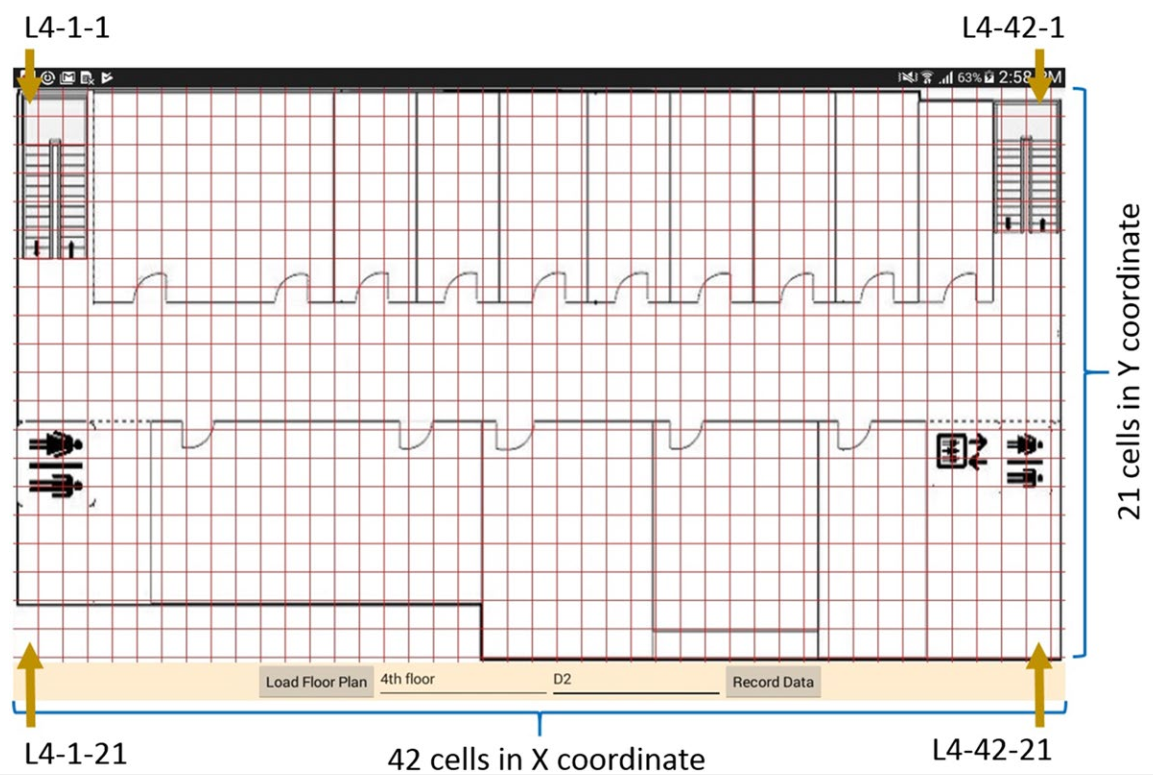
1. شبیه‌سازی و ارزیابی نتایج
   1. مقدمه

ابتدا مجموعه‌دادگان موقعیت‌یابی مبتنی بر RSS که اطلاعات اساسی برای ارزیابی الگوریتم‌ها و روش‌های موقعیت‌یابی هستند، معرفی می‌شود. سپس، به بررسی نتایج شبیه‌سازی روش‌های آموزش مدل موقعیت‌یاب با استفاده از الگوریتم‌های یادگیری گروهی که راهکار اصلی در بهبود دقت موقعیت‌یابی هستند پرداخته می‌شود. پس از آن، پیاده‌سازی دو روش پیشنهادی برای بهبود مدل یادگیری گروهی؛ یعنی روش‌های کاهش ابعاد و الگوریتم ژنتیک بررسی می‌شود. روش کاهش ابعاد برای کاهش پیچیدگی محاسباتی و بهبود کارایی الگوریتم‌های موقعیت‌یاب ارزیابی می‌شود و روش‌های تنظیم ابرپارامترها با استفاده از الگوریتم ژنتیک نیز به‌عنوان یک ابزار قدرتمند برای بهینه‌سازی پارامترها در این الگوریتم‌ها، بررسی می‌شود. در نهایت، با جمع‌بندی نتایج به‌دست‌آمده از تمامی این روش‌ها، عملکرد و کارایی هر یک از آن‌ها در تعیین موقعیت به‌صورت کامل ارزیابی می‌گردد و با نتایج مشابه مقایسه می‌شود.

* 1. دادگان موقعیت‌یابی مبتنی بر RSS

تعداد زیادی از مجموعه‌دادگان موقعیت‌یابی مبتنی بر RSS وجود دارد. در میان آن‌ها، چندین مجموعه‌دادگان به سبب ویژگی‌ها و اطلاعات غنی‌تر، محبوب‌تر هستند و بیشتر مورداستفاده قرار می‌گیرند. از جمله معروف‌ترین مجموعه‌دادگان، مجموعه‌دادگان UJIndoorLoc که در سال 2014 میلادی معرفی شد برای شبکه غیر بی سیم Wi-Fi معرفی شد [1]. گرچه این مجموعه دادگان به عنوان معیار سنجش الگوریتم‌های موقعیت‌یاب به کار می‌رود، اما شامل شرایط مختلف محیطی نمی‌شود. مجموعه دادگان IPIN در سال 2016 تا 2022 میلادی برای مسابقات ردیابی ربات‌ها معرفی شدند که عمده اطلاعات آن مربوط به سنسورهای ربات است که مبتنی بر موقعیت‌یابی آفلاین است [2].

دادگان استفاده شده در این پژوهش، مجموعه‌دادگان JUIndoorLoc است که در سال ۲۰۱۹ میلادی ارائه شده است [3] و به‌خاطر ضبط داده در شرایط مختلف محیطی، اعتبار صحت این مجموعه دادگان در طول زمان بیشتر است. داده‌ها از طبقه سوم، چهارم و پنجم ساختمانی در دانشگاه جوادپور گردآوری شده‌اند. در این ساختمان، هر طبقه ۸۸۲ متر مربع به طول ۴۲ متر و عرض ۲۱ متر را پوشش می‌دهد و از اتاق‌های اساتید، کلاس‌های درس، اتاق‌های سمینار، آزمایشگاه‌های تحقیقاتی و راهرو تشکیل شده است. این ناحیه به سلول‌های ۱ متر × ۱ متر تقسیم می‌شود. بنابراین، مختصات و هر طبقه به ترتیب به 42 و 21 سلول تقسیم می‌شود که در شکل 2 نشان داده شده است.



نمایشی از تقسیم‌بندی اثر انگشت طبقه چهارم محیط مجموعه‌دادگان استفاده شده [3]

گردآوری این مجموعه‌دادگان، با استفاده از نرم‌افزار WDC که بر روی دستگاه‌های اندرویدی نصب می‌شود، انجام شده است و برای هر بلوک از موقعیت مقادیر RSSI به مدت ۱۲۰ ثانیه ضبط می‌شود و در پایگاه داده سیستم موقعیت یاب ذخیره می‌شود. با توجه به مشاهدات، مقادیر RSSI با سخت‌افزار‌های مختلف و شرایط مختلف محیطی متفاوت است. لذا این مجموه دادگان برای استحکام الگوریتم‌ها در شرایط گوناگون جمع شده است. از جمله آن، جمع‌آوری داده از نقاط مختلف یک بلوک به جای جمع‌آوری از مرکز بلوک است. همچنین، داده‌ها در زمان‌های مختلف روز جمع‌آوری شده‌اند تا ماهیت‌های مختلف را به دلیل اثرات محیطی مختلف درک کنند. به علاوه در هنگام جمع‌آوری داده‌ها، ناهمگونی‌های زمینه‌ای مختلف مانند بسته یا باز بودن درب اتاق‌ها و حضور و عدم حضور انسان‌ها در نظر گرفته شده است. برای درنظرگیری سخت‌افزار‌های مختلف نیز از چهار دستگاه اندرویدی متفاوت استفاده شده است.

مجموعه‌دادگان شامل 25364 نمونه است و هر نمونه دارای 177 ویژگی است که شامل 172 مقدار RSSI از نقاط دسترسی مختلف ، شناسه منحصر به فرد هر بلوک ، باز و بسته بودن درب اتاق، حضور یا عدم حضور انسان، شناسه دستگاه ثبت کننده و زمان جمع‌آوری داده برحسب میلی ثانیه است.

* + 1. پیش‌پردازش دادگان موقعیت‌یابی

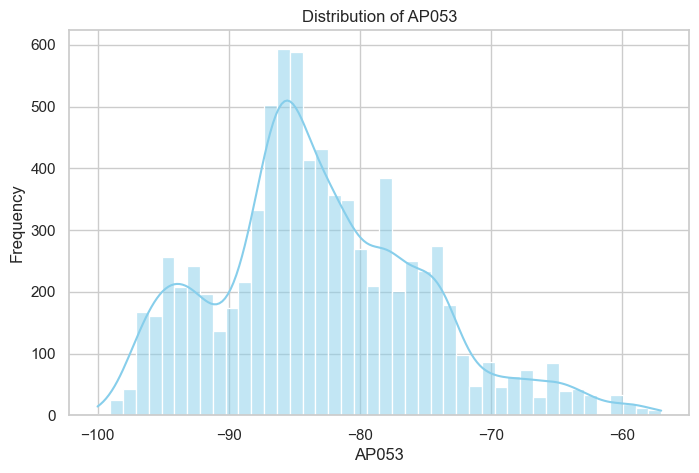
پیش‌پردازش دادگان، مجموعه اقداماتی است که پیش از استفاده از دادگان موقعیت‌یابی اجرا می‌شود. این اقدامات در راستای پاک‌سازی و تقویت داده‌ها صورت می‌گیرد تا دقت و قابلیت استفاده از آن‌ها در مراحل بعدی افزایش یابد. اقدامات انجام شده به شرح زیر است:

* پاک‌سازی داده

این مرحله شامل حذف مقادیر تکراری و پاک‌سازی مقادیر گمشده است. باتوجه‌به نتایج برآمده از پردازش مجموعه‌دادگان، مقادیر تکراری در آن وجود ندارد. همچنین در هر نمونه، مقادیر RSSI از نقاط دسترسی خارج از محدوده برابر با جای‌گذاری شده است.

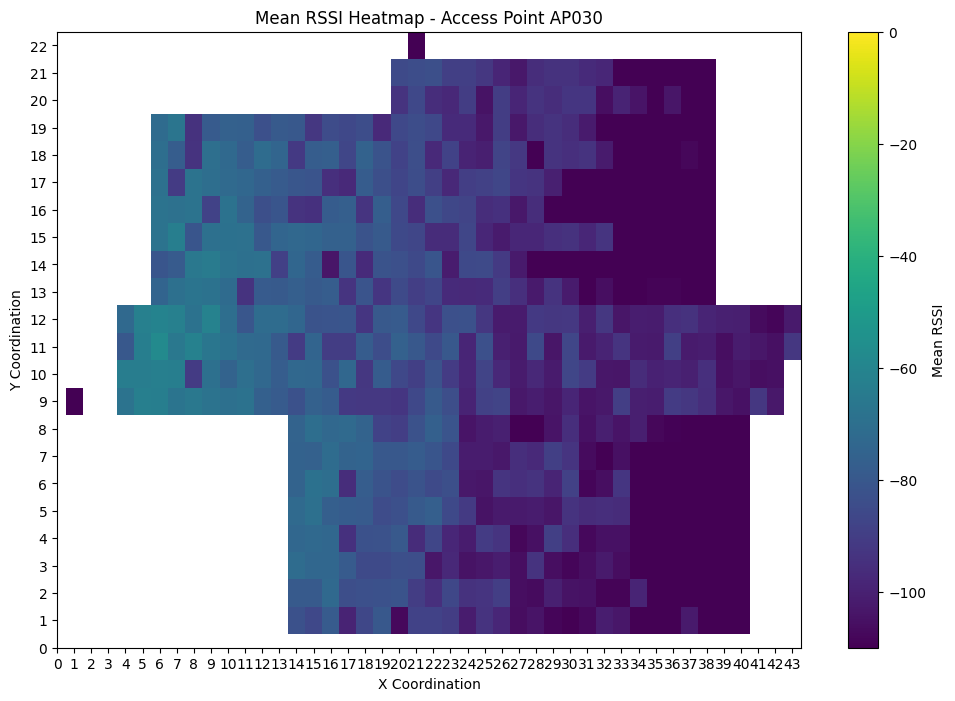
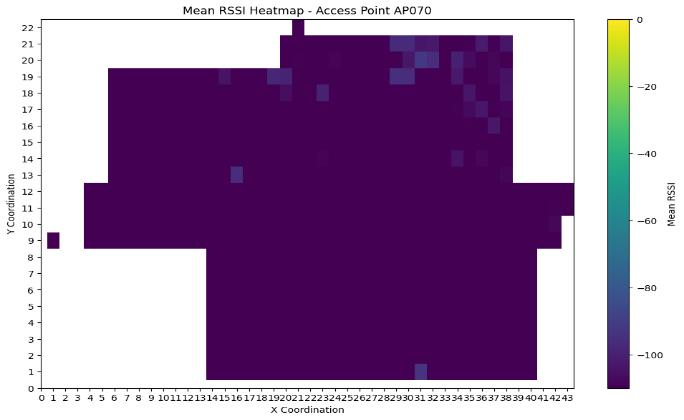
* *تجزیه‌وتحلیل اکتشافی دادگان[[1]](#footnote-2) (*EDA*)*

این مرحله از پیش‌پردازش برای بررسی توزیع هر ویژگی انجام می‌شود. به‌عنوان‌مثال، باتوجه‌به ‏شکل (4˗2) که تعداد دفعات تکرار هر مقدار RSSI را برای هر نقطه دسترسی نشان می‌دهد، نقطه دسترسی AP053 نسبت به نقطه دسترسی AP133، از اطلاعات بیشتری برخوردار است. این تجزیه‌وتحلیل، ایده استفاده از دو الگوریتم پیشنهادی در جهت بهبود عملکرد مدل‌های یادگیری گروهی را تأیید می‌کند.

نمودار هیستوگرام تعداد دفعات تکرار هر مقدار RSSI در نقاط دسترسی AP053 و AP133

علاوه‌برآن، با میانگین‌گیری از مقادیر RSSI در هر بلوک، نقشه‌ای از توزیع مقادیر RSSI در مختصات و حاصل می‌گردد که میزان اطلاعات دریافتی از هر نقطه دسترسی را به طور شهوری مصور می‌سازد. نمونه ای از این این نقشه در ‏شکل (4˗3) نشان داده شده است.

نقشه توزیع مقادیر RSSI در مختصات و از نقاط دسترسی AP030 و AP070 در طبقه چهارم

* نرمال‌سازی

نرمال‌سازی داده‌ها یک فرایند مهم در پیش‌پردازش داده‌ها است که به مقیاس‌بندی داده‌ها به محدوده‌ای مشخص و یکنواخت اشاره دارد. در این فرایند مقادیر هر ویژگی از مجموعه‌دادگان به یک بازه معین (معمولاً بین 0 و 1)، با هدف داشتن مقیاس مشابه برای همه ویژگی‌ها تبدیل می‌شود.

نرمال‌سازی باعث افزایش سرعت و تعمیم‌پذیری مدل می‌شود. همچنین هنگامی که مقادیر ویژگی‌ها دارای مقیاس‌های متفاوت هستند، الگوریتم‌های یادگیری گروهی ممکن است به ویژگی‌هایی با مقیاس بزرگ‌تر بیش‌برازش کنند. با نرمال‌سازی داده‌ها، این احتمال کاهش می‌یابد و عملکرد مدل در داده‌های جدید بهبود می‌یابد.

* 1. آموزش موقعیت‌یاب با الگوریتم‌های یادگیری گروهی

مدل های یادگیری گروهی معرفی شده در فصل 3 با مجموعه‌دادگان معرفی شده در بخش 4-2 آموزش داده می‌شود. برای مقایسه عملکرد مدل‌ها در شرایط یکسان، می‌بایست تمام پردازش‌ها در یک سیستم سخت‌افزاری واحد انجام شود. مشخصات این سیستم سخت‌افزاری در ‏جدول (4˗1) ذکر شده است. مطابق آن، سیستم سخت‌افزاری مورداستفاده، یک سیستم پایین رده برای محاسبات موقعیت محسوب می‌شود.

همچنین در پیاده‌سازی الگوریتم‌های یادگیری گروهی، از به‌کارگیری الگوریتم‌های جاسازی شده برای توقف

مشخصات سیستم سخت‌افزاری موقعیت‌یاب برای اجرای شبیه‌سازی

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CPU** | **Memory** | **Disk** | **OS** | **GPU** |
| Intel Core i5-4670 @3.4GHz | 16.0 GB DDR3 | HDD | Windows 11 | × |

زودهنگام الگوریتم در روش های تکرار شونده ممانعت به عمل آمده است تا مدل‌های خروجی قابل مقایسه باشد. برای هر الگوریتم یادگیرنده از درخت با حداکثر عمق به عنوان یادگیرنده پایه استفاده شده است.

برای ارزیابی بهتر، از روش اعتبارسنجی متقابل طبقه‌بندی شده 5 گانه[[2]](#footnote-3) استفاده می‌شود. در این روش ۲۰ درصد از دادگان به‌عنوان دادگان اعتبارسنجی، با درنظرگیری نسبت برچسب‌ها انتخاب می‌گردد و مابقی به‌عنوان آموزش به مدل داده می‌شود. در نهایت پس از ۵ تکرار آموزش مدل، از دقت تخمین دادگان اعتبارسنجی میانگین‌گیری می‌شود.

در ابتدا، یک درخت تصمیم به‌عنوان یادگیرنده پایه با چهار مقدار حداکثر عمق متفاوت آموزش داده شده است و نتایج ارزیابی آن برای 20 درصد از مجموعه‌دادگان در ‏جدول (4˗2) عنوان شده است. اشاره شد که ابعاد هر بلوک 1 متر در 1 متر است؛ لذا در محاسبات میانگین خطا مطابق رابطه می‌توان به‌سادگی و مستقیم مختصات و را جای‌گذاری کرد. همچنین زمان آزمایش برای 5072 داده محاسبه شده است.

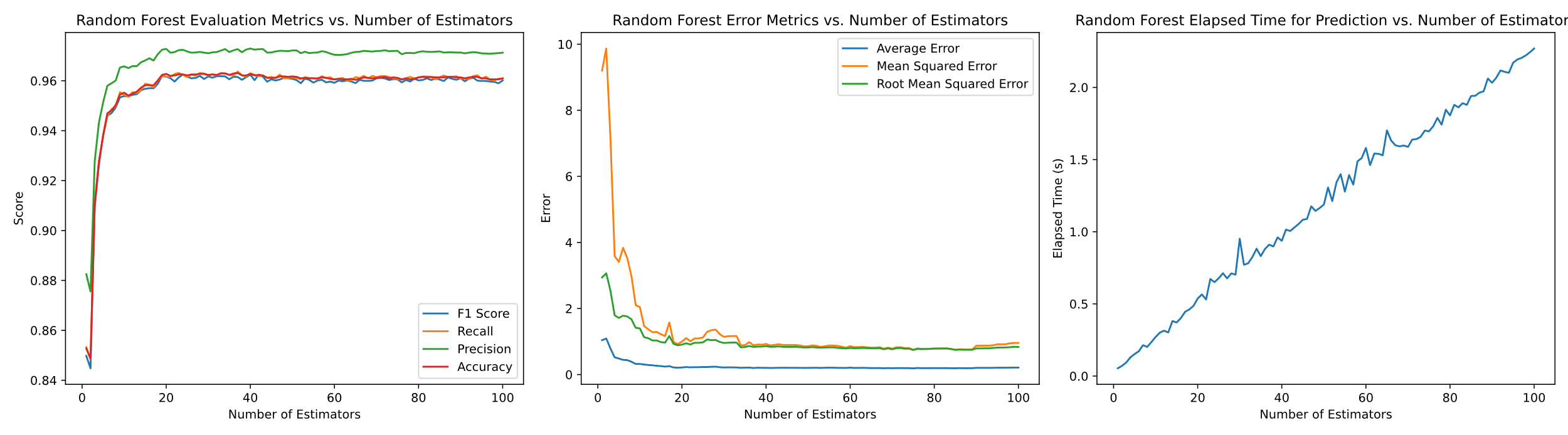
نتایج شبیه‌سازی درخت تصمیم با دادگان ارزیابی و حداکثر عمق‌های متفاوت

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| حداکثر عمق  درخت تصمیم |  |  |  |  | (متر) | (مترمربع) | (متر) | زمان آزمایش |
| 10 | 8.21% | 12.21% | 8.22% | 7.97% | 15.58 | 362.46 | 17.98 | 0.054 ثانیه |
| 50 | 82.50% | 85.73% | 82.49% | 83.10% | 1.06 | 9.85 | 3.11 | 0.056 ثانیه |
| 100 | 85.02% | 87.94% | 85.27% | 84.85% | 0.93 | 7.48 | 2.69 | 0.057 ثانیه |
| 176 | 85.02% | 88.80% | 85.01% | 84.84% | 0.093 | 7.47 | 2.69 | 0.058 ثانیه |

در ادامه با توجه به نتایج حاصل از یادگیرنده پایه، مدل‌های یادگیری گروهی مورد ارزیابی قرار می‌گیرد.

* + 1. مدل موقعیت‌یاب الگوریتم جنگل تصادفی

همان‌طور که در مطرح شد، الگوریتم جنگل تصادفی به سه پارامتر )تعداد ویژگی‌های هر گره(، (تعداد درختان در جنگل) و اندازه درخت نیاز دارد. برابر با نیمی از تعداد ویژگی‌های کل (176 ویژگی) درنظر گرفته شده است. اندازه درخت برابر با حداکثر عمق درخت تعیین شده است که در این بخش مقدار آن محدود نشده است. نتایج شبیه‌سازی با تعداد متفاوت درختان () در ‏شکل (4˗4) نمایش داده شده است. باتوجه‌به‌ آن، با افزایش تعداد درختان در جنگل، زمان آزمایش افزایش می‌یابد. همچنین مشاهده می‌شود که خطا و معیارهای سنجش، پس از تعداد مشخصی از درختان، با افزایش آن تغییر چندانی نمی‌کند. در ‏جدول (4˗3) نیز مقدار دقیق متغیرها برحسب سه مقدار از تعداد درختان ذکر شده است.



نمودار نتیجه ارزیابی، خطا و زمان آزمایش جنگل تصادفی برای تعداد متفاوت درختان و حداکثر عمق ثابت



نمودار نتیجه ارزیابی، خطا و زمان آزمایش جنگل تصادفی برای عمق متفاوت درختان و تعداد 40 درخت

برای بررسی تأثیر عمق درختان بر معیارهای ارزیابی و زمان، تعداد درختان در مقدار 40 نگه داشته شده است و عمق درختان از صفر تا حداکثر مقدار تغییر داده شده است. نتایج ارزیابی در ‏شکل (4˗5) نمایش داده شده است. مطابق آن پس از یک عمق مشخص، دقت و خطا تغییر چندانی ندارد. دلیل این مسئله برآورده‌شدن شرایط توقف در ساخت درخت پس از تعداد معینی از گره‌ها است. این موضوع در نمودار زمان آزمایش برحسب حداکثر عمق و ‏جدول (4˗4) نیز تأیید می‌شود.

نتایج شبیه‌سازی جنگل تصادفی و تعداد متفاوت درختان و عمق ثابت درختان (حداکثر)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| تعداد درختان |  |  |  |  | (متر) | (مترمربع) | (متر) | زمان آزمایش |
| 10 | 85.38% | 88.26% | 85.66% | 89.17% | 0.91 | 8.43 | 2.38 | 0.082 ثانیه |
| 50 | 95.80% | 97.19% | 95.80% | 95.84% | 0.19 | 1.13 | 0.99 | 1.17 ثانیه |
| 100 | 95.76% | 97.17% | 95.76% | 95.81% | 0.61 | 15.26 | 3.90 | 3.00 ثانیه |

نتایج شبیه‌سازی جنگل تصادفی برای عمق متفاوت درختان و تعداد 40 درخت

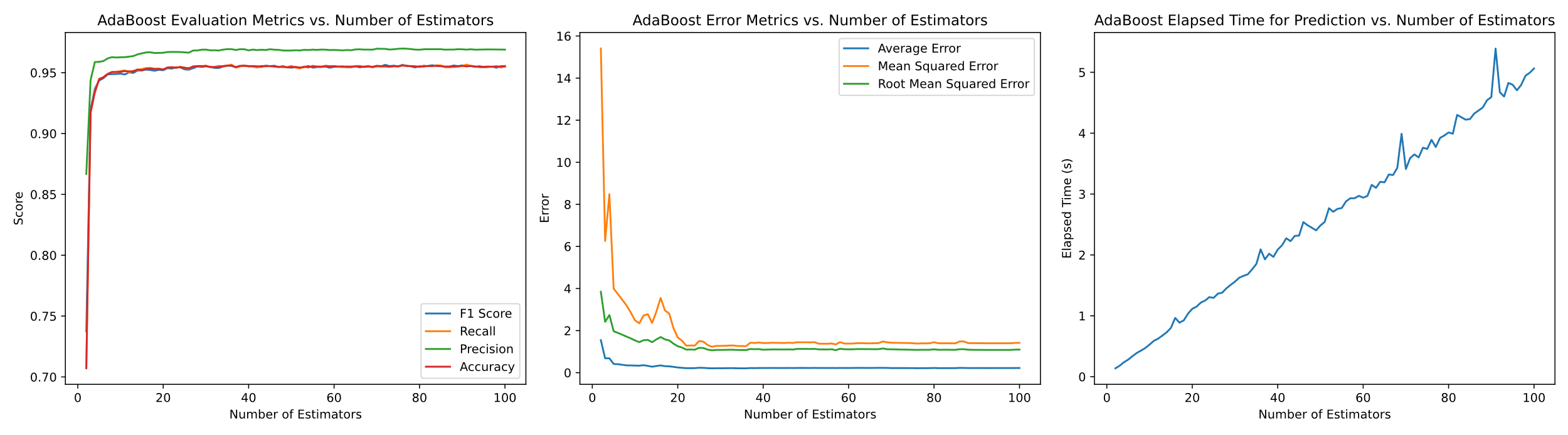
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| حداکثر عمق درختان |  |  |  |  | (متر) | (مترمربع) | (متر) | زمان آزمایش |
| 10 | 50.99% | 61.46% | 60.00% | 50.70% | 9.58 | 228.90 | 11.59 | 0.80 ثانیه |
| 50 | 95.68% | 97.04% | 95.63% | 95.69% | 0.21 | 1.43 | 1.12 | 0.86 ثانیه |
| 100 | 95.68% | 97.04% | 95.63% | 95.69% | 0.21 | 1.43 | 1.12 | 0.87 ثانیه |

با درنظرگرفتن دو پارامتر آزاد از نظر محاسباتی، امکان ساخت جنگل تصادفی برای تمام حالات ممکن بسیار گران است. پیداکردن مقدار بهینه این پارامتر در بخش با استفاده از الگوریتم ژنتیک توضیح داده می‌شود.

* + 1. مدل موقعیت‌یاب الگوریتم AdaBoost

در الگوریتم 2 که برای روش AdaBoost ارائه شده است، تنها از دو پارامتر (فرضیه یا یادگیرنده پایه) و (تعداد یادگیرندگان پایه) به عنوان ابرپارامتر تنظیم مدل یاد شده است. از آن جا که در این پژوهش از درخت تصمیم به‌عنوان یادگیرنده پایه استفاده شده است. پارامتر ، به (حداکثر عمق درخت) اشاره دارد. در این بخش همچنان نتایج ارزیابی در دو حالت تعداد یادگیرندگان متغیر/ثابت و حداکثر عمق متفاوت ثابت/متغیر

بحث می‌گردد. در حالت اول، حداکثر عمق درخت برابر با بیشینه مقدار ممکن آن؛ یعنی 176 گره، در نظر گرفته می‌شود و تعداد یادگیرندگان از صفر تا 100 تغییر می‌کند. این نتایج در ‏شکل (4˗6) رسم شده است. در اینجا نیز مشابه جنگل تصادفی با افزایش تعداد یادگیرندگان، زمان سپری شده برای دادگان آزمایش به صورت خطی افزایش می‌یابد. همچنین در مقایسه با جنگل تصادفی دقت، با تعداد کمتری از یادگیرندگان به مقدار بیشینه خود همگرا می‌شود. مطابق با ‏جدول (4˗5) زمان سپری شده برای دادگان آزمایش، با تعداد یکسانی از درختان با عمق برابر، در الگوریتم AdaBoost نسبت به الگوریتم جنگل تصادفی افزایش داشته است. این در حالی است که دقت با معیارهای گوناگون در تعداد 50 و 100 درخت تغییری نداشته است.

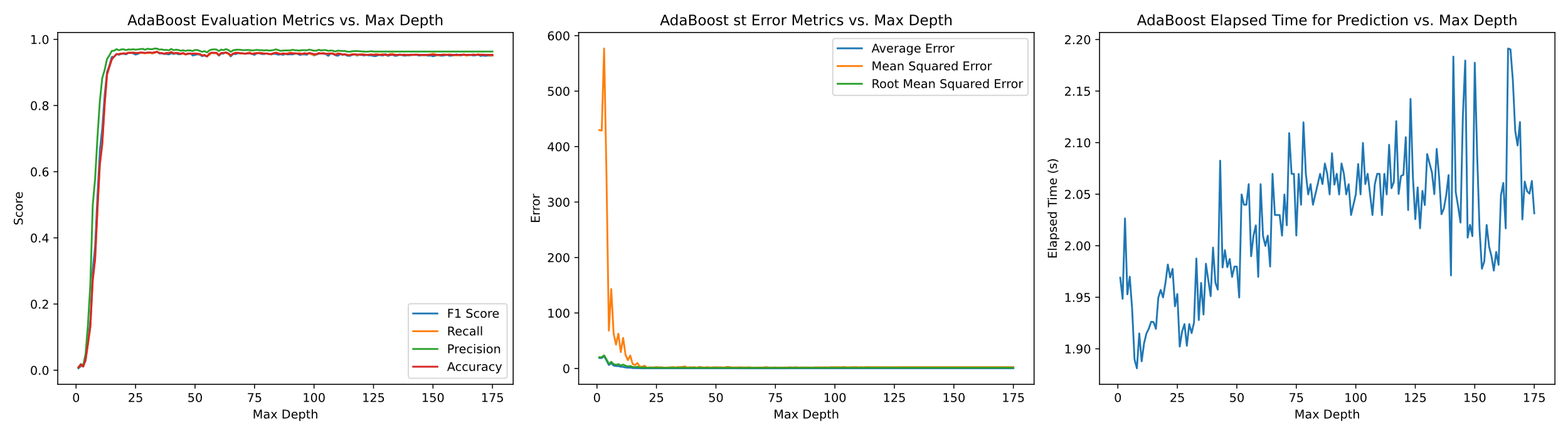


نمودار نتیجه ارزیابی، خطا و زمان آزمایش AdaBoost برای تعداد متفاوت درختان و حداکثر عمق ثابت

نتایج شبیه‌سازی AdaBoost با تعداد متفاوت درختان و عمق ثابت درختان (حداکثر)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| تعداد درختان |  |  |  |  | (متر) | (مترمربع) | (متر) | زمان آزمایش |
| 10 | 91.76% | 94.38% | 91.77% | 92.02% | 0.68 | 6.25 | 2.41 | 0.018 ثانیه |
| 50 | 95.45% | 96.82% | 95.46% | 95.45% | 0.22 | 1.44 | 1.13 | 2.48 ثانیه |
| 100 | 95.51% | 96.89% | 95.50% | 95.48% | 0.22 | 1.41 | 1.09 | 5.06 ثانیه |

در حالت دوم، تعداد درختان ثابت و برابر با 40 درنظر گرفته شده است و حداکثر عمق درخت تغییر داده می‌شود. نتایج تغییر حداکثر عمق درحت در روش AdaBoost در ‏شکل (4˗7) نمایش داده شده است. افزایش عمق درختان اثر کمتری نسبت به تعداد یادگیرندگان در زمان آزمایش دارد. همچنین، نسبت به روش جنگل تصادفی، این روش با درختان کم عمق‌تری به مقدار بیشینه خود همگرا شده است.



نمودار نتیجه ارزیابی، خطا و زمان آزمایش AdaBoost برای عمق متفاوت درختان و تعداد 40 درخت

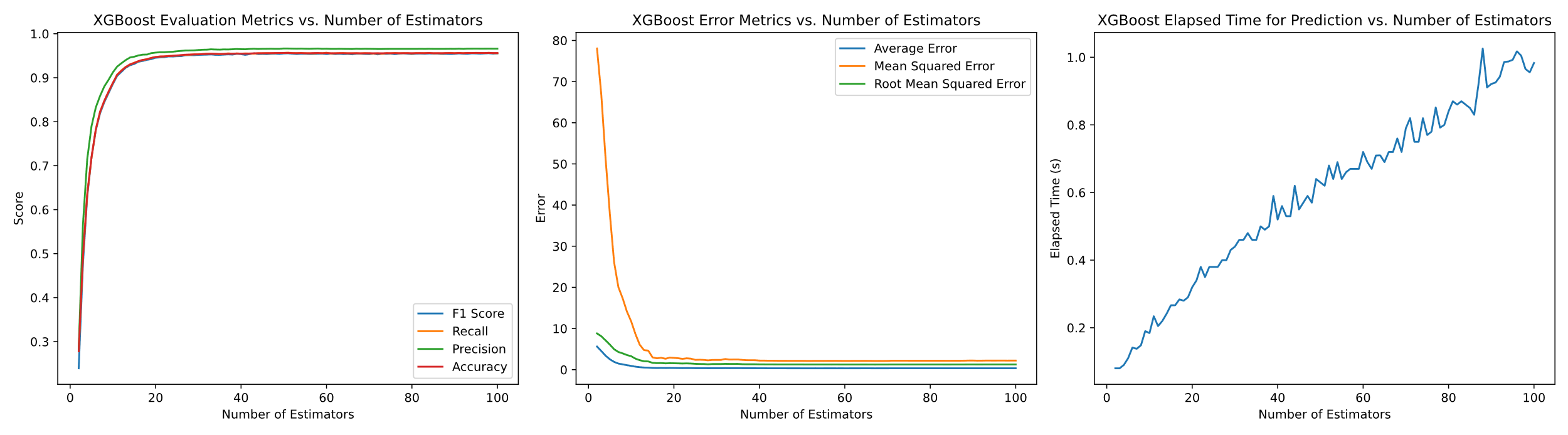
در ‏جدول (4˗6) نیز مشخص است که الگوریتم AdaBoost نسبت به جنگل تصادفی، مدت زمان بیشتری را برای تخمین دادگان آزمایش محاسبه می‌کند و در این تعداد مشخص از درختان دقت در معیار فاصله اندکی کمتر شده است.

نتایج شبیه‌سازی AdaBoost برای عمق متفاوت درختان و تعداد 40 درخت

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| حداکثر عمق درختان |  |  |  |  | (متر) | (مترمربع) | (متر) | زمان آزمایش |
| 10 | 62.31% | 0.81% | 62.29% | 66.24% | 3.21 | 29.43 | 5.38 | 1.88 ثانیه |
| 50 | 95.66% | 96.51% | 95.63% | 95.56% | 0.36 | 2.28 | 1.29 | 1.97 ثانیه |
| 100 | 95.52% | 96.56% | 95.76% | 95.34% | 0.36 | 2.26 | 1.24 | 2.04 ثانیه |

* + 1. مدل موقعیت‌یاب الگوریتم XGBoost

برخلاف دو الگوریتم قبلی، XGBoost به ابرپارامترهای بیشتری برای تنظیم نیاز دارد. (نرخ یادگیری) به‌عنوان ضریبی برای کاهش اندازه گام، برابر با مقدار 0.3 درنظر گرفته شده است تا از بیش‌برازش جلوگیری کند. (ضریب کاهش اولیه) برابر با صفر و (ضریب تنظیم برگ) نیز برابر با یک تنظیم شده است. این مقادیر، مقادیر پیش‌فرضی است که در بسیاری از پژوهش‌های مشابه استفاده شده است و نتایج مطلوبی از آن حاصل شده است [4]. (تعداد تکرارهای Boosting) و همچنان حداکثر عمق درخت به‌عنوان ابر پارامتر آزاد برای تنظیم در نظر گرفته می‌شود و نتایج ارزیابی بررسی می‌گردد.



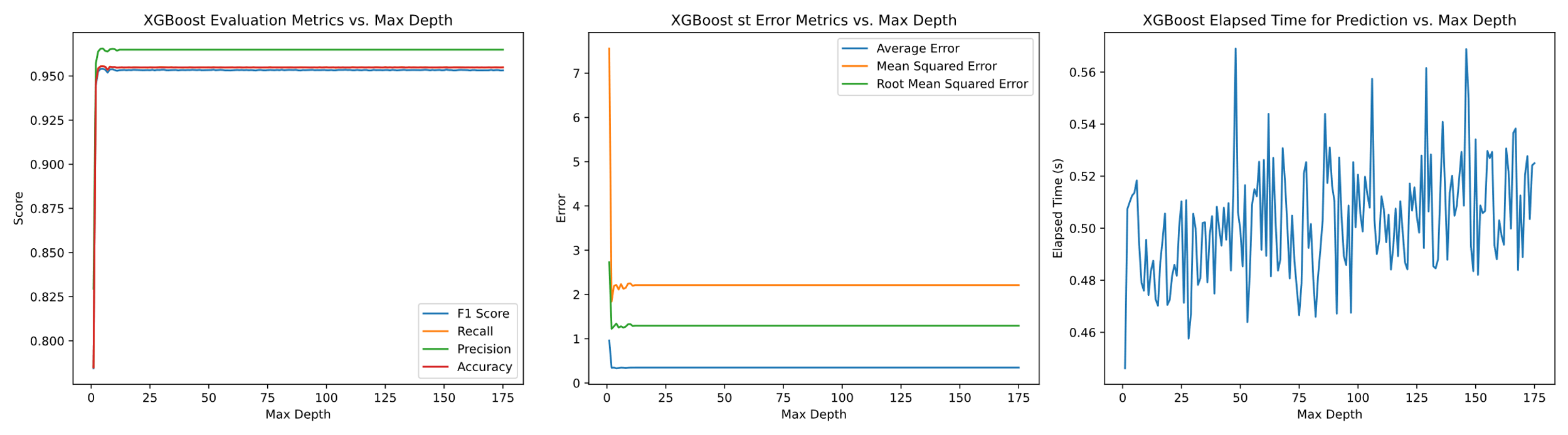
نمودار نتیجه ارزیابی، خطا و زمان آزمایش XGBoost برای تعداد متفاوت درختان و حداکثر عمق ثابت

همانطور که در ‏شکل (4˗8) مشاهده می‌شود، در این روش نیز‌ زمان آزمایش با تعداد یادگیرندگان رابطه خطی دارد. همچنین بیشینه مقدار دقت، با تعداد بیشتری از درختان نسبت به روش AdaBoost همگرا می‌شود؛ اما مطابق با ‏جدول (4˗7) زمان آزمایش روش XGBoost نسبت به روش AdaBoost بسیار کم‌تر است.

نتایج شبیه‌سازی XGBoost با تعداد متفاوت درختان و عمق ثابت درختان (حداکثر)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| تعداد درختان |  |  |  |  | (متر) | (مترمربع) | (متر) | زمان آزمایش |
| 10 | 88.80% | 91.21% | 88.34% | 88.58% | 0.92 | 11.74 | 3.28 | 0.19 ثانیه |
| 50 | 95.64% | 96.64% | 95.46% | 95.50% | 0.33 | 2.16 | 1.27 | 0.63 ثانیه |
| 100 | 95.62% | 96.60% | 95.62% | 95.48% | 0.33 | 2.21 | 1.27 | 0.98 ثانیه |

با ثابت قرار دادن تعداد درختان در مقدار 40 و تغییر حداکثر عمق درختان، نتایج حاصل از آن در ‏شکل (4˗9) ترسیم شده است که مطابق آن روش XGBoost نسبت به دو روش قبلی AdaBoost و جنگل تصادفی در عمق کم‌تری از درختان به مقدار بیشینه خود همگرا می‌شود. همچنین زمان آزمایش، اثر کمتری از عمق درختان نسبت به تعداد درختان می‌پذیرد.



نمودار نتیجه ارزیابی، خطا و زمان آزمایش XGBoost برای عمق متفاوت درختان و تعداد 40 درخت

نتایج مندرج در ‏جدول (4˗8) نشان می‌دهد که الگوریتم XGBoost در عمق کمتر و با زمان کمتری نسبت به دو روش قبلی به دقت مشابه رسیده است.

نتایج شبیه‌سازی XGBoost برای عمق متفاوت درختان و تعداد 40 درخت

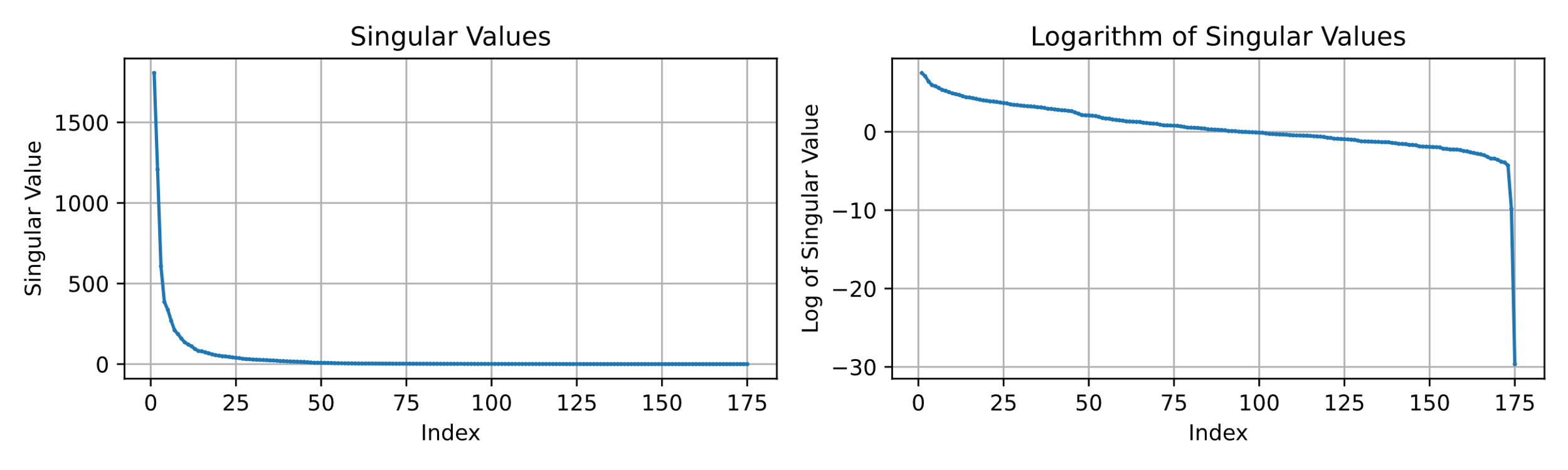
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| حداکثر عمق درختان |  |  |  |  | (متر) | (مترمربع) | (متر) | زمان آزمایش |
| 10 | 95.50% | 96.53% | 95.49% | 95.36% | 0.34 | 2.24 | 1.32 | 0.53 ثانیه |
| 50 | 95.48% | 96.49% | 95.48% | 95.33% | 0.34 | 2.21 | 1.29 | 0.52 ثانیه |
| 100 | 95.52% | 96.49% | 95.48% | 95.33% | 0.34 | 2.26 | 1.24 | 0.53 ثانیه |

در نتایج حاصل از پیاده سازی روش های یادگیری گروهی به کار رفته در این پژوهش، مشاهده می‌شود که چه میزان تنظیم پارامترهای مدل در دقت و زمان اثر گذار است. بدین منظور از دو روش کاهش ابعاد و الگوریتم ژنتیک برای یافتن مدل با دقت و زمان اجرای مطلوب و قابلیت پیاده سازی استفاده می‌شود. در ادامه نتایج استفاده از این دو الگوریتم بررسی می‌شود.

* 1. کاهش ابعاد دادگان موقعیت‌یابی

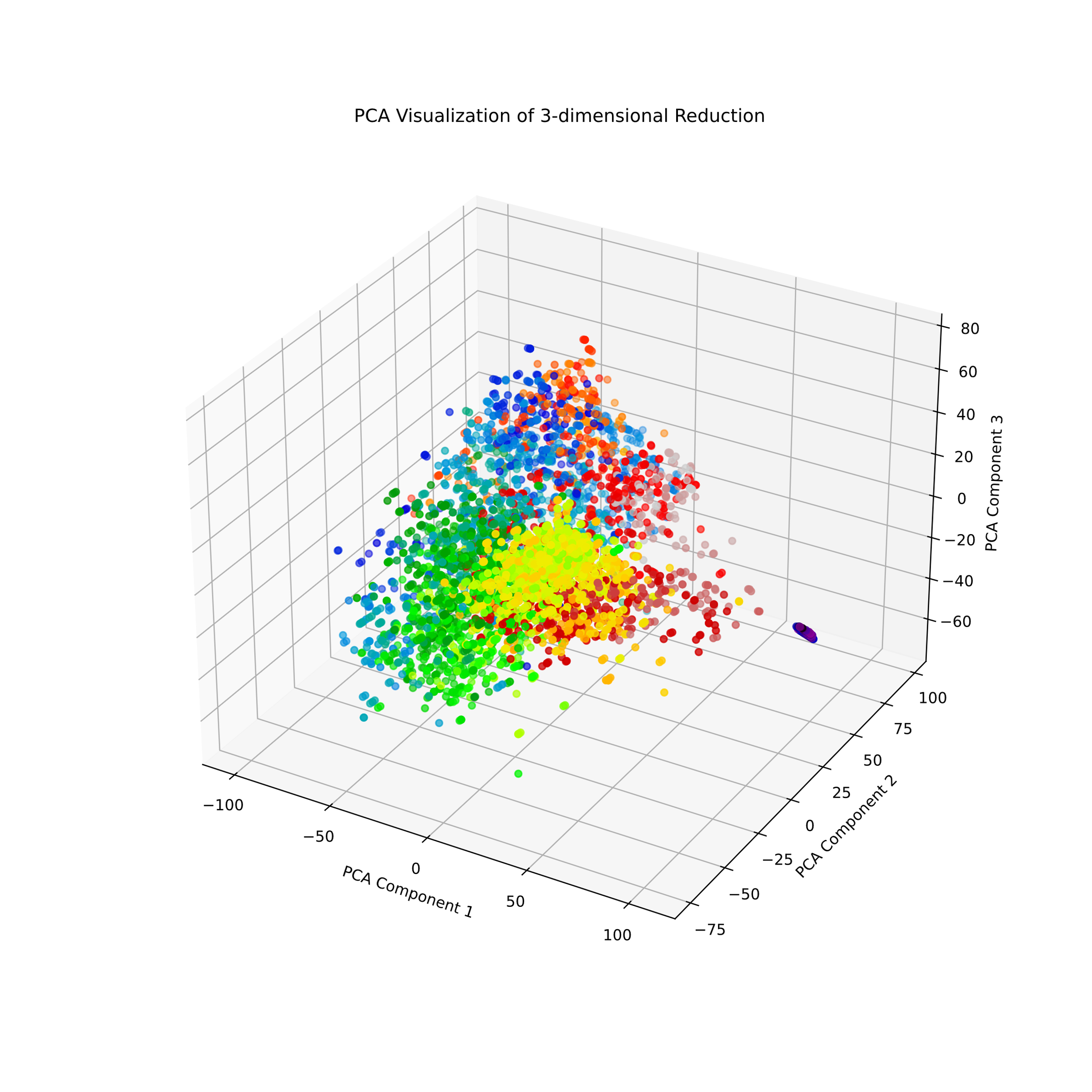
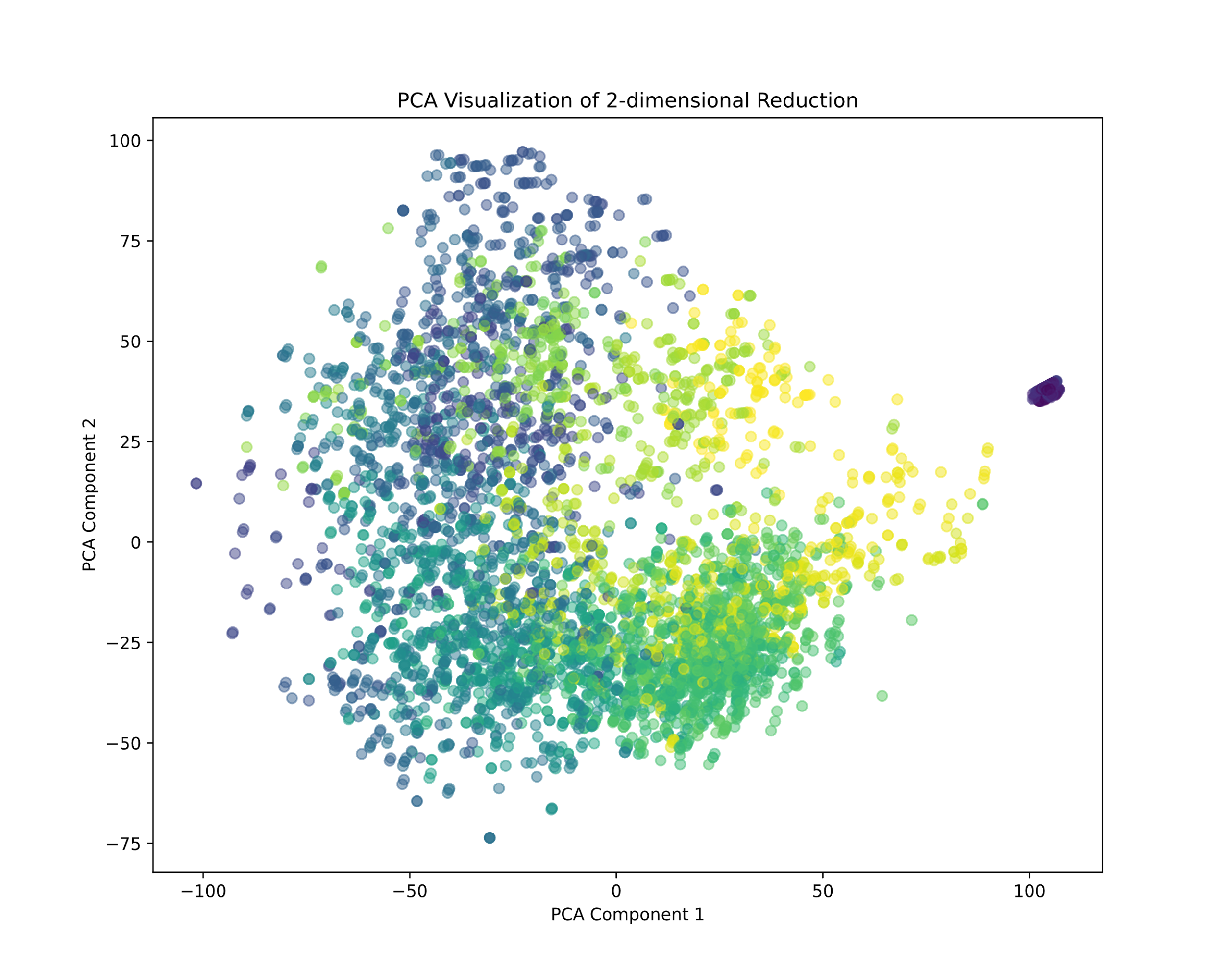
با توجه به تئوری بیان شده در کاهش ابعاد بردار ویژگی‌ها، به طور غیرمستقیم در روش‌های یادگیری گروهی با کاهش حداکثر عمق درختان اثرگذار است. از این رو، نتایج روش‌ معرفی شده برای کاهش ابعاد در این بخش بررسی می شود.

در روش تجزیه مؤلفه های اساسی (PCA) پس از استانداردسازی و محاسبه کوواریانس، با استفاده از تجزیه مقادیر منفرد (SVD) مقادیر قطر اصلی ماتریس به ترتیب بزرگ به کوچک مطابق با نمودار ‏شکل (4˗10) محاسبه شده است. بر اساس آن، تغییرات وسیع قبل از مرتبه 15 است، لذا می‌توان در محدوده این عدد، مرتبه کاهش را انتخاب نمود.



نمودار مقادیر منفرد ماتریس کوواریانس ویژگی‌ها

همچنین اشاره شد که روش PCA در تجسم داده می‌تواند مؤثر باشد. به همین منظور، همانطور که در ‏شکل (4˗11) مشخص است، نقاط همسایه (دارای رنگ مشابه) در فضای دو و سه بعدی از ویژگی‌ها نیز نزدیک به هم قرار گرفته اند. همچنین مشاهده می‌شود که نقاط داده در فضای سه بعدی نسبت به فضای دو بعدی قابلیت جداسازی بهتری دارند.

تجسم داده در فضای دو و سه بُعدی با استفاده از روش کاهش ابعاد PCA

با توجه به نتایج بخش قبلی یادگیری گروهی و امکان مقایسه با آن مرتبه کاهش 10 انتخاب شده است. ‏جدول (4˗9) نتایج به کار گیری روش کاهش ابعاد را در الگوریتم های یادگیری گروهی نشان می‌دهد. با مقایسه این جدول با جداول نتایج حداکثر عمق متفاوت بخش قبلی، در الگوریتم های جنگل تصادفی و AdaBoost افزایش دقت موقعیت یابی با حفظ زمان آزمایش مشهود است. در روش XGBoost، روش کاهش مرتبه PCA دقت و سرعت موقعیت‌یابی را کاهش داده است که نشان دهنده ناکارمدی کاهش ابعاد در الگوریتم XGBoost است.

نتایج روش PCA با کاهش مرتبه 10 در روش‌های یادگیری گروهی با 40 درخت

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| الگوریتم یادگیری |  |  |  |  | (متر) | (مترمربع) | (متر) | زمان آزمایش |
| جنگل تصادفی | 94.68% | 95.49% | 94.67% | 94.83% | 0.67 | 1.10 | 1.05 | 0.81 ثانیه |
| AdaBoost | 91.38% | 92.54% | 91.37% | 91.27% | 1.44 | 13.23 | 3.01 | 1.9 ثانیه |
| XGBoost | 88.30% | 89.43% | 88.29% | 88.09% | 1.62 | 13.62 | 3.12 | 0.72 ثانیه |

روش PCA گرچه با کاهش عمق درخت باعث بهبود نسبی در دقت و پیاده سازی برخی از الگوریتم های یادگیری می شود؛ اما با انتقال فضای ویژگی‌ها به فضای دیگر، تفسیرپذیری داده را کاهش می‌دهد. همچنین تنها تأثیر این روش بر عمق درختان است که در بخش قبل تأثیر کم عمق درخت در زمان آزمایش مشاهده شد. این روش، مرتبه کاهش را به عنوان ابرپارامتری که تنظیم آن ضروری است به مدل اضافه می‌کند.

* 1. تنظیم ابرپارامترها با استفاده از الگوریتم ژنتیک

در بخش قبل عنوان شد که روش PCA تنها می‌تواند بر کاهش عمق درختان اثرگذار باشد. گرچه این روش در پژوهش‌های موقعیت‌یابی مبتنی بر اثرانگشت مطرح است [5, 6] و به طورغیرمستقیم باعث بهبود نسبی دقت و سادگی در پیاده سازی شده است، اما با توجه به نتایج حاصل شده، این روش تأثیر چندانی در زمان سیستم یادگیری گروهی ندارد. در این بخش از روش الگوریتم ژنتیک در تنظیم ابرپارامترهای مدل یادگیری گروهی استفاده شده است. هدف، پیدا کردن یک مقدار بهینه برای متغیرهای آزاد مدل با درنظرگیری محدودیت‌های عملیاتی است تا مدل یادگیری گروهی، پاسخ بهتری را از نظر دقت و سرعت داشته باشد و همچنین در پیاده‌سازی مدل، منابع کمتری استفاده شود.

در الگوریتم ژنتیک از تابع برازش برای رتبه بندی وضعیت مسئله استفاده می‌شود. برای تعیین این تابع در این پژوهش، از ترکیب خطی دقت با معیار F1-Score، زمان آزمایش و حداکثر عمق درختان استفاده شده است. انتخاب معیار F1-Score به این دلیل است که این معیار علاوه بر درنظرگیری دقت طبقه بندی، حساسیت مدل را نیز درنظر می‌گیرد و مدل یادگیرنده را مقاوم می‌کند. در بخش ‏4˗3˗ مشاهده شد که زمان آزمایش با تعداد یادگیرندگان رابطه مستقیم دارد. از این رو با درنظرگیری زمان آزمایش در تابع برازش، مدل به سمت کاهش تعداد یادگیرندگان تشویق می‌شود. به کارگیری حداکثر عمق درختان در تابع برازش نیز باعث کاهش پیچیدگی و استفاده کمتر از منابع می‌شود. به این ترتیب تابع برازش به صورت زیر تعریف می شود:

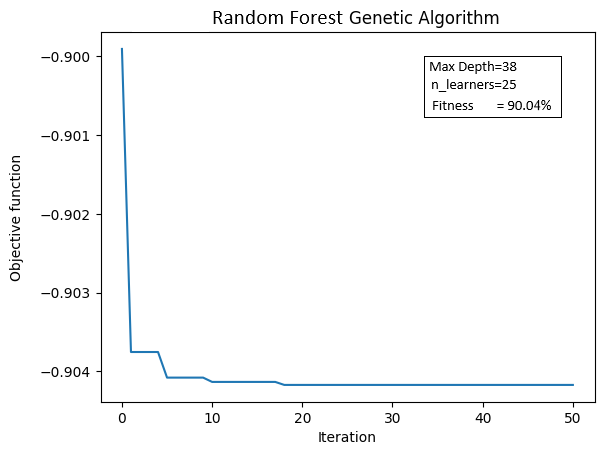
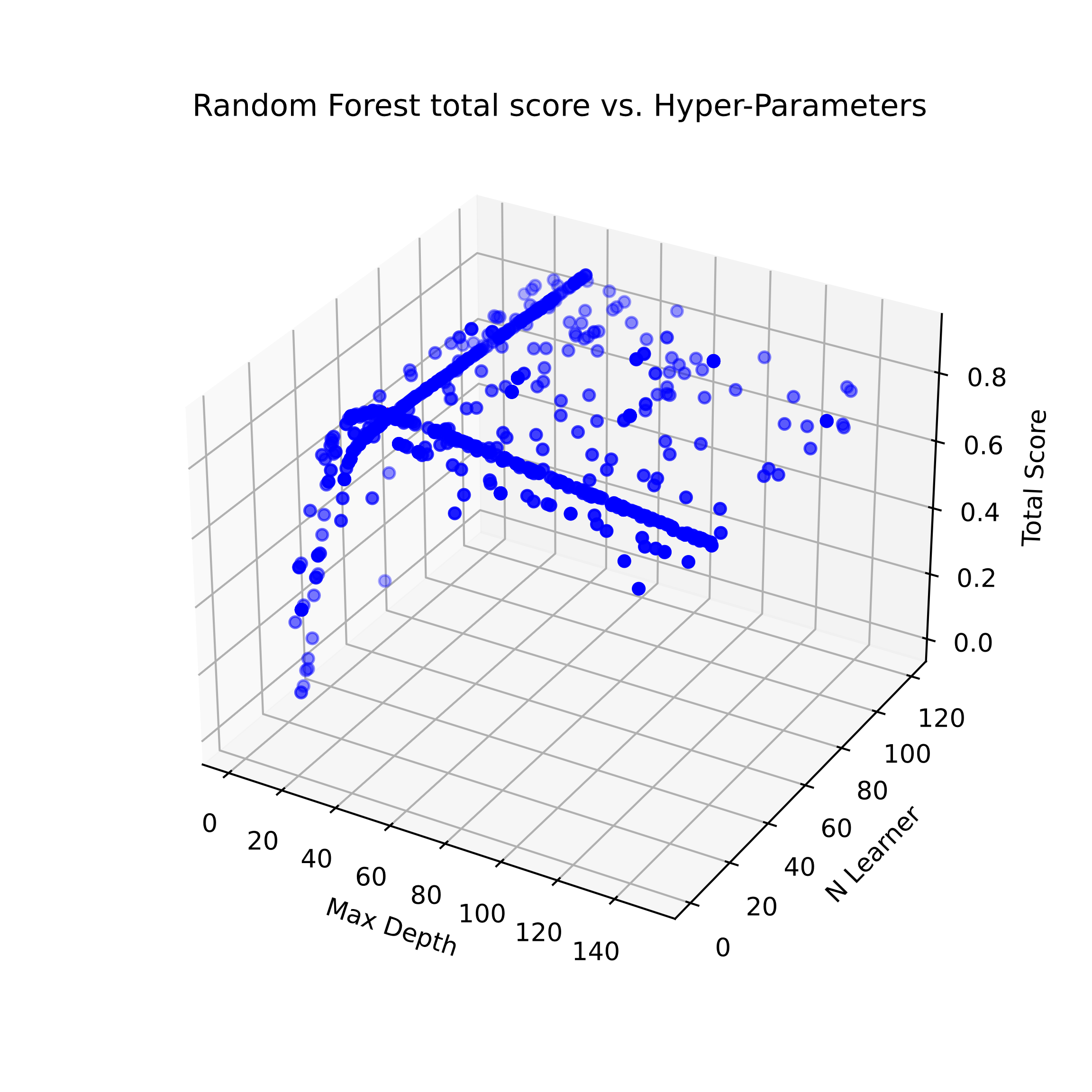
که در آن زمان آزمایش برای دادگان آزمایش، حداکثر عمق درخت و و ضرایب تنظیم تابع برازش است. در رابطه ‏(4˗1) F1-Score در بازه‌ی قرار دارد. بنابراین مقادیر و نیز در این بازه به گونه‌ای انتخاب می‌شود که زمان و حداکثر عمق درختان نسبت به مقدار F1-Score اثر کمتری را در تابع برازش داشته باشد. این مقادیر به صورت تجربی در تابع برازش قرار داده شده و مقادیر آن برای هر مدل یادگیری گروهی در ‏جدول (4˗10) ذکر شده است.

مقادیر ضرایب خطی تابع برازش الگوریتم ژنتیک

|  |  |  |
| --- | --- | --- |
| الگوریتم یادگیری |  |  |
| جنگل تصادفی |  |  |
| AdaBoost |  |  |
| XGBoost |  |  |

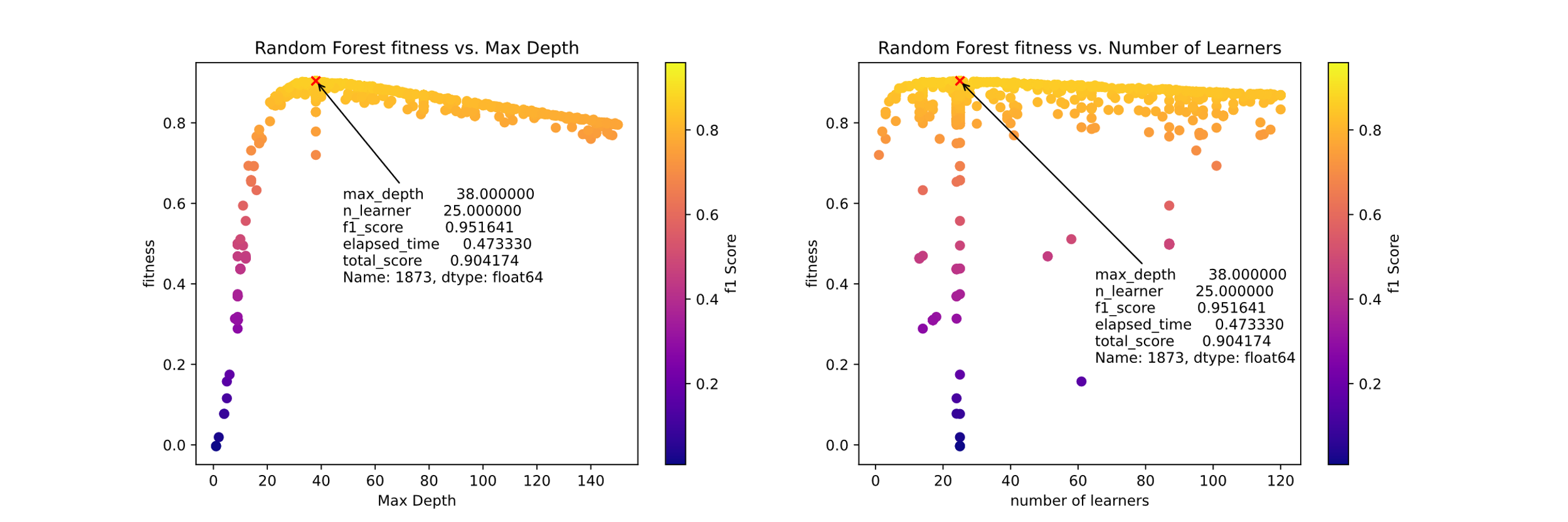
*در الگوریتم ژنتیک، حداکثر عمق درختان و تعداد یادگیرندگان به عنوان ابرپارامتر تنظیم معرفی می‌شود. اندازه جمعیت در این الگوریتم برابر با 20، احتمال جهش برابر 0.1، تعداد تکرار الگوریتم ژنتیک برابر با 50 در نظر گرفته شده است. همچنین برای انتخاب تصادفی جمعیت از تابع توزیع یکنواخت استفاده شده است.*

با اجرای الگوریتم ژنتیک در جنگل تصادفی، مقدار تابع هدف(منفی مقدار تابع برازش) در تکرارهای الگوریتم مطابق با نمودار راست ‏شکل (4˗12) پس از اتمام 50 تکرار در الگوریتم ژنتیک، جنگل تصادفی شامل 25 درخت با حداکثر عمق 38 برای رسیدن به مقدار بهینه %90.04 تابع برازش مناسب دیده شده است.

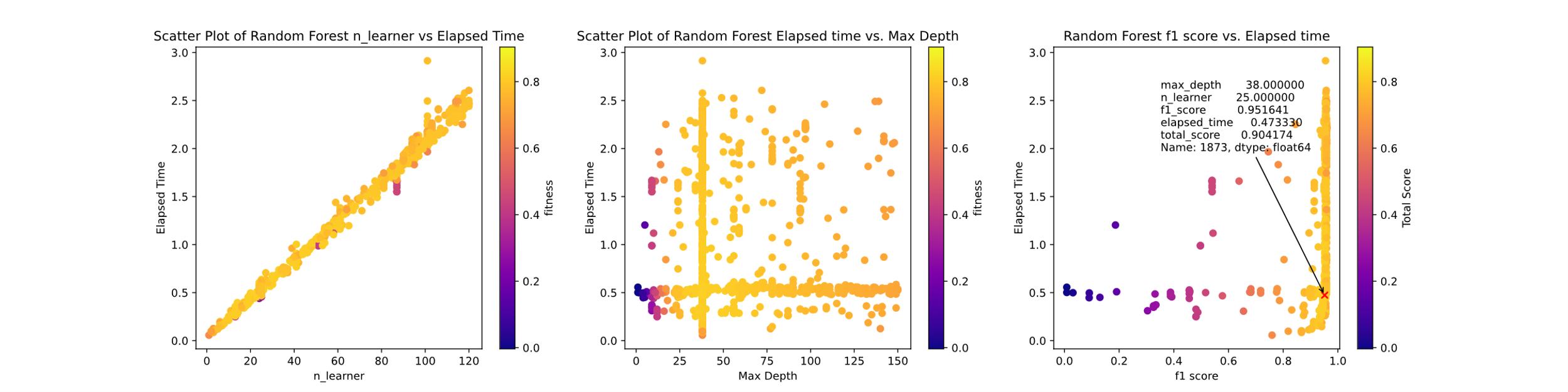
نمودار تابع هدف برحسب تکرار (راست) و نمودار مقدار تابع برازش برحسب ابرپارامتر به‌ازای هر نمونه از جمعیت (چپ) در جنگل تصادفی

در نمودار سمت چپ ‏شکل (4˗12) نیز مقدار تابع برازش به ازای هر نمونه برحسب حداکثر عمق درختان و تعداد یادگیرندگان نشان داده شده است. با ترسیم دو بعدی تابع برازش برحسب مقادیر پارامتر آزاد ‏شکل (4˗13) حاصل می‌شود که در آن رنگ نمونه‌های الگوریتم ژنتیک نشان‌دهنده معیار F1-Score است. با مشاهده این نمودار مشخص است که نمونه انتخابی الگوریتم ژنتیک که با علامت مشخص شده است، دارای بیشترین مقدار تابع برازش است و همچنین مقدار معیار دقت آن نیز مطلوب است.



معیار دقت و مقدار تابع برازش الگوریتم ژنتیک برحسب ابرپارامترهای جنگل تصادفی

نمودارهای زمان



* 1. مقایسه و جمع‌بندی نتایج

متن

[1] J. Torres-Sospedra *et al.*, "UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," in *2014 international conference on indoor positioning and indoor navigation (IPIN)*, 2014, pp. 261-270: IEEE.

[2] J. Torres-Sospedra *et al.*, "The smartphone-based offline indoor location competition at IPIN 2016: Analysis and future work," *Sensors,* vol. 17, no. 3, p. 557, 2017.

[3] P. Roy, C. Chowdhury, D. Ghosh, and S. Bandyopadhyay, "JUIndoorLoc: A ubiquitous framework for smartphone-based indoor localization subject to context and device heterogeneity," *Wireless Personal Communications,* vol. 106, pp. 739-762, 2019.

[4] S. Putatunda and K. Rama, "A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of XGBoost," in *Proceedings of the 2018 international conference on signal processing and machine learning*, 2018, pp. 6-10.

[5] J. Yoo, "Change detection of RSSI fingerprint pattern for indoor positioning system," *IEEE sensors journal,* vol. 20, no. 5, pp. 2608-2615, 2019.

[6] A. H. Salamah, M. Tamazin, M. A. Sharkas, M. Khedr, and M. Mahmoud, "Comprehensive investigation on principle component large-scale Wi-Fi indoor localization," *Sensors,* vol. 19, no. 7, p. 1678, 2019.

1. Exploratory Data Analysis [↑](#footnote-ref-2)
2. Stratified 5-Fold Cross Validation [↑](#footnote-ref-3)