شناسایی نشانه‌های حساب‌های جعلی با استفاده از تکنیک‌های داده‌کاوی

محسن خالقی[۱] و داریوش حسن‌پور آده[۲]
[۱]دانشگاه صنعتی اصفهان، دانشکده مهندسی برق و کامپیوتر
Email: mohesen.khaleghi@ec.iut.ac.ir
[۲]دانشگاه صنعتی اصفهان، دانشکده مهندسی برق و کامپیوتر
Email: d.hasanpoor@ec.iut.ac.ir

چکیده—روش‌های شناسایی تقلب را می‌توان به دو دسته‌ی روش های تشخیص سوءاستفاده و تشخیص ناهنجاری تقسیم کرد. تکنیک‌های مبتنی بر رویکرد تشخیص سوء استفاده زمانی استفاده می‌شود که تشخیص تقلب به صورت از پیش‌شناخته شده بوده و بر اساس امضای تقلب می توان رفتار جاری مشتریان را بررسی نمود. طبیعتا به دلیل شناخت کامل رفتار قبلی مشتریان، دقت شناسایی تقلب در این روش بسیار بالاست. اما نقطه ضعف این روش‌ها، عدم پوشش‌دهی کامل محدوده‌ی تقلب می باشد. بدین معنی که فقط تقلب‌هایی شناسایی می‌شوند که حداقل یک بار رخ داده و یا امضای آن به سیستم تشخیص تقلب ارائه شده باشد. در مقابل این روش‌ها، روش های دیگری هم هستند که مبتنی بر تشخیص ناهنجاری اند. یعنی سعی می‌کنند رفتار آتی مشتری را پیش‌بینی کرده و به این منظور تاریخچه‌ی رفتار وی را مورد بررسی قرار می‌دهند. در این‌گونه روش‌ها، قاعده‌ی ثابت و مشخصی برای تعریف تقلب تعریف نمی‌شود بلکه رفتار عادی و نرمال مشتری به سیستم تشخیص تقلب آموخته شده و هرگونه انحراف از آن به معنی تقلب فرض می‌گردد. بدیهی است که دقت در این روش پایین‌تر ولی گستره‌ی تقلب های کشف شده بالاتر است. که در این کار بروی پیاده‌سازی روش اول تمرکز شده است.

کلمات‌کلیدی—*داده‌کاوی، تشخیص تقلب، بانکداری‌الکترونیکی*

۱. مقدمه

در تکنولوژی امروزی فرصت‌های بسیاری برای تقلب الکترونیکی به دلیل گسترش رسانه‌ها و شبکه‌های کامپیوتری وجود دارد. یکی از تقلب‌های معمول، تقلب در فرایندها و تراکنش‌های خودپردازهای بانکی است. بدون توجه به نوع تقلبی که استفاده می‌شود، متقلبان تنها می‌توانند پول های قربانی را از حساب‌های متقلبانه جمع آوری کنند. بنابراین، شناسایی و کشف امضاهای این حساب‌های تقلبی بر اساس این امضاها خیلی مهم است.

روش شناسایی تقلب در این کار از تکنیک‌های مبتنی بر رویکرد تشخیص سوء استفاده زمانی می‌باشد، که تشخیص تقلب به صورت از پیش‌شناخته شده بوده که امضای تقلب می‌توان رفتار جاری مشتریان را مورد بررسی قرار می‌گیرد. طبیعتا به دلیل شناخت کامل رفتار قبلی مشتریان، دقت شناسایی تقلب در این روش بسیار بالاست. اما نقطه ضعف این روش‌ها، عدم پوشش دهی کامل محدوده‌ی تقلب می‌باشد. بدین معنی که فقط تقلب هایی شناسایی می‌شوند که حداقل یک بار رخ داده و یا امضای آن به سیستم تشخیص تقلب ارائه شده باشد.

این مطالعه، رده‌بندی و قواعدانجمنی را برای شناسایی علائم حساب‌های جعلی و الگوهای معاملات جعلی مورد استفاده قرار می‌دهد که این قوانین بر اساس علائم شناسایی شده از قبل برای طراحی یک سیستم نظارتی مورد استفاده قرار می‌گیرد. در حالت کلی این کار شامل بخش‌های اصلی زیر خواهد بود:
مشابه هر پروژه‌ی داده‌کاوی دیگر داشتن یک دیتاست مناسب شرط اصلی به پایان رساندن آن پروژه است، لذا جستجو برای یک دیتاست مناسب به عنوان اولین قدم منطقی به نظر می‌رسد، ولی متاسفانه به علت ماهیت پروژه و داده‌های مورد استفاده در آن تقریبا تمامی‌داده‌های مورد استفاده در مقاله‌های مرتبط با این زمینه در دسترس عموم قرار داده نشده‌اند؛ بنابرین برخلاف حساسیت مساله به علت کمبود مجموعه‌ی داده‌های واقعی، مدل‌های زیادی برای تشخیص تقلب توسعه داده نشده‌اند. ولی با این‌حال ما از مجموعه

داده‌های تقلب کارت‌های اعتباری آلمان[۲] که تقریبا تنها دیتاست در دسترس برای پروژه می‌باشد. با این حال این دیتاست خالی از ایراد نمی‌باشد؛ از ایرادات این دیتاست می‌توان به عدم نسبت حساب‌های تقلبی به حساب‌های عادی ۳۰٪ می‌باشد که بسیار زیاد است و همچنین ساختار داده‌ای ضعیفی دارد که در قسمت ۴ بیشتر توضیح داده‌خواهد شد و دلیل دیگری برای اینکه چرا لزوما این دیتاست ایده‌ال برای کارهای تشخیص حساب‌های جعلی نمی باشد به تعداد بسیار کم این دیتاست(۱۰۰۰ رکورد) می‌توان اشاره کرد. فایل این دیتاست با فرمت ARFF می‌باشد که یکی از فرمت های پشتیبانی شده در نرم‌افزار WEKA می‌باشد که در قسمت ۳ توضیح مختصری درباره‌ی این نرم‌افزار ارائه می‌شود.

بعد از یافتن دیتاست طبق اصول یادگرفته شده در طول ترم به کاوش و ارزیابی اولیه در میان داده‌ها می‌پردازیم که بتوان یک دیدگاه کلی نسبت به داده‌ها موجود در دست بدست بیاید که در نتایج این کاوش در بخش ۴ آورده شده است. سپس مطابق الگوی ارائه شده در مقاله نشانه‌های حساب‌های جعلی بدست می آوریم و نشانه‌های بدست آمده را با نشانه‌های معرفی شده در مقاله مقایسه می‌کنیم. در انتها یک بهبودی بر روی مدل ارائه شده در مقاله برای معماری سیستم تشخیص‌دهنده حساب‌های جعلی ارائه می کنیم.

۲. پیشینه‌ی تحقیق

Ghosh et al. در سال ۱۹۹۴[۳] یک مدل شبکه‌عصبی[۱] سه‌لایه پیش‌خور[۲] را ارائه کرده اند. این مدل برای شناسایی الگوهای تقلب استفاده شده است. در لایه‌ی خروجی یک مقدار عددی به عنوان رتبه‌ی تراکنش[۳] ایجاد می‌شود که اگر از یک مقدار آستانه ای پایین‌تر باشد آن تراکنش به عنوان تراکنش تقلبی تشخیص داده می‌شود.

Dorronsoro et al. در سال ۱۹۹۷[۴] نیز از روش شبکه‌های عصبی برای تشخیص تقلب استفاده کرده‌اند. برای ساختن مدل شبکه‌عصبی از یک آنالیز تفکیک‌کننده غیرخطی[۴] استفاده می‌شود. از آنجا که حجم بالایی از تراکنش‌ها باید در یک زمان مشخص مورد پردازش قرار گیرند، از یک سیستم امتیاز دهی استفاده شده است که برای امتیازدهی تنها از تراکنش‌هایی استفاده می‌کند که در یک بازه کوچک زمانی قبل از تراکنش مورد نظر رخ داده‌اند. این روش نسبت به روشی‌هایی که از همه‌ی تراکنش‌های پروفایل یک دارنده‌ی کارت استفاده می‌کنند، بخش بسیار کوچکتری از داده‌ها را بررسی می‌کند و در نتیجه کارایی بالاتری دارد.

Chan et al. در سال ۱۹۹۹[۵] یک روش توزیع شده برای کشف تقلب بکار برده‌اند. در این روش یک دیتاست بزرگ با داده‌های برچسب‌دار به چند زیرمجموعه‌ی کوچکتر تقسیم می‌شوند. سپس تکنیک‌های داده‌کاوی روی هریک از این زیرمجموعه‌ها اعمال شده و یک طبقه‌کننده[۵] برای هر کدام از زیرمجموعه‌ها بدست می آید. سپس از یک روش Meta-Learning استفاده می‌شود تا از ترکیب این طبقه‌کننده‌ها یک فرا-طبقه‌کننده ساخته می‌شود. در این روش هر یک از طبقه‌کننده‌ها به صورت یک جعبه سیاه دیده می‌شود. به این معنی که از هر الگوریتم یادگیری می‌توان در ایجاد آن‌ها استفاده نمود. در این تحقیق از ۵ الگوریتم Bayes, C4.5, ID3, CART, Ripper استفاده شده‌است. هم چنین در سیستم ارائه شده می‌توان از داده‌های توزیع شده نیز استفاده نمود، به این معنی که طبقه‌کننده هایی که از داده‌های مجزا و توزیع شده بدست آمده‌اند با هم ترکیب می‌شوند. اگر فرمت دیتاست‌ها با یکدیگر سازگار نباشد از یکی از دو روش برای سازگاری استفاده می‌شود؛ در روش اول یک عامل پل‌زنی در دیتاستی که شامل صفاتی اضافه بر دیتاست دیگر است ساخته می‌شود که وظیفه آن تخمین مقادیر این صفت در دیتاست فاقد این صفت است. سپس از روی این مقادیر تخمینی طبقه‌کننده دیتاست دوم ایجاد می‌شود. در روش دوم برای دیتاستی که شامل صفاتی اضافه بر دیتاست دیگر است دو طبقه‌کننده یکی فاقد صفت مزبور و دیگری با در نظر گرفتن صفت مزبور ساخته می‌شود. از طبقه‌کننده اول برای تبادل اطلاعات و از طبقه‌کننده دوم به صورت محلی استفاده می‌شود. هم چنین در این سیستم به دلیل اینکه طبقه کننده‌های زیادی ساخته می‌شود ممکن است برخی از آنها افزونه باشند. لذا از روش هایی برای هرس کردن طبقه‌کننده‌ها و ایجاد یک مجموعه بهینه از آنها استفاده می‌گردد. برای ترکیب طبقه‌کننده‌ها نیز از روش های متعددی می‌توان استفاده نمود. یک روش Class-Combiner است که یک مجموعه‌ی آزمایشی می‌سازد که صفات آن پیش‌بینی‌های هر یک از دسته‌کننده‌هاست و برچسب آن نیز برچسب واقعی طبقه‌بندی است. سپس از این مجموعه‌ی آزمایشی برای یادگیری یک طبقه‌کننده‌ی ترکیبی استفاده می‌شود. این روش از روش‌های رای‌گیری مثل روش Adaboost بسیار کاراتر عمل می کند. این سیستم به موسسات مالی امکان می‌دهد که مدل‌های تقلب

[۱] Neural Network
[۲] Feed-Forward
[۳] Transaction Rank
[۴] Non-linear Discriminant Analysis
[۵] Classifier

خود را از طریق طبقه‌کننده‌ها با یکدیگر مبادله کنند.
Tue et al. در سال ۲۰۰۴[۶] چارچوبی بر پایه سیستم‌های ایمنی و Case Based Reasoning برای تشخیص تقلب ارائه داده‌اند. ابتدا مجموعه‌ای از موارد نرمال و تقلبی از روی داده‌های برچسب دار ساخته می‌شود. سپس کشف‌کننده‌های اولیه با الگوریتم‌های تصادفی و یا ژنتیک ساخته می‌شوند. سپس عملیات Negative Selection و Clonal Selection بر روی کشف‌کننده‌های اولیه اعمال می‌شود تا مجموعه‌ای از کشف‌کننده‌ها با الگوهای متفاوت بدست آید که می‌توانند انواع تقلب را کشف کنند. زمانی که یک تراکنش جدید می‌آید، شباهت این تراکنش با کشف‌کننده‌ها محاسبه می‌شود و اگر این شباهت از آستانه‌ای بالاتر بود هشدار داده می شود. پس از تایید یک فرد خبره مبنی بر تقلبی بودن یا نبودن آن تراکنش، از آن برای بهبود مدل استفاده می‌شود.
Vatsa et al. در سال ۲۰۰۵[۷] یک سیستم دو لایه ارائه داده‌اند که در لایه‌ی اول قوانین عمومی‌و قوانین خاص هر مشتری بکار گرفته می‌شود تا درجه‌ی مشکوک بودن یک تراکنش را مشخص کند. اما از آنجا که این قوانین ثابت هستند و ممکن است میزان تراکنش های تقلبی که تشخیص داده نمی‌شوند و تراکنش‌های غیرتقلبی که به عنوان تقلبی معرفی می‌شوند زیاد باشد لایه دومی‌به کار گرفته می‌شود که از تکنیک‌های تئوری‌بازی برای تشخیص تقلب استفاده می‌کند. فرد متقلب و سیستم تشخیص تقلب دو طرف این بازی هستند که هریک سعی دارند سود خود را ماکزیمم کنند.
Gadi et al. در سال ۲۰۰۸[۸] از ۵ روش دسته‌بندی برای تشخیص تقلب استفاده کرده‌اند: Neural Network, Naive Bayes, Bayesian Network, Artificial Immune System و Decision Tree. برای پیاده‌سازی این روش‌ها از ابزار WEKA استفاده شده است. به جز روش Artificial Immune System که برنامه جداگانه ای برای آن نوشته شده‌است، هریک از این روش‌ها در دو حالت حساس به هزینه و ساده ارزیابی شده‌اند. به‌این معنی که در حالت اول هزینه‌ی مربوط به موارد نرمالی که به اشتباه تقلبی تشخیص داده می‌شوند با هزینه‌ی مربوط به موارد تقلبی که به اشتباه نرمال تشخیص داده می‌شوند متفاوت است. هم چنین برای روش‌هایی که پارامتری هستند یکبار با پارامترهای پیش فرض WEKA و یکبار با پارامترهای بهینه شده مورد ارزیابی قرار گرفته‌اند. نتایج مقایسه‌ی این دو حالت نشان می‌دهد که در هیچ یک از روش‌ها پارامترهای پیش فرض WEKA بهینه نبوده‌اند. برای بهینه‌سازی پارامترها از الگوریتم ژنتیک استفاده شده‌است.

۳. ابزارهای مورد استفاده

برای به نهایت رساندن پروژه در کنار از نرم‌افزار WEKA که به خاطر راحتی برای مراحل تجزیه و تحلیل اکتشافی داده‌ها[۶]؛ از زبان برنامه‌نویسی R برای مراحل مدل‌سازی استفاده می‌کنیم. که در زیر توضیحی خلاصه‌ای در مورد هریک از ابزارهای فوق الذکر ارائه می‌دهیم.

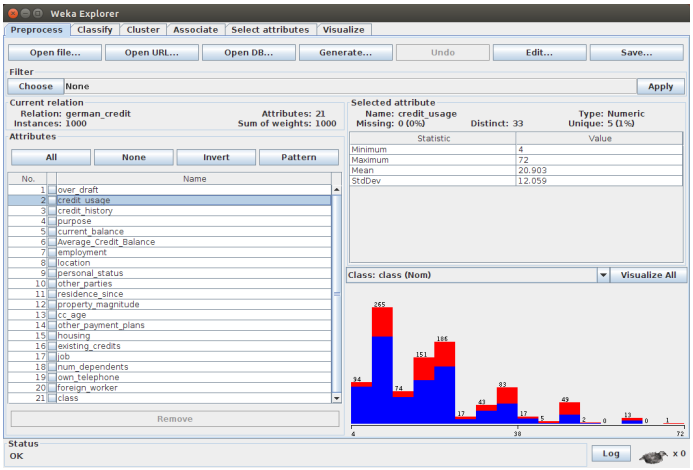
آ. نرم‌افزار *WEKA*

وکا نام یک نرم‌افزار متن‌باز[۷] است که شامل مجموعه‌ای از الگوریتم‌های یادگیری ماشینی و داده‌کاوی می‌شود. این ابزار در دانشگاه وایکاتو در کشور نیوزلند توسعه داده شده است. وکا در تحلیل داده‌های عظیم کاربرد دارد. تمام قسمت‌های این نرم‌افزار به زبان‌جاوا نوشته شده‌است و در نتیجه می‌تواند بر روی هر پلتفرمی[۸] اجرا گردد. رابط اصلی کاربری WEKA اکسپلورر[۹] است، اما با استفاده از خط فرمان نیز امکانات اکسپلورر قابل دسترسی است. همچنین آزمونگر[۱۰] نیز امکان اجرای الگوریتم‌های مختلف رده بندی به صورت هم‌زمان و مقایسه نتایج آن‌ها وجود دارد. تمامی شاخص‌های مورد نیاز به منظور بررسی مدل‌های رده‌بندی در این قسمت تعریف شده و قرار دارند و گزارشات مفصلی را از جمله آزمون T می توان در این قسمت پس از مدلسازی استخراج نمود.

ب. زبان برنامه‌نویسی *R*

R یک زبان برنامه‌نویسی و محیط نرم‌افزاری متن‌باز برای محاسبات آماری و تحلیل داده است، که بر اساس زبان‌های S و Scheme پیاده‌سازی شده است. از ویژگی‌های R می‌توان گفت که R ، حاوی محدوده‌ی گسترده‌ای از تکنیک‌های آماری (از جمله: مدل‌سازی خطی و غیرخطی، آزمون‌های کلاسیک آماری، تحلیل سری‌های زمانی، رده‌بندی، خوشه‌بندی و غیره) و قابلیت های گرافیکی است. در محیط R ، کدهای C, C++, Fortran قابلیت اتصال و فراخوانی هنگام اجرای برنامه را دارند.

---
[۶] Exploratory Data Analysis (EDA)
[۷] Open Source
[۸] Platform
[۹] Explorer
[۱۰] Experimenter



شکل ۱: نمونه مثال نمایش توزیع داده‌ای در WEKA مرتبط با متغییر credit_usage

۴. تجزیه و تحلیل اکتشافی دیتاست

در این قسمت به بررسی دیتاست انتخاب شده می‌پردازیم؛ در جواب این سوال که آیا این دیتاست متشکل از داده‌های واقعی می‌باشد یا خیر با توجه به تعداد کم داده‌ها با تردید می‌توان سخن گفت. بعد از آنکه دیتاست را در قسمت اکسپلورر نرم‌افزار WEKA بارگذاری کردیم گزاش اولیه‌ای از وضعیت دیتاست بدست می‌دهد این است که این دیتاست شامل ۱۰۰۰ رکورد است و هر رکورد از ۲۱ ویژگی متشکل شده است که در هیچ از رکوردها ویژگی ای گم نشده است. جدول ۱ مشخصات این ویژگی‌ها آورده شده است.

بعد از بارگذاری دیتاست در WEKA در تب[۱۱] Preprocess نرم افزار یک دید کلی نسبت به توزیع داده‌ای هریک از ویژگی‌ها و نسبت آنها به متغییر هدف که در اینجا ویژگی Class می‌باشد، ارائه می‌دهد. که در اینجا کارت‌های اعتباری‌ای که به عنوان جعلی نشانه‌گذاری[۱۲] شده‌اند با رنگ قرمز نشان داده شده‌اند. در این تب با انتخاب هر کدام از ویژگی‌ها می‌توان توزیع داده‌ای آن متغییر نسبت به متغییر هدف و همچنین توزیع آماری داده‌های آن متغییر در کل دیتاست را مشاهده کرد(که برای متغییرهای دسته‌ای تعداد رکوردهای مرتبط با هر کدام از دسته‌ها را می‌آورد و برای متغییر عددی مقادیر حداقل و حداکثر و میانگین و انحراف از معیار را ارائه می‌دهد.)، که یک نمونه در شکل ۱ آورده شده است.

همچنین در تب Visualize که مختص به تصویر کردن داده ها می‌باشد، توزیع هر دو متغییر را نسبت به هم ارائه می‌دهد. با انتخاب بر روی یکی از نمودارها یک پنجره‌ی دیگر باز می‌شود که متغییرها را براساس یک‌دیگر نمایش داده است؛ از این پنجره اگر بروی هریک از نمودارهای موجود در سمت راست پنجره انتخاب کنیم متغییر روی محور افقی نمودار را عوض میکنیم. اگر در تب Visualize بروی یکی از نمودارهای سطر اول کلیک کنیم از پنجره ای که باز می‌شود، می‌توانیم توزیع متغییر هدف را بر اساس تک تک متغییرها مشاهده کنیم، که در این نمودارها محور افقی متغییر مستقل دیتاست و محور عمودی متغییر وابسته(هدف) می‌باشد. که بعد از گشتی در میان این نمودارها به راحتی می‌توان نتیجه گرفت که هیچ یک از متغییرها **به تنهایی** ضریب همبستگی بالایی با متغییر هدف ندارند(به علت عدم مشاهده‌ی عینی الگویی میان متغییر مستقل و متغییر وابسته)، لذا برای اینکه بتوان مدلی ارائه داد نیاز داریم متغییرهای توامی را انتخاب کنیم که دارای ضریب همبستگی بالایی با متغییر هدف دارند، که انتخاب اینکه کدامین متغییرها مدل بهتری برای پیشبینی و توصیف متغییر هدف را می تواند ارائه دهد را با استفاده از الگوریتم‌های داده‌کاوی که بخش ۵ آورده شده است.

۵. استخراج مدل

ما تصمیم گرفتیم که متغییر پیش‌بین را با استفاده از ۳ الگوریتم بیزین‌ساده[۱۳]، SVM و درخت‌تصمیم پیش‌بینی کنیم، ما بعد از بارگذاری دیتاست، آن را به میزان ۱۰۰ بار بر زدیم[۱۴] که احتمال اینکه یک توزیع متناسب و یکسان برای هریک از مجموعه داده‌های آموزشی و تست بالا باشد. سپس ما داده‌ی آموزشی را به نسبت ۷ به ۱۰ به دو مجوعه آموزشی و تست تقسیم می‌کنیم و سپس این مجموعه‌ها را در دو فایل ذخیره می‌کنیم که از دفعات بعدی فقط آن‌ها بارگذاری کنیم و از تقسیم مکرر و غیر ضروری داده‌ی آموزشی و تقسیم بعدی آن اجتناب کنیم.

بعد از این‌که داده‌های آموزشی و تست بارگذاری شدند، داده‌ی آموزشی را به هریک از ۳ الگوریتم‌های یادگیری مدل اشاره شده داده می‌شود و مدل استخراج شده هریک را در یک متغییر ذخیره می‌کنیم، سپس داده‌های تست را به مدل‌های استخراج شده می‌دهیم و نتایج را در قابل یک ماتریس درهم‌ریختگی[۱۵]

[۱۱] Tab
[۱۲] Flag
[۱۳] Naïve Bayesian
[۱۴] Shuffle
[۱۵] Confusion Matrix

Attributes	Description	Data Type	Valid Ranges/Categories
over_draft	Status of existing checking account	Qualitative	$\{< 0,\ 0 \leq \ldots < 200,\ \geq 20,$ no checking$\}$
credit_usage	Duration in month	Numerical	–
credit_history	Credit history	Qualitative	'no credits/all paid', 'all paid', 'existing paid', ...
purpose	Purpose	Qualitative	'new car', 'used car', 'furniture/equipment', ...
current_balance	Credit amount	Numerical	–
Average_Credit_Balance	Savings account/bonds	Qualitative	$\{< 100,\ 100 \leq \ldots < 500,\ 500 \leq \ldots < 1000,\ \ldots\}$
employment	Present employment since	Qualitative	$\{$unemployed, $< 1,\ 1 \leq \ldots < 4,\ 4 \leq \ldots < 7,\ \geq 7\}$
location	Installment rate in percentage of disposable income	Numerical	–
personal_status	Personal status and sex	Qualitative	'male div/sep', 'female div/dep/mar', 'male single', ...
other_parties	Other debtors / guarantors	Qualitative	'none', 'co applicant', 'guarantor'
residence_since	Present residence since	Numerical	–
property_magnitude	Property	Qualitative	'real estate', 'life insurance', 'car', 'no known property'
cc_age	cc_age in months	Numerical	–
other_payment_plans	Other installment plans	Qualitative	'bank', 'stores', 'none'
housing	Housing	Qualitative	'rent', 'own', 'for free'
existing_credits	Number of existing credits at this bank	Numerical	–
job	Job	Qualitative	'unemp/unskilled non res', 'unskilled resident', ...
num_dependents	Number of people being liable to provide maintenance for	Numerical	–
own_telephone	Telephone	Qualitative	'yes', 'none'
foreign_worker	Foreign worker	Qualitative	'yes', 'no'
class	Fraud status	Qualitative	'good', 'bad'

جدول۱: جزئیات ویژگی‌های دیتاست[۲]

نمایش می‌دهیم. نتایج حاصل از این مدل‌ها در جدول ۲ آمده است.

Measures	Naive Bayesian	SVM	Decision Tree
Accuracy	0.76	0.77	0.72
95% CI	(0.70, 0.80)	(0.71, 0.81)	(0.66, 0.77)
Sensitivity	0.53	0.36	0.45
Specificity	0.85	0.94	0.84
P-Value	0.013	0.003	0.173

جدول۲: نتایج هریک از مدل‌های استخراج شده توسط ۳ الگوریتم بیزین‌ساده، SVM و درخت تصمیم برای داده‌های تست

همان‌طور که می‌بینیم الگوریتم SVM مدل بهتری را برای پیش‌بینی متغییر هدف ارائه داده‌است ولی در اینجا سوالی مطرح می‌شود که چگونه می‌توان سیستم جامعی طراحی کرد که بتواند این مدل‌ها را باهم به کار گیرد؟ زیرا که لزوما برای یک نمونه هر ۳تای این مدل‌ها یک پیش‌بینی یکسانی را ارائه نمی‌دهند، مثلا برای یک نمونه احتمال این‌که آن نمونه به کدام یک از کلاس‌ها می‌باشد در جدول ۳ آمده است[۱۶].

Algorithm \ Class	Bad	Good
Naive Bayesian	0.165	0.835
SVM	0.502	0.498
Decision Tree	0.192	0.808

جدول۳: احتمالات تعلق به کلاس‌ها برای یک نمونه از مجموعه داده‌ی تست

نمونه‌ای در جدول فوق که احتمالات تعلقش به کلاس‌ها آورده شده است در اصل به کلاس Good متعلق است، همان‌طور که می‌بینیم دو مدل بیزین‌ساده و درخت‌تصمیم درست ارزیابی کرده‌اند ولی مدل SVM در تصمیم‌گیری راجع به کلاس نمونه مردد است. با اینکه همان‌طور که در جدول ۲ آمده است، SVM مدل بهتری نسبت به دیگر الگوریتم‌ها ارائه داده است ولی در اینجا دیگر مدل ها بهتر توانسته‌اند دسته‌بندی کنند.

حال با برگشت به سوالی که قبلا مطرح کرده بودیم میبنی بر اینکه «چگونه می‌توان سیستم جامعی طراحی کرد که بتواند این

[۱۶]این نمونه و احتمالات یک مثال واقعی از مجموعه‌داده‌ی تست می‌باشد.

مدل‌ها را باهم به کار گیرد؟» ما انتگرال فازی چوکت[۱۷] را برای این منظور معرفی می‌کنیم.

آ. انتگرال فازی چوکت

توضیح دقیق راجع به انتگرال فازی چوکت از حوصله‌ی موضوع این نوشتار خارج است ولی بطور خلاصه اگر بخواهیم هدف این انتگرال را معرفی کنیم، می‌توان گفت که این انتگرال برای ترکیب اطلاعات منابع که دارای خاصیت اندازه‌گیری‌ـ غیرافزایشی[۱۸] می‌باشد؛ معرفی شده است. خاصیت اندازه‌گیری‌ـ غیرافزایشی در حالت کلی می‌گوید که اهمیت بودن چند منبع اطلاعاتی با هم برابر با مجموع اهمیت تک‌تک آن‌ها نمی‌باشد. به عنوان مثال:

$$\mu(\{۱,\ ۲\}) \neq \mu(\{۱\}) + \mu(\{۲\})$$

انتگرال فازی چوکت را به صورت ۱ معرفی می‌کنیم.

$$\mathcal{C}_\mu(x) = \sum_{i=۱}^{n} (x_{\tau(i)} - x_{\tau(i-۱)}) \mu(\{\tau(i), \ldots, \tau(n)\}) \qquad (۱)$$

$$x_{\tau(۱)} \leq x_{\tau(۲)} \leq \ldots \leq x_{\tau(n)},\ x_{\tau(۰)} = ۰$$

که در آن n تعداد مدل‌های بدست آمده است(که در مورد کار ما $n = ۳$ می‌باشد.) و x_i میزان احتمال تعلق نمونه به کلاس Good می‌باشد و مجموعه‌ی τ اندیس مدل‌های مرتب شده با توجه به میزان احتمال تعلق به کلاس Good می‌باشد. μ نیز میزان اهمیت نتایج ترکیب الگوریتم‌ها با هم می‌باشد که بطور تجربی و با استفاده از دید کلی‌ای که از عملکرد ۳ الگوریتم داریم تنظیم کرده‌ایم که به صورت زیر می‌باشند.

$$\mu(\{\emptyset\}) = 0,\ \mu(\{\text{NB, SVM, TR}\}) = 1$$
$$\mu(\{\text{NB}\}) = \mu(\{\text{SVM}\}) = \mu(\{\text{TR}\}) = \frac{1}{3}$$
$$\mu(\{\text{NB, SVM}\}) = 0.8$$
$$\mu(\{\text{NB, TR}\}) = \mu(\{\text{SVM, TR}\}) = \frac{2}{3}$$

همان‌طور که می‌بینیم خواصیت اندازه‌گیری‌ـغیرافزایشی در درجات اهمیت فوق مشاهده می‌شود.

ب. استخراج مدل نهایی با استفاده از انتگرال فازی چوکت

با درنظر گرفتن انتگرال چوکت می‌توان به مدل‌های بدست آمده قبلی به عنوان مدل‌های میانی نگاه کرد و مدل نهایی که از ترکیب مدل‌های آن‌ها با استفاده از انتگرال چوکت بدست می آید به عنوان مدل نهایی در نظر گرفت. که نتایج مدل‌هایی بدست آمده برای طبقه‌بندی نمونه‌های تست مورد استفاده در جدول ۲ به صورت ۴ آمده است. همان‌طور که می‌بینیم در حالت کلی مدل ارائه شده توسط انتگرال چوکت از مدل‌های تکی بهتر عمل کرده است.

۶. نتایج بدست آمده با استفاده از دیگر الگوریتم‌های طبقه‌بند

مدل ارائه و کد شده‌ای که در قسمت ۵ ارائه شد، سعی شده بود علاوه بر کدنویسی و ارائه این ایده که از انتگرال فازی می توان برای ایجاد یک مدل که در صورت نتظیم صحیح و معقولانه ی پارامترهای این انتگرل می‌تواند لزوما یک مدل بهتری نسبت به دیگر مدل‌ها ارائه دهد، در این قسمت نتایج الگوریتم‌های متنوع ای که با استفاده از جعبه‌ابزار ارائه شده توسط WEKA بدست آمده است را در جدول ۵ آورده‌ایم. در تمامی این الگوریتم‌ها از روش تست 10-FOLD استفاده شده است. اگر نتایج نشان داده شده در جدول ۴ با جدول ۵ مقایسه کنیم مشاهده می‌شود که دقت بدست آمده توسط مدل ارائه شده توسط انتگرال چوکت از تمامی الگوریتم‌های تست شده(شامل ۳ الگوریتم اصلی کد شده) بهتر بوده است.

۷. نتیجه‌گیری

در مقاله‌ی اصلی[۱] مورد مطالعه‌ی این پروژه‌ی ما به استفاده از دو روش قوانین‌انجمنی و بیزین‌ساده تلاشی برای ارائه‌ی مدلی برای تشخیص و طبقه‌بندی حساب‌های جعلی کرده است که در این تلاش ایرادات فراوانی را می‌توان نسبت داد، اول اینکه فقط به دو نمونه الگوریتم طبقه‌بند اکتفا کرده است و در مورد نتایج دیگر روش‌ها نظری نداده است. دوم این که مدل استخراجی از هرکدام از الگوریتم‌ها توسط یک انسان باید به کار گرفته می‌شد یعنی یک سیستم اتوماتیک ارائه نشده بود. که در این پروژه ما به بررسی نتایج چندین الگوریتم پرداختیم و همچنین با ارائه این ایده که می‌توان از انتگرال فازی برا ترکیب مدل‌ها به عنوان مدل واسط و بدست

---

[۱۷] Fuzzy Choquet Integral
[۱۸] Non-Additive Measure

Measures	Naive Bayesian	SVM(kernel → rbfdot)	Decision Tree	**FCI**
Accuracy	0.76	0.77	0.72	**0.78**
95% CI	(0.70, 0.80)	(0.71, 0.81)	(0.66, 0.77)	**(0.73, 0.83)**
Sensitivity	0.53	0.36	0.45	**0.52**
Specificity	0.85	0.94	0.84	**0.90**
P-Value	0.013	0.003	0.173	**0.0003**

جدول۴: مقایسه‌ی نتایج بدست آمده با مدل نهایی با استفاده از انتگرال چوکت با نتایج قبلی

Algorithm	Accuracy	ROC
Neural Network	0.720	0.733
LOG	0.752	0.785
LOG Simple	0.759	0.792
Bayas Net	0.755	0.780
Meta. Classification via Regression	0.759	0.780
Meta. Multiclass Classifier	0.752	0.785
Meta. Random Committee	0.739	0.785
Hoeffding Tree	0.750	0.785
Tree. LMT	0.759	0.792

جدول۵: دقت‌ها و مساحت زیر سطح نمودار ROC برای الگوریتم های متفاوت – اجرا شده در WEKA

آوردن یک مدل نهایی می‌توان به یک سیستم کاملا اتوماتیک دست یافت که فاکتور عامل انسانی را از معادله حذف کردیم.

همچنین طبق نتایج ارائه شده نشان دادیم که مدلی که از انتگرال فازی بدست آمده است از تمامی الگوریتم‌های تست شده نتایج بهتری را بدست آورده است. درانجام این پروژه تنها مشکلی که داشتیم پیدا کردن یک دیتاست مرغوب و واقعی بود که به علت ماهیت پروژه و حساسیت امنیتی حاکم بر افشای داده‌های موجود در این زمینه، داده‌ای در این رابطه در دسترس عموم موجود نمی باشد.

مراجع

[1] Li, S.-H., D.C. Yen, W.-H. Lu, and C. Wang, *"Identifying the signs of fraudulent accounts using data mining techniques"*, Computers in Human Behavior,Vol. 28,No. 3, 2012, p. 1002-1013.

[2] German credit fraud dataset, *http://weka.8497.n7.nabble.com/file/n23121/credit_fruad.arff*, Online; accessed May 30 2015.

[3] Ghosh and D.L. Reilly, *Credit Card Fraud Detection with a Neural-Network*, IEEE, vol. 3, pp. 621-630, 1994.

[4] Dorronsoro, Ginel, Sanchez, and Cruz, *Neural fraud detection in credit card operations* IEEE, vol. 8, pp. 827-843, 1997.

[5] Chan, Fan, Prodromidis and Stolfo, *Distributed Data Mining in Credit Card Fraud Detection*, IEEE, vol. 14, pp. 67-74, 1999.

[6] Tue, Ren and Liu, *Artificial Immune System for Fraud Detection*, IEEE, vol. 2, pp. 1407-1411, 2004.

[7] Vatsa, Sural and Majumdar, *A Game-Theoretic Approach to Credit Card Fraud Detection* Springer, vol. 3803, pp. 263-276, 2005.

[8] Gadi, Wang and Lago, *Comparison with Parametric Optimization in Credit Card Fraud Detection* IEEE, pp. 279-285, 2008.