Virtual Machine Failure Prediction using Log Analysis

MS Thesis Defense

Sukhyun Nam
Supervisor: Prof. James Won-Ki Hong
DPNM Lab, CSE, POSTECH, Korea
obiwan96@postech.ac.kr

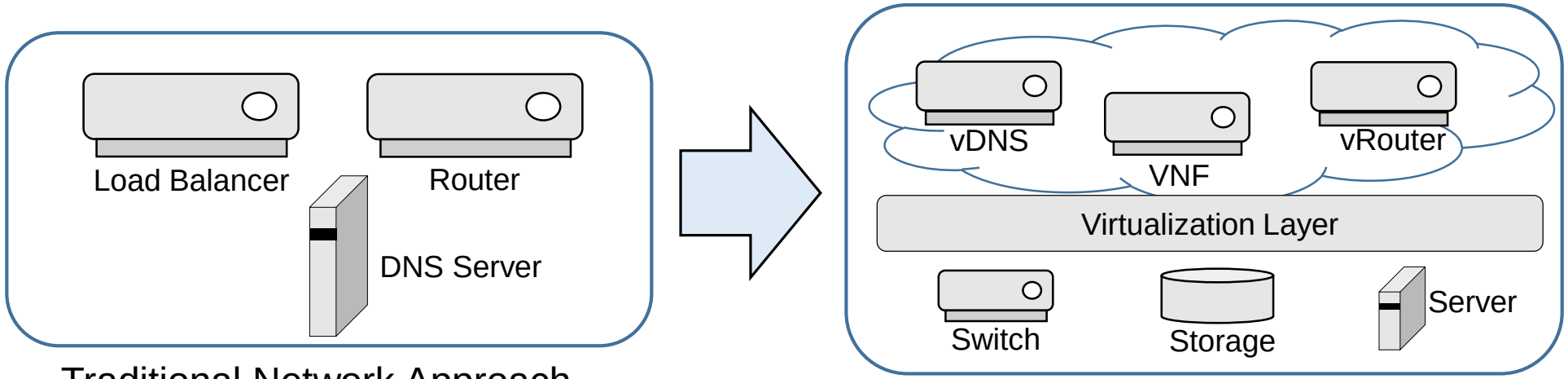
Jun. 22, 2021

Table of Contents

Introduction

Introduction

- Network Function Virtualization (NFV)
  - NFV technologies decouple functions (e.g. firewall, load balancer etc.) from hardware and move them to virtual servers
  - NFV reduced OPEX and CAPEX
  - Made difficult to monitor and take action on virtual machines (VMs) and server failures
    - Faults in clouds system can take hours and days to fix [1]



Traditional Network Approach

NFV Approach

Introduction

- VM Failure Prediction Tasks
  - Predict the failures in VM in advance, to use in live migration of VNF before the failures occur to minimize service quality degradation
  - Some of the failures have early errors or faults associated with
    - Errors or faults of computer equipment can be found in the log

- Challenges of Failure Prediction Tasks
  - Complicated failure causes
  - Complex failure-indicating signals
  - Highly imbalanced data

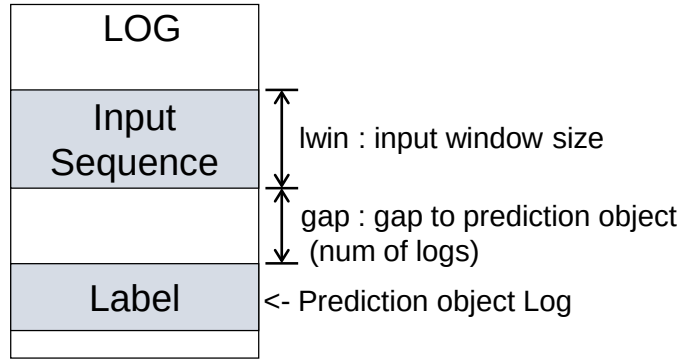
Introduction

- Research Goal
  - Predict at least one minute before a failure occurs using the logs that the VM outputs
    - Failure is a state that the VM fails to network function
    - Live migration takes an average of 45 seconds before based on VMs with size of 5GB on OpenStack

Background & Related work

Background

- Network Failure Prediction
  - W. Ji et al (CCDC 2018) [2]
    - Predict whether logs contain failure messages in wireless communication systems
    - CNN showed best performance when experimenting with GRU, LSTM, CNN
    - Accuracy 0.75 with gap 2000, accuracy 0.57 with gap 5000
  - MING (ESEC/FSE 2018) [3]
    - Predict node failure before 6 hours in cloud service
    - Use temporal features (e.g. performance counters, resource usage) and spatial features (e.g. rack location, load balance group, update domain)
    - Average recall of 0.63, precision of 0.92 and F1 score of 0.75
- Log based Anomaly Detection
  - Deeplog (SIGSAC 2017) [4]
    - Use deep neural network (DNN) to learn log patterns from normal execution
    - Show F1 score of 0.98 in the OpenStack data set

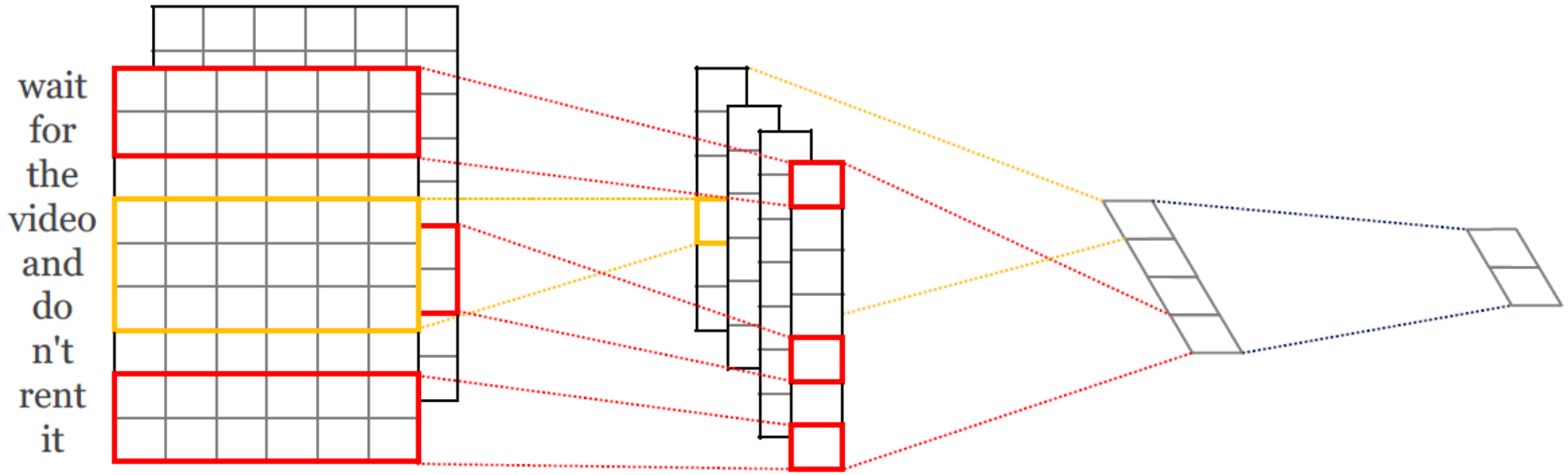


Background

- Fault and Failure [5, 6]
  - Mandelbug is a kind of bug whose activation and propagation are complex
    - Hard to reproduce
    - Takes longer time to fix than regular bugs
  - In Linux, many failures related to networking are caused by Mandelbug [6]
  - Two types of Mandelbug generate early symptoms
    - ARB (Aging Related Bug)
      - A kind of bugs that can cause an increasing failure rate and/or degraded performance, known as software aging
      - Symptoms : errors or faults due to overload (memory leaks or increase in total system runtime)
    - LAG
      - A kind of bugs that are non-aging related Mandelbug (NAM), but there exist a time **lag** between the activation of the bug and the occurrence of its failure
      - Symptoms : variety

Background

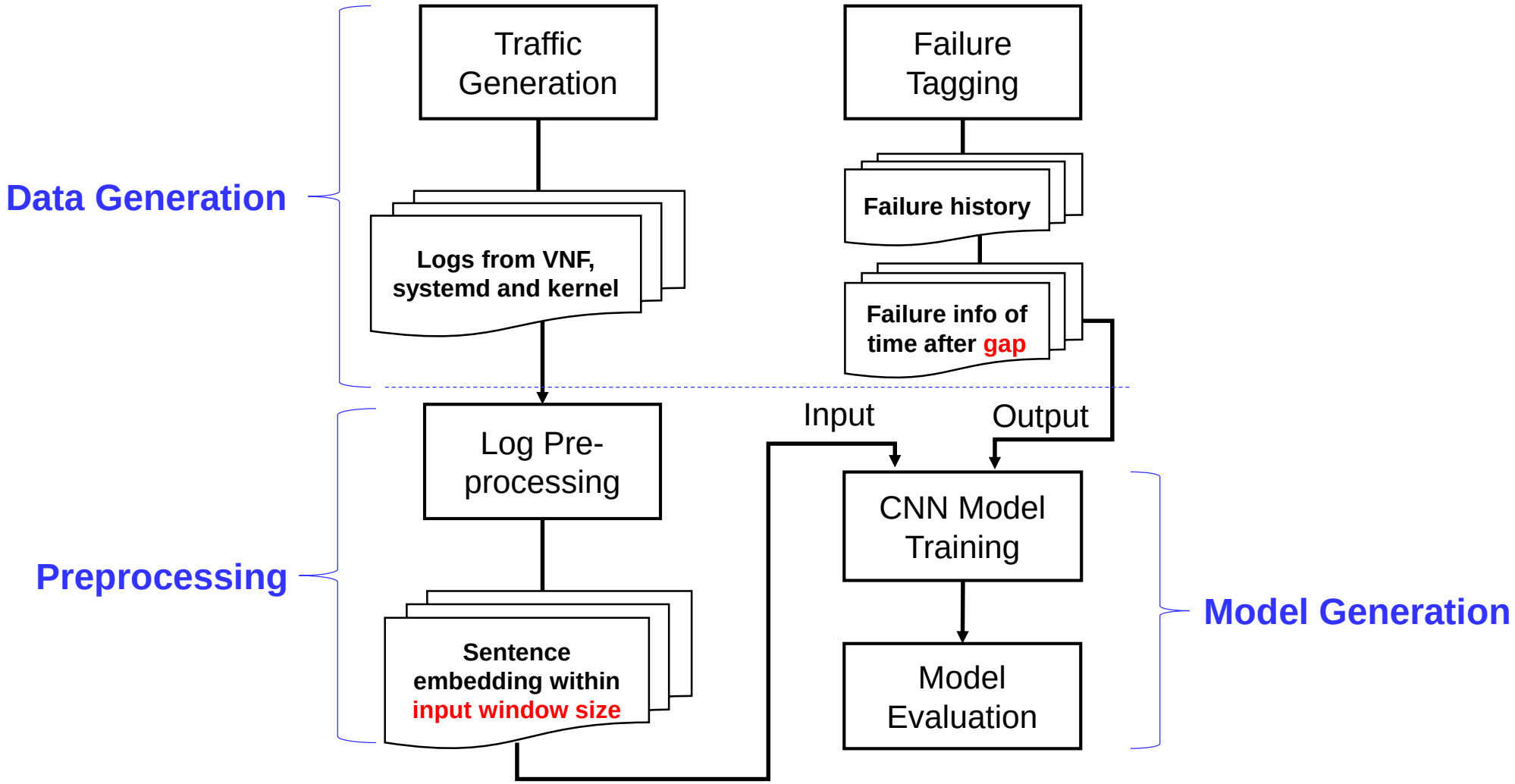
- CNN (Convolutional Neural Network)
  - Artificial neural networks specialized for learning that extract features without losing information from large amounts of data
  - Typically contains multiple convolution layers and pooling layers
  - The operation is simple and the number of parameters is small
  - CNN performs best in studies on sentence classification problems [7]



Methodology & Implementation

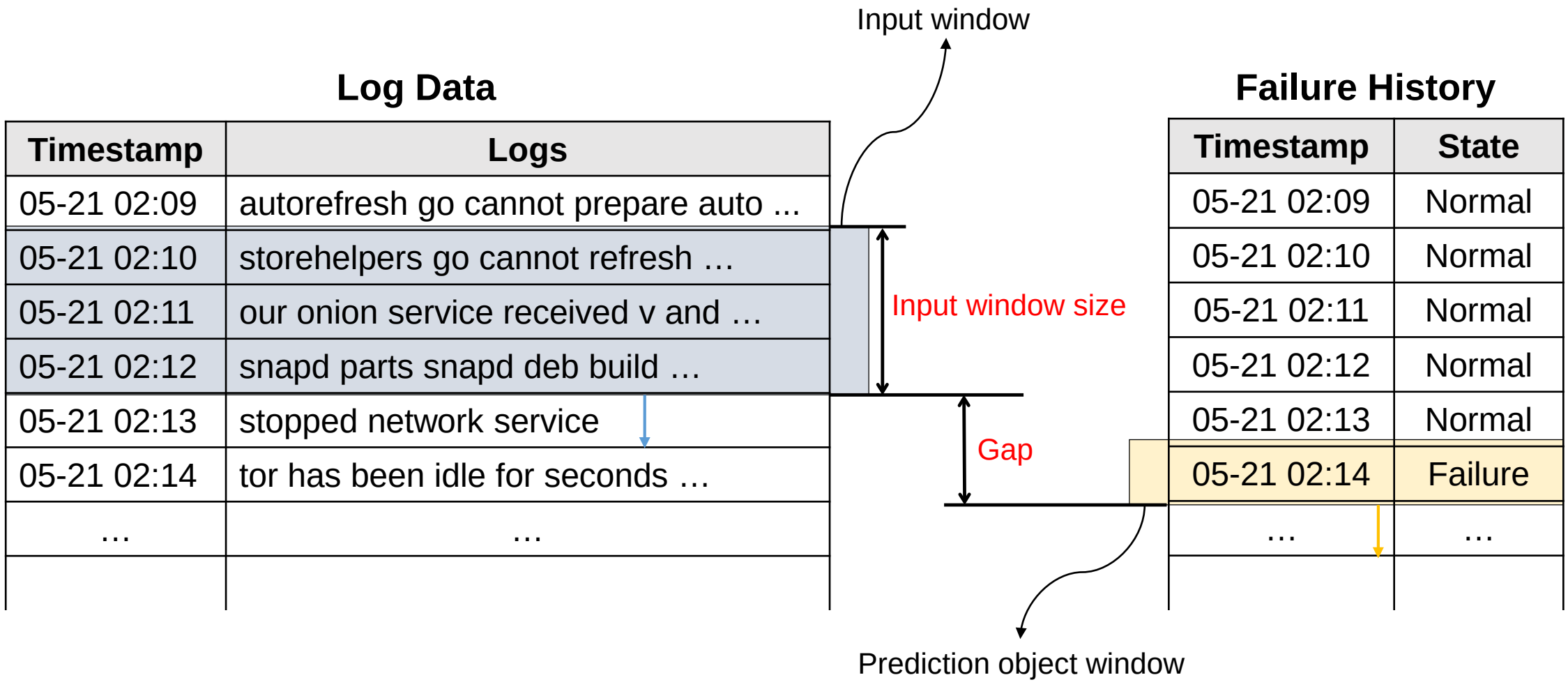
Methodology

- Overview of VM failure prediction model development



Methodology

- Two sliding windows for input and output



Input

- Pre-processing
  - Remove numbers and replace symbols with space
  - Translate time info as timestamps and remove VM name, application name
  - Delete duplicate log

<Time> <VM name> <Application name>

Raw Logs

May 21 02:10:13 225-2c-4 snapd[2463]: storehelpers.go:551: cannot refresh: snap has no updates available: "core18", "lxd", "snapd"
May 21 02:10:14 225-2c-4 snapd[2463]: autorefresh.go:479: auto-refresh: all snaps are up-to-date
.
.

Pre-processing

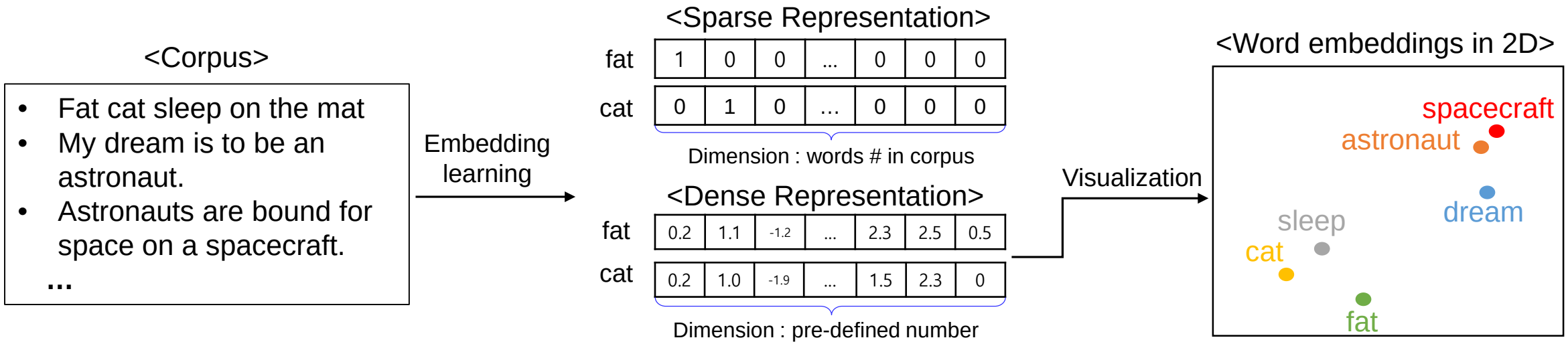
<Log table for VM 225-2c-4>

Timestamps	Logs
05-21 02:10	storehelpers go cannot refresh snap has no updates available core lxd snapd
05-21 02:10	autorefresh go auto refresh all snaps are up to date
.	.
.	.

Input

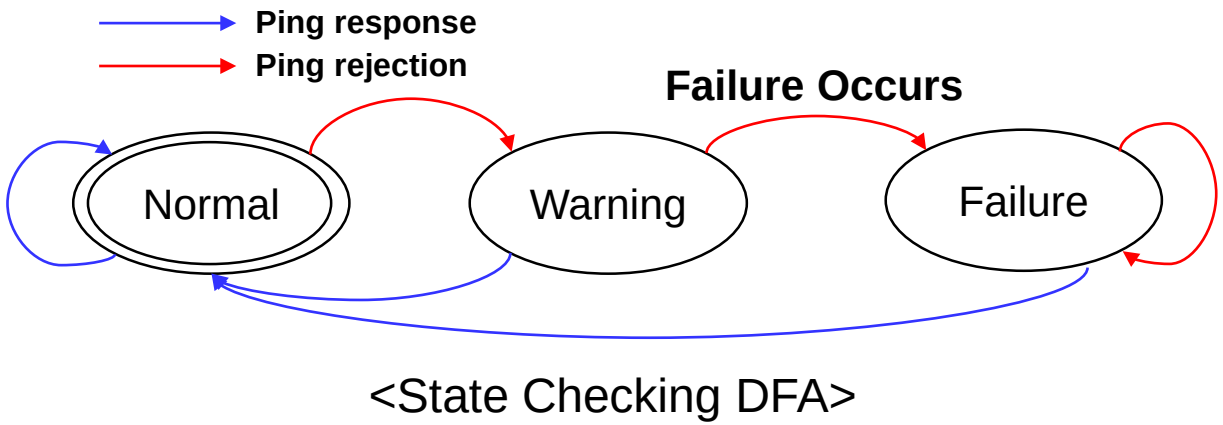
- Word Embedding
  - Express a word as a dense vector with preserving the characteristics of the word
  - Public word embedding is not appropriate for log analysis

- Embedding Vectors for Log Corpus
  - Generated with Google's open-source project word2vec [8]
  - Contains 265,452 words
  - ex) most similar words with 'err'
    - errors, over, dropped, rx, crc, tx, collisons, miss



Output –Failure history

- State checker send periodic ping to each VM and tag state based on DFA
  - Tag as failure if VM reject ping for a minute
- Save failure history for each VM



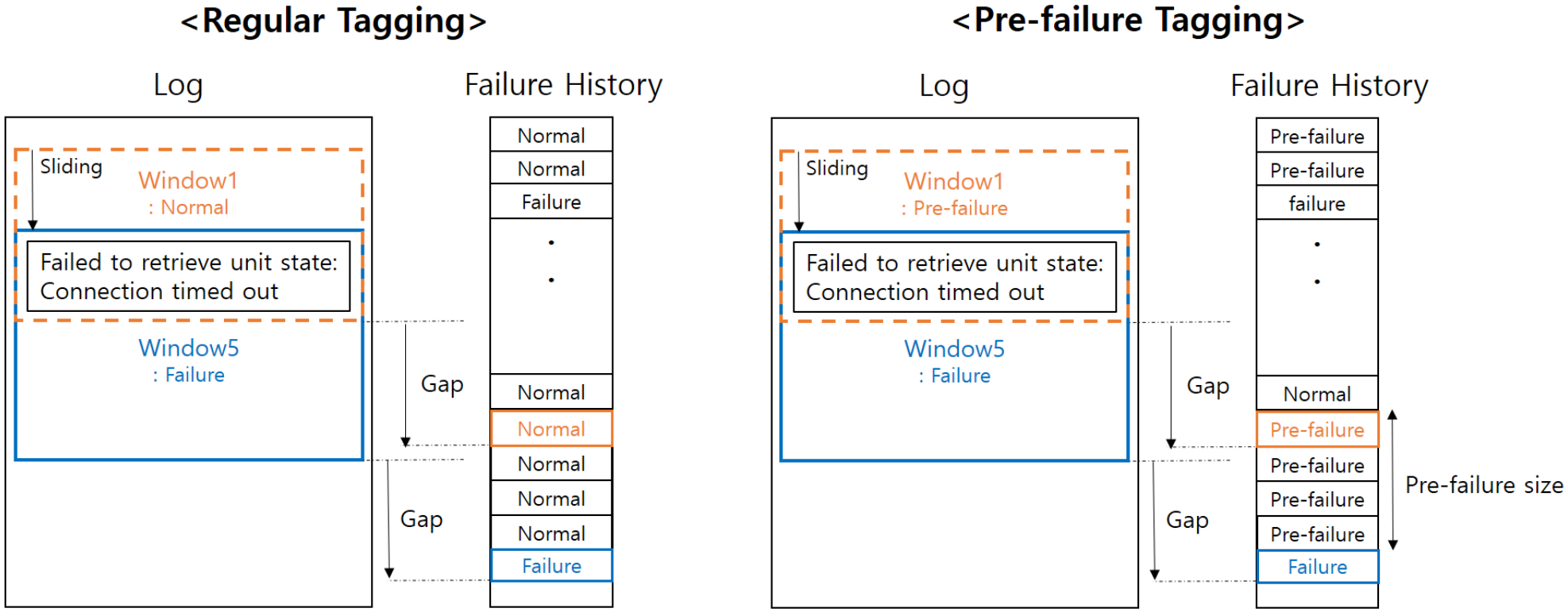
<State Checking DFA>

<Failure History>

VM1	VM2	VM6
Normal	Normal	Normal
Normal	Failure	Normal
Failure	Failure	Normal
Failure	Normal	Normal
Failure	Normal	Normal
.	.	.
.	.	.
Normal	Normal	Normal
Normal	Normal	Normal
Failure	Normal	Normal

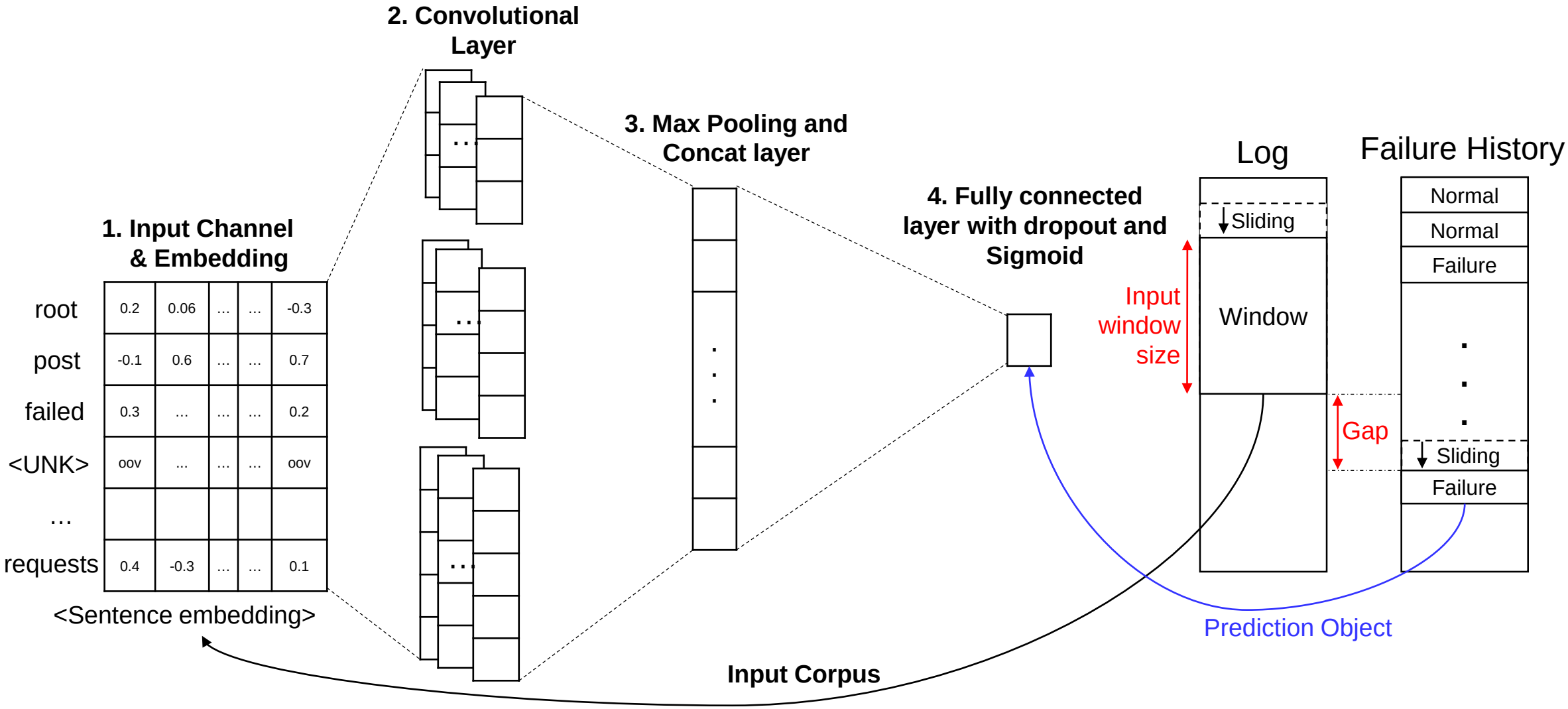
Output - Pre-failure tagging

- CNNs are trained to extract features regardless of order
- Tags the states before the failure occurred to pre-failure rather than normal
- Tags pre-failure value during pre-failure size
- To enable pre-failure to be applied to loss, use KL Divergence-based custom loss



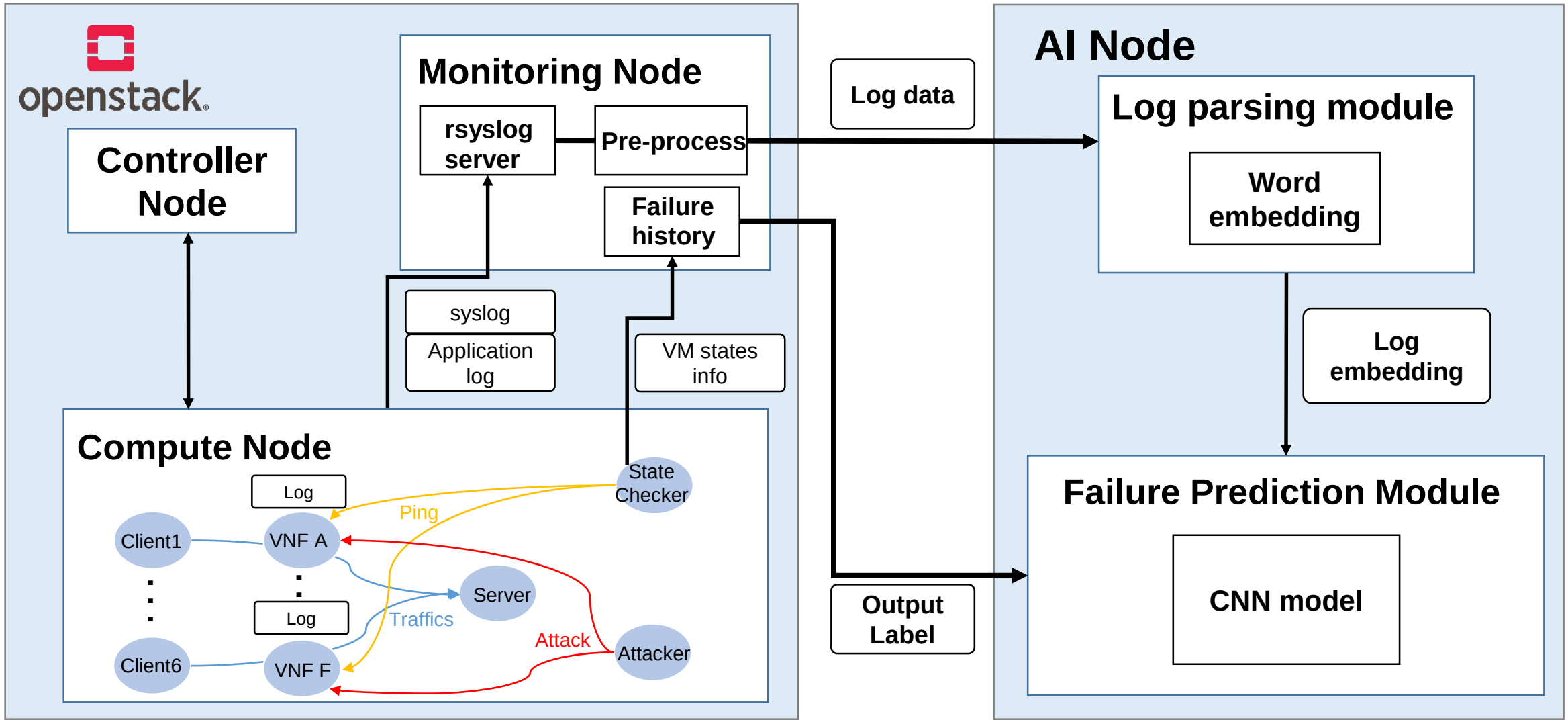
Learning Algorithm

- CNN



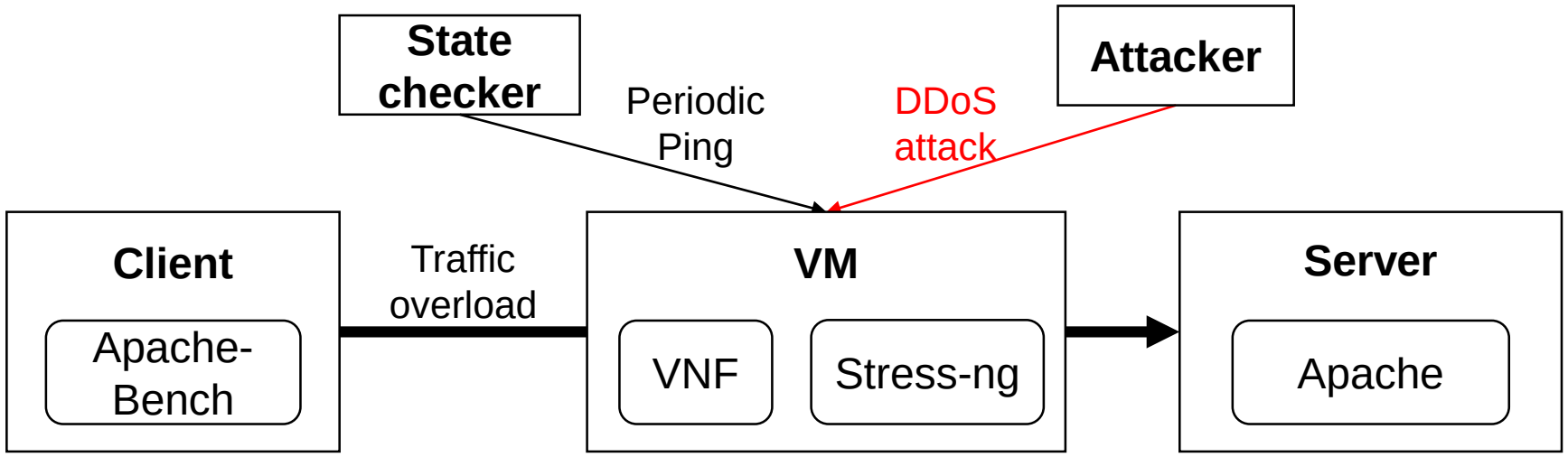
Experiment & Evaluation

Experimental setup



Experimental setup

- Data Collection
  - Generate multiple client-VNF-server chains
  - Fault inject
    - In Microsoft cloud system, each day less than 0.1% of the nodes encounter failures [7]
    - Generate traffic, resource overload and external attack



Experiments

- Failure Data
  - Collected 44 failures in 1 months
    - Server failures : 13
    - None symptom : 4
    - Failures with error before
      - ARB : 21
      - LAG : 6
      - 5 of them had a gap of more than 30 minutes
    - Total 22 number of failures were used for learning

Type	gap	Application	log
ARB	5	systemd-timesyncd	Timed out waiting for reply from
ARB	2	apt-helper	Failed to retrieve unit state: Connection timed out
LAG	14	kernel	blk_update_request: I/O error, dev vda, sector op READ flags phys_seg prio class
LAG	1	kernel	fail to add MMCONFIG information, can't access extended PCI configuration space under this bridge.

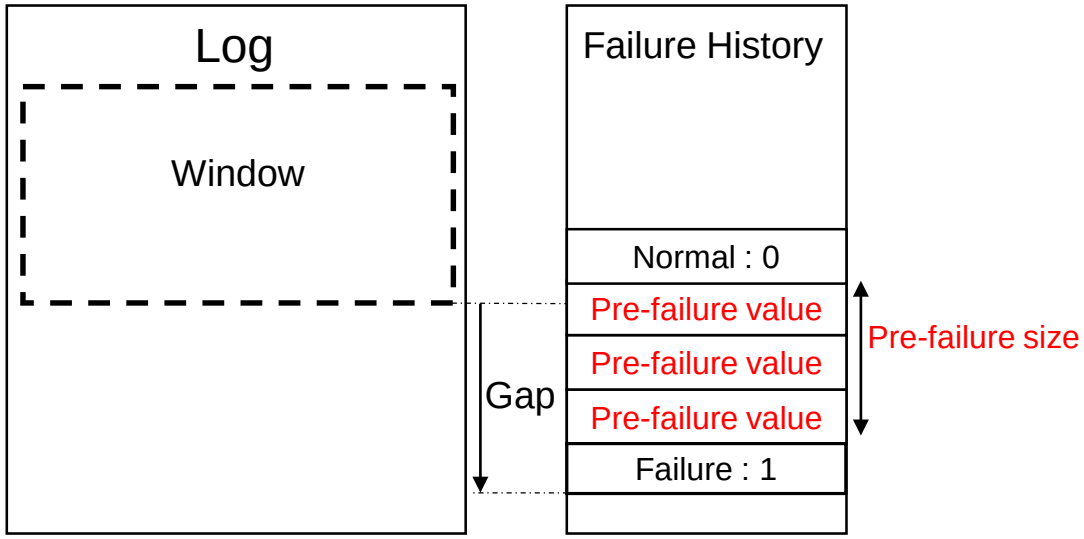
Experiments

- Unbalanced data
  - Total 43,364 number of windows are made, but only 22 of them are tagged as failure
  - Apply oversampling with 2 to failure data and random undersampling with 60 to normal data
  - Apply class weight for loss function as the reciprocal number of each class (normal/failure) in data

- Test
  - Shuffle the data and divide by a ratio of 8:2 as train set and test set
  - Divide by a ratio of 8:2 again the train set as train set and validation set

Evaluation

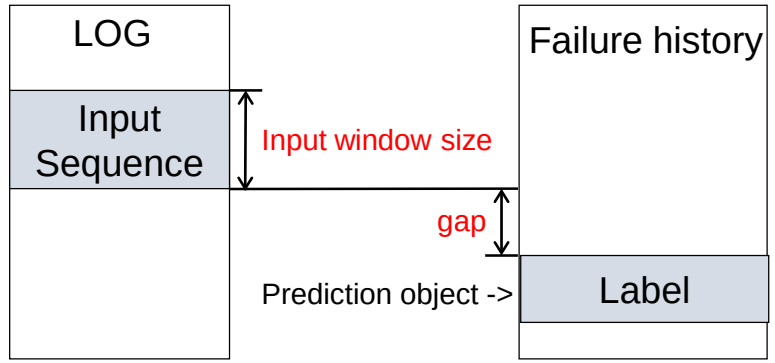
- Pre-failure size and Pre-failure value test
  - Use 5 minutes for gap and input window size
  - Without pre-failure tagging
    - Acc: 0.95, Rec: 0.14, F1: 0.25



Size (min) \ Value	0.5	0.65	0.8
3	Acc: 0.98, Rec: 0.75, F1: 0.60	**Acc: 0.95, Rec: 1.00, F1: 0.67**	Acc: 0.97, Rec: 0.60, F1: 0.55
5	Acc: 0.97, Rec: 0.57, F1: 0.57	Acc: 0.91, Rec: 0.33, F1: 0.29	Acc: 0.96, Rec: 0.33, F1: 0.40
10	Acc: 0.86, Rec: 0.75, F1: 0.44	Acc: 0.90, Rec: 0.40, F1: 0.25	Acc: 0.93, Rec: 0.50, F1: 0.36

Evaluation

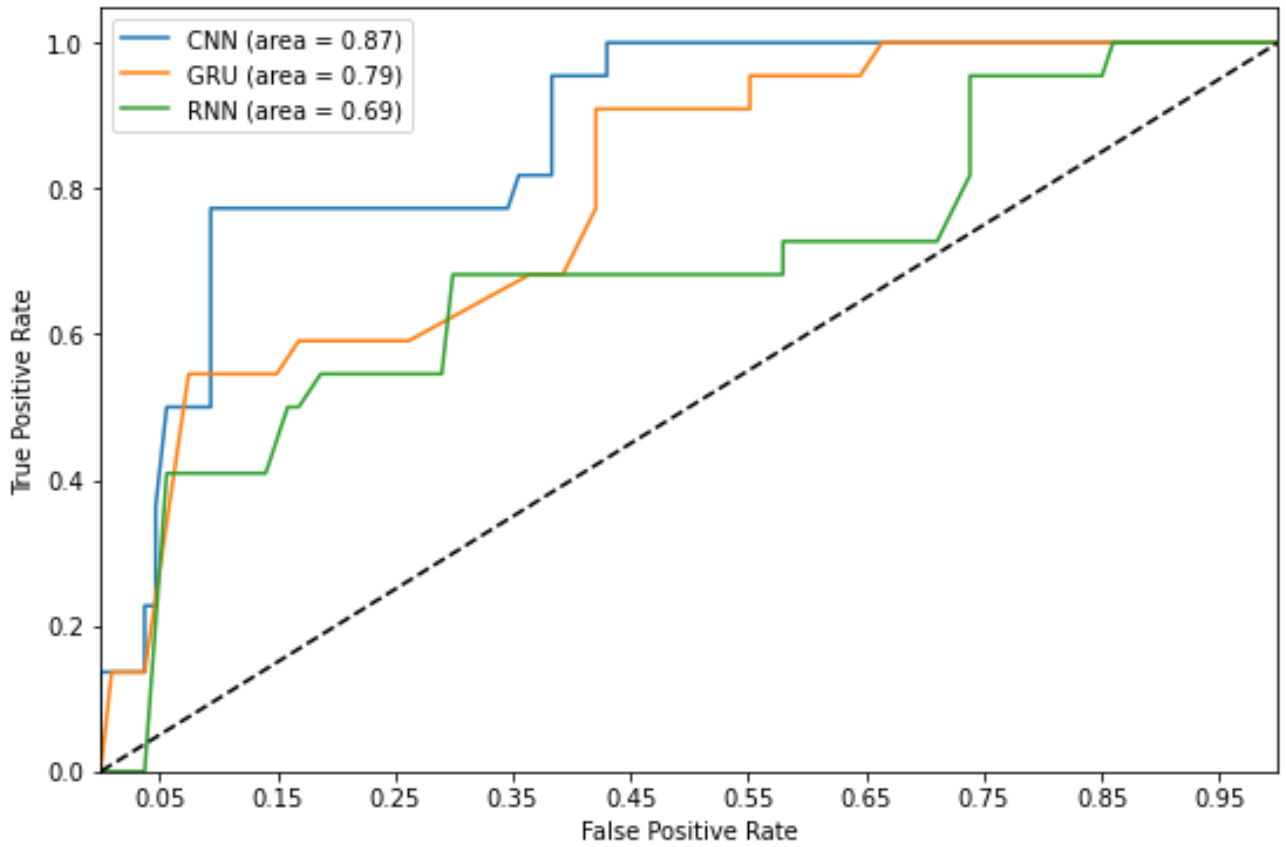
- Gap and input window size test
  - Use 3 minutes for pre-failure size and 0.65 for pre-failure value
  - Predict failures before 5 minutes with 0.67 of F1 score



Win (min) / Gap (min)	5	10	20
1	Acc: 0.93, Rec: 0.45, F1: 0.56	Acc: 0.94, Rec: 0.60, F1: 0.57	Acc: 0.88, Rec: 0.43, F1: 0.22
3	Acc: 0.93, Rec: 0.37, F1: 0.46	Acc: 0.95, Rec: 0.33, F1: 0.43	Acc: 0.94, Rec: 0.57, F1: 0.44
5	**Acc: 0.95, Rec: 1.00, F1: 0.67**	Acc: 0.95, Rec: 0.33, F1: 0.36	Acc: 0.82, Rec: 0.71, F1: 0.26
10	Acc: 0.93, Rec: 1.00, F1: 0.17	Acc: 0.91, Rec: 0.43, F1: 0.35	Acc: 0.92, Rec: 0.40, F1: 0.24

Evaluation

- Performance Comparison with other models
  - Use 5 minutes as input window size, gap and pre-failure size, 0.65 as pre-failure value
  - Show angular line because test data is not large (num : 129)
  - CNN show best performance



Conclusion

Conclusion

- Summary
  - We propose a model that analyze logs extracted from VMs which execute VNFs and determine whether failures will occur in the future
  - Use pre-failure tagging method to get higher performance
  - Could predict failures before 5 minutes with 0.67 of F1 score

- Future work
  - Gather more failures data
  - Learn about failures that occur on the server
  - Use CNN’s outputs as input of RNN
  - Apply to container based environment

감사합니다

Publications (1/2)

- International Conference Papers (3, 1 under review)

1. **Sukhyun Nam**, Jibum Hong, Jae-Hyoung Yoo, James Won-Ki Hong, “Virtual Machine Failure Prediction using Log Analysis”, The 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS 2021), under review.
2. Jiyoon Lim, **Sukhyun Nam**, Jae-Hyoung Yoo, James Won-Ki Hong, "Best next hop Load Balancing Algorithm with Inband network telemetry", 16th International Conference on Network and Service Management (CNSM 2020), Virtual Conference, Nov. 2-6, 2020.
3. **Sukhyun Nam**, Jiyoon Lim, Jae-Hyoung Yoo, James Won-Ki Hong, "Network Anomaly Detection Based on In-band Network Telemetry with RNN", The Fifth International Conference On Consumer Electronics (ICCE-Asia 2020), Seoul, Korea, Nov. 1-3, 2020.
4. Jiyoon Lim, **Sukhyun Nam**, Jae-Hyoung Yoo, James Won-Ki Hong, "Load Balancing Algorithm with Programmable Switch", The 21st Asia-Pacific Network Operations and Management Symposium (APNOMS 2020), Daegu, Korea, Sep. 23-25, 2020

Publications (2/2)

- Domestic Conference Papers (5)
  1. **남석현**, 홍지범, 유재형, 홍원기, "로그 및 자원 분석을 통한 VNF 고장 예측에 관한 연구", KNOM Conference 2021, On-line KNOM Conference Venue, Korea, April 30, 2021, pp. 17-20.
  2. 홍지범, **남석현**, 유재형, 홍원기, "가상 네트워크 관리를 위한 기계학습 기반 이상 탐지 시스템 설계", KNOM Conference 2021, On-line KNOM Conference Venue, Korea, April 30, 2021, pp. 120-122.
  3. 임지윤, **남석현**, 유재형, 홍원기, "강화학습 기반 링크가중치 조정 로드밸런싱 알고리즘 연구", KNOM Conference 2020, On-line KNOM Conference Venue, Korea, May 15, 2020, pp. 28-31.
  4. **남석현**, 임지윤, 유재형, 홍원기, "네트워크 텔레메트리 기반 통합 네트워크 관리 시스템 연구", KNOM Conference 2020, On-line KNOM Conference Venue, Korea, May 15, 2020, pp. 130-132.
  5. **남석현**, 현종환, 유재형, 홍원기, "네트워크 텔레메트리를 활용한 머신 러닝 기반 네트워크 이상 탐지 기법 연구", KNOM Conference 2019, Daegu, Korea, May. 30, 2019, pp. 75-77.
- Domestic Patents (2)
  1. 홍원기, 유재형, 임지윤, **남석현**, "네트워크 제어 방법 및 장치", 출원번호: 10-2019-0147915, 2019.11.18 (출원인: 포항공과대학교 산학협력단)
  2. 홍원기, 유재형, 임지윤, **남석현**, "네트워크의 이상 감지 방법 및 장치", 출원번호: 10-2019-0147898, 2019.11.18 (출원인: 포항공과대학교 산학협력단)

Appendix. Loss with Pre-failure Tagging

- KL Divergence
  - Measure the different degrees of the two probability distributions
  - $KL(p|q) := -\sum_{i=1}^{N} p_i \log q_i - \left(-\sum_{i=1}^{N} p_i \log p_i\right) = -\sum_{i=1}^{N} p_i \log\left(\frac{q_i}{p_i}\right)$ (N is # of classes)
  - Use custom loss based on KL divergence to learn even for pre-failure value
  - $Custom\ loss = y_{true} \times class_{weight} \times \log\left(\frac{y_{true}}{y_{pred}}\right) + (1 - y_{true}) \times \log\left(\frac{1-y_{true}}{1-y_{pred}}\right)$
- Keras KL Divergence loss function
  - $loss = y_{true} \times \log\left(\frac{y_{true}}{y_{pred}}\right)$
  - retun 0 for all normal state (since y_{true} is 0)
- Cross-entropy loss function
  - $Cross - entropy = -\sum_{i=1}^{N} p_i \log q_i$
  - Only work when prediction object is 0 either 1

Reference

[1] Ayush Goel, Sukrit Kalra, and Mohan Dhawan. Gretel: Lightweight failure localization for openstack. In Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies, CoNEXT ’16. 2016.

[2] W. Ji, S. Duan, R. Chen, S. Wang and Q. Ling, "A CNN-based network failure prediction method with logs," 2018 Chinese Control And Decision Conference (CCDC), 2018, pp. 4087-4090.

[3] Lin, Qingwei, et al. "Predicting node failure in cloud service systems." Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2018.

[4] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pages 1285–1298, 2017..

[5] D. Cotroneo, M. Grottke, R. Natella, R. Pietrantuono, and K. S. Trivedi, “Fault triggers in open-source software: An experience report," in Software Reliability Engineering (ISSRE), 2013 IEEE 24th International Symposium on. IEEE, 2013, pp. 178–187.

[6] G. Xiao, Z. Zheng, B. Yin, K. S. Trivedi, X. Du and K. Cai, "Experience Report: Fault Triggers in Linux Operating System: from Evolution Perspective," 2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE), 2017, pp. 101-111.

[7] Y. Kim, "Convolutional neural networks for sentence classification," In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1746–1751.

[8] Google, "word2vec," 2013. [Online]. Available: https://code.google.com/archive/p/word2vec/ .