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

سمیرا فرزانه[1]، دانشجوی کارشناسی ارشد، جواد سلیمی سرتختی[2*]، استادیار

[1] دانشکده مهندسی برق و کامپیوتر- دانشگاه کاشان ـ کاشان ـ ایران ـ samira.farzaneh@grad.kashanu.ac.ir

[2] دانشکده مهندسی برق و کامپیوتر ـ دانشگاه کاشان ـ کاشان ـ ایران ـ salimi@kashanu.ac.ir

**چکیده: حجم وسیعی از تحقیقات در زمینه‌ی یادگیری برخط به مسئله غلبه بر فراموشی فاجعه‌بار تمرکز کرده‌اند و تحقیقات اندکی در زمینه‌ی 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 مقایسه‌شده 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.**

**واژه‌های کلیدی: داده‌های جریانی، صحت و زمان اجرا، گرادیان کاهشی تصادفی، یادگیری افزایشی، یادگیری تقویتی**

* نویسنده مسئول: جواد سلیمی سرتختی، ایمیل: **salimi@kashanu.ac.ir**

# Designing a Model for Data Stream Classification Using RL and SGD

Samira Farzaneh [1], Masters Student, Javad Salimi Sartakhti [2*], Assistant Professor

[1] Faculty of Electrical and Computer Engineering, University of Kashan, Kashan, Iran, samira.farzaneh@grad.kashanu.ac.ir

[2] Faculty of Electrical and Computer Engineering, University of Kashan, Kashan, Iran, salimi@kashanu.ac.ir

***Abstract***: A large amount of research in the field of online learning has focused on the problem of overcoming catastrophic forgetting, and few researches have been focused on the classification of the data stream with appropriate accuracy and running time. On the other hand, due to the volume and type of data stream, many traditional machine learning algorithms do not have the necessary efficiency when faced with it.

Thus, in this paper, a novel model using reinforcement learning and stochastic gradient descent algorithm is presented for classification stream data with appropriate accuracy and running time. One of the important features of reinforcement learning is that the agent can adapt its behaviour gradually to the changes that occur and gradually add to its previous knowledge. In this research, because of the using of reinforcement learning and definition of reward, the agent has a better performance in the environment. The proposed algorithm has been tested on various data, including the dataset of human activity recognition, and compared with several incremental algorithms in terms of accuracy and running time. According to the experimental results, the proposed algorithm has the best performance both in terms of accuracy and running time compared to other incremental algorithms.

**Keywords:** *Accuracy and Running Time; Data Stream; Incremental Learning; Reinforcement Learning; Stochastic Gradient Descent.*

*** corresponding author: Javad Salimi Sartakhti, Email: salimi@kashanu.ac.ir**

## ۱. مقدمه

امروزه حجم عظیمی از داده‌ها در حوزه‌ها و زمینه‌های مختلف تولید می‌شوند و اغلب داده‌های تولیدشده 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 تولیدشده و نوع آنها باعث به وجود آمدن چالش‌هایی شده‌اند. یکی از مسائل چالش برانگیز، مسئله‌ی 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 براین است که تمامی داده‌ها به صورت یک‌جا و با توزیع یکسان در اختیار مدل قرار دارند و مدل می‌تواند از تمامی داده‌های جمع‌آوری‌شده 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، داده‌های جریانی عموما همراه با تغییرتوزیع 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[2] مطرح شده است [۲]. این نوع الگوریتم غالبا برای کلان داده‌ها اعمال می‌شود. در این نوع یادگیری مرحله‌ی آموزش و یادگیری در طول زمان و به طور مداوم صورت می‌گیرد.

الگوریتم گرادیان کاهشی تصادفی[3] 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، پردازش متن و... 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 نمونه‌ی جدید، مدل بازآموزش شود. اما چنین روشی بسیار زمانبر و غیر ممکن 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. ۲) پیروی داده‌های جریان‌های مختلف از توزیع‌های متفاوت. بنابراین، سوالی که مطرح می‌شود، آیا با ورود هر نمونه‌ی جدید باید مدل بازآموزش شود یا خیر؟

در این مقاله قصد بر این است به چالش‌ها و سوال ذکرشده 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

[1] Data Stream
[2] Incremental Learning
[3] Stochastic Gradient Descent

بررسی کند که آیا نمونه به مدل اضافه شود و مدل مجددا اجـرا شود یا خیر.

در مدل پیشنهادی از چندین مجموعه داده‌ی متعلق به بانـک داده‌ی UCI و مجموعه داده‌های جریانی تشخیص فعالیـت‌هـای انسانی برای آزمایش عملکرد مدل استفاده شده است. شایان ذکر است، عملکرد مدل پیشنهادی با سـایر الگـوریتم‌هـای یـادگیری افزایشی مقایسه شده اسـت و نشـان داده‌شـده اسـت کـه مـدل پیشنهادی نسبت به سایر الگوریتم‌های افزایشـی دارای عملکـرد بهتری است.

ادامه‌ی مقاله به صورت زیر سازماندهی شده است. در بخش دوم به تعریف مفاهیم اولیه، در بخش سوم بطور مختصرکارهای گذشته مرور می‌شود، در بخش چهارم به توضیح الگوریتم پیشنهادی و در بخش پنجم به ارزیابی پرداخته می‌شود. در نهایت در بخش پایانی نتیجه‌گیری ارائه می‌شود.

## ۲. مفاهیم پایه

به منظور طبقه‌بندی نمودن داده‌های جریانی بـا صـحت و زمـان اجرای مناسب از الگوریتم‌های سنتی یادگیری ماشین نمـی‌تـوان استفاده نمود. بنابراین، مدلی مبتنی بر یادگیری تقویتی به همـراه دسته‌بند گرادیان کاهشی تصادفی بـه منظـور یـادگیری افزایشـی طراحی شده است. در ادامه تمامی موارد فوق به تفصیل توضـیح داده شده است.

### ۱.۲. گرادیان کاهشی تصادفی

گرادیان کاهشی تصادفی یک الگوریتم بهینه‌سازی ساده و در عین حال کارآمد برای بهینه‌سازی تابع زیان[4] 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) بدست می‌آید و سپس برای هر تکرار t گرادیان تخمین زده می‌شود:

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

که $n$ برابر با تعداد کل نمونه‌های مجموعـه آموزشـی و $\mathcal{L}$ تـابع زیاتی است که باید کمینه شود.

همانگونه که در رابطه (۱) مشاهده مـی‌گـردد بـرای تخمـین گرادیان باید میانگین همه نمونه‌ها محاسبه شـود. بنـابراین، اگـر داده‌ها از نوع کلان داده باشند، محاسبه گرادیان بـه روش فـوق-الذکر بسیار زمانبر است. برای ممانعت از بـروز چنـین مشـکلی گرادیان کاهشی تصادفی مطرح شد. در این نـوع گرادیـان طبـق رابطه (۲) در هر تکرار برای دسته‌ای از نمونه‌هـا بـه جـای کـل نمونه‌ها، گرادیان تخمین زده می‌شود:

$$g_{GD} = \frac{1}{n'}\sum_{i=1}^{n'} \nabla_{\theta} \mathcal{L}\,(x^i, y^i, \theta_t) \qquad (۲)$$

که $n'$ بیانگر نمونه‌هـایی هسـتند کـه از مجموعـه آموزشـی انتخـاب مـی‌شـوند و انـدازه‌اش نسـبت بـه مجموعـه آموزشـی کوچکتر است.

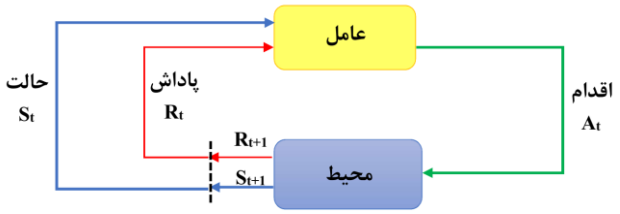
### ۲.۲. یادگیری تقویتی

یکی از تکنیک‌های اصـلی یـادگیری ماشـین، یـادگیری تقـویتی است. این نوع یادگیری برای محیط‌هـایی کـه عامـل هـیچ نـوع شناختی نسبت به آنها نـدارد، مناسـب اسـت [۵]. در ایـن نـوع

[4] Loss Function

یادگیری به عامل[5] 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[6] $S = \{s_i\}$، مجموعه کنش‌ها[7] $A = \{a_i\}$، تابع انتقال حالت[8] $T(s, a, s')$ که حالت بعدی پس از اجرای کنش $a$ را مشخص می‌کند، تابع پاداش[9] $R(s, a)$ که بیانگر میزان پاداش انتخاب کنش $a$ در حالت $s$ است.

در شکل (۱) می‌توان نحوه‌ی 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. رویه‌ی یادگیری افزایشی بدین صورت است که داده‌های ورودی به طور مداوم برای گسترش دانش موجود به مدل اضافه می‌شوند. بنابراین، مرحله‌ی 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 دسته‌ی بازآموز و غیر بازآموز تقسیم نمود. در الگوریتم‌های غیر بازآموز هنگامی که نمونه‌های جدیدی به مدل اضافه می‌شوند، نیازی به بازآموزی با تمام نمونه‌های آموزشی نیست. بنابراین، مصرف حافظه و هزینه‌ی زمان آموزش کاهش می‌یابد. اما در الگوریتم‌های بازآموز، تمامی نمونه‌ها نگهداری می‌شوند و هنگامی که نمونه‌ی 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، جنگل‌های تصادفی و ... به منظور طبقه‌بندی نمودن 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 [۱۳] در پژوهشی اغلب الگوریتم‌های معروف یادگیری افزایشی را بر روی مجموعه داده‌های مختلفی آزمایش کردند که در ادامه به توضیح آنها پرداخته می‌شود.

برخی از محققین [۳, ۱۴, ۱۵] الگوریتم‌هایی مبتنی بر یادگیری افزایشی با استفاده از الگوریتم گرادیان کاهشی تصادفی با به حداقل رساندن تابع ضرر برای یادگیری یک مدل خطی ارائه کرده‌اند. به عنوان مثال، الگوریتم گرادیان کاهشی تصادفی

---

[5] Agent
[6] States
[7] Actions
[8] Transition Function
[9] Reward

یک روش بهینه‌سازی کارآمد برای یادگیری یک مدل متمایز با به حداقل رساندن تابع ضرر مانند تابع لاجستیک[10] یا تابع هینج[11] است. الگوریتم آنها برای داده‌های پراکنده و با ابعاد بالا که اغلب در حوزه‌ی 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[12] ارائه کرده است. مدل خلوت[13] 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[14] تک لایه ارائه کردند. الگوریتمشان به عنوان ماشین یادگیری افراطی متوالی آنلاین[15] شناخته می‌شود و می‌تواند داده‌ها را تک به تک یا قسمت به قسمت (یک بلاک از داده‌ها) با اندازه‌ی قسمت ثابت یا متغیر یاد بگیرد که به طور قابل توجهی زمان پردازش کلی را کاهش می‌دهد.

صفری و همکاران [۲۱] یک نسخه‌ی افزایشی از الگوریتم جنگل تصادفی تحت عنوان جنگل تصادفی برخط[16] را ارائه کرده‌اند. رویکرد آنها ترکیب کیسه‌بندی[17] برخط و جنگل‌های تصادفی است. رویکردشان با حذف کردن 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، رویکردشان می‌تواند در برنامه‌هایی با نویز زمانی یا تغییرات ظاهری بالا در طول ردیابی بصری مفید باشد.

لوزینگ و همکاران [۲۲] یک الگوریتم کوانتیزه سازی بردار یادگیری افزایشی[18] که اقتباسی از کوانتیزه سازی بردار یادگیری تعمیم‌یافته‌ی استاتیک[19] [۲۳] است، ارائه کرده‌اند. در رویکردشان رویکردشان تابع ضرر را به گونه دیگری تعریف می‌کنند که در صورت لزوم نمونه‌های اولیه جدیدی را به مدل اضافه می‌کند. نرخ درج بر اساس تعداد نمونه‌های طبقه‌بندی اشتباه صورت می‌گیرد.

پلیکار و همکاران [۲] الگوریتم Learn++ را مطرح نموده‌اند. در الگوریتم آنها نمونه‌های دریافتی در تکه‌هایی[20] با اندازه‌ی از پیش تعریف‌شده پردازش می شود. برای هر قسمت، گروهی[21] از طبقه‌بندی‌کننده‌های پایه آموزش داده می‌شوند و از طریق رای اکثریت وزنی به «گروهی از گروه‌ها» ترکیب می‌شوند. مشابه الگوریتم AdaBoost [۲۰]، هر طبقه بندی‌کننده با زیر مجموعه‌ای از نمونه‌های قسمتی که بر اساس توزیع داده‌ها است، آموزش داده می‌شود.

سید و همکاران [۲۴] رویکرد ماشین بردار پشتیبان افزایشی به منظور طبقه‌بندی داده‌ها ارائه کرده‌اند. رویکرد آنان 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

[10] Logistic Loss
[11] Hinge Loss
[12] Naive Bayesian Gauss
[13] Sparse
[14] Feed Forward
[15] Online Sequential Extreme Learning Machine
[16] Online Random Forest
[17] Bagging

[18] Incremental Learning Vector Quantization
[19] Generalized Learning Vector Quantization
[20] Chunks
[21] Ensemble

حذف می‌شوند. ولی هرگاه نمونه جدیدی به مجموعه داده قبلی اضافه می‌شود، مدل از ابتدا اجرا می‌شود. با این حال، رویکرد مذکور دارای معایبی است که عبارتند از: ۱- نیاز به حافظه‌ی بزرگ. ۲- زمان اجرای طولانی. ۳- فرض یکسان بودن توزیع داده‌ها.

همانطور که قبلا بیان کردیم، یکی از معایب یادگیری افزایشی مشکل فراموشی فاجعه‌بار است. از این‌رو، برخی از محققین الگوریتم‌هایی را برای مقابله با مسئله مذکور ارائه کردند، اما اکثر الگوریتم‌های ارائه‌شده دارای بار محاسباتی زیادی هستند و به حافظه بزرگی نیاز دارند. بنابراین، ژاو و همکاران [۲۵] یک الگوریتم یادگیری افزایشی مبتنی بر بافر حافظه محدود مطرح کردند. در الگوریتم مذکور به منظور حفظ دانش قدیمی تعدادی نمونه در حافظه نگهداری می‌شود. شایان ذکر است، اگر تعداد نمونه‌های ذخیره‌شده در حافظه کم باشد، سبب کاهش دقت مدل می‌شود. از سوی دیگر، برای افزایش دقت مدل نیاز به ذخیره نمونه‌های زیادی است که این امرمنجر به افزایش هزینه می‌شود. بنابراین، در الگوریتم مذکور از یک روش نمونه‌کاهی[۲۲] 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 ذخیره‌شده قبلی بازآموز 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، دسته‌بند جنگل تصادفی و...، همچنین برخی از الگوریتم‌ها همانند الگوریتم‌های [۲] و [۲۰] را به نسخه افزایشی تبدیل کرده‌اند. آنها مجموعه داده‌ی تشخیص فعالیت‌های انسانی را به منظور بررسی صحت و صحت الگوریتم‌هایشان 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 رمز-گذاری‌شده 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[۲۳] 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 می‌شوند 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 آموزش‌دیده قبلی در حین یادگیری اثرانگشت‌های جدید نیست، که این امر منجر به نیاز به حافظه کم می‌شود. با این حال، اگر از مجموعه داده خیلی بزرگ استفاده شود، سبب افزایش بار محاسباتی می‌شود.

---

[22] Down Sampling

[23] k-means

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[۲۴] را برای طبقه‌بندی داده‌ها مطرح کردند. به منظور استخراج ویژگی‌ها از شبکه عصبی خود رمزگذار متغیر استفاده می‌شود. سپس از یک شبکه از پیش‌آموزش‌دیده برای طبقه‌بندی داده‌ها استفاده می‌شود. همچنین خروجی‌های خود رمزگذار متغیر به عنوان داده کمکی برای داده‌های جدید استفاده می‌شود. 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 داده خیلی بزرگ استفاده شود زمان آموزش مدل افزایش می‌یابد و سبب افزایش بار محاسباتی می‌شود.

شان و همکاران [۳۱] به منظور طبقه‌بندی داده‌های متنی الگوریتمی مبتنی بر شبکه عصبی حافظه کوتاه مدت طولانی[۲۵] را مطرح کردند. الگوریتم آنها شامل ۴ مولفه دانش‌آموز، یادگیری تقویتی، معلم و متمایزکننده است. در مولفه دانش‌آموز ابتدا پیش- بینی‌های اولیه توسط شبکه عصبی مذکور صورت می‌گیرد. سپس مولفه یادگیری تقویتی نتایج چندین مولفه دانش‌آموز را فیلتر می‌کند. پس از آن، مولفه معلم نتایج فیلترشده را مجددا طبقه‌بندی می‌کند و طبقه‌بندی متون نهایی را بدست می‌آورد و در نهایت برای جلوگیری از تعداد افزایش نامحدود مولفه‌های دانش‌آموز با افزایش تعداد نمونه‌ها، از مولفه متمایزکننده برای فیلتر کردن مولفه‌های دانش آموز 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 مسئله فراموشی فاجعه‌بار، تغییر توزیع داده‌ها و مدیریت حافظه استفاده شود. نتایج تجربی نشان می‌دهد که الگوریتم مذکور 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. سپس، 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 کسب‌شده برای طبقه‌بندی کیفیت آب استفاده می‌شود. اگرچه الگوریتم مذکور 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 برای طبقه‌بندی داده‌ها و اضافه شدن کلاس جدید به مجموعه کلاس‌های قدیمی داده‌ها پیشنهاد

---

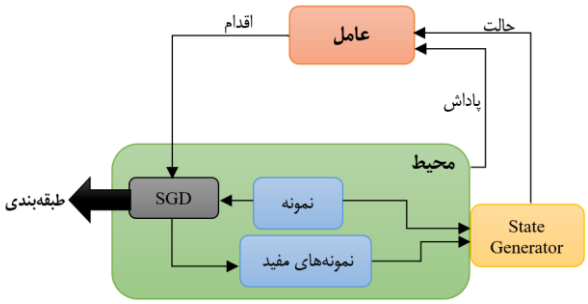
[24] Variational Autoencoder

[25] Long-Short Term Memory (LSTM)

کردند. در الگوریتم آنها به طور مداوم دانش جدید با حافظه محدود یاد گرفته می‌شوند در حالیکه 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 می‌شود. سپس، ویژگی‌های استخراج‌شده 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 الگوریتم از عملکرد خوبی برخوردار است، اما به دلیل استفاده از شبکه عصبی عمیق زمانبر است.

## ۴. الگوریتم پیشنهادی

همانطور که قبلا بیان کردیم، الگوریتم‌های ارائه‌شده از یادگیری افزایشی به همراه دسته‌بندهای مختلف برای طبقه‌بندی نمودن داده‌ها استفاده‌شده است، 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 شده است به این معیار توجهی نشده است. بنابراین، در این پژوهش قصد بر این است که برای طبقه‌بندی داده‌های مورد بررسی با صحت و زمان اجرای مناسب از یادگیری تقویتی به همراه دسته‌بند گرادیان کاهشی تصادفی به منظور 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 تعداد پارامترهای آن دسته‌بند کم باشد، بتوان دسته‌بند را در قالب یک بردار نمایش داد و 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

همانگونه که قبلا بیان کردیم، گرادیان کاهشی تصادفی یک الگوریتم بهینه‌سازی است. برخی از محققان یک الگوریتم یادگیری افزایشی مبتنی بر گرادیان کاهشی تصادفی و دسته‌بندها مطرح کردند [۳, ۱۴]. آنها در الگوریتمشان 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].

همانطور که قبلا ذکر شده است؛ هدف از یادگیری تقویتی به حداکثر رساندن پاداش تجمعی است [۵] و هدف در این پژوهش دستیابی به صحت و زمان اجرای مناسب است. نحوه‌ی یادگیری تقویتی برای دستیابی به پاداش مورد نظر بدین صورت است که ابتدا مقداردهی اولیه مدل صورت می‌گیرد و سپس وارد مرحله‌ی 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 بدست‌آمده 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 درنظرگرفت. هنگامی که حالت جدید دریافت می‌شود از بین کنش‌های موجود، عامل باید بهترین کنش را انتخاب کند.

در حالت کلی، برای مسئله 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، 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 مجددا بازآموزی کنیم؛ اما این روش بسیار زمانبر است. بنابراین، کنش جدیدی را می‌توان تعریف کرد تا به هدف اصلی پژوهش دست یاقت. 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 صرفاً هدف دستیابی به بیشترین صحت طبقه‌بندی باشد تنها معیار صحت کافی است و می‌توان با پذیرش نمونه جدید مدل را مجددا بازآموزی نمود تا به بهترین صحت ممکن دست یافت. ولی عملا این روش بسیار زمانبر است. بنابراین، می‌بایست توازنی بین صحت و زمان در نظر گرفته شود تا هر دفعه با اضافه شدن نمونه جدید، مدل از ابتدا یازآموزی 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 اگـر در تـابع پـاداش دو معیـار صحت و زمان یادگیری در نظر گرفته شود، طبقه‌بندی داده‌ها بـا 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 ارائـه شـده است که در ادامه به توضیح آنها پرداخته می‌شود.

**الگوریتم (۱): مرحله آموزش الگوریتم Inc-RL-SGD**

- INPUT: Training data D = {($x_1$, $y_1$), ($x_2$, $y_2$),…, ($x_t$, $y_t$)}
- 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 = {($x_1$, $y_1$), ($x_2$, $y_2$),…, ($x_t'$, $y_t'$ ) where t'<t
- Initialize simulation environment
- For episode k = 1 to K
  - Receive sample $x_t$
  - Shuffle the training data D
  - Initialize state $s_t = x_t$ □ $M_{t-1}$
  - 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 $s_{t+1} = x_{t+1}$ □$M_t$
  - 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(θ)

**الگوریتم (۲): مرحله شبیه‌سازی محیط الگوریتم Inc-RL-SGD**

- **FUNCTION**: STEP($a_t$ ∈A, ($x_t$, $y_t$) ∈D, $M_{t-1}$)
- If $a_t$ = TA
  - $M_{t+1}$ ←Train SGD by all samples ∈$D_t$;
  - $US_{t+1} = US_t$ ∪ New useful samples ∈ $M_t$;
- If $a_t$ = TUS
  - $M_{t+1}$ ←Train SGD by $US_t$;
  - $US_{t+1} = US_t$ ∪ New useful samples ∈ $M_t$;
- 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 (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 می‌شود. 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 می‌شود 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، در هر دور از مرحله یادگیری نمونه‌های مفید تولیدشده ذخیره می‌شوند. در گام بعد بردار حالت و کنش انتخاب‌شده به محیط ارسال می‌شوند.

همانطور که در الگوریتم (۲) مشاهده می‌شود، آرگومان‌های تابع STEP نمونه، مدل و کنش انتخاب‌شده از مجموعه کنش‌های A است. اگر عامل کنش TA را انتخاب کرد، آنگاه دسته‌بند با تمام نمونه‌ها بازآموزی می‌شود و همچنین نمونه مفید جدید بدست‌آمده 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 بدست‌آمده بازآموزی می‌شود و همچنین نمونه مفید جدید بدست‌آمده در

این مرحله به مجموعه نمونه‌های مفید قبلی اضافه می‌شود. در غیر 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 می‌شود. در نهایت به منظور سنجش مدل از نظر قابلیت و توانایی در پیش‌بینی مقدارهای جدید از تابع زیان استفاده می‌شود.

## ۵. ارزیابی

برای پیاده سازی مدل پیشنهادی و پیش پردازش داده‌های ورودی از زبان برنامه نویسی 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، الگوریتم پیشنهادی با چندین الگوریتم افزایشی مقایسه شده است. بنابراین، در هر بار مقایسه با الگوریتم مورد بررسی پارامترهایی از قبیل نرخ یادگیری، وزن‌های شبکه، تابع زیان، تعداد دورهای اجرا و... 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] و سید [۲۴] و وکیلی [۲۶] با استفاده از تعدادی مجموعه داده‌های بانک مخزن UCI و مجموعه داده-ی تشخیص فعالیت‌های انسانی مقایسه شده است.

از این‌رو، مدل پیشنهادی با شرایطی مشابه الگوریتم‌های مورد مقایسه [۲, ۳, ۱۳, ۱۶, ۲۰-۲۲, ۲۴, ۲۶, ۳۵] پیاده‌سازی شده است تا مقایسه در شرایط عادلانه صورت گیرد. بنابراین، در الگوریتم‌های مورد مقایسه نسبت نمونه‌های اموزشی به تست 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 کثیفی[۲۶] هستند. بنابراین، ابتدا داده‌ها با استفاده از روش‌های مرسوم پاکسازی که سایر الگوریتم‌ها از آنها استفاده کرده‌اند، پاکسازی می‌شوند. سپس از آنها در الگوریتم پیشنهادی استفاده می‌شود.

در ابتدا به علت عدم شناخت محیط و نبودن نمونه‌های مفید عامل به صورت تصادفی یکی کنش‌های ۱) پذیرفتن نمونه‌ی جدید و بازآموزی مدل توسط گرادیان کاهشی تصادفی یا ۲) نپذیرفتن نمونه‌ی 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 موردنظر 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 می‌شود و در ادامه معیارهای ارزیابی دسته‌بند معرفی و نهایتا نتایج حاصل از اعمال الگوریتم پیشنهادی بر روی مجموعه داده‌های مورد بررسی به تفصیل بیان می‌شود.

---

[26] Dirty Data

### ۱.۶. مجموعه داده‌ها

برای بررسی قدرت و صحت الگوریتم پیشنهادی از چندین مجموعه داده‌ی متعلق به بانک مخزن UCI و مجموعه داده‌های مربوط به تشخیص فعالیت‌های انسانی استفاده‌شده است. طبق مرجع [۲۶] داده‌های تشخیص فعالیت‌های انسانی توسط حسگرهای گوشی‌های هوشمند و از طریق جریان‌های بلادرنگ اینترنت اشیا تولید شده‌است. در این مجموعه داده حداکثر ۲۰ فعالیت وجود دارد. در این پژوهش منظور از فعالیت همان برچسب[۲۷] 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 داده‌شده 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 اعمال 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 استفاده‌شده 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 می‌شود و نشان داده می‌شود که الگوریتم پیشنهادی دارای صحت بهتری نسبت به سایر الگوریتم‌ها است. شایان ذکر است، الگوریتم پیشنهادی در شرایطی مشابه با [۲, ۳, ۱۳, ۱۶, ۲۰-۲۲, ۲۴, ۳۵] پیاده‌سازی شده است و نتایج مربوط به الگوریتم‌های افزایشی از [۱۳, ۳۵] اخذ شده است تا شرایطی عادلانه برای مقایسه وجود داشته باشد. جداول (۳) و (۵) نتایج حاصل از الگوریتم پیشنهادی و الگوریتم‌های مورد مقایسه را نشان می‌دهد. همانگونه که مشاهده می‌شود الگوریتم پیشنهادی در همه مجموعه داده‌ها از صحت بهتری نسبت به سایر الگوریتم‌های افزایشی برخوردار است.

[27] Label

شایان ذکر است، در الگوریتم [۲, ۳, ۱۳, ۱۶, ۲۰-۲۲, ۲۴, 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 می‌شود. سپس یادگیری تقویتی با توجه 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 الگوریتم [۲۴, ۳۵] دارد.

یکی‌دیگر از معیارهای مهم زمان اجرا است. لازم به ذکر است که تمامی نسخه‌های افزایشی الگوریتم‌ها از نسخه غیر افزایشی آنها زمان اجرای به مراتب بیشتری دارند. این امر هم به آن دلیل است که عموماً در یک الگوریتم افزایشی یک مدل ممکن است هزاران بار اجرا شود که قاعدتاً 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 جداول (۴) و (۶) زمان اجرای مجموعه داده‌های مذکور نشان داده‌شده است و همانطور که مشاهده می‌شود الگوریتم پیشنهادی از زمان اجرای خوبی برخوردار است.

لازم به ذکر است که به دلیل نامتوازن 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 این‌رو، در جدول (۸) میزان F1-score الگوریتم پیشنهادی یا الگوریتم Inc-SVM مقایسه شده است. همانطور که مشاهده می‌شود، الگوریتم پیشنهادی به دلیل استفاده از یادگیری تقویتی از عملکرد بهتری برخوردار است.

در کاربردی دیگر به منظور بررسی صحت و دقت الگوریتم پیشنهادی از مجموعه داده‌های مربوط به تشخیص فعالیت‌های انسانی استفاده شده‌است. شایان ذکر است الگوریتم پیشنهادی در شرایطی مشابه با الگوریتم‌های افزایشی [۲۶] پیاده‌سازی شده‌است. در شکل (۳) و جدول (۹) صحت، F1-score و زمان اجرای الگوریتم پیشنهادی با الگوریتم‌های افزایشی [۲۶] مقایسه شده‌است و نشان داده شده‌است که الگوریتم پیشنهادی نه تنها دارای صحت و F1-score بهتری نسبت به سایر الگوریتم‌های افزایشی [۲۶] است بلکه از نظر زمان اجرا هم زمان اجرای کمتری دارد.

صحت مجموعه داده‌های مورد بررسی در شکل (۴) 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 می‌باشد.

## نتیجه گیری

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 به منظور طبقه‌بندی داده‌ها و از یادگیری تقویتی برای انتخاب کنش مناسب استفاده می‌شود. الگوریتم ارائه‌شده 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 ارائه‌شده بنا به دلیل ارائه کنشی جدید (باز آموزی نمونه‌های جدید به وسیله نمونه‌های مفید) زمان اجرا کاهش

یافته است و همچنین به دلیل در نظر گرفتن معیار زمان اجرا، برتـری بیشتری نسبت به سایر الگوریتم‌های افزایشی ذکر شده دارد.

جدول (۳): مقایسه صحت الگوریتم پیشنهادی Inc-RL-SGD با سایر الگوریتم‌های افزایشی (برحسب در صد)

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

جدول (۴): زمان اجرای الگوریتم پیشنهادی Inc-RL-SGD (برحسب ثانیه)

| **الگوریتم پیشنهادی** | |
|---|---|
| Datasets | Inc-RL-SGD |
| Letter | ۱۵۶/۳۶ |
| Susy | ۶۸۶/۳۴ |
| Mnist | ۷۹۶/۲۰ |
| Gisette | ۹۶/۵۳ |

جدول (۵): مقایسه صحت الگوریتم پیشنهادی Inc-RL-SGD با الگوریتم افزایشی Inc-SVM (برحسب در صد)

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

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

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

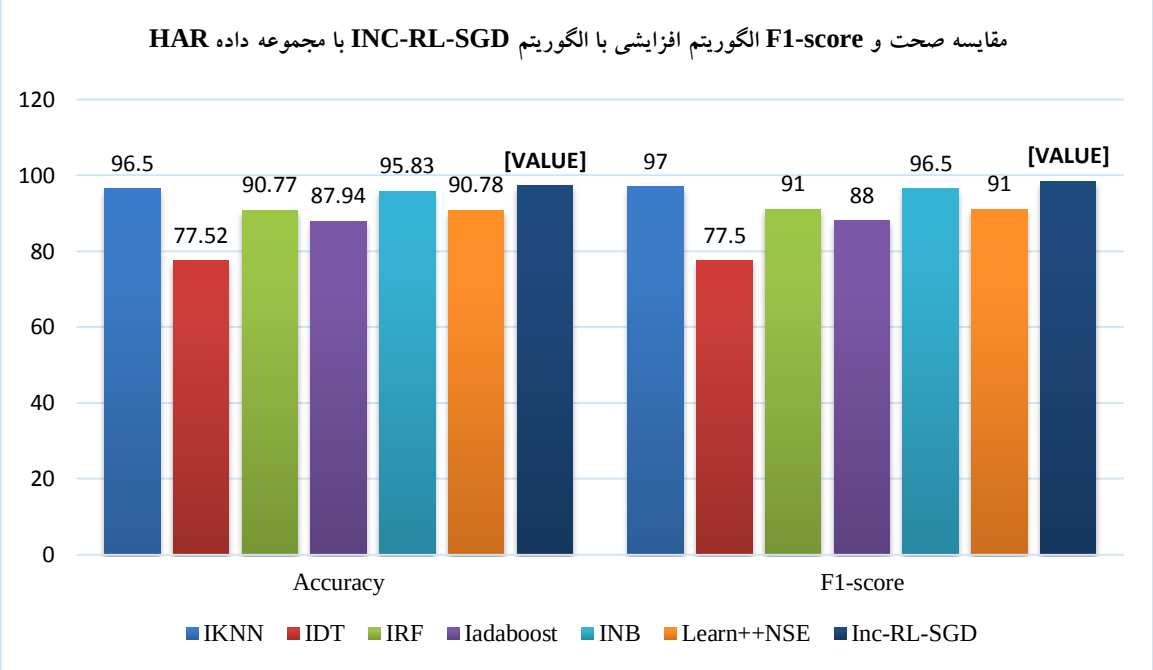
پذیرفته شده در مجله محاسبات نرم

جدول (۷): بررسی F1-score داده‌های مورد بررسی (برحسب در صد)

| Algorithm | Heart-C | Austral-ians | Ionosp-here | Votes | Letter | Susy | Mnist | Gisette | HAR |
|---|---|---|---|---|---|---|---|---|---|
| **INC-RL-SGD** | ۹۱/۱۵ | ۹۱/۳۵ | ۹۲/۳۹ | ۹۸/۰۱ | ۹۶/۸۵ | ۸۵/۷۳ | ۹۶/۹۹ | ۹۶/۵۹ | ۹۸/۵ |

جدول (۸): مقایسه F1-score الگوریتم پیشنهادی Inc-RL-SGD با الگوریتم افزایشی Inc-SVM (برحسب در صد)

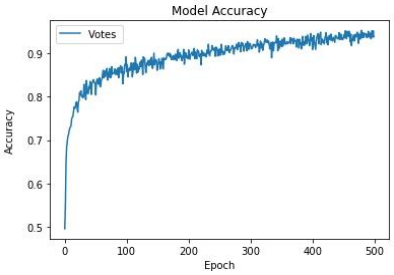
| Datasets | [۳۵]Inc-SVM | Inc-RL-SGD |
|---|---|---|
| Australians | ۹۰/۸۰ | **۹۱/۳۵** |
| Ionosphere | ۹۰/۷۵ | **۹۲/۳۹** |
| Heart-C | ۸۹/۹۰ | **۹۱/۱۵** |
| Votes | ۹۶/۸۵ | **۹۸/۰۱** |



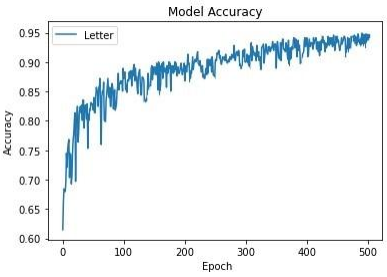

شکل (۳): مقایسه و بررسی صحت و F1-score الگوریتم پیشنهادی با الگوریتم افزایشی [۲۶]

جدول (۹): مقایسه زمان اجرای الگوریتم پیشنهادی Inc-RL-SGD با الگوریتم‌های افزایشی [۲۶]

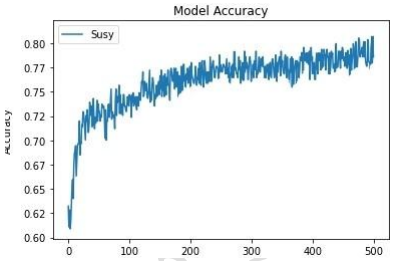
| Algorithms | Accuracy (برحسب درصد) | Run Time (برحسب ثانیه) |
|---|---|---|
| IKNN | ۹۶/۵۰ | ٤۲ |
| IDT | ۷۷/۵۲ | ۳۸ |
| IRF | ۹۰/۷۷ | ۱٦۷ |
| Iadaboost | ۸۷/۹۴ | ٤۷ |
| INB | ۹۵/۸۳ | ٤۰ |
| Learn++NSE | ۹۰/۷۸ | ۷٤ |




| | | |
|---|---|---|
| **Inc-RL-SGD** | ۹۷/۳۰ | ۳۳ |

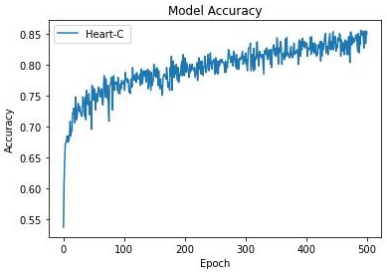



(پ) صحت داده Letter

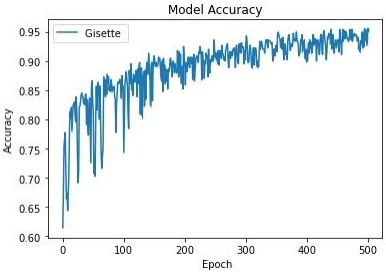



(ب) صحت داده Susy

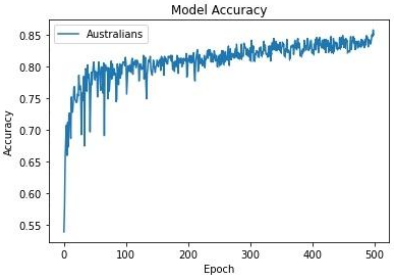
(الف) صحت داده Votes



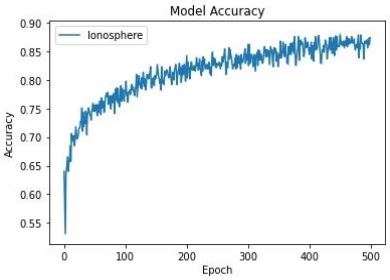

(ج) صحت داده Heart-C



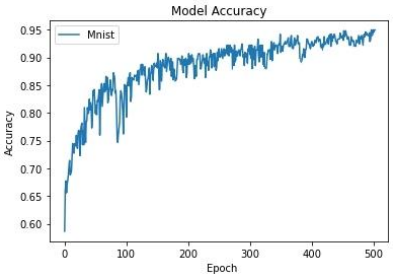

(ث) صحت داده Gisette




(ت) صحت داده Australians




(خ) صحت داده Ionosphere




(ح) صحت داده Mnist

شکل (۴): نمودار صحت داده‌ها

## مراجع


[۱] D. Bhattacharya, and M. Mitra, *Analytics on big fast data using real time stream data processing architecture*: EMC Corporation, 2013.

[۲] R. Polikar, L. Upda, S. S. Upda *et al.*, "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, vol. 31, no. 4, pp. 497-508, 2001.

[۳] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms." p. 116.

[٤] Q. Wang ,Y. Ma, K. Zhao *et al.*, "A comprehensive survey of loss functions in machine learning," *Annals of Data Science*, pp. 1-26, 2020.

[٥] V. Mnih, K. Kavukcuoglu, D. Silver *et al.*, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602* , ۲۰۱۳.

[٦] M. G. Lagoudakis, and R. Parr, "Reinforcement learning as classification: Leveraging modern classifiers." pp. 424-431.

[۷] S. Maliah, and G. Shani, "MDP-based cost sensitive classification using decision trees".

[۸] S. Chen, A. Choi, and A. Darwiche, "Value of information based on Decision Robustness".

[۹] L.-P. Liu, Y. Yu, Y. Jiang *et al.*, "Tefe: A time-efficient approach to feature extraction." pp. 423-432.

[۱۰] خسروی، عبدالمالکی، هادی، فیاضی، و همکاران.، "پیش‌بینی وضعیت تحصیلی متقاضیان پذیرش‌شده دانشگاه، مبتنی بر داده‌های آموزشی و پذیرشی با استفاده از تکنیک‌های داده کاوی"، محاسبات نرم، vol. 9 ،no. 2، pp. 94-113، ۲۰۲۱.

[۱۱] زارع، مهرجردی، فاطمه، یزدیان، دهکردی، مهدی، و همکاران.، "ارزیابی روش‌های یادگیری کلاسیک و یادگیری عمیق در تجزیه و تحلیل احساسات داده‌های تلگرام فارسی,"، محاسبات نرم، ۲۰۲۲.

[۱۲] ویسی، هادی، قایدشرف، و همکاران.، "بهبود کارایی الگوریتم‌های یادگیری ماشین در تشخیص بیماری‌های قلبی با بهینه‌سازی داده‌ها و ویژگی‌ها،" محاسبات نرم، vol. 8 ،no. 1، pp. 70-85، ۲۰۲۱.

[۱۳] 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.

[۱٤] L. Bottou, "Large-scale machine learning with stochastic gradient descent," *Proceedings of COMPSTAT'2010*, pp. 177-186: Springer, 2010.

[۱٥] P. Richtárik, and M. Takáč, "Parallel coordinate descent methods for big data optimization," *Mathematical Programming*, vol. 156, no. 1, pp. 433-484, 2016.

[۱٦] H. Zhang, "The optimality of naive Bayes," *Aa*, vol. 1, no. 2, pp. 3, 2004.


پذیرفته شده در مجله دانش نرم

[۱۷] C. Salperwyck, and V. Lemaire, "Learning with few examples: An empirical study on leading classifiers." pp. 1010-1019.

[۱۸] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes?." pp. 28-69.

[۱۹] S. Ting, W. Ip, and A .H. Tsang, "Is Naive Bayes a good classifier for document classification," *International Journal of Software Engineering and Its Applications*, vol. 5, no. 3, pp. 37-46, 2011.

[۲۰] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, pp. 1612, 1999.

[۲۱] A. Saffari, C. Leistner, J. Santner *et al.*, "On-line random forests." pp. 1393-1400.

[۲۲] V. Losing, B. Hammer, and H. Wersing, "Interactive online learning for obstacle classification on a mobile robot." pp. 1-8.

[۲۳] A. Sato, and K. Yamada, "Generalized learning vector quantization," *Advances in neural information processing systems*, vol. 8, 1995.

[۲۴] N. A. Syed, H. Liu, and K. K. Sung, "Incremental learning with support vector machines," 1999.

[۲۵] H. Zhao, H. Wang, Y. Fu *et al.*, "Memory-Efficient Class-Incremental Learning for Image Classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 5966-5977, 2021.

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

[۲۷] I. Lee, H. Roh, and W. Lee, "Encrypted malware traffic detection using incremental learning." pp. 1348-1349.

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

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

[۳۰] J. Huo, and T. L. van Zyl, "Incremental class learning using variational autoencoders with similarity learning," *Neural Computing and Applications*, pp. 1-16, 2023.

[۳۱] G .Shan, S. Xu, L. Yang *et al.*, "Learn#: a novel incremental learning method for text classification," *Expert Systems with Applications*, vol. 147, pp. 113198, 2020.

[۳۲] Á. C. Lemos Neto, R. A. Coelho, and C. L. d. Castro, "An incremental learning approach using long short-term memory neural networks," *Journal of Control, Automation and Electrical Systems*, vol. 33, no. 5, pp. 1457-1465, 2022.

[۳۳] B. Nemade, and D. Shah, "An efficient IoT based prediction system for classification of water using novel adaptive incremental learning framework," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 5121-5131, 2022.

[۳۴] M. Kang, J. Park, and B. Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation." pp. 16071-16080.

[۳۵] R. Khemchandani, Jayadeva, and S. Chandra, "Incremental twin support vector machines," *Modeling, Computation and Optimization*, pp. 263-272: World Scientific, 2009.