캡스톤 18 조 김민송 멘토링 활용 보고서

프로젝트명	O24Sec	팀명	멜러리를 찾아서

팀 멘토링

[03/16] 팀 미팅 - 윤명근 교수님

(Q) 현재 암호화 판별에 대해 Entropy 임계 값을 최대와 평균의 차이를 평균에서 뺀 걸로 잡아보았습니다, 이렇게 해도 될까요?
[검증을 통해 괜찮은 수치의 임계 값이지만 수학적 근거가 없는 단순 조합 값이기 때문에. 이렇게 만든 임계 값이 신뢰감을 줄 수 있을지 의논]
(A) 암호화 판별 알고리즘에서 임계 값이 너무 데이터 중심적이다, 정규 분포 같은 통계적 수치를 사용해 보자
=> 엔트로피 값들이 정규 분포를 따르는 지 확인해보기 만약 된다면 정규 분포 같은 내용은 수학적으로 정말 타당한 근거 이기 때문에 확실한 임계 값 설정으로 할 수 있을 거라는 확신이 듬

(Q) 내부/외부 판별 기준으로 상대 network 카디널리티 기준을 18 로 잡아 봤습니다. 현재 가지고 있는 데이터 셋에서는 분리가 잘되어서 한번 이 수치에 대해서 얘기해 보려고 합니다.
[현재까지 목표 1 번에 대한 완성도 검증과 발견된 사항에 대한 공유]

(A) 내부/외부 서버/클라이언트 판별에 대해서 만들어 놓은 알고리즘에 대한 검증을 해보면서 진행을 해야한다. 현재 만들어 놓은 임계값도 해당 기간에 대해서만 통하는 것일 수 있으니 다른 기간을 정해 해당 값으로 다른 데이터에서도 사용이 가능한지 검증을 해보자.
=> 단순히 지금 가지고 있는 데이터에 대한 분석이 아니라 현재 분석해 놓은 데이터에 대해 다른 데이터에 적용 했을 때도 통용 할 수 있게 검증을 꾸준히 해야 한다는 생각을 다시 얻을 수 있었다 너무 현재 데이터에만 몰두를 하고 있었 던 것 같다.

[03/19] Ceeya 미팅 - 문성익 멘토님

(Q) 현재 데이터를 통계적으로 분석하여 순수한 내부만을 찾을 수 있는 특정 임계 값으로 설정했는데, 내부/외부 임계 값을 어떻게 잡을까요???
[현직에서 임계 값을 정할 때 다루는 방법에 대한 질문]

(멘토) 현재 클러스터링 실험을 하고 있으므로, 일단 순수한 내부 데이터만을 뽑을 수 있는 임계 값으로 잘 설정했다고 생각합니다. 클러스터링 성능을 보면서 임계 값을 낮추는 식으로 진행하면 좋을 것 같습니다.
=> 일단 지금 만들어진 임계 값에 대해서 실험용으로는 괜찮다고 하심, 실험 결과에 따라 임계 값을 어느정도 바꿔가면서 진행해 보는 것을 권장해 주심.

(Q) 기사에 대해 자연어 처리를 해서 비슷한 부류끼리 유사도 기반을 통한 클러스터링을 진행하셨었는데 해당 프로젝트 진행 시 어떤 과정을 통해서 진행하셨나요?
[진행하셨던 프로젝트가 현재 캡스톤 팀의 최종 목표인 유사도에 기반한 클러스터링이기 때문에 과정속에서 우리 프로젝트에 적용시킬 부분이나 어떻게 시작하셨는지 여쭙고 싶었음]

(멘토) 먼저 기사에 나오는 언어를 사전 기반으로 처리를 하였습니다. 자연어 처리 라이브러리를 통해 나온 단어들을 체크하고 tf-idf 유사도 분류 방법으로 진행을 했습니다. 이렇게 안하더라도 n-gram 방식으로 언어를 처리해서 하는 방법도 있었네요. 그 당시 n-gram 에서 n 의 크기를 계속 바꿔가면서 적절한 결과를 얻어낼 때까지 주먹구구 방식으로 진행을 했었습니다.

현재 진행하시는 프로젝트에서 적용을 해볼때 바이트 단위를 n-gram 으로 길이별로 쪼개서 나오는 바이트 열을 벡터로 만들어 해당 벡터에 대해 tf-idf 로 검증해보는 것도 해 볼만한 시도일 것 같습니다.
=> 유사도 처리를 위한 기본적인 자연어 처리방법이나 시도해볼만한 실험을 많이 얻을 수 있었다.

(Q) 그렇다면 tf-idf 검증을 하실때 기준을 잡아야 하셨을텐데 어떤걸 기준으로 잡으셨나요? 그때 기준을 뽑는 방법은 어떤게 있으셨나요?
[현재 단순한 청킹으로 유사도 검사를 tf-idf 로 할 때 데이터가 너무 많이 때문에 어떤 데이터를 기준으로 잡아야 하는지 어려움을 가지고 있어서 비슷한 프로젝트를 하실 때 어떻게 하셨는지 질문함]

(멘토) 사실 이때 기준을 뽑을 때는 확연한 기준이 있었습니다. 사람이 한 기사를 봐서 딱 봐도 해당 분류에 대한 기사라는 것을 알 수 도 있었죠. 그리고 이미 약간 분류 되어있는 것에 잘못된 분류가 없는지 실험을 할 때도 있어서 이때 잘 분류되어 있는 기사를 통해 분류를 진행하기도 하였습니다.
=> 확실한 해답은 얻지 못했지만 그래도 어느정도 특징이 확실한 기준을 잡아서 비교를 하면 좋을 것 같다는 방법을 얻었다 우리 데이터 중 페이로드에서 이벤트의 특징을 확실하게 가지고 있는 데이터를 골라서 기준으로 만들어서 실험을 해보는 것 도 좋을 것 같다.

(멘토) 그리고 만약 단어의 순서에도 중요한 연관이 있을 수 있으니 그럴때는 n-gram 방식에서 n 의 크기를 늘려 연속되는 문자열을 벡터에 넣기도 하였습니다. 이럴 경우에 전체적인 문장이 벡터화 되기 때문에 그 문장에 대해 검증을 할 수 있으니까요 만약 프로젝트 진행하시면서 순서나 긴 길이에 대한 시그니처가 있다면 n 의 크기를 키워서 자르는 방식이 있을 수 도 있을것 같습니다. 근데 일단 n 의 크기를 작게 만들어서 확인해보면서 n 의 크기를 키우면 계속 비교해가면서 진행할 수 있으니 더 좋겠네요
=> 지금 해외 논문에서 소개된 AE 청킹 방법으로 문자열 벡터를 만들고 있다는 얘기를 통해 AE 청킹 방법에 대해 대화를 나누고 그 방법이 적용이 잘되면 정말 효과적일 것 같다라는 답을 받았다 청킹 사이즈를 통해서 문맥을 담는 방법을 알 수 있어서 청킹 크기 별로 실험을 해보면서 결과를 보아야 할 것 같다.

(Q) 현재 프로젝트 진행중에

1. 너무 많은 데이터로 인한 클러스터링 과정에서 소요되는 시간 문제
2. 데이터에 대한 피처를 조금만 크게 잡아도 나타나는 메모리 문제

가 있습니다. 혹시 이럴 때 어떻게 진행을 하면 효율적으로 진행할 수 있을까요?

[데이터 량이 방대한 반면 그래도 어느 정도의 데이터는 포함하고 실험을 해야 하는 상황이라 시간과 컴퓨터 메모리와의 싸움이 있었다. 그에 대한 현직자 분의 조언을 얻고 싶었다.]

(멘토) 일단 너무 많은 데이터로 인한 시간소요 문제에 대해 말해보자면 우리가 프로젝트를 진행하면서 처음부터 프로젝트 마지막에 쓸 결과를 보는 것이 아니라 단계 단계 진행하면서 나오는 결과를 검증하는 것이 우선일 것입니다. 따라서 전체 데이터에 대해서 실험을 하는 것이 아닌 샘플링을 통해 데이터 개수를 줄여 놓고 먼저 만든 클러스터링 알고리즘이나 자연어 처리에 대한 효율성을 점검해 가면서 나갈 필요가 있습니다. 즉, 각각 해 나아갈 때 중간과정에서 결과를 볼 수 있을 만큼의 데이터만 샘플링 해서 시간을 줄여서 실험을 하는 것이 효과적일 수 있습니다.

	메모리 문제에 대해서는 사실 로컬에서 실험을 하다 보면 어쩔 수 없는 문제이기도 합니다. AWS 에 sagemaker 에서는 데이터를 맞춰 넣어주면 거기에서 클러스터링을 해주기 때문에 가능하다면 AWS 를 써보는 것도 방법 중 하나일 것 같습니다. 아니면 만약에 TensorFlow 를 사용 가능하다면 TensorFlow 를 통해 API 식 접근으로 분산처리를 통해 좀 완화 될 수 도 있는데 이 방법은 조금 과한 것 같군요 일단 먼저 말했던 샘플링 처럼 데이터 개수를 줄여서 메모리 할당량을 낮춰보는 방법이 좋을 것 같습니다. => 앞으로 실험은 어느정도 기간이나 기관을 특정 지어서 적은 량의 샘플을 사용해서 진행해 보고 이렇게 했을 때 시간이 너무 걸리면 AWS 를 알아보기로 하였다. 일단 학교에서 AWS 를 지원해주고 있기 때문에 해당 부분에 대해서 한번 알아보면 좋을 것 같다. (Q) 기사를 유사도 기반으로 분류하셨을 때 기준 기사와 분류 대상 기사의 유사도 임계치를 정하셨을 텐데 이때 어떤 방법으로 임계치를 얻으실 수 있었나요? [현재 다른 목표에서도 임계치로 인한 골치를 얻고 있어서 어떤 상황에선 어떻게 만들어야 하는지 조언을 얻고자 함] (멘토) 임계치는 데이터를 하나씩 넣어보면서 제대로 분류가 되나 안되나 확인해가면서 기준을 잡았습니다. 결국 사람이 직접 판단을 한 것이죠 => 데이터 특성상 사람이 쉽게 분류할 수 있는 것에 대해서는 직접 참여를 통해 쉽게 분리가 가능하다고 하심. (Q) 분류 대상이 완벽하게 나눌 수 없는 데이터이다 보니 만든 알고리즘으로 100% 분별에 대한 확신성이 없습니다. 그래서 현재 임계값에 따라 FN 와 FP 사이의 Trade-off 관계에서 딜레마에 빠졌는데 이 경우에 어떤 식으로 진행하는 것이 좋을까요?? [여러 상황에 통용될 수 있는 임계치를 정하고 있기 때문에 오탐과 미탐에 대한 수용 가능 한 수치를 정하기 애매했음 회사나 현직 프로젝트적 시선으로 봤을 때 어떤 것에 대해 초점을 맞춰야 할지 질문을 드림] (멘토) 음.. 사실 이 문제는 우리가 항상 고민하는 부분입니다. 그래도 현재 프로젝트에서는 이 과정이 결과 나타나는 것이 아닌 결과를 만들기 위한 중간 처리 과정이기 때문에 마지막 처리 과정에서 좀 더 깨끗이 처리 할 수 있도록 클린한 데이터가 많이 남는 쪽으로 방향을 잡으면 좋을 것 같습니다. 예를 들어 내부 외부 둘 다 조금씩 섞이게 될 바에는 내부는 완전히 깔끔하게 남기고 외부에는 조금 섞이더라도 한쪽이라도 클린하게 만들어 좋은 결과를 얻는 것이 좋을 것 같습니다. => 일단 미탐이 있더라도 오탐을 줄여서 한 분류에 대해서라도 완벽히 정제된 데이터를 얻어서 실험하는 것이 우선이라고 해 주셨음. 말씀에 따라 일단 한쪽 결과가 좀더 깨끗이 나올 수 있는 방법을 통해 결과를 얻고 그걸로 실험을 진행하면 좋을 것 같음.

개인 멘토링	▸ **[03/15] 1 대 1 미팅 - 윤명근 교수님** (Q) 현재 데이터에서 개발하는 알고리즘을 통한 목표를 어떻게 잡는게 좋을까요?? [알고리즘을 통해 잡을 수 있는 목표가 정탐/오탐에 대한 말끔한 분류, 이벤트에 대한 분류, 유사한 것들만 뭉쳐진 모습 등 여러 결과가 있어서 어떤 목표를 초점으로 잡고 개발하는 것이 좋을지 효율적인 개발을 위해 교수님께 여쭤봄] (A) 클러스터링 결과 정답률을 기준으로 하면 정탐/오탐 이분 분류기 때문에 목표에 안맞을 수 있음 일단 특정 서버와 특정 이벤트를 잡아서 클러스터링을 했을때 밀집도가 높게 나올 수 있도록 알고리즘을 만들어 볼것. => 비슷한 이벤트 끼리 높은 밀집도를 가질 수 있는 자연어 처리나 다른 청킹 방법을 찾아 적용해보기 (Q) 현재 IPS 로그 데이터로 이벤트들을 가지고 있는데 어느 정도의 피처까지 사용해서 만들어 보는 것이 좋을까요?? [기업에서 더 선호할 만한 상황으로 시작하기 위해 사용한 데이터 한정을 시키려고 했음.] (A) 일단 payload 위주로 만들어 보는 것이 좋을 것 같다. 그리고 처음에 시도해보는 거면 아예 다 버리고 TCP 페이로드인 애플리케이션 데이터만 사용해서 정말 유사한 패턴들을 모으는 것을 목표로 해보자. => 데이터 사용을 애플리케이션 데이터로 한정시키고 진행, 일단 같은 이벤트에서는 비슷한 애플리케이션 데이터의 특징을 보일 것이니 이렇게 모아 놓고 사용해보기, 아무래도 기업에서는 최소한의 데이터로 최대의 값을 만들어 내는 것이 메모리나 시간적으로 효율 적이니 교수님의 조언대로 최소화 시켜서 시작 해보자 ▸ **[03/17] 1 대 1 미팅 - 윤명근 교수님** (Q) 0~ 50 바이트 까지는 엔트로피의 분포가 정규분포를 따르지 않지만 51~1600 바이트의 길이를 가지는 페이로드의 경우 엔트로피의 분포가 정규분포를 따르고 있음 해당 특징을 이용하여 암호화 된 패킷이 있을 수 있는 각 바이트별 신뢰구간을 만들어 보았습니다. 수식 자체는 괜찮게 되었지만 이제 미탐을 하는 경우와 오탐의 경우가 조금 생겨서 상담을 하고자 합니다. 오탐이 생기는 경우(암호화 안된건데 암호화로 탐지하는 경우): TLS 통신 중 사전 정보교환인 change chipher 나 handshake 과정에서 파라미터는 필요정보인데 이 모두 암호화로 처리를 하는 경우가 생김 (길이가 길어질 경우 파라미터중 랜덤값이 있어 엔트로피를 높이고 있는걸로 확인) 미탐이 생기는 경우 (암호화가 된건데 암호화가 안되었다고 넘기는경우) : 페이로드 길이가 길어질 경우 엔트로피의 표준편차가 적게 나와 암호화가 되었더라도 우연히 중복 단어가 생기는 경우 3 시그마에 포함이 되지 않은 경우가 있음 (A) 오탐이 생기는 경우: 현재 식별된 오탐의 발생 중 80 퍼센트이상이 TLS 의 핸드쉐이킹과 암호 파라미터 교환 중에 생기는 걸로 파악 해당 페이로드에는 특정 시그니처가 있기 때문에 이 시그니처를 통해 화이트 리스트를 처리하기로 하자 미탐이 생기는 경우: 해당 사항에 대해서는 추후 변동 가능하지만 일단 우리의 목적은 식별 불가능한 페이로드를 제외하는 것이 목표이기 때문에 시그마값을 더 낮추어도 된다고 하셨다. 예를

	들어 모토로라 같은경우 6 시그마를 통해 불량률을 관리하는 것 처럼 과장해서 표준편차의 6 배 까지 걸어도 암호화 패킷을 더 줄일 수 있다면 그 정도로 하도록 말씀하셨다. =>일단 최대한 암호화 되지 않은 패킷을 만들어야 하기 때문에 임계치를 6 시그마 까지 낮게 잡아서 다음 실험을 진행하기로 하였다. 실험 결과 본 후 조금씩 줄여가거나 그대로 하거나 결정할 예정이다. 그리고 화이트리스트를 할까 말까 고민을 많이 했는데 교수님과 이야기를 통해 화이트리스트는 패킷 자체의 특별한 특징을 찾은 것이기 때문에 편히 사용해도 된다고 하여서 맘 편히 사용할 수 있을 것 같다.