악성코드 파일기반 탐지방법에 대한 연구

윤주영*, 김상훈*, 최선오*
*호남대학교 컴퓨터공학과
pulpul8282@naver.com, dj95390@naver.com, suno@honam.ac.kr

Research on File-based Malware Detection Method

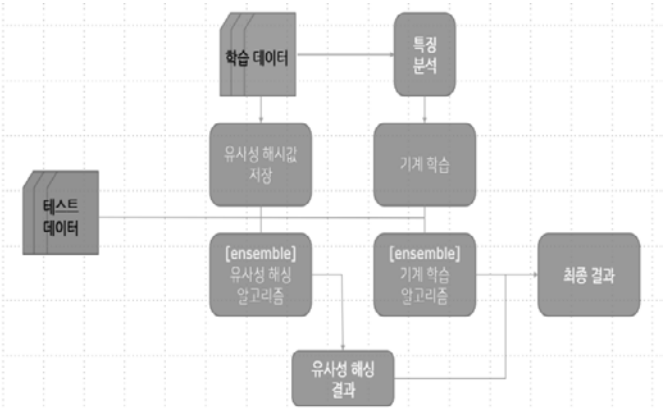
Ju Young Yoon*, Sang Hoon Kim*, Seun O Kim*
*Dept. of Computer Science, Ho Nam University

요 약

본 연구는 파일기반 악성파일 탐지시간을 줄이는 알고리즘 사용에 대해 기술하고 있다. 기존 탐지방식은 파일의 시그니처 값에 대한 유사도를 단순히 비교하는 것에만 그쳐 오탐율이 높거나 새롭게 생성되는 악성파일을 대응할 수 없는 제한점이 있다. 또한 정확도를 높이고자 딥 러닝을 통한 탐지방식이 제안되고 있으나 이 또한 동적분석으로 진행이 되기 때문에 시간이 오래 걸리는 제한이 있다. 그래서 우리는 이를 보완하는 VP Tree 탐지를 제안한다. 이 방법은 시그니처 값이 아닌 다차원에서의 해시 값의 데이터 위치를 기반으로 거리를 척도 한다. 유클리드 거리 법, 맨해튼 거리법이 사용되며 삼각부등식의 만족하는 기준으로 K-NN 이 생성이 되며, K-NN 을 이진 트리로 구성하여 인덱스를 통한 탐지를 진행하기에 기존 방법들을 보완할 수 있는 대안점이 될 수 있으며, 악성파일과 정상파일이 섞여 존재하는 총 3 만개의 데이터를 대상으로 악성파일 탐지 테스트를 진행하였으며 기본 방식에 비해 약 15~20%정도 속도가 단축된다는 것을 입증했다.

1. 서론

2017 년 5 월 12 일 랜섬웨어 라는 사이버 공격에 150 여개국 20 만대 이상의 컴퓨터를 감염시키며 전세계는 공포에 휩싸였다. 비트 코인이 등장하면서 랜섬웨어 같은 파일 기반 악성코드들이 더욱더 활개치고 있고, 새로운 형태의 랜섬웨어 들이 생성되면서 사이버 생태계를 위협하고 있다. 이 때문에 EDR(End point Detection & Response) 필요성이 대두되고 있다. EDR 의 과정은 탐지-대응-조사이다. 조직 내 엔드 포인트에 대한 가시성을 확보하고, 이를 통한 위협분석, 탐지하여 대응하는 것이다. 본 연구는 탐지 과정의 초점을 맞춰 탐지 알고리즘 방법에 대해 제시하고 있다. 파일기반 탐지방법은 기존 3 가지의 방식이 있다. 첫째는 시그니처를 기반으로 탐지하는 것이며 파일 자체의 특정부분 또는 고유한 부분의 문자열을 대상으로 삼아 검사를 진행하며 빠른 스캔이 장점이라 할 수 있다. 그러나 새로운 내용이 추가되어 파일의 바이트 값이 조금만 바뀌더라도 미탐율이 현저하게 높아져 이는 새롭게 생성되는 악성파일을 대응할 수 없는 단점이 있다. 이를 보안하여 둘째 휴라스틱 탐지기법이 존재하며 악성코드가 갖고 있는 특정 폴더에 파일쓰기, 특정 레지스트리의 키생성과 같은 명령어들을 스캐닝엔진 핸들러로 시그니처화 하여 이 시그니처를 비교를 하나 윈도우 팝업 창, 생성 같은 명령어들의 유무에 따라서 악성파일을 판단하는 치명적 단점이 있다. 세 번째 제네릭 탐지 기법은 비슷한 악성파일은 비슷한 opcode 를 갖고 있기 때문에 opcode 의 명령어 진행순서를 비교해서 파악을 하게 된다. 내용이 새롭게 변경되더라도 탐지가 가능하나 완전히 opcode 의 명령어 진행순서를 뒤집어 놓으면 탐지를 못하게 된다.

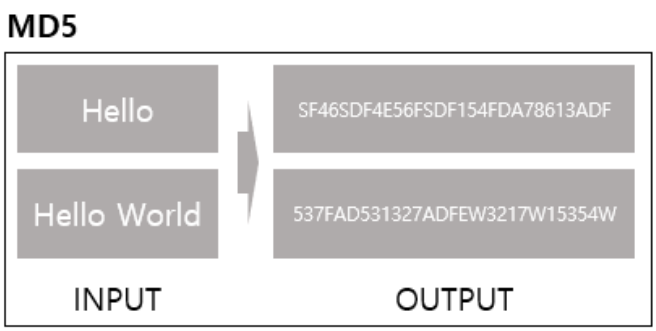


(그림 1) VP-Tree 와 딥러닝을 이용한 탐지 프로세스

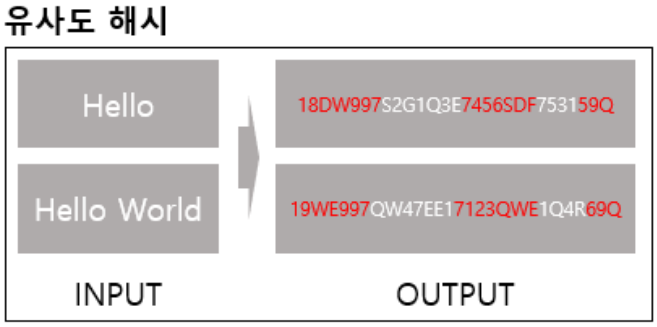
또한, mvHash-B, Sdhash 등 기존의 알고리즘들 또한 입력 순서가 달라지거나, 데이터의 길이가 다른 경우, 또는 파일 안에 무의미한 내용이 추가되는 경우 일관성 없는 결과를 발생시킨다. 결국, 이 3 가지 방식은 각각이 제한점들이 존재하여 탐지의 정확성이 떨어진다. 그리하여 최근 인공지능(회귀분석 기법 또는 합성곱 추출기법을 사용한 딥 러닝 기반 탐지방법)과 같은 악성파일 탐지방법이 제안되었고 정확성은 확연히 높아지나 동적 분석이 필요하여 오랜 기간이 소모된다는 단점이 있다 이를 해결하기 위하여 VP-Tree 를 제안하고 있다.VP-Tree 는 K-NN 과 유사도 해시 값을

이용하여 탐지하는데 정확도 면에서는 인공지능보단 떨어지나 탐지 속도면에서 빠르기 때문에 초기 탐지 과정에서 VP-Tree 를 이용하고 추후 딥러닝 방식을 융합하여 (그림 1)과 같이 진행한다면 기존 방식보다 훨씬 빠르고 정확성이 높은 효율을 기대할 수 있게 된다.

2. 관련 연구



(그림 2-1) MD5

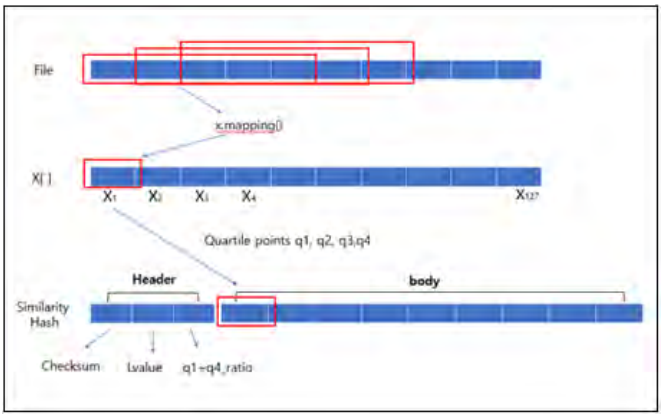


(그림 2-2) 유사도 해시

유사도 해시의 방법을 이용하면 유사한 파일들은 유사한 해시 값을 생성 해내기 때문에, 파일간의 유사도를 측정할 수 있게 된다. MD5형식으로 해시 생성시 (그림 2-1)과 같이 비슷한 파일이더라도, 완전히 다른 해시 값을 생성하여 두 파일의 유사도를 알 수 없다. 그러나, 유사도 해시 형식으로 생성시 MD5와 다르게 비슷한 파일은 비슷한 해시 값을 생성하게 된다.

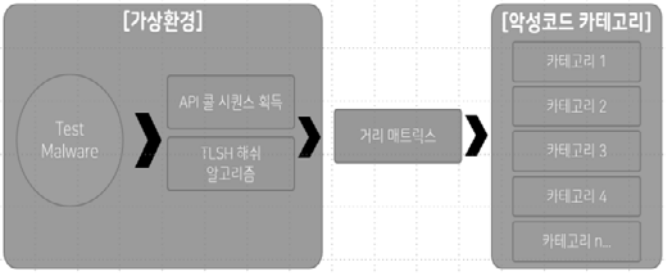
(그림 3) new_code, old_code의 유사도 해시 생성

(그림 3)은 기존 'new_code' 텍스트 파일과 새로운 내용을 추가하여 만든 1KB 용량이 증가한 'old_code' 을 유사도 해시를 통하여 해시 값을 생성하였으며, 코드 상의 diff 함수를 통하여 두 값의 비교를 진행하였으며, 두 파일은 비슷하다 라는 결과 값이 추출된 것을 확인할 수가 있다." 유사한 파일들은 유사한 해시 값을 갖는다." 라는 것이 기본 원리이며, 해시 된 데이터를 유클리드 거리 측정법과 같은 유사도 값을 도출하여 파일의 유사 정도를 비교하여 추후 악성 파일들을 탐지해낼 수 있다.



(그림 4) TLSH 사의 유사도해시

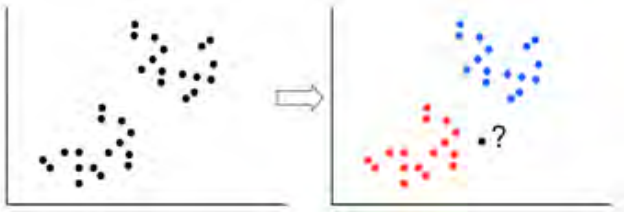
Trendmicro 사의 유사도 해시 생성 및 비교과정을 예로 들 수 있다. 첫째, 슬라이딩 윈도우를 5바이트씩 진행하여 파일을 순차적으로 읽도록 한다. 읽혀서 나온 값들은 다시 카운팅이 된다. 둘째, 카운팅 된 값들을 128차원의 배열상에 매핑이 되고 매핑 된 값들은 다시 1~4분위수의 조건을 통하여 X_i<1 분위수인 경우 00 값이 대입되고, 1분위수 < X_i <2분위수 인 경우 01을, 2분위수< X_i <3분위수인 경우 10을, 3분위수 < 인 경우 11을 대입하여 0~127까지의 TLSH body 부분이 생성이 되며, 검사 합, 파일길이, 1~4분위수의 비율로 구성된 헤더 부분을 추가하면 비로소 hash 값이 완벽히 구성된다. 단, 파일 비교 시 헤더 부분을 제외하고, body 부분만 사용하여 비교해야 할 파일 부분과 기존 파일을 뺄셈을 통하여 차이 값을 계산한다. 차이 값이 크면, 두 파일은 다른 파일이 되고, 작으면 비슷한 파일로 판단할 수 있게 된다. 허나, 멜 웨어 같은 수 억 개의 파일들을 가지고 있는 파일은 일일이 수 억 번의 비교를 해야 되기 때문에, 많은 시간과 연산이 필요하여 실질적으로는 힘들다. 이 때문에 본 연구에서는 유사성 해시 색인을 이용하는 방법을 제안하고 있다. 유사 파일들로 구성되어있는 트리형태에서 인덱스를 통하여 검색하기 때문에, 검색시간이 감소하여 보다 효율적이다.



(그림 5) 유사도해시 탐지 모델

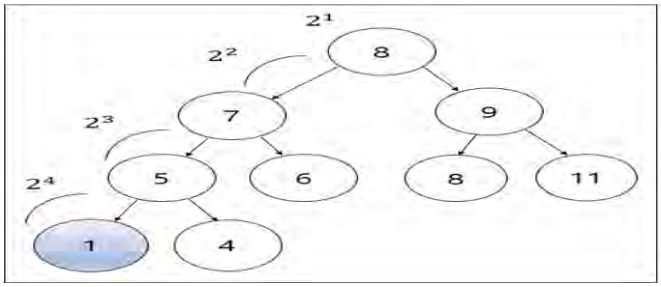
(그림 5)는 유사도 해시 기반 모델 프로세스이다. 테스트 파일로부터 시퀀스 데이터를 추출한다. 추출

된 값들은 TLSH사의 해시 알고리즘을 적용하여 각 파일의 유사도를 비교한다. 그러나, 앞서 말했던 제한점이 있기 때문에 거리 매트릭스 알고리즘을 한 번 더 이용하여 보완할 수 있다. 그리하여 다음은 거리 매트릭스를 이용하는 VP-Tree에 대해 설명한다.



(그림 6) K-NN을 통한 데이터 분류

(그림 6)를 보면, 왼쪽은 임의로 분포되어 있는 데이다. 이 때, K-NN의 방법이 사용이 된다. K-NN의 방법이 사용이 된다. K-NN은 특징 공간 내 K개의 가장 가까운 훈련 데이터로 구성되어 있다. 비 분류된 임의의 데이터 들에서 하나의 데이터로부터 가장 비슷하게 분류된 데이터들의 클래스를 할당해 부류하는 과정이다. 지도 학습 알고리즘이라고도 불리며, 새로운 데이터가 들어왔을 때, 이를 어디에 분류를 할지 정해준다. 하지만, K-NN을 사용하기 위해선 제약 사항이 존재한다. 고차원의 데이터에서는 차원의 저주. 즉, 차원이 증가하면서 그 안에 데이터의 수가 차원보다 적을시, 빈 공간이 많아져 오히려 성능이 저하되는 현상이 발생한다. K-NN에서 차원의 저주는 기본적으로 유클리드 거리가 고차원에선 도움이 되지 않음을 의미한다. 왜냐하면, 모든 백터가 탐색질의 백터와 대부분 같은 거리를 갖고 있기 때문이다. 이를 방지하기 위하여 미리 각 데이터의 유사도를 측정하고, 각 데이터들을 2차 평면에 특정 조건에 따라 나열해 놓은 후, 이를 분류하게 된다. 분류된 데이터들은 다시 거리 매트릭스 기반인 VP-Tree를 사용하여 트리화 시키면, 검색을 진행했을 시 효과를 볼 수 있게 된다.



(그림 7) 이진트리

(그림 7)의 이진 탐색 트리를 보며 확인할 수 있는데, '1' 이라는 데이터를 찾고자 한다면 Brute Force Search를 통하여 일일이 9번을 검색해야 하지만, 이진 탐색 트리는 4번만에 찾을 수 있어, 검색 횟수가 이론적으로 효율적이다. 본 연구는 이진 트리에서 좀 더 변형된 형태인 VP-Tree를 사용할 것이다. 이진 탐색 트리 같은 경우, 노드 생성기준이 단순히 부모 노드의 크기로부터 결정된다. 부모 노드 보다 크기가 작으면, 왼쪽 노드로 배치되고, 크기가 크면 오른쪽 노드로 배치되나, VP-Tree는 부모 노드의 데이터의 검색 반경 값을 기준으로 반경 값 안에 포함되면 왼쪽 노드로 배치되고 반경 값 밖에 있으면, 오른쪽 노드로 배치된다. 해당 반경 값에 위치되어 있는 데이터와 다른 데이터와의 거리를 비교해서 두 데이터간 유사도를 판단하게 된다. 다음에선 좀 더 자세한 노드 생성 과정을 설명한다.

3.생성 알고리즘

N개의 데이터들이 있다는 가정 하에 VP=Tree 생성 알고리즘 순서에 따라 최초의 루트 O_1을 생성한 후 O_1의 원의 반경 안에 포함된 데이터들을 제외후 (1)~(3)을 다시 적용하여 새로운 반경을 생성한다. 최초로 생성된 O_1가 데이터 검색의 시작점이다. 생성된 O_1을 기준으로 반경 값 안에 있는 데이터들은 K-NN에 기반하여 가까운 값들끼리 왼쪽 노드들 구성해 나간다. 오른쪽 노드 또한 반경 값 밖에 있는 데이터들 중 거리가 가까운 값들을 오른쪽 노드를 우선순위로 배치하여 구성해 나간다. 단, 거리계측에 있어서 삼각형 부등식에 꼭 만족해야 한다는 제약 조건에 성립 되어야 한다.

***VP 트리 생성 알고리즘**

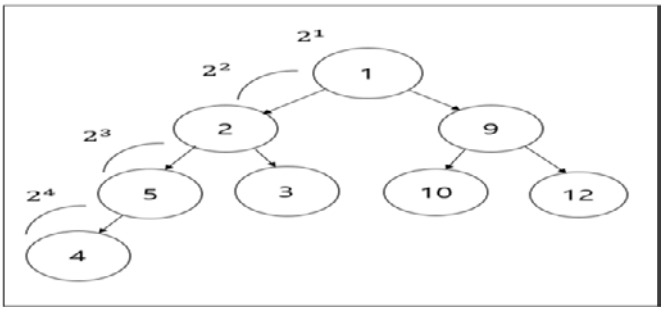
(1) 데이터의 집합 으로부터 임의의 데이터 하나를 선정하여 N-1개의 데이터들까지의 거리 값 을 계산한다

(2) 계산된 거리 값 들을 길이순으로 나열하여 중간 값을 추출한다.

(3) 추출된 중간 값 을 반지름으로 택하여 원을 생성한다.

(4) (1)~(3)을 반복하여 원이 최대값가지 이르는 지점을 VP로 결정한다.

(그림 8-1) VP-Tree 생성 알고리즘



(그림 8-2) VP-Tree

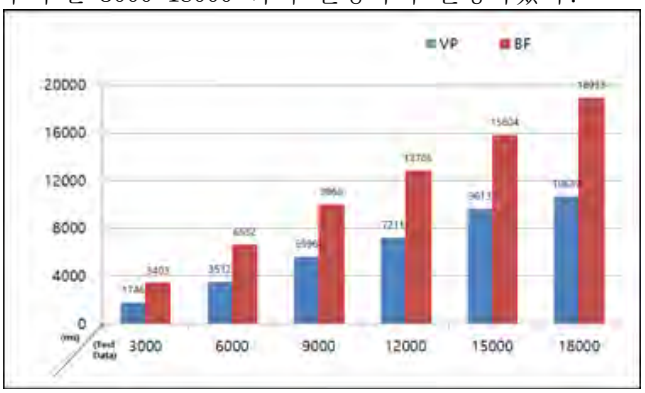
4. 검색 알고리즘

검색 방법에는 K-NN (최 근접 이웃 : K-Nearest Neighbor) 방식과, 범위 지정 검색 방식 2 가지를 사용 하였다. K-NN 은 삼각형 부등식과 유클리드 거리법, 맨허튼 거리법이 거리 계측에 사용된다. 임의의 데이터 K 를 지정하여 K 를 기준으로 가까운 거리에 있는 데이터들을 검색해 나가 결국 노드에 따라 이어나가는 방식이다. 범위지정 검색방식은 검색 반경을

임의로 설정해주어 원의 최대 값 안에 있는 데이터들을 검색해 나가는 방식이다. 그래서 검색 진행시 지정해둔 반경 내의 데이터 들을 모두 검색하는 Brute Force 검색을 진행해야 한다. K-NN 검색은 거리를 기준으로 검색을 진행해 나가는데, 생성된 VP 에서 O_1(루트노드), O_2 찾아내야 할 쿼리 노드에서 트리 생성 과정에서 루트 노드 원 안에 포함된 데이터 들은 왼쪽 자식 노드 값으로 밖에 있는 값들은 오른쪽 자식 노드로 구성이 된다. 이 때문에 O_1반경 값 밖에 O_2가 있을 시, 트리에선 왼쪽 노드를 제외하고 오른쪽만 탐색한다. 루트 노드 안에 쿼리 노드가 포함되어 있으면, 오른쪽은 제외시키고, 왼쪽 노드만 탐색하면 된다. 이로 인해서 탐색시간을 절반을 줄일 수 있는 효과가 있다. 단, O_1, O_2의 반경 원이 겹쳐 있을 시, 그 사이 교집합 값을 찾아야 하기 때문에 모든 값을 찾는 Brute Force 검색을 진행해야 한다. K-NN 은 이러한 검색 방식 덕분에 범위지정 검색방식보다 검색시간을 절반 단축 시킬 수 있는 이점이 있다.

5.실험결과

본 실험에서는 백신 회사 하우리에서 제공하는 악성파일 15000개 정상파일 15000개 학습 데이터로 구성된 전체표본 3만여개의 데이터로 총 2가지의 테스트를 진행하였다. 첫째는 Vp-Tree 검색과 BF 검색의 검색시간비교, 둘째는 탐지율 테스트이다. 두 실험은 총 실험표본 개수를 3만여개의 내에서 진행 하였으며 첫 번째는 학습데이터 12000개로부터 테스트 데이터의 수만 3000~18000 까지 변경하여 진행하였다.

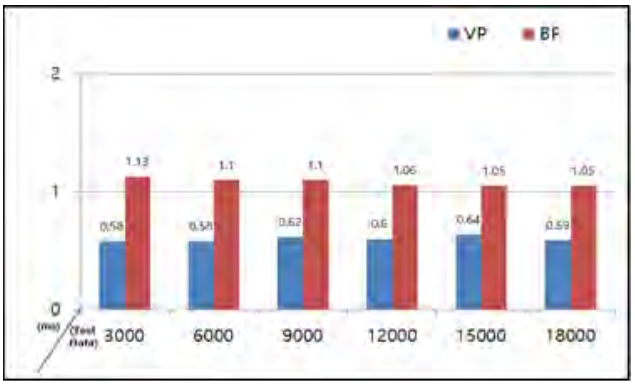


(그림9-1) 총 검색시간

Test Data(개수)	3000	6000	9000	12000	15000	18000
VP(MS)	1746	3532	5596	7211	9613	10639
BF(MS)	3403	6632	9966	12786	15804	18933

(그림 9-2) 그림9-1의 수치표

결과값으로는 (그림 9-1)와 같이 약2배정도의 시간차이로서 VP- Tree가 더 효율적이라는 것을 확인하였다. 두 번째는 학습데이터를 두 번째는 학습데이터를 24000개로 고정시킨 뒤, 각각의 테스트 데이터만 1000, 3000, 6000로 변화 값 만 주어 탐지율 테스트를 진행하였다.(그림 10)은 단순히 악성파일에서의 악성탐지(TF),정상파일에서의 정상탐지(TN),이 두가지에도 속하지 않은 미탐비율을 나타낸 것이다.최종적으로는 평균 탐지율은 91.3%,평균 미오탐율은 8.6%



(그림 9-3) 평균 검색시간

Test Data(개수)	1000	3000	6000
A그룹(TP)	428(85.6%)	1340(89.3%)	2717(90.6%)
B그룹(TN)	475(95%)	1404(93.6%)	2811(93.7%)
C그룹(미탐율n오탐율)	97(9.7%)	258(8.5%)	471(7.9%)
탐지율	90.3%	91.5%	92.1%

(그림 10) Test data 개수에 따른 TN,TF 및 미오탐비율

6. 결론

본 연구에서는 악성파일 탐지에서 이용되는 유사도 해시와 거리 매트릭스 기반인 VP-Tree를 사용하여 본 파일과 유사한 파일을 찾는 검색시간을 단축시키는 기법을 제안하였다. 기존 검색 방식인 BF와 비교하여 테스트를 진행하였으며, 약 2배의 효율을 내는 것을 확인하였다. 탐지율은 평균91.6%이며 미오탐율 또한 8.6%로서 기존 백신들의 갖고있는 미오탐율의 리스크보다 적음을 알 수 있다.이를 통해 본 연구가 제안하는 유사도 해시와, VP- Tree를 사용하는 것이 효율적인 것 임을 최종 확인하였다. 추후 딥러닝 방식과 연계하여 악성파일 탐지에 적용한다면 정확성 및 탐지속도의 향상이 기대된다.

참고문헌

[1] Sunoh Choi, Youngsoo Kim, Jonghyun Kim . ”Similarity Hash Index ICTC2018”

[2] GilYang Park. Sangkon Lee, JeaJeong Hwang. “Study of Improvement of Search Range Compression Method of VP-tree for Video Indexes”

[3] Palol Ciaccia , Marco Patella , Pave1 Zezula. “M-tree: An Efficient Access Method for Similarity Search in Metric Spaces”

[4] Sujeong Kim, Jihee Ha, Soohyun Oh, Taejin Lee “A Study on Malware Identification System Using Static Analysis Based Machine Learning Technique”