استخراج تراکنش‌های مشکوک به تقلب در داده‌های کارت‌بانکی بدون برچسب

سیدمرتضی سیدرضایی[1]، قربان خردمندیان[2] و سیدجواد کاظمی‌تبار*[3]

[1]گروه پشتیبانی عملیات کارت، شرکت خدمات انفورماتیک، تهران، ایران

[2]شرکت داده‌کاوان هوشمند توسن، تهران، ایران

[3]گروه مخابرات، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی نوشیروانی بابل، بابل، ایران

چکیده

با پیشرفت و گسترش فنّاوری شاهد رشد بالای استفاده از کارت‌های عابر بانک هستیم. با افزایش استفاده از کارت‌های بانکی، همواره فرصت‌هایی برای مهاجمان فراهم می‌شود؛ لذا به‌کارگیری الگوریتم‌های تشخیص تقلب به‌منظور جلوگیری از اقدامات متقلبانه در کارت‌های بانکی اجتناب‌ناپذیر است. داده‌کاوی به‌عنوان یک تکنیک که قادر به شناسایی الگوهای مفید از میان انبوهی از داده‌هاست، یکی از روش‌های مؤثر در تشخیص تقلب در این حوزه است. هدف اصلی این مقاله ارائه یک روش جدید در تشخیص داده‌های پرت بدون نظارت است که از دقت و فراخوانی بالایی برخوردار باشد. روش پیشنهادی این مقاله، ترکیب تکنیک‌های NMF، hierarchical k-means، k-means و k نزدیک‌ترین همسایه است. برای ارزیابی روش پیشنهادی تشخیص داده‌های پرت، ارزیابی‌های متعددی با استفاده از داده‌های استاندارد انجام شد و از منظر دقت و فراخوانی با الگوریتم‌های معیار همچون Isolation Forest، K Nearest Neighbors، Median kNN، Average kNN و غیره مقایسه شد. مطابق نتایج به‌دست‌آمده از آزمایش‌ها، روش پیشنهادی از دقت و فراخوانی بالاتری نسبت به دیگر الگوریتم‌ها برخوردار است.

واژگان کلیدی: تشخیص تقلب کارت‌های بانکی، داده‌کاوی، شناسایی داده‌های پرت، NMF، kmeans سلسله‌مراتبی

Detecting Suspicious Card Transactions in unlabeled data of bank Using Outlier Detection Techniqes

Seyed Morteza Seyed Rezaie[1], Ghorban Kheradmandian[2] & Javad Kazemitabar*[3]

[1] Mechanics, Electrical Power and Computer Faculty, Science and Research Branch Islamic Azad University, Tehran, Iran

[2]Tosan Intelligent Data Miners, Tehran, Iran

[3]Electrical and Computer Engineering Faculty, Babol Noshirvani University of Technology, Babol, Iran

Abstract

With the advancement of technology, the use of ATM and credit cards are increased. Cyber fraud and theft are the kinds of threat which result in using these Technologies. It is therefore inevitable to use fraud detection algorithms to prevent fraudulent use of bank cards. Credit card fraud can be thought of as a form of identity theft that consists of an unauthorized access to another person's card information for the purpose of charging purchases to the account or removing funds from it. Credit card fraud schemes are divided into two categories: application fraud and account takeover. When a credit card account gets opened without someone's permission is called application fraud. Account takeovers, on the other hand, is when an existing credit card account is hijacked, and the criminal obtains enough personal information to modify the account's information. The criminal then subsequently reports the card lost or stolen in order to obtain a new card and make unauthorized purchases with it. Data mining as a technique capable of identifying useful patterns among a great deal of data is an effective method in detecting fraud in this regard. The main purpose of this paper is to present a new method for detecting unattended outliers that require high accuracy and recall. The method presented in this study is based

* نویسندهٔ عهده‌دار مکاتبات

* Corresponding author

• تاریخ ارسال مقاله: ۱۳۹۸/۱۲/۲۸ • تاریخ پذیرش: ۱۴۰۱/۸/۱۵ • تاریخ انتشار: ۱۴۰۱/۱۰/۳ • نوع مطالعه: پژوهشی

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]

[DOI: 10.52547/jsdp.19.3.87]

on a combination of NMF, hierarchical k-means, k-means and k-nearest neighbors' techniques. To evaluate the proposed method of outlier detection, several experiments were performed using standard data, in terms of accuracy and recall with Isolation Forest, k-nearest neighbors, Median kNN, and Average kNN. The dataset used in this paper is one that was provided in a 2016 Kaggle competition and was provided by a European bank after anonymization. The results, corroborate that the proposed method has higher accuracy and recall than other algorithms.

Keywords: Fraud detection, Data mining, Outlier detection, hierarchical k means, NMF

۱- مقدمه

همان‌طور که تعداد کاربران کارت‌های عابر بانک در سراسر جهان افزایش می‌یابد، فرصت‌هایی برای مهاجمان برای به‌سرقت‌بردن جزئیات کارت و پس‌ازآن ارتکاب تقلب نیز افزایش می‌یابد. خریدهای مبتنی بر کارت عابر بانک به دو نوع دسته‌بندی می‌شوند: ۱-کارت فیزیکی و ۲- کارت مجازی. در خرید مبتنی بر کارت فیزیکی، دارندهٔ کارت، کارتش را به‌صورت فیزیکی برای پرداخت به دستگاه پوز می‌دهد. در این نوع از خریدها برای ارتکاب یک تراکنش تقلبی، مهاجم باید کارت‌بانکی را بدزد. اگر دارندهٔ کارت، نبود کارتش را نفهمد، ممکن است منجر به ازدست‌رفتن مبلغ قابل‌توجهی شود [1,2]. در نوع دوم خرید، فقط اطلاعات مهمی در مورد یک کارت، شماره کارت، تاریخ انقضا، کد امنیتی(برای پرداخت کافی هست. تنها راه تشخیص این نوع تقلب‌ها این است که الگوهای مخارج در هر کارت تجزیه‌وتحلیل شود تا هرگونه تناقض نسبت به الگوهای مخارج معمول، کشف شود. تشخیص تقلب براساس تجزیه‌وتحلیل داده‌های موجود خرید دارندهٔ کارت، یک راه امیدوارکننده برای کاهش نرخ تقلب در کارت‌های عابر بانک است. ازآنجاکه انسان‌ها تمایل به نمایش نمایه‌های رفتاری خاصی دارند، رفتار هر دارندهٔ کارت نیز می‌تواند به‌وسیلهٔ مجموعه‌ای از الگوها معرفی شود. این الگوها شامل اطلاعات دربارهٔ دسته‌بندی خریدهای رایج، مدت‌زمان سپری‌شده از آخرین خرید، مقدار پول خرج شده، محل‌های جغرافیایی خرید و غیره هستند. انحراف ازاین‌گونه الگوها تهدید بالقوه‌ای برای سامانه است.

داده‌کاوی نقش مهمی در تشخیص تقلب مالی بازی می‌کند، به‌طوری‌که اغلب برای استخراج و کشف حقایق پنهان در مقادیر بزرگی از داده‌ها از رویکردهای آن استفاده می‌شود. برخی داده‌کاوی را به‌عنوان فرآیندی می‌دانند که از آمار، ریاضی، هوش مصنوعی و الگوریتم‌های یادگیری ماشین برای استخراج و تشخیص اطلاعات مفید استفاده می‌کند و به‌تبع آن از یک پایگاه داده بزرگ، دانش را به‌دست می‌آورد [3]. عده‌ای از پژوهش‌گران بیان کردند که مزیت مهم داده‌کاوی این است که می‌تواند برای توسعهٔ دستهٔ جدیدی از مدل‌ها برای تشخیص حملات جدید، قبل از اینکه توسط انسان‌های خبره تشخیص داده شوند، استفاده شود [4]. مسئولان کشف تقلب بانکی برای جلوگیری از هرگونه سوءاستفاده از کارت‌های عابر بانکی، باید استفاده از این کارت را به‌دقت بررسی کنند و مواردی را که به‌صورت غیرطبیعی در آن‌ها هزینه شده است مشخص کنند. برای مثال اگر یک خرید از خریدهای عادی صاحب کارت عابر بانک بیشتر باشد و اینکه این خرید در محلی بسیار دور از محل زندگی صاحب کارت انجام گیرد؛ یک خرید شک‌برانگیز اتفاق افتاده است. به‌این‌ترتیب باید برای جلوگیری از سوءاستفاده‌های احتمالی، چنین تراکنش‌های مشکوکی تشخیص داده شوند.

هاوکینز نقاط پرت را این‌طور تعریف می‌کند: «دادهٔ پرت مشاهده‌ای است که از دیگر مشاهدات متفاوت است که ما مشکوک می‌شویم که با یک سازوکار متفاوت تولید شده است» [5]. به‌طورکلی داده‌های پرت را می‌توان به سه‌طبقهٔ:

۱. دادهای پرت سراسری
۲. داده‌های پرت زمینه‌ای یا شرطی
۳. داده‌های پرت اجماعی

تقسیم کرد [4]. درصورتی‌که داده‌ای انحراف بسیار زیادی با دیگر داده‌ها داشته باشد، داده پرتی سراسری خوانده می‌شود. درصورتی‌که داده‌ای انحراف بسیار زیادی با توجه به زمینه خاصی با دیگر داده‌های مجموعه‌داده‌ها داشته باشد، داده پرت زمینه‌ای خوانده می‌شود. و درصورتی‌که با در اختیار داشتن یک مجموعه‌داده، زیرمجموعه‌ای از آن مجموعه، انحراف کلی زیادی با بقیه مجموعه‌داده‌ها داشته باشد، یک پرت اجماعی گفته می‌شود. باید به این نکته توجه داشت که لزومی ندارد همه داده‌هایی که در یک پرت اجماعی هستند، به‌تنهایی داده پرت باشند.

یکی از دیدگاه‌هایی که می‌توان از مسئله تشخیص داده‌های پرت داشت، به‌عنوان مسأله طبقه‌بندی‌ای[1] است که در آن برچسب‌ها («عادی» یا «پرت») نامشخص است؛ بنابراین دیدگاه و حقیقت که داده‌های عادی بیش از

[1] Classification

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]

[DOI: 10.52547/jsdp.19.3.87]

داده‌های پرت هستند، می‌توان وانمود کرد که کل مجموعه‌داده‌ها حاوی فقط ردهٔ عادی است و ایجاد یک مدل از داده‌ها را به‌عنوان مدلی از داده‌های عادی در نظر گرفت. انحراف از این مدل عادی را می‌توان به‌عنوان نمرات پرت‌بودن[1] در نظر بگیریم. این ارتباط بین طبقه‌بندی و تشخیص داده‌های پرت خیلی مهم است؛ زیرا بسیاری از نظریه‌ها و الگوریتم‌های طبقه‌بندی تعمیم‌یافته الگوریتم‌های تشخیص داده‌های پرت هستند. ماهیت نامعلوم بودن برچسب‌ها (نمرات پرت‌بودن) دلیلی است بر این‌که مسئله تشخیص داده‌های پرت یک مسئله از نوع بدون نظارت است. در مواردی که برچسب‌ها معلوم باشند، مسئله تشخیص داده‌های پرت به یک مسئله از نوع با نظارت نامتوازن[2] تبدیل می‌شود [5].

با بررسی‌های انجام‌شده در داده‌های بانک‌های ایرانی با توجه به اینکه کارشناسان مربوطه برچسبی به‌ازای داده‌ها ارائه نمی‌دهند؛ بنابراین در مورد مسئله تشخیص داده‌های پرت در این بانک‌ها با مسئله‌ای بدون نظارت روبه‌رو هستیم. هدف این مقاله این است که الگوریتمی جدید برای تشخیص داده‌های مشکوک به تقلب از این داده‌های بدون برچسب ارائه کند. الگوریتم پیشنهادی دارای ویژگی‌های زیر خواهد بود:

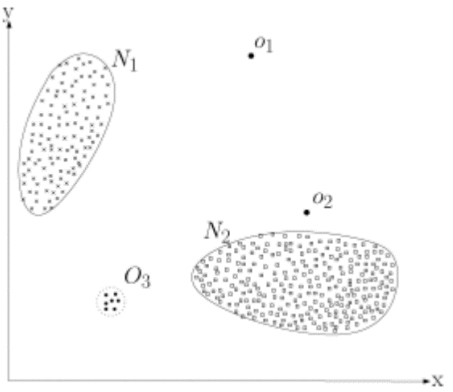
- ✓ استفاده از الگوریتم‌های استخراج ویژگی
- ✓ مبتنی بر الگوریتم‌های تشخیص داده‌های پرت چگالی محور
- ✓ استفاده از خوشه‌بندی داده‌ها

۲- مطالعات انجام‌شده

پژوهش‌های زیادی در زمینهٔ تشخیص موارد مشکوک به تقلب کارت‌های عابر بانک انجام شده است. یکی از رویکردهای پرکاربرد جهت تشخیص تقلب در کارت‌های عابر بانک، استفاده از الگوریتم‌های تشخیص نقاط پرت است [6, 7, 8]. در این بخش به بررسی انواع الگوریتم‌های مختلف تشخیص نقاط پرت خواهیم پرداخت. الگوریتم‌های تشخیص داده‌های پرت سال‌هاست که مورداستفاده قرار می‌گیرند و یا این نقاط پرت نقاط کم‌اهمیتی هستند، که موجب می‌شود که این نقاط غیر نرمال از داده‌ها حذف شوند و یا این نقاط از اهمیت بالایی برخوردار هستند، که موجب می‌شود این نقاط به‌طور خودکار تشخیص داده شوند.

تشخیص نقاط پرت یک مسألهٔ مهم و بزرگی است که در حیطه‌های پژوهشی و کاربردی گسترده‌ای بررسی شده است. بسیاری از الگوریتم‌های تشخیص داده‌های پرت برای حوزه‌های کاربردی ویژه‌ای توسعه داده شده‌اند، و بعضی از آن‌ها نیز جنبه عمومی دارند. در این پژوهش انواع الگوریتم‌های موجود به بخش‌های گوناگونی تقسیم‌بندی می‌شوند.

در شکل (۱) داده‌ها شامل دو ناحیهٔ نرمال N_1 و N_2 است که عمدهٔ داده‌ها در این دو ناحیه قرار می‌گیرند. نقاطی که به‌طور قابل‌توجهی از این دو ناحیه دور هستند، برای مثال O_1 و O_2 و O_3، پرت محسوب می‌شوند.



(شکل-۱): مثالی از داده‌های پرت در مجموعه‌داده‌های دوبعدی. [7]

(Figure-1): Example of outlier data in 2d data set [8]

برخی نویسندگان روش‌های مبتنی تابع توزیع انباشته را برای شناسایی داده پرت پیشنهاد کرده‌اند [10]. روش‌های ابتکاری جهت پیرایش داده‌های حجیم و امکان کشف داده پرت با محاسبات کمتر پیشنهاد شده‌اند [11]. برخی نویسندگان با ارائه یک الگوریتم امکان درجه پرت‌بودن داده را با فرمول نسبت ردریگز ارائه داده‌اند [12].

یک رویکرد جهت تشخیص داده‌های پرت این است، ناحیه‌ای که رفتار عادی دارد مشخص شود و هر داده‌ای که به آن ناحیه تعلق ندارد را به‌عنوان پرت قلمداد کنیم. ولی فاکتورهای زیادی این رویکرد به‌ظاهر ساده را به چالش می‌کشد:

- ✓ تعیین ناحیه‌ای با رفتار نرمال، آسان نیست. همچنین مرزبندی بین رفتار نرمال و رفتار غیر نرمال اغلب دقیق نیست.
- ✓ متقلبان سعی می‌کند رفتارشان را نرمال نشان دهند؛ بنابراین تعیین رفتار نرمال سخت می‌شود.

[1] Outlier Scores
[2] Imbalanced

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]
[DOI: 10.52547/jsdp.19.3.87]

- ✓ مفهوم غیر نرمال‌بودن در حوزه‌های مختلف متفاوت است. برای مثال یک انحراف کوچک از داده‌های نرمال در حوزۀ پزشک، غیر نرمال محسوب می‌شود، درحالی‌که چنین انحرافی در حوزۀ بازار سهام می‌تواند نرمال تلقی شود.
- ✓ یک موضوع پراهمیت در استفاده از الگوریتم‌های تشخیص داده‌های پرت، موجودبودن مجموعه‌داده‌های آموزشی و تستی بر چسب‌دار است.
- ✓ چالش دیگر تفاوت بین داده دارای اختلال و دادۀ پرت است.

انتخاب تکنیک تشخیص داده‌های پرت به حوزۀ مسأله و فاکتورهای مرتبط به آن وابسته است. مثل ماهیت داده‌ها، برچسب‌داشتن یا نداشتن داده‌ها، نوع ناهنجاری که بایستی تشخیص شود و غیره. پژوهش‌گران مفاهیمی را از رشته‌های گوناگون مانند یادگیری ماشین، داده‌کاوی، تئوری اطلاعات و آمار اتخاذ کرده و آن‌ها را برای مسائل خاص فرموله کرده‌اند.

چهار جنبۀ مختلف یک مسألۀ تشخیص داده‌های پرت عبارت است از:

ماهیت داده‌ها: هر داده با مجموعه‌ای از ویژگی‌ها معرفی می‌شود. ویژگی‌ها می‌توانند انواع مختلفی مانند دودویی، دسته‌ای یا پیوسته داشته باشند. یک داده می‌تواند شامل یک ویژگی و یا چند ویژگی باشد. ماهیت ویژگی‌ها، کارایی الگوریتم مورداستفاده برای تشخیص داده‌های پرت را تعیین می‌کند.

نوع غیر نرمال‌بودن: یک جنبۀ مهم در الگوریتم‌های تشخیص داده‌های پرت، نوع ناهنجاری موردنظر است. انواع این نوع ناهنجاری، مواردی از قبیل داده‌های پرت سراسری، داده‌های پرت زمینه‌ای یا شرطی، داده‌های پرت اجماعی هستند که در بخش نخست همین پژوهش شرح داده شد.

برچسب داده‌ها: امکان تشخیص داده‌های پرت باعث به‌وجودآمدن راه‌کارهای مختلف داده‌کاوی در زمینه تشخیص داده‌های پرت شده است. سه راهبرد کلی عبارتنداز:

۱. یادگیری بانظارت[1]
۲. یادگیری نیمه نظارتی[2]
۳. یادگیری بدون نظارت[3]

خروجی الگوریتم: چاروآگرووال خروجی یک الگوریتم تشخیص داده‌های پرت را یکی از دو نوع زیر دانسته است: ۱. امتیاز دورافتادگی ۲. برچسب‌های دودویی [6] در این پژوهش خروجی الگوریتم پیشنهادشده برچسب‌های دودویی تولید می‌کند. برحسب فرضی که در بخش نخست انجام گرفت، داده‌ها برچسب ندارند، به همین دلیل از الگوریتم‌های بدون نظارت باید استفاده کنیم. برای مقایسه الگوریتم پیشنهادی از الگوریتم‌های جدول (۱) استفاده خواهیم کرد.

جهت انتخاب و استفاده از یک تکنیک تشخیص داده‌های پرت بایستی به مسائلی از قبیل ماهیت داده‌ها، وجود داده‌های آموزشی بر چسب‌دار توجه داشت. هریک از روش‌های شناسایی داده‌های پرت دارای مزایا و معایب خاص خود هستند؛ اما می‌توان بر اساس کاربرد و داده‌های مورداستفاده، از ترکیب روش‌ها استفاده کرد و به یک روش بهینه جهت تشخیص نمونه‌های پرت دست‌یافت.

(جدول-۱): مقایسه الگوریتم‌های مختلف
(Table-1): Diffrent algorithm comparsion

ردیف	نوع الگوریتم	نام اختصاری	نام الگوریتم	سال انتشار
1	مبتنی بر مجاورت	LOF	Local Outlier Factor	2000
2	مبتنی بر مجاورت	CBLOF	Clustering-Based Local Outlier Factor	2003
3	مبتنی بر مجاورت	KNN	k Nearest Neighbors	2000
4	مبتنی بر مجاورت	AvgKNN	Average k Nearest Neighbors	2002
5	مبتنی بر مجاورت	MedKNN	Median k Nearest Neighbors	2002
6	مدل خطی	OCSVM	One-Class Support Vector Machines	2002
7	احتمالاتی	ABOD	Angle-Based Outlier Detection	2008
8	ترکیبی	IForest	Isolation Forest	2008
9	ترکیبی		Feature Bagging	2005
10	شبکه عصبی	AutoEncoder	Fully connected AutoEncoder	2015

۳- روش پیشنهادی

در این بخش رویکرد پیشنهادی ما جهت شناسایی تراکنش‌های مشکوک به تقلب در کارت‌های عابر بانکی

[1] Supervised learning
[2] Semi-supervised learning
[3] Unsupervised learning

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]
[DOI: 10.52547/jsdp.19.3.87]

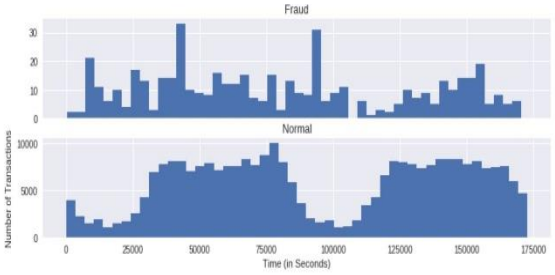
ارائه می‌شود. این الگوریتم پیشنهادی تشخیص داده‌های پرت به‌طوراساسی یک روش بی نظارت است و برای داده‌های با ابعاد بزرگ نیز کارایی دارد. در شکل (۲) نمودار جعبه‌ای فرایند این الگوریتم را می‌توان مشاهده کرد. در ادامه به بررسی گام‌های مطرح شده در روش پیشنهادی خواهیم پرداخت.

۱-۳- معرفی داده

این مجموعه‌داده‌ها که از [9] به‌دست آمده یک مجموعه‌داده برچسب‌دار استاندارد است که شامل معاملات انجام‌شده با کارت‌های عابر بانک در سپتامبر ۲۰۱۳ توسط مشتریان کارت‌های اروپایی است. این مجموعه‌داده تراکنش‌هایی را که طی دو روز اتفاق افتاده است ارائه می‌دهد و دراین‌بین ۴۹۲ کلاه‌برداری از ۲۸۴،۸۰۷ معامله داریم. مجموعه‌داده بسیار نامتعادل است، طبقه تقلب در مجموعه‌دادهٔ ما ٪۰/۱۷۲ از کل معاملات را به خود اختصاص داده‌اند. این مجموعه‌داده شامل فیلدهای عددی حاصل از تحول PCA است. متأسفانه، به‌دلیل مسائل مربوط به محرمانه‌بودن، ما نمی‌توانیم ویژگی‌های اصلی و اطلاعات بیشتر در مورد داده‌ها را ارائه دهیم. ویژگی‌های V1، V2، V28 ...اجزای اصلی به‌دست‌آمده با PCA هستند، تنها ویژگی‌هایی که با PCA متحول نشده‌اند "Time" و "Amount" هستند. ویژگی "Time" شامل ثانیه‌هایی است که بین هر تراکنش و نخستین معامله در مجموعه‌داده است. ویژگی "Class" متغیر پاسخ است و در صورت کلاه‌برداری مقدار یک را می‌گیرد و در غیر این صورت مقدار آن صفر است. اجازه دهید کمی به خصوصیت "Time" توجه کنیم و چگونه این خصوصیت در معاملات تقلبی و معمولی تغییر می‌کند. در شکل (۳) می‌توانید توزیع تعداد تراکنش به تفکیک تراکنش‌های معمولی و تقلب را برحسب خصوصیت "Time" را مشاهده کرد.

خصوصیت "Time" در هر دو نوع معاملات بسیار شبیه به نظر می‌رسد. می‌توانید استدلال کنیم که معاملات کلاه‌برداری به‌طور یک‌نواخت توزیع می‌شود؛ درحالی‌که معاملات عادی توزیع چرخه‌ای دارند. این امر باعث می‌شود تشخیص یک معامله تقلب در یک‌زمان"خارج از اوج" آسان‌تر شود. همچنین می‌توانید توزیع تعداد تراکنش‌ها را به تفکیک تراکنش‌های معمولی و تقلب را برحسب خصوصیت "Amount" " در شکل (۴) مشاهده کرد. بیشتر معاملات مبلغشان کمتر از صد دلار است. مقدار معاملات کلاه‌برداری حداکثر ۲۱۲۵/۸۷ دلار است که مقدارشان به‌مراتب کمتر از معاملات عادی ۲۵۶۹۱/۱۶ دلار است.

(شکل-۲): نمودار جعبه‌ای الگوریتم پیشنهادی
(Figure-2): Block Diagram of the proposed algorithm



(شکل-۳): توزیع تعداد تراکنش به تفکیک تراکنش‌های معمولی و تقلب برحسب خصوصیت "Time"
(Figure-3): Distribution of transaction number based on normal and suspicious transaction versus Time property

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]

[DOI: 10.52547/jsdp.19.3.87]

(شکل-۴): توزیع تعداد تراکنش‌ها به تفکیک تراکنش‌های معمولی و تقلب برحسب خصوصیت Amount

(Figure-4): Distribution of transaction number based on normal and suspicious transaction versus Amount property

۲-۳- مرحله پیش‌پردازش

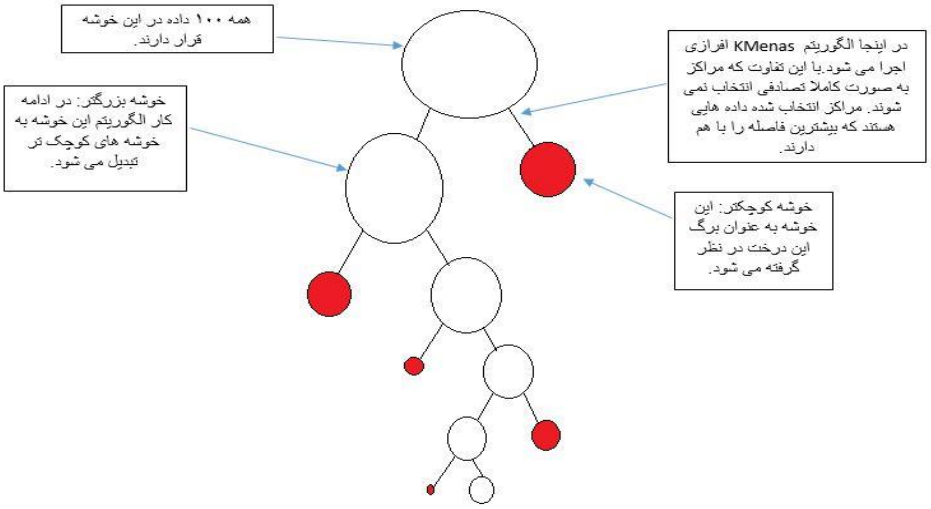
به مجموعه عملیاتی که منجر به تولید مجموعه‌ای از داده‌های پالایش‌شده قابل کاوش خواهد شد، پیش‌پردازش می‌گویند. برای پیش‌پردازش داده‌ها، نیاز است تا آن‌ها را از شکل و حالتِ اولیه، خارج کرده و به شکلی که برای الگوریتم مناسب باشد، تبدیل کنیم. در این قسمت به تشریح عملیات پیش‌پردازشی می‌پردازیم که در الگوریتم پیشنهادی لازم است که انجام شود.

۱-۲-۳- نرمال‌کردن داده‌ها

در این قسمت به نرمال‌کردن خصوصیات می‌پردازیم. به‌دلیل اینکه خصوصیات "V1" تا "V28" نرمال بودند ما نیاز داریم تا دو خصوصیت "Time" و "Amount" را نرمال کنیم تا با بقیه خصوصیات هم مقیاس باشند. ما برای نرمال‌سازی این دو فیلد از (۱) استفاده کرده‌ایم.

$$Zi = \frac{Xi-\mu}{\sigma} \qquad (۱)$$

گفتنی است در فرمول بالا Zi نمره استاندارد برای داده Xi است و μ میانگین و σ انحراف معیار برای داده‌ها است. با این کار Zi ها دارای میانگین صفر و واریانس یک می‌شوند.



(شکل-۵): تصویر شماتیک از اجرای Kmeans سلسله مراتبی

(Figure-5): Scheme of hierarchical Kmeans implementation

۲-۲-۳- استخراج ویژگی‌ها

در ادامه می‌توانیم هرکدام از خصوصیات را تحلیل (برای مثال تحلیل همبستگی[1]) کنیم تا بفهمیم تا چه حد خصوصیات باکیفیتی هستند و بودنشان می‌تواند به پیداکردن تراکنش‌های تقلبی کمک کند و در ادامهٔ تحلیل تنها روی خصوصیاتی کار می‌شود که می‌تواند نتایج بهتری را تولید کند. در این نوشتار از این روش استفاده نشده است. به‌جای این روش از الگوریتمی به نام NMF استفاده شده که کار استخراج ویژگی‌های باکیفیت بالاتر را انجام می‌دهد. در این روش بدون اینکه فیلدی به‌صورت دستی حذف شود، می‌توان با ویژگی‌هایی با تعداد کمتر ولی باکیفیت‌تر کارکرد.

برای اینکه از این الگوریتم بتوانیم استفاده کنیم در ابتدای کار باید یک مرحلهٔ نرمال‌کردن داده‌ها را داشته باشیم. این کار به‌خاطر این انجام می‌شود که ورودی این الگوریتم باید مقادیر مثبت باشند. در این روش نرمال‌کردن هرکدام از داده‌ها به بازهٔ بین صفر تا یک باید برده شود. رابطه کلی برای این نرمال‌کردن به‌صورت (۲) است:

$$Z = \frac{x-\min(X)}{[\max(X)-\min(X)]} \qquad (۲)$$

که در این فرمول x به عددی که می‌خواهد نرمال شود و min(x) به کمترین عدد در آن مجموعه و max(X) به بزرگ‌ترین عدد در آن مجموعه داده اشاره دارد.

[1] Correlation

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]
[DOI: 10.52547/jsdp.19.3.87]

در این مرحله هدف ما این بوده که از الگوریتم‌های بدون نظارت استخراج ویژگی (کاهش ابعاد[1]، یادگیری ویژگی[2]) استفاده کنیم نه در جهت کاهش ویژگی‌ها بلکه در جهت به‌دست‌آوردن خصوصیاتی که از کیفیت بالاتری برخوردار باشند. این خصوصیات باکیفیت بالاتر دارای سطح بالاتری است که مفهومی‌تر نیز است و نیز می‌تواند الگوهای موجود را بهتر از هم جدا کند. الگوریتم‌های یادگیری ویژگی، ویژگی‌های اولیه مجموعه‌داده را به ویژگی‌های ثانویه‌ای تبدیل می‌کنند (البته می‌توان این کار تا تعداد دلخواه تکرار کرد به‌عنوان‌مثال الگوریتم‌های یادگیری عمیق[3] مثل الگوریتم Auto Encoder). این الگوریتم‌ها سعی دارند علاوه‌بر اینکه همه اطلاعات موجود در داده‌ها حفظ شوند (برای مثال اگر در مجموعه ویژگی‌های اولیه دو داده نزدیک هم بودند در مجموعه ویژگی‌های ثانویه نیز این دو داده نزدیک هم باقی بمانند، به‌عبارتی‌دیگر نظم موجود در داده‌ها با این تبدیل ویژگی‌ها از بین نرود)، خود این الگوریتم‌ها بتوانند گروه‌بندی‌ای از اطلاعات داشته باشند. پس خود این الگوریتم‌ها به زبانی راحت‌تر می‌توانند کار خوشه‌بندی نیز انجام دهند. به این دلیل که الگوها در ویژگی‌های ثانویه قابل‌دسترس‌تر هستند و به مقدار مطلق داده‌ها وابستگی ندارد. به جدول (۲) توجه کنید.

این جدول بعد از اجرای الگوریتمی که کارش یادگیری ویژگی است به جدول (۳) تبدیل می‌شود. در اینجا فرض کردیم که این الگوریتم مقادیر جدول مبدأ را به اعداد صفر یا یک نگاشت می‌کند.

(جدول-۲): مثالی از یک مجموعه داده

(Table-2): An example of a data set

ردیف	شناسه داده	ویژگی ۱	ویژگی ۲	...	ویژگی ۳۰
1	Id_1	221.1	15.3	...	43.34
2	Id_2	156.7	-50.6	...	12.63
3	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
N	Id_n	87.4-	22.1		43.98

(جدول-۳): مثالی از استخراج ویژگی‌ها به‌صورت ساده‌شده

(Table-3): A simplified example of feature extraction

ردیف	شناسه داده	ویژگی ۱	ویژگی ۲	...	ویژگی ۱۰
1	Id_1	1	1	...	0
2	Id_2	1	0	...	1
3	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
N	Id_n	0	1	...	0

همان‌طور که می‌بینید، درون الگوریتم یادگیری ویژگی به‌نوعی خوشه‌بندی نیز انجام می‌شود. شناسه داده‌هایی که در ویژگی‌های مختلف دارای صفر و یک یکسانی هستند، در یک گروه قرار می‌گیرند. به‌دلیل اینکه اعداد، صفر یا یک هستند و اینکه به تعداد $k = 10$ ویژگی وجود دارد، در اینجا $2K$ عدد الگو قابل‌شناسایی است.

در این مسأله از الگوریتم‌های موجود (PCA یا SVD) به‌دلیل قابلیت تفسیرپذیری بالا (به‌دلیل اینکه خروجی این الگوریتم همانند ورودی آن اعداد مثبت هستند) NMF از این الگوریتم NMF استفاده کردیم. در این الگوریتم می‌توان تعداد ابعاد را به تعداد دلخواه کاهش داد. ما نیز حالت‌های مختلف (۵،۱۰،۱۵) را برای این الگوریتم امتحان کردیم.

۳-۳- پیداکردن داده‌های نامزد پرت‌بودن

بعد از پیش‌پردازش داده‌ها که داده‌ها نرمال شدند و بعد از استخراج ویژگی‌های جدید، در این بخش هدف این است که از الگوریتم‌های خوشه‌بندی استفاده شود و درنهایت داده‌های متعلق به خوشه‌های کوچک را می‌خواهیم که به‌عنوان نامزد پرت‌بودن در نظر بگیریم؛ بنابراین نخستین کاری که در جهت تشخیص داده‌های پرت در این الگوریتم پیشنهادی انجام‌شده است همین استفاده از الگوریتم NMF است، زیرا بعد از اعمال این الگوریتم، در مرحلهٔ بعد خوشه‌های باکیفیت‌تری تولید می‌شوند. در مرحله بعد الگوریتم خوشه‌بندی‌ای انتخاب می‌شود که بتواند خوشه‌بندی را به‌صورت نامتوازن انجام دهد، به این دلیل نامتوازن که ایده اولیه این است که داده‌های متعلق به خوشه‌های کوچک نامزدهای خوبی برای پرت‌بودن هستند. اگر این الگوریتم خوشه‌بندی بعد اجراشدن دو خوشه‌ی متوازن را پیدا کند، از این خوشه‌ها نمی‌توانیم به نتیجه‌ای

[1] Feature Reduction
[2] Feature Learning
[3] Deep Learning

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]
[DOI: 10.52547/jsdp.19.3.87]

برسیم. با توجه به آزمایش‌های متعددی که بر روی الگوریتم‌های مختلف انجام شد، منطق الگوریتم خوشه‌بندی Kmeans سلسله‌مراتبی مناسب بود.

این الگوریتم به‌صورت عامدانه انتخاب شده است. خود Kmeans در اصل یک روش افرازی است و سلسله‌مراتبی نیست. الگوریتم Kmeans سلسله‌مراتبی ترکیبی از الگوریتم Kmeans و منطق الگوریتم‌های سلسله‌مراتبی است. به این نحو که در ابتدای کار کل داده‌ها را یک خوشه در نظر می‌گیرد. این خوشه به دو خوشهٔ نامتوازن تقسیم می‌شود. خوشه کوچک به‌عنوان برگ این درخت و خوشه بزرگ‌تر برای تقسیم‌شدن به دو خوشه برای مراحل بعدی الگوریتم در نظر گرفته می‌شود. در شکل (۵) می‌توان منطق این الگوریتم را تا پنج مرحله به‌صورت شماتیک برای صد داده دیده می‌شود. در الگوریتم پیشنهادی، در مرحلهٔ اجرای الگوریتم Kmeans سلسله‌مراتبی برای گذر از هر مرحله یک‌بار الگوریتم Kmeans افرازی اجرا می‌شود و حاصل این اجرا اضافه‌شدن یک خوشه در هر مرحله به درخت است. همان‌طور که دیده می‌شود، این الگوریتم به‌صورت سلسله‌مراتبی، دودویی و از بالا به پایین است. در این الگوریتم در هر مرحله یک خوشه، به یک خوشه بزرگ و یک خوشه کوچک تقسیم می‌شود. پس این الگوریتم بخشی از هدف الگوریتم پیشنهادی را آنجا می‌دهد. در پایان یک تعداد خوشه‌های کوچک و یک تعداد خوشه‌های بزرگ تولید شده‌اند. فرض اولیه نیز برای الگوریتم پیشنهادی این بود که داده‌های متعلق به خوشه‌های کوچک نامزدهای خوبی برای داده‌های پرت است. در رابطه با معیار کوچک‌بودن یا بزرگی یک خوشه باید گفت که همه الگوریتم‌هایی که به‌اندازه خوشه توجه می‌کنند، پارامتری را از کاربر دریافت می‌کنند که معیار کوچک‌بودن خوشه‌ها را مشخص می‌کند.

در انتهای انجام الگوریتم Kmeans سلسله‌مراتبی کل داده‌ها به K تا خوشه تقسیم‌شده است. این K تا خوشه به‌ترتیب از کوچک‌ترین خوشه به بزرگ‌ترین خوشه مرتب می‌شوند؛ پس ترتیب خوشه‌ها به‌صورت فرمول ۳-۳ در می‌آید:

$$S = C1, C2, C3, \dots, Ck$$
$$|C1| < |C2| < |C3| < \cdots < |Ck| \qquad (۳)$$

در رابطه (۳) S نشان‌دهنده کل مجموعه‌داده‌ها، C1 خوشه شماره ۱ و |C1| تعداد اعضای خوشه شماره ۱ است. مثل همه الگوریتم‌های تشخیص داده‌های پرت که پارامتری را به‌عنوان درصد داده‌های پرت می‌گیرند، الگوریتم پیشنهادی نیز این پارامتر را به‌عنوان ورودی می‌گیرد. ازآنجایی‌که طبقه تقلب در مجموعه‌داده ٪۰/۱۷۲ است؛ پس این عدد به‌عنوان ورودی برای الگوریتم پیشنهادی و هم برای همه الگوریتم‌هایی که برای مقایسه و ارزیابی درنظرگرفته شده‌اند به‌عنوان ورودی داده می‌شوند. در ادامهٔ کار درصد داده‌های پرت را در تعداد کل داده‌ها ضرب می‌شوند تا تعداد کل داده‌های پرت را به‌دست آورده شوند، که در اینجا به این صورت محاسبه می‌شود:

$$284807 * 0.172\,\% = 492$$

بنابراین در الگوریتم پیشنهادی به ترتیب از کوچک‌ترین خوشه تا بزرگ‌ترین خوشه به تعداد ۴۹۲ داده از این خوشه‌ها از کوچک‌ترین خوشه‌ها باید انتخاب کرد. به‌این‌ترتیب که ابتدا تعداد اعضای خوشه شماره ۱ را در نظر می‌گیرم، این تعداد اگر ۴۹۲ کوچک‌تر بود خوشه شماره ۱ را به مجموعه جواب اضافه می‌کنیم، بعد تعداد اعضای خوشه شماره ۲ را در نظر می‌گیریم اگر جمع این تعداد با مجموعه تعداد اعضای اضافه‌شده در مرحله قبل کوچک‌تر از عدد ۴۹۲ بود، این خوشه نیز به مجموعه‌جواب اضافه می‌شود و به همین ترتیب تا تعداد اعضای اضافه‌شده حداکثر به تعداد ۴۹۲ برسد، همین کار را برای خوشه‌ها انجام می‌دهیم.

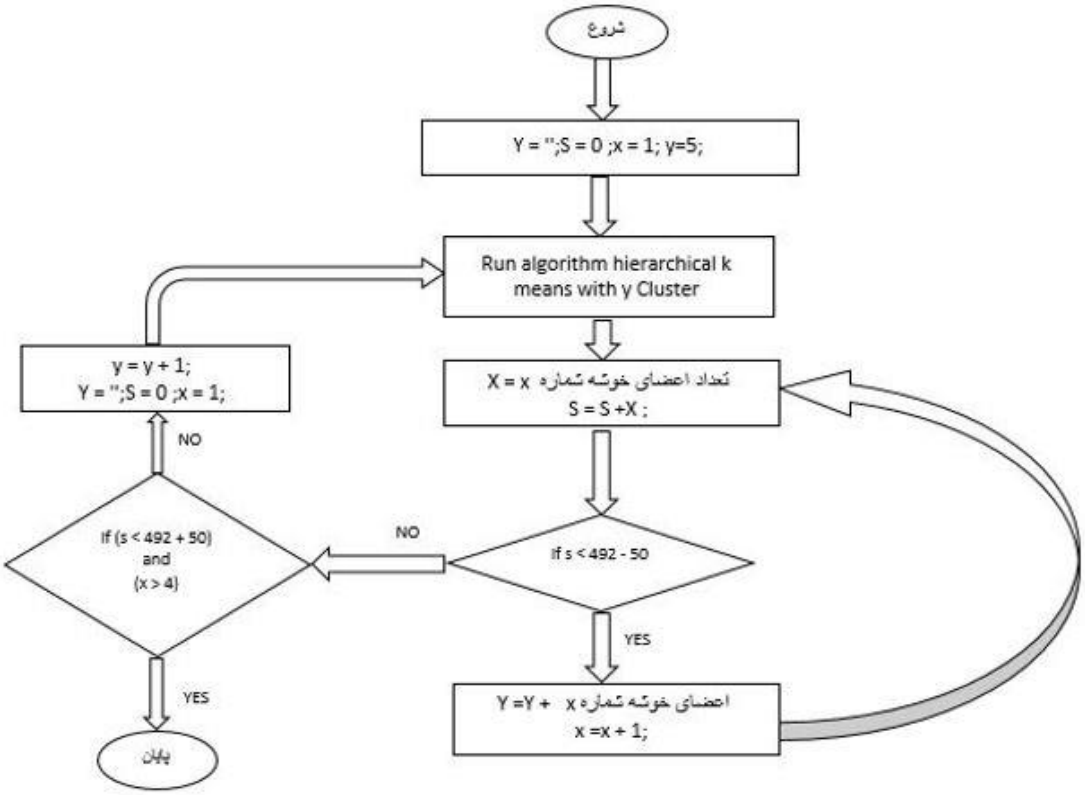
مسأله مهمی که باید به آن توجه داشت، تعداد خوشه‌هایی است که باید الگوریتم Kmeans سلسله‌مراتبی به‌عنوان جواب برمی‌گرداند، است. این الگوریتم به‌عنوان ورودی خود تعداد خوشه‌هایی را که می‌خواهیم ایجاد شود، می‌گیرد. برای اینکه بتوان به خوشه‌هایی باکیفیت مناسب برسیم (خوشه‌های کوچک که تعداد داده‌های پرت‌شان بالاتر از تعداد داده‌های نرمال‌شان باشد) در الگوریتم پیشنهادی به‌این‌ترتیب عمل می‌کنیم که ابتدا الگوریتم Kmeans سلسله‌مراتبی را با پنج خوشه ایجاد می‌کنیم. در مرحلهٔ بعد باید به ترتیبی که در بالا توضیح داده شد با این شرط که اگر توانستیم تعداد (۴۹۲-۵۰) یا (۴۹۲+۵۰) داده از خوشه‌های کوچک جدا کنیم به اجرای الگوریتم Kmeans سلسله‌مراتبی خاتمه می‌دهیم. به این دلیل در اجرای الگوریتم Kmeans سلسله‌مراتبی با پنج خوشه شروع شده است که منطقی نیست که خوشه‌های کوچک در دو یا سه خوشهٔ کوچک قرار گیرد. همچنین در شرط پایانی که به کار این الگوریتم خاتمه می‌دهد، چک می‌شود که آیا تعداد خوشه‌ها بالاتر از ۴ خوشه است یا

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]

[DOI: 10.52547/jsdp.19.3.87]

خیر. در شکل (۶) روندنمای اضافه‌کردن خوشه‌های کوچک را ملاحظه می‌فرمایید.

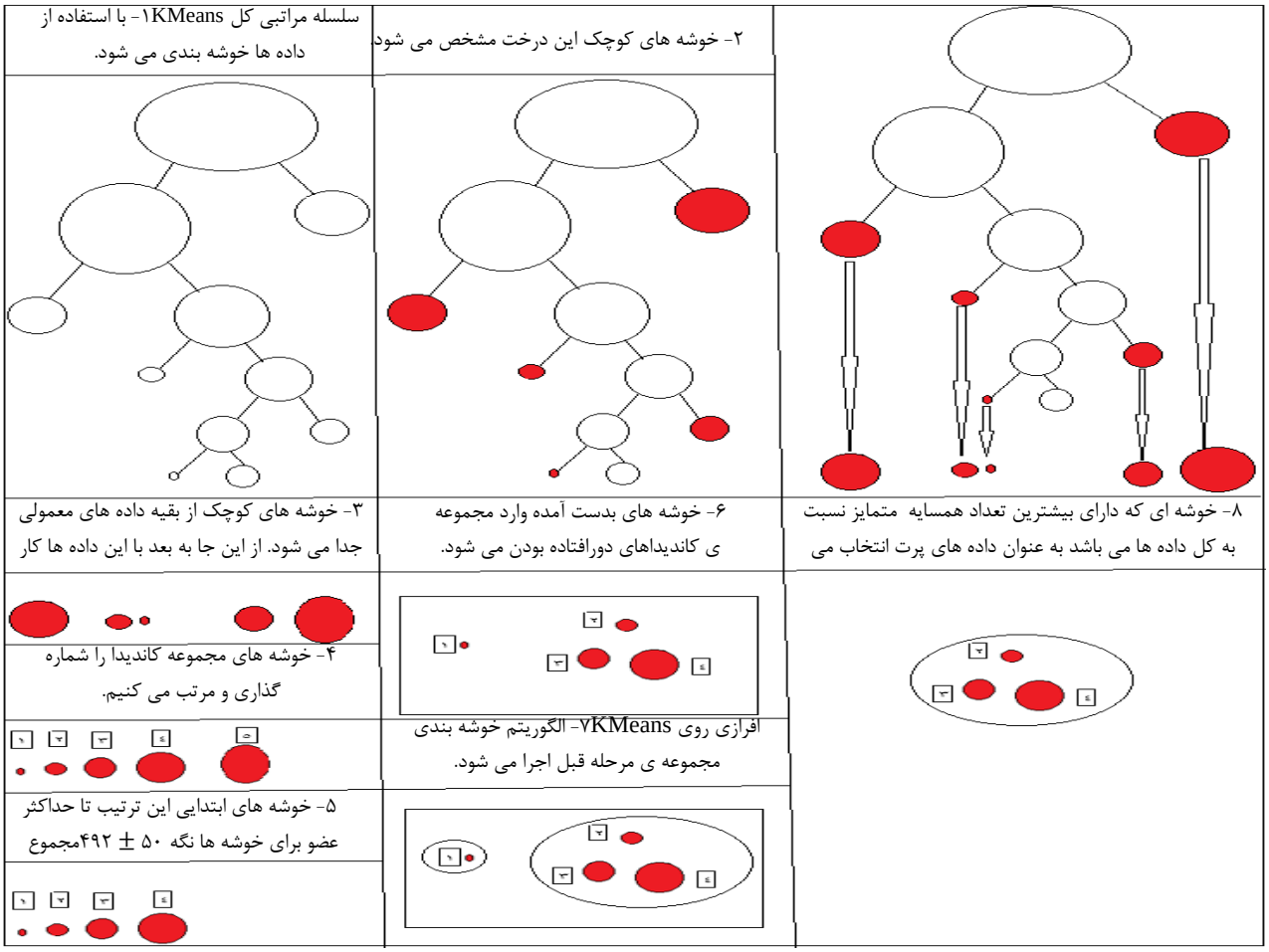
با انجام مرحله قبل زیرمجموعه‌ای از خوشه‌ها به‌عنوان خوشه‌های کوچک به دست آورده شد که جمع اعضای این خوشه‌ها ۵۰ ± ۴۹۲ است. باز پردازشی انجام می‌شود که سعی دارد مجموعه‌اعضای خوشه یا خوشه‌هایی را که شبیه به بقیهٔ خوشه‌های کوچک نیستند، از مجموعه‌داده‌های پرت به مجموعه‌داده‌ای معمولی بازگرداند و در مرحله بعد به همان تعداد اعضا که به مجموعه‌داده‌های معمولی بازگردانده شد، از مجموعه‌داده‌های معمولی به مجموعه‌داده‌های پرت اضافه شود. پس ایدهٔ ما در اینجا این است که ممکن است جواب مرحله قبل که همان خوشه‌های کوچک به‌عنوان مجموعه‌داده‌های پرت بود، بهترین مجموعه‌جواب برای پایان کار نباشد و این احتمال نیز وجود داشته باشد که خوشه‌ای کوچک باشد، اما این خوشه نرمال باشد. به خاطر این‌که به‌صورت محض گفته نشود که همهٔ خوشه‌های کوچک پرت هستند، لایه‌ای به الگوریتم اضافه می‌شود که در درون خوشه‌های کوچک نیز خوشه یا خوشه‌هایی که نسبت به بقیه فاصله بیشتری دارند، شناسایی شود (مثل‌اینکه با یک مسأله شناسایی داده‌های پرت ثانویه روبه‌رو باشیم) و این خوشه‌ها به داده‌های معمولی اضافه می‌شود و دوباره به‌دنبال اعضای پرت بیشتری باشیم. در این مرحله فرض این است که اکثریت این داده‌ها پرت و اقلیت این داده‌ها داده‌های نرمال هستند، درست عکس حالت مرحله قبل.



(شکل-۶): الگوریتم انتخاب خوشه‌های کاندید داده‌های پرت
(Figure-6): cluster selection algorithm for outlier data

در ابتدای این مرحله به‌ازای داده‌هایی که نامزد پرت‌بودن هستند، یک الگوریتم KMeans افرازی انجام می‌شود که خروجی آن دو خوشه باشد. یکی از این خوشه‌ها باید به مجموعه‌داده‌های نرمال اضافه شود. این خوشه به این نحو انتخاب می‌شود که به‌ازای هردوی این خوشه‌ها، الگوریتم K نزدیک‌ترین همسایه به‌ازای کل مجموعه‌داده اجرا می‌شود تا k همسایه نزدیک داده‌های هر خوشه که نامزد پرت‌بودن به‌دست آید. در جدول (۴) می‌توان نتیجه اجرای این الگوریتم را مشاهده کرد. (در جدول (۴) منظور از Od_1، دادهٔ پرت شماره ۱ و Nd_1 دادهٔ نرمال شماره ۱ است).

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]
[DOI: 10.52547/jsdp.19.3.87]



(شکل-۷): اجرای شماتیک الگوریتم پیشنهادی

(Figure-7): implementation of the proposed algorithm Scheme

(جدول-۴): اجرای الگوریتم K نزدیک‌ترین همسایه مرتبه اول

(Table-4): execution K nearest neighbor algorithm first-order

ع.ر	داده پرت	همسایه اول	همسایه دوم	همسایه سوم	همسایه چهارم	همسایه پنجم
۱	Od_1	Nd_1	Nd_3	Nd_100	Nd_120	Nd_150
۲	Od_2	Nd_3	Nd_170	Nd_1	Nd_77	Nd_100
۳	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
N	Od_n	Nd_1	Nd_77	Nd_100	Nd_3	Nd_100

(جدول-۵): مجموعه همسایگان متمایز خوشه

(Table-5): A set of distinct neighbors of the cluster

ع.ر	داده پرت
1	Nd_105
2	Nd_2242
3	Nd_43190
4	...
...	...
...	...
...	...
K	Nd_n

(جدول-۶): مجموعه همسایگان متمایز خوشه ۲

(Table-6): A set of distinct neighbors of the cluster

ع.ر	داده پرت
1	Nd_324
2	Nd_56320
3	Nd_112768
4	...
...	...
...	...
...	...
K´	Nd_n

در این مرحله به‌ازای هر یک از دو خوشه یک جدول داریم که همسایه‌های داده‌های هر خوشه را نشان می‌دهد. به جدول‌های (۵ و ۶) که هر یک متعلق به یک خوشه است، توجه کنید. به‌ازای هر یک از دو خوشه روابط (۴) و (۵) را حساب می‌کنیم. به‌طوری‌که K تعداد همسایه

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]

[DOI: 10.52547/jsdp.19.3.87]

متمایز خوشه ۱ ، K' تعداد همسایه متمایز خوشه ۲، N تعداد اعضای خوشه ۱ و N' تعداد اعضای خوشه ۲ باشد:

$$\frac{K}{N} = X \quad (۴)$$

$$\frac{K'}{N'} = X' \quad (۵)$$

کار تقسیم به این دلیل انجام می‌شود که مقایسه به‌صورت منصفانه باشد؛ زیرا تعداد اعضای دو خوشه برابر نیستند. تا به این جای کار دو عدد نرمال به‌ازای دو خوشه به‌دست آورده شد. کوچک یا بزرگ‌بودن این اعداد نسبت به هم دارای معنا است. برای مثال عدد کوچک‌تر نشان‌دهنده این است که خوشه متناظر با عدد به داده‌های نرمال نزدیک‌تر است و بالعکس. پس این دو عدد باهم مقایسه می‌شود و خوشه‌ای متناظر با عدد کوچک‌تر به مجموعه داده‌های نرمال اضافه و خوشه متناظر با عدد بزرگ‌تر در الگوریتم پیشنهادی با قطعیت داده‌های پرت در نظر گرفته می‌شود

دلیل ایجاد این دو خوشه در این مرحله این بود که می‌خواستیم داده‌های پرت که در اینجا اکثریت داده‌ها را شامل می‌شوند از داده‌هایی که از آن‌ها فاصله دارند و در اقلیت هستند، جدا شوند. در شکل (۷) می‌توان مراحلی را که تا اینجا کار انجام شده، به‌صورت شماتیک دید. (فرض شده که تعداد خوشه‌ها برای اجرای الگوریتم KMeans سلسله‌مراتبی به‌دست آورده شده است). بعدازاین قسمت به‌دلیل اینکه در نخستین مرحله از کار الگوریتم، درصد داده‌های پرت به‌عنوان ورودی الگوریتم از کاربر گرفته‌شده بود و اینکه در مرحله آخر این قسمت بخشی از داده‌های کاندیدا برای پرت‌بودن به مجموعه‌داده‌های معمولی برگردانده شد، در اینجا با روشی به مجموعه‌داده‌های پرت داده‌هایی اضافه می‌شود که به‌احتمال بیشتری داده پرت هستند. در این مرحله نیز از الگوریتم K همسایه نزدیک (KNN) استفاده شده است. الگوریتم KNN الگوریتمی بانظارت است، اما دلیلی که در اینجا استفاده شد، این است که کدام داده در مجموعه‌دادهٔ معمولی بیشتر در همسایگی داده‌های پرت قرار می‌گیرد؛ به‌عبارتی‌دیگر همسایه‌هایی از داده‌های پرت که به‌احتمال بیشتری باید در مجموعه‌داده‌های پرت قرار گیرد، پیدا شوند. در اینجا به این نحو عمل می‌شود که الگوریتم KNN برای هر داده‌ای که در درون مجموعه‌داده‌های پرت قرارگرفته در درون خوشه، یکی بعد از بزرگ‌ترین خوشه‌ای که به‌عنوان داده پرت در نظر گرفته‌شده بود، حساب می‌شود.

به‌عنوان‌مثال پنج همسایهٔ نزدیک هر یک از داده‌های پرت را حساب می‌کنیم. به جدول (۷) توجه کنید. در جدول (۷) منظور از Od_1، داده‌ی پرت شماره ۱ و Nd_1 داده‌ی نرمال شماره ۱ است. این جدول نشان می‌دهد که کدام داده‌ی نرمال بیشتر در نزدیکی داده‌های پرت اتفاق افتاده است. به‌ازای هر دادهٔ معمولی که در فیلدهای "همسایه نخست" تا فیلد "همسایه پنجم" قرارگرفته تعداد حساب می‌شود تا جدولی که شامل نام داده و تعداد تکرار آن است، به‌وجود آید. این جدول برحسب تعداد تکرار به‌صورت نزولی مرتب‌سازی می‌شود تا برای مثال جدولی مشابه جدول (۸) حاصل شود.

(جدول-۷): اجرای الگوریتم K نزدیک‌ترین همسایه مرتبه دوم
(Table-7): execution K nearest neighbor algorithm second-order

ردیف	داده پرت	همسایه اول	همسایه دوم	همسایه سوم	همسایه چهارم	همسایه پنجم
1	Od_1	Nd_1	Nd_3	Nd_100	Nd_120	Nd_150
2	Od_2	Nd_3	Nd_170	Nd_1	Nd_77	Nd_100
3	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
N	Od_n	Nd_1	Nd_77	Nd_100	Nd_3	Nd_100

(جدول-۸): پیداکردن نزدیک‌ترین همسایه به داده‌های پرت
(Table-8): Finding the nearest neighbor to outlier data

ردیف	داده معمولی	تعداد تکرار
1	Nd_1	120
2	Nd_3	113
3	...	...
...	...	...
...	...	...
...	...	...
N	Nd_n	20

از این جدول به تعدادی که در مرحله پیش از داده‌های پرت حذف‌شده بود به همان تعداد از ابتدای این جدول، دادهٔ معمولی انتخاب می‌شود و به مجموعه‌داده‌های پرت اضافه می‌شود. به‌این‌ترتیب می‌توان تضمین کرد که به همان درصد مشخص‌شده توسط کاربر دادهٔ پرت از مجموعه‌داده انتخاب شده است.

۴- پیاده‌سازی و ارزیابی نتایج

۴-۱- مجموعه‌داده‌ها

یکی از موانعی که در انجام پروژه‌ها و پژوهش‌های داده‌کاوی با استفاده از داده‌های بومی داخلی کشور در

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]
[DOI: 10.52547/jsdp.19.3.87]

پیش روی پژوهش‌گران قرار دارد، دست‌یابی به داده‌ها و پایگاه داده‌های اطلاعاتی است. در اغلب موارد سازمان‌ها و نهادهایی که این اطلاعات را در اختیار دارند به دلایل مختلف از ارائه آن خودداری می‌کنند، یکی از دلایل این امر محرمانه و خصوصی‌بودن داده‌هاست. داده‌هایی که برای این پژوهش مورداستفاده قرارگرفته شده، تنها مجموعه‌دادهٔ استاندارد موجود است که در سال ۲۰۱۶ توسط یکی از بانک‌های اروپایی از طریق سایت کاگل[1] در اختیار عموم قرارگرفته شده است.

الف- توصیف داده‌ها:

هنگامی‌که یک تراکنش مالی توسط دارنده کارت در یک ترمینال بانکی از قبیل دستگاه کارت‌خوان، دستگاه خودپرداز، اینترنت بانک، موبایل بانک و کیوسک بانک انجام می‌شود این تراکنش شامل ویژگی‌هایی مانند مبلغ، مانده، نوع فعالیت، شماره ترمینال، شماره کارت مبدأ، شماره کارت مقصد، شماره سند، تاریخ سند، کد شعبه و ... هستیم، اما به‌دلایل مختلف از قبیل محرمانه و خصوصی‌بودن داده‌ها از توضیح و دادن مقادیر اصلی در این مجموعه‌دادهٔ استاندارد جلوگیری شده است. از ۳۱ ویژگی‌ای که در این مجموعه‌داده وجود دارد، تنها ۳ ویژگی دارای مقادیر قابل توضیح هستند و مابقی ویژگی‌ها از طریق الگوریتم PCA نرمال شده‌اند. سه ویژگی که دارای مقادیر مشخص هستند در جدول (۹) مشخص‌شده است.

به‌دلیل اینکه الگوریتم پیشنهادشده الگوریتمی بدون نظارت است و برای زمانی کاربرد دارد که داده‌ها برچسب ندارند، برای ارزیابی الگوریتم پیشنهادشده این ویژگی حذف شده و ویژگی‌ای به نام شناسه تراکنش به مجموعه‌داده اضافه شده است. این ویژگی برای زمانی که می‌خواهیم دقت و فراخوانی محاسبه شود، کاربرد دارد.

(جدول-۹): ویژگی‌های مشخص در مجموعه‌دادهٔ استاندارد

(Table-9): Specific features in the standard dataset

ردیف	ویژگی	توضیحات
1	Amount	مبلغ
2	Time	زمان
3	Class	برچسب فراد یا نرمال بودن تراکنش
4	V1 - V28	ویژگی‌هایی که توسط ارائه‌دهنده دیتاست نرمال شده‌اند.

[1] kaggle

۲-۴- پیش‌پردازش داده‌ها برای اجرای الگوریتم

قبل از اجرای الگوریتم پیشنهادی بایستی داده‌های اولیه ازنظر ویژگی و مقادیر آماده‌سازی شوند. به‌دلیل اینکه ۲۸ ویژگی این مجموعه‌داده نرمال شده است، برای آماده‌سازی داده‌ها تنها دو ویژگی باقی‌مانده را نرمال می‌کنیم تا همه ویژگی‌ها نرمال شده باشند. نرمال‌سازی درواقع بحث هم‌مقیاس‌سازی داده‌ها است. به‌طوراساسی دامنهٔ برخی ویژگی‌ها با یکدیگر متفاوت است و نادیده‌گرفتن این تفاوت در دامنه‌ها سبب بروز اختلاف جدی در محاسبات خواهد شد. ویژگی مبلغ خرید و زمان بایستی نرمال شود. در شکل (۸) مجموعه‌داده بدون هیچ تغییری آورده شده و در شکل (۹) مجموعه‌داده بعد از تغییرات قابل‌مشاهده است.

```
select * from CREDIT_CARD t
```

	TIME_FRAUD	AMOUNT	CLS	V1	V2	V3	V4	V5
1	0	149.62	0	-1.3598071336738	-0.0727811733098497	2.53634673796914	1.37815522427443	-0.33
2	0	2.69	0	1.19185711131486	0.26615071205963	0.16648011335321	0.448154078460911	0.060
3	1	378.66	0	-1.35835406159823	-1.34016307473609	1.77320934263119	0.379779593034328	-0.50
4	1	123.5	0	-0.966271711572087	-0.185226008082898	1.79299333957872	-0.863291275036453	-0.010
5	2	69.99	0	-1.15823309349523	0.877736754848451	1.548717846511	0.403033933955121	-0.40
6	2	3.67	0	-0.425965884412454	0.960523044882985	1.14110934232219	-0.168252079760302	0.4
7	4	4.99	0	1.22965763450793	0.141003507049326	0.045370773589449	1.20261273673594	0.19
8	7	40.8	0	-0.644269442348146	1.41796354547385	1.0743803763556	-0.492199018495015	0.94
9	7	93.2	0	-0.89428608220282	0.286157196276544	-0.113192212729871	-0.271526130088604	2.4
10	9	3.68	0	-0.33826175242575	1.11959337641566	1.04436655157316	-0.222187276738296	0.4
11	10	7.8	0	1.44904378114715	-1.17633882535966	0.913859832832795	-1.37566665499943	-1.9
12	10	9.99	0	0.38497821518095	0.616109459176472	-0.874299702595052	-0.094018625967911 5	2.9
13	10	121.5	0	1.249998742053	-1.22163680921816	0.383930151282291	-1.23489868766892	-1.4
14	11	27.5	0	1.0693735878819	0.287722129331455	0.828612726634281	2.71252042961718	-0.17
15	12	58.8	0	-2.7918547659339	-0.327770756658658	1.64175016056605	1.76747274389883	-0.13
16	12	15.99	0	-0.752417042956605	0.345485415344747	2.05732291276727	-1.46864329840046	-1.
17	12	12.99	0	1.10321543528383	-0.0402962145973447	1.2673320885949	1.28909146962552	-0.73
18	13	0.89	0	-0.436905071360625	0.918966212909322	0.92459077438817	-0.727219053596792	0.91
19	14	46.8	0	-5.40125766315825	-5.45014783420644	1.18630463143652	1.73623880012095	3.0
20	15	5	0	1.4929359769862	-1.02934573189487	0.45479473374366	-1.43802587991702	-1.5
21	16	231.71	0	0.694884775607337	-1.36181910308009	1.02922103956032	0.834159299216716	-1.1
22	17	34.09	0	0.962496069914852	0.32846102605212	-0.17147905415064	2.10920406774016	1.1

(شکل-۸): مجموعه دادهٔ استاندارد اولیه

(Figure-8): Basic standard dataset

```
select * from CREDIT_CARD_WITHOUT_LABLE_NORM t
```

	ID	TIME_FRAUD_NORM	AMOUNT_NORM	V1	V2	V3
1	1	-1.99657951830322	-0.353228772845648	-1.3598071336738	-0.0727811733098497	2.53634673796
2	2	-1.99657951830322	-0.353228772845648	1.19185711131486	0.26615071205963	0.16648011335
3	3	-1.99655846041646	-0.349230693670656	-1.35835406159823	-1.34016307473609	1.77320934263
4	4	-1.99655846041646	-0.349230693670656	-0.966271711572087	-0.185226008082898	1.79299333957
5	5	-1.9965374025297	-0.345232614495664	-1.15823309349523	0.877736754848451	1.548717846
6	6	-1.9965374025297	-0.345232614495664	-0.425965884412454	0.960523044882985	1.14110934232
7	7	-1.99649528675617	-0.337235456145679	1.22965763450793	0.141003507049326	0.045370773589
8	8	-1.99643211309589	-0.325242218620702	-0.644269442348146	1.41796354547385	1.0743803763
9	9	-1.99643211309589	-0.325242218620702	-0.89428608220282	0.286157196276544	-0.113192212729
10	10	-1.99638999732237	-0.317246060270718	-0.33826175242575	1.11959337641566	1.04436655157
11	11	-1.99636893943561	-0.313247981095725	1.44904378114715	-1.17633882535966	0.913859832832
12	12	-1.99636893943561	-0.313247981095725	0.38497821518095	0.616109459176472	-0.874299702595
13	13	-1.99636893943561	-0.313247981095725	1.249998742053	-1.22163680921816	0.383930151282
14	14	-1.99634788154885	-0.309249901920733	1.0693735878819	0.287722129331455	0.828612726634
15	15	-1.99632682366209	-0.305251822745741	-2.7918547659339	-0.327770756658658	1.64175016056
16	16	-1.99632682366209	-0.305251822745741	-0.752417042956605	0.345485415344747	2.05732291276
17	17	-1.99632682366209	-0.305251822745741	1.10321543528383	-0.0402962145973447	1.26733208859
18	18	-1.99630576577533	-0.301253743570748	-0.436905071360625	0.918966212909322	0.92459077438
19	19	-1.99628470788857	-0.297255664395756	-5.40125766315825	-5.45014783420644	1.18630463143
20	20	-1.9962636500018	-0.293257585220764	1.4929359769862	-1.02934573189487	0.45479473374
21	21	-1.99624259211504	-0.289259506045771	0.694884775607337	-1.36181910308009	1.02922103956
22	22	-1.99622153422828	-0.285261426870779	0.962496069914852	0.32846102605212	-0.17147905415

(شکل-۹): مجموعه داده استاندارد بعد از نرمال کردن و استخراج ویژگی‌ها

(Figure-9): Standard dataset after normalizing and extracting features

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]

[DOI: 10.52547/jsdp.19.3.87]

۳-۴- آزمایش‌های انجام‌شده جهت ارزیابی الگوریتم پیشنهادی

اکنون الگوریتم معرفی‌شده را بر داده‌های استاندارد بانکی اعمال کرده و تراکنش‌های مشکوک به تقلب را که درواقع همان داده‌های پرت هستند، شناسایی می‌شود. در ادامه نحوهٔ پیاده‌سازی و مراحل آن توضیح داده خواهد شد.

در این بخش، آزمایش‌های انجام‌شده جهت ارزیابی دقت و فراخوانی الگوریتم پیشنهادی توضیح داده خواهد شد. برای ارزیابی الگوریتم پیشنهادی، دو مرحله آزمایش انجام شده است.

۱. آزمایش نخست با ویژگی‌های حاصل از مجموعه‌دادهٔ استاندارد اولیه
۲. آزمایش دوم با ۱۰ ویژگی حاصل از الگوریتم NMF و ۲۰ خوشه

بعدازاین دو مرحله آزمایش به مقایسه الگوریتم پیشنهادی با الگوریتم‌های استاندارد در زمینه تشخیص داده‌های پرت می‌پردازیم.

۱-۳-۴- آزمایش نخست

ویژگی‌های مورداستفاده‌شده: ویژگی‌های مجموعه‌دادهٔ استاندارد اولیه

شرح آزمایش: تمامی ویژگی‌های مجموعه‌داده‌های استاندارد به‌عنوان ورودی به الگوریتم KMeans سلسله‌مراتبی داده می‌شود. در اینجا باید تعداد خوشه مناسب برای ادامه کار پیدا شود. همان‌طور که در فصل سوم توضیح داده شد، روند کار به این صورت است که برای شروع کار، KMean سلسله‌مراتبی با پنج خوشه اجرا می‌شود و به‌دنبال تعداد خوشه‌های مناسب خواهیم بود که تعداد خوشه‌های کوچک آن۵۰ $\pm$ ۴۹۲ باشد. اجرای الگوریتم ادامه داده می‌شود تا به این تعداد برسیم. در جدول (۱۰) اجراهای مختلف انجام‌شده برای این آزمایش آورده شده است. قابل‌توجه است که خوشه‌ها بعد از اجرای الگوریتم KMean سلسله‌مراتبی مرتب به‌صورت صعودی مرتب می‌شوند. در اینجا اعداد نمایان‌گر تعداد اعضای خوشه‌ها از کوچک به بزرگ هستند.

بعدازاینکه تعداد خوشه‌های مناسب برای اجرای الگوریتم Kmeans سلسله‌مراتبی انجام شد، مرحله تحلیل خوشه‌های کوچک انجام می‌شود. بعد از اجرای این مرحله دقت و فراخوانی به‌دست‌آمده به‌ترتیب ۰/۱۲ درصد و ۰/۱۰۵ درصد است.

(جدول-۱۰): مشخص‌کردن تعداد خوشه‌های مناسب برای کل مجموعه‌داده استاندارد

(Table-10): Specify the number of suitable clusters for the entire standard dataset

ردیف	تعداد خوشه	تعداد اعضای خوشه‌های کوچک	شرط
1	5	17820	×
2	10	10258	×
3	15	5655	×
4	20	3250	×
5	25	1750	×
6	30	1700	×
۷	35	1700	×
8	40	14+24+1385	×
9	45	7+8+23+682	×
10	50	4+4+7+10+23+607	×
11	55	3+3+4+5+10+10+13 +568	×
12	60	1+2+3+4+5+10+10+13+60+604	×
13	65	1+1+2+3+4+5+10+10+13+30+35+495	×
14	70	1+1+1+2+2+4+5+10+10+13+30+35+149+495	×
15	75	1+1+1+2+2+4+5+8+10+10+13+17+18+18+24+48+58+187+495	×
16	76	1+1+1+2+2+4+5+7+10+10+13+17+18+18+24+29+30+97+178	♡

۲-۳-۴- آزمایش دوم

ویژگی‌های مورداستفاده‌شده: ده ویژگی حاصل از الگوریتم NMF.

شرح آزمایش: ده ویژگی حاصل از الگوریتم NMF به‌عنوان ورودی به الگوریتم KMeans سلسله‌مراتبی داده می‌شود. در اینجا نیز باید تعداد خوشه مناسب برای ادامه کار پیدا شود. همان‌طور که در فصل قبل توضیح داده شد، روند کار به این صورت است که برای شروع کار، KMean سلسله‌مراتبی با پنج خوشه اجرا می‌شود و به دنبال تعداد خوشه‌ای مناسب خواهیم بود که تعداد خوشه‌های کوچک آن ۵۰ $\pm$ ۴۹۲ باشد. اجرای الگوریتم ادامه داده می‌شود تا به این تعداد برسیم. در جدول (۱۱) اجراهای مختلف انجام‌شده برای این آزمایش آورده شده است. قابل‌توجه است که خوشه‌ها بعد از اجرای الگوریتم KMean سلسله‌مراتبی مرتب به‌صورت صعودی مرتب می‌شوند. در اینجا اعداد نمایان‌گر تعداد اعضای خوشه‌ها از کوچک به بزرگ هستند.

تعداد خوشه بیست، جمع اعضایش ۵۳۴ می‌شود که نخستین مرحله‌ای بود که شرط خاتمه پیداکردن تعداد خوشه‌ها را داشت. بعدازاینکه تعداد خوشه‌های مناسب برای اجرای الگوریتم Kmeans سلسله‌مراتبی انجام شد،

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]
[DOI: 10.52547/jsdp.19.3.87]

مرحله تحلیل خوشه‌های کوچک انجام می‌شود. بعد از اجرای این مرحله دقت و فراخوانی به‌دست‌آمده به‌ترتیب ۰/۶۲ و ۰/۵۶۷ است.

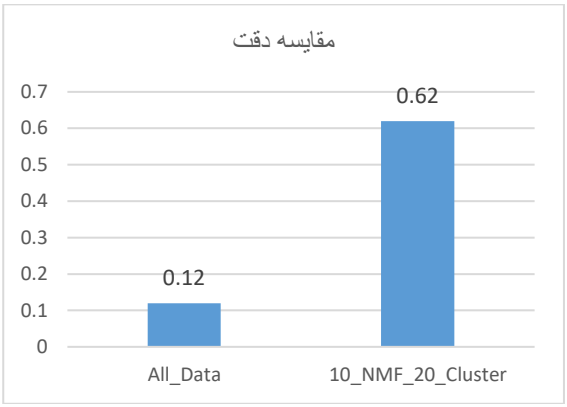
(جدول-۱۱): مشخص‌کردن تعداد خوشه‌های مناسب برای مجموعه داده استاندارد NMF ۱۰

(Table-11): Specify the number of suitable clusters for the standard NMF 10 dataset

	تعداد خوشه	جمع تعداد اعضای خوشه‌های کوچک	شرط
1	5	36506	×
2	10	11678	×
3	15	259 + 558 + 2276	×
7	19	10 + 85 + 108 + 154 + 461	×
8	20	11 + 85 + 103 + 153 + 182	♡

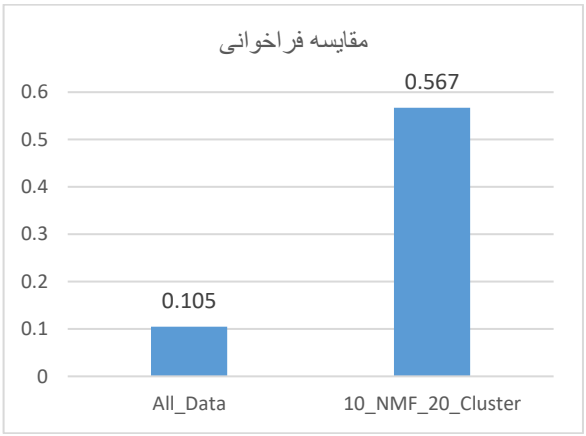
۴-۳-۳- مقایسه دقت آزمایش‌ها

در قسمت‌های قبلی مقادیر دقت به‌ازای دو مجموعه داده در ۲ آزمایش به‌دست آورده شد. در شکل (۱۰) می‌توان نتیجه این دو آزمایش را باهم مقایسه کرد.



(شکل-۱۰): مقایسه دقت آزمایش ۱ و آزمایش ۲

(Figure-10): Comparison of the accuracy of Experiment 1 and Experiment 2



(شکل-۱۱): مقایسه فراخوانی آزمایش ۱ و آزمایش ۲

(Figure-11): Comparison of the calling of Experiment 1 and Experiment 2

همان‌طور که مشاهده می‌شود، در موردی که روی داده‌های اولیه الگوریتم NMF اجراشده، نتیجه به‌مراتب بهتر از حالتی است که داده‌های اولیه استفاده شده است.

۴-۴- مقایسه فراخوانی آزمایش‌ها

در قسمت‌های قبلی مقادیر فراخوانی به‌ازای دو مجموعه داده در ۲ آزمایش به‌دست آورده شد. در شکل (۱۱) می‌توان نتیجه این دو آزمایش را باهم مقایسه کرد.

همان‌طور که می‌بینید در موردی که روی داده‌های اولیه الگوریتم NMF اجرا شده، نتیجه به‌مراتب بهتر از حالتی است که داده‌های اولیه استفاده شده است.

۴-۴-۱- مقایسه دقت و فراخوانی الگوریتم پیشنهادی با دیگر الگوریتم‌ها

هدف در این بخش، ارزیابی دقت و فراخوانی الگوریتم‌های CBLOF، Isolation Forest، ABOD، LOF، K-Nearest Neighbors، Median kNN، Average kNN، One Class SVM، Auto Encoder، Feature Bagging در مقایسه با الگوریتم پیشنهادی است. برای ارزیابی این الگوریتم‌ها دو آزمایش انجام شده که شامل موارد زیر است:

۱. آزمایش نخست با ویژگی‌های مجموعه‌دادهٔ استاندارد اولیه

۲. آزمایش دوم با ده ویژگی حاصل از الگوریتم NMF

آزمایش نخست

ویژگی‌های مورداستفاده شده: ویژگی‌های مجموعه‌دادهٔ استاندارد اولیه

شرح آزمایش: تمامی ویژگی‌های مجموعه داده‌های استاندارد به‌عنوان ورودی به ده الگوریتم یادشده در بالا داده می‌شود و دقت و فراخوانی برای این ده الگوریتم محاسبه می‌شود. مقدار محاسبه‌شده به‌ازای همه الگوریتم‌ها در جدول (۱۲) آورده شده است.

(جدول-۱۲): اجرای الگوریتم‌ها بر روی مجموعه داده استاندارد

(Table-12): execution of the algorithms based on the standard dataset

ردیف	نام الگوریتم	دقت	فراخوانی	F-Score
1	Clustering-Based Local Outlier Factor(CBLOF)	0.183	0.183	0.183
2	Isolation Forest	0.341	0.341	0.341

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]

[DOI: 10.52547/jsdp.19.3.87]

0.000	0.000	0.000	Angle-based Outlier Detector (ABOD)	3
0.000	0.000	0.000	Local Outlier Factor (LOF)	4
0.044	0.095	0.081	K Nearest Neighbors (KNN)	5
0.063	0.063	0.063	OneClassSVM	6
0.232	0.232	0.232	AutoEncoder	7
0.03	0.069	0.053	Median kNN (M-KNN)	8
0.062	0.062	0.030	Average kNN (A-KNN)	9
0.193	0.193	0.193	Feature Bagging	10

آزمایش دوم

ویژگی‌های مورداستفاده‌شده: ده ویژگی حاصل از الگوریتم NMF

شرح آزمایش: ده ویژگی حاصل از الگوریتم NMF به‌عنوان ورودی به ده الگوریتم یادشده در بالا داده و دقت و فراخوانی برای این ده الگوریتم محاسبه می‌شود. مقدار محاسبه‌شده به‌ازای همه الگوریتم‌ها در جدول (۱۳) آورده شده است.

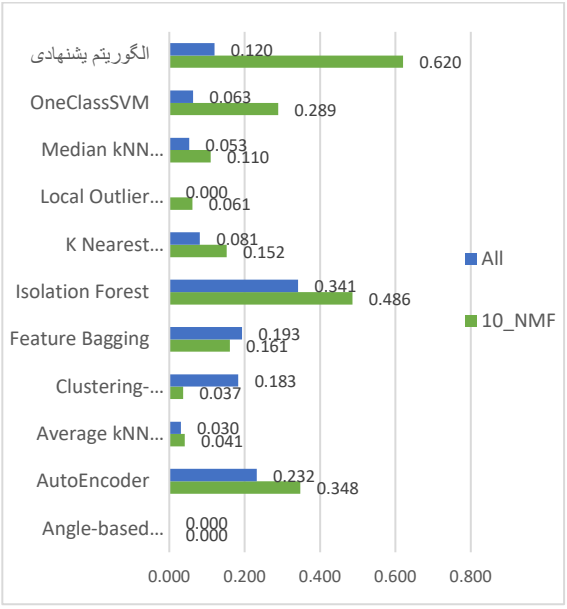
(جدول-۱۳): اجرای الگوریتم‌ها بر روی مجموعه‌داده استاندارد 10 NMF

(Table-13): execution of the algorithms based on the standard 10-NMF dataset

ردیف	نام الگوریتم	دقت	فراخوانی
1	Clustering-Based Local Outlier Factor(CBLOF)	0.037	0.037
2	Isolation Forest	0.486	0.486
3	Angle-based Outlier Detector (ABOD)	0.000	0.000
4	Local Outlier Factor (LOF)	0.061	0.074
5	K Nearest Neighbors (KNN)	0.152	0.188
6	OneClassSVM	0.289	0.289
7	AutoEncoder	0.348	0.348
8	Median kNN (M-KNN)	0.110	0.165
9	Average kNN (A-KNN)	0.041	0.142
10	Feature Bagging	0.161	0.161

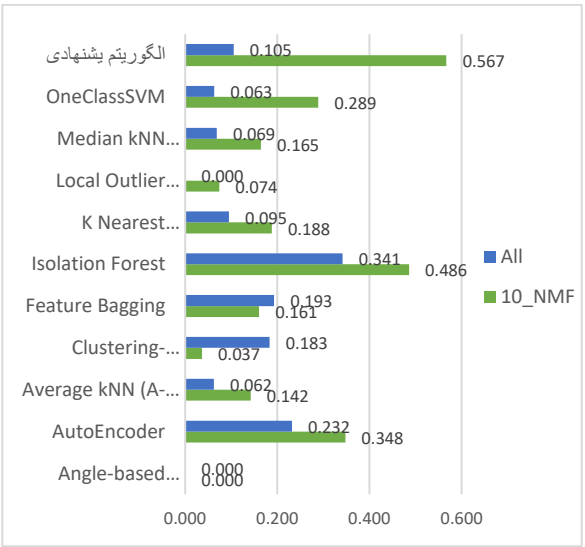
۲-۴-۴- مقایسه دقت بین الگوریتم پیشنهادی و دیگر الگوریتم‌ها

در شکل (۱۲) مقایسه دقت بین الگوریتم‌های مختلف که اطلاعات آن‌ها در جدول‌های (۱۲ و ۱۳) و آزمایش‌های ابتدای این بخش برای الگوریتم‌های مختلف به‌دست‌آمده، آورده شده است. همان‌گونه که در نتایج به‌دست‌آمده از آزمایش‌ها و شکل (۱۲) قابل‌مشاهده است، در این ده الگوریتم، بهترین نتیجه برای الگوریتم Isolation Forset است که دقت آن ۰/۴۸۶ است. این مقادیر نسبت به مقادیری که در آزمایش الگوریتم پیشنهادی به دست آورده شد پایین‌تر است.



(شکل-۱۲): مقایسه دقت الگوریتم پیشنهادی با دیگر الگوریتم‌ها

(Figure-12): Comparison of the accuracy of the proposed algorithm with other algorithms



(شکل-۱۳): مقایسه فراخوانی الگوریتم پیشنهادی با دیگر الگوریتم‌ها

(Figure-13): Comparison of the calling of the proposed algorithm with other algorithms

۳-۴-۴- مقایسه فراخوانی بین الگوریتم پیشنهادی و دیگر الگوریتم‌ها

در شکل (۱۳) مقایسه فراخوانی بین الگوریتم‌های مختلف که اطلاعات آن‌ها در جدول‌های (۱۲ و ۱۳) و آزمایش‌های ابتدای این بخش، برای الگوریتم‌های مختلف، آورده شده است. همان‌گونه که در نتایج به‌دست‌آمده از آزمایش‌ها و

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]

[DOI: 10.52547/jsdp.19.3.87]

شکل (۱۳) قابل‌مشاهده است، در این ده الگوریتم، بهترین نتیجه برای الگوریتم Isolation Forset است که فراخوانی آن ۰/۴۸۶ است. این مقادیر نسبت به مقادیری که در آزمایش الگوریتم پیشنهادی به‌دست آورده‌شد پایین‌تر است.

۵- بحث و نتیجه‌گیری

نظام بانکی با تکیه‌بر فناوری‌های نوظهور سعی بر ارائه خدمات متنوع به مشتریان خود دارد. یکی از این خدمات، ارائه کارت‌های عابر بانک به مشتریان است. با گسترش این کارت‌های عابر بانک، انواع تقلب‌های جدید و پیچیده نیز توسط متخلفان افزایش یافته است. به‌دلیل حجم بالای داده‌ها و همچنین وجود انواع تقلب‌های پیچیده، کشف تقلب به‌صورت دستی امکان‌پذیر نیست؛ بنابراین به روش‌های خودکار نیاز است که داده‌های حجیم را تحلیل کرده و بتواند موارد مشکوک به تقلب را استخراج کند. داده‌کاوی می‌تواند یک راه‌حل مناسب برای حل این مشکل باشد. الگوریتم‌های مورداستفاده در داده‌کاوی در سه دسته کلی بانظارت، نیمه‌نظارتی و بدون نظارت قرار می‌گیرند. در روش‌های بانظارت به داده‌های آموزشی بر چسب‌دار که ردهٔ آن‌ها (معمولی، غیرمعمولی) مشخص دارد، نیاز است تا سیستم بتواند با استفاده از آن‌ها آموزش ببیند و مدلی ارائه کند تا بتوانیم به‌صورت خودکار داده‌های معمولی را از داده‌های غیرمعمولی جدا کنیم. در روش‌های نیمه‌نظارتی به‌کل برچسب‌ها دسترسی نداریم و تنها برچسب برخی از داده‌ها مشخص هستند. در این حالت تنها توانایی بازشناسی داده‌هایی را داریم که برچسب‌شان را در اختیار داشته‌ایم. به‌عنوان‌مثال اگر تنها برچسب‌های معمولی را در اختیار داشته باشیم، مدل ما تنها توانایی شناخت داده‌های معمولی را دارد. درنتیجه اگر داده‌ای که می‌خواهیم آن را به مدل بدهیم، اگر مدل آن را یک دادهٔ معمولی تشخیص ندهد، پس آن را یک دادهٔ پرت در نظر می‌گیرد. بر اساس آزمایش‌های انجام‌شده و مقایسه نتایج این آزمایش‌ها، روش پیشنهادی مزایای زیر را دارد:

- ✓ دقت الگوریتم پیشنهادی از دقت الگوریتم‌هایی که با آن‌ها مقایسه انجام شد، بالاتر است.
- ✓ فراخوانی الگوریتم پیشنهادی از فراخوانی الگوریتم‌هایی که با آن‌ها مقایسه انجام شد، بالاتر است.

در ضمن به‌علت اینکه الگوریتم پیشنهادشده کار خود را بر روی همه مجموعه‌داده انجام نمی‌دهد و تنها با حجم کمی از داده‌ها کار می‌کند از سرعت بالایی برخوردار است. به‌خصوص هنگامی‌که تعداد داده‌ها و یا تعداد ویژگی‌ها افزایش می‌یابد، این اختلاف سرعت نیز بالا خواهد رفت.

الگوریتم پیشنهادی با الگوریتم‌های CBLOF، Isolation Forest، ABOD، LOF، K Nearest Neighbors، Median kNN، Average kNN، One Class SVM، Auto Encoder، Feature Bagging مقایسه شد. در آزمایش‌هایی که در این فصل انجام شد، مشخص شد دقت و فراخوانی الگوریتم پیشنهادی به‌ترتیب ۰/۶۲ و ۰/۵۶۷ است و دقت و فراخوانی بهترین الگوریتم(Isolation Forest) از دیگر الگوریتم‌ها به‌ترتیب ۰/۴۸۶ و ۰/۴۸۶ است؛ که نشان‌دهنده این است که الگوریتم پیشنهادشده از دقت و فراخوانی بالاتری نسبت به مابقی الگوریتم‌ها برخوردار است.

۶- مراجع 6- References

[1] V. V. Vlasselaer, C. Bravo, O. Caelen, T. Eliassi-Rad, L. Akoglu, M. Snoeck, and B. Baesens, "APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions." *Decision Support Systems*, vol.75, pp. 38-48, July 2015.

[2] A. Srivastava, A. Kundu, S. Sural and A. Majumdar, "Credit Card Fraud Detection Using Hidden Markov Model," *in IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 1, pp. 37-48, Jan.-March 2008.

[3] M. J. Zaki, and M. Jr. Wagner, *Data mining and analysis: fundamental concepts and algorithms.* Cambridge University Press, 2014.

[4] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques.* Elsevier, 2011.

[5] D. M. Hawkins, *Identification of outliers*. Springer, 1980

[6] C. C. Aggarwal, *An introduction to outlier analysis*, in Outlier analysis. Springer. p. 1-34, 2017.

[۷] حریرچی گ، "تشخیص تقلب کارت‌های اعتباری با تمرکز بر تشخیص داده‌های پرت"، پایان‌نامه کارشناسی ارشد، دانشگاه صنعتی امیرکبیر، تهران، ایران، ۱۳۹۳

[7] G. Harirchi, "Credit card fraud detection with a focus on outbound data detection", M.S. thesis, Amirkabir Univ., Tehran, Iran, 2015

[8] F. Carcillo, Y. L. Borgne, O. Caelen and G. Bontempi, "An Assessment of Streaming Active Learning Strategies for Real-Life Credit Card

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]

[DOI: 10.52547/jsdp.19.3.87]

Fraud Detection," 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 631-639, Tokyo, 2017.

[9] N. Malini and M. Pushpa, "Analysis on credit card fraud identification techniques based on KNN and outlier detection," *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai*, 2017, pp. 255-258.

[10] Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and H. George Chen, "Ecod: unsupervised outlier detection using empirical cumulative distribution functions", *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[11] Y. Zhao, X. Hu, Cheng Cheng, C. Wang, Ch. Wan, W. Wang, J. Yang, H. Bai, Zh. Li, C. Xiao, Y. Wang, Zhi Qiao, J. Sun, and L. Akoglu, "Suod: accelerating large-scale unsupervised heterogeneous outlier detection", *Proceedings of Machine Learning and Systems*, 2021.

[12] Y. Almardeny, N. Boujnah, and F. Cleary, "A novel outlier detection method for multivariate data", *IEEE Transactions on Knowledge and Data Engineering*, 2020.

سید مرتضی سیدرضایی، مدرک کارشناسی و کارشناسی ارشد خود را به‌ترتیب در دانشگاه فنی و حرفه‌ای شمسی پور و دانشگاه آزاد واحد علوم و تحقیقات در رشته مهندسی نرم‌افزار دریافت کرد. در حال حاضر او در شرکت خدمات انفورماتیک مشغول به کار است.

نشانی رایانامه ایشان عبارت است از:

smortezasr@gmail.com

قربان خردمندیان دارای مدرک دکترای هوش مصنوعی از دانشگاه امیرکبیر است. در حال حاضر وی به‌عنوان متخصص داده‌کاوی در شرکت داده‌کاوان هوشمند توسن مشغول به کار است.

نشانی رایانامه ایشان عبارت است از:

kheradmand@aut.ac.ir

سیدجواد کاظمی تبار دوره کارشناسی خود را در سال ۱۳۸۲ در دانشگاه صنعتی شریف به پایان رسانده است. در سال ۱۳۸۷ وی مدرک دکترای خود را در رشته مخابرات در دانشگاه کالیفرنیا در شهر ارواین کسب کرد و تا سال ۱۳۹۳ در شرکت‌های مختلف مهندسی در آمریکا به فعالیت پرداخت. از جمله از سال ۱۳۹۱ تا ۱۳۹۳ در سیلیوکون ولی به‌عنوان متخصص داده‌کاوی در شرکت گاردین آنالیتیکس به کشف تقلب‌های بانکی کمک می‌کرد. از سال ۱۳۹۴ وی عضو هیأت علمی دانشگاه صنعتی نوشیروانی بابل است.

نشانی رایانامه ایشان عبارت است از:

j.kazemitabar@nit.ac.ir

[Downloaded from jsdp.rcisp.ac.ir on 2026-09-15]

[DOI: 10.52547/jsdp.19.3.87]