데이터 압축 알고리즘을 활용한 엣지 컴퓨터 전송 효율 향상 기법

이민정, 오성빈, 김진호*

경남대학교

goodsaram7@naver.com, ginbeat21@outlook.com, *kimjh@kyungnam.ac.kr

Using data compression algorithm Edge computer transmission efficiency improvement technique

Lee Min Jeong, Oh Sung Vin, Kim Jin Ho*

Kyungnam Univ.

요 약

제조 산업현장에서는 계획되지 않은 다운타임에 대비하여 데이터를 기반으로 한 원격모니터링 및 예지보전 기술이 활발하게 사용되고 있다. 원격모니터링과 예지보전을 위해서는 실시간으로 데이터 수집과 학습을 위한 많은 양의 데이터 축적이 필요하다. 예지보전 및 모니터링을 하기 위해서 센서 데이터들이 서버나 클라이언트로 전송되어야 하는 데 이때 데이터 양이 많기 때문에 네트워크 부하 발생 또는 서버의 저장 용량이 많이 요구된다. 실시간성 확보가 가능한 EDR 압축기술 기반으로 제조 데이터를 실시간으로 압축하여 전송하면 네트워크 부하 및 서버 저장 공간을 줄 일 수 있는 기술을 연구하였다.

Ⅰ. 서 론

스마트팩토리에서 예지보전은 설비 이상에 대한 사전 진단이 가능하고 설비와 부품의 수명을 예측하여 최적의 설비상태를 지속하도록 돕는다. 이를 통해 품질 향상과 납기를 준수하여 제조산업의 경쟁력을 강화할 수 있다. 또한, 통계와 분석을 토대로 객관적인 근거에서 공장의 다운타임을 줄여 실질적인 비용손실을 줄여준다.[1]

제조 현장에서는 계획되지 않은 다운타임에 대비하여 원격모니터링과 예지보전 도입이 늘고 있다. 생산 공정이 다운될 경우 엄청난 손실이 발생하기 때문이다. 원격모니터링과 예지보전을 위해서는 실시간으로 데이터가 수집되어야 하고 많이 양의 데이터 축적이 필요하다. 이러한 데이터는 필요에 따라 또는 실시간으로 서버나 클라이언트 측으로 전송되어야 한다. 많은 데이터양에 의한 네트워크 부하, 큰 저장 공간이 요구되는 단점이 따른다. 이를 해결 하는 방안으로 실시간성 확보가 가능하고 전송 데이터프레임의 크기도 줄일 수 있는 EDR 압축기술을 제안한다.

EDR 압축기술은 다른 압축기술들[3, 5, 6]에 비해 단순해서 데이터 전송 시간을 줄여줄 수 있다. 자동차 분야에서 차량 내 네트워크 사용량을 줄이기 위해 연구되고 있는 분야[2, 4]로써 자동차의 안전과 실시간을 모두 가능하게 해준다. 이를 데이터를 기반으로 하는 제조산업에 적용, 활용할 수 있다.

Ⅱ. 본론

2.1 EDR(Enhanced Data Reduction) 알고리즘 분석

EDR 알고리즘에는 신호 SDN(Signal, Data, no-change), SN(Signal, no-change)이 있다. SDN 신호는 신호 전체, 신호 값 차이, 변경 없음으로 표현되며 SDN(Signal, Data, no-change)은 5bit 이상의 길이를 가지는 신호로 주기적으로 보낸다. 차량 속도, 엔지 온도 등이 있다. SN(Signal, no-change)은 변경 없음 혹은 신호 전체로 표현, 4bit 이하의 길이를 가지며 기어 상태, 엔진 상태 등과 같은 상태 신호이며 SN 신호들을 그룹화하여 처리한다. [2]

EDR 알고리즘에서 DCC는 압축 메시지의 첫 번째 바이트이고 각 bit의 위치는 압축 안 된 원본 메시지의 데이터 바이트 위치이다. DCC의 각 bit 값이 '0'이면 해당 위치의 신호가 압축되었음을 의미하고, '1'이면 해당 위치의 신호가 완전압축 또는 델타 압축을 의미한다. 압축된 SDN 신호는 RT(Reduction Type) bit로 완전압축(RT=1) 또는 델타 압축(RT=0)을 나타낸다.[2]

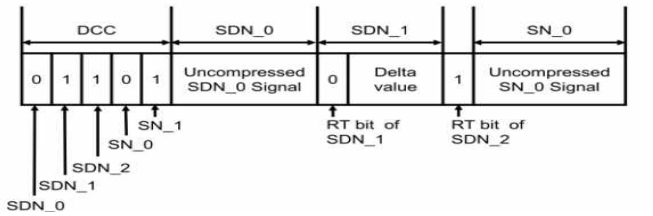

그림1.Example of compressed message using EDR algorithm[2]

2.2 EDR 알고리즘 구현

EDR 알고리즘의 이론을 바탕으로 하여 압축프로그램을 구현하였다. 그림2는 실시간으로 들어오는 3가지의 센서값들의 압축 과정을 보여주고 있다. 이 알고리즘은 이전 데이터와 현재 데이터의 차를 계산하여 그 차이 값만 보내어서 데이터의 크기를 줄이는 방식이다. 첫 번째 센서값은 압축 안한 원본 데이터 그대로 보내고 두 번째 들어온 센서값부터 이전 센서값과 현재 센서값의 차이를 계산하여 전송하는 방식이다. 차이값(Difference)의 저장 공간의 크기(2^n)를 정해 놓고 이 범위 내의 크기를 넘어서는 값이라면 압축 없이 원본 데이터 그대로 보내고 범위 내의 Difference라면 이 값을 전송한다.

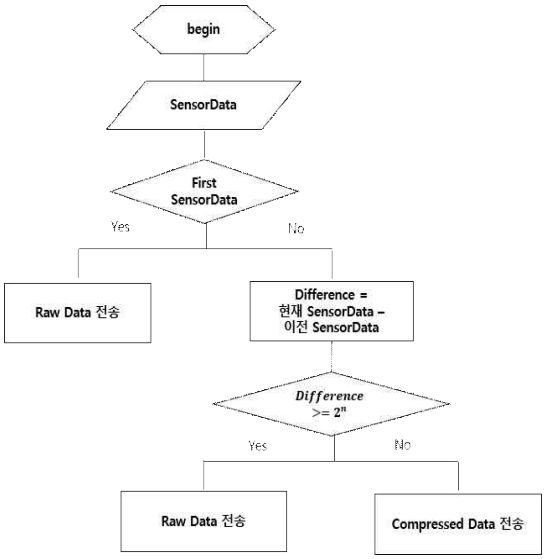

그림 2 Flowchart of data compression algorithm

그림3은 전송되는 데이터프레임이다. CIB는 센서(S1, S2, S3)의 압축유무를 나타내고 SIB는 센서(S1, S2, S3)의 부호를 나타내는 bit이다

그림3의 CIB = '101'의 의미는 S1, S2, S3 센서값의 압축 유무이다. S1은 '1'로 '압축'을 의미하고 표1에서 Raw data size=10bit를 압축을 통해 Compressed data size= 2bit로 줄였음을 알 수 있다. S2은 '0'이므로 '압축 없음'을 의미하므로 표1에서 Raw data size=7bit 그대로 보낸다. S3는 '1'로 '압축'을 의미하므로 Raw data size=12bit를 Compressed data size=8bit로 줄였다. 다음 그림3은 위 과정을 통해 실제 전송되는 데이터 프레임으로 Data Field 길이에 CIB, SIB의 6bit를 합친 길이가 전송된다. 이다. 표2는 하나의 데이터프레임에서의 Data Field 전체 길이다.

CIB			SIB			Data Field		
						S1	S2	S3
1	0	1	0	0	1	2bit	7bit	8bit

그림 3 데이터프레임

	S1	S2	S3
Raw data size	10	7	12
Compressed data size	2	4	8

표 1 Data Storage Size (단위 : bit)

CIB	Data Field Length(단위:bit)
000	29
001	25
010	26
011	22
100	21
101	17
110	18
111	14

표 2 Compression Data Field Length

2.3 실험환경

그림4와 같이 Server PC에는 윈도우 10, Python3.9를 설치했으며 Client(Edge computer)는 Raspberry pi 3에 Raspbian과 Pyhthon3.9를 설치하여 센서 작동과 전송을 위한 통신(소켓 통신), 데이터 압축 등을 구현하기 위한 개발 환경을 마련하였다.

가변저항, 포토레지스터(광센서), 초음파 센서를 Raspberry pi에 연결, Python code를 통해 센서값 수집, 데이터 압축, 데이터 전송 등 일련의 과정을 이 환경에서 구현하였다.

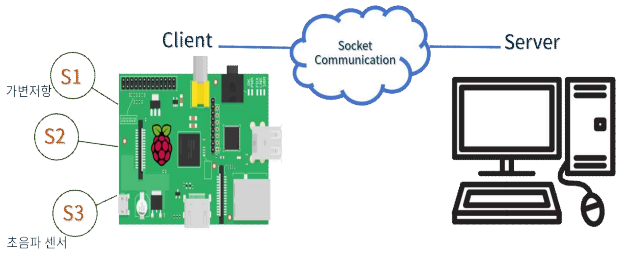

그림 4 Server(Windows 10)-Client(Raspbian)/Raspberry pi 3 센서 연결

2.4 실험 결과

Client 측에서 Raw data(870bit)를 압축하여 Compressed data (619bit)를 Server 측에 전송하였다. 이 과정에서 데이터 손실이나 오류 없이 동일한 크기와 자료 형태로 수신된다.

현재 데이터와 이전 데이터의 Difference를 Compressed data size와 비교하여 Raw data와 Difference 중 전송데이터가 결정된다. 여기서 Difference가 Compressed data size보다 크거나 같으면 Raw data를 보내고 작으면 Difference를 전송한다. 모든 데이터가 압축없이 Raw data만 보내는 경우는 Worst compressed data size의 결과를 보이며 모두 압축된 데이터만 보내는 경우는 Best compressed data size의 결과를 보인다. Real compressed data size는 실제 실험 데이터를 압축 코드에 실행시켜서 나온 압축후의 데이터 크기이다.

다음은 실험 결과이다.

Raw data size = 870(bit) 또는 108.75(byte)
Worst compressed data size = 1050(bit) 또는 131.25(byte)
Best compressed data size = 615(bit) 또는 76.88(byte)
Real compressed data size = 619(bit) 또는 77.38(byte)

압축률(C_{pr}) 공식에 위의 결과 B_{cs}(Raw data size)와 A_{cs}(Real compressed data size)를 적용하면 C_{pr}은 28.85%이다. .

$$C_{pr} = \frac{(B_{cs} - A_{cs})}{B_{cs}} \times 100 = 28.85(\%)$$

C_{pr} : Compression rate
B_{cs} : Before Compression Size
A_{cs} : After Compression Size

아래 그림5는 실험환경에서 수집한 Raw data(10진수)와 압축 후 Compressed data(16진수)의 출력화면이다.

그림5의 결과화면처럼 데이터값의 크기 변화가 급격하게 차이를 보이기보다는 일정하게 작은 변화를 보이는 데이터들에 더 좋은 압축률을 보인다. Difference가 작을수록 압축률(C_{pr})을 높일 수 있다. Difference가 작

은 값을 가지려면 수집되는 데이터의 크기의 폭이 크지 않고 Difference가 0에 가까울수록 좋은 압축률(C_{pr})을 보인다. 또는 Compressed data size를 Difference 보다 크게 지정하면 압축률을 높일 수 있다. 하지만 모두에 적용되는 것은 아니다. 이 방법은 오히려 오버헤드 6비트가 더해져서 압축률이 Raw data를 보낼 때보다 더 낮은 압축률을 보일 수도 있다.

1023	109	2533	1 ff ed 9e 50
1023	108	2599	e8 0 42 0 0
1023	109	2210	c0 0 8a 20 0
1022	109	2254	f0 20 2c 0 0
1019	109	2353	f0 60 63 0 0
1018	108	2311	fc 20 2a 0 0
1015	109	2141	f4 60 aa 0 0
1013	109	2137	f4 40 4 0 0
1010	99	2178	f8 60 29 0 0
1011	109	2060	e4 20 76 0 0
1011	107	2232	e8 0 ac 0 0
1010	109	2196	f4 20 24 0 0
1010	109	2163	e4 0 21 0 0
1008	108	2193	f8 40 1e 0 0
1008	108	2149	e4 0 2c 0 0
1006	108	2285	f0 40 88 0 0
1003	110	2071	f4 60 d6 0 0
1003	110	2206	e0 0 87 0 0
1000	109	2293	f8 60 57 0 0
1000	108	2325	e8 0 20 0 0
999	108	2285	f4 20 28 0 0
996	109	2343	f0 60 3a 0 0
994	108	2173	fc 40 aa 0 0
994	108	2298	e0 0 7d 0 0
994	109	2396	e0 0 62 0 0
992	108	2443	f8 40 2f 0 0
992	108	2530	e0 0 57 0 0
992	111	2501	e4 0 1d 0 0
992	108	2538	e8 0 25 0 0
995	108	2574	e0 60 24 0 0

그림 5 Raw data & Compressed data

Ⅲ. 결론

본 연구는 제조 산업현장에서 데이터 기반 원격모니터링 및 예지보전 기술의 활발한 도입을 위해서 학습을 위한 많은 양의 데이터 축적이 요구된다. 필요에 따라 즉각적으로 클라이언트 및 서버로의 데이터 전송이 되어야 한다. 저장 공간의 절약과 전송 효율을 높일 수 있는 한 방법으로 데이터 압축 알고리즘을 제안하였고 실제 현장에서 EDR 압축 알고리즘을 적용한 결과 압축률 28.85%를 보였다. 그 결과 데이터 크기를 줄였고 그에 따른 데이터 전송과 데이터 축적에서 오는 문제점인 네트워크 부하와 많은 저장 공간의 필요성을 줄여줄 수 있다.

향후 계획으로 본 연구에서는 Differenc 결과를 직접 확인한 후 Compressed data size를 사람이 직접 결정하였다 이를 보완한 가장 최적화된 Compressed data size를 찾아내는 알고리즘을 구현하여 데이터를 기반으로 하는 분야에 많은 도움이 될 것이라고 본다. 또한 압축률을 지금보다 더 높이는 방법에 관한 연구도 필요하다.

ACKNOWLEDGMENT

본 과제(결과물)는 2021년도 교육부의 재원으로 한국재단의 지원을 받아 수행된 지자체-대학 협력기반 지역 혁신 사업의 결과 및 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2020R1G1A1015210).

참 고 문 헌

[1] industry NEWS, “스마트팩토리, ‘예지보전’통해 다운타임 줄여야”, 2019.09.25., https://www.industrynews.co.kr/news/articleView.html?idxno=34646

[2] S.Oh and J. Kim, “Comparison and Analysis of Controller Area Network Compression Algorithms,” Transactions of KSAE, Vol.28, No.9, pp.629-636, 2020.

[3] S. Kelkar and R. Kamal, “Boundary of fifteen compression algorithm for controller area network based automotive applications,” International Conference on Circuits, Systems, Communication and Information Technology Applications(CSCITA), pp.162-167, 2014.

[4] R. Miucic, S. M. Mahumd and Z. Popovic, “An enhanced data-reduction algorithm for event-triggered networks,” IEEE Transactions on Vehicular Technology Vol.58, No.6, pp.2663-2678, 2009

[5] S. Kelker and R. Kamal, “Controller area network based quotient remainder compression-algorithm for automotive applications,”, IECON 2021-38th Annual Conference of the IEEE Industrial Electronics Society, pp.3030-3036, 2012.

[6] Y. -J Wu and J. -G. Chung, “An improved controller area network data-reduction algorithm for in-vehicle networks,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, pp.346-352, 2017