THE UNIVERSITY of York

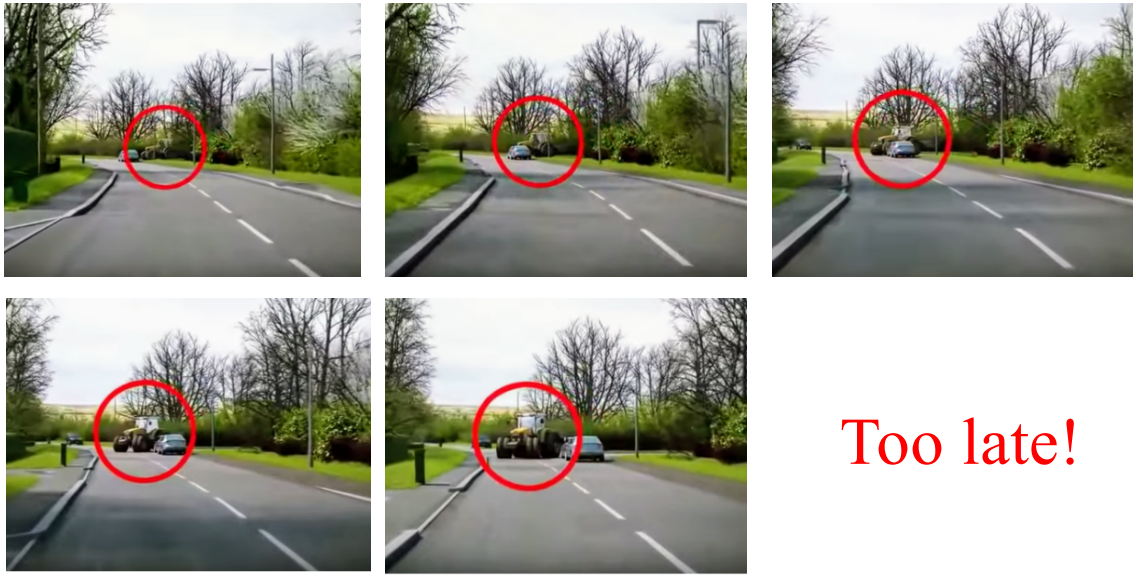
RTS York

Scheduling Classifiers for Real-Time Hazard Perception Considering Functional Uncertainty

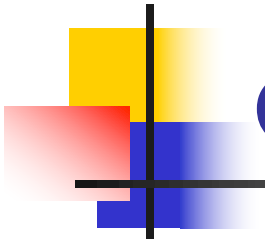
Tarek Abdelzaher, Sanjoy Baruah, Iain Bate, Alan Burns,
Robert I. Davis, Yigong Hu

Overview: Problem

- **Focus of this Research**
  - Perception in autonomous mobile Cyber-Physical Systems which is typically performed using classifiers that are based on Deep Learning (Deep Neural Networks)
- **Motivation**
  - Problems where the system must check that a designated area ahead is free of hazards
    Must identify any hazard within a specified latency constraint
  - Example from UK driving test hazard perception scenarios



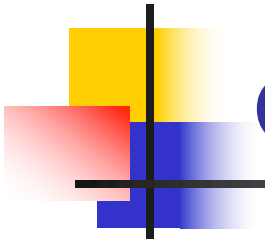
Too late!



Overview: Problem

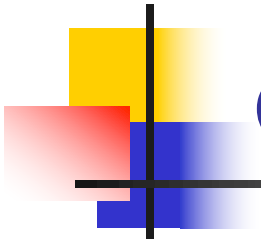
- **Functional uncertainty**
  - Classifiers are imperfect – in the majority of cases a classifier will determine correctly whether there is a hazard present or the area is clear
  - BUT it may produce ***false positives***, i.e. indicate hazard when the area is clear
  - and ***false negatives***, i.e. indicate clear when there is in fact a hazard

- **Problem**
  - Correctly identify hazards
  - Within a latency constraint as late identification can be as bad as no identification
  - ***False positives are undesirable*** as they reduce quality of service, e.g., unnecessarily slowing the vehicle wastes energy and lengthens journey times
  - ***False negatives are a potential safety concern,*** e.g. an emergency braking system have to take over, hence a hard constraint is placed on the maximum permitted probability of false negatives



Overview: Solution

- **One classifier is typically not enough**
  - No single classifier alone may be effective enough to meet the constraint on the maximum permitted probability of false negatives
  - Use multiple classifiers and logically-OR together their outputs (1 = hazard, 0 = clear)
    - If ***any*** classifier indicates hazard then we assume hazard
    - Only if ***all*** classifiers that are run indicate clear do we assume clear
- **Trade-off**
  - Using multiple classifiers:
    - This can reduce the probability of false negatives
    - But inevitably increases the probability of false positives
    - Also increases the overall execution time

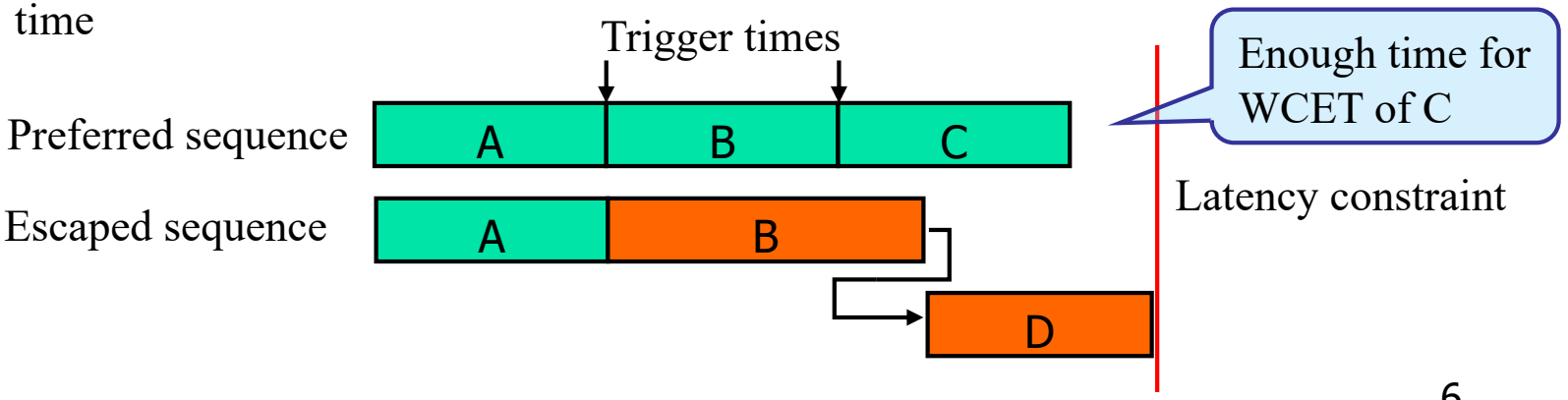


Overview: Solution

- **This research**
  - Provides a method for characterizing the (***arbitrary) statistical dependences*** between the functional behaviours of different classifiers that occur in practice
  - Enables the calculation of the probabilities of false negatives and false positives
- **Derives a Typical-Case Optimal Algorithm**
  - For scheduling classifiers that:
    - minimizes the probability of false positives
    - meets the constraint on the maximum permitted probability of false negatives
    - meets the latency constraint
  - Solution is optimal assuming that the classifiers execute for their typical-case execution times, but ***crucially are not guaranteed to do so***
    - if some classifier exceeds its typical-case execution time, e.g. takes its worst-case execution time then the algorithm still ensures the constraints are met by making use of other classifiers

Overview: Solution

- **Typical-case optimal algorithm**
  - Determines:
    - ***Preferred sequence*** of classifiers to run
    - ***Trigger times*** and ***escape sets*** giving the subsets of classifiers to run if a preferred classifier does not complete by its trigger time
    - Trigger times are computed with respect to typical-case execution times, so if these are observed the preferred sequence of classifiers will run
    - The escape sets guarantee that the constraint on the maximum permitted probability of false negatives will be met even if some or all of the classifiers take their worst-case execution times
    - Control switches to an escape set if a classifier does not complete by its trigger time



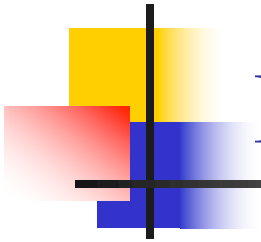
Detail: System Model

- **Classifiers**
  - n classifiers $K_1, K_2, \ldots, K_n$ designed to solve the same problem
  - Output either 1 = hazard or 0 = clear
  - Outputs of multiple classifiers are OR-ed together

  - S represents a subset of the classifiers, with n classifiers there are 2^n such subsets
  - $FN(S)$ probability of the classifiers in S returning a false negative
  - $FP(S)$ probability of the classifiers in S returning a false positive
  - $WCET(S)$ worst-case execution time of the classifiers in S
  - $TCET(S)$ typical-case execution time of the classifiers in S
  - $ACET(S)$ actual-case execution time of the classifiers in S for a specific run-time instance

  - H maximum permitted probability of false negatives
  - L latency constraint

  - $ESCAP(S)$ is the subset V with the smallest $WCET(V)$ such that $FN(S \cup V) \leq H$
    If $FN(S) \leq H$ then $ESCAP(S) = \{\}$



Method

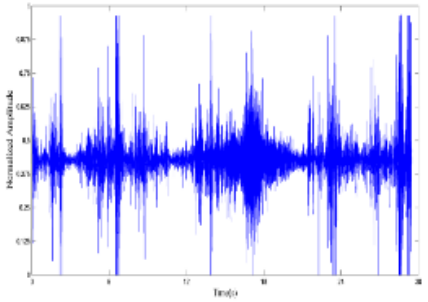
- **Profiling**
  - Collects representative data necessary to characterized the behaviour of the classifiers and their dependences
  - Processes this data into a form usable by the algorithm
- **Offline part of the algorithm**
  - Uses a Directed Acyclic Graph (DAG) representation of the problem to determine the preferred sequence of classifiers to run
  - Takes ***exponential time*** which is Ok in practice as at most 10 to 12 classifiers might be used for this type of problem and the method can cater for up to 20 (in around 20 minutes on a laptop)
- **Online part of the algorithm**
  - Makes decisions between pre-computed choices
  - Takes ***linear time*** at each scheduling point (classifier completion)

Multi-Modal Case Study

- **Data from a previous project**
  - Seeks to autonomously detect the presence of a potentially hostile enemy vehicle in a battlefield environment
  - ***Electronic tripwire functionality***: aim is to determine if a vehicle of the designated type is present in the detection area and generate an alert, but ignore other traffic
  - Vehicle types used in the case study were: Polaris ATV (All Terrain Vehicle), Warthog UGV (Unmanned Ground Vehicle), and Chevrolet Silverado
  - Warthog UGV designated as a hazard

Multi-Modal Case Study

- **Classifiers used**
  - That analyse camera images, ***acoustic***, and ***seismic*** data
  - Initially five classifiers A-E were studied so the profile table could be illustrated
    - A deepsense_both
    - B deepsense_both_contras
    - C deepsense_acoustic
    - D deepsense_seismic
    - E cnn_both
    - F cnn_acoustic
    - G cnn_seismic



  - Up to seven classifiers A-G were used in all, with different combinations of acoustic and seismic data, different neural network architectures, and contrastive learning
  - Used 1800 randomly selected input samples, 600 with a hazard (Warthog UGV) and 1200 without (Polaris ATV and Chevrolet Silerado)
  - Classifiers were run on one core of a Raspberry Pi 4
  - (Yolo classifiers using visual data were not considered due to much longer execution times)

Profile Table

- **Profile Table records**
  - GT1 number of times that the binary pattern in column 1 occurred when ***all*** classifiers were run and the ground truth was hazard = 1
  - GT0 number of times that the binary pattern in column 1 occurred when ***all*** classifiers were run and the ground truth was clear = 0
- **From this data calculate:**
  - FP(S) probability of false positives when running the classifiers in subset S
  - FN(S) probability of false negatives when running the classifiers in subset S
  - Also record WCET(S) for the subset of classifiers S

Multi-Modal Case Study: Profile Table

Binary	Classifiers S	$GT1$	$GT0$	$FP(S)$	$FN(S)$	$WCET(S)$
00000	∅	36	1107	0.0000	1.0000	0
00001	A	3	2	0.0075	0.1183	0.025121
00010	B	1	1	0.0042	0.1383	0.023854
00011	AB	0	0	0.0100	0.0933	0.048975
00100	C	3	18	0.0200	0.3600	0.017554
00101	AC	9	1	0.0250	0.0933	0.042675
00110	BC	2	0	0.0242	0.1067	0.041408
00111	ABC	5	0	0.0275	0.0850	0.066529
01000	D	9	36	0.0358	0.2083	0.01618
01001	AD	4	1	0.0417	0.0883	0.0413
01010	BD	1	1	0.0383	0.1067	0.040033
01011	ABD	22	1	0.0433	0.0750	0.065154
01100	CD	0	2	0.0542	0.0850	0.033734
01101	ACD	1	0	0.0575	0.0700	0.058854
01110	BCD	2	0	0.0567	0.0767	0.057587
01111	ABCD	13	0	0.0592	0.0667	0.082708
10000	E	4	22	0.0250	0.1850	0.0053
10001	AE	3	1	0.0292	0.0900	0.030421
10010	BE	1	1	0.0275	0.1083	0.029154
10011	ABE	3	1	0.0308	0.0800	0.054274
10100	CE	2	1	0.0425	0.1267	0.022854
10101	ACE	4	2	0.0458	0.0783	0.047975
10110	BCE	4	0	0.0450	0.0867	0.046708
10111	ABCE	45	0	0.0475	0.0750	0.071828
11000	DE	2	2	0.0592	0.0983	0.021479
11001	ADE	3	0	0.0617	0.0700	0.0466
11010	BDE	2	0	0.0600	0.0850	0.045333
11011	ABDE	122	0	0.0625	0.0650	0.070454
11100	CDE	0	0	0.0750	0.0667	0.039033
11101	ACDE	0	0	0.0767	0.0617	0.064154
11110	BCDE	2	0	0.0758	0.0650	0.062887
11111	ABCDE	292	0	0.0775	0.0600	0.088008
	Sum	600	1200			

Optimal Algorithms

- **Statically Optimal Algorithm**
  - Select the subset S of classifiers (row in the table) with the lowest FP(S) such that FN(S) ≤ H and WCET(S) ≤ L and run the classifiers in any order
- **Clairvoyant Optimal Algorithm**
  - Assumed to know the actual execution times of the classifiers for each run-time instance, but not their behaviour
  - Select the subset S of classifiers (row in the table) with the lowest FP(S) such that FN(S) ≤ H and ACET(S) ≤ L and run the classifiers in any order

Multi-Modal Case Study: Profile Table

Binary	Classifiers S	$GT1$	$GT0$	$FP(S)$	$FN(S)$	$WCET(S)$
00000	∅	36	1107	0.0000	1.0000	0
00001	A	3	2	0.0075	0.1183	0.025121
00010	B	1	1	0.0042	0.1383	0.023854
00011	AB	0	0	0.0100	0.0933	0.048975
00100	C	3	18	0.0200	0.3600	0.017554
00101	AC	9	1	0.0250	0.0933	0.042675
00110	BC	2	0	0.0242	0.1067	0.041408
00111	ABC	5	0	0.0275	0.0850	0.066529
01000	D	9	36	0.0358	0.2083	0.01618
01001	AD	4	1	0.0417	0.0883	0.0413
01010	BD	1	1	0.0383	0.1067	0.040033
01011	ABD	22	1	0.0433	0.0750	0.065154
01100	CD	0	2	0.0542	0.0850	0.033734
01101	ACD	1	0	0.0575	0.0700	0.058854
01110	BCD	2	0	0.0567	0.0767	0.057587
01111	ABCD	13	0	0.0592	0.0667	0.082708
10000	E	4	22	0.0250	0.1850	0.0053
10001	AE	3	1	0.0292	0.0900	0.030421
10010	BE	1	1	0.0275	0.1083	0.029154
10011	ABE	3	1	0.0308	0.0800	0.054274
10100	CE	2	1	0.0425	0.1267	0.022854
10101	ACE	4	2	0.0458	0.0783	0.047975
10110	BCE	4	0	0.0450	0.0867	0.046708
10111	ABCE	45	0	0.0475	0.0750	0.071828
11000	DE	2	2	0.0592	0.0983	0.021479
11001	ADE	3	0	0.0617	0.0700	0.0466
11010	BDE	2	0	0.0600	0.0850	0.045333
11011	ABDE	122	0	0.0625	0.0650	0.070454
11100	CDE	0	0	0.0750	0.0667	0.039033
11101	ACDE	0	0	0.0767	0.0617	0.064154
11110	BCDE	2	0	0.0758	0.0650	0.062887
11111	ABCDE	292	0	0.0775	0.0600	0.088008
	Sum	600	1200			

Profiling: Dependences

- **Dependences**
  - Examine the ***profiling data*** directly to assess the level of dependences and correlations between the different classifiers
  - Illustrated via Pearson's correlation coefficient
  - Classifier behaviour is strongly positively correlated, coefficient between 0.433 and 0.931 for each pair of classifiers
  - Must therefore account for ***arbitrary dependences*** between classifier behaviours
  - Classifier execution times are very weakly correlated, abs(coefficient) < 0.08, meaning that ***independence*** of execution times is a reasonable assumption

Pearson's correlation coefficient: Behaviour

	A	B	C	D	E	F	G
A	1	0.931	0.725	0.815	0.860	0.717	0.687
B	0.931	1	0.721	0.832	0.872	0.723	0.6944
C	0.725	0.721	1	0.570	0.687	0.819	0.440
D	0.815	0.832	0.570	1	0.747	0.579	0.699
E	0.860	0.872	0.687	0.747	1	0.711	0.717
F	0.717	0.723	0.819	0.579	0.711	1	0.433
G	0.687	0.694	0.440	0.699	0.717	0.433	1

Pearson's correlation coefficient: Execution times

	A	B	C	D	E	F	G
A	1	0.031	0.036	-0.011	-0.027	-0.009	0.022
B	0.031	1	0.009	0.024	-0.040	-0.013	-0.024
C	0.036	0.009	1	0.000	0.029	-0.004	0.031
D	-0.011	0.024	0.000	1	-0.020	0.062	-0.058
E	-0.027	-0.040	0.029	-0.020	1	-0.007	0.076
F	-0.009	-0.013	-0.004	0.062	-0.007	1	0.024
G	0.022	-0.024	0.031	-0.058	0.076	0.024	1

Typical-Case Optimal Algorithm

- **Specific Problem Instance**
  - Latency L = 50ms, maximum permitted probability of false negatives H = 0.085
- **Augment the profile table**
  - ***Add TCET(S):*** sum of the values of the chosen percentile of the execution time distribution for each classifier in S (Summation is valid as execution times can be assumed independent)
  - Use 70-percentile as an example – return to this choice later
  - ***Add ESCAP(S)*:** equates to the subset V with the smallest WCET(V) such that FN(S∪V) ≤ H
    If FN(S) ≤ H then ESCAP(S) = {}
  - Computing ESCAP(S) for all 2^n subsets takes O(4^n) time

Multi-Modal: Augmented Profile Table

Binary	Classifiers *S*	*FP*(*S*)	*FN*(*S*)	*WCET*(*S*)	*TCET*(*S*)	*ESCAP*(*S*)
00000	∅	0.0000	1.0000	0	0	CD
00001	A	0.0075	0.1183	0.025121	0.018166	DE
00010	B	0.0042	0.1383	0.023854	0.017788	DE
00011	AB	0.0100	0.0933	0.048975	0.035954	E
00100	C	0.0200	0.3600	0.017554	0.012263	D
00101	AC	0.0250	0.0933	0.042675	0.030429	E
00110	BC	0.0242	0.1067	0.041408	0.030051	D
00111	ABC	0.0275	0.0850	0.066529	0.048217	∅
01000	D	0.0358	0.2083	0.01618	0.011878	C
01001	AD	0.0417	0.0883	0.0413	0.030044	E
01010	BD	0.0383	0.1067	0.040033	0.029666	E
01011	ABD	0.0433	0.0750	0.065154	0.047832	∅
01100	CD	0.0542	0.0850	0.033734	0.024141	∅
01101	ACD	0.0575	0.0700	0.058854	0.042307	∅
01110	BCD	0.0567	0.0767	0.057587	0.041929	∅
01111	ABCD	0.0592	0.0667	0.082708	0.060095	∅
10000	E	0.0250	0.1850	0.0053	0.004112	CD
10001	AE	0.0292	0.0900	0.030421	0.022277	D
10010	BE	0.0275	0.1083	0.029154	0.0219	D
10011	ABE	0.0308	0.0800	0.054274	0.040066	∅
10100	CE	0.0425	0.1267	0.022854	0.016374	D
10101	ACE	0.0458	0.0783	0.047975	0.03454	∅
10110	BCE	0.0450	0.0867	0.046708	0.034162	D
10111	ABCE	0.0475	0.0750	0.071828	0.052328	∅
11000	DE	0.0592	0.0983	0.021479	0.01599	C
11001	ADE	0.0617	0.0700	0.0466	0.034156	∅
11010	BDE	0.0600	0.0850	0.045333	0.033778	∅
11011	ABDE	0.0625	0.0650	0.070454	0.051944	∅
11100	CDE	0.0750	0.0667	0.039033	0.028252	∅
11101	ACDE	0.0767	0.0617	0.064154	0.046418	∅
11110	BCDE	0.0758	0.0650	0.062887	0.046041	∅
11111	ABCDE	0.0775	0.0600	0.088008	0.064206	∅

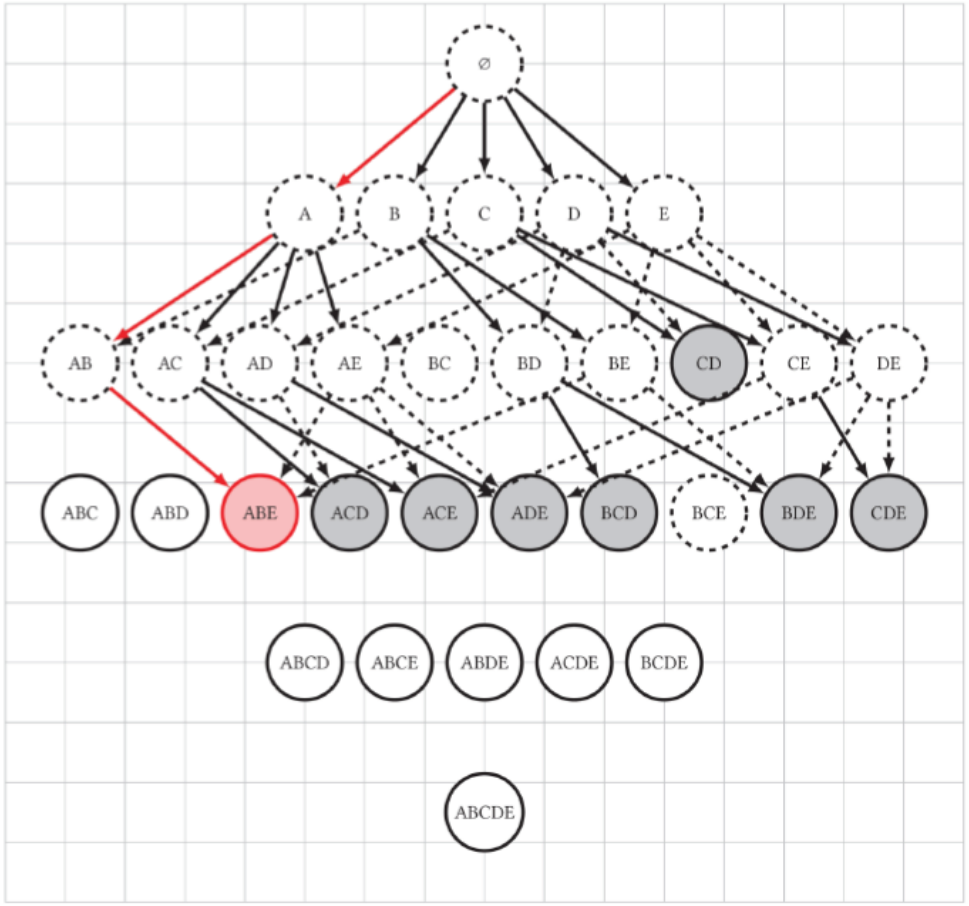
Typical-Case Optimal Algorithm

- **DAG-based representation**
  - Example for classifiers A-E
  - Each vertex corresponds to a unique subset of classifiers
  - An edge connects each vertex P with another Q extended by adding one classifier
  - An edge from P to Q is valid if:

$$TCET(P) + WCET(Q-P) + WCET(ESCAP(Q)) \leq L$$

(invalid edges removed)

  - Need only consider the incoming edge with the maximum slack time (solid arrows)
  - Optimal solution derives from the vertex with the lowest FP(S) of any vertex that meets the constraint $FN(S) \leq H$ and is reachable from the start (solid boundary and shaded)
  - Path A → AB → ABE is optimal (shown in red), i.e. classifiers A, B, and E in that order



Typical-Case Optimal Algorithm

- **Run-time solution**
  - To make best use of slack at run-time the trigger points are set as late as possible
  - The typical-case optimal solution is ((A, 0.0034, CD), (B, 0.02085, DE), (E, 0.0447, E)) for the concrete problem considered
  - Each triplet indicates the preferred classifier to run, the latest permitted start time for that classifier, and the escape set to switch to if that start time is not met

- **Offline complexity**
  - Overall complexity of the off-line part of the typical-case optimal algorithm is $O(4^n)$ dominated by the construction of the extended profile table
  - DAG-based component of the algorithm has $O(n2^n)$ complexity, since there are 2^n vertices and at most n outgoing edges per vertex

Comparison

- **Typical-case optimal vs. Static optimal vs. Clairvoyant optimal**
  - Assume ACET(S) = TCET(S) for clairvoyant algorithm

Static: *ACE* (any order):

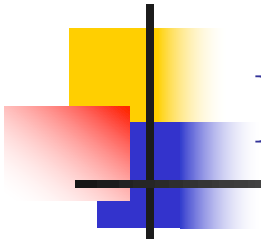
$FP(S) = 0.0458, WCET(S) = 0.047975, TCET(S) = 0.03454.$

Typical: *ABE* (specific order):

$FP(S) = 0.0308, WCET(S) = 0.054274, TCET(S) = 0.040066.$

Clairvoyant: *ABC* (any order):

$FP(S) = 0.0275, WCET(S) = 0.066529, TCET(S) = 0.048217.$



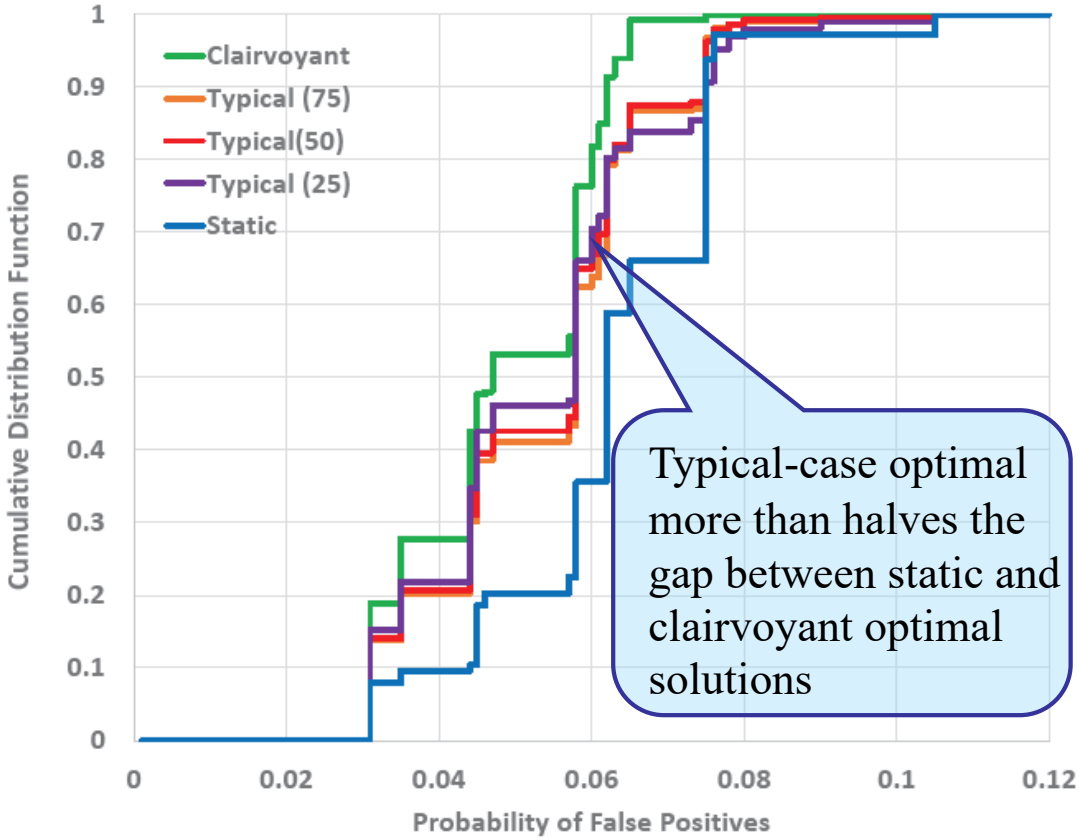
Evaluation

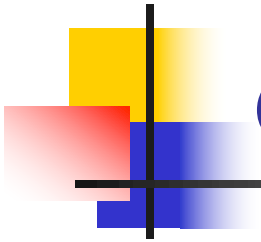
- **Extended Multi-Modal case study**
  - All seven classifiers considered so profile table has 2^7=128 rows
  - Experiment involved 1000 runs:
    - Latency constraint was randomly selected in the range [0.03333, 0.06667], i.e. 33ms to 67ms, typically achievable using three classifiers.
    - Constraint on the maximum probability of false negatives was randomly selected in the range [0.06667, 0.08333], again typically achievable using three classifiers
    - Only pairs of constraints on latency and the probability of false negatives that admitted a static solution were used
    - Actual execution times for the classifiers were selected at random from the sets of execution times obtained processing the 1800 input samples used during profiling
  - Compared solutions from:
    - Static optimal algorithm
    - Clairvoyant optimal algorithm
    - Typical-case optimal algorithm using 25-, 50-, and 75-percentiles for typical-case execution times

Evaluation

- **Expt 1:**
  - Plot shows the CDF of the probability of false positives for the chosen solutions
  - Smallest value is 0.0308 for ABE largest was 0.104 for DEFG
  - Static optimal outperformed typical-case optimal in 3.3%, 4.1% and 9.4% of cases (75-, 50-, and 25-percentile settings)
  - ***Optimizing for the typical case doesn't necessarily optimize for the worst case***

Multi-Modal: Seven classifiers

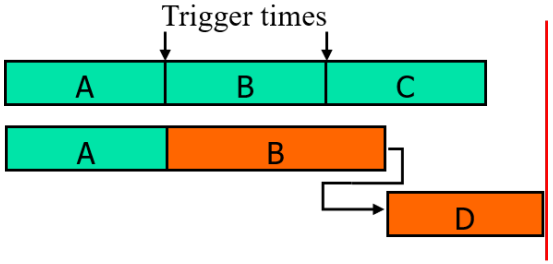


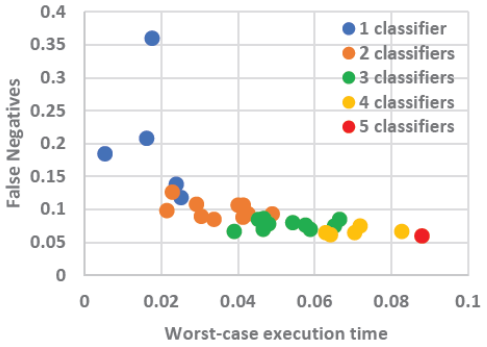


Conclusions

- **Summary**
  - This research addressed the problem of real-time classification-based machine perception, specifically the "hazard detection classifier sequencing problem"
  - Our main contribution was the derivation of optimal algorithms for the scheduling of classifiers that minimize the probability of false positives, while meeting a latency constraint and a constraint on the maximum permitted probability of false negatives
  - The classifiers can have practical attributes: arbitrary statistical dependences between their functional behaviours and variability in their execution times
  - The solutions proposed were applicable to real-world scenarios and are practical with $O(1)$ run-time overheads (up to 20 classifiers could be considered for the same problem)
  - Evaluation showed that the typical-case optimal algorithm provides a significant improvement over statically optimal solutions, more than halving the performance gap to a hypothetical clairvoyant algorithm

Discussion and Questions?





Binary	Classifiers S	$FP(S)$	$FN(S)$	$WCET(S)$	$TCET(S)$	$ESCAP(S)$
00000	∅	0.0000	1.0000	0	0	CD
00001	A	0.0075	0.1183	0.025121	0.018166	DE
00010	B	0.0042	0.1383	0.023854	0.017788	DE
00011	AB	0.0100	0.0933	0.048975	0.035954	E
00100	C	0.0200	0.3600	0.017554	0.012263	D
00101	AC	0.0250	0.0933	0.042675	0.030429	E
00110	BC	0.0242	0.1067	0.041408	0.030051	D
00111	ABC	0.0275	0.0850	0.066529	0.048217	∅
01000	D	0.0358	0.2083	0.01618	0.011878	C
01001	AD	0.0417	0.0883	0.0413	0.030044	E
01010	BD	0.0383	0.1067	0.040033	0.029666	E
01011	ABD	0.0433	0.0750	0.065154	0.047832	∅
01100	CD	0.0542	0.0850	0.033734	0.024141	∅
01101	ACD	0.0575	0.0700	0.058854	0.042307	∅
01110	BCD	0.0567	0.0767	0.057587	0.041929	∅
01111	ABCD	0.0592	0.0667	0.082708	0.060095	∅
10000	E	0.0250	0.1850	0.0053	0.004112	CD
10001	AE	0.0292	0.0900	0.030421	0.022277	D
10010	BE	0.0275	0.1083	0.029154	0.0219	D
10011	ABE	0.0308	0.0800	0.054274	0.040066	∅
10100	CE	0.0425	0.1267	0.022854	0.016374	D
10101	ACE	0.0458	0.0783	0.047975	0.03454	∅
10110	BCE	0.0450	0.0867	0.046708	0.034162	D
10111	ABCE	0.0475	0.0750	0.071828	0.052328	∅
11000	DE	0.0592	0.0983	0.021479	0.01599	C
11001	ADE	0.0617	0.0700	0.0466	0.034156	∅
11010	BDE	0.0600	0.0850	0.045333	0.033778	∅
11011	ABDE	0.0625	0.0650	0.070454	0.051944	∅
11100	CDE	0.0750	0.0667	0.039033	0.028252	∅
11101	ACDE	0.0767	0.0617	0.064154	0.046418	∅
11110	BCDE	0.0758	0.0650	0.062887	0.046041	∅
11111	ABCDE	0.0775	0.0600	0.088008	0.064206	∅

rob.davis@york.ac.uk

	A	B	C	D	E	F	G
A	1	0.931	0.725	0.815	0.860	0.717	0.687
B	0.931	1	0.721	0.832	0.872	0.723	0.6944
C	0.725	0.721	1	0.570	0.687	0.819	0.440
D	0.815	0.832	0.570	1	0.747	0.579	0.699
E	0.860	0.872	0.687	0.747	1	0.711	0.717
F	0.717	0.723	0.819	0.579	0.711	1	0.433
G	0.687	0.694	0.440	0.699	0.717	0.433	1

	A	B	C	D	E	F	G
A	1	0.031	0.036	-0.011	-0.027	-0.009	0.022
B	0.031	1	0.009	0.024	-0.040	-0.013	-0.024
C	0.036	0.009	1	0.000	0.029	-0.004	0.031
D	-0.011	0.024	0.000	1	-0.020	0.062	-0.058
E	-0.027	-0.040	0.029	-0.020	1	-0.007	0.076
F	-0.009	-0.013	-0.004	0.062	-0.007	1	0.024
G	0.022	-0.024	0.031	-0.058	0.076	0.024	1