دسته‌بندی بیزین جمعی با استفاده از انتخاب ویژگی رپر مبتنی بر الگوریتم ژنتیک در تشخیص هرزنامه

وحید نصرتی[1]
دانشجوی دکتری، مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه اراک، اراک، ایران

محسن رحمانی
دانشیار، مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه اراک، اراک، ایران

مدیریت اطلاعات
دوره ۶، شماره ۲
پاییز و زمستان ۱۳۹۹

چکیده: جایگاه ایمیل در ارتباطات، با ورود پدیده‌ای به نام هرزنامه با تهدید جدی مواجه شده است. تاکنون، به‌منظور مقابله با این پدیده، روش‌های فراوانی پیشنهاد شده که یکی از مهم‌ترین این روش‌ها، دسته‌بندی آنها بر اساس محتوا به دو دسته هرزنامه و غیرهرزنامه است. دسته‌بندی بر اساس محتوا با استفاده از کلمات به‌عنوان ویژگی انجام می‌شود که به‌دلیل تعداد زیاد ویژگی‌ها، استفاده از یک سازوکار انتخاب ویژگی کارآمد موضوعی حیاتی به نظر می‌رسد. بر این اساس، تمرکز روش پیشنهادی در این مقاله روی انتخاب ویژگی‌های مفید بوده و یک فرایند انتخاب ویژگی رپر با بهره‌گیری از الگوریتم قدرتمند ژنتیک و با همکاری دسته‌بند بیزین که دارای کارایی بالایی در مسائل دسته‌بندی متون است، ارائه می‌شود. روش کار نیز به این صورت است که ابتدا یک بردار ویژگی اولیه ساخته شده، سپس با ضرب کردن آن در یک ماتریس با عنوان ماتریس انتقال، با استفاده از الگوریتم ژنتیک، روی آن عملیات بهینه‌سازی اعمال شده و در پایان، k بردار ویژگی نهایی ساخته می‌شوند. عملیات دسته‌بندی نیز به‌صورت جمعی و با اعمال k دسته‌بند بیزین روی بردارهای ویژگی اعمال شده و از بین آنها رأی‌گیری انجام می‌شود. روش پیشنهادی روی دو پایگاه داده اجرا شده که بر اساس نتایج، روش پیشنهادی با مقدار ۷ = k دارای نرخ صحت ۸۷/۷۶ و ۸۷/۹۱ در دو پایگاه داده PU۱ و PU۲ است. همچنین نتایج مقایسه روش پیشنهادی، حاکی از کارآمدی روش پیشنهادی در مقایسه با بیزین پایه و دو دسته‌بند SVM و KNN است.

کلیدواژه‌ها: ایمیل، هرزنامه، دسته‌بندی، الگوریتم ژنتیک، انتخاب ویژگی، ماتریس انتقال، یادگیری جمعی.

---

۱. نویسنده مسئول: vh_nosraty@yahoo.com

مقدمه

سرویس ایمیل امروزه به‌عنوان یک روش ارتباطی مهم، در بین افراد جامعه، دارای محبوبیت فراوان و روزافزونی است. همین محبوبیت و رایگان بودن ایمیل، برای تبلیغات با قیمت ارزان، زمینهٔ مناسبی را فراهم می‌آورد. نامه‌های ناخواسته که معمولاً به‌عنوان هرزنامه شناخته می‌شوند، یکی از تهدیدها برای امنیت اینترنت به شمار می‌روند. هرزنامه‌ها، صندوق پستی افراد را با حجم وسیعی از پیام‌های ناخواسته اشغال می‌کنند که علاوه بر هدررفت پهنای باند شبکه و فضای ذخیره‌سازی، به توزیع سریع اطلاعات نادرست و گسترش کدهای مخرب در بین کاربران اینترنت منجر شده و افراد خردسال را نیز در معرض محتوای نامناسب قرار می‌دهد (Faris et al, 2019). گسترش نامه‌های مخرب به حدی است که بر اساس پژوهش اخیر آزمایشگاه کسپرسکی[1]، بیش از ۵۶ درصد از کل ایمیل‌ها هرزنامه هستند (Saidani et al, 2020: 1). آنچه این مسئله را بدتر می‌کند این است که هرزنامه‌نویس‌ها به‌طور پیوسته، برای دور زدن فیلترها، روش‌های جدیدی ابداع می‌کنند. از طرف دیگر، حجم گستردهٔ جریان داده بین صدها میلیون نفر و تعداد زیاد صفات، این مشکل را بسیار پیچیده‌تر می‌کند، بنابراین، پیشنهاد مدل‌های کارآمد و سازگار با تشخیص هرزنامه به یک ضرورت تبدیل شده است (Faris et al, 2019: 67).

تاکنون، برای مقابله با هرزنامه روش‌های مختلفی ارائه شده است. دادا و همکاران[2] (۲۰۱۹)، مسائل و مشکلات موجود در این زمینه و آخرین روش‌های موجود را بررسی کرده‌اند. به‌طور کلی، به‌منظور شناسایی پیام‌های مخرب، از دو نوع اطلاعات شامل اطلاعات محتوایی[3] و غیرمحتوایی[4] استفاده می‌شود (Hu et al, 2010). در روش‌های مبتنی بر اطلاعات غیرمحتوایی از سرصفحه‌های[5] ایمیل و اطلاعات فرستندهٔ مربوطه مانند دامنهٔ آدرس ایمیل فرستنده (آدرس IP)، سابقه[6]، سبک نوشتن و زمان ارسال استفاده می‌شود و روش‌های مبتنی بر محتوا از اطلاعات متنی موجود در متن ایمیل‌ها و موضوعات ایمیل استفاده می‌کنند. در پژوهش حاضر، تمرکز اصلی بر روش‌های مبتنی بر محتوا برای شناسایی هرزنامه قرار دارد. در بحث تشخیص و شناسایی هرزنامه، با مجموعه‌ای از سندهای متنی با عنوان ایمیل سروکار داریم که می‌توان برای این منظور، از روش‌های پردازش متنی کمک گرفت. یکی از روش‌هایی که در این زمینه کاربرد فراوانی دارد، دسته‌بندی ایمیل‌ها بر اساس محتوا با عنوان عملیات دسته‌بندی متن[7] است که در آن، تعدادی فایل متنی (ایمیل) وجود دارد که بایستی (با تجزیه و تحلیل آنها) دسته‌بندی شوند.

الگوریتم‌های دسته‌بند بر اساس یک سری ویژگی به دسته‌بندی پیام‌های رسیده اقدام می‌کنند. همین موضوع باعث می‌شود کارایی آنها تا حدود زیادی مشروط به انتخاب ویژگی مناسب باشد. هرچقدر این ویژگی‌ها به‌صورت بهینه‌تر انتخاب شوند، باعث افزایش کارایی الگوریتم دسته‌بندی می‌شوند. به همین

1. Kaspersky
2. Dada et. al
3. Content based information
4. Non-content
5. Header
6. Reputation
7. Text classification

دلیل تاکنون برای بهبود کارایی دسته‌بند با بهینه کردن روش انتخاب ویژگی تلاش‌های مختلفی شده است (Kołcz et al, 2004).

انتخاب ویژگی می‌تواند به‌صورت فیلتر[1]، رپر[2] یا جاسازی‌شده[3] انجام شود. روش رپر به‌دلیل بهره‌گیری از یک الگوریتم دسته‌بندی، در مقایسه با فیلتر معمولاً نتایج بهتری دارد. انتخاب ویژگی رپر مبتنی بر الگوریتم ژنتیک، برای انتخاب ویژگی روشی رایج به شمار می‌رود و کارایی آن در حیطه‌های مختلف همچون صنعت پزشکی، پردازش تصویر، پردازش متن، بیوانفورماتیک و کاربردهای صنعتی اثبات شده است (Jadhav et al, 2018)، بنابراین در پژوهش حاضر، یک روش انتخاب ویژگی مبتنی بر الگوریتم ژنتیک در مسئلهٔ تشخیص هرزنامه ارائه می‌شود. برخلاف روش‌های پیشین، در روش پیشنهادی، به‌عنوان جواب نهایی از یک کروموزوم استفاده نمی‌شود و برای انتخاب k بردار از ویژگی‌های برتر از مجموعه‌ای از k کروموزوم برتر استفاده شده و در مرحلهٔ بعد، k دسته‌بند بیزین روی آنها اعمال شده و دسته‌بندی نهایی با رأی‌گیری ساده از بین دسته‌بندها انجام می‌شود. با توجه به اینکه الگوریتم ژنتیک شناخته‌شده‌ترین روش بهینه‌سازی هوشمند و الگوریتم تکاملی به شمار می‌رود، می‌توان با قرار دادن آن در کنار روش آماری همچون بیزین که خود یکی از روش‌های آماری مهم محسوب می‌شود، نتایج بهتری به دست آورد.

ایدهٔ استفاده از k الگوریتم دسته‌بند از مفهوم یادگیری جمعی[4] الهام گرفته شده است که در آنها برخلاف سازوکارهایی که فقط مبتنی بر یک دسته‌بند هستند، از مجموعه‌ای از دسته‌بندها استفاده می‌شود. به‌طور کلی، این نوع دسته‌بندها در تشخیص هرزنامه در مقایسه با روش‌های تکی کارایی بهتری دارند (Chinavle et al, 2009). بوستینگ[5] و بکینگ[6] از روش‌های مهمی هستند که به‌صورت گروهی عمل می‌کنند. روش بوستینگ با اندازه‌گیری خطای دسته‌بندی هر گروه[7] وزن مثال‌های اشتباه دسته‌بندی‌شده را برای گروه دسته‌بند بعدی افزایش می‌دهد. روش بکینگ نیز با آموزش دادن تعدادی دسته‌بند با مجموعه‌های مستقل از مثال‌های آموزشی سعی در کاهش واریانس دارد (DeBarr et al, 2012).

با توجه به اهمیت دو فرایند دسته‌بندی و انتخاب ویژگی در سیستم‌های تشخیص هرزنامه، باید برای یافتن روش کارآمد، به هر دوی این فرایندها توجه شود. در حالی که در اکثر پژوهش‌های حاضر تلاش شده است تا فقط یکی از این دو فرایند بهبود بخشیده شود، در پژوهش حاضر به‌دلیل وابستگی این دو بخش و اینکه یک الگوریتم انتخاب ویژگی کارآمد تأثیر بسیار زیادی بر کارایی فرایند دسته‌بندی دارد، به هر دوی آنها توجه شده و برای بهبود آنها تلاش شده است. همچنین در پژوهش حاضر، به‌منظور ساخت یک مدل قدرتمند تشخیص هرزنامه، به‌طور هم‌زمان از دو روش الگوریتم ژنتیک و یادگیری جمعی، به‌عنوان روش‌هایی موفق در عرصهٔ یادگیری ماشین، بهره گرفته شده است. در بخش انتخاب ویژگی، یک

1. Filter
2. Wrapper
3. Embedded
4. Ensemble learning
5. Boosting
6. Bootstrap aggregation (bagging)
7. Ensemble

الگوریتم رپر مبتنی بر الگوریتم ژنتیک و در بخش دسته‌بندی، یک سازوکار جمعی مبتنی بر الگوریتم بیزین ارائه می‌شود.

در ادامه، مقاله به این صورت ساختاربندی شده است: ابتدا، برخی روش‌های موجود در زمینهٔ موضوع پژوهش بیان می‌شوند. سپس در بخش بعد، مبحث فیلترینگ بررسی می‌شود که خود دارای دو قسمت است؛ ابتدا در قسمت نخست، خلاصه‌ای از الگوریتم بیزین مطرح می‌شود و سپس در قسمت دوم، به مبحث انتخاب ویژگی به‌عنوان یکی از مهم‌ترین بخش‌های الگوریتم دسته‌بند پرداخته می‌شود. در ادامه، شرح مختصری از الگوریتم ژنتیک داده شده و پس از آن، الگوریتم پیشنهادی مطرح می‌شود. در بخش بعدی، نتایج بیان شده و در نهایت، بخش پایانی به نتیجه‌گیری اختصاص دارد.

پیشینه پژوهش

تاکنون پژوهش‌های مختلفی به مطالعه و گروه‌بندی روش‌های دسته‌بندی پرداخته‌اند. میچین و همکاران[1] (۱۹۹۴) در پژوهش خود روش‌های موجود را به سه دستهٔ کلی روش‌های آماری، یادگیری ماشین و شبکه‌های عصبی تقسیم‌بندی کردند. از روش‌های آماری مهم می‌توان بیزین را نام برد که به‌دلیل سادگی و قدرت بالا، یکی از روش‌های موفق در این زمینه بوده و تاکنون نیز برای بهبود آن تلاش‌های بسیاری شده است (Rajaram et al., 2011)، اگرچه در بعضی مقاله‌ها بیزین را به‌عنوان یک روش یادگیری ماشین نیز در نظر گرفته‌اند (Hua Li et al, 2012). یادگیری ماشین نیز از روش‌های موفق دیگر در زمینهٔ دسته‌بندی متون است. تاکنون الگوریتم‌های یادگیری ماشینی متعددی ارائه شده است که از مهم‌ترین آنها می‌توان درخت تصمیم و k نزدیک‌ترین همسایه[2] (Crawford et al, 2004) را نام برد. اساس کار این‌گونه الگوریتم‌ها، یادگیری یک وظیفهٔ خاص بر اساس تعدادی نمونهٔ آموزشی است. شبکه‌های عصبی نیز شاخه‌ای از هوش مصنوعی هستند که از عملکرد مغز انسان الهام گرفته‌اند (Wang et al, 2006). تاکنون، برای مسئلهٔ دسته‌بندی، نسخه‌های مختلفی از این الگوریتم ارائه شده است (Xu et al, 2010 & Hua Li et al, 2012). علاوه بر این روش‌ها، روش‌هایی نیز وجود دارند که برای دسته‌بندی هرزنامه از ترکیب چندین روش استفاده می‌کنند (Mohammad et al, 2011& Su et al, 2010).

مبحث دیگری که به مبحث دسته‌بندی بسیار مرتبط است و حتی در بسیاری از پژوهش‌ها از آن به‌عنوان پیش‌پردازش نام برده‌اند، انتخاب ویژگی است. رویکرد رپر، سازوکار انتخاب ویژگی را با بهره‌گیری از الگوریتم دسته‌بندی انجام داده و زیرمجموعه‌ای از ویژگی‌ها را انتخاب می‌کند. روش رپر، به‌خصوص برای حل مسائلی مفید است که در آنها نمی‌توان تابع ارزندگی را به‌راحتی با یک معادلهٔ ریاضی دقیق بیان کرد. این روش به‌دلیل اجرای ساده، توجه بسیاری از پژوهش‌ها را به خود جلب کرده است (Kohavi & George, 1998). صحت الگوریتم دسته‌بند، به‌عنوان معیار ارزیابی برای انتخاب ویژگی‌ها استفاده می‌شود. مزیت روش‌های رپر این است که چون یک الگوریتم دسته‌بندی زیرمجموعهٔ نهایی انتخاب‌شده را

1. Michie et. al
2. K nearest neighber

تعیین می‌کند، بر کل فرایند انتخاب ویژگی کنترل بیشتری وجود دارد. همچنین استفاده از ظرفیت یادگیری ارائه‌شده توسط الگوریتم دسته‌بندی می‌تواند دقت بسیار بالایی ایجاد کند (Jadhav et al, 2018). فرایند انتخاب ویژگی هنگام محاسبهٔ ارزندگی یک زیرمجموعه از ویژگی، همهٔ زیرمجموعه‌ها در ویژگی‌های موجود را جست‌وجو می‌کند. به‌دلیل اینکه جست‌وجوی جامع از نظر محاسباتی بسیار گران است، برای انتخاب ویژگی، به روش‌های جست‌وجوی فراابتکاری، مانند الگوریتم ژنتیک (GA) علاقه نشان داده شده است.

تاکنون، به‌منظور انتخاب ویژگی در کاربردهای مختلف از جمله تشخیص هرزنامه، روش‌های مختلفی از الگوریتم ژنتیک ارائه شده است. امینی و یونزه[1] (۲۰۲۱) یک روش انتخاب ویژگی دولایه با استفاده از الگوریتم ژنتیک و شبکهٔ الاستیک را ارائه دادند. در لایهٔ نخست، الگوریتم ژنتیک (GA) به‌عنوان یک رپر برای جست‌وجوی زیرمجموعهٔ بهینه استفاده شده که هدف آن، کاهش ابعاد و خطای پیش‌بینی است. شبکهٔ الاستیک (EN) در لایهٔ دوم به روش پیشنهادی اضافه شده است تا پیش‌بینی‌های اشتباه احتمالی برای بهبود دقت پیش‌بینی از بین برود.

محمدزاده و همکاران[2] (۲۰۲۱)، برای انتخاب ویژگی در زمینهٔ شناسایی هرزنامهٔ ایمیل بر اساس مفهوم یادگیری مبتنی بر مخالفت[3] (OBL)، یک الگوریتم جدید از ترکیب الگوریتم‌های بهینه‌سازی وال (WOA) و گرده‌افشانی گل (FPA) ارائه دادند. روش کار بدین صورت است که ابتدا WOA اجرا می‌شود و هم‌زمان در حین اجرا، جمعیت WOA توسط OBL تغییر می‌یابد و برای افزایش دقت و سرعت هم‌گرایی، از آن به‌عنوان جمعیت اولیه FPA استفاده می‌شود.

فارسیس و همکاران[4] (۲۰۱۹)، برای شناسایی هرزنامه و شناسایی مهم‌ترین ویژگی‌ها بر اساس الگوریتم ژنتیک (GA) و شبکه‌های وزنی تصادفی (RWN)[5]، یک سیستم هوشمند ارائه دادند. یک قابلیت شناسایی خودکار نیز در سیستم پیشنهادی تعبیه شده است تا مهم‌ترین ویژگی‌ها را طی فرایند شناسایی انتخاب کند. هوانگو همکاران[6] (۲۰۰۷)، برای انتخاب ویژگی رپر[7] بر اساس اطلاعات متقابل، یک الگوریتم ژنتیک ترکیبی ارائه داده‌اند که دارای دو مرحلهٔ بهینه‌سازی بیرونی و درونی است. مرحلهٔ بیرونی به جست‌وجوی سراسری برای بهترین زیرمجموعهٔ ویژگی‌ها می‌پردازد که در آن اطلاعات متقابل بین کلاس‌های پیش‌بینی‌شده و واقعی به‌عنوان تابع ارزش الگوریتم ژنتیک عمل می‌کند و بهینه‌سازی داخلی جست‌وجوی محلی را به‌صورت فیلتر انجام می‌دهد.

با توجه به کارایی بسیار مناسب الگوریتم ژنتیک، در پژوهش‌های بسیاری، از آن به‌منظور بهینه‌سازی فرایندهای مختلفی بهره‌گیری شده است که در پژوهش‌های ذکرشده به برخی از آنها اشاره شد. با توجه به کاربرد فراوان الگوریتم ژنتیک در کاربردهای مختلف و به‌خصوص در انتخاب ویژگی، در این مقاله قصد

1. Amini & Yunze
2. Mohammadzadeh et. al
3. Opposition-based learning
4. Faris et. al
5. Random Weight Networks
6. Huang et. al
7. Wrapper

داریم برای تشخیص هرزنامه، سازوکار نوینی ارائه دهیم که الگوریتم ژنتیک را برای انتخاب ویژگی خود به کار می‌برد و با الهام از روش‌های یادگیری گروهی، چندین مدل دسته‌بندی مبتنی بر دسته‌بند بیزین ایجاد کرده و دسته‌بندی نهایی بر اساس رأی‌گیری بین این مدل‌ها به دست می‌آید.

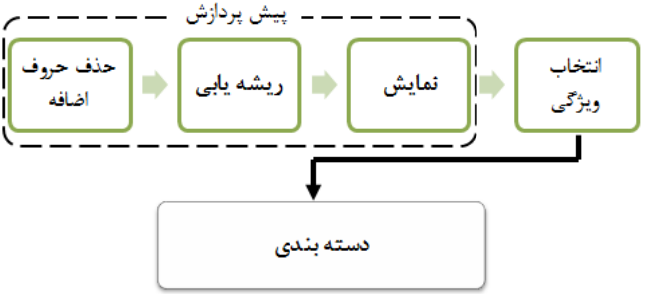
دسته‌بندی به‌عنوان روشی برای فیلترینگ هرزنامه‌ها

مراحل یک سیستم دسته‌بند هرزنامه، در شکل ۱ مشاهده می‌شود. همان‌طور که در این شکل دیده می‌شود، عملیات فیلترینگ شامل سه فرایند کلی پیش‌پردازش، انتخاب ویژگی و دسته‌بندی است که در ادامه هر یک از این سه فرایند بررسی می‌شوند.

پیش‌پردازش

همان‌طور که در شکل ۱ نشان داده شده است، قبل از انتخاب ویژگی بایستی روی نمونه‌های آموزشی، سه مرحله که به آنها پیش‌پردازش[1] گفته می‌شود، انجام شود. هدف اصلی پیش‌پردازش، حذف داده‌هایی است که دربارهٔ کلاس یک پیام هیچ اطلاعات مفیدی ندارند. همچنین یکی دیگر از اهداف پیش‌پردازش حذف افزونگی[2] است. این سه مرحله عبارت‌اند از:

- **حذف حروف اضافه بی‌تأثیر**[3]: شامل لیستی از کلمات است که در فرایند انتخاب ویژگی هیچ نقشی ندارند مانند ‘for’, ‘the’, ‘as’, ‘a’ و ... و حذف می‌شوند.
- **ریشه‌یابی کلمات**[4]: این مرحله عبارت است از برگرداندن کلمات به ریشهٔ اصلی خود، زیرا معمولاً بهتر است که فقط ریشهٔ کلمات استفاده شوند.
- **نمایش داده:** در این مرحله هر پیام ایمیل به‌صورت برداری از ویژگی‌ها (که در اینجا ویژگی‌ها همان کلمات استخراج‌شده از متن پیام‌ها هستند)، نمایش داده خواهد شد.



شکل ۱. نمایش مراحل اصلی در سیستم‌های دسته‌بندی هرزنامه

1. Pre-processing
2. Redundancy
3. Stop terms
4. Word stemming

انتخاب ویژگی

پس از اینکه اطلاعات متن‌های اولیهٔ ایمیل‌ها پالایش شده و اطلاعات غیرضروری تا حدودی حذف شدند، در مرحلهٔ بعد می‌توان به انتخاب ویژگی‌های برتر اقدام کرد. این کار معمولاً به‌وسیلهٔ یک سری الگوریتم انتخاب ویژگی[1] انجام می‌شود. هدف الگوریتم انتخاب ویژگی این است که ویژگی‌هایی را انتخاب کند که دارای اطلاعات بیشتری بوده و بتوانند نرخ خطای دسته‌بندی را تا حد ممکن کاهش دهند و نرخ پیش‌بینی آن را بهبود بخشند (Amini & Guiping, 2021).

انتخاب ویژگی در کلیهٔ مسائل دسته‌بندی[2] از جمله تشخیص هرزنامه، یکی از موارد مهم به شمار می‌رود. اینکه چه مجموعه‌ای از ویژگی‌ها انتخاب شوند و تعداد این ویژگی‌ها چقدر باشد، یکی از مسائل مهم در مسائل دسته‌بندی به شمار می‌رود. در مسئلهٔ تشخیص هرزنامه، ویژگی‌ها همان کلمات یا لغات هستند که دسته‌بندی پیام‌های رسیده اغلب بر اساس آنها انجام می‌شود. به‌دلیل اینکه ممکن است تعداد ویژگی‌هایی که بتوان بر اساس آنها به دسته‌بندی پیام‌ها اقدام کرد، بسیار زیاد شود، بایستی با استفاده از یک الگوریتم کارآمد، ویژگی‌های برتر را برای عملیات دسته‌بندی انتخاب کرد. تاکنون برای انتخاب ویژگی، الگوریتم‌های مختلفی ارائه شده است و اکثر این الگوریتم‌ها سعی می‌کنند ویژگی‌هایی را انتخاب کنند که قادر باشند بهتر به دسته‌بندی کلاس‌ها اقدام کنند.

یکی از روش‌های سادهٔ انتخاب ویژگی، تکرار سند (DF)[3] است (Yang et al, 1997) که عبارت است از تعداد سندهایی که آن ویژگی در آنها تکرار شده است. روش کار به این صوت است که ابتدا میزان DF برای تمام ویژگی‌ها محاسبه می‌شود و در پایان، ویژگی‌هایی که مقدار DF آنها از یک مقدار آستانه کمتر باشد، حذف می‌شوند. دلیل انجام کار نیز این است که ویژگی‌هایی که به‌ندرت در سندها اتفاق می‌افتند، برای دسته‌بند ما اطلاعات مفیدی ندارند. روش‌های دیگر انتخاب ویژگی، اطلاعات متقابل (MI)[4] (Zorkadis et al, 2005) و نفع اطلاعاتی (IG)[5] (Mitchell, 1996) هستند. روش دیگر TF-IDF است که از TF (تکرار ویژگی[6]) ضرب در IDF (معکوس تکرار سند[7]) به دست می‌آید (Wu, 2009).

در تمامی روش‌های گفته‌شده سعی شده تا بر اساس پارامتری خاصی به گزینش ویژگی‌های برتر اقدام شود. همین موضوع باعث می‌شود که هرزنامه‌نویس‌ها نیز به شناسایی این ویژگی‌ها پرداخته و بتوانند به‌راحتی با آن مقابله کنند. به همین دلیل، روش‌های دیگر انتخاب ویژگی وجود دارند که به‌جای انتخاب ویژگی‌هایی برتر، به‌صورت تصادفی ویژگی‌ها را انتخاب می‌کنند (DeBarr et al, 2012). این کار باعث افزایش مقاومت[8] مدل نسبت به نویز می‌شود. برای انجام این کار، برای تبدیل بردار ورودی به بردار خروجی، از یک ماتریس با مقادیر تصادفی استفاده می‌شود. بیشتر مواقع برای تولید اعداد تصادفی از

1. Feature detection algorithms
2. Classification
3. Document frequency
4. Mutual information
5. Information gain
6. Term frequency
7. Inverse document frequency
8. Robust

توزیع گاوسی استفاده می‌شود. سپس، جمع تمام مقادیر یک ستون بایستی برابر با یک (نرمالیزه) شود. بردارهای ورودی (همان بردار ویژگی‌ها) در این ماتریس ضرب شده و فضای بردار جدید را نتیجه می‌دهند. این کار در واقع، همان ترکیب خطی ویژگی‌ها است.

فرض کنید n نمونهٔ آموزشی (پیام) وجود دارد که بایستی عملیات انتخاب ویژگی روی آنها اعمال شود. ابتدا پس از پیش‌پردازش، یک بردار از تمامی m ویژگی اولیه به شکل $F=\{f_1, f_2, f_3,.., f_m\}$ ساخته می‌شود که در آن f_j نشان‌دهندهٔ ویژگی (کلمه) jام است. سپس، هر پیام i به یک بردار منطقی به‌صورت $V_i=\{v_{i1},v_{i2}, \dots ,v_{im}\}$ تبدیل می‌شود که در آن اگر v_{ij} برابر تعداد تکرار ویژگی f_j در پیام iام است، بنابراین برای مجموعهٔ تمامی پیام‌های آموزشی، ماتریسی به‌صورت زیر خواهیم داشت:

$$\text{Input} = \begin{bmatrix} v_{11}v_{12}v_{13} & \cdots & v_{1m} \\ \vdots & \ddots & \vdots \\ v_{n1}v_{n2}v_{n3} & \cdots & v_{nm} \end{bmatrix}$$

این ماتریس حاوی ویژگی مد نظر ورودی است. حال، می‌توان با ضرب کردن این ماتریس در یک ماتریس انتقال تعداد این ویژگی‌ها را کاهش داده و عملیات انتخاب ویژگی را انجام داد. ماتریس انتقال به‌صورت زیر خواهد بود:

$$M_{\text{Transform}} = \begin{bmatrix} t_{11}t_{12}t_{13} & \cdots & t_{1s} \\ \vdots & \ddots & \vdots \\ t_{m1}t_{m2}t_{m3} & \cdots & t_{ms} \end{bmatrix}$$

همان‌طور که در بالا مشاهده می‌کنید، تعداد سطرهای این ماتریس برابر تعداد ویژگی موجود در ماتریس ورودی (m) است و تعداد ستون‌ها (s) نیز برابر تعداد ویژگی‌های کاهش‌یافتهٔ پایانی است که مقدار آن توسط کاربر تعیین می‌شود. در واقع، با تنظیم مقدار s با هر عدد دلخواه می‌توان طول بردار نهایی ویژگی را تعیین کرد. این عملیات ضرب را می‌توان در زیر مشاهده کرد:

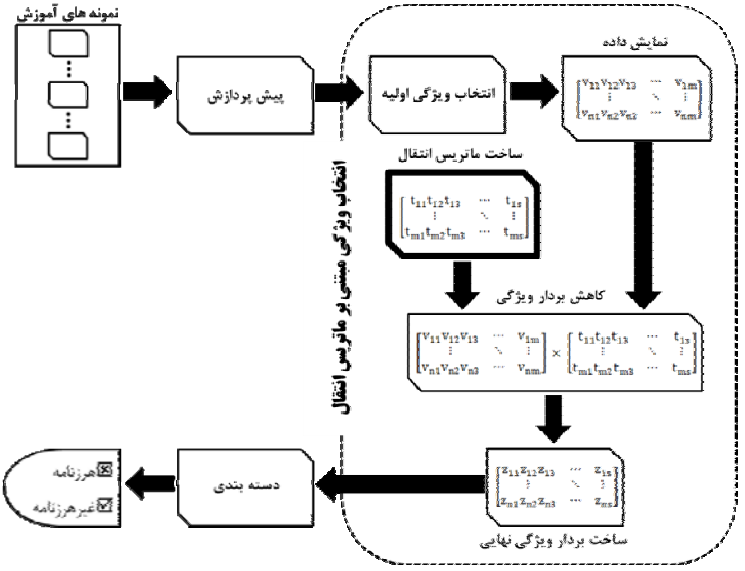
$$\begin{bmatrix} v_{11}v_{12}v_{13} & \cdots & v_{1m} \\ \vdots & \ddots & \vdots \\ v_{n1}v_{n2}v_{n3} & \cdots & v_{nm} \end{bmatrix} \times \begin{bmatrix} t_{11}t_{12}t_{13} & \cdots & t_{1s} \\ \vdots & \ddots & \vdots \\ t_{m1}t_{m2}t_{m3} & \cdots & t_{ms} \end{bmatrix} = \begin{bmatrix} z_{11}z_{12}z_{13} & \cdots & z_{1s} \\ \vdots & \ddots & \vdots \\ z_{n1}z_{n2}z_{n3} & \cdots & z_{ns} \end{bmatrix}$$

این موضوع باعث می‌شود که هر ویژگی نهایی از ترکیب خطی ویژگی‌های اولیه به دست آید. بنابراین، ماتریس انتقال در روند انتخاب ویژگی‌های قدرتمند و بهبود عملکرد دسته‌بندی نقش بسیار مهمی دارد و تمرکز اصلی پژوهش حاضر بر ساخت این ماتریس است. هدف روش پیشنهادی این است که این ماتریس انتقال را بر اساس الگوریتم ژنتیک و طی چند مرحلهٔ آموزش سیستم به‌نحوی بسازد که مقدار صحت سیستم به حداکثر مقدار خود برسد. شکل ۱ فرایند پایه‌ای سیستم دسته‌بندی منفرد مبتنی بر انتخاب ویژگی با استفاده از ماتریس انتقال را نشان می‌دهد که در آن مرحلهٔ ساخت ماتریس انتقال با خط چین نشان داده شده است. روند کار به این صورت است که ابتدا عملیات پیش‌پردازش روی نمونه‌های ورودی انجام می‌گیرد. به‌دلیل اینکه ممکن است تعداد ویژگی‌ها بسیار زیاد باشد و به یک ماتریس انتقال بزرگ و محاسبات زیاد منجر شود، در ابتدا بایستی روی نمونه‌های آموزشی یک انتخاب

ویژگی اولیه انجام شده و تعداد کمتری از ویژگی‌ها انتخاب شوند. سپس، با ساخت یک ماتریس انتقال و ضرب کردن در ویژگی‌های اولیهٔ بردار ویژگی نهایی ساخته می‌شود. در انتها، عملیات دسته‌بندی بر اساس بردار ویژگی ساخته‌شده انجام می‌شود.

دسته‌بندی

پس از ساخت بردار ویژگی معادل هر پیام ایمیل، در مرحلهٔ بعد کافی است به یک الگوریتم دسته‌بند تحویل داده شود تا روی آنها عملیات دسته‌بندی انجام شود. در این مقاله، برای عملیات دسته‌بندی، از بیزین ساده استفاده شده است. اگرچه تاکنون، برای فیلترینگ هرزنامه‌ها، الگوریتم‌های یادگیری ماشینی فراوانی ارائه و حتی الگوریتم‌هایی مثل بوستینگ[1] (Freund et al, 1999) و ماشین بردار پشتیبان[2] (Drucker et al, 1999) نسبت به بیزین در زمینهٔ دسته‌بندی متون دارای کارایی بالاتری هستند، اما از بیزین به‌طور خاص در کاربردهای تجاری و متن‌باز[3] استفادهٔ بسیاری می‌شود (Metsis et al, 2006). دلیل این موضوع شاید سادگی آن باشد که پیاده‌سازی آن را راحت می‌کند. پیچیدگی اجرایی خطی و دقت به‌نسبت بالای آن نیز باعث می‌شود که بتوان آن را با سایر الگوریتم‌های یادگیری ماشین مقایسه کرد (Androutsopoulos et al, 2004).



شکل ۲. انتخاب ویژگی مبتنی بر ماتریس انتقال در یک سیستم دسته‌بندی منفرد

1. Boosting
2. SVM
3. Open-source

بر اساس تئوری بیزین، احتمال اینکه یک پیام با بردار ویژگی $\vec{x} = \langle x_1, \dots, x_m \rangle$ متعلق به کلاس c باشد، برابر است با (Metsis et al, 2006):

$$p(c|\vec{x}) = \frac{p(c).p(\vec{x}|c)}{p(\vec{x})}$$ رابطهٔ ۱)

که در این رابطهٔ $p(c)$ برابر احتمال پیشین[1] کلاس c، $p(\vec{x}|c)$ برابر است با احتمال اینکه $\vec{x}$ در کلاس c ظاهر شود و مقدار $p(\vec{x})$ نیز به‌صورت زیر محاسبه می‌شود:

$$p(c_s).p(\vec{x}|c_s) + p(c_h).p(\vec{x}|c_h)$$ رابطهٔ ۲)

که c_s کلاس هرزنامه و c_h کلاس غیرهرزنامه است.

از آنجا که عبارت مخرج برای هر دو کلاس برابر است، بنابراین پیام متعلق به کلاسی است که به‌ازای آن کلاس مقدار $p(c).p(\vec{x}|c)$ ماکزیمم شود. در واقع، $p(c)$ که برابر احتمال پیش‌بینی کلاس برای هر یک از کلاس‌های موجود است (در اینجا دو کلاس هرزنامه و غیرهرزنامه) و از تقسیم تعداد اعضای این کلاس در مجموعهٔ آموزش به کل مجموعهٔ آموزش به دست می‌آید.

روش بیز به‌کاررفته در پژوهش حاضر همان نیو بیز[2] است که با فرض استقلال ویژگی‌ها (در اینجا کلمات) استفاده می‌شود. هرچند احتمال دارد در اینجا کلماتی که در هرزنامه‌ها ظاهر می‌شوند به‌طور کامل از هم مستقل نباشند، اما دومینگوس و پازانی[3] (۱۹۹۶) نشان دادند تا زمانی که بتوان به‌طور صحیح احتمالات شرطی کلاس را مرتب کرد، وابستگی بین ویژگی‌ها بر عملکرد دسته‌بند تأثیری نخواهد داشت. اگرچه کلیت روش بیزین یک اصل ثابت است، اما تاکنون برای تعیین مقدار $p(\vec{x}|c)$ روش‌های مختلفی ارائه شده که در اکثر مقالات به این مسئله اشاره‌ای نشده است. اینکه مقدار $p(\vec{x}|c)$ را به چه روشی تعیین کنیم، ممکن است عملکرد الگوریتم دسته‌بند را نیز تحت تأثیر قرار دهد (Metsis et al, 2006). در این مقاله از نسخه‌ای از بیزین با عنوان روش «بیزین چندجمله‌ای با TF»[4] استفاده می‌شود.

در این روش فرض می‌شود که یک پیام با بردار ویژگی به شکل $x = \langle x_1, \dots, x_m \rangle$ است که در آن هر x_i نشان‌دهندهٔ میزان تکرار t_i در پیام مد نظر است. سپس هرزنامه بودن آن پیام بر اساس رابطهٔ ۳ تعیین می‌شود:

$$\frac{p(c_s).\prod_{i=1}^{m} p(t_i|c)^{x_i}}{\sum_{c\in\{c_s,c_h\}} p(c).\prod_{i=1}^{m} p(t_i|c)^{x_i}} > T$$ رابطهٔ ۳)

1. Priori probabilities
2. Naive Bayes
3. Domingos & Pazzani
4. Multinomial NB, TF attributes

در عبارت بالا مقدار $p(t|c)$ به‌صورت زیر محاسبه می‌شود:

رابطهٔ ۴)
$$p(t|c) = \frac{1 + N_{t,c}}{m + N_c}$$

که در آن $N_{t,c}$ برابر تعداد رخداد توکن t در پیام‌های مجموعه آموزشی کلاس c و $N_c = \sum_{i=1}^{m} N_{t_i,c}$ است.

الگوریتم ژنتیک

الگوریتم ژنتیک، یکی از متدولوژی‌های جست‌وجوی بهینه بر اساس اصول انتخاب طبیعی داروین است. در واقع، این الگوریتم نگاشتی از یک سیر طبیعی است که به‌صورت مسئلهٔ ریاضی فرموله شده و همین الهام گرفتن از طبیعت باعث شده است که در حل مسائل بهینه‌سازی و به‌خصوص مسئلهٔ جست‌وجو کارایی بالایی داشته باشد. ایدهٔ اصلی این الگوریتم این است که خصوصیات موروثی توسط ژن[1] از نسلی به نسل بعد منتقل می‌شوند. کارایی این الگوریتم در فضاهای جست‌وجوی وسیع مناسب بوده و در مقایسه با سایر الگوریتم‌ها احتمال کمتری دارد که در بهینهٔ محلی گرفتار شود.

روش کار در این الگوریتم به این صورت است که طی چندین دوره[2] پردازش کروموزوم‌ها و تولید جمعیت‌های پی‌درپی جواب بهینه را به دست می‌آورند. در ابتدا، مجموعه‌ای از راه‌حل‌های یک مسئله با عنوان یک جمعیت[3] اولیه تشکیل می‌شود. در واقع، در این مرحله مسئلهٔ مد نظر به‌صورت ریاضی فرموله شده و ساختار هر کروموزوم تعیین می‌شود. در مرحلهٔ بعد، کیفیت و ارزش تمام اعضای جمعیت که کروموزوم[4] نامیده می‌شوند، توسط یک تابع ارزش[5] محاسبه شده و کروموزوم‌ها بر اساس ارزش به‌دست‌آمده مرتب می‌شوند. در مرحلهٔ بعد بایستی تعدادی از کروموزوم‌ها برای ترکیب و تولید جمعیت بعدی انتخاب شوند. بعد از مشخص شدن کروموزوم‌هایی که بایستی با هم ترکیب شوند، عملگر ادغام[6] و جهش[7] روی آنها انجام شده و کروموزوم‌های جدید به دست می‌آیند. عملگرهای جهش و ادغام، عملگرهای اصلی الگوریتم ژنتیک به شمار می‌روند. مهم‌ترین عملگر در الگوریتم ژنتیک، عملگر ادغام است که باعث جست‌وجوی کامل فضای جست‌وجو شده و در این فرایند دو کروموزوم برای بالا بردن ارزش خود به تبادل ژن‌های خود اقدام می‌کنند. عملگر جهش نیز می‌تواند از گیر افتادن کروموزوم‌ها در نقطهٔ بهینهٔ محلی جلوگیری به عمل آورد که برای پیاده‌سازی آن روش‌های متفاوتی وجود دارد. با استراتژی نخبه‌گزینی، نسل جدید جایگزین نسل قبل شده و جمعیت جدید را شکل می‌دهد. فرایند پردازش تا زمان ارضا شدن یک شرط خاص ادامه پیدا می‌کند (Goldberg et al, 1989; Davis et al,

1. Genome
2. Iterative
3. Population
4. Chromosomes
5. Fitness
6. Crossover
7. Mutation

1991 & Stanimirović, 2012). این روند چندین دوره اجرا شده و برترین کروموزوم‌ها به‌عنوان جواب نهایی برگردانده می‌شوند.

روش پیشنهادی برای انتخاب ویژگی و دسته‌بندی جمعی

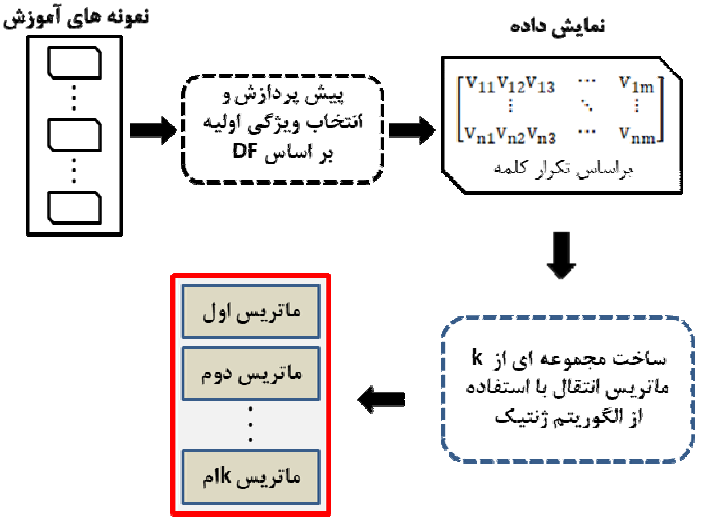
روش پیشنهادی در این مقاله به‌عنوان یک راهکار مبتنی بر یادگیری ماشین با ناظر، دارای دو فاز آموزش (یادگیری) و آزمون (تست) است. در فاز آموزش، الگوریتم بر اساس نمونه‌های آموزشی یاد می‌گیرد که چگونه به دسته‌بندی نمونه‌های موجود اقدام کند و در فاز آزمون، الگوریتم برای دسته‌بندی نمونه‌های آزمایش به‌کار برده می‌شود. بدین ترتیب، میزان یادگیری الگوریتم امتحان و بررسی شده و کارایی آن مشخص می‌شود. در ادامه، روند هر یک از دو فاز نام‌برده در روش پیشنهادی تشریح خواهد شد.

فاز آموزش در روش پیشنهادی

در این فاز، قوانینی که بایستی فرایند یادگیری بر اساس آنها انجام گیرد، ساخته خواهند شد. در شکل ۳ فاز آموزش به‌صورت کلی نمایش داده شده است. بر اساس این شکل، فاز آموزش شامل دو مرحله است که عبارت‌اند از:

- پیش‌پردازش و انتخاب ویژگی اولیه
- ساخت مجموعه‌ای از k ماتریس انتقال با استفاده از الگوریتم ژنتیک

بر اساس شکل ۳ هر یک از این مراحل دارای یک خروجی مربوط به خود هستند. خروجی مرحلهٔ انتخاب ویژگی مجموعه‌ای از بردار ویژگی‌های اولیه در قالب یک ماتریس ورودی است. خروجی مرحلهٔ ساخت ماتریس نیز مجموعه‌ای از ماتریس‌های انتقال است. در ادامه، هر یک از این مراحل به‌تفصیل بیان خواهند شد.



شکل ۳. فاز آموزش در روش پیشنهادی

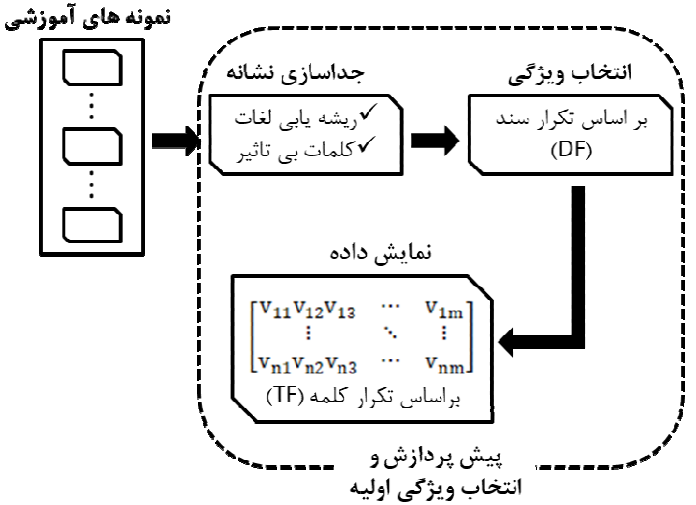
پیش‌پردازش و انتخاب ویژگی اولیه

روند کلی این مرحله در شکل ۴ نشان داده شده است. همان‌طور که در این شکل مشاهده می‌شود، این روند شامل سه زیرفعالیت است که عبارت‌اند از:

۱. **جداسازی نشانه‌ها و استخراج تمام ویژگی‌های موجود:** در این مرحله پس از عملیات پیش‌پردازش (که خود شامل ریشه‌یابی لغات و کلمات بی‌تأثیر و غیره است) پایگاه داده اولیه‌ای از تمام ویژگی‌های موجود تشکیل می‌شود.

۲. **انتخاب ویژگی اولیه:** در این مرحله بر اساس الگوریتم تکرار سند (DF) از بین تمام ویژگی موجود در پایگاه دادهٔ اولیه مجموعه‌ای از ویژگی برتر انتخاب می‌شوند.

۳. **نمایش داده:** در این مرحله تمام نمونه‌های آموزشی (ایمیل‌ها) بر اساس بردار ویژگی خود بیان می‌شوند.



شکل ۴. عملیات پیش‌پردازش و انتخاب ویژگی اولیه

طی گام‌های مطرح‌شده، نمونه‌های آموزشی از شکل تعدادی متن به یک ماتریس عددی تبدیل می‌شوند که تعداد سطرهای آن برابر تعداد نمونه‌ها و تعداد ستون‌های آن برابر تعداد ویژگی‌ها خواهد بود.

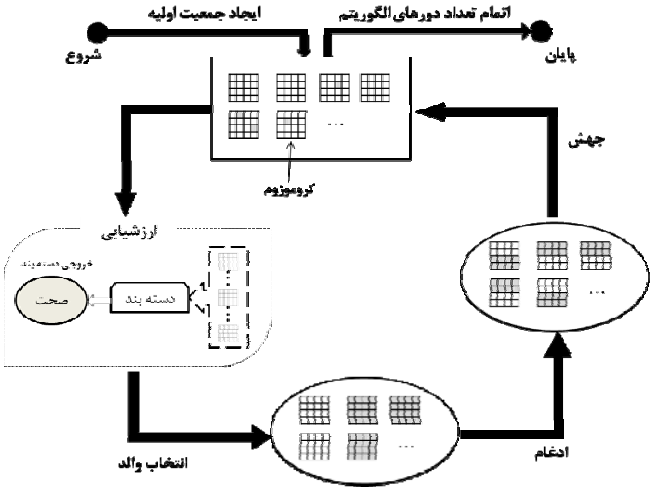
ساخت ماتریس انتقال با استفاده از الگوریتم ژنتیک

ورودی این مرحله، ماتریسی است که در مرحلهٔ انتخاب ویژگی اولیه به دست آمده بود. الگوریتم ژنتیک وظیفه دارد که بهینه‌ترین ماتریس انتقال را تولید کند. روند کار الگوریتم ژنتیک در شکل ۵ نشان داده شده است. چرخهٔ این الگوریتم دارای مراحل مختلفی است و در هر مرحله فرایند خاصی انجام می‌شود و

بدین ترتیب امکان ادامهٔ چرخه مقدور می‌شود. این مراحل عبارت‌اند از: ۱. ایجاد نسل اولیه؛ ۲. تابع ارزش؛ ۳. انتخاب؛ ۴. ادغام؛ ۵. جهش. در ادامه، هر یک از این مرحله‌ها به‌تفصیل بیان می‌شوند.

تولید جمعیت اولیهٔ کروموزوم‌ها

نخستین مرحله، ایجاد نسل اولیه‌ای از کروموزوم‌ها است. همان‌طور که گفته شد، کروموزوم‌ها در الگوریتم ژنتیک همان راه‌حل‌های بالقوهٔ مسئله هستند. در اینجا هدف به دست آوردن بهینه‌ترین ماتریس انتقال است، بنابراین کروموزوم‌ها همان ماتریس‌های انتقال هستند. تعداد سطرهای این ماتریس انتقال برابر تعداد ستون‌های ماتریس ویژگی ورودی و تعداد ستون‌های آن توسط کاربر تعیین می‌شود و تعیین‌کنندهٔ طول بردار ویژگی نهایی خواهد بود. مقدار المان‌های این ماتریس نیز بر اساس توزیع گوسی[1] به‌صورت تصادفی مقداردهی می‌شوند. از آنجا که این اعداد تصادفی هستند، ممکن است شرط لازم برای این ماتریس که بایستی جمع المان‌های هر ستون برابر ۱ باشد رعایت نشود. به همین دلیل، بعد از اختصاص مقادیر تصادفی به المان‌های این ماتریس بایستی روی تمامی ستون‌های این ماتریس، عملیات نرمالیزه کردن را اعمال کرد. عملیات نرمالیزه کردن نیز بسیار ساده است و از تقسیم هر المان بر جمع مقادیر تمامی المان‌های ستونی که در آن قرار گرفته است، به دست می‌آید.



شکل ۵. فرایند ساخت ماتریس انتقال توسط الگوریتم ژنتیک

تابع ارزش

در مرحلهٔ بعد، بایستی کروموزوم‌ها بر اساس تابع ارزش مرتب شوند. تابع ارزش یکی از قسمت‌های مهم الگوریتم ژنتیک به شمار می‌رود و کارایی الگوریتم تا حدود زیادی در گروی انتخاب تابع ارزش مناسب

1. Gaussian

است، بنابراین باید این تابع طوری انتخاب شود که باعث افزایش کارایی الگوریتم شود. در روش پیشنهادی، الگوریتم ژنتیک در تعامل با دسته‌بند سعی در به دست آوردن ماتریسی دارد که بتواند بالاترین بازدهی را در دسته‌بندی پیام‌ها داشته باشد، بنابراین در روش پیشنهادی، الگوریتم ژنتیک محاسبهٔ ارزش هر کروموزوم (ماتریس انتقال) را به الگوریتم دسته‌بند می‌سپارد و این الگوریتم دسته‌بند است که تعیین می‌کند کدام ماتریس برای دسته‌بندی پیام‌ها دارای ارزش بالاتری است. این ویژگی الگوریتم‌های رپر باعث می‌شود تا الگوریتم دسته‌بند نیز در فرایند انتخاب ویژگی تأثیر داشته باشد و بتواند برای خود ویژگی‌های کاراتر را انتخاب کند. تابع ارزش در نظر گرفته‌شده در الگوریتم ژنتیک در روش پیشنهادی پارامتر صحت است که به‌صورت زیر تعریف می‌شود (Androutsopoulos et al, 2000):

$$\text{صحت} = \frac{\text{تعداد کل پیام‌هایی که به‌درستی دسته‌بندی شده‌اند}}{\text{تعداد کل پیام‌های مجموعه آموزش}} \times ۱۰۰ \qquad \text{رابطهٔ ۵)}$$

الگوریتم ژنتیک، هر کروموزوم را به الگوریتم دسته‌بند داده و برحسب اینکه بر اساس این ماتریس انتقال میزان صحت دسته‌بند چقدر است، به مرتب‌سازی کروموزوم‌ها اقدام می‌کند. بر این اساس، بیشترین ارزش به کروموزومی (ماتریس انتقال) متعلق است که تابع دسته‌بند به‌ازای آن بیشترین مقدار صحت را داشته باشد.

انتخاب (والد)

در این قسمت باید بررسی شود که چه کروموزوم‌هایی بایستی در ساخت نسل بعدی دخیل باشند. در پژوهش حاضر، عملیات انتخاب بر اساس چرخ رولت[1] و بر پایهٔ تقاطع یکسان انجام می‌شود. همچنین نرخ انتخاب والد (بازترکیبی) برابر با $p_c = ۰/۸$ تنظیم شد.

ادغام (ترکیب)

در اینجا چون کروموزوم‌ها ماتریسی‌شکل و دوبعدی هستند، برای ترکیب دو کروموزوم نسبت به حالت تک‌بعدی، راه‌های بیشتری وجود دارد. ترکیب در نظر گرفته‌شده روش سنتی پخش است که به‌صورت یگانه پیاده‌سازی شده است. در حالت یگانه هر کروموزوم به دو قسمت شکسته شده و تکهٔ اول کروموزوم اول با تکهٔ دوم کروموزوم دوم و تکهٔ دوم کروموزوم دوم با تکهٔ اول کروموزوم دوم ترکیب می‌شود. برای شکستن هر ماتریس (کروموزوم‌ها) بایستی دو مورد مشخص شود:

- ماتریس به چه شکل شکسته شود. به‌صورت عرضی، طولی، مورب و ...
- تعیین نقطه‌ای برای شکستن ماتریس

در اینجا ماتریس‌ها به‌صورت عرضی شکسته می‌شوند و نقطهٔ شکست وسط ماتریس در نظر گرفته شده است که باعث می‌شود اندازهٔ دو قسمت شکسته‌شده با هم برابر شود. در شکل ۶ می‌توان این

1. Roulette wheel

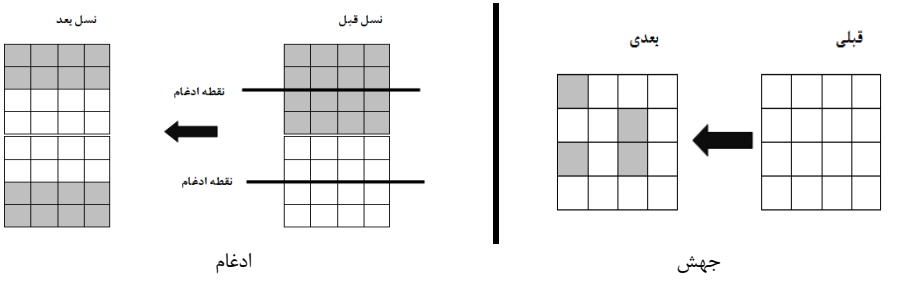
عملیات را به‌صورت گرافیکی مشاهده کرد. همان‌طور که در شکل مشاهده می‌شود، ماتریس‌ها به‌صورت عرضی و از وسط به دو نیم شکسته می‌شوند، سپس هر کروموزوم یک نیمهٔ خود را با کروموزوم دیگر تعویض کرده و تشکیل دو کروموزوم جدید را می‌دهند. بعد از تشکیل کروموزوم‌های جدید بایستی شرط نرمال بودن روی آنها بررسی شده و در صورت لزوم به نرمال کردن آنها اقدام شود.

جهش

در انتخاب یک ماتریس به‌عنوان کروموزوم، المان‌های این ماتریس ژن‌های آن به حساب خواهند آمد. بر این اساس، روش پیاده‌سازی جهش به این شکل است که برای تمام المان‌های ماتریس (کروموزوم) یک عدد تصادفی بین صفر و ۱ تولید شده و در صورتی که عدد تولیدشده کوچک‌تر از مقدار ۰/۰۰۱ بود مقدار آن عضو تغییر می‌کند. معادلهٔ تغییر ژن‌ها به‌صورت زیر است:

$$Gene\ [i,j]\ =\ (Gene\ [i,j]\ +\ 1.0)\ /\ 2.0$$ رابطهٔ ۶)

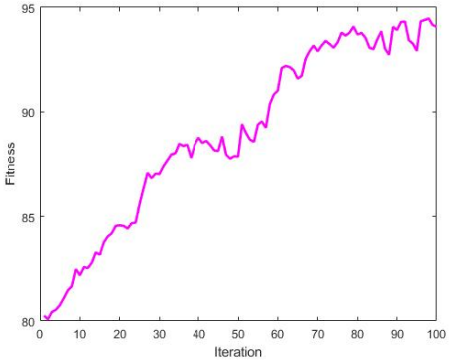
برای تغییر ژن‌ها هر عبارت دیگری را نیز می‌توان انتخاب کرد که به نظر کاربر بستگی دارد. پس از انجام عملیات جهش روی اعضا، به‌دلیل اینکه ممکن است جمع مقادیر یک ستون برابر ۱ نباشد، بایستی یک بار دیگر آن را نرمال کرد. در پژوهش حاضر، نرخ جهش p_m برابر با ۰/۲ مقداردهی شد.



شکل ۶. شماتیک عملگرهای جهش و ادغام ماتریس‌های انتقال در الگوریتم ژنتیک

در پایان اجرای الگوریتم ژنتیک، همانند شکل ۵ مجموعه‌ای از بهترین کروموزوم‌های آخرین نسل (k تا) برگردانده می‌شوند. این مجموعه k ماتریس حاصل فرایند آموزش سیستم هستند.

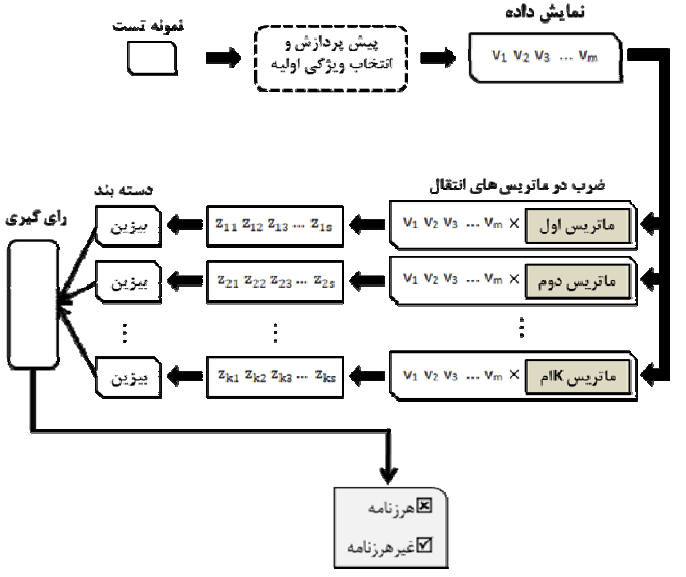
همان‌طور که در بالا گفته شد، پس از انجام عملگر جهش و ادغام، به نرمال کردن ستون‌های ماتریس انتقال اقدام می‌شود که موجب می‌شود اسکیماهای موجود در پاسخ‌های موفق دستخوش تغییر شده و ممکن است باعث گمراه شدن الگوریتم و عدم هم‌گرایی الگوریتم شود. بدین منظور، در شکل ۷ نمونه‌ای از تغییرات مقدار برازندگی (مقدار تابع ارزش) طی اجرای الگوریتم ژنتیک نشان داده شده است. همان‌طور که در این شکل مشاهده می‌شود، هم‌گام با پیشرفت الگوریتم، مقدار ارزش (Fitness) افزایش یافته و هم‌گرایی الگوریتم در این شکل مشهود است.



شکل ۷. نمونه‌ای از تغییرات مقدار تابع ارزش طی اجرای الگوریتم ژنتیک

فاز آزمون در روش پیشنهادی

کلیت فاز آزمون در روش پیشنهادی در شکل ۸ نمایش داده شده است. در این فاز، الگوریتم بر اساس آنچه در فاز آموزش فراگرفته است به دسته‌بندی پیام‌ها اقدام می‌کند، بنابراین در این مرحله هر سند ورودی (همانند فاز آموزش) به یک بردار ویژگی اولیه تبدیل می‌شود. سپس، به‌طور مجزا در k ماتریس انتقال (به‌دست‌آمده از فاز آموزش) ضرب شده و در نهایت، k بردار ویژگی نهایی را تولید می‌کند. همان‌طور که در شکل ۸ مشهود است، به‌ازای هر یک از این k ماتریس یک عملیات دسته‌بندی بر اساس دسته‌بند بیزین انجام شده و در نهایت، بین دسته‌بندهای مختلف رأی‌گیری شده و کلاسی که بیشترین رأی را داشته باشد به‌عنوان کلاس نمونهٔ جدید معرفی می‌شود.



شکل ۸. فاز آزمون در روش پیشنهادی

ارزیابی روش پیشنهادی

پایگاه دادهٔ استفاده‌شده

پایگاه داده‌های استفاده در این مقاله شامل PU۱ و PU۲ است[1] که مجموعه‌ای از ایمیل‌های شخصی هستند. به همین دلیل، برای اینکه محتوای آنها شخصی بماند، تمام کلمات با یک شناسهٔ عددی جایگزین شده‌اند. هر یک از این پایگاه داده‌ها شامل یازده زیردایرکتوری (part1, ..., part10, unused) هستند که طی آزمایش‌ها ده‌تای آنها استفاده می‌شوند. ارزیابی دسته‌بندی با یک رویهٔ اعتبارسنجی ده‌گانهٔ معروف اندازه‌گیری و گزارش شده است. در حقیقت در هر آزمایش، مجموعهٔ داده به ده بخش مجزا تقسیم می‌شود و به‌ازای هر بخش الگوریتم یک بار اجرا می‌شود. هر بار یک بخش مختلف به‌عنوان مجموعهٔ آزمایشی استفاده می‌شود و نُه بخش دیگر با یکدیگر گروه‌بندی شده و به‌عنوان مجموعهٔ آموزشی استفاده می‌شوند. پارامترهای دسته‌بندی (در مجموعه تست) در ده دور اجرا و سپس میانگین‌گیری شده و به‌عنوان صحت کل مجموعهٔ داده برگردانده می‌شود. PU1 شامل ۱۰۹۹ پیام است که ۴۴ درصد آنها هرزنامه هستند و PU۲ نیز شامل ۷۱۰ پیام است که ۲۰ درصد آنها هرزنامه هستند.

معیارهای کارایی

برای ارزیابی دسته‌بندهای هرزنامه معمولاً چندین پارامتر ارزیابی ارائه می‌شود (Mohammad et al. 2011). در پژوهش حاضر نیز برای ارزیابی روش پیشنهادی، از سه پارامتر مهم به‌صورت زیر استفاده می‌شود.

- معیار صحت

$$\text{Accuracy}\ (\%) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100$$ رابطهٔ ۷)

- معیار دقت

$$\text{Precision}\ (\%) = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100$$ رابطهٔ ۸)

- معیار بازخوانی

$$\text{Recall}(\%) = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100$$ رابطهٔ ۹)

که در آن داریم:

TN (منفی درست): برابر است با تعداد پیام‌هایی که دستهٔ واقعی آنها منفی بوده و الگوریتم دسته‌بندی آنها را به‌درستی منفی تشخیص داده است.

1. The PU corpora is available at http://www.csmining.org/index.php/cdmc-2013.html

TP (مثبت درست): برابر است با تعداد پیام‌هایی که دستهٔ واقعی آنها مثبت بوده و الگوریتم دسته‌بندی آنها را به‌درستی مثبت تشخیص داده است.

FP (مثبت اشتباه): برابر است با تعداد پیام‌هایی که دستهٔ واقعی آنها منفی بوده و الگوریتم دسته‌بندی آنها را به‌اشتباه مثبت تشخیص داده است.

FN (منفی اشتباه): برابر است با تعداد پیام‌هایی که دستهٔ واقعی آنها مثبت بوده و الگوریتم دسته‌بندی آنها را به‌اشتباه منفی تشخیص داده است.

ارزیابی

در این بخش، روش پیشنهادی ارزیابی می‌شود. قبل از بیان نتایج به‌دست‌آمده، تنظیمات روش پیشنهادی تشریح می‌شوند:

- در مرحلهٔ انتخاب ویژگی اولیه فقط ویژگی‌هایی انتخاب شدند که مقدار DF آنها از ۱۰ بیشتر باشد، یعنی در بیشتر از ده سند تکرار شده باشند.
- در تمامی روش‌ها شامل روش پیشنهادی، بیزین پایه و دو روش SVM و KNN، ویژگی‌های اولیه بر اساس DF مرتب شده و از بین آنها ویژگی‌های برتر انتخاب می‌شوند.
- در تمامی روش‌های پیشنهادی s (اندازهٔ ستون ماتریس انتقال) برابر با ۱۰ مقداردهی شده است.
- الگوریتم ژنتیک استفاده‌شده دارای ۱۰۰ کروموزوم بوده و در ۱۰۰ دور اجرا شد.

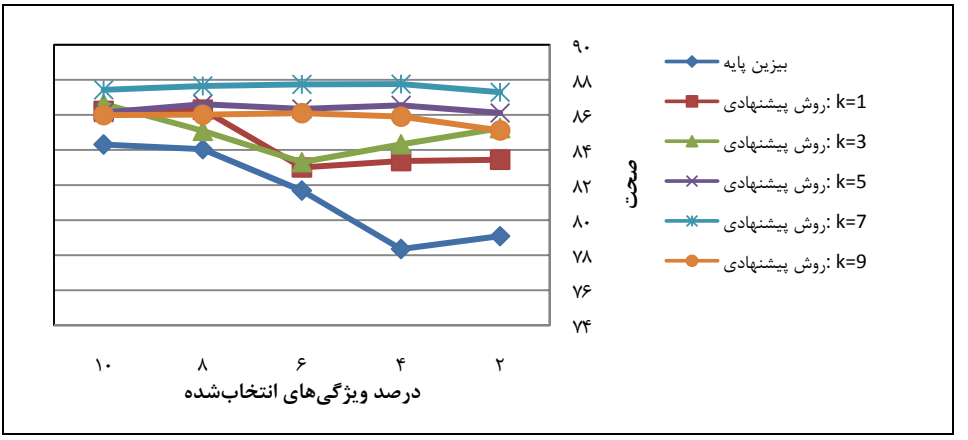
قبل از ارائهٔ نتایج پژوهش، نتایج به‌دست‌آمده از پیش‌پردازش داده‌ها بیان می‌شود. تعداد ویژگی‌های (کلمات) هر یک از پایگاه داده‌ها پیش از عملیات پیش‌پردازش و همچنین تعداد ویژگی‌های کاهش‌یافته پس از مرحلهٔ پیش پردازش، در جدول ۱ نشان داده شده است. همان‌طور که مشاهده می‌شود، تعداد کلماتی که در پیام‌های غیرهرزنامه ظاهر می‌شوند بیشتر از پیام‌هایی است که در هرزنامه‌ها وجود دارند. همچنین پس از اعمال پیش‌پردازش، یک بردار ویژگی یکسان به‌ازای کل پایگاه داده ایجاد می‌شود که اندازهٔ بردار ویژگی پایگاه دادهٔ PU۱ بزرگ‌تر از PU۲ است.

جدول ۱. تعداد ویژگی‌ها در مرحلهٔ پیش‌پردازش پایگاه داده‌ها

پایگاه داده	نوع کلاس	تعداد ویژگی‌ها	
		قبل از پیش‌پردازش	بعد از پیش‌پردازش
PU۱	هرزنامه	۱۲۲۱۳	۳۳۹۷
	غیرهرزنامه	۱۶۲۸۶	
	مجموع	۲۳۳۷۰	
PU۲	هرزنامه	۵۰۷۸	۱۸۶۸
	غیرهرزنامه	۱۳۱۵۳	
	مجموع	۱۵۳۳۸	

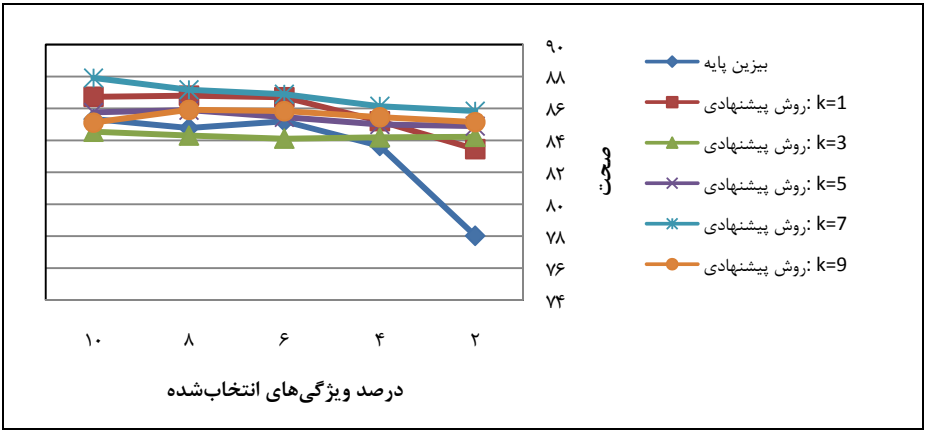
پس از مرحلهٔ پیش‌پردازش و ایجاد یک مجموعهٔ اولیه از ویژگی‌ها، در مرحلهٔ بعد ویژگی‌های اولیه بر اساس DF مرتب شده و به‌منظور اعمال روش پیشنهادی و فرایند دسته‌بندی زیرمجموعه‌ای از آنها انتخاب می‌شود. در پژوهش حاضر، در کلیهٔ روش‌ها از ۲ الی ۱۰ درصد از ویژگی‌های برتر انتخاب‌شده در مرحلهٔ پیش‌پردازش استفاده شده است (برای PU1 از بین کل ۳۳۹۷ ویژگی اولیه در مرحلهٔ پیش‌پردازش، به‌ازای ۲ درصد: ۶۹ ویژگی، ۴ درصد: ۱۳۶ ویژگی، ۶ درصد: ۲۰۴ ویژگی، ۸ درصد: ۲۷۲ ویژگی و ۱۰ درصد: ۳۴۰ ویژگی بر اساس معیار DF انتخاب شد، الگوریتم ژنتیک روی آنها اعمال شده و برای PU2 نیز به‌نحو مشابه اقدام شد).

روش‌های پیشنهادی به‌ازای ۵ مقدار مختلف k شامل ۱، ۳، ۵، ۷، ۹ ارزیابی شدند. دلایل انتخاب این اعداد این است که مشاهده شد، به‌ازای مقادیر k بالاتر از ۱۰ روش پیشنهادی دارای کارایی کاهشی است. نتایج مقایسه روش‌های پیشنهادی بر اساس سه پارامتر صحت، دقت و بازخوانی با روش بیزین پایه در دو پایگاه داده PU1 و PU2 در شکل‌های ۹ تا ۱۴ نشان داده شده‌اند.



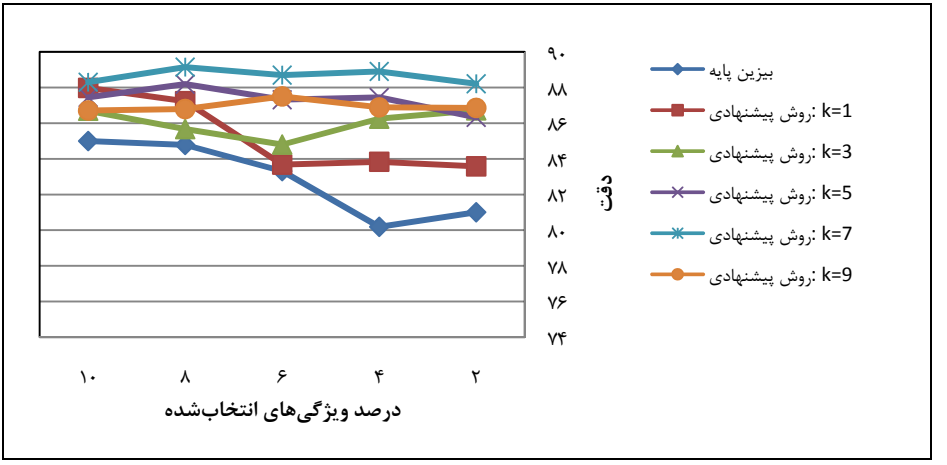
شکل ۹. نتایج پارامتر صحت (٪) روی پایگاه داده PU1

نتایج به‌دست‌آمده از مقایسهٔ میزان صحت روش‌ها در پایگاه داده‌های PU1 و PU2 در شکل‌های ۹ و ۱۰ نشان داده شده است. نتایج نشان می‌دهد که بر اساس پارامتر صحت، روش پیشنهادی (k=۷) در هر دو پایگاه داده دارای عملکرد بهتری است. در شکل ۹ مشاهده می‌شود که تمامی روش‌های پیشنهادی از بیزین پایه بهتر عمل کرده‌اند و در شکل ۱۰ مشهود است که روش بیزین پایه فقط از روش پیشنهادی (k=۳) بهتر بوده است و سایر روش‌ها همگی از بیزین پایه بهتر بوده‌اند. همچنین در روش‌های پیشنهادی با افزایش درصد ویژگی‌های انتخابی میزان صحت تغییر چندانی پیدا نمی‌کند، در حالی که در بیزین پایه با افزایش درصد ویژگی‌ها، عملکرد بهبود می‌یابد.

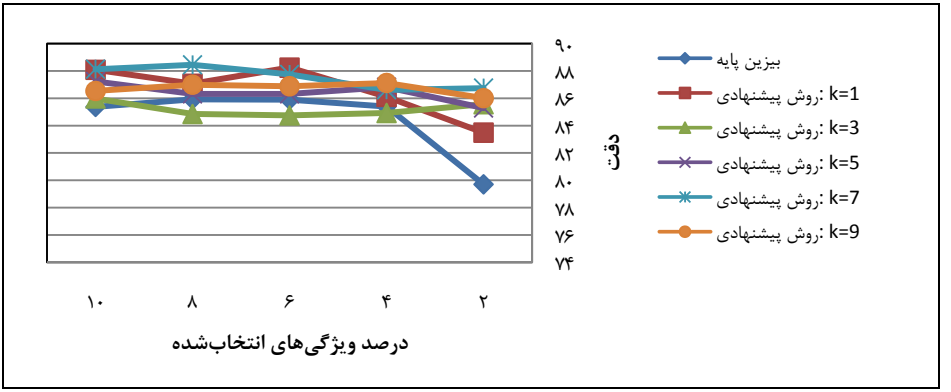


شکل ۱۰. نتایج پارامتر صحت (٪) روی پایگاه داده PU۱

نتایج به‌دست‌آمده از مقایسه میزان دقت روش‌ها در پایگاه داده‌های PU۱ و PU۲ در شکل‌های ۱۱ و ۱۲ نشان داده شده است. در اینجا نیز روش پیشنهادی (k = ۷) در بین تمامی روش‌های پیشنهادی بهترین کارایی را داشته است. در شکل ۱۱ مشاهده می‌شود که روش پیشنهادی (k = ۷) در پایگاه داده PU۱ دارای بهترین عملکرد است و بیزین پایهٔ بدترین خروجی را داشته است. در شکل ۱۲ فاصلهٔ بین روش‌ها کمتر است، با وجود این، تمامی روش‌های پیشنهادی به‌غیر از (k = ۹) از بیزین پایه بهتر عمل کرده‌اند.

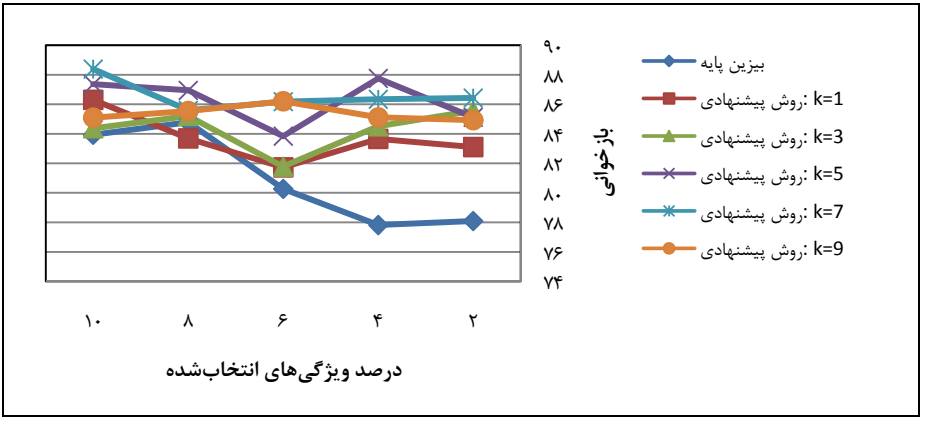


شکل ۱۱. نتایج پارامتر دقت (٪) روی پایگاه داده PU۱



شکل ۱۲. نتایج پارامتر دقت (٪) روی پایگاه داده PU۲

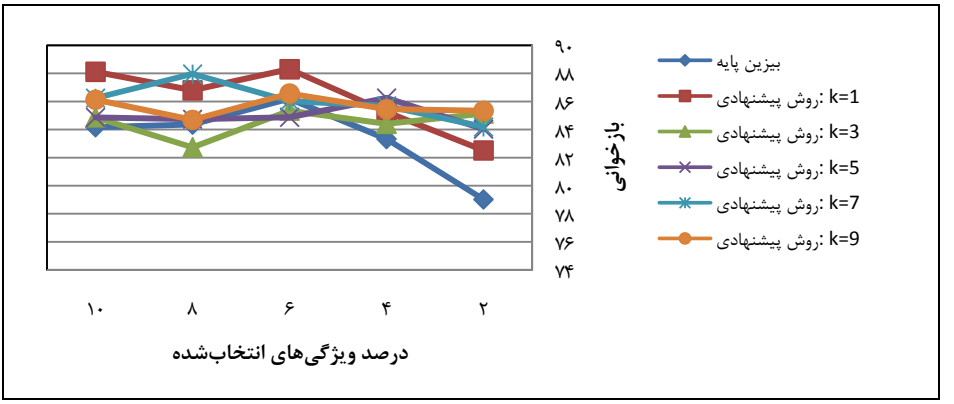
نتایج به‌دست‌آمده از مقایسهٔ میزان بازخوانی روش‌ها در پایگاه داده‌های PU۱ و PU۲ در شکل‌های ۱۳ و ۱۴ نشان داده شده است. اگرچه در این دو شکل روش‌های پیشنهادی به‌طور کلی از بیزین پایه بهتر عمل کرده‌اند، اما برخلاف پارامترهای قبلی، هیچ یک از روش‌های پیشنهادی به‌صورت مطلق در مقایسه با سایر روش‌ها، دارای عملکرد بهتری نبوده‌اند.



شکل ۱۳. نتایج پارامتر بازخوانی (٪) روی پایگاه داده PU۱

پس از مقایسهٔ روش‌های پیشنهادی با بیزین پایه و اثبات عملکرد بهتر آنها نسبت به روش بیزین پایه، در ادامه، عملکرد روش پیشنهادی با دو دسته‌بند پایهٔ مهم و پرکاربرد SVM و KNN مقایسه می‌شود. برای الگوریتم SVM نسخه‌های مختلفی وجود دارد که در اینجا از SVM خطی استفاده شده است. در هر دو دسته‌بند پایه از انتخاب ویژگی پایهٔ DF مشابه با روش‌های پیشنهادی استفاده شده است. نتایج بخش‌های قبل نشان داد که نتایج هر سه پارامتر صحت و دقت و بازخوانی روش‌های پیشنهادی

تقریباً دارای رفتار یکسانی است و روش پیشنهادی با مقدار k=۷ در مقایسه با سایر روش‌ها دارای عملکرد بهتری است، بنابراین در این بخش به مقایسهٔ پارامتر صحت روش پیشنهادی (k=۷) با دو دسته‌بند مهم پرداخته می‌شود که نتایج در جدول ۲ نشان داده شده است.



شکل ۱۴. نتایج پارامتر بازخوانی (٪) روی پایگاه داده PU۲

همان‌طور که در جدول ۲ مشاهده می‌شود، روش پیشنهادی در مقایسه با هر دو دسته‌بند دارای عملکرد بهتری است. اگرچه دسته‌بند SVM در مقایسه با KNN دارای نتایج بهتری است، اما هنوز هم در مقایسه با روش پیشنهادی دارای عملکرد پایین‌تری است.

جدول ۲. مقایسهٔ پارامتر صحت روش پیشنهادی (k = ۷) با دو دسته‌بند پایه

پایگاه داده	روش	درصد ویژگی‌های انتخابی				
		۲	۴	۶	۸	۱۰
PU۱	روش پیشنهادی (k=۷)	۸۷/۲۹	۸۷/۷۶	۸۷/۷۴	۸۷/۶۵	۸۷/۴۳
	SVM	۸۶/۰۱	۸۶/۴۴	۸۶/۲۳	۸۶/۶۶	۸۶/۸۹
	KNN	۸۴/۲۷	۸۴/۶۳	۸۴/۱۲	۸۴/۰۵	۸۴/۳۴
PU۲	روش پیشنهادی (k=۷)	۸۵/۸۳	۸۶/۱۴	۸۶/۸۹	۸۷/۱۷	۸۷/۹۱
	SVM	۸۵/۶۹	۸۵/۸۳	۸۵/۹۴	۸۶/۱۱	۸۶/۳۴
	KNN	۸۳/۷۴	۸۴/۱۷	۸۴/۶۹	۸۴/۳۸	۸۶/۵۵

بحث

در بخش قبل، پس از تجزیه‌وتحلیل روش پیشنهادی، عملکرد آن روی زیرمجموعه‌های مختلف ویژگی (۲ الی ۱۰ درصد) بررسی شد. مطالعهٔ پیشینهٔ پژوهش نشان می‌دهد که برخی پژوهشگران روی پایگاه داده‌های پژوهش حاضر، روش‌هایی ارائه داده‌اند که نتایج خود را بر اساس درصدهای مختلف ویژگی‌های انتخاب‌شده گزارش کرده‌اند. رزم آرا و همکاران[1] (۲۰۱۲) با استفاده از تمامی ویژگی‌ها (۱۰۰ درصد کل ویژگی‌ها) توانستند به نرخ صحت ۹۶/۲۶ و ۹۴/۲۳ در دو پایگاه دادهٔ یادشده دست یابند. مقایسهٔ پژوهش حاضر با این روش‌ها نشان می‌دهد که روش پیشنهادی با توجه به استفاده از درصد کمی از ویژگی‌ها (۲ الی ۱۰ درصد) توانسته است نتایج قابل قبولی کسب کند.

همچنین بر اساس نتایج پژوهش، در روش پیشنهادی، برخلاف روش بیزین ساده، با کاهش تعداد ویژگی‌ها کارایی افزایش می‌یابد که می‌تواند به‌عنوان یک مزیت برای الگوریتم دسته‌بند به شمار رود و می‌توان با استفاده از تعداد ویژگی‌های کمتر در مقایسه با بیزین پایه، به کارایی بالاتری دست یافت. دلیل کارآمدی روش پیشنهادی با تعداد ویژگی کم این است که انتخاب درصد کمتری از ویژگی‌ها باعث کاهش ابعاد ماتریس انتقال و در پی آن، کاهش فضای حل مسئلهٔ الگوریتم ژنتیک می‌شود. بدیهی است هرچقدر فضای جست‌وجو کوچک‌تر باشد، احتمال پیدا کردن جواب بهینه افزایش می‌یابد و همین موضوع باعث می‌شود که با کاهش تعداد ویژگی‌های ورودی، پاسخ‌های به‌دست‌آمده دارای کارایی بالاتر بوده و در نهایت، باعث افزایش عملکرد دسته‌بندی شوند.

استفاده از یک ترکیب خطی از ویژگی‌ها در روش پیشنهادی، باعث می‌شود که هرزنامه‌نویس‌ها نتوانند به‌راحتی ویژگی‌ها را شناسایی کرده و تغییر دهند. در واقع، در روش‌هایی که در آنها به‌دلیل حجم بسیار زیاد ویژگی‌ها از زیرمجموعه‌ای از ویژگی‌ها استفاده می‌شود، هرزنامه‌نویس‌ها ممکن است به شناسایی ویژگی‌های استفاده‌شده اقدام کرده و با تغییر یا حذف آنها سبب گمراهی و اشتباه سیستم دسته‌بند شوند. همچنین، استفاده از مجموعه‌ای از راه‌حل‌های برتر موجب استحکام و اعتمادپذیری بیشتر مدل دسته‌بندی می‌شود.

روش پیشنهادی به‌ازای مقادیر مختلف k بررسی، تجزیه و تحلیل شد. نتایج نشان داد که افزایش مقادیر k تا هر مقدار دل‌خواه به‌معنای افزایش کارایی دسته‌بندی نیست. به این معنا که با انتخاب تعداد بسیار زیاد ماتریس‌های انتقال برتر به‌عنوان راه‌حل‌های حاصل از آخرین گام الگوریتم ژنتیک، نمی‌توان به‌طور افزایشی کارایی را بهبود بخشید. بر اساس نتایج، افزایش تعداد ماتریس‌ها تا مقدار ۷ می‌تواند تأثیر مثبتی روی نتایج داشته باشد و پس از آن دیگر تغییرات مثبتی رخ نخواهد داد و حتی شاهد تأثیر منفی آن نیز خواهیم بود. دلیل این موضوع تا حدودی منطقی است. در الگوریتم ژنتیک راه‌حل‌ها بر اساس برازندگی مرتب شده و به‌طور کلی چند پاسخ برتر الگوریتم‌های تکاملی حاوی راه‌حل بهینه بوده و راه‌حل‌های بعدی دارای برازندگی کاهشی هستند. بنابراین، انتخاب تعداد مناسب راه‌حل‌های بهینهٔ مرحلهٔ آخر، مسئله‌ای مهم است که در پژوهش حاضر مقدار آن برابر با ۷ تعیین شد.

1. Razmara

نتیجه‌گیری

در این مقاله، برای شناسایی و تشخیص هرزنامه‌ها، روش نوینی ارائه شد. هدف روش پیشنهادی در این مقاله این بود که با ارائهٔ یک روش انتخاب ویژگی رپر مبتنی بر دو روش مهم و پرکاربرد شامل الگوریتم ژنتیک و دسته‌بند بیزین، در گام نخست ویژگی‌هایی که به‌نحو بهتری قادر به تفکیک پیام‌های هرزنامه هستند را استخراج کرده و با توجه به عملکرد پذیرفته‌شدهٔ دسته‌بندهای جمعی نسبت به دسته‌بندهای منفرد، در گام بعد با اعمال یک روش دسته‌بندی جمعی بیزین به دسته‌بندی آنها اقدام کند.

کارایی روش پیشنهادی روی دو پایگاه دادهٔ PU۱ و PU۲ بررسی شد. نتایج نشان داد که روش‌های پیشنهادی با مقادیر مختلف k در مقایسه با بیزین پایه، عملکرد بهتری دارند. همچنین بر اساس نتایج، روش پیشنهادی با مقدار k = ۷ در بین روش‌های پیشنهادی دارای بهترین کارایی بود. همچنین نتایج مقایسهٔ روش پیشنهادی (k = ۷) با دو دسته‌بند پرکاربرد SVM و KNN حاکی از برتری روش پیشنهادی بود. به‌عنوان کارهای آینده می‌توان سایر الگوریتم پردازش تکاملی را در ترکیب با سایر الگوریتم‌های دسته‌بندی همچون درخت تصمیم و روش‌های دسته‌بند جمعی به‌منظور انتخاب ویژگی به‌کار برد. همچنین می‌توان با تغییرات تابع ارزش الگوریتم ژنتیک و با انتخاب معیارهای بهینه‌تر از معیار صحت برای تابع ارزش، سعی در بهبود کارایی الگوریتم کرد.

فهرست منابع

Amini, F., and Hu, G. (2021). A two-layer feature selection method using genetic algorithm and elastic net. *Expert Systems with Applications*, 166: 114072.

Androutsopoulos, I., Koutsias, J., Chandrinos, K.V. and Spyropoulos, S.D. (2000). An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. *In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 160-167.

Androutsopoulos, I., Paliouras, G. and Michelakis, E. (2004). *Learning to filter unsolicited commercial e-mail*. NCSR "Demokritos" Technical Report, No. 2004/2.

Balamurugan, A.A., Rajaram, R., Pramala, S., Rajalakshmi, S., Jeyendran, C. and Surya Prakash, J.D. (2011). NB+: an improved naive Bayesian algorithm. *Knowledge-Based Systems*, 24(5), 563-569.

Chinavle, D., Kolari, P., Oates, T. and Finin, T. (2009). Ensembles in adversarial classification for spam. *In Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 2015-2018.

Crawford, E., Koprinska, I. and Patrick, J. (2004). Phrases and Feature Selection in E-Mail Classification. *Conference: ADCS 2004, Proceedings of the Ninth Australasian Document Computing Symposium*, December 13.

Dada, E. G., Bassi, G.S., Chiroma, H., Adetunmbi, A.O. and Ajibuwa, O.E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon,* 5(6), e01802.

Davis, L. (1991). *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold.

DeBarr, D. and Wechsler, H. (2012). Spam detection using random boost. *Pattern Recognition Letters*, 33(10), 1237-1244.

Domingos, P., and Pazzani, M. (1996). Beyond independence: Conditions for the optimality of the simple bayesian classier. *In Proc. 13th Intl. Conf. Machine Learning*. pp. 105-112.

Drucker, H., Wu, D. and Vapnik, V.N. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural networks,* 10(5), 1048-1054.

Faris, H., Al-Zoubi, A.M., Heidari, A.A., Aljarah, I., Mafarja, M., Hassonah, M.A. and Fujita, H. (2019). An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. *Information Fusion* 48: 67-83.

Freund, Y., Schapire, R. (1999). A short introduction to boosting. *Journal-Japanese Society for Artificial Intelligence*, 14(771-780), 1612.

Goldberg, D. E. (1989). Genetic algorithms in search. *Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.

Hu, Y., Guo, C., Ngai, E. W. T., Liu, M. and Chen, Sh. (2010). A scalable intelligent non-content-based spam-filtering framework. *Expert systems with applications*, 37(12), 8557-8565.

Huang, J., Cai, Y. and Xu, X. (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern recognition letters*, 28 (13), 1825-1844.

Jadhav, S., He, H. and Jenkins, K. (2018). Information gain directed genetic algorithm wrapper feature selection for credit rating. *Applied Soft Computing*, 69: 541-553.

Kohavi, R., and John, G.H. (1998). The wrapper approach. *In: Liu H., Motoda H. (eds) Feature Extraction, Construction and Selection, pp. 33-50.* The Springer International Series in Engineering and Computer Science, vol 453. Springer, Boston, MA.

Kołcz, A., Chowdhury, A. and Alspector, J. (2004). The impact of feature selection on signature-driven spam detection. *In Proceedings of the 1st Conference on Email and Anti-Spam (CEAS-2004).*

Li, Ch.H. and Huang, J.X. (2012). Spam filtering using semantic similarity approach and adaptive BPNN. *Neurocomputing*, 92: 88-97.

Metsis, V., Androutsopoulos, I. and Paliouras, G. (2006). Spam filtering with naive bayes-which naive bayes?. *CEAS 2006 - The Third Conference on Email and Anti-Spam,* July 27-28, 2006, Mountain View, California, USA.

Michie, D., Spiegelhalter, D.J. and Taylor, C.C. (1994). *Machine learning, neural and statistical classification*. *Technometrics*, 37(4). DOI:10.2307/1269742

Mitchell, T. M. (1997). *Machine Learning, McGraw-Hill Higher Education*. New York.

Mohammad, A.H., and Abu Zitar, R. (2011). Application of genetic optimized artificial immune system and neural networks in spam detection. *Applied Soft Computing*, 11(4), 3827-3845.

Mohammadzadeh, H. and Soleimanian Gharehchopogh, F. (2021). A novel hybrid whale optimization algorithm with flower pollination algorithm for feature selection: Case study Email spam detection. *Computational Intelligence*, 37(1), 176– 209.

Nadjate, S., Adi, K. and Allili, M.S. (2020). A semantic-based classification approach for an enhanced spam detection. *Computers & Security,* 94: 101716.

Razmara, M., Asadi, B., Narouei, M. and Ahmadi, M. (2012). A novel approach toward spam detection based on iterative patterns*. In 2012 2nd International eConference on Computer and Knowledge Engineering (ICCKE)*: pp 318-323

Stanimirović, Z. (2012). A genetic algorithm approach for the capacitated single allocation p-hub median problem. *Computing and Informatics*, 29(1), 117-132.

Su, M.C., Lo, H.H. and Hsu, F.H. (2010). A neural tree and its application to spam e-mail detection. *Expert Systems with Applications*, 37(12), 7976-7985.

Wang, B., Jones, G. JF and Pan, W. (2006). Using online linear classifiers to filter spam emails. *Pattern analysis and applications*, 9(4), 339-351.

Wu, C.H. (2009). Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks. *Expert Systems with Applications*, 36(3), 4321-4330.

Xu, H., and Yu, B. (2010). Automatic thesaurus construction for spam filtering using revised back propagation neural network. *Expert Systems with Applications*, 37(1), 18-23.

Yang, Y., and Pedersen, J.O. (1997). A comparative study on feature selection in text categorization. *Proceedings of the Fourteenth International Conference on Machine*, 97(412-420), p. 35.

Zorkadis, V., Karras, D.A. & Panayotou, M. (2005). Efficient information theoretic strategies for classifier combination, feature extraction and performance evaluation in improving false positives and false negatives for spam e-mail filtering. *Neural Networks*, 18(5), 799-807.

Ensemble Bayesian Classification Using Genetic Algorithm Wrapper Feature Selection in Spam Detection

Vahid Nosrati[1]
Ph.D. Candidate, Computrt Department, Faculty of Engineering, Arak University, Arak, Iran

Mohsen Rahmani
Associate Prof., Computer Department, Faculty of Engineering, Arak University, Arak, Iran

Abstract

The role of email in communication is seriously threatened by a phenomenon called spam. So far, many methods have been proposed to deal with this phenomenon, one of the most important of which is to classify emails based on their content into two categories: spam and non-spam. Content-based classification mechanisms use the words as features, where applying an efficient feature selection mechanism is critical due to the large number of features. Therefore, the main focus of this paper is to select useful features via proposing a wrapper feature selection approach based on a powerful genetic algorithm. We then apply a Bayesian classifier, which has demonstrated a high efficiency in text classification. The main steps of the proposed method is as follows: first, an initial feature vector is chosen, then it is optimized by multiplying the vector in a matrix called the transformation matrix made by the genetic algorithm, and finally, a set of k feature vectors is generated. An ensemble classification approach composed of k Bayesian classifiers is applied to the feature vectors, and the ultimate class label is determined by voting among ensemble members. The proposed method is implemented on two datasets PU1 and PU2. The results show that the classification accuracy of the proposed method with k=7 reaches 87.86 and 87.91 in PU1 and PU2, rspectively. The results also indicate the efficiency of the proposed method compared to naïve Bayes and two well-known classifiers SVM and KNN.

Keywords: Email, Spam, Classification, Genetics Algorithm, Feature Selection, Transformation Matrix, Ensemble Learning

1. Corresponding Author: vh_nosraty@yahoo.com