ارائه روشی برای تشخیص بیماری کووید-۱۹ بر پایه الگوریتم روابط اجتماعی درختان و طبقه بند نایو بیز

حسین ازگومی [۱]، اعظم عندلیب[۱]*

[۱] گروه مهندسی کامپیوتر، واحد رشت، دانشگاه آزاد اسلامی، رشت، ایران

Hossein.Azgomi@iau.ac.ir , Azam.Andalib@iau.ac.ir

* نویسنده مسئول

چکیده

بیماری کووید-۱۹، که به طور عمده به عنوان کرونا شناخته می‌شود، یک بیماری ویروسی است که توسط ویروس SARS-CoV-2 ایجاد می‌گردد. علائم رایج این بیماری شامل تب، سرفه، احساس خستگی و از دست دادن حس بویایی می‌باشد. روش استاندارد برای تشخیص دقیق کووید-۱۹، آزمایش rRT-PCR که نیازمند نمونه‌برداری تنفسی است که زمان‌بر می‌باشد. بنابراین توسعه روش‌های تشخیصی سریع این بیماری اهمیت زیادی دارد.در این مقاله روشی جدید برای تشخیص کووید-۱۹ با استفاده از هوش مصنوعی معرفی شده است. این روش، که سریع و غیر تهاجمی است، بر اساس الگوریتم روابط اجتماعی درختان (TSR) و طبقه‌بندی نایو بیز طراحی شده است. روش پیشنهادی شامل دو مرحله اصلی انتخاب ویژگی و تشخیص بیماری است. انتخاب ویژگی‌ها با استفاده از الگوریتم TSR و تشخیص بیماری توسط طبقه‌بند نایو بیز انجام می‌شود. روش پیشنهاد شده به صورت عملی با مجموعه داده COVID-19 Dataset بررسی شد. ارزیابی‌های عملی نشان داده‌اند که این روش جدید در تشخیص کووید-۱۹ عملکرد بهتری نسبت به سایر روش‌های موجود دارد و تشخیص این بیماری را به صورت متوسط با دقت ۹۶ درصد، فراخوانی ۹۷ درصد و امتیاز-F1 ۹۶ درصد انجام می‌دهد.

کلمات کلیدی: کووید-۱۹؛ داده‌کاوی؛ طبقه‌بندی؛ الگوریتم روابط اجتماعی درختان؛ نایو بیز

۱. مقدمه

ویروس‌های کرونا به عنوان یک خانواده وسیع از ویروس‌ها شناخته می‌شوند که قادر به ایجاد بیماری در حیوانات و انسان‌ها هستند. انواع مختلفی از این ویروس‌ها که تا به حال شناسایی شده‌اند، می‌توانند باعث بروز بیماری‌های تنفسی متفاوتی در انسان‌ها شوند. از جمله این بیماری‌ها میتوان به سرماخوردگی ساده تا بیماری‌های جدی‌تر مانند سندروم تنفسی خاورمیانه و سندروم تنفسی حاد شدید اشاره کرد [۱]. ویروس جدید کرونا عامل بیماری کووید-۱۹ است. این ویروس پیش از شیوع گسترده‌ای که از دسامبر ۲۰۱۹ در شهر ووهان چین آغاز شد، ناشناخته بود.

علائم اصلی کووید-۱۹ شامل تب، خستگی و سرفه خشک است [۲]. حدود یک ششم از مبتلایان به کووید-۱۹ دچار بیماری شدید می‌شوند و تنگی نفس را تجربه می‌کنند. افراد مسن و کسانی که دارای بیماری‌های زمینه‌ای مانند فشار خون بالا، بیماری‌های قلبی یا دیابت هستند، بیشتر در معرض خطر هستند [۳].

بیماری کووید-۱۹ از افراد بیمار که حامل ویروس هستند، به افراد سالم قابل انتقال است. این بیماری از طریق قطرات تنفسی که هنگام سرفه یا عطسه از فرد مبتلا پخش می‌شود، منتقل می‌شود. این قطرات می‌توانند روی اشیاء و سطوح مختلف نشسته و باعث آلودگی شوند. افراد دیگر با لمس این سطوح و سپس لمس چشم‌ها، بینی یا دهان خود، می‌توانند به ویروس آلوده شوند [۴]. با توجه به سرعت بالای شیوع کووید-۱۹، توسعه روش‌های تشخیصی هوشمند و غیرتهاجمی که به صورت خودکار و با استفاده از داده‌های بیماران قبلی کار می‌کنند، از اهمیت بالایی برخوردار است.

داده‌کاوی فرآیندی است که در آن از تکنیک‌های پیشرفته برای کاوش و تحلیل مجموعه‌های بزرگی از داده‌ها به منظور یافتن الگوهای پنهان و اطلاعات معنادار استفاده می‌شود. داده‌کاوی به دلیل گسترش استفاده از کامپیوتر و فناوری اطلاعات در حوزه‌های مختلف و همچنین پیشرفت‌هایی در زمینه ذخیره‌سازی داده‌ها، اهمیت یافته است. با توجه به حجم عظیم داده‌های موجود، استفاده از روش‌های سنتی برای پردازش و کشف دانش از داده‌ها ناکارآمد و بسیار زمان‌بر است. بنابراین، استفاده از روش‌های مبتنی بر داده‌کاوی که قادر به استخراج سریع و کارآمد این دانش هستند، ضروری است [۵]. داده‌کاوی دارای شاخه‌های متعدد و گوناگونی است که هر کدام برای حل مشکلات خاصی به کار می‌روند و در حوزه‌های مختلفی مانند تجارت، پزشکی، صنعت و کشاورزی کاربرد گسترده دارند.
یکی از شاخه‌های مهم و اصلی داده‌کاوی، طبقه‌بندی است که به عنوان یک نوع پیش‌بینی عمل می‌کند. در طبقه‌بندی، یک مجموعه داده موجود است و هر نمونه داده در آن دارای برچسب است که کلاس مربوط به آن نمونه داده را مشخص می‌کنند [۶]. هدف الگوریتم‌های طبقه‌بندی، یادگیری از این مجموعه داده موجود و پیش‌بینی برچسب نمونه‌های جدید است. الگوریتم نایو بیز یکی از الگوریتم‌های شناخته‌شده در طبقه‌بندی است. این الگوریتم بر اساس نظریه بیز عمل می‌کند و ویژگی‌های داده‌ها را به صورت ساده و مستقل در نظر می‌گیرد [۷]. از نظر کاربرد نیز الگوریتم نایو بیز جز الگوریتم‌های پرکاربرد در حوزه طبقه بندی و داده‌کاوی به حساب می‌آید.
الگوریتم‌های فراابتکاری، تکنیک‌هایی هستند که برای حل مسائل پیچیده در زمان کوتاه‌تر طراحی شده‌اند. این الگوریتم‌ها مناسب مسائلی هستند که روش‌های سنتی نمی‌توانند راه‌حل‌های دقیقی برای آنها ارائه دهند. الگوریتم‌های فراابتکاری با ارائه راه‌حل‌های نزدیک به بهینه، مسائل را در زمان کمتری حل می‌کنند [۸]. الگوریتم TSR[1] [۹] یکی از این الگوریتم‌های فراابتکاری جدید است که عملکرد مناسبی دارد. در این الگوریتم ابتدا یک جمعیت اولیه تصادفی از راه‌حل‌ها تولید شده و سپس به تعدادی زیرجنگل تقسیم می‌شوند. در ادامه این الگوریتم در یک حلقه تکرار از دو فرآیند اصلی زیرجنگل‌ها و فرآیند سراسری برای بهبود جمعیت استفاده می‌کند. در فرآیند زیرجنگل‌ها راه‌حل‌های جدید به کمک سه عملگر تولید می‌شوند و در فرآیند سراسری توازن بین زیرجنگل‌ها حفظ می‌شود [۹]. الگوریتم TSR راه‌حل‌هایی که نرخ رشد بالایی دارند را به عنوان نهال در نظر می‌گیرد. در فرآیند تولید راه‌حل‌ها جدید به نهال‌ها توجه ویژه‌ای می‌شود [۱۰].
این مقاله به دنبال ایجاد یک روش هوشمند برای تشخیص بیماری کووید-۱۹ می‌باشد. روش پیشنهادی بر اساس الگوریتم TSR و الگوریتم نایو بیز توسعه یافته است. روش پیشنهاد شده شامل دو مرحله اصلی می‌باشد. در مرحله اول، فرآیند انتخاب ویژگی‌ها انجام می‌گیرد. این انتخاب ویژگی‌ها به شکل یک مسئله بهینه‌سازی در نظر گرفته شده و با استفاده از الگوریتم فراابتکاری TSR حل می‌شود. در مرحله دوم، تشخیص بیماری با استفاده از یادگیری نظارت شده یا طبقه‌بندی صورت می‌گیرد. در این مرحله، الگوریتم نایو بیز برای طبقه‌بندی و تعیین کلاس بیماری به کار برده می‌شود.
استفاده از داده‌کاوی و انتخاب ویژگی برای تشخیص کووید-۱۹ می‌تواند بهتر از روش‌های مبتنی بر پردازش تصویر و یادگیری عمیق باشد. زیرا این روش‌ها معمولاً با پردازش کمتر و انتخاب ویژگی‌های مهم و مرتبط با هم می‌توانند نتایج دقیقی را با سرعت بیشتری ارائه دهند. اما پردازش تصویر و یادگیری عمیق نیاز به سخت‌افزارهای قدرتمند و منابع محاسباتی بالا دارند که می‌تواند هزینه‌بر باشد. در مقابل، داده‌کاوی و انتخاب ویژگی معمولاً با منابع محاسباتی کمتری قابل انجام هستند. همچنین داده‌کاوی و انتخاب ویژگی می‌توانند با انواع مختلف داده‌ها کار کنند. از جمله آنها میتوان به داده‌های بالینی، آزمایشگاهی و ژنتیکی اشاره کرد. این انعطاف‌پذیری می‌تواند به تشخیص دقیق‌تر و جامع‌تر کمک کند. مشارکت و نوآوری این مقاله را میتوان به صورت موارد زیر مطرح کرد:

[1] Trees Social Relations

- استفاده از الگوریم TSR به صورت چند شروعی[2] جهت انتخاب ویژگی
- استفاده از الگوریتم‌های TSR و نایو بیز به صورت ترکیبی جهت تشخیص بیماری

ادامه مقاله به این صورت سازماندهی شده است. در بخش ۲ راه‌کارهای مرتبط با تحقیق جاری مرور می‌شود. در بخش ۳ روش پیشنهاد شده برای تشخیص بیماری کووید-۱۹ ارائه می‌گردد. در بخش ۴ روش پیشنهادی به صورت عملی مورد ارزیابی قرار می‌گیرد. نتیجه‌گیری حاصل از این تحقیق نیز در بخش ۵ ذکر می‌شود.

۲. راه‌کارهای مرتبط

در سال‌های اخیر، به دنبال شیوع کووید-۱۹، تعداد زیادی مطالعات با هدف توسعه روش‌های هوشمند برای شناسایی این بیماری انجام شده است. اکثر این مطالعات بر استفاده از روش‌های طبقه‌بندی مبتنی بر داده‌کاوی برای تشخیص کووید-۱۹ تمرکز دارند. در ادامه، برخی از پژوهش‌های برجسته در این حوزه معرفی می‌گردد.

روشی بر اساس الگوریتم جنگل تصادفی[3] برای پیش‌بینی بهبود یا مرگ بیماران مبتلا به کووید-۱۹ معرفی شده است که عملکرد مناسبی دارد [۱۱]. این روش از نسخه‌ای تقویت‌شده از الگوریتم جنگل تصادفی استفاده کرده و داده‌های مربوط به جغرافیا، سفر، سلامت و جمعیت بیماران را برای تشخیص به کار برده است. در پژوهش مربوطه دقت روش پیشنهادی ۹۴ درصد گزارش شده است. از مزایای این روش میتوان به دقت بالا، مدیریت داده‌های نامتوازن و استفاده از داده‌های متنوع اشاره کرد. معایب این روش نیز پیچیدگی محاسباتی، نیاز به داده‌هاب با کیفیت بالا و محدودیت در تعمیم‌پذیری می‌باشد. روش دیگری بر پایه الگوریتم ماشین بردار پشتیبان[4] برای تشخیص بهبود یا مرگ ناشی از کووید-۱۹ پیشنهاد شده است که بر داده‌های سری زمانی متکی است [۱۲]. داده‌های استفاده شده در تحقیق مربوطه در یک دوره سه‌ماهه جمع‌آوری شده‌اند. نتایج این مطالعه می‌تواند در برنامه‌ریزی برای کنترل بیماری کووید-۱۹، کمک‌کننده باشد. مزایای اصلی روش پیشنهاد شده دقت بالا و مقیاس‌پذیری مناسب می‌باشد و مشکل اصلی آن نیاز به تنظیم پارامتر دقیق است که امری زمانبر می‌باشد. روشی هوشمند بر اساس تصاویر اشعه X قفسه سینه برای تشخیص کووید-۱۹ ارائه شده که از الگوریتم درخت تصمیم[5] مبتنی بر یادگیری عمیق استفاده می‌کند [۱۳]. این روش شامل سه درخت تصمیم باینری است که هر کدام با استفاده از مدل‌های یادگیری عمیق و شبکه‌های عصبی کانولوشن آموزش دیده‌اند. روش پیشنهادی در این تحقیق از سرعت تشخیص بالایی برخوردار است اما هزینه محاسباتی بالا و تفسیرپذیری پایینی دارد. محققان روشی با استفاده از انتخاب ویژگی و الگوریتم کی-نزدیکترین همسایه[6] برای شناسایی بیماران مبتلا به کووید-۱۹ پیشنهاد کرده‌اند [۱۴]. این روش از انتخاب ویژگی ترکیبی و نسخه‌ای پیشرفته از الگوریتم کی-نزدیکترین همسایه استفاده می‌کند و بر اساس تصاویر توموگرافی قفسه سینه کار می‌کند که دقت قابل توجهی را ارائه می‌دهد. این روش با توجه به این که بر اساس KNN می‌باشد نیاز به منابع محاسباتی قوی و پرهزینه دارد. روش دیگری برای تشخیص کووید-۱۹ بر پایه الگوریتم ماشین بردار پشتیبان معرفی شده است که بر تصاویر اشعه X قفسه سینه متکی است [۱۵]. روش پیشنهاد شده در این پژوهش از گروه‌بندی مجدد تصاویر برای انتخاب ویژگی و از الگوریتم ماشین بردار پشتیبان برای تشخیص بیماری استفاده می‌کند. دقت روش پیشنهادی در آزمایش‌های علمی ۹۳ درصد گزارش شده است. دقت ارائه شده توسط این روش بالا می‌باشد و مدل از پایداری مناسبی برخوردار است اما مدل پیچیده می‌باشد و نیاز به تنظیمات دقیق دارد.

---

[2] Multi Start
[3] Random forest
[4] SVM
[5] Decision tree
[6] KNN

تشخیص بیماری کووید-۱۹ با استفاده از آزمایش CBC و روش‌های یادگیری ماشین در تحقیقی مورد بررسی قرار گرفته است [۱۶]. در این تحقیق، از چندین طبقه‌بند مختلف برای تشخیص استفاده شده که شامل کی-نزدیکترین همسایه، نایو بیز تابع پایه شعاعی، کی-استار، جنگل تصادفی، درخت تصمیم،PART، OneR، ماشین بردار پشتیبان و پرسپترون چند لایه می‌باشند در این تحقیق از معیارهای ارزیابی مختلف برای ارزیابی عملکرد مدل‌ها استفاده شده است که به شفافیت و قابل اعتماد بودن نتایج کمک می‌کند روش جدیدی در این تحقیق ارائه نشده است. یک روش جدید برای شناسایی بیماران کووید-۱۹ از طریق توالی ژنوم انسان با استفاده از طبقه‌بندی کی-نزدیکترین همسایه پیشنهاد گردیده است [۱۷]. روش پیشنهادی را میتوان به عنوان یک روش ناپارامتریک ساده و موثر در نظر گرفت که بیش از ۹۸ درصد دقت در ارزیابی‌های عملی داشته است. از مزیت این روش میتوان به استفاده از ویژگی‌های مبتنی بر ژنوم اشاره کرد که به دلیل توانایی در شناسایی نواحی خاصی از ژنوم که ممکن است با ویروس‌های خاصی مرتبط باشند، میتواند دقت تشخیص را کاهش دهد. اما در مقابل نیاز به داده‌های ژنومی بزرگ از معایت این روش می‌باشد، زیرا جمع‌آوری چنین داده‌هایی میتواند چالش برانگیز باشد. یک استراتژی جدید و کارآمد برای شناسایی بیمارهای مبتلا به کووید-۱۹ بر پایه الگوریتم نایو بیز نیز در تحقیقی معرفی شده است [۱۸]. در این تحقیق، انتخاب ویژگی با استفاده از یک نسخه بهبود یافته از الگوریتم ازدحام ذرات انجام شده و برای تشخیص نیز از نسخه بهبود یافته الگوریتم نایو بیز استفاده شده است. از مزایای این روش میتوان به تشخیص سریع و همچنین انتخاب ویژگی‌های مناسب اشاره کرد. اما نیاز به داده‌های با کیفیت بالا ضعف این روش می‌باشد. یک روش برای تست سریع کووید-۱۹ ارائه شده که ترکیبی از الگوریتم طبقه بندی جنگل تصادفی و آزمایش خون است [۱۹]. الگوریتم جنگل تصادفی مورد استفاده در این روش دارای ۹۰ درخت تصمیم است. روش پیشنهادی در پژوهش مربوطه دقت ۹۲ درصدی در تشخیص بیماران را در ارزیابی‌های عملی ارائه کرده است. روش پیشنهاد شده در این پژوهش مبتنی بر وب می‌باشد که باعث دسترسی آسان آن می‌شود. اما این عامل باعث می‌شود که روش محدودیت‌های زیرساختی داشته باشد و برای اجرای آن نیاز به زیرساخت‌های اینترنتی و شبکه‌ای باشد. محققان یک روش برای شناسایی بیمارهای کووید-۱۹ بر اساس استراتژی طبقه‌بندی نایو بیز با ویژگی‌های مرتبط ارائه داده‌اند [۲۰]. این روش شامل چهار فاز است. فازهای روش پیشنهادی در این تحقیق شامل فاز انتخاب ویژگی، فاز خوشه‌بندی ویژگی‌ها، فاز وزن‌دهی ویژگی اصلی و فاز بیز ساده مرتبط با ویژگی است. مزایای این روش دقت بالا، سرعت پردازش بالا و مقاومت در برابر داده‌های ناقص است و معایب اصلی آن شامل پیچیدگی در انتخاب ویژگی‌ها و نیاز به داده‌های باکیفیت بالا می‌باشد.

یک روش بر پایه یادگیری نمادین[7] و ویژگی‌های صوتی برای تشخیص کووید-۱۹ نیز معرفی شده است [۲۱]. یادگیری نمادین، که یک رویکرد منطقی در یادگیری ماشین است، به دنبال استخراج و بیان اطلاعات منطقی از داده‌ها به شکلی قابل فهم است. ویژگی‌های صوتی مورد استفاده در این تحقیق شامل داده‌های مربوط به سرفه و تنفس بیماران است. روش پیشنهادی از الگوریتم‌های درخت تصمیم و جنگل تصادفی برای طبقه‌بندی استفاده می‌کند. در این تحقیق نتایج تجربی نشان داده‌اند که روش پیشنهادی با استخراج دانش صریح، دقت بسیار بالایی در تشخیص ارائه می‌کند. مزیت دیگر این روش این است که روشی کاملاً غیرتهاجمی می‌باشد و از صدای سرفه و تنفس به عنوان ورودی استفاده می‌کند. اما ضعف اصلی آن نیاز به داده‌های آموزشی گسترده از نوع صوتی می‌باشد که جمع‌آوری آنها چالش برانگیز می‌باشد. مدلی جهت شناسایی بیماران کووید-۱۹ بر اساس تصاویر سی تی اسکن و الگوریتم‌های یادگیری ماشین نیز ارائه گردیده است[۲۲]. چهار الگوریتم یادگیری ماشین شامل درخت تصمیم، کی-نزدیک‌ترین همسایه، ماشین بردار پشتیبان و نایو بیز برای طبقه‌بندی اسکن‌ها در مدل پیشنهادی به کار رفته‌اند. مزیت این روش این است که علاوه بر تشخیص بیماری، به ارزیابی شدت بیماری نیز در سه درجه خفیف، متوسط و شدید می‌پردازد. از معایب اصلی این روش نیز میتوان به استفاده

[7] Symbolic learning

از مدل‌های زیاد اشاره کرد. در جدول ۱، فهرستی از روش‌های مرور شده جهت تشخیص کووید-۱۹ بر اساس الگوریتم‌های یادگیری ماشین، ارائه شده است. علاوه بر پژوهش‌های مرور شده، در برخی از پژوهش‌ها از دیدگاه‌های دیگری به بیماری کووید-۱۹ پرداخته شده که از جمله میتوان به [۲۳]، [۲۴] و [۲۵] اشاره کرد.

جدول ۱: راه‌کارهای مرورشده تشخیص کووید-۱۹ بر اساس یادگیری ماشین

پژوهش	سال	الگوریتم‌های استفاده شده
(Iwendi &el al)	۲۰۲۰	جنگل تصادفی
(Singh &el al)	۲۰۲۰	ماشین بردار پشتیبان
(Yoo &el al)	۲۰۲۰	درخت تصمیم
(Shaban &el al)	۲۰۲۰	کی-نزدیک‌ترین همسایه
(Zhou &el al)	۲۰۲۱	ماشین بردار پشتیبان
(Akhtar &el al)	۲۰۲۱	چندین طبقه‌بند شامل کی-نزدیک‌ترین همسایه، درخت تصمیم، جنگل تصادفی و ...
(Arslan &el al)	۲۰۲۱	کی-نزدیک‌ترین همسایه
(Shaban &el al)	۲۰۲۱	نایو بیز
(Barbosa &el al)	۲۰۲۲	جنگل تصادفی
(Mansour &el al)	۲۰۲۲	نایو بیز
(Manzella &el al)	۲۰۲۳	یادگیری نمادین، درخت تصمیم، جنگل تصادفی
(Albataineh &el al)	۲۰۲۴	درخت تصمیم، کی-نزدیک‌ترین همسایه، ماشین بردار پشتیبان، نایو بیز

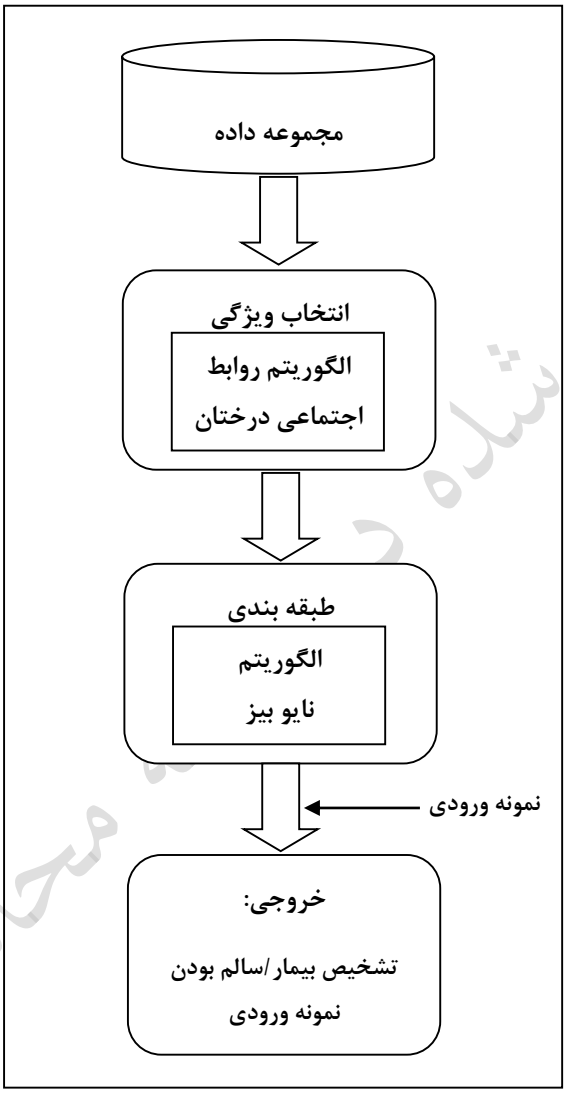
با توجه به مروری که بر روش‌های موجود انجام شده است، به نظر می‌رسد که استفاده از انتخاب ویژگی در کنار الگوریتم نایو بیز می‌تواند یک رویکرد نوآورانه برای تشخیص کووید-۱۹ باشد. انتخاب ویژگی می‌تواند به کاهش پیچیدگی مدل‌ها و افزایش سرعت و دقت تشخیص کمک کند؛ در حالی که اکثر تحقیق‌ها از آن استفاده نمی‌کنند. انتظار می‌رود این ترکیب منجر به ایجاد یک مدل قدرتمند و کارآمد شود که قادر به تشخیص سریع و دقیق کووید-۱۹ باشد.

۳. روش پیشنهادی

روشی که در این مقاله برای شناسایی بیماران کووید-۱۹ پیشنهاد شده، دارای دو مرحله اصلی است. در مرحله اول، از میان تمام ویژگی‌های موجود در مجموعه داده، بهترین ویژگی‌ها انتخاب می‌شوند. این انتخاب به صورت یک فرآیند بهینه‌سازی انجام می‌شود که توسط الگوریتم فراابتکاری TSR صورت می‌پذیرد. الگوریتم TSR یکی از الگوریتم‌های نوین و کارآمد در حوزه الگوریتم‌های فراابتکاری می‌باشد. در روش پیشنهادی الگوریتم TSR به صورت چند شروعی مورد استفاده قرار می‌گیرد.

مرحله دوم، فرآیند تشخیص بیماری است که با استفاده از روش‌های یادگیری نظارت شده یا طبقه‌بندی انجام می‌شود. در این مرحله، الگوریتم نایو بیز به کار گرفته می‌شود که یک روش طبقه‌بندی بر پایه آمار است. این الگوریتم احتمال تعلق هر نمونه داده‌ای را به کلاس‌های مختلف محاسبه می‌کند و بر اساس بیشترین احتمال، پیش‌بینی را انجام می‌دهد. این روش بر اساس قضیه بیز کار می‌کند و فرض می‌کند که هر ویژگی به طور مستقل از دیگر ویژگی‌ها عمل می‌کند. به عبارتی دیگر در این الگوریتم، ویژگی‌های داده‌ها،

مستقل از یکدیگر در نظر گرفته می‌شوند. این امر به کاهش محاسبات و افزایش سرعت تشخیص کمک می‌کند. قابل ذکر است که الگوریتم نایو بیز نیازی به مرحله جداگانه‌ای برای آموزش ندارد. مراحل روش پیشنهادی برای تشخیص بیماری کووید-۱۹ در شکل ۱ نمایش داده شده است.



شکل ۱: مراحل کلی روش پیشنهادی برای تشخیص کووید-۱۹

همان طور که قابل مشاهده است، داده‌های مربوط به بیماران کووید-۱۹ در ابتدای کار موجود می‌باشد. این داده‌ها برچسب‌دار هستند و برچسب آنها نشان‌دهنده وضعیت افراد از نظر سالم و بیمار بودن است. مرحله اول روش پیشنهادی شامل فرآیند انتخاب ویژگی است که در آن ویژگی‌های مؤثر برای تشخیص انتخاب می‌شوند. مرحله دوم بر اساس قانون بیز، تشخیص بیمار یا سالم بودن افراد را انجام می‌دهد. با تلفیق این دو مرحله، انتظار می‌رود که تشخیص با دقت بالا صورت پذیرد. در ادامه به مراحل روش پیشنهاد شده اشاره خواهد شد.

۱،۳. انتخاب ویژگی با الگوریتم روابط اجتماعی درختان

در روش پیشنهادی، اولین گام انتخاب ویژگی‌ها است که با استفاده از الگوریتم فراابتکاری TSR انجام می‌پذیرد. در این گام از روش پیشنهاد شده، ویژگی‌هایی که برای تشخیص بیماری‌ها مؤثر هستند، انتخاب می‌شوند. این کار دو فایده دارد: نخست اینکه با انتخاب دقیق ویژگی‌ها، صحت تشخیص بیماری‌ها بهبود می‌یابد و دوم اینکه با کاهش تعداد ویژگی‌ها، حجم پردازش داده‌ها کمتر می‌شود. بنابراین انتخاب ویژگی منجر به تشخیص بیماری با سرعت بالاتر و زمان کوتاه‌تر می‌گردد. در نتیجه اهمیت این مرحله بسیار بالا می‌باشد.

الگوریتم TSR با تولید تصادفی مجموعه‌ای از راه‌حل‌ها کار را آغاز می‌کند. در این الگوریتم هر یک از راه‌حل‌ها به عنوان یک درخت در نظر گرفته می‌شود. در ادامه این راه‌حل‌ها به چندین زیرجنگل تقسیم می‌شوند و در هر زیرجنگل، تعدادی از درخت‌ها به صورت تصادفی به عنوان نهال‌ها انتخاب می‌شوند. سپس الگوریتم وارد یک حلقه تکرار می‌شود که در آن به دنبال راه‌حل‌های بهینه می‌گردد. این حلقه تکرار شامل دو فرآیند کلی زیرجنگل‌ها[8] و سراسری[9] است که بر روی تمام زیرجنگل‌ها اجرا می‌شوند.

در فرآیند زیرجنگل‌ها، با استفاده از سه عملگر، راه‌حل‌های جدیدی در هر زیرجنگل ایجاد می‌شوند. این عملگرها شامل تکثیر[10]، تکثیر نهال[11] و لایه‌بندی[12] می‌باشند. عملگر تکثیر مشابه عملگر تقاطع در الگوریتم ژنتیک عمل می‌کند و عملگر تکثیر نهال نیز به همین شکل است، با این تفاوت که بر روی درخت‌های نهال تمرکز دارد. درختان نهال میزان رشد بالایی دارند و میتوانند باعث ایجاد راه‌حل‌های مناسب در تکرارهای بعدی شوند. عملگر لایه‌بندی مانند عملگر جهش در الگوریتم ژنتیک عمل می‌کند. در پایان این فرآیند، ضعیف‌ترین راه‌حل‌های موجود در زیرجنگل‌ها حذف شده و نهال‌ها با درخت‌هایی که رشد بالاتری دارند، به‌روزرسانی می‌شوند.

در فرآیند سراسری، هدف ایجاد تعادل در میانگین برازندگی بین زیرجنگل‌ها است. برای این منظور، تعدادی از راه‌حل‌های مناسب از زیرجنگل‌های بهتر به زیرجنگل‌های ضعیف‌تر کپی می‌گردند و به همان اندازه، راه‌حل‌های ضعیف از زیرجنگل‌های ضعیف‌تر حذف می‌شوند.

در نهایت، پس از اینکه شرط توقف حلقه تکرار برآورده شود، کار الگوریتم به پایان می‌رسد و بهترین راه‌حل موجود در تمامی زیرجنگل‌ها (کل جنگل) به عنوان راه‌حل نهایی ارائه می‌شود. در روش پیشنهادی، انتخاب ویژگی‌ها بر اساس این الگوریتم صورت می‌گیرد. علت انتخاب الگوریتم TSR برای انتخاب ویژگی در روش پیشنهادی این است که این الگوریتم یک الگوریتم فراابتکاری گسسته می‌باشد و برای حل مسئله انتخاب ویژگی (که یک مسئله گسسته است) مناسب می‌باشد. از سایر الگوریتم‌های فراابتکاری مانند الگوریتم ژنتیک، تبرید شبیه‌سازی شده، کلونی مورچگان و غیره برای انتخاب ویژگی استفاده می‌شود. اما برتری الگوریتم TSR نسبت به سایر الگوریتم‌های فراابتکاری این است که این الگوریتم به راه‌حل‌های در حال رشد توجه خاصی دارد و این امر باعث همگرایی بهتر آن نسبت به دیگر روش‌ها می‌شود. در ادامه، مراحل انتخاب ویژگی در روش پیشنهادی بیان خواهد شد.

۱،۱،۳. نحوه نمایش راه‌حل‌ها

در روش پیشنهاد شده، هر راه‌حل نشان‌دهنده‌ی ویژگی‌هایی است که انتخاب شده‌اند. از روش نمایش دودویی برای نمایش راه‌حل‌ها در روش پیشنهادی استفاده می‌شود. در این روش هر راه‌حل به صورت یک رشته از بیت‌ها تعریف می‌شود. طول این رشته بیت‌ها برابر با تعداد کل ویژگی‌های موجود است. اگر ویژگی‌ای در راه‌حل موجود باشد، بیت متناظر با آن مقدار یک می‌گیرد؛ در حالی که

[8] Sub-jungle process
[9] Global king process
[10] Proliferation
[11] SeedlingsProliferation
[12] Layering

اگر یک ویژگی در راه‌حل انتخاب نشده باشد، بیت مربوطه صفر خواهد شد. به عنوان مثال، اگر مجموعه داده‌ای شامل ۸ ویژگی با نام‌های a1 الی a8 باشد، یک راه‌حل فرضی می‌تواند مطابق با شکل ۲ باشد.

0	1	0	1	0	0	0	1
a1	a2	a3	a4	a5	a6	a7	a8

شکل ۲: یک راه‌حل نمونه برای نمایش راه‌حل‌ها

همانطور که قابل مشاهده است، طول راه‌حل، هشت بیت است. ویژگی‌های دوم، چهارم، و هشتم در این راه‌حل انتخاب شده‌اند، چرا که بیت‌های مربوط به آن‌ها مقدار یک دارند. بیت‌های دیگر که مقدار صفر دارند، نشان‌دهنده‌ی ویژگی‌هایی هستند که انتخاب نشده‌اند.

۲،۱،۳. ایجاد جمعیت اولیه و زیرجنگل‌ها

در روش پیشنهادی، جمعیت اولیه به شکل تصادفی ساخته می‌شود. درختان یا راه‌حل‌ها بر اساس ساختار توضیح داده شده در قسمت پیشین، به صورت کاملاً تصادفی تولید می‌شوند. تعداد راه‌حل‌های ایجاد شده یا همان اندازه جمعیت توسط پارامتر NP تعیین می‌گردد که این پارامتر از ورودی و توسط کاربر تعیین می‌شود.

بعد از ایجاد راه‌حل‌های اولیه، باید زیرجنگل‌ها ایجاد شوند. در این مرحله، تمامی اعضای جمعیت به چندین گروه با تعداد اعضای برابر تقسیم می‌شوند. تعداد زیرجنگل‌ها توسط پارامتر NK معین می‌شود که از ورودی دریافت می‌گردد. پس هر زیرجنگل شامل NP/NK درخت خواهد بود. در مرحله ساخت زیرجنگل‌ها، هر درخت تنها در یک زیرجنگل قرار می‌گیرد، به این معنا که زیرجنگل‌ها در ابتدا هیچ عضو مشترکی ندارند.

پس از تشکیل زیرجنگل‌ها، تعدادی از راه‌حل‌های موجود در آن به عنوان نهال‌های اولیه در نظر گرفته می‌شوند. در ابتدای کار این راه‌حل‌ها به صورت تصادفی انتخاب می‌شوند. تعداد راه‌حل‌های نهال هر زیرجنگل با پارامتر PS مشخص می‌شود که مقدار آن از ورودی دریافت می‌گردد. پس از مشخص شدن نهال‌های هر زیرجنگل، الگوریتم وارد حلقه‌ی تکرار اصلی می‌شود. همان‌طور که اشاره شد این حلقه از دو فرآیند کلی تشکیل شده است.

پس از ساخت زیرجنگل‌ها، برخی از راه‌حل‌های موجود به عنوان نهال‌های اولیه انتخاب می‌شوند. در ابتدا، این نهال‌ها به طور تصادفی انتخاب می‌گردند. تعداد نهال‌های هر زیرجنگل توسط پارامتر PS تعیین می‌شود که مقدار آن نیز از ورودی دریافت می‌شود. بعد از تعیین نهال‌های هر زیرجنگل، الگوریتم وارد حلقه تکرار اصلی می‌شود. همانطور که قبلاً ذکر شد، حلقه تکرار شامل دو فرآیند کلی می‌باشد.

۳،۱،۳. تابع برازندگی

در روش پیشنهادی، بعد از تولید هر راه‌حل، باید میزان مناسب بودن آن تعیین شود. در الگوریتم‌های فراابتکاری، از تابعی به نام تابع برازندگی برای ارزیابی راه‌حل‌ها استفاده می‌شود. این تابع، یک راه‌حل را دریافت کرده و یک عدد را به عنوان خروجی می‌دهد که میزان مناسبت آن راه‌حل را نشان می‌دهد. در روش پیشنهادی، با استفاده از ویژگی‌های مشخص شده در راه‌حل، یک طبقه‌بندی با

استفاده از الگوریتم نایو بیز انجام می‌شود و میزان دقت حاصل محاسبه می‌گردد. سپس، برازندگی راه‌حل بر اساس رابطه‌ی ۱ محاسبه می‌گردد.

$$fitness = \beta \times (1 - acc) + (1 - \beta) \times \frac{SF}{AF} \quad (۱)$$

در این رابطه، acc نشان‌دهنده میزان دقت حاصل از طبقه‌بندی است. پارامتر SF تعداد ویژگی‌های انتخاب شده در راه‌حل را نشان می‌دهد، در حالی که پارامتر AF به تعداد کل ویژگی‌های موجود اشاره دارد. پارامتر β، که مقداری بین صفر و یک دارد، یک ضریب تعدیل بین دقت و میزان تأثیر کاهش تعداد ویژگی‌ها می‌باشد که مقدار آن از ورودی دریافت می‌گردد. بر اساس تابع برازندگی تعریف شده، راه‌حل‌هایی با برازندگی کمتر، مناسب‌تر هستند. لازم به ذکر است که در محاسبه برازندگی، تمام داده‌های موجود، در طبقه‌بندی به کار گرفته نمی‌شوند و تنها ده درصد از آن‌ها برای جلوگیری از زمان‌بر بودن فرآیند طبقه‌بندی مورد استفاده قرار می‌گیرند. این ده درصد به صورت تصادفی در ابتدای کار انتخاب می‌شوند.

۴،۱،۳. فرآیند زیرجنگل‌ها

در فرآیند زیرجنگل‌ها، در هر زیرجنگل جهت یافتن راه‌حل‌های بهینه جستجو می‌شود و در هر تکرار، تعدادی راه‌حل جدید تولید می‌گردد. این فرآیند شامل سه عملگر تکثیر، تکثیر نهال، و لایه‌بندی می‌باشد. پس از اجرای این سه عملگر، مراحل حذف آفت‌زده‌ها و به‌روزرسانی نهال‌ها صورت می‌پذیرد. در ادامه، جزئیات عملگرها و مراحل فرآیند زیرجنگل‌ها بررسی خواهد شد.

● عملگر تکثیر

در عملگر تکثیر، از دو راه‌حل والد، دو راه‌حل فرزند جدید تولید می‌گردد. در ابتدا، دو راه‌حل با استفاده از روش چرخ رولت انتخاب می‌گردند. در این روش راه‌حل‌های بهتر شانس بیشتری برای انتخاب شدن دارند. سپس، با استفاده از یکی از روش‌های تقاطع موجود در الگوریتم‌های ژنتیک، دو راه‌حل فرزند جدید از این راه‌حل والد تولید می‌شوند. تعداد راه‌حل‌های فرزند تولید شده توسط عملگر تکثیر در هر تکرار الگوریتم و برای هر زیرجنگل توسط پارامتر PP تعیین می‌شود که مقدار این پارامتر از ورودی دریافت می‌گردد.
در روش پیشنهادی، برای تولید راه‌حل‌های فرزند جدید با استفاده از عملگر تکثیر، از تقاطع یک‌نقطه‌ای استفاده می‌گردد. در این روش، ابتدا یک نقطه تصادفی روی راه‌حل‌های والد انتخاب می‌شود. سپس، راه‌حل‌های فرزند بخشی از خصوصیات را از یک والد و بخش دیگری را از والد دیگر به ارث می‌برند، که در نتیجه دو راه‌حل فرزند جدید تولید می‌شود. در هر تکرار از الگوریتم، راه‌حل‌های فرزند جدید تولید شده به همان زیرجنگلی که از آن تولید شده‌اند، اضافه می‌گردند. نمونه‌ای از تولید راه‌حل‌های جدید با استفاده از عملگر تکثیر در روش پیشنهادی در شکل ۳ آورده شده است.

راه‌حل ۱

0	1	0	1	0	0	0	1
a1	a2	a3	a4	a5	a6	a7	a8

راه‌حل ۲

1	0	0	0	1	1	0	1

a1	a2	a3	a4	a5	a6	a7	a8

راه‌حل جدید ۱

0	1	0	0	1	1	0	1
a1	a2	a3	a4	a5	a6	a7	a8

راه‌حل جدید ۲

1	0	0	1	0	0	0	1
a1	a2	a3	a4	a5	a6	a7	a8

شکل ۳: نمونه‌ای از عملگر تکثیر

در این مثال، می‌توان دید که نقطه سوم راه‌حل‌ها به طور تصادفی انتخاب شده و بر این اساس، راه‌حل‌های والد به دو دسته تقسیم می‌شوند. پس از آن، راه‌حل‌های جدید بر مبنای این تقسیم‌بندی ساخته می‌شوند.

بعد از ایجاد راه‌حل‌های جدید، میزان برازندگی آن‌ها با استفاده از تابع پیشنهاد شده ارزیابی می‌گردد. در ادامه، باید نرخ رشد این راه‌حل‌ها محاسبه شود. همانطور که قبلاً ذکر شد، نرخ رشد نشان‌دهنده میزان رشد برازندگی راه‌حل است. نرخ رشد برای راه‌حل‌هایی که توسط عملگر تکثیر ایجاد شده‌اند، بر اساس رابطه ۲ محاسبه می‌گردد.

$$GR = \frac{fitness_{new}}{fitness_{P1} + fitness_{P2}} \quad (۲)$$

با توجه به این رابطه، می‌توان دریافت که نرخ رشد برابر با برازندگی راه‌حل فرزند تقسیم بر جمع برازندگی‌های والدهای آن است.

● عملگر تکثیر نهال

در عملگر تکثیر نهال، دو راه‌حل فرزند از دو راه‌حل والد، مشابه عملگر تکثیر، ایجاد می‌گردد. با این تفاوت که در عملگر تکثیر نهال، راه‌حل‌های والد بر اساس روش چرخ رولت انتخاب نمی‌گردند. این عملگر به درختان نهال توجه خاصی دارد. در هر زیرجنگل، یکی از والدها از درختان نهال (به صورت تصادفی) و والد دیگر از درختان غیرنهال (به صورت تصادفی) انتخاب می‌شود. تعداد راه‌حل‌های ایجاد شده توسط عملگر تکثیر نهال در هر تکرار از الگوریتم و برای هر زیرجنگل با پارامتر PSP تعیین می‌گردد که مقدار آن از ورودی دریافت می‌شود.

در روش پیشنهاد شده، جهت تولید راه‌حل‌های جدید با استفاده از عملگر تکثیر نهال، از تقاطع دونقطه‌ای استفاده می‌گردد. در این حالت ابتدا دو نقطه به صورت تصادفی در راه‌حل‌های والد در نظر گرفته می‌شود. سپس راه‌حل‌های فرزند یک بخش خود را از یک والد و دو بخش دیگر را از والد دیگر به ارث می‌برند. به این ترتیب، دو راه‌حل فرزند جدید تولید می‌گردد. در هر تکرار از الگوریتم، راه‌حل‌های جدید تولید شده توسط عملگر تکثیر نهال برای هر زیرجنگل، به همان زیرجنگل اضافه می‌گردد. نمونه‌ای از تولید راه‌حل جدید با استفاده از عملگر تکثیر نهال در روش پیشنهادی در شکل ۴ قابل مشاهده می‌باشد.

راه‌حل ۱ (غیرنهال)

0	1	0	1	0	0	0	1
a1	a2	a3	a4	a5	a6	a7	a8

راه‌حل ۲ (نهال)

1	0	0	0	1	1	0	1
a1	a2	a3	a4	a5	a6	a7	a8

راه‌حل جدید ۱

0	1	0	0	1	0	0	1
a1	a2	a3	a4	a5	a6	a7	a8

راه‌حل جدید ۲

1	0	0	1	0	1	0	1
a1	a2	a3	a4	a5	a6	a7	a8

شکل ۴: نمونه‌ای از عملگر تکثیر نهال

همان طور که قابل مشاهده است، در این مثال نقطه‌های سوم و پنجم در راه‌حل‌ها به صورت تصادفی انتخاب شده‌اند و بر اساس آن‌ها، راه‌حل‌های والد به سه قسمت تقسیم شده‌اند. سپس راه‌حل‌های جدید با استفاده از این تقسیم‌بندی تشکیل شده‌اند.
پس از تولید راه‌حل‌های جدید، مقدار برازندگی آن‌ها طبق تابع پیشنهادی محاسبه می‌گردد. در ادامه نرخ رشد آن‌ها نیز محاسبه می‌شود. در عملگر تکثیر نهال، مشابه عملگر تکثیر، نرخ رشد برای راه‌حل‌های ایجاد شده بر اساس رابطه‌ی ۲ محاسبه می‌گردد. این محاسبات به منظور به‌روزرسانی نهال‌ها در پایان هر تکرار از الگوریتم انجام می‌شود تا در تکرارهای بعدی الگوریتم، نهال‌های مناسبی جهت انتخاب در عملگر تکثیر نهال موجود باشند.

- **عملگر لایه بندی**

در عملگر لایه بندی، از یک راه‌حل موجود، راه‌حل جدیدی تولید می‌شود. در این عملگر ابتدا یک راه‌حل به صورت تصادفی از بین راه‌حل‌های موجود در زیرجنگل انتخاب می‌گردد. در ادامه با استفاده از یکی از روش‌های عملگر جهش در الگوریتم‌های ژنتیک، راه‌حل جدیدی ایجاد می‌شود. تعداد راه‌حل‌های تولید شده توسط عملگر لایه بندی در هر تکرار و برای هر زیرجنگل توسط پارامتر PL تعیین می‌شود که مقدار آن از ورودی دریافت می‌گردد.
در روش پیشنهاد شده، برای ایجاد راه‌حل جدید در عملگر لایه بندی، از روش معکوس استفاده می‌شود. در این روش، ابتدا دو ویژگی به صورت تصادفی در راه‌حل انتخاب می‌گردد. سپس، مقدارهای بین این دو ویژگی در راه‌حل، معکوس می‌گردند. در هر تکرار از

الگوریتم، راه‌حل‌های جدید تولید شده در هر زیرجنگل به همان زیرجنگل افزوده می‌شوند. نمونه‌ای از تولید راه‌حل جدید با استفاده از عملگر لایه‌بندی در شکل ۵ قابل مشاهده می‌باشد.

راه‌حل

0	1	0	1	0	0	0	1
a1	a2	a3	a4	a5	a6	a7	a8

راه‌حل جدید

1	0	0	0	0	1	0	1
a1	a2	a3	a4	a5	a6	a7	a8

شکل ۵: نمونه‌ای از عملگر لایه‌بندی

در این مثال، ویژگی‌های سوم و هفتم به طور تصادفی انتخاب شده‌اند. پس از آن، مقادیر بین این دو ویژگی در راه‌حل معکوس شده‌اند و نتیجه به عنوان راه‌حل جدید در نظر گرفته می‌شود.

پس از ایجاد راه‌حل‌های جدید، مقدار برازندگی آن‌ها با استفاده از تابع پیشنهاد شده محاسبه می‌گردد. در ادامه، لازم است که نرخ رشد این راه‌حل‌ها نیز محاسبه شود. نرخ رشد برای راه‌حل‌های تولید شده توسط این عملگر بر اساس رابطه‌ی ۳ محاسبه می‌گردد.

$$GR = \frac{fitness_{new}}{fitness_P} \quad (۳)$$

طبق این رابطه، نرخ رشد برابر است با برازندگی راه‌حل فرزند تقسیم بر برازندگی راه‌حل والد؛ چرا که در این عملگر تنها از یک والد برای تولید راه‌حل جدید استفاده می‌شود.

● حذف آفت زده‌ها

پس از به کارگیری سه عملگر تکثیر، تکثیر نهال و لایه‌بندی، تعدادی راه‌حل جدید ایجاد می‌شود که منجر به افزایش تعداد راه‌حل‌ها در زیرجنگل‌ها نسبت به حالت اولیه می‌گردد. در مرحله حذف آفت‌زده‌ها، ضعیف‌ترین راه‌حل‌های موجود در زیرجنگل‌ها حذف شده تا تعداد راه‌حل‌های هر زیرجنگل به حالت ابتدایی بازگردد.

در مرحله حذف آفت زده‌ها، ابتدا راه‌حل‌های موجود در هر زیرجنگل صورت صعودی بر اساس برازندگی مرتب می‌گردند. در ادامه، به اندازه تعداد راه‌حل‌های جدید اضافه شده، راه‌حل‌های ضعیف از زیرجنگل‌ها حذف می‌گردند. این اقدام باعث می‌شود که راه‌حل‌های مناسب‌تر در زیرجنگل‌ها باقی بمانند و راه‌حل‌های ضعیف‌تر در تکرارهای بعد به کار گرفته نشوند.

● بروزرسانی نهال‌ها

در هر تکرار از الگوریتم و برای هر زیرجنگل، پس از ایجاد راه‌حل‌های جدید و کنار گذاشتن آفت‌زده‌ها، لازم است که درختان نهال به‌روزرسانی گردند. درختان نهال، راه‌حل‌هایی با نرخ رشد بالا هستند. نهال‌های اولیه به صورت تصادفی انتخاب شده‌اند، اما در مراحل بعدی، نهال‌ها راه‌حل‌هایی می‌باشند که در حال رشد هستند و میزان رشد آنها بالا می‌باشد.

جهت به‌روزرسانی درخت‌های نهال، راه‌حل‌ها در هر زیرجنگل به صورت نزولی بر اساس میزان نرخ رشد مرتب می‌گردند. سپس، بر اساس پارامتر PS، بهترین راه‌حل‌ها بر اساس نرخ رشد انتخاب شده و به عنوان نهال‌های آن زیرجنگل در نظر گرفته می‌شوند. این نهال‌ها در تکرار بعدی الگوریتم و در عملگر تکثیر نهال برای انتخاب یکی از والدها استفاده خواهند شد.

۵،۱،۳. فرآیند سراسری

بعد از اجرای فرآیند زیرجنگل‌ها در هر تکرار از الگوریتم، فرآیند سراسری اجرا می‌گردد. هدف این فرآیند برقراری توازن بین تمامی زیرجنگل‌ها می‌باشد. به عبارت دیگر هدف کمک به زیرجنگل‌های ضعیف‌تر است. در این فرآیند تعدادی از بهترین راه‌حل‌ها از زیرجنگل‌های قوی‌تر به زیرجنگل‌های ضعیف‌تر کپی می‌شوند.

در این مرحله، ابتدا میانگین برازندگی راه‌حل‌ها برای هر زیرجنگل محاسبه می‌گردد. در ادامه، زیرجنگل‌ها بر اساس میانگین برازندگی‌ها به صورت صعودی در یک لیست مرتب می‌شوند. پس از آن، از زیرجنگل ابتدایی در لیست تعدادی از بهترین راه‌حل‌ها انتخاب و در زیرجنگل انتهایی کپی می‌شوند. این تعداد با استفاده از پارامتر PE که مقدار آن از ورودی دریافت می‌شود، مشخص می‌گردد. پس از کپی راه‌حل‌های مناسب در زیرجنگل‌های ضعیف، به همین تعداد راه‌حل ضعیف از آن‌ها حذف می‌گردد تا تعداد راه‌حل‌های زیرجنگل تغییر نکند. در ادامه این فرآیند، انتقال بین درخت دوم و ماقبل آخر در لیست انجام می‌شود و همین روند با بقیه زیرجنگل‌های موجود در لیست ادامه پیدا می‌کند. اعمال فرآیند سراسری باعث می‌شود که همه زیرجنگل‌ها با میانگین برازندگی بالا در تکرارهای بعدی الگوریتم به جستجوی راه‌حل‌های بهینه در فضای جستجوی مسئله بپردازند.

۶،۱،۳. شرط توقف

بعد از اجرای یک تکرار از الگوریتم و انجام فرآیندهای اشاره شده بر روی راه‌حل‌های موجود در زیرجنگل‌ها، همین مراحل دوباره تکرار می‌شود. این کار به تولید راه‌حل‌های بهینه‌تر و حرکت الگوریتم به سمت بهینه‌سازی کمک می‌کند. تکرار الگوریتم تا زمانی ادامه می‌یابد که شرط توقف الگوریتم برآورده شود. در روش پیشنهادی، تعداد تکرار مشخص برای حلقه تکرار به عنوان شرط توقف الگوریتم مد نظر قرار گرفته است. این تعداد با استفاده از پارامتر MI تعیین می‌گردد و مقدار آن از ورودی دریافت می‌شود.

۷،۱،۳. اجرای TSR به صورت چند شروعی

در روش پیشنهادی جهت انتخاب ویژگی؛ الگوریتم TSR به صورت چند شروعی مورد استفاده قرار می‌گیرد. برای این منظور مراحل الگوریتم TSR تا چند مرحله ادامه می‌یابد و بعد از آن بهترین راه‌حل ذخیره خواهد شد. سپس الگوریتم مجدداً از ابتدا شروع به جستجو خواهد کرد. یعنی تمامی راه‌حل‌های موجود در جمعیت کنار گذاشته می‌شود و مجدداً جمعیت جدیدی تولید می‌شود. این کار چندین بار تکرار خواهد شد. به عبارت دیگر هر بار تا یک شعاعی فضای مسئله جستجو می‌شود و بعد جستجو از ابتدا آغاز می‌شود. در پایان بهترین راه حل در شروع‌های مختلف به عنوان راه حل نهایی در نظر گرفته می‌شود. مراحل اجرای الگوریتم TSR به صورت چند شروعی را میتوان به صورت موارد زیر بیان نمود:

۱. ایجاد چندین راه‌حل اولیه: در ابتدای هر دوره جستجو، چندین نقطه شروع اولیه به صورت تصادفی ایجاد می‌شود. این نقاط شروع به عنوان راه‌حل‌های اولیه در نظر گرفته می‌شوند.
۲. اجرای الگوریتم TSR: الگوریتم TSR بر روی هر یک از این نقاط شروع اجرا می‌شود و جستجوی بهینه‌ترین جواب ممکن آغاز می‌شود. این مرحله تا چندین تکرار ادامه می‌یابد و هر بار جواب‌های به دست آمده مورد ارزیابی قرار می‌گیرد.

۳. ذخیره بهترین راه‌حل: پس از چندین تکرار، بهترین راه‌حل‌های به دست آمده از هر دوره جستجو ذخیره می‌شود. این راه‌حل‌ها به عنوان راه‌حل‌های بهینه محلی در نظر گرفته می‌شوند.
۴. شروع مجدد جستجو: در این مرحله، تمامی راه‌حل‌های موجود در جمعیت کنار گذاشته می‌شود و مجدداً جمعیت جدیدی تولید می‌شود. این جمعیت جدید شامل نقاط شروع اولیه جدیدی است که از همان مراحل قبلی تولید می‌شوند. الگوریتم TSR دوباره از ابتدا شروع به جستجو می‌کند و بهینه‌ترین راه‌حل جدید را پیدا می‌کند.
۵. تکرار فرآیند: این فرآیند چندین بار تکرار می‌شود. در هر بار جستجو، فضای مسئله تا یک شعاع مشخصی جستجو می‌شود و پس از آن جستجو از ابتدا آغاز می‌شود. این تکرارهای مکرر به الگوریتم امکان می‌دهد تا بهینه‌ترین جواب را در فضای جستجو پیدا کند و از گیر کردن در جواب‌های محلی جلوگیری کند.
۶. انتخاب بهترین راه حل نهایی: در پایان، بهترین راه حل به دست آمده از تمامی شروع‌های مختلف به عنوان راه حل نهایی در نظر گرفته می‌شود. این راه حل نهایی به عنوان نتیجه بهینه‌ترین راه حل در کل فضای جستجو در نظر گرفته می‌شود.

تعداد شروع‌های الگوریتم با پارامتر NOS مشخص می‌شود که از ورودی دریافت می‌گردد. بنابراین هر شروع از الگوریتم به تعداد $\frac{MI}{NOS}$ تکرار خواهد شد. اجرای الگوریتم به صورت چند شروعی باعث می‌شود که الگوریتم واگرایی مناسبی داشته باشد و بتواند به صورت مناسب‌تری به جستجوی فضای مسئله بپردازد.

تفاوت اصلی میان روش چند شروعی و روش پایه در نحوه شروع و فرآیند جستجو برای یافتن راه حل بهینه است. در روش پایه، الگوریتم بهینه‌سازی تنها از یک راه حل یا جمعیت شروع به کار می‌کند و جستجو را تا یافتن جواب بهینه ادامه می‌دهد. این روش به دلیل محدودیت در نقاط شروع ممکن است نتواند جواب مناسب را پیدا کند. به عبارت دیگر، روش پایه تنها با یک جستجوی متمرکز و محدود روبرو است که در برخی موارد ممکن است نتایج مطلوبی به همراه نداشته باشد. در مقابل، روش چند شروعی با ایجاد تنوع در نقاط شروع اولیه، بهینه‌سازی بهتری را ارائه می‌دهد. در این روش، الگوریتم چندین بار از نقاط شروع مختلف آغاز می‌شود و هر بار جستجوی بهینه‌ای را انجام می‌دهد. این فرآیند با افزایش شانس یافتن راه حل‌های بهینه‌تر همراه است. در نهایت، با مقایسه و انتخاب بهترین راه حل از میان تمامی جستجوهای انجام شده، روش چند شروعی می‌تواند به طور قابل ملاحظه‌ای نتایج بهینه‌تری را نسبت به روش پایه ارائه دهد. این مزیت به خصوص در مسائل پیچیده و غیرخطی که فضای جستجوی بزرگی دارند، بیشتر مشهود است.

۲,۳. تشخیص بیماری با الگوریتم نایو بیز

الگوریتم طبقه‌بندی بیزین، یک تکنیک آماری است که بر پایه احتمال تعلق داده‌ها به کلاس‌ها به طبقه بندی می‌پردازد. این روش بر اساس قضیه بیز پایه‌گذاری شده و در مقایسه با سایر تکنیک‌ها، عملکرد مناسبی دارد. همچنین این الگوریتم سریع عمل می‌کند. مدل ابتدایی این تکنیک، به نام الگوریتم نایو بیز شناخته می‌شود. این مدل بر این فرض استوار است که تأثیر هر ویژگی بر روی دیگر ویژگی‌ها مستقل است، که این امر به کاهش حجم محاسبات کمک می‌کند.

فرض کنید که x یک نمونه داده است که توسط دسته‌ای از ویژگی‌ها تعریف می‌شود. فرضیه H نیز وجود دارد که تعلق x را با یک کلاس خاص مثل C مشخص می‌کند. در این صورت، برای انجام طبقه‌بندی، محاسبه P(H|x) نیاز می‌باشد. فرمول محاسبه احتمال بیز به صورت رابطه ۴ است.

$$P(H|x) = \frac{P(x|H) \times P(H)}{P(x)} \quad (۴)$$

در این رابطه، $P(H|x)$ احتمال پسین نامیده می‌شود. مقادیر $P(H)$ و $P(x)$ را میتوان از داده‌های آموزشی مشخص کرد که به آنها احتمالات پیشین گفته می‌شود. همچنین $P(x|H)$ نشان‌دهنده احتمال مشاهده داده x است، به شرطی که فرضیه H صادق باشد.
جهت تشخیص بیماری نمونه داده‌ای $X = (x1, x2, \dots, xn)$ (n تعداد ویژگی‌ها می‌باشد)، کلاسی برای X پیش بینی می‌شود که بالاترین احتمال پسین را داشته باشد. پس باید کلاس Ci جستجو شود که بالاترین مقدار $P(Ci|X)$ را داشته باشد.
قابل ذکر است $P(X)$ که مخرج کسر احتمال بیز است، برای تمامی کلاس‌ها یکسان می‌باشد. در نتیجه فقط لازم است عبارت $P(X|Ci)\times P(Ci)$ بیشینه گردد. اگر در مجموعه داده تعداد ویژگی‌ها زیاد باشد، آنگاه محاسبه $P(X|Ci)$ خیلی زمان‌بر می‌شود. برای ساده‌سازی محاسبات در الگوریتم نایو بیز، فرض می‌شود تمامی ویژگی‌ها مستقل از یکدیگر می‌باشند. بنابراین احتمال تعلق داده X به کلاس Ci برابر با ضرب احتمالات تعلق ویژگی‌های داده X به کلاس Ci خواهد بود.
در ادامه یک نمونه مثال برای تشخیص بیماری کووید-۱۹ بررسی خواهد شد. فرض کنید یک مجموعه داده آموزشی از بیماران کووید-۱۹ مطابق با جدول ۲ در دسترس باشد.

جدول ۲: مجموعه داده نمونه

ID	age	pneumonia	intubed	Class
1	yong	yes	T1	C1
2	middle	no	T2	C1
3	old	yes	T1	C1
4	yong	no	T2	C2
5	old	no	T3	C1
6	middle	yes	T1	C1
7	yong	no	T2	C2
8	middle	no	T1	C1
9	yong	no	T3	C1
10	old	yes	T1	C2

قابل مشاهده است که این مجموعه داده دارای ده نمونه داده است که هر کدام شامل سه ویژگی هستند و دو برچسب کلاس $C1$ و $C2$ به آن‌ها اختصاص داده شده است. کلاس $C1$ نشان‌دهنده افراد مبتلا به کووید-۱۹ و کلاس $C2$ نشان‌دهنده افراد سالم است.
اکنون فرض کنید که یک داده ورودی با ویژگی مشابه این دیتاست به شکل $X = (age=old, pneumonia=yes, intubed=T1)$ وجود دارد. بر اساس الگوریتم نایو بیز، تشخیص بیماری این داده ورودی به روش زیر صورت می‌پذیرد:

$P(C1) = 7/10 = 0.7$

$P(C2) = 3/10 = 0.3$

$P(age=old \mid Class=C1) = 2/7 = 0.285$
$P(age=old \mid Class=C2) = 1/3 = 0.333$
$P(pneumonia=yes \mid Class=C1) = 3/7 = 0.428$
$P(pneumonia=yes \mid Class=C2) = 1/3 = 0.333$
$P(intubed=T1 \mid Class=C1) = 4/7 = 0.571$
$P(intubed=T1 \mid Class=C2) = 1/3 = 0.333$

$P(X \mid Class=C1) = 0.285 \times 0.428 \times 0.571 = 0.069$
$P(X \mid Class=C2) = 0.333 \times 0.333 \times 0.333 = 0.036$

$P(X \mid Class=C1) \times P(C1) = 0.069 \times 0.7 = 0.048$ ✓
$P(X \mid Class=C2) \times P(C2) = 0.036 \times 0.3 = 0.010$ ×

بر اساس محاسبه‌های انجام شده، داده ورودی به کلاس مبتلایان به کووید-۱۹ تعلق دارد. زیرا احتمال تعلق این داده، به کلاس C1 (افراد بیمار) بیشتر از احتمال تعلق آن به کلاس C2 (افراد سالم) است.

۴. نتایج تجربی

در این بخش، روش پیشنهادی برای شناسایی بیماران کووید-۱۹ با سایر روش‌ها مورد مقایسه قرار می‌گیرد. در ادامه ابتدا شرایط ارزیابی توضیح داده می‌شود و پس از آن، نتایج ارزیابی‌ها ارائه می‌شوند. توجه می‌شود که در ارزیابی‌ها، شرایط برای تمامی روش‌ها یکسان فراهم شده تا ارزیابی‌های عملی به نحوی عادلانه انجام پذیرد.

● **محیط ارزیابی:** روش پیشنهادی برای شناسایی بیماران کووید-۱۹ به کمک زبان برنامه‌نویسی Matlab پیاده‌سازی شده است. آزمایش‌ها و اررزیابی‌های عملی بر روی یک کامپیوتر با پردازنده‌ی مرکزی Intel 2410M 2.3GHz Core i5 2.9 GHz، حافظه‌ی رم ۴GB DDR3 و حافظه‌ی هارد ۶۰۰GB SATA انجام شده است.

● **مجموعه داده:** برای آزمایش‌های عملی و مقایسه‌ی روش پیشنهاد شده با سایر روش‌ها، از مجموعه داده COVID-19 Dataset [۲۶] استفاده شده است. این مجموعه داده شامل اطلاعات افراد مشکوک به کووید-۱۹ است که شامل افراد مبتلا و سالم می‌باشد. مجموعه داده دارای ۲۱ ویژگی منحصر به فرد است و برچسب‌ها شامل مقادیر ۱ تا ۴ هستند. مقادیر ۱ تا ۳ نشان‌دهنده‌ی درجه‌های مختلف بیماری کووید-۱۹ و مقدار ۴ و بیشتر نشان‌دهنده‌ی عدم ابتلا به این بیماری است. اطلاعات ۱،۰۴۸،۵۷۶ بیمار مختلف در این مجموعه موجود می‌باشد. برای هر داده، اطلاعاتی مانند تاریخ بستری، جنسیت، وضعیت دیابت، وضعیت اینتوبه، درگیری ریه و غیره جمع‌آوری شده است. از بین داده‌های موجود در این مجوعه داده، ۶۲,۶ درصد مربوط به افراد سالم و ۳۷,۴ درصد مربوط به افراد بیمار می‌باشد. در مجموع ۷ کلاس در این مجموعه داده موجود می‌باشد که داده‌های متعلق به برخی از کلاس‌ها کمتر و برخی بیشتر می‌باشد. یعنی همه کلاس‌ها یه صورت متوازن نمی‌باشند. اما مجموع داده‌های کلاس‌های افراد بیمار (۱ تا ۳) و مجموع داده‌های کلاس‌های افراد سالم (۴ تا ۷) تا حد قابل قبولی متوازن می‌باشد. بنابراین مشکلات داده‌های نامتوازن در آزمایش‌ها رخ نخواهد داد. علاوه بر نسخه کامل این مجموعه داده، یک مجموعه کوچکتر تحت عنوان Sampled-C19 از داده‌ها انتخاب و در

آزمایش‌ها مورد استفاده قرار گرفت. این مجموعه داده جدید حاوی اطلاعات ۱۰۰،۰۰۰ بیمار می‌باشد. نمونه‌برداری داده‌ها برای این مجموعه به صورت کاملاً تصادفی صورت گرفت. علت این امر این است که روش‌ها در شرایطی با تنوع کمتر از داده‌ها نیز مورد ارزیابی قرار بگیرند. قابل ذکر است که در تمامی آزمایش‌های عملی انجام شده، ۸۰ درصد داده‌ها برای مرحله‌ی آموزش و ۲۰ درصد داده‌ها برای مرحله‌ی آزمایش استفاده شده است.

● **روش‌ها:** برای آزمایش‌های عملی از سه روش استفاده شد. توجه شود که تمامی روش‌های انتخاب شده مبتنی بر انتخاب ویژگی و الگوریتم بیزین ساده می‌باشد. علت این انتخاب این است که مقایسه روش پیشنهادی با دیگر روش‌ها به صورت عادلانه صورت بگیرد. جزئیات این روش‌ها به شرح ذیل می‌باشد:

- **روش PSO-NB:** این روش از الگوریتم‌های ازدحام ذرات و نایو بیز برای تشخیص بیماری کووید-۱۹ استفاده می‌کند. انتخاب ویژگی با استفاده از الگوریتم ازدحام ذرات و تشخیص بیماری با الگوریتم نایو بیز انجام می‌شود. جزئیات این روش در [۱۸] بیان شده است.

- **روش CL-NB:** این روش بر اساس انتخاب ویژگی و الگوریتم نایو بیز به شناسایی بیماران کووید-۱۹ می‌پردازد. نوع انتخاب ویژگی استفاده شده در این روش، انتخاب ویژگی مبتنی بر خوشه‌بندی می‌باشد. جزئیات این روش در [۲۰] بیان شده است.

- **روش GA-NB:** این روش به کمک الگوریتم ژنتیک به انتخاب ویژگی‌های مناسب از بین ویژگی‌های موجود می‌پردازد. تشخیص بیماری کووید-۱۹ نیز به وسیله الگوریتم نایو بیز صورت می‌گیرد. این روش مشابه تحقیق [۱۸] پیاده‌سازی شده، با این تفاوت مه به جای الگوریتم PSO از الگوریتم ژنتیک استفاده شده است.

- **روش DKS-NB :** این روش از چهار الگوریتم درخت تصمیم، کی-نزدیک‌ترین همسایه، ماشین بردار پشتیبان و نایو بیز برای تشخیص بیماران کووید-۱۹ استفاده می‌کند. جزئیات این روش در [۲۲] بیان شده است. در آزمایش‌ها به جای تصاویر سی تی اسکن، از مجموعه داده معرفی شده برای ارزیابی این روش استفاده می‌شود.

- **روش TSR-NB:** این روش، راهکار پیشنهادی برای تشخیص بیماری کووید-۱۹ در این مقاله است. در روش پیشنهاد شده، انتخاب ویژگی بر اساس الگوریتم TSR و تشخیص بر اساس الگوریتم نایو بیز صورت می‌گیرد. جزئیات این روش در بخش ۳ بیان شده است.

● **معیار مقایسه و متریک‌ها:** برای مقایسه‌ی روش‌های اشاره شده به صورت عملی، از سه معیار دقت، فراخوانی و امتیاز-F1 استفاده می‌شود که این معیارها رایج‌ترین معیارها برای مقایسه‌ی الگوریتم‌های طبقه‌بندی در پژوهش‌های مختلف است. این معیارها نشان می‌دهند که روش تا چه اندازه به درستی عمل کرده است. معیارهای دقت، فراخوانی و امتیاز-F1 به ترتیب به صورت رابطه‌های ۵، ۶ و ۷ محاسبه می‌شوند.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (۵)$$

$$Recall = \frac{TP}{TP + FN} \quad (۶)$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} \quad (۷)$$

پارامترهای به کار رفته در این رابطه‌ها عبارت اند از:

- TP: نشان دهنده تعداد مثبت صحیح است.
- FP: نشان دهنده تعداد مثبت کاذب است.
- TN: نشان دهنده تعداد منفی صحیح است.
- FN: نشان دهنده تعداد منفی کاذب است.

جهت مقایسه روش‌ها از نظر این معیارها، از ارزیابی‌های مختلف و محاسبه بهترین، میانگین و انحراف معیار استفاده شده است. در این ارزیابی‌ها، هر روش چند بار اجرا شده و هر بار معیار مربوطه محاسبه و ثبت گردیده است. در نهایت، موارد ذکر شده برای هر روش محاسبه و بر پایه آن‌ها مقایسه انجام شده است.

در ادامه ارزیابی‌های زمانی با استفاده از دو متریک انجام شده است. متریک اول، افزایش تعداد ویژگی‌های به کار رفته و متریک دوم، افزایش تعداد نمونه‌های داده در فرآیند تشخیص است. در آزمایش‌های انجام شده، مقدار زمان اجرای روش‌ها با توجه به افزایش این متریک‌ها مورد بررسی قرار گرفته است.

در انتها روش‌ها بر اساس منحنی ROC مورد ارزیابی و مقایسه قرار گرفتند. این منحنی یک روش کمی ارزیابی برای روش‌های مبتنی بر طبقه‌بندی می‌باشد و به صورت گرافیکی میزان عملکرد روش‌ها را نمایش می‌دهد.

۱،۴. ارزیابی با اجراهای مختلف

در این آزمایش‌ها، روش‌ها از طریق اجراهای متفاوت مورد سنجش و مقایسه قرار گرفته‌اند. در این آزمایش‌ها، هر کدام از روش‌ها، پانزده بار به اجرا درآمدند. در هر بار اجرا، مقدار دقت، فراخوانی و امتیاز-F1 برای تمامی روش‌ها محاسبه گردید. همان طور که اشاره شد در مرحله اول روش پیشنهادی ویژگی‌های مناسب به کمک الگوریتم TSR انتخاب می‌شوند. ویژگی‌های مجموعه داده COVID-19 Dataset در جدول ۳ قابل مشاهد می‌باشد.

جدول ۳: ویژگی‌های مجموعه داده COVID-19 Dataset

ردیف	ویژگی	توضیحات
۱	USMER	تحت مراقبت قرار گرفته یا خیر
۲	MEDICAL_UNIT	نوع واحد پزشکی بستری
۳	SEX	جنسیت بیمار
۴	PATIENT_TYPE	نوع بیمار (بستری/سرپایی)
۵	DATE_DIED	تاریخ فوت بیمار
۶	INTUBED	وضعیت انتوبه (لوله‌گذاری) بیمار

۷	PNEUMONIA	وضعیت ابتلا به ذات‌الریه در بیمار
۸	AGE	سن بیمار
۹	PREGNANT	وضعیت بارداری بیمار
۱۰	DIABETES	وجود بیماری دیابت در بیمار
۱۱	COPD	وجود بیماری مزمن انسدادی ریه در بیمار
۱۲	ASTHMA	وضعیت ابتلا به آسم در بیمار
۱۳	INMSUPR	وضعیت سرکوب سیستم ایمنی بیمار
۱۴	HIPERTENSION	وجود فشار خون بالا در بیمار
۱۵	OTHER_DISEASE	وجود بیماری‌های دیگر در بیمار
۱۶	CARDIOVASCULAR	وضعیت بیماری‌های قلبی و عروقی در بیمار
۱۷	OBESITY	وضعیت چاقی بیمار
۱۸	RENAL_CHRONIC	وجود بیماری مزمن کلیوی در بیمار
۱۹	TOBACCO	وضعیت مصرف دخانیات در بیمار
۲۰	CLASIFFICATION_FINAL	نوع نهایی تشخیص بیماری در بیمار
۲۱	ICU	وضعیت بستری شدن بیمار در ICU

یک نمونه از انتخاب ویژگی‌ها برای بهترین حالت (بهترین میزان برازندگی) که ویژگی‌های انتخاب شده در آن به تعداد آن ۱۴ عدد است، به صورت زیر می‌باشد:

SEX, PATIENT_TYPE, INTUBED, PNEUMONIA, AGE, PREGNANT, DIABETES, ASTHMA, HIPERTENSION, CARDIOVASCULAR, OBESITY, RENAL_CHRONIC, TOBACCO, ICU

همان طور که مشاهده می‌شود ویژگی‌های جنسیت، نوع بیمار، وضعیت انتوبه، وضعیت ذات الریه، سن، بارداری، دیابت، آسم، فشار خون بالا، بیماری قلبی و عروقی، چاقی، بیماری کلیوی، مصرف دخانیات و وضعیت بستری در ICU تاثیر بیشتری در دقت شناسایی بیماری دارند.

پس از محاسبه معیارهای مطرح شده برای هر سه روش در تمامی اجراها، سه شاخص برای ارزیابی آنها در نظر گرفته شد. شاخص نخست، بهترین اجرا می‌باشد؛ به این معنا که یک اجرایی که بالاترین میزان مقدار معیار را داشته باشد، برای مقایسه روش‌ها مد نظر قرار می‌گیرد. شاخص دوم برای ارزیابی روش‌ها، شاخص میانگین مقدار معیار در اجراها می‌باشد. برای این منظور، میانگین مقدار معیار هر روش در پانزده بار اجرا محاسبه شده و برای مقایسه روش‌ها با یکدیگر به کار رفته است. هر اندازه که میزان میانگین مقدار معیار بالاتر باشد، نشان‌دهنده‌ی نتیجه‌ی بهتری است. شاخص سوم برای ارزیابی روش‌ها، انحراف از معیار یا پراکندگی داده‌ها نسبت به میانگین است. انحراف از معیار نشان می‌دهد که به طور متوسط، داده‌ها چه مقدار از میانگین فاصله دارند. هر اندازه که انحراف از معیار به صفر نزدیک‌تر باشد، روش عملکرد بهتری دارد. نتایج به دست آمده از ارزیابی با اجراهای مختلف برای مجموعه داده COVID-19 Dataset، در جداول شماره ۴ تا ۶ قابل ملاحظه است.

جدول ۴: ارزیابی دقت روش‌ها با مجموعه داده COVID-19 Dataset

اجرا	DKS-NB	GA-NB	PSO-NB	CL-NB	TSR-NB
۱	0.93	0.92	0.90	0.85	0.95

۲	0.95	0.94	0.93	0.93	0.98
۳	0.94	0.90	0.91	0.92	0.97
۴	0.93	0.92	0.89	0.90	0.95
۵	0.96	0.88	0.92	0.91	0.98
۶	0.94	0.90	0.90	0.85	0.98
۷	0.94	0.92	0.93	0.90	0.96
۸	0.88	0.91	0.89	0.86	0.95
۹	0.96	0.90	0.88	0.87	0.97
۱۰	0.95	0.89	0.93	0.88	0.98
۱۱	0.94	0.90	0.88	0.91	0.94
۱۲	0.96	0.93	0.92	0.88	0.96
۱۳	0.95	0.93	0.91	0.90	0.94
۱۴	0.94	0.89	0.93	0.91	0.95
۱۵	0.93	0.94	0.88	0.86	0.94
بهترین	**0.96**	**0.94**	**0.93**	**0.91**	**0.98**
میانگین	**0.94**	**0.91**	**0.90**	**0.88**	**0.96**
انحراف از معیار	**0.0003**	**0.0003**	**0.0003**	**0.0006**	**0.0002**

جدول ۵: ارزیابی فراخوانی روش‌ها با مجموعه داده COVID-19 Dataset

اجرا	DKS-NB	GA-NB	PSO-NB	CL-NB	TSR-NB
۱	0.97	0.94	0.92	0.89	0.99
۲	0.98	0.93	0.94	0.88	0.96
۳	0.98	0.94	0.89	0.93	0.98
۴	0.97	0.92	0.90	0.87	0.98
۵	0.99	0.91	0.92	0.93	0.95
۶	0.98	0.89	0.93	0.87	0.98
۷	0.95	0.96	0.91	0.92	0.97
۸	0.99	0.93	0.91	0.87	0.96
۹	0.98	0.95	0.90	0.92	0.97
۱۰	0.99	0.93	0.90	0.87	0.95
۱۱	0.99	0.90	0.94	0.87	0.99
۱۲	0.98	0.93	0.91	0.92	0.95
۱۳	0.97	0.92	0.93	0.91	0.99
۱۴	0.98	0.95	0.94	0.94	0.96

۱۵	0.99	0.95	0.94	0.92	0.97
بهترین	**0.99**	**0.96**	**0.94**	**0.94**	**0.99**
میانگین	**0.97**	**0.93**	**0.91**	**0.90**	**0.97**
انحراف از معیار	**0.0001**	**0.0003**	**0.0002**	**0.0006**	**0.0002**

جدول ۶: ارزیابی امتیاز-F1 روش‌ها با مجموعه داده COVID-19 Dataset

اجرا	DKS-NB	GA-NB	PSO-NB	CL-NB	TSR-NB
۱	0.98	0.95	0.93	0.93	0.98
۲	0.96	0.89	0.89	0.88	0.97
۳	0.98	0.92	0.90	0.93	0.98
۴	0.96	0.93	0.88	0.93	0.95
۵	0.99	0.92	0.93	0.91	0.95
۶	0.99	0.93	0.91	0.90	0.95
۷	0.98	0.91	0.90	0.87	0.98
۸	0.95	0.92	0.93	0.92	0.97
۹	0.96	0.93	0.90	0.87	0.96
۱۰	0.97	0.93	0.93	0.86	0.97
۱۱	0.98	0.90	0.91	0.87	0.96
۱۲	0.99	0.94	0.93	0.91	0.95
۱۳	0.98	0.92	0.92	0.86	0.98
۱۴	0.98	0.95	0.93	0.92	0.95
۱۵	0.97	0.94	0.92	0.88	0.98
بهترین	**0.99**	**0.95**	**0.93**	**0.93**	**0.98**
میانگین	**0.97**	**0.92**	**0.91**	**0.89**	**0.96**
انحراف از معیار	**0.0001**	**0.0002**	**0.0002**	**0.0006**	**0.0001**

طبق آنچه که در جدول ۴ مشخص است، میزان دقت بهترین اجرای روش پیشنهاد شده یعنی TSR-NB برابر ۹۹ درصد است. میزان دقت بهترین اجرای روش CL-NB معادل با ۹۱ درصد، میزان دقت بهترین اجرای روش PSO-NB برابر با ۹۳ درصد، میزان دقت بهترین اجرای روش GA-NB معادل با ۹۴ درصد و میزان دقت بهترین اجرای روش DKS-NB برابر با ۹۶ درصد می‌باشد. بنابراین، بر اساس شاخص بهترین اجرا، روش پیشنهاد شده از لحاظ دقت عملکردی بهتری از چهار روش دیگر ارائه داده است. با توجه به داده‌های ارائه شده در جدول ۴، میانگین دقت روش پیشنهادی یعنی TSR-NB، ۹۶ درصد می‌باشد. در مقابل، میانگین دقت روش‌های CL-NB، PSO-NB، GA-NB و DKS-NB به ترتیب ۸۸ درصد، ۹۰ درصد، ۹۱ درصد و ۹۴ درصد است. بنابراین بر اساس شاخص میانگین، روش پیشنهادی عملکرد بهتری از چهار روش دیگر دارد. همچنین بر اساس اطلاعات ارائه

شده در جدول ۴، پایداری روش پیشنهادی یعنی TSR-NB با نزدیک بودن انحراف معیار به صفر، بیشتر از روش‌های دیگر است. روش‌های PSO-NB، GA-NB و DKS-NB عملکرد مشابه‌ای دارند، اما روش CL-NB از لحاظ پایداری ضعیف‌ترین عملکرد را دارد. یکی از برتری روش پیشنهادی انتخاب ویژگی مناسب در روش پیشنهادی است.

همان طور که در جدول ۵ قابل مشاهده است، میزان فراخوانی بهترین اجرای روش‌های TSR-NB و DKS-NB برابر با ۹۹ درصد است که بهترین عملکرد می‌باشد. پس از آنها GA-NB بهترین عملکرد را دارد و عملکرد روش‌های CL-NB و PSO-NB مشابه با یکدیگر و ضعیف‌تر می‌باشد. از نظر میانگین فراخوانی نیز روش پیشنهادی یعنی TSR-NB و روش DKS-NB بهترین عملکرد با مقدار ۹۷ درصد را دارند.میانگین فراخوانی روش‌های GA-NB، PSO-NB و CL-NB به ترتیب برابر با ۹۳ درصد، ۹۱ درصد و ۹۰ درصد می‌باشد. از نظر انحراف معیار روش DKS-NB بهترین عملکرد را دارد و بعد از آن روش‌های TSR-NB و PSO-NB عملکرد بهتری دارند و مقدار انحراف از معیار آنها به صفر نزدیکتر می‌باشد. در بین روش‌های GA-NB و CL-NB نیز روش GA-NB عملکرد بهتری دارد. دلیل اصلی عملکرد مناسب روش TSR-NB، استفاده مؤثر از الگوریتم TSR برای انتخاب ویژگی‌های مناسب است.

در جدول ۶ عملکرد روش‌ها از نظر امتیاز-F1 مشخص شده است. از نظر بهترین مقدار امتیاز-F1 روش‌های DSK-NB بهترین عملکرد را دارد، زیرا از طبقه‌بندهای زیادی استفاده می‌کند. بعد از آن روش پیشنهادی یعنی TSR-NB عملکرد بهتری از سایر روش‌ها دارد. در ادامه روش GA-NB علکرد مناسب‌تری دارد و عملکرد روش‌های PSO-NB و CL-NB مشابه با یکدیگر است. از نظر میانگین مقدار امتیاز-F1 نیز روش DSK-NB بهترین عملرد را دارد و بعد از آن روش TSR-NB بهترین می‌باشد. در ادامه عملکرد سایر روش‌ها به ترتیب از بهترین به ضعیف‌ترین به صورت GA-NB، PSO-NB و CL-NB است. از نظر انحراف معیار امتیاز-F1 روش‌های TSR-NB و DKS-NB بهترین عملکرد را دارند. سپس عملکرد روش‌های GA-NB و PSO-NB مناسب‌تر می‌باشد. ضعیف‌ترین عملکرد را نیز روش CL-NB دارد. یکی از دلایل عملکرد مناسب روش پیشنهادی اجرای الگوریتم TSR به صورت چندشروعی می‌باشد.

در ادامه روش‌ها بر اساس مجموعه داده کوچکتر Sampled-C19 و معیارهای دقت، فراخوانی و امتیاز-F1 در اجراهای مختلف مورد ارزیابی قرار گرفت. نتایج حاصل از این ارزیابی‌ها در جداول شماره ۷ تا ۹ قابل مشاهده می‌باشد.

جدول ۷: ارزیابی دقت روش‌ها با مجموعه داده Sampled-C19

اجرا	DKS-NB	GA-NB	PSO-NB	CL-NB	TSR-NB
۱	0.87	0.86	0.82	0.85	0.88
۲	0.84	0.80	0.82	0.83	0.91
۳	0.88	0.87	0.85	0.79	0.90
۴	0.87	0.85	0.83	0.84	0.88
۵	0.86	0.85	0.86	0.79	0.90
۶	0.88	0.83	0.81	0.83	0.91
۷	0.86	0.85	0.86	0.78	0.87
۸	0.81	0.83	0.80	0.78	0.91
۹	0.87	0.86	0.85	0.84	0.88
۱۰	0.85	0.81	0.84	0.79	0.88

۱۱	0.89	0.83	0.83	0.81	0.91
۱۲	0.87	0.85	0.84	0.84	0.89
۱۳	0.89	0.84	0.84	0.81	0.87
۱۴	0.86	0.80	0.86	0.82	0.87
۱۵	0.87	0.82	0.81	0.84	0.89
بهترین	**0.89**	**0.87**	**0.86**	**0.85**	**0.91**
میانگین	**0.86**	**0.83**	**0.83**	**0.81**	**0.89**
انحراف از معیار	**0.0003**	**0.0004**	**0.0003**	**0.0005**	**0.0002**

جدول ۸: ارزیابی فراخوانی روش‌ها با مجموعه داده Sampled-C19

اجرا	DKS-NB	GA-NB	PSO-NB	CL-NB	TSR-NB
۱	0.89	0.85	0.84	0.81	0.88
۲	0.92	0.86	0.85	0.86	0.91
۳	0.91	0.86	0.82	0.85	0.91
۴	0.88	0.88	0.82	0.80	0.92
۵	0.92	0.83	0.83	0.79	0.88
۶	0.91	0.86	0.81	0.85	0.92
۷	0.92	0.87	0.87	0.84	0.92
۸	0.91	0.85	0.84	0.87	0.89
۹	0.91	0.82	0.86	0.85	0.89
۱۰	0.91	0.86	0.87	0.81	0.90
۱۱	0.90	0.82	0.87	0.86	0.88
۱۲	0.92	0.89	0.83	0.79	0.91
۱۳	0.92	0.86	0.86	0.85	0.90
۱۴	0.91	0.88	0.84	0.82	0.89
۱۵	0.89	0.85	0.85	0.80	0.90
بهترین	**0.92**	**0.89**	**0.87**	**0.87**	**0.92**
میانگین	**0.90**	**0.85**	**0.84**	**0.83**	**0.90**
انحراف از معیار	**0.0001**	**0.0003**	**0.0003**	**0.0007**	**0.0002**

جدول ۹: ارزیابی امتیاز-F1 روش‌ها با مجموعه داده Sampled-C19

اجرا	DKS-NB	GA-NB	PSO-NB	CL-NB	TSR-NB

۱	0.90	0.88	0.85	0.81	0.88
۲	0.88	0.86	0.86	0.84	0.90
۳	0.89	0.85	0.84	0.79	0.91
۴	0.89	0.88	0.82	0.79	0.88
۵	0.91	0.82	0.82	0.84	0.91
۶	0.92	0.83	0.81	0.79	0.89
۷	0.91	0.87	0.86	0.85	0.91
۸	0.92	0.85	0.86	0.86	0.88
۹	0.92	0.86	0.84	0.84	0.91
۱۰	0.91	0.84	0.86	0.86	0.88
۱۱	0.89	0.83	0.86	0.81	0.91
۱۲	0.91	0.85	0.84	0.85	0.88
۱۳	0.91	0.86	0.84	0.83	0.90
۱۴	0.89	0.84	0.83	0.80	0.89
۱۵	0.89	0.86	0.83	0.78	0.89
بهترین	**0.92**	**0.88**	**0.86**	**0.86**	**0.91**
میانگین	**0.90**	**0.85**	**0.84**	**0.82**	**0.89**
انحراف از معیار	**0.0001**	**0.0002**	**0.0002**	**0.0007**	**0.0001**

در جدول ۷ نتایج ارزیابی دقت روش‌ها بر اساس مجموعه داده Sampled-C19 ارائه شده است. همان طور که قابل مشاهده است مقادیر دقت حاصل شده کمتر از آزمایش‌های قبلی است. زیرا مجموعه داده Sampled-C19 تعداد داده کمتری دارد و در نتیجه یادگیری در آن در سطح پایین‌تری صورت می‌گیرد. در این ارزیابی‌ها روش پیشنهادی یعنی TSR-NB با ۹۱ درصد دقت بهترین عملکرد را از نظر بهترین اجرا دارد. از نظر میانگین دقت نیز روش پیشنهادی با ۸۹ درصد دقت از سایر روش‌ها عملکرد بهتری دارد. از نظر انحراف معیار نیز بهترین عملکرد برای روش TSR-NB می‌باشد. بعد از روش پیشنهادی DKS-NB بهترین عملکرد را دارد.

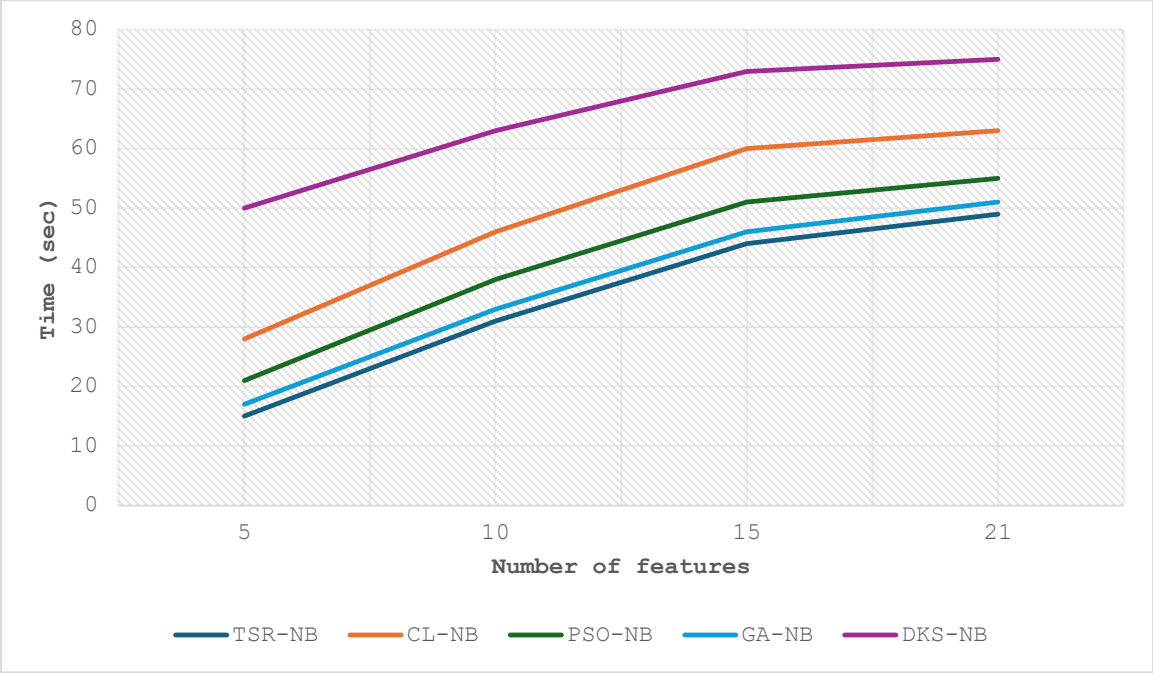
در جدول ۸ نتایج ارزیابی فراخوانی روش‌ها بر اساس مجموعه داده Sampled-C19 ارائه شده است. در این این ارزیابی‌ها از نظر بهترین اجرا و میانگین روش‌های TSR-NB و DKS-NB بهترین عملکرد را به ترتیب با ۹۲ درصد و ۹۰ درصد ارائه کرده‌اند. از نظر انحراف معیار، روش DLS-NB عملکرد بهتری از روش TSR-NB دارد و مقدار آن به صفر نزدیکتر می‌باشد. روش GA-NB بعد از این دو روش عملکرد مناسب‌تری دارد.

در جدول ۹ نتایج ارزیابی امتیاز-F1 روش‌ها بر اساس مجموعه داده Sampled-C19 ارائه شده است. در این ارزیابی‌ها از نظر بهترین اجرا و میانگین روش DKS-NB بهترین عملکرد را دارد و بعد از آن روش پیشنهادی بالاترین میزان مقادیر امتیاز-F1 را ارائه می‌کند. مقادیر امتیاز-F1 برای روش پیشنهادی برای بهترین اجرا و میانگین به ترتیب برابر با ۹۱ درصد و ۸۹ درصد است. از نظر انحراف معیار عملکرد دو روش مشابه با یکدیگر می‌باشد.

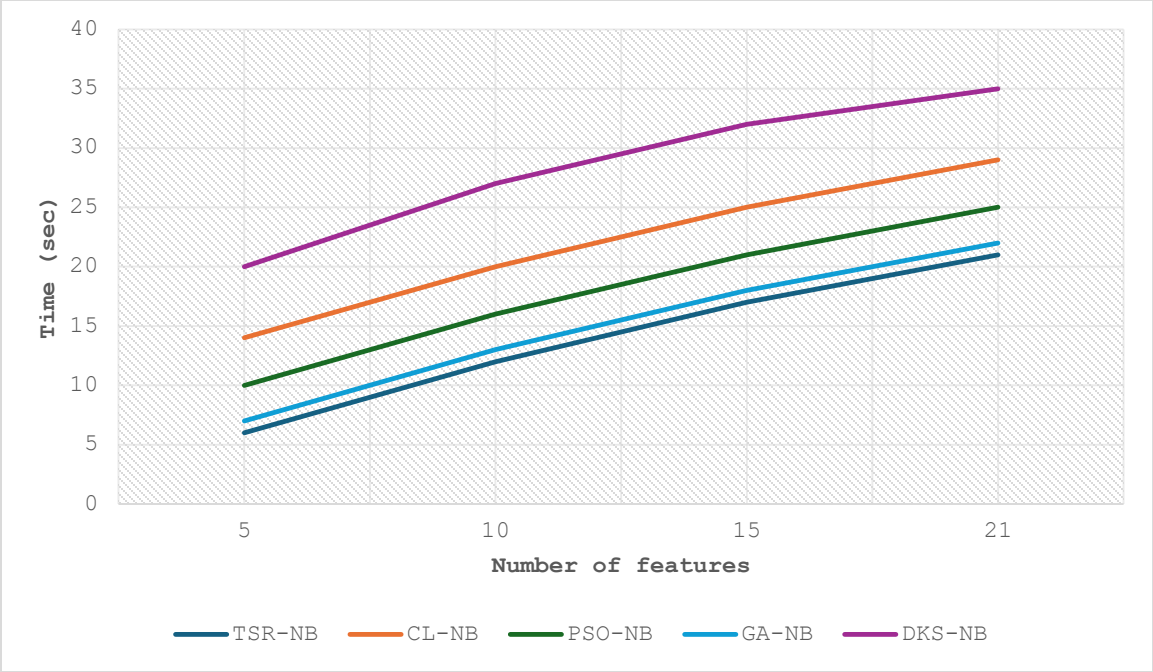
۲،۴. ارزیابی‌های زمانی

در این بخش، روش‌های مذکور جهت مقایسه‌های عملی با استفاده از دو متریک و بر پایه معیار زمان اجرا مورد سنجش قرار می‌گیرند. همانطور که بیان شده است، دو متریک به کار رفته شامل افزایش تعداد ویژگی‌ها و افزایش تعداد داده‌ها می‌باشند. در این ارزیابی‌ها هر چقدر زمان اجرای روشی کمتر باشد، عملکرد آن روش بهتر است.

ابتدا، روش‌ها با استفاده از متریک افزایش تعداد ویژگی‌های استفاده شده و بر پایه زمان اجرا، مورد مقایسه قرار گرفتند. جهت این کار، تعداد مختلفی از ویژگی‌ها انتخاب شد و روش‌ها با این تعداد متفاوت اجرا و زمان اجرای آن‌ها مورد ارزیابی قرار گرفت. تعداد ویژگی‌های مورد استفاده در ارزیابی‌های متفاوت برابر ۵، ۱۰، ۱۵ و ۲۱ بوده است. لازم به ذکر است که در ارزیابی‌ها، این تعداد ویژگی‌ها مد نظر قرار گرفته و پس از آن انتخاب ویژگی صورت می‌گیرد. نتایج به دست آمده از این ارزیابی‌ها در شکل‌های ۶ و ۷ آورده شده است.



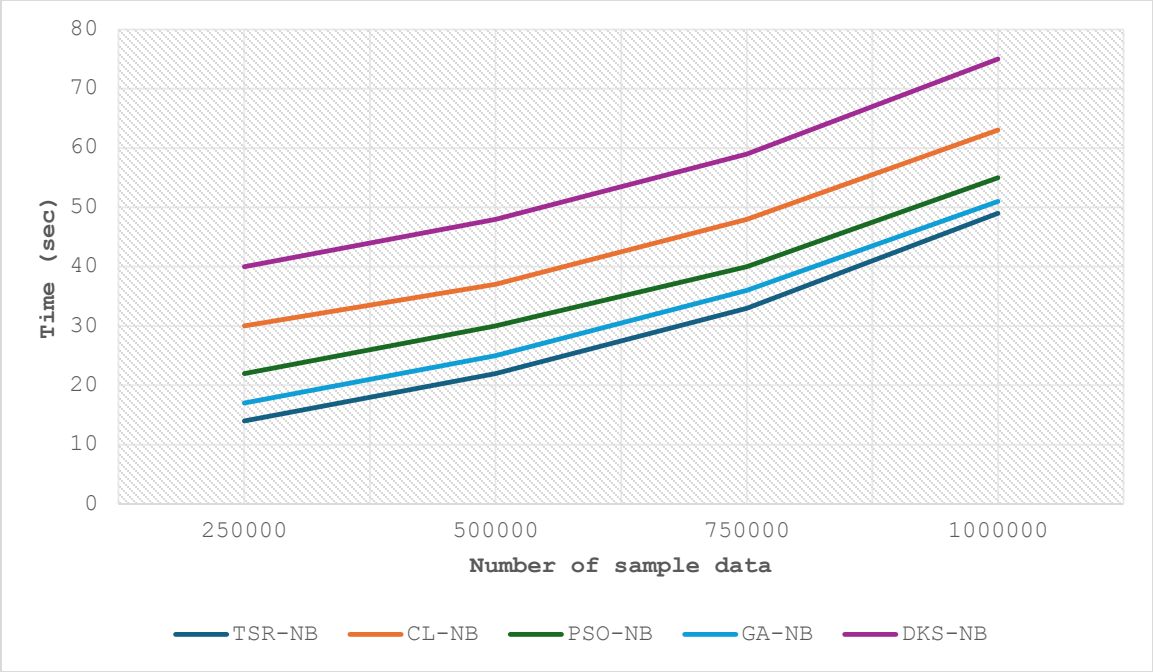
شکل ۶: ارزیابی زمان اجرای روش‌ها با متریک افزایش تعداد ویژگی‌ها و مجموعه داده COVID-19 Dataset



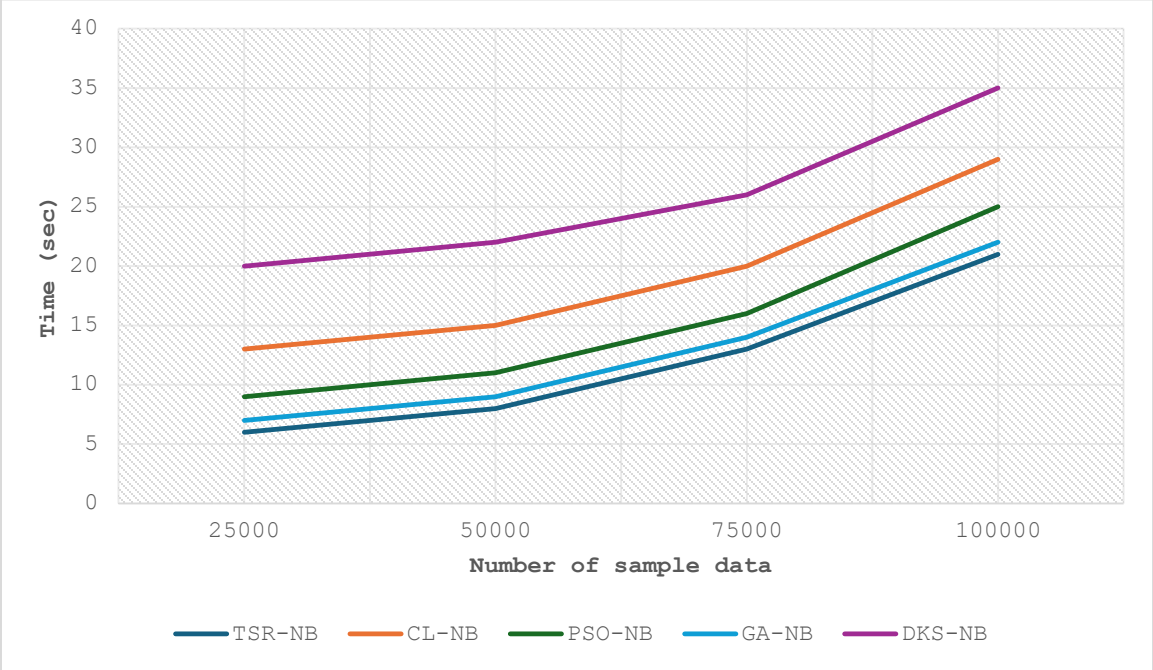
شکل ۷: ارزیابی زمان اجرای روش‌ها با متریک افزایش تعداد ویژگی‌ها و مجموعه داده Sampled-C19

همان طور که در هر دو شکل قابل مشاهده است، با افزایش تعداد ویژگی‌ها، زمان اجرای روش‌ها افزایش می‌یابد. در نمودارهای ارائه شده، نمودار روش پیشنهادی یعنی TSR-NB در تمامی موارد پایین‌تر از روش‌های دیگر قرار دارد که به این معنی است که روش پیشنهادی در همه ارزیابی‌ها عملکرد بهتری داشته است. پس از روش پیشنهادی، روش GA-NB بهترین عملکرد را دارد که نزدیک به روش پیشنهادی می‌باشد. روش بعدی که عملکرد بهتری دارد، روش PSO-NB است. عملکرد روش CL-NB مناسب نمی‌باشد، زیرا انتخاب ویژگی در آن زمانبر است. ضعیف‌ترین عملکرد را روش DKS-NB دارد. زیرا از تعداد مدل‌های زیادی استفاده می‌کند. دلیل اصلی برتری روش TSR-NB، استفاده مؤثر از الگوریتم TSR برای انتخاب ویژگی‌های مناسب است.

در ادامه این بخش، روش‌ها با استفاده از متریک افزایش تعداد داده و بر اساس زمان اجرا، مورد مقایسه قرار می‌گیرند. در این راستا، تعداد مختلفی از نمونه داده‌ها در نظر گرفته شده و روش‌ها با این تعداد مختلف اجرا و زمان اجرای آن‌ها سنجیده شده است. منظور از نمونه‌های داده اطلاعات ثبت‌شده بیماران است. تعداد داده‌ها در ارزیابی‌های متفاوت برای مجموعه داده COVID-19 Dataset برابر با ۲۵۰۰۰۰، ۵۰۰۰۰۰، ۷۵۰۰۰۰ و ۱۰۰۰۰۰۰ بوده است. برای مجموعه داده Sampled-C19 نیز تعداد داده‌های در نظر گرفته شده برابر با ۲۵۰۰۰، ۵۰۰۰۰، ۷۵۰۰۰ و ۱۰۰۰۰۰ می‌باشد. نتایج به دست آمده از این مقایسات در شکل‌های ۸ و ۹ ارائه شده است.



شکل ۸:ارزیابی زمان اجرای روش‌ها با متریک افزایش تعداد نمونه داده‌ها و مجموعه داده COVID-19 Dataset



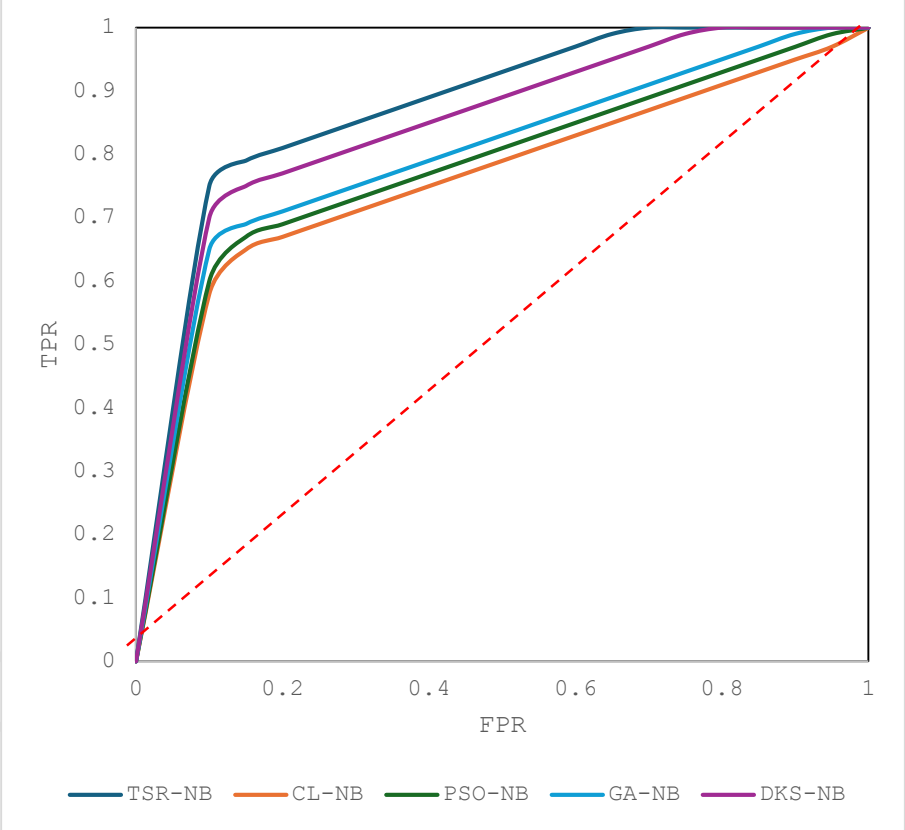
شکل ۹: ارزیابی زمان اجرای روش‌ها با متریک افزایش تعداد نمونه داده‌ها و مجموعه داده Sampled-C19

با توجه به نتایج ارزیابی‌ها، مشخص است که با افزایش حجم نمونه‌های داده، زمان اجرای روش‌های مورد بررسی، افزایش می‌یابد. روش‌هایی که بر پایه الگوریتم‌های فراابتکاری ارائه شده‌اند، یعنی TSR-NB، GA-NB و PSO-NB ، عملکرد مناسب‌تری دارند؛ زیرا این روش‌ها در انتخاب ویژگی‌ها کارآمدتر هستند. روش‌های CL-NB و DKS-NB زمان اجرای زیادی دارند. ضعیف‌ترین عملکرد برای روش DKS-NB می‌باشد که علی رغم داشتن دقت مناسب، زمان اجرای بالایی دارد. در میان روش‌های مبتنی بر

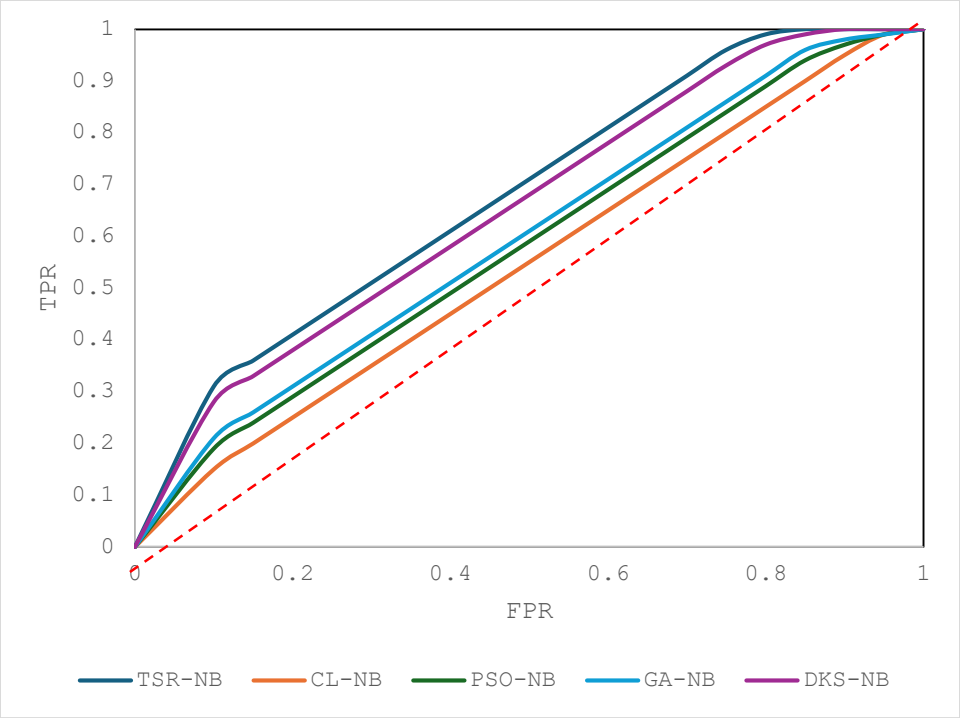
الگوریتم‌های فراابتکاری، روش پیشنهادی یعنی TSR-NB عملکرد برتری از خود نشان داده است. عملکرد روش PSO-NB ضعیف‌تر از دو روش دیگر فراابتکاری است، زیرا مسئله انتخاب ویژگی یک مسئله گسسته است و الگوریتم PSO یک الگوریتم پیوسته می‌باشد. در بین روش‌های TSR-NB و GA-NB، عملکرد روش پیشنهادی با اختلاف کمی بهتر است. علت این امر مدلسازی بهتر در روش پیشنهادی می‌باشد.

۳,۴. ارزیابی بر اساس منحنی ROC

در این بخش روش‌ها بر اساس منحنی ROC مورد ارزیابی و مقایسه قرا می‌گیرند. منحنی ROC یک ابزار کارآمد برای ارزیابی کارایی مدل‌های طبقه‌بندی در یادگیری ماشین می‌باشد. در این منحنی رابطه بین نرخ مثبت صحیح (محور عمودی) و نرخ مثبت کاذب (محور افقی) نمایش داده می‌شود. هر چقدر سطح زیر نمودار منحنی ROC برای روشی بیشتر باشد، عملکرد آن بهتر می‌باشد. منحنی ROC برای مجموعه داده‌های COVID-19 Dataset و Sampled-C19 در شکل‌های ۱۰ و ۱۱ قابل مشاهده می‌باشد.



شکل ۱۰:ارزیابی روش‌ها بر اساس منحنی ROC و مجموعه داده COVID-19 Dataset



شکل ۱۱:ارزیابی روش‌ها بر اساس منحنی ROC و مجموعه داده Sampled-C19

همان طور که در این شکل‌ها قابل مشاهده است، نمودار روش پیشنهادی یعنی TSR-NB برای هر دو مجموعه داده از سایر روش‌ها بالاتر است. بنابراین سطح زیر نمودار آن بیشتر از سایر روش‌ها و در نتیجه عملکرد آن بهتر از سایر روش‌ها می‌باشد. یعنی نرخ مثبت صحیح در آن بیشتر از نرخ مثبت کاذب، نسبت به دیگر روش‌ها می‌باشد. سطح زیر نمودار روش‌ها در مجموعه داده COVID-19 Dataset بیشتر از سطح زیر نمودار روش‌ها در مجموعه داده Sampled-C19 می‌باشد. زیرا تعداد داده‌ها در مجموعه داده COVID-19 Dataset بیشتر است و یادگیری در آن به صورت بهتری انجام می‌شود. پس از روش پیشنهادی، روش DSK-NB عملکرد بهتری دارد. در ادامه روش‌های GA-NB و PSO-NB عملکرد بهتری دارند که زدیک به یکدیگر می‌باشد. البته عملکرد روش GA-NB کمی بهتر از روش PSO-NB است. ضعیف‌ترین عملکرد را نیز روش NB-CL دارد.

در آزمایش‌های عملی انجام شده، کارایی روش پیشنهادی از دیدگاه‌های دقت، فراخوانی و امتیاز-F1 مورد بررسی قرار گرفت. در این راستا، روش پیشنهادی با چهار روش دیگر مقایسه شد. در ابتدا، مقایسه‌ها بر اساس اجراهای مختلف انجام شد. در این آزمایش‌ها، شاخص‌های بهترین اجرا، میانگین و انحراف معیار در نظر گرفته شد که روش پیشنهادی در بیشتر معیارها جز روش‌های برتر بود. سپس، روش‌ها بر اساس دو متریک و معیار زمان اجرا مورد مقایسه قرار گرفتند. دو متریک به کار رفته شامل افزایش تعداد ویژگی‌ها و افزایش تعداد نمونه‌های داده بود. در این مقایسه‌ها نیز، روش پیشنهادی عملکرد بهتری را نسبت به سایر روش‌ها نشان داد. در انتها نیز روش‌ها بر اساس منحنی ROC مورد مقایسه قرار گرفتند که در اینجا نیز روش پیشنهادی مناسب‌ترین عملکرد را داشت. دلیل اصلی برتری روش پیشنهادی، استفاده مؤثر از الگوریتم فراابتکاری TSR بوده است. علاوه بر این، در روش پیشنهادی، مسئله انتخاب ویژگی به خوبی در الگوریتم TSR مدل‌سازی شده و به صورت چند شروعی مورد استفاده قرار گرفته است. بر این اساس، بررسی‌های انجام شده نشان می‌دهند که روش پیشنهادی، عملکرد بهتری در مقایسه با سایر روش‌های مشابه ارائه می‌دهد.

۵. نتیجه‌گیری

کووید-۱۹ یک بیماری واگیردار می‌باشد که بر اثر کروناویروس سندرم حاد تنفسی ایجاد می‌گردد. شیوع این بیماری با تلفات زیاد در سراسر دنیا به صورت قابل توجهی بر زندگی افراد تأثیر گذاشته است. سرعت شیوع بیماری کووید-۱۹ نیز بسیار زیاد می‌باشد. بنابراین تشخیص به موقع بیماران مبتلا به کووید-۱۹ جهت انجام درمان زودتر و به منظور جلوگیری از گسترش آن، از اهمیت بسزایی برخوردار است. با در نظر گرفتن این موارد، پیشنهاد و ارائه روش‌هایی هوشمند برای تشخیص زودهنگام بیماری کووید-۱۹ بسیار حائز اهمیت می‌باشد. زیرا این روش‌ها غیرتهاجمی می‌باشند و به صورت خودکار با پردازش و تحلیل داده‌های بیماران گذشته به تشخیص بیماری می‌پردازند.

در این مقاله، یک روش برای شناسایی بیماری کووید-۱۹ با استفاده از الگوریتم‌های TSRو نایو بیز معرفی شده است. این روش ترکیبی از دو مرحله انتخاب ویژگی و تشخیص بیماری است. فرآیند انتخاب ویژگی با بهره‌گیری از الگوریتم TSR صورت می‌پذیرد، و تشخیص بیماری با استفاده از طبقه‌بند نایو بیز انجام می‌گیرد. روش پیشنهاد شده به صورت عملی با سایر روش‌های موجود مورد مقایسه قرار گرفته است. ارزیابی‌ها بر پایه معیارهای دقت، فراخوانی و امتیاز-F1 و با شاخص‌های بهترین، میانگین و انحراف معیار دقت در اجراهای متفاوت انجام شده‌اند. علاوه بر این، روش پیشنهادی با استفاده از معیار زمان اجرا و دو متریک افزایش تعداد ویژگی‌ها و افزایش تعداد نمونه داده‌ها و همچنین با استفاده از نمودار ROC با دیگر روش‌ها مقایسه شده است. در حالت کلی در ارزیابی‌های انجام شده، روش پیشنهاد شده، بیماران کووید-۱۹ را با دقت بالاتر و زمان کمتر نسبت به سایر روش‌های مشابه در این حوزه شناسایی کرده است.

منابع

[1] F. He, Y. Deng, and W. Li, "Coronavirus disease 2019: What we know?," *Journal of medical virology*, vol. 92, no. 7, pp. 719-725, Jul. 2020, https://doi.org/10.1002/jmv.25766.

[2] S. Chauhan, "Comprehensive review of coronavirus disease 2019 (COVID-19)," *Biomedical journal*, vol. 43, no. 4, pp. 334-340, Dec. 2020, https://doi.org/10.1016/j.bj.2020.05.023.

[3] I. Rahimi, F. Chen, and A. H. Gandomi, "A review on COVID-19 forecasting models," *Neural Computing and Applications*, vol. 1-11, 2021, https://doi.org/10.1007/s00521-020-05626-8.

[4] T. Singhal, "A review of coronavirus disease-2019 (COVID-19)," *The indian journal of pediatrics*, vol. 87, no. 4, pp. 281-286, Apr. 2020, https://doi.org/10.1007/s12098-020-03263-6.

[5] A. Dogan and D. Birant, "Machine learning and data mining in manufacturing," *Expert Systems with Applications*, vol. 166, p. 114060, Jan. 2021, https://doi.org/10.1016/j.eswa.2020.114060.

[6] P. C. Sen, M. Hajra, and M. Ghosh, "Supervised classification algorithms in machine learning: A survey and review," in *Emerging Technology in Modelling and Graphics*, Proc. IEM Graph 2018, Springer Singapore, 2018, pp. 99-111, https://doi.org/10.1007/978-981-13-7403-6_11.

[7] S. Chen, G. I. Webb, L. Liu, and X. Ma, "A novel selective naïve Bayes algorithm," *Knowledge-Based Systems*, vol. 192, p. 105361, Oct. 2020, https://doi.org/10.1016/j.knosys.2019.105361.

[8] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019)," *IEEE Access*, vol. 9, pp. 26766-26791, Feb. 2021, https://doi.org/10.1109/ACCESS.2021.3056407.

[9] M. Alimoradi, H. Azgomi, and A. Asghari, "Trees social relations optimization algorithm: A new Swarm-Based metaheuristic technique to solve continuous and discrete optimization problems," *Mathematics and Computers in Simulation*, vol. 194, pp. 629-664, Jan. 2022, https://doi.org/10.1016/j.matcom.2021.12.010.

[10] S. Shams, H. Azgomi, and A. Asghari, "Fuel assemblies loading pattern optimization of pressurized water reactors using the trees social relations algorithm," *Annals of Nuclear Energy*, vol. 192, p. 109963, Mar. 2023, https://doi.org/10.1016/j.anucene.2023.109963.

[11] C. Iwendi, A. K. Bashir, A. Peshkar, R. Sujatha, J. M. Chatterjee, S. Pasupuleti, et al., "COVID-19 patient health prediction using boosted random forest algorithm," *Frontiers in public health*, vol. 8, p. 357, May 2020, https://doi.org/10.3389/fpubh.2020.00357.

[12] V. Singh, R. C. Poonia, S. Kumar, P. Dass, P. Agarwal, V. Bhatnagar, et al., "Prediction of COVID-19 corona virus pandemic based on time series data using Support Vector Machine," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 23, no. 8, pp. 1583-1597, Dec. 2020, https://doi.org/10.1080/09720529.2020.1784535.

[13] S. H. Yoo, H. Geng, T. L. Chiu, S. K. Yu, D. C. Cho, J. Heo, et al., "Deep learning-based decision-tree classifier for COVID-19 diagnosis from chest X-ray imaging," *Frontiers in medicine*, vol. 7, p. 427, 2020, https://doi.org/10.3389/fmed.2020.00427.

[14] W. M. Shaban, A. H. Rabie, A. I. Saleh, and M. A. Abo-Elsoud, "A new COVID-19 Patients Detection Strategy (CPDS) based on hybrid feature selection and enhanced KNN classifier," *Knowledge-Based Systems*, vol. 205, p. 106270, 2020, https://doi.org/10.1016/j.knosys.2020.106270.

[15] C. Zhou, J. Song, S. Zhou, Z. Zhang, and J. Xing, "COVID-19 detection based on image regrouping and ResNet-SVM using chest X-ray images," *IEEE Access*, vol. 9, pp. 81902-81912, Jun. 2021, https://doi.org/10.1109/ACCESS.2021.3086229.

[16] A. Akhtar, S. Akhtar, B. Bakhtawar, A. A. Kashif, N. Aziz, and M. S. Javeid, "COVID-19 detection from CBC using machine learning techniques," *International Journal of Technology, Innovation and Management (IJTIM)*, vol. 1, no. 2, pp. 65-78, 2021, https://doi.org/10.54489/ijtim.v1i2.22.

[17] H. Arslan and H. Arslan, "A new COVID-19 detection method from human genome sequences using CpG island features and KNN classifier," *Engineering Science and Technology, an International Journal*, vol. 24, no. 4, pp. 839-847, Oct. 2021, https://doi.org/10.1016/j.jestch.2020.12.026.

[18] W. M. Shaban, A. H. Rabie, A. I. Saleh, and M. A. Abo-Elsoud, "Accurate detection of COVID-19 patients based on distance biased Naïve Bayes (DBNB) classification strategy," *Pattern Recognition*, vol. 119, p. 108110, Sep. 2021, https://doi.org/10.1016/j.patcog.2021.108110.

[19] V. A. D. F. Barbosa, J. C. Gomes, M. A. de Santana, C. L. de Lima, R. B. Calado, C. R. Bertoldo Junior, et al., "Covid-19 rapid test by combining a random forest-based web system and blood tests," *Journal of Biomolecular Structure and Dynamics*, vol. 40, no. 22, pp. 11948-11967, Dec. 2022, https://doi.org/10.1080/07391102.2021.1966509.

[20] N. A. Mansour, A. I. Saleh, M. Badawy, and H. A. Ali, "Accurate detection of Covid-19 patients based on Feature Correlated Naïve Bayes (FCNB) classification strategy," *Journal of ambient intelligence and humanized computing*, pp. 1-33, 2022, https://doi.org/10.1007/s12652-020-02883-2.

[21] F. Manzella, G. Pagliarini, G. Sciavicco, and I. E. Stan, "The voice of COVID-19: Breath and cough recording classification with temporal decision trees and random forests," *Artificial Intelligence in Medicine*, vol. 137, p. 102486, Mar. 2023, https://doi.org/10.1016/j.artmed.2022.102486.

[22] Z. Albataineh, F. Aldrweesh, and M. A. Alzubaidi, "COVID-19 CT-images diagnosis and severity assessment using machine learning algorithm," *Cluster Computing*, vol. 27, no. 1, pp. 547-562, Feb. 2024, https://doi.org/10.1007/s10586-023-03972-5.

[۲۳] اکرمی ع.، پارسامنش م.، بررسی یک مدل اپیدمیک فازی ریاضی برای انتشار ویروس کرونا در یک جمعیت، *محاسبات نرم*، ۱۱ (۱)، صص ۲-۹، ۱۴۰۱، https://doi.org/10.22052/scj.2022.246053.1045 .

[۲۴] یداللهی ا. ح.، صباغیان بیگدلی ح.، ارائه مدلی برای شبیه‌سازی انتشار ویروس کووید-۱۹ بر اساس زنجیره مارکوف گسسته زمان، *محاسبات نرم*، ۱۱ (۲)، صص ۸۸-۱۰۳، ۱۴۰۱، https://doi.org/10.22052/scj.2023.246527.1076 .

[۲۵] سرچاهی م.، مهدی‌پور ا.، ارائه راهکاری بر پایه یادگیری عمیق ترکیبی جهت بهبود دقت شناسایی تصاویر سی تی- اسکن ریه بیماران کووید-۱۹، *محاسبات نرم*، ()، ۱۴۰۲، https://doi.org/10.22052/scj.2023.253142.1158.

[26] Kaggle, "COVID-19 dataset," Available: https://www.kaggle.com/datasets/meirnizri/covid19-dataset, Accessed on: Jan. 2025.