

자가학습 가능한 SVM 기반 가스 분류기의 설계

Design of SVM-Based Gas Classifier with Self-Learning Capability

정우재*, 정윤호*

Woojae Jeong*, Yunho Jung*

Abstract

In this paper, we propose a support vector machine (SVM) based gas classifier that can support real-time self-learning. The modified sequential minimal optimization (MSMO) algorithm is employed to train the proposed SVM. By using a shared structure for learning and classification, the proposed SVM reduced the hardware area by 35% compared to the existing architecture. Our system was implemented with 3,337 CLB (configurable logic block) LUTs (look-up table) with Xilinx Zynq UltraScale+ FPGA (field programmable gate array) and verified that it can operate at the clock frequency of 108MHz.

요약

본 논문은 실시간 자가학습과 분류 기능을 모두 지원하는 support vector machine (SVM) 기반 가스 분류기의 하드웨어 구조 설계 및 구현 결과를 제시한다. 제안된 가스 분류기는 학습 알고리즘으로 modified sequential minimal optimization (MSMO)을 사용하였고, 학습과 분류 기능을 공유구조를 사용하여 설계함으로써 기존 논문 대비 하드웨어 면적을 35% 감소시켰다. 설계된 가스 분류기는 Xilinx Zynq UltraScale+ FPGA를 사용하여 구현 및 검증되었고, 108MHz의 동작 주파수에서 3,337개의 CLB LUTs로 구현 가능함을 확인하였다.

Key words : gas classifier, machine learning, support vector machine, MSMO, FPGA

1. 서론

실내외 가스 유출은 사람의 건강에 심각한 문제를 일으킨다. 연소성과 발암성의 특성을 갖는 무색 무취의 유독가스에 일정 농도 이상으로 노출될 경우, 질병을 유발할 수 있으며, 심하면 죽음에 이를 수 있다. 따라서 산업현장, 인구밀집지역, 가정, 학

교 등 다양한 환경에서 인명피해를 막기 위해 유독 가스를 감시하는 시스템은 필수적이다[1].

가스 감시 시스템은 특정 가스를 정확히 검출하기 위해 높은 분별력, on-chip 집적화 기능, 민감한 감도, 그리고 다양한 종류의 가스를 검출할 수 있는 반도체 기반의 마이크로 가스 센서를 사용한다. 하지만 가스 센서는 특정 가스에 대한 과다노출,

* Dept. of Electronics and Information Engineering, Korea Aerospace University

★ Corresponding author

E-mail : yjung@kau.ac.kr, Tel : +82-2-300-0133

※ Acknowledgment

This work was supported by the Technology Innovation Program, 10073122, funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea) and CAD tools were supported by IDEC.

Manuscript received Dec. 12, 2019; revised Dec. 26, 2019; accepted Dec. 30, 2019.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

온도 및 습도의 변화 등으로 노화가 진행된다[2]. 따라서 같은 환경에서 같은 가스에 대해 다른 가스로 분류할 수 있는 문제가 발생할 수 있다. 그러므로 센서의 상태 및 환경을 파악하여 실시간으로 자가학습할 수 있는 기계학습 기반의 시스템이 필요하다.

최근까지 가스의 분류를 위하여 k-nearest neighborhood(KNN)[3], binary decision tree(BDT)[4], multilayer perceptron(MLP)[5], support vector machine(SVM)[6] 등의 기계학습 기반 알고리즘이 적용되어 왔다. 이중 KNN과 BDT는 주로 선형 데이터를 사용하는 경우에 적용된다. 하지만 가스 센서의 개수가 증가할수록 필연적으로 데이터의 비선형성이 증가하므로, 가스 응용에서 두 알고리즘을 사용하는 데 어려움이 있다. MLP는 비선형 데이터에 대해 적용할 수 있다. 하지만 MLP는 학습 시간이 오래 걸릴 뿐만 아니라 과적합 문제가 발생하여 좋지 않은 성능을 가질 수 있다. 이에 반해, SVM은 간단한 구조를 가짐에도 불구하고 높은 예측 성능과 우수한 일반화 성능을 가진다. 또한, 가스 응용에서 가장 우수한 성능을 보이는 것으로 제시되어 있다.

SVM은 기본적으로 계산량이 많고 복잡한 quadratic programming(QP) 문제를 풀어야 하므로, 구현을 위하여 효율적인 학습 알고리즘을 사용하는 것이 중요하다. 기존의 SVM 학습 기법으로 chunking 알고리즘[7], Osuna 알고리즘[8], shrinking 알고리즘[9], digesting 알고리즘[10], Gilbert 알고리즘[11], sequential minimal optimization(SMO) 알고리즘[12] 등이 있다. Chunking 알고리즘은 값이 0인 라그랑주 승수를 삭제함으로써 QP 문제의 크기를 줄인 후 학습을 수행한다. 하지만 학습 데이터가 많을 경우, 여전히 계산량이 많은 문제점이 있다. Osuna, shrinking, digesting 알고리즘은 학습 데이터를 처리하기 위해 decomposition 방법을 사용하여 working set 단위로 학습을 진행한다. 하지만 학습 데이터가 많아질수록 선형적으로 working set도 증가하기 때문에 학습에 필요한 메모리와 계산 시간도 함께 증가한다. Gilbert 알고리즘은 많은 문제를 풀 때, 최종 해를 찾는 데 많은 시간이 소요되어 속도가 느리다. SMO는 decomposition 방법을 사용하지만, chunk size를 2로 고정하기 때문에, 많은 학습 데이터를 처리할 수 있다. 하지만 SMO 알고리

즘은 최적성 조건 만족 여부를 확인하는 과정에서 비효율성을 가지고 있다. 이로 인해 학습 과정에서 불필요한 iteration이 발생하고 또한 하드웨어 설계를 위해서 복잡한 제어가 필요하다[13].

Modified SMO(MSMO)는 이러한 문제를 해결하기 위해 최적화 과정에서 학습 데이터의 subset의 boundaries를 사용하여 샘플들의 최적성 조건 만족 여부를 확인한다. 이를 통해 SMO의 비효율성을 해결하고 더 적은 iteration으로 효율적인 학습을 수행한다[13].

본 논문에서는 MSMO 알고리즘을 사용하여 가스 센서 응용을 위한 학습을 지원하는 SVM 기반의 가스 분류기를 제안한다. 또한, 학습과 분류 알고리즘의 분석을 통한 공유구조의 사용을 통해 최적의 SVM 하드웨어 구조를 제안하고, FPGA (field programmable gate array) 기반 구현 및 검증 결과를 제시한다. 본 논문의 구성은 다음과 같다. II장에서는 SVM의 구조와 MSMO 알고리즘을 설명하고, UCI gas dataset 기반 성능 평가 결과를 제시한다. III장에서는 제안된 SVM 하드웨어 구조와 각 unit별 구조를 설명한다. IV장에서는 FPGA 기반 구현 결과를 제시하고, 마지막으로 V장에서는 본 논문의 결론을 맺는다.

II. Support Vector Machine(SVM)

1. SVM 구조

SVM은 기계학습 알고리즘 중 하나로, 선형 데이터뿐만 아니라, 커널 연산을 활용하여 비선형 데이터의 분류에도 사용할 수 있어서 다양한 응용에 활용되어왔다[14-16]. 대부분의 기계학습 알고리즘은 학습 오류를 최소화하는 경험적 위험 최소화 원칙을 사용하여 학습을 진행하는 반면에, SVM은 전체 집단을 하위 집단으로 세분화한 뒤, 이 집단에 대한 경험적 위험도를 최소화하는 구조적 위험 최소화 원칙을 사용하여 학습을 진행한다. 이에 따라, SVM은 간단한 구조를 가짐에도 불구하고 높은 예측 성능과 우수한 일반화 성능을 가진다.

SVM은 학습을 통해 그림 1과 같이 margin을 최대로 하는 결정 경계($f(X)$)를 찾은 후 분류 작업을 수행한다. Margin은 결정 경계로부터 가장 가까운 벡터까지의 거리의 두 배를 뜻하고, b 는 bias를 의미하는 상수이다. 이때 결정 경계에서 가장 가까운

거리에 존재하는 벡터를 support vector라고 한다. 즉, SVM의 학습은 가장 좋은 결정 경계를 가지도록 하는 support vector를 찾는 과정이다.

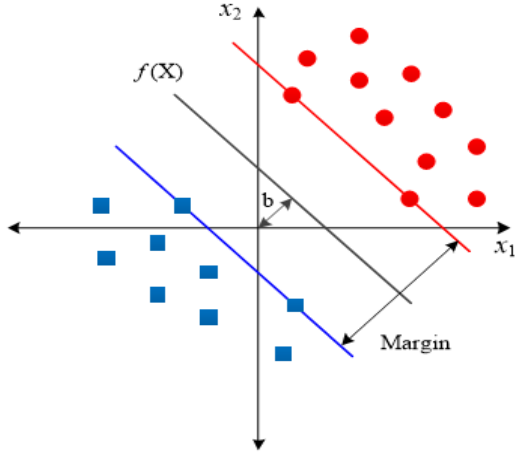


Fig. 1. Architecture of SVM.
그림 1. SVM의 기본 구조

2. MSMO 알고리즘

SVM은 식 (1)을 풀어서 결정 경계를 찾는다.

$$W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j K(X_i, X_j) \alpha_i \alpha_j \quad (1)$$

여기서, 모든 i 에 대해 $0 \leq \alpha_i \leq C$ 이어야 하고, 식 (2)를 만족하여야 한다.

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad (2)$$

식 (1)에서 α 는 조건부 최적화 문제를 풀기 위해 도입된 라그랑주 승수이다. $W(\alpha)$ 는 라그랑주 승수에 대한 함수를 뜻하고, y_i 는 i 번째 샘플의 클래스를 뜻한다. i 번째 샘플의 클래스가 1이면 $y_i = 1$, i 번째 샘플의 클래스가 2이면 $y_i = -1$ 이다. X_i 는 i 번째 샘플을 뜻하며, $K(X_i, X_j)$ 는 i 번째 샘플과 j 번째 샘플의 커널 함수를 통한 연산값을 의미한다.

커널 함수는 선형 분리가 가능하지 않는 샘플을 더 높은 차원의 새로운 공간으로 매핑하여 비선형 분류를 가능하게 한다. 커널 함수에는 다양한 함수가 존재하는데, 본 논문에서는 식 (3)과 같은 하드웨어 친화적 커널을 사용한다.

$$K(X_i, X_j) = 2^{-\gamma \|X_i - X_j\|_1} \quad (3)$$

식 (3)에서 $\|X_i - X_j\|_1$ 은 i 번째 샘플과 j 번째 샘플

간의 1-norm을 계산한 결과이고, γ 는 사용자 설정 상수이다.

MSMO 알고리즘은 식 (1)의 $W(\alpha)$ 를 최대화하는 라그랑주 승수를 찾는다. 식 (1)과 같은 조건부 최적화 문제는 KKT 조건을 사용하여 푼다. KKT 조건의 만족 여부를 확인하기 위한 식을 간단히 나타내기 위하여, α_i 와 y_i 의 값의 범위에 따라서 $I_0 = \{i : 0 < \alpha_i < C\}$; $I_1 = \{i : \alpha_i = 0, y_i = 1\}$; $I_2 = \{i : \alpha_i = C, y_i = -1\}$; $I_3 = \{i : \alpha_i = C, y_i = 1\}$; $I_4 = \{i : \alpha_i = 0, y_i = -1\}$;과 같이 집합을 정의한다. 여기서 C 는 사용자 설정 상수이다.

MSMO 알고리즘은 크게 세 단계로 학습을 수행한다. 첫 번째 단계에서는 KKT 조건을 만족하지 않는 (최적성 조건을 위반하는) 두 개의 라그랑주 승수를 선택한다. 첫 번째 라그랑주 승수는 식 (4)를 만족해야 하고, 두 번째 라그랑주 승수는 첫 번째 선택된 라그랑주 승수에 의해 정해지는 파라미터 값에 따른 조건식 (5)를 만족해야 한다.

$$i \in I_0 \cup I_1 \cup I_2, \text{ and } F_i < b_{low} - \tau \quad (4)$$

$$i \in I_0 \cup I_3 \cup I_4, \text{ and } F_i > b_{up} + \tau$$

$$i1 = i_{up}, \text{ if } b_{low} - F_{i2} > F_{i2} - b_{up} \quad (5)$$

$$i1 = i_{low}, \text{ if } b_{low} - F_{i2} < F_{i2} - b_{up}$$

여기서, F_i 는 i 번째 샘플의 예측값과 실제값의 차이인 예측 오류를 의미하고, b_{up} 은 $I_0 \cup I_1 \cup I_2$ 집합에 속하는 모든 샘플의 예측 오류 중 가장 작은 값을 의미하고, i_{up} 은 b_{up} 값을 가지는 샘플의 인덱스를 의미한다. b_{low} 는 $I_0 \cup I_3 \cup I_4$ 집합에 속하는 모든 샘플의 예측 오류 중 가장 큰 값을 의미하고 i_{low} 는 b_{low} 값을 가지는 샘플의 인덱스를 의미한다. τ 는 사용자 설정 상수이다. $i2$ 는 식 (4)를 통해 선택된 라그랑주 승수의 인덱스이고, $i1$ 은 식 (5)를 통해 선택된 라그랑주 승수의 인덱스이다.

두 번째 단계에서는 선택된 두 개의 라그랑주 승수를 업데이트한다. 먼저 식 (6)을 통해 $i2$ 번째 라그랑주 승수를 업데이트한다.

$$\alpha_{i2}^{new} = \alpha_{i2}^{old} + \frac{y_{i2}(F_{i2}^{old} - F_{i1}^{old})}{\eta} \quad (6)$$

여기서, η 는 $2K(X_{i1}, X_{i2}) - K(X_{i1}, X_{i1}) - K(X_{i2}, X_{i2})$ 의 계산 결과를 의미한다. 업데이트 된 α_{i2} 는 식 (7)에 따라 잘라낸다.

$$\alpha_{i2}^{\text{new,clipped}} = \begin{cases} L, & \alpha_{i2}^{\text{new}} \leq L \\ \alpha_{i2}^{\text{new}}, & L \leq \alpha_{i2}^{\text{new}} \leq H \\ H, & \alpha_{i2}^{\text{new}} \geq H \end{cases} \quad (7)$$

L 과 H 는 라그랑주 승수 업데이트에 사용하는 임계값으로 식 (8)과 식 (9)를 통해 구한다.

$$L = \begin{cases} \max(0, \alpha_{i2}^{\text{old}} - \alpha_{i1}^{\text{old}}), & s = -1 \\ \max(0, \alpha_{i2}^{\text{old}} + \alpha_{i1}^{\text{old}} - C), & s = 1 \end{cases} \quad (8)$$

$$H = \begin{cases} \min(C, C + \alpha_{i2}^{\text{old}} - \alpha_{i1}^{\text{old}}), & s = -1 \\ \min(C, \alpha_{i2}^{\text{old}} + \alpha_{i1}^{\text{old}}), & s = 1 \end{cases} \quad (9)$$

이때, s 는 $s = y_{i1}y_{i2}$ 를 의미한다. 최종적으로 업데이트 된 α_{i2} 를 사용하여 식 (10)을 통해 라그랑주 승수 α_{i1} 을 업데이트한다.

$$\alpha_{i1}^{\text{new}} = \alpha_{i1}^{\text{old}} - s(\alpha_{i2}^{\text{new,clipped}} - \alpha_{i2}^{\text{old}}) \quad (10)$$

마지막 세 번째 단계에서는 업데이트 된 라그랑주 승수를 바탕으로 예측 오류 F 와 $b_{low}, b_{up}, i_{low}, i_{up}$ 을 업데이트 한다. 예측 오류는 식 (11)을 통해 계산한다.

$$F_i^{\text{new}} = F_i^{\text{old}} + y_{i1}\Delta\alpha_{i1}K(X_{i1}, X_i) + y_{i2}\Delta\alpha_{i2}K(X_{i2}, X_i) \quad (11)$$

$\Delta\alpha_{i1}$ 과 $\Delta\alpha_{i2}$ 는 각각 업데이트 전의 라그랑주 승수와 업데이트 후의 라그랑주 승수의 차이를 의미한다. MSMO 알고리즘은 학습에 사용된 모든 샘플이 최적성 조건을 만족할 경우, 학습을 종료한다.

3. UCI gas dataset 기반 성능 평가

제안된 SVM의 성능 평가를 위해 UCI gas dataset을 활용하였다[17-18]. UCI gas dataset은 6종류의 가스로 구성되어 있으며, 16개의 gas sensor array에서 36개월간 얻은 총 13,910개의 샘플이다. 샘플은 10개의 batch로 나누어져 있고, 각 batch는 시간 순으로 6종류의 가스가 균일한 분포로 구성되어 있다. 가스의 종류는 암모니아, 아세트알데히드, 아세톤, 에틸렌, 에탄올, 톨루엔이며, 각각의 가스는 다양한 농도로 수집되었다. 그림 2는 UCI gas dataset을 사용하여 제안된 SVM의 성능 평가를 수행한 결과이다. 성능 평가를 위해 각 batch별로 3-fold 교차검증 방법을 사용하였다. 평가 결과, 평균 99.38%의 분류 정확도를 보이는 것을 확인하였다. 분류 정확도는 평가를 위해 사용된 총 샘플 수에서 총 예측 오차를 뺀 후, 그 값을 총 샘플 수로 나눈 값을 의미한다.

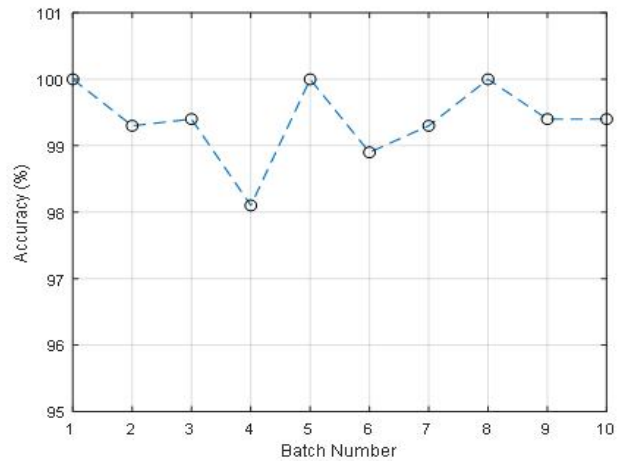


Fig. 2. Performance evaluation of the proposed SVM.

그림 2. 제안된 SVM의 성능 평가

III. SVM 하드웨어 구조

그림 3과 같이 제안된 SVM 기반 가스 분류기는 크게 학습부(training part), 예측부(classification part), 공유구조(sharing block), 총 세 부분으로 나누어져 있다. 학습부는 학습에 사용하는 두 개의 라그랑주 승수 중 α_{i2} 를 찾는데 사용하는 인덱스 선택 유닛(index selecting unit), 나머지 α_{i1} 을 찾는데 사용하는 비교 유닛(comparing unit), 라그랑주 승수를 업데이트하는데 사용하는 라그랑주 승수 업데이트 유닛(lagrange multiplier updating unit), 예측 오류 및 나머지 학습 파라미터들을 업데이트하는데 사용하는 학습 파라미터 업데이트 유닛(training parameter updating unit)으로 구성된다. 예측부는 샘플의 클래스를 예측하기 위해 사용되는 결정 유닛(decision unit)으로 구성된다. 공유구조는 학습과

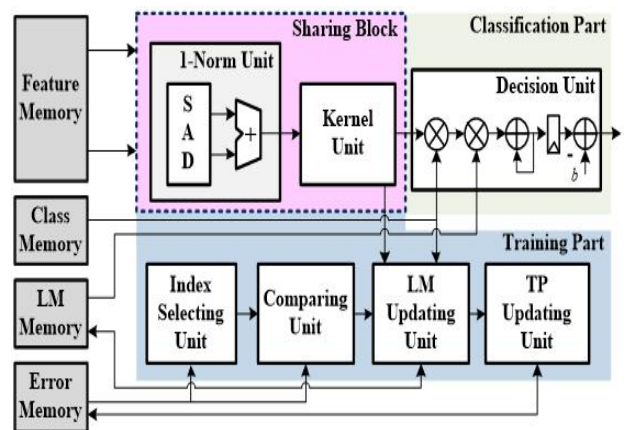


Fig. 3. Block diagram of the proposed SVM gas classifier.

그림 3. 제안된 SVM 가스 분류기의 블록도

예측에 모두 사용되는 커널 연산을 수행하는 1-norm 유닛과 커널 유닛(kernel unit)으로 구성된다. 또한, 메모리는 학습 및 분류에 사용하는 샘플을 저장하는 특징 메모리(feature memory), 샘플의 클래스를 저장하는 클래스 메모리(class memory), 라그랑주 승수를 저장하는 라그랑주 승수 메모리(LM memory), 예측 오류를 저장하는 오류 메모리(error memory)로 구성된다.

그림 4는 커널 연산을 위해 필요한 값인 1-norm을 계산하는 1-Norm 유닛의 구조도를 나타낸다. x_i^k 는 i 번째 샘플의 k 번째 차원의 벡터를 의미한다. 따라서 1-Norm 유닛은 i 번째 샘플과 j 번째 샘플간의 1-norm을 계산할 때, 각 차원에 해당하는 벡터간의 sum of absolute differences (SAD) 연산을 병렬적으로 수행한다. 이후 출력된 1-norm 값은 Kernel Unit으로 입력되어 최종 커널 연산값을 출력한다.

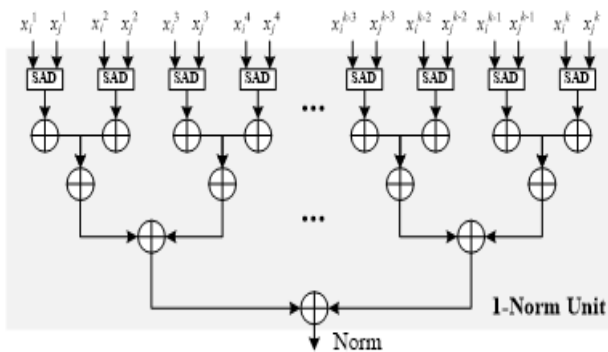


Fig. 4. Block diagram of the 1-Norm unit.
그림 4. 1-Norm 유닛 구조도

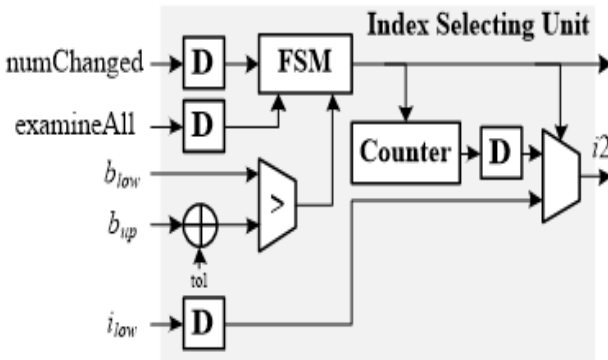


Fig. 5. Block diagram of the index selecting unit.
그림 5. 인덱스 선택 유닛 구조도

그림 5는 인덱스 선택 유닛의 구조도를 나타낸다. numChanged를 통해 라그랑주 승수 업데이트 횟수

를 확인하고, examineAll을 통해 모든 샘플에 대한 최적성 조건 확인 여부를 검사한다. 모든 샘플이 최적성 조건을 만족하면 인덱스 선택 유닛의 FSM에서 종료 신호를 출력하여 학습을 완료한다.

그림 6과 같이 설계된 비교 유닛은 α_{i1} 을 선택하는데 사용한다. 만약, 식 (5)를 만족하지 않을 경우, 인덱스 선택 유닛에서 새로운 α_{i2} 를 찾는다.

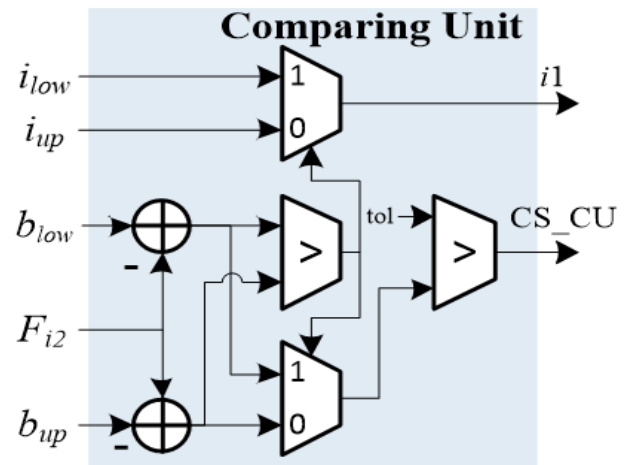


Fig. 6. Block diagram of the comparing unit.
그림 6. 비교 유닛 구조도

그림 7은 라그랑주 승수 업데이트 유닛의 구조도를 나타낸다. Interrupt 신호인 CS_LM은 학습 과정에서 선택된 두 라그랑주 승수의 인덱스 $i1$ 과 $i2$ 가 같을 경우, L 과 H 의 값이 같을 경우, 업데이트된 $\alpha_{i2}^{new, dipped}$ 값과 업데이트 전의 α_{i2}^{old} 값의 차이가 임계값 τ 이하일 경우, 총 세 가지 중 하나라도 만족할 경우에 1이 되고, 인덱스 선택 유닛으로 돌아가

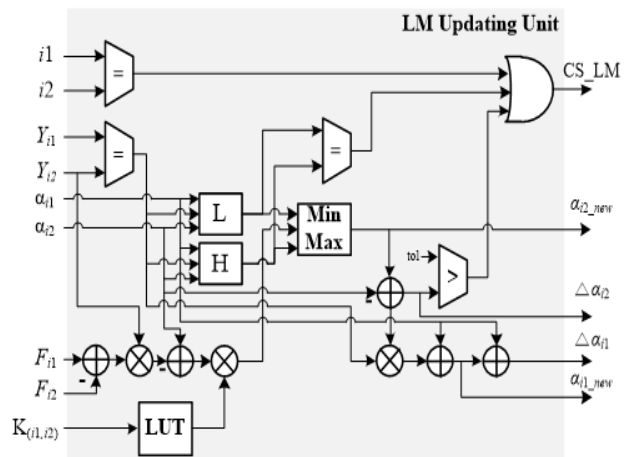


Fig. 7. Block diagram of the LM updating unit.
그림 7. 라그랑주 승수 업데이트 유닛 구조도

MSMO 알고리즘의 첫 번째 단계를 수행한다. LUT는 α_{i2} 를 업데이트할 때 사용하는 $\frac{1}{\eta}$ 값을 look-up table로 구현한 것이다.

그림 8은 학습 파라미터 업데이트 유닛의 구조도를 나타낸다. 비교기를 사용하여 모든 라그랑주 승수의 값을 확인하고, 라그랑주 승수 값이 0과 C 사이의 값을 가질 경우에만 그에 해당하는 예측 오류를 업데이트한다.

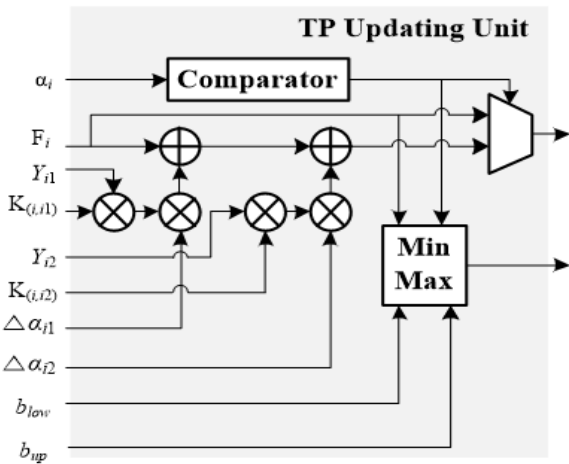


Fig. 8. Block diagram of the TP updating unit. 그림 8. 학습 파라미터 업데이트 유닛 구조도

IV. FPGA 기반 구현 결과

제안된 SVM 기반 가스 분류기의 실시간 학습 가능 여부 및 성능 검증을 위해 그림 9와 같이 Xilinx Zynq UltraScale+ FPGA 기반 검증 환경을

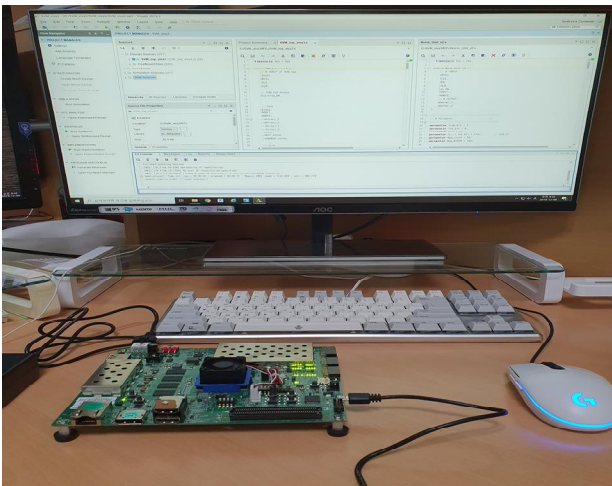


Fig. 9. FPGA test platform. 그림 9. FPGA 기반 검증 플랫폼

구축한 뒤, 구현 및 실시간 검증을 수행하였다. 구현 결과, 표 1과 같이 총 3,337개의 CLB(configurable logic block) LUTs(look-up table), 219개의 CLB Registers로 구현 가능함을 확인하였고, 성능 검증 결과 108MHz의 최대 동작 주파수로 146.67Mbps의 학습 속도를 보이는 것을 확인하였다. 이를 통해 제안된 가스 분류기의 실시간 학습이 가능함을 확인하였다.

Table 1. Implementation results of the proposed gas classifier with Zynq UltraScale+ FPGA.

표 1. Zynq UltraScale+ FPGA 기반 가스 분류기 구현 결과

Modules	CLB LUTs	CLB Registers
Sharing Block	2,581	12
IS Unit	27	19
Comparing Unit	26	20
LM Updating Unit	134	10
TP Updating Unit	313	40
Total	3,337	219

기존 논문[16]과의 하드웨어 면적 비교를 위하여 제안된 가스 분류기는 Altera cyclone II FPGA 기반 구현되었고, 표 2에 비교 결과를 제시하였다. 구현 결과, 제안된 가스 분류기는 총 4,478개의 logic elements, 33개의 embedded multiplier로 구현 가능함을 확인하였다. 기존 논문[16]은 총 6,842개의 logic elements, 41개의 embedded multiplier로 SVM 기반 하드웨어를 구현하였다. 이를 통해, 제안된 가스 분류기는 [16] 논문의 전체 logic elements 대비 35% 감소한 결과를 가지는 것을 확인하였다.

Table 2. Implementation results of the proposed gas classifier with cyclone II FPGA.

표 2. Cyclone II FPGA 기반 가스 분류기 구현 결과

FPGA Resources	[16]	This work	Reduction
Logic Elements	6,842	4,478	35%
Embedded Multiplier	41	33	20%
Clock Frequency	50MHz	62MHz	-

V. 결론

본 논문에서는 가스 분류를 위한 SVM 기반 가

스 분류기를 제안하고, 이의 하드웨어 구조 설계 결과를 제시하였다. UCI gas dataset을 사용하여 제안된 가스 분류기의 성능 평가 결과, 99.4%의 분류 정확도를 가지는 것을 확인하였다. 이의 실시간 학습 및 검증을 위한 하드웨어 구조 설계를 진행하였으며, FPGA 기반 구현 결과, 3,337개의 CLB LUTs로 구현 가능함을 확인하였고, 108MHz의 동작 주파수에서 실시간 동작 가능성이 확인되었다.

References

- [1] Muhammad Hassan, Muhammad Umar, and Amine Bermak, "Computationally efficient weighted binary decision codes for gas identification with array of gas sensors," *IEEE Sensors Journal*, vol.17, no.2, pp.487-497, 2017. DOI: 10.1109/JSEN.2016.2631476
- [2] Woojae Jeong, Minwoo Kim, Jaechan Cho, Yunho Jung, "Design of gas classifier based on artificial neural network," *Journal of IKEEE*, vol.22, no.3, pp.700-705, 2018. DOI: 10.7471/ikeee.2018.22.3.700
- [3] Jingli Yang, Zhen Sun, and Yinsheng Chen, "Fault detection using the clustering-kNN rule for gas sensor arrays," *Sensors*, vol.16, pp.1-21, 2016. DOI: 10.3390/s16122069
- [4] Muhammad Hassan, Amine Bermak, "Gas classification using binary decision tree classifier," *IEEE International Symposium on Circuits and System (ISCAS)*, pp.2579-2582, 2014. DOI: 10.1109/ISCAS.2014.6865700
- [5] Xiaojun Zhai, Amine Ait Si Ali, Abbes Amira, and Faycal Bensaali, "MLP neural network based gas classification system on zynq soc," *IEEE Access*, vol.4, pp.8138-8146, 2016. DOI: 10.1109/ACCESS.2016.2619181
- [6] Kun Wang, Wenbin Ye, Xiaojin Zhao, and Xiaofang Pan, "A support vector machine-based genetic algorithm method for gas classification," *International Conference on Frontiers of Sensors Technologies (ICFST)*, pp.363-366, 2017.
- [7] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Computational Learning Theory*, Pittsburgh, Pennsylvania, USA, pp.144-152, 1992. DOI: 10.1145/130385.130401
- [8] Edgar Elias Osuna, Robert M. Freund, Federico Girosi, "An improved training algorithm for support vector machines," in *Proc. IEEE Signal Processing Society Workshop*, pp.276-285, 1997. DOI: 10.1109/NNSP.1997.622408
- [9] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA, MIT Press, 1998. DOI: 10.1.1.62.2274
- [10] D. DeCoste and B. Schölkopf, "Training invariant support vector machines," *Machine Learn*, vol.46, no.1, pp.161-190, 2002. DOI: 10.1023/A:1012454411458
- [11] S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. Murthy, "A fast iterative nearest point algorithm for support vector machine classifier design," *IEEE Trans. Neural Network*, vol.11, no.1, pp.124-136, 2000. DOI: 10.1109/72.822516
- [12] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, Cambridge, MA, USA, pp.185-208, 1999. DOI: 10.5555/299094.299105
- [13] S. S. Keerthi, C. Bhattacharyya, K. R. K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Computation*, vol.13, no.3, pp.637-649, 2001. DOI: 10.1162/089976601300014493
- [14] Jia-Ching Wang, Li-Xun Lian, Yan-Yu Lin, and Jia-Hao Zhao, "VLSI design for SVM-based speaker verification system," *IEEE Trans. on Very Large Scale Integration (VLSI)*, vol.23, no.7, pp. 1355-1359, 2015. DOI: 10.1109/TVLSI.2014.2335112
- [15] Zehra Camlica, H. R. Tizhoosh, and Farzad Khalvati, "Medical image classification via SVM using LBP features from saliency-based folded data," *International Conf. on Machine Learning and Applications*, pp.128-132, 2015.

DOI: 10.1109/ICMLA.2015.131

[16] Lichen Feng, Zunchao Li, and Yuan Wang, "VLSI design of SVM-based seizure detection system with on-chip learning capability," *IEEE Trans. on Biomedical Circuits and Systems*, vol.12, no.1, pp.171-181, 2018.

DOI: 10.1109/TBCAS.2017.2762721

[17] Alexander Vergara, Shankar Vembu, Tuba Ayhan, Margaret A. Ryan, Margie L. Homer, and Ramon Huerta, "Chemical gas sensor drift compensation using classifier ensembles," *Sensors and Actuators B: Chemical*, vol.166-167, no.20, pp.320-329, 2012.

DOI: 10.1016/j.snb.2012.01.074

[18] Alexander Vergara, "UCI machine learning Repository," <https://archive.ics.uci.edu/ml/datasets/gas+sensor+array+drift+dataset>.

BIOGRAPHY

Woojae Jeong (Student Member)



2018 : BS degree in School of Electronics and Information Engineering, Korea Aerospace University.

2018~present : MS degree course in Department of Electronics and Information Engineering, Korea Aerospace University.

Yunho Jung (Member)



1998 : BS degree in Department of Electrical and Electronic Engineering, Yonsei University.

2000 : MS degree in Department of Electrical and Electronic Engineering, Yonsei University.

2005 : Ph.D degree in Department of Electrical and Electronic Engineering, Yonsei University.

2005~2007 : Senior Engineer, Samsung Electronics.

2007~2008 : Research professor, Institute of Information Engineering, Yonsei University.

2008~present : Professor, School of Electronics and Information Engineering, Korea Aerospace University