

# 자가치유 시스템을 위한 하이브리드 예측모델

유길종<sup>1</sup>, 박정민<sup>2</sup>, 정철호<sup>3</sup>, 이은석<sup>4</sup>  
성균관대학교 컴퓨터공학과<sup>1234</sup>  
{nglover<sup>1</sup>, jmpark<sup>2</sup>, jesus98<sup>3</sup>, eslee<sup>4</sup>}@selab.skku.ac.kr

## Hybrid Prediction Model for Self-Healing System

Giljong Yoo<sup>1</sup>, Jeongmin Park<sup>2</sup>, Chulho Jung<sup>3</sup> and Eunseok Lee<sup>4</sup>  
Dept. of Computer Engineering, Sungkyunkwan University<sup>1234</sup>

### 요약

오늘날 분산 컴퓨팅 환경에서 운용되는 시스템이 증가함에 따라 시스템의 관리작업은 고수준(high-level)의 자동화에 대한 요구가 증가하고 있다. 이에 따라 시스템 관리방식이 전통적인 관리자 중심의 방식에서 시스템 스스로가 자신의 문제를 인식하고 상황을 분석하여 해결하는 자율 컴퓨팅 방식으로 변화하고 있으며, 이에 대한 연구가 많은 연구기관에서 다양한 방법으로 이루어지고 있다. 그러나 이러한 대부분의 기존 연구들은 문제가 발생한 이후의 치유에 주로 초점이 맞추어져 있다. 이러한 문제를 해결하기 위해서는 시스템 스스로가 동작환경을 인식하고 에러의 발생을 예측하기 위한 예측 모델이 필요하다. 따라서 본 논문에서는 자율 컴퓨팅 환경에서 자가 치유를 지원하는 4가지의 예측 모델 설계 방법을 제안한다. 본 예측 모델은 ID3 알고리즘, 퍼지 추론, 퍼지 뉴럴 네트워크 그리고 베이지안 네트워크가 각 시스템 상황에 맞춰 적절하게 사용되는 방식이며, 이를 통해 보다 정확한 에러 예측이 가능해진다. 우리는 제안모델의 평가를 위해 본 예측모델을 자가치유 시스템에 적용하여 기존 연구와 예측의 효율을 비교하였으며, 그 결과를 통해 제안 모델의 유효성을 증명하였다.

Keyword : Ubiquitous Computing, Self-Healing, Prediction Model, ID3 Algorithm, Fuzzy Inference, Fuzzy Neural Network, Bayesian Network

## 1. 서론

오늘날 분산 컴퓨팅 환경에서 운용되는 시스템이 증가함에 따라 시스템의 관리작업은 고수준(high-level)의 자동화에 대한 요구가 증가하고 있다. 예를 들면, 컴퓨터 이벤트의 실시간 스트림에 기반한 진단과 예측, 경고의 세팅, 그리고 지속적인 모니터링을 수행할 수 있는 것들이 포함된다. 자율 컴퓨팅[1]의 핵심은, 차세대 IT 시스템을 위해 최근 제안되고 있는, 시스템의 잠재적인 문제를 예측하고 실시간으로 데이터를 분석하여 이에 대처하는 능력이다. 현재 대부분의 자가 치유시스템(Self-healing)은 에러가 발생한 후 치유를 수행하는 방식으로, 치유의 효율이 떨어진다. 본 논문에서는 이러한 문제점을 극복할 수 있도록, 시스

템이 스스로의 문제를 인식하고 예측하기 위해서 시스템의 상황에 따라서 모델을 선택하여 적용할 수 있도록 하는 하이브리드한 예측 모델을 제안한다. 이를 위해 우리는 각각 ID3 알고리즘(ID3 Algorithm), 퍼지 추론(Fuzzy Inference), 퍼지 뉴럴 네트워크(Fuzzy Neural Network) 그리고 베이지안 네트워크(Bayesian Network)를 자가치유 시스템에 적용하였다.

## 2. 관련연구

### 2-1 기존의 자가 치유 시스템

자가 치유는 시스템 스스로가 자신의 정보를 분석 및 진단하고, 그에 대응 할 수 있는 시스템으로써, 자율 컴퓨팅의 기술 중 하나이다[2]. 현재

자가 치유 시스템은 IBM[3], Sun[4]과 같은 기업에서 논문들을 발표하는 등, 활발한 연구를 진행 중이다. 하지만 IBM의 경우 자사제품에서 발생하는 로그를 기반으로 습득된 데이터를 분석하여 이에 대처하는 방식으로써, 사후처리 중심으로 인한 뒤늦은 복구로 시스템 관리의 비효율성이라는 단점을 갖고 있다[5]. 이는 긴급한 상황에 대해서 즉각적인 행동이 부족하기 때문에 많은 치유 시간을 소비하게 된다. Sun의 자가 치유 시스템은 Solaris 10의 내부에 사전에 예측이 가능한 자가 치유 시스템을 연구, 개발하고 있다. 하지만 IBM과 마찬가지로 Sun에서 개발한 시스템에서 자가치유를 하기 때문에 한정된 요소에 의해 예측되게 되며, 특정 예측모델을 사용함으로써, 범용성이 떨어진 다. 이러한 문제를 해결하기 위해 우리의 제안 시스템에서는 여러 시스템에서 범용성을 갖는 예측 모델을 적용한 자가 치유 시스템을 설계, 구현 하였다.

### 2-2 자가 치유 시스템을 위한 예측 알고리즘

자가 치유 시스템에서 이용될 수 있는 예측 알고리즘은 ID3 알고리즘, 퍼지 추론, 퍼지 뉴럴 네트워크, 베이지안 네트워크가 있다. ID3 알고리즘은 주어진 데이터를 사용하여 결정 트리(Decision Tree)를 학습하는 알고리즘이다. 퍼지 추론은 개념(concept)이 적용되거나 적용되지 않는 상황 사이에 분명한 경계가 존재하지 않을 때, 애매 모호한 상황을 여러 근사값으로 구분 지어 놓는 논리이다. 퍼지 뉴럴 네트워크는 뉴럴 네트워크가 가지는 학습 능력과 퍼지를 결합한 것이다. 이를 통해 퍼지

추론의 단점을 보완할 수 있다. 퍼지 뉴럴 네트워크는 퍼지 규칙을 자동적으로 식별할 수 있으며, 역전파 알고리즘(Backpropagation algorithm)을 통해 연결 가중치를 수정함으로써 소속함수를 튜닝할 수 있다[6]. 마지막으로 베이지안 네트워크는 불확실한 조건하에 부분적인 신뢰도를 추론하기 위한 이론이다. 베이지안 정리의 조건부 독립을 확률이 부여된 방향성 비순환 그래프(DAG:Directed Acyclic Graph)로 표시한 것이 베이지안 네트워크이며 인과 네트워크(Casual Network), 믿음 네트워크(Belief Network)라고도 한다. 이 그래프의 각 노드는 신뢰도를 나타낸다. 신뢰도들은 확률론의 규칙에 따라서 조합되고 조종된다.

위 4가지의 알고리즘은 각각의 특징과 장점이 명확하며, [표 1]과 같이 구분이 될 수 있다.

우리는 이와 같은 관련 연구 분석을 바탕으로, 다양한 환경에서 적용 가능하도록 예측모델들을 상황에 맞게 선택할 수 있으며, 이를 이용하여 효과적으로 시스템을 치유 및 관리 할 수 있는 자가 치유 시스템을 설계, 구현 하였다. 다음 장에서는 이에 대한 구체적인 언급을 할 것이다.

### 3. 제안모델

시스템에 발생될 에러를 미연에 찾아내어 시스템의 성능을 항상 일정하게 유지시켜주기 위해 다음의 4가지 알고리즘을 사용하게 된다. 본 제안 시스템은 [표 1]과 같이 기술된 특성에 따라 주어진 시스템의 상황에 맞추어 적절한 알고리즘을 선택할 수 있으며, 그러한 알고리즘을 선택하여 자가 치유 시스템에서 적용될 수 있는 방법들

[표 1] 자가 치유 시스템을 위한 4가지 알고리즘에 대한 비교

	ID3 Algorithm	Fuzzy Inference	Fuzzy Neural Network	Bayesian Network
특징	치유에 대한 히스토리 정보(History Information)를 바탕으로 Decision Tree를 생성하여 입력된 데이터에 대한 추론 수행	시스템의 리소스(CPU, RAM, Bandwidth) 사용률에 대한 수준(level)을 추론하여 그 결과로부터 시스템의 상황을 판단	퍼지의 특징을 바탕으로 Membership Function에 대한 자동 튜닝 지원하여 학습적인 효과를 볼 수 있음	인과추론과 진단추론을 사용하여 조건부확률을 기반으로 현재 시스템 상태의 각 상황(Emergency, Error, Warn, Normal)에 대한 신뢰도를 추론
장점	단시간에 정확한 추론을 수행할 수 있음	여러 리소스타입에 대한 애매모호한(utilization의 정도) 시스템의 상황에 대해서 인간의 지식을 반영해 원하는 추론 결과를 얻을 수 있음	- Fuzzy Inference의 단점 보완 - 학습 가능	- 부분적인 Backtracking 기능을 지원 - 즉, 인과추론 이외에 진단 추론을 이용하여 결과로부터 원인을 추론하는 것이 가능 함
단점	- Decision Tree로 정해져 있는 범위 안에서만 추론 가능 - Backtracking이 지원되지 않음	- 개발자의 개입이 많음 (소속함수 입력) - 규칙 생성의 비효율성	- 원하는 출력값이 나오기까지 학습의 시간이 오래 걸릴 수 있음 - 구현의 어려움	네트워크의 구성이 잘못되었을 때, 수정이 어려움
수행 속도	67 Milliseconds	1000 Milliseconds	원하는 데이터의 복잡성에 따라 다름	67 ~ 1000 Milliseconds



(Linguistic information + Numeric information)

- 비선형에 관한 정보 처리의 용이성
- 인간이 읽기 쉬운 지식

(Human-readable knowledge)

제안 시스템에서 이용하는 퍼지 추론의 소속함수의 정도는 총 5 가지 단계로 나뉘게 된다. 즉, MH, H, N, L, ML 은 다음의 식으로써 만들어 진다. 식(1)의 Duration degree 는 리소스가 그 상태를 유사하게 유지하는 시간을 [0.0, 1.0]으로 나타내는 정도이다.

$$Level = \frac{ResourceUtilization}{100} \times Duration\ degree \quad (1)$$

여기서 소속함수의 정도와 그로부터 생겨나는 룰들은 개발자 및 사용자가 직접 입력을 해줘야 한다. 퍼지 추론을 하기 위한 룰은 식(2)처럼 총 125 개가 생성된다.

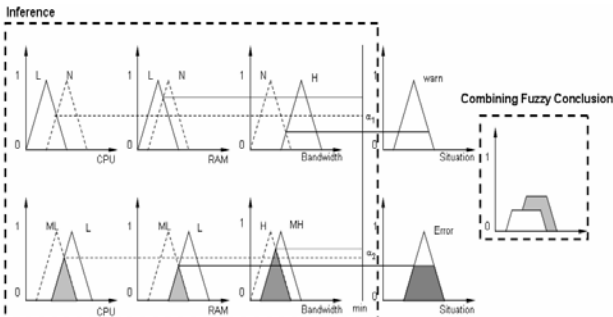
$$\begin{aligned} & CPU\_Level \times RAM\_Level \times Bandwidth\_Level \\ & = 5 \times 5 \times 5 \\ & = 125 Rules \end{aligned} \quad (2)$$

룰은 [그림 3]과 같은 기술 방법을 이용하여 표현 된다.

R1: If CPU is MH and RAM is N and Bandwidth is N then Situation is Warn  
 R2: If CPU is H and RAM is L and Bandwidth is N then Situation is Normal  
 ...  
 R125: ...

[그림 3] 퍼지규칙의 기술방법

퍼지추론을 하기 위한 순서는 조건부 만족도 검사(Fuzzy Matching), 추론(Inference): Mamdani model, 퍼지결론 조합(Combining Fuzzy Conclusion), 비퍼지화(Defuzzification)의 4 단계로 구분된다[6].



[그림 4] 퍼지 추론의 방법 및 추론 결과의 결합의 예 첫째, 식(3)과 같이 조건부 만족도를 검사하기 위해 다음과 같은 식을 적용한다. MD 는 조건부

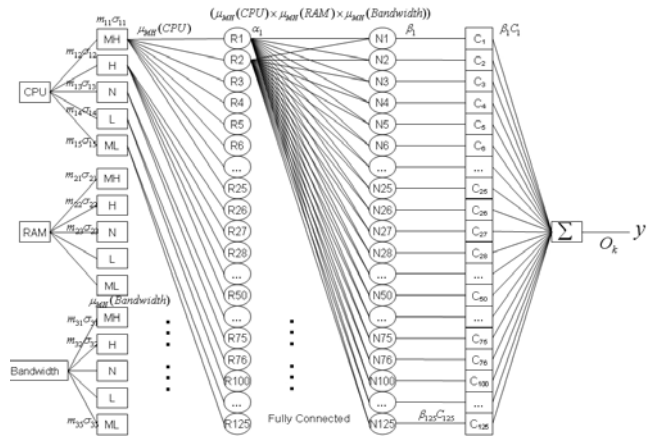
만족도의 결과를 나타내고, A 는 리소스 타입, i 는 룰의 개수를 나타낸다.

$$I = (CPU, RAM, Bandwidth) \quad (3)$$

$$Min(\mu_{Ai}(CPU), \mu_{Ai}(RAM), \mu_{Ai}(Bandwidth)) = MD_{RULEi}$$

위의 식을 적용 각 룰에 대해서 해당되는 룰을 찾게 되고 그 룰과 어느 정도 매칭되는지를 판단한다. 둘째, 추론 단계에서 [그림 4]와 같은 방법으로 각 리소스들의 상태를 최소화된 값을 출력 함수로 보내어 셋째, 추론 결과들의 합집합을 구하게 된다. 마지막으로 이렇게 만들어진 [그림 5]의 퍼지 모델(Combining Fuzzy Conclusion)을 바탕으로 마지막으로 비퍼지화(Crisp value)를 수행하게 된다.

### 3-3 퍼지 뉴럴 네트워크(Fuzzy Neural Network)



[그림 5] 퍼지 뉴럴 네트워크를 이용한 상황 예측 모델

[그림 5]의 소속함수와 가중치 C 는 뉴럴네트워크에 의해 자동 튜닝이 지원된다.  $\eta$  은 학습률, K 는 출력층의 노드의 수,  $t_k$  는 기대되는 출력값,  $O_k$  는 실제 출력값, C 는 가중치값, m 은 소속함수에서의 해당되는 출력값,  $\sigma$  는 현재 소속함수의 X 좌표를 나타낸다.

퍼지 뉴럴 네트워크의 최종 출력값은 식(4)과 같다.

$$O_k = \frac{\alpha_1 C_1 + \alpha_2 C_2 + \dots + \alpha_{125} C_{125}}{\alpha_1 + \alpha_2 + \dots + \alpha_{125}} = \frac{\alpha_1}{\sum \alpha_i} C_1 + \frac{\alpha_2}{\sum \alpha_i} C_2 + \dots + \frac{\alpha_{125}}{\sum \alpha_i} C_{125} \quad (4)$$

$\alpha_i$  는 그림의  $\beta_1$  을 말한다.  $O_k$  는 실제 출력값  $\sum \alpha_i$

으로써 기대되는 출력값과 다를 수가 있다. 이것은 위에서 기술된 에러 역전과 학습알고리즘으로 해결될 수 있다. 이것은 조금씩 가중치에 변화를 주어 반복적으로 퍼지 뉴럴 네트워크를 수행함으로써 에러를 줄이게 된다. 이 가중치는 식(5)의

에러( $E$ )를 최소화하기 위해 변경된다.

$$E = \frac{1}{2} \sum_k (t_k - O_k)^2 = \frac{1}{2} \sum_k E_k \quad (5)$$

$$E_k = (t_k - O_k)^2, \quad \frac{\partial E}{\partial C_i} = \sum_k \frac{\partial E_k}{\partial C_i}$$

$$\frac{\partial E_k}{\partial C_i} = \frac{\partial}{\partial C_i} (t_k - O_k)^2 = \frac{\partial O_k}{\partial C_i} \frac{\partial (t_k - O_k)^2}{\partial O_k}$$

$$\frac{\partial E}{\partial m_{ij}} = \frac{\partial E}{\partial O_k} \frac{\partial O_k}{\partial \beta} \frac{\partial \beta}{\partial \alpha} \frac{\partial \alpha}{\partial m_{ij}}$$

$$(\Delta C_i = -\eta \frac{\partial E}{\partial C_i}, \Delta m_{ij} = -\eta \frac{\partial E}{\partial m_{ij}}, \Delta \sigma_{ij} = -\eta \frac{\partial E}{\partial \sigma_{ij}})$$

### 3-4 베이저안 네트워크(Bayesian Network)

본 논문은 주어진 신뢰도를 바탕으로 시스템의 상황을 예측하는 방법에 인과추론과 진단추론을 이용하였다. 인과추론(casual inference)은 주어진 원 인변수로부터 결과변수를 추론하는 것을 말하며, 진단추론(diagnostic inference)은 주어진 결과로부터 원인을 추론하는 것으로써, 예를 들어 “시스템이 응급한 상태가 되었을 때 CPU 가 많은 사용률을 보일 확률은 얼마인가?”와 같은 질문에 대한 답을 계산할 수 있다. 다음은 본 시스템에서 자가 치유를 위해 베이저안 네트워크를 적용한 경우에 대한 결과이다.

“CPU 가 MH(Max High)일 때 시스템이 응급한 상황이 될 확률은 어떻게 되는가?”

위의 질문에 대해 CPU 와 RAM 은 각각 다음과 같은 신뢰도를 갖는다고 가정하자.

$$P(CPU\_MH) = 0.9$$

$$P(RAM\_H) = 0.6$$

CPU_MH	RAM_H	Emergency
T	T	0.95
T	F	0.8
F	T	0.4
F	F	0.1

위와 같은 조건을 가지고 식(6)과 같이 계산할 수 있다.

$$p(Emergency = 참 | CPU\_MH = 참)$$

$$= P(Emergency = 참, RAM\_H = 참 | CPU\_MH = 참)$$

$$+ P(Emergency = 참, RAM\_H = 거짓 | CPU\_MH = 참)$$

$$P(Emergency = 참 | CPU\_MH = 참) \quad (6)$$

$$= P(Emergency = 참 | RAM\_H = 참, CPU\_MH = 참)$$

$$P(RAM\_H = 참 | CPU\_MH = 참)$$

$$+ P(Emergency = 참 | RAM\_H = 거짓, CPU\_MH = 참)$$

$$P(RAM\_H = 거짓 | CPU\_MH = 참)$$

$$= (0.95 \cdot 0.6) + (0.8 \cdot 0.4) = 0.89$$

위와 같은 방법으로 CPU 가 MH 가 아닐 때를 구

하면 식(7)과 같다.

$$P(Emergency = 참 | CPU\_MH = 거짓)$$

$$= P(Emergency = 참 | RAM\_H = 참, CPU\_MH = 거짓)$$

$$P(RAM\_H = 참) \quad (7)$$

$$+ P(Emergency = 참 | RAM\_H = 거짓, CPU\_MH = 거짓)$$

$$P(RAM\_H = 거짓) = (0.4 \cdot 0.6) + (0.1 \cdot 0.4) = 0.28$$

다음의 질문은 진단 추론을 이용하는 방법을 제안 시스템에 적용한 것을 기술한 것이다.

“시스템이 응급한 상황일 때 CPU 가 MH(Max High)일 확률은 얼마나 되는가?”

베이저안 규칙에 의해, 식(8)과 같이 정리된다.

$$P(CPU\_MH = 참 | Emergency = 참)$$

$$= \frac{P(Emergency = 참 | CPU\_MH = 참)P(CPU\_MH = 참)}{P(Emergency = 참)}$$

$$(0.89 \cdot 0.9) / P(Emergency = 참) = 0.801 / P(Emergency = 참) \quad (8)$$

$$P(CPU\_MH = 거짓 | Emergency = 참)$$

$$= \frac{P(Emergency = 참 | CPU\_MH = 거짓)P(CPU\_MH = 거짓)}{P(Emergency = 참)}$$

(0.28 · 0.1) / P(Emergency = 참) = 0.028 / P(Emergency = 참)

두 경우의 합은 1 이 되어야 하기 때문에 P(Emergency=참)의 값은 0.829 가 된다. 이 값을 위의 두 식에 대입해보면 시스템이 응급한 상황일 때 CPU 의 상태에 대한 확률을 계산할 수 있게 된다.

$$0.801 / P(Emergency = 참) + 0.028 / P(Emergency = 참) = 1$$

$$P(Emergency = 참) = 0.829$$

$$P(CPU\_MH = 참 | Emergency = 참) = 0.97 \quad (9)$$

$$P(CPU\_MH = 거짓 | Emergency = 참) = 0.03$$

위의 식(9)와 같은 방법으로 베이저안 네트워크를 사용하면 시스템 상황의 결과를 바탕으로 역으로 리소스의 상태도 추론할 수 있는 장점이 있다.

지금까지 총 4 가지의 모델을 살펴보았으며, 각각의 모델들은 위에서 기술된 장단점과 특성을 바탕으로 시스템에서 요구하는 예측의 기능에 따라 하이브리드하게 사용될 수가 있다.

## 4. 평가

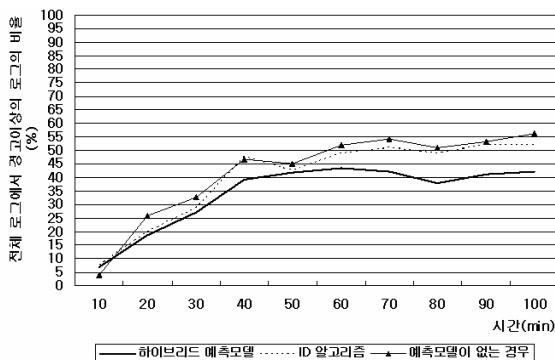
본 논문에서 제안한 하이브리드 예측 모델의 성능을 평가하기 위해, 자가치유시스템의 프로토타입을 구현하여, 본 알고리즘을 적용하였으며, ID3 알고리즘만을 수행하였을 경우와 비교를 수행하였으며, 다음과 같은 항목을 평가하였다.

- 시스템을 운영하면서 시간의 흐름에 따른 예러 수의 비교
- 시스템을 운영하면서 발생하는 상황에 대한 예측 시간

본 실험에는 OS 는 Windows XP professional 이고, CPU 는 Intel Pentium 3GHz, RAM 512Gbyte 의 성능을 가지며, 100.0Mbps 를 지원하는 유선랜으로 연결되어있는 데스크톱 PC 를 실험환경으로 하였다. 위와 같은 실험환경에서 이전에 제안한 자가치유 시스템[7]을 설치하여 하이브리드한 알고리즘 사용에 대한 효율성을 평가하였다.

먼저 첫 번째 실험을 위해 우리는 예측을 위해 ID3 알고리즘만을 이용하였던 기존 시스템과의 비교를 수행하였다. 실험 결과, 우리는 하이브리드 예측 모델을 사용함으로써 시스템의 요구에 따라 적합한 알고리즘을 적용함으로써 전체 로그에서 경고(warning)이상의 로그 비율이 감소함을 알 수 있었다. 위 실험의 결과는 [그림 6]과 같다.

두 번째 실험을 위해 우리는 예측하는데 걸리는 시간을 측정하였다. 먼저 학습된 데이터가 일정 수준 이상을 갖추고 있다고 가정하고, 측정을 위해 ID3 알고리즘만을 사용한 경우에 에러의 수에 대해 시스템의 상황을 판단하는 정도와의 비교를 통해 제안 모델의 효율성을 증명하였다. ID3 의 특성에 맞게 주어진 데이터에 대해 하이브리드 모델을 적용한 것과 마찬가지로 에러의 수가 적을수록 비슷한 성능을 나타내지만 에러의 수가 많아 질수록 리소스의 정보뿐만 아니라 시스템의 특성을 반영하여 예측을 하는 하이브리드 모델의 예측시간이 더 단축되는 것을 알 수 있다.



[그림 6] 시간의 흐름에 따른 에러 수의 변화

## 5. 결론

지금까지 우리는 자율 컴퓨팅 환경에서 자가 치유 시스템을 위해 시스템의 상황을 예측할 수 있는 예측 모델을 설계하기 위한 알고리즘을 정리하

였다. 본 논문에서 제안된 4 가지의 알고리즘은 서로 가지고 있는 특징이 명확히 구분되며 시스템의 요구에 따라 하이브리드하게 예측 모델을 설계하기 위해 사용될 수 있다. 향후 연구에 대한 방향은 다음과 같다.

- Context-aware Prediction Model: 모니터링 기능을 강화하여 리소스와 시스템의 단편적인 요소들 뿐만 아니라, 애플리케이션의 성능에 관계된 파라미터, 데이터의 흐름과 같은 것들을 인식하고 이러한 정보를 예측 모델에 적용하기 위한 연구를 진행하려고 한다.

- Self-growing Prediction Model: 이 연구의 핵심 알고리즘은 퍼지 뉴럴 네트워크다. 학습의 기능을 강화하여 Context-aware 의 기능을 결합하여, 스스로 학습하고 자가 성장 할 수 있는 예측 모델을 연구하려 한다.

## 참고문헌

- [1] R.K. Sahoo, A. J. Oliner, I.Rish, M. Gupta, J.E. Moreira, S. Ma, "Critical Event Prediction for Proactive Management in Large-scale Computer Clusters", ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 426-435, 2003
- [2] A. G. Ganek and T. A. Corbi, "The dawning of the autonomic computing era", IBM Systems Journal, Mar.2003
- [3] B. Topol, D. Ogle, D. Pierson, J. Thoensen, J. Sweitzer, M. Chow, M. A. Hoffmann, P. Durham, R. Telford, S. Sheth, T. Studwell, "Automating problem determination: A first step toward self-healing computing system", IBM white paper, Oct. 2003
- [4] Sun Microsystems: Predictive Self-Healing in the Solaris 10 Operating System, <http://www.sun.com/bigadmin/content/selfheal>
- [5] Jeongmin Park, Hyunsang Youn and Eunseok Lee, "A Multi-agent Based Context Aware Self-healing System", LNCS3578, pp. 515-523, Jul.2005
- [6] Kwang H.Lee, 'First Course on Fuzzy Theory and Applications', Advances in Soft Computing, Springer, 2005
- [7] Jeongmin Park, Giljong Yoo and Eunseok Lee, "Proactive Self-Healing System based on Multi-Agent Technologies", IEEE Computer Society, pp. 256-263, Aug.2005