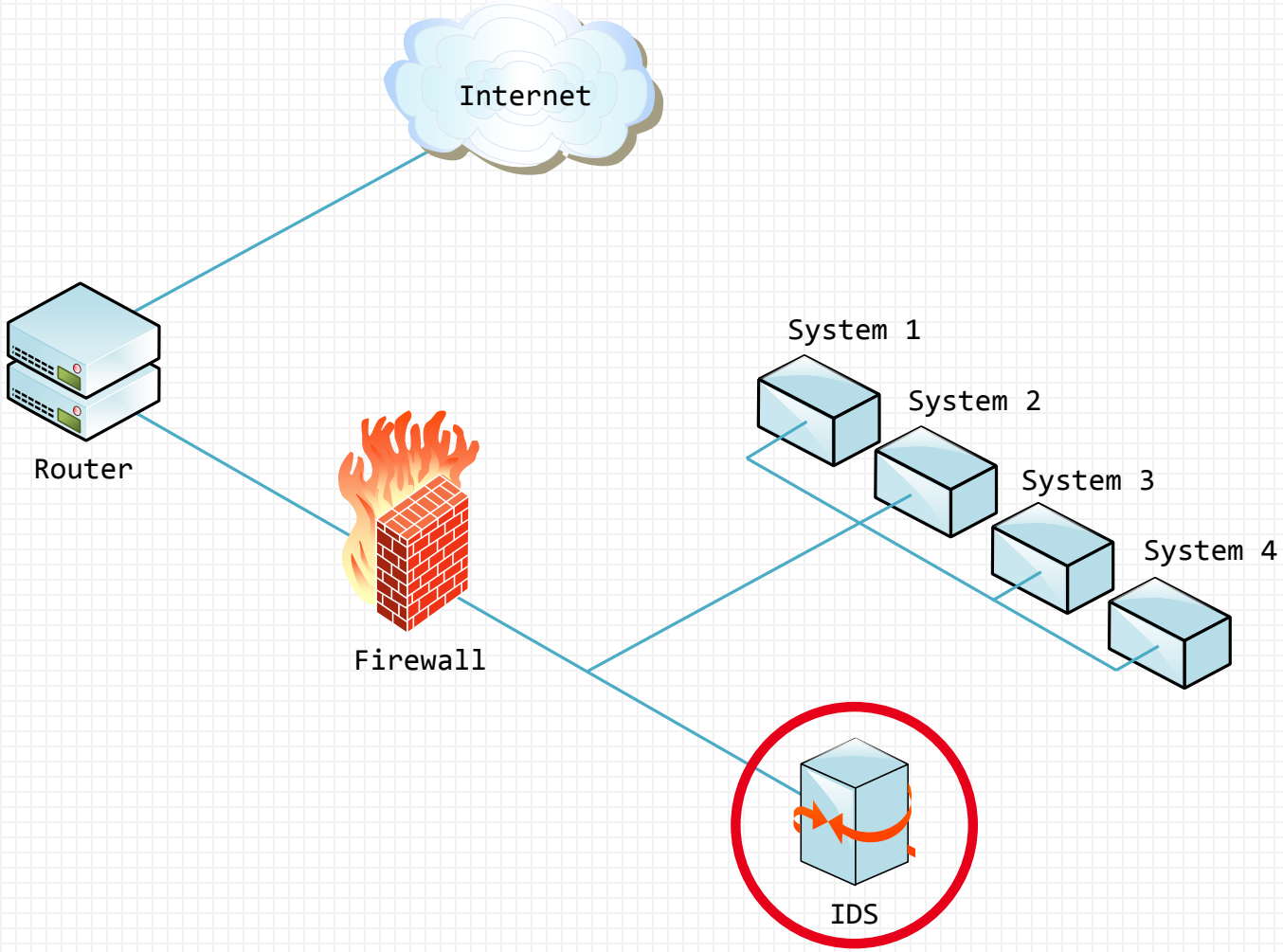
AUTOMATISATION DU PROCESSUS D'ENTRAÎNEMENT POUR LA DÉTECTION D'INTRUSION

OUTLINE

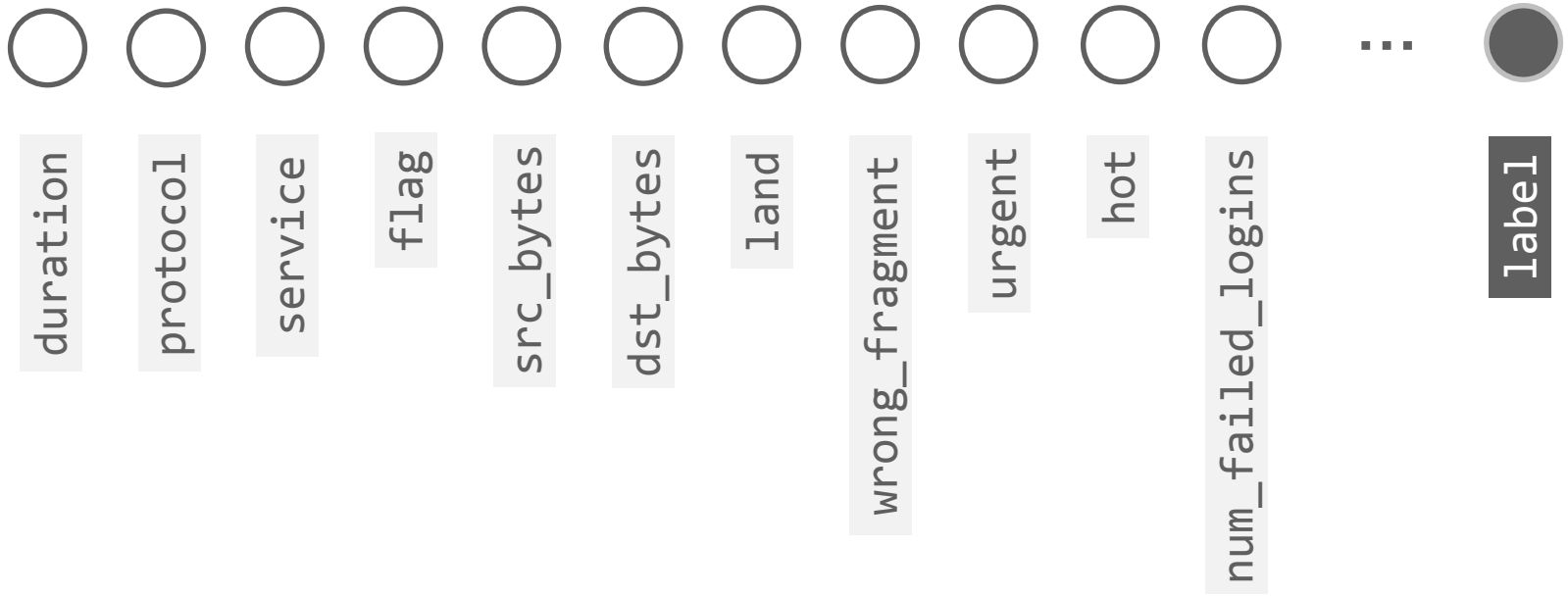
- **Intrusion Detection Systems**
- **Problem refinement**
  - NSL-KDD dataset
  - Classification process
- **Preprocessing**
- **Classifiers**
- **Ensemble learning**
- **Results**

INTRUSION DETECTION SYSTEMS



NSL-KDD DATASET

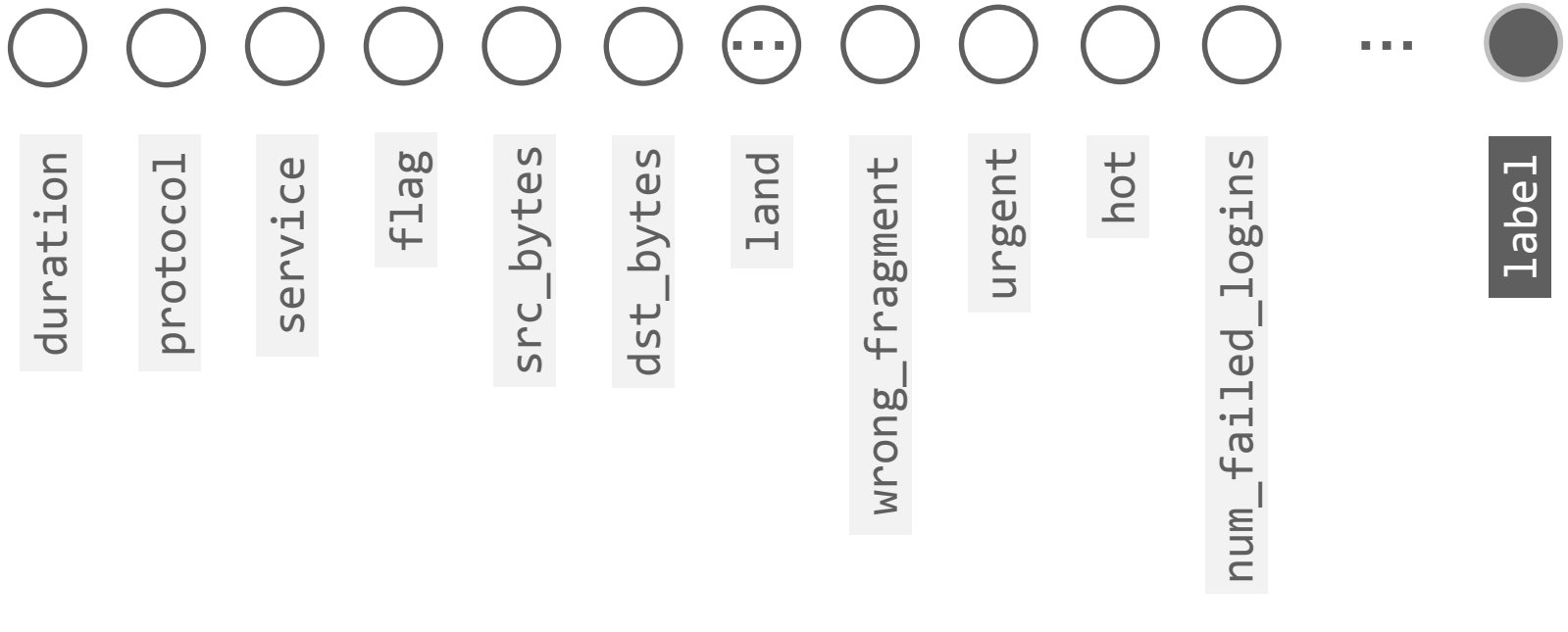
NSL-KDD DATASET



41 features + 1 label

Example 1:	2	tcp	ftp	SF	491	0	0	0	0	0	0	...	normal
Example 2:	0	tcp	priv	REJ	0	0	0	0	0	0	0	...	neptune

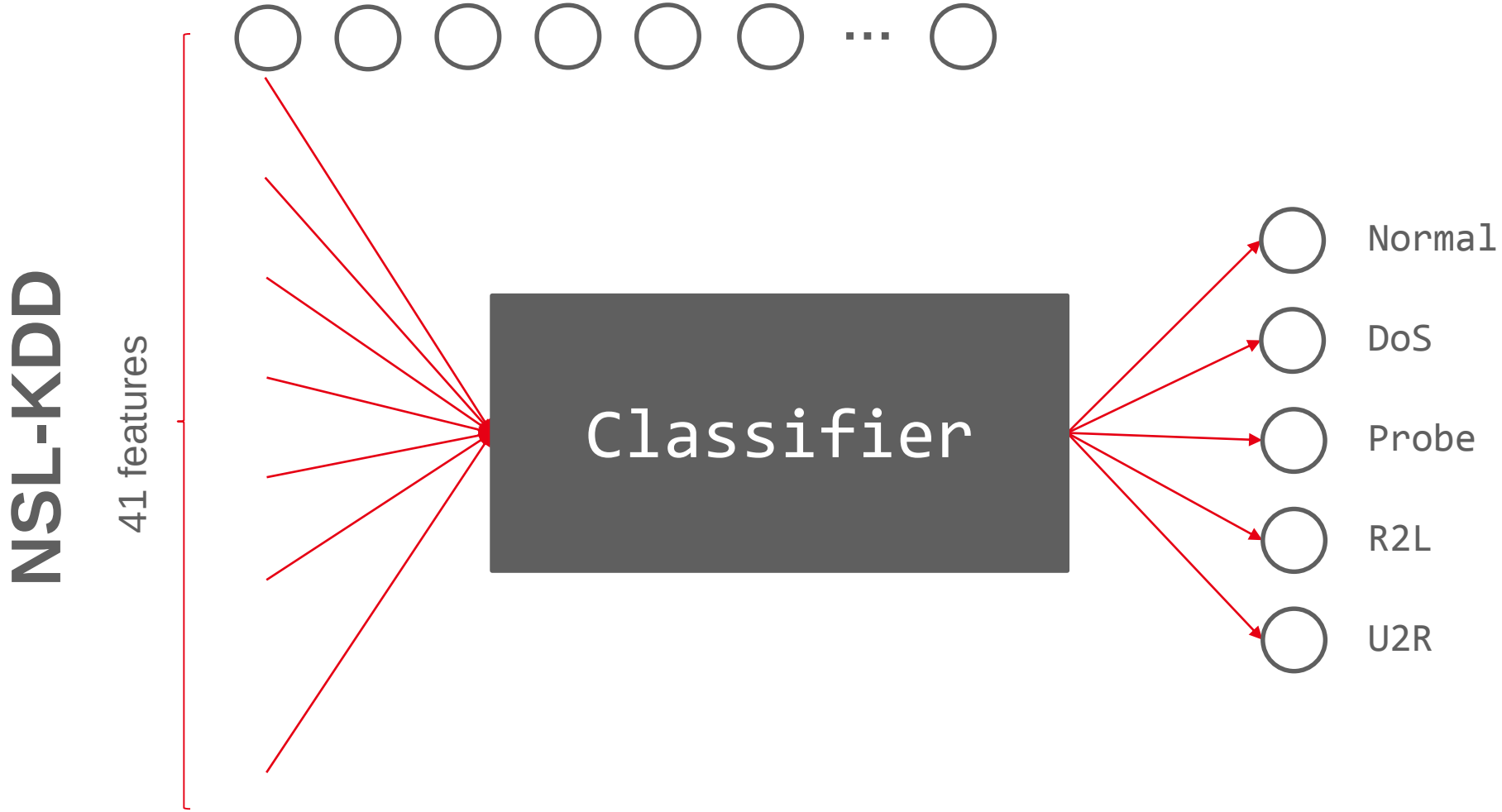
CLASSIFICATION PROCESS



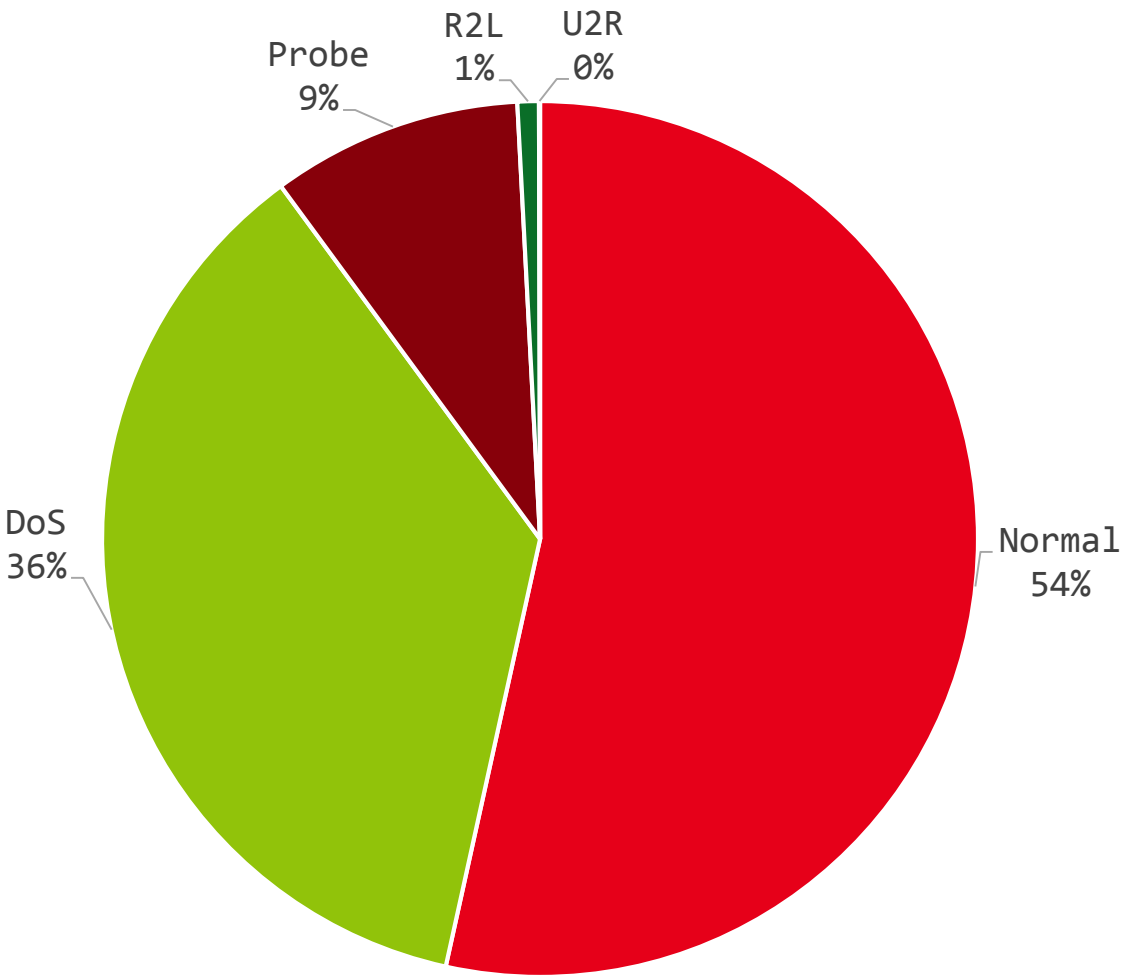
41 features + 1 label

Example 1:	2	tcp	ftp	SF	491	0	0	0	0	0	0	...	normal
Example 2:	0	tcp	priv	REJ	0	0	0	0	0	0	0	...	neptune

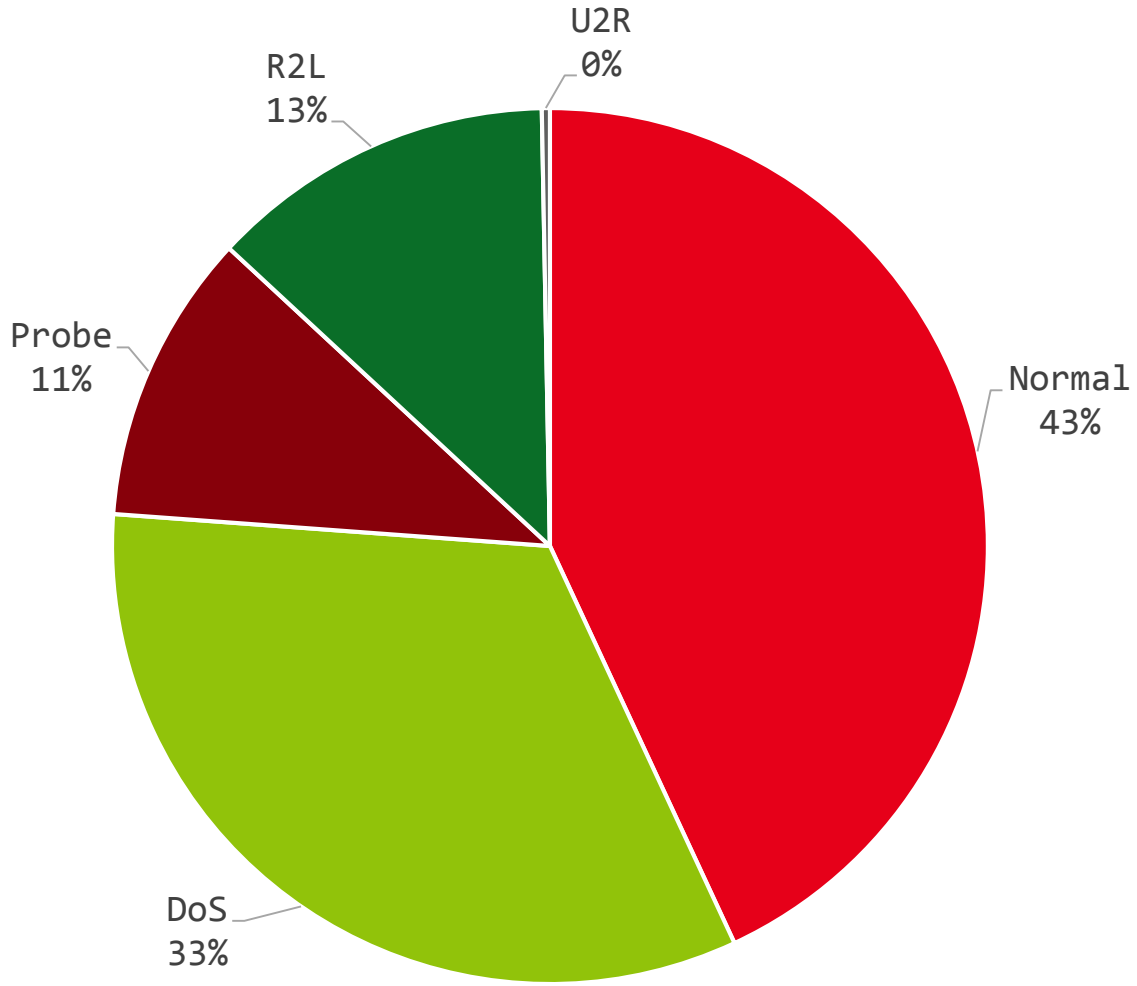
CLASSIFICATION PROCESS



PREPROCESSING

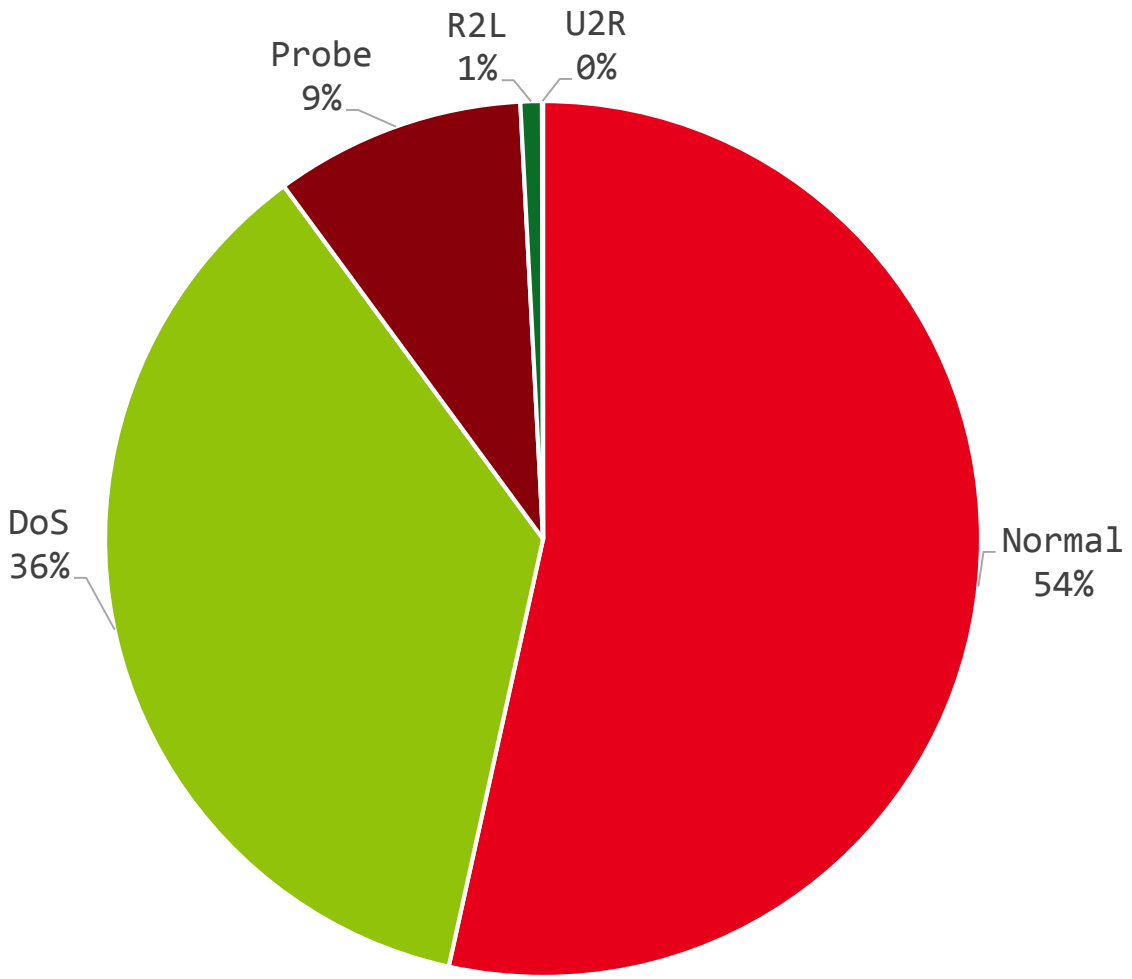


Training set



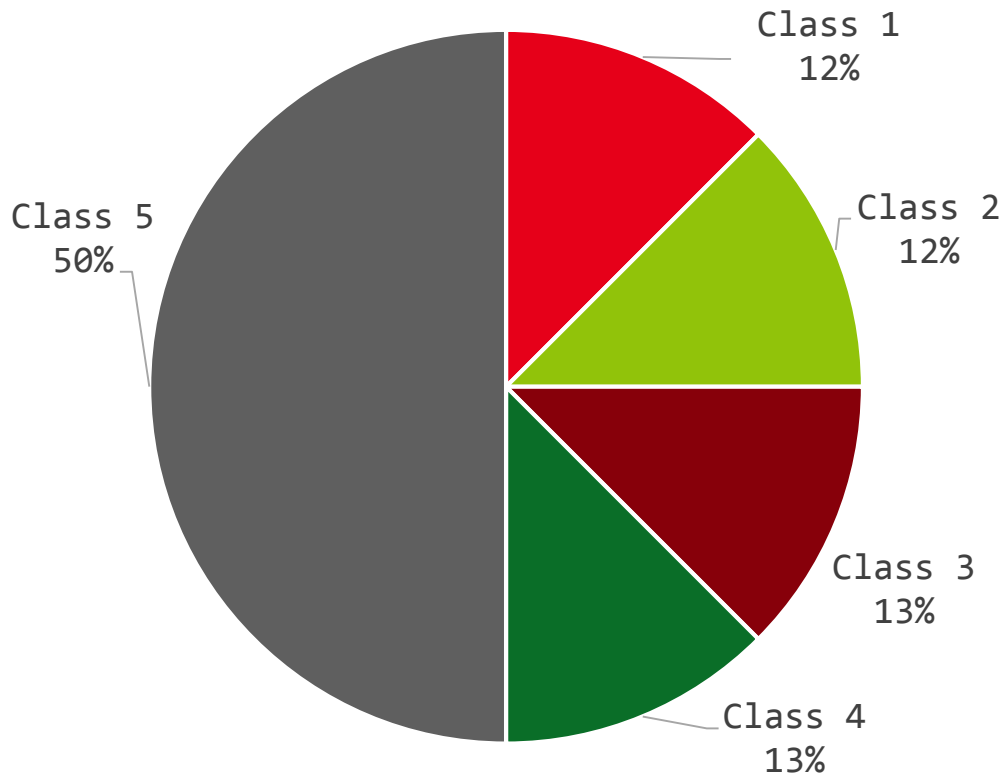
Test set

PREPROCESSING



Training set

Each class is successively **overrepresented** to create **5** specialized training sets



Specialized training + validation set

PREPROCESSING

ALGORITHMS

- *No sampling*

Under-sampling

- *Cluster Centroids*
- *Random Under Sampler*
- *NearMiss*
- *Edited Nearest Neighbours*
- *Repeated Edited Nearest Neighbours*
- *Condensed Nearest Neigbour*
- *One Sided Selection*
- *AllKNN*
- *Instance Hardness Threshold*

Over-sampling

- *Random Over Sampler*
- *SMOTE*
- *Borderline 1 SMOTE*
- *Borderline 2 SMOTE*
- *SVM SMOTE*
- *ADASYN*
- *SMOTEENN*
- *SMOTE Tomek*

We want to **combine** the best **under-sampling** algorithm with the best **over-sampling** algorithm

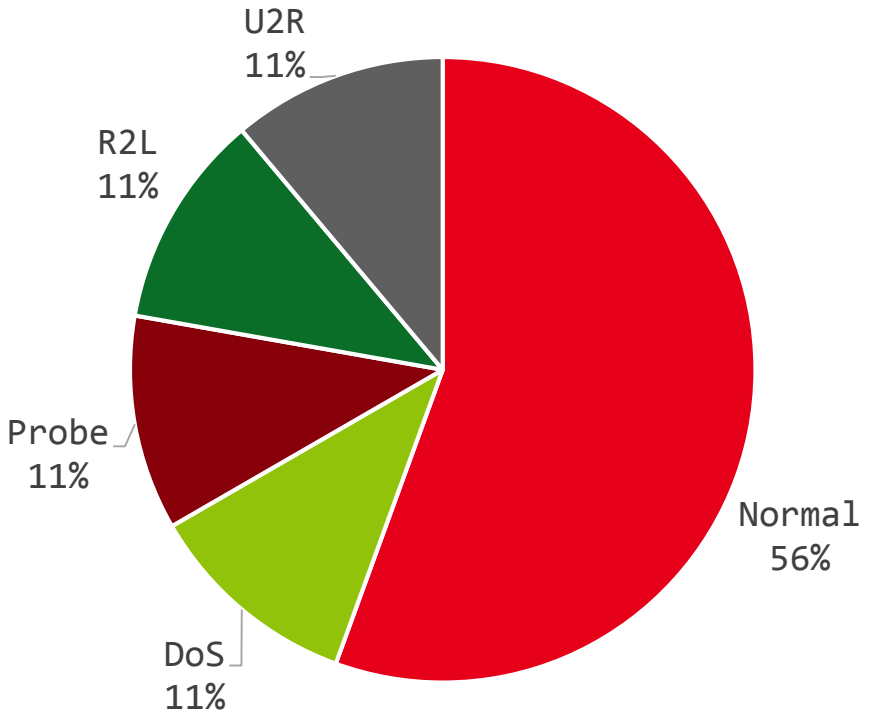
Best algorithm = best ROC AUC score

(classification with a simple MLP)

PREPROCESSING

	ALGORITHMS	ROC AUC SCORE Normal
	No sampling	0.9488
Under-sampling	*Cluster Centroids*	0.9378
	Random Under Sampler	0.9616
	NearMiss	0.9377
	Edited Nearest Neighbours	0.9382
	Repeated Edited Nearest Neighbours	0.9614
	Condensed Nearest Neigbour	0.9022
	One Sided Selection	0.9415
	AllKNN	**0.9659**
	Instance Hardness Threshold	0.6944
Over-sampling	*Random Over Sampler*	0.9611
	SMOTE	0.9423
	Borderline 1 SMOTE	0.9597
	Borderline 2 SMOTE	0.9542
	SVM SMOTE	**0.9644**
	ADASYN	0.9485
	SMOTEENN	0.9500
	SMOTE Tomek	0.9627

Normal training + validation set



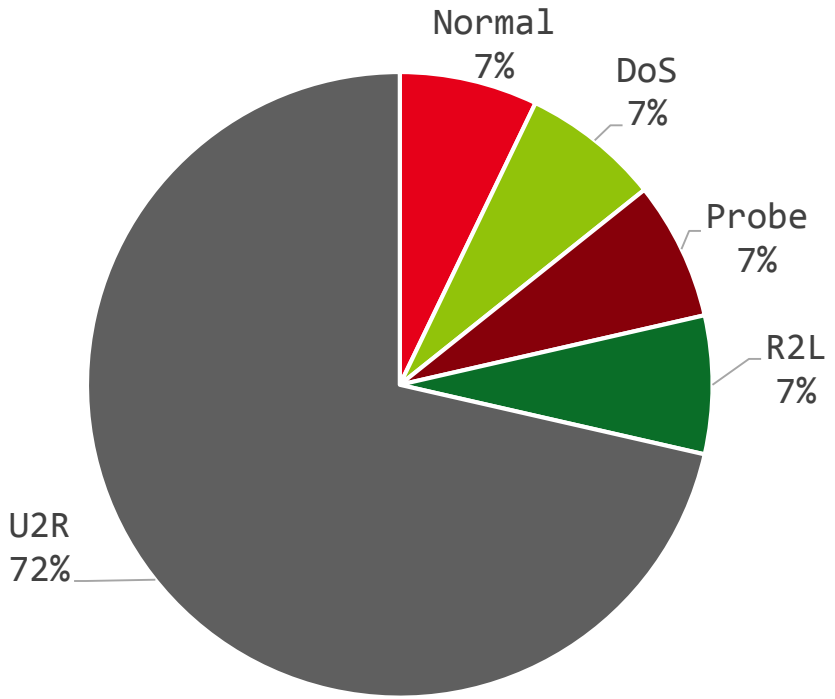
AllKNN + **SVM SMOTE** are used to:

- undersample *Normal*, *DoS* and *Probe*
- oversample *R2L* and *U2R*

PREPROCESSING

	ALGORITHMS	ROC AUC SCORE U2R
	No sampling	0.9605
Under-sampling	**Cluster Centroids**	**0.9739**
Under-sampling	*Random Under Sampler*	0.9715
Under-sampling	*NearMiss*	0.2762
Under-sampling	*Edited Nearest Neighbours*	0.9575
Under-sampling	*Repeated Edited Nearest Neighbours*	0.9626
Under-sampling	*Condensed Nearest Neigbour*	0.9174
Under-sampling	*One Sided Selection*	0.9435
Under-sampling	*AllKNN*	0.9701
Under-sampling	*Instance Hardness Threshold*	0.9458
Over-sampling	*Random Over Sampler*	0.9714
Over-sampling	*SMOTE*	0.9511
Over-sampling	*Borderline 1 SMOTE*	0.9675
Over-sampling	*Borderline 2 SMOTE*	0.9601
Over-sampling	*SVM SMOTE*	0.9696
Over-sampling	*ADASYN*	0.9717
Over-sampling	*SMOTEENN*	0.9560
Over-sampling	**SMOTE Tomek**	**0.9799**

U2R training + validation set



Cluster Centroids + **SMOTE Tomek** are used to:

- undersample *Normal*, *DoS, Probe* and *R2L*
- oversample *U2R*

CLASSIFIERS

For each 5 preprocessed datasets:

- **Decision Tree**
- **Random Forest**
- **Extra Tree**
- **Extra Trees**
- **k-NN**
- **SVM**
- **Logistic regression**
- **Naive Bayes**
- **Gradient boosting**
- **Multilayer perceptron**

Parameters optimization

Tree-structured

Parzen Estimator

Trained classifiers with optimized parameters

ENSEMBLE LEARNING

For each 5 preprocessed datasets:

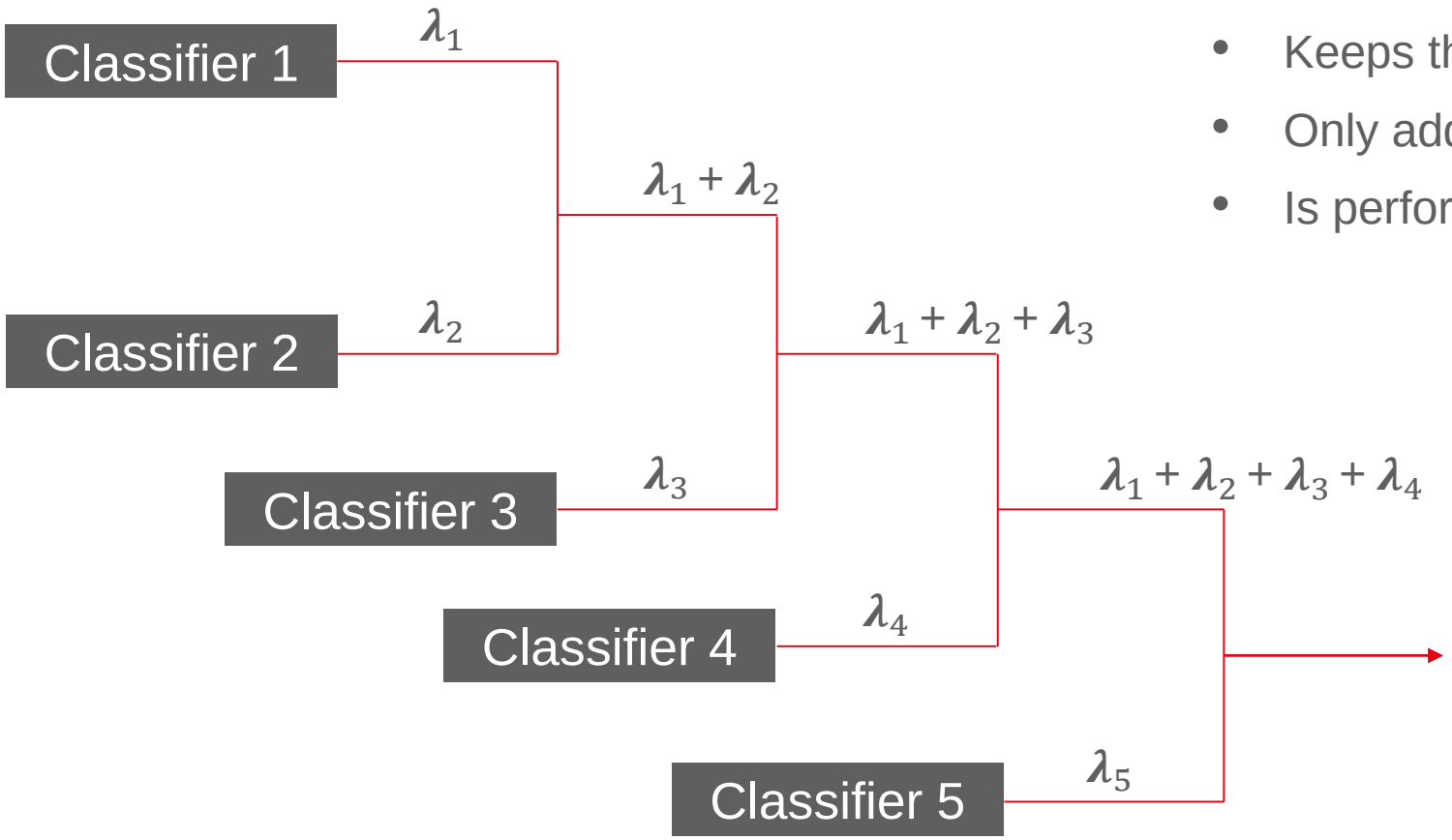
- Decision Tree p_1
- Random Forest p_2
- Extra Tree p_3
- Extra Trees p_4
- k-NN p_5
- SVM p_6
- Logistic regression p_7
- Naive Bayes p_8
- Gradient boosting p_9
- Multilayer perceptron p_{10}

Ensemble

$$p_{class} = \sum_{i=1}^{N} \lambda_i \, p_i$$

How to choose the best λ_i?

ENSEMBLE LEARNING



Each step:

- Tests 2000 values for λ_i (with $\lambda_j = 1 - \lambda_i$)
- Keeps the best λ_i and λ_j in terms of accuracy
- Only adds classifiers that improve accuracy
- Is performed on a validation set

$$p_{class} = \sum_{i=1}^{5} \lambda_i \, p_i$$

ENSEMBLE LEARNING

CLASSIFIER	ROC AUC score (test set)	λ_i
MLP3	0.8953	0.29
Decision Tree	0.7446	0.14
Extra Trees	0.8345	0.14
k-NN	0.8178	0.14
MLP1	**0.9022**	0.14
MLP2	0.8953	0.14

Example:

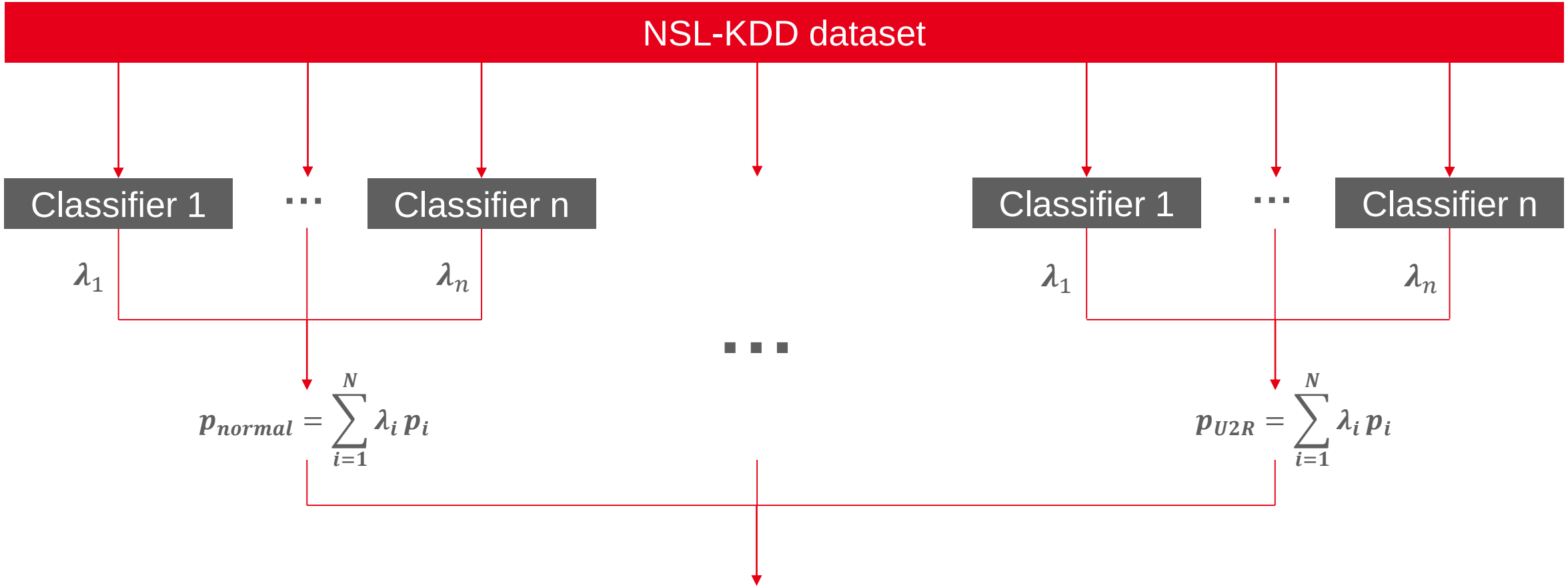
Probe meta-classifier

$$
\begin{aligned}
p_{probe} = \; & 0.56 \times p_{MLP1} + \\
& 0.05 \times p_{MLP3} + \\
& 0.01 \times p_{MLP2} + \\
& 0.11 \times p_{k-NN} + \\
& 0.14 \times p_{Decision\,Tree} + \\
& 0.12 \times p_{Extra\,Trees}
\end{aligned}
$$

Probe ROC AUC score = 0.9184

(0.9022 at best without ensemble)

ENSEMBLE LEARNING



RESULTS

NSL-KDD (global accuracy = 86.59%)

	Normal	*DoS*	*Probe*	*R2L*	*U2R*
Accuracy	95.20%	90.91%	95.87%	91.73%	99.48%
TPR	88.85%	95.99%	86.70%	50.02%	43.28%
FPR	**1.66%**	12.94%	3.03%	2.15%	0.36%
F1 score	0.9245	0.9009	0.8185	0.6076	0.3295
AUC ROC	0.9359	0.9153	0.9184	0.7394	0.7146

Comparison of results on NSL-KDD

Study	*Accuracy*	*FPR*
Our solution	**86.59%**	**1.66%**
Two-level classifier ensemble	85.016%	12.6%
Bagging (J48) + feature selection	84.25%	2.79%
GAR-forest + feature selection	85.05%	12.2%
SVM + feature selection	82.37%	15%

- **Very high global accuracy and good FPR**
- **R2L and U2R attacks are still poorly detected**
- **All optimization steps are fully automated**

- **Can be applied to:**
  - Other datasets (CICIDS 2017)
  - Real-life networks (for classification)

THANK YOU!

Commissariat à l'énergie atomique et aux énergies alternatives
Institut List | CEA SACLAY NANO-INNOV | BAT. 861 – PC142
91191 Gif-sur-Yvette Cedex - FRANCE
www-list.cea.fr

Établissement public à caractère industriel et commercial | RCS Paris B 775 685 019