<https://pdfs.semanticscholar.org/dc3b/592aa24e9aac1b751e181c818d8121941229.pdf?_ga=2.149680624.777198538.1584411180-964089746.1583287686>

**[0. INTRODUCTION]**

|  |
| --- |
| **Delay Tolerant Network (DTN): 통신이 간헐적으로 이루어지는 영역에서 적합**한데, 이는 환경적 조건이나 멀웨어 때문이다.   * End-to-end 연결이 없다. * **Store and forward 방법으로 통신**이 이루어진다. * 정보는 packet보다는 **bundle을 통해** 조직되어 전송된다.   **Malware:** Malicious Software로, **바이러스, 트로이 목마, 웜, 루트킷 등을 포함**한다.   * **Proximity malware:** **멀웨어의 malicious code로, node를 공격하여 그 기능을 수정**하기도 한다.   + DTN을 부적절하게 이용하여 DTN의 통신을 **멀웨어의 증식**에 사용하게 한다.   + Infrastructure model에서는 **DTN에 central monitoring이 없으므로, DTN에서의 proximity malware 탐지가 필요**하다. |

**[1. Methodology – Naïve Bayes]**

|  |
| --- |
| Naïve Bayes classifier는 **Bayes 확률 이론에 기반하여 예측하며, 단순하고 성능이 좋은 모델을 만드는 데 사용**된다.   * 데이터의 규모가 작은 경우 Naïve Bayes를 사용하는 것이 가장 좋다. * **멀웨어의 행동 정보를 얻기 위한 목적으로 Naïve Bayes classifier를 사용한다.**   + **Node를 normal node와 malicious node로 구분한다.**   Proximity malware를 다루는 만큼, **각 node와 가까이 있는 이웃 node들을 검사**할 필요가 있다.   * 1. 모든 node가 **각 node의 이웃에 대해 안다**고 가정한다. * 2. **One-hop neighbor가 malware에 감염**되었는지 검사한다. * 3. 같은 방법으로 **two-hop neighbor**를 검사한다. * 모든 data transmission에 대하여 packet delivery를 계산한다.   + **Packet delivery가 낮은 전송을 하는 node가 malware infected node로 간주**된다. * 1-hop, 2-hop node에 대한 모든 assessment를 끝낸 후 **수집한 정보를 컴파일하고 그것을 이용하여 SVM을 훈련**시킨다.   + **Naïve-Bayes 모델에 제공된 데이터가 SVM classifier의 최적성에 영향**을 준다. |

**[2. Methodology – Support Vector Machine]**

|  |
| --- |
| SVM은 state-of-the-art 머신러닝 기술 중 하나로, statistical learning 이론에 기반하고 있다.   * **작동 방법:** **2개의 dataset에 대한 margin**을 구한다. * **Supervised learning model**로, 데이터를 분석하고 패턴을 인식할 수 있다. * 서로 다른 여러 분야와 실제 상황에서 좋은 성능을 보인다.   + 텍스트 분류, 이미지 프로세싱, 생물정보학, 인공신경망, 최적화 등 * **최소한의 parameter를 통해 problem을 잘 생성**하며, 그것은 **quadratic programming problem**이다. * **입력 데이터를 high dimensional space로 매핑**시키며, 이것은 분류를 돕는다.   + Linear kernel을 이용한 SVM은 특히 binary 분류 문제에 잘 적용된다. * **Linear classification과 non-linear classification 모두** 가능하다.   + 2개의 dimension인 경우 하나의 line으로 구분 가능   + 더 높은 차원의 경우에는 **hyperplane을 이용하며, 이것 역시 perceptron을 이용한 linear programming**으로 할 수 있다. * **Support Vector:** SVM classifier에서 사용되는 **훈련된 데이터의 부분집합**   **[SVM을 이용한 node 분류 방법]**  어떤 **dataset을 이용하여 SVM을 훈련**시킨 후, **새로운 data를 이용하여 테스트**한다.   * SVM은 **새로운 데이터가 separator의 어떤 side에 맞는지** 알아낸다. * 이때 데이터는 오직 1개의 side에만 맞을 수 있고, 양쪽 다 맞을 수는 없다.   훈련을 위하여 **Naïve Bayes classifier의 데이터**를 입력받는다.   * SVM은 **Proximity malware와 그것의 행동 사이의 관계**를 찾아내서 **Proximity malware의 기존 행동**을 탐지하게 한다. * SVM의 장점은 학습된 데이터를 이용하여 학습하는 것인데, 따라서 **새로운 행동을 예측 및 탐지**할 수 있다.   SVM은 **overfitting이 비교적 나타나기 어려운데**, 따라서 large network에서 적합하다. |

**[3. System Design]**

|  |
| --- |
| 1. Network를 만든 다음, **node가 초기화**되고 **각 node는 그것들의 neighbor를 찾는다**. 2. **DTN은 proximity malware에 더 취약**하므로, **1-hop, 2-hop neighbor**들이 고려된다. 3. 데이터 전송 시작 후, **Naïve Bayes는 packet delivery ratio를 이용하여 malware에 영향을 받는 노드**를 찾는다. 4. 수집된 정보를 이용하여 SVM을 훈련한다. 5. SVM은 **trained data와 predictable data를 이용하여 malware를 탐지**한다. |

**[4. My Model Design] (테스트 시 각 packet의 destination node는 알 수 없음)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **[DTN의 node 설계]**  DTN의 node는 **그래프 자료구조로 설계**한다. (인접행렬 또는 인접 리스트로 연결 관계 표현)  일반 node와 악성 node의 경로 탐색 방법:   * 일반 node는 **목표 지점까지 Dijkstra 알고리즘**을 이용하여 탐색한 후, 거리가 최적이 되는 다음 node로 이동한다. * 악성 node는 **다음 node를 랜덤하게 선택**하여 이동한다.   그래프 구조 상세 설계:   * Node는 **20~25개** 사이에서 랜덤으로 한다. * 임의의 node A와 B가 연결되어 있을 확률은 **15%**로 한다. * 전체 node 중 임의의 node가 악성 node일 확률은 **10%**로 한다.   + 시간이 흐르면서 **malware가 증식하면서 악성 node의 개수가 증가**한다.   **[DTN 시뮬레이션 설계]**   1. [DTN의 node 설계] 에 따라 node를 구성한다. 2. 매 time마다 **임의의 node A에서 다른 임의의 node B로 패킷을 전송**한다. (총 100회)    1. [일반 node와 악성 node의 경로 탐색 방법] 에 따라 다음 node로 이동한다. 3. **패킷의 이동 경로를 학습 데이터**로 한다.   **[Naïve Bayes 모델 설계]**   |  |  | | --- | --- | | 입력값 | 1. 해당 node에서 **다음 node (first hop neighbor)로 갈 때의 noise**의 전체 node에 대한 **Z 값** (구간: Z<=-1, -1<Z<=0, 0<Z<=1, Z>1) 2. 다음 node (first hop neighbor)에서 **그 다음 node (second hop neighbor)로 갈 때의 noise**의 평균값의 전체 node에 대한 **Z 값** (구간: Z<=-1, -1<Z<=0, 0<Z<=1, Z>1) | | 출력값 | **일반 node 또는 악성 node** (binary) |   **[SVM 모델 설계]**   |  |  | | --- | --- | | 입력값  (좌표) | 1. 해당 node에서 다음 node 중 **최다 선택 node의 선택률**에 대한 표준정규분포에서의 **z값** 2. 다음 node (first hop neighbor)에서 **그 다음 node (second hop neighbor)들** 중 **최다 선택 node의 선택률**에 대한 표준정규분포에서의 **z값** | | 출력값 | Naïve Bayes model에서의 결과값 **(일반 node 또는 악성 node, binary)** |  * 여기서 **noise는 각 node를 선택할 확률 에 대한 엔트로피**를 이용하여 계산한다.   **[Dataset 설계]**  각 DTN 및 그것에 대한 시뮬레이션 내용을 학습 데이터 1개로 하고, **총 학습 데이터는 1000개로 한다. 이들 중 900개를 training data, 100개를 test data**로 한다. |