



دانشکده مهندسی برق

## بهبود دقیق موقعیت یابی مبتنی بر RSS با استفاده از الگوریتم‌های یادگیری گروهی

پایان‌نامه برای دریافت درجه کارشناسی ارشد  
در رشته مهندسی برق گرایش کنترل

نام دانشجو

حمید عرب سرخی

استاد راهنما:

دکتر سعید عباداللهی

اسفندماه ۱۴۰۲



دانشکده مهندسی برق

## بهبود دقیق موقعیت یابی مبتنی بر RSS با استفاده از الگوریتم‌های یادگیری گروهی

پایان‌نامه برای دریافت درجه کارشناسی ارشد  
در رشته مهندسی برق گرایش کنترل

نام دانشجو

حمید عرب سرخی

استاد راهنما:

دکتر سعید عباداللهی

۱۴۰۲ اسفندماه

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

## تأییدیه‌ی هیئت داوران جلسه‌ی دفاع از پایان‌نامه/رساله

نام دانشکده: دانشکده مهندسی برق

نام دانشجو: حمید عرب سرخی

عنوان پایان‌نامه یا رساله: بهبود دقیقت موقعیت‌یابی مبتنی بر RSS با استفاده از الگوریتم‌های یادگیری گروهی

تاریخ دفاع: ۱۴۰۲ اسفند ۱۹

رشته: مهندسی برق

گرایش: کنترل

| ردیف | سمت              | نام و نام خانوادگی | مرتبه دانشگاهی | دانشگاه یا مؤسسه | امضا |
|------|------------------|--------------------|----------------|------------------|------|
| ۱    | استاد راهنمای    | دکتر سعید عبدالعلی | استادیار       | علم و صنعت ایران |      |
| ۵    | استاد مدعو داخلی | دکتر سعید شمقدری   | دانشیار        | علم و صنعت ایران |      |
| ۶    | استاد مدعو خارجی | دکتر حامد کبریانی  | دانشیار        | تهران            |      |

## تأییدیه‌ی صحت و اصالت نتایج

### با اسمه تعالی

اینجانب حمید عرب سرخی به شماره دانشجویی ۴۰۰۶۱۶۱۴۸ دانشجوی رشته مهندسی برق گرایش کنترل مقطع تحصیلی کارشناسی ارشد تأیید می‌نمایم که کلیه‌ی نتایج این پایان‌نامه/رساله حاصل کار اینجانب و بدون هرگونه دخل و تصرف است و موارد نسخه‌برداری شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده‌ام. در صورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انضباطی...) با اینجانب رفتار خواهد شد و حق هرگونه اعتراض در خصوص احراق حقوق مکتب و تشخیص و تعیین تخلف و مجازات را از خویش سلب می‌نمایم. در ضمن، مسئولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذی‌صلاح (اعم از اداری و قضایی) به عهده اینجانب خواهد بود و دانشگاه هیچ‌گونه مسئولیتی در این خصوص نخواهد داشت.

نام و نام خانوادگی:

امضا و تاریخ:

## مجوز بهره برداری از پایان نامه

بهره برداری از این پایان نامه در چهار چوب مقررات کتابخانه و با توجه به محدودیتی که توسط استاد راهنما به شرح زیر تعیین می شود، بلامانع است:

- بهره برداری از این پایان نامه / رساله برای همگان بلامانع است.
- بهره برداری از این پایان نامه / رساله با اخذ مجوز از استاد راهنما، بلامانع است.
- بهره برداری از این پایان نامه / رساله تا تاریخ ..... ممنوع است.

نام استاد راهنما: دکتر سعید عباداللهی

تاریخ:

امضا:

## **تشکر و قدردانی:**

سپاس خدای بزرگ را که مرا یاری رساند تا بتوانم این مقطع تحصیلی را به پایان رسانده و گامی در راستای اعتلای علم بردارم. از استاد راهنمای گرانقدرم جناب آقای دکتر سعید عباداللهی که وجودشان همیشه قوتی برای انجام کارهایم بوده است و بدون شک انجام این پایاننامه بدون کمک و راهنمایی‌های ارزنده ایشان امکان پذیر نبوده است، کمال تشکر را دارم.

## چکیده

سیستم‌های مبتنی بر موقعیت (LBS) با ارائه اطلاعات موقعیت کاربر، خدمات مبتنی بر اینترنت اشیا (IoT) را ارتقا می‌دهند. در فضاهای داخلی، GPS و Galileo دقت کافی ندارند، به همین دلیل از سایر فناوری‌های بی‌سیم برای موقعیت‌یابی استفاده می‌شود. معیار قدرت سیگنال دریافتی (RSS) به دلیل سادگی محاسبات، برای موقعیت‌یابی بکار می‌رود. اما این معیار تحت تاثیر شرایط محیطی و فیزیکی قرار می‌گیرد. الگوریتم‌های یادگیری ماشین می‌توانند موقعیت کاربر را در محیط‌های داخلی با دقت بیشتری تخمین بزنند، اما نیاز به کالیبراسیون متناسب با تغییرات محیط دارند. این پایان‌نامه به بررسی کاربرد الگوریتم‌های یادگیری گروهی در تخمین موقعیت کاربر در محیط‌های داخلی با استفاده از Wi-Fi می‌پردازد. این الگوریتم‌ها با ترکیب خروجی چند مدل یادگیری ماشین، دقت تخمین موقعیت را افزایش می‌دهند. با وجود مزایای یادگیری گروهی، چالش‌هایی مانند نیاز به منابع بیشتر و زمان پردازش بالا وجود دارد که با سیستم‌های بلاذرنگ و کم‌هزینه سازگار نیست. برای رفع این مشکل، دو راهکار کاهش ابعاد و الگوریتم ژنتیک برای یافتن مدل بهینه ارائه شده است. روش‌های پیشنهادی بر روی مجموعه داده‌های جمع‌آوری شده در شرایط مختلف محیطی پیاده‌سازی و با روش‌های دیگر مقایسه شده‌اند. نتایج این پایان‌نامه نشان داده است که با درنظرگیری معیاری شامل دقت و حساسیت تخمین مدل در تابع برازش، میانگین خطای موقعیت‌یابی با درنظرگیری مجموعه متشکل از دادگان همه طبقات به  $0.27$  متر برسد که این میزان خطای حاصل مدل یادگیری گروهی با دقتی در حدود  $95\%$  است. برای مقایسه نتایج این پژوهش با سایر پژوهش‌ها، دادگان آزمایش شامل یک طبقه مورد استفاده قرار گرفته است که میانگین خطای به  $0.25$  متر از مدلی با دقت  $98.3\%$  رسیده است.

**واژه‌های کلیدی:** موقعیت‌یابی، یادگیری ماشین، یادگیری گروهی، الگوریتم ژنتیک

# فهرست مطالب

| عنوان   | شماره صفحه |
|---|------------|
| فصل ۱: مقدمه  | ۱          |
| ۱-۱- پس زمینه   | ۲          |
| ۱-۲- مروری بر کارهای انجام شده                          | ۳          |
| ۱-۳- بیان مسئله   | ۵          |
| ۱-۴- اهداف و دستاوردهای پژوهش                           | ۶          |
| ۱-۵- مروری بر فصل های پایان نامه                        | ۷          |
| فصل ۲: مفاهیم و تعاریف                                  | ۸          |
| ۲-۱- مقدمه  | ۹          |
| ۲-۲- مفاهیم و تعاریف موقعیت یابی                        | ۹          |
| ۲-۲-۱- خدمات مبتنی بر مکان                              | ۹          |
| ۲-۲-۲- سیستم تعیین موقعیت داخلی                         | ۹          |
| ۲-۳-۲- موقعیت و مکان                                    | ۱۰         |
| ۲-۴-۲- نقطه دسترسی                                      | ۱۰         |
| ۲-۳-۲- انواع داده در موقعیت یابی                        | ۱۰         |
| ۲-۱-۳-۲- قدرت سیگنال دریافتی (RSS)                      | ۱۱         |
| ۲-۲-۳-۲- زمان ورود (ToA)                                | ۱۲         |
| ۲-۳-۳-۲- زمان تفاضلی رسیدن (TDoA)                       | ۱۲         |
| ۲-۴-۳-۲- زاویه رسیدن (AoA) و زاویه تفاضلی رسیدن (ADoA)  | ۱۳         |
| ۲-۵-۳-۲- فاز رسیدن (PoA) و فاز تفاضلی رسیدن (PDoA)      | ۱۳         |
| ۲-۶-۳-۲- اطلاعات حالت کانال (CSI)                       | ۱۴         |
| ۲-۷-۳-۲- قدرت و کیفیت سیگنال مرجع دریافتی (RSRP و RSRQ) | ۱۴         |
| ۲-۴-۲- فناوری های موقعیت یابی مبتنی بر RSS و چالش ها    | ۱۴         |
| ۲-۴-۲- Wi-Fi  | ۱۵         |
| ۲-۴-۲- Bluetooth  | ۱۶         |
| ۲-۴-۲- RFID   | ۱۶         |
| ۲-۴-۲- ZigBee   | ۱۶         |
| ۲-۴-۲- Cellular   | ۱۷         |
| ۲-۴-۲- چالش های موقعیت یابی مبتنی بر RSS                | ۱۷         |
| ۲-۵-۲- روش های پایه در موقعیت یابی                      | ۱۹         |
| ۲-۵-۲- ۱- الگوریتم مجاورت                               | ۱۹         |
| ۲-۵-۲- ۲- الگوریتم مثلث بندی                            | ۱۹         |
| ۲-۵-۲- ۳- الگوریتم چندوجهی بندی                         | ۲۰         |
| ۲-۵-۲- ۴- الگوریتم Min-Max                              | ۲۰         |
| ۲-۵-۲- ۵- الگوریتم حداقل درست نمایی                     | ۲۱         |

|         |  |
|---------|--|
| ۲۱..... | ۶-۵-۲- الگوریتم موقعیت یابی اثر انگشت                        |
| ۲۲..... | ۶-۲- الگوریتم های یادگیری ماشین                              |
| ۲۲..... | ۱-۶-۲- ساختار موقعیت یابی با یادگیری ماشین                   |
| ۲۴..... | ۲-۶-۲- تکنیک های تحت نظارت یادگیری ماشین                     |
| ۲۷..... | ۳-۶-۲- تکنیک های نیمه نظارتی یادگیری ماشین                   |
| ۲۷..... | ۴-۶-۲- تکنیک های بدون نظارت یادگیری ماشین                    |
| ۲۸..... | ۵-۶-۲- تکنیک های یادگیری ماشین تحت شرایط مختلف آموزش و آزمون |
| ۲۸..... | ۶-۶-۲- چالش های موقعیت یابی با الگوریتم های یادگیری ماشین    |
| ۳۰..... | ۷-۲- جمع بندی  |

### فصل ۳: راهکارهای پیشنهادی بهبود دقت در تعیین موقعیت.....۳۱

|         |   |
|---------|---|
| ۳۲..... | ۱-۳- مقدمه                              |
| ۳۲..... | ۲-۳- روش های یادگیری گروهی              |
| ۳۴..... | ۲-۲-۳- توسعه الگوریتم های یادگیری گروهی |
| ۳۵..... | ۳-۲-۳- اجزای الگوریتم یادگیری گروهی     |
| ۳۷..... | ۴-۲-۳- روش Bagging                      |
| ۳۹..... | ۵-۲-۳- روش Boosting                     |
| ۴۶..... | ۳-۳- پیش پردازش داده                    |
| ۴۷..... | ۱-۳-۳- تجزیه مؤلفه های اساسی (PCA)      |
| ۴۸..... | ۴-۳- الگوریتم ژنتیک                     |
| ۵۱..... | ۳-۵- معیارهای ارزیابی                   |
| ۵۱..... | ۱-۵-۳- معیارهای قابل اندازه گیری عملکرد |
| ۵۳..... | ۲-۵-۳- معیارهای کیفی عملکرد             |
| ۵۴..... | ۳-۶- جمع بندی                           |

### فصل ۴: شبیه سازی و ارزیابی نتایج.....۵۶

|         |  |
|---------|--|
| ۵۷..... | ۱-۴- مقدمه   |
| ۵۷..... | ۴-۴- دادگان موقعیت یابی مبتنی بر RSS                 |
| ۵۹..... | ۱-۲-۴- پیش پردازش دادگان موقعیت یابی                 |
| ۶۰..... | ۴-۳- آموزش موقعیت یاب با الگوریتم های یادگیری گروهی  |
| ۶۱..... | ۱-۳-۴- مدل موقعیت یاب الگوریتم جنگل تصادفی           |
| ۶۳..... | ۲-۳-۴- مدل موقعیت یاب الگوریتم AdaBoost              |
| ۶۵..... | ۳-۳-۴- مدل موقعیت یاب الگوریتم XGBoost               |
| ۶۶..... | ۴-۴- کاهش ابعاد دادگان موقعیت یابی                   |
| ۶۸..... | ۵-۴- تنظیم ابرپارامترها با استفاده از الگوریتم ژنتیک |
| ۷۳..... | ۶-۴- مقایسه و جمع بندی نتایج                         |

### فصل ۵: جمع بندی و پیشنهادها.....۷۵

|          |                              |
|----------|------------------------------|
| ۷۶.....  | ۱-۵ جمع‌بندی                 |
| ۷۷.....  | ۲-۵ پیشنهادها                |
| ۷۸.....  | مراجع                        |
| ۸۲ ..... | پیوست                        |
| ۸۳.....  | پیوست الف: شبه کد درخت تصمیم |

## فهرست اشکال

### شماره صفحه

### عنوان

|   |    |
|---|----|
| شکل (۱-۲) داده‌های سیستم موقعیت‌یابی برای اندازه‌گیری‌های فاصله و جهت [۳].....  | ۱۱ |
| شکل (۲-۲) محدوده دقت فناوری‌های موقعیت‌یابی بر حسب استفاده در داخل یا خارج ساختمان [۱۹].....  | ۱۵ |
| شکل (۳-۲) طرح گرافیکی از چندمسیره شدن، خط غیر دید و خط دید در موقعیت‌یابی داخلی .....   | ۱۸ |
| شکل (۴-۲) روش موقعیت‌یابی مبتنی بر اثر انگشت [۳].....   | ۲۲ |
| شکل (۵-۲) طرح ساختار موقعیت‌یابی با الگوریتم یادگیری ماشین [۲۰].....  | ۲۳ |
| شکل (۱-۳) کاهش واریانس با استفاده از الگوریتم‌های یادگیری گروهی [۲۱].....   | ۳۳ |
| شکل (۲-۳) رویکرد Bagging در مسائل طبقه‌بندی [۲۱].....   | ۳۷ |
| شکل (۳-۳) نمایش گرافیکی ایده اولین روش Boosting ارائه شده در [۱۱].....  | ۴۰ |
| شکل (۴-۳) عملیات ترکیب متقطع و جهش در تولید جمعیت جدید الگوریتم ژنتیک [۳۲].....   | ۵۱ |
| شکل (۱-۴) نمایشی از تقسیم‌بندی اثر انگشت طبقه چهارم محیط مجموعه دادگان استفاده شده [۳۵].....  | ۵۸ |
| شکل (۲-۴) نمودار هیستوگرام تعداد دفعات تکرار هر مقدار RSSI در نقاط دسترسی AP133 و AP053 .....<br>۵۹   |    |
| شکل (۳-۴) نقشه توزیع مقادیر RSSI در مختصات X و Y از نقاط دسترسی AP030 و AP070 در طبقه چهارم.....  | ۶۰ |
| شکل (۴-۴) نمودار نتیجه ارزیابی، خطأ و زمان آزمایش جنگل تصادفی برای تعداد متفاوت درختان و حداقلر عمق ثابت.....                                 | ۶۲ |
| شکل (۵-۴) نمودار نتیجه ارزیابی، خطأ و زمان آزمایش جنگل تصادفی برای عمق متفاوت درختان و تعداد درخت.....  | ۶۲ |
| شکل (۶-۴) نمودار نتیجه ارزیابی، خطأ و زمان آزمایش AdaBoost برای تعداد متفاوت درختان و حداقلر عمق ثابت.....                                    | ۶۳ |
| شکل (۷-۴) نمودار نتیجه ارزیابی، خطأ و زمان آزمایش AdaBoost برای عمق متفاوت درختان و تعداد درخت .....  | ۶۴ |
| شکل (۸-۴) نمودار نتیجه ارزیابی، خطأ و زمان آزمایش XGBoost برای تعداد متفاوت درختان و حداقلر عمق ثابت.....                                     | ۶۵ |
| شکل (۹-۴) نمودار نتیجه ارزیابی، خطأ و زمان آزمایش XGBoost برای عمق متفاوت درختان و تعداد درخت .....   | ۶۶ |
| شکل (۱۰-۴) نمودار مقادیر منفرد ماتریس کوواریانس ویژگی‌ها.....   | ۶۷ |
| شکل (۱۱-۴) تجسم داده در فضای دو و سه بعدی با استفاده از روش کاهش ابعاد PCA .....  | ۶۷ |
| شکل (۱۲-۴) نمودار تابع هدف بر حسب تکرار (راست) و نمودار مقدار تابع برآش بر حسب ابر پارامتر بهازای هر نمونه از جمعیت (چپ) در جنگل تصادفی ..... | ۶۹ |
| شکل (۱۳-۴) معیار دقت و مقدار تابع برآش الگوریتم ژنتیک بر حسب ابر پارامترهای جنگل تصادفی .....   | ۷۰ |



## فهرست جداول

### شماره صفحه

### عنوان

---

|  |    |
|--|----|
| جدول (۱-۴) مشخصات سیستم سختافزاری موقعیتیاب برای اجرای شبیه‌سازی.....  | ۶۰ |
| جدول (۲-۴) نتایج شبیه‌سازی درخت تصمیم با دادگان ارزیابی و حداکثر عمق‌های متفاوت.....   | ۶۱ |
| جدول (۳-۴) نتایج شبیه‌سازی جنگل تصادفی و تعداد متفاوت درختان و عمق ثابت درختان (حداکثر)....  | ۶۲ |
| جدول (۴-۴) نتایج شبیه‌سازی جنگل تصادفی برای عمق متفاوت درختان و تعداد ۴۰ درخت.....   | ۶۳ |
| جدول (۵-۴) نتایج شبیه‌سازی AdaBoost با تعداد متفاوت درختان و عمق ثابت درختان (حداکثر) .....  | ۶۴ |
| جدول (۶-۴) نتایج شبیه‌سازی AdaBoost برای عمق متفاوت درختان و تعداد ۴۰ درخت.....  | ۶۴ |
| جدول (۷-۴) نتایج شبیه‌سازی XGBoost با تعداد متفاوت درختان و عمق ثابت درختان (حداکثر).....  | ۶۵ |
| جدول (۸-۴) نتایج شبیه‌سازی XGBoost برای عمق متفاوت درختان و تعداد ۴۰ درخت.....   | ۶۶ |
| جدول (۹-۴) نتایج روش PCA با کاهش مرتبه ۱۰ در روش‌های یادگیری گروهی با ۴۰ درخت .....  | ۶۸ |
| جدول (۱۰-۴) مقادیر ضرایب خطی تابع برازش الگوریتم ژنتیک.....  | ۶۹ |
| جدول (۱۱-۴) نتایج اجرای مدل گروهی با مقادیر بهینه در ناحیه جستجو.....  | ۷۳ |
| جدول (۱۲-۴) مقایسه نتایج اجرای مدل گروهی تنظیم شده با روش‌های ارائه شده در [۱۱] و [۳۱] و<br>به کارگیری دادگان آزمایش طبقه چهارم..... | ۷۴ |

## فهرست علائم اختصاری

|      |                                   |
|------|-----------------------------------|
| AP   | Access Point                      |
| ADoA | Angel Difference of Arrival       |
| AoA  | Angel of Arrival                  |
| AoA  | Angel of Arrival                  |
| CSI  | Channel State Information         |
| GPS  | Global Positioning System         |
| IPS  | Indoor Positioning System         |
| LoS  | Line-of sight                     |
| LBS  | Location Based Services           |
| NLoS | Non-Line of Sight                 |
| PDoA | Phase Difference of Arrival       |
| PoA  | Phase of Arrival                  |
| RFID | Radio Frequency IDentification    |
| RSS  | Received Signal Strength          |
| RSSI | Received Signal Strength Index    |
| RSRP | Reference Signal Received Power   |
| RSRQ | Reference Signal Received Quality |
| TDoA | Time Difference of Arrival        |
| ToA  | Time of Arrival                   |

## **فصل ۱:**

### **مقدمه**

## ۱-۱- پس زمینه

باتوجه به افزایش سریع تعداد گوشی‌های هوشمند و گسترش دستگاه‌های بی‌سیم مختلف در دهه گذشته، اینترنت اشیا (IoT) به عنوان یک حوزه بالاترین اهمیت برای انواع صنایع به وجود آمده است. نظارت بر ترافیک، دستگاه‌های قابل پوشش، مراقبت‌های بهداشتی و صرفه‌جویی در مصرف انرژی تنها چندین حوزه از بسیاری از زمینه‌های کاربردی مختلف برای IoT هستند. افروختن سیستم‌های مبتنی بر موقعیت (LBS) به چنین حوزه‌ی در حال توسعه‌ای، خدماتی با غنی‌ترین و دقیق‌ترین مشخصات را به کاربران نهایی ارائه می‌دهد.

فناوری‌های مبتنی بر موقعیت مانند GPS و Galileo به عنوان ابزارهای مهم و معتمد در سناریوهای فضای باز ثابت شده‌اند. با این وجود، در فضای داخلی که نیاز به دقت بیشتری نسبت به دقت چند متری موقعیت یابی در فضای باز دارند، انتقال چندمسیره اطلاعات انتقالی، نقش حیاتی را در عملکرد سیستم موقعیت یابی ایفا می‌کند. GPS و Galileo به دلیل عملکرد ضعیف در محیط‌های داخلی استفاده نمی‌شود؛ بنابراین، فناوری‌هایی مانند Wi-Fi و BLE برای مواجهه با مسائل موقعیت یابی در داخل استفاده شوند که با مشکلات ایجاد شده توسط انتشار چندمسیره<sup>۱</sup> به دلیل وجود اشیا در هر اتاق مواجه است.

موقعیت یابی داخلی یک حوزه تحقیقاتی بوده که بیش از ۳۰ سال است مورد مطالعه قرار گرفته و در عین حال، فناوری‌های جدید به طور مداوم در حال توسعه هستند. بسته به نوع سیگنال و محیط، برخی از فناوری‌های موقعیت یابی داخلی موجود نتایج بهتری ارائه می‌دهند. الزامات سیستمی مانند مصرف انرژی و پیچیدگی محاسباتی نیز نقش مهمی در تصمیم‌گیری در مورد استفاده از کدام تکنولوژی برای موقعیت یابی داخلی ایفا می‌کنند.

انگیزه‌ی معرفی یادگیری ماشین به عنوان تخمین‌گر مکان، توان فوق العاده‌ای است که این روش‌ها برای سازگاری با سناریوهای مختلف را دارد. این ویژگی‌ها، یادگیری ماشین را در بسیاری از زمینه‌های کاربردی، از جمله مهندسی، اقتصاد، و حتی پزشکی، کارا می‌سازد. دلیل اصلی این جاذبه این است که یادگیری ماشین در اصل، آمار و مهندسی اعداد است که در هر جنبه‌ای از زندگی به شکل دادگان حضور دارند. استفاده از این دادگان برای بهبود درک یک سیستم از نحوه تولید یک خروجی، مفید است؛ زیرا سیستم نه تنها با داده سازگار می‌شود، بلکه نیز به وسیله آن یاد می‌گیرد که چگونه از طریق تنوع گسترده‌ای از روش‌های یادگیری ماشین که امروزه شناخته شده‌اند، داده را یاد بگیرد، پردازش کند و استفاده کند.

الگوریتم‌های یادگیری گروهی، یکی از مهم‌ترین راهکارهای بهبود عملکرد سیستم‌های مبتنی بر یادگیری

<sup>1</sup> Multipath Propagation

ماشین هستند. این الگوریتم‌ها با توجه به مفهوم یادگیری توسط گروه یا جمعیت، به سیستم‌ها امکان می‌دهند تا از تجربیات و دانش گسترده‌تری بهره‌مند شوند. در این الگوریتم‌ها، یک گروه از مدل‌های یادگیری ماشین به طور همزمان و همکاری با یکدیگر آموزش می‌بینند. این تعامل بین مدل‌ها، امکان بهاشتراك‌گذاری اطلاعات و تجربیات را فراهم می‌کند که می‌تواند به بهبود دقت و کارایی سیستم منجر شود [۱].

رویکرد گروهی به سیستم اجازه می‌دهد تا از زمینه‌های گسترده‌تری اطلاعات جذب کرده و در مقابل مسائل ناشناخته، بهبود یابد. علاوه‌بر آن، این الگوریتم‌ها می‌توانند در موقعي که داده‌های محدودی در دسترس هستند، عملکرد بهتری نسبت به روش‌های تک مدل ارائه دهند. از این‌رو، الگوریتم‌های یادگیری گروهی به عنوان یک راهکار کلان و مؤثر در بهبود عملکرد سیستم‌های مبتنی بر یادگیری ماشین مورد توجه قرار گرفته‌اند.

## ۱-۲- مروری بر کارهای انجام شده

موقعیت‌یابی در مکان‌های سرپوشیده همواره با چالش‌های جدی چندمسیره شدن انتشار سیگنال، تضعیف سیگنال، حضور و یا عدم حضور افراد و اثرات تغییرات تدریجی محیط همرا به است که این چالش‌ها مانع بر گسترش سیستم‌های مبتنی بر موقعیت شده است؛ لذا از سال ۱۹۹۰ میلادی این موضوع به طور گسترده مورد تحقیق و توسعه پژوهشگران قرار گرفته است [۲]. با رشد تکنولوژی و فناوری‌های ارتباطی بی‌سیم زمینه رشد در سیستم‌های موقعیت‌یابی نیز افزایش یافت. روش‌های مختلفی برای هر یک از این فناوری و متناسب با ویژگی‌های آن ارائه شده‌اند. روش‌های رایج تعیین محدوده مبتنی بر قدرت سیگنال دریافتی (RSS)، زمان ورود (ToA)، زمان ورود تفاضلی (TD0A)، زاویه رسیدن (AoA) و اطلاعات حالت کانال (CSI) پیشنهاد شده است مقالات مرتبط با این روش‌ها در [۳] معرفی و بررسی شده‌اند.

در سال‌های اخیر، تکنیک‌های یادگیری ماشین در تعیین موقعیت مکان‌های سرپوشیده به کار رفته است و اثربخشی این تکنیک‌ها در استخراج داده، یادگیری و بهبود دقت موقعیت‌یابی مشاهده شده است [۴]. این رویکردها نسبت به روش‌های سنتی ریاضی برای مسائل غیرخطی پیچیده که با معادلات دست‌نویس بسیار دشوار هستند، بسیار مؤثر است. تکنیک‌های یادگیری ماشین همچنین توانسته است راه حلی برای مقیاس‌پذیری مدل در فضاهای داخلی بزرگ را فراهم کند. چراکه مدل‌های یادگیرنده به راحتی می‌تواند با مجموعه‌داده‌های جدید به روزرسانی شود [۶-۴].

برخلاف رویکردهای آماری، الگوریتم‌های یادگیری ماشین به سهولت قابل گسترش هستند تا عملکرد پایداری

را در شرایط محیطی مختلف ارائه کنند. ارائه مدل برای شرایط مختلف محیطی با استفاده از روش‌های یادگیری گروهی امکان‌پذیر است. از این‌رو گرایش به سمت استفاده از الگوریتم‌های یادگیری گروهی افزایش یافته است. چرا که این روش‌ها می‌توانند اطلاعات شرایط مختلف را تجمعی کنند و اعلام موقعیت را با دقت بیشتر انجام دهند. قابلیت یادگیری آنلاین نیز از دیگر مزایای استفاده از روش‌های یادگیری گروهی است و امکان تطبیق گام‌به‌گام با تغییرات محیطی را نیز فراهم می‌کند. در صورتی که این فرایند با روش‌های سنتی بسیار دشوار است [۷، ۸]. بنابراین، تکنیک‌های یادگیری ماشین گروهی مانند Bagging و Boosting برای شرایط مختلف محیطی بسیار مناسب هستند.

در روش Bagging [۹] یک سیستم موقعیت‌یابی با رده‌بندی مبتنی بر دو روش Bagging و روش ترکیبی Bagging با فضای تصادفی استفاده کرده است. همچنین [۱۰] یک سیستم موقعیت‌یابی داخلی بلادرنگ را با استفاده از Bagging و درخت تصمیم مبتنی بر یادگیری گروهی معرفی کرده است. در میان رده‌بندهای مختلف، جنگل تصادفی بهترین نتیجه را در دقت داشته است، با این حال در بسته کردن برخی نمونه‌های آموزش ممکن است بارها و بارها در کیسه‌های اشتباه طبقه‌بندی شوند. همچنین در برخی موارد جنگل تصادفی با تغییر کوچک در داده‌ها دچار تغییر در ساختار عظیم آن می‌شود و ناپایدار می‌گردد.

در روش Boosting [۱۱] از یادگیرنده‌گان ضعیف که به صورت متوالی بر اساس عملکرد یادگیرنده قبلی تولید می‌شوند استفاده کرده است. برخلاف روش Bagging نمونه‌ها با پیش‌بینی نادرست، در فرایند انتخاب احتمال انتخاب بیشتری را دارند. به طور مشابه در کارهای [۱۲، ۱۳] از یک مدل مبتنی بر روش AdaBoost برای تعیین موقعیت نامعلوم کاربر معرفی شده است. روش‌های Boosting پیچیدگی سیستم را افزایش می‌دهد، به علاوه روش AdaBoost بهشدت نسبت به داده‌های نویزی حساس است و از داده‌های پرت بسیار متأثر می‌گردد.

جدا از این دو روش، برخی از تحقیقات نیز برای بهبود دقت موقعیت‌یابی با استفاده از ترکیب رده‌بندهای گوناگون یادگیری ماشین انجام شده است. به عنوان مثال در [۱۴، ۱۵] روی دادگان RSSI در شرایط مختلف جمع‌آوری شده است و بهبود دقت از حدود ۵۸ درصد الی ۸۵ درصد با به کارگیری چندین نوع از رده‌بندها به دقت حدودی ۹۳ درصد رسیده است. در [۱۶] با استفاده از یک الگوریتم وزن‌دهی ابداعی به رده‌بندهای SVM، KNN، دقت رده‌بندی به حدود ۹۴ درصد رسیده است.

## ۱-۳- بیان مسئله

در یک سیستم تعیین موقعیت داخلی، کارکرد ساده و هزینه پایین، از معیارهای اصلی سنجش عملکرد آن است؛ بنابراین با توجه به فرآگیری فناوری‌های Wi-Fi و BLE استفاده از این فناوری‌ها برای کاهش هزینه کارآمد است. در بین این دو فناوری نیز با توجه به پوشش سیگنال فناوری Wi-Fi (در حدود ۵۰ الی ۱۰۰ متر) نسبت به پوشش سیگنال فناوری BLE (در حدود ۱۰ الی ۲۰ متر) در فضای داخلی بزرگ، فناوری Wi-Fi مورد استقبال بیشتری قرار گرفته است. با توجه به ویژگی‌های سیگنال این دو فناوری، داده سطح توان سیگنال دریافتی (RSS) به دلیل سادگی در محاسبات و صرف هزینه پایین‌تر در پژوهش‌های گوناگون به کار رفته است. اما این ویژگی بسیار متأثر از شرایط محیطی است و با اندک تغییراتی در محیط قدرت سیگنال تغییر می‌کند. از این‌رو برای مواجهه با این موضوع، ناگزیر به استفاده از تکنیک‌هایی هستیم که بتواند اثرات این تغییرات را کنترل کند.

بهره‌گیری از الگوریتم‌های یادگیری ماشین به‌واسطه پتانسیل مدل‌سازی غیرخطی در تعیین موقعیت کاربر مورد توجه قرار گرفته است. همچنین مدل‌های به‌دست آمده از این الگوریتم‌ها می‌توانند در مقابل چالش‌های موجود اعم از تداخلات محیطی، محوشدگی چندمسیره شدن و غیره مقاومت خوبی نشان دهند. مدل‌های پایه یادگیری ماشین برخلاف روش‌های سنتی تخمین موقعیت بهتری در شرایط مختلف محیطی دارند، اما همچنان با تغییرات این شرایط محیطی، نیازمند کالیبراسیون دوره‌ای هستند.

هدف این پایان‌نامه استفاده از زیرمجموعه‌ای از الگوریتم‌های یادگیری با عنوان الگوریتم‌های یادگیری گروهی است که بتواند با شرایط مختلف محیطی، بهترین کارکرد را داشته باشد. الگوریتم‌های یادگیری گروهی با بهره‌گیری از نظر جمعی از مدل‌ها می‌توانند تصمیم بهتری در تبیین موقعیت کاربر اعلام کند. این مدل‌ها لزوماً موظف به اعلام بهترین پاسخ را اعلام نیستند؛ از این‌رو به آن‌ها یادگیرنده ضعیف<sup>۱</sup> می‌گویند.

نظر جمعی این یادگیرنده‌های ضعیف، یک سیستم با پاسخ مطلوب را نتیجه می‌دهد.

گرچه استفاده از الگوریتم‌های یادگیری گروهی سیستم را به سمت پاسخ مطلوب‌تر سوق می‌دهد، اما باید در نظر داشت که استفاده از آن‌ها، هزینه سخت‌افزاری بالایی را به همراه خواهد داشت و می‌تواند در سرعت پاسخ سیستم تعیین موقعیت اثر منفی بگذارد. چرا که تعداد زیاد یادگیرنده‌های ضعیف در حافظه موقت واحد پردازش داده، فضای زیادی را اشغال می‌کند و همچنین تعیین موقعیت نهایی مستلزم اجرا و اعلام موقعیت هر یک از آن‌ها است که برای یک سیستم موقعیت‌یاب بلادرنگ مناسب نیست. برای رویارویی با این مسئله،

<sup>1</sup> Weak Learner

می‌بایست با تعیین ابرپارامترها<sup>۱</sup> و یا پیش‌پردازش داده‌ها به یک مدل بهینه با سرعت و دقت مناسب دست یافت که برای یک سیستم موقعیت‌یاب مطلوب باشد. بدین منظور، لازم است یک تابع هزینه معرفی شود تا محدودیت‌های دقت، زمان و سایر هزینه‌ها در آن منظور شود و مدل را به سمت مدل بهینه سوق دهد.

## ۴-۱- اهداف و دستاوردهای پژوهش

هدف این پایان‌نامه بهبود موقعیت‌یابی مبتنی بر RSS در مکان‌های سرپوشیده با استفاده از الگوریتم‌های یادگیری گروهی است. بدین منظور ابتدا باید با توجه‌به محدودیت‌های فناوری‌های ارتباطی بی‌سیم، فناوری متداول و کم‌هزینه بررسی شود. در دهه اخیر، با توجه‌به رشد دستگاه‌های تلفن همراه و استفاده عموم از آن‌ها، فناوری‌های ارتباطی آن‌ها نظیر بلوتوث و Wi-Fi در کاربرد موقعیت‌یابی بیشتر مورد توجه است؛ بنابراین جمع‌آوری مجموعه‌ای مشتمل بر قدرت سیگنال دریافتی از یکی از این دو فناوری گام ابتدایی حل مسئله است. مجموعه‌دادگان جمع‌آوری شده می‌بایست شامل اطلاعاتی از محیط که در شرایط مختلف گردآوری شده است باشد تا مدل یادگیرنده نسبت به این شرایط مقاوم باشد. همچنین نیاز است فاقد داده‌های بدون برچسب موقعیت نیز باشد. تعداد و تنوع داده‌ها در یادگیری بهتر مدل مؤثر است، لذا دادگان از نظر تعداد و تنوع باید غنی باشد.

در گام دوم تحلیل دادگان جمع‌آوری شده با معیارهای آماری برای انتخاب راهکار و مدل یادگیری گروهی در حل مسئله حائز اهمیت است. در گام بعدی، مدل‌های یادگیری گروهی و نحوه آموزش و آزمایش آن‌ها معرفی می‌شود و این مدل‌ها با بخشی از دادگان جمع‌آوری شده که به عنوان «دادگان آموزش» شناخته می‌شود، آموزش می‌بینند و بر آن منطبق می‌شود. پس از آن، با بخش دیگری از دادگان جمع‌آوری شده به عنوان «دادگان آزمایش»، مدل‌های آموزش‌دیده آزمایش می‌شوند و نتایج حاصل از آن با انتخاب معیار مناسب ارزیابی، از نظر دقت و سرعت سنجیده می‌شوند و در نهایت الگوریتم‌های یادگیری گروهی با یکدیگر مقایسه شده و نتایج آن تفسیر و تحلیل می‌شود.

با توجه‌به چالش‌ها و محدودیت‌های موجود در روش‌های یادگیری گروهی، تعیین ابرپارامترهای این مدل‌ها در سرعت، دقت و پیاده‌سازی سخت‌افزاری آن‌ها نقش کلیدی دارد؛ بنابراین برای مواجهه با این موضوع، دو راهکار پیشنهادی برای تعیین ابرپارامترها بیان می‌شود. راهکار پیشنهادی اول، کاهش ابعاد مجموعه‌دادگان است. در

<sup>1</sup> Hyperparameters

این راهکار، با توجه میزان اهمیت و اثرگذاری ویژگی‌های مجموعه‌دادگان در فضای اولیه، به فضای ثانویه با بعد کمتر منتقل می‌شود. با انجام این کار، به طور غیرمستقیم، زمان یادگیری و آزمایش در الگوریتم‌های یادگیری گروهی کاهش می‌یابد، به علاوه نیازمندی سیستم موقعیت‌یاب به سختافزار قوی برای ذخیره و یا انتقال به حافظه موقت و پردازش داده حجیم نیز برطرف می‌شود.

راهکار پیشنهادی دوم، تعیین تابع هزینه و استفاده از الگوریتم ژنتیک<sup>۱</sup> برای به‌دست‌آوردن مقادیر بهینه ابرپارامترها است. تابع هزینه برای منظور کردن پارامترهای ارزیابی، نظیر دقت، سرعت و محدودیت‌های سختافزاری تعیین می‌شود. الگوریتم ژنتیک بر اساس اصول تکامل و انتخاب طبیعی عمل می‌کند و با تولید و ترکیب جمعیت از ابرپارامترها، به دنبال یافتن مقادیر بهینه می‌گردد. این رویکرد امکان تنظیم هوشمندانه پارامترها را فراهم می‌آورد و به سیستم اجازه می‌دهد تا با تغییرات در محیط یا وظایف مختلف، به بهینه‌ترین تنظیمات ممکن برسد. با استفاده از این راهکار، امکان بهبود عملکرد سیستم در مقابل چالش‌ها و محدودیت‌های مختلف استفاده از الگوریتم‌های یادگیری گروهی فراهم می‌شود و بهره‌وری سیستم افزایش می‌یابد.

## ۱-۵- مروری بر فصل‌های پایان‌نامه

در فصل دوم این پایان‌نامه ابتدا مفاهیم و تعاریف اولیه موقعیت‌یابی داخلی بیان می‌شود. پس از آن انواع دادگان در تخمین موقعیت و فناوری‌های مورداستفاده در آن معرفی می‌شود. سپس الگوریتم‌های پایه و الگوریتم‌های یادگیری ماشین بررسی می‌شود و در انتهای این فصل به چالش‌های این روش‌ها پرداخته می‌شود. در فصل سوم راهکارهای پیشنهادی برای بهبود دقت و عملکرد موقعیت‌یابی با استفاده از روش‌های یادگیری گروهی ارائه شده است. در فصل چهارم این راهکارهای پیشنهادی بر روی یک مجموعه‌دادگان موقعیت‌یابی پیاده‌سازی شده است و نتایج آن بررسی و با سایر پژوهش‌ها مقایسه شده است. در نهایت در فصل پنجم، جمع‌بندی از مطالب این پایان‌نامه صورت گرفته و پیشنهاداتی برای ادامه این کار در آینده عنوان شده است.

<sup>1</sup> Genetic Algorithm

## **فصل ٢:**

# **مفاهيم و تعاريف**

## ۱-۱- مقدمه

هدف از این فصل آشنایی با تعاریف و مفاهیم موقعیت‌یابی و ارائه انواع داده‌های مورداستفاده در تعیین موقعیت اشیا در مکان‌های سرپوشیده است. استفاده از هر کدام از انواع دادگان بسته به کاربرد و تکنولوژی مورداستفاده می‌تواند مزایا و محدودیت‌هایی داشته باشد و بر اساس معیارهای ارزیابی انتخاب می‌گردد. همچنین در این فصل به بررسی فناوری‌های مورد استقبال در موقعیت‌یابی مبتنی بر RSS پرداخته شده است و چالش‌های موقعیت‌یابی مبتنی بر RSS ارائه می‌گردد.

در این فصل الگوریتم‌هایی که با استفاده از دادگان موقعیت‌یابی، محاسبه موقعیت را انجام می‌دهد، به دو بخش الگوریتم‌های پایه و یادگیری ماشین تقسیم‌بندی شده‌اند. الگوریتم‌های پایه در حوزه محاسبه موقعیت از روابط ریاضیاتی سنتی برای محاسبه فاصله و موقعیت استفاده می‌کنند. این الگوریتم‌ها عمدهاً بر اساس اصول هندسی عمل می‌کنند و معمولاً برای محاسبات در محیط‌های کم پیچیده و با داده‌های موقعیت دقیق و کمترین نویز مناسب هستند. در مقابل الگوریتم‌های یادگیری ماشین با استفاده از داده‌های موقعیت‌یابی به صورت خودکار مدل‌های پیچیده‌تری را ایجاد می‌کنند. در انتهای این فصل به این دو بخش پرداخته می‌شود.

## ۲-۱- مفاهیم و تعاریف موقعیت‌یابی

برخی تعاریف و مفاهیم اولیه در موقعیت‌یابی در زیر ذکر شده است.

### ۲-۱-۱- خدمات مبتنی بر مکان

خدمات مبتنی بر مکان (LBS) به خدماتی اشاره دارد که به ادغام موقعیت جغرافیایی (مختصات فیزیکی) با مفهوم عام خدمت اشاره می‌کند. مانند خدمات اضطراری، سیستم‌های ناوبری خودرو، خدمات توریستی.

### ۲-۱-۲- سیستم تعیین موقعیت داخلی

سیستم تعیین موقعیت داخلی (IPS)، سیستمی است که به طور بلادرنگ و پیوسته می‌تواند موقعیت اجسام یا افراد را در یک فضای فیزیکی سرپوشیده مشخص کند. بر اساس این تعریف، این سیستم‌ها باید در همه

زمان‌ها موقعیت را به روز کند.

### ۳-۲-۲- موقعیت<sup>۱</sup> و مکان<sup>۲</sup>

موقعیت به مختصات یک نقطه مشخص در سیستم مختصاتی مانند عرض و طول و ارتفاع اشاره دارد. مکان، موقعیت را می‌دهد، اما در چهارچوب نقطه خاص، مانند قرارگرفتن یک مغازه در یک پاساژ. مکان، نسبی است و می‌تواند نسبت به یک جسم متحرک و یا ثابت بیان شود.

### ۴-۲-۲- نقطه دسترسی<sup>۳</sup>

یک نقطه دسترسی (AP) در سیستم موقعیت‌یابی با فناوری Wi-Fi به دستگاهی در شبکه با موقعیت معلوم و ثابت گفته می‌شود که ارتباط بی‌سیم با دستگاه کاربر (موقعیت نامعلوم) را برقرار می‌سازد. در سیستم موقعیت‌یابی، مکان دقیق نقاط دسترسی، مشخص است.

### ۳-۲- انواع داده در موقعیت‌یابی

داده در موقعیت‌یابی به مجموعه‌ای از اطلاعاتی اشاره دارد که به منظور محاسبه موقعیت استفاده می‌شود. داده‌های موقعیت‌یابی عمدهاً اطلاعاتی است که از سیگنال‌های فناوری‌های بی‌سیم استخراج می‌گردد. داده‌های فناوری‌های غیر سیگنال‌های مخابراتی بی‌سیم مانند سنسورهای شتاب‌سنج و انکودرها نیز در موقعیت به کار برده می‌شود. سایر داده‌ها نیز وجود دارد که با سنجش فیزیکی سایر پدیده‌ها به دست می‌آید، داده‌های خروجی از نشانگرهای تصویری<sup>۴</sup>، سنسورهای مادون‌قرمز، لیزر، لیدار<sup>۵</sup>، سونار<sup>۶</sup> و سنسورهای فراصوت نیز داده‌هایی است که در موقعیت‌یابی استفاده شده است [۱۷]. داده‌ها می‌توانند با هم ترکیب شوند و مجموعه داده غنی‌تری از موقعیت را بسازند. شکل (۱-۲) داده‌های اندازه‌گیری سیگنال مبتنی بر فاصله و جهت را برای موقعیت‌یابی

<sup>1</sup> Position

<sup>2</sup> Location

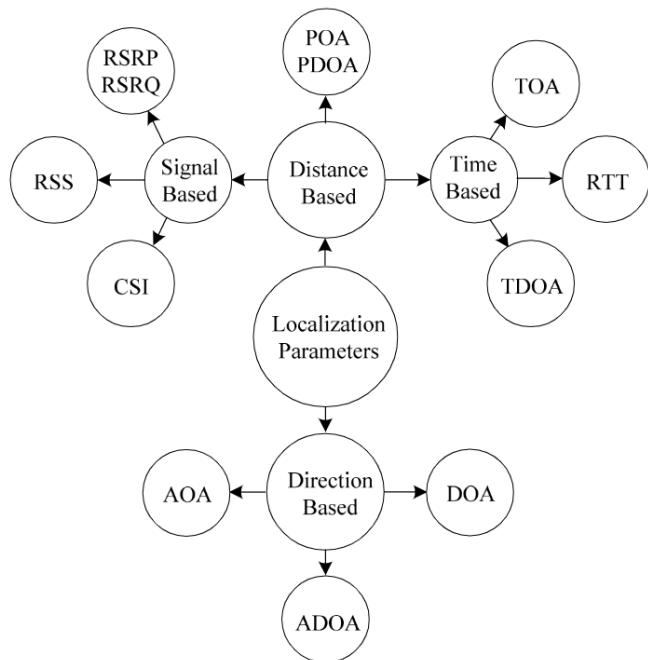
<sup>3</sup> Access Point

<sup>4</sup> Image Markers

<sup>5</sup> Lidar

<sup>6</sup> Sonar

مکان سرپوشیده نشان می‌دهد. در ادامه این بخش، به برخی از انواع داده‌های دریافتی از فناوری‌های بی‌سیم پرداخته می‌شود و چالش‌های هر یک عنوان خواهد شد.



شکل (۱-۲) داده‌های سیستم موقعیت‌یابی برای اندازه‌گیری‌های فاصله و جهت [۳]

### ۱-۳-۲- قدرت سیگنال دریافتی (RSS)

یکی از معیارهایی که برای استخراج فواصل، در عین دقت پایین، بسیار مورداستفاده قرار می‌گیرد، قدرت سیگنال دریافتی است که تقریباً از تمام سیگنال‌های مخابراتی مانند WiFi، GSM، Bluetooth و ... استخراج می‌شود. می‌توان بر اساس قدرت سیگنال دریافتی فاصله بین گیرنده و فرستنده را تخمین زد. RSS عموماً با واحد (mW) یا دسی‌بل میلی‌وات اندازه‌گیری می‌شود و هر چقدر فاصله بین فرستنده و گیرنده کم باشد، میزان قدرت سیگنال دریافتی RSS بیشتر خواهد بود.

RSSI عموماً واحد ساده‌ای برای اندازه‌گیری است، اما به خاطر اثرات محوش‌گی<sup>۱</sup>، پنهان‌شدن<sup>۲</sup>، انعکاس و پراکندگی<sup>۳</sup>، بسیار نادقيق است. از این‌رو، استفاده از فیلترهایی، مانند فیلتر کالمن توسعه‌یافته (EKF) و سایر روش‌های یادگیری ماشین برای کاهش نوسانات RSSI به کاررفته است. برای محاسبه فاصله با RSSI از مدل افت لگاریتمی استفاده می‌شود. مدل افت لگاریتمی چندین متغیر را در دسترس قرار می‌دهد تا بتوان با توجه به

<sup>1</sup> fading

<sup>2</sup> shadowing

<sup>3</sup> scattering

شرایط متفاوت مکانی آن‌ها را تنظیم نمود. مدل انتشار از طریق رابطه (۱-۲) بیان می‌شود:

$$\text{RSSI} = P_0 - 10n\log\left(\frac{d}{d_0}\right) + X_{\sigma} \quad (1-2)$$

که در این رابطه RSSI نشانگر توان سیگنال دریافتی در فاصله  $d$ ،  $P_0$  توان سیگنال دریافتی در فاصله  $d_0$  و  $n$  یک فاصله مشخص از مرجع است.  $n$  ضریب تلفات توان یا ضریب محیط و  $X_{\sigma}$  متغیر تصادفی گوسی میانگین صفر است. در این پژوهش به خاطر سادگی در محاسبات RSS و دسترسی‌پذیری آن از RSS استفاده شده است. چراکه برای کاربردهای عمومی نظری موقعیت یابی در فروشگاه بدون نیاز به تجهیزات و تنها با استفاده از تلفن همراه کاربر موقعیت یابی را انجام داد.

### ۲-۳-۲- زمان ورود (ToA)

زمان ورود (ToA) که همچنان به اختصار ToF شناخته می‌شود، اولین دوره‌ای که در آن سیگنال به گیرنده می‌رسد توصیف می‌شود. طرح‌های مرسوم ToA به حداقل دو یا سه گره مرجع به صورت خط دید باهدف نیازمند است. برای رسیدن به دقت بالاتر در موقعیت، گره‌ها می‌توانند همگام<sup>۱</sup> و یا غیرهمگام در زمان رسیدن باشند؛ اما سیگنال باید دارای نشان زمانی<sup>۲</sup> باشد. برای حل این موضوع روش TDoA که به روش زمان ورود رفت و برگشتی<sup>۳</sup> (RToA) نیز شناخته می‌شود معرفی و پیاده‌سازی شد. مکانیزم حوزه ToA مشابه TDoA مشابه است، اما نیازی به مرجع زمانی متحدد در گره‌ها ندارد. ToA از چندمسیرگی و نویز جمع‌شونده اثر می‌پذیرد، نویز جمع‌شونده در دقت زمان ورود سیگنال اثر می‌گذارد [۳].

### ۲-۳-۳- زمان تفاضلی رسیدن (TDoA)

TDoA تکنیکی برای محاسبه اطلاعات فاصله بین دو گره است. TDoA واریانس زمان رسیدن (نشان زمانی) بین گره‌های لنگر<sup>۴</sup> در یک بسته ارسالی یکسان از هدف را تعیین می‌کند. این روش به حداقل سه گره لنگر با مختصات معلوم برای یافتن مختصات موقعیت شیء هدف نیاز دارد. گره‌های لنگر به ارسال‌های هدف را دریافت

<sup>1</sup> Synchronized

<sup>2</sup> Timestamp

<sup>3</sup> Round Trip of Arrival

<sup>4</sup> Anchors

می‌کند و موقعیت تخمین زده شده را با مقایسه واریانس در زمان رسیدن محاسبه می‌کند. این روش با تعیین تغییرات زمانی بین چند گره لنگر عمل می‌کند [۳].

دقت تخمین TDoA به پنهانی باند سیگنال، نرخ نمونه‌برداری گیرنده وجود خط دید بین فرستنده و گیرنده بستگی دارد. همچنین به هم‌زمان بودن سخت‌گیرانه نیازمند است، اما برخلاف ToA که هم‌زمانی بین گیرنده و فرستنده نیاز بود، در TDoA تنها به هم‌زمانی بین فرستنده‌ها نیاز است [۱۸].

### ۴-۳-۲- زاویه رسیدن (AoA) و زاویه تفاضلی رسیدن (ADoA)

AoA تکنیکی برای تعیین موقعیت اشیا با گرفتن داده‌های زاویه‌ای جهت قرارگیری شیء و گیرنده‌ها است. یک محاسبه ساده از AoA را می‌توان در کارکرد آنتن‌های آرایه‌ای مشاهده کرد. AoA نیز توسط نویز، خط غیر دید و چندمسیرگی مختلف می‌شود، همچنین نقص در LoS می‌تواند جدی‌تر از روش‌های مبتنی بر TDoA یا RSS باشد. AoA به فضای بیشتر و سخت‌افزارهای اضافه نیازمند است که اتلاف انرژی بالاتری دارد، اما نیاز به هم‌زمان‌سازی بین گره وجود ندارد [۳].

ADoA به اطلاعات زاویه‌ها نیاز ندارد؛ زیرا می‌تواند آن را در واریانس بین دو مقدار AoA نادیده گرفت. این بدان معناست که گیرنده‌ها باید در یک زاویه مشخص قرار بگیرند [۳].

### ۵-۳-۲- فاز رسیدن (PoA) و فاز تفاضلی رسیدن (PDoA)

روش PoA برای تخمین فاصله با اندازه‌گیری فاز انتقال‌دهنده سیگنال است. همچنین با نام فاز سیگنال دریافتی<sup>۱</sup> (RSP) نیز شناخته می‌شود. تعداد قابل توجهی از اندازه‌گیری‌های PoA در سیستم‌های موقعیت‌یابی RFID وجود دارد. روش‌های مبتنی بر PoA می‌توانند با سایر تکنیک‌ها نظیر RSS، ToA و TDOfA تجمعی شود. برای بالارفتن دقت در PoA ممکن است نیاز به خط دید داشته باشد. همچنین ابهامات غیر قابل حذفی ممکن است در طول ارزیابی فاصله صحیح به خاطر چندمسیرگی و تناوب فاز  $2\pi$  به وجود آید [۳].

<sup>1</sup> Received Signal Phase

## ۳-۶- اطلاعات حالت کانال (CSI)

روش مبتنی بر CSI از اطلاعات وضعیت کانال لایه فیزیکی یک پیوند ارتباطی استفاده می‌کند. هنگامی که یک هدف در مکان سرپوشیده ظاهر می‌شود، CSI مربوط به آن قابل اندازه‌گیری خواهد بود [۳].

روش مبتنی بر CSI پایداری خوبی را نشان داده است و می‌تواند به دقت مکان بالاتری نسبت به روش مبتنی بر RSSI دست یابد. از طرفی دیگر، طرح Wi-Fi مبتنی بر CSI نیازمند کار فشرده میدانی بیشتر برای کالیبره کردن مکان نقاط دسترسی و جهت آنتن‌های آرایه‌ای است که مانع اجرای آن در دنیای واقعی می‌شود. یکی دیگر از معایب CSI این است که در CSI به فضای بزرگ‌تر و زمان بیشتری نیازمند است که برای اکثر موقعیت‌ها مناسب نیست [۳].

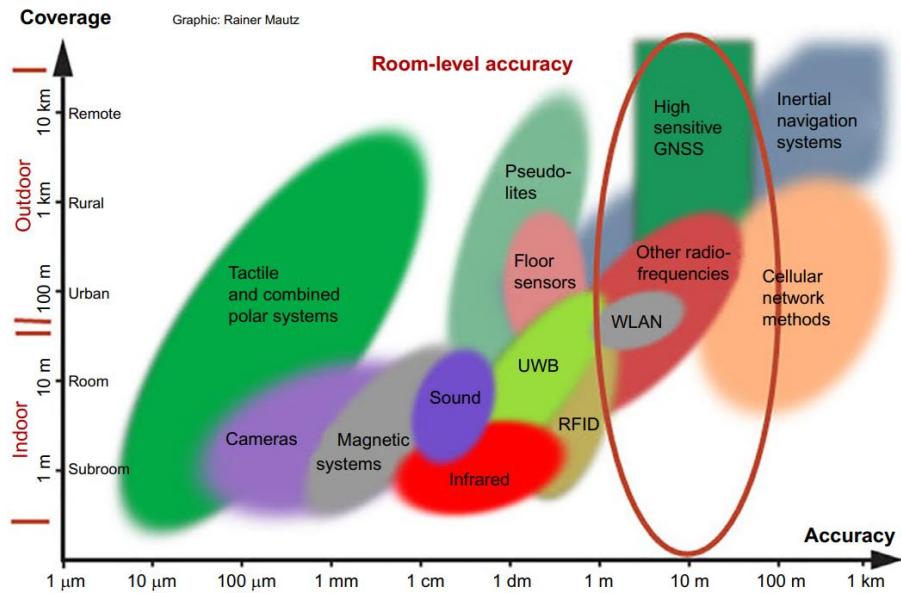
## ۷-۳-۲- قدرت و کیفیت سیگنال مرجع دریافتی (RSRQ و RSRP)

دو پارامتر RSRQ و RSRP، پارامترهای داده لایه فیزیکی شبکه 4G است که برای موقعیت کاربر به کار می‌رود. محاسبه RSRP مبتنی بر RSSI است که میانگین دستاوردهای قدرت سیگنال‌های مرجع سلول مشخص را محاسبه می‌کند.

پارامتر RSRQ که مقدار سیگنال‌های دریافتی را در دستگاه هدف ارائه می‌کند و از مقدار RSSI و RSRP توسعه یافته است. RSRQ از تداخل ایستگاه‌ها و نویز گرمایی اثر می‌پذیرد و دقت واریانسی کمتری نسبت به RSRP دارد. اما دقت میانگینی آن از RSRP بیشتر است؛ لذا ترکیب این دو، دقت کلی را افزایش می‌دهد [۳].

## ۴-۲- فناوری‌های موقعیت‌یابی مبتنی بر RSS و چالش‌ها

سیگنال‌های رادیویی به طور گسترده در سیستم‌های موقعیت‌یابی داخلی به کار می‌روند. یکی از روش‌های رایج برای موقعیت‌یابی در فضاهای داخلی، استفاده از اطلاعات قدرت سیگنال دریافتی از فناوری‌های ارتباط بی‌سیم است. شکل (۲-۲) نشان دهنده برد انواع تکنولوژی‌های بی‌سیم بر اساس دقت اندازه‌گیری فاصله است. این بخش به بررسی فناوری‌های رادیویی مورد استفاده در سیستم‌های موقعیت‌یابی بر پایه قدرت سیگنال دریافتی (RSS) می‌پردازد و چالش‌ها و پیشرفت‌های اخیر در این زمینه را مورد بررسی قرار می‌دهد.



شکل (۲-۲) محدوده دقت فناوری‌های موقعیت‌یابی بر حسب استفاده در داخل یا خارج ساختمان [۱۹]

## Wi-Fi - ۲-۴-۲

Wi-Fi یک فناوری شبکه محلی بی‌سیم<sup>۱</sup> (WLAN) است که به عنوان یک فناوری ارتباطی میان‌بند<sup>۲</sup> بسیار مشهور و پرکاربرد است. این شبکه از طریق دستگاه‌های مختلف نظری لپ‌تاپ، تبلت، تلفن همراه و سایر دستگاه‌های هوشمند قابل دسترسی است که در نتیجه می‌تواند یک بستر ساده و در دسترس برای موقعیت‌یابی داخلی فراهم کند. Wi-Fi به دلیل نصب آسان نقاط دسترسی در مکان‌های مختلف، هزینه کمتر نسبت به فناوری‌های دیگر بی‌سیم داخلی، مصرف کم‌انرژی و عدم نیاز به سخت‌افزار اضافی برای موقعیت‌یابی و ناوبری در محیط‌های سرپوشیده مورد استفاده قرار می‌گیرد.

روش‌های اثر انگشت و مثلث‌بندی معمولاً در موقعیت‌یابی داخلی مبتنی بر Wi-Fi استفاده می‌شوند. الگوریتم‌های موقعیت‌یابی اثر انگشت به دلیل پیاده‌سازی ساده، پیچیدگی کم، و عدم نیاز به سخت‌افزار خاص و اندازه‌گیری خط دید نقاط دسترسی، بهترین عملکرد را ارائه می‌دهند. میانگین خطای موقعیت‌یابی مبتنی بر Wi-Fi تا ۳ متر اعلام شده است. پیاده‌سازی موقعیت‌یابی مبتنی بر Wi-Fi تحت تأثیر اثرات محیطی نظری موانع، تحرک افراد و غیره قرار می‌گیرد. چندمسیره شدن سیگنال‌های Wi-Fi می‌تواند بر RSS سیگنال‌های متغیر با زمان تأثیر گذارد و در نتیجه بر دقت موقعیت‌یابی با فناوری Wi-Fi تأثیر بگذارد.

<sup>1</sup> Wireless local area network

<sup>2</sup> Broadband

**Bluetooth - ۳-۴-۲**

بلوتوث کم انرژی (BLE)، فناوری امروزی است که در بیشتر دستگاه‌های هوشمند پشتیبانی می‌شود و در محدوده باند ۲.۴ گیگاهرتز تا ۲.۴۸۳۵ گیگاهرتز عمل می‌کند. BLE برای انتقال بی‌سیم با برد کوتاه طراحی شده و در مقایسه با سیستم‌های مبتنی بر Wi-Fi، دارای خطاهای کمتری است. طول سنجش بلوتوث تا ۱۰ متر است بنابراین مناسب برای فضای کوچک است. RSS در دستگاه‌های بلوتوث تا ۲۰ میلی‌متر تغییر کند. سیستم مبتنی بر بلوتوث برای دستیابی به دقت‌های تا ۱.۲ متر مناسب است و در موقعیت‌یابی داخلی و اینترنت اشیا به دلیل هزینه کم و توان کم استفاده می‌شود. این سیستم‌ها اغلب از الگوریتم‌های تشخیص مجاورت، مثلث‌بندی و اثر انگشت با استفاده از مشخصه RSS استفاده می‌کنند. با این حال، دقت موقعیت‌یابی تحت تأثیر پایداری گره‌ها و محیط انتشار داخلی قرار دارد و نیازمند تحقیقات بیشتر است.

**RFID - ۴-۴-۲**

فناوری RFID یک فناوری بی‌سیم بدون تماس است که توانایی شناسایی خودکار را از طریق ارسال اطلاعات از تگ به خوانشگر آن با استفاده از سیگنال الکترومغناطیسی دارد. این فناوری از خوانشگران، تگ‌ها و یک کامپیوتر تشکیل شده است. تگ‌های فعال محدوده تشخیص وسیع‌تری دارند؛ اما انرژی بیشتری مصرف می‌کنند. تگ‌های غیرفعال برای فضای کوچک و مکان‌های ثابت مناسب هستند.

سیستم‌های موقعیت‌یابی مبتنی بر RFID معمولاً از ویژگی‌های سیگنال‌های رادیویی مانند قدرت سیگنال، زمان ارسال و جهت استفاده می‌کنند. برای طرح‌های موقعیت‌یابی فرکانس فوق بالا مبتنی بر RSSI سیگنال‌های توان دریافتی توسط خوانشگران بر مبنای RSSI محاسبه می‌شود. روش‌های مبتنی بر شامل روش‌های مبتنی بر تگ مرجع و روش‌های مبتنی بر فاصله است.

**ZigBee - ۵-۴-۲**

فناوری ZigBee یک فناوری بی‌سیم با بُرد کوتاه است که در فرکانس ۲.۴ گیگاهرتز فعالیت می‌کند. دستگاه‌های ZigBee از مشخصه‌های انرژی و پیوند برای اندازه‌گیری RSS استفاده می‌کنند. ZigBee با بُرد وسیع‌تری نسبت به بلوتوث کم انرژی، ارتباطات گسترشده‌تری را فراهم می‌کند و در سیستم‌های موقعیت‌یابی از

نشان کیفیت ارتباط<sup>۱</sup> (LQI) به جای RSSI استفاده می‌شود. موقعیت‌یابی ZigBee محدودیت‌های مشخصی در موقعیت‌یابی بلادرنگ در هنگام استفاده از RSSI دارد. این محدودیت به خاطر برد کوتاه و تأخیر زیاد این فناوری بی‌سیم است.

## Cellular - ۶-۴-۲

در سیستم‌های موقعیت‌یابی، سیگنال‌های بی‌سیم مانند 2G، 3G، 4G و 5G استفاده می‌شوند. در سیستم‌های LTE 4G، از مقادیر RSRP و RSRQ به عنوان معیارهای قدرت سیگنال استفاده می‌شود. تغییرات در پراکندگی سیگنال‌های رادیویی با تغییر مکان ایستگاه‌های پایه متغیر است، اما ترکیب کانال‌های رادیویی می‌تواند وابستگی موقعیت‌یابی را به یک کانال خاص محدود کند و از اشتباهات موقعیت‌یابی جلوگیری کند.

## 7-۴-۲- چالش‌های موقعیت‌یابی مبتنی بر RSS

در این بخش به چالش‌های موقعیت‌یابی مبتنی بر RSS در مکان‌های داخلی پرداخته شده و تعاریفی از این چالش‌ها گفته شده است.

### ■ پدیده چندمسیره شدن

در محیط‌های سرپوشیده، پدیده چندمسیره شدن اتفاق می‌افتد که ناشی از تداخل سیگنال با سطوح موانع و بازتاب از آن‌ها است. این پدیده موجب افت توان سیگنال می‌شود و پیش‌بینی آن دشوار است. در شکل (۳-۲) پدیده چندمسیره شدن نشان داده شده است.

### ■ تداخل سیگنال‌ها

در محیط سرپوشیده، وجود منابع تداخلی می‌تواند بر مقادیر RSS اثر بگذارد و تداخل سیگنال را ایجاد کند. این تداخل ممکن است باعث خوانش متناقض مقادیر RSS شده و در نتیجه، دقت موقعیت‌یابی را کاهش دهد.

### ■ تغییر قدرت سیگنال

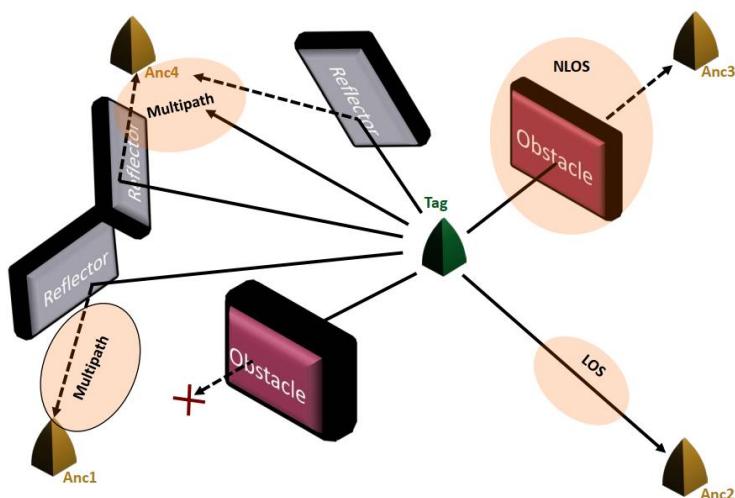
تغییرات در قدرت سیگنال RSS در یک مکان ممکن است ناشی از عوامل مختلفی مانند تغییرات محیط،

<sup>۱</sup> Link Quality Index

حرکت افراد یا اشیا، تداخل دستگاه‌های بی‌سیم و نقص سنسور باشد. این تغییرات ممکن است تأثیر زیادی بر دقت و قابلیت اطمینان سیستم‌های موقعیت‌یابی داخلی مبتنی بر RSS داشته باشد.

### □ شرایط خط غیر دید

در مکان‌های سرپوشیده، خط دید (LoS) میان منبع و هدف زمانی ایجاد می‌شود که مسیر آزاد است و خط غیر دید (NLoS) هنگامی رخ می‌دهد که مسیر با موانع مسدود شده است. این شرایط، یکی از چالش‌های اساسی موقعیت‌یابی در مکان‌های سرپوشیده است و تاثیر قابل توجهی بر تغییرات قدرت سیگнал دارد. در شکل (۳-۲) شرایط خط دید، خط غیر دید نمایش داده شده است.



شکل (۳-۲) طرح گرافیکی از چندمسیره شدن، خط غیر دید و خط دید در موقعیت‌یابی داخلی

### □ تغییرات دینامیکی محیط انتشار سیگنال

روش اثر انگشت، نقشه رادیویی را برون خط ایجاد و در پایگاهداده ذخیره می‌کند. در مرحله برخط از داده‌های ذخیره شده در پایگاهداده استفاده شده و موقعیت‌یابی هدف انجام می‌شود. اما تغییرات دینامیکی محیط، مانند جایه‌جایی اشیای ثابت و تغییرات در حضور افراد و تعداد آن‌ها، ممکن است قابلیت اعتبار داده‌های مرحله برون خط را کاهش دهد.

### □ کالیبراسیون و نگهداری

تغییرات محیط و فرسودگی دستگاه‌ها در مکان‌های سرپوشیده موقعیت‌یابی را تحت تأثیر قرار می‌دهد. اجرای متناسب برنامه کالیبراسیون و نگهداری دستگاه‌ها برای حفظ دقت موقعیت‌یابی نیازمند وقت، نیروی انسانی و هزینه بالا است.

## ۲-۵- روشهای پایه در موقعیت‌یابی

الگوریتم‌های پایه، در موقعیت‌یابی، با بهره‌گیری از روابط ریاضیاتی و هندسی، امکان تخمین فواصل و موقعیت اشیا را در فضای داخلی فراهم می‌کنند. این روشهای پایه اصول هندسه عمل کرده و تلاش می‌کنند تا موقعیت اشیا را تعیین کنند.

الگوریتم‌های پایه، با وجود مزایایی آن مانند سادگی و محاسبات ارزان، دارای برخی محدودیت‌ها نیز هستند. یکی از این محدودیت‌ها، این است که این الگوریتم‌ها به داده‌های موقعیت دقیق و کمترین نویز نیاز دارند. در محیط‌های پیچیده، مانند محیط‌های داخلی، به دلیل وجود موانع و سایر عوامل، داده‌های موقعیت ممکن است دقیق نباشند. همچنین، در این محیط‌ها، نویز نیز ممکن است بیشتر باشد. در این شرایط، عملکرد الگوریتم‌های پایه ممکن است کاهش یابد.

برای غلبه بر این محدودیت‌ها، الگوریتم‌های پیشرفته‌تری توسعه یافته‌اند. این الگوریتم‌ها، با استفاده از روشهای پیچیده‌تر، می‌توانند دقت موقعیت‌یابی را در محیط‌های پیچیده بهبود بخشنند. در ادامه، برخی از محبوب‌ترین روشهای پایه در موقعیت‌یابی معرفی شده است.

### ۲-۱- الگوریتم مجاورت

یکی از ساده‌ترین تکنیک‌های تخمین موقعیت، تکنیک تخمین مجاورت است. در این روش، موقعیت شیء هدف، زمانی که به یک موقعیت معین نزدیک می‌شود، تخمین زده می‌شود و موقعیت دقیق مختصاتی را محاسبه نمی‌کند. تخمین مجاورت به دلیل سادگی و سرعت بالای اجرا، در کاربردهایی که دقت بالایی موردنیاز نیست، می‌تواند گزینه مناسبی باشد.

### ۲-۲- الگوریتم مثلث‌بندی<sup>۱</sup>

روش مثلث‌بندی یک روش موقعیت‌یابی مبتنی بر فاصله است که از خواص هندسی مثلثات برای محاسبه مکان کاربر استفاده می‌کند. این روش از نقطه همپوشانی سه دایره که توسط سه نقطه مرجع تشکیل می‌شود، برای تعیین موقعیت استفاده می‌کند. فواصل بین کاربر و نقاط مرجع از طریق روشهای مختلف اندازه‌گیری

<sup>1</sup> Triangulation

سیگنال مانند RSS، ToA، TDoA و AoA تخمین زده می‌شود. روش مثلث‌بندی قادر به محاسبه دقیق مکان است؛ اما به دقت و صحت تخمین فاصله حساس است.

### ۲-۵-۳- الگوریتم چندوجهی‌بندی<sup>۱</sup>

الگوریتم چندوجهی‌بندی توسعه‌یافته الگوریتم مثلث‌بندی با بیش از سه نقطه مرجع در تخمین موقعیت هدف است. چندوجهی سازی رایج‌ترین روش برای استخراج موقعیت است. معادلات سیستم از فواصل تخمین زده شده لنگرها به صورت زیر به دست می‌آید.

$$\begin{aligned} A &= \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix}, \\ b &= \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix} \end{aligned} \quad (۲-۲)$$

که در آن موقعیت هدف با  $(x, y)$  و  $(x_n, y_n)$  مختصات معلوم نقاط مرجع و  $d_n$  فاصله نقطه هدف تا نقاط مرجع نشان داده می‌شود. با حل سیستم با استفاده از رویکرد استاندارد حداقل مربعات:

$$\hat{x} = (A^T A)^{-1} A^T b. \quad (۳-۲)$$

که نماد  $\hat{x}$  موقعیت تخمین زده شده را بیان می‌کند.

### ۲-۵-۴- الگوریتم Min-Max

الگوریتم Min-Max یک روش موقعیت‌یابی مبتنی بر برد است که از فضای جعبه (مربع) برای تعیین موقعیت استفاده می‌کند. این الگوریتم ابتدا برای هر گره مرجع، یک مربع با فاصله دوبرابر فاصله تخمین زده شده از گره هدف ترسیم می‌کند. سپس، همپوشانی این مربع‌ها مشخص می‌شود. موقعیت گره هدف در مرکز جعبه (مربع) قرار می‌گیرد. تقاطع جعبه‌های مرزی با گرفتن بیشینه همه کمینه‌ها و کمینه همه بیشینه‌ها محاسبه می‌شود.

<sup>1</sup> Multilateration

## ۲-۵-۵-الگوریتم حداکثر درستنمایی<sup>۱</sup>

الگوریتم حداکثر درستنمایی بر مبنای اصل استنتاج آماری مرسوم است. این روش با حداقل رساندن واریانس خطای فاصله تخمین زده است مکان گره هدف را حدس می‌زند. این تقریب می‌تواند با معیار حداقل میانگین مربعات خطأ<sup>۲</sup> (MMSE) پیاده شود. با این حال، عملکرد این روش با توجه به تعداد گره‌های مرجع ناپایدار است. موقعیت طبق این الگوریتم به صورت زیر محاسبه می‌شود:

$$\begin{aligned} X &= \begin{bmatrix} 2(x_k - x_1) & 2(y_k - y_1) \\ \vdots & \vdots \\ 2(x_k - x_{k-1}) & 2(y_k - y_{k-1}) \end{bmatrix}, \\ y &= \begin{bmatrix} -x_1^2 - y_1^2 - d_1^2 - (x_k^2 - y_k^2 + d_k^2) \\ \vdots \\ -x_{k-1}^2 - y_{k-1}^2 + d_{k-1}^2 - (x_k^2 - y_k^2 + d_k^2) \end{bmatrix}. \end{aligned} \quad (4-2)$$

در معادله (۴-۲)،  $(x_0, y_0) = b$  موقعیت ناشناخته گره هدف است و  $(x_i, y_i)$  موقعیت  $i$  امین گره مرجع است. این الگوریتم با تخمین MMSE موقعیت هدف، یعنی  $(x_0, y_0) = b$  را تخمین می‌زند:

$$b = (X^T X)^{-1} X^T y \quad (5-2)$$

## ۲-۵-۶-الگوریتم موقعیت یابی اثر انگشت<sup>۳</sup>

الگوریتم یابی اثر انگشت یک روش موقعیت یابی مبتنی بر داده‌های بدون استفاده از فاصله در ساختمان‌ها است. این الگوریتم از دو مرحله تشکیل شده است:

- ۱) مرحله آموزش آفلاین
- ۲) مرحله آزمایش آنلاین

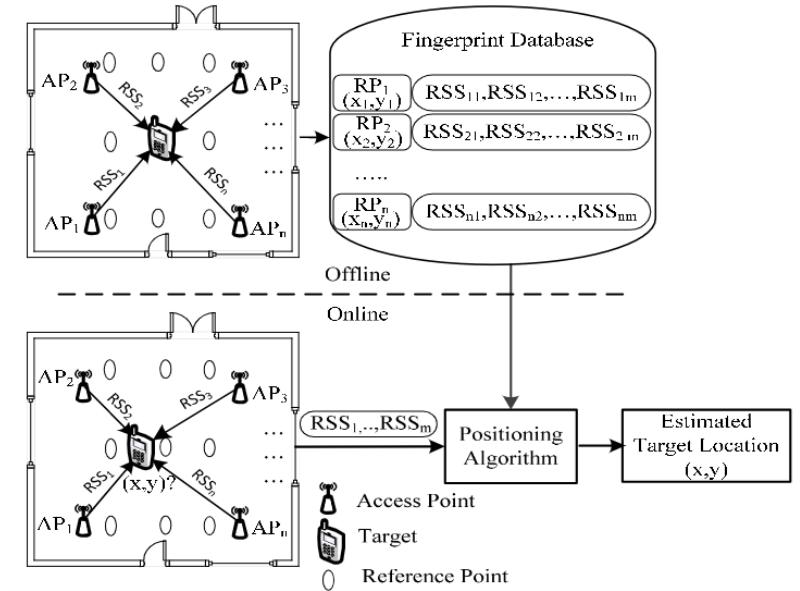
در مرحله آموزش آفلاین، داده‌های CSI یا RSSI از نقاط دسترسی برای گره‌های مرجع متفاوت جمع‌آوری می‌شود. این داده‌ها برای ایجاد یک نقشه رادیویی استفاده می‌شوند. در مرحله آزمایش آنلاین، موقعیت بلاذرنگ با مقایسه اطلاعات اندازه‌گیری‌ها در نقاط دسترسی برای تخمین موقعیت هدف استفاده می‌شود. از جمله

<sup>1</sup> Maximum Likelihood

<sup>2</sup> Minimum mean square error

<sup>3</sup> Fingerprinting

مزایای این الگوریتم دقت موقعیت یابی بالا، کاهش پیچیدگی سخت‌افزار و کاهش تأثیر نامطلوب چندمسیره شدن است. عملکرد مقدماتی این روش در شکل (۴-۲) ترسیم شده است.



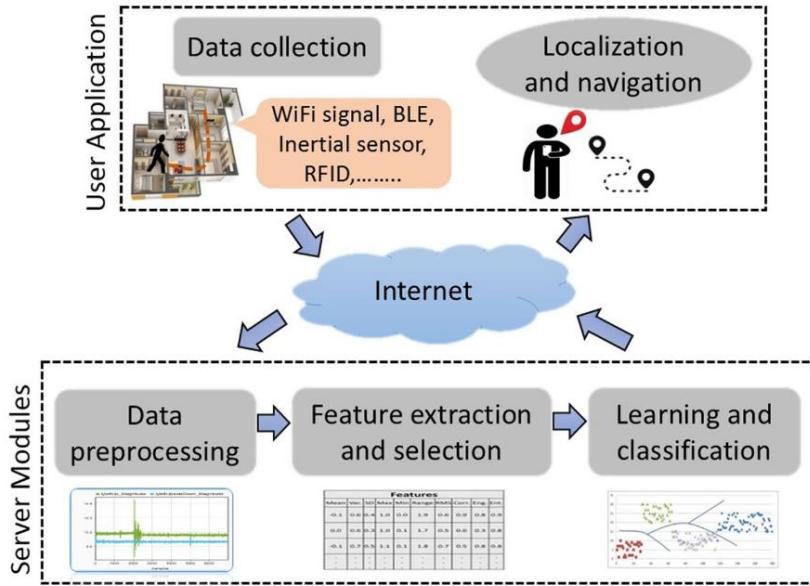
شکل (۴-۲) روش موقعیت یابی مبتنی بر اثر انگشت [۳]

## ۶-۲-الگوریتم‌های یادگیری ماشین

یادگیری ماشین در موقعیت یابی کاربردهای زیادی دارد و نتایج شگرفی در کاهش خطای موقعیت یابی، کاهش هزینه نیرو و زمان و استفاده آسان در دستگاهها داشته است. همچنین از یادگیری ماشین برای تشخیص و رفع چالش‌های موقعیت یابی مکان سرپوشیده مانند خط غیر دید، چندمسیره شدن و غیره استفاده می‌شود. در این فصل، پس زمینه و مقدماتی از یادگیری ماشین در موقعیت یابی بیان می‌شود. پس از آن، چالش‌های موقعیت یابی با الگوریتم‌های یادگیری ماشین ارائه می‌شوند.

## ۶-۲-۱-ساختار موقعیت یابی با یادگیری ماشین

شکل (۵-۲) ساختار کلی موقعیت یابی با الگوریتم‌های یادگیری ماشین نشان می‌دهد که طبق آن ارتباط برنامه کاربر با سرور در بستر اینترنت انجام می‌شود. داده‌های موقعیت یابی کاربر به سرور ارسال می‌شود و پس از پردازش، موقعیت کاربر به برنامه کاربر ارسال می‌شود. در ادامه به جزئیات ساختار پرداخته می‌شود.



شکل (۵-۲) طرح ساختار موقعیت یابی با الگوریتم یادگیری ماشین [۲۰]

## □ جمع آوری داده

در این مرحله، داده موردنیاز از هر نقطه ممکن در همه مکان‌ها یا نقاط از پیش تعريف شده جمع آوری می‌شود. با درنظر گرفتن فناوری‌های موقعیت یابی داخلی، داده‌ها با انواع مختلف مانند RSS در بلوتوث، WiFi، RFID یا مقادیر اندازه‌گیری شده از سنسور اینرسی و غیره می‌تواند جمع آوری شود. سه نوع فرایند جمع آوری داده وجود دارد، مبتنی بر کاربر خاص، جمع‌سپاری، دستگاه‌های خودکار (مانند ربات، پهپاد) [۲۰].

## □ پیش‌پردازش داده‌ها

مجموعه‌داده‌های خام ممکن است دارای مقادیر گم شده، نویز و یا در قالب نامطلوب باشد؛ بنابراین، قبل از ساخت مدل‌های یادگیری ماشین، باید داده‌ها پاک‌سازی و پردازش شوند. این مرحله، پیش‌پردازش داده‌ها نامیده می‌شود و یکی از مهم‌ترین مراحل در ساخت مدل‌های یادگیری ماشین است. در زیر به چند مرحله از پیش‌پردازش داده‌ها اشاره شده است:

- ۱) **درون‌یابی داده گم شده:** هر نمونه از یک مجموعه داده خام ممکن است حاوی تمام ویژگی‌ها نباشد. قبل از ایجاد یک مدل یادگیری ماشین، این ورودی‌های گمشده باید با مقادیری پر شود. یکی از راهبردها برای پر کردن ورودی استفاده از نمونه‌های مجاور است.
- ۲) **فیلتر کردن:** فیلتر کردن فرایندی برای حذف نویزهای ناخواسته است. هدف اصلی این فرایند به حداقل رساندن میانگین مربعات خطای بین الگوهای سیگنال دریافتی و تخمین زده شده است.

(۳) استخراج ویژگی: استخراج ویژگی یک مرحله مهم در یادگیری ماشین است که برای کاهش ابعاد داده‌ها و بهبود عملکرد مدل‌ها استفاده می‌شود. تکنیک استخراج ویژگی به عنوان «کاهش ابعاد» نیز شناخته می‌شود. در استخراج ویژگی، ویژگی‌های غیرضروری از داده‌ها حذف می‌شوند تا مدل یادگیری ماشین بتواند الگوهای مهم را شناسایی کند. این کار می‌تواند دقت مدل را بهبود بخشد و زمان و منابع محاسباتی موردنیاز برای آموزش مدل را کاهش دهد.

(۴) انتخاب ویژگی: ویژگی‌های مهمی که بیشتر به یادگیری یک مدل کمک می‌کند از طریق تکنیک‌های انتخاب ویژگی انتخاب می‌شوند. این تکنیک‌ها باعث کاهش بیش‌برازش، زمان آموزش و پیچیدگی مدل پیش‌بینی‌شده در عین بهبود دقت پیش‌بینی می‌شود. انتخاب ویژگی به‌طور کلی به سه دسته روش فیلترکردن، بسته‌بندی<sup>۱</sup> و جاسازی<sup>۲</sup> کردن تقسیم‌بندی می‌شود.

## ۲-۶-۲- تکنیک‌های تحت نظرارت یادگیری ماشین

این تکنیک‌های یادگیری ماشین با داده‌های برچسب‌گذاری شده سروکار دارند. یعنی در مرحله آموزش و جمع‌آوری داده‌های موقعیت‌یابی داخلی، برچسب‌های معناداری به داده‌های جمع‌آوری‌شده اطلاق می‌شود. برخی از تکنیک‌های تحت نظرارت یادگیری ماشین در ادامه توضیح داده شده است.

### □ k نزدیک‌ترین همسایه (kNN)

الگوریتم k-NN یک الگوریتم یادگیری ماشین مبتنی بر نمونه است که برای حل مسائل طبقه‌بندی و رگرسیون استفاده می‌شود. در این الگوریتم، داده‌های آموزشی به عنوان نمونه‌های آموزشی استفاده می‌شوند. داده جدید با مقایسه آن با k نزدیک‌ترین همسایه‌اش در فضای ویژگی‌ها طبقه‌بندی یا رگرسیون می‌شود.

پیاده‌سازی الگوریتم kNN آسان است و تنها به دو پارامتر مقدار وتابع فاصله (مانند اقلیدسی، مینکفسکی یا منهتن) نیاز دارد. با رشد مجموعه داده، زمان بیشتری برای محاسبه فاصله بین یک نقطه جدید و هر نقطه داده موجود موردنیاز است؛ بنابراین عملکرد kNN به سرعت با رشد مجموعه دادگان کاهش می‌یابد. به علاوه، عدم تعادل داده‌ها مشکلاتی را در kNN ایجاد می‌کند.

<sup>1</sup> Wrapper

<sup>2</sup> Embedded

## □ الگوریتم $K^*$

تفاوت عمدۀ  $K^*$  با سایر الگوریتم‌های مبتنی بر نمونه، استفاده از آنتروپی<sup>۱</sup> برای تعریف فاصله است. آنتروپی با احتمال تبدیل یک نمونه به نمونه دیگر تعیین می‌شود. احتمال تبدیل نمونه جدید به تمام نمونه‌های موجود در یک کلاس، جمع بسته می‌شود. این فرایند برای تمام کلاس‌ها انجام می‌شود و در نهایت کلاسی که بیشترین احتمال را دارد به نمونه جدید اختصاص می‌یابد.

## □ ماشین بردار پشتیبان (SVM)

هدف SVM یافتن یک ابر صفحه یا مجموعه‌ای از ابر صفحه‌ها در  $N$  بعد (که  $N$  تعداد ویژگی‌ها است) برای رده‌بندی نقاط نمونه است. در مورد مسائل رده‌بندی دو کلاسه، هدف SVM یافتن یک ابر صفحه بهینه است که حداکثر حاشیه را بین نقاط داده در هر دو کلاس داشته باشد. مدل‌های SVM در برابر تغییرات داده‌ها پایدار هستند، اما انتخاب هسته مناسب و ذخیره بردارهای پشتیبان مشکل‌ساز است.

## □ شبکه بیزین<sup>۲</sup>

شبکه بیزین یک مدل گرافیکی احتمالی بر اساس قضیه بیز است. شبکه اساساً یک گراف غیر چرخه‌ای جهت‌دار (DAG) است که گره‌های آن توسط متغیرها نشان داده می‌شود و هر یال آن نشان‌دهنده وابستگی شرطی بین دو متغیر است. این الگوریتم با این فرض کار می‌کند که همه ویژگی‌ها از یکدیگر مستقل هستند که در سناریو واقعی این امکان وجود ندارد.

## □ روش بیز ساده<sup>۳</sup>

نوع دیگری از رده‌بندی احتمالی است که بر اساس قضیه بیز با یک فرض ساده، یعنی استقلال میان ویژگی‌ها است؛ بنابراین، فرض اساسی این الگوریتم این است که هر ویژگی سهم مستقل و مساوی در نتیجه موقعیت یابی دارد. مشابه شبکه بیزی، الگوریتم بیز ساده در پیاده‌سازی آسان است و به تعداد کمی از نمونه‌های آموزشی نیاز دارد. با این حال اگر فرض استقلال در میان ویژگی‌ها صادق نباشد، عملکرد بیز ساده بسیار ضعیف است.

## □ درخت تصمیم

این تکنیک از ساختار درختی برای ساخت مدل‌های طبقه‌بندی استفاده می‌کند. در درخت هر گره نشان‌دهنده

<sup>1</sup> Entropy

<sup>2</sup> Bayesian Network

<sup>3</sup> Naïve Bayes

آزمایش یک ویژگی خاص از یک نمونه است. هر شاخه از آن گره با توجه به یکی از نتایج احتمالی به گره یا برگ بعدی متصل می‌شود. گره‌های برگ نتایج را پیش‌بینی می‌کنند. درخت تصمیم تفسیرپذیری ساده‌ای دارد و به راحتی پیاده و اجرا می‌شود. در درخت تصمیم مقادیر گمشده‌ای که اغلب در جمع‌آوری داده مکان سرپوشیده از دست می‌رود مدیریت می‌شود. در پیوست الف شبه کد درخت تصمیم به عنوان یادگیرنده پایه الگوریتم‌های یادگیری گروهی ارائه شده است.

### □ شبکه عصبی مصنوعی<sup>۱</sup> (ANN)

شبکه عصبی از لایه‌های شامل گره‌های به هم پیوسته تشکیل شده است و عمداً شامل یک لایه ورودی، یک یا چند لایه پنهان و یک لایه خروجی است. در طول آموزش، خروجی واقعی با خروجی شبکه مقایسه شده و برای اساس خطای محاسبه می‌شود. سپس خطای از طریق شبکه پس انتشار<sup>۲</sup> می‌شود تا وزن اتصال بین گره‌ها تغییر کند. این فرایند پس انتشار چندین بار تکرار می‌شود تا زمانی که شبکه تنظیم شود و دقت معقولی را در داده‌های ارزیابی ارائه کند. شبکه‌های عصبی می‌توانند روابط غیرخطی و پیچیده بین ویژگی‌ها را تشخیص دهند. همچنین از نظر محاسباتی کارآمد هستند و در عین حال مستعد بیش‌برازش هستند.

### □ روش یادگیری عمیق

یادگیری عمیق یک زیرشاخه از یادگیری ماشین است که از شبکه‌های عصبی مصنوعی برای استخراج ویژگی‌های سطح بالا از داده‌های خام استفاده می‌کند. یادگیری عمیق در برخی شرایط حتی با تخصص انسان قابل مقایسه است. الگوریتم‌های یادگیری عمیق به مقدار زیادی داده آموزشی برچسب‌گذاری شده و قدرت پردازش بالا برای دستیابی به سطح دقت بالا نیاز دارد.

دو نوع از محبوب‌ترین شبکه‌های عصبی عمیق، شبکه‌های عصبی کانولوشنی<sup>۳</sup> (CNN) و شبکه‌های عصبی بازگشتی<sup>۴</sup> (RNN) است. ورودی CNN از طریق تعداد متوالی لایه‌های کانولوشنی و ادغامی<sup>۵</sup> در مرحله یادگیری داده آموزش منتقل می‌شود. در مرحله بعد، خروجی CNN از طریق یک سری لایه کاملاً متصل عبور داده می‌شود و در نهایت، از تابع Softmax برای ردیابی استفاده می‌شود.

<sup>1</sup>Artificial Neural Network

<sup>2</sup>Back propagation

<sup>3</sup>Convolutional Neural Network

<sup>4</sup>Recursive Neural Network

<sup>5</sup>Pooling

## ۲-۳-۶- تکنیک‌های نیمه نظارتی یادگیری ماشین

یادگیری نیمه نظارتی به داده‌های برچسبدار و بدون برچسب در طول آموزش نیاز دارد. به دست آوردن همه نمونه‌های برچسبدار بسیار وقت‌گیر است؛ بنابراین داشتن ترکیبی از داده‌های برچسب‌گذاری شده و بدون برچسب انتخاب خوبی است. زیرا داده‌های برچسب‌گذاری شده نگاهی اجمالی به حضور کلاس‌های ممکن دارد. داده‌های بدون برچسب را می‌توان بعداً بر اساس آن کلاس‌های ممکن رده‌بندی کرد. الگوریتم‌های یادگیری نیمه نظارتی از مفروضاتی مانند فرض تداوم، فرض خوش‌های و فرض چندگانه بودن داده‌ها استفاده می‌کند. الگوریتم‌های نیمه نظارتی زمانی قابل استفاده هستند که داده‌های برچسبدار و بدون برچسب در دسترس باشند. همچنین، دادگان برچسب‌گذاری شده باید به گونه‌ای به دست آیند که نمونه‌های آن نگاه اجمالی به همه مکان‌های ممکن در سراسر منطقه آزمایش داشته باشد.

## ۲-۶-۴- تکنیک‌های بدون نظارت یادگیری ماشین

این تکنیک یادگیری با داده‌های بدون برچسب سروکار دارد و بدون هیچ راهنمایی بر روی داده‌ها کار می‌کند. از این‌رو نام آن بدون نظارت است. الگوریتم‌های بدون نظارت عموماً از خوش‌بندی برای گروه‌بندی نقاط داده استفاده می‌کنند به‌طوری که نقاط داخل خوش‌های مشابه باشند و خوش‌های متفاوت از یکدیگر باشند. تکنیک‌های مختلف خوش‌بندی در زیر مورد بحث قرار می‌گیرد:

### ▪ روش k-means

الگوریتم k-means یک الگوریتم خوش‌بندی بدون نظارت است که داده‌ها را به  $k$  خوش‌های تقسیم می‌کند. در ابتدا، مراکز خوش‌های به طور تصادفی تعیین می‌شوند. سپس، هر نقطه داده به نزدیک‌ترین مرکز خوش‌های اختصاص داده می‌شود. در نهایت، موقعیت مراکز خوش‌های مجدداً محاسبه می‌شود و این کار تا زمانی که مراکز خوش‌های تغییر نکنند تکرار می‌شود.

### ▪ روش فازی C-means (FCM)

در خوش‌بندی غیرفازی، هر نقطه داده به یک خوش‌های تعلق دارد. در خوش‌بندی فازی، هر نقطه داده می‌تواند به بیش از یک خوش‌های تعلق داشته باشد. خوش‌بندی فازی C-means یک الگوریتم خوش‌بندی فازی محبوب است که عملکرد بهتری را در مجموعه داده‌های همپوشان نسبت به الگوریتم‌های خوش‌بندی سخت دارد. مانند

k-means، مشخصات قبلی تعداد خوشها یک اشکال عمدۀ خوشبندی C-means است.

## □ روش DBSCAN

خوشبندی فضایی چگالی محور برای کاربردهای نویزی (DBSCAN) یک الگوریتم خوشبندی مبتنی بر چگالی است که می‌تواند خوشها بی با اشکال و اندازه‌های متفاوت را از تعداد زیادی داده در حضور داده نویزی و پرت کشف کند. این الگوریتم به دو پارامتر  $\text{eps}$  و  $\text{minPts}$  نیاز دارد.  $\text{eps}$  نشان‌دهنده همسایگی اطراف یک نقطه داده است و  $\text{minPts}$  به حداقل تعداد نقاط داده موردنیاز در شعاع  $\text{eps}$  برای تشکیل یک منطقه متراکم اشاره دارد.

## ۲-۶-۵- تکنیک‌های یادگیری ماشین تحت شرایط مختلف آموزش و آزمون

شرایط مکان سرپوشیده می‌تواند در طول زمان تغییر کند. از این‌رو، اطمینان از این که داده‌های آزمایش می‌تواند در همان شرایطی که داده‌های آموزش جمع‌آوری شده است، به دست آید دشوار است. در یک مکان ثابت، RSS به دلیل تغییر شرایط محیط سرپوشیده بهشدت تحت‌تأثیر قرار می‌گیرد؛ به عنوان مثال، داده‌های RSS جمع‌آوری شده با باز نگهداشتن درهای مکان می‌تواند بسیار نویزی تر از بسته نگهداشتن درها باشد. این مسئله می‌تواند در دقت موقعیت یابی هر رده‌بند اثر بگذارد. در این سناریو، تکنیک‌های یادگیری ماشین گروهی مانند Boosting و Bagging بسیار مناسب هستند. زیرا نسبت به الگوریتم‌های یادگیری ماشین تکی، قابلیت تعمیم را بهتر حفظ می‌کند. علاوه بر این، نقاط دسترسی جدید Wi-Fi می‌توانند در طول زمان، نصب، جایگزین و یا حذف شوند. این عوامل باعث تغییر ابعاد دادگان جمع‌آوری شده و درنتیجه تغییر فضای ویژگی‌ها می‌شود؛ بنابراین، این تکنیک‌ها می‌توانند به طور پویا مدل رده‌بند را به روز کنند. رویکردهای یادگیری ماشین گروهی ذکر شده به تفصیل در فصل ۳ آمده است.

## ۲-۶-۶- چالش‌های موقعیت یابی با الگوریتم‌های یادگیری ماشین

على‌رغم تلاش محققان همچنان چالش‌هایی در موقعیت یابی مبتنی بر RSS، با استفاده از الگوریتم‌های یادگیری ماشین وجود دارد. این چالش‌ها در ادامه ذکر شده است.

## □ یکپارچگی در برچسب‌زن

از آن جا که الگوریتم‌های یادگیری ماشین عمدهاً دنباله مسیر را پیش‌بینی می‌کنند، نمی‌توان برای مقیاس‌های فضایی متفاوت، یک مدل را استفاده کرد. از این‌رو، رویکردهای موقعیت‌یابی باید به صورت جداگانه برای هر مقیاس فضایی انجام بگیرد.

#### □ انتخاب و استخراج ویژگی‌های معنادار

داده‌های موقعیت‌یابی معمولاً دارای ویژگی‌های زیادی هستند که اغلب همپوشانی دارند و ویژگی‌های معنادار برای موقعیت‌یابی در محیط‌های مختلف متفاوت است. علاوه بر این، جمع‌آوری داده‌های آموزشی برای موقعیت‌یابی داخلی دشوار و پرهزینه است.

#### □ هزینه، پیچیدگی و سهولت استفاده

سیستم‌های موقعیت‌یابی موجود دارای چالش‌هایی از جمله گران‌قیمت بودن، پیچیدگی هستند. سیستم‌های موقعیت‌یابی ساده‌تر از نظر محاسباتی سریع‌تر هستند و می‌توانند به درخواست چندین کاربر به طور همزمان پاسخ دهند.

#### □ امنیت و حریم خصوصی

یک سیستم موقعیت‌یابی امن نیاز است تا توسط خرابکاران مورد حمله قرار نگیرد. اگر پایگاه‌داده اثر انگشت دست‌کاری شود، موقعیت‌یابی کاربر می‌تواند به یک کار چالش‌برانگیز تبدیل شود. حریم خصوصی، محرمانه بودن داده‌های موقعیت‌یابی را تضمین می‌کند.

#### □ دستگاه‌ها و سنسورهای نوظهور

در طراحی سیستم موقعیت‌یابی، باید نسبت به ترکیب سنسورها و دستگاه‌های مختلف انعطاف‌پذیر باشد و اتصال سنسورها و دستگاه‌های جدید به سیستم موقعیت‌یابی یک چالش تحقیقاتی مهم است.

#### □ موقعیت‌یابی در شرایط اضطراری

موقعیت‌یابی در شرایط اضطراری بسیار مهم است. داده‌های اضطراری را می‌توان به همراه توزیع داده‌های اصلی گذشته برای تنظیم عملکرد در سیستم‌های موقعیت‌یابی ترکیب کرد.

## ۷-۲- جمع‌بندی

در این فصل، پس از تعریف مفاهیم مقدماتی موقعیت‌یابی مکان داخلی، دادگان اندازه‌گیری مورد بررسی قرار گرفت. این دادگان بر حسب استفاده از فناوری و الگوریتم مشخص، می‌تواند در مکان‌های مختلف به کار برد شود. علاوه بر آن الگوریتم‌های موقعیت‌یابی معرفی شد که این تکنیک‌های تخمین موقعیت را می‌توان برای تعیین مختصات مکان به کار برد.

در موقعیت‌یابی مکان سرپوشیده، فناوری‌های بی‌سیم نقش اصلی را بازی می‌کنند. استفاده از مشخصه‌های سیگنال رادیویی برای موقعیت‌یابی می‌تواند در هر فناوری متفاوت باشد. قدرت سیگنال دریافتی (RSS) از مهم‌ترین ویژگی‌های سیگنال رادیویی است. اکثر فناوری‌های ارتباط بی‌سیم RSS را محاسبه می‌کنند و مقادیر آن قابل دسترسی است. در این فصل به برخی از فناوری‌هایی که در موقعیت‌یابی مبتنی بر RSS است نیز پرداخته شد. موقعیت‌یابی مبتنی بر RSS با چالش‌هایی مواجه است که به طور قابل توجهی در دقت موقعیت‌یابی اثر گذارند. چندمسیره شدن، شرایط خط غیر دید، تداخل، تغییر قدرت سیگنال و محیط دینامیکی انتشار سیگنال از جمله این چالش‌ها هستند. همچنین موقعیت‌یابی مکان سرپوشیده نیازمند برنامه‌ای برای کالibrاسیون دوره و نگهداری ادوات است که باعث صرف هزینه و زمان می‌شود.

در ادامه ساختار موقعیت‌یابی با یادگیری ماشین بیان شد. اولین مرحله برای یک طرح موقعیت‌یابی مبتنی بر یادگیری ماشین جمع‌آوری دادگان مناسب برای آموزش مدل است. پس از آن پیش‌پردازش داده‌ها برای غنی‌کردن مجموعه دادگان به کار برد می‌شود که می‌تواند شامل درون‌یابی داده‌های گم شده و یا فیلتر کردن برای حذف نویزهای ناخواسته باشد. انتخاب و استخراج ویژگی‌ها، در مرحله بعد، برای شناسایی الگو و یا کاهش پیچیدگی مدل و زمان آموزش مورداً استفاده قرار می‌گیرد.

پس از بیان چارچوب کلی موقعیت‌یابی با الگوریتم‌های یادگیری ماشین این تکنیک‌ها در موقعیت‌یابی مبتنی بر RSS بیان شدند و الگوریتم‌های یادگیری تحت نظارت، بدون نظارت و نیمه نظارتی معرفی شدند. همچنین چالش‌هایی در موقعیت‌یابی با الگوریتم‌های یادگیری ماشین وجود دارد که به آن‌ها پرداخته شد. وجود چنین چالش‌هایی، مباحث جدید را برای ادامه پژوهش در زمینه موقعیت‌یابی پدید می‌آورد.

## **فصل ۳:**

**راهکارهای پیشنهادی بهبود دقیق در تعیین**

**موقعیت**

## ۱-۳- مقدمه

در فصل قبل انواع روش‌های موقعیت یابی با استفاده از الگوریتم‌های یادگیری ماشین بیان شد. روش‌های پایه با وجود سادگی در پیاده‌سازی، قادر به تخمین موقعیت با دقت بالا نیستند. به همین منظور، روش‌های یادگیری ماشین برای بهبود دقت موقعیت یابی مکان‌های سرپوشیده به کار گرفته شده‌اند. همچنین بر اساس فناوری‌های ارتباطی بی‌سیم، در عمل، عمدتاً از شبکه‌های بی‌سیم Wi-Fi و BLE استفاده می‌شود. در میان منابع پژوهشی، فناوری Wi-Fi پرکاربردترین فناوری مورداستفاده در موقعیت یابی مکان‌های سرپوشیده است. دو رویکرد برای موقعیت یابی با استفاده از Wi-Fi وجود دارد. رویکرد اول، استفاده از مدل انتشاری سیگنال در تخمین RSSI برای محاسبه فاصله و رویکرد دوم، ساخت یک نقشه اثر انگشت Wi-Fi و استفاده از سیگنال Wi-Fi برای تخمین موقعیت است. روش اثر انگشت به طور قابل توجهی در ارتقای موقعیت یابی مکان سرپوشیده در حال توسعه است.

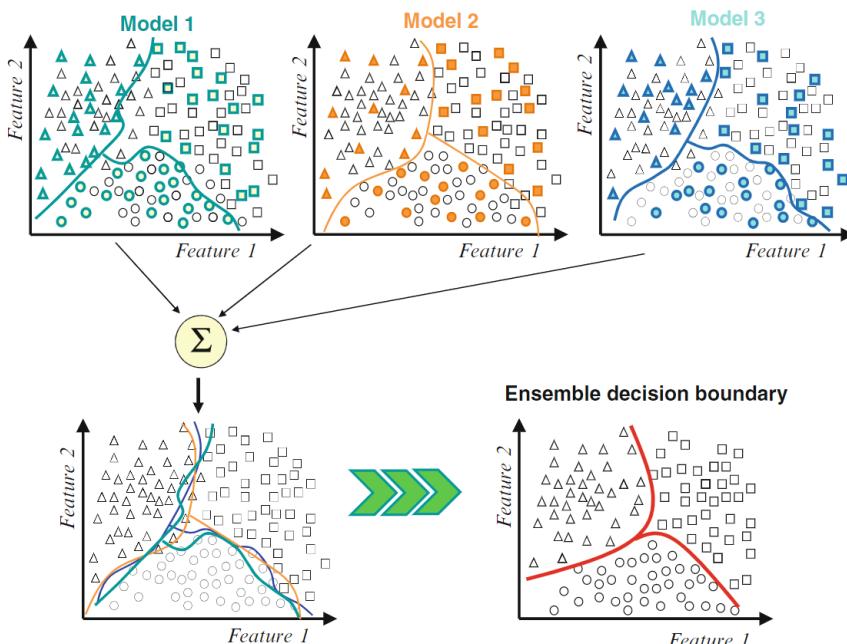
بنا به تعریف، یک سیستم موقعیت یاب باید موقعیت هدف را با دقت مناسب و به صورت بلاذرنگ تخمین بزند؛ بنابراین انتخاب یک روش یادگیری ماشین مناسب در تخمین موقعیت که بتواند الزامات سیستم موقعیت یاب را برآورده کند ضروری است. در این فصل راهکارهای پیشنهادی الگوریتم‌های یادگیری ماشین در بهبود دقت موقعیت یابی با درنظرگیری امکان پیاده‌سازی و سرعت اجرا ارائه می‌شود. این راهکارها برای یک سیستم موقعیت یاب مبتنی بر نقشه اثر انگشت RSSI در تشکیل نقشه اثر انگشت ارائه شده‌اند.

## ۲-۳- روش‌های یادگیری گروهی

بسیاری از پژوهش‌های انجام شده در موقعیت یابی سرپوشیده با استفاده از نقشه اثر انگشت Wi-Fi، بر اساس روش kNN شکل گرفته‌اند. از آنجاکه برخی از نقاط دسترسی در محیط می‌تواند از یک نقطه مشخص دور باشد، آن نقاط دسترسی در یک موقعیت مشخص ممکن است مشاهده نشود و بردار RSSI در برخی از مکان‌ها شامل سیگنال‌های دریافت شده توسط همه نقاط دسترسی نباشد. همچنین ممکن است برخی نقاط مرجع در بردار شامل RSSI مشابه باشد. روش kNN همه نقاط مرجع روی نقشه اثر انگشت را بدون لحاظ کردن این نکته در نظر می‌گیرد. از سوی دیگر، همسایه‌هایی که از طریق الگوریتم kNN یافت می‌شود، ممکن است فراتر از اندازه‌گیری‌های ممکن در محیط باشد. زیرا تضعیف سیگنال هر نقطه دسترسی نه تنها به فاصله آن مرتبط

است، بلکه تحت تأثیر بسیاری از عوامل محیطی نیز قرار می‌گیرد. این موضوع، باعث پدیدارشدن اثر حداقل فاصله سیگنالی بردار RSSI و نقاط مرجع می‌شود. با توجه به محدودیت‌های بیان شده در روش kNN، استفاده از الگوریتم‌های یادگیری گروهی گزینه‌ای ایدئال برای جایگزینی با این روش‌ها است.

فرضیه استفاده از سیستم‌های تصمیم‌گیری بر اساس رأی‌گیری در زندگی روزمره ما اساساً با کاربرد آن‌ها در هوش محاسباتی متفاوت نیست. ما اغلب پیش از تصمیم‌گیری با دیگران مشورت می‌کنیم؛ زیرا گذشته و دقت تصمیم‌گیری هر یک از تصمیم‌گیرندگان فردی متفاوت است. هر خطای طبقه‌بندی از دو مؤلفه تشکیل شده است که می‌بایست آن‌ها را کنترل کرد: بایاس<sup>۱</sup>، دقت طبقه‌بندی‌کننده<sup>۲</sup> و واریانس<sup>۲</sup>، حساسیت طبقه‌بندی‌کننده‌هایی با بایاس پایین تمایل به واریانس بالا دارند و بر عکس، بنابراین، هدف سیستم‌های طبقه‌بندی‌کننده‌هایی میانگین‌گیری، برای کاهش واریانس است.



شکل (۱-۳) کاهش واریانس با استفاده از الگوریتم‌های یادگیری گروهی [۲۱]

کاهش واریانس به معنای کاهش نوسانات در یک مقدار است. این کار می‌تواند با میانگین‌گیری مقادیر مختلف انجام شود. در زمینه طبقه‌بندی، کاهش واریانس می‌تواند به بهبود دقت طبقه‌بندی کمک کند. این به این دلیل است که طبقه‌بندی‌کننده‌ها اغلب در طبقه‌بندی نمونه‌های جدید اشتباه می‌کنند. با میانگین‌گیری

<sup>1</sup> Bias

<sup>2</sup> Variance

خروجی‌های چند طبقه‌بندی کننده مختلف، می‌توان خطای آن‌ها را کاهش داد. روش‌های مختلفی برای ترکیب خروجی‌های طبقه‌بندی کننده‌ها وجود دارد. میانگین‌گیری تنها یکی از این روش‌ها است. شکل (۱-۳) نشان می‌دهد که چگونه میانگین‌گیری خروجی‌های دو طبقه بندی کننده با واریانس بالا می‌تواند واریانس خروجی را کاهش دهد.

الگوریتم‌های یادگیری گروهی در موقعیت‌یابی می‌تواند چالش‌های موجود در فرایند موقعیت‌یابی، از جمله چندمسیره شدن سیگنال و شرایط NLOS، را نیز بهبود بخشد. این الگوریتم‌ها با بهره‌گیری از تحلیل گروهی چندین مدل از داده‌ها، می‌تواند الگوهای پیچیده چندمسیره شدن را تشخیص داده و با ترکیب اطلاعات دقت موقعیت‌یابی را افزایش دهد. علاوه بر این، الگوریتم‌های یادگیری گروهی به کاهش اثرات تضعیف RSSI نیز می‌پردازد. در شرایطی که سیگنال‌ها به دلیل موانع موجود، تداخلات سیگنال در مسیر انتقال دچار تضعیف شوند، این الگوریتم‌ها با تجمعی تجربه‌های مختلف مدل‌های یادگیری ماشین، می‌توانند اطلاعات صحیح‌تری از قدرت سیگنال به دست آورده و اثر تضعیف را به حداقل برسانند.

## ۲-۲-۳- توسعه الگوریتم‌های یادگیری گروهی

بسیاری از بررسی‌ها به کار Sheela و Dasarathy در سال ۱۹۷۹ میلادی به عنوان یکی از اولین نمونه‌های الگوریتم‌های یادگیری گروهی، با ایده‌های آن‌ها در مورد تقسیم فضای ویژگی‌ها و استفاده از چندین طبقه‌بندی کننده اشاره می‌کنند [۲۲]. با این حال، نخستین بار در [۱۱] روشی به نام Boosting معرفی شد و نشان داد که در یک مسئله طبقه‌بندی دو کلاسه، یک طبقه‌بندی کننده قوی با خطای کم می‌تواند از مجموعه‌ای از طبقه‌بندی کننده‌هایی که خطای هر یک از آن‌ها از خطای یک طبقه بندی کننده حدس تصادفی بیشتر باشد، ساخته شود. نظریه Boosting پایه و اساس الگوریتم یادگیری گروهی بعدی، AdaBoost را فراهم کرد که از محبوب‌ترین الگوریتم‌های یادگیری گروهی است و نظریه را به مسائل چند کلاسه و تقریب تابع گسترش می‌دهد [۲۳].

به دلیل موفقیت در این کارهای پایه، از آن به بعد تحقیقات در الگوریتم‌های یادگیری گروهی گسترش یافته است و این الگوریتم‌ها تحت نام‌های مختلف ظهور پیدا کردند. Bagging [۲۴]، جنگل تصادفی<sup>۱</sup> (گروهی از درختان تصمیم)، سیستم‌های طبقه‌بندی کننده مرکب، ترکیب ماهرها (MoE) [۲۵]، روش تعمیم پشته‌سازی [۲۶]، XGBoost [۲۷] و بسیاری دیگر از الگوریتم‌ها معرفی شده‌اند.

<sup>1</sup> Random Forest

### ۳-۲-۳-اجزای الگوریتم یادگیری گروهی

لازم است تا سه رویه برای تشکیل یک سیستم یادگیری گروهی در پیش گرفته شود. این سه رویه به ترتیب عبارت است از نمونه‌برداری و انتخاب داده، آموزش یادگیرندگان ضعیف و درنهایت ترکیب یادگیرندگان که در ادامه به آن پرداخته می‌شود.

#### □ نمونه‌برداری و انتخاب داده

مواجهه با خطاها گوناگون به‌ازای هر نمونه در سیستم‌های یادگیری گروهی از اهمیت بالایی برخوردار است. چراکه، اگر همه مدل‌ها به‌ازای هر نمونه خروجی یکسانی داشته باشند، هیچ دانشی از ترکیب آن‌ها به دست نمی‌آید. از این‌رو، در تشکیل یک سیستم یادگیری گروهی نیاز به تنوع در تصمیم‌گیری اعضای گروه، بهویژه زمانی که اعضا در تصمیم‌گیری با خطا مواجه می‌شوند، است. این موضوع اهمیت تنوع داده را در این سیستم‌ها نشان می‌دهد.

با استفاده از رویکردهای متفاوت، می‌توان به تنوع داده در یادگیرندگان رسید. رایج‌ترین رویکرد استفاده از مجموعه‌دادگان آموزشی متفاوت است که در شکل (۱-۳) نیز نشان داده شده است. رویکردهای مختلف در انتخاب داده منجر به پدیدآمدن الگوریتم‌های گروهی متفاوت می‌شود. به عنوان مثال انتخاب با جای‌گذاری دادگان آموزشی، روش Bagging را نتیجه می‌دهد، در حالی که نمونه‌برداری با توزیعی که به نفع نمونه‌های اشتباه طبقه‌بندی شده‌اند، هسته اصلی الگوریتم‌های Boosting است. از سوی دیگر، می‌توان از زیرمجموعه‌های مختلفی از ویژگی‌های موجود برای آموزش هر یادگیرنده استفاده کرد که منجر به روش‌های زیر فضای تصادفی<sup>۱</sup> می‌شود. علی‌رغم اهمیت تنوع داده در عملکرد یادگیرندگان که به خوبی اثبات شده است، اما رابطه‌ای صریح بین تنوع و دقت الگوریتم گروهی شناسایی نشده است.

#### □ آموزش یادگیرندگان

استراتژی مورداستفاده برای آموزش اعضای گروه، هسته هر سیستم یادگیری گروهی است. الگوریتم‌های رقابتی متعددی برای آموزش طبقه‌بندی‌کننده‌های گروه توسعه داده شده‌اند. با این حال، Bagging (و الگوریتم‌های مرتبط مانند جنگل تصادفی)، Boosting (و انواع آن)، تعمیم پشت‌سازی به عنوان رایج‌ترین رویکردهای به کار گرفته شده حساب می‌شود. این رویکردها در ۴-۲-۳ و ۵-۲-۳-بیان می‌شود.

<sup>1</sup> Random subspace methods

## □ ترکیب یادگیرندگان

مرحله نهایی در سیستم‌های مبتنی بر مجموعه شامل ترکیب پیش‌بینی‌های یادگیرندگان است. استراتژی ترکیب به نوع یادگیرندگان مورداستفاده بستگی دارد. برای یادگیرندگان گسته مانند SVM، رأی اکثریت رایج است. برای طبقه‌بندی کننده‌های پیوسته؛ مانند شبکه عصبی، ترکیب کننده‌های حسابی یا الگوهای تصمیم‌گیری پیچیده گزینه‌های مناسبی هستند. بسیاری از ترکیب کننده‌ها را می‌توان بدون آموزش اضافی استفاده کرد، در حالی که ترکیب‌های پیچیده‌تر ممکن است نیاز به یک مرحله اضافی داشته باشد.

ابتدا فرض می‌شود که فقط برچسب‌ها از خروجی یادگیرندگان در دسترس است. تصمیم یادگیرنده  $t$  را به صورت  $\{0,1\}$  که  $d_{t,c} \in \{0,1\}$  و  $c = 1, \dots, C$  و  $t = 1, \dots, T$  و همچنین  $T$  تعداد یادگیرندگان و  $C$  تعداد برچسب‌ها است. اگر یادگیرنده (یا فرضیه)  $t$  که با  $h_t$  نمایش داده می‌شود، طبقه  $c$  را انتخاب کند، آنگاه  $d_{t,c} = 1$  و در غیر این صورت برابر با صفر خواهد بود. طبق تعریف فوق قوانین ترکیبی که برای یادگیرندگان ارائه شده است در ادامه توضیح داده می‌شود.

(۱) **رأی اکثریت<sup>۱</sup>**: رأی اکثریت دارای سه نوع است بسته به این که آیا تصمیم گروه (۱) آن چیزی است که همه یادگیرندگان با آن موافق هستند (اتفاق نظر آرا) (۲) پیش‌بینی‌شده توسط حداقل بیش از نیمی از تعداد یادگیرندگان (اکثریت ساده) یا (۳) بیشترین مجموعه تعداد آرا فارغ از آن که بیش از ۵۰ درصد موافق باشند (تجمیع رأی). درصورتی که نوع رأی اکثریت مشخص نشده باشد، این ترکیب به تجمیع رأی اشاره دارد و به زبان ریاضی به صورت رابطه (۱-۳) بیان می‌شود.

طبقه  $*_c$  انتخاب می‌شود اگر

$$\sum_{t=1}^T d_{t,c^*} = \max_c \sum_{t=1}^T d_{t,c} \quad (1-3)$$

(۲) **رأی اکثریت وزن دار**: اگر دلیلی بر این باور داشته باشیم که برخی از یادگیرندگان بیشتر از بقیه صحیح هستند، وزن دادن به تصمیمات آن یادگیرندگان می‌تواند عملکرد کلی را در مقایسه با اکثریت آرا بهبود بخشد. می‌توانیم وزن  $w_t$  را به یادگیرنده  $h_t$  بر اساس عملکرد کلی تخمین آن اختصاص دهیم. گروه با توجه به رأی اکثریت آرا کلاس  $c^*$  را انتخاب می‌کند اگر

$$\sum_{t=1}^T w_t d_{t,c^*} = \max_c \sum_{t=1}^T w_t d_{t,c} \quad (2-3)$$

يعنى کل آرای وزن دار دریافت شده توسط طبقه  $*_c$  بیشتر از تمام آرای وزن دار دریافت شده توسط

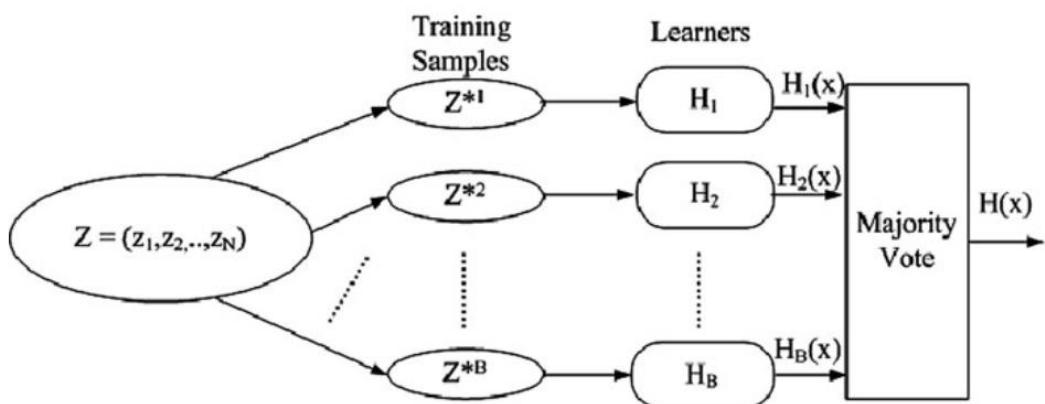
<sup>۱</sup> Majority voting

سایر طبقه‌ها باشد. به‌طور کلی، وزن‌های رأی به گونه‌ای نرمال می‌شوند که مجموع آن‌ها برابر با یک باشد.

مجموعه‌ای غنی از الگوریتم‌های گروهی در چند سال گذشته توسعه یافته است. با این وجود، برخی از این الگوریتم‌های منتخب و به خوبی تثبیت شده هستند که قابلیت‌های آن‌ها نیز به طور گسترده آزمایش و گزارش شده است. در ادامه برخی از برجسته‌ترین الگوریتم‌های یادگیری گروهی ارائه شده است و در فصل بعدی نتایج حاصل از شبیه‌سازی آن‌ها مقایسه شده است.

### Bagging - ۴-۲-۳

کوتاه شده عبارت Bagging (Bootstrap Aggregation) و به معنای «تجمیع خود راهانداز» است. این روش توسط Breiman در سال ۱۹۹۶ معرفی شده است و یکی از اولین، ساده‌ترین و در عین حال مؤثرترین الگوریتم‌های یادگیری گروهی است [۲۴]. با توجه مجموعه‌دادگان داده شده  $S$ ، روش Bagging به طور ساده  $T$  یادگیرنده مستقل که هر کدام با انتخاب و جایگذاری داده و همچنین استفاده از یادگیرنده‌گان ضعیف که در این روش تنوع در داده با تغییرات انتخاب و جایگذاری داده و همچنین استفاده از یادگیرنده‌گان ضعیف که مرز تصمیم گیری آن‌ها با تغییرات نسبتاً کوچک در دادگان تغییر می‌کند، تضمین می‌شود. طبقه‌بندی کنندگان خطی مانند درخت تصمیم، SVM خطی و شبکه‌های عصبی تک‌لایه کاندیدهای خوبی به عنوان یادگیرنده‌گان ضعیف هستند. یادگیرنده‌گان در نهایت با روش اکثریت ساده ترکیب می‌شوند. شکل (۳-۲) نمای کلی از روش Bagging را برای مسائل طبقه‌بندی نمایش می‌دهد.



شکل (۳-۲) رویکرد Bagging در مسائل طبقه‌بندی [۲۱]

## □ جنگل تصادفی

یکی از نسخه‌های خلاقانه روش Bagging، الگوریتم جنگل تصادفی است که اساس آن ترکیب مجموعه از درختان تصمیم آموزش دیده با مکانیزم Bagging است. جنگل تصادفی توسط Breiman معرفی شد [۲۴]. جنگل تصادفی را می‌توان هم برای متغیرهای گستته که در موقعیت یابی می‌تواند شرایط محیطی منظور شود، و هم برای متغیرهای پیوسته که در موقعیت یابی می‌تواند RSS سیگنال ارتباطی بی‌سیم باشد، استفاده کرد. به طور مشابه، متغیرهای پیش‌بینی شده نیز می‌تواند گستته یا پیوسته باشد.

عبارت «تصادفی» علاوه بر انتخاب تصادفی دادگان، به انتخاب تصادفی متغیرهای ویژگی در هر گره درخت تصمیم اشاره دارد. این به این معناست که هنگام انتخاب ویژگی در هر گره درخت،  $m$  ویژگی از  $p$  ویژگی در بردار انتخاب می‌شود. این انتخاب تصادفی بردار ویژگی‌ها باعث ساخت درختان متمایز از هم می‌شود و با درنظرگیری روش انتخاب و جای‌گذاری دادگان، بیشترین تنوع به وجود می‌آید. شبکه کد الگوریتم جنگل تصادفی در الگوریتم ۱ بیان شده است.

در [۲۴] برای مسائل طبقه‌بندی، رشد درختان را تا زمانی که به یک گره خالص (برگ) برسد، پیشنهاد کرده است. اما در [۲۸] که پژوهشی جدیدتر است، کنترل عمق درختان پیشنهاد شده است.

### الگوریتم ۱: جنگل تصادفی

فرض کنید  $\{(x_1, y_1), \dots, (x_N, y_N)\} = D$  به مجموعه دادگان آموزش اشاره دارد که در آن  $x_i = (x_{i,1}, \dots, x_{i,p})^T$  که  $p$  تعداد ویژگی‌ها است. به ازای  $1 \leq j \leq J$  (تعداد یادگیرندگان ضعیف) انجام دهید:

(۱) یک انتخاب با جای‌گذاری  $D_i$  با اندازه  $N$  از  $D$  انجام دهید.

(۲) با استفاده از نمونه‌برداری با جای‌گذاری انتخاب شده به عنوان دادگان آموزشی، درخت تصمیم را تشکیل دهید:  
ا) با تمام ویژگی‌ها در یک گره شروع کن.

ب) گام‌های زیر را برای گره‌های تقسیم نشده تکرار کن تا زمانی که شرط توقف درخت (گره با یک ویژگی/عمق درخت) ارضاء شود:

(i)  $m$  ویژگی را به صورت تصادفی از  $p$  ویژگی در دسترس انتخاب کنید.

(ii) بهترین ویژگی تقسیم در میان  $m$  ویژگی انتخاب شده در گام (i) را پیدا کنید.

(iii) گره را با استفاده از ویژگی انتخابی در گام (ii) تقسیم کنید.

برای پیش‌بینی در نقطه جدید  $x$ :

$$\hat{f}(x) = \arg \max_y \sum_{j=1}^J I(\hat{h}_j(x) = y) \quad (3-3)$$

که در آن  $(x_j, \hat{h}_j(x))$  تخمین درخت  $j$  برای ورودی  $x$  وتابع  $I$  به صورت زیر تعریف می‌شود:

$$I(Y = f(X)) = \begin{cases} 0 & \text{if } Y \neq f(X) \\ 1 & \text{otherwise.} \end{cases} \quad (4-3)$$

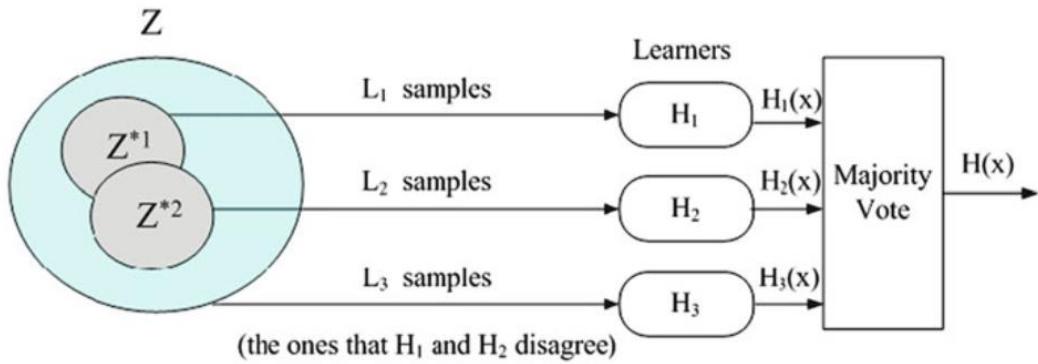
جنگل تصادفی به این شهرت دارد که با تنظیم اولیه کاملاً خوب کار می‌کند. اما برای کاربرد موقعیت‌یابی که نیازمند سرعت بالا در محاسبات و سادگی پیاده‌سازی است. نیاز به تنظیم سه پارامتر وجود دارد:

- $m$ , تعداد ویژگی‌هایی که به صورت تصادفی در هر گره انتخاب می‌شود.
- $J$ , تعداد درختان موجود در جنگل
- اندازه درخت که یا با کمترین اندازه برگ مشخص می‌شود یا بیشینه تعداد گره‌های نهایی درختان، در مسائل طبقه‌بندی مقدار  $m = \sqrt{M}$  پیشنهاد شده است [۲۹]. برای تنظیم پارامترهای  $J$  و اندازه درخت، در این پژوهش روش الگوریتم ژنتیک را پیشنهاد شده است که در بخش ۴-۳-۴ معرفی می‌شود.

## Boosting - ۵-۲-۳

Boosting کار ابتدایی Schapire در [۱۱] معرفی شد. این روش، یک روش تکرارشونده برای تولید یک طبقه‌بندی کننده قوی است و قابلیت رسیدن به خطای آموزش پایین، با گروهی از یادگیرندگان ضعیفی که کمی بهتر از حدس تصادفی عمل می‌کنند را دارد. این روش با اینکه مجموعه یادگیرندگان ضعیف را با استفاده از رأی اکثریت ترکیب می‌کند، اما از یک منظر مهم با روش Bagging تفاوت دارد. در روش Bagging انتخاب نمونه‌ها برای آموزش هر یک از یادگیرندگان ضعیف از طریق انتخاب بدون جای‌گذاری صورت می‌گیرد، به این معنی که هر نمونه شناس برابر برای قرار گرفتن در هر مجموعه‌داده آموزشی دارد. اما در روش Boosting مجموعه‌دادگان آموزش برای هر یادگیرنده بعدی بر نمونه‌هایی تمرکز می‌کند که توسط یادگیرنده‌های قبلی تولید شده، اشتباہ طبقه‌بندی شده‌اند و احتمال انتخاب این نمونه‌ها افزایش می‌یابد.

ساده‌ترین روش Boosting برای مسائل طبقه‌بندی دوتایی طراحی شده است که مجموعه‌ای از سه یادگیرنده ضعیف را در یک‌زمان ایجاد می‌کند. اولین یادگیرنده (یا فرضیه)  $h_1$  بر روی زیرمجموعه‌ای تصادفی از دادگان آموزشی، آموزش داده می‌شود (مشابه روش Bagging). یادگیرنده دوم،  $h_2$ ، بر روی زیرمجموعه‌ای متفاوت از مجموعه دادگان اصلی، آموزش داده می‌شود که دقیقاً نیمی از آن به درستی توسط  $h_1$  تخمین زده شده و نیمی دیگر به اشتباہ طبقه‌بندی شده‌اند. به این زیرمجموعه آموزشی که با توجه به تصمیم  $h_1$  ساخته شده است «با اطلاعات‌ترین» مجموعه داده گفته می‌شود. یادگیرنده سوم،  $h_3$ ، با نمونه‌هایی آموزش داده می‌شود که  $h_1$  و  $h_2$  در مورد آن‌ها اختلاف نظر دارند. در نهایت این سه یادگیرنده از طریق رأی اکثریت سه‌تایی ترکیب می‌شوند. شکل (۳-۳) نمایش گرافیکی از ایده اولیه الگوریتم Boosting را ترسیم کرده است.



شکل (۳-۳) نمایش گرافیکی ایده اولین روش Boosting ارائه شده در [۱۱]

همچنین در [۲۵] اثبات شده است که خطای آموزش این گروه با سه یادگیرنده به زیر  $2\varepsilon^2 - 3\varepsilon^2 < \varepsilon$  محدود می‌شود که  $\varepsilon$  خطای هر یک از یادگیرنده‌گان است، مشروط بر این که هر یادگیرنده دارای خطای  $< 0.5$ ، کمترین مقداری که می‌توان از یک یادگیرنده دو تایی انتظار داشت، باشد.

### □ الگوریتم AdaBoost

AdaBoost مخفف کلمه Adaptive Boosting و به معنای «تقویت تطبیقی»، یکی از پراستفاده‌ترین الگوریتم‌های یادگیری گروهی است. این الگوریتم پس از معرفی با چندین تغییر به الگوریتم AdaBoost.M1 و AdaBoost.M2 گسترش یافته شده است. در این بخش به الگوریتم AdaBoost.M1 که محبوب‌ترین الگوریتم در میان الگوریتم‌های AdaBoost است پرداخته می‌شود.

دو تفاوت اساسی با روش‌های Boosting دارد: (۱) نمونه‌هایی از توزیع نمونه به روز رسانی شده در هر مرحله به مجموعه دادگان بعدی نیز کشیده می‌شوند و (۲) یادگیرنده‌گان از طریق اکثریت آرای وزن دار ترکیب می‌شوند که در آن وزن‌های رأی‌گیری بر اساس اشتباها آموزش یادگیرنده‌گانی است که هر یک بر اساس توزیع نمونه برداری وزن دار شده اند.

شبیه کد AdaBoost.M1 در الگوریتم ۲ آورده شده است. توزیع نمونه‌برداری ( $D_t$ )، وزنی را برای هر نمونه آزمایش  $x_i$  که  $i = 1, \dots, N$  اختصاص می‌دهد که از آن زیرمجموعه دادگان آموزشی  $S_t$  برای هر یادگیرنده متوالی  $h_t$  استخراج می‌شود. مقداردهی اولیه یک توزیع یکنواخت<sup>۱</sup> است و از این رو، تمام نمونه‌ها احتمال برابری برای استخراج شدن در مجموعه دادگان آموزشی ابتدایی دارند. در گام بعد خطای آموزش یادگیرنده  $h_t$ ، یعنی  $\varepsilon_t$ ، با جمع کردن وزن‌های نمونه‌های اشتباها طبقه‌بندی شده مطابق با رابطه (۵-۳) محاسبه می‌شود. در این رابطه  $\| \cdot \|$  برابر با ۱ است اگر عبارت شرطی داخل آن صادق باشد و در غیر این صورت برابر با صفر

<sup>1</sup> uniform

خواهد شد. این روش نیازمند آن است که خطای مطرح شده در آن کمتر از  $\frac{1}{\beta_t}$  باشد، لذا مطابق با رابطه (۳-۶)،  $\beta_t$  نیاز است تا نرمال‌سازی شود.

## الگوریتم ۲: AdaBoost.M1

وروودی: دادگان آموزش  $\{y_i\}$  و  $i = 1, \dots, N$  که  $D = \{x_i, y_i\}$  تعداد یادگیرندگان.

$$D_1(i) = 1/N$$

به ازای  $t = 1, 2, \dots, T$  انجام دهید:

(۱) زیرمجموعه  $S_t$  را با استفاده از توزیع  $D_t$  استخراج کنید.

(۲) یادگیرنده پایه را بر روی  $S_t$  آموزش دهید و فرضیه را دریافت کنید  $Y \rightarrow h_t: X$

(۳) خطای فرضیه  $h_t$  را محاسبه کنید:

$$\varepsilon_t = \sum_i I[h_t(x_i) \neq y_i] D_t(x_i) \quad (۵-۳)$$

اگر  $\varepsilon_t > 1/2$  ادامه را رها کنید.

(۴) مقدار  $\beta_t$  را تنظیم کنید:

$$\beta_t = \varepsilon_t / (1 - \varepsilon_t) \quad (۶-۳)$$

(۵) توزیع نمونه‌برداری را به روز کنید:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \cdot \begin{cases} \beta_t, & \text{if } h_t(x_i) = y_i \\ 1, & \text{otherwise} \end{cases} \quad (۷-۳)$$

که در آن (۷-۳) یک ثابت نرمال‌سازی برای اطمینان از صحت درست بودنتابع توزیع است.

پایان حلقه

اکثریت آرای وزن دار: برای نمونه  $Z$ ، تمام آرا را برای هر طبقه به دست آورید:

$$V_c = \sum_{t:h_t(z)=\omega_c} \log\left(\frac{1}{\beta_t}\right), c = 1, \dots, C \quad (۸-۳)$$

خروجی: طبقه با بیشترین مقدار  $V_c$

قلب الگوریتم AdaBoost.M1، قاعده به روزرسانی توزیع در رابطه (۷-۳) است. وزن‌های توزیع نمونه‌هایی که به درستی توسط یادگیرنده  $h_t$  طبقه‌بندی شده‌اند، با ضریب  $\beta_t$  کاهش می‌یابد. در حالی که وزن نمونه‌هایی که به اشتباه طبقه‌بندی شده‌اند بدون تغییر باقی می‌ماند. پس از آن وزن‌های به روزشده با ضریب نرمال‌سازی  $Z_t$  نرمال‌سازی می‌شوند. تا اطمینان حاصل شود که  $D_{t+1}(i)$  یک توزیع احتمال درست است. از این رو با هر یادگیرنده جدیدی که به گروه اضافه می‌شود، AdaBoost بر روی نمونه‌های دشوار‌تر تمرکز می‌کند. در هر

تکرار  $t$ ، رابطه (۷-۳) وزن نمونه‌های اشتباہ طبقه‌بندی شده را به گونه‌ای افزایش می‌دهد که مجموع آن به  $\frac{1}{\epsilon}$  برسد و وزن نمونه‌های درست طبقه‌بندی شده را به گونه‌ای کاهش می‌دهد که مجموع آن‌ها نیز به  $\frac{1}{\epsilon}$  برسد. از آنجا که یادگیرندگان ضعیف می‌باشد خطا ای کمتر از  $\frac{1}{\epsilon}$  را داشته باشند، بنابراین تضمین می‌شود که حداقل یک نمونه آموزشی که قبلاً به اشتباہ طبقه‌بندی شده است، به درستی طبقه‌بندی شود. هنگامی که آن یادگیرنده نتواند این کار را انجام دهد، روش AdaBoost آن را رها می‌کند. این روش تا جایی تکرار می‌شود که  $T$  یادگیرنده ساخته شود. درنهایت این یادگیرندگان با رأی اکثریت وزن دار با هم ترکیب می‌شوند.

## □ الگوریتم XGBoost

XGBoost کوتاه شده عبارت eXtreme Gradient Boosting و به معنای «تقویت گرادیانی تشدید شده» است. این الگوریتم در تحقیق [۲۷] در سال ۲۰۱۶ میلادی توسعه داده شده است و در بسیاری از مسائل تقریب تابع و طبقه‌بندی استفاده شده است و بسیاری از برندهای رقابتی هوش مصنوعی از این روش به عنوان بخشی از راه حل‌های خود استفاده کرده اند. اگرچه پیشرفت قابل توجهی در شبکه‌های عصبی عمیق حاصل شده است اما در بسیاری از کارها به واسطه نیازمندی تنظیم پارامترهای کمتر نسبت به مدل‌های عمیق، از XGBoost استفاده می‌شود.

XGBoost مانند سایر روش‌های تقویت گرادیانی، گروهی از درختان بازگشته را استفاده می‌کند که شامل  $T$  تابع افزایشی است:

$$F(x) = \sum_{t=1}^T f_t(x), \quad f_t \in \mathcal{F} \quad (9-3)$$

که در آن  $T$  تعداد درختان و  $\mathcal{F}$  مجموعه تمام درختان ممکن است. باید به این نکته توجه داشت که در (۹-۳) به جای استفاده از  $f_t$  از  $h_t$  استفاده شده است، چراکه حرف  $h$  در این الگوریتم بیانگر هسین<sup>۱</sup> تابع ضرر<sup>۲</sup> است. با توجه به تابع ضرر  $L(y, F(x))$  که معیاری برای تفاوت برچسب  $y$  و پیش‌بینی یادگیرندگان  $F(x)$  است، XGBoost در صدد آن است که گروهی را بیابد که تابع ضرر را به حداقل برساند. بنابراین، تابع هزینه  $J$  به صورت زیر تعریف می‌گردد:

$$J = \sum_{i=1}^n L(y_i, F_k(x_i)) \quad (10-3)$$

<sup>1</sup> Hessian

<sup>2</sup> Loss Function

می‌توان رابطه (۱۰-۳) را در تکرار  $k$ ، به صورت زیر بازنویسی کرد:

$$J = \sum_{i=1}^n L(y_i, F_{k-1}(x_i) + f_k(x)) \quad (11-3)$$

که  $(x, f_k)$ ، درخت جدیدی است که به گروه اضافه شده است. یافتن بهترین درخت  $f_k(x)$  برای تابع ضرر  $L$  غیرممکن است، چراکه برشماردن تمام درختان را الزامی می‌کند. لذا با استفاده از رویکرد بهینه‌سازی تکراری XGBoost سعی می‌شود درختی انتخاب شود که در هر گام تابع هزینه را به حداقل نزدیک‌تر کند. در روش XGBoost از روش نیوتون-رافسون برای همگرایی سریع‌تر به حداقل مقدار، استفاده شده است. این به این دلیل است که معمولاً تابع ضرر مربعی یا لگاریتمی است و متفرق دوم آن به سادگی قابل محاسبه است. با استفاده از روش نیوتون-رافسون در حل مسئله یافتن حداقل مقدار تابع ضرر، بهترین کاهش تابع هزینه از درخت  $f_k$  به صورت زیر بدست آمده است (به [۲۷] مراجعه شود):

$$J^* = -\frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} \quad (12-3)$$

که در آن  $T$  تعداد برگ‌های درخت آخر،  $\lambda$  ضریب تنظیم وزن برگ است.  $G_j$  و  $H_j$  به ترتیب برابر با مجموع گرادیان و هسین تابع ضرر متناظر با نمونه‌های موجود در برگ  $j$  است و به صورت زیر تعریف می‌شود:

$$G_j = \sum_{i \in I_j} \frac{\partial L(y_i, F_{k-1}(x_i))}{\partial F_{k-1}(x_i)} \quad (13-3)$$

$$H_j = \sum_{i \in I_j} \frac{\partial^2 L(y_i, F_{k-1}(x_i))}{\partial F_{k-1}(x_i)^2} \quad (14-3)$$

همچنین وزن بهینه برای برگ‌های درخت  $f_k$  (خروجی‌های درخت تصمیم  $k$ ام) نیز از طریق رابطه زیر محاسبه می‌شود:

$$\omega_j^* = -\frac{G_j}{H_j + \lambda} \quad (15-3)$$

باتوجه به این که در عمل نمی‌توان همه درخت‌های ممکن را برای رسیدن به چنین تابع هزینه حداقلی برشمارد، لذا در هنگام ساخت درخت تصمیم (پیوست الف)، بررسی شرط تقسیم در گره یعنی انتخاب ویژگی

با بیشترین بهره اطلاعاتی<sup>۱</sup> (بیشترین کاهش تابع هزینه) به صورت زیر تعریف می‌شود:

$$Gain = \frac{1}{2} \left[ \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{G^2}{H^2 + \lambda} \right] \quad (16-3)$$

که در آن  $H = H_L + H_R$  و  $G = G_L + G_R$  به ترتیب گرادیان و هسین گره والد و گرادیان و هسین فرزند به صورت زیر تعریف می‌شود:

$$G_L = \sum_{i \in I_L} g_i, \quad H_L = \sum_{i \in I_L} h_i \quad (17-3)$$

$$G_R = \sum_{i \in I_R} g_i, \quad H_R = \sum_{i \in I_R} h_i \quad (18-3)$$

مطابق روابط ریاضی گفته شده شبه کد روش XGBoost در الگوریتم ۳ آورده شده است.  
در مسائل طبقه‌بندی از تابع ضرر لگاریتمی استفاده می‌شود:

$$L(y_i, F_{k-1}(x_i)) = -y_i \log p_i - (1 - y_i) \log(1 - p_i) \quad (19-3)$$

که در آن  $p_i$  احتمال پیش‌بینی نمونه  $i$  توسط یادگیرنده گروهی قبلی،  $F_{k-1}(x_i)$ ، است. با تعریف چنین تابع ضرری، در مسائل طبقه‌بندی محاسبات گرادیان و هسین بدون نیاز به محاسبه لگاریتم است و هزینه محاسباتی کاهش می‌یابد:

$$G_i = \frac{\partial L(y_i, F_{k-1}(x_i))}{\partial F_{k-1}(x_i)} = p_i - y_i \quad (20-3)$$

$$H_i = \frac{\partial^2 L(y_i, F_{k-1}(x_i))}{\partial F_{k-1}(x_i)^2} = p_i(1 - p_i) \quad (21-3)$$

با جای‌گذاری  $G_i$  و  $H_i$  در سایر روابط، فرم ساده آن‌ها برای مسائل طبقه‌بندی به دست خواهد آمد.

---

<sup>1</sup> Information Gain

**XGBoost : ۳**

وروودی:

I : دادگان آموزش

L: تابع ضرر مشتق پذیر

K: تعداد تکرارهای

η: نرخ یادگیری

λ ضریب تنظیم برگ

γ: ضریب کاهش اولیه

مقداردهی اولیه:

$$F_0(x) = \operatorname{argmin}_x \sum_{i=1}^n L(y_i, \gamma) \quad (22-3)$$

بهازای k = 1, 2, ..., K انجام دهید:

\*/محاسبه مجموع گرادیان و هسین تمام نمونهها\*/

$$G \leftarrow \sum_{i \in I} \frac{\partial L(y_i, F_{k-1}(x_i))}{\partial F_{k-1}(x_i)} \quad (1)$$

$$H \leftarrow \sum_{i \in I} \frac{\partial^2 L(y_i, F_{k-1}(x_i))}{\partial F_{k-1}(x_i)^2} \quad (2)$$

(I, G, H) → ساخت درخت با استفاده از بهره (16-۳) و پارامترهای ورودی (3)

تمام برگ‌های درخت T را R\_j درنظر بگیرید که j = 1, ..., J\_k

بهازای j = 1, ..., J\_k محاسبه کنید:

$$\omega_j = -\frac{G_j}{H_j + \lambda} \quad (23-3)$$

بروزرسانی مدل:

$$F(x) = F_{k-1}(x) + \eta \sum_{j=1}^{J_K} \omega_j \mathbb{I}(x \in R_j) \quad (24-3)$$

پایان حلقه

خروجی: طبقه با بیشترین مقدار  $V_c$ 

گرچه در حالت کلی و در بسیاری از مسائل طبقه‌بندی، استفاده از الگوریتم‌های یادگیری گروهی باعث افزایش دقت طبقه‌بندی شده است، اما به کارگیری این الگوریتم‌ها با چالش‌هایی نیز همراه است. مهم‌ترین چالش آن که مانع از استفاده از این الگوریتم‌ها در موقعیت‌یابی مکان سرپوشیده است، پیچیدگی در محاسبات و نیازمندی به منابع محاسباتی و ذخیره‌ای بالا است. در نتیجه بلاذرنگ بودن سیستم موقعیت‌یابی با مشکل مواجه می‌شود و همچنین هزینه تهیه چنین سیستمی زیاد خواهد بود. الگوریتم‌های ذکر شده از درخت تصمیم به عنوان یادگیرنده پایه استفاده می‌کنند. حجم محاسباتی و ذخیره‌ای این الگوریتم با تعداد گره‌های درخت تصمیم یا

با عمق درخت ارتباط مستقیم دارد. در این پژوهش دو ایده برای کاهش عمق درخت پیشنهاد شده است. روش اول، به طور مستقیم بر روی مجموعه دادگان اعمال می‌شود و با کاهش ابعاد داده ورودی، به طور غیرمستقیم عمق درخت را کاهش می‌دهد. روش پیشنهادی بعدی استفاده از الگوریتم ژنتیک است. در این روش نه تنها سعی می‌شود که عمق درختان کاهش یابد، بلکه با درنظرگیری پارامترهای تنظیم الگوریتم‌های گروهی، زمان پردازش در عین حفظ دقت، کاهش یابد. در ادامه این دو پیشنهاد ارائه می‌گردد.

### ۳-۳- پیش‌پردازش داده

این مرحله از اهمیت بسیاری برخوردار است؛ زیرا کیفیت و کمیت داده‌های ورودی مستقیماً بر توانایی و سرعت مدل در یادگیری موقعیت و تعمیم آن تأثیر می‌گذارد. در این مرحله، اطلاعات جمع‌آوری شده از نقاط دسترسی و دستگاه‌های کاربران از جمله مکان واقعی، زمان، قدرت سیگنال، و سایر ویژگی‌های مرتبط با موقعیت، پس از پردازش وارد مرحله آموزش مدل می‌شوند. پیش‌پردازش شامل پاک‌سازی و نرمال‌سازی داده‌ها و استخراج ویژگی‌های اصلی است. این اقدامات در راستای افزایش یکنواختی و قابلیت تعمیم مدل در مواجهه با شرایط متفاوت در محیط‌های مختلف اجرا می‌شوند. داده‌های جمع‌آوری شده ممکن است شامل نویز، مقادیر گمشده باشند. پاک‌سازی داده‌ها شامل شناسایی و حذف داده‌های نامعتبر، تصحیح مقادیر گمشده و رفع ناسازگاری‌ها است.

یکی از پیشنهادهای اصلی در پردازش داده‌های موقعیت یابی استفاده از روش‌های کاهش ابعاد دادگان ورودی مدل‌های یادگیری گروهی است. روش‌های کاهش ابعاد اطلاعات با ابعاد بالا که در واقع، تعداد ویژگی‌های داده‌های موقعیت است و می‌تواند شامل مشخصات سیگنال و شرایط محیطی باشد را به فضای با ابعاد کمتر تبدیل می‌کنند. با اعمال تکنیک‌های کاهش ابعاد مانند «تحلیل مؤلفه‌های اساسی»<sup>۱</sup> (PCA) یا روش «t-SNE»<sup>۲</sup> جاسازی همسایگی تصادفی توزیع شده<sup>۳</sup> (t-SNE)، داده‌های موقعیت یابی به صورت مؤثرتری برای مدل‌های یادگیری گروهی آماده می‌شوند. این گونه روش‌ها با درنظرگیری میزان تأثیر ویژگی‌ها، یعنی سطح سیگنال نقاط دسترسی و شرایط محیطی، به افزایش سرعت آموزش مدل‌ها کمک کرده و از مشکل نفرین ابعاد<sup>۴</sup> در داده‌ها جلوگیری می‌کنند. همچنین استفاده از روش‌های کاهش مرتبه می‌تواند باعث افزایش مقاوم‌بودن<sup>۵</sup> یک

<sup>1</sup> Principal Component Analysis

<sup>2</sup> t-distributed Stochastic Neighbor Embedding

<sup>3</sup> Curse of dimensionality

<sup>4</sup> Robustness

سیستم موقعیت یاب شود. به عنوان مثال، استفاده از روش PCA در [۳۰] برای استخراج وابستگی اجزای سیگنال به موقعیت هدف در یک سیستم موقعیت یابی مبتنی بر CSI با فناوری Wi-Fi استفاده شده است و نرخ تشخیص اشتباه موقعیت هدف در یک دفتر کار را ۳ درصد کاهش داده است. به عنوان کار مشابه در [۳۱] نیز روش PCA را برای افزایش دقت و سرعت در موقعیت یابی با استفاده از الگوریتم های SVM و kNN و ANN پیشنهاد داده است.

کاهش ابعاد بردار ویژگی‌ها، به طور غیرمستقیم در روش‌های یادگیری گروهی بیان شده در ۲-۳- اثرگذار است. از آنجا که عمق درختان یادگیرنده در یک الگوریتم گروهی، حداکثر می‌تواند برابر با تعداد ویژگی‌های بردار ورودی باشد، لذا با کاهش ابعاد، حداکثر عمق درختان کاهش یافته و درنتیجه مدل‌های یادگیرنده سریع‌تر آموزش می‌بینند. در ادامه روش محبوب و پر استفاده از روش‌های کاهش ابعاد معرفی می‌شود.

### ۱-۳-۳- تجزیه مؤلفه‌های اساسی (PCA)

تجزیه و تحلیل مؤلفه‌های اساسی (PCA) یک تکنیک کاهش ابعاد است که معمولاً در یادگیری ماشین و آمار استفاده می‌شود. هدف اصلی این تکنیک، تبدیل مجموعه داده با ابعاد بالا به یک فضای با ابعاد کمتر، در عین حفظ تنوع داده اصلی است. به عبارت دیگر، PCA به ساده‌تر کردن مجموعه داده‌های پیچیده، با پیدا کردن یک مجموعه دادگان با ستون‌های جدید (مؤلفه‌های اصلی) که در آن داده‌ها بیشترین تغییر را دارند می‌پردازد. PCA با انتقال داده‌های اصلی با ابعاد بالا به یک سیستم مختصات جدید که توسط مؤلفه‌های اصلی آن تعریف شده است عمل می‌کند. مؤلفه‌های اصلی، ترکیب‌های خطی از ویژگی‌های اصلی هستند که حداقل واریانس در داده را حفظ می‌کنند. روش PCA با گام‌های زیر توصیف می‌شود:

- **گام اول، استانداردسازی:** اگر ویژگی‌های اصلی مقیاس‌های متفاوتی داشته باشند، اولین قدم استاندارد کردن داده‌ها با تفربیق میانگین و تقسیم بر انحراف استاندارد برای هر ویژگی است. این کار تضمین می‌کند که همه ویژگی‌ها دارای مقیاس قابل قیاس هستند.
  - **گام دوم، محاسبه کوواریانس:** ماتریس کوواریانس نشان می‌دهد که چگونه ویژگی‌های مختلف نسبت به یکدیگر متفاوت هستند:

$$Cov(x) = \frac{1}{n-1} (X - \bar{X})^T (X - \bar{X}) \quad (45-3)$$

که در آن،  $X$  ماتریس داده در فضای اصلی است،  $\bar{X}$  بردار میانگین از ستون‌های  $X$  و  $n$  تعداد دادگان

وروودی است.

• ۳-۲-۳. گام سوم، محاسبه بردارها و مقادیر ویژه:

$$\text{Eigenvalue decomposition: } \text{Cov}(X) = W\Lambda W^{-1} \quad (26-3)$$

که در آن،  $W$  ماتریس ویژه (بردارهای ویژه به صورت ستونی)،  $\Lambda$  ماتریس قطری مقادیر ویژه است

• ۳-۴. گام چهارم، محاسبه بردارها و مقادیر ویژه: بردارهای ویژه بر اساس مقادیر ویژه مربوطه به ترتیب نزولی مرتب می‌شوند. بردارهای ویژه (مولفه‌های اصلی) یک مبنای متعامد جدید برای داده‌ها تشکیل می‌دهند. داده‌های اصلی بر روی این سیستم مختصات جدید تصویر می‌شود.

• ۳-۵. گام آخر، کاهش ابعاد: زیرمجموعه‌ای از مؤلفه‌های اصلی متناسب با مرتبه کاهش (ابعاد فضای ثانویه) انتخاب کنیم ( $W_k$ ). این مؤلفه‌ها به مهمترین ویژگی‌ها وزن بیشتر و برای داده‌های ویژگی‌ها با تغییرات کم، وزن کمتری را اختصاص می‌دهد. با استفاده از مؤلفه‌های اساسی انتخاب شده، داده‌ها به فضای جدید منتقل می‌شود:

$$Y = X \cdot W_k \quad (27-3)$$

که در آن  $Y$  مجموعه دادگان جدید در فضای ثانویه است.

PCA اثر ویژگی‌های مرتبط را کم می‌کند و بیش برآذش مدل را کاهش می‌دهد. توجه به این نکته حائز اهمیت است که روش‌های کاهش مرتبه، علی‌رغم مزایایی که دارند، تفسیرپذیری مجموعه دادگان را بسیار کم می‌کند، علاوه بر آن، PCA یک روش کاهش ابعاد خطی است و ارتباط خطی میان ویژگی‌ها را در نظر می‌گیرد. اگرچه استفاده از سایر روش‌های کاهش ابعاد، فرض سادگی محاسبات سیستم را نقض می‌کند، اما در تئوری استفاده از آن‌ها قابل بررسی است.

#### ۳-۴-۴- الگوریتم ژنتیک

به کارگیری الگوریتم ژنتیک برای تنظیم ابرپارامترهای یک مدل یادگیری ماشین بسیار متداول است. این الگوریتم یکی از شاخه‌های پردازش تکاملی<sup>۱</sup> است که بر اساس ایده تکامل طبیعی و فرضیه داروین توسعه

<sup>1</sup> Evolutionary Computing

یافته است. بر اساس این ایده، در هر جامعه‌ای معمولاً افراد قوی‌تر از منابع بیشتری استفاده می‌کنند و با احتمال بیشتری زنده می‌مانند. در مقابل، افراد ضعیف‌تر با احتمال کمتری باقی می‌مانند. افراد با قیمانده از هر نسلی، تولید مثل کرده و فرزندان نسل بعد را تشکیل می‌دهند. انتظار می‌رود که افراد هر نسل از جامعه، قوی‌تر از افراد نسل قبل خود باشد. در این انتقال نسل نیز ممکن است جهش ژنتیکی رخ دهد که باعث تنوع و پراکندگی در افراد یک نسل می‌شود. یکی از مزایای بزرگ الگوریتم‌های ژنتیک نسبت به روش‌های مبتنی بر گرادیان، مقاومت بیشتر در برابر گیر افتادن در بهینه‌های محلی است. به دلیل فرایند تکاملی و امکان اجرای همزمان، الگوریتم‌های ژنتیک می‌توانند به سرعت به بهینه‌های جهانی هدایت شوند. همچنین، این الگوریتم‌ها انعطاف بیشتری در مقابل نویز یا ناپیوستگی‌ها دارند.

مرحله ابتدایی در الگوریتم ژنتیک تعیین روش بازنمایی و تابع برازش<sup>۱</sup> است. «روش بازنمایی» مشخص می‌کند که چگونه هر وضعیت مسئله به صورت مناسب مدل و ذخیره شود. در این روش فنوتیپ<sup>۲</sup>ها وضعیت‌های واقعی مسئله هستند و ژنوتیپ<sup>۳</sup>ها یا کروموزوم<sup>۴</sup>ها به عنوان آرایه‌های مدل شده از هر وضعیت است و در فضای راه حل تعریف می‌شود ژن نیز به هر یک از آرایه‌های یک کروموزوم گفته می‌شود. نکته حائز اهمیت آن است که برای امکان یافتن جواب بهینه روش بازنمایی باید تمام راه حل‌های ممکن را پوشش دهد.

در مسئله تنظیم ابرپارامترهای یادگیری گروهی، فنوتیپ‌ها، ابرپارامترهای مدل یادگیری گروهی، یعنی «عمق درخت» و «تعداد یادگیرندگان» در نظر گرفته می‌شود که با کدگذاری به ژنوتیپ‌ها تبدیل می‌شود.

هر وضعیت مسئله توسط «تابع برازش» یا «ارزیاب» رتبه‌بندی می‌شود. تابع برازش باید به گونه‌ای تعریف شود که برای حالت‌های بهتر مقادیر بزرگ‌تری را برگرداند. در تنظیم ابرپارامترهای یادگیرندگه گروهی موقعیت‌یاب، تابع برازش ترکیبی از متغیرهایی است که برای مدل یادگیری گروهی که بهترین دقت، سرعت و کمترین هزینه را دارد، بیشترین مقدار را بازگرداند؛ بنابراین، ترکیب خطی از معیار دقت، زمان تخمین موقعیت و عمق درخت (برای کاهش هزینه پیاده‌سازی) به عنوان تابع برازش استفاده می‌شود. ضرایب ترکیب خطی متناسب با میزان اهمیت هر یک از متغیرها انتخاب می‌گردد.

الگوریتم ژنتیک یک الگوریتم تکرارشونده برای یافتن کمینه مقدار تابع برازش است. در هر تکرار از الگوریتم ژنتیک دو عمل اصلی انجام می‌شود. عمل ترکیب متقطع<sup>۵</sup> و عمل جهش<sup>۶</sup> در ساخت جمعیت جدید به کار

<sup>1</sup> Fitness function

<sup>2</sup> Phenotype

<sup>3</sup> Genotype

<sup>4</sup> Chromosome

<sup>5</sup> Crossover

<sup>6</sup> Mutation

می‌روند. در عمل ترکیب متقطع دو کروموزوم والد به شیوه‌های مختلفی می‌توانند با هم ترکیب شوند و دو فرزند تولید کنند. متداول‌ترین شیوه ترکیب، ترکیب یک نقطه‌ای است که در آن یک نقطه از موقعیت‌های داخل کروموزوم به طور تصادفی انتخاب می‌شود. پس از آن فرزندان از برخورد کروموزوم‌های والد در نقطه پیوند متولد می‌شوند. در هر مرحله با یک احتمال مشخص، ممکن است هر کروموزوم مورد جهش واقع شود. در عمل جهش، هم ژن و هم مقدار جدید آن به طور تصادفی انتخاب می‌شود. در شکل (۴-۳) کروموزوم‌های آبی و قرمز با یکدیگر ترکیب متقطع شده‌اند و ژن‌های زرد رنگ با مقدار تصادفی جهش یافته‌اند.

در عمل ترکیب متقطع، کروموزوم انتخابی  $i$  با احتمال  $p_{select}$  انتخاب می‌شود:

$$p_{select}(i) = \frac{fitness(i)}{\sum_{k=1}^n fitness(k)} \quad (28-3)$$

که در آن  $fitness(x)$  تابع برازش کروموزوم  $x$  و  $n$  جمعیت نسل است.

#### الگوریتم ۴: الگوریتم ژنتیک

وروودی:

$population$ : جمعیت

$fitness(i)$ : تابع برازش

$K$ : تکرار الگوریتم

بهازای  $K = 1, 2, \dots, K$  انجام دهید:

(۱)  $\rightarrow$  مجموعه تهی  $\rightarrow new\_population$

(۲) بهازای  $i = 1$  تا اندازه  $population$  انجام دهید:

(۳)  $x \rightarrow$  انتخاب تصادفی از  $population$  با احتمال  $p_{select}$  در رابطه (۲۸-۳)

(۴)  $y \rightarrow$  انتخاب تصادفی از  $population$  با احتمال  $p_{select}$  در رابطه (۲۸-۳)

(۵)  $\rightarrow$  ترکیب متقطع  $x$  و  $y$ .

(۶) اگر احتمال تصادفی جهش برآورده شود آنگاه  $child$  را جهش دهید.

(۷)  $new\_population \rightarrow population$  اضافه کنید.

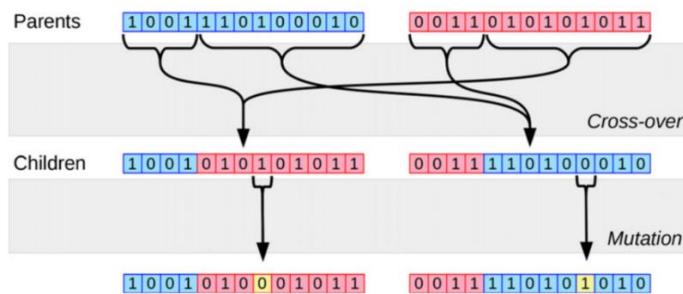
(۸)  $new\_population \rightarrow population$

پایان حلقه

خروجی: بهترین  $child$  در  $population$  مطابق با تابع  $fitness(child)$

شبه کد الگوریتم ژنتیک در الگوریتم ۴ آورده شده است. در الگوریتم ژنتیک معمولاً نسل‌های ابتدایی، افراد جامعه پراکندگی خوبی دارند، همچنین عملگر ترکیب بر روی وضعیت‌های مختلف والد، می‌تواند حالت‌های متفاوت نسبت به هر دو والد تولید کند. به تدریج، افراد با شباهت بیشتر در جمعیت ظهور می‌یابند و درنهایت

بهترین فرد در جمعیت نهایی به عنوان دارنده بهترین ابرپارامتر الگوریتم یادگیری گروهی معرفی می‌گردد.



شكل (٤-٣) عمليات تركيب متقاطع و جهش در تولید جمعیت جدید الگوریتم زنتیک [٣٢]

۳-۵- معیارهای ارزیابی

پارامترهای ارزیابی به دو دسته تقسیم‌بندی می‌شود. دسته اول معیارهای قابل اندازه‌گیری عملکرد و دسته دوم موضوعات کیفی عملکرد است:

### ۳-۵-۱- معیارهای قابل اندازه‌گیری عملکرد

عملکرد یک سیستم موقعیت یاب بر اساس معیارهای قابل اندازه‌گیری زیر ارزیابی می‌شود:

## Accuracy □

accuracy یا دقت یکی از مهم‌ترین معیارهای عملکردی در هر سیستم موقعیت یاب است. در ادبیات یادگیری ماشین دقت به صورت  $N^c/N^t$  است که در آن  $N^c$  تعداد نمونه‌ها با پیش‌بینی درست و  $N^t$  تعداد کل نمونه‌ها است. طبق پیش‌بینی مثبت و منفی نمونه‌ها، دقت به صورت زیر تعریف می‌شود:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (29-3)$$

که در آن  $TP$  و  $TN$  به ترتیب تعداد نمونه‌های مثبت و منفی هستند که به درستی تخمین زده‌اند و به همین ترتیب  $FP$  و  $FN$  نیز نمونه‌های مثبت و منفی هستند که به اشتباه تخمین زده شده‌اند.

Precision □

Precision یا صحت برای اندازه‌گیری این که چقدر می‌توان به طور مداوم به Accuracy دست یافت به کار پرده می‌شود. در بیان یادگیری ماشین، نسبت موارد مثبت است که به درستی طبقه‌بندی شده است؛ بنابراین

بهصورت زیر تعریف می‌شود:

$$\text{Preceision} = \frac{TP}{TP+FP} \quad (30-3)$$

در موقعیت‌یابی با اثر انگشت، Preceision نشان‌دهنده نسبت تشخیص درست موقعیت هر بلوک هدف به مجموع تشخیص درست و نادرست همان بلوک است.

### Recall □

از Recall بهعنوان حساسیت نیز نامبرده می‌شود و به نرخ مثبت درست (TP) اشاره دارد:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (31-3)$$

و در موقعیت‌یابی با اثر انگشت، به معنای نسبت تشخیص درست هر بلوک هدف به مجموع تشخیص درست همان بلوک و تشخیص نادرست بلوک‌های دیگر بهعنوان بلوک هدف است.

### F1-Score □

بهعنوان معیاری برای توازن میان Recall و Preceision F1-Score معرفی می‌شود:

$$F1 - Score = \frac{2.\text{Preceision}.\text{Recall}}{\text{Preceision}+\text{Recall}} \quad (32-3)$$

استفاده از F1-Score بهجای Accuracy در ارزیابی یک مدل یا سیستم دارای مزایای متعدد است که می‌تواند اطلاعات مفیدتری ارائه دهد. وقتی که تشخیص خطاهای (FN و FP) مهم است، F1-Score بهترین معیار اندازه‌گیری است. اگر تعداد خطاهای در یک کلاس به اندازه قابل توجهی باشد، Accuracy ممکن است زیاد باشد اما تشخیص خطاهای را نادیده بگیرد.

### □ میانگین خطای موقعیت‌یابی

نرخ خطای  $E^{avg}$  در یک سیستم موقعیت‌یابی به صورت  $N^m/N^t$  که در آن  $N^m$  تعداد نمونه‌های طبقه‌بندی نشده یا به اشتباه طبقه‌بندی شده است. این خطای موقعیت‌یابی عموماً با فاصله اقلیدسی بین مکان واقعی ( $x, y$ ) و مکان پیش‌بینی شده ( $\hat{x}, \hat{y}$ ) تعریف می‌شود که ارزیابی را در فضای پیوسته انجام می‌دهد. میانگین خطای موقعیت‌یابی بهصورت زیر تعریف می‌شود.

$$E^{avg} = \frac{1}{N^t} \sum_{i=1}^{N^t} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \quad (33-3)$$

دقت (به درصد) تعداد دفعاتی را که یک سیستم به درستی یک مکان را مشخص می‌کند نشان می‌دهد؛ لذا، درصد پیش‌بینی‌های نادرست نیز به دست می‌آید. در صورت پیش‌بینی نادرست، فاصله مکان پیش‌بینی از مکان واقعی را با این معیار نمی‌توان به دست آورد.

### □ میانگین زمان تخمین

میانگین زمان تخمین،  $t^{avg}$ ، به عنوان معیار زمانی عملکرد سیستم موقعیت یاب معرفی می‌شود. چنان‌چه مجموع میانگین زمان تخمین موقعیت و مدت زمان ارتباطی کاربر با واحد پردازش، بسیار کمتر از دوره نمونه برداری مقادیر حسگر باشد، سیستم موقعیت یاب بلادرنگ خوانده می‌شود. این معیار به صورت زیر محاسبه می‌شود:

$$t^{avg} = \frac{1}{N^t} \sum_{i=1}^{N^t} t_i \quad (34-3)$$

که در آن  $t_i$  مدت زمان تخمین نمونه  $i$  است.

### ۳-۵-۲- معیارهای کیفی عملکرد

سیستم‌های موقعیت یاب برای ارائه خدمات یکپارچه، باید معیارهای کیفی زیر را رعایت کنند:

### □ مقیاس‌پذیری

یک سیستم زمانی مقیاس‌پذیر است که به راحتی در یک منطقه آزمایشی بزرگ قابلیت استقرار داشته باشد و به تعداد دستگاه‌های متنوع با عملکرد مشابه خدمات ارائه کند.

### □ مقاوم‌بودن

مقاوم‌بودن این اطمینان را ایجاد می‌کند که یک سیستم موقعیت یاب می‌تواند خدمات خود را در یک محیط پیش‌بینی‌نشده مانند تغییر اشیای اطراف، وجود دستگاه‌های مختل‌کننده، عملکرد نادرست برخی از نقاط دسترسی و غیره ارائه دهد.

## □ مصرف انرژی

یکی از مسائل مهم در ارزیابی کیفی سیستم‌های موقعیت‌یاب مصرف بهینه توان است. نیاز به تعویض باتری و یا استفاده از منابع تغذیه علاوه بر افزایش هزینه، کار میدانی را نیز افزایش می‌دهد. محاسبات و پیچیدگی الگوریتم‌ها می‌تواند در مصرف انرژی سیستم‌های موقعیت‌یاب تأثیرگذار باشد.

## □ دردسترس‌بودن

یک سیستم موقعیت‌یاب در محیط سرپوشیده، نیازمند دردسترس‌بودن تجهیزات و فناوری را دارد. فناوری بلوتوث و Wi-Fi بر خلاف تکنولوژی ZigBee و غیره که نیازمند ادوات خاص خود هستند، معمولاً در اکثر تلفن‌های هوشمند وجود دارد؛ لذا علی‌رغم دقت خوب در سیستم‌های موقعیت‌یاب مبتنی بر UWB، انتخاب سیستم‌های مبتنی بر Wi-Fi و بلوتوث ارجحیت دارند.

## □ هزینه

پایین‌بودن هزینه سیستم‌های موقعیت‌یابی، تعداد کاربران آن را افزایش می‌دهد. هزینه زیرساخت‌های اولیه، تأمین نیروی انسانی برای تعمیر و نگهداری و کار میدانی از مهم‌ترین آن‌ها در سیستم‌های موقعیت‌یابی است؛ بنابراین ایجاد توازن بین سایر معیارهای ارزیابی و هزینه‌های پیاده‌سازی یک سیستم موقعیت‌یاب نیز نیازمند مهندسی و محاسبات است.

## ۳-۶- جمع‌بندی

در این فصل راهکارهای پیشنهادی جهت بهبود دقت در تعیین موقعیت مورد بررسی قرار گرفته است. ابتدا به بررسی روش‌های یادگیری گروهی پرداخته شده است. مروری بر توسعه الگوریتم‌های یادگیری گروهی انجام شد و اجزای مهم آن که شامل نمونه‌برداری و انتخاب داده، آموزش یادگیرندگان و ترکیب یادگیرندگان ضعیف مورد بررسی قرار گرفته است. سپس روش‌های Bagging و Boosting به تفصیل بررسی شده‌اند که در این‌بین الگوریتم‌های جنگل تصادفی، AdaBoost و XGBoost که محبوب‌ترین الگوریتم‌های یادگیری گروهی هستند، بیان شده‌اند. الگوریتم‌های یادگیری گروهی، علی‌رغم دقت تخمین خوب، نیازمند منابع محاسباتی و ذخیره‌ای زیاد هستند و به سبب آن می‌توانند تخمین بلادرنگ موقعیت را با چالش همراه کنند. از این‌رو، دو راهکار کاهش ابعاد و به کارگیری الگوریتم ژنتیک ارائه شد. فرایند پیش‌پردازش داده با استفاده از تجزیه مؤلفه‌های اساسی (PCA) به عنوان راهکار غیرمستقیم مورد ارزیابی قرار گرفته است. الگوریتم ژنتیک نیز به عنوان راهکار

مستقیم در تعیین ابرپارامترهای یادگیری گروهی برای موقعیت یابی معرفی شده است. در پایان این فصل، معیارهای ارزیابی با تأکید بر معیارهای قابل اندازه‌گیری عملکرد مانند Accuracy، Precision، Recall، F1- Score، میانگین خطای موقعیت یابی و میانگین زمان تخمین بررسی و معیارهای کیفی عملکرد معرفی شده‌اند.

## **فصل ۴:**

# **شبیه سازی و ارزیابی نتایج**

## ۱-۴- مقدمه

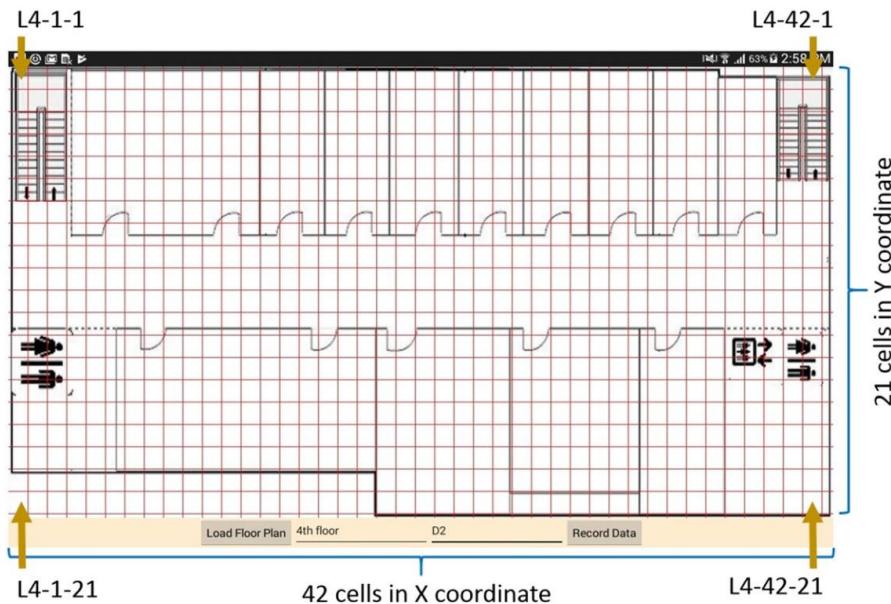
ابتدا مجموعه‌دادگان موقعیت یابی مبتنی بر RSS که اطلاعات اساسی برای ارزیابی الگوریتم‌ها و روش‌های موقعیت یابی هستند، معرفی می‌شود. سپس، به بررسی نتایج شبیه‌سازی روش‌های آموزش مدل موقعیت یاب با استفاده از الگوریتم‌های یادگیری گروهی که راهکار اصلی در بهبود دقت موقعیت یابی هستند پرداخته می‌شود. پس از آن، 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 موقعیت یاب ارزیابی می‌شود و روش‌های تنظیم ابرپارامترها با استفاده از الگوریتم ژنتیک نیز به عنوان یک ابزار قدرتمند برای بهینه‌سازی پارامترها در این الگوریتم‌ها، بررسی می‌شود. در نهایت، با جمع‌بندی نتایج به دست آمده از تمامی این روش‌ها، عملکرد و کارایی هر یک از آن‌ها در تعیین موقعیت به صورت کامل ارزیابی می‌گردد و با نتایج مشابه مقایسه می‌شود.

## ۲-۴- دادگان موقعیت یابی مبتنی بر RSS

تعداد زیادی از مجموعه‌دادگان موقعیت یابی مبتنی بر RSS وجود دارد. در میان آن‌ها، چندین مجموعه‌دادگان به سبب ویژگی‌ها و اطلاعات غنی‌تر، محبوب‌تر هستند و بیشتر مورد استفاده قرار می‌گیرند. از جمله معروف‌ترین مجموعه‌دادگان، مجموعه‌دادگان UJIndoorLoc که در سال ۲۰۱۴ میلادی معرفی شد برای شبکه غیر بی‌سیم Wi-Fi معرفی شد [۳۳]. گرچه این مجموعه دادگان به عنوان معیار سنجش الگوریتم‌های موقعیت یاب به کار می‌رود، اما شامل شرایط مختلف محیطی نمی‌شود. مجموعه دادگان IPIN در سال ۲۰۱۶ تا ۲۰۲۲ میلادی برای مسابقات رדיابی ربات‌ها معرفی شدند که عمدۀ اطلاعات آن مربوط به سنسورهای ربات است که مبتنی بر موقعیت یابی آفلاین است [۳۴].

دادگان استفاده شده در این پژوهش، مجموعه‌دادگان JUIndoorLoc است که در سال ۲۰۱۹ میلادی ارائه شده است [۳۵] و به خاطر ضبط داده در شرایط مختلف محیطی، اعتبار صحت این مجموعه‌دادگان در طول زمان بیشتر است. داده‌ها از طبقه سوم، چهارم و پنجم ساختمانی در دانشگاه جوداپور گردآوری شده‌اند. در این ساختمان، هر طبقه ۸۸۲ متر مربع به طول ۴۲ متر و عرض ۲۱ متر را پوشش می‌دهد و از اتاق‌های استادی، کلاس‌های درس، اتاق‌های سینما، آزمایشگاه‌های تحقیقاتی و راهرو تشکیل شده است. این ناحیه به سلول‌های ۱ متر × ۱ متر تقسیم می‌شود؛ بنابراین، مختصات X و Y هر طبقه به ترتیب به ۴۲ و ۲۱ سلول تقسیم می‌شود.

که در شکل (۴-۱) نشان داده شده است.



شکل (۴-۱) نمایشی از تقسیم‌بندی اثر انگشت طبقه چهارم محیط مجموعه‌دادگان استفاده شده [۳۵]

گرددآوری 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 پایگاه‌داده سیستم موقعیت‌یاب ذخیره می‌شود. با توجه به مشاهدات، مقادیر RSSI با سخت‌افزارهای مختلف و شرایط مختلف محیطی متفاوت است؛ لذا این مجموعه‌دادگان 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 ۲۵۳۶۴ نمونه است و هر نمونه دارای ۱۷۷ ویژگی است که شامل ۱۷۲ مقدار RSSI از نقاط دسترسی مختلف (AP001 – AP172)،  $C_{id}$  شناسه منحصر به فرد هر بلوک (شامل طبقه و موقعیت در راستای x و y)،  $R_s$  باز (یک) و بسته (صفراً) بودن درب اتاق،  $H_{pr}$  حضور (یک) یا عدم حضور (صفراً) انسان،  $T_s$  شناسه دستگاه ثبت کننده و  $D_{id}$  زمان جمع‌آوری داده بر حسب میلی ثانیه است.

## ۴-۲-۱- پیش‌پردازش دادگان موقعیت یابی

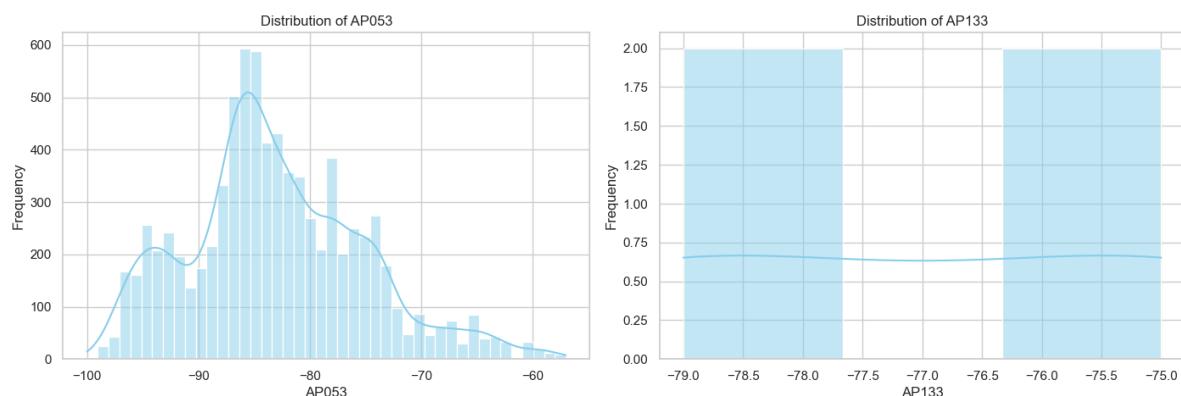
پیش‌پردازش دادگان، مجموعه اقداماتی است که پیش از استفاده از دادگان موقعیت یابی اجرا می‌شود. این اقدامات در راستای پاک‌سازی و تقویت داده‌ها صورت می‌گیرد تا دقت و قابلیت استفاده از آن‌ها در مراحل بعدی افزایش یابد. اقدامات انجام شده به شرح زیر است:

### □ پاک‌سازی داده

این مرحله شامل حذف مقادیر تکراری و پاک‌سازی مقادیر گمشده است. با توجه به نتایج برآمده از پردازش مجموعه دادگان، مقادیر تکراری در آن وجود ندارد. همچنین در هر نمونه، مقادیر RSSI از نقاط دسترسی خارج از محدوده برابر با  $-110dBm$  - جای‌گذاری شده است.

### □ تجزیه و تحلیل اکتشافی دادگان<sup>۱</sup> (EDA)

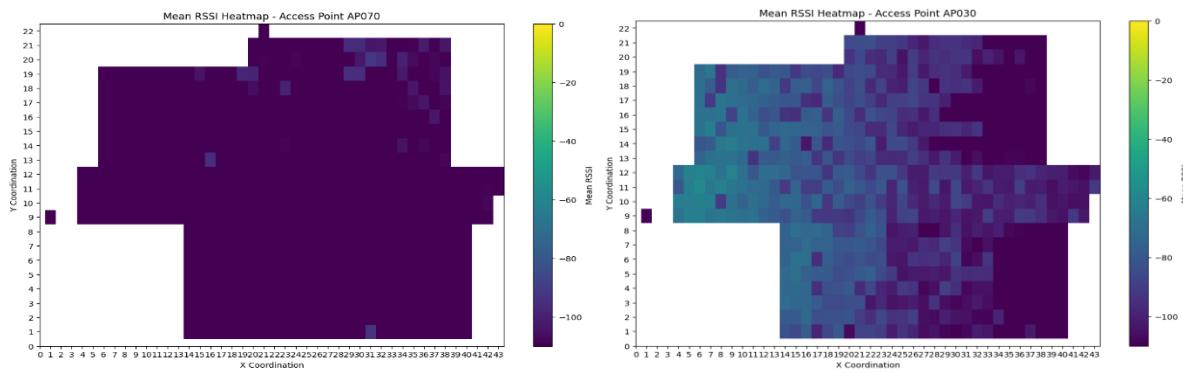
این مرحله از پیش‌پردازش برای بررسی توزیع هر ویژگی انجام می‌شود. به عنوان مثال، با توجه به شکل (۴-۲) که تعداد دفعات تکرار هر مقدار RSSI را برای هر نقطه دسترسی نشان می‌دهد، نقطه دسترسی AP053 نسبت به نقطه دسترسی AP133، از اطلاعات بیشتری برخوردار است. این تجزیه و تحلیل، ایده استفاده از دو الگوریتم پیشنهادی در جهت بهبود عملکرد مدل‌های یادگیری گروهی را تأیید می‌کند.



شکل (۴-۲) نمودار هیستوگرام تعداد دفعات تکرار هر مقدار RSSI در نقاط دسترسی AP133 و AP053

علاوه بر آن، با میانگین‌گیری از مقادیر RSSI در هر بلوک، نقشه‌ای از توزیع مقادیر RSSI در مختصات  $X$  و  $Y$  حاصل می‌گردد که میزان اطلاعات دریافتی از هر نقطه دسترسی را به طور شهری مصور می‌سازد. نمونه‌ای از این نقشه در شکل (۳-۴) نشان داده شده است.

<sup>1</sup> Exploratory Data Analysis



شکل (۳-۴) نقشه توزیع مقادیر RSSI در مختصات  $X$  و  $Y$  از نقاط دسترسی AP070 و AP030 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 (معمولأً بین ۰ و ۱)، 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 بیش‌پردازش کنند. با نرمال‌سازی داده‌ها، این احتمال کاهش می‌یابد و عملکرد مدل در داده‌های جدید بهبود می‌یابد.

## ۳-۴-آموزش موقعیت‌یاب با الگوریتم‌های یادگیری گروهی

مدل‌های یادگیری گروهی معرفی شده در فصل ۳ با مجموعه‌دادگان معرفی شده در بخش ۲-۴ 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 (۱-۴) 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

جدول (۱-۴) مشخصات سیستم سخت‌افزاری موقعیت‌یاب برای اجرای شبیه‌سازی

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

زودهنگام الگوریتم در روش‌های تکرارشونده 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 ۵ گانه<sup>۱</sup> استفاده می‌شود. در این روش ۲۰ درصد از دادگان به عنوان دادگان اعتبارسنجی، با درنظرگیری نسبت برچسب‌ها انتخاب می‌گردد و مابقی به عنوان 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 به عنوان یادگیرنده پایه با چهار مقدار حداکثر عمق متفاوت آموزش داده شده است و نتایج ارزیابی آن برای ۲۰ درصد از مجموعه دادگان در جدول (۲-۴) عنوان شده است. اشاره شد که ابعاد هر بلوک ۱ متر در ۱ متر است؛ لذا در محاسبات میانگین خطای مطابق رابطه (۳-۳) می‌توان به سادگی و مستقیم مختصات  $X$  و  $Y$  را جای‌گذاری کرد. همچنین زمان آزمایش برای ۵۰۷۲ داده محاسبه شده است.

جدول (۲-۴) نتایج شبیه‌سازی درخت تصمیم با دادگان ارزیابی و حداکثر عمق‌های متفاوت

| زمان آزمایش | RMSE<br>(متر) | MSE<br>(مترمربع) | $E^{avg}$<br>(متر) | F1-Score | Recall | Precision | Accuracy | حداکثر عمق<br>درخت تصمیم |
|-------------|---------------|------------------|--------------------|----------|--------|-----------|----------|--------------------------|
| ۰.۰۵۴ ثانیه | ۱۷.۹۸         | ۳۶۲.۴۶           | ۱۵.۵۸              | ۷۹.۷%    | ۸۲.۲%  | ۱۲.۲۱%    | ۸۲.۱%    | ۱۰                       |
| ۰.۰۵۶ ثانیه | ۳.۱۱          | ۹.۸۵             | ۱.۰۶               | ۸۳.۱۰%   | ۸۲.۴۹% | ۸۵.۷۳%    | ۸۲.۵۰%   | ۵۰                       |
| ۰.۰۵۷ ثانیه | ۲.۶۹          | ۷.۴۸             | ۰.۹۳               | ۸۴.۸۵%   | ۸۵.۲۷% | ۸۷.۹۴%    | ۸۵.۰۲%   | ۱۰۰                      |
| ۰.۰۵۸ ثانیه | ۲.۶۹          | ۷.۴۷             | ۰.۰۹۳              | ۸۴.۸۴%   | ۸۵.۰۱% | ۸۸.۸۰%    | ۸۵.۰۲%   | ۱۷۶                      |

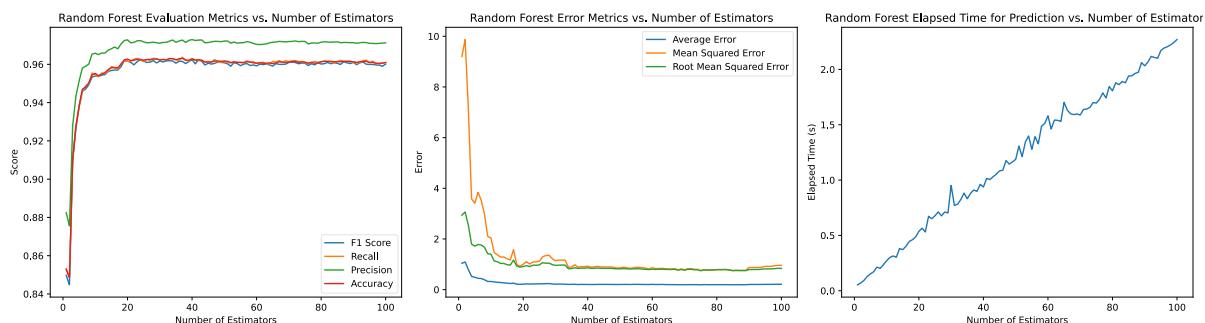
در ادامه با توجه به نتایج حاصل از یادگیرنده پایه، مدل‌های یادگیری گروهی مورد ارزیابی قرار می‌گیرد.

### ۴-۳-۱- مدل موقعیت یاب الگوریتم جنگل تصادفی

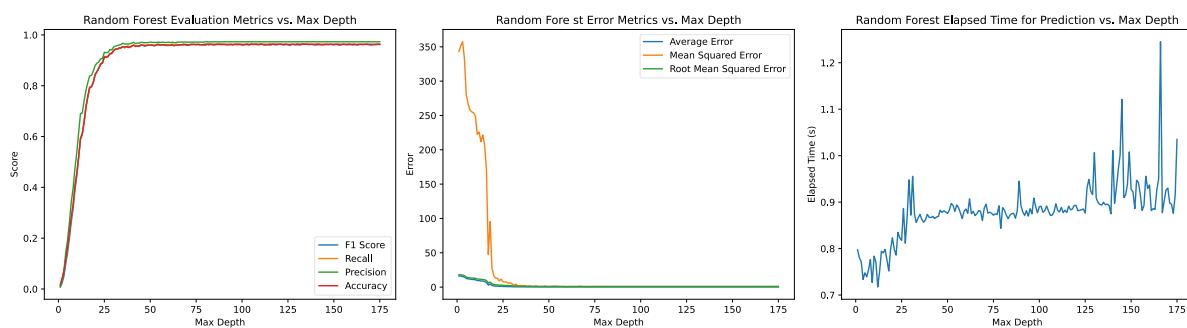
همان‌طور که در ۴-۲-۳ مطرح شد، الگوریتم جنگل تصادفی به سه پارامتر  $m$  (تعداد ویژگی‌های هر گره)،  $J$  (تعداد درختان در جنگل) و اندازه درخت نیاز دارد.  $m$  برابر با نیمی از تعداد ویژگی‌های کل (۱۷۶ ویژگی) درنظر گرفته شده است. اندازه درخت برابر با حداکثر عمق درخت تعیین شده است که در این بخش مقدار آن محدود نشده است. نتایج شبیه‌سازی با تعداد متفاوت درختان ( $J$ ) در شکل (۴-۴) نمایش داده شده است. با توجه به آن، با افزایش تعداد درختان در جنگل، زمان آزمایش افزایش می‌یابد. همچنین مشاهده می‌شود که

<sup>۱</sup> Stratified 5-Fold Cross Validation

خطا و معیارهای سنجش، پس از تعداد مشخصی از درختان، با افزایش آن تغییر چندانی نمی کند. در جدول (۳-۴) نیز مقدار دقیق متغیرها بر حسب سه مقدار از تعداد درختان ذکر شده است.



شکل (۴-۴) نمودار نتیجه ارزیابی، خطای زمان آزمایش جنگل تصادفی برای تعداد متفاوت درختان و حداقل عمق ثابت



شکل (۴-۴) نمودار نتیجه ارزیابی، خطای زمان آزمایش جنگل تصادفی برای عمق متفاوت درختان و تعداد ۴۰ درخت

برای بررسی تأثیر عمق درختان بر معیارهای ارزیابی و زمان، تعداد درختان در مقدار ۴۰ نگه داشته شده است و عمق درختان از صفر تا حداقل مقدار تغییر داده شده است. نتایج ارزیابی در شکل (۴-۴) نمایش داده شده است. مطابق آن پس از یک عمق مشخص، دقت و خطای تغییر چندانی ندارد. دلیل این مسئله برآورده شدن شرایط توقف در ساخت درخت پس از تعداد معینی از گرهها است. این موضوع در نمودار زمان آزمایش بر حسب حداقل عمق و جدول (۴-۴) نیز تأیید می شود.

جدول (۳-۴) نتایج شبیه سازی جنگل تصادفی و تعداد متفاوت درختان و عمق ثابت درختان (حداقل)

| تعداد درختان | Accuracy | Precision | Recall | F1-Score | $E^{avg}$ (متر) | MSE (مترمربع) | RMSE (متر) | زمان آزمایش |
|--------------|----------|-----------|--------|----------|-----------------|---------------|------------|-------------|
| ۱۰           | ۸۵.۳۸٪   | ۸۸.۲۶٪    | ۸۵.۶۶٪ | ۸۹.۱۷٪   | ۰.۹۱            | ۸.۴۳          | ۲.۳۸       | ۰.۰۸۲ ثانیه |
| ۵۰           | ۹۵.۸۰٪   | ۹۷.۱۹٪    | ۹۵.۸۰٪ | ۹۵.۸۴٪   | ۰.۱۹            | ۱.۱۳          | ۰.۹۹       | ۱.۱۷ ثانیه  |
| ۱۰۰          | ۹۵.۷۶٪   | ۹۷.۱۷٪    | ۹۵.۷۶٪ | ۹۵.۸۱٪   | ۰.۶۱            | ۱۵.۲۶         | ۳.۹۰       | ۳.۰۰ ثانیه  |

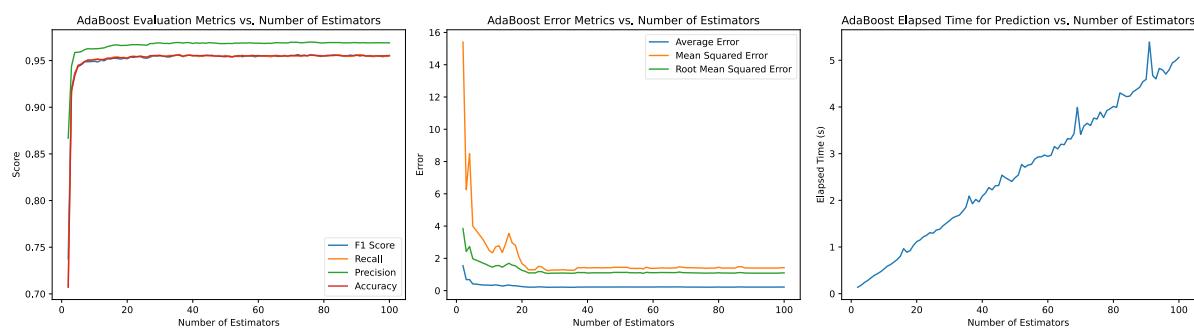
جدول (۴-۴) نتایج شبیه‌سازی جنگل تصادفی برای عمق متفاوت درختان و تعداد ۴۰ درخت

| زمان آزمایش | $RMSE$<br>(متر) | $MSE$<br>(مترمربع) | $E^{avg}$<br>(متر) | $F1-Score$ | $Recall$ | $Precision$ | $Accuracy$ | حداکثر عمق<br>درختان |
|-------------|-----------------|--------------------|--------------------|------------|----------|-------------|------------|----------------------|
| ۰.۸۰ ثانیه  | ۱۱.۵۹           | ۲۲۸.۹۰             | ۹.۵۸               | ۵۰.۷۰٪     | ۶۰.۰۰٪   | ۶۱.۴۶٪      | ۵۰.۹۹٪     | ۱۰                   |
| ۰.۸۶ ثانیه  | ۱.۱۲            | ۱.۴۳               | ۰.۲۱               | ۹۵.۶۹٪     | ۹۵.۶۳٪   | ۹۷.۰۴٪      | ۹۵.۶۸٪     | ۵۰                   |
| ۰.۸۷ ثانیه  | ۱.۱۲            | ۱.۴۳               | ۰.۲۱               | ۹۵.۶۹٪     | ۹۵.۶۳٪   | ۹۷.۰۴٪      | ۹۵.۶۸٪     | ۱۰۰                  |

با درنظر گرفتن دو پارامتر آزاد از نظر محاسباتی، امکان ساخت جنگل تصادفی برای تمام حالات ممکن بسیار گران است. پیدا کردن مقدار بهینه این پارامتر در بخش ۴-۵- با استفاده از الگوریتم ژنتیک توضیح داده می‌شود.

## ۲-۳-۴- مدل موقعیت یاب الگوریتم AdaBoost

در الگوریتم ۲ که برای روش AdaBoost ارائه شده است، تنها از دو پارامتر  $h_t$  (فرضیه یا یادگیرنده پایه) و  $T$  (تعداد یادگیرندگان پایه) به عنوان ابرپارامتر تنظیم مدل یاد شده است. از آن جا که در این پژوهش از درخت تصمیم به عنوان یادگیرنده پایه استفاده شده است. پارامتر  $h_t$ ، به  $m$  (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 ۱۷۶ گره، در نظر گرفته می‌شود و تعداد یادگیرندگان از صفر تا ۱۰۰ تغییر می‌کند. این نتایج در شکل (۶-۴) 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 (۵-۴) 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 ۵۰ و ۱۰۰ درخت تغییری نداشته است.

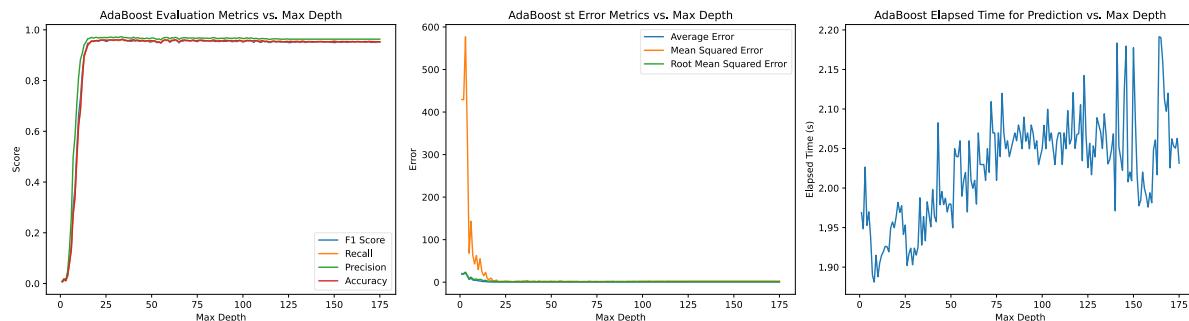


شکل (۶-۴) نمودار نتیجه ارزیابی، خطای زمان آزمایش AdaBoost برای تعداد متفاوت درختان و حداکثر عمق ثابت

جدول (۵-۴) نتایج شبیه سازی AdaBoost با تعداد متفاوت درختان و عمق ثابت درختان (حداکثر)

| زمان آزمایش   | $RMSE$<br>(متر) | $MSE$<br>(مترمربع) | $E^{avg}$<br>(متر) | $F1-Score$ | $Recall$ | $Precision$ | $Accuracy$ | تعداد<br>درختان |
|---------------|-----------------|--------------------|--------------------|------------|----------|-------------|------------|-----------------|
| ۰۰۰ ثانیه ۱۸  | ۲.۴۱            | ۶.۲۵               | ۰.۶۸               | ۹۲.۰۲٪     | ۹۱.۷۷٪   | ۹۴.۳۸٪      | ۹۱.۷۶٪     | ۱۰              |
| ۰۰۲ ثانیه ۲۴۸ | ۱.۱۳            | ۱.۴۴               | ۰.۲۲               | ۹۵.۴۵٪     | ۹۵.۴۶٪   | ۹۶.۸۲٪      | ۹۵.۴۵٪     | ۵۰              |
| ۰۰۵ ثانیه ۵۰۶ | ۱.۰۹            | ۱.۴۱               | ۰.۲۲               | ۹۵.۴۸٪     | ۹۵.۵۰٪   | ۹۶.۸۹٪      | ۹۵.۵۱٪     | ۱۰۰             |

در حالت دوم، تعداد درختان ثابت و برابر با ۴۰ درنظر گرفته شده است و حداکثر عمق درخت تغییر داده می‌شود. نتایج تغییر حداکثر عمق درخت در روش AdaBoost در شکل (۷-۴) 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 عمق تری به مقدار بیشینه خود همگرا شده است. نوسانات موجود در نمودار خطای بیانگر آن است که افزایش دقت مدل با افزایش حداکثر عمق درختان، با خطای موقعیت یابی ارتباط کاملاً مستقیم ندارد؛ چراکه ممکن است علی‌رغم کم شدن تعداد خطای موقعیت‌های دورتری نسبت به نقطه هدف به اشتباه طبقه‌بندی شود و باعث افزایش خطای موقعیت یابی شود.



شکل (۷-۴) نمودار نتیجه ارزیابی، خطای زمان آزمایش AdaBoost برای عمق متفاوت درختان و تعداد ۴۰ درخت

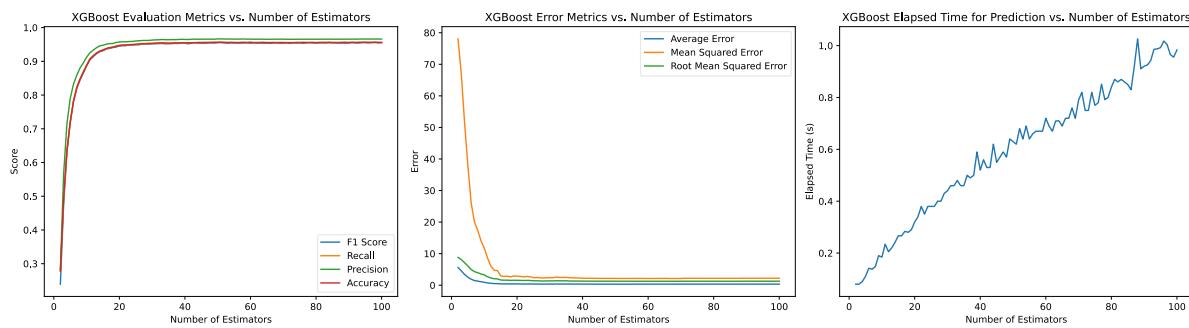
در جدول (۶-۴) نیز مشخص است که الگوریتم AdaBoost نسبت به جنگل تصادفی، مدت زمان بیشتری را برای تخمین دادگان آزمایش محاسبه می‌کند و در این تعداد مشخص از درختان دقت در معیار فاصله اندکی کمتر شده است.

جدول (۶-۴) نتایج شبیه سازی AdaBoost برای عمق متفاوت درختان و تعداد ۴۰ درخت

| زمان آزمایش | $RMSE$<br>(متر) | $MSE$<br>(مترمربع) | $E^{avg}$<br>(متر) | $F1-Score$ | $Recall$ | $Precision$ | $Accuracy$ | حداکثر عمق<br>درختان |
|-------------|-----------------|--------------------|--------------------|------------|----------|-------------|------------|----------------------|
| ۱.۸۸ ثانیه  | ۵.۳۸            | ۲۹.۴۳              | ۳.۲۱               | ۶۶.۲۴٪     | ۶۲.۲۹٪   | ۰.۸۱٪       | ۶۲.۳۱٪     | ۱۰                   |
| ۱.۹۷ ثانیه  | ۱.۲۹            | ۲.۲۸               | ۰.۳۶               | ۹۵.۵۶٪     | ۹۵.۶۳٪   | ۹۶.۵۱٪      | ۹۵.۶۶٪     | ۵۰                   |
| ۲.۰۴ ثانیه  | ۱.۲۴            | ۲.۲۶               | ۰.۳۶               | ۹۵.۳۴٪     | ۹۵.۷۶٪   | ۹۶.۵۶٪      | ۹۵.۵۲٪     | ۱۰۰                  |

### ۴-۳-۳- مدل موقعیت یاب الگوریتم

برخلاف دو الگوریتم قبلی، XGBoost به ابرپارامترهای بیشتری برای تنظیم نیاز دارد.  $\eta$  (نرخ یادگیری) به عنوان ضریبی برای کاهش اندازه گام، برابر با مقدار  $0.3$  درنظر گرفته شده است تا از بیش برازش جلوگیری کند.  $\gamma$  (ضریب کاهش اولیه) برابر با صفر و  $\lambda$  (ضریب تنظیم برگ) نیز برابر با یک تنظیم شده است. این مقادیر، مقادیر پیش فرضی است که در بسیاری از پژوهش های مشابه استفاده شده است و نتایج مطلوبی از آن حاصل شده است [۳۶].  $K$  (تعداد تکرارهای Boosting) و همچنان حداکثر عمق درخت به عنوان ابر پارامتر آزاد برای تنظیم در نظر گرفته می شود و نتایج ارزیابی بررسی می گردد.



شکل (۴-۴) نمودار نتیجه ارزیابی، خطأ و زمان آزمایش XGBoost برای تعداد متفاوت درختان و حداکثر عمق ثابت

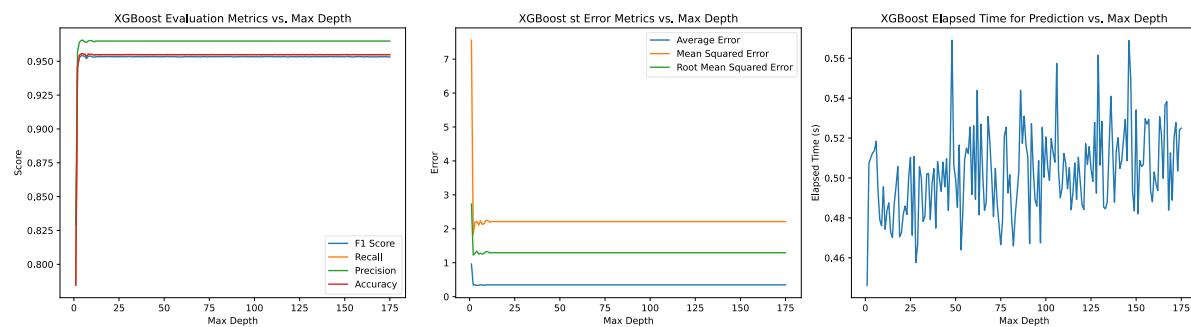
همان طور که در شکل (۴-۴) مشاهده می شود، در این روش نیز زمان آزمایش با تعداد یادگیرندگان رابطه خطی دارد. همچنین بیشینه مقدار دقت، با تعداد بیشتری از درختان نسبت به روش AdaBoost همگرا می شود؛ اما مطابق با جدول (۷-۴) زمان آزمایش روش XGBoost نسبت به روش AdaBoost بسیار کمتر است.

جدول (۷-۴) نتایج شبیه سازی XGBoost با تعداد متفاوت درختان و عمق ثابت درختان (حداکثر)

| تعداد درختان | Accuracy | Precision | Recall | F1-Score | $E^{avg}$ (متر) | MSE (مترمربع) | RMSE (متر) | زمان آزمایش (ثانیه) |
|--------------|----------|-----------|--------|----------|-----------------|---------------|------------|---------------------|
| ۱۰           | ۸۸.۸۰٪   | ۹۱.۲۱٪    | ۸۸.۳۴٪ | ۸۸.۵۸٪   | ۰.۹۲            | ۱۱.۷۴         | ۳.۲۸       | ۰.۱۹                |
| ۵۰           | ۹۵.۶۴٪   | ۹۶.۶۴٪    | ۹۵.۴۶٪ | ۹۵.۵۰٪   | ۰.۳۳            | ۲.۱۶          | ۱.۲۷       | ۰.۶۳                |
| ۱۰۰          | ۹۵.۶۲٪   | ۹۶.۶۰٪    | ۹۵.۶۲٪ | ۹۵.۴۸٪   | ۰.۳۳            | ۲.۲۱          | ۱.۲۷       | ۰.۹۸                |

با ثابت قراردادن تعداد درختان در مقدار  $40$  و تغییر حداکثر عمق درختان، نتایج حاصل از آن در شکل (۹-۴) 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 می شود. همچنین زمان آزمایش، اثر کمتری از عمق

درختان نسبت به تعداد درختان می‌پذیرد.



شکل (۹-۴) نمودار نتیجه ارزیابی، خطأ و زمان آزمایش XGBoost برای عمق مختلف درختان و تعداد ۴۰ درخت

نتایج مندرج در جدول (۸-۴) نشان می‌دهد که الگوریتم XGBoost در عمق کمتر و با زمان کمتری نسبت به دو روش قبلی به دقت مشابه رسیده است. این همگرایی سریع‌تر، به علت درنظرگیری گرادیان و هسین در ساخت درختان تصمیم است.

جدول (۸-۴) نتایج شبیه‌سازی XGBoost برای عمق مختلف درختان و تعداد ۴۰ درخت

| زمان آزمایش | RMSE (متر) | MSE (مترمربع) | $E^{avg}$ (متر) | F1-Score | Recall | Precision | Accuracy | حداکثر عمق درختان |
|-------------|------------|---------------|-----------------|----------|--------|-----------|----------|-------------------|
| ۰.۵۳ ثانیه  | ۱.۳۲       | ۲.۲۴          | ۰.۳۴            | ۹۵.۳۶٪   | ۹۵.۴۹٪ | ۹۶.۵۳٪    | ۹۵.۵۰٪   | ۱۰                |
| ۰.۵۲ ثانیه  | ۱.۲۹       | ۲.۲۱          | ۰.۳۴            | ۹۵.۳۳٪   | ۹۵.۴۸٪ | ۹۶.۴۹٪    | ۹۵.۴۸٪   | ۵۰                |
| ۰.۵۳ ثانیه  | ۱.۲۴       | ۲.۲۶          | ۰.۳۴            | ۹۵.۳۳٪   | ۹۵.۴۸٪ | ۹۶.۴۹٪    | ۹۵.۵۲٪   | ۱۰۰               |

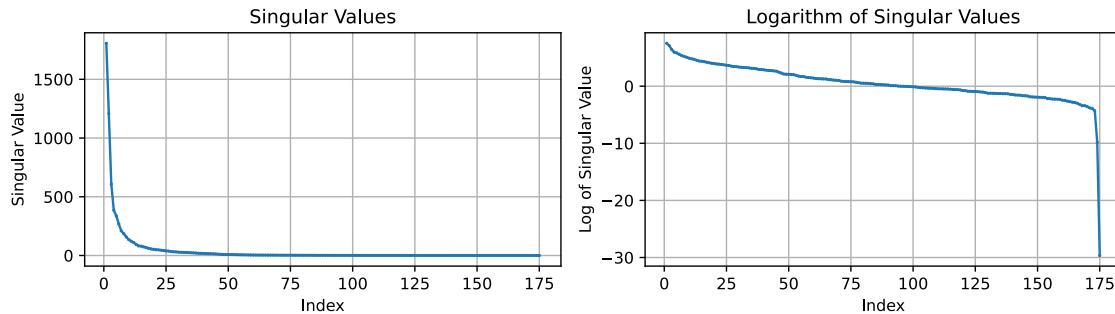
در نتایج حاصل از پیاده‌سازی روش‌های یادگیری گروهی به کاررفته در این پژوهش، مشاهده می‌شود که چه میزان تنظیم پارامترهای مدل در دقت و زمان اثرگذار است. بدین منظور از دو روش کاهش ابعاد و الگوریتم ژنتیک برای یافتن مدل با دقت و زمان اجرای مطلوب و قابلیت پیاده‌سازی استفاده می‌شود. در ادامه نتایج استفاده از این دو الگوریتم بررسی می‌شود.

#### ۴-۴- کاهش ابعاد دادگان موقعیت یابی

باتوجه به تئوری بیان شده در ۳-۳- کاهش ابعاد بردار ویژگی‌ها، به طور غیرمستقیم در روش‌های یادگیری گروهی با کاهش حداکثر عمق درختان اثرگذار است. از این‌رو، نتایج روش معرفی شده برای کاهش ابعاد در این بخش بررسی می‌شود.

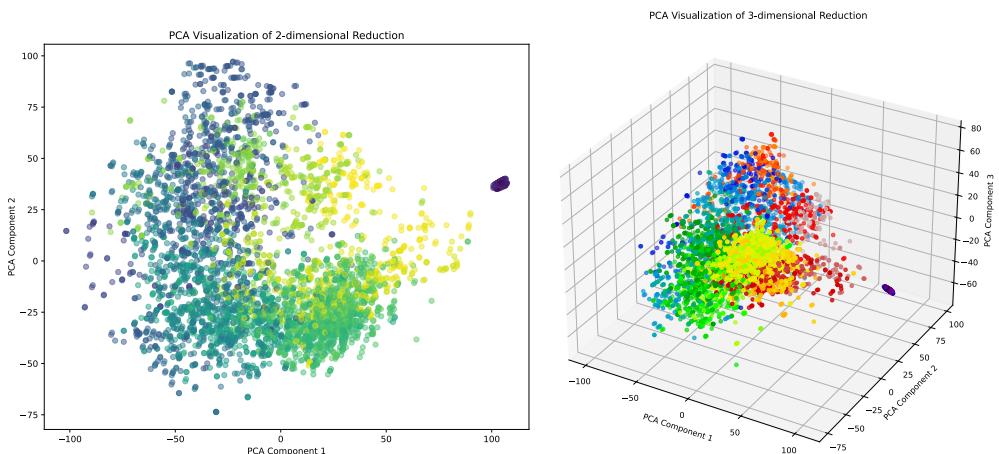
در روش تجزیه مؤلفه‌های اساسی (PCA) پس از استانداردسازی و محاسبه کوواریانس، با استفاده از تجزیه

مقادیر منفرد (SVD) قطر اصلی ماتریس  $\Lambda$  به ترتیب بزرگ به کوچک مطابق با نمودار شکل (۱۰-۴) محاسبه شده است. بر اساس آن، تغییرات وسیع قبل از مرتبه ۱۵ است، لذا می‌توان در محدوده این عدد، مرتبه کاهش را انتخاب نمود.



شکل (۱۰-۴) نمودار مقادیر منفرد ماتریس کوواریانس ویژگی ها

همچنین اشاره شد که روش PCA در تجسم داده می‌تواند مؤثر باشد. به همین منظور، همان‌طور که در شکل (۱۱-۴) مشخص است، نقاط همسایه (دارای رنگ مشابه) در فضای دو و سه بعدی از ویژگی ها 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.



شکل (۱۱-۴) تجسم داده در فضای دو و سه بعدی با استفاده از روش کاهش ابعاد PCA

باتوجه به نتایج بخش قبلی یادگیری گروهی و امکان مقایسه با آن مرتبه کاهش ۱۰ انتخاب شده است. جدول (۹-۴) نتایج به کارگیری روش کاهش ابعاد را در الگوریتم های یادگیری گروهی نشان می‌دهد. با مقایسه این جدول با جداول نتایج حداقل 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 موقعیت یابی را کاهش داده است که نشان‌دهنده ناکارمدی کاهش ابعاد در الگوریتم XGBoost است.

جدول ۹-۴) نتایج روش PCA با کاهش مرتبه ۱۰ در روش‌های یادگیری گروهی با ۴۰ درخت

| الگوریتم<br>یادگیری | Accuracy | Precision | Recall | F1-Score | $E^{avg}$<br>(متر) | MSE<br>(مترمربع) | RMSE<br>(متر) | زمان آزمایش |
|---------------------|----------|-----------|--------|----------|--------------------|------------------|---------------|-------------|
| جنگل<br>تصادفی      | ۹۴.۶۸٪   | ۹۵.۴۹٪    | ۹۴.۶۷٪ | ۹۴.۸۳٪   | ۰.۶۷               | ۱.۱۰             | ۱.۰۵          | ۰.۸۱ ثانیه  |
| AdaBoost            | ۹۱.۳۸٪   | ۹۲.۵۴٪    | ۹۱.۳۷٪ | ۹۱.۲۷٪   | ۱.۴۴               | ۱۳.۲۳            | ۳.۰۱          | ۱.۹ ثانیه   |
| XGBoost             | ۸۸.۳۰٪   | ۸۹.۴۳٪    | ۸۸.۲۹٪ | ۸۸.۰۹٪   | ۱.۶۲               | ۱۳.۶۲            | ۳.۱۲          | ۰.۷۲ ثانیه  |

روش PCA گرچه با کاهش عمق درخت باعث بهبود نسبی در دقت و پیاده‌سازی برخی از الگوریتم‌های یادگیری می‌شود؛ 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 ابر پارامتری که تنظیم آن ضروری است به مدل اضافه می‌کند.

## ۴-۵- 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 اثر انگشت مطرح است [۳۷، ۳۸] و به طور غیرمستقیم باعث بهبود نسبی دقت و سادگی در پیاده‌سازی 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 درنظر گیری 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 رتبه‌بندی 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 درنظر گیری دقت طبقه‌بندی، حساسیت مدل را نیز در نظر می‌گیرد و مدل یادگیرنده را مقاوم می‌کند. در بخش ۳-۴ مشاهده شد که زمان آزمایش با تعداد یادگیرنده‌گان رابطه مستقیم دارد. از این رو با درنظر گیری زمان آزمایش در تابع برازش، مدل به سمت کاهش تعداد یادگیرنده‌گان 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 می‌شود:

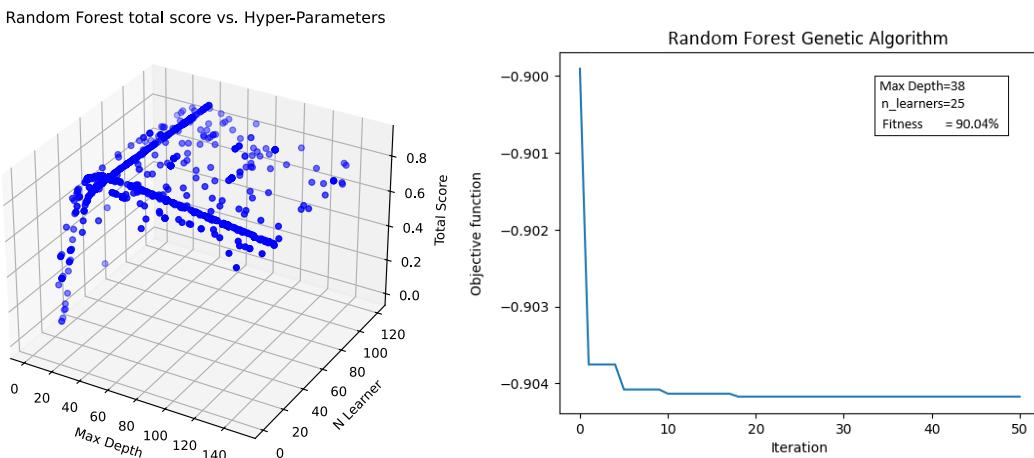
$$fitness(x) = F1-Score - \alpha T_{test} - \beta D_M \quad (1-4)$$

که در آن  $T_{test}$  زمان آزمایش برای دادگان آزمایش،  $D_M$  حداکثر عمق درخت و  $\alpha$  و  $\beta$  ضرایب تنظیم تابع برازش است. در رابطه (1-4)  $F1-Score$  در بازه  $[0,1]$  قرار دارد. بنابراین مقادیر  $\alpha$  و  $\beta$  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 (۱۰-۴) ذکر شده است.

جدول (۱۰-۴) مقادیر ضرایب خطی تابع برازش الگوریتم ژنتیک

| $\beta$            | $\alpha$           | الگوریتم یادگیری |
|--------------------|--------------------|------------------|
| $1 \times 10^{-3}$ | $2 \times 10^{-2}$ | جنگل تصادفی      |
| $1 \times 10^{-3}$ | $1 \times 10^{-2}$ | AdaBoost         |
| $1 \times 10^{-3}$ | $1 \times 10^{-2}$ | XGBoost          |

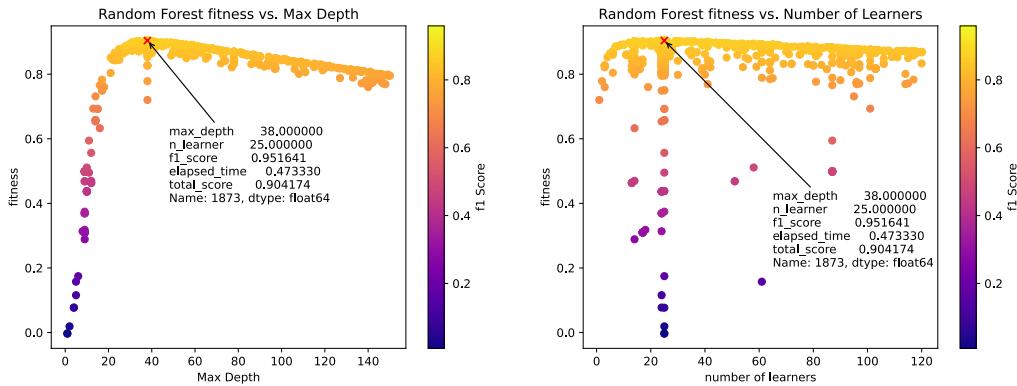
در الگوریتم ژنتیک، حداکثر عمق درختان و تعداد یادگیرندگان به عنوان ابر پارامتر تنظیم معرفی می‌شود. اندازه جمعیت در این الگوریتم برابر با ۲۰، احتمال جهش برابر  $0.1$ ، تعداد تکرار الگوریتم ژنتیک برابر با ۵۰ در نظر گرفته شده است. همچنین برای انتخاب تصادفی جمعیت از تابع توزیع یکنواخت استفاده شده است. با اجرای الگوریتم ژنتیک در جنگل تصادفی، مقدار تابع هدف (منفی مقدار تابع برازش) در تکرارهای الگوریتم مطابق با نمودار راست شکل (۱۲-۴) پس از اتمام ۵۰ تکرار در الگوریتم ژنتیک، جنگل تصادفی شامل ۲۵ درخت با حداکثر عمق ۳۸ برای رسیدن به مقدار بهینه  $90.04\%$  تابع برازش مناسب دیده شده است.



شکل (۱۲-۴) نمودار تابع هدف بر حسب تکرار (راست) و نمودار مقدار تابع برازش بر حسب ابر پارامتر بهایزی هر نمونه از جمعیت (چپ) در جنگل تصادفی

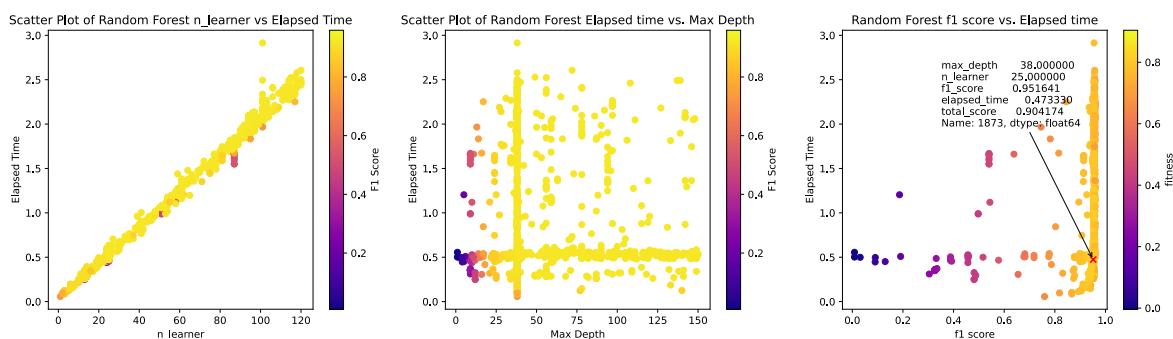
در نمودار سمت چپ شکل (۱۲-۴) نیز مقدار تابع برازش به ازای هر نمونه بر حسب حداکثر عمق درختان و تعداد یادگیرندگان نشان داده شده است. با ترسیم دو بعدی تابع برازش بر حسب مقادیر پارامتر آزاد شکل

(۱۳-۴) 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  $\times$  مشخص شده است، دارای بیشترین مقدار تابع برازش است و همچنین مقدار معیار دقت آن نیز مطلوب است. با توجه به شکل (۱۶-۴) و شکل (۱۹-۴) این موضوع در روش‌های AdaBoost و XGBoost نیز صادق است.



شکل (۱۳-۴) معیار دقت و مقدار تابع برازش الگوریتم ژنتیک بر حسب ابرپارامترهای جنگل تصادفی

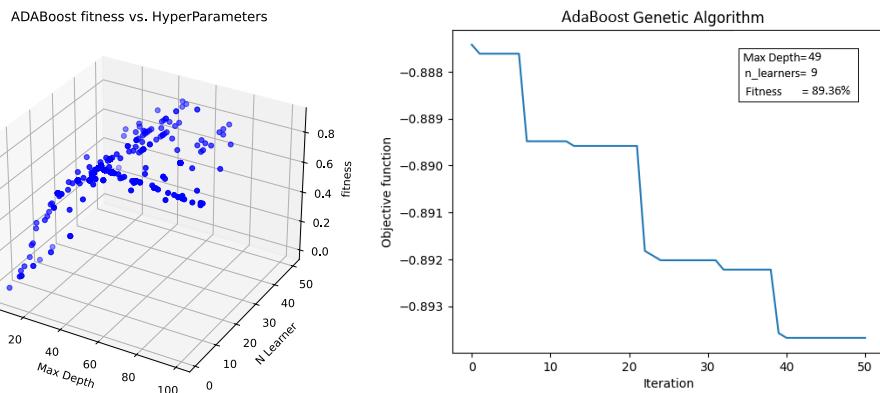
نمودارهای زمان آزمایش بر حسب معیار دقت و ابرپارامترهای جنگل تصادفی در شکل (۱۴-۴) نمایش داده شده است. نمودار زمان آزمایش بر حسب تعداد یادگیرندگان نشان می‌دهد که فارغ از عمق درختان، افزایش تعداد یادگیرندگان تأثیر مستقیم بر زمان آزمایش دارد. به علاوه، امکان دستیابی به دقتی در حدود دقت مطلوب در اکثر مقادیر مختلف تعداد یادگیرندگان وجود دارد. نمودار زمان آزمایش بر حسب حداکثر عمق درخت نشان دهنده عدم ارتباط معنادار میان این دو را نشان می‌دهد و همچنین نمایانگر وجود یک مقدار از حداکثر عمق است که پس از آن تغییرات F1 Score تقریباً ثابت باقی می‌ماند. نمودار زمان آزمایش بر حسب F1 Score مؤید انتخاب مناسب ضرایب  $\alpha$  و  $\beta$  در تابع برازش رابطه (۱-۴) است. این تحلیل برای نتایج روش‌های AdaBoost و XGBoost که در شکل (۱۷-۴) و شکل (۲۰-۴) نشان داده شده نیز برقرار است.



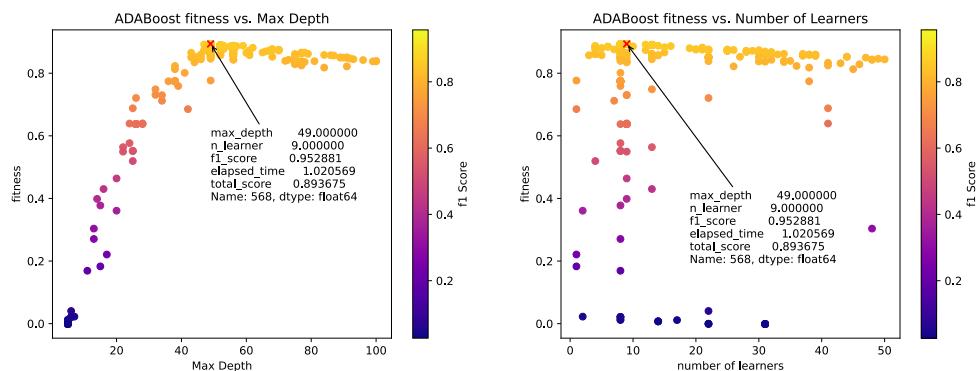
شکل (۱۴-۴) زمان آزمایش بر حسب ابرپارامترهای جنگل تصادفی و تابع برازش

مشابه جنگل تصادفی، در الگوریتم ژنتیک با همان پارامترهای اولیه اجرا شده است. تابع هدف به

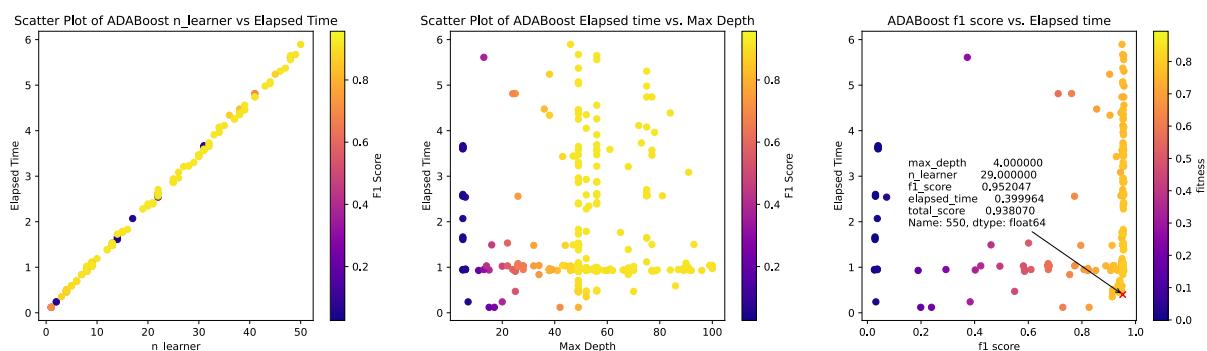
ازای هر تکرار از الگوریتم ژنتیک و نمودار مقدار تابع برازش بر حسب ابرپارامترها در شکل (۱۵-۴) ارائه شده است. مشاهده می‌شود که تابع برازش به مقدار بهینه خود همگرا شده است. همچنین وجود نقطه‌ای بیشینه در نمودار سه بعدی مشهود است. نتیجه نهایی الگوریتم ژنتیک، مدل AdaBoost با حداقل عمق ۴۹ و با ۹ یادگیرنده را مدل مناسب معرفی می‌کند.



شکل (۱۵-۴) نمودار تابع هدف بر حسب تکرار (راست) و نمودار مقدار تابع برازش بر حسب ابرپارامتر بهایزی هر نمونه از جمعیت (چپ) در AdaBoost

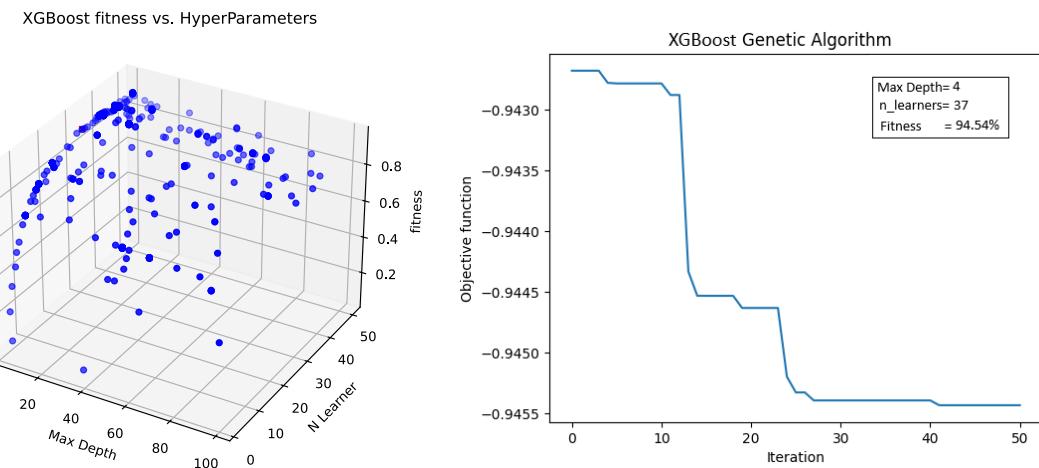


شکل (۱۶-۴) معیار دقت و مقدار تابع برازش الگوریتم ژنتیک بر حسب ابرپارامترهای AdaBoost

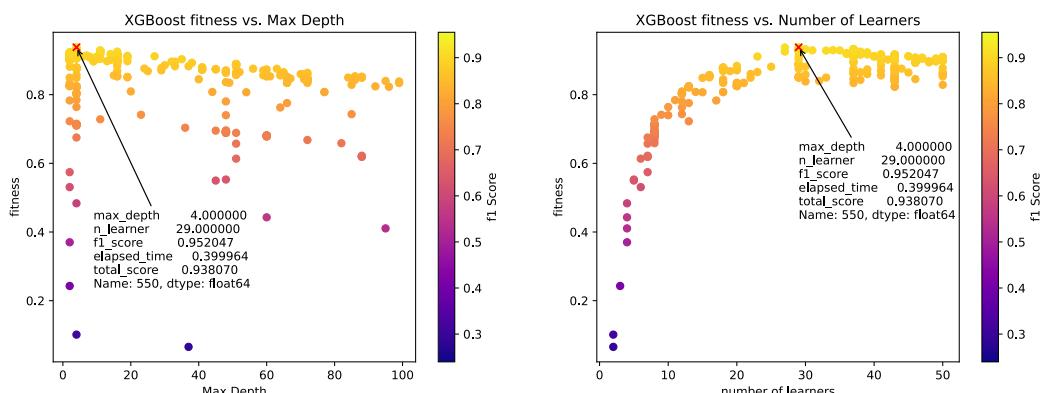


شکل (۱۷-۴) زمان آزمایش بر حسب ابرپارامترهای AdaBoost و تابع برازش

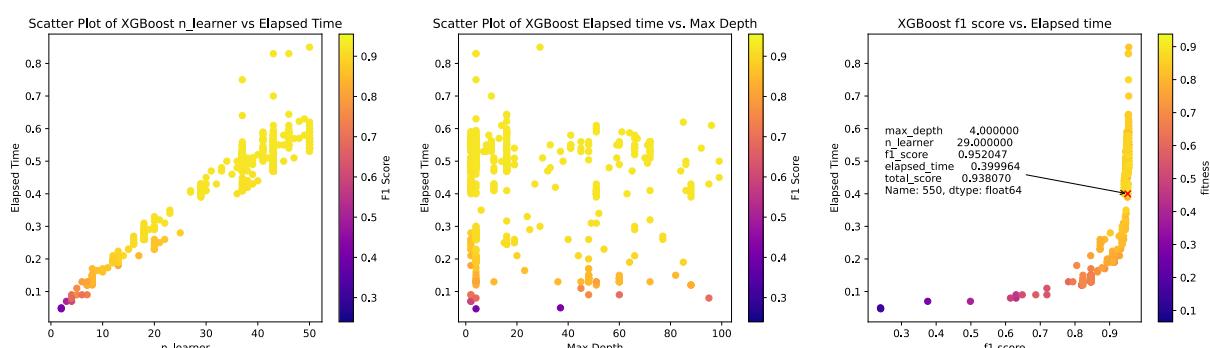
پس از اجرای الگوریتم ژنتیک برای تنظیم مدل XGBoost با همان مقادیر اولیه دو اجرای قبلی، شکل (۱۸-۴) برای مقدار تابع هدف و برازش حاصل گردیده است. همگرایی تابع برازش به مقدار ۹۴.۵۴٪ مشهود است. همچنین، مقدار ۴ به عنوان حداقل عمق درختان و ۳۷ به عنوان تعداد یادگیرندگان، مقدار معروفی شده الگوریتم ژنتیک برای بیشینه سازی تابع برازش است.



شکل (۱۸-۴) نمودار تابع هدف بر حسب تکرار (راست) و نمودار مقدار تابع برازش بر حسب ابر پارامتر به ازای هر نمونه از XGBoost جمعیت (چپ) در



شکل (۱۹-۴) معیار دقت و مقدار تابع برازش الگوریتم ژنتیک بر حسب ابر پارامترهای XGBoost



شکل (۲۰-۴) زمان آزمایش بر حسب ابر پارامترهای XGBoost و تابع برازش

## ۴-۶- مقایسه و جمع‌بندی نتایج

در این فصل مدل‌های الگوریتم یادگیری گروهی به منظور استفاده در موقعیت‌یابی با روش اثر انگشت آموزش داده شده‌اند. ارزیابی این الگوریتم‌ها نشان داد که امکان به کارگیری این الگوریتم‌ها در گروی خروجی دقیق، بلادرنگ و به کارگیری حداقل منابع محاسباتی است. از این‌رو، دو روش پیشنهادی کاهش ابعاد با روش PCA و بهره‌گیری از الگوریتم‌های پردازش تکاملی مانند الگوریتم ژنتیک در تنظیم ابرپارامترهای مدل برای رسیدن به مدل بهینه ارائه شد. روش کاهش ابعاد PCA، تنها با کاهش ابعاد بردار ورودی، سعی در کاهش حداقل عمق درختان و درنتیجه کاهش منابع محاسباتی با حفظ دقت را داشت. مشاهده شد که به علت تأثیرات ناچیز حداقل عمق درختان بر زمان آزمایش، این روش پیشنهادی در سرعت اجرای مدل تقریباً بی‌اثر است. همچنین، این روش به علت تغییر فضای ویژگی‌های ورودی، تفسیرپذیری مدل را کاهش داده و علاوه‌بر آن، مجموعه‌ای از دستورات اجرایی را به پردازش سیستم موقعیت‌یاب اضافه می‌کند. در روش پیشنهادی دوم، الگوریتم ژنتیک برای تنظیم ابرپارامترهای مدل به کار گرفته شد. در این روش با درنظرگیری یکتابع برآش سعی در پیداکردن مدلی با حداقل تعداد یادگیرنده و زمان آزمایش با حفظ تقریبی دقت دارد. با اجرای این الگوریتم مقادیر بهینه ابرپارامترها در ناحیه جستجو مشخص شد.

جدول (۱۱-۴) نتایج اجرای مدل گروهی با مقادیر بهینه در ناحیه جستجو

| الگوریتم یادگیری | Accuracy | Precision | Recall | F1-Score | $E^{avg}$ (متر) | MSE (مترمربع) | RMSE (متر) | زمان آزمایش |
|------------------|----------|-----------|--------|----------|-----------------|---------------|------------|-------------|
| جنگل تصادفی      | ۹۵.۲۳٪   | ۹۶.۵۲٪    | ۹۵.۲۲٪ | ۹۵.۱۶٪   | ۰.۲۷            | ۲.۱۷          | ۱.۳۴       | ۰.۵۷ ثانیه  |
| AdaBoost         | ۹۵.۳۱٪   | ۹۶.۲۷٪    | ۹۵.۳۰٪ | ۹۵.۱۵٪   | ۰.۳۸            | ۲.۳۲          | ۱.۴۵       | ۰.۴۹ ثانیه  |
| XGBoost          | ۹۵.۳۹٪   | ۹۶.۲۶٪    | ۹۵.۳۸٪ | ۹۵.۲۰٪   | ۰.۳۴            | ۲.۱۹          | ۱.۳۵       | ۰.۵۰ ثانیه  |

جدول (۱۱-۴) نتایج استفاده از این مقادیر بهینه را در مدل‌های یادگیری گروهی نشان می‌دهد. در مدل‌های تنظیم شده این جدول جنگل تصادفی از دقت بالاتری نسبت به مدل‌های AdaBoost و XGBoost برخوردار است اما دارای زمان آزمایش بیشتری نسبت به دو مدل قبلی است. همچنین مطابق با نتایج بخش ۴-۵، الگوریتم XGBoost، به علت استفاده از عمق کمتر درختان نسبت به دو روش گروهی دیگر، از منابع سخت افزاری کمتری برای ذخیره‌سازی درختان استفاده می‌کند.

برای مقایسه نتایج به دست آمده از این پژوهش با سایر کارها، دو مقاله با مجموعه‌دادگان Yicksan و استفاده از روش‌های دیگر، به کار گرفته شده است. در کار [۳۵] که اولین کار مرتبط با مجموعه‌دادگان JUIndoorLoc است و این مجموعه‌دادگان به تفصیل معرفی شده است. از روش ترکیب متخصصان با به کارگیری مدل‌های

SVM و Bayse Net استفاده شده است و در نهایت مقایسه‌ای بین رأی‌گیری اکثریت و رأی‌گیری وزن‌دار مدل kNN انجام شده است. پژوهش [۱۶]، یک روش جدید برای محاسبه وزن هر مدل با به کارگیری تئوری Dempster-Shafer belief ارائه کرده است و بر اساس آن روابطی از عدم قطعیت، اعتقاد و عدم اعتقاد<sup>۱</sup> مدل را بیان و آن را در محاسبه وزن هر مدل استفاده کرده است. در این پژوهش از مدل‌های kNN، K\*، SVM و Bayse Net به عنوان یادگیرنده پایه استفاده شده است.

در هر دو پژوهش تنها از دادگان طبقه چهارم برای آزمایش الگوریتم گروهی استفاده شده است. از این رو برای مقایسه، مدل پیشنهادی این پژوهش نیز بر روی دادگان آزمایش طبقه چهارم (۱۴۶۰ نمونه) اجرا می‌شود. نتیجه اجرا در جدول (۱۲-۴) ذکر شده است. جنگل تصادفی با دقت بالا، توانسته است موقعیت را تخمین بزند.

جدول (۱۲-۴) مقایسه نتایج اجرای مدل گروهی تنظیم شده با روش‌های ارائه شده در [۱۶] و [۳۵] و به کارگیری دادگان آزمایش طبقه چهارم<sup>۲</sup>

| الگوریتم یادگیری | Accuracy | $E^{avg}$ (متر) | زمان آزمایش   |
|------------------|----------|-----------------|---------------|
| جنگل تصادفی      | ۹۸.۳۰٪   | ۰.۲۵            | ۰.۱۷ ثانیه    |
| AdaBoost         | ۹۵.۰۴٪   | ۰.۳۴            | ۰.۳۳ ثانیه    |
| XGBoost          | ۹۷.۵۳٪   | ۰.۳۱            | ۰.۱۵ ثانیه    |
| پژوهش [۳۵]       | ۹۱.۷۴٪   | ۱.۸۷            | (۰.۲۸ ثانیه)  |
| پژوهش [۱۶]       | ۹۴.۸۷٪   | ۱.۴۱            | (۰.۵۰۶ ثانیه) |

در مقایسه با پژوهش‌های دیگر در [۱۶، ۳۵]، دقت بالاتر روش ارائه شده در این پایان‌نامه تأیید می‌گردد. نسبت به روش پژوهش [۱۶]، به علت درنظرگیری معیار F1-Score در تنظیم مدل‌های یادگیری گروهی، خطای  $E^{avg}$  به طور قابل توجهی کاهش یافته است. مدل‌های ارائه شده در این پایان‌نامه با سرعت بسیار بالاتری تخمین را انجام داده‌اند. به این علت که روش پژوهش [۱۶]، روشی داده-محور است و در مرحله آزمایش، نمونه‌ها با نمونه‌های آزمایش مقایسه می‌شوند. همچنین با همین استدلال، در [۱۶] نیازمند ذخیره مجموعه دادگان در پایگاه داده است و درنتیجه به منابع ذخیره‌سازی بیشتری نیاز است.

<sup>۱</sup> Belief and disbelief

<sup>۲</sup> مقادیر درج شده در داخل پرانتز، مقادیری است که در داخل مقالات ذکر نشده است بلکه با پیاده‌سازی مجدد الگوریتم و اجرا در سخت افزار یکسان این پژوهش حاصل شده است.

## فصل ۵:

# جمع‌بندی و پیشنهاد‌ها

## ۱-۵ جمع‌بندی

در فصل مقدمه به بررسی کارهای انجام شده، مسئله مطرح، اهداف و دستاوردهای پژوهش و مروری از فصل‌های پایان‌نامه پرداخته شده است.

در فصل دوم، پس از تعریف موقعیت‌یابی داخلی، به بررسی دادگان و الگوریتم‌های مورداستفاده در این زمینه پرداخته شد. نقش فناوری‌های بی‌سیم، بهویژه قدرت سیگنال دریافتی در موقعیت‌یابی مورد بحث قرار گرفت. همچنین چالش‌های پیش روی این تکنیک، مانند چندمسیره شدن، خط غیر دید، تداخل و پویایی محیط بررسی شد. در ادامه، به ساختار موقعیت‌یابی با یادگیری ماشین پرداخته شد و مراحل جمع‌آوری، پیش‌پردازش و استخراج ویژگی‌های داده‌ها برای آموزش مدل‌های یادگیری ماشین شرح داده شد. سپس، الگوریتم‌های یادگیری تحت نظارت، بدون نظارت و نیمه نظارتی در موقعیت‌یابی مبتنی بر RSS معرفی و چالش‌های این روش بررسی شده است.

در فصل سوم به بررسی راهکارهای افزایش دقت موقعیت‌یابی با استفاده از روش‌های یادگیری گروهی پرداخته شد. مروری بر الگوریتم‌های یادگیری گروهی، اجزای آن و روش‌های Bagging و Boosting ارائه و الگوریتم‌های جنگل تصادفی، AdaBoost و XGBoost به عنوان محبوب‌ترین الگوریتم‌های یادگیری گروهی معرفی شد. با توجه به نیاز بالای الگوریتم‌های یادگیری گروهی به منابع محاسباتی، دو راهکار کاهش ابعاد و به کارگیری الگوریتم ژنتیک ارائه شدند. تجزیه مؤلفه‌های اساسی (PCA) به عنوان روش‌های پیش‌پردازش و الگوریتم ژنتیک برای تنظیم ابرپارامترهای یادگیری گروهی معرفی شده است. در پایان این فصل، معیارهای ارزیابی عملکرد بررسی و معیارهای کیفی عملکرد معرفی شد.

در فصل چهارم، مدل‌های الگوریتم یادگیری گروهی برای موقعیت‌یابی با روش اثر انگشت آموزش داده شده است. ارزیابی این الگوریتم‌ها نشان داد که استفاده از آن‌ها به خروجی دقیق، بلادرنگ و حداقل منابع محاسباتی نیاز دارد. دو روش پیشنهادی برای کاهش ابعاد با روش PCA و استفاده از الگوریتم‌های پردازش تکاملی مانند الگوریتم ژنتیک برای تنظیم ابرپارامترهای مدل ارائه شد. در این فصل، نتایج حاصل از شبیه‌سازی مدل‌ها ارائه و مقایسه شد. مشاهده شد جنگل تصادفی از دقت بالاتری نسبت به AdaBoost و XGBoost برخوردار است، اما زمان آزمایش بیشتری دارد. XGBoost به دلیل استفاده از عمق کمتر درختان، از منابع سخت‌افزاری کمتری استفاده می‌کند. برای مقایسه نتایج با سایر تحقیقات، دو مقاله با مجموعه دادگان مشابه بررسی شده است. در هر دو مقاله، فقط از داده‌های طبقه چهارم برای آزمایش الگوریتم گروهی استفاده شده است. روش F1-Score ارائه شده در این پایان‌نامه دقت بالاتری نسبت به روش‌های دیگر دارد. به دلیل درنظرگرفتن معیار

در تنظیم مدل‌های یادگیری گروهی، خطای موقعیت یابی به طور قابل توجهی کاهش یافته است. مدل‌های ارائه شده در این پایان‌نامه با سرعت بسیار بالاتری تخمین را انجام می‌دهند.

## ۲-۵- پیشنهادها

در این پایان‌نامه، به تشخیص شرایط محیطی به عنوان زمینه پژوهش پرداخته نشده است. چراکه از شرایط فیزیکی و محیطی در ویژگی‌های موجود در مجموعه دادگان استفاده شده است و فرض بر دسترس بودن آن وجود دارد؛ بنابراین، می‌توان به طراحی یک سیستم موقعیت یابی دو مرحله‌ای پرداخت؛ به این ترتیب که در مرحله اول، ابتدا شرایط فیزیکی و محیطی را تخمین بزند؛ سپس در مرحله دوم، از خروجی تخمین زده شده در مرحله اول برای تخمین موقعیت استفاده نماید. با این اصلاحات، می‌توان به روند پژوهشی اضافه‌ای داد که به بررسی تأثیر شرایط محیطی بر دقت موقعیت یابی پرداخته و دقت را افزایش دهد.

می‌توان از تکنیک ارائه شده در سایر داده‌های موقعیت یابی مانند CSI نیز استفاده کرد تا قابلیت تعمیم این تکنیک به سایر دادگان ارتقا یابد. همچنین، امکان استفاده از این روش در مسائل رگرسیونی نیز وجود دارد و می‌توان نتایج را موردمطالعه قرار داد. برای ارزیابی کارایی و عملکرد مدل‌ها، می‌توان آزمایشاتی را در محیط‌های واقعی اجرا کرد و نتایج را با استفاده از معیارهای دقت و حساسیت مورد بررسی قرار داد.

در زمینه ردیابی در مکان سرپوشیده، با گنجاندن داده‌های حسگرهای متعدد مانند شتاب‌سنج، ژیروسکوپ با موقعیت یابی مبتنی بر RSS، می‌تواند مجموعه‌ای از مدل‌ها با اطلاعات متنوع را تشکیل دهد که به طور چشمگیری دقت ردیابی را افزایش دهند. در مسئله ردیابی ضروری است که با دقت به بررسی همبستگی بین داده‌ها، اطمینان حاصل شود که مجموعه به دست آمده به طور مؤثر از اطلاعات هر حسگر استفاده کند و همچنین از بیش‌برازش جلوگیری شود. تکنیک‌هایی مانند انتخاب ویژگی، روش‌های ضد همبستگی یا وزن‌دهی مدل بر اساس اهمیت داده‌های هر حسگر می‌تواند برای حل این چالش‌ها در سیستم‌های ردیابی داخلی مورداستفاده قرار گیرند.

بررسی و استفاده از روش‌های تطبیقی برای تنظیم خودکار ابرپارامترهای مدل در شرایط مختلف، یکی از موارد مهم در بهبود کارایی و عملکرد سیستم‌های ردیابی داخلی است. این روش‌ها امکان تغییر پویای پارامترهای مدل بر اساس دادگان جدید را فراهم می‌کنند و می‌توانند باعث افزایش انعطاف‌پذیری سیستم‌های ردیابی داخلی گردند و بهبود چشمگیری در عملکرد آن‌ها داشته باشند.

# **مراجع**

- [1] R. Uttarwar and J. Valentín, "Indoor Positioning and Machine Learning Algorithms," 2021.
- [2] Z. Li, K. Xu, H. Wang, Y. Zhao, X. Wang, and M. Shen, "Machine-learning-based positioning: A survey and future directions," *IEEE Network*, vol. 33, no. 3, pp. 96-101, 2019.
- [3] T. Kim Geok *et al.*, "Review of indoor positioning: Radio wave technology," *Applied Sciences*, vol. 11, no. 1, p. 279, 2020.
- [4] H. Zou, B. Huang, X. Lu, H. Jiang, and L. Xie, "A robust indoor positioning system based on the procrustes analysis and weighted extreme learning machine," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1252-1266, 2015.
- [5] M. Abbas, M. Elhamshary, H. Rizk, M. Torki, and M. Youssef, "WiDeep: WiFi-based accurate and robust indoor localization system using deep learning," in *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2019, pp. 1-10: IEEE.
- [6] K. S. Kim, S. Lee, and K. Huang, "A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting," *Big Data Analytics*, vol. 3, pp. 1-17, 2018.
- [7] M. Cooper, J. Biehl, G. Filby, and S. Kratz, "LoCo: boosting for indoor location classification combining Wi-Fi and BLE," *Personal and Ubiquitous Computing*, vol. 20, no. 1, pp. 83-96, 2016.
- [8] P. Menéndez, C. Campomanes, K. Trawiński, and J. M. Alonso, "Topology-based indoor localization by means of WiFi fingerprinting with a computational intelligent classifier," in *2011 11th International Conference on Intelligent Systems Design and Applications*, 2011, pp. 1020-1025: IEEE.
- [9] K. Trawiński, J. M. Alonso, and N. Hernández, "A multiclassifier approach for topology-based WiFi indoor localization," *Soft Computing*, vol. 17, pp. 1817-1831, 2013.
- [10] V. Singh, G. Aggarwal, and B. Ujwal, "Ensemble based real-time indoor localization using stray WiFi signal," in *2018 IEEE International Conference on Consumer Electronics (ICCE)*, 2018, pp. 1-5: IEEE.
- [11] R. E. Schapire, "The strength of weak learnability," *Machine learning*, vol. 5, pp. 197-227, 1990.
- [12] D. Sánchez-Rodríguez, P. Hernández-Morera, J. M. Quinteiro, and I. Alonso-González, "A low complexity system based on multiple weighted decision trees for indoor localization," *Sensors*, vol. 15, no. 6, pp. 14809-14829, 2015.
- [13] D. Taniuchi and T. Maekawa, "Robust Wi-Fi based indoor positioning with ensemble learning," in *2014 IEEE 10th International conference on wireless and mobile computing, networking and communications (WiMob)*, 2014, pp. 592-597: IEEE.
- [14] Ó. Belmonte-Fernández, R. Montoliu, J. Torres-Sospedra, E. Sansano-Sansano, and D. Chia-Aguilar, "A radiosity-based method to avoid calibration for indoor positioning systems," *Expert Systems with Applications*, vol. 105, pp. 89-101, 2018.

- [15] D. Ghosh, P. Roy, C. Chowdhury, and S. Bandyopadhyay, "An ensemble of condition based classifiers for indoor localization," in *2016 IEEE international conference on advanced networks and telecommunications systems (ANTS)*, 2016, pp. 1-6: IEEE.
- [16] P. Roy, C. Chowdhury, M. Kundu, D. Ghosh, and S. Bandyopadhyay, "Novel weighted ensemble classifier for smartphone based indoor localization," *Expert Systems with Applications*, vol. 164, p. 113758, 2021.
- [17] N. Samama, *Indoor positioning: technologies and performance*. John Wiley & Sons, 2019.
- [18] F. Safari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568-2599, 2019.
- [19] A. Pérez-Navarro *et al.*, "Challenges of fingerprinting in indoor positioning and navigation," in *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*: Elsevier, 2019, pp. 1-20.
- [20] P. Roy and C. Chowdhury, "A survey of machine learning techniques for indoor localization and navigation systems," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 3, p. 63, 2021.
- [21] C. Zhang and Y. Ma, *Ensemble machine learning: methods and applications*. Springer, 2012.
- [22] B. V. Dasarathy and B. V. Sheela, "A composite classifier system design: Concepts and methodology," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 708-713, 1979.
- [23] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119-139, 1997.
- [24] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, pp. 123-140, 1996.
- [25] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79-87, 1991.
- [26] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241-259, 1992.
- [27] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," presented at the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, 2016.
- [28] M. Segal and Y. Xiao, "Multivariate random forests," *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, vol. 1, no. 1, pp. 80-87, 2011.
- [29] R. Díaz-Uriarte and S. Alvarez de Andrés, "Gene selection and classification of microarray data using random forest," *BMC bioinformatics*, vol. 7, pp. 1-13, 2006.
- [30] G. Gottardi, M. A. Hannan, B. Li, A. Polo, M. Salucci, and F. Viani, "PCA-Based inversion of WiFi signal for robust device-free indoor target detection," in *Journal of Physics: Conference Series*, 2020, vol. 1476, no. 1, p. 012015: IOP Publishing.
- [31] Y. Basiouny, M. Arafa, and A. M. Sarhan, "Enhancing Wi-Fi fingerprinting for indoor positioning system using single multiplicative neuron and PCA algorithm," in *2017 12th International Conference on Computer Engineering and Systems (ICCES)*, 2017, pp. 295-305: IEEE.

- [32] A. Urso, A. Fiannaca, M. La Rosa, V. Ravì, and R. Rizzo, "Data mining: Prediction methods," *Encycl. Bioinforma. Comput. Biol. ABC Bioinforma*, vol. 1, p. 3, 2018.
- [33] 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.
- [34] 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.
- [35] P. Roy, C. Chowdhury, D. Ghosh, and S. Bandyopadhyay, "JUIIndoorLoc: A ubiquitous framework for smartphone-based indoor localization subject to context and device heterogeneity," *Wireless Personal Communications*, vol. 106, pp. 739-762, 2019.
- [36] 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.
- [37] J. Yoo, "Change detection of RSSI fingerprint pattern for indoor positioning system," *IEEE sensors journal*, vol. 20, no. 5, pp. 2608-2615, 2019.
- [38] A. H. Salamat, 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.

# پیوست

## پیوست الف: شبہ کد درخت تصمیم

الگوریتم ساخت درخت تصمیم یک الگوریتم بازگشتی یا به عبارت دیگر خود فراخوان است به این ترتیب که در قسمتی از این الگوریتم مجدداً اجرای الگوریتم فراخوانده می‌شود. قسمت‌های رنگ شده الگوریتم بخش‌هایی است که برای استفاده از درخت تصمیم در الگوریتم XGBoost اضافه شده‌اند.

---

**Function Build-Tree( $I, G, H, F_{k-1}$ )**

---

**Input:**

$I$ : the training set

$G$ : sum of the gradients of the samples in  $I$

$H$ : sum of the Hessians of the samples in  $I$

$F_{k-1}$ : the ensemble from the previous iteration

- 1: Create a *root* node for the tree and store the tuple  $(I, G, H)$  in it
  - 2:  $(gain, I_L, G_L, H_L, I_R, G_R, H_R) \leftarrow \text{Find-Best-Split}(I, G, H, F_{k-1})$
  - 3: **if**  $\frac{1}{2} \cdot gain < \gamma$  **then**
  - 4:     **return** *root*
  - 5: **else**
  - 6:     *subtree*  $\leftarrow \text{Build-Tree}(I_L, G_L, H_L, F_{k-1})$
  - 7:     Add *subtree* to the left branch of *root*
  - 8:     *subtree*  $\leftarrow \text{Build-Tree}(I_R, G_R, H_R, F_{k-1})$
  - 9:     Add *subtree* to the right branch of *root*
  - 10:   **return** *root*
- 

در شبہ کد فوق تابع Find-Best-Split استفاده شده است که برای انتخاب بهترین ویژگی در گره به کار برده می‌شود. در این تابع به طور کلی رابطه  $\Delta$  به صورت:

$$\Delta = \sum_{i=1}^c -p_i \log_2 p_i$$

تعریف می‌شود و به آن آنتروپی (Entropy) می‌گویند. شبہ کد فوق تابع Find-Best-Split در ادامه ذکر شده است. در این شبہ کد نیز قسمت‌های رنگ شده، تنها در درخت تصمیمی که به عنوان یادگیرنده ضعیف الگوریتم XGBoost است، استفاده می‌شود:

---

**Function** Find-Best-Split( $I, G, H, F_{k-1}$ )

---

**Input:**

$I$ : the set of samples at the current node  
 $G$ : sum of the gradients of the samples in  $I$   
 $H$ : sum of the Hessians of the samples in  $I$   
 $F_{k-1}$ : the ensemble from the previous iteration

```
1: gain  $\leftarrow 0$ 
2: best_split  $\leftarrow \text{None}$ 
3: /* Consider all the features */
4: for  $j = 1$  to  $m$  do
5:    $I_L \leftarrow \emptyset, G_L \leftarrow 0, H_L \leftarrow 0$ 
6:   /* Compute the gain from each possible split point of feature  $j$  */
7:   Sort the samples in  $I = \{(x_i, y_i)\}$  by the values of  $x_{ij}$ 
8:   for  $i \in I_{\text{sorted}}$  do
9:      $g_i \leftarrow \frac{\partial L(y_i, F_{k-1}(x_i))}{\partial F_{k-1}(x_i)}$ 
10:     $h_i \leftarrow \frac{\partial^2 L(y_i, F_{k-1}(x_i))}{\partial F_{k-1}(x_i)^2}$ 
11:     $I_L \leftarrow I_L \cup \{i\}, G_L \leftarrow G_L + g_i, H_L \leftarrow H_L + h_i$ 
12:     $I_R \leftarrow I - I_L, G_R \leftarrow G - G_L, H_R \leftarrow H - H_L$ 
13:     $\Delta \leftarrow \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{G^2}{H + \lambda}$ 
14:    if  $\Delta > \text{gain}$  then
15:       $gain \leftarrow \Delta$ 
16:       $best\_split \leftarrow (I_L, G_L, H_L, I_R, G_R, H_R)$ 
17: return gain, best_split
```

---

**Abstract:**

Location-based systems (LBSs) enhance Internet of Things (IoT)-based services by providing user location information. While GPS and Galileo are accurate in outdoor spaces, they lack sufficient accuracy indoors. Therefore, Wi-Fi and Bluetooth are utilized for indoor positioning. The received signal strength (RSS) metric is commonly used for positioning with Wi-Fi and Bluetooth due to its simplicity in computation. However, RSS is susceptible to environmental and physical conditions. Machine learning algorithms can estimate user location more accurately in indoor environments, but they require calibration with respect to environmental changes. This thesis investigates the application of ensemble learning algorithms for indoor user location estimation using Wi-Fi. These algorithms improve the location estimation accuracy by combining the outputs of multiple machine learning models. Despite the advantages of ensemble learning, challenges such as increased resource requirements and high processing time exist, which are incompatible with real-time and low-cost systems. To address this issue, two solutions, dimensionality reduction and a genetic algorithm for finding the optimal ensemble model, are proposed. The proposed methods were implemented on collected datasets under various environmental conditions and compared with other methods. The results of this thesis demonstrate that by considering a metric that includes the accuracy and sensitivity of the model estimation in the fitness function, the average positioning error can reach 0.27 meters considering a dataset consisting of all classes. This error is achieved by an ensemble learning model with an accuracy of approximately 95%. For comparison with other studies, the test data was limited to a single class, resulting in an average error of 0.25 meters from a model with an accuracy of 98.3%.

**Keywords:** Indoor positioning, Machine Learning, Ensemble Learning, Genetic Algorithm



**Iran University of Science and Technology  
School of Electrical Engineering**

# **Improving the accuracy of RSS-based positioning systems using ensemble learning algorithms**

**A Thesis Submitted in Partial Fulfillment of the Requirement for the  
Degree of Master of Science in Control**

**By:  
Hamid Arabsorkhi**

**Supervisor:  
Dr. Saeed Ebadollahi**

**March 2024**