فصلنامه علمی پژوهشی دانش سرمایه‌گذاری
دوره ۱۴ / شماره ۳ (پیاپی ۵۵) / پائیز ۱۴۰۴
صفحه ۵۳۱ تا ۵۵۶

مدلسازی ریسک اعتباری بازار رمزارزها با استفاده از یادگیری ماشین: کاربرد در تشخیص پولشویی در معاملات بیت‌کوین

زهرا بزرگ تبار بائی
دانشجوی دکتری تخصصی مهندسی مالی، گروه مدیریت، واحد رشت، دانشگاه آزاد اسلامی، رشت، ایران
z_bozorgtabar@yahoo.com

رضا آقاجان نشتائی
گروه مدیریت بازرگانی، واحد رشت، دانشگاه آزاد اسلامی، رشت، ایران (نویسنده مسئول)
Nashtaei@iaurasht.ac.ir

محمد حسن قلیزاده
گروه مدیریت، دانشگاه گیلان، رشت، ایران
gholizadeh@guilan.ac.ir

تاریخ دریافت: ۱۴۰۲/۰۴/۲۷ تاریخ پذیرش: ۱۴۰۲/۰۵/۲۲

چکیده

هدف از این پژوهش ارائه درک عمیق‌تری از مدل‌سازی ریسک اعتباری و ارزیابی عملکرد الگوریتم‌های یادگیری ماشین و یادگیری عمیق در تشخیص پولشویی (به عنوان جنبه‌ای از ریسک اعتباری) در تراکنش‌های بیت‌کوین است. برای این منظور از شش الگوریتم مختلف یادگیری ماشین، شامل شبکه عصبی مصنوعی (ANN)، جنگل تصادفی (RF)، K-نزدیکترین همسایه (KNN)، ماشین بردار پشتیبان (SVM)، و دو الگوریتم یادگیری عمیق شامل شبکه باور عمیق (DBN) و حافظه کوتاه‌مدت بلند (LSTM) استفاده شده است. به علاوه، داده‌های تشخیص پولشویی الیپتیک مربوط به معاملات بیت‌کوین در این پژوهش به عنوان مجموعه داده مورد استفاده در روش‌های یادگیری ماشین استفاده شده است. نمونه آماری داده‌های تراکنش‌های مربوط به سال ۲۰۲۱ میلادی را پوشش می‌دهد. تجزیه و تحلیل محاسباتی با استفاده از نرم افزار R (نسخه ۳.۴.۰) و متلب انجام شده است. نتایج نشان داد الگوریتم‌های جنگل تصادفی، ماشین بردار پشتیبان (SVM) و DBN بهترین عملکرد را ارائه کردند. سایر الگوریتم‌ها، از جمله LSTM ، KNN، و ANN نیز عملکرد خوبی داشتند، اما عملکرد آنها در مقایسه با جنگل تصادفی، SVM و DBN پایین‌تر است. به طور کلی، این مطالعه پتانسیل یادگیری ماشین و الگوریتم های یادگیری عمیق را در تشخیص پولشویی در شبکه بیت کوین برجسته می‌کند.

واژه‌های کلیدی: تشخیص پولشویی، یادگیری ماشین، بیت‌کوین، یادگیری عمیق.

۱- مقدمه

مشکل دسترسی محدود مالی (مانند محدودیت‌های حساب بانکی) تا حدی نتیجه ناخواسته مقررات ضد پولشویی (AML[1])سختگیرانه است، که اگرچه برای حفاظت در برابر ریسک مالی ضروری بوده، اما به طور نامتناسبی منجر به سختگیری شده و دارای تأثیر منفی بر افراد کم درآمد، مهاجران و پناهندگان است. تقریباً ۱.۷ میلیارد بزرگسال در دنیا بدون حساب بانکی هستند. مشکل هزینه‌های بانکی بالاتر نیز تا حدی تابعی از سیاست AML است که باعث هزینه‌های ثابت بالای انطباق با کسب‌وکارهای خدمات پولی همراه با ترس از مجازات‌های کیفری و پولی برای مشتریان کم درآمد می‌شود زیرا برای سیستم‌های مالی ارزش ریسک را ندارند (وبر[۲] و همکاران، ۲۰۱۹).
با این حال، سیاست AML را نمی‌توان از میان برداشت. صنایع غیرقانونی بسیار بزرگ مانند گروه‌های مواد مخدر، قاچاق انسان و سازمان‌های تروریستی باعث مشکلات بسیار و همچنین حجم زیادی از عملیات پولشویی هستند. پولشویی یک جرم وسیع بوده و روش‌های فعلی سیستم مالی سنتی عملکرد ضعیفی برای جلوگیری از آن انجام می‌دهند. بر این اساس، سوال این است که آیا با ابزارهای مناسب و داده‌های در دسترس، می‌توانیم به تطبیق نیاز به ایمنی با تشخیص بهتر عملیات پولشویی کمک کنیم؟

ظهور ارز رمزنگاری شده توسط بیت کوین باعث بروز علاقه شدید به فناوری و کارآفرینی در حوزه پردازش عملیات پرداخت شد. در سرتاسر جهان، استارت‌آپ‌های انتقال پول برای رقابت با بانک‌های قدیمی شروع به کار کردند. آنها بر روی امکان انتقال پول نقد کم هزینه و همتا به همتا در داخل و خارج از مرزها با استفاده از بیت‌کوین و سایر ارزهای رمزنگاری شده تمرکز کردند. بسیاری به صراحت مسئله ارزابی ریسک را هدف قرار دادند و از رمزارزها به عنوان راهی برای شمول مالی حمایت کردند. در کنار این کارآفرینان، جامعه‌ای از دانشگاهیان و سیاستمدارن رشد کردند که از ملاحظات نظارتی به روز شده برای ارزهای دیجیتال حمایت می‌کردند (آلوبیتی[۳] و همکاران، ۲۰۲۲).

پس از کاهش این هیجان، بیت‌کوین با چالش‌های امنیتی روبه‌رو شد. بسیاری از مجرمان از بیت‌کوین برای مخفی‌شدن در معرض دید عموم، انجام حملات باج‌افزار و راه اندازی بازارهای سیاه برای مبادله کالاها و خدمات غیرقانونی استفاده کردند (وبر و همکاران، ۲۰۱۹).

در سال ۲۰۱۹، شبکه مدیریت جرایم مالی (Fin-CEN[4]) از ایالات متحده دستورالعمل جدیدی در مورد چگونگی اعمال قانون رازداری در عملیات بانکی (BSA[5]) در مورد ارزهای دیجیتال یا آنچه FinCEN آن را ارزهای مجازی قابل تبدیل (CVC[6]) می‌نامد ارائه کرد (سی[۷]، ۲۰۲۳). مطابق با BSA، این دستورالعمل از کسب‌وکارهای خدمات پولی می‌خواهد تا ارزیابی‌های ریسک فردی را برای اندازه‌گیری قرار گرفتن در معرض پولشویی، تأمین

[1] Anti-Money Laundering
[2] Weber
[3] Alotibi
[4] Financial Crimes Enforcement Network
[5] Bank Secrecy Act
[6] Convertible Virtual Currencies
[7] See

مالی تروریسم و سایر جرایم مالی ایجاد کنند. این ارزیابی ها بر اساس ترکیب مشتری، جغرافیای ارائه شده و محصولات یا خدمات مالی ارائه شده است. ارزیابی‌ها باید مدیریت روابط با مشتری، از جمله اجرای کنترل‌های متناسب با ریسک را مشخص کند. به عبارت دیگر، کسب‌وکارهای خدمات پولی نه تنها باید حساب های مشکوک را گزارش کنند، بلکه باید علیه آنها نیز اقدام کنند (مثلاً آنها را مسدود کنند). این راهنما یک مدل ارزیابی ریسک به خوبی توسعه یافته را به عنوان کمکی به کسب‌وکارهای خدمات پولی در شناسایی و ارائه تجزیه و تحلیل جامع از مشخصات ریسک فردی خود تعریف می‌کند. با تقویت الزامات ناظر بر شناخت مشتریان BSA، این دستورالعمل از کسب‌وکارهای خدمات پولی می‌خواهد که «در مورد مشتریان خود اطلاعات کافی داشته باشند تا بتوانند سطح ریسک آن‌ها را تعیین کنند».

معنای "اطلاعات کافی" در مورد مشتری موضوع بحث‌های زیادی در محافل مالی است. در عمل، یکی از چالش‌برانگیزترین جنبه‌های این امر، الزام ضمنی به این است که نه تنها مشتری باید به خوبی شناخته شود، بلکه باید مبدا و مقصد جریان‌های مالی مشتری نیز به خوبی شناخته شود. در اکوسیستم داده‌های مالی پراکنده سنتی، این جنبه اغلب با تماس‌های تلفنی یا استعلام‌های بین کسب‌وکارهای خدمات پولی اجرا می‌شود. اما در سیستم بیت‌کوین، داده‌های شبکه تراکنش‌های آن نموداری کامل در دسترس عموم است، البته به شکل بی‌نام و بدون برچسب. برای استفاده کردن از فرصتی که این داده‌های عمومی ارائه می‌کند، شرکت‌های تحلیل اطلاعات ارزهای دیجیتال به وجود آمده‌اند تا راه‌حل‌های AML را متناسب با حوزه ارزهای دیجیتال ارائه دهند. در حالی که بی‌نام بودن معاملات بیت‌کوین یک مزیت برای مجرمان است، در دسترس بودن عمومی داده‌ها یک مزیت کلیدی برای محققان به شمار می‌رود.

هدف پژوهش حاضر ارزیابی عملکرد الگوریتم‌های یادگیری ماشین و یادگیری عمیق در تشخیص پولشویی در تراکنش‌های بیت‌کوین است. برای این منظور از شش الگوریتم مختلف یادگیری ماشین، شامل شبکه عصبی مصنوعی (ANN)، جنگل تصادفی (RF)، K-نزدیکترین همسایه (KNN)، ماشین بردار پشتیبان (SVM)، و دو الگوریتم یادگیری عمیق شامل شبکه باور عمیق (DBN) و حافظه کوتاه‌مدت بلند (LSTM) استفاده شده است. ادامه مقاله شامل روش پژوهش، یافته‌ها، و در نهایت بحث و نتیجه‌گیری است.

مبانی نظری و پیشینه پژوهش

با تمرکز فزاینده بر فعالیت های غیرقانونی، ادبیات دانشگاهی بر ارائه طیف گسترده‌ای از سیستم‌های تشخیص خودکار برای شناسایی چنین فعالیت‌های غیرقانونی تاکید کرده است (بادر و کرچمار[1]، ۲۰۱۸؛ باتاگلیا[2] و همکاران، ۲۰۱۸؛ چانگ[3] و همکاران، ۲۰۰۸؛ گپ[4]، ۲۰۱۶؛ گپ[5] و همکاران، ۲۰۱۸؛ گپ، ۲۰۱۵؛ خالد[6] و همکاران، ۲۰۱۸؛

[1] Baader and Krcmar
[2] Battaglia
[3] Chang
[4] Gepp
[5] Gepp
[6] Khaled

نگای[1] و همکاران، ۲۰۱۱؛ پرولز[2]، ۲۰۱۱؛ فوآ[3] و همکاران، ۲۰۱۰؛ راوندا[4] و همکاران، ۲۰۱۵؛ ساهین[5]، ۲۰۱۳؛ سینگ و بست[6]، ۲۰۱۹؛ سونگ[7] و همکاران، ۲۰۱۴؛ ون ولاسلر[8] و همکاران، ۲۰۱۷؛ وج[9] و همکاران، ۲۰۱۷). طبق گفته نگای و همکاران (۲۰۱۱)، اگرچه کاربرد تکنیک‌های داده‌یابی به سمت کشف تقلب گسترش یافته است، اما فقدان تحقیق جامع در مورد کلاهبرداری بانکی، پولشویی و کلاهبرداری در اوراق بهادار و کالاها وجود دارد. از آن زمان، محققان هر دو روش یادگیری ماشینی و آماری سنتی را برای شناسایی پولشویی بررسی کرده‌اند (ایروین[10] و همکاران، ۲۰۱۲؛ بیداباد[11]، ۲۰۱۷؛ چانگ[12] و همکاران، ۲۰۰۸؛ دنگ[13] و همکاران، ۲۰۰۹؛ درزوسکی[14] و همکاران، ۲۰۱۲؛ کولادون و ریموندی[15]، ۲۰۱۷؛ گائو و یه[16]، ۲۰۰۷؛ گیلمور[17]، ۲۰۱۷؛ جو و ژنگ[18]، ۲۰۰۹؛ نگای[19] و همکاران، ۲۰۱۱؛ پرولز[20]، ۲۰۱۱؛ ریگان[21] و همکاران، ۲۰۱۷؛ ساواژ[22] و همکاران، ۲۰۱۶؛ ترنر و ایروین[23]، ۲۰۱۸؛ آنگر[24] و همکاران، ۲۰۱۱؛ وانگ[25] و همکاران، ۲۰۰۷؛ زدانوویچ[26]، ۲۰۰۴؛ زدانوویچ، ۲۰۰۹؛ ژانگ[27] و همکاران، ۲۰۰۳؛ گائو[28]، ۲۰۰۹، و لو[29] و همکاران (۲۰۲۳)).

ژانگ و همکاران (۲۰۰۳) با استفاده از روش کشف لینک بر اساس تجزیه و تحلیل همبستگی (LDCA[30]) پیشنهاد کرد که امکان ارتباط پولشویی با همبستگی بین الگوهای تراکنش مالی دو شخص وجود دارد. زدانوویچ (۲۰۰۴) و زدانوویچ (۲۰۰۹) استفاده از تجزیه و تحلیل آماری را برای نظارت و تشخیص پولشویی مبتنی بر

[1] Ngai
[2] Perols
[3] Phua
[4] Ravenda
[5] Sahin
[6] Singh and Best
[7] Song
[8] Van Vlasselaer
[9] Wedge
[10] Irwin
[11] Bidabad
[12] Chang
[13] Deng
[14] Drezewski
[15] Colladon and Remondi
[16] Gao and Ye
[17] Gilmour
[18] Ju and Zheng
[19] Ngai
[20] Perols
[21] Regan
[22] Savage
[23] Turner and Irwin
[24] Unger
[25] Wang
[26] Zdanowicz
[27] Zhang
[28] Gao
[29] Lo
[30] Link Detection using Correlation Analysis

معاملات پیشنهاد کرد. این پژوهش شواهد تجربی پولشویی مبتنی بر معاملات را در ادبیات دانشگاهی و حرفه‌ای گسترش داد. بر اساس این پژوهش، کشف دستکاری در قیمت معاملات با شناسایی ناهنجاری‌ها در داده‌های معاملات می‌تواند به شناسایی پولشویی وجوه ناشی از فعالیت‌هایی مانند تامین مالی اقدامات تروریستی، فرار مالیاتی، تخلیه کالا، مخفی کردن کمیسیون غیرقانونی و غیره با تمرکز بر کشور، منطقه گمرکی، محصول و ویژگی های ریسک قیمت معامله کمک کند.

چانگ و همکاران (۲۰۰۸) استفاده از مجموعه‌ای از روش‌های هماهنگ مبتنی بر شناسایی ویژگی‌های کلیدی در تراکنش‌های الکترونیکی را برای شناسایی تراکنش‌های غیرقانونی برجسته کردند. نویسندگان با مطالعه رابطه بین ویژگی‌های کلیدی و حساب‌ها در طول زمان توانستند تراکنش‌ها و حساب‌هایی را که رفتارهای مشکوک نشان می‌دهند شناسایی کنند. به طور مشابه، دنگ و همکاران (۲۰۰۹) یک رویکرد آماری را برای شناسایی موارد پولشویی ارائه کرد. این مطالعه با انگیزه نیاز به شناسایی و اولویت بندی تراکنش های مشکوک مربوطه در میان حجم زیادی از تراکنش های مالی که به صورت روزانه اتفاق می افتد، انجام شد. ایده این بود که به محققان کمک کند توجه خود را متمرکز کنند و منابع را به حساب‌هایی که ماهیت آنها مشکوک هستند هدایت کنند و با کمترین زمان و تلاش، شناسایی پولشویی را بهبود بخشند. نویسندگان دریافتند که این مدل از تقریب‌های تصادفی در تشخیص تراکنش‌های پولشویی بهتر عمل می‌کند.

ساویج و همکاران (۲۰۱۶) سیستمی را برای تشخیص فعالیت های مشکوک مالی از طریق استفاده از ترکیب تحلیل شبکه و یادگیری نظارت‌شده ارائه کردند. نویسندگان این مقاله با استفاده از این سیستم بر روی داده های دنیای واقعی دریافتند که این سیستم قادر به شناسایی فعالیت های مشکوک با نرخ پایین مثبت کاذب است.

به طور کلی، ادبیات بر تلاش برای شناسایی پولشویی که از طریق استفاده از املاک و مستغلات، تجارت بین‌المللی و کالاهای قابل حمل با ارزش بالا انجام می‌شود، تمرکز کرده است. فروردا[1] و همکاران (۲۰۱۳)، زدانوویچ (۲۰۰۹) و آنگر (۲۰۱۳) توجه خود را به سمت کشف پولشویی که از طریق املاک و مستغلات و تجارت انجام می‌شود معطوف کردند. ترنر و ایروین (۲۰۱۸) فرصت‌هایی را یافتند که نوآوری‌های فناورانه مانند بیت‌کوین برای پولشویی فراهم می‌کردند و راه‌هایی برای شناسایی آن پیشنهاد کردند. گیلمور (۲۰۱۷) استفاده از کالاهای قابل حمل با ارزش بالا را که برای پولشویی در بریتانیا و خارج از کشور استفاده می‌شود، مورد مطالعه قرار داد. بیداباد (۲۰۱۷) مکانیسم هایی را برای شناسایی پولشویی که از طریق استفاده از تراکنش‌های بانکی انجام می‌شود، پیشنهاد کرد. در نهایت، مقاله لو[2] و همکاران (۲۰۲۳) یک چارچوب شبکه عصبی مبتنی بر گراف (GNN[3]) و برخی دیگر از روش‌های یادگیری ماشین را برای تشخیص معاملات غیرقانونی برای مبارزه با پولشویی پیشنهاد کردند.

[1] Ferwerda
[2] Lo
[3] Graph-Based Neural Network

روش پژوهش

این مطالعه از روش‌های کمی استفاده نموده و از یادگیری ماشین برای تجزیه و تحلیل داده‌ها استفاده می‌کند.هدف اصلی پژوهش تشخیص معاملات مشکوک به منظور مدیریت ریسک اعتباری در فرآیند مدیریت ریسک مالی با استفاده از فناوری بلاکچین می باشد.

مجموعه داده‌های مورد استفاده جهت تشخیص پولشویی

تشخیص پولشویی با استفاده از مجموعه داده الیپتیک شامل شناسایی و تجزیه و تحلیل تراکنش های مشکوکی است که ممکن است شامل تبدیل یا انتقال وجوه غیرقانونی از طریق استفاده از ارزهای دیجیتال مانند بیت کوین باشد. الیپتیک[1] یک شرکت خدمات اطلاعاتی ارزهای دیجیتال است که بر حفاظت از اکوسیستم ارزهای دیجیتال در برابر فعالیت‌های مجرمانه متمرکز است. مجموعه داده الیپتیک، یک شبکه گراف از تراکنش‌های بیت‌کوین همراه با ویژگی‌های استخراج شده از آن را تشکیل می‌دهد. شرکت الیپتیک این مجموعه داده را به صورت عمومی به اشتراک گذاشته است. این مجموعه بزرگترین مجموعه داده تراکنش برچسب‌دار جهان را تشکیل می‌دهد که به صورت عمومی برای بیت‌کوین در دسترس است.مجموعه داده‌های الیپتیک تراکنش‌های بیت‌کوین را به اشخاص حقیقی متعلق به دسته‌های قانونی (صرافی‌ها، ارائه دهندگان کیف پول، استخراج کنندگان، خدمات قانونی و غیره) در مقابل غیرقانونی (کلاهبرداری، بدافزار، سازمان‌های تروریستی، باج افزار، و غیره) برچسب‌گذاری می‌کند. با استفاده از داده‌های خام بیت‌کوین، نموداری ساخته و برچسب‌گذاری شده است که در آن گره‌ها معاملات و یال‌ها جریان ارز بیت‌کوین را نشان می‌دهند که از یک گره به گره بعدی می‌رود. اگر نهادی که تراکنش را آغاز می‌کند به یک دسته غیرقانونی تعلق داشته باشد، یک تراکنش قانونی (و در غیر این صورت غیرقانونی) تلقی می‌شود. نکته مهم این است که همه ویژگی‌ها تنها با استفاده از اطلاعات در دسترس عموم ساخته می‌شوند.

در این مجموعه داده ۲۰۳۷۶۹ گره تراکنش و ۲۳۴۳۵۵ جریان پرداخت نشان داده شده با یال جهت‌دار وجود دارد. در مجموعه داده‌های الیپتیک، دو درصد (۴۵۴۵) معامله دارای برچسب "غیرقانونی" هستند. بیست و یک درصد (۴۲۰۱۹) از معاملات نیز دارای برچسب "قانونی" هستند. سایر تراکنش‌های باقیمانده در وضعیت "نامعین" قرار دارند، اما دارای مجموعه ویژگی‌های یکسانی هستند.

یک بازه زمانی برای هر گره ثبت شده که نشان دهنده تخمین زمانی است که تراکنش توسط شبکه بیت‌کوین تایید می‌شود. ۴۹ مرحله زمانی متمایز در این مجموعه داده وجود دارد که هرکدام به طور مساوی حدود دو هفته از بازه زمانی قبلی فاصله دارند. هر بازه زمانی شامل مجموعه‌ای از تراکنش‌هایی است که در طول سه ساعت در بلاک چین ظاهر می‌شوند. هیچ یالی برای اتصال بازه‌های مختلف زمانی وجود ندارد. واضح است که گره‌ها در یک بازه زمانی خاص دارای زمان‌های بسیار نزدیک به یکدیگر هستند، بنابراین هر یک از آنها را می‌توان به‌عنوان یک تصویر آنی در زمان در نظر گرفت. در این پژوهش از داده‌های مربوط به ۲۰ بازه زمانی آخر مجموعه داده (مربوط

[1] Elliptic

به جدیدترین ۴۰ هفته موجود در داده‌ها) استفاده شده است. نمونه مورد استفاده در این پژوهش داده‌های سال ۲۰۲۱ را پوشش می‌دهد.

متغیرهای مدل

هر گره دارای ۱۶۶ ویژگی است. اولین ۹۴ ویژگی نشان دهنده اطلاعات محلی در مورد تراکنش است - از جمله بازه زمانی، تعداد ورودی/خروجی‌ها، کارمزد تراکنش، حجم خروجی و آماره‌های محاسبه شده مانند میانگین بیت‌کوین دریافتی (صرف شده) توسط ورودی ها/خروجی ها و میانگین تعداد ورودی‌ها (خروجی ها) و تراکنش‌های مرتبط با ورودی/خروجی ها. ۷۲ ویژگی باقیمانده، با جمع‌آوری اطلاعات مربوط به یک گره قبل یا بعد از تراکنش مورد نظر به دست می‌آیند - مانند حداکثر، حداقل، انحراف‌معیار و ضرایب همبستگی تراکنش‌های همسایه برای داده‌های اطلاعاتی مختلف مانند کارمزد معامله و غیره).

در این پژوهش از الگوریتم مختلف یادگیری ماشین، شامل شبکه عصبی مصنوعی (ANN)، جنگل تصادفی (RF)، K-نزدیکترین همسایه (KNN)، ماشین بردار پشتیبان (SVM)، و دو الگوریتم یادگیری عمیق شامل شبکه باور عمیق (DBN) و حافظه کوتاه‌مدت بلند (LSTM) استفاده شده است. متغیرهای ورودی مدل‌های یادگیری ماشین مورد استفاده شامل ۱۶۶ ویژگی اشاره شده است. خروجی مدل‌های یادگیری ماشین نیز دسته‌بندی نوع معامله (قانونی، غیرقانونی، و نامعین) است که عملکرد الگوریتم ها با ماتریس درهم ریختگی مورد ارزیابی قرار گرفته اند.

روش‌های یادگیری ماشین

ماشین بردار پشتیبان

ماشین‌های بردار پشتیبان یا SVM[1] یکی از محبوب‌ترین الگوریتم‌های یادگیری ماشین هستند که برای دسته‌بندی و رگرسیون استفاده می‌شوند. در دسته‌بندی چندکلاسه، SVMها برای دسته‌بندی نمونه‌ها به یکی از چند کلاس مختلف استفاده می‌شوند. در این حالت، SVMها به عنوان یک مدل به کار می‌روند که با استفاده از مجموعه‌ای از ویژگی‌ها، قادر به پیش‌بینی کلاس یک نمونه جدید هستند.

فرمول ریاضی SVM برای دسته‌بندی چندکلاسه به شکل زیر است:

فرض کنید N نمونه با مجموعه‌ای از d ویژگی و C کلاس وجود داشته باشد. هدف پیدا کردن یک ابرصفحه جداساز است که نمونه‌ها را به C کلاس مختلف تقسیم کند. ابرصفحه جداساز باینری به روش زیر تعریف می‌شود:

$$w * x + b = 0$$

در اینجا، w برداری است که عمود بر ابرصفحه جداساز است و x برداری است که نماینده یک نمونه است.

[1] Support Vector Machines

برای آموزش مدل SVM برای دسته‌بندی، باید مسئله بهینه‌سازی زیر را حل کنیم:

$$minimize: \frac{1}{2} * \ ||w||^2$$
$$subject\ to:$$
$$y_i * \ (w * x_i + \ b) \geq \ 1 \ for \ i = 1,2, \dots, N$$

در اینجا، ||w|| نرم بردار وزن است، y_i برچسب (کلاس) نمونه i ، و x_i بردار ویژگی نمونه i ام است.
مسئله بهینه‌سازی با استفاده از ضرایب چندجمله‌ای لاگرانژ حل می‌شود. شکل دوگان مسئله بهینه‌سازی به شکل زیر است:

$$maximize: \sum_i \alpha_i - \frac{1}{2} * \sum_{i,j} y_i * \ y_j * \ \alpha_i * \ \alpha_j * \ x_i * \ x_j$$
$$subject\ to: \sum_i \ y_i * \ \alpha_i = \ 0 \ and \ \alpha_i \geq \ 0 \ for \ i = 1,2, \dots, N$$

در اینجا، α_i ضرایب چندجمله‌ای لاگرانژ هستند که برای هر نمونه محاسبه می‌شوند و از آن‌ها برای پیدا کردن وزن و ابرصفحه جداساز استفاده می‌شود.
پس از حل مسئله بهینه‌سازی، بردار وزن برابر است با:

$$w \ = \sum_i \alpha_i * \ y_i * \ x_i$$

و ابرصفحه جداساز به شکل زیر تعریف می‌شود:

$$w * x + b \ = \ 0$$

در صورتی α_i که برای یک نمونه برابر با صفر باشد، آن نمونه در نتیجه دسته‌بندی تاثیر داده نمی‌شود و در نتیجه نقطه بر روی ابرصفحه جداساز وجود ندارد. به علاوه، اگر مقدار α_i برای هر نمونه برابر با صفر باشد، این به این معناست که آن نقطه بر روی بردار پشتیبان قرار دارد. در نهایت، برای دسته‌بندی نمونه‌های جدید با استفاده از مدل SVM، بردار ویژگی آن‌ها محاسبه می‌شود و سپس با استفاده از ابرصفحه جداساز، نمونه به یکی از کلاس‌ها تخصیص داده می‌شود.

جنگل تصادفی[1]

جنگل تصادفی یک الگوریتم یادگیری ماشینی محبوب است که برای طبقه بندی، رگرسیون و سایر وظایف استفاده می‌شود. این روش یک روش یادگیری مجموعه‌ای[2] است که چندین درخت تصمیم را برای بهبود دقت و کاهش بیش‌برازش[3] درختان جداگانه ترکیب می کند.

[1] Random Forest
[2] Ensemble
[3] Overfitting

نمای کلی الگوریتم به صورت زیر است:

۱) یک زیر مجموعه تصادفی از داده های آموزشی با جایگزینی انتخاب می‌شود.
۲) یک درخت تصمیم بر روی داده‌های انتخاب شده با تقسیم بازگشتی داده‌ها به زیر مجموعه‌های کوچکتر بر اساس مقادیر ویژگی ها رشد می‌کند.
۳) در هر گره، یک زیرمجموعه تصادفی از ویژگی‌ها برای تقسیم در نظر گرفته می‌شود.
۴) درخت تا زمانی رشد می کند که یک معیار توقف مانند حداکثر عمق یا حداقل تعداد نمونه در یک گره برگ برآورده شود.
۵) مراحل ۱-۴ چندین بار تکرار می شود تا جنگلی از درختان تصمیم ایجاد شود.
۶) برای هر ورودی جدید، جنگل طبقه را بر اساس رای اکثریت پیش‌بینی‌های تک درختان پیش‌بینی می‌کند.

فرمول بندی ریاضی مسئله به صورت زیر است:

انتخاب تصادفی زیر مجموعه:

برای ایجاد یک زیرمجموعه تصادفی از داده‌های آموزشی، ابتدا n نمونه از مجموعه داده را با جایگزینی به صورت تصادفی انتخاب می‌کنیم که n اندازه زیر مجموعه است.

درخت تصمیم:

درخت تصمیم یک درخت باینری است که در آن هر گره یک تصمیم را بر اساس یک مقدار ویژگی نشان می‌دهد. درخت به صورت بازگشتی با تقسیم داده‌ها به زیر مجموعه‌های کوچکتر بر اساس مقادیر ویژگی ها رشد می‌کند. فرض کنید T یک درخت تصمیم باشد. هر گره i از درخت با یک ویژگی f_i و یک آستانه t_i مرتبط است، به طوری که درخت داده ها را به دو زیر مجموعه تقسیم می کند:

$$S_{left(i)} = \{x \in S : x[f_i] \leq t_i\}$$
$$S_{right(i)} = \{x \in S : x[f_i] > t_i\}$$

در اینجا $x[f_i]$ مقدار ویژگی f_i برای نمونه x است.

انتخاب ویژگی:

در هر گره i درخت، یک زیرمجموعه تصادفی از k ویژگی برای تقسیم انتخاب می شود. فرض کنید F مجموعه‌ای از تمام ویژگی های مجموعه داده باشد، و K زیرمجموعه ای از ویژگی ها باشد که به طور تصادفی انتخاب شده اند.

$$K = \{f_1, f_2, \ldots, f_k\}$$

که در آن f_i یک ویژگی تصادفی انتخاب شده از F است.

معیار توقف: درخت به صورت بازگشتی رشد می کند تا زمانی که یک معیار توقف برآورده شود. این می تواند حداکثر عمق درخت یا حداقل تعداد نمونه در یک گره برگ باشد.

پیش‌بینی: برای پیش‌بینی کلاس ورودی x جدید، جنگل اکثریت پیش‌بینی‌های درختان را در نظر می‌گیرد. فرض کنید T جنگل درختان تصمیم باشد و y کلاس پیش بینی شده x باشد.

$$y = argmax_c \sum_i T_{i(x)} = c$$

در اینجا $T_{i(x)}$ پیش‌بینی درخت i برای ورودی x است و c یک برچسب کلاس است.

مزایای جنگل تصادفی:

- ✓ جنگل تصادفی یک الگوریتم قدرتمند است که می‌تواند داده‌های طبقه‌ای و پیوسته را مدیریت کند و در برابر داده‌های نویزی[1] استوار است.
- ✓ جنگل تصادفی می تواند فضاهای ویژگی با ابعاد بالا را مدیریت کند و نسبت به درخت‌های تصمیم منفرد کمتر مستعد بیش‌برازش است.
- ✓ جنگل تصادفی به راحتی اجرا می شود و می‌توان آن را به سرعت در مجموعه داده‌های بزرگ آموزش داد.

معایب:

- ✓ جنگل تصادفی می تواند برای مجموعه داده های بزرگ با ویژگی‌های بسیار کند و حافظه‌بر باشد.
- ✓ الگوریتم جنگل تصادفی ممکن است روی مجموعه داده های نامتعادل با ویژگی‌های نامربوط عملکرد خوبی نداشته باشد.
- ✓ تفسیر مدل به سادگی مدل‌های خطی نیست.

شبکه عصبی مصنوعی (ANN[2])

یک شبکه عصبی مصنوعی برای طبقه بندی متشکل از چندین لایه از نورون های به هم پیوسته است. لایه ورودی داده های ورودی را دریافت می کند و سپس از یک یا چند لایه پنهان قبل از تولید خروجی نهایی عبور می کند. هر نورون در شبکه یک مجموع وزنی از ورودی ها را دریافت می کند و یک تابع فعال سازی را به نتیجه اعمال می کند تا خروجی خود را تولید کند.

فرمول ریاضی برای یک نورون منفرد به شرح زیر است:

$$z = w_1 * x_1 + w_2 * x_2 + \dots + w_n * x_n + b$$
$$y = f(z)$$

که در آن x_n ، ... ، x_2 ، x_1 ورودی‌های نورون هستند، w_n ، ... ، w_2 ، w_1 وزن‌های مربوطه، b جمله بایاس، z مجموع وزنی ورودی‌ها و بایاس، f(z) تابع فعال سازی و y خروجی نورون است.

توابع فعال سازی رایج مورد استفاده در شبکه های عصبی شامل سیگموئید، ReLU و tanh است. انتخاب تابع فعال سازی به کاربرد خاص و ویژگی های داده های مورد تجزیه و تحلیل بستگی دارد.

بعد از پردازش توسط هر نورون، خروجی یک نورون به عنوان ورودی به لایه بعدی نورون منتقل می شود تا خروجی نهایی تولید شود. وزن شبکه معمولاً از طریق فرآیندی به نام پس‌انتشار[3] خطا محاسبه می‌شود که شامل

[1] Noisy
[2] Artificial Neural Network
[3] Back Propagation

به حداقل رساندن یک تابع زیان[1] است که تفاوت بین خروجی پیش‌بینی‌شده و خروجی واقعی را اندازه‌گیری می‌کند.
فرمول ریاضی تابع زیان احتمال لگاریتمی[2] (به عنوان یک تابع زیان پرکاربرد) به شرح زیر است:

$$L(y، y_{hat}) = -[y * \log(y_{hat}) + (1 - y) * \log(1 - y_{hat})]$$

که در آن y برچسب واقعی، y_{hat} خروجی پیش‌بینی‌شده شبکه و log لگاریتم طبیعی است.

در طول آموزش، وزن‌های شبکه برای به حداقل رساندن تابع زیان با استفاده از یک الگوریتم بهینه‌سازی مانند نزول گرادیان تصادفی (SGD[3]) تنظیم می‌شوند. هدف الگوریتم بهینه‌سازی یافتن مقادیر وزن‌هایی است که تابع زیان را به حداقل می‌رسانند و پیش‌بینی‌های دقیقی تولید می‌کنند.

به طور خلاصه، یک شبکه عصبی مصنوعی برای طبقه‌بندی شامل چندین لایه از نورون‌های به هم پیوسته است که هر نورون یک تابع فعال‌سازی را برای مجموع وزنی ورودی‌های خود اعمال می‌کند. وزن‌های شبکه از طریق پس‌انتشار یاد گرفته شده و با استفاده از یک الگوریتم بهینه‌سازی بهینه می‌شوند. شبکه یک خروجی نهایی تولید می کند که نشان دهنده برچسب کلاس پیش بینی شده برای داده های ورودی است.

شبکه حافظه کوتاه‌مدت بلند (LSTM[4])

شبکه حافظه کوتاه‌مدت بلند (LSTM) نوعی شبکه عصبی بازگشتی (RNN[5]) است که قادر است وابستگی‌های بلندمدت را در داده‌های متوالی ثبت کند. در این بخش، ما بر روی استفاده از LSTM برای کارهای طبقه بندی تمرکز خواهیم کرد.

فرض کنید مجموعه ای از داده های ورودی $x_1, x_2, \dots, x_m$ داریم که در آن هر ورودی با تعداد ثابتی از ویژگی نمایش داده می شود. ما می خواهیم این ورودی ها را در یکی از چندین دسته طبقه بندی کنیم.

اولین قدم این است که داده های ورودی نرمال سازی شوند تا اطمینان حاصل شود که هر ویژگی مقیاس مشابهی دارد. این مهم است زیرا برخی از ویژگی‌ها ممکن است مقادیر عددی بزرگتری نسبت به سایرین داشته باشند، که می‌تواند منجر به وزن‌های دارای سوگیری شود.

سپس، می‌توانیم ورودی‌های نرمال‌شده را به یک لایه LSTM وارد کنیم. لایه LSTM شامل یک سری سلول LSTM است که شامل سه دروازه (ورودی، فراموشی و خروجی) است که جریان اطلاعات را از طریق سلول کنترل می‌کند.

[1] Loss Function
[2] Log-Probability
[3] Stochastic Gradient Descend
[4] Long Short-Term Memory
[5] Recurrent Neural Network

دروازه ورودی i_t تعیین می کند که چه مقدار از ویژگی ورودی جدید x_t در حالت سلولی C_t گنجانده شده است. دروازه فراموشی f_t تعیین می‌کند که چه مقدار از حالت سلول قبلی C_{t-1} حفظ می شود. دروازه خروجی o_t تعیین می کند که چه مقدار از حالت سلول به روز شده C_t برای تولید خروجی لایه LSTM h_t استفاده می شود. محاسبات برای سلول LSTM به شرح زیر است:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$
$$h_t = o_t \odot \tanh(C_t)$$

جایی که W_{xi}، W_{xf}، W_{xo}، W_{xc}، W_{hi}، W_{hf}، W_{ho}، W_{hc} ماتریس‌های وزن، b_i، b_f، b_o، b_c بردارهای بایاس، σ تابع فعال‌سازی سیگموئید، tanh فعال‌سازی مماس هذلولی، $\odot$ عملگر ضرب عنصر به عنصر است، و x_{t-1} و h_t به ترتیب بردار ویژگی ورودی و خروجی سلول LSTM هستند.

بعد از لایه LSTM می توان یک یا چند لایه کاملا متصل[1] اضافه و خروجی نهایی شبکه را تولید کرد. لایه های کاملا متصل خروجی لایه LSTM را به عنوان ورودی می گیرند و یک تبدیل خطی و به دنبال آن یک تابع فعال سازی غیرخطی (مانند ReLU یا softmax) برای تولید برچسب پیش بینی شده کلاس اعمال می‌کنند.

شبکه باور عمیق[2] (DBN)

شبکه باور عمیق (DBN) نوعی شبکه عصبی مصنوعی است که برای یادگیری بدون نظارت، استخراج ویژگی و طبقه‌بندی استفاده می‌شود. DBN ها از چندین لایه مدل های احتمالی تشکیل شده اند که هر لایه یاد می گیرد تا داده های ورودی را در سطح متفاوتی از انتزاع نمایش دهد. لایه اول ویژگی‌های سطح پایین را یاد می‌گیرد، در حالی که لایه های بعدی ویژگی های پیچیده تر را یاد می‌گیرند.

DBNها از دو نوع لایه اصلی تشکیل شده‌اند: لایه قابل مشاهده که لایه ورودی است و لایه های پنهان. لایه‌های پنهان از ماشین‌های محدود بولتزمن[3] (RBM) تشکیل شده‌اند که مدل‌های گراف بدون جهت هستند که برای یادگیری توزیع احتمال داده‌های ورودی آموزش دیده‌اند.

به صورت ریاضی یک DBN را می‌توان به صورت زیر توصیف کرد:

فرض کنید $X = \{x_1, x_2, \dots, x_n\}$ مجموعه‌ای از داده‌های ورودی باشد که در آن هر xi بردار ویژگی‌ها است. لایه قابل مشاهده DBN از داده های ورودی X تشکیل شده است و لایه‌های پنهان به صورت H_1، H_2، ... ، H_k نمایش داده می‌شوند که k تعداد لایه‌های پنهان است.

[1] Fully Connected
[2] Deep Belief Network
[3] Restricted Boltzmann Machine

هر لایه پنهان H_i از m نورون تشکیل شده است که m تعداد واحدهای پنهان است. فعال‌سازی یک واحد پنهان h_j در لایه H_i به صورت زیر ارائه می‌شود:

$h_j = 1$ اگر $P(h_j = 1|v) > r$، و در غیر این صورت برابر صفر

که در آن $P(h_j = 1|v)$ احتمال فعال‌شدن واحد پنهان h_j با توجه به ورودی قابل مشاهده v است، و r یک عدد تصادفی است که از یک توزیع یکنواخت گرفته شده است.

RBM برای یادگیری توزیع احتمال مشترک بین لایه مرئی و لایه پنهان آموزش دیده است. تابع انرژی RBM به صورت زیر تعریف می شود:

$$E(v, h) = -b'v - c'h - h'Wv$$

که در آن b و c به ترتیب بایاس‌های لایه های مرئی و پنهان هستند و W ماتریس وزن بین لایه های مرئی و پنهان است.

احتمال یک ورودی قابل مشاهده v و یک فعال‌سازی پنهان h به صورت زیر داده می شود:

$$P(v, h) = \exp(-E(v, h)) / Z$$

که در آن Z یک ثابت نرمال کننده است که تضمین می‌کند مجموع همه احتمالات برابر با ۱ باشد.

DBN با استفاده از یک فرآیند دو مرحله‌ای به نام پیش‌آموزش[1] و تنظیم دقیق آموزش داده می‌شود. در مرحله پیش‌آموزش، هر RBM برای یادگیری ویژگی‌های داده‌ها آموزش داده می‌شود. در مرحله تنظیم دقیق، DBN با استفاده از پس انتشار آموزش داده می‌شود تا خطای طبقه‌بندی را به حداقل برساند.

هنگامی که DBN آموزش داده شد، می‌توان از آن برای طبقه‌بندی با دادن داده‌های ورودی از طریق شبکه و محاسبه احتمالات خروجی برای هر کلاس با استفاده از یک تابع فعال‌سازی سافتمکس[2] استفاده کرد.

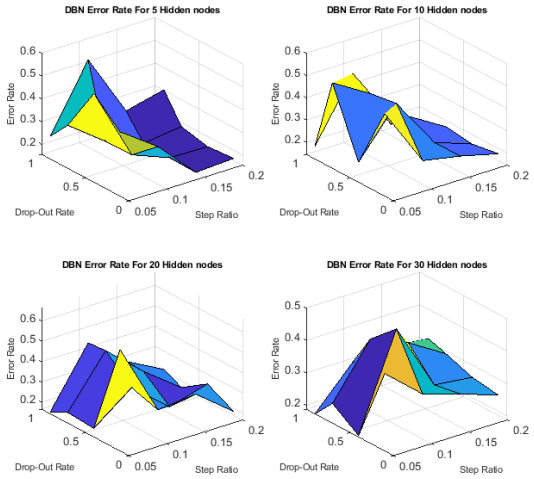
به طور خلاصه، DBN یک الگوریتم یادگیری ماشینی قدرتمند است که می‌تواند برای یادگیری بدون نظارت، استخراج ویژگی و وظایف طبقه‌بندی استفاده شود. مراحل کلی کار آن شامل استفاده از RBM برای مدل‌سازی توزیع احتمال مشترک بین لایه‌های مرئی و پنهان است و با استفاده از یک فرآیند دو مرحله‌ای به نام پیش‌آموزش و تنظیم دقیق آموزش داده می‌شود.

یافته‌ها

در این بخش به ارائه نتایج پیاده‌سازی هریک از مدل‌های مورد استفاده در این پژوهش می‌پردازیم. ابتدا نحوه تنظیم پارامترهای الگوریتم‌ها پرداخته و سپس نتایج پیاده‌سازی هر الگوریتم ارائه می‌شود. تنظیم پارامترهای پنج الگوریتم ANN, RF, KNN, SVM, LSTM به صورت خودکار توسط توابع برازش مدل در متلب انجام شده است. با این حال سه پارامتر مهم الگوریتم DBN شامل تعداد گره لایه پنهان در ۴ سطح از ۵ تا ۳۰ گره، نرخ خروج

[1] Pretraining
[2] softmax

(Drop-Out Rate) به ازای ۴ سطح از ۰.۱ تا ۰.۹ و اندازه گام (Step Ratio) در ۴ سطح از ۰.۰۵ تا ۰.۲ مورد آزمون قرار گرفته است. شکل ۱ مقادیر خطا را به ازای ترکیبات مختلف از پارامترها نشان می‌دهد. بر اساس این پیاده‌سازی‌های آزمایشی، بهترین پارامترها (متناظر با کمترین خطا) به صورت ۱۰ گره لایه پنهان، نرخ خروج (Drop-Out Rate) ۰.۹ و اندازه گام (Step Ratio) ۰.۲ مورد استفاده قرار گرفته است.



شکل ۱: خطای دسته‌بندی توسط الگوریتم شبکه باور عمیق با لایه‌ها و پارامترهای مختلف

به منظور بررسی کارایی الگوریتم‌ها از ماتریس و نمودار درهم‌ریختگی استفاده شده است. ماتریس درهم‌ریختگی ابزاری است که برای ارزیابی عملکرد یک مدل طبقه‌بندی بر روی یک مجموعه داده با برچسب‌های شناخته شده استفاده می شود. در این مورد، ماتریس سردرگمی برای یک مسئله طبقه بندی ۳ برچسبی است که در آن برچسب ها "قانونی" (برچسب ۰)، "ناشناخته" (برچسب ۱) و "غیر قانونی" (برچسب ۲) هستند. هر ردیف در ماتریس نشان دهنده کلاس واقعی نمونه ها و هر ستون نشان دهنده کلاس پیش بینی شده نمونه ها است. اعداد در سلول ها تعداد نمونه هایی را نشان می دهد که به هر جفت کلاس-پیش‌بینی تعلق دارند.

برای محاسبه صحت[1]، دقت[2]، پوشش[3] و امتیاز F1 برای هر کلاس در ماتریس درهم‌ریختگی طبقه‌بندی ۳ برچسبی، ابتدا باید معیارهای زیر را محاسبه کنیم:

- مثبت صحیح (TP) : تعداد نمونه هایی که به درستی متعلق به یک کلاس مشخص طبقه‌بندی شده اند.

[1] Accuracy
[2] Precision
[3] Recal

- مثبت کاذب (FP): تعداد نمونه هایی که به اشتباه متعلق به یک کلاس مشخص طبقه‌بندی شده‌اند، در حالی که آنها واقعاً به کلاس دیگری تعلق دارند.
- منفی کاذب (FN): تعداد نمونه‌هایی که به اشتباه غیر متعلق به یک کلاس مشخص طبقه بندی شده‌اند، در صورتی که واقعاً به آن کلاس تعلق دارند.
- منفی صحیح (TN) : تعداد نمونه هایی که به درستی غیر متعلق به یک کلاس مشخص طبقه بندی شده اند.

بر این اساس، نحوه محاسبه هر معیار عملکرد به صورت زیر است:

صحت = (TP + TN) / (TP + TN + FP + FN)

دقت = TP / (TP + FP)

پوشش = TP / (TP + FN)

F1-Score = 2 * دقت * پوشش / (دقت + پوشش)

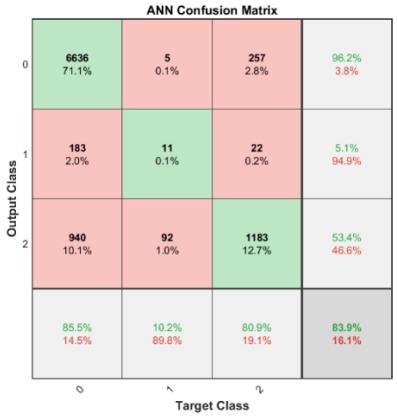
جدول ۱ نشان دهنده ماتریس درهم ریختگی برای طبقه بندی ۳ برچسبی با استفاده از مدل ANN است. سطرها نشان دهنده برچسب های واقعی و ستون ها نشان دهنده برچسب های پیش بینی شده هستند. این ماتریس تعداد نمونه هایی را نشان می دهد که در هر برچسب طبقه بندی شده اند.

به عنوان مثال، 6636 نمونه از کلاس " نامعین " وجود دارد که به درستی به عنوان چنین طبقه بندی شده اند، 5 مورد از کلاس "نامعین" که به اشتباه به عنوان " غیرقانونی" طبقه بندی شده اند، و ۳۷۰ نمونه از کلاس "نامعین" که به اشتباه به عنوان "قانونی" طبقه بندی شده اند.

با استفاده از این ماتریس، می‌توانیم معیارهای عملکرد مختلفی را برای هر برچسب، از جمله صحت، دقت، پوشش و امتیاز F1 محاسبه کنیم (که در ادامه ارائه خواهد شد). با این حال، شایان ذکر است که این مدل برای برچسب "قانونی" و "نامعین" عملکرد خوبی دارد اما برای برچسب های "نامعین" و "غیر قانونی" عملکرد خوبی ندارد. دقت کلی خوب است، اما عملکرد ضعیف روی برچسب "غیر قانونی" نشان دهنده نیاز به داده های آموزشی بیشتر یا رویکردی متفاوت است. این ماتریس بر روی نمودار درهم‌ریختگی شکل ۲ با برچسب‌های 0 (برای نامعین)، 1 برای (غیرقانونی) و 2 (برای قانونی) نیز نشان داده شده است.

جدول ۱ : ماتریس درهم ریختگی برای طبقه بندی با استفاده از مدل ANN

	نامعین	غیرقانونی	قانونی
نامعین	6636	183	940
غیرقانونی	5	11	92
قانونی	370	22	1183

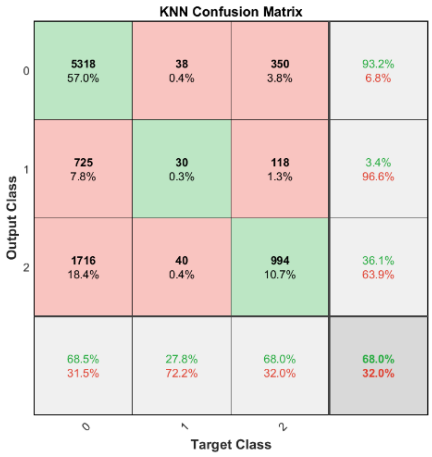


شکل ۲ : نمودار درهم ریختگی برای طبقه بندی با استفاده از مدل ANN

جدول ۲ نشان دهنده ماتریس درهم ریختگی برای طبقه بندی با استفاده از مدل KNN است. این ماتریس نشان می دهد 5318 نمونه از کلاس " نامعین " وجود دارد که به درستی به عنوان چنین طبقه بندی شده اند، 38 مورد از کلاس "نامعین" که به اشتباه به عنوان " غیرقانونی" طبقه بندی شده اند، و 350 نمونه از کلاس "نامعین" که به اشتباه به عنوان "قانونی" طبقه بندی شده اند.این مدل نیز برای برچسب "قانونی" و "نامعین" عملکرد خوبی دارد اما برای برچسب "غیر قانونی" عملکرد خوبی ندارد. دقت کلی خوب است، اما عملکرد ضعیف روی برچسب "غیر قانونی" نشان دهنده نیاز به داده های آموزشی بیشتر یا رویکردی متفاوت است. البته باید گفت که عملکرد این مدل برای برچسب "غیرقانونی" بهتر از مدل ANN است. این ماتریس بر روی نمودار درهم‌ریختگی شکل ۳ با برچسب‌های 0 (برای نامعین)، 1 برای (غیرقانونی) و 2 (برای قانونی) نیز نشان داده شده است.

جدول ۲ : ماتریس درهم ریختگی برای طبقه بندی با استفاده از مدل KNN

	نامعین	غیرقانونی	قانونی
نامعین	5318	725	1716
غیرقانونی	38	30	40
قانونی	350	118	994

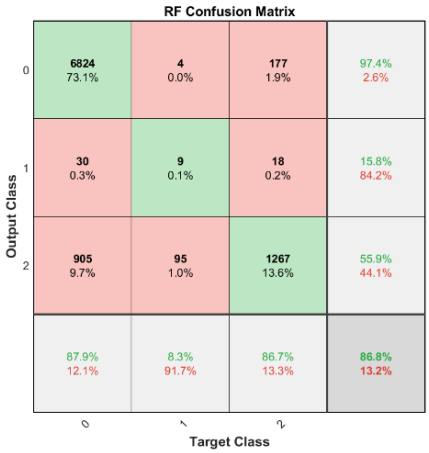


شکل ۳ : نمودار درهم ریختگی برای طبقه بندی با استفاده از مدل KNN

جدول ۳ نشان دهنده ماتریس درهم ریختگی برای طبقه بندی با استفاده از مدل RF است. این ماتریس نشان می دهد 6824 نمونه از کلاس "نامعین" وجود دارد که به درستی به عنوان چنین طبقه‌بندی شده‌اند، 4 مورد از کلاس "نامعین" که به اشتباه به عنوان "غیرقانونی" طبقه بندی شده باشد، و 177 نمونه از کلاس "نامعین" وجود دارد که به اشتباه به عنوان "قانونی" طبقه بندی شده اند. این مدل نیز برای برچسب "قانونی" و "نامعین" عملکرد خوبی دارد اما برای برچسب "غیر قانونی" عملکرد خوبی ندارد. دقت کلی خوب است، اما عملکرد ضعیف روی برچسب "غیر قانونی" نشان دهنده نیاز به داده های آموزشی بیشتر یا رویکردی متفاوت است. این ماتریس بر روی نمودار درهم‌ریختگی شکل ۴ با برچسب‌های 0 (برای نامعین)، 1 برای (غیرقانونی) و 2 (برای قانونی) نیز نشان داده شده است.

جدول ۳ : ماتریس درهم ریختگی برای طبقه بندی با استفاده از مدل RF

	نامعین	غیرقانونی	قانونی
نامعین	6824	30	905
غیرقانونی	4	9	95
قانونی	177	18	1267



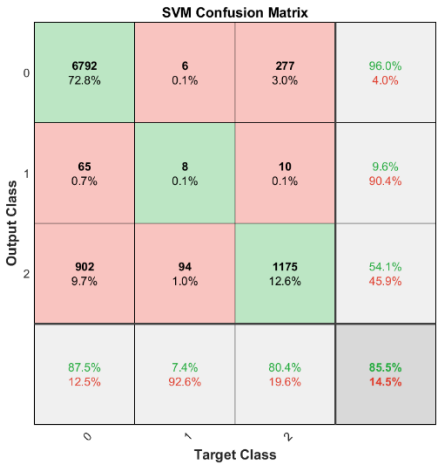
شکل ۴ : نمودار درهم ریختگی برای طبقه‌بندی با استفاده از مدل RF

جدول ۴ نشان دهنده ماتریس درهم ریختگی برای طبقه بندی با استفاده از مدل SVM است. این ماتریس نشان می دهد 6792 نمونه از کلاس "نامعین" وجود دارد که به درستی به عنوان چنین طبقه‌بندی شده‌اند، تنها ۶ مورد از کلاس "نامعین" وجود دارد که به اشتباه به عنوان "غیرقانونی" طبقه بندی شده باشد، و ۲۷۷ نمونه از کلاس "نامعین" وجود دارد که به اشتباه به عنوان "قانونی" طبقه بندی شده اند.

این مدل نیز برای برچسب "قانونی" و "نامعین" عملکرد خوبی دارد اما برای برچسب "غیر قانونی" عملکرد خوبی ندارد. دقت کلی خوب است، اما عملکرد ضعیف روی برچسب "غیر قانونی" نشان دهنده نیاز به داده های آموزشی بیشتر یا رویکردی متفاوت است. این ماتریس بر روی نمودار درهم‌ریختگی شکل ۵ با برچسب‌های 0 (برای نامعین)، 1 برای (غیرقانونی) و 2 (برای قانونی) نیز نشان داده شده است.

جدول ۴ : ماتریس درهم ریختگی برای طبقه بندی با استفاده از مدل SVM

	نامعین	غیرقانونی	قانونی
نامعین	6792	65	902
غیرقانونی	6	8	94
قانونی	277	10	1175



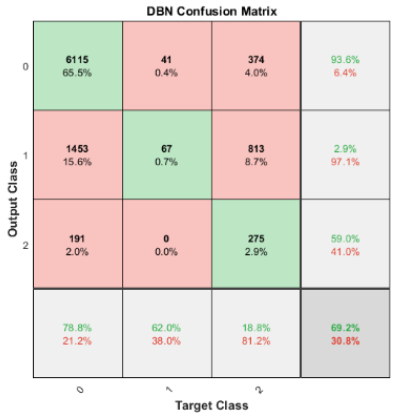
شکل ۵ : نمودار درهم ریختگی برای طبقه‌بندی با استفاده از مدل SVM

جدول ۵ نشان دهنده ماتریس درهم ریختگی برای طبقه بندی با استفاده از مدل DBN است. این ماتریس نشان می‌دهد ۶۱۱۵ نمونه از کلاس "نامعین" وجود دارد که به درستی به عنوان چنین طبقه‌بندی شده‌اند، ۴۱ مورد از کلاس "نامعین" وجود دارد که به اشتباه به عنوان "غیرقانونی" طبقه‌بندی شده است، و ۳۷۴ نمونه از کلاس "نامعین" وجود دارد که به اشتباه به عنوان "قانونی" طبقه بندی شده اند. توجه داشته باشید که این الگوریتم نیز ۶۷ معامله غیرقانونی را به درستی تشخیص داده و ۱۴۵۳ مورد غیرقانونی را نامعین تشخیص داده است.

این مدل نیز برای برچسب "نامعین" عملکرد خوبی دارد اما برای برچسب‌های "قانونی" و "غیر قانونی" عملکرد خوبی ندارد. دقت کلی خوب است، اما عملکرد ضعیف روی برچسب "قانونی" نشان دهنده نیاز به داده‌های آموزشی بیشتر یا رویکردی متفاوت است. این ماتریس بر روی نمودار درهم‌ریختگی شکل ۶ با برچسب‌های 0 (برای نامعین)، 1 برای (غیرقانونی) و 2 (برای قانونی) نیز نشان داده شده است.

جدول ۵ : ماتریس درهم ریختگی برای طبقه بندی با استفاده از مدل DBN

	نامعین	غیرقانونی	قانونی
نامعین	6115	1453	191
غیرقانونی	41	67	0
قانونی	374	813	275

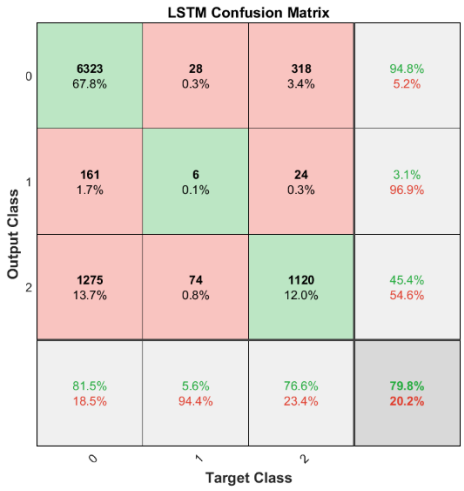


شکل ۶ : نمودار درهم ریختگی برای طبقه‌بندی با استفاده از مدل DBN

جدول ۶ نشان دهنده ماتریس درهم ریختگی برای طبقه بندی با استفاده از مدل LSTM است. این ماتریس نشان می‌دهد ۶۳۲۳ نمونه از کلاس "نامعین" وجود دارد که به درستی به عنوان چنین طبقه‌بندی شده‌اند، ۲۸ مورد از کلاس "نامعین" وجود دارد که به اشتباه به عنوان "غیرقانونی" طبقه‌بندی شده باشد، و ۳۱۸ نمونه از کلاس "نامعین" وجود دارد که به اشتباه به عنوان "قانونی" طبقه‌بندی شده‌اند. این الگوریتم نیز تنها ۶ معامله غیرقانونی را به درستی تشخیص داده و معاملات غیرقانونی را نامعین (۱۶۱ مورد) یا قانونی (۲۴ مورد) تشخیص داده است. این مدل نیز برای برچسب "نامعین" و "قانونی" عملکرد خوبی دارد اما برای برچسب "غیر قانونی" عملکرد خوبی ندارد. دقت کلی خوب است، اما عملکرد ضعیف روی برچسب "غیر قانونی" نشان دهنده نیاز به داده‌های آموزشی بیشتر یا رویکردی متفاوت است. این ماتریس بر روی نمودار درهم‌ریختگی شکل ۷ با برچسب‌های 0 (برای نامعین)، 1 برای (غیرقانونی) و 2 (برای قانونی) نیز نشان داده شده است.

جدول ۶ : ماتریس درهم ریختگی برای طبقه بندی با استفاده از مدل LSTM

	نامعین	غیرقانونی	قانونی
نامعین	6323	161	1275
غیرقانونی	28	6	74
قانونی	318	24	1120



شکل ۷ : نمودار درهم ریختگی برای طبقه‌بندی با استفاده از مدل LSTM

جدول ۷ معیارهای عملکرد مختلف را برای ۶ مدل یادگیری ماشین استفاده شده نشان می‌دهد. بر اساس این جدول، بیشترین صحت به ترتیب مربوط به مدلRF (با صحت 87 درصد)، DBN (با صحت 85 درصد)، و SVM (با صحت 85 درصد) است.

دقت‌ها به ترتیب مربوط به مدلRF (با صحت 56 درصد)، DBN (با دقت 53 درصد)، SVM (با دقت 53 درصد)، است. بیشترین پوشش به ترتیب مربوط به مدل RF (با پوشش 61 درصد)، ANN (با پوشش 59 درصد)، و SVM (با پوشش 58 درصد) است. در نهایت، بیشترین امتیاز F1 به ترتیب مربوط به مدل RF (با امتیاز F1 57 درصد)، SVM (با امتیاز F1 55 درصد)، و ANN (با امتیاز F1 ۵۴ درصد) است.

جدول ۷: معیارهای عملکرد مدل‌های مختلف

	دقت	پوشش	صحت	امتیاز F1
SVM	0.53	0.58	0.85	0.55
KNN	0.44	0.55	0.68	0.44
RF	0.56	0.61	0.87	0.57
ANN	0.52	0.59	0.84	0.54
LSTM	0.48	0.55	0.80	0.50
DBN	0.53	0.53	0.85	0.40

بحث و نتیجه‌گیری

هدف از این تحقیق ارزیابی عملکرد الگوریتم های یادگیری ماشین و یادگیری عمیق در تشخیص پولشویی در تراکنش های بیت کوین بود. نتایج به‌دست‌آمده نشان می‌دهد که RF، SVM و DBN بهترین عملکرد را در تشخیص تراکنش‌های مشکوک ارائه می‌کنند. سایر الگوریتم‌ها، ANN، LSTM، و KNN نیز عملکرد خوبی از خود نشان دادند، اما عملکرد آنها در مقایسه با جنگل تصادفی، SVM و DBNکمتر است.

یکی از دلایلی که جنگل تصادفی، SVM وDBN بهتر از سایر الگوریتم‌ها عمل کردند، توانایی آنها در مدیریت داده‌های پیچیده و درنظرگرفتن روابط غیرخطی بین متغیرها است. جنگل تصادفی یک روش یادگیری گروهی است که چندین درخت تصمیم را ایجاد می‌کند و آن‌ها را برای پیش‌بینی نهایی ترکیب می‌کند. این کار در مدیریت داده‌های نویزی و شناسایی ویژگی‌های مهم موثر است.

عامل مهم دیگری که به دقت مدل‌ها کمک کرده، کیفیت ویژگی‌های استفاده شده بود. ویژگی‌های انتخاب شده برای این مطالعه بر اساس تحقیقات قبلی بوده است. شایان ذکر است که نتایج به‌دست‌آمده در این مطالعه بر اساس مجموعه داده‌های خاص و مجموعه‌ای از پارامترها بوده است. عملکرد مدل ها ممکن است بسته به مجموعه داده مورد استفاده و فراپارامترهای انتخاب شده متفاوت باشد. بنابراین، انجام تحقیقات بیشتر و اعتبارسنجی نتایج به‌دست‌آمده در این مطالعه با استفاده از مجموعه داده‌ها و پارامترهای مختلف مهم است.

نتایج ما نشان داد که جنگل تصادفی، SVMو DBN از نظر عملکرد از سایر الگوریتم‌ها بهتر عمل کردند. این نشان می‌دهد که این الگوریتم ها برای شناسایی تراکنش های مشکوک در شبکه بیت کوین امیدوار کننده هستند.

مجرمان به طور فزاینده ای در استفاده از ارزهای رمزپایه مانند بیت کوین برای پولشویی تجربه کرده اند. استفاده از ارزهای دیجیتال می تواند هویت مجرمانه را پنهان کند و صدها میلیون دلار پول کثیف را از طریق کیف پول دیجیتال جنایی آنها منتقل کند. با این حال، این یک پارادوکس در نظر گرفته می‌شود، زیرا ارزهای دیجیتال منبع ارزشمندی برای اطلاعات منبع باز هستند و به سازمان‌های مجری قانون قدرت بیشتری در هنگام انجام تحلیل‌های قانونی می‌دهند. مقاله لو و همکاران (۲۰۲۳) یک چارچوب شبکه عصبی گراف (GNN) را همراه با الگوریتم‌های یادگیری نظارت شده، یعنی جنگل تصادفی (RF) را برای تشخیص معاملات غیرقانونی پیشنهاد کرد. روش پیشنهادی بر روی مجموعه داده‌ای استفاده شده و از نظر معیارهای کلیدی طبقه‌بندی نشان می‌دهد که GNN نیز در تشخیص تراکنش‌های غیرقانونی ارز دیجیتال پتانسیل بالایی دارد. این نتایج همسو با یافته‌های پژوهش حاضر مبنی بر قابلیت مدل‌های یادگیری ماشین در تشخیص پولشویی است.

یافته های این مطالعه پیامدهای مهمی برای سازمان های مجری قانون و موسسات مالی دارد. استفاده از یادگیری ماشین و الگوریتم‌های یادگیری عمیق می‌تواند به خودکارسازی فرآیند شناسایی پولشویی در تراکنش‌های بیت‌کوین و بهبود کارایی تحقیقات کمک کند. با این حال، توجه به این نکته ضروری است که استفاده از این الگوریتم ها نباید جایگزین تخصص و قضاوت انسان شود. نتایج به دست آمده توسط الگوریتم ها باید قبل از هر اقدامی توسط متخصصان آموزش دیده بررسی شود.

به طور کلی، این مطالعه پتانسیل یادگیری ماشینی و الگوریتم های یادگیری عمیق را در تشخیص پولشویی در شبکه بیت کوین برجسته می کند. با این حال، تحقیقات بیشتری برای تأیید این یافته ها و کشف استفاده از ویژگی ها و مجموعه داده های بیشتر در این زمینه مورد نیاز است.

محدودیت‌های پژوهش حاضر شامل موارد زیر است:

۱) در دسترس بودن داده ها: یک محدودیت ذاتی در تجزیه و تحلیل بلاک‌چین، در دسترس بودن و دسترسی به داده های جامع و قابل اعتماد است. دقت و کامل بودن داده‌ها ممکن است بر اثربخشی مدل‌سازی شما و شناسایی فعالیت‌های پولشویی تأثیر بگذارد.

۲) کیفیت داده ها: کیفیت داده های مورد استفاده در تحقیق نیز می تواند یک محدودیت بالقوه باشد. داده‌های بلاک چین ممکن است حاوی خطاها، ناسازگاری‌ها یا اطلاعات ناقص باشند که ممکن است بر قابلیت اطمینان و دقت مدل‌سازی تأثیر بگذارد.

۳) پیچیدگی مدل: پیچیدگی ریسک مالی و شناسایی پولشویی می تواند چالش هایی را در توسعه مدل های یادگیری ماشینی دقیق و قابل اعتماد ایجاد کند. ایجاد یک مدل جامع و موثر که تمام متغیرها و عوامل لازم را در نظر می گیرد ممکن است چالش برانگیز باشد.

۴) تفسیرپذیری: برخی از الگوریتم‌های یادگیری ماشین، مانند مدل‌های یادگیری عمیق، اغلب به عنوان جعبه‌های سیاه در نظر گرفته می‌شوند که تفسیر و توضیح منطق پشت پیش‌بینی‌های آنها را به چالش می‌کشد. ممکن است تفسیر و توضیح نتایج مدل برای ذینفعان یا قانون‌گذاران دشوار باشد.

فهرست منابع

* Baader, G. and Krcmar, H. (2018), "Reducing false positives in fraud detection: combining the red flag approach with process mining", International Journal of Accounting Information Systems, Vol. 31, pp. 1-16.
* Battaglia, P.W. Hamrick, J.B. Bapst, V. Sanchez-Gonzalez, A. Zambaldi, V. Malinowski, M. Tacchetti, A. Raposo, D. Santoro, A. Faulkner, R. Gulcehre, C. Song, F. Ballard, A. Gilmer, J. Dahl, G. Vaswani, A. Allen, K. Nash, C. Langston, V. Dyer, C. Heess, N. Wierstra, D. Kohli, P. Botvinick, M. Vinyals, O. Li, Y. and Pascanu, R. (2018), "Relational inductive biases, deep learning, and graph networks", arXiv e-prints.
* Bidabad, B. (2017), "Money laundering detection system (MLD) (a complementary system of Rastin Banking)", Journal of Money Laundering Control, Vol. 20 No. 4, pp. 354-366.
* Chang, R., Lee, A., Ghoniem, M., Kosara, R., Ribarsky, W., Yang, J., Suma, E., Ziemkiewicz, C., Kern, D. and Sudjianto, A. (2008), "Scalable and interactive visual analysis of financial wire transactions for fraud detection", Information Visualization, Vol. 7 No. 1, pp. 63-76.
* Colladon, A.F. and Remondi, E. (2017), "Using social network analysis to prevent money laundering", Expert Systems with Applications, Vol. 67, pp. 49-58.
* Deng, X., Joseph, V.R., Sudjianto, A. and Wu, C.F.J. (2009), "Active learning through sequential design, with applications to detection of money laundering", Journal of the American Statistical Association, Vol. 104 No. 487, pp. 969-981.
* Drezewski, R., Sepielak, J. and Filipkowski, W. (2012), "System supporting money laundering detection", Digital Investigation, Vol. 9 No. 1, pp. 8-21.

* Ferwerda, J., Kattenberg, M., Chang, H.H., Unger, B., Groot, L. and Bikker, J.A. (2013), "Gravity models of trade-based money laundering", Applied Economics, Vol. 45 No. 22, pp. 3170-3182.
* Gao, Z. and Ye, M. (2007), "A framework for data mining-based anti-money laundering research", Journal of Money Laundering Control, Vol. 10 No. 2, pp. 170-179.
* Gepp, A. (2015), "Financial statement fraud detection using supervised learning methods", Gold Coast, Queensland, Bond University.
* Gepp, A. (2016), "Addressing the problem of financial statement fraud: Better detection through improved models", 8th Asia-Pacific Interdisciplinary Research in Accounting (APIRA) Conference, Melbourne.
* Gepp, A., Linnenluecke, M.K., O'Neill, T.J. and Smith, T. (2018), "Big data techniques in auditing research and practice: Current trends and future opportunities", Journal of Accounting Literature, Vol. 40, pp. 102-115.
* Gilmour, N. (2017), "Blindingly obvious and frequently exploitable- Money laundering through the purchasing of high-value portable commodities", Journal of Money Laundering Control, Vol. 20 No. 2, pp. 105-115.
* Irwin, A.S.M., Kim-Kwang, R.C. and Liu, L. (2012), "An analysis of money laundering and terrorism financing typologies", Journal of Money Laundering Control, Vol. 15 No. 1, pp. 85-111.
* Ju, C. and Zheng, L. (2009), "Research on suspicious financial transactions recognition based on Privacy-Preserving of classification algorithm", 2009 First International Workshop on Education Technology and Computer Science, pp. 525-528.
* Khaled, H., Kuldeep, K. and Adrian, G. (2018), "Using Cutting-Edge Tree-Based stochastic models to predict credit risk", Risks, Vol. 6 No. 2, p. 55.
* Ngai, E.W.T., Hu, Y., Wong, Y.H., Chen, Y. and Sun, X. (2011), "The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature", Decision Support Systems, Vol. 50 No. 3, pp. 559-569.
* Perols, J. (2011), "Financial statement fraud detection: an analysis of statistical and machine learning algorithms", Auditing: A Journal of Practice and Theory, Vol. 30 No. 2, pp. 19-50.
* Phua, C. Lee, V. Smith, K. and Gayler, R. (2010), "A comprehensive survey of data mining-based fraud detection research", arXiv preprint arXiv:1009.6119.
* Ravenda, D., Argilés, -Bosch, J.M., Valencia, -. and Silva, M.M. (2015), "Detection model of legally registered mafia firms in Italy", European Management Review, Vol. 12 No. 1, pp. 23-39.
* Regan, S. Adams, H. Guiral, P. and Chouri, S. (2017), "Evolving AML Journey – Leveraging machine learning within anti-Money laundering transaction monitoring", Accenture Consulting.
* Sahin, Y., Bulkan, S. and Duman, E. (2013), "A cost-sensitive decision tree approach for fraud detection", Expert Systems with Applications, Vol. 40 No. 15, pp. 5916-5923.
* Savage, D. Wang, Q. Chou, P. Zhang, X. and Yu, X. (2016), "Detection of money laundering groups using supervised learning in networks".
* Singh, K. and Best, P. (2019), "Anti-money laundering: using data visualization to identify suspicious activity", International Journal of Accounting Information Systems, Vol. 34
* Song, X., Hu, Z., Du, J. and Sheng, Z. (2014), "Application of machine learning methods to risk assessment of financial statement fraud: evidence from China", Journal of Forecasting, Vol. 33 No. 8, pp. 611-626.
* Turner, A. and Irwin, A.S.M. (2018), "Bitcoin transactions: a digital discovery of illicit activity on the blockchain", Journal of Financial Crime, Vol. 25 No. 1, pp. 109-130.
* Unger, B., Ferwerda, J., Nelen, H. and Ritzen, L. (2011), Money Laundering in the Real Estate Sector: Suspicious Properties, Edward Elgar, Cheltenham.
* Van Vlasselaer, V., Eliassi-Rad, T., Akoglu, L., Snoeck, M. and Baesens, B. (2017), "GOTCHA! NetworkBased fraud detection for social security fraud", Management Science, Vol. 63 No. 9, pp. 3090-3110.

* Wang, Y., Xu, D., Wang, H., Ye, K. and Gao, S. (2007), "Agent-oriented ontology for monitoring and detecting money laundering process", Proceedings of the 2nd international conference on Scalable information systems, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Suzhou, pp. 1-4.
* Wedge, R. Kanter, J.M. Rubio, S.M. Perez, S.I. and Veeramachaneni, K. (2017), "Solving the" false positives" problem in fraud prediction", arXiv preprint arXiv:1710.07709.
* Zdanowicz, J.S. (2004b), "U.S. Trade with the world and Al Qaeda watch list countries - 2001: an estimate of money moved out of and into the U.S. Due to suspicious pricing in international trade".
* Zdanowicz, J.S. (2009), "Trade-based money laundering and terrorist financing", Review of Law and Economics, Vol. 5 No. 2, pp. 855-878.
* Lo, W. W., Kulatilleke, G. K., Sarhan, M., Layeghy, S., & Portmann, M. (2023). Inspection-L: self-supervised GNN node embeddings for money laundering detection in bitcoin. Applied Intelligence, 1-12.
* Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., & Leiserson, C. E. (2019). Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. arXiv preprint arXiv:1908.02591.
* Alotibi, J., Almutanni, B., Alsubait, T., Alhakami, H., & Baz, A. (2022). Money Laundering Detection using Machine Learning and Deep Learning. International Journal of Advanced Computer Science and Applications, 13(10).
* See, K. (2023). The Satoshi laundromat: a review on the money laundering open door of Bitcoin mixers. Journal of Financial Crime, (ahead-of-print).

Financial Economics
Vol. (14) Issue (55) Autumn 2024

Credit risk modeling of cryptocurrency market using machine learning: Application in detection of money laundering in Bitcoin transactions

Zahra Bozorg Tabar baei
PhD student of Financial engineering, Department of Management, Rasht Branch, Islamic Azad University, Rasht, Iran
z_bozorgtabar@yahoo.com

Reza Aghajan Nashtaei
Department of Business Management, Rasht Branch, Islamic Azad University, Rasht, Iran
(Corresponding Author)
Nashtaei@iaurasht.ac.ir

Mohammad Hassan Gholizadeh
Department of Management, University of Guilan, Rasht, Iran
gholizadeh@guilan.ac.ir

Abstract
The purpose of this research is to provide a deeper understanding of credit risk modeling and to evaluate the performance of machine learning and deep learning algorithms in detecting money laundering (as an aspect of credit risk) in Bitcoin transactions. For this purpose, six different machine learning algorithms, including artificial neural network (ANN), random forest (RF), K-nearest neighbor (KNN), support vector machine (SVM), and two deep learning algorithms including deep belief network (DBN)) and long short-term memory (LSTM) have been used. In addition, elliptical money laundering detection data related to Bitcoin transactions have been used in this research as a dataset used in machine learning methods. The statistical sample covers transaction data for the year 2021. Computational analysis was performed using R software (version 3.4.0) and MATLAB. The results showed that random forest, support vector machine (SVM) and DBN algorithms provided the best performance. Other algorithms, including LSTM, KNN, and ANN, also perform well, but their performance is lower compared to random forest, SVM, and DBN. Overall, this study highlights the potential of machine learning and deep learning algorithms in detecting money laundering in the Bitcoin network.
Keywords: money laundering detection, machine learning, Bitcoin, deep learning