مقاله پژوهشی

یک الگوریتم انتخاب ویژگی برخط در جریان داده‌ها با استفاده از اطلاعات متقابل چندمتغیره

مریم رحمانی‌نیا و پرهام مرادی

چکیده: امروزه در بسیاری از مسایل دنیای واقعی همچون شبکه‌های اجتماعی، با جریان داده مواجه هستیم که در هر لحظه داده جدیدی به مجموعه داده‌های موجود اضافه می‌شود. از آنجا که کارایی بیشتر الگوریتم‌های داده‌کاوی با افزایش ابعاد داده‌ها کاهش می‌یابد، تحلیل این جریان داده‌ها در سال‌های اخیر به یکی از مسایل مهم در داده‌کاوی تبدیل شده است. روش‌های انتخاب ویژگی در جریان داده‌های برخط، روش‌های کارآمدی هستند که با حذف ویژگی‌های افزونه و نامربوط باعث کاهش ابعاد کلان داده‌ها و در نتیجه بهبود کارایی الگوریتم‌ها می‌شوند. از چالش‌های اساسی در رابطه با الگوریتم‌های انتخاب ویژگی برخط، در دسترس نبودن همه داده‌ها قبل از شروع الگوریتم، مقیاس‌پذیری، دقت ویژگی‌های انتخاب‌شده و اندازه زیرمجموعه انتخابی را می‌توان نام برد. تا کنون الگوریتم‌های انتخاب ویژگی موجود تنها توانسته‌اند بخش محدودی از این چالش‌ها را به صورت هم‌زمان مرتفع کنند. به همین منظور در این مقاله یک راهکار انتخاب ویژگی برخط به نام MMIOSFS با استفاده از اطلاعات متقابل ارائه داده‌ایم که حد واسط بهتری را میان چالش‌های ذکرشده به دست می‌آورد. در روش پیشنهادی در ابتدا مجموعه ویژگی‌ها با استفاده از تکنیک متغیرهای تصادفی توأم به یک ویژگی نگاشت و سپس اطلاعات متقابل ویژگی جدید با برچسب به عنوان میزان ارتباط مجموعه ویژگی‌های اولیه در نظر گرفته می‌شود. کارایی روش پیشنهادی با چند الگوریتم انتخاب ویژگی برخط با استفاده از دسته‌بندهای مختلف مورد ارزیابی قرار گرفته و نتایج به دست آمده نشان می‌دهد الگوریتم پیشنهادی معمولاً حد واسط بهتری میان چالش‌ها به دست می‌آورد.

کلیدواژه: **انتخاب ویژگی، داده‌های آموزشی برخط، اطلاعات متقابل، متغیر تصادفی مشترک.**

۱- مقدمه

در سال‌های اخیر، گسترش استفاده از شبکه‌های اجتماعی منجر به تولید داده‌های مقیاس بزرگ یا همان کلان داده[1] شده است [۱]. یکی از چالش‌هایی که در رابطه با کلان داده‌ها وجود دارد، مسأله ازدحام ابعاد[2] است که در آن تعداد ویژگی‌ها بسیار بیشتر از تعداد نمونه‌های آموزشی است. ازدحام ابعاد سبب می‌شود کارایی الگوریتم‌های داده‌کاوی و یادگیری ماشین به خصوص انواع دسته‌بندها کاهش زیادی داشته باشد [۲] و [۳]. انتخاب ویژگی یک راهکار برای حل مسأله ازدحام ابعاد است که که طی آن ویژگی‌های اضافی[3] و نامربوط[4] از مجموعه ویژگی‌های موجود حذف می‌شوند [۴] و [۵]. ویژگی اضافی، ویژگی‌ای است که اگر آن را از مجموعه ویژگی‌ها حذف کنیم، هنوز زیرمجموعه‌ای از ویژگی‌های دیگر وجود دارد که توانایی یکسانی با آن در پیشگویی برچسب‌ها دارد. به عبارت دیگر، ویژگی مشابه با این ویژگی در مجموعه داده‌ها وجود دارد. ویژگی نامربوط نیز به ویژگی اطلاق می‌شود که هیچ گونه اطلاعات باارزشی در مورد برچسب کلاس ارائه نمی‌دهد. تا کنون الگوریتم‌های انتخاب ویژگی متعددی به منظور فرایند انتخاب ویژگی ارائه شده است اما یک چالش اساسی در رابطه با این الگوریتم‌ها، برخط‌بودن داده‌ها با جریان ویژگی است. به این معنی که ویژگی‌ها با گذر زمان به مجموعه داده‌ای اضافه می‌شوند. در صورتی که در الگوریتم‌های انتخاب ویژگی سنتی فرض بر این است که قبل از شروع فرایند انتخاب ویژگی همه ویژگی‌ها در دسترس هستند [۶]. در بسیاری از برنامه‌های دنیای واقعی، همه ویژگی‌ها را نمی‌توان قبل از فرایند یادگیری به الگوریتم انتخاب ویژگی اعمال کرد. به عنوان مثال، در تجزیه و تحلیل تصاویر [۷]، توصیفگرهای متعددی به صورت پویا و با گذر زمان به منظور توصیف جنبه‌های مختلف تصاویر مانند نمودار گرادیان درونی[5]، نمودار رنگی[6] و انتقال مقیاس ویژگی‌های غیر قابل تغییر[7] اضافه می‌شوند. بنابراین انتظار برای تولید تمام ویژگی‌ها با استفاده از همه توصیفگرها خیلی زمان‌بر و حتی در بسیاری از موارد غیر واقعی است. به همین منظور اخیراً روش‌های انتخاب ویژگی برخط با جریان ویژگی ارائه شده که یکی از شاخه‌های انتخاب ویژگی برخط به حساب می‌آید [۸] و [۹]. در این دسته الگوریتم‌ها فرض بر این است که ویژگی‌ها به صورت مرحله‌ای و با مرور زمان به مجموعه داده‌ها اضافه می‌شوند. اما با وجود پیشرفت‌های زیادی که در این الگوریتم‌ها انجام شده است همچنان دارای چالش‌هایی چون نداشتن دانش اولیه در مورد داده‌ها قبل از شروع الگوریتم، زمان محاسباتی، مقیاس‌پذیری[8] و تعداد ویژگی‌های انتخابی هستند.

یک الگوریتم انتخاب ویژگی برخط مطلوب، الگوریتمی است که می‌تواند حد واسط بهتری میان چالش‌های مذکور به وجود آورد. اما الگوریتم‌های انتخاب ویژگی برخط با جریان ویژگی که تا کنون معرفی

این مقاله در تاریخ ۲۷ دی ماه ۱۳۹۸ دریافت و در تاریخ ۱۰ آذر ماه ۱۳۹۹ بازنگری شد.

مریم رحمانی‌نیا (نویسنده مسئول)، گروه مهندسی کامپیوتر، واحد قصرشیرین، دانشگاه آزاد اسلامی، قصرشیرین، ایران، (email: ma.rahmaninia@gmail.com).

پرهام مرادی، گروه مهندسی کامپیوتر، دانشگاه کردستان، سنندج، ایران، (email: p.moradi@uok.ac.ir).

1. Big Data
2. Curse of Dimensionality
3. Redundant
4. Irrelevant
5. Histogram of Oriented Gradients
6. Color Histogram
7. Scale Invariant Feature Transform
8. Scalability

شده‌اند هر کدام تنها سعی در مرتفع‌کردن یک یا دو چالش داشته‌اند [۱۰] تا [۱۴]. به عنوان مثال روش ارائه‌شده در [۱۱]، اگرچه معمولاً زیرمجموعه‌ای با حداقل اندازه از میان مجموعه ویژگی‌های اولیه انتخاب می‌کند، اما زمان محاسباتی بالایی دارد و مقیاس‌پذیر نیست. همچنین روش [۱۵] همواره ویژگی‌های باثباتی را تولید می‌کند، اما از آنجا که میزان افزونگی ویژگی‌ها را در نظر نمی‌گیرد معمولاً ویژگی‌های تولیدشده نسبت به یکدیگر اطلاعات اضافی دارند و در نتیجه کارایی و دقت الگوریتم را تحت تأثیر قرار می‌دهد. روش ارائه‌شده در [۱۰] از لحاظ زمان محاسباتی و مقیاس‌پذیری کارآمد است، اما چون وابستگی ویژگی‌ها را بدون در نظر گرفتن برچسب‌ها محاسبه می‌کند، معمولاً ویژگی‌های انتخاب‌شده از دقت کافی برخوردار نیستند. از طرف دیگر، همه این روش‌ها به مقادیر از پیش تعیین شده‌ای جهت رد یا پذیرش ویژگی‌های جدید نیاز دارند که وابسته به داده‌ها است [۱۰]، [۱۳] و [۱۵].

به دلایل مذکور، در این مقاله یک روش انتخاب ویژگی در جریان داده‌های برخط با جریان ویژگی به منظور انتخاب مجموعه‌ای از ویژگی‌های مرتبط و غیر افزونه با مرتفع‌کردن چالش‌های مربوط به راهکارهای پیشین ارائه شده است. در راهکار پیشنهادی که به نام MMIOSFS[1] نام‌گذاری شده است ویژگی‌ها به صورت مرحله‌ای و منفرد به مجموعه داده‌ها اضافه و از نظریه اطلاعات متقابل چندتایی[2] و راهکار تابع توزیع احتمالی توأم[3] استفاده شده است.

برای نشان‌دادن کارایی الگوریتم پیشنهادی، مجموعه‌ای از شبیه‌سازی‌ها بر روی دوازده مجموعه داده‌ای پرکاربرد انجام شده و نتایج به دست آمده از روش پیشنهادی با پنج الگوریتم انتخاب ویژگی با جریان داده برخط به نام SAOLA [۱۰]، OSFS [۱۴]، fast-OSF [۱۴]، Alpha-Investing [۱۵] و MMIOSFS [۱۶] مورد مقایسه قرار گرفته است. نتایج به دست آمده نشان می‌دهد الگوریتم پیشنهادی معمولاً حد واسط بهتری میان چالش‌های ذکرشده به وجود می‌آورد.

ساختار کلی این مقاله به این صورت است: در بخش دوم پیشینه تحقیق و مفاهیم مرتبط با نظریه اطلاعات و تکنیک تابع توزیع احتمالی توأم متغیرها شرح داده شده است. در بخش‌های سوم و چهارم الگوریتم پیشنهادی و نتایج آزمایشگاهی آورده شده و همچنین در بخش‌های پنجم و ششم بحث و نتجه‌گیری کلی ارائه گردیده است.

۲- پیشینه تحقیق و مروری بر کارهای قبلی

۲-۱ پیشینه تحقیق

تا کنون الگوریتم‌های انتخاب ویژگی مختلفی ارائه شده که بر حسب نحوه در دسترس بودن داده‌ها می‌توان آنها را در دو دسته اصلی قرار داد: ۱) الگوریتم‌های انتخاب ویژگی برون‌خط و ۲) الگوریتم‌های انتخاب ویژگی برخط. در الگوریتم‌های انتخاب ویژگی برون‌خطی فرض بر این است که در شروع الگوریتم، همه نمونه‌های آموزشی و ویژگی‌ها در دسترس است. این راهکارها را می‌توان در سه دسته فیلتر[4]، پوششی[5] و تعبیه‌شده[6] دسته‌بندی کرد. راهکارهای انتخاب ویژگی فیلتر، مناسب‌بودن یک ویژگی را بدون استفاده از انواع دسته‌بندها یا الگوریتم‌های یادگیری ماشین ارزیابی می‌کنند [۱۷] تا [۲۰]. روش‌های زیرفضای تصادفی[7] [۲۱]، [۲۱]، وابستگی متقابل[8] (MC) [۲۲]، روش انتخاب ویژگی افزونگی- ارتباط[9] [۲۳] و روش حداقل افزونگی- حداکثر ارتباط[10] [۶] از نوع روش‌های انتخاب ویژگی فیلتر هستند. همچنین در [۲۴] و [۲۵] دو روش انتخاب ویژگی فیلتر به ترتیب با استفاده از نظریه مجموعه‌های سخت[11] و تحلیل پوششی داده‌ها[12] ارائه شده است. در مقابل، راهکارهای پوششی از یک الگوریتم یادگیر یا دسته‌بند برای ارزیابی مناسب‌بودن یک زیرمجموعه ویژگی انتخاب‌شده استفاده می‌کنند [۲۶]. به همین دلیل دارای پیچیدگی محاسباتی بالایی هستند ولی معمولاً جواب‌هایی با دقت بالا تولید می‌کنند. به عنوان مثال در [۲۷] و [۲۸] دو الگوریتم انتخاب ویژگی پوششی که از الگوریتم تپه‌نوردی برای تشخیص خودکار گفتار استفاده کرده‌اند آورده شده است. در راهکارهای تعبیه‌شده، فرایند انتخاب ویژگی به عنوان بخشی از الگوریتم یادگیر در نظر گرفته می‌شود. به عبارت دیگر، جستجو برای زیرمجموعه مناسب از ویژگی‌ها به وسیله یک الگوریتم یادگیر انجام می‌شود [۲۹] و [۳۰].

از طرف دیگر روش‌های انتخاب ویژگی برخط روش‌های هستند که در آن داده‌های آموزشی به تدریج و با گذر زمان به مجموعه داده‌ها اضافه می‌شوند. در بعضی از این روش‌ها فرض بر این است که تعداد ویژگی‌ها ثابت و نمونه‌های آموزشی با گذر زمان به مجموعه آموزشی اضافه می‌شوند [۳۱] و [۳۲]. در بعضی دیگر از روش‌های انتخاب ویژگی برخط فرض شده که ویژگی‌ها به تدریج به مجموعه داده‌ای اضافه می‌شوند در حالی که تعداد نمونه‌های آموزشی ثابت در نظر گرفته شده است. در [۳۲] یک روش انتخاب ویژگی برخط ارائه شده که با استفاده از یک فرایند یادگیری افزایشی و دسته‌بند فازی، هر زمان که یک نمونه آموزشی جدید وارد می‌شود ویژگی‌ها را امتیازدهی می‌کند. در این روش به ویژگی‌هایی که دارای تأثیر کمتری در فرایند یادگیری هستند امتیاز کمتری تعلق می‌گیرد و از مجموعه داده‌های آموزشی حذف می‌شوند. در این مقاله فرض بر این است که تعداد نمونه‌های آموزشی ثابت و ویژگی‌ها با گذر زمان به داده‌های آموزشی اضافه می‌شوند. در ادامه تنها روش‌هایی بررسی می‌شود که تعداد نمونه‌های آموزشی در آنها ثابت و ویژگی‌ها با گذر زمان اضافه می‌شوند.

در [۱۳] یک روش انتخاب ویژگی برخط ارائه شده که توسط یک بردار وزنی w و با استفاده از تنظیم‌سازی نرم اول[13] مدل می‌شود. این روش که به نام گرفتین[14] معروف است، در هر مرحله، با اعمال یک الگوریتم ابتکاری مبتنی بر کاهش شیب خطا یک ویژگی را که به احتمال زیاد منجر به بهبود مدل موجود می‌شود انتخاب می‌کند. در [۱۵] یک الگوریتم انتخاب ویژگی برخط با استفاده از روش تطبیقی خطا به نام Alpha-Investing معرفی شده که به صورت پویا آستانه خطا را که مورد نیاز برای پذیرش یک ویژگی جدید است تغییر می‌دهد. در [۱۴] دو الگوریتم انتخاب ویژگی برخط دیگر به نام OSFS و fast-OSFS ارائه شده که با

1. Multivariate Mutual Information Online Stream Feature Selection
2. Multivariate Mutual Information
3. Joint Probability Function
4. Filter
5. Wrapper
6. Embedded
7. Random Space Methods
8. Mutual Correlation
9. Robust and Relevant Feature Selection
10. Maximal Relevancy-Minimal Redundancy
11. Rough Set Theory
12. Data Envelopment Analysis
13. l_1-Norm Regularization
14. Grafting

استفاده از زنجیره مارکوف و نظریه احتمالات به انتخاب ویژگی در داده‌های با جریان ویژگی پرداخته‌اند. این روش‌ها در دو مرحله ۱) تجزیه و تحلیل برخط ارتباط و ۲) تجزیه و تحلیل برخط افزونگی، ویژگی‌های مهم را شناسایی می‌کند. در [۱۰] یک الگوریتم انتخاب ویژگی برخط به نام SAOLA ارائه شده که از معیار اطلاعات متقابل جهت محاسبه ارتباط و افزونگی بین ویژگی‌ها استفاده می‌کند. اخیراً در [۱۶]، یک روش انتخاب ویژگی برخط با جریان ویژگی با استفاده از اطلاعات متقابل شرطی ارائه شده که به صورت کارآمد به شناسایی و حذف ویژگی‌های مهم و افزونه می‌پردازد. در [۳۳] یک الگوریتم انتخاب ویژگی برخط به نام OS-NRRSAR-SA بر مبنای نظریه مجموعه‌های سخت ارائه شده است. در [۳۴] یک روش انتخاب ویژگی برخط با جریان ویژگی چندبرچسبی[1] به نام ML-NRS ارائه شده که با استفاده از نظریه همسایگی مجموعه‌های سخت[2] به انتخاب ویژگی‌های مرتبط و غیر افزونه می‌پردازد. در [۳۵] نیز با استفاده از نظریه همسایگی اطلاعات متقابل[3]، یک الگوریتم به نام MI-OSFS به انتخاب ویژگی‌های مهم در داده‌های آموزشی برخط با جریان ویژگی چندبرچسبی می‌پردازد.

علاوه بر این روش‌های انتخاب ویژگی برخط متعددی نیز ارائه شده که در آن ویژگی‌ها به صورت گروهی به مجموعه داده‌ها اضافه می‌شوند. به عبارت دیگر در این دسته الگوریتم‌ها ویژگی‌ها به صورت گروهی و نه منفرد به مجموعه داده‌ها اضافه می‌شوند. از جمله این روش‌ها می‌توان به GFSSF[4] در [۳۶] اشاره کرد که با استفاده از اطلاعات متقابل شرطی قادر است هم در سطح انفرادی و هم گروهی به انتخاب ویژگی‌ها بپردازد. در [۳۷] نیز یک الگوریتم انتخاب ویژگی برخط گروهی با جریان ویژگی به نام OGFS[5] ارائه شده که با استفاده از راهکار نظریه گراف طیفی به انتخاب ویژگی‌ها می‌پردازد. این الگوریتم که یک الگوریتم کارآمد است از اطلاعات اولیه گروه‌ها استفاده می‌کند و دارای ۲ فاز انتخاب درون گروهی و بین گروهی برخط است. الگوریتم دیگر به نام Group-SAOLA که گسترش‌یافته الگوریتم SAOLA است می‌تواند به فرایند انتخاب ویژگی هم در سطح فردی و هم گروهی بپردازد [۱۰]. در این روش در ابتدا، در هر گروه با استفاده از اطلاعات متقابل دوتایی، ویژگی‌های افزونه شناسایی و حذف می‌شوند و سپس الگوریتم به تجزیه و تحلیل افزونگی میان ویژگی‌های بین گروه‌ها می‌پردازد. همچنین روش OMGFS در [۳۸] با استفاده از نظریه همسایگی اطلاعات متقابل در یک ساختار گروهی به انتخاب ویژگی‌ها در داده‌های آموزشی چندبرچسبی پرداخته است. این الگوریتم نیز دارای دو فاز برون گروهی و درون گروهی است. فاز برون گروهی مربوط به انتخاب گروه‌های مؤثر و فاز درون گروه به انتخاب ویژگی‌های مهم در هر گروه می‌پردازد. در روش پیشنهادی فرض شده که ویژگی‌ها به صورت منفرد و یکی‌یکی به مجموعه داده‌ها اضافه می‌شوند.

۲-۲ نظریه اطلاعات و تکنیک تابع توزیع احتمالی

گاهی اوقات وابستگی میان دو یا بیشتر از دو متغیر مد نظر است. به عنوان مثال ممکن است شخصی بخواهد ارتباط میان میزان هوش و وزن کودکان را اندازه بگیرد. در این مواقع متغیرهای تصادفی با یکدیگر در نظر گرفته می‌شوند و دارای یک توزیع مشترک خواهند بود که به ما اجازه می‌دهند احتمال رخداد هم‌زمان و میزان وابستگی آنها را به دست آوریم.

تعریف ۱) متغیرهای تصادفی توأم[6] (JRVs): فرض کنید X و Y دو دو متغیر تصادفی باشند که هر کدام می‌توانند مقادیر $\{x_1, x_2, \ldots, x_n\}$ و $\{y_1, y_2, \ldots, y_n\}$ را اتخاذ کنند. حال متغیر تصادفی توأم X و Y عبارت است

$$JRVs(X,Y) = \{(x_i, y_j) \\ \text{s.t. } x_i \in X, y_j \in Y \ , \ \forall i \le n \ , \ j \le m\} \quad (۱)$$

به طور کلی متغیر تصادفی توأم میان مجموعه متغیرهای $S = \{X_1, X_2, \ldots, X_n\}$ عبارت است از

$$JRVs(S) = \{(x_{i_1}, x_{i_2}, \ldots, x_{i_N}) \\ \text{s.t. } x_{i_1} \in X_1, x_{i_2} \in X_2, \ldots, x_{i_N} \in X_N\} \quad (۲)$$

مثال ۱: اگر مجموعه مقادیر ممکن برای دو متغیر X و Y به ترتیب $X = \begin{bmatrix} ۱ \\ ۰ \\ ۲ \end{bmatrix}$ و $Y = \begin{bmatrix} ۲ \\ ۱ \\ ۱ \end{bmatrix}$ باشد، آن گاه متغیرهای تصادفی توأم حاصل از X و Y که با Z نشان داده می‌شود عبارت است از

$$Z = [(۱,۲);(۱,۱);(۱,۱);(۱,۲);(۰,۱); \\ (۰,۱);(۲,۲);(۲,۱);(۲,۱)] = [۱;۲;۲;۱;۳;۴;۴;۵;۶;۶] \quad (۳)$$

در نظریه اطلاعات، اطلاعات متقابل بین دو متغیر تصادفی، معیاری برای نشان‌دادن میزان وابستگی متقابل آن دو متغیر است [۳۹]. در حقیقت این معیار، میزان اطلاعات به دست آمده در مورد یک متغیر از طریق متغیر دیگر را نشان می‌دهد. مفهوم اطلاعات متقابل ذاتاً با مفهوم آنتروپی[7] مرتبط است که میزان اطلاعات موجود در یک متغیر تصادفی یا به عبارتی دیگر عدم قطعیت یا پراکندگی متغیر را نشان می‌دهد. اگر $\{x_1, x_2, \ldots, x_n\}$ به مجموعه مقادیر ممکن برای متغیر تصادفی X اشاره کند، آن گاه آنتروپی متغیر X عبارت است از

$$H(X) = -\sum_{x_i} p(x_i) \log p(x_i) \big| x_i \in X \quad (۴)$$

به طوری که $p(x_i)$ احتمال رخداد مقدار x از مجموعه مقادیر ممکن برای متغیر X را نشان می‌دهد. هنگامی که X یک متغیر تصادفی گسسته است $p(x_i)$ را می‌توان از شمارش رخدادهای x مانند $p(x_i) = \#x_i / N$ به دست آورد. به عبارت دیگر به معنای کسری از مشاهدات روی مقدار x از کل نمونه‌های N است. حال اگر تعداد نمونه‌های آموزشی و ویژگی‌ها زیاد باشد یا متغیر X، یک متغیر با مقادیر پیوسته باشد به منظور محاسبه آن می‌توان از تخمینگر هیستوگرام با اندازه عرض ثابت استفاده کرد [۴۰].

مفهوم دیگر، آنتروپی توأم است که به آنتروپی یا عدم قطعیت مجموعه‌ای از متغیرها (ویژگی‌ها) اشاره می‌کند که می‌توان آن را به صورت زیر تعریف کرد

$$H(X,Y) = \\ \sum_{x_i \in X} \sum_{y_i \in Y} p(x_i, y_i) \log p(x_i, y_i) \big| x_i \in X, y_i \in Y \quad (۵)$$

1. Multi Label
2. Neighborhood Rough Set Theory
3. Neighborhood Mutual Information
4. Group Feature Selection with Streaming Feature
5. Online Group Feature Selection
6. Joint Random Variables
7. Entropy

به طوری که $p(x_i,y_i)$ به تابع توزیع احتمال توأم دو متغیر X و Y اشاره می‌کند.

آنتروپی شرطی نیز میزان عدم قطعیت یک متغیر را با فرض مشخص‌بودن متغیر دیگر نشان می‌دهد که همانند زیر می‌توان آن را تعریف کرد

$$H(Y|X)=H(X,Y)-H(X)= -\sum_{x_i\in X}\sum_{y_j\in Y}p(x_i,y_j)\log p(y_j|x_i) \quad (۶)$$

به طوری که $p(y_j|x_i)$ به احتمال y_j به شرط x_i اشاره می‌کند که از رابطه زیر محاسبه می‌شود

$$p(y_j|x_i)=\frac{p(x_i,y_j)}{p(x_i)} \quad (۷)$$

در نظریه اطلاعات، اطلاعات متقابل بین دو متغیر به مقداری از اطلاعات که بین دو متغیر به اشتراک گذاشته شده است، اشاره می‌کند که عبارت است از

$$I(X;Y)=H(X,Y)-H(X)-H(Y) \quad (۸)$$

مقدار (۸) معمولاً مثبت است. حال اگر اطلاعات متقابل میان دو ویژگی زیاد باشد به این معنی است که وابستگی بین دو متغیر نیز زیاد است.

اطلاعات متقابل شرطی میان دو متغیر X و Y به شرط وجود متغیر Z از طریق رابطه زیر به دست می‌آید

$$I(X;Y|Z)=I(X:Y)-I(X;Y;Z)= \sum_{x_i\in X}\sum_{y_j\in Y}\sum_{z_k\in Z}p(x_i,y_j,z_k)\log\frac{p(z_k)p(x_i,y_j,z_k)}{p(x_i|z_k)p(y_j|z_k)} \quad (۹)$$

بر اساس [۴۱] اطلاعات متقابل میان زیرمجموعه‌ای از ویژگی‌ها مانند $S=\{X_1,X_2,\dots,X_n\}$ و برچسب C عبارت است از

$$I(S;C)=H(S)-H(S,C)+H(C)= H(\{X_1,X_2,\dots,X_n\})- H(\{X_1,X_2,\dots,X_n,C\})+H(C) \quad (۱۰)$$

به طوری که هر کدام از معیارهای $H(.)$ به آنتروپی مشترک میان مجموعه‌ای از متغیرها اشاره می‌کند که طبق (۵) به صورت زیر به دست می‌آید

$$H(\{X_1,X_2,\dots,X_n\})= -\sum p(\{X_1,X_2,\dots,X_n\}).\log p(\{X_1,X_2,\dots,X_n\}) \quad (۱۱)$$

همچنین $p(\{X_1,X_2,\dots,X_n\})$ طبق تعریف ۱ به تابع جرم احتمالی توأم به ازای مجموعه متغیرهای $\{X_1,X_2,\dots,X_n\}$ اشاره می‌کند.

به طور کلی با استفاده از [۳۹] می‌توان اطلاعات متقابل چندگانه را به صورت حاصل‌جمع آنتروپی‌ها همانند زیر نوشت

$$I(X_1;X_2;\dots;X_n)=H(X_1)+H(X_2)+\dots+H(X_n) -H(X_1,X_2)-\dots-H(X_{n-1},X_n)+\dots+ (-1)^nH(X_1,\dots,X_n)=-\sum_{X_i\subseteq T'}(-1)^{|X_i|}H(X_i) \quad (۱۲)$$

به طوری که $\sum_{X_i\subseteq T}$ به مجموع تمام زیرمجموعه‌های مشتق‌شده از T' و T' به مجموعه توانی اعضای T اشاره می‌کند. به عنوان مثال فرض کنید $T=\{X_1,X_2\}$، پس مجموعه توانی آن عبارت است از $T'=\{\{\},X_1,X_2,\dots,\{X_1,X_2\}\}$.

۳- الگوریتم پیشنهادی

در این بخش به ارائه جزئیات الگوریتم پیشنهادی با نام MMIOSFS که در آن ویژگی‌ها به تدریج و با مرور زمان به صورت یکی‌یکی به مجموعه داده‌ها اضافه می‌شوند می‌پردازیم. فرض کنید $D(F_t,C)$ یک جریان داده‌ای را نشان می‌دهد که $F_t\subseteq\{f_1,f_2,\dots,f_t\}$ به دنباله‌ای از ویژگی‌های دیده‌شده تا لحظه t و C به برچسب کلاس اشاره می‌کند. همچنین متغیرهای S_t و f_t به ترتیب به مجموعه ویژگی‌های انتخاب‌شده تا لحظه t و ویژگی تازه‌وارد در لحظه t اشاره می‌کند. در این الگوریتم هدف اصلی انتخاب زیرمجموعه $S_t\subseteq\{S_{t-1}\cup f_t\}$ با کمترین اندازه ممکن است به طوری که ارتباط بین ویژگی‌های انتخاب‌شده با برچسب کلاس بیشینه و افزونگی میان ویژگی‌های انتخاب‌شده کمینه باشد. به طور کلی می‌توان گفت، هدف اصلی عبارت است از

$$S_t=\arg\min_{S'_t}\{|S'_t|:S'_t=\arg\min_{X\subseteq\{S_{t-1}\cup f_t\}}H(C|X)\} \quad (۱۳)$$

به عبارت دیگر، هدف اصلی انتخاب زیرمجموعه‌ای از میان ویژگی‌های دیده‌شده با کمترین تعداد است به طوری که آنتروپی یا عدم قطعیت برچسب کلاس C که همانند زیر تعریف می‌شود کمینه شود

$$H(C|S_t)=H(C)-I(C;S_t) \quad (۱۴)$$

از آنجا که جستجوی همه زیرمجموعه‌های ممکن از S_t به منظور کمینه‌سازی تابع (۱۰) از مرتبه نمایی $O(2^{|S_t|})$ است، پس پیداکردن آن سخت و زمان‌بر می‌باشد. به همین منظور، در این مقاله سعی شده که یک روش تقریبی ارائه شود. در روش ارائه‌شده برای اضافه (حذف) کردن یک ویژگی به (از) مجموعه ویژگی‌های انتخابی از میزان ارتباط یک ویژگی با برچسب کلاس استفاده شده است. در این صورت، این تضمین وجود دارد که اضافه‌کردن یک ویژگی باعث کاهش عدم قطعیت یا آنتروپی برچسب کلاس می‌شود.

همچنین به منظور کمینه‌کردن تعداد ویژگی‌های انتخابی در هر مرحله از اجرای الگوریتم تعدادی از ویژگی‌ها که با اضافه‌شدن ویژگی جدید هیچ گونه تأثیری در کاهش آنتروپی برچسب کلاس ندارند از مجموعه ویژگی‌های انتخابی حذف می‌شود. به طور کلی می‌توان گفت MMIOSFS شامل دو فاز اصلی است که در ادامه هر کدام از این فازها به صورت مفصل توضیح داده شده است.

فاز اول، ویژگی مرتبط: هدف از این فاز محاسبه میزان ارتباط و وابستگی ویژگی تازه‌وارد f_t با برچسب کلاس C است. میزان وابستگی یک ویژگی با برچسب کلاس از طریق رابطه زیر انجام می‌شود

$$\gamma_{f_t}=I(C;f_t) \quad (۱۵)$$

همان طور که قبلاً گفته شد، I به اطلاعات متقابل میان دو ویژگی f_t و C اشاره می‌کند که با استفاده از (۵) قابل محاسبه است. حال اگر γ_{f_t} یک مقدار بزرگ‌تر از صفر داشته باشد به این معنی است که ویژگی تازه‌وارد دارای اطلاعات مؤثر در مورد برچسب کلاس است. در نتیجه باعث کاهش عدم قطعیت (فرمول (۱۴)) برچسب کلاس می‌شود و می‌توان آن را به مجموعه ویژگی‌های انتخاب‌شده اضافه کرد و بعد از آن الگوریتم وارد فاز دوم می‌شود. در غیر این صورت اگر مقدار γ_{f_t} کمتر از صفر باشد ویژگی مربوط حذف می‌شود و الگوریتم منتظر ورود ویژگی جدید می‌ماند.

MMIOSFS: Multivariate Mutual Information Online Stream Feature Selection

Input: f_t, the new arrival feature f at time t,

Output: S_t, the selected feature subset till time t, $S_0 = \{\}$,

1. **Repeat**
2. $f_t \leftarrow$ new arrival feature at time t

//**Step 1:** Checking for relevance of new arrival feature f_t

3. **Compute** relevancy γ_{f_t} using Eq. (15)
4. **If** $\gamma_{f_t} > 0$ **then** $S_t \leftarrow S_{t-1} \cup f_t$ **Else** go to line 16 **End**

//**Step 2:** Checking for the redundant features in S_t

5. $S'_t \leftarrow S_t$
6. **For** each selected feature f in S_t
7. $B \leftarrow S'_t \setminus \{f\}$
8. $V \leftarrow JVRs(B)$ //Based on Def. 1
9. $V' \leftarrow JVRs(S'_t)$ //Based on Def. 1
10. $E(f) \leftarrow I(V';C) - I(V;C)$ //Based on Eq. (16)
11. **If** $E(f) < 0$
12. $S'_t \leftarrow b$
13. **End if**
14. **End for**
15. $S_t \leftarrow S'_t$
16. **End repeat**

شکل ۱: شبه‌کد الگوریتم پیشنهادی MMIOSFS.

جدول ۱: مشخصات داده‌های آموزشی.

نام داده آموزشی	تعداد نمونه‌ها	تعداد ویژگی‌ها
Breast-cancer	۲۸۶	۹
Breast-w	۶۹۹	۹
Car	۱۷۲۸	۶
Credit-a	۳۱۹	۱۵
Credit-g	۱۰۰۰۰	۲۰
Glass	۲۱۴	۹
Hypothiroi	۳۷۷۲	۲۲
Ionsphere	۳۵۱	۳۴
Letter	۲۰۰۰۰	۱۶
Mushroom	۸۱۲۴	۲۲
Sonar	۲۰۸	۶۰
Soybean	۶۸۳	۳۵

فاز دوم، حذف ویژگی‌های اضافی: در این فاز به منظور کاهش تعداد ویژگی‌های انتخاب‌شده تا لحظه t، ویژگی‌های انتخاب‌شده‌ای که هیچ گونه تأثیر در کاهش عدم قطعیت برچسب کلاس C ندارند حذف می‌شوند. به همین منظور، میزان تأثیر حذف‌شدن هر کدام از ویژگی‌های انتخاب‌شده از مجموعه ویژگی‌های قبلاً انتخاب‌شده S_t محاسبه می‌شود. میزان تأثیر حذف‌شدن یک ویژگی به نام f در کاهش آنتروپی برچسب کلاس را می‌توان با محاسبه تفاضل اطلاعات متقابل میان برچسب کلاس C و مجموعه ویژگی‌های انتخابی S_t یک بار با در نظر گرفتن ویژگی f و یک بار بدون در نظر گرفتن آن در S_t همانند زیر محاسبه می‌کنیم

$$E(f) = I(S_t;C) - I(\{S_t \setminus f\};C) \quad \forall f \in S_t \qquad (۱۶)$$

به طوری که I به اطلاعات متقابل میان مجموعه‌ای از متغیرها (شامل ویژگی‌ها و برچسب) اشاره می‌کند که طبق (۱۲) می‌توان آن را به دست آورد. اما از آنجا که محاسبه آن مستلزم محاسبه اطلاعات متقابل میان تک‌تک زیرمجموعه‌ها است، پس محاسبه آن از مرتبه نمایی و هزینه‌بر است. به همین منظور در این مقاله در ابتدا، متغیرهای تصادفی مشترک (JRVs) حاصل از دو مجموعه $\{S_t\}$ و $\{S_t \setminus f\}$ را محاسبه کرده و سپس تفاضل اطلاعات متقابل میان این متغیرهای جدید و برچسب کلاس C محاسبه می‌شود. متغیر تصادفی مشترک میان یک مجموعه از متغیرها (ویژگی‌ها) را با استفاده از تعریف ۱ و همانند مثال ۱ می‌توان به دست آورد. شبه‌کد روش پیشنهادی در الگوریتم شکل ۱ نشان داده شده است.

طبق این الگوریتم، در ابتدا به ازای هر ویژگی تازه‌وارد f_t میزان ارتباط آن با برچسب کلاس طبق (۱۵) محاسبه می‌شود (خط ۴). اگر ویژگی دارای ارتباط بود، آن گاه به مجموعه ویژگی‌های انتخاب‌شده اضافه و الگوریتم وارد فاز دوم (خط ۵) می‌شود. در غیر این صورت ویژگی تازه‌وارد حذف شده و الگوریتم منتظر ورود ویژگی دیگر می‌ماند.

در فاز دوم که دستورات آن در خطوط ۵ تا ۱۴ شکل ۱ آورده شده است، تأثیر حذف‌شدن هر ویژگی f_t از ویژگی‌های موجود در مجموعه S_t در کاهش عدم قطعیت برچسب کلاس با استفاده از (۱۶) محاسبه می‌شود و ویژگی‌ای که تأثیر آن صفر است از مجموعه (S_t) حذف می‌شود. همان طور که قبلاً اشاره شد به منظور محاسبه (۱۶)، ابتدا متغیر مشترک دو مجموعه $\{S_t\}$ و $\{S_t \setminus f_t\}$ طبق خطوط ۸ و ۹ در شکل ۱ و سپس اطلاعات متقابل میان متغیرهای به دست آمده و ویژگی f محاسبه می‌شود. این مرحله به تعداد ویژگی‌های موجود در مجموعه S_t تکرار می‌شود.

٤- نتایج آزمایشگاهی

در این بخش الگوریتم پیشنهادی با پنج الگوریتم انتخاب ویژگی برخط به نام SAOLA [۱۰]، OSFS [۱۴]، fast-OSFS [۱۴]، Alpha-Investing [۱۵] و OSFSMI [۱۶] مقایسه شده است. در تمام این الگوریتم‌ها فرض بر این است که ویژگی‌ها به تدریج به مجموعه داده‌ای اضافه می‌شوند. پس به منظور شبیه‌سازی این سناریو، ویژگی‌ها یکی‌یکی به مجموعه داده‌ای اضافه شده و در لحظه شروع فرض بر این است که هیچ ویژگی‌ای وجود ندارد.

همچنین در آزمایش‌ها، از ۱۲ مجموعه داده آموزشی در حوزه‌های مختلف هوش مصنوعی استفاده گردیده که جزئیات هر کدام از آنها در جدول ۱ آورده شده است[1]. معیارهای مقایسه کارایی الگوریتم پیشنهادی متوسط دقت به دست آمده از دسته‌بندهای KNN [۴۲] و DT [۴۳]، تعداد ویژگی‌های انتخاب‌شده، زمان اجرا و مقیاس‌پذیری است. منظور از دقت دسته‌بندی، تعداد نمونه‌هایی است که برچسب آنها با استفاده از دسته‌بندهای KNN و DT به درستی حدس زده شده است. در واقع این معیار نشان می‌دهد که دسته‌بند چند درصد از کل مجموعه نمونه‌های آزمایشی را به درستی دسته‌بندی کرده است.

همچنین در آزمایش‌ها از آزمون فریدمن [۴۴] به منظور مقایسه و تحلیل آماری نتایج به دست آمده استفاده شده است. بر اساس این آزمون هرچه رتبه یک الگوریتم به ازای همه داده‌ها و الگوریتم‌ها کمتر باشد، مطلوبیت الگوریتم بیشتر است. در این آزمایش‌ها با استفاده از نرم‌افزار Matlab و کدهای موجود در ابزارهای انتخاب ویژگی چون LOFS [۴۵] و KOFS به منظور پیاده‌سازی روش پیشنهادی و سایر الگوریتم‌های انتخاب ویژگی استفاده شده است.

1. https://archive.ics.uci.edu/ml/datasets.php

جدول ۲: متوسط دقت دسته‌بندی درخت تصمیم الگوریتم‌های انتخاب ویژگی برخط.

	MMIOSFS	SAOLA	OSFS	Fast-OSFS	Alpha-Investing	OSFSMI
Breast-cancer	۶۶٫۶	۷۱٫۰	۷۵٫۴	۷۲٫۸	۷۲٫۸	۷۴٫۵
Breast-w	۹۳٫۹	۹۳٫۹	۹۳٫۹	۹۳٫۹	۹۴٫۶	۹۴٫۹
Car	۹۳٫۹	۹۳٫۹	۹۳٫۹	۹۳٫۹	۸۷٫۱	۷۶٫۶
Credit-a	۸۳٫۷	۸۲٫۱	۸۱٫۸	۸۲٫۹	۸۶٫۱	۸۵٫۰
Credit-g	۶۹٫۰	۶۳٫۲	۶۹٫۵	۶۹٫۵	۷۶٫۷	۶۹٫۰
Glass	۷۲٫۲	۵۴٫۲	۶۳٫۸	۶۳٫۸	۶۶٫۲	۶۳٫۸
Hypothiroi	۹۷٫۲	۹۵٫۹	۹۶٫۷	۹۶٫۷	۹۶٫۶	۹۷٫۱
Ionsphere	۹۰٫۵۷	۸۸٫۵۷	۸۲٫۸۵	۸۲٫۸۵	۸۶٫۴	۸۲٫۲
Letter	۸۳٫۲۹	۸۴٫۹	۷۶٫۹۸	۷۶٫۹۸	۸۱٫۹۵	۸۲٫۱
Mushroom	۱۰۰	۹۹٫۱۶	۹۹٫۱۶	۹۹٫۸۱	۱۰۰	۹۸٫۶
Sonar	۶۲٫۱۹	۶۰٫۹۷	۵۹٫۷۵	۵۹٫۷۵	۶۸٫۲۹	۷۵٫۶
Soybean	۸۵٫۵۰	۷۷٫۳۲	۶۲٫۰۸	۶۱٫۳۳	۸۱٫۰۴	۶۵٫۹
Vote	۹۳٫۶۴	۹۲٫۴۸	۹۲٫۴۸	۹۲٫۴۸	۹۲٫۴۸	۹۴٫۷
Mean	۸۳٫۹۹	۸۱٫۳۷	۸۰٫۶۹	۸۰٫۵۲	۸۳٫۱۹	۸۱٫۵۸
Fridman-rank	۲٫۴۶	۴٫۱۹	۳٫۶۹	۴٫۰۷	۳٫۱۱	۳٫۱۹

جدول ۳: متوسط دقت دسته‌بندی $k-$ نزدیک‌ترین همسایه به ازای الگوریتم‌های انتخاب ویژگی برخط مختلف.

	MMIOSFS	SAOLA	OSFS	Fast-OSFS	Alpha-Investing	OSFSMI
Breast-cancer	۶۵٫۷	۵۳٫۵	۴۸٫۲	۴۵٫۶	۵۵٫۲	۱٫۵۹
Breast-w	۹۳٫۹	۹۴٫۲	۹۵٫۶	۹۵٫۶	۹۵٫۳	۹۲٫۱
Car	۹۱٫۰	۹۱٫۰	۹۱٫۰	۹۱٫۰	۸۱٫۷	۷۰٫۱
Credit-a	۶۶٫۶	۵۵٫۲	۵۵٫۶	۵۴٫۵	۶۹٫۴	۶۸٫۰
Credit-g	۵۶٫۵	۶۲٫۲	۶۴٫۷	۶۴٫۷	۶۷٫۰	۶۷٫۰
Glass	۷۷٫۱	۶۰٫۲	۶۸٫۶	۶۸٫۶	۶۹٫۸	۶۱٫۴
Hypothiroi	۹۵٫۵	۹۵٫۲	۹۶٫۰	۹۶٫۰	۹۵٫۶	۹۶٫۱
Ionsphere	۸۶٫۲۸	۸۲٫۱۴	۸۷٫۱۴	۸۷٫۱۴	۸۹٫۲۸	۸۵٫۷
Letter	۹۲٫۵۵	۹۵٫۹۵	۸۱٫۹۳	۸۱٫۹۳	۹۳٫۲۶	۹۱٫۹
Mushroom	۱۰۰	۹۹٫۱۶	۹۹٫۸۱	۹۹٫۸۱	۱۰۰	۹۸٫۲
Sonar	۷۵٫۶	۷۶٫۱۲	۶۸٫۲۹	۶۸٫۲۹	۶۸٫۲۹	۶۹٫۵
Soybean	۸۵٫۵	۷۳٫۹۷	۵۴٫۲۷	۵۴٫۲۷	۸۱٫۴۱	۸۳
Vote	۹۲٫۴۸	۹۱٫۳۲	۹۲٫۴۸	۹۲٫۴۸	۹۲٫۴۸	۹۱٫۶
Mean	۸۳٫۸۴	۷۹٫۳۱	۷۷٫۲۲	۷۶٫۹۳	۸۱٫۴۶	۷۹٫۵۱
Fridman-rank	۲٫۵	۴٫۳۴	۳٫۶۹	۳٫۹۲	۲٫۷۳	۳٫۸۰

علاوه بر این به منظور محاسبه اطلاعات متقابل در داده‌های آموزشی با مقادیر پیوسته، ابتدا داده‌ها به صورت گسسته تبدیل شده‌اند. همه نتایج به دست آمده با استفاده از نرم‌افزار Matlab و روی سیستمی با حافظه ۸ گیگابایت و پردازنده ۲٫۴ گیگاهرتز آزمایش گردیده و همچنین در تمام آزمایش‌ها از روش اعتبارسنجی $k-$فولد[1] با اندازه $k=5$ استفاده شده است. به این صورت که داده‌های آموزشی در ۵ قسمت مساوی تقسیم می‌شود که یکی از این ۵ قسمت برای آزمایش و ۴ قسمت باقیمانده برای آموزش در نظر گرفته شده است و متوسط نتایج به دست آمده روی مجموعه‌های آموزشی و آزمایشی به عنوان نتیجه نهایی در نظر گرفته شده است.

٤-١ دقت دسته‌بندی

در این بخش، دقت دسته‌بندی الگوریتم‌های انتخاب ویژگی مختلف، به ازای دسته‌بندهای $k-$نزدیک‌ترین همسایه (KNN) و درخت تصمیم (DT) در جدول‌های ۲ و ۳ نشان داده شده است. این نتایج، متوسط دقت دسته‌بندها روی پنج ترتیب مختلف از ورود ویژگی‌ها روی همه مجموعه‌های داده‌ای را نشان می‌دهد. در همه این جدول‌ها، ستون، بیانگر الگوریتم‌های انتخاب ویژگی برخط و سطرها به مجموعه‌های داده‌ای اشاره دارد. بهترین مقدار در هر سطر به صورت پررنگ و زیرخط‌دار نشان داده شده است. دو سطر آخر در هر جدول نیز نشان‌دهنده متوسط دقت دسته‌بند مربوط و رتبه به دست آمده با استفاده از آزمون فریدمن روی همه مجموعه داده‌ها است.

جدول ۲، دقت‌های به دست آمده با استفاده از دسته‌بند DT را نشان می‌دهد. می‌توان مشاهده کرد که MMIOSFS در شش مورد (شامل Car، Glass، Hypothiroid، Ionsphere، Mushroom و Soybean) بهترین دقت را در میان روش‌ها به دست آورده است. این در حالی است که SAOLA، OSFS، fast-OSFS، Alpha-Investing و OSFSMI به ترتیب تنها در ۲، ۳، ۲، ۲ و ۳ مورد بهترین نتایج را به دست آورده‌اند.

1. https://en.wikipedia.org/wiki/Cross-validation_(statistics)

جدول ۴: متوسط تعداد ویژگی‌های انتخاب‌شده توسط الگوریتم‌های انتخاب ویژگی برخط.

	MMIOSFS	SAOLA	OSFS	Fast-OSFS	Alpha-Investing	OSFSMI
Breast-cancer	۴	۳	۲	۱	۳	۳
Breast-w	۲	۸	۵	۵	۵	۵
Car	۵	۵	۵	۵	۴	۲
Credit-a	۴	۲	۳	۳	۷	۴
Credit-g	۵	۱	۴	۴	۳	۶
Glass	۴	۳	۳	۳	۴	۴
Hypothiroi	۵	۱	۴	۵	۶	۴
Ionsphere	۳	۳	۳	۳	۵	۴
Letter	۹	۱۰	۹	۱۲	۱۵	۹
Mushroom	۳	۴	۸	۸	۱۶	۷
Sonar	۵	۸	۳	۳	۳	۵
Soybean	۸	۱۳	۷	۷	۱۴	۵
Vote	۳	۴	۳	۳	۸	۵
Mean	۴٫۶	۵	۴٫۵۳	۴٫۶۷	۷٫۱۵	۴٫۸
Fridman-rank	۳٫۵	۳٫۳	۲٫۷	۳٫۰	۴٫۶	۳٫۶

همچنین به منظور مقایسه بیشتر، مقدار میانگین و رتبه الگوریتم‌ها با استفاده از آزمون فریدمن، در دو سطر آخر نشان داده شده است. همان طور که جدول ۲ نشان می‌دهد مقدار میانگین دقت این دسته‌بند به ازای الگوریتم‌های SAOLA، OSFS، fast-OSFS، Alpha-Investing و OSFSMI عبارت است از ۸۱٫۳۷، ۸۰٫۶۹، ۸۰٫۵۲، ۸۳٫۱۹ و ۸۱٫۵۸. در حالی که متوسط دقت دسته‌بند به ازای الگوریتم MMIOSFS، ۸۳٫۹۹ است. پس می‌توان نتیجه گرفت الگوریتم پیشنهادی به طور متوسط تقریباً ۳٪ بهتر از سایر الگوریتم‌ها عمل می‌کند. علاوه بر این، بر اساس رتبه به دست آمده توسط آزمون فریدمن که در سطر آخر نشان داده شده است، می‌توان مشاهده کرد که الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها رتبه کمتری دارد که این مطلوبیت الگوریتم پیشنهادی را از دیدگاه دقت دسته‌بندی DT نشان می‌دهد.

نتایج به دست آمده به ازای دسته‌بند KNN روی همه مجموعه‌های داده‌ای و الگوریتم‌ها در جدول شماره ۳ آورده شده است. طبق نتایج به دست آمده MMIOSFS، SAOLA، OSFS، fast-OSFS، Alpha-Investing و OSFSMI به ترتیب در ۷، ۳، ۳، ۳، ۴ و ۱ مورد بهترین دقت دسته‌بندی را داشته‌اند. علاوه بر این میانگین دقت به دست آمده و رتبه فردیمن به ازای همه مجموعه‌های داده‌ای در دو سطر آخر نشان داده شده است. بر اساس نتایج به دست آمده، متوسط دقت MMIOSFS تقریباً ۵٪ از سایر الگوریتم‌ها بیشتر است.

همچنین رتبه فردیمن الگوریتم‌های MMIOSFS، SAOLA، OSFS، fast-OSFS، Alpha-Investing و OSFSMI به ترتیب عبارت است از ۲٫۵، ۴٫۳۴، ۳٫۶۹، ۳٫۹۲، ۲٫۷۳ و ۳٫۸۰ که نشان‌دهنده برتری روش پیشنهادی نسبت به سایر روش‌ها از دیدگاه دقت KNN است.

٤-۲ تعداد ویژگی‌های انتخاب‌شده

جدول ۴، میانگین تعداد ویژگی‌های انتخاب‌شده توسط الگوریتم‌های انتخاب ویژگی مختلف را نشان می‌دهد. بر اساس نتایج آورده‌شده در جدول ۴، می‌توان مشاهده کرد که MMIOSFS، SAOAL، OSFS، fast-OSFS، Alpha-Investing و OSFSMI به ترتیب در ۵، ۵، ۶، ۶، ۱ و ۲ مورد کمترین تعداد ویژگی را انتخاب کرده‌اند. این نتایج نشان می‌دهد که الگوریتم‌های OSFS و fast-OSFS معمولاً تعداد ویژگی‌های کمتری و روش Alpha-Investing تعداد ویژگی‌های بیشتری نسبت به الگوریتم پیشنهادی انتخاب می‌کنند. همچنین میانگین تعداد ویژگی‌های انتخاب‌شده و رتبه فریدمن به ازای همه داده‌ها نیز نشان می‌دهد که از دیدگاه تعداد ویژگی‌های انتخابی الگوریتم‌های OSFS و fast-OSFS معمولاً از مطلوبیت بالاتر و Alpha-Investing از مطلوبیت کمتری در مقایسه با سایر الگوریتم‌ها برخوردار هستند.

به طور کلی از آنجا که روش‌های OSFS و fast-OSFS تمام زیرمجموعه‌های ممکن از ویژگی‌های انتخاب‌شده را به منظور پیداکردن ویژگی‌های اضافی جستجو می‌کنند پس معمولاً تعداد کمتری ویژگی انتخاب می‌کنند. همچنین از آنجا که روش Alpha-Investing توانایی حذف ویژگی‌های اضافی قبلاً انتخاب‌شده را ندارد در نتیجه تعداد ویژگی‌های بیشتری در مقایسه با سایر روش‌ها انتخاب می‌کند. از این میان، روش‌های MMIOSFS، SAOLA و OSFSMI معمولاً تعداد یکسانی ویژگی انتخاب می‌کنند.

٤-۳ زمان اجرا

متوسط زمان اجرای هر کدام از الگوریتم‌ها بر حسب ثانیه در جدول ۵ نشان داده شده است. بر اساس این نتایج، متوسط زمان اجرای MMIOSFS از الگوریتم‌های SAOLA، Alpha-Investing و OSFSMI بالاتر و تنها از روش‌های OSFS و fast-OSFS پایین‌تر است. از آنجا که روش‌های Alpha-Investing و SAOLA هر کدام به ترتیب از روش گراف طیفی و اطلاعات متقابل دوتایی استفاده می‌کنند پس مرتبه زمانی این الگوریتم‌ها همواره کمتر و در نتیجه سریع‌تر از سایر روش‌ها عمل می‌کنند. همچنین چون روش‌های OSFS و fast-OSFS تمام زیرمجموعه‌های ممکن از مجموعه ویژگی‌های انتخاب‌شده را به منظور پیداکردن ویژگی‌های افزونه جستجو می‌کنند پس زمان اجرای این الگوریتم‌ها به طور متوسط از الگوریتم پیشنهادی بالاتر است.

طبق نتایج به دست آمده در سطر آخر رتبه فریدمن MMIOSFS، SAOLA، OSFS، fast-OSFS، Alpha-Investing و OSFSMI به ترتیب عبارت است از ۳٫۸، ۲، ۵٫۱، ۳٫۸، ۱٫۱۱ و ۵٫۰۳ که نشان می‌دهد مطلوبیت روش پیشنهادی از روش‌های OSFS، fast-OSFS و OSFSMI بهتر و از SAOLA و Alpha-Investing کمتر است.

جدول ۵: متوسط زمان اجرای الگوریتم‌های انتخاب ویژگی برخط مختلف بر حسب ثانیه.

	MMIOSFS	SAOLA	OSFS	Fast-OSFS	Alpha-Investing	OSFSMI
Breast-cancer	۰/۰۳	۰/۰۰۸	۰/۰۳	۰/۰۱	۰/۰۰۵	۰/۰۸
Breast-w	۰/۰۴	۰/۰۹	۰/۱۳	۰/۱۲	۰/۰۰۶	۰/۰۹
Car	۰/۰۴	۰/۰۰۷	۰/۰۷	۰/۰۶	۰/۰۰۶	۰/۰۸
Credit-a	۰/۱۲	۰/۰۱	۰/۰۵	۰/۰۳	۰/۰۰۸	۰/۰۹
Credit-g	۰/۱۰	۰/۰۴	۰/۱۱	۰/۰۵	۰/۰۱۱	۰/۱۰
Glass	۰/۰۳	۰/۰۰۷	۰/۰۵	۰/۰۲	۰/۰۰۵	۰/۰۹
Hypothiro	۰/۱۱	۰/۰۵	۰/۱۱	۰/۱۴	۰/۰۲	۰/۷۹
Ionsphere	۰/۰۹	۰/۰۱	۰/۱۵	۰/۰۴	۰/۰۱	۰/۱۵
Letter	۱/۰۱	۰/۰۹	۲/۲۲	۱/۱۱	۰/۰۸	۰/۱
Mushrom	۰/۵۷	۰/۰۲	۲/۳۳	۰/۷۴	۰/۰۵	۰/۱۴
Sonar	۰/۰۲	۰/۰۱	۰/۰۸	۰/۰۵	۰/۰۰۹	۰/۱۶
Soybean	۰/۱۰	۰/۰۴	۰/۵۶	۰/۱۷	۰/۰۱	۰/۲۶
Vote	۰۴/	۰/۰۱	۰/۰۷	۰/۰۲	۰/۰۰۶	۰/۰۹
Mean	۰/۱۸	۰/۰۳	۰/۴۶	۰/۲۰	۰/۰۱	۰/۱۷
Fridman-rank	۳/۸	۲	۵/۱	۳/۸	۱/۱۱	۵/۰۳

جدول ۶: مقایسه نرخ مقیاس‌پذیری الگوریتم‌های انتخاب ویژگی برخط مختلف (به طوری که P به تعداد کل ویژگی‌های تا آن لحظه دیده شده اشاره می‌کند).

MMIOSFS	SAOLA	Fast-OSFS	OSFS	Alpha-Investing	OSFSMI
$O(S_t)$	$O(S_t)$	$O(\lvert S_t\rvert^{\lvert S_t\rvert})$	$O(\lvert S_t\rvert^{\lvert S_t\rvert})$	$O(P\lvert S_t\rvert^2)$	$O(S_t)$

٤-٤ مقیاس‌پذیری

یکی از چالش‌های اساسی در الگوریتم‌های انتخاب ویژگی برخط، مقیاس‌پذیری است که می‌توان میزان آن را با پیچیدگی محاسباتی الگوریتم‌ها به دست آورد. هرچه پیچیدگی محاسباتی یک الگوریتم بیشتر باشد پس مقیاس‌پذیری آن کمتر است. به عبارت دیگر، مقیاس‌پذیری به سرعت رشد پیچیدگی محاسباتی الگوریتم اشاره دارد.

همان طور که گفته شد، الگوریتم دارای دو فاز اصلی است. فاز اول مربوط به محاسبه میزان ارتباط ویژگی تازه‌وارد با برچسب کلاس است که تنها یک بار و با ورود هر ویژگی جدید اجرا می‌شود و فاز دوم که مربوط به حذف ویژگی‌های افزونه است، به ازای همه ویژگی‌های انتخاب‌شده تا لحظه t (S_t) اجرا می‌شود. با این فرض که تعداد مقادیر ممکن برای برچسب C و ویژگی دلخواه f به ترتیب n و k باشد، میزان پیچیدگی محاسباتی هر کدام از فازها را در ادامه شرح می‌دهیم.

پیچیدگی فاز اول: در این فاز میزان ارتباط هر ویژگی f با برچسب کلاس با استفاده از (۷) محاسبه می‌شود. طبق این فرمول مرتبه زمانی اطلاعات متقابل عبارت است از $O(n.k)$.

پیچیدگی محاسباتی فاز دوم: در این فاز به ازای هر ویژگی f در مجموعه ویژگی‌های انتخابی S_t، میزان تأثیر حذف‌شدن آن در کاهش آنتروپی برچسب کلاس با استفاده از (۱۵) محاسبه می‌شود. اما همان طور که گفته شد، در این رابطه در ابتدا متغیر تصادفی مشترک میان مجموعه متغیرهای موجود در S_t و $S_t \setminus \{f\}$ محاسبه می‌شود که پیچیدگی آن عبارت است از $O(N^2)$. همان طور که گفته شد، N به تعداد نمونه‌های آموزشی یا همان سطرها اشاره می‌کند و در نتیجه می‌توان گفت پیچیدگی محاسباتی (۱۶) عبارت است از $O(N^2.n.k)$. حال از آنجا که این مقدار به ازای تمام ویژگی‌های موجود در S_t محاسبه می‌شود پس پیچیدگی محاسباتی فاز دوم عبارت است از $O(\lvert S_t\rvert.N^2.n.k)$. در نهایت چون متغیرهای n، k و N مقادیر ثابتی هستند، پیچیدگی محاسباتی فاز دوم عبارت است از $O(\lvert S_t\rvert)$. پس می‌توان گفت پیچیدگی محاسباتی الگوریتم پیشنهادی در هر مرحله تنها به تعداد ویژگی‌های انتخاب‌شده تا لحظه t (S_t) وابسته است و نه به تعداد همه ویژگی‌ها، پس می‌توان گفت الگوریتم پیشنهادی مقیاس‌پذیر است. در جدول ۶، پیچیدگی محاسباتی الگوریتم‌ها آورده شده است. طبق این جدول الگوریتم‌های OSFS، fast-OSFS و Alpha-Investing مقیاس‌پذیر نیستند، چون پیچیدگی محاسباتی آنها یا نمایی است (الگوریتم‌های OSFS و fast-OSFS) و یا به تعداد ویژگی‌های دیده‌شده تا آن لحظه وابسته است (Alpha-Investing). پس می‌توان گفت، پیچیدگی محاسباتی الگوریتم پیشنهادی از الگوریتم‌های OSFS، fast-OSFS و Alpha-Invetsing بهتر و همانند الگوریتم‌های SAOLA و OSFSMI و از مرتبه خطی تعداد ویژگی‌های انتخاب‌شده است.

۵- بحث

یک روش انتخاب ویژگی خوب، روشی است که حد واسط خوبی میان همه چالش‌های مربوط به جریان داده‌های برخط (مانند دقت دسته‌بندی، اندازه زیرمجموعه انتخاب‌شده از ویژگی‌ها، مقیاس‌پذیری و مدت زمان انتخاب زیرمجموعه از ویژگی‌ها) ایجاد کند. طبق نتایج به دست آمده می‌توان ادعا کرد الگوریتم پیشنهادی نتایج بهتری در مقایسه با سایر الگوریتم‌ها به دست آورده است. چون طبق جدول‌های ۲ و ۳، دقت دسته‌بند KNN و DT مربوط به الگوریتم پیشنهادی از سایر الگوریتم‌ها بالاتر است. همچنین اگرچه بر اساس جدول‌های ۴ و ۵ الگوریتم‌های OSFS و fast-OSFS، تعداد ویژگی‌های کمتری نسبت به سایر الگوریتم‌ها از جمله الگوریتم پیشنهادی انتخاب کرده‌اند، اما مدت زمان لازم به منظور انجام فرایند انتخاب ویژگی در این الگوریتم‌ها خیلی بیشتر از سایر روش‌ها است. از طرف دیگر، اگرچه الگوریتم Alpha-Investing

نسبت به سایر روش‌ها مدت زمان کمتری به منظور انتخاب ویژگی‌ها طی می‌کند اما تعداد ویژگی‌های انتخابی توسط این الگوریتم خیلی بیشتر است و نیز الگوریتم مقیاس‌پذیر نیست، چون پیچیدگی محاسباتی آن به همه ویژگی‌های دیده‌شده وابسته است (جدول ۶). همچنین الگوریتم SAOLA با صرف زمان کمتری، تعداد بیشتری ویژگی با دقت دسته‌بندی کمتر نسبت به الگوریتم پیشنهادی انتخاب می‌کند. علاه بر این دقت، تعداد ویژگی‌های انتخابی و زمان اجرای روش پشنهادی از الگوریتم OSFSMI بهتر است. پس می‌توان نتیجه گرفت الگوریتم پیشنهادی حد واسط بهتری میان همه چالش‌ها ایجاد کرده است.

٦- نتیجه‌گیری و کارهای آینده

در این مقاله یک الگوریتم انتخاب ویژگی با جریان ویژگی برخط به نام MMIOSFS ارائه شد. ایده اصلی در این الگوریتم استفاده از اطلاعات متقابل چندمتغیره و راهکار متغیرهای تصادفی توأم در محاسبه ارتباط و افزونگی ویژگی‌ها به روش برخط به منظور انتخاب ویژگی‌های مؤثر و حذف ویژگی‌های افزونه بود. برای ارزیابی کارایی روش پیشنهادی آزمایش‌های متعددی در مقایسه با چندین روش انتخاب ویژگی برخط از جمله SALOA، OSFS، fast-OSFS، Alpha-Investing و OSFSMI انجام شد. در این آزمایش‌ها کارایی الگوریتم پیشنهادی از چندین دیدگاه متفاوت مانند دقت دسته‌بند، تعداد ویژگی‌های انتخاب‌شده و زمان اجرا ارزیابی شد. با این که نتایج به دست آمده برتری الگوریتم پیشنهادی نسبت به سایر روش‌ها را نشان داد، اما هنوز چالش‌های بسیاری در مورد داده‌های برخط با جریان ویژگی وجود دارد که الگوریتم پیشنهادی این مقاله نیز از حل آنها ناتوان است. یکی از این چالش‌ها، مسأله چند برچسبی بودن نمونه‌های آموزشی است به این معنی که در هر لحظه هر نمونه به بیش از یک برچسب متعلق باشد. در آینده این راهکار را به گونه‌ای توسعه خواهیم داد که در داده‌های آموزشی برخط و چندبرچسبی قابل استفاده باشد. یکی دیگر از مسایلی که به عنوان کارهای آینده پیشنهاد می‌شود، بررسی بحث تغییر مفهوم در الگوریتم‌های انتخاب ویژگی در جریان داده‌ها است. تغییر مفهوم یک بحث اساسی در تحلیل جریان داده‌ها است. یکی از پیشنهادهایی که برای کارهای آینده مطرح است، این است که الگوریتم‌های انتخاب ویژگی جریان داده را با روش‌های خوشه‌بندی این نوع داده‌ها ترکیب کرد. در این پیشنهاد ویژگی‌ها بر اساس میزان شباهتی که به ویژگی‌های انتخاب‌شده دارند می‌توانند به عنوان یک خوشه جدید مطرح گردند یا به خوشه‌های فعلی اضافه شوند.

مراجع

[1] J. Dean, Big Data, Data Mining, and Machine Learning: Value Creation for Business Leaders and Practitioners, CreateSpace Independent Publishing Platform. 318, 2014.

[2] S. Gilpin, B. Qian, and I. Davidson, "Efficient hierarchical clustering of large high dimensional datasets," in *Proc. of the 22nd ACM Int. Conf. on Information, Knowledge Management,* pp. 1371-1380, *San Francisco,* CA, USA, 27 Oct.-1 Nov. 2013.

[3] A. K. Farahat and A. Elgohary, A. Ghodsi, and M. Kamel, "Greedy column subset selection for large-scale data sets," *Knowledge and Information Systems*, vol. 45, pp. 1-34, 2014.

[4] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Applied Soft Computing*, vol. 43, pp. 117-130, Jun. 2016.

[5] M. Labani, *et al.*, "A novel multivariate filter method for feature selection in text classification problems," *Engineering Applications of Artificial Intelligence*, vol. 70, pp. 25-37, Apr. 2018.

[6] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226-1238, Aug. 2005.

[7] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu, "Multimodal graph-based reranking for web image search," *IEEE Trans. on on Image Processing*, vol. 21, no. 11, pp. 4649-4661, Jul. 2012.

[8] X. Hu, P. Zhou, P. Li, J. Wang, and X. Wu, "A survey on online feature selection with streaming features," *Frontiers of Computer Science*, vol. 12, no. 3, pp. 479-493, May 2018.

[9] S. C. H. Hoi, J. Wang, P. Zhao, and R. Jin, "Online feature selection for mining big data," in Proc. of the 1st Int. Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, pp. 93-100, Beijing, China, 12-14 Aug. 2012.

[10] K. Yu, *et al.*, "Scalable and accurate online feature selection for big data," *ACM Trans. on Knowledge Discovery from Data*, vol. 11, no. 2, pp. 1-39, Jul. 2016.

[11] X. Wu, X. Wu, W. Ding, and J. Pei, "Online streaming feature selection," in *Proc. of the 27th Int. Conf. on Machine Learning*, pp. 1159-1166, 21-30, Jun. 2010.

[12] J. Zhou, D. P. Foster, R. A. Stine, and L. H. Ungar, "Streamwise feature selection," *Journal of Machine Learning Research*, vol. 7, no. 67, pp. 1861-1885, Jul. 2006.

[13] S. Perkins and J. Theiler, "Online feature selection using grafting," in *Proc. of the 12th Int. Conf. on Machine Learning*, pp. 592-599, Washington, DC, USA,12-21Aug. 2003.

[14] X. Wu, *et al.*, "Online feature selection with streaming features," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1178-1192, May 2013.

[15] J. Zhou, et al., "Streaming feature selection using alpha-investing," in Proc. of the *11th ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining*, pp. 384-393, Chicago, IL, USA, Chicago, IL, USA, 21-24 Aug. 2005.

[16] M. Rahmaninia and P. Moradi, "OSFSMI: online stream feature selection method based on mutual information," *Applied Soft Computing*, vol. 68, pp. 733-746, Jul. 2018.

[17] J. Liang, F. Wang, C. Dang and Y. Qian, "A group incremental approach to feature selection applying rough set technique," *IEEE Trans. on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 294-308, Feb. 2014.

[18] K. Henni, N. Mezghani, and C. Gouin-Vallerand, "Unsupervised graph-based feature selection via subspace and pagerank centrality," *Expert Systems with Applications*, vol. 114, pp. 46-53, Dec. 2018.

[19] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection based on joint mutual information," *Expert Systems with Applications,* vol. 14, pp. 1-55, Sept. 2015.

[20] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. on Neural Networks*, vol. 5, no. 4, pp. 537-550, Jul. 1994.

[21] C. Lai, M. J. T. Reinders, and L. Wessels, "Random subspace method for multivariate feature selection," *Pattern Recognition Letters*, vol. 27, no. 10, pp. 1067-1076, Jul. 2006.

[22] M. Haindl, P. Somol, D. Ververidis, and C. Kotropoulos, "Feature selection based on mutual correlation," *Pattern Recognition, Image Analysis and Applications*, *CIARP'06*, Lecture Notes in Computer Science, vol. 4225, pp. 569-577, Jul. 2006.

[23] A. J. Ferreira and M. A. T. Figueiredo, "An unsupervised approach to feature discretization and selection," *Pattern Recognition*, vol. 45, no. 9, pp. 3048-3060, Sept. 2012.

[24] J. Liang, F. Wang, C. Dang, and Y. Qian, "A group incremental approach to feature selection applying rough set technique," *IEEE Trans. on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 294-308, Feb. 2014.

[25] Y. Zhang, A. Yang, C. Xiong, T. Wang, and Z. Zhang, "Feature selection using data envelopment analysis," *Knowledge-Based Systems*, vol. 64, pp. 70-80, Apr. 2014.

[26] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2005.

[27] D. B. Skalak, "Prototype and feature selection by sampling and random mutation hill climbing algorithms," in *Proc. 11th Int. Conf. on Machine Learning*, vol. 11, pp. 293-301, New Brunswick, NJ, USA, 10-13 Jul. 1994.

[28] D. Gelbart, N. Morgan, and A. Tsymbal, "Hill-climbing feature selection for multi-stream ASR," in *Proc. 10th Annual Conf. of the Int. Speech Communication Association, INTERSPEECH'09,* 4 pp., Brighton, UK, Sept. 2009.

[29] S. Tabakhi, P. Moradi, and F. Akhlaghian, "An unsupervised feature selection algorithm based on ant colony optimization," *Engineering*

Applications of Artificial Intelligence, vol. 32, pp. 112-123, Feb. 2014.
[30] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389-422, Jan. 2002.
[31] M. Pratama, G. Zhang, M. J. Er, and S. Anavatti, "An incremental type-2 meta-cognitive extreme learning machine," *IEEE Trans. on Cybernetics*, vol. 47, no. 2, pp. 339-353, Feb. 2017.
[32] E. Lughofer, "On-line incremental feature weighting in evolving fuzzy classifiers," *Fuzzy Sets and Systems*, vol. 163, no. 1, pp. 1-23, Jan. 2011.
[33] S. Eskandari and M. M. Javidi, "Online streaming feature selection using rough sets," *International Journal of Approximate Reasoning*, vol. 69, pp. 35-57, Feb. 2016.
[34] J. Liu, Y. Lin, T. Li, W. Wang, and S. Wu, "Online multi-label streaming feature selection based on neighborhood rough set," *Pattern Recognition*, vol. 84, pp. 273-287, Jul. 2018.
[35] H. Wang, *et al.*, "Multi-label online streaming feature selection based on spectral granulation and mutual information," in *Rough Sets*, Cham: Springer International Publishing, 2018.
[36] H. Li, X. Wu; Z. Li, and W. Ding, "Group feature selection with streaming features," in *Proc. IEEE 13th Int. Conf. on Data Mining* pp. 1109-1114,, Dallas, TX, USA, Dec. 2013.
[37] J. Wang, *et al.*, "Online feature selection with group structure analysis," *IEEE Trans. on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 3029-3041, Nov. 2015.
[38] J. Liu, Y. Lin, S. Wu, and C. Wang, "Online multi-label group feature selection," *Knowledge-Based Systems*, vol. 143, pp. 42-57, Mar. 2018.
[39] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, 2006.
[40] K. He and G. Meeden, "Selecting the number of bins in a histogram: a decision theoretic approach," *J. of Statistical Planning and Inference*, vol. 61, no. 1, pp. 49-59, May 1997.
[41] W. McGill, "Multivariate information transmission," *Trans. of the IRE Professional Group on Information Theory*, vol. 4, no. 4, pp. 93-111, Jun. 1954.
[42] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175-185, Aug. 1992.
[43] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers Inc., 2011.
[44] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86-92, Mar. 1940.
[45] K. Yu, W. Ding, and X. Wu, "LOFS: a library of online streaming feature selection," *Knowledge-Based Systems*, vol. 113, pp. 1-3, Mar. 2016.

مریم رحمانی‌نیا در سال ۱۳۸۷ مدرک کارشناسی خود را در رشته علوم کامپیوتر از دانشگاه شهید بهشتی تهران و در سال ۱۳۸۹ مدرک کارشناسی ارشد علوم کامپیوتر خود را از دانشگاه تحصیلات تکمیلی علوم پایه زنجان دریافت نمود. از سال ۱۳۹۰ نام برده به عنوان عضو هیأت علمی در دانشگاه آزاد اسلامی مرکز قصرشیرن مشغول به تدریس بوده و تحصیلات خود را در مقطع دکتری مهندسی کامپیوتر در سال ۱۳۹۸ در دانشگاه آزاد سنندج به پایان رسانید. زمینه‌های علمی مورد علاقه ایشان داده‌کاوی، شبکه‌های اجتماعی، گراف، شبکه‌های عصبی و سایر موضوعات مرتبط با هوش مصنوعی است.

پرهام مرادی دوره کارشناسی را در رشته مهندسی کامپیوتر در سال ۱۳۸۱ در دانشگاه صنعتی امیرکبیر به پایان رسانید. ایشان همچنین دوره کارشناسی ارشد و دکتری را در رشته علوم کامپیوتر دانشگاه صنعتی امیرکبیر به ترتیب در سال‌های ۱۳۸۴ و ۱۳۸۹ به پایان رسانید. ایشان بخشی از تحقیقات دوره دکتری را به عنوان فرصت مطالعاتی را در دانشگاه پلی تکنیک فدرال لوزان سویس (EPFL) در سال ۱۳۸۸ به انجام رسانید. نام‌برده در سال ۱۳۸۹ به عنوان هیات علمی در گروه مهندسی کامپیوتر دانشگاه کردستان مشغول فعالیت شدند. در ادامه در سال ۱۳۹۵ به مرتبه دانشیاری ارتقاء یافتند. حاصل تحقیقات ایشان تاکنون چاپ بیش از ۹۰ مقاله در مجلات و کنفرانس‌های معتبر بوده است. ایشان همچنین بر اساس گزارش منتشره دانشگاه استنفورد، به عنوان دانشمند دو درصد برتر در سال ۲۰۱۹ شناخته شدند. تحقیقات ایشان شامل یادگیری ماشین، یادگیری تقویتی، تحلیل شبکه¬های اجتماعی، علوم داده و سیستم‌های توصیه‌گر است. ایشان در دانشگاه کردستان مسئولیت‌هایی همچون مدیر گروه، مدیر آموزش‌های آزاد و مجازی و ریاست دانشکده مهندسی را بر عهده داشته است.