

Sistem za prediktivno održavanje industrijskih mašina

Članovi tima

- Stevan Sapun, SV7/2021
-

Motivacija

U industrijskim postrojenjima neplanirani kvarovi mašina uzrokuju velike finansijske gubitke. Ovi gubici uključuju direktne troškove popravki i rezervnih delova, kao i indirektne gubitke u proizvodnji i ugrožene isporuke kupcima. Tradicionalni modeli održavanja zasnovani na fiksnim intervalima često su neefikasni jer tretiraju sve mašine isto, bez obzira na njihovo stvarno stanje. Rezultat su ili prečeste, nepotrebne intervencije ili zakašnjele reakcije koje vode do havarija.

Prediktivno održavanje, posebno kada se oslanja na **sisteme bazirane na znanju (SBNZ)**, omogućava: - prepoznavanje obrazaca koji ukazuju na potencijalni kvar, - pružanje transparentnog objašnjenja odluka (za razliku od nekih „black-box“ ML metoda), - i definisanje pravila koje eksperti lako mogu proširivati.

Na ovaj način postiže se smanjenje broja zastoja, bolja iskorišćenost opreme i veća bezbednost postrojenja.

Pregled problema

Sistem se fokusira na rotacione mašine (motore, pumpe, ventilatore), koje su najčešći izvor kvarova u industrijskim postrojenjima. Tipični problemi su: - **Neuravnoteženost rotora** (povećane vibracije), - **Habanje ležajeva** (karakteristični vibracioni spektri i pregrevanje), - **Problemi sa hlađenjem** (povećanje temperature), - **Električna preopterećenja** (porast struje, overload trip događaji).

Naš sistem obezbeđuje sveobuhvatnu detekciju kroz integraciju različitih izvora podataka i pravila. Fokus nije samo na pojedinačnim senzorima, već i na kombinaciji događaja i njihovom redosledu (Complex Event Processing). Time se postiže prednost u odnosu na klasične metode, jer korisnici dobijaju i **predikciju i objašnjenje odluke**.

Metodologija rada

Korisnici sistema

- **Operateri održavanja** - primaju alarmne notifikacije i konkretne preporuke za akciju.
- **Inženjeri dijagnostike** - imaju uvid u pravila i mogu da podešavaju pragove ili dodaju nova pravila.
- **Menadžeri** - primaju agregirane izveštaje sa KPI pokazateljima, poput broja sprečenih zastoja ili ušteda u troškovima.

Ulazi u sistem

Ulazi se dele na tri grupe:

- **Telemetrijski podaci:** vibracije (merene akcelerometrima, RMS vrednosti i spektri), temperatura ležaja (termopar ili IR senzori), struja motora (strujni transformatori), broj obrtaja (tachometar).
- **Događaji:** StartMachine, StopMachine, OverloadTrip, MaintenanceAction (registrovani putem PLC/SCADA sistema).
- **Konteksti rada:** Normal, HighLoad, PostMaintenance, Idle.

Izlazi iz sistema

Izlazi se mogu podeliti na:

- **Statusi mašina:** Normalna, Sumnjiva, Rizična, Kritična.
 - **Notifikacije:** alarmni panel u SCADA sistemu, email/SMS upozorenja.
 - **Preporuke za održavanje:** „Proveri balans“, „Podmaži ležaj“, „Zameni hlađenje“.
 - **Izveštaji:** detaljni grafički prikazi trendova, lista „top rizičnih mašina“, KPI analize i prediktivni scenariji.
-

Baza znanja

Baza znanja je sa kompletnim setom pravila (20+) podeljenih u kategorije.

1. Pravila za vibracije

```
when vibration > 7.1 mm/s for 5 minutes and K = Normal  
then setStatus(Sumnjiva) and recommend("Proveri  
balans")
```

```
when vibration > 10 mm/s for 5 minutes  
then setStatus(Rizična) and recommend("Proveri ležaj i balans")
```

```
when vibration increase > 20% within 1 min  
then setStatus(Sumnjiva) and recommend("Nagla promena vibracija –  
proveri rotor")
```

```
when repeated vibration anomalies > 3 times in  
24h then setStatus(Rizična) and
```

2. Pravila za temperaturu

```
when temperature > 85°C and K = Normal  
then setStatus(Rizična) and recommend("Proveri hlađenje")
```

```
when temperature > 90°C and K = HighLoad  
then setStatus(Kritična) and shutdownMachine()
```

```
when temperature > 80°C for 30 minutes  
then setStatus(Rizična) and recommend("Moguće habanje ležaja")
```

3. Pravila za struju i opterećenje

```
when current > 120% rated and overloadOccurred()
then setStatus(Kritična) and shutdownMachine()

when current > 100% rated for 15 minutes
then setStatus(Rizična) and recommend("Smanji opterećenje")

when overloadTripCount >= 3 in 24h
then setStatus(Rizična) and recommend("Proveri električne komponente")
```

4. Pravila za kombinovane anomalije

```
when vibrationAnomaly() and temperatureAnomaly() within 10 minutes
then setStatus(Rizična) and recommend("Moguće oštećenje ležaja")

when vibrationAnomaly() and overloadOccurred()
then setStatus(Kritična) and dispatch(AlertMaintenance)

when vibration increase + temperature increase + current anomaly in
1h then setStatus(Kritična) and shutdownMachine()
```

5. Pravila za kontekste

```
when K = PostMaintenance and temperature > 80°C
then informUser("Temperatura raste posle servisa") and schedule(Inspection)

when K = HighLoad and vibration > threshold[K].vibration
then setStatus(Sumnjiva) and recommend("Proveri balans pri opterećenju")

when K = Idle and temperature > 60°C
then setStatus(Rizična) and recommend("Ventilacija ne radi ispravno")
```

Pravila na nivou pogona (accumulate)

```
when accumulate(status = Rizična across >20% of machines)
then setPlantStatus(Kritičan) and notify(Management)

when accumulate(overloadTripCount > 5 in last 7 days)
then schedule(PlantWideInspection)
```

Primeri rezonovanja i objašnjenja

Forward chaining (3 nivoa + accumulate)

1. Lokalna detekcija anomalije (npr. vibracije > prag).
2. Agregacija anomalija u vremenskom prozoru (≥ 2 anomalije u 10 min).
3. Eskalacija na nivou postrojenja ($\geq 20\%$ rizičnih mašina).

Complex Event Processing (CEP)

- Pravilo „vibracije + temperatura“ omogućava otkrivanje habanja ležajeva.
- Pravilo „post-servis + temperatura“ ukazuje na potencijalnu lošu popravku.
- Pravilo „naglo povećanje vibracija + overload“ ukazuje na mehaničko-električni problem.

Backward chaining (dizajn)

Cilj (goal-driven) pristup:

- **Top-level ciljevi (GOALS):**
 - AtRisk(M)
 - BearingFault(M)
 - Imbalance(M)
 - Overheat(M)
 - ElectricalOverload(M)
- **Potciljevi (SUBGOALS):**
 - TwoAnomaliesIn(M, Δt)
 - VibrationHigh(M)
 - TemperatureHigh(M)
 - CurrentHigh(M)
 - RecentOverload(M, Δt)
 - Context(M, K)
 - Windowed(M, predicate, Δt)
- **Askable činjenice (od senzora/SCADA):**
 - vibration_rms(M)
 - temperature(M)
 - current(M)
 - overload_trip(M, t)
 - context_active(M, K, t)
- **Derived (izvedene) činjenice:**
 - logičke kompozicije koje engine zaključuje iz askable + vremenskih prozora

Struktura pravila (Horn klauzule, bez side-effect akcija):

Pravila za backward moraju biti **deklarativna** (isti skup kao za forward, ali bez setStatus/notify u THEN delu). Akcije se izvode **nakon** što je cilj dokazan.

Primeri pravila (za backward dokazivanje):

```
% Top-Level cilj: mašina je u riziku ako su ispunjena ≥2 različita potcilja u Δt
AtRisk(M) :- TwoAnomaliesIn(M, 10min).

% BearingFault kad su istovremeno visoke vibracije i temperatura u prozoru
BearingFault(M) :- VibrationHigh(M), TemperatureHigh(M), Windowed(M, both, 10min).

% Imbalance ako je RMS vibracija iznad ISO praga ili naglo poraste
Imbalance(M) :- VibrationHigh(M).
Imbalance(M) :- VibrationJump(M, 20_percent, 1min).

% Overheat ako je T preko praga (uzimajući kontekst)
Overheat(M) :- TemperatureHigh(M), Context(M, Normal).
CriticalOverheat(M) :- TemperatureVeryHigh(M), Context(M, HighLoad).

% Electrical overload
ElectricalOverload(M) :- CurrentHigh(M), RecentOverload(M, 24h).

% Potciljevi definicije
TwoAnomaliesIn(M, Δ) :- VibrationHigh(M), TemperatureHigh(M), Windowed(M, both, Δ).
TwoAnomaliesIn(M, Δ) :- VibrationHigh(M), CurrentHigh(M), Windowed(M, both, Δ).

VibrationHigh(M) :- vibration_rms(M) > 7.1.
VibrationVeryHigh(M) :- vibration_rms(M) > 10.
VibrationJump(M, Pct, Δ) :- percent_increase(vibration_rms, M, Δ) >= Pct.

TemperatureHigh(M) :- temperature(M) > 85.
TemperatureVeryHigh(M) :- temperature(M) > 90.

CurrentHigh(M) :- current(M) > 1.0*rated(M).
RecentOverload(M, Δ) :- overload_trip(M, t), now(T0), T0 - t =< Δ.

Context(M, K) :- context_active(M, K, _).
Windowed(M, both, Δ) :- within_window(M, Δ).
```

Kontrola pretrage (strategy):

- **Goal ordering:** prvo proveriti jeftine potciljeve (Context, RecentOverload), zatim merenja, pa aggregate (TwoAnomaliesIn)
- **Depth- first sa ograničenjem + memoizacija** (tabling) za ciljeve po mašini da se izbegnu ciklusi.
- **Time- aware unifikacija:** svi potciljevi se proveravaju u istom vremenskom prozoru Δ (podrazumevano 10 min, konfigurisano)
- **Conflict resolution:** ako se dokaže više hipoteza (BearingFault, Imbalance), sistem vraća najspecifičniju (subsumption: BearingFault > Imbalance).

I/O povezivanje (ask the world):

ask vibration_rms(M) čita poslednju agregiranu RMS vrednost; ako nije dostupna, engine inicira **query plan** (npr. zahteva poslednjih 5 min podataka). - ask overload_trip(M, t) proverava događaje u logu (SCADA) i filtrira u odnosu na Δ.

Kada i kako izvršiti akcije:

Nakon što je cilj **dokazan** (npr. AtRisk(M) ili CriticalOverheat(M)), pokreće se **policy sloj** koji mapira ciljeve na akcije: AtRisk → setStatus(Rizična), CriticalOverheat → shutdownMachine().

Primer upita i dokaza (proof tree):

```
AtRisk(M12)?  
└─ TwoAnomaliesIn(M12,10min)?  
   └─ VibrationHigh(M12)? → yes (RMS=6.5>6.0 thr)  
      └─ TemperatureHigh(M12)? → yes (T=82>80 thr)  
         └─ Windowed(M12,both,10min)? → yes (t_vib,t_temp ∈ [T0-10,T0])
```

Zaključak: AtRisk(M12) je **dokazan**. Policy: setStatus=Rizična, recommend="Proveri balans i ležaj".

Interakcija sa korisnikom (why/what):

WHY AtRisk(M12)? → prikaži dokazno stablo i aktivirana pravila. - WHAT IF analiza: privremeno promeni Context(M,HighLoad) u Normal i re-evaluiraj cilj (sensitivity check).

Primeri upita (queries)

Backward (goal- oriented) upiti:

- AtRisk(MX) . – da li je mašina MX u riziku?
- BearingFault(MX) . – da li postoje indikatori habanja ležaja?
- Imbalance(MX) . – da li je neuravnoteženost uzrok anomalija?
- CriticalOverheat(MX) . – da li treba hitna reakcija gašenja?
- ElectricalOverload(MX) . – da li postoji električno preopterećenje?

What/Why upiti:

- WHY AtRisk(MX)? - objasni koje činjenice i pravila su korišćeni.
- WHAT IF Context(MX,Normal)? - analiza osetljivosti na promenu konteksta.

Primer upotrebe (trace)

Scenario: Motor M12 radi u režimu visokog opterećenja.

1. Motor ulazi u HighLoad kontekst.
2. Vibracije = 6.5 mm/s (pravilo V1 → anomalija).
3. Temperatura = 82°C (pravilo T2 → anomalija).
4. CEP spaja događaje (2 anomalije u 10 min → pravilo C1 → status „Sumnjiva“).
5. Nakon 30 min bez poboljšanja → pravilo eskalacije → status „Rizična“.
6. Accumulate: 25% mašina rizične → pravilo A1 → pogon = „Kritičan“.
7. Menadžer dobija email: „Pogon u kritičnom stanju. Najčešći uzrok: habanje ležajeva.“
8. Operater dobija SCADA alarm: „Proveri balans i ležajeve na M12“.

Trace detaljno ilustruje kako sistem od senzora preko pravila i rezonovanja dolazi do različitih korisnika.

Idejni prikaz toka delanja sistema

[Senzori] → [Prikupljanje podataka] → [Pravila (Forward/CEP/Backward)] → [Statusi i akcije] → [Korisnici]

Izveštaji

Izveštaji nisu samo lista, već strukturisani dokumenti koji korisnicima pomažu da donesu odluke:

- **Trend izveštaj:** grafikon RMS vibracija sa obeleženim anomalijama i servisnim akcijama.
- **Izveštaj o rizičnim mašinama:** lista mašina sa statusima, preporučenim akcijama i procenjenim rokom do kvara (ETA).
- **Agregatni izveštaj:** broj sprečenih kvarova, sati zastoja koji su izbegnuti i procenjena finansijska ušteda.
- **Prediktivni izveštaj:** simulacija naredne nedelje sa verovatnoćom kvara za svaku mašinu.
- **Root-cause analiza:** detaljna lista pravila koja su se aktivirala i dovela do zaključka.

Zaključak

Kroz ovaj projekat imao sam priliku da istražim kako sistemi bazirani na znanju mogu da se primene u industrijskom okruženju i na realan način unaprede procese održavanja. Najveća vrednost leži u tome što ovakav pristup omogućava da se ekspertsko iskustvo formalizuje kroz pravila i šablone, i da postane dostupno svima u organizaciji. Ujedno, svaki zaključak koji sistem donosi je transparentan i objašnjiv, što gradi poverenje kod korisnika i olakšava donošenje odluka.

Iako sistem nije samoučeći i zavisi od kvaliteta pravila koja definišu stručnjaci, upravo ta osobina ga čini pouzdanim i kontrolisanim u kritičnim industrijskim uslovima. Dalji razvoj može uključiti kombinaciju sa mašinskim učenjem radi automatske detekcije novih obrazaca, ali osnova u vidu jasnih ekspertskih pravila ostaje ključna.