Data-Driven Management
and
An Application to Network Intrusion Detection

December 2020

Kohei Shiomoto

Tokyo City University

Agenda

- **Data-Driven Management**
- **Use-case: Network Intrusion Detection System**

Issues in Network Management

- **Diversified services: Modern Web**
- **Non best effort services: Availability & Quality**
- **Complex system: devices, subsystem, OS, middleware, software**
- **Interaction of players: Customer, ISPs, CDN**
- **Encrypted traffic**

How to deal with complex system?

- **Model-driven approach**
  - **Understand detailed mechanisms of components**
  - **Build up a model of entire system**
- **Data-driven approach**
  - **Obtain data**
  - **Infer the relationship between inputs and outputs**
  - **Machine learning is a key enabler**

Data

- Traffic load
- Performance
- Syslog
- Trouble tickets
- SNS messages (e.g. Twitter)

Data-Driven Management

- **Data-driven approach: Mining data of inputs and outputs of Black-box**
- **Expectations**
  - **Correlation and Causality Inference**
  - **Anomaly Detection**
  - **Root Cause Analysis**
  - **Traffic Prediction**
  - **Knowledge Discovery**
  - **…**

Data-Driven Management anomaly detection

- **Correlation detection: NICE [1], G-RCA[2]**
- **Syslog analytics: SyslogDigest [3], Spatio-Temporal [4]**

[1] Ajay Mahimkar, et al., "Troubleshooting chronic conditions in large ip networks," CoNEXT '08.
[2] H. Yan, et al., "G-rca: A generic root cause analysis platform for service quality management in large ip networks," CoNEXT '10.
[3] T. Qiu, et al., "What happened in my network: Mining network events from router syslogs," IMC '10.
[4] T. Kimura, et al., "Spatio-temporal factorization of log data for understanding network events," IEEE INFOCOM 2014.

Data-Driven Management
Root cause analysis

- **IP over Fiber: SCORE [5], Shrink [6]**
- **Enterprise network: Sherlock [7]**
- **CDN: WISE [8]**

[5] R.R. Kompella, et al., “Ip fault localization via risk modeling,” NSDI’05.
[6] S. Kandula, et al., “Shrink: A tool for failure diagnosis in ip networks,” MineNet ’05.
[7] P. Bahl, et al., “Towards highly reliable enterprise network services via inference of multi-level dependencies,” SIGCOMM ’07.
[8] M. Tariq, et al., “Answering what-if deployment and configuration questions with wise,” SIGCOMM ’08.

Data-Driven Management Knowledge Discovery

- **Trouble ticket analysis [9]**
- **Router config error detection: Mineral [10]**
- **Mobile network eNodeB: AESOP [11]**

[9] A. Watanabe, et al., "Workflow extraction for service operation using multiple unstructured trouble tickets,". NOMS 2016.
[10] F. Le, et al., "Minerals: Using data mining to detect router misconfigurations," MineNet '06.
[11] S. Deb, et al., "Aesop: Automatic policy learning for predicting and mitigating network service impairments," KDD '17.

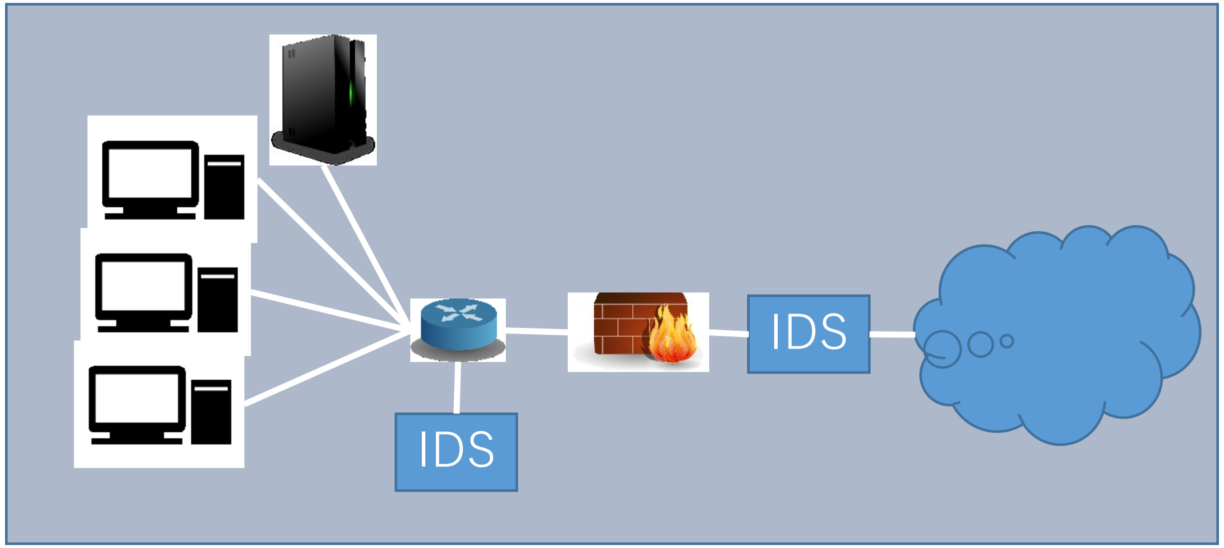
Intrusion Detection System using Semi-Supervised Learning with Adversarial Auto-encoder

[12] K. Hara and K. Shiomoto, "Intrusion Detection System using Semi-Supervised Learning with Adversarial Auto-encoder," ***NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*****, Budapest, Hungary, 2020, pp. 1-8, doi: 10.1109/NOMS47738.2020.9110343.**

Introduction: IDS

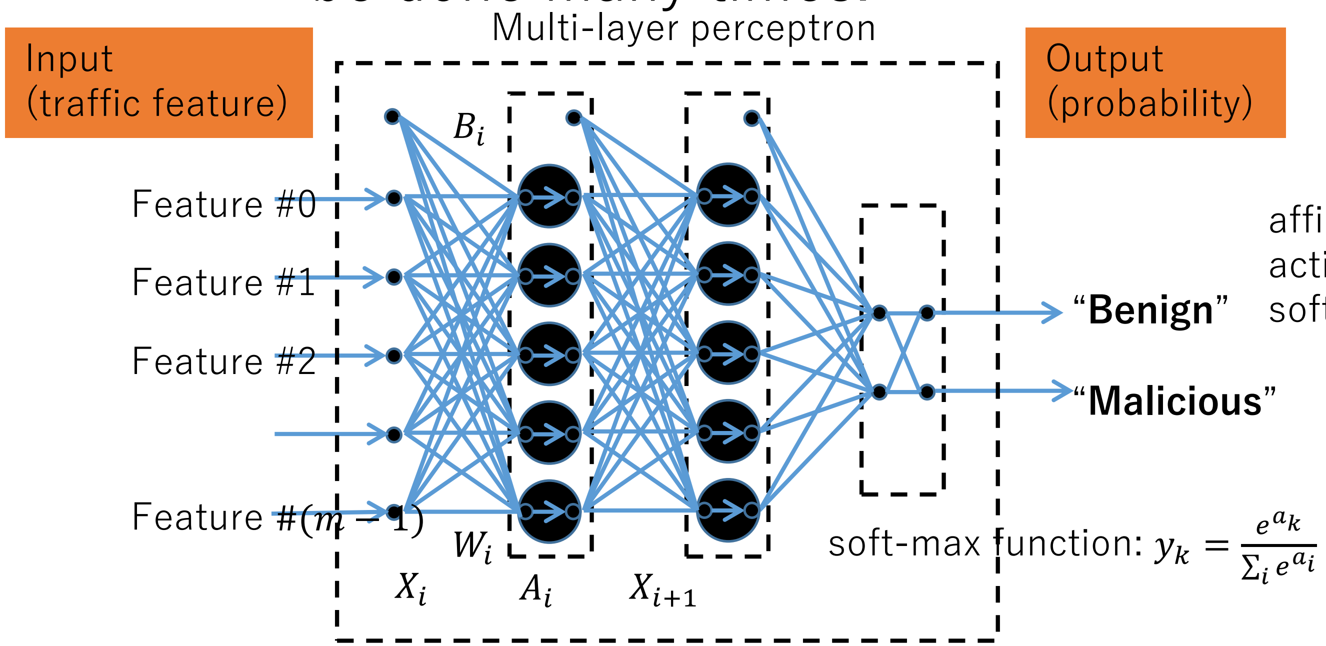
- An IDS is a detection system put in place to monitor computer networks.
- IDS monitors activities of computer and network systems and classifies them as either normal or anomalous.
- By analyzing patterns of data, IDS helps to detect threats that can be devastating.

Task of IDS:
IDS examines traffic **feature** to classify the traffic: “**benign**” or “**malicious**”.



Introduction: Machine Learning

- Supervised machine learning methods need to be trained with a large number of training data annotated with the correct labels.
  - Costly task; Human operator examines data, classifies them, and annotates them with an appropriate label.
  - Trends in network traffic change from day to day; labeling work needs to be done many times.



affine function: $A_i = X_i W_i + B_i$
activation function: $X_{i+1} = H_i(A_i)$
soft-max with loss function:

$$L = -\sum_k t_k \log y_k, \text{ where } y_k = \frac{e^{a_k}}{\sum_i e^{a_i}}$$

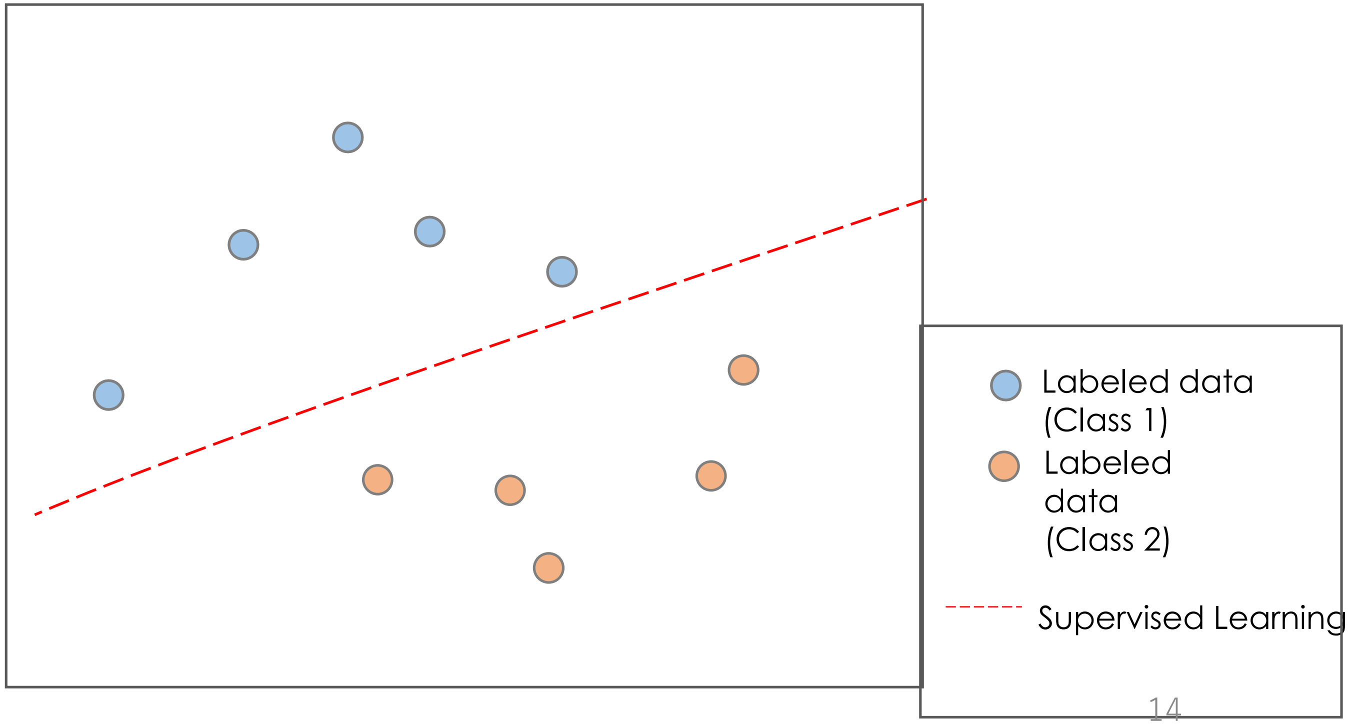
and t_k is a training data (one-hot)

Introduction: Contribution

- We propose an IDS that employs semi-supervised learning based on **Adversarial Auto-encoder (AAE)**.
- **Semi-supervised learning** uses **a small number of labeled data** in training dataset to reduce costly human-labor tasks and improves the performance with support of unlabeled data in training dataset.
- Our approach realizes a detection rate of 82.78% using **only 1.0% of labeled data** compared to other state-of-the-art approaches.

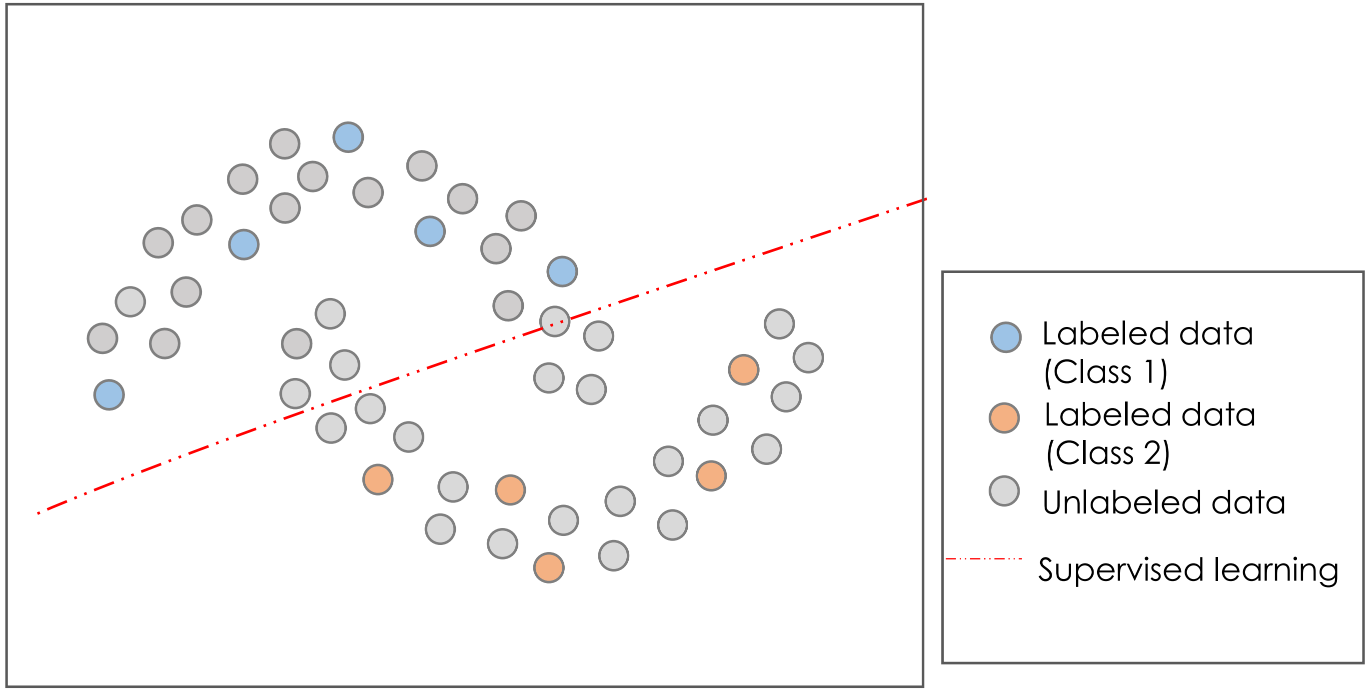
Supervised Learning & Semi-Supervised Learning

- Supervised learning (SL)
  - Only labeled data used
- We may draw a line between two classes



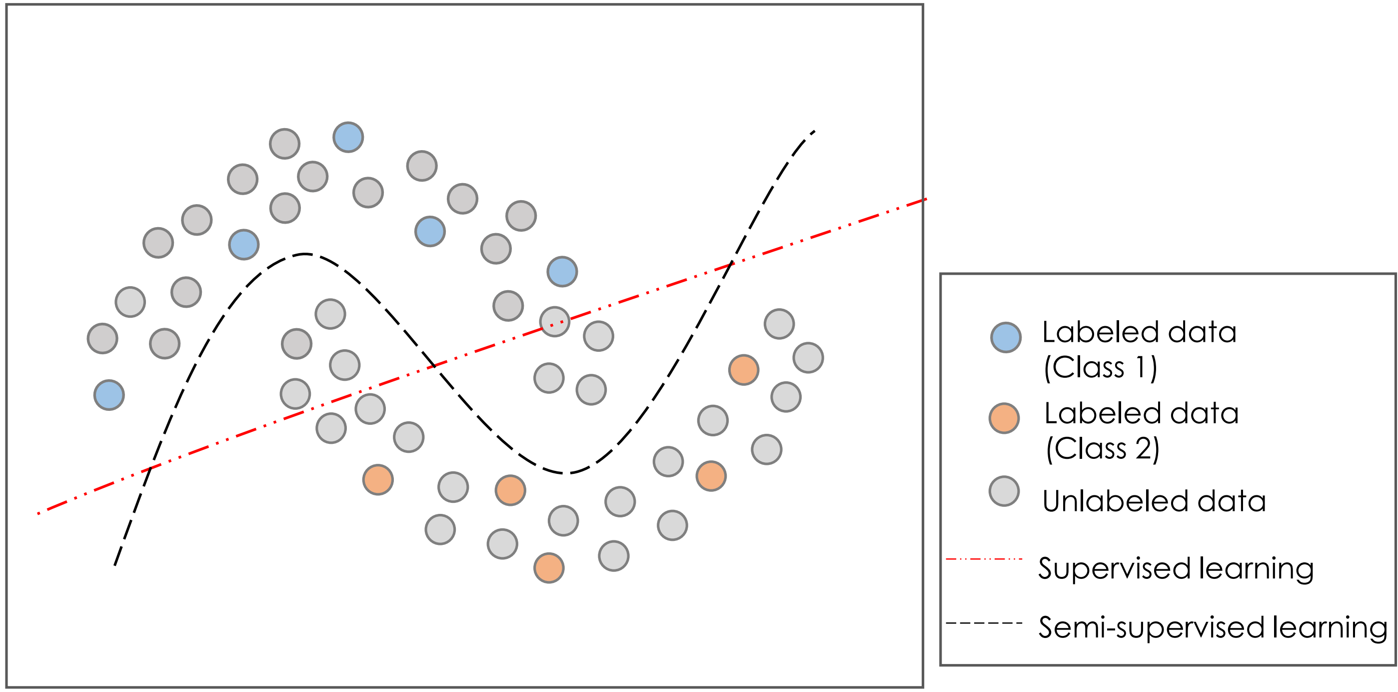
Supervised Learning & Semi-Supervised Learning

- What if we have unlabeled distributed like this?



Supervised Learning & Semi-Supervised Learning

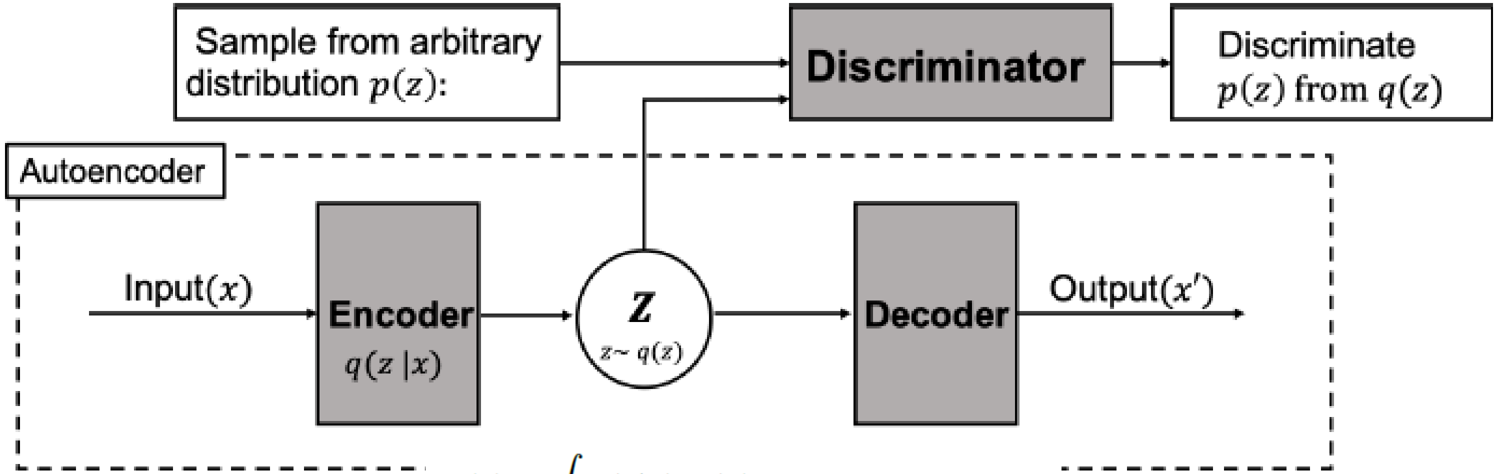
- Semi-supervised learning (SSL)
  - Train classifier using ***small*** labeled data in support with unlabeled data
- We may draw another line between two classes



[13] Avital Oliver, Augustus Odena, Colin Raffel, Ekin D. Cubuk, and Ian J. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. CoRR, abs/1804.09170, 2018.

Machine Learning: Adversarial Auto-Encoder (AAE)

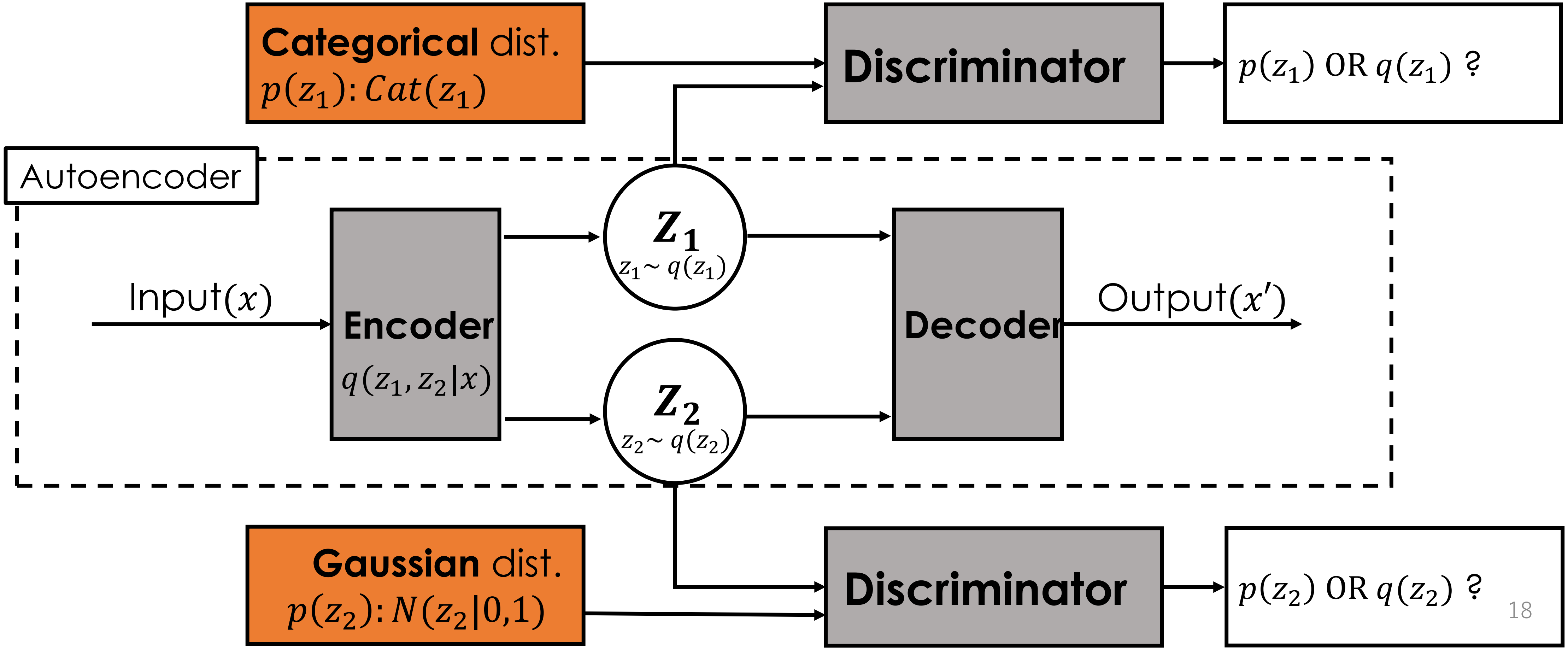
- Adversarial Auto-Encoder (AAE) employs AE and GAN as a key building block.
  - The AE reduces the dimension of input data by extracting and maintaining important features as the latent variable vector *z*.
  - The GAN employs the generator and the discriminator in such a way that the latent variable vector *z* of the AE follows an arbitrary distribution for regularization.



$$q(z) = \int_x q(z|x) p_d(x) dx \qquad (4)$$

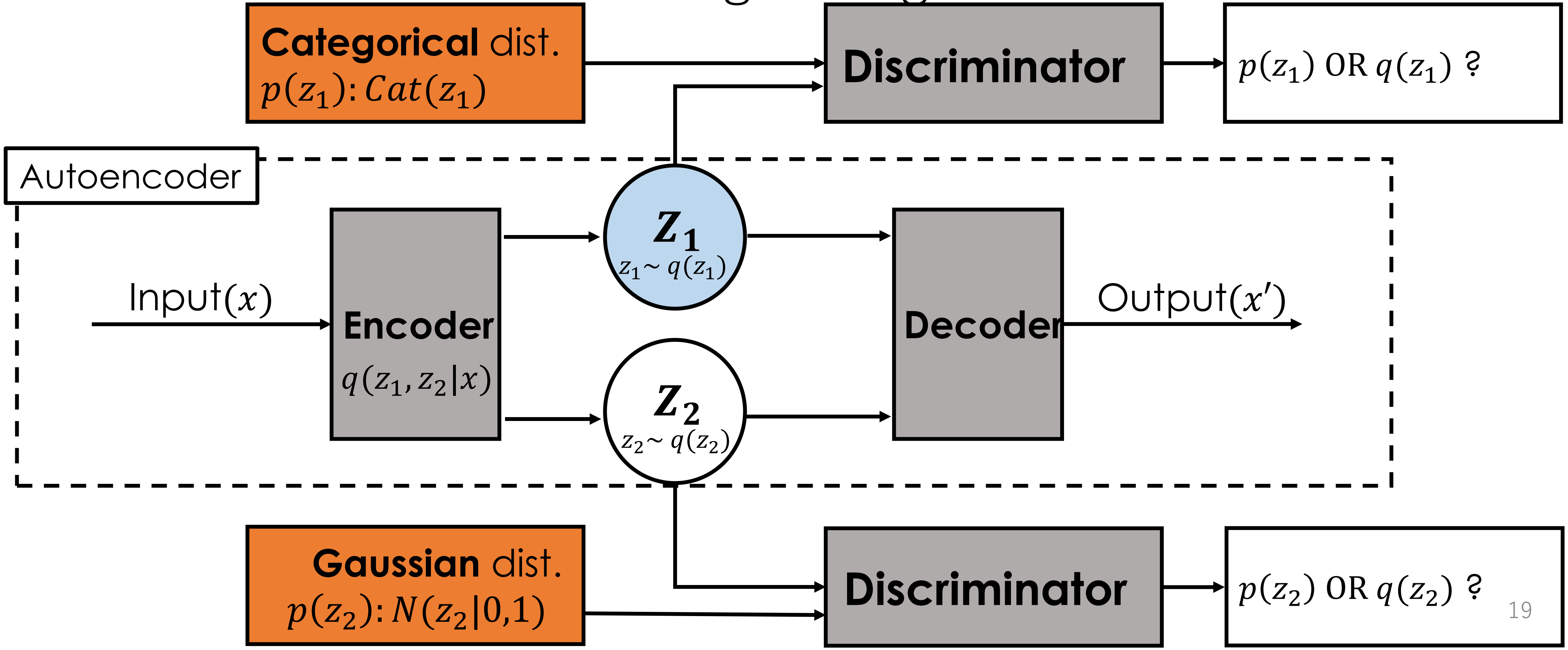
Proposed Method: Proposed AAE model

- The AAE is used as semi-supervised learning.
  - The latent variable vector ***z*1** holds the features representing the class information ("normal" or "attack")
  - The latent variable vector ***z*2** holds the other features.



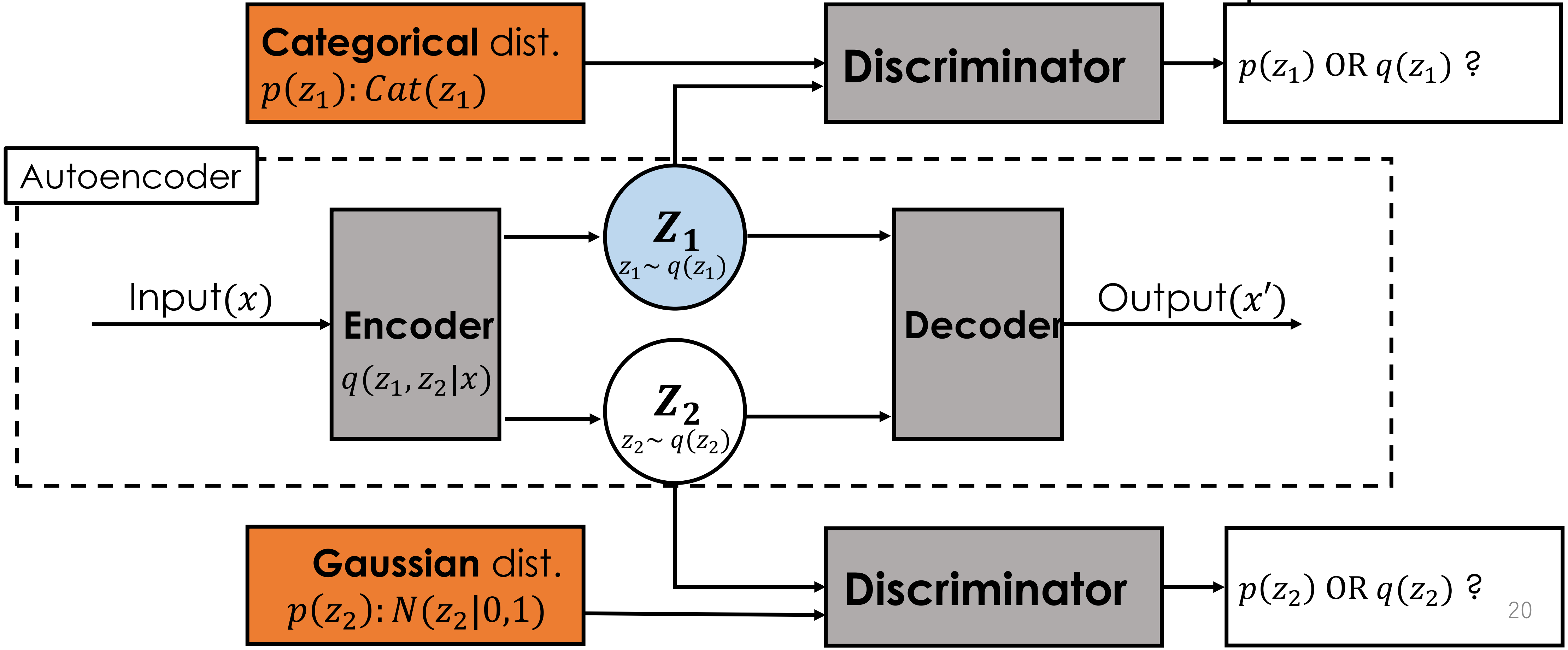
Proposed Method: Training and Inference

- We train the AAE using **unlabeled data**.
- When a **labeled data** is available, we train the AAE by using the **label** instead of the categorical generative model.



Proposed Method: Training and Inference

- Once the AAE is trained, it is used to classify new coming data; the **latent variable *z1*** in the middle hidden layer indicates the inferred class associated with the input data.



Evaluation: Dataset

- We have used NSL-KDD dataset that is widely used in performance evaluation other IDS methods.
- This dataset consists of records of traffic sent and received between the source and destination IP address, and is divided into **KDDTrain+** (125,973 data records) for train and **KDDTest+** (22,544 data records) for evaluation.
- Each traffic sample has **41 features** that are categorized into three types of features: basic features, content-based features, traffic-based features.
- Among 41 features, some of them are **categorical** such as protocol type that takes three values (tcp, udp icmp), flag that takes 11 values (SF, S1, REJ, etc.), and service that takes 70 values (http, telnet, ftp, etc.). Instead of coding each categorical data into a scalar value, we adopt an **one-hot vector** representation, resulting in **122 features**.

NSL-KDD Dataset

Category	Training Set	Testing Set
DoS	back, land, neptune, pod, smurf, teardrop	back, land, neptune, pod, smurf, teardrop, **mailbomb**, **processtable**, **udpstorm**, **apache2**, **worm**
R2L	fpt-write, guess-passwd, imap, multihop, phf, spy, warezclient, warezmaster	fpt-write, guess-passwd, imap, multihop, phf, spy, warezmaster, **xlock**, **xsnoop**, **snmpguess**, **snmpgetattack**, **httptunnel**, **sendmail**, **named**
U2R	buffer-overflow, loadmodule, perl, rootkit	buffer-overflow, loadmodule, perl, rootkit, **sqlattack**, **xterm**, **ps**
Probe	ipsweep, nmap, portsweep, satan	ipsweep, nmap, portsweep, satan, **mscan**, **saint**

Training and Testing sets include 125,973 and 22,544 records.
Some specific attack types in the testing set do not appear in the training set. That makes the detection task more realistic.

DOS: denial-of-service, e.g. syn flood;
R2L: unauthorized access from a remote machine, e.g. guessing password;
U2R: unauthorized access to local superuser (root) privileges, e.g., various ``buffer overflow'' attacks;
probing: surveillance and other probing, e.g., port scanning.

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A detailed analysis of the kdd cup 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6, July 2009.

Comparison with Conventional Methods

- Proposed method yields comparable accuracy using small labeled data

Method	Accuracy	Labeled	Unlabeled
XGboost with K-Means [17]	84.25	125,973	0
Bagging(Base classifier -J48) [11]	84.25	125,973	0
RNN [19]	83.28	125,973	0
AAE (10%labeled)	**83.11**	**12,597**	**113,376**
AAE (1%labeled)	**82.78**	**1,259**	**124,714**
Support Vector Machine [10]	82.37	125,973	0
NBTree [16]	82.02	125,973	0
Random Tree [16]	81.59	125,973	0
J48 [16]	81.05	125,973	0
Random Forest [16]	80.68	125,973	0
Multilayer Perceptron [16]	77.41	125,973	0

Accuracy

Labeling Rate	90%	10%	1%	0.10%	0.01%
Labeled Data	113,375	12,597	1,259	125	11
Unlabeled Data	12,598	113,376	124,714	125,848	125,962
Adversarial Autoencoder	83.20	83.11	82.78	81.37	53.66
Deep neural network	81.07	77.33	75.88	76.87	71.55

Misdetection

- FPR
  - Proposed AAE (13.5%) < Conventional DNN (7.8%)
- FNR
  - Proposed AAE (20.0%) > Conventional DNN (36.5%)

AAE	DNN	Predicted Class			
		Normal		Anomaly	
Actual Class	Normal	86.5%	92.2%	13.5%	7.8%
	Anomaly	20.0%	36.5%	80.0%	63.5%

The experiment was performed using 1.0% labeled data with AAE.

Summary

- **Data-driven Approach**
- **Intrusion Detection System**

Closing Remarks
Final Remarks --We need talent