مجله محاسبات نرم، جلد ۱۲، شماره ۲ (پاییز و زمستان ۱۴۰۲)، صفحه ۲ـ۱۵
https://doi.org/10.22052/scj.2023.248781.1124

مجله محاسبات نرم
SOFT COMPUTING JOURNAL
تارنمای مجله: scj.kashanu.ac.ir

دانشگاه کاشان
University of Kashan

طراحی مدلی برای طبقه‌بندی داده‌های جریانی با استفاده از یادگیری تقویتی و گرادیان کاهشی تصادفی ✧

سمیرا فرزانه[1]، دانشجوی کارشناسی ارشد، جواد سلیمی سرتختی[1*]، استادیار
[1] گروه کامپیوتر، دانشکده مهندسی برق و کامپیوتر، دانشگاه کاشان، کاشان، ایران.

اطلاعات مقاله

تاریخچه مقاله:
دریافت ۱۴ دی ماه ۱۴۰۱
پذیرش ۱۰ تیر ماه ۱۴۰۲

کلمات کلیدی:
داده‌های جریانی
صحت و زمان اجرا
گرادیان کاهشی تصادفی
یادگیری افزایشی
یادگیری تقویتی

چکیده

حجم وسیعی از تحقیقات در زمینه یادگیری برخط به مساله غلبه بر فراموشی فاجعه‌بار تمرکز کرده‌اند و تحقیقات اندکی در زمینه طبقه‌بندی داده‌های جریانی با صحت و زمان اجرای مناسب تمرکز کرده‌اند. از سوی دیگر، به دلیل حجم و نوع داده‌های جریانی بسیاری از الگوریتم‌های سنتی یادگیری ماشین به خودی خود کارایی لازم هنگام مواجه با آنها را ندارند. بنابراین، در این مقاله برای طبقه‌بندی داده‌های جریانی با صحت و زمان یادگیری مناسب یک مدل جدید با استفاده از یادگیری تقویتی و الگوریتم گرادیان کاهشی تصادفی ارائه شده است. یکی از قابلیت‌های مهم یادگیری تقویتی این است که عامل می‌تواند رفتار خود را به تدریج با تغییراتی که رخ می‌دهد سازگار کند و به صورت تدریجی بر دانش قبلی خود بیافزاید. در این پژوهش به دلیل استفاده از یادگیری تقویتی و تعریف پاداش، عامل عملکرد بهتری در محیط دارد. الگوریتم پیشنهادی بر روی داده‌های مختلف از جمله مجموعه داده جریانی تشخیص فعالیت‌های انسانی آزمایش شده و از لحاظ صحت و زمان اجرا با چندین الگوریتم افزایشی مقایسه شده است. طبق نتایج آزمایشگاهی الگوریتم پیشنهادی بهترین کارایی را هم از نظر صحت و هم از نظر زمان اجرا در مقایسه با سایر الگوریتم‌های افزایشی دارد.

© ۱۴۰۲ نویسندگان. مقاله با دسترسی آزاد تحت مجوز CC-BY

۱. مقدمه

امروزه حجم عظیمی از داده‌ها در حوزه‌ها و زمینه‌های مختلف تولید می‌شوند و اغلب داده‌های تولید شده به صورت جریانی هستند. داده‌های جریانی[1] داده‌هایی هستند که در طول زمان تولید می‌شوند و در هر لحظه دسترسی به همه داده‌ها وجود ندارد [۱]. از این‌رو، حجم داده‌های تولید شده و نوع آنها باعث به وجود آمدن چالش‌هایی شده‌اند. یکی از مسائل چالش برانگیز، مساله طبقه‌بندی کردن داده‌های جریانی است.

در راستای طبقه‌بندی کردن داده‌های جریانی با صحت مناسب الگوریتم‌های سنتی یادگیری ماشین با مشکل مواجه می‌شوند، زیرا در الگوریتم‌های سنتی فرض بر این است که تمامی داده‌ها به صورت یک‌جا و با توزیع یکسان در اختیار مدل قرار دارند و مدل می‌تواند از تمامی داده‌های جمع‌آوری شده در راستای دستیابی به بیشترین صحت طبقه‌بندی استفاده نماید. اما داده‌های جریانی در طول زمان، در حجم و سرعتی بالا تولید می‌شوند. به عنوان مثال، می‌توان به داده‌هایی که در کاربردهای اینترنت اشیا، لاگ‌های امنیتی تولید می‌شوند، اشاره نمود [۱]. از سوی دیگر،

✧ نوع مقاله: پژوهشی

* نویسنده مسئول
پست(های) الکترونیک: samira.farzaneh@grad.kashanu.ac.ir (فرزانه)
salimi@kashanu.ac.ir (سلیمی سرتختی)

[1] Data Stream

نحوه ارجاع به مقاله: سمیرا فرزانه و جواد سلیمی سرتختی، «طراحی مدلی برای طبقه‌بندی داده‌های جریانی با استفاده از یادگیری تقویتی و گرادیان کاهشی تصادفی»، مجله محاسبات نرم، جلد ۱۲، شماره ۲، صفحات ۲ـ۱۵، پاییز و زمستان ۱۴۰۲.

داده‌های جریانی عموما همراه با تغییر توزیع هستند؛ به این معنا که داده‌ها تا یک زمان خاص از یک توزیع مشخص تبعیت می‌کنند اما به تدریج توزیع داده‌ها تغییر می‌کند. این ویژگی سبب می‌شود که الگوریتم‌های سنتی یادگیری ماشین به خودی خود کارایی لازم را هنگام مواجهه با آنها نداشته باشند و دچار بیش برازش شوند. بدین ترتیب، برای طبقه‌بندی نمودن داده‌های مذکور الگوریتم‌های یادگیری افزایشی[1] مطرح شده است [۲]. این نوع الگوریتم اغلب برای کلان داده‌ها اعمال می‌شود. در این نوع یادگیری مرحله آموزش و یادگیری در طول زمان و به طور مداوم صورت می‌گیرد. الگوریتم گرادیان کاهشی تصادفی[2] قدمت طولانی در حوزه یادگیری ماشین دارد و توسط بهینه‌ساز گرادیان کاهشی بهینه شده است [۳]. الگوریتم مذکور برای حل انواع مسائل مختلف همانند طبقه‌بندی داده‌ها و پردازش متن استفاده می‌شود، اما الگوریتم مذکور هنگام مواجه با داده‌های جریانی کارایی مناسبی ندارد.

به صورت کلی، بهترین حالت از نظر صحت طبقه‌بندی داده‌های جریانی حالتی است که با ورود هر نمونه جدید، مدل بازآموزش شود. اما چنین روشی بسیار زمانبر و غیرممکن است. از این‌رو، برای طبقه‌بندی نمودن داده‌های فوق‌الذکر با صحت و زمان یادگیری مناسب با دو چالش روبه‌رو هستیم که عبارتند از: (۱) انتخاب نمونه‌های بهینه جهت کاهش زمان اجرا و (۲) پیروی داده‌های جریان‌های مختلف از توزیع‌های متفاوت. بنابراین، سوالی که مطرح می‌شود این است که آیا با ورود هر نمونه جدید باید مدل بازآموزش شود یا خیر؟

در این مقاله قصد بر این است که به چالش‌ها و سوال ذکر شده پرداخته شود و مدلی مبتنی بر یادگیری تقویتی به همراه دسته‌بند گرادیان کاهشی تصادفی تحت عنوان Inc-RL-SGD برای دستیابی به صحت و زمان اجرای مناسب در راستای طبقه‌بندی داده‌های جریانی به منظور یادگیری افزایشی طراحی شود تا هر نمونه‌ای که به عنوان ورودی برای مدل در نظر گرفته شد را بررسی کند که آیا نمونه به مدل اضافه شود و مدل مجدد اجرا شود یا خیر. در مدل پیشنهادی از چندین مجموعه داده متعلق به بانک داده UCI و مجموعه داده‌های جریانی تشخیص فعالیت‌های انسانی برای آزمایش عملکرد مدل استفاده شده است. شایان ذکر است، عملکرد مدل پیشنهادی با سایر الگوریتم‌های یادگیری افزایشی مقایسه شده و نشان داده شده است که مدل پیشنهادی نسبت به سایر الگوریتم‌های افزایشی دارای عملکرد بهتری است.

ادامه مقاله به صورت زیر سازماندهی شده است. در بخش دوم به تعریف مفاهیم اولیه پرداخته شده، در بخش سوم به طور مختصر کارهای گذشته مرور می‌شود. در بخش چهارم الگوریتم پیشنهادی را شرح داده و در بخش پنجم به ارزیابی آن پرداخته می‌شود. در نهایت در بخش پایانی نتیجه‌گیری ارائه می‌شود.

۲. مفاهیم پایه

به منظور طبقه‌بندی کردن داده‌های جریانی با صحت و زمان اجرای مناسب از الگوریتم‌های سنتی یادگیری ماشین نمی‌توان استفاده نمود. بنابراین، مدلی مبتنی بر یادگیری تقویتی به همراه دسته‌بند گرادیان کاهشی تصادفی به منظور یادگیری افزایشی طراحی شده است. در ادامه تمامی موارد فوق به تفصیل توضیح داده شده است.

۱.۲. گرادیان کاهشی تصادفی

گرادیان کاهشی تصادفی یک الگوریتم بهینه‌سازی ساده و در عین حال کارآمد برای بهینه‌سازی تابع زیان[3] است که این روش یک تقریب تصادفی از روش گرادیان کاهشی است [۳]. تابع زیان، معیاری برای سنجش مناسب بودن مدل از نظر قابلیت و توانایی در پیش‌بینی مقدارهای جدید است [۴]. انتخاب تابع زیان مناسب به عوامل متعددی نظیر وجود نقاط یا داده‌های پرت، نوع الگوریتم یادگیری ماشین و عوامل دیگر بستگی دارد. در الگوریتم مذکور به ازای تمامی نمونه‌هایی که در مجموعه داده آموزشی وجود دارند، میانگین گرادیان طبق رابطه (۱) به دست آمده و برای هر تکرار t گرادیان تخمین زده می‌شود.

[1] Incremental Learning
[2] Stochastic Gradient Descent
[3] Loss Function

$$g_{GD} = \frac{1}{n}\sum_{i=1}^{n} \nabla_{\theta}\mathcal{L}\left(x^{i}, y^{i}, \theta_{t}\right) \qquad (۱)$$

که n برابر با تعداد کل نمونه‌های مجموعه آموزشی و $\mathcal{L}$ تابع زیانی است که باید کمینه شود.

همان‌گونه که در رابطه (۱) مشاهده می‌گردد، برای تخمین گرادیان باید میانگین همه نمونه‌ها محاسبه شود. بنابراین، اگر داده‌ها از نوع کلان داده باشند، محاسبه گرادیان به روش ذکر شده در بالا بسیار زمانبر است. برای ممانعت از بروز چنین مشکلی گرادیان کاهشی تصادفی مطرح شد. در این نوع گرادیان طبق رابطه (۲) در هر تکرار برای دسته‌ای از نمونه‌ها به جای کل نمونه‌ها، گرادیان تخمین زده می‌شود.

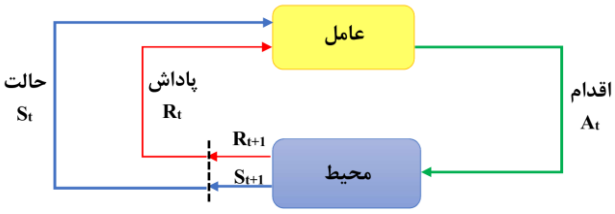
$$g_{GD} = \frac{1}{n'}\sum_{i=1}^{n'} \nabla_{\theta}\mathcal{L}\left(x^{i}, y^{i}, \theta_{t}\right) \qquad (۲)$$

که n' بیانگر نمونه‌هایی هستند که از مجموعه آموزشی انتخاب می‌شوند و اندازه آن نسبت به مجموعه آموزشی کوچکتر است.

۲.۲. یادگیری تقویتی

یکی از روش‌های اصلی یادگیری ماشین، یادگیری تقویتی است. این نوع یادگیری برای محیط‌هایی که عامل هیچ نوع شناختی نسبت به آنها ندارد، مناسب است [۵]. در این نوع یادگیری به عامل[1] اجازه داده می‌شود تا از طریق تعامل با محیط و نشان‌دادن عکس‌العمل به آن، پاداش تجمعی را به حداکثر برساند. عناصر یادگیری تقویتی عبارتند از: مجموعه حالت‌ها[2] $S = \{s_i\}$، مجموعه کنش‌ها[3] $A = \{a_i\}$، تابع انتقال حالت[4] $T(s, a, s')$ که حالت بعدی پس از اجرای کنش a را مشخص می‌کند، تابع پاداش[5] $R(s, a)$ که بیانگر میزان پاداش انتخاب کنش a در حالت s است.

در شکل (۱) می‌توان نحوه عملکرد یادگیری تقویتی را مشاهده کرد. ابتدا عامل کنش A_t را بر روی محیط انجام می‌دهد. سپس محیط که در حالت S_t بود، عامل کنش A_t را انتخاب می‌کند و به حالت بعدی S_{t+1} می‌رود و پاداش R_{t+1} را به عامل گزارش می‌دهد [۶].



شکل (۱): معماری یادگیری تقویتی

۳.۲. یادگیری افزایشی

الگوریتم‌های یادگیری افزایشی به منظور طبقه‌بندی نمودن کلان داده‌ها و داده‌های جریانی مطرح شده‌اند [۲]. این نوع الگوریتم‌ها به طور کامل دانش مدل موجود را یاد می‌گیرند. رویه یادگیری افزایشی بدین صورت است که داده‌های ورودی به طور مداوم برای گسترش دانش موجود به مدل اضافه می‌شوند. بنابراین، مرحله آموزش به طور مداوم انجام می‌شود.

هدف از یادگیری افزایشی این است که مدل با داده‌های جدید سازگار شود بدون اینکه دانش موجود خود را فراموش کند. تاکنون الگوریتم‌های یادگیری افزایشی متعددی ارائه شده است. به صورت کلی، الگوریتم‌های یادگیری افزایشی را می‌توان به دو دسته بازآموز و غیربازآموز تقسیم کرد. در الگوریتم‌های غیربازآموز هنگامی که نمونه‌های جدیدی به مدل اضافه می‌شوند، نیازی به بازآموزی با تمام نمونه‌های آموزشی نیست. بنابراین، مصرف حافظه و هزینه زمان آموزش کاهش می‌یابد. اما در الگوریتم‌های بازآموز، تمامی نمونه‌ها نگهداری می‌شوند و هنگامی که نمونه جدیدی به مدل اضافه می‌شود، نیازمند بازآموزی مدل با تمام نمونه‌های آموزشی است. بنابراین، مصرف حافظه و زمان آموزش افزایش می‌یابد.

۳. مرور کارهای گذشته

در دهه‌های گذشته محققان از الگوریتم‌های سنتی یادگیری ماشین مانند شبکه‌های بیزین، درخت تصمیم و جنگل‌های

---

[1] Agent
[2] States
[3] Actions
[4] Transition Function
[5] Reward

تصادفی برای طبقه‌بندی داده‌ها استفاده می‌کردند [۷]- [۱۲]. اما الگوریتم‌های مذکور هنگام مواجه با کلان داده‌ها و داده‌های جریانی ممکن است با شکست روبه‌رو شوند. برای غلبه بر این موضوع، برخی از محققین الگوریتم‌های یادگیری افزایشی [۲] را مطرح کردند. در ادامه به توضیح موارد فوق پرداخته می‌شود.

لوزینگ و همکاران در [۱۳]، اغلب الگوریتم‌های معروف یادگیری افزایشی را بر روی مجموعه داده‌های مختلفی آزمایش کردند. از طرف دیگر، برخی از محققین [۳]، [۱۴]، [۱۵] الگوریتم‌هایی مبتنی بر یادگیری افزایشی با استفاده از الگوریتم گرادیان کاهشی تصادفی را برای به حداقل رساندن تابع ضرر برای یادگیری یک مدل خطی ارائه کرده‌اند. به عنوان مثال، الگوریتم گرادیان کاهشی تصادفی یک روش بهینه‌سازی کارآمد برای یادگیری یک مدل متمایز با به حداقل رساندن تابع ضرر مانند تابع لاجستیک[1] یا تابع هینج[2] است. الگوریتم آنها برای داده‌های پراکنده و با ابعاد بالا که اغلب در حوزه طبقه‌بندی متن یا پردازش زبان طبیعی مورد استفاده قرار می‌گیرند، عملکرد خوبی داشته است.

ژانگ [۱۶]، الگوریتم بیزین ساده با یک توزیع گاوسین موازی محور[3] ارائه کرده است. مدل خلوت[4] امکان یادگیری بسیار کارآمد را از نظر زمان پردازش و نیازهای حافظه فراهم می‌کند. این الگوریتم به طور موثر چندین مثال آموزشی را یاد می‌گیرد [۱۷] و در مواردی مانند فیلتر کردن هرزنامه و طبقه‌بندی اسناد نتایج قابل قبولی داشته است [۱۸]، [۱۹].

فروند و همکاران [۲۰]، یک الگوریتم یادگیری متوالی برخط را برای شبکه‌های فیدفوروارد[5] تک لایه ارائه کردند. الگوریتم آنها به عنوان ماشین یادگیری افراطی متوالی برخط[6] شناخته می‌شود و می‌تواند داده‌ها را تک به تک یا قسمت به قسمت (یک بلاک از داده‌ها) با اندازه قسمت ثابت یا متغیر یاد بگیرد که به طور قابل توجهی زمان پردازش کلی را کاهش می‌دهد.

صفری و همکاران [۲۱]، یک نسخه افزایشی از الگوریتم جنگل تصادفی تحت عنوان جنگل تصادفی برخط[7] را ارائه کرده‌اند. رویکرد آنها ترکیب کیسه‌بندی[8] برخط و جنگل‌های تصادفی است. این رویکرد با حذف کل درختان (بر اساس خطاهای تخمینی خارج از کیسه آنها) در فواصل زمانی ثابت و رشد متوالی درختان جدید، وزن‌دهی زمانی دانش را امکان پذیر می‌کند. از این‌رو، این رویکرد می‌تواند در برنامه‌هایی با نویز زمانی یا تغییرات ظاهری بالا در طول ردیابی بصری مفید باشد.

لوزینگ و همکاران [۲۲]، یک الگوریتم کوانتیزه‌سازی بردار یادگیری افزایشی[9] که اقتباسی از کوانتیزه‌سازی بردار یادگیری تعمیم‌یافته استاتیک[10] [۲۳] است، ارائه کرده‌اند. در رویکرد آنها تابع ضرر به گونه دیگری تعریف می‌شود که در صورت لزوم نمونه‌های اولیه جدیدی را به مدل اضافه کند. در ضمن نرخ درج براساس تعداد نمونه‌های طبقه‌بندی اشتباه صورت می‌گیرد.

پلیکار و همکاران [۲]، الگوریتم Learn++ را مطرح نموده‌اند. در الگوریتم آنها نمونه‌های دریافتی در تکه‌هایی[11] با اندازه از پیش تعریف شده پردازش می‌شود. برای هر قسمت، گروهی[12] از طبقه‌بندی‌کننده‌های پایه آموزش داده می‌شوند و از طریق رای اکثریت وزنی به «گروهی از گروه‌ها» ترکیب می‌شوند. مشابه الگوریتم AdaBoost [۲۰]، هر طبقه‌بندی‌کننده با زیرمجموعه‌ای از نمونه‌های قسمتی که بر اساس توزیع داده‌ها است، آموزش داده می‌شود.

سید و همکاران [۲۴]، رویکرد ماشین بردار پشتیبان افزایشی را به منظور طبقه‌بندی داده‌ها ارائه کرده‌اند. رویکرد آنها بدین صورت است که مجموعه داده‌ها به چندین دسته تقسیم می‌شوند و دسته‌ها به ترتیب به عنوان ورودی دسته‌بندها در نظر گرفته می‌شوند. از این‌رو، طبقه‌بند برای هر دسته، بردارهای پشتیبان را کسب می‌کند و تمامی داده‌های قبلی به جز بردارهای پشتیبان حذف می‌شوند، ولی هرگاه نمونه جدیدی به مجموعه

---

1 Logistic Loss
2 Hinge Loss
3 Naive Bayesian Gauss
4 Sparse
5 Feed Forward
6 Online Sequential Extreme Learning Machine
7 Online Random Forest
8 Bagging
9 Incremental Learning Vector Quantization
10 Generalized Learning Vector Quantization
11 Chunks
12 Ensemble

داده قبلی اضافه می‌شود، مدل از ابتدا اجرا می‌شود. با این حال، این رویکرد دارای معایبی است که عبارتند از: (۱) نیاز به حافظه بزرگ، (۲) زمان اجرای طولانی و (۳) فرض یکسان بودن توزیع داده‌ها.

همان‌طور که پیش از این بیان کردیم، یکی از معایب یادگیری افزایشی مشکل فراموشی فاجعه‌بار است. از این‌رو، برخی از محققین الگوریتم‌هایی را برای مقابله با این مساله ارائه کردند، اما اکثر الگوریتم‌های ارائه شده دارای بار محاسباتی زیادی بوده و به حافظه بزرگی نیاز دارند. بنابراین، ژاو و همکاران [۲۵]، یک الگوریتم یادگیری افزایشی مبتنی بر بافر حافظه محدود را مطرح کردند. در این الگوریتم برای حفظ دانش قدیمی تعدادی نمونه در حافظه نگهداری می‌شود. شایان ذکر است، اگر تعداد نمونه‌های ذخیره شده در حافظه کم باشد، سبب کاهش دقت مدل می‌شود. از سوی دیگر، برای افزایش دقت مدل نیاز به ذخیره نمونه‌های زیادی است که این امر منجر به افزایش هزینه می‌شود. بنابراین، در این الگوریتم از یک روش نمونه‌کاهی[1] به منظور افزایش تعداد نمونه‌ها در بافر حافظه و کاهش هزینه استفاده می‌شود و هنگام اضافه شدن نمونه جدید، مدل با استفاده از این نوع نمونه و نمونه‌های ذخیره شده قبلی بازآموزی می‌شود. اگرچه الگوریتم مذکور از دقت خوبی برخوردار است، اما زمانبر است.

وکیلی و همکاران [۲۶]، برخی از دسته‌بندها را همانند دسته‌بند بیزین ساده، دسته بند k-نزدیکترین همسایه و دسته‌بند جنگل تصادفی، همچنین برخی از الگوریتم‌ها همانند الگوریتم‌های ارائه شده در مراجع [۲] و [۲۰] را به نسخه افزایشی تبدیل کرده‌اند. آنها مجموعه داده تشخیص فعالیت‌های انسانی را برای بررسی صحت الگوریتم‌های خود استفاده نموده‌اند.

داده‌های ترافیک شبکه گونه‌ای از داده‌های جریانی هستند که به طور مداوم و در طول زمان می‌رسند. لی و همکاران [۲۷]، الگوریتمی مبتنی بر یادگیری افزایشی با استفاده از الگوریتم گرادیان کاهشی تصادفی برای شناسایی ترافیک‌های بدافزار رمزگذاری شده در شبکه‌ها را طراحی نموده‌اند.

دوو و همکاران [۲۸]، برای طبقه‌بندی مجموعه کلان داده‌های تصاویر، الگوریتمی مبتنی بر یادگیری افزایشی به همراه الگوریتم گرادیان کاهشی تصادفی محلی طراحی کرده‌اند. گرادیان کاهشی تصادفی محلی بلوک داده را با استفاده از الگوریتم میانگین k[2] به k قسمت تقسیم می‌کند و سپس به صورت موازی مدل گرادیان کاهشی تصادفی را در هر قسمت به صورت محلی می‌آموزد.

آگروال و همکاران [۲۹]، برای تشخیص اثر انگشت جعلی و واقعی یک الگوریتم یادگیری افزایشی با استفاده از دسته‌بند ماشین بردار پشتیبان و الگوریتم‌های خوشه‌بندی ارائه کرده‌اند. در الگوریتم آنها از شباهت ذاتی موجود در داده‌ها استفاده می‌شود. بنابراین، ابتدا داده‌ها خوشه‌بندی شده و سپس خوشه‌ها با استفاده از ماشین بردار پشتیبان طبقه‌بندی می‌شوند. مزیت این الگوریتم بدین صورت است که هنگام اضافه شدن نمونه جدید به مدل، نیازی به بازآموزی مجدد مدل نیست. بنابراین، برای تشخیص اثرانگشت جعلی نیازی به دسترسی به اثرانگشت‌های آموزش دیده قبلی در حین یادگیری اثرانگشت‌های جدید نیست، که این امر منجر به نیاز کمتر به حافظه می‌شود. با این حال، اگر از مجموعه داده خیلی بزرگ استفاده شود، سبب افزایش بار محاسباتی می‌گردد.

برخی از محققان برای طبقه‌بندی داده‌ها از شبکه‌های عصبی به همراه الگوریتم‌های یادگیری افزایشی استفاده کرده‌اند که در ادامه به توضیح آنها پرداخته می‌شود.

هوو و همکاران [۳۰]، الگوریتم یادگیری افزایشی مبتنی بر شبکه عصبی خود رمزگذار متغیر[3] را برای طبقه‌بندی داده‌ها مطرح کردند. در واقع آنها برای استخراج ویژگی‌ها از شبکه عصبی خود رمزگذار متغیر استفاده کرده و سپس از یک شبکه از پیش آموزش دیده برای طبقه‌بندی داده‌ها استفاده کردند. همچنین خروجی‌های خود رمزگذار متغیر به عنوان داده کمکی برای داده‌های جدید استفاده شده است. بنابراین، هنگام اضافه شدن داده جدید مدل از ابتدا بازآموزی نمی‌شود. در نتیجه هزینه حافظه و محاسبات کاهش می‌یابد. با این حال، اگر از مجموعه

[1] Down Sampling

[2] k-means

[3] Variational Autoencoder

داده خیلی بزرگ استفاده شود زمان آموزش مدل افزایش می‌یابد و سبب افزایش بار محاسباتی می‌شود.

شان و همکاران [۳۱]، برای طبقه‌بندی داده‌های متنی الگوریتمی مبتنی بر شبکه عصبی حافظه کوتاه مدت طولانی[1] را مطرح کردند. الگوریتم آنها شامل ۴ مولفه دانش‌آموز، یادگیری تقویتی، معلم و متمایزکننده است. در مولفه دانش‌آموز ابتدا پیش‌بینی‌های اولیه توسط شبکه عصبی مذکور صورت می‌گیرد. سپس مولفه یادگیری تقویتی نتایج چندین مولفه دانش‌آموز را فیلتر می‌کند. پس از آن، مولفه معلم نتایج فیلتر شده را مجدد طبقه‌بندی می‌کند و طبقه‌بندی متون نهایی را به دست می‌آورد و در نهایت برای جلوگیری از افزایش نامحدود مولفه‌های دانش‌آموز با افزایش تعداد نمونه‌ها، از مولفه متمایزکننده برای فیلتر کردن مولفه‌های دانش‌آموز بر اساس شباهت آنها استفاده می‌شود. آزمایش‌ها بر روی مجموعه داده‌های مختلف نشان می‌دهد که الگوریتم مذکور از عملکرد خوبی برخوردار است.

لموس و همکاران [۳۲]، یک الگوریتم یادگیری افزایشی مبتنی بر شبکه عصبی حافظه کوتاه مدت طولانی برای طبقه‌بندی داده‌های جریانی پیشنهاد کردند. ایده اصلی الگوریتم به این صورت است که از ویژگی‌های ذاتی شبکه عصبی مذکور برای کنترل کردن مساله فراموشی فاجعه‌بار، تغییر توزیع داده‌ها و مدیریت حافظه استفاده شود. نتایج تجربی نشان می‌دهد که این الگوریتم به تغییر توزیع داده‌ها واکنش نشان می‌دهد و همچنین در برابر داده‌های نویزی مقاوم است.

بوشانکومار و همکاران [۳۳]، یک الگوریتم یادگیری افزایشی تطبیقی جدید برای طبقه‌بندی کیفیت آب هوشمند مبتنی بر اینترنت اشیا را مطرح کردند. در الگوریتم آنها برای رسیدگی به مشکل نامتوزان بودن داده‌ها از الگوریتم ژنتیک و SMOTE، استخراج ویژگی‌ها و طبقه‌بندی کیفیت آب از شبکه عصبی استفاده شده است. از این‌رو، ابتدا داده‌های کیفیت آب با استفاده از حسگرهای اینترنت اشیا جمع‌آوری می‌شود. سپس، ویژگی‌ها استخراج می‌شوند و ویژگی‌های زائد و نامربوط حذف می‌شوند. در ادامه، برای متوازن‌سازی داده‌ها دو الگوریتم ژنتیک و SMOTE با یکدیگر ترکیب می‌شوند. در گام بعد، از یک شبکه عصبی عمیق برای طبقه‌بندی استفاده می‌شود. ذکر این نکته الزامی است که هنگام اضافه شدن نمونه جدید مدل از ابتدا اجرا نمی‌شود، بلکه از دانش قبلی کسب شده برای طبقه‌بندی کیفیت آب استفاده می‌شود. اگرچه این الگوریتم از عملکرد خوبی برخوردار است، اما به دلیل استفاده از شبکه عصبی عمیق و تعداد زیاد لایه‌های پنهان زمانبر است.

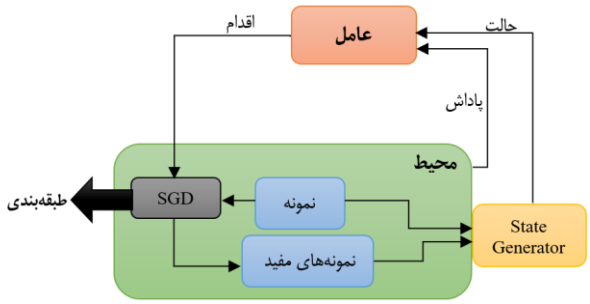
کانگ و همکاران [۳۴]، یک الگوریتم یادگیری افزایشی مبتنی بر شبکه عصبی عمیق را برای طبقه‌بندی داده‌ها و اضافه شدن کلاس جدید به مجموعه کلاس‌های قدیمی داده‌ها پیشنهاد کردند. در الگوریتم آنها به طور مداوم دانش جدید با حافظه محدود یاد گرفته می‌شوند در حالی که دانش قدیمی حفظ می‌شود. در الگوریتم مذکور احتمال اهمیت هر ویژگی تخمین زده می‌شود، سپس ویژگی‌هایی که بیشترین احتمال اهمیت را داشته باشند، نگهداری شده و برای طبقه‌بندی استفاده می‌شوند. از سوی دیگر، هنگام اضافه شدن نمونه جدید ابتدا ویژگی‌های آن استخراج شده و در ادامه، ویژگی‌های استخراج شده و ویژگی‌هایی که در مراحل قبلی نگهداری شده بودند با یکدیگر به منظور بررسی این که آیا نمونه جدید متعلق به کلاس جدید است یا خیر، مقایسه می‌شوند. بنابراین، هنگام اضافه شدن نمونه جدید الگوریتم از ابتدا اجرا نمی‌شود. اگرچه این الگوریتم از عملکرد خوبی برخوردار است، اما به دلیل استفاده از شبکه عصبی عمیق زمانبر است.

۴. الگوریتم پیشنهادی

همان‌طور که پیش از این بیان کردیم، الگوریتم‌های ارائه شده از یادگیری افزایشی به همراه دسته‌بندهای مختلف برای طبقه‌بندی داده‌ها استفاده کرده‌اند، اما اکثر الگوریتم‌ها هنگام مواجه با داده‌های جریانی با شکست مواجه شده‌اند یا عملکرد ضعیفی از خودشان نشان می‌دهند. از سوی دیگر، یکی از معیارهای اصلی در یادگیری افزایشی زمان اجرا است ولی در راهکارهایی که به آنها اشاره شد، به این معیار توجهی نشده است. بنابراین، در این پژوهش قصد بر این است که برای طبقه‌بندی داده‌های مورد

[1] Long-Short Term Memory (LSTM)

بررسی با صحت و زمان اجرای مناسب از یادگیری تقویتی به همراه دسته‌بند گرادیان کاهشی تصادفی برای یادگیری افزایشی استفاده شود. شایان ذکر است، الگوریتم پیشنهادی محدود به استفاده از دسته‌بند گرادیان کاهشی تصادفی نیست و امکان استفاده از دسته‌بندهای دیگر به شرط آنکه (۱) تعداد پارامترهای آن دسته‌بند کم باشد، (۲) بتوان دسته‌بند را در قالب یک بردار نمایش داد و (۳) برای آن تعدادی کنش تعریف کرد، وجود دارد. معماری الگوریتم پیشنهادی Inc-RL-SGD در شکل (۲) نشان داده شده است.



شکل (۲): معماری الگوریتم پیشنهادی INC-RL-SGD

همان‌طور که بیان شد، گرادیان کاهشی تصادفی یک الگوریتم بهینه‌سازی است. برخی از محققان یک الگوریتم یادگیری افزایشی مبتنی بر گرادیان کاهشی تصادفی و دسته‌بندها را مطرح کردند [۳]، [۱۴]. آنها در الگوریتم خود ثابت کردند که اگر گرادیان کاهشی تصادفی با تابع زیان هینج برای یادگیری یک مدل خطی استفاده شود، می‌توان به دسته‌بندی معادل دسته‌بند ماشین بردار پشتیبان دست یافت [۱۳]. از طرفی، هدف از یادگیری تقویتی به حداکثر رساندن پاداش تجمعی است [۵] و هدف در این پژوهش دستیابی به صحت و زمان اجرای مناسب است. نحوه یادگیری تقویتی برای دستیابی به پاداش مورد نظر بدین صورت است که ابتدا مقداردهی اولیه مدل صورت می‌گیرد و سپس وارد مرحله آموزش می‌شود. در این مرحله تعداد محدودی از نمونه داده‌ها با کمک دسته‌بند گرادیان کاهشی تصادفی مدل آموزش داده می‌شود. بدین ترتیب، نمونه‌ها به عنوان ورودی دسته‌بند مذکور در نظر گرفته می‌شوند و گرادیان کاهشی تصادفی آنها را طبقه‌بندی می‌کند.

از سوی دیگر، یادگیری تقویتی شامل مجموعه حالت‌ها است [۵]. بردار حالت در یادگیری تقویتی نمایشی از محیط فعلی است که عامل در آن قرار دارد و این حالت توسط عامل قابل مشاهده است و شامل تمام اطلاعات مربوط به محیط است که عامل برای تصمیم‌گیری باید بداند. در این پژوهش بردار حالت شامل دو مولفه نمونه دریافتی x_t و مدل M_{t-1} است. حال می‌بایست راهکاری ارائه شود که مدل در قالب یک بردار بیان شود. در مدل‌های مختلف این کار متغیر است. به عنوان مثال در دسته‌بند ماشین بردار پشتیبان، مدل یک ابرصفحه است. در دسته‌بند گرادیان کاهشی تصادفی یک رابطه خط همانند ابرصفحه SVM وجود دارد که این رابطه خط توسط الگوریتم گرادیان کاهشی تصادفی محاسبه می‌شود. بنابراین، می‌توان پارامترهای رابطه خط به دست آمده را به عنوان بازنمایی مدل در نظر گرفت و به بردار حالت اضافه نمود. از سوی دیگر، نمونه دریافتی x_t در قالب یک بردار دریافت می‌شود. بنابراین، دو مولفه مذکور را می‌توان در کنار یکدیگر قرار داد و در قالب یک بردار حالت در نظر گرفت. هنگامی که حالت جدید دریافت می‌شود از بین کنش‌های موجود، عامل باید بهترین کنش را انتخاب کند. در حالت کلی، برای مساله مورد نظر سه کنش تعریف شده است که عبارتند از:

- TA: پذیرفتن نمونه جدید و بازآموزی مجدد مدل.
- TUS: پذیرفتن نمونه جدید و بازآموزی مدل به وسیله نمونه‌های مفید؛ در هر گام نمونه‌هایی که حاوی خطای کمتری هستند و به رابطه خط نزدیک‌تر هستند، ذخیره می‌شوند. ذکر این نکته الزامی است که با توجه به نوع داده‌ها تعداد نمونه‌های مفید متغیر است.
- NT: نپذیرفتن نمونه جدید با فرض اینکه نمونه حاوی اطلاعات جدیدتری نیست یا حاوی اطلاعات کم ارزشی است و با در نظر گرفتن سربار ناشی از بازآموزی به صرفه نیست که با این نمونه بازآموزی انجام شود.

از سوی دیگر، اگر هدف دستیابی به بیشترین صحت ممکن باشد، بهترین راهکار این است که تمامی نمونه‌های جدید را بپذیریم و مدل را مجدد بازآموزی کنیم؛ اما این روش بسیار

زمانبر است. بنابراین، کنش جدیدی را می‌توان تعریف کرد تا به هدف اصلی پژوهش دست یافت. از این‌رو، به جای بازآموزی مدل با تمامی نمونه داده‌ها، صرفا مدل با استفاده از نمونه‌های مفیدی که جمع‌آوری شده است، بازآموزی می‌شود. مزیت این روش این است که تعداد این نوع نمونه‌ها نسبت به تمام نمونه‌ها کمتر است. بنابراین، زمان بازآموزی کاهش می‌یابد.

حال عامل با توجه به موقعیت دریافتی از بین سه کنش موجود بهترین کنش ممکن را انتخاب می‌کند. پس از دریافت حالت و کنش، پاداش محاسبه می‌شود. اگر صرفا هدف دستیابی به بیشترین صحت طبقه‌بندی باشد تنها معیار صحت کافی است و می‌توان با پذیرش نمونه جدید مدل را مجدد بازآموزی نمود تا به بهترین صحت ممکن دست یافت. ولی در عمل این روش بسیار زمانبر است. بنابراین، می‌بایست توازنی بین صحت و زمان در نظر گرفته شود تا هر بار با اضافه شدن نمونه جدید، مدل از ابتدا بازآموزی نشود. به همین دلیل، تابع پاداش به صورت رابطه (۳) تعریف می‌شود.

$$Reward^t = \frac{Acc^t}{Training\ Time^t} \qquad (۳)$$

که Acc^t بیانگر صحت یادگیرنده در واحد درصد در دوره t و $Training\ Time^t$ بیانگر زمان یادگیری در دوره t و در واحد ثانیه است. همان‌طور که مشاهده می‌شود، اگر در تابع پاداش دو معیار صحت و زمان یادگیری در نظر گرفته شود، طبقه‌بندی داده‌ها با صحت و سرعت بالا انجام می‌شود.

شبه کد الگوریتم پیشنهادی Inc-RL-SGD در الگوریتم‌های (۱) و (۲)، در دو مرحله آموزش و شبیه‌سازی محیط ارائه شده است که در ادامه به توضیح آنها پرداخته می‌شود.

همان‌طور که در الگوریتم (۱) مشاهده می‌شود، ورودی مدل، مجموعه داده D است. در گام‌های بعدی تعداد دورهای یادگیری و مقداردهی اولیه تعیین می‌شود. سپس مقداردهی اولیه دسته‌بند به صورت تصادفی انجام می‌شود. در گام بعد دسته‌بند توسط مجموعه آموزشی، آموزش داده شده و سپس مقداردهی اولیه محیط صورت می‌گیرد. در گام بعد، نمونه جدید دریافت می‌شود. پس از دریافت نمونه، بردار حالت که شامل نمونه و مدل است، تشکیل شده و به عامل فرستاده می‌شود. سپس عامل با استفاده از سیاست ε–حریصانه و از بین مجموعه کنش‌های تعریف‌شده، کنشی را انتخاب می‌کند. شایان ذکر است که در هر دور از مرحله یادگیری نمونه‌های مفید تولیدشده ذخیره می‌شوند. در گام بعد بردار حالت و کنش انتخاب شده به محیط ارسال می‌شوند.

الگوریتم (۱): مرحله آموزش الگوریتم INC-RL-SGD

INPUT: Training data $D = \{(x_1, y_1), \dots, (x_t, y_t)\}$
  Episode number K
  Action set A = {TA, TUS, NT}
  Initialize experience replay memory RM
  Randomly initialize neural network parameters θ (i.e., weights)
  Train model M_t (i.e., SGD classifier) with $D = \{(x_1, y_1), \dots, (x_{t'}, y_{t'})\}$ where $t' < t$
  Initialize simulation environment
  for episode k = 1 to K
    Receive sample x_t
    Shuffle the training data D
    Initialize state $s_t = x_t \cap M_{t-1}$
    for i = t+1 to T
      Choose an action based ε-greedy policy
      if ε < random number
        a_t = choose a random action from A
      else
        $a_t = \max_a Q(s_t, a, \theta)$
    $r_t, M_t = STEP(a_t, (x_t, y_t), M_{t-1})$
    Set $s_{t+1} = x_{t+1} \cap M_t$
    Store (s_t, a_t, r_t, s_{t+1}) to RM
    Randomly sample (s_i, a_i, r_i, s_{i+1}) from RM
    Set $O_i = (r_i + \gamma \max_{a'} Q(s_i + 1, a', \theta))$
    Calculate loss function $L(\theta)$

الگوریتم (۲): مرحله شبیه‌سازی محیط الگوریتم INC-RL-SGD

FUNCTION: STEP($a_t \in$ A, $(x_t, y_t) \in D, M_{t-1}$)
  if $a_t = TA$
    $M_{t+1} \leftarrow$Train SGD by all samples $\in D_t$;
    $US_{t+1} = US_t \cup$ New useful samples $\in M_t$;
  if $a_t = TUS$
    $M_{t+1} \leftarrow$Train SGD by US_t;
    $US_{t+1} = US_t \cup$ New useful samples $\in M_t$;
  if $a_t = NT$
    $M_{t+1} \leftarrow M_t$;
  Compute accuracy and running time of M_t on the test data;
  $r_t = \frac{Accuracy\ (M_t)}{Training\ Time\ (M_t)}$;
  return r_t, M_t;

همان‌طور که در الگوریتم (۲) مشاهده می‌شود، آرگومان‌های تابع STEP نمونه، مدل و کنش انتخاب شده از مجموعه کنش‌های

A است. اگر عامل کنش TA را انتخاب کرد، آنگاه دسته‌بند با تمام نمونه‌ها بازآموزی می‌شود و همچنین نمونه مفید جدید به دست آمده در این مرحله به مجموعه نمونه‌های مفید قبلی اضافه می‌شود. در غیر اینصورت اگر عامل کنش TUS را انتخاب کرد، آنگاه دسته‌بند با تمام نمونه‌های مفید به دست آمده بازآموزی شده و همچنین نمونه مفید جدید به دست آمده در این مرحله به مجموعه نمونه‌های مفید قبلی اضافه می‌شود. درغیر اینصورت اگر عامل کنش NT را انتخاب کرد، آنگاه نمونه پذیرش نمی‌شود و مدل تغییری نمی‌کند. پس از اینکه کنش مورد نظر انتخاب شد نوبت به محاسبه پاداش از سوی محیط است. از سوی دیگر، یادگیری تقویتی با توجه به بردار حالت و میزان پاداش تصمیم می‌گیرد که نمونه دریافتی را بپذیرد یا حذف کند. در نهایت دو مقدار پاداش و مدل از سمت محیط به عامل برگردانده شده و به منظور سنجش مدل از نظر قابلیت و توانایی در پیش‌بینی مقدارهای جدید از تابع زیان استفاده می‌شود.

۵. ارزیابی

برای پیاده‌سازی مدل پیشنهادی و پیش‌پردازش داده‌های ورودی از زبان برنامه‌نویسی پایتون و کتابخانه Pytorch استفاده شده است. آزمایش‌ها بر روی پردازنده 1.80 GHz Intel Core i5 و رم 8.0 GB انجام شده است. شایان ذکر است که الگوریتم پیشنهادی با چندین الگوریتم افزایشی مقایسه شده است. بنابراین، در هر بار مقایسه با الگوریتم مورد بررسی متغیرهایی از قبیل نرخ یادگیری، وزن‌های شبکه، تابع زیان و تعداد دورهای اجرا همانند الگوریتم مورد بررسی در نظر گرفته شده است تا شرایطی عادلانه برای مقایسه وجود داشته باشد.

در این پژوهش الگوریتم پیشنهادی Inc-RL-SGD با الگوریتم‌های ژانگ [۳]، ژانگ [۱۶]، فروند [۲۰]، صفری [۲۱]، لوزینگ [۲۲]، پلیکار [۲]، سید [۲۴] و وکیلی [۲۶] با استفاده از تعدادی مجموعه داده‌های بانک مخزن UCI و مجموعه داده تشخیص فعالیت‌های انسانی مقایسه شده است. از این‌رو، مدل پیشنهادی با شرایطی مشابه الگوریتم‌های مورد مقایسه [۲]، [۳]، [۱۳]، [۱۶]، [۲۰]-[۲۲]، [۲۴]، [۲۶] و [۳۵] پیاده‌سازی شده است تا مقایسه در شرایط عادلانه صورت گیرد. بنابراین، در الگوریتم‌های مورد مقایسه نسبت نمونه‌های اموزشی به آزمون ۸۰ به ۲۰ در نظر گرفته شده است و در این پژوهش نسبت مذکور لحاظ شده است. از سوی دیگر، برخی از داده‌های مورد استفاده داده‌های کثیف[1] هستند. بنابراین، ابتدا داده‌ها با استفاده از روش‌های مرسوم پاکسازی که سایر الگوریتم‌ها از آنها استفاده کرده‌اند، پاکسازی شده و سپس از آنها در الگوریتم پیشنهادی استفاده می‌شود.

در ابتدا به علت عدم شناخت محیط و نبودن نمونه‌های مفید عامل به صورت تصادفی یکی از کنش‌های (۱) پذیرفتن نمونه جدید و بازآموزی مدل توسط گرادیان کاهشی تصادفی یا (۲) نپذیرفتن نمونه جدید را انتخاب می‌کند. سپس با طی کردن ۱۰۰ دور از آموزش و بازآموزی مدل توسط دسته‌بند گرادیان کاهشی تصادفی در جهت کسب نمونه‌های مفید، عامل با محیط آشنا می‌شود و دانش مورد نظر را کسب می‌کند. بنابرین، عامل کنشی متناسب با شرایط کنونی انتخاب می‌کند. در بخش نتایج نشان داده می‌شود که مدل پیشنهادی نسبت به سایر رویکردهای افزایشی مورد مقایسه، از عملکرد بهتری برخوردار است.

۶. نتایج

در این بخش ابتدا به معرفی داده‌های مورد استفاده پرداخته می‌شود. سپس، نتایج حاصل از الگوریتم پیشنهادی بر روی داده‌های مورد بررسی بیان شده و در ادامه معیارهای ارزیابی دسته‌بند معرفی و در نهایت نتایج حاصل از اعمال الگوریتم پیشنهادی بر روی مجموعه داده‌های مورد بررسی به تفصیل بیان می‌شود.

۱.۶. مجموعه داده‌ها

برای بررسی قدرت و صحت الگوریتم پیشنهادی از چندین مجموعه داده متعلق به بانک مخزن UCI و مجموعه داده‌های مربوط به تشخیص فعالیت‌های انسانی استفاده شده است. طبق مرجع [۲۶]، داده‌های تشخیص فعالیت‌های انسانی توسط

[1] Dirty Data

حسگرهای گوشی‌های هوشمند و از طریق جریان‌های بلادرنگ اینترنت اشیا تولید شده است. در این مجموعه داده حداکثر ۲۰ فعالیت وجود دارد. در این پژوهش منظور از فعالیت همان برچسب[1] داده‌ها است. در جدول (۱) مشخصات کلی داده‌های بانک مخزن UCI و در جدول (۲) کلاس‌هایی که در داده‌های مربوط به تشخیص فعالیت‌های انسانی وجود دارد، نشان داده شده است.

جدول (۱): توصیف ویژگی‌های مجموعه داده‌های بانک مخزن UCI

Datasets	#Features	#Classes	#Samples
Australians	14	2	270
Heart-C	13	2	303
Ionosphere	34	2	354
Votes	16	2	445
Gisette	5000	2	7000
Letter	16	26	20000
Mnist	784	10	70000
Susy	18	2	5000000

جدول (۲): توصیف کلاس‌های مجموعه داده تشخیص فعالیت‌های انسانی

NO	Activity	NO	Activity
1	Walking	11	Squats
2	Running	12	Mountain Climber Twist
3	Standing Still	13	Arm Swings
4	Sitting on a chair	14	Forearm Rotation
5	Side Leg Lifts	15	Dumbbell Biceps Curl
6	Boxer Shuffle	16	Jumping Jack
7	Knee Lifts	17	Chest Expansion
8	Cycling using Exercise Bicycle	18	Cross Toe Touch
9	Forward Lunge	19	Straight Punch
10	Torso Rotation	20	Big Arm Circles

۲.۶. نتایج الگوریتم پیشنهادی بر روی داده‌ها

برای مقایسه الگوریتم‌های افزایشی چندین معیار وجود دارد که در این پژوهش به علت نامتوازن بودن داده‌ها از معیارهای صحت، F1-score و زمان اجرا استفاده شده است که در ادامه به توضیح آنها پرداخته می‌شود.

در ابتدا با استفاده از مجموعه داده‌های بانک مخزن UCI الگوریتم پیشنهادی Inc-RL-SGD از نظر صحت با چندین الگوریتم افزایشی که عبارتند از: ژانگ [۳]، ژانگ [۱۶]، فروند [۲۰]، صفری [۲۱]، لوزینگ [۲۲]، پلیکار [۲] و سید [۲۴] مقایسه شده و نشان داده می‌شود که الگوریتم پیشنهادی دارای صحت بهتری نسبت به سایر الگوریتم‌ها است. شایان ذکر است که الگوریتم پیشنهادی در شرایطی مشابه با [۲]، [۳]، [۱۳]، [۱۶]، [۲۰]–[۲۲]، [۲۴] و [۳۵] پیاده‌سازی شده است و نتایج مربوط به الگوریتم‌های افزایشی از مراجع [۱۳] و [۳۵] اخذ شده است تا شرایطی عادلانه برای مقایسه وجود داشته باشد.

جدول‌های (۳) و (۵) نتایج حاصل از الگوریتم پیشنهادی و الگوریتم‌های مورد مقایسه را نشان می‌دهد. همان گونه که مشاهده می‌شود، الگوریتم پیشنهادی در همه مجموعه داده‌ها از صحت بهتری نسبت به سایر الگوریتم‌های افزایشی برخوردار است.

جدول (۳): مقایسه صحت الگوریتم پیشنهادی با سایر الگوریتم‌های افزایشی (برحسب درصد)

الگوریتم ها	Datasets			
	Letter	Susy	Mnist	Gisette
ORF	۹۳/۲	۷۹/۵	۹۴/۳	۹۴/۶
ILVQ	۹۳/۹	۷۹/۰	۹۴/۸	۹۳/۰
LPP_{CART}	۸۷/۰	–	۹۲/۴	۹۴/۲
IELM	۷۰/۰	–	۸۹/۱	۹۱/۴
SGD_{Lin}	۵۶/۴	۷۸/۷	۸۶/۰	۹۳/۱
NB_{Gauas}	۶۳/۴	۷۳/۵	۵۵/۶	۷۵/۴
Inc-RL-SGD	۹۴/۶	۸۰/۱	۹۵/۱	۹۶/۰

شایان ذکر است، در الگوریتم‌های [۲]، [۳]، [۱۳]، [۱۶]، [۲۰]–[۲۲]، [۲۴] و [۳۵] هنگام دریافت نمونه جدید، تمامی مقادیر قبلی دسته‌بند براساس آن به‌روزرسانی و بازآموزی می‌شود، اما در الگوریتم Inc-RL-SGD با استفاده از یادگیری تقویتی میزان مفید بودن نمونه بررسی شده و سپس یادگیری تقویتی با توجه

[1] Label

به داده‌های دریافتی تصمیم می‌گیرد که مدل را بپذیرد یا آن را نادیده بگیرد. بنابراین، الگوریتم Inc-RL-SGD در همه مجموعه داده‌ها از لحاظ صحت و زمان اجرا عملکرد بهتری نسبت به الگوریتم‌های [۲۴] و [۳۵] دارد.

جدول (۴): زمان اجرای الگوریتم پیشنهادی (برحسب ثانیه)

Datasets	زمان اجرا
Letter	۱۵۶/۳۶
Susy	۶۸۶/۳۴
Mnist	۷۹۶/۲۰
Gisette	۹۶/۵۳

جدول (۵): مقایسه صحت الگوریتم پیشنهادی با الگوریتم افزایشی INC-SVM (برحسب درصد)

Datasets	الگوریتم‌ها	
	Inc-SVM [۳۵]	Inc-RL-SGD
Australians	۸۵/۵۱	۸۸/۸۵
Ionosphere	۸۶/۸۸	۸۸/۲۳
Heart-C	۸۲/۴۸	۸۵/۵۱
Votes	۹۵/۴	۹۷/۶۵

یکی دیگر از معیارهای مهم زمان اجرا است. لازم به ذکر است که تمامی نسخه‌های افزایشی الگوریتم‌ها از نسخه غیرافزایشی آنها زمان اجرای به مراتب بیشتری دارند. این امر هم به آن دلیل است که عموما در یک الگوریتم افزایشی یک مدل ممکن است هزاران بار اجرا شود که منجر به افزایش زمان خواهد شد. با این حال فقط در الگوریتم افزایشی کاهشی ماشین بردار پشتیبان [۲۴]، [۳۵]، معیار زمان اجرا در نظر گرفته شده است و در مابقی الگوریتم‌های افزایشی مذکور به زمان اجرا توجهی نشده است. در جدول‌های (۴) و (۶) زمان اجرای مجموعه داده‌های مذکور نشان داده شده است و همان‌طور که مشاهده می‌شود، الگوریتم پیشنهادی از زمان اجرای خوبی برخوردار است.

لازم به ذکر است که به دلیل ناموازن بودن مجموعه داده‌ها، از معیار F1-score استفاده شده است و نتایج حاصل در جدول (۷) نشان داده شده است. از سوی دیگر، در الگوریتم Inc-SVM [۲۴]، [۳۵]، علاوه بر معیار صحت از معیار F1-score استفاده کرده‌اند. از این رو، در جدول (۸) میزان F1-score الگوریتم پیشنهادی با الگوریتم Inc-SVM مقایسه شده و همان‌طور که مشاهده می‌شود، الگوریتم پیشنهادی به دلیل استفاده از یادگیری تقویتی از عملکرد بهتری برخوردار است.

جدول (۶): مقایسه زمان اجرای الگوریتم پیشنهادی با الگوریتم افزایشی INC-SVM (برحسب ثانیه)

Datasets	الگوریتم‌ها	
	Inc-SVM [۳۵]	Inc-RL-SGD
Australians	۱۰۵/۱۰	۱۲/۲۵
Ionosphere	۸۰/۸۴	۱۱/۳۲
Heart-C	۶۷/۴۷	۱۰/۸۴
Votes	۸۱/۶۵	۹/۷۶

جدول (۷): بررسی F1-SCORE داده‌های مورد بررسی (برحسب درصد)

Datasets	F1-score
Heart-C	۹۱/۱۵
Australians	۹۱/۳۵
Ionosphere	۹۲/۳۹
Votes	۹۸/۰۱
Letter	۹۶/۸۵
Susy	۸۵/۷۳
Mnist	۹۶/۹۹
Gisette	۹۶/۵۹
HAR	۹۸/۵

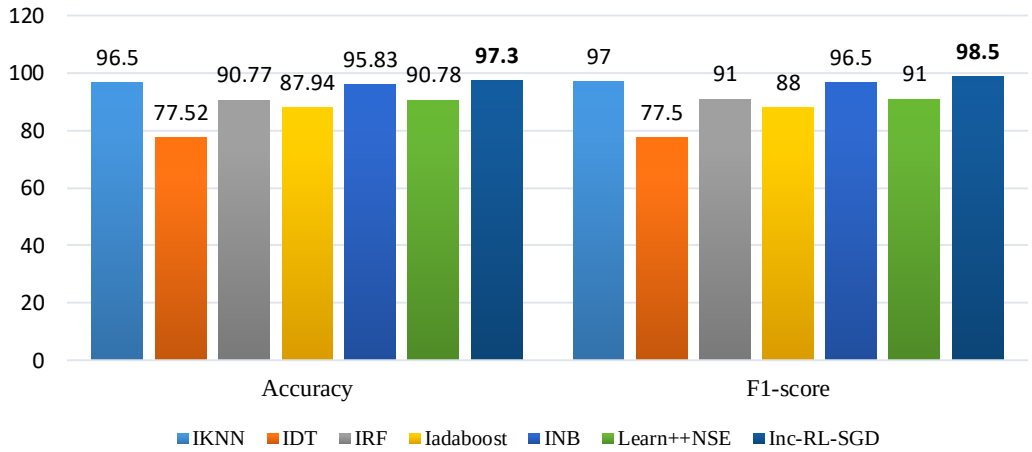
جدول (۸): مقایسه F1-SCORE الگوریتم پیشنهادی با الگوریتم افزایشی INC-SVM (برحسب درصد)

Datasets	الگوریتم‌ها	
	Inc-SVM [۳۵]	Inc-RL-SGD
Australians	۹۰/۸۰	۹۱/۳۵
Ionosphere	۹۰/۷۵	۹۲/۳۹
Heart-C	۸۹/۹۰	۹۱/۱۵
Votes	۹۶/۸۵	۹۸/۰۱

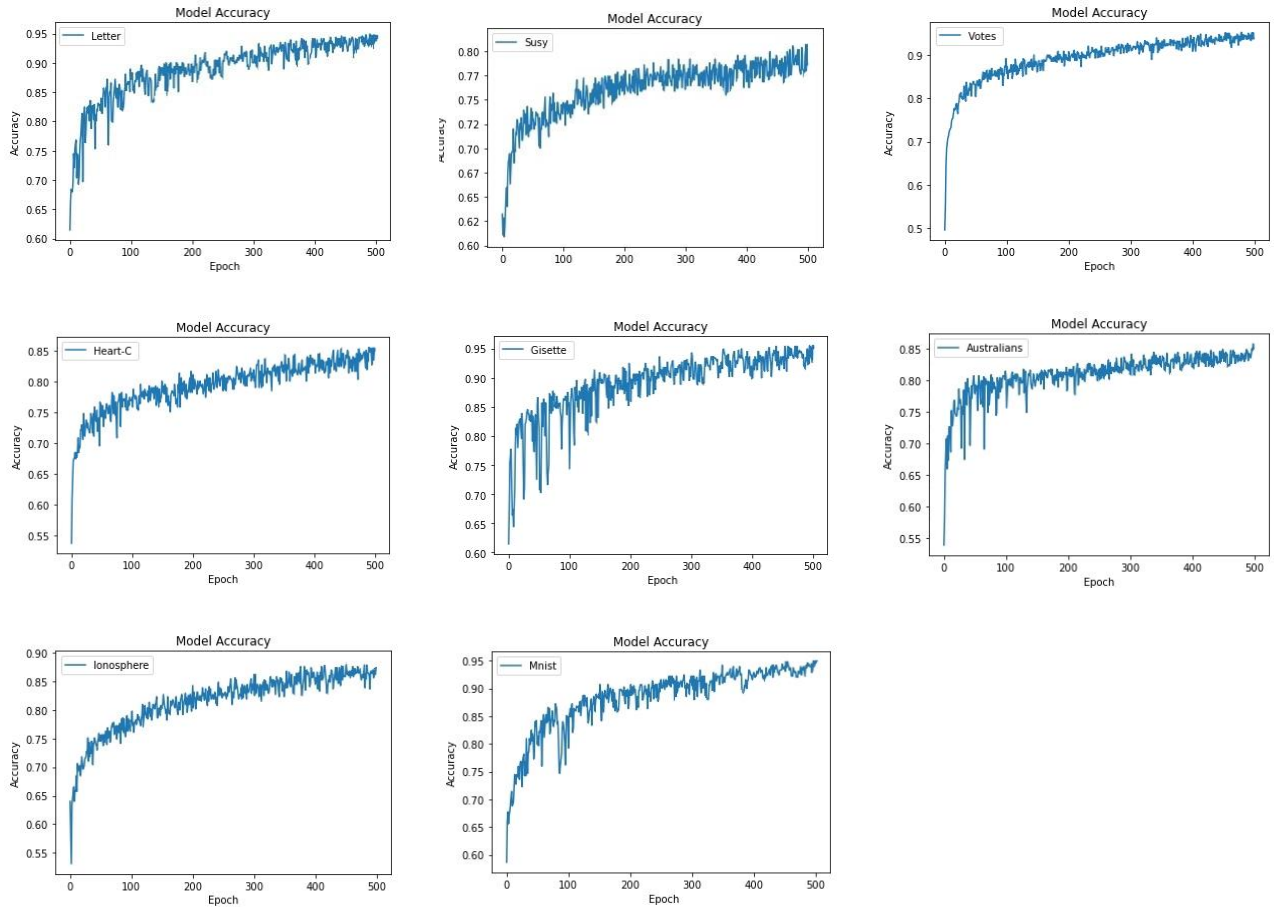
در کاربردی دیگر به منظور بررسی صحت و دقت الگوریتم پیشنهادی از مجموعه داده‌های مربوط به تشخیص فعالیت‌های انسانی استفاده شده است. شایان ذکر است، الگوریتم پیشنهادی در شرایطی مشابه با الگوریتم‌های افزایشی [۲۶]، پیاده‌سازی شده است. در شکل (۳) و جدول (۹) صحت، F1-score و زمان اجرای الگوریتم پیشنهادی با الگوریتم‌های افزایشی مرجع [۲۶] مقایسه شده و نشان داده شده است که الگوریتم پیشنهادی نه

تنها دارای صحت و F1-score بهتری نسبت به الگوریتم‌های افزایشی مرجع [۲۶] است، بلکه از نظر زمان اجرا هم زمان اجرای کمتری دارد.

در نهایت، صحت مجموعه داده‌های مورد بررسی در شکل (۵) نشان داده شده است. همان‌طور که مشاهده می‌شود در تمامی نمودارها پس از طی کردن تقریبا ۱۰۰ دور از آموزش صحت روند صعودی پیدا می‌کند و این ثمره استفاده از یادگیری تقویتی در الگوریتم پیشنهادی است.



شکل (۳): مقایسه صحت و F1-SCORE الگوریتم پیشنهادی با الگوریتم‌های افزایشی مرجع [۲۶]



شکل (۴): نمودار صحت داده‌ها

جدول (۹): مقایسه زمان اجرای الگوریتم پیشنهادی با الگوریتم‌های افزایشی مرجع [۲۶]

Algorithms	Accuracy (برحسب درصد)	Run Time (برحسب ثانیه)
IKNN	۹۶/۵۰	۴۲
IDT	۷۷/۵۲	۳۸
IRF	۹۰/۷۷	۱۶۷
Iadaboost	۸۷/۹۴	۴۷
INB	۹۵/۸۳	۴۰
Learn++NSE	۹۰/۷۸	۷۴
Inc-RL-SGD	۹۷/۳۰	۳۳

۷. نتیجه‌گیری

در این مقاله برای طبقه‌بندی داده‌های جریانی با صحت و زمان اجرای مناسب یک الگوریتم افزایشی با استفاده از یادگیری تقویتی به همراه دسته‌بند گرادیان کاهشی تصادفی به منظور یادگیری افزایشی ارائه شد. در الگوریتم پیشنهادی از دسته‌بند گرادیان کاهشی تصادفی برای طبقه‌بندی داده‌ها و از یادگیری تقویتی برای انتخاب کنش مناسب استفاده می‌شود. الگوریتم ارائه شده بر روی انواع مجموعه داده‌ها اجرا شد، که نتایج آزمایشگاهی از کارایی بالای این الگوریتم نسبت به نمونه‌های مشابه آن در نسخه‌های متفاوت الگوریتم‌های افزایشی حکایت دارد. در الگوریتم ارائه شده بنا به دلیل ارائه کنشی جدید (بازآموزی نمونه‌های جدید به وسیله نمونه‌های مفید) زمان اجرا کاهش یافته است و همچنین به دلیل در نظر گرفتن معیار زمان اجرا، برتری بیشتری نسبت به سایر الگوریتم‌های افزایشی ذکر شده دارد.

تعارض منافع: نویسندگان اعلام می‌کنند که هیچ تعارض منافعی ندارند.

مراجع

[1] D. Bhattacharya and M. Mitra, "Analytics on big fast data using real time stream data processing architecture," EMC Corp., 2013.

[2] R. Polikar, L. Upda, S. S. Upda, and V.G. Honavar, "Learn++: an incremental learning algorithm for supervised neural networks," IEEE Trans. Syst. Man Cybern. Part C, vol. 31, no. 4, pp. 497-508, 2001, doi: 10.1109/5326.983933.

[3] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in Mach. Learn. Proc. 21st Int. Conf. (ICML), Banff, Alberta, Canada, 2004, p. 116, doi: 10.1145/1015330.1015332.

[4] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, "A comprehensive survey of loss functions in machine learning," Ann. Data Sci., vol. 9, pp. 187-212, 2022, doi: 10.1007/s40745-020-00253-5.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M.A. Riedmiller, "Playing atari with deep reinforcement learning," arXiv preprint arXiv: 1312.5602, 2013.

[6] M.G. Lagoudakis and R. Parr, "Reinforcement learning as classification: leveraging modern classifiers," in Mach. Learn. Proc. 20th Int. Conf. (ICML), Washington, DC, USA, 2003, pp. 424-431.

[7] S. Maliah and G. Shani, "MDP-based cost sensitive classification using decision trees," in Proc. 22nd AAAI Conf. Artif. Intell., New Orleans, Louisiana, USA, 2018, pp. 3746-3753, doi: 10.1609/aaai.v32i1.11779.

[8] S.J. Chen, A. Choi, and A. Darwiche, "Value of information based on Decision Robustness," in Proc. 29th AAAI Conf. Artif. Intell., Austin, Texas, USA, 2015, pp. 3503-3510, doi: 10.1609/aaai.v29i1.9684.

[9] L.-P. Liu, Y. Yu, Y. Jiang, and Z.-H. Zhou, "Tefe: A time-efficient approach to feature extraction," in Proc. 8th IEEE Int. Conf. Data Mining (ICDM), Pisa, Italy, 2008, pp. 423-432, doi: 10.1109/ICDM.2008.48.

[10] A. Khosravi, H. Abdulmaleki, and M. Fayazi, "Predicting the academic status of admitted applicants based on educational and admission data using data mining techniques," Soft Comput. J., vol. 9, no. 2, pp. 94-113, 2021, doi: 10.22052/scj.2021.242837.0 [In Persian].

[11] F. Zare Mehrjardi, M. Yazdian-Dehkordi, and A. Latif, "Evaluating classical machine learning and deep-learning methods in sentiment analysis of Persian telegram message," Soft Comput. J., vol. 11, no. 1, pp. 88-105, 2022, doi: 10.22052/scj.2023.246553.1077 [In Persian].

[12] H. Veisi, H.R. Ghaedsharaf, and M. Ebrahimi, "Improving the Performance of Machine Learning Algorithms for Heart Disease Diagnosis by Optimizing Data and Features," Soft Comput. J., vol. 8, no. 1, pp. 70-85, 2019, doi: 10.22052/8.1.70 [In Persian].

[13] V. Losing, B. Hammer, and H. Wersing, "Incremental on-

line learning: A review and comparison of state of the art algorithms," Neurocomputing, vol. 275, pp. 1261-1274, 2018, doi: 10.1016/j.neucom.2017.06.084.

[14] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in 19th Int. Conf. Comput. Stat. (COMPSTAT), Paris, France, 2010, pp. 177-186, doi: 10.1007/978-3-7908-2604-3_16.

[15] P. Richtarik and M. Takac, "Parallel coordinate descent methods for big data optimization," Math. Program., vol. 156, no. 1-2, pp. 433-484, 2016, doi: 10.1007/s10107-015-0901-6.

[16] H. Zhang, "The optimality of naive Bayes," in Proc. 17th Int Florida Artif. Intell. Res. Soc. Conf., Miami Beach, Florida, USA, 2004, pp. 562-567.

[17] C. Salperwyck and V. Lemaire, "Learning with few examples: An empirical study on leading classifiers," in Int. Joint Conf. Neural Networks (IJCNN), San Jose, California, USA, 2011 pp. 1010-1019, doi: 10.1109/IJCNN.2011.6033333.

[18] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes?," in 3rd Conf. Email Anti-Spam (CEAS), Mountain View, California, USA, 2006, pp. 28-69.

[19] S.L. Ting, W.H. Ip, and A.H.C. Tsang, "Is Naive Bayes a good classifier for document classification," Int. J. Softw. Eng. Appl., vol. 5, no. 3, pp. 37-46, 2011.

[20] Y. Freund and R.E. Schapire, "A short introduction to boosting," J. Japanese Soc. Artif. Intell., vol. 14, no. 5, pp. 771-780, 1999.

[21] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in 12th IEEE Int. Conf. Comput. Vis. Worksh. (ICCV), Kyoto, Japan, 2009, pp. 1393-1400, doi: 10.1109/ICCVW.2009.5457447.

[22] V. Losing, B. Hammer, and H. Wersing, "Interactive online learning for obstacle classification on a mobile robot," in Int. Joint Conf. Neural Networks (IJCNN), Killarney, Ireland, 2015, pp. 1-8, doi: 10.1109/IJCNN.2015.7280610.

[23] A. Sato and K. Yamada, "Generalized learning vector quantization," in Adv. Neural Inf. Process. Syst., NIPS, Denver, CO, USA, 1995, pp. 423-429.

[24] N.A. Syed, H. Liu, and K.K. Sung, "Handling concept drifts in incremental learning with support vector machines," in Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, San Diego, CA, USA, 1999, pp. 317-321, doi: 10.1145/312129.312267.

[25] H. Zhao, H. Wang, Y. Fu, F. Wu, and X. Li, "Memory-Efficient Class-Incremental Learning for Image Classification," IEEE Trans. Neural Networks Learn. Syst., vol. 33, no. 10, pp. 5966-5977, 2022, doi: 10.1109/TNNLS.2021.3072041.

[26] M. Vakili and M. Rezaei, "Incremental Learning Techniques for Online Human Activity Recognition," arXiv preprint arXiv: 2109.09435, 2021.

[27] I. Lee, H. Roh, and W. Lee, "Poster abstract: encrypted malware traffic detection using incremental learning," in 39th IEEE Conf. Comput. Commun. (INFOCOM), Toronto, ON, Canada, 2020, pp. 1348-1349, doi: 10.1109/INFOCOMWKSHPS50562.2020.9162971.

[28] T.-N. Do, "ImageNet Challenging Classification with the Raspberry Pi: An Incremental Local Stochastic Gradient Descent Algorithm," arXiv preprint arXiv: 2203.11853, 2022.

[29] S. Agarwal, A. Rattani, and C.R. Chowdary, "AILearn: An Adaptive Incremental Learning Model for Spoof Fingerprint Detection," arXiv preprint arXiv: 2012.14639, 2020.

[30] J. Huo and T.L. van Zyl, "Incremental class learning using variational autoencoders with similarity learning," Neural Comput. Appl., vol. 37, no. 2, pp. 769-784, 2025, doi: 10.1007/s00521-023-08485-1.

[31] G. Shan, S. Xu, L. Yang, S. Jia, and Y. Xiang, "Learn#: a novel incremental learning method for text classification," Expert Syst. Appl., vol. 147, p. 113198, 2020, doi: 10.1016/j.eswa.2020.113198.

[32] A.C. Lemos Neto, R.A. Coelho, and C.L.d. Castro, "An incremental learning approach using long short-term memory neural networks," J. Control Autom. Electr. Syst., vol. 33, pp. 1457-1465, 2022, doi: 10.1007/s40313-021-00882-y.

[33] B. Nemade and D. Shah, "An efficient IoT based prediction system for classification of water using novel adaptive incremental learning framework," J. King Saud Univ. Comput. Inf. Sci., vol. 34, no. 8, pp. 5121-5131, 2022, doi: 10.1016/j.jksuci.2022.01.009.

[34] M. Kang, J. Park, and B. Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation," in IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), New Orleans, LA, USA, 2022, pp. 16050-16059, doi: 10.1109/CVPR52688.2022.01560.

[35] R. Khemchandani, Jayadeva, and S. Chandra, "Incremental twin support vector machines," Model. Comput. Optim., pp. 263-272, 2009, doi: 10.1142/9789814273510_0017.