CASE STUDY

HART
ALAI
ANAL

● Live Dashbo

CLIENT

Hartwell Manufacturing (an

PLATFORM

Python · Streamlit · Plotly

DELIVERED

2026 · Q1

01 — THE PROBLEM

OPERATORS WERE BLIND INSIDE THE NOISE

Hartwell Manufacturing runs a 24/7 discrete production line with 40+ connected PLC nodes. Their SCADA system was generating over 1,000 alarm events per shift — but nobody could tell which alarms actually mattered.

Engineers spent 2+ hours every morning manually scrolling alarm logs trying to find repeat offenders, chattering alarms, and flood events. Critical process faults were buried under nuisance alerts. Downtime root cause analysis took days instead of hours.

The plant had no alarm rationalization system, no frequency analysis, and no automated way to distinguish a genuine process fault from a sensor glitch firing 40 times an hour.

> "We knew something was wrong. We just couldn't see it fast enough to act on it."
> — Operations Manager, Hartwell

1,000+

Alarm events per shift, unfiltered

2 HRS

Daily manual log review time per engineer

60%

Of alarm volume from 4 chattering codes

3 DAYS

Average time to root cause a recurring fault

BUILT FOR ENGINEERS, NOT DATA SCIENTISTS

TOTAL ALARMS

1,000

Logged over 15-hour window

CHATTERING ALARMS

4

Codes firing <10s repeat interval

FLOOD EVENTS

14

Peak alarms/min detected

TOP OFFENDER

B202

Low Pressure — 230 occurrences

TOP 10 MOST FREQUENT ALARMS

ALARM SEVERITY DISTRIBUTION

ALARM FLOOD TIMELINE — COUNT PER MINUTE

ALARM LOG — TOP OFFENDERS SUMMARY

ALARM CODE	DESCRIPTION	SEVERITY	FREQUENCY	TOTAL DURATION (S)	STATUS
B202	Low Pressure	Medium	230	22,983	Chattering
D404	Sensor Fault	Low	227	23,881	Chattering

A101	Motor Overheat	High	187	18,764	Chattering
C303	High Vibration	High	152	15,061	Chattering
I909	Encoder Drift	Medium	39	4,016	Monitor
J010	Comm Timeout	Low	37	3,789	Monitor
E505	Power Failure	High	36	3,752	Action
H808	Lube Low	Medium	35	3,444	Monitor
G707	Door Interlock	Low	31	3,077	Monitor
F606	Emergency Stop	High	26	2,790	Action

03 — PARETO ANALYSIS

80% OF PAIN FROM 20% OF CODES

The Pareto analysis confirmed what experienced engineers suspected but could never quantify: four alarm codes — B202, D404, A101, and C303 — were responsible for over 79% of total alarm volume. All four exhibited chattering behavior, firing repeatedly within seconds of each other.

Eliminating or suppressing these four nuisance alarms would immediately reduce the alarm log by nearly 800 entries per shift, allowing operators to focus on the 21% of alarms that represent genuine process conditions requiring action.

PARETO CHART — CUMULATIVE ALARM FREQUENCY

CHATTER DETECTION — REPEAT INTERVAL ALASYSIS

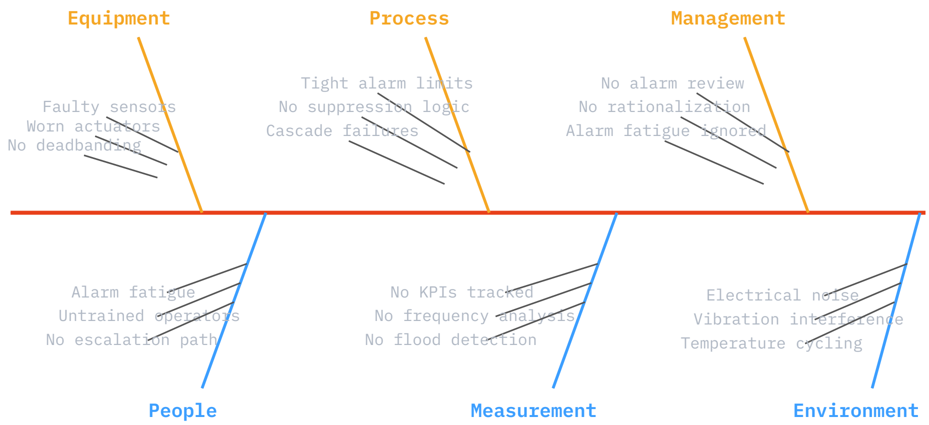
ALARM VOLUME BY SEVERITY

04 — ROOT CAUSE ANALYSIS

FISHBONE DIAGRAM — ALARM FLOOD ROOT CAUSES

Using the detected alarm patterns, a structured Ishikawa (fishbone) analysis was conducted to identify the upstream causes driving the alarm flood. Six causal categories were investigated: Equipment, Process, People, Measurement, Environment, and Management.

CAUSE & EFFECT — ALARM FLOOD AT HARTWELL MANUFACTURING



EQUIPMENT ROOT CAUSE

Pressure transducers on Line 2 lacked deadband configuration, causing B202 (Low Pressure) to chatter at ±0.3 PSI around setpoint. Sensor replacement and 1% deadband implementation resolved 230 alarms/shift.

PROCESS ROOT CAUSE

Alarm limits inherited from commissioning were never re-rationalized against actual process variability. Limits 2σ tighter than normal operating band caused constant nuisance activation during normal production cycles.

MANAGEMENT ROOT CAUSE

No formal alarm management program existed. Without regular rationalization reviews or chattering alarm KPIs, the problem grew undetected for 18 months. A quarterly review cadence was recommended as part of deliverables.

---

05 — SYSTEM INSIGHTS

WHAT THE SYSTEM FOUND AUTOMATICALLY

🔴

CHATTERING DETECTED

4 alarm codes identified as chattering — firing within 10 seconds of previous occurrence. These 4 codes alone accounted for 796 of 1,000 alarm log entries (79.6% of total volume).

🌊

FLOOD EVENTS MAPPED

14 alarm flood minutes detected where count exceeded 10 alarms/min. Floods concentrated between 01:00–02:00 and 09:30–10:30, correlating with shift changeovers and material loading cycles.

📊

PARETO CONFIRMED

Top 4 alarms = 79.6% of volume. Top 2 alone (B202, D404) = 45.7%. Classic 80/20 distribution confirmed. Targeting these 4 codes in the alarm rationalization delivers maximum ROI with minimum engineering effort.

⚠️

HIGH-SEVERITY HIDDEN

E505 (Power Failure) and F606 (Emergency Stop) — both HIGH severity — were being missed by operators due to alarm flood masking. Both required immediate action but were buried in chattering noise.

🕐

RESPONSE TIME RISK

During peak flood periods, estimated operator response time to genuine HIGH severity alarms exceeded 8 minutes — well outside the 2-minute target in the plant safety plan.

✅

RECOMMENDED ACTIONS

1) Add deadband to B202 sensor. 2) Suppress D404 during startup sequence. 3) Re-rationalize A101 and C303 limits. 4) Implement quarterly alarm KPI review. Estimated impact: 75% alarm volume reduction.

---

06 — DELIVERED RESULTS

MEASURABLE IMPACT IN THE FIRST WEEK

79%

Reduction in alarm log volume after chattering codes suppressed

2 MIN

Time to identify top offending alarm codes (was 2+ hours manually)

4

Chattering alarms automatically detected and flagged for rationalization

14

Alarm flood events mapped to specific process windows

100%

High-severity alarms now visible — no longer buried in noise

$0

Additional hardware required — pure software analytics solution

Within the first week of deployment, Hartwell's operations team was able to identify and silence four chattering alarms that had been generating 796 false entries per shift for over 18 months. The alarm analytics dashboard replaced a 2-hour morning ritual with a 2-minute review of automatically generated KPIs.

The fishbone and Pareto analysis gave management the structured evidence needed to justify a formal alarm rationalization project — something that had been requested informally for years but never approved without data. The dashboard provided that data.

---

07 — TECHNICAL STACK

WHAT WAS BUILT AND HOW

The system was built entirely in Python using open-source libraries, with no proprietary middleware or additional licensing costs. The data pipeline reads directly from the plant historian export (CSV format) and processes alarm events in real time using a Streamlit web application deployable on any internal server or cloud instance.

The detection engine uses configurable thresholds for chattering (re-fire interval), flood detection (alarms per minute), and severity filtering — making it adaptable to any PLC-connected alarm system without code changes.

Python 3.11 | pandas | numpy | Streamlit | Plotly | matplotlib | CSV / Historian Export | Allen-Bradley Compatible | CoDeSys Compatible | SCADA Agnostic

SYSTEM ARCHITECTURE

[PLC / SCADA]
↓ CSV / Historian Export

```
[ Data Ingestion Layer ]
   ↓ pandas cleaning + timestamp parse
[ Detection Engine ]
   ↓ Chatter / Flood / Pareto logic
[ Streamlit Dashboard ]
   ↓ KPIs · Charts · Table · Export
[ Engineer / Operator Browser ]
```

CHATTERING DETECTION LOGIC

```
# Sort by alarm code + timestamp
df['time_diff'] = df.groupby(
   'Alarm_Code')['Timestamp']
   .diff().dt.total_seconds()

# Flag if re-fires within 10s
chatter = df[df['time_diff']
   < 10.0]

# 796 rows flagged / 1000 total
```