

لیست پروژه‌های درس سامانه‌های نهفته

(1) پروژه‌های عملی

پروژه‌های عملی عمومی

پروژه یک) تشخیص خرابی موتور بر اساس مدل یادگیری ماشین

تولید داده از طریق سنسور شتاب سنج (ADXL345 یا MPU6050) متصل به گیربکس موتور (سخت افزار لازم تحویل داده می‌شود) در این پروژه 10 دستگاه شامل سنسور شتاب سنج متصل به گیربکس در نظر گرفته می‌شود. اما بدلیل کمبود قطعات آزمایش تولید داده را بجای 10 دستگاه، 10 بار متفاوت در یک دستگاه اجرا و داده های تولید شده را ذخیره شود. در هر بار تولید داده های برچسب دار، از 10 کلاس انواع نویز به صورت non-iid صرفا دارای 4 نوع کلاس برای هر دستگاه به صورت انتخاب تصادفی در نظر گرفته شود. منظور از کلاس های داده Pattern خطا های موجود در گیربکس هست. تزریق اشکال برای ساخت انواع خطا (نویز) به صورت مصنوعی یک Pattern به PWM پالس ارسالی به گیربکس در نظر گرفته می‌شود. ابتدا یک مدل یادگیری ماشین به دو صورت متمرکز و Federated Learning (FedAVG) آموزش داده شده و دو مدل ساخته شده را از نظر مدت زمان آموزش و دقت مورد ارزیابی قرار می‌گیرد. پردازنده مرکزی دستگاه یک ESP32 است.

• فاز Train مدل یادگیری ماشین:

سناریو 1: آموزش مدل به صورت متمرکز Central Learning

در این سناریو داده های هر 10 دستگاه در یک سرور متمرکز ارسال و سپس مدل آموزش داده می‌شود. انرژی مصرفی مورد نیاز برای آموزش محاسبه گردد.

سناریو 2: آموزش مدل به صورت یادگیری مشارکتی Federated Learning

در این سناریو آموزش مدل به صورت یادگیری مشارکتی (FL) انجام گرفته از الگوریتم FedAVG استفاده گردد. سپس مدت زمان و انرژی مصرفی و دقت مدل محاسبه گردد.

آموزش مدل از نظر مدت زمان، انرژی مصرفی، دقت مدل در دو سناریو مقایسه گردد. لازم به ذکر است مدل زمان و انرژی مصرفی ارتباطات و انتقال مدل ها یا ارسال داده های خام به سرور مورد اندازه گیری قرار گیرد.

• فاز Inference برای تشخیص خرابی

سناریو 1: قرار گیری مدل در سرور مرکزی (Cloud) - ارسال داده های خام به سرور مرکزی

سناریو 2: قرار گیری مدل در یک سرور Edge - ارسال داده های خام به سرور Edge

سناریو 3: قرار گیری مدل در خود Device - عدم ارسال داده های خام

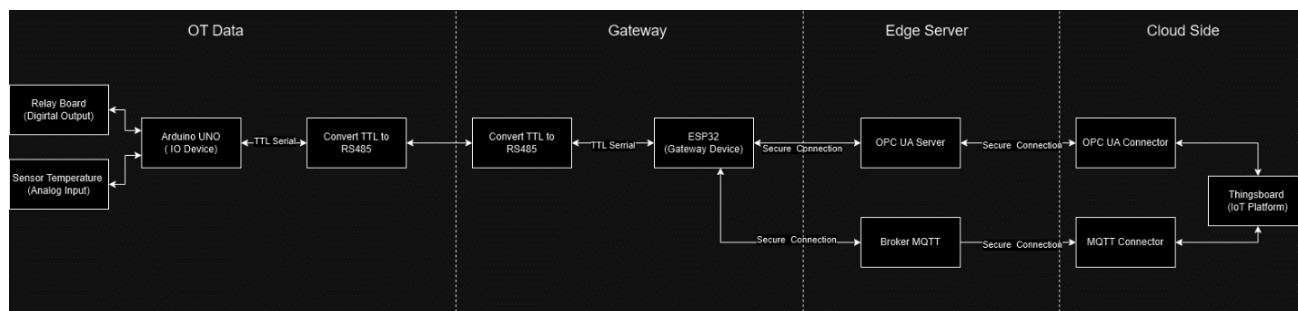
در سناریو های Inference در دو حالت کل Sample داده ارسال گردد در یک مدت زمان مشخص ارسال گردد یا با استفاده از الگوریتم FFT داده های peak تشخیص داده شود و صرفا داده های مورد نیاز با احتمال رخداد خطا ارسال گردد.

انرژی مصرفی مدت زمان دریافت پاسخ با در نظر گرفتن ارتباطات برای سناریو های Inference مورد ارزیابی قرار گیرد. همچنین جهت تست صحت عملکرد اپراتور یکی از 10 الگو تزریق فالت را اضافه می کند که می بایست دستگاه این خطا را تشخیص دهد و یک الارم صادر و نوع خطا را گزارش کند.

همچنین برای طرح امتیازی می توان تشخیص خطا به صورت Anomaly Detection تشخیص داد. طرح امتیازی دوم (در صورت وجود تیم دوم): یک سناریو برای قابلیت Predictive Maintenance پیاده سازی شود.

پروژه دو) ساخت و طراحی یک دستگاه Gateway جهت تبدیل پروتکل Modbus RTU بر روی بستر RS485 به OPC UA و MQTT جهت ارسال داده به پلتفرم Thingsboard با استفاده از ESP32

دستگاه Gateway باید با پریود زمانی 5 ثانیه داده سنسور دما را از سمت سنسور نصب شده در کارخانه بخواند و سپس مقدار آن برای پلتفرم Thingsboard ارسال کند و مقدار آن روی داشبورد نمایش دهد. لازم به ذکر است این داده از طریق بستر ارتباطی RS485 تحت پروتکل مدباس RTU دریافت و بر روی دو پروتکل و ساختار OPC UA و MQTT در دو حالت برای سرور Thingsboard به واسطه لایه Edge که یک سرور Raspberry pi می تواند باشد، ارسال کند. اجرای پروژه می بایست هردو قابلیت OPC UA و MQTT در Gateway فعال داشته باشد و قابلیت تعویض آن وجود داشته باشد. لازم به ذکر است ارتباطات بین لایه Edge و Cloud روی بستر اینترنت است. ارتباطات Gateway و Edge در بستر شبکه داخلی قرار دارد، با این حال تمامی اطلاعات از دستگاه Gateway تا سرور Thingsboard به صورت رمزنگاری شده ارسال می گردد. در سناریو دوم با ساخت ویجت های کنترلی در داشبورد، دستورات خاموش و روشن شدن برای سخت افزار Relay جهت on/off کردن ارسال گردد.



پروژه سه) طراحی سیستمی جهت لحیم خودکار قطعات DIP

تصویری سیاه و سفید مانند عکس زیر در اختیاران قرار می‌گیرد. کادر مستطیل شکلی که مشاهده می‌کنید حاشیه برد می‌باشد و همچنین نقاط سیاه رنگی که در داخل آن وجود دارد در واقع سوراخ پایه قطعات می‌باشند. ابتدا داخل کامپیوتر پردازشی بر روی تصویر صورت می‌گیرد و می‌بایست فایلی ایجاد گردد و در داخل آن در ابتدا ابعاد برد نوشته شده و در خطوط بعدی فاصله (X,Y) هر نقطه از حاشیه را بر حسب میلی متر نوشته می‌شود. سیستم شما شامل یک میکروکنترلر، دو استپر موتور، سروو موتور، یک رله و کلید می‌باشد. استپر موتورها برای حرکت در جهت X , Y می‌باشند و سروو موتور برای پایین آوردن جهت لحیم کاری، کلید برای وصل و قطع کردن و رله برای روشن و خاموش کردن دستگاه لحیم می‌باشد.

از طریق پورت سریال میکرو و PC بهم متصل می‌باشند. ابتدا می‌بایست ابعاد برد برای سیستم ارسال گردد و سپس محل هر نقطه را برای سیستم ارسال می‌شود (باید توجه داشته باشید در این امر وقتی دیتایی برای سیستم ارسال می‌شود باید دستگاه لحیم را در نقطه قرار دهد و سروو موتور بابت لحیم کاری، لحیم را به پایین ببرد و بعد از 2 ثانیه به بالا بازگردد. و سپس برای کامپیوتر تاییدی ارسال کند حال کامپیوتر می‌تواند داده بعدی را ارسال کند).

نکته: در ابتدا باید محاسبه شود چرخش 360 درجه شفت استپر موتور چه مسافتی را طی می‌کند.

نکته: ابعاد را در پردازش خود بر حسب پیکسل بدست آورده و آن را به میلی متر تبدیل کنید.

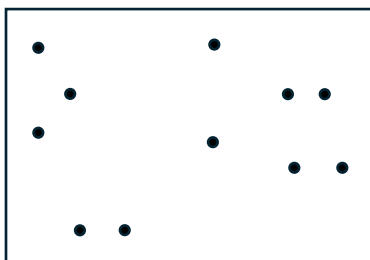
نکته: سروو موتور باید در محل پایه قطعه 90 درجه چرخیده و سپس بعد از 2 ثانیه به جای خود بازگردد.

نکته: در ارسال داده ها باید حتما تایید از جانب برد به کامپیوتر ارسال گردد.

نکته: تصویر ارساله را باید بدون Zoom In/Zoom Out با همان 100 Scale درصد پردازش کنید تا مقادیر فاصله ای بهم نریزد.

امتیازی: در صورتی که برق قطع شود، دستگاه به کامپیوتر آخرین نقطه لحیم شده را اعلام کند.

امتیازی: ساخت رابط گرافیکی در کار با میکروکنترلر نمره امتیازی دارد.



پروژه چهار) طراحی سیستم جهت بازیابی موقعیت مکانی

از طریق یک ژيروسکوپ قرار هست تا تمامی حرکات (جهت و طول) به مسافت 10 متر ثبت و ذخیره گردد. در واقع مسیری را که طی می کنیم همواره تمامی حرکتها را در 10 متر آخر، هم از نظر مسافت طولی و هم نظر جهت در حافظه موجود می باشد. این سیستم از طریق پورت سریال به سریال مانیتور متصل می باشد، باید وقتی که از طریق درگاه سریال عبارت Back را برای سیستم ارسال کردیم، ابتدا سیستم به ما تعداد حرکت های رفته شده را بر روی صفحه نمایش دهد و سپس 10 متر آخر را که طی کرده ایم به تمامی حرکات (جهت + مسافت) را به ترتیب به ما بگوید به طوری که وقتی هر حرکت را کامل انجام دادیم (بازگشتیم)، حرکت بعدی را برای ما تعریف کند. تا در نهایت با انجام تمامی این حرکات ها به جایگاه خود در 10 متر قبلی بازگردیم.

نکته: در حرکت های بازگشتی وجود 15 درصد تلورانس خطا در مسافت و زاویه مجاز می باشد.

نکته: در صورتی که مسافت پیموده شده کمتر از 10 متر باشد باید سیستم بتواند کل مسیر را بازگردد.

امتیازی: می توان به جای استفاده از سریال مانیتور، از مازول بلوتوث استفاده کرد و تمامی اطلاعات و داده ها بین تلفن و سیستم دریافت و ارسال گردد.

پروژه پنج) راه اندازی دوربین OV7670 با میکروکنترلر

در این پروژه، باید با استفاده از میکروکنترلرهای STM32 یا برد آردوینو، دوربین ov7670 را راه اندازی کرده و

یک فریم را با استفاده از آن ذخیره و در کامپیوتر (از طریق USB یا UART) نمایش دهید. توجه کنید این

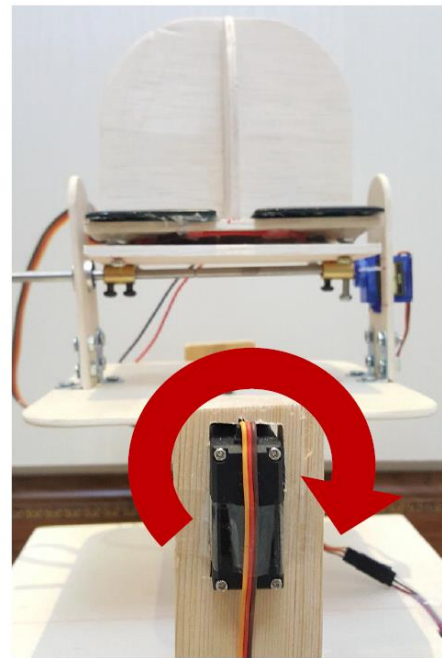
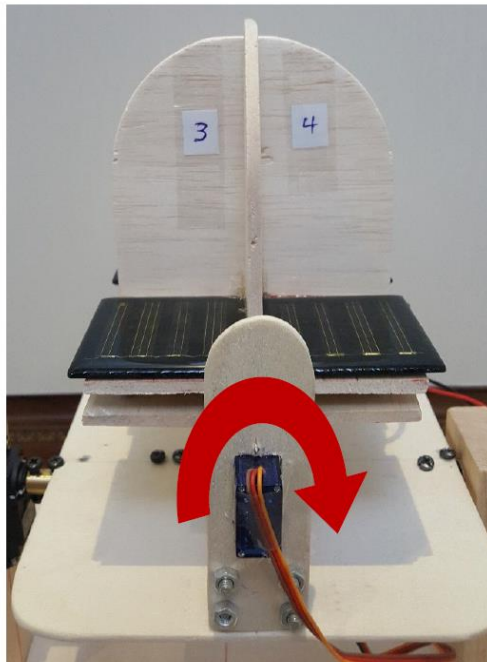
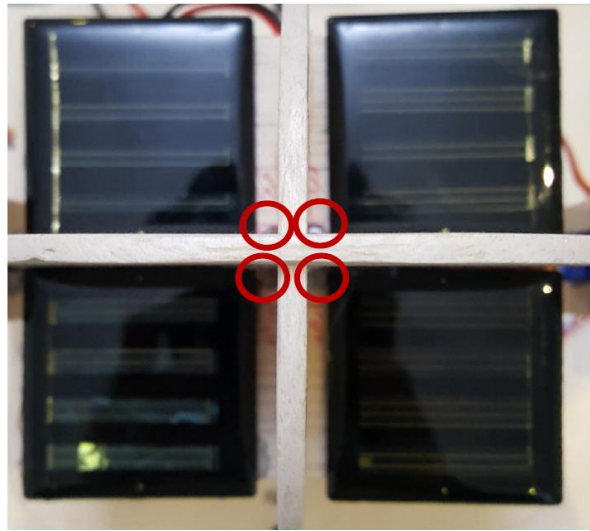
دوربین بدون FIFO بوده و بخاطر محدودیت میکروکنترلرها، راه اندازی آن نیازمند خلاقیت می باشد، البته این کار کاملاً شدنی است و آموزش های بسیاری در سطح اینترنت وجود دارد که از جمله آن می توان به [مسابقه علمی سال 1396](#) اشاره کرد.

پروژه شش) زاویه سنج

در این پروژه، باید با استفاده از سنسور شتاب سنج و جایرو (مثل سنسور MPU6050) ، یک زاویه سنج طراحی کنید. یکی از روش های بسیار مرسوم استفاده از فیلتر کالمن است که برای اطلاعات بیشتر به [این لینک](#) می توانید مراجعه کنید. در این پروژه باید از میکروکنترلرهای STM32 یا بردهای آردوینو استفاده کنید. پروژه شما باید قابلیت نمایش مقادیر بدست آمده از سنسورها و همچنین زاویه اندازه گیری شده را داشته باشد.

پروژه هفت) تعقیب کننده نور

در اکثر مزرعه های خورشیدی فعلی، صفحات خورشیدی به صورت ثابت قرار گرفته اند. زاویه نصب صفحات خورشیدی متناسب با عرض جغرافیایی هر منطقه تعیین می شود. بیشترین توان تولیدی یک صفحه خورشیدی زمانی حاصل خواهد شد که زاویه تابش خورشید، به صفحه خورشیدی موردنظر عمود باشد؛ یعنی صفحه خورشیدی بیشینه تابش را دریافت کند. در این پروژه، به صورت یک نمونه اولیه کوچک شده، رباتی برای تعقیب نور به منظور حفظ زاویه عمود تابش برای 4 عدد سلول کوچک خورشیدی طراحی و ساخته خواهد شد. این ربات همواره روی خود را به سمت بیشینه تابش موجود در فضا نگه می دارد. تصاویر زیر، یک نمونه پیاده شده از این پروژه می باشد که در آن محل قرارگیری سنسورها و موتورها معلوم است. توجه کنید در این پروژه باید توانایی جابجایی در دو محور وجود داشته باشد.



پروژه هشت) پیاده‌سازی روش CLB در ثابت افزار

در این پروژه، روش Cryptographically Obfuscated Logic Bomb یا به اختصار CLB که یکی از تکنیک‌های Anti repackaging می‌باشد باید روی ثابت‌افزار میکروکنترلرهای STM32 یا ESP32 پیاده‌سازی و تست گردد.

ابزار مورد نیاز:

شرح پروژه

در این پروژه ابتدا باید [این](#) مقاله مطالعه و بررسی شود تا مباحث نظری و نحوه کار تکنیک CLB مشخص گردد، سپس بعد از آشنایی، باید با استفاده از این تکنیک، یک ثابت‌افزار مقاوم به repackaging توسعه داده شود. در repackaging، فرد مهاجم به (فایل باینری) ثابت‌افزار اصلی یکسری دستور اضافه کرده و یک ثابت‌افزار جدید و مشتق شده از ثابت‌افزار اصلی می‌سازد. متأسفانه اکثر ثابت‌افزارها مکانیزم‌های تأیید یکپارچگی مناسب نداشته و در معرض حملات repackaging قرار دارند. علاوه بر این، بسیاری از راه‌حل‌های موجود برای دستگاه‌های اینترنت اشیا تنها بر روی بخشی از فرآیند بروزرسانی ثابت‌افزار تمرکز دارند (به عنوان مثال، از سرور به‌روزرسانی به دستگاه) و تأیید کاملی از ثابت‌افزار دانلود شده انجام نمی‌دهند و از این رو نمی‌توانند از یکپارچگی آن اطمینان حاصل کنند. البته برای این منظور راه‌حلهایی ارائه شده اما اکثر آن‌ها به یک عامل خارجی مثل کلیدهای امضا یا فناوری‌های ذخیره‌سازی ایمن نیاز دارند که می‌تواند استفاده از آن‌ها را در پلتفرم‌هایی با منابع محدود، دچار مشکل کند.

بطور خلاصه در این ایده، در زمان ساخت محتویات logic bomb را رمزنگاری می‌کنند که این محتویات رمزنگاری شده در زمان اجرا یا ران تایم، باید رمزگشایی و اجرا شود. برای رمزنگاری محتویات logic bomb، لازمه ما کلید را در فایل اجرایی قرار بدیم، اما با این کار فرد مهاجم خیلی راحت و با یک بررسی ساده می‌تواند کلید را با تکنیک‌های مهندسی معکوس پیدا کند. برای جلوگیری از سرقت کلید، بهترین راه این است که ما از منطق اجرای خود ثابت‌افزار برای مخفی کردن کلید استفاده کنیم. می‌دانیم یک متغیر، مقادیر مختلفی در طول اجرا خواهد داشت، بهترین کار برای مخفی کردن کلید، این است که یکی از همین مقادیری که متغیر در مدت زمان اجرا دارد را به عنوان کلید قرار بدیم، با این کار دیگر نیازی نیست بصورت مستقیم به مقدار کلید در کد اشاره کنیم.

پروژه نه) پوزیشن افراد

با کمک دو سنسور IMU MPU6050 نصب شده بر پا و سینه فرد و دو برد ESP32 دیتا‌های وضعیت فرد جمع‌آوری می‌شود. سپس با ارسال این دیتا‌ها توسط پروتکلی مانند MQTT به یک برد Raspberry Pi و اجرای یک مدل ماشین لرنینگ، پوزیشن فرد (ایستاده، نشسته، خوابیده به پشت، خوابیده به پهلو راست و خوابیده به پهلو چپ) را تشخیص داده و به Things Board ارسال می‌کند. داشبورد باید وضعیت لحظه‌ای،

تعداد دفعات تغییر وضعیت (ایستادن، نشستن و خوابیدن) در طول روز، مدت زمان حضور در هر وضعیت و سلامت خواب را گزارش دهد. (همچنین می توان به جای استفاده از برد Raspberry Pi در صورت امکان مدل ماشین لرنینگ را روی یکی از ESP32 ها اجرا کرد و دیتا برد دیگر را توسط یک پروتکل ارتباطی جابه جا کرد) نکته: برای preprocess دیتا های سنسور می توان از روش Kalman filter بهره برد.

پروژه ده) سنگ کاغذ قیچی

یک برد ESP32 از هر نوبت بازی عکس می گیرد. در واقع عکس ها زمانی گرفته می شوند که یک/دو سنسور فاصله مانند اولتراسونیک حضور دست ها را تشخیص دهد. سپس تصویر را به سیستم خود یا برد Raspberry pi بفرستد. اینجا یک مدل ماشین لرنینگ وظیفه دارد با تشخیص حالت دست ها امتیاز هر نوبت را محاسبه کرده و پس از نمایش بر یک LCD به یک UI ارسال کند.

روند بازی ها باید توسط UI مدیریت شود. به این صورت که ابتدا افراد شرکت کننده در بازی (حداقل 3 نفر) در سیستم ثبت شده و بر اساس تعداد، یک چارت برنامه بازی ها به صورت حذفی رسم می شود. در ادامه UI طراحی شده وظیفه اعلام نفرات هر بازی بر روی برد و نمایش امتیازات را دارد.

پروژه یازده) شبیه سازی انتقال وظایف در شبکه های پردازش مهی

در این پروژه، می خواهیم یک شبیه سازی از فرایند انتقال وظیفه در شبکه های پردازش مهی داشته باشیم. در این پروسه، یک دستگاه کاربری، وظیفه خود را به لایه می منتقل می کند تا سرعت پردازش بالا رود و همچنین انرژی مصرفی در لایه کاربری کاهش یابد. به همین منظور، برای وظیفه، یک تسک پردازش تصویر به دلخواه تنظیم کنید که حجم داده مشخصی دارد. سپس بر روی یک سرور، سرویسی برای پردازش این وظیفه فراهم کنید. دستگاه کاربری شما که می تواند یک برد رزبری باشد، پس از بدست آوردن تصویر و آماده کردن داده های مربوط به وظیفه، آن را به سرور که می تواند لپتاپ شما باشد، می فرستد. سرور پس از پردازش، نتیجه را به دستگاه کاربری بر می گرداند. تاخیر ناشی از این عملیات را یکبار به صورت عملی، (با استفاده از نوشتن کد) و یکبار به صورت نظری (با استفاده از فرمول های مربوط به نرخ ارسال داده) بدست آورده و مقایسه کنید. اندازه تصویر را افزایش و کاهش داده و نتایج را دوباره گزارش کنید. همچنین تأثیر فاصله را نیز در این شبیه سازی گزارش کنید.

پروژه دوازده) پیاده سازی راهکار کم توان الگوریتم پایه FedAVG برای Federated Learning

Learning با استفاده از سخت افزار ESP32

یادگیری مشارکتی Federated Learning (FL) با رویکرد توزیع شده و غیرمتمرکز آموزش مدل سراسری در سخت افزار های نهفته در سمت لبه شبکه انجام می دهد. از آنجایی که دستگاه های موجود در محیط لبه منابع

محدود مانند انرژی، پهنای باند و قدرت پردازشی پایین برای محاسبات یادگیری ماشین دارند، مدیریت منابع امری مهم تلقی می‌شود. در این پروژه راهکار پیشنهادی جهت بهینه سازی مصرف انرژی نسبت به پیاده سازی حالت عادی FL معرفی و اجرا کنید. همچنین راهکاری جهت اندازه گیری پارامتر انرژی مصرفی پیشنهاد و اجرا گردد. رویکرد اجرایی می‌بایست قابلیت مقیاس پذیری جهت امکان اضافه کردن دستگاه‌های جدید را داشته باشد. تأثیر محدودیت‌های منابع (مانند حافظه محدود، قدرت پردازش و انرژی) را بر فرآیند FL ارزیابی کنید. در این پروژه الگوریتم پایه FedAVG پیاده سازی گردد.

تعداد کافی سخت افزار ESP32 برای کلاینت‌ها در نظر بگیرید. ابزار های توسعه برنامه نویسی و کتابخانه های مرتبط با راه اندازی یادگیری ماشین بر روی ESP32 نصب کنید. یک زیرساخت سرور برای فرایند جمع‌آوری مدل و هماهنگی های داندلود و اپلود مدل از کلاینت ها آماده کنید. پیاده سازی پروتکل های ارتباطی بین دستگاه های سرور و سرویس گیرنده. این پروتکل می‌بایست سازگار با فریمورک انتخابی برای راه اندازی FL باشد. یک دیتاست مناسب برای آموزش مدل خود در نظر بگیرید. این دیتاست به مجموعه دیتاست های ناهمگن تقسیم شده برای هر کلاینت توزیع کنید. تحلیل آنالیز عملکرد، از جمله مدت زمان فرایند یادگیری در FL، دقت مدل در هر دور FL و هزینه های ارتباطی را اندازه گیری کنید. تاثیر محدودت منابع بر پارامتر های اندازه گیری را مشخص کنید. با بررسی استراتژی بهینه سازی برای FL راهکاری برای بهینه سازی انرژی معرفی کنید. تکنیک‌هایی از جمله Quantization، Compression و یا Client Selection میتواند باعث بهینه سازی انرژی فرایند FL گردد. استراتژی بهینه سازی خود را پیاده سازی کنید و اثر بخشی آن را ارزیابی کنید.

پروژهی عملی با امتیازات ویژه

پروژه سیزده) سیستم پایش سلامت بیمار

پروژه پیش‌رو دارای نمره اضافی نسبت به سایر پروژه‌ها بوده و همچنین قابلیت تبدیل شدن به پروژه کارشناسی برای یک الی دو نفر از اعضای گروه انجام دهنده پروژه را دارد. نمره سایر پروژه‌ها در صورت انجام کامل پروژه از 100 و در صورت انجام کامل بخش امتیازی تا 110 محاسبه می‌شود اما در پروژه پیش‌رو بسته به میزان کیفیت انجام پروژه از 120 تا 140 نمره می‌تواند داشته باشد.

در پروژه پیش‌رو هدف ارائه خدمات به افراد مراقبت کننده از بیماران است که برای این منظور نیاز است اطلاعاتی از وضعیت بیمار در اختیار این افراد قرار دهیم. ضربان قلب، دمای بدن، فشار خون، تشخیص زمین خوردن بیمار و نوار قلب از جمله مهم‌ترین اطلاعاتی است که در بررسی سلامت بیمار باید بررسی شوند.

همچنین در این پروژه با استفاده دوربین در داخل منزل و مازول GPS در خارج از منزل افراد می‌توانند به بیمار خود دسترسی داشته باشند. دوربین در داخل منزل زمانی فعال می‌شود که بیمار از روی تخت بلند شده و حرکت کند در این صورت دوربین فعال شده و شروع به چرخش می‌کند. همچنین با استفاده از مازول دوربین زمانی که بیمار روی تخت قرار دارد می‌توانید سه حالت شامل درد کشیدن، استرس و نرمال برای چهره بیمار تعیین کنید.

بخش دیگری از پروژه امکانات نرم افزاری است که کاربران این سیستم از آن بهره‌مند می‌شوند. مهمترین نکته در ارائه خدمات به مراقبت کنندگان از بیمار در اختیار قرار دادن اطلاعاتی است که از طریق سنسورها از وضعیت بیمار به دست آمده‌اند. در صورتی که برای مثال بیمار از نظر پارامترهای مورد ارزیابی (مانند ضربان قلب) در خطر قرار داشت این موضوع باید به تمام افراد مراقبت کننده از بیمار اطلاع داده شود. اطلاع رسانی از طریق اس ام اس، برنامه تلفن همراه و برنامه ویندوز باید انجام گیرد. در برنامه طراحی شده همچنین تعدادی امکانات جانبی شامل نگه داری اطلاعات ضروری بیمار (از جمله داروها، بیماری‌های پیشین، بیمارهای درمان شده و...)، یادآور مسائل مهم بیمار (مانند مصرف دارو در زمان‌های مشخص، انجام آزمایش‌ها و...)، نمایش موقعیت مکانی بیمار در خارج از خانه و نمایش اطلاعات دوربین در منزل، امکان انتخاب از بین بیماری‌های مختلف برای کاربر (پیش فرض پنج بیماری در نظر بگیرید) که متناسب با هر بیماری اعداد مربوط به پارامترهای مورد ارزیابی ممکن است تغییر کند و امکان تعیین دستی اعدادی که برای یک بیماری خاص در پارامترهای مورد ارزیابی که ممکن است خطرناک باشند. در این پروژه همچنین اطلاعات دو پارامتر ارزیابی شامل قند خون و وزن بیمار به صورت دستی از فرد مراقبت کننده از بیمار دریافت می‌شود.

نکات قابل توجه در پروژه امتیازی:

نکته اول) حدود بیش از نیمی از بخش سخت افزاری پروژه امتیازی پیش از این نیز به عنوان پروژه درس تعریف شده بود که با استفاده از رزبری پای این پروژه پیاده سازی شده اما در ترم پیش رو مانند سایر پروژه‌ها از میکرو کنترلر دیگری برای این منظور استفاده خواهد شد. تمامی کدهای پروژه در دسترس بوده و جهت الگو برداری در اختیار شما قرار خواهند گرفت. همچنین برای بخش نرم افزاری پروژه نیز یک کد پایه با استفاده از فریم ورک React Native فراهم شده است که هر دو بخش توسط TA مستقیم پروژه در اختیار شما قرار گرفته و در صورت نیاز به راهنمایی می‌توانید از ایشان مشورت بگیرید.

نکته دوم) با توجه با اینکه پروژه امتیازی نسبت به سایر پروژه نیازمند پیاده سازی نرم افزاری بیشتر است دو امکان برای سهولت در انجام پروژه فراهم است. بر خلاف سایر پروژه‌ها که بین دو الی سه نفر می‌توانند عضو داشته باشند در این پروژه می‌توانید یک نفر اضافه‌تر داشته و پروژه را چهار نفره انجام دهید. امکان دومی که

برای انجام پروژه فراهم است هماهنگی بین دو گروه دو الی سه نفره برای انجام پروژه است. به اینصورت که دو گروه می‌توانند با یکدیگر هماهنگ شده و هر کدام وظایف مشخصی برای انجام در پروژه تعیین کنند و در زمان انتخاب پروژه یک ایمیل مشترک برای بنده ارسال کنند.

پروژه‌ی عملی خانه هوشمند مبتنی بر Thingsboard

در ادامه، چندین پروژه با موضوع خانه هوشمند آمده است. هر کدام از این پروژه‌ها توسط تیم‌های مختلف تکمیل خواهد شد و در نهایت در قالب پلتفرم Thingsboard به صورت یک پارچه جمع خواهند شد. در نتیجه توصیه می‌شود در حین انجام پروژه خود، طراحی قابل فهم و تمیز را مدنظر داشته باشید تا ترکیب آن‌ها کار سختی نباشد. همچنین توجه کنید که در تمامی موارد، سیستم‌ها باید با کمک اینترنت اطلاعات خود را به Thingsboard ارسال کنند و در صورتی که اینترنت در دسترس نباشد، آن را به یک سرور محلی مرکزی ارسال کرده و بر اساس اطلاعات آن سرور برای عملکرد خود تصمیم بگیرند.

پروژه چهارده) لوستر هوشمند

در این پروژه قصد طراحی یک لوستر هوشمند را داریم که هم دارای سیستم تنظیم نور خودکار و هم سیستم هشداردهی است. در مرحله اول باید با کمک ESP32، سیستمی طراحی شود که به کمک سنسور نور محیط، وضعیت نور را در محیط اندازه‌گیری کرده و آن را به یک سرور مرکزی ارسال کند.

در مرحله دوم، باید دو سنسور تشخیص دود به این سیستم اضافه بشود. این سنسور وضعیت هوای اتاق (از لحاظ وجود دود، بسته به نوع سنسور مورد استفاده) را اندازه‌گیری کرده و به سیستم مرکزی گزارش می‌کند.

در مرحله سوم باید سیستم روشنایی هوشمند به یک ESP32 متصل شود که هم به صورت خودکار و هم به صورت دستی قابل کنترل باشد. این سیستم از طریق سرور مرکزی (Thingsboard) قابل کنترل است. در حالت دستی، مقدار روشنایی قابل استفاده به صورت درصدی توسط کاربر مشخص می‌شود و نور خروجی بر اساس آن مشخص می‌شود. در حالت اتوماتیک، در صورتی که سیستم در حالت طبیعی کارکرد خود باشد، بر اساس نور محیط، میزان نور دستگاه تغییر می‌کند. در صورتی که دود در سیستم تشخیص داده شود، وارد حالت اضطراری می‌شویم که در آن حالت تا زمانی که دود در محیط وجود داشته باشد، نور به صورت متناوب و به حالت هشدار کم‌و زیاد می‌شود.

توصیه می‌شد که از Rule Engine در Thingsboard برای کنترل حالت اتوماتیک استفاده کنید. همچنین توجه کنید که سنسور دود در پروژه ۳ هم وجود دارد؛ اما قرار نیست این دو سنسور یکی باشند و به‌صورت دو سنسور مجزا خواهند بود ولی باید در نسخه ترکیب شده پروژه‌ها، از طریق Thingsboard امکان مشخص کردن این که از کدام یک از این سنسورها برای لوستر استفاده بشود وجود داشته باشد. در صورتی که تنها یکی از سنسورهای دود در دسترس بود و داده دیگری به هر دلیلی به سیستم نمی‌رسید، از سنسور قابل استفاده برای هر عملکرد هر دو بخش تهویه هوا و لوستر استفاده می‌شود.

پروژه پانزده) سیستم دزدگیر هوشمند

در این پروژه قصد داریم یک سیستم دزدگیر هوشمند تصویری برای خانه طراحی کنیم. در این پروژه دو دوربین مختلف داریم که یکی برای بیرون و یکی برای داخل منزل استفاده می‌شود. این دوربین‌ها از طریق ESP32 اطلاعات خود را برای سرور ارسال می‌کنند.

در حالتی که سیستم در حالت کاربری عادی قرار داشته باشد و فردی داخل خانه باشد، دوربین داخل خانه هیچ تصویری به جهت حفظ حریم شخصی ضبط نمی‌کند و تنها دوربین بیرونی آن عمل می‌کند. این دوربین باید هنگامی که فردی قصد ورود داشت، تصویر آن شخص را ذخیره کرده و کنترل کند که آیا آن شخص در لیست افراد مجاز برای ورود به خانه قرار دارد یا نه. بدین منظور می‌توان از کتابخانه‌های تشخیص چهره انسان استفاده کرد. با توجه به محدودیت سخت‌افزاری ESP32، تصویر گرفته شده باید برای یک سرور ارسال شود و در آنجا پردازش مربوطه صورت گرفته و نتیجه اطلاع داده شود.

این خانه از طریق سنسورهای فراصوت، عبور افراد به داخل و خارج خانه را هم مشخص می‌کند. در صورتی که هیچ فردی داخل خانه نباشد، به‌صورت خودکار دوربین داخلی خانه هم شروع به ضبط تصاویر می‌کند. در صورتی که سیستم در حالت دزدگیر نباشد و کسی داخل خانه باشد، تنها سیستم بیرونی آن تصاویر را ضبط می‌کند. در صورتی که سیستم در حالت دزدگیر قرار داشته باشد و فردی وارد خانه بشود و پیش از آن تصویرش برای عبور به خانه مورد تأیید قرار نگرفته باشد، عکس آن به‌صورت هشدار در سیستم ثبت شده و در قالب یک ایمیل/پیامک به صاحب‌خانه اطلاع داده می‌شود. در صورتی که کسی داخل خانه باشد و سیستم دزدگیر را فعال کند، تا زمانی که فرد جدیدی به داخل خانه وارد نشده، دوربین‌های داخلی تصویری ارسال نمی‌کنند اما به‌محض این که فرد ناشناسی در این حالت وارد خانه بشود، دوربین‌های داخلی هم تصاویر را ضبط و ارسال خواهند کرد.

تصاویر ثبت شده و همچنین تعداد افراد حاضر در خانه باید از طریق Thingsboard قابل مشاهده باشند. همچنین توجه کنید که تغییر وضعیت برنامه بین حالت‌های عادی و دزدگیر از طریق Thingsboard صورت می‌گیرد.

2) پروژه‌های علمی

پروژه شانزدهم) تخصیص وظایف با در نظر گرفتن تعادل بارکاری و محدودیت‌های زمانی در معماری سه لایه ابر – مه – لبه

یک معماری سه لایه محاسبات مه به شرح زیر داریم که کد پایه آن به زبان پایتون ارائه خواهد شد:

لایه لبه (edge) تنها خودرو خواهند بود و همگی تحرک دارند. لایه مه (fog) شامل گره‌های مه ثابت و گره‌های خودرویی خواهد بود. لایه ابر (cloud) نیز شامل سرور مرکزی بوده که قدرت بالایی برای محاسبات دارد. معماری طرح مورد نظر شامل منطقه سرویس (service zone) خواهد بود و در مرکز هر کدام یک مدیر منطقه (zone manager) قرار دارد که یک نوع دکل مخابراتی بوده و وظایف از خودروهای صاحب وظیفه به این مدیر فرستاده شده تا تصمیم مناسب را برای تخصیص وظایف به گره‌های مه یا انتقال به لایه ابر بگیرد. پروتکل ارتباطی بین گره‌های صاحب وظیفه با مدیر منطقه و گره‌های مه با مدیر منطقه تاخیر کمتر، برد کمتر و سرعت بالاتر ولی پروتکل ارتباطی بین مدیرها و لایه ابر تاخیر بیشتر، برد بیشتر و سرعت نسبتاً کمتری دارد.

در معماری ارائه شده، باید یک روش تخصیص وظایف (task offloading) مناسب ارائه نمایید. روش مورد نظر باید وظایف را به گونه‌ای بین گره‌های مه ثابت و خودرویی توزیع نماید که:

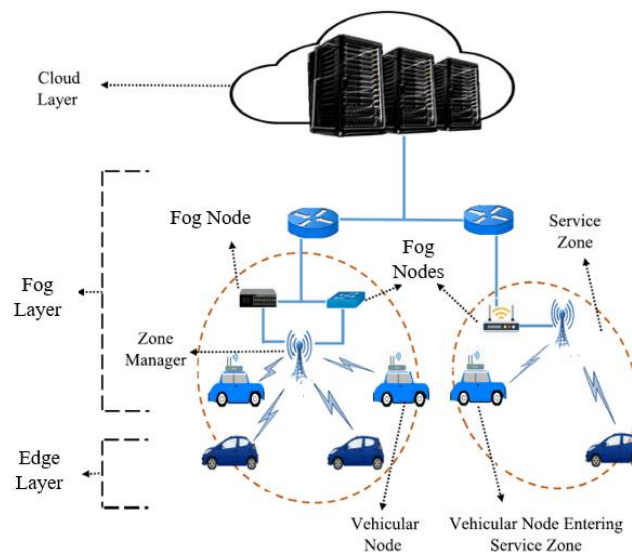
1. به عنوان هدف اصلی باید یک توازن بار پردازشی (load balancing) بین گره‌ها برقرار بوده و وظایف به طور متناسب بین گره‌ها توزیع شوند. به بیان دیگر، بعضی از گره‌ها بدون بار پردازشی و بعضی در حال پردازش با حداکثر ظرفیت خود نباشند.

2. زمان تاخیر وظایف به عنوان یک محدودیت (constraint) رعایت شود.

به منظور پیاده سازی روش توضیح داده شده می‌توانید از تکنیک‌های heuristic، meta-heuristic و تکنیک‌های یادگیری استفاده نمایید. در انتها باید بهینه بودن عملکرد روش خود را با ارائه نتایج مناسب توجیه نمایید. برای این کار باید یک پارامتر تحت عنوان load balancing تعریف نمایید که میزان بهینه بودن روش شما از نظر توزیع وظایف را نشان دهد. این پارامتر می‌تواند در بازه صفر تا یک بوده یا به صورت درصدی گزارش شود. برای اثبات برآورده شدن زمان تاخیر وظایف نیز باید عملکرد روش خود را استفاده از پارامتر delay به ازای

وظایف مختلف گزارش نمایید. مدل مربوط به وظایف از نظر حجم وظایف به بیت، تاخیر قابل قبول برای هر وظیفه و... و همچنین مدل حرکتی (mobility model) نیز به همراه کد پایه برای پیاده سازی به صورت آماده ارائه خواهد شد.

بخش امتیازی: یک روش اضافی مطابق با توضیحات قبل با استفاده از تکنیک های heuristic، meta-heuristic و تکنیک های یادگیری برای پیاده سازی خود ارائه داده و آن را با پیاده سازی اصلی خود از نظر پارامترهایی که قبلا به آن اشاره گردید، مقایسه نمایید.



پروژه هفده) تخصیص وظایف با در نظر گرفتن پدیده مهاجرت در دستگاه های متحرک در معماری سه لایه ابر - مه - لبه

یک معماری سه لایه محاسبات مه به شرح زیر داریم که کد پایه آن به زبان پایتون ارائه خواهد شد:

لایه لبه (edge) تنها خودرو خواهند بود و همگی تحرک دارند. لایه مه (fog) شامل گره های مه ثابت و گره های خودرویی خواهد بود. لایه ابر (cloud) نیز شامل سرور مرکزی بوده که قدرت بالایی برای محاسبات دارد. معماری طرح مورد نظر شامل منطقه سرویس (service zone) خواهد بود و در مرکز هر کدام یک مدیر منطقه (zone manager) قرار دارد که یک نوع دکل مخابراتی بوده و وظایف از خودروهای صاحب وظیفه به این مدیر فرستاده شده تا تصمیم مناسب را برای تخصیص وظایف به گره های مه یا انتقال به لایه ابر بگیرد. پروتکل ارتباطی بین گره های صاحب وظیفه با مدیر منطقه و گره های مه با مدیر منطقه تاخیر کمتر، برد کمتر و سرعت بالاتر ولی پروتکل ارتباطی بین مدیرها و لایه ابر تاخیر بیشتر، برد بیشتر و سرعت نسبتا کمتری دارد.

در معماری ارائه شده، باید مهاجرت وظایف را به شرح زیر پیاده نمایید:

در صورتی که گره خودرویی یا خودروی صاحب وظیفه منطقه سرویس مورد نظر را ترک کند، مهاجرت وظایف را باید در نظر گرفت. با توجه به این که هم گره مه ثابت (fog node) و هم گره خودرویی (vehicular node) داریم، لذا دو حالت مهاجرت خواهیم داشت:

الف) وظیفه مربوطه از خودرو صاحب وظیفه به گره مه ثابت تخصیص داده شده باشد.

ب) وظیفه مربوطه از خودرو صاحب وظیفه به گره خودرویی تخصیص داده شده باشد.

در حالت الف، شرایط ساده تر بوده و تنها جابجایی و مهاجرت خودرو صاحب وظیفه را برای مهاجرت در نظر می گیریم. در حالت ب، ممکن است هر کدام از خودروهای صاحب وظیفه یا گره خودرویی یا هر دو مهاجرت کنند که باید همه ی حالات در نظر گرفته شوند. هنگامی که هر کدام از خودروها از منطقه سرویس قبلی خارج شوند، وظیفه انتقال داده نمی شود؛ این اتفاق زمانی می افتد که اجرای وظیفه به پایان رسیده، و در نهایت نتیجه آن با استفاده از ارتباط بین مدیر منطقه جدید و مدیر منطقه قبلی، به خودروی صاحب وظیفه انتقال داده می شود. با توجه به اینکه مسئله NP-hard است، لذا نیاز به پیاده سازی الگوریتم های یادگیری (مانند یادگیری تقویتی) خواهد بود. شکل معماری پایه این پروژه نیز مانند پروژه شانزده است.

پروژه هجده) توزیع وظایف و اختصاص منابع در معماری سه لایه ابر – مه – لبه با استفاده از نظریه بازی ها

کد پایه پروژه به زبان پایتون ارائه خواهد شد:

در این پروژه یکی از روش های پیشنهاد شده برای task offloading و resource allocation در معماری سه لایه باید پیاده سازی شود. هدف اصلی این روش، افزایش computation efficiency شبکه vehicular edge computing به عنوان یک سامانه کلی است. computation efficiency به عنوان نسبت بیت های محاسبه شده کل به انرژی مصرف شده در خودروهای الکتریکی (EVs) تعریف می شود. برای این منظور، یک مسئله بهینه سازی تعریف شده تا از طریق بهینه سازی استراتژی Task Offloading وظایف و تخصیص منابع F ، بهره وری سیستم

حداکثر شود و در نهایت computation efficiency را افزایش یابد. به صورت خلاصه در این روش برای اتخاذ سیاست‌های task offloading از نظریه بازی‌ها کمک گرفته شده و با انجام task offloading و resource allocation به صورت تکرار شونده و وراثتی، به بهینه‌ترین جواب دست پیدا می‌کند. الگوریتم زیر نحوه کار این روش را به صورت خلاصه بیان می‌کند.

Input: Vehicle $\mathcal{N} = \{1, 2, \dots, N\}$, task $\varphi_i = \{C_i, \alpha_i^{in}, r_i^{max}\}$, $i \in N$, vehicle speed $v \sim CN(U, \sigma^2)$, vehicle position s_i , initial offloading strategy D_Θ

Output: The computation efficiency \mathbb{E} , the computation resource allocation \mathcal{F}^* , and offloading strategy \mathcal{D}^*

```

1 Allocate computation resource  $f_i^{vec*}$  to the vehicles by Algorithm 1, calculate  $U_i^{vec}$  based on (16)
2 for each vehicle  $i \in \mathcal{N}$  do
3   if  $U_i^{loc} > U_i^{vec}$  then
4      $d_i^{vec} = 0, d_i^{loc} = d_i^{loc}$ 
5   else
6      $d_i^{vec} = 1, d_i^{loc} = 0$ 
7   end
8    $D(t) = \{D_i^{loc}(t), D_i^{vec}(t)\}$ 
9   Calculate  $\mathbb{E}$  according to (17)
10 end
11  $t = t + 1$ 
12 while  $D(t-1) \neq D_\Theta$  do
13    $D_\Theta = D(t-1)$ 
14    $i = 1$ 
15   while  $i \leq N$  do
16      $d'_i = d_i^{vec} = 1$ , and compute  $U_{update}(i) = U(d'_i, d_{-i}(t-1))$  based on Algorithm 1
17      $i = i + 1$ 
18   end
19   for all vehicle  $i \in \mathcal{N}$  do
20     if vehicle  $i$  has maximal  $U_{update}$ , or  $d_\Theta(i) = d_i^{vec} = 1$  then
21        $d_i(t) = d'_i$ 
22     else
23        $d_i(t) = d_i(t-1)$ 
24     end
25   end
26    $t = t + 1$ 
27 end
```

در این پروژه نیازی به اثبات روابط و پرداختن به مباحث نظریه بازی‌ها نیست و صرفاً باید الگوریتم بالا را بر اساس سیستم مدل فرضی خود پیاده‌سازی کرده و نتایج آن را با یکی از روش‌های پایه در این حوزه مثل MACTER یا CTORA مقایسه کنید.

پروژه نوزده) پیاده سازی یک مدل وظایف شبیه به واقعیت برای خودروهای خودران

همانطور که می‌دانیم خودروهای خودران برای داشتن عملکرد مناسب در سطح شهر یا جاده نیاز دارند تا تعدادی بسیار زیادی از وظیفه (Task) اجرا کنند. از جمله مهم‌ترین وظایفی که در خودروهای خودران می‌توان اشاره کرد

تشخیص اشیاء به منظور جلوگیری از برخورد، درک محیط پیرامون، تبادل اطلاعات با سایر خودروها، استفاده نقشه و اشاره کرد. در این پروژه لازم است تا ده وظیفه اصلی که یک خودرو خودران دارد را تعیین کنید و سپس وظایف مربوط به این خودروها را به صورت کاملاً ساده پیاده سازی کنید. در پیاده سازی وظایف این خودروها به وابستگی یا عدم وابستگی بین بسته‌های هر وظیفه، اندازه هر بسته و زمان انتشار هر وظیفه (تناوبی یا غیر تناوبی و ...) توجه فرمایید. در نهایت باید یک کد ساده به زبان دلخواه بنویسد که یک مجموعه داده از تمام وظایفی که ویژگی‌های آن را بررسی کردید را تولید کند.

نکته) لازم به ذکر است در پیاده سازی وظایف لازم نیست ماهیت وظایف پیاده‌سازی شود صرفاً ویژگی‌های وظایف باید پیاده‌سازی شوند. برای مثال اگر یک وظیفه دارید که هر ثانیه 30 عکس پردازش می‌کند صرفاً کافی است نوع وابستگی عکس‌ها در این وظیفه، حجم وظیفه و ... به عنوان ویژگی این وظیفه پیاده‌سازی شود.

پروژه بیست) ارائه یک الگوریتم برای یافتن مدت زمان اقامت

در پردازش مهی سیار، از مفهومی به نام *sojourn time* یا زمان اقامت برای بدست آوردن مدت زمان اقامت یک دستگاه کاربری در شعاع پوشش یک سرور مهی استفاده می‌شود. این مفهوم در روش‌های فعلی، از توزیع‌های احتمالاتی بدست می‌آید. در این پروژه، الگوریتمی ارائه کنید، که این مدت زمان ماندگاری را بر اساس ویژگی‌های فیزیکی محیط خروجی دهد. به همین منظور فرض کنید که مکان سرور مهی ثابت است و x و y آن در اختیار است. همچنین برای دستگاه‌های کاربری، موقعیت و سرعت و جهت به ازای هر ثانیه در دسترس است. در نهایت درستی الگوریتم خود را با استفاده از شبیه‌ساز SUMO روی یک سناریو واقعی تست کنید. (موقعیت سرورهای مهی را تصادفی در سطح سناریو پخش کنید). این نتایج را با *sojourn time* مقایسه کنید.

3) پروژه‌های پیاده سازی مقالات پیشین

پروژه بیست و یک) حل مسئله تصمیم انتقال وظایف و اختصاص منابع با

استفاده از روش‌های فراابتکاری در مدل 2 لایه پردازش مه

مدل پردازش مهی ارائه شده در مقاله *Mobility-Aware Task Offloading and Migration Schemes in Fog Computing Networks* را پیاده‌سازی کنید. در این مدل، وظایف از طریق دستگاه‌های کاربری تولید می‌شوند و می‌توانند به صورت محلی اجرا شده یا به یکی از سرورهای مهی در دسترس آن دستگاه کاربری، انتقال داده شوند. فرمول‌های تاخیر ناشی از اجرای محلی و همچنین ارسال وظیفه و اجرای داخل سرور مهی در مقاله آورده شده است. همچنین مدل پردازش مهی این مقاله، تحرک یا *mobility* دستگاه‌های کاربری را در نظر می‌گیرد و پارامتری تحت عنوان *sojourn time* یا زمان اقامت برای دستگاه‌ها تنظیم می‌کند. این پارامتر برای کاهش پدیده مهاجرت در نظر گرفته می‌شود و الگوریتم تصمیم‌گیری باید وظایف را به سرورهایی منتقل کند که احتمال مهاجرت کم باشد چون هزینه این پدیده زیاد است. پس از پیاده‌سازی مدل پردازش مهی، باید مسئله تصمیم‌انتقال و تخصیص منابع را حل کنید. روشی که قرار است مسئله را حل کنید، با مقاله متفاوت است و باید با استفاده از یکی از روش‌های فراابتکاری و با جستجوی فضای حالت به جواب نزدیک بهینه برسید. به طور کلی ورودی مسئله پارامترهای زیر هستند:

- تعداد دستگاه‌های کاربری
 - تعداد سرورهای مهی
 - گراف شبکه و موقعیت فیزیکی هر دستگاه کاربری و هر سرور مهی
 - سرعت پردازش، توان انتقال داده و پارامترهای مربوط به نوع سخت‌افزار دستگاه‌ها و سرورها که اعداد ثابت تصادفی در نظر گرفته می‌شود. (مطابق مقاله)
 - هزینه اضافی ناشی از مهاجرت وظایف که مطابق مقاله یک ضریب ثابت می‌تواند در نظر گرفته شود.
- خروجی‌های مسئله نیز دو مجموعه هستند.
- مجموعه تصمیم‌گیری‌ها برای هر وظیفه که نشان می‌دهد هر وظیفه به صورت محلی اجرا می‌شود یا بر روی لایه مه اجرا می‌شود.
- مجموعه تمام مجموعه‌های مقدار منبع اختصاص داده شده به هر تسک برای هر سرور مهی. دقت شود که این مجموعه مقصد هر وظیفه را نیز در بر می‌گیرد چون منبع اختصاصی هر وظیفه می‌تواند تنها روی یک سرور مهی غیر صفر باشد.

Federated Learning in Heterogeneous Mobile-Edge Computing

در این پروژه با شبیه سازی یک روش کارآمد برای کاهش مصرف انرژی مصرفی دستگاه کاربران مشارکت کننده در فرایند یادگیری به صورت Federated Learning ارائه شده است. در این روش با توجه به تاخیرهای موجود در شبکه به علت مکانیزم های موجود مثل TDMA برای دسترسی به پهنای باند دستگاه کاربران باید منتظر دستگاه دیگری باشد. از این جهت، با استفاده از روش DVFS که با کاهش فرکانس و ولتاژ کاری هسته CPU انرژی مصرفی دستگاه کاربران را کاهش می دهد. با استفاده از این تکنیک فرایند یادگیری مشارکتی را بهینه می کنیم تا از تاخیرهای احتمالی به وجود آمده در شبکه استفاده کرده تا انرژی مصرفی کاهش یابد. همچنین با استفاده از روش Client Selection دسته کاربرانی که سریع تر فرایند آموزش و ارسال مدل خود را به همراه دارا بودن حداقل قابلیت اطمینان انجام می دهند، انتخاب می گردد. منظور از قابلیت اطمینان در اینجا دستگاه کاربرانی که حداقل دقت مدل را دارا باشند انتخاب می شود. ارائه نتایج برای مجموعه داده CIFAR-10 در دو حالت IID و Non-IID الزامیست. این نتایج شامل مدت زمان آموزش در دقت های متفاوت، میزان انرژی مصرفی و دقت مدل در هر دوره مشابه مقاله نشان دهد.

پروژه بیست و سه) پیاده سازی Data -Driven Participant Selection and

Bandwidth Allocation for Heterogeneous Federated Edge Learning

در این پروژه از راهکار یادگیری ماشین توزیع شده با رویکرد Federated Learning (FL) استفاده شده است. با توجه به محدودیت منابع ارتباطی و قابلیت پردازشی دستگاه های نهفته بدنبال افزایش مشارکت دستگاه ها در فرایند یادگیری مشارکتی هستیم. از طریق شبیه سازی، مسئله بهینه سازی با توجه زمان های آموزش مدل در دستگاه و مدت زمان تاخیر در ارسال مدل به همراه وضعیت توزیع تعداد سَمپل های داده در دستگاه های کاربران مشخص می گردد. سپس از طریق یک استراتژی انتخاب کاربران موثر، دستگاه کاربران مشخص در یکی deadline دینامیک با هدف در نظر گرفتن محدودیت های منابع ارتباطی انجام می گردد. تمامی نتایج ارائه شده در مقاله، برای 3 دیتاست MNIST و CIFAR-10 و FEMNIST در استراتژی های Client Selection مثل انتخاب تصادفی یا انتخاب Greedy و روش پیشنهادی مقایسه گردد.

Energy-Efficient Resource Management for پیاده سازی (چهار) Federated Edge Learning With CPU-GPU Heterogeneous Computing

جهت پیاده سازی یادگیری مشارکتی در این مقاله از روش تقسیم مجموعه دیتاست برای روی دو هسته CPU/GPU و پردازش موازی (WorkLoad partitioning) مدل Federated learning استفاده کرد. این روش با هدف کاهش مصرف انرژی (روش DVFS) و اجرای یادگیری مشارکتی در مدل زمان کمتر انجام شده است. هدف از انجام این پروژه پیاده سازی عملی این روش ارائه شده بر روی سخت افزار NVIDIA JETSON NANO و NVIDIA JETSON TX2 NX و Orange pi 5 و Raspberry pi 3/4 انجام گردد. لازم به ذکر است هدف راه اندازی یادگیری مشارکتی بر روی این سخت افزار ها است. بعضی از آنها قابلیت DVFS را دارا هستند و بعضی از آنها صرفاً CPU دارند. هدف ارائه نتایج برای تمامی حالت های ترکیب دستگاه ها است. از طرفی در این مقاله روش های دیگری مانند Bandwidth allocation و C2 time division و Device Scheduling معرفی شده است که نیازی به پیاده سازی آن نیست و صرفاً از نظر نظری این قسمت ها در گزارش ارائه گردد. لازم به ذکر است موارد پیاده سازی عملی از نظر مدت زمان آموزش، انرژی مصرفی و نمودار همگرایی دقت مدل در دوره های فرایند یادگیری با استفاده از مجموعه داده های CIFAR-10 و MNIST پیاده سازی گردد.

EdgeMatch: A Smart Approach for Scheduling IoT- پیاده سازی (پنج) Edge Tasks With Multiple Criteria Using Game Theory

Adaptive workload adjustment for cyber-physical (پروژه بیست و شش)
systems using deep reinforcement learning

Energy-aware standby-sparing on heterogeneous (پروژه بیست و هفت)
multicore systems

Application and Thermal-reliability-aware (پروژه بیست و هشت)
Reinforcement Learning Based Multi-core Power
Management