Feature Importance 분석 기반의 랜섬웨어 탐지 모델 성능 평가

전혜민[1], 최두섭[2], 임을규[2]
[1]한양대학교 정보보안학과
[2]한양대학교 컴퓨터·소프트웨어학과
crow0506@hanyang.ac.kr, dslab0915@hanyang.ac.kr, imeg@hanyang.ac.kr

A Comprehensive Performance Evaluation of a Ransomware Detection Model Based on Feature Importance Analysis

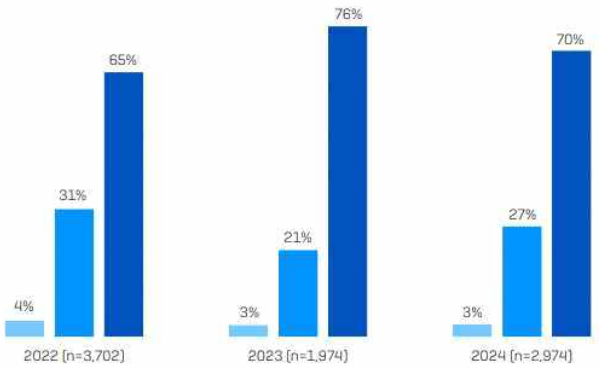
Hye-Min Jeon[1], Doo-Seop Choi[2], Eul Gyu Im[2]
[1]Dept. of Information Security, Hanyang University
[2]Dept. of Computer Science, Hanyang University

요 약

랜섬웨어는 공격자가 시스템 침투 후 데이터를 암호화하여 정상적인 접근을 차단하고 복호화를 위해 금전을 요구하는 악성 소프트웨어이다. 이러한 랜섬웨어 공격에 대응하기 위해 현재 많은 연구가 진행되고 있다. 기존 연구들은 랜섬웨어가 가진 다수의 특성을 활용해 탐지를 시도하나, 많은 특성 사용으로 인한 시간·자원 소모와 분류 정확도의 한계가 문제로 남는다. 이러한 문제를 해결하기 위해 본 논문은 Feature Importance 알고리즘 중 Gain Ratio를 활용하여 K값 기준으로 최적화된 데이터셋을 구성하고 이를 통해 머신러닝 모델을 학습하여 탐지하는 방법을 제안한다. 실험 결과, RF 모델은 K=0.3에서 99.41%, DT는 96.96%, SVM은 K=0.1에서 97.65%, MLP는 K=0.3에서 98.14%의 분류 정확도를 달성하였다.

1. 서론

랜섬웨어(Ransomware)는 공격자가 시스템에 침투하여 데이터를 암호화함으로써 정상적인 접근을 차단하고, 복호화를 위해 금전을 요구하는 악성 소프트웨어이다. 공격자는 이메일 피싱, 취약점 악용 등 다양한 기법을 활용하여 침투하며, 피해자는 암호화된 데이터를 복구하기 위해 상당한 비용을 부담해야 한다.



(그림 1) 2023년 랜섬웨어 공격 동향

Sophos의 2023년 『State of Ransomware』 보고서에 따르면, 2023년 한 해 동안 랜섬웨어 공격에 노출된 기업의 수가 전년 대비 약 35% 증가하였으며, 공격 기법 역시 점차 정교해지는 추세를 보이고 있다[1]. 보고서에서는 또한 랜섬웨어 변종의 다변화와 함께, 전통적인 보안 체계로는 탐지가 어려운 새로운 공격 방식이 등장하고 있음을 강조한다. 이러한 현실은 기업 및 개인 사용자가 지속적인 사이버 위협에 노출되어 있음을 말하며, 보다 효과적인 탐지 및 대응 전략의 필요성을 부각시킨다.

한편, 기존 랜섬웨어 탐지 연구들은 랜섬웨어가 가진 다수의 특성(feature)을 활용함으로써 어떤 특성이 효과적으로 탐지할 수 있는지 연구를 진행하고 있다. 랜섬웨어가 가진 다수의 특성 중에서 탐지할 때 중요한 특성을 알기 위해 Feature Importance 알고리즘을 이용하여 특성을 찾아낸다. 하지만 다수의 특성을 학습할 때 사용되는 시간과 자원이 많이 사용되기 때문에 연구를 진행함에 하나의 문제점으로 남는다. 그리고 특성 전부를 사용하여 학습한 후 분류 정확도에서 한계를 보이는 경우가 많았다. 본 논문에서는 이러한 문제점을 해결하기 위해, Feature Importance 알고리즘 중 하나인 Gain Ratio를 활용하여 K값을 이용해 새로운 데이터셋을 생성하고, 생

성된 최적화된 특성 집합을 구성하여 머신러닝 모델을 학습시키는 새로운 분류 방법론을 제안한다.

2. 관련 연구

Muhammad Ijaz 외 2인(2019)은 정적 분석과 동적 분석을 결합한 접근 방식을 제안하였다. PE(Portable Executable) 파일에서 추출한 DLL(Dynamic-Link Library)과 API(Application Programming Interface) 특징을 추출한 후 다양한 머신러닝 알고리즘을 이용하여 실험을 진행하였다. Logistic Regression, Decision Tree, Random Forest, Bagged Classifier, AdaBoost, Gradient Boosting 모델을 사용하였으며. 이 중 Gradient Boosting 모델이 94.64%의 가장 높은 정확도를 보였다. 이는 Gradient Boosting의 단계적 학습 방식이 복잡한 악성코드 패턴을 효과적으로 탐지할 수 있음을 보여준다[2].

S Usharani 외 1인(2021)은 TF-IDF와 n-gram을 이용하여 랜섬웨어를 탐지하는 방법을 제안하였다. 랜섬웨어에서 추출한 정적 특징들을 TF-IDF(Term Frequency-Inverse Document Frequency)와 n-gram 알고리즘을 이용하여 실험을 진행하였다. Decision Tree, Naive Bayes, AdaBoost, Gradient Tree Boosting 모델을 사용하였으며, Gradient Tree Boosting 모델이 99.99%라는 정확도를 보였다[3].

Sibel Gülmez 외 1인(2020)의 연구는 Opcode 시퀀스를 그래프로 변환하여 탐지하는 방식을 제안하였다. Opcode 시퀀스를 가중치 및 방향 그래프로 변환함으로써, 단순한 시퀀스 분석을 넘어 코드의 구조적 특성을 포착할 수 있었다. 특징 추출 과정에서 그래프 노드 각도의 히스토그램을 생성하는 방법은 그래프의 복잡한 구조를 수치화하는 효과적인 방법이다. Random Forest, Decision Tree, SVM, KNN 모델을 사용한 실험에서 Random Forest 모델이 98%의 가장 높은 정확도를 보였다. 이는 Random Forest가 그래프 구조에서 추출된 복잡한 특징을 효과적으로 학습할 수 있음을 보여준다[4].

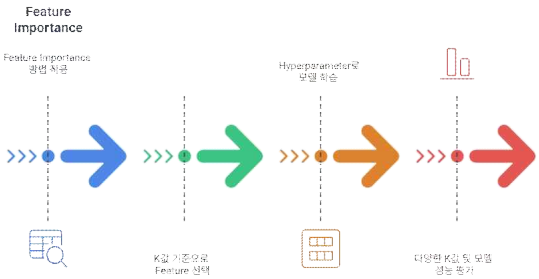
Rami Sihwail 외 2인(2021)의 연구는 메모리 이미지 분석이라는 독특한 접근 방식을 제안하였다. Volatility 도구를 사용하여 메모리 이미지에서 API, DLL, 핸들, 권한, 네트워킹, 코드 주입 관련 특징을 추출하는 방식은 악성코드의 실시간 동작을 포착할 수 있는 효과적인 방법이다. 특히 코드 주입과 같은 고급 악성코드 기술을 탐지하는 데 유용할 수 있다. Naive Bayes와 SVM 모델을 사용한 실험에서 SVM 모델이 98.5%의 높은 정확도를 보였다[5].

3. Feature Importance 알고리즘

3.1 Gain Ratio

Gain ratio는 기존 정보 이득을 보완하여, 각 분할의 고유 정보를 고려함으로써 속성 값이 많은 특성에 대한 편향을 줄인다[6]. 이는 데이터 분할 시 발생하는 정보의 잠재적 양을 정량화하는 고유 정보를 산출하여, 분할 기준을 보다 신뢰성 있게 조정한다. 이를 통해 단순 정보 이득만을 사용할 때 발생할 수 있는 과도한 분할을 방지하며, 특성 선택 과정의 균형을 이룬다. 결과적으로, Gain ratio는 정보량과 분할의 일반화 가능성을 동시에 고려한 평가 척도를 제공한다. 다양한 실험에서 Gain ratio를 적용한 분할 방식은 고차원 및 이질적인 데이터셋에서 분류 성능 향상에 기여한다.

4. 방법론



(그림 2) Featrue Importance 기반의 랜섬웨어 탐지

4.1. 데이터 전처리

머신러닝 학습하기 전, 기존의 데이터셋을 새로운 데이터셋을 생성하기 위해 Feature Importance 알고리즘 중 하나인 Gain Ratio를 사용한다. 해당 알고리즘에서 K값을 통해 탐지율이 얼마나 변화하는지 확인하기 위해 K값을 0.1부터 0.4까지 총 4개를 지정하여 데이터셋을 생성한다.

Gain Ratio는 각 분할의 정보 이득을 고유 정보로 정규화하여, 특성 값이 많은 속성에 내재할 수 있는 편향을 줄이고, 신뢰할 수 있는 특성 선택을 가능하게 한다. 이를 통해 랜섬웨어와 정상 파일 간의 차별적 특성을 명확히 파악할 수 있으며, 모델이 불필요한 특성에 의한 과적합 없이 핵심 패턴을 효과적으로 학습할 수 있도록 돕는다.

4.2. 실험 환경

실험은 Anaconda의 Jupyterlab에서 진행하였으며[7], 데이터셋은 랜섬웨어 PE 특징으로 구성되어 있다. 데이터셋의 분포는 정상 2,500개, 악성 2,500개로 구성되어 있다[8][9]. 표 2는 실험을 진행한 환경 및 소프트웨어 버전을 보여준다.

<표 2> 실험 환경

OS	Windows 10
CPU	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
Memory	16GB
Jupyterlab	3.5.3
Python	3.8.16
Anaconda	4.12.0

3.3. 실험 결과

표 3, 4, 5, 6은 Gain Ratio에서 K값을 0.1부터 0.4까지 설정하여 데이터셋을 생성 후 모델 학습한 결과이다.

실험 결과, RF 모델이 K값 변화에 영향 없이 꾸준히 높은 성능을 유지했고, DT 모델도 안정적인 결과를 보여준다. 반면 SVM은 K가 0.4로 커지면서 성능이 급격히 저하되는 특징을 보였고, MLP는 대부분의 K값 구간에서 안정적인 성능을 유지하되, K=0.4에서 하락하는 결과를 보여준다. 이는 K값 변화에 따라 모델별 정확도가 달라질 수 있음을 알 수 있으며, 랜섬웨어 탐지 환경 및 어떤 특성을 이용할지에 따라 적절한 모델을 선택해야 한다는 점을 알 수 있다.

<표 3> K=0.1일 때 모델 학습 결과

Model	RF	DT	SVM	MLP
Accuracy	99.22%	98.14%	97.65%	98.04%
F1-score	99.22%	98.14%	97.65%	98.02%
Precision	99.21%	98.13%	97.64%	98.04%
Recall	99.23%	98.15%	97.66%	98.03%

<표 4> K=0.2일 때 모델 학습 결과

Model	RF	DT	SVM	MLP
Accuracy	99.21%	98.43%	96.28%	97.75%
F1-score	99.21%	98.43%	96.28%	97.75%
Precision	99.20%	98.42%	96.34%	97.74%
Recall	99.22%	98.44%	97.32%	97.74%

<표 5> K=0.3일 때 모델 학습 결과

Model	RF	DT	SVM	MLP
Accuracy	99.41%	98.92%	95.49%	98.14%
F1-score	99.41%	98.92%	95.49%	98.14%
Precision	99.22%	98.91%	95.50%	98.13%
Recall	99.21%	98.93%	95.52%	98.14%

<표 6> K=0.4일 때 모델 학습 결과

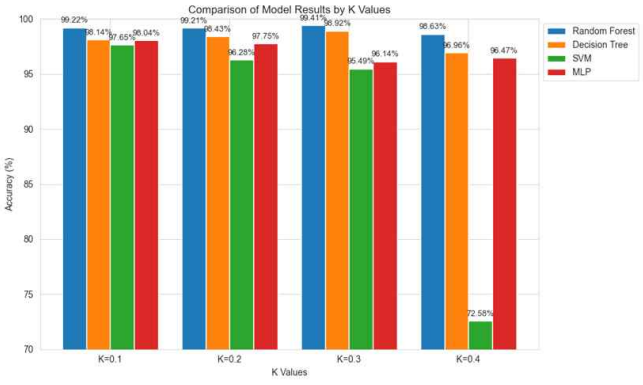
Model	RF	DT	SVM	MLP
Accuracy	98.63%	96.96%	72.58%	96.47%
F1-score	98.63%	96.96%	70.19%	96.47%
Precision	99.61%	98.98%	81.34%	98.48%
Recall	99.62%	98.95%	72.02%	98.47%

<표 7> 모델 Best Hyperparameters

Model	Best Hyperparameter
RF	'n_estimators': 50, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_depth': 10, 'bootstrap': False
DT	'min_samples_split': 2, 'min_samples_leaf': 1, 'max_depth': 20, 'criterion': 'gini'
SVM	'kernel': 'linear','gamma': 'scale', 'C': 1
MLP	'solver': 'adam', 'learning_rate': 'invscaling', 'hidden_layer_sizes': (100, 50), 'alpha': 0.0001, 'activation': 'relu'

<표 8> K값에 따른 Feature 개수

K Value	Feature 개수
0.1	42
0.2	32
0.3	25
0.4	16



(그림 3) K값에 따른 전체적인 모델 결과값

5. 결론

본 연구에서는 기존 분류 기법의 한계를 보완하고자, Feature Importance 알고리즘 중 Gain Ratio를 활용하여 K값을 기준으로 새로운 데이터셋을 구

성함으로써, 최적화된 특성 집합을 도출하고 이를 기반으로 머신러닝 모델을 학습시키는 새로운 분류 방법론을 제안하였다. K값을 활용한 데이터셋 재구성을 통해 특성을 줄여나갔으며, 이후 다양한 모델 학습을 진행하였다. 실험 결과 RF 모델은 K=0.3일 때 99.41%, DT 모델은 K=0.3일 때 96.96%, SVM 모델은 K=0.1일 때 97.65%, MLP 모델은 K=0.3일 때 98.14%의 정확도를 보였다. 이를 통해 K값의 변화에 따라 모델마다 정확도가 달라질 수 있음을 알 수 있다.

향후 연구에서는 본 논문에서 제안한 Gain Ratio 알고리즘을 제외한 다른 Feature Importance 방법을 적용하여, 어떤 방법이 탐지하는데 좋은 결과를 보이는지 비교하는 실험을 진행할 예정이다.

6. Acknowledgements

이 논문은 대한민국 교육부의 재원으로 한국연구재단 융합연구지원사업 (NRF-2024S1A5C3A02043653)의 지원을 받아 수행된 연구임.

또한, 과학기술정보통신부의 재원으로 정보통신기획평가원(IITP)의 지원을 받아 수행된 연구임(No. 2710008542, 대규모 노드에서 블록단위의 효율적인 거래 확정을 위한 최종성 보장 기술개발).

참고문헌

[1] Roger S. Pressman "Software Engineering A Practltiners' Approach" 3rd Ed. McGraw Hill

[2] Ijaz, Muhammad, Muhammad Hanif Durad, and Maliha Ismail. "Static and dynamic malware analysis using machine learning." 2019 16th International bhurban conference on applied sciences and technology (IBCAST). IEEE, 2019.

[3] Usharani, S., and S. G. Sandhya. "Detection of ransomware in static analysis by using Gradient Tree Boosting Algorithm." 2020 International Conference on System, Computation, Automation and Networking (ICSCAN). IEEE, 2020.

[4] Gülmez, Sibel, and Ibrahim Sogukpinar. "Graph-based malware detection using opcode sequences." 2021 9th International Symposium on Digital Forensics and Security (ISDFS). IEEE, 2021.

[5] Sihwail, Rami, Khairuddin Omar, and Khairul Akram Zainol Arifin. "An Effective Memory Analysis for Malware Detection and Classification." Computers, Materials & Continua 67.2 (2021).

[6] HARRIS, Earl. Information Gain Versus Gain Ratio: A Study of Split Method Biases. In: AI&M. 2002.

[7] Anaconda, Available: https://www.anaconda.com/

[8] RansomwareDetection, Available: https://github.com/mudimathur2020/RansomwareDetection

[9] MarauderMap, Available: https://github.com/THU-WingTecher/MarauderMap