Data Stream Mining with Limited Validation Opportunity: Towards Instrument Failure Prediction

Atkinson, K.[1], Coenen, F.[1], Goddard, P.[2], Payne, T[1] and Riley, L[1,2].

1 UNIVERSITY OF LIVERPOOL

Knowledge Transfer Partnerships

CSols 2

Motivation

- Analytical instruments routinely quantify many metals and some non-metals at high speeds in liquids.

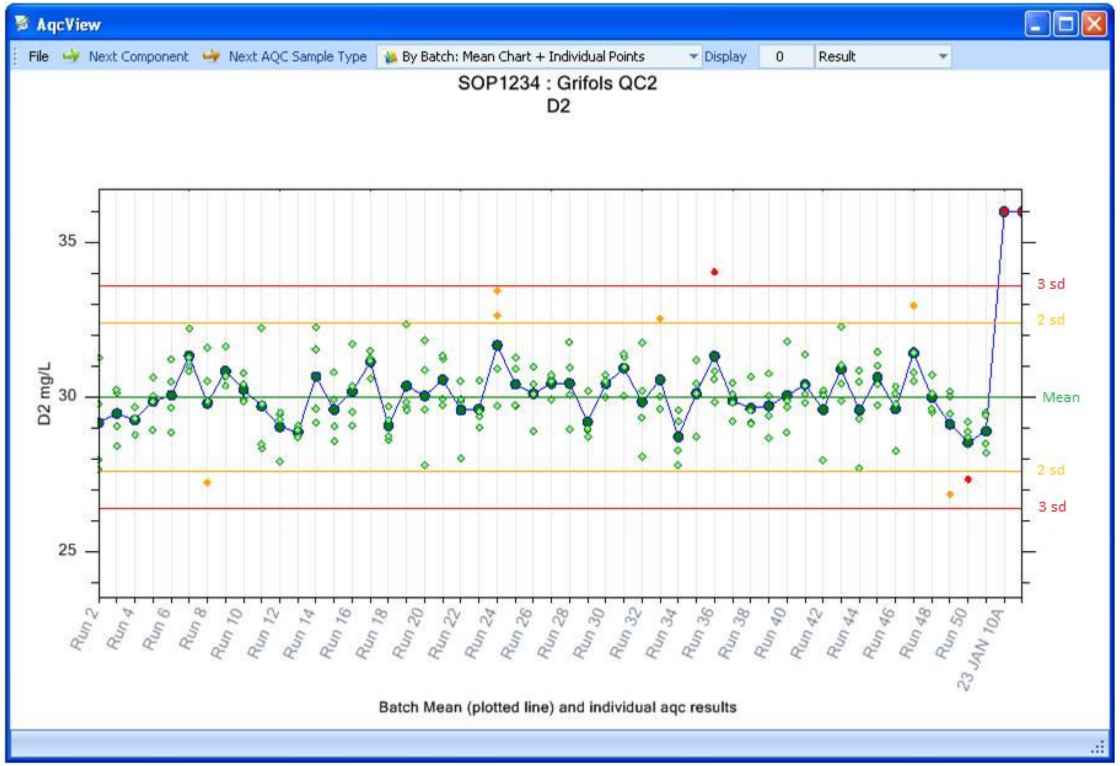
But…. How does an analyst know that these readings are correct?

Methods to Detect Instrument Errors

- Declining Instrument Sensitivity
- Inconsistent results for Analytical Standards
- System Suitability

Inconsistent Results for Analytical Standards

- The most effective way of monitoring an instrument's performance is to regularly use control samples with independently verified results



Initial Summary

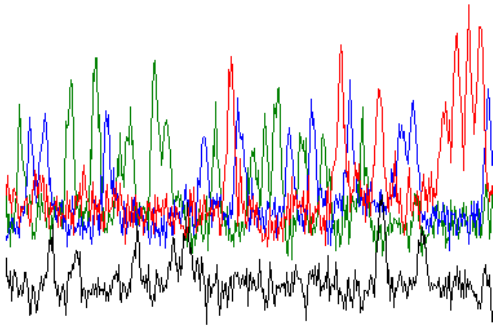
- The paper describes a model that can predict instrument failure so that some mitigation can be invoked so as to prevent failure.

- Specifically: This paper presents a probabilistic time-series analysis technique applied to data stream subsequences to predict instrument failure, where significant attributes in the data stream are separated from noise attributes using a probabilistic learning approach

Challenges

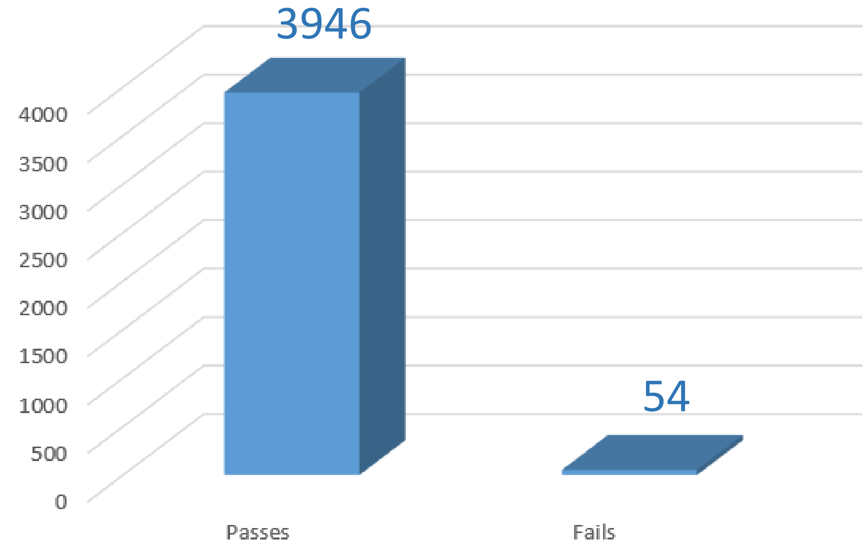
- Streamed data

- Noisy data



How to pick out the significant attribute?

- Unbalanced data



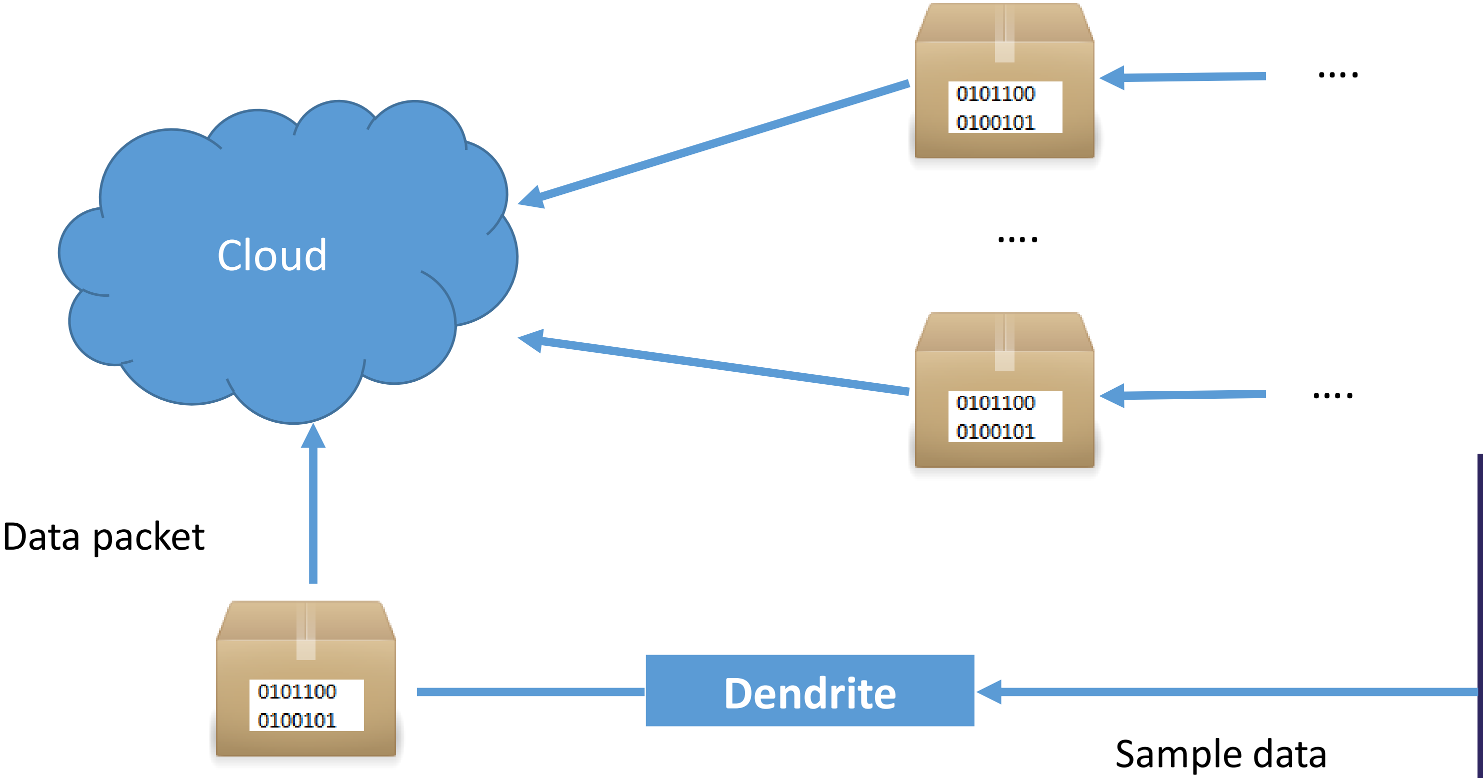
- Also cannot confirm the instrument failed

System Setup

A Data packet contains:
A set of attributes
A set of values
(1-to-1 mapping)

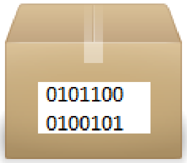
Principle challenge here:

Data is potentially infinite yet only a fixed proportion can be stored

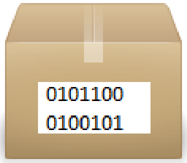


Two classes of data packets:
Failure and non-failure

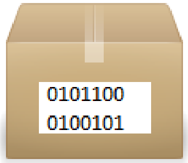
System Setup (2)



&



&



& .... = data stream

The data packets are processed to produce continuous time series

One continuous time series for every attribute on every instrument

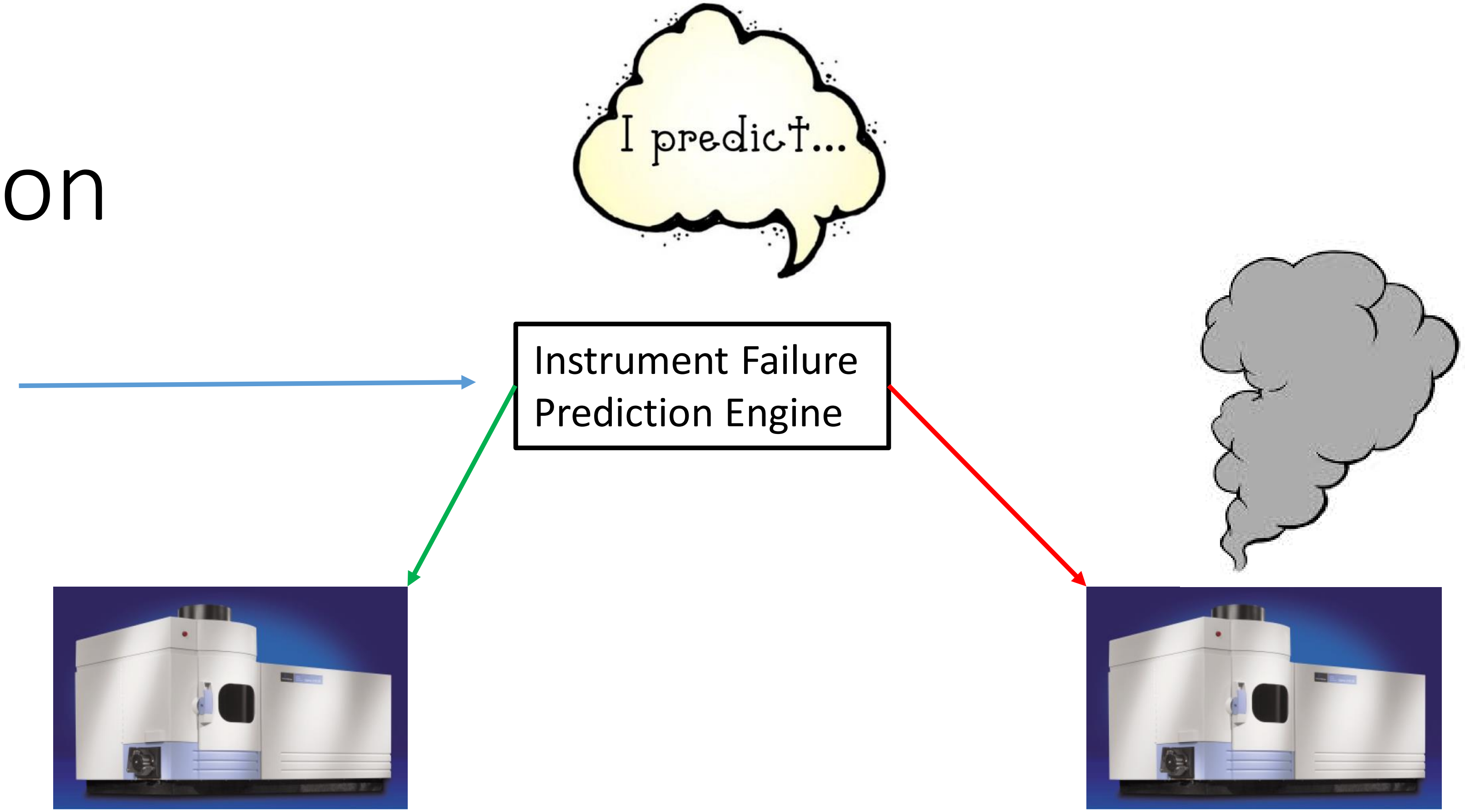
---

Note:

- All instruments are assumed to have the same attributes
- The length of the time series may not be the same across all the instruments
- The time series can be broken up into subsequences

Failure Prediction

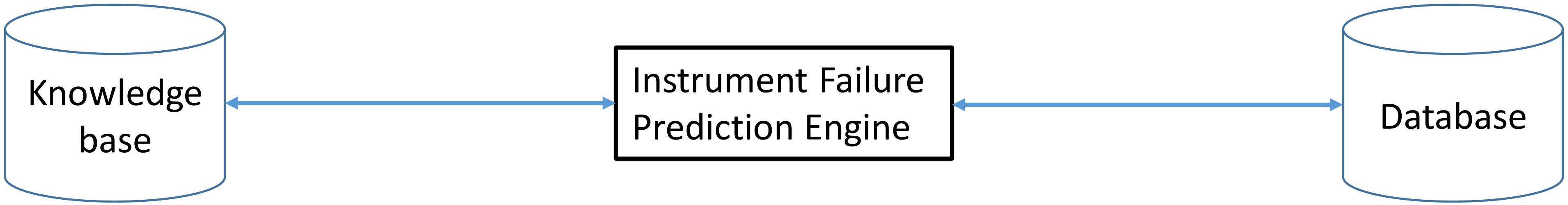
- Given a subsequence



- A learning phase is required.

- We can learn the nature (shape) of subsequences that are good predictors of failure from observing the subsequences associated with the instruments that have failed

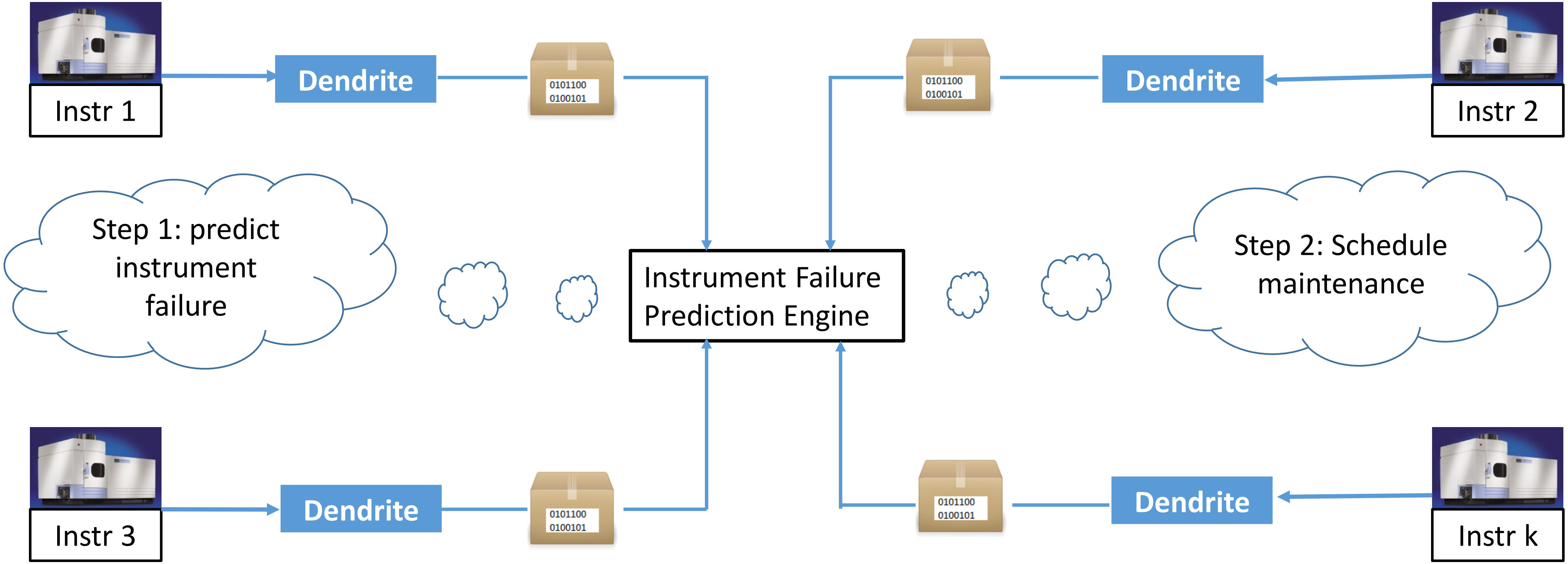
Instrument Failure Prediction Engine



- The database stores subsequences of length p for all possible attributes of each instrument.
- The knowledge base stores subsequences also of length p that are associated with instrument failure.
- Both the database and the knowledge base are empty on startup.

Simulation Environment (Diagram)

A multi-agent based simulation environment was used:



Instrument Failure Prediction Engine

= ∅ because

Main()

If Datapacket = empty then

    addSubsequencesToKB()

    pruneKB()

Else

    Update each time series in DB by adding the new attribute values and removing the last value

    If Not in learning phase then

        prediction(instr_i)

    End If

End If

Adding Subsequences to the KB

AddSubsequenceToKB(instrumentsSubsequence)

For each attribute in the subsequence:

    If the attribute's Subsequence ∈ KB then

        increase count of attribute's subsequence

    Else

        add attribute's subsequence to the KB

    End If

End For

For each subsequence in the knowledge base:

    recalculate weight $= \dfrac{\text{times the subsequence has been recorded as a result of instrument failure}}{|KB|}$

End For

Knowledge Base Pruning

Selection of the most appropriate value for ω is thus important and discussed later

pruneKB(instrumentsSubsequence)

For each instruments and instrument's attribute:

    If an attribute's subsequence's weight is < ω then

        remove the subsequence from the KB

    End If

End For

If KB has changed:

    recalculate weight as described in AddSubsequenceToKB()

End If

Making a prediction: comparing subsequences

- To predict failure, a compassion between subsequences is required

Subsequences:

$$M = \{m_1, m_2, \ldots, m_l\}$$

$$N = \{n_1, n_2, \ldots, n_l\}$$

$$dist(M, N) = \sum_{i=1}^{i=l} (m_i - n_i)^2$$

- We compare the subsequences via Euclidean distance

How to predict failure?

```
prediction(instr_i)
For each attribute subsequence in the database
        if dist(attribute subsequence in DB, attribute's subsequence in KB) ≤ σ then
                Predict failure for instr_i
        End If
End For
```

σ is a pre-set similarity threshold. Selection of the most appropriate threshold value is discussed later

The Simulation Environment

- *k* instruments with *n* attributes each
- The simulation operated on a loop
- Each iteration, each instrument may perform a sampling activity
- Two attribute types: (a) activity dependant; and (b) activity independent
- One attribute was chosen as the sentinel attribute, others provided noise
- The sentinel attribute was an activity dependant attribute and was designated to cause instrument failure once a particular value was reached

Evaluation Metrics

- As we wish to intervene prior to failure, we have no information as to whether our predictions were correct or not.

- Instead we use an accumulated gross profit measure.
- When a sample is completed, a profit of g_{sample} is made.
- When instrument maintenance occurs, a cost of g_{maint}. is incurred.
- When instrument replacement occurs, a cost of $g_{replace}$. Is incurred.

$$g_{sample} < g_{maint} < g_{replace}$$

Single Sentinel Attribute Evaluation

- The aim of these experiments was to determine the effect of different:
- Similarity thresholds (σ)
- Subsequence lengths (p)

Clear correlation between σ and p: A larger p value requires a larger σ for best results.

Best overall result had the lowest σ and p values

p	Similarity threshold σ									
	0	1	2	3	4	5	6	7	8	9
2	**711**	706	687	657	627	593	561	528	504	480
3	691	**696**	692	683	666	650	623	601	583	557
4	650	683	**687**	677	672	663	654	639	622	605
5	575	657	**677**	675	667	660	654	645	640	627
6	470	615	659	**668**	667	661	650	644	636	633
7	351	558	634	657	**660**	658	651	644	636	632
8	248	479	595	635	650	**655**	653	646	639	632
9	167	389	540	606	635	646	**648**	645	641	634

Table 1. Comparison in terms of gross profit ($k = 20$).

Each parameter combination had:

1000 simulations. 200 iterations in each.

p	Similarity threshold σ									
	0	1	2	3	4	5	6	7	8	9
2	**1445**	1425	1381	1319	1260	1186	1126	1064	1014	967
3	**1425**	1409	1396	1374	1338	1300	1256	1210	1167	1120
4	1381	**1400**	1395	1369	1352	1333	1312	1277	1245	1211
5	1299	1374	**1387**	1368	1343	1330	1310	1299	1285	1259
6	1156	1326	**1369**	1364	1348	1331	1306	1291	1277	1268
7	949	1250	1338	**1353**	1347	1336	1316	1291	1276	1262
8	712	1136	1292	1332	**1340**	1334	1321	1303	1284	1265
9	501	986	1223	1299	1325	**1328**	1320	1308	1291	1275

Table 2. Comparison in terms of gross profit ($k = 40$).

Maintained/Failed Instruments

p	Similarity threshold σ									
	0	1	2	3	4	5	6	7	8	9
2	**2**	1	1	1	1	1	1	1	1	1
3	**3**	2	1	1	1	1	1	1	1	1
4	**7**	3	2	2	1	1	1	1	1	1
5	**12**	5	3	2	2	2	1	1	1	1
6	**20**	9	5	3	2	2	2	2	1	1
7	**29**	13	7	4	3	2	2	2	2	2
8	**37**	19	10	6	4	3	3	2	2	2
9	**44**	26	14	9	6	4	4	3	2	2

Table 3. Comparison in terms of number of failed machines ($k = 20$).

p	Similarity threshold σ									
	0	1	2	3	4	5	6	7	8	9
2	**2**	1	1	1	1	1	1	1	1	1
3	**4**	2	1	1	1	1	1	1	1	1
4	**7**	3	2	2	1	1	1	1	1	1
5	**13**	6	3	2	2	2	1	1	1	1
6	**24**	10	5	3	2	2	2	2	1	1
7	**39**	16	8	5	3	3	2	2	2	2
8	**57**	25	12	7	5	4	3	2	2	2
9	**74**	36	18	10	7	5	4	3	3	2

Table 4. Comparison in terms of number of failed machines ($k = 40$).

Note – the average number of failed instruments when maintenance was not scheduled (with 1000 simulations and 200 iterations), was as follows:
54 (when k = 20)
109 (when k = 40)

p	Similarity threshold σ									
	0	1	2	3	4	5	6	7	8	9
2	60	62	63	66	68	70	73	75	76	**78**
3	58	61	62	64	65	66	68	70	71	**73**
4	55	60	62	63	64	65	66	67	68	**69**
5	48	57	60	62	63	64	65	66	**67**	**67**
6	40	53	58	61	63	64	65	66	**67**	**67**
7	30	47	55	59	61	63	64	65	66	**67**
8	21	41	51	56	59	61	63	64	65	**66**
9	13	33	46	53	57	60	61	63	64	**65**

Table 5. Comparison in terms of number of maintained machines ($k = 20$).

p	Similarity threshold σ									
	0	1	2	3	4	5	6	7	8	9
2	122	125	129	133	138	143	147	151	155	**158**
3	120	125	127	129	132	135	138	141	144	**148**
4	116	123	126	129	131	132	134	136	139	**141**
5	109	119	124	127	130	131	133	134	136	**138**
6	97	115	121	125	128	131	133	135	136	**137**
7	80	107	118	123	126	129	131	133	135	**137**
8	60	97	112	119	124	127	130	132	134	**136**
9	42	84	106	115	121	125	128	130	132	**134**

Table 6. Comparison in terms of number of maintained machines ($k = 40$).

The higher the σ, the least precise the prediction, resulting in maintenance frequently being conducted when it was not necessary.

This is why the lower σ, the more likely that an instrument will fail.

Knowledge base size

p	Similarity threshold σ									
	0	1	2	3	4	5	6	7	8	9
2	**2**	1	1	1	1	1	1	1	1	1
3	**3**	2	1	1	1	1	1	1	1	1
4	**6**	3	2	2	1	1	1	1	1	1
5	**12**	5	3	2	2	2	1	1	1	1
6	**20**	9	4	3	2	2	2	2	1	1
7	**29**	13	7	4	3	2	2	2	2	2
8	**37**	19	10	6	4	3	3	2	2	2
9	**44**	26	14	9	6	4	4	3	2	2

Table 7. Comparison in terms of *KB* size ($k = 20$).

p	Similarity threshold σ									
	0	1	2	3	4	5	6	7	8	9
2	**2**	1	1	1	1	1	1	1	1	1
3	**3**	2	1	1	1	1	1	1	1	1
4	**6**	3	2	2	1	1	1	1	1	1
5	**13**	6	3	2	2	2	1	1	1	1
6	**23**	10	5	3	2	2	2	2	1	1
7	**39**	16	8	5	3	3	2	2	2	2
8	**57**	25	12	7	5	4	3	2	2	2
9	**74**	36	17	11	7	5	4	3	3	2

Table 8. Comparison in terms of *KB* size ($k = 40$).

- The number of subsequences in KB decreases as the σ value increases. This is because as σ is increased the prediction becomes less precise so the KB requires fewer subsequences.
- The number of subsequences in the KB also decreases as p decreases; this is because as the p value is reduced the number of possible value combinations making up a time series subsequence also decreases

Finding the best parameter settings for different number of attributes

Note – the average number of failed instruments when maintenance was not scheduled (with 1000 simulations and 1000 iterations), was 281 and the average GP was -140.

# Atts. (n)	σ	p	ω	λ	# Fail. Inst.	# Main. Inst.	Final *KB* Size	# *KB* Values Pruned	GP
2	1	2	0.250	22	23	319	1	28	2884
3	1	2	0.225	25	26	323	1	60	2727
4	1	2	0.250	27	28	320	1	93	2701
5	1	2	0.175	31	32	325	1	137	2527
6	1	2	0.150	29	30	342	1	160	2333
7	1	2	0.125	30	31	352	1	194	2176
8	1	2	0.125	35	36	332	1	261	2308
9	1	2	0.100	36	37	352	1	300	2009
10	2	2	0.100	33	34	394	1	313	1502

Table 9. Best parameter settings for a range of attribute set sizes, and $k = 20$. Average results obtained from 500 simulation runs per parameter permutation, 1000 iterations per simulation

As n was increased, the number of noise attributes were increased and it became harder to predict instrument failure, hence the value of λ (learning window size) increases with n.

learning window and weighting threshold value

ω	Learning window size (λ)							
	24	26	28	30	32	34	36	38
0.050	-14543	-14492	-13685	-13652	-13360	-13448	**-12686**	-12774
0.075	-6801	-6072	-5522	-5010	-5033	-4498	**-4460**	-4518
0.100	-1131	-1056	-529	-637	-173	-12	**22**	12
0.125	890	**1540**	1174	1377	1359	1306	1373	1245
0.150	1913	1788	2049	2097	2194	2061	**2245**	2125
0.175	2251	2291	2307	2355	**2401**	2377	2393	2285
0.200	2276	2406	2364	2292	2454	2428	**2481**	2448
0.225	-138	-140	-139	-138	-139	**-137**	-139	-138
0.250	-139	**-137**	-138	-137	-138	-138	-139	-139

Table 10. Learning window size (λ) versus Weighting threshold (ω), comparison in terms of gross profit ($k = 20$, $n = 5$, $\sigma = 1$ and $p = 2$)

Reminder: ω is the threshold for pruning a subsequence from the KB

For the experiments, ω = 1 and p = 2 were used as these setting had produced the best results.

Experiments show that the choice of ω is important, either side of the optimum value, GP quickly starts to fall.

Yet the size of λ can still influence the results.

Conclusions

- A mechanism, founded on time series analysis, for predicting instrument failure using data stream mining has been proposed.
- The presented evaluation indicated that best results are obtained when the similarity threshold $\sigma = 1$ (almost exact matching between current time series subsequences associated with individual instruments and subsequences in KB) and the size of the subsequences are $p = 2$.
- The optimum value for the learning window size λ increases with n.

Future work

- Lowering the sensitivity associated with the KB pruning threshold value ω.
- Investigate scenarios where we have several sentinel/significant attributes
- Investigate alternative prediction mechanisms, e.g.: dynamic classification, association rule or decision tree based techniques.
- Predict non-failure and well as failure.
- Implement this functionality into a real world app via CSols Dendrite instrument interfaces.
- Finding more accurate valuations for the profit of a sample, the cost of maintenance and the cost of replacement.

Any questions?