
Use of Uncertainty with Autoencoder Neural Networks for Anomaly Detection

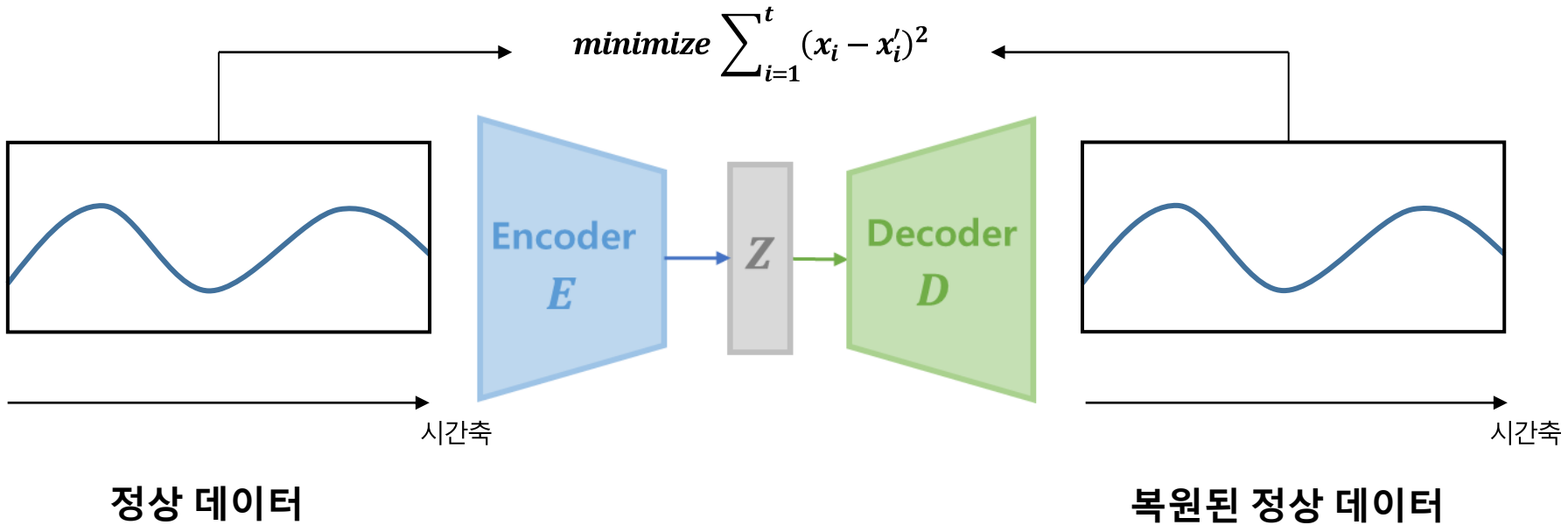
23.05.02

이정민

연구 배경

❖ 재구축 기반 이상치 탐지(Autoencoder)

- Dimensionality reduction
- Feature learning

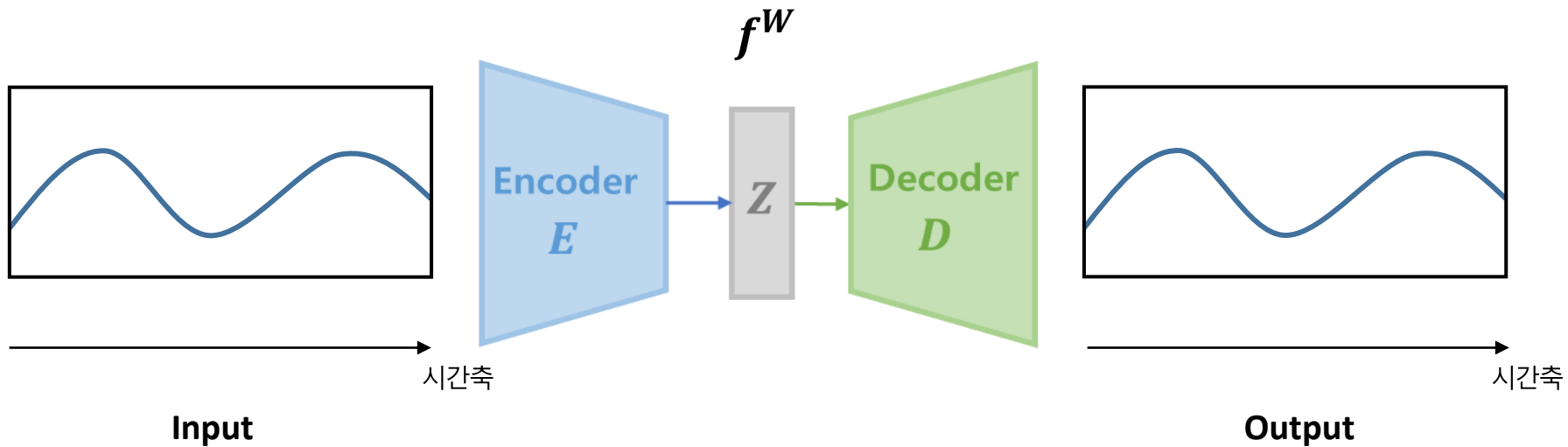


❖ Use of Prediction Uncertainty in Anomaly Detection

- Prediction Uncertainty를 포함한 새로운 score function 제안
- 기존 score function들과의 성능 비교

❖ Standard score functions

- Input과 output 사이의 거리 기반

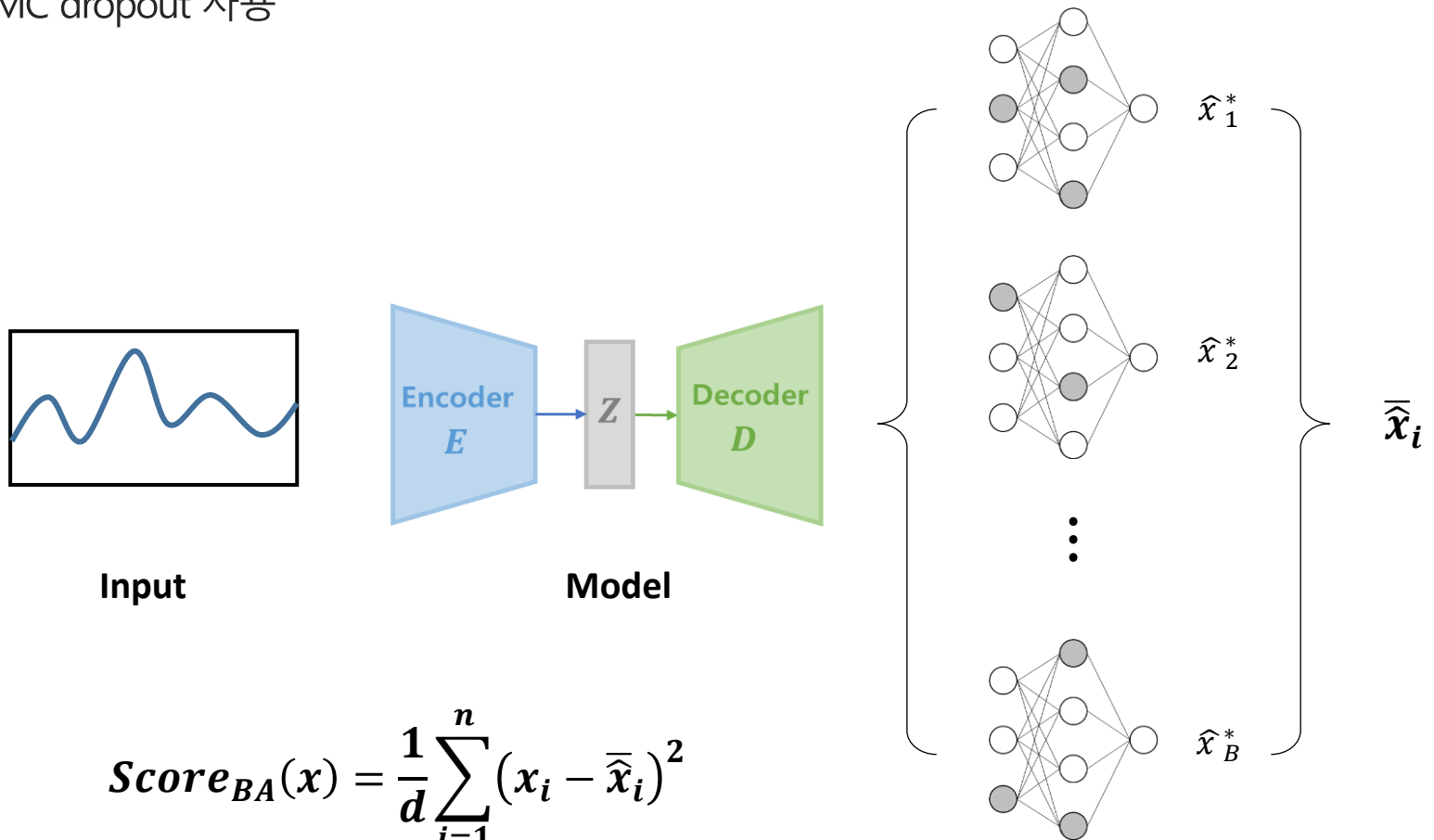


$$Score_s(x) = \frac{1}{d} \sum_{i=1}^n (x_i - f^W(x)_i)^2$$

$$\bar{\hat{x}}_i = \frac{1}{B} \sum_{b=1}^B \hat{x}_b^*$$

❖ Bayesian approximation based score functions

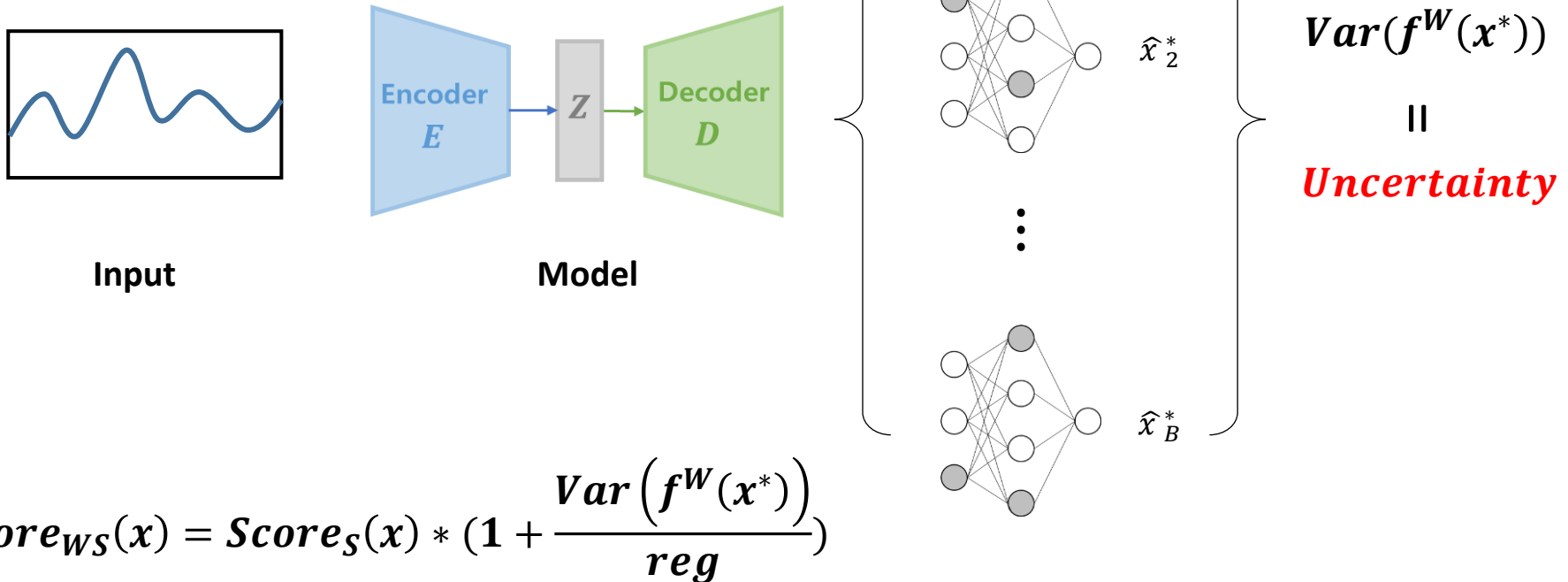
- MC dropout 사용



$$Score_s(x) = \frac{1}{d} \sum_{i=1}^n (x_i - f^W(x)_i)^2$$

❖ Score functions weighted by uncertainty

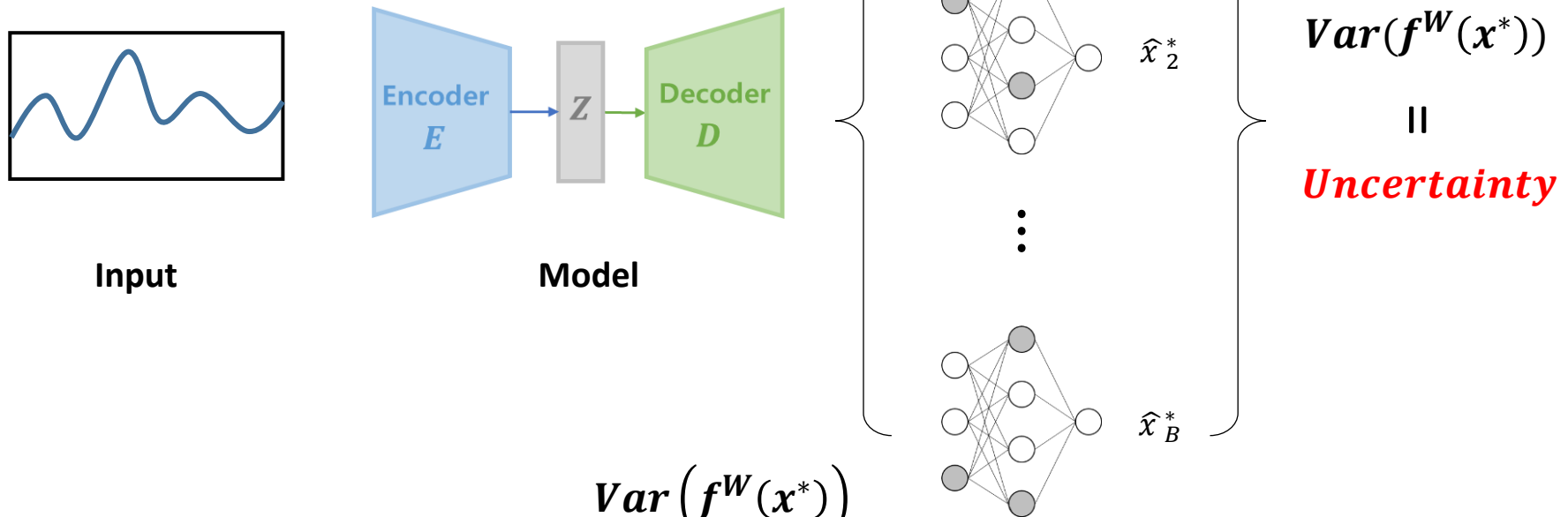
- MC dropout 사용
- Standard score를 불확실성으로 스케일링



$$Score_{BA}(x) = \frac{1}{d} \sum_{i=1}^n (x_i - \bar{\hat{x}}_i)^2$$

❖ Score functions weighted by uncertainty

- MC dropout 사용
- Bayesian score를 불확실성으로 스케일링



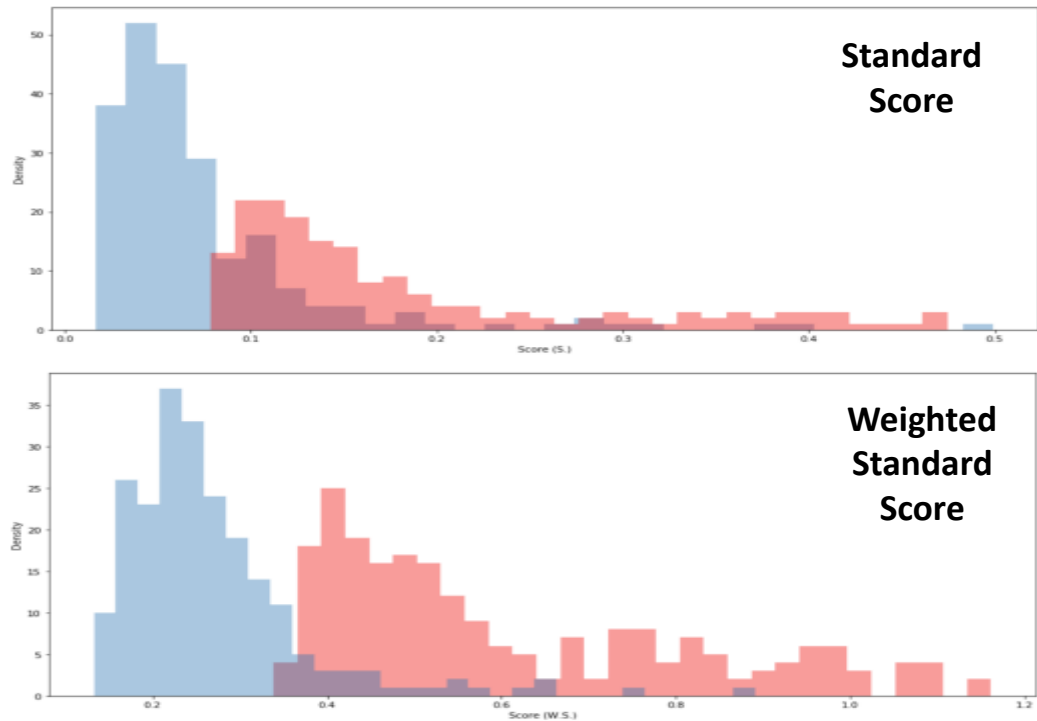
$$Score_{WBA}(x) = Score_{BA}(x) * \left(1 + \frac{Var(f^W(x^*))}{reg}\right)$$

실험 결과

❖ Score 분포 비교

- 불확실성으로 score를 스케일링 한 경우 이상과 정상의 score 차이가 더 명확해짐

정상 ————
이상 ————



실험 결과

❖ 실험 결과

- 불확실성으로 score를 스케일링 할 경우가 대부분 좋은 성능을 보임

TABLE II
EVALUATION OF THE DIFFERENT SCORE FUNCTION.

Dataset		AUC ROC for given score function			
Name	% Anomaly	S.	B.A.	W.S.	W.B.A.
LRS	2.2	0.7557	0.7571	0.7561	0.7577
	5.4	0.6959	0.6932	0.6971	0.6942
	10	0.6408	0.6506	0.6375	0.6458
Isolet	1.2	0.9075	0.909	0.914	0.9148
	3.5	0.8743	0.8718	0.8814	0.8807
	6.8	0.8115	0.8101	0.8193	0.8173
Satimage	2.9	0.9386	0.936	0.966	0.9645
	5.7	0.9159	0.9061	0.9469	0.9383
	10.8	0.824	0.805	0.8607	0.8447
Ionosphere	4.6	0.9927	0.9930	0.9924	0.9913
	10.6	0.9743	0.9725	0.9718	0.9706
	18.8	0.985	0.9868	0.9831	0.9837
WBC	5.2	0.9212	0.9188	0.9268	0.9316
	9.7	0.9184	0.918	0.9212	0.9236
	15	0.7748	0.7796	0.788	0.8036
Musk	7.2	0.9868	0.9871	0.9854	0.9854
	10.1	0.9288	0.9277	0.931	0.93
	16.3	0.9269	0.9263	0.9285	0.9289
Letter Recognition	1.9	0.8887	0.887	0.8884	0.8866
	3.3	0.8572	0.8583	0.847	0.8463
	6.3	0.8327	0.834	0.8247	0.8256