NW-AI Self-evolving Framework for Fault-tolerant Robust Networks

Yousuke Takahashi, Hiroki Ikeuchi, and Akio Watanabe

Abstract

We at NTT laboratories are focusing on the research and development of zero-touch operations using artificial intelligence (AI) for early recovery from network failures, which requires learning a large amount of network failure data. We have established a framework that allows AI to learn autonomously by artificially generating a large amount of network failures in a simulated environment and learning how to respond to them. This article describes the concept of this framework and how AI learns in the framework.

Keywords: self-evolving AI, chaos engineering, digital twin

1. Introduction

NTT laboratories are aiming to develop robust networks that are resilient to unexpected failures. To achieve this goal, we are researching and developing how to automate network operations and ultimately achieve zero-touch operations using artificial intelligence (AI). At the heart of these efforts is an AI for network operations, which we call NW-AI.

In zero-touch operations, NW-AI operates and manages the network with minimal human intervention. Specifically, it automatically performs tasks such as anomaly detection and failure location estimation. For example, NW-AI can monitor network communication patterns and performance data to detect network anomalies. When a failure occurs, NW-AI can analyze network topology information and a large amount of log data to estimate the cause of the failure.

One of the most challenging tasks is the automated recovery of network failures by NW-AI. Automated recovery of network failures is a key element in maintaining network stability and quality of service, and its implementation is critical to the success of zero-touch operations.

To achieve automated recovery by NW-AI, it is necessary for NW-AI to learn a large amount of performance data and log data during failures. This enables it to acquire the ability to respond to various failure scenarios and respond to a failure quickly and efficiently when it occurs.

However, it is difficult to collect sufficient data on unexpected failures that occur under conditions that deviate from the general operating range because they occur infrequently; as a result, it is difficult to train NW-AI to respond appropriately to these failures. Unexpected failures can take a long time to recover and have a serious impact on users because recovery procedures have not been established. Unexpected failures occur under conditions that deviate from the general operating range, but it is impractical to manually search for such conditions due to the enormous amount of time and effort required.

To address these issues, NTT laboratories are engaged in research and development to achieve advanced coordination between verification and operation. Specifically, by combining a network digital twin environment and chaos engineering tools that enable artificial failure injection, NTT laboratories have established a framework in which NW-AI can autonomously continue to learn how to handle various failures. In this framework, the chaos engineering tool is run in a digital twin environment to generate various types of failures, and NW-AI learns how to respond to them. This process can be automated over a long period to generate an enormous number of

failures, enabling the collection of data on failures that would not occur in normal operations. Therefore, NW-AI can evolve to minimize unexpected failures by increasing the number of network failures it can handle.

2. Concept of the NW-AI self-evolving framework

The concept of the NW-AI self-evolving framework is shown in **Fig. 1**. The framework creates a digital twin environment that mimics a real-world network and artificially injects failure conditions into the environment to collect failure data necessary for NW-AI training. Chaos engineering tools are used to generate these failure conditions.

Chaos engineering is an experimental approach to improving the durability of a system by finding and correcting system weaknesses. This is done by intentionally causing failures in the system and observing the results. The goal with this approach is to understand how the system responds to unexpected problems and failures, thereby identifying and correcting system weaknesses. Chaos engineering tools are software that automate and manage this process.

A chaos engineering tool allows NW-AI to collect data on failures that rarely or never occur in real network environments by generating them in a copy environment. NW-AI learns how to respond to failures by learning from these data.

Thus, NW-AI trained on a large amount of data can be deployed in a real network environment to achieve automated recovery from network failures.

3. Building an automated recovery AI using the NW-AI self-evolving framework

Here is a step-by-step description of the process of building an automated recovery AI using the NW-AI self-evolving framework [1, 2]. The architecture for building an automated recovery AI using this framework is shown in **Fig. 2**.

The framework consists of an AI agent and environment, all actions of which can be automated by scripts; the AI agent and environment interact, and the recovery strategy is built autonomously in accordance with the following steps.

(1) Failure injection

In this step, various failures are injected into the target system using chaos engineering tools. Failure injection is an important process for identifying system vulnerabilities and developing countermeasures.

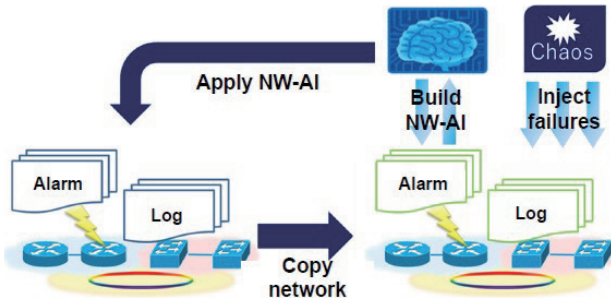

Fig. 1. Concept of NW-AI self-evolving framework.

The framework injects not only failures that occur frequently but also failures that occur infrequently but could have a significant impact when they do occur. This allows the system to manage a wide variety of failure situations.

(2) System-state observation

In this step, the AI agent observes the state of the target system and collects information about it. Specifically, the agent collects data such as various metrics and logs generated by the system over a certain period. These data are important indicators of system performance and status, and the agent uses them to understand the current state of the system and determine appropriate actions.

(3) Determine and execute recovery actions

In this step, the reinforcement learning module determines the optimal recovery action based on observed data and previous learning results, and the recovery action execution module executes the recovery action on the system. The recovery action is a specific action to restore the abnormal state of the system to normal and is determined based on the observed data and the current system state. For example, if a failure of a particular device is observed, restarting the device is selected as the recovery action. On the other hand, if a problem is detected in a network route, the route is reconfigured.

(4) Rewards based on health checks and observation of new system states

In this step, the health-check module monitors the system state and rewards the AI agent on the basis of the results. Reward settings are adjusted so that the closer the system state is to normal, the higher the reward, so that the agent learns behaviors aimed at normalizing the system. The health-check module's monitoring and reward settings provide an important feedback mechanism for the agent to properly understand the system state and choose the optimal behavior.

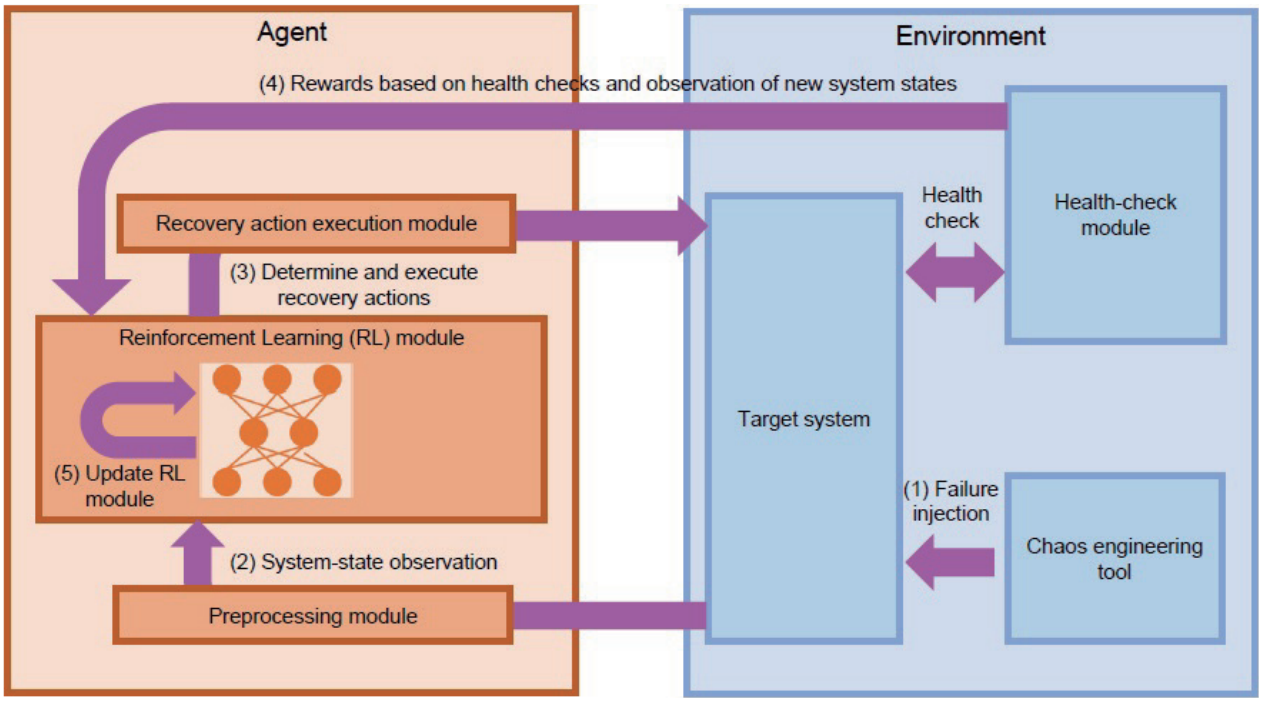

Fig. 2. Building an automated recovery AI using the NW-AI self-evolving framework.

(5) Update reinforcement learning module

In this step, the reinforcement learning module is updated using a combination of the system state observed by the agent, recovery action chosen, reward obtained, and newly observed system state. The agent stores this information and uses it as training data. Specifically, the agent learns how the selected action changed the system state and the degree to which it led to the reward. This improves the agent's ability to choose actions that produce better results when similar system conditions occur. Updating this reinforcement learning module is an important process for the agent to understand the system state and its changes and choose the best course of action. This allows the agent to continuously learn and evolve, improving the efficiency and stability of the system.

4. Future work

We have introduced the NW-AI self-evolving framework for developing fault-tolerant robust networks and the construction of an automated recovery AI using this framework. In the future, we aim to develop robust networks that are resilient to failures by building NW-AIs that automatically execute various operations, including automated recovery, using the NW-AI self-evolving framework.

References

[1] H. Ikeuchi, J. Ge, Y. Matsuo, and K. Watanabe, "A Framework for Automatic Failure Recovery in ICT Systems by Deep Reinforcement Learning," Proc. of 40th IEEE International Conference on Distributed Computing Systems (ICDCS 2020), pp. 1310–1315, Nov. 2020. https://doi.org/10.1109/ICDCS47774.2020.00170

[2] H. Ikeuchi, Y. Takahashi, K. Matsuda, and T. Toyono, "Recovery Process Visualization based on Automaton Construction," Proc. of 17th IFIP/IEEE International Symposium on Integrated Network Management (IM 2021), pp. 10–18, May 2021.

Yousuke Takahashi
Senior Research Engineer, Traffic Engineering Research Group, Communication Traffic, Quality and Operation Research Project NTT Network Service Systems Laboratories.
He received a B.E. and M.E. in information science and technology from Osaka University in 2007 and 2009. He joined NTT in 2009 and was engaged in research on network management and traffic engineering. From 2016 to 2019, he was engaged in research on network management at NTT Communications Corporation. He is currently engaged in research on network management using machine learning at NTT Network Service Systems Laboratories. He received the best paper award from the Institute of Electronics, Information and Communication Engineers (IEICE) in 2019. He is a member of IEICE.

Hiroki Ikeuchi
Researcher, Traffic Engineering Research Group, Communication Traffic, Quality and Operation Research Project, NTT Network Service Systems Laboratories.
He received a B.S. and M.S. in physics from the University of Tokyo in 2014 and 2016. He joined NTT in 2016. His research interest includes application of machine learning to network operations. He received the Network Systems Research Award from the IEICE Technical Committee on Network Systems and the Young Researcher's Award from IEICE in 2021. He is a member of IEICE.

Akio Watanabe
Senior Research Engineer, Traffic Engineering Research Group, Communication Traffic, Quality and Operation Research Project, NTT Network Service Systems Laboratories.
He received a B.E. and M.E. in informatics engineering from the University of Electro-Communications, Tokyo, in 2010 and 2012. He joined NTT in 2012 and studied optimized network management using machine-learning technologies. He is a member of IEICE.