ISSS610 PROJECT
DDOS ATTACK DETECTION USING MACHINE LEARNING

By Group 3:

Aishwarya KRISHNA PRASAD, HAI Dan, LAN Chengcheng, LU Di, Nicole TAY Zi Min

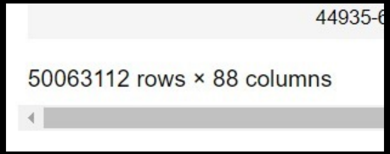
INTRODUCTION

- A Distributed Denial of Service (DDoS) attack is a menace to network security that aims at exhausting the target networks with malicious traffic. It is one of the most common and dangerous types of attacks as: 1) they can be hard to discover and, 2) have huge repercussions on organizations.
- This is an issue in the Software-Defined Networking (SDN) paradigm. With the rise of DDoS, it shows that legacy defense mechanisms are only partially effective and there is a need for better detection and prevention of attacks.
- Using a dataset CICDDoS2019 which contains a comprehensive variety of DDoS attacks and addresses the gap in many currently available datasets, we seek to propose models capable of detecting anomalies in network traffic and DDoS attacks.

DATASETS

Training sets	DNS	LDAP	MSSQL	NetBIOS	NTP	SNMP	SSDP	UDP	Syn	TFTP	UDPLag	-
Testing sets	-	LDAP	MSSQL	NetBIOS	-	-	-	UDP	Syn	-	UDPLag	Portmap

- CICDDoS2019: collected in two separate days for training and testing
  - Training data: captured on Jan 12, 2019, contains 12 different kinds of DDoS attacks, recorded in 12 separate files.
  - Testing data: captured on Mar 11, 2019, contains 7 different kinds of DDoS attacks ,recorded in 7 separate files.
  - PortScan attack only executed in test dataset. Purpose is to evaluate the ML models and ensure that they do not overfit the training data.
- 87 features + 1 target variable: ‘BENIGN’ or ‘Attack_type’(imbalance)
- Number of rows across training & testing: >50M



DATA PRE-PROCESSING

In each Train and Test datasets

- Drop 7 static features

Flow ID, Source IP, Source Port, Destination IP, Destination Port, Protocol and Timestamp

- Drop 13 constant features

Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, FIN Flag Count, PSH Flag Count, ECE Flag Count, Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk, SimillarHTTP and Bwd Avg Bulk Rate

- Drop 6 duplicate features

RST Flag Count, Fwd Header Length, Subflow Fwd Packets, Subflow Fwd Bytes, Subflow Bwd Packets and Subflow Bwd Bytes

- Remove 4603 NA rows
- Remove Inf number to process further Normalization Process

DATA PRE-PROCESSING:

DEALING WITH IMBALANCED DATASET

- Training datasets and testing datasets are imbalanced

Training Datasets	Before	After
Attack	99.9%	86.4%
Benign	0.1%	13.6%

- Combine training and testing datasets

Combined	Training Datasets	Testing Datasets
Attack	215192	133868
Benign	34036	56308

- Reserved only 20000 DDoS attack rows and keep all the benign in each training and testing datasets

Testing Datasets	Before	After
Attack	99.9%	70.4%
Benign	0.1%	29.6%

MODELS

- Supervised:
  - Logistic Regression as baseline
  - RamdomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier, XGBoostClassifier
  - Deep learning ANN
- Semi-supervised:
  - Encoders

LOGISTIC REGRESSION

- GridSearchCV() to find best parameters

Parameter	C	fit_intercept	penalty
Value	15	True	l2

- Confusion Matrix

Confusion Matrix (Test Dataset)		
	Predicted Benign	Predicted Attack
Actual Benign	33130	**23176**
Actual Attack	1320	132548

- Metrics

Evaluation Metrics	Validation Datasets	Test Datasets
Precision	0.9494	**0.8512**
Recall	0.9980	0.9901
F1 Score	0.9731	0.9154
AUC	0.9471	0.9120

- Performing better in Recall
- In cyber security, we need the model to detect as many DDoS attacks as possible, while not misclassifying benign instances.
- Precision 0.8512 is not enough.
- Need higher Precision.

SUPERVISED – RANDOMFOREST, ADABOOST, GRADIENTBOOSTING, XGBOOST – PARAMETERS

- Use RandomizedSearchCV() to find best parameters

RandomForestClassifier	n_estimators	min_samples_split	min_samples_leaf	max_features	max_depth	criterion	bootstrap
Best Parameters	200	8	1	sqrt	None	entropy	True

AdaBoostClassifier	n_estimators	learning_rate	algorithm
Best Parameters	50	1	SAMME.R

GradientBoostingClassifier	n_estimators	min_samples_split	min_samples_leaf	max_features	max_depth	learning_rate
Best Parameters	150	2	1	auto	5	0.1

XGBoostClassifier	min_child_weight	gamma	max_depth	learning_rate	colsample_bytree
Best Parameters	1	0.0	6	0.3	0.5

EVALUATION – RANDOMFOREST, ADABOOST

- RandomForestClassifier

Evaluation Metrics	Validation Datasets	Test Datasets
Precision	0.99995	0.9964
Recall	0.9998	0.9988
F1 Score	0.9999	0.9976
AUC	0.9997	0.9951

Confusion Matrix (Test Dataset)		
	Predict Benign	Predict Attack
Actual Benign	55824	482
Actual Attack	159	133709

- AdaBoostClassifier

Evaluation Metrics	Validation Datasets	Test Datasets
Precision	0.9995	0.9839
Recall	0.9996	0.9996
F1 Score	0.9996	0.9917
AUC	0.9983	0.9804

Confusion Matrix (Test Dataset)		
	Predict Benign	Predict Attack
Actual Benign	54122	2184
Actual Attack	58	133810

EVALUATION – GRADIENTBOOSTING, XGBOOST

- GradientBoostingClassifier

Evaluation Metrics	Validation Datasets	Test Datasets
Precision	0.9999	0.9914
Recall	0.9997	0.9999
F1 Score	0.9998	0.9957
AUC	0.9996	0.9897

Confusion Matrix (Test Dataset)		
	Predict Benign	Predict Attack
Actual Benign	55149	1157
Actual Attack	10	133858

- XGBoostClassifier

Evaluation Metrics	Validation Datasets	Test Datasets
Precision	0.99995	0.9967
Recall	0.9998	0.9999
F1 Score	0.9999	0.9983
AUC	0.9997	0.9960

Confusion Matrix (Test Dataset)		
	Predict Benign	Predict Attack
Actual Benign	55862	444
Actual Attack	15	133853

FEATURE IMPORTANCE – RANDOMFOREST, ADABOOST, GRADIENTBOOSTING, XGBOOST

Feature Importance (top 3)	RandomForest	AdaBoost	GradientBoosting	XGBoost
1	Inbound	Init_Win_bytes_backward	Inbound	Inbound
2	Bwd Packets/s	Init_Win_bytes_foward	URG Flag Count	URG Flag Count
3	Total Backward Packets	Flow IAT Std	Init_Win_bytes_forward	CWE Flag Count

Different models appear to favor different features, it is important to note that most important features are related to derived statistical features of packets sent or abnormal flags.

ANN MODEL –SUPERVISED

Normalize feature values

```
from sklearn import preprocessing
```

```
# Normalize Value
x_final = preprocessing.MinMaxScaler().fit_transform(x)
x_test_final = preprocessing.MinMaxScaler().fit_transform(x_test)
```

Build Base Model

```
# The following variables
learning_rate = 0.01
number_epochs = 50
batch_size = 256
```

```
def create_model(my_learning_rate):
  # initial ANN
  model = Sequential()

# layers
  model.add(Dense(units = 32,  activation = 'relu', input_dim = 60))

  model.add(Dense(units = 1, activation = 'sigmoid'))

  opt = Adam(learning_rate= my_learning_rate)
  model.compile(optimizer = opt, loss = 'binary_crossentropy', metrics = ['accuracy'])

  return model
```

Build Deeper Model

```
# The following variables
learning_rate = 0.01
number_epochs = 50
batch_size = 256
```

```
22] def create_model_deep(my_learning_rate):
      # initial ANN
      model = Sequential()

    # layers
      model.add(Dense(units = 32,  activation = 'relu', input_dim = 60))
      model.add(Dense(units = 32,  activation = 'relu'))
      model.add(Dense(units = 16,  activation = 'relu'))
      model.add(Dense(units = 16,  activation = 'relu'))
      model.add(Dense(units = 1, activation = 'sigmoid'))

      opt = Adam(learning_rate= my_learning_rate)
      model.compile(optimizer = opt, loss = 'binary_crossentropy', metrics = ['accuracy'])

      return model
```

Build Deeper Model with Dropout Layer

```
# The following variables
learning_rate = 0.01
number_epochs = 50
batch_size = 256
```

```
model.add(Dense(units = 32,  activation = 'relu', input_dim = 60))
model.add(Dense(units = 32,  activation = 'relu'))
model.add(Dense(units = 16,  activation = 'relu'))
model.add(Dropout(0.1))
model.add(Dense(units = 8,  activation = 'relu'))
model.add(Dropout(0.2))
model.add(Dense(units = 1, activation = 'sigmoid'))
```

ANN MODEL – BASE MODEL

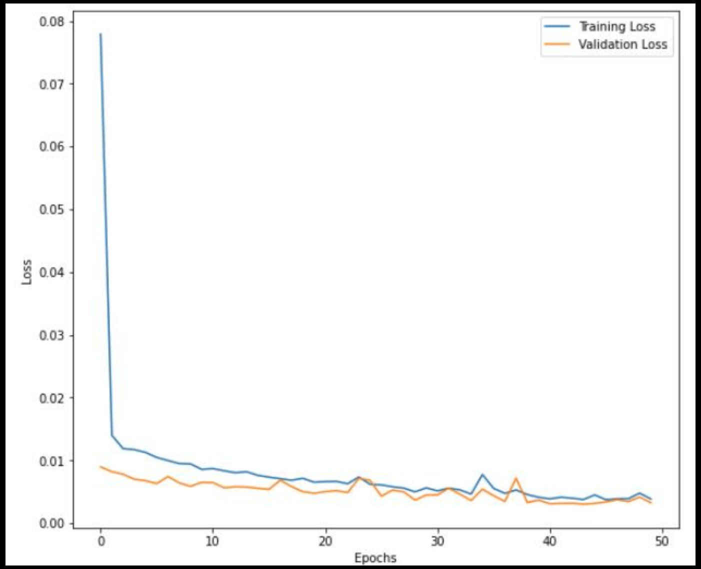
Precision

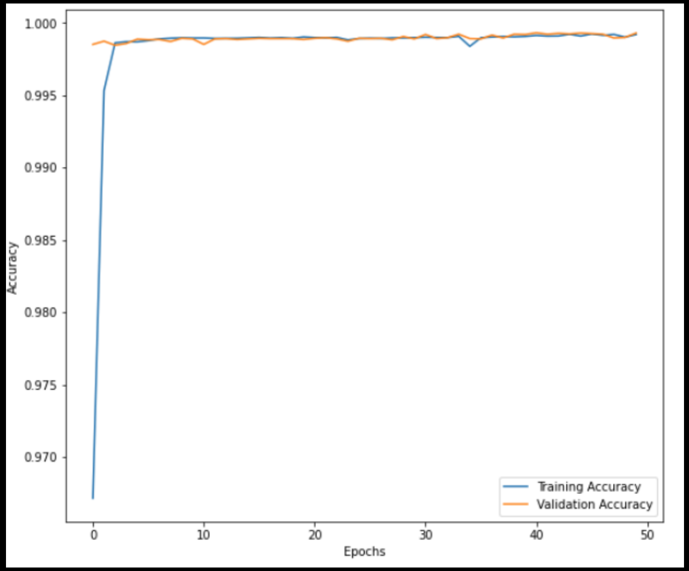
$$\mathcal{P}_c = \frac{TP_c}{TP_c + FP_c}.$$

Recall

$$\mathcal{R}_c = \frac{TP_c}{TP_c + FN_c}.$$

- No obvious model overfitting
- Loss is 0.012
- Accuracy as high as 99.7%
- For cyber security, we want the model to minimize misclassification of benign instances.
- Thus, we focus on Precision index; we want high TP and low FP.






```
 Evaluate the new model on the test set:
93/93 [==============================] - 0s 2ms/step - loss: 0.0116 - accuracy: 0.9972
loss 0.012
accuracy 0.997
```

Confusion Matrix (Test Dataset)

	Predict Benign	Predict Attack
Actual Benign	55814	492
Actual Attack	47	133821

ANN MODEL –DEEPER MODEL

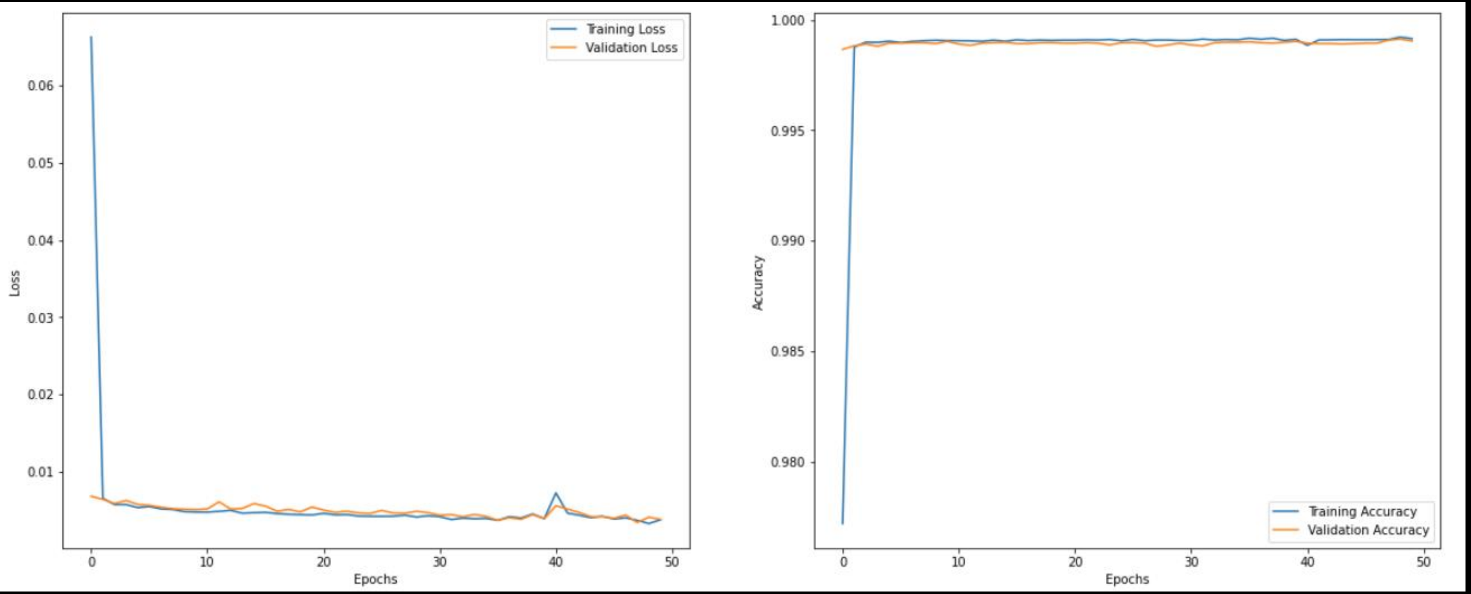
Precision

$$\mathcal{P}_c = \frac{TP_c}{TP_c + FP_c}.$$

Recall

$$\mathcal{R}_c = \frac{TP_c}{TP_c + FN_c}.$$

- No obvious model overfitting
- Loss is 0.014
- Accuracy as high as 99.9%
- We have 231 FP, which is highly reduced compared to the base model




```
 Evaluate the new model on the test set:
93/93 [==============================] - 0s 2ms/step - loss: 0.0138 - accuracy: 0.9985
loss 0.014
accuracy 0.999
```

Confusion Matrix (Test Dataset)

	Predict benign	Predict Attack
Actual Benign	56075	231
Actual Attack	46	133822

ANN MODEL – DEEPER MODEL WITH DROPOUT

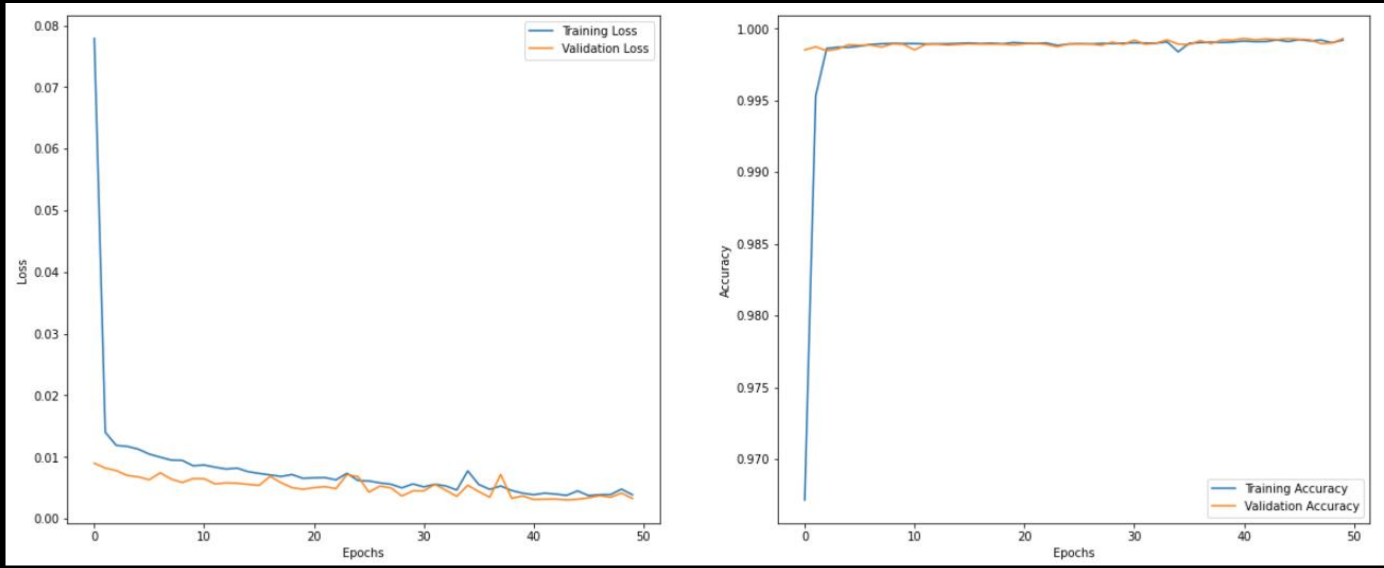
Precision

$$\mathcal{P}_c = \frac{TP_c}{TP_c + FP_c}.$$

Recall

$$\mathcal{R}_c = \frac{TP_c}{TP_c + FN_c}.$$

- No obvious model overfitting
- Loss is 0.031
- Accuracy as high as 99.9%
- For cyber security purpose, we want to minimize the total number of FPs
- Total FP was reduced to 196, lower than the earlier deeper model.




```
 Evaluate the new model on the test set:
93/93 [==============================] - 0s 2ms/step - loss: 0.0314 - accuracy: 0.9987
loss 0.031
accuracy 0.999
```

Confusion Matrix (Test Dataset)

	Predict Benign	Predict Attack
Actual Benign	56110	196
Actual Attack	46	133822

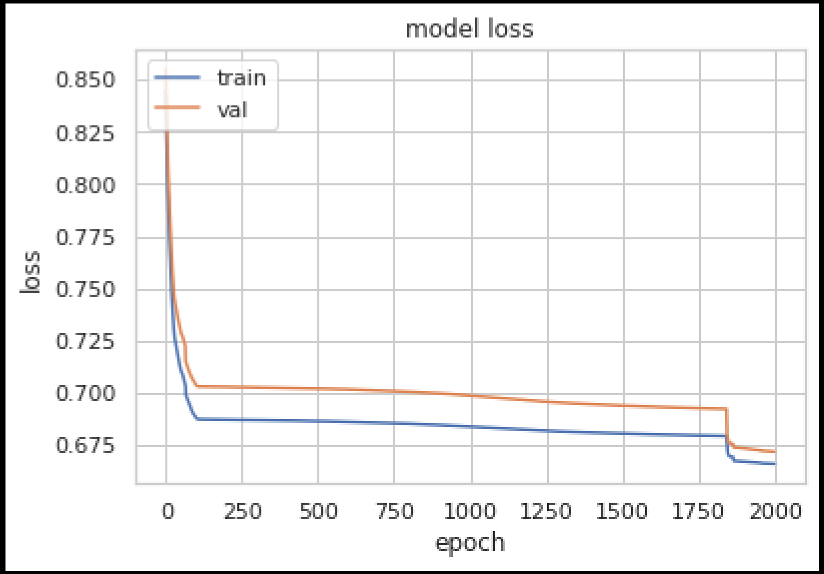
SEMI-SUPERVISED MODELS WITH ENCODERS

Encoder layers:
6 dense layers
to compresses input data to
10 dimensions

Latent space representation:
10 dimensions

Decoder layers:
6 dense layers to
reconstruct data to original
dimensionality

SEMI-SUPERVISED MODELS WITH ENCODERS



- Model was ultimately trained on 2000 epochs. Larger training time resulted in better results.
- At 2000 epochs, the error rates are still dropping, not increasing, so the model was not overfitting the data.

Classification Report			
	Precision	Recall	F1 score
Benign	0.61	0.69	0.65
Attack	0.86	0.81	0.84

Classification Report			
	Precision	Recall	F1 score
Accuracy			0.78
Macro Avg	0.73	0.75	0.74
Weighted Avg	0.79	0.78	0.78

CONCLUSION

- For cyber security, we use precision as our criteria to rank the models.
- The ANN model with deeper layers and dropout shows the precision (0.9985), which is better to correctly identify attacks. It also helps organizations to not waste resources by dealing with as many false alarms.

FUTURE WORK

- Feature Importance is helpful for predicting attack and enhancing models. Most important features are related to packets sent or abnormal flags. These features make business sense and would be closely monitored to predict attack. We can use feature engineering to further improve the models in the future.
- In future, the models can be applied in actual network by extracting features from the network nodes. The detection engine can be a module or an application running on a centralized node that can be constantly fed with the data. The trained model can raise alarm to network admin for action when the anomaly is detected. By having the detection engine in a centralized node, the attack can be detected earlier especially when the attacker is distributed in the network.


THANK YOU

REFERENCES

- https://www.unb.ca/cic/datasets/ddos-2019.html
- M. S. Elsayed, N. -A. Le-Khac, S. Dev and A. D. Jurcut, "DDoSNet: A Deep-Learning Model for Detecting Network Attacks," 2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), 2020, pp. 391-396.
- Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak, and Ali A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy", IEEE 53rd International Carnahan Conference on Security Technology, Chennai, India, 2019.
- Bawany, N., Shamsi, J., & Salah, K. (2017). DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions. Arabian Journal for Science & Engineering (Springer Science & Business Media B.V.), 42(2), 425–441..
- Cil, A. E., Yildiz, K., & Buldu, A. (2021). Detection of DDoS attacks with feed forward based deep neural network model. Expert Systems with Applications, 169, 114520–.
- Ferrag, M. A., Shu, L., Djallel, H., & Choo, K.-K. R. (2021). Deep Learning-Based Intrusion Detection for Distributed Denial of Service Attack in Agriculture 4.0. *Electronics (Basel)*, *10*(11), 1257–.