

به نام حق



گزارش پروژه پایانی – درس IoT 4021

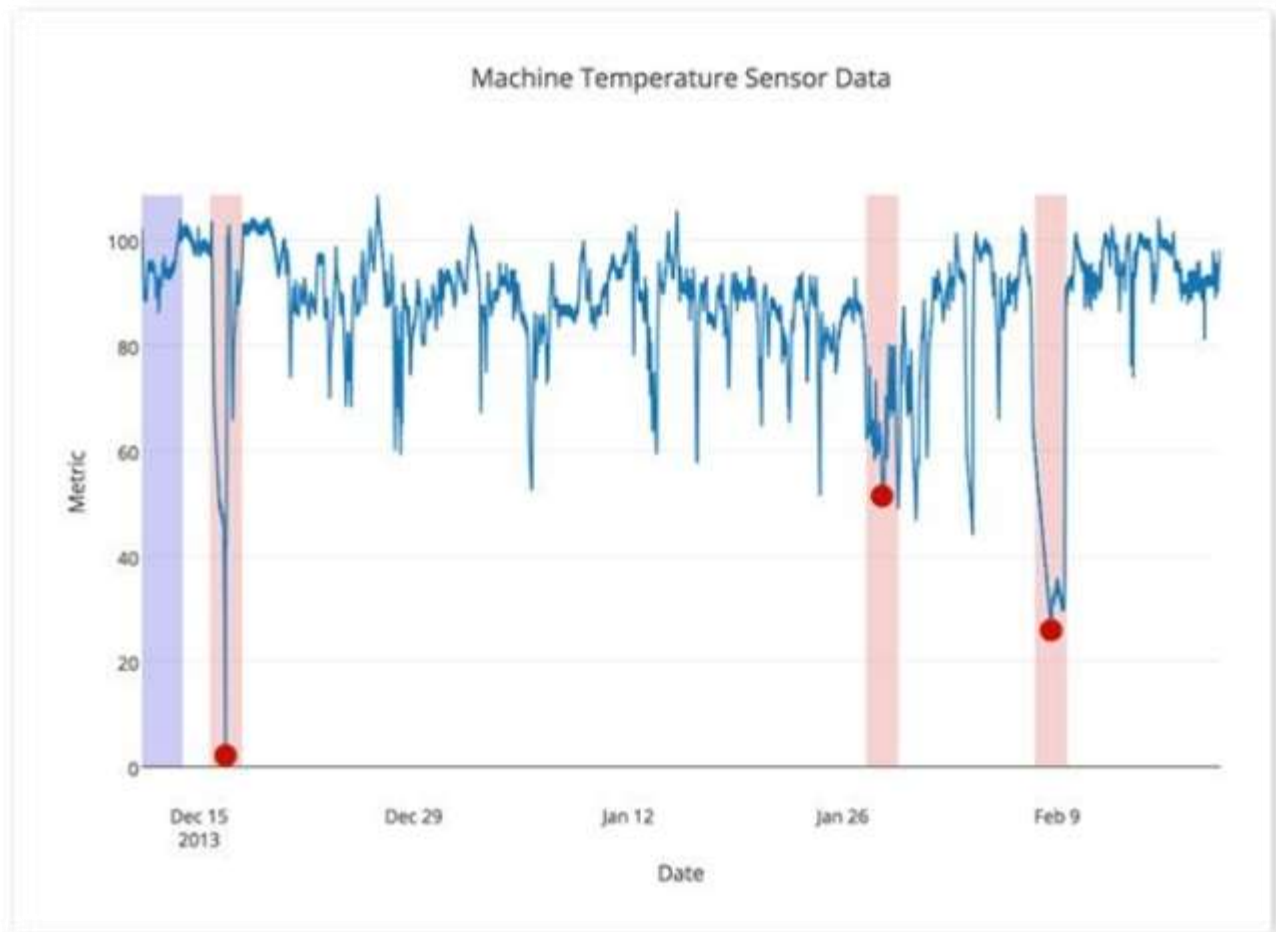
تشخیص ناهنجاری در دما

با استفاده از الگوریتم های یادگیری ماشین

آرمین عطارزاده

ارشیا گشتاسبی

بخش صفر: مقدمه و تعریف هدف



مقدمه:

در دنیای امروزی که اینترنت اشیا (IoT) به یکی از فناوری‌های کلیدی تبدیل شده است، تحلیل داده‌های حاصل از این اشیا و استخراج اطلاعات مفید از آنها از اهمیت بسیاری برخوردار است. یکی از کاربردهایی که در صنعت بسیار مورد نیاز است، تشخیص نقص‌ها و ناهنجاری‌ها در داده‌های حسگری می‌باشد. در این پروژه، ما به تشخیص ناهنجاری‌ها در داده‌های دما با استفاده از روش‌های یادگیری ماشین می‌پردازیم.

▪ اهداف:

هدف اصلی این پروژه، پیاده‌سازی یک سیستم تشخیص ناهنجاری بر روی داده‌های دما است که از حسگرهای موجود در دستگاه‌های IoT جمع‌آوری می‌شود. این سیستم قادر است به صورت خودکار و هوشمند نوعی از ناهنجاری‌ها را تشخیص دهد که ممکن است به عوامل مختلفی نظیر خرابی حسگر، مشکل در محیط محیطی یا حتی حمله مخرب به شبکه مربوطه برگردد. به طور خاص، اهداف این پروژه عبارتند از:

- آشنایی با فناوری‌های IoT و مفاهیم مرتبط با آنها
- ارائه روش‌های یادگیری ماشین برای تشخیص ناهنجاری‌ها در داده‌های دما
- پیاده‌سازی یک سیستم عملی و قابل استفاده برای تشخیص ناهنجاری در داده‌های سنسوری
- ارائه سیستم رمزنگاری شده برای ارتباط به همراه پیاده‌سازی سیستم پایگاه داده

▪ کاربرد های تشخیص ناهنجاری در دما:

تشخیص نوسانات یا ناهنجاری‌ها در داده‌های دما از اهمیت ویژه‌ای برخوردار است و در موارد مختلفی از جمله در صنایع مختلف و حوزه‌های متعددی در امنیت و کارایی به کار می‌رود:

پیشگیری از خرابی تجهیزات: در بخش‌های صنعتی، مانند کارخانه‌ها و تولیدی‌ها، دما می‌تواند یکی از پارامترهای بسیار حیاتی باشد. با استفاده از تشخیص نوسانات در دما، می‌توان به صورت زودهنگام از خرابی یا نقص در تجهیزات هشدار داد و از خسارات بیشتر جلوگیری کرد.

مانیتورینگ: در محیط‌هایی که نیاز به کنترل دما دارند، مثل اتاق‌های سرور، انبارها، یا محیط‌های زیستی، تشخیص ناهنجاری‌ها در دما می‌تواند به مانیتورینگ بهتر و مدیریت اثربخش‌تر این محیط‌ها کمک کند.

کشاورزی هوشمند: در بخش کشاورزی، نظارت بر دما می‌تواند برای کشت و برداشت محصولات بسیار مهم باشد. با تشخیص نوسانات ناگهانی دما، می‌توان از بروز مشکلاتی مانند خشکسالی یا خرابی سیستم‌های آبیاری جلوگیری کرد.

بهبود کارایی مکان‌های سلامت: در بخش‌هایی مانند بیمارستان‌ها، آشپزخانه‌ها، و محیط‌های عمومی، تشخیص نوسانات در دما می‌تواند به مدیران کمک کند تا بهداشت و ایمنی را بهبود بخشند و از بروز مشکلاتی نظیر بیماری‌های انتقالی جلوگیری کنند.

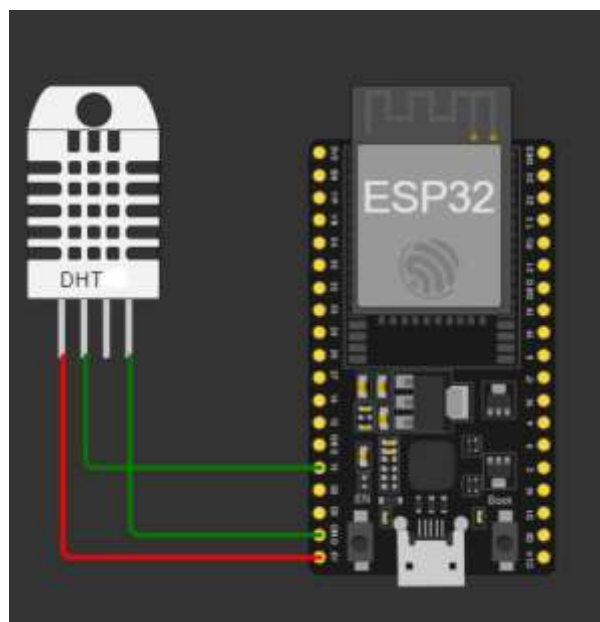
در کل، تشخیص نوسانات در داده‌های دما با استفاده از روش‌های یادگیری ماشین و الگوریتم‌های تشخیص ناهنجاری می‌تواند بهبود عملکرد و امنیت در انواع مختلفی از سیستم‌ها و بخش‌های صنعتی و غیرصنعتی را فراهم کند.

بخش اول: سخت افزار IoT



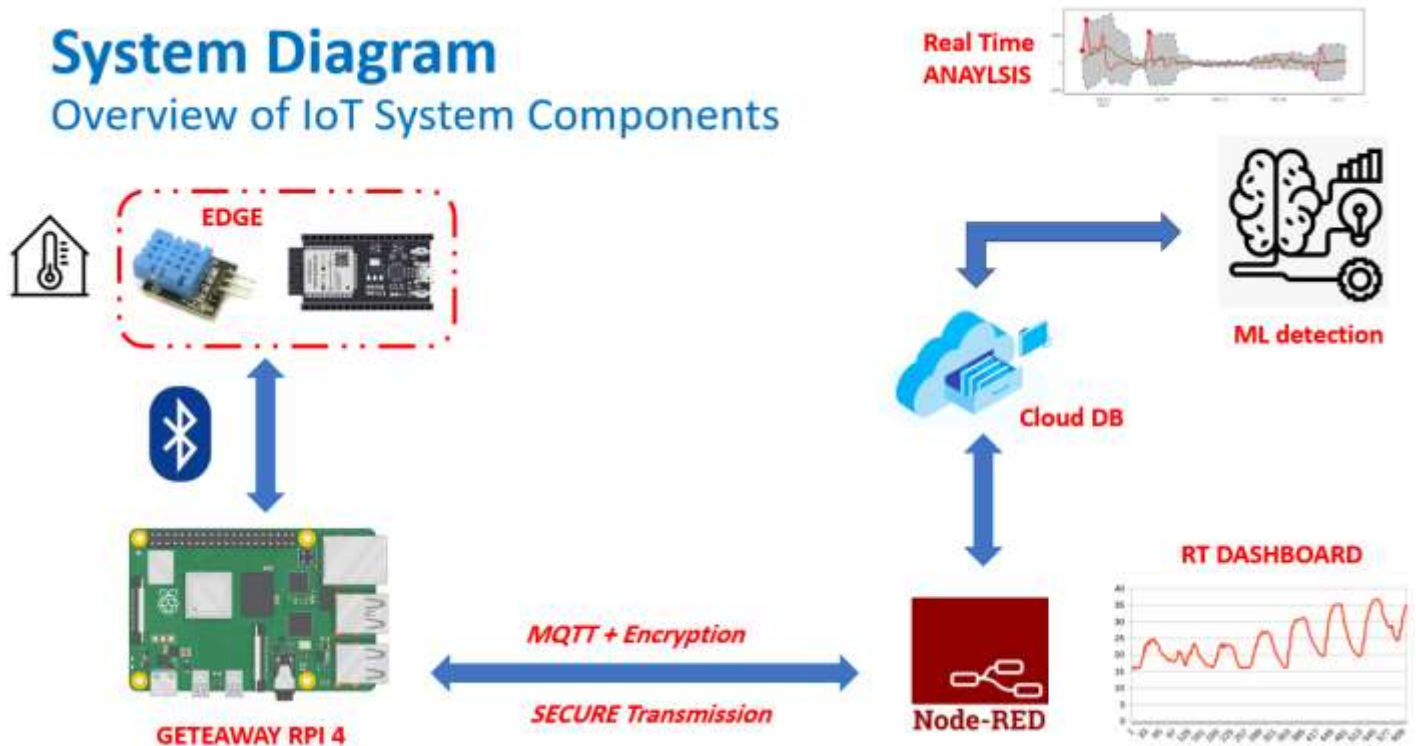
در لایه EDGE، از یک ESP32 و یک سنسور DHT11 استفاده شده است که هدف آن ارسال داده به Gateway یا همان Raspberry Pi می باشد. برای اینکار از یک برنامه در Arduino IDE استفاده شده است. یک بخش برنامه مربوط به خواندن داده از روی سنسور و دیگر ارسال آن توسط بلوتوث می باشد. برای چک کردن این روند از برنامه nrf connect نیز استفاده شده است.

شماتیک مدار مربوط به ESP32:



کد این بخش در فایل sendDHTBLE.ino موجود می باشد.

برای بلوک دیاگرام کلی سیستم داریم:



بخش دوم: ارتباطات IoT

در این پروژه از دو نوع ارتباط استفاده شده است:

(۱) تبادل داده بین ESP32 و Raspberry Pi که توسط بلوتوث انجام شده است. برای این امر از دو برنامه (یکی به زبان C و دیگر Python) در دو دستگاه استفاده شده است. در بخش Python، از کتابخانه‌ی Bluepy استفاده شده است.

(۲) تبادل داده بین Raspberry Pi و Cloud که توسط پروتکل MQTT انجام گرفته است. بعد از رمزنگاری داده‌ها توسط الگوریتم RC4، به روی Cloud فرستاده می‌شوند.

کد بخش پایتون از ۳ بخش مجزا تشکیل شده است:

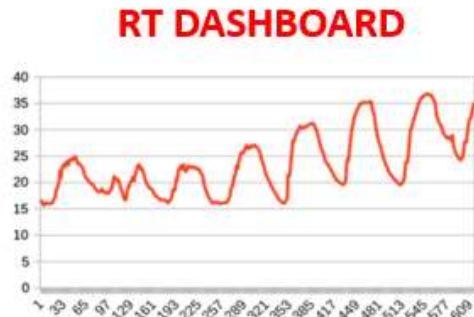
(۱) دریافت داده از بلوتوث (کتابخانه‌ی Bluepy)

(۲) رمزنگاری داده‌ها توسط RC4 (پیاده سازی توسط توابع خود پایتون)

(۳) ارسال داده‌ها بر روی Cloud (کتابخانه‌ی paho-mqtt)

کد برنامه‌ی این بخش در فایل sendToCloud.py موجود است.

بخش سوم: نمایش داده ها و Dashboard



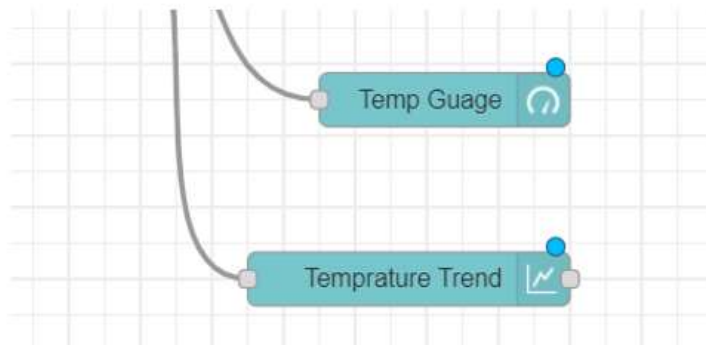
برای بحث مصورسازی داده ها، از داشبورد Node-RED برای پیاده‌سازی یک رابط کاربری برای نمایش داده های دما استفاده شد. بعد از جمع آوری داده از سنسور و انتقال امن آن از طریق سرور های ابری، در محیط برنامه نویسی Node-RED داده ارسال رمز گشایی می‌شود و سپس داده عددی دما به صورت زمان حقیقی در یک نمودار زمانی نمایش داده می‌شود.

برای نمایش لحظه‌ای یک Gauge و برای نشان دادن روند زمانی دما از Chart استفاده شد.

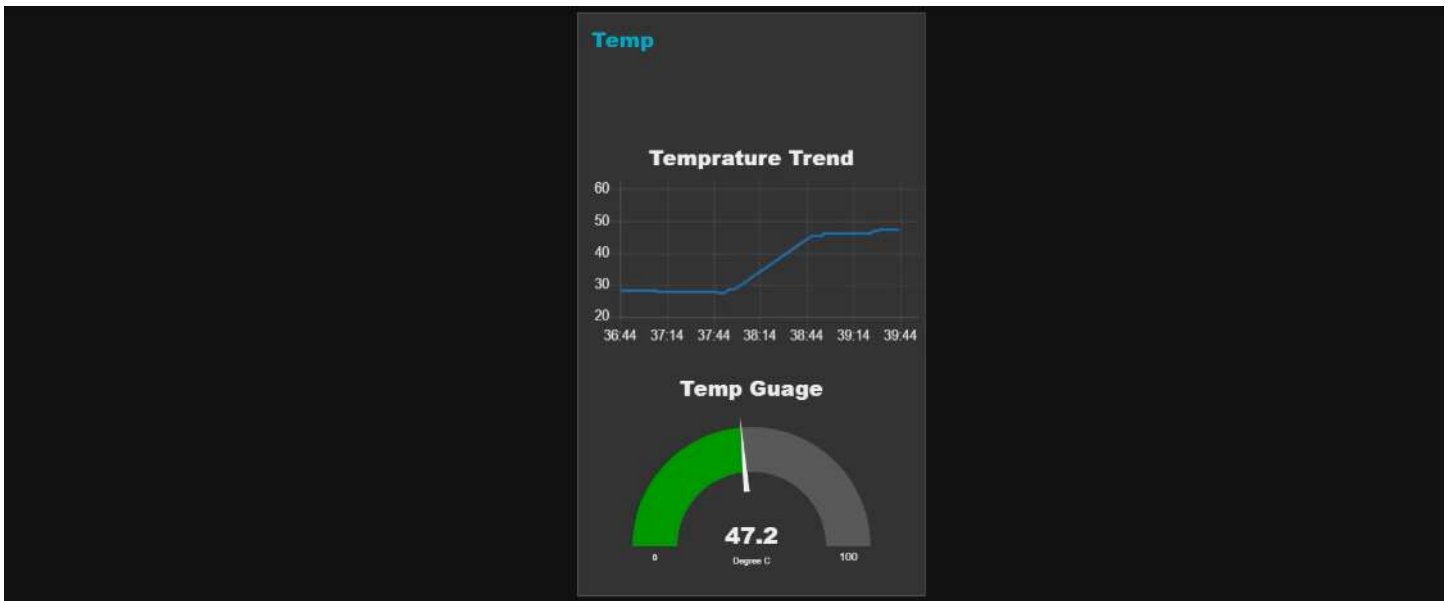
برای تشکیل این رابط کاربری نیاز به نصب کتابخانه زیر در محیط Node-RED هستیم

```
node-red-dashboard 3.6.2
```

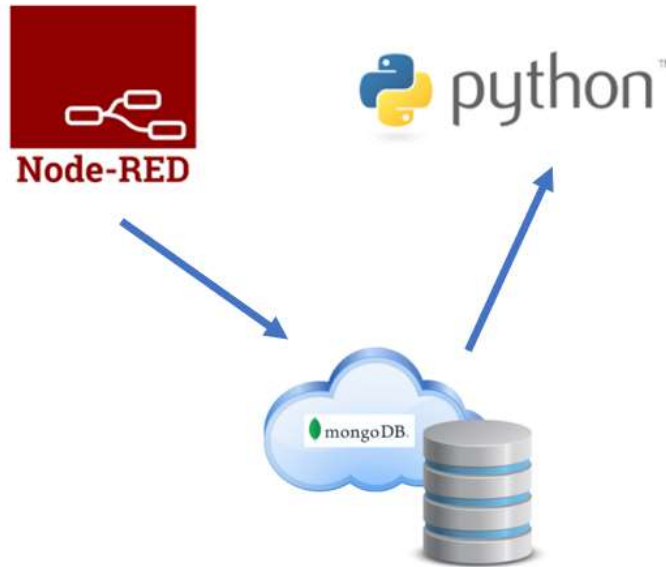

بعد از نصب از بلوک های زیر استفاده شد:



نمایشی از سیستم مصورسازی داده در داشبورد Node-RED



بخش چهارم: پایگاه داده و ذخیره داده ها



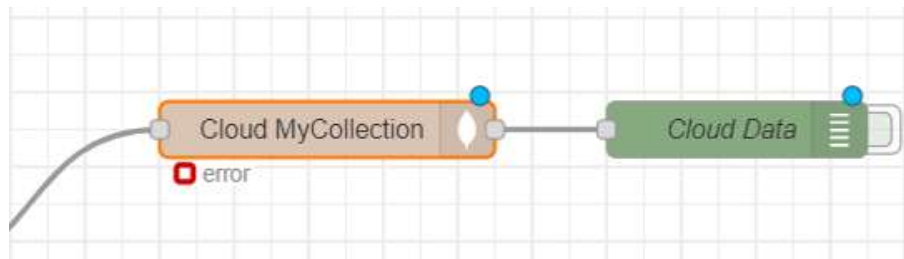
به همراه بخش مصورسازی، نیاز است داده بعد از رمزگشایی به صورت مناسبی در پایگاه داده به صورت Cloud ذخیره شود. در این پروژه از سرویس رایگان ابری MongoDB استفاده شد. در محیط Node-RED بعد از نصب کتابخانه مربوطه می توان داده ها را به صورت JSON در فضای ابری ذخیره کرد. بعد از ساختن یک Collection در سایت MongoDB، می توان بر روی آن داده ذخیره کرد یا از روی خواند.

بعد از ذخیره سازی داده، در محیط Google Colab و با زبان پایتون، داده ها را می خوانیم تا برای الگوریتم یادگیری ماشین استفاده شود.

برای ذخیره مقادیر دما در پایگاه داده در محیط Node-RED کتابخانه زیر نصب شود:

node-red-contrib-mongodb-aleph 0.3.0

با بلوک MongoDB out می‌توان داده‌ها را ذخیره کرد.



در بلوک، باید لینک اتصال به سرور و گذرواژه و نام Collection را از وب سایت MongoDB ایجاد و استفاده نمود. در تصویر زیر اطلاعات ذخیره شده در پایگاه داده قابل مشاهده است:

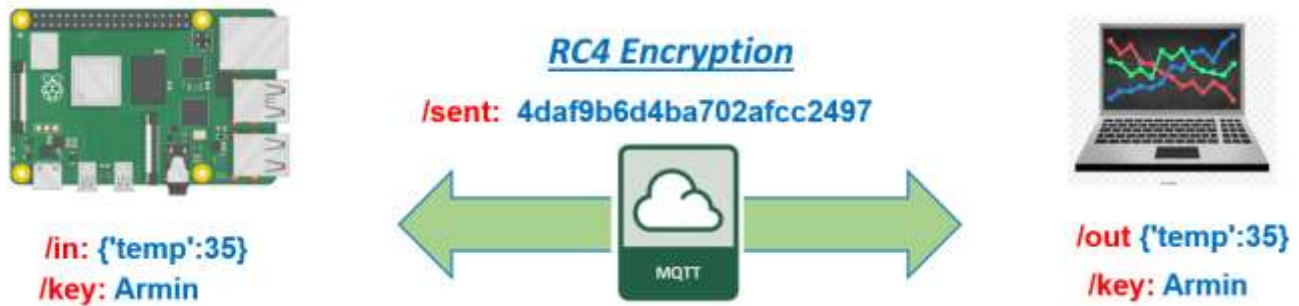
QUERY RESULTS: 1-20 OF MANY

```
_id: ObjectId('65b3d3ad01da125fe019f048')
payload: 26.3
_msgId: "c8ea51d0d5658175"
```

```
_id: ObjectId('65b3d3ae01da125fe019f049')
payload: 26.3
```

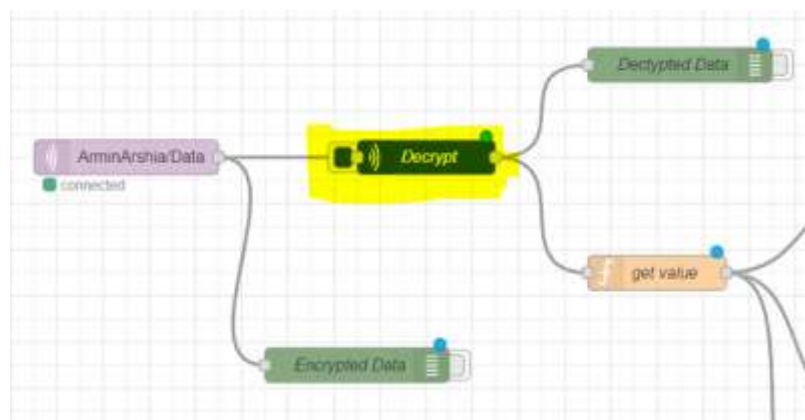
در مرحله تحلیل داده‌ها با یادگیری ماشین، از این پایگاه داده برای تحلیل و تشخیص ناهنجاری استفاده می‌شود

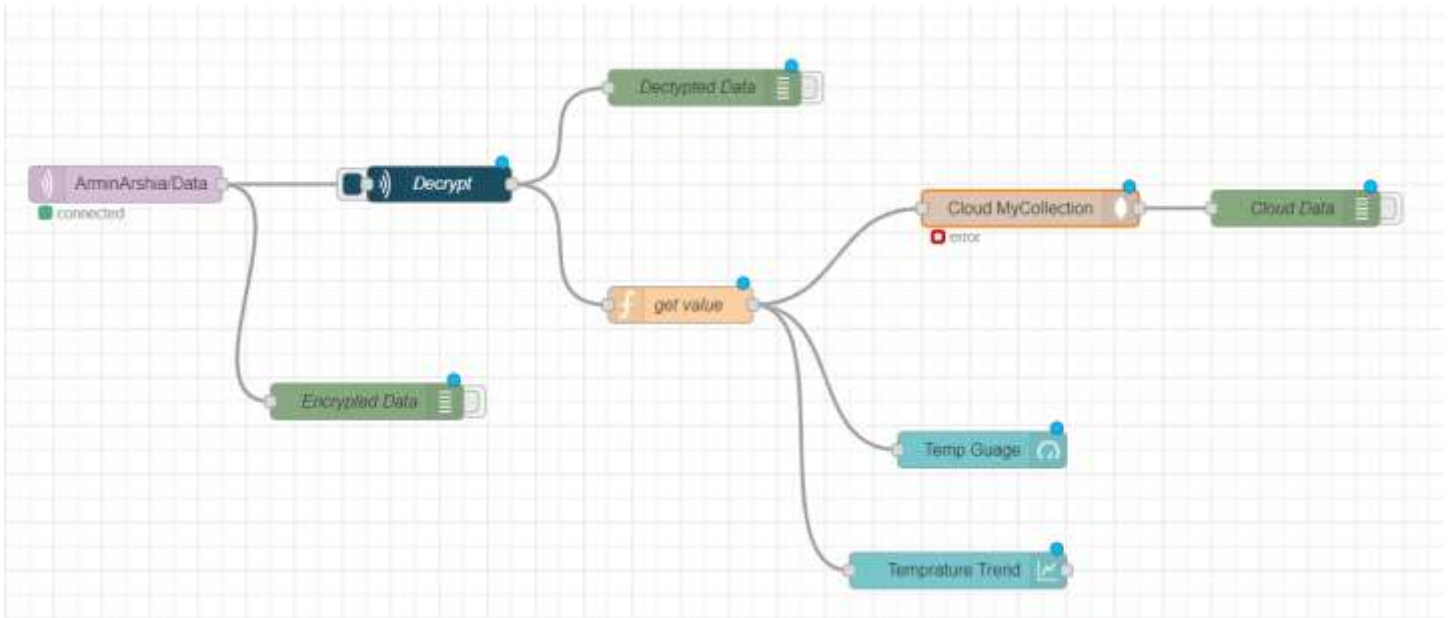
بخش پنجم: رمزنگاری و رمزگشایی داده ها



به منظور امنیت و بالا بردن قابلیت اطمینان سیستم، قبل و بعد از اتصال به سرور های MQTT داده ها به ترتیب رمزنگاری و در سمت دریافت، رمزگشایی می شود. الگوریتم های متعددی تست و ارزیابی شد که در نهایت الگوریتم رمزگشایی RC4 در این پروژه مورد استفاده قرار گرفت. هر دو سمت مبدا و مقصد با دانستن گذرواژه یا Key می توانند به داده ها دسترسی داشته باشند.

دو کد مربوط به `Decrypt.py` و `Encrypt.py` با الگوریتم RC4 موجود است که مورد استفاده قرار گرفته است. در سمت `Getaway` (برد رزبری پای) رمزنگاری صورت می گیرد و در `Node-RED` داده ها رمزگشایی و مورد استفاده قرار می گیرد. تصویری از نحوه رمزگشایی به صورت زیر نشان داده می شود:

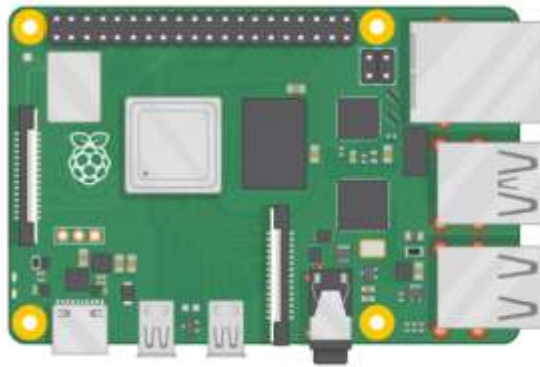




بلوک‌های Node-red

این فایل تحت عنوان `final_nodered` ذخیره شده است.

بخش ششم: رزبری پای به عنوان Gateway



در اولین بخش امتیازی از یک رزبری پای به عنوان یک Gateway استفاده شده است. این امر امکان اتصال چندین ESP32 را به طور همزمان به پروژه می‌دهد. وظیفه‌ی رزبری در این بخش به شرح زیر است:

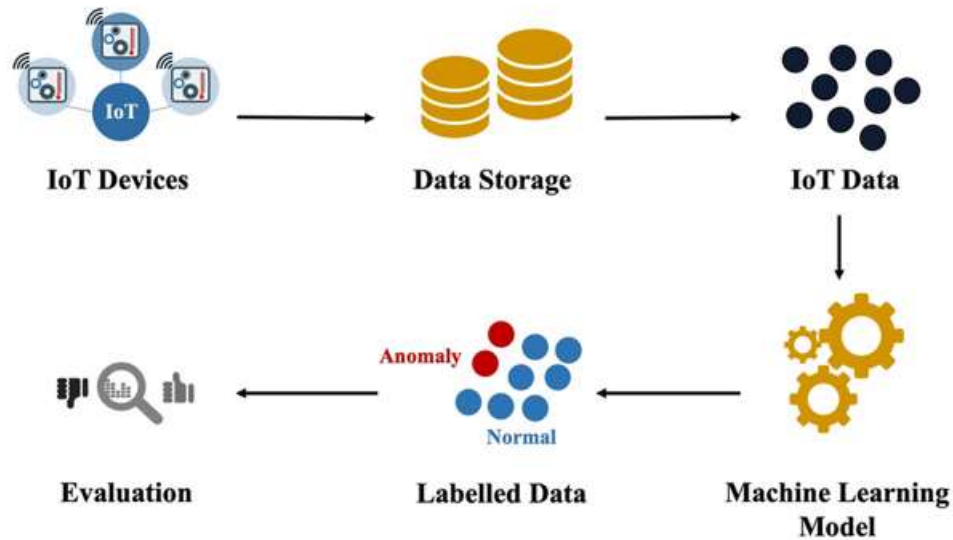
(۱) دریافت داده از بلوتوث (کتابخانه‌ی Bluepy)

(۲) رمزنگاری داده‌ها توسط RC4 (پیاده‌سازی توسط توابع خود پایتون)

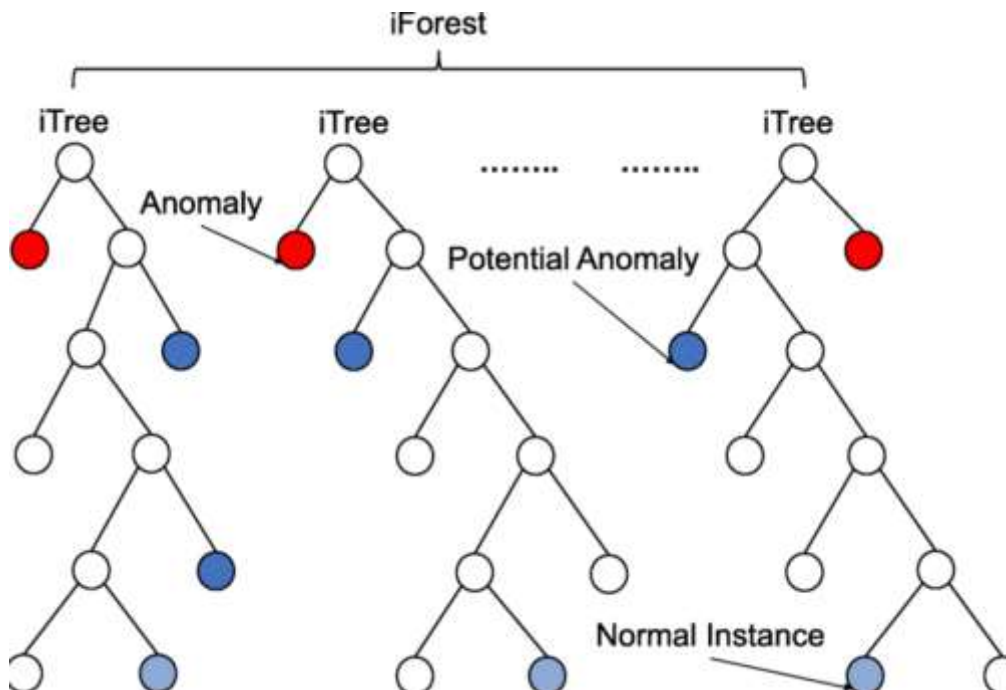
(۳) ارسال داده‌ها بر روی Cloud (کتابخانه‌ی paho-mqtt)

کد برنامه‌ی این بخش در فایل `sendToCloud.py` موجود است.

بخش هفتم: یادگیری ماشین و تحلیل داده



روند کلی پردازش داده برای یادگیری ماشین



به عنوان بخش امتیازی دوم پروژه، برای تحلیل داده ها از الگوریتم های یادگیری ماشین استفاده شد. هدف پروژه تشخیص ناهنجاری ها در روند زمانی داده های دما است. از متداول ترین الگوریتم ها برای این کاربرد می توان به Isolation Forrest اشاره کرد که یک دسته بندی میان داده ها انجام می دهد تا داده های ناهنجار را از داده های نرمال تشخیص دهد. این الگوریتم نیاز به Label ندارد و به صورت Unsupervised Learning اجرا می شود.

نحوه پیاده سازی به صورت شبه زمان حقیقی است و هر ۱۰ ثانیه داده ها از روی پایگاه داده خوانده شده و در گام بعدی الگوریتم یادگیری ماشین تشخیص ناهنجاری را انجام می دهد.

```
while True:
    # Read temperature data
    temperature_data = read_temperature_data()

    # Perform anomaly detection and visualization
    perform_anomaly_detection(temperature_data)

    # Wait for 10 seconds
    time.sleep(10)
```

کدهای مربوطه در محیط Google Colab زده شده و در پوشه همراه این گزارش قابل مشاهده است.

نتیجه نهایی با تحلیل حدود ۷۰۰۰ داده به صورت زیر خلاصه می شود:

