تشخیص خودکار خطا در پایگاه داده، مبتنی بر خوشه‌بندی و نزدیک‌ترین همسایگی

مهدیه عطاییان و نگین دانشپور

چکیده: کیفیت داده‌ها در امر تصمیم‌گیری سازمان‌ها تأثیرگذار می‌باشد، به گونه‌ای که تصمیم‌گیری مبتنی بر داده‌های فاقد کیفیت سازمان را متحمل هزینه‌های بالایی می‌کند. کیفیت داده‌ها دارای ابعاد متنوعی می‌باشد که صحت از مهم‌ترین این ابعاد است. جهت تصحیح داده‌ها نیاز به تشخیص خطا وجود دارد که با توجه به حجم بالای داده‌ها، نیاز به یک سیستم خودکار است تا بدون دخالت کاربر این فرایند انجام گیرد. در این مقاله راهکاری خودکار مبتنی بر خوشه‌بندی k-means جهت تشخیص خطا ارائه شده است. در ابتدا به ازای هر ویژگی، داده‌ها خوشه‌بندی می‌شوند و سپس به ازای هر داده در آن خوشه از روش شبه k نزدیک‌ترین همسایه، جهت شناسایی خطا استفاده می‌شود. روش پیشنهادی توانایی تشخیص چندین خطا در یک رکورد را دارد و همچنین قادر است خطا در فیلدهایی با انواع داده متفاوت را نیز شناسایی کند. آزمایشات نشان می‌دهد که به طور متوسط این روش می‌تواند ۹۱٪ خطاهای موجود در داده‌ها را شناسایی نماید. همچنین روش پیشنهادی با یک روش تشخیص خطا به وسیله قوانین که همانند راهکار پیشنهادی روشی خودکار برای تشخیص خطا در انواع داده‌ای متفاوت است نیز مورد مقایسه قرارگرفته و نتایج نشان می‌دهد که روش پیشنهادی به طور متوسط ۲۵٪ عملکرد بهتری در تشخیص خطا داشته است.

کلیدواژه: **تصحیح داده، تشخیص خودکار خطا، خوشه‌بندی، k-means.**

۱- مقدمه

رشد روزافزون داده‌ها موجب گردیده تا سازمان‌ها با داده‌های حجیم و منابع ناهمگون و توزیع‌شده روبه‌رو گردند. سازمان‌ها جهت تصمیم‌گیری نیازمند استفاده از این داده‌ها و استخراج دانش از آنها می‌باشند. جمع‌آوری این حجم از داده‌ها موجب ایجاد موضوع دیگری به نام کیفیت داده‌ها[1] برای سازمان‌ها می‌گردد. زمانی که داده‌ها از منابع و سیستم‌های مختلف به یک سیستم دیگر منتقل می‌شوند ممکن است دچار خطا گردند و یا این که با مشکلاتی از قبیل فرمت و دامنه ناهمگون[2] مواجه گردند. تصمیم‌گیری بر مبنای این داده‌های فاقد کیفیت علاوه بر این که به ساختار سازمان آسیب می‌زند باعث می‌گردد هزینه‌های زیادی به سازمان وارد شود. بر اساس مطالعات انجام‌شده بیش از ۳۰٪ داده‌ها فاقد کیفیت هستند و این گونه داده‌ها موجب گردیده سالانه ۳ تریلیون دلار به دولت امریکا خسارت وارد گردد [۱]. لذا کیفیت داده‌ها در منابع داده از اهمیت بالایی برخوردار است. فراهم‌نمودن داده‌هایی با کیفیت بالا کاری زمان‌بر و هزینه‌بر می‌باشد [۲] اما در قبال هزینه‌های شکست بر اثر داده‌های فاقد کیفیت بسیار ارزشمند است.

کیفیت داده‌ها دارای ابعاد متنوعی است که صحت[3] درمیان ابعاد دیگر حایز اهمیت بالایی می‌باشد. مشکلاتی از قبیل ناقص‌بودن[4]، ناسازگاری[5] و مقادیر از دست رفته[6] باعث نقض این بعد می‌شوند. تصحیح داده‌ها فرایندی می‌باشد که برای تشخیص داده‌های ناقص، نادرست و ناسازگار، و بهبود کیفیت داده‌ها با اصلاح خطاهای شناسایی‌شده به کار می‌رود [۳]. روند تصحیح داده‌ها می‌تواند وقت‌گیر و خسته‌کننده باشد اما نمی‌توان آن را نادیده گرفت [۴]. با توجه به حجم داده‌ها، راهکارهای تصحیح تعاملی غیر کاربردی می‌باشد و نیاز به راهکارهای خودکار وجود دارد.

داده‌کاوی یکی از تکنیک‌های کلیدی برای تصحیح داده‌ها می‌باشد و [۳] تا کنون راهکارهای مختلفی جهت تصحیح داده‌ها ارائه شده است. روش‌های مطرح‌شده در این حوزه از هستان‌شناسی [۵]، طبقه‌بندها (درخت تصمیم [۶] و [۷]، شبکه عصبی [۸] و قانون بیز [۹])، قوانین [۱۰] و [۱۱]، یادگیری مرکب [۸] و وابستگی تابعی [۱۲] برای تصحیح داده استفاده نموده‌اند. برخی از این روش‌ها دارای معایبی از قبیل تعامل با کاربر، عدم تشخیص خطا و تفکیک‌ناپذیری فاز تشخیص و تصحیح می‌باشند. برای انجام این فرایند باید فرمت و دامنه ویژگی‌ها مورد بررسی قرار گیرد، ارتباط بین ویژگی‌ها برای کشف ناسازگاری‌ها بررسی شود و فیلدهای دارای مقادیر از دست رفته یافته شود.

با توجه به این که در برخی راهکارها تعامل با کاربر الزامی است [۵] و این تعامل در حجم داده‌ای بالا مقدور نمی‌باشد، در این مقاله رویکردی خودکار برای تشخیص خطا ارائه شده است. همچنین برخی راهکارها فقط توانایی تشخیص خطا در یک نوع داده‌ای خاص را دارا می‌باشند [۶] و [۷] که راهکار ارائه‌شده در این مقاله توانایی تشخیص خطا در تمام انواع داده‌ای را دارا است. علاوه بر این در برخی روش‌های پیشین ارائه‌شده، فاز تشخیص و تصحیح خطا غیر قابل تفکیک می‌باشد و کاربر نمی‌تواند از رویکرد دیگری برای تصحیح استفاده نماید [۸] و یا در برخی راهکارها فرض شده که خطا تشخیص داده شده و فقط به تصحیح می‌پردازد [۹] که راهکار ارائه‌شده فقط به تشخیص خطا پرداخته و به کاربر امکان اتخاذ هر رویکرد دلخواهی را برای تصحیح می‌دهد. همچنین برخی راهکارها تنها توانایی تشخیص یک خطا در یک رکورد را دارا هستند در صورتی که روش پیشنهادی قادر است چندین خطا در یک رکورد را شناسایی کند. این رویکرد مبتنی بر خوشه‌بندی k-means و روشی شبه k نزدیک‌ترین همسایگی[7] که در هر خوشه اجرا می‌گردد می‌باشد. نتایج آزمایشات بر

این مقاله در تاریخ ۶ مهر ماه ۱۳۹۴ دریافت و در تاریخ ۱۷ خرداد ماه ۱۳۹۵ بازنگری شد.

مهدیه عطاییان، دانشکده مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، (email: m.ataeyan@srttu.edu).

نگین دانشپور، دانشکده مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، (email: ndaneshpour@srttu.edu).

1. Data Quality
2. Heterogeneous
3. Accuracy
4. Incomplete
5. Inconsistent
6. Missing Value
7. K-Nearest Neighbor

روی سه مجموعه داده با ابعاد و ویژگی‌های متفاوت از پایگاه داده UCI machine learning[1] نشان می‌دهد که به طور متوسط این روش ۹۱٪ از خطاهای موجود در منبع داده را شناسایی نموده است. علاوه بر این، روش پیشنهادی با روشی مشابه که از قوانین برای تشخیص خطا [۳] استفاده نموده، مقایسه شده است. نتایج بیانگر آن است که روش پیشنهادی به طور متوسط ۲۵٪ توانایی بیشتری در تشخیص خطا در مقایسه با راهکار مبتنی بر قوانین [۳] داشته است.

بخش‌های بعدی این مقاله بدین شرح است. در بخش ۲ ابتدا پیشینه‌ای از روش‌های تصحیح خطا بیان شده و معایب هر روش مطرح می‌شود. در بخش ۳ روش پیشنهادی شرح داده می‌شود و در بخش ۴ نتایج آزمایشات بر روی سه مجموعه داده بیان شده که مجموعه داده اول شامل ویژگی‌های عددی[2] و اسمی[3]، مجموعه دوم شامل ویژگی‌های ترتیبی[4] و عددی و مجموعه داده سوم شامل ویژگی‌های اسمی، عددی و ترتیبی می‌باشد. همچنین در بخش ۴ راهکار پیشنهادی با روش تشخیص خطا مبتنی با قانون [۳] مقایسه می‌گردد و نهایتاً در بخش پایانی خلاصه و نتیجه‌گیری مقاله ارائه می‌شود.

۲- پیشینه تحقیق

داده‌های دنیای واقعی دارای مشکلاتی از قبیل ناسازگاری، خطا و مقادیر از دست رفته می‌باشند و این خطا ممکن است در انواع داده‌ای مختلف (عددی، اسمی و ترتیبی) رخ دهد. برای جلوگیری از تأثیر این داده‌های خطادار بر روی تصمیمات مبتنی بر این داده‌ها، خطاها باید اصلاح شوند. روند اصلاح داده‌ها از دو فاز تشخیص ویژگی حاوی خطا و جایگزینی آن با مقادیر صحیح تشکیل شده است. تشخیص خطا شامل شناسایی ناسازگاری بین ویژگی‌ها و رکوردها، نقض وابستگی‌های تابعی آن مجموعه داده و مقادیر از دست رفته می‌باشد. پس از شناسایی خطاها کاربر می‌تواند آنها را حذف و یا با استفاده از یکی از تکنیک‌های موجود تصحیح نماید. راهکارهای ارائه‌شده در این حوزه از رویکرد داده‌کاوی برای تصحیح استفاده نموده‌اند. بعضی از این راهکارها توانایی تشخیص و تصحیح بر روی نوع داده‌ای خاص و برخی دیگر توانایی تشخیص و تصحیح بر روی تمام انواع داده‌ای را دارا می‌باشند. در این بخش به معرفی این راهکارها و بیان معایب هر یک پرداخته شده است.

در [۵] روش ارائه‌شده از هستان‌شناسی برای شناسایی مقادیر معتبر و غیر معتبر رکوردها استفاده نموده است. مقادیر غیر معتبر پس از شناسایی به همراه تمامی مقادیر معتبر که در هستان‌شناسی موجود است به کاربر نشان داده شده و از او خواسته می‌شود که برای رکورد مورد نظر از میان مقادیر معتبر مقداری را انتخاب نماید. مقادیر انتخاب‌شده به وسیله کاربر به همراه مقادیر غیر معتبر ذخیره می‌گردد. زمانی که مقادیر تصحیح‌شده توسط کاربر به یک حد آستانه از پیش تعیین شده رسید، از آنها قوانین استخراج می‌شود. از این پس، هنگامی که مقادیر غیر معتبر به وسیله هستان‌شناسی تشخیص داده شد به جای تعامل با کاربر برای تصحیح، از قوانین استخراج‌شده برای اصلاح استفاده می‌شود. این روش به دلیل نیاز به کاربر در فاز اولیه در حجم داده‌ای بالا مناسب نمی‌باشد.

روش DMI بر پایه الگوریتم EM و درخت تصمیم با الگوریتم C۴.۵ طراحی شده است [۶] و [۷]. این شیوه برای تصحیح مقادیر از دست رفته به کار می‌رود. رکوردهای دارای مقدار از دست رفته شناسایی و از رکوردهای صحیح به ازای هر ویژگی یک درخت تصمیم رسم می‌شود. درخت تصمیم رکوردها را به تعدادی برگ دسته‌بندی می‌کند به نحوی که رکوردهای موجود در یک برگ دارای کلاس یکسان می‌باشد. رکوردهای دارای مقدار از دست رفته را به هر درخت انتساب می‌دهد و برچسب برگی را که در آن واقع گردید به عنوان مقدار صحیح جایگزین انتخاب می‌کند. این روش فقط قادر به تصحیح خطای مقادیر از دست رفته بوده و همچنین این تصحیح مختص داده‌های عددی و دسته‌ای می‌باشد.

الگوریتم SiMI توسعه‌یافته الگوریتم DMI می‌باشد [۶]. در این روش به جای استفاده از درخت تصمیم از جنگل تصمیم برای ساخت درختان استفاده شده است. این الگوریتم را می‌توان برای یافتن مقادیر عددی و دسته‌ای استفاده کرد. این شیوه پس از ساخت درختان به جستجوی تقاطع برگ‌های متفاوت می‌پردازد. کوچک‌ترین تقاطع را با تقاطعی که شباهت بین رکوردها در آن تقاطع بیشتر است و همچنین در هنگام ادغام با آن تقاطع وابستگی بین ویژگی‌ها را بیشینه می‌نماید، ادغام می‌کند. در مرحله آخر هر رکورد دارای مقدار از دست رفته را به یک تقاطع برای یافتن مقدار جایگزین نسبت می‌دهد. این روش به دلیل آن که توسعه‌یافته روش DMI می‌باشد فقط قادر به تصحیح مقادیر از دست رفته برای مقادیر عددی و دسته‌ای است.

در [۸] نویسنده به معرفی روش پیش‌بینی bagging پرداخته است. در این روش الگوریتم به تولید نسخه‌های متعدد از پیشگوها می‌پردازد و در نهایت بر اساس نظر تمام پیشگوها، پیش‌بینی انجام می‌دهد. ایجاد نسخه‌های متعدد با ساخت مکرر bootstrap بر روی مجموعه یادگیری حاصل می‌گردد و به عنوان مجموعه‌های یادگیری جدید استفاده می‌شود. در این روش فاز تصحیح و تشخیص غیر قابل تفکیک می‌باشد و لذا کاربر نمی‌تواند رویکرد دیگری را برای فاز تصحیح به کار گیرد.

الگوریتم SCARE با ترکیب الگوریتم‌های یادگیری ماشین با درست‌نمایی[5] به تصحیح داده‌های حاوی خطا می‌پردازد [۹]. هدف در این این روش تصحیح مقادیر ناصحیح به وسیله مقادیر صحیح می‌باشد. برای استفاده از درست‌نمایی از دو معیار برای بیشینه‌کردن سود و کمینه‌کردن هزینه تغییرات بهره گرفته شده است. در این روش به ازای هر ویژگی یک طبقه‌بند ایجاد و مقدار هر ویژگی غلط پیش‌بینی می‌شود. در مرحله بعد برای هر رکورد یک گراف که نودها مقادیر پیش‌بینی شده و یال‌ها وابستگی میان ویژگی‌ها را نشان می‌دهد، رسم می‌شود. پس از رسم گراف تا زمانی که از هر ویژگی تنها یک مقدار باقی بماند، نودی که یال‌های آن دارای کمترین وزن هستند، حذف می‌شود. در این روش فرض شده که خطا تشخیص داده شده و فقط به تصحیح پرداخته است.

الگوریتم ارائه‌شده در [۱۰] با استفاده از قوانین تعریف‌شده به تصحیح یک منبع ناسازگار می‌پردازد. این قوانین از سه بخش اصلی تشکیل می‌شوند که بخش اول و دوم سمت چپ قانون و بخش سوم سمت راست قانون را تشکیل می‌دهند. بخش اول الگوهای شاهد نامیده می‌شود که بیانگر آن دسته از ویژگی‌هایی که با یکدیگر در ارتباط هستند، می‌باشد. بخش دوم شامل الگوهای منفی است که بیانگر مقادیر اشتباه برای ویژگی‌ها می‌باشد. بخش آخر مقادیر واقعی ویژگی‌هایی می‌باشد که مقدار صحیح را به ازای مقدار غلط بیان می‌کند. در ابتدا هر یک از ویژگی‌های ارائه‌شده در بخش الگوهای شاهد به تنهایی به عنوان یک کلید انتخاب

1. https://archive.ics.uci.edu/ml/datasets
2. Numeric
3. Nominal
4. Categorical
5. Likelihood

شده و سپس هر یک با مقادیر خود در یک لیست با نام قانون ذخیره می‌شود. در هر رکورد به جستجوی کلیدهای ذخیره‌شده در لیست پرداخته شده و در صورتی که تطابقی یافت گردد، به بررسی بخش دوم قانون پرداخته می‌شود. در صورتی که بخش دوم در رکورد موجود بود، آن رکورد باید با بخش سوم تصحیح گردد و این روند باید برای تمامی رکوردها تکرار شود. هدف از تعیین کلید در این الگوریتم صرفه‌جویی در هزینه مقایسات است زیرا با این کار لازم نیست تمام قسمت‌های یک قانون برای یک کلید مورد بررسی قرار گیرد. در حجم داده‌های بالا تعداد قوانین افزایش خواهد یافت و همچنین هر قانون حاوی چندین کلید می‌باشد. بررسی این حجم کلید برای داده‌هایی با حجم بالا بسیار زمان‌گیر است.

در [۱۱] روشی جهت تصحیح ناسازگاری‌ها با استفاده از قوانین معرفی شده است. در این روش ابتدا پایگاه داده مورد بررسی انتخاب می‌شود. قوانین موجود در آن منبع با استفاده از الگوریتم‌های موجود استخراج می‌گردد و پس از آن الگوریتم اطمینان[1] هر قانون را محاسبه می‌کند. هر تراکنشی که با این قوانین تناقض داشته باشد، مشکوک به داشتن کاستی و خطا است. باید تمام قوانین استخراج‌شده، مورد بررسی قرار گیرند. الگوریتم به هر تراکنش بر مبنای تعداد قوانینی که نقض نموده نمره‌ای اختصاص می‌دهد به گونه‌ای که تراکنشی که قوانین بیشتری را نقض کرده، نمره بالاتری را می‌گیرد. برای یافتن نمره مختص هر تراکنش، الگوریتم از حاصل جمع اطمینان‌های نقض‌شده به توان τ استفاده می‌نماید. مقدار τ یک مقدار تجربی می‌باشد. تراکنش‌ها به همراه نمرات اختصاص داده شده به کاربر نمایش داده می‌شود و به او امکان می‌دهد تا با توجه به نمرات درباره تراکنش‌ها تصمیم‌گیری نماید. این روش به دلیل نیاز به کاربر در فاز تصحیح در حجم داده‌های بالا مناسب نمی‌باشد.

روش‌های زیادی از وابستگی تابعی برای تشخیص خطا استفاده کرده‌اند اما به ازای ناسازگاری‌های شناخته‌شده، تصحیح‌های متنوعی را می‌توان ارائه نمود که باید از میان آنها یک تصحیح به عنوان تصحیح نهایی معرفی گردد. برای انتخاب تصحیح بهینه دو پارامتر هزینه و تنوع مطرح می‌باشد. الگوریتم ارائه‌شده در [۱۲] برای نخستین بار هر دو پارامتر را به عنوان هدف خود قرار داده است. برای محاسبه میزان تنوع از تابع فاصله استفاده می‌شود که میزان عدم شباهت رکوردها را محاسبه می‌نماید. برای محاسبه هزینه نیز میزان تغییرات در منبع اولیه با منبع در دسترس را در نظر می‌گیرد. در حجم بالای داده‌ها محاسبه دو تابع بسیار زمان‌گیر می‌باشد.

در [۱۳] یک روش مبتنی بر قانون که در آن با کاربر تعامل وجود دارد بیان شده است. در این روش ابتدا قوانین از منابع استخراج می‌گردد و همچنین فرمت و دامنه معتبر برای هر ویژگی نیز تعیین می‌شود. این الگوریتم در حین تعامل با کاربر به او امکان می‌دهد تا منبع مورد نظر خود را به همراه قوانین دلخواه از میان قوانین استخراج‌شده انتخاب نماید. این الگوریتم سعی می‌کند تا خطاهای لغوی، خطای دامنه و فرمت و مقادیر از دست رفته را برطرف نماید. این روش نیز به دلیل نیاز به کاربر در فاز تصحیح در حجم داده‌های بالا مناسب نمی‌باشد.

در [۱۴] تا [۱۶] هدف شناسایی ویژگی‌های مشکوک به خطا و کلاس آن ویژگی‌ها و اصلاح آن به وسیله الگوریتم polishing که دارای دو فاز پیش‌بینی و تنظیم مقادیر می‌باشد است. در فاز پیش‌بینی از میان الگوریتم‌های طبقه‌بندی، یک روش انتخاب شده و به ازای هر ویژگی ۱۰ دسته بر روی داده‌ها ایجاد می‌شود. در این مرحله مقدار ویژگی مورد نظر برای هر رکورد به ازای هر ۱۰ طبقه‌بند پیش‌بینی می‌گردد. در صورتی که مقدار اصلی آن رکورد با مقدار پیش‌بینی شده تناقض داشته باشد، مقادیر حاصل از پیش‌بینی برای آن رکورد جهت تصحیح در فاز بعدی ذخیره می‌گردد. در فاز تنظیم، مجدداً ۱۰ دسته بر روی ویژگی‌ها ایجاد می‌کند. سپس به ازای هر رکورد دارای مقدار نادرست که در فاز قبل تشخیص داده شد، مقدار آن را بر اساس مقادیر پیش‌بینی شده و توسط ۱۰ طبقه‌بند تصحیح می‌نماید. مقدار صحیح تنها می‌تواند از میان مقادیر پیش‌بینی شده در فاز قبل انتخاب شود. هدف در این فاز به حداقل رساندن تغییرات است. در این روش نیز فاز تصحیح و تشخیص غیر قابل تفکیک می‌باشد و لذا کاربر نمی‌تواند رویکرد دیگری را برای فاز تصحیح به کار گیرد.

در [۱۷] از محدودیت موجود در پایگاه داده برای تصحیح ناسازگاری‌ها استفاده شده است. ابتدا محدودیت‌های موجود در پایگاه داده استخراج می‌شود و پس از آن به جستجوی رکوردهای نقض‌کننده محدودیت‌ها می‌پردازد. رکوردهای دارای ناسازگاری به وسیله یک الگوریتم حریصانه تصحیح می‌گردند. استخراج محدودیت‌ها نیاز به تعامل کاربر دارد و از این رو این روش نیز در حجم داده‌های بالا کارا نیست.

روش [۳] یک راهکار تشخیص خطای مبتنی بر قوانین است. در این روش ابتدا باید تمام ویژگی‌ها باینری گردد. پس از باینری‌نمودن ویژگی‌ها، قوانین با کمترین support از منبع داده‌ای استخراج می‌شوند. تعداد دفعاتی که هر قانون توسط داده‌های منبع نقض‌شده محاسبه گردیده و قوانینی که بیش از یک حد آستانه نقض شده باشند، حذف می‌گردند. در مرحله بعدی قوانینی که دارای پدر هستند حذف می‌شود، بدین صورت که اگر دو قانون به صورت $x,y \to z$ و $x \to z$ در مجموعه قوانین موجود باشد، $x \to z$ پدر $x,y \to z$ است و به همین علت $x,y \to z$ از مجموعه قوانین حذف می‌گردد. سپس تعداد قوانینی را که هر رکورد از مجموعه نهایی قوانین نقض نموده است، محاسبه می‌شود و رکوردهایی که بیش از یک حد آستانه را نقض نموده باشند به عنوان رکورد حاوی خطا معرفی می‌گردند. یکی از معایب اصلی این روش آن است که با این شیوه فقط می‌توان نتیجه گرفت که یک رکورد خطادار است ولی نمی‌توان گفت که کدام ویژگی باعث ایجاد خطا گردیده است.

روش‌های مورد بررسی، در خودکار یا نیمه خودکار بودن، توانایی تشخیص خطا، نوع و تعداد خطاهایی که توسط یک روش قابل شناسایی است با یکدیگر متفاوت می‌باشند. هدف در این مقاله ارائه روشی جهت تشخیص خودکار ویژگی خطادار به صورت مجزا از فاز تصحیح برای انواع داده‌ای متفاوت می‌باشد. در بخش بعدی روش پیشنهادی بیان می‌شود و در بخش ۴ با راهکار [۳] مورد مقایسه قرار می‌گیرد. هدف از انتخاب این راهکار برای مقایسه آن است که این روش نیز به تشخیص خطا به طور خودکار و به صورت مجزا برای انواع داده‌ای مختلف پرداخته است.

۳- روش پیشنهادی

در این بخش یک روش تشخیص خودکار خطا مبتنی بر خوشه‌بندی k-means پیشنهاد شده است. این روش توانایی تشخیص خطا در انواع داده‌ای متفاوت را داراست. همچنین با این روش می‌توان ویژگی‌ای را که خطا در آن رخ داده است به صورت مشخص تعیین نمود. روش ارائه‌شده فقط به تشخیص خطا می‌پردازد و برای تصحیح می‌توان از هر یک از رویکردهای موجود استفاده نمود. این روش شامل ۵ مرحله است که در زیربخش‌های بعدی شرح داده می‌شود. همچنین شبه‌کد مراحل ۳-۱ تا ۳-۴ که در واقع قسمت اصلی الگوریتم می‌باشد در شکل ۱ نشان داده شده و در هر مرحله به توضیح هر قسمت از شکل ۱ پرداخته شده است.

1. Confidence

Input
D: a dataset $\{t_1, t_2, \ldots, t_k\}$ with k attribute
K: number of $K-means$ cluster
K': number of neighbor
Φ: threshold for difference between prediction value and examined value
Output
Out: set of erroneous record
Procedure
1. **foreach** attribute t_k in D
2. **foreach** *record* in D
3. $D \leftarrow$ detect outlier of t_k *attribute* and delete them
4. **end**
5. $D' \leftarrow$ remove t_k from D
6. $RecordsCusterNumber \leftarrow K-\text{means}(D', K)$
7. **foreach** *record* in D
8. $idx \leftarrow$ find cluster of *record* from *RecordsClusterNumber*
9. **foreach** r in D and in idx cluster
10. $dis(r) \leftarrow$ Euclidean distance between *record*, r
11. **end**
12. $Indexes \leftarrow$ ascending sort $Indexes$ of records according dis
13. $Indexes \leftarrow$ select K' head of $Indexes$
14. **if** t_k is numeric
15. $Sum \leftarrow 0$
16. **foreach** i in $Indexes$
17. $Sum \leftarrow Sum + D(i,k)$
18. **end**
19. $Sum \leftarrow Sum/K'$
20. **if** $(Sum - D(record,k)) > \Phi$
21. $Out \leftarrow Out \cup record$
22. **end**
23. **end**
24. **if** t_k is nominal or ordinal
25. $Sum \leftarrow Null$
26. **foreach** i in $Indexes$
27. $Sum \leftarrow Sum \cup i$
28. **end**
29. **if** $(Sum \neq D(record,k))$
30. $Out \leftarrow Out \cup record$
31. **end**
32. **end**
33. **return** Out
34. **end**
35. **end**

شکل ۱: شبه‌کد مراحل ۳-۱ تا ۳-۴ الگوریتم پیشنهادی.

۳-۱ مرحله اول- حذف داده‌های پرت

در ابتدا باید داده‌های پرت[1] به ازای هر ویژگی از داده‌های منبع به وسیله یکی از روش‌های موجود [۱۸] شناسایی گردد. در این مقاله از میان روش‌های موجود از خوشه‌بندی k-means استفاده شده است [۱۹]. برای تعیین k می‌توان از یکی از روش‌های ارزیابی خوشه‌ها[2] [۲۰] استفاده نمود. در این مقاله از ارزیابی Silhouette index برای تعیین k استفاده گردیده است [۲۱]. در صورتی که داده‌های پرت حذف نگردد، این روش در حین تشخیص خطا می‌تواند داده‌های پرت را نیز به عنوان خطا به همراه خطاها معرفی کند. به دلیل این که رویکرد تصحیح داده‌های پرت، غالباً حذف می‌باشد و با توجه به این که تفکیکی بین داده‌های پرت و خطادار وجود ندارد، کاربر به اجبار باید داده‌های خطادار را نیز حذف نماید.

1. Outlier
2. Validity Index

از این رو برای این که کاربر بتواند هر رویکردی برای تصحیح خطا اتخاذ کند باید در ابتدا داده‌های پرت شناسایی و از منبع داده‌ای حذف شوند. خطوط ۲-۴ در شکل ۱ این قسمت از الگوریتم را نشان می‌دهد.

۳-۲ مرحله دوم- خوشه‌بندی

فرض کنید منبع داده ما شامل m ویژگی به صورت $\{t_1, t_2, \ldots, t_m\}$ باشد. در این مرحله باید به ازای هر m ویژگی خوشه‌بندی k-means اجرا گردد بدین صورت که برای ویژگی t_m، t_m از میان ویژگی‌های منبع برداشته می‌شود و بر اساس ویژگی‌های دیگر خوشه‌بندی k-means انجام می‌گیرد. برای انتخاب k نیز می‌توان از یکی از شیوه‌های ارزیابی خوشه‌ها [۲۰] استفاده نمود. در این مقاله از Silhouette index برای تعیین و ارزیابی تعداد خوشه‌ها بهره گرفته شده است [۲۱]. دلیل انتخاب k-means از میان الگوریتم‌های خوشه‌بندی این است که k-means در مقایسه با روش‌های دیگر نسبت به خطا حساس‌تر می‌باشد [۲۲]. بعد از هر خوشه‌بندی، به ازای هر ویژگی، مراحل سوم، چهارم و پنجم اجرا می‌گردد که خط ۵ و ۶ در شکل ۱ این مرحله از الگوریتم را نمایش می‌دهد. در خط ۵ ویژگی مورد بررسی از میان داده‌ها حذف می‌شود و در خط ۶ به ازای بقیه ویژگی‌ها خوشه‌بندی k-means انجام می‌گیرد. شماره خوشه‌هایی که هر رکورد در آن قرار می‌گیرد به عنوان نتیجه خوشه‌بندی ذخیره می‌گردد.

۳-۳ مرحله سوم- مقایسه

بعد از هر خوشه‌بندی به ازای هر ویژگی t_m، در این مرحله در هر خوشه فاصله هر رکورد با بقیه رکوردها محاسبه می‌شود. سپس روشی شبیه k نزدیک‌ترین همسایگی به ازای تمام رکوردها در تمامی خوشه‌ها اجرا می‌گردد. فرض کنید رکورد r در خوشه k قرار گرفته است. برای انجام روش شبه k نزدیک‌ترین همسایگی باید برای رکورد r، k' رکوردی که در خوشه k دارای فاصله کمتری نسبت به رکورد r هستند انتخاب شوند. مقدار k' یک مقدار تجربی است و چنانچه ویژگی t_m از نوع داده‌ای عددی باشد، باید میانگین k' همسایه برای ویژگی t_m محاسبه شود اما چنانچه t_m ترتیبی یا اسمی باشد، مقداری که بیش از بقیه در این k' همسایه تکرار شده انتخاب می‌شود. این مقدار a نامیده می‌شود. خطوط ۷-۱۹ و ۲۴-۲۸ در شکل ۱ این مرحله از الگوریتم را نشان می‌دهد. در خطوط ۷-۱۳، فاصله رکورد r در خوشه k با بقیه رکوردها محاسبه و k' رکوردی که در خوشه k دارای فاصله کمتری نسبت به رکورد r هستند، انتخاب شده است. خطوط ۱۴-۱۹ و ۲۴-۲۸ مقدار a را به ترتیب به ازای ویژگی‌های عددی، ترتیبی و اسمی انجام می‌دهد. برای محاسبه فاصله در این روش از فاصله اقلیدسی استفاده شده است.

۳-۴ مرحله چهارم- تشخیص خطا

در صورتی که ویژگی t_m عددی باشد، اگر تفاضل مقدار آن ویژگی در آن رکورد با مقدار a از یک حد آستانه بیشتر باشد، آن رکورد خطادار است. چنانچه ویژگی ترتیبی یا اسمی باشد، در صورتی که مقدار آن ویژگی در آن رکورد با مقدار a تناقض داشته باشد، آن رکورد خطادار می‌باشد. در شکل ۱ در خطوط ۲۰-۲۲ تشخیص خطا برای ویژگی‌های عددی و در خطوط ۲۹-۳۲ تشخیص خطا برای ویژگی‌های اسمی و دسته‌ای انجام می‌گیرد.

۳-۵ مرحله پنجم- کاهش خطا

این الگوریتم علاوه بر شناسایی ویژگی‌های خطادار، تعدادی ویژگی صحیح را نیز به عنوان خطا شناسایی می‌کند. برای به حداقل رساندن این مقدار خطا، مراحل ۳-۲ تا ۳-۴ حداقل ۲ و حداکثر ۵ بار با k' همسایگی اجرا می‌گردد که این تعداد تکرار از آزمایشات بر روی داده‌های مختلف حاصل شده است. نظر به این که رویکرد ویژگی‌های عددی با ویژگی‌های اسمی و ترتیبی متفاوت می‌باشد، دو رویکرد متفاوت برای کاهش خطا ارائه شده است. چنانچه ویژگی عددی باشد، تکراری که تعداد خطای بیشتری را نشان می‌دهد، انتخاب می‌شود. این تکرار n_1 نامیده شده و با بقیه تکرارها مقایسه می‌گردد. فرض می‌شود که n تعداد تکرارها باشد، در صورتی که هر ویژگی خطادار در n_1 در کمتر از $n-1$ تکرار دیگر وجود داشت، آن ویژگی از رکورد صحیح است.

برای ویژگی‌های اسمی و ترتیبی همانند ویژگی عددی بزرگ‌ترین تکرار انتخاب و n_1 نامیده می‌شود و در مقایسه با تکرارهای دیگر قرار می‌گیرد. عناصری که در کمتر از $n-1$ تکرار وجود داشته باشند، حذف شده و مابقی در یک حافظه موقت ذخیره می‌گردد. بار دیگر مراحل ۳-۲ تا ۳-۴ با $k'-1$ همسایگی حداقل ۲ و حداکثر ۵ بار اجرا می‌گردد اما این بار تکرارها با عناصر موجود در حافظه موقت مقایسه می‌شوند. هر عضوی در حافظه که در n تکرار موجود باشد به عنوان خطای نهایی شناسایی می‌شود.

۳-۶ پیچیدگی زمانی

در این قسمت به شرح پیچیدگی الگوریتم پیشنهادی پرداخته می‌شود. فرض می‌شود m تعداد رکوردهای منبع داده، k تعداد ویژگی‌های آن و K تعداد خوشه‌های الگوریتم k-means باشد. یک حلقه کلی در خط ۱ وجود دارد که خطوط ۲-۳۴ الگوریتم را k مرتبه اجرا می‌کند. در این خطوط الگوریتم k-means یک بار برای حذف داده‌های پرت و بار دیگر برای خوشه‌بندی بعد از حذف داده‌های پرت اجرا می‌شود که مرتبه اجرایی آن Kmt است که t تعداد دفعات تکرار بدنه اصلی الگوریتم k-means می‌باشد. در خط ۹ یک حلقه وجود دارد که m بار تکرار می‌شود و داخل آن و در خطوط ۹-۱۱ فاصله هر رکورد مورد بررسی از رکوردهایی که داخل خوشه یکسان هستند، محاسبه می‌گردد. چنانچه $K=m$ باشد این خطوط ۱ بار اجرا و اگر $K=1$ باشد این خطوط m بار تکرار می‌شود. در خط ۱۳ فاصله‌های به دست آمده به صورت صعودی مرتب‌سازی می‌شود و در صورتی که شرط اول برقرار باشد، مرتبه اجرایی یک و چنانچه شرط دوم برقرار باشد مرتبه اجرایی $m\log m$ است. در خطوط ۱۶-۱۸ و ۲۶-۲۸ یک حلقه وجود دارد که K' مرتبه تکرار می‌شود که اگر شرط اول برقرار باشد $K'=1$ شده و این حلقه تکرار نمی‌گردد اما چنانچه شرط دوم برقرار باشد و $K'=m$ این حلقه m مرتبه تکرار می‌گردد. بنابراین مرتبه اجرایی این الگوریتم در بدترین حالت $O(km^2\log m)$ است. از آنجایی که روند تشخیص و تصحیح خطا، فرایند پیش‌پردازش محسوب شده و در پردازش اصلی مبتنی بر داده‌ها تأثیرگذار نیست، مرتبه اجرایی الگوریتم‌ها در فرایند تصحیح در پردازش اصلی تأثیرگذار نمی‌باشد.

٤- آزمایشات تجربی

در این بخش راهکار پیشنهادی بر روی سه مجموعه داده متفاوت از پایگاه داده UCI machine learning[1] آزمایش شده است. این سه مجموعه داده صحیح و عاری از خطا می‌باشد و برای پیاده‌سازی این راهکار از MATLAB R۲۰۱۴a استفاده شده است. مجموعه داده اول مربوط به مشتریان عمده‌فروشی[2] است که شامل ۴۴۰ رکورد و ۸ ویژگی می‌باشد. ۲ ویژگی از این منبع اسمی و مابقی ویژگی‌های عددی می‌باشد. مجموعه داده دوم مدل‌سازی دانش کاربران[3] است و حاوی ۴۰۰ رکورد و ۶ ویژگی است که ۱ ویژگی ترتیبی و مابقی عددی می‌باشد. مجموعه داده سوم درباره درآمد افراد بر اساس اطلاعات سرشماری[4] است که شامل ۴۸۸۴۲ رکورد و ۱۵ ویژگی می‌باشد که ۶ ویژگی عددی، ۱ ویژگی ترتیبی و ۸ ویژگی اسمی می‌باشد. همان طور که اشاره شد این راهکار تعداد محدودی از فیلدهای صحیح را به عنوان خطا نمایش می‌دهد. اگر کاربر هر رویکرد تصحیحی به جز حذف خطاها را به کار بندد، مشکلی برای رکوردهای صحیح که به عنوان خطا معرفی شده‌اند روی نمی‌دهد. به طور نمونه اگر از درخت تصمیم برای تصحیح استفاده کند، رکورد صحیح شناسایی‌شده به عنوان رکورد خطادار، در برگی قرار خواهد گرفت که یا عیناً مقدار قبلی خود رکورد است و یا این که به مقدار خودش بسیار نزدیک می‌باشد. اما چنانچه از رویکرد حذف استفاده نماید، یک سری از داده‌های صحیح را از دست خواهد داد.

در این مقاله با ۴ معیار عملکرد الگوریتم پیشنهادی مورد بررسی قرار گرفته است. پیش از بیان معیارها، پارامترهای مورد استفاده در آنها معرفی می‌گردد. N و P به ترتیب بیانگر تعداد فیلدهای حاوی خطا و تعداد فیلدهای صحیح می‌باشد. TN و TP به ترتیب تعداد فیلدهای خطا و صحیحی هستند که توسط الگوریتم به درستی برچسب‌گذاری شده‌اند. FP و FN نیز به ترتیب تعداد فیلدهای خطا و صحیحی می‌باشند که به وسیله الگوریتم به اشتباه برچسب‌گذاری شده است. حال به معرفی معیارها پرداخته می‌شود. در معیار اول که در (۱) آمده است، مقدار false alarm rate محاسبه شده که معادل تعداد خطاهایی که شناسایی نشده بخش بر تعداد کل خطاها می‌باشد. در معیار دوم که در (۲) آورده شده است، error rate از مجموع خطاهای شناسایی‌نشده و تعداد فیلدهای صحیح شناسایی‌شده به عنوان خطا، بخش بر تعداد کل رکوردها محاسبه می‌شود. با (۳) true negative rate محاسبه می‌شود که برابر است با تعداد خطاهایی که شناسایی شده بخش بر مجموع خطاهایی که شناسایی شده و خطاهایی که شناسایی نشده‌اند. در (۴) detection rate بیان شده که معادل با تعداد ویژگی‌های صحیحی که به درستی شناسایی شده بخش بر مجموع تعداد ویژگی‌های صحیحی که به درستی شناسایی و تعداد ویژگی‌های صحیحی که به عنوان خطا شناسایی شده است می‌باشد

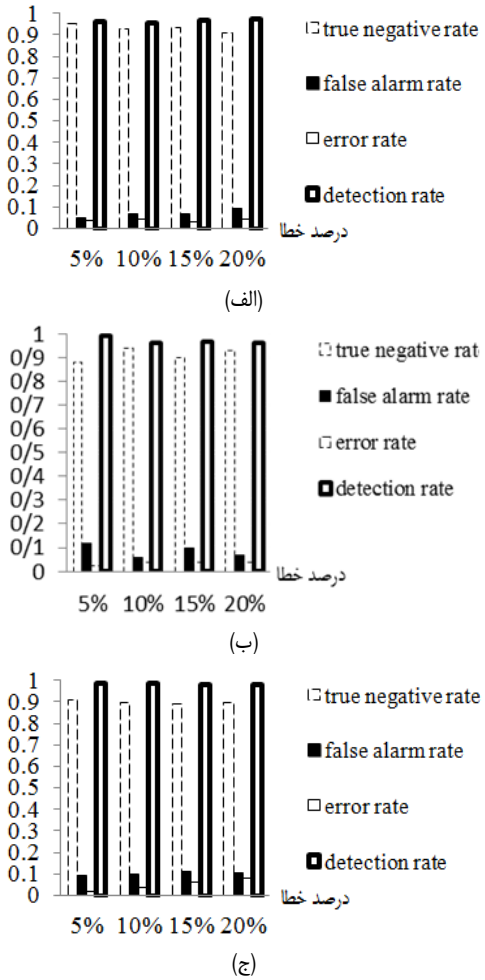
$$\text{false alarm rate} = \frac{FP}{FP+TN} \quad (۱)$$

$$\text{error rate} = \frac{FP+FN}{N+P} \quad (۲)$$

$$\text{true negative rate} = \frac{TN}{TN+FP} \quad (۳)$$

$$\text{detection rate} = \frac{TP}{TP+FN} \quad (۴)$$

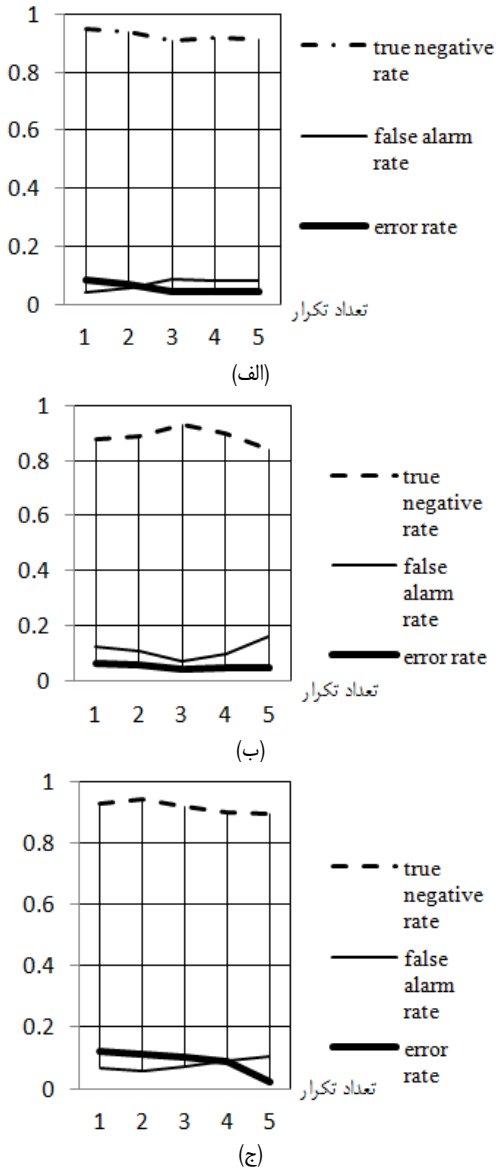
1. https://archive.ics.uci.edu/ml/datasets
2. https://archive.ics.uci.edu/ml/datasets/Wholesale+customers
3. https://archive.ics.uci.edu/ml/datasets/User+Knowledge+Moding
4. https://archive.ics.uci.edu/ml/datasets/Adult



شکل ۲: میزان true negative rate، false alarm rate، error rate و detection rate بر روی سه مجموعه داده به ازای نرخ خطای متفاوت، (الف) مجموعه داده Wholesale custmoers، (ب) مجموعه داده User Knowledge Moding و (ج) مجموعه داده Adult.

در مجموعه داده‌ها که فاقد خطا بوده‌اند با استفاده از توزیع نرمال، ۵٪، ۱۰٪، ۱۵٪ و ۲۰٪ خطا ایجاد شده و نحوه عملکرد با ۴ معیار محاسبه شده است. همان طور که در شکل ۲ نشان داده شده، در تمام نرخ‌های خطا به طور متوسط true negative rate ۹۱٪، detection rate ۹۷٪، false alarm rate ۸٪ و error rate ۵٪ بوده است. شکل ۳ نشان‌دهنده عملکرد راهکار پیشنهادی در نرخ خطای ۲۰٪ در مجموعه داده‌ها به ازای تکرارهای مختلف می‌باشد. مقدار true negative rate در تمام تکرارها بالای ۸۹٪ و میزان false alarm rate به طور متوسط ۹٪ بوده است. در تکرار ۱ میزان error rate بیش از ۱۲٪ است و با افزایش تعداد تکرارها این میزان به ۴٪ تنزل یافته است. بهترین توازن بین پارامترها برای دو مجموعه داده اول در تکرار ۳ و برای مجموعه داده سوم در تکرار ۵ رخ داده است.

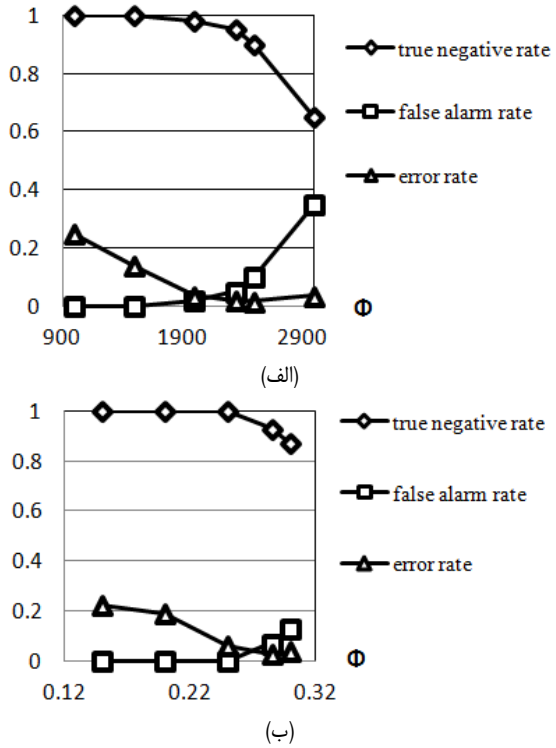
در ویژگی‌های عددی، مقدار Φ به عنوان تفاضل بین مقدار پیش‌بینی شده از k همسایه نزدیک با مقدار ویژگی در رکورد مورد بررسی، معرفی شده است. برای به دست آوردن بهترین مقدار برای Φ، در یک ویژگی عددی ۱۰٪ خطا ایجاد شده و true negative rate، false alarm rate و error rate به ازای مقادیر مختلف Φ مورد بررسی قرار گرفته است. شکل ۴ نشان‌دهنده مقادیر مختلف برای پارامتر Φ می‌باشد. همان طور که در شکل ۴ مشخص است، مقادیر پایین برای Φ، true negative rate ۱۰۰٪ را داشته است اما باعث شده که میزان error rate بالا باشد



شکل ۳: میزان true negative rate، false alarm rate و error rate بر روی سه مجموعه داده به ازای تکرارهای مختلف برای نرخ خطای ۲۰٪، (الف) مجموعه داده Wholesale custmoers، (ب) مجموعه داده User Knowledge Moding و (ج) مجموعه داده Adult.

که بیشترین میزان خطا ۲۵٪ بوده است. با افزایش مقدار Φ میزان error rate کاهش به سزایی داشته است. علاوه بر این true negative rate نیز از ۱۰۰٪ کاهش یافته ولی بیش از ۹۰٪ شده است. بهترین توازن بین true negative rate و error rate برای مجموعه داده اول در مقدار Φ، ۲۳۵۰ و برای مجموعه دوم در مقدار Φ، ۰٫۲۸۵ رخ داده است. از این مقدار بیشتر Φ کارایی کاهش نموده است و لذا این نقطه بهترین مقدار برای Φ در مجموعه داده‌ها می‌باشد. بر طبق آزمایشات انجام‌شده این بهترین مقدار Φ، مقداری بین میانگین و کواریانس آن ویژگی می‌باشد.

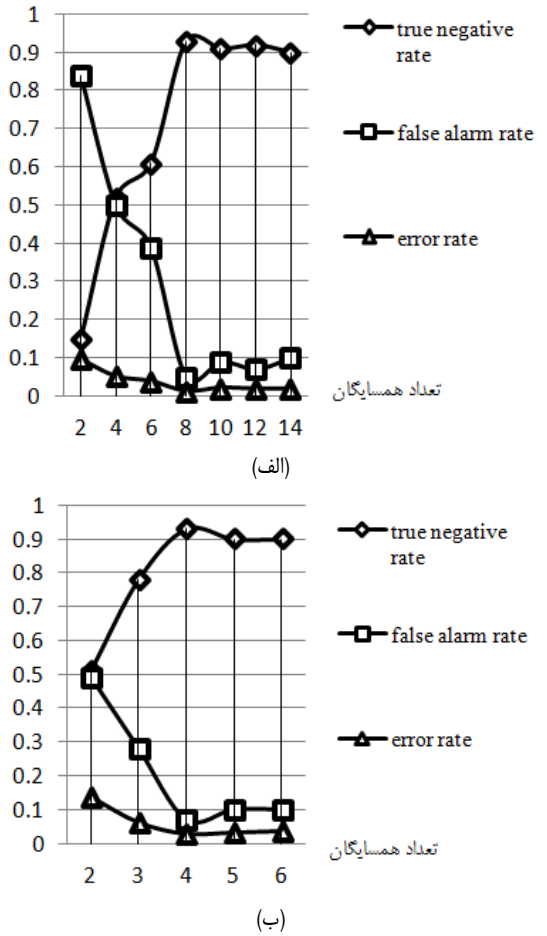
در شکل ۵ از مجموعه داده اول، ویژگی عددی مورد بررسی در شکل ۴ و از مجموعه داده دوم، یک ویژگی ترتیبی انتخاب شده است. در این شکل تعداد همسایگی‌های مختلف برای یافتن بهترین تعداد همسایگی تست شده است. به ازای همسایگی‌های مختلف میزان true negative rate، false alarm rate و error rate مورد بررسی قرار گرفته است. همان طور که در شکل مشخص شده است در ابتدا میزان



(الف)

(ب)

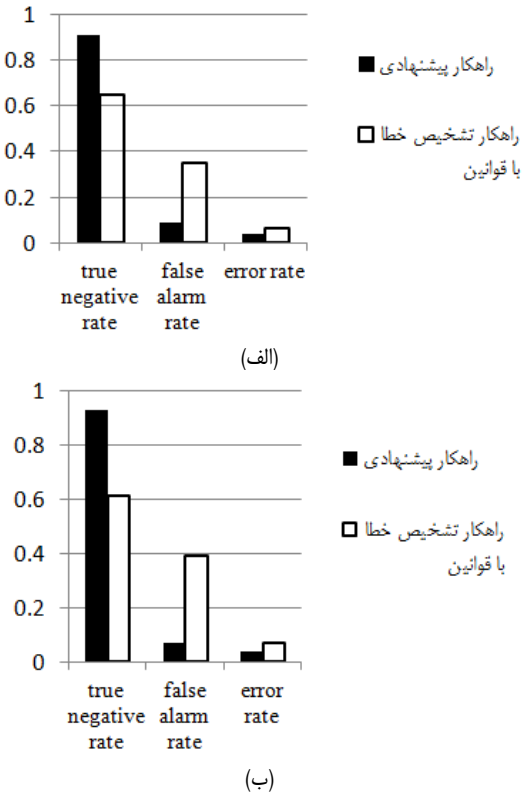
شکل ۴: بررسی مقدار Φ در یک ویژگی عددی به ازای ۱۰٪ خطا در true negative rate، false alarm rate و error rate، (الف) مجموعه داده Wholesale custmoers و (ب) مجموعه داده User Knowledge Moding.



(الف)

(ب)

شکل ۵: بررسی تأثیر تعداد همسایگان true negative rate، false alarm rate و error rate به ازای نرخ خطای ۱۰٪، (الف) مجموعه داده Wholesale custmoers و (ب) مجموعه داده User Knowledge Moding.



(الف)

(ب)

شکل ۶: مقایسه true negative rate، false alarm rate و error rate راه حل پیشنهادی و راه تشخیص با قوانین به ازای ۲۰٪ خطا، (الف) مجموعه داده Wholesale customers و (ب) مجموعه داده User Knowledge Moding.

جدول ۱: پارامترهای اولیه برای روش تشخیص خطا با قانون.

	minimum support	minimum confidence	α	β
مجموعه داده اول	۰/۰۳۴	۰/۹۸	۲۵۰	۱
مجموعه داده دوم	۰/۰۴۹	۰/۹۵	۳۱۰	۱

true negative rate پایین و میزان false alarm rate و error rate بالا می‌باشد. با افزایش همسایگی true negative rate افزایش و false alarm rate و error rate کاهش یافته است. بهترین توازن بین true negative rate، false alarm rate و error rate برای مجموعه داده اول در همسایگی ۸ و برای مجموعه داده دوم در همسایگی ۴ رخ داده که در آن true negative rate بیشترین حد خود و false alarm rate و error rate به میزان قابل قبولی کاهش یافته است.

در شکل ۶ روش پیشنهادی ارائه‌شده در این مقاله با بهترین مقادیر ورودی به دست آمده از آزمایشات با روش تشخیص خطا با استفاده از قوانین [۳] مورد مقایسه قرار داده شده است. جدول ۱ پارامترهای اتخاذشده برای روش تشخیص خطا با قوانین را نشان می‌دهد. این مقادیر متناسب با شرایطی که در [۳] برای رسیدن به بهترین نتایج پیشنهاد کرده، انتخاب شده است. ستون اول و دوم بیانگر minimum support و minimum confidence می‌باشد. در [۳] بیان شده است با توجه به این که خطاها از نظر تعداد کم هستند، قوانینی که بیش از یک حد آستانه نقض شده‌اند، حذف می‌شوند که ستون سوم نشان‌دهنده این حد آستانه می‌باشد. همچنین در روش تشخیص خطا با قانون [۳]، یک حد آستانه برای رکوردهای خطادار تعریف شده است. بدین صورت که رکوردی که بیش از تعدادی قانون را نقض کرده باشد به عنوان خطا معرفی می‌شود. ستون چهارم نیز بیانگر این مقدار می‌باشد. برای پیاده‌سازی روش

تشخیص خطا با قانون [۳] از زبان #c و محیط visual studio ۲۰۱۰ استفاده شده و برای انجام مقایسه در هر دو مجموعه داده، داده‌های پرت منبع برطرف شده و ۲۰٪ خطا با استفاده از توزیع نرمال ایجاد شده است.

همان طور که در شکل ۶ نشان داده شده روش پیشنهادی در هر سه معیار true negative rate، false alarm rate و error rate در هر دو مجموعه داده، از راهکار تشخیص خطای خودکار با قوانین [۳]، نتیجه بهتری داشته است. true negative rate به طور متوسط در روش پیشنهادی ۲۵٪ نسبت به راهکار تشخیص خطای خودکار با قوانین بهبود داشته است. false alarm rate و error rate نیز نسبت به راهکار تشخیص خودکار با قوانین به ترتیب ۲۵٪ و ۲٪ کاهش یافته است.

۵- نتیجه‌گیری

صحت داده‌ها به عنوان یکی از ابعاد حایز اهمیت در کیفیت داده‌ها به شمار می‌رود به گونه‌ای که تصمیم‌گیری بر مبنای داده‌های ناصحیح سازمان را متقبل هزینه‌های بالا و شکست می‌نماید. فرایند تصحیح از دو فاز تشخیص خطا و تصحیح خطاهای شناسایی‌شده تشکیل شده است. فرایند تصحیح به کشف ناسازگاری‌ها، مقادیر از دست رفته و تکرار می‌پردازد و خطاهای شناسایی‌شده را اصلاح می‌کند. با توجه به حجم بالای داده‌ها، یکی از پارامترهای مهم در الگوریتم‌های تصحیح، حذف نیاز به تعامل با کاربر در حین تصحیح و در واقع ارائه یک راهکار تصحیح خودکار می‌باشد. از این رو در این مقاله یک روش تشخیص خطای خودکار مبتنی بر خوشه‌بندی k - means پیشنهاد شد. در این روش ابتدا به ازای هر ویژگی، خوشه‌بندی انجام گرفت و سپس در هر خوشه، راهکاری شبه k نزدیک‌ترین همسایگی اجرا شد. این راهکار توانایی تشخیص خطا در انواع داده‌ای مختلف را دارا می‌باشد. همچنین این روش به دلیل این که ویژگی‌ها را به صورت مجزا مورد بررسی قرار می‌دهد، توانایی تشخیص چندین خطا در ویژگی‌های یک رکورد را دارا است. بنا بر آزمایشات انجام‌شده، این روش به طور متوسط دارای true negative rate ۹۱٪ می‌باشد و همچنین الگوریتم پیشنهادی نسبت به روش‌های مشابه ۲۵٪ true negative rate بالاتری دارد.

مراجع

[1] G. Beskales, I. F. Ilyas, L. Golab, and A. Galiullin, "Sampling from repairs of conditional functional dependency violations," *The VLDB Journal*, vol. 23, no. 1, pp. 103-128, Feb, 2014.

[2] W. Fan, "Dependencies revisited for improving data quality," in *Proc. 27th Int. Conf. on Management of Data,* pp. 159-170, Vancouver, Canada, 9-12 Jun. 2008.

[3] W. Ahmed Malik and A. Unwin, "Automated error detection using association rules," *Intelligent Data Analysis*, vol. 15, no. 5, pp. 749-761, Sept. 2011.

[4] P. H. Williams, C. R. Margules, and D. W. Hilbert, "Data requirements and data sources for biodiversity priority area selection," *J. of Biosciences*, vol. 27, no. 4, pp. 327-338, Jul. 2002.

[5] S. Bruggemann, "Rule mining for automatic ontology based data cleaning," in *Progress in WWW Research and Development*, pp. 522-527, 2008.

[6] G. Rahman and Z. Islam, "Missing value imputation using decision trees and decision forests by splitting and merging records: two novel techniques," *Knowledge-Based Systems*, vol. 53, pp. 51-65, Nov. 2013.

[7] G. Rahman and Z. Islam, "Decision tree-based missing value imputation technique for data pre-processing," *Research and Practice in Information Technology*, vol. 121, no. 1, pp. 41-50, Dec. 2011.

[8] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123-140, Aug. 1996.

[9] M. Yakout and L. Berti-Equille, and A. K. Elmagarmid, "Don't be SCAREd: use scalable automatic repairing with maximal likelihood and bounded changes," in *Proc. 13th Int. Conf. on Management of Data*, pp. 553-564, New York, USA, 22-27 Jun. 2013.

[10] N. Tang, "Big data cleaning," in *Proc. 16th Int. Conf.in Web Technologies and Applications,* pp. 13-24, Changsha, China, 5-7 Sept. 2014.

[11] J. Hipp, U. Guntzer, and U. Grimmer, "Data quality mining-making a virute of necessity," in *Proc. 6th Int. SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD'01,* pp. 52-57, Santa Barbara, California, USA, May, 2001.

[12] C. He, Z. Tan, Q. Chen, C. Sha, Z. Wang, and W. Wang, "Repair diversification for functional dependency violations," in *Proc. 19th Int. Conf.in Database Systems for Advanced Applications,*, pp. 468-482, Bali, Indonesia, 21-24 April, 2014.

[13] M. Hamad and A. Abdulkhar Jihad, "An enhanced technique to clean data in the data warehouse," in *Proc. 11thInt. Conf. in Developments in E-systems Engineering,* pp. 306-311, Washington, DC, USA, 6-8 Dec. 2011.

[14] C. Teng, "Correcting noisy data," in *Proc. 16th Int. Conf.in Machine Learning,*, pp. 239-248, San Francisco, CA, USA, 27-30 Jun. 1999.

[15] C. Teng, "A comparison of noise handling techniques," in *Proc. 14th Int. Florida Artificial Intelligence Research Society,* pp. 269-273, Key West, FL, USA, 21 – 23 May, 2001.

[16] C. Teng, "Polishing blemishes: issues in data correction," *Intelligent Systems*, vol. 19, no. 2, pp. 34-39, Mar. 2004.

[17] A. Lopatenko and L. Bravo, "Efficient approximation algorithms for repairing inconsistent databases," in *Proc. IEEE 23rd Int. Conf. on Data Engineering, ICDE'07*, pp. 216-225, 15-20 Apr. 2007.

[18] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85-126, Oct. 2004.

[19] S. Chawla and A. Gionis, "k-means: a unified approach to clustering and outlier detection," in *Proc. 13th SIAM Int. Conf. on Data Mining,* pp. 189-197, Austin, Texas, USA, 2-4 May 2013.

[20] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Perez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognition*, vol. 46, no. 1, pp. 243-256, Jan. 2013.

[21] P. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *J. of Computational and Applied Mathematics*, vol. 20, no. 1, pp. 53-65, Nov. 1987.

[22] J. Han, M. Kamber, and J. Pei, *Data Mining Concept and Technieques*, pp. 451-471, 3 Edition, 2011.

مهدیه عطاییان دانشجوی مهندسی کامپیوتر- نرم‌افزار در دانشگاه تربیت دبیر شهید رجایی می‌باشد. نام‌برده تحصیلات خود را در مقطع کارشناسی مهندسی فناوری اطلاعات(IT) در سال ۱۳۹۳ با کسب رتبه چهارم در دانشگاه سراسری سمنان به اتمام رسانده است، و در سال ۱۳۹۳ بدون کنکور با شرایط استعداد درخشان در مقطع کارشناسی ارشد دانشگاه تربیت دبیر شهید رجایی پذیرفته شده است. زمینه‌های تحقیقاتی ایشان عبارتند از: تصحیح داده‌ها، پیش‌پردازش داده‌ها و داده کاوی.

نگین دانشپور استادیار دانشکده مهندسی کامپیوتر دانشگاه تربیت دبیر شهید رجایی می‌باشد. نام‌برده تحصیلات خود را در مقطع کارشناسی مهندسی کامپیوتر- سخت‌افزار در سال ۱۳۷۸ با کسب رنبه اول در دانشگاه شهیدبهشتی، و کارشناسی ارشد مهندسی کامپیوتر- نرم‌افزار در سال ۱۳۸۱ در دانشگاه صنعتی امیرکبیر به پایان رسانده است، و در سال ۱۳۸۹ دکتری خود در رشته مهندسی کامپیوتر- نرم‌افزار را از دانشگاه صنعتی امیرکبیر اخذ کرده است. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: پایگاه داده تحلیلی، سیستم‌های تصمیم‌یار، پیش پردازش داده‌ها، و داده کاوی.