Journal of Korea Multimedia Society Vol. 22, No. 2, February 2019(pp. 203-209)
https://doi.org/10.9717/kmms.2019.22.2.203

기계학습 기반의 실시간 악성코드 탐지를 위한 최적 특징 선택 방법

주진걸†, 정인선††, 강승호†††

An Optimal Feature Selection Method to Detect Malwares in Real Time Using Machine Learning

Jin-Gul Joo†, In-Seon Jeong††, Seung-Ho Kang†††

ABSTRACT

The performance of an intelligent classifier for detecting malwares added to multimedia contents based on machine learning is highly dependent on the properties of feature set. Especially, in order to determine the malicious code in real time the size of feature set should be as short as possible without reducing the accuracy. In this paper, we introduce an optimal feature selection method to satisfy both high detection rate and the minimum length of feature set against the feature set provided by PEFeatureExtractor well known as a feature extraction tool. For the evaluation of the proposed method, we perform the experiments using Windows Portable Executables 32bits.

Key words: Malware, Machine Learning, PEFeatureExtractor, Feature Selection, Real Time Detection, Intelligent Classifier

1. 서 론

네트워크 인프라의 발전과 멀티미디어 응용프로그램이 네트워크를 이용해 성공적으로 사용되면서 멀티미디어 산업 분야에 큰 효율성과 다양성을 부여하고 많은 이득을 가져왔다. 하지만 이와 같이 눈부신 멀티미디어 기술의 성공 이면엔 개인의 호기심이든 경제적 이득이든 다양한 이유로 네트워크 인프라 및 응용프로그램에 손실을 야기하는 다양한 악성코드의 개발도 지속되어 왔다. 실제 보안침해 사고로 인한 경제적 피해 규모는 자연재해로 인한 피해 규모를 넘어서 있다. 특히 악성코드를 이용한 보안침해 사고는 사건의 발생 빈도뿐 아니라 기술적 진보 측면에서도 다른 ICT 기술에 발전 속도에 비례해 끊임없이 발전하고 있다.

악성코드란 컴퓨터, 서버, 클라이언트 및 컴퓨터 네트워크에 피해를 야기할 목적으로 의도적으로 설계된 소프트웨어를 말한다[1]. 실행 코드, 스크립트, 액티브 콘텐츠 등 다양한 형식을 띄고 있으며 컴퓨터 바이러스, 웜, 토로이 목마, 스파이 웨어, 에드웨어를 포함해 최근에 여러나라에 걸쳐 많은 피해를 가져온 랜섬웨어까지 다양한 이름으로 불리는 코드들을 포괄한다. 악성코드의 배포형식도 다양하여 이메일과 같은 미디어 콘텐츠에 악성 문서파일을 첨부해 중요

※ Corresponding Author : Seung-Ho Kang, Address: (58245) Gunjae Road 185, Naju-si, Jeonna, TEL : +82-61-330-3953, FAX : +82-61-330-3959, E-mail : drminor@dsu.ac.kr
Receipt date : Jan. 10, 2019, Approval date : Jan. 30, 2019
† Dept. of Civil Eng., School of Engineering, Dongshin University
(E-mail : jgjoo@dsu.ac.kr)
†† School of Electronics and Computer Eng., Chonnam National University
(E-mail : jis0755@gmail.com)
††† Dept. of Information Security, College of Energy Convergence, Dongshin University
※ This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2017R1C1B1008998).

자료를 빼내는 해킹사고를 유발하기는 방법도 있다[2].

이러한 악성코드에 대한 대응책으로 여러 가지 방법들이 제시되었지만, 그중에서 코드의 시그니처를 기반으로 블랙 리스트 혹은 화이트 리스트를 작성하고 이를 이용해 악성코드를 탐지하는 방법이 주로 사용되었다. 하지만 이러한 기법은 기존에 알려진 악성코드를 탐지하는 데는 적합하지만 새롭게 등장한 악성코드를 탐지하는 데는 한계가 있다. 한편 역공학이 제시하는 다양한 분석 방법을 이용해 코드의 악성 여부를 전문가가 직접 밝혀내는 방법이 있다. 이는 악성코드를 가장 정확하게 분석하는 방법이지만 역공학 분야의 전문적인 지식을 가진 전문가를 필요로 할 뿐 아니라 코드를 분석하는데 많은 시간이 요구되어 대량의 악성코드를 실시간에 분석하기는 어렵다는 단점이 있다.

최근에 기계학습을 비롯한 인공지능의 괄목할 만한 발전에 힘입어 기계학습의 다양한 방법들을 악성코드 탐지에 적용하려는 다양한 연구들이 시도되고 있다[3-6]. 시그니처를 이용한 방법이나 전문가의 정적, 동적 분석이 갖는 한계를 극복하고자 기계학습의 학습 기능을 활용하려는 노력이다. 기계학습을 이용해서 학습을 시키고 악성코드를 판별하기 위해서는 적합한 기계학습의 선택 못지않게 사용하는 특징 집합이 중요한 의미를 가진다. 즉, 기계학습을 사용하는 분류기의 성능이 코드를 대표하는 특징 집합에 크게 의존하므로 악성코드와 정상코드를 구분하는 능력이 탁월한 특징들을 추출하고 선택해야 한다. 하지만 일반적인 분류 문제와는 달리 바이너리로 되어 있는 코드로부터 악성과 정상을 구분 짓는 적절한 특징을 추출하는 문제는 직관적이지 않다. 일반적으로 사용되는 특징으로는 바이너리로부터 직접 얻는 n-gram을 사용하거나 파일 포맷의 규칙에 따른 헤더 정보나 파일 크기, 링커 버전 등의 메타 정보를 사용한다[7].

한편, 사용하는 특징의 종류 못지않게 중요한 사항은 특징 집합의 크기이다. '차원의 저주'라는 말이 의미하듯 지나치게 많은 특징들을 사용하는 경우 많은 학습 시간과 판별 시간을 소요하게 되므로 실시간을 요구하는 악성코드 판별의 경우 이는 특히 문제가 된다. 따라서 탐지율을 떨어뜨리지 않으면서도 가능한 한 크기가 작은 특징 조합을 선택하는 방법이 중요하다.

일반적으로 특징 선택을 해결하기 위한 접근 방법으로는 필터 방법(filter method)과 래퍼 방법(wrapper method)이 있다[8]. 필터 방법은 개별 특징과 라벨과의 관계성을 상관계수나 정보 이론적인 수치로 측정하여 순위를 매긴 후 주어진 임계치를 기준으로 특징들을 선택하거나 제거하는 방법이다. 필터 방법은 계산량이 적어 속도가 빠르다는 장점이 있으나 상관성이 높은 비슷한 특징들이 중복해 사용될 수 있고 특징 조합이 주는 창발적(emergent) 특성을 무시하게 되는 단점이 거론된다. 이에 반해 래퍼 방식은 사용하는 기계학습 방법의 성능을 직접 목적함수로 사용하여 이를 극대화하는 특징 조합을 해공간으로부터 탐색하는 방법을 사용한다. 단점으로는 주어진 특징 집합으로부터 지수승개의 가능한 특징 조합을 대상으로 탐색을 해야 하므로 많은 시간이 걸린다는 점과 사용하는 기계학습 방법에 과적합 문제를 발생시킬 수 있다는 점이 거론된다. 하지만 필터 방법에 비해 명시적으로 분류기의 성능을 사용한다는 점에서 일반적으로 탐지율이 높은 것으로 알려져 있다.

최근 [9]는 이상 징후 탐지를 위해 기존의 래퍼 방법이 사용하던 분류기의 성능 대신 k-means 클러스터링의 클러스터링 정확성을 목적함수로 하는 한편 다목적 유전자 알고리즘을 이용해 높은 탐지율과 짧은 특징 조합이라는 두 가지 목적을 만족하는 특징 선택 방법을 제안하였다. 제안한 방법은 NSL_KDD 데이터 집합을 대상으로 훌륭한 성능을 보여주었다. 본 논문은 [9]가 제안한 접근법을 악성코드 탐지에도 성공적으로 적용할 수 있음을 보이고 Pearson 상관계수를 추가하여 이를 개선하고자 한다.

논문의 구성은 다음과 같다. 2장에서 특징 추출 방법을 설명한다. 3장에서는 다목적 유전자 알고리즘 기반의 특징 선택 방법을 제안하고, 4장에서는 서포트 벡터 머신(SVM)을 이용한 분류기를 설계한다. 5장에서는 각각 10,000개씩으로로 구성된 학습 데이터와 테스트 데이터를 이용해 성능을 평가하고 6장에서 결론을 맺는다.

2. 특징 추출

앞에서도 언급했듯이 바이너리로 되어있는 코드로부터 코드를 대표하는 특징들을 추출하는 일은 다

른 분류 문제와 달리 직관적이지 않다. 코드로부터 직접 얻는 n-gram을 비롯해 역공학을 이용한 정적 혹은 동적 분석 방법에서 얻어진 여러 정보들이 코드를 대변하는 특징으로 사용되고 있다[7]. 이 중 Hyrum Anderson에 의해 추진된 yourarespecial[10] 프로젝트에서 개발한 PEFeatureExtractor 툴은 Windows PE 코드를 대상으로 포괄적인 정적 분석 특징을 제공한다. 이 정적 특징들은 PE 파일의 파싱을 요구하지 않는 원시 특징(raw feature) 유형과 파싱을 요구하는 파싱 특징(parsed feature) 유형으로 구성되어 있다[11]. 아래 Table 1은 대표적인 원시 특징과 파싱 특징을 보여준다.

PEFeatureExtractor 툴이 제공하는 특징들의 종류는 총 2350가지이다. 논문이 사용하는 학습 및 테스트 데이터를 대상으로 PEFeatureExtractor 툴이 제공하는 2350개의 특징을 추출해 본 결과 이 중 504개만이 의미 있는 값을 가지고 있고 나머지 특징들은 모두 0값 만을 가지고 있음을 발견하였다. 따라서 PEFeatureExtractor 툴이 제공하는 특징 중 의미 있는 값을 가진 504개의 특징만을 기본 특징으로 사용한다.

다음 장에서 제안하는 특징 선택 알고리즘은 이들 504개로 구성된 특징 집합을 대상으로 최적 특징 조합을 선택하고자 한다. 504개로 구성된 특징 집합으로부터 선택 가능한 특징 조합의 수는 $2^{504}-1$이다. 따라서 모든 특징 조합을 대상으로 일일이 성능을 측정한 후 최적의 특징 조합을 찾는 일은 사실상 불가능한 일이다.

3. 특징 선택 알고리즘

기계학습을 기반으로 하는 악성코드 자동 분류기의 성능은 선택된 특징 조합의 구성 및 크기에 좌우된다. 본 장에서는 악성코드 분류기의 탐지율을 떨어뜨리지 않으면서도 선택된 특징 집합의 크기를 최소화하기 위해 [9]가 제안한 방법을 악성코드 탐지에 적합하게 수정하고 Pearson 상관계수를 이용함으로써 실시간성을 높이는 방법을 제안한다.

Table 1. Representative features of PEFeatureExtractor [11]

Feature type	Representative featurs
raw features	Byte histogram Byte entropy histogram Strings
parsed features	General file info Header file info Section info Imports info Exports info

3.1 다목적 유전자 알고리즘

유전자 알고리즘은 주어진 해 공간으로부터 최적해를 구하기 위해 사용되는 대표적인 탐색 알고리즘 중 하나이다. 특히 다목적 유전자 알고리즘은 두 개 이상의 목적을 동시에 만족시키는 파레토 최적인 해를 찾는 문제에 사용된다.

크기가 N인 주어진 특징 집합으로부터 선택 가능한 모든 특징 조합은 2^N-1개임을 알 수 있다. 이 중 최고의 탐지율을 보장하는 한편 가장 짧은 길이를 갖는 특징 조합을 찾는 문제가 본 논문의 최종 목적이다. 하지만 SVM과 같은 특정 분류기를 목적 함수로 사용하는 래퍼 방법의 경우 모든 특징 조합의 성능을 측정하는 일은 불가능한 일이다. 이러한 문제를 해결하기 위해 다양한 탐색 방법이 제안되어 왔으나 최근 [9]는 NSL_KDD 데이터를 대상으로 네트워크 기반의 이상 징후 탐지를 위해 다목적 유전자 알고리즘을 사용하여 좋은 결과를 얻었다.

3.1.1 특징 집합의 표현

크기 N인 주어진 특징 집합 $F = \{f_1, f_2, f_3..., f_N\}$으로부터 부분 집합에 해당하는 특징 조합 S는 아래와 같이 2진 문자열로 표현할 수 있다.

$$S = s_1 s_2 s_3 ... s_i ... s_N, \quad s_i = 0 \text{ or } 1 \tag{1}$$

i번째 특징 f_i가 선택되면 s_i는 1 값을 갖고 그렇지 않은 경우엔 0 값을 갖는다. 이와 같이 표현하면 PEFeatureExtractor가 제공하는 특징을 사용하는 경우 가능한 특징 조합의 개수가 $2^{2350}-1$개(모든 특징이 선택되지 않은 해는 제외)가 된다. 이 중 길이가 가장 짧으면서도 악성코드 탐지율을 최대로 하는 특징 조합 S^*를 찾는 것이 목적이다. 여기서 주어진 S에 대한 길이는 1의 개수로 정의된다.

3.1.2 목적함수

목적함수는 두 가지 목적함수 즉, 탐지율 관련 목

적함수와 특징 조합의 크기 관련 목적함수로 구성된다. 우선 탐지율 관련 목적함수 $O_d(S)$는 다음과 같이 정의된다.

$$O_d(S) = K\text{-Means\_Cluseter\_Accuracy}(S, T) / |T| \quad (2)$$

K-Means_Cluster_Accuracy()는 해당 특징 조합 S를 이용하여 라벨을 가진 훈련 집합 T를 대상으로 군집화의 정확도를 계산한 값이다. 예를 들어 정상코드 50개, 악성코드 50로 구성된 훈련 집합을 대상으로 특징 조합 S'이 K-평균 군집화 알고리즘을 사용해 80개를 정확히 원래의 자기 소속 군집으로 분류하였다면 이 값은 80이 된다. 그리고 사용한 K-평균 군집화 알고리즘의 초기 군집 중점은 랜덤하게 선정하였고 알고리즘의 반복되는 프로세스는 더 이상 중점의 변동이 없는 경우 종료하도록 하였다.

두 번째 목적함수인 크기 관련 목적함수는 다음과 같이 정의된다.

$$O_s(S) = (N - |S|)/N \quad (3)$$

따라서 주어진 특징 조합 S의 크기 관련 목적함수 $O_s(S)$는 S의 크기 $|S|$가 작을수록 1에 가까운 값을 갖는 반면, 특징 집합의 크기 N에 근접하면 0에 가까운 값을 갖는다.

최종 목적함수는 아래 식 (4)와 같이 위 두 목적함수의 가중치 합으로 정의된다. ω 값의 크기에 따라 탐지율에 관한 목적함수와 특징 조합의 크기에 관한 목적함수의 중요도를 적절히 선택하여 사용할 수 있다. 즉 ω가 1에 가까운 값을 갖도록 결정하면 특징 조합의 크기보다는 탐지율에 비중을 두고 목적함수를 선정하게 되고 0에 가까운 값을 선택하면 특징 조합의 크기에 비중을 두고 선정하게 된다.

$$O(S) = \omega * O_d(S) + (1-\omega) O_s(S), \ 0 \leq \omega \leq 1 \quad (4)$$

3.1.3 유전자 알고리즘 연산과 파라미터들

우선 유전자 알고리즘에서 사용하는 세대의 크기는 100으로 하였다. 즉 알고리즘이 진행하면서 생성하는 해집합의 크기가 100이라는 의미이다. 알고리즘의 반복 횟수는 40으로 고정하였다. 여러 번의 실험 후 결정한 것으로 40회 이상의 반복이 성능에 큰 변화를 가져오지 않음을 확인하였기 때문이다.

새로운 세대 생성에는 엘리티즘을 사용하여 이전 세대의 해 중 상위 20%에 해당하는 해는 다음 세대에 그대로 해로 사용하였고 나머지 80%의 해는 새로 생성하여 세대 구성에 사용하였다. 새로운 해를 생성하기 위한 부모 해 선택 방식에는 룰렛휠 방식을 사용하였다. 각 해가 갖는 목적함수 값에 비례하도록 두 해를 선택한 후 임의의 위치에서 이분하여 상호 교차해 연결하는 한 점 교배 연산을 이용해 새로운 두 해를 생성한다. 돌연변이율은 1%의 비율로 새로 생성된 자식 해들을 대상으로 적용하였다. 임의의 위치를 선정해 0이면 1로 1이면 0으로 변경한다.

Algorithm: Multi-Objective Genetic Algorithm
Input: Training data set with labels, Threshold T
Output: Feature Combination S_t

1. $t \leftarrow 1$
2. $S_t \leftarrow$ random sample(100) // Generate 100 random solutions
3. **while** $t < T$:
4. Calculate $O(S_t)$; // Calculate objective function for each solution
5. $t \leftarrow t+1$
6. $S_t \leftarrow$ Upper 20% of (S_t-1) // Select elites and insert them into the next generation
7. $S_t \leftarrow S_t$ + Roulette Wheel & Cross Operation (Lower 80% of S_t-1)
8. $S_t \leftarrow$ Mutation(S_t)
9. **return** S_t

3.2 Pearson 상관계수

아래 식 (5)로 정의되는 Pearson 상관계수 ρ는 두 변수(특징) 사이의 상관성을 나타내는 대표적인 통계치로 -1과 1사이의 값을 갖는다. 만약 상관계수가 -1값을 가지면 두 특징 X, Y의 관계는 역의 상관관계에 있다 하고 +1 값을 가지면 양의 상관관계, 0을 가지면 상관관계가 없다고 한다.

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \quad (5)$$

여기서, $cov(X, Y)$는 두 특징 X, Y의 공분산을 , σ_X는 X의 표준 편차를, 그리고 σ_Y는 Y의 표준 편차를 각각 나타낸다.

임의의 두 특징이 +1에 가까운 높은 양의 상관관계를 갖는다면 이는 악성코드의 판별에 있어 비슷한 영향을 미치는 특징들일 가능성이 높다. 따라서 이들 두 특징 중 하나의 특징만 탐지에 사용하더라도 전체

적인 탐지율에 미치는 영향은 작거나 거의 없을 것으로 생각된다.

따라서 모든 특징 사이의 Pearson 상관계수를 계산하고 사전에 정의한 임계치 이상의 상관계수 값을 보이는 특징 중 하나를 제거하면 탐지율에 변화를 거의 가져오지 않으면서도 전체 특징 조합의 크기는 줄어들어 탐지 효율을 높일 수 있다. 504개의 특징들을 대상으로 총 504*503/2 개의 Pearson 상관계수를 계산하고 이 중 0.7 이상인 경우 한쪽 특징을 제거하였다. 다시 말해 유전자 알고리즘에 의해 구해진 해를 대상으로 0.7 이상인 Pearson 상관계수를 갖는 특징들이 있으면 이 중 하나를 제거하였다. 아래 Table 2는 Pearson 상관계수가 0.7 이상인 특징들의 일부를 나타낸 것이다. 각 숫자는 특징에 해당한다.

Table 2. features with Pearson correlation coefficient threshold 0.7 for each feature

feature number	features with Pearson correlation coefficient greater than or equal to 0.7
44	526
48	124, 530
49	531
51	53, 55, 57, 533, 537
53	51,55,57,535
...	...

4. 악성코드 분류기

모든 특징에 식 (6)과 같은 표준 정규화(standard scaling)을 적용하였다. 이는 모든 특징들이 분류기에 미치는 영향을 동등하게 하기 위해서이다.

$$z_{ij} = (x_{ij} - \mu_i) / \sigma_i \quad (6)$$

여기서 x_{ij}는 특징값이고 μ_i는 i 번째 특징의 평균값, σ_i 는 표준 편차를 각각 나타낸다.

악성코드를 탐지하기 위해 사용한 분류기는 비선형 SVM을 사용하였다. SVM은 이진 분류기로써 다른 기계학습 방식에 비해 클래스 간의 최대 마진을 결정 초평면을 생성하는데 명시적으로 고려한다는 점에서 보다 안정적인 성능을 보장하는 기계학습 방식으로 알려져 있다. 한편 다차원의 복잡한 데이터들을 대상으로 분류 정확성을 높이기 위해 비선형 SVM을 사용하였으며 아래 식으로 대표되는 가우시안 Radial Basis Function(RBF) 커널을 사용하였다.

$$K(\gamma, x, l) = \exp(-\gamma \| x - l \|^2 / 2\sigma^2) \quad (7)$$

이때 r은 5로 설정하였다.

5. 성능 평가

위에서 설명한 특징 선택 알고리즘의 성능을 평가하기 위해 각각 10000개의 훈련 집합과 테스트 집합을 이용해 실험하였다.

5.1 데이터 집합

크기가 10000인 훈련 집합은 악성코드 7000개와 정상코드 3000개로 구성되어 있다. 테스트 집합도 동일한 구성과 크기로 되어있지만 훈련 집합과는 별개의 코드로 되어있다. 아래 Fig. 1은 사용한 코드의 파일 일부를 보여준다.

특징 선택 알고리즘에 의해 구해진 특징들만을 이용해 별도로 특징을 추출하였고 훈련 집합에 속한 코드들로부터 추출한 특징을 사용해 SVM 기반의 분류기를 학습하였다. 테스트 집합도 마찬가지 방법으로 특징을 추출하고 학습된 분류기를 사용해 예측치를 구하고 이를 실제 라벨과 비교하였다.

5.2 성능 분석

분류기의 탐지율 $Accuracy$은 아래 식 (8)에 의해 구한다.

$$Accuracy = 1 - \frac{\sum_{i=1}^{N} |pred(s_i) - label(s_i)|}{N} \quad (8)$$

```
3e63af019abcd2f15cd197882b0df457.vir  7cf699700b3672f7adff475f5a69ce0c.vir
3e6669be7e09d92eb3e78da70c12df03.vir  7cfd157d371044005a3a7ad4e758527f.vir
3e6b77a723543a6c67131afda0e08553.vir  7d053c91facb55e2d6db8942ce4e32ac.vir
3e85765d0db4396ab4eecc050b952c2d.vir  7d099ff16f227342b92db9c6dbb08642.vir
3e89b1e8f35d0ad4ecd874a053498bde.vir  7d1000fa5c4cba486752143802 5d1720.vir
3e8e90136a12b2339144136567678089.vir  7d14e89bd9c92f820bc71500848b03aa.vir
3e914b45b8a8c96083a0d22a27613 87f.vir  7d18ac9c1a1ac8067958686671eb8db5.vir
3e93b128e32e228940bce34b21043b14.vir  7d22fc369e71db6f641cd3901c231115.vir
3ea4db66e9ec86ff7386151c6c189699.vir  7d23fe44a75b2b0c0fb0c66c85719 58d.vir
3ea5913203e41834459b873fbc0609a2.vir  7d256401cceab22fde0367bdf2ccdda2.vir
3ebc4c5d19919f252baaeec8ef9dd44d.vir  7d299a82db9f6d89230af3d2d4d6dc85.vir
3edd8d3db6708467050c5873f98b77f7.vir  7d2e831a1a0b2425642ee58fe7d3df73.vir
3edd9d385350321483cff0d2ec48ac19.vir  7d32eac4617c067b3c8858e666924c27.vir
3edf4a9b65403aab49ef2a335d580bdf.vir  7d340a00d8fb5a892f86410db49add86.vir
3ef20ed57718b4e04538df4a3835301f.vir  7d3467db717a35d845976273e33fcb69.vir
3efc85fc4847f5f5d6c117bcaf14ab7f.vir  7d45c266a666136fcc6e9ec6f3104e27.vir
3f0aaa08fadaba774c77c82c0dcbc34e.vir  7d481ab710684907512099e9ec07bed6.vir
3f16c16dc9a39a09bc65cdb0e07c5123.vir  7d4dab8fb45a7558e428ccc54aef9f7a.vir
3f1ad6c0efa77988d9706a318960ff72.vir  7d5495487d852267ecd3c6dc0eb04cf7.vir
3f1faadb0e769c0b596a23ba6d2c6b9f.vir  7d559c7d466f3c66bc2bc6bf6a601140.vir
3f20f294cf3e557eb607d2e6d3e1a957.vir  7d6582875072c24d47adc475e98893d5.vir
3f277838f72825cf9eed3491c4b1fbfc.vir  7d681c1611f06f999bfd7a3e0536c2f4.vir
3f292c441baa074764309f892734f93d.vir  7d6e8958167e2a96c9763b6e60d7ae53.vir
3f2df19329920608739352c23d484cca.vir  7d7126161ea8b0d76f49dd788e6ed754.vir
3f349b119a13c9f1503d84e559632ae8.vir  7d728cc07ccbad54a30988a5e3cf778a.vir
3f34f721fa117281696d2a8be83eecbe.vir  7d74399c5a8e89591b65ac9ef59cf96c.vir
3f39b019483c22f704a8cbf7e3bba4e9.vir  7d7694b317ee3e71de97cdbeac7aab09.vir
3f3b40e01a08ef951e80572e118c8d3e.vir  7d7fafb4d4b8efb1dca9c20c2054ebce.vir
```

Fig. 1. Sample codes.

Table 3. Performance of feature selection algorithm

Size of feature set	Accuracy for training data set	Accuracy for test data set	Training time	Test time
2350	1.0	0.724	1036.96	299.01
504	1.0	0.737	229.05	70.21
217	0.998	0.749	89.44	27.58

여기서 $pred(s_i)$는 i번째 코드 s_i를 대상으로 분류기에 의해 결정된 악성코드 여부로 악성코드이면 1 아니면 0을 갖는다. $label(s_i)$는 s_i의 실제 라벨을 나타낸다. 그리고 N은 테스트 집합을 구성하는 코드들의 개수이다.

한편, 훈련 시간에는 데이터 스케일링과 실제 훈련에 걸리는 시간, 훈련 집합을 대상으로 정확성을 측정하는 시간까지 포함하였다. 그리고 테스트 시간에는 데이터 스케일링에 드는 시간과 정확성 측정 시간만을 포함하였다.

총 10번의 실험을 통해 정확성 및 시간, 특징 조합의 크기를 계산하고 평균하였다(Table 3). 특징 길이가 2350인 경우는 원래 PEFeatureExtractor 툴이 제공하는 모든 특징을 사용한 경우이고 504는 0 값만을 갖는 특징들을 제거한 후의 기본 특징 조합을 말하며 217은 다목적 유전자 알고리즘과 Pearson 상관계수로 구성된 특징 선택 알고리즘을 적용한 후의 특징 조합을 각각 말한다. 다목적 유전자 알고리즘에서 탐지율과 길이의 중요도를 나타내는 ω는 두 목적함수의 비중이 같도록 0.5를 사용하였다.

Table 3에서 알 수 있듯이 특징 선택 알고리즘을 적용함으로써 특징 조합의 길이가 기본 특징 길이 504에 비해 2배 이상 작은 217인 특징 조합을 얻을 수 있었다. 따라서 훈련 시간 및 테스트 시간도 그 이상 단축시킬 수 있음을 확인하였다. 특히 훈련 데이터를 대상으로 한 정확성은 원래 길이와 504개로 구성된 특징 조합을 이용했을 때 100%의 정확성을 보여줘 217개로 구성된 특징 조합의 정확성보다 0.2 % 정도 높았으나 테스트 데이터를 대상으로 실험 결과는 오히려 1% 이상의 정확성 향상을 보임을 확인하였다. 이는 많은 수의 특징이 훈련 데이터에 과적합 효과를 발생시켜 오히려 일반성을 상실함으로써 실제 테스트 데이터를 대상으로 좋지 않은 결과를 가져온 것으로 분석할 수 있다.

이러한 결과는 네트워크의 진입단에서 오고가는 미디어 콘텐츠에 포함된 모든 코드들을 대상으로 사전에 악성코드를 판별하려는 중앙 집중식 시스템의 경우 적은 수의 특징만을 사용해 빠른 시간에 보다 정확한 탐지를 할 수 있다는 점에서 중요한 연구 결과라고 할 수 있다.

6. 결 론

본 논문은 기계학습 기반의 지능형 악성코드 판별을 실시간에 가능하도록 하는 특징 조합을 선택하는 알고리즘을 제안하였다. 특징 선택 알고리즘은 다목적 유전자 알고리즘을 이용해 분류기의 탐지율과 길이 최소화를 동시에 목적으로 한 특징 조합을 선택하도록 하였다. 10000개의 훈련 코드와 10000개의 테스트 코드를 사용한 결과 선택 알고리즘을 사용하기 전보다 훈련 및 테스트 시간을 크게 줄일 수 있었으며 테스트 정확성도 작긴 하지만 높일 수 있음을 확인하였다.

다만, 테스트 데이터를 대상으로 한 분류기의 정확성이 약 75% 정도여서 이는 앞으로 개선의 여지가 크다. 이는 SVM이 아닌 다른 기계학습 방법을 이용하거나 PEFeatureExtractor 툴이 제공하는 특징 이외의 특징들을 포함해서 연구를 확장할 필요가 있다.

REFERENCE

[1] Wikipedia, https://en.wikipedia.org/wiki/Malware, (accessed Jan., 10, 2019).

[2] C.S. Park, “An Email Vaccine Cloud System for Detecting Malcode-Bearing Documents,” *Journal of Korea Multimedia Society*, Vol. 13, No. 5, pp. 754-762, 2010.

[3] Y. Elovici, A. Shabtai, R. Moskovitch, G. Tahan, and C. Glezer, “Applying Machine Learning Techniques for Detection of Malicious Code in Network Traffic,” *Proceeding of the IEEE Symposium on Annual Conference on Artificial Intelligence*, pp. 44-50, 2007.

[4] M.H. Nguyen, D.L. Nguyen, X.M. Nguyen, and T.T. Quan, “Auto-Detection of Sophisticated Malware Using Lazy-Binding Control Flow Graph and Deep Learning,” *Computers*

and Security Vol. 76, pp.128-155, 2018.

[5] C.I. Rene and J. Abdullah, "Malicious Code Intrusion Detection Using Machine Learning And Indicators of Compromise," *International Journal of Computer Science and Information Security*, Vol. 15, No. 9, pp. 160-171, 2017.

[6] P. Singhal and N. Raul, "Malware Detection Module Using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks," *International Journal of Network Security and Its Applications*, Vol. 4, No. 1, pp. 61-67, 2012.

[7] C.T. Lin, N.J. Wang, H. Xiao, and C. Eckert, "Feature Selection and Extraction for Malware Classification," *Journal of Information Science and Engineering,* Vol. 31, No. 3, pp. 965-992, 2015.

[8] G. Chandrashekar and F. Sahin, "A Survey on Feature Selection Methods," *Computers and Electrical Engineering*, Vol. 40, No. 1, pp. 16-28, 2014.

[9] T.H. Kim and S.H. Kang, "An Intrusion Detection System Based on the Artificial Neural Network for Real Time Detection," *Journal of Information and Security*, Vol. 17, No. 1, pp. 31-38, 2017.

[10] youarespecial, https://github.com/endgameinc/youarespecial, (accessed Jan., 10, 2019).

[11] C. Chio and D. Freeman, *Machine Learning and Security*, O'Reilly Media, Sebastpol, 2018.

주 진 걸

2003년 고려대학교 토목환경공학과(공학사)
2005년 고려대학교 사회환경시스템공학과(공학석사)
2011년 고려대학교 사회환경시스템공학과 (공학박사)

2011년~2012년 고려대학교 공과대학 연구교수
2012년~2014년 전북과학대학교 조교수
2014년~현재 동신대학교 조교수
관심분야 : 도시재해, 빗물펌프장 최적화, 비점오염원, 기후변화, 기계학습

정 인 선

2001년 여수대학교 전자계산학과 (이학사)
2006년 전남대학교 전산학과 (이학석사)
2011년 전남대학교 전산학과 (공학박사)

2017년~현재 전남대학교 박사후연구원
관심분야 : 생물정보학, 최적화 알고리즘, 인공지능

강 승 호

1994년 8월 전남대학교 전산학과 (이학사)
2003년 2월 전남대학교 전산학과 (이학석사)
2009년 8월 전남대학교 전산학과 (이학박사)

2013년 9월~현재 동신대학교 정보보안전공 조교수
관심분야 : 정보보안, 알고리즘, 기계학습