باشیم یک دسته‌بند جدید با نمونه‌های جدید برچسب‌خورده به موازات دسته‌بند قدیمی یاد گرفته می‌شود. در صورتی که به سطح رانش مفهوم برسیم دسته‌بند جدید یاد گرفته شده جایگزین دسته‌بند قدیمی خواهد شد. در بخش‌های بعد مراحل روش پیشنهادی به تفصیل آمده است.

## ۳-۱ پیش‌پردازش

اولین گام از مرحله پیش‌پردازش جداسازی واژه‌ها و عبارات از متن است. Tokenizer ابزاری برای شکستن یک متن بر اساس واحدهای با معنی مانند کلمه، پاراگراف، نمادهای معنادار مانند Space و Tab و غیره است. لازمه ایجاد این ابزار جمع‌آوری واحدهایی است که در زبان به عنوان واحدهای مستقل معنایی شناخته می‌شوند. سپس بر اساس انتخاب هر کدام از این واحدها متن بر اساس آن شکسته خواهد شد. از نمونه‌های انگلیسی آن می‌توان به Flex، JLex، JFLex، ANTLR، Ragel و Quex اشاره کرد.

1. Concept Drift Handling

**Input**: labeling budget $B$, Strategy (parameters)
**Output**: at every time iteration output classifier $L$
**Initialize**: labeling expenses $\hat{b} \leftarrow 0$

**repeat**
  receive incoming instance $X$;
  **if** $\hat{b} < B$ **then**
    //budget is not exceeded
    **if** Strategy ($X$,parameters) = true **then**
      request the true label $y$ of instance $X$;
      update labeling expenses $\hat{b}$ ;
      update classifier $L$ with $(X, y)$;
      **if** $L_n$ *exists* **then**
        update classifier $L_n$ with $(X, y)$;
      **if** *change warning is signaled* **then**
        start a new classifier $L_n$;
      **if** *change is detected* **then**
        replace classifier $L$ with $L_n$;
**until** *forever*;

شکل ۲: الگوریتم یادگیری فعال [۲۰].

در گام بعدی ریشه‌یابی واژه‌ها انجام می‌شود. در این گام به منظور یکسان‌سازی اشکال مختلف یک واژه، یکپارچه‌سازی و همچنین اعمال پردازش‌های بعدی بایستی واژه‌ها ریشه‌یابی شوند. ریشه‌یابی به فرایند تبدیل واژه‌ها به فرم ریشه‌ای و پایه‌ای آنها اشاره دارد. برای مثال به جای واژه‌های "Fishing"، "Fished" و "Fisher" ریشه آنها یعنی واژه "Fish" قرار داده می‌شود. لازم به ذکر است که منظور از ریشه در این بخش، دقیقاً ریشه واژه‌ها که در زبان‌شناسی استفاده می‌شود نیست. بلکه منظور از ریشه، یک نماینده برای واژه‌هایی است که از لحاظ معنایی و نحوی در یک حوزه قرار می‌گیرند. این فرایند در پردازش متن، اهمیت بسیاری دارد چرا که باعث می‌شود در نظرکاوی با دو واژه هم‌خانواده اما ظاهراً متفاوت، مانند دو واژه‌ای که از لحاظ ریشه‌ای هیچ ارتباطی با هم ندارند برخورد نشود. الگوریتم‌های مختلفی برای ریشه‌یابی لغات پیشنهاد شده و مورد استفاده قرار می‌گیرد. الگوریتم پورتر و ریشه‌یاب WordNet از رایج‌ترین الگوریتم‌های ریشه‌یابی در زبان انگلیسی هستند.

در گام بعدی ایست‌واژه‌ها می‌بایست حذف شوند. ایست‌واژه‌ها لغاتی هستند که علی‌رغم تکرار فروان در متن، از لحاظ معنایی دارای اهمیت کمی هستند مثل "a"، "an"، "the"، "out"، "any" و غیره. در نگاه اولیه کلمات ربط و تعریف، ایست‌واژه به نظر می‌آیند، در عین حال بسیاری از افعال، افعال کمکی، اسم‌ها، قیدها و صفات نیز ایست‌واژه شناخته شده‌اند. در اغلب کاربردهای متن، حذف این کلمات، نتایج پردازش را به شدت بهبود می‌دهد و سبب کاهش بار محاسبات و افزایش سرعت خواهد شد. به همین دلیل این کلمات غالباً در مرحله پیش‌پردازش حذف می‌گردند.

در آخرین گام مرحله پیش‌پردازش بردار واژه‌ها ساخته می‌شود. برای این منظور از معیار $tf-idf$ برای وزن‌دهی واژگان یک سند استفاده می‌شود. $tf$ نشانگر فراوانی یک واژه در سند است و $idf$ معیاری است برای جریمه واژه‌هایی که در کلیه متون بسیار متداول هستند و معمولاً تکرار می‌شوند. برای وزن‌دهی هر واژه در متن حاصل‌ضرب $tf \times idf$ استفاده می‌شود. طریقه به دست آورن $idf$ بدین صورت است که از لگاریتم، تقسیم تعداد کل متون بر تعداد متون شامل کلمه متداول به دست می‌آید. برای مثال فرض کنیم در کل پایگاه داده ۱۰۰۰ متن وجود دارد. اگر در هر ۱۰۰۰ تای آن یک واژه خاص مثلاً book وجود داشته باشد حاصل، لگاریتم ۱۰۰۰ تقسیم بر ۱۰۰۰ می‌شود که برابر با صفر است. یعنی حتماً این واژه جزء واژه‌های متداول بوده و باید ضریب صفر بگیرد. در صورتی که تکرار در ۵۰۰ متن اتفاق افتاده باشد می‌شود لگاریتم ۲ که حاصل ۱ است و $idf$ برابر با ۱ می‌شود. هرچه تعداد متونی که واژه در آن تکرار شده بیشتر باشد $idf$ کوچک‌تر می‌شود و چون ممکن است اصلاً تکرار نشده باشد و مخرج صفر شود در مخرج +۱ اضافه می‌شود.

## ۳-۲ یادگیری فعال

هدف از یادگیری فعال انتخاب نمونه‌هایی است که ارزش اطلاعاتی بالایی دارند و باعث بهبود یادگیری و دقت دسته‌بند شوند. پس از انتخاب نمونه‌های ارزشمند برچسب صحیح آنها توسط مکانیزم برچسب‌گذاری که می‌تواند یک فرد خبره باشد به دست می‌آید و سپس این نمونه به همراه برچسب به عنوان نمونه آموزشی مورد استفاده قرار می‌گیرد.

همان طور که در [۲۰] بیان شده است یادگیری فعال برای داده‌های جریانی به‌دلیل تغییر توزیع نمونه‌ها به مرور زمان و رانش مفهوم بسیار چالش‌انگیز است. در [۲۰] سه نیازمندی برای مؤثربودن یادگیری فعال برای داده‌های جریانی ذکر شده است. اولین نیازمندی بیانگر آن است که راهکار پرسش باید به گونه‌ای باشد که تعادلی در طول زمان برای برچسب‌گذاری ایجاد کند به نحوی که در هر بازه زمانی تعداد پرسش‌ها از بودجه برچسب‌گذاری $B \in [0,1]$ بیشتر نشود. بودجه برچسب‌گذاری $B$، درصد نمونه‌هایی است که برچسب واقعی آنها درخواست می‌شود. دومین نیازمندی بیانگر آن است که راهکار یادگیری فعال باید به نحوی باشد که از کل فضای نمونه‌ها پرسش کرده تا توانایی وفق پیداکردن با تغییراتی را که در هر نقطه از فضای نمونه‌ها ممکن است به وجود آید داشته باشد. نیازمندی سوم دلالت بر این دارد که راهکار یادگیری فعال باید توزیع نمونه‌های ورودی را حفظ کرده تا بتواند تغییرات را رصد کرده و آنها را تشخیص دهد. نیازمندی‌های ذکرشده به‌صورت رسمی در ادامه بیان شده است.

**نیازمندی ۱:** برچسب‌گذاری در هر زمانی باید از بودجه برچسب‌گذاری تخطی نکند. به بیان دیگر $\sum_{x_t \in D} P(labeling = True|x_t)P(x_t) \le B$، که در آن $P(labeling = True|x_t)$ احتمال برچسب‌گذاری نمونه ورودی و $P(x_t)$ چگالی احتمال داده‌ها است.

**نیازمندی ۲:** برای هر نمونه ورودی $x_t \in D$ احتمال برچسب‌گذاری نباید صفر باشد و به عبارت دیگر $P(labeling = True|x_t) > 0$.

**نیازمندی ۳:** چگالی احتمال داده‌های برچسب‌خورده باید با چگالی احتمال داده‌های برچسب‌نخورده برابر باشد و به بیان دیگر $P(x_t|labeling = True) = P(x_t)$.

در [۲۰]، ۴ استراتژی برای یادگیری فعال ذکر گردیده که نشان داده شده است استراتژی چهارم با نام استراتژی تقسیم، تمامی نیازمندی‌های ذکرشده را برطرف می‌نماید. در استراتژی تقسیم، جریان داده به صورت تصادفی به دو جریان تقسیم می‌شود. یکی از جریان‌ها بر اساس استراتژی عدم قطعیت و جریان دیگر به صورت تصادفی برچسب‌گذاری می‌شود. هر دو جریان برای آموزش دسته‌بند استفاده می‌شوند ولی برای تشخیص رانش مفهوم تنها جریان تصادفی استفاده می‌گردد. در شکل ۲ الگوریتم یادگیری فعال آورده شده است.

در الگوریتم فوق ابتدا بودجه برچسب‌گذاری و پارامترهای دیگر استراتژی یادگیری فعال به عنوان ورودی گرفته می‌شود و خروجی در هر تکرار یک دسته‌بند است. الگوریتم به این صورت عمل می‌کند که ابتدا نمونه ورودی وارد شده و سپس اگر بودجه برچسب‌گذاری کاملاً مصرف نشده است از استراتژی مورد نظر برای تصمیم‌گیری در مورد پرسش یا عدم پرسش برچسب نمونه استفاده می‌گردد. اگر قرار بر پرسش بود، برچسب واقعی نمونه به دست می‌آید و سپس با استفاده از این برچسب

**Input**: incoming instance $X_t$, trained classifier $L$, threshold adjustment step $s \in (0, 1]$, proportion of random labeling $v \in (0, 1)$, budget $B$
**Output**: *labeling* $\in$ {true, false}
**Initialize**: initialize labeling threshold $\theta \leftarrow 1$ and store the latest value during operation

1 **if** $\eta < v$, *where* $\eta \sim U[0, 1]$ *is random* **then**
2  **if** *change detected* **then**
3  cut the training window
4  **return** *labeling* $\leftarrow$ RANDOM($B$)
5
6 **else**
7  **return** *labeling* $\leftarrow$ VARUNCERTAINTY($X_t, L, s$)

شکل ۳: الگوریتم استراتژی تقسیم [۲۰].

**Input**: labeling budget $B$
**Output**: *labeling* $\in$ {true, false} indicates whether to request the true label $y_t$ for $X_t$

1 generate a uniform random variable $\xi \in [0, 1]$;
2 **return** *labeling* $\leftarrow \mathbf{1}(\xi \leq B)$

شکل ۴: استراتژی یادگیری فعال تصادفی [۲۰].

دسته‌بند فعلی $(L)$ به روز می‌شود و در صورتی که دسته‌بند جدید $(L_n)$ هم موجود باشد این دسته‌بند نیز به روز می‌شود. سپس اگر رانش مفهوم در سطح هشدار اتفاق افتاده باشد شروع به یادگیری یک دسته‌بند جدید می‌شود و اگر رانش مفهوم اتفاق بیفتد دسته‌بند جدید جایگزین دسته‌بند قدیمی می‌گردد.

در خط ۳، $\hat{b}$ بودجه برچسب‌گذاری تخمین زده شده که توسط رابطه‌ای که در [۲۰] پیشنهاد شده است در خط ۶ به روز می‌گردد.

در خط ۴ الگوریتم یادگیری فعال از استراتژی تقسیم که در ادامه آمده برای برچسب‌گذاری استفاده می‌شود. خطوط ۱۰ و ۱۲ مربوط به رانش مفهوم می‌باشد که در قسمت ۳-۴ شرح داده می‌شود.

در شکل ۳ الگوریتم استراتژی تقسیم آمده است. استراتژی تقسیم به این صورت کار می‌کند که جریان ورودی به دو قسمت تقسیم می‌شود. یک قسمت برچسب‌ها توسط استراتژی تصادفی و قسمت دوم از استراتژی عدم قطعیت متغیر برچسب‌ها به دست می‌آید. قسمت اول در واقع برای تشخیص رانش مفهوم ضرورت دارد و قسمت دوم سعی می‌کند از استراتژی مناسب یادگیری فعال استفاده کند.

الگوریتم‌های مربوط به توابع خطوط ۴ و ۷ در شکل‌های ۴ و ۵ آورده شده است.

## ۳-۳ یادگیری افزایشی

در یادگیری افزایشی با جریان داده‌ها سروکار داریم که هر نمونه برچسب‌خورده از جریان داده، مدل دسته‌بندی را به‌روز کرده و سپس دور انداخته می‌شود. بدین ترتیب مشکل حجم عظیم داده‌ها که ذخیره‌سازی آنها امکان‌پذیر نیست حل می‌گردد. برای یادگیری افزایشی روش‌های گوناگونی وجود دارد که روش بیزی ساده در دسته‌بندی متون کارایی خوبی دارد. بنابراین در روش پیشنهادی از این روش دسته‌بند استفاده می‌گردد که در ادامه شرح داده می‌شود.

یک روش بسیار کاربردی یادگیری بیزی، روش یادگیرنده ساده بیزی است که در برخی زمینه‌ها نشان داده شده که کارایی آن قابل قیاس با کارایی روش‌هایی مانند شبکه عصبی و درخت تصمیم است. دسته‌بندی ساده بیزی برای مسایلی که هر نمونه $x$ در آن توسط مجموعه‌ای از

**Input**: incoming instance $X_t$, trained classifier $L$, threshold adjustment step $s \in (0, 1]$
**Output**: *labeling* $\in$ {true, false}
**Initialize**: initialize labeling threshold $\theta \leftarrow 1$ and store the latest value during operation

1 $\hat{y}_t \leftarrow \arg\max_y P_L(y|X_t)$, where $y \in \{1, \ldots, c\}$;
2 **if** $P_L(\hat{y}_t|X_t) < \theta$ **then**
  //uncertainty below the threshold
3  decrease the uncertainty region $\theta \leftarrow \theta(1 - s)$;
4  **return** *labeling* $\leftarrow$ true
5
6 **else**
  //certainty is good
7  make the uncertainty region wider $\theta \leftarrow \theta(1 + s)$;
8  **return** *labeling* $\leftarrow$ false

شکل ۵: استراتژی یادگیری فعال عدم قطعیت متغیر [۲۰].

مقادیر ویژگی‌ها و تابع هدف $f(x)$ از مجموعه‌ای مانند $V$ انتخاب می‌گردد (یعنی $|V|$ دسته وجود دارد) کاربرد دارد. روش کار بیزی برای دسته‌بندی نمونه جدید این است که محتمل‌ترین دسته یا مقدار هدف $v_{MAP}$ را با داشتن مقادیر ویژگی‌ها $\langle a_1, a_2, \ldots, a_n \rangle$ که توصیف‌کننده نمونه جدید است شناسایی کند، $v_{MAP} = \arg\max_{v_j \in V} P(v_j | a_1, a_2, \ldots, a_n)$.

با استفاده از قضیه بیز می‌توان عبارت بالا را به صورت زیر بازنویسی کرد

$$v_{MAP} = \arg\max_{v_j \in V} \frac{P(a_1, a_2, \ldots, a_n | v_j) P(v_j)}{P(a_1, a_2, \ldots, a_n)} = \arg\max_{v_j \in V} P(a_1, a_2, \ldots, a_n | v_j) P(v_j) \quad (۱)$$

حال با استفاده از داده‌های آموزشی سعی می‌کنیم دو جمله معادله بالا را تخمین بزنیم. محاسبه از روی داده‌های آموزشی به این صورت که میزان تکرار $v_j$ در داده‌ها چقدر است آسان می‌باشد. اما محاسبه جملات مختلف $P(a_1, a_2, \ldots, a_n | v_j)$ به این صورت قابل قبول نخواهد بود مگر این که حجم بسیار بسیار زیادی از داده‌های آموزشی در اختیار داشته باشیم. مشکل اینجاست که تعداد این جملات برابر تعداد نمونه‌های ممکن ضرب در تعداد مقادیر تابع هدف می‌باشد. بنابراین باید هر نمونه را چندین بار مشاهده کنیم تا تخمین مناسبی از آن به دست آید.

فرض روش دسته‌بندی ساده بیزی بر اساس این ساده‌سازی است که مقادیر ویژگی‌ها با داشتن مقادیر تابع هدف از یکدیگر مستقل شرطی می‌باشند. به عبارت دیگر، این فرض بیانگر آن است که به شرط مشاهده خروجی تابع هدف احتمال مشاهده ویژگی‌های $a_1, a_2, \ldots, a_n$ برابر ضرب احتمالات هر صفت به طور جداگانه می‌باشد. اگر این مقدار را جایگزین معادله بالا کنیم روش دسته‌بندی ساده بیزی را نتیجه می‌دهد

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \quad (۲)$$

که $v_{NB}$ خروجی دسته‌بندی ساده بیزی برای تابع هدف می‌باشد. تعداد جملات $P(a_i | v_j)$ که در این روش باید محاسبه شوند برابر تعداد ویژگی‌ها ضرب در تعداد دسته‌های خروجی برای تابع هدف می‌باشد که این مقدار از تعداد جملات $P(a_1, a_2, \ldots, a_n | v_j)$ بسیار کمتر است.

نتیجه این که یادگیری ساده بیزی سعی در تخمین مقادیر مختلف $P(v_j)$ و $P(a_i | v_j)$ با استفاده از میزان تکرار آنها در داده‌های آموزشی دارد. این مجموعه تخمین‌ها متناظر با فرض یاد گرفته شده و سپس از این فرض برای دسته‌بندی نمونه‌های جدید استفاده می‌شود.

برای محاسبه $P(a_1,a_2,\ldots,a_n|v_j)$، روش یادگیری بیزی ساده چندجمله‌ای از توزیع چندجمله‌ای استفاده می‌نماید و برای دسته‌بندی متون کارایی بیشتری نسبت به روش بیزی ساده معمولی دارد [۲۱]. برای مسأله دسته‌بندی متون، اگر هر متن را مجموعه‌ای از واژه‌ها در نظر بگیریم و هر واژه را به عنوان یک ویژگی لحاظ کنیم، در این صورت یک متن را می‌توان با مجموعه ویژگی‌های $a_1,a_2,\ldots,a_n$ نمایش داد که مقدار هر ویژگی فرکانس وقوع آن واژه در متن است. حال با استفاده از توزیع دوجمله‌ای می‌توان $P(a_1,a_2,\ldots,a_n|v_j)$ را از طریق (۳) و (۴) به دست آورد

$$P(a_1,a_2,\ldots,a_n|v_j)=\frac{(\sum_i a_i)!}{\prod_i (a_i!)}\prod_i (P(a_i|v_j))^{a_i} \qquad (۳)$$

و سپس با استفاده از $v_{MAP}=\arg\max_{v_j\in V} P(v_j|a_1,a_2,\ldots,a_n)P(v_j)$ دسته نمونه جدید را به دست آورد.

روش دسته‌بندی ذکرشده هنگامی که یک متن با بردار ویژگی‌های فرکانسی تعریف شود قابل استفاده است یعنی مقادیر ویژگی‌ها عدد صحیح مثبت باشد. حال اگر مقادیر ویژگی‌ها پیوسته و عددی باشد مثلاً از معیار $tf-idf$ برای نمایش هر متن استفاده نماییم آیا می‌توان از روش دسته‌بندی بیزی ساده چندجمله‌ای استفاده نمود؟ پاسخ این پرسش مثبت است و روشی در [۲۲] پیشنهاد شده که کارایی بالایی در دسته‌بندی متون دارد. در این روش دسته‌بند ساده بیزی چندجمله‌ای در فضای لگاریتمی به صورت یک دسته‌بند خطی بیان شده است. بدین ترتیب داریم

$$\begin{aligned}&\log(P(v_j|a_1,a_2,\ldots,a_n))\propto\\&\log(P(v_j)\prod_i (P(a_i|v_j))^{a_i})=\\&\log(P(v_j))+\sum_{i=1}^{n} a_i\log(P(a_i|v_j))\end{aligned} \qquad (۴)$$

و می‌توان از رابطه $v_{MAP}=\arg\max_{v_j\in V}\log[P(v_j|a_1,a_2,\ldots,a_n)]$ دسته یک نمونه را به دست آورد.

برای محاسبه $P(a_i|v_j)$ ممکن است مقدار یکی از ویژگی‌های متن صفر باشد و در این صورت لگاریتم تعریف نشده است. برای رفع این مشکل از اصلاح لاپلاسی استفاده می‌شود و $P(a_i|v_j)$ به صورت زیر محاسبه می‌گردد

$$P(a_i|v_j)=\frac{N(a_i,v_j)+\lambda_i}{N(v_j)+\lambda} \qquad (۵)$$

که $N(a_i,v_j)$ تعداد دفعاتی است که واژه $i$ در متون دسته $v_j$ ظاهر شده و $N(v_j)$ تعداد کل واژه‌های دسته $v_j$ است. $\lambda_i$ مقدار اضافه‌شونده برای واژه $i$ و $\lambda$ مجموع $\lambda_i$ هاست. معمولاً برای تمام واژه‌ها $\lambda_i=1$ در نظر گرفته می‌شود.

### ۳-٤ تشخیص و مدیریت رانش مفهوم

رانش مفهوم هنگامی رخ می‌دهد که تابع زمینه‌ای که نمونه‌ها را تولید می‌کند در طول زمان ثابت نبوده و تغییر کند. عوامل مختلفی وجود دارد که منجر به رانش مفهوم می‌شود. دسته‌بندی نمونه $x_t$ را در نظر بگیرید. برای دسته‌بندی صحیح نمونه $x_t$ به اطلاعات احتمالات اولیه مشاهده دسته‌ها، $P(v_i)$، و احتمال شرطی مشاهده $x_t$ با فرض هر دسته، $P(x_t|v_i)$، نیاز داریم. با داشتن این اطلاعات با استفاده از تئوری بیز احتمال آن که $x_t$ نمونه‌ای از دسته $v_i$ باشد را می‌توان توسط (۶) به دست آورد

$$P(v_i|x_t)=\frac{P(v_i)P(x_t|v_i)}{P(x_t)} \qquad (۶)$$

در (۶)، $P(x_t)$ احتمال مشاهده $x_t$ است که تغییر آن (تغییر توزیع نمونه‌ها) باعث رانش مفهوم می‌شود. از طرفی تغییر در $P(v_i)$ و $P(x_t|v_i)$ نیز منجر به رانش مفهوم می‌گردد. تغییر در احتمالات اولیه دسته‌ها به منزله تغییر وضعیت نامتعادل‌بودن دسته‌ها است [۲۳]، برای مثال ممکن است دسته اقلیت در طول زمان به دسته اکثریت تبدیل شود. تغییر در احتمال شرطی دسته‌ها به دلیل تغییر مرزهای دسته‌بندی از علل رایج رانش مفهوم است.

روش‌های برخورد با رانش مفهوم به دو دسته روش‌های مبتنی بر راه‌اندازی و روش‌های تکاملی تقسیم می‌شوند. روش‌های راه‌اندازی از یک تشخیص‌دهنده صریح برای مشخص‌کردن زمان به روز رسانی مدل استفاده می‌کنند. از طرف دیگر روش‌های تکاملی، رانش مفهوم را تشخیص نمی‌دهند بلکه با به روز رسانی دوره‌ای با رانش مفهوم مقابله می‌کنند. روش‌های تکاملی دقت بهتری دارند اما نیاز به زمان و حافظه بیشتری دارند و به همین منظور با توجه به اهمیت زمان در تعیین قطبیت متون، از روش‌های راه‌اندازی DDM استفاده می‌نماییم.

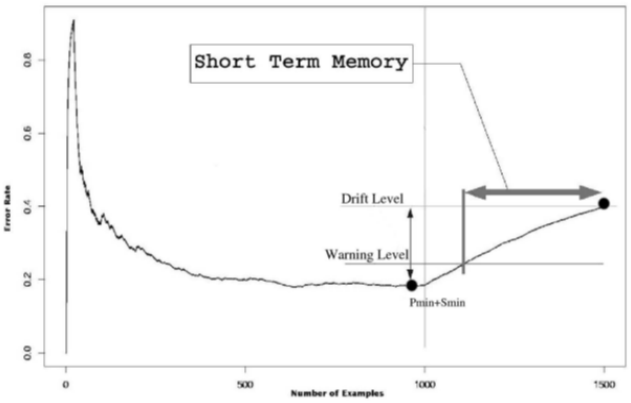
روش تشخیص رانش مفهوم DDM از نرخ خطای کل برای تشخیص رانش مفهوم استفاده می‌کند [۲۴]. در مدل یادگیری احتمالاً تقریباً درست (PAC) فرض بر آن است که در صورت ثابت‌بودن توزیع نمونه‌ها، نرخ خطای الگوریتم یادگیری با افزایش تعداد مثال‌های آموزشی کاهش می‌یابد. DDM برای سناریوهای یادگیری افزایشی طراحی شده و بر این باور است که افزایش قابل توجه نرخ خطا و تغییر در توزیع دسته‌ها نشان می‌دهد مدل یادگرفته‌شده مناسب نیست. DDM در حین فرایند یادگیری نرخ خطای $p_t^e$ و انحراف معیار خطا، $s_t^e$، برای نمونه $x_t$ را محاسبه می‌کند. انحراف معیار خطا از (۷) محاسبه می‌شود

$$s_t^e=\sqrt{\frac{p_t^e(1-p_t^e)}{t}} \qquad (۷)$$

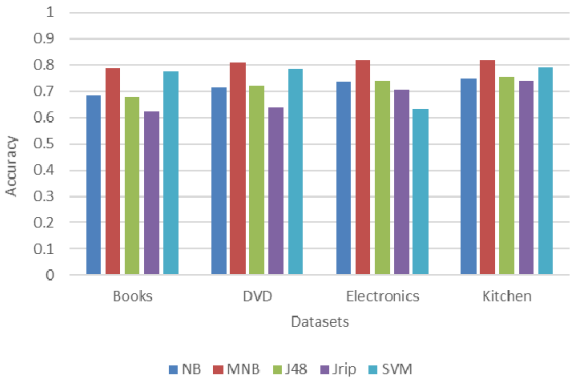
DDM در حین یادگیری از دو مقدار $p_{\min}^e$ و $s_{\min}^e$ برای تشخیص رانش مفهوم استفاده می‌نماید. هر گاه با ورود نمونه جدید $x_t$، مقدار $p_t^e+s_t^e$ کمتر از $p_{\min}^e+s_{\min}^e$ شود، مقادیر $p_{\min}^e$ و $s_{\min}^e$ به روز می‌شود. این الگوریتم هنگامی که $p_t^e+s_t^e\geq p_{\min}^e+2s_{\min}^e$ هشدار می‌دهند (سطح اطمینان ۹۵٪) و هنگامی که $p_t^e+s_t^e\geq p_{\min}^e+3s_{\min}^e$ (سطح اطمینان ۹۹٪) رانش مفهوم را تشخیص می‌دهد. وقتی DDM هشدار می‌دهد مدل جدیدی آموزش دیده می‌شود و هنگامی که رانش مفهوم تشخیص داده شده مدل جدید جایگزین مدل قبلی خواهد شد. DDM نیاز به بروز حداقل ۳۰ خطای دسته‌بندی برای تشخیص رانش مفهوم دارد. در شکل ۶ مفهوم روش DDM نشان داده شده است.

## ٤- ارزیابی روش پیشنهادی

از مجموعه داده "Multi-Domain Sentiment Dataset" برای ارزیابی استفاده گردیده که توسط آقای Blitzer و همکارانشان ارائه شده است [۶]. این مجموعه داده شامل ۴ مجموعه داده است که از مرورهای گرفته‌شده از سایت Amazon.com بر روی محصولات مختلف شامل "کتاب‌ها"، "DVD"ها، "لوازم الکتریکی" و "لوازم آشپزخانه" به دست




شکل ۶: تشخیص رانش مفهوم توسط روش DDM.




شکل ۷: دقت‌های به دست آمده از دسته‌بندهای مختلف روی ۴ مجموعه داده.

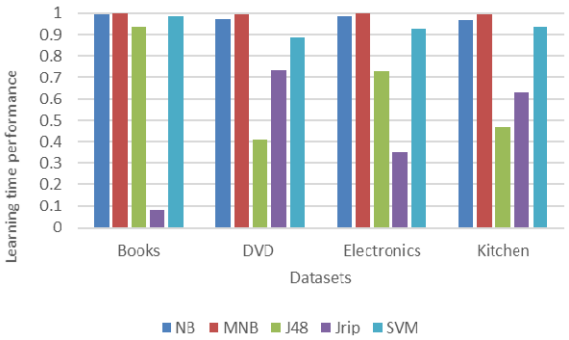
آمده است. در هر مجموعه داده، هر مرور شامل یک رتبه‌بندی بین ۰ تا ۵ است که مرورهای دارای رتبه بیشتر ۳ به عنوان مرورهای مثبت و مرورهای با رتبه کمتر یا مساوی با ۳ به عنوان مرورهای منفی برچسب خورده‌اند. به این ترتیب هر مجموعه داده شامل ۱۰۰۰ مرور مثبت و ۱۰۰۰ مرور منفی است. مجموعه داده دیگری که برای آزمایش روش تشخیص رانش مفهوم استفاده شده است، "Polarity dataset v۲.۰" می‌باشد که شامل ۱۰۰۰ مرور مثبت و ۱۰۰۰ مرور منفی است [۸].

برای ارزیابی روش پیشنهادی نظرکاوی، ابتدا روش‌های غیر افزایشی را بر روی مجموعه داده‌ها آزمایش کرده و بهترین روش را انتخاب می‌نماییم. سپس کارایی روش پیشنهادی که افزایشی است با در نظر گرفتن تمام داده‌های برچسب‌خورده، در مقایسه با روش‌های افزایشی و بهترین روش غیر افزایشی ارزیابی می‌شود. سپس روش یادگیری فعال و تشخیص رانش مفهوم پیشنهادی را اعمال و نتایج را بررسی می‌نماییم. برای اجرای تمام آزمایش‌ها از رایانه‌ای با مشخصات سیستم عامل Microsoft Windows ۷ Ultimate ۶۴-bit، میزان حافظه رم ۴ GB و پردازشگر Intel(R) Core(TM) i۵-۴۷۰UM ۱٫۸۶GHz استفاده شده است.

## ٤-١ ارزیابی روش‌های غیر افزایشی

برای آزمایش روش‌های غیر افزایشی از نرم‌افزار Weka [۲۵] استفاده شده است. به منظور ارزیابی جامع برای انتخاب روش‌های آزمایش‌شده از هر کدام از گروه‌های روش‌های یادگیری ماشین الگوریتم‌های کارامد انتخاب شده است.

روش‌های دسته‌بندی آزمایش‌شده عبارتند از روش بیزی ساده (NB) [۲۶]، روش بیزی ساده چندجمله‌ای (MNB) [۲۱] از گروه روش‌های یادگیری بیزی، روش درخت تصمیم‌گیری (J۴۸) [۲۷] از گروه روش‌های




شکل ۸: مقایسه زمان اجرای دسته‌بندهای مختلف.

مبتنی بر درخت، روش دسته‌بندی RIPPER (Jrip) [۲۸] از گروه روش‌های مبتنی بر قانون و روش ماشین بردار پشتیبان (LibSVM) [۲۹] از گروه روش‌های یادگیری تابع.

در این بخش دقت به دست آمده توسط روش‌های مختلف دسته‌بندی روی ۴ مجموعه داده مقایسه می‌گردد. برای این منظور از ارزیابی متقاطع $k$تایی استفاده می‌شود که یک روش ارزیابی استاندارد برای دسته‌بندی می‌باشد. معمولاً از ارزیابی متقاطع ۱۰تایی استفاده می‌شود که کارایی قابل قبولی در پژوهش‌های مختلف داشته است. در ارزیابی متقاطع $k$تایی مجموعه داده به $k$ قسمت مختلف تقسیم شده و سپس هر بار $k-1$ قسمت از مجموعه داده برای آموزش و ۱ قسمت باقیمانده برای آزمایش استفاده می‌شود. بدین ترتیب $k$ دقت به دست می‌آید و دقت کلی را می‌توان میانگین $k$ دقت در نظر گرفت. در شکل ۷ دقت‌های به دست آمده توسط الگوریتم‌های مختلف دسته‌بندی با استفاده از روش ارزیابی متقاطع ۱۰تایی نشان داده شده است.

در شکل ۷ دقت‌های به دست آمده از دسته‌بندهای مختلف روی ۴ مجموعه داده نشان داده شده است. همان طور که مشخص است دقت به دست آمده برای روش دسته‌بندی MNB در تمام موارد از همه بیشتر بوده و پس از آن می‌توان گفت الگوریتم SVM در رتبه دوم قرار داشته است. به نظر می‌رسد بدترین روش Jrip بوده که کمترین دقت‌ها را داشته است.

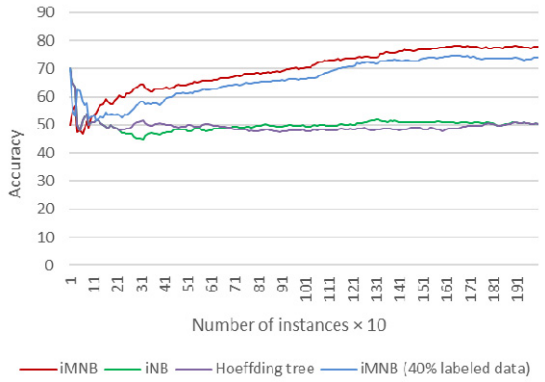
علاوه بر دقت، زمان اجرای دسته‌بندها نیز با هم مقایسه گردیده که در شکل ۸ نشان داده شده است. برای نرمال‌سازی زمان اجرا از رابطه $1-t_i/\sum_j t_j$ استفاده شده تا مقادیر بین ۰ و ۱ باشد. در این رابطه $t_i$ زمان اجرای الگوریتم دسته‌بندی $i$ام است.

شکل ۸ نشان می‌دهد بهترین زمان اجرا متعلق به روش MNB است و بدترین زمان را Jrip دارد. از آزمایش‌هایی که در این بخش انجام شده است می‌توان نتیجه گرفت بهترین روش دسته‌بندی غیر افزایشی برای نظرکاوی چه از نظر دقت و چه از نظر زمان اجرا MNB است. علت بالابودن دقت MNB تبعیت واژگان متنی از توزیع چندجمله‌ای است و همین باعث شده تا این روش غالباً در دسته‌بندی متون مورد استفاده قرار گیرد.

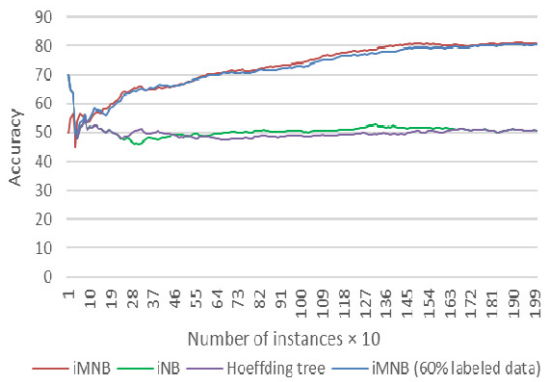
## ٤-٢ آزمایش روش‌های افزایشی

در این بخش روش‌های افزایشی نظرکاوی آزمایش و ارزیابی می‌شود. روش‌های iNB، iMNB که نسخه افزایشی روش‌های NB و MNB است به همراه روش یادگیری درخت Hoeffding که یکی از الگوریتم‌های کارا در زمینه دسته‌بندی افزایشی است و روش iMNB با استفاده از یادگیری فعال مورد بررسی قرار گرفته‌اند.

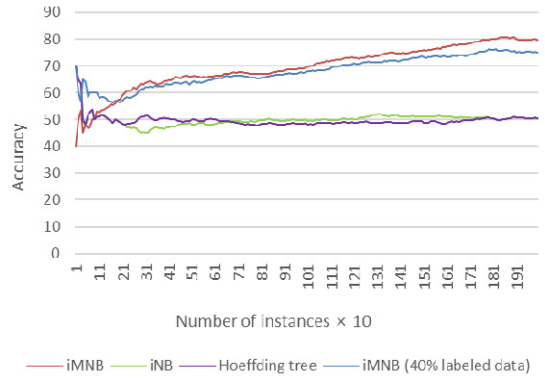
در شکل ۹ مقدار دقت‌های به دست آمده روش‌های افزایشی بر روی مجموعه داده Books نشان داده شده است. همان طور که در شکل



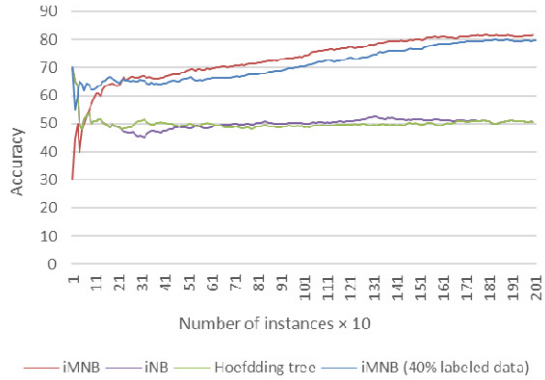

شکل ۹: مقدار دقت‌های به دست آمده روش‌های افزایشی بر روی مجموعه داده Books.




شکل ۱۱: مقدار دقت‌های به دست آمده روش‌های افزایشی بر روی مجموعه داده Electronics.




شکل ۱۰: مقدار دقت‌های به دست آمده روش‌های افزایشی بر روی مجموعه داده DVD.




شکل ۱۲: مقدار دقت‌های به دست آمده روش‌های افزایشی بر روی مجموعه داده Kitchen.

جدول ۱: مقدار میانگین دقت‌های به دست آمده برای روش‌های آزمایش‌شده.

| دسته‌بند | میانگین دقت | | | |
|---|---|---|---|---|
| | Books | DVD | Electronics | Kitchen |
| iMNB | ۶۹٫۶۳ | ۶۹٫۵۴ | **۷۲٫۸۷** | **۷۳٫۰۵** |
| iNB | ۴۹٫۹۵ | ۵۰٫۰۳ | ۵۰٫۶۸ | ۵۰٫۳۸ |
| Hoeffding Tree | ۴۹٫۴۴ | ۴۹٫۶۶ | ۴۹٫۸۹ | ۵۰٫۰۷ |
| iMNB+AL | **۶۶٫۵۰** | **۶۸٫۰۶** | ۷۲٫۳۲ | ۷۱٫۲۳ |

جدول ۲: مقدار میانگین زمان اجرای الگوریتم‌های افزایشیِ آزمایش‌شده بر حسب ثانیه.

| دسته‌بند | زمان اجرا | | | |
|---|---|---|---|---|
| | Books | DVD | Electronics | Kitchen |
| iMNB | **۱۰۹** | **۹۹** | **۴۴** | **۴۴** |
| iNB | ۲۴۷ | ۲۲۰ | ۱۰۴ | ۱۰۲ |
| Hoeffding Tree | ۲۷۷ | ۲۵۳ | ۱۱۵ | ۱۱۲ |
| iMNB+AL | **۱۱۱** | **۹۸** | **۴۶** | **۴۶** |

مشخص است بهترین دقت مربوط به iMNB و iMNB با استفاده از یادگیری فعال است. با استفاده از تنها ۴۰٪ داده‌های برچسب‌خورده به دقت معادل استفاده از کل داده‌های برچسب‌خورده رسیده‌ایم.

در شکل ۱۰ مقدار دقت به دست آمده روش‌های افزایشی بر روی مجموعه داده DVD نشان داده شده است. همان طور که مشخص است بهترین دقت مربوط به iMNB و iMNB با استفاده از یادگیری فعال است. با استفاده از تنها ۴۰٪ داده‌های برچسب‌خورده به دقت معادل استفاده از کل داده‌های برچسب‌خورده رسیده‌ایم.

در شکل ۱۱ مقدار دقت‌های به دست آمده روش‌های افزایشی بر روی مجموعه داده Electronics آمده و همان طور که مشخص است بهترین دقت مربوط به iMNB و iMNB با استفاده از یادگیری فعال است. با استفاده از ۶۰٪ داده‌های برچسب‌خورده به دقت معادل استفاده از کل داده‌های برچسب‌خورده رسیده‌ایم.

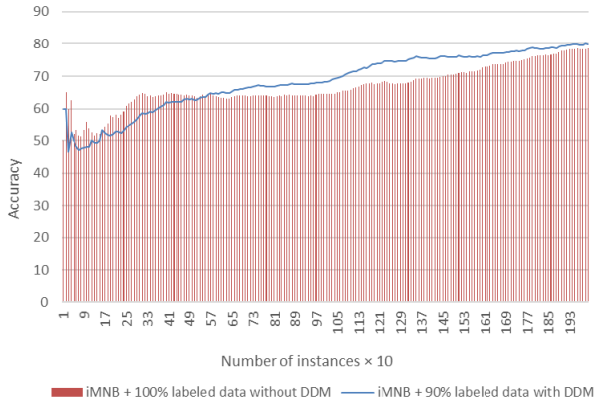
در شکل ۱۲ مقدار دقت‌های به دست آمده روش‌های افزایشی بر روی مجموعه داده Kitchen نشان داده شده است. همان طور که مشخص است بهترین دقت مربوط به iMNB و iMNB با استفاده از یادگیری فعال است. با استفاده از تنها ۴۰٪ داده‌های برچسب‌خورده به دقت معادل استفاده از کل داده‌های برچسب‌خورده رسیده‌ایم.

در جدول ۱ مقدار میانگین دقت‌های به دست آمده برای روش‌های آزمایش‌شده آمده است (مقادیر پررنگ‌شده بهترین دقت‌های به دست آمده را نشان می‌دهند).

همان طور که در جدول ۱ مشاهده می‌شود دقت‌های iMNB و iMNB با استفاده از یادگیری فعال (iMNB+AL) اختلاف ناچیزی (کمتر از ۰٫۰۴) داشته و از سایر روش‌ها در دقت پیشی گرفته‌اند.

در جدول ۲ مقدار میانگین زمان اجرای الگوریتم‌های افزایشی آزمایش‌شده آمده است (مقادیر پررنگ‌شده بهترین زمان‌های اجرا را نشان می‌دهند). همان طور که مشاهده می‌شود iMNB و iMNB+AL اختلاف ناچیزی (کمتر از ۴ ثانیه) داشته و از سایر روش‌ها در زمان اجرا پیشی گرفته‌اند.

همان طور که انتظار می‌رفت به دلیل کارایی بالای روش MNB در زمینه دسته‌بندی متون از نظر دقت و زمان اجرا، نسخه افزایشی این روش نیز بهترین کارایی را در مقایسه با روش‌های آزمایش‌شده دارد. استفاده از روش پیشنهادی نظرکاوی افزایشی با استفاده از یادگیری فعال توانسته کارایی مشابه با روش iMNB داشته باشد ضمن این که از تعداد کمتری




شکل ۱۳: مقایسه دقت‌های به دست آمده بدون/ با استفاده از DDM.

داده برچسب‌خورده استفاده شده است.

برای نشان‌دادن تأثیر روش تشخیص و مدیریت رانش مفهوم از مجموعه داده polarity dataset v۲.۰ استفاده نمودیم [۸]. این مجموعه داده شامل ۱۰۰۰ مرور مثبت و ۱۰۰۰ مرور منفی است. برای این آزمایش یک بار iMNB را با استفاده از کل داده‌های برچسب‌خورده و یک بار iMNB + AL را با بودجه برچسب‌گذاری ۱۰۰٪ و استفاده از روش تشخیص و مدیریت رانش مفهوم اجرا نمودیم. در این آزمایش درصد برچسب‌های واقعاً پرسیده‌شده در iMNB + AL برابر با ۹۰ به دست آمد. در شکل ۱۳ نتایج ارزیابی نشان داده شده که برای ساده‌ترشدن مقایسه، دقت‌های به دست آمده برای iMNB توسط نمودار میله‌ای با رنگ قرمز و دقت‌های به دست آمده برای iMNB + AL توسط منحنی آبی‌رنگ آمده است.

همان طور که در شکل ۱۳ مشخص است استفاده از روش iMNB + AL با استفاده از روش تشخیص و مدیریت رانش مفهوم نه تنها باعث کاهش ۱۰٪ برچسب‌های استفاده‌شده است بلکه میانگین دقت بالاتری نیز توسط آن به دست آمده است. در واقع پس از ورود ۶۰۰ نمونه همیشه iMNB + AL دقت بیشتری نسبت به iMNB داشته است. به طور میانگین استفاده از روش پیشنهادی تشخیص و مدیریت رانش مفهوم باعث افزایش ۱٫۸٪ دقت دسته‌بند شده است. شایان ذکر است به دلیل کوچک‌بودن مجموعه داده (۲۰۰۰ نمونه) و رانش مفهوم خفیف آن میزان بهبود میانگین دقت‌ها کم است. در صورتی که از مجموعه داده‌های بزرگ‌تر با رانش مفهوم شدیدتر برای آزمایش استفاده شود استفاده از روش تشخیص و مدیریت رانش مفهوم باعث بهبود بیشتری می‌شود.

## ۵- نتیجه‌گیری و کارهای آتی

در آزمایش روش‌های دسته‌بندی غیر افزایشی روش‌های MNB، NB، J۴۸، Jrip و SVM با استفاده از ۴ مجموعه داده مقایسه شدند. در این آزمایش، بهترین روش‌های غیر افزایشی به ترتیب MNB و SVM بودند که چه از لحاظ زمان اجرا و چه از لحاظ دقت از سایر روش‌ها بهتر بودند. در آزمایش بعدی روش‌های افزایشی iMNB، iNB، درخت Hoeffding و iMNB به همراه روش پیشنهادی یادگیری فعال آزمایش شدند. نتایج به دست آمده از این آزمایش نشان می‌دهد روش iMNB بهترین دقت و زمان اجرا را داشت و با آزمایش روش پیشنهادی یادگیری فعال با استفاده از درصدی کمتر نمونه‌های برچسب‌خورده به دقتی تقریباً برابر با استفاده از تمام نمونه‌های برچسب‌خورده رسیدیم که نشانگر اثربخشی روش پیشنهادی نظرکاوی افزایشی است.

علاوه بر این در آخرین آزمایش ثابت شد استفاده از روش تشخیص و مدیریت رانش مفهوم باعث بالارفتن دقت دسته‌بند می‌شود.

در کارهای آتی می‌توان مکانیزم‌های برچسب‌گذاری غیر از فرد خبره را به منظور هرچه بیشتر خودکارسازی روش یادگیری فعال در زمینه نظرکاوی افزایشی بررسی و آزمایش نمود. همچنین به عنوان کارهای آینده می‌توان از ارائه روش‌های یادگیری فعالی که تمامی انواع رانش مفهوم را مدیریت نماید و بررسی روش‌های خودکار تعیین بودجه برچسب‌گذاری در نظرکاوی افزایشی نام برد.

## مراجع


[1] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: tasks, approaches and applications," *Knowledge-Based Syst.*, vol. 89, pp. 14-46, Nov. 2015.

[2] J. A. Balazs and J. D. Velasquez, "Opinion mining and information fusion: a survey," *Inf. Fusion*, vol. 27, pp. 95-110, Jan. 2016.

[3] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proc. of the ACL-02 Conf. on Empirical Methods in Natural Language Processing*, vol. 10, pp. 79-86, 2002.

[4] R. McDonald, K. Hannan, T. Neylon, M. Wells, and J. Reynar, "Structured models for fine-to-coarse sentiment analysis," in *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 432-439, Prague, Czech Republic, 2007.

[5] Y. Dang, Y. Zhang, and H. Chen, "A lexicon-enhanced method for sentiment classification: an experiment on online product reviews," *IEEE Intell. Syst.*, vol. 25, no. 4, pp. 46-53, Jul./Aug. 2010.

[6] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification," in *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 440-447, 2007.

[7] M. Rushdi-Saleh, M. T. Maritn-Valdivia, A. M. Raez, and L. A. U. Lpez, "Experiments with SVM to classify opinions in different domains," *Expert Syst. Appl.*, vol. 38, no. 12, pp. 14799-14804, Nov. /Dec. 2011.

[8] B. Pang and L. Lee, "A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. of the 42nd Annual Meeting on Association for Computational Linguistics*, Article No. 271, 8 pp., Barcelona, Spain, 21-26 Jul. 2004.

[9] M. Taboada and J. Grieve, "Analyzing appraisal automatically," in Proc. of the AAAI Spring Symp. on Exploring Attitude and Affect in Text: Theories and Applications, pp. 158-161, Mar. 2004.

[10] X. Bai, "Predicting consumer sentiments from online text," *Decis. Support Syst.*, vol. 50, no. 4, pp. 732-742, Mar. 2011.

[11] Z. Zhang, Q. Ye, Z. Zhang, and Y. Li, "Sentiment classification of internet restaurant reviews written in Cantonese," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7674-7682, Jun. 2011.

[12] L. K. W. Tan, J. C. Na, Y. L. Theng, and K. Chang, "Phrase-level sentiment polarity classification using rule-based typed dependencies and additional complex phrases consideration," *J. Comput. Sci. Technol.*, vol. 27, no. 3, pp. 650-666, Jan. 2012.

[13] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, "Sentiment classification: the contribution of ensemble learning," *Decis. Support Syst.*, vol. 57, pp. 77-93, Jan. 2014.

[14] R. Moraes, J. F. Valiati, and W. P. G. Neto, "Document-level sentiment classification: an empirical comparison between SVM and ANN," *Expert Syst. Appl.*, vol. 40, no. 2, pp. 621-633, 1 Feb. 2013.

[15] A. S. H. Basari, B. Hussin, I. G. P. Ananta, and J. Zeniarja, "Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization," *Procedia Eng.*, vol. 53, pp. 453-462, 2013.

[16] M. Ghiassi, J. Skinner, and D. Zimbra, "Twitter brand sentiment analysis: a hybrid system using N-gram analysis and dynamic artificial neural network," *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6266-6282, Nov. 2013.

[17] J. Smailovic, M. Grcar, N. Lavrac, and M. Znidarsic, "Stream-based active learning for sentiment analysis in the financial domain," *Inf. Sci.*, vol. 285, pp. 181-203, 20 Nov. 2014.

[18] M. Zimmermann, E. Ntoutsi, and M. Spiliopoulou, "Incremental active opinion learning over a stream of opinionated documents," *Proc. Workshop on Issues of Sentiment Discovery and Opinion Mining*, *WISDOM'15*, 10 pp., Sydney, Australia, 10 Aug. 2015.

[19] E. Serrao and M. Spiliopoulou, "Active stream learning with an oracle of unknown availability for sentiment prediction," in *Proc.*

*Workshop on Interactive Adaptive Learning, IAL'18,* pp. 36-47, Dublin, Ireland, 2018.
[20] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with drifting streaming data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 1, pp. 27-39, Jan. 2014.
[21] A. McCallum and K. Nigam, "A comparison of event models for Naive Bayes text classification," in *Proc. AAAI-98 Workshop on Learning for Text Categorization*, pp. 41-48, 1998.
[22] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Tackling the poor assumptions of Naive Bayes text classifiers," in *Proc. of the 20th Int. Conf. on Machine Learning ICML'03*, pp. 616-623, Washington, DC, USA, 21-24 Nov. 2003.
[23] T. R. Hoens, R. Polikar, and N. V. Chawla, "Learning from streaming data with concept drift and imbalance: an overview," *Prog. AI*, vol. 1, no. 1, pp. 89-101, 2012.
[24] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Proc. of Advances in Artificial Intelligence, SBIA*, vol. 3171, pp. 286-295, 2004.
[25] I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, and S. J. Cunningham, "Weka: practical machine learning tools and techniques with java implementations," 1999.
[26] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Proc. 11th Conf. on Uncertainty in Artificial Intelligence*, pp. 338-345, Montreal, QC, Canada, 18-20 Aug. 1995.
[27] R. Quinlan, *C4.5: Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann Publishers, 1993.
[28] W. W. Cohen, "Fast effective rule induction," in *Proc. 12th Int. Conf. on Machine Learning*, pp. 115-123, Tahoe City, CA, USA, 9-12 Jul. 1995.
[29] C. C. Chang and C. J. Lin, LIBSVM-A Library for Support Vector Machines, 2001.

**سیدفخرالدین نوربهبهانی** مقطع کارشناسی خود را در رشته مهندسی کامپیوتر گرایش نرم‌افزار در دانشگاه صنعتی اصفهان سال ۱۳۸۵ به پایان رساند و فارغ التحصیل کارشناسی ارشد مهندسی فناوری اطلاعات گرایش تجارت الکترونیکی از دانشگاه صنعتی امیرکبیر در سال ۱۳۸۸ است. وی موفق به اخذ دکترای مهندسی کامپیوتر گرایش هوش مصنوعی و امنیت اطلاعات با رتبه اول از دانشگاه صنعتی اصفهان در سال ۱۳۹۴ شد و هم اکنون استاد دانشکده مهندسی کامپیوتر دانشگاه خوانسار می‌باشد. زمینه‌های پژوهشی مورد علاقه ایشان، تعامل انسان کامپیوتر، تجارت الکترونیکی، امنیت اطلاعات و یادگیری ماشین می‌باشد.