MACHINE LEARNING TECHNIQUES FOR INTRUSION DETECTION IN SCADA SYSTEMS

Author:
Rocío LÓPEZ PÉREZ

Supervisor:
Prof. Dr. Thomas ENGEL

Reviewer:
Prof. Dr. Ulrich SORGER

Advisors:
Dr. Ridha SOUA
Dr. Florian ADAMSKY
Dr. Andriy PANCHENKO

AGENDA

- Problem Statement
- Background
  - Introduction to SCADA systems
  - SCADA systems in the spotlight of cyber attacks
  - Machine learning techniques for Network Intrusion Detection Systems
- Gas pipeline SCADA system dataset
  - Analysis of the dataset
  - Data mining tasks
  - Experiments
  - Results & Comparison
- Conclusion & Future Work

PROBLEM STATEMENT

- Critical systems use isolation as security strategy
- This is unrealistic in an increasingly connected world
- Many SCADA systems do not use the air gap strategy anymore
- Networks demand more elaborate measures of protection
- Machine learning techniques are suitable for NIDS

BACKGROUND

SCADA - *Supervisory Control and Data Acquisition*

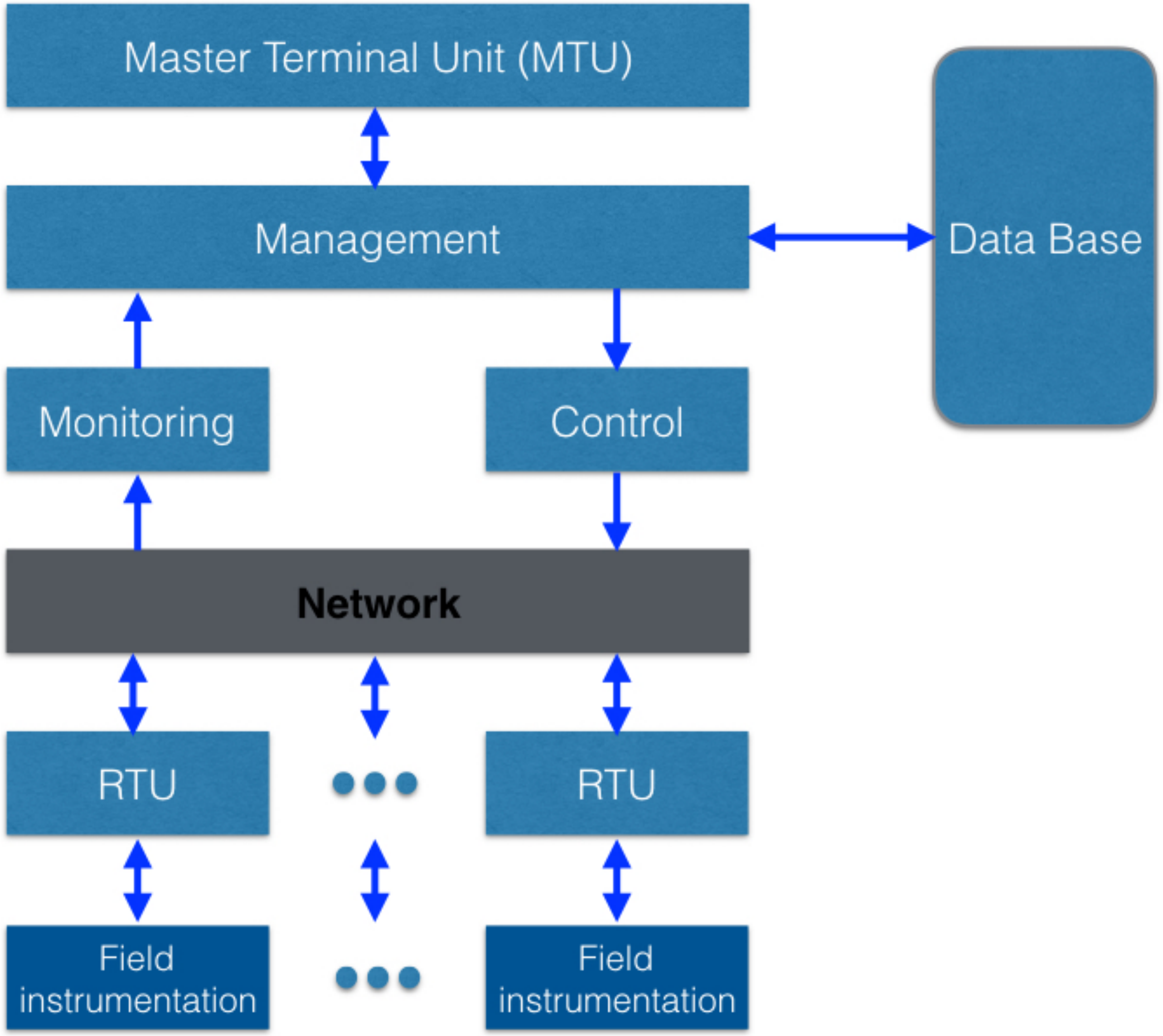


Figure 1: SCADA system architecture

SCADA SYSTEMS IN THE SPOTLIGHT OF CYBER ATTACKS

- Traditional SCADA systems communicated over serial analog circuits
- Modern SCADA systems increased their interconnectivity
- Deployment of IP communication protocols
- Additional level of connectivity
- Malicious network packets can reach the system from anywhere

SCADA SYSTEMS IN THE SPOTLIGHT OF CYBER ATTACKS

- Gazprom gas plant (April 1999, Russia)
- Davis-Besse nuclear power plant (January 2003, Ohio)
- Stuxnet (January 2010, Iran)
- The first electric blackout (December 2015, Ukraine)

MACHINE LEARNING TECHNIQUES FOR NIDS

- NIDS - mechanism that silently listens to network traffic to detect anomalies or suspicious activities
- ML - if a computer goes through an experience, and through that experience learns to do that task better
- Presented techniques:
  - Support Vector Machine
  - Random Forests
  - Long Short Term Memory

SVM

- Linear machine for pattern classification
- Kerneled learning algorithm (RBF)
- Feed-forward neural network

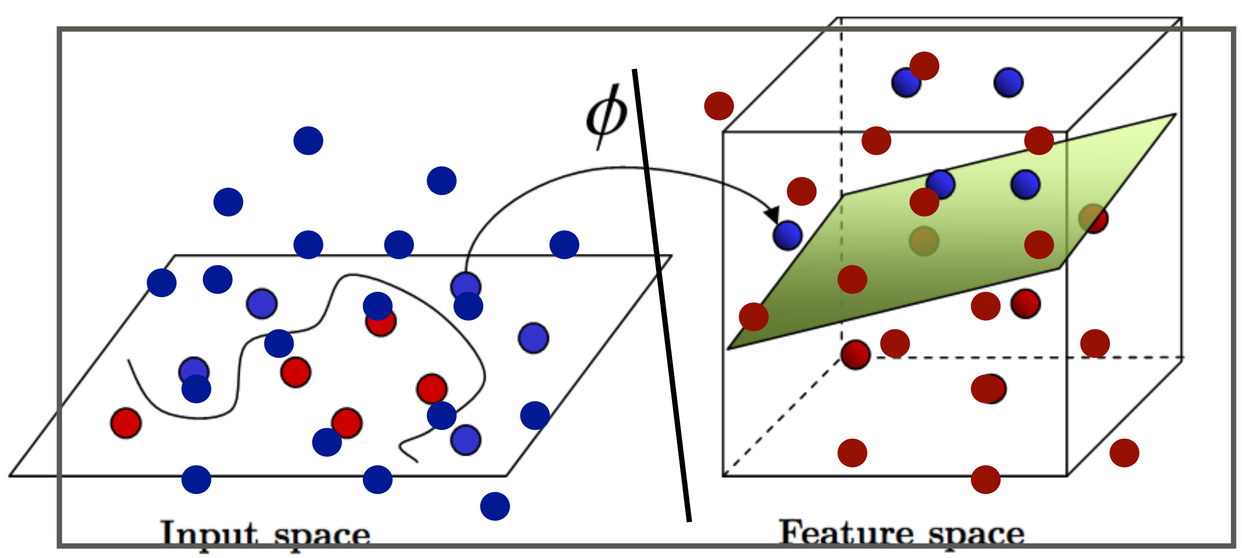


Figure 2: Optimal Hyperplane for Linearly Separable Patterns

Figure 3: Transformation from input space to feature space

HYPER-PARAMETERS SVM

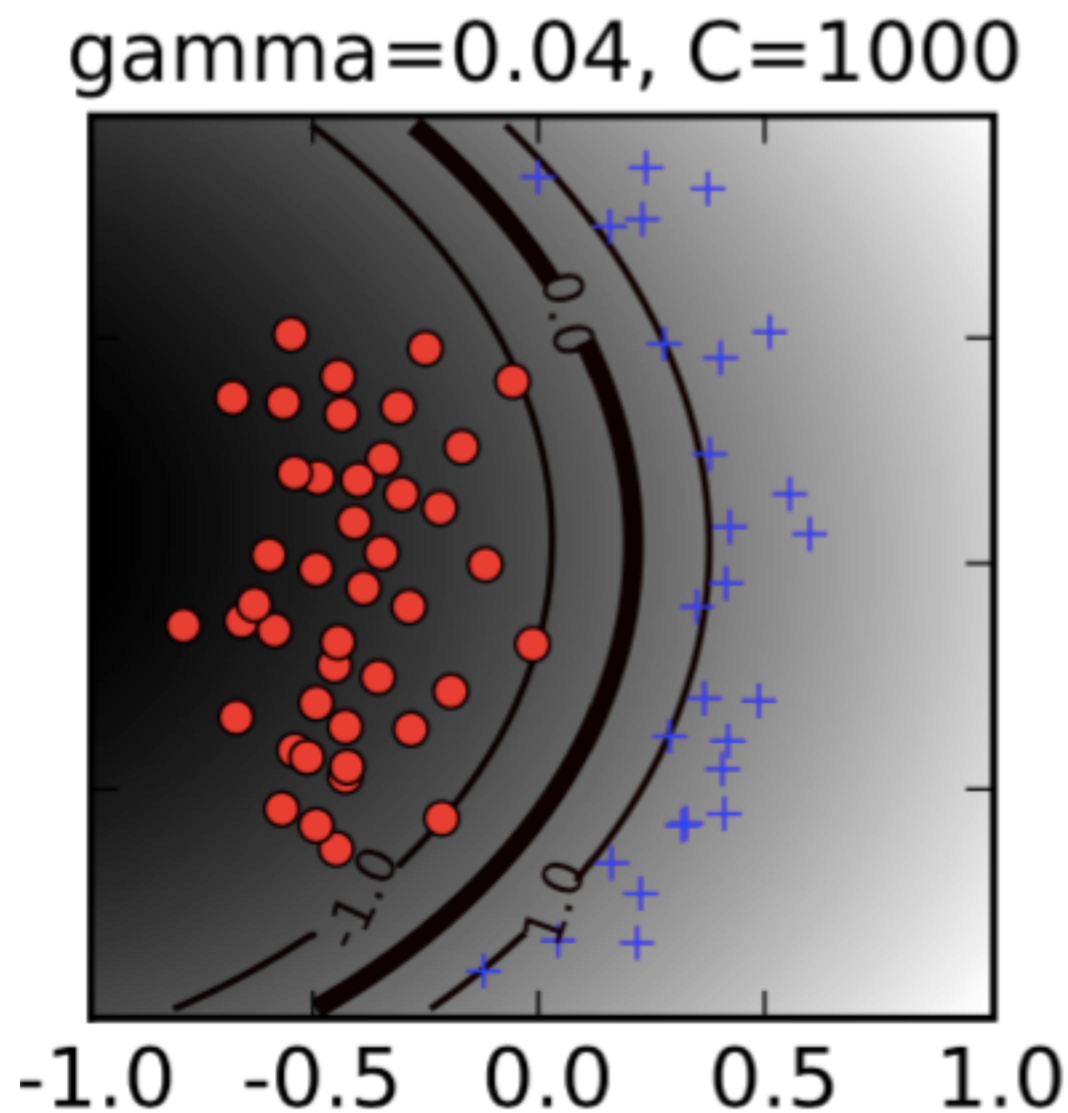


Figure 6: Decision boundary shape depending on C and γ hyper-parameters values

RANDOM FOREST

- Based in the decision tree technique
- Improves classification accuracy by incorporating randomness (e.g. bagging, max-features per split)
- Easy and fast algorithms
- The split criterion is the *entropy measure* of a feature

$$E(v_i) = -\sum_{j=1}^{k} p_j log_2(p_j)$$

HYPER-PARAMETERS RANDOM FOREST

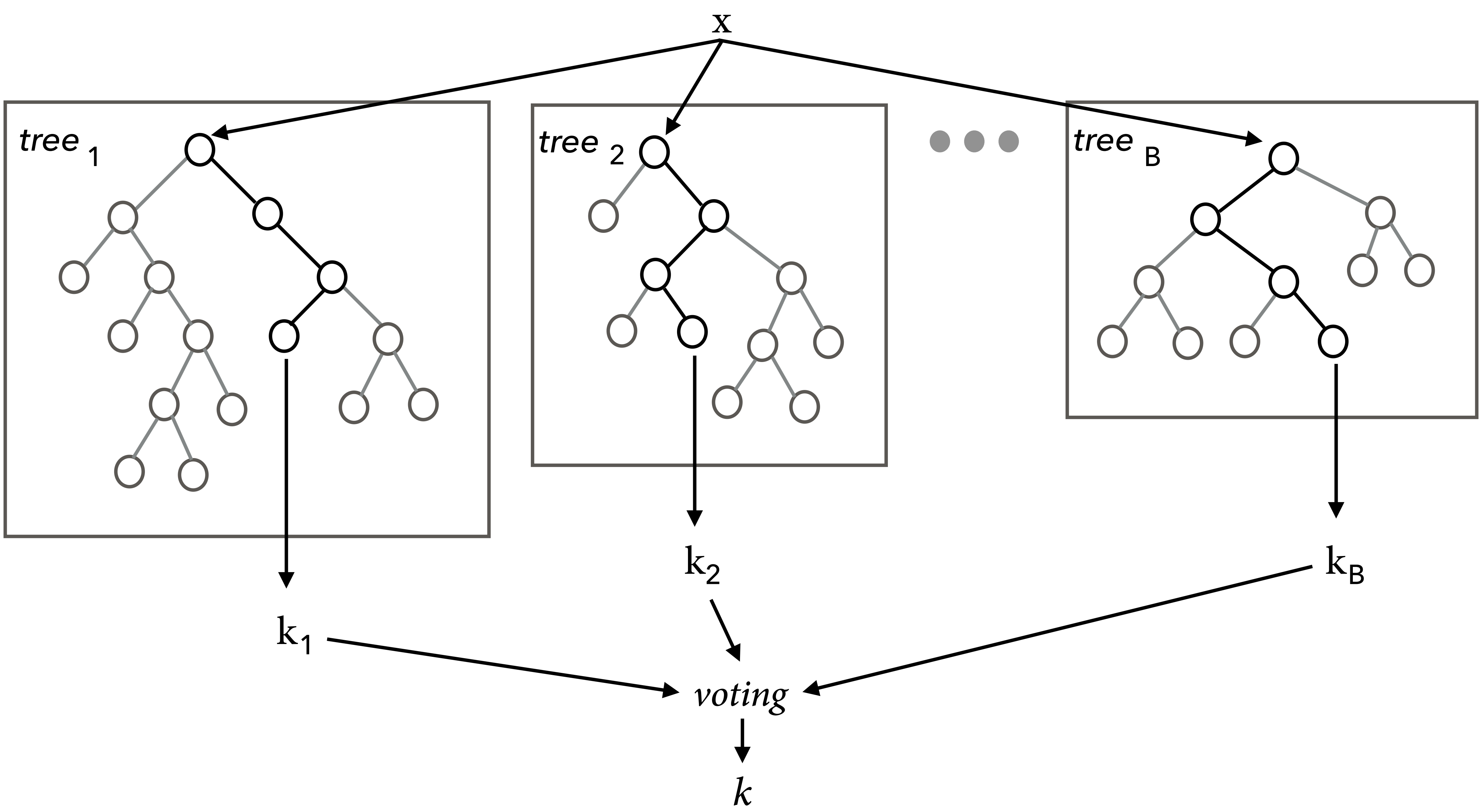
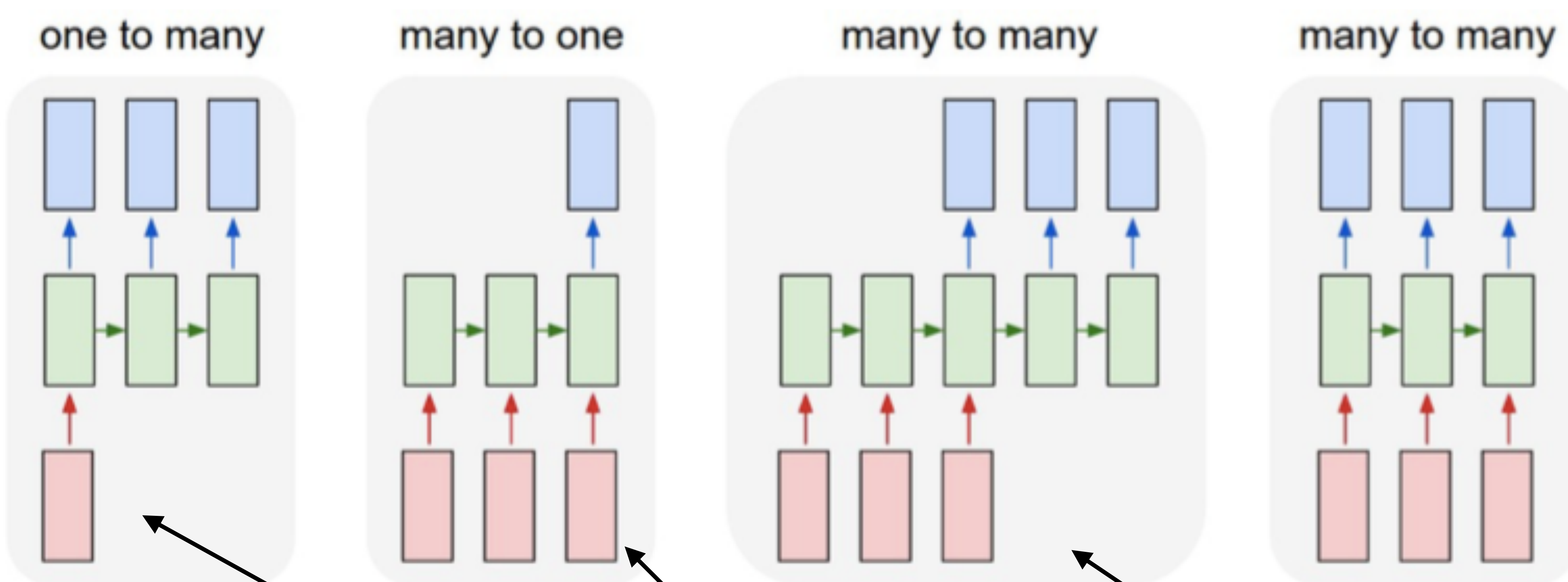


Figure 7: Number of estimators & Maximal tree depth

LSTM

- Recurrent Neural Network
- Composed of memory blocks containing memory cells



e.g. **Image Captioning**
image –> seq. of words

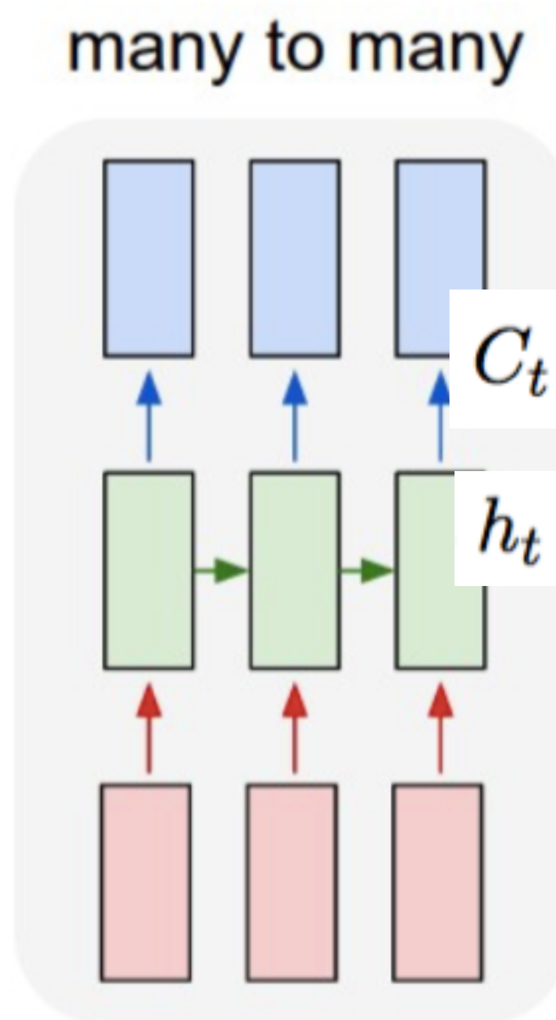
e.g. **Sentiment Classification**
seq. of words –>sentiment

e.g. **Machine Translation**
seq. of words –> seq. of words

e.g. **Video classification on frame level**

Figure 8: LSTM - different architectures

LSTM



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$h_t = o_t * tanh(\tilde{C}_t)$$

Figure 9: LSTM - computational operations within a cell

HYPER-PARAMETERS LSTM

- Learning rate
- Sequence length
- Dropout rate
- Hidden layer length

METHODOLOGY

SCADA DATASET

- Gas pipeline system provided by the Mississippi State University's in-house SCADA lab
  - 274628 instances
  - 17 features
  - Binary labels and categorical labels

	Attack type	Acronym	#	Category
1	Naive Malicious Response Injection	NMRI	7753	response injection
2	Complex Malicious Response Injection	CMRI	13035	response injection
3	Malicious State Command Injection	MSCI	7900	command injection
4	Malicious Parameter Command Injection	MPCI	20412	command injection
5	Malicious Function Code Injection	MFCI	4898	command injection
6	Denial of Service	DoS	2176	denial of service
7	Reconnaissance	Recon	3874	reconnaissance

Table 1: Types and categories of attacks

SCADA DATASET

Figure 10: dataset experiment pipeline

DATA ANALYSIS

	Features	Type		Features	Type
1	address	Network	11	control scheme	Command Payload
2	function	Command Payload	12	pump	Command Payload
3	length	Network	13	solenoid	Command Payload
4	setpoint	Command Payload	14	pressure measurement	Response Payload
5	gain	Command Payload	15	crc rate	Network
6	reset rate	Command Payload	16	command response	Network
7	deadband	Command Payload	17	time	Network
8	cycle time	Command Payload	18	binary result	Label
9	rate	Command Payload	19	categorized result	Label
10	system mode	Command Payload	20	specific result	Label

Table 2: Gas pipeline dataset – feature list

Gas Pipeline Dataset - Packet features						
addr	funct	length	payload	crc	c/r	time stamp
4,	3,	16,	?,?,?,?,?,?,?,?,?,?,?,	12869,	1,	1418682163.170388
4,	3,	46,	?,?,?,?,?,?,?,?,?,?,0.689655,	12356,	0,	1418682163.269946
4,	16,	90,	10,115,0.2,0.5,1,0,0,1,0,0,?,	17219,	1,	1418682164.99559

Table 3: Missing values in the gas pipeline packets

SCADA DATASET

Figure 10: dataset experiment pipeline

DATA MINING TASKS

- Dealing with missing values (4 approaches):
  - Clustering the payloads with GMM
  - Clustering the payloads with K-means
  - Zeros imputation & indicators
  - Imputing missing values by keeping the prior existing value
- Data normalization (2 approaches)
  - Mean & Std deviation
  - Min-Max

DEALING WITH MISSING VALUES

Clustering techniques - Gaussian Mixture Model		
raw payload	preprocessed payload	#k = 9
?,?,?,?,?,?,?,?,?,?,?,?	1,0,0,0,0,0,0,0,0	#k_tp1=1
?,?,?,?,?,?,?,?,?,?,?,0.689655	0,0,1,0,0,0,0,0,0	#k_tp2=2
10,115,0.2,0.5,1,0,0,1,0,0,?	0,0,0,1,0,0,0,0,0	#k_tp3=6
3.06,117,0.3372,0.46,1,0,2,1,0,0,?	0,0,0,0,1,0,0,0,0	#k_tp3=6
?,?,?,?,?,?,?,?,?,?,?,1.91862e-38	0,1,0,0,0,0,0,0,0	#k_tp2=2
12.2,117,0.3471,0.71,1,0,0,1,0,0,?	0,0,0,0,1,0,0,0,0	#k_tp3=6
?,?,?,?,?,?,?,?,?,?,?,0.528736	0,0,1,0,0,0,0,0,0	#k_tp2=2
11.4,117,0.2255,0.45,0.56,0,0,0,1,1,?	0,0,0,0,0,0,1,0,0	#k_tp3=6

Zeros imputation & indicators	
raw payload	preprocessed payload
?,?,?,?,?,?,?,?,?,?,?,?,	0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1
?,?,?,?,?,?,?,?,?,?,?,0.689655,	0,0,0,0,0,0,0,0,0,0,0.689655,1,1,1,1,1,1,1,1,1,1,0
10,115,0.2,0.5,1,0,0,1,0,0,?,	10,115,0.2,0.5,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1

Keep prior value	
raw payload	preprocessed payload
?,?,?,?,?,?,?,?,?,?,?,0.689655	10,115,0.2,0.5,1,0,0,1,0,0,0.689655
10,115,0.2,0.5,1,0,0,1,0,0,?	10,115,0.2,0.5,1,0,0,1,0,0,0.689655
?,?,?,?,?,?,?,?,?,?,?,?	10,115,0.2,0.5,1,0,0,1,0,0,0.689655
?,?,?,?,?,?,?,?,?,?,?,0.666667	10,115,0.2,0.5,1,0,0,1,0,0,0.666667

Table 4: Different approaches to deal with missing values

SCADA DATASET

https://github.com/rocionightwater/ML-techniques-for-NIDS

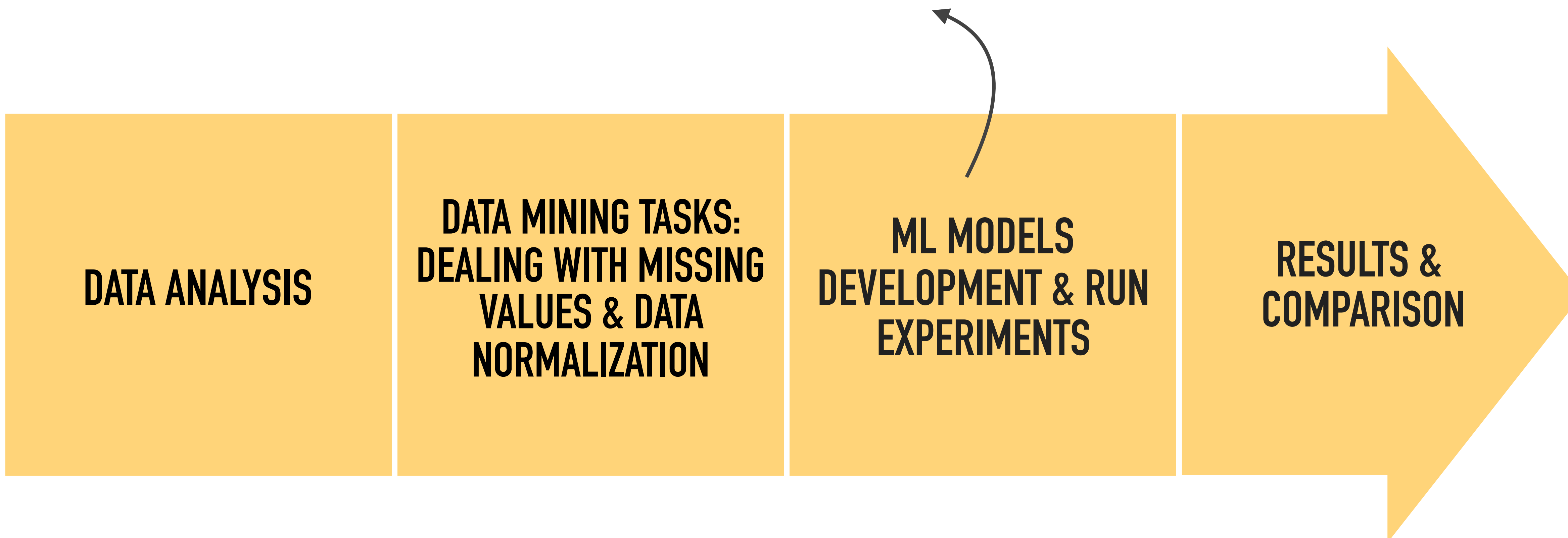


Figure 10: dataset experiment pipeline

EXPERIMENTS

- 16 datasets
- SVM, RFs and LSTM classification models
- Training set 60% (164776 instances), validation set 20% (54926 instances) and test set 20% (54926 instances)

$$accuracy = \frac{truePositive + trueNegative}{truePositive + trueNegative + falsePositive + falseNegative}$$

Figure 15: measurements to evaluate the classifier accuracy

EXPERIMENTS – SVM

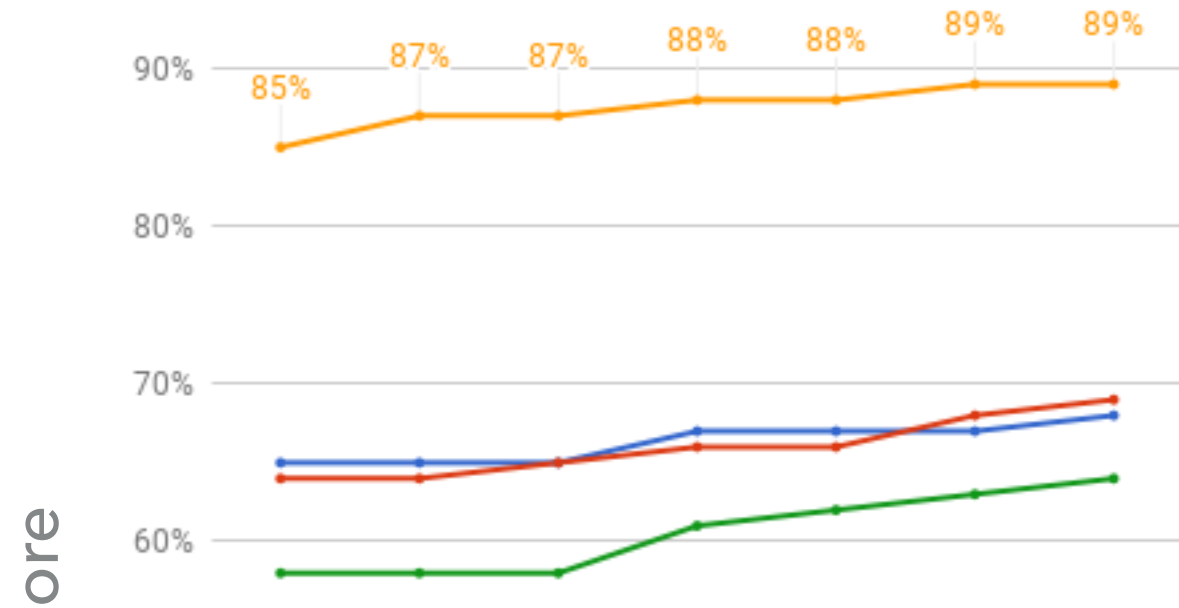


Fig11: SVM - binary & mean normalization

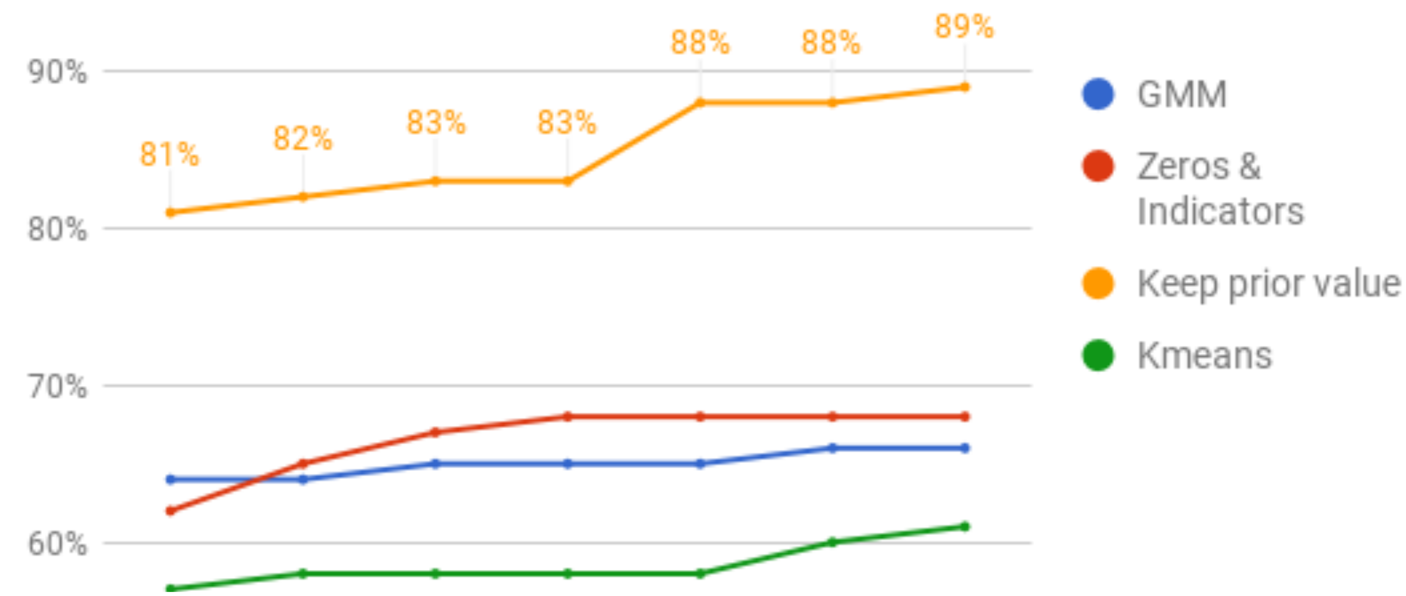


Fig12: SVM - binary & minmax normalization

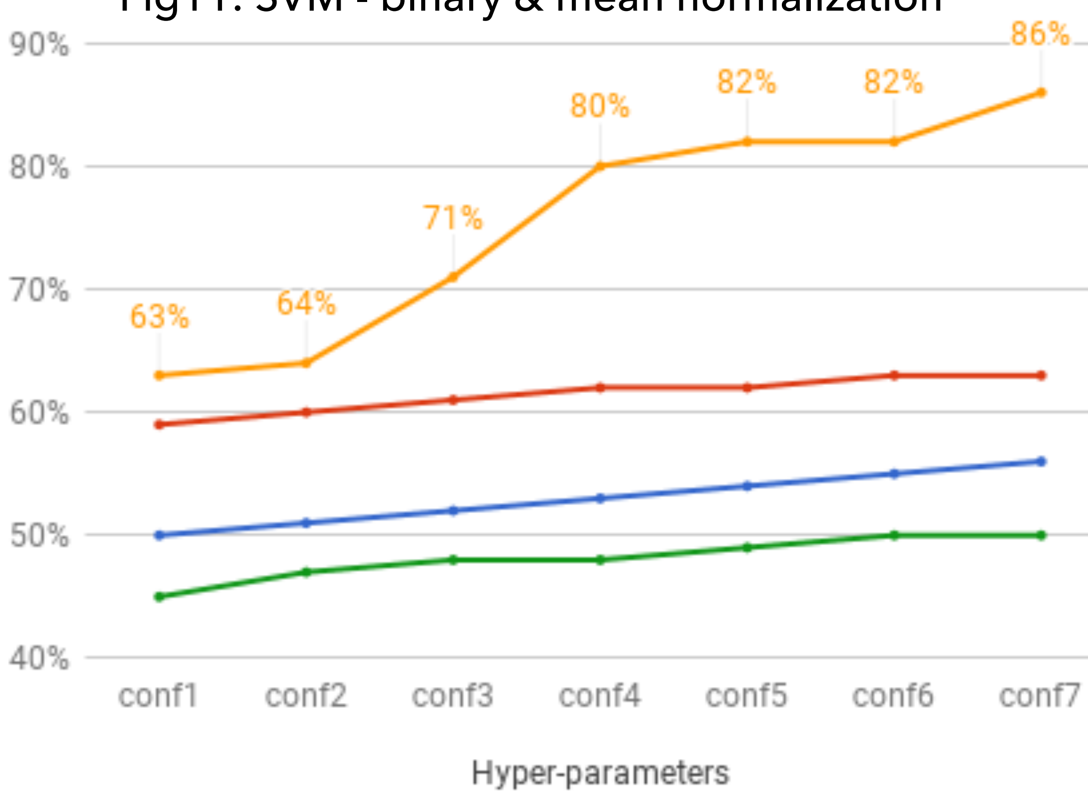


Fig13: SVM - category & mean normalization

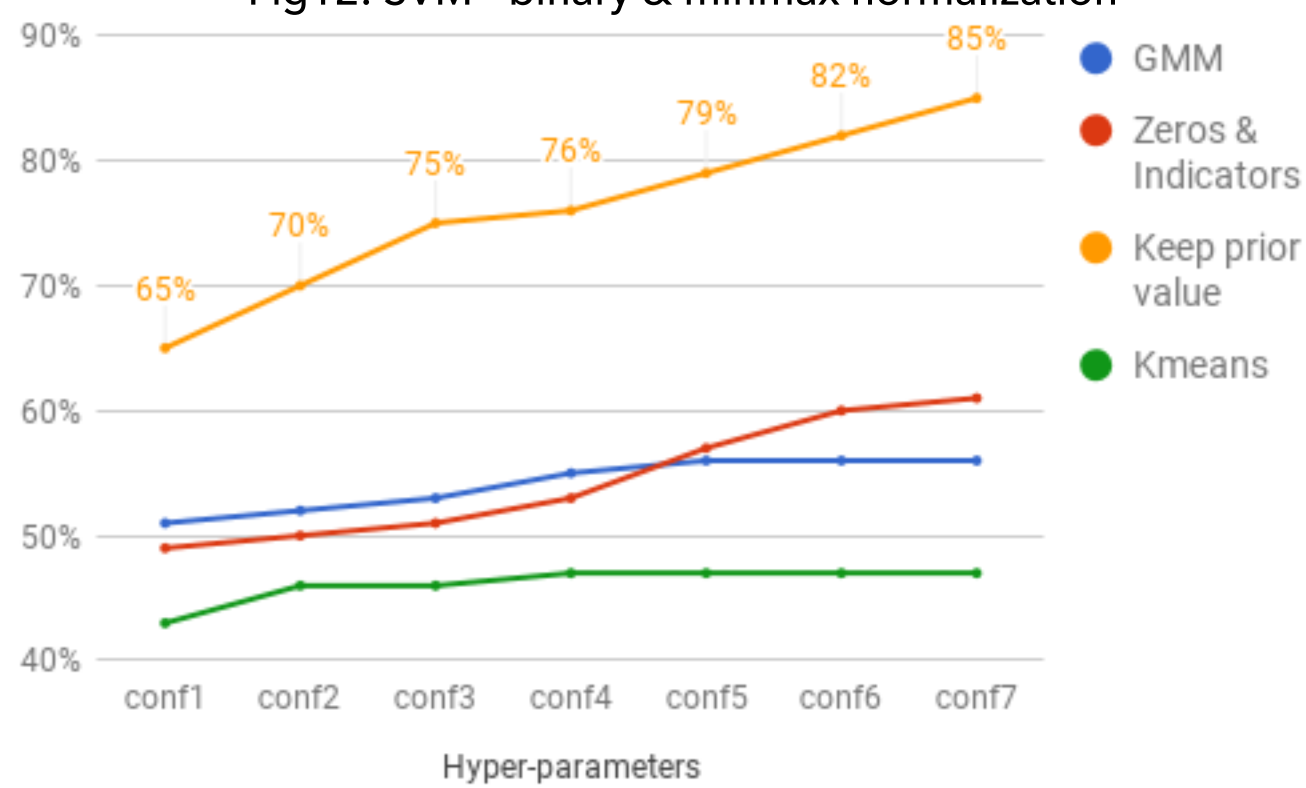


Fig14: SVM - category & minmax normalization

EXPERIMENTS – RF

f1-score

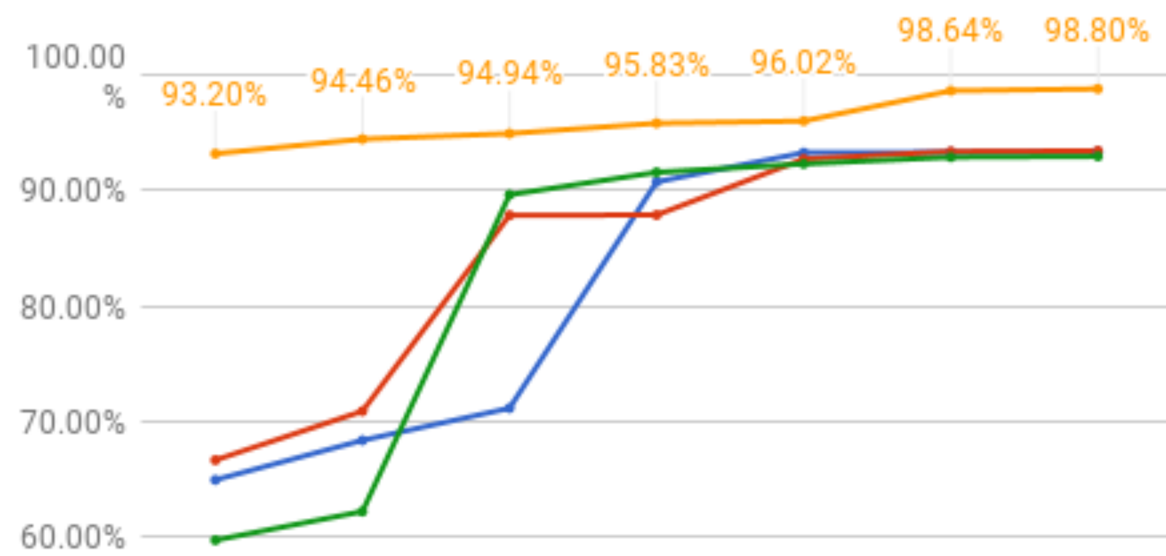


Fig15: RF - binary & mean normalization

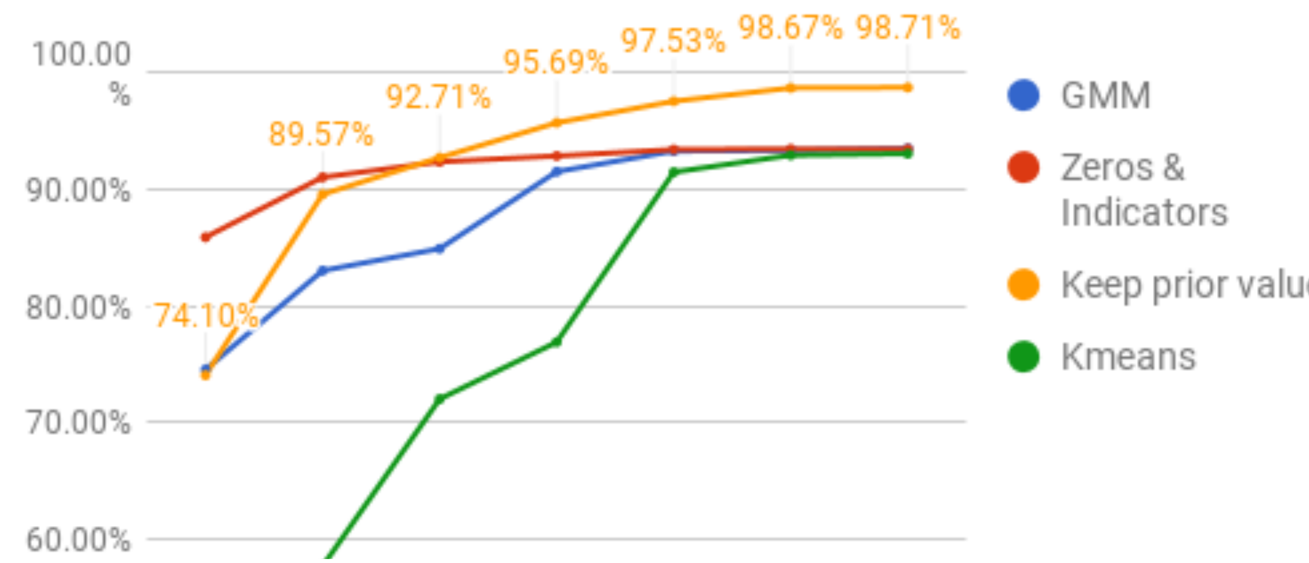


Fig16: RF - binary & minmax normalization

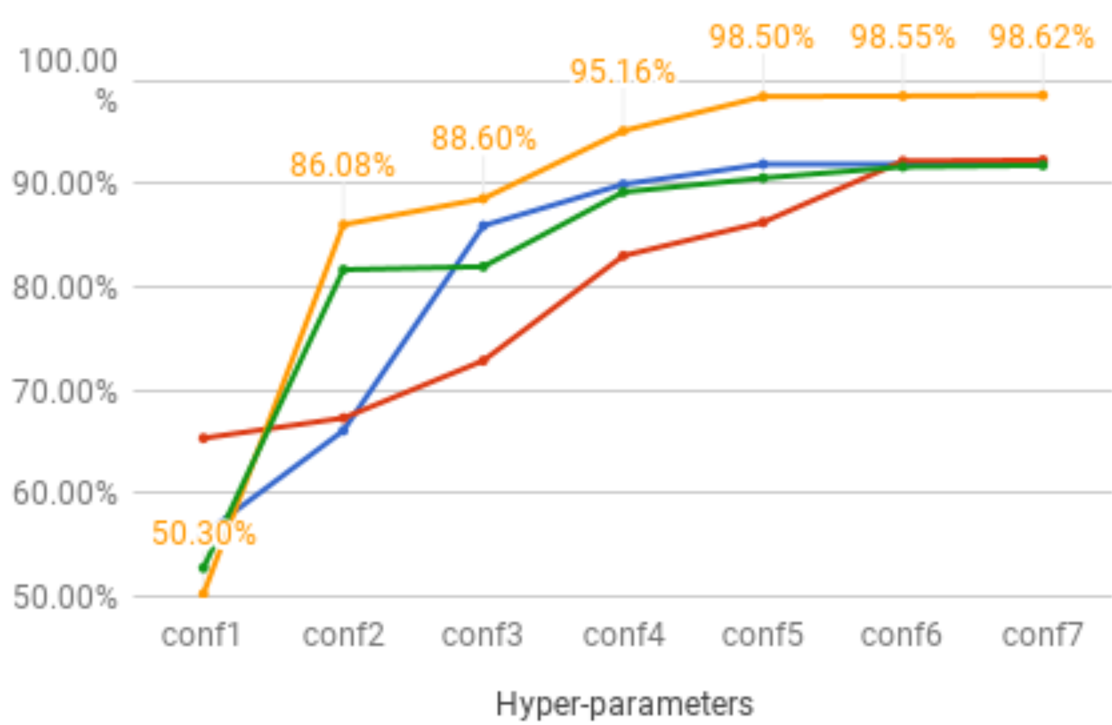


Fig17: RF - category & mean normalization

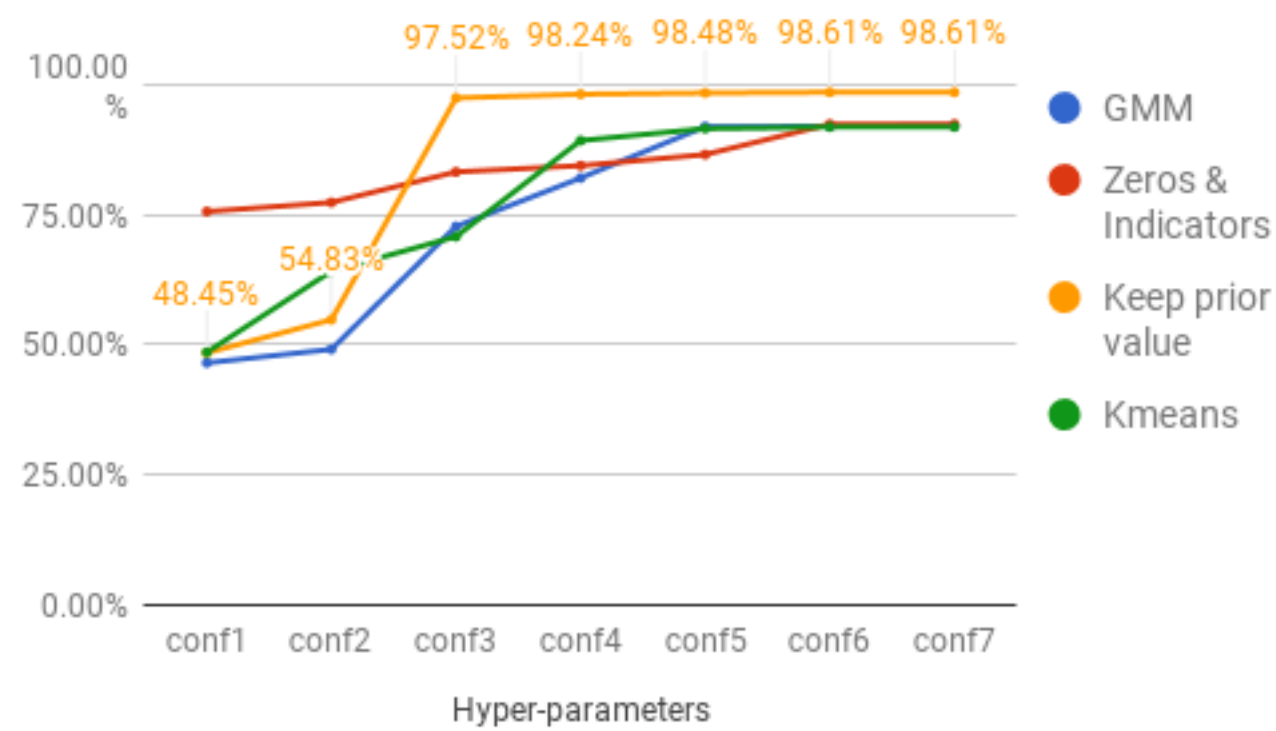


Fig18: RF - category & minmax normalization

EXPERIMENTS – LSTM

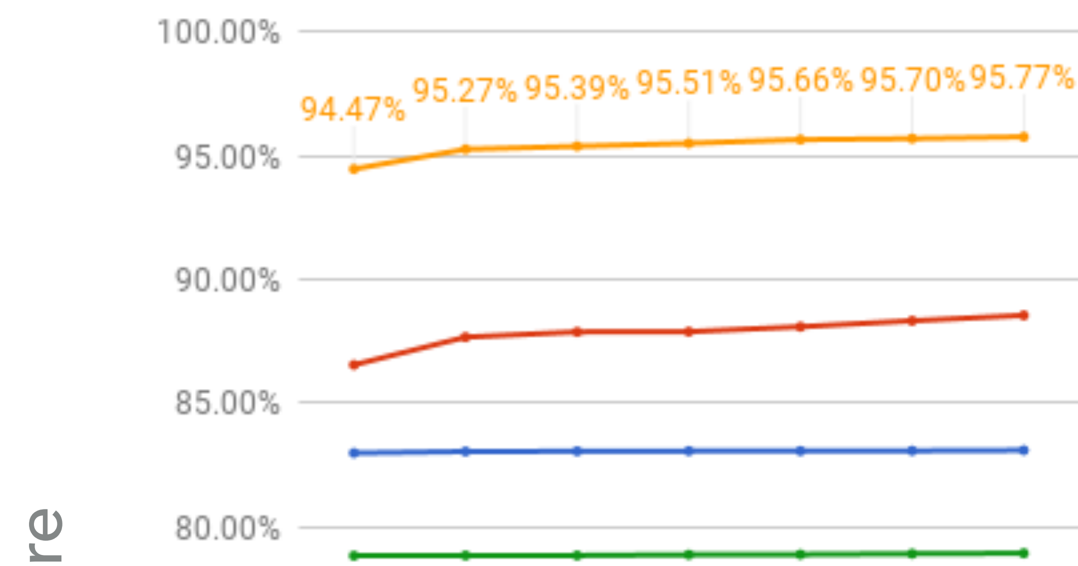


Fig19: LSTM - binary & mean normalization

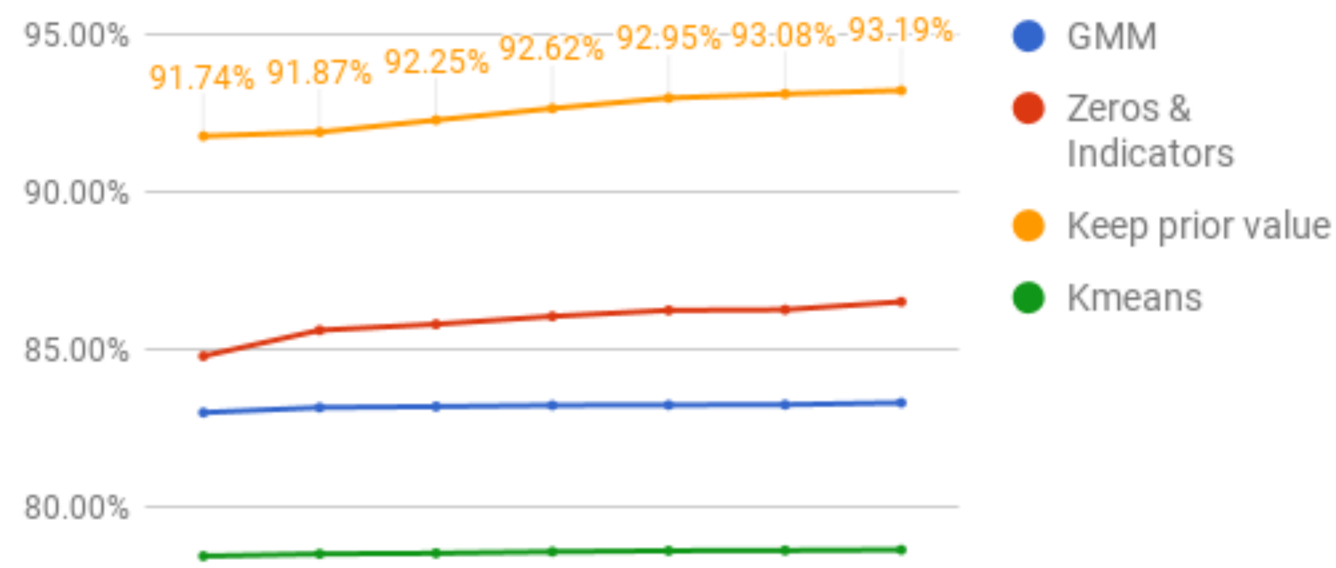


Fig20: LSTM - binary & minmax normalization

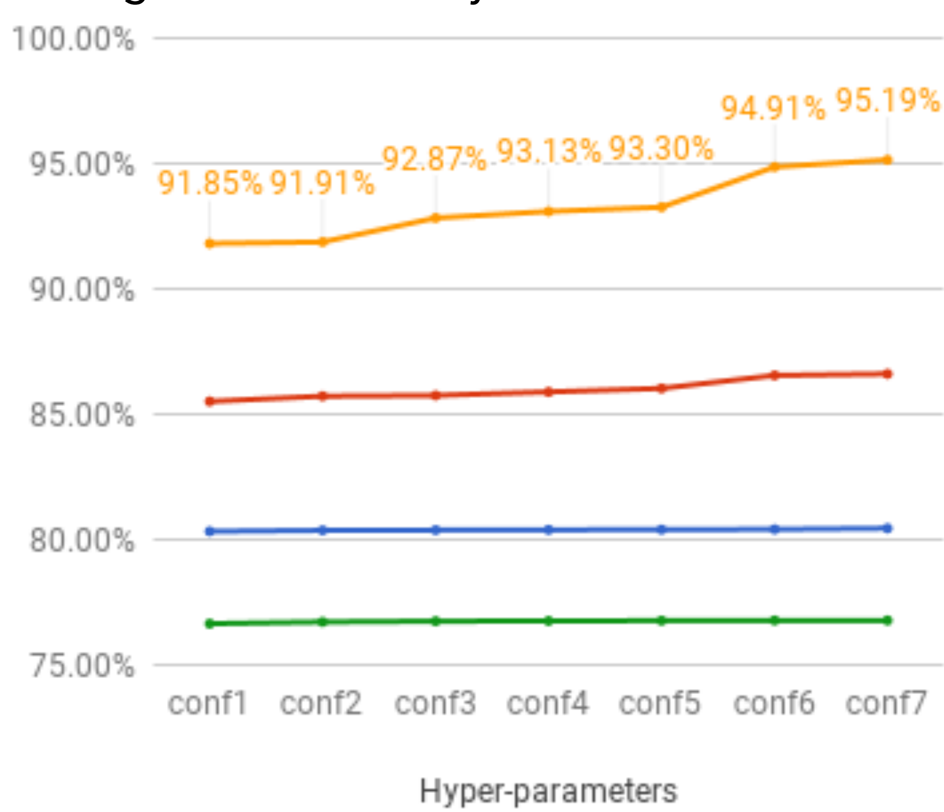


Fig21: LSTM - categorical & mean normalization

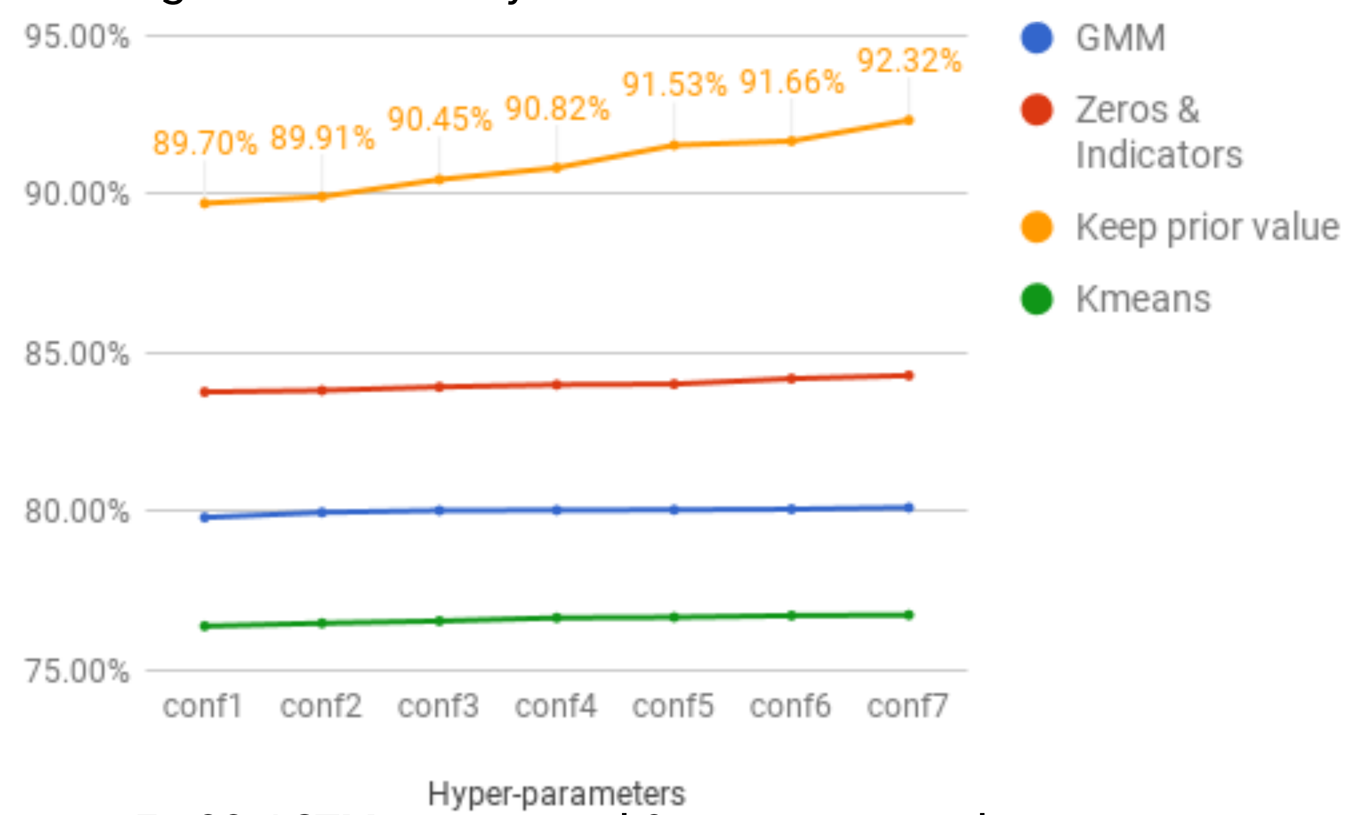


Fig22: LSTM - categorical & minmax normalization

SCADA DATASET

Figure 10: dataset experiment pipeline

RESULTS

SVM	Hyper-parameters		Measurements			
Test sets	C	gamma	Acc	Prec	Recall	F1-score
binary-mean-keep	**252.253**	**0.0434**	**0.9489**	**0.9489**	**0.9488**	**0.9488**
binary-minmax-keep	639.375	0.3298	0.9466	0.9501	0.9467	0.9469
categorical-mean-keep	**62.837**	**0.4025**	**0.9577**	**0.9576**	**0.9577**	**0.9575**
categorical-minmax-keep	58.629	0.3014	0.8920	0.9130	0.8921	0.8988

Random Forest	Hyper-parameters		Measurements			
Test sets	ne	md	Acc	Prec	Recall	F1-score
binary-mean-keep	**45**	**57**	**0.9954**	**0.9954**	**0.9954**	**0.9954**
binary-minmax-keep	41	43	0.9953	0.9953	0.9953	0.9952
categorical-mean-keep	**59**	**29**	**0.9948**	**0.9948**	**0.9948**	**0.9948**
categorical-minmax-keep	70	30	0.9947	0.9947	0.9947	0.9947

LSTM	Hyper-parameters					Measurements			
Test sets	lr	batch	seq	drop	h_layer	Acc	Prec	Recall	F1-score
binary-mean-keep	**0.00805**	**77**	**4**	**0.19019**	**151**	**0.9689**	**0.9688**	**0.9689**	**0.9686**
binary-minmax-keep	0.0100	73	4	0.2096	112	0.9517	0.9515	0.9518	0.9506
categorical-mean-keep	**0.0100**	**127**	**4**	**0.0982**	**229**	**0.9658**	**0.9652**	**0.9658**	**0.9651**
categorical-minmax-keep	0.01426	123	4	0.0680	76	0.9388	0.9367	0.9388	0.9358

Table 5: Best binary and categorical classifiers modeled with SVM, RFs and LSTM

RESULTS

Random Forest	Accuray test data = 0.9872			
Type of Data	**precision**	**recall**	**f1-score**	**support**
Normal	99.46%	99.96%	99.71%	42818
NMRI	98.92%	96.76%	97.83%	1605
CMRI	99.19%	97.10%	98.13%	2515
MSCI	99.75%	97.60%	98.67%	1628
MPCI	99.95%	98.13%	99.03%	4177
MFCI	99.60%	100%	99.80%	993
DoS	99.30 %	95.94%	97.59%	443
Recon	99.86%	98.93%	99.39%	747
avg / total	**99.48%**	**99.48%**	**99.48%**	**54926**

Normal	NMRI	CMRI	MSCI	MPCI	MFCI	DoS	Recon	
42801	1	8	3	1	0	3	1	**Normal**
40	**1553**	12	0	0	0	0	0	**NMRI**
57	16	**2442**	0	0	0	0	0	**CMRI**
38	0	0	**1589**	1	0	0	0	**MSCI**
78	0	0	0	**4099**	0	0	0	**MPCI**
0	0	0	0	0	**993**	0	0	**MFCI**
17	0	0	0	0	0	**425**	0	**DoS**
4	0	0	0	0	4	0	**739**	**Recon**

Table 6: Random Forest-Classification matrix (above) and confusion matrix (below)

RELATED WORK

Dataset	PART		Random Forest	
Category	**Precision**	**Recall**	**Precision**	**Recall**
Normal	0.90	0.99	0.95	0.99
NMRI	0.58	0.74	0.81	0.72
CMRI	0.84	**0.41**	0.77	**0.67**
MSCI	0.82	**0.32**	0.91	**0.72**
MPCI	0.93	**0.44**	0.99	**0.84**
MFCI	0.99	1.00	0.98	1.00
DoS	1.00	**0.45**	0.80	**0.75**
Recon	1.00	0.92	1.00	0.97
Weighted Avg.	0.89	0.89	**0.94**	**0.94**

Table 7: Categorical classifiers – comparison between a related work & our work

CONCLUSIONS

- Preparation of sixteen preprocessed datasets by applying different interesting data mining techniques
- Use of machine learning algorithms to implement diverse NIDS classifiers
- Correct use of test set accuracy measurements
- Random Forest has given us excellent results (benign data recall = 99.97%, attacks recall = 98.04%; overall detection rate (recall) = 99.54%)

FUTURE WORK

- LSTM algorithm – worth further investigation in future research
- Extraction of rules from RFs to integrate them with signature-based NIDS (e.g. Snort)

THANK YOU VERY MUCH!

Q&A

https://github.com/rocionightwater/ML-techniques-for-NIDS

DATASET - SEQUENCE LENGTH LSTM

```
4,16,90,10,115,0.2,0.5,1,0,0,1,0,0,?,17219,1,1418682164.995592,0,0,0
4,16,16,?,?,?,?,?,?,?,?,?,?,?,17718,0,1418682165.146975,0,0,0
4,3,16,?,?,?,?,?,?,?,?,?,?,?,12869,1,1418682166.785678,0,0,0
4,3,46,?,?,?,?,?,?,?,?,?,?,0.666667,14393,0,1418682166.870868,0,0,0

4,16,90,10,115,0.2,0.5,1,0,0,1,0,0,?,17219,1,1418682168.649917,0,0,0
4,16,16,?,?,?,?,?,?,?,?,?,?,?,17718,0,1418682168.792187,0,0,0
4,3,16,?,?,?,?,?,?,?,?,?,?,?,12869,1,1418682170.439552,0,0,0
4,3,46,?,?,?,?,?,?,?,?,?,?,0.701149,17221,0,1418682170.515108,0,0,0

4,16,90,10,115,0.2,0.5,1,0,0,1,0,0,?,17219,1,1418682172.264295,0,0,0
4,16,16,?,?,?,?,?,?,?,?,?,?,?,17718,0,1418682172.424168,0,0,0
4,3,16,?,?,?,?,?,?,?,?,?,?,?,12869,1,1418682174.093794,0,0,0
4,3,46,?,?,?,?,?,?,?,?,?,?,0.689655,12355,0,1418682174.182478,0,0,0
```

CONFUSION OR ERROR MATRIX

		Predicted class	
		P	**N**
Actual class	**P**	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

SVM

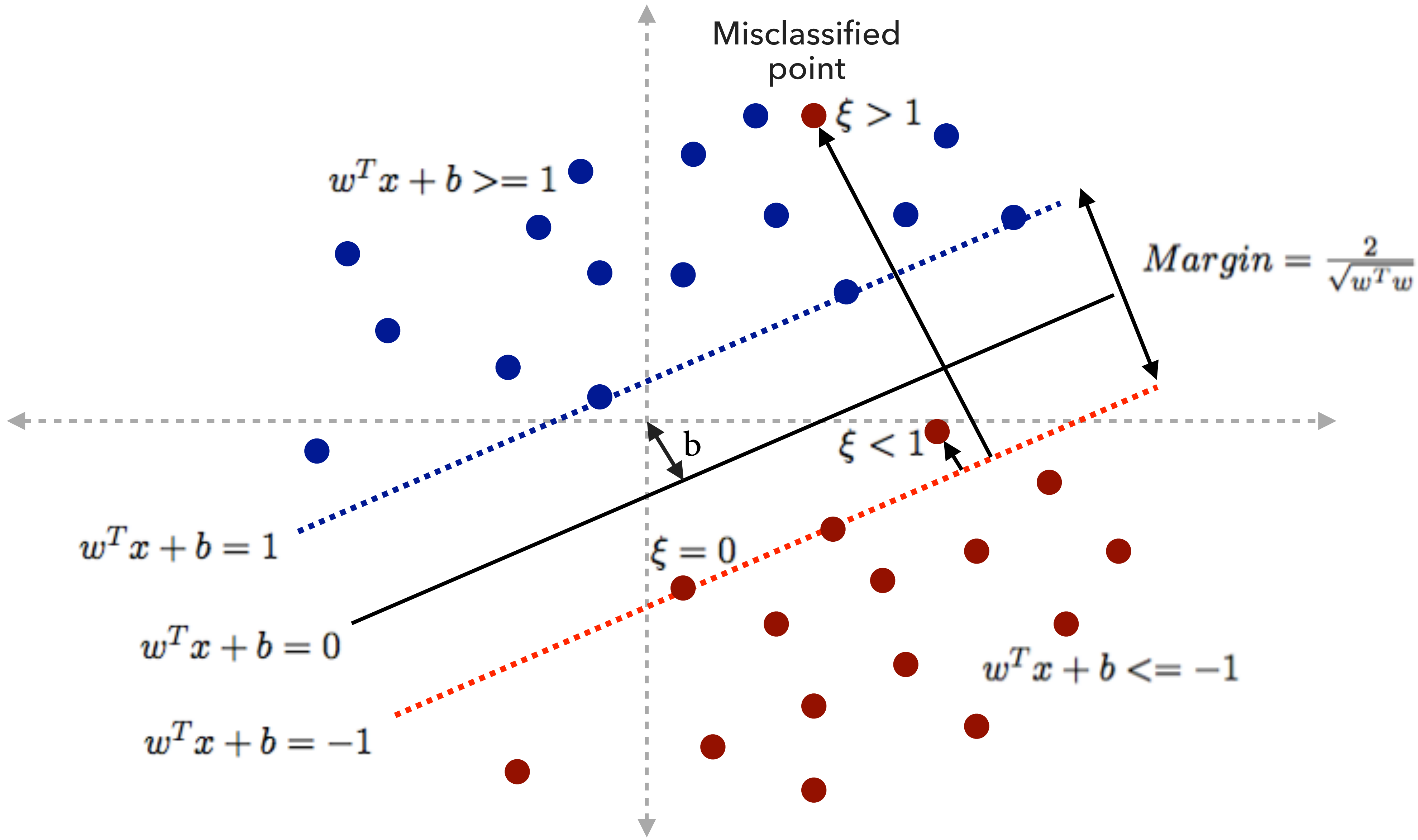


Figure 4: Optimal Hyperplane for Non-separable Patterns

HYPER-PARAMETERS SVM

C

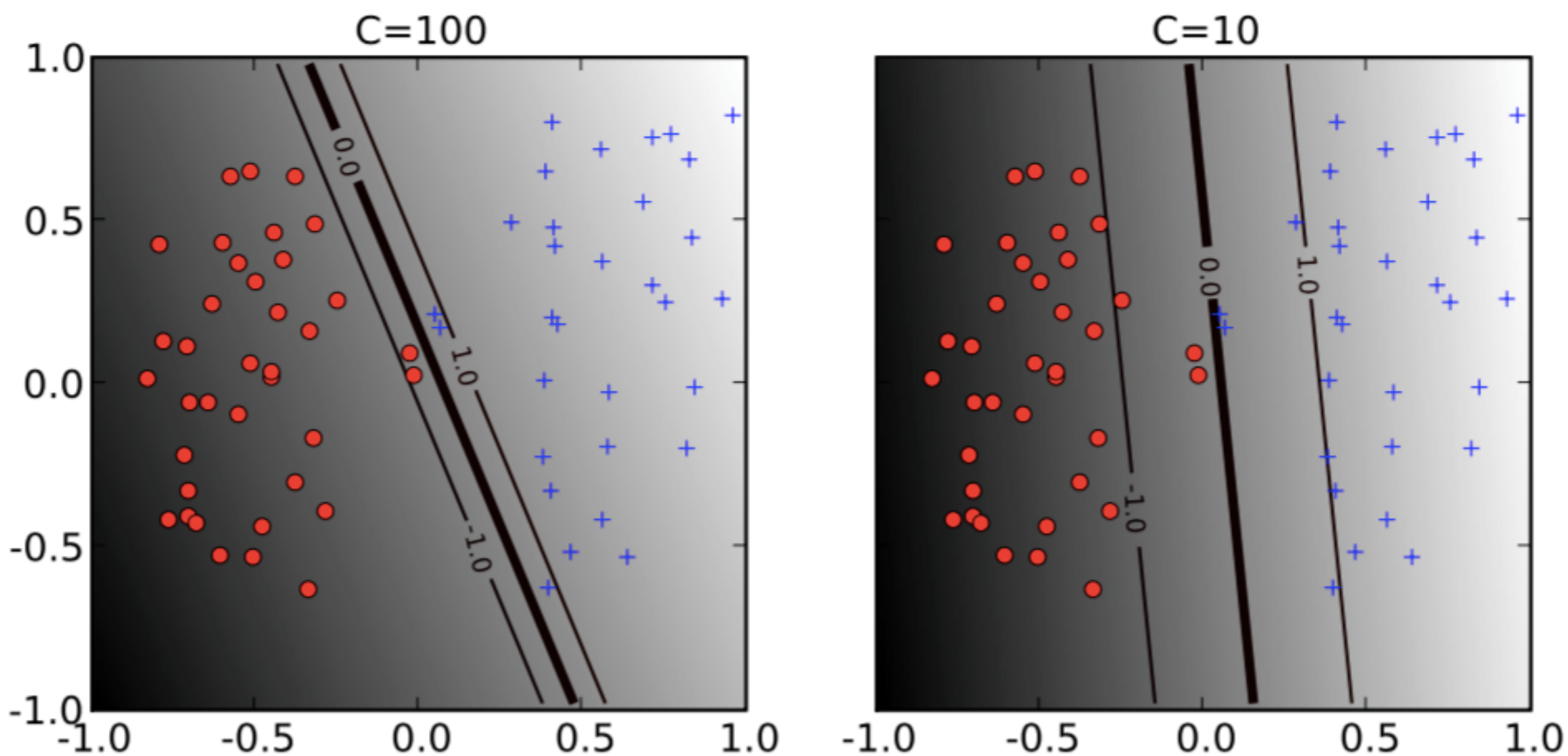


Figure 5: The effect of the soft-margin constant, C, on the decision boundary.
Low value of C (right) ignores more points; High values (left) aims at classifying all observations correctly.

HYPER-PARAMETERS SVM

gamma (γ)

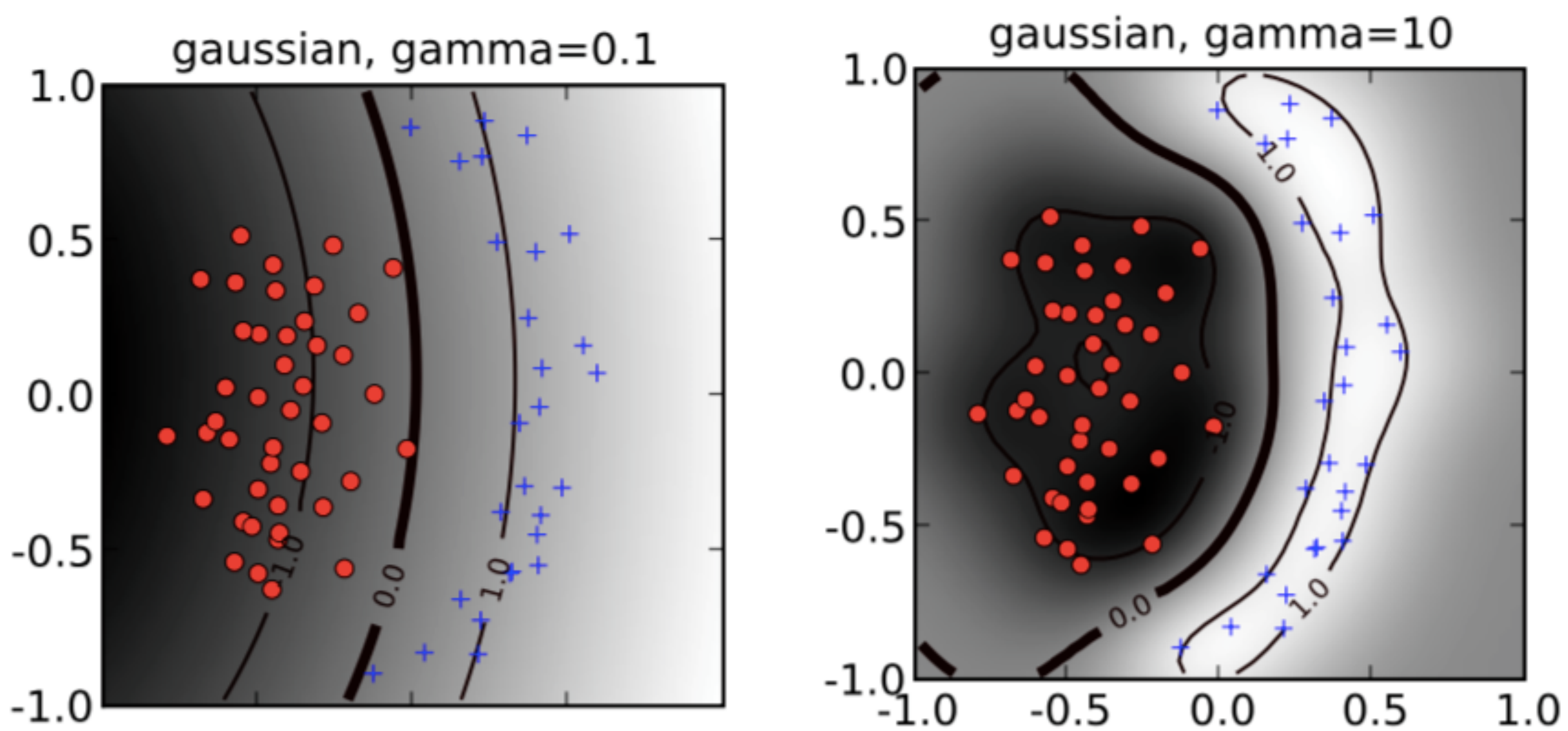


Figure 6: The effect of the inverse-width parameter of the Gaussian kernel (γ).
Small values (left) of γ lead in a decision boundary almost linear.
Large values of γ (right) lead to overfitting.

HYPER-PARAMETERS SVM

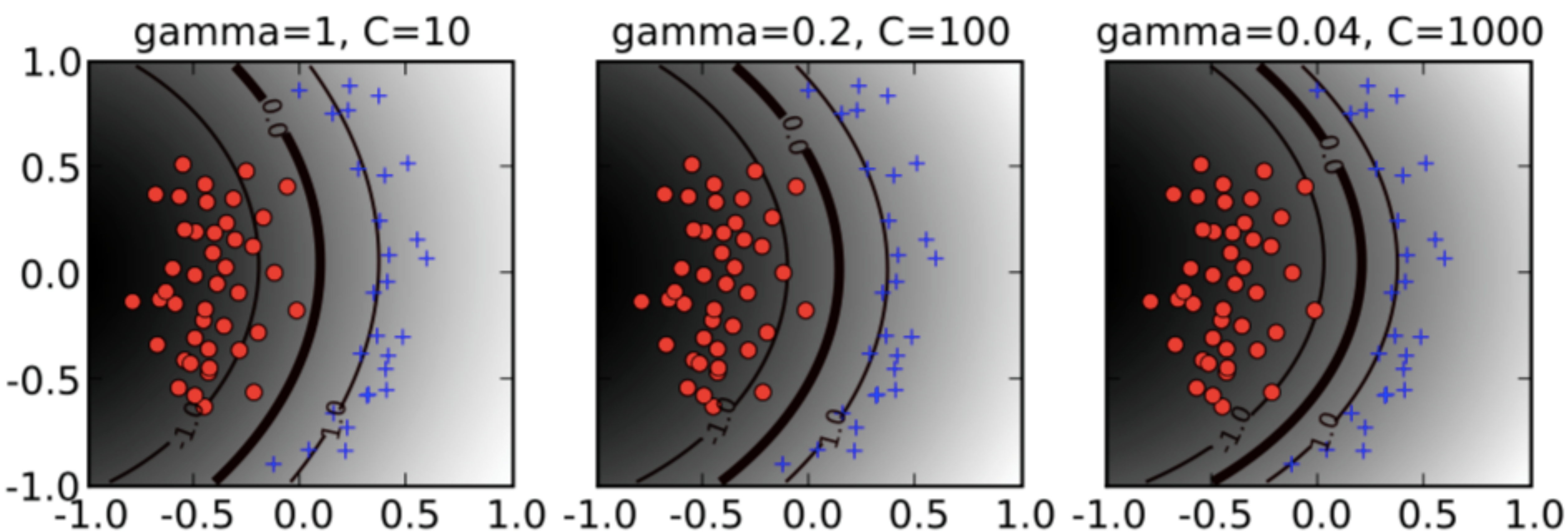


Figure7: Decision boundaries obtained by different combinations of SVM hyper-parameters

HYPER-PARAMETERS LSTM

- Epoch: one forward and backward pass for an entire training set
- Learning rate: it controls how fast or slow the synaptic weights of the RNN are updated in each epoch
- Batch size: the number of training samples used in one forward/ backward pass
- Sequence length: separation of the input samples into partitions
- Dropout rate: randomly select neurons to be ignored during training
- Hidden layer: number of neurons/cells in a hidden layer

GRID SEARCH VS RANDOM SEARCH

- Grid search and manual search are the most widely used techniques for hyper-parameter optimization but it has been empirically and theoretically demonstrated that randomly chosen tests are more efficient for hyper-parameter optimization than tests on a predefined grid: *Random Search for Hyper-Parameter Optimization, Yoshua Bengio* (http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf)

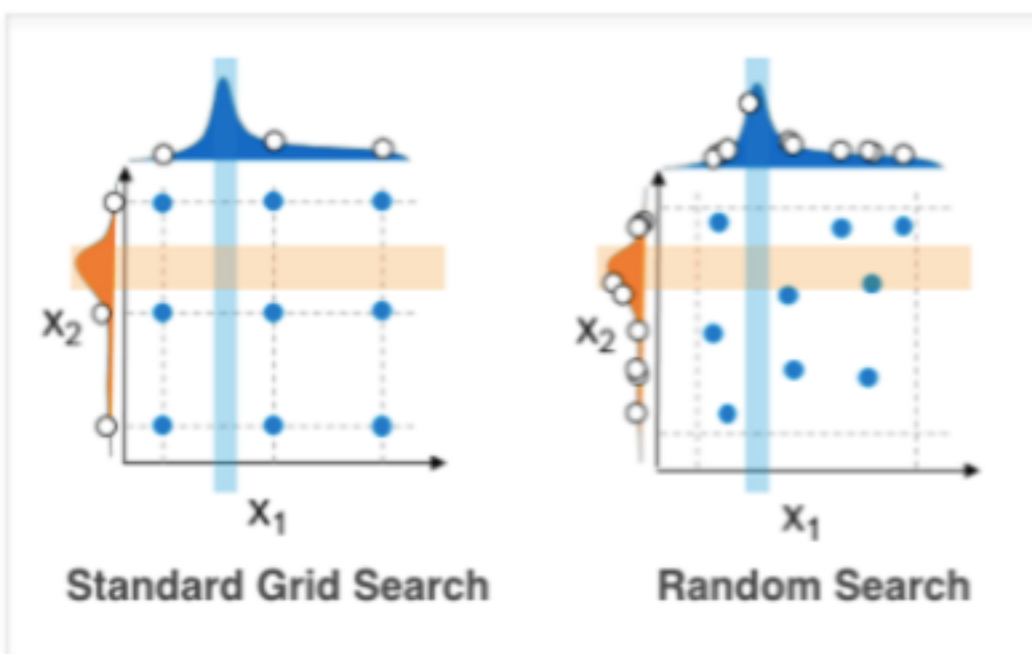


TABLE 2.3: Specific results of attack categories

Attack subtype		Short description	Attack type
Setpoint Attacks	1-2	Set values (setpoint, gain, reset rate, rate, deadband or cycle time) outside and inside of the range of normal operation.	MPCI
PID GainAttacks	3-4		MPCI
PID Reset RateAttacks	5-6		MPCI
PID RateAttacks	7-8		MPCI
PID DeadbandAttacks	9-10		MPCI
PID Cycle Time Attacks	11-12		MPCI
Pump Attack	13	Randomly changes the state of the pump, solenoid or system mode	MSCI
Solenoid Attack	14		MSCI
System Mode Attack	15		MSCI
Critical Condition Attacks	16-17	Places the system in a Critical Condition	MSCI
Bad CRC Attack	18	Sends Modbus packets with incorrect CRC values	DoS
Clean Registers Attack	19	Cleans registers in the slave device	MFCI
Device Scan Attack	20	Scan for all possible devices controlled by the master	Recon
Force Listen Attack	21	Forces the slave to only listen	MFCI
Restart Attack	22	Restart communication on the device	MFCI
Read Id Attack	23	Read ID of slave device	Recon
Function Code Scan Attack	24	Scans for possible functions that are being used on the system	Recon
Rise/Fall Attacks	25-26	Create trends on the pressure readings graph by sending back pressure readings	CMRI
Slope Attacks	27-28	Randomly increases/decreases pressure reading by a random slope	CMRI
Random Value Attacks	29-31	Sends random pressure measurements to the master	NMRI
Negative Pressure Attack	32	Sends back a negative pressure reading from the slave	NMRI
Fast Attacks	33-34	Sends back a high set point then a low setpoint which changes 'fast'/'slow'	CMRI
Slow Attack	35		CMRI