



## ITL 사이버보안 연구보고서

이 보고서는 ITL 테크노트 사이트 있는 보고서이며, 문서로 허가되지 않고서는 다른 곳에 게재 할 수 없습니다.

### 제목: 머신 러닝과 정보보안

자주 방문하는 사이트에서 내가 좋아하는 상품들을 자동으로 추천해 주고, 주식 상품이 앞으로 어떤 흐름을 가지게 될 지 자동으로 분석해 주며, 질병 세포의 패턴을 분석해 빠르고 정확하게 환자를 판단하고 치료를 한다. 모 기업에서는 기존 직원들의 자기 소개서 어휘 선택 패턴과 실제 업무 성과를 분석해 자동으로 신입 사원의 미래 성과를 가늠해 선발에 활용한 사례도 있다. 한번쯤은 이러한 컴퓨터의 '지능'에 소름이 돋았던 경험이 있을 것이다. 이 모든 것을 가능하게 해 주는 분야가 바로 '머신 러닝'이다. 이러한 다재 다능한 능력에도 불구하고 아직 사이버 보안 분야에서는 머신 러닝이 크게 환영 받지 못하고 있다. 본 연구 보고서에서는 머신 러닝의 개략적인 이해와 함께 보안 분야에 머신 러닝을 적용해 성공한 사례와 적용 방법 등을 살펴본다.

작성자 및 [ITL](#)에 저작권이 있습니다.



# 머신 러닝과 정보보안

작성자: 서준석, nababora@naver.com

승인일자: 2016. 1. 11

## 요 약

머신 러닝(Machine learning)이란 단어 그대로 '기계가 스스로 답을 찾아낼 수 있도록' 만들어 내는 분야를 의미한다. 엄밀히 말하자면 컴퓨터가 주어진 데이터에서 의미 있는 정보를 자동으로 찾아낼 수 있도록 해 주는 모든 기술 영역을 일컫는다. 많은 분야에서 머신 러닝이 활약하고 있는 반면 국내 보안 시장에서 머신 러닝은 아직 생소한 분야이면서 많은 사람들에게 추상적인 하나의 마술과 같이 받아들여 지고 있다. 본 문서에서는 머신 러닝의 기술적 개념과 함께 정보보안의 각 영역에서 어떻게 머신 러닝이 활용 가능한지 사례를 통해 소개하고자 한다.

## 1. 머신 러닝 개요

머신 러닝(Machine learning)이란 단어 그대로 '기계가 스스로 답을 찾아낼 수 있도록' 만들어 내는 분야를 의미한다. 엄밀히 말하자면 컴퓨터가 주어진 데이터에서 의미 있는 정보를 자동으로 찾아낼 수 있도록 해 주는 모든 기술 영역을 일컫는다. 머신 러닝에 대한 본격적인 논의를 하기 전에 우선 사람들이 가장 많이 혼용해서 쓰는 세 단어인 머신 러닝, 딥 러닝(deep learning), 패턴 인식(pattern recognition)의 의미와 차이점에 대해 간단히 짚고 넘어갈 필요가 있다.

구글 트렌드 검색 서비스에서 머신 러닝(파랑), 패턴 인식(노랑), 딥 러닝(빨강) 단어를 검색한 결과 아래와 같은 그림을 확인할 수 있다. 초기에 머신 러닝과 패턴 인식 분야가 그 흐름을 같이 해 온 것에 비해 시간이 갈수록 머신 러닝에 비해 패턴 인식에 대한 사람들의 관심이 점차 시들고 있다. 딥 러닝의 경우 2013년을 기점으로 관심도가 빠르게 상승하고 있다.

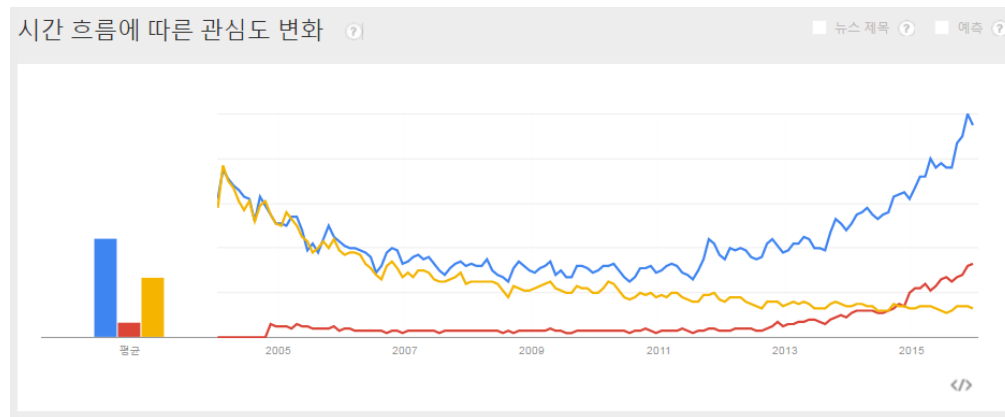


그림1. 구글 트렌드로 살펴본 머신 러닝 관련 용어의 검색 빈도

사실 이 세 분야는 동일한 기술적 원리를 배경으로 하며 분류에 큰 의미가 없다. 하지만 실제 적용되는 분야와 그 목표에 있어 다르게 해석될 수 있다는 점을 주목할 만 하다. 70년대부터 연구되어 왔던 패턴 인식 분야는 쉽게 말해 '어떻게 하면 컴퓨터가 사람의 인식 체계를 이해할 수 있을까?'에 대한 답을 찾는 분야라고 할 수 있다. 90년대 초기에 등장한 머신 러닝 분야의 근간을 이루는 기술적 개념들은 대부분 패턴 인식 분야에서 비롯된다. 하지만 패턴 인식이 주어진 데이터에서 패턴 자체를 찾아내는 것이 목표인 반면, 머신 러닝은 이러한 패턴 정보를 통해 사물을 분류 하거나 과거의 연속적인 패턴 데이터를 통해

미래의 패턴을 예측하는 것처럼 패턴을 이용해 특정 '작업'을 수행 하는 것을 목표로 한다. 마지막으로, 딥 러닝이란 머신 러닝의 한 영역으로 복잡하고 정교한 학습을 수행하는 분야라고 할 수 있다(딥 러닝은 아직 학문적 체계가 명확히 정립되지 않았다). 이 세 분야 모두 컴퓨터의 '판단 능력'과 관련된 기술이라는 점에서 공통점을 갖는다.

앞서 언급한 것처럼, 머신 러닝은 컴퓨터가 스스로 판단을 내릴 수 있게 해 주는 과정과 같다. 이미지에서 숫자를 찾아내거나, 사진에서 사람들의 얼굴을 자동으로 인식하고, 세포를 촬영한 사진 또는 영상에서 암세포로 추정되는 세포를 판별해 내고, 고객의 과거 제품 검색 유형을 분석해 관심을 가질 만한 제품을 추천해 주는 등 다양한 분야에서 머신 러닝이 사용되고 있다.

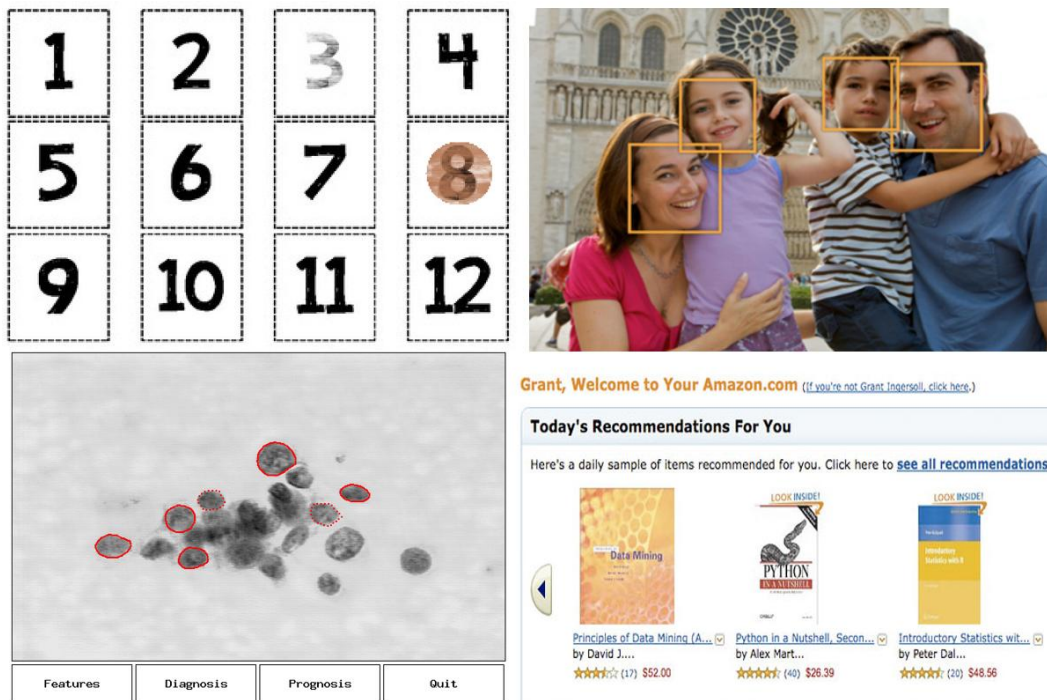


그림2. 머신 러닝 사례(왼쪽 상단부터 시계 방향으로 숫자 인식<sup>1</sup>, 얼굴 인식<sup>2</sup>, 종양 판단<sup>3</sup>, 아마존의 상품 추천)

앞서 소개한 네 가지 사례는 모두 공통적인 전제 조건이 필요하다. 바로 판단을 가능하게

<sup>1</sup> <https://mcdn1.teacherspayteachers.com/thumbitem/Number-Recognition-Memory-Game/original-399882-1.jpg>

<sup>2</sup> [https://developer.apple.com/library/mac/documentation/GraphicsImaging/Conceptual/CoreImaging/ci\\_detect\\_faces/ci\\_detect\\_faces.html](https://developer.apple.com/library/mac/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_detect_faces/ci_detect_faces.html)

<sup>3</sup> <http://pages.cs.wisc.edu/~street/saves/xcyt1.gif>

해 주는 '데이터'의 필요성이다. 인간이 손가락으로 밥을 먹고, 몸이 아플 때 약을 먹는 이유는 그렇게 '배워 왔기' 때문이다. 손가락의 형태가 음식을 먹기 편하고, 약 속에 있는 특정 성분이 통증을 완화시켜 준다는 것을 경험을 통해 학습했기 때문이다. 컴퓨터가 판단을 할 때도 이와 마찬가지로 판단의 근거가 되는 데이터(경험)들이 필요하다.

얼굴 인식 분야를 예로 들어 보자. 주어진 사진에서 사람의 얼굴을 판별해 내기 전에 우선은 어떤 형태가 사람의 얼굴인지 알고 있어야 한다. 일반적인 사람의 얼굴 형태와 눈, 코, 입의 위치, 비율 등 최대한 다양한 얼굴 사진들을 미리 확보해 사람의 얼굴만이 가지는 특징 정보들을 데이터화 해야 한다. 다음으로, 자동차, 동물의 몸, 옷과 같이 사람의 얼굴이 포함되지 않은 사진을 가져와 이전과 동일한 방식으로 이미지를 데이터화 한다. 충분한 양의 데이터가 확보된 뒤에는 이제 컴퓨터를 학습 시킬 차례다. 어린 아이에게 그림 놀이를 하며 사물의 이름을 가르치듯이 컴퓨터에게 얼굴 사진은 얼굴이라고, 그 밖의 다른 사진은 얼굴이 아니라고 가르쳐 주면 된다. 이 과정을 통해 컴퓨터는 사람의 얼굴을 찾아낼 수 있는 일종의 '인식 체계'를 갖추게 된다. 이러한 학습 과정을 끝마친 뒤에 새로운 사진을 컴퓨터에게 보여 주면 얼굴을 찾아낼 수 있다. 아래 그림은 이러한 과정을 절차에 맞게 체계화 한 것이다(이해를 돕기 위해 세부 절차는 생략한다).

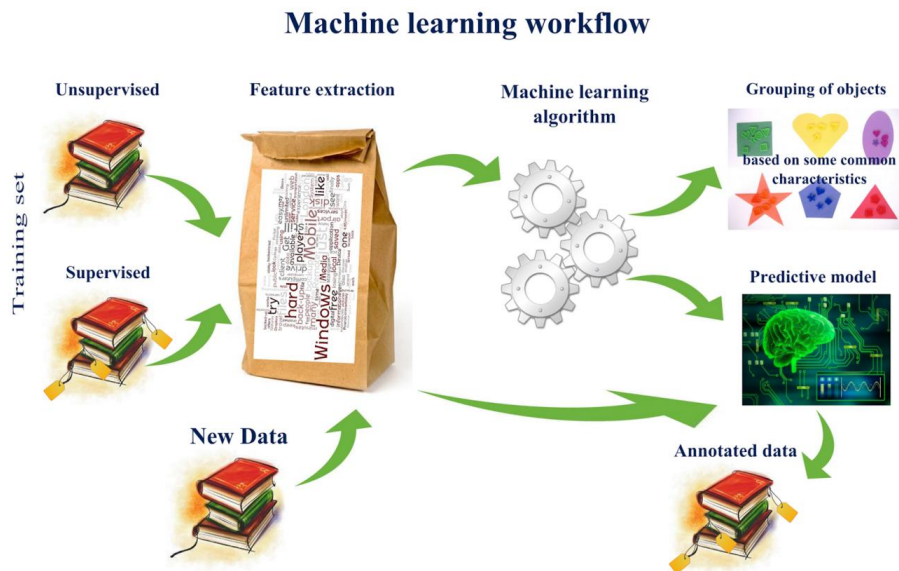


그림3. 머신 러닝 절차<sup>4</sup>

가장 먼저 학습에 사용할 데이터를 수집해야 한다. 여기에는 그림 파일, 이메일, 네트워크

<sup>4</sup> <http://www.computervisionblog.com/2015/03/deep-learning-vs-machine-learning-vs.html>

트래픽, 파일 시그니처 등 분석을 원하는 대상과 관련된 데이터가 모두 포함될 수 있다. 데이터는 최대한 많이 확보하는 것이 좋지만 무조건 많은 데이터가 좋은 성능을 보장하지는 않는다. 그림의 가장 왼쪽에 위치한 두 단어인 Unsupervised와 Supervised는 수집한 데이터의 특성과 밀접한 관련이 있다. 앞서 소개한 얼굴 인식 사례에서, 사람 얼굴이 확실한 사진과 그렇지 않은 사진을 모두 확보한 다음 컴퓨터를 학습 시켰다. 이는 Supervised<sup>5</sup> 데이터, 즉 정답과 오답 데이터를 모두 가지고 학습을 수행하는 경우를 의미한다. 이와 반대로, Unsupervised는 정답 데이터만 가지고 있는 경우를 의미한다. 이 경우 주어진 데이터만으로 학습을 해야 하는데 컴퓨터에게 오답은 어떤 것이라는 정보를 주지 못하는 관계로 Supervised인 경우와 컴퓨터를 학습시키는 방법이 조금 다르다.

학습을 위한 충분한 양의 데이터를 확보한 다음에는 데이터에서 의미 있는 정보를 추출하는 과정인 '특징 추출(feature extraction)' 과정을 거치게 된다. 특징 추출은 머신 러닝의 성능을 좌우할 수 있는 가장 중요한 단계로, 정답과 오답을 판가름할 수 있는 속성들을 선택해야 한다. 예를 들어, 얼굴 인식 예제에서 사람 얼굴을 좌표로 분할한 뒤, 눈, 코, 입의 상대적인 위치 또는 얼굴의 전체적인 비율을 특징으로 사용 가능하다(그림4).

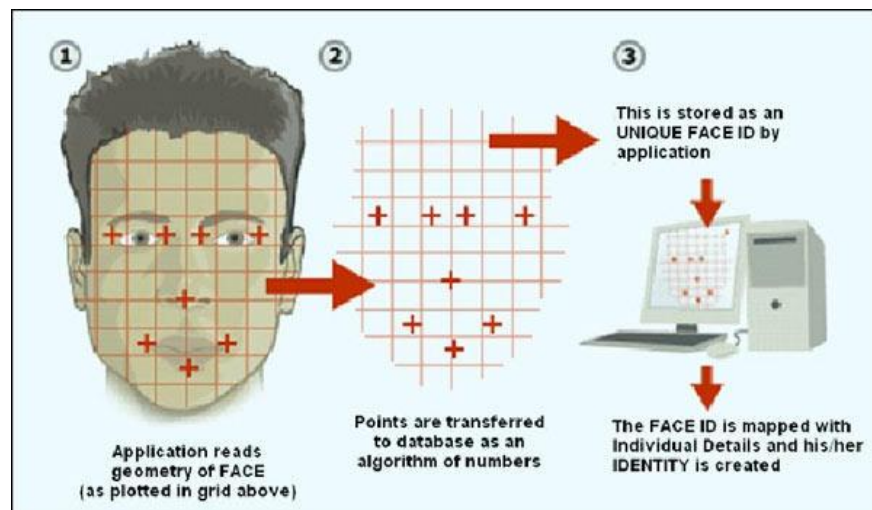


그림4. 얼굴 사진에서 특징을 추출하는 방법<sup>6</sup>

다음으로, 특징 데이터의 특성과 머신 러닝의 목적(분류, 예측 등)에 맞는 머신 러닝

<sup>5</sup> Supervise는 사전에서는 '감독하다'는 의미로 정의하고 있다. 또한, 위키피디아에서는 supervised learning을 지도 학습으로, unsupervised learning을 자율 학습으로 정의하고 있으나 출처마다 조금씩 한글 정의가 다른 관계로 영어 단어를 그대로 사용했다.

<sup>6</sup> <https://creativentechno.wordpress.com/2012/02/18/face-recognition/>

알고리즘을 사용해 컴퓨터를 학습 시킨다. 어느 정도 정형화 된 기법들이 존재하지만, 특징 데이터의 분포, 개수, 성격뿐만 아니라 학습 알고리즘의 종류 및 적용 방법에 따라 학습의 효율이 현저하게 차이가 날 수 있다.

마지막으로, 컴퓨터의 학습 성과를 판단하기 위해 컴퓨터에게 시험 문제를 낸다. 예를 들어, 사람의 얼굴이 포함된 사진 100장과 동물 사진 100장을 컴퓨터에게 제시하고 얼마나 사람 얼굴을 잘 찾아내는지 확인해 볼 수 있다. 만약 사람 얼굴을 잘 찾아내지 못한다면 다른 머신 러닝 알고리즘을 사용하거나, 그래도 문제가 해결이 안 된다면 특징 추출 혹은 데이터 수집 단계로 다시 돌아가 새롭게 시작해야 한다. 또 다른 예시로, 쇼핑몰 웹사이트에서 고객이 좋아할 만한 제품을 추천하는 시스템의 경우를 생각해 보자. 이 경우 과거의 기록을 토대로 추천한 제품이 실제로 고객이 관심을 가질 만한 제품들인 경우 컴퓨터가 성공적으로 학습을 했다고 간주할 수 있다.

지금까지 소개한 내용은 개략적인 머신 러닝의 과정을 설명한 것으로, 실제 시스템 구현은 보기보다 간단하지 않다. 특히 특징 추출 단계는 분석하려는 데이터의 특성에 대한 깊은 이해와 함께 의미 있는 정보를 데이터로 만들어 내는 능력이 필요하다. 또한, 최적의 성능을 보장하는 모델을 만들기 위해 오랜 시간을 들여 다양한 변수를 테스트 해야 하는데 이는 분석가의 이해와 경험이 부족하다면 쉽지 않은 여정이 될 것이다.

## 2. 머신 러닝과 보안

보안 분야에 머신 러닝 기법을 도입하려는 시도는 10여년 전부터 있었으나 큰 주목을 받지 못했다. 당시에는 지금처럼 악의적인 공격자들의 공격 패턴이 그리 다양하지 않았으며 전통적인 방식의 침입 탐지 및 공격 분석 시스템만으로 공격을 효과적으로 방어할 수 있었기 때문이다. 하지만 예측할 수 없는 공격자들의 다양한 변종 패턴과 넘쳐 나는 데이터의 양으로 인해 최근에 다시 머신 러닝이 주목을 받고 있다. 많은 사람들이 머신 러닝이 네트워크 침입 탐지 영역에서만 활용 가능한 것으로 알고 있지만 사실 그렇지 않다. 네트워크 침입 탐지뿐만 아니라 디지털 포렌식, 악성코드 분석, 취약점 분석뿐만 아니라 정보보호 정책 분야에서도 머신 러닝을 활용해 의미 있는 결과를 만들어 내려는 연구가 진행 중이다. 이번 섹션에서는 보안의 각 영역에서 어떻게 머신 러닝을 활용 가능한지 살펴본다. 필자 또한 머신 러닝의 전문가가 아니므로 세부 알고리즘에 대한 설명은 최대한 배제하고 알기 쉽게 개념적인 예시 위주로 내용을 소개하겠다.



## 2.1 네트워크 침입탐지

대부분 네트워크 침입 탐지는 네트워크 트래픽 모델을 기반으로 하는 네트워크 비정상 행위(network anomaly) 탐지 방식을 사용한다. 비정상 행위 탐지는 unsupervised 학습의 한 종류로, '정상'으로 간주되는 트래픽의 분포를 분석해 새로운 데이터가 정상 데이터에서 얼마나 '벗어나 있는지(outlier)' 탐지하는 방식을 사용한다. 일반적인 업무용 네트워크를 오가는 트래픽을 모두 수집한 뒤 IP 주소, 헤더 정보, 트래픽의 길이 등을(특징) 분석해 시각화 해 보면 보통 일정한 데이터 분포가 형성되는 것을 확인할 수 있다. 이 때, 순수하게 정상적인 트래픽만 모아 시각화 하더라도 분포의 중심을 벗어나는 데이터들을 발견할 수 있는데, 경계선의 범위와 모양을 어떻게 잡는지에 따라 비정상 트래픽 탐지 결과가 달라질 수 있다(그림5 참조).

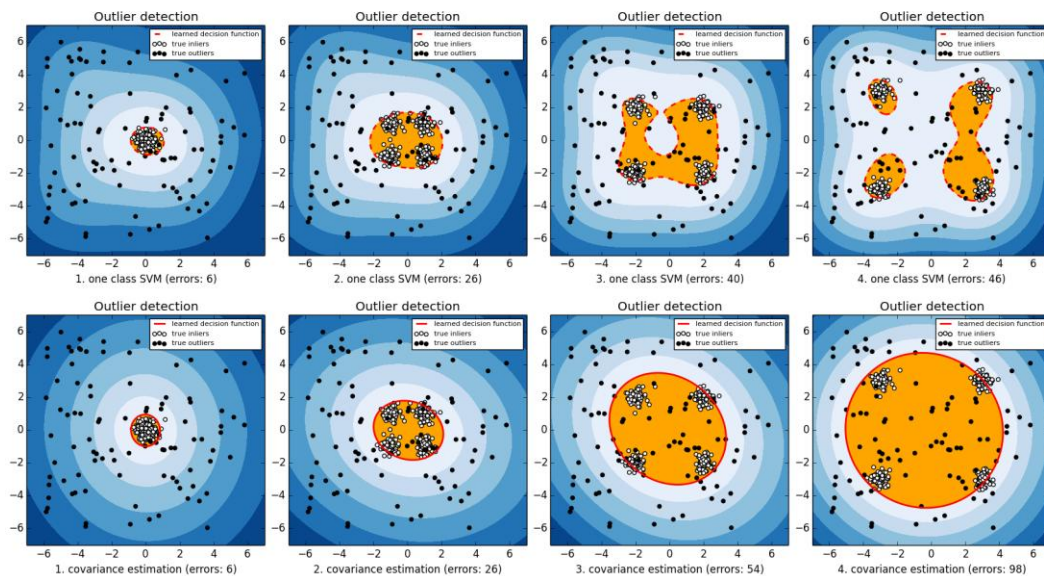


그림5. 아웃라이어(outlier) 분석 예시<sup>7</sup>(그림에서 보이는 주황색 음영 영역 바깥에 위치한 검은 점들을 비정상 트래픽으로 간주할 수 있다)

침입 탐지 시스템의 경우 높은 탐지율을 보장하더라도 내부 시스템에 치명적인 영향을 주는 단 하나의 트래픽을 잡아내지 못하면 문제가 될 수 있다(이 부분은 영원히 해결하지 못할 미제가 아닐까 생각한다). 머신 러닝 방식은 기존의 시그니처 방식의 탐지 기법과 달리

<sup>7</sup> <http://www.hongyusu.com/programming/2015/10/10/novelty-detection/>



그 근본 개념이 확률을 기반으로 하는 탓에 문제 발생 시 책임 소재가 모호해 질 수 있다. 뿐만 아니라, 기계가 자동으로 트래픽 패턴을 학습하고 지속적으로 성능을 개선해 나간다는 장점이 있지만 전통적인 방식에 비해 관리가 어렵고 시스템에 문제가 발생하더라도 그 원인을 쉽게 찾아내기 힘들다는 단점이 있다.

## 2.2 악성코드 분석

악성코드 분석 기법에는 크게 정적 분석과 동적 분석 방식이 존재한다. 지금까지 수많은 분석가들이 효율적이고 정확한 악성코드 분류 및 분석 방법들을 제시해 왔다. 머신 러닝과 결합한 연구 또한 다양한 성과를 만들어 내고 있으며, 본 문서에서는 머신 러닝을 이용한 행위 분석 예시를 하나만 소개한다<sup>8</sup>.

아무리 복잡한 난독화 알고리즘과 정교한 구조를 악성코드라도 결국 시스템 파괴, 정보 탈취와 같이 특정 목적을 가지고 있다. 샌드박스 환경에서 악성코드를 실행한 뒤 행위 정보를 수집해 악성코드의 특성을 파악하는 방식에 머신 러닝을 결합하면 재미있는 결과를 얻을 수 있다. 단순히 특정 시스템 콜의 호출 및 작업 수행의 유무로 악성코드를 분류하고 분석하는 것이 아니라 이러한 정보들을 데이터화 한 뒤 벡터 공간에 그린 후 군집화(clustering)와 분류(classification) 기법을 이용해 특성을 분석할 수 있다. 쉽게 설명하자면 군집화를 통해 판단을 원하는 대상 파일이 대표적인 악성코드의 행위 유형(prototype)에서 얼마나 벗어나 있는지 판단하거나, 머신 러닝을 통해 만들어 둔 악성코드 분류 모델에 대상 파일의 행위 정보를 대입해 악성코드 유형을 판단하는 것도 가능하다.

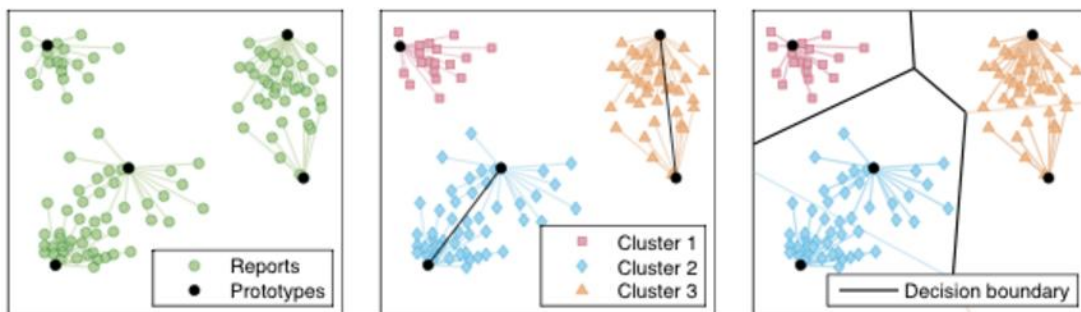


그림6. 군집화와 분류 기법을 이용한 악성코드 행위 분석 예시

<sup>8</sup> Automatic Analysis of Malware Behavior using Machine Learning

## 2.3 소프트웨어 취약점 분석

취약점 분석과 머신 러닝을 결합하려는 연구 또한 다양하게 진행돼 왔다. 초기 연구들은 NVD 및 EDB와 같은 공개 데이터베이스의 정보를 이용해 머신 러닝을 수행했다. 하지만 대부분 연구들이 실제 소프트웨어에 존재하는 취약점을 찾는 것보다 익스플로잇의 사용과 배포 등과 관련된 타임 라인 분석과 취약점 연관 키워드 분석에 초점을 맞췄다. 구문 트리(syntax tree)를 활용해 프로그래밍 패턴을 식별하고 이를 바탕으로 머신 러닝을 수행해 새로운 취약점을 자동으로 찾아내는 실용적인 연구 또한 선행된 바 있다(분석을 수행한 결과 제로데이 취약점을 발견했다고 한다)<sup>9</sup>.

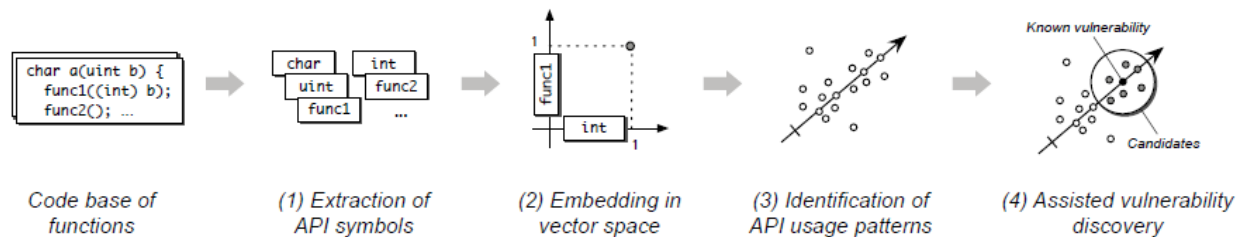


그림7. 취약점 추론(vulnerability extrapolation) 방법 흐름도

이전 섹션에서도 설명한 것처럼, 머신 러닝의 높은 정확도를 위해서는 충분한 양의 데이터가 수집돼야 한다. 하지만 소프트웨어 취약점의 경우 공개 데이터베이스를 통해 수집할 수 있는 정보에 한계가 있다. 또한, 컴파일된 바이너리를 분석하는 경우 프로그래밍 언어와 코딩 방식, 컴파일러의 종류에 따라 많은 변수가 존재하는 관계로 실제로 머신 러닝을 이용해 자동으로 소프트웨어에 존재하는 취약점을 찾는 것은 결코 쉬운 문제는 아니다.

## 2.4 디지털 포렌식

디지털 포렌식 영역과 머신 러닝을 결합한 재미있는 두 가지 사례를 소개하고자 한다. 첫째, 머신 러닝을 이용해 파일 조각만으로 해당 파일이 어떠한 형식을 가지고 있는지 판별해 내는 연구<sup>10</sup>가 있다. 파일 카빙 과정에서 대상 파일의 시그니처가 조각나 있는 경우

<sup>9</sup> Vulnerability Extrapolation: Assisted Discovery of Vulnerabilities using Machine Learning

<sup>10</sup> CarveML: application of machine learning to file fragment classification

파일의 확장자를 찾아내는 것이 쉽지 않다(물론 시그니처 기반이 아닌 다른 특성 정보를 이용해 파일 카빙이 가능하다). 물론 궁극적으로는 카빙으로 복구해 내지 못하는 파일을 살려 내는 것이 더 의미가 있겠지만 파일의 유형을 찾아내는 것만으로 분석에 충분한 도움을 줄 수 있다. 둘째, 방대한 양의 데이터에서 디지털 증거를 수집해야 하는 경우에 머신 러닝 기법을 활용 가능하다. 대용량 데이터를 그대로 사용하는 것이 아니라 데이터를 가장 잘 대표할 수 있는 특징 정보를 추출하거나 데이터를 변환하는 과정을 통해 부하를 줄인 뒤 적절한 학습 알고리즘을 사용해 빠르고 정확하게 유효한 정보를 판단할 수 있다.

디지털 포렌식 분야에 머신 러닝 기법을 적용하기 전에 주의해야 할 점이 있다. 의미 있는 모델을 도출하고 이를 통해 증거를 효과적으로 수집할 수 있다고 해도 분석 과정에서 사용한 모델과 데이터의 신뢰성을 꼼꼼하게 확인해야 한다. 분석 과정에서 정보가 왜곡되거나 분석 결과의 정확도가 낮은 경우 증거 자료로 신뢰할 수 없다는 점을 반드시 명심해야 한다. 직접 판례를 찾아보지는 않았지만 분석을 통해 얻어낸 증거 판단 모델이 99.9%의 정확도를 보장한다고 하더라도, 0.1%의 오차가 재판에서 인정될 수 있을지는 생각해 봐야 할 문제인 듯 하다.

## 2.5 정보보호 정책

언뜻 보기에는 정책과 머신 러닝이 전혀 조화가 되지 않는 이질적인 분야처럼 보인다. 대부분의 업무가 법률과 문서로 표현되는 영역인 관계로 의미 있는 데이터를 도출해 내기도 힘들뿐더러 복잡한 계산보다는 직관과 사람의 판단이 중요시되는 정책의 특성 상 머신 러닝이 끼어들 틈이 그리 많지는 않다. 하지만 불가능한 것은 아니다.

2012년 발표된 논문<sup>11</sup>에서는 웹 사이트에서 제시하는 개인 정보 사용 동의서<sup>12</sup>와 관련해 재미있는 연구를 수행했다. 일반적인 사용자들은 회원 가입을 하거나 개인 정보 제공이 필요한 서비스를 이용 시 업체에서 제공하는 개인정보 수집·이용에 관한 동의서와 같이 개인 정보 활용 범위와 법적 근거에 대한 장문의 안내서를 제대로 읽지 않는다. 솔직히 꼼꼼히 내용을 읽어 내려간다고 해도 그 의미를 이해하기는 쉽지 않다. 해당

<sup>11</sup> A Machine Learning Solution to Assess Privacy Policy Completeness

<sup>12</sup> 외국의 사례를 실험한 것으로, 정확하게는 privacy policy를 연구한 것으로 국내의 경우 개인 정보 사용 동의서와 유사하다고 판단해 이해를 돕기 위해 개인 정보 사용 동의서로 표현했다.

논문에서는 개인 정보 사용 동의서의 문장들을 분석해 사용자가 알아보기 쉽게 체크리스트 형식으로 표현해 냈다(그림8).


물론 법과 관련된 문장의 경우 문장이 내포한 법적 의미까지 100% 정확하게 분석해 내지는 못하지만 머신 러닝 기법을 이용해 사용자가 간단하게 확인하고 직접 판단할 수 있는 시각적인 결과물을 제공한다는 점에서 큰 의미가 있다고 본다.



그림8. 프라이버시 정책 문서를 자동으로 분석해 사용자에게 시각적으로 보여 주는 크롬 확장 도구 예시

### 3. 결론


지금까지 머신 러닝의 개념, 절차와 함께 정보 보안의 각 영역에 머신 러닝을 적용한 사례들을 함께 알아보았다. 사람들의 이해를 돕기 위해 최대한 원리와 개념 위주로 설명한 것일 뿐 실제로 머신 러닝 시스템을 구현하는 것은 그렇게 간단한 일이 아니다. 필자 또한 머신 러닝의 전문가가 아닌 관계로 끊임없이 연구에 임하고 있다. 하지만 개인적으로 머신 러닝 영역은 이론에만 통달한다고 전문가가 아니라고 생각한다. 다양한 기법과 원리들을 적재적소에 활용하는 능력 또한 중요하며 이것은 오로지 다양한 분석 경험을 통해 얻을 수 있다고 믿는다. 앞으로 국내 보안 분야에서도 머신 러닝을 이용한 참신한 성과가 많이 나오길 기대해 본다.

	<b>2016년 ITL 교육행사 일정</b> 전체 행사일정보기: <a href="http://www.itlkorea.kr/ITG/">http://www.itlkorea.kr/ITG/</a>
---	--

## 라이브 ITL 교육행사

<a href="#">Cyber Break Asia 2016</a>	서울, 잭팟(ZagPot)	2월 17일 ~ 19일	<a href="#">M200</a> , <a href="#">M230</a>
<a href="#">ITL AppSec 2016</a>	서울	4월 19일 ~ 22일	<a href="#">M250</a> , <a href="#">M330</a>
Network Security 2016	판교	5월 26일 ~ 27일	<a href="#">M130</a> , <a href="#">M220</a>
Cyber Security Summer 2016	서울	8월 24일 ~ 27일	<a href="#">M700</a> , <a href="#">M300</a> , <a href="#">M330</a>
ITL Capital Defense 2016	대전	11월 3일 ~ 4일	<a href="#">M220</a> , <a href="#">M250</a>
<a href="#">인하우스 교육</a>	원하는 장소	선택가능	<a href="#">과정 선택</a>



## 라이브 SANS 교육행사

 <a href="#">SANS Seoul 2016</a>	서울	6월 20일 ~ 25일	<a href="#">SEC560</a>
SANS Korea 2016 (TBD)	(TBD)	(TBD)	(TBD)

## 2016년 ITL 아카데미

2016-1기 아카데미 [사이버포렌식전문가]	서울	7월 11일 ~ 14일	M300, M230
2016-2기 아카데미 [침투시험전문가]	서울	11월 23일 ~ 26일	M220, M250
2016-1기 아카데미 [시큐리티 리더쉽]	서울	12월 12일 ~ 14일	M100, M700

## CyberForce::사이버보안 역량평가

 <a href="#">개인 역량평가(CyberForce-P)</a>	원하는 장소	원하는 일정	100문제
 <a href="#">조직 역량평가(CyberForce-G)</a>	원하는 장소	원하는 일정	50문제 (도메인별)