Adversarial training for intrusion detection with GANs

Group 2

Mohamed Mahgoub
Ahmed Abdelmaksoud
Michael Khalil
Dina Abdelhaيdy

Introduction:

- Machine learning is a double-edged weapon that can both protect against vulnerabilities or create them.
- Generate new samples to enhance an existing model, then compare results before and after enhancement.
- Investigate effectiveness of GANs in improving IDS models.

Objective:

1. Train the models on pure training data.
2. Generate & validate new samples.
3. Train on both data.
4. Compare results.

NSL-KDD Dataset:

- 41 features " 9 discrete and 32 continuous "
- 148517 records
- 40 classes

F#	Feature name	F#	Feature name	F#	Feature name
F1	Duration	F15	Su attempted	F29	Same srv rate
F2	Protocol type	F16	Num root	F30	Diff srv rate
F3	Service	F17	Num file creations	F31	Srv diff host rate
F4	Flag	F18	Num shells	F32	Dst host count
F5	Source bytes	F19	Num access files	F33	Dst host srv count
F6	Destination bytes	F20	Num outbound cmds	F34	Dst host same srv rate
F7	Land	F21	Is host login	F35	Dst host diff srv rate
F8	Wrong fragment	F22	Is guest login	F36	Dst host same src port rate
F9	Urgent	F23	Count	F37	Dst host srv diff host rate
F10	Hot	F24	Srv count	F38	Dst host serror rate
F11	Number failed logins	F25	Serror rate	F39	Dst host srv serror rate
F12	Logged in	F26	Srv serror rate	F40	Dst host rerror rate
F13	Num compromised	F27	Rerror rate	F41	Dst host srv rerror rate
F14	Root shell	F28	Srv rerror rate	F42	Class label

Preprocessing:

1. Map the class label into either an attack (1) or normal (0).

2. Apply feature selection by manual inspection to include only 15 features out of 41.

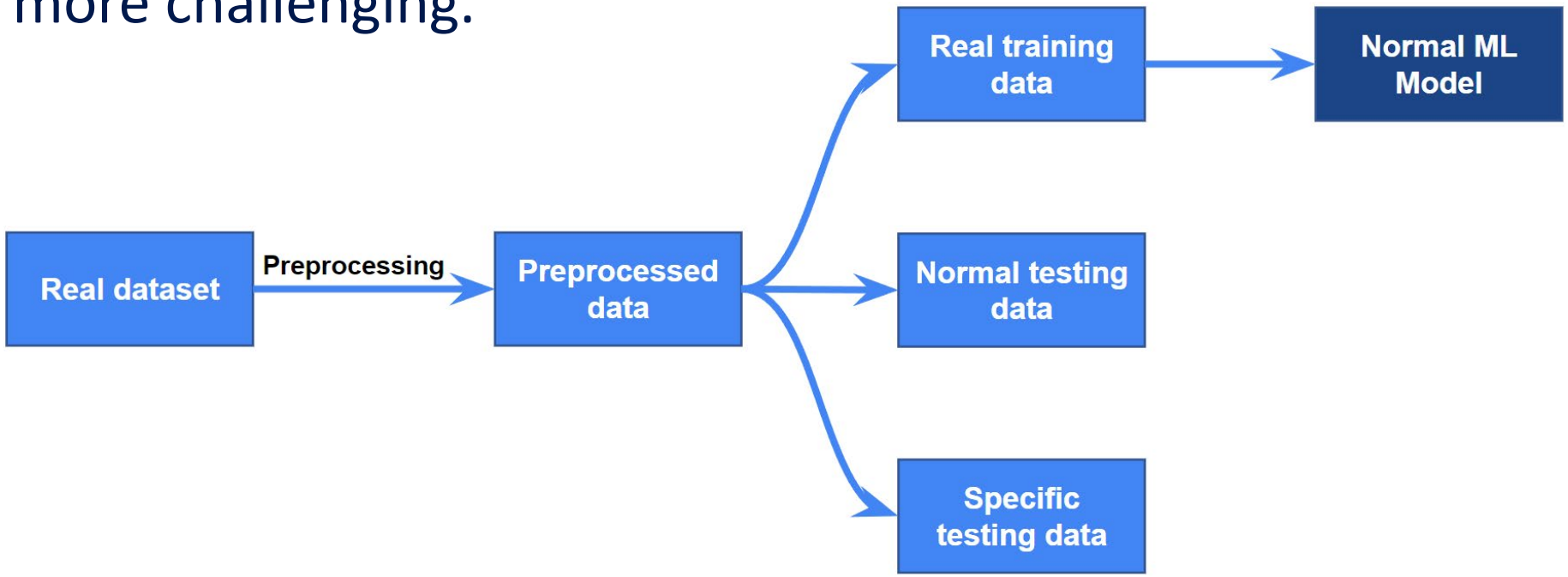
3. Apply outlier removal with z-score.

4. Apply log transformation – normalization – binning – label encoding – etc...

Methodology:

1. Data splitting:

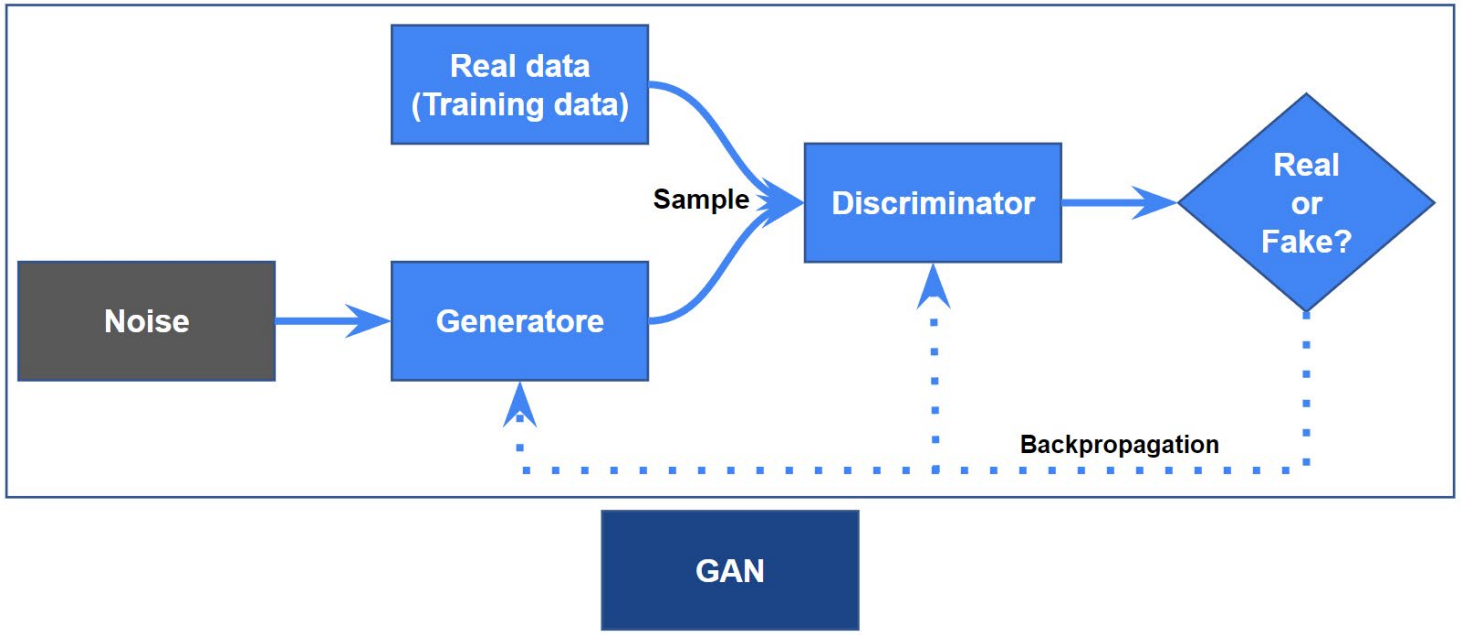
- 85% train – 15% normal test – 15% specific test.
- The normal test data is somewhat close to the train data.
- The specific test data is carefully selected and therefore is more challenging.



Methodology:

2. The GAN model includes:

- A generator network generates new data instances.
- a discriminator network discriminates between original data and fake data produced by the generator.

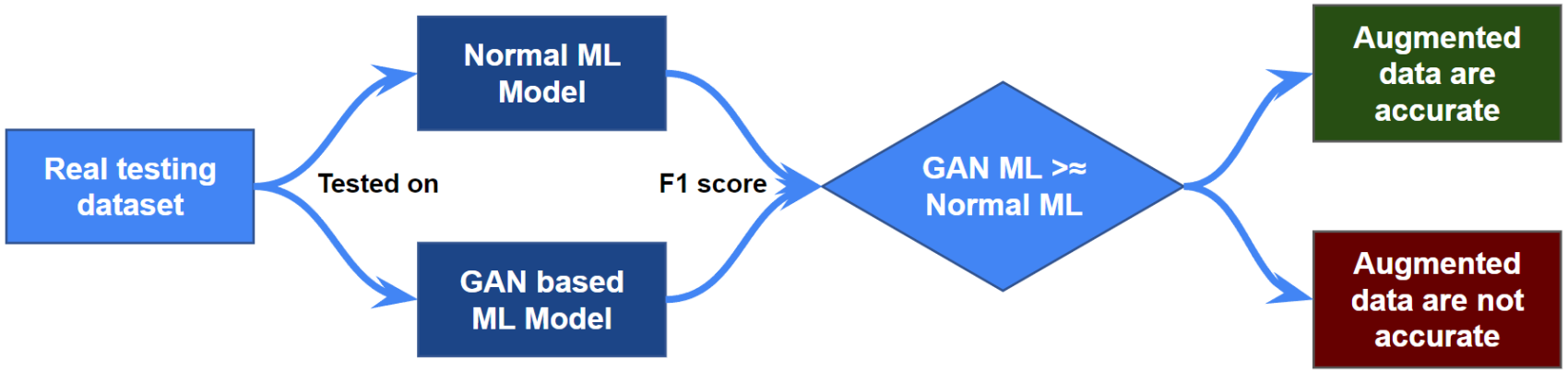


Methodology:

3. We used the trained GAN model to produce a new huge augmented (fake) dataset.
Then we train a model based on these data.

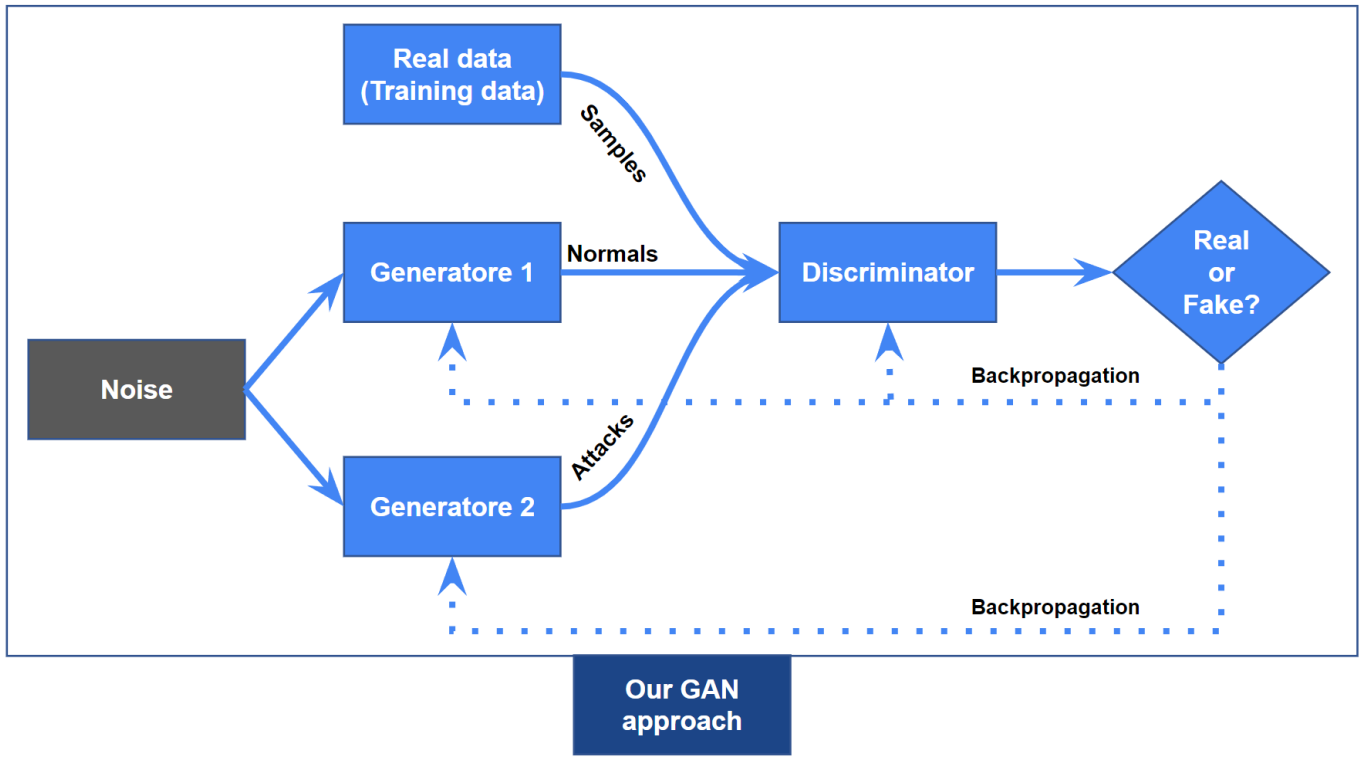
Methodology:

4. We will test the two models on the normal and specific datasets.
The more accuracy achieved by the augmented model the more the probability that it creates new attacks.



GANs details

Two Generators Architecture



GANs details

Noise trick

	protocol_type	service	flag
0	1	29	1
1	2	48	9
2	2	12	9
3	1	6	5

	protocol_type	service	flag
0	0.983290	28.965550	1.079388
1	2.025259	48.021805	9.051507
2	1.849349	11.991931	9.117724
3	0.999470	5.988304	5.052765

GANs details

Rounding trick

	protocol_type	service	flag
0	0.806366	48.653862	9.434033
1	0.360062	36.938915	4.837991
2	0.550718	30.243591	5.581261
3	0.591307	33.678612	6.065347

	protocol_type	service	flag
0	1.0	49.0	9.0
1	0.0	37.0	5.0
2	1.0	30.0	6.0
3	1.0	34.0	6.0

GANs details

Settings and Generation

Settings

- Epoch: 1000
- Batch: 1024
- Noise: 8
- Layers: Dense

Generation

- Normal: 50,000
- Attack: 50,000
- Total: 100,000

Experiment

1. Building 6 different models trained on the original training data.

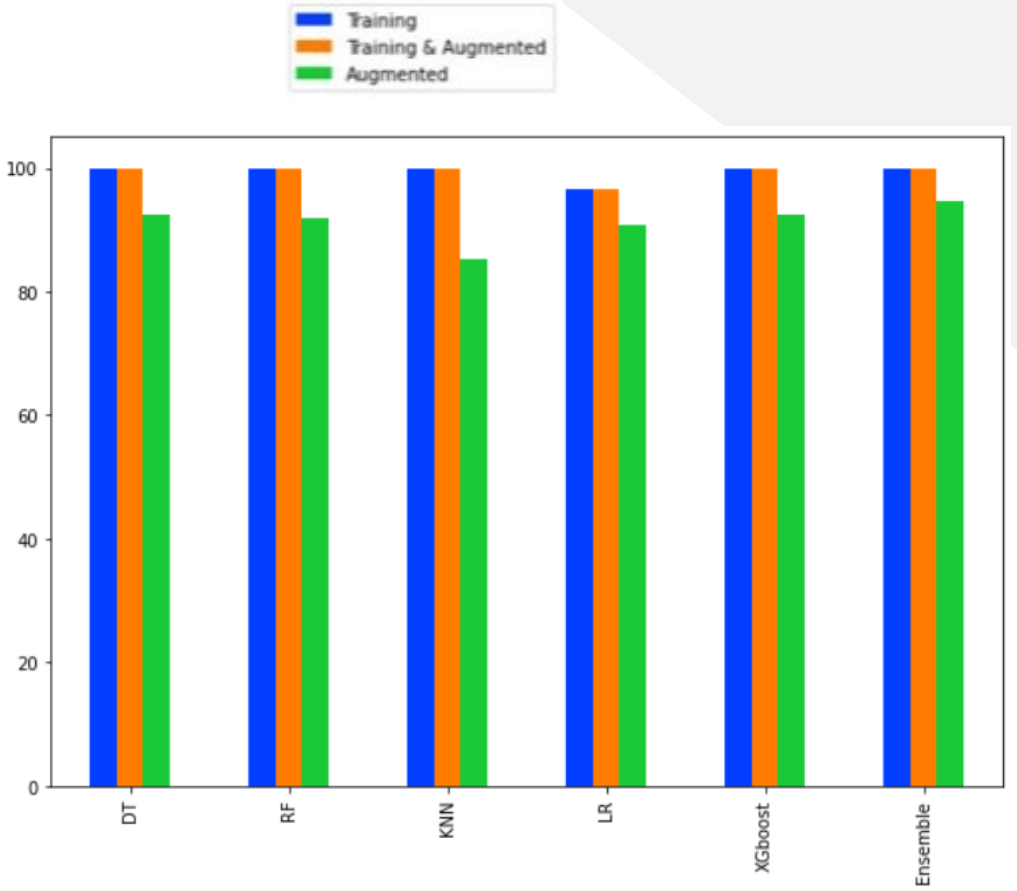
2. Building 6 models trained on the augmented data which is generated by GAN.

3. Building 6 models trained on the original training data and the augmented data.

4. Testing the models on the original testing data and comparing their results.

Normal Test Results:

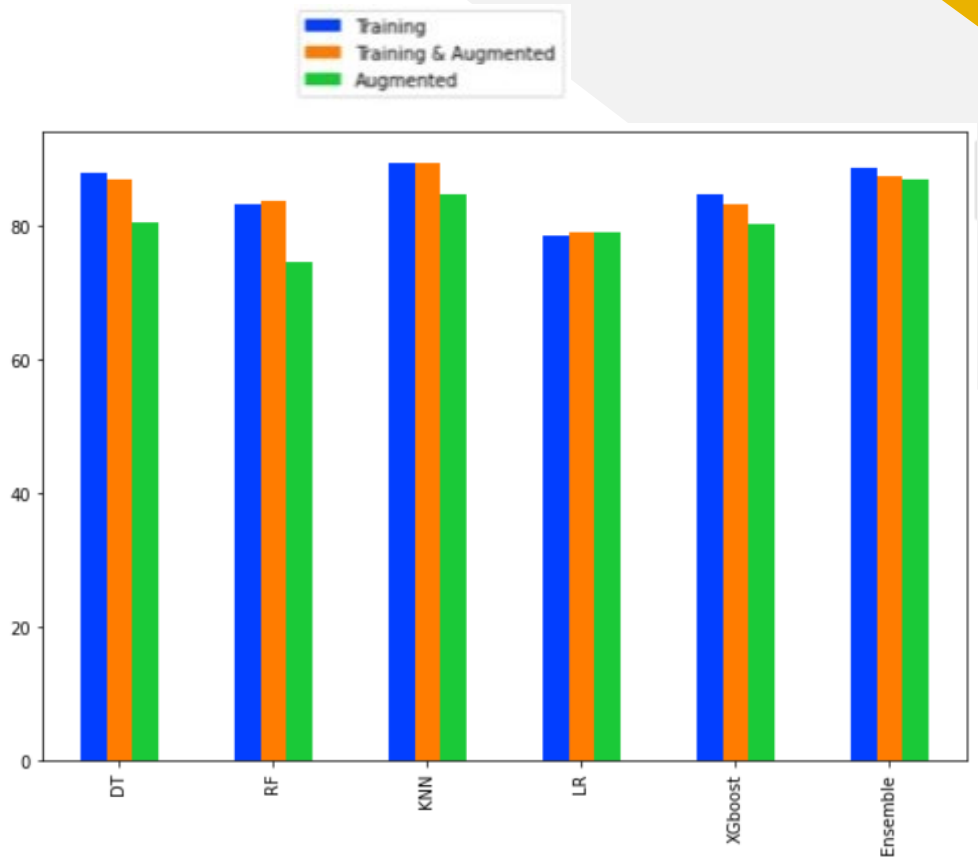
	Original Data	Generated Data	Original + Generated Data
RF	**0.999**	**0.918**	**0.999**
KNN	**0.999**	**0.853**	**0.999**
LR	**0.965**	**0.908**	**0.966**
DT	**0.999**	**0.923**	**0.998**
XGB	**0.999**	**0.924**	**0.999**
Ensemble	**0.999**	**0.947**	**0.999**



F1_score

Specific Test Results:

	Original Data	Generated Data	Original + Generated Data
RF	**0.827**	**0.744**	**0.836**
KNN	**0.895**	**0.846**	**0.895**
LR	**0.78**	**0.79**	**0.791**
DT	**0.887**	**0.804**	**0.87**
XGB	**0.846**	**0.803**	**0.832**
Ensemble	**0.886**	**0.867**	**0.875**



F1_score

Conclusion:

- metrics indicate that Generating data is similar to real intrusions
- Some generated data could resemble real potential attacks that could be done in the future.
- GANs show promise for generating attacks to enhance intrusion detection models

Future work:

- Augmented data validation.
- Early stopping for GANs (WGANs Wasserstein loss).
- Inverse pre-processing for generated data.
- More sophisticated GAN architecture and/or other GAN types.

Thank you