Platooning:
collision prediction by machine learning

SafeCOP Technical WP workshops @ DTU Compute

Copenhagen, Denmark, Nov. 14-15 2016

Enrico Ferrari (Impara), Maurizio Mongelli, Marco Muselli (CNR-IEIIT)

e.ferrari@rulex-inc.com

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

Index of the presentation

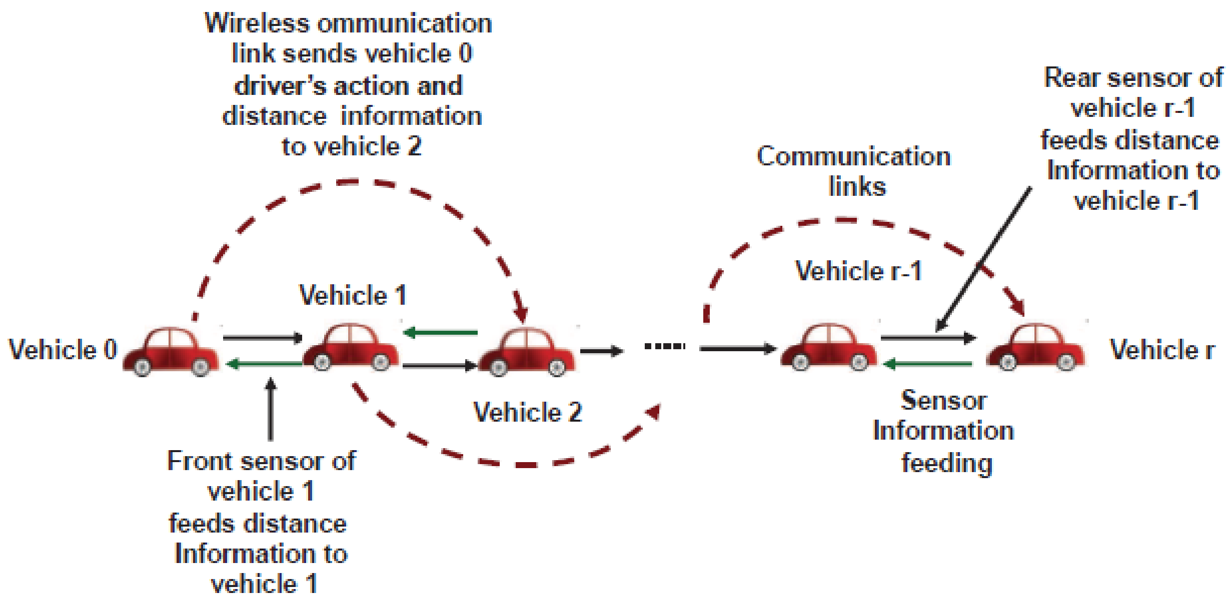
- Platooning
  - performance prediction?

  - Machine learning
    - database of metrics and performance
    - knowledge extraction

  - Conclusions and open issues

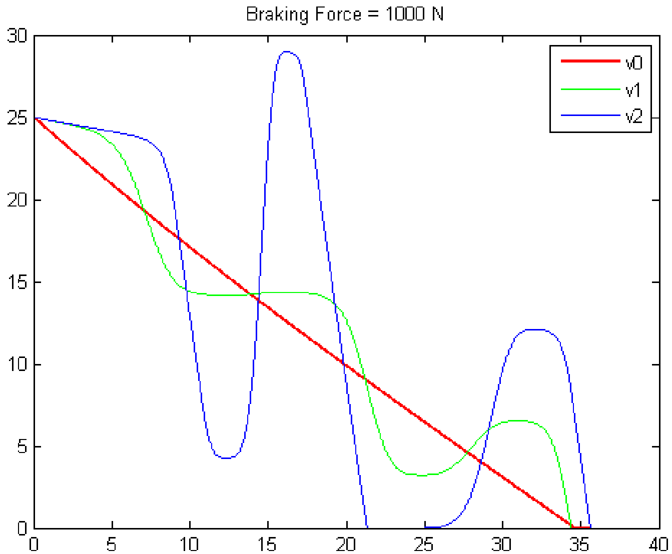
Platooning: problem

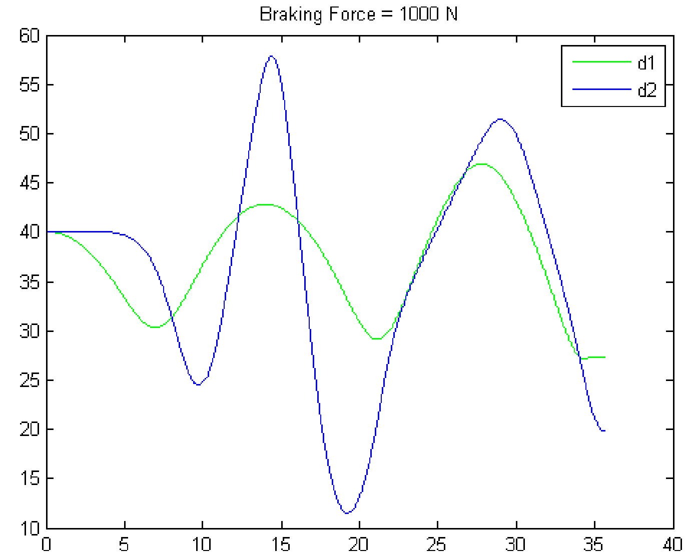
Leading vehicle (#0) applies a braking force
Parameters: # vehicles, initial distance, initial speed, force, weight, communication delay (control law assumed fixed)
Can we predict collision?

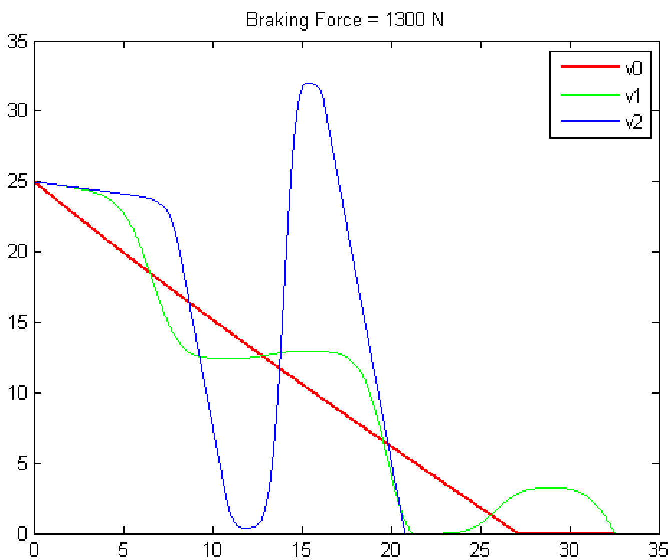


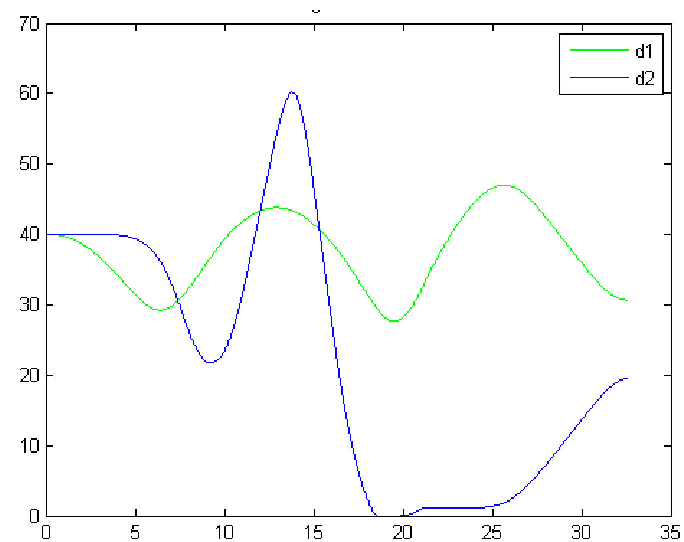
L. Xu, L. Y. Wang, G. Yin and H. Zhang, "Communication Information Structures and Contents for Enhanced Safety of Highway Vehicle Platoons," in *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4206-4220, Nov. 2014. doi: 10.1109/TVT.2014.2311384.

Platooning: example

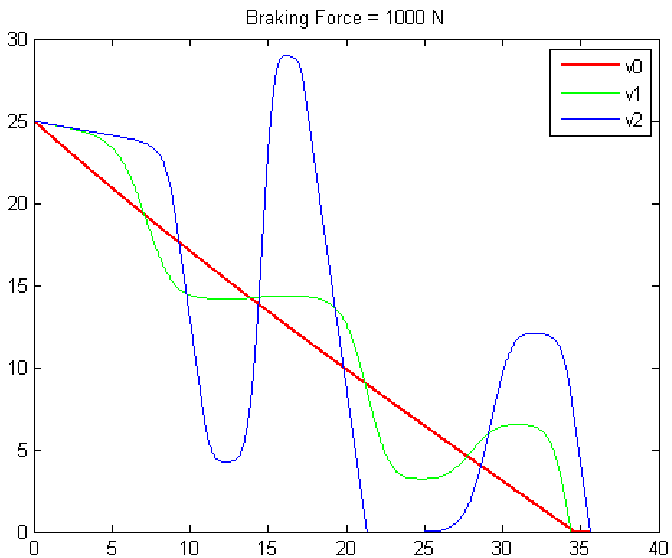


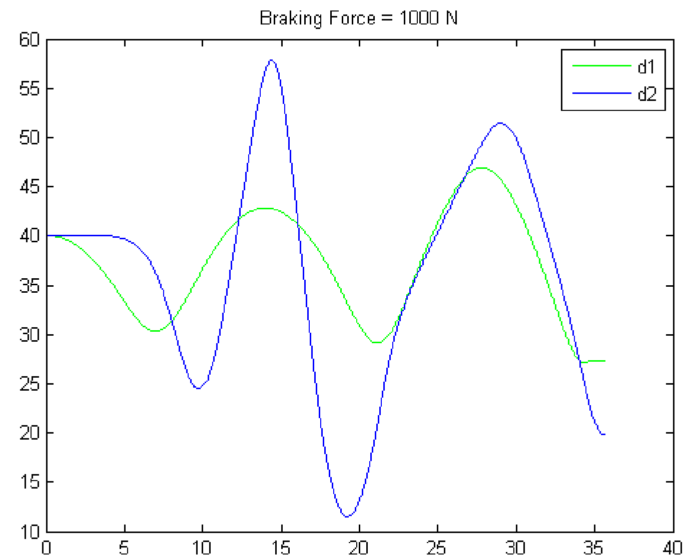


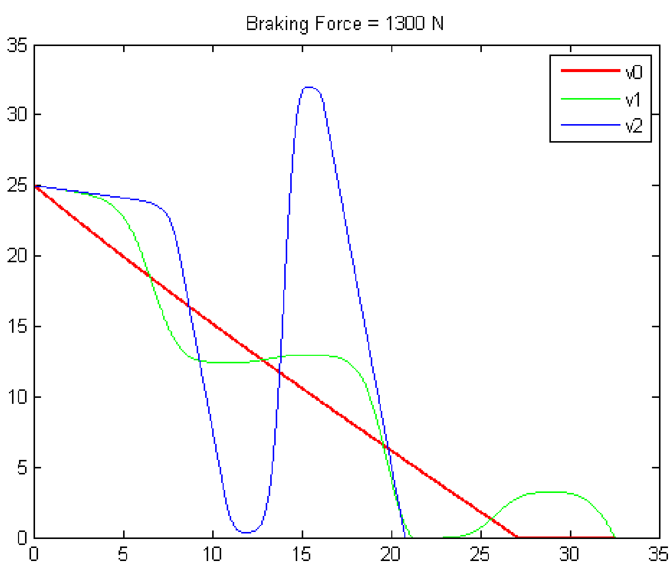


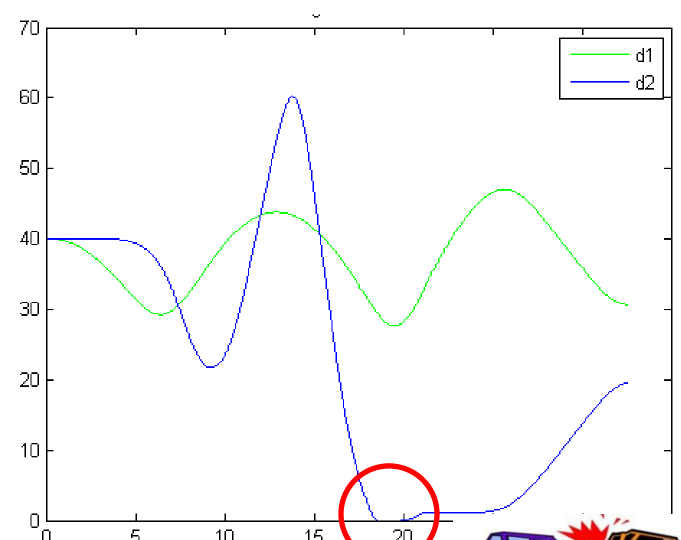


Platooning: example









Performance prediction: state of the art

Performance prediction: state of the art

Many control algorithms

Mathematical modeling vs brute force simulation

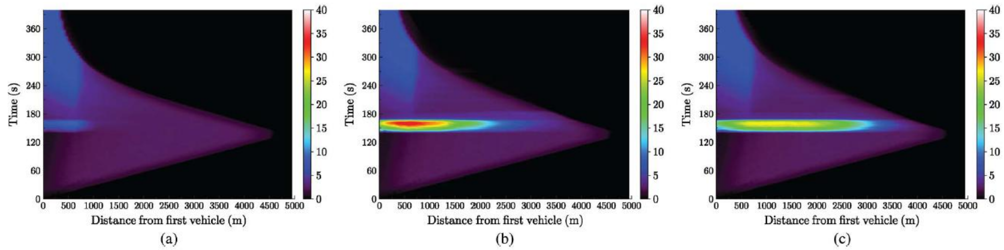


Fig. 2. Average load as a function of the time and the distance from the platoon head for the five-lane scenario with an average speed of 130 km/h. (a) EEB. (b) EEBR. (c) EEBA.

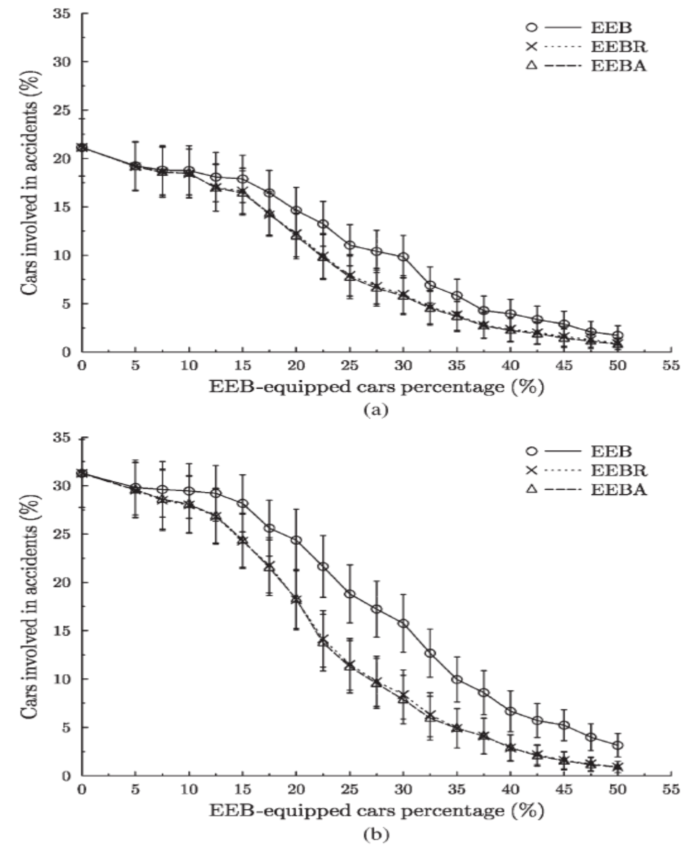


Fig. 3. Percentage of cars involved in accidents versus MPR for single-lane tests for the different protocols and average speeds of 130 and 150 km/h. (a) Reference speed of 130 km/h. (b) Reference speed of 150 km/h.

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.

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

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

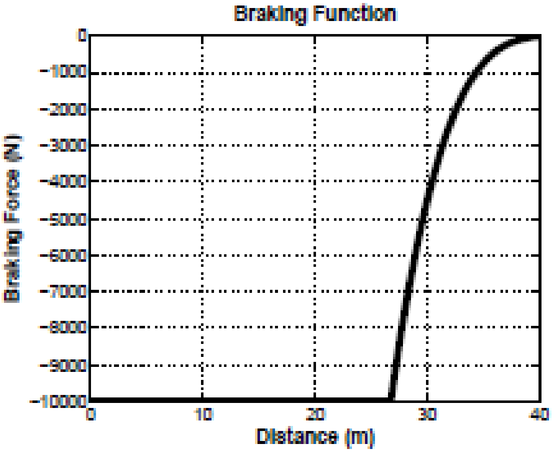
L. Xu, L. Y. Wang, G. Yin and H. Zhang, "Communication Information Structures and Contents for Enhanced Safety of Highway Vehicle Platoons," in *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4206-4220, Nov. 2014. doi: 10.1109/TVT.2014.2311384.

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

2^{nd} step: knowledge extraction

Machine Learning

2nd step: knowledge extraction

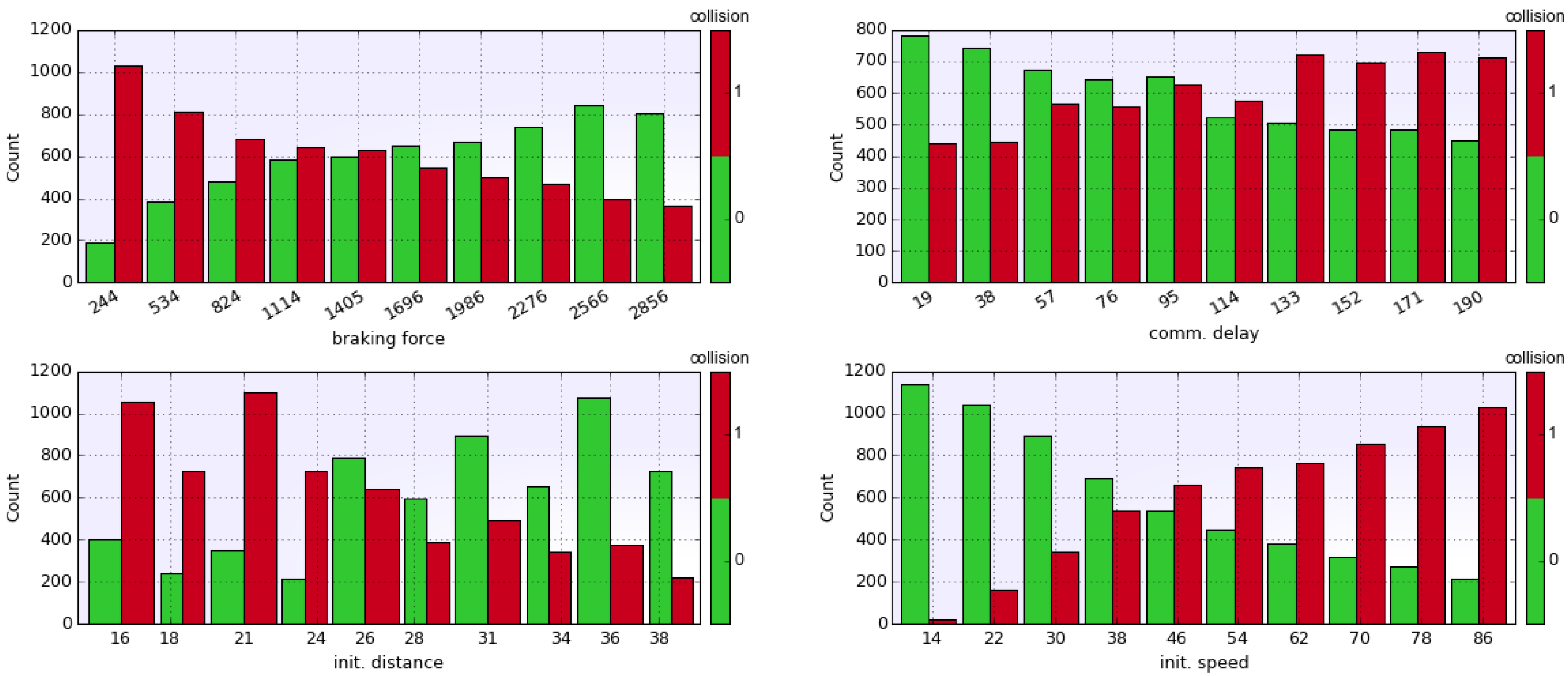
Is the problem difficult?

Is this problem difficult?

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

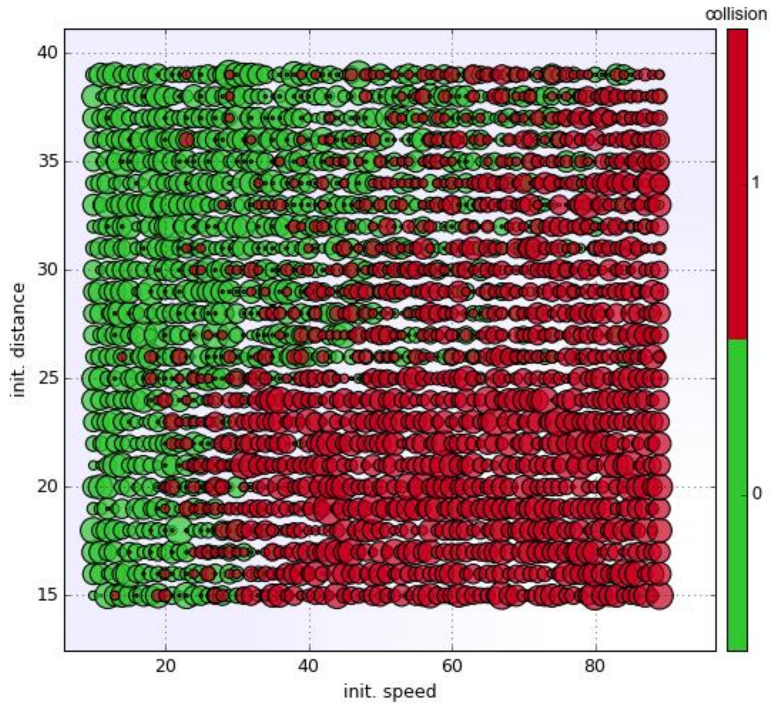
vehicles = 3,
initial distance in [15, 55] m,
initial speed in [10, 90] km/h,
force in [100, 3000] N,
vehicle weight in [500, 2500] Kg,
communication delay in [10, 200] ms .

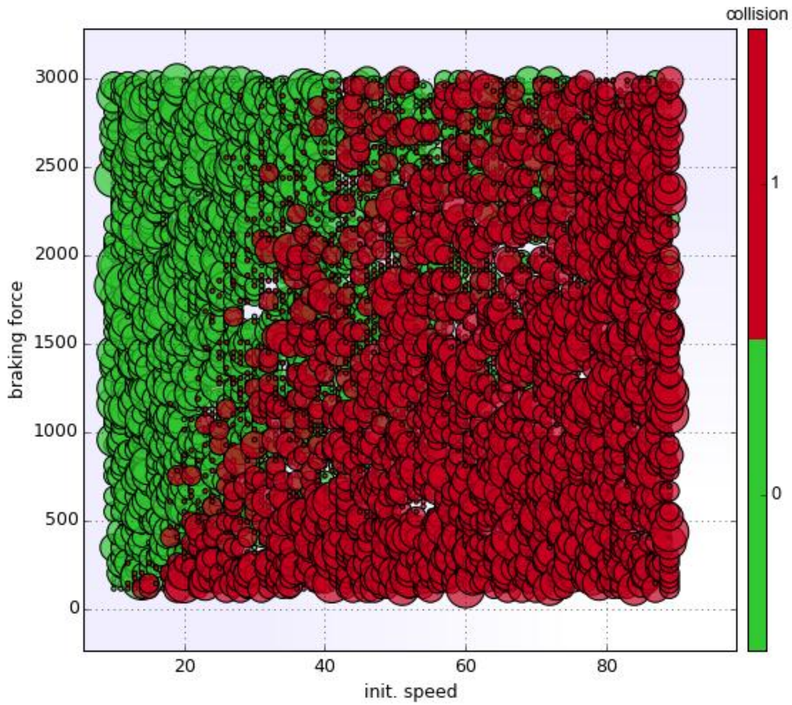
Univariate analysis by histograms
not easy to understand!



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

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

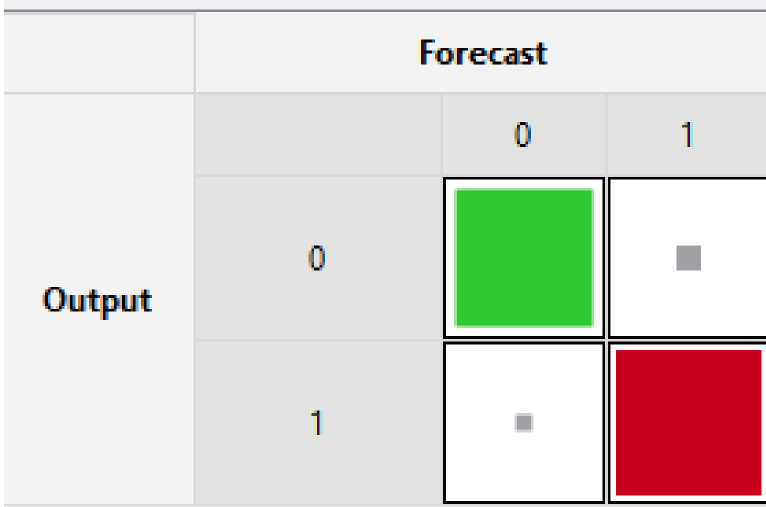
Rulex platform: intelligible rules

```
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%)**

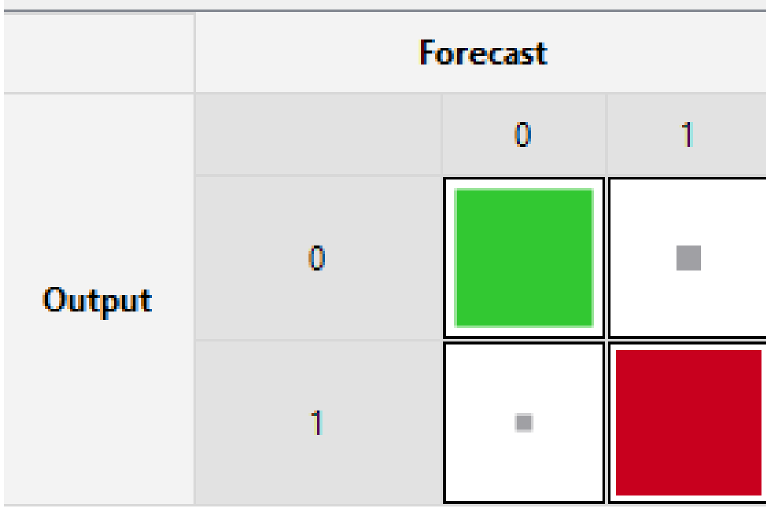


84% True Negatives
(no collision correctly predicted).

90% True Positives (collisions
correctly predicted).

Confusion matrix

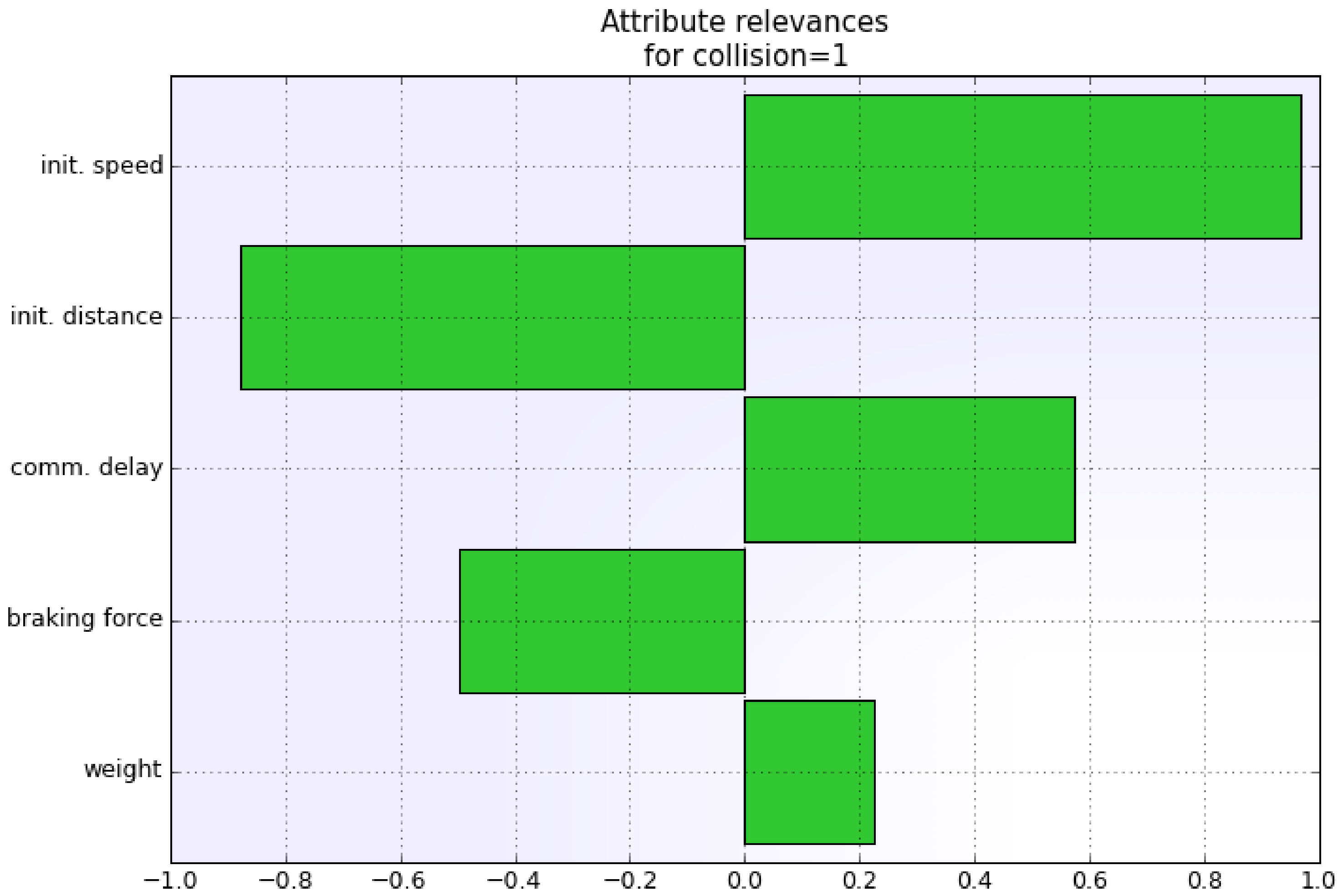
		Forecast		
Output		0	1	Total
	0	**5024 (84.6503791...**	911 (15.3496208930...	5935 (49.458333333...
	1	577 (9.5136026381%)	**5488 (90.4863973...**	6065 (50.541666666...
	Total	5601 (46.675000000...	6399 (53.325000000...	**12000 (100%)**



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

A further elaboration on how to characterize FNs is needed

Feature ranking

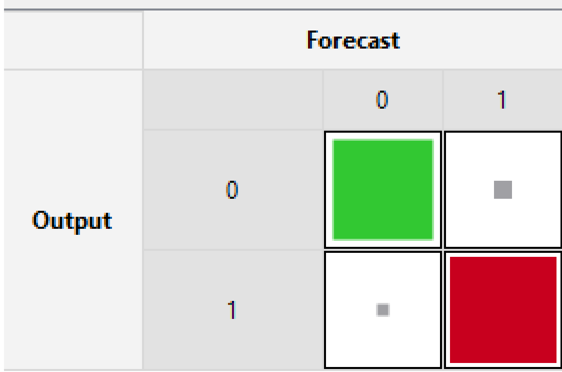


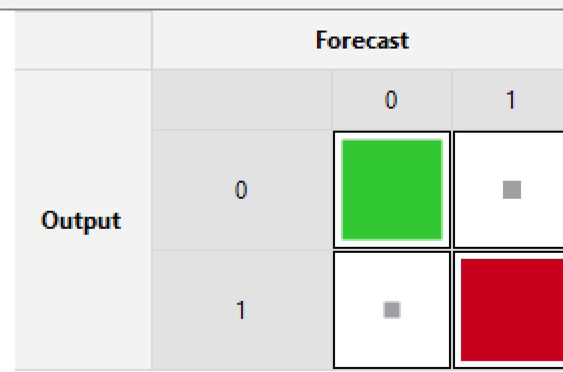
Increasing initial speed has the highest relevance on collisions.
The opposite holds true for the initial distance.

Confusion matrixes of 2 models: with and without delay

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

	Forecast		
Output	0	1	Total
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%)**

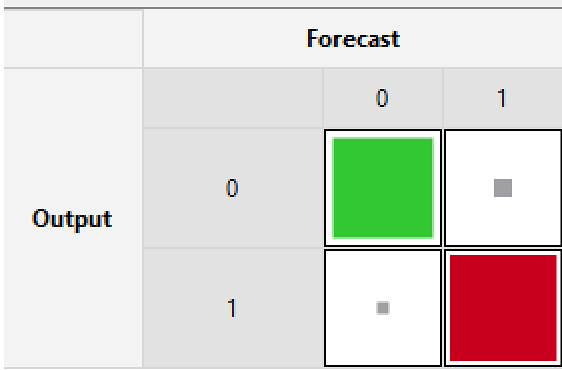


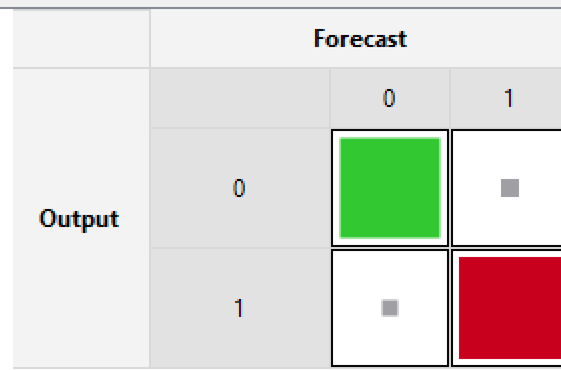


Confusion matrixes of 2 models: with and without delay

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

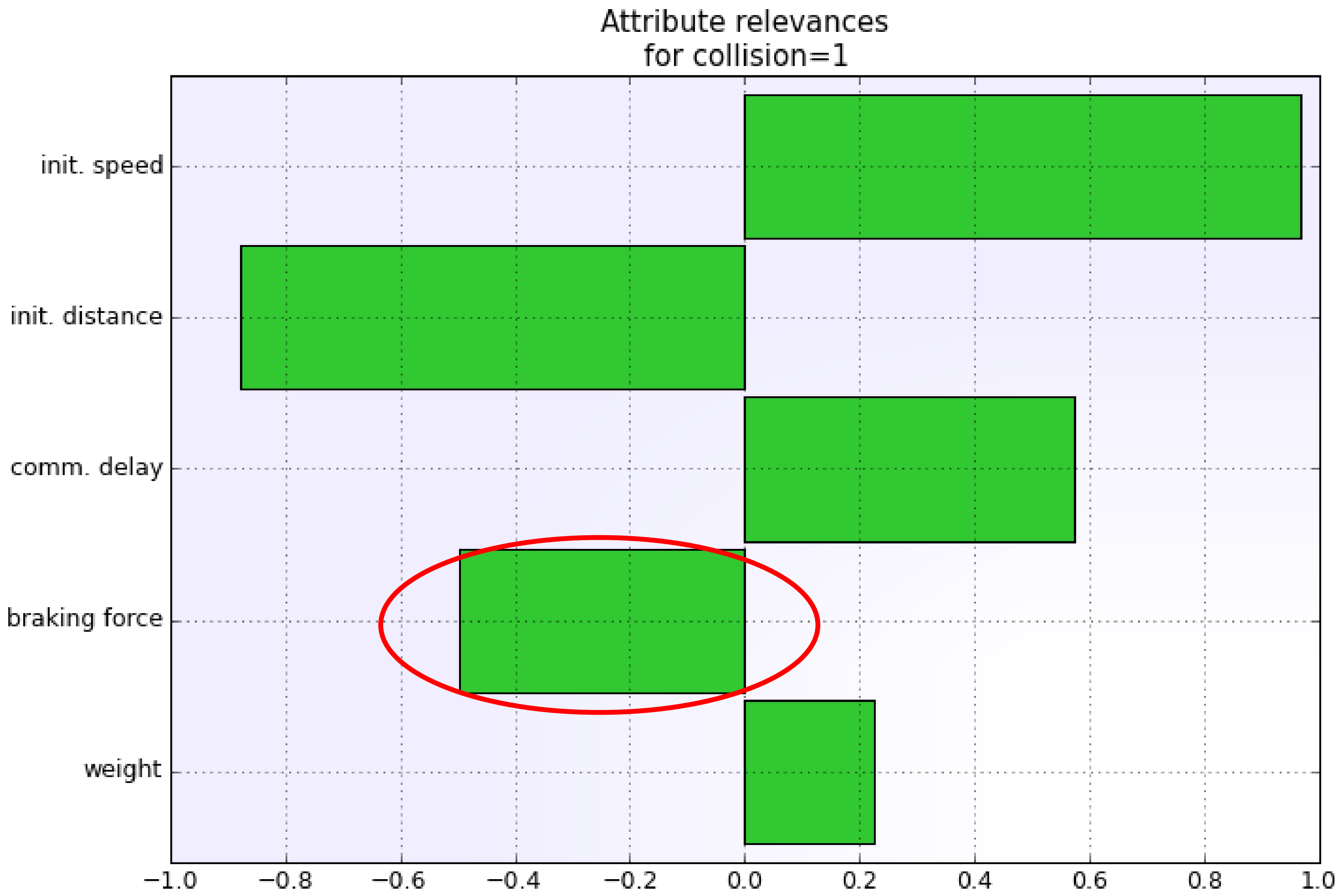
	Forecast		
Output	**0**	**1**	**Total**
0	**5021 (84.5998315...**	914 (15.4001684920...	5935 (49.458333333...
1	862 (14.2126957955...	**5203 (85.7873042...**	6065 (50.541666666...
Total	5883 (49.025000000...	6117 (50.975000000...	**12000 (100%)**





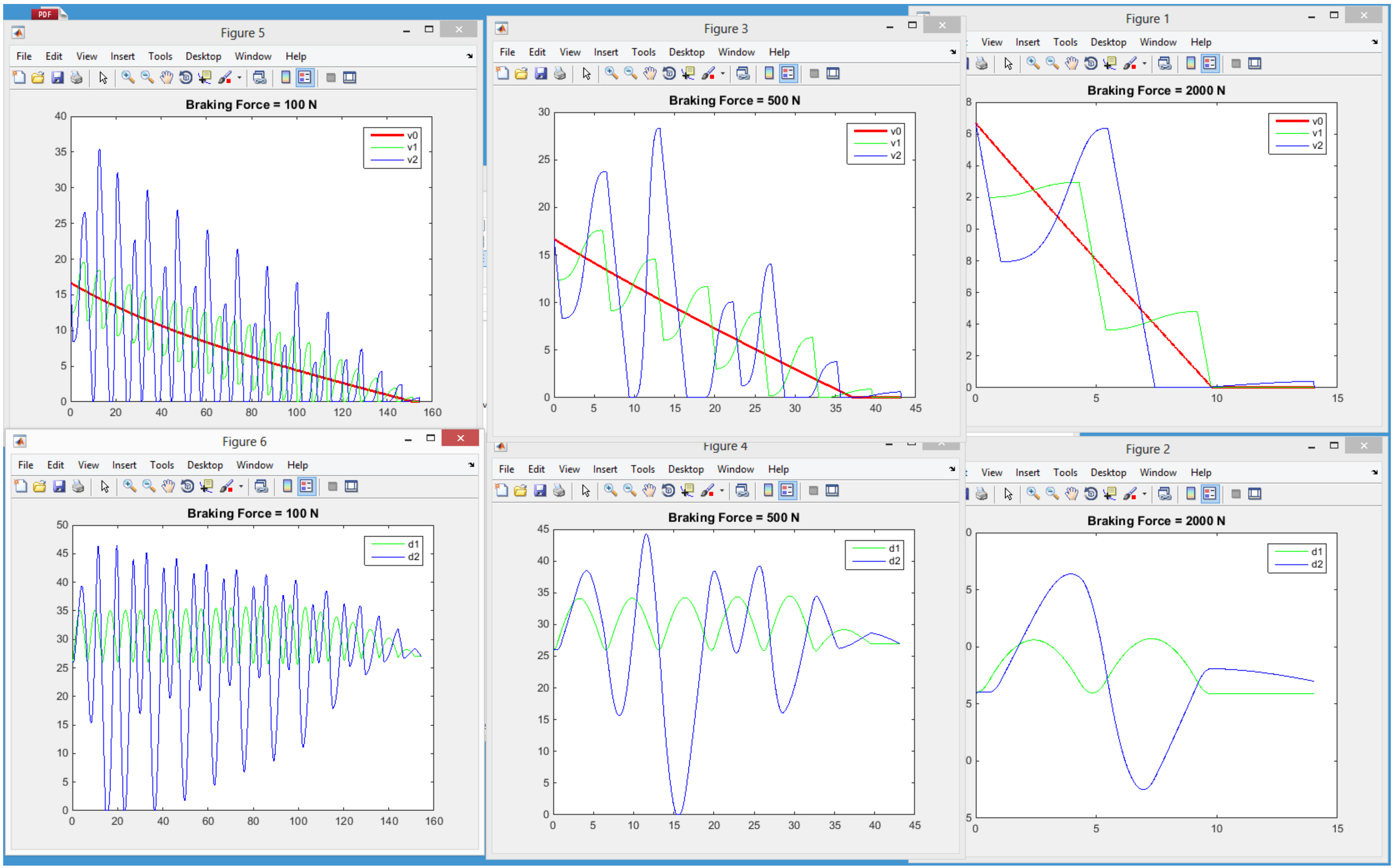
Information on delay is not crucial!

Feature ranking



Decreasing braking force -> more collisions, why?

Rationale of decreasing braking force -> more collisions



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))
  - UC3 (V2V)
  - UC5 (V2I).

- Future work: online data analytics.

THANK YOU

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.

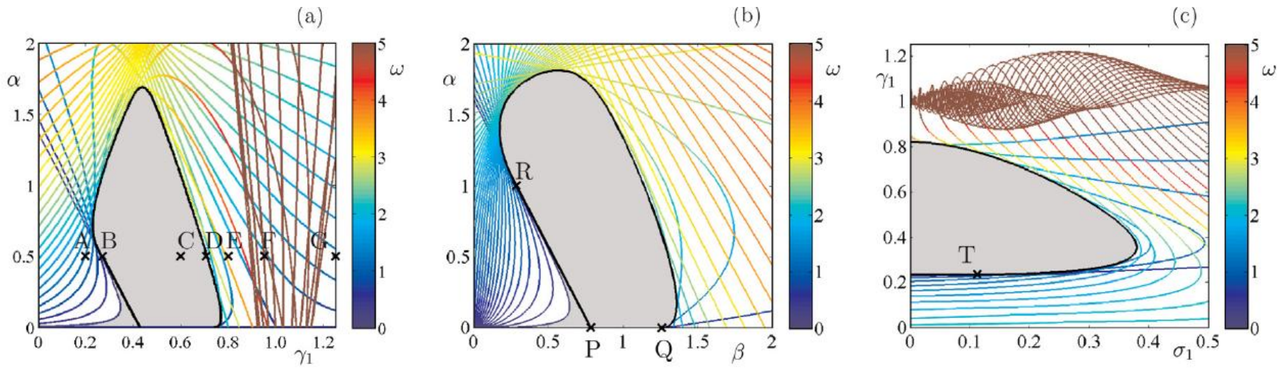
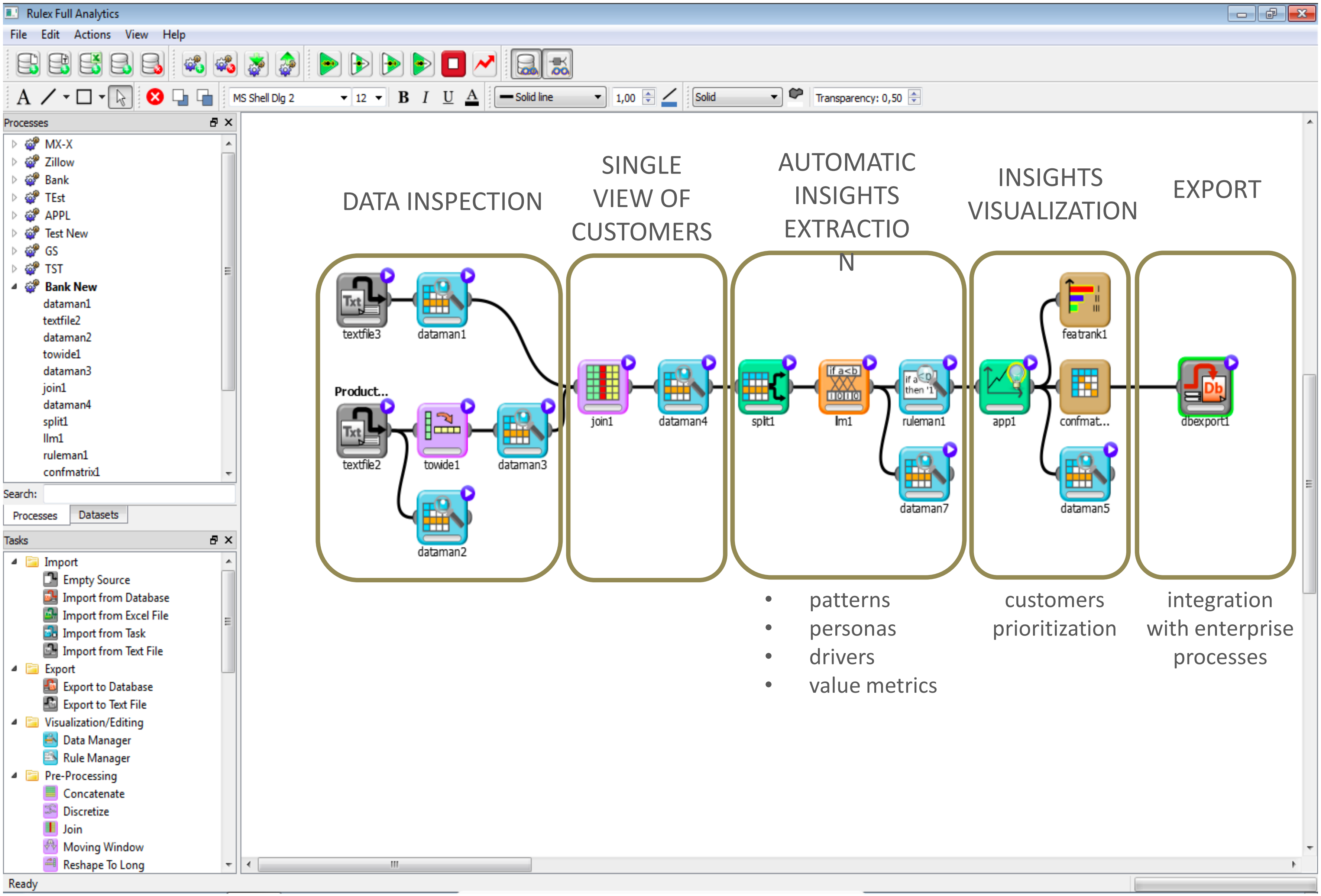


Fig. 11. (a) Stability chart in the (γ_1, α)-plane for the ring configuration using $N = 33$ vehicles and the same parameters as in Fig. 2(e). (b and c) Stability

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.

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