"نوآوری‌های فناوری اطلاعات و ارتباطات کاربردی"
سال چهارم، شماره ۱، بهار ۱۴۰۴، ص ۷۰-۵۱

روشی برای بهبود دقت شناسایی کدهای نابسامان با استفاده از یادگیری گروهی پشته سازی

مهدی عبدنیا[۱]، علی کریمی [۲*]، فرهاد کریمی[۳]

۱- کارشناسی ارشد، ۲- استادیار، ۳- دانشجوی دکتری، دانشگاه جامع امام حسین (ع)، تهران، ایران

(دریافت: ۱۴۰۳/۱۲/۱۵، بازنگری: ۱۴۰۴/۰۱/۳۱، پذیرش: ۱۴۰۴/۰۲/۱۶، انتشار: ۱۴۰۴/۰۳/۰۷)

چکیده

با گسترش کاربرد فناوری اطلاعات در تمامی حوزه‌های زندگی انسان، تولید نرم‌افزارهای باکیفیت، بیش از قبل اهمیت پیداکرده است. عوامل مختلفی وجود دارند که کیفیت نرم‌افزارهای تولیدی را کاهش می‌دهند. یکی از این عوامل، وجود کدهای نابسامان است. آن‌ها از نقص‌های ساختاری برنامه‌های نرم‌افزاری محسوب می‌شوند که اغلب به دلیل پیاده‌سازی نادرست فرآیندهای مهندسی نرم‌افزار یا عدم تجربه کافی توسعه‌دهندگان نرم‌افزار به وجود می‌آیند. برای رفع این مشکل نیاز است که آن‌ها را شناسایی و سپس با بازآرایی مجدد برنامه، آن‌ها را برطرف کرد. برای این منظور، استفاده از روش‌ها و فنون مناسب و دقیق در زمینه شناسایی کدهای نابسامان، از اهمیت ویژه‌ای برخوردار است. استفاده از فنون و الگوریتم‌های یادگیری ماشین یکی از راه‌حل‌های پیشنهادی و پرکاربرد برای شناسایی این‌گونه کدها است. بنابراین، در این مقاله راه‌حلی جهت بهبود دقت شناسایی کدهای نابسامان شامل؛ ویژگی حسادت، متد طولانی، کلاس داده، کلاس بزرگ، فهرست طولانی پارامترها و گزاره‌های تعویض با استفاده از ترکیب فنون انتخاب ویژگی مجموعه‌ای و یادگیری گروهی پشته سازی ارائه‌شده است. نتایج نهایی حاصل آزمایش‌های مختلف، بیشینه عملکرد ۹۹٪ در معیار دقت را برای برخی از کدهای نابسامان نشان می‌دهد.

کلیدواژه‌ها: کدهای نابسامان، یادگیری ماشین، یادگیری گروهی، فن پشته سازی، انتخاب ویژگی مجموعه‌ای.

A method to detection code smells with Stacking Ensemble Learning

M. Abdnia[1] , M. Karimi[2], F. Karimi[2]*

Assistant Professor, Imam Hossein University, Tehran, Iran

. (Received: 2025/03/05, Revised: 2025/04/20, Accepted: 2025/05/06, Published: 2025/05/28)

Abstract

With the expansion of the use of information technology in all areas of human life, the production of high-quality software has become more important than ever. There are various factors that reduce the quality of produced software. One of these factors is the presence of messy code or code smells. They are structural defects in software programs that often arise due to incorrect implementation of software engineering processes or lack of sufficient experience of software developers. To solve this problem, it is necessary to identify them and then fix them by refactoring the program. For this purpose, the use of appropriate and accurate methods and techniques in the field of identifying messy codes is of particular importance. The use of machine learning techniques and algorithms is one of the proposed and widely used solutions for identifying such codes. Therefore, in this article, a solution to improve the accuracy of identifying messy codes including; The Feature Envy, Long Method, data class, Large Class, Long Parameter List and Switch Statements are presented using a combination of ensemble feature selection and stacking ensemble learning techniques. The final results from various experiments show a maximum performance of 99% in the accuracy benchmark for some code smells.

Keywords: Code smells, Machine learning, Ensemble learning, Stacking technique, Ensemble feature selection.

استناد: عبد نیا ، مهدی، کریمی، علی، کریمی، فرهاد " روشی برای بهبود دقت شناسایی کدهای نابسامان با استفاده از یادگیری گروهی پشته‌سازی " نوآوری‌های فناوری اطلاعات و ارتباطات کاربردی، ۴ (۱)، ۷۰-۵۱، ۱۴۰۴.

ناشر: دانشگاه جام امام حسین(ع).

رایانامه نویسنده مسئول: a.karimi@ihu.ac.ir

۱. مقدمه

گسترش استفاده از فناوری اطلاعات اهمیت تولید نرم‌افزارهای باکیفیت را ضروری کرده است. کیفیت نرم‌افزارهای تولیدشده به عوامل مختلفی بستگی دارد. ازجمله این عوامل، می‌توان رعایت اصول مهندسی نرم‌افزار، طراحی مناسب، کد نویسی صحیح و تجربه کافی توسعه‌دهندگان را نام برد. وجود بعضی از عوامل کیفیت نرم‌افزارهای تولیدی را کاهش می‌دهد که ازجمله این عوامل می‌توان به کدهای نابسامان[1] اشاره کرد. کدهای نابسامان یکی از نقص‌های ساختاری برنامه‌های نرم‌افزاری هستند که اغلب از عدم رعایت اصول مهندسی نرم‌افزار، طراحی نامناسب یا کد نویسی غیراصولی برنامه به وجود می‌آیند. وجود آن‌ها منجر به افزایش رخداد خطا، کاهش خوانایی برنامه و درنهایت کاهش کیفیت نرم‌افزار می‌شود. به همین دلیل نیاز است که آن‌ها را شناسایی و با بازآرایی مجدد برنامه آن‌ها را برطرف نمود [۱].

تاکنون راه‌حل‌های مختلفی جهت شناسایی خودکار کدهای نابسامان ارائه‌شده است که هر یک از آن‌ها، ویژگی‌ها و چالش‌های خاص خود رادارند. یکی از این راه‌حل‌های ارائه‌شده جهت خودکارسازی شناسایی کدهای نابسامان، استفاده از یادگیری ماشین است. ارائه روشی در جهت استفاده مناسب از الگوریتم‌ها و فنون یادگیری ماشین جهت شناسایی دقیق کدهای نابسامان ازجمله چالش‌هایی است که در این زمینه مطرح می‌شود [۲].

در همین راستا در این مقاله، روشی مبتنی بر ترکیب فنون انتخاب ویژگی مجموعه‌ای[2] و یادگیری گروهی پشته سازی[3] جهت شناسایی کدهای نابسامان ارائه‌شده است. بررسی روش پیشنهادی با تمرکز بر شناسایی کدهای نابسامان ویژگی حسادت[4]، متد طولانی[5]، کلاس بزرگ[6]، کلاس داده[7]، فهرست طولانی پارامترها[8] و گزاره‌های تعویض[9] انجام‌شده است.

نوآوری‌های این تحقیق عبارت‌اند از:

۱- ارائه روشی جهت شناسایی کدهای نابسامان با عملکرد مناسب با استفاده از فنون انتخاب ویژگی مجموعه‌ای و فنون یادگیری گروهی پشته سازی.

۲- بهبود دقت در شناسایی کدهای نابسامانِ «فهرست طولانی پارامترها» و «گزاره‌های تعویض».

در ادامه و در بخش ۲، به بیان تحقیقات انجام‌شده درزمینه شناسایی کدهای نابسامان توسط سایر پژوهشگران، پرداخته می‌شود. در بخش ۳، مرور ادبیات تحقیق و مفاهیم اولیه لازم برای درک بهتر روش پیشنهادی ارائه می‌گردد. در بخش ۴، روش پیشنهادی به‌طور کامل تشریح می‌شود. در بخش ۵، نتایج حاصل از آزمایش‌های مختلف برای اثبات کارایی و عملکرد روش پیشنهادی ارائه و موردبررسی قرار می‌گیرد. همچنین، در ادامه این فصل، نتایج حاصل با نتایج مقالات مرتبط مقایسه می‌گردد. در بخش آخر نیز، جمع‌بندی نهایی و شرح کارهای آینده ارائه می‌گردد.

۲. پیشینه تحقیق

تاکنون تحقیقات زیادی در زمینه استفاده از یادگیری ماشین برای شناسایی کدهای نابسامان انجام‌شده است. بررسی این پژوهش‌ها به جهت تشخیص نقاط ضعف و قوت راه‌حل‌های موجود، برای ارائه یک راه‌حل بهتر ضروری است.

فونتانا و همکاران [۳]، در مقاله خود به بررسی مشکلات ابزارها و روش‌های شناسایی کدهای نابسامان پرداختند. آن‌ها استفاده از یادگیری ماشین را به عنوان روشی برای غلبه بر مشکلات راه‌حل‌های موجود، پیشنهاد کردند. در ادامه، با تحلیل ۷۴ پروژه نرم‌افزاری، مجموعه داده‌ای شامل چهار کد نابسامانِ کلاس داده، متد طولانی، کلاس بزرگ، ویژگی حسادت و دو کد نابسامانِ گزاره‌های تعویض و فهرست طولانی پارامترها (به‌عنوان ضمیمه پژوهش) ارائه نمودند. سپس تعدادی الگوریتم یادگیری ماشین را برای شناسایی آن‌ها به کاربردند. نتایج تحقیق آن‌ها نشان می‌دهد که به‌طور میانگین الگوریتم‌های جنگل تصادفی و J48 عملکرد بهتری داشتند.

کریمی و کریمی [۴]، از فن انتخاب ویژگی مجموعه‌ای برای بهبود عملکرد الگوریتم‌های یادگیری ماشین در شناسایی کدهای نابسامانِ ویژگی حسادت و کلاس داده استفاده کردند. نتایج تحقیق آن‌ها نشان می‌دهد که فن انتخاب ویژگی مجموعه‌ای سبب بهبود عملکرد در شناسایی کدهای نابسامان می‌شود. همچنین، این فن نسبت به الگوریتم‌های انتخاب ویژگی پایه بر عملکرد مدل یادگیری ماشین تأثیر بیشتری می‌گذارد.

کائور و کائور [۵]، از ترکیب فنون انتخاب ویژگی مجموعه‌ای و یادگیری گروهی کیسه بندی جهت پیش‌بینی احتمال وجود کدهای نابسامان در پروژه‌های نرم‌افزاری استفاده کردند. نتایج تحقیقات آن‌ها نشان می‌دهد که انتخاب ویژگی ترکیبی می‌تواند سبب بهبود عملکرد در پیش‌بینی وجود کدهای نابسامان شود.

کریمی و خسروی [۶]، از فن یادگیری گروهی رأی‌گیری اکثریت مبتنی بر ترکیب سه الگوریتم یادگیری ماشین درخت

[1] Code smells
[2] Ensemble feature selection
[3] Stacking ensemble learning
[4] Feature envy
[5] Long method
[6] Large class (LG)
[7] Data class (DC)
[8] Long parameters list (LPL)
[9] Switch statements (SS)

تصمیم‌گیری[1]، نزدیک‌ترین همسایه[2] و ماشین بردار پشتیبان[3] برای شناسایی کدهای نابسامانِ کلاس داده، کلاس بزرگ، متد طولانی و ویژگی حسادت از مجموعه داده فونتانا استفاده کردند. آن‌ها همچنین، الگوریتم گرگ خاکستری را برای انتخاب ویژگی‌های مرتبط به کاربردند. بر اساس نتایج حاصل از آزمایش‌های مختلف، فن یادگیری گروهی رأی‌گیری اکثریت از سه الگوریتم مورداستفاده به‌عنوان مدل‌های پایه بهتر عمل می‌کند.

العذبه و الجمن [۷]، فن یادگیری گروهی پشته سازی را برای بهبود عملکرد در شناسایی کدهای نابسامان ویژگی حسادت، کلاس داده، متد طولانی، کلاس بزرگ، گزاره های تعویض و فهرست طولانی پارامترها از مجموعه داده فونتانا به کاربردند. نتایج تحقیقات آن‌ها بیان می‌کند که الگوریتم‌های یادگیری ماشین تا حدودی حساس به نابسامانی هستند و الگوریتم‌های مختلف نتایج متفاوتی در شناسایی کدهای نابسامان دارند. همچنین به‌طور میانگین فن پشته سازی عملکرد بهتری نسبت به الگوریتم‌های به‌کاررفته در مدل‌های پایه دارد.

الجمن [۸]، از فن یادگیری گروهی رأی‌گیری[4] برای شناسایی کدهای نابسامان کلاس داده، متد طولانی، کلاس بزرگ، ویژگی حسادت، گزاره های تعویض و فهرست طولانی پارامترها از مجموعه داده فونتانا استفاده کرد. بر اساس نتایج به‌دست‌آمده در این تحقیق، الگوریتم‌های یادگیری ماشین تا حدودی حساس به نابسامانی هستند. همچنین، فن رأی‌گیری به‌طور میانگین عملکرد بهتری نسبت به مدل‌های پایه دارد.

جین و ساشا [۹]، در راستای بررسی تأثیر فنون انتخاب ویژگی ترکیبی[5] و یادگیری گروهی بر عملکرد الگوریتم‌های یادگیری ماشین در شناسایی کدهای نابسامان به تحقیق پرداختند. نتایج این تحقیق نشان می‌دهد که انتخاب ویژگی ترکیبی می‌تواند باعث بهبود در شناسایی کدهای نابسامان شود.

شن و همکاران [۱۰]، در مقاله خود، تأثیر بهینه‌سازی فرا پارامترها را بر عملکرد الگوریتم‌های یادگیری ماشین در شناسایی کدهای نابسامان موردبررسی قراردادند. این آزمایش با تمرکز بر شناسایی کدهای نابسامان ویژگی حسادت، متد طولانی، کلاس بزرگ و کلاس داده از مجموعه داده فونتانا انجام شد. بر اساس نتایج به‌دست‌آمده، رویکرد جستجوی شبکه‌ای از سایر رویکردهای بهینه‌سازی فرا پارامترها مؤثرتر عمل می‌کند.

دوانگان و همکاران [۱۱]، در جهت افزایش کیفیت نرم‌افزارهای تولیدی از طریق شناسایی کدهای نابسامان به تحقیق پرداختند. آن‌ها در این تحقیق بر شناسایی کدهای نابسامان ویژگی حسادت، کلاس بزرگ، کلاس داده و متد طولانی از مجموعه داده فونتانا با استفاده از الگوریتم‌های یادگیری ماشین، تمرکز کردند. نتایج تحقیق آن‌ها نشان می‌دهد که به‌طور میانگین الگوریتم جنگل تصادفی[6] بهتر از سایر الگوریتم‌ها عمل می‌کند.

خلیل و نحیز [۱۲]، در راستای بررسی تأثیر متعادل‌سازی داده‌ها بر عملکرد مدل‌های یادگیری ماشین در شناسایی کدهای نابسامان به تحقیق پرداختند. مطالعه آن‌ها بر شناسایی کدهای نابسامان کلاس بزرگ، کلاس داده، ویژگی حسادت و متد طولانی از مجموعه داده فونتانا متمرکز بود. نتایج حاصل‌شده در این تحقیق، اثربخشی متعادل‌سازی داده‌ها بر عملکرد الگوریتم‌های یادگیری ماشین در شناسایی کدهای نابسامان را نشان می‌دهد.

بررسی تحقیقات مرتبط در زمینه شناسایی کدهای نابسامان نشان می‌دهد که یک الگوریتم یادگیری ماشین به‌طورکلی برای شناسایی تمامی کدهای نابسامان مناسب نیست. در این حالت گفته می‌شود، الگوریتم‌ها حساس به نابسامانی هستند و در شناسایی کدهای نابسامانِ مختلف به‌طور متفاوت عمل می‌کنند. در همین راستا، برخی محققان استفاده از فنون یادگیری گروهی را به‌عنوان روشی مناسب جهت رفع این حساسیت پیشنهاد کرده‌اند. نتایج تحقیقات آن‌ها نشان می‌دهد، استفاده از فنون یادگیری گروهی می‌تواند باعث کاهش این حساسیت شود. آنچه از نتایج پژوهش‌ها می‌توان استناد کرد، فن یادگیری گروهی پشته سازی، می‌تواند نتایج بهتری نسبت به الگوریتم‌های منفرد ارائه دهد. همچنین، استفاده از این فنون می‌تواند سبب بهبود عملکرد در شناسایی کدهای نابسامان شود.

همچنین، برخی از محققان نیز تأثیر فرآیند انتخاب ویژگی بر عملکرد الگوریتم‌های یادگیری ماشین در شناسایی کدهای نابسامان را موردبررسی قرار داده‌اند. نتایج تحقیقات انجام‌شده نشان می‌دهد که در صورت استفاده از روش‌ها و الگوریتم‌های انتخاب ویژگی مناسب، عملکرد الگوریتم‌های یادگیری ماشین در شناسایی کدهای نابسامان بهبود می‌یابد. در برخی از این تحقیقات، محققان به بررسی تأثیر استفاده از تلفیق روش‌های مختلف انتخاب ویژگی بر عملکرد الگوریتم‌های یادگیری ماشین در شناسایی کدهای نابسامان پرداخته‌اند. بررسی نتایج این تحقیقات، تأثیر مثبت فنون انتخاب ویژگی ترکیبی و مجموعه‌ای را نشان می‌دهد.

مسئله دیگر، نحوه تفسیر نتایج به‌دست‌آمده توسط برخی محققین است. در این تحقیقات، محققین از روش ارزیابی

[1] Decision Tree (DT)
[2] k-Nearest Neighbors (kNN)
[3] Support Vector Machine (SVM)
[4] Voting ensemble learning
[5] Hybrid feature selection
[6] Random Forest (RF)

مجموعه اعتبارسنجی استفاده کرده‌اند که این کار باعث ایجاد نتایج غیرقابل استناد شده است.

۳. مرور ادبیات تحقیق

در این بخش، به توضیح مفاهیم اولیه موردنیاز جهت درک بهتر و آشنایی با کدهای نابسامان و روش‌های ارائه‌شده جهت شناسایی آن‌ها پرداخته‌شده است. این مفاهیم شامل تعریف کدهای نابسامان، آشنایی با کدهای نابسامان موردپژوهش در این مقاله، فن یادگیری گروهی پشته سازی، فن انتخاب ویژگی مجموعه‌ای و الگوریتم‌های یادگیری ماشین و انتخاب ویژگی به‌کاررفته درروش پیشنهادی است.

۳-۱. کدهای نابسامان

کدهای نابسامان یا به‌عبارتی‌دیگر بوی کد[1] یکی از چالش‌های اصلی در برنامه‌های نرم‌افزاری محسوب می‌شوند. آن‌ها اغلب ناشی از عدم رعایت اصول مهندسی نرم‌افزار، عدم تجربه کافی توسعه‌دهندگان و یا پیاده‌سازی ناکارآمد برنامه هستند. این مفهوم برای نخستین بار در سال ۱۹۹۹ توسط فاولر [۱۳] تعریف شد. او در کتاب خود با اشاره به کدهای نابسامان، آن‌ها را نشانه ای در ساختار کد دانست که ممکن است نشان‌دهنده مشکلات عمیق‌تری در طراحی نرم‌افزار باشند. تاکنون انواع مختلفی از کدهای نابسامان در برنامه‌های نرم‌افزاری شناسایی و معرفی‌شده‌اند. در این تحقیق تمرکز بر بهبود دقت در شناسایی کدهای نابسامان کلاس داده، متد طولانی، کلاس بزرگ، ویژگی حسادت، گزاره‌های تعویض و فهرست طولانی پارامترها است که در ادامه به توضیح آن‌ها پرداخته می‌شود.

۳-۱-۱. کلاس داده

کد نابسامان کلاس داده به کلاسی اشاره می‌کند که فقط از متغیرها و توابعی برای دسترسی به آن متغیرها تشکیل‌شده است. این کلاس‌ها معمولاً به‌عنوان کلاس‌های توصیفی شناخته می‌شوند و عملکرد خاصی را در آن‌ها پیاده‌سازی نکرده‌اند [۱۴].

۳-۱-۲. ویژگی حسادت

کد نابسامان ویژگی حسادت به مواردی اشاره می کند که متدهای یک کلاس وابستگی زیادی به استفاده از متغیرها و متدهای سایر کلاس‌ها دارند. این وابستگی اغلب باعث افزایش وابستگی بین کلاس‌ها می‌شود که منجر به کاهش قابلیت نگهداری و تغییرپذیری برنامه می‌شود [۱۴].

۳-۱-۳. کلاس بزرگ

کد نابسامان کلاس بزرگ به وضعیتی اشاره دارد که تعداد متغیرها و متدهای یک کلاس بیش‌ازحد معمول باشد و حجم زیادی از کد برنامه را اشغال کند. همچنین، عملکردهای زیادی را نیز در آن پیاده‌سازی کرده‌اند. به همین دلیل این نوع کلاس‌ها معمولاً به‌سختی قابل مدیریت و نگهداری هستند و نیاز است که با تقسیم کلاس بزرگ به چند کلاس کوچک‌تر و تقسیم وظایف بین این کلاس‌ها، مشکل را برطرف کرد [۱۳].

۳-۱-۴. متد طولانی

متد طولانی به متدها یا توابعی اشاره دارد که تعداد خطوط کد بدنه آن‌ها بسیار زیاد است. به‌طورمعمول تعداد زیاد خطوط کد یک متد، فهم و درک آن را دشوارتر می‌کند. با کوچک‌تر سازی متد اصلی به تعدادی متد کوچک‌تر و استفاده از محاسبات مستقیم به‌جای استفاده صرف از متغیرهای موقت می‌توان این کد نابسامان را برطرف نمود [۱۴].

۳-۱-۵. فهرست طولانی پارامترها

فهرست طولانی پارامترها هنگامی شکل می گیرد که تعداد پارامترهای دریافتی یک متد بیش‌ازاندازه معمول باشد. وجود فهرست طولانی پارامترها در کد برنامه، می‌تواند خطاهای ناخواسته‌ای در استفاده از پارامترها و پیچیدگی ارتباط بین بخش‌های مختلف یک متد ایجاد کند [۱۴].

۳-۱-۶. گزاره‌های تعویض

کد نابسامان گزاره‌های تعویض به وضعیتی اشاره می‌کند که در آن دستورات شرطی پیچیده‌ای در برنامه وجود دارند. این پیچیدگی می‌تواند منجر به بروز خطاها و استثناهای غیرمنتظره‌ای در زمان اجرا شود. اغلب مدیریت این کدهای نابسامان سخت است و احتمال بروز خطا را نیز افزایش می‌دهند. مشکلاتی در نگهداری و آزمون برنامه ایجاد می‌کنند [۱۴].

۳-۲-۱. یادگیری گروهی

یادگیری گروهی[2] یکی از رویکردهای مؤثر در زمینه یادگیری ماشین است. در این رویکرد از ترکیب نتایج چند مدل یادگیری ماشین ضعیف‌تر برای ایجاد یک مدل یادگیری ماشین با عملکرد بهتر استفاده می‌شود [۱۶].

۳-۲-۱-۱. فن یادگیری گروهی پشته سازی

پشته سازی یکی از فنون یادگیری گروهی است که در این

[2] Ensemble Learning

فن برای ساخت یک مدل یادگیری ماشین قوی‌تر، نتایج چندین مدل پایه با الگوریتم‌های مختلف توسط یک مدل ثانویه به نام فرا مدل ترکیب می‌شوند. در این فرایند، داده‌های آموزشی به دو بخش تقسیم می‌گردند. ابتدا، مدل‌های پایه بر روی بخش اول آموزش می‌بینند. سپس، بر روی بخش دوم آزمون می‌شوند و نتایج پیش‌بینی‌های آن‌ها ذخیره می‌گردد. این پیش‌بینی‌ها همراه با برچسب اصلی متغیر هدف، یک مجموعه داده موقتی را برای آموزش فرا مدل تشکیل می‌دهند. پس‌ازآن، فرا مدل با استفاده از مجموعه داده موقت آموزش می‌بینند و بدین شکل می‌تواند پیش‌بینی نهایی را ارائه دهد [۱۶].

۳-۲-۲. الگوریتم‌های یادگیری ماشین

مدل یادگیری ماشین مورداستفاده در این تحقیق مبتنی بر فن پشته سازی است که همان‌طور که گفته شد، این فن از تعدادی مدل‌های پایه و از یک فرا مدل برای ترکیب نتایج آن‌ها جهت ایجاد یک مدل یادگیری ماشین قوی‌تر استفاده می‌کند. در ادامه به توضیح الگوریتم‌های یادگیری ماشین به‌کاررفته درروش پیشنهادی پرداخته می‌شود.

۳-۲-۲-۱. الگوریتم پرسپترون چندلایه

الگوریتم پرسپترون چندلایه[1] یکی از الگوریتم‌های یادگیری ماشین است که برای مسائل یادگیری تحت نظارت استفاده می‌شود. این الگوریتم شامل سه لایه اصلی، ورودی، لایه پنهان و خروجی است. هر لایه (به‌جز لایه ورودی) شامل چند نرون است که با نرون‌های لایه بعدی ارتباط برقرار می‌کنند. در پرسپترون چندلایه، اطلاعات به‌صورت پیش‌خور از یک‌لایه به لایه بعدی منتقل می‌شود. با انجام این روند تابع وزن نرون‌ها بهینه‌سازی می‌شود تا پیش‌بینی نهایی با دقت بیشتری انجام شود [۱۸][۱۹].

۳-۲-۲-۲. درخت تصمیم‌گیری

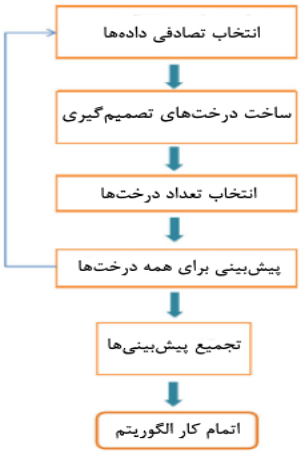
درخت تصمیم‌گیری یکی از الگوریتم‌های رایج در یادگیری ماشین است. این الگوریتم برای پیش‌بینی برچسب‌های داده‌های ورودی از یک درخت دودویی حاوی تصمیم‌ها استفاده می‌کند. این درخت از دو بخش اصلی گره‌ها و برگ‌ها تشکیل‌شده است که گره‌ها نمایانگر تصمیم‌ها هستند و برگ‌ها نیز برچسب‌های نهایی را مشخص می‌کنند. ریشه درخت شامل داده‌های اولیه است و هر گره تصمیمی را بر اساس داده‌های قبلی اتخاذ می‌کند. این فرآیند ادامه می‌یابد تا زمانی که به گره‌های برگ رسیده و پیش‌بینی نهایی انجام شود [۱۷].

۳-۲-۲-۳. ماشین بردار پشتیبان

ماشین بردار پشتیبان یکی از الگوریتم‌های پرکاربرد در یادگیری ماشین است. در این الگوریتم، از یک تابع تصمیم‌گیری به نام ابر صفحه استفاده می‌شود. در ماشین بردار پشتیبان، هدف این است که ابر صفحه به‌گونه‌ای تعیین شود که فاصله بین مرزهای تصمیم‌گیری در فضای ویژگی به حداکثر برسد. این عمل باعث کاهش خطاهای پیش‌بینی می‌شود [۲۰].

۳-۲-۲-۴. الگوریتم جنگل تصادفی

الگوریتم جنگل تصادفی یکی از الگوریتم‌های یادگیری ماشین است که از ترکیب نتایج تعدادی درخت تصمیم‌گیری برای طبقه‌بندی داده‌ها استفاده می‌کند. در این الگوریتم، زیرمجموعه‌های مختلفی از داده‌های آموزشی به‌صورت نمونه‌برداری با جایگزینی انتخاب می‌شوند و درختان تصمیم‌گیری بر اساس این داده‌ها آموزش می‌بینند. سپس هر یک از این درخت‌ها بر روی‌داده‌های آزمون، پیش‌بینی خود را انجام می‌دهند. درنهایت نیز نتایج پیش‌بینی‌های انجام‌شده با استفاده سازوکارهایی مانند رأی‌گیری اکثریت و یا میانگین‌گیری ترکیب می‌شوند. در شکل شماره (۱)، می‌توان روند کلی الگوریتم جنگل تصادفی را مشاهده کرد [۲۱].



شکل (۱). روند کلی الگوریتم جنگل تصادفی [۲۱]

۳-۲-۲-۵. الگوریتم رگرسیون لجستیک

تحلیل رگرسیون یک روش آماری برای پیش‌بینی متغیر وابسته بر اساس متغیرهای مستقل است. رگرسیون لجستیک[2]، یک الگوریتم یادگیری ماشین بر پایه این نظریه است که احتمال تعلق داده‌ها به کلاس‌های مختلف را محاسبه می‌کند [۲۲].

[1] MultiLayer Perceptron (MLP)

[2] Logistic Regression (LR)

۳-۲-۲-۶. الگوریتم تحلیل تشخیص خطی

الگوریتم تحلیل تشخیص خطی[1] یکی از الگوریتم‌های یادگیری ماشین تحت نظارت است که با یافتن ترکیبات خطی بهینه از ویژگی‌ها، به تفکیک و تشخیص کلاس‌های مختلف کمک می‌کند. این الگوریتم با کاهش ابعاد داده‌ها، می‌تواند دقت طبقه‌بندی را بهبود ببخشد [۲۳].

۳-۲-۲-۷. الگوریتم فرآیند گاوسی

الگوریتم فرآیند گاوسی[2] یک الگوریتم غیر پارامتریک در یادگیری ماشین است. در این الگوریتم با استفاده از ماتریس هسته، ارتباطات بین داده‌های آموزشی بررسی می‌شود. سپس میانگین و واریانس پیش‌بینی‌ها برای داده‌های جدید محاسبه می‌گردد. این ویژگی‌ها می‌تواند به تعیین دقیق‌تر کلاس هدف کمک کنند [۲۴].

۳-۳. انتخاب ویژگی

انتخاب ویژگی[3] یک فرآیند پیش‌پردازشی در یادگیری ماشین است. در این فرآیند ویژگی‌هایی از داده‌ها که با متغیر هدف ارتباط بیشتری دارند، برای ادامه فرآیند یادگیری ماشین انتخاب می‌شوند. این فرآیند به کاهش ابعاد داده‌ها، افزایش سرعت یادگیری و بهبود عملکرد مدل‌های یادگیری ماشین کمک می‌کند. الگوریتم‌های انتخاب ویژگی بر اساس راهبردهای انتخاب ویژگی و معیارهای ارزیابی به سه دسته اصلی روش‌های فیلتر[4]، پوشش[5] و تعبیه‌شده[6] تقسیم می‌شوند که در این مقاله از الگوریتم‌های انتخاب ویژگی مبتنی بر پوشش استفاده می‌شود [۲۵].

۳-۳-۱. روش انتخاب ویژگی پوشش

درروش انتخاب ویژگی پوشش، ویژگی‌ها بر اساس تأثیری که بر عملکرد مدل یادگیری ماشین می‌گذارند، انتخاب می‌شوند. مدل بر روی زیرمجموعه‌ای از ویژگی‌ها آموزش‌دیده و عملکرد آن ارزیابی می‌گردد. این فرآیند به‌صورت تکراری انجام می‌شود تا بهترین زیرمجموعه از ویژگی‌ها انتخاب شوند [۲۵].

۳-۳-۱-۱. الگوریتم حذف ویژگی بازگشتی

الگوریتم حذف ویژگی بازگشتی[7] یک الگوریتم انتخاب ویژگی پوششی است. این الگوریتم ابتدا یک مدل را با تمام ویژگی‌ها آموزش می‌دهد. سپس اهمیت هر ویژگی را محاسبه می‌کند. پس‌ازآن، ویژگی‌های کم‌اهمیت را حذف می‌نماید و این فرایند تا رسیدن به تعداد مشخصی از ویژگی‌ها، ادامه می‌یابد [۲۶].

۳-۳-۱-۲. الگوریتم انتخاب پیش‌رونده متوالی

الگوریتم انتخاب پیش‌رونده متوالی[8] در یک‌روند تکراری ویژگی‌های مؤثر را از یک مجموعه ویژگی‌های اولیه انتخاب می‌کند. فرآیند انتخاب ویژگی از یک مجموعه خالی آغاز می‌شود. سپس در هر مرحله ویژگی‌هایی انتخاب می‌گردند که بیشترین تأثیر را بر بهبود عملکرد مدل ارزیابی می‌گذارند. این فرایند تا زمان رسیدن به تعداد خاصی از ویژگی‌ها یا عدم وجود ویژگی جدید برای بهبود عملکرد مدل ارزیابی ادامه دارد [۲۷].

۳-۳-۱-۳. الگوریتم حذف ویژگی بازگشتی با ارزیابی متقابل

الگوریتم حذف ویژگی بازگشتی با اعتبارسنجی متقابل[9] نسخه ارتقاءیافته الگوریتم حذف ویژگی بازگشتی است. این الگوریتم با شروع از تمام ویژگی‌های موجود در داده‌ها، به‌تدریج ویژگی‌های کم‌اهمیت را شناسایی و حذف می‌کند. این مراحل به‌صورت تکراری انجام می‌شوند و در هر مرحله، اعتبارسنجی متقابل برای ارزیابی عملکرد مدل ارزیابی به کار می‌رود. این کار از تأثیر بیش برازش بر انتخاب ویژگی‌های مؤثر جلوگیری می‌کند [۲۶].

۳-۳-۱-۴. الگوریتم ژنتیک

الگوریتم ژنتیک یک روش انتخاب ویژگی پوششی است که بر اساس اصول الگوریتم‌های ژنتیک عمل می‌کند. در این الگوریتم، ویژگی‌ها به‌عنوان ژن‌ها در کروموزوم‌های دودویی نمایش داده می‌شوند. با تولید یک جمعیت اولیه، ارزیابی شایستگی و اعمال عملگرهای ژنتیکی مانند تقاطع و جهش، نسل‌های جدیدی از کروموزوم‌ها ایجاد می‌گردند. این فرایند تا رسیدن به شرایط توقف (مانند تعداد نسل‌ها یا شایستگی مطلوب) ادامه می‌یابد تا درنهایت، بهترین ویژگی‌ها انتخاب شوند [۲۸].

۳-۳-۴. فن انتخاب ویژگی مجموعه‌ای

حاصل به‌کارگیری ایده یادگیری گروهی در فرآیند انتخاب ویژگی باعث ایجاد فن انتخاب ویژگی مجموعه‌ای است. در این فن ویژگی‌های نهایی از ترکیب نتایج چند الگوریتم انتخاب ویژگی مختلف به دست می‌آید. در این حالت، هر الگوریتم به‌طور

[1] Linear Discriminant Analysis (LDA)
[2] Gaussian Process (GP)
[3] Feature selection
[4] Filter feature selection
[5] Wrapper feature selection
[6] Embedded feature selection
[7] Recursive Feature Elimination (RFE)
[8] Step-Forward Selection (SFS)
[9] Recursive Feature Elimination with Cross-Validation (RFECV)

مستقل نتایج خود را تولید می‌کند. سپس با استفاده از سازوکارهایی مانند رأی‌گیری حداکثری یا اجتماع مجموعه‌ها، نتایج ادغام می‌شوند و بدین شکل مجموعه داده شامل ویژگی‌های منتخب نهایی انتخاب می‌شوند. این روش دارای مزایای زیادی، ازجمله کاهش خطر بیش برازش، بهبود دقت پیش‌بینی و انعطاف‌پذیری بالا در استفاده از انواع مختلف الگوریتم‌ها است [۲۹].

۳-۴. مجموعه داده مورداستفاده

مجموعه داده شامل کدهای نابسامان موردتحقیق، توسط فونتانا و همکارانش [۳]، ارائه شد. آن‌ها برای ساخت این مجموعه داده از ۷۴ پروژه نرم‌افزاری توسعه‌یافته با زبان جاوا از مجموعه نرم‌افزاری Qualitas Corpus استفاده کردند. در جدول شماره (۱)، می‌توان اطلاعات مربوط به این مجموعه داده شامل کدهای نابسامان موجود، تعداد کل نمونه‌ها و تعداد شاخص‌های هر مجموعه داده را مشاهده کرد.

جدول (۱). خلاصه اطلاعات مربوط به مجموعه داده فونتانا

کد نابسامان	سطح	تعداد ویژگی‌ها	تعداد نمونه‌ها		
			کل نمونه‌ها	نمونه مثبت	نمونه منفی
متد طولانی	متد	۶۱	۴۲۰	۱۴۰	۲۸۰
ویژگی حسادت			۴۲۰	۱۴۰	۲۸۰
کلاس بزرگ	کلاس	۸۲	۴۲۰	۱۴۰	۲۸۰
کلاس داده			۴۲۰	۱۴۰	۲۸۰
فهرست طولانی پارامترها		۵۵	۴۲۰	۱۴۰	۲۸۰
گزاره‌های تعویض			۴۲۰	۱۴۰	۲۸۰

۴. روش پیشنهادی

در این مقاله روشی مبتنی بر ترکیب فنون انتخاب ویژگی مجموعه‌ای و یادگیری گروهی پشته سازی برای شناسایی کدهای نابسامان کلاس بزرگ، کلاس داده، متد طولانی، ویژگی حسادت، فهرست طولانی پارامترها و گزاره‌های تعویض پیشنهاد می‌شود که شمای کلی آن در شکل شماره (۲)، قابل‌مشاهده است. همان‌طور که در این شکل مشخص است، در این روش ابتدا مجموعه داده فونتانا که شامل شش کد نابسامان مذکور است، گردآوری شد. سپس به دلیل وجود داده‌هایی با مقادیر نامعلوم، روش جایگزینی داده‌های ازدست‌رفته با مقدار میانگین[1] به کار می‌رود. در مرحله بعد نیز، برای یکسان‌سازی دامنه مقادیر ویژگی‌ها، از روش نرمال‌سازی کمینه - بیشینه[2] استفاده می‌شود.

در ادامه، برای کاهش ابعاد داده‌ها و حذف ویژگی‌های غیر مرتبط، از فن انتخاب ویژگی مجموعه‌ای مبتنی بر الگوریتم‌های حذف ویژگی بازگشتی، انتخاب ویژگی پیش رونده متوالی، ژنتیک و حذف ویژگی بازگشتی با اعتبارسنجی متقابل به‌عنوان انتخاب‌گرهای پایه استفاده می‌شود. در این مرحله، برای تجمیع نتایج حاصل از انتخاب‌گرهای پایه و ایجاد مجموعه داده با ویژگی‌های منتخب نهایی، سازوکار رأی‌گیری حداکثری به کار می‌رود.

سپس مجموعه داده شامل ویژگی‌های منتخب نهایی به مدل یادگیری ماشین مبتنی بر فن یادگیری گروهی پشته سازی داده می‌شود. در مدل یادگیری ماشین مورداستفاده، الگوریتم پرسپترون چندلایه به‌عنوان فرا مدل و الگوریتم‌های درخت تصمیم گیری، جنگل تصادفی، تحلیل تشخیص خطی، ماشین بردار پشتیبان، رگرسیون لجستیک و فرآیند گاوسی به‌عنوان الگوریتم‌های پایه به کار می‌روند.

آموزش و آزمون مدل یادگیری ماشین نیز، با استفاده از روش اعتبارسنجی متقابل ۱۰-بخشی انجام می‌شود. در این روش داده‌ها به ۱۰ بخش تقریباً مساوی تقسیم می‌شوند. مدل‌های پایه بر روی نه بخش اول آموزش‌دیده و روی یک بخش باقی‌مانده مورد آزمون قرار می‌گیرند. این روند به‌صورت تکراری ادامه می‌یابد تا زمانی که از هر ۱۰ بخش موجود برای آموزش و آزمون مدل‌های پایه استفاده شود. سپس، نتایج پیش‌بینی آن‌ها به همراه برچسب اصلی هدف، یک مجموعه داده موقت را تشکیل می‌دهند.

در ادامه، فرآیند آموزش و آزمون فرا مدل بر روی مجموعه داده موقت نیز، با استفاده از روش اعتبارسنجی متقابل ۱۰-بخشی انجام می شود. در این مرحله، مجموعه داده موقت به ۱۰ بخش تقسیم می‌گردد و مانند مدل‌های پایه، فرآیند آموزش و آزمون نیز برای آن انجام می‌پذیرد. درنهایت، با میانگین‌گیری از

[1] Missing value management by mean imputation
[2] Min-max scaler algorithm

عملکرد پیش‌بینی فرا مدل بر روی هر یک از ۱۰ بخش، عملکرد نهایی فرا مدل بر اساس معیارهای دقت، امتیاز F1 و AUC به دست می‌آید.

درروش پیشنهادی، به‌کارگیری فن انتخاب ویژگی مجموعه‌ای با استفاده از یک سازوکار مناسب جهت ترکیب نتایج حاصل از الگوریتم‌های انتخاب ویژگی پایه می‌تواند، سبب انتخاب ویژگی‌های مهمی شود که ممکن است در استفاده از یک الگوریتم انتخاب ویژگی منفرد، آن ویژگی‌ها انتخاب نشوند. همچنین، این روش می‌تواند به بهبود عملکرد مدل یادگیری ماشین نیز کمک کند.

با استفاده از فن یادگیری گروهی پشته سازی می‌توان یک مدل یادگیری ماشین با عملکرد بهتری برای شناسایی کدهای نابسامان ایجاد کرد. همچنین، فن پشته سازی به دلیل استفاده از مدل‌های پایه با الگوریتم‌های متفاوت می‌تواند از مزایای هر یک از الگوریتم‌ها در شناسایی کدهای نابسامان استفاده کند و بر حساسیت به نابسامانی مدل‌های یادگیری ماشین پایه غلبه کند.

۴-۱. فرآیند انتخاب ویژگی

درروش پیشنهادی فرآیند انتخاب ویژگی‌های مرتبط، با استفاده از فن انتخاب ویژگی مجموعه‌ای انجام می‌شود که فرآیند کلی آن در شکل شماره (۳)، مشاهده می‌گردد. همان‌طور که بیان شد، در فرآیند انتخاب ویژگی مبتنی بر فن انتخاب ویژگی مجموعه‌ای الگوریتم‌های انتخاب ویژگی حذف ویژگی تکراری، انتخاب پیش‌رونده متوالی، حذف ویژگی تکراری با اعتبارسنجی متقابل و ژنتیک به‌عنوان انتخاب‌گرهای پایه استفاده می‌شوند.

در این فرآیند، ابتدا مجموعه داده شامل تمام ویژگی‌های اولیه به انتخاب‌گرهای پایه داده می‌شود. سپس، هر یک از الگوریتم‌ها بر اساس رویه انتخاب ویژگی خود زیرمجموعه‌های مختلفی از ویژگی‌های اولیه را تولید کرده و مطابق با آن عملکرد مدل ارزیاب را که در این روش پیشنهادی از الگوریتم رگرسیون لجستیک به‌عنوان مدل ارزیاب استفاده‌شده است، اندازه‌گیری می‌کنند. استفاده از الگوریتم رگرسیون لجستیک به‌عنوان الگوریتم ارزیاب در فرآیند انتخاب ویژگی، به دلیل ثبات بالای نتایج آن است. این الگوریتم معمولاً ویژگی‌هایی را انتخاب می‌کند که در تکرارهای مختلف مشابه هستند که این امر به بهبود قابلیت اطمینان مدل اصلی کمک می‌کند. این رویه تا زمان اتمام کار الگوریتم‌های انتخاب ویژگی و تولید زیرمجموعه ویژگی‌های منتخب، ادامه دارد. در ادامه زیرمجموعه‌های تولیدشده با استفاده از سازوکار رأی‌گیری حداکثری با یکدیگر ترکیب‌شده و مجموعه داده شامل ویژگی‌های نهایی را تشکیل می‌دهند.

دلیل استفاده از فن انتخاب ویژگی مجموعه‌ای درروش پیشنهادی این است که در این فن، با ترکیب نتایج انتخاب‌گرهای پایه با یک سازوکار مناسب، امکان انتخاب ویژگی‌های مهم‌تری وجود دارد. این در حالتی است که این ویژگی‌ها ممکن است، توسط یک روش انتخاب ویژگی منفرد انتخاب نشوند. این فن به بهبود عملکرد و کاهش خطای پیش‌بینی مدل یادگیری ماشین نیز کمک می‌کند.

دلایلی نیز جهت استفاده از چهار الگوریتم انتخاب ویژگی مذکور وجود دارد. هر یک از این الگوریتم‌ها، دارای خواصی هستند که آن‌ها را مکمل یکدیگر می‌کند. در الگوریتم حذف ویژگی بازگشتی، انتخاب ویژگی با حذف گام‌به‌گام کم‌اهمیت‌ترین ویژگی‌ها و ارزیابی مداوم تأثیر آن‌ها بر عملکرد مدل، انجام می‌شود. به همین دلیل، این الگوریتم دقت بالایی در شناسایی ویژگی‌های کلیدی دارد. استفاده از این الگوریتم می‌تواند، به انتخاب ویژگی‌های مهم منجر شود. زیرا به‌طور مکرر ویژگی‌ها را ارزیابی می‌کند و ویژگی‌هایی را که تأثیر کمی بر عملکرد مدل‌دارند، شناسایی و حذف می‌کند.

الگوریتم حذف ویژگی بازگشتی با اعتبارسنجی متقابل، از اعتبارسنجی متقابل برای ارزیابی کیفیت انتخاب ویژگی‌ها استفاده می‌کند. این رویکرد تضمین می‌کند که مجموعه ویژگی‌های نهایی نه‌تنها بر داده‌های آموزشی، بلکه بر روی‌داده‌های جدید نیز عملکرد بهینه‌ای دارد. این ویژگی‌ها به‌گونه‌ای انتخاب می‌شوند که موجب بیش برازش مدل یادگیری ماشین نشوند.

الگوریتم انتخاب ویژگی ژنتیک نیز به دلیل قابلیت جستجوی فراگیر و بهینه‌سازی، توانایی شناسایی ترکیب‌های بهینه از ویژگی‌ها را دارد. این الگوریتم با تقلید از فرآیند انتخاب طبیعی، می‌تواند از فضای جستجوی بزرگ‌تری بهره‌برداری کند و ترکیب‌هایی را پیدا کند که ممکن است با سایر روش‌ها شناسایی نشوند. این خاصیت امکانی را فراهم می‌کند که به ویژگی‌های پیچیده‌تری دست پیدا کرد که به‌راحتی توسط الگوریتم‌های دیگر شناسایی نمی‌شوند.

دلیل استفاده از الگوریتم انتخاب پیش‌رونده متوالی آن است که این الگوریتم نه‌تنها به بررسی اثربخشی هر ویژگی به‌طور مستقل می‌پردازد، بلکه در هر مرحله، تأثیر ترکیبی ویژگی‌های منتخب را بر عملکرد مدل ارزیابی می‌کند. این امر امکانی را فراهم می‌کند که الگوریتم مذکور، ویژگی‌های مکمل و مهم را شناسایی کند که ممکن است، به‌تنهایی تأثیر چندانی نداشته باشند، اما در ترکیب با سایر ویژگی‌ها به بهبود دقت کلی مدل کمک می‌کنند.

۴-۱-۱. پارامترهای الگوریتم‌های انتخاب ویژگی

همان‌طور که قبلاً بیان شد، فرآیند انتخاب ویژگی مبتنی بر فن انتخاب ویژگی مجموعه‌ای با استفاده از الگوریتم‌های انتخاب ویژگی ژنتیک، حذف ویژگی بازگشتی، انتخاب پیش‌رونده متوالی و حذف ویژگی بازگشتی با اعتبارسنجی متقابل انجام می‌شود. هر یک از این الگوریتم‌ها پارامترهای خاصی را دریافت می‌کنند که با توجه به آن‌ها رفتار این الگوریتم‌ها را می‌توان تنظیم کرد. در جدول شماره (۲)، پارامترهای دریافتی هر یک از الگوریتم‌های انتخاب ویژگی پایه و تنظیمات آن‌ها ارائه‌شده است.

جدول (۲). پارامترهای مربوط به الگوریتم‌های انتخاب ویژگی

الگوریتم	پارامترها
RFECV	Step=1, cv=’ StratifiedKFold (5)’, scoring= ’accuracy’, estimator=LogisticRegression (solver='liblinear')
Genetic	Pop_size=20, n_generations=50, mutation_rate=0.01, estimator=LogisticRegression (solver='liblinear')
SFS	Direction= ’forward’, n_features_to_select= ’auto’, scoring= ’accuracy’, cv=’ StratifiedKFold (5)’, estimator=LogisticRegression (solver='liblinear')
RFE	Step=1, estimator=LogisticRegression (solver='liblinear'), n_features_to_select= 0.25

۴-۲. مدل یادگیری ماشین

همان‌گونه که اشاره شد، درروش پیشنهادی، فرآیند طبقه‌بندی و شناسایی کدهای نابسامان با استفاده از یک مدل یادگیری ماشین مبتنی بر فن پشته سازی انجام می‌شود که شمای کلی این فرآیند و مدل را می‌توان در شکل شماره (۴)، مشاهده نمود. همان‌طور که در این شکل مشاهده می‌شود، مدل یادگیری ماشین موردنظر، از الگوریتم‌های یادگیری ماشین درخت تصمیم‌گیری، جنگل تصادفی، ماشین بردار پشتیبان، فرآیند گاوسی، تحلیل تشخیص خطی و رگرسیون لجستیک به‌عنوان مدل‌های پایه و از الگوریتم پرسپترون چندلایه برای فرا مدل استفاده می‌شود. فرآیند آموزش و آزمون مدل یادگیری ماشین نیز با استفاده از روش اعتبارسنجی متقابل ۱۰-بخشی انجام می‌پذیرد. در این فرآیند، داده‌های آموزشی به دو بخش تقسیم می‌شوند که یک بخش آن برای آموزش مدل‌های پایه و بخش دیگر برای آزمون مدل‌های پایه و تولید داده آموزشی برای آموزش فرا مدل استفاده می‌شوند. دلیل استفاده از روش اعتبارسنجی متقابل ۱۰-بخشی برای فرآیند آموزش و آزمون این است که در صورت کمبود تعداد داده‌های آموزشی، مدل یادگیری ماشین پیشنهادی دچار کم برازش می‌شود. همچنین، در صورت استفاده مجدد از داده‌های آموزشی، مشکل بیش برازش برای مدل یادگیری ماشین به وجود می‌آید. به همین دلیل برای جلوگیری از این مشکلات، از روش اعتبارسنجی متقابل ۱۰-بخشی استفاده می‌شود. در این مرحله، داده‌های موجود به ۱۰ بخش تقسیم می‌شوند که نه بخش آن برای آموزش مدل‌های پایه و یک بخش دیگر نیز برای آزمون آن‌ها مورداستفاده قرار می‌گیرد. این روال به‌صورت تکراری ادامه می‌یابد تا زمانی که تمامی ۱۰ بخش برای آموزش و آزمون مدل‌های پایه به کار گرفته شوند. سپس با استفاده از پیش‌بینی‌های انجام‌شده و برچسب اصلی متغیر هدف یک مجموعه داده موقتی برای آزمون فرا مدل تشکیل می‌شود. برای آموزش و آزمون فرا مدل نیز روندی مشابه مدل‌های پایه صورت می‌پذیرد. درنهایت، با آموزش و آزمون مدل‌های پایه، پیش‌بینی نهایی انجام می‌شود. سپس عملکرد مدل یادگیری ماشین بر اساس نتایج آزمون فرا مدل بر روی هر یک از ده بخش محاسبه می‌گردد.

همان‌طور که گفته شد، با استفاده از فن یادگیری گروهی پشته سازی می‌توان با ترکیب چند مدل یادگیری ماشین مختلف یک مدل یادگیری ماشین با عملکرد بهتری برای شناسایی کدهای نابسامان ایجاد کرد که به همین دلیل از فن پشته سازی جهت ایجاد مدل یادگیری ماشین درروش پیشنهادی استفاده‌شده است. الگوریتم‌های یادگیری ماشین مورداستفاده به‌عنوان مدل‌های پایه در مدل پشته سازی نیز بر اساس عملکرد آن‌ها در تحقیقات پیشین انتخاب‌شده‌اند.

۴-۲-۱. پارامترهای الگوریتم‌های یادگیری ماشین

همانند الگوریتم‌های انتخاب ویژگی، الگوریتم‌های به‌کاررفته در مدل یادگیری ماشین پیشنهادی، شامل الگوریتم‌های مورداستفاده به‌عنوان مدل‌های پایه و الگوریتم به‌کاررفته برای فرا مدل، هر یک دارای فرا پارامترهایی هستند که تنظیم آن‌ها رفتار مدل یادگیری ماشین را مشخص می‌کند. در جدول شماره (۳)، می‌توان تنظیمات مربوط به پیکربندی فرا پارامترهای الگوریتم‌های یادگیری ماشین مورداستفاده را مشاهده کرد. برای پیاده‌سازی الگوریتم‌های یادگیری ماشین مذکور از کتابخانه اس‌کی‌لرن[1] استفاده‌شده است. برخی از فرا پارامترهایی که این الگوریتم‌ها دریافت می‌کنند، در اینجا ذکر نشده‌اند. دلیل این امر استفاده از پارامترهایی با پیکربندی پیش‌فرض خودشان است.

[1] sklearn

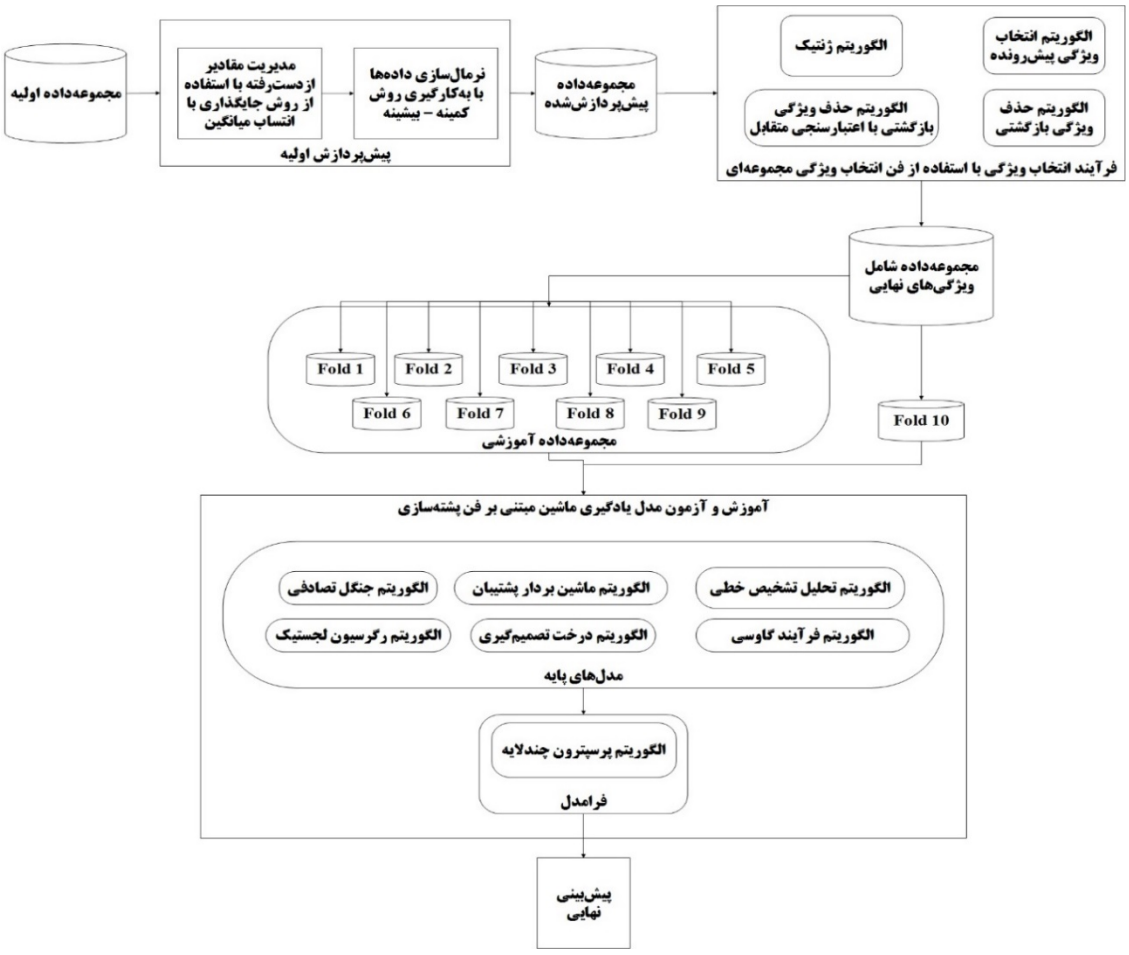
جدول (۳). پارامترهای مربوط به الگوریتم‌های یادگیری عمیق

الگوریتم	پارامترها
DT	criterion='gini', max_depth='100'
LDA	Solver='svd', shrinkage=None, priors= None,
LR	Solver='liblinear', C=10, kernel='linear'
SVM	C=10, gamma='scale', kernel='linear', probability=true
RF	n_estimators=50, criterion='gini', max_depth=None, bootstrap=True
GP	Kernel=none, max_iter_predict=100
MLP	max_iter=200, early_stopping=True, activation='relu', solver='adam',

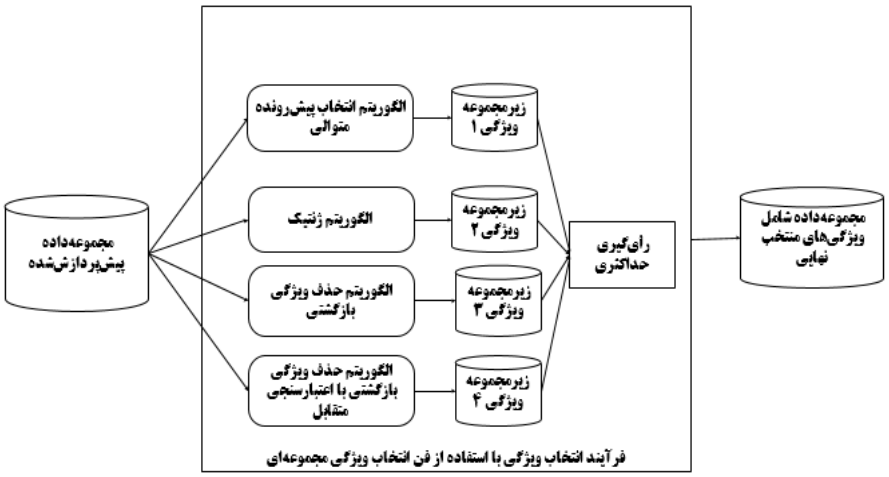
۵. ارزیابی روش پیشنهادی

در این بخش، نتایج حاصل از اجرای روش پیشنهادی برای شناسایی کدهای نابسامان کلاس بزرگ، متد طولانی، ویژگی حسادت، کلاس داده، فهرست طولانی پارامترها و گزاره‌های تعویض مورد ارزیابی قرار می‌گیرد. همچنین، نتایج به‌دست‌آمده با نتایج مقالات مرتبط و نتایج حاصل از مدل‌های پایه (بدون اعمال فن پشته سازی) نیز مقایسه می‌شود که در ادامه این بخش به آن‌ها پرداخته خواهد شد.

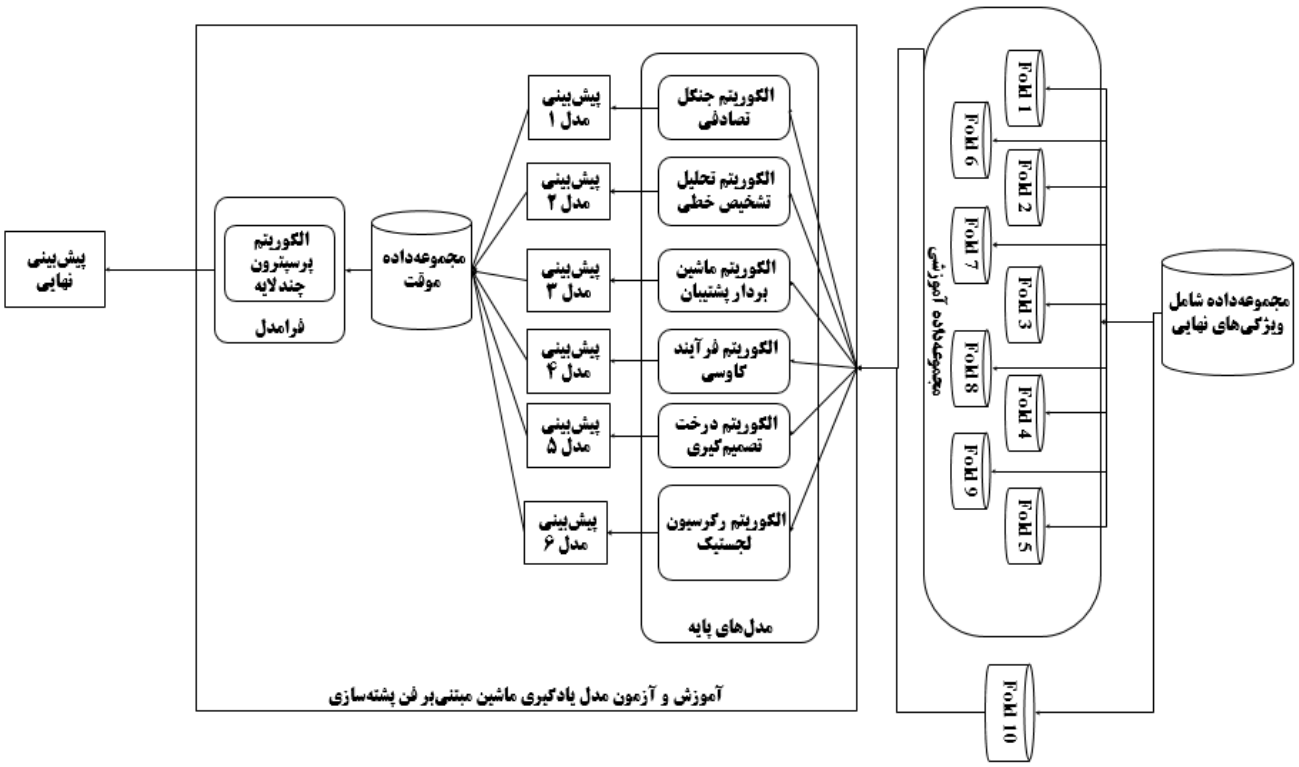
پیاده‌سازی آن مطابق فرآیند شکل شماره (۳) و (۴)، انجام‌شده است. فرآیندهای پیش‌پردازشی نیز انجام شدند. انتخاب ویژگی مبتنی بر فن انتخاب ویژگی مجموعه‌ای با استفاده از الگوریتم‌های پوششی مذکور انجام گردید. سپس مدل یادگیری ماشین بر اساس فن پشته سازی با مدل‌های پایه و فرا مدل ذکرشده پیاده‌سازی شد و روش اعتبارسنجی متقابل ۱۰-بخشی برای فرآیند آموزش و آزمون آن مورداستفاده قرار گرفت. درنهایت، عملکرد نهایی مدل بر اساس عملکرد فرا مدل و برحسب معیارهای ارزیابی دقت، AUC و امتیاز F1 محاسبه شد.



شکل (۲). شمای کلی روش پیشنهادی



شکل (۳). شمای فرآیند انتخاب ویژگی



شکل (۴). شمای کلی فرآیند آموزش و آزمون مدل یادگیری ماشین مورداستفاده در روش پیشنهادی

۵-۱. نتایج و تحلیل

در جدول شماره (۴)، نتایج مربوط به شناسایی کدهای نابسامان مذکور بر اساس معیارهای دقت، AUC و امتیاز F1 مشاهده می‌شود.

جدول (۴). نتایج مربوط به شناسایی کدهای نابسامان با استفاده از روش پیشنهادی

معیار			
کد نابسامان	**دقت (٪)**	**AUC (٪)**	**امتیاز F1 (٪)**
کلاس داده	۹۹/۷۶ ± ۰/۱۷	۹۹/۹۸ ± ۰/۰۵	۹۹/۶۳ ± ۰/۲۳
کلاس بزرگ	۹۷/۶۲ ± ۱/۲۵	۹۹/۷۴ ± ۰/۲۳	۹۶/۴۳ ± ۲/۴۱
متد طولانی	۹۹/۷۸ ± ۰/۱۱	۹۹/۹۷ ± ۰/۰۹	۹۹/۶۶ ± ۰/۱۴
ویژگی حسادت	۹۷/۱۴ ± ۱/۲۵	۹۹/۱۱ ± ۰/۸۳	۹۵/۷۰ ± ۲/۴۱
فهرست طولانی پارامترها	۹۴/۵ ±۲/۳۵	۹۷/۴۲ ± ۱/۹۶	۸۸/۹۱ ± ۲/۷۷
گزاره‌های تعویض	۹۰/۰۱/ ± ۳/۴۱	۹۴/۵۱ ± ۲/۵۴	۸۳/۴۳ ± ۳/۹۳

مطابق جدول شماره (۴)، نتایج اعمال روش پیشنهادی برای شناسایی شش کد نابسامان مذکور بر اساس معیارهای دقت، AUC و امتیاز F1 به شرح زیر است. برای شناسایی کد نابسامان کلاس داده، به ترتیب نتایج ۰/۱۷ ± ۹۹/۷۶، ۰/۰۵ ± ۹۹/۹۸ و ۰/۲۳ ± ۹۹/۶۳ ٪ برای این سه معیار حاصل شد. برای کد نابسامان کلاس بزرگ نیز مقادیر ۱/۲۵ ± ۹۷/۶۲، ۰/۲۳ ± ۹۹/۷۴ و ۲/۴۱ ± ۹۶/۴۳ ٪ برای معیارهای دقت، AUC و امتیاز F1 به دست آمد. از اعمال این روش برای شناسایی کد نابسامان متد طولانی نیز به ترتیب نتایج ۰/۱۱ ± ۹۹/۷۸، ۰/۰۹ ± ۹۹/۹۷ و ۰/۱۴ ± ۹۹/۶۶ ٪ برای معیارهای مورد نظر کسب شد. همچنین، برای شناسایی کد نابسامان ویژگی حسادت نیز به ترتیب مقادیر ۱/۲۵ ± ۹۷/۱۴، ۰/۸۳ ± ۹۹/۱۱ و ۲/۴۱ ± ۹۵/۷۰ ٪ برای معیارهای دقت، AUC و امتیاز F1 به دست آمد. نتایج مربوط به شناسایی کد نابسامان فهرست طولانی پارامترها نیز به ترتیب برابر ۲/۳۵ ± ۹۴/۵، ۱/۹۶ ± ۹۷/۴۲ و ۲/۷۷ ± ۸۸/۹۱ ٪ حاصل شد. درنهایت، برای کد نابسامان گزاره‌های تعویض نیز مقادیر ۳/۴۱ ± ۹۰/۰۱/، ۲/۵۴ ± ۹۴/۵۱ و ۳/۹۳ ± ۸۳/۴۳ ٪ ازنظر معیارهای عملکردی موردنظر به دست آمد.

بر اساس نتایج حاصل‌شده برای شناسایی کدهای نابسامان مذکور، به‌طور میانگین بهترین عملکرد برای شناسایی کدهای نابسامان کلاس داده و متد طولانی با عملکرد حدود ۹۹ ٪ در معیار دقت، حاصل شد. در رتبه بعدی برای شناسایی کدهای نابسامان کلاس بزرگ و کلاس بزرگ نیز به ترتیب عملکرد حدود ۹۷ ٪ در معیار ارزیابی دقت به دست آمد. پایین‌ترین عملکرد نیز به ترتیب برای کدهای نابسامان گزاره‌های تعویض و فهرست طولانی پارامترها به دست آمد.

همچنین، با توجه به انحراف معیارهای به‌دست‌آمده می‌توان اظهارنظر کرد که روش پیشنهادی در شناسایی کدهای نابسامان کلاس داده و متد طولانی عملکرد قابل اتکایی داشته است. در رابطه با انحراف معیار به‌دست‌آمده برای شناسایی کدهای نابسامان ویژگی حسادت، کلاس بزرگ، گزاره‌های تعویض و فهرست طولانی پارامترها قابلیت اتکای مدل پایین‌تر است.

در جدول شماره (۵)، نتایج مربوط به مقایسه این روش با نتایج مربوط به اعمال الگوریتم‌های به‌کاررفته در مدل‌های پایه برای شناسایی کدهای نابسامان مذکور، آورده شده است که در این حالت فن پشته سازی مورداستفاده قرار نگرفته است.

جدول (۵). نتایج مربوط به مقایسه روش پیشنهادی با الگوریتم‌های پایه

کد نابسامان	**کلاس بزرگ**	**کلاس داده**	**متد طولانی**	**ویژگی حسادت**	**فهرست طولانی پارامترها**	**گزاره‌های تعویض**
الگوریتم	**معیار دقت (٪)**					
درخت تصمیم‌گیری	۹۵/۷۱ ± ۱/۳۸	۹۹/۰۴ ± ۰/۸۰	۹۹/۰۴ ± ۰/۳۱	۹۵/۷۱ ± ۱/۳۸	۹۲/۳۴ ± ۲/۲۷	۸۶/۵۲ ± ۴/۵۱
رگرسیون لجستیک	۹۶/۴۲ ± ۲/۶۴	۹۷/۵۴ ± ۱/۸۹	۹۹/۴۹ ± ۰/۳۸	۹۳/۱۹ ± ۲/۸۱	۸۹/۹۰ ± ۳/۶۵	۸۹/۰۵ ± ۳/۹۵
تحلیل تشخیص خطی	۹۵/۲۲ ± ۲/۸۸	۹۶/۴۳ ± ۲/۲۱	۹۶/۴۳ ± ۲/۲۰	۸۹/۲۸ ± ۲/۳۸	۸۷/۸۶ ± ۳/۲۲	۸۵/۰۱ ± ۴/۵۵
ماشین بردار پشتیبان	۹۶/۶۶ ± ۲/۵۳	۹۹/۲۳ ± ۰/۶۸	۹۹/۳۰ ± ۰/۲۳	۹۵/۹۴ ± ۱/۷۸	۹۰/۴۵ ± ۳/۲۳	۸۸/۷۴ ± ۴/۶۱
فرآیند گاوسی	۹۶/۴۲ ± ۱/۷۳	۹۹/۳۴ ± ۰/۳۶	۹۸/۵۴ ± ۱/۷۲	۹۷/۰۴ ± ۱/۲۷	۸۳/۹۰ ± ۲/۷۲	۸۵/۴۸ ± ۴/۸۴
جنگل تصادفی	۹۷/۴۲ ± ۱/۳۳	۹۹/۵۴ ± ۰/۲۱	۹۹/۵۲ ± ۰/۲۴	۹۶/۹۸ ± ۱/۵۱	۹۳/۴۳ ± ۲/۲۱	۸۹/۵۰ ± ۳/۷۱
روش پیشنهادی	۹۷/۶۲ ± ۱/۲۵	۹۹/۷۶ ± ۰/۱۴	۹۹/۷۸ ± ۰/۱۹	۹۷/۳۸ ± ۱/۲۵	۹۴/۵ ± ۲/۳۵	۹۰/۰۱ ± ۳/۴۱

بر اساس نتایج موجود در جدول شماره (۵)، برای شناسایی کد نابسامان کلاس بزرگ از بین الگوریتم‌های به‌کاررفته در مدل‌های پایه، حداکثر دقت شناسایی با استفاده از الگوریتم جنگل تصادفی معادل ۱/۳۳ ± ۹۷/۴۲ ٪ به دست آمد. از میان الگوریتم‌های پایه نیز در جهت شناسایی کد نابسامان کلاس داده بیشترین دقت شناسایی برابر ۰/۲۱ ± ۹۹/۵۴ ٪ و با استفاده از الگوریتم جنگل تصادفی حاصل شد. برای کد نابسامان متد طولانی نیز از میان الگوریتم‌های پایه به‌کاررفته برای شناسایی آن، حداکثر دقت شناسایی با به‌کارگیری الگوریتم فرآیند گاوسی و معادل ۰/۲۴ ± ۹۹/۵۲ ٪ حاصل گشت. از میان نتایج مربوط به اعمال الگوریتم‌های پایه، حداکثر دقت شناسایی برای کد نابسامان ویژگی حسادت با استفاده از الگوریتم فرآیند گاوسی و معادل ۱/۲۷ ± ۹۷/۰۴ ٪ به دست آمد. همچنین، برای کد نابسامان فهرست طولانی پارامترها نیز، حداکثر دقت شناسایی ۲/۲۱ ± ۹۳/۴۳ ٪ و با به‌کارگیری الگوریتم جنگل تصادفی حاصل شد. در رابطه با شناسایی کد گزاره‌های تعویض نیز با استفاده از الگوریتم‌های پایه بیشینه دقت شناسایی ۳/۷۱ ± ۸۹/۵۰ ٪ و با استفاده از الگوریتم جنگل تصادفی به دست آمد.

مقایسه بیشینه نتایج حاصل از شناسایی کدهای نابسامان مذکور با استفاده از الگوریتم‌های به‌کاررفته در مدل‌های پایه، با نتایج حاصل از روش پیشنهادی، تأثیر فن یادگیری گروهی پشته سازی بر عملکرد بهتر این روش در مقایسه با مدل‌های یادگیری ماشین پایه را نشان می‌دهد.

در جدول شماره (۶)، نیز نتایج مربوط به مقایسه تحلیلی این روش بر اساس آزمون ویلکاکسون با نتایج مربوط به اعمال الگوریتم‌های به‌کاررفته در مدل‌های پایه برای شناسایی کدهای نابسامان مذکور، آورده شده است.

جدول (۶). ادامه نتایج مربوط به مقایسه روش پیشنهادی با الگوریتم‌های پایه

کد نابسامان	کلاس بزرگ	کلاس داده	متد طولانی	ویژگی حسادت	فهرست طولانی پارامترها	گزاره‌های تعویض
الگوریتم	معیار p-value					
درخت تصمیم‌گیری	۰/۱۸	۰/۶۸	۰/۶۸	۰/۴۱	۰/۰۴۶	۰/۰۴۹
رگرسیون لجستیک	۰/۳۳	۰/۰۴۹	۰/۷۱	۰/۰۴۰	۰/۰۳۹	۰/۵۵
تحلیل تشخیص خطی	۰/۱۲	۰/۰۳۳	۰/۰۴۹	۰/۰۱	۰/۰۲۹	۰/۰۲۷
ماشین بردار پشتیبان	۰/۴۷	۰/۷۸	۰/۷۲	۰/۴۷	۰/۰۴۱	۰/۱۱
فرآیند گاوسی	۰/۳۶	۰/۸۰	۰/۸۴	۰/۹۱	۰/۰۱۱	۰/۰۳۴
جنگل تصادفی	۰/۹۲	۰/۸۳	۰/۸۱	۰/۸۹	۰/۰۷۵	۰/۷۸

بر اساس نتایج جدول شماره (۶)، برای کد نابسامان کلاس بزرگ، مقدار p-value برای تمامی الگوریتم‌های پایه مقداری بزرگ‌تر از ۰/۰۵ است، یعنی الگوریتم‌های پایه ازنظر عملکردی به مدل پشته سازی نزدیک هستند و نظر قطعی در مورد تفاوت عملکرد آن‌ها نمی‌توان ارائه کرد.

در جهت شناسایی کد نابسامان کلاس داده نیز، برای الگوریتم‌های رگرسیون لجستیک و تحلیل تشخیص، مقدار p-value کمتر از ۰/۰۵ به‌دست‌آمده است که از این مقدار می‌توان به‌طورقطع استنباط کرد که عملکرد مدل پشته سازی نسبت به این دو الگوریتم بهتر است. اما برای سایر الگوریتم‌ها مقدار آن بیشتر از ۰/۰۵ است و نمی‌توان در مورد برتری قطعی مدل پشته سازی نسبت به الگوریتم‌های ماشین بردار پشتیبان، جنگ تصادفی، درخت تصمیم‌گیری و فرآیند گاوسی اظهارنظر کرد.

برای کد نابسامان متد طولانی، به‌جز الگوریتم تحلیل تشخیص خطی، برای سایر الگوریتم‌ها مقدار p-vlaue بزرگ‌تر از ۰/۰۵ است و این الگوریتم‌ها تفاوت آماری معناداری با مدل پشته سازی ندارند. این مسئله می‌تواند نشان‌دهنده عملکرد مشابه مدل پشته سازی با سایر الگوریتم‌های پایه باشد.

برای کد نابسامان ویژگی حسادت نیز، مقدار p-value الگوریتم‌های تحلیل تشخیص خطی و رگرسیون لجستیک کمتر از ۰/۰۵ است که می‌توان به‌طورقطع در مورد عملکرد بهتر مدل یادگیری ماشین اصلی، نسبت به این دو الگوریتم اظهارنظر کرد. همچنین، در مورد سایر الگوریتم‌ها، الگوریتم‌های فرآیند گاوسی و جنگل تصادفی، مقدار p-value نزدیک به ۱ دارند که این مقدار نشان‌دهنده عملکرد نزدیک این الگوریتم‌ها به عملکرد مدل اصلی است. اما بر اساس مقدار p-value برای الگوریتم‌های درخت

تصمیم‌گیری و ماشین بردار پشتیبان نمی‌توان اظهارنظر قطعی کرد.

برای کد نابسامان فهرست طولانی پارامترها نیز، به‌جز برای الگوریتم جنگل تصادفی، مقدار p-value برای سایر الگوریتم‌ها کمتر از ۰/۰۵ است که بر اساس آزمون ویلکاکسون، می‌توان بیان کرد که عملکرد مدل پیشنهادی از عملکرد تمامی الگوریتم‌های یادگیری ماشین پایه به‌جز الگوریتم جنگل تصادفی بهتر است. اما در مورد الگوریتم جنگل تصادفی نمی‌توان اظهارنظر قطعی کرد.

در مورد کد نابسامان گزاره‌های تعویض، مقدار p-value برای الگوریتم‌های تحلیل تشخیص خطی، فرآیند گاوسی و درخت تصمیم‌گیری کمتر از ۰/۰۵ است. به همین دلیل، به‌طورقطع می‌توان اظهار داشت که مدل پیشنهادی ازنظر عملکردی از این الگوریتم‌ها بهتر عمل می‌کند. اما برای الگوریتم‌های پایه دیگر به علت داشتن مقدار p-value بزرگ‌تر از ۰/۰۵، نمی‌توان اظهارنظر قطعی در مورد تفاوت عملکردی آن‌ها با مدل پشته سازی بیان کرد.

۵-۱-۱. مقایسه با تحقیقات مرتبط

همان‌طور که بیان شد، مجموعه داده شامل کدهای نابسامان مذکور توسط فونتانا و همکاران [۳]، ارائه شد.

از میان تحقیقات بررسی‌شده در بخش ۲، تحقیقات کریمی و خسروی [۶]، العذبه و الجمن [۷]، الجمن [۸]، جین و ساشا [۹]، دوانگان و همکاران [۱۱]، خلیل و نهیز [۱۲]، در راستای شناسایی کدهای نابسامان مذکور از مجموعه داده فونتانا انجام‌شده است و می‌توان نتایج حاصل از روش پیشنهادی را با نتایج آن‌ها مقایسه نمود.

بر اساس نتایج تحقیقات انجام‌شده در زمینه شناسایی کد نابسامان کلاس بزرگ، فونتانا و همکاران [۳]، در تحقیق خود با به‌کارگیری الگوریتم بیز ساده برای این کد نابسامان بیشینه دقت شناسایی ۹۷/۵۵ ٪ را به دست آوردند. کریمی و خسروی [۶]، با اعمال فن یادگیری گروهی رأی‌گیری دقت شناسایی ۹۳ ٪ را کسب کردند. در تحقیق العذبه و الجمن [۷]، با اعمال فن پشته سازی با فرا مدل ماشین بردار پشتیبان حداکثر دقت شناسایی ۹۷ ٪ حاصل شد. انجمن [۸]، با استفاده از فن رأی‌گیری، دقت شناسایی ۹۶/۶۶ ٪ را برای این کد نابسامان به دست آورد. جین و ساشا [۹]، با به‌کارگیری فن پشته سازی دقت شناسایی ۸۶ ٪ را به دست آوردند. دوانگان و همکاران [۱۱]، با استفاده از الگوریتم جنگل تصادفی به بیشینه دقت شناسایی ۹۷/۸۸ ٪ دست پیدا کردند. خلیل و نهیز [۱۲]، با به‌کارگیری الگوریتم گرادیان تقویتی دقت ۹۸ ٪ را برای شناسایی کد نابسامان کلاس بزرگ به دست آوردند. با اعمال روش پیشنهادی نیز، بیشینه دقت شناسایی ۹۷/۶۲ ٪ برای این کد نابسامان حاصل شد.

از بین تحقیقات انجام پذیرفته در زمینه شناسایی کد نابسامان کلاس داده، فونتانا و همکاران [۳]، با استفاده از نسخه بهبودیافته الگوریتم J48 به بیشینه دقت ۹۹/۰۲ ٪ برای شناسایی این کد نابسامان دست پیدا کردند. کریمی و خسروی [۶]، با اعمال فن یادگیری گروهی رأی‌گیری، دقت شناسایی ۹۸/۵۰ ٪ را به دست آوردند. در تحقیق العذبه و الجمن [۷]، با اعمال فن پشته سازی با فرا مدل رگرسیون لجستیک نیز، حداکثر دقت شناسایی ۹۸/۹۲ ٪ به دست آمد. الجمن [۸]، با استفاده از الگوریتم درخت تصمیم‌گیری دقت شناسایی ۹۸/۶۴ ٪ را کسب کرد. در تحقیق جین و ساشا [۹]، دقت شناسایی ۹۰/۱۰ ٪ با به‌کارگیری الگوریتم جنگل تصادفی حاصل شد. در تحقیق دوانگان و همکاران [۱۱]، با استفاده از الگوریتم جنگل تصادفی دقت شناسایی ۹۸/۹۴ ٪ به دست آمد. درنهایت، خلیل و نهیز [۱۲]، با استفاده از الگوریتم گرادیان تقویتی دقت ۹۹ ٪ را برای شناسایی این کد نابسامان کسب کردند. در قیاس با تحقیقات مرتبط، در این مقاله نیز حداکثر دقت شناسایی ۹۹/۷۶ ٪ حاصل شد.

در میان تحقیقات انجام‌شده در راستای شناسایی کد نابسامان متد طولانی، فونتانا و همکاران [۳]، برای شناسایی این کد نابسامان به حداکثر دقت ۹۹/۴۳ ٪ با استفاده از نسخه بهبودیافته الگوریتم J48 دسته یافتند. کریمی و خسروی [۶]، با استفاده از فن رأی‌گیری دقت شناسایی ۹۵ ٪ را به دست آوردند. به همین ترتیب، در تحقیق العذبه و الجمن [۷]، با اعمال فن پشته سازی با فرا مدل ماشین بردار پشتیبان دقت شناسایی ۹۹/۲۴ ٪ حاصل شد. الجمن [۸]، با به‌کارگیری الگوریتم درخت تصمیم‌گیری دقت شناسایی ۹۹/۰۹ ٪ را کسب کرد. جین و ساشا [۹]، دقت شناسایی ۸۶ ٪ را با استفاده از الگوریتم رگرسیون لجستیک به دست آوردند. در تحقیق دوانگان و همکاران [۱۱]، با اعمال الگوریتم‌های جنگل تصادفی و رگرسیون لجستیک دقت شناسایی ۹۹/۵۲ ٪ حاصل شد. خلیل و نهیز [۱۲]، با به‌کارگیری الگوریتم درخت تصمیم‌گیری بیشینه دقت شناسایی ۹۹/۴۱ ٪ را کسب کردند. با اعمال روش پیشنهادی نیز، حداکثر دقت ۹۹/۷۲ ٪ برای شناسایی کد نابسامان متد طولانی به دست آمد.

در راستای شناسایی کد نابسامان ویژگی حسادت، در میان تحقیقات مرتبط، فونتانا و همکاران [۳]، با استفاده از نسخه بهبودیافته الگوریتم JRip به حداکثر دقت شناسایی ۹۶/۶۴ ٪ رسیدند. کریمی و خسروی [۶]، با استفاده از فن یادگیری گروهی رأی‌گیری دقت شناسایی ۹۳/۵۰ ٪ به دست آوردند. در تحقیق العذبه و الجمن [۷]، با اعمال فن پشته سازی با فرا مدل رگرسیون لجستیک دقت شناسایی ۹۵/۳۸ ٪ کسب شد. الجمن

[۸]، با استفاده از فن رأی‌گیری برای شناسایی این کد نابسامان دقت ۹۵/۰۵ ٪ را به دست آورد. جین و ساشا [۹]، با به‌کارگیری الگوریتم بیز ساده گاوسی دقت شناسایی ۸۶/۰۶ ٪ را کسب کردند. دوانگان و همکاران [۱۱]، با استفاده از الگوریتم درخت تصمیم‌گیری دقت شناسایی ۹۸/۲۱ ٪ی را برای شناسایی این کد نابسامان به دست آوردند. خلیل و نهیز [۱۲]، با به‌کارگیری الگوریتم گرادیان تقویتی به‌دقت شناسایی ۹۵ ٪ رسیدند. در مقابل، دقت ۹۷/۱۴ ٪ برای شناسایی کد نابسامان ویژگی حسادت در این مقاله حاصل شد.

همان‌طور که گفته شد، فونتانا و همکاران [۳]، مجموعه داده مربوط به کد نابسامان فهرست طولانی پارامترها را به‌عنوان بخشی از ضمیمه تحقیق خود ارائه کردند. العذبه و الجمن [۷]، برای شناسایی این کد نابسامان با استفاده از الگوریتم فرآیند گاوسی به حداکثر دقت ۹۲/۴۰ ٪ دست یافتند. در تحقیق الجمن [۸]، با اعمال فن یادگیری گروهی رأی‌گیری دقت شناسایی ۹۱/۸۳ ٪ حاصل شد. در قیاس با نتایج آن‌ها، با استفاده از روش پیشنهادی نیز برای کد نابسامان فهرست طولانی پارامترها دقت شناسایی ۹۴/۵۰ ٪ به دست آمد.

به مشابه کد نابسامان فهرست طولانی پارامترها، کد نابسامان گزاره‌های تعویض نیز توسط فونتانا و همکاران [۳]، به‌عنوان ضمیمه پژوهش ارائه شد. العذبه و الجمن [۷]، با به‌کارگیری الگوریتم فرآیند گاوسی به‌دقت ۸۸/۸۹ ٪ برای شناسایی این کد نابسامان دست یافتند. الجمن [۸]، با اعمال فن یادگیری گروهی رأی‌گیری بیشینه دقت شناسایی ۸۷/۸۱ ٪ را به دست آورد. در این مقاله نیز، برای کد نابسامان گزاره‌های تعویض حداکثر دقت شناسایی ۹۰/۰۱ ٪ حاصل شد.

مقایسه نتایج حاصل‌شده با نتایج مقالات مرتبط نشان‌دهنده تطابق و یا عملکرد بهتر روش پیشنهادی، برای شناسایی دو کد نابسامان متد طولانی و کلاس داده، در معیار دقت است. برای شناسایی دو کد نابسامان ویژگی حسادت و کلاس بزرگ نیز عملکرد بالای ۹۷ ٪ در معیار دقت حاصل شد که نتایج بهتری نسبت به اکثر مقالات مرتبط دارد. همچنین، برای شناسایی کدهای نابسامان فهرست طولانی پارامترها و گزاره‌های تعویض نیز نتایجی بهتری نسبت به مقالات مرتبط به‌دست‌آمده است. خلاصه مقایسه نتایج حاصل‌شده با مقالات مرتبط را می‌توان در جداول شماره (۷) و جدول شماره (۸) مشاهده کرد.

جدول (۷). بخش اول نتایج مربوط به مقایسه روش پیشنهادی با مقالات مرتبط

ردیف	نویسنده	کد نابسامان							
		کلاس بزرگ		متد طولانی		ویژگی حسادت		کلاس داده	
		دقت (٪)	الگوریتم	دقت (٪)	الگوریتم	دقت (٪)	الگوریتم	دقت (٪)	الگوریتم
۱	فونتانا و همکاران [۳]	۹۷/۵۵	بیز ساده	۹۹/۴۳	J48	۹۶/۶۴	JRip	۹۹/۰۲	J48
۲	کریمی و خسروی [۶]	۹۳	فن رأی‌گیری	۹۵	فن رأی‌گیری	۹۳/۵۰	فن رأی‌گیری	۹۸/۵۰	فن رأی‌گیری
۳	العذبه و الجمن [۷]	۹۷	پشته سازی با فرا مدل SVM	۹۹/۲۴	پشته سازی با فرا مدل SVM	۹۵/۳۸	پشته سازی با فرا مدل رگرسیون لجستیک	۹۸/۹۲	پشته سازی با فرا مدل رگرسیون لجستیک
۴	الجمن [۸]	۹۶/۶۶	فن رأی‌گیری	۹۹/۰۹	درخت تصمیم‌گیری	۹۵/۰۵	فن رأی‌گیری	۹۸/۶۴	درخت تصمیم‌گیری
۵	جین و ساشا [۹]	۸۶	پشته سازی	۸۶	رگرسیون لجستیک	۸۶/۰۶	بیز ساده گاوسی	۹۰/۱۰	جنگل تصادفی
۶	دوانگان و همکاران [۱۱]	۹۷/۸۸	جنگل تصادفی	۹۹/۵۲	جنگل تصادفی	۹۸/۲۱	درخت تصمیم‌گیری	۹۸/۹۴	جنگل تصادفی
۷	خلیل و نهیز [۱۲]	۹۸	گرادیان تقویتی	۹۹/۴۱	درخت تصمیم‌گیری	۹۵	گرادیان تقویتی	۹۹	گرادیان تقویتی
۸	روش پیشنهادی	۹۷/۶۲	پشته سازی با فرا مدل MLP	۹۹/۷۲	پشته سازی با فرا مدل MLP	۹۷/۱۴	پشته سازی با فرا مدل MLP	۹۹/۷۶	پشته سازی با فرا مدل MLP

جدول (۸). بخش دوم مقایسه نتایج حاصل از روش پیشنهادی با مقالات مرتبط

ردیف	نویسنده	کد نابسامان			
		فهرست طولانی پارامترها		گزاره‌های تعویض	
		دقت (٪)	الگوریتم	دقت (٪)	الگوریتم
	العذبه و الجمن [۷]	۹۲/۴۰	پشته سازی با فرا مدل SVM	۸۸/۸۹	پشته سازی با فرا مدل SVM
۱	الجمن [۸]	۹۱/۸۳	فن رأی‌گیری	۸۷/۸۱	درخت تصمیم‌گیری
۲	روش پیشنهادی	۹۴/۵۰	پشته سازی با فرا مدل MLP	۹۰/۰۱	پشته سازی با فرا مدل MLP

۵-۱-۲. مقایسه با مدل‌های یادگیری عمیق

در این بخش، به مقایسه روش پیشنهادی با الگوریتم های یادگیری عمیق می پردازیم. در این راستا، مقایسه عملکرد روش پیشنهادی با الگوریتم‌های یادگیری عمیق شبکه عصبی پیچشی[1]، شبکه عصبی مصنوعی[2] و شبکه عصبی بازگشتی[3] به‌عنوان کاندید از میان الگوریتم‌های یادگیری عمیق انجام می‌شود. نتایج مربوط به این مقایسه را می‌توان در جدول شماره (۹) مشاهده کرد. در این جدول، مقایسه بر اساس معیارهای عملکردی دقت و p-value انجام‌شده است. با توجه به نتایج جدول شماره (۹)، بهترین عملکرد از میان الگوریتم‌های یادگیری عمیق برای شبکه عصبی مصنوعی به‌دست‌آمده است. این الگوریتم برای هر شش کد نابسامان کلاس بزرگ، کلاس داده، متد طولانی، ویژگی حسادت، فهرست طولانی پارامترها و گزاره‌های تعویض عملکرد بهتری نسبت به دو الگوریتم شبکه عصبی مصنوعی و شبکه عصبی بازگشتی نشان داده است.

مقدار p-value برای شبکه عصبی مصنوعی بزرگ‌تر از ۰/۰۵ است و بر اساس آن نمی‌توان به‌طورقطع درباره برتری عملکرد مدل پیشنهادی اظهارنظر کرد. از سوی دیگر، روش پیشنهادی در برخی از موارد ازجمله در شناسایی کدهای نابسامان کلاس بزرگ، ویژگی حسادت، فهرست طولانی پارامترها و گزاره‌های تعویض عملکرد بهتری در معیار دقت دارد. اما برای شبکه عصبی پیچشی، مقدار p-value در اکثر موارد به‌جز برای کد نابسامان کلاس بزرگ، مقدار آن کوچک‌تر از ۰/۰۵ است که عملکرد بهتر وبرتری معنایی روش پیشنهادی نسبت به این الگوریتم را نشان می‌دهد. در مورد الگوریتم شبکه عصبی بازگشتی نیز مقدار p-value در اکثر موارد به‌جز برای شناسایی کدهای نابسامان کلاس بزرگ و متد طولانی، مقداری بزرگ‌تر از ۰/۰۵ دارد که برتری عملکرد روش پیشنهادی نسبت به این الگوریتم را نشان می‌دهد.

۵-۲. تحلیل هزینه محاسباتی

روش پیشنهادی بر روی یک سیستم کامپیوتری مجهز به سیستم‌عامل لینوکس با مشخصات سخت‌افزاری شامل ۱۲ گیگابایت حافظه اصلی، ۱۰۰ گیگابایت حافظه ثانویه و یک کارت گرافیک ۱۶ گیگابایتی اجرا شد. هزینه‌های زمانی مراحل مختلف شامل پیش‌پردازش اولیه، انتخاب ویژگی، آموزش و ارزیابی مدل یادگیری ماشین و همچنین زمان کل اجرا در جدول شماره (۱۰)، ارائه‌شده است. بر اساس نتایج به‌دست‌آمده، بیشینه هزینه محاسباتی نسبت به هزینه کل در مرحله آموزش و ارزیابی مدل یادگیری ماشین انجام می‌شود. در مرحله بعد نیز هزینه محاسباتی بخش فرآیند انتخاب ویژگی قرار دارد که نسبت به هزینه مرحله آموزش و ارزیابی مدل یادگیری ماشین کمتر است، ولی تا حدودی قابل‌توجه است. بخش پیش‌پردازش اولیه نیز هزینه محاسباتی کمتری برحسب واحد زمانی نسبت به هزینه محاسباتی کل دارد.

۵-۳. تهدیدات اعتبار تحقیق

تهدیداتی[4] وجود دارند که ممکن است، اعتبار این تحقیق را تحت تأثیر قرار دهند. این تهدیدات در دسته‌بندی‌های تهدیدات داخلی[5]، خارجی[6] و ساختاری[7] تقسیم‌بندی می‌شوند که در ادامه به این تهدیدات و راه‌حلی که جهت مواجه با آن‌ها اتخاذ گردیده است، پرداخته می‌شود.

تهدید ساختاری که اعتبار تحقیق را مورد تأثیر قرار می‌دهد، احتمال رخداد بیش برازش و یا کم برازش شدن مدل یادگیری ماشین است. راه‌حلی که جهت مواجه با آن اتخاذ شد، استفاده از روش اعتبارسنجی متقابل ۱۰ بخشی است. استفاده از این روش می‌تواند تا حدی این مشکل را برطرف سازد.

[1] Convolutional Neural Network (CNN)
[2] Artificial Neural Networks (ANN)
[3] Recurrent Neural Networks (RNN)
[4] Threats to research validity
[5] Threats to internal validity
[6] Threats to external Validity
[7] Threats to structural validity

جدول (۹). جدول مربوط به نتایج مقایسه الگوریتم‌های یادگیری عمیق و روش پیشنهادی

کد نابسامان	کلاس بزرگ	کلاس داده	متد طولانی	ویژگی حسادت	فهرست طولانی پارامترها	گزاره‌های تعویض
الگوریتم	**معیار (دقت ٪)**					
شبکه عصبی پیچشی	۹۴/۲۳ ± ۲/۹۱	۹۶/۰۳ ± ۲/۳۵	۹۶/۲۳ ± ۲/۲۴	± ۲/۳۸ ۸۸/۲۸	۸۶/۳۴ ± ۳/۴۶	۸۴/۹۸ ± ۴/۶۵
شبکه عصبی مصنوعی	۹۶/۴۴ ± ۱/۳۳	۹۹/۰۱ ± ۰/۲۳	۹۹/۱۱ ± ۰/۲۵	۹۶/۱۱ ± ۱/۹۳	۹۰/۴۳ ± ۲/۷۲	۸۸/۹۰ ± ۳/۶۳
شبکه عصبی بازگشتی	۹۵/۷۲ ± ۲/۸۸	۹۷/۱۳ ± ۲/۲۱	۹۷/۷۳ ± ۲/۲۷	۸۹/۰۱ ± ۲/۳۳	۸۷/۵۶ ± ۳/۳۷	۸۶/۰۱ ± ۴/۶۱
روش پیشنهادی	۹۷/۶۲ ± ۱/۲۵	۹۹/۷۶ ± ۰/۱۴	۹۹/۷۸ ± ۰/۱۹	۹۷/۳۸ ± ۱/۲۵	۹۴/۵ ± ۲/۳۵	۹۰/۰۱ ± ۳/۴۱
الگوریتم	**معیار (p-value)**					
شبکه عصبی پیچشی	۰/۱۲	۰/۰۳۳	۰/۰۴۹	۰/۰۱	۰/۰۲۹	۰/۰۲۷
شبکه عصبی مصنوعی	۰/۵۲	۰/۶۳	۰/۶۵	۰/۷۱	۰/۰۷۲	۰/۵۹
شبکه عصبی بازگشتی	۰/۲۲	۰/۰۴۹	۰/۰۵۱	۰/۰۳۸	۰/۰۳۹	۰/۰۳۷

جدول (۱۰) . جدول هزینه محاسباتی اجرای روش پیشنهادی

کد نابسامان	پیش‌پردازش (s)	فرآیند انتخاب ویژگی (s)	آموزش و ارزیابی مدل یادگیری ماشین (s)	زمان کل (s)
ویژگی حسادت	۰/۰۳	۸۰/۱۴	۲۴۶/۰۲	۳۴۲/۱۵
متد طولانی	۰/۲۵	۸۷/۰۴	۲۳۲/۳۷	۳۳۱/۷۲
کلاس بزرگ	۰/۰۳	۲۷/۷۵	۲۴۹/۷۸	۲۸۹/۴۷
کلاس داده	۰/۰۲	۳۰/۴۵	۲۷۹/۲۶	۳۲۲/۰۶
فهرست طولانی پارامترها	۰/۳۰	۴۶/۹۰	۲۴۷/۸۵	۳۰۸/۴۷
گزاره‌های تعویض	۰/۰۵	۴۷/۲۲	۲۶۵/۸۸	۳۲۵/۴۰

یکی از تهدیدات خارجی اعتبار تحقیق، به مجموعه داده مورداستفاده مرتبط می‌شود. این مجموعه داده از تحلیل و بررسی پروژه‌های توسعه‌یافته با زبان جاوا تولیدشده است. این احتمال وجود دارد که روش پیشنهادی بر روی مجموعه داده‌های حاصل از پروژه‌های توسعه‌یافته با سایر زبان‌های برنامه‌نویسی مانند پایتون کارآمد نباشد. تهدید خارجی دیگر این است که مجموعه داده مورداستفاده از تحلیل کدهای نرم‌افزاری یک مجموعه نرم‌افزاری خاص به وجود آمده است. محتمل است که روش پیشنهادی بر روی مجموعه داده‌های حاصل از سایر پروژه‌های نرم‌افزاری متفاوت عمل کند.

از تهدیدات داخلی اعتبار پژوهش به مسئله کمبود تعداد داده‌های در دسترس برای آموزش مدل یادگیری ماشین مبتنی بر فن پشته سازی، مرتبط است. مجموعه داده مورداستفاده برای هر یک از شش کد نابسامان موردمطالعه شامل ۴۲۰ نمونه داده است. از این ۴۲۰ نمونه داده، ۱۴۰ عدد نمونه مثبت و ۲۸۰ نمونه منفی است. تعداد نمونه‌های منفی به تعداد نمونه‌های مثبت به نسبت ۲ به ۱ است که نسبت میان آن‌ها را مناسب نشان می‌دهد و می‌توان مجموعه داده را متعادل در نظر گرفت و مجموعه داده موجود به متعادل‌سازی نیاز ندارد. اما مسئله اصلی، تعداد داده‌های موردنیاز برای آموزش و آزمون مدل یادگیری ماشین مبتنی بر فن پشته سازی است. برای آموزش مدل یادگیری

ماشین مذکور نیاز است که داده‌های آموزشی به دو بخش تقسیم شوند. یک بخش آن برای آموزش مدل‌های پایه و بخش دیگر آن برای آزمون مدل‌های پایه جهت تولید داده آموزشی برای آموزش فرا مدل استفاده می‌شود. درصورتی‌که داده‌های آموزشی اولیه به دو بخش تقسیم شوند، احتمال دارد به علت کمبود داده‌های آموزشی، مدل یادگیری ماشین دچار مشکل کم برازش شود. همچنین، درصورتی‌که از داده‌های آموزشی به‌صورت مجدد برای آزمون مدل‌های پایه استفاده شود، مدل یادگیری ماشین دچار بیش برازش می‌شود. برای حل این مسئله، روش اعتبارسنجی متقابل ۱۰-بخشی برای آموزش و آزمون آن مورداستفاده قرارگرفته است. به‌کارگیری این روش می‌تواند علاوه بر جلوگیری از احتمال رخداد بیش برازش و یا کم برازش، تا حدودی اثر کمبود داده‌ها را جبران نماید. یک تهدید داخلی دیگر، در مورد ویژگی‌ها و معیارهای نرم‌افزاری مجموعه داده به‌کاررفته است. همان‌طور که گفته شد، مجموعه داده مورداستفاده در این تحقیق توسط فونتانا و همکاران [۳]، با تحلیل ۷۴ پروژه نرم‌افزاری از مجموعه داده نرم‌افزاری Qualitas Corpus تولیدشده است. این مجموعه داده برای شش کد نابسامان موردمطالعه در هر یک از سطوح کلاس و متد، تعدادی ویژگی مشخص بر اساس معیارهای نرم‌افزاری دارد. تمامی این ویژگی‌ها بر شناسایی برچسب متغیر هدف تأثیرگذار نیستند.

ممکن است که وجود آن‌ها باعث کاهش دقت مدل یادگیری ماشین در پیش‌بینی نهایی و شناسایی کدهای نابسامان شود. برای حل این مسئله از الگوریتم‌های انتخاب ویژگی پوششی به‌عنوان انتخاب‌گرهای پایه استفاده‌شده است. الگوریتم‌های پوششی اغلب ویژگی‌هایی را برای فرآیند یادگیری انتخاب می‌کنند که بیشترین تأثیر را بر عملکرد مدل یادگیری ماشین می‌گذارند. به همین دلیل، برای رفع این مسئله نسبت به سایر روش‌ها مناسب‌تر به نظر می‌آیند.

۶. نتیجه‌گیری و کارهای آینده

کدهای نابسامان یا بوهای کد، یکی از مشکلات ساختاری کد برنامه‌های نرم‌افزاری هستند که اغلب از عدم رعایت اصول مهندسی نرم‌افزار یا برنامه‌نویسی ضعیف به وجود می‌آیند. وجود آن‌ها در برنامه می‌تواند منجر به افزایش رخداد خطا، کاهش قابلیت نگهداری و توسعه برنامه، افزایش هزینه‌های نگهداری برنامه و درنهایت کاهش کیفیت نرم‌افزار شود.

برای جلوگیری از مشکلات احتمالی یادشده، نیاز به شناسایی دقیق‌تر کدهای نابسامان و بازآرایی مجدد برنامه برای رفع اثر آن‌ها وجود دارد. تاکنون روش‌های مختلفی جهت خودکارسازی شناسایی کدهای نابسامان ارائه‌شده است.

استفاده از فنون و الگوریتم‌های یادگیری ماشین یکی از روش‌های نوینی است که جهت شناسایی خودکار و دقیق کدهای نابسامان پیشنهاد می‌شود. راه‌حل‌های مختلفی جهت افزایش دقت در شناسایی کدهای نابسامان به‌وسیله مدل‌های یادگیری ماشین وجود دارد. استفاده از ترکیب مناسبی از فنون انتخاب ویژگی و یادگیری ماشین ازجمله روش‌هایی است که می‌تواند به پیشبرد این هدف کمک نماید.

از بین روش‌های انتخاب ویژگی پوشش، فیلتر و تعبیه‌شده، الگوریتم‌های انتخاب ویژگی مبتنی بر پوشش نسبت به سایر روش‌ها در بهبود دقت مدل یادگیری ماشین می‌توانند مؤثرتر عمل کنند. در این روش ویژگی‌ها بر اساس تأثیری که بر عملکرد مدل یادگیری ماشین می‌گذارند، انتخاب می‌شوند. در همین راستا، این الگوریتم‌ها اغلب ویژگی‌هایی را انتخاب می‌کنند که تأثیر بهتری بر بهبود عملکرد مدل یادگیری ماشین می‌گذارند. لذا در این مقاله، از الگوریتم‌های مبتنی بر پوشش شامل؛ ژنتیک، انتخاب پیش‌رونده متوالی، حذف ویژگی بازگشتی و حذف ویژگی بازگشتی با اعتبارسنجی متقابل برای انتخاب ویژگی استفاده‌شده است.

همچنین، استفاده از فن انتخاب ویژگی مجموعه‌ای، علاوه بر استفاده از الگوریتم‌های انتخاب ویژگی پوششی می‌تواند بر بهبود دقت مدل یادگیری ماشین تأثیرگذار باشد. در این فن با استفاده از یک سازوکار مناسب می‌توان ویژگی‌های مؤثرتری را نسبت به انتخاب‌گرهای منفرد، انتخاب کرد.

علاوه بر استفاده از فنون و الگوریتم‌های انتخاب ویژگی مذکور، می‌توان با به‌کارگیری الگوریتم‌ها و فنون یادگیری ماشین مناسب به بهبود دقت در شناسایی کدهای نابسامان رسید. فنون یادگیری گروهی ازجمله فنون یادگیری ماشین هستند که با ترکیب نتایج چند مدل یادگیری ماشین با عملکرد پایین می‌توان یک مدل یادگیری با عملکرد بهتر و دقیق‌تر نسبت به مدل‌های یادگیری ماشین رایج ایجاد کرد.

از میان فنون یادگیری گروهی رایج، فن پشته سازی مناسب‌تر از سایر فنون به نظر آمد. در این فن نتایج چند مدل یادگیری ماشین پایه با الگوریتم‌های متفاوت توسط فرا مدل باهم ترکیب می‌شود. انعطاف‌پذیری این فن در استفاده از ترکیب الگوریتم‌های مختلف، باعث استفاده از مزایای هر یک از الگوریتم‌های به‌کاررفته در جهت غلبه بر ضعف‌های سایر الگوریتم‌ها می‌شود.

به همین جهت، در این مقاله روشی مبتنی بر ترکیب فنون انتخاب ویژگی مجموعه‌ای مبتنی بر استفاده از الگوریتم‌های انتخاب ویژگی پوششی و یادگیری گروهی پشته سازی برای

بهبود دقت در شناسایی کدهای نابسامان پیشنهادشده است. این امر با تمرکز بر شناسایی کدهای نابسامان کلاس بزرگ، کلاس داده، متد طولانی، ویژگی حسادت، فهرست طولانی پارامترها و گزاره‌های تعویض انجام گردید.

فرآیندهای آموزش و آزمون مدل یادگیری ماشین نیز با استفاده از روش اعتبارسنجی متقابل ۱۰-بخشی انجام شد. نتایج حاصل از روش پیشنهادی باحالت‌های بدون اعمال فن پشته سازی و با مقالات مرتبط نیز مقایسه شد.

بر اساس نتایج حاصل‌شده، بهترین عملکرد برای شناسایی کدهای نابسامان کلاس داده و متد طولانی با عملکرد حدود ۹۹ ٪، در معیار دقت به دست آمد. مقایسه این نتایج با نتایج مقالات مرتبط نشان‌دهنده عملکرد بهتر یا تطابق عملکرد روش پیشنهادی برای این دو کد نابسامان نسبت به مقالات مرتبط است.

همچنین، برای کدهای نابسامان کلاس بزرگ و ویژگی حسادت نیز دقت شناسایی حدود ۹۷ ٪ حاصل شد. مقایسه نتایج به‌دست‌آمده برای این دو کد نابسامان با نتایج مقالات مرتبط نشان‌دهنده عملکرد بهتر روش پیشنهادی برای شناسایی این دو کد نابسامان نسبت به اکثر مقالات مرتبط را نشان می‌دهد.

برای کدهای نابسامان فهرست طولانی پارامترها و گزاره‌های تعویض نیز در معیار دقت عملکرد بهتری در مقایسه با مقالات مرتبط داشته‌ایم. همچنین، برای این دو کد نابسامان به ترتیب حدود ۲ و ۳ ٪ در معیار دقت بهبود عملکرد حاصل‌شده است.

مقایسه نتایج حاصل از روش پیشنهادی و نتایج مربوط به الگوریتم‌های یادگیری به‌کاررفته در مدل‌های پایه، تأثیر فن پشته سازی بر بهبود عملکرد مدل یادگیری ماشین مورداستفاده درروش پیشنهادی را نشان می‌دهد.

به‌عنوان کارهای آینده نیز، تحقیق در راستای ترکیب فن انتخاب ویژگی مجموعه‌ای با سایر فنون یادگیری گروهی پیشنهاد می‌شود. همچنین، می‌توان به تحقیق در مورداستفاده از روش پیشنهادی برای شناسایی سایر کدهای نابسامان علاوه بر شش کد نابسامان موردمطالعه در این مقاله اقدام کرد.

می‌توان مجموعه داده‌ای شامل شش کد نابسامان مذکور بر روی پروژه‌های توسعه‌یافته با سایر زبان‌های برنامه‌نویسی مانند پایتون یا سی شارپ را تولید کرد. سپس، روش پیشنهادی را جهت شناسایی آن‌ها بر روی مجموعه داده جدید ارزیابی نمود. این کار به جهت بررسی تعمیم‌پذیری روش پیشنهادی بر روی پروژه‌های توسعه‌یافته با سایر زبان‌های برنامه‌نویسی انجام می‌شود.

همچنین، می‌توان از سایر پروژه‌های توسعه‌یافته با زبان جاوا برای ساخت یک مجموعه داده جدید، شامل این شش کد نابسامان استفاده کرد. این امر در جهت بررسی امکان تعمیم روش پیشنهادی برای شناسایی کدهای نابسامان مذکور بر روی سایر پروژه‌های نرم‌افزاری اجرا می‌گردد.

۷. مراجع

[1] J. A. M. Santos, J. B. Rocha-Junior, L. C. L. Prates, R. S. Do Nascimento, M. F. Freitas, and M. G. De Mendonça, "A systematic review on the code smell effect," Journal of Systems and Software, vol. 144, pp. 450-477, 2018. https://doi. rg/10.1016/j.jss.2018.07.035

[2] G. Rasool and Z. Arshad, "A review of code smell mining techniques," Journal of Software: Evolution and Process, vol. 27, no. 11, pp. 867-895, 2015. https://doi.org/10.1002/smr.1737

[3] F. Arcelli Fontana, M. V. Mäntylä, M. Zanoni, and A. Marino, "Comparing and experimenting machine learning techniques for code smell detection," Empirical Software Engineering, vol. 21, pp. 1143-1191, 2016. https://doi. rg/10.1007/s10664-015-9378-4

[4] A. Karimi and F. Karimi, "A method to prediction of software system's code smells using neural network," Electronic and Cyber Defense, vol. 11, no. 3, pp. 67-76, 2023 (In Persian). [Online]. Available: https://ecdj.ihu.ac.ir/article_208726_en.html

[5] I. Kaur and A. Kaur, "A novel four-way approach designed with ensemble feature selection for code smell detection," IEEE Access, vol. 9, pp. 8695-8707, 2021. doi: https://doi.org/10.1109/access.2021.3049823.

[6] Karimi.A and Khosravi.M, "Improving the accuracy of code smell identification using the gray wolf algorithm based on machine learning techniques and majority voting ", Scientific Journal of Electronic & Cyber Defense, vol. 12, no. 1, pp. 109-122, 2024 (In Persian). https://dor.isc.ac/dor/20.1001.1.23224347.1403.12.1.9.7

[7] A. Alazba and H. Aljamaan, "Code smell detection using feature selection and stacking ensemble: An empirical investigation," Information and Software Technology, vol. 138, p. 106648, 2021. https://doi. rg/10.1016/j.infsof.2021.106648

[8] H. Aljamaan, "Voting heterogeneous ensemble for code smell detection," in 2021 20th IEEE international conference on machine learning and applications (ICMLA), 2021: IEEE, pp. 897-902. https://doi. rg/10.1109/ICMLA52953.2021.00148

[9] S. Jain and A. Saha, "Improving performance with hybrid feature selection and ensemble machine learning techniques for code smell detection," Science of Computer Programming, vol. 212, p. 102713, 2021. https://doi. rg/10.1016/j.scico.2021.102713

[10] L. Shen, W. Liu, X. Chen, Q. Gu, and X. Liu, "Improving machine learning-based code smell detection via hyper-parameter optimization," in 2020 27th Asia-Pacific Software Engineering Conference (APSEC), 2020:

IEEE, pp. 276-285. https://doi. rg/10.1109/APSEC51365.2020.00036

[11] S. Dewangan, R. S. Rao, A. Mishra, and M. Gupta, "A novel approach for code smell detection: an empirical study," IEEE Access, vol. 9, pp. 162869-162883, 2021. https://doi. rg/10.1109/ACCESS.2021.3133810

[12] N. A. A. Khleel and K. Nehéz, "Detection of code smells using machine learning techniques combined with data-balancing methods," International Journal of Advances in Intelligent Informatics, vol. 9, no. 3, pp. 402-417, 2023.https://doi. rg/10.26555/ijain.v9i3.981

[13] M. Fowler, Refactoring: improving the design of existing code. Addison-Wesley Professional, 2018.

[14] A. Karimi, A. Tolui Far, and F. Karimi, "A Survey and Evaluation of Suspicious Code Types in Software Code Refactoring Process", Passive Defense, vol. 16, no. 1, pp. 11-32, 2025. (In Persian) https://dor.isc.ac/dor/20.1001.1.20086849.1404.16.1.2.6

[15] J. M. Helm, A. M. Swiergosz, H. S. Haeberle, J. M. Karnuta, J. L. Schaffer, V. E. Krebs, P. N. Ramkumar, "Machine learning and artificial intelligence: definitions, applications, and future directions," Curr. Rev. Musculoskelet. Med., vol. 13, pp. 69–76, 2020. https://doi. rg/10.1007/s12178-020-09600-8

[16] S. Seibt, B. V. R. Lipinski, and M. E. Latoschik, "Dense feature matching based on homographic decomposition," IEEE Access, vol. 10, pp. 21236-21249, 2022. https://doi.org/10.1109/ACCESS.2022.3207287

[17] M. A. El Mrabet, K. El Makkaoui, and A. Faize, "Supervised machine learning: A survey," in 2021 4th International Conference on Advanced Communication Technologies and Networking (CommNet), 2021: IEEE, pp. 1-10. https://doi.org/10.1109/CommNet52204.2021.9641998

[18] L. B. Almeida, "Multilayer perceptrons," in Handbook of Neural Computation: CRC Press, 2020, pp. C1. 2: 1-C1. 2: 30. [Online]. Avaliable: http://www.lx.it.pt/~lbalmeida/papers/AlmeidaHNC.pdf

[19] H. Taud and J.-F. Mas, "Multilayer perceptron (MLP)," in *Geomatic Approaches for Modeling Land Change Scenarios*, 2018, pp. 451-455. https://doi. rg/10.1007/978-3-319-60801-3_27

[20] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," Neurocomputing, vol. 408, pp. 189-215, 2020. https://doi. rg/10.1016/j.neucom.2019.10.118

[21] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A comparative analysis of logistic regression, random forest and KNN models for the text classification," Augmented Human Research, vol. 5, no. 1, p. 12, 2020. https://doi. rg/10.1007/s41133-020-00032-0

[22] A. Das, "Logistic Regression," Springer eBooks, pp. 3985–3986, Jan. 2023. doi: https://doi.org/10.1007/978-3-031-17299-1_1689.

[23] P. Xanthopoulos, P. M. Pardalos, T. B. Trafalis, P. Xanthopoulos, P. M. Pardalos, and T. B. Trafalis, "Linear discriminant analysis," Robust data mining, pp. 27-33, 2013.https://doi. rg/10.1007/978-1-4419-9878-1_4

[24] T. N. A. Nguyen, A. Bouzerdoum, and S. L. Phung, "A scalable hierarchical Gaussian process classifier," IEEE Transactions on Signal Processing, vol. 67, no. 11, pp. 3042-3057, 2019. https://doi. rg/10.1109/TSP.2019.2911251

[25] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," ACM Comput. Surv., vol. 50, no. 6, pp. 1–45, 2017. https://doi. rg/10.1145/3136625

[26] P. Misra and A. S. Yadav, "Improving the classification accuracy using recursive feature elimination with cross-validation," Int. J. Emerg. Technol, vol. 11, no. 3, pp. 659-665, 2020. [Online]. Avaliable: https://www.researchgate.net/profile/Puneet-Misra-3/publication/344181117_Improving_the_Classification_Accuracy_using_Recursive_Feature_Elimination_with_Cross-Validation/links/5f59a4124585154dbbc40337/Improving-the-Classification-Accuracy-using-Recursive-Feature-Elimination-with-Cross-Validation.pdf

[27] Y. B. Wah, N. Ibrahim, H. A. Hamid, S. Abdul-Rahman, and S. Fong, "Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy," Pertanika Journal of Science & Technology, vol. 26, no. 1, 2018. [Online]. Available: http://pertanika2.upm.edu.my/resources/files/Pertanika%20PAPERS/JST%20Vol.%2026%20(1)%20Jan.%202018/21%20JST(S)-0296-2017-3rdProof.pdf

[28] O. H. Babatunde, L. Armstrong, J. Leng, and D. Diepeveen, "A genetic algorithm-based feature selection," 2014. [Online]. Available: https://ro.ecu.edu.au/ecuworkspost2013/653/

[29] V. Bolón-Canedo and A. Alonso-Betanzos, "Ensembles for feature selection: A review and future trends," Information fusion, vol. 52, pp. 1-12, 2019. https://doi.org/10.1016/j.inffus.2018.11.008