

차 례

차 례	1
1 요구조건 및 제약 사항 분석에 대한 수정사항	2
1.1 요구조건	2
1.2 제약 사항	2
2 설계 상세화 및 변경 내역	2
2.1 모델 설계	2
2.2 모델 구조	2
3 갱신된 과제 추진 계획	3
4 구성원별 진척도	4
5 보고 시점까지의 과제 수행 내용 및 중간 결과	4
5.1 데이터 전처리	4
5.2 레이블링	6
5.3 머신러닝 모델	10
참고 문헌	11

1 요구조건 및 제약 사항 분석에 대한 수정사항

1.1 요구조건

Hot/Cold 데이터를 구분하기 위하여 머신러닝 모델을 설계하고, 학습 시킨 후 이를 이용한 GC(Garbage Collection) 알고리즘을 통하여 SSD write의 성능을 개선하는 것이 과제의 목표이다. 이를 위하여 기존 SSD에 실 사용되는 알고리즘과 머신러닝을 통해 도출한 GC 알고리즘을 비교 평가해야 한다.

1.2 제약 사항

이번 과제의 제약 사항은 실제 SSD에 구현한 모델을 적용시켜 실험할 수 없다는 점에 있다. 이에 대한 해결책으로 기존에 사용되는 GC 알고리즘과 과제의 모델이 도출하는 출력을 비교하는 방식을 사용할 것이다.

2 설계 상세화 및 변경 내역

2.1 모델 설계

사용되는 워크로드 데이터가 시계열 데이터이므로 RNN(Recurrent Neural Network)을 선정했다. 그러나 RNN 모델은 규모가 있는 데이터셋에 대해 두 가지 문제를 갖는다. 하나는 기울기 소멸 문제로 신경망을 거쳐 가며 작은 미분값이 곱해져 값이 0에 가까워져 소실되는 문제이다. 이는 학습을 저해시키고 신경망 전반부의 데이터를 잃게 한다. 다른 문제는 기울기 폭발 문제이다. 기울기 소멸과 반대로 신경망을 거쳐 가며 큰 미분값이 곱해져 훈련 가중치가 비대해지는 문제이다. 이는 모델의 예측 능력을 떨어뜨리는 결과를 초래한다. 이에 대한 여러 해결책이 제시되었으며, 그중 LSTM(Long Short Term Memory)을 사용했다. 이번 과제에 사용되는 데이터셋은 Hot/Cold 분류를 위해 LBA(Logical Block Address)를 장기간 기억해야 한다. 이를 위하여 RNN 보다 메모리 유지가 긴 LSTM이 적절하다.

2.2 모델 구조

LSTM에는 각기 특성을 갖은 구조들이 존재한다. 그러한 여러 LSTM 모델 중에서 이번 과제에 사용되는 데이터 셋에 가장 좋은 출력을 내는 모델을 선정해야 할 것이다. 이를 위해 각 모델을 실험하고 그 결과를 비교해 볼 필요가 있다. 먼저 기본적인 단변량 LSTM 모델에 대한 실험을 진행했으며 그 구조는 아래 그림 1과 같다.

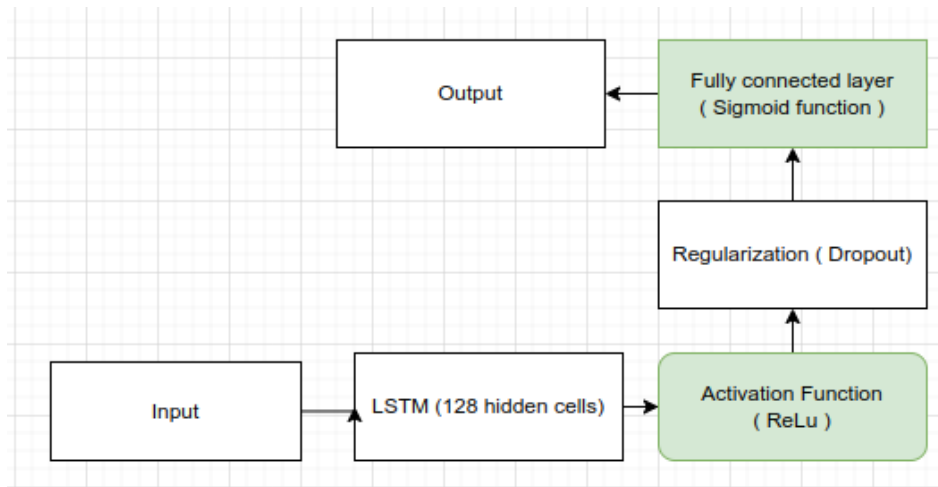


그림 1: 기본적인 LSTM 구조

단변량 모델로서 학습에 특징값으로 LBA만을 사용했다. LSTM 한 계층만을 사용하고 활성화 함수로는 ReLU를 사용했다. ReLU는 양수는 반환하고 음수는 0으로 치환하는 활성화 함수다. 해당 함수는 기울기 소실 문제가 발생하지 않고 학습 속도가 빠른 장점이 있다. 앞서 설명한 바와 같이 사용되는 학습 데이터가 기울기 소실 문제가 일어날 위험성이 있으므로, ReLU를 선정했다. 정규화 함수로는 Dropout을 사용했다. 해당 모델은 데이터를 Hot/Cold 이진 분류를 수행하므로, 마지막에 sigmoid 함수를 사용하여 예측값을 0, 1로 치환하였다. 이러한 활성화 함수, 정규화 함수, 신경망 계층은 학습 결과를 다른 함수와 비교 및 교체하며 실험을 수행할 것이다. 최적화 알고리즘으로는 Adam 경사하강법 알고리즘을 사용했다. 이는 각 단계에서 가중치를 최적화하고, 기울기에 가해지는 편향을 조정한다. 마찬가지로 이진 분류를 수행하는 모델이므로 손실함수로는 교차 엔트로피 함수를 사용했다.

3 갱신된 과제 추진 계획

5 월					6 월					7 월					8 월					9 월		
2 주	3 주	4 주	5 주		1 주	2 주	3 주	4 주	5 주	1 주	2 주	3 주	4 주	5 주	1 주	2 주	3 주	4 주	5 주	1 주	2 주	3 주
착수보고서																						
	입력 데이터 및 모델 자료공부						신경망 모델 개발															
										머신러닝 모델 개발												
													모델 학습									
														중간보고서								
														GC 결과 비교								
														모델 개량								
																				최종 발표/보고서 준비		

4 구성원별 진척도

최성찬 - 데이터 전처리 및 레이블링. K-means 군집화 수행

Ariunbold Odgerel - 신경망 모델 개발 및 훈련 데이터 처리

Ganchuluun Narantsatsralt - 머신러닝 모델 개발 및 성능 평가

공통 - 머신러닝 모델 훈련 및 자료조사

5 보고 시점까지의 과제 수행 내용 및 중간 결과

5.1 데이터 전처리

사용된 워크로드 데이터는 SSD 기반 워크로드 분석[1]에 사용된 데이터를 선정했다. 해당 데이터는 UNIX의 blkparse를 통해 작성된 워크로드로, gz 파일로 압축된 상태로 약 22.9GB 크기이며 다음과 같은 형식을 갖는다.

```
259,2 13 2861663 1227.460991324 4073 C WS 499573680 + 8 [0]
[Device Major Number,Device Minor Number] [CPU Core ID]
[Record ID] [Timestamp (in nanoseconds)] [ProcessID] [Trace Action]
[OperationType] [SectorNumber + I/O Size] [ProcessName]
```

해당 과제의 목표는 Hot/Cold 분류를 통해 WA(Write Amplification)를 줄여 쓰기 효율을 증가시킴에 있다. 고로 워크로드 데이터 중 읽기 부분은 관련이 없으므로, [Operation Type] 열의 데이터에서 W 즉 Write가 포함된 데이터만을 분리해 두었다. 또한 [Trace Action] 열은 데이터가 I/O 큐에 들어가고, 이후 드라이브에 보내지고 완료되는 것에 대한 정보를 담는다. 우리는 그 중에서 D 즉 IO가 드라이브로 보내진 데이터만을 분리했다. 더하여 위 열들 중 주소에 대한 접근 간격을 알 수 있는 Timestamp, LBA를 알 수 있는 SectorNumber 마지막으로 IO가 몇 개 블록의 크기를 갖는지 나타내는 I/O Size 열만을 표기하도록 하였다.

```
1.652816521 7487488 2048
[Timestamp (in nanoseconds)] [SectorNumber] [I/O Size]
```

위 형식이 처리 과정을 거친 데이터로, 첫 열은 경과 시간, 두 번째 열은 LBA 마지막은 IO의 블록 수를 의미하다. 2048이 의미하는 바는 한 블록이 512byte이므로 512*2048byte를 의미

한다. 위 형식으로 처리된 데이터를 모델 훈련용 데이터로 사용한다. 해당 데이터 전처리에 더하여 레이블링을 수행하기 위하여, Hot/Cold 구분을 위한 특징을 추출하도록 다음과 같은 처리를 더했다. 먼저 각 LBA가 총 몇 번 접근되었는지 횟수를 명시한 Frequency를 계산하였다. 다음으로 각 LBA 하나에 대한 접근 시의 Timestamp간의 차이 즉 접근 간격의 평균을 계산하였다. 마지막으로 각 LBA에 수행된 쓰기의 총 블록 수를 합산하였고 결과는 다음과 같다.

8488 966 36.0808 7728

[SectorNumber] [Frequency] [Time Interval Avg(in nanoseconds)] [I/O Size]

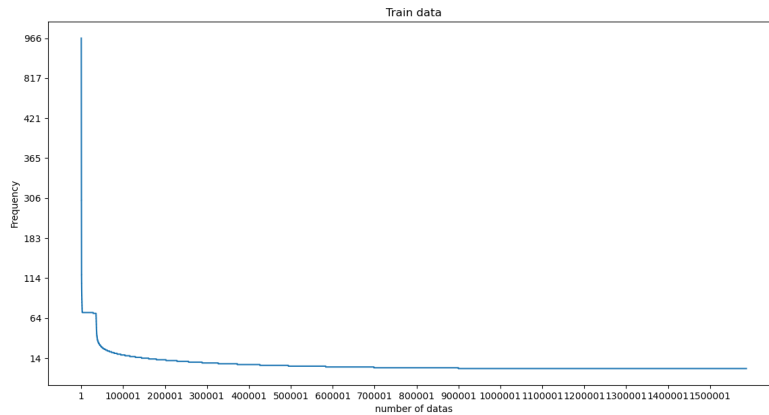


그림 2: 훈련 데이터 Frequency 시각화 자료

머신러닝 모델 훈련 후 검증과정을 위해 테스트 데이터 역시 준비했다. 데이터는 UMass Trace Repository[3]에서 제공되는 OLTP(Online Transaction Processing) Application I/O를 이용했다. 해당 데이터는 금융기관에서 발행한 데이터이다. 형식은 다음과 같다.

0,20941264,8192,W,0.551706,Alpha/NT

[Application specific unit] [LBA] [Size] [Opcode] [Timestamp] [Optional]

해당 데이터 역시 훈련용 데이터와 동일한 형식으로 만들기 위해 Opcode가 W인 것만 분리하여 [Timestamp] [LBA] [Size] 형식으로 재구성했다. 이때 해당 데이터는 Size가 byte 단위로 쓰여 있으므로 512로 나누어 블록 수로 변환하여 사용한다. Timestamp의 경우 여기서는 나노초가 아닌 초 단위를 사용하나 이러한 단위는 혼용하지 않는 이상 영향을 주지 않는다는 판단 하에 그대로 사용했다. 해당 과정을 통해 훈련 데이터와 동일한 형식으로 변환하였으므로 이후 레이블링을 수행하기 위한 전처리는 훈련 데이터와 동일한 과정을 거쳤다.

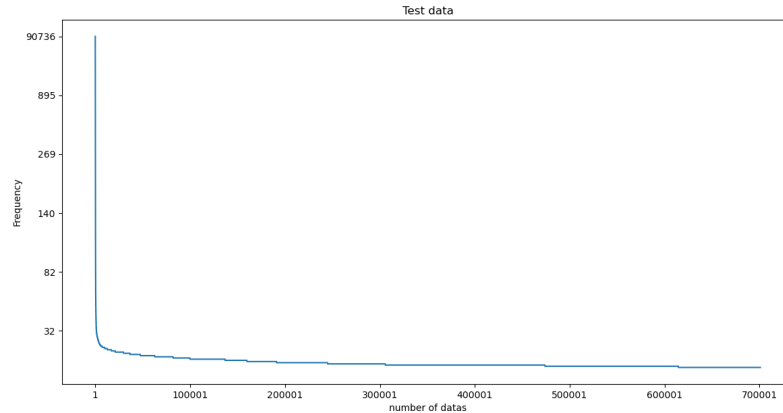


그림 3: 테스트 데이터 Frequency 시각화 자료

그림2과 그림3를 통해 두 데이터 모두 접근 빈도수가 전체 LBA 중 5% 이내에 편중되어 있음을 확인할 수 있다. 그 외 95% 데이터는 빈도수가 한 자릿수이며 그중 절반은 1회 혹은 2회의 접근이 이루어진다.

5.2 레이블링

1) 임의적 방법

레이블링은 두 가지 방식으로 진행했다. 첫 번째는 자체적인 기준을 적용하여 Hot/Cold를 구분하는 방식이다. 기준으로는 위 그림들을 참고하여 Frequency가 10 보다 큰 것을 모두 Hot 데이터로 지정했다. Frequency 10 이하의 데이터들은 TIA(Time Interval Average)를 Frequency가 가장 높은 것의 TIA에 근사한 값을 기준으로 하여 기준 보다 작으면 접근 빈도수가 적더라도, 짧은 시간 안에 다시 쓰기를 수행하였으므로 Hot 데이터로 지정했다. 이러한 기준 TIA는 훈련 데이터에서 35.9나노초 테스트 데이터에서 0.51초로 지정했다. 위 기준을 통해 Hot으로 분류된 데이터는 훈련 데이터에서는 모든 LBA의 1.5%, 테스트 데이터에서는 2.7%를 기록했다. 이러한 자체적인 기준은 머신러닝 모델 훈련을 테스트하기 위해 임의로 지정한 것이지만 이후 해당 기준을 통해 훈련한 모델 역시 성능 비교를 위해 사용할 것이다.

2) 군집화 방법

두 번째는 비지도 학습 클러스터링을 이용한 구분이다. 초기에는 DBSCAN(Density-Based Spatial Clustering of Applications with Noise)을 이용한 클러스터링을 시도했다. DBSCAN은 K-means와 달리 군집의 수를 지정해주지 않아도 되는 점, 이상치에 대한 대응이 강화된 점을

들어 테스트를 수행하려 했다. 그러나 sklearn을 이용하여 DBSCAN을 사용할시 12GB의 램을 모두 사용하여 클러스터링 수행이 불가능한 문제가 발생했다. 이는 sklearn에 구현된 DBSCAN이 $O(N^2)$ 공간복잡도를 갖는 까닭이다. 이에 tensorflow로 K-means 알고리즘을 구현한 코드를 이용하여 분류를 진행하였으나 군집 형성이 적절하지 않아 알고리즘 구현에 문제가 있는지 확인하기 위해 sklearn에 구현된 K-means를 이용한 클러스터링과 비교하였고 sklearn의 K-means가 더 적절한 분류를 수행하였다. 이에 sklearn의 k-means를 사용했다.

k-means에 사용되는 변수로는 초기에 주소 접근 빈도수, 주소 접근 간격 평균, 주소 쓰기 블록 수를 사용했다. 또한 데이터에 표준화를 수행하였는데 이는 군집화가 각 변수가 아닌 값이 큰 데이터에 영향을 받는 것을 최소화하기 위함이다. 초기에는 이러한 표준화 함수로 sklearn의 MinMaxScaler를 사용했다. 해당 함수는 데이터를 0과 1 사이의 수로 표준화 하는 함수이다. K 값은 yellowbrick의 KElbowVisualizer 메소드를 이용한 엘보우 기법을 사용해 2로 결정했다. 그 결과 훈련 데이터의 56% 테스트 데이터의 84%가 Hot 데이터로 분류되었고, 이는 부적절하다고 판단했다. 이에 K-means를 통하여 Hot/Cold 분류를 시도한 논문[2]을 참조하여 변수에서 주소 쓰기 블록 수를 탈락시키고, 주소 접근 간격의 표준편차를 추가했다. 또한 표준화 함수를 sklearn의 StandardScaler로 교체했다. 접근 간격의 표준편차가 의미하는 바는 작은 표준 편차는 주소 접근 간격 평균이 일정하며, 큰 표준편차는 이것이 일정하지 않고 들쭉날쭉함을 의미한다. 이는 비슷한 주소 접근 간격을 가진 두 군집에 대해 어느 것이 더 hot에 가까운지 판별하는 것에 도움을 준다. 다시 말해 표준편차가 큰 데이터는 짧은 연속의 업데이트 후 긴 시간 동안 업데이트되지 않은 경우를 분류하는 기준이 된다. 교체한 표준화 메소드는 평균을 0으로 분산을 1로 조정한다. 표준화 함수의 변화는 변수를 추가하기 전에도 군집화에 개선을 가져왔다. 이는 앞선 그림2과 그림3에서 보았듯 데이터가 편중된 상황에서 MinMaxScaler는 본래 데이터의 분포를 유지하는 데에서 기인한 것으로 보인다. StandardScaler의 경우 분포 역시 변화하므로 엘보우 기법을 통한 k값이 4로 결정되었고 이는 그림4과 같고 군집화 결과는 그림5와 같다.

Cluster	Frequency Avg	Time Interval Avg	Std Derivation	Blocks Avg	Amount
0	1.01	34297.24	0	1826.60	697571
1	8.51	3050.38	13280.39	16673.72	324263
2	2.78	1871.91	1081.52	5433.33	242486
3	17.20	1778.10	22539.26	17664.52	322380

표 1: 훈련 데이터 군집화 수치

분류된 데이터의 결과는 표1과 같다. 해당 표에서 군집 0이 표준편차가 0을 갖는 이유는 접근 빈도수가 2 이하인 데이터는 표준 편차를 구할 수 없으므로, 0을 할당했기 때문이다. 이는 해당 군집의 모든 데이터가 2회 이하의 빈도수를 갖음을 의미한다. 또한 처리 과정에서 빈도수 1인 데이터는 접근 간격을 구할 수 없으므로 최대 접근 간격에 1을 더한 수를 할당했다. 그 결과

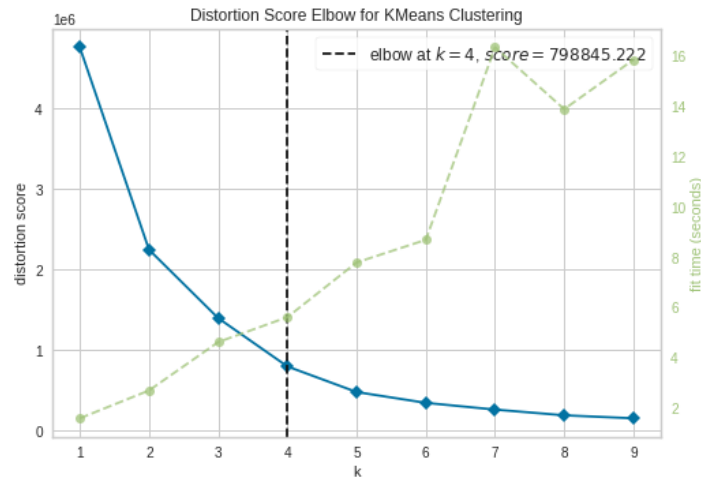


그림 4: 훈련 데이터 Frequency 엘보우 기법

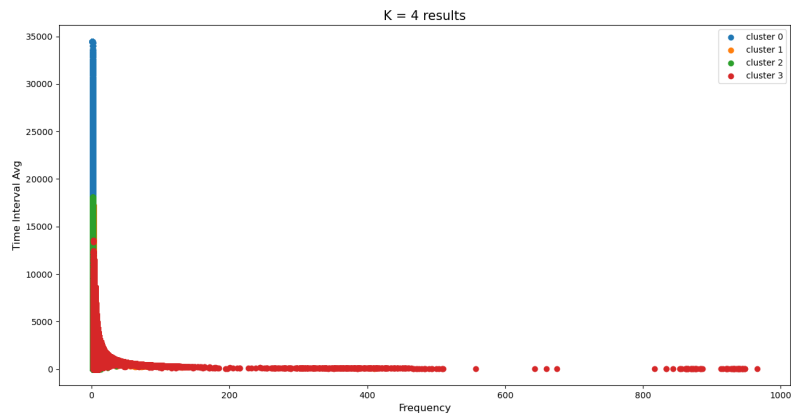


그림 5: 훈련 데이터 군집화 결과

군집0이 가장 큰 접근 간격 평균을 갖게 된 것이다. 고로 군집 0은 Cold 데이터이다. 군집1은 군집2보다 접근 빈도수가 크지만, 접근 간격 평균역시 더 크고 표준 편차 역시 크다. 이는 군집2가 평균 접근 빈도수가 작더라도, 짧은 접근 빈도수를 가질 뿐 아니라 접근 간격의 편차가 적어 접근 간격의 측면에서는 군집2가 군집1에 비해 더 Hot에 가깝다고 볼 수 있다. 군집 3은 가장 높은 빈도수 평균, 가장 낮은 접근 간격 평균을 갖고 있다. 또한 쓰인 블록 역시 가장 많다. 군집 3은 총 데이터 중 20%를 차지하므로 이를 Hot 데이터로 레이블링하여 실험을 진행할 것이다. 테스트 데이터에 대해 K-means 군집화를 적용한 결과 그림6, 그림7와 같다.

테스트 데이터에 대한 군집화 역시 엘보우 기법에 따라 K값을 4로 설정하여 수행했다. 그 결과가 표2이다. 군집 중 평균 접근 빈도수가 가장 높은 1번 군집을 Hot 데이터로 설정하면 전

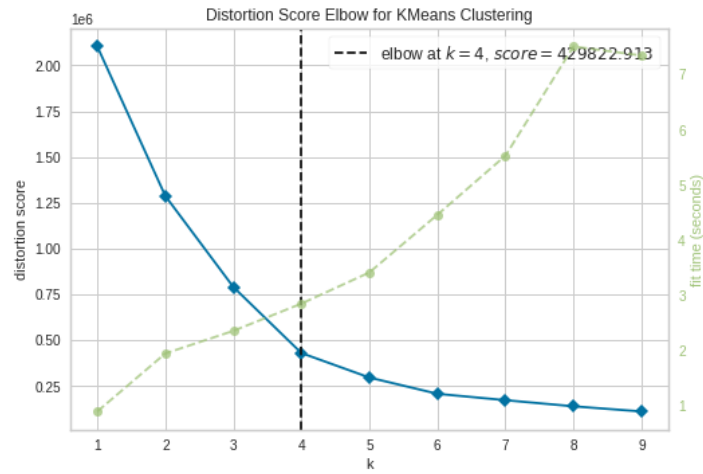


그림 6: 테스트 데이터 Frequency 엘보우 기법

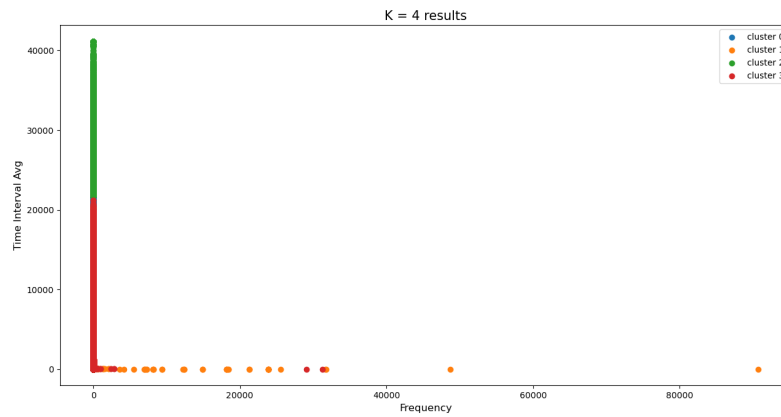


그림 7: 테스트 데이터 군집화 결과

Cluster	Frequency Avg	Time Interval Avg	Std Derivation	Blocks Avg	Amount
0	2.26	4354.45	955.93	16.25	149575
1	11.15	1990.28	26167.93	76.73	184942
2	1.15	39117.26	0	17.68	101204
3	5.97	5373.69	16545.31	45.80	265152

표 2: 테스트 데이터 군집화 수치

채 LBA 중 26%의 데이터가 Hot으로 분류된다. 이후 기존 파라미터에 추가로 쓰기 블록 수를 더했을 때 결과는 아래 그림 및 표와 같다. 이후 이를 이용한 모델 훈련 결과에 대한 평가 및 비교를 진행할 수 있다.

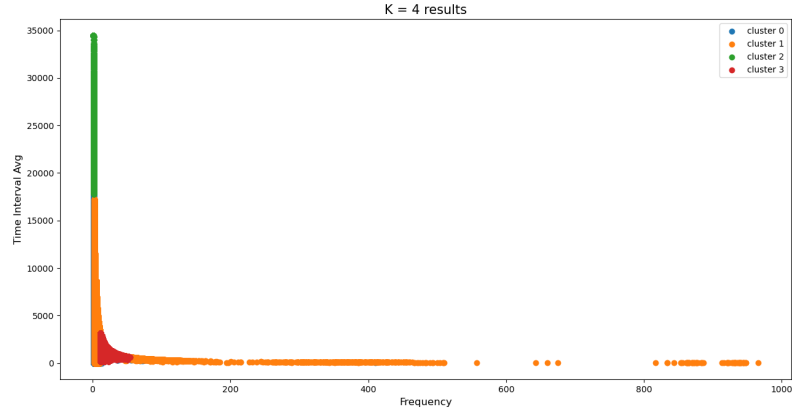


그림 8: 훈련 데이터 군집화 결과 - Size 추가

Cluster	Frequency Avg	Time Interval Avg	Std Derivation	Blocks Avg	Amount
0	2.75	2110.45	2143.05	5232.92	277663
1	11.80	2662.60	18461.54	11187.22	449568
2	1.01	343007.17	0	1826.04	697444
3	17.95	1453.81	18145.60	366422.99	162025

표 3: 훈련 데이터 군집화 수치 - Size 추가

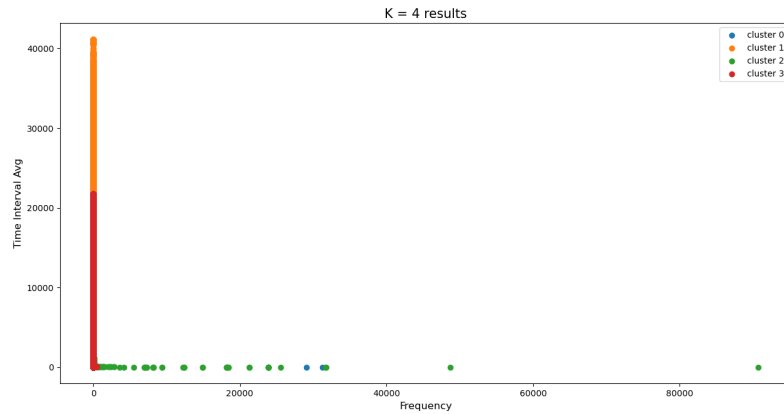


그림 9: 테스트 데이터 군집화 결과 - Size 추가

5.3 머신러닝 모델

모델 설계와 구조는 앞선 장에서 설명을 진행했다. 기본적인 LSTM 모델을 통해 실험중에 있으며, Convolutional LSTM과 Stacked LSTM 모델을 실험하기 위해 모델을 구성 중에 있다. 또한 현재는 단변량 모델을 실험 중에 있으나 I/O 크기나 Timestamp를 추가한 다변량 모델에

Cluster	Frequency Avg	Time Interval Avg	Std Derivation	Blocks Avg	Amount
0	5.94	5380.52	16522.53	45.49	264245
1	1.14	39150.31	0	17.67	101012
2	11.16	1997.42	0 26139.72	76.97	185996
3	2.26	4378.62	946.90	16.25	149620

표 4: 테스트 데이터 군집화 수치 - Size 추가

대한 실험 역시 진행할 것이다.

앞서 설명한 데이터 전처리에 더하여 모델에 사용하기 위해 추가적인 전처리를 수행했다. 모델을 훈련시키는 데에 문제가 발생하여 문제 진단을 한 결과 문제는 다음과 같다. LBA는 주소이므로 1000000000과 같이 큰 수를 포함한다. 이는 신경망을 거치며 곱셈이 일어나는 과정에서 수의 비대화를 발생시켜 모델 학습 속도와 예측 능력을 저해시키는 결과를 낸다. 이에 one-hot 인코딩을 적용했다. 이는 큰 수를 행렬로 표현할 수 있게 해주어 GPU 내 연산을 통한 빠른 학습을 가능하게 한다. 그러나 여전히 기울기 폭발 문제가 발생하여 손실함수의 출력이 NaN이 나오는 문제가 발생중에 있다.

참고 문헌

- [1] Gala Yadgar et al. SSD-based Workload Characteristics and Their Performance Implications, ACM Transactions on Storage, Volume 17, Issue 1, February 2021, Article No.: 8, pp 1-26
- [2] 정윤희, 정혜미, 김재호 (2021). SSD 쓰기 증폭을 줄이기 위한 머신러닝 기반 정교한 Hotness 분류 방안. 2021년 한국소프트웨어종합학술대회 논문집
- [3] UMassTraceRepository <https://traces.cs.umass.edu/index.php/Storage/Storage>