

## **HERÄTEVASTEANALYYSIOHJELMISTON KEHITYS**

Ville-Valtteri Yritys  
Opinnäytetyö  
Kevät 2025  
Tietotekniikan tutkinto-ohjelma  
Oulun ammattikorkeakoulu

# TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Tietotekniikan tutkinto-ohjelma  
Ohjelmistokehitys

Tekijä: Ville-Valtteri Yritys  
Opinnäytetyön otsikko: Herätevasteanalyysiohjelmiston kehitys  
Työn ohjaaja(t): Jukka Nevalainen  
Työn valmistumislukukausi ja -vuosi: Kevät 2025  
Sivumäärä: 25

Opinnäytetyössä oli tarkoitus toteuttaa analyysiohjelmisto, joka pystyy tunnistamaan äänisignaaleissa tapahtuvat herätteet ja sen vasteet. Tämä analyysiohjelma auttaa tunnistamaan poikkeuksia laitteistoissa ja edesauttaa laitteiston kestoikää.

Tärkein tavoite tässä opinnäytetyössä oli tunnistaa heräte sekä vaste. Herätteen ja vasteen tunnistettua näillä voitiin luoda algoritmi, joka voi oppia sekä verrata mittadataa. Opetteluvaiheessa algoritmi pystyy opettelemaan tietyt raja-arvot. Vertailuvaiheessa algoritmi käyttää opittuja raja-arvoja ja vertailee niitä mittadatan.

Toteutuksen yhteydessä asennettiin testilaitteisto, jolla pystyttiin simuloimaan lyönnistä syntyvät herätteet ja vasteet.

Testaaminen sekä suunnittelu oli tärkeä osa tätä prosessia, jotta voitiin luoda luotettava algoritmi. Testausvaiheessa käytettiin Nome Oy:n omia mittalaitteistoja kuten mikrofonia, äänikorttia sekä tietokonetta.

Opinnäytetyö oli toteutettu yhteistyössä Nome Oy:n kanssa.

## **ABSTRACT**

Oulu University of Applied Sciences  
Degree Programme in Information technology  
Software development

Author(s): Ville-Valtteri Yrity  
Title of thesis: Trigger-Response Analysis Software  
Supervisor(s): Jukka Nevalainen  
Term and year when the thesis was submitted: spring 2025  
Number of pages: 25

The goal of this thesis project was to develop an analysis software capable of detecting triggers and responses within audio signals. This software helps identify anomalies in hardware and contributes to the longevity of devices.

The main objective was to identify both the trigger and the response. Once these were detected, an algorithm was created to learn and compare measurement data. During the learning phase, the algorithm learns specific threshold values, and in the comparison phase, it uses these learned thresholds to compare against the measured data.

The implementation involved setting up a test system that could simulate the triggers and responses generated by impacts. Testing and design were essential in ensuring a reliable algorithm. During the testing phase, custom measurement equipment, including microphones, an audio interface, and a computer, were used.

This thesis was carried out in collaboration with Nome Oy.

# SISÄLLYS

TIIVISTELMÄ .....	2
ABSTRACT .....	3
SISÄLLYS .....	4
1 JOHDANTO .....	5
2 YMPÄRISTÖ .....	6
2.1 Teollisuus .....	6
2.2 Teollisuuden kunnonvalvonta .....	6
3 ERILAISET MENETELMÄT .....	8
3.1 Valvottu koneoppiminen .....	8
3.2 Valvomaton koneoppiminen .....	9
4 LAITTEET SEKÄ OHJELMISTOT .....	11
4.1 Tietokone (Raspberry).....	11
4.2 Äänikortti .....	12
4.3 Mikrofoni.....	12
4.4 IoT Viestintä (MQTT).....	13
4.5 Ohjelmointi (Python).....	14
5 TOTEUTUS .....	17
5.1 Äänen visualisointi suunnittelu .....	17
5.2 Analyysiohjelmiston toteuttaminen .....	18
5.3 Analyysiohjelmiston testauksen alkuvaihe .....	19
6 TESTAUS .....	21
7 YHTEENVETO .....	22
LÄHTEET .....	23

# 1 JOHDANTO

Tämä opinnäytetyö keskittyi pääomaisesti äänen analyysiin, jolla voidaan ottaa huomioon erilaisia äänisignaaleissa tapahtuvia ilmiöitä. Tärkeä osa tämän ohjelmiston rakennetta on äänite, mittauslaitteet, ohjelmisto ja niiden menetelmät. Ääni voidaan lukea näytteisiin käyttämällä erilaisia ohjelmointikirjastoja. Tässä opinnäytetyössä keskityttiin herätteen sekä vasteen analysointiin äänessä. Herätteellä tarkoitetaan, kun kone voi tehdä mekaanista värähtelyä, joka toimii herätteenä (Hentinen ym. 30.04.2002, 16). Vaste tarkoittaa, kun heräte alkaa vaikuttamaan rakenteeseen, jolloin se alkaa värähtelemään tietyllä tavalla (Lamula & Saarinen 29.10.2019, 2).

Tavoitteena oli luoda teollisuuden kunnonvalvontaan toimiva algoritmi, joka huomioi herätteen sekä vasteen äänisignaaleissa. Tämän avulla voidaan analysoida teollisuudessa olevia laitteistoja ja mahdollisesti ennaltaehkäistä alkavia vikoja.

Merkitys sovelluksesta on laaja. Sovelluksella voidaan huomata erilaisia ilmiöitä, joita ihminen ei välttämättä itse huomaa. Esimerkki tällaisesta ilmiöstä voi olla koneiston äänen muuttuminen ajan kanssa. Äänen muuttumisen huomioiminen voi vaikuttaa koneen kestoikään.

Tämä opinnäytetyö toteutettiin Nome Oy:lle. Nome Oy on suomalainen yritys, joka tarjoaa teollisuuden kunnonvalvontapalveluita sekä ohjelmisto- ja tuotekehitystä.

## 2 YMPÄRISTÖ

Tässä kappaleessa käydään läpi teollisuuteen ja teollisuuden kunnonvalvontaan liittyvät käsitteet. Alaluvuissa selitetään opinnäytetyöhön liittyvää analysoinnin aihealuetta. Myös alaluvussa käydään läpi äänen analysointia.

### 2.1 Teollisuus

Teollisuus on tuotantotoimintaa, jolla voidaan jalostaa raaka-aineita tuotteiksi. Pääomaisesti nykyään teollisuudessa käytetään sähkötoimisia koneita. Teollinen valmistus on riippuvainen tehtaan tuotantotekniikasta. (Wikipedia, 3.2.2024.) Wikipedian (3.2.2024) mukaan on olemassa useita eri teollisuuden toimialoja, esimerkiksi

- elintarviketeollisuus
- kaivosteollisuus
- kemianteollisuus
- lasiteollisuus
- lääketeollisuus
- metalliteollisuus.

Tässä opinnäytetyössä tehty analysointityökalu toteutettiin voimalaitoksen osaprosessiin. On tärkeä ymmärtää teollisuuden alasta, jotta voidaan luoda oikeanlainen työkalu ja sen ohjelmisto. Teollisuudessa voidaan käyttää monenlaisia eri analyysimetodeja, jotta voidaan varmistaa laitteiston toimivuus.

### 2.2 Teollisuuden kunnonvalvonta

Teollisuuden kunnonvalvonta on joukko teknisiä mittausmenetelmiä, joilla voidaan määritellä koneen tai laitteiston kunto käynnin aikana. Tämä mahdollistaa alkavan vian ennakoimisen, jolla voidaan välttää laiterikosta syntyvät seisokit. (MIP 2025.)

Yleisiä kunnonvalvonnan mittaus menetelmiä esimerkiksi laakereille ovat muun muassa akustinen värähtely, lämpötila sekä mekaaninen värähtely (Verma, A 2014).

Tässä opinnäytetyössä keskitytään erityisesti äänianalyysiin, jossa tutkitaan koneesta syntyviä herätteitä, kuten lyöntejä tai kolahduksia, sekä niihin liittyviä vasteita, eli miten kone reagoi näihin ääni-ilmiöihin. Herätteiden ja vasteiden avulla voidaan tunnistaa erilaisia alkavia vikoja, esimerkiksi jos vaste tai heräte on ollut erilainen vuosi sitten verrattuna nykyhetkeen. Herätteen tai vasteen erilaisuus voi tarkoittaa, että laitteistossa on tapahtunut jonkinlainen poikkeama.

Tärkeä on kuitenkin ymmärtää mahdolliset analysointimetodit. Seuraavissa luvuissa käydään läpi opinnäytetyössä käytettyjä analysointiin liittyviä menetelmiä.

## 3 ERILAISET MENETELMÄT

Tärkeänä osana analyysin tekemistä on oikeanlainen tapa tutkia erilaisia menetelmiä ja toteutuksia, joita voidaan hyödyntää tämän tyyppisessä analyysin tekemisessä. Tässä kappaleessa käydään läpi analysointiin liittyviä menetelmiä.

### 3.1 Valvottu koneoppiminen

Valvottu koneoppiminen on algoritmi, joka voidaan kouluttaa ennustamaan erilaisia arvoja. Algoritmille voidaan antaa tietty määrä erilaisia tietoaineistoja, joita voidaan käyttää ennustamaan tuloksia ja tunnistamaan malleja. Valvomattomalle koneoppimiselle taas ei anneta tiettyä tiedettyä tietoaineistoa. (Google Cloud s.a.) Valvomattomasta koneoppimisesta käydään lisää läpi luvussa 3.2.

Valvotut algoritmit mahdollistavat organisaatioita luomaan monimutkaisia malleja, joilla he pystyvät ennustamaan tarkkoja arvoja. Monilla toimialoilla kuten esimerkiksi terveydenhuollossa, markkinoinnissa sekä rahoituspalveluissa käytetään valvottua koneoppimista. (Google Cloud s.a.)

Esimerkki koneoppimisen hyödystä: Mikäli malli halutaan opettaa tunnistamaan kuvasta puu, voidaan mallille antaa nimetty tietojoukko erilaisista puista ja niiden lajien nimet. Tämän jälkeen voit antaa mallin arvata, mikä puu on kyseessä. (Google Cloud s.a.)

Valvottu koneoppiminen jaetaan yleensä kahteen eri kategoriaan: Luokiteltu ja regressio. (Google Cloud s.a.)

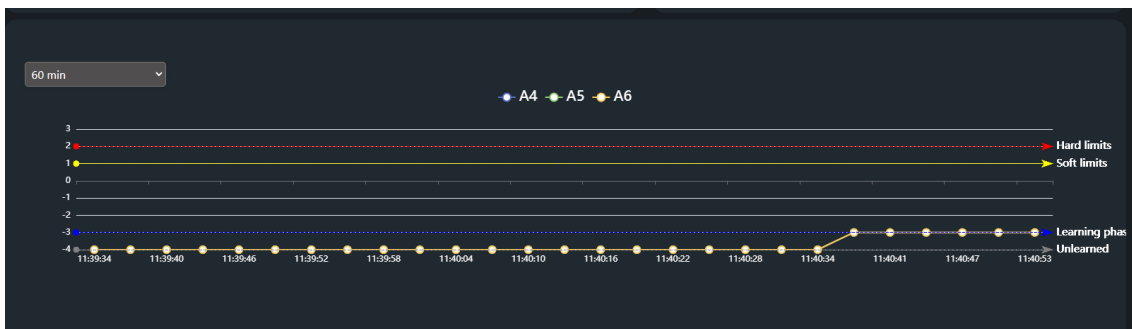
Luokittelua voidaan käyttää tietojen ryhmittelyyn. Luokittelu voi ennustaa kategoriaita syöttötietojen perusteella. Regressiota käytetään ennustamaan todellisia tai jatkuvia arvoja, jotka algoritmi voi tunnistaa. Regressio havaitsee suhteen kahden tai useamman muuttujan välillä. (Google Cloud s.a.)

Valvotun koneoppimisen vahvuutena on sen korkea tarkkuus ja selkeä ennustettavuus, kun käytettävissä on hyvin merkitty opetusdata. Tämä tekee siitä erinomaisen työkalun esimerkiksi teollisuuteen, missä laiterikko voi aiheuttaa esimerkiksi menetettyä tuotantoa.

Haasteena on opetusdatan laatu ja määrä. Mallin tehokkuus riippuu suuresti siitä, kuinka paljon dataa on ja kuinka tarkasti data on opetettu. Tällaisen datan kerääminen ja esikäsittely voi olla aikaavievää ja kallista.

Tässä opinnäytetyössä käytettiin valvottua koneoppimista sekä valvomatonta koneoppimista. Tällöin pystyttiin analysoimaan ääntä, tapahtuuko siinä erilaisia ilmiöitä sekä mitkä olivat äänessä tapahtuvat herätteet ja niiden vasteet.

Nome Oy oli suunnitellut ja toteuttanut tietynlaisen valvotun algoritmin, jolla voidaan opettaa sekä tunnistaa erilaisia äänessä tapahtuvia ilmiöitä (kuva 1).



Kuva 1. Havainnollistaminen oppimisvaiheesta.

### 3.2 Valvomaton koneoppiminen

Valvomaton koneoppiminen on osa koneoppimista, jolla voidaan oppia datasta ilman ihmisen valvontaa. Valvomaton koneoppiminen eroaa valvotusta koneoppimisesta. Valvomattomalle koneoppimiselle voidaan antaa nimeämätöntä dataa, jolla voidaan löytää erilaisia oivalluksia ilman erityisiä ohjeita. (google s.a.)

Valvomaton koneoppiminen käyttää itseoppivia algoritmeja, joilla voidaan opettaa tunnistamaan tunnistamattomia tietoja. Esimerkiksi mallille voidaan antaa raakoja nimeämättömiä tietoja, ja algoritmin on tämän avulla pääteltävä omat sääntönsä. (google s.a.)

Valvomattomat algoritmit sopivat hyvin monimutkaisiin käsittelytehtäviin. Malli on hyödyllinen havaitsemattomien tietojen tunnistamisessa. Tätä mallia voidaan käyttää tietojen luokittelussa. (google s.a.)

Esimerkki valvomattoman koneoppimisen hyödyistä: Mikäli mallilla on suuri tietojoukko säästä, valvoton malli voi käydä tiedot läpi ja identifioida datapisteiden kuviot. (google s.a.)

Valvoton koneoppiminen on hyödyllinen erityisesti tilanteissa, joissa datan manuaalinen luokittelu on aikaa vievää tai mahdotonta.

Tässä työssä käytettiin valvomatonta koneoppimista äänessä tapahtuvien piikkien ja muiden poikkeamien automaattiseen tunnistamiseen, minkä jälkeen valvottua oppimista hyödynnettiin niiden luokittelussa ja jatkoanalyysissä.

## 4 LAITTEET SEKÄ OHJELMISTOT

Tässä kappaleessa käydään läpi tarvittavia laitteita, menetelmiä ja ohjelmistoja. Nämä ovat tärkeitä luotettavan ohjelmiston luomiseksi. Ohjelmistolla voidaan analysoida erilaisia ääninäytteitä. Alaluvuissa käydään läpi käytettyä laitteistoa, tärkeitä menetelmiä sekä ohjelmointikieltä, jota opinnäytetyössä käytettiin. Alaluvuissa kerrotaan myös mahdollisista puutteista sekä hyödyistä.

### 4.1 Tietokone (Raspberry)

Raspberry on yhdestä piiristä koostuva tietokone, joka toimii samalla periaatteella kuin perinteinen tietokone. Raspberry oli aluksi suunniteltu tietojenkäsittelyyn. Raspberry tuli erittäin halutuksi sen hinnan takia ja kuinka kätevän kokoinen se on. Raspberry oli erittäin kysytty varsinkin harrastelijoiden puolesta. Alkuperäisessä Raspberrissä oli 512 megatavua keskusmuistia, mutta nykyisissä malleissa saattaa olla jopa 16 gigatavua. (Wikipedia 2011.) Wikipedia (2011) mukaan Raspberrystä on useita eri versioita kuten

- Raspberry Pi Model B+, julkaistu vuonna 2012
- Raspberry Pi 2 B, julkaistu vuonna 2015
- Raspberry Pi 3 Model B, julkaistu vuonna 2016
- Raspberry Pi 3 Model B+, julkaistu vuonna 2018
- Raspberry Pi 4 Model B, julkaistu vuonna 2019
- Raspberry Pi 5, julkaistu vuonna 2023.

Raspberry Pi on ollut keskiössä lukuisissa tutkimuksissa ja kehitysprojekteissa. Esimerkiksi valvontajärjestelmässä, jossa Raspberry Pi valvoo teollisuusympäristöä hyödyntäen lämpötila-antureita, sormenjälkitunnistusta ja Python-pohjaisia ohjelmia tiedon keruuseen ja ohjaukseen (Dency & Swetha, 2012).

Tässä opinnäytetyössä käytettiin Raspberry Pi 4. Raspberry on erittäin monipuolinen tietokone. Raspberrissä voi olla myös mahdollisia haasteita. Haasteita voi olla varsinkin suoritustehossa, jos sovellus on vaativa ja se tarvitsee paljon

keskusmuistia tai prosessorin ytimiä. Tällöin Raspberry ei välttämättä pysty ajamaan ohjelmaa kunnolla suoritustehon takia.

## 4.2 Äänikortti

Äänikortti on tietokonelaitteisto. Äänikortilla voidaan syöttää äänisignaali tai lähettää se tietokoneelle. Äänikortit ovat suunniteltu äänen toistamiseen sekä taltiointiin sillä ne pystyvät tarjoamaan matalan latenssin, kun muutetaan äänisignaali analogisesta digitaaliseen. Äänilaitteessa olevat audioliitännät voivat sisältää sekä mikrofoniin esivahvistimia että analogisia linjatuloja. Äänilaite saattaa sisältää myös digitaalisia tuloja, joita ovat: DI, ADAT- tai S/PDIF (Wikipedia 22.01.2025.)

Adam Kagen (16.02.2021) mukaan hyötyjä äänikorteissa on

- Voi käyttää mitä tahansa XLR mikrofonia
- Voi käyttää useampaa mikrofonia
- Mahdollistaa äänen säätämisen suoraan äänikortissa.

Analyysia tehdessä äänikortissa pitää huomioida esimerkiksi näytteenottotaajuus. Näytteenottotaajuus määritellään siten, että kuinka monta näytettä voidaan ottaa sekunnissa. Ääninäytteet mitataan kilohertseinä (kHz). (Stewart 04.12.2024.)

Opinnäytetyössä käytettiin enimmäkseen neljäkanavaista äänikorttia, joka mahdollisti usean mikrofoniin kytkemisen yhtäaikaaisesti. Äänikortti mahdollisti laitteiston tarkkailun useasta eri kohdasta.

## 4.3 Mikrofoni

Mikrofoni on laite, jonka avulla voidaan muuttaa ääniaaltoja signaaliksi. Sillä voidaan nauhoittaa äänitteitä esimerkiksi tietokoneelle. On olemassa monenlaisia mikrofoneja, kuten dynaamisia mikrofoneja, kondensaattorimikrofoneja, nauhamikrofoneja sekä haulikkomikrofoneja. Oikean mikrofoniin valinta riippuu siitä,

mihin tarkoitukseen sitä halutaan käyttää. Esimerkiksi kondensaattorimikrofoni on hyvä studiokäyttöön sen korkean herkkyyden ja tarkkuuden takia. (Lenovo 2025.)

Tässä opinnäytetyössä käytetään Nome Oy:n suunnittelemaa ja toteuttamaa mikrofonia, joka pystyy tunnistamaan äänet tarkasti. Tarkkuuden takia sitä voidaan käyttää hyvin analysoinnissa. Tällä mikrofonilla pystytään tunnistamaan jopa 80 kHz:n taajuudella ääntä. Äänitiedosto voidaan tallentaa 192 kHz:n näytteenotteenotaajuudella ja 24 bitin resoluutiolla. Tällä mikrofonilla voidaan tunnistaa esimerkiksi rakenteissa johtuvia ääniä (kuva 2).



*Kuva 2. Nome Oy:n suunnittelema ja toteuttama mikrofoni.*

#### **4.4 IoT Viestintä (MQTT)**

MQTT on tohtori Andy Standard-Clarkin ja Arlen Nipperin suunnittelema IoT-protokolla. MQTT oli pääomaisesti suunniteltu öljy- ja kaasuteollisuuteen, sillä se mahdollisti datan lähettämisen etäisille palvelimille. MQTT käyttää PUBLISH SUBSCRIBE topologiaa, siinä on kahden tyyppisiä järjestelmiä: asiakkaat tai välittäjät. Välittäjä voi vastaanottaa viestejä ja lähettää ne eteenpäin. Asiakkaat taas vastaanottavat viestejä, mutta eivät kommunikoi toistensa välillä. MQTT lähettää viestejä vain silloin kun on tarve. Tämä pitää lähetyksen minimoituna. Jokaisella lähetyksellä on maksimi hyötykuorma, joka on 256 megatavua. Hyötykuorman

kokoa on mahdollista myös säätää. QoS 0 on hyvä valinta tilanteisiin, joissa viestintä on luotettavaa, mutta rajoitettua. Vastaavasti QoS 2 on hyvä valinta tilanteisiin, joissa viestintä ei ole niin luotettavaa, mutta yhteys on hyvä. MQTT mahdollistaa neljä eri tapaa kommunikoida

- **Publish** – lähettää viestejä tiettyyn aiheeseen
- **Subscribe** – mahdollistaa kuuntelun tietystä aiheesta
- **PING** – voi olla hyödyllinen, jos haluaa tarkistaa, onko yhteys hyvin voimassa
- **Disconnect** – kuuntelija voi katkaista yhteyden kuuntelusta

MQTT myös mahdollistaa turvallisuuden välittäjän ja asiakkaan välillä. (Paessler 2025.)

MQTT on suunniteltu yksinkertaiseksi, tehokkaaksi sekä resursseja säästäväksi. Verrattuna muihin protokolliin, MQTT:llä on pieni koodi jalanjälki, pieni yleiskustannus sekä pieni virrankulutus. MQTT on hyvä vaihtoehto siirrettäessä dataa verkon välityksellä. MQTT on myös erinomainen vaihtoehto laitteisiin, joissa on rajallinen suoritusteho. (HiveMQ, 2015.)

IoT protokollana käytettiin MQTT (Message Queuing Telemetry Transport). Tämä mahdollistaa raa'an datan sekä metadatan lähettämisen ääntä tarkkailtaessa. Mikäli äänessä tapahtuu jonkinlainen poikkeama, tai ilmiö, MQTT-viesti voidaan lähettää haluttuun paikkaan. Tätä viestiä voidaan analysoida tarkemmin jälkikäteen.

#### 4.5 Ohjelmointi (Python)

Python on ohjelmointikieli, joka on helposti opittava sekä tehokas. Pythonissa on korkean tason tietorakenne. Siinä on yksinkertainen lähestymistapa olio-ohjelmointiin. Python on myös vapaasti käytettävä, se on saatavilla melkein kaikilla alustoilla. (Python Software Foundation 13.02.2025.) Pythonissa on myös laaja määrä eri kirjastoja, joita voidaan käyttää hyväksi. Python-kirjasto on tiedostokoelma, jota voidaan käyttää ohjelmassa (Library Carpentry 11.05.2024).

(Pandey, P 10.2.2025) mukaan python soveltuu erityisen hyvin monenlaisiin käyttökohteisiin, kuten:

- Tekoäly ja koneoppiminen
- Tiedonkeruu ja -analyysi
- Web-kehitys
- IoT-sovellukset
- Automaatio

Amazon aws (2025) sivun mukaan hyötyjä Python ohjelmoinnissa on

- Pythonissa käytetään englannin tyyppistä syntaksia, joka mahdollistaa sen helpon ymmärrettävyyden
- Pythonissa kirjoitetaan vähemmän koodia verrattuna muihin ohjelmointikieliin
- Python tarjoaa laajan kirjastovalikoiman, minkä ansiosta koodia ei tarvitse kirjoittaa alusta asti itse.
- Ongelmatilanteissa python yhteisö on laaja, joka mahdollistaa avun saamisen helposti
- Python toimii monilla eri alustoilla, kuten: Windows, macOS, Linux ja Unix
- Laaja valikoima erilaisia tutoriaaleja, miten jokin asia voidaan tehdä.

Python ohjelmointikielenä on hitaampi, kuin C++ tai Java. Python suorittaa tulkin kääntäjän sijaan, kun taas C++ tai Java suorittaa kääntäjän. Käännetyissä kielissä kuten Javassa tai C++ sovelluskoodi käännetään alkuperäiseen järjestelmä koodiin, joka mahdollistaa paremman sovelluksen optimoinnin. (Malik. 16.7.2019.)

Pythonilla on monenlaisia suunnittelurajoituksia. Tämä johtuu siitä, että se kirjoitetaan dynaamisesti. Dynaaminen kirjoittaminen viittaa muuttujatyyppeiden arviointiin. Pythonilla tarvitsee enemmän aikaa sovellusten testaamiseen. (Malik 16.7.2019.)

Python mahdollisti äänen analysoinnin hyvin, koska sillä oli useita erilaisia kirjastoja. Käytettyjä kirjastoja oli esimerkiksi soundfile, joka mahdollisti äänitiedoston lukemisen numeraalisena. Projektissa täytyi myös suunnitella toiminnallisuus

hyvin, sillä useita eri funktioita piti ajaa yhtäaikaisesti analyysin takia. Tähän käytettiin Pythonin omaa threading kirjastoa. Threading mahdollistaa esimerkiksi usean funktion ajamisen yhtäaikaisesti ohjelmassa (Anderson. J 2025).

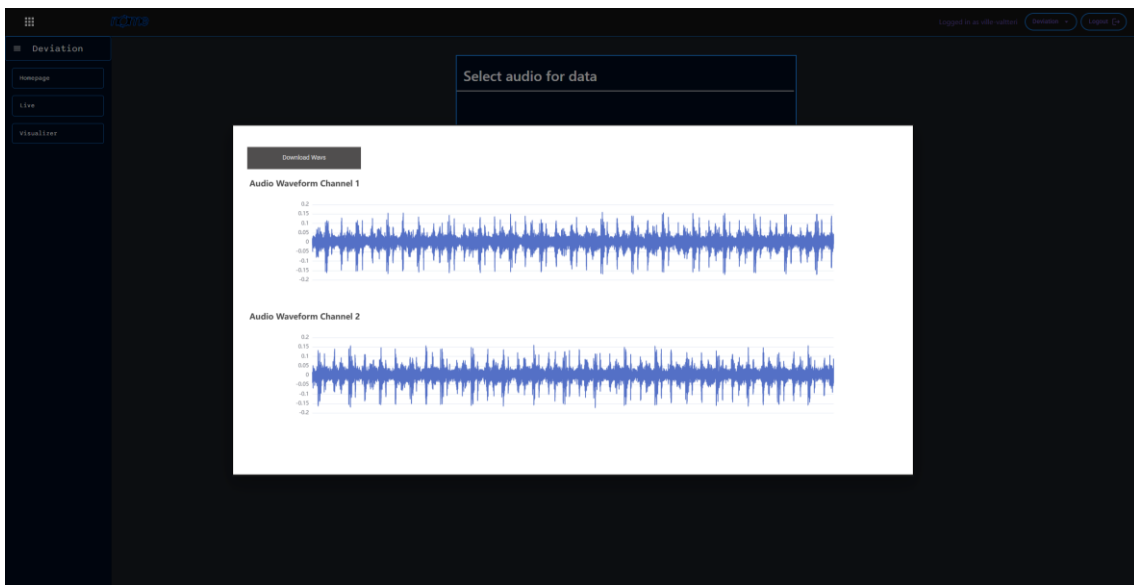
## 5 TOTEUTUS

Tässä kappaleessa käydään läpi toteutusvaiheita ja sen prosesseja.

### 5.1 Äänen visualisoinnin suunnittelu

Alkuvaiheessa toteutettiin yksinkertainen visualisointityökalu, jota voidaan käyttää havainnollistamaan äänessä tapahtuvat herätteet ja sen vasteet.

Kuvassa 3 voidaan havainnoida kahdelta mikrofonilta tulevat äänisignaalit. Kyseinen visualisointi on tehty luomalla yksinkertainen ohjelmisto käyttäen Python ohjelmointikieltä. Ohjelmistossa luettiin äänitiedosto (.wav) tiedostopolusta, jonka jälkeen se voitiin lähettää frontendiin käyttäen WebSocket protokollaa. Frontendissä äänitiedosto otettiin vastaan kuuntelemalla websockettia. Frontend oli ennuudestaan suunniteltu ja toteutettu Nome Oy:n toimesta. Tämä käyttöliittymäkuva (kuva 3) pohjautuu aiemmin toteutettuun käyttöliittymään, joka rakennettiin aikaisemmassa projektissa React.js-kirjastolla.



Kuva 3. Äänen havainnollistaminen äänitiedostosta kahdelta eri kanavalta.

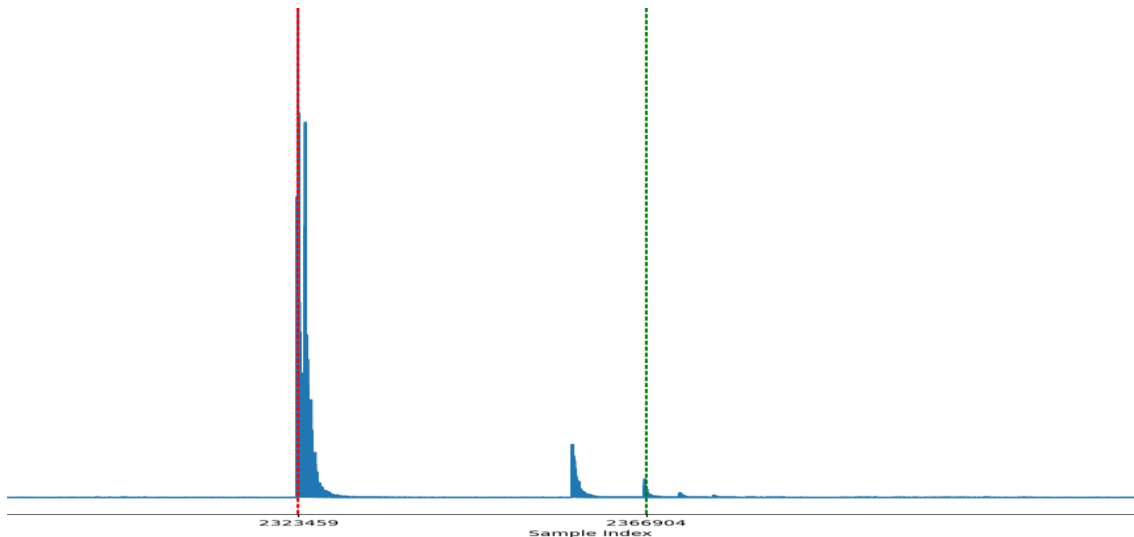
## 5.2 Analyysiohjelmiston toteuttaminen

Tässä kappaleessa käydään läpi toteutettua ohjelmistoa. Analyysiohjelmiston alkuvaiheessa luotiin yksinkertainen ohjelmisto Python ohjelmointikielellä. Tällä ohjelmistolla voitiin lukea äänitiedostoja ja visualisoida ne käyttämällä matplotlib kirjastoa (kuva 4).

```
1  import matplotlib.pyplot as plt
2  import numpy as np
3  import soundfile as sf
4
5  def read_wav(file_path):
6      audio_array, framerate = sf.read(file_path)
7      audio_array = np.gradient(audio_array)
8      return audio_array
9
10 plot_data = read_wav('location/of/the/wav.wav')
11 plt.plot(plot_data, color='black')
12 plt.show()
```

*Kuva 4. Äänitiedoston lukeminen ja visualisointi.*

Tämän visualisoinnin jälkeen oli mahdollista aloittaa analysointi. Analysoinnin alkuvaiheissa oli tärkeää pystyä tunnistamaan alkavat herätteet ja sen vasteet (kuva 5). Tämän ansiosta saimme myös mahdolliset herätteen alkamisajat sekä herätteen keston (kuva 6). Tämän jälkeen pystyin opettelemaan herätteen sekä vasteen mallit, joita voitiin käyttää vertailussa. Sovelluksessa pystyttiin määrittelemään, kuinka pitkä heräte tai vaste on. Pystyttiin myös lajittelemaan iskujen välit. Tietyissä tapauksissa äänitiedoston sisältäessä useamman lyönnin, ne halettiin luokitella yhdeksi herätteeksi tai vasteeksi.



Kuva 5. Herätteen tunnistaminen.

```

Start time of the peak: 5.44 s, duration of peak: 0.21 seconds , start and end peak 1044357 1084882
Start time of the peak: 9.71 s, duration of peak: 0.19 seconds , start and end peak 1865075 1901357
Start time of the peak: 15.00 s, duration of peak: 0.01 seconds , start and end peak 2880162 2882244
3

```

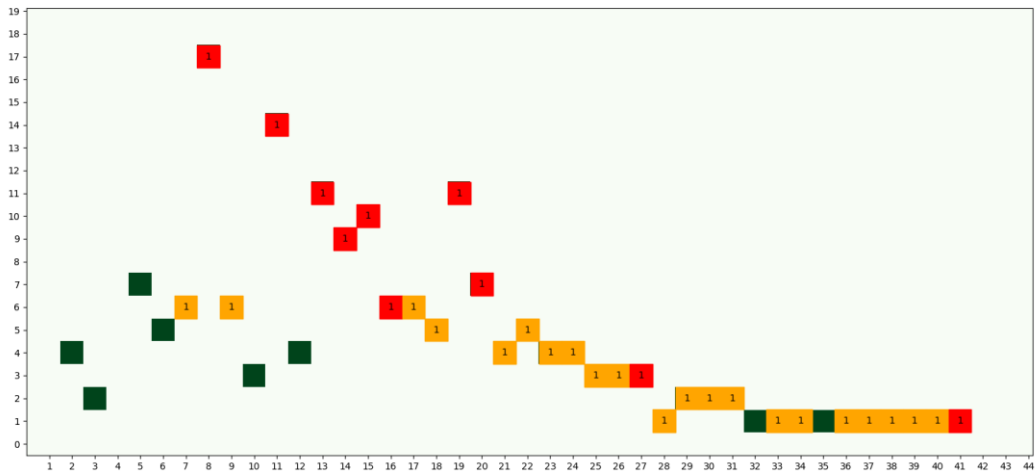
Kuva 6. Herätteeseen liittyvät tiedot

### 5.3 Analyysiohjelmiston testauksen alkuvaihe

Toteutuksen äärellä tehtiin useita erilaisia testauksia, millä voitiin hahmottaa sovelluksen tekemistä. Tässä alaluvussa käydään esimerkki visualisoinnista ja vertailusta.

Kuvassa 7 voidaan huomata herätteestä tehty matriisi, joka auttaa visualisoimaan äänessä tapahtuvia poikkeuksia. Kuvassa oleva y-akseli on äänessä tapahtuva amplitudi ja x-akseli on ääninäyte. Jotta ihminen voi visualisoida dataa helpommin, niin ääninäytteitä on summattu yhteen. Yksi x-akselin arvo on kolmekymmentäkahdeksan ääninäytettä. Kuvassa olevat vihreät neliöt kuvastavat missä alueella herätteen kuuluisi olla. Oranssit neliöt kuvastavat pientä poikkeusta kyseisellä alueella. Punaiset neliöt kuvastavat suurta poikkeusta äänessä. Ohjelmisto myös palauttaa kokonaistuloksen, kuinka vastaava kyseinen heräte (kuva 8), sekä vaste (kuva 9) on prosentteina.

Kuvan 7 visuaalinen analyysi mahdollistaa myös sen, että käyttäjä pystyy nopeasti tunnistamaan potentiaaliset ongelmakohdat koneen toiminnassa. Punaiset neliöt toimivat visuaalisena hälytyksenä, jotka voivat viitata esimerkiksi kuluneeseen komponenttiin tai tärähdyksen poikkeavuuteen. Tällainen selkeä esitys helpottaa myös ei-tekniisten käyttäjien tulkintaa.



Kuva 7. Mittaustesti.

```

● Event probability: 48.05231683777511%
Event probability: 69.30467578302641%
Event probability: 53.29038811075903%
File: ch1.wav - Probability: 56.88246024385352%

```

Kuva 8. Herätteen todennäköisyys.

```

Event probability: 89.7153099012163%
Event probability: 95.45973425095711%
Event probability: 81.59314197642134%
File: ch2.wav - Probability: 88.9227287095316%

```

Kuva 9. Vasteen todennäköisyys.

## 6 TESTAUS

Testausvaiheessa mikrofoni kytkettiin ulkoiseen äänikorttiin, mikä mahdollisti korkealaatuisen äänisignaalin tallennuksen. Tallennusympäristö pyrittiin pitämään mahdollisimman vakiona, jotta toistuvien lyöntien analysointi olisi luotettavaa. Analyysin avulla voitiin tarkastella esimerkiksi lyöntien vastaavuutta. Tämän perusteella saatiin arvokasta tietoa siitä, kuinka ohjelmisto tunnistaa toistuvia ja poikkeavia ääni-ilmiöitä.

Äänisignaali analysoitiin Pythonilla käyttäen kirjastoja kuten Soundfile, NumPy. Tarkoituksena oli tunnistaa lyöntien vastaavuus sekä poikkeavat ääni-ilmiöt. Lyöntien analyysi perustui huippujen havaitsemiseen. Tämän avulla voitiin tunnistaa, eroavatko jotkut lyönnit merkittävästi muista, mikä voi olla merkki alkavasta koneellisesta viasta.

Kun ohjelmisto oli suorittanut äänidatan analyysin, tulokset välitettiin MQTT-protokollaa hyödyntäen. Tämä mahdollisti analyysin siirtämisen kevyesti ja tehokkaasti esimerkiksi käyttöliittymälle. Viestit lähetettiin tiettyyn MQTT-topiciin, jonka tilaaja pystyi vastaanottamaan reaaliaikaisesti. Jokainen viesti sisälsi rakenteellisesti analyysin keskeiset tunnusluvut (kuva 11).



```
monitor/last_result ("timestamp": "2025-04-22 05:53:17.727790+00:00", "channel_names": ("CH5": "tacho1", "CH6": "A4", "CH7": "A5", "CH8": "A0"), "key_figures": {"peak": [0.99, 0.01, 0.01, 0.05], "rms": [0.22, 0.0, 0.0, ...] 08:53:04.811 ^
JSON v 00s: 0 Q
46      "impact_analysis": [
47        0,
48        [
49          {
50            "overall": 93.7241765848966,
51            "start": 99.90,
52            "mid": 90.34,
53            "end": 82.06
54          }
55        ],
56        0,
57        0
```

Kuva 11. Vastaanotettu MQTT-viesti, missä näkyy lyönnin vastaavuudet prosentteina.

## 7 YHTEENVETO

Analyysiohjelmiston kehitys onnistui kokonaisuutena hyvin. Työn tarkoituksena oli kehittää analyysiohjelmisto, joka pystyy tunnistamaan iskusta syntyvät herätteet sekä vasteet. Tähän vaikutti moni asia kuten mittauslaitteisto, ohjelmisto sekä testivälineet. Ohjelmisto tarjosi voimallituksen osaprosessiin hyvän analyysiohjelmiston, jonka avulla voidaan tarkkailla laitteiston kuntoa.

Suunnittelu ohjelmiston tekemisen kannalta oli erittäin olennaista, jotta voitiin luoda toimiva ja luotettava ohjelmisto. Suunnittelussa oli tärkeää luoda oikeanlainen testialusta, jolla voitiin simuloida äänessä syntyvät herätteet ja sen vasteet.

Ohjelmiston suunnittelu sekä toteutus onnistui hyvin. Testialusta olisi voitu luoda paremmaksi, jotta voidaan simuloida lyöntejä paremmin. Ohjelmiston tekeminen toi erittäin paljon hyötyjä kuten: herätteen sekä vasteen tunnistamisen, mallin opetuksen sekä mahdollisen mallin konfiguroimisen.

Käyttöliittymän kannalta Nome Oy on toteuttanut etukäteen käyttöliittymän, joka voidaan soveltaa tähän analyysiohjelmistoon. Tämä mahdollistaa asiakkaalle laitteiston kunnon seuraamisen.

Yksi projektin merkittävimmistä haasteista oli kehittää testialusta, joka mahdollistaisi toistettavan ja yhdenmukaisen iskun simuloinnin joka kerta. Myös ohjelmiston suunnittelu olisi ollut yksinkertaisempaa, jos käytettävissä olisi ollut aitoa mitausdataa.

Jatkokehityksessä ohjelmistoa voisi laajentaa tunnistamaan erilaisia ääniherätteitä tai hyödyntämään koneoppimista tarkemman analyysin saavuttamiseksi, sekä vertaamaan paremmin äänestä syntyvää herätettä sekä vastetta toisiinsa.

## LÄHTEET

Amazon. 2025. What is Python? Luettavissa: <https://aws.amazon.com/what-is/python/> Luettu: 13.02.2025

Anderson, J. 2025. An Intro to Threading in Python Luettavissa: <https://real-python.com/intro-to-python-threading/#what-is-a-thread> Luettu: 13.02.2025

Dency, F & Swetha, A. 2012. IoT Based End to End Solutions for Industrial Surveillance using Raspberry Pi. Luettavissa: <https://www.ijitee.org/wp-content/uploads/papers/v9i3/B6337129219.pdf> Luettu 23.04.2025

Google Cloud. s.a. What is Supervised Learning? Luettavissa: <https://cloud.google.com/discover/what-is-supervised-learning> Luettu 19.02.2025

Google Cloud. s.a. What is unsupervised learning? Luettavissa: <https://cloud.google.com/discover/what-is-unsupervised-learning> Luettu: 19.02.2025

Hentinen, M., Hynnä, P., Lahti, T., Nevala, K., Vähänikkilä, A., Järviluoma, M., 30.04.2002. Värähtelyn ja melun vaimennuskeinot kulkuvälineissä ja liikkuvissa työkoneissa. VTT Tiedotteita 2160. VTT. Espoo. Luettavissa: <https://publications.vtt.fi/pdf/tiedotteet/2002/T2160.pdf> Luettu: 12.02.2025.

HiveMQ. 2015. Introducing the MQTT Protocol – MQTT Essentials: Part 1. Luettavissa: <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>. Luettu: 10.04.2025

Kagen, A. 16.02.2021. What Is an Audio Interface and Do You Need It? Luettavissa: <https://www.mi.edu/in-the-know/do-you-need-audio-interface/> Luettu: 13.02.2025

Lamula, L. & Saarinen, K. 29.10.2019. Värähtelyn eteneminen rakenteissa ja äänen säteily. Luettavissa: [https://www.akustinenseura.fi/wp-content/uploads/2019/10/akustiikkapaivat\\_2019\\_s229.pdf](https://www.akustinenseura.fi/wp-content/uploads/2019/10/akustiikkapaivat_2019_s229.pdf) Luettu 12.02.2025

Lenovo. 2025. What is a microphone? Luettavissa: <https://www.lenovo.com/us/en/glossary/microphone/> Luettu 13.02.2025

Library Carpentry. 11.05.2024. Python Intro for Libraries Luettavissa: <https://librarycarpentry.github.io/lc-python-intro/libraries.html> Luettu: 13.02.2025

Malik U. 16.7.2019. Advantages and Disadvantages of the Python Programming Language. Luettavissa: <https://learnpython.com/blog/python-programming-advantages-disadvantages/>. Luettu 15.04.2025

MIP. 2025. Kunnonvalvonta. Luettavissa: <https://www.mip.fi/fi/tuotteet/tarina-ja-varahtely/kunnonvalvonta> Luettu: 28.02.2025

Paessler. 2025. IT Explained: MQTT Luettavissa: <https://www.paessler.com/it-explained/mqtt> Luettu: 12.02.2025

Pandey, P. 10.2.2025. Python and Its Implications. Luettavissa: <https://ijsrem.com/download/python-and-its-implications/> Luettu: 23.04.2025

Python Software Foundation. 13.02.2025. The Python Tutorial Luettavissa: <https://docs.python.org/3/tutorial/index.html> Luettu: 13.02.2025

Stewart, I. 04.12.2024. Digital audio basics: audio sample rate and bit depth Luettavissa: <https://www.izotope.com/en/learn/digital-audio-basics-sample-rate-and-bit-depth.html> Luettu 13.02.2025

Verma, A. 2014. Review on Condition Monitoring Techniques Oil Analysis, Thermography and Vibration Analysis. Luettavissa: <https://www.semanticscholar.org/paper/Review-on-Condition-Monitoring-Techniques-Oil-and-Verma-Srivastava/a7c433236942f020c6a0709a29dd227de11d7c40#extracted> Luettu: 21.04.2025

Wikipedia. 03.02.2024. Teollisuus. Luettavissa: <https://fi.wikipedia.org/wiki/Teollisuus> Luettu: 28.02.2025

Wikipedia. 2011. Raspberry Pi. Luettavissa: [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi) Luettu 12.02.2025

Wikipedia. 22.01.2025. Audio interface Luettavissa: [https://en.wikipedia.org/wiki/Audio\\_interface](https://en.wikipedia.org/wiki/Audio_interface) Luettu: 13.02.2025