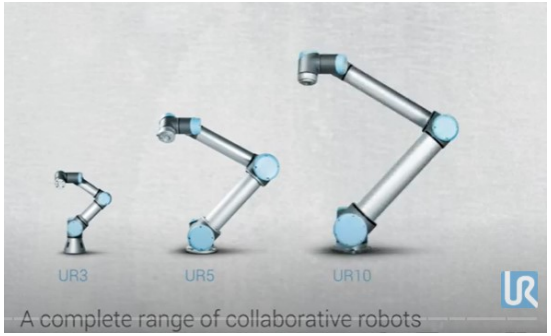
UR3 CobotOps Automation Analysis

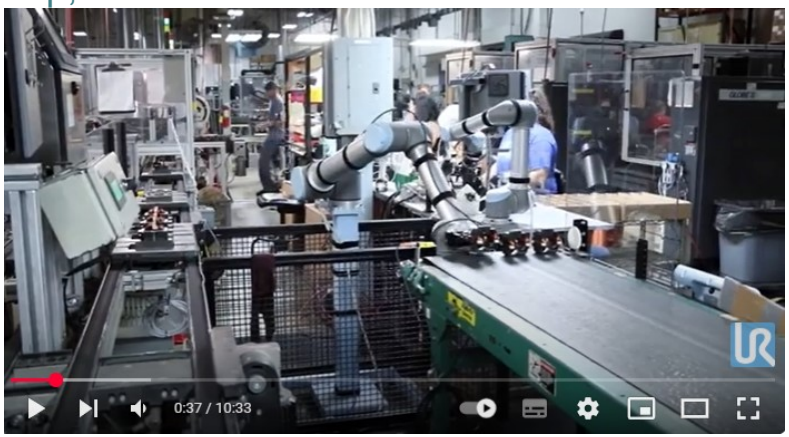
Deirdre Boland

28 Mar 2025

Introduction and Dataset

- **Robotics/automation:** Working with robotics and automation creates workflow efficiencies.
  - To get the most out of your robotic system and optimise planning, you must understand its operation and mitigate for issues.
- **Dataset:** UR3 CobotOps Dataset - time-series data from the UR3 cobot
  - offers insights into operational parameters and faults.
  - Currents, speeds across joints (J0-J5), gripper/tool current, operation cycle count, protective stops, and grip losses
- **Objective:** Explore data and investigate feasibility of error prediction
- **Pre-processing:** All features continuous except cycle, grip lost and protective stop
  - Only continuous features had missing data - filled missing data with data from previous row
  - Grip lost changed False to 0 and True to 1
  - Extracted timestamp, date and time data in correct format





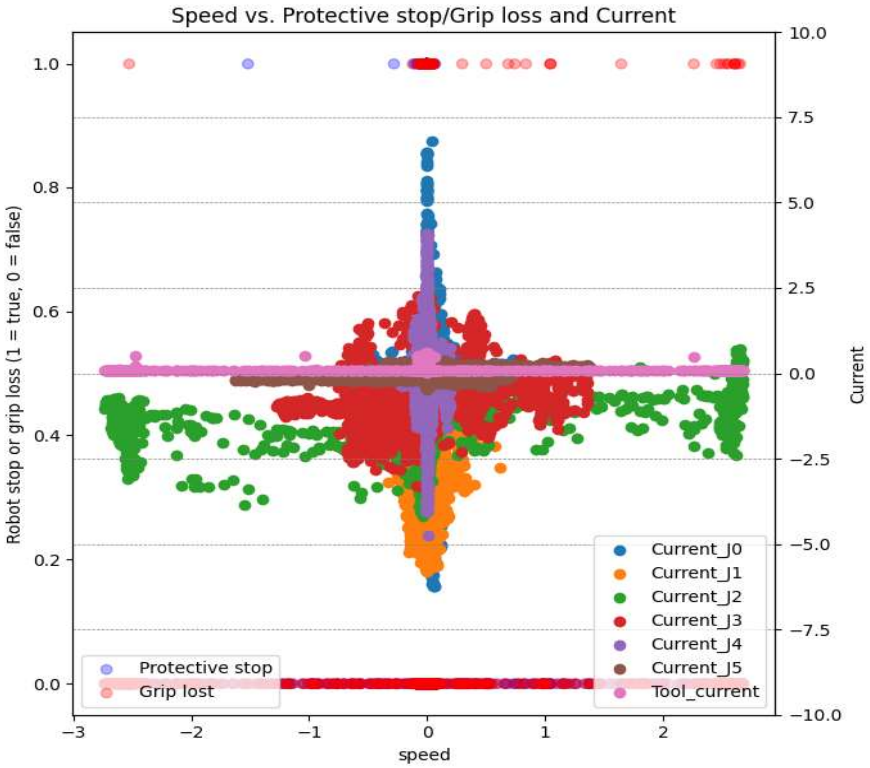
• Dataset Reference: M. Tyrovolas, K. Aliev, D. Antonelli, and C. Stylios. "UR3 CobotOps," UCI Machine Learning Repository, 2024. https://doi.org/10.24432/C5J891

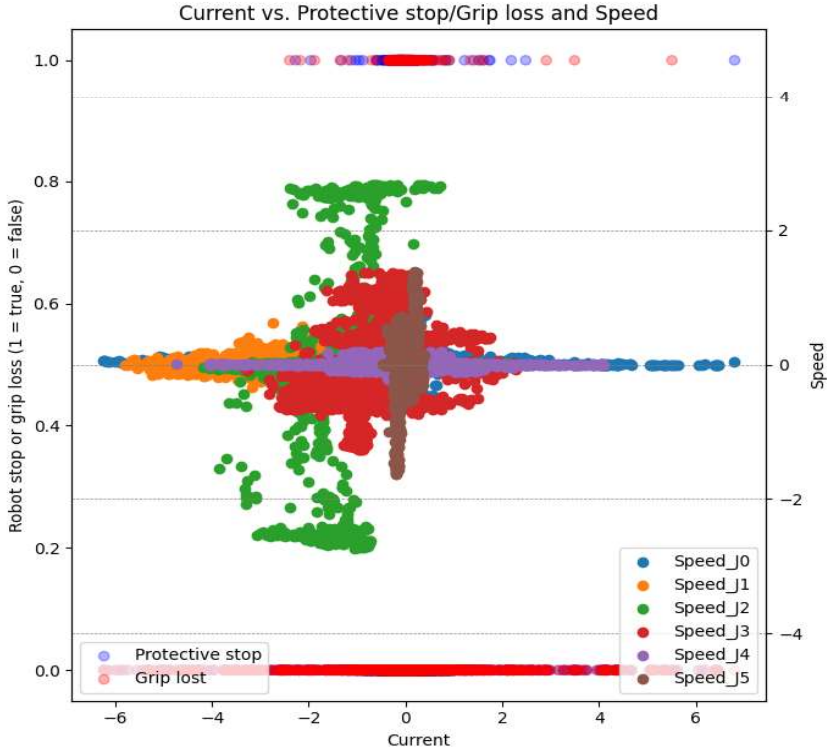
• Video reference: Robotics applications by Universal Robots - Easy Automation with Collaborative Robots [2020]
• Collaboration at all levels – experience Universal Robots in 21 applications at Hanover Fair 2017

EDA: Current and speed

- Current and speed centralised on zero with negative and positive indicating or controlling cobot arm movement in different directions
- Each arm J0-J5 and the tool/gripper has it's own pattern
- Grip loss or protective stop faults happen less in negative range

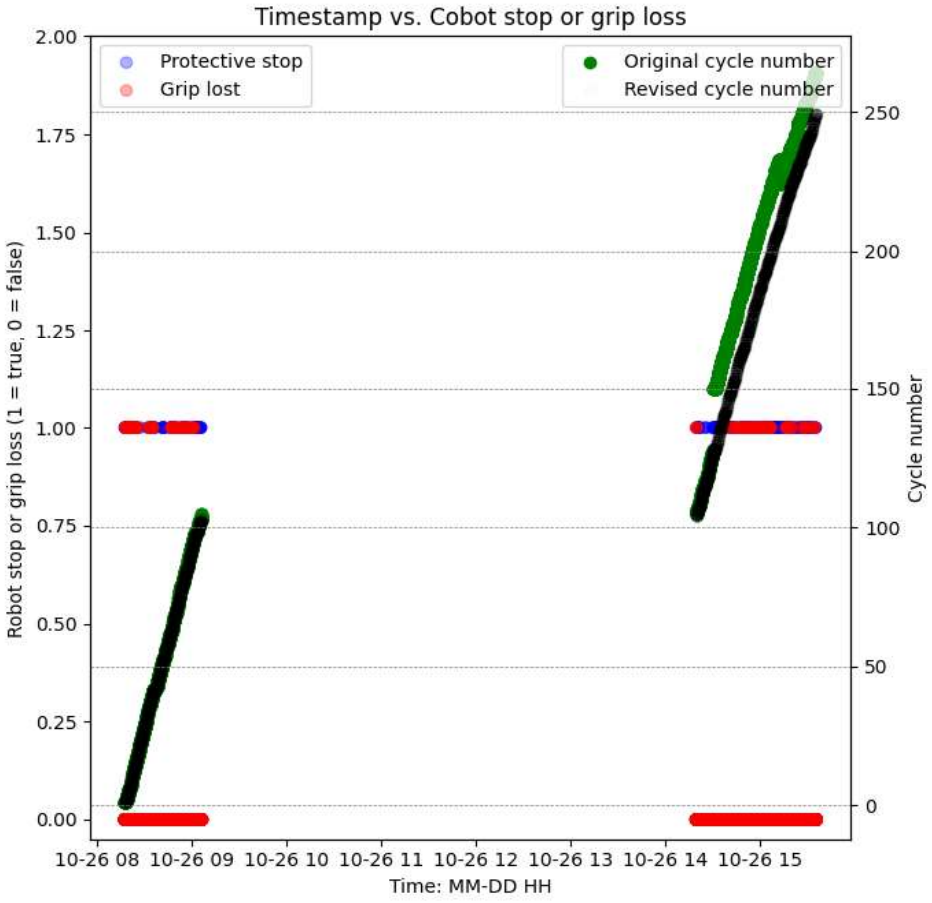
*EDA:Exploratory data analysis





EDA: Cycles and faults over time

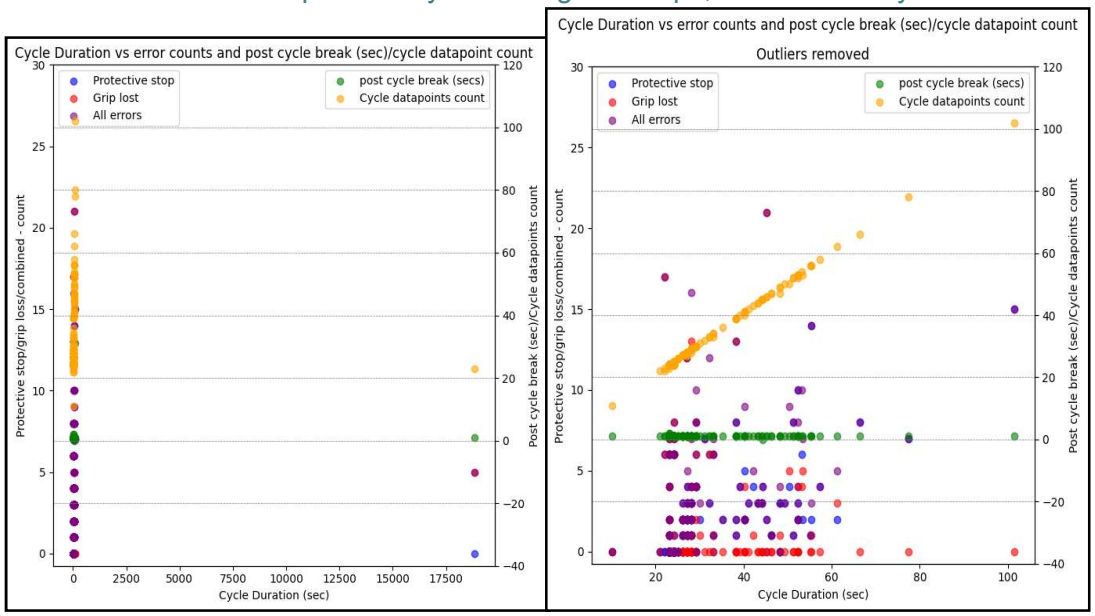
- Major stop in operations from approx 09:00 until 14:00
  - Cycles out of sync of expected after this
  - Could be planned or human intervention and re-planning due to robot issues
- Generated new features
  - New cycle such that cycles are continuous with timestamp
    - Likely errors that cycles were skipped/repeated
  - Cycle timepoint sec
    - Compare cycle stages as cycles are varying durations

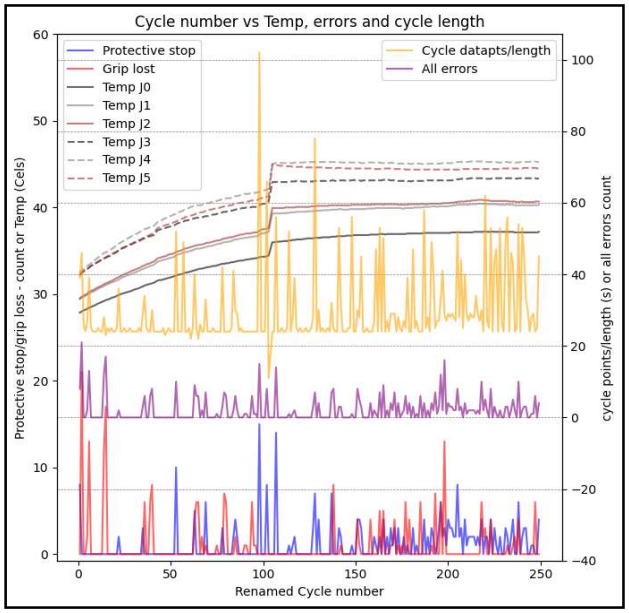


EDA: Cycle Summary – Temperature, duration, post cycle & error behaviour

- Generated summary table for each cycle
  - 2 outlier data rows removed
- Break post cycle consistently approx 1sec
  - Automation controlled unless human interference (deleted outlier)
- Expected might be a relationship of higher error counts and longer cycle duration - linear relationship is only cycle datapoints count
  - Different activities each cycle?
    - or
  - Robot automatically devising next best move, making cycles length inconsistent – is timing an issue for the cycles/process?
- Temperature increase over course of day with uptick at cycle 104 where long break occured
  - Protective stops most cycles at higher temps, later in the day

cyc_rename	104	98	42	128
cyc_original	105	99	42	150
grip_lost	5	0	0	0
protective_stop	0.0	15.0	7.0	7.0
cyc_datapts	23	102	80	78
duration_sec	18835.24	101.482	88.459	77.394
break_post_c_sec	1.004	1.007	31.15	1.006

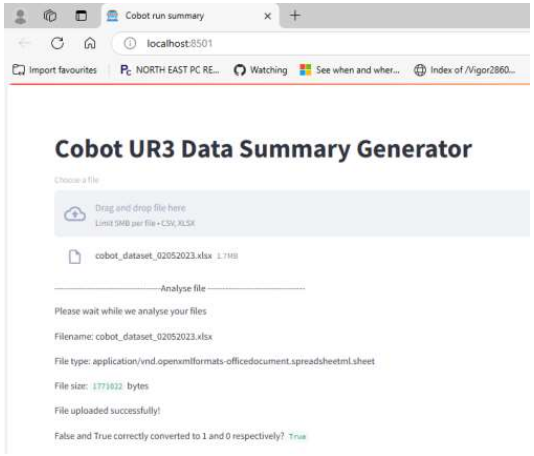


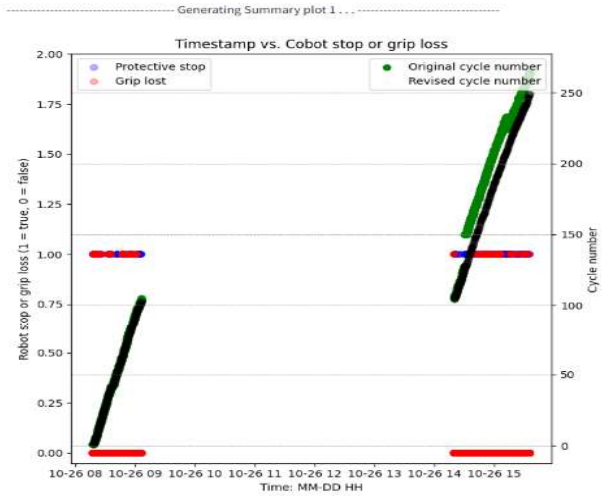


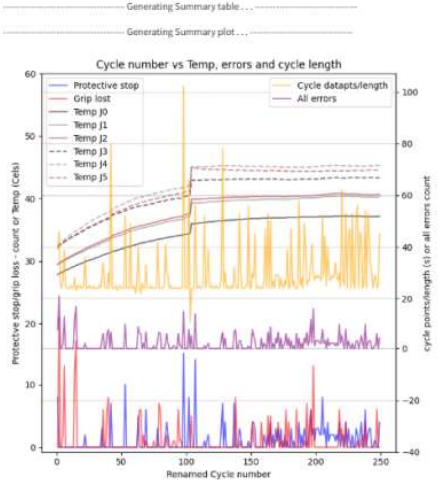
Streamlit (video): Updated 02Apr Data Summary Generator Application

- DevAcProj_CobotOps/Cobot_data_summ_app_vid.mp4 at main · Ddbol/DevAcProj_CobotOps

Streamlit (screenshots): Data Summary Generator Application







These are the cycles where grip loss(es) occured:


```
[
  0 : "2"
  1 : "5"
  2 : "6"
  3 : "14"
  4 : "15"
  5 : "36"
  6 : "39"
  7 : "40"
  8 : "63"
  9 : "64"
  10 : "65"
  11 : "67"
  12 : "69"
  13 : "75"
  14 : "79"
```


These are the cycles where grip loss(es) occured:

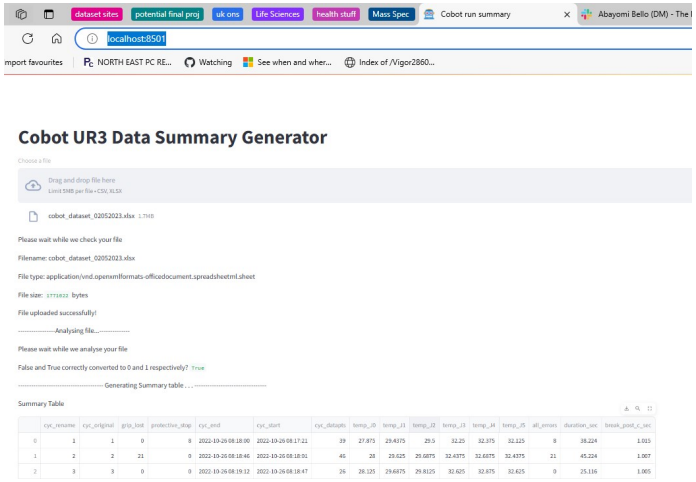
' [. . .]

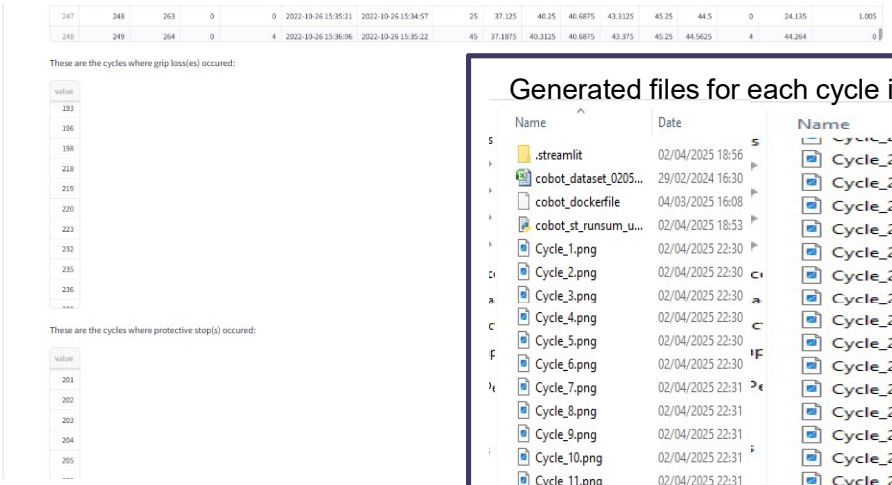
These are the cycles where protective stop(s) occured:

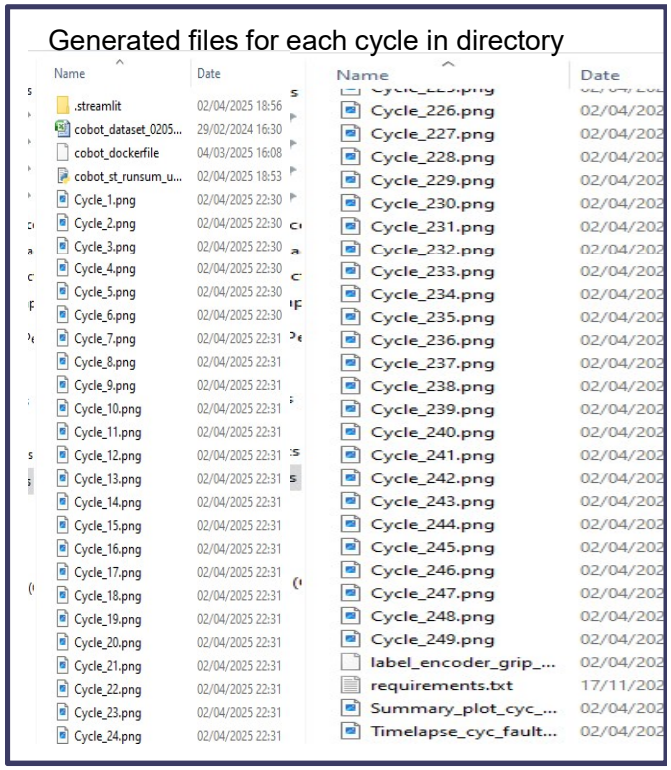

```
[
  0 : "1"
  1 : "22"
  2 : "35"
  3 : "42"
  4 : "53"
  5 : "63"
  6 : "69"
  7 : "78"
  8 : "84"
  9 : "85"
  10 : "86"
  11 : "98"
  12 : "102"
  13 : "107"
  14 : "114"
  15 : "116"
```

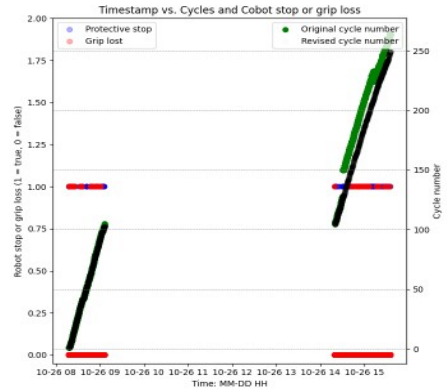

	cyc_rename	cyc_original	grip_lost	protective_stop	cyc_end	cyc_start	duration	cyc_datapts	temp_J0	temp_J1	temp_J2	temp_J3	temp_J4	temp_J5	all_errors	break_point_cyc	duration_sec	break_point_c_sec
0	1	1	0	8.0000	2022-10-26 08:18:00	2022-10-26 08:17:21	a few seconds	39	27.8750	29.4375	29.5000	32.2500	32.3750	32.1250	8.0000	a few seconds	38.2240	1.0150
1	2	2	21	0.0000	2022-10-26 08:18:46	2022-10-26 08:18:01	a minute	46	28.0000	29.6250	29.6875	32.4375	32.6875	32.4375	21.0000	a few seconds	45.2240	1.0070
2	3	3	0	0.0000	2022-10-26 08:19:12	2022-10-26 08:18:47	a few seconds	26	28.1250	29.6875	29.8125	32.6250	32.8750	32.6250	0.0000	a few seconds	25.1160	1.0050
3	4	4	0	0.0000	2022-10-26 08:19:36	2022-10-26 08:19:13	a few seconds	24	28.1875	29.8125	29.9375	32.8125	33.0625	32.7500	0.0000	a few seconds	23.1200	1.0060
4	5	5	2	0.0000	2022-10-26 08:20:03	2022-10-26 08:19:37	a few seconds	27	28.3125	29.9375	30.0625	32.9375	33.2500	32.8750	2.0000	a few seconds	26.1050	1.0020
5	6	6	13	0.0000	2022-10-26 08:20:42	2022-10-26 08:20:04	a few seconds	39	28.4375	30.0625	30.2500	33.0625	33.3750	33.0625	13.0000	a few seconds	38.1990	1.0060
6	7	7	0	0.0000	2022-10-26 08:21:08	2022-10-26 08:20:43	a few seconds	25	28.5000	30.1875	30.3750	33.2500	33.5625	33.2500	0.0000	a few seconds	24.1460	1.0060
7	8	8	0	0.0000	2022-10-26 08:21:32	2022-10-26 08:21:09	a few seconds	24	28.6250	30.2813	30.4375	33.3750	33.7500	33.3750	0.0000	a few seconds	23.1060	1.0050
8	9	9	0	0.0000	2022-10-26 08:21:56	2022-10-26 08:21:33	a few seconds	24	28.6875	30.3750	30.5625	33.5000	33.9063	33.5000	0.0000	a few seconds	23.1030	1.0020
9	10	10	0	0.0000	2022-10-26 08:22:20	2022-10-26 08:21:57	a few seconds	24	28.8125	30.5000	30.6875	33.6250	34.0625	33.6250	0.0000	a few seconds	23.1200	1.0050
10	11	11	0	0.0000	2022-10-26 08:22:45	2022-10-26 08:22:21	a few seconds	25	28.9375	30.6250	30.8125	33.7500	34.2500	33.7500	0.0000	a few seconds	24.1330	1.0030
11	12	12	0	0.0000	2022-10-26 08:23:09	2022-10-26 08:22:46	a few seconds	24	29.0000	30.7500	30.8750	33.8750	34.4063	33.8750	0.0000	a few seconds	23.1540	1.0030
12	13	13	0	0.0000	2022-10-26 08:23:33	2022-10-26 08:23:10	a few seconds	24	29.1250	30.8125	31.0000	34.0000	34.5625	34.0000	0.0000	a few seconds	23.0860	2.0020
13	14	14	12	0.0000	2022-10-26 08:24:02	2022-10-26 08:23:35	a few seconds	28	29.1875	30.9375	31.1250	34.1250	34.7500	34.1250	12.0000	a few seconds	27.0750	1.0030
14	15	15	17	0.0000	2022-10-26 08:24:25	2022-10-26 08:24:03	a few seconds	23	29.3125	31.0625	31.2500	34.1875	34.8125	34.2500	17.0000	a few seconds	22.1080	1.0050
15	16	16	0	0.0000	2022-10-26 08:24:51	2022-10-26 08:24:26	a few seconds	25	29.3750	31.1250	31.3125	34.3125	34.9375	34.3750	0.0000	a few seconds	24.1210	1.0140
16	17	17	0	0.0000	2022-10-26 08:25:15	2022-10-26 08:24:52	a few seconds	24	29.4375	31.2500	31.4375	34.3750	35.0625	34.5000	0.0000	a few seconds	23.1520	1.0020
17	18	18	0	0.0000	2022-10-26 08:25:40	2022-10-26 08:25:16	a few seconds	25	29.5625	31.3125	31.5000	34.5000	35.1875	34.6250	0.0000	a few seconds	24.0820	1.0050
18	19	19	0	0.0000	2022-10-26 08:26:04	2022-10-26 08:25:41	a few seconds	24	29.6250	31.4375	31.6250	34.6563	35.3125	34.7188	0.0000	a few seconds	23.1380	1.0040

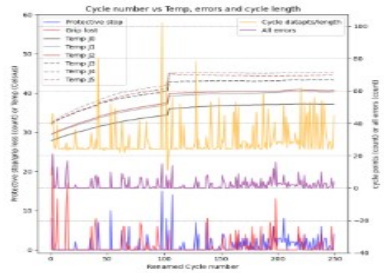
Streamlit (screenshots): Updated 02Apr Data Summary Generator Application





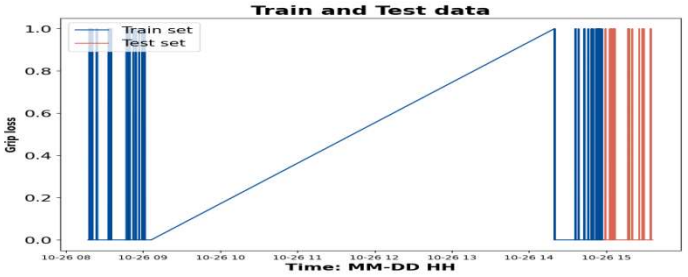


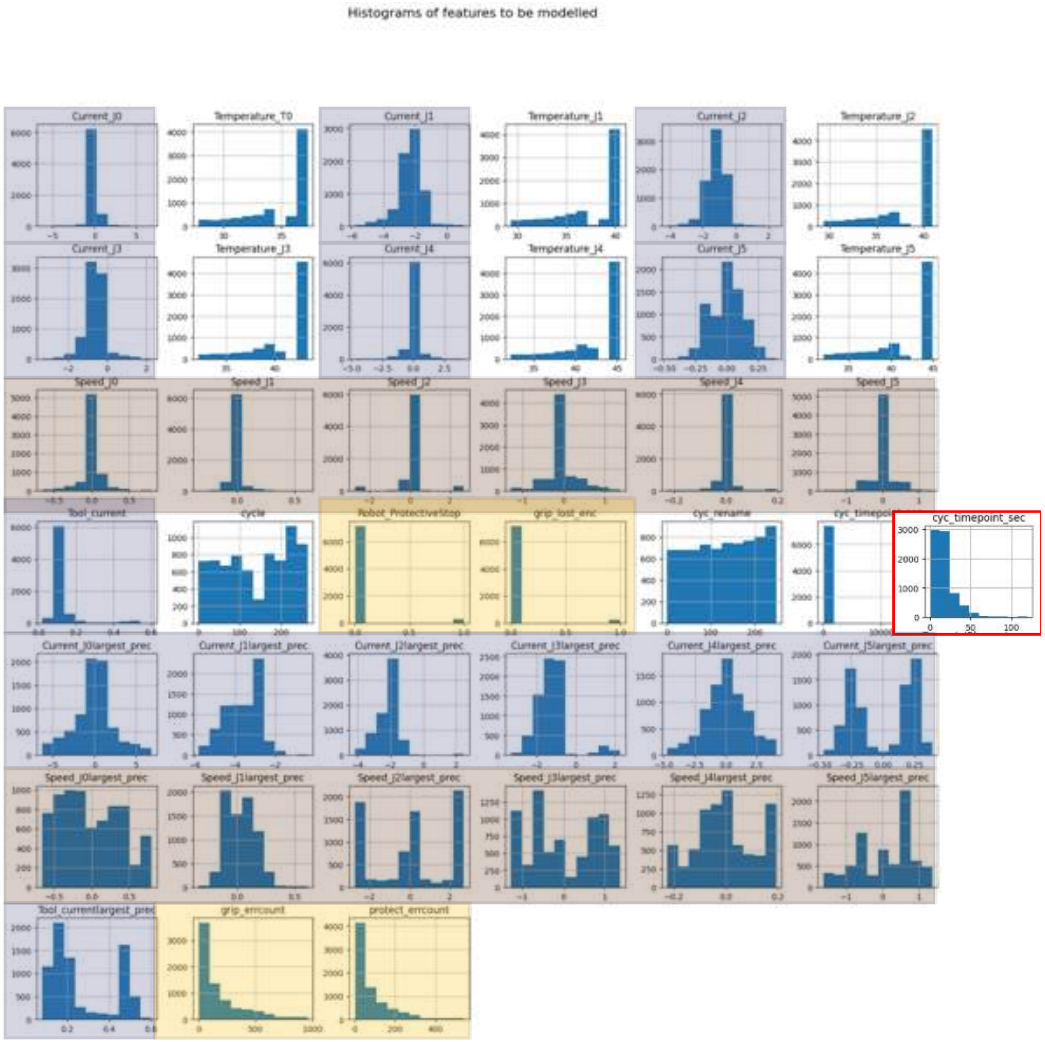




Machine Learning (ML) modelling prep

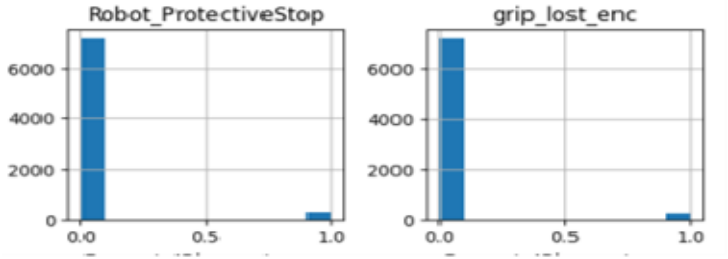
- Timeseries - Ideally predict next fault/error based on previous data of day. (80/20 trainin/testing dataset)
  - Used AutoML AutoGluon to check feasibility but returned no models – likely due to large chunks missing data
- Transform the dataset so can solve with traditional, tabular classification algorithms - Connect data from previous rows
  - Errors/faults – Very imbalanced, mostly no errors/faults
  - Errors/faults – Count up how long since last error happened
  - Current and speed – look back at previous 10 rows and extract the number of largest magnitude (positive or negative)
  - Removed timestamp - covered by cycle and cycle timepoint
  - Clipped cycle timepoint (secs) to remove several hour outlier
  - AutoML AutoGluon suggested RandomForest and KNN, among others
- Ref: https://stackoverflow.com/questions/44744584/multiple-time-series-with-binary-grip_lost_enc-prediction

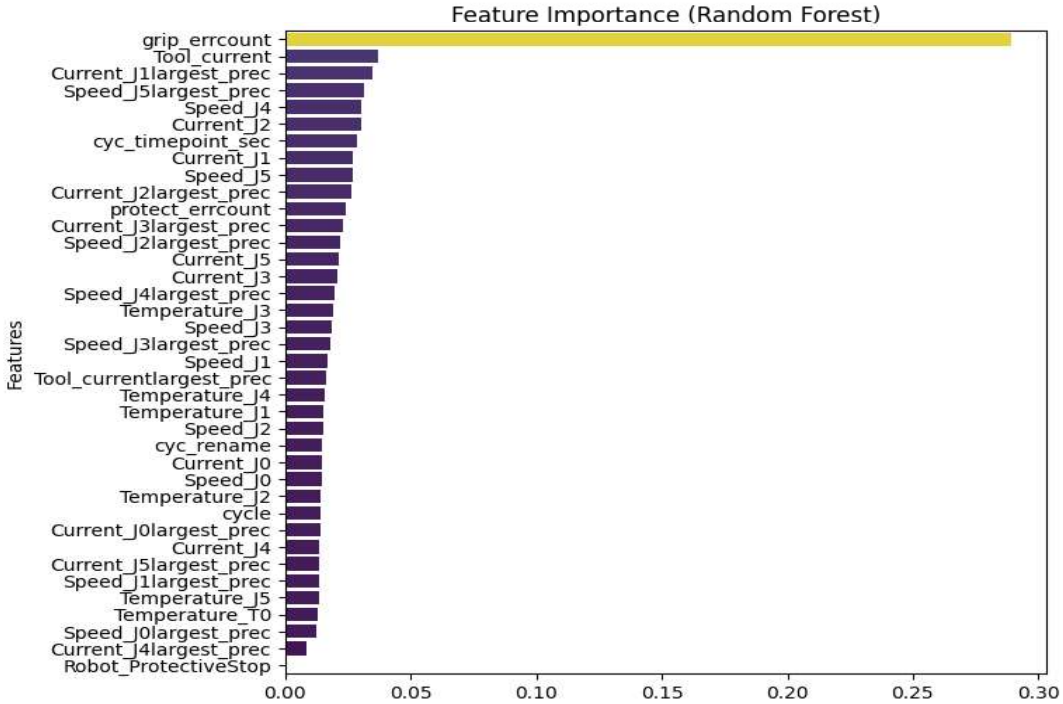


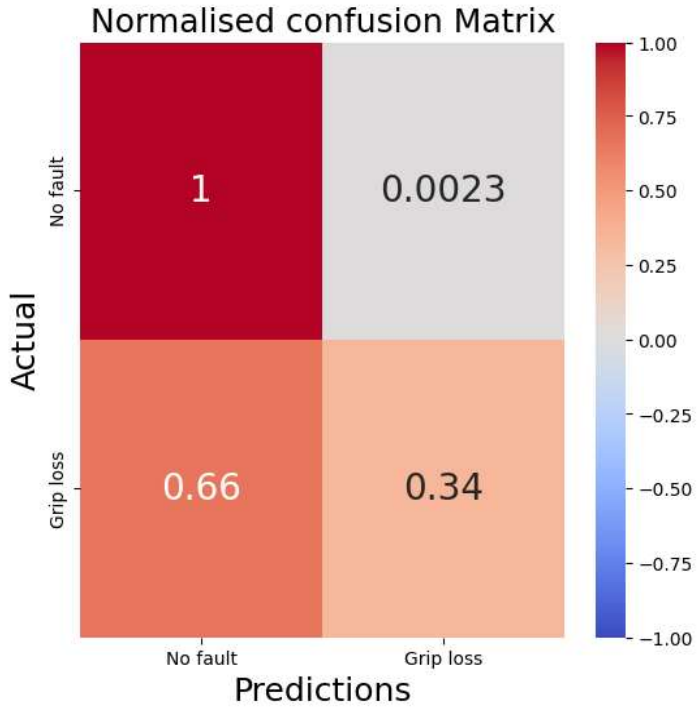


ML modelling - results

- Data scaled
- Random Forest and KNN trialled based on AutoGluon results
- Due to imbalanced data, high accuracy but not good at predicting the grip loss correctly
- Best Random Forest gave AUC of 0.67 (measures how well predicts both outcomes), accuracy 0.97 or 97%
- Unsurprising that the count of how long since last error happened has high importance







ML modelling considerations

- Want to predict rest of days risk of errors/faults but need some information about what robot plans to do
  - Inconsistent cycle time which errors add to
    - hard to provide in advance for modelling
  - Discuss with team if can correlate robot programmed actions to this data better
    - Potentially better predictability if can generate expected robot actions and relate it back to the data the UR3 cobot is generating
- Dataset authors using Information Flow Based – Fuzzy Cognitive Maps (IF- FCMs) to help predict machine operations – may be better approach to investigate
  - modification of FCMs – visual modelling approach. Image on right shows FCM of government control of tobacco. Part of modelling is to adjust weightings of interactions

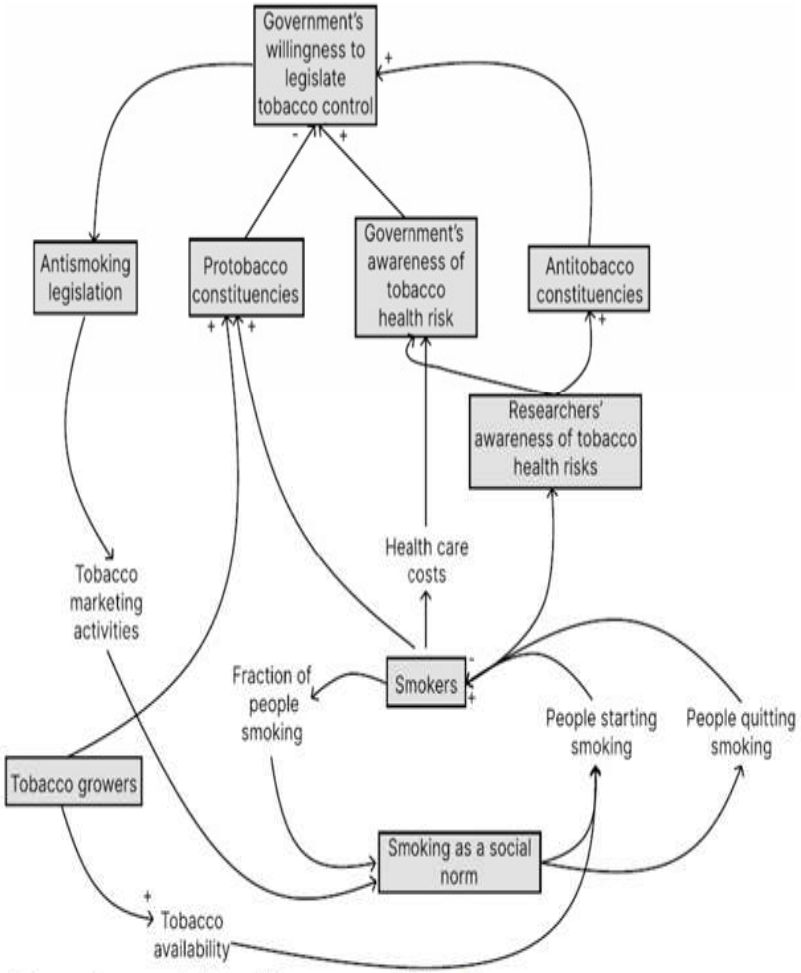


Fig. 2 An excerpt of a more complex logic model of government control of tobacco [7]

References IF-FCMs and FCMs

- Tyrovolas, M., Stylios, C., Aliev, K., Antonelli, D. (2024). Leveraging Information Flow-Based Fuzzy Cognitive Maps for Interpretable Fault Diagnosis in Industrial Robotics. In: Camarinha-Matos, L.M., Ferrada, F. (eds) Technological Innovation for Human-Centric Systems. DoCEIS 2024. IFIP Advances in Information and Communication Technology, vol 716. Springer, Cham. https://doi.org/10.1007/978-3-031-63851-0_6
- Tyrovolas, M., Liang, X.S. & Stylios, C. Information flow-based fuzzy cognitive maps with enhanced interpretability. *Granul. Comput.* **8**, 2021–2038 (2023). https://doi.org/10.1007/s41066-023-00417-7 (Link to github marios-tyrovolas/Information-Flow-Based-Fuzzy-Cognitive-Maps-with-Enhanced-Interpretability: Information Flow-Based Fuzzy Cognitive Maps with Enhanced Interpretability)
- Mkhitaryan et al. BMC Public Health. (2023) How to use machine learning and fuzzy cognitive maps to test hypothetical scenarios in health behavior change interventions: a case study on fruit intake https://doi.org/10.1186/s12889-023-17367-z

Conclusions and Recommendations

- Extensive exploratory data analysis lead to greater understanding of UR3 Cobot behaviour
  - This should help with process understanding, optimisation and planning
  - Generation of application to view summary data from UR3 Cobot logs to support this - to be deployed on Docker
- More knowledge from team about desired outcome of cycles, impacts of inconsistent cycle times and what other data it would be useful to explore/make available to them
- IF-CMs may be a better approach to automation/robotics operations modelling compared to traditional machine learning methods

Thank you! Q&A?