An Empirical Study on Network Anomaly Detection using Convolutional Neural Networks

July 2, 2018
(Presented at SNTA 2018)

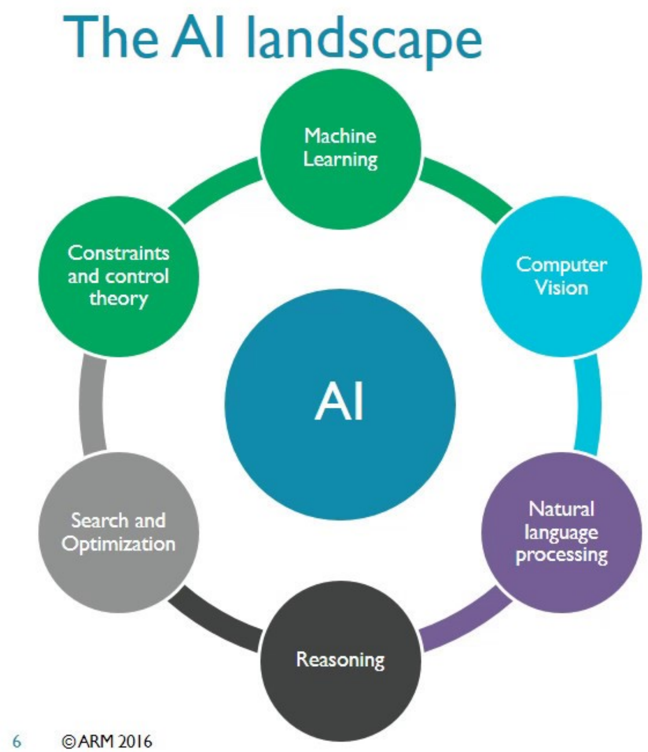
Jinoh Kim
Computer Science Department
Texas A&M University, Commerce, TX 75429, USA

What are Anomalies?

- Anomaly is a pattern in the data that does not conform to the expected behaviour
  - Outliers, exceptions, peculiarities, etc.
- Real world anomalies
  - Cyber intrusions
  - Credit card fraud

What is Machine Learning?

- A branch of **artificial intelligence**, concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data.



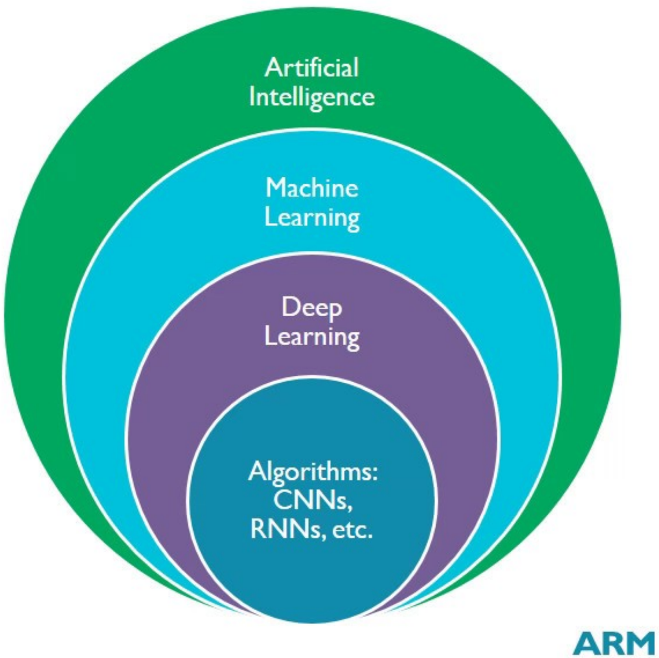
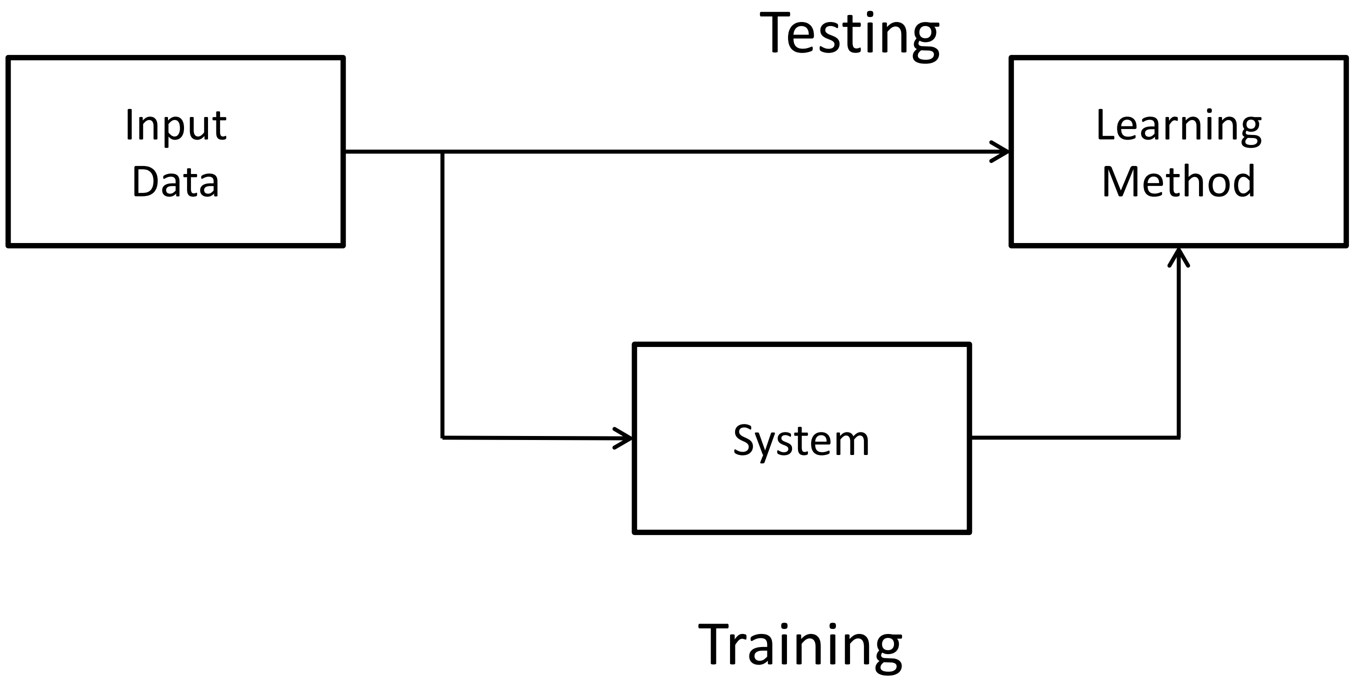


Image from Google image

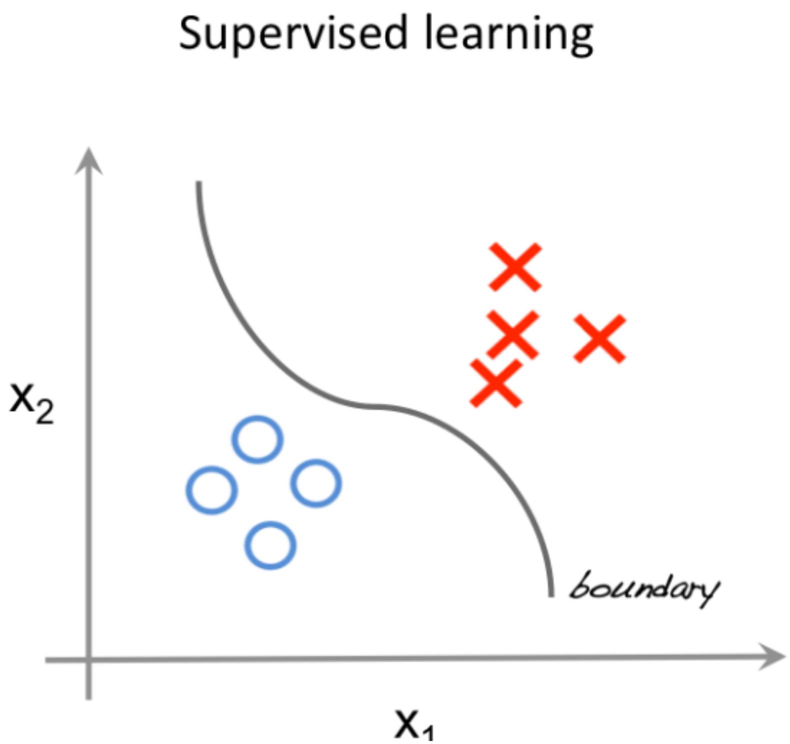
Learning System Model



Supervised vs. unsupervised

- Supervised learning
  - Provision of the associated "label"
  - Trying to "predict" a specific quantity
  - Example: neural networks, decision trees, etc
- Unsupervised learning
  - No assumption of the provision of labels
  - Trying to "understand" the data
  - Example: clustering

Supervised vs. unsupervised



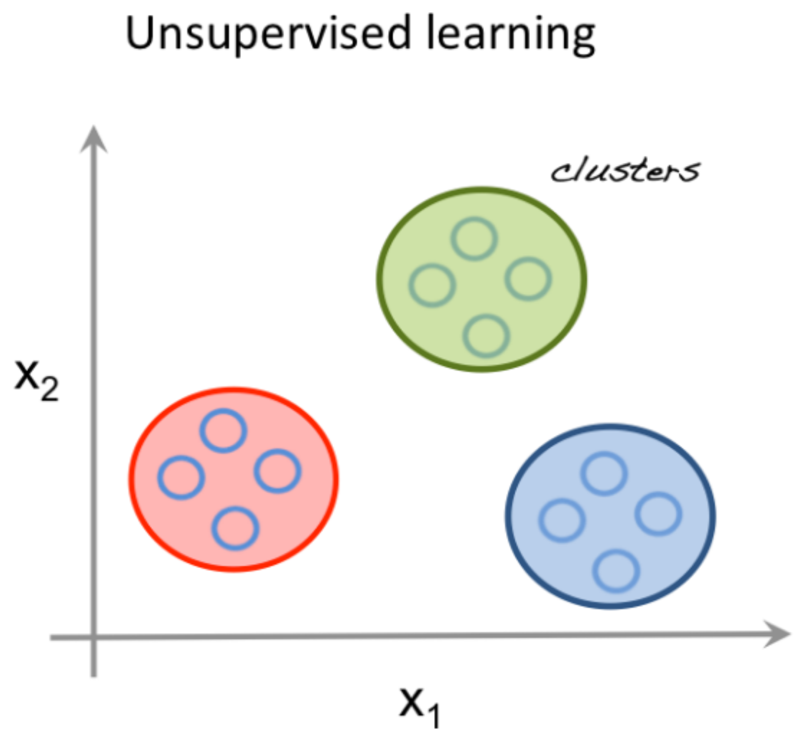
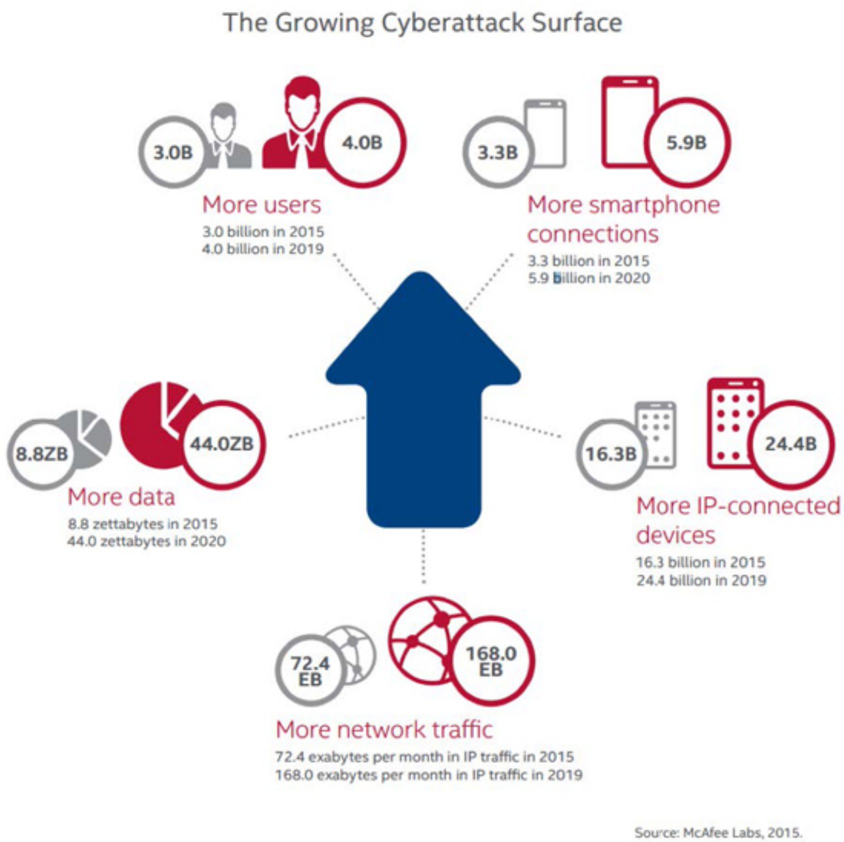


Image from https://towardsdatascience.com/unsupervised-learning-with-python-173c51dc7f03

Cyber Intrusion Detection

- Evolution of cyber-attacks
  - Growing cyber-attack surface
  - Greater scale & impacts
  - Increasingly difficult to identify
- Incidents
  - WannaCry affected 10,000 organizations in 150+ countries in May 2017
  - DDoS caused Twitter, Spotify, etc to close down in Oct. 2016
  - New type of DOS attacks utilize IoT devices (2016)
- Need more Intelligent tools to identify network anomalies



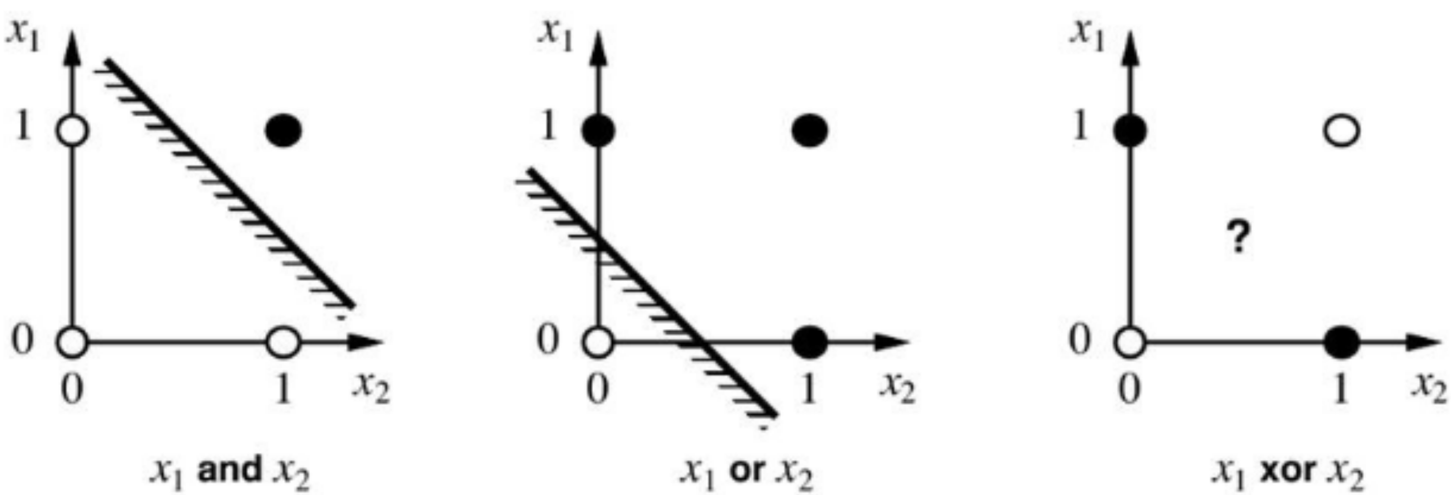
https://www.workiva.com/blog/improve-your-cybersecurity-risk-managemet

Intrusion Detection Approaches

- Misuse detection
  - Based on rules (or signatures)
  - Accurate with well-known text patterns
  - Limited due to:
    - Encryption of packets
    - Legal issue concerning privacy
- Anomaly detection
  - Based on profiling of normal and/or anomalous behaviors
  - Statistical information is widely used
    - e.g., duration, number of packets/connection, etc
  - Less accurate than signature-based detection (in general)
  - Gained greater attention with significantly improving machine learning technologies

Shallow vs. Deep ML

- The "deep" in "deep learning" refers to the number of layers through which the data is transformed. (from Wikipedia)
- Shallow learning is one other than deep learning
- Shallow learning works well for relatively simple questions (Fig. a and b)
- However, shallow learning cannot deal with a question like Fig. c
- Deep learning works better to deal with more complicated data



(a) (b) (c)

Image from https://towardsdatascience.com/radial-basis-functions-neural-networks-all-we-need-to-know-9a88cc053448

Learning-based Anomaly Detection

- Lots of studies for network anomaly detection with its advantages
  - However, using conventional ***shallow*** ML techniques is limited in accuracy to identify (< 83% accuracy)
  - E.g., SVM, random forest, Adaboosting, etc.

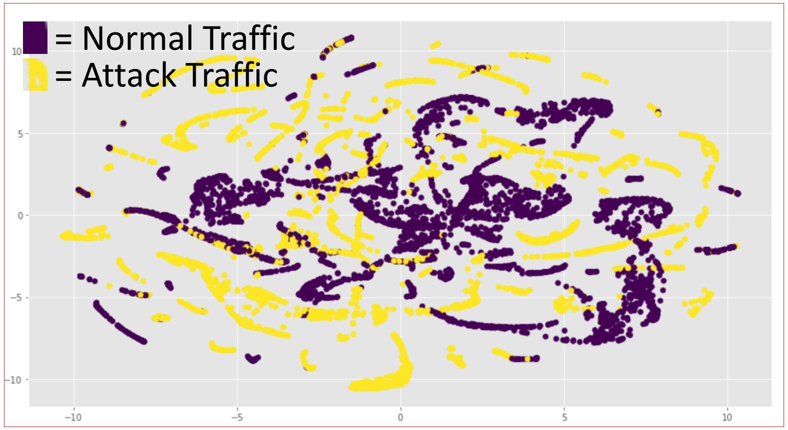
(Identification accuracy against NSL-KDD datasets)

Training	Testing	Adaboosting	SVM	Random Forest
Train-	Test+	82.5%	79.6%	78.3%
Train-	Test-	65.5%	56.5%	53.4%
Train+	Test+	80.5%	79.1%	76.1%
Train+	Test-	58.7%	56.4%	50.3%

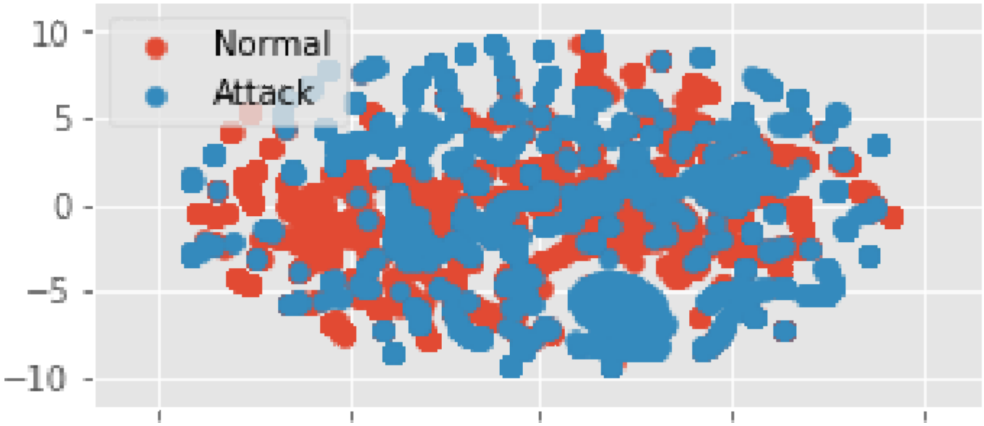
Non-linear Property

- Why not good enough with shallow ML techniques?
  - Network data sets often have non-linear property
- t-SNE (t-Distributed Stochastic Neighbor Embedding)
  - Dimension reduction tool widely employed
- t-SNE results show normal and attack data points share the same feature space
  - Hard to classify well using a shallow learning method due to non-linearity

t-SNE result against NSL-KDD



t-SNE result against Kyoto-Honeypot



Deep learning is known as good at dealing with high dimensional data with the non-linearity property

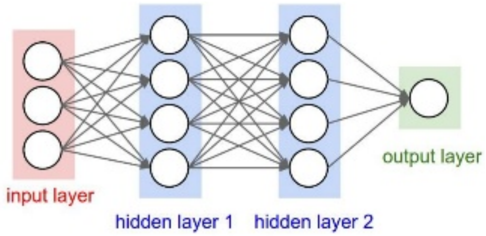
Earlier and current work

- In our earlier work:
  - We set up a set of deep learning models for network anomaly detection
    - Based on Fully Connected Neural Network (FCN), Variational AutoEncoder (VAE), and Seq2Seq structure (Seq2Seq)
  - We evaluated the deep learning models with two data sets with different characteristics wrt the population of normal and attack records
    - NSL-KDD is balanced, while Kyoto University Honeypot data is highly skewed with a lot of attack records
  - Malaiya, Ritesh K., et al. "An Empirical Evaluation of Deep Learning for Network Anomaly Detection." *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2018.

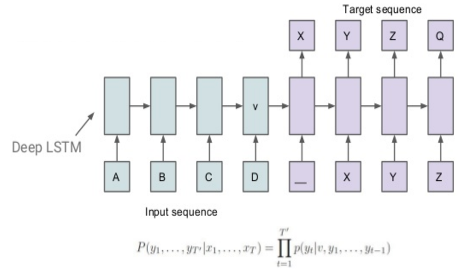
- We are currently evaluating CNN models for network anomaly detection
  - Tested CNN models taking the input as one-dimensional vector
  - But observed not that interesting results
  - Currently, we are studying on making two-dimensional input data to better use CNN models

In the earlier work …

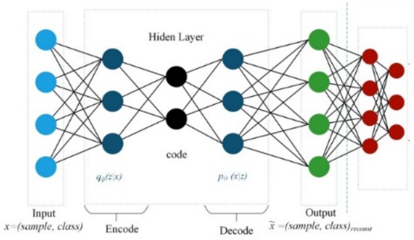
Fully Connected Network (FCN)

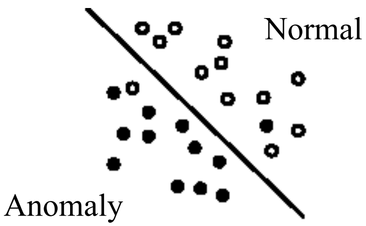


LSTM-Seq2Seq

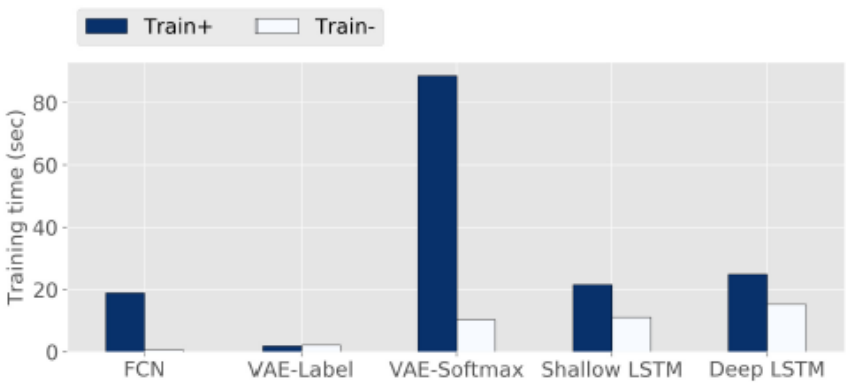


Variational Autoencoder (VAE)

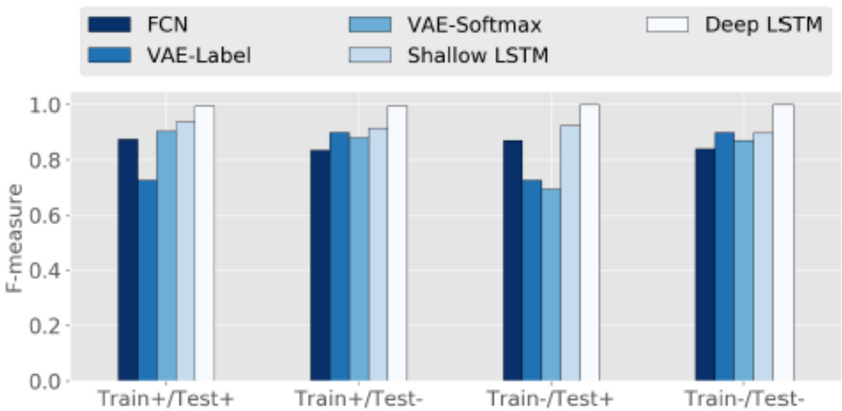




Evaluation Result: NSL-KDD



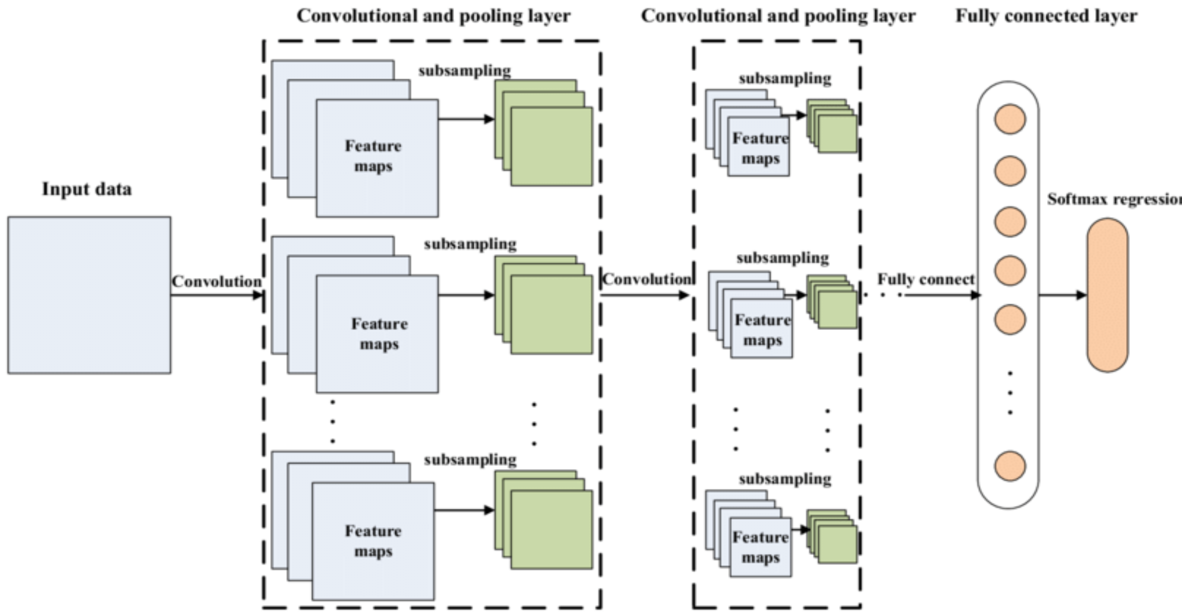
(a) Training time



(b) Performance (F-measure)

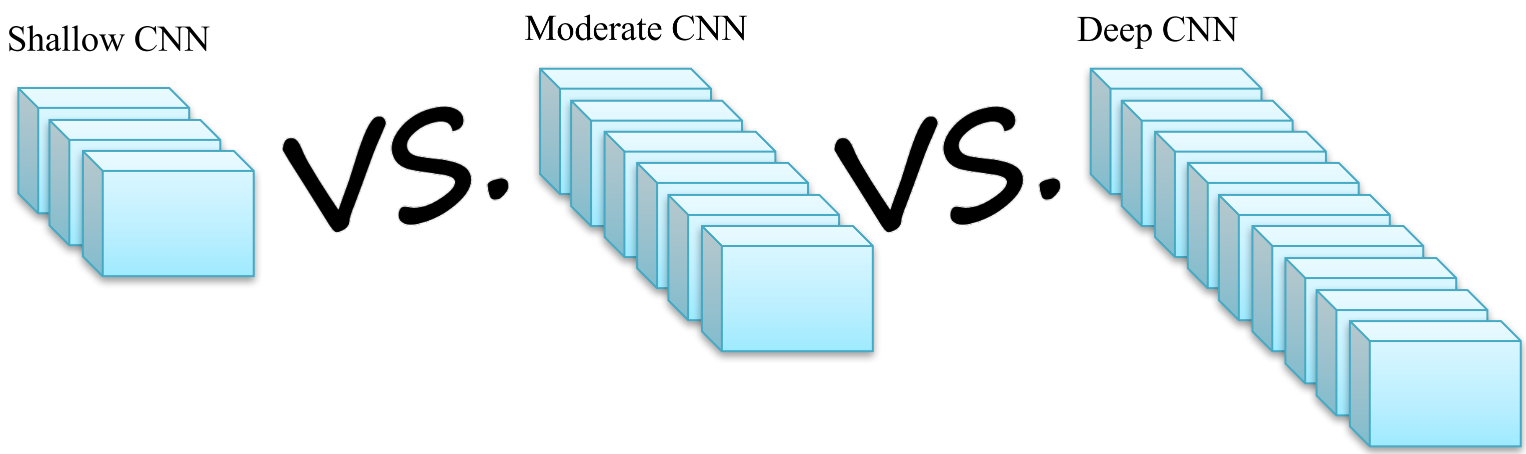
- Seq2Seq models show moderate training complexities (conducted on Google cloud)
- Seq2Seq models much outperform the others wrt anomaly detection performance
  - Deep-LSTM yields 99% of accuracy to identify for all combinations of training and testing data sets
- Seq2Seq models also work great against other network traces (Kyoto Honeypot data and MAWILab data)

What about CNN models?



In this presentation …

- This is very initial work to see the following questions:
  - Can we simply feed in a1D vector to CNN for network anomaly detection?
  - Can detection accuracy be improved once the CNN model gets deeper?



Evaluation Data Sets

- NSL-KDD data
  - Modified version of KDDCup 1999 connection data
  - Consists of 41 features with the labels
  - http://www.unb.ca/cic/datasets/nsl.html
- Kyoto-Honeypot data
  - Collected from honeypots, and thus the vast majority of records are for attacks (97% of data points)
  - http://www.takakura.com/Kyoto_data/
- MAWILab data
  - Collected from the backbone network in Japan, with the labels indicating traffic anomalies
  - http://www.fukuda-lab.org/mawilab/

NSL-KDD Data

- Modified version of KDDCup 1999 connection data
- Consists of 41 features with the associate label
  - Extended 122 features using one-hot encoding
- Four files in the dataset: 2 for training and 2 for testing

File	Description	# data points	# normal	% anomaly
Train+	Full NSL-KDD Training set	125,973	67,343	46.5%
Train20	A 20% of subset of the NSL-KDD training set	25,192	13,449	46.5%
Test+	Full NSL-KDD testing set	22,544	9,711	57%
Test-	A subset of NSL-KDD testing set	11,850	2,152	81.9%

Kyoto-Honeypot Data

- Collected from honeypots, and thus the vast majority of records are for attacks (97% of data points)
- Number of features is 24 (14 basic and 10 extended features)
  - Excluded six minor features related to the host and port information in our experiments.
  - Extended to 47 features by one-hot encoding including labels
- Kyoto-Honeypot is severely imbalanced
  - F-measure can mislead to the failure of the interpretation of results
  - MCC (Mattew Correlation Coefficient) estimates the quality of binary classification: -1.0 (poor), 0.0 (random), and 1.0 (good)

File	Dates	# records / day
Training	January 1-7, 2014	268K
Testing	December 1-31, 2015	236K

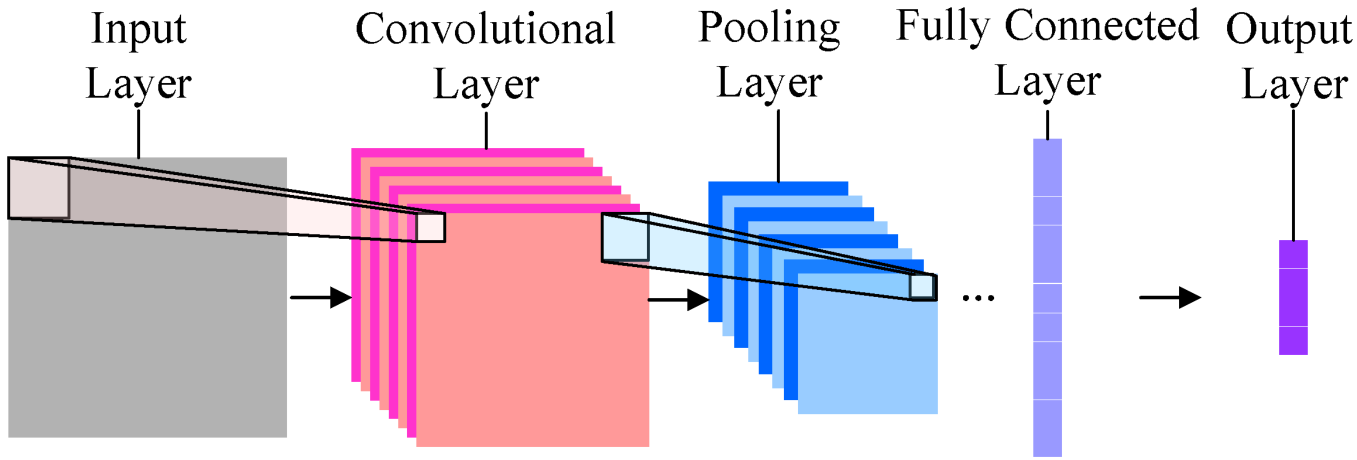
MAWILab Data

- Converted the traffic data to NetFlow format data
- 5 features out of 29 features are extracted:
  - pro, packets, bytes, durat, and status plus label
  - Categorical features: one-hot encoded

TABLE III
MAWILab dataset on August 27, 2017

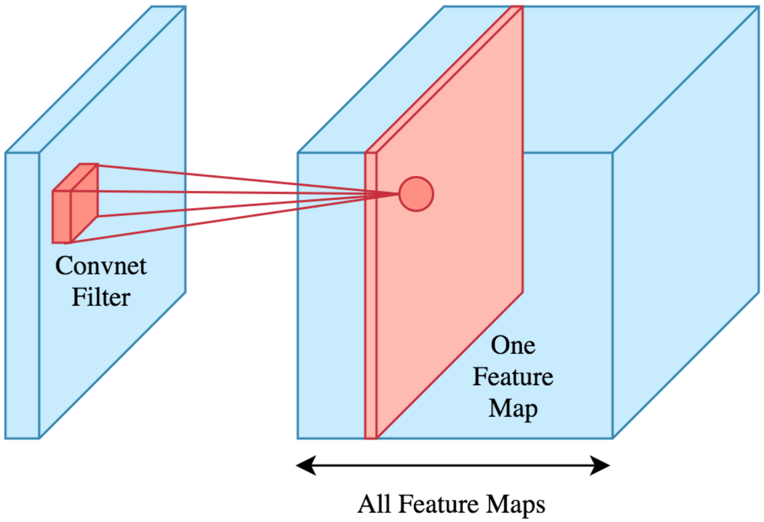
Flow	# data points	# normal	# anomaly	% anomaly
Flow 001	407,807	291,488	116,319	28.5%
Flow 002	472,654	327,413	145,241	30.7%
Flow 003	423,984	261,426	162,558	38.3%
Flow 004	425,994	300,759	125,235	29.4%

CNN framework



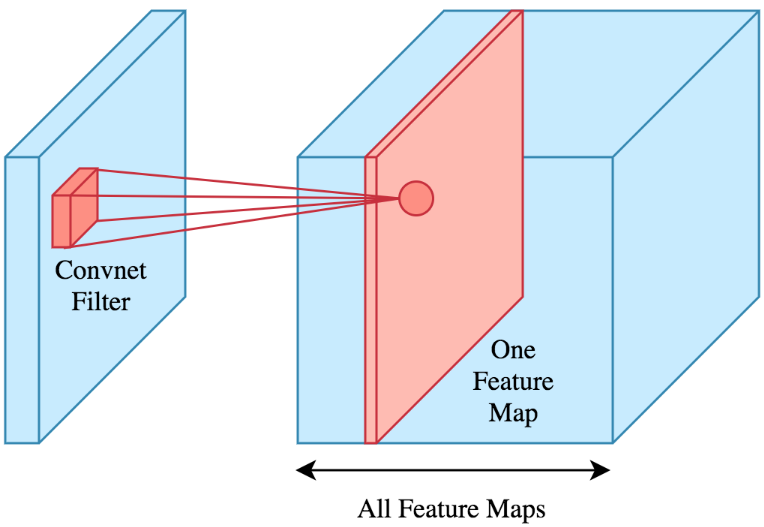
- Designed three CNN models
  - Shallow, moderate, and deep
  - Takes 1D vector as input
  - Evaluated with three different data sets (NSL-KDD, Kyoto Honeypot, MAWILab)

Feature Maps



1. Pre-processed input data are given to convolutional 1D layer(s).
   - Shallow: 1 Conv1D Layer
   - Moderate: 2 Conv1D Layers
   - Deep: 3 Conv1D Layers
2. Filters
   - Shallow: 64 filters with size 3*1
   - Moderate: 64 and 128 filters with size 3*1
   - Deep: 64, 128, and 256 filters with size 3*1
3. Stride: 2
4. Padding: Same

Feature Maps



- Pre-processed input data are given to convolutional 1D layer(s).
  - Shallow: 1 Conv1D Layer
  - Moderate: 2 Conv1D Layers
  - Deep: 3 Conv1D Layers
- Filter: size 3*1
- Batch size
  - Shallow: 64
  - Moderate: 64 and 128
  - Deep: 64, 128, and 256
- Stride: 2
- Padding: set to “same” to make outputs of the convolutional layer same as inputs.

Feature Map Calculation

1. With ReLu non-linear activation

$$h_i^k = max(w^k x_i, 0)$$

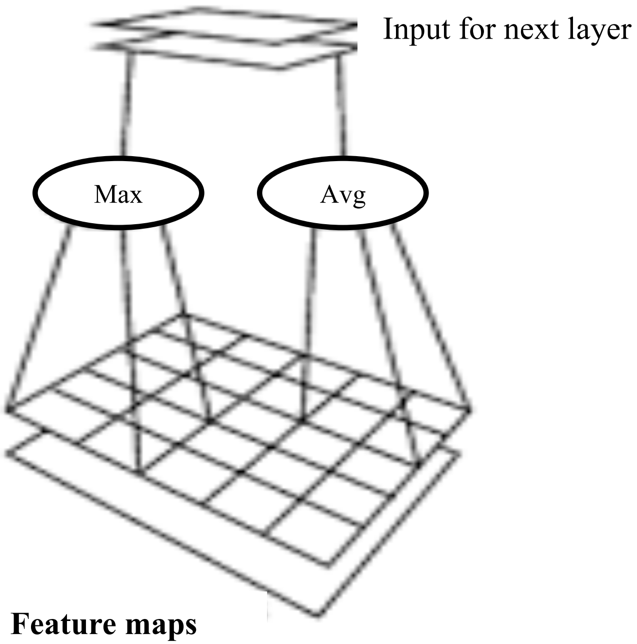
2. With t $h_i^k = tanh(w^k x_i)$

where h^k denotes the kth feature map at a given layer, i is the index in the feature map, x_i indicates the input, and w^k denotes the weights.

– No significant difference observed in our evaluation

Pooling Methods

- Pooling Layer



1. Average Pooling
   $- f_{avg}(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$

2. Max Pooling
   $- f_{max}(x) = \max(x_i)$

where x denotes a vector of input data with activation values and N indicates a local pooling region.

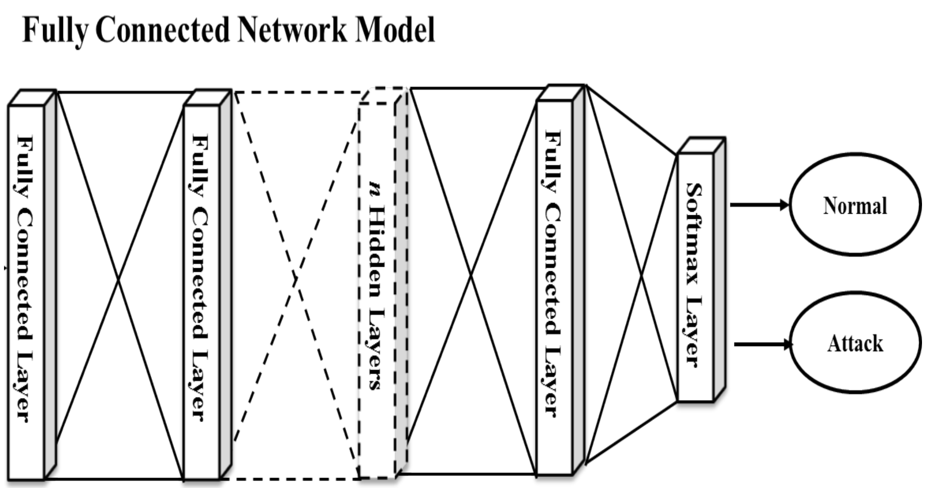
- Max Pooling is widely employed

Pooling Layers

1. Shallow CNN
   - 1 max pooling layer in the Conv1D layer
   - Flatten: 3904 for NSL-KDD, 1408 for Kyoto, and 192 for MAWILab
2. Moderate CNN
   - 1 max pooling layer in the 2^{nd} Conv1D layer
   - Flatten: 7808 for NSL-KDD, 2816 for Kyoto, and 384 for MAWILab
3. Deep CNN
   - 2 max pooling layers in the 2^{nd} and 3^{rd} Conv1D layer
   - Flatten: 7680 for NSL-KDD, 2816 for Kyoto, and 512 for MAWILab

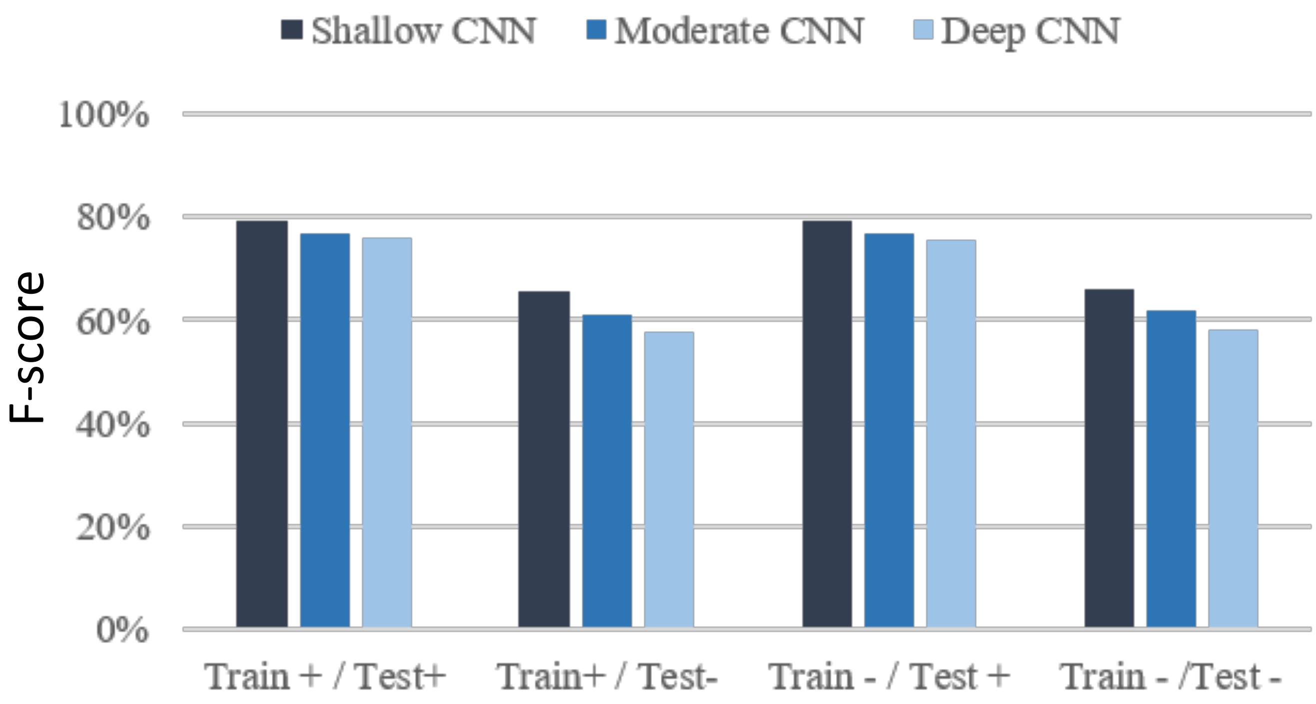
Fully Connected Network Layer

- Fully Connected Network



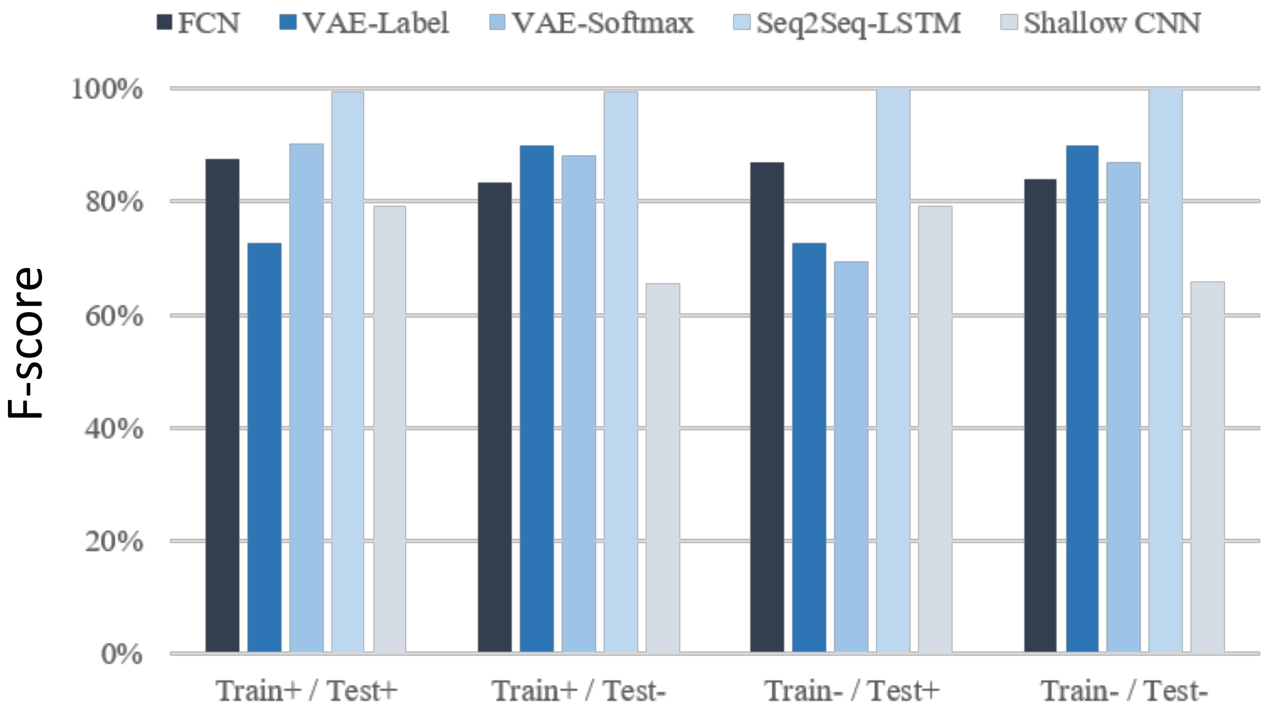
1. Hidden layers
   - Shallow: 1 hidden layer with 64 neurons
   - Moderate: 2 hidden layers with 64 and 32 neurons
   - Deep: 3 hidden layers with 64, 32, and 16 neurons
2. Other parameters
   - Batch normalization
   - Dropout = 0.5
   - Loss: binary cross entropy
   - Epochs: 10 and 20
   - Learning rate: 1e-3

Experimental Results: NSL-KDD



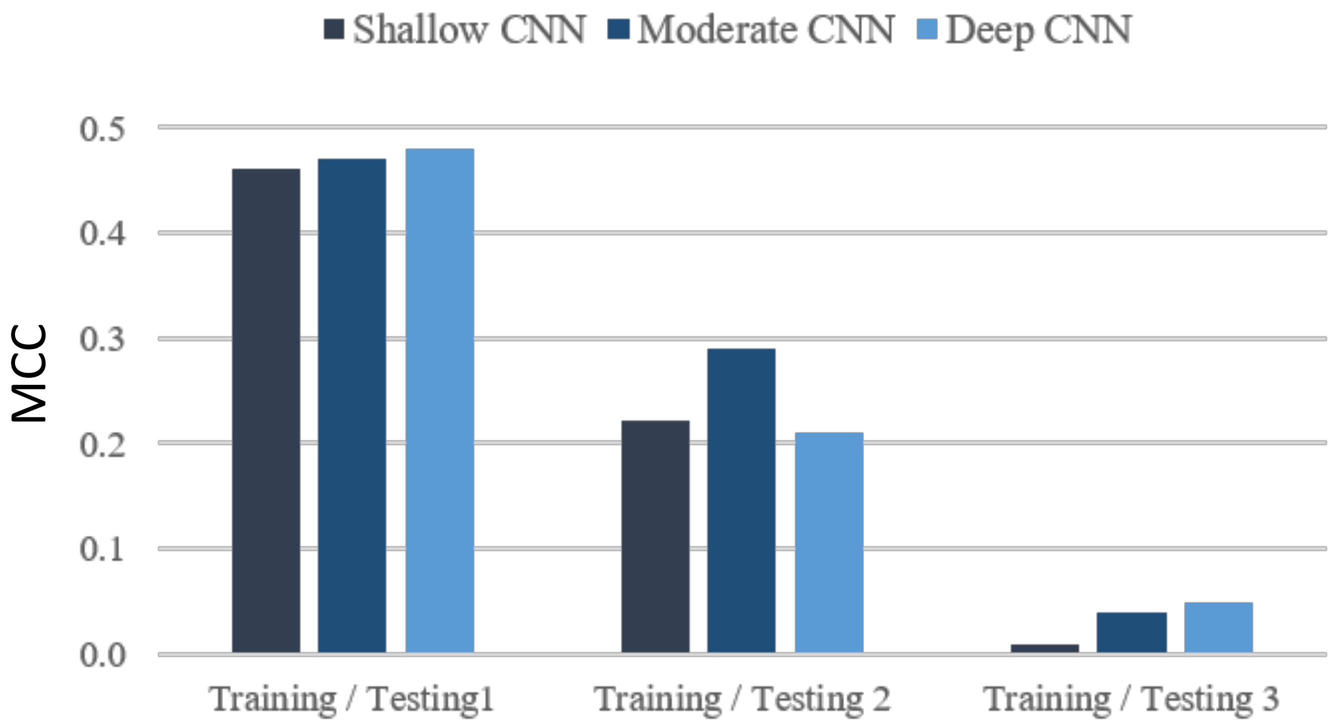
- Shows less than 80% F-measure score
- Using more layers does not improve the performance

Comparison with other DL models



- CNN model does not work better than other DL models

Experimental Results: Kyoto Honeypot



- MCC measures the quality of binary classification: -1.0 (poor), 0 (random), 1.0 (good)
- Training: January 1, 2014
- Testing: December 1 (#1), 15 (#2), 31 (#3), 2015
- Result sensitive to testing data sets

Experimental Results: MAWILab

Flow	# data points	# normal	# anomaly	% anomaly
Flow 001	407,807	291,488	116,319	28.5%
Flow 002	472,654	327,413	145,241	30.7%
Flow 003	423,984	261,426	162,558	38.3%
Flow 004	425,994	300,759	125,235	29.4%

Model	Flow 001 / Flow 002	Flow 001 / Flow 003	Flow 001 / Flow 003
Shallow CNN	65.44%	59.27%	61.33%
Moderate CNN	65.41%	67.66%	59.76%
Deep CNN	65.45%	67.86%	56.76%

- F-score ranges between 56% - 68%, which are not that satisfactory

Summary

- Deep learning is essential for network anomaly detection in considering the characteristics of network traffic data with a high degree of non-linearity
- Evaluated three simple CNN models taking a 1D vector as input with different internal depths
- The CNN models with 1D vector as input does not work better than other deep learning structures
- Also simply adding more layers would not be helpful to improve the detection performance

Future Direction

- Plan to develop a 2D construction method for the input data format from network data
- Previous work converted a NSL-KDD record into a 8x8x1 grayscale image
  - Converting is based on binning and one-hot encoding
  - Reported ~90% accuracy with existing CNN models (ResNet and GoogLeNet)
  - Li, Zhipeng, et al. "Intrusion Detection Using Convolutional Neural Networks for Representation Learning." *International Conference on Neural Information Processing*. Springer, Cham, 2017
- Need to design of CNN models taking 2D matrix as input, optimized for network anomaly detection

Questions?

Contact: Jinoh.kim@tamuc.edu