Network Intrusion Detection

Kai Wong

Introduction

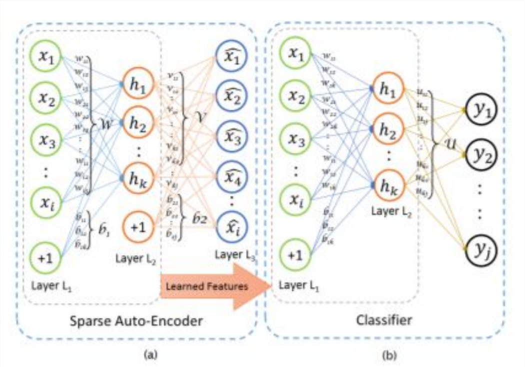
- Cybersecurity
  - How to better protect a company’s network
  - Network Intrusion Detection System (NIDS)
  - Harnessing power of deep learning and neural networks
    - Accurate detection/classification for different attacks

Dataset

- KDDCup99 Dataset
  - Annual ACM Data Mining and Knowledge Discovery competition
  - Task: distinguish between connections that are bad (intrusions) and good (normal)
  - 4 categories of attacks, 14 attack types with add. 14 types in test
  - NSL-KDDCup99
    - Improved and reduced

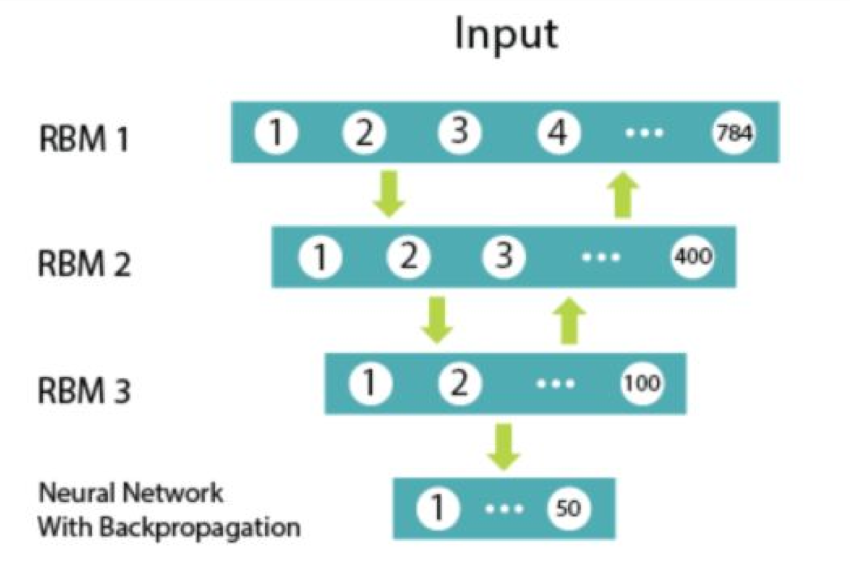
Past Work in NIDS

- Deep Belief Networks
  - Have been used for classification in intrusion detection [1,2]
  - Done with sparse auto-encoder and layers of RBMs
    - Labelling dataset from raw network traffic = difficult
    - Append with softmax regression [1]
  - Proper/meaningful feature selection key and difficult
- RNNs/LSTMs
  - Accurate models built [3,4,5,6]
  - Sequential info
    - Single pt in collective data anomaly may not seem as an anomaly



Methodology

- Deep Belief Network
  - Reduce dimensionality of dataset and pretrain, feature selection
    - Using layers of RBMs
  - Pass through a neural network for classification
    - Shallow general neural network



Data Transformations

- Transform categorical data columns to a 1HE
- Split label columns from data columns
- Map values between 0 - 1 and remove useless columns
    - RBM code originally for MNIST dataset (bounded values)

Data Transformation

duration	protocol	service	flag	src_byte:	dst_byte:	land	wrong_fr	urgent	hot	num_fail	logged_i	num_cor	root_she	su_atten	num_roc	num_file_	num_she	num_acc	num_out	is_host_	is_guest_	count	srv_cour	serror_ra	srv_serr	rerror_ra	srv_rerrc	same_sr	diff_srv_	srv_diff_	dst_host	dst_host	dst_host	dst_host	dst_host	dst_host	dst_host	dst_host	dst_host	dst_host	label	idk
0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	1	0	0	150	25	0.17	0.03	0.17	0	0	0	0.05	0	normal	20
0	udp	other	SF	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	1	0	0	0	0	0.08	0.15	0	255	1	0	0.6	0.88	0	0	0	0	0	normal	15
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	6	1	1	0	0	0.05	0.07	0	255	26	0.1	0.05	0	0	1	1	0	0	neptune	19
0	tcp	http	SF	232	8153	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	5	5	0.2	0.2	0	0	1	0	0	30	255	1	0	0.03	0.04	0.03	0.01	0	0.01	normal	21
0	tcp	http	SF	199	420	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	30	32	0	0	0	0	1	0	0.09	255	255	1	0	0	0	0	0	0	0	normal	21
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	121	19	0	0	1	1	0.16	0.06	0	255	19	0.07	0.07	0	0	0	0	1	1	neptune	21
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166	9	1	1	0	0	0.05	0.06	0	255	9	0.04	0.05	0	0	1	1	0	0	neptune	21
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	117	16	1	1	0	0	0.14	0.06	0	255	15	0.06	0.07	0	0	1	1	0	0	neptune	21
0	tcp	remote_j	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	270	23	1	1	0	0	0.09	0.05	0	255	23	0.09	0.05	0	0	1	1	0	0	neptune	21
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	133	8	1	1	0	0	0.06	0.06	0	255	13	0.05	0.06	0	0	1	1	0	0	neptune	21
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	205	12	0	0	1	1	0.06	0.06	0	255	12	0.05	0.07	0	0	0	0	1	1	neptune	21
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	199	3	1	1	0	0	0.02	0.06	0	255	13	0.05	0.07	0	0	1	1	0	0	neptune	21
0	tcp	http	SF	287	2251	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	7	0	0	0	0	1	0	0.43	8	219	1	0	0.12	0.03	0	0	0	0	normal	21
0	tcp	ftp_data	SF	334	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	1	0	0	2	20	1	0	1	0.2	0	0	0	0	warezclie	15
0	tcp	name	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	233	1	1	1	0	0	0	0.06	0	255	1	0	0.07	0	0	1	1	0	0	neptune	19
0	tcp	netbios_	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96	16	1	1	0	0	0.17	0.05	0	255	2	0.01	0.06	0	0	1	1	0	0	neptune	18
0	tcp	http	SF	300	13788	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	9	0	0.11	0	0	1	0	0.22	91	255	1	0	0.01	0.02	0	0	0	0	normal	21
0	icmp	eco_i	SF	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	1	16	1	0	1	1	0	0	0	0	ipsweep	18

Post-1HE shape: (125973, 146)
Post-1HE and label removal shape: (125973, 122)

RBMs

- Unsupervised greedy pre-training
- Dimensionality reduction
- Training 3 layers, 50 epochs
    - First layer: 40 hidden units
        - 43 initial features in dataset
    - Second layer: 20 hidden units
    - Third layer: 10 hidden units

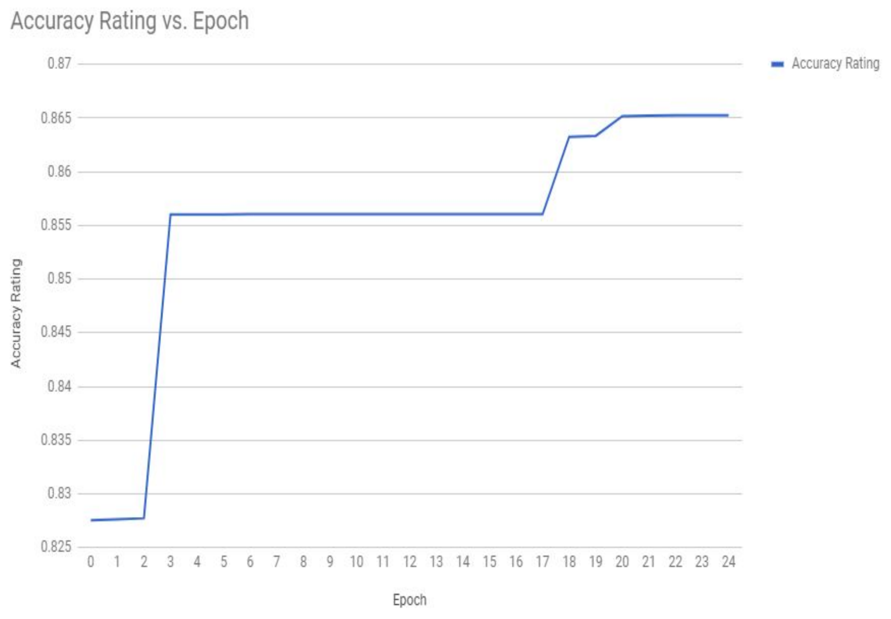
Shallow Neural Network

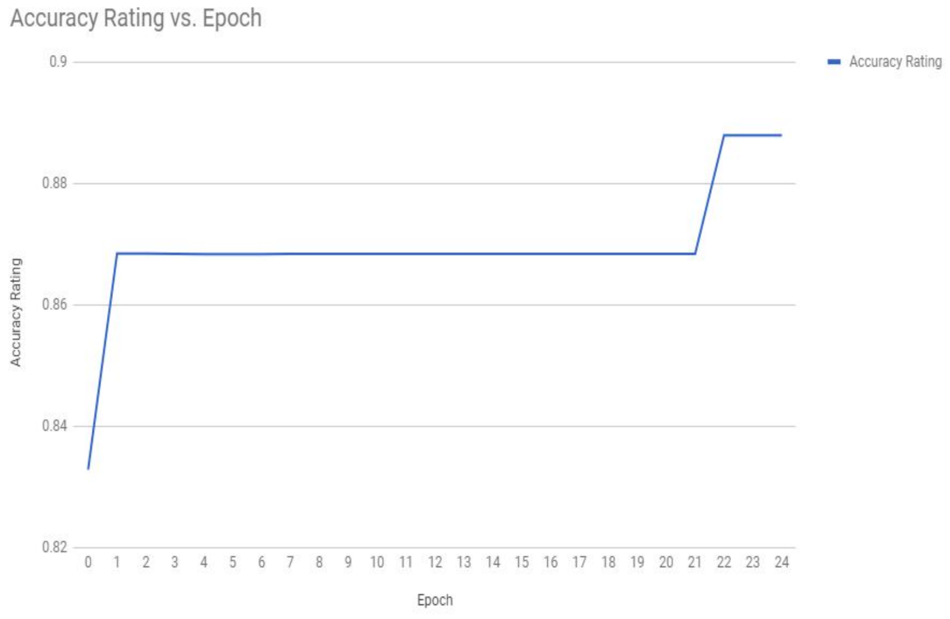
- Take output from last layer of pre-trained RBMs
- Use output for classification in shallow neural net with backprop

Results

- RBM layer 1 final reconstruction error: 0.011173
- RBM layer 2 final reconstruction error: 0.009147
- RBM layer 3 final reconstruction error: 0.019889
- Final accuracy rating over 25 epochs: **0.89**

Results


Accuracy Rating vs. Epoch
Accuracy Rating
0.87
0.865
0.86
0.855
0.85
0.845
0.84
0.835
0.83
0.825
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Epoch


Accuracy Rating vs. Epoch
Accuracy Rating
0.9
0.88
0.86
0.84
0.82
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Epoch

Continued Work for Final Model

- Vary training of RBMs and NN to increase testing accuracy
    - epochs, learning rates, hidden units
- Use an RNN/LSTM for classification as opposed to shallow general neural network