다변량 시계열 검사 데이터와 필드클레임 데이터 결합을 통한 엔진 인젝터 문제 유형 분류

이제혁 조성준
jehyuk.lee@dm.snu.ac.kr zoon@snu.ac.kr

Dec 28, 2015

Abstract

데이터마이닝의 다양한 기법들을 반도체 등의 공정에 활용해, 품질을 향상시키려는 노력은 최근 들어 이전보다 더 활발해지고 있다. 이에 활용되는 데이터들의 형태도 다양화되고 있다. 본 연구에서는, 엔진 제조 공정에서 실제로 발생된 다변량 시계열 검사데이터와 이를 실제로 사용하는 차량의 클레임 데이터를 통합하여, 검사 데이터만으로 어떠한 유형의 문제가 발생할지 분류하는 분류기를 만들고자 한다. 여기서 사용한 방법은 DTW-1NN(Dynamic Time Warping with 1 Nearest Neighbor)방법이고, 단변량에서 다변량으로 확장하기 위해, warping distance를 변수별로 합하는 방법을 선택하였다. 그 결과, 모든 변수를 사용하지 않고, 일부의 변수를 사용하였을 때 더 좋은 성능을 보이는 것을 확인하였다. 뿐만 아니라, 문제 유형 자체에 클래스 불균형이 존재함에도 불구하고, 소수 클래스도 잘 분류해 내는 것을 확인할 수 있었다.

Keywords. 다변량 시계열 분류, Dynamic Time Warping, 엔진 문제 유형 분류, 데이터 결합

1 서론

현대 제조업에서는 하나의 제품을 생산하는데 굉장히 많은 프로세스가 필요하다. 이에 따라, 각 프로세스에서 나오는 부품, 생산 등의 데이터양도 매우 급속도로 늘어나고 있다. 이런 데이터들을 통해, 제품 디자인, 제조, 재고관리, 품질관리 등의 각 프로세스의 향상을 이루고자 했던 노력도 꾸준히 있어왔다. 그러나, 데이터 양에 비해서, 활용되는 방법이 충분히 발전하지 못했기 때문에, 많은 유용한 정보를 발견하지 못해왔다. 즉, “rich data but poor information”의 현상이 일어난 것이다. [1] 이러한 다양한 원인들 중 하나로, 데이터의 통합이 제대로 이루어지지 않은 것을 들 수 있다.

제품의 복잡도가 높아지면서, 한 회사에서 할 수 있는 역할은 점점 줄어들어가고 있다. 이에 따라, 한 제품을 생산하는데, 점점 더 많은 수의 회사가 참여하게 되고, 이들은 각자만의 관리 체계를 갖추고 있다. 심지어, 생산에 필요한 부품 관리체계가 ERP(Enterprise Resource Planning)등으로

관리 되지 않는 회사도 많이 있어, 생산 이력 자체를 정확히 찾아내는 것도 쉽지 않은 실정이다. 또한, 대부분의 회사는 자신들의 제품 생산 이력 데이터를 외부에 유출하는 것을 꺼리기 때문에, 데이터에 대한 특별한 협약이 있지 않는한, 이를 얻기는 쉽지 않다. 이러한 상황에서, 데이터 통합을 통한 유용한 정보의 추출을 언급하는 것은 아직까지도 요원해 보인다.

그럼에도 불구하고, 데이터마이닝의 다양한 기법들을 반도체 등의 공정에 활용해, 품질을 향상시키려는 노력은 최근 들어 이전보다 더 활발해지고 있다. [2, 3] 이에 활용되는 데이터들의 형태도, 정적인 수치 데이터에서 시작하여 문자, 이미지 등의 범위로도 확장되어 가고 있는 추세이다. 비록, 대부분의 경우는 단일 단계에서 생산되는 중간단계의 부품이나, 완제품 단일 단계에서만 수행되는 것이지만, 단일 단계에서만의 프로세스 향상이 전체적인 공정 프로세스상의 향상에 영향을 끼치는 것은 놀랄 일이 아니다.

그러나, 이러한 데이터 통합의 어려움을 넘어설 수 있다면, 전체적인 프로세스상에서 더 큰 향상이 이뤄지게 될 것이다. 본 연구에서 공급 체인의 두 개의 다른 단계에서 데이터를 통합하였을 때, 기존에 발견하지 못했던 문제를 해결할 가능성이 있음을 보일 것이다.

공정에서 생성되는 제품은 다양한 특징을 갖고 있다. 많은 기준을 통해 이 제품들을 구분할 수 있지만, 하나의 중요한 기준이 되는 것은 사용자가 제품을 사용할 때, 제품의 특성의 변화 유무이다. 대부분의 제품의 경우는 사용자의 제품 사용 유무와 상관없이, 거의 일정한 특성을 갖고 있게 된다. 하지만, 엔진 혹은 자동차의 경우, 사용자가 제품을 사용할 때, 속도, 출력, 온도, 배기압 등 제품에서 나오는 모든 특성 값이 변하게 된다. 이러한 제품의 경우, 공정 단계에서 품질관리를 하기 위해서는 이 변하는 특성 값을 관측하면서, 제품의 정상 여부를 검사해야 한다. 즉, 시계열 형태를 갖게 되는 것이다. 특히, 이러한 공정관리를 위한 데이터는 여러 개의 특성들이 시간에 따라 어떻게 변하는지를 관측해야 하므로, 다변량 시계열 데이터라는 특성도 갖고 있다. 본 연구에서는, 공정에서 생성되는 다양한 데이터형태 중 하나인 다변량 시계열 검사데이터와 고객의 클레임 데이터를 활용하여, 엔진에 문제가 있다고 판명이 되었을 때, 이 데이터를 활용하여 어떠한 형태의 고장이 났다고 빠르게 진단할 수 있음을 보일 것이다. 이를 통해, 기존 품질 관리 체계에서는 발견해내지 못했던 고장 유형의 파악을 공정단계에서 미리 함으로써, 제품의 출시 전에, 이를 미리 감지할 수 있음을 보일 것이다. 논문은 다음과 같이 구성된다. 2절에서는 기존 연구들에 대해서 소개하고, 3절과 4절에서는 사용한 데이터와 함께 제안하는 방법론에 대해서 설명한다. 이 방법으로 실험을 하였을 때의 결과는 5절에 있으며, 6절에서는 이 연구의 결론을 지으면서, 추후 이 연구가 나아갈 방향을 제시한다.

2 관련연구

2.1 시계열 분류

일반적인 수치 데이터와는 다르게, 시계열 데이터는 연속적으로 데이터 포인트들이 모여서 하나의 데이터를 이루는 특징을 갖고 있다. 이 때문에, 이 데이터를 활용하여 일반적인 데이터마이닝의

문제에 적용시키기 위해서는, 일반적인 방법과는 다른 방식으로 문제에 접근해야 한다. 시계열 데이터의 분류 문제는 크게 두 가지 결정해야 할 문제가 존재한다.

첫째로, 시계열 데이터의 유사성 척도를 나타내는 문제이다. 일반적인 수치형 데이터와 다르게, 시계열 데이터는 연속성이라는 특징이 있다. 그러므로, 두 개의 시계열 데이터를 비교할 때는 전체 시계열을 비교할 것인지, 혹은 전체 시계열에서 중요한 일부분만 볼 것인지 결정해야 한다. 전체 시계열을 비교할 경우, 가장 흔한 방법으로는 Discrete Fourier Transform, 혹은 Discrete Wavelet Transform의 계수간의 Euclidean 거리를 구하여 비교하는 것이다. ([4], [5]) 다른 주요 방법으로, 시계열을 time warping을 시켜서, 시계열 데이터를 'time warping'시켜서 데이터를 비교하는 Dynamic Time Warping이 있다. [6]

둘째로, 시계열 데이터를 어떠한 형식으로 나타내느냐의 문제가 있다. 한개의 시계열 데이터는 몇 개의 데이터 포인트가 연속적으로 나열되어 이루어져 있어서 차원이 클 뿐만 아니라, 일반적으로 시간에 따라서 측정이 되기 때문에, 잡음이 섞여 들어갈 가능성이 크다. 그러므로, 이를 보완해주기 위해서 다양한 형태의 표현 방법이 있다. 가장 간단하게는 샘플링 방법부터 [7], 시계열을 segmentation하고 그 segment의 평균값으로 대표값을 나타내어, 압축된 시계열로 표현하는 방법인 Piecewise Aggregate Approximation(PAA) [8], 시간 도메인이 아닌 다른 도메인의 형태로 바꿔주는 Discrete Fourier Transform [4], Discrete Wavelet Transform [9]도 있다. 본 연구에서 쓰인 SAX도 시계열 표현 방법 중 하나로, 시계열 데이터를 수치형 데이터에서 부호 형태의 데이터로 변환시켜서 표현하는 방법이다 [10].

이러한 두 개의 큰 문제점이 있음에도 불구하고, 시계열 데이터를 대상으로 한 분류는 많이 연구되어왔다. 시계열을 Wavelet변형한 후, 그 계수로 비교하는 방법 [11], DTW를 이용한 분류하는 방법 [12], SVM을 이용하여 분류하는 방법 [13], Shapelet을 학습하여 분류하는 방법 [14]등 여러 가지 방법으로 연구가 진행되어왔다. 그 결과, 단변량 시계열 분류문제에서는 시계열 유사 척도를 DTW로 사용하고, 분류기를 1-nearest neighbor로 하는 DTW-1NN방법은 많은 경우에 좋은 성능을 보인다고 알려져 있다. [15]

2.2 Dynamic Time Warping(DTW)

시계열 데이터간의 유사성을 나타내기 위해서는 정적인 데이터에서 흔하게 쓰이는 유클리디안 거리는 실제 시계열 간의 유사성을 제대로 측정하지 못하는데, 그 이유는 시계열 데이터가 갖고 있는 특성 때문이다. 그림 1에서 두 개의 시계열 데이터는 매우 유사하다. 그러나, 두 개의 데이터는 일정 시간 간격을 두고 비슷한 패턴이 등장한다는 점에서 다르다. 만약 이를 무시하고, 그림 1의 왼쪽과 같이 단순히 시간대별로 유클리디안 거리를 측정하면, 두 데이터는 분명히 패턴이 비슷하게 나타나 비슷한 데이터임에도 불구하고, 그들간의 유사성은 매우 적다고 결론짓게 된다. 이러한 오류를 보정하기 위해서, 그림 1의 오른쪽과 같이 시간간격이 있는 것을 고려하여 데이터 포인트들을 매칭시켜야 한다. 이를 시계열 데이터를 time warping한다고 한다. 이때, 이 warping 거리를 기준으로 해야 유클리디안 거리보다 더 정확한 유사성 척도가 되는 것이다.

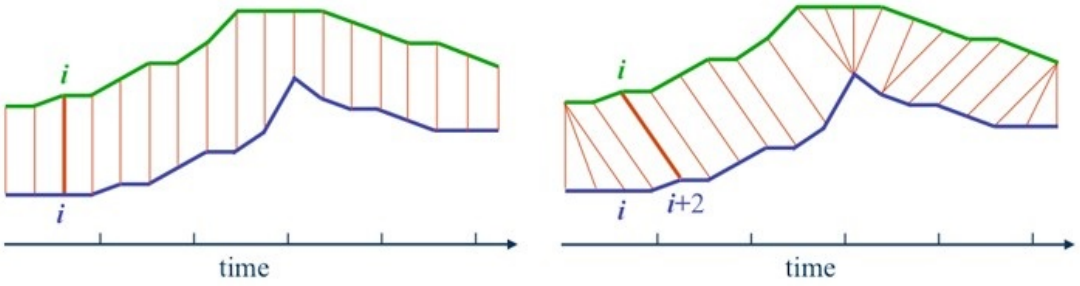

그림 1: 시계열 데이터 유사성 척도 (왼쪽): 유클리디안, (오른쪽): DTW

두 개의 길이가 각각 l, m인 시계열 데이터 $A, B(A = a_1, a_2, ..., a_p, B = b_1, b_2, ..., b_m)$가 있다고 하자. 이 때, 공통된 warping path를 만들기 위하여, $p - by - m$의 행렬을 만든다. 이 행렬의 (i, j) 번째 원소는 A의 i번째 원소와, B의 j번째 원소간의 거리를 나타낸다.$(d(A_i, B_j) = (a_i - b_j)^2)$ 여기서 Warping path W는 A와 B의 인접성을 나타내는 mapping된 path이다. 즉, Warping path는 $W = w_1, w_2, ..., .w_k, ..., w_K, w_k = (i, j)_k, max(m, p) \leq K \leq m + p - 1$이다. 이 때, W는 몇 가지 중요 조건들을 만족시켜야 한다.

- Boundary Condition: $w_1 = (1, 1), w_K = (p, m)$
- Continuity: $w_k = (a, b)$일 때, $w_(k - 1) = (a', b')$, where $a - a' \leq 1$ and $b - b' \leq 1$
- Monotonicity: $w_k = (a, b)$일 때, $w_(k - 1) = (a', b')$, where $a - a' \geq 0$ and $b - b' \geq 0$

그리고, 이를 만족하는 수많은 path들 중, warping cost $(DTW(A, B) = min\{\sum_{k=1}^{K} w_k\})$를 최소화 하는 path가 중요하기 때문에, 모든 path를 다 구하는게 아니고, warping cost를 최소화 하는 path를 동적 프로그래밍으로 구하는 방법을 제안하였다.(Berndt and Clifford(1996)) 현재 (i, j) 번째 원소에 있다고 하고, 현재까지의 누적 거리가 $\gamma(i, j)$이고, 현재 $d(i, j)$의 거리를 구했을 때, 인접 원소의 누적 거리의 최소값은 다음 식을 추가하여 구할 수 있다. 그리고 동적 프로그래밍으로 실제 warping path를 구하는 예시는 다음 그림 2과 같다.

$$\gamma(i, j) = d(a_i, b_i) + min\{\gamma(i - 1, j - 1), \gamma(i - 1, j), \gamma(i - 1, j)\}$$

이 때, DTW의 복잡도는 $O(pm)$이다. 여기서, 두개의 시계열 데이터 사이의 유클리디안 거리는 warping path W의 k번째 원소인 $w_k = (i, j)_k, i = j = k$로 제한되고, 두 시계열의 길이가 같을 때 구할 수 있다.

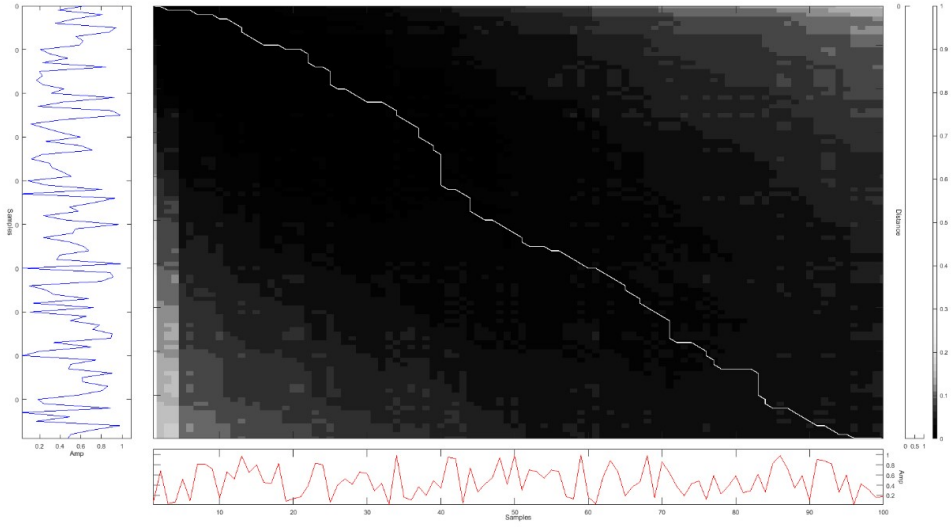

그림 2: DTW를 구하는 과정

2.3 k-최근접 이웃 분류 알고리즘(k-NN classification)

k-nearest neighbor알고리즘은 분류 혹은 회귀 문제에서 쓰이는 간단한 비모수적인 방법이다 [16]. 분류문제에서는, 어떤 개체에서 가장 가까운 k개의 이웃들을 통해, 해당 개체의 클래스를 분류하게 된다. 이 때, 해당 개체의 클래스는 k개 이웃 개체의 클래스들의 다수결 투표로 결정되게 된다. 그 중, 특별한 경우로 k를 1로 설정하는 경우가 있는데, 이는 1-최근접 이웃 방법(1-NN)으로, 해당 개체의 클래스는 가장 가까운 이웃의 클래스를 갖게 된다. 이 k-최근접 이웃 방법은 매우 간단한 반면, 몇 가지의 단점이 존재한다. 첫째로, 최적화된 k값이 정해지지 않았다는 것이다. 이는 데이터의 형태에 매우 의존적인 성질 때문이며, k값에 따라 불필요한 항목에 대한 민감도와 클래스간의 경계 사이에 trade-off가 생기게 된다. 또한, 이 알고리즘의 정확성은 데이터 분포에 따라 매우 차이가 난다는 단점이 있다. 만약 데이터가 한쪽으로 편향되어 있을 경우, 각 개체의 k 인접 이웃의 대부분은 그 편향된 데이터에서 나올 가능성이 있기 때문이다 [17]. 이를 막기 위하여, 거리에 반비례하는 가중치를 두어서 다수결 투표로 구하는 방법을 사용하기도 한다 [18]. 그리고, k-최근접 이웃 분류 방법은 고차원에서는 '차원의 저주'에 매우 민감하다. 이를 피하기 위해, 일반적으로 분류 전에 고차원에서 저차원으로 투영시킨 후에 분류를 시행하는 것이 일반적이고, 이를 저차원 매장이라 한다 [19]. Wrapper Approach같은 변수 선택법을 이용하거나 [20], 주성분 분석(Principal Component Analysis, PCA)나 선형 판별 분석(Linear Discriminant Analysis) [21]같은 변수 추출법이 있다.

2.4 Symbolic aggregate approximation(SAX)

SAX는 단변량 시계열 데이터에 대해서, 이 전체 데이터 포인트들의 분포를 Gaussian distributed를 가정하고, 이를 그룹화하는 알고리즘이다 [10]. 이 알고리즘은 DFT(Discrete Fourier

Transform), DWT(Discrete Wavelet Transform)등과 같은 시계열 데이터의 표현 방법이며, 시계열의 가장 큰 문제점 중의 하나인 차원의 축소를 이룸과 동시에, lower bounding 거리 척도를 만족하는 indexing방법이다.

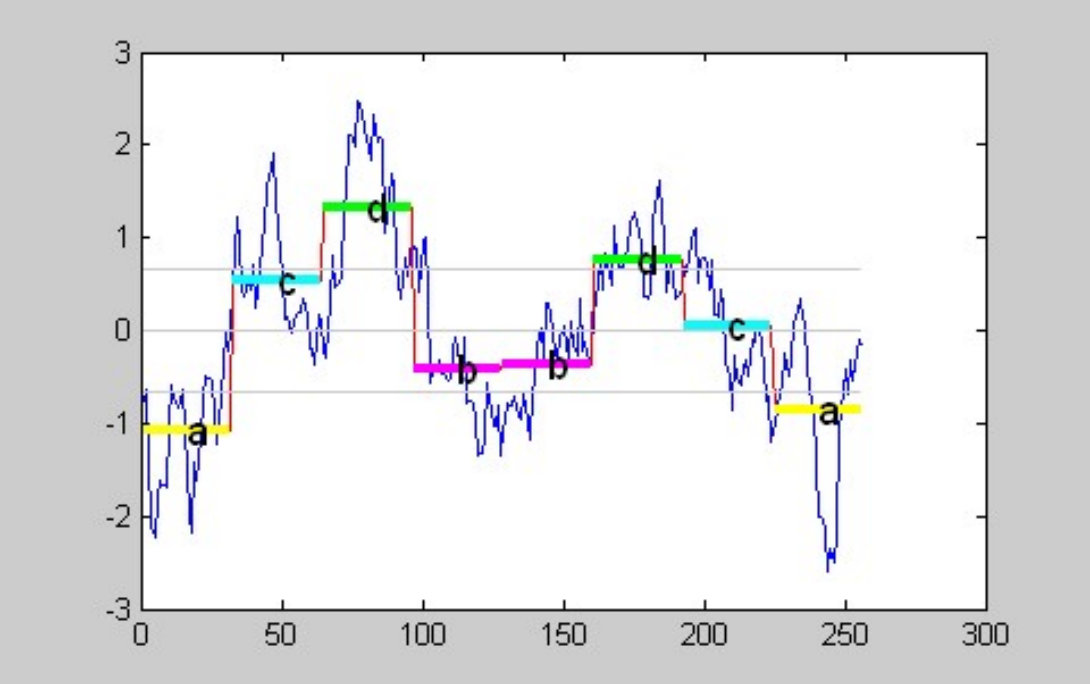

그림 3: Symbolic aggregate approximation

3 실험과정

본 연구는 DTW-1NN classifier의 실제 품질관리의 향상에 도움이 될 수 있는지 파악하기 위한 연구로, 분류문제에만 초점을 둘 것이다. 또한, 문제의 대부분을 차지하고 있던 부품인 인젝터의 고장과 다른 원인으로 인한 고장을 잘 분류할 수 있는지 확인할 것이다. 이를 위해서, 고장의 클래스를 인젝터의 고장과 다른 모든 고장 케이스를 모아 놓은 2가지의 클래스로 나눈다. 분류에 쓰이는 항목들은 다음과 같이 나눈다: (1) 모든 변수를 사용, (2) 각 스텝마다 일정 변수 집합만을 사용, (3) 각 스텝별로 모든 변수를 사용한다. 여기서, 스텝이라는 것은 실제 공정데이터에서 나온 특징으로, 다음 섹션에서 설명하도록 하겠다. 실험 (2)의 경우에는 고르는 변수 갯수와 종류는 임의로 결정하여, 가장 성능이 좋은 경우를 선택한다. 실험(3)의 경우에는 스텝별로 측정한 뒤에, 가장 성능이 좋은 스텝의 경우를 선택한다. 각 변수들의 스케일이 서로 다른 것을 보정해 주기 위해, z-정규화를 수행하였다.

분류는 DTW-1NN 분류기를 사용하며, 단변량에서 다변량으로 확장하기 위해서, 각 변수별로 warp된 거리를 계산한 후, 이들을 모두 합친 것을 두 다변량 시계열 데이터의 유사성 척도로 사용한다. 이 때, 데이터 수 자체가 얼마 없는 관계로, Leave-one-out Cross Validation을 통한 validation

오분류율을 계산하여 분류기의 성능을 측정하였다.

4 데이터

본 연구에서는 크게 두 개의 데이터를 활용한다. 첫째는, 엔진 제조 과정에서 생성된 엔진 검사 데이터이다. 이 데이터는 실제로 엔진에 연료를 주입하고, 매 시간 마다 엔진의 속도를 변화시켜가면서, 엔진에서 발생하는 배기압, 온도, 토크 등 39개의 측정 항목에 대한 값을 수집한다. 그림 4는 그 실제 예를 보여준다.

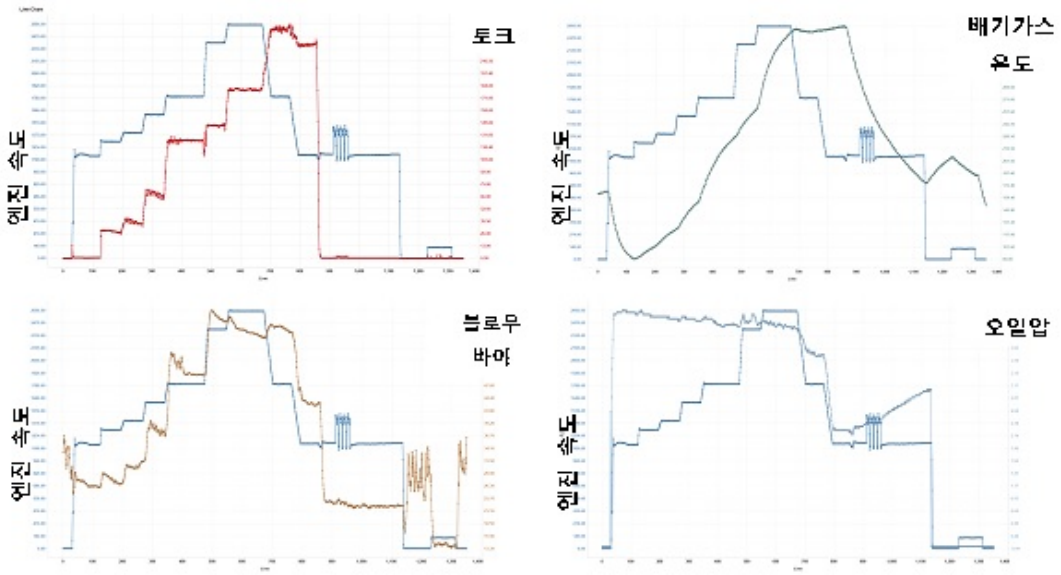

그림 4: 여러 스텝으로 나뉘어진 실제 검사 데이터

현재는, 아직 이 데이터를 통해 품질관리에 활용하지 못하고 있다. 이 과정에서 Data의 정합성을 확인한 결과, 총 9,925개의 엔진이 정상적으로 측정되었던 기록과, 32개의 측정 항목이 정상적으로 측정되었음을 확인할 수 있었다. 이 데이터에서는 실제로 운행할 때, 엔진의 속력을 일정시간 동안 유지한다. 다음 그림에서 일정하게 유지되는 구간 있는데, 이를 '스텝'이라고 명명한다. 이 검사 데이터는 총 6개의 스텝으로 구분할 수 있고, 이는 실제 운행시에 발생할 수 있는 특정 상황들을 검사하기 위함이다. 제조사에서는 각 스텝에서 변수들의 변화 행태가 중요할 것으로 생각하고 있으나, 아직 이에 대한 활용방안은 아주 미약한 수준이다. 각 스텝에 대한 정보는 검사데이터에 포함되어 있지 않았으며, 이는 속도의 범위에 따라 segmentation 작업을 진행하였다. 이 때, 사용한 알고리즘이 symbolic aggregate approximation (SAX)이다. 이 알고리즘으로 속도가 일정 범위 안에 있는 데이터 포인트들을 하나로 묶을 수 있게 스텝을 나눈 것이다.

데이터의 전처리과정은 다음과 같다. 먼저, 이 두 개의 데이터는 서로 다른 회사에서 관리되고 있었기 때문에, 전처리 과정을 통해 이 둘의 데이터를 각 회사에서 관리하고 있던 고유의 엔진번호를 하나로 통합하는 과정이 필요했다. 이 과정을 통해, 고장난 엔진 이력과 이들의 검사 내역을

통합할 수 있었다. 그 다음으로는, 스텝에 대한 정보를 추출하였다. 앞에서 언급한 엔진의 스텝은 현업에서 임시로 쓰이고 있었던 용어일 뿐, 실제로 검사데이터에는 이에 대한 정보가 없었기 때문에, 앞에서 언급한 바와 같이, SAX기법을 사용하여 엔진의 속도가 어느 정도 범위안에서 일정하게 유지되는 경우를 하나의 스텝으로 분리했다. 그리고, 데이터의 정합성을 검사하여, 실제로 제대로 측정이 되고 있지 않은 항목들을 제거하고, 스텝이 제대로 분리되어 있지 않은 개체들을 제거하는 과정을 수행했다.

5 실험결과

실험과정 섹션에서 정의한 세가지 경우에 따른 실험결과는 다음 표1과 같이 나온다.

표 1: 각 실험 별 오분류율

실험방법	실험 (1)	실험 (2)	실험 (3)
Validation Error	0.2619	0.1667	0.2857

전체적으로 오분류율이 낮게 나오고 있고, 그 중에서도 변수선택을 할 때, 오분류율이 더 낮아지는 현상을 보이고 있다. 스텝 별로 모든 변수 선택을 하였을 때는, 오분류율이 오히려 더 높아지는 현상을 보인다. 그러나, 이 문제의 경우 (Target):(Non-Target) = 2:1에 가까우므로, 약간의 클래스 불균형이 있다. 그래서, 분류결과의 오분류율만 갖고 판단할 것이 아니라, 실제 분류 행렬을 통해, 분류기가 Target과 Non target을 어떻게 분류하는지 알아봐야 한다. 그 결과는 다음 표 2와 같다.

표 2: 분류행렬 결과 - (1)전체 변수 사용, (2)임의 변수 사용, (3)스텝별 변수 사용

	실제	예측(Target)	예측(Non-Target)
실험(1)	Target	25	7
	Non-Target	2	8
실험(2)	Target	24	4
	Non-Target	3	11
실험(3)	Target	20	5
	Non-Target	7	10

모든 변수들을 활용했을 때보다, 스텝별로 나누어서 분류했을 때, 분류기 자체의 성능은 좋지 않지만, 소수 클래스에 대해서는 상대적으로 잘 분류해 내는 것을 알 수 있다. 이를 통해, Target에 해당하지 않는 문제들은 각 중요한 step이 따로 있을 것임을 알 수 있다. 게다가, 실험(2)의 결과를 보았을 때, 적절한 변수선택이 이루어 질 경우, 전체적인 오분류율에 있어서도 더 나은 성능을 보일

뿐만 아니라, 소수 클래스의 데이터도 더 잘 분류해 내는 것을 볼 수 있다. 이를 통해, 변수 선택이 좀 더 정교하게 이루어질 경우, 더 좋은 성능의 분류기를 만들 수 있음을 짐작할 수 있다.

6 결론

본 연구에서는 실제 엔진 제조 과정에서 생성된 다변량 시계열 검사데이터와 엔진이 부착되어 나온 산업차량의 클레임 데이터를 통합해, 기존에는 감지하지 못하고 있던 엔진의 문제 유형을 파악할 수 있는 방법론을 제시하였다. 이를 통해, 기존에는 충분히 활용되고 있지 못했던 검사데이터를 통해, 엔진의 문제 유형을 파악할 수 있게 되었다. 뿐만 아니라, 이 과정에서 정교한 변수선택법을 통해, 엔진 문제 유형 분류기의 성능을 향상시킬 수 있는 가능성을 보였다. 이 변수 선택을 통하여, 목표로 삼았던 중요한 엔진 문제 유형은 검사데이터에서 어떤 변수들을 면밀히 검토해야 하는 가에 대한 insight를 제조사에게 제공할 수 있다. 또한, 이 문제 자체가 클래스 불균형이 있음에도 불구하고, 소수 문제 유형에 대해서도 잘 분류해 내는 것을 볼 수 있었다. 추후, 본 연구에서 존재했던 임의 변수 선택이라는 한계를 극복하기 위해, 기존에 존재하는 변수 선택법을 이 문제에 적용하여, 더 나은 성능을 갖는 분류기를 만들 것이다. 뿐만 아니라, 분류기가 소수 클래스의 분류도 잘 하는 것으로 보아, 분류 문제를 문제 유형별이 아닌, 전체 엔진 중에서 고장 엔진을 탐지하는 방법으로 확장할 수 있을 것이다.

Acknowledgement

This work was supported by the BK21 Plus Program(Center for Sustainable and Innovative Industrial Systems, Dept. of Industrial Engineering, Seoul National University) funded by the Ministry of Education, Korea (No. 21A20130012638), the National Research Foundation(NRF) grant funded by the Korea government(MSIP) (No. 2011-0030814), and the Institute for Industrial Systems Innovation of SNU.

참고문헌

[1] X. Z. Wang and C. McGreavy. Automatic classification for mining process operational data. *Industrial & Engineering Chemistry Research*, 37(6):2215–2222, 1998.

[2] Seokho Kang, Sungzoon Cho, Daewoong An, and Jaeyoung Rim. Using wafer map features to better predict die-level failures in final test. *Semiconductor Manufacturing, IEEE Transactions on*, 28(3):431–437, 2015.

[3] Yongwon Park, Seokho Kang, and Sungzoon Cho. Memory die clustering and matching for optimal voltage window in semiconductor. *Semiconductor Manufacturing, IEEE Transactions on*, 28(2):180–187, 2015.

[4] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient similarity search in sequence databases. In *Foundations of Data Organization and Algorithms*, volume 730 of *Lecture Notes in Computer Science*, pages 69–84. Springer Berlin Heidelberg, 1993.

[5] Kin-Pong Chan and A.W.-C. Fu. Efficient time series matching by wavelets. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 126–133, 1999.

[6] Donald J. Berndt and James Clifford. Advances in knowledge discovery and data mining. chapter Finding Patterns in Time Series: A Dynamic Programming Approach, pages 229–248. American Association for Artificial Intelligence, 1996.

[7] K.J. Åström. On the choice of sampling rates in parametric identification of time series. *Information Sciences*, 1(3):273–278, 1969.

[8] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286, 2000.

[9] ZbigniewR. Struzik and Arno Siebes. Wavelet transform in similarity paradigm. In *Research and Development in Knowledge Discovery and Data Mining*, volume 1394 of *Lecture Notes in Computer Science*, pages 295–309. Springer Berlin Heidelberg, 1998.

[10] Tae Yano, Noah A Smith, and John D Wilkerson. Lin, jessica and keogh, eamonn and lonardi, stefano and chiu, bill. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 2–11. ACM, 2003.

[11] Hui Zhang, TuBao Ho, and MaoSong Lin. A non-parametric wavelet feature extractor for time series classification. In *Advances in Knowledge Discovery and Data Mining*, volume 3056 of *Lecture Notes in Computer Science*, pages 595–603. Springer Berlin Heidelberg, 2004.

[12] Young-Seon Jeong, Myong K. Jeong, and Olufemi A. Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240, 2011.

[13] Damian R. Eads, Daniel Hill, Sean Davis, Simon J. Perkins, Junshui Ma, Reid B. Porter, and James P. Theiler. Genetic algorithms and support vector machines for time series classification. In *Proc. SPIE*, pages 74–85. International Society for Optics and Photonics, 2002.

[14] Lexiang Ye and Eamonn Keogh. Time series shapelets: A new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 947–956. ACM, 2009.

[15] Xiaopeng Xi, Eamonn Keogh, Christian Shelton, Li Wei, and Chotirat Ann Ratanamahatana. Fast time series classification using numerosity reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 1033–1040. ACM, 2006.

[16] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[17] D. Coomans and D.L. Massart. Alternative k-nearest neighbour rules in supervised pattern recognition. *Analytica Chimica Acta*, 136:15–27, 1982.

[18] Micheline Kamber Han, Jiawei and Jian Pei. *Data mining: concepts and techniques: concepts and techniques*. Elsevier, 2011.

[19] Blake Shaw and Tony Jebara. Structure preserving embedding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 937–944. ACM, 2009.

[20] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[21] Aleix M. Martinez and A.C. Kak. Pca versus lda. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(2):228–233, 2001.