Machine learning for vehicle platooning

DataScienceSeed, Genova, Digital Tree, 04 July 2019

Maurizio Mongelli, Marco Muselli, Andrea Scorzoni (CNR-IEIIT)
Alessandro Fermi, Enrico Ferrari (Rulex Innovation Labs)

maurizio.mongelli, marco.muselli@ieiit.cnr.it

m.muselli, a.fermi, e.ferrari@rulex-inc.com





Vehicle platooning

https://www.youtube.com/watch?v=X7vziDnNXEY&t=73s

Index of the presentation

- Vehicle platooning
  - Machine Learning (ML), why?
  - Traditional analysis (counter-intuitive example)
  - Reduce false negatives
  - Cyber security (cognitive ML and control)
- Safety
  - Safety in cyber physical system
  - Safety and AI
- Conclusions and open issues

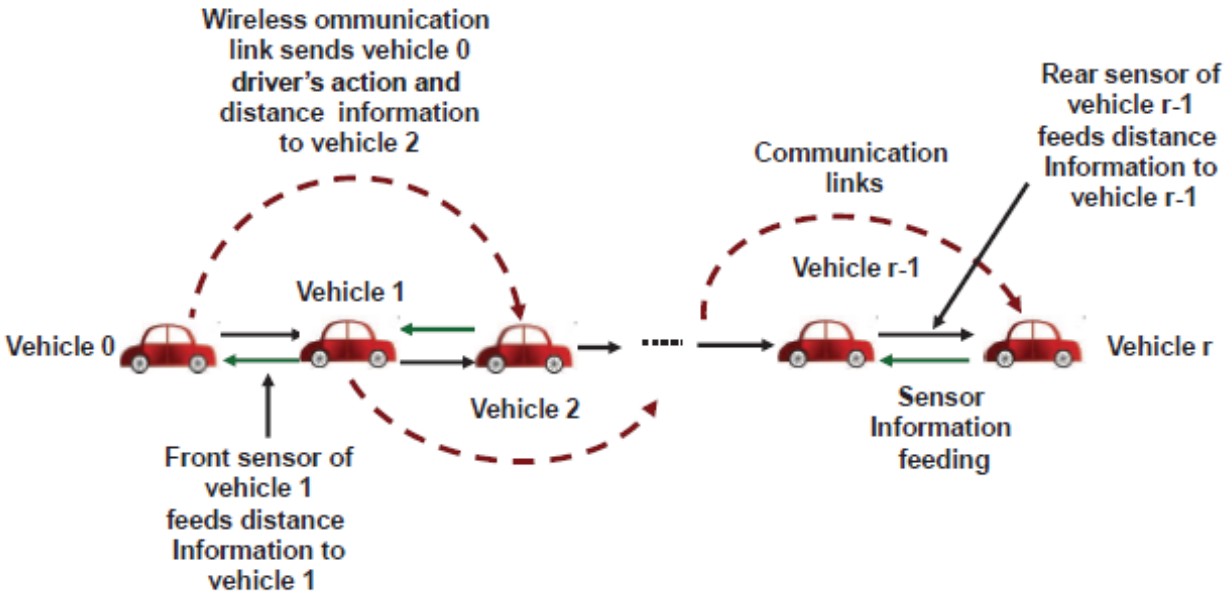
Vehicle platooning: the problem

Platooning: problem

Leading vehicle (#0) applies a braking force

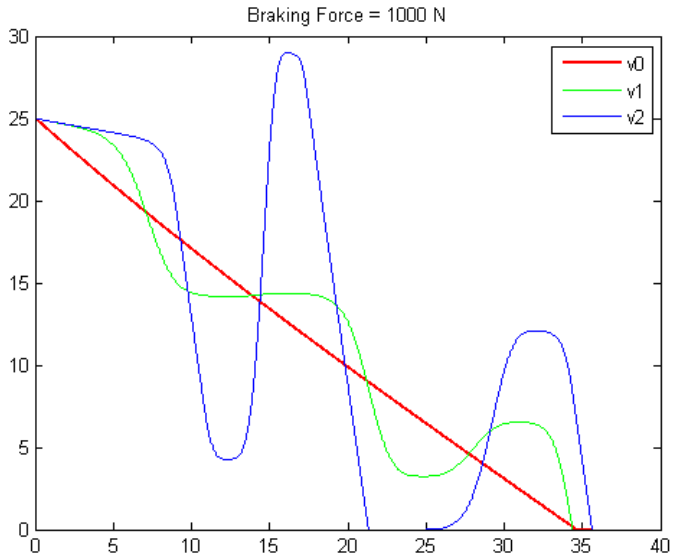
Parameters: # vehicles, initial distance, initial speed, force, weight, communication delay (control law assumed fixed)

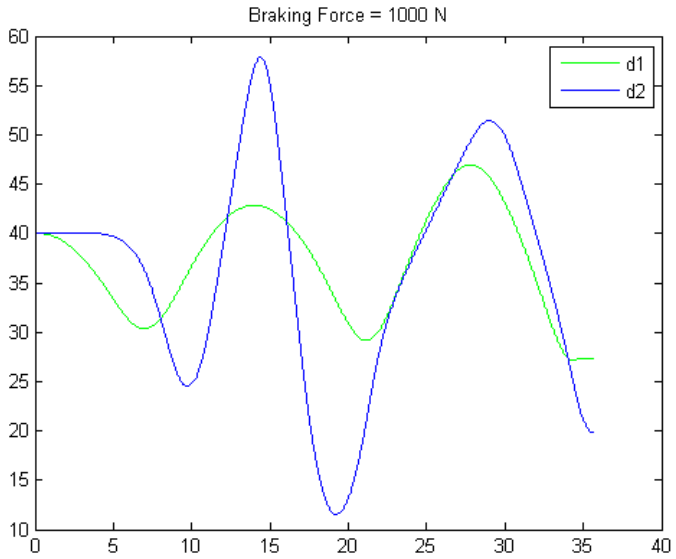
Can we predict collision?

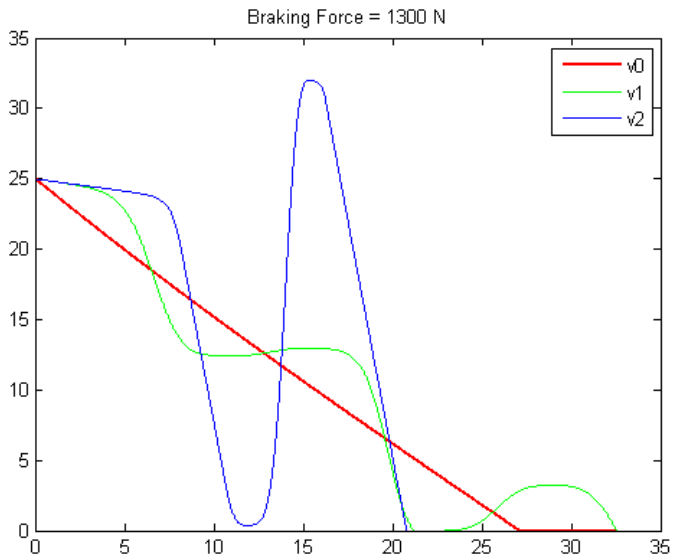


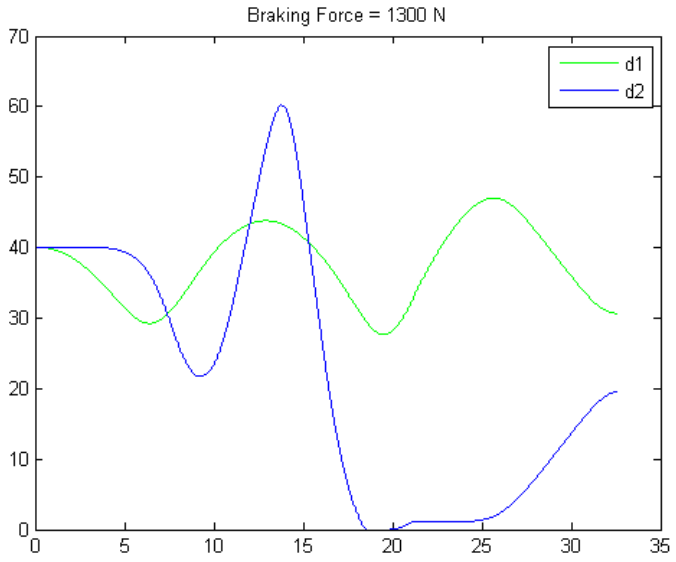
L. Xu, L. Y. Wang, G. Yin and H. Zhang, "Communication Information Structures and Contents for Enhanced Safety of Highway Vehicle Platoons," in *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4206-4220, Nov. 2014. doi: 10.1109/TVT.2014.2311384.

Platooning: example

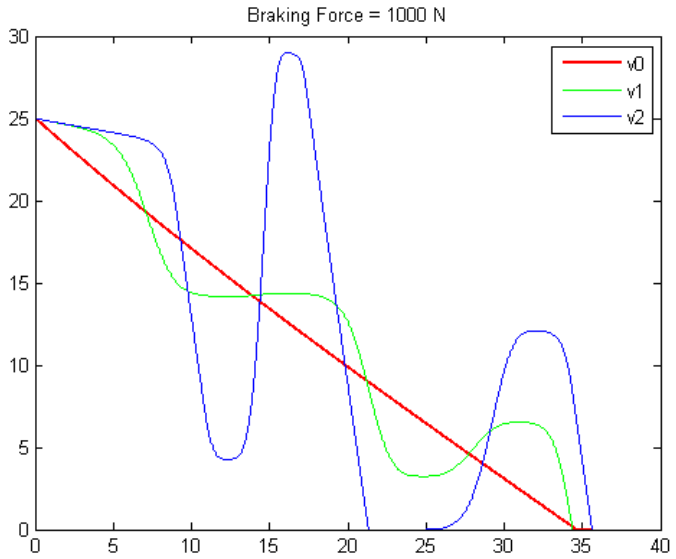


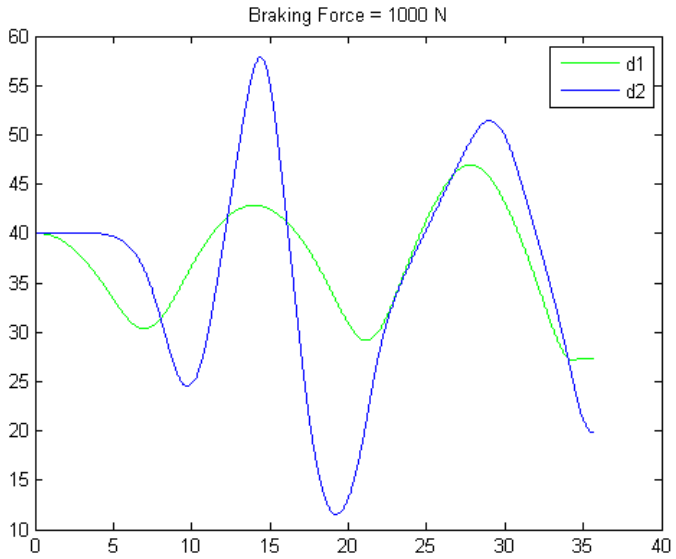


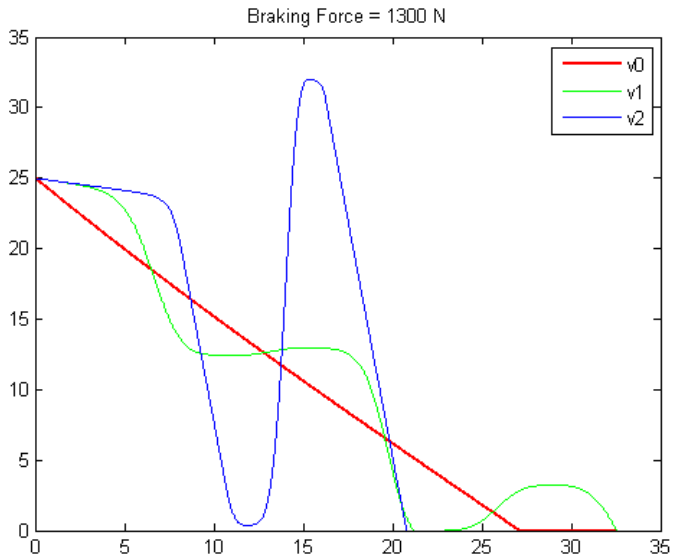


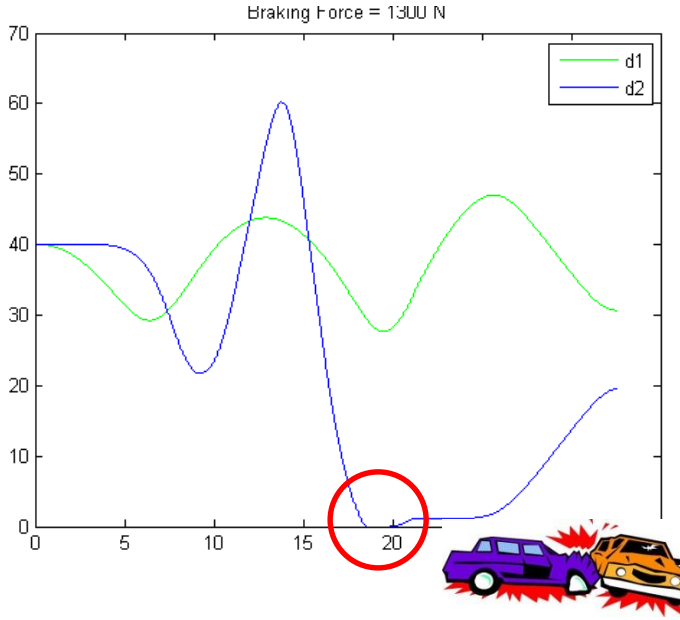


Platooning: example









Performance prediction: state of the art

Performance prediction: state of the art

A lot of control algorithms

Mathematical modeling for stability of the string of vehicles

Brute force simulation analysis

> Moreover in this scenario we have re-tuned the controller to ensure a constant and very small (5 m) bumper to bumper distance and not a constant time headway.

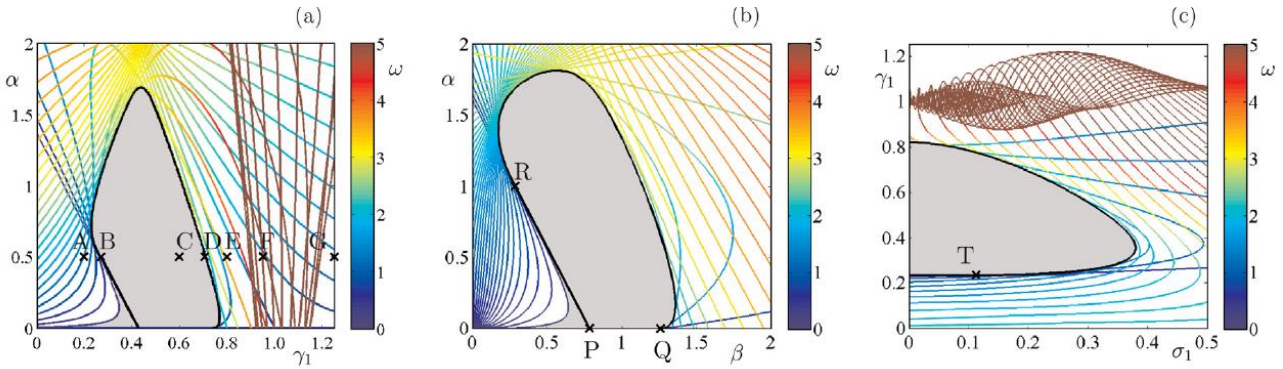


Fig. 11. (a) Stability chart in the (γ_1, α)-plane for the ring configuration using $N = 33$ vehicles and the same parameters as in Fig. 2(e). (b and c) Stability

S. Santini, A. Salvi, A. S. Valente, A. Pescapè, M. Segata and R. L. Cigno, "A consensus-based approach for platooning with inter-vehicular communications," *2015 IEEE Conference on Computer Communications (INFOCOM)*, Kowloon, 2015, pp. 1158-1166. doi: 10.1109/INFOCOM.2015.7218490.

Jin I. Ge, Gábor Orosz, Dynamics of connected vehicle systems with delayed acceleration feedback, Transportation Research Part C: Emerging Technologies, Volume 46, September 2014, Pages 46-64, ISSN 0968-090X, http://dx.doi.org/10.1016/j.trc.2014.04.014.

Machine Learning

Machine Learning

1st step: database of metrics and performance

Platooning: model

Model based on differential equations to generate sample paths of the system

$$\begin{cases} \dot{v}_0 &= \frac{1}{m_0}(F_0 - (a_0 + b_0 v_0^2)) \\ \dot{v}_1 &= \frac{1}{m_1}(F_1 - (a_1 + b_1 v_1^2)) \\ \dot{v}_2 &= \frac{1}{m_2}(F_2 - (a_2 + b_2 v_2^2)) \\ \dot{d}_1 &= v_0 - v_1 \\ \dot{d}_2 &= v_1 - v_2, \end{cases}$$

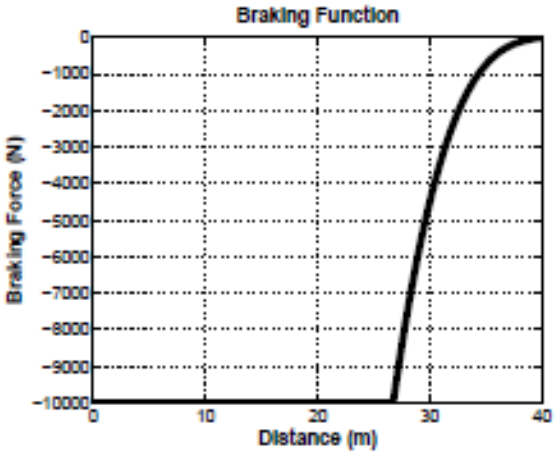
L. Xu, L. Y. Wang, G. Yin and H. Zhang, "Communication Information Structures and Contents for Enhanced Safety of Highway Vehicle Platoons," in *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4206-4220, Nov. 2014. doi: 10.1109/TVT.2014.2311384.

Platooning: model

Model based on differential equations to generate sample paths of the system

$$\begin{cases} \dot{v}_0 &= \frac{1}{m_0}(F_0 - (a_0 + b_0 v_0^2)) \\ \dot{v}_1 &= \frac{1}{m_1}(F_1 - (a_1 + b_1 v_1^2)) \\ \dot{v}_2 &= \frac{1}{m_2}(F_2 - (a_2 + b_2 v_2^2)) \\ \dot{d}_1 &= v_0 - v_1 \\ \dot{d}_2 &= v_1 - v_2, \end{cases}$$

$$\max\{k_1(d - dref) + k_2(d - dref)^3, -F_{max}\}$$



L. Xu, L. Y. Wang, G. Yin and H. Zhang, "Communication Information Structures and Contents for Enhanced Safety of Highway Vehicle Platoons," in *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4206-4220, Nov. 2014. doi: 10.1109/TVT.2014.2311384.

Platooning: model

Model based on differential equations to generate sample paths of the system

$$\begin{cases} \dot{v}_0 &= \frac{1}{m_0}(F_0 - (a_0 + b_0 v_0^2)) \\ \dot{v}_1 &= \frac{1}{m_1}(F_1 - (a_1 + b_1 v_1^2)) \\ \dot{v}_2 &= \frac{1}{m_2}(F_2 - (a_2 + b_2 v_2^2)) \\ \dot{d}_1 &= v_0 - v_1 \\ \dot{d}_2 &= v_1 - v_2, \end{cases}$$

- Each vehicle communicates with the previous one only (no multiple coverage of vehicles by the communication channel, for now).
- Each vehicle sends current position and speed.
- Braking force applied in each vehicle on the basis of received information (speed not used by control law, for now).

Platooning: model

Random sampling of system conditions as follows: …

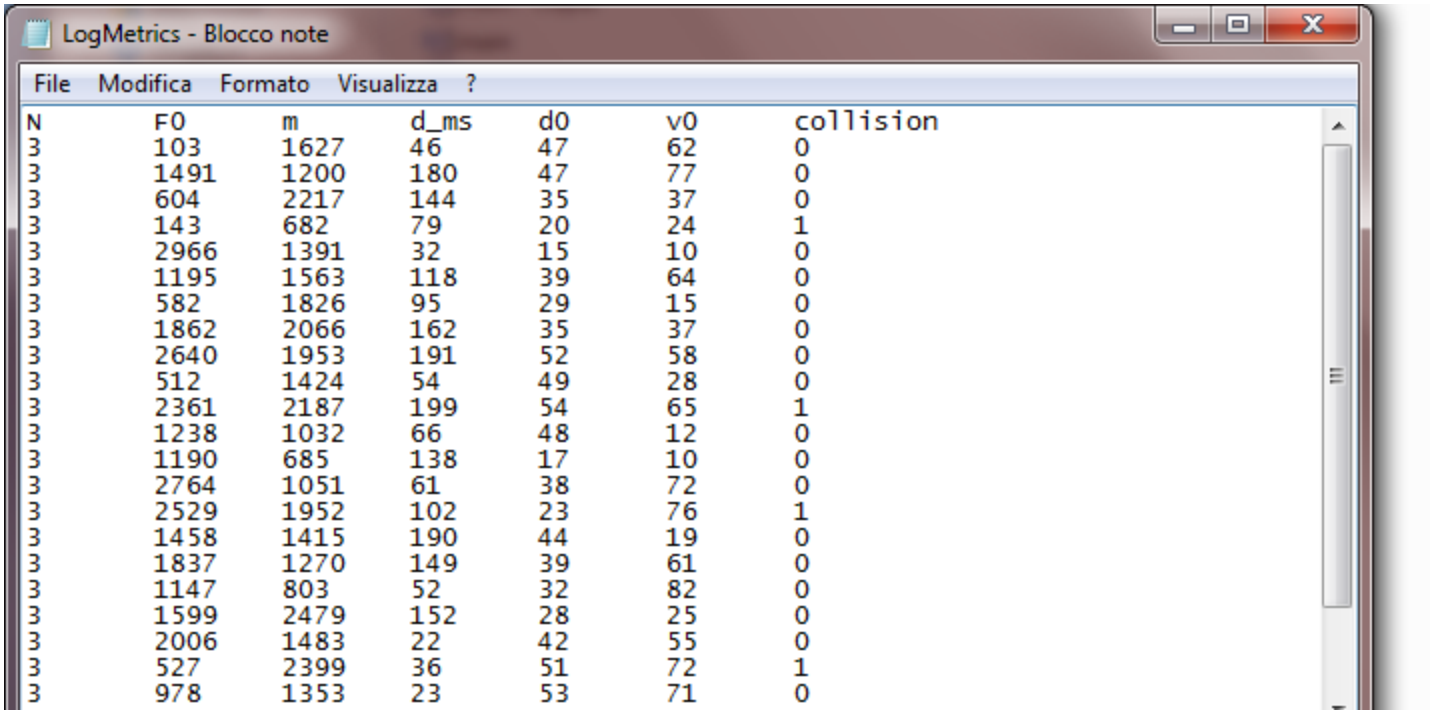
$$\begin{cases} \dot{v}_0 &= \frac{1}{m_0}(F_0 - (a_0 + b_0 v_0^2)) \\ \dot{v}_1 &= \frac{1}{m_1}(F_1 - (a_1 + b_1 v_1^2)) \\ \dot{v}_2 &= \frac{1}{m_2}(F_2 - (a_2 + b_2 v_2^2)) \\ \dot{d}_1 &= v_0 - v_1 \\ \dot{d}_2 &= v_1 - v_2, \end{cases}$$

… # vehicles = 3, initial distance in [15, 55] m, initial speed in [10, 90] km/h, force in [100, 3000] N, vehicle weight in [500, 2500] Kg, communication delay in [10, 200] ms (fixed*, for now).

* Probabilistic models applicable (->runs within the main loop to cope with randomness).

Platooning: database of performance

At the end of each run (corresponding to 1 sample of system parameters) we register if there was a collision or not.



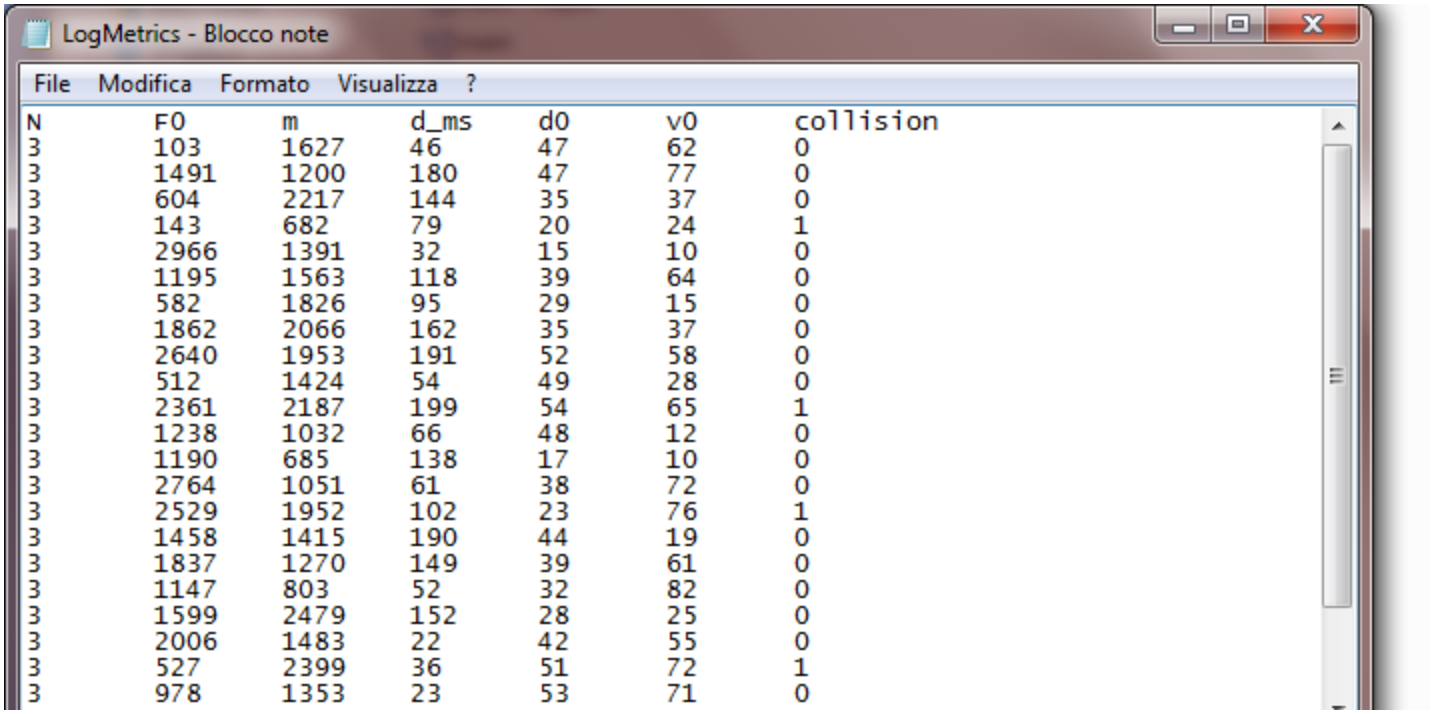
LogMetrics - Blocco note

File Modifica Formato Visualizza ?

N	F0	m	d_ms	d0	v0	collision
3	103	1627	46	47	62	0
3	1491	1200	180	47	77	0
3	604	2217	144	35	37	0
3	143	682	79	20	24	1
3	2966	1391	32	15	10	0
3	1195	1563	118	39	64	0
3	582	1826	95	29	15	0
3	1862	2066	162	35	37	0
3	2640	1953	191	52	58	0
3	512	1424	54	49	28	0
3	2361	2187	199	54	65	1
3	1238	1032	66	48	12	0
3	1190	685	138	17	10	0
3	2764	1051	61	38	72	0
3	2529	1952	102	23	76	1
3	1458	1415	190	44	19	0
3	1837	1270	149	39	61	0
3	1147	803	52	32	82	0
3	1599	2479	152	28	25	0
3	2006	1483	22	42	55	0
3	527	2399	36	51	72	1
3	978	1353	23	53	71	0

Platooning: database of performance

12000 extractions of system parameters (=rows in the db).
6 hours of simulation on Intel 2.4Ghz i7 processor.


LogMetrics - Blocco note

File Modifica Formato Visualizza ?

N	F0	m	d_ms	d0	v0	collision
3	103	1627	46	47	62	0
3	1491	1200	180	47	77	0
3	604	2217	144	35	37	0
3	143	682	79	20	24	1
3	2966	1391	32	15	10	0
3	1195	1563	118	39	64	0
3	582	1826	95	29	15	0
3	1862	2066	162	35	37	0
3	2640	1953	191	52	58	0
3	512	1424	54	49	28	0
3	2361	2187	199	54	65	1
3	1238	1032	66	48	12	0
3	1190	685	138	17	10	0
3	2764	1051	61	38	72	0
3	2529	1952	102	23	76	1
3	1458	1415	190	44	19	0
3	1837	1270	149	39	61	0
3	1147	803	52	32	82	0
3	1599	2479	152	28	25	0
3	2006	1483	22	42	55	0
3	527	2399	36	51	72	1
3	978	1353	23	53	71	0

Machine Learning

Machine Learning
2nd step: knowledge extraction

Machine Learning

2nd step: knowledge extraction

Is the problem difficult?

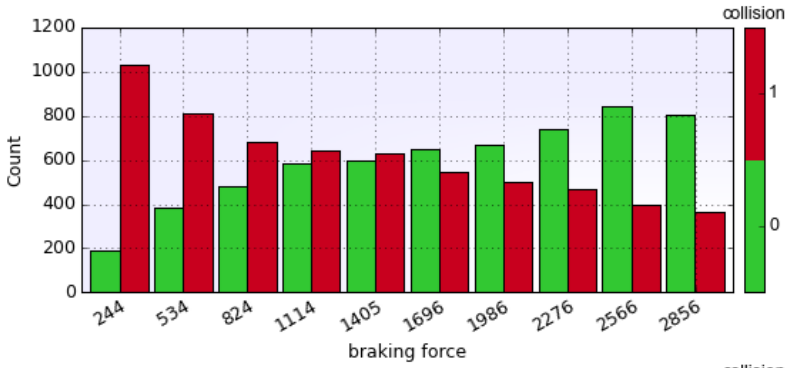
Is this problem difficult?

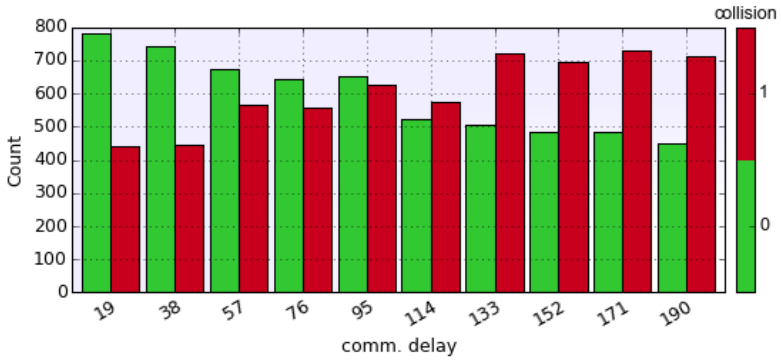
$$\begin{cases} \dot{v}_0 &= \frac{1}{m_0}(F_0 - (a_0 + b_0 v_0^2)) \\ \dot{v}_1 &= \frac{1}{m_1}(F_1 - (a_1 + b_1 v_1^2)) \\ \dot{v}_2 &= \frac{1}{m_2}(F_2 - (a_2 + b_2 v_2^2)) \\ \dot{d}_1 &= v_0 - v_1 \\ \dot{d}_2 &= v_1 - v_2, \end{cases}$$

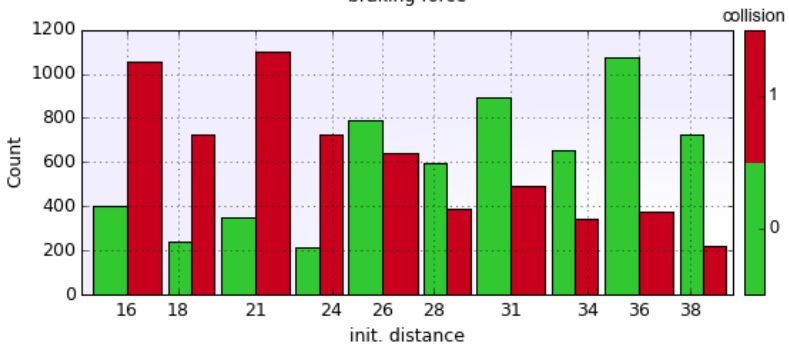
vehicles = 3,
initial distance in [15, 55] m,
initial speed in [10, 90] km/h,
force in [100, 3000] N,
vehicle weight in [500, 2500] Kg,
communication delay in [10, 200] ms .

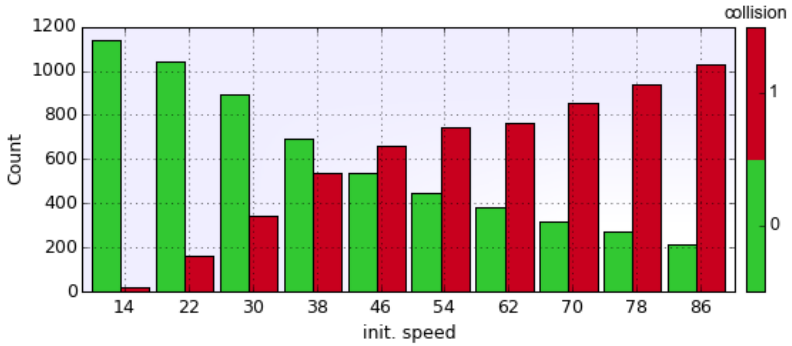
Univariate analysis by histograms

not easy to understand!





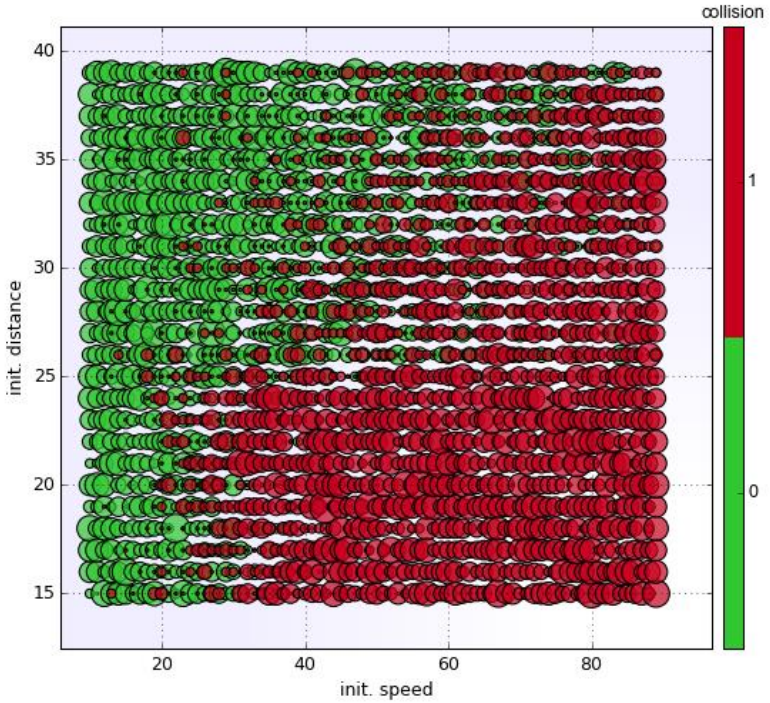


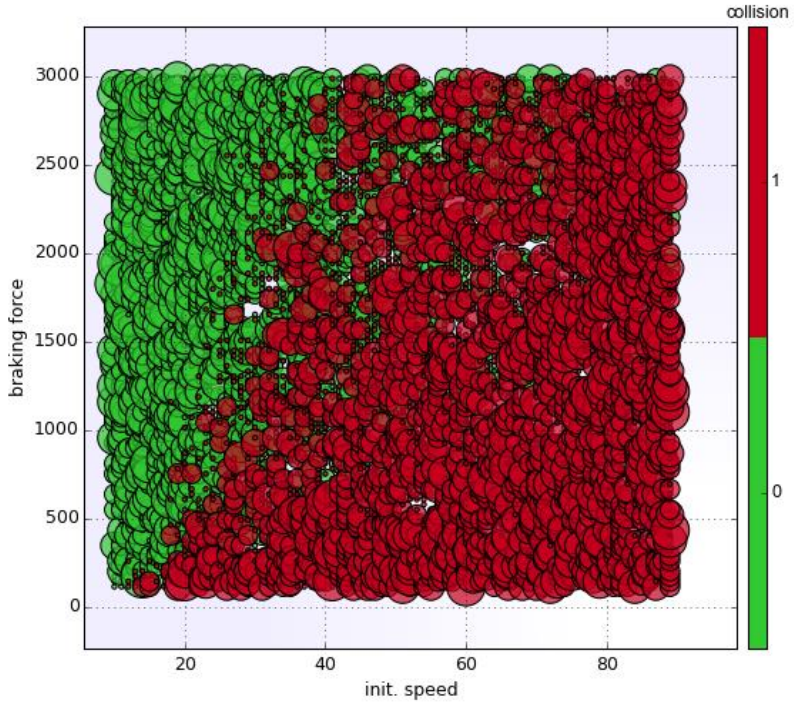


Each single variable experiences collision and no collision

Bivariate analysis by scatter plots

not easy to understand!





Does a clear boundary between collision and no collision exist in each bi-dimensional space of the features?

Machine Learning

2nd step: knowledge extraction

Logic Learning Machine in the Rulex platform:

If-then rules with accuracy

Neural network models

Data

$$
\begin{aligned}
f(\mathbf{x}) \;=\;& 0.293\,tanh(0.113\,x_0 + 0.337\,x_1 - 0.329\,x_2 + 0.251\,x_3 - 0.288\,x_4 - 0.297\,x_5 + 0.436\,x_6 + \\
&+\; 0.166\,x_7 - 0.184\,x_8 + 0.219\,x_9 + 0.483\,x_{10} - 0.222\,x_{11} + 0.173\,x_{12} + 0.012\,x_{13} + \\
&+\; 0.352\,x_{14} + 0.259\,x_{15} + 0.176\,x_{16} + 0.345\,x_{17} + 0.314\,x_{18} + 0.177\,x_{19} - 0.329\,x_{20} + \\
&-\; 0.363\,x_{21} + 0.216\,x_{22} - 0.148\,x_{23} - 0.043\,x_{24} + 0.316\,x_{25} - 0.068\,x_{26} - 0.421\,x_{27(0)} + \\
&+\; 0.15\,x_{27(1)} - 0.289\,x_{27(2)} - 0.241\,x_{28} + 0.16\,x_{29} + 0.199\,x_{30} - 0.111\,x_{31} - 0.164\,x_{32} + \\
&+\; 0.117\,x_{33} + 0.466\,x_{34} + 0.457\,x_{35} + 0.133\,x_{36} + 0.331\,x_{37} - 0.362\,x_{38} - 0.43\,x_{39} + \\
&-\; 0.491\,x_{40} - 0.155\,x_{41} + 0.371\,x_{42} - 0.05\,x_{43} - 0.177\,x_{44} - 0.044\,x_{45} + 0.225\,x_{46} + \\
&+\; 0.328\,x_{47} - 0.118\,x_{48} - 0.3) + \\
&-\; 1.934\,tanh(-0.233\,x_0 + 0.174\,x_1 - 0.252\,x_2 - 0.501\,x_3 - 0.125\,x_4 + 0.311\,x_5 - 0.573\,x_6 + \\
&-\; 0.299\,x_7 + 1.123\,x_8 + 0.318\,x_9 - 1.169\,x_{10} + 0.105\,x_{11} - 0.429\,x_{12} - 0.075\,x_{13} + \\
&-\; 0.143\,x_{14} + 0.146\,x_{15} - 0.531\,x_{16} + 0.077\,x_{17} - 0.133\,x_{18} - 0.122\,x_{19} + 0.162\,x_{20} + \\
&-\; 0.08\,x_{21} - 0.496\,x_{22} - 0.21\,x_{23} - 0.113\,x_{24} + 0.485\,x_{25} + 0.575\,x_{26} - 0.126\,x_{27(0)} + \\
&+\; 0.135\,x_{27(1)} + 0.022\,x_{27(2)} - 0.352\,x_{28} - 0.693\,x_{29} + 0.379\,x_{30} + 0.409\,x_{31} - 0.109\,x_{32} + \\
&+\; 0.228\,x_{33} + 0.292\,x_{34} + 0.161\,x_{35} - 0.086\,x_{36} - 0.3\,x_{37} - 0.089\,x_{38} + 0.163\,x_{39} + \\
&-\; 0.074\,x_{40} + 0.31\,x_{41} - 0.849\,x_{42} + 0.14\,x_{43} + 0.754\,x_{44} + 0.291\,x_{45} - 0.533\,x_{46} + 0.273\,x_{47} + \\
&-\; 0.285\,x_{48} - 0.286) + 0.252
\end{aligned}
$$

Rulex platform: intelligible rules

```
char *ApplyRules(int 'braking force', int 'weight',
                 int 'comm. delay', int 'init. distance', int 'init. speed') {

  if (('init. distance' <= 24) &&
     ('init. speed' > 30)) return "collision";

  if (('braking force' > 500) &&
     ('init. speed' <= 30)) return "no collision";

  if (('braking force' <= 1345) &&
     ('init. distance' <= 33) && ('init. speed' >35)) return "collision";

  [...]

}
```

A model made by boolean rules was built in Rulex by reading the database and applying the Logic Learning Machine algorithm (2' of computation, plug&play without tuning the algorithm)

Confusion matrix

		Forecast		
		0	1	Total
Output	0	**5024 (84.6503791...**	911 (15.3496208930...	5935 (49.458333333...
	1	577 (9.5136026381%)	**5488 (90.4863973...**	6065 (50.541666666...
	Total	5601 (46.675000000...	6399 (53.325000000...	**12000 (100%)**

		Forecast	
		0	1
Output	0		
	1		

84% True Negatives
(no collision correctly predicted).

90% True Positives (collisions correctly predicted).

Confusion matrix

		Forecast		
		0	1	Total
Output	0	**5024 (84.6503791...**	911 (15.3496208930...	5935 (49.458333333...
	1	577 (9.5136026381%)	**5488 (90.4863973...**	6065 (50.541666666...
	Total	5601 (46.675000000...	6399 (53.325000000...	**12000 (100%)**

		Forecast	
		0	1
Output	0		
	1		

9% of false negatives (FNs) (collisions not correctly predicted)

A further elaboration on how to characterize FNs is needed

Feature ranking

Rule *r* **if** <*premise*> **then** <*consequence*>

N	Disease (Output)	Condition 1	Condition 2	Covering
1	No	Age $\leq$ 65	Marker > 10.60 units	80%
2	No	Male Gender	Marker $\leq$ 29.40 units	30%

Importance of a condition *c*: error variation with and without *c*

$$\Delta E(c) = E(r') - E(r)$$

Relevance: error variation and covering *C(r)*

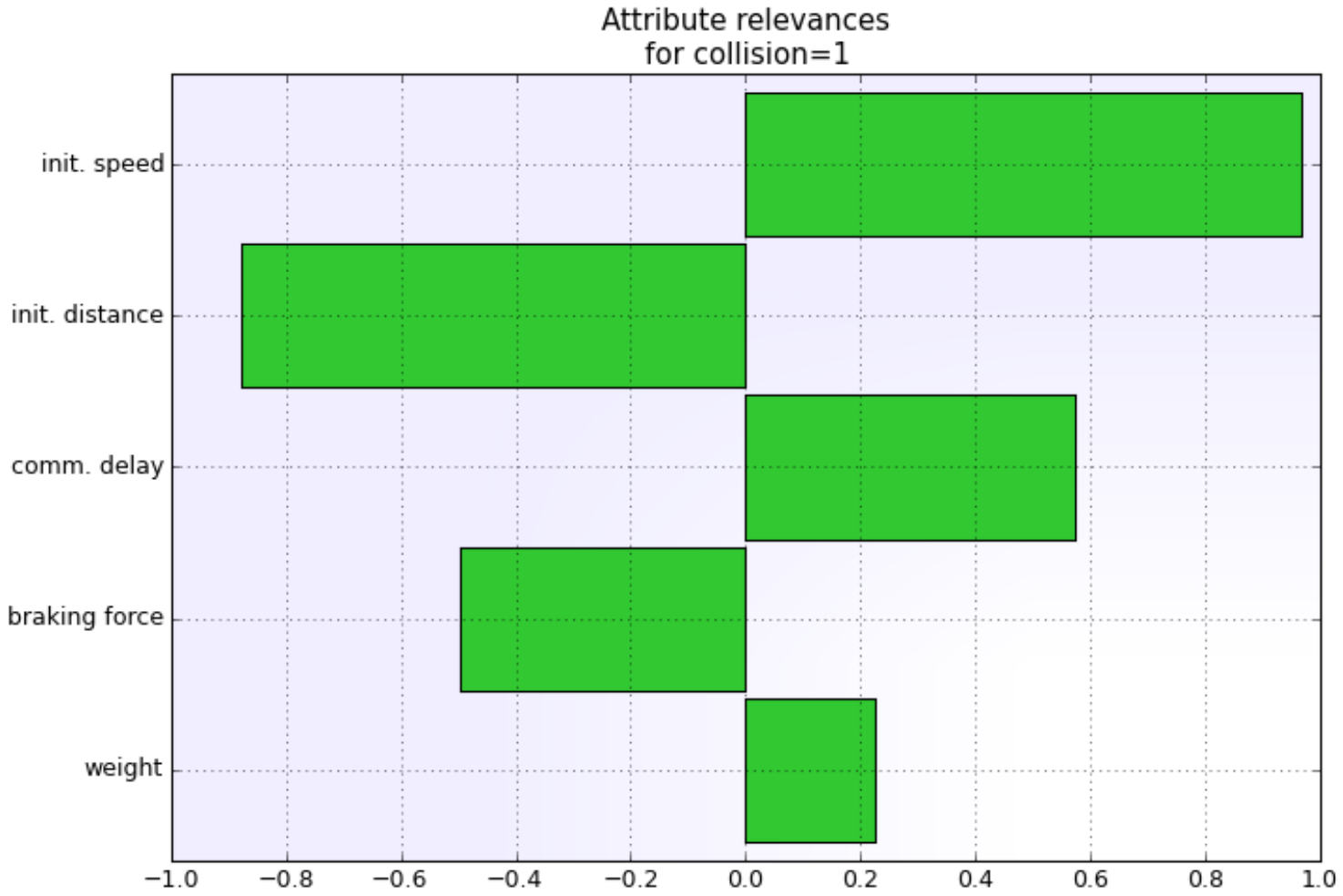
$$R(c) = \Delta E(c) C(r)$$

Relevance R_v of feature x_j

$$R_v(x_j) = 1 - \prod_k \left(1 - R(c_{kl})\right)$$

Rule of thumb: R_v<5%: marginal contribution; *C(r)*<2%: outliers

Feature ranking



Increasing initial speed has the highest relevance on collisions.
The opposite holds true for the initial distance.

Confusion matrixes of 2 models: with and without delay

		Forecast	
		0	1
Output	0	**5024 (84.6503791...**	911 (15.3496208930...
	1	577 (9.5136026381%)	**5488 (90.4863973...**
	Total	5601 (46.675000000...	6399 (53.325000000...

		Forecast		
		0	1	Total
Output	0	**5021 (84.5998315...**	914 (15.4001684920...	5935 (49.458333333...
	1	862 (14.2126957955...	**5203 (85.7873042...**	6065 (50.541666666...
	Total	5883 (49.025000000...	6117 (50.975000000...	**12000 (100%)**

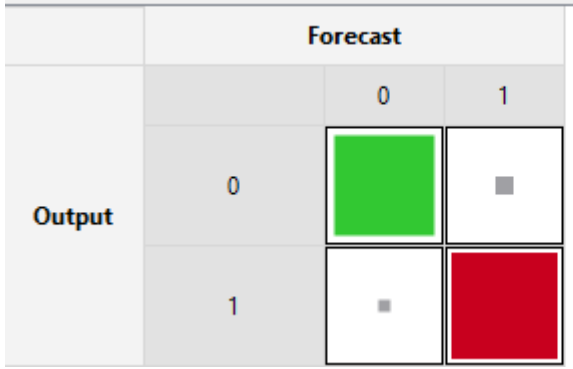
		Forecast	
		0	1
Output	0		
	1		

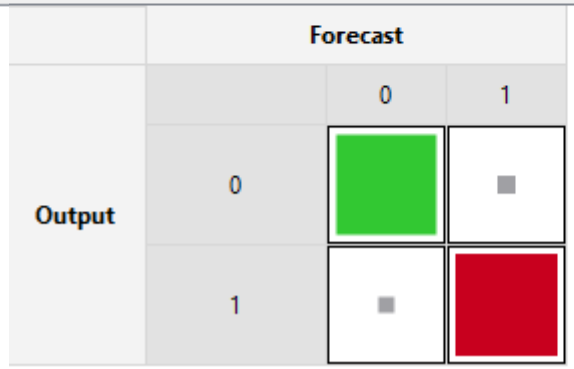
		Forecast	
		0	1
Output	0		
	1		

Confusion matrixes of 2 models: with and without delay

		Forecast	
		0	1
Output	0	**5024 (84.6503791...**	911 (15.3496208930...
	1	577 (9.5136026381%)	**5488 (90.4863973...**
	Total	5601 (46.675000000...	6399 (53.325000000...

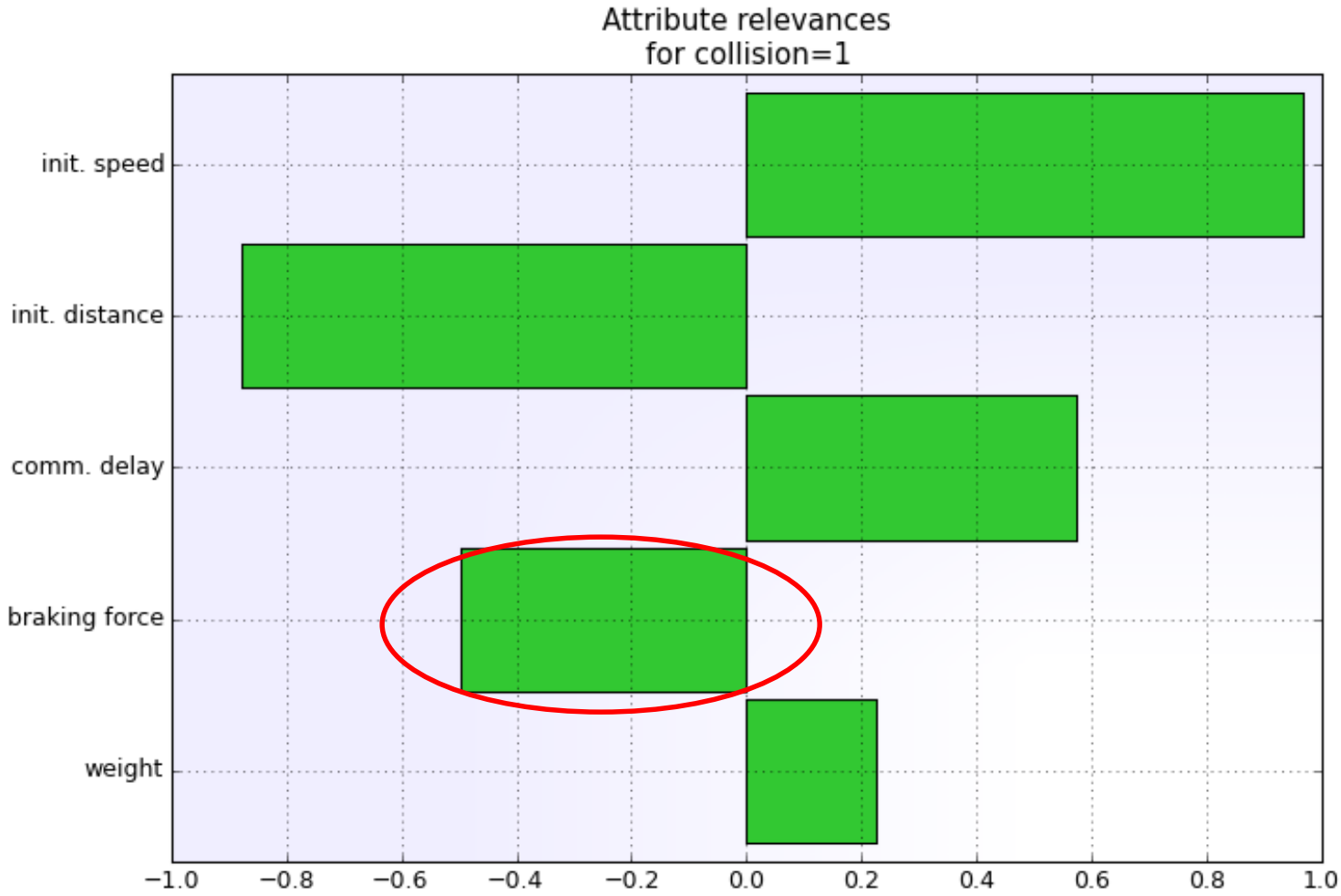
		Forecast		
		0	1	Total
Output	0	**5021 (84.5998315...**	914 (15.4001684920...	5935 (49.458333333...
	1	862 (14.2126957955...	**5203 (85.7873042...**	6065 (50.541666666...
	Total	5883 (49.025000000...	6117 (50.975000000...	**12000 (100%)**





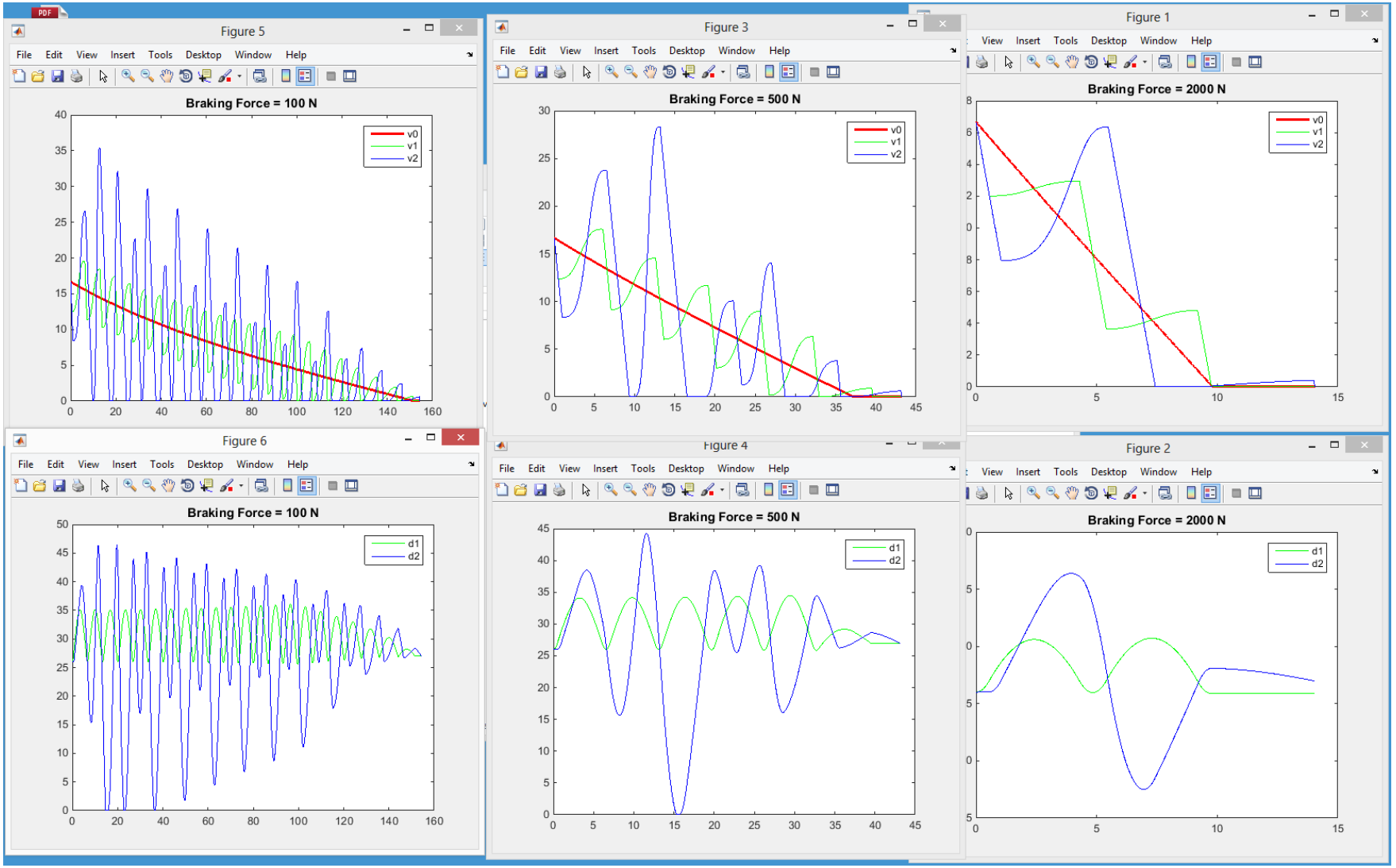
Information on delay is not crucial!

Feature ranking



Decreasing braking force -> more collisions, why?

Rationale of decreasing braking force -> more collisions



FNR=0%

Procedure for safety rule extraction FNR=0%

LLM with 0% of error in rule generation.
DT…
Refinement with cross validation as follows.

Data set $\aleph$ divided into κ portions $\aleph_\kappa$.
$\aleph_{1\kappa}$ includes only unsafe points from $\aleph_\kappa$

repeat κ times
  repeat
    1. Train$(\aleph_\kappa + \aleph_{1\kappa})$
    2. $R_{0\kappa}$ subset of 'safe' rules
    3. Manual inspection $R_{0\kappa} \rightarrow R'_{0\kappa}$
    4. Apply$(R'_{0\kappa}, \aleph_\kappa)$
  until $\aleph_\kappa$ contains safe points only

Choose most stringent conditions from $R'_{0\kappa}$.

System setting and manual calibration

plexe.car2x.org simulator
Simulated scenario

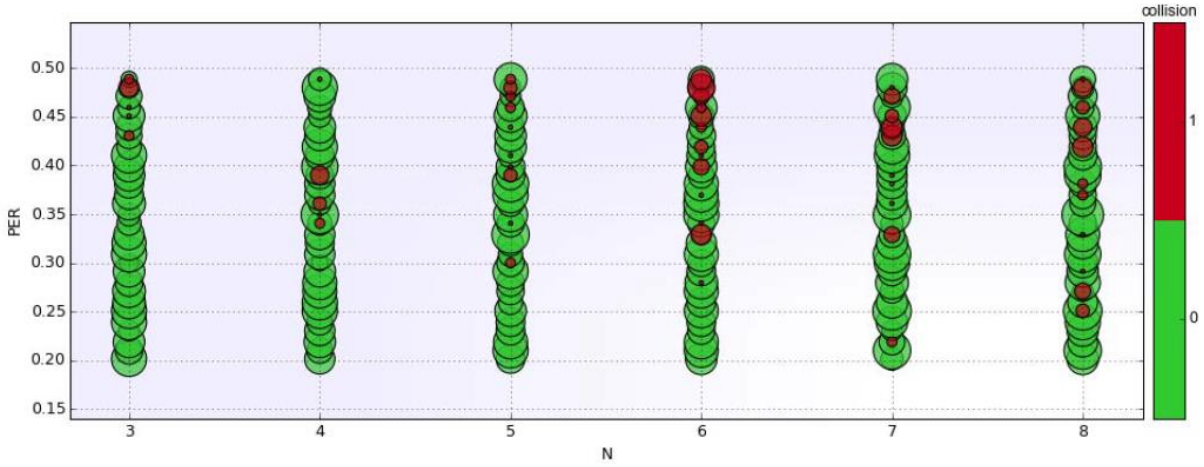
$$N \in [3, 8],\ F_0 \in [-8, -1] \cdot 10^3 \text{ N},\ PER \in [0.2, 0.5],$$
$$d(0) \in [4, 9] \text{ m},\ v(0) \in [10, 90] \text{ Km/h}.$$

System setting and manual calibration

plexe.car2x.org simulator
Simulated scenario

$$N \in [3, 8],\ F_0 \in [-8, -1] \cdot 10^3 \text{ N},\ PER \in [0.2, 0.5],$$
$$d(0) \in [4, 9] \text{ m},\ v(0) \in [10, 90] \text{ Km/h}.$$

Dataset with respect to scatter plot of *N-PER*



manual calibration:

if $((N = 6) \wedge (PER < 0.253))$ **then safe**;
if $((N = 5) \wedge (PER < 0.258))$ **then safe**;
if $((N = 4) \wedge (PER < 0.325))$ **then safe**;
if $((N = 3) \wedge (PER < 0.42))$ **then safe**;

Intelligible analytics: Logic Learning Machine (LLM) & Decision Tree (DT)

Objective 1: safety rules with 0% of false negative rate (FNR)
Objective 2: finding largest ranges of system parameters

LLM:

if $((PER \leq 0.325) \wedge (N \leq 7) \wedge (F_0 \geq -8) \wedge (d(0) \geq 4.2385))$ $(C = 30\%, E = 0\%)$

$\vee$ (**if** $(F_0 \geq -8) \wedge (d(0) \geq 4.69) \wedge (v(0) \leq 37))$ $(C = 27\%, E = 0\%)$

$\vee$ (**if** $(F_0 \geq -7) \wedge (PER \leq 0.445) \wedge (v(0) \leq 41))$ $(C = 26\%, E = 0\%)$

$\vee$ (**if** $(F_0 \geq -8) \wedge (PER \leq 0.405) \wedge (d(0) \geq 5.5055) \wedge (v(0) \leq 53))$ $(C = 26\%, E = 0\%)$

$\vee$ (**if** $(v(0) \leq 28))$ $(C = 25\%, E = 0\%)$

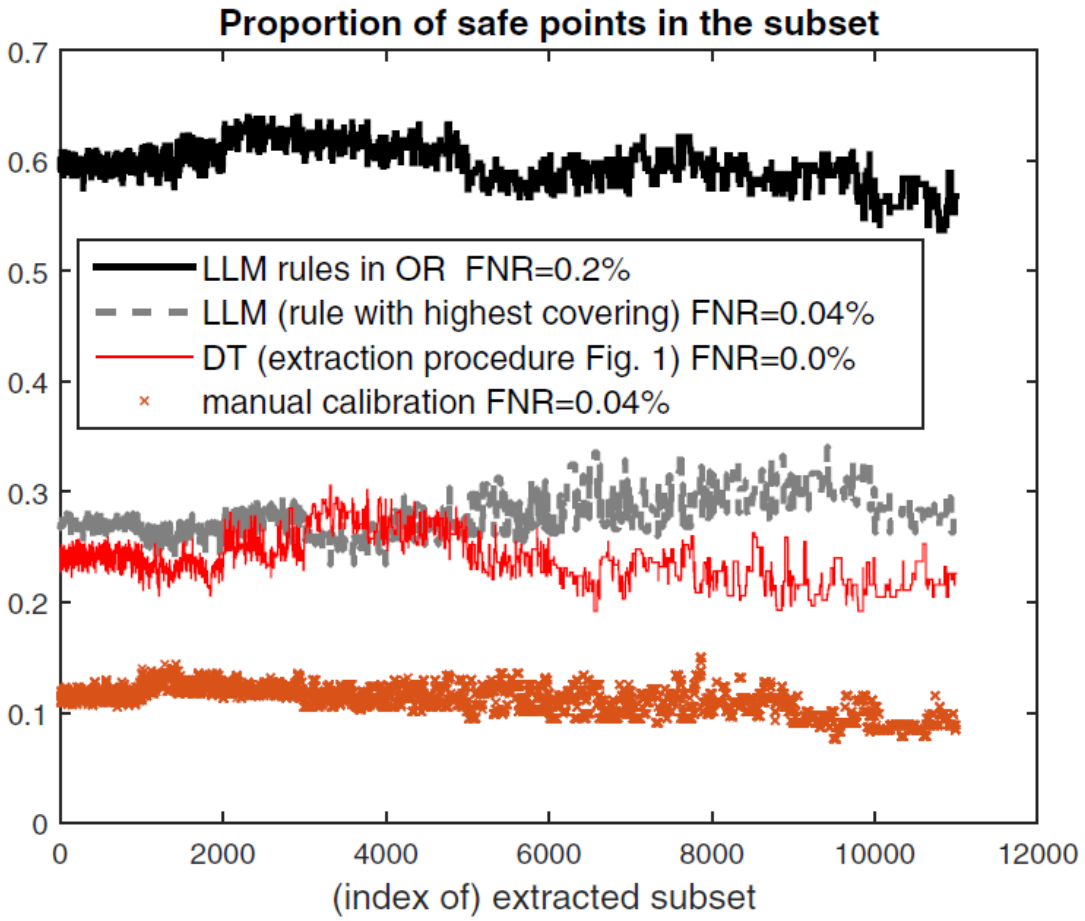
then **safe**

DT:

if $(v(0) \leq 28)$ **then** **safe** $C = 59\%, E = 0\%;$

Results: Size of safety regions and FNR

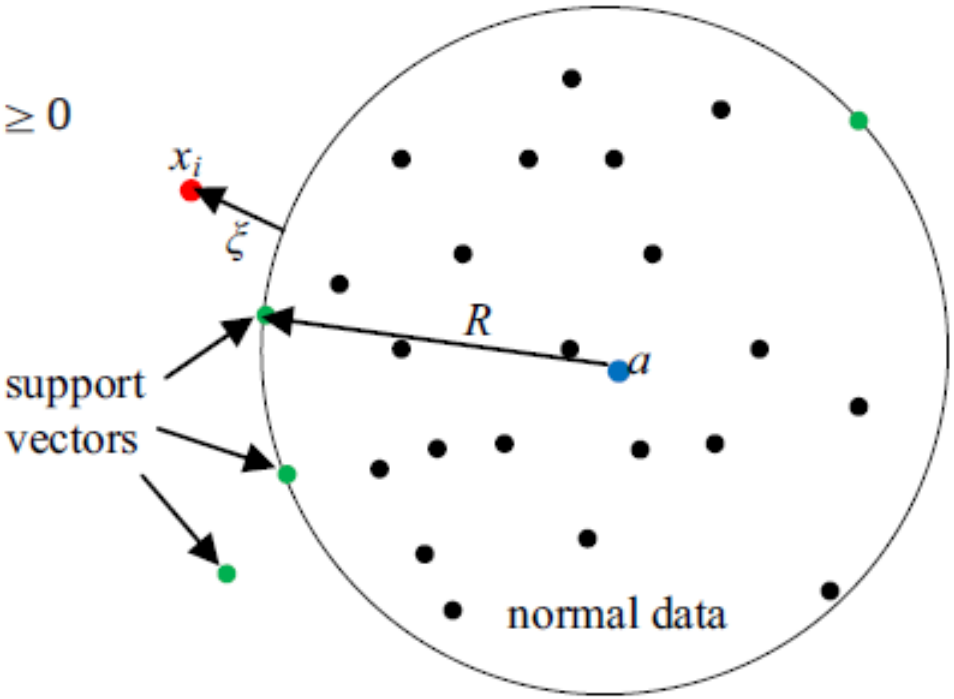
Evidence: Up to 60% of points are safe with 0.2% FNR.
Open issues: optimal solution? Comparison with black-box.



Black-box approaches?

Support Vector Data Description (SVDD)

$$\arg\min(R^2 + C\sum\xi_i),$$
$$s.t.\ \ ||x_i - a||^2 \le R^2 + \xi_i, \xi_i \ge 0$$



A hypersphere surrounding the normal dataset.

D.M.J. Tax,R.P.W.Duin,Support vector data description, Mach.Learn.54(1) (2004)45–66.
D.M.J. Tax,R.P.W.Duin,Support vector domain description,Pattern Recoginit. Lett. 20(11–13)(1999)1191–1199.
Xuemei Ding, Yuhua Li, Ammar Belatreche, Liam P. Maguire, An experimental evaluation of novelty detection methods, Neurocomputing, Volume 135, 5 July 2014, Pages 313-327, ISSN 0925-2312.

Conclusions

- Machine learning was able to cope with a non-trivial example:
  - Overlapping collision/no collision on univariate and bivariate analysis
  - Decreasing braking force -> more collisions

Conclusions and open issues

- What we have: intelligible algorithms for data analytics of platooning.

- What we are doing:
  - refinement of the models
    - a model of false negatives?
    - understanding the impact of the features
    - discrete event simulation (driven by diff. eqs.) for delay models (e.g., delay=f(distance))
  - Interaction with pilot V2I

- Future work: rule-based streaming analytics.

Cybersecurity

Packet falsification

Packet falsification consists in manipulation of the acceleration field of IEEE 802.11p, i.e., sending unreal indications to follower vehicle (whenever vehicle decelerates, the malicious packet is as if vehicle accelerates and vice versa).

S. Ucar, S. C. Ergen, and O. Ozkasap, “Security vulnerabilities of ieee 802.11p and visible light communication based platoon,” in 2016 IEEE Vehicular Networking Conference (VNC), Dec 2016, pp. 1–4.

Packet falsification

Packet falsification consists in manipulation of the acceleration field of IEEE 802.11p, i.e., sending unreal indications to follower vehicle (whenever vehicle decelerates, the malicious packet is as if vehicle accelerates and vice versa).

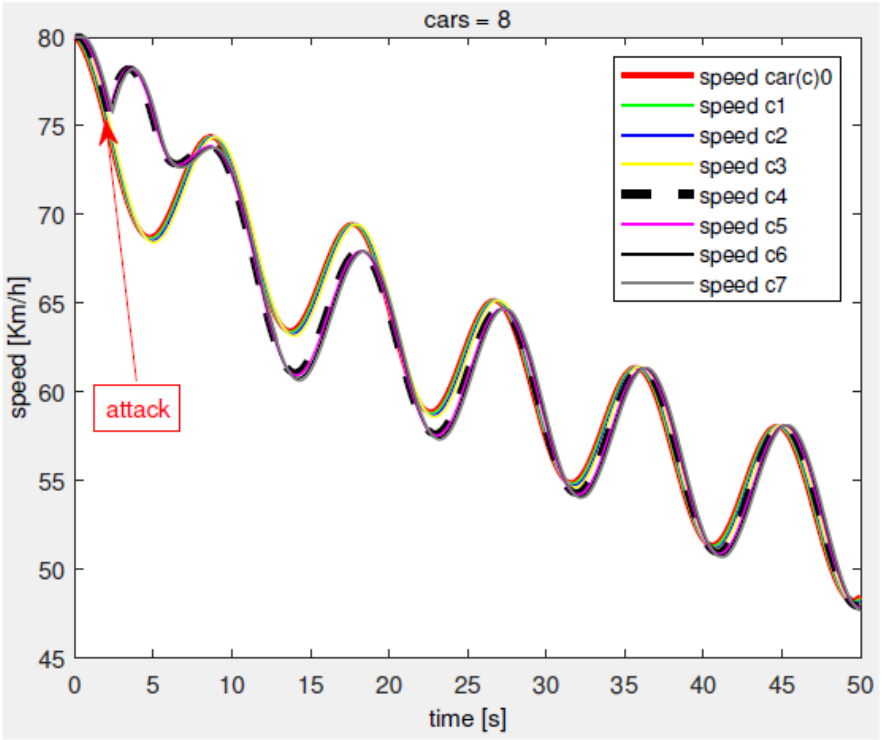


Fig. 1. Speed evolution with attack (at time 2 s).

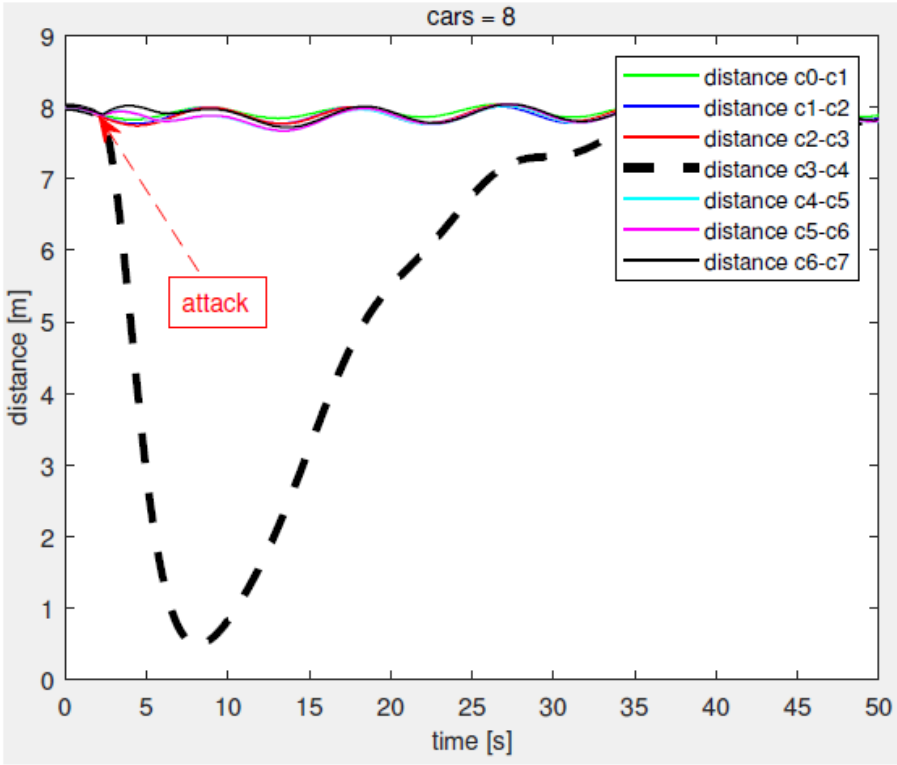


Fig. 2. Distance evolution with attack (at time 2 s).

Attack at t'=2 s. Duration of the attack D=3 s.

Approaching the problem

We approach the problem through the above methodology.

$$\boldsymbol{I_{sys}} = [N, \boldsymbol{x}(0), \boldsymbol{m}, \boldsymbol{F}, D] \;\; \boldsymbol{F} = [\bar{F_0}, \bar{b_F}] \;\; F = F_0 \cdot sin(b_F \cdot t) \;\; \boldsymbol{x}(0) = [\boldsymbol{d}(0), \boldsymbol{v}(0)]$$

Intuition

Let's have a look at integrals of differences of speeds and distances

$$\boldsymbol{I_{sys}} = [N, \boldsymbol{x}(0), \boldsymbol{m}, \boldsymbol{F}, D] \quad \boldsymbol{F} = [\bar{F_0}, b_F] \quad F = F_0 \cdot sin(b_F \cdot t) \quad \boldsymbol{x}(0) = [\boldsymbol{d}(0), \boldsymbol{v}(0)]$$

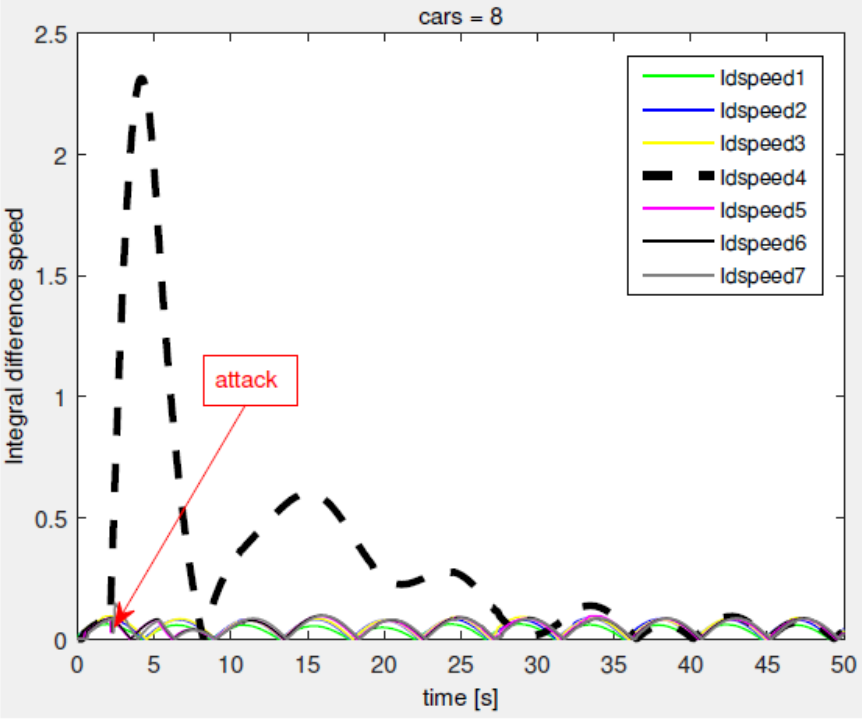


Fig. 3. Integrals of difference of speeds with attack (at time 2 s).

New features

In formulas…

$$\boldsymbol{I_{sys}} = [N, \boldsymbol{x}(0), \boldsymbol{m}, \boldsymbol{F}, D] \;\; \boldsymbol{F} = [\bar{F_0}, \bar{b_F}] \;\; F = F_0 \cdot sin(b_F \cdot t) \;\; \boldsymbol{x}(0) = [\boldsymbol{d}(0), \boldsymbol{v}(0)]$$

$$\iota_{v_i} = \int_0^T |v_{i+1}(t) - v_i(t)| \; dt, i = 0, ..., N-1$$

$$\iota_{d_i} = \int_0^T |d_{i+2}(t) - d_{i+1}(t) \; dt|, i = 0, ..., N-2$$

$$T = t' + 1.1 \cdot D_{Max}$$

Temporal dynamics into ML

Does machine learning (ML) help us in synthesizing temporal dynamics into detection?

$$\boldsymbol{I_{sys}} = [N, \boldsymbol{x}(0), \boldsymbol{m}, \boldsymbol{F}, D] \;\; \boldsymbol{F} = [F_0, b_F] \;\; F = F_0 \cdot sin(b_F \cdot t) \;\; \boldsymbol{x}(0) = [\boldsymbol{d}(0), \boldsymbol{v}(0)]$$

$$\iota_{v_i} = \int_0^T |v_{i+1}(t) - v_i(t)| \; dt, i = 0, ..., N-1$$

$$\iota_{d_i} = \int_0^T |d_{i+2}(t) - d_{i+1}(t) \; dt|, i = 0, ..., N-2$$

$$T = t' + 1.1 \cdot D_{Max}$$

$$\boldsymbol{I} = [\boldsymbol{I_{sys}}, \boldsymbol{I_\iota}]$$

Temporal dynamics into ML

Does machine learning (ML) help us in synthesizing temporal dynamics into detection?

$$\boldsymbol{I_{sys}} = [N, \boldsymbol{x}(0), \boldsymbol{m}, \boldsymbol{F}, D] \;\; \boldsymbol{F} = [F_0, b_F] \;\; F = F_0 \cdot sin(b_F \cdot t) \;\; \boldsymbol{x}(0) = [\boldsymbol{d}(0), \boldsymbol{v}(0)]$$

$$\iota_{v_i} = \int_0^T |v_{i+1}(t) - v_i(t)| \; dt, i = 0, ..., N-1$$

$$\iota_{d_i} = \int_0^T |d_{i+2}(t) - d_{i+1}(t) \; dt|, i = 0, ..., N-2$$

$$T = t' + 1.1 \cdot D_{Max}$$

$$\boldsymbol{I} = [\boldsymbol{I_{sys}}, \boldsymbol{I_\iota}]$$

prediction function $f(\boldsymbol{I}(\cdot), \cdot)$

$\aleph = \{(\boldsymbol{I}^\kappa, \omega^\kappa), \kappa = 1, ..., K\}$ dataset

Temporal dynamics into ML

Does machine learning (ML) help us in synthesizing temporal dynamics into detection?

$$\boldsymbol{I_{sys}} = [N, \boldsymbol{x}(0), \boldsymbol{m}, \boldsymbol{F}, D] \;\; \boldsymbol{F} = [\bar{F_0}, \bar{b_F}] \;\; F = F_0 \cdot sin(b_F \cdot t) \;\; \boldsymbol{x}(0) = [\boldsymbol{d}(0), \boldsymbol{v}(0)]$$

$$\iota_{v_i} = \int_0^T |v_{i+1}(t) - v_i(t)| \; dt, i = 0, ..., N-1$$

$$\iota_{d_i} = \int_0^T |d_{i+2}(t) - d_{i+1}(t) \; dt|, i = 0, ..., N-2$$

$$T = t' + 1.1 \cdot D_{Max}$$

$$\boldsymbol{I} = [\boldsymbol{I_{sys}}, \boldsymbol{I_\iota}]$$

boundary function $f(\boldsymbol{I}(\cdot), \cdot)$ separating the $\boldsymbol{I}^\kappa$ points in $\aleph$, according to the two classes defined by ω.

Temporal dynamics into ML: results

Does machine learning (ML) help us in synthesizing temporal dynamics into detection?

$$F_0 \in [-F_{0Max}, -10] \text{ N}, F_{0Max} = 3358$$
$$b_F \in [0.1, 0.7]$$
$$d(0) \in [5, 10] \text{ m}, v(0) \in [50, 150] \text{ Km/h}$$
$$D \in [0.1, D_{Max}] \text{ s}, D_{Max} = 5$$

Temporal dynamics into ML: results

Does machine learning (ML) help us in synthesizing temporal dynamics into detection?

DT:

if $((\iota_{v_4} > 15) \wedge (D \geq 1.7))$ **then** collision (C=70%)

if $((11 < \iota_{v_4} < 15) \wedge (d(0) \leq 6))$ **then** collision (C=14%)

if $((\iota_{v_4} > 15) \wedge (D \leq 1.7) \wedge (d(0) \leq 8))$ **then** collision (C=8%)

LLM:

if $((\iota_{v_4} > 11) \wedge (b_F \leq 0.55))$ **then** collision (C=96%)

if $((\iota_{v_4} > 6) \wedge (d(0) \geq 7) \wedge (\iota_{d_5} > 32))$ **then** collision (C=66%)

if $((|F_0| > 3103))$ **then** collision (C=7%)

Temporal dynamics into ML: results

Does machine learning (ML) help us in synthesizing temporal dynamics into detection?

DT:

if $((\iota_{v_4} > 15) \wedge (D \geq 1.7))$ **then** collision (C=70%)

if $((11 < \iota_{v_4} < 15) \wedge (d(0) \leq 6))$ **then** collision (C=14%)

if $((\iota_{v_4} > 15) \wedge (D \leq 1.7) \wedge (d(0) \leq 8))$ **then** collision (C=8%)

LLM:

if $((\iota_{v_4} > 11) \wedge (b_F \leq 0.55))$ **then** collision (C=96%)

if $((\iota_{v_4} > 6) \wedge (d(0) \geq 7) \wedge (\iota_{d_5} > 32))$ **then** collision (C=66%)

if $((|F_0| > 3103))$ **then** collision (C=7%)

Feature ranking: $\iota_{v_4}, \iota_{d_5}, b_F, d(0), F_0,$

Complex rules on integrals, still preserving reliable prediction, if I_{sys} is not used.

From intelligible analytics to cognitive machine learning

The analysis makes evident that integrals modulated by the compromised car capture sufficient knowledge to infer the attack, thus disregarding the other features.

However, $f(\boldsymbol{I}_{\iota})$ outlines a complex relationship of ι_{v_i} and ι_{d_i} with $i = 4, ..., 6$ (11 rules with LLM and 8 with DT with 4 conditions, on average). A simplification is therefore introduced.

From intelligible analytics to cognitive machine learning

The analysis makes evident that integrals modulated by the compromised car capture sufficient knowledge to infer the attack, thus disregarding the other features.

However, $f(\boldsymbol{I}_{\iota})$ outlines a complex relationship of ι_{v_i} and ι_{d_i} with $i = 4, ..., 6$ (11 rules with LLM and 8 with DT with 4 conditions, on average). A simplification is therefore introduced.

Cognitive ML: the human operator tends to reproduce and reinterpret the reasoning carried out by artificial intelligence and change it in a new "man-machine“ model.

From intelligible analytics to cognitive machine learning

The analysis makes evident that integrals modulated by the compromised car capture sufficient knowledge to infer the attack, thus disregarding the other features.

However, $f(\boldsymbol{I}_{\iota})$ outlines a complex relationship of ι_{v_i} and ι_{d_i} with $i = 4, ..., 6$ (11 rules with LLM and 8 with DT with 4 conditions, on average). A simplification is therefore introduced.

Cognitive ML*: the human operator tends to reproduce and reinterpret the reasoning carried out by artificial intelligence and change it in a new "man-machine“ model.

*

https://www.quora.com/What-is-the-difference-between-cognitive-computing-and-machine-learning
https://www.ibm.com/blogs/nordic-msp/artificial-intelligence-machine-learning-cognitive-computing/

Cognitive ML is often understood with other meanings:
http://www.lscp.net/persons/dupoux/bootphon/index.html

From intelligible analytics to cognitive machine learning

The analysis makes evident that integrals modulated by the compromised car capture sufficient knowledge to infer the attack, thus disregarding the other features.

However, $f(\boldsymbol{I}_{\iota})$ outlines a complex relationship of ι_{v_i} and ι_{d_i} with $i = 4, ..., 6$ (11 rules with LLM and 8 with DT with 4 conditions, on average). A simplification is therefore introduced.

Cognitive ML here:

$$\textbf{if}((\iota_{v_i} > K \cdot \iota_{v_{i-1}})) \vee ((\iota_{d_i} > K \cdot \iota_{d_{i-1}}) \textbf{ then apply } F_{res}$$

From intelligible analytics to cognitive machine learning

The analysis makes evident that integrals modulated by the compromised car capture sufficient knowledge to infer the attack, thus disregarding the other features.

However, $f(\boldsymbol{I}_{\iota})$ outlines a complex relationship of ι_{v_i} and ι_{d_i} with $i = 4, ..., 6$ (11 rules with LLM and 8 with DT with 4 conditions, on average). A simplification is therefore introduced.

Cognitive ML here:

$$\textbf{if}((\iota_{v_i} > K \cdot \iota_{v_{i-1}})) \vee ((\iota_{d_i} > K \cdot \iota_{d_{i-1}}) \textbf{ then apply } F_{res}$$

Including (F_{res}) control!

F_{res} example

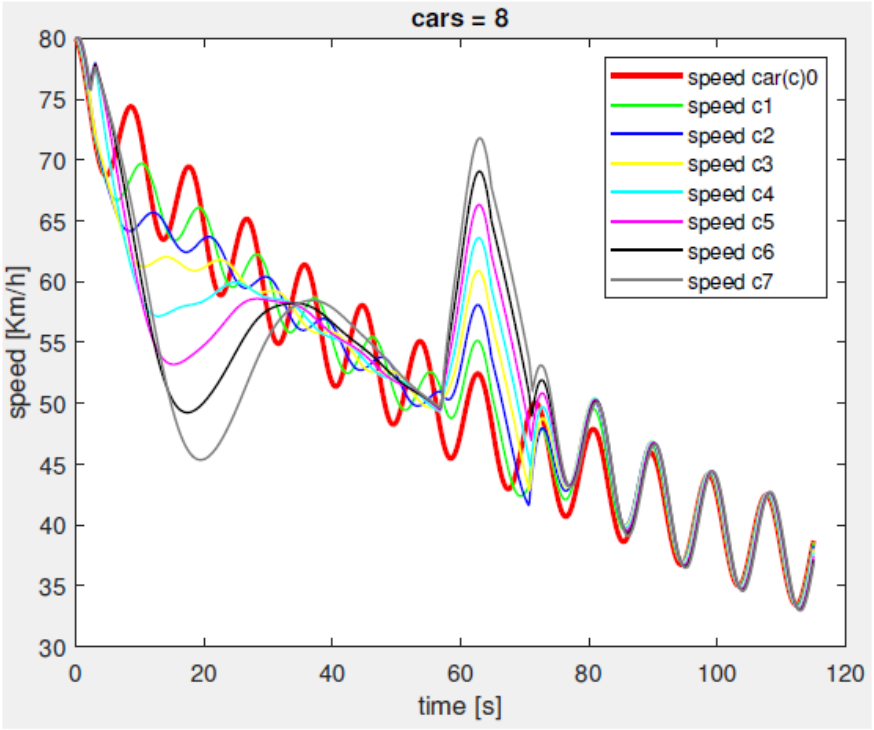


Fig. 4. Countermeasure to the attack: speed.

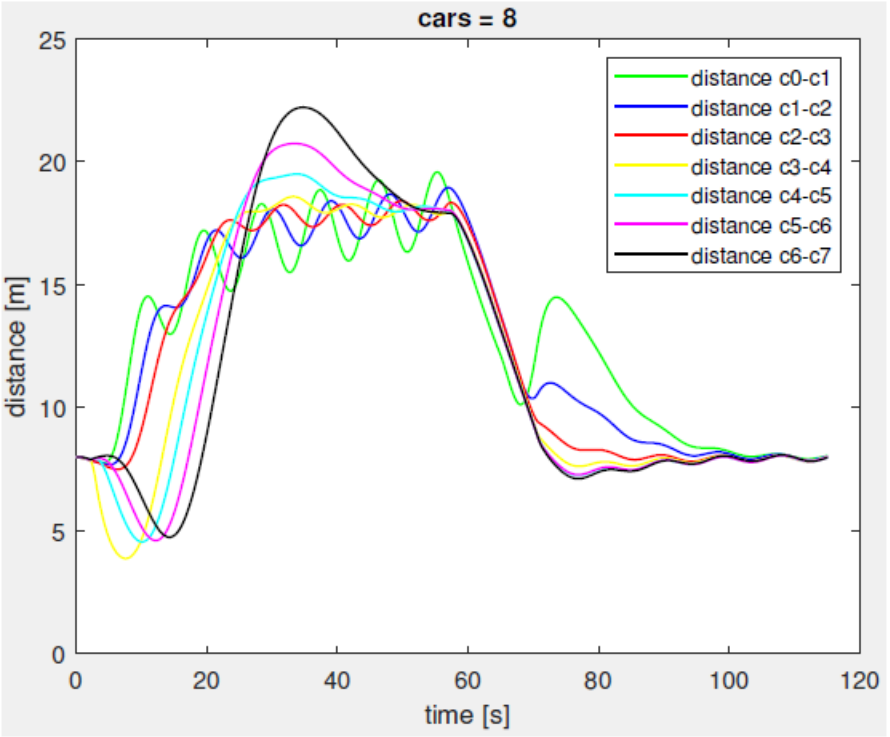


Fig. 5. Countermeasure to the attack: distance.

K=2, F_{res}=-500.

K configuration via ML

$$\textbf{if}((\iota_{v_i} > K \cdot \iota_{v_{i-1}})) \vee ((\iota_{d_i} > K \cdot \iota_{d_{i-1}}) \textbf{ then apply } F_{res}$$

$$\delta\iota_{d_i} = \iota_{d_i} - \iota_{d_{i-1}},\ i = 1, ...N$$

$$\boldsymbol{I_{\delta\iota}} = [\boldsymbol{\delta\iota_v}, \boldsymbol{\delta\iota_d}]$$

$$\boldsymbol{I} = [\boldsymbol{I_{sys}}, \boldsymbol{I_{\delta\iota}}]$$

K configuration via ML

$$\textbf{if}((\iota_{v_i} > K \cdot \iota_{v_{i-1}})) \vee ((\iota_{d_i} > K \cdot \iota_{d_{i-1}}) \textbf{ then apply } F_{res}$$

$$\delta\iota_{d_i} = \iota_{d_i} - \iota_{d_{i-1}},\ i = 1, ...N$$

$$\boldsymbol{I_{\delta\iota}} = [\boldsymbol{\delta\iota_v}, \boldsymbol{\delta\iota_d}]$$

$$\boldsymbol{I} = [\boldsymbol{I_{sys}}, \boldsymbol{I_{\delta\iota}}]$$

Find $f(\boldsymbol{I_{\delta\iota}})$!

K configuration via ML

$$\textbf{if}((\iota_{v_i} > K \cdot \iota_{v_{i-1}})) \vee ((\iota_{d_i} > K \cdot \iota_{d_{i-1}}) \textbf{ then apply } F_{res}$$

$$\delta\iota_{d_i} = \iota_{d_i} - \iota_{d_{i-1}},\ i = 1, ...N$$

$$\boldsymbol{I_{\delta\iota}} = [\boldsymbol{\delta\iota_v}, \boldsymbol{\delta\iota_d}]$$

$$\boldsymbol{I} = [\boldsymbol{I_{sys}}, \boldsymbol{I_{\delta\iota}}]$$

Find $f(\boldsymbol{I_{\delta\iota}})$!

$$\textbf{if}(\iota_{v_i} - \iota_{v_{i+1}} < 2) \vee (\iota_{v_i} - \iota_{v_{i-1}} < 2) \vee (\iota_{d_i} - \iota_{d_{i-1}} < 0.18) \vee (\iota_{d_i} - \iota_{d_{i+1}} < 1) \textbf{ then apply } F_{res}$$

F_{res} configuration

Find $f(\boldsymbol{I_{sys}}, F_{res})$!

F_{res} configuration

Find $f(\boldsymbol{I_{sys}}, F_{res})$!

if $(|F_0| \leq 2762) \wedge (D \leq 1.7)$ **then** safe

if $(|F_0| \leq 1253) \wedge (D \leq 3.5)$ **then** safe

if $(D \leq 1.3)$ **then** safe

if $(|F_0| > 560) \wedge (D > 3.7) \wedge (b_F \leq 0.6) \wedge (|F_{res}| > 97)$ **then** collision

if $(|F_0| < 719)$ **then** safe

if $(|F_0| > 1173) \wedge (d(0) > 5) \wedge (v(0) \leq 132) \wedge (D > 3)$ **then** collision

if $(|F_0| > 1089) \wedge (d(0) > 5) \wedge (v(0) \leq 132) \wedge (D > 2.6) \wedge (b_F \leq 0.56) \wedge (|F_{res}| > 126)$ **then** collision

$C \in [30\%, 45\%]$

F_{res} is not relevant! Feature ranking: $\{D, F_0, v(0), b_F, F_{res}, d(0)\}$

F_{res} configuration

Objective: safety regions with FNR=0%.

F_{res} optimal thresholds are found for different F_0 intervals => F_0 should be known to calibrate the response to the attack:

F_{res} configuration

Objective: safety regions with FNR=0%.

F_{res} optimal thresholds are found for different F_0 intervals => F_0 should be known to calibrate the response to the attack:

$$F_0 \in [-1000, -10]\text{: } F^o_{res} = -2000,\ d^o(0) = 10,\ v(0) < 100$$
$$F_0 \in [-2000, -1000]\text{: } F^o_{res} = -2000,\ d^o(0) = 17,\ v(0) \leq 80$$
$$F_0 \in [-F_{0Max}, -2000]\text{: } F^o_{res} = -4000,\ d^o(0) = 26.5,\ v(0) \leq 55$$

independently to D and b_F

F_{res} configuration

Objective: safety regions with FNR=0%.

F_{res} optimal thresholds are found for different F_0 intervals => F_0 should be known to calibrate the response to the attack:

$$F_0 \in [-1000, -10]\text{: } F^o_{res} = -2000,\ d^o(0) = 10,\ v(0) < 100$$
$$F_0 \in [-2000, -1000]\text{: } F^o_{res} = -2000,\ d^o(0) = 17,\ v(0) \leq 80$$
$$F_0 \in [-F_{0Max}, -2000]\text{: } F^o_{res} = -4000,\ d^o(0) = 26.5,\ v(0) \leq 55$$

independently to D and b_F

The worst case is actually impractical as it leads to platoons working at low speed and large distances. This is however not surprising as it is a platoon able to resist to attack under extreme braking conditions.

Safety

Safety: air bag

Automotive systems are required to operate under strict safety constraints.
FMEA (Failure Mode and Effects Analysis) analyses potential failures of system components, assessing and ranking the risks associated with them, and then identifying and addressing the most serious problems.
The FMEA process can be time-intensive and the analysis is sometimes informal.

Safety: air bag

The airbag system consists of three major component types: sensors, crash evaluators and actuators. The sensors are used to detect accidents such as impacts or the car rolling, and the information from the sensors is then processed by two independent crash evaluators. If both evaluators agree that a crash has occurred, then the actuators respond by deploying the airbags.

Safety: air bag

The use of a second crash evaluator is a recent addition to airbag systems, aimed at avoiding unnecessary deployment, which is seen as the most dangerous malfunction that can occur.
FMEA considers variants of the airbag system with both one and two crash evaluators.

Gethin Norman and David Parker, Quantitative Verification Formal Guarantees for Timeliness, Reliability and Performance, Knowledge Transfer Report, London Mathematical Society and Smith Institute for Industrial Mathematics and System Engineering.

Safety in cyber physical system

safecop.eu

Safety:

safeCOP project - italian use case

Cyber-physical systems, such as automobiles, cars, and medical devices, comprise both a physical part and a software part, whereby the physical part of the system sends information about itself to the software part, and the software sends information, usually in the form of commands, to the physical part.

P. G. Larsen, J. Fitzgerald, J. Woodcock, and T. Lecomte. Trustworthy Cyber-Physical Systems Engineering, Chapter 8: Collaborative Modelling and Simulation for Cyber-Physical Systems. Chapman and Hall/CRC, September 2016. ISBN 9781498742450.

SafeCOP ITA pilot

https://www.youtube.com/watch?v=4VirpA0HzP0&t=93s

Safety:

safeCOP project - italian use case

Hazard risk analysis example (On Board Unit, Road Side Unit).

ID	Hazard Description	Safety Goal	ASIL	Safe state
H0	The OBU warns the driver of a potential danger while there is no actual danger	None	QM	N/A
H1	The OBU notifies the infrastructure of a potential danger while there is no actual danger	None	QM	N/A
H2	The OBU fails to warn the driver about the presence of danger	None	QM	N/A
H3	The OBU fails to notify the infrastructure about the presence of a danger	The OBU shall guarantee notification of the presence of a danger to the infrastructure	ASIL-A	Inhibit transmission to the infrastructure
H9	The OBU communicates wrong dynamic or operational state information to the infrastructure	The OBU shall transmit correct dynamic and operational state data to the infrastructure	ASIL-A	Inhibit transmission to the infrastructure

ID	Hazard Description	Safety Goal	ASIL	Safe state
H0	The RSU-C fails to detect a car accident or a stationary vehicle in a dangerous position	The RSU-C shall guarantee that there are no car accident or stationary vehicles if it has detected none	ASIL-B	Notify a ASIL reduction and inhibit any other transmission to the CU
H2	The RSU-C fails to detect a vehicle moving along a forbidden direction	The RSU-C shall guarantee that there are no vehicle moving along forbidden direction if it has detected none	ASIL-B	

Safety and AI

Safety and AI

- Sistemi di guida autonomi

https://www.dmove.it/news/tesla-annuncio-shock-il-computer-per-la-guida-autonoma-e-gia-pronto-presentazione-il-19-aprile

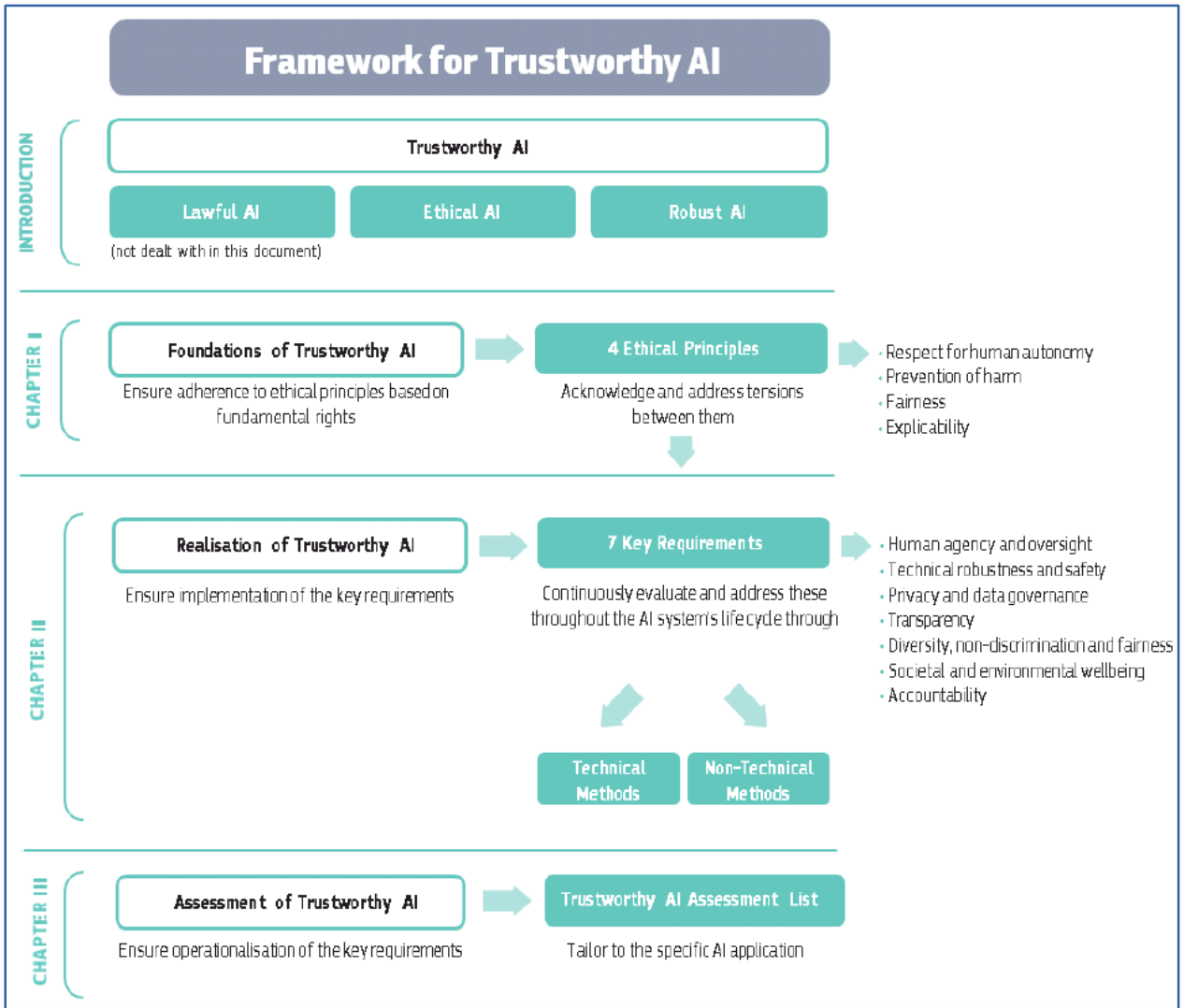
Safety and AI

Trustworthy AI
Trustworthiness is a prerequisite for people and societies to develop, deploy and use AI systems.
Without AI systems – and the human beings behind them – being demonstrably worthy of trust, unwanted consequences may ensue and their uptake might be hindered, preventing the realisation of the potentially vast social and economic

https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

Safety and AI

Trustworthy AI
https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai



Safety and AI

Gruppo ISO/IEC JTC 1/SC 42:
provide guidance to JTC 1, IEC, and ISO committees developing Artificial Intelligence applications

https://jtc1info.org/jtc1-press-committee-info-about-jtc-1-sc-42

Safety and AI

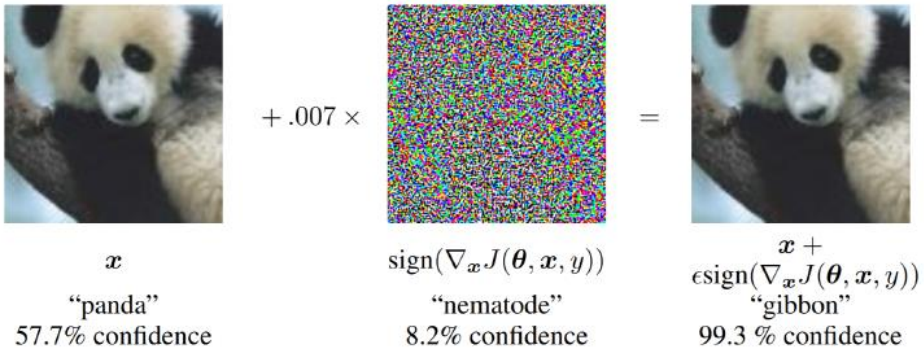


Figure 1: Adding an imperceptibly small vector to an image changes the GoogLeNet [39] image recognizer's classification of the image from "panda" to "gibbon." Figure taken from Goodfellow *et al.* [9].

- Approccio controllistico
  ✓Regions of attraction: The Lyapunov Neural Network: Adaptive Stability Certification for Safe Learning of Dynamical Systems.
- Approcci di verifica formale
  ✓Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks.
  ✓Tools (PRISM, NuSMV) extesions for cyber physical systems (our idea: starting with intelligible safety regions…!)

Our approach:

refinement of ML models via formal logic

Intelligible analytics

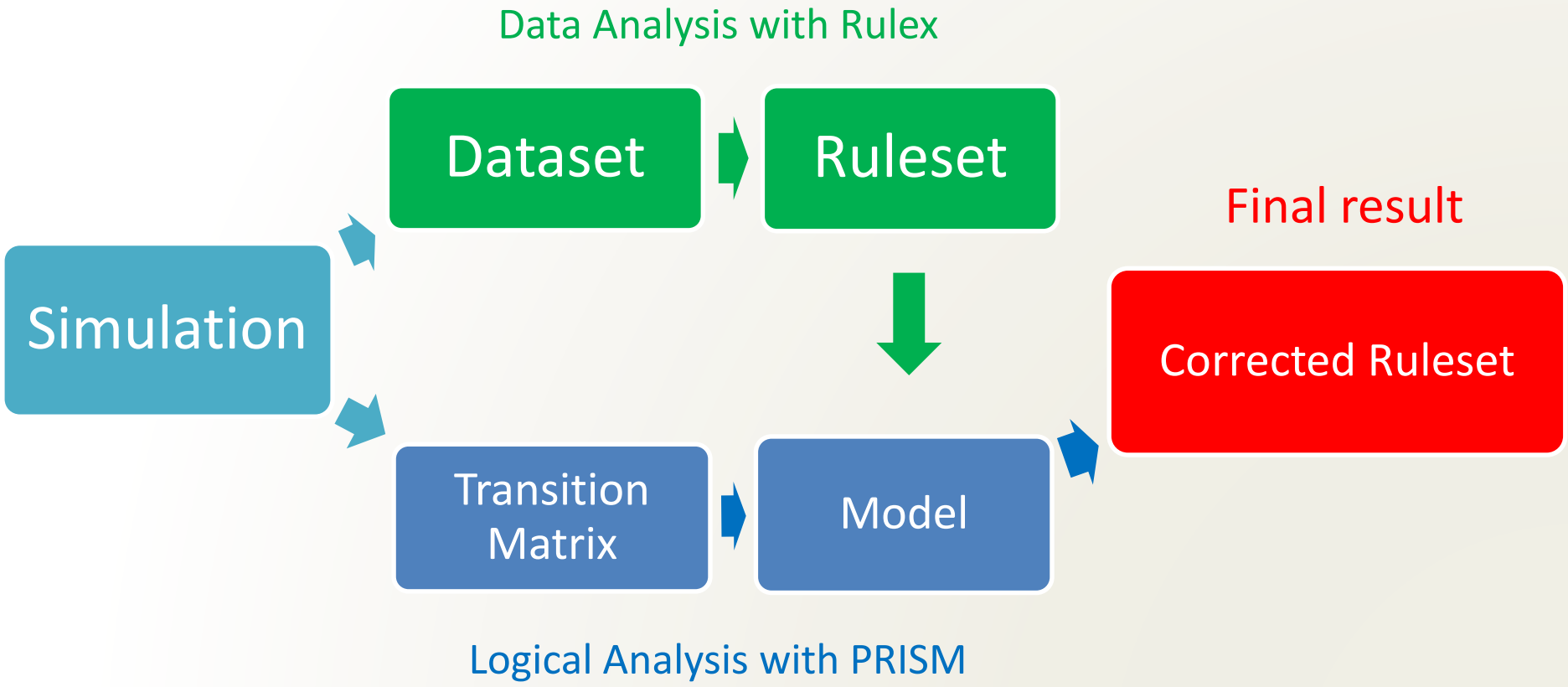
N	F0	m	d_ms	d0	v0	prob	collision
7	6585	1050	0.005	5.092	60	0.380	collision
3	6369	1050	0.005	4.101	49	0.290	safe
8	3531	1050	0.005	2.326	59	0.290	collision
5	4726	1050	0.005	2.108	44	0.350	collision
7	2395	1050	0.005	2.016	27	0.380	collision
6	3196	1050	0.005	2.612	18	0.110	safe

Extraction of intelligibles rules for safety (classification problem) after brute force simulation method.

```
if ((d0 <= 2.380932) && (v0 > 29)) return "collision";
if ((v0 > 24 && v0 <= 73) && (prob <= 0.165000)) return "safe";
if ((d0 > 2.387646 && d0 <= 2.818141) && (v0 > 35)) return "collision";
if ((F0 > 3770) && (v0 > 79) && (prob > 0.155000)) return "collision";
if ((d0 > 2.455520) && (v0 <= 34) && (prob <= 0.385000)) return "safe";
if ((F0 <= 3771) && (d0 > 2.880642)) return "safe";
if ((F0 <= 3018)) return "safe";
```

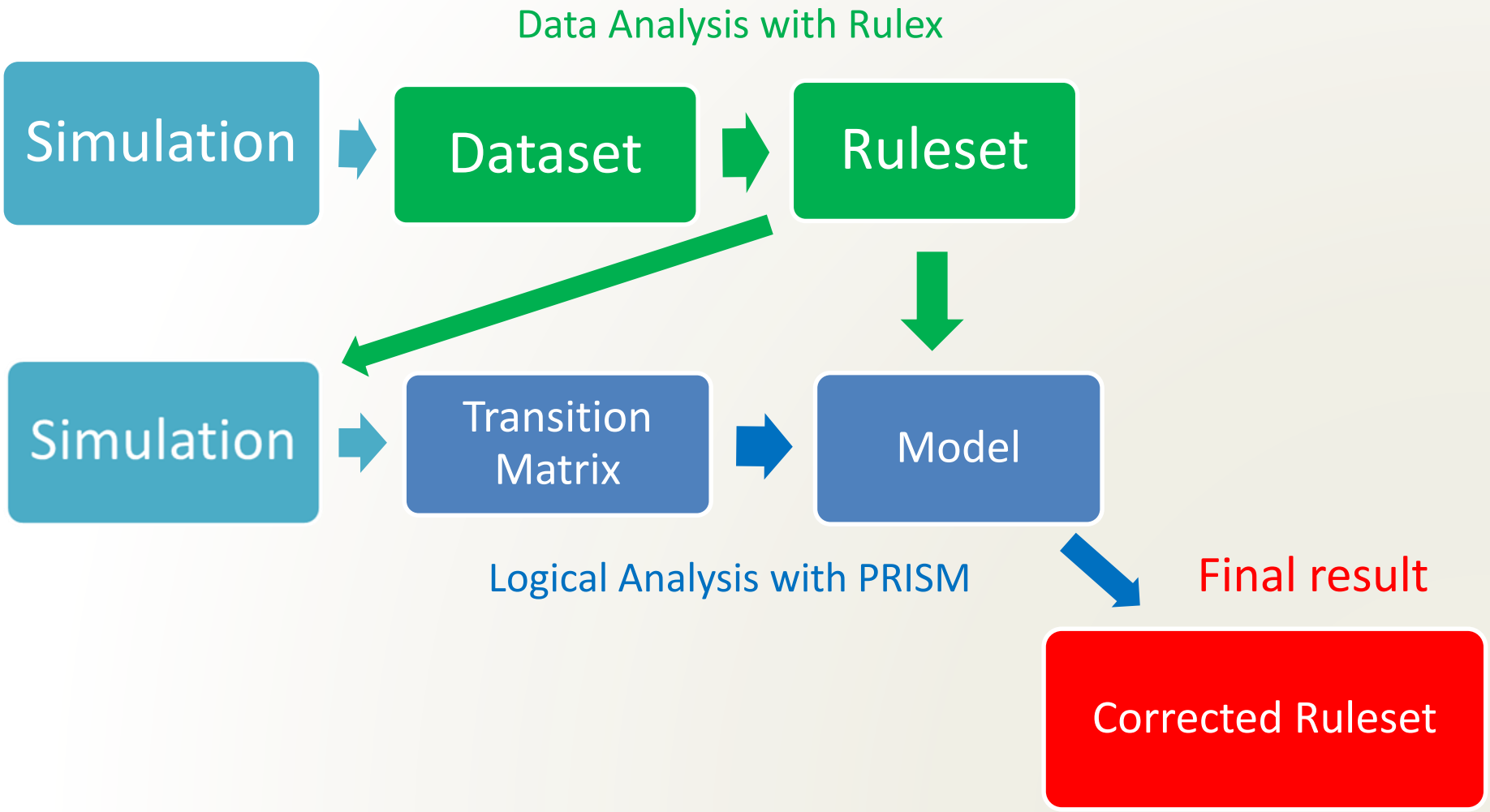
Correction of ruleset with PRISM

Correction scheme when we are able to create in PRISM a Discrete Time Markov chain with n-dimensional status in a n-variables classification problem.



Correction of ruleset with PRISM

When dimension of status of DTM in PRISM are lower the number of variables involved in data analysis, a new simulation after rules extraction have to be done before correction.



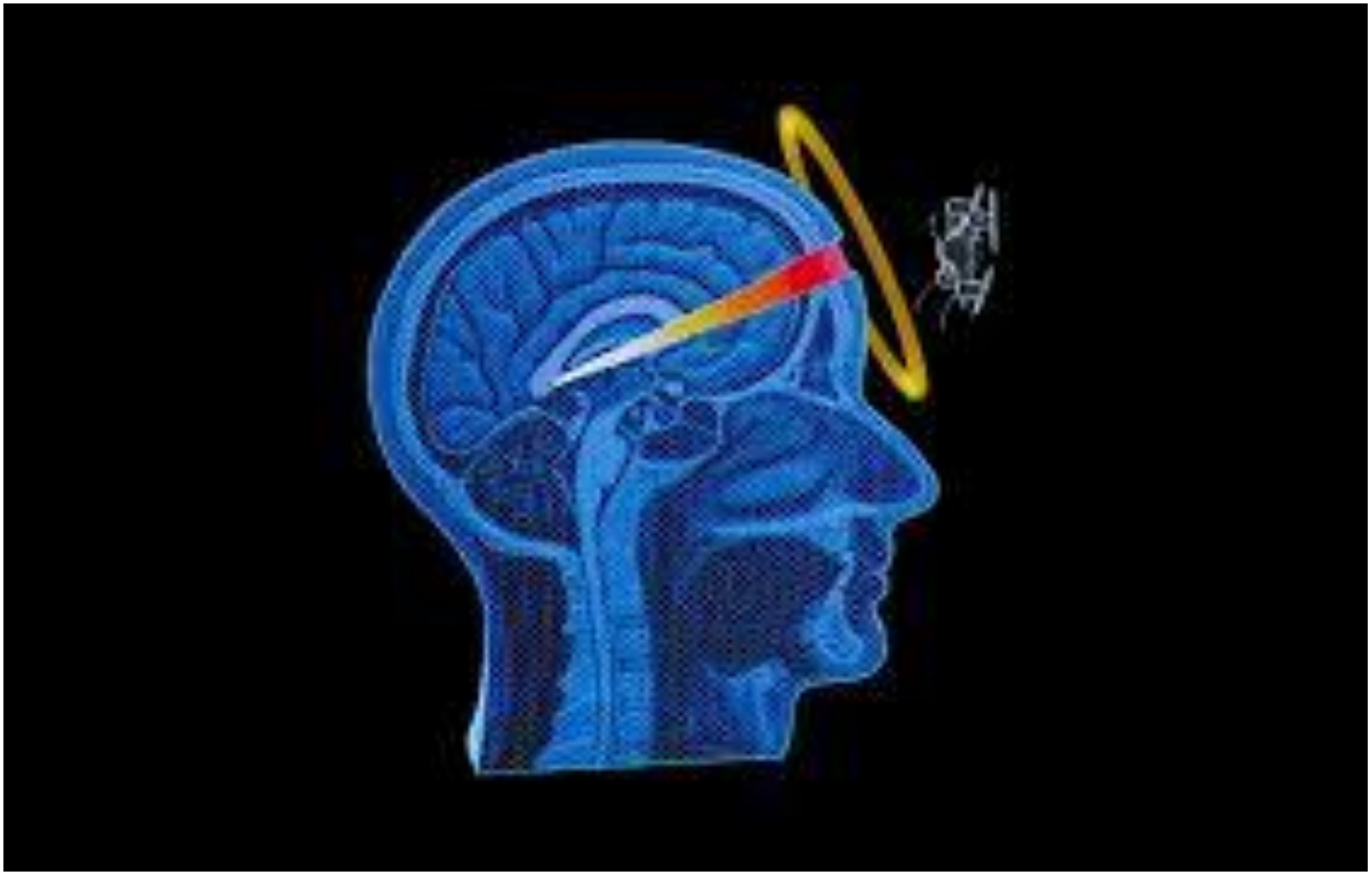
Correction of ruleset with PRISM

```
if ((d0 < 1.876431) && (v0 > 36)) return "collision";
if ((d0 > 3.372723) && (v0 <= 85)) return "safe";
```

We can use the probabilistic informations derived from PRISM for having more control on ruleset obtained in Rulex. A rule can became more strict or flexible, depending on our goals.

```
if ((d0 < 1.876431) && (v0 > 40)) return "collision";
if ((d0 > 4.000000) && (v0 <= 85)) return "safe";
```

THANK YOU



AI ethical issues: micro drone killer

https://www.youtube.com/watch?v=TlO2gcs1YvM

Performance prediction: state of the art

Many control algorithms

Mathematical modeling vs brute force simulation

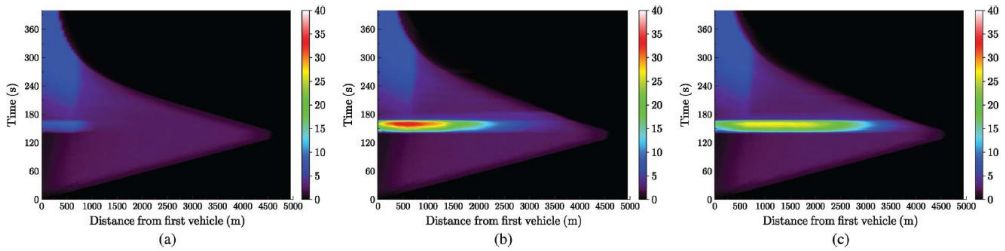


Fig. 2. Average load as a function of the time and the distance from the platoon head for the five-lane scenario with an average speed of 130 km/h. (a) EEB. (b) EEBR. (c) EEBA.

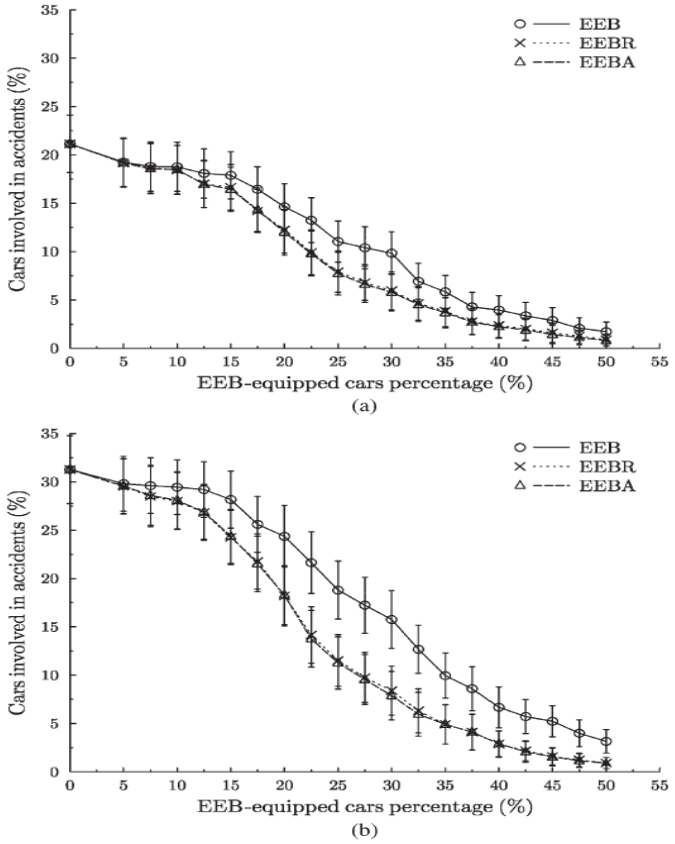
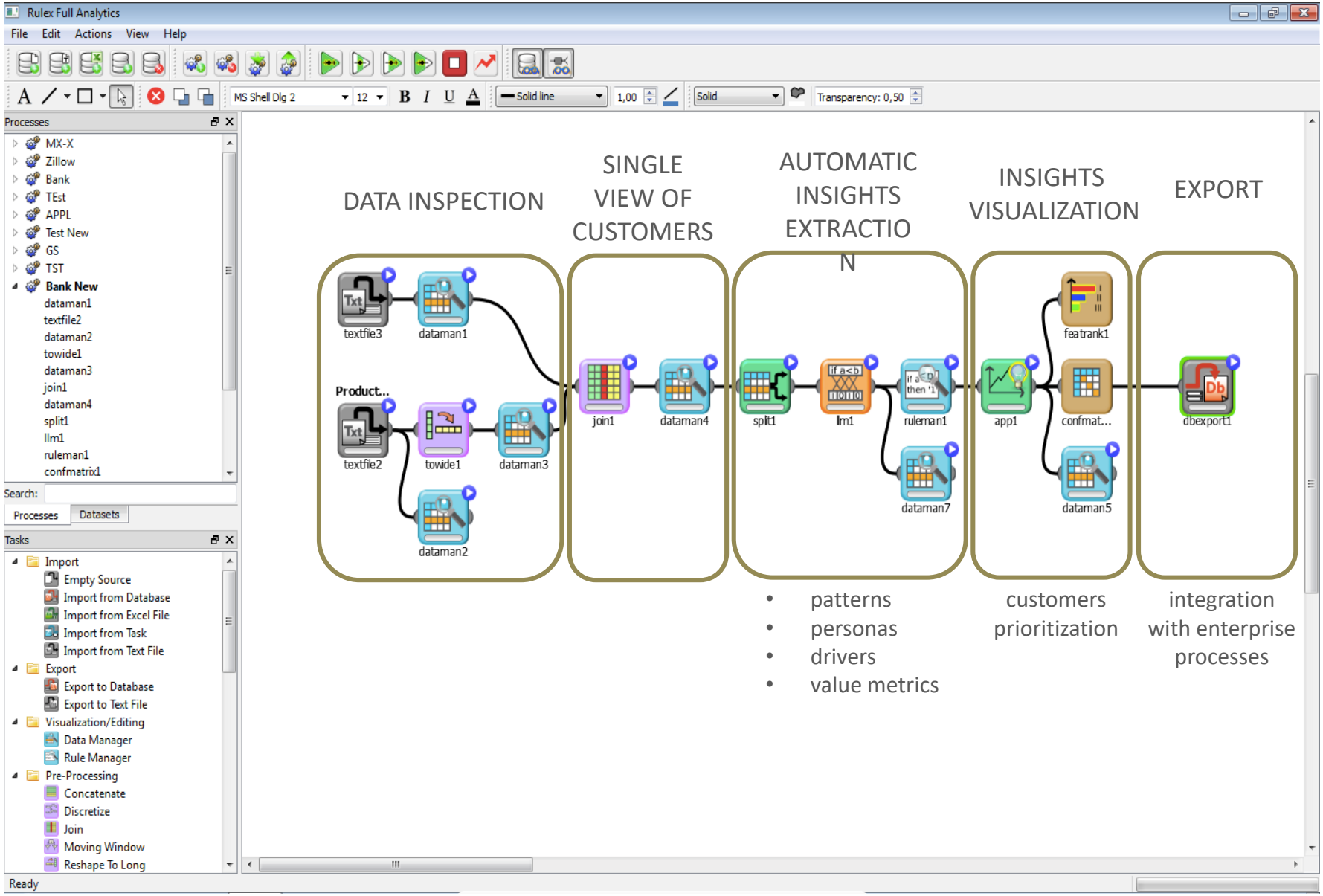


Fig. 3. Percentage of cars involved in accidents versus MPR for single-lane tests for the different protocols and average speeds of 130 and 150 km/h. (a) Reference speed of 130 km/h. (b) Reference speed of 150 km/h.

M. Segata and R. Lo Cigno, "Automatic Emergency Braking: Realistic Analysis of Car Dynamics and Network Performance," in *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4150-4161, Nov. 2013.
doi: 10.1109/TVT.2013.2277802.

Fully developed platform



Rulex Logic Learning Machines (LLM) Compared to Traditional Methods

	rulex analytics LLM	RANDOM FOREST	DECISION TREES	NEURAL NETWORKS	FUZZY LOGIC	TRADITIONAL STATISTIC
MODELS ARE FULLY INTELLIGIBLE (RULES)	✓		✓		✓	
RULES WITH MULTI-VARIABLE CORRELATIONS	✓	✓			✓	
CAN TREAT QUALITATIVE VARIABLES	✓	✓	✓		✓	
PRIOR INFORMATION NOT NEEDED	✓	✓	✓	✓		
MODELING IS HARDLY AFFECTED BY PARAMETERS SETTING	✓		✓			✓
REDUNDANT VARIABLES DETECTED AND IGNORED	✓		✓			✓
KEY VALUES FOR ORDERED VARIABLES ARE AUTOMATICALLY DETERMINED	✓		✓		✓	
RELEVANCE INDICATORS FOR RULES, VARIABLES & THRESHOLDS	✓				✓	
MODELS CAN BE MODIFIED AND TESTED INTERACTIVELY	✓		✓			
HIGH ACCURACY	✓	✓		✓		

Covering and error of rules:
understand the impact of the features

	# Cond	Output	Cond 1	Cond 2	Cond 3	Cond 4
1	2	collision = 1	init. distance ≤ 24	init. speed > 30		
2	2	collision = 0	braking force > 500	init. speed ≤ 30		
3	3	collision = 1	braking force ≤ 1345	init. distance ≤ 33	init. speed > 35	
4	2	collision = 1	init. distance ≤ 30	init. speed > 64		
5	1	collision = 0	init. speed ≤ 24			
6	3	collision = 1	braking force ≤ 1816	comm. delay > 54	init. speed > 55	
7	4	collision = 0	braking force > 217	weight > 1019	init. distance > 24	init. speed ≤ 47
8	4	collision = 1	braking force ≤ 2510	weight > 1221	init. distance ≤ 25	init. speed > 17

	# Patt.	Covering	w\o Cond 1	w\o Cond 2	w\o Cond 3	w\o Cond 4
1	4261	55.0105609012	39.1222717672	3.6376437456		
2	4139	43.2713215753	3.5757429331	51.7274704035		
3	4261	37.8549636236	39.2396151138	8.2140342643	7.1579441446	
4	4261	35.4846280216	13.0485801455	43.4639755926		
5	4139	34.6943706209	65.3056293791			
6	4261	33.9122271767	17.3433466322	7.8150668857	24.0553860596	
7	4139	32.0125634211	0.2416042522	18.3619231698	10.6789079488	15.9942014979
8	4261	28.5379019010	5.0222952359	25.5339122272	22.1544238442	0.0704060080

	# Patt.	Error	w\o Cond 1	w\o Cond 2	w\o Cond 3	w\o Cond 4
1	4139	4.0589514375	49.0939840541	16.0666827736		
2	4261	2.8162403192	3.0509270124	75.5925839005		
3	4139	4.8562454699	20.1014737859	5.7501812032	13.3607151486	
4	4139	3.9864701619	9.8574534912	44.9867117661		
5	4261	2.5580849566	97.4419150434			
6	4139	3.5515825079	9.7608117903	3.6965450592	28.0502536845	
7	4261	4.8345458812	1.2907768130	1.0795587890	12.3210513964	23.8206993663
8	4139	3.7448659096	1.7153901909	7.0306837400	22.6866392849	3.1166948538

Visualization of rules helps understand

collision

0

1

Total number of rules: 31

Rule #19:

Output value: 1
Covering:
55.0105609012%
Error:
4.0589514375%

Conditions:

init. distance <= 24

init. speed > 30