

Prediktivt vedlikehold med Lattice:

I industrien har prediktivt vedlikehold, med hjelp av AI/ML, blitt interessant for å sikre effektiv drift. Ved å benytte AI/ML-metoder kan man forutsi vedlikeholdsbehov i sanntid, noe som bidrar til å forhindre unødvendig nedetid og redusere kostnader.

Problemstilling: Detektere tidlige feil i kulelagre ved hjelp av vibrasjonsmåling. Implementasjon på Lattice FPGA ved hjelp av [Lattice SenseAI Studio](#).

Aktuelle datasett:

- [HUST bearing: a practical dataset for ball bearing fault diagnosis](#)
- [NASA Bearing Dataset](#)
- [Case Western Reserve University Bearing Data Center Website](#)

Oppgave:

1. Trene en ML-modell (Tensorflow/PyTorch) som kan gjenkjenne kulelagre med feil fra eksisterende datasett
2. Konvertere ML-modellen til et format som kan kjøres på Lattice FPGA (hvilke kompromisser/optimaliseringer må gjøres i denne prosessen? Hvordan påvirker samplingsfrekvens og ADC-oppløsning ytelsen? Hvilke begrensninger har FPGA i forhold til en floating-point basert datamaskin?)
3. Sammenligne ytelse på original modell og Lattice FPGA-modell
4. Implementer ML-modell på et Lattice FPGA utviklingskort med ADC-kort (f.eks. type PMOD) for innsamling av vibrasjonssignal. Vurder ulike arkitekturer for implementering: Koble ADC data direkte til ML-modell i FPGA-logikk vs. la en CPU hente data fra ADC og så videresende data til ML-modell i FPGA-logikk.
5. Sett opp en praktisk test hvor en aksling med kulelager roteres av en motor. Vurder sensortype og plassering av sensor. Lag ulike defekter på kulelageret. Test og evaluer ML-modellen på sanntidsdata.
6. Samle inn nye data med det praktiske oppsettet, og optimaliser modellen ytterligere ved å trene den med ekstra data. Test og evaluer den oppdaterte ML-modellen på sanntidsdata.

Prediktivt vedlikehold med AMD/Xilinx:

I industrien har prediktivt vedlikehold, med hjelp av AI/ML, blitt interessant for å sikre effektiv drift. Ved å benytte AI/ML-metoder kan man forutsi vedlikeholdsbehov i sanntid, noe som bidrar til å forhindre unødvendig nedetid og redusere kostnader.

Problemstilling: Detektere tidlige feil i kulelagre ved hjelp av vibrasjonsmåling. Implementasjon på Xilinx FPGA ved hjelp av [VitisAI](#).

Aktuelle datasett:

- [HUST bearing: a practical dataset for ball bearing fault diagnosis](#)
- [NASA Bearing Dataset](#)
- [Case Western Reserve University Bearing Data Center Website](#)

Oppgave:

1. Trene en ML-modell (Tensorflow/PyTorch) som kan gjenkjenne kulelagre med feil fra eksisterende datasett
2. Konvertere ML-modellen til et format som kan kjøres på Xilinx FPGA (hvilke kompromisser/optimaliseringer må gjøres i denne prosessen? Hvordan påvirker samplingsfrekvens og ADC-oppløsning ytelsen? Hvilke begrensninger har FPGA i forhold til en floating-point basert datamaskin?)
3. Sammenligne ytelse på original modell og Xilinx FPGA-modell

4. Implementer ML-modell på et [Xilinx FPGA utviklingskort](#) med ADC-kort (f.eks. type PMOD) for innsamling av vibrasjonssignal. Vurder ulike arkitekturer for implementering: Koble ADC data direkte til ML-modell i FPGA-logikk vs. la en CPU hente data fra ADC og så videresende data til ML-modell i FPGA-logikk.
5. Sett opp en praktisk test hvor en aksling med kulelager roteres av en motor. Vurder sensortype og plassering av sensor. Lag ulike defekter på kulelageret. Test og evaluer ML-modellen på sanntidsdata.
6. Samle inn nye data med det praktiske oppsettet, og optimaliser modellen ytterligere ved å trene den med ekstra data. Test og evaluer den oppdaterte ML-modellen på sanntidsdata.