طبقه‌بندی مؤثر نیمه نظارتی مبتنی بر ریزخوشه جریان داده تکاملی

میثم معصومی [*۱]، مرتضی یوسف صنعتی [۲]

۱- دانشجوی کارشناسی ارشد، دانشکده فنی مهندسی، دانشگاه بوعلی سینا

۲- هیئت‌علمی گروه نرم‌افزار، دانشکده فنی مهندسی، دانشگاه بوعلی سینا

چکیده

در عصر حاضر، داده‌ها با سرعت و حجم بسیار زیادی به‌صورت بدون توقف و به شکل جریان داده در حال تولید هستند. یکی از روش‌های کاربردی در کار با جریان داده، طبقه جریان ورودی است. از این رو در سال‌های اخیر روش‌های زیادی برای طبقه‌بندی جریان داده ارائه شده است. در روش‌های طبقه‌بندی نظارتی جریان داده، جهت به‌روزرسانی الگوریتم نیاز به دسترسی به برچسب واقعی هر نمونه ورودی بعد از عمل طبقه‌بندی می‌باشد درحالی‌که در دنیای واقعی دسترسی به برچسب واقعی داده‌ها امری زمان‌بر و دشوار است. از این روی روش‌های یادگیری نیمه‌نظارتی، عملکرد بهتری در کاربردهای دنیای واقعی ارائه می‌دهند. از سویی در این دسته از روش‌ها، چالش اصلی ارائه دقت کافی با حفظ سرعت و حافظه اجرایی مناسب است. در این مقاله سعی در بهبود الگوریتمی جهت طبقه‌بندی جریان داده با مورد لحاظ قراردادن دقت، زمان اجرایی و حافظه مصرفی بوده است. به طور خاص، هدف این مقاله بهبود زمان و مرتبه اجرایی با حفظ دقت و حافظه مصرفی است. به همین جهت یکی از الگوریتم‌های کارآمد طبقه‌بندی جریان داده بهبود داده شده است. نتایج خروجی بر روی داده‌های دنیای واقعی و ساختگی نشان از بهبود زمان اجرایی الگوریتم با حفظ دقت و حافظه مصرفی دارد.

کلمات کلیدی: Data Stream، Classification، Micro Cluster، Semi-Supervised Learning.

۱. مقدمه

امروزه با توسعه چشم‌گیر حوزه‌های مرتبط با فناوری اطلاعات و ارتباطات و فراگیری استفاده از فناوری‌های جدید مانند اینترنت اشیا، شبکه‌های اجتماعی، سیستم‌های پایش و کنترل هوشمند، در اکثر حوزه‌ها، داده‌ها با حجم زیاد و سرعت بالا در حال تولید است. بمنظور استخراج دانش از چنین داده‌هایی روش‌های متفاوتی ایجاد و مورد استفاده قرار گرفته است. یکی از این روش‌ها طبقه‌بندی جریان داده است.

جریان داده دنباله‌ای از نمونه‌های داده است که به‌صورت پیوسته، با طول نامحدود و سرعت بالا دریافت می‌شود همچنین جریان داده دارای توزیع داده‌ای است که می‌تواند در طول زمان تغییر کند[1].

[*] **Email Corresponding Author: M.Masomi@eng.basu.ac.ir**

با توجه به ماهیت جریان داده‌ها، روش‌های طبقه‌بندی جریان داده نیز با چالش‌های متفاوتی از جمله طول بی‌نهایت جریان، سرعت بالای آن، انحراف مفهوم[1]، تکامل مفهوم[2] و داده‌های پرت[3] روبه‌رو هستند. همچنین امکان ذخیره‌سازی داده‌های ورودی و به‌تبع آن پردازش چند باره آن‌ها مقدور نمی‌باشد لذا الگوریتم‌های پیشنهادی باید بتوانند دانش مفید موجود در جریان را با یکبار پردازش داده‌های در حال گذر استخراج نمایند. علاوه بر این به دلیل پویایی جریان داده امکان تغییر ماهیت یک مفهوم در طول جریان وجود دارد که این امر انحراف مفهوم [2] نامیده می‌شود. باید توجه داشت که ممکن است در جریان ورودی کلاس‌های جدیدی [3] نیز ظاهر گردند که تشخیص کلاس های مذکور از داده‌های پرت امری مهم و پیچیده است.

تاکنون روش‌های متفاوتی برای طبقه‌بندی داده‌های جریان ارائه گردیده است. در این روش‌ها می‌توان از تکنیک‌های متفاوتی از جمله طبقه‌بندهای گروهی [4]، KNN[4]، Naïve Bayes [5]، درخت‌های تصمیم [6]، روش‌های فازی [7]، شبکه‌های عصبی و یادگیری عمیق [8] برای انجام طبقه‌بندی استفاده نمود. در برخی از این روش‌ها، طبقه‌بندی با استفاده از یادگیری نظارتی انجام می‌شود. در این‌گونه روش‌ها به‌روزرسانی الگوریتم یا مدل ساخته شده منوط بوجود برچسب واقعی داده‌ها است. بطور معمول دسترسی به برچسب واقعی تمامی داده‌ها امری دشوار یا ناممکن است. از این رو چنین فرضی یک محدودیت جدی برای این دسته از الگوریتم‌ها تلقی می‌شود [9–12]. برای رفع چنین مشکلی از روش‌های یادگیری نیمه نظارتی استفاده می‌گردد که در آن‌ها وجود برچسب واقعی تمام نمونه‌ها ضروری نبوده و حضور بخشی از آن‌ها کافی است [12–16].

در بعضی از این روش‌ها، عمل طبقه‌بندی به کمک ریزخوشه‌ها انجام می‌گردد که در آن ابتدا داده‌های ورودی جریان خوشه‌بندی شده و سپس ریزخوشه‌های متناظر با خوشه‌ها ساخته می‌شود [17–20]. هر ریزخوشه شامل یکسری اطلاعات آماری از خوشه متناظر با آن است که هدف آن خلاصه سازی اطلاعات و داده‌های موجود در خوشه است. پایه روش‌های طبقه‌بندی مبتنی بر ریزخوشه مشابه بسیاری از روش‌های دیگر طبقه‌بندی جریان داده شامل دو مرحله آموزش و آنلاین می‌باشد[21–23].

در مرحله آموزش، الگوریتم از روی نمونه‌های دارای برچسب واقعی آموزش داده شده و یک مدل مطلوب ساخته می‌شود که شامل مجموعه‌ای از ریزخوشه‌ها می‌باشد. در مرحله آنلاین مدل ایجاد شده برای عمل طبقه‌بندی استفاده می‌گردد. همچنین برای پاسخ‌گویی به انحراف مفهوم، تکامل مفهوم و حالت پویای جریان داده باید مدل ساخته شده در طول مرحله آنلاین به‌صورت مداوم به‌روزرسانی شود.

در این مقاله الگوریتمی پیشنهاد شده است که بهبود یافته یکی از روش‌های کارآمد طبقه‌بندی نیمه نظارتی مبتنی بر ریزخوشه جریان داده است [18]. در این بهبود مرتبه زمانی الگوریتم از $O(maxMC^2)$ به مرتبه $O(maxMC.\log(maxMC))$ کاهش یافته است. لازم به ذکر است که کارایی الگوریتم پیشنهادی نسبت به الگوریتم پایه بیشتر شده است در حالی‌که میزان از دست رفتن دقت الگوریتم به کارایی حاصل شده چندان چشم‌گیر نمی‌باشد.

در بخش ۲ برخی از کارهای مرتبط صورت گرفته در طبقه‌بندی نیمه نظارتی جریان داده ذکر گردیده است. سپس روش پیشنهادی در بخش ۳ ارائه شده و در پایان، بخش ۴ به ارزیابی و جمع‌بندی روش پیشنهادی پرداخته است.

۲. کارهای مرتبط

در [18] الگوریتمی پیشنهاد شده است که با ارائه یک ساختار یادگیری نیمه نظارتی مبتنی بر ریزخوشه، طبقه‌بندی جریان داده را با دقت و کارایی مناسب انجام می‌دهد. در این روش نمونه داده‌ها به‌صورت تک تک بررسی گردیده و مدل ساخته بر اساس نتایج بررسی به‌روز می‌گردد. این نوع بررسی و به‌روزرسانی مدل که به ازای هر نمونه انجام می‌شود موجب بهبود عملکرد الگوریتم در برخورد با انحراف مفهوم و شناسایی سریع‌تر کلاس‌های جدید می‌شود. البته لازم به ذکر است که در بررسی جریان داده و تشخیص کلاس‌های جدید معتبر

[1] Concept Drift
[2] Concept Evolution
[3] Outliers
[4] K Nearest Neighbors

(تکامل مفهوم [19]) و هم چنین تمایز آن با داده‌های پرت نمی‌توان بر اساس یک نمونه تنها تصمیم‌گیری نمود. از طرف دیگر کاهش کارایی ناشی از ایجاد یک ریزخوشه برای هر نمونه‌ای که می‌تواند نمایانگر یک کلاس‌های جدید باشد چشم‌گیر است.

در [24] یک روش طبقه‌بندی نیمه نظارتی گروهی مبتنی بر ریزخوشه ارائه شده است که در آن جریان داده به‌صورت مبتنی بر بسته پردازش می‌شود. در این روش از نمونه‌های دارای برچسب واقعی برای تولید طبقه‌بند جدید استفاده می‌شود و طبقه‌بند جدید جایگزین کم دقت‌ترین طبقه‌بند می‌گردد. در [25]، روشی مشابه همین روش ارائه شده است که در آن از الگوریتم انتشار برچسب[1] برای برچسب گذاری نمونه‌های بدون برچسب استفاده شده است.

در [26] روش SCo-Forest که نسخه بهبود یافته روش Co-Forest [27] بر روی جریان داده می‌باشد ارائه گردیده است. این روش داده ورودی را به صورت بسته‌هایی شامل داده‌های دارای برچسب و بدون برچسب دریافت می‌نماید. از روی برخی از زیر مجموعه های داده‌های دارای برچسب تعدادی درخت تصادفی تولید می‌شود. لازم به ذکر است جهت بهبود و به‌روزرسانی یک درخت از نتایج حاصل از طبقه‌بندی نمونه‌های برچسب خورده توسط سایر درخت‌ها استفاده می‌شود. در این روش به دلیل هزینه بالای تولید درخت‌های تصمیم در جریان داده از روش VFDT [28] استفاده شده است. در روش پیشنهادی مذکور در صورت تشخیص یک تغییر ناگهانی یا یک تغییر بزرگ، عمل حذف درخت‌های با عملکرد پایین توسط روش ADWIN [29] صورت می‌پذیرد.

در [17] روشی نیمه نظارتی مبتنی بر ریزخوشه ارائه شده است. در این روش برای طبقه‌بندی از معیارهایی مانند Cluster Structure Distance، Cluster Reliablity و توزیع برچسب‌ها استفاده شده است. در این روش بر اساس ضریب پوشش تصمیم‌گیری در مورد نحوه به‌روزرسانی مدل صورت می‌گیرد. همچنین در [30] برای تولید مدل یادگیری از درخت‌های تصمیم استفاده شده است. بعد از تولید درخت‌های تصمیم، این روش از خوشه‌بندی برای تشخیص برچسب نمونه‌های بدون برچسب هر یک از گره‌های برگ درخت‌ها استفاده می‌کند.

در [16] یک روش نیمه‌نظارتی و نظارتی مبتنی بر شبکه عصبی ELM[2] پیاده سازی شده است. در مدل نظارتی این روش یک Ensemble از طبقه‌بندهای ELM تولید می‌شود. در فاز آنلاین برای هر طبقه‌بند خطای طبقه‌بندی محاسبه شده و در صورتی که این خطا بیشتر از میزان آستانه تعیین شده باشد این طبقه‌بند از Ensemble حذف می‌شود و یک طبقه جدید آموزش دیده با تمامی نمونه‌های پنجره جاری جایگزین آن می‌شود. در مدل نیمه‌نظارتی آن از یک روش افزایشی برای محاسبه پارامترها و به‌روزرسانی شبکه استفاده می‌شود. از جمله مزایای این روش به استفاده از شبکه عصبی ELM و به‌روزرسانی پارامترها بر اساس برچسب هر نمونه می‌توان اشاره کرد. این رویکرد موجب پویایی این روش در برخود با انحراف مفهوم شده است. از سوی دیگر در شناسایی کلاس‌های جدید نیز به نسبت روش‌های مشابه خود هوشمندانه عمل کرده است.

۳. طبقه‌بندی جریان داده

در این بخش ابتدا به طرح مسئله پرداخته و سپس الگوریتم پیشنهادی ارائه شده است.

۳. ۱. طرح مسئله

با توجه به مشکلاتی که برای طبقه‌بندی جریان داده‌ها در بخش مقدمه ذکر گردید سؤال اساسی این است که چگونه می‌توان روشی نیمه‌نظارتی جهت طبقه‌بندی جریان داده ارائه نمود که ضمن دارا بودن دقت مناسب، با سرعت مطلوب عمل طبقه‌بندی را به انجام رساند. به‌منظور تحقق این هدف روش‌های متفاوت نیمه‌نظارتی مورد بررسی قرار گرفت و سپس روش مطرح شده

[1] Label Propagation
[2] Extreme Learning Machine

در [18] به عنوان یکی از بهترین الگوریتم‌های موجود در این رده - انتخاب گردید. سپس سعی شد تا با بهبود عملکرد الگوریتم مذکور ضمن حفظ دقت نسبی سرعت الگوریتم به میزان چشم‌گیری افزایش یابد.

۳. ۲. روش پیشنهادی

روش پیشنهاد شده در [18] مانند سایر روش‌های طبقه‌بندی مبتنی بر ریزخوشه شامل سه بخش اساسی است. بخش اول الگوریتم وظیفه ساخت مدل اولیه مبتنی بر ریزخوشه از مجموعه نمونه‌های یادگیری را برعهده دارد. بخش دوم نیز کار طبقه‌بندی جریان داده ورودی در مرحله آنلاین را انجام می‌دهد که این کار بر اساس یک روش مبتنی بر فاصله با استفاده از طبقه‌بندهای گروهی KNN صورت می‌گیرد. بخش سوم، مدل ساخته شده بر اساس نمونه‌های جدید را به منظور درک حالت پویای جریان بهبود داده و به‌روزرسانی می‌کند. علیرغم اینکه این روش نسبت به سایر روش‌های مطرح در این حوزه دقت و سرعت بالاتری دارد ولی هنوز از کارایی لازم برای پردازش جریان های پرسرعت برخوردار نیست؛ لذا بهبود سرعت الگوریتم می‌تواند تاثیر بسزایی در قابلیت استفاده آن در جریان‌های سریع داشته باشد. از این رو در این مقاله بخش‌هایی از الگوریتم مذکور تغییر داده شده است. در الگوریتم پیشنهادی بجای اینکه نمونه‌هایی که نماینده کلاس‌های جدید هستند به‌صورت تک تک بررسی و مدل بر اساس آن به‌روزرسانی شود، از این گونه نمونه‌ها بسته‌هایی ایجاد شده و سپس مورد بررسی قرار می‌گیرند. انجام چنین رویکردی منجر به افزایش چشمگیر سرعت می‌شود. شکل (۱) فلوچارت روش پیشنهادی را نشان می‌دهد.

۳. ۲. ۱. ساخت مدل اولیه

این قسمت از الگوریتم پیشنهادی کاملا مشابه [18] است. الگوریتم (۱) نحوه ساخت مدل اولیه توسط داده‌های آموزشی را نشان می‌دهد. مجموعه داده‌های آموزشی D_{init} توسط الگوریتم دریافت شده و هر کلاس موجود در آن توسط الگوریتم kmeans به $\mathcal{K}$ خوشه تقسیم می‌شود. عدد $\mathcal{K}$ یکی از پارامترهای ورودی الگوریتم است. برای هر خوشه، ریزخوشه‌ای به‌صورت $MC = (LS, SS, N, W, T, C, CL, R)$ تشکیل می‌شود. در این توصیف، LS و SS به ترتیب نشانگر مجموع خطی و مجموع مربعات همه نقاط موجود در آن خوشه هستند که به‌صورت زیر محاسبه می‌شوند.

$$LS = \sum_{1}^{N} x_i \qquad (۱)$$

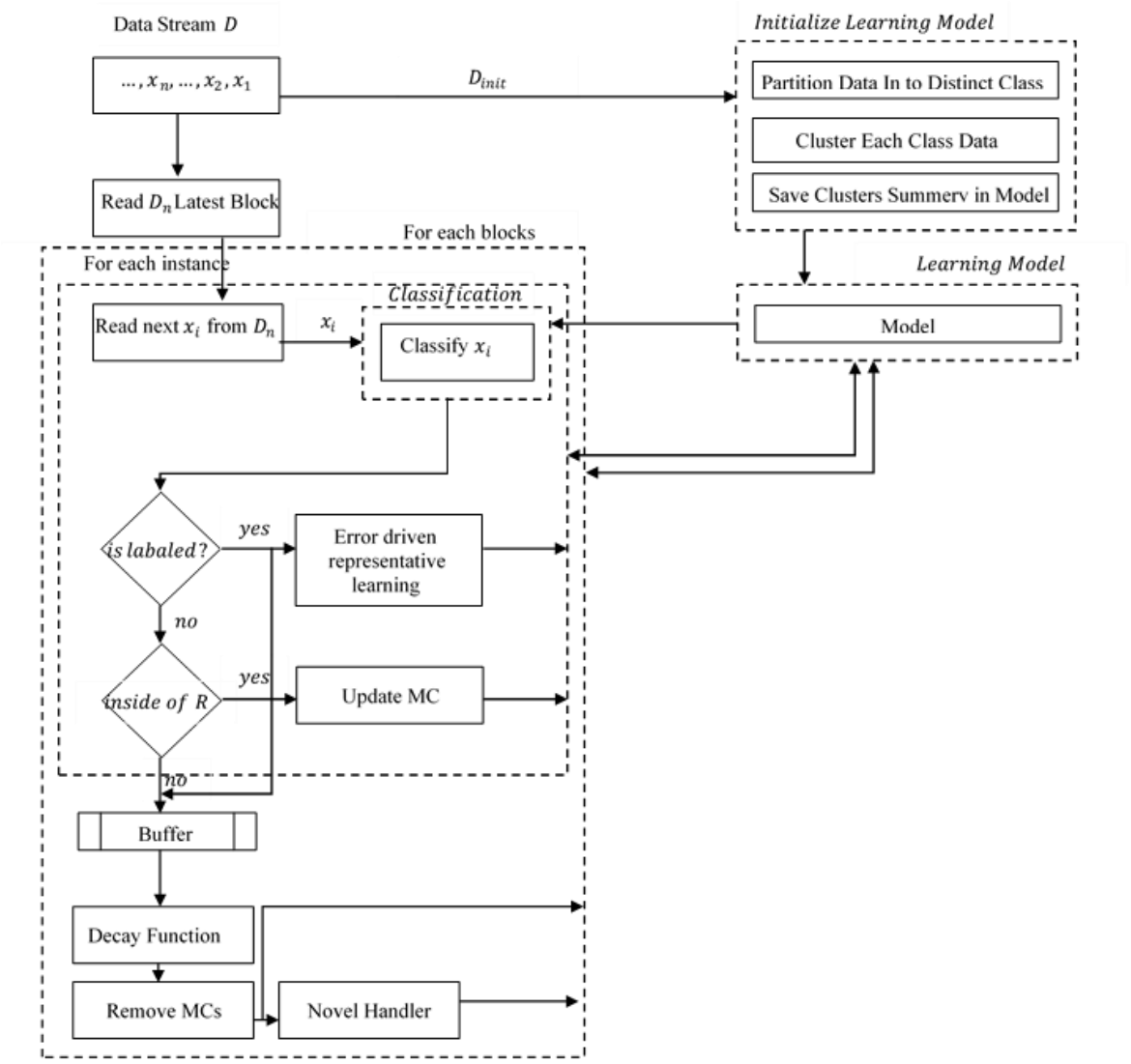
$$SS = \sum_{1}^{N} (x_i)^2 \qquad (۲)$$

x_i بردار توصیف کننده نمونه‌های موجود در خوشه و N نیز نشانگر تعداد کل نقاط موجود در آن خوشه می‌باشد. W عددی است که میزان اهمیت یا اطمینان پذیری ریزخوشه در هر لحظه را نشان می‌دهد. این عدد در لحظه تولید یک ریزخوشه برابر ۱ می‌باشد که با گذشت مقدار آن تغییر می‌نماید. T زمان سپری شده از آخرین به‌روزرسانی ریزخوشه جاری است. C برداری است که مرکز خوشه را نشان می‌دهد. CL برچسب کلاس ریزخوشه است. اگر یک ریزخوشه دارای برچسب واقعی نباشد این مقدار برابر $\emptyset$ خواهد بود. R نیز شعاع ریزخوشه را نشان می‌دهد که به صورت زیر محاسبه می‌شود.

$$C = \frac{LS}{N} \qquad (۳)$$

$$R = \left(\frac{N \times SS - LS^2}{N^2}\right)^{1/2} \qquad (۴)$$

بعد از ساخت مدل اولیه، مرحله آنلاین شروع می‌شود.



شکل ۱- فلوچارت الگوریتم پیشنهادی

۳. ۲. ۲. مرحله آنلاین – فاز طبقه‌بندی

در این بخش، جریان داده به‌صورت بسته‌هایی از نمونه‌ها وارد الگوریتم می‌شود، اندازه بسته‌ها جزء پارامترهای الگوریتم می‌باشد. در هر بسته برای هر نمونه عمل طبقه‌بندی بر اساس روش گروهی مبتنی بر فاصله انجام می‌گردد. در این روش از Θطبقه‌بند متمایز KNN استفاده شده‌است که تعداد همسایه‌ها $K \in \{1, 3, 5, 7, \ldots\}$ بوده و Θ تعداد طبقه‌بندهای گروه را مشخص می‌نماید. طبقه بندی هر یک از نمونه‌های موجود در هر بسته توسط بهترین طبقه‌بند انجام می‌پذیرد. فرآیند انتخاب بهترین طبقه‌بند در رابطه (۵) نشان داده شده است.

$$\Pi_{K-NN} = \frac{1}{m} \sum_{i=t_i-m+1}^{t_i} (KNN(x_i) = y_i) \; [18] \qquad (۵)$$

Π وزن هر طبقه‌بند در لحظه t_i است. مقدار m تعداد نمونه‌های برچسب خورده اخیر را که باید بررسی شوند مشخص می‌نماید. $KNN(x_i)$ برچسب اختصاص یافته به نمونه x_i و y_i برچسب واقعی کلاس آن نمونه است. طبقه‌بندی که بالاترین

مقدار Π را داشته باشد بهترین طبقه‌بند محسوب شده و عمل طبقه‌بندی توسط آن انجام خواهد شد. پس از انجام طبقه‌بندی مدل ساخته شده به‌روز می‌گردد. الگوریتم (۱) نحوه تولید مدل اولیه را نشان می‌دهد.

۳. ۲. ۳. مرحله آنلاین – فاز بروز رسانی مدل

در [18] جریان داده به صورت مبتنی بر نمونه وارد الگوریتم شده و عمل طبقه‌بندی روی آن صورت می‌گیرد. در صورتی که این نمونه به عنوان نماینده یک کلاس جدید تشخیص داده شود ریزخوشه‌ای به مرکزیت آن نمونه با شعاعی برابر شعاع نزدیکترین ریزخوشه به نمونه مذکور ساخته می‌شود. به دلیل اینکه ممکن است تعداد چنین نمونه‌هایی در جریان بسیار زیاد باشد استفاده از چنین رویکردی موجب کاهش سرعت طبقه‌بندی می‌شود. روش پیشنهادی پردازش نماینده‌های کلاس‌های جدید را تا بعد از طبقه‌بندی سایر نمونه‌های بسته جاری، به تاخیر می‌اندازد. نتایج حاصل نشان می‌دهد که چنین امری منجر به افزایش سرعت انجام طبقه‌بندی می‌شود.

در این فاز مدل ساخته شده به مرور زمان به‌روز می‌گردد. در ابتدا این به‌روزرسانی بر اساس نتایج طبقه‌بندی نمونه‌هایی که نماینده کلاس جدیدی نمی‌باشند صورت می‌پذیرد. در ادامه پس از مشخص شدن وضعیت نمونه‌هایی که نماینده کلاس جدیدی هستند به‌روزرسانی مدل بر اساس آن‌ها نیز انجام خواهد شد. لذا در گام ابتدایی، اگر نمونه رسیده برچسب واقعی نیز داشته باشد با عمل تنبیه و تشویق صورت پذیرد و این برچسب واقعی با آنچه بهترین طبقه‌بند برای آن در نظر گرفته است یکسان باشد، میزان اهمیت ریزخوشه‌هایی که در عمل طبقه‌بندی این نمونه مؤثر بوده‌اند یک واحد افزایش می‌یابد. از اهمیت ریزخوشه‌هایی که در همسایگی این نمونه بوده و در عمل طبقه‌بندی مؤثر نبوده‌اند نیز یک واحد کاهش خواهد یافت.

الگوریتم (۱) تولید مدل اولیه

```
1   Function InitializeModel(D_init, K)
2   |   Model ← ∅
3   |   𝒳 ← Partition(D_init)
4   |   foreach 𝒳^c ∈ 𝒳 do
5   |   |   ℋ ← Cluster(𝒳^c, K)
6   |   |   foreach ℋ_i ∈ ℋ do
7   |   |   |   N ← |ℋ_i|
8   |   |   |   LS ← Σ_{j=1}^{N} x_j
9   |   |   |   SS ← Σ_{j=1}^{N} (x_j)^2
10  |   |   |   W ← 1
11  |   |   |   T ← 0
12  |   |   |   CL ← c
13  |   |   |   R ← ((N × SS − LS^2) / N^2)^{1/2}
14  |   |   |   C ← LS / N
15  |   |   |   MC ← (LS, SS, N, W, T, C, CL, R)
16  |   |   |   Model ← Model ∪ MC
17  |   |   end
18  |   end
19  |   return Model
20  End Function
```

لازم به ذکر است اگر فاصله نمونه‌ای از مرکز نزدیک‌ترین ریزخوشه به آن، بیشتر از شعاع آن ریزخوشه باشد نمونه مذکور به لیست نماینده‌های کلاس‌های جدید اضافه می‌گردد. در غیر این صورت با افزوده شدن نمونه به ریزخوشه به‌روزرسانی صورت می‌پذیرد. این فعالیت برای همه نمونه‌های موجود در بسته انجام می‌شود.

در انتهای بررسی نمونه‌های هر بسته، درجه اهمیت تمامی ریزخوشه‌ها مجددا متناسب با زمان جاری به‌روز می‌شود. در صورتی که لیست نماینده‌های کلاس‌های جدید خالی نباشد، نمونه‌های موجود در آن به $\mathcal{K}$ خوشه تقسیم می‌شود. اگر خوشه‌ای شامل نمونه‌ای باشد که دارای برچسب واقعی است، برچسب واقعی آن نمونه به خوشه مذکور اختصاص می‌یابد. در غیر این صورت برچسب واقعی آن خوشه برابر تهی خواهد بود. باید توجه داشت که از روی این خوشه‌ها ریزخوشه‌های متناظر ساخته می‌شود.

الگوریتم (۲) طبقه‌بندی جریان داده

Input: D: data stream,
$blockSize$: Size of Blocks
$maxMC$: maximum number of MC in model
Θ: number of K-NN classifiers
m: number of most recent examples to calculate classifier weight
K: number of clusters per class
Output: Classification results: P_labels

$\boldsymbol{Model} \leftarrow$ InitializeModel(D_{init},K)
while Not end of stream **do**
  read the last data block D_n
  novel_buffer $\leftarrow$ {}
  $cur\_t \leftarrow 0$
  foreach $x_i \in D_n$ **d**
    $cur\_t \leftarrow cur\_t + 1$
    [p_label, MCs] $\leftarrow$ Classify($x_i, Model$)
    Output p_label
    if $x_i.label \neq \emptyset$ **then**
      foreach $all\ MC_i \in MCs$ **do**
        if $MC_i.label = x_i.label$ **then**
          $W_{MC_i} \leftarrow W_{MC_i} + 1$
          $T_{MC_i} \leftarrow cur\_t$
        else
          $W_{MC_i} \leftarrow W_{MC_i} - 1$
        end
      end
    end
    [Dist, MC] $\leftarrow$ NearestMC($Model, x_i$)
    if $Dist > R_{MC}\ or\ (x_i.label \neq \emptyset\ and\ MC.label \neq \emptyset\ \ and\ \ MC.label \neq x_i.label)$ **then**
      novel_buffer.add($x_i.label$)
    else
      UpdateMC($MC, (x_i.cur\_t$)
    end
  end
  UpdateModel($Model, cur\_t$)
  if len(novel_buffer) > 0 **then**
    $Model \leftarrow$ novelHandler($novel\_buffer, Model$)
  end
end

همچنین به دلیل اهمیت نماینده‌های کلاس‌های جدید، در صورتی که تعداد ریزخوشه‌های موجود به سقف تعیین شده توسط خبره سامانه رسیده باشد، فرایند ادغام ریزخوشه‌ها مطابق روش زیر اجرا می‌شود.

ابتدا از بین ریزخوشه‌ها، دو ریزخوشه‌ای که کمترین فاصله را از هم دارند و یکی دارای برچسب واقعی است و دیگری برچسب واقعی ندارد، ادغام می‌شوند. در صورت عدم وجود ریزخوشه بدون برچسب واقعی، کلاسی که بیشترین تعداد ریزخوشه را دارد انتخاب شده و دو ریزخوشه از آن که کمترین فاصله را از هم دارند ادغام می‌گردد. در روش پیشنهادی از ساختار KDTree برای کاهش مرتبه اجرایی یافتن نزدیک‌ترین ریزخوشه‌ها استفاده شده است. در الگوریتم (۲) ساختار روش پیشنهادی نمایش داده شده است.

پیچیدگی زمانی: الگوریتم پیشنهادی شامل سه بخش عمده می‌باشد. بخش آموزش و بخش طبقه‌بندی و به‌روزرسانی و بخش شناسایی کلاس‌های جدید. در بخش آموزش و تولید مدل اولیه با جدا سازی نمونه‌های هرکلاس l ، عمل خوشه

بندی و استخراج k خوشه از آن صورت گرفته و برای هرخوشه عمل تولید ریزخوشه صورت می‌گیرد و هر کلاس نهایت به تعداد $|X^c|$ نمونه خواهد داشت. در نتیجه پیچیدگی زمانی این بخش برابر $\boldsymbol{O}(l.k.|X^c|)$ می‌باشد. در بخش آنلاین، برای هر نمونه درون بسته در خط ۸ عمل طبقه‌بندی صورت می‌گیرد که پیچیدگی زمانی این بخش برابر $\boldsymbol{O}(\boldsymbol{maxMC})$ می‌باشد. سپس بر اساس نتیجه خروجی عمل تشویق یا تنبیه ریزخوشه‌های مشارکت کننده در طبقه‌بندی صورت می‌گیرید که پیچیدگی زمانی این بخش نیز برابر $\boldsymbol{O}(\boldsymbol{maxMC}.k)$ می‌باشد که مقدار k ثابت است. سپس بخش به‌روزرسانی مدل در خطهای ۲۰ تا ۲۵ صورت می‌گیرد که پیچیدگی این بخش‌ها نیز برابر $\boldsymbol{O}(\boldsymbol{maxMC})$ می‌باشد. پس از بررسی بسته، بافر نمایندگان کلاس‌های جدید بررسی می‌شود که پیچیدگی این بخش برابر $\boldsymbol{O}(\boldsymbol{maxMC}.\log(\boldsymbol{maxMC}))$ می‌باشد که در الگوریتم پیشنهادی حداکثر اندازه مدل جزء پارامترهای مسئله بوده و مقدار ثابتی می‌باشد.

پیچیدگی حافظه: روش پیشنهادی مدلی به اندازه حداکثر $\boldsymbol{maxMC}$ از ریزخوشه‌ها را نگهداری می‌کند. هر ریز خوشه شامل اطلاعات محاسبه شده و آماری از خصوصیات خوشه بصورت $MC = (LS, SS, N, W, T, C, CL, R)$ می‌باشد که LS, SS و C برداری با ابعاد فضای ویژگی‌ها می‌باشد. ویژگی‌های N, W, T, CL, R نیز متغییرهایی عددی می‌باشد در نتیجه برای هر ریزخوشه $MC = \boldsymbol{O}(3 \times V_{1\times d} + 5)$ می‌باشد. از سوی دیگر نمایندگان کلاس‌های جدید درون بافر قرار می‌گیرند و چون بافر نمایندگان کلاس‌های جدید در هر بسته بررسی می‌شود حداکثر فضای این بافر نیز از مرتبه $B = \boldsymbol{O}(b.V_{1\times d})$ می‌باشد که b اندازه بسته و $V_{1\times d}$ بردار فضای ویژگی می‌باشد. در نتیجه پیچیدگی حافظه روش پیشنهادی از مرتبه $\boldsymbol{O}(\boldsymbol{maxMC} + B)$ می‌باشد.

۴. نتایج

روش پیشنهادی روی حداقل ۹ مجموعه از مجموعه داده‌های مشهور و پرکاربرد بیشتر از ۱۰ بار اجرا گردیده است. اما بدلیل محدودیت صفحات مقاله، در این بخش صرفاً میانگین حاصل از ۱۰ بار اجرای الگوریتم پیشنهادی بر روی هر یک از مجموعه داده‌های درج شده در جدول (۱) مدنظر قرار گرفته است. لازم به ذکر است که نتایج ارائه شده در این بخش مشابه نتایج حاصل از تست الگوریتم بر روی سایر مجموعه داده‌های ذکر نشده در این مقاله است. ضمناً در جدول (۱) F# و C# به ترتیب تعداد ویژگی‌ها و تعداد کلاس‌های موجود در مجموعه داده را نشان می‌دهد.

جدول (۱) مجموعه داده‌های مورد آزمایش

Dataset	#F	#C	Size
Shuttle	9	7	58,000
Electricity	8	2	45,312
KDD-Cup99	42	23	494,021

جدول (۲) مقایسه مدت زمان اجرای الگوریتم پایه (ستون Mian) و برخی روش‌های مشابه با روش پیشنهادی (ستون Our) را نشان می‌دهد. همانطور که در این جدول قابل مشاهده است، نتایج درج شده نشان از بهبود ۴۴ درصدی زمان اجرای الگوریتم پیشنهادی دارد.

جدول (۲) مقایسه زمان اجرایی الگوریتم پایه و پیشنهادی بر حسب ثانیه

Dataset	Main(s)	Our(s)	Improvement
Shuttle	540.9	322.4	%40.04
Electricity	503.9	311	%38.28
KDD-Cup99	5557.2	3384.8	%39.09
Average Improvement			%44.77

از نقطه نظر دقت نیز هر دو الگوریتم با یگدیگر مقایسه شده‌اند که نتایج آن در جدول (۳) ذکر گردیده است. دقت الگوریتم برای هر یک از مجموعه داده‌ها در مقایسه با روش پایه نشان داده شده است.

جدول (۳) مقایسه دقت الگوریتم پیشنهادی

Dataset	Label%	Our	Main	ReSSL	TLP
Shuttle	1%	**97.73**	97.28	90.98	86.71
	5%	**98.56**	98.55	97.63	94.22
	10%	98.81	**98.85**	98.46	96.49
	20%	**99.20**	99.13	98.73	97.50
	30%	**99.30**	99.25	98.26	98.19
Electricity	1%	**62.13**	59.27	54.14	57.07
	5%	65.27	**65.55**	58.43	63.35
	10%	66.99	**67.88**	60.60	66.49
	20%	69.34	**72.68**	64.52	71.38
	30%	71.09	**73.36**	63.71	72.81
KDD-Cup99	1%	**98.91**	94.00	95.30	76.47
	5%	**99.46**	98.00	97.71	94.74
	10%	**99.60**	98.41	98.13	93.13
	20%	**99.69**	98.73	98.41	89.23
	30%	**99.72**	98.92	98.46	86.11

۵. نتیجه‌گیری

برای طبقه‌بندی جریان داده روش‌های متعددی وجود دارد که در پژوهش انجام شده برخی از الگوریتم‌ها بررسی گردید. در نهایت یکی از الگوریتم‌هایی که دارای کارایی و دقت بهتری نسبت به سایر الگوریتم‌ها بود به عنوان روش پایه انتخاب گردید. سپس با تغییر رویکرد الگوریتم پایه در برخورد با نماینده‌های کلاس‌های جدید بهبود چشمگیری از نقطه نظر مدت زمان اجرا حاصل گردید. از طرف دیگر میزان کاهش یا افزایش دقت نیز بسیار ناچیز بود.

اشکال اساسی روش پیشنهادی حساسیت این روش به اندازه بسته جریان ورودی است که توسط خبره سامانه تعیین می‌شود. اگر اندازه بسته بزرگ در نظر گرفته شود سرعت الگوریتم افزایش یافته ولی از دقت و چابکی الگوریتم در برخورد با تکامل مفهوم و انحراف مفهوم کاسته می‌شود. در نقطه مقابل اگر اندازه بسته کوچک در نظر گرفته شود گرچه انحراف و تکامل مفهوم و کلاس‌های جدید ممکن است سریع‌تر شناسایی شده و دقت افزایش یابد ولی سرعت اجرایی آن کم خواهد شد. از این رو با بهبود الگوریتم طبقه‌بندی و اعمال برخی تغییرات در مرحله طبقه‌بندی، می‌توان دقت الگوریتم را نیز بهبود بخشید. همچنین با استفاده از یک روش هوشمندتر ضمن حفظ سرعت اجرایی می‌توان دقت الگوریتم را نیز افزایش داد.

۶. مراجع

[1] SILVA, Jonathan A., Elaine R. FARIA, Rodrigo C. BARROS, Eduardo R. HRUSCHKA, André C. P. L. F. de CARVALHO a João GAMA. Data stream clustering: A survey. *ACM Computing Surveys* [online]. 2013, **46**(1), 13:1-13:31. ISSN 0360-0300. Dostupné z: doi:10.1145/2522968.2522981

[2] TSYMBAL, Alexey. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*. 2004, **106**(2), 58.

[3] *Novelty detection in data streams | SpringerLink* [online]. [vid. 2020-10-23]. Dostupné z: https://link.springer.com/article/10.1007/s10462-015-9444-8

[4] ZAREMOODI, Poorya, Hamid BEIGY a Sajjad KAMALI SIAHROUDI. Novel class detection in data streams using local patterns and neighborhood graph. *Neurocomputing* [online]. 2015, **158**, 234–245. ISSN 0925-2312. Dostupné z: doi:10.1016/j.neucom.2015.01.037

[5] BAHRI, M., S. MANIU a A. BIFET. A Sketch-Based Naive Bayes Algorithms for Evolving Data Streams. In: *2018 IEEE International Conference on Big Data (Big Data)*: *2018 IEEE International Conference on Big Data (Big Data)* [online]. 2018, s. 604–613. Dostupné z: doi:10.1109/BigData.2018.8622178

[6] GOMES, Heitor M., Albert BIFET, Jesse READ, Jean Paul BARDDAL, Fabrício ENEMBRECK, Bernhard PFHARINGER, Geoff HOLMES a Talel ABDESSALEM. Adaptive random forests for evolving data stream classification. *Machine Learning* [online]. 2017, **106**(9), 1469–1495. ISSN 1573-0565. Dostupné z: doi:10.1007/s10994-017-5642-8

[7] ZAREMOODI, Poorya, Sajjad KAMALI SIAHROUDI a Hamid BEIGY. Concept-evolution detection in non-stationary data streams: a fuzzy clustering approach. *Knowledge and Information Systems* [online]. 2019, **60**(3), 1329–1352. ISSN 0219-3116. Dostupné z: doi:10.1007/s10115-018-1266-y

[8] XU, Shuliang a Junhong WANG. Dynamic extreme learning machine for data stream classification. *Neurocomputing* [online]. 2017, **238**, 433–449. ISSN 0925-2312. Dostupné z: doi:10.1016/j.neucom.2016.12.078

[9] *A survey on data stream clustering and classification | SpringerLink* [online]. [vid. 2021-07-27]. Dostupné z: https://link.springer.com/article/10.1007/s10115-014-0808-1

[10] *A Survey on Ensemble Learning for Data Stream Classification | ACM Computing Surveys* [online]. [vid. 2021-07-27]. Dostupné z: https://dl.acm.org/doi/abs/10.1145/3054925

[11] *A Survey on Multi-Label Data Stream Classification - IEEE Journals & Magazine* [online]. [vid. 2020-03-22]. Dostupné z: https://ieeexplore.ieee.org/document/8941052/

[12] *Semi-Supervised Classification of Data Streams by BIRCH Ensemble and Local Structure Mapping | SpringerLink* [online]. [vid. 2021-07-27]. Dostupné z: https://link.springer.com/article/10.1007%2Fs11390-020-9999-y

[13] CHU, Zhe, Peipei LI a Xuegang HU. Co-training Based on Semi-Supervised Ensemble Classification Approach for Multi-label Data Stream. In: *2019 IEEE International Conference on Big Knowledge (ICBK)*: *2019 IEEE International Conference on Big Knowledge (ICBK)* [online]. 2019, s. 58–65. Dostupné z: doi:10.1109/ICBK.2019.00016

[14] CASALINO, Gabriella, Giovanna CASTELLANO a Corrado MENCAR. Data Stream Classification by Dynamic Incremental Semi-Supervised Fuzzy Clustering. *International Journal on Artificial Intelligence Tools* [online]. 2019, **28**(08), 1960009. ISSN 0218-2130. Dostupné z: doi:10.1142/S0218213019600091

[15] LI, Qiude, Qingyu XIONG, Shengfen JI, Yang YU, Chao WU a Min GAO. Incremental semi-supervised Extreme Learning Machine for Mixed data stream classification. *Expert Systems with Applications* [online]. 2021, **185**, 115591. ISSN 0957-4174. Dostupné z: doi:10.1016/j.eswa.2021.115591

[16] BI, Xin, Chao ZHANG, Xiangguo ZHAO, Donghang LI, Yongjiao SUN a Yuliang MA. CODES: Efficient Incremental Semi-Supervised Classification Over Drifting and Evolving Social Streams. *IEEE Access* [online]. 2020, **8**, 14024–14035. ISSN 2169-3536. Dostupné z: doi:10.1109/ACCESS.2020.2965766

[17] *Reliable Semi-supervised Learning | IEEE Conference Publication | IEEE Xplore* [online]. [vid. 2021-07-24]. Dostupné z: https://ieeexplore.ieee.org/document/7837972

[18] UD DIN, Salah, Junming SHAO, Jay KUMAR, Waqar ALI, Jiaming LIU a Yu YE. Online reliable semi-supervised learning on evolving data streams. *Information Sciences* [online]. 2020, **525**, 153–171. ISSN 0020-0255. Dostupné z: doi:10.1016/j.ins.2020.03.052

[19] DIN, Salah Ud a Junming SHAO. Exploiting evolving micro-clusters for data stream classification with emerging class detection. *Information Sciences* [online]. 2020, **507**, 404–420. ISSN 0020-0255. Dostupné z: doi:10.1016/j.ins.2019.08.050

[20] DIN, Salah Ud, Jay KUMAR, Junming SHAO, Cobbinah Bernard MAWULI a Waldiodio David NDIAYE. Learning High-Dimensional Evolving Data Streams With Limited Labels. *IEEE Transactions on Cybernetics*. 2021.

[21] KRAWCZYK, Bartosz, Leandro L. MINKU, Joao GAMA, Jerzy STEFANOWSKI a Michał WOŹNIAK. Ensemble learning for data stream analysis: A survey. *Information Fusion*. 2017, **37**, 132–156.

[22] HOI, Steven CH, Doyen SAHOO, Jing LU a Peilin ZHAO. Online learning: A comprehensive survey. *arXiv preprint arXiv:1802.02871*. 2018.

[23] BIFET, Albert, Gianmarco DE FRANCISCI MORALES, Jesse READ, Geoff HOLMES a Bernhard PFAHRINGER. Efficient Online Evaluation of Big Data Stream Classifiers. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* [online]. New York, NY, USA: Association for Computing Machinery, 2015, s. 59–68 [vid. 2021-07-28]. KDD '15. ISBN 978-1-4503-3664-2. Dostupné z: doi:10.1145/2783258.2783372

[24] HOSSEINI, Mohammad Javad, Ameneh GHOLIPOUR a Hamid BEIGY. An ensemble of cluster-based classifiers for semi-supervised classification of non-stationary data streams. *Knowledge and Information Systems* [online]. 2016, **46**(3), 567–597. ISSN 0219-3116. Dostupné z: doi:10.1007/s10115-015-0837-4

[25] MASUD, Mohammad M., Clay WOOLAM, Jing GAO, Latifur KHAN, Jiawei HAN, Kevin W. HAMLEN a Nikunj C. OZA. Facing the reality of data stream classification: coping with scarcity of labeled data. *Knowledge and Information Systems* [online]. 2012, **33**(1), 213–244. ISSN 0219-3116. Dostupné z: doi:10.1007/s10115-011-0447-8

[26] WANG, Yi a Tao LI. Improving semi-supervised co-forest algorithm in evolving data streams. *Applied Intelligence* [online]. 2018, **48**(10), 3248–3262. ISSN 1573-7497. Dostupné z: doi:10.1007/s10489-018-1149-7

[27] LI, M. a Z. ZHOU. Improve Computer-Aided Diagnosis With Machine Learning Techniques Using Undiagnosed Samples. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* [online]. 2007, **37**(6), 1088–1098. ISSN 1558-2426. Dostupné z: doi:10.1109/TSMCA.2007.904745

[28] *Mining high-speed data streams | Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* [online]. [vid. 2020-12-12]. Dostupné z: https://dl.acm.org/doi/abs/10.1145/347090.347107

[29] BIFET, Albert a Ricard GAVALDÀ. Learning from Time-Changing Data with Adaptive Windowing. In: *Proceedings of the 2007 SIAM International Conference on Data Mining* [online]. B.m.: Society for Industrial and Applied Mathematics, 2007 [vid. 2020-12-12], Proceedings, s. 443–448. ISBN 978-0-89871-630-6. Dostupné z: doi:10.1137/1.9781611972771.42

[30] WU, Xindong, Peipei LI a Xuegang HU. Learning from concept drifting data streams with unlabeled data. *Neurocomputing*. 2012, **92**, 145–155.