تصحیح خودکار داده‌ها مبتنی بر وابستگی تابعی و سیستم یادگیری مرکب

مهدیه عطاییان[۱]، کارشناس ارشد؛ نگین دانشپور[۲]، استادیار

۱- دانشکده مهندسی کامپیوتر - دانشگاه تربیت دبیر شهید رجایی- تهران - ایران - m.ataeyan@sru.ac.ir
۲- دانشکده مهندسی کامپیوتر - دانشگاه تربیت دبیر شهید رجایی - تهران - ایران - ndaneshpour@sru.ac.ir

چکیده: صحت داده‌ها یکی از مهم‌ترین ابعاد کیفیت داده‌ها به‌شمارمی‌رود. با توجه به حجم بالای منابع داده‌ای نیاز به روش‌هایی خودکار وجود دارد. در این مقاله راهکاری خودکار برای تصحیح داده‌هایی با انواع داده‌ای متفاوت ارائه شده است. در این راهکار در ابتدا رکوردهایی که احتمالاً حاوی ویژگی خطا است با استفاده از وابستگی تابعی شناسایی‌می‌گردد، بدین‌صورت که رکوردی که به ازای یک وابستگی تابعی با بیش از δ از رکوردها در تناقض باشد، مشکوک به خطا است. سپس به ازای هر ویژگی از منبع داده مورد بررسی، سیستم یادگیری مرکب ساخته‌می‌شود. سیستم یادگیری مرکب از سه طبقه‌بند بیز، درخت تصمیم و شبکه عصبی MLP تشکیل‌شده است و دارای استراتژی ترکیب رأی اکثریت است. سیستم یادگیری مرکب به‌وسیله رکوردهای صحیح شناسایی‌شده مورد آموزش قرارداده می‌شود. پس از آموزش طبقه‌بندها، هر ویژگی غلط به‌عنوان کلاس هدف سیستم یادگیری‌مرکب قرارمی‌گیرد و مقداری برای آن پیش‌بینی‌می‌گردد. روش پیشنهادی قادراست چندین خطا در یک رکورد را شناسایی نماید. آزمایش‌ها نشان‌می‌دهد که true negative rate الگوریتم پیشنهادی در بخش تشخیص خطا به‌طور متوسط ۹۳/۷٪ و در بخش تصحیح خطا به‌طور متوسط ۹۰/۶٪ است. هم‌چنین آزمایش‌ها نشان‌می‌دهد که میزان پارامترهای ارزیابی در الگوریتم پیشنهادی در مقایسه با دو الگوریتم مشابه مبتنی بر وابستگی تابعی بهبود داشته است.

واژه‌های کلیدی: تصحیح داده، تشخیص خطا، وابستگی تابعی، سیستم یادگیری مرکب.

Automatic Data Cleaning using Functional Dependency and Ensemble Learning

M. Ataeyan[1], M.Sc; N. Daneshpour[2], Assistant Professor

1- Faculty of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran, Email: m.ataeyan@sru.ac.ir
2- Faculty of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran, Email: ndaneshpour@sru.ac.ir

Abstract: Data accuracy is one of the important aspects of data quality. According to high volume of data sources an automatic method is required. In this article an automatic method is proposed for cleaning of data with various data types. Initially, records that may contain incorrect attributes are detected using functional dependency, so that each record that inconsistent more than δ records for one functional dependency, probably is error. Thereafter ensemble learning is done for each attribute of data source. Ensemble learning contains 3 classifiar naïve bayes, decision tree and MLP, and voting is combination strategy. It is trained using correct records that identified in the initial steps. After training, each incorrect attribute is placed as target and predict values for it. Proposed method is able to clean data in data sources with different data types. Experiments show the true negative rate in detecting error part of the proposed algorithm is averagely 93.7% and in cleaning error part is averagely 90.6%. Also experiments show that evaluation parametrs are improved in proposed method compared with 2 similar methods.

Keywords: Data cleaning, error detection, functional dependency, ensemble learning.

تاریخ ارسال مقاله: ۱۳۹۵/۰۹/۱۰
تاریخ اصلاح مقاله: ۱۳۹۶/۰۱/۲۰
تاریخ پذیرش مقاله: ۱۳۹۶/۰۵/۰۹
نام نویسنده مسئول: نگین دانشپور
نشانی نویسنده مسئول: ایران - تهران - لویزان- خیابان شعبانلو - دانشگاه تربیت دبیر شهید رجایی - دانشکده مهندسی کامپیوتر.

۱- مقدمه

امروزه سازمان‌ها جهت اتخاذ تصمیمات خود نیاز به جمع‌آوری داده، ذخیره‌سازی و تجزیه و تحلیل آن دارند. داده جمع‌آوری شده اغلب شامل داده‌های ناصحیح و مقادیر ازدست‌رفته[۱] است [۱]. به همین دلیل تصمیم‌گیری مبتنی بر این‌گونه داده‌ها موجب هزینه‌های بالا و درنهایت شکست آن‌ها می‌گردد. بررسی‌ها نشان‌داده که ۳۰٪-۸۰٪ زمان توسعه پروژه‌های پایگاه داده تحلیلی[۲] صرف تصحیح داده‌ها می‌گردد [۲]. علاوه‌براین گزارش‌ها منتشرشده نشان‌می‌دهد سازمان‌های آمریکایی سالانه چندین تریلیون دلار به دلیل داده‌های ناصحیح متحمل هزینه می‌گردند [۲]. به همین علت، علی‌رغم زمان و هزینه بالا جهت تصحیح داده‌ها، فرایند تصحیح ضروری و غیرقابل اغماض است [۳].

به‌طورکلی فرایند پیش‌پردازش شامل انتساب مقدار به مقادیر ازدست‌رفته، شناسایی داده‌های پرت[۳]، برطرف نمودن داده‌های نویزی و اصلاح داده‌های ناسازگار[۴] است [۴]. این مشکلات ممکن است به دلایلی از قبیل معیوب بودن تجهیزات جمع‌آوری اطلاعات، اشتباه کاربر و یا رایانه در هنگام واردنمودن اطلاعات و یا هنگام انتقال داده‌ها ایجاد شوند. علاوه‌براین به دلیل فرمت‌های ناهمگون و نام‌گذاری‌های ناسازگار، در طی فرآیند جمع‌آوری داده از منابع توزیع‌شده نیز ممکن است با این مشکلات و یا مشکل رکوردهای تکراری[۵] مواجه‌شوند. درواقع فرایند پیش‌پردازش نقش اساسی در حصول اطمینان از کیفیت داده‌ها[۶] دارد [۴].

تصحیح داده‌ها[۷]، اغلب پردازشی دو مرحله‌ای است، که در مرحله اول داده‌های خطادار شناسایی و در مرحله دوم مقداری صحیح به داده خطادار انتساب داده‌می‌شود[۵]. تاکنون روش‌های متعددی برای تصحیح داده‌ها ارائه‌شده که در این راهکارها از وابستگی تابعی شرطی[۸] [۲]، طبقه‌بندها [۱، ۴، ۶، ۷]، قوانین[۹] [۸،۹] ، هستان‌شناسی[۱۰] [۱۰]، وابستگی‌های تابعی[۱۱] [۱۲،۱۱] و قوانین فازی [۱۳] استفاده‌شده است. قابل‌ذکر است که از این راهکارها در حوزه‌های دیگر از جمله صنایع نفت و گاز [۱۴]، پردازش تصویر[۱۵] و علوم پزشکی[۱۶] استفاده‌می‌شود. این روش‌ها از دیدگاه‌هایی ازجمله خودکار یا نیمه خودکار[۱۲] بودن، نوع داده‌ای[۱۳] مورد تصحیح، دارا بودن فاز تشخیص خطا و تفکیک‌پذیری فاز تشخیص و تصحیح خطا با یکدیگر تمایز دارند. قابل‌ذکر است که با توجه به این‌که تصحیح داده‌ها جز فرایند پیش‌پردازش است که زمان در آن فاقد اهمیت است، ازاین‌رو هیچ‌یک از راهکارهایی که تاکنون ارائه‌شده‌اند، از بعد پیچیدگی زمانی با یکدیگر مورد مقایسه قرار نگرفته‌اند.

راهکارهای پیشین دارای معایبی می‌باشند از جمله این‌که برخی از آن‌ها نیمه خودکار می‌باشند [۱۱، ۱۰]. درواقع این راهکارها حین تصحیح، نیازمند تعامل با کاربر هستند که در حال حاضر با توجه به حجم بالای منابع داده‌ای، استفاده از چنین روش‌هایی برای تصحیح، امکان‌پذیر نیست. از معایب دیگر برخی از روش‌ها نوع داده‌ای است که آن‌ها قادر به تصحیح می‌باشند. این روش‌ها تنها توانایی تصحیح منابع داده‌ای با نوع داده‌ای خاص را دارند [۱، ۴]، اما اغلب منابع داده‌ای موجود دارای ویژگی‌هایی با انواع داده‌ای گوناگون هستند. ازاین‌رو روشی کارآمد است که توانایی تصحیح تمام انواع داده‌ای را دارا باشد. علاوه بر این برخی از روش‌های موجود تنها توانایی تشخیص و تصحیح یک خطا را دارند، درحالی‌که در دنیای واقعی امکان بروز بیش از یک خطا در یک رکورد وجود دارد. هم‌چنین در برخی از راهکارهای ارائه‌شده، فاز تشخیص و تصحیح غیرقابل‌تفکیک است، لذا برای کاربر امکان تعامل در صورت دلخواه پس از فاز تشخیص وجود ندارد [۱۹-۱۷]. ازاین‌رو در این مقاله روشی خودکار برای تشخیص و تصحیح خطا ارائه‌شده‌است که توانایی تشخیص خطا در مجموعه داده‌ای با انواع داده‌ای متفاوت را دارا است. هم‌چنین این روش توانایی تشخیص بیش از یک خطا را در یک رکورد دارا است. در روش پیشنهادی فاز تشخیص و تصحیح کاملاً تفکیک‌پذیر بوده و لذا کاربر در صورت تمایل می‌تواند پس از تشخیص خطا هر رویکرد دلخواهی را برای هر خطا اتخاذ نماید.

راهکار پیشنهادی در این مقاله از دو فاز تشخیص خطا و تصحیح آن تشکیل شده است. در فاز تشخیص ابتدا رکوردهای مشکوک به خطا با استفاده از وابستگی‌های تابعی از رکوردهای صحیح تشخیص داده‌می‌شوند، بدین‌صورت‌که رکوردی که به ازای یک وابستگی تابعی با بیش از δ از رکوردها در تناقض باشد، مشکوک به خطا است. سپس به‌وسیله رکوردهای صحیح و وابستگی‌های تابعی محل وقوع خطاها در هر رکورد مشکوک به خطا تعیین‌می‌شود. برای انجام این‌کار تمام مقادیری که به ازای هر وابستگی تابعی در رکوردهای صحیح وجود دارد، استخراج‌می‌شود. سپس براساس این مقادیر تعداد دفعاتی که هر ویژگی وابستگی‌های تابعی را نقض می‌کند، محاسبه‌می‌شود. براساس نظر کاربر حداکثر تعداد خطا در هر رکورد (β) مشخص‌می‌شود. سپس β ویژگی که به‌دفعات بیشتری مقادیر ذخیره‌شده برای هر وابستگی تابعی را نقض‌می‌کنند، به‌عنوان خطا انتخاب می‌شوند.

در فاز تصحیح راهکار پیشنهادی، خطای شناسایی‌شده در فاز قبل به‌وسیله سیستم یادگیری مرکب[۱۴] اصلاح‌می‌شود. در راهکار پیشنهادی سیستم یادگیری مرکب شامل ۳ طبقه بند، درخت تصمیم[۱۵]، طبقه‌بند بیز[۱۶] و شبکه عصبی MLP است و از رأی اکثریت به‌عنوان استراتژی ترکیب در آن استفاده شده است. به ازای هر ویژگی یک سیستم یادگیری مرکب ساخته و طبقه‌بندهای آن به‌وسیله رکوردهای فاقد خطا، آموزش‌داده‌می‌شوند. پس از مرحله آموزش، هر ویژگی خطادار به‌عنوان کلاس هدف انتخاب و توسط سیستم یادگیری مرکب مربوطه تصحیح می‌شوند.

تاکنون راهکارهای متعددی از وابستگی تابعی برای تصحیح داده‌ها استفاده نموده‌اند؛ ولی در راهکار پیشنهادی این مقاله از ترکیب وابستگی تابعی و سیستم یادگیری مرکب استفاده شده است که به نسبت روش‌هایی که فقط از وابستگی تابعی برای تصحیح استفاده نموده‌اند کارایی بالاتری داشته است.

بخش‌های بعدی مقاله بدین شرح است. در بخش دوم مروری بر راهکارهای پیشین در حوزه تصحیح داده‌ها صورت می‌گیرد. در بخش چهارم راهکار پیشنهادی به‌طور کامل تشریح‌می‌شود و در بخش پنجم

آزمایش‌ها انجام‌گرفته بر روی مجموعه داده‌ای با انواع داده‌ای اسمی، ترتیبی و عددی بیان‌می‌شود. علاوه‌بر این در بخش پنجم روش پیشنهادی با روشی مشابه مورد مقایسه‌قرارگرفته‌است. در بخش پایانی خلاصه و نتیجه‌گیری از مقاله ارائه‌می‌گردد.

۲- پیشینه تحقیق

پیش از آن‌که داده‌های دنیای واقعی مورد استفاده‌قرارگیرند، باید از کیفیت آن‌ها اطمینان حاصل‌گردد؛ ازاین‌رو تأمین کیفیت، پیش‌پردازش محسوب‌می‌شود. برای تأمین صحت داده‌ها باید مقادیر ازدست‌رفته، ناسازگاری‌ها و داده‌های ناقص شناسایی و تصحیح گردد. برای تصحیح داده‌ها رویکردهای متفاوتی ارائه‌شده‌است که در این بخش به‌اختصار به شرح برخی از آن‌ها پرداخته‌می‌شود.

در [۱۰] راهکاری متعامل با کاربر مبتنی بر هستان‌شناسی برای تصحیح داده‌ها ارائه‌شده‌است. در مرحله اول جهت شناسایی خطاها از هستان‌شناسی استفاده‌می‌شود، سپس مقادیر صحیح از هستان‌شناسی استخراج و جهت انتخاب به کاربر نمایش داده‌می‌شود. پس از انتخاب کاربر، این رفتار او در سیستم ذخیره‌می‌شود. زمانی‌که انتخاب‌های کاربر به یک حد آستانه رسید، از رفتار او قوانین استخراج و ازاین‌پس برای تصحیح از قوانین استفاده‌می‌شود. این روش به دلیل تعامل با کاربر در بخش ابتدایی در حجم بالای داده‌ای امکان‌پذیر نیست.

روش DMI [۴] و SiMI [۱] برای انتساب مقدار صحیح به رکوردهایی که مقادیر خود را ازدست‌داده‌اند، ارائه‌شده است. این دو الگوریتم بر پایه الگوریتم EM هستند با این تفاوت که SiMI بهبودیافته الگوریتم DMI است. در الگوریتم DMI پس از تفکیک داده‌های صحیح و داده‌های حاوی مقادیر ازدست‌رفته، برای هر ویژگی از مجموعه داده مورد بررسی، درخت تصمیمی ایجاد می‌شود. برای تعیین مقدار ویژگی در برگ‌ها چنان‌چه ویژگی عددی باشد از الگوریتم EM و در غیر این صورت از رأی اکثریت در آن برگ استفاده‌می‌گردد. سپس هر ویژگی که مقدار خود را ازدست داده است به‌وسیله درخت متناظر آن تصحیح می‌گردد. در الگوریتم SiMI که توسعه‌یافته DMI است، از جنگل تصمیم استفاده شده است. در این روش نیز به ازای هر ویژگی یک جنگل تصمیم ساخته می‌شود و به جستجوی تقاطع میان برگ‌ها می‌پردازد. سپس به ادغام تقاطع‌ها با هدف بیشنه نمودن شباهت بین رکوردها پرداخته و با انتساب هر رکورد دارای مقدار ازدست‌رفته به یک تقاطع آن را تصحیح می‌نماید. این دو الگوریتم تنها قادر به تصحیح ویژگی‌هایی با دو نوع داده‌ای عددی و دسته‌ای می‌باشند.

الگوریتم SCARE با استفاده از الگوریتم‌های یادگیرنده و درست‌نمایی به تصحیح خطا پرداخته‌است [۷]. هدف در این الگوریتم حفظ ارتباط بین ویژگی در هنگام تصحیح است. هم‌چنین در این الگوریتم برخلاف برخی روش‌ها که هدف تنها کمینه‌کردن تغییرات است، علاوه‌بر محدود نمودن تغییرات، بیشینه‌کردن درست‌نمایی نیز درنظرگرفته‌شده‌است. روش تصحیح مقادیر ناصحیح نیز به‌وسیله مقادیر صحیح است. برای استفاده از درست‌نمایی از دو معیار برای بیشینه کردن سود و کمینه کردن هزینه تغییرات بهره‌گرفته‌شده‌است. در این روش به ازای هر ویژگی یک طبقه‌بند ایجاد و مقدار هر ویژگی غلط پیش‌بینی‌می‌شود. در مرحله بعد برای هر رکورد یک گراف که گره‌ها مقادیر پیش‌بینی‌شده و یال‌ها وابستگی میان ویژگی‌ها را نشان‌می‌دهد، رسم‌می‌شود. پس از رسم گراف تا زمانی‌که از هر ویژگی تنها یک مقدار باقی‌بماند، گره‌ای که یال‌های آن دارای کمترین وزن‌هستند، حذف می‌شود. این الگوریتم می‌تواند چندین ویژگی ناصحیح در یک رکورد را پیش‌بینی‌نماید، به‌علاوه این‌که در حین پیش‌بینی وابستگی بین داده‌های صحیح و داده‌های ناسازگار را به‌صورت مجزا درنظرمی‌گیرد. از معایب این روش این است که در این روش فرض شده که خطا تشخیص داده‌شده و فقط به تصحیح پرداخته‌است.

الگوریتم ارائه‌شده در [۸] با استفاده از محدودیت‌ها به تصحیح یک منبع ناسازگار می‌پردازد. این محدودیت‌ها از سه بخش اصلی تشکیل و فرم عمومی آن‌ها به‌صورت $((X,t_p[X]),(B,t_p^-[B])) \rightarrow t_p^+[B]$ می‌باشد. X بیانگر ویژگی‌هایی است که با یکدیگر در ارتباط هستند، $t_p[X]$ مقادیر آن ویژگی‌ها، B ویژگی‌هایی است که برای آن الگوهای منفی $t_p^-[B]$ ارائه‌شده‌است و $t_p^+[B]$ مقادیر صحیح ارائه‌شده برای مقادیر منفی است. در ابتدا هر یک از ویژگی‌های ارائه‌شده در بخش الگوهای شاهد، به‌تنهایی به‌عنوان یک کلید انتخاب‌می‌شود و هر یک با مقادیر خود در یک لیست با نام قانون ذخیره‌می‌شوند. به ازای هر رکورد موجود در منبع بررسی می‌شود که آیا از میان مقادیر کلیدهای ذخیره‌شده در لیست با رکورد مورد نظر تطابقی یافت‌می‌شود. درصورتی‌که تطابقی یافته نشد، نشانگر این است که این رکورد دارای مقادیر صحیح است. اما درصورتی‌که تطابقی یافت‌گردید باید آن قانون بازیابی و از قسمت الگو صحیح آن رکورد تصحیح گردد. به ازای یک رکورد باید تمامی کلیدها مورد بررسی قرارگیرند. 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 قرارگیرد.

در [۱۲] مدلی برای تصحیح داده‌ها مبتنی بر وابستگی‌های تابعی معرفی‌شده‌است. در این راهکار وابستگی‌های تابعی استخراج‌شده ثابت نیستند بلکه هم‌زمان با تصحیح داده‌ها ممکن است وابستگی‌های تابعی نیز تغییر نمایند. درواقع زمانی‌که تعداد زیادی از رکوردها یک وابستگی تابعی را نقض‌می‌کنند، با اضافه‌نمودن یک ویژگی به وابستگی‌های تابعی به‌گونه‌ای که سازگاری رکوردها با آن برقرارشود، تصحیح‌انجام‌می‌گیرد. هم‌چنین این روش هنگامی‌که دو رکورد یک وابستگی تابعی را نقض‌می‌کنند، تنها رویکرد تصحیح ویژگی سمت راست وابستگی تابعی را به‌کار نمی‌گیرد. این روش به جستجوی رکوردی که فاصله کمی با یکی از دو رکورد متناقض دارد و هم‌چنین وابستگی تابعی مورد بررسی را نقض‌نمی‌کند، می‌پردازد. در صورت یافتن این رکورد بر اساس تابع

هزینه تصمیم‌گیری می‌نماید که به تصحیح ویژگی سمت راست بپردازد و یا این‌که ویژگی سمت چپ را بر اساس رکورد یافته‌شده، اصلاح‌نماید.

در [۱۹-۱۷] هدف شناسایی ویژگی‌های مشکوک به خطا و اصلاح آن به‌وسیله الگوریتم *polishing* است. این الگوریتم از دو فاز پیش‌بینی و تنظیم مقادیر تشکیل‌شده‌است. در این الگوریتم از یک‌رویه به‌صورت بازگشتی استفاده‌می‌شود، که ترکیب‌های مختلف از مقادیر مختلف ویژگی‌ها را امتحان‌می‌نماید. در فاز پیش‌بینی از میان الگوریتم‌های طبقه‌بندی، یک روش انتخاب‌می‌شود. به ازای هر ویژگی ۱۰ دسته بر روی داده‌ها ایجادمی‌کند و برچسب آن ویژگی به‌عنوان هدف درنظرگرفته‌می‌شود. مقدار ویژگی هدف در هر رکورد به ازای ۱۰ طبقه‌بند پیش‌بینی‌می‌شود. درصورتی‌که مقدار اصلی آن ویژگی در رکورد مورد نظر با مقادیر پیش‌بینی شده‌در تناقض‌باشد، مقادیر حاصل از پیش‌بینی برای آن رکورد ذخیره‌می‌گردد. این مقادیر در فاز بعدی جهت تصحیح مورد استفاده‌قرارمی‌گیرد. در فاز تنظیم مجدداً به ازای هر ویژگی ۱۰ دسته بر روی داده‌ها ایجادمی‌شود. سپس به ازای هر ویژگی متناقض شناسایی شده در فاز قبل، مقدار آن براساس مقادیر پیش‌بینی‌شده و توسط ۱۰ طبقه‌بند تصحیح‌می‌شود. مقدار جایگزین باید از میان مقادیر پیش بینی شده در مرحله قبل انتخاب شود. باید توجه‌نمود که در این روش تمامی مقادیر تغییر نمی‌کند، بلکه تنها مقادیری که دارای مقدار غیرطبقه‌بندی هستند، دچار تغییرمی‌گردند، به‌علاوه این‌که سعی‌می‌شود تغییرات را به حداقل برساند. در این روش نیز فاز تصحیح و تشخیص غیر قابل تفکیک می‌باشند و لذا کاربر نمی‌تواند رویکرد دیگری را برای فاز تصحیح به‌کارگیرد.

الگوریتم [۱۱] تصحیحی مبتنی بر وابستگی تابعی ارائه‌نموده و برای انتخاب بهترین تصحیح از دو تابع هزینه و تنوع استفاده‌کرده‌است. برای محاسبه میزان تنوع از تابع فاصله استفاده‌می‌شود که میزان عدم شباهت رکوردها را محاسبه‌می‌نماید. برای تعیین فاصله از وزن‌های اختصاص‌داده‌شده به فیلدها که درواقع میزان اعتماد کاربر به آن‌ها است، استفاده‌می‌گردد. در این روش برای محاسبه میزان تنوع از دو تابع استفاده‌شده‌است. در تابع تنوع اول تفاضل بین مجموع هزینه را با مجموع فاصله برای تمام تصحیح‌ها ارائه‌شده، محاسبه‌می‌کند. در این تابع پارامتری برای تعادل بین هزینه و فاصله وجود دارد. زمانی که این پارامتر افزایش می‌یابد، برای بهینه‌شدن تابع تنوع مقدار تابع هزینه نیز باید افزایش‌یابد. مقدار بین ۰-۰/۳ برای این پارامتر پیشنهادمی‌شود. تابع تنوع دوم تفاضل بین حداکثر هزینه و حداقل فاصله را بیان‌می‌کند. سپس این روش با استفاده از الگوریتمی از میان K تصحیح بهینه با هدف کمینه نمودن دو تابع تنوع که در واقع حداقل نمودن بیشینه هزینه و حداکثر نمودن کمینه فاصله است، یک تصحیح را به‌عنوان تصحیح بهینه معرفی‌می‌کند. این الگوریتم به دلیل تعامل با کاربر برای تعیین وزن‌های اولیه در حجم بالای داده‌ای مناسب نیست.

در [۲۰] یک الگوریتم تقریبی برای تصحیح مجموعه داده ناسازگار مبتنی بر مجموعه وابستگی‌های تابعی ثابت بیان‌شده‌است. هدف این الگوریتم یافتن یک مجموعه سازگار است که فاصله آن از مجموعه اولیه، نزدیک به فاصله تصحیح بهینه از مجموعه اولیه باشد. در واقع در این روش هدف به‌دست‌آوردن رکوردهایی است که با وابستگی‌های تابعی تعریف شده در تناقض می‌باشند و اصلاح آن‌ها به‌وسیله وابستگی تابعی به نحوی است‌که دارای کمترین فاصله از مجموعه‌داده اولیه باشد. این الگوریتم به جستجوی مقادیری که وابستگی‌های تابعی تعریف‌شده را نقض‌می‌کنند، می‌پردازد و آن‌ها را در یک گراف قرارمی‌دهد. در این گراف برای مجموعه ناسازگاری‌ها، گره‌ها مقادیر ناسازگار و یال‌ها وابستگی بین مقادیر را تعیین می‌کنند. پس از این‌که تمامی گراف‌ها محاسبه‌گردید؛ الگوریتم، کوچک‌ترین گراف همپوشان را پیدامی‌کند. سپس به ازای هر دو رکورد t_1 و t_2 که هر وابستگی تابعی را نقض‌می‌کند، درصورتی‌که (t_1, a) تنها مقدار در گراف همپوشان باشد، مقدار t_2 بر اساس t_1 تصحیح‌می‌شود و t_1 را از گراف همپوشان حذف‌می‌نماید. پس از این‌که تصحیح انجام‌شد، امکان‌دارد در اثر رفع برخی تناقض‌ها در رکوردها اشکال ایجاد شود و مجدداً این اشکال مورد بررسی قرارمی‌گیرد.

در ابتدا وابستگی‌های تابعی بر اساس امتیازی که از میانگین دو پارامتر حاصل‌می‌شود، در یک لیست به‌صورت نزولی مرتب‌می‌گردند. پارامتر اول درجه ناسازگاری هر وابستگی تابعی را با منبع داده مشخص‌می‌کند و پارامتر دوم میزان تناقض وابستگی‌های تابعی دیگر با وابستگی تابعی درنظرگرفته‌شده را تعیین‌می‌کند. وابستگی‌های تابعی به ترتیب از لیست برداشته‌می‌شوند. این الگوریتم به‌صورت حریصانه عمل‌می‌کند. میزان هزینه تصحیح تناقض با وابستگی تابعی مورد نظر محاسبه‌می‌شود؛ سپس هزینه تصحیح ناسازگاری وابستگی تابعی مورد نظر با وابستگی‌های دیگر مورد محاسبه قرارمی‌گیرد. تصحیحی که کمترین هزینه را داشته‌باشد، اعمال می‌شود و این روند برای وابستگی تابعی دیگر اجرا می‌شود.

در [۲۱] الگوریتمی ارائه‌شده که به‌طور هم‌زمان به تصحیح وابستگی‌های تابعی و داده‌ها می‌پردازد. در این الگوریتم به ازای هر وابستگی تابعی که توسط رکوردها نقض‌می‌شود، تصحیح‌های متنوعی برای وابستگی تابعی و داده‌ها ارائه‌می‌شود، به‌نحوی‌که دارای کمترین تغییرات باشند. این تصحیح‌ها به کاربر نمایش‌داده‌می‌شود، تا او با توجه به دانشی که نسبت به مجموعه داده‌ها دارد، بهترین تصحیح را انتخاب‌نماید. هدف اصلی در این روش محدود نمودن حداکثر تعداد فیلدهایی است که در منبع داده دچار تغییر می‌گردند و به‌دست‌آوردن مجموعه‌ای از وابستگی‌های تابعی است که به مجموعه اولیه وابستگی تابعی نزدیک و تناقضی با مجموعه داده‌ها نداشته‌باشد. در این راهکار ابتدا به کمک گراف تناقض، مجموعه داده و وابستگی‌های تابعی اولیه به جستجوی تصحیحاتی برای وابستگی‌های تابعی می‌پردازد. این روش برای یافتن مجموعه وابستگی تابعی بهینه، گراف تناقض را با ترکیب A^* و یک الگوریتم تجربی حرص‌می‌نماید. پس از اصلاح وابستگی تابعی، به ازای مجموعه جدید وابستگی تابعی و مجموعه داده‌ها گراف تناقض رسم‌می‌شود و سپس هر تراکنش به‌صورت تصادفی انتخاب و جهت

تصحیح مورد بررسی قرارمی‌گیرد. هم‌چنین بار دیگر رکوردهای متناقض بدون اصلاح وابستگی تابعی اصلاح‌می‌گردد. تصحیحات یافته‌شده برای مجموعه وابستگی‌های تابعی و مجموعه داده‌ها برای مشخص‌نمودن تصحیح بهینه به کاربر پیشنهادمی‌گردد. این روش به دلیل تعامل با کاربر در حجم داده‌ای بالا مناسب نیست.

در [۲۲] چارچوبی برای تصحیح داده‌هایی که به‌طور مداوم در حال تغییر می‌باشند، ارائه‌شده است. این چارچوب مبتنی بر وابستگی تابعی بوده، و در هنگام یافتن تناقض علاوه بر جستجو برای یافتن تصحیح داده‌ها، به جستجوی تصحیح وابستگی تابعی نقض‌شده می‌پردازد. سپس به‌وسیله یک طبقه‌بند مشخص‌می‌کند که تصحیح باید بر روی وابستگی‌تابعی، داده و یا به‌صورت ترکیبی بر روی هر دو صورت‌گیرد. نویسنده این مقاله معتقد است که روش‌های تصحیح داده اغلب ذهنی است، لذا زمانی که طبقه‌بند تشخیص می‌دهد که داده‌ها باید اصلاح‌گردد، داده‌های متناقض را به همراه مقادیری جهت تصحیح به کاربر نمایش‌می‌دهد و از او می‌خواهد تا آن‌ها را اصلاح‌نماید. پس از انتخاب کاربر، داده اصلاح شده جهت یادگیری به طبقه‌بند داده‌می‌شود.

در این روش از پارامترهای آماری نیز برای یافتن تصحیح بهینه استفاده‌شده‌است. مرحله اول یادگیری طبقه‌بند است. در این مرحله طبقه‌بند بر اساس تصحیحی که توسط کاربر انتخاب‌شده‌است، تحت آموزش قرارمی‌گیرد. در مرحله بعد تناقض بعدی شناسایی‌می‌شود و طبقه‌بند بر اساس داده‌های آماری و آموزشی که صورت‌گرفته، نوع تصحیح (تصحیح داده، تصحیح وابستگی تابعی و یا تصحیح ترکیبی) را پیش‌بینی می‌نماید. پس از تعیین نوع تصحیح وارد مرحله سوم می‌گردد. در این مرحله به جستجوی تمام مقادیر معتبر برای رفع ناسازگاری‌ها می‌پردازد. پس از یافتن تمام مقادیر معتبر جهت جایگزینی، این مقادیر جهت انتخاب تصحیح بهینه به کاربر نمایش‌داده‌می‌شود. پس از انتخاب کاربر این تصحیح منتخب بر روی مجموعه داده‌ها و یا وابستگی تابعی اعمال می‌گردد و هم‌چنین این انتخاب مجدداً جهت آموزش وارد مرحله اول می‌شود. این الگوریتم به دلیل تعامل با کاربر در هر دور از چرخه در حجم بالای داده‌ای مناسب نیست.

در [۲۳] راهکاری برای تصحیح ناسازگاری‌ها با نمونه‌برداری از تصحیحات مبتنی بر وابستگی تابعی شرطی ارائه‌شده‌است. این راهکار به ازای هر ناسازگاری در مجموعه داده‌ها تنها یک تصحیح ارائه‌نمی‌کند، بلکه به ازای هر ناسازگاری چندین تصحیح از فضای وابستگی تابعی شرطی ارائه‌می‌کند. سپس با استفاده از الگوریتمی که ارائه‌شده، به‌طور تصادفی یک تصحیح کارآمد را از فضای تصحیحات انتخاب‌می‌کند. هم‌چنین این روش به‌گونه‌ای طراحی‌شده، که علاوه بر وابستگی تابعی و وابستگی تابعی شرطی، می‌تواند محدودیت‌هایی را که از سوی کاربر تعیین‌می‌گردد، در تصحیحات به‌کارگیرد. محدودیت‌های کاربر می‌تواند شامل فیلدهایی باشد که کاربر درنظردارد در طول تصحیح بدون تغییر باقی‌بمانند. این روش تنها رکوردهای ناسازگار را تصحیح‌می‌نماید و برای رفع ناسازگاری هیچ رکوردی را حذف نمی‌کند.

ظهور پایگاه‌های دانش[۱۷] موجب ارائه فرصت‌های بیشتر برای دستیابی به داده‌های صحیح در مقیاس بزرگ‌تر شده است. در [۲۴] روشی مبتنی بر پایگاه دانش و سیستم‌های هوشمندسازی محاسبات[۱۸] ارائه‌شده‌است. در این شیوه جدول حاوی خطا، پایگاه دانش و *crowd* به‌عنوان ورودی دریافت‌می‌شوند. در ابتدا جدول الگوها برای نگاشت جدول داده‌ها به پایگاه دانش استخراج‌می‌شود. سپس برای هم‌تراز کردن با پایگاه دانش جدول معانی را تفسیرمی‌کند. پس از آن با جدول الگوها داده‌ها را به‌عنوان صحیح و ناصحیح به‌وسیله جایگذاری پایگاه دانش مشخص‌می‌کند. برای داده‌های ناصحیح k بالاترین نگاشت را از پایگاه دانش به‌عنوان مقدار صحیح جهت تصحیح معرفی می‌کند. در صورتی‌که اطلاعات لازم برای انتخاب k تصحیح در پایگاه دانش وجودنداشته‌باشد، این روش از فرد خبره در حوزه پایگاه دانش استفاده‌می‌کند.

در [۲۵] یک مسئله جدید به نام اثر متقابل بین تصحیح داده‌ها و تطابق آن‌ها بیان‌شده‌است. در این راهکار بیان‌شده که تصحیح داده‌ها به‌طور مؤثر به تطابق داده‌ها کمک‌می‌کند. در این شیوه چارچوبی متشکل از عملیات تطابق و تصحیح برای دستیابی به مجموعه داده صحیح مبتنی بر محدودیت‌ها، قوانین تطابق و *master data* ارائه‌شده است. در این راهکار از وابستگی تابعی شرطی و وابستگی تطابق به‌عنوان محدودیت‌ها برای کشف ناسازگاری‌ها استفاده‌شده‌است. برای اطمینان از دقت رفع خطاها سه پارامتر، میزان اطمینان کاربر به داده‌ها، اندازه‌گیری آنتروپی و داده *master* به‌کار گرفته‌شده است. بدین ترتیب سه الگوریتم برای رفع خطاها در این چارچوب ارائه‌شده‌است. الگوریتم اول مبتنی بر آنالیز مقدار اطمینان و داده *master* به رفع قطعی خطاها می‌پردازد، که دقت بالاتری نسبت به دو الگوریتم دیگر دارد. زمانی‌که اطمینان مقدار پایینی داشته‌باشد یا ناموجود باشد، الگوریتم برای رفع منطقی خطاها از آنتروپی استفاده‌می‌کند. برای باقی‌مانده خطاها یک روش تجربی ارائه‌شده‌است.

در [۲۶] یک سیستم یکپارچگی برای شناسایی مقادیر ناسازگار داده‌ها همراه با مقدار پیشنهادی جهت تصحیح آن، مبتنی بر وابستگی تابعی بیان‌شده‌است. در این روش برای رفع ناسازگاری‌ها تصحیحاتی بر روی داده یا وابستگی‌های تابعی انجام‌می‌گیرد. در این راهکار برای تصحیح وابستگی‌های تابعی ۳ شیوه مورد استفاده قرارگرفته‌است: (۱) اضافه نمودن ویژگی به وابستگی تابعی، (۲) تبدیل وابستگی تابعی به وابستگی تابعی شرطی، (۳) شناسایی افزونگی در وابستگی تابعی. این سیستم از ۴ جز تشکیل‌شده‌است: (۱) ماژول تشخیص‌دهنده ناسازگاری، (۲) ماژول تصحیح‌کننده داده، (۳) ماژول تصحیح‌کننده وابستگی تابعی، (۴) موتور تصحیح. در ابتدا داده و وابستگی‌های تابعی برای تشخیص داده ناسازگار به بخش تشخیص‌دهنده ناسازگاری هدایت می‌شوند، سپس هر رکورد که با هر وابستگی تابعی در تناقض باشد، شناسایی می‌شود. رکورد ناسازگار وارد ماژول تصحیح‌کننده داده و وابستگی تابعی نقض‌شده وارد تصحیح‌کننده وابستگی‌تابعی می‌شود. ماژول تصحیح‌کننده داده، الگوهای رکوردها را با یکدیگر مورد مقایسه

قرارمی‌دهد و مجموعه‌ای از تصحیحات را به موتور تصحیح می‌فرستد. در ماژول تصحیح‌کننده وابستگی تابعی نیز با یکی از ۳ روش بیان‌شده تصحیحاتی ارائه و به موتور تصحیح فرستاده‌می‌شود. در موتور تصحیح، تصحیحی که کم‌ترین هزینه را داشته باشد، انتخاب‌می‌گردد. هم‌چنین در این سیستم امکان این‌که بهترین تصحیح توسط کاربر انتخاب گردد، فراهم‌شده است.

در [۲۷] راهکاری برای تصحیح داده‌های ناصحیح مبتنی بر محدودیت‌ها ارائه‌شده‌است. در این روش تصحیح داده‌ها با تغییرات کوچک شامل درج یا حذف ویژگی بر روی محدودیت‌ها همراه است. این مقاله تصحیحی کمینه برای داده‌ها ارائه‌می‌کند، به‌نحوی‌که حداقل یک نوع محدودیت که دارای تنوع بیشتر از θ نسبت به محدودیت‌های دیگر داده‌شده باشد را برآورده نماید. این روش قابلیت تصحیح ویژگی‌های عددی را دارا است، که این امکان در روش‌های مشابه مبتنی بر محدودیت وجود نداشته است.

در [۲۸] یک مدل مبتنی بر هزینه که داده و محدودیت در آن به‌طور تساوی با یکدیگر مورد مقایسه قرارمی‌گیرند، ارائه‌شده است. در این راهکار از وابستگی تابعی به‌عنوان محدودیت استفاده‌شده‌است. در این مدل دو الگوریتم جداگانه برای تعمیر داده و محدودیت بیان‌شده‌است. در الگوریتم تعمیر داده به ازای هر داده ناسازگار، به جستجوی تغییراتی برای داده می‌پردازد، به‌نحوی‌که با محدودیت‌ها سازگار و هزینه تعمیرات را حداقل گرداند. برای تصحیح دو رکورد ناسازگار با یک وابستگی تابعی، مقدار ویژگی سمت راست وابستگی تابعی در دو رکورد را یکسان قرارمی‌دهد یا این‌که مقدار ویژگی سمت چپ آن‌ها را متفاوت می‌گذارد. در الگوریتم تصحیح وابستگی تابعی برای رفع ناسازگاری‌ها، دو رویکرد ارائه‌شده‌است. در رویکرد اول به جستجوی ویژگی‌ای می‌پردازد که با اضافه‌نمودن آن به وابستگی تابعی، ناسازگاری رفع‌می‌شود. برای انتخاب ویژگی از مفهوم واریانس اطلاعات بین ویژگی جدید و وابستگی تابعی ناسازگار استفاده‌می‌کند، به‌صورتی‌که ویژگی بدون کاهش افزونگی باعث تغییر ناسازگاری گردد. در رویکرد دوم مجموعه‌ای از مقادیر ویژگی‌ها از رکوردهایی که فقط آن‌ها وابستگی تابعی را برآورده‌می‌کنند، شناسایی‌می‌گردد. این مقادیر، زیرمجموعه‌ای از رکوردها را تعیین‌می‌کند، که با وابستگی تابعی سازگار است. سپس مدلی مبتنی بر هزینه برای انتخاب بهترین تصحیح اجرا می‌شود. هدف اصلی در این مدل انتخاب تعمیری با کمترین میزان تغییرات است.

راهکارهای تصحیح معرفی‌شده از چندین دیدگاه با یکدیگر تمایز دارند که از جمله می‌توان به خودکار یا نیمه خودکار بودن، انواع داده‌ای که قادر به تصحیح آن هستند، تفکیک‌پذیری فاز تشخیص و تصحیح و تعداد خطایی که می‌توانند دریک رکورد شناسایی کنند، اشاره‌نمود. هدف اصلی این مقاله معرفی راهکاری جهت تصحیح خودکار داده‌ها برای تمام انواع داده‌ای است. هم‌چنین روش پیشنهادی توانایی تشخیص چندین خطا را در یک رکورد دارد. در بخش پنجم جزئیات راهکار پیشنهادی به‌طور کامل شرح‌داده‌می‌شوند.

۳- پیش‌زمینه

در این بخش تعریفی از انواع داده‌ای و سیستم یادگیری مرکب ارائه‌می‌گردد.

تعریف ۱: ویژگی‌های یک مجموعه داده در فرمت، دامنه و نوع با یکدیگر متفاوت هستند. ویژگی‌های یک منبع داده ممکن است همگی از یک نوع داده‌ای باشند و یا امکان دارد از انواع داده‌ای مختلف در یک مجموعه داده وجود داشته باشد [۲۹]. انواع داده‌ای شامل نوع‌داده‌ای اسمی، ترتیبی، عددی و باینری است که در ادامه به شرح هر یک پرداخته‌می‌شود.

- ویژگی اسمی: نوع داده‌ای است که مربوط به اسامی است. مقدار یک ویژگی اسمی سمبل‌ها یا نام اشیاء است. هر مقدار نشان‌دهنده نوعی از دسته‌ها، کدها و حالت‌ها است، ازاین‌رو به ویژگی‌های اسمی، دسته‌ای[۱۹] نیز گفته‌می‌شود. این مقادیر هیچ ترتیب معناداری ندارند.
- ویژگی باینری: ویژگی باینری، نوعی از ویژگی اسمی است که تنها دو مقدار ۰ و ۱ دارد، که ۰ نشان‌دهنده این است که آن ویژگی حضور ندارد و ۱ به معنای این است که آن ویژگی وجود دارد.
- ویژگی ترتیبی: نوعی از ویژگی‌ها است که مقادیر آن دارای ترتیب معناداری هستند و یا این‌که بین مقادیر آن نوعی رتبه‌بندی وجود دارد.
- ویژگی عددی: ویژگی‌های عددی کمی هستند؛ در واقع دارای یک مقدار قابل اندازه‌گیری است که صحیح یا حقیقی است.

تعریف ۲: مدل‌های یادگیری مرکب، الگوریتم‌های یادگیری‌ای هستند که از مجموعه‌ای از طبقه‌بندها ساخته‌می‌شوند. سیستم‌های یادگیری مرکب به‌طور عمده شامل دو بخش هستند. بخش اول نحوه تقسیم‌بندی مجموعه داده آموزش، انتخاب نوع طبقه بندها، تعداد طبقه‌بندها و ویژگی‌های مناسب برای هر طبقه‌بند است. راهکارهای متنوعی برای تقسیم‌بندی مجموعه داده آموزش وجود دارد که از جمله می‌توان به cross validation، boostrapping و تقسیم‌بندی بر اساس میزان شباهت اشاره‌نمود. بخش دوم شامل نحوه ترکیب خروجی طبقه‌بندها به‌منظور حصول بهترین نتیجه برای طبقه‌بندی الگوهاست . روش‌های گوناگونی برای ترکیب طبقه‌بندها وجود دارد که از پرکاربردترین شیوه‌ها می‌توان به میانگین، رأی اکثریت، boosting، borda counts، ECOC و mixture of expert اشاره‌نمود [۳۵].

۴- راهکار پیشنهادی

در این بخش یک راهکار خودکار برای تشخیص و تصحیح خطا ارائه‌شده‌است. در فاز تشخیص، ابتدا رکوردهای مشکوک به خطا و سپس ویژگی‌های مشکوک به خطا در هر رکورد با استفاده از وابستگی تابعی تشخیص‌داده‌می‌شوند. در فاز تصحیح خطاهای شناسایی‌شده به‌وسیله یادگیری مرکب که شامل الگوریتم‌های درخت تصمیم، طبقه‌بند بیز و

شبکه عصبی MLP است، تصحیح می‌گردد. این راهکار قادراست که خطا را در تمام انواع داده‌ای تشخیص و تصحیح نماید. هم‌چنین به دلیل تفکیک‌پذیری فاز تشخیص و تصحیح، کاربر امکان تعامل با الگوریتم را دارد، تا در صورت دلخواه در خطاها تغییر ایجاد نماید. البته در صورت عدم تعامل کاربر هیچ مشکلی برای راهکار تصحیح پیش نمی‌آید و راهکار بدون اشکال به تصحیح می‌پردازد. علاوه بر این، راهکار پیشنهادی توانایی تشخیص چندین خطا را در یک رکورد دارا است. در بخش‌های ۴-۱ و ۴-۲ به ترتیب به شرح فاز تشخیص و تصحیح پرداخته‌می‌شود.

۴-۱- مرحله اول. تشخیص خطا

در این مرحله پیش از شروع الگوریتم، ویژگی‌های عددی باید گسسته‌سازی[۲۰] شوند و به ویژگی‌های ترتیبی تبدیل گردند. برای این کار می‌توان از یکی از الگوریتم‌های موجود استفاده‌نمود [۲۹]. پس از این مرحله وابستگی‌های تابعی منبع داده باید استخراج گردد. قابل‌ذکر است که الگوریتم‌های استخراج وابستگی در پیچیدگی زمانی با یکدیگر متفاوت هستند، اما کیفیت وابستگی‌های تابعی استخراج‌شده در تمام الگوریتم‌ها یکسان است [۳۰]. هم‌چنین تصحیح داده‌ها جز فرایند پیش‌پردازش است و لذا پیچیدگی زمانی، پارامتر فاقد اهمیت است. بنابراین در راهکار پیشنهادی از الگوریتم TANE [۳۱] که پایه روش‌های بعدی بوده است، استفاده‌شده است. با استفاده از وابستگی‌های تابعی استخراج‌شده، باید رکوردهای مشکوک به خطا شناسایی‌گردد. رکوردهایی که بیش از یک حد آستانه وابستگی‌های تابعی را نقض نمایند، مشکوک به خطا می‌باشند. این حد آستانه توسط کاربر نیز قابل تنظیم است، در این مقاله هر رکوردی که به ازای یک وابستگی تابعی با δ رکوردها ناسازگار باشد، مشکوک به خطا درنظرگرفته‌شده‌است. پس از شناسایی رکوردهای مشکوک به خطا، رکوردهای صحیح از این رکوردها تفکیک‌می‌گردد. این قسمت از مرحله تشخیص خطا بخش اول نامیده‌می‌شود و شبه کد آن در شکل ۱ آورده‌شده‌است.

Input
D: a set of records
δ: a threshold for number of inconsistent records
Output
D: a set of correct records
D′: a set of erroneous records
Procedure
1 **foreach** numeric attribute in D
2 $D \leftarrow$ discretized numeric attribute
3 end
4 $FD \leftarrow$ functional dependencies of D extracted with TANE algorithm
5 $D' \leftarrow \emptyset$
6 **foreach** *record* in *D*
7 **foreach** fd_i in *FD*
8 **if** *record* inconsist with more than δ *records*
9 $D' \leftarrow D' \cup record$
10 $D \leftarrow$ remove *record* from *D*
11 **end**
12 **end**
13 **end**
14 **return**(D, D')

شکل ۱: شبه کد بخش اول مرحله تشخیص خطا راهکار پیشنهادی

حال به‌وسیله رکوردهای صحیح و وابستگی‌های تابعی باید محل وقوع خطا در هر رکورد مشخص گردد. این قسمت از مرحله تشخیص خطا بخش دوم نامیده‌می‌شود و شبه کد آن در شکل ۲ آورده‌شده‌است. ابتدا باید تمام مقادیری که حداقل دومرتبه، به ازای یک وابستگی تابعی در مجموعه داده صحیح تکرار شده است، استخراج‌شود. خطوط ۱-۶ در شکل ۲ این قسمت از روش پیشنهادی را بیان‌می‌کند. به‌طورمثال فرض‌کنید در یک پایگاه داده مربوط به اطلاعات کارمندان، وابستگی تابعی $age, educate \rightarrow income$ استخراج‌شده است، که بدین مفهوم است که کسانی که سن و مدرک تحصیلی یکسان دارند، باید درآمد یکسان داشته‌باشند. به ازای این وابستگی مقادیر "$32, master, 100\$$" و "$26, Bachelor, 70\$$" در مجموعه داده حداقل دومرتبه تکرار شده است. چنین مقادیری باید به ازای تمام وابستگی‌های تابعی استخراج‌گردد. این مقادیر به ازای هر وابستگی تابعی در ماتریس φ_i ذخیره‌می‌شوند. چنان‌چه m وابستگی تابعی $\{fd_1, fd_2, \cdots, fd_m\}$ از منبع داده استخراج‌شده‌باشد، مجموعه این مقادیر به‌صورت $\varphi_1, \varphi_2, \cdots, \varphi_m$ است. پس از استخراج این مقادیر، به کمک آن‌ها محل وقوع خطا در رکوردهای خطادار شناسایی‌می‌شود.

Input
FD: a set of functional dependency $\{fd_1, fd_2, \cdots, fd_m\}$
β: number of user thinks that each record had error
D: a set of correct records
D': a set of erroneous records
Output
Out: set of erroneous attribute in each record in D'

Procedure
1 **foreach** fd_i in FD
2 **foreach** *record* in D
3 $\gamma \leftarrow$ value of attribute in fd_i from *record*
4 **end**
5 $\varphi_i \leftarrow$ the value is repeated at least twice in γ
6 **end**
7 **foreach** *record* in D'
8 *left* $\leftarrow 0$
9 *right* $\leftarrow 0$
10 **foreach** fd_i in FD
11 *attl* $\leftarrow$ attribute in the left side of fd_i
12 **if** value of *attl* attribute in *record* exist in φ_i
13 *attr* $\leftarrow$ attribute in the right side of fd_i
14 **if** value of *attr* attribute in *record* inconsistence with value of *attr* attribute in founded φ_i
15 *right* $\leftarrow$ *right* + 1
16 *cacher* $\leftarrow$ *attr* attribute
17 **end**
18 **else**
19 *left* $\leftarrow$ *left* + 1
20 *cachel* $\leftarrow$ *attl* attribute
21 **end**
22 **if** *right* $\geq {}^{m}/_{\frac{k}{2}}$
23 error $\leftarrow$ find repeated number of each attribute in *cacher* & select β the most frequent
24 **else if** *left* $\geq {}^{m}/_{\frac{k}{2}}$
25 error $\leftarrow$ find repeated number of each attribute in *cachel* & select β the most frequent
26 **end**
27 **return** error
28 **end**

شکل ۲: شبه کد بخش دوم تشخیص خطا در الگوریتم تصحیح مبتنی بر وابستگی تابعی و یادگیری مرکب

ابتدا کاربر لازم است تعیین نماید، که در یک رکورد حداکثر چند ویژگی ناصحیح ممکن است موجود باشد. این مقدار β نامیده‌می‌شود. به ازای هر رکورد r باید تمام وابستگی‌های تابعی مورد بررسی قرار گیرند. فرض‌می‌شود fd_i در حال بررسی است. ابتدا ویژگی‌های سمت چپ fd_i مورد بررسی قرارمی‌گیرد. به ازای ویژگی‌های موجود در fd_i، به جستجوی مقدار این ویژگی‌ها در رکورد r، در مقادیر ذخیره‌شده در ماتریس φ_i، پرداخته‌می‌شود. چنانچه ترکیب مقادیر موجود در رکورد r در φ_i یافت نگردید، ویژگی‌های سمت چپ مشکوک به خطا هستند. لذا این ویژگی‌ها ذخیره شده و به پارامتر *left* که نشان‌می‌دهد که ویژگی‌های سمت چپ مشکوک به خطا هستند، یک واحد اضافه‌می‌گردد. اما چنان‌چه این ترکیب مقادیر در رکورد r در φ_i موجود بود، ویژگی سمت راست مورد بررسی قرارمی‌گیرد. چنان‌چه مقدار ویژگی سمت راست در رکورد r، با مقدار آن ویژگی در ترکیبی که در φ_i یافت شد، برابر نباشد، ویژگی سمت راست مشکوک به خطا است.

لذا این ویژگی ذخیره و به پارامتر *right* که بیانگر تعداد دفعاتی است که ویژگی‌های سمت راست مشکوک به خطا هستند، یک واحد اضافه‌می‌شود. خطوط ۱۰-۲۱ در شکل ۲ این بخش از راهکار پیشنهادی را نشان‌می‌دهد. این روند به ازای تمام وابستگی‌های تابعی انجام‌می‌گیرد. پس از اتمام این مرحله، مقدار *left* و *right* مورد بررسی قرارمی‌گیرد. فرض‌می‌شود k تعداد ویژگی‌های منبع داده‌ای باشد، در صورتی‌که مقدار هرکدام از $\frac{m}{\frac{k}{2}}$ وابستگی‌های تابعی بیشتر باشد، ویژگی آن سمت از وابستگی تابعی مشکوک به خطا هستند. پس از این‌که مشخص‌گردید، ویژگی‌های کدام سمت مشکوک به خطا هستند، به بررسی تعداد دفعاتی که هر ویژگی در آن سمت مشکوک به خطا معرفی‌شده، پرداخته‌می‌شود. این تعداد بر تعداد دفعاتی که هر ویژگی در وابستگی تابعی تکرار شده، تقسیم‌می‌شود و β ویژگی که این مقدار را بیشینه نمایند، به‌عنوان خطای نهایی معرفی می‌شوند. این بخش از راهکار، در خطوط ۲۲-۲۷ بیان‌شده‌است.

۴-۲- مرحله دوم. تصحیح خطا

در این مرحله به ازای خطاهای یافته‌شده در هر رکورد، از یادگیری مرکب برای تصحیح خطا استفاده‌می‌شود. روش‌های متنوعی برای ساخت یک سیستم یادگیری مرکب وجود دارد، که یکی ازاین‌رویکردها استفاده از الگوریتم‌های متنوع یادگیری است. در راهکار پیشنهادی از سه الگوریتم طبقه‌بند بیز، درخت تصمیم و شبکه‌عصبی MLP استفاده‌شده است. هدف از انتخاب این سه الگوریتم قدرتمندی آن نسبت به الگوریتم‌های طبقه‌بندی دیگر مانند نزدیک‌ترین همسایگی است. هم‌چنین استفاده از درخت تصمیم، طبقه بند بیز و شبکه‌عصبی MLP به‌تنهایی در راهکارهای پیشین کارایی نسبتاً خوبی در پیش‌بینی مقادیر صحیح داشته است[۱، ۴، ۶، ۷، ۱۷-۱۹]. سیستم‌های یادگیری مرکب دارای استراتژی‌های ترکیب متنوعی است، که در راهکار پیشنهادی از رأی اکثریت برای استراتژی ترکیب استفاده‌شده‌است.

Input
D: a set of correct records with k attributes $\{t_1, t_2, \cdots, t_k\}$
D': a set of erroneous records with k attributes $\{t_1, t_2, \cdots, t_k\}$
ER: incorrect attributes in erroneous records
Output
Out: set of correct values for incorrect attributes in each record in D'

Procedure
1 **foreach** t_i in D
2 $tree_i \leftarrow$ train decision tree with D data set & t_i as target class
3 $bayes\_model_i \leftarrow$ train Bayes with D data set & t_i as target class
4 $MLP\_model_i \leftarrow$ train MLP with D data set & t_i as target class
5 **end**
6 **foreach** *record* in D'
7 **foreach** incorrect attribute e in ER
8 $tree\_estimate\_class \leftarrow$ select $tree_e$ & predict value for attribute e with $tree_e$
9 $bayes\_estimate\_class \leftarrow$ select $bayes\_model_e$ & predict value for attribute e with $bayes\_model_e$
10 $MLP\_estimate\_class \leftarrow$ select MLP_model_e & predict value for attribute e with MLP_model_e
11 $correct\_value \leftarrow$ most repeated value between $tree\_estimate\_class$, $bayes\_estimate\_class$, $MLP\_estimate\_class$
12 **return** $correct\_value$
13 **end**
14 **end**

شکل ۳: شبه کد بخش تصحیح خطا از الگوریتم تصحیح مبتنی بر وابستگی تابعی و سیستم یادگیری مرکب

ابتدا درخت تصمیم مورد آموزش قرارمی‌گیرد. برای آموزش هر بار یک ویژگی به‌عنوان کلاس هدف قرارمی‌گیرد و درخت به‌وسیله رکوردهای صحیح آموزش ساخته‌می‌شود. برای ساخت درختان از روش CART استفاده‌شده است. سپس هر درخت ساخته‌شده برای دستیابی به درخت بهینه، هرس‌می‌گردد. پس از هرس‌نمودن درختان برای فاز پیش‌بینی ذخیره‌می‌شوند. در طبقه‌بند بیز نیز، به ازای هر ویژگی ناصحیح، آن ویژگی به‌عنوان کلاس هدف قرارمی‌گیرد و طبقه‌بند بیز ساخته‌می‌شود. در طبقه بند بیز از توزیع گوسی[۲۱] و احتمال پیشین[۲۲] استفاده‌می‌شود. برای آموزش شبکه‌عصبی MLP چنان‌چه کاربر درباره انتخاب پارامترهای شبکه عصبی ازجمله ضریب یادگیری، تعداد لایه‌های میانی و تعداد تکرار برای داده مورد بررسی دانشی داشته باشد، می‌تواند به ساخت یک شبکه عصبی به ازای هر ویژگی اکتفا نماید. اما در اغلب موارد کاربر چنین دانشی را در اختیار ندارد، لذا به ازای هر ویژگی چندین شبکه عصبی ساخته‌می‌شود که در آن‌ها پارامترهای اولیه شبکه عصبی متفاوت است. در این سیستم یادگیری مرکب به ازای هر ویژگی، ۵ شبکه عصبی با پارامترهای اولیه متفاوت استفاده‌شده‌است.

پس از آموزش طبقه‌بندها، می‌توان مقادیر صحیح را برای ویژگی‌های ناصحیح پیش‌بینی‌نمود. هر بار یک ویژگی از یک رکورد وارد سیستم یادگیری مرکب می‌گردد. سپس در درخت تصمیم، از میان درختان ساخته‌شده، درخت مربوط به ویژگی ورودی انتخاب و مقداری برحسب رکورد ورودی برای ویژگی ناصحیح به‌وسیله درخت پیش‌بینی‌می‌شود. در طبقه‌بند بیز و شبکه عصبی نیز به همین روش مدل آموزش‌داده‌شده برای ویژگی ناصحیح انتخاب و مقدار صحیح به‌وسیله مدل پیش‌بینی‌می‌گردد. پس از این‌که پیش‌بینی به‌وسیله تمام طبقه‌بندها صورت‌گرفت، به‌وسیله استراتژی یادگیری مرکب که رأی اکثریت است، مقداری که بیشترین تکرار را داشته به‌عنوان مقدار صحیح معرفی‌می‌شود. شبه کد این بخش از الگوریتم در شکل ۳ آمده است. در این شبه کد فرض شده که کاربر پارامترهای اولیه شبکه عصبی را می‌داند و لذا از یک شبکه عصبی استفاده‌شده‌است. در راهکار پیشنهادی به دلیل این‌که

کاربر اطلاعی از تعداد خطاها در یک رکورد ندارد، کاربر برای حداکثر تعداد ویژگی‌هایی که در یک رکورد ناصحیح است مقداری را تخمین‌می‌زند. لذا ممکن است یک ویژگی صحیح به‌عنوان غلط در نظر گرفته‌شده‌باشد، اما آزمایش‌ها نشان‌داده که چنان‌چه این رویداد پیش‌آید، سیستم یادگیری مرکب، مقداری بسیار نزدیک به مقدار اولیه برای آن ویژگی پیش‌بینی‌می‌نماید.

۵- آزمایش‌ها تجربی

در این بخش راهکار پیشنهادی بر روی دو مجموعه‌داده آزمایش‌می‌شود که مجموعه اول (adult) درباره درآمد افراد براساس اطلاعات سرشماری[۲۳] و مجموعه دوم (protein) درباره ویژگی‌های نوعی پروتئین[۲۴] است. مجموعه داده adult شامل ۴۸۸۴۲ رکورد و ۱۵ ویژگی است، که ۶ ویژگی عددی، ۱ ویژگی ترتیبی و ۸ ویژگی اسمی است. مجموعه داده protein شامل ۴۵۷۳۰ رکورد و ۹ ویژگی است که تمام ویژگی‌های آن عددی است. هدف از انتخاب این مجموعه داده‌ها این است که دو مجموعه داده‌ای هستند که از لحاظ انواع داده‌ای با یکدیگر کاملاً متفاوت می‌باشند، مجموعه adult شامل تمام انواع داده‌ای است درحالی‌که مجموعه protein فقط شامل یک نوع داده‌ای است. هم‌چنین این مجموعه داده‌ها از پراستفاده‌ترین منابع داده‌ای است [۱، ۳۴-۳۲]. در این مجموعه داده‌ها که فاقد خطا هستند، با استفاده از توزیع نرمال، ۵٪ ، ۱۰٪ ، ۱۵٪ و ۲۰٪ خطا ایجادشده، و نحوه عملکرد با پنج معیاری که در فرمول ۱ تا ۵ آورده‌شده مورد ارزیابی قرارگرفته است.

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

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

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

$$recall = \frac{TP}{TP + FN} \quad (۴)$$

$$precision = \frac{TP}{TP + FP} \quad (۵)$$

در ابتدای بخش ۴-۱ بیان شد که رکوردی که به ازای یک وابستگی تابعی با بیش از δ رکورد ناسازگار باشد، آن رکورد به‌عنوان رکورد مشکوک به خطا در نظر گرفته‌می‌شود. جدول ۱ مقادیر مختلف را برای این پارامتر بر روی دو مجموعه داده مورد بررسی قرار داده است. مقادیر true negative rate و error rate برای دو مجموعه داده به‌صورت میانگین بیان‌شده‌است. با توجه به این دو پارامتر بهترین مقدار برای پارامتر δ برای دو مجموعه داده مورد بررسی در مقدار $\frac{n}{3}$ رخ داده است.

راهکار پیشنهادی از دو فاز تشخیص و تصحیح تشکیل‌شده، که در ابتدا به بررسی کارایی بخش تشخیص پرداخته‌می شود. همان‌طور که در شکل ۴ نشان‌داده شده مقدار *true negative rate* در تمام نرخ‌های خطا برای هر دو مجموعه داده بالای ۹۲٪ است، که نشان‌دهنده توانایی بالای راهکار پیشنهادی در تشخیص خطا است. مقدار *recall* نیز در مجموعه داده *adult* به‌طور متوسط ۸۶٪ و در مجموعه داده *protein* به‌طور متوسط ۸۴٫۷٪ است. این معیار در حقیقت نشان‌دهنده فیلدهای صحیحی می‌باشند که به‌درستی به‌عنوان فیلد صحیح تشخیص‌داده‌شده‌است. به دلیل این‌که هیچ اطلاعاتی درباره تعداد خطا در هر رکورد وجود ندارد، یک مقداری به‌عنوان تعداد خطا حدس‌زده‌می‌شود و این باعث‌می‌گردد تعدادی فیلد صحیح به‌عنوان خطا تشخیص داده‌شوند و به همین علت مقدار معیار *recall* کمتر از *true negative rate* است. اما آزمایش‌ها نشان‌خواهد داد، که به اکثر این فیلدها در فاز تصحیح مقداری یکسان یا نزدیک به مقدار اولیه تخصیص‌داده‌می‌شود. هم‌چنین میزان *false alarm rate* در تمام نرخ‌های خطا در هر دو مجموعه داده کمتر از ۷٫۵٪ است و مقدار *error rate* به‌طور متوسط در مجموعه داده *adult 14%* و در مجموعه داده *protein 15%* است. هم‌چنین مقدار *precision* برای هر دو مجموعه داده بالای ۹۹٪ است. در شکل ۵ عملکرد بخش تشخیص خطای الگوریتم در مجموعه داده *adult* که شامل انواع داده‌ای متفاوت است، نشان‌داده‌شده‌است. همان‌طور که در شکل مشخص است بهترین عملکرد در تشخیص خطا در نوع داده‌ای اسمی و بدترین عملکرد در نوع داده‌ای ترتیبی رخ‌داده است. در بدترین عملکرد که در نوع داده‌ای ترتیبی رخ‌داده است، مقدار *true negative rate* به‌طور متوسط ۹۰٪ است و در نوع داده‌ای اسمی بالای ۹۷٪ است. هم‌چنین مقدار *error rate* در نوع داده‌ای ترتیبی به‌طور متوسط ۱۱٪ و در نوع داده‌ای اسمی ۶٪ است. مقدار *precision* نیز در تمام انواع داده‌ای و در تمام نرخ‌های خطا بالاتر از ۹۸٪ است.

جدول ۱: بررسی مقادیر مختلف برای پارامتر δ

Error rate	*true negative rate*	δ
٪۳۱	٪۹۸/۵	$\frac{n}{4}$
٪۹	٪۹۶/۷	$\frac{n}{3}$
٪۱۵	٪۹۲	$\frac{n}{2}$
٪۳۰	٪۸۱	$\frac{2n}{3}$

پس از تشخیص خطا، با استفاده از یادگیری مرکب به تصحیح خطا پرداخته‌می‌شود. همان‌طور که در بخش پیش اشاره شد، از درخت تصمیم، طبقه‌بند بیز و شبکه عصبی MLP برای ساخت سیستم یادگیری مرکب استفاده‌شده‌است. برای ساخت درخت تصمیم برای هر ویژگی از روش CART استفاده‌شده، سپس درختان هرس‌می‌گردد، جدول ۲ و ۳ به ترتیب بیان می‌کند که درختان ساخته‌شده از هر ویژگی، در مجموعه داده adult و protein در کدام سطح هرس‌شده‌اند. برخی از درختان نیز بدون هرس باقی‌مانده که با خط تیره در جدول نشان‌داده‌شده‌است. برای ساخت شبکه عصبی MLP به ازای هر ویژگی همان‌طور که در بخش

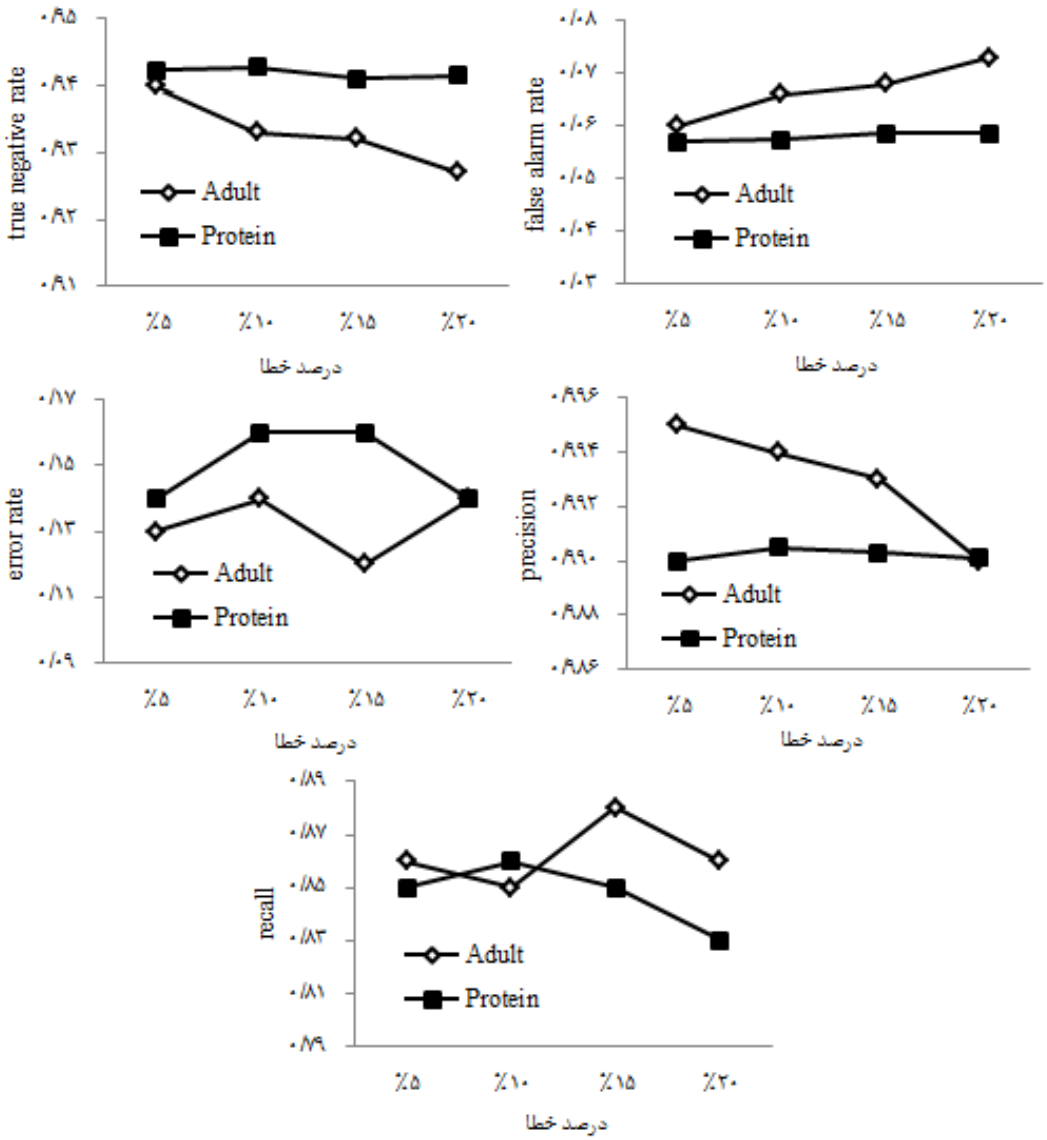
قبل بیان‌گردید، به دلیل عدم آگاهی از پارامترهای اولیه شبکه عصبی، از ۵ شبکه عصبی با پارامترهای اولیه متفاوت استفاده‌می‌گردد. بهترین مقادیر براساس آزمایش‌ها انجام‌شده برای هر ویژگی از مجموعه داده‌ها در جدول ۴ و ۵ آورده‌شده‌است. در واقع این مقادیر پس از آزمایش‌ها حاصل‌شده، و پیش از آزمایش‌ها به ۵ شبکه عصبی مقادیر مختلف داده‌شده‌بود، که بهترین نتایج از این مقادیر حاصل‌گردید.

جدول ۲: سطح هرس شده برای هر درخت ساخته شده به ازای هر ویژگی برای مجموعه داده adult

ویژگی	۱	۲	۳	۴	۵	۶	۷	۸
سطح	۱۱۰	۹۵	۱۰۰	-	-	۶۰	۱۰۵	۸۰
ویژگی	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	
سطح	۶۵	۷۵	-	-	۱۱۵	۲۸	۸۰	

جدول ۳: سطح هرس شده برای هر درخت ساخته شده به ازای هر ویژگی برای مجموعه داده protein

ویژگی	۱	۲	۳	۴	۵
سطح	۱۴۵	۱۵۶	۱۰۰	۱۴۱	۹۵
ویژگی	۶	۷	۸	۹	
سطح	۱۶۴	۱۵۳	۱۴۶	۸۶	



شکل ۴: نحوه عملکرد بخش تشخیص خطا در الگوریتم پیشنهادی

در شکل ۶، ۷ و ۸ نحوه عملکرد بخش تصحیح در طبقه‌بندهای مختلف نشان‌داده‌شده‌است. از میان این سه طبقه‌بند، در مجموعه داده adult، شبکه عصبی MLP و در مجموعه داده protein درخت تصمیم بهترین عملکرد را داشته‌اند. میزان true negative rate به ترتیب در مجموعه داده adult، ۸۲/۷٪ و در مجموعه داده protein، ۹۱/۵٪ محاسبه‌شده است. میزان false alarm rate و error rate نیز در این طبقه‌بندها در تمام نرخ‌های خطا در مجموعه داده adult، کمتر از ۲۰٪ و ۱۱٪ و مجموعه protein کمتر از ۱۰٪ و ۱۲٪ است. هم‌چنین مقدار precision و recall نیز در تمام نرخ‌های خطا در تمام طبقه‌بندها، در مجموعه داده adult بالاتر از ۹۱٪ و ۹۲٪ و در مجموعه protein بالاتر از ۸۸٪ و ۹۱٪ است. در شکل ۹ نیز عملکرد ترکیب این طبقه‌بندها در سیستم یادگیری مرکب که در واقع استراتژی تصحیح در راهکار پیشنهادی است، بیان شده‌است. در سیستم یادگیری از رأی اکثریت طبقه‌بندها به‌عنوان استراتژی ترکیب استفاده‌شده‌است. همان‌طور که در شکل ۹ مشخص است، میزان true negative rate، که دراقع بیان‌کننده توانایی سیستم یادگیری مرکب در تصحیح خطاها است، در تمام نرخ‌های خطا در مجموعه داده adult بالای ۸۶٪ و در مجموعه داده protein بالای ۹۳٪ است. هم‌چنین میزان false alarm rate در سیستم یادگیری مرکب به‌طور متوسط در مجموعه داده adult ۱۲٪ و در مجموعه داده protein نیز به‌طور متوسط ۶/۲٪ است. مقدار recall و precision نیز در این سیستم به‌طور میانگین در مجموعه داده adult ۹۳٪ و۹۳/۷٪و در مجموعه protein ۹۱/۵٪ و ۹۶/۸٪ است. مقدار error rate در مجموعه داده adultکمتر از ۱۰٪ و در مجموعه protein کمتر از ۹٪ است.

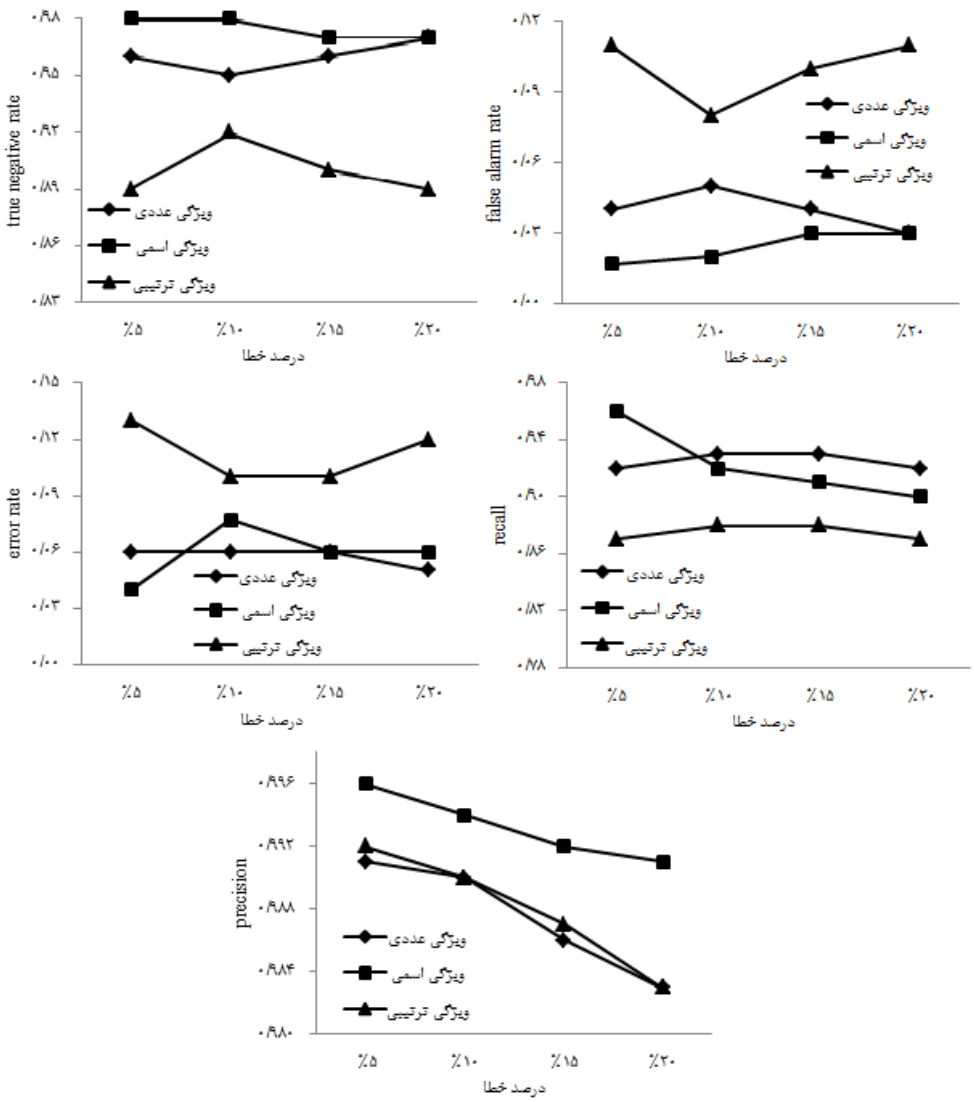
در شکل ۱۰ همانند شکل ۵ کارایی کل الگوریتم در مجموعه داده adult که شامل انواع داده‌ای مختلف است، نشان‌داده‌شده‌است. در این شکل همان ویژگی‌های موجود در شکل ۵ که کارایی بخش تشخیص را نشان‌داد، انتخاب‌شده است. عملکرد کل الگوریتم در نوع داده‌ای اسمی بهترین و در نوع داده‌ای عددی بدترین بوده است. در نوع داده‌ای اسمی میزان true negative rate به‌طور متوسط ۹۶٪ و recall ۹۷٪ است. در نوع داده‌ای عددی میزان true negative rate ۸۱٪ و recall ۹۷٪ است. همان‌طور که مشخص است میزان recall در کل الگوریتم نسبت به مرحله تشخیص خطا افزایش‌یافته است. علت این افزایش در این معیار این است که برخی از ویژگی‌هایی که در فاز تشخیص به‌عنوان خطا درنظرگرفته‌شدند، در مرحله تصحیح مقداری نزدیک یا مشابه مقدار اصلی به آن‌ها اختصاص داده شد و لذا باعث بهبود معیار recall شده است.

جدول ۴: پارامترهای ورودی شبکه عصبی MLP برای مجموعه داده Adult

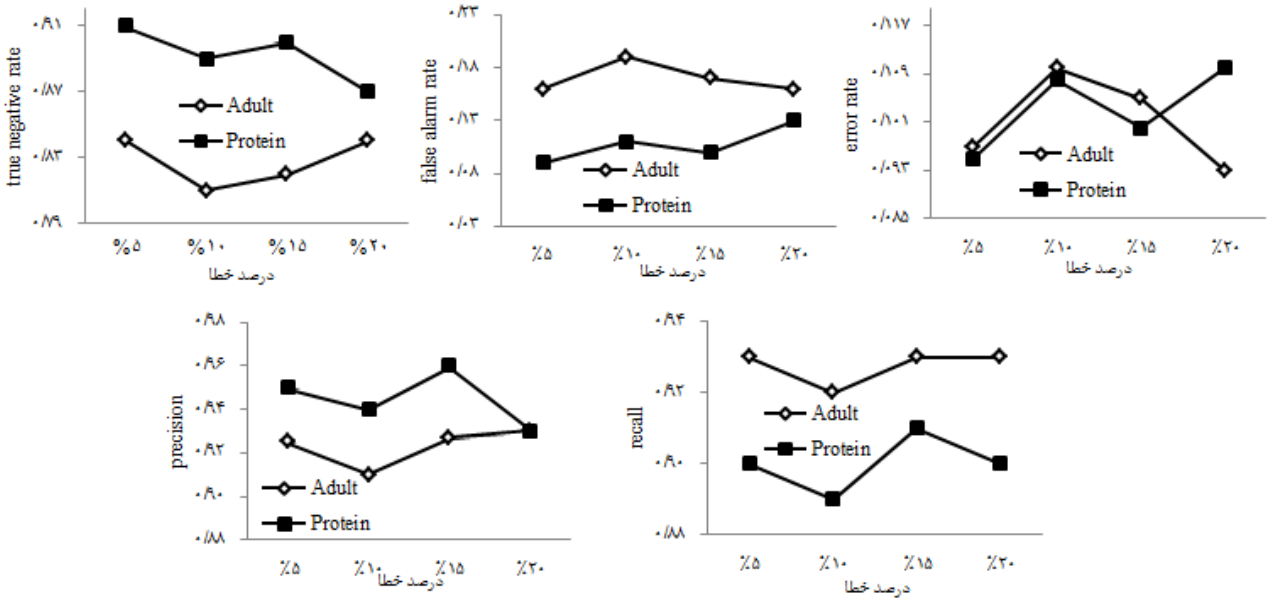
ویژگی	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵
ضریب یادگیری	۰/۱۴	۰/۱	۰/۱۲	۰/۱	۰/۱	۰/۱۲	۰/۱۴	۰/۱۴	۰/۱۲	۰/۱۴	۰/۱	۰/۱	۰/۱	۰/۱۱	۰/۱۲
Momentum	۰/۹	۰/۸	۰/۷	۰/۹	۰/۸	۰/۹	۰/۶	۰/۹	۰/۷	۰/۷	۰/۶	۰/۹	۰/۹	۰/۸	۰/۸
Epoch	۱۰	۱۰	۸	۵	۵	۱۰	۸	۵	۱۰	۵	۱۰	۱۰	۸	۸	۵
تعداد دسته‌ها	۵	۸	۵	۵	۵	۵	۵	۵	۵	۵	۵	۵	۵	۵	۵

جدول ۵: پارامترهای ورودی شبکه عصبی MLP برای مجموعه داده Protein

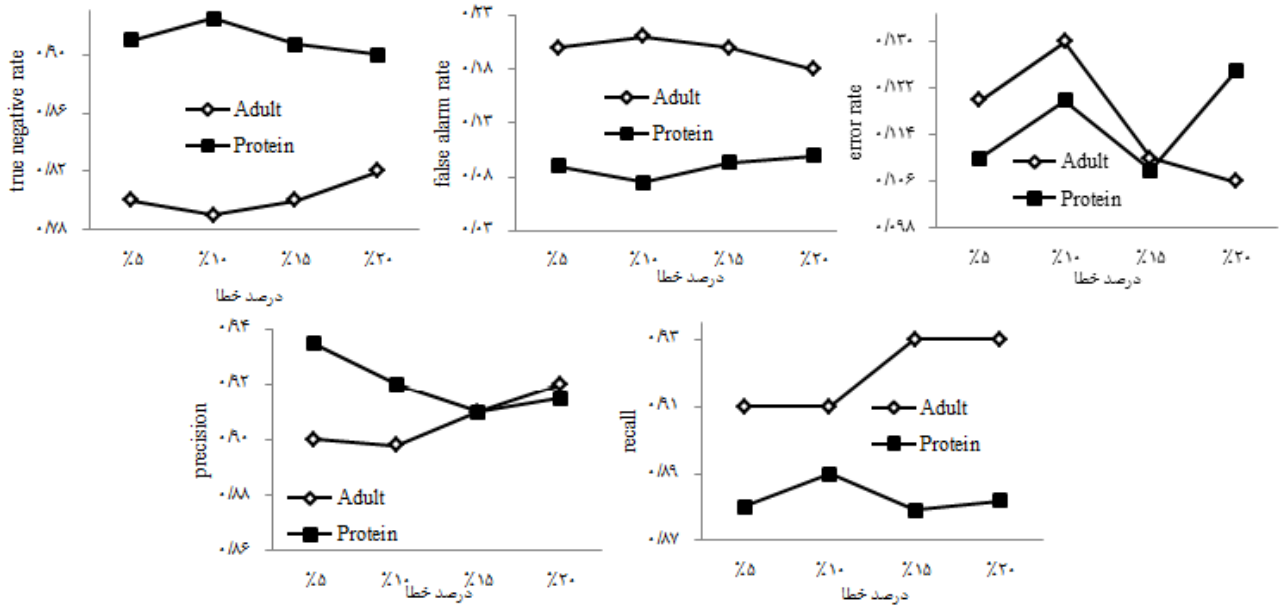
ویژگی	۱	۲	۳	۴	۵	۶	۷	۸	۹
ضریب یادگیری	۰/۲۳	۰/۲۱	۰/۲۴	۰/۱۲	۰/۲	۰/۲۳	۰/۲	۰/۱	۰/۲۱
Momentum	۰/۷	۰/۶	۰/۶	۰/۷	۰/۶	۰/۶	۰/۸	۰/۹	۰/۸
Epoch	۵	۸	۱۲	۲۰	۵۰	۱۰	۸	۲۰	۲۵
تعداد دسته‌ها	۱۰	۱۰	۵	۷	۵	۵	۶	۸	۹



شکل ۵: نحوه عملکرد الگوریتم پیشنهادی در بخش تشخیص خطا در انواع داده‌ای متفاوت در مجموعه داده adult



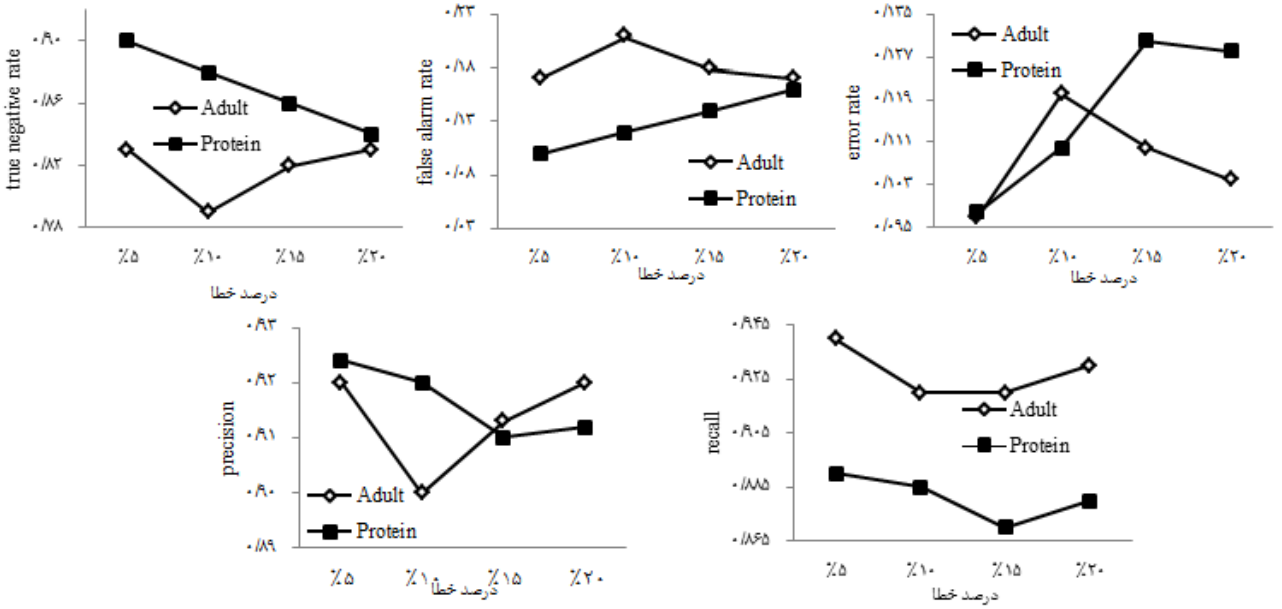
شکل ۶: نحوه عملکرد الگوریتم پیشنهادی در مرحله تصحیح خطا در شبکه عصبی MLP



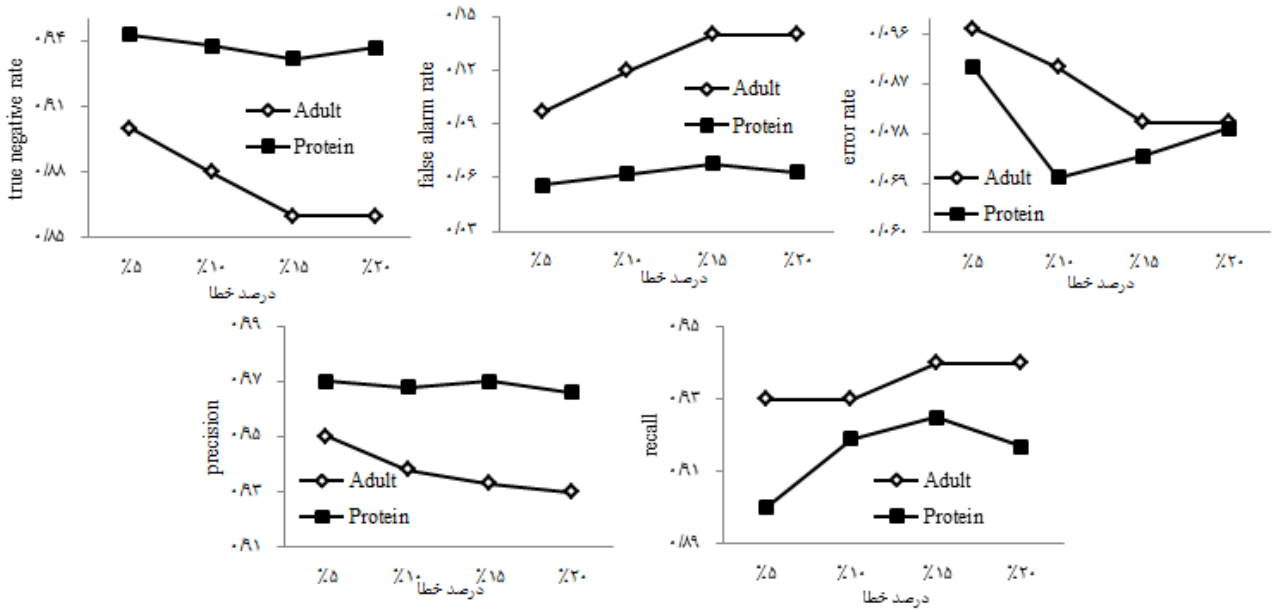
شکل ۷: نحوه عملکرد الگوریتم پیشنهادی در مرحله تصحیح خطا در درخت تصمیم

در شکل ۱۱ کارایی کل الگوریتم بر روی هر دو مجموعه داده مورد بررسی نشان‌داده‌شده‌است. میزان recall در کل الگوریتم به‌طور متوسط در مجموعه داده adult ۹۴٪ و در مجموعه داده protein به‌طور متوسط ۹۱/۵٪ است. مقدار error rate در تمام نرخ‌های خطا در مجموعه داده adult کمتر از ۱۰٪ و در مجموعه داده protein کمتر از ۹٪ است. مقدار true negative rate به‌طور متوسط در مجموعه داده adult ۸۳٪ و در مجموعه داده protein ۸۹/۸٪ است. میزان false alarm rate در تمام نرخ‌ها در مجموعه adult کمتر از ۱۷٪ و در مجموعه protein کمتر از ۱۱٪ است. مقدار precision به‌طور میانگین در مجموعه داده adult ۹۰/۹ ٪ و در مجموعه protein ۹۴/۴٪ است.

در شکل ۱۲ راهکار پیشنهادی با روش [۱۲] و[۲۸] که با استفاده از وابستگی تابعی به‌صورت خودکار به تصحیح داده‌ها پرداخته‌است، مورد مقایسه قرارگرفته‌است. هدف از انتخاب روش [۱۲] برای مقایسه این است که برخی از روش‌های پیشین که مبتنی بر وابستگی بودند [۲۱، ۲۲] برای مقایسه روش پیشنهادی خود ازاین‌روش استفاده نموده‌اند. براساس آزمایش‌ها میزان true negative rate در روش پیشنهادی به‌طور متوسط ۷٪ بالاتر از روش تصحیح با وابستگی تابعی [۱۲] است. هم‌چنین true negative rate در راهکار پیشنهادی به‌طور متوسط ۳٪ بالاتر از راهکار تصحیح [۲۸] است.



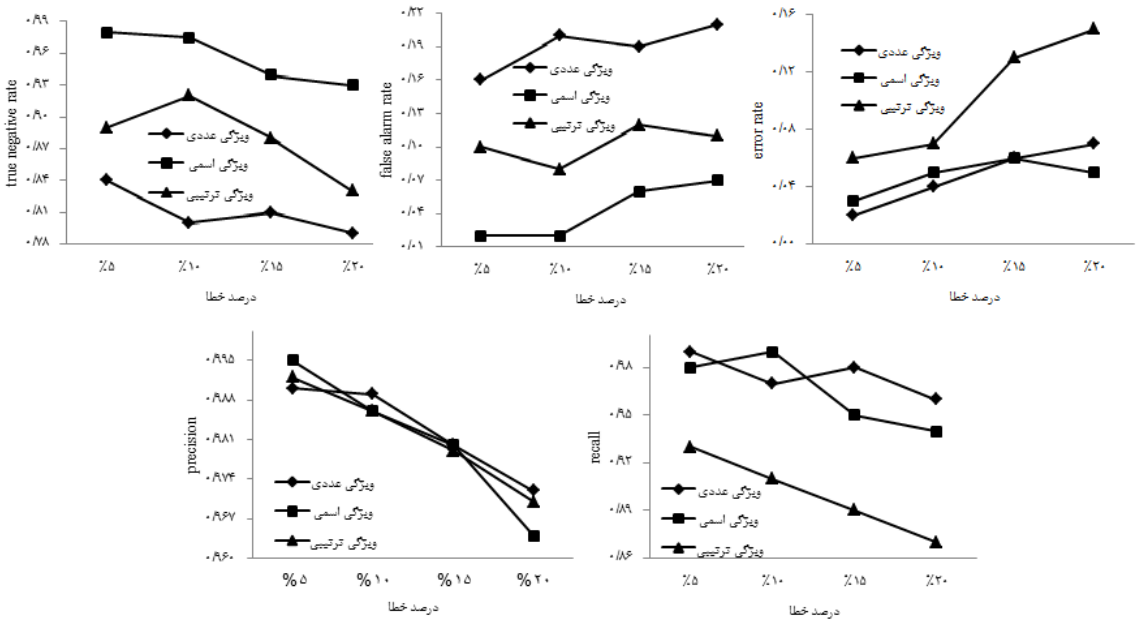
شکل ۸: نحوه عملکرد الگوریتم پیشنهادی در مرحله تصحیح خطا در طبقه بند بیز



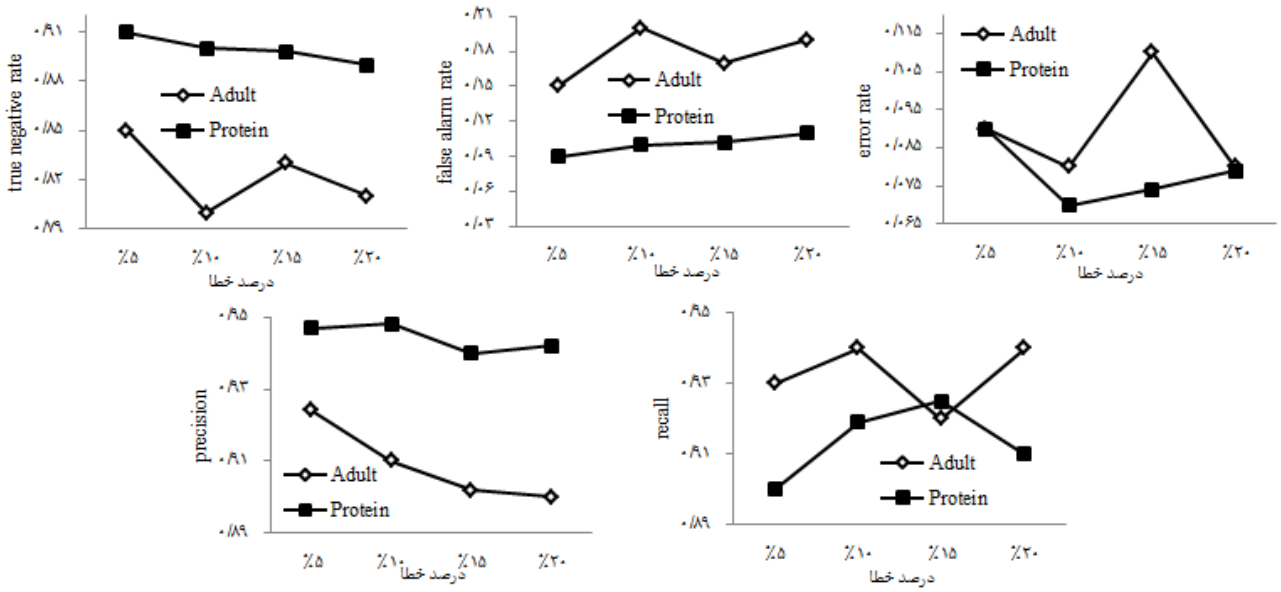
شکل ۹: نحوه عملکرد الگوریتم پیشنهادی در مرحله تصحیح خطا در سیستم یادگیری مرکب

۶- نتیجه‌گیری

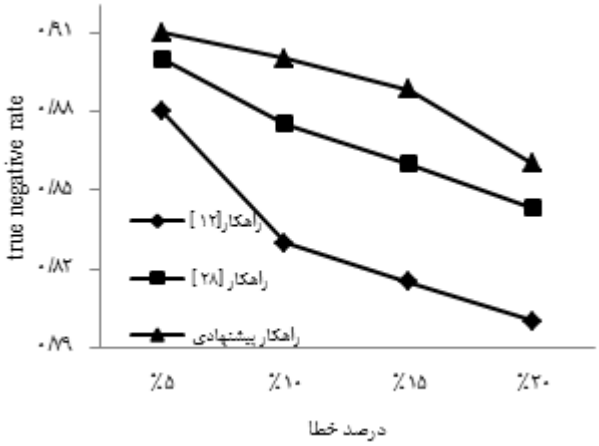
در این مقاله یک راهکار خودکار برای تشخیص و تصحیح فیلد حاوی خطا پیشنهاد شد. راهکار پیشنهادی توانایی تصحیح خطا بر روی مجموعه داده با انواع داده‌ای مختلف را دارد. هم‌چنین راهکار پیشنهادی توانایی تصحیح چندین خطا در یک رکورد را دارا است. در روش ترکیبی پیشنهادی، برای تشخیص خطا از وابستگی تابعی استفاده شد. در فاز تصحیح نیز فیلد خطادار تشخیص‌داده‌شده، به‌وسیله سیستم یادگیری مرکب اصلاح گردید. سیستم یادگیری مرکب مورد استفاده در راهکار پیشنهادی از سه طبقه‌بند بیز، درخت تصمیم و شبکه عصبی MLP تشکیل شده و استراتژی ترکیب آرای طبقه‌بندها رأی اکثریت است. بنابر آزمایش‌ها انجام‌شده کارایی روش پیشنهادی در تشخیص خطا، به‌طور متوسط ۹۳/۷٪ و در بخش تصحیح خطا به‌طور متوسط ۹۰/۶٪ است. هم‌چنین روش پیشنهادی با دو روش مشابه مبتنی بر وابستگی تابعی مورد مقایسه قرار گرفت که روش پیشنهادی ۵٪ توانایی بالاتری در تصحیح خطا نسبت به دو روش مورد مقایسه دارا است.



شکل ۱۰: نحوه عملکرد کل الگوریتم پیشنهادی بر روی انواع داده‌ای متفاوت در مجموعه داده adult



شکل ۱۱: نحوه عملکرد کل الگوریتم در نرخ‌های متفاوت خطا



شکل ۱۲: مقایسه راهکار پیشنهادی با راهکار [۱۲] و [۲۸] در نرخ‌های متفاوت خطا

مراجع

[1] 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, 2013.

[2] W. Fan, F. Geerts, X. Jia and A. Kementsietsidis, "Conditional functional dependencies for capturing data inconsistencies," *ACM Transactions on Database Systems (TODS)*, vol. 33, no. 2, pp. 1-48, 2008.

[3] P. H. WilliAMS, C. R. Margules, and D. W. Hilbert, "Data requirements and data sources for biodiversity priority area selection," *Journal of biosciences*, vol. 27, no. 4, pp. 327–338, 2002.

[4] G. Rahman and Z. Islam, "Decision Tree-based Missing Value Imputation Technique for Data Pre-processing," *Proceedings of the 9th Conference on Australasian Data Mining*, pp. 41-50, Ballarat, Australia, 2011.

[5] X. Chu and I. F. Ilyas, "Qualitative Data Cleaning," *The VLDB Journal*, vol. 9, no. 13, pp. 1605 - 1608, 2016.

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

[7] M. Yakout, L. Berti-Équill, and A. K. Elmagarmid, "Don't be SCAREd: Use SCalable Automatic REpairing with Maximal Likelihood and Bounded Changes", *Proceedings of the 13th International Conference on Management of Data*, pp. 553-564, 2013.

[8] N. Tang, "Big Data Cleaning", *Proceedings of the 16th International Conference on Web Technologies and Applications*, pp.13-24, 2014.

[9] J. Hipp, U. Gu¨ntzer, and U. Grimmer, "Data quality mining-making a virute of necessity," *Proceedings of the 6th Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD)*, pp. 52-57, 2001.

[10] S. Bruggemann, "Rule Mining for Automatic Ontology Based Data Cleaning," *Proceedings of the 10th International Conference on Asia-Pacific Web Conference Progress*, pp. 522-527, 2008.

[11] C. He, Z. Tan, Q. Chen, C. Sha, Z. Wang, and W. Wang, "Repair Diversification for Functional Dependency Violations," *Proceedings of the 19th International Conference on Database Systems for Advanced Applications*, pp. 468-482, 2014.

[12] F. Chiang and R. J. Miller, "A Unified Model for Data and Constraint Repair," *Proceedings of the 27th International Conference on Data Engineering*, pp. 46-457, 2011.

[۱۳] الناز زعفرانی معطر، محمدرضا فیضی درخشی و آزاده روحانی، «تشخیص هوشمند و خودکار غلط‌های تایپی در پایگاه داده‌های بزرگ بدون استفاده از لغت نامه»، مجله علمی پژوهشی مهندسی برق دانشگاه تبریز، جلد ۴۷ ، شماره ۱، صفحات ۸۱-۹۱، ۱۳۹۶.

[۱۴] مرتضی خرّم کشکولی و مریم دهقانی، «تشخیص، شناسایی و جداسازی عیب توربین گاز پالایشگاه دوم پارس جنوبی با استفاده از روش‌های ترکیبی داده کاوی، k-means، تحلیل مؤلفه‌های اصلی (PCA) و ماشین بردار پشتیبان (SVM)» ، مجله علمی پژوهشی مهندسی برق دانشگاه تبریز، جلد ۴۷ ، شماره ۲، صفحات ۵۰۱-۵۱۵، ۱۳۹۶.

[۱۵] یوکابد صدری، علی آقاگلزاده و مهدی ازوجی،«ادغام تصویر چندفوکوسه با استفاده از همدوسی فاز و خوشه‌بند K-means »، مجله علمی پژوهشی مهندسی برق دانشگاه تبریز، جلد ۴۵ ، شماره ۴، صفحات ۱۱۵-۱۲۵، ۱۳۹۴.

[16] M. Fijałkowska and B. Antoszewski, "Classification of congenital nasal deformities: a proposal to amend the existing classification, " *European Archives of Oto-Rhino-Laryngology*,vol. 274, no. 3, pp. 1231-1235.

[17] C. M. Teng, "Correcting Noisy Data," *Proceedings of the 16th International Conference on Machine Learning*, pp. 239-248, 1999.

[18] C. M. Teng, "A Comparison of Noise Handling Techniques," *Proceedings of the 14th International Conference on Artificial Intelligence Research Society*, pp. 269-273, 2001.

[19] C. M. Teng, "Polishing Blemishes: Issues in Data Correction," *Intelligent Systems*, vol. 19, no. 2, pp. 34-39, 2004.

[20] S. Kolahi and L. Lakshmanan, "On Approximating Optimum Repairs for Functional Dependency Violations," *Proceedings of the 12th International Conference on Database Theory*, pp. 53-62, 2009.

[21] G. Beskales, I. F. Ilyas, L. Golab and A. Galiullin, "On the Relative Trust between Inconsistent Data and Inaccurate Constraints," *Proceedings of the 29th International Conference on Data Engineering*, pp. 541-552, 2013.

[22] M. Volkovs, F. Chiang, J. Szlichta and R. J. Miller, "Continuous data cleaning," *Proceedings of the 30th International Conference on Data Engineering*, pp. 244-255, 2014.

[23] 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, 2014.

[24] X. Chu, J. Morcos, I. F. Ilyas, M. Ouzzani, P. Papotti, N. Tang and Y. Ye, "KATARA: A Data Cleaning System Powered by Knowledge Bases and Crowdsourcing," *Proceedings of the 15th International Conference on Management of Data*, pp. 1247 - 1261, 2015.

[25] W. Fan, S. Ma, N. Tang and W. Yu, "Interaction Between Record Matching and Data Repairing," *J. Data and Information Quality*, vol. 4, no. 4, 2014.

[26] J. Segeren, D. Gairola and F. Chiang, "CONDOR: A System for CONstraint DiscOvery and Repair," in *Proceedings of the 23rd International Conference on Information and Knowledge Management*, pp. 2078 - 2089, 2014.

[27] S. Song, H. Zhu and J. Wang, "Constraint-Variance Tolerant Data Repairing," *Proceedings of the 16th International Conference on Management of Data*, pp. 877 - 892, 2016.

[28] F. Chiang and S. Sitaramachandran, "Unifying Data and Constraint Repairs," *J. Data and Information Quality*, vol. 7, no. 3, pp. 1 - 26, 2016.

[29] J. Han, M. Kamber and J. Pei, *Data Mining Concept And Techniques*, Morgan Kaufmann Publishers, 3 Edition, 2011.

[30] H. Yao and H. J. Hamilton, "Mining functional dependencies from data," *Data Mining and Knowledge Discovery*, vol. 16, no. 2, pp. 197-219, 2008.

[31] Y. Huhtala, J. Kärkkäinen, P. Porkka and H. Toivonen, "Tane: An Efficient Algorithm for Discovering Functional and Approximate Dependencies," *The Computer Journal*, vol. 42, no. 2, pp.100-111, 1999.

[32] P. Kontkanen, J. Lahtinen, P. Myllymaki, T. Silander and H. Tirri, "Pre- and Post-processing in Machine Learning and Data Mining: Theoretical Aspects and Applications," *The workshop within Machine Learning and Applications Complex Systems Computation Group (CoSCo)*, pp. 38-47, 1999

[33] R. Agrawal and R. Ikant and D. Thomas, "Privacy Preserving OLAP," *Proceedings of the 5th International Conference on Management of Data*, pp. 251-262, 2005.

[34] I. Yoncaci, "Maximum a Posteriori Tree Augmented Naive Bayes Classifiers," *Lecture Notes in Artificial Intelligence*, 2004.

[35] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd.
Englewood Cliffs, NJ: Prentice-Hall, 1999.

زیرنویس‌ها

۱ Missing value
۲ Data warehouse
۳ Outlier
۴ Inconsistent
۵ Duplicate
۶ Data quality
۷ Data cleaning
۸ Conditional functional dependency
۹ Rules
۱۰ Ontology
۱۱ Functional dependency
۱۲ Semi-automatic
۱۳ Data type
۱۴ Ensemble learning
۱۵ Decision tree
۱۶ Bayes theorem
۱۷ Knowledge base
۱۸ Crowd powered

[۱۹] Categorical
[۲۰] Discretization
[۲۱] Gaussian distribution
[۲۲] Prior probability distribution

[۲۳] https://archive.ics.uci.edu/ml/datasets/Adult
[۲۴] http://archive.ics.uci.edu/ml/datasets/Physicochemical+Properties+of+Protein+Tertiary+Structure