The Applications of Deep Learning on Traffic Identification

Zhanyi Wang

whoami

- Zhanyi Wang, Signal & Information Processing, Ph.D.
- Data mining researcher @ Qihoo 360
- Machine Learning - rich experience/Cyber Security – beginner
- Colleagues
  - Zhuo Zhang, Bo Liu, Chuanming Huang
- Focus on "Data-driven Security“
  - Statistical Analysis
  - Deep Learning
  - Pattern Recognition
  - Anomaly Detection

Outline

- Traditional Methods of Traffic Identification
- Neural Networks and Deep Learning
- Applications
  - Protocol Classification
  - Automatic Feature Learning
  - Application Identification
  - Unknown Protocol Identification
- Conclusions and Future Work

Traditional Methods of Traffic Identification

- An accurate mapping of traffic to protocols or applications is important for network management, anomaly detection
- Base on special or predefined ports
  - Standard HTTP port is 80, default port of SSL is 443
  - Weakness: doesn’t work when ports are new or changed
- Signature-based traffic identification
  - Static, dynamic and distinguishable features
  - Weakness: very time-consuming and labor-intensive

HTTP?

SSL?

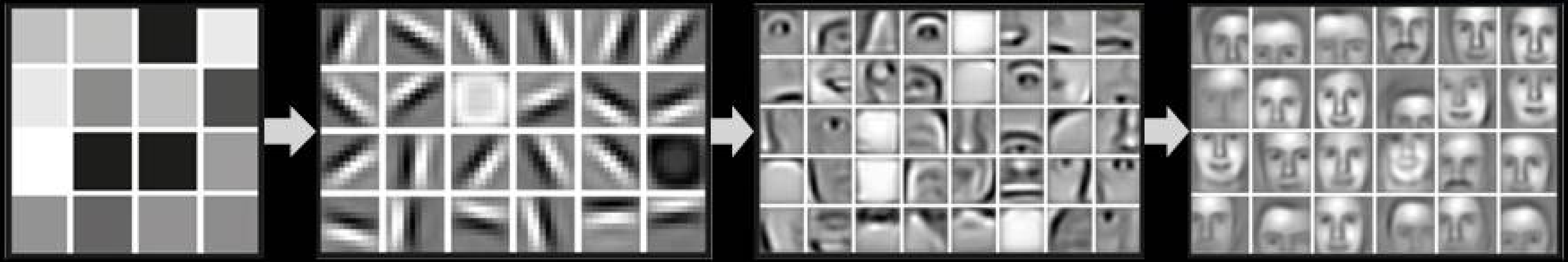
Why we choose deep learning

- Base on statistical features and machine learning
    - Identification process: automatic
    - Difficulty: how to choose appropriate features

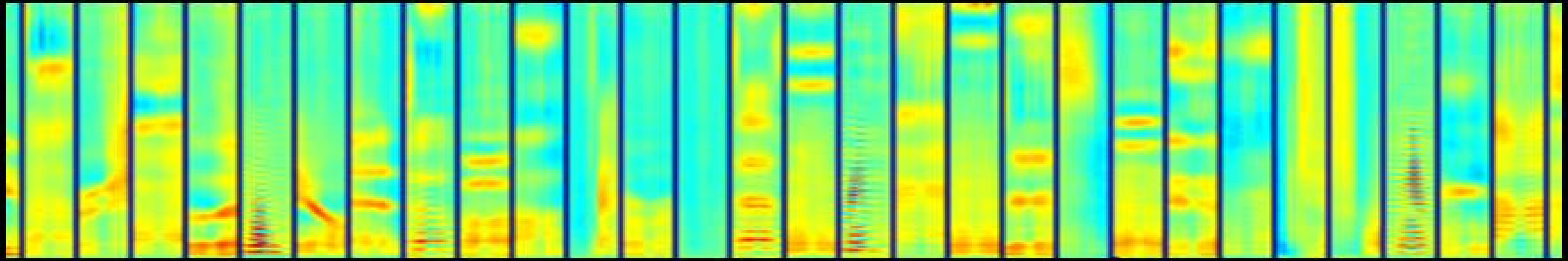
- Is there any ways not to depend on experts?
- Is unsupervised feature learning possible?
- Answer: Deep Learning in artificial intelligence

The power of deep learning techniques

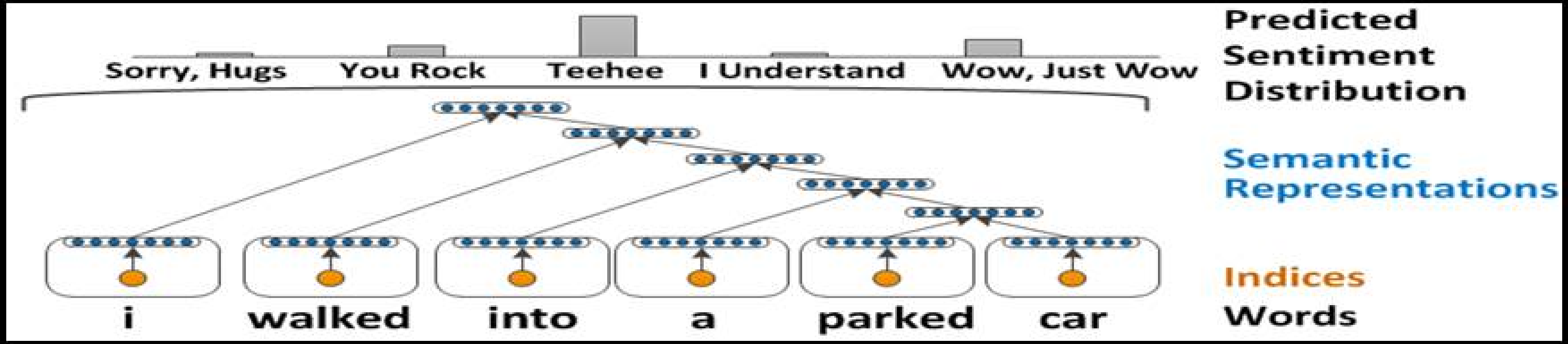
- Image



- Speech

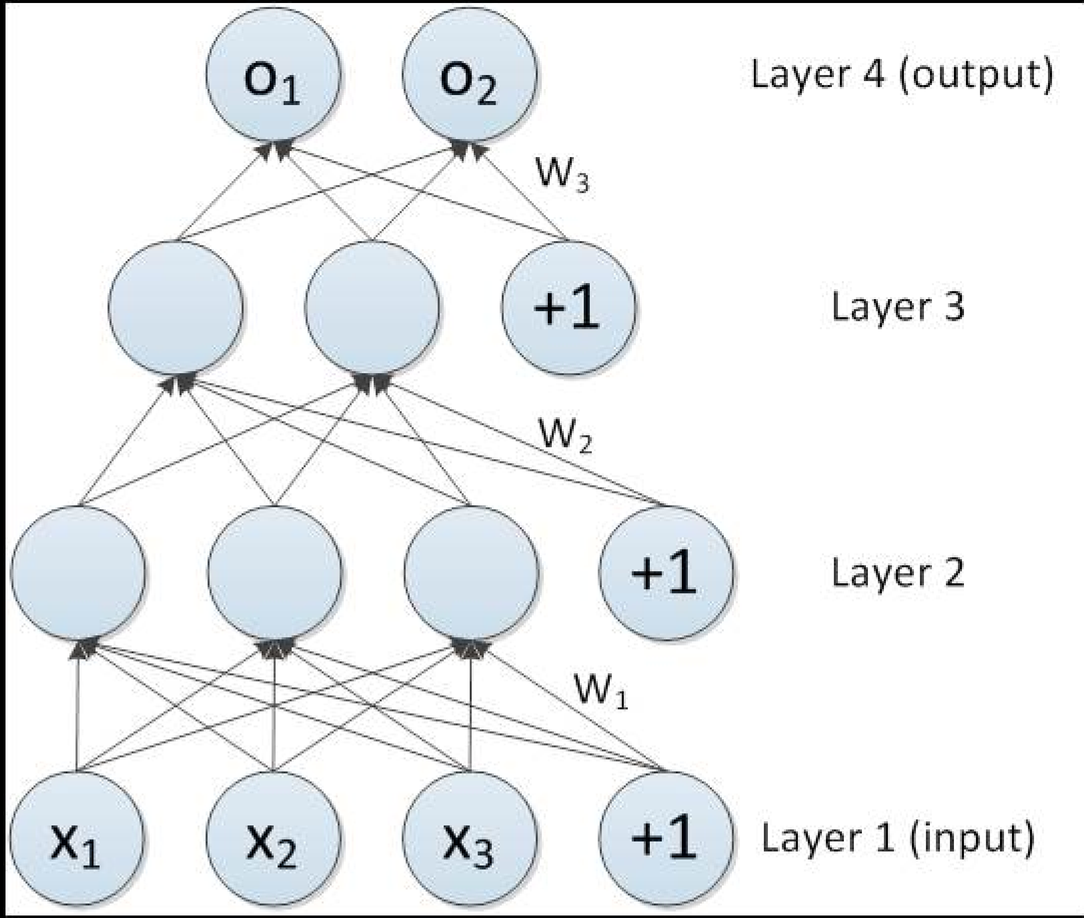


- NLP



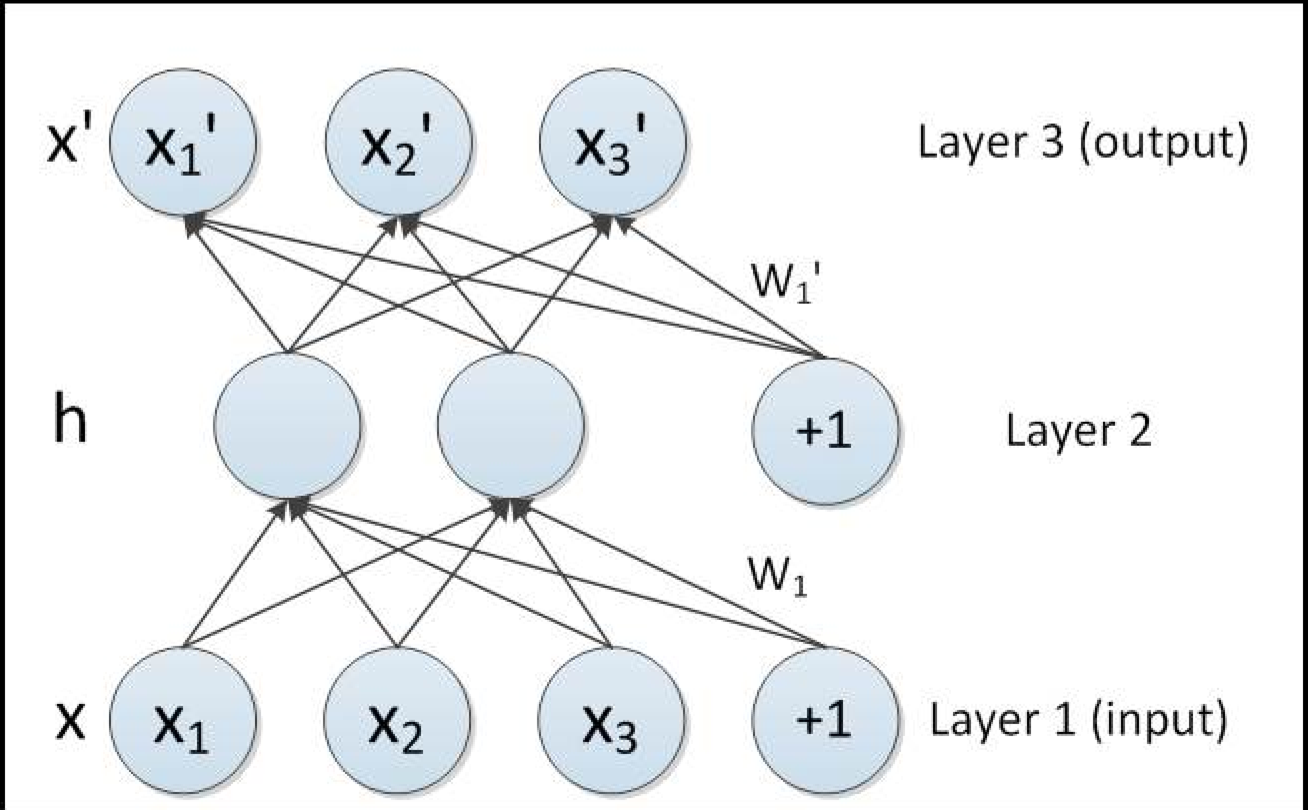
Neural Networks

- Neural Networks
- Basic unit
  - neuron
- Structure:
  - Input layer
  - Hidden layers
  - Output layer
- Each pair of neighboring layers is connected
- neurons in the same layer are not connected



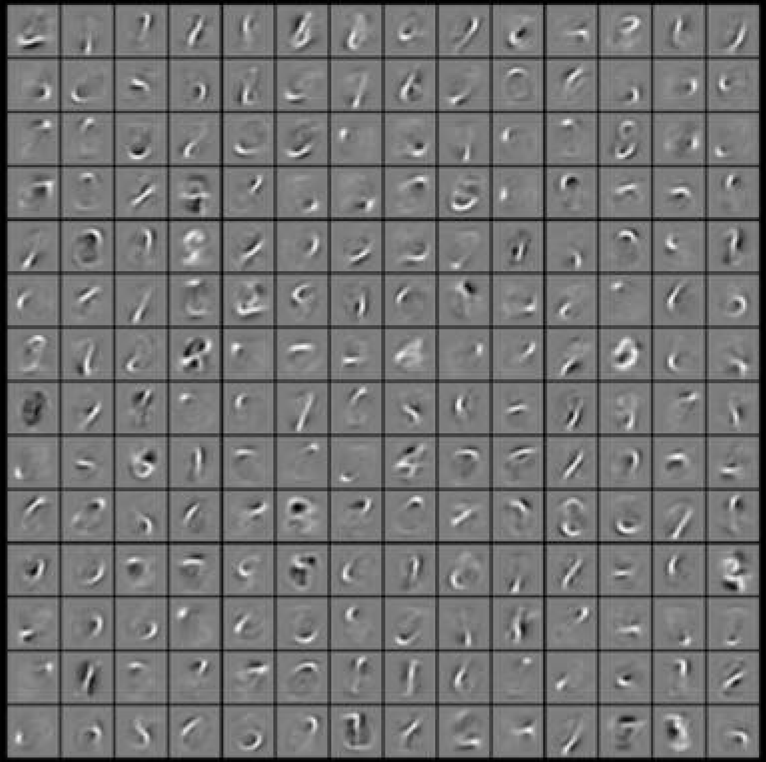
Auto-Encoder

- Auto-Encoder
- a specific type of neural network
- Only one hidden layer
- Output layer is identical with input layer!



Auto-Encoder in image recognition

- handwritten digits experiment



Stacked Auto-Encoder

- Stacked Auto-Encoder (SAE)
- Consisting of multiple layers of AE
- SAE is a neural network essentially

- Use greedy layer-wise training
- Use fine-tuning

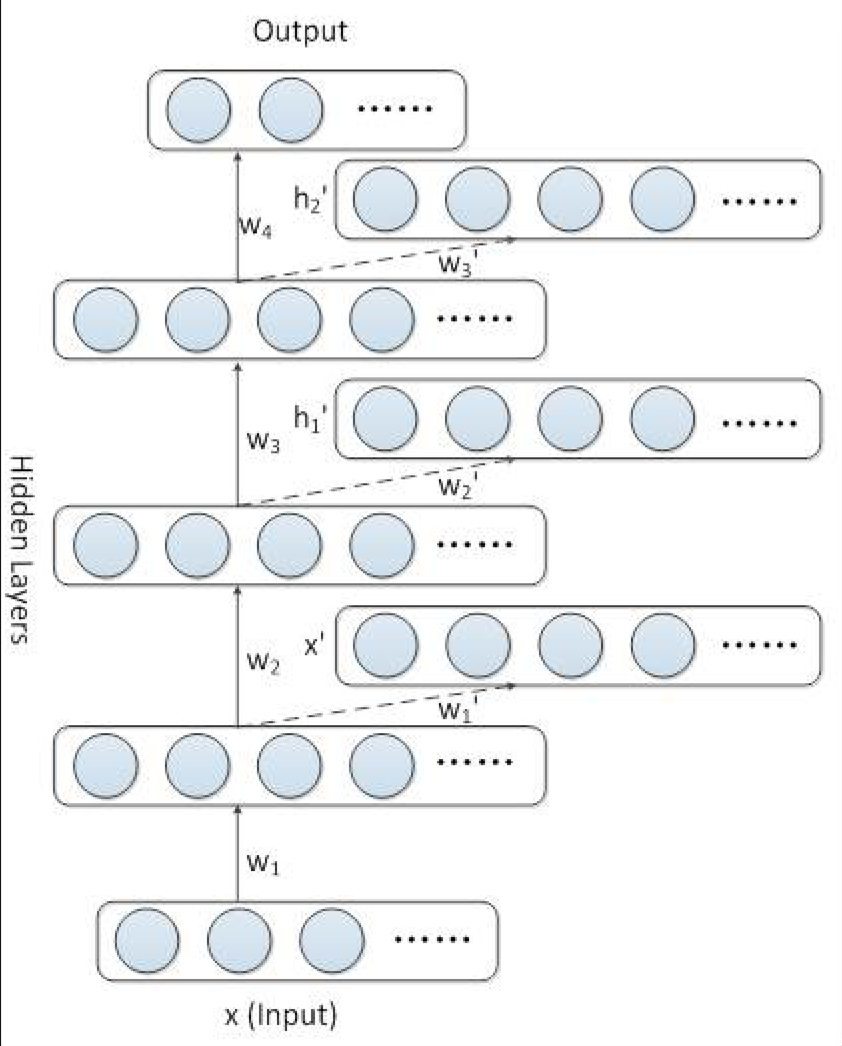
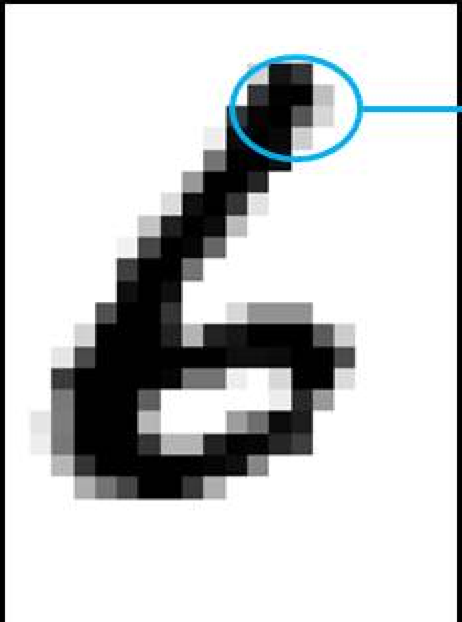
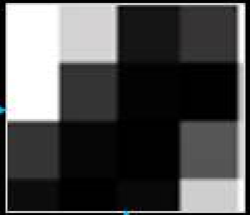


Image VS Payload

- Do they look alike?





255	210	21	53 …
255	52	3	0 …
52	6	0	85 …
…	…	…	…

TCP flow Payloads

474554206874……727665720020……732048545450……33a31353a323……

732048545450……33a31353a323……

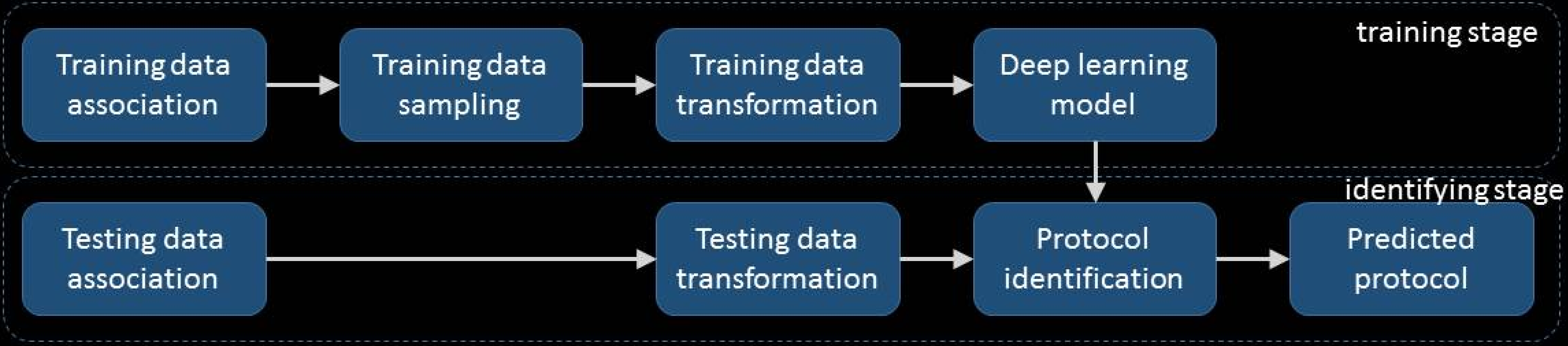
115 32 72 84 84 80……51 163 19 83 163 35……

range of values: [0,255]

Both 256 numbers!

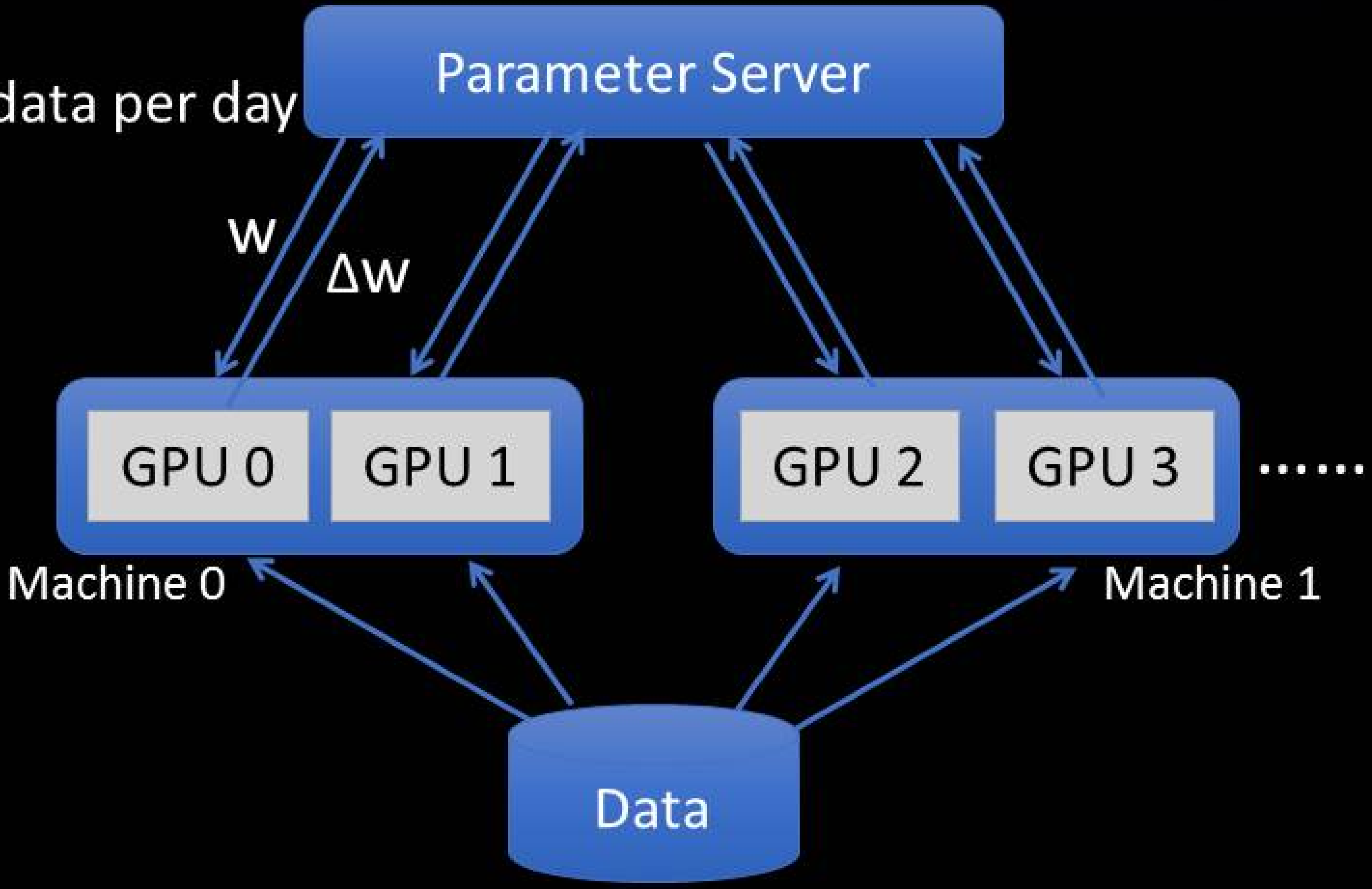
Implementation of protocol identification

- Data: collected in our intranet
- Experimental environment
  - Scheme 1 - CPU: E5-2630 * 2 + GPU: AMD S9150 * 4
  - Scheme 2 - only use CPU cluster: 2~10 servers
- Training time: less than 3 hours in Scheme 1



Parallel computing based on multi-GPU

- Large amount of data
  - Hundreds of millions original data per day
- Too many parameters
  - More than 5 millions
- Very long training time
  - Several days if just use CPU
- Solution
  - OpenCL
  - Multi-GPU
  - Multi-machine



*OpenCL is a framework for writing programs that execute across heterogeneous platforms consisting of CPUs, GPUs, DSPs, FPGAs and other processors.

Data association & transformation

up_payload1		down_payload1		up_payload2		down_payload2	
474554206874……	+	727665720020……	+	732048545450……	+	33a31353a323……	……

concatenate together

474554206874……727665720020……732048545450……33a31353a323……

Length = 1024

HEX to DEC

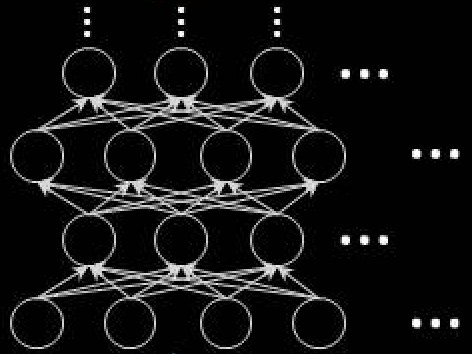
71 69 84 32 104 116……114 118 101 114 0 32……115 32 72 84 84 80……51 163 19 83 163 35……

The process of protocol identification

0.85, 0.6, 0.02, 0.00, 0.01, 0.00 ......

How to predict?

Deep Learning Model



71 69 84 32 104 116......114 118 101 114 0 32......115 32 72 84 84 80......51 163 19 83 163 35......

116 199 225 220 82 116......14 211 51 17 37 110......139 18 253 58 80 172......172 26 91 146 1 23......

180 39 27 205 22 76......226 123 177 230 163 14......77 76 150 167 3 237......183 9 78 44 30 162......

......

The process of protocol identification

Logistic Regression

More than one outputs

0.85, 0.6, 0.02, 0.00, 0.01, 0.00 ......

MySQL, SSH, FTP_CONTROL, HTTP_Proxy, SMB, SMTP ......

>0.5?

Predictions: 1. MySQL 2.SSH

Softmax Regression

Just one output

0.91, 0.01, 0.02, 0.00, 0.01, 0.00 ......

MySQL, SSH, FTP_CONTROL, HTTP_Proxy, SMB, SMTP ......

Maximum?

Prediction: MySQL

Protocol Classification

- Overall Precision: >99% Average Precision: 97.9%

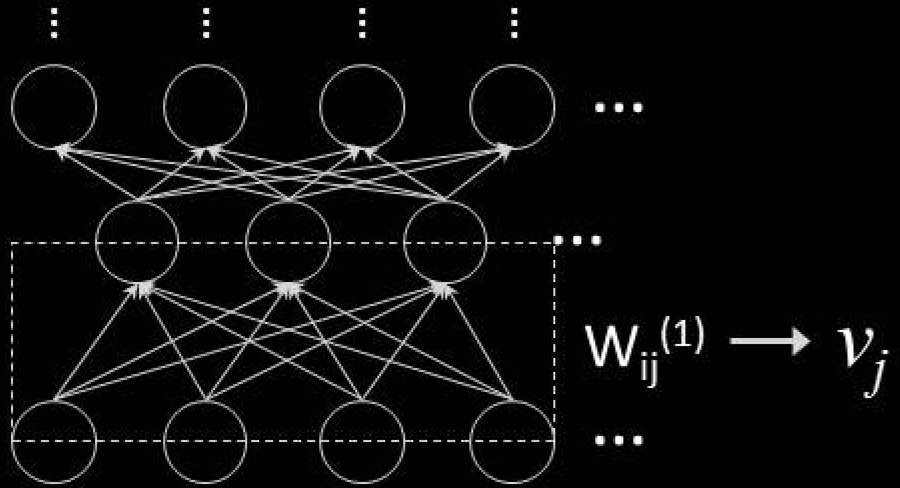
Protocol	Precision	Protocol	Precision
SMB	1.0000	RSYNC	0.9987
DCE_RPC	1.0000	Redis	0.9985
NetBIOS	1.0000	FTP_CONTROL	0.997
TDS	1.0000	HTTP_Connect	0.9967
SSH	0.9996	SMTP	0.9949
Kerberos	0.9996	Whois-DAS	0.9943
LDAP	0.9996	IMAPS	0.9814
BitTorrent	0.9992	Apple	0.964
MySQL	0.9989	SSL	0.9513
DNS	0.9989	HTTP_Proxy	0.9174

Automatic Feature Learning

- take the sum of all absolute weights $|W_{ij}^{(1)}|$ with regard to every node in the input layer as the value

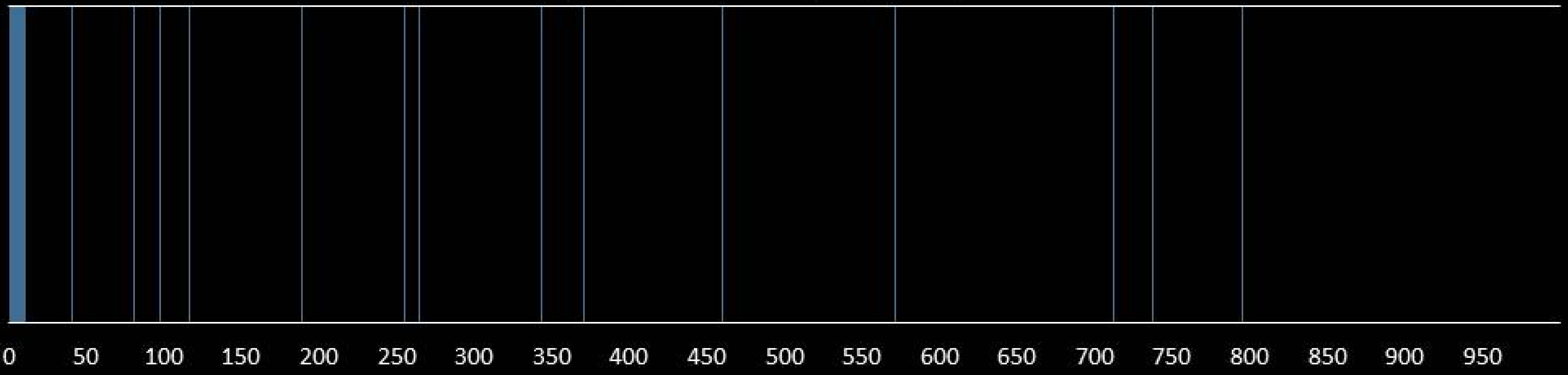
$$v_j = \sum_{i=1}^{n} \left| w_{ij}^{(1)} \right|$$

- v_j: the larger, the more important the *j*-th feature is.

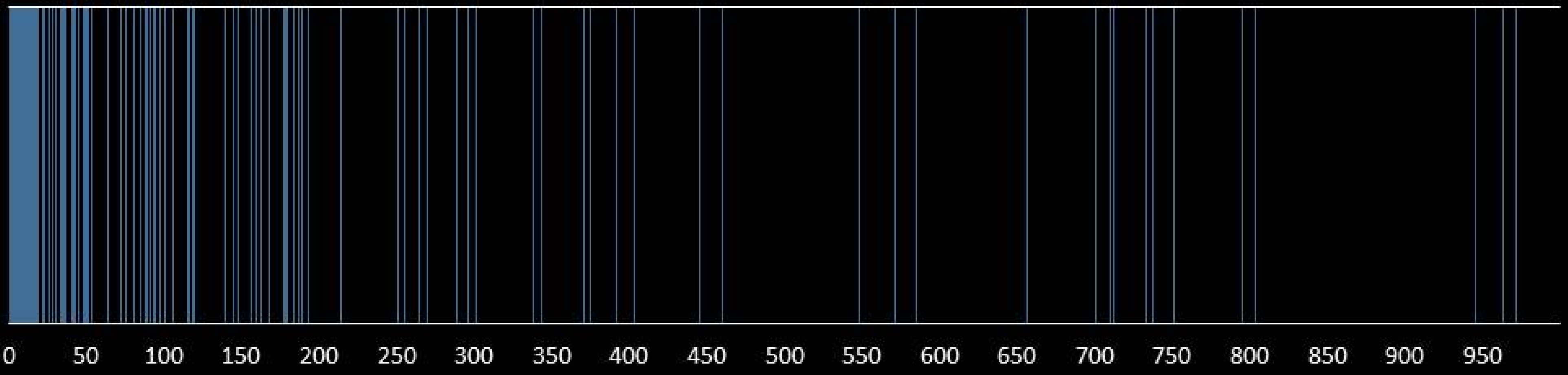


Automatic Feature Learning

- The distribution of TOP 25 (A) & 100 (B) important features
- A



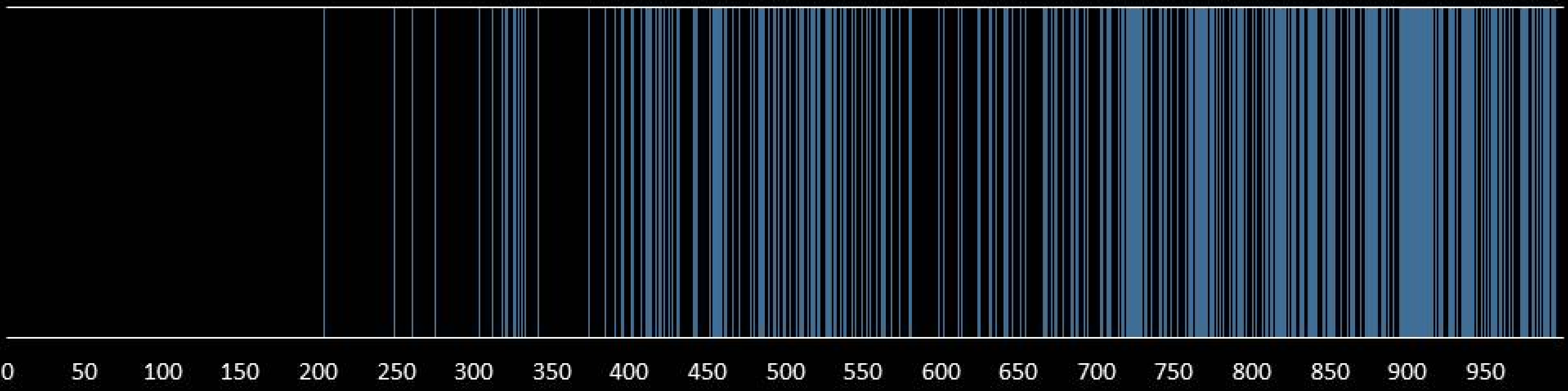
- B



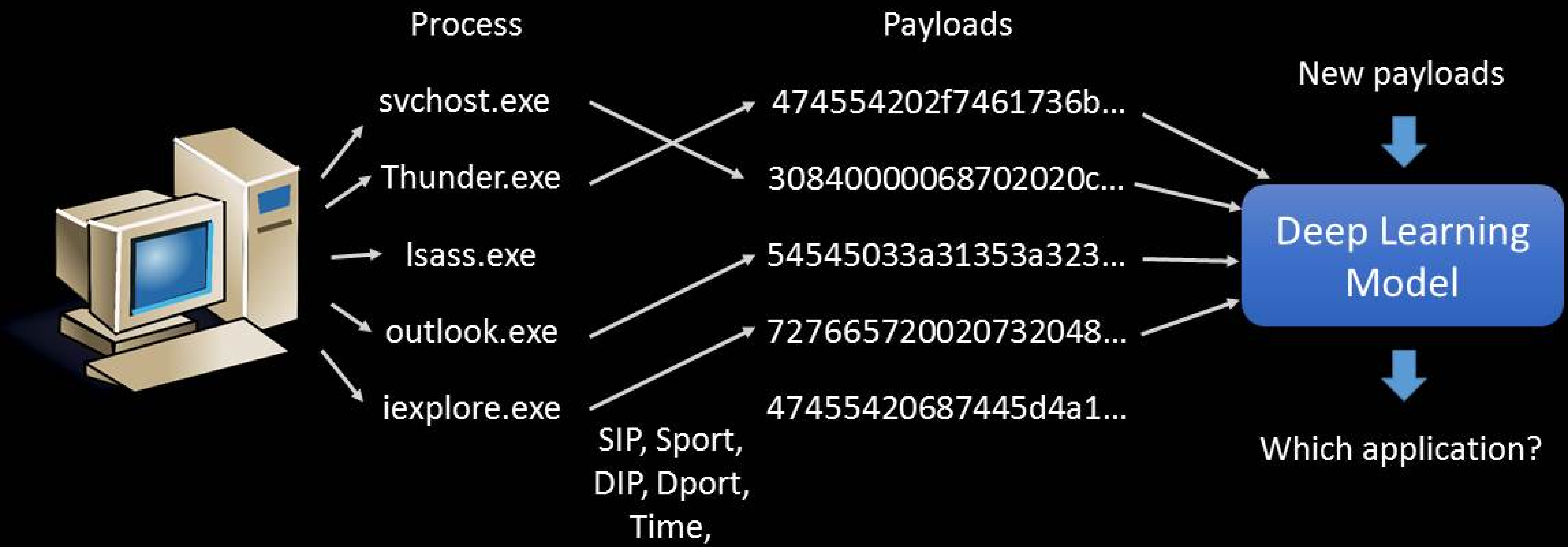
Automatic Feature Learning

- The distribution of 300 least important features

- C



Application Identification



Application Identification

- More than 800 applications in our training data
- Precision: 96.3% (testing on applications that are more than 200 records)

Application	Precision	Protocol	Precision
foxmail.exe	1.0000	xshell.exe	0.9813
wpservice.exe	1.0000	baidumusic.exe	0.9808
taobaoprotect.exe	0.9984	fetion.exe	0.9779
wechat.exe	0.9983	qqmusic.exe	0.9730
liebao.exe	0.9978	qqdownload.exe	0.9615
weibo2015.exe	0.9974	yodaodict.exe	0.9542
lsass.exe	0.9945	itunes.exe	0.9429
sogoucloud.exe	0.9897	outlook.exe	0.9219
qq.exe	0.9884	thunder.exe	0.9168
pplive.exe	0.9870	iexplore.exe	0.8860

Unknown Protocol Identification

- Randomly choose 10,000 records that labeled “unknown” by traditional ways
- our method can also

find out 6,337 of them

	number	ratio
SSL	1956	29.12%
DCE_RPC	1454	21.65%
Skype	873	13.00%
Kerberos	517	7.70%
MSN	360	5.36%
Google	311	4.63%
DNS	260	3.87%
RTMP	234	3.48%
TDS	202	3.01%
H323	170	2.53%

Conclusions and Future Work

- The Applications of Deep Learning on Traffic Identification
  - Protocol Classification
  - Automatic Feature Learning
  - Application Identification
  - Unknown Protocol Identification
- Future Work
  - Applying Convolutional Neural Networks (CNN) model
  - Analysis of encrypted traffics

Thanks!

Zhanyi Wang
wangzhanyi@360.cn