Auteur: Max Huiskes (2151960)

Klas: 32022INF1

Vak Beroepsproduct 5/6

School: Avans Stratenlaan te Den Bosch

Datum: 24-10-2023

Versie: 1.0

Onderbouwde infrastructuur data

Inhoudsopgave

[Inleiding 2](#_Toc149059326)

[Data pipeline 3](#_Toc149059327)

[Ingestion 3](#_Toc149059328)

[Lora Gateway/The Things Network 4](#_Toc149059329)

[FIWARE 4](#_Toc149059330)

[Node-Red 4](#_Toc149059331)

[Dashboard 4](#_Toc149059332)

[Dataverwerking 5](#_Toc149059333)

[Data ingestion 6](#_Toc149059334)

[Data transformation 6](#_Toc149059335)

[Samenvatting 7](#_Toc149059336)

[Bijlage A 8](#_Toc149059337)

# Inleiding

Dit artikel beschrijft een end-to-end datapijplijn voor het verzamelen, verwerken en presenteren van gegevens van verschillende sensoren, met een focus op voorspellende analyses. De pijplijn begint met sensoren die via LoRa verbonden zijn met een Arduino Uno. Temperatuur, gewicht, beweging en radarinformatie worden allemaal gemeten door deze sensoren. De gegevens worden verzameld door de Arduino en vervolgens geformatteerd in een geschikt formaat voor LoRa-transmissie.

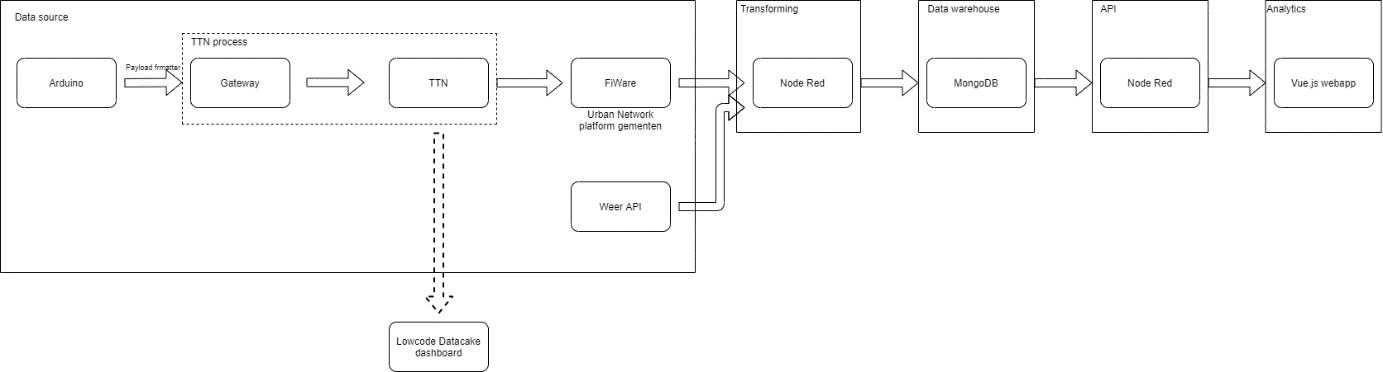
De gegevens worden verzonden van de Arduino naar The Things Network via een LoRa Gateway. Het Things Network functioneert als een datatransferstation. De ontvangen informatie is in hexadecimale indeling en moet worden gedecodeerd. Deze decodering wordt uitgevoerd met behulp van een payload-formatter, die de gegevens structureert tot leesbare informatie.

Vervolgens worden de gedecodeerde gegevens verzonden naar een FIWARE-database van de gemeente 's-Hertogenbosch. Binnen FIWARE is een REST API geïmplementeerd, waardoor de data toegankelijk is voor verdere verwerking. Deze verwerking vindt plaats in Node-Red, waar gegevenstransformaties, geavanceerde formules en algoritmen worden gebruikt om de gegevens op te slaan in een MongoDB-database. Verder wordt machine learning gebruikt om forecasting te doen op basis van de verstrekte informatie.

Node-Red dient als een cruciale stap in het gegevensverwerkingsproces en transformeert onbewerkte gegevens zodat deze klaar zijn voor opslag en analyse. Machine learning wordt gebruikt om voorspellingen te doen, bijvoorbeeld over het gedrag van vogels op basis van sensordata. Alle gegevens en voorspellingen worden in realtime weergegeven in een dashboard voor gebruikersgemak.

Dataverwerking omvat periodiek gegevensinname, batchverwerking, opslag, gegevensanalyse, voorspellingen, opslag van voorspellingen, en een feedbackmechanisme voor continu verbetering van voorspellingsmodellen.

# Data pipeline



Zie bijlage A als voor het bestanden van de afbeelding.

# Ingestion

Als eerste beginnen we met een Arduino uno met een Lora-shield, hier aan zijn vier sensoren aan gesloten. Deze sensoren zijn een gewicht sensor, een temperatuur sensor, een ir breaker sensor en een radar sensor. De data van de sensoren worden opgehaald door de Arduino en getransformeerd naar de juiste payload format. De data die wordt opgehaald wordt als volgende in de payload gezet en verzonden door de volgen de functie:

void do\_send(osjob\_t\* j) {

  if (LMIC.opmode & OP\_TXRXPEND) {

    Serial.println(F("OP\_TXRXPEND, not sending"));

  } else {

    int16\_t temp = tempSensor.getTemperature() + 127;

    uint16\_t gewicht = gewichtSensor.getWeight();

    uint8\_t battery = (int)bat.checkBattery();

    byte payload[8];

    int cursor = 0;

    payload[cursor++] = highByte(temp);

    payload[cursor++] = lowByte(temp);

    payload[cursor++] = highByte(gewicht);

    payload[cursor++] = lowByte(gewicht);

    payload[cursor++] = byte(uit);

    payload[cursor++] = byte(in);

    payload[cursor++] = byte(radar);

    payload[cursor++] = byte(battery);

    LMIC\_setTxData2(1, payload, cursor, 0);

    Serial.println(F("Packet queued"));

    uit = 0;

    in = 0;

    radar = 0;

  }

}

## Lora Gateway/The Things Network

De data die de Arduino verstuurd gaat naar The Things Network waar deze zichtbaar. Dit gaat via een Lora Gateway, via deze gateway wordt de data als hexadecimale getalen doorgestuurd. Deze data moeten worden “vertaald” om het leesbaar te maken en dit doen we door een payload formatter. Zie hieronder voor payload formatter.

function decodeUplink(input) {

  return {

    data: {

      Temp: input.bytes[0] << 8 | input.bytes[1] - 127,

      Weight: input.bytes[2] << 8 | input.bytes[3],

      Out: input.bytes[4],

      In: input.bytes[5],

      Radar: input.bytes[6],

      Battery: input.bytes[7]

    },

    warnings: [],

    errors: []

  };

}

## FIWARE

De FIWARE-database is van de gemeente ’s-Hertogenbosch. De bedoeling is om hier alle data op te slaan als een grote datawarehouse. Binnen FIWARE is mogelijk om een Rest API te genereren. Dit is belangrijk voor de mogelijkheid om machine learning toe te passen. Dit doen we in een Node-Red applicatie via wiskundige formules en algoritmes. Dit wordt opgeslagen binnen een MongoDB-database.

## Node-Red

Node-Red wordt gebruikt om de data te transformeren zodat deze in de MongoDB-database kan opslaan. Verder kan ik de machine learning toe passen om een voorspelling te maken. Bijvoorbeeld als het regent vliegen de vogels minder in en uit het vogelhuisje dan met mooi weer. Uit eindelijk wordt de data weer gegeven in een dashboard.

## Dashboard

Op het dashboard laat ik alle data van de sensoren en de voorspellingen zien.

# Dataverwerking

Binnen de Arduino applicatie werken we met batch processing. Van iedere 5 minuten sturen we de data via the things network naar FIWARE. Dit proces gaat als volgende:

* Data ingestion,
  + Door elke 5 minuten te versturen krijgen we geen te grote pakketten, die we moeten versturen.
* Batch verwerking,
  + Na 5 minuten wordt de verzamelde data ingevoerd in een verwerkingssysteem.
* Dataopslag,
  + De opgeslagen data worden verwerkt en klaargezet om voorspellingen te doen en te verbeteren.
* Data-analyse,
  + In de data-analyse worden er voorspellingen gedaan voor de in- en uitvluchten, temperatuur en de vlucht erlangs. Hierbij worden er verschillende technieken zoals machine learning toegepast.
* Voorspelling,
  + De voorspellingen worden gemaakt voor een hele week. Dit wordt gedaan op verwachtingen en patronen en trends die zijn vastgesteld.
* Opslag,
  + De voorspellingen worden opgeslagen en deze worden gebruikt voor toekomstige voorspellingen.
* Feedback en verbetering.
  + Als er meer voorspellingen worden gedaan hebben we meer vergelijkingspunten om te kijken of dat deze data goed is en hiermee verbeteren we de voor spellingen.

# Data ingestion

Data ingestion begint bij de Arduino uno Rev3, hart van het dataverzamelingssysteem. Hier zitten de sensoren op aangesloten. Deze zijn een Temperatuur sensor, Gewichtssensor, Radar sensor en IR Breaker sensor om informatie van vogels te meten.

Lora shield is een Arduino-extensie waarmee u gegevens kunt verzenden en ontvangen via het Lora-communicatieprotocol. LoRa staat bekend om zijn vermogen om gegevens over lange afstanden te verzenden. Het LoRa-schild stuurt het naar een LoRa-gateway of -ontvanger binnen een paar kilometer. De Arduino is verantwoordelijk voor het lezen van gegevens van sensoren en het verzenden ervan naar de gateway.

In een Cloud omgeving worden de ontvangen gegevens verwerkt, opgeslagen en geanalyseerd. Hier kunnen we afbeeldingen en algoritmen toepassen op basis van de verstrekte informatie.

# Data transformation

Onze data transformatie vindt plaats aan de kant van waar we de data opvangen voor het de database in gaat. Verder aan de Arduino hebben we het volgende gedaan.

 int16\_t temp = tempSensor.getTemperature() + 127;

We tellen 127 op bij de temperatuur sensor. Dit doen we want de data wordt verstuurd als een unsigned 16 bits integer en het laagste getal dat hij kan versturen is -127. -127 is een error code van de sensor de minimale waarde van de sensor is -55. Aan de andere kant vang ik dit af door het getal min 127 te doen.

# Samenvatting

Dit artikel beschrijft een datapijplijn voor het verzamelen, verwerken en verspreiden van data van verschillende sensoren, met als doel aanbevelingen en inzichten te genereren. De datapijplijn begint met sensoren die via LoRa verbonden zijn met een Arduino Uno. Voordat ze via LoRa naar The Things Network worden verzonden, worden de sensorgegevens verzameld en georganiseerd in een geschikt formaat.

De gegevens worden vanuit The Things Network naar de FIWARE-database van de gemeente 's-Hertogenbosch gestuurd. Binnen FIWARE is een REST API geïmplementeerd, waardoor de data toegankelijk is voor verdere verwerking. Deze groenere verwerking wordt uitgevoerd met behulp van Node-Red, dat geavanceerde formules en algoritmen gebruikt om gegevens op te slaan in een MongoDB-database. Machine learning wordt ook gebruikt voor voorspellende analyses.

Node-Red speelt een belangrijke rol in de transformatie van data zodat deze geschikt is voor opslag in een database. Machine learning wordt gebruikt om taken uit te voeren zoals het voorspellen van het gedrag van vogels op basis van historische gegevens. Alle gegevens en instructies worden in realtime opgeslagen op een dashboard.

Het gegevensverwerkingsproces omvat periodieke gegevensinvoer, batchverwerking, gegevensanalyse, voorspelling, voorspellingsopslag en feedback voor het verbeteren van voorspellingsmodellen.

# Bijlage A

