ارائه مدل ترکیبی طبقه‌بندی سریع جریان داده‌ها با استفاده از الگوریتم درخت هافدینگ

مهدی جرگه*[1]، حمید بهادر[۲]

۱- کارشناسی ارشد، دانشگاه آزاد اسلامی واحد فردوس ، گروه کامپیوتر خراسان جنوبی، ایران،
mahdijorgeh@gmail.com

۲- دکتری مهندسی کامپیوتر و صنایع، دانشگاه آزاد و دانشگاه فنی و حرفه‌ای خوی دوم ، ایران،
Hamidbahador52@gmail.com

چکیده

امروزه با پیشرفت فن‌آوری، بسیاری از برنامه‌های کاربردی مقادیر زیادی از داده‌های جریانی با سرعت بالا را تولید می‌کنند. ترافیک در شبکه، نظارت تصویری و شبکه‌های حسگر مثال‌هایی از این مورد می‌باشند. برخلاف داده‌کاوی سنتی که داده‌ها به صورت استاتیک است و می‌توانند خیلی وقت‌ها چند بار خوانده شوند. الگوریتم‌های کاوش داده جریانی با چالش‌های زیادی از قبیل محدودیت حافظه، پاسخ در زمان واقعی و مفهوم رانش روبه‌رو هستند. درهرصورت داده جریانی تعدادی از ویژگی‌های متمایز خواهد بود که باعث می‌شود مقدار حافظه کش با حافظه سیستم موجود برای ذخیره‌سازی جریان داده‌های کل مناسب نباشد. مشکل اصلی داده‌های جریانی، سرعت است که در آن داده‌های جریانی در دسترس است بسیار سریع‌تر از این‌که داده بتواند پردازش و ذخیره شود. بهره‌گیری از روش‌هایی همچون داده‌کاوی برای استخراج دانش و اطلاعات نهفته در داده‌ها، امری غیرقابل اجتناب است. به دلیل حجم بسیار بالای داده‌ها و اهمیت بیشتر داده‌های جدید، ذخیره‌سازی این داده‌ها در بسیاری از کاربردها امری مقرون به صرفه نیست، لذا داده‌هایی که باید مورد پردازش قرار گیرند، همواره به صورت پویا در حال تغییر و تحول هستند.دراین پژوهش دستاوردهای اصلی این تحقیق عبارت‌اند از ارائه مدل ترکیبی طبقه‌بندی سریع جریان داده‌ها با استفاده از الگوریتم درخت هافدینگ برای طبقه‌بندی جریان داده‌ها است با استفاده از این طبقه‌بندی ابتدا یک مدل توسط داده جریانی که متد پیشنهادی می‌باشد آموزش داده می‌شود و سپس توسط دو معیار ارزیابی به روش‌های, Holdout مدل ارزیابی می‌شود. نتایج ارزیابی این دو روش را می‌توان با میزان دقت و زمان مدل آموزشی و حافظه صرف شده روی‌داده‌ها به دست آورد. بالاتر بودن درصد دقت طبقه‌بندی و کمتر بودن زمان وحافظه صرف شده بیانگر سریع بودن مدل آموزشی در داده‌ها خواهد بود.

کلمات کلیدی: جریان داده، طبقه‌بندی، داده‌کاوی، درخت هافدینگ

۱–مقدمه

پیشرفت‌های اخیر در فنّاوری سخت‌افزار به ما اجازه داده است تا اطلاعات روزمره خویش را با سرعت بالایی ثبت نماییم. این فرایندها منجر به ظهور حجم وسیعی از داده‌ها گشته که با سرعت نامعلوم رو به افزایش هستند. پردازش این داده‌ها اشاره به داده‌های جریانی دارد. یک داده جریانی دنباله‌ای از نمونه x1,..., xn است که باید به ترتیب مورد دسترسی قرارگرفته و می توان تنها یک‌بار یا تعداد بسیار کمی آن‌ها خواند. هر خواندن از دنباله یک اسکن خطی یا گذر نامیده می‌شود. مدل داده‌های جریانی با پیدایش برنامه‌های شامل مجموعه داده‌های بزرگ مورد توجه قرار گرفت. برای مثال، عملیات کارت اعتباری، شبکه‌های حسگر ، خدمات مخابراتی، ثبت وقایع و کاوش پرس‌وجو جریانی و نظارت بر شبکه و شبکه‌های اجتماعی جریانی

و رهگیری کلیک کاربران اینترنت، مجموعه‌های بزرگ صفحات وب، داده‌های چندرسانه‌ای، معاملات مالی و داده‌های گردآوری‌شده علمی، بهترین مدل‌های داده‌های جریانی می‌باشند[2] ,[1]. این مجموعه داده‌ها بیش از اندازه بزرگ هستند که تا در حافظه اصلی کامپیوتر جای بگیرند و لذا معمولاً در حافظه ثانویه قرار دارند. اسکن‌های خطی، روش‌های پرهزینه پردازش این داده‌ها هستند و دسترسی تصادفی به‌طور اجتناب ناپذیری پرخرج است. برخی داده‌ها از قبیل آمار بسته‌های مسیریاب‌ها، اطلاعات هواشناسی و داده‌های شبکه‌های حسگر، زودگذر بوده و نیازی نیست آن‌ها را بر روی دیسک ذخیره کرد، این داده‌ها باید در همان لحظه تولید، مورد پردازش قرارگرفته و بعد از ایجاد خلاصه‌ای از آن‌ها، دور انداخته شوند. زمانی که اندازه این داده‌ها از حافظه اصلی در اختیار الگوریتم تجاوز کرد، این امکان برای الگوریتم‌های داده‌های جریانی وجود ندارد که داده‌های اسکن شده درگذشته دور را به خاطر بیاورند. این محدودیت حافظه، ما را ناگزیر به طراحی انواع جدیدتری از الگوریتم‌ها ساخته است که تنها خلاصه‌ای از داده‌های گذشته را ذخیره نموده و بخشی از حافظه را برای پردازش آینده خالی نگه می‌دارند. هراسکن از داده‌های بزرگ بر روی یک دستگاه کند بسیار هزینه‌بر است، درصورتی‌که معیار ارزیابی کارایی الگوریتم‌های داده‌های جریانی علاوه بر زمان اجرا و حافظه مصرفی مبتنی بر تعداد اسکن‌های خطی است. در حالت داده‌های جریانی زودگذر، تنها یک اسکن امکان‌پذیر است.مدل داده‌های جریانی و مدل افزایشی برخط) از جهت اینکه هر دو قبل از در اختیار داشتن همه داده‌ها، تصمیم‌گیری‌می‌کنند، شبیه به یکدیگر هستند. اگرچه این مدل‌ها کاملاً مشابه نیستند.

- ارائه مدل ترکیبی طبقه‌بندی سریع جریان داده‌ها با استفاده از الگوریتم درخت هافدینگ طبقه‌بندی و جستجوی سریع جریان داده نیز یک مسئله باز است و در این مقاله از روش کاوش داده‌های جریانی برای طبقه‌بندی سریع جریان داده با الگوریتم درخت هافدینگ موردبررسی قرارگرفته است. طبقه‌بندی داده‌های جریانی روشی برای استخراج دانش و اطلاعات از نقاط داده‌ای مستمر است.جریان داده‌ها ویژگی‌های بسیار متنوعی نسبت به داده‌های بانک‌های اطلاعاتی ایستای متداول دارند، ازجمله اینکه پویا، نامحدود، چندبعدی، منظم، غیرتکراری، با سرعت‌بالا و متغیر با زمان هستند[۳]. مقادیر زیاد و سرعت بالای جریان داده‌ها ذخیره داده‌ها را برای تجزیه و تحلیل‌های بعدی غیرممکن ساخته است،بنابراین داده‌ها باید در هنگام ورود به‌صورت مستقیم پردازش شوند[۴]. الگوریتم‌های کاوش جریانی با چالش‌های زیادی از قبیل: طبیعت بسیار سریع بودن جریان داده‌ها(پاسخ در زمان واقعی)، نیازمندی‌های حافظه‌ای نامحدود محدودیت حافظه)، تغییر مفهوم یا(مفهوم رانش)، مصالحه بین دقت و کارایی روبه‌رو هستند. داده جریانی تعدادی از ویژگی‌های متمایز خواهد بود که باعث می‌شود مقدار حافظه کش سیستم با حافظه موجود برای ذخیره‌سازی جریان داده‌های کل مناسب نباشد. مشکل اصلی داده‌های جریانی سرعت است که در آن داده‌های جریانی در دسترس است بسیار سریع‌تر از اینکه داده بتواند پردازش و ذخیره شود. درحالی‌که روش‌های بسیاری تاکنون پیشنهاد شده تا این مشکلات را برطرف نمایند، اما اغلب آن‌ها قادر نیستند تمام این چالش‌ها و محدودیت‌ها را پاسخگو باشند. با توجه به این چالش‌ها از مصالحه بین دقت و کارایی و کیفیت بهبود که یکی از ویژگی‌هایی اصلی است که هر الگوریتم داده‌کاوی بر روی جریان داده‌ها باید حتماً به آن دقت شود.
- در این مقاله بر روی موضوع ارائه مدل ترکیبی طبقه‌بندی سریع جریان داده‌ها با استفاده از الگوریتم درخت هافدینگ که به بحث طبقه‌بندی سریع جریان داده‌ها با استفاده از الگوریتم درخت هافدینگ می‌پردازیم، متمرکزشده [4]–[8] و هدف پژوهش این است با بررسی و ارائه مدلی آموزشی توسط الگوریتم درخت هافدینگ طبقه‌بندی سریع جریان داده را برای معیارهای دقت و کیفیت و زمان مورد ارزیابی قرار گیرد:.که طبقه‌بندی سریع جریان داده توسط الگوریتم درخت هافدینگ را به‌طور خودکار در طبقه‌بندی دیگر بررسی شود. این نوع داده‌ها به شکل دنباله‌های مرتب و بالقوه نامحدود از نقاط داده‌ای هستند کهنوعاً توسط فرآیند مولد داده غیرایستا ایجاد می‌شوند.چنین داده‌های به‌طور پیوسته و در جریان زمان تولید می‌شوند و باید خیلی سریع و به‌صورت بلادرنگ پردازش شوند.

- نتایج اولیه و بررسی‌های مقدماتی آزمایش‌ها و تحلیل‌های نشان داده شده است که به‌کارگیری روش یادشده منجر به رسیدن به‌دقت بالاتر و همچنین حد هافدینگ بهتر می‌گردد. در این پژوهش در درجه اول نتایج حاصل از طبقه‌بندی داده جریانی با الگوریتم درخت هافدینگ را تحلیل و سپس مورد ارزیابی قرارگرفت. روش پیشنهادی مدل خود بر اساس پارامترهایی که مشخص‌شده و با توجه به هدف اصلی در این پژوهش ارائه مدلی آموزشی توسط الگوریتم درخت هافدینگ است، به‌دقت و کیفیت بهبود و زمان این مدلی که ارائه شده به این دقت و کیفیت و زمان رسیده است.

۳- داده جریانی

دنباله نامتناهی از اقلام داده معمولا با استفاده از برچسب زمانی یا شاخص تعریف می‌شود را داده جریانی می‌گویند. داده جریانی شامل سری زمانی، و داده‌های توالی [۹] می‌باشد. یک داده جریانی در واقع یک توالی از داده‌هایی به شکل $(x_1,\ldots,x_n)$ می‌باشد، که هرکدام از x_i ها یک خصوصیت دارای مقدار می‌باشد [۱۰].

۲-۱- طبقه‌بندی داده‌های جریانی

طبقه‌بندی در دو مرحله انجام می‌شود: گام آموزش و گام تست. در مرحله یادگیری، سیستم سعی می‌کند برای یادگیری مدل از مجموعه ای از اشیا داده به نام مجموعه آموزشی استفاده کند. در مرحله آزمایش، یک مدل برای اختصاص یک برچسب کلاس برای اشیا داده‌ها بدون برچسب در مجموعه داده آزمایش استفاده می‌شود. درخت تصمیم، طبقه‌بندی بیزین، شبکه عصبی، بردار ماشین پشتیبان، نزدیکترین همسایه، طبقه‌بندی گروهی، تکنیک‌های طبقه‌بندی برای داده‌های جریانی می‌باشند که در ادامه به معرفی درخت‌تصمیم‌گیری برای جریان داده‌ها و درخت تصمیم هافدینگ می‌پردازیم [۲].

۲-۳- درخت‌های تصمیم‌گیری برای جریان داده

دو فاز در فرآیند کلاسه‌بندی وجود دارد: ساخت مدل و استفاده از آن. فاز دوم یا فاز استفاده از مدل با جریان داده مطابقت دارد و با رسیدن یک داده جدید کلاس آن مشخص می‌شود. در طول فاز اول یک مدل بر اساس مجموعه آموزشی که کلاس‌های شناخته شده دارند ساخته می شود. در حالت‌های متعارف مجموعه آموزشی و داده‌هایی که کلاس نامشخص دارند در یک پایگاه‌داده ایستا قرار دارند. اکثر روش‌های کلاسه‌بندی مجموعه آموزشی را چندین بار پیمایش می کنند که این باعث کندی فرایند می شود. در جریان داده، داده‌ها بسرعت در جریان هستند و ذخیره آنها برای چند بار پیمایش مناسب نیست. از سوی دیگر در جریان داده داده‌ها بصورت پیوسته می‌رسند. به همین جهت باید روش‌های کلاسه‌بندی جدیدی را ارایه دهیم که قادر به روبرویی با مقدار بسیار زیاد جریان داده و سرعت بالای آنها باشند. در ادامه یک ساختار برای‌کلاسه‌بندی جریان داده ارایه داده و یک بهینه‌سازی برای آن ارایه می‌دهیم [۵].یک سیستم یادگیرنده درخت تصمیم‌گیری که بر اساس الگوریتم درخت Hoeffding قابل پیاده‌سازی استVFDTمی‌باشد. VFDT از Information gain یا Gini Index به عنوان معیار ارزیابی صفات استفاده می‌کند و شامل بهبودهایی بر الگوریتم ارائه شده برای Hoeffding است[۷] .درختHoeffding یک درخت تصمیم‌گیری برای داده‌های‌جریانی می‌باشد[5]. درخت تصمیم‌گیری سنتی نیاز به اسکن داده‌آموزش در خیلی از زمان ها با انتخاب ویژگی‌های تقسیم دارد. به هر حال، این نیاز در یک محیط داده‌های جریانی غیرممکن است[۵]. Hoeffding برای انتخاب یک ویژگی تقسیم بهینه در مقدار کافی از اشیا داده به‌صورت محدود استفاده می‌شود5]. الگوریتم درخت Hoeffding یک الگوریتم افزایشی است که باعث برآورده کردن نیاز محدودیت تک پاس-Single

Pas [۱], [۳], [۵], [۶]. برای کاوش درداده‌های‌جریانی می‌شود. برای هر ورود داده جدید، الگوریتم درخت Hoeffding ازمحدودیت ا Hoeffding برای بررسی اینکه آیا بهترین ویژگی‌تقسیم به اندازه کافی اعتماد به نفس به ایجاد گره درخت سطح بعدی می‌دهد، استفاده می‌کند[۷]. طبقه‌بندی داده‌های جریانی از طبقه‌بندی داده سنتی به ارث برده است و رویکرد محاسباتی و پنجره‌های زمان مختلف به کار برده است. درخت تصمیم بسیار سریع قابل تطبیق با مفهوم توسعه یافته الگوریتم درخت تصمیم بسیار سریع است . این الگوریتم با سازگار نگه داشتن قالب خود با پنجره لغزانی از نمونه داده‌ها ازتغییر مفهوم در توزیع داده‌ها حمایت می‌کند[۷]. درخت، VFDT یا (درخت تصمیم‌گیری بسیارسریع) یک فرمت مناسب از درخت‌های تصمیم‌گیری برای داده‌های جریانی است[۹]. VFDT (درخت تصمیم‌گیری بسیارسریع) با استفاده از Hoeffding محدود به ساخت یک گره درخت وقتی که مقدار کافی از داده‌ها را دارد، به دنبال این رویکرد VFDT را بر روی Sliding Window اجرا می‌کند تا همواره به روزترین طبقه‌بندی را داشته باشد[۵]. هنگامی که تغییر مفهومی اتفاق افتد، مقدار معیارهای جداکننده شاخه‌های درخت بصورت قابل توجهی تغییر می‌کنند[۱۶]. در الگوریتمVFDT بسط داده شده است تا قادر باشد که بصورت کارا، خصوصیات عددی نیز پردازش نماید،آن یک رویکرد یادگیری افزایشی پنجره برجسته می‌باشد [17].CVFDT نسخه بهبود یافته(درخت تصمیم‌گیری بسیارسریع) VFDT که تواند با مفهوم رانش با ساخت درخت‌های جایگزینی سازگار باشد[۵]. مساله دشواری که اینجا وجود دارد این است که چه تعداد نمونه در هر گره برای شکاف لازم است ؟ این مساله با یک نتیجه آماری حل می‌شود که به نام Hoeffding Bound شـناخته می‌شود.

۲–۴– روند شکل گیری درخت هافدینگ

برای یافتن بهترین صفت در هر گره، در نظر گرفتن یک زیر مجموعه کوچک از نمونه‌های آموزشی که از آن گره عبور می‌کنند کافی است. با در دست داشتن جریانی از نمونه‌ها، اولین نمونه‌ها برای انتخاب صفت ریشه استفاده می‌شوند. با تعیین شدن صفت ریشه، نمونه‌های بعدی به سمت پایین و برگ‌های مربوطه عبور داده می‌شوند تا برای انتخاب صفت در آنجا استفاده شوند. این عمل به‌صورت بازگشتی تکرار می‌شود.چه تعداد نمونه در هر گره لازم است ؟ یک مmتغییر تصادفی با نام r که دارای مقادیر حقیقی و بـرد R اسـت در نظـر بگیریـد(مـثلا بـرای احتمالات برد برابر 1 است). فرض کنید که ما n مشاهده مستقل از این متغیر انجـام مـی‌دهـیم و میانگین $\bar{r}$ آنها را محاسبه می کنیم. Hoeffding Bound (حد هافدینگ) نشان می‌دهـد کـه میـانگین واقعـی‌متغیر r بعد از این n مشاهده با احتمال δ-1 حداقل برابر ε- $\bar{r}$ است که در آن:

$$\varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}} \qquad (۱)$$

۲–5– معیارهای انتخاب صفت

معیارهای مختلفی برای تعیین صفاتی که شکاف باید بر اساس آن انجام شود، وجود دارد از جمله:

- بهره اطلاعاتی
- نسبت بهره
- شاخص چینی

2–6– بهره اطلاعاتی

اطلاعات مورد نیاز برای طبقه‌بندی یک تاپل در D برابر است با:

$$Info(D) = -\sum_{I=1}^{M} pi\ log2\ (pi) \qquad (2)$$

اطلاعات مورد نیاز برای کلاسه‌بندی‌یک تاپل از D برحسب صفت A برابراست با :

$$\mathrm{InfoA}(D) = \sum_{j=1}^{V} \frac{|\mathrm{Dj}|}{|D|} * info(\mathrm{Dj}) \qquad (3)$$

عبارت $|D_j|/|D|$ در واقع وزن بخش j را نشان می‌دهد. اطلاعات حاصل از انشعاب برحسب صفت A را به صورت زیر تعریف می‌کنیم:

$$\mathrm{Gain(A)= Info(D)- Info}_{A}\,(D) \qquad (4)$$

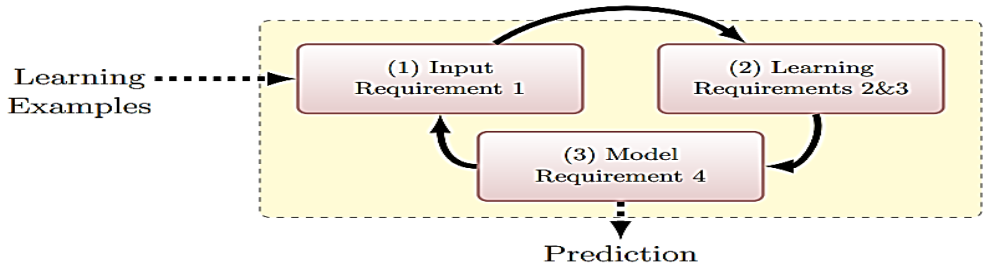
هر چه مقدار بهره صفت A (Gain(A)) بیشتر باشد یا به عبارت دیگر هر چه $\mathrm{Info}_A(D)$ کمتر باشد صفت A به عنوان صفت شکاف انتخاب [۱۲]می شود.

۳- مدل‌سازی جریان‌داده

برای ارزیابی جریان داده‌ها محیط جریان داده‌ها دارای نیازهای متفاوتی از محیط سنتی است ، که مهم‌ترین آن‌ها به شرح زیر است:

۱- در یک زمان و تنها در یک فاز باشد.
۲- از یک مقدار محدود حافظه استفاده می‌کند.
۳- کار در یک مقدار محدود از زمان انجام شود.
۴- آماده برای پیش‌بینی در هر زمان باشد.

آموزش طبقه‌بندی درخت‌تصمیم حد هافدینگ به‌عنوان یک مدل برای روش پیشنهادی می‌باشد. طبقه‌بندی جریان داده به وسیله یک چرخه سه مرحله‌ای انجام می‌شود[۱۳].



شکل (۱) چرخه سه مرحله‌ای طبقه‌بندی جریان داده [۱۳]

۴- ارزیابی و بررسی نتایج

جریان داده یک توالی نامحدود و حجیم از عناصر داده‌ای است که متولیاً با سرعت زیاد تولید می‌شوند. به دلیل ویژگی‌های ذکرشده امکان ذخیره‌سازی تمام داده‌های جریان وجود ندارد، درنتیجه جریان‌های داده باید به‌صورت روی خط پردازش شوند. در روش‌های پردازش جریان‌های داده لازم است به سه نکته توجه شود. هر داده ورودی باید حداکثر یک‌بار آنالیز شود باوجود تولید ادامه‌دار داده‌ها، باید حافظه محدودی برای پردازش جریان‌های داده در نظر گرفته شود .داده‌های جدید باید با حداکثر سرعت پردازش شوند و آنالیزهای آن‌ها برای به‌روزرسانی نتایج استفاده شود، درنتیجه خروجی دقیق و بروز در هرلحظه ارائه گردد. موقعی از روش داده‌های جریانی استفاده می‌کنیم که داده‌های ورودی حجیم هستند; یعنی کل داده‌ها در حافظه اصلی قابل بارگذاری نیستند. مدل درخت تصمیم با حد هافدینگ یک روش داده جریانی است که بدون اینکه کل داده‌ها در حافظه اصلی بارگذاری شوند داده‌های ورودی به‌صورت جریانی از تاپل یا تراکنش‌ها وارد هر نود درخت می‌شوند; یعنی یک‌بخشی از داده‌های ورودی بارگذاری می‌شوند و اندازه آن بخش‌بر اساس حد

هافدینگ محاسبه می‌گردد. بنابراین کل داده‌ها یک‌بار اسکن می‌شوند و درنتیجه درخت سریعتر ساخته می‌شود. روش درخت تصمیم معمولی برای ساخت مدل آموزشی نیاز دارد که کل داده‌ها را در حافظه اصلی بارگذاری کند و با چند بار اسکن کل داده‌ها مدل را بسازد، بنابراین داده‌ای که حجیم است کل آن قابل بارگذاری در حافظه نیست. اگر درخت تصمیم را بدین روش بسازیم زمان بسیار زیادی طول می‌کشد، اما چرا طبقه‌بندی سریع که در عنوان مقاله است آن بدین خاطر است که نمی‌تواند کل داده‌ها در داخل حافظه بارگذاری شود، اگر از روش درخت تصمیم معمولی برای طبقه‌بندی استفاده کنید ساخت مدل زمان طولانی طول می‌کشد، بنابراین برای طبقه‌بندی سریع از روش درخت تصمیم با حد هافدینگ استفاده می‌شود تا ساخت درخت تصمیم سریع‌تر انجام گیرد. برای ارزیابی روش پیشنهادی در این مقاله از نرم‌افزار MOA (Massive Online Analysis) به معنی تجزیه وتحلیل زمان واقعی برای جریان داده‌ها است Java8 version moa-release-2014.11 استفاده شده است. این نرم‌افزار که مخصوص داده‌های جریانی توسعه داده شده است از ویژگی‌های این نرم‌افزار متن باز(Open Source) بودن آن می‌باشد. این نرم‌افزار تحت زبان جاوا پیاده سازی شده است و قابل استفاده درسیستم عامل‌های Windows,Mav,Linux می‌باشد.کلیه‌ی آزمایشات بر روی یک کامپیوتر شخصی با پردازنده Intel® Core(TM) i7-4510u cpu @2.00GHZ و حافظه اصلی ۸ GB با سیستم عامل Windows 8 انجام شده است.

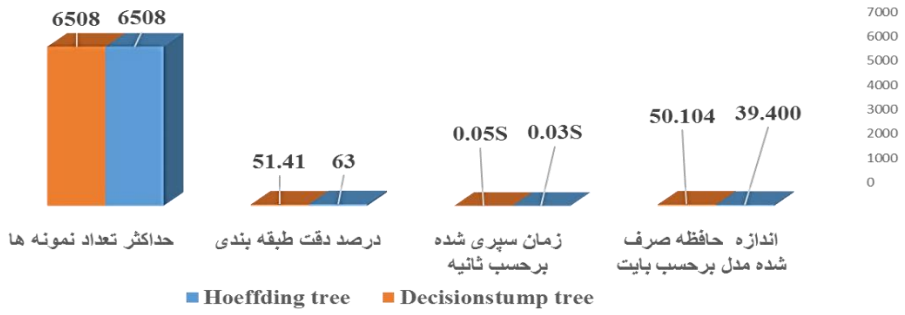
۵– معرفی ویژگی های داده نشریات

این داده‌ها شامل ۶۵۰۸ رکورد در ۵ صفت (فیلد) می باشد.این دیتاست شامل تعدادی تاپل می باشد،که هر تاپل شامل ۵ ویژگی زیر است. ویژگی‌های استفاده شده در این تحقیق شامل موضوع ، محل ناشر، محل نشر، سال نشر، مجوزها، است. ویژگی ناشر یا publisher type متغییر هدف در این پژوهش می باشد.

۶– مقایسه نتایج مدل‌های آموزشی درخت های تصمیم روش پیشنهادی

جدول (۱) مقایسه نتایج مدل های آموزشی درخت تصمیمDecision stump , Hoeffding در MOA

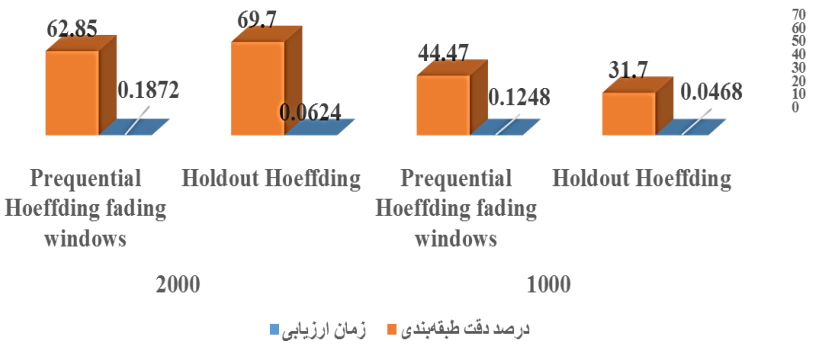
ردیف	نام الگوریتم آموزش	حداکثر تعداد نمونه ها	درصد دقت طبقه بندی	زمان سپری شده برحسب ثانیه	اندازه حافظه صرف شده مدل برحسب(بایت)
1	Hoeffding tree	6508	63	0.03 s	39.400
2	Decision stump tree	6508	51.41	0.05 s	50.104



نمودار(۱) مقایسه نتایج ارزیابی دو مدل یادگیری Decision stump , Hoeffding توسط لایهHeld out test

جدول (۲) مقایسه نتایج دو روش Holdout Hoeffding ,Prequential Hoeffding fading windows

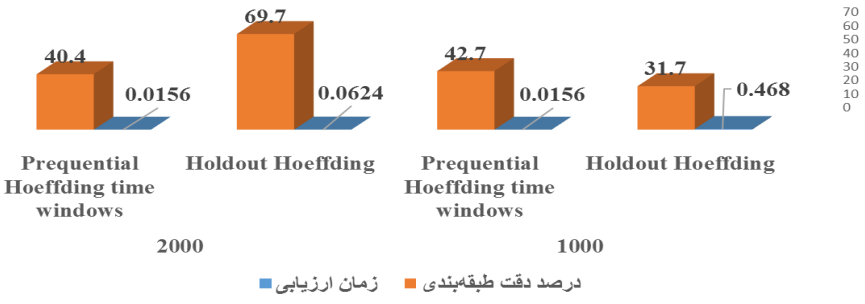
نمونه‌های ارزیابی شده	نام الگوریتم	زمان ارزیابی	درصد دقت طبقه‌بندی
1000	Holdout Hoeffding	0.0468	31.7
	Prequential Hoeffding fading windows	0.1248	44.47
2000	Holdout Hoeffding	0.0624	69.7
	Prequential Hoeffding fading windows	0.1872	62.85



نمودار(۲) مقایسه نتایج دو روش Holdout Hoeffding ,Prequential Hoeffding fading windows

جدول (۳) مقایسه نتایج دو روش Hold out Hoeffding ,Prequential Hoeffding time windows

نمونه‌های ارزیابی شده	نام الگوریتم	زمان ارزیابی	درصد دقت طبقه‌بندی
1000	Holdout Hoeffding	0.468	31.7
	Prequential Hoeffding time windows	0.0156	42.7
2000	Holdout Hoeffding	0.0624	69.7
	Prequential Hoeffding time windows	0.0156	40.4



نمودار(۳) مقایسه نتایج دو روش Hold out Hoeffding , Prequential Hoeffding time window

۷- نتیجه‌گیری

در این مقاله روشی برای طبقه بندی داده های جریانی با استفاده از الگوریتم درخت تصمیم هافدینگ ارایه شد.در این مقاله ابتدا به معرفی داده های جریانی و درخت تصمیم با حد هافدینگ پرداختیم. نتایج انجام شده به ما نشان داد که درخت تصمیم حد هافدینگ زمان آموزش و آزمایش حداقل، سرعت بالاتر در پیمایش داده ها، میزان مصرف حافظه کم، دقت بالاتر در ارزیابی نسبت الگوریتم درخت تصمیم Decision دارد. بنابراین می توان با توجه به معیارهای گفته شده در بالا نتیجه گرفت که درخت تصمیم حد هافدینگ یک روش سریع برای طبقه بندی مبتنی بر داده نشریات(داده جریانی) می باشد.

۸- مراجع

[1]. A. Kumar, P. Kaur, and P. Sharma, "A Survey on Hoeffding Tree Stream Data Classification Algorithms," CPUH-Research J., vol. 1, no. 2, pp. 28–32, 2015.

[2]. H. L. Nguyen, Y. K. Woon, and W. K. Ng, "A survey on data stream clustering and classification," Knowl. Inf. Syst., vol. 45, no. 3, pp. 535–569, 2015.

[3]. D. Farid, L. Zhang, A. Hossain, and C. Mofizur, "An Adaptive Ensemble Classifier for Mining Concept-Drifting Data Streams," 2013.

[4]. P. M. Gonçalves Jr and R. S. M. de Barros, "RCD: A recurring concept drift framework," Pattern Recognit. Lett., vol. 34, no. 9, pp. 1018–1025, 2013.

[5] P. Domingos and G. Hulten, "Mining High-Speed Data Streams," Proc. Sixth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 71–80, 2000.

[6]. G. Dong, J. Han, P. S. Yu, L. V. S. Lakshmanan, J. Pei, and H. Wang, "Online Mining of Changes from Data Streams : Research Problems and Preliminary Results," ACM SIGMOD MPDS '03 San, pp. 11–13, 2003.

[7]. G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," Proc. seventh ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '01, vol. 18, pp. 97–106, 2001.

[8]. R. Tripathi and S. K. Dwivedi, "A Quick Review of Data Stream Mining Algorithms," Imp. J. Interdiscip. Res., vol. 2, no. 7, pp. 870–873, 2016.

[9]. M. S. B. PhridviRaj and C. V. GuruRao, "Data Mining – Past, Present and Future – A Typical Survey on Data Streams," Procedia Technol., vol. 12, pp. 255–263, 2014.

[10]. C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "Data Streams: Models and Algorithms," pp. 9–38, 2007.

[11]. I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with drifting streaming data," IEEE Trans. Neural Networks Learn. Syst., vol. 25, no. 1, pp. 27–

[12]. C. C. Aggarwal, "Scientific Data Mining and Knowledge Discovery," pp. 377–397, 2010.

[13]. A. Bifet, R. Kirkby, P. Kranen, and P. Reutemann, "Massive online analysis manual," Univ. Waikato, New Zeal. Cent. Open Softw. Innov., no. March, 2009.

[14] J. Gama, "Data Stream Mining: the Bounded Rationality," Informatica, vol. 37, pp. 21–25, 2013.

[15]. S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. OCallaghan, "Clustering data streams: Theory and practice," Knowl. Data Eng. IEEE Trans., vol. 15, no. 3, pp. 515–528, 2003.

[16]. R. Jin and G. Agrawal, "Efficient Decision Tree Construction on Streaming Data," Proc. Ninth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 571–576, 2003.

[17]. W. Fan, "Systematic data selection to mine concept-drifting data streams," Proc. 2004 ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '04, p. 128, 2004.