



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

گزارش پروژه کارشناسی

ایجاد سامانه موقعیت‌یابی با استفاده از یادگیری ماشین

نگارش
حمیدرضا همتی

استاد راهنما
دکتر مهدی راستی

فروردین ۱۴۰۲

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

به نام خدا

تعهدنامه اصالت

تاریخ: مرداد ۱۴۰۱

اثر

اینجانب حمیدرضا همتی متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی-اینجانب تحت نظارت و راهنمایی اساتید دانشگاه امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذکر گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطلب با ذکر مآخذ بلامانع است.

حمیدرضا همتی

امضا

تقدیم به برادر عزیزم که در سختی‌ها و دشواری‌های
زندگی، همواره یآوری دلسوز و فداکار و پشتیبانی محکم
برایم بوده.

سپاس‌گزاری

از استاد دلسوز و گرانقدرم ؛ جناب دکتر مهدی راستی که در کمال سعه صدر، با حسن خلق و فروتنی، از هیچ کمکی در این عرصه بر من دریغ نداشتند و در سخت‌ترین شرایط همراه من بودند کمال تشکر را دارم.

حمیدرضا همتی

فروردین ۱۴۰۲

چکیده

اینترنت اشیاء مفهوم نوظهوری است که زندگی انسان‌ها را متحول کرده است. این مفهوم دنیایی را توصیف می‌کند که در آن اشیاء مختلف می‌توانند با یکدیگر و با انسان‌ها بر بستر یک فناوری شبکه ارتباط برقرار کنند. در دنیای اینترنت اشیاء مهم‌ترین رکن انتقال و استفاده مناسب از داده‌ها می‌باشد. با بهره‌گیری از الگوریتم‌ها و مفاهیم هوش مصنوعی می‌توان تجزیه و تحلیل مناسبی روی این داده‌ها انجام و پتانسیل نهان آن‌ها را برای ارائه سرویس‌هایی که زندگی را در ابعاد مختلفی بهبود بخشند استفاده کرد.

فناوری‌های ارتباطی دوربرد با توان پایین (LPWAN) امکان اتصال دستگاه‌ها در محدوده‌ای وسیع با مصرف کم توان را فراهم می‌کند. شبکه LoRaWAN (Long Range Wide Area Network) از مطرح‌ترین و پرکاربردترین شبکه‌های LPWAN است. LoRaWAN یک فناوری شبکه بی‌سیم با برد بالا و انرژی مصرفی پایین است که از چهار قسمت اصلی شامل دستگاه انتهایی، دروازه، سرور شبکه و سرور کاربرد تشکیل شده است.

در این پروژه یک زیرساخت برای شبکه LoRaWAN متشکل از ۳ دروازه، سرور شبکه و سرور کاربرد پیاده‌سازی شده است. با استفاده از دستگاه‌های انتهایی، مجموعه داده‌هایی متشکل از قدرت سیگنال‌های دریافتی دروازه‌ها جمع‌آوری و سپس با بهره‌گیری از الگوریتم‌های یادگیری ماشین تلاش شده تا سامانه‌ای برای موقعیت یابی پیاده‌سازی شود.

واژه‌های کلیدی:

اینترنت اشیاء – شبکه‌های دوربرد و کم‌توان – شبکه LoRaWAN – یادگیری ماشین

فهرست مطالب

| صفحه | عنوان |
|------|---|
| ۱ | فصل ۱ مقدمه |
| ۵ | فصل ۲ مفاهیم پایه |
| ۶ | ۱-۲-۱ مقدمه |
| ۶ | ۲-۲-۱ اینترنت اشیاء |
| ۷ | ۱-۲-۲ معماری اینترنت اشیاء |
| ۱۲ | ۳-۲-۱ شبکه‌های دوربرد توان پایین (LPWAN) |
| ۱۲ | ۱-۳-۲ شبکه SigFox |
| ۱۴ | ۲-۳-۲ شبکه LoRaWAN |
| ۱۴ | ۴-۲-۱ فناوری شبکه LoRaWAN |
| ۱۵ | ۱-۴-۲ معرفی LoRa و LoRaWAN |
| ۱۶ | ۲-۴-۲ بخش LoRa |
| ۲۲ | ۳-۴-۲ بخش LoRaWAN |
| ۲۸ | ۴-۴-۲ امنیت شبکه LoRaWAN |
| ۳۰ | ۵-۲-۱ تضعیف سیگنال |
| ۳۱ | ۱-۵-۲ تضعیف ناشی از فاصله |
| ۳۱ | ۲-۵-۲ تضعیف ناشی از جذب |
| ۳۲ | ۳-۵-۲ تضعیف ناشی از انعکاس |
| ۳۲ | ۴-۵-۲ تضعیف ناشی از شکست |
| ۳۳ | ۵-۵-۲ تضعیف ناشی از پراکندگی و انکسار |
| ۳۳ | ۶-۲-۱ پیش پردازش داده‌ها |
| ۳۳ | ۱-۶-۲ فیلتر کالمن گسسته |
| ۳۸ | ۲-۶-۲ فیلتر ذره |
| ۳۹ | ۷-۲-۱ الگوریتم‌های یادگیری نظارت شده |
| ۴۰ | ۱-۷-۲ الگوریتم K نزدیک‌ترین همسایگی |
| ۴۰ | ۲-۷-۲ بردار پشتیبان |
| ۴۱ | ۳-۷-۲ XGBoost |
| ۴۲ | ۴-۷-۲ شبکه عصبی |
| ۴۳ | جمع‌بندی |
| ۴۴ | فصل ۳ پیاده‌سازی زیرساخت سامانه موقعیت‌یابی |
| ۴۵ | ۱-۳-۱ مقدمه |
| ۴۶ | ۲-۳-۱ دستگاه انتهایی شبکه LoRaWAN |
| ۴۶ | ۱-۲-۳ پردازنده دستگاه انتهایی |

| | |
|--|---|
| ۵۳ | ۲-۲-۳- مازول رادیویی LoRa و اتصال آن به پردازنده..... |
| ۵۶ | ۳-۲-۳- منبع تغذیه |
| ۵۹ | ۴-۲-۳- اتصال و راه اندازی دستگاه انتهایی |
| ۶۲ | ۳-۳- دروازه شبکه LoRaWAN..... |
| ۶۸ | ۱-۳-۳- ثبت دروازه ها در پلتفرم TTN..... |
| ۷۱ | ۴-۳- ایجاد سرور شبکه و سرور کاربرد..... |
| ۷۵ | جمع بندی..... |
| فصل ۴ طراحی و پیاده سازی سامانه تشخیص موقعیت مکانی با استفاده از یادگیری ماشین.... ۷۶ | |
| ۷۷ | ۱-۴- مقدمه..... |
| ۷۷ | ۲-۴- مجموعه دادگان..... |
| ۷۷ | ۱-۲-۴- استقرار زیرساخت و ایجاد محوطه تحت پوشش شبکه..... |
| ۸۱ | ۲-۲-۴- جمع آوری داده..... |
| ۸۳ | ۳-۲-۴- تحلیل داده ها..... |
| ۹۲ | ۳-۴- اعمال فیلتر بروی مجموعه دادگان..... |
| ۹۳ | ۱-۳-۴- پیاده سازی فیلتر..... |
| ۹۵ | ۲-۳-۴- تاثیر فیلتر بروی داده ها..... |
| ۱۰۱ | ۴-۴- به کارگیری یادگیری ماشین..... |
| ۱۰۳ | جمع بندی..... |
| فصل ۵ ارزیابی..... ۱۰۴ | |
| ۱۰۵ | ۱-۵- معیار ارزیابی..... |
| ۱۰۶ | ۵-۲- مقایسه نتایج به دست آمده..... |
| فصل ۶ جمع بندی، نتیجه گیری و پیشنهادات..... ۱۱۰ | |
| ۱۱۱ | ۱-۶- جمع بندی و نتیجه گیری..... |
| ۱۱۱ | ۲-۶- پیشنهادات..... |
| ۱۱۳ | منابع..... |

| شکل | فهرست اشکال | صفحه |
|--------|---|------|
| شکل ۱ | مدل معماری ۳ لایه‌ای اینترنت اشیا..... | ۸ |
| شکل ۲ | معماری ۴ لایه‌ای اینترنت اشیا..... | ۸ |
| شکل ۳ | دستگاه انتهایی مربوط به لایه ادراک..... | ۹ |
| شکل ۴ | دسته‌بندی شبکه‌های بی‌سیم از نظر برد [۱۹]..... | ۱۰ |
| شکل ۵ | دسته بندی شبکه‌های بی‌سیم از نظر مصرف انرژی، نرخ داده و برد [۱۹]..... | ۱۱ |
| شکل ۶ | معماری تکنولوژی SigFox [۷]..... | ۱۳ |
| شکل ۷ | توپولوژی شبکه LoRaWAN..... | ۱۶ |
| شکل ۸ | لایه بندی شبکه LoRaWAN برحسب مدل ۷ لایه‌ای OSI..... | ۱۸ |
| شکل ۹ | مدل‌اسیون LoRa با استفاده از CSS [۱۳]..... | ۱۹ |
| شکل ۱۰ | پشته پروتکل LoRaWAN [۱۲]..... | ۲۲ |
| شکل ۱۱ | کلاس نوع A [۱۱]..... | ۲۷ |
| شکل ۱۲ | کلاس نوع B [۱۱]..... | ۲۸ |
| شکل ۱۳ | کلاس نوع C [۱۱]..... | ۲۸ |
| شکل ۱۴ | شماتیک فیلتر kalman..... | ۳۶ |
| شکل ۱۵ | تفسیر معادله ۲..... | ۳۷ |
| شکل ۱۶ | الگوریتم پیش‌بینی و فیلتر مورد استفاده..... | ۳۷ |
| شکل ۱۷ | مراحل انجام الگوریتم فیلتر ذره..... | ۳۹ |
| شکل ۱۸ | معماری ساده شبکه عصبی مصنوعی..... | ۴۲ |
| شکل ۱۹ | برد Arduino pro mini و ابعاد آن..... | ۴۷ |
| شکل ۲۰ | مقایسه جریان مصرفی بردهای Arduino..... | ۴۷ |
| شکل ۲۱ | تغییرات لازم بروی فایل config.h کتابخانه LMIC..... | ۴۹ |
| شکل ۲۲ | تغییر مقدار ورودی تابع LMIC_selectSubBand به منظور کارکرد دستگاه در فرکانس ۹۱۵ مگاهرتز..... | ۵۰ |
| شکل ۲۳ | تغییرات لازم بروی کد برای ارسال داده مرد نظر در بازه‌های زمانی دلخواه..... | ۵۱ |
| شکل ۲۴ | کدهای مربوط به تنظیم عامل گسترش، پهنای باند و توان ارسالی..... | ۵۲ |
| شکل ۲۵ | نرخ داده‌های متفاوت با ترکیب عامل گسترش و پهنای باندهای مختلف..... | ۵۲ |
| شکل ۲۶ | نگاشت پایه‌ها (pin mapping) درون کد دستگاه انتهایی..... | ۵۳ |
| شکل ۲۷ | ماژول رادیویی LoRa Hope RF96..... | ۵۴ |
| شکل ۲۸ | اتصالات بین میکرو کنترلر و ماژول LoRa..... | ۵۵ |
| شکل ۲۹ | نمای PCB طراحی شده در نرم افزار [۲۰]..... | ۵۶ |
| شکل ۳۰ | برد PCB چاپ شده..... | ۵۶ |

| | | |
|--------|---|----|
| شکل ۳۱ | باتری قابل شارژ Sonikcell با ظرفیت ۱۰۰۰ میلی آمپر ساعت (سمت راست) و ۲۰۰۰ میلی آمپر ساعت (سمت چپ)..... | ۵۷ |
| شکل ۳۲ | ماژول TP4056..... | ۵۷ |
| شکل ۳۳ | ماژول تغذیه..... | ۵۸ |
| شکل ۳۴ | آنتن‌های ۹۱۵ مگاهرتز در سایز کوچک و بزرگ..... | ۵۹ |
| شکل ۳۵ | دستگاه انتهایی..... | ۵۹ |
| شکل ۳۶ | دستگاه انتهایی به همراه ماژول تغذیه..... | ۶۰ |
| شکل ۳۷ | کلید NwkSKey و AppSKey و آدرس دستگاه در پلتفرم TTN..... | ۶۱ |
| شکل ۳۸ | تعریف کلید NwkSKey و AppSKey و آدرس دستگاه درون کد Arduino..... | ۶۱ |
| شکل ۳۹ | مشخصات پیام دریافتی دستگاه انتهایی در پلتفرم TTN..... | ۶۲ |
| شکل ۴۰ | ماژول فرستنده-گیرنده RAK831..... | ۶۳ |
| شکل ۴۱ | محیط پیکربندی Raspberry pi..... | ۶۴ |
| شکل ۴۲ | پنجره رابط‌های Raspberry pi..... | ۶۴ |
| شکل ۴۳ | فعالسازی رابط جانبی سریال..... | ۶۵ |
| شکل ۴۴ | درخواست reboot شدن دستگاه..... | ۶۵ |
| شکل ۴۵ | نحوه اتصال پایه های Raspberry pi و RAK831..... | ۶۷ |
| شکل ۴۶ | صفحه Gateways در پلتفرم TTN..... | ۶۸ |
| شکل ۴۷ | تایید آدرس یکتای دروازه..... | ۶۹ |
| شکل ۴۸ | انتخاب نام و فرکانس کاری دروازه..... | ۷۰ |
| شکل ۴۹ | ثبت کامل دروازه و دریافت فایل global_conf.json..... | ۷۱ |
| شکل ۵۰ | پلتفرم TTN..... | ۷۳ |
| شکل ۵۱ | بخش کاربردها در پلتفرم TTN..... | ۷۳ |
| شکل ۵۲ | سرور کاربرد در پلتفرم TTN..... | ۷۴ |
| شکل ۵۳ | روند جریان داده با توجه به کلیدهای جلسه و شبکه [۱۰]..... | ۷۵ |
| شکل ۵۴ | محل قرارگیری دروازه‌های شبکه LoRaWAN و فضای شبکه ایجاد شده..... | ۷۸ |
| شکل ۵۵ | فاصله دروازه‌های قرار گرفته در دانشکده‌های کامپیوتر و معدن..... | ۷۸ |
| شکل ۵۶ | فاصله دروازه‌های قرار گرفته در دانشکده‌های عمران و معدن..... | ۷۹ |
| شکل ۵۷ | فاصله دروازه‌های قرار گرفته در دانشکده‌های عمران و کامپیوتر..... | ۷۹ |
| شکل ۵۸ | تقسیم بندی نواحی در منطقه تحت پوشش شبکه..... | ۸۱ |
| شکل ۵۹ | مجموعه داده جمع‌آوری شده..... | ۸۲ |
| شکل ۶۰ | هیستوگرام داده‌های دریافتی دروازه‌ها، از سمت راست، دانشکده‌های کامپیوتر، عمران و معدن..... | ۸۴ |
| شکل ۶۱ | هیستوگرام داده‌های دریافتی توسط دروازه دانشکده کامپیوتر در تمام نواحی..... | ۸۶ |
| شکل ۶۲ | هیستوگرام داده‌های دریافتی توسط دروازه دانشکده عمران در تمام نواحی..... | ۸۷ |
| شکل ۶۳ | تاثیر نویز بر داده‌های جمع‌آوری شده دروازه دانشکده کامپیوتر از ناحیه ۵..... | ۸۸ |

| | | |
|--------|---|-----|
| شکل ۶۴ | تاثیر نویز بر داده‌های جمع‌آوری شده دروازه دانشکده عمران از ناحیه ۵ | ۸۸ |
| شکل ۶۵ | تاثیر نویز بر داده‌های جمع‌آوری شده دروازه دانشکده معدن از ناحیه ۵ | ۸۸ |
| شکل ۶۶ | داده‌های دریافتی دروازه دانشکده کامپیوتر در تمام نواحی | ۸۹ |
| شکل ۶۷ | داده‌های دریافتی دروازه دانشکده عمران در تمام نواحی | ۸۹ |
| شکل ۶۸ | داده‌های دریافتی دروازه دانشکده معدن در تمام نواحی | ۹۰ |
| شکل ۶۹ | پراکندگی مجموعه‌دادگان به صورت سه‌بعدی | ۹۱ |
| شکل ۷۰ | پراکندگی مجموعه‌دادگان به صورت خطی | ۹۱ |
| شکل ۷۱ | پیاده‌سازی تابع تقسیم‌بند مقادیر دروازه-ناحیه | ۹۳ |
| شکل ۷۲ | پیاده‌سازی تابع مربوط به میانگین مقادیر ۲۱ دروازه-ناحیه | ۹۴ |
| شکل ۷۳ | پیاده‌سازی فیلتر $\alpha-\beta-\gamma$ | ۹۵ |
| شکل ۷۴ | مقایسه داده‌های خام و فیلتر شده در ۷ ناحیه از دروازه دانشکده کامپیوتر | ۹۷ |
| شکل ۷۵ | مقایسه داده‌های خام و فیلتر شده در ۷ ناحیه از دروازه دانشکده عمران | ۹۸ |
| شکل ۷۶ | مقایسه داده‌های خام و فیلتر شده در ۷ ناحیه از دروازه دانشکده معدن | ۹۹ |
| شکل ۷۷ | پراکندگی مجموعه‌دادگان بعد از اعمال فیلتر به صورت سه‌بعدی | ۱۰۰ |
| شکل ۷۸ | پراکندگی مجموعه‌دادگان بعد از اعمال فیلتر به صورت خطی | ۱۰۱ |
| شکل ۷۹ | قطعه‌کد پیاده‌سازی الگوریتم‌های یادگیری ماشین | ۱۰۲ |
| شکل ۸۲ | شمایی از ماتریس درهم‌ریختگی | ۱۰۵ |
| شکل ۸۰ | نتایج الگوریتم‌های یادگیری ماشین بر روی داده‌های خام | ۱۰۶ |
| شکل ۸۱ | نتایج الگوریتم‌های یادگیری ماشین بر روی داده‌های فیلتر شده | ۱۰۷ |
| شکل ۸۳ | ماتریس درهم‌ریختگی الگوریتم SVM بر روی داده‌های خام | ۱۰۸ |
| شکل ۸۴ | ماتریس درهم‌ریختگی الگوریتم SVM بر روی داده‌های فیلتر شده | ۱۰۹ |

فصل ١

مقدمه

امروزه، عبارت «اینترنت اشیاء»^۱ همه‌ی دنیای فناوری اطلاعات و ارتباطات را فراگرفته است. شبکه هوشمند انرژی، کشاورزی هوشمند، دامپروری هوشمند، خانه و ساختمان هوشمند، حمل‌ونقل هوشمند، سلامت هوشمند و به‌طور کلی شهر و محیط‌زیست هوشمند، اصطلاحاتی هستند که شمار بسیاری از سیاستمداران، مدیران، متخصصان و کسب‌وکارها را در سراسر جهان، شیفته‌ی ویژگی‌های بی‌مانند این جلوه‌ی نوپیدای فناوری کرده‌اند. امروزه اینترنت اشیاء با راهکارهای هوشمند خود در همه‌ی ابعاد زندگی بشر وارد شده است و نویدبخش آینده‌ای بهتر، دلپذیر و آسوده‌تر است.

اکوسیستم اینترنت اشیاء مجموعه‌ای از دستگاه‌های هوشمند مبتنی بر شبکه است که با بهره‌گیری از پردازنده‌های جاسازی‌شده، سنسورها و سخت‌افزار ارتباطی، به گردآوری، ارسال و پردازش بروی داده‌های دریافت‌شده از محیط می‌پردازد. داده‌های گردآوری شده توسط حسگرها از طریق اتصال به یک درگاه IoT یا دستگاه لبه^۲ برای تحلیل به فضای ابری ارسال شده یا اینکه به‌صورت محلی تحلیل می‌شوند. گاهی اوقات، این دستگاه‌ها با سایر دستگاه‌های مرتبط، ارتباط برقرار کرده و برپایه‌ی اطلاعاتی که از یکدیگر می‌گیرند، عمل می‌کنند. این دستگاه‌ها بیشتر کارها را بدون دخالت انسان انجام می‌دهند. البته افراد می‌توانند با دستگاه‌ها تعامل داشته باشند؛ برای نمونه، می‌توانند آن‌ها را تنظیم کنند؛ به آن‌ها دستورالعمل بدهند یا به داده‌ها دسترسی پیدا کنند.

یکی از شبکه‌های مهم و پرکاربرد مورد استفاده در اینترنت اشیاء شبکه لوراون^۳ است. شبکه لوراون یکی از پروتکل‌های^۴ اصلی دوربرد و کم‌مصرف^۵ ویژه اینترنت اشیاء است [۱]. این فناوری به سیگنال‌ها اجازه می‌دهد تا حتی در سطوح پایین‌تر از نویز نیز منتشر و بازیابی شوند. تجهیزات مبتنی بر لوراون می‌توانند تا سال‌ها فقط با یک باتری کار کنند. یکی از مهمترین مشخصه‌های شبکه لورا که توانسته است در کنار مزیت‌های فنی این پروتکل زمینه رشد سریع آن را فراهم کند، رویکرد غیر انحصاری توسعه این پروتکل بر بستر یک جامعه آزاد و با مشارکت مجموعه‌های مختلف فناوری باشد.

^۱ Internet of Things

^۲ Edge device

^۳ LoRaWAN

^۴ Protocol

^۵ Long range wide area network (LPWAN)

فناوری‌های شبکه دوربرد و کم مصرف مانند لوراون با هدف ارائه راه‌حل‌های با توان مصرفی پایین برای کاربردهای زمینه اینترنت اشیا به وجود آمدند. برای بسیاری از کاربردها، خدمات مبتنی بر مکان^۱ مانند موقعیت یابی و تعیین مکان فیزیکی دستگاه‌ها مهم و حیاتی است. شبکه لوراون با توجه به مشخصات سیگنالی که دارد دارای تضعیف کم، نفوذ سیگنال و برد بسیار زیاد در حدود ۱۰ تا ۳۰ کیلومتر در محیط‌های شهری و غیر شهری است [۲]. این مشخصات باعث می‌شود بتوان از این شبکه برای مکان یابی هم در محیط‌های شهری و هم محیط‌های سرپوشیده^۲ استفاده کرد.

هدف از انجام این پروژه مکان‌یابی با استفاده از شبکه لوراون در محیط‌های شهری و فضای باز است. با وجود گستردگی استفاده از سیستم تعیین موقعیت جهانی^۳ تراشه‌های این سیستم گران قیمت و دارای مصرف توان زیادی می‌باشند که استفاده از آن را برای دستگاه‌های کم مصرف و دوربرد اینترنت اشیا نامناسب می‌کند.

روش‌های مختلفی برای توسعه‌ی سیستم تعیین موقعیت ارائه شده که عموماً بر اساس دریافت امواج رادیویی ارسالی از فرستنده‌هایی با موقعیت مشخص هستند. زمان دریافت سیگنال، اختلاف زمان دریافت سیگنال، زاویه دریافت و اثرانگشت مکانی از جمله این روش‌ها هستند. روش‌های مبتنی بر زمان دریافت سیگنال، اختلاف زمان دریافت سیگنال و زاویه دریافت سیگنال برپایه تکنیک‌های مثلث بندی هستند که نیاز به دید مستقیم فرستنده و گیرنده خواهد بود. همچنین سنجش دقیق زمان و زاویه سیگنال دریافتی نیاز به ابزارهای خاص دارند که در بیشتر مواقع گران و پرهزینه هستند.

برای تعیین موقعیت، روش اثرانگشت مکانی می‌تواند به عنوان روشی بهینه مورد استفاده قرار گیرد. روش اثرانگشت مکانی به علت عدم نیاز به زیرساخت ویژه و امکان ایجاد ساده‌تر، به عنوان یک روش رایج مورد استفاده قرار می‌گیرد. این روش برای تخمین موقعیت دستگاه همراه کاربر از قدرت سیگنال دریافتی^۴ استفاده می‌کند.

هدف این پروژه طراحی و برپایی یک شبکه لوراون متشکل از گیرنده^۵ و فرستنده‌های بی‌سیم کم مصرف مختص شبکه لورا جهت جمع آوری داده‌های قدرت سیگنال دریافتی می‌باشد. سپس بعد از جمع‌آوری

^۱ Location-based services (LBS)

^۲ indoor

^۳ Global positioning System (GPS)

^۴ Received Signal Strength Indicator (RSSI)

^۵ Gateway

داده‌های مورد نیاز با به کارگیری روش اثر مکانی^۱ جهت سنجش قدرت سیگنال دریافتی و پیاده سازی روش‌های یادگیری ماشین اعم از الگوریتم K نزدیک ترین همسایگی^۲ و شبکه عصبی به تحلیل این داده‌ها پرداخته می‌شود.

مقصود نهایی این است که با استفاده تحلیل‌های به دست آمده موقعیت نسبی دستگاه فرستنده لوراون را پیدا کرد. یکی از چالش‌های مهم این پروژه مسئله جمع آوری داده می‌باشد. از این جهت که قرار است داده‌های جمع آوری شده به وسیله روش‌های یادگیری ماشین تحلیل شوند باید مقدار زیادی داده برای آموزش مدل هوش مصنوعی جمع آوری کرد. همچنین از دیگر چالش‌های این پروژه نويز پذیری داده‌های قدرت سیگنال دریافتی هست. بنابراین باید از روش‌ها و تکنیک‌های فیلتر کردن مانند فیلتر کالمن^۳ برای آماده سازی و حذف داده‌های پرت پیش از تحلیل داده‌ها استفاده کرد.

^۱ Location fingerprinting

^۲ K nearest neighbor

^۳ Kalman filter

فصل ۲

مفاهیم پایه

۲-۱- مقدمه

در این فصل مفاهیم و اصول اولیه مورد نیاز برای انجام این پروژه معرفی می شود. در ابتدا مفاهیم اینترنت اشیا و سپس شبکه های دوربرد کم توان که از مهم ترین شبکه های موجود برای کاربردهای اینترنت اشیا هست معرفی می شود. سپس به بررسی شبکه LoRaWAN پرداخته شده و ویژگی های برجسته آن مطرح می شود. در ادامه مفاهیمی چون تضعیف سیگنال، الگوریتم های پیش پردازش داده و روش های یادگیری نظارت شده که برای انجام این پروژه استفاده شده، توضیح داده شده اند.

۲-۲- اینترنت اشیا

ارتباط از سان با کامپیوترها و یا تلفن های هوشمند یک پدیده رایج در قرن بیست و یکم، به ویژه در چند سال اخیر است و این امر بدون بهره وری از تکنولوژی شبکه های بی سیم غیرممکن است. اینترنت اشیا یک اکوسیستم برای برقراری ارتباط و انتقال داده بین دستگاه های مختلف دیجیتال است [۳]. به عبارتی اشیا و تجهیزات محیط پیرامون مان که به شبکه اینترنت متصل شده و توسط برنامه های کاربردی^۱ موجود در تلفن های هوشمند و سیستم های رایانه ای قابل کنترل و مدیریت هستند را سامانه مبتنی بر اینترنت اشیا می گویند.

واژه اینترنت اشیا برای نخستین بار در سال ۱۹۹۹ توسط کوین اشتون^۲، یکی از هم بنیان گذاران مرکز شناسایی خودکار، در جلسه ای در شرکت P&G مطرح شد. اشتون به منظور آگاه کردن مدیران ارشد از سامانه بازشناسی با امواج رادیویی، ارائه خود را اینترنت اشیا نام نهاد. بعد از آن مرکز شناسایی خودکار چشم انداز اینترنت اشیا خود را در سال ۲۰۰۱ ارائه کرد.

^۱ Application

^۲ Kevin Ashton

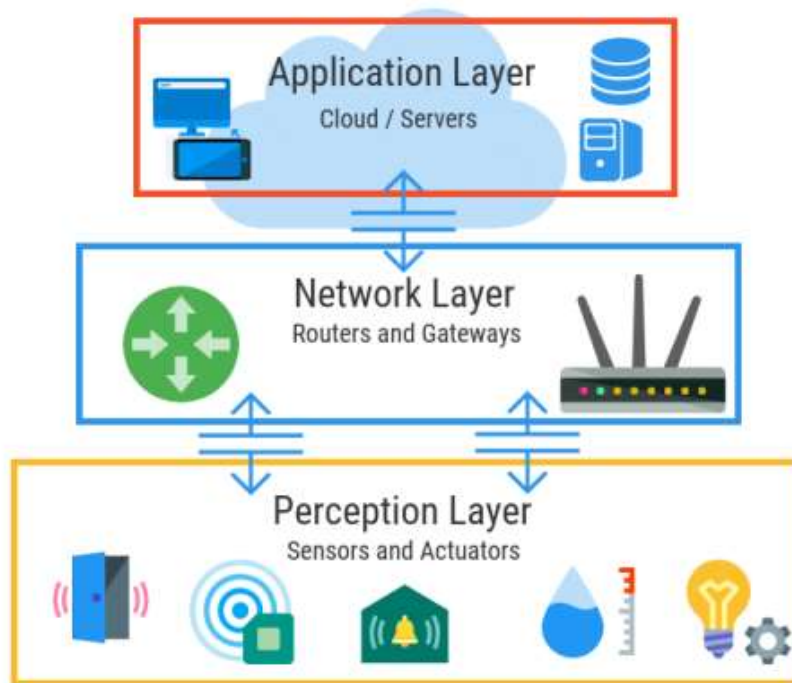
۲-۲-۱- معماری اینترنت اشیاء

برای درک بهتر ساختار این تکنولوژی ابتدا باید معماری آن بررسی شود. معماری اینترنت اشیاء یک معماری لایه‌ای است که هر لایه با لایه‌های بالا و پایین خود در ارتباط است. مدل مرجع و معمول معماری اینترنت اشیاء دارای ۳ لایه اصلی می‌باشد که متشکل از لایه ادراک^۱ (دستگاه‌های انتهایی)، لایه شبکه و لایه کاربرد^۲ است [۴]. علاوه بر مدل مرجع ۳ لایه‌ای، مدل ۴ لایه‌ای نیز ارائه شده که یک لایه به نام پلتفرم^۳ در بین لایه شبکه و لایه کاربرد قرار می‌گیرد و وظیفه آن تجمیع، تحلیل و پردازش داده‌های جمع‌آوری شده است. در شکل‌های ۱ و ۲ به ترتیب مدل ۳ لایه‌ای و ۴ لایه‌ای معماری اینترنت اشیاء نمایش داده شد است.

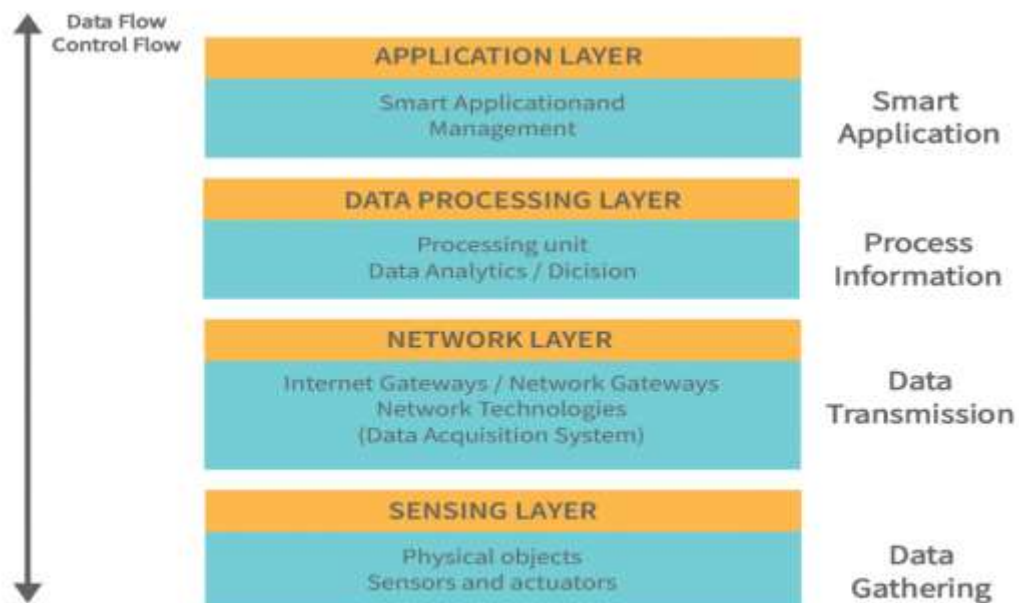
^۱ Perception Layer

^۲ Application Layer

^۳ Platform



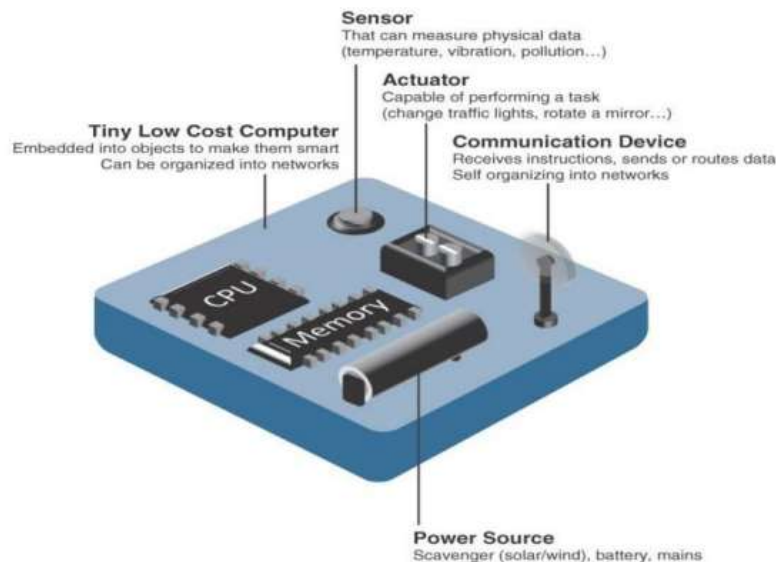
شکل ۱ مدل معماری ۳ لایه‌ای اینترنت اشیا



شکل ۲ معماری ۴ لایه‌ای اینترنت اشیا

در این پروژه از معماری ۴ لایه استفاده شده است. در ادامه هر یک از لایه‌های موجود در معماری ۴ لایه‌ای به اختصار توضیح داده می‌شوند.

لایه درک: این لایه که در پایین‌ترین سطح قرار دارد شامل حسگرها^۱، محرک‌ها^۲ و دستگاه‌هایی است که داده‌ها و یا پارامترهای فیزیکی و محیطی را می‌پذیرند و این اطلاعات را پردازش کرده و از طریق شبکه منتشر می‌کنند. وظیفه اصلی حسگرها جمع‌آوری اطلاعات از محیط پیرامون است و انواع مختلفی از جمله حسگر دما، حرارت، روشنایی، شتاب و مانند آن‌ها را شامل می‌شود. پس از آن داده‌های جمع‌آوری شده را به صورت سیگنال الکترونیکی ارسال می‌کنند. محرک‌ها نیز می‌توانند با پردازش داده‌ها عملی را در محیط انجام دهند. در شکل ۳ ساختار کلی یک دستگاه انتهایی را مشاهده می‌نمایید.

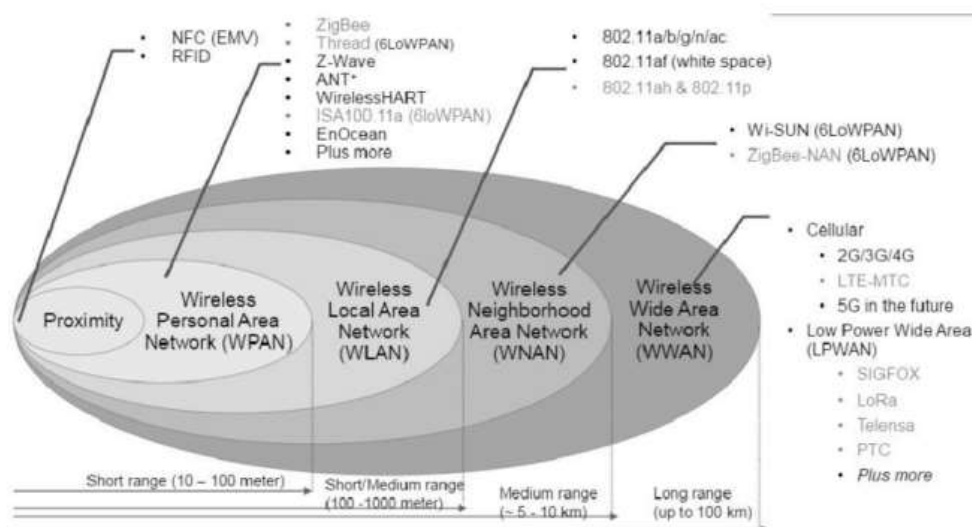


شکل ۳ دستگاه انتهایی مربوط به لایه ادراک

¹ Sensor

² Actuators

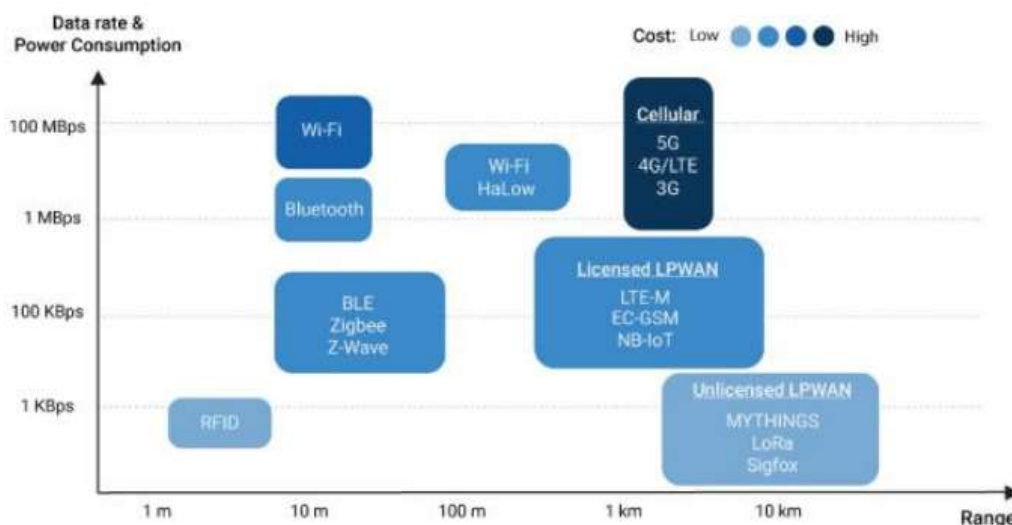
لایه شبکه: داده‌های جمع‌آوری شده به وسیله حسگرها از طریق لایه شبکه منتشر می‌شود. فناوری‌های شبکه به طور کلی به دو دسته شبکه سیمی^۱ و شبکه بی‌سیم^۲ تقسیم می‌شوند که در کاربردهای اینترنت اشیاء غالباً از شبکه‌های بی‌سیم استفاده می‌شود. فناوری‌های شبکه بی‌سیم را می‌توان از منظرهای مختلفی دسته‌بندی کرد. به عنوان مثال می‌توان آن‌ها را از نظر برد و پوشش شبکه و یا انرژی مصرفی در دسته‌های متفاوت قرار داد. در شکل‌های ۴ و ۵ فناوری‌های شبکه با معیارهای برد، مصرف انرژی و نرخ داده مقایسه شده‌اند.



شکل ۴ دسته‌بندی شبکه‌های بی‌سیم از نظر برد [۱۹]

^۱ Wired Network

^۲ Wireless Network



شکل ۵ دسته بندی شبکه‌های بی‌سیم از نظر مصرف انرژی، نرخ داده و برد^۱ [۱۹]

با توجه به اهمیت طول عمری باتری‌ها در اینترنت اشیاء و همچنین نیاز به ارسال داده‌ها در دوره‌های مشخص زمانی و یا با رخداد اتفاقی، شبکه مورد استفاده در اکوسیستم اینترنت اشیاء نیاز به پهنای باند زیادی ندارد و دارا بودن توان مصرفی پایین و برد^۱ بالا از مهم‌ترین ویژگی‌هایی است که این شبکه نیاز دارد. از جمله شبکه‌های دوربرد با توان پایین در حوزه اینترنت اشیاء می‌توان به SigFox و LoRaWAN که در بخش ۲-۳ توضیح داده شده اشاره کرد.

لایه پردازش داده: در این لایه داده‌های جمع‌آوری شده به وسیله حسگرها قبل از ارسال به لایه بالاتر پردازش شده و اطلاعات نهایی بدست می‌آید. امروزه سازمان‌ها و شرکت‌های بسیاری اقدام به سرمایه‌گذاری در این حوزه کرده‌اند.

لایه کاربرد: این لایه به خانواده بزرگی از برنامه‌های کاربردی که ممکن است مختص یک صنعت بخصوص و یا چندین صنعت طراحی و پیاده‌سازی شده باشند، اشاره دارد و عهده‌دار نمایش گرافیکی اطلاعات است. این لایه بالاترین لایه از معماری اینترنت اشیاء است و برنامه‌های استفاده شده در این لایه باید در جهت برقراری ارتباط با لایه پردازش داده توسعه یابند. [۵]

^۱ Range

۲-۳- شبکه‌های دوربرد توان پایین (LPWAN)

با افزایش کاربردهای اینترنت اشیاء، نیاز به توسعه تکنولوژی‌های بی‌سیم با مصرف انرژی پایین و برد سیگنال بالا افزایش پیدا کرد. فناوری شبکه‌های دوربرد توان پایین یا شبکه‌های LPWAN (یا LPWA) امکان اتصال دستگاه‌ها در محدوده‌ای وسیع، با مصرف توان کم را فراهم می‌کند. LPWAN عنوان طیف گسترده‌ای از پروتکل‌ها و فناوری‌هاست که امکان اتصال سنسورها و کنترل‌کننده‌ها را بدون استفاده از شبکه‌های قدیمی همچون WiFi و شبکه‌های تلفن همراه فراهم می‌کند. شبکه‌های LPWAN نسبت به غالب شبکه‌های بی‌سیم متداول نرخ داده کمتری دارد. با استفاده از این شبکه سنسورها و دستگاه‌ها می‌توانند اطلاعات را در محدوده چندین کیلومتر ارسال کنند و برای این کار با باتری AA معمول تا سال‌ها کار خواهند کرد.

بسیاری از راهکارهای اینترنت اشیاء به سنسورهایی نیاز دارند تا داده‌ها را در محدوده وسیعی منتشر کنند. کاربردهای آن را می‌توان در مجتمع‌های ساختمانی هوشمند، شهر هوشمند، صنعت هوشمند و غیره دید. زمانی که سنسورهای زیاد را در محدوده وسیعی قرار می‌دهید، برای اتصال آن‌ها به ارتباط بی‌سیم با برد بلند و توان مصرفی پایین نیاز دارید.

تا اوایل سال ۲۰۱۳ عبارتی به نام LPWA یا Low Power Wide Area (دوربرد با توان پایین) وجود نداشت، اما نیازمندی و شواهدی که از پتانسیل بسیار بالای تکنولوژی‌های LPWA بدست آمد، این فناوری را تبدیل به یکی از جنبه‌های مهم و در حال توسعه بازار اینترنت اشیاء کرد. البته مسأله‌ای که لازم است به آن توجه شود این است که شبکه‌های LPWAN با ایجاد ناحیه پوشش وسیع، با هزینه پایین‌تر و مصرف توان بهینه، به عنوان مکمل و نه جایگزین شبکه‌های سلولی و فناوری‌های برد کوتاه به حساب می‌آیند.

۲-۳-۱- شبکه SigFox

شبکه SigFox ابداع شرکتی فرانسوی است که در سال ۲۰۰۹ شروع به فعالیت نموده و عمده تمرکز این شرکت، ساخت و ایجاد یک بستر مناسب برای شبکه‌ای کم مصرف می‌باشد. این شرکت در سال ۲۰۱۲ از شبکه SigFox خود رونمایی کرد و هم‌اکنون منطقه بزرگی از دنیا تحت پوشش این شبکه است. از اهداف بلند مدت این شرکت می‌توان به سعی برای کاهش مازول‌ها، ایجاد سازوکاری که سخت‌افزارها بتوانند بدون نیاز به باتری و انرژی‌های بادی و خورشید کار کنند، اشاره کرد. این تکنولوژی از توپولوژی ستاره استفاده می‌کند. این شبکه وابستگی به

اپراتورهای همراه ندارد و شبکه مخصوص خود را داراست. سرعت انتقال اطلاعات در این شبکه پایین و حجم داده مورد انتقال نیز محدود است و همچنین در این شبکه مسافت‌های بالا پوشش داده می‌شوند. داده‌ها تا حجم ۱۲ بایت در در باندهای فرکانسی بدون نیاز به مجوز (ISM) و از طریق مدولاسیون فوق باند باریک^۱ (UNB) منتقل می‌شود. در این شبکه روزانه تنها ۱۴۰ پیام انتقال داده می‌شود. پروتوکل‌های لایه شبکه SigFox محرمانه بوده و تنها در اختیار کمپانی آن است.

در بحث مدل تجاری، SigFox رویکردی بالا به پایین اتخاذ کرده‌است. این شرکت تمام فناوری‌های لازم، از بخش دیتاها و سرور ابری تا نرم‌افزار سمت دستگاه‌های انتهایی را به صورت کامل و انحصاری در اختیار خود دارد و تنها بخشی که به سایر شرکت‌ها اجازه مشارکت داده می‌شود، فناوری سمت دستگاه‌های انتهایی است که در حال حاضر شرکت‌هایی نظیر Atmel و STMicroelectronics، ماژول‌های رادیویی این شرکت را تولید می‌کنند. همانطور که در شکل ۶ نشان داده شده، داده‌ها از طریق گره‌های حسگر از محیط دریافت شده و به دروازه SigFox ارسال می‌شود و از طریق درگاه امن به بخش پلتفرم و برنامه کاربردی ارسال می‌شود. [۶]



شکل ۶ معماری تکنولوژی SigFox [۷]

^۱ Ultra Narrow Band

۲-۳-۲- شبکه LoRaWAN

شبکه LoRaWAN یکی از شبکه‌هایی است که در باند بدون مجوز^۱ فرکانسی، کار می‌کند. باند بدون مجوز فرکانسی توسط سازمان‌های مقرراتی هر کشور برای کاربردهای صنعتی، علمی و پزشکی در نظر گرفته شده‌است. استفاده از این باند فرکانسی به مجوز فرکانسی، نیاز ندارد. این فناوری دستگاه‌ها را در محدوده وسیع و با توان پایین متصل می‌کند. در هر بسته LoRaWAN می‌توان داده‌هایی تا اندازه ۲۴۳ بایت را ارسال کرد. پوشش جغرافیایی هر دروازه شبکه LoRaWAN در مناطق باز و حومه شهر تا ۱۵ کیلومتر نیز می‌رسد. این پروتکل توسط انجمن LoRa با پیش از ۵۰۰ عضو از سراسر دنیا حمایت و پشتیبانی می‌شود. مهم‌ترین مشخصه شبکه LoRaWAN که توانسته در کنار مزیت‌های فنی این شبکه زمینه رشد سریع آن را فراهم کند. رویکرد غیر انحصاری توسعه این شرکت پروتکل بر بستر یک جامعه آزاد و با مشارکت مجموعه‌های مختلف فناوری باشد. در بخش ۲-۴ جزئیات شبکه LoRaWAN تشریح خواهد شد.

۲-۴- فناوری شبکه LoRaWAN

همانطور که در بخش ۲-۳ گفته شد، شبکه LoRaWAN پروتکلی در حوزه شبکه LPWAN است که توان مصرفی پایین و برد وسیع دارد. این شبکه از دو بخش LoRa و LoRaWAN تشکیل شده است. در ادامه هر بخش توضیح داده شده و جزئیات این شبکه بررسی می‌شود.

^۱ Unlicensed

۲-۴-۱- معرفی LoRa و LoRaWAN

LoRa یک فناوری مدولاسیون^۱ فرکانس‌های رادیویی^۲ برای شبکه‌های دوربرد کم‌توان (LPWAN) است [۱]. این نام که از مفهوم برد و سیع^۳ گرفته شده، اشاره‌ای به پیوندهای داده بسیار دوربرد است که این فناوری فعال می‌کند. LoRa که توسط Semtech برای استانداردسازی LPWAN ها ایجاد شده است، ارتباطات دوربرد را تأمین می‌کند: حداکثر تا پنج کیلومتر در مناطق شهری و ۱۵ کیلومتر یا بیشتر در مناطق روستایی [۲]. یکی از ویژگی‌های کلیدی راه حل های مبتنی بر LoRa، نیاز به توان فوق‌العاده کم است، که امکان ایجاد دستگاه‌هایی با باتری را فراهم می‌کند که می‌توانند تا ۱۰ سال دوام بیاورند. این شبکه از توپولوژی ستاره‌ای^۴ استفاده می‌کند که می‌توانید این توپولوژی را در شکل ۷ مشاهده کنید [۱۰].

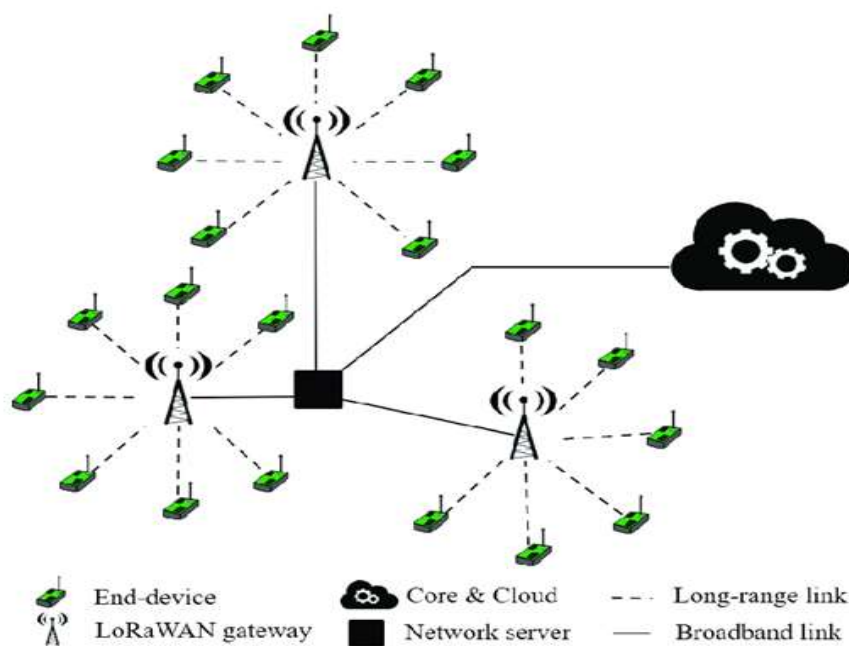
گرچه لایه فیزیکی LoRa در اختیار شرکت Semtech می‌باشد اما لایه MAC این فناوری به نام LoRaWAN یک فناوری متن-باز است و می‌توان به راحتی آن را پیاده‌سازی و از آن استفاده کرد. این پروتکل و فناوری شبکه برای کاربردهایی که نیاز به برد بالا یا ارتباطاتی در عمق زیاد در ساختمان‌ها دارند، و همچنین نیاز به توان مصرفی پایین (دستگاه‌هایی که با باتری کار می‌کنند)، و نرخ ارسال اطلاعات پایین دارند، یک انتخاب بسیار مناسب می‌باشد.

^۱ Modulation

^۲ Radio Frequencies (RF)

^۳ Long Range

^۴ Star of Stars



شکل ۷ توپولوژی شبکه LoRaWAN

۲-۴-۲- بخش LoRa

LoRa یک روش مدلاسیون طیف گسترده^۱ مشتق شده از روش طیف گسترده جاروب^۲ (CSS) می‌باشد که یک مصالحه^۳ بین نرخ ارسال داده و حساسیت سیگنال را پیشنهاد می‌دهد. این فناوری روی کانال‌هایی با پهنای باند ثابت ۲۵۰ کیلوهرتز و ۵۰۰ کیلوهرتز (برای کانال uplink) و ۵۰۰ کیلوهرتز (برای کانال downlink) عمل می‌کند [۱۳]. علاوه بر این، LoRa از عامل‌های گسترش^۴ متعامد^۵ استفاده می‌کند. همچنین فناوری شبکه

^۱ Spread spectrum

^۲ Chirp Spread Spectrum

^۳ Trade-off

^۴ Spreading Factor(SF)

^۵ Orthogonal

LoRa از الگوریتم نرخ داده تطبیقی استفاده می‌کند. نرخ داده تطبیقی^۱ به فناوری LoRa اجازه می‌دهد که نرخ داده و قدرت سیگنال ارسالی در گره‌های انتهایی قابل تغییر و متناسب با قدرت سیگنال دریافتی از گره انتهایی در دروازه‌ها باشد. به این صورت شبکه می‌تواند طول عمر باتری دستگاه‌های انتهایی را افزایش داده و عملکردش را به بهینه‌سازی کند [۲].

به عنوان مثال، یک دستگاه انتهایی که نزدیک یک دروازه قرار دارد باید داده‌ها را با عامل گسترش کم منتقل کند، زیرا بودجه پیوند بسیار کمی مورد نیاز است. با این حال، یک دستگاه انتهایی که در فاصله چندین کیلومتری از یک دروازه قرار دارد، باید از عامل گسترش بیشتری استفاده کند. این عامل گسترش بیشتر، افزایش پردازش و حساسیت دریافت بالاتر را فراهم می‌کند، اگرچه ضرورتاً سرعت انتقال داده را کمتر خواهد کرد.

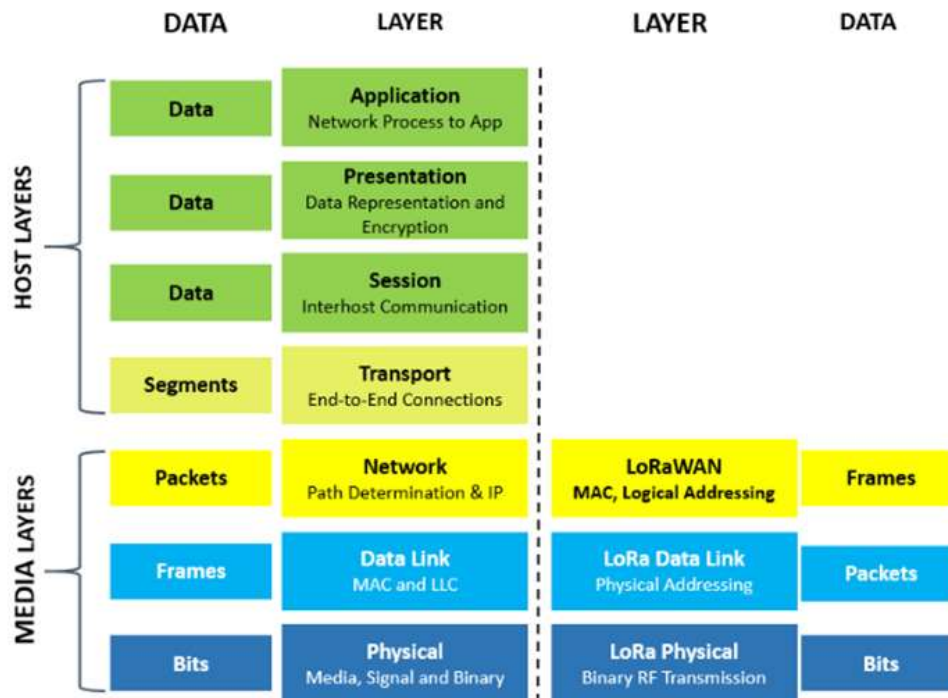
یکی از عوامل دیگری که باعث افزایش برد سیگنال‌ها در این فناوری می‌شود، استفاده از فرکانس‌های زیر ۱ گیگاهرتز یا به اصطلاح Sub Gigahertz می‌باشد. همانطور که می‌دانید، فرکانس‌های رادیویی با افزایش فرکانس قدرت نفوذشان کمتر شده و با کاهش فرکانس قدرت نفوذ بیشتری خواهند داشت. در نتیجه استفاده از فرکانس‌های زیر ۱ گیگاهرتز می‌تواند برد بالایی را برای این فناوری به ارمغان بیاورد [۲].

LoRa به صورت محض یک پیاده سازی فیزیکی (PHY) است، همانطور که در مدل شبکه هفت لایه OSI تعریف شده است، که در شکل ۸ نشان داده شده است. در این فناوری به جای کابل کشی، هوا به عنوان رسانه‌ای برای انتقال امواج رادیویی LoRa از فرستنده فرکانس رادیویی در دستگاه انتهایی به گیرنده فرکانس رادیویی در دروازه و بالعکس می‌باشد.

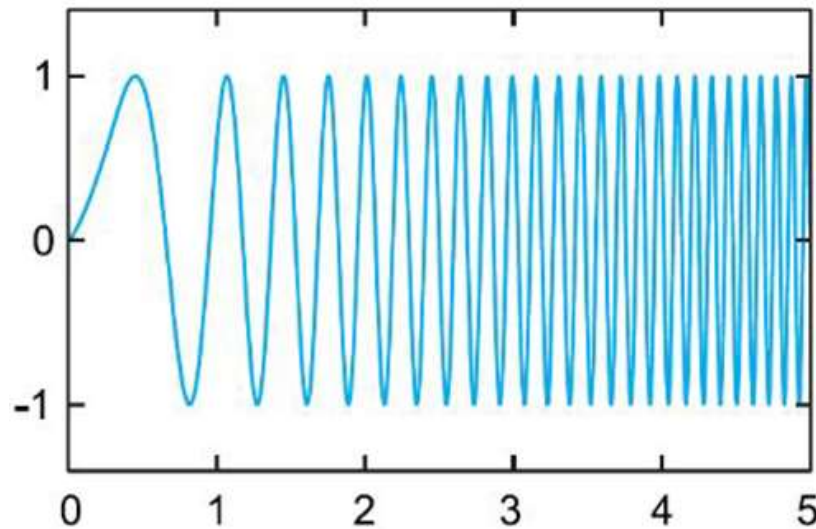
در این فناوری رابطه یک به یکی بین دروازه‌ها و دستگاه‌های انتهایی وجود ندارد، یا به عبارت دیگر اتصالی بین آن‌ها برقرار نمی‌شود. در عوض بسته‌ای که توسط دستگاه انتهایی ارسال می‌شود در هر دروازه‌ای که در برد آن قرار دارد و سالم به آن می‌رسد، دریافت می‌شود و دروازه نیز آن را برای سرور شبکه ارسال می‌کند. تکرار بسته‌ها و حذف بسته‌های اضافه و تکراری در سرور شبکه رسیدگی می‌شود.

^۱ Adaptive Data Rate (ADR)

در مدلاسیون LoRa، گسترش طیف سیگنال از طریق تولید سیگنال چیرپ (Chirp) که به صورت پیوسته فرکانسش تغییر می‌کند به دست می‌آید. طیف گسترده چیرپ LoRa در شکل ۹ قابل مشاهده می‌باشد.



شکل ۸ لایه بندی شبکه LoRaWAN برحسب مدل ۷ لایه‌ای OSI



شکل ۹ مدلاسیون LoRa با استفاده از CSS [۱۳]

۲-۴-۱- ویژگی‌های کلیدی مدلاسیون LoRa

همانطور که در بالا گفته شد، بهره پردازشی LoRa در کانال فرکانس رادیویی توسط ضرب سیگنال داده در کد گسترش یا سری چیرپ معرفی می‌شود. با افزایش نرخ چیرپ، در حقیقت مؤلفه‌های فرکانسی طیف کلی سیگنال افزایش می‌یابد. به عبارت دیگر، انرژی کل سیگنال روی طیف وسیع‌تری از فرکانس‌ها گسترش می‌یابد که این امر به گیرنده امکان تشخیص سیگنال با نسبت سیگنال به نویز^۱ پایین‌تر (که بدتر است) را می‌دهد [۹].

در فناوری LoRa، به مقدار کد گسترشی که به سیگنال اصلی اعمال می‌شود، عامل گسترش گفته می‌شود (SF). در مدلاسیون LoRa از ۶ عامل گسترش مختلف (SF7 تا SF12) استفاده می‌شود. هر چه اندازه عامل گسترش بیشتر باشد، میزان مسافتی که سیگنال می‌تواند طی کند و بدون خطا در گیرنده دریافت شود افزایش می‌یابد [۹].

در جدول ۱ چهار عامل گسترش مختلف (SF7 تا SF10) برای جریان داده به سمت بالا^۲ روی کانال ۱۲۵ کیلوهرتز استفاده می‌شود را مشاهده می‌کنید. در این جدول نرخ بیت و برد تقریبی (برد بستگی به فضای پیاده‌سازی دارد

^۱ Signal to Noise Ration (SNR)

^۲ Uplink Data Stream

و در محیط های روستایی نسبت به محیط شهری برد بیشتری خواهیم داشت) را نشان می دهد. همچنین زمان روی هوا^۱ برای عامل گسترش های مختلف در این جدول قابل مشاهده است. قابل ذکر است که جریان داده به سمت پایین^۲ می تواند از هر ۶ عامل گسترش (SF7 تا SF12) بر روی کانال ۵۰۰ کیلوهرتز استفاده کند.

یک موضوع مهم که مزیت فناوری LoRa محسوب می شود این است که عامل های گسترش بر هم متعامد هستند. این به این معناست که سیگنال هایی که با عوامل گسترش مختلفی مازوله شده اند و روی یک کانال فرکانسی و در یک زمان ارسال می شوند، با هم تداخل نداشته و برای یکدیگر مانند نویز عمل می کنند.

جدول ۱ تاثیر عامل گسترش بر روی مشخصات شبکه [۹]

| زمان روی هوا (ms) | برد (km) | نرخ بیت (bps) | عامل گسترش |
|-------------------|----------|---------------|------------|
| ۳۷۱ | ۸ | ۹۸۰ | SF10 |
| ۱۸۵ | ۶ | ۱۷۶۰ | SF9 |
| ۱۰۳ | ۴ | ۳۱۲۵ | SF8 |
| ۶۱ | ۲ | ۵۴۷۰ | SF7 |

^۱ Time On Air (TOA)^۲ Downlink Data Stream

سیگنال‌های LoRa بسیار پایدار و مقاوم هستند و در برابر تداخل داخل-باند^۱ و خارج-باند^۲ مقاومت می‌کنند. مدلاسیون LoRa همچنین در برابر اثر چند-مسیری^۳ و محو شدگی^۴ (به علت نفوذ سیگنال بالا) در امان هست، که این موضوع این فناوری را برای استفاده در محیط‌های شهری و حومه شهری مناسب می‌سازد.

همانطور که پیشتر گفته شد، فناوری LoRaWAN از باندهای فرکانسی بدون مجوز و زیر ۱ گیگاهرتز استفاده می‌کند. استفاده از این فرکانس‌ها باعث می‌شود که برد سیگنال‌های فناوری افزایش یابد. فرکانس‌های مورد استفاده در مناطق مختلف فرق دارد، به عنوان مثال در ایران و آمریکای شمالی از فرکانس ۹۱۵ مگاهرتز، در اروپا از فرکانس ۸۶۸ مگاهرتز و همچنین ۴۳۳ مگاهرتز بنا به موقعیت جغرافیایی استفاده می‌شود [۸].

با این وجود اشکال استفاده از فرکانس‌های بدون مجوز این است که باند فرکانسی را باید با رعایت مقررات تعیین شده توسط سازمان قانون گذاری باندهای فرکانسی کشور استفاده کرد. این مقررات استفاده از این باندها را محدود می‌کند و نمی‌توان به صورت نامحدود و دلخواه از این کانال‌ها استفاده کرد. دلیل این امر نیز رعایت مساوات بین استفاده کنندگان از این باندهای فرکانسی بدون مجوز می‌باشد. برای این منظور از اصطلاحی به نام چرخه وظیفه^۵ استفاده می‌شود که نشان دهنده درصد مجاز استفاده از کانال در روز است. به عنوان مثال چرخه وظیفه معادل ۱ درصد به معنی حق استفاده به اندازه ۱ درصد از طول شبانه‌روز می‌باشد. مشخصه‌های مدلاسیون LoRa برای هر منطقه در سند پارامترهای منطقه‌ای LoRaWAN موجود در سایت LoRa Alliance قابل مشاهده می‌باشد [۸].

^۱ In-band

^۲ Out-of-band

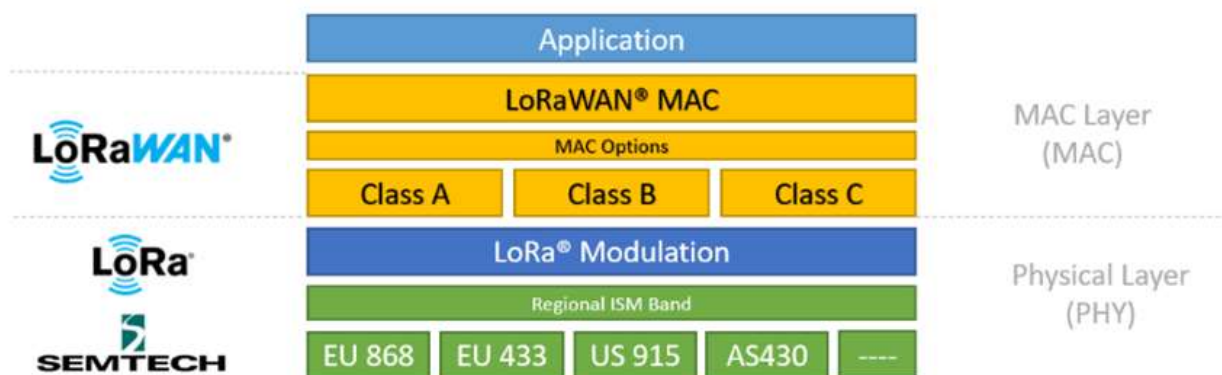
^۳ Multi-path

^۴ Fading

^۵ Duty cycle

۲-۴-۳- بخش LoRaWAN

یک پروتکل کنترل دسترسی به رسانه (MAC)^۱ است که توسط انجمن LoRa تعریف شده است و می‌تواند هزاران دستگاه انتهایی را با توان مصرفی کم و محدوده وسیع پوشش دهد و در لایه بالای مدولاسیون LoRa پیاده‌سازی شده است. برای درک بهتر شبکه LoRaWAN به بررسی پشته^۲ پروتکل این فناوری می‌پردازیم. همانطور که در شکل ۱۰ مشهود است، LoRa فناوری مدولاسیون لایه فیزیکی این پشته می‌باشد که وظیفه آن مدولاسیون بیسیم داده‌ها در لایه فیزیکی و ارسال آن‌ها توسط موج‌های رادیویی بر روی کانال ارتباطی (هوا) است. از طرفی LoRaWAN یک پروتکل متن‌باز شبکه است که بر روی مدولاسیون LoRa سوار شده و یک ارتباط دوطرفه را برای دستگاه‌های انتهایی و دروازه‌ها فراهم می‌کند.



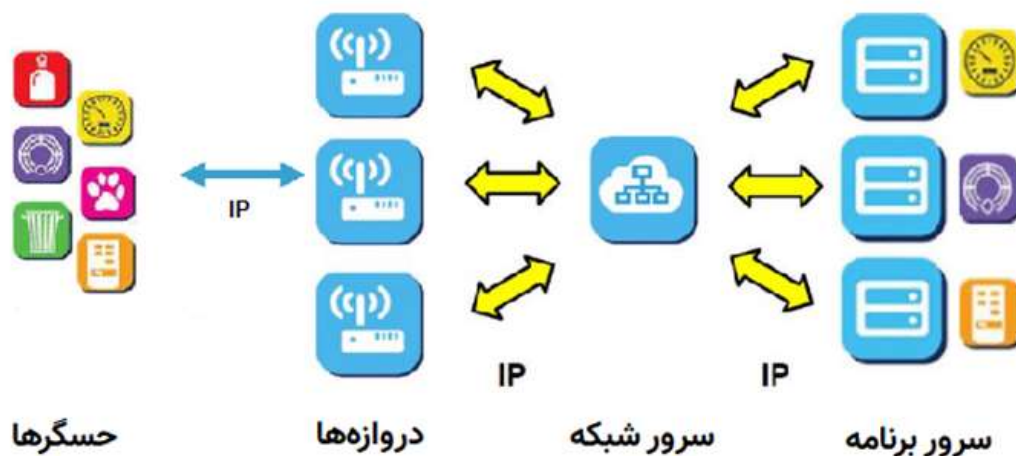
شکل ۱۰ پشته پروتکل LoRaWAN [۱۲]

همانطور که گفته شد، شبکه LoRaWAN براساس توپولوژی ستاره پیاده‌سازی شده است. این امر باعث کاهش انرژی مصرفی و افزایش طول عمر باتری در دستگاه‌های انتهایی می‌شود. در توپولوژی ستاره هر گره به صورت مستقیم با دروازه‌ها در ارتباط است و دیگر نیازی به مسیریابی و ارسال بسته به سایر گره‌های شبکه نیست و در نتیجه توان مصرفی در این شبکه بسیار کم است. همانطور که در شکل ۱۱ مشاهده

^۱ Medium Access Control

^۲ Stack

می‌شود، این شبکه شامل چهار عنصر پایه‌ای دستگاه انتهایی یا گره LoRa، دروازه، سرور شبکه و سرور برنامه است که در ادامه به بررسی هر بخش می‌پردازیم.



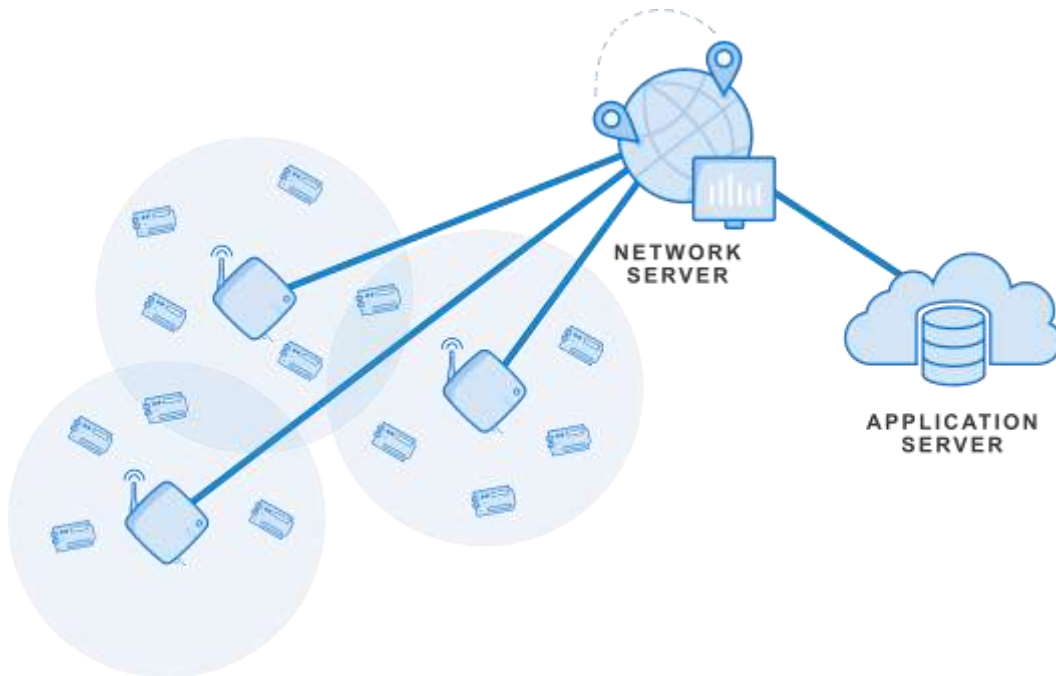
شکل ۱۱ معماری شبکه LoRaWAN [۱۰]

دستگاه انتهایی یا گره LoRa: این بخش شامل حسگرها یا عملگرها است. هر دستگاه انتهایی ابتدا در سیستم ثبت و فعال می‌شود و پس از آن با توجه به نیاز برنامه اطلاعات را از محیط پیرامون دریافت کرده و برای بخش دروازه ارسال می‌کند. پس از ارسال اطلاعات دستگاه انتهایی به حالت خواب رفته و تا ارسال بعدی در این حالت به سر می‌برد و به همین دلیل باتری آن تا سال‌ها می‌تواند به کار خود ادامه دهد. از جمله این دستگاه‌ها می‌توان به حسگر دود، دما و یا برنامه‌هایی از آن‌ها اشاره کرد.

دروازه: در این بخش اطلاعات دریافت شده از دستگاه‌های انتهایی برای سرور شبکه ارسال می‌شود و در حقیقت ارتباط مابین دستگاه انتهایی و سرور شبکه از طریق دروازه‌ها صورت می‌گیرد. یک دروازه شبکه LoRaWAN در واقع یک کامپیوتر کوچک^۱ است که مجهز به ماژول رادیویی LoRa (تک کاناله یا چند کاناله) می‌باشد. این دروازه پیام‌های رادیویی دستگاه‌های انتهایی را که با مدلاسیون LoRa ماژوله شده‌اند، در صورتی که در برد مناسب باشند و سیگنال به درستی به دروازه برسد، دریافت کرده و به سمت سرور شبکه LoRaWAN ارسال می‌کند. ارتباط دروازه با سرور شبکه از طریق اینترنت و شبکه قابل اطمینان IP برقرار می‌شود.

^۱ Mini Coputer

همانطور که شکل ۱۲ نشان می‌دهد، در شبکه LoRaWAN ارتباط پایدار و اتصالاتی بین دستگاه انتهایی و دروازه برقرار نمی‌شود، بلکه اطلاعات یک دستگاه انتهایی می‌تواند توسط چند دروازه دریافت شود و به سمت سرور شبکه ارسال شود. در حقیقت یک پیام به سمت بالا که توسط یک دستگاه انتهایی ارسال می‌شود، توسط هر تعداد دروازه‌ای که در برد آن دستگاه انتهایی قرار داشته باشند دریافت می‌شود.



شکل ۱۲ دریافت داده‌های ارسال شده‌ی دستگاه‌های انتهایی توسط چند دروازه [۱۳]

این موضوع باعث می‌شود که میزان خطا در دریافت پیام‌ها در سرور شبکه به طور چشمگیری کاهش پیدا کند، چرا که شانس دریافت پیام دستگاه انتهایی توسط حداقل یک دروازه بسیار بالا می‌باشد. این قابلیت همچنین باعث کاهش چشمگیر مصرف انرژی در دستگاه انتهایی که متحرک می‌باشند می‌شود و اجازه می‌دهد که قابلیت مکان‌یابی از طریق شبکه LoRaWAN فعال شود (در صورتی که دروازه‌ها مجهز به مکان‌یاب باشند).

ارتباط دروازه با سرور شبکه می‌تواند توسط شبکه WiFi، ارتباط شبکه سلولی و یا ارتباط سیمی با کابل اترنت برقرار شود. دروازه LoRaWAN به صورت کامل در لایه فیزیکی عمل می‌کند و درواقع نقش آن‌ها تنها ارسال پیام‌ها به سمت سرور شبکه می‌باشد. در دروازه تنها صحت پیام رادیویی دریافت شده بررسی می‌شود و در صورت درست بودن پیام به سمت سرور شبکه فرستاده می‌شود، در غیر این صورت دروازه این پیام را دور می‌اندازد.

علاوه بر پیام اصلی، تعدادی اطلاعات اضافی مانند قدرت سیگنال دریافت شده^۱ (RSSI) و نسبت سیگنال به نویز (SNR) و همچنین برچسب زمانی توسط دروازه به سرور شبکه ارسال می‌شود. برای پیام‌های به سمت پایین در LoRaWAN (از سمت سرور شبکه به سمت دستگاه انتهایی)، دروازه درخواست ارسال داده را بدون تفسیر متن پیام اصلی اجرا می‌کند. از آنجایی که چند دروازه می‌توانند پیام یک دستگاه انتهایی را برای سرور شبکه ارسال کنند، تکراری بودن پیام و دور انداختن پیام‌های تکراری در سرور شبکه رسیدگی می‌شود. همچنین برای ارسال پیام به سمت پایین، سرور شبکه دروازه‌ای را که پیام به سمت بالا را با مشخصات بهتری ارسال کرده RSSI (کمتری داشته) انتخاب می‌کند، زیرا این دروازه احتمالاً به دستگاه انتهایی نزدیک تر است و پیام را بهتر به آن می‌رساند.

LoRa پیاده‌سازی دروازه‌ها با هزینه کم و با قابلیت مقیاس‌پذیری بالا را میسر کرده است، برای مثال در آمریکای شمالی دروازه‌های ۸ کاناله ۱۶، کاناله و ۶۴ کاناله در دسترس هستند و قابل پیاده‌سازی می‌باشند. مشخصاً دروازه ۸ کاناله کمترین هزینه را برای پیاده‌سازی خواهد داشت. نوع دروازه انتخابی بستگی به کاربردی دارد که قصد پیاده‌سازی آن را داریم. دروازه‌های ۸ کانال و ۱۶ کاناله را می‌توان هم برای محیط داخلی و هم محیط خارجی استفاده کرد، اما دروازه ۶۴ کاناله تنها برای محیط خارجی استفاده می‌شوند و باید در مکان‌هایی مثل برج‌های مخابراتی، بالای سقف ساختمان‌های بلند و مانند این‌ها پیاده‌سازی شوند.

سرور شبکه: در این بخش داده‌های دریافت شده از طریق دروازه‌ها دریافت می‌شود و یکی از وظایف اصلی سرور شبکه انتخاب بهترین دروازه برای انتقال پیام به دستگاه انتهایی و حذف بسته‌های تکراری دریافتی از دروازه‌ها است. ارتباط بین سرور شبکه و درگاه با استفاده از مسیریاب‌ها^۲ میسر می‌شود. داده‌های دریافت شده از دروازه‌های متفاوت از نظر امنیت، مقدار تطبیقی نرخ داده (ADR) و مواردی مانند آن بررسی می‌شود. سرور شبکه بخشی است که وظیفه دارد داده‌های دریافت شده را بررسی کند و به سرور برنامه ارسال نماید.

به طور کلی، تمام سرورهای شبکه LoRaWAN موارد ذیل را انجام می‌دهند [۱۵]:

^۱ Received Signal Strength

^۲ Routers

- بررسی آدرس دستگاه‌های انتهایی برای احراز هویت آن‌ها.
- احراز هویت فریم‌ها و مدیریت شمارش فریم‌ها.
- تصدیق^۲ پیام‌های دریافت شده.
- تغییر و تطبیق نرخ داده با استفاده از پروتکل نرخ داده تطبیقی.
- پاسخ به تمام درخواست‌های لایه MAC که از دستگاه‌های انتهایی می‌آید.
- ارسال پیام‌های به سمت بالا برای سرور کاربرد مناسب.
- ارسال پیام‌های به سمت پایین که از سرور کاربرد برای دستگاه‌های انتهایی ارسال می‌شود.
- رد و بدل کردن پیام‌های درخواست-پیوست^۳ و پذیرش-پیوست^۴ میان دستگاه‌های انتهایی و سرور-پیوست^۵.

سرور برنامه: در این بخش داده‌های ارسال شده از سرور شبکه جمع‌آوری شده و برای کاربردی خاص مدیریت و استفاده می‌شوند و همچنین در سرور برنامه پارامترهای انتقال و انرژی مصرفی دستگاه‌های انتهایی مدیریت می‌شود [۱۶].

۲-۴-۳-۱- کلاس‌های LoRaWAN

LoRaWAN دارای سه کلاس مختلف A، B و C از دستگاه‌های پایانی^۶، برای پاس-خگویی به نیازهای متفاوت است که در ادامه به بررسی آن‌ها می‌پردازیم.

^۱ Frame

^۲ Acknowledgement

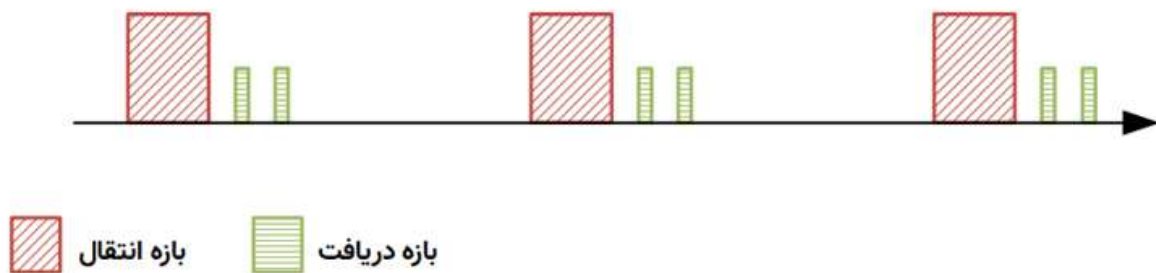
^۳ Join-request

^۴ Join-accept

^۵ Join-server

^۶ End Device

کلاس A: این کلاس در تمام دستگاه‌های پایانی LoRaWAN پشتیبانی می‌شود. ارتباط کلاس A همواره توسط دستگاه پایانی آغاز می‌شود و کاملاً ناهمگام است. همانطور که در شکل ۱۱ نمایش داده شده‌است، هرگونه جریان داده به سمت بالا می‌تواند در هر زمانی ارسال شود و همچنین توسط دو انتقال به سمت پایین کوتاه دنبال می‌شود و در نتیجه در این کلاس بیشترین تاخیر و کمترین مصرف انرژی را شاهد هستیم. برای دسترسی به کانال از پروتکل ALOHA استفاده می‌شود. این کلاس مناسب کاربردهایی است که نرخ اطلاعات کمی در downlink نیاز دارند و همچنین دستگاه‌های پایانی این قابلیت را دارند که وارد حالت خواب شوند که در این حالت با کم شدن قدرت مصرف دستگاه به حداقل می‌رسد و به همین دلیل باتری آن‌ها می‌تواند تا چندین سال دوام داشته‌باشد.



شکل ۱۱ کلاس نوع A [۱۱]

کلاس B: استفاده از این کلاس در شبکه LoRaWAN اختیاری بوده و با استفاده از موج‌های دوره‌ای همگام با شبکه می‌شوند. این کلاس مناسب برای کاربردهایی است که ضمن حفظ انرژی، نیاز به تبادل اطلاعات با سرور دارند و همانطور که در شکل ۱۲ نمایش داده شده‌است، در زمان‌های مشخص بازه انتقال به سمت پایین باز می‌شود و در این کلاس، سرور شروع‌کننده ارتباط به سمت پایین است و در صورتی که داده‌ای برای ارسال داشته‌باشد ابتدا به دستگاه اطلاع داده و بعد از آن پنجره‌ای برای دریافت داده باز می‌شود. مصرف انرژی در این کلاس از کلاس A بیشتر است ولی با این حال مصرف باتری به اندازه کافی کم است.



شکل ۱۲ کلاس نوع B [۱۱]

کلاس C: همانطور که در شکل ۱۳ نمایش داده شده است، در این کلاس دستگاه انتهایی همواره آماده برای دریافت اطلاعات از سرور می باشد و به همین علت مصرف انرژی در آن بسیار بالا است. این کلاس امکان برقراری ارتباط دو طرفه را دارد و همیشه در حال گوش کردن به سرور است و به همین دلیل کمترین تاخیر را دارد. در این کلاس تنها زمانی نمی توان اطلاعات را دریافت کرد که در حال ارسال اطلاعات باشد. پیاده سازی آن مانند کلاس A است با این تفاوت که در آن پنجره اول و دوم دائماً برای دریافت اطلاعات باز می شوند.



شکل ۱۳ کلاس نوع C [۱۱]

۲-۴-۴- امنیت شبکه LoRaWAN

با وجود آن که بیشتر تکنولوژی ها دارای تنها یک لایه امنیتی هستند، در LoRaWAN دو لایه امنیتی وجود دارد و این دو لایه شامل امنیت سرور شبکه و امنیت سرور برنامه است. لایه امنیت سرور شبکه برای احراز هویت گره در شبکه استفاده می شود و وظیفه لایه امنیت سرور برنامه محافظت از داده های برنامه کاربر است.

تکنولوژی LoRa از دو کلید NwkSKey^۱ و AppSKey^۲ برای امنیت و احراز هویت استفاده می‌کند. برای برقراری ارتباط با دستگاه انتهایی در شبکه نیاز به فعال‌سازی و احراز هویت دستگاه است و این تکنولوژی دو روش برای فعال‌سازی و احراز هویت دارد که شامل فعال‌سازی بر بستر هوا^۳ و فعال‌سازی شخصی^۴ می‌باشد. در جدول ۲ مشخصات و ویژگی‌های مختلف این دو روش ذکر و مقایسه شده است.

جدول ۲ مقایسه دو روش راه‌اندازی دستگاه‌های انتهایی

| Over-the-Air-Activation (OTAA) | Activation by Personalization (ABP) |
|--|--|
| روشن شدن دستگاه به صورت خودکار پارامترهای ضروری را تولید می‌کنند | روش ساده شده و در عین حال با امنیت کمتر برای راه‌اندازی دستگاه‌های انتهایی |
| سریال شناسایی و کلیدها در زمان ساخت دستگاه روی آن تنظیم می‌شوند | کلیدهای رمزگذاری به طور منظم می‌توانند دوباره ساخته شوند |
| دستگاه به محض روشن شدن فعالیت خودش را آغاز می‌کند، چرا که رویه پیوست حذف می‌شود | دستگاه انتهایی می‌تواند چندین هویت مختلف را در خود ذخیره کند و به صورت پویا و امن سرور شبکه خود را در طول عمر خود تغییر دهند |
| دستگاه‌های انتهایی به یک سرویس/سرور خاص وصل می‌شوند و نمی‌توانند به صورت پویا سرور خود را تغییر دهند | امنیت سطح بالا و امنیت غیرقابل دستکاری برای دستگاه‌های انتهایی فراهم می‌شود |

^۱ Network Session Key^۲ Application Session Key^۳ Over the Air Activation (OAA)^۴ Activation by Personalization (ABP)

۲-۵- تضعیف سیگنال

سیگنالهای رادیویی بسته به فضای اطرافشان و مسیری که در آن حرکت می‌کنند، برد محدودی دارند. زمانی که سیگنال در هوا یا در کابل مسی یا در فیبر نوری حرکت خود را آغاز می‌کند، از عوامل درونی و بیرونی مختلفی تاثیر می‌پذیرد. برای نمونه، در شبکه‌های کامپیوتری کابلی، مقاومت سیم‌های مسی بخشی از انرژی سیگنال را به‌صورت گرما هدر می‌دهد. هرچه طول کابل و میزان مقاومت سیم بیشتر باشد، و هر چه سیگنال مسافت بیشتری بپیماید، انرژی بیشتری از دست می‌دهد و ضعیف‌تر و حتی خوانش‌ناپذیر می‌شود، یعنی حتی اگر دستگاه گیرنده سیگنال، آن را دریافت کند، نمی‌تواند آن را بخواند.

سیگنال‌های فرستنده‌ها و گیرنده‌های بی‌سیم نیز از عارضه تضعیف مصون نیستند. به همین سبب اگر یک دستگاه مجهز به فرستنده و گیرنده بی‌سیم بیش از حد از روتر یا نقطه‌ی دسترسی^۱ دور شود ممکن است یا سیگنال ضعیفی دریافت کند و یا اصلاً سیگنالی دریافت نشود، زیرا سیگنال‌ها در شبکه‌های بی‌سیم برد محدودی دارند. تضعیف سیگنال برحسب دسیبل^۲ (dB) سنجیده می‌شود. دسیبل استاندارد است که در سامانه‌های ارتباطی برای سنجش نرخ ولتاژ و توان الکتریکی یا دیگر کمیت‌های مرتبط با سیگنال به کار می‌رود.

پدیده‌ی تضعیف سیگنال در حالاتی از قبیل تضعیف ناشی از فاصله^۳، تضعیف ناشی از جذب^۴، تضعیف ناشی از انعکاس^۵، تضعیف ناشی از شکست^۶ و تضعیف ناشی از پراکندگی و انکسار^۷ ممکن است رخ دهد و

^۱ Access Point

^۲ Decibel

^۳ Free Space Path Loss

^۴ Absorption

^۵ Reflection

^۶ Refraction

^۷ Scattering And Diffraction

برای رفع نمودن خطای حاصل از این پدیده، روشهای پیش پردازش داده در ادامه توضیحات ارائه خواهند شد.

۲-۵-۱- تضعیف ناشی از فاصله

امواج رادیویی بعثت عواملی طبیعی که از اتمسفر زمین ناشی می شود با طی نمودن مسافت دچار میرایی و تضعیف می گردد. بنابراین یکی از عوامل که در سیگنال دریافتی در شبکه‌های بی‌سیم و در نتیجه سرعت و کارایی آنها نقش دارد فاصله آنها از اکسس پوینت خواهد بود، علت این موضوع این است که با افزایش فاصله، تجهیزات فرستنده و گیرنده جهت تشخیص دادگان از مدولاسیون با پیچیدگی کمتر استفاده کرده و در نتیجه سرعت و کارایی کاهش خواهد یافت.

میزان تضعیف سیگنال ناشی از فاصله بصورت غیر خطی بوده و از طریق معادله ۱ که در عبارت زیر اشاره شده است قابل محاسبه است. در روابط زیر مقدار پارامترهای f و D به ترتیب برابر با فرکانس کاری برحسب مگاهرتز و فاصله‌ی میان آنتن فرستنده و گیرنده برحسب کیلومتر است.

$$\text{FSPL} = 32.44 + (20\log_{10}(f)) + (20\log_{10}(D)) \quad \text{معادله ۱}$$

۲-۵-۲- تضعیف ناشی از جذب

موانع موجود در مسیر سیگنال رادیویی، مقداری از توان آنرا جذب نموده و در نتیجه سیگنال دچار تضعیف خواهد شد. مهمترین عامل تضعیف سیگنال در شبکه‌های بی‌سیم تضعیف ناشی از جذب است و موانع موجود در محیط در طراحی و چگونگی چینش نقاط دسترسی تاثیر به سزایی دارند. در جدول ۳ میزان تضعیف ناشی از جذب در زمانی که اشیاء مختلف در مسیر ارتباط قرار می گیرند را مشاهده می کنید.

جدول ۳ میزان تضعیف ناشی از جذب به وسیله موانع مختلف [۱۷]

| میزان تضعیف سیگنال | شیء موجود در مسیر سیگنال رادیویی |
|--------------------|----------------------------------|
| ۳ dB | دیوار گچی |
| ۶ dB | پنجره‌ی شیشه‌ای با کادر فلزی |
| ۴ dB | دیوار بتنی |

| | |
|----------------------------------|----------|
| پنجره تمام شیشه‌ای | ۳ dB |
| در فلزی | ۶ dB |
| در فلزی میان دیوار آجری | ۱۲ dB |
| یک انسان با گوشی همراه در دستانش | ۳ - ۶ dB |

۲-۵-۳- تضعیف ناشی از انعکاس

موانع صیقلی نظیر اجسام فلزی، شیشه و سطح آب موجب انعکاس سیگنال رادیویی می‌گردند. این انعکاس زمانی نامطلوب خواهد بود که سیگنالهای ارسالی از مسیرهای مختلف و در زمان‌هایی متفاوت به گیرنده رسیده و موجب ایجاد پدیده چند مسیری گردد، در این حالت به علت جمع دامنه چندین سیگنال با فازهای مختلف، سیگنال دریافتی حاصل شده در سمت گیرنده مطلوب نخواهد بود. این مورد بخصوص در انبارها و کارخانه‌ها که دارای موانع فلزی متعدد هستند و همچنین در ساختمانهای دارای پوشش و دیوارهای شیشه‌ای بیشتر بچشم می‌خورد. همانطور که پیش تر گفته شد، این نوع تضعیف بر روی سیگنال‌های فناوری LoRaWAN تاثیری نداشته زیرا که قدرت نفوذ این سیگنال‌ها بسیار زیاد است.

۲-۵-۴- تضعیف ناشی از شکست

شکست نور بعثت تغییر در ضریب شکست محیط که سیگنال رادیویی از آن عبور می‌کند رخ می‌دهد و موجب تغییر در زاویه تابش و یا به عبارتی انعکاس سیگنال می‌شود. از عوامل موثر در شکست سیگنال رادیویی می‌توان رطوبت، وضعیت اتمسفر و میزان فشار هوا را برشمرد. پدیده شکست بخصوص در پیاده‌سازی شبکه‌های بی‌سیم در محیطهای بیرونی از عوامل تضعیف سیگنال به شمار می‌روند.

۲-۵-۵- تضعیف ناشی از پراکندگی و انعکاس

برخورد سه سیگنال رادیویی با ذرات ریز نظیر گرد و غبار و همچنین موانع نوک تیز نظیر برگ درختان موجب پراکندگی^۱ و انتشار آن در تمامی جهات می‌گردد. این پدیده در شبکه‌های بی‌سیم که در محیط‌های غیر مسقف پیاده‌سازی شده‌اند از عوامل تضعیف سیگنال رادیویی است. از سوی دیگر برخورد سیگنال با موانع موجب ایجاد خمیدگی و انعکاس سیگنال شده^۲ که تغییر در فرکانس، فاز و دامنه آن را در پی خواهد داشت. در محیط‌های غیر مسقف در شبکه‌های بی‌سیم پدیده خمیدگی موجب بوجود آمدن نقاط کور در پوشش دهی در نزدیکی موانع می‌شود.

۲-۶-۲- پیش پردازش دادگان

پدیده‌ی تضعیف سیگنال رادیویی در اغلب ارتباطات شبکه‌های بی‌سیم بوجود می‌آید و برای حل این موضوع راهکارهای گوناگونی توسط محققان تا کنون ارائه شده است. یکی از این راهکارها فیلتر کردن نویز قدرت سیگنال دریافتی با به کارگیری روابط ریاضیاتی و احتمالاتی است. بدین منظور در ادامه دو روش مشهور برای فیلتر کردن نویز قدرت سیگنال دریافتی معرفی خواهند شد که در هریک از پارامتر اصلی قدرت سیگنال دریافتی خواهد بود.

۲-۶-۱- فیلتر کالمن گسسته

اکثر سیستم‌های مدرن دارای حسگرهای متعددی هستند که بر اساس یک سری اندازه‌گیری، حالت‌های پنهان (ناشناخته) را تخمین می‌زنند. به عنوان مثال، یک گیرنده GPS تخمین مکان و سرعت را (که متغیرهای پنهان هستند) با استفاده از اختلاف زمان رسیدن سیگنال‌های ماهواره‌ها اندازه‌گیری می‌کند.

¹ Scattering

² Diffraction

یکی از بزرگترین چالش‌های سیستم‌های ردیابی و موقعیت‌یابی، برآورد دقیق حالت‌های پنهان در صورت وجود عدم قطعیت است. در گیرنده‌های GPS، عدم قطعیت اندازه‌گیری به بسیاری از عوامل خارجی مانند نویز حرارتی، اثرات جوی، تغییرات جزئی در موقعیت ماهواره، دقت ساعت گیرنده و بسیاری موارد دیگر بستگی دارد. در این پروژه عدم قطعیت‌هایی چون، نویز حرارتی، اثرات جوی و ... برای گیرنده‌های LoRaWAN نیز وجود دارد.

فیلتر کالمن گسسته^۱ یکی از مهم‌ترین و رایج‌ترین الگوریتم‌های تخمین است. این الگوریتم حالت یک سیستم پویا را با استفاده از مجموعه‌ای از اندازه‌گیری‌های شامل خطا در طول زمان برآورد می‌کند. همچنین این الگوریتم از تخمین حالت قبل و مشاهدده فعلی برای محاسبه تخمین حالت فعلی استفاده می‌کند و یک ابزار بسیار قوی برای ترکیب اطلاعات در حضور عدم قطعیت است. این فیلتر از نام رودولفای کالمن، یکی از پایه‌گذاران این تئوری گرفته شده است. امروزه فیلتر کالمن در ردیابی هدف با استفاده از رادار، سیستم‌های مکان‌یابی و ناوبری، سیستم‌های کنترل، گرافیک کامپیوتری و موارد دیگر استفاده می‌شود.

هدف این الگوریتم حداقل نمودن میزان نویز فرآیند است که بدین منظور در ۴ گام کار را مطابق شکل زیر انجام می‌دهد. در گام صفر، که تنها یک بار و در ابتدای فرآیند رخ می‌دهد، مقدار دهی‌های اولیه انجام می‌شوند. خروجی این مرحله حالت اولیه سیستم^۲ $(x^0, 0)$ و مقدار عدم قطعیت اولیه است^۳ $(p_0, 0)$. در گام بعدی اندازه‌گیری‌ها انجام می‌شود. خروجی این گام، وضعیت اندازه‌گیری شده سیستم^۴ (z_n) و عدم قطعیت اندازه‌گیری^۵ (r_n) است. مقدار n نمایانگر دفعه n ام انجام این کار است. همچنین عدم قطعیت اندازه‌گیری تفاوت مقداری، بین اندازه‌گیری‌ها و مقادیر واقعی است. مقادیر اندازه‌گیری شده یا z_n در این پروژه، قدرت سیگنال‌های دریافتی هستند.

¹ Discrete Kalman

² Initial System State

³ Initial State Uncertainty

⁴ Measured System State

⁵ Measurement Uncertainty

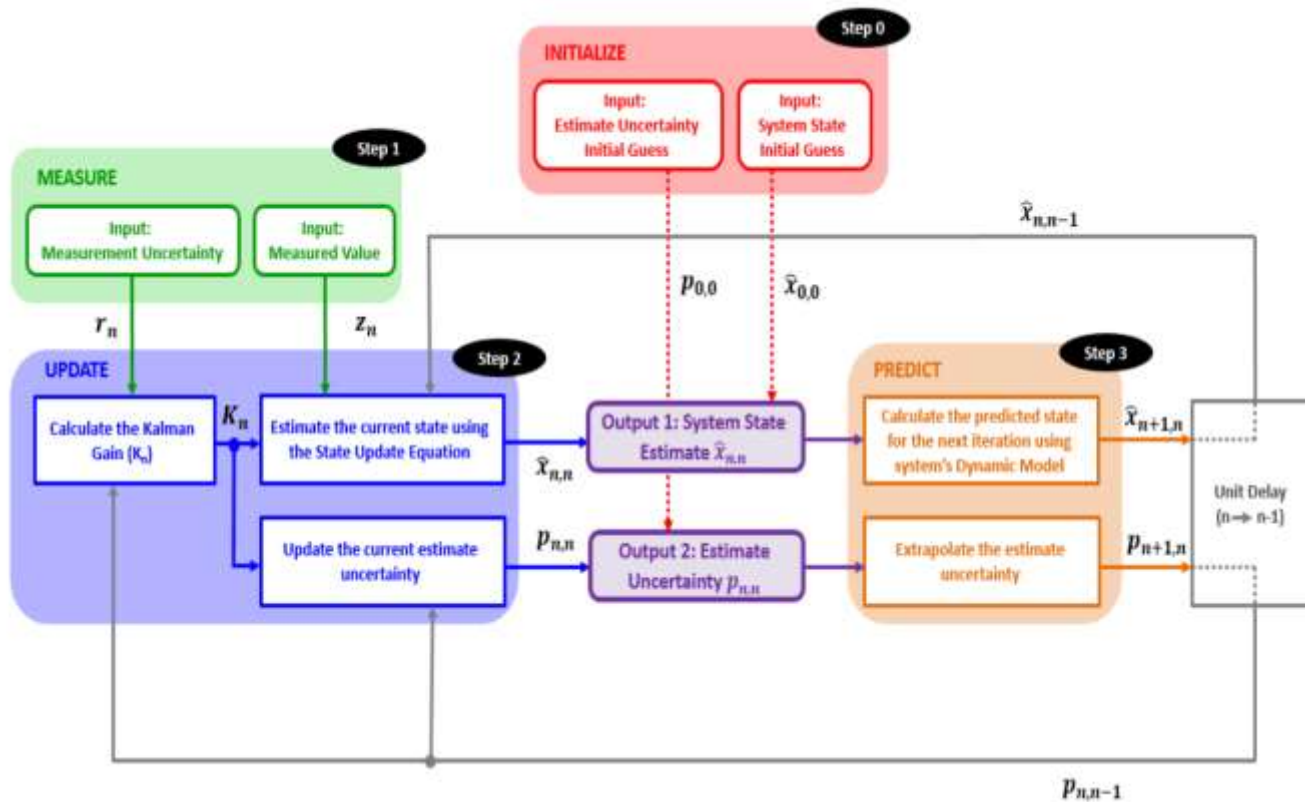
گام دوم فرآیند به روز رسانی حالت و مسئول برآورد وضعیت فعلی سیستم است. ورودی‌های این گام مقادیر x_n ، z_n ، تخمین وضعیت سیستم پیش‌بینی شده قبلی^۱ ($\hat{x}_{n,n-1}$) و عدم قطعیت تخمین وضعیت سیستم پیش‌بینی شده قبلی^۲ ($p_{n,n-1}$) است. بر اساس ورودی‌ها، فرآیند به روز رسانی حالت، بهره^۳ Kalman را محاسبه می‌کند و دو خروجی ارائه می‌دهد: برآورد وضعیت فعلی سیستم ($\hat{x}_{n,n}$) و عدم قطعیت برآورد وضعیت فعلی ($p_{n,n}$). این پارامترها خروجی‌های فیلتر Kalman هستند.

گام سوم و آخر مرحله پیش‌بینی می‌باشد. فرآیند پیش‌بینی برآورد وضعیت فعلی سیستم و عدم قطعیت آن را بر اساس مدل دینامیکی سیستم استنتاج می‌کند. در اولین تکرار فیلتر ($n = 1$)، مقداردهی اولیه به عنوان تخمین حالت قبلی و عدم قطعیت در نظر گرفته می‌شود. خروجی‌های پیش‌بینی به عنوان تخمین حالت پیش‌بینی شده قبلی و عدم قطعیت در تکرار فیلتر زیر استفاده می‌شود [۱۸]. شمای کلی این روش را در شکل ۱۴ مشاهده می‌کنید.

¹ prior Predicted System State Estimate

² prior Predicted System State Estimate Uncertainty

³ Gain



شکل ۱۴ شماتیک فیلتر kalman

به دلیل ماهیت داده‌های استفاده شده و روش انجام موقعیت یابی به کار رفته در این پروژه، که در فصل‌های آینده توضیح داده خواهد شد، می‌توان از نسخه ساده شده این فیلتر موسوم به α - β - γ استفاده کرد. هدف این الگوریتم به حداقل رساندن نویز وارده بر مقدار قدرت سیگنال دریافتی، در هر دروازه LoRaWAN است. مقادیر مورد استفاده در این فیلتر به صورت زیر هستند.

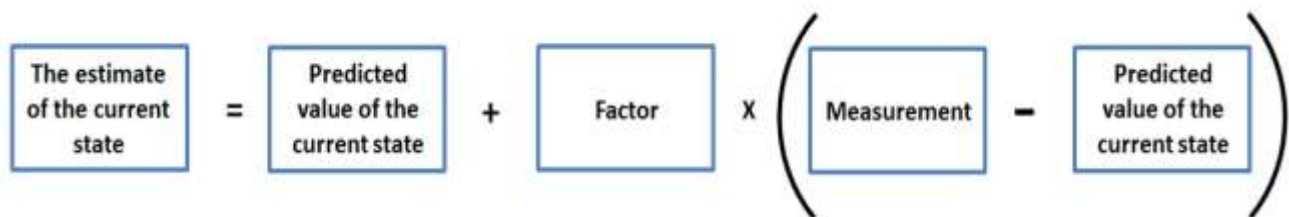
- \mathbf{X} : مقدار واقعی قدرت سیگنال
- \mathbf{Z}_n : مقدار اندازه‌گیری شده قدرت سیگنال در بار n ام
- $\hat{\mathbf{X}}_{n,n}$: تخمین \mathbf{X} در زمان n (تخمین پس از اندازه‌گیری \mathbf{Z}_n انجام می‌شود).
- $\hat{\mathbf{X}}_{n+1,n}$: تخمین وضعیت آینده ($n+1$) از \mathbf{X} . برآورد در زمان n انجام می‌شود. به عبارت دیگر $\hat{\mathbf{X}}_{n+1,n}$ یک حالت پیش بینی شده است.
- $\hat{\mathbf{X}}_{n-1,n-1}$: تخمین \mathbf{X} است در زمان $n-1$ (تخمین پس از اندازه‌گیری \mathbf{Z}_{n-1} انجام می‌شود).

هدف نهایی این فیلتر تصحیح مقدار اندازه‌گیری فعلی یا z_n است که با $\hat{x}_{n,n}$ نمایش می‌دهیم. معادله ۲ نشان دهنده روش کار این الگوریتم است.

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \frac{1}{n} (z_n - \hat{x}_{n,n-1})$$

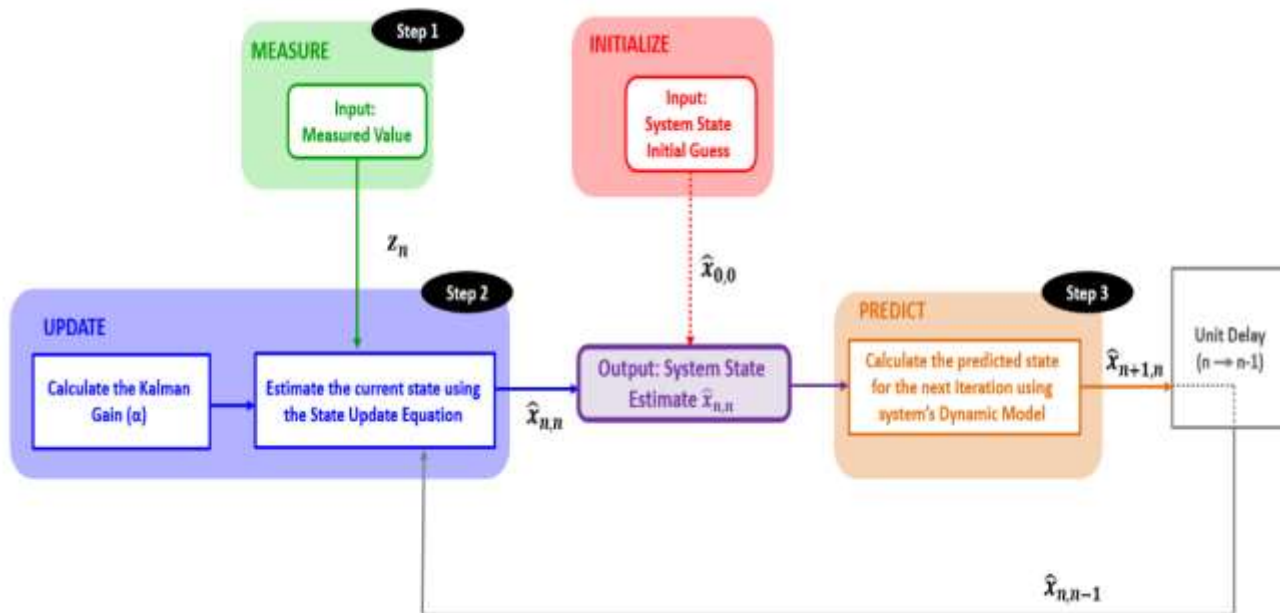
معادله ۲

معادله ۲ را می‌توان به صورت شکل زیر تفسیر نمود که در آن factor همان بهره kalman یا $\frac{1}{n}$ است.



شکل ۱۵ تفسیر معادله ۲

در نهایت شماتیک الگوریتم به کار رفته به صورت زیر است.



شکل ۱۶ الگوریتم پیشبینی و فیلتر مورد استفاده

۲-۶-۲- فیلتر ذره

روش فیلتر ذره^۱ یک روش عددی است که اجازه می‌دهد یک راه حل تقریبی جهت تخمین پی‌درپی پیدا شود و برای مسائل غیرگوسین و غیرخطی می‌تواند بسیار کارا باشد. ایده اولیه فیلتر ذره در سال ۱۹۴۹ توسط آقای متروپلیس مطرح شد که بررسی‌های جزئیات مجموعه‌ای از ذره‌ها را به جای یک ذره مورد مطالعه قرار داد. اما در سال ۱۹۴۵ یکی از اولین ایده‌های اصلی تر و پایه و اساس فیلتر ذره با محاسبه و بکارگیری الگوریتم مونت کارلو توسط آقایان همرسلی و مورتون ارائه گردید.

فیلتر ذره یک مبحث کاملاً آماری است و دلیل مطرح شدن آن این است که بتواند برای مسائلی به کار گرفته شود که تخمین آنها توسط فیلتر کالمن امکان پذیر نباشد. همانند الگوریتم کالمن گسسته، مازول فیلتر ذره نیز سعی بر آن دارد که با به حداقل رساندن نویز میزان پارامتر را تخمین بزند، با این تفاوت که بدون اعمال یک مدل معادله خطی و توزیع طبیعی برای نویز سیگنال این کار را انجام می‌دهد.

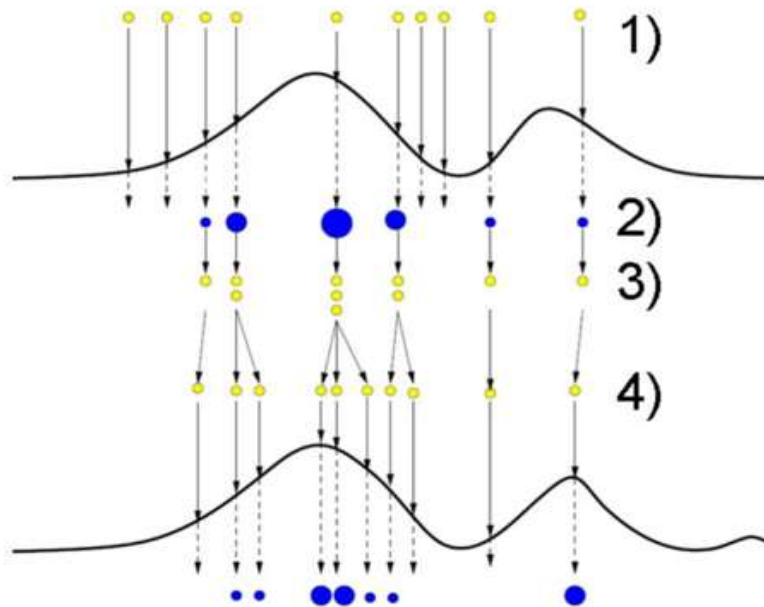
نحوه‌ی عملکرد این الگوریتم به این گونه است که برحسب پارامترهای بدست آمده در گذشته، برای پارامتر کنونی یکس - ری کاندید قرار داده و به هریک وزنی را اختصاص می‌دهیم. سپس در هر مرحله که پارامتر جدید وارد می‌شود می‌توان بین کاندیدها آنهایی که به پیش بینی نزدیک تر بودند وزنشان را برای مراحل بعد افزایش داد.

برای تفهیم بهتر این الگوریتم، مطابق با شکل ۱۴ فرض می‌شود که ۱۰ ذره داریم و می‌خواهیم ۱۰ کاندید برای پارامتر اصلی پیدا کنیم و بدین منظور این الگوریتم را در چهار قدم که انجام می‌دهیم. در قدم اول همان گونه که در شکل نشان داده شده است، ۱۰ تا کاندید که همگی وزن یکسانی دارند را در اختیار داریم.

در قدم دوم با بهره‌گیری از احتمال تخمین حالت (منحنی سیاه) حاصل از پارامترهای جدیدی که بدست آمده اند هریک از وزن‌ها را به میزان لازم تغییر می‌دهیم و در شکل ۱۴ با رنگ آبی نمایش می‌دهیم. در قدم سوم از پارامترهایی که وزن کم دارند صرف نظر می‌کنیم. در قدم چهارم ترتیب و اولویت پارامترهای کاندیدی

¹ Particle Filter

که هنوز وجود دارند را به صورت رندوم جابه جا می کنیم. در شکل ۱۷ شماره‌ی مراحل با اعداد ۱ الی ۴ نمایش داده شده است.



شکل ۱۷ مراحل انجام الگوریتم فیلتر ذره

۲-۷- الگوریتم‌های یادگیری نظارت شده

این روش، یک روش عمومی در یادگیری ماشین است که در آن به یک سیستم، مجموعه‌ای از جفت‌های ورودی و خروجی ارائه شده و سیستم تلاش می‌کند تا تابعی از ورودی به خروجی را فرا گیرد. یادگیری تحت نظارت نیازمند تعدادی داده ورودی به منظور آموزش سیستم است. در ادامه تعدادی از الگوریتم‌های یادگیری نظارت شده که در این پروژه به کار رفته اند معرفی خواهند شد.

۲-۷-۱- الگوریتم K نزدیک‌ترین همسایگی

الگوریتم k نزدیک‌ترین همسایه^۱ در مسائل طبقه‌بندی^۲ و رگرسیون^۳ مورد استفاده قرار می‌گیرد [۲۳]. این الگوریتم از تشابه ویژگی برای پیش‌بینی مقادیر نقاط داده جدید استفاده می‌کند؛ که به این معنی است که به نقطه داده جدید بر اساس میزان مطابقت آن با نقاط مجموعه آموزشی، یک مقدار تخصیص می‌دهد. با کمک مراحل زیر می‌توان نحوه عملکرد آن را درک کرد. برای پیاده‌سازی این الگوریتم ابتدا باید مقدار k را انتخاب کنیم. سپس با کمک یکی از روشهای فاصله اقلیدسی^۴، فاصله همینگ^۵ یا فاصله منهن^۶، فاصله بین داده تست و هریک از داده آموزشی محاسبه می‌شود. متداول‌ترین روش برای محاسبه فاصله، محاسبه فاصله اقلیدسی است. سپس برای هر نقطه‌ای k نزدیک‌ترین نقطه به آن را بدست می‌آورند و برچسبی که میان آن k نقطه متداول تر است به آن نقطه اختصاص داده می‌شود.

۲-۷-۲- بردار پشتیبان

بردار پشتیبان^۷ (SVM) به زبان ساده، مجموعه‌ای از نقاط در فضای n بعدی داده‌ها هستند که مرز دسته‌ها را مشخص می‌کنند و مرزبندی و دسته‌بندی داده‌ها براساس آنها انجام می‌شود و با جابجایی یکی از آنها، خروجی دسته‌بندی ممکن است تغییر کند [۲۴].

برای انجام این کار و ساخت یک دسته‌بند بهینه، محاسبه فاصله مرزهای به دست آمده با بردارهای

^۱ K-nearest neighbor

^۲ Classification

^۳ Regression

^۴ Euclidean distance

^۵ Hamming distance

^۶ Manhattan distance

^۷ Support Vector Machine

پشتیبان هر دسته (مرزی ترین نقاط هر دسته یا کلاس) و در نهایت انتخاب مرزیست که از دسته‌های موجود، مجموعاً بیشترین فاصله را داشته باشد که در این روش خط میانی، تقریب خوبی از این مرز است که از هر دو دسته فاصله‌ی زیادی دارد. این عمل تعیین مرز و انتخاب خط بهینه (در حالت کلی، آبر صفحه مرزی) به راحتی با انجام محاسبات ریاضی نه چندان پیچیده قابل پیاده سازی است.

بدین منظور نیاز است که داده‌ها را به کمک یک تابع هسته^۱ به یک فضای دیگر ببریم و عملایک نگاشت انجام دهیم. که در آن فضا، داده‌ها تفکیک پذیر باشند و بتوان الگوریتم را به سادگی برای آنها به کار برد. تعیین درست‌این تابع نگاشت در عملکرد ماشین بردار پشتیبان موثر است. بدین خاطر چند تابع هسته با نام‌های هسته خطی^۲، هسته چند جمله‌ای^۳ و هسته RBF برای کاربردهای گوناگون موجود است.

۲-۷-۳ XGBoost

روش XGBoost به عنوان یکی از برترین و به‌روزترین روش‌های معرفی شده در مبحث یادگیری ماشین است. این روش به عنوان اصلی ترین و بهترین روش پیشنهادی در یکی از مسابقات برنامه نویسی در سایت Kaggle معرفی شده است.

این الگوریتم یک پیاده سازی مبتنی بر تقویت گرادیان درخت تصمیم است که برای سرعت و کارایی بالا طراحی شده است که در آن محاسبات به صورت توزیع شده و با موازی سازی ساخت درخت صورت می‌گیرد.

این الگوریتم از توابع هدف از پیش تعریف شده برای حل مسائل طبقه بندی، رگرسیون و رتبه بندی، بهره می‌برد. همچنین این الگوریتم از یک تابع هدف برای اندازه‌گیری عملکرد مدل استفاده می‌کند. این مدل مجموعه مشخصی از پارامترها را خروجی می‌دهد. علاوه براین، از معیارهای ارزیابی تعریف شده توسط کاربر نیز پشتیبانی می‌کند.

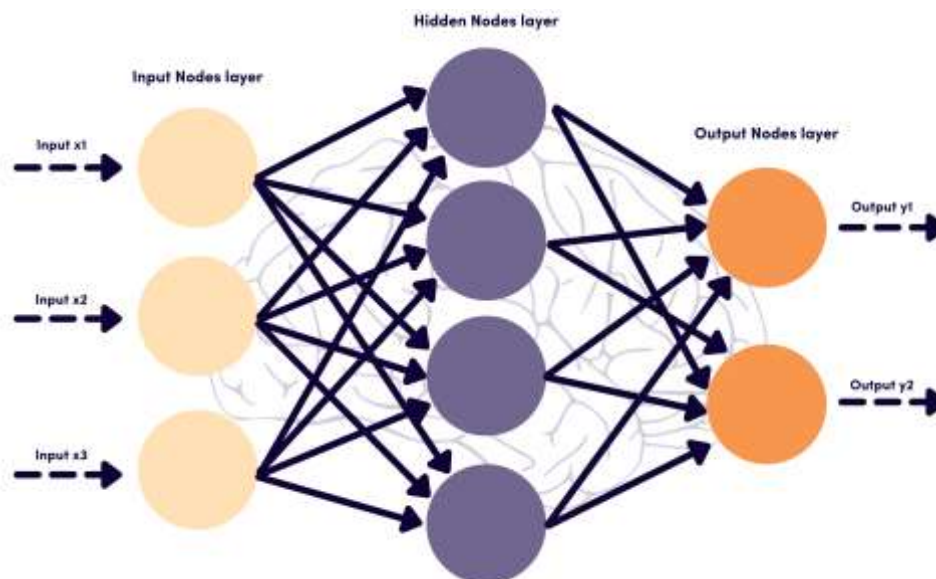
^۱ Kernel

^۲ Linear kernel

^۳ Polynomial Kernel

۲-۷-۴- شبکه عصبی

مغز انسان در خود تعداد بسیار زیادی از نورون‌ها را جای داده است تا اطلاعات مختلف را پردازش کرده و جهان اطراف را بشناسد. به صورت ساده، نورون‌ها در مغز انسان، اطلاعات را از نورون‌های دیگر به وسیله‌ی دندرویدها می‌گیرند. این نورون‌ها اطلاعات ورودی را با هم جمع کرده و اگر از یک حد آستانه‌ای فراتر رود به اصلاح فعال^۱ می‌شود و این سیگنال فعال شده از طریق آکسون‌ها به نورون‌های دیگر متصل می‌شود. البته که این توضیح در علم پزشکی مفید است ولی در علوم کامپیوتر و هوش مصنوعی ما فقط از نورون‌های مغز برای ساخت الگوریتمی به نام شبکه‌ی عصبی مصنوعی^۲ بهره می‌گیریم [۲۵]. توسط الگوریتم شبکه‌های عصبی، می‌توان مدل‌های مختلف و پیچیده‌ای را شناخت. برای مثال می‌توان طبقه‌بندی‌هایی با دقت خوب یا خوشه‌بندی‌هایی بر روی داده‌های بزرگ انجام دهیم. برای مثال در شکل ۱۸ یک معماری ساده شبکه عصبی مصنوعی با ۳ ورودی، یک لایه پنهان با اندازه ۵ و ۲ نورون خروجی مشاهده می‌کنید.



شکل ۱۸ معماری ساده شبکه عصبی مصنوعی

^۱ Fire

^۲ Artificial Neural Network

جمع‌بندی

در این فصل در ابتدا به مفاهیم پایه‌ای همچون معماری اینترنت اشیاء و شبکه‌های LPWAN علی‌الخصوص شبکه مورد نظر ما برای پیاده‌سازی سامانه موقعیت‌یابی یعنی LoRaWAN پرداخته شد. از آنجایی که پیاده‌سازی این سامانه به طور مستقیم به قدرت سیگنال‌های دریافتی در دروازه‌های شبکه LoRaWAN وابسته است و یکی از عوامل موثر بر قدرت سیگنال در هر شبکه‌ای مقادیر نویز محیطی هستند، مفاهیمی همچون عوامل تضعیف سیگنال و الگوریتم‌های فیلترینگ در قالب پیش‌پردازش داده، مطرح شدند. در انتها نیز به روش‌های یادگیری نظارت‌شده به کار رفته در این پروژه اشاره شد. در فصل بعدی پیاده‌سازی زیرساخت این سامانه مورد بحث قرار خواهد گرفت.

فصل ۳

پیاده‌سازی زیرساخت سامانه موقعیت‌یابی

۳-۱- مقدمه

در فصل دوم با فناوری ارتباطی LoRaWAN و جزئیات فنی و اجزاء تشکیل دهنده آن آشنا شدیم و دانستیم که این فناوری ویژگی‌ها و مزیت‌هایی داراست که آن را برای زیرساخت سامانه موقعیت‌یابی مناسب می‌کند. از جمله این ویژگی‌ها این است که این فناوری در باند بدون مجوز فرکانسی فعالیت می‌کند. همچنین به دلیل متن‌باز بودن پروتکل MAC این فناوری و قیمت مناسب لایه فیزیکی آن (LoRa) ایجاد یک زیرساخت شبکه LoRaWAN نسبت به باقی فناوری‌ها ارزان قیمت تر و ساده‌تر است.

لازم به ذکر است که سامانه‌ای موقعیت‌یابی که از روش‌های اثر انگشت مکانی یعنی قدرت سیگنال دریافتی (RSSI) و نسبت سیگنال-نویز (SNR) به عنوان معیاری برای تخمین موقعیت مکانی استفاده می‌کند، دقت کمتری در مقایسه با سیستم موقعیت‌یابی جهانی (GPS) دارد. اما GPS محدودیت‌هایی دارد که در شبکه LoRaWAN نقطه قوت محسوب می‌شوند. برای مثال از GPS در محیط‌های مسقف نمی‌توان بهره برد ولی به دلیل نفوذ سیگنال زیاد LoRaWAN از آن می‌توان در عمق ساختمان‌ها نیز استفاده کرد. همچنین مصرف توان GPS بسیار زیاد است در حالی که دستگاه‌های انتهایی LoRaWAN تا سال‌ها نیاز به شارژ شدن نخواهند داشت. به همین دلیل موقعیت‌یابی با استفاده از روش‌های اثر انگشت مکانی بر ستر شبکه LoRaWAN می‌تواند برای کاربردهایی به کار رود که برای سیستم‌های دارای GPS مناسب نیست. در کل، به دلیل مزیت‌هایی که شبکه LoRaWAN داراست در این پروژه از این فناوری به عنوان زیرساخت سامانه موقعیت‌یابی استفاده شده است.

در این فصل به ساخت، ایجاد و به‌کارگیری اجزاء مختلف این شبکه می‌پردازیم. همانطور که در بخش ۲-۲-۱ گفته شد، معماری مورد استفاده ما در این پروژه مطابق شکل ۲ خواهد بود. در این فصل از پایین‌ترین لایه این معماری (لایه درک) شروع نموده و مراحل ساخت و ایجاد یک دستگاه انتهایی LoRaWAN شرح داده می‌شود. سپس به لایه بعدی (لایه شبکه) رفته و مراحل ایجاد و به‌کارگیری دروازه گفته می‌شود. در این پروژه از پلتفرم TTN^۱ که سرور شبکه و سرور کاربرد را به صورت رایگان در اختیار قرار می‌دهد استفاده می‌شود.

^۱ TheThingsNetwork

۳-۲- دستگاه انتهایی شبکه LoRaWAN

هر دستگاه انتهایی در شبکه LoRaWAN دارای چند چهار جزء اصلی می‌باشد:

۱. پردازنده
۲. ماژول رادیویی LoRa
۳. منبع تغذیه
۴. حسگر^۱ یا عملگر یا هردوی آن‌ها

از آنجایی که هدف این پروژه ایجاد سامانه موقعیت‌یابی بر بستر شبکه LoRaWAN می‌باشد و پیاده‌سازی یک کاربرد خاص مدنظر نیست، در این پروژه نیازی به استفاده از حسگر و عملگر نداریم چراکه محتوای پیام ارسال شده از دستگاه انتهایی اهمیت ندارد و تنها ویژگی‌های سیگنال ارسالی و دریافتی حائز اهمیت است. همچنین به دلیل این که تنها مشخصات سیگنال دریافت شده در سرور شبکه مدنظر است و نیازی به ارسال داده به سمت پایین برای دستگاه انتهایی نیست کلاس دستگاه‌های انتهایی، کلاس A انتخاب شده است.

در بخش‌های پیش‌رو درمورد سه جزء مورد نیاز برای ساخت یک دستگاه انتهایی مبتنی بر LoRa (پردازنده، ماژول رادیویی و منبع تغذیه) متناسب برای این پروژه، دلیل انتخاب هر کدام، نحوه برنامه‌نویسی و کتابخانه مورد استفاده و همچنین نحوه اتصال تمام این بخش‌ها به هم به طوری که یک دستگاه انتهایی کاربردی^۲ داشته باشیم شرح داده شده است.

۳-۲-۱- پردازنده دستگاه انتهایی

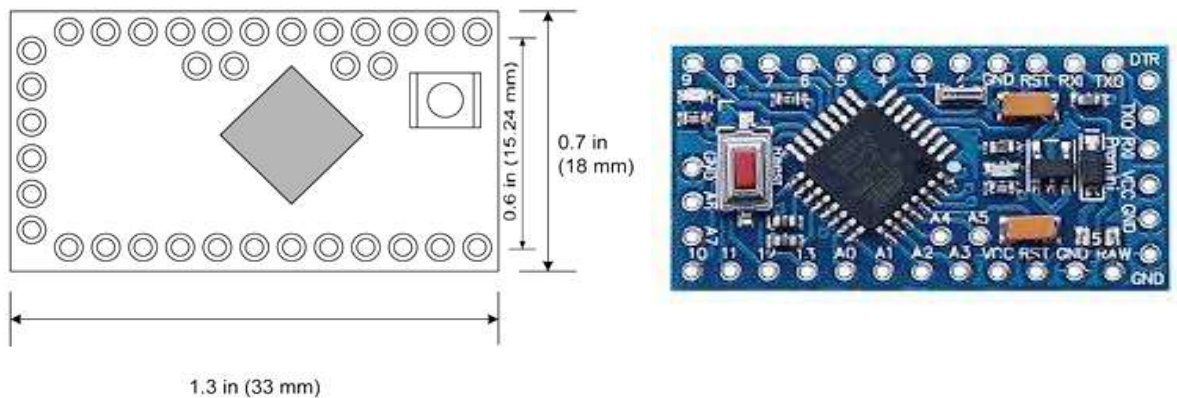
پردازنده مورد استفاده برای ساخت دستگاه انتهایی، مدل Pro Mini از سری میکروکنترلرهای آردوینو^۳ می‌باشد. این سری پردازنده علاوه بر اندازه کوچک، که باعث می‌شود دستگاه انتهایی اندازه مناسب و قابل حملی داشته

^۱ Sensor

^۲ Functional

^۳ Arduino

با شد، مصرف توان بسیار کمی نیز دارد به طوری که برای یک باتری کوچک با توان پایین می‌تواند این پردازنده را مدت زمان زیادی روشن نگاه دارد. در شکل‌های ۱۹، پردازنده مورد استفاده و همچنین ابعاد آن مشخص است. همچنین در شکل ۲۰ جریان مصرفی بردهای آردوینو با هم مقایسه شده است و همانطور که مشهود است برد Pro Mini کمترین جریان مصرفی را در میان بردهای آردوینو داراست.



شکل ۱۹ برد Arduino pro mini و ابعاد آن

| Microcontroller | Reference 9V | Reduce Clock Speed 9V | Reduce Clock Speed and Operation Voltage 3.3V | Enable Low Power Mode 3.3V | Enable Low Power Mode 9V |
|--------------------------|-----------------|--------------------------|--|-------------------------------|-----------------------------|
| Arduino Nano | 22.1 mA | 18.5 mA (-16%) | 3.4 mA (-85%) | 3.4 mA (-84%) | 4.8 mA (-78%) |
| Arduino Pro Mini 5V | 14.6 mA | 10.0 mA (-32%) | 3.7 mA (-75%) | 1.6 mA (-89%) | 3.2 mA (-78%) |
| Arduino Pro Mini 3.3V | 5.1 mA | 3.8 mA (-25%) | 3.7 mA (-27%) | 1.6 mA (-69%) | 3.2 mA (-38%) |
| Arduino Uno | 98.4 mA | 42.8 mA (-57%) | 11.6 mA (-88%) | 11.5 mA (-88%) | 27.9 mA (-72%) |
| Arduino Mega | 73.2 mA | 61.8 mA (-16%) | 16.7 mA (-77%) | 11.9 mA (-84%) | 26.9 mA (-63%) |

شکل ۲۰ مقایسه جریان مصرفی بردهای Arduino

۳-۲-۱-۱- برنامه نویسی دستگاه انتهایی

برای برنامه نویسی دستگاه انتهایی از کتابخانه^۱ LMIC آردوینو استفاده می‌کنیم. این کتابخانه امکان فعال‌سازی یک دستگاه انتهایی مبتنی بر LoRa را برای ما فراهم می‌کند همچنین به صورت پیش فرض در بخش example های محیط برنامه نویسی^۲ آردوینو موجود می‌باشد. برای استفاده از این کتابخانه نکاتی باید مورد توجه قرار بگیرد که در ادامه توضیح داده می‌شود.

در فایل Config.h از کتابخانه LMIC باید چند خط از کد را از حالت نظر^۳ خارج کرده و تغییراتی را بر روی فایل اعمال کنیم. این کتابخانه به صورت پیش فرض روی فرکانس کاری ۸۶۸ مگاهرتز کار می‌کند و باید آن را به ۹۱۵ مگاهرتز تغییر دهیم. تغییرات ذکر شده در شکل ۲۱ قابل مشاهده می‌باشد.

^۱ Library

^۲ IDE

^۳ Comment

```

8 // #define CFG_eu868 1
9 #define CFG_us915 1 To change frequency plan from 868 Mhz to 915 Mhz
10 // This is the SX1272/SX1273 radio, which is also used on the HopeRF
11 // RFM92 boards.
12 // #define CFG_sx1272_radio 1
13 // This is the SX1276/SX1277/SX1278/SX1279 radio, which is also used on
14 // the HopeRF RFM95 boards
15 #define CFG_sx1276_radio 1
16
17 // 16 µs per tick
18 // LMIC requires ticks to be 15.5µs - 100 µs long
19 #define US_PER_OSTICK_EXPONENT 4
20 #define US_PER_OSTICK (1 << US_PER_OSTICK_EXPONENT)
21 #define OSTICKS_PER_SEC (1000000 / US_PER_OSTICK)
22
23 // Set this to 1 to enable some basic debug output (using printf) about
24 // RF settings used during transmission and reception. Set to 2 to
25 // enable more verbose output. Make sure that printf is actually
26 // configured (e.g. on AVR it is not by default), otherwise using it can
27 // cause crashing.
28 #define LMIC_DEBUG_LEVEL 2
29
30 // Enable this to allow using printf() to print to the given serial port
31 // (or any other Print object). This can be easy for debugging. The
32 // current implementation only works on AVR, though.
33 #define LMIC_PRINTF_TO Serial For debuggin
34
35 // Any runtime assertion failures are printed to this serial port (or
36 // any other Print object). If this is unset, any failures just silently
37 // halt execution.
38 #define LMIC_FAILURE_TO Serial
39
40 // Uncomment this to disable all code related to joining
41 // #define DISABLE_JOIN
42 // Uncomment this to disable all code related to ping
43 #define DISABLE_PING
44 // Uncomment this to disable all code related to beacon tracking.
45 // Requires ping to be disabled too
46 #define DISABLE_BEACONS
47

```

شکل ۲۱ تغییرات لازم بر روی فایل config.h کتابخانه LMIC

بعد از تغییر فایل config.h، درون تابع setup باید مقدار ورودی تابع LMIC_selectSubBand را از 0 به 1 تغییر داد. در شکل زیر این تغییر مشخص شده است.

```
// setting up channels should happen after LMIC_setupSession, as that
// configures the minimal channel set.
// NA-US channels 0-71 are configured automatically
LMIC_setupChannel(0, 868100000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(1, 868300000, DR_RANGE_MAP(DR_SF12, DR_SF7B), BAND_CENTI); // g-band
LMIC_setupChannel(2, 868500000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(3, 867100000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(4, 867300000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(5, 867500000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(6, 867700000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(7, 867900000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(8, 868800000, DR_RANGE_MAP(DR_FSK, DR_FSK), BAND_MILLI); // g2-band
// TTN defines an additional channel at 869.525Mhz using SF9 for class B
// devices' ping slots. LMIC does not have an easy way to define set this
// frequency and support for class B is spotty and untested, so this
// frequency is not configured here.
#ifdef CFG_us915
// NA-US channels 0-71 are configured automatically
// but only one group of 8 should (a subband) should be active
// TTN recommends the second sub band, 1 in a zero based count.
// https://github.com/TheThingsNetwork/gateway-conf/blob/master/US-global_conf.json
LMIC_selectSubBand(1);
#endif
```

شکل ۲۲ تغییر مقدار ورودی تابع LMIC_selectSubBand به منظور کارکرد دستگاه در فرکانس ۹۱۵ مگاهرتز

برای بررسی راحت تر روند انجام کد و مشاهده logهای سیستم، از دستور Serial.begin(115200) برای شروع ارتباط سریال و از دستورهای Serial.println() برای نمایش مقادیر مورد نیاز بروی نمایشگر سریال، استفاده می‌کنیم. از دستور اول تنها یک بار و در ابتدای تابع setup و از دستور دوم به طور نامحدود در هر جای کد می‌توان استفاده کرد.

نکته مهم در مورد دستگاه انتهایی که با این کتابخانه کار می‌کند این است که این برنامه طوری تنظیم شده است که دستگاه انتهایی از تکنیک پرش کانال^۱ استفاده می‌کند و برای هر بار ارسال فرکانس خود را از فرکانس پایه ۲۰۰ هزار افزایش می‌دهد. این عمل ۵ بار تکرار شده و سپس دوباره از فرکانس پایه شروع به ارسال. این امر باعث می‌شود که احتمال تداخل سیگنال‌های دستگاه‌های انتهایی بسیار کاهش یابد و در نتیجه بسته‌ها با موفقیت به مقصد (دروازه‌ها) برسند.

برای تغییر محتوای پیام، که در اکثر کاربردها مقادیر حسگرها می‌باشند اما در این پروژه یک مقدار ثابت و دلخواه است، و همچنین مشخص کردن دوره زمانی ارسال پیام‌ها باید به ترتیب آرایه mydata[] و متغیر TX_INTERVAL را تغییر بدهیم. در شکل ۲۳ این تغییرات مشخص شدند.

^۱ Channel Hopping

```

57
58 uint8_t mydata[] = "node0";
59 static osjob_t sendjob;
60 // Schedule TX every this many seconds (might become longer due to duty
61 // cycle limitations).
62 const unsigned TX_INTERVAL = 30;
63

```

شکل ۲۳ تغییرات لازم بر روی کد برای ارسال داده مرد نظر در بازه‌های زمانی دلخواه

همانطور که در شکل ۱۸ مشخص است فاصله ارسال بسته‌ها برای دستگاه انتهایی، ۳۰ ثانیه تنظیم شده است. ارسال بسته در هر ۳۰ ثانیه در فرکانس‌های بدون مجوز^۱ ISM از شبکه LoRaWAN صحیح نمی‌باشد چرا که باید از مقررات چرخه وظیفه^۲ پیروی کرد. این مقررات برای ارسال بسته توسط دستگاه انتهایی ۱ درصد از زمان شبانه روز را مجاز می‌دانند. اما در این پروژه به منظور ارزیابی پژوهشی و به مدت کوتاه چند روزه این محدودیت را نادیده می‌گیریم و هر ۱۵ ثانیه از دستگاه انتهایی داده ارسال خواهیم کرد.

حال به تنظیم پارامترهای پهنای‌باند، عامل گسترش و توان ارسال دستگاه انتهایی می‌پردازیم. با توجه به قوانین LoRa Alliance نمی‌توان در دستگاه‌های انتهایی عامل گسترش را به صورت کد سخت^۳ بر روی مقادیر ۱۱ و ۱۲ تنظیم کرد. در صورت انجام این کار، سرور شبکه بسته‌ها را دریافت نکرده و اجازه ارتباط و ارسال داده از دستگاه انتهایی به سرور کاربرد را نمی‌دهد. در این پروژه از عامل گسترش ۹، پهنای باند ۱۲۵ کیلوهرتز و ماکسیمم توان ارسال یعنی ۱۴ دسیبل استفاده می‌کنیم. در شکل ۲۴ کد مربوط به تغییرات ذکر شده آورده شده است.

^۱ Industrial, Scientific, and Medical

^۲ Duty Cycle

^۳ Hard code

```
// Disable link check validation
LMIC_setLinkCheckMode(0);

// TTN uses SF9 for its RX2 window.
LMIC.dn2Dr = DR_SF9;

// Set data rate and transmit power for uplink (note: txpow seems to be ignored by the library)
LMIC_setDrTxpow(DR_SF9,14);

// Start job
do_send(&sendjob);
```

شکل ۲۴ کدهای مربوط به تنظیم عامل گسترش، پهنای باند و توان ارسالی

همانطور که در شکل بالا دیده می‌شود، پهنای باند به طور صریح مشخص نمی‌شود؛ بلکه ترکیب پهنای باند و عامل‌های گسترش متفاوت، عاملی با عنوان نرخ ارسال داده یا DR تشکیل می‌دهند. در شکل ۲۵ چند نمونه از این ترکیب‌ها نمایش داده شده است.

| Data Rate (DR) | Configuration | | | Physical Bit Rate (bit/s) |
|----------------|-------------------------|-----------------------|-----------|---------------------------|
| | Modulation | Spreading Factor (SF) | Bandwidth | |
| 0 | LoRa | SF12 | 125 kHz | 250 |
| 1 | LoRa | SF11 | 125 kHz | 440 |
| 2 | LoRa | SF10 | 125 kHz | 980 |
| 3 | LoRa | SF9 | 125 kHz | 1760 |
| 4 | LoRa | SF8 | 125 kHz | 3125 |
| 5 | LoRa | SF7 | 125 kHz | 5470 |
| 6 | LoRa | SF7 | 250 kHz | 11,000 |
| 7 | FSK | 50 kbit/s | | 50,000 |
| 8-15 | Reserved for Future Use | | | |

شکل ۲۵ نرخ داده‌های متفاوت با ترکیب عامل گسترش و پهنای باندهای مختلف

در نهایت برای اتصال برد Arduino به ماژول رادیویی LoRa نکته بسیار مهم رعایت نگاشت پایه‌ها^۱ است. در بخش ۲-۲-۳ پایه‌های مهم ماژول رادیویی مورد استفاده و نحوه اتصال آن‌ها توضیح داده خواهد شد؛ اما درون کد آردوینو باید اتصالات بین بردها و نگاشت پایه‌ها در قالب دستور مشخص شده باشند. با توجه به شکل ۲۹

^۱ Pin Mapping

آمده در بخش ۳-۲-۲ و مسیرهای^۱ مشخص شده در مدار چاپی بین بردها نگاشت مطلوب باید مشابه شکل ۲۶ باشد.

```
// Pin mapping
const lmic_pinmap lmic_pins = {
    .nss = 10,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 4,
    .dio = {2, 3, 5},
};
```

شکل ۲۶ نگاشت پایه‌ها (pin mapping) درون کد دستگاه انتهایی

۳-۲-۲-۲- LoRa و اتصال آن به پردازنده

برای ماژول رادیویی از LoRa Hope RF96 SX1276 استفاده شده است. این ماژول با ولتاژ ۳,۳ ولت کار می‌کند و همچنین برای ارتباط آن با پردازنده باید از پروتکل SPI^۲ استفاده نمود. بر روی این ماژول لایه فیزیکی LoRa پیاده‌سازی شده است که باعث می‌شود بتواند پیام‌های مورد نظر که از میکروکنترلر و از طریق پروتکل SPI به آن رسیده را با استفاده از ماژولیشن^۳ LoRa ارسال کند. در شکل ۲۷، این ماژول به همراه پایه‌های آن را مشاهده می‌نمایید.

^۱ Track

^۲ Serial Peripheral Interface

^۳ Modulation



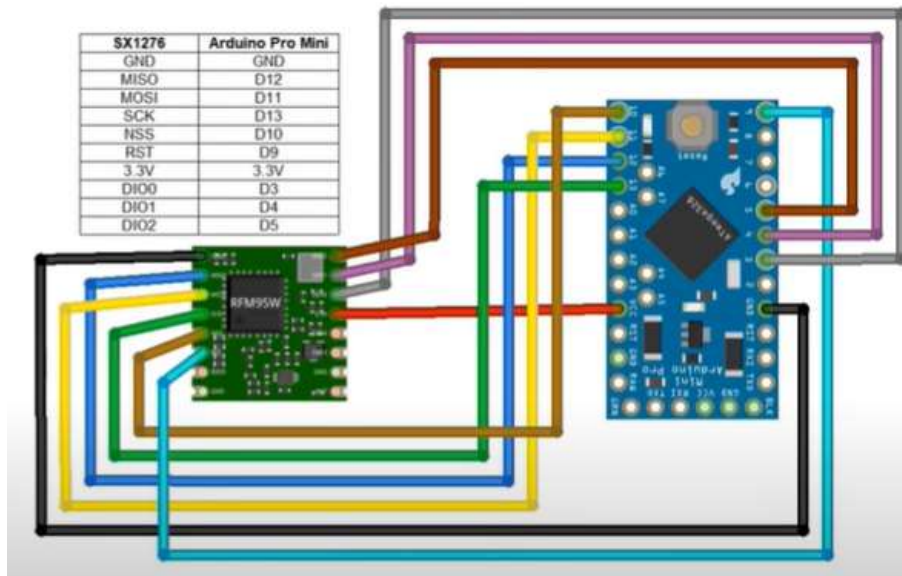
شکل ۲۷ ماژول رادیویی LoRa Hope RF96

همانطور که گفته شد برد Arduino و ماژول رادیویی برای ارتباط از پروتکل SPI استفاده می‌کنند که این پروتکل دارای ۴ پایه اصلی می‌باشد:

- پایه MOSI (Master out Slave in) که اطلاعات از دستگاه Master که، کنترلر Arduino می‌باشد به Slave که برد رادیویی است انتقال می‌یابد
- پایه MISO (Master in Slave out) که جریان انتقال اطلاعات برعکس MOSI را برقرار می‌کند
- پایه NSS (Slave Select) که در صورت وجود چند دستگاه جانبی می‌توان با استفاده از آن بین دستگاه‌ها (slave) انتخاب کرد.
- پایه SCK (Serial Clock) که نقش همگام‌سازی را با ارسال سیگنال ساعت از آردوینو به ماژول رادیویی انجام می‌دهد.

علاوه بر پایه‌های ذکر شده، برای استفاده از ماژول رادیویی RF96 چند پایه دیگر نیز مورد نیاز هستند:

- پایه ANT که آنتن به آن متصل می‌شود
 - پایه GND که به زمین متصل می‌شود
 - پایه 3.3V که به منبع تغذیه متصل می‌شود
 - پایه DIO (Digital Input/Output) برای ورودی و خروجی‌های دیجیتال
- اتصالات بین میکروکنترلر و ماژول رادیویی LoRa باید مشابه شکل زیر باشد.

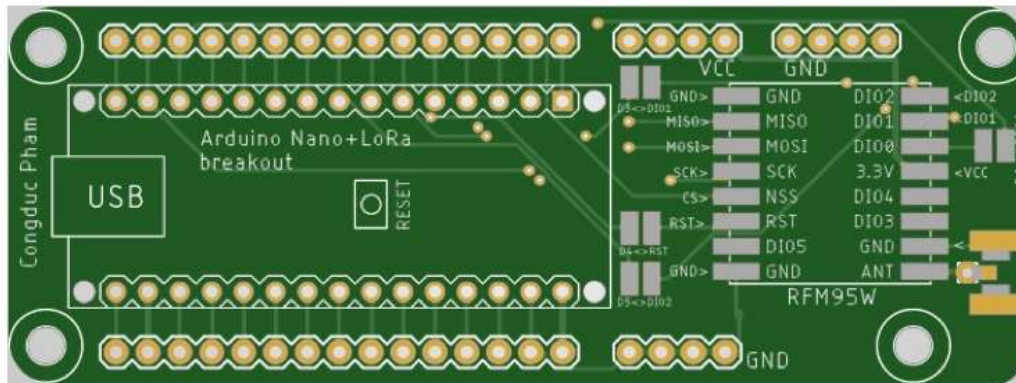


شکل ۲۸ اتصالات بین میکرو کنترلر و ماژول LoRa

حال برای برقراری ارتباط‌های ذکر شده بین برد Arduino و ماژول رادیویی LoRa به جای استفاده از برد هزارسوراخ^۱ و یا اتصال مستقیم با سیم مانند شکل ۲۸ از مدار چاپی^۲ استفاده می‌کنیم. مزیت استفاده از مدارهای چاپی برای اتصال ماژول‌های الکترونیکی حذف سیم‌های جامپر و برطرف شدن امکان قطعی اتصالات و نویز سیم‌ها بر روی هم و همچنین یکپارچگی، استحکام و قابل حمل بودن دستگاه انتهایی است. در شکل‌های ۲۹ و ۳۰ فایل طراحی شده این PCB در نرم افزار و همچنین برد چاپ شده را مشاهده می‌نمایید.

^۱ Bread Board

^۲ Printed Circuit Board (PCB)



شکل ۲۹ نمای PCB طراحی شده در نرم افزار [۲۰]



شکل ۳۰ برد PCB چاپ شده

۳-۲-۳- منبع تغذیه

همانطور که در بخش‌های ۱-۲-۳ و ۲-۲-۳ گفته شد به دلیل اینکه هم میکروکنترلر Arduino Pro Mini و هم ماژول رادیویی LoRa Hope RF96 کم مصرف هستند، دستگاه انتهایی LoRa مصرف جریان کمی خواهد داشت. علاوه براین به دلیل این که می‌خواهیم دستگاه نهایی از لحاظ ابعاد کوچک و همچنین سبک وزن باشد، از باتری قابل شارژ لیتیوم پلیمر با ظرفیت ۱۰۰۰ میلی آمپر ساعت مانند شکل ۳۱ استفاده می‌کنیم. برای اینکه از این باتری قابل شارژ در دستگاه انتهایی استفاده کنیم، باید از یک برد ماژول تغذیه که در ادامه جزئیات آن را بیان می‌شود استفاده کنیم.



شکل ۳۱ باتری قابل شارژ Sonikcell با ظرفیت ۱۰۰۰ میلی آمپر ساعت (سمت راست) و ۲۰۰۰ میلی آمپر ساعت (سمت چپ) برای ساخت ماژول تغذیه مهم‌ترین قطعه، ماژول شارژر TP4056 می‌باشد. در صورت اتصال باتری به پایه‌های B+ و B- این ماژول، می‌توان با استفاده از کابل Micro USB باتری را شارژ نمود. همچنین ۲ پایه out+ و out- بر روی ماژول برای اتصال به دستگاه مورد نظر استفاده می‌شود.

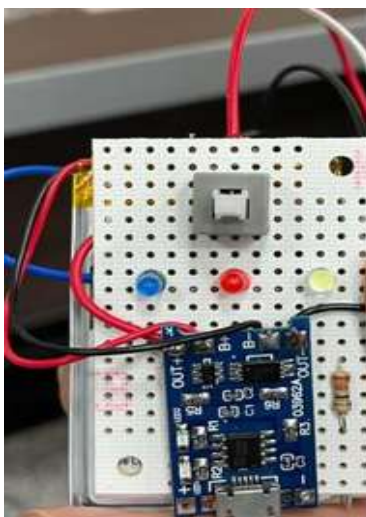


شکل ۳۲ ماژول TP4056

به دلیل این‌که ماژول تغذیه به دستگاه انتهایی متصل می‌شود می‌توان به طراحی آن علاوه بر باتری و TP4056 چند LED نیز متصل کرد. دلیل انجام این کار این است که میکروکنترلر مورد استفاده برای این کاربرد چند

پایه چندمنظوره ورودی و خروجی^۱ استفاده نشده دارد و می‌توان از اتصال این پایه‌ها به LED، لحظه ارسال و دریافت داده توسط دستگاه انتهایی را مشخص کرد. همچنین با اضافه کردن یک دکمه فشاری^۲ و یک LED جدا، روشن و خاموش بودن ماژول تغذیه را می‌توان تشخیص داد. در نهایت برای ساخت ماژول تغذیه با ویژگی‌های گفته شده از ۳ چراغ LED کوچک به رنگ‌های آبی، سفید و قرمز، مقاومت‌های ۳۳۰ اهم و دکمه ۶ پایه استفاده می‌کنیم. در شکل ۳۳ ماژول تغذیه ساخته شده را مشاهده می‌کنید.

با فشردن دکمه این ماژول، LED قرمز روشن می‌شود که نشانگر وصل شدن جریان باتری می‌باشد. با وصل کردن این ماژول به دستگاه انتهایی، هنگام ارسال داده از دستگاه انتهایی LED آبی به مدت ۲ ثانیه روشن می‌شود و همچنین هنگام دریافت داده توسط دستگاه انتهایی LED سفید به مدت ۲ ثانیه روشن خواهد شد.



شکل ۳۳ ماژول تغذیه

^۱ General purpose Input/Output (GPIO)

^۲ Dip Switch

۳-۲-۴- اتصال و راه‌اندازی دستگاه انتهایی

برای راه‌اندازی و استفاده از دستگاه انتهایی نیاز به آنتن‌های مناسب داریم. فرکانسی مورد استفاده برای LoRaWAN در ایران مقدار ۹۱۵ مگاهرتز می‌باشد. بنابراین آنتن‌های مورد نیاز برای دستگاه انتهایی باید از نوع ۹۱۵ مگاهرتز باشند. در شکل ۳۴ آنتن بزرگ و کوچک در این فرکانس کاری را مشاهده می‌نمایید.



شکل ۳۴ آنتن‌های ۹۱۵ مگاهرتز در سایز کوچک و بزرگ

حال که تمام بخش‌های مورد نیاز برای دستگاه انتهایی معرفی شدند به اتصال آن‌ها و ایجاد دستگاه می‌پردازیم. در ابتدا میکروکنترلر و ماژول LoRa Hope RF96 را به PCB چاپ شده در شکل ۲۵ متصل می‌کنیم. سپس رابط آنتن که به رنگ طلایی در شکل ۳۵ مشخص است را به PCB لحیم می‌کنیم. در نهایت با وصل کردن آنتن به برد دستگاه نهایی مطابق شکل ۳۰ می‌شود.



شکل ۳۵ دستگاه انتهایی

پس از اتمام اتصالات، ماژول تغذیه را به دستگاه انتهایی می‌چسبانیم و پایه‌های آن را متصل می‌کنیم. در شکل ۳۶ نمونه تمام شده دستگاه انتهایی را مشاهده می‌کنید.



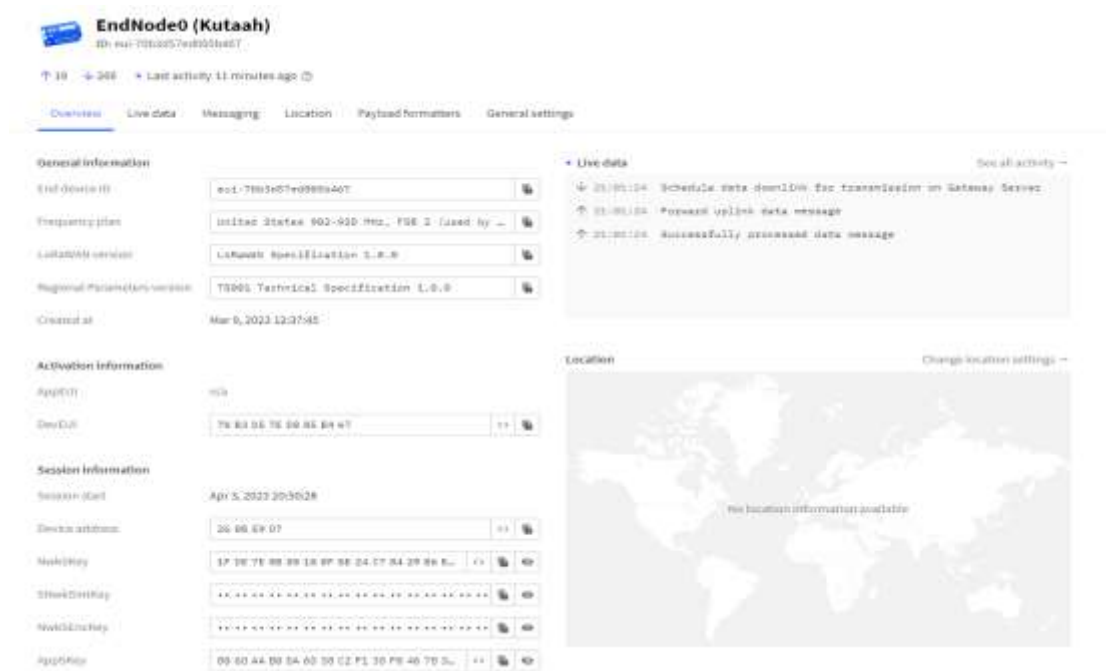
شکل ۳۶ دستگاه انتهایی به همراه ماژول تغذیه

۳-۲-۴-۱- راه‌اندازی دستگاه انتهایی

در بخش‌های قبل سخت افزار دستگاه انتهایی تکمیل شد و همچنین توضیحاتی درمورد کد دستگاه انتهایی داده شد. حال به بررسی تغییرات نهایی در کد Arduino و استفاده از پلتفرم TTN می‌پردازیم. همانطور که در فصل ۲ توضیح داده شد، دستگاه انتهایی در شبکه LoRaWAN را به یکی از دو روش ABP یا OTAA می‌توان راه‌اندازی کرد. در این پروژه به منظور جلوگیری از پیچیدگی اضافی و با توجه به اینکه پیام‌های ارسالی توسط دستگاه انتهایی محتوایی ندارند و هدف بررسی نویز شبکه و مشخصات سیگنال‌ها می‌باشد، از روش ABP برای راه‌اندازی دستگاه انتهایی استفاده می‌کنیم.

برای استفاده از روش ABP باید کلیدهای NwkSKey و AppSKey را از پلتفرم TTN استخراج کرده (استفاده از این پلتفرم در بخش ۳-۴ توضیح داده می‌شود)، و در کد دستگاه انتهایی قرار دهیم. همچنین آدرس دستگاه را باید

مطابق آدرس موجود در سرور شبکه تغییر دهیم تا دستگاه انتهایی توسط سرور شبکه احراز هویت شود و داده‌های آن دریافت شود. در شکل ۳۷ کلیدها و آدرس گفته شده را در پلتفرم TTN و در شکل ۳۸ محل قرار گرفتن آن‌ها در کد دستگاه انتهایی را مشاهده می‌کنید.



شکل ۳۷ کلید NwkSKey و AppSKey و آدرس دستگاه در پلتفرم TTN

```
static const PROGMEM u1_t NwksKey[16] = {0x1F, 0xDE, 0x7E, 0x00, 0x80, 0x10, 0x8F, 0x0E, 0x24, 0xC7, 0x84, 0x29, 0x86, 0xEF, 0x6D, 0x46};

// LoRaWAN AppSKey, application session key
// This is the default Semtech key, which is used by the early prototype TTN
// network.
static const u1_t PROGMEM AppSKey[16] = {0x80, 0x6D, 0xAA, 0x00, 0xDA, 0x6D, 0x38, 0xC2, 0xF1, 0x35, 0xF0, 0x46, 0x78, 0x32, 0x04, 0x47};

// LoRaWAN end-device address (DevAddr)
static const u1_t DevAddr = 0x260B5907; // <-- Change this address for every node!
```

شکل ۳۸ تعریف کلید NwkSKey و AppSKey و آدرس دستگاه درون کد Arduino

پس از انجام این کار، با روشن کردن دستگاه و ارسال داده از سمت آن، در صورت دریافت داده به صورت سالم در دروازه LoRaWAN، در سرور شبکه و سرور کاربرد قادر به مشاهده داده ارسال شده توسط دستگاه انتهایی خواهیم بود. کد دستگاه انتهایی در پیوست این پایان‌نامه قرار داده شده است.

پس از تنظیم کلیدهای گفته شده در صورت روشن کردن دستگاه، در صفحه مربوط به دستگاه‌های انتهایی پلتفرم TTN داده‌های ارسال شده به همراه مشخصات سیگنال و زمان ارسال نمایش داده می‌شوند. در شکل زیر نمایی از این صفحه را مشاهده می‌کنید.

| Time | Type | Data preview |
|----------|-----------------------------------|--|
| 21:17:31 | Schedule data downlink for trans. | DevAddr: 26 08 59 07 MAC payload: 74 C4 06 07 FC 2B 07 76 ... End Delay: 1 |
| 21:17:31 | Forward uplink data message | DevAddr: 26 08 59 07 Payload: { text: "hello" } 0E 6F 04 05 38 ... RPort: 1 Data rate: SF08x125 BW: 0 RSSI: -69 |
| 21:17:31 | Successfully processed data mess. | DevAddr: 26 08 59 07 |
| 21:16:59 | Schedule data downlink for trans. | DevAddr: 26 08 59 07 MAC payload: 8B 7F 20 45 49 20 5F 0D ... End Delay: 1 |
| 21:16:59 | Forward uplink data message | DevAddr: 26 08 59 07 Payload: { text: "hello" } 0E 6F 04 05 38 ... RPort: 1 Data rate: SF08x125 BW: 11.8 RSSI: -75 |
| 21:16:59 | Successfully processed data mess. | DevAddr: 26 08 59 07 |
| 21:16:33 | Schedule data downlink for trans. | DevAddr: 26 08 59 07 MAC payload: E5 78 70 07 FE 00 0A 3A ... End Delay: 1 |
| 21:16:33 | Forward uplink data message | DevAddr: 26 08 59 07 Payload: { text: "hello" } 0E 6F 04 05 38 ... RPort: 1 Data rate: SF08x125 BW: 12.5 RSSI: -67 |
| 21:16:33 | Successfully processed data mess. | DevAddr: 26 08 59 07 |

شکل ۳۹ مشخصات پیام دریافتی دستگاه انتهایی در پلتفرم TTN

۳-۳- دروازه شبکه LoRaWAN

دروازه در زیر ساخت LoRaWAN نقش اساسی دارد، دروازه‌ها حد فاصلی بین دستگاه‌های انتهایی و سرور شبکه می‌باشد. دروازه‌ها پیام‌های ارسال شده توسط دستگاه‌های انتهایی را با ماژولیشن LoRa دریافت می‌کنند و سپس این پیام‌ها را با استفاده از شبکه اینترنت به سمت سرور شبکه ارسال می‌کنند. دروازه‌های شبکه LoRaWAN به دو دسته کلی تک کاناله^۱ و چند کاناله^۲ تقسیم می‌شوند.

برای ساخت دروازه شبکه LoRaWAN به یک سخت افزار با قابلیت اجرای سیستم عامل داشته احتیاج داریم. به این منظور از یک کامپیوتر کوچک Raspberry pi استفاده می‌کنیم. سپس برنامه دروازه LoRaWAN به صورت پس‌زمینه در سیستم عامل اجرا می‌شود و سرویس دروازه را ارائه می‌دهد. علاوه بر سخت افزار ذکر شده،

¹ Single channel Gateway

² Multi channel Gateway

به ماژول رادیویی LoRaWAN که نقش دریافت و ارسال سیگنال‌های رادیویی بین دروازه و دستگاه‌های انتهایی را بر عهده دارد نیاز خواهیم داشت. همانطور که گفته شد دروازه‌ها به دو دسته کلی تک کاناله و چند کاناله تقسیم می‌شوند. دروازه‌های تک کاناله تنها در یک بازه فرکانسی داده‌ها را دریافت می‌کنند ولی دروازه‌های چند کاناله تمام پیام‌ها را در بازه‌های فرکانسی مجزا دریافت می‌نمایند. از آنجایی که دستگاه انتهایی طراحی شده از تکنیک پرش کانال استفاده می‌کند بنابراین در این پروژه باید از دروازه‌های چند کاناله بهره برد.

برای پیاده‌سازی یک دروازه چند کاناله، به یک کامپیوتر کوچک Raspberry pi و یک ماژول فرستنده-گیرنده^۱ LoRa نیاز داریم. همانطور که در فصل قبل گفته شد، دروازه‌های شبکه LoRaWAN می‌توانند دارای فرستنده-گیرنده ۸ کاناله، ۱۶ کاناله و یا ۶۴ کاناله باشند. در این پروژه ما از فرستنده-گیرنده ۸ کاناله RAK831 استفاده خواهیم کرد [۲۱]. در شکل ۴۰ ماژول فرستنده-گیرنده RAK831 را مشاهده می‌کنید.



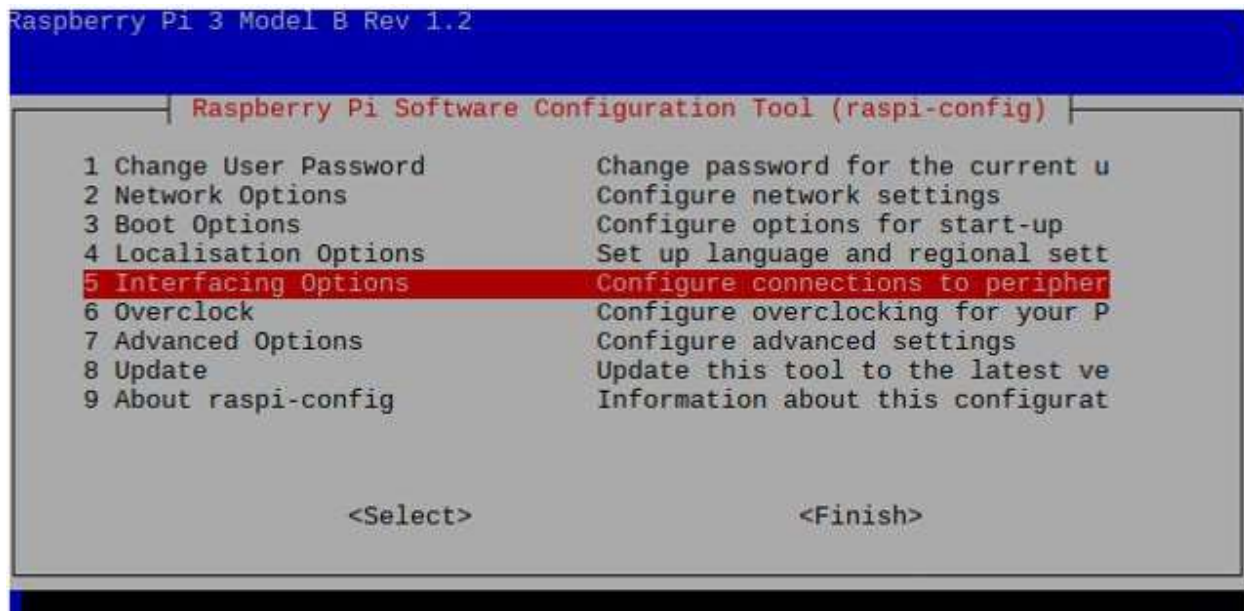
شکل ۴۰ ماژول فرستنده-گیرنده RAK831

برای برقراری ارتباط میان Raspberry pi و ماژول فرستنده-گیرنده باید پایه‌های SPI را در Raspberry pi فعال کنیم. برای این کار به بخش ترمینال دستگاه رفته و دستور زیر را وارد می‌کنیم و سپس مطابق شکل‌های ۴۱ تا ۴۴ عمل می‌کنیم تا پروتکل SPI روی Raspberry pi فعال شود.

- sudo raspi-config

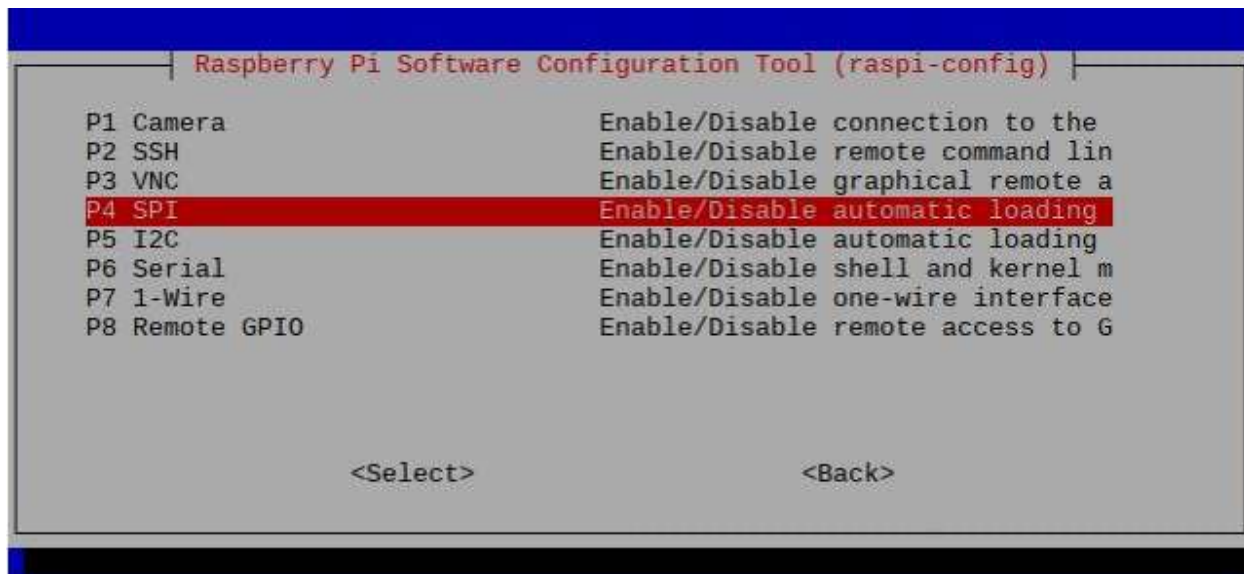
با زدن این دستور پنجره زیر باز می‌شود:

^۱ Transceiver



شکل ۴۱ محیط پیکربندی Raspberry pi

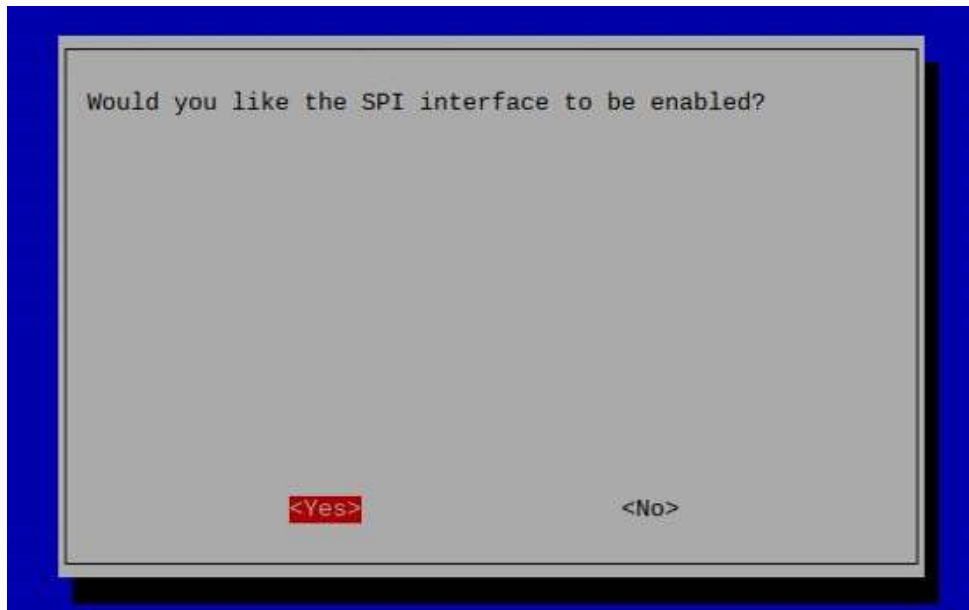
در پنجره باز شده قسمت Interfacing Options را انتخاب می‌کنیم تا وارد پنجره زیر شویم:



شکل ۴۲ پنجره رابط‌های^۱ Raspberry pi

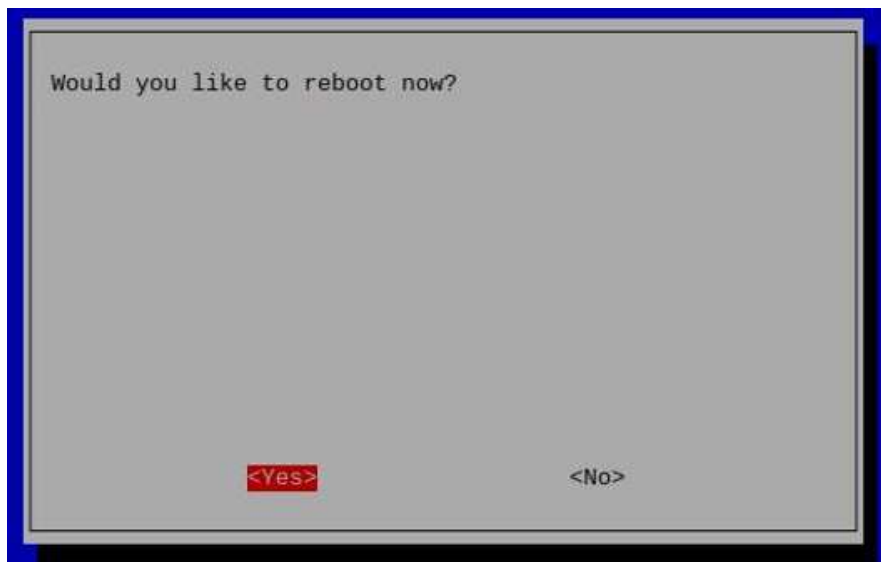
در این پنجره گزینه SPI را انتخاب کرده و آن را مطابق شکل زیر فعال می‌سازیم:

^۱ interface



شکل ۴۳ فعالسازی رابط جانبی سریال

پس از انجام این مراحل پیشنهاد سیستم مبتنی بر reboot کردن دستگاه را انجام می‌دهیم:



شکل ۴۴ درخواست reboot شدن دستگاه

حال Raspberry pi می‌تواند با ماژول RAK831 ارتباط برقرار کند و با reboot شدن دستگاه، کار آن به عنوان دروازه LoRaWAN آغاز می‌شود.

باید توجه داشت که ارتباط Raspberry pi با سرور شبکه از طریق شبکه اینترنت انجام می‌شود. لذا باید Raspberry pi را به اینترنت متصل کرد. برای این کار یا باید از شبکه WiFi و یا از کابل Ethernet استفاده کنیم. با اتصال کابل Ethernet دستگاه خود به خود در هنگام روشن شدن به اینترنت متصل می‌شود. اما برای اتصال خودکار به شبکه WiFi باید اطلاعات مربوط به شبکه WiFi شامل شناسه سرویس شبکه و رمز آن را از قبل بر روی Raspberry pi ثبت کنیم. این کار را در فایلی به نام wpa_supplicant.conf انجام می‌دهیم. با دستور زیر می‌توانیم به این فایل دسترسی پیدا کنیم:

```
- sudo nano/etc/wpa_supplicant/wpa_supplicant.conf
```

و سپس اطلاعات ذکر شده را با قالب زیر در آن ثبت می‌کنیم:

```
network={
    ssid="Wifi network name (SSID)"
    psk="Wifi password"}
```

با انجام این کار، Raspberry pi به محض روشن شدن و در صورت در دسترس بودن شبکه WiFi که برای آن تعریف کردیم، به آن وصل می‌شود.

حال برای اتصال پایه‌های مرتبط Raspberry pi و RAK831 به یکدیگر از شکل زیر استفاده می‌کنیم. نکته حائز اهمیت در اتصالات شکل زیر آن است که پایه شماره ۱۷ در Raspberry pi مقدار 3.3v است. بنابراین این پایه نباید همواره به پایه شماره ۱۹ RAK831 متصل باشد و تنها در صورت نیاز به reset کردن دستگاه باید به طور لحظه‌ای به آن متصل گردد. برای این کار می‌توان بین پایه‌های نام برده شده یک عدد کلید فشاری^۱ قرار داد.

^۱ Push Button

| RAK831 Pins | Description | Raspberry Pi Pin |
|-------------|-------------------|------------------|
| 1 | 5V | 2 |
| 3 | GND | 6 |
| 19 | RST (Reset Pin) | 17 |
| 18 | SCK (SPI Clock) | 23 |
| 17 | MISO | 21 |
| 16 | MOSI | 19 |
| 15 | CSN (Chip select) | 24 |

شکل ۴۵ نحوه اتصال پایه های Raspberry pi و RAK831

بعد از این کار باید برنامه فرستنده چند کاناله را روی دستگاه Raspberry pi نصب کنیم تا در هنگام روشن کردن Raspberry pi، این دستگاه به عنوان دروازه LoRaWAN عمل کند. این برنامه در پس زمینه کامپیوتر پس از روشن شدن دستگاه اجرا خواهد شد. این برنامه را می‌توان از مخزن گیت هاب^۱ بر روی Raspberry pi نصب کرد [22].

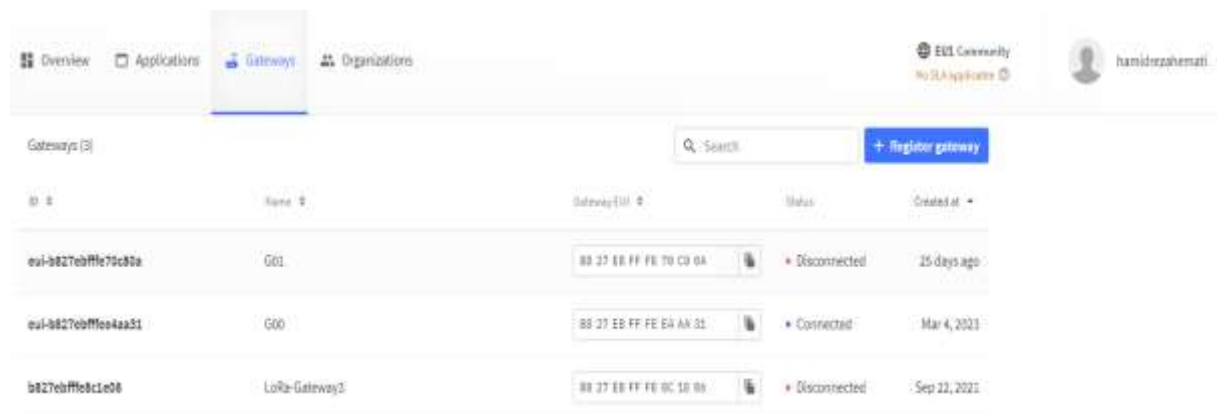
در ابتدای شروع نصب برنامه ذکر شده (بعد از زدن دستور `./install.bash` درون ترمینال) آدرس یکتای دروازه (EUI) بر روی ترمینال نمایش داده می‌شود. الزامی است که مقدار EUI دروازه حتما درجایی ذخیره شود زیرا که همانطور که در بخش ۳-۳-۱ گفته می‌شود برای ثبت دروازه بر روی پلتفرم TTN از EUI استفاده می‌شود. پس از این مراحل و انجام تغییرات نام دروازه و تنظیمات اتصالات شبکه آن، دروازه LoRaWAN آماده به کار خواهد بود. نکته مهم این است که باید قبل از روشن کردن دروازه ساخته شده، آنتن را به فرستنده-گیرنده متصل کنیم. لازم به ذکر است که برای این پروژه با توجه به منابع موجود، به پیاده‌سازی ۳ دروازه LoRaWAN پرداخته‌ایم.

^۱ Github repository

پس از اتمام پیاده‌سازی دروازه‌های LoRaWAN باید آنها را در سرور شبکه ثبت کنیم و همچنین تغییراتی را در فایل global_conf.json مربوط به برنامه‌ای که در آن‌ها اجرا می‌شود اعمال کنیم تا ارتباط دروازه‌ها با سرور شبکه به درستی برقرار شود. در بخش ۳-۳-۱ ابتدا نحوه ثبت کردن دروازه در سرور شبکه TTN و سپس به تغییراتی که باید در فایل‌های پیکربندی اعمال کنیم می‌پردازیم.

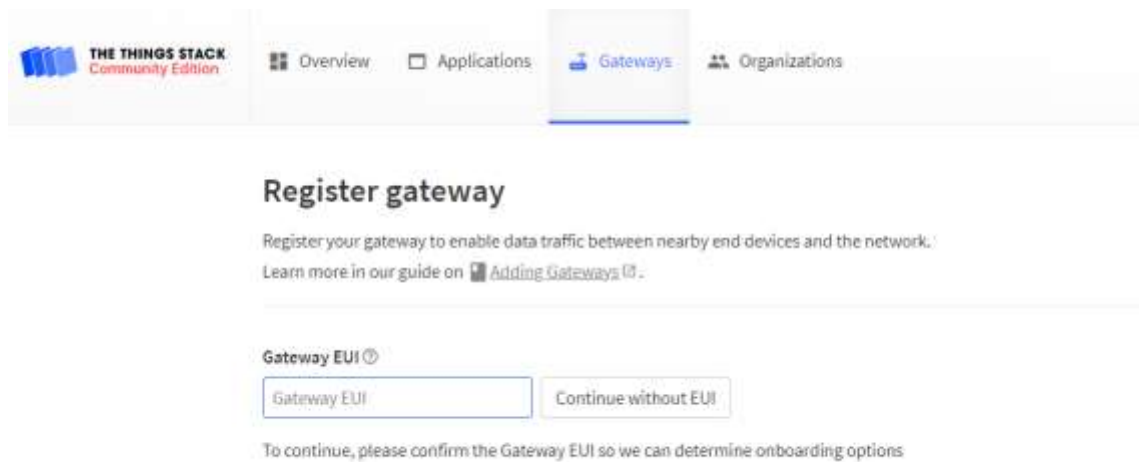
۳-۳-۱- ثبت دروازه‌ها در پلتفرم TTN

برای اضافه کردن دروازه در پلتفرم TTN ابتدا به بخش console شخصی خود رفته و سپس در بخش Gateways، اقدام به اضافه کردن دروازه می‌کنیم. در شکل ۴۶ صفحه مربوط به دروازه‌ها را مشاهده می‌کنید. همانطور که مشخص است دروازه‌های ایجاد شده به همراه وضعیت فعلی (متصل یا غیر متصل بودن آن‌ها) و همچنین کد منحصر به فرد آن‌ها را نشان داده شده است. در سمت بالا راست، گزینه اضافه کردن دروازه مشخص شده است.



شکل ۴۶ صفحه Gateways در پلتفرم TTN

پس از انتخاب گزینه register gateway به صفحه زیر منتقل می‌شوید. در اینجا باید مقدار EUI که در بخش قبل توضیح داده شد را وارد نموده و سپس تایید کنید.



THE THINGS STACK
Community Edition

Overview Applications **Gateways** Organizations

Register gateway

Register your gateway to enable data traffic between nearby end devices and the network.
Learn more in our guide on [Adding Gateways](#).

Gateway EUI

Gateway EUI Continue without EUI

To continue, please confirm the Gateway EUI so we can determine onboarding options

شکل ۴۷ تایید آدرس یکتای دروازه

در ادامه باید نام و فرکانس کاری دروازه را مشخص نمود. فرکانس کاری باید مطابق با شکل زیر و حول ۹۱۵ مگاهرتز و FSB2 انتخاب شود.

Register gateway

Register your gateway to enable data traffic between nearby end devices and the network.
Learn more in our guide on [Adding Gateways](#).

Gateway EUI ⓘ

22 22 22 22 22 22 22 22

Reset

Gateway ID ⓘ *

eui-2222222222222222

Gateway name ⓘ

test gateway

Frequency plan ⓘ *

United States 902-928 MHz, FSB 2 (used by TTN) ▼

☐ Require authenticated connection ⓘ

Choose this option eg. if your gateway is powered by [LoRa Basic Station](#)

Share gateway information

Select which information can be seen by other network participants, including [Packet Broker](#)

☒ Share status within network ⓘ

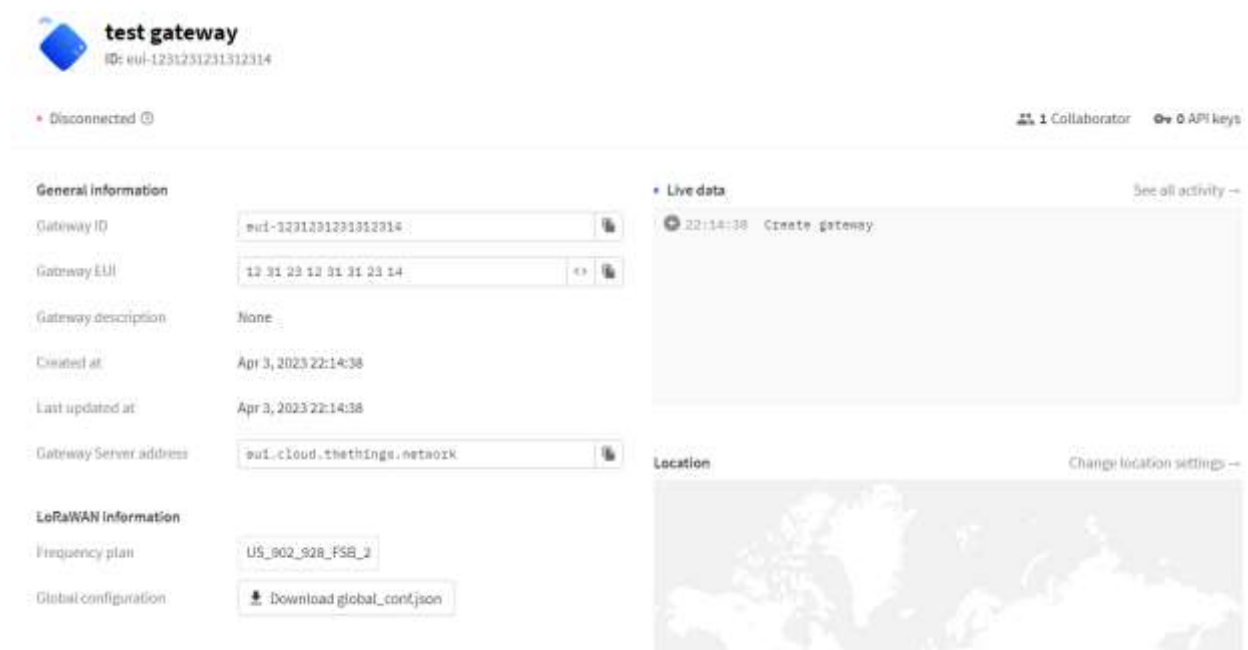
☒ Share location within network ⓘ

Register gateway

شکل ۴۸ انتخاب نام و فرکانس کاری دروازه

پس از ثبت دروازه به صفحه زیر منتقل می‌شویم. پس از این کار باید پیکربندی دروازه را طوری تغییر دهیم که ارتباط آن با سرور شبکه برقرار شود و اطلاعات را برای آن ارسال کند. برای این کار باید فایل global_conf.json را دانلود کرده و درون مسیر زیر در raspberry pi قرار دهیم.

/opt/ttn-gateway/bin/global_conf.json



شکل ۴۹ ثبت کامل دروازه و دریافت فایل global_conf.json

پس از انجام این کار، دروازه روی پورت شماره ۱۷۰۰ اطلاعات را با سرور شبکه رد و بدل می‌کند و همچنین در بخش دروازه‌ها در سرور شبکه وضعیت دروازه به حالت connected تغییر می‌کند. پس از راه‌اندازی دروازه‌ها، زیر ساخت فیزیکی شبکه LoRaWAN کامل می‌شود و حال تنها باید اطلاعات لازم را در سرور کاربرد تعریف کنیم.

۳-۴- ایجاد سرور شبکه و سرور کاربرد

یک عضو مهم و اصلی دیگر در شبکه LoRaWAN سرور شبکه می‌باشد که نقش اساسی در شبکه ایفا می‌کند. دروازه‌ها با اتصال به سرور شبکه داده‌های دریافت شده‌شان را برای سرور می‌فرستند و سرور شبکه آنها را دریافت و بسته‌های تکراری را دور می‌اندازد. ویژگی‌هایی که یک سرور شبکه باید دارا باشد را می‌توان به این صورت بیان کرد:

- پشتیبانی از کلاس A، B و C دستگاه‌های انتهایی
- پشتیبانی از الگوریتم ADR

- نشان دادن داده‌های دریافت شده در سرور به صورت زنده
- امکان پیکربندی پارامترهای مختلف در آن مانند پارامترهای مربوط به کانال مورد استفاده
- سازگاری با لایه MAC فناوری LoRaWAN
- امکان استفاده از API^۱ ها و امکان ادغام^۲ آن با سرورها و کاربردهای دیگر

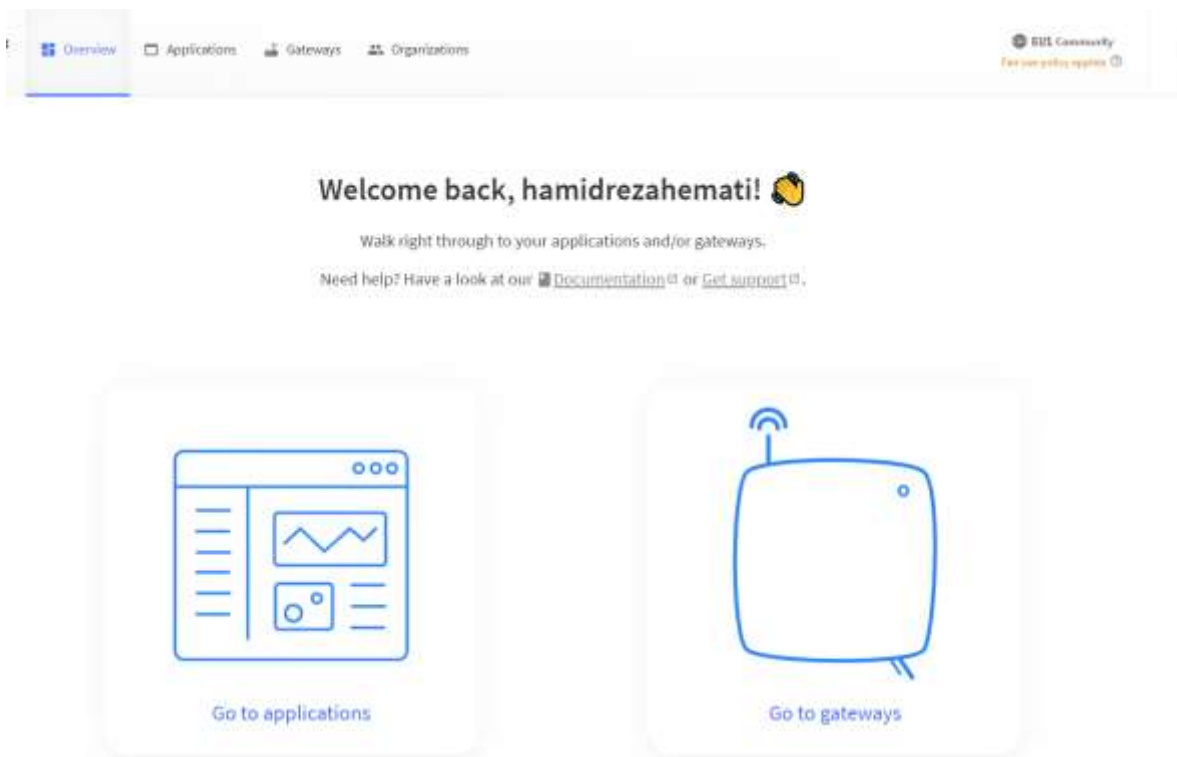
علاوه بر موارد ذکر شده سرور می‌تواند توانایی‌های دیگری هم مانند تحلیل داده‌های دریافتی و یا امکان ذخیره داده‌ها را داشته باشد، اما ضروری است که امکانات ذکر شده در آن پیاده‌سازی شده باشد. سرورهای شبکه مختلفی برای LoRaWAN وجود دارد و همچنین می‌توان از منابع متن‌باز مانند سرور شبکه ChirpStack استفاده کرد.

انتخاب ما برای این پروژه پلتفرم TTN می‌باشد زیرا هدف ما پژوهشی است و نیازی به خرید سرویس‌های خاص نداریم. پلتفرم TTN نسخه Community دارد و می‌توان به صورت رایگان برای تعداد محدودی دستگاه انتهایی و دروازه از آن استفاده کرد. پس از ثبت نام در پلتفرم می‌توان در بخش Console جزئیات شکل ۵۰ را مشاهده کرد و در بخش Gateways همانطور که در قسمت پیاده‌سازی دروازه توضیح داده شد، می‌توانیم دروازه‌های ساخته شده را در سرور شبکه معرفی کنیم و ارتباط آنها را برقرار سازیم.

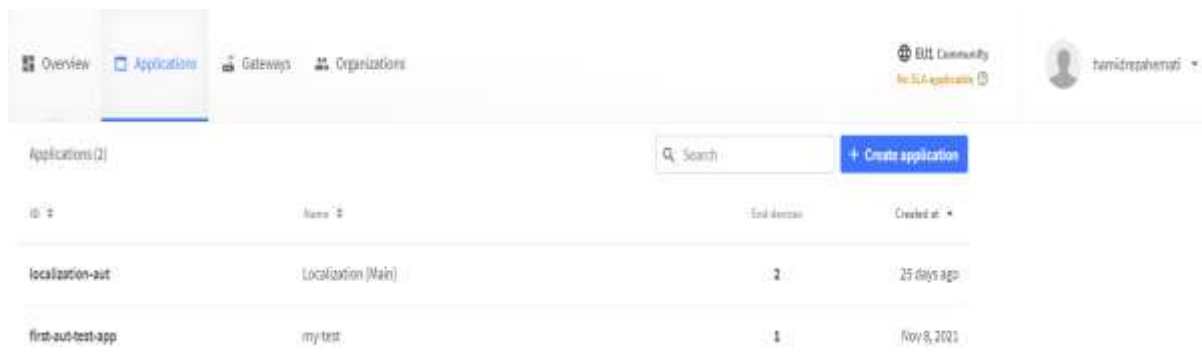
برای پیاده‌سازی سرور کاربرد به بخش Applications می‌رویم و با تعریف یک کاربرد جدید می‌توانیم به تعداد دلخواه دستگاه انتهایی به آن کاربرد اضافه کنیم. در این بخش با زدن Add application می‌توانیم کاربرد جدید دلخواه را اضافه کنیم. در شکل ۵۱ فضای کاربرد قابل مشاهده است.

^۱ Application programming interface

^۲ Integration



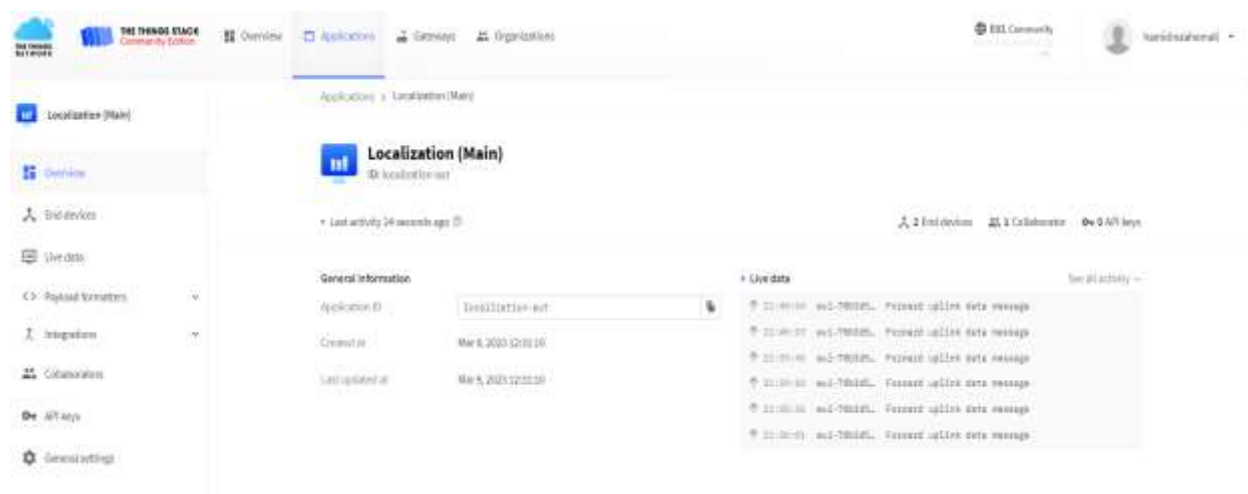
شکل ۵۰ پلتفرم TTN



شکل ۵۱ بخش کاربردها در پلتفرم TTN

همانطور که در شکل بالا مشاهده می‌شود، کاربردهایی از قبل در این سرور تعریف کرده‌ایم. سپس به کاربرد مورد نظر رفته و دستگاه‌های انتهایی را اضافه می‌کنیم. برای اضافه کردن دستگاه انتهایی Add end-device را می‌زنیم. برای دیدن داده‌های دریافتی به صورت زنده از منو سمت چپ به قسمت Live data می‌رویم و برای ادغام کردن سرور کاربرد با کاربردهای دیگر یا وب سرویس‌های دیگر از بخش Integrations اقدام می‌کنیم. موارد ذکر شده

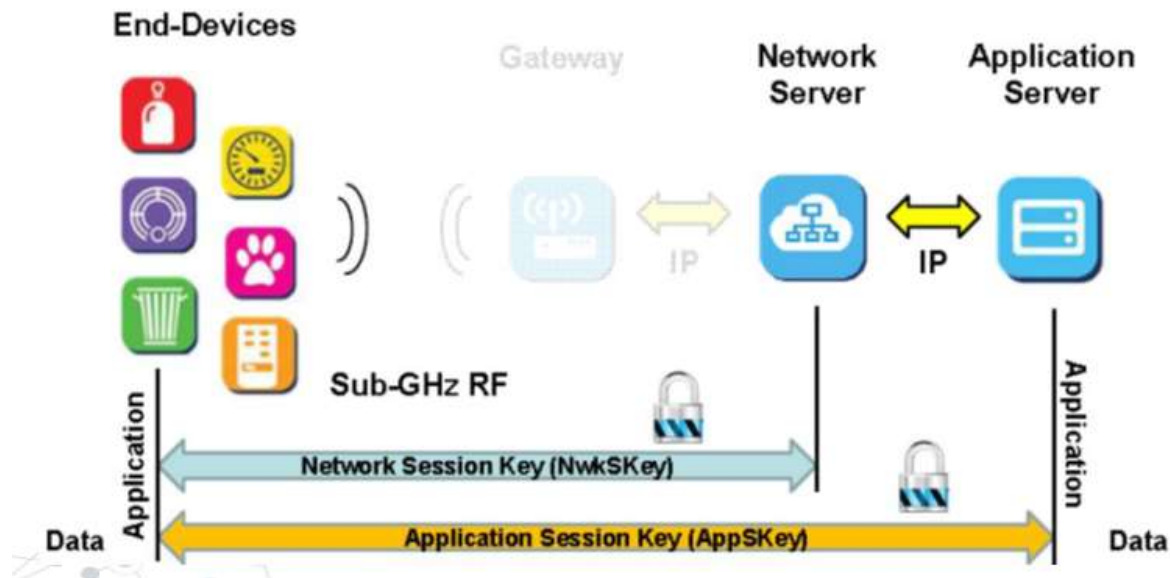
را در شکل ۵۲ مشاهده می‌کنید.



شکل ۵۲ سرور کاربرد در پلتفرم TTN

همانطور که مورد انتظار است داده‌های دریافت شده در پلتفرم TTN در بخش سرور شبکه که در قسمت Gateways قابل مشاهده است، به صورت رمز شده می‌باشند و محتوای پیام‌ها قابل رویت نمی‌باشد. اما در سرور کاربرد با توجه به اینکه رمزگذاری لایه دوم باز می‌شود می‌توان محتوای پیام‌های دستگاه‌های انتهایی را مشاهده کرد.

برای اضافه کردن یک دستگاه انتهایی جدید گزینه‌های متعددی وجود دارد که می‌توان از امکانات از پیش تعیین شده در خود پلتفرم و یا به صورت دستی اقدام کرد. اطلاعاتی از قبیل کلید جلسه شبکه و کلید جلسه کاربرد و آدرس دستگاه در قسمت سرور کاربرد ساخته و استخراج می‌شود. در بخش ۳-۲-۴-۱ این موارد همراه با تصویر از پلتفرم و کد توضیح داده شدند. در تصویر زیر روند جریان داده با توجه به کلیدهای جلسه و شبکه را در سرور شبکه و سرور کاربرد مشاهده می‌نمایید.



شکل ۵۳ روند جریان داده با توجه به کلیدهای جلسه و شبکه [۱۰]

جمع‌بندی

در این فصل به جزئیات پیاده‌سازی شبکه LoRaWAN پرداختیم. در ابتدا نحوه طراحی و ساخت دستگاه انتهایی را توضیح دادیم، سپس به پیاده‌سازی دروازه‌های شبکه پرداختیم. در ادامه با سرور شبکه و سرور کاربرد مورد استفاده آشنا شدیم و نحوه ثبت دروازه‌های در سرور شبکه و دستگاه‌های انتهایی در سرور کاربرد را شناختیم. برای این پروژه ۱ دستگاه انتهایی و ۴ دروازه پیاده‌سازی شده و همچنین از پلتفرم TTN به عنوان بستری برای سرور شبکه و سرور کاربرد استفاده کرده‌ایم.

فصل ۴

طراحی و پیاده‌سازی سامانه تشخیص موقعیت مکانی با استفاده از یادگیری ماشین

۴-۱- مقدمه

در فصل‌های گذشته با مفاهیم مورد نیاز برای ایجاد یک سیستم موقعیت‌یابی و همچنین نحوه ایجاد زیرساخت شبکه لازم مبتنی بر LoRaWAN آشنا شدیم. حال در این فصل با استفاده از مفاهیم ذکر شده و همچنین زیرساخت آماده شده به پیاده‌سازی این سامانه می‌پردازیم. در این فصل در ابتدا با مجموعه دادگان استفاده شده و نحوه جمع‌آوری آن آشنا خواهیم شد، سپس پیاده‌سازی و نتیجه روش پیش‌پردازش دادگان معرفی شده در بخش ۲-۶-۱ مطرح می‌شود. در انتها الگوریتم‌های یادگیری ماشین به کار رفته برای دسته‌بندی داده‌ها معرفی و نمایش داده می‌شوند.

۴-۲- مجموعه دادگان

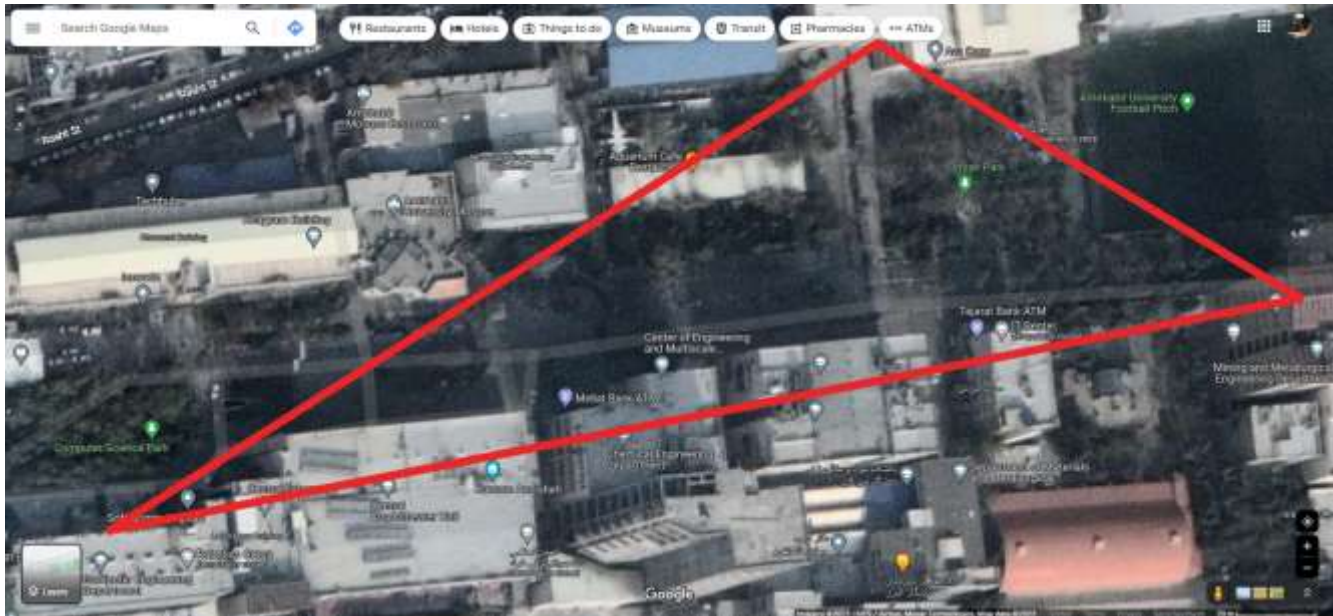
برای ایجاد سامانه تشخیص موقعیت مکانی بیش از هر چیز به یک مجموعه دادگان با تعداد داده مناسب نیاز داریم. این مجموعه دادگان شامل داده‌های جمع‌آوری شده از محوطه تحت پوشش می‌باشد که مرجعی برای تجزیه و تحلیل و همچنین ورودی الگوریتم‌های یادگیری ماشین هستند. در بخش‌های ۴-۲-۱ و ۴-۲-۲ محوطه‌ای که مجموعه دادگان در آن جمع‌آوری شده و همچنین چگونگی جمع‌آوری داده معرفی می‌شود. همچنین در بخش ۴-۲-۳ تحلیل‌های مقدماتی از داده‌ها ارائه خواهند شد.

۴-۲-۱- استقرار زیرساخت و ایجاد محوطه تحت پوشش شبکه

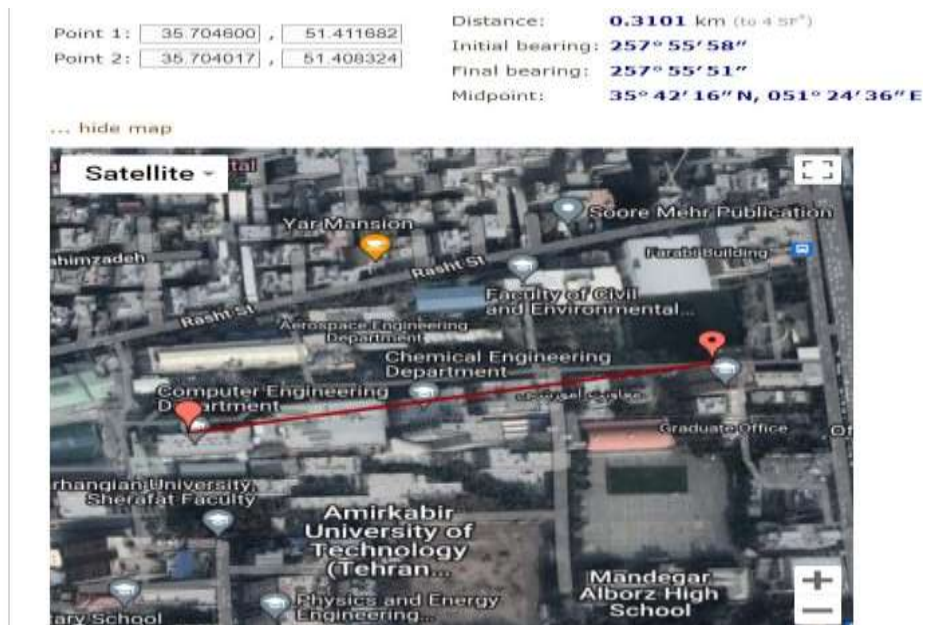
در این پروژه، فضای استفاده شده برای پیاده‌سازی زیرساخت و جمع‌آوری داده محوطه دانشگاه امیرکبیر است. با توجه به این که ۳ عدد دروازه شبکه LoRaWAN در اختیار داریم، دروازه‌ها در این فضا به شکل یک مثلث مطابق شکل ۵۴ قرار داده شدند. همانطور دیده می‌شود موقعیت قرارگیری دروازه‌ها به نحوی انتخاب شده است که بهترین پوشش ممکن از فضای باز دانشگاه ایجاد شود. رأس‌های مثلث شکل زیر محل قرارگیری دروازه‌ها هستند. دروازه‌ها در دانشکده‌های مهندسی کامپیوتر (رأس سمت چپ)، عمران (رأس وسط) و معدن (رأس سمت راست)

فصل ۴ طراحی و پیاده‌سازی سامانه با استفاده از یادگیری ماشین

قرار گرفتند. مساحت منطقه تحت پوشش با استفاده چینش مطرح شده دروازه‌ها، تقریباً ۱۲ کیلومتر مربع می‌باشد. در شکل‌های ۵۵ الی ۵۷ طول تقریبی هر ضلع این مثل را مشاهده می‌نمایید.



شکل ۵۴ محل قرارگیری دروازه‌های شبکه LoRaWAN و فضای شبکه ایجاد شده



شکل ۵۵ فاصله دروازه‌های قرار گرفته در دانشکده‌های کامپیوتر و معدن



شکل ۵۶ فاصله دروازه‌های قرار گرفته در دانشکده‌های عمران و معدن



شکل ۵۷ فاصله دروازه‌های قرار گرفته در دانشکده‌های عمران و کامپیوتر

همانطور که گفته شد الگوریتم‌های یادگیری ماشین مورد استفاده برای این پروژه از نوع یادگیری با نظارت هستند. بنابراین در هنگام جمع آوری داده علاوه بر مقادیر قدرت سیگنال دریافتی (RSSI) در دروازه‌ها، به یک مقدار هدف^۱ یا داده مرجع^۲ برای هر سیگنال نیز نیاز داریم. حال با توجه به این که هدف این پروژه موقعیت‌یابی^۳ و نه مکان‌یابی^۴ است، داده‌های مرجع موقعیت دقیق (طول و عرض جغرافیایی هر سیگنال ارسال شده) نیستند بلکه ناحیه‌ای^۵ است که سیگنال از آنجا ارسال شده است. در شکل ۵۸، نواحی حاصل از تقسیم بندی کل منطقه تحت پوشش شبکه نمایش داده شده است. هر کدام از ۷ ناحیه مشخص شده در این شکل، داده مرجع برای سیگنال‌های ارسال شده از همان ناحیه می‌باشد. برای مثال تمام ۷۰ داده ایی که از هرکجای ناحیه ۳ ارسال شدند مرجع یکسان و برابر با ۳ خواهند داشت.

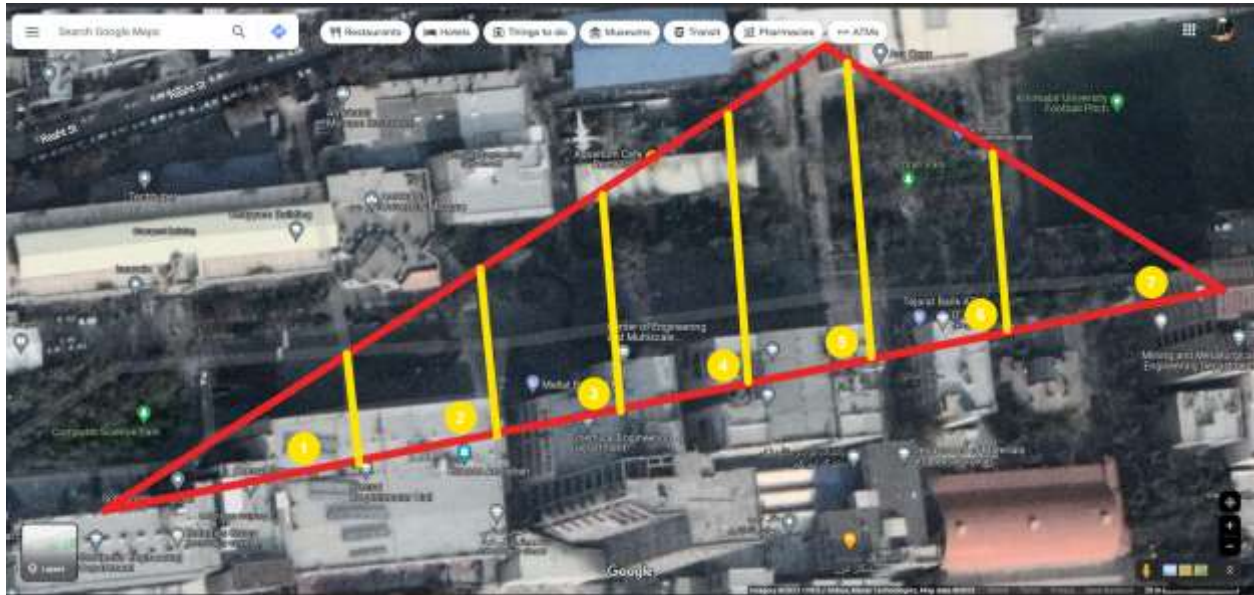
¹ Target value

² Gound-truth

³ Localization

⁴ Positioning

⁵ Region



شکل ۵۸ تقسیم بندی نواحی در منطقه تحت پوشش شبکه

۴-۲-۲- جمع‌آوری داده

پیش از این که به جمع‌آوری داده در محیط دانشگاه بپردازیم باید مشخص کنیم که به چه تعداد داده برای این پروژه نیاز داریم. این مسئله از این رو حائز اهمیت است که مجموعه دادگان ایجاد شده در انتها به مدل‌های یادگیری ماشین داده می‌شوند و اگر تعداد داده‌ها مناسب نباشد یادگیری به خوبی اتفاق نیفتاده و نتیجه به دست آمده قابل اعتماد نمی‌باشد. یک قاعده خوب و مورد استفاده برای تعداد مجموعه دادگان این است که حداقل به اندازه ۱۰ برابر تعداد ویژگی‌ها داده داشته باشیم [۲۶].

از آنجایی که در این مسئله تعداد ویژگی‌ها برابر تعداد دروازه‌ها (۳) می‌باشد و در کل ۷ ناحیه موجود است، برای هر ناحیه حداقل ۳۰ داده و در مجموع حداقل ۲۱۰ داده نیاز داریم. حال برای اینکه نتیجه به دست آمده قابل اعتمادتر باشد تعداد داده‌های جمع‌آوری شده کمی بیش از دو برابر این تعداد محاسبه شده می‌باشد. در هر ناحیه حداقل ۶۹ داده و در مجموع ۵۰۰ داده جمع‌آوری شده است. در شکل ۵۹ نمای کلی از مجموعه داده ایجاد شده را مشاهده می‌نمایید. ستون region مشخص کننده نواحی شکل ۵۸ است. همچنین ستون‌های civil, comp و

mining به ترتیب مربوط به قدرت سیگنال دریافتی از دروازه‌های قرار گرفته در دانشکده‌های مهندسی کامپیوتر، عمران و معدن می‌باشد.

| | region | comp | civil | mining |
|-----|--------|------|-------|--------|
| 0 | 1 | -72 | -120 | -95 |
| 1 | 1 | -79 | -120 | -120 |
| 2 | 1 | -83 | -91 | -97 |
| 3 | 1 | -75 | -88 | -120 |
| 4 | 1 | -75 | -89 | -120 |
| ... | ... | ... | ... | ... |
| 495 | 7 | -120 | -84 | -83 |
| 496 | 7 | -120 | -81 | -80 |
| 497 | 7 | -97 | -83 | -79 |
| 498 | 7 | -120 | -85 | -75 |
| 499 | 7 | -120 | -82 | -77 |

شکل ۵۹ مجموعه داده جمع‌آوری شده.

همانطور که در فصل قبل اشاره شد، دستگاه انتهایی پیاده‌سازی شده در هر ۳۰ ثانیه یک بار داده ارسال می‌کند، با توجه به این نکته برای جمع‌آوری ۷۰ داده از هر ناحیه ۳۵ دقیقه و در مجموع برای جمع‌آوری داده از تمام نواحی تقریباً ۴ ساعت زمان صرف شد.

در هنگام جمع‌آوری داده‌ها در بعضی نقاط سیگنال ارسال شده از دستگاه انتهایی به یک یا دو دروازه نمی‌رسد. این پدیده دلایل متعددی دارد، مهم‌ترین آن نبود خط دید مستقیم بین دروازه و دستگاه انتهایی و همچنین قرار داشتن موانع مختلف مانند ساختمان دانشکده‌ها در مسیر سیگنال ارسال شده از دستگاه انتهایی است. در صورت بروز این اتفاق (که در مرحله جمع‌آوری داده متداول بود) مقدار داده مربوط در مجموعه دادگان ۱۲۰- قرار داده شده است. بنابراین مقدار ۱۲۰- به معنی عدم اتصال است. همانطور که در شکل ۵۹ مشخص است، در سطرهای ۱، ۳ و ۴ که مربوط به ناحیه ۱ می‌باشد مقادیر دروازه مستقر در دانشکده معدن ۱۲۰- می‌باشد.

۴-۲-۳- تحلیل داده‌ها

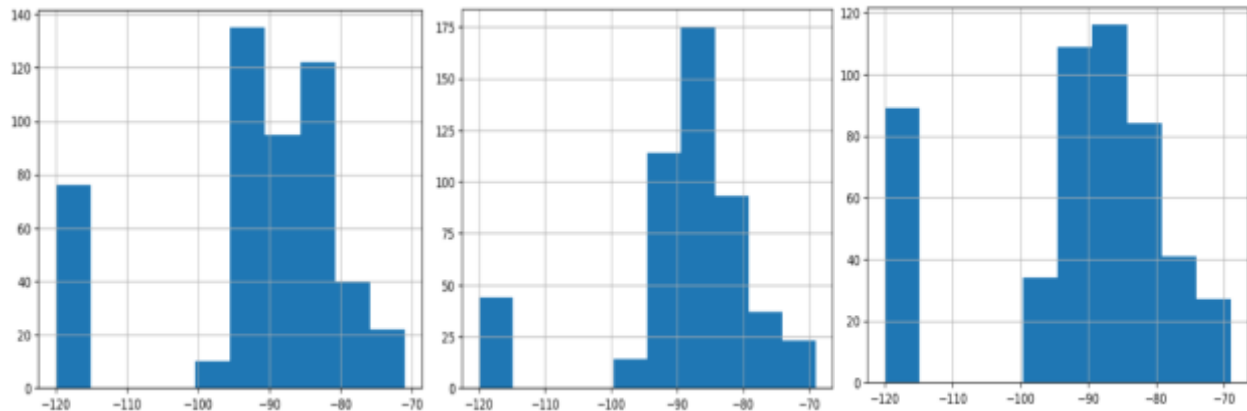
برای انجام وظایف^۱ یادگیری ماشین باید به خوبی با داده‌ها و ویژگی‌های کلیدی آن آشنا باشیم و بتوانیم تفسیرهای معنی‌داری از داده‌ها ارائه کنیم. مطالعه و تحلیل ویژگی‌هایی همچون میانگین داده‌ای دریافت شده از هر دروازه، میانگین داده‌ها در هر ناحیه، تاثیر نویز بروی داده‌ها، نمودارهای هیستوگرام و غیره، می‌توانند در این امر بسیار تاثیرگذار باشند. حال با توجه به اهمیت تحلیل داده‌ها، در این بخش به ارائه تحلیل‌ها و ویژگی‌های مهم مجموعه دادگان ایجاد شده می‌پردازیم.

مقایسه کیفیت داده‌ای دریافت شده در دروازه‌ها: شکل ۶۰ نمایشی از هیستوگرام برای ۳ دروازه کامپیوتر (سمت راست)، عمران (وسط) و معدن (سمت چپ) می‌باشد. با توجه به این نمودارها میتوان متوجه شد دروازه مستقر در دانشکده عمران بهترین وضعیت اتصال را داشته است. کمترین تعداد عدم اتصال (مقدار ۱۲۰-) با ۴۰ تکرار از مجموع ۵۰۰ داده در این دروازه مشاهده شده در صورتی که در دروازه دانشکده معدن نزدیک به دو برابر (حدود ۷۰ بار) و در دروازه دانشکده کامپیوتر بیش از دو برابر (نزدیک به ۹۰) بار شاهد عدم اتصال و نرسیدن سیگنال به دروازه بوده‌ایم. این پدیده با توجه به شکل ۵۸ قابل توجیه است زیرا که دروازه دانشکده عمران نسبت به ۲ دروازه دیگر وسط‌تر قرار گرفته و به طور میانگین به نواحی ۷ گانه نزدیک تر است.

با توجه به شکل ۶۰ کیفیت اتصال دروازه دانشکده معدن تقریباً نزدیک به دروازه دانشکده کامپیوتر است. اما مقدار عدم اتصال‌های دروازه دانشکده کامپیوتر به طور واضحی بیشتر از دروازه دانشکده معدن است. برای توجیه این پدیده باید به محل دقیق قرارگیری دقیق این دروازه‌ها توجه کنیم. دروازه دانشکده معدن در طبقه ۷ این ساختمان با ارتفاع حدودی ۲۱ متر قرار گرفته است. این در حالی است که دروازه دانشکده کامپیوتر در طبقه ۳ و ارتفاع حدودی ۱۰ متر قرار دارد. با توجه به ارتفاع بیشتر دروازه دانشکده معدن، فضای بیشتری وجود دارد که دستگاه انتهایی خط دید مستقیم با دروازه داشته باشد در صورتی که دانشکده ارتفاع محل قرارگیری دروازه دانشکده کامپیوتر از ساختمان مجاور (تالار مرکزی) کمتر است که همین امر باعث ایجاد اختلال زیادی بروی سیگنال‌های دریافتی توسط این دروازه می‌شود.

¹ Task

حال اگر از مقادیر ۱۲۰- در این دروازه‌ها چشم پوشی کنیم کیفیت اتصال دروازه کامپیوتر کمی بهتر از دروازه دانشکده معدن است. دلیل این امر نیز باز به ارتفاع قرار گیری این دروازه‌ها مربوط است. یک سیگنال برای رسیدن به دروازه دانشکده کامپیوتر مسافت کمتری را طی می‌کند و همین امر باعث می‌شود که نویز کمتری بر روی بیافتد.



شکل ۶۰ هیستوگرام داده‌های دریافتی دروازه‌ها، از سمت راست، دانشکده‌های کامپیوتر، عمران و معدن

حال به مقایسه میانگین^۱ قدرت سیگنال‌های دریافت شده و اختلاف از معیار این مقادیر می‌پردازیم. همانطور که در جدول ۴ مشاهده می‌شود، میانگین مقادیر دریافت شده در دروازه دانشکده عمران از باقی داروازه‌ها بهتر است. این درحالی است که بدترین میانگین مربوط به دروازه دانشکده کامپیوتر است که با توجه به تحلیل‌های انجام شده بر روی نمودار هیستوگرام قابل پیش‌بینی بود. انحراف از معیار مقادیر دروازه دانشکده عمران نیز به طور قابل توجهی از باقی دروازه‌ها کمتر است که با توجه به شکل ۶۰ و کمتر بودن تعداد اتصال‌های ناموفق این دروازه قابل توجه است.

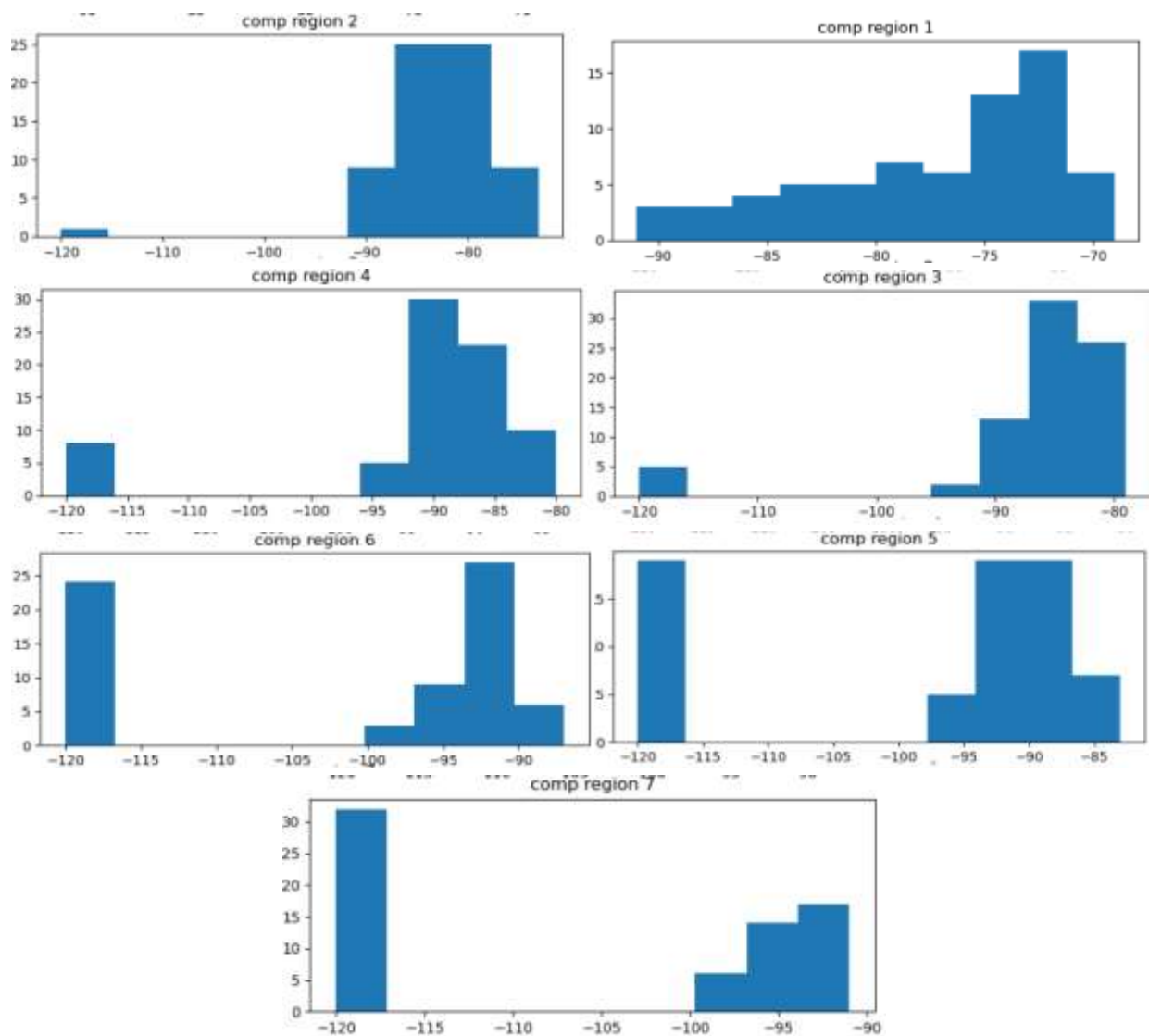
جدول ۴ معیارهای ارزیابی مقادیر قدرت سیگنال دریافتی دروازه‌ها

¹ Mean

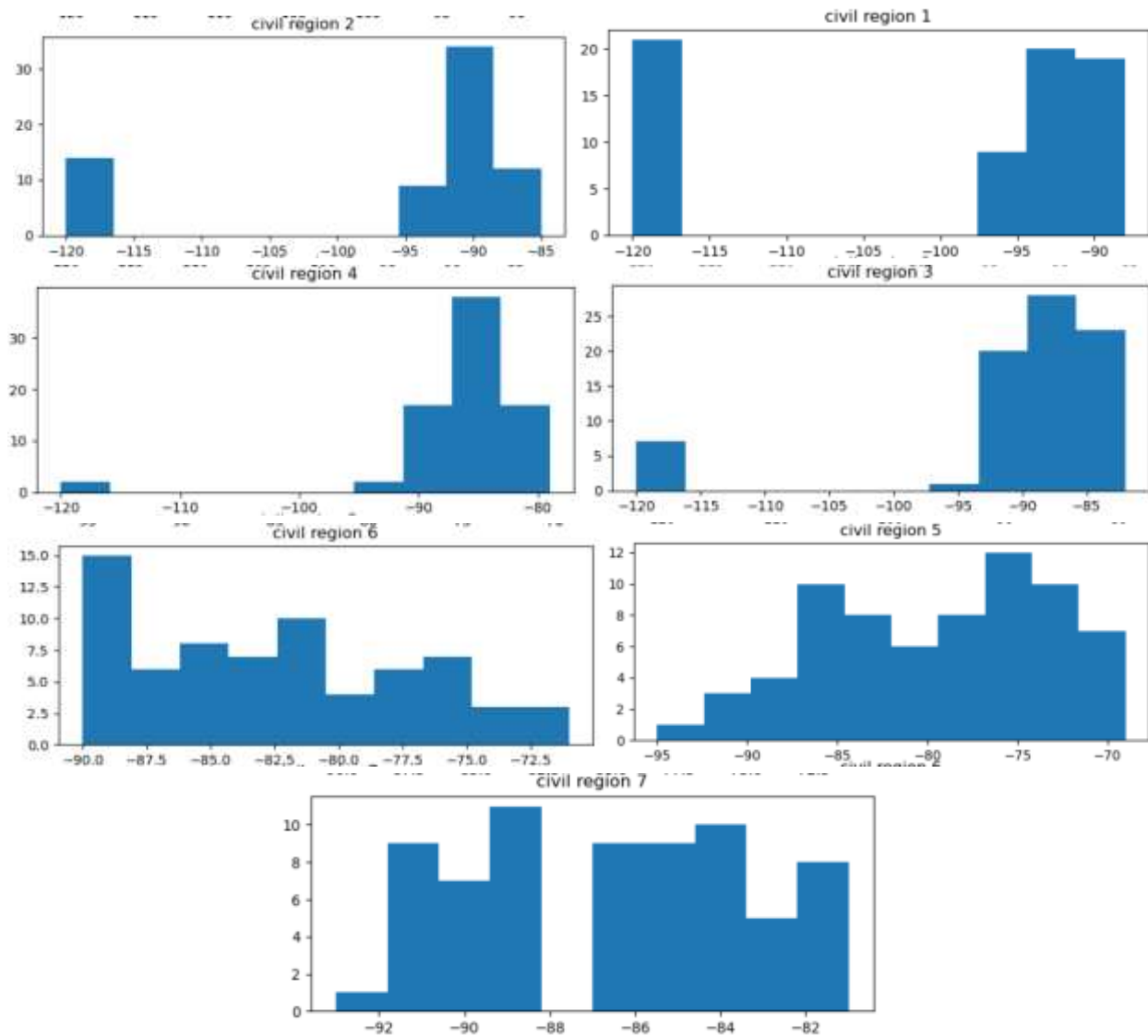
| | region | comp | civil | mining |
|-------|------------|-------------|-------------|-------------|
| count | 500.000000 | 500.000000 | 500.000000 | 500.000000 |
| mean | 3.980000 | -92.062000 | -88.830000 | -91.878000 |
| std | 1.972659 | 14.254562 | 11.078021 | 13.110411 |
| min | 1.000000 | -120.000000 | -120.000000 | -120.000000 |
| 25% | 2.000000 | -94.000000 | -91.000000 | -93.000000 |
| 50% | 4.000000 | -89.000000 | -87.000000 | -89.000000 |
| 75% | 6.000000 | -83.000000 | -84.000000 | -83.000000 |
| max | 7.000000 | -69.000000 | -69.000000 | -71.000000 |

مقایسه کیفیت اتصال دروازه‌ها در نواحی مختلف: در این بخش به مقایسه یک‌به‌یک کیفیت اتصال دروازه‌ها در دانشکده‌های کامپیوتر و عمران در تمامی نواحی می‌پردازیم. دلیل عدم استفاده از دروازه دانشکده معدن در این بخش این است که تفاوت کیفیت دروازه کامپیوتر با عمران معنی‌دارتر از تفاوت کیفیت دروازه‌های کامپیوتر و معدن باهم هستند. با توجه به هیستوگرام‌های شکل ۶۱ می‌توان مشاهده کرد که به جز ناحیه ۱ در تمامی نواحی دیگر اتصال ناموفق بین دستگاه انتهایی و دروازه دانشکده کامپیوتر وجود دارد. این در حالی است که با مشاهده شکل ۶۲ در سه ناحیه ۵، ۶ و ۷ اتصال‌ها با دروازه دانشکده عمران کاملاً موفق بودند و از کیفیت خوبی نیز برخوردار هستند.

نکته قابل توجه دیگری که در شکل ۶۱ وجود دارد این است که در ناحیه ۷ تعداد اتصال‌های ناموفق تقریباً برابر با اتصال‌های برقرار شده است. همچنین مقادیر اتصال‌های برقرار شده نیز از کیفیت بسیار کمی (قدرت سیگنال دریافتی) برخوردار هستند.

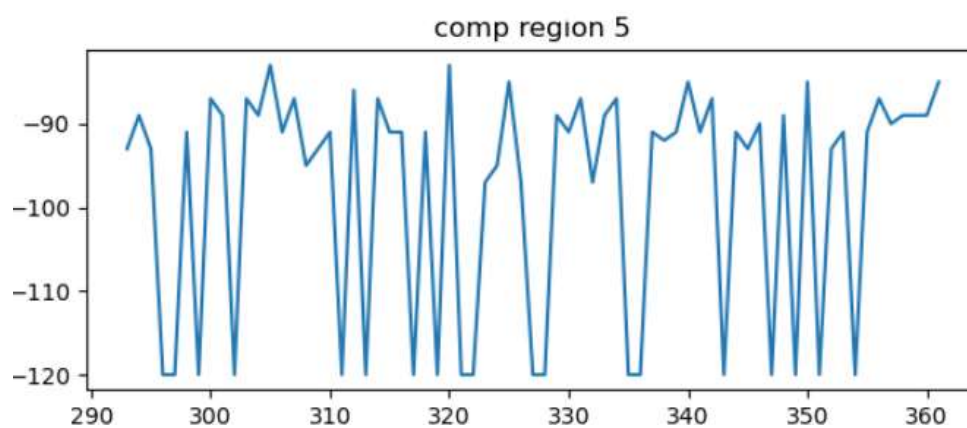


شکل ۶۱ هیستوگرام داده‌های دریافتی توسط دروازه دانشکده کامپیوتر در تمام نواحی

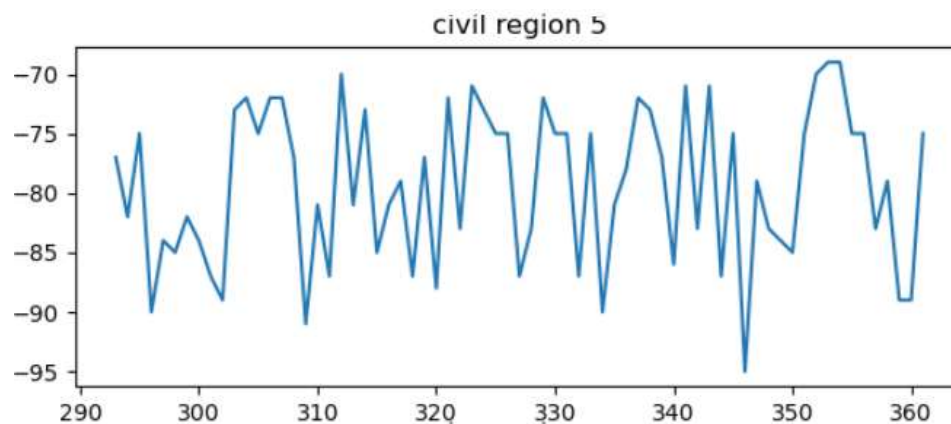


شکل ۶۲ هیستوگرام داده‌های دریافتی توسط دروازه دانشکده عمران در تمام نواحی

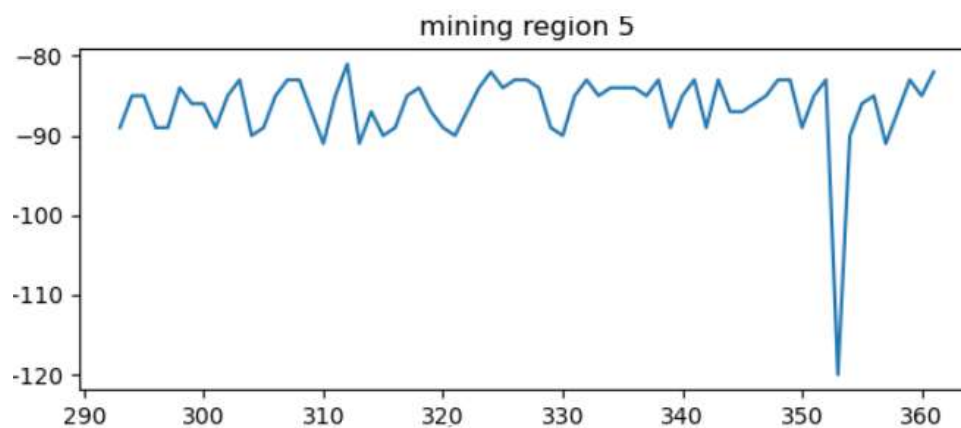
تاثیر نویز بر روی داده‌ها: همانطور که بارها اشاره شد، داده‌های مورد استفاده ما در این پروژه مقادیر قدرت سیگنال دریافتی یا RSSI هستند. این داده‌ها بسیار نویز پذیر بوده و حتی در صورت ثابت نگاه داشتن موقعیت دستگاه انتهایی و دروازه شبکه، مقادیر آن‌ها می‌توانند پرش زیادی داشته باشند. برای اثبات این ادعا در شکل‌های ۶۳ الی ۶۵ مقادیر RSSI دریافتی هر سه دروازه در ناحیه ۵ نمایش داده شده است.



شکل ۶۳ تاثیر نویز بر داده‌های جمع‌آوری شده دروازه دانشکده کامپیوتر از ناحیه ۵



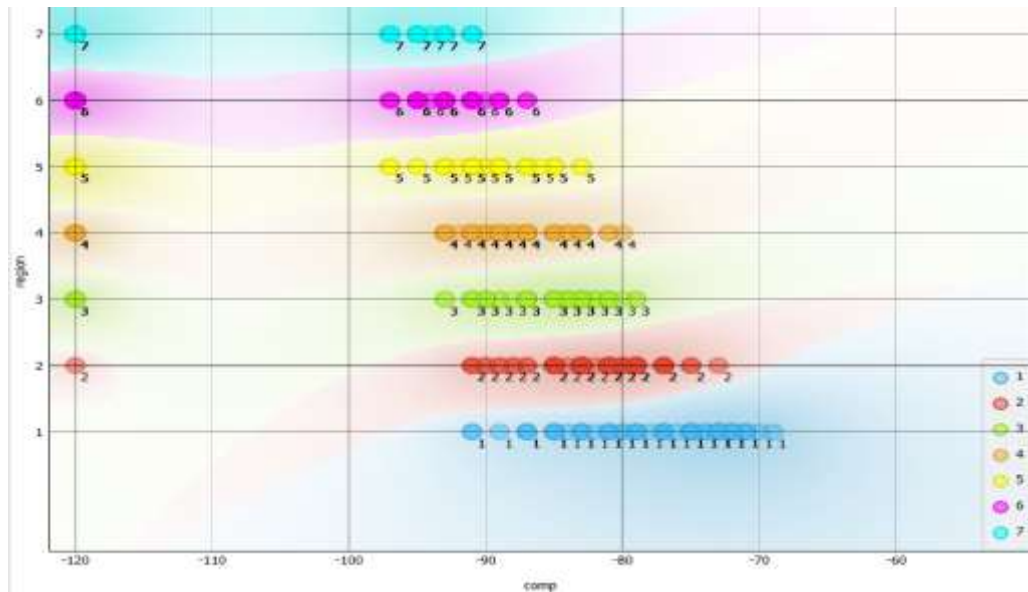
شکل ۶۴ تاثیر نویز بر داده‌های جمع‌آوری شده دروازه دانشکده عمران از ناحیه ۵



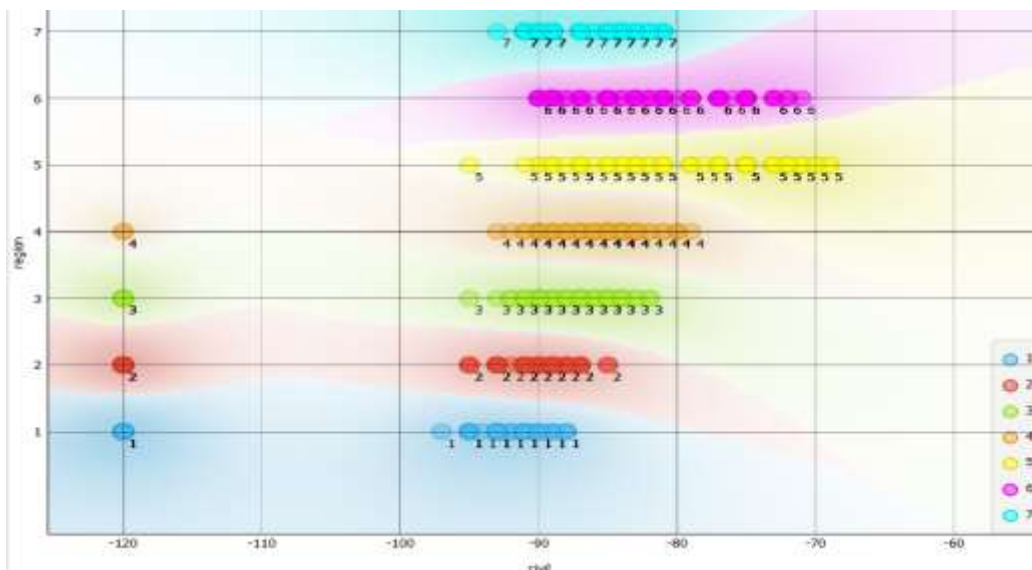
شکل ۶۵ تاثیر نویز بر داده‌های جمع‌آوری شده دروازه دانشکده معدن از ناحیه ۵

فصل ۴ طراحی و پیاده‌سازی سامانه با استفاده از یادگیری ماشین

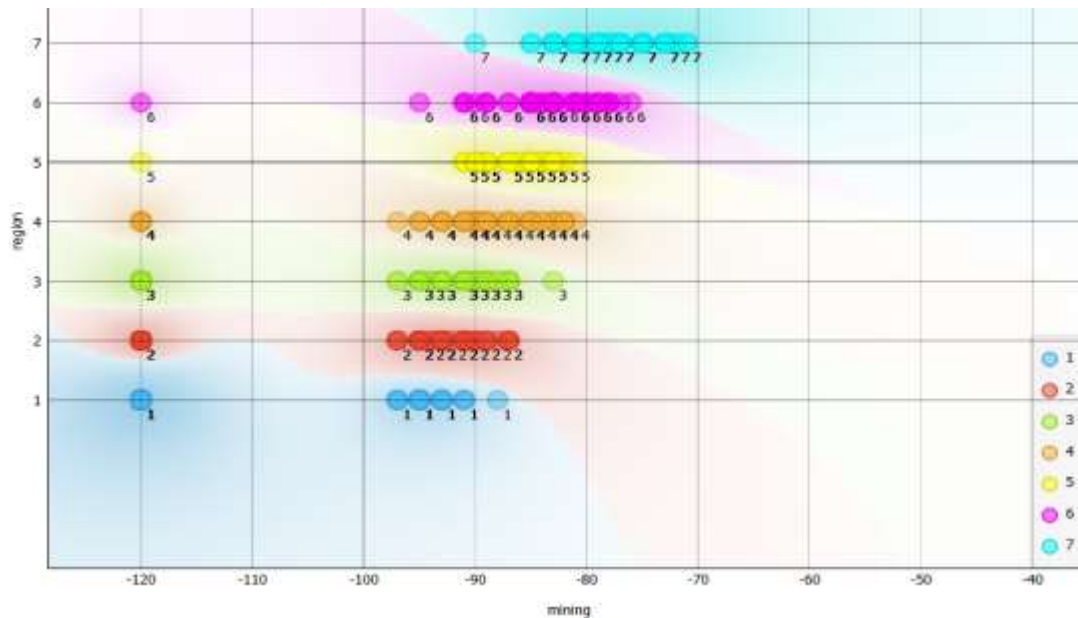
همانطور که دیدیم، تاثیر نویز بر روی داده‌ها بسیار زیاد می‌باشد. اما برای اطمینان و اثبات بیشتر این ادعا مقادیر مشاهده شده در دروازه در نواحی مختلف را با خود آن دروازه مقایسه می‌کنیم. در شکل‌های ۶۶ لی ۶۸ مقادیر مشاهده شده در هر دروازه در نواحی مختلف نمایش داده شده‌اند.



شکل ۶۶ داده‌های دریافتی دروازه دانشکده کامپیوتر در تمام نواحی



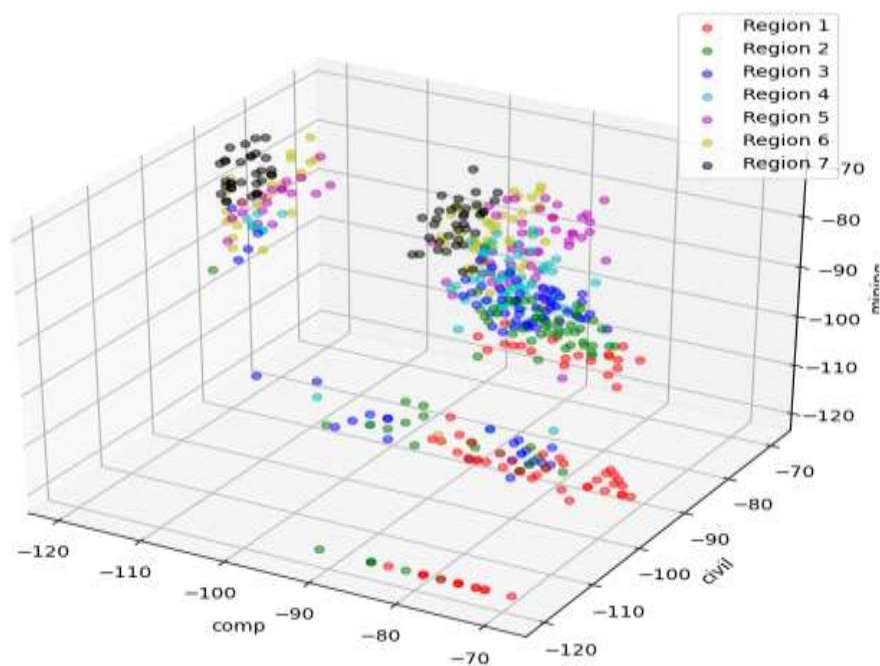
شکل ۶۷ داده‌های دریافتی دروازه دانشکده عمران در تمام نواحی



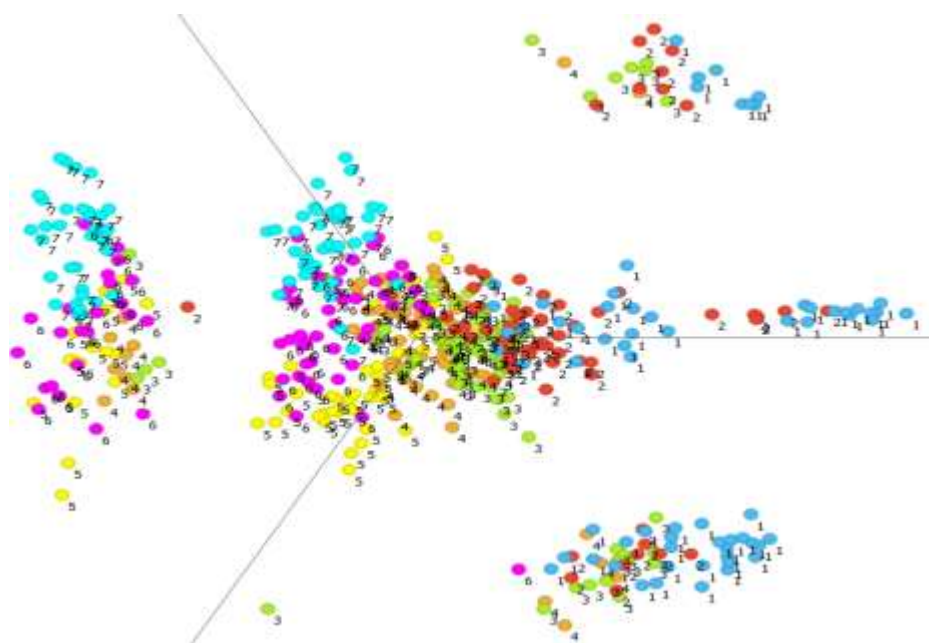
شکل ۶۸ داده‌های دریافتی دروازه دانشکده معدن در تمام نواحی

تحلیل‌های مهمی می‌توان با توجه به شکل‌های ۶۶ الی ۶۸ ارائه داد. اولین تحلیل این است که ضعیف‌ترین مقدار سیگنال دریافتی مشاهده شده قبل از عدم اتصال ۹۷- می‌باشد. دوم این‌که در تمامی نواحی و تمامی دروازه‌ها بین مقادیر RSSI مشاهده شده اشتراک وجود دارد. برای مثال در دروازه دانشکده عمران همان مقداری که در ناحیه ۱ مشاهده شده‌اند در هر ۶ ناحیه ۲ الی ۷ نیز وجود دارند. این امر می‌تواند دسته‌بندی داده‌ها را برای تکنیک‌های یادگیری ماشین سخت کند.

حال که با مشخصات مجموعه داده‌گان به طور کامل آشنا شدیم می‌توانیم نگاهی به پراکندگی تمام ۵۰۰ داده جمع‌آوری شده برحسب ناحیه و دروازه داشته باشیم. شکل ۶۹ این پراکندگی را به صورت سه‌بعدی و شکل ۷۰ به صورت خطی نمایش می‌دهد. همانطور که در هر دو شکل مشخص است مقادیر RSSI تداخل زیادی باهم دارند به طوری که با چشم هرگز نمی‌توان دسته‌بندی مناسبی برای تمییز دادن داده‌ها ارائه کرد. این تحلیل گواهی بر این است که برای به دست آوردن نتیجه بهتر نیاز به فیلتر کردن داده‌ها و کم کردن تاثیر نویز داریم.



شکل ۶۹ پراکندگی مجموعه‌دادگان به صورت سه‌بعدی



شکل ۷۰ پراکندگی مجموعه‌دادگان به صورت خطی

۴-۳- اعمال فیلتر بر روی مجموعه دادگان

به طور معمول در اکثر مسائل مربوط به هوش مصنوعی به خصوص داده‌کاوی^۱، یکی از مهم‌ترین فعالیت‌ها پیش پردازش داده‌ها است. به دلیل اهمیت این موضوع بسیاری از شرکت‌ها و سازمان‌ها منابع و هزینه زیادی برای توسعه روش‌های بهتر و پیشرفته‌تر برای پیش پردازش مجموعه دادگان خود صرف می‌کنند. پیش پردازش دادگان فرآیندی است که بعد از جمع‌آوری داده و قبل از اعمال الگوریتم‌ها انجام می‌شود. در کل می‌توان فرآیند پیش پردازش داده را به دو دسته کلی تقسیم کرد. ۱. پاک‌سازی داده^۲. کاهش عدم قطعیت و تاثیر نویز بر روی داده‌ها با استفاده از فیلتر.

پاک‌سازی داده‌ها فرآیند اصلاح یا حذف داده‌های نادرست، خراب، تکراری، ناقص و یا با فرمت نادرست در یک مجموعه داده است. در مقابل برخی داده‌ها نیاز به پاک‌سازی ندارند، اما عدم قطعیت‌های موجود در محیط و یا دستگاه و نویزهای محیطی بر روی مقدار آن‌ها تاثیر می‌گذارد. در این موارد باید با استفاده از تکنیک‌های مناسب، عدم قطعیت و تاثیر نویز را بر روی داده‌ها کاهش داد تا بتوان در انتها به نتیجه مناسب‌تر و واقعی‌تری رسید. در کل نوع مجموعه داده و وظیفه مورد نظر در انتخاب هر کدام از این دو روش تاثیرگذار هست. با توجه به شرایط می‌توان از تنها یک یا هر دو روش‌ها با هم استفاده نمود.

در این پروژه وزن بیشتر پیش پردازش داده بر روی دسته دوم است. تنها کاربرد مورد نیاز پاک‌سازی داده برای مجموعه دادگان جمع‌آوری شده در این پروژه جایگزین کردن مقدار ۱۲۰- در صورت عدم دریافت سیگنال دستگاه انتهایی در دروازه است که در بخش ۴-۲-۲ به طور کامل توضیح داده شده است. در ادامه به توضیحات مربوط به پیاده‌سازی یک فیلتر برای کاهش عدم قطعیت و تاثیر نویز و همچنین نتایج به دست آمده از آن می‌پردازیم.

¹ Data Mining

² Data cleaning

۴-۳-۱- پیاده‌سازی فیلتر

فیلتر مورد استفاده و پیاده‌سازی شده در این پروژه یک نسخه ساده از فیلتر کالمن گسسته به نام $\alpha\text{-}\beta\text{-}\gamma$ است. مبانی و معادلات به کار رفته در این فیلتر به طور کامل در بخش ۲-۶-۱ شرح داده شده است. معادله ۲ پایه اصلی این فیلتر می‌باشد. همچنین در شکل ۱۶ فلوچارتی نمایش داده شده است مبنای پیاده‌سازی کد این فیلتر می‌باشد.

برای پیاده‌سازی این فیلتر ابتدا باید بدانیم تقسیم‌بندی داده‌ها به چه صورت است و کدام داده‌ها با هم فیلتر می‌شوند. همانطور که می‌دانیم در این پروژه ۳ عدد دروازه و ۷ ناحیه وجود دارد (برای مثال در شکل‌های ۶۳ الی ۶۵ مقادیر داده‌ها را برای هر سه دروازه در ناحیه ۵ مشاهده کردیم). حال برای تقسیم‌بندی باید توجه کنیم که مقادیر دریافت شده توسط هر کدام از دروازه‌ها در هر کدام از ناحیه‌ها را باید در یک دسته قرار داده و فیلتر را اعمال کنیم. بنابراین در مجموع ۲۱ دروازه-ناحیه وجود دارد. در شکل زیر تابعی را مشاهده می‌نمایید که تقسیم‌بندی گفته شده را بر روی مجموعه داده‌گان اعمال می‌کند و این ۲۱ دروازه-ناحیه را درون لیستی ۲ بعدی (بعد اول تعداد دروازه‌ها و بعد دوم تعداد نواحی) قرار می‌دهد.

```
# returns all the gateway-region RSSI values. all the RSSI values of each gateway in each region
# returns a 2D list [3][7]
# the first dimension are gateways(comp, civil, mining). (the gateways are also the columns of data set and main features of this)
# second dimension are the 7 regions
def get_regionGateway_Info(df):
    gateways = []
    comp = []
    civil = []
    mining = []
    for i in range(1,8):
        comp.append(df[df["region"] == i]["comp"])
        civil.append(df[df["region"] == i]["civil"])
        mining.append(df[df["region"] == i]["mining"])
    gateways.append(comp)
    gateways.append(civil)
    gateways.append(mining)
    return gateways
```

شکل ۷۱ پیاده‌سازی تابع تقسیم‌بند مقادیر دروازه-ناحیه

فیلتر $\alpha\text{-}\beta\text{-}\gamma$ نیازمند یک مقداردهی یا حدس اولیه برای شروع فرآیند می‌باشد. این مقدار در واقع اولین پیش‌بینی انجام شده در طی مراحل این الگوریتم است. در این پیاده‌سازی منطقی‌ترین مقدار اولیه، میانگین هر کدام از دروازه-ناحیه‌های مورد نظر است. به این منظور نیاز به تابعی وجود دارد که مقادیر میانگین هر کدام از

این ۲۱ دروازه-ناحیه را همچون تابع آمده در شکل ۷۱، در قالب لیستی ۲ بعدی نمایش دهد. بدین منظور تابع نمایش داده شده در شکل ۷۲ پیاده‌سازی شده است.

```
# display RSSI mean of all the gateways in each of the 7 regions:
def get_regions_mean():
    means = []
    comp_means = []
    civil_means = []
    mining_means = []
    for i in range(1,8):
        region_i_means = df[df["region"] == i].mean()
        comp_means.append(region_i_means.comp)
        civil_means.append(region_i_means.civil)
        mining_means.append(region_i_means.mining)
    means.append(comp_means)
    means.append(civil_means)
    means.append(mining_means)
    return means
```

شکل ۷۲ پیاده‌سازی تابع مربوط به میانگین مقادیر ۲۱ دروازه-ناحیه

اکنون بعد از پیاده‌سازی دو تابع شکل‌های ۷۱ و ۷۲ هرآنچه که برای فیلتر کردن داده‌ها طبق معادله ۲ نیاز داریم را در اختیار داریم. نکته مهم این است که مقادیر خروجی تابع شکل ۷۱ همان اندازه‌گیری‌ها^۱ یا Z_n هستند. شکل ۷۳ نحوه پیاده‌سازی این فیلتر را نمایش می‌دهد.

¹ measurements

```
def alpha_beta_gamma(z, initial_guess, number_of_values):
    x = [0] * number_of_values
    x[0] = initial_guess
    for n in range(1, number_of_values):
        if n >= 10:
            kn = 1/10
        else:
            kn = 1/n
        # z[n-1] is z[n] here
        x[n] = x[n-1] + ( kn * (z[n-1] - x[n-1]))
    return x

rssi_values = get_regionGateway_info(df)
filtered_values = []
means = get_regions_mean()
def _filter():
    filtered_rssi_values = []
    for region in range(number_of_regions):
        regional_filtered_rssi_values = []
        for gateway in range(number_of_gateways):
            initial_guess = means[gateway][region]
            z = list(rssi_values[gateway][region])
            x = alpha_beta_gamma(z, initial_guess, number_of_rows_in_region[region])
            regional_filtered_rssi_values.append(x)
        filtered_rssi_values.append(regional_filtered_rssi_values)
    return filtered_rssi_values
```

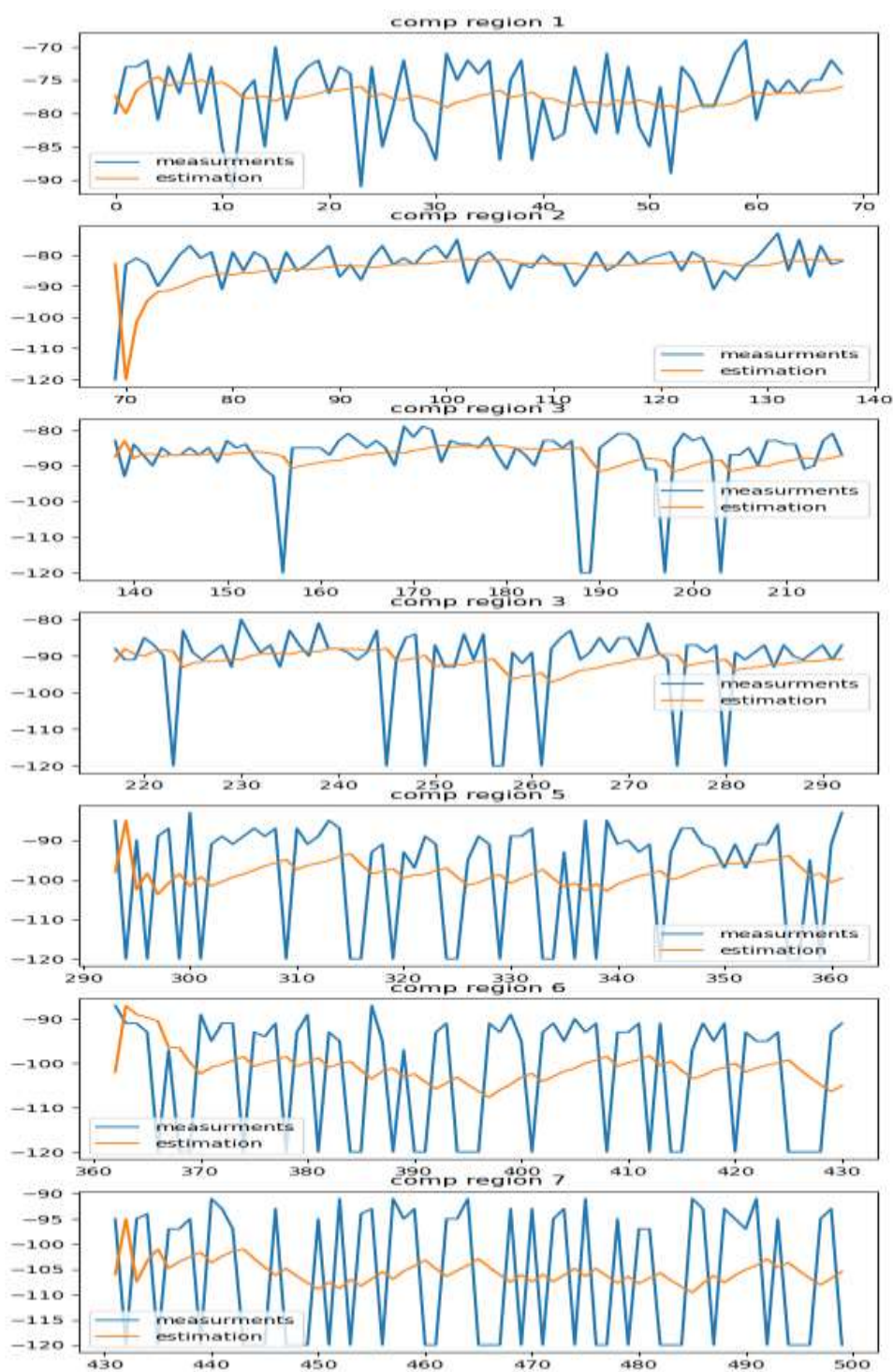
شکل ۷۳ پیاده‌سازی فیلتر α - β - γ

نکته حائز اهمیت در پیاده‌سازی صورت گرفته برای فیلتر α - β - γ این است که برخلاف معادله ۲، بهره‌کالمن (عبارت factor در شکل ۱۵ بخش ۲-۶-۱) تنها تا مشاهده دهم مقدار $\frac{1}{n}$ دارد و در تمام مشاهدات بعدی مقدار بهره ثابت و برابر $\frac{1}{10}$ است. در غیر این صورت، تاثیر تغییرات مقادیر RSSI از مشاهده ۱۰ به بعد بسیار جزئی خواهد بود که این مطلوب نیست.

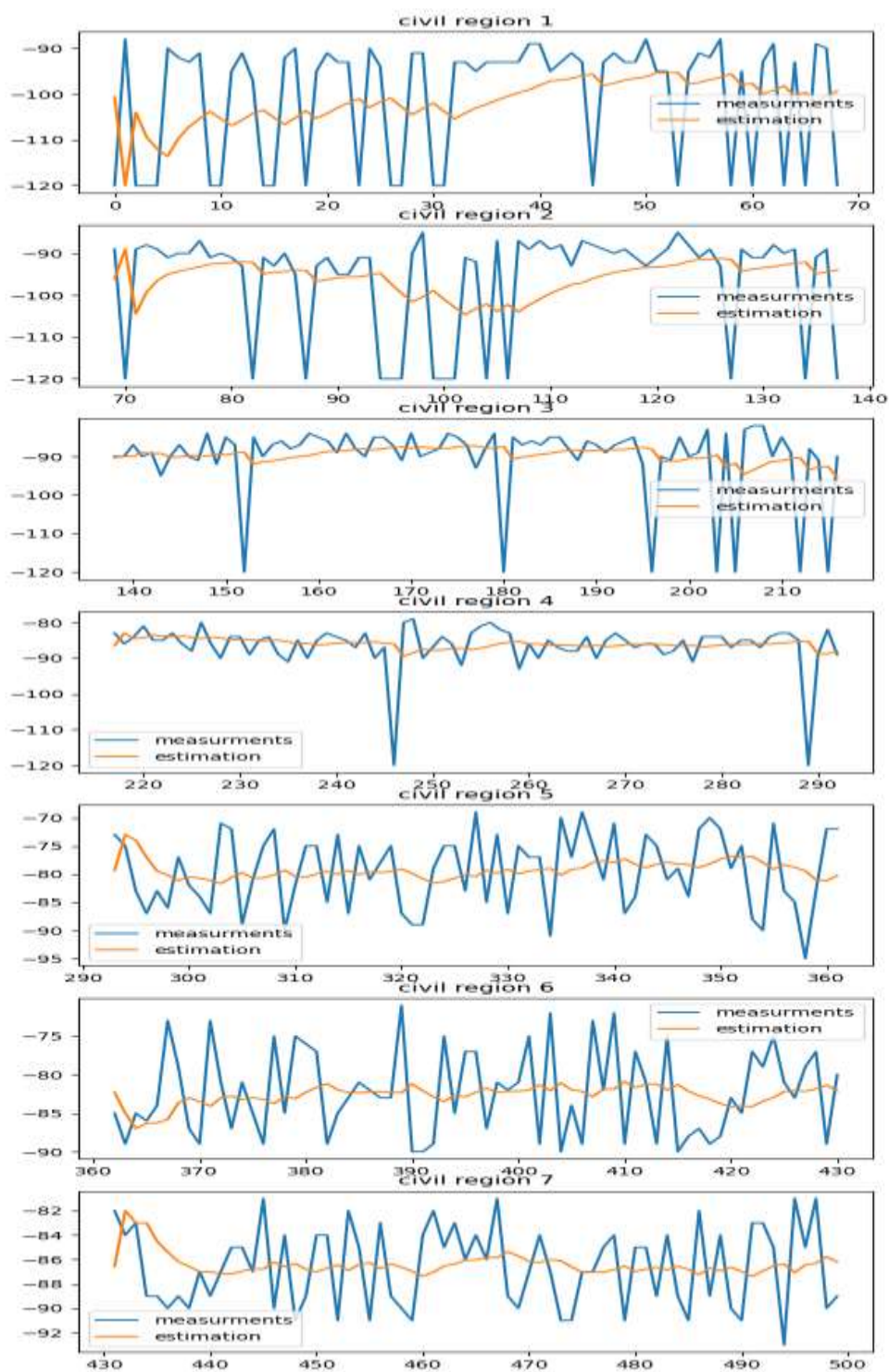
۴-۳-۲- تاثیر فیلتر بر روی داده‌ها

در بخش ۴-۲-۳، تاثیر نویز بر روی داده‌ها را در شکل‌های ۶۳ الی ۶۵ و همچنین ۶۹ و ۷۰ به خوبی مشاهده کردیم. در این بخش برای اینکه تاثیر فیلتر داده‌ها را به خوبی نمایش دهیم و همچنین مقایسه معنی‌دار و واضحی از داده‌های خام و فیلتر شده داشته باشیم، از همین شکل‌ها برای نمایش داده‌های فیلتر شده استفاده می‌کنیم. در شکل‌های ۷۴، ۷۵ و ۷۶ مقایسه کاملی از تمامی مقادیر خام و فیلتر شده در ۲۱ دروازه-ناحیه موجود را مشاهده می‌نمایید.

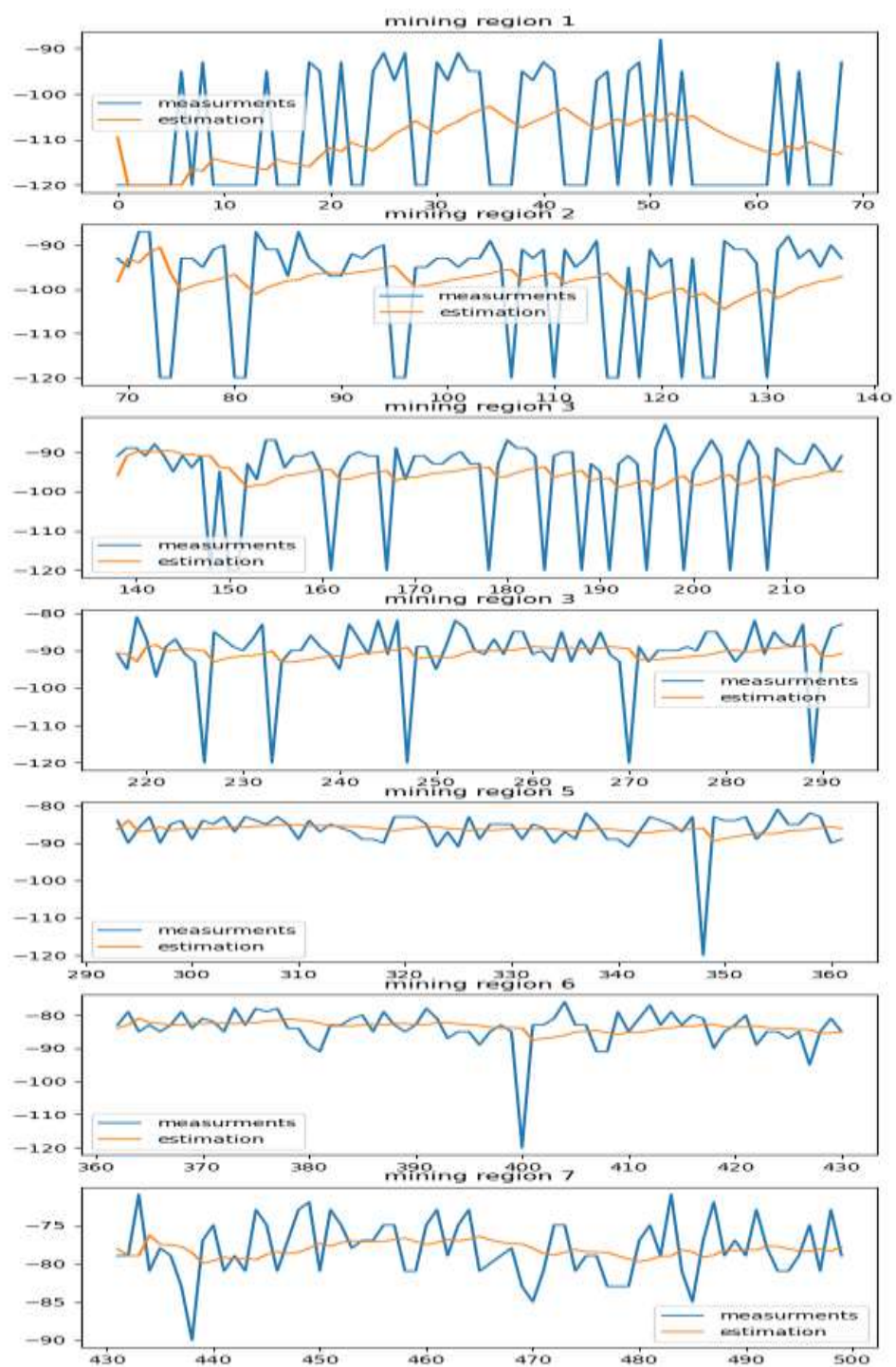
در تمام نمودارهای آمده در شکل‌های ۷۴ الی ۷۶ خط آبی نمایانگر داده‌های خام و خط نارنجی رنگ داده‌های فیلتر شده هستند. همانطور که مشخص است الگوریتم فیلتر موفق بوده است که تاثیر نویز را بسیار کاهش دهد و همچنین قدرت سیگنال‌های دریافتی را در هر ناحیه به حدی پایدار نزدیک کند.



شکل ۷۴ مقایسه داده‌های خام و فیلتر شده در ۷ ناحیه از دروازه دانشکده کامپیوتر

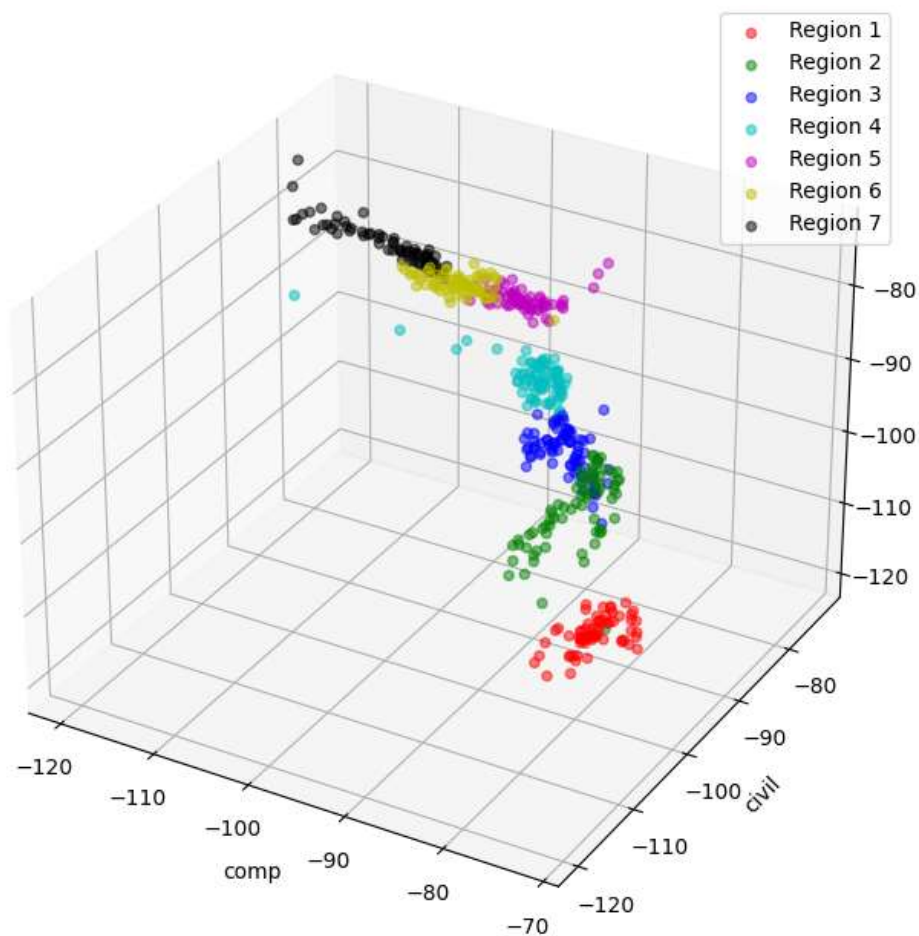


شکل ۷۵ مقایسه داده‌های خام و فیلتر شده در ۷ ناحیه از دروازه دانشکده عمران

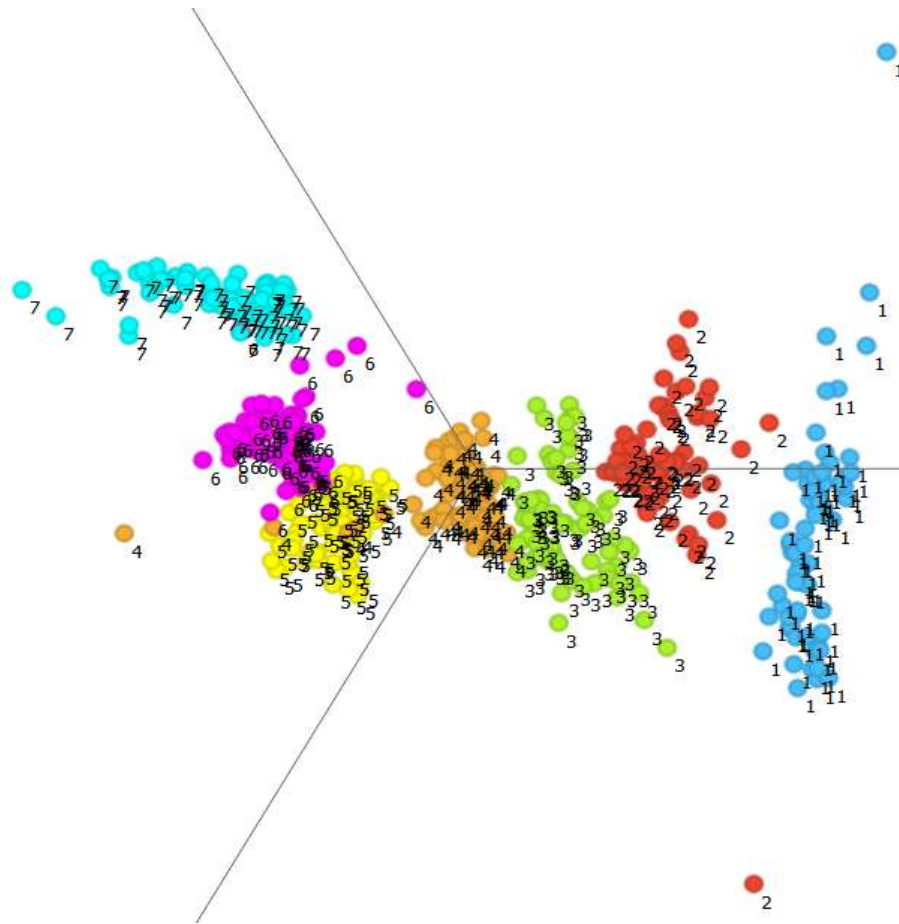


شکل ۷۶ مقایسه داده‌های خام و فیلتر شده در ۷ ناحیه از دروازه دانشکده معدن

حال مانند شکل‌های ۶۹ و ۷۰ به نمایش پراکندگی کل داده‌ها بعد از اعمال فیلتر می‌پردازیم. در شکل‌های ۷۷ و ۷۸ می‌توانید نمایش سه بعدی و خطی از این پراکندگی را مشاهده نمایید. همانطور که مشاهده می‌شود الگوریتم فیلتر پیاده‌سازی شده تاثیر فوق‌العاده زیادی بر روی کاهش نویز داشته است. با مقایسه نمودارهای شکل‌های ۶۹ و ۷۰ با شکل‌های ۷۷ و ۷۸ می‌توانیم به راحتی تشخیص دهیم که تداخل ناحیه‌ها بعد از اعمال فیلتر بسیار معنی‌دار کاهش پیدا کرده است و تمییز دادن نواحی از هم به شدت ساده تر شده است.



شکل ۷۷ پراکندگی مجموعه داده‌ها بعد از اعمال فیلتر به صورت سه بعدی



شکل ۷۸ پراکندگی مجموعه داده‌ها بعد از اعمال فیلتر به صورت خطی

۴-۴- به کارگیری یادگیری ماشین

همانطور که در فصل دوم اشاره شد، در این پروژه از چهار الگوریتم یادگیری نظارت شده برای موقعیت‌یابی استفاده می‌کنیم. این الگوریتم‌ها عبارتند از الگوریتم k نزدیک‌ترین همسایگی (KNN)، بردار پشتیبان (SVM)، XGBoost و شبکه عصبی. قطعه کد نمایش داده شده در شکل ۷۹ پیاده‌سازی این الگوریتم‌ها با استفاده از کتابخانه sklearn را نشان می‌دهد.


```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df.drop("region", axis=1), df["region"], test_size=0.2, random_state=42)

# Create the KNN model and fit it to the training data
knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train, y_train)

# Create the SVM model and fit it to the training data
svm = SVC(kernel='linear', C=1, random_state=42, probability=True)
svm.fit(X_train, y_train)

# Create the neural network model and fit it to the training data
nn = MLPClassifier(hidden_layer_sizes=(100,), max_iter=500, random_state=42)
nn.fit(X_train, y_train)

# Create the XGBoost model and fit it to the training data
xgb = XGBClassifier(random_state=42)
xgb.fit(X_train, y_train)

# Evaluate the performance of the models
models = [knn, svm, nn, xgb]
model_names = ["KNN", "SVM", "Neural Network", "XGBoost"]
# auc_scores = []
f1_scores = []
precision_scores = []
recall_scores = []
accuracy_scores = []
for i, model in enumerate(models):
    y_pred = model.predict(X_test)
    y_proba = model.predict_proba(X_test)[:, 1]

    f1 = f1_score(y_test, y_pred, average='weighted')
    precision = precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')
    accuracy = accuracy_score(y_test, y_pred)

    f1_scores.append(f1)
    precision_scores.append(precision)
    recall_scores.append(recall)
    accuracy_scores.append(accuracy)

    print(model_names[i])
    # Plot confusion matrix
    cm = confusion_matrix(y_test, y_pred)
    plt.imshow(cm, cmap=plt.cm.Blues)
    plt.colorbar()
    plt.xlabel('Predicted label')
    plt.ylabel('True label')
    plt.show()
    print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
    print("F1 Score: {:.4f}".format(f1))
    print("Precision Score: {:.4f}".format(precision))
    print("Recall Score: {:.4f}".format(recall))
    print("Accuracy Score: {:.4f}\n".format(accuracy))

# Create a table of scores
table = pd.DataFrame({
    "Model": model_names,
    "F1": f1_scores,
    "Precision": precision_scores,
    "Recall": recall_scores,
    "Accuracy": accuracy_scores
})
```

شکل ۷۹ قطعه کد پیاده‌سازی الگوریتم‌های یادگیری ماشین

جمع‌بندی

در این فصل با نحوه جمع‌آوری داده و محوطه تحت پوشش شبکه معرفی شد. سپس به تحلیل و تفسیر داده‌های جمع‌آوری شده از منظرهای مختلف پرداخته و درک خوبی نسبت به مجموعه دادگان به دست آوردیم. در انتها در مورد نحوه پیاده‌سازی فیلتر برای پیش پردازش داده‌ها و تاثیر آن بر روی داده‌ها و سپس پیاده‌سازی الگوریتم‌های یادگیری نظارت شده بحث شد. در فصل ۵ تحلیل و بررسی دقیق‌تری بر روی نتایج به دست آمده در این فصل ارائه خواهد شد.

فصل ۵

ارزیابی

۵-۱- معیار ارزیابی

برای ارزیابی می‌بایست از معیارهای استاندارد که در حوزه داده‌کاوی به خصوص موقعیت‌یابی استفاده می‌شود بهره برد، بدین منظور ابتدا ماتریس درهم‌ریختگی و سپس معیارهای دقت^۱ و بازیابی^۲ معرفی می‌شود تا با آشنایی با این مفاهیم بتوانیم معیارهای دقیق‌تری برای ارزیابی مدل‌های یادگیری ماشین اعمال شده داشته باشیم.

ماتریس درهم‌ریختگی: ماتریس در هم ریختگی از بنیادی ترین روش‌ها برای ارزیابی الگوریتم‌های طبقه بندی است. به‌خصوص زمانی که در الگوریتم دسته‌بندی، تعداد عناصر هر طبقه با طبقات دیگر متفاوت باشد. پس با ایجاد این ماتریس می‌توان دقت و صحت الگوریتم طبقه بندی را به همراه انواع خطاهای پیش آمده در زمان یادگیری سنجید. این ماتریس همانند شکل ۸۲ است. همانگونه که در شکل مشاهده می‌شود، عناصر این ماتریس نمایان گر برچسب پیش بینی شده آری یا خیر بودن به همراه حقیقت درست بودن و غلط بودن است.

| | | برچسب پیش‌بینی شده | |
|------------------|------|--------------------|------|
| | | مثبت | منفی |
| برچسب شناخته شده | مثبت | TP | FN |
| | منفی | FP | TN |

شکل ۸۰ شمایی از ماتریس درهم‌ریختگی

معیارهای دقت و بازیابی: معیار دقت و معیار بازیابی دو معیار مهم در مقایسه‌ی عملکرد الگوریتم‌های دسته‌بندی هستند. معیار دقت، نسبت تعداد موارد درست تشخیص داده شده توسط سیستم به تعداد کل مواردی است که سیستم تشخیص داده است. به عبارت دیگر، این معیار نشان می‌دهد که چه مقدار از مواردی که سیستم تشخیص داده است، درست تشخیص داده شده‌اند. فرمول معیار دقت به صورت زیر است:

^۱ Precision

^۲ Recall

$$\frac{TP}{TP+FP} = \text{دقت}$$

معیار بازیابی، نسبت تعداد موارد درست تشخیص داده شده توسط سیستم به تعداد کل موارد واقعی است. به عبارت دیگر، این معیار نشان می‌دهد که چه مقدار از موارد واقعی، توسط سیستم شناسایی شده‌اند. فرمول معیار بازیابی به صورت زیر است:

$$\frac{TP}{TP+FN} = \text{بازیابی}$$

۲-۵- مقایسه نتایج به دست آمده

قطعه کد نمایش داده شده در شکل ۷۹ یک بار بر روی داده‌های خام و بار دیگر بر روی داده‌های فیلتر شده اجرا شد. در شکل ۸۰ نتایج اجرای این الگوریتم‌ها بر روی داده‌های خام و در شکل ۸۱ همین نتایج را برای داده‌های فیلتر شده مشاهده می‌نمایید.

| Model | AUC | CA | F1 | Precision | Recall |
|-------------------|-------|-------|-------|-----------|--------|
| kNN | 0.857 | 0.564 | 0.560 | 0.560 | 0.564 |
| SVM | 0.877 | 0.540 | 0.531 | 0.544 | 0.540 |
| Gradient Boosting | 0.854 | 0.496 | 0.498 | 0.503 | 0.496 |
| Neural Network | 0.864 | 0.476 | 0.457 | 0.452 | 0.476 |

شکل ۸۱ نتایج الگوریتم‌های یادگیری ماشین بر روی داده‌های خام

| Model | AUC | CA | F1 | Precision | Recall |
|-------------------|-------|-------|-------|-----------|--------|
| SVM | 0.999 | 0.973 | 0.973 | 0.973 | 0.973 |
| kNN | 0.995 | 0.968 | 0.968 | 0.968 | 0.968 |
| Gradient Boosting | 0.998 | 0.965 | 0.965 | 0.965 | 0.965 |
| Neural Network | 0.995 | 0.955 | 0.955 | 0.957 | 0.955 |

شکل ۸۲ نتایج الگوریتم‌های یادگیری ماشین بر روی داده‌های فیلتر شده

همانطور که پیش‌بینی می‌شد تفاوت نتایج بین داده‌های فیلتر شده و خام بسیار زیاد و برای الگوریتم‌های مختلف بین ۴۰ تا ۵۰ درصد می‌باشد. از جدول شکل ۸۱ می‌توان نتیجه گرفت که با استفاده از تکنیک فیلتر کردن مقادیر RSSI می‌توان تقریباً تمام داده‌ها را درست دسته‌بندی نمود. همچنین الگوریتم SVM بر روی هر دو مجموعه داده خام و فیلتر شده بهترین عملکرد را داراست.

همانطور که مشاهده کردیم، نتایج به دست آمده از اعمال مدل‌ها بر روی داده‌های فیلتر شده بسیار بهتر از داده‌های خام بوده است. اما آیا این به این معنی است که نتایج به دست آمده از مجموعه داده‌های خام قابل قبول نیست؟ برای جواب به این سوال باید بررسی کنیم که در مواقعی که الگوریتم ناحیه درست را پیش‌بینی نمی‌کند، آیا مقدار پیش‌بینی شده مقداری پرت است یا ناحیه‌ای اطراف ناحیه واقعی را پیش‌بینی کرده است. برای این کار باید ماتریس درهم ریختگی را بررسی کنیم. شکل ۸۳ ماتریس درهم ریختگی ایجاد شده بعد از اعمال الگوریتم SVM بر روی داده‌های تست را نشان می‌دهد. دلیل انتخاب این الگوریتم آن است که بهترین نتایج دسته‌بندی چه بر روی داده‌های خام و چه بر روی داده‌های فیلتر شده با استفاده از این الگوریتم به دست آمده است.

سطرهای این شکل نمایان‌گر داده‌های حقیقی و ستون‌ها نمایان‌گر پیش‌بینی‌های انجام شده است. دلیل این که تعداد داده‌های هر سطر باهم برابر نیست این است که الگوریتم جداسازی داده‌های تست و تعلیم^۱ به طور تصادفی ۲۰ درصد از مجموعه داده را برای تست انتخاب می‌کند. حال با بررسی این ماتریس می‌توان به نتایج زیر پی برد.

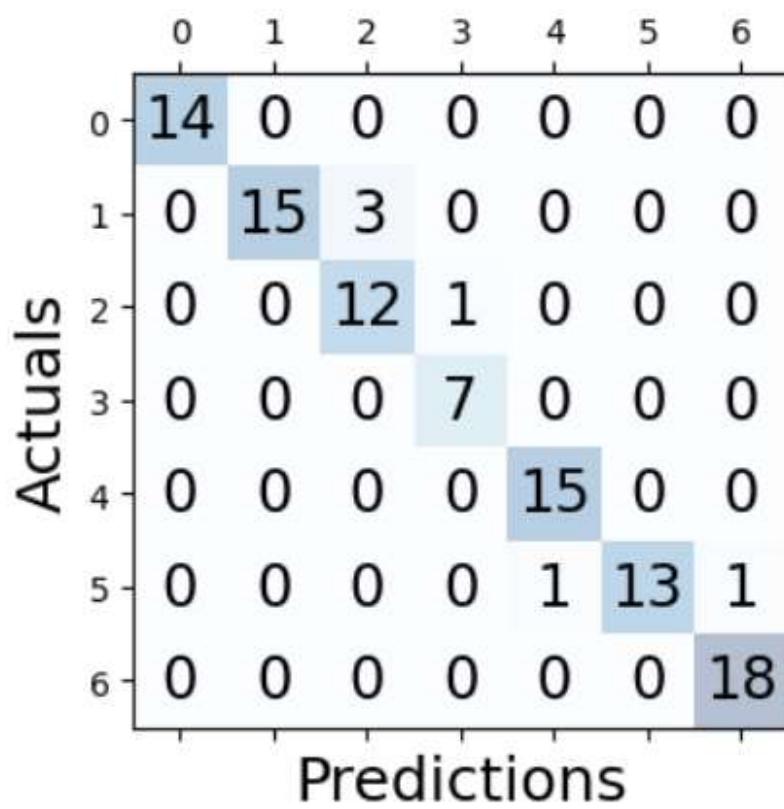
^۱ Train

ناحیه ۱ (سطر ۰) با دقت ۱۰۰ درصد پیش‌بینی شده است. در ناحیه‌های ۷، ۶، ۵ و ۲ (سطرهای ۶، ۵، ۴ و ۱) اکثر پیش‌بینی‌های اشتباه، ناحیه‌های اطراف ناحیه واقعی را نمایش می‌دهند. این امر به این معنی است که نتایج به دست آمده از این ناحیه‌ها تا حد خوبی قابل استفاده در یک سامانه موقعیت‌یابی می‌باشند. اما در ناحیه‌های ۳ و ۴ نتایج به دست از دقت خوبی برخوردار نیستند و در صورت اشتباه در پیش‌بینی ناحیه واقعی، ناحیه‌هایی با فاصله بیش از یک از ناحیه واقعی انتخاب شده‌اند. بنابراین با تحلیل این ماتریس درهم ریختگی می‌توان فهمید که نتایج بهتر از آن چیزی هستند که تصور می‌شد و تا حدی قابل اتکا می‌باشند.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|----|----|---|---|---|---|----|
| 0 | 14 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 3 | 10 | 4 | 1 | 0 | 0 | 0 |
| 2 | 0 | 4 | 1 | 8 | 0 | 0 | 0 |
| 3 | 1 | 1 | 1 | 3 | 1 | 0 | 0 |
| 4 | 0 | 0 | 0 | 3 | 7 | 5 | 0 |
| 5 | 0 | 0 | 2 | 1 | 5 | 6 | 1 |
| 6 | 0 | 0 | 0 | 1 | 0 | 5 | 12 |

شکل ۸۳ ماتریس درهم ریختگی الگوریتم SVM بر روی داده‌های خام

حال این ماتریس درهم ریختگی را برای داده‌های فیلتر شده نمایش می‌دهیم. همانطور که در شکل ۸۴ مشاهده می‌نمایید اکثر ناحیه‌ها در ست پیش‌بینی شده‌اند و حتی در صورتی که پیش‌بینی اشتباه بوده باشد ماکسیمم فاصله ناحیه پیش‌بینی شده با ناحیه واقعی ۱ می‌باشد. بنابراین اگر در سامانه مورد نظر فاصله ۱ ناحیه از ناحیه واقعی را قابل قبول بدانیم دقت مدل بعد از فیلتر کردن داده‌ها تقریباً ۱۰۰٪ می‌باشد.



شکل ۸۴ ماتریس درهم ریختگی الگوریتم SVM بر روی داده‌های فیلتر شده

فصل ۶ جمع‌بندی، نتیجه‌گیری و پیشنهادات

۶-۱- جمع‌بندی و نتیجه‌گیری

در این پروژه اینترنت اشیاء، سیستمی برای موقعیت‌یابی در فضای باز بر بستر شبکه LoRaWAN با استفاده از اندازه‌گیری قدرت سیگنال‌های دریافتی توسعه دادیم. برای ایجاد زیرساخت این پروژه سه دروازه LoRaWAN با استفاده از Raspberry Pi و RAK Wireless Multichannel LoRa Gateways و همچنین یک دستگاه‌انتهایی به عنوان فرستنده سیگنال LoRa پیاده‌سازی شد. سپس با استفاده از زیرساخت ایجاد شده و استقرار آن‌ها در مکان‌های مناسب در فضای دانشگاه امیرکبیر، بر بستر این معماری داده‌های قدرت سیگنال دریافتی را برای ایجاد یک مجموعه داده جمع‌آوری کردیم.

سپس بر روی دادگان جمع‌آوری شده روش‌های کاهش اثر نویز از قدرت سیگنال دریافتی را به همراه چهار الگوریتم یادگیری با نظارت، یعنی KNN، SVM، XGBoost و MLP برای ایجاد مدلی برای پیش‌بینی منطقه بر اساس اندازه‌گیری‌های قدرت سیگنال دریافتی استفاده شد. در نهایت با به کارگیری روش‌های پیش پردازش داده، اثر نویز بر قدرت سیگنال دریافتی به میزان قابل قبولی کاهش داده شد. مشاهده نتایج به دست آمده از اعمال الگوریتم‌های یادگیری ماشین بر روی این داده‌ها مشخص می‌کند که نیازمندی‌های مورد نظر این سامانه اقماع شده‌اند.

۶-۲- پیشنهادات

کارهای انجام شده در این پروژه می‌تواند بستر و پایه‌ای برای احداث یک سامانه موقعیت‌یابی باشد. یک سامانه موقعیت‌یابی در حله اول نیازمند یک سامانه مانند وبسایت است. این سامانه باید قابلیت‌هایی مانند ثبت نام، ثبت دستگاه‌انتهایی و نمایش موقعیت دستگاه‌های انتهایی کاربران صورت بلادرنگ را داشته باشد. علاوه بر این برای ایجاد یک سامانه واقعی چالش‌هایی وجود دارد که در این پروژه به آن‌ها پرداخته نشده است چراکه این پروژه نمایشی از امکان ایجاد سامانه موقعیت‌یابی بوده است. در ادامه برخی از چالش‌های مهم و پیشنهادات مطرح می‌شوند.

ایجاد زیرساخت: برای ایجاد یک زیرساخت مناسب برای موقعیت‌یابی نیازمند دروازه‌های بیشتری است. در این

پروژه از حداقل تعداد لازم برای پیاده‌سازی این کاربرد بهره‌برده شد اما در یک سامانه واقعی برای کوچک کردن اندازه ناحیه‌ها و همچنین پوشش داده منطقه‌ایی بزرگتر به تعداد دروازه‌های بیشتری نیاز است. همچنین مسئله مهم بعدی چالش چینش بهینه دروازه‌ها است به صورتی که یک برتری ابتدایی به کل زیرساخت تزریق کند. برای مثال باید تعیین کرد که چینش مناسب چهار عدد دروازه بر روی محیط یک مربع به این صورت است که دروازه‌ها بر روی رأس‌ها قرار بگیرند یا هر دروازه دقیقاً وسط هر ضلع مربع مستقر شود.

قوانین چرخه وظیفه و عامل گسترش: در یک سامانه موقعیت یابی بر بستر LoRaWAN دستگاه‌های انتهایی باید قوانین چرخه وظیفه مبتنی بر ۱٪ ارسال سیگنال و ۹۹٪ سکوت رادیویی را رعایت کنند. در این صورت موقعیت‌یابی دستگاه‌ها به صورت کاملاً بلادرنگ ممکن نیست و تنها به صورت دوره‌ایی می‌توان موقعیت دستگاه‌ها را شناسایی کرد. همچنین الگوریتم‌های مختلفی مانند ALOHA بر روی شبکه‌های LoRaWAN اعمال می‌شوند که وظیفه آن‌ها تغییر عامل گسترش دستگاه‌های انتهایی به منظور کاهش تداخل سیگنال‌ها در کل سیستم است. حال می‌دانیم که تغییر عامل گسترش تاثیر مستقیم بر روی قدرت سیگنال دریافتی دارد. در این صورت ممکن است سامانه موقعیت‌یابی تغییر عامل گسترش را جابه‌جایی و تغییر موقعیت تفسیر کند.

ادغام سامانه موقعیت‌یابی فضای باز با محیط‌های مسقف: یکی از مهم‌ترین ویژگی‌های LoRaWAN این است که به دلیل فرکانس پایین نفوذ سیگنالی آن بسیار زیاد است. بنابراین در تئوری می‌توان از این شبکه برای موقعیت یابی درون محیط‌های مسقف و ساختمان‌ها نیز بهره برد. همچنین می‌توان یک سامانه موقعیت‌یابی فضای باز مبتنی بر شبکه LoRaWAN را با یک سامانه موقعیت‌یابی محیط مسقف بر بستر شبکه‌های بلوتوث و وایفای ادغام کرد.

منابع

- [1] L. Labs, "A mprehensive Look At Low Power, Wide Area Networks For Internet of Things Engineers and Decision Makers," White Paper, 2016.
- [2] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "LoRa Specification 1.0," Lora Alliance Standard specification., Jan. 2015.
- [3] P. Yadav and S. Vishwakarma, "Application of Internet of Things and Big Data towards a Smart City," in 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT SIU), Bhimtal, India , 2018.
- [4] F. R. a. O. Boissier, "Smart devices embedding multi-agent technologies for a pro-active world," Uniquitous Comput. Work, 2002.
- [5] IoT Architecture Layers, [Online], Available: <https://www.hiotron.com/iot-architecture-layers>
- [6] K. Mekki, E. Bajic, F. Chaxel and F. Meyer, "A comparative study of LPWAN technologies for largescale IoT deployment," ICT Express, vol. 5, no. 1, pp. 1-7, 2019.
- [7] SigFox Networking Guide, [Online]. Available: https://www.development.libelium.com/sigfox_networking_guide/introduction
- [8] LoRaWAN Regional Parameters, [Online], Available: <https://lora-alliance.org/resource-hub/rp002-100- lorawanr-regional-parameters>
- [9] Umber Noreen, Ahcene Bounceur, Laurent Clavier, "A Study of LoRa Low Power and Wide Area Network Technology" IEEE 3rd International Conference on Advanced Technologies for Signal and Image Processing, 2017.

- [10] Jonathan de Carvalho, Silva, Joel J. P. C. Rodrigues, Antonio M. Alberti, Petar Solic, “LoRaWAN – A Low Power WAN Protocol for Internet of Things: a Review and Opportunities”, 2017.
- [11] Po-Yu Chen, Laksh Bhatia, Roman Kolcun, David Boyle, Julie A. McCann, “Contact-Aware Opportunistic Data Forwarding in Disconnected LoRaWAN Mobile Networks” , 2020.
- [12] J. Babaki, M. Rasti and R. Aslani, "Dynamic Spreading Factor and Power Allocation of LoRa Networks for Dense IoT Deployments," 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, 2020, pp. 1-6, doi: 10.1109/PIMRC48278.2020.9217283
- [13] V. Talla, M. Hesar, B. Kellogg, A. Najafi, J. R. Smith, S. Gollakota, “LoRa Backscatter: Enabling The Vision of Ubiquitous Connectivity”, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2017/9/11.
- [14] TheThingsNetwork Network Architecture, [Online]. Available: <https://www.thethingsnetwork.org/docs/network/architecture/>
- [15] "LoRaWAN specification (V1.1)," LoRa Alliance, 2017.
- [16] Shila Devalal, A. Karthikeyan, “LoRa technology-an overview”, 2018.
- [17] attenuation effects, [Online]. Available: dataloggerinc.com/wp-content/uploads/2016/11/16_Basics_of_signal_attenuation.pdf
- [18] Greg Welch, and Gary Bishop, “An Introduction to the Kalman Filter” 1995.
- [19] IOT-wireless use cases, [Online]. Available: <https://behrtech.com/blog/6-leading-types-of-iot-wireless-tech-and-their-best-use-cases/>
- [20] <https://github.com/CongducPham/LowCostLoRaGw/tree/master/PCBs>
- [21] <https://www.rakwireless.com/en-us/products/lpwan-gateways-and-concentrators/rak831>
- [22] <https://github.com/ttn-zh/ic880a-gateway>

- [23] W. Liu, X. Fu, and Z. Deng, "Coordinate-based clustering method for indoor fingerprinting localization in dense cluttered environments," *Sensors*, vol. 16, no. 12, 2016, ISSN: 1424-8220. DOI: 10.3390/s16122055. [Online]. Available: <https://www.mdpi.com/1424-8220/16/12/2055>.
- [24] X. Wang, L. Gao, S. Mao, and S. Pandey, "Csi-based fingerprinting for indoor localization: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, 2017. DOI: 10.1109/TVT.2016.2545523.
- [25] J. Luo and H. Gao, "Deep belief networks for fingerprinting indoor localization using ultrawideband technology," *International Journal of Distributed Sensor Networks*, vol. 2016, p. 18, 2016.
- [26] postindustria.com/how-much-data-is-required-for-machine-learning