Phishcheck

비지도학습 기반 피싱 URL 탐지 시스템

지도교수 양환석 교수님

피싱클리너 - 팀장 정여진 팀원 서장석 양승원 정채영

목차

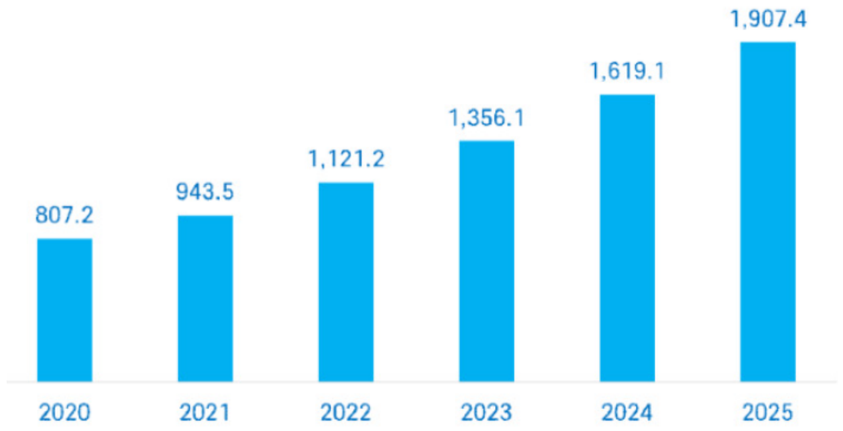
chapter 1

프로젝트 개요

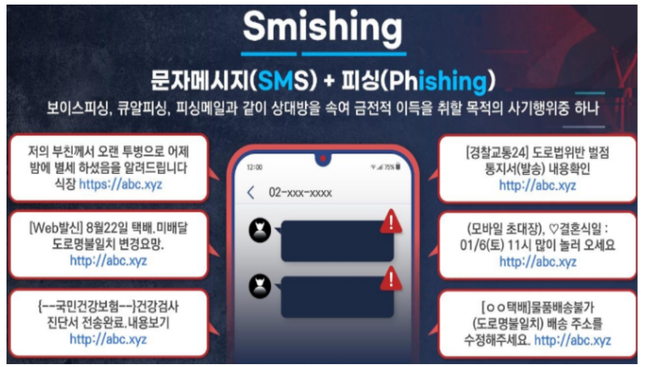
- 프로젝트 주제 및 선정 배경
- 팀원소개 및 역할분담
- 프로젝트 추진일정

프로젝트 주제 및 선정배경

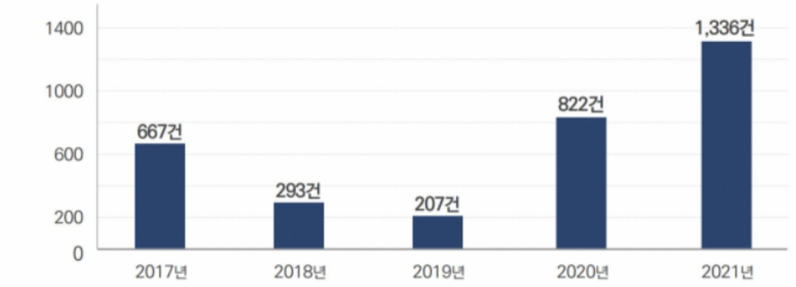
AI에 대한 이용률 상승



피싱 URL 발생률 증가추이



최근 5년 간 스미싱 발생건수 추이



프로젝트 주제 및 선정배경

기존피싱 진단과의 차별점 고안

↓

비지도학습을 통한 알고리즘 학습

팀원소개 및 역할분담

			
정여진	서장석	양승원	정채영
GMM 알고리즘	MeanShift 알고리즘	K-Means 알고리즘	Agglomerative Cluster 알고리즘

프로젝트 추진일정

-	추진내용	3월	4월	5월	6월	7월	8월	9월	10월	11월
기획	프로젝트 주제 확정 및 데이터 수집									
개발	**특징값 추출 및** 모델 별 학습									
구축	**GUI 만들기**									
마무리	**보고서 작성 및** 프로젝트 마무리									

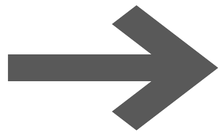
chapter 2

프로젝트 진행

- 프로젝트 구성
- 프로젝트 개발환경
- 특징값 리스트
- GUI 구현

프로젝트 구성

입력창을 통해 확인할 URL 입력



입력된 URL에 대한 피싱확률을
GUI에서 알고리즘 별로 출력

총 4개의 알고리즘 별 피싱확률의 평균값으로
정상사이트와 피싱사이트로 분류

프로젝트 개발환경

특징값 리스트

1. IP Address in URL
2. URL Length
3. Shortening Service
4. Having @ Symbol
5. Double Slash Redirecting
6. Prefix/Suffix
7. Having Subdomain
8. Domain Registration Length
9. Favicon
10. Port
11. HTTPS Token
12. Count Redirection
13. Disabling Right Click
14. Age of Domain
15. DNS Record

알고리즘 별 학습

1. K-means

▶ KMeans

```
# KMeans 클러스터링
kmeans = KMeans(n_clusters=2, n_init=10, random_state=42)
result = kmeans.fit_predict(X)

# 클러스터의 분포 확인
print("클러스터 분포:", np.unique(result, return_counts=True))

# 레이블과 클러스터 결과 비교
label_df = pd.DataFrame({'label': y, 'cluster': result})
print("레이블과 클러스터 결과 비교:")
print(label_df.groupby('label')['cluster'].value_counts())

# 클러스터와 레이블 매핑
cluster_label_mapping = label_df.groupby('cluster')['label'].mean().to_dict()
# 클러스터의 평균 레이블을 기반으로 매핑
cluster_to_label = {cluster: 1 if cluster_label_mapping[cluster] > 0.5 else 0
```

K개의 초기 중심점을 선택하고
각 데이터 포인트를 가장 가까운 중심에 할당하여
클러스터를 형성

2. GMM

▶ GMM(가우시안분포)

```
# GMM 클러스터링
gmm = GaussianMixture(n_components=2, n_init=10, random_state=42)
gmm.fit(X)
result = gmm.predict(X)

# 클러스터의 분포 확인
print("클러스터 분포:", np.unique(result, return_counts=True))

# 레이블과 클러스터 결과 비교
label_df = pd.DataFrame({'label': y, 'cluster': result})
print("레이블과 클러스터 결과 비교:")
print(label_df.groupby('label')['cluster'].value_counts())
```

여러 개의 가우시안 분포(정규 분포)를 혼합하여
데이터의 분포를 모델링하는 확률적 모델

알고리즘 별 학습

3. Meanshift

▶ MeanShift(평균이동)

```
# MeanShift 생성 및 훈련
ms = MeanShift(bandwidth=2.27)
result = ms.fit_predict(X)

# 클러스터의 분포 확인
print("클러스터 분포:", np.unique(result, return_counts=True))

# 레이블과 클러스터 결과 비교
label_df = pd.DataFrame({'label': y, 'cluster': result})
print(label_df.groupby('label')['cluster'].value_counts())

# 정확도 출력
accuracy = accuracy_score(y, result)
print(f'정확도:', accuracy * 100)
```

데이터의 밀도가 높은 지역을 찾아
클러스터를 형성하는 데 사용

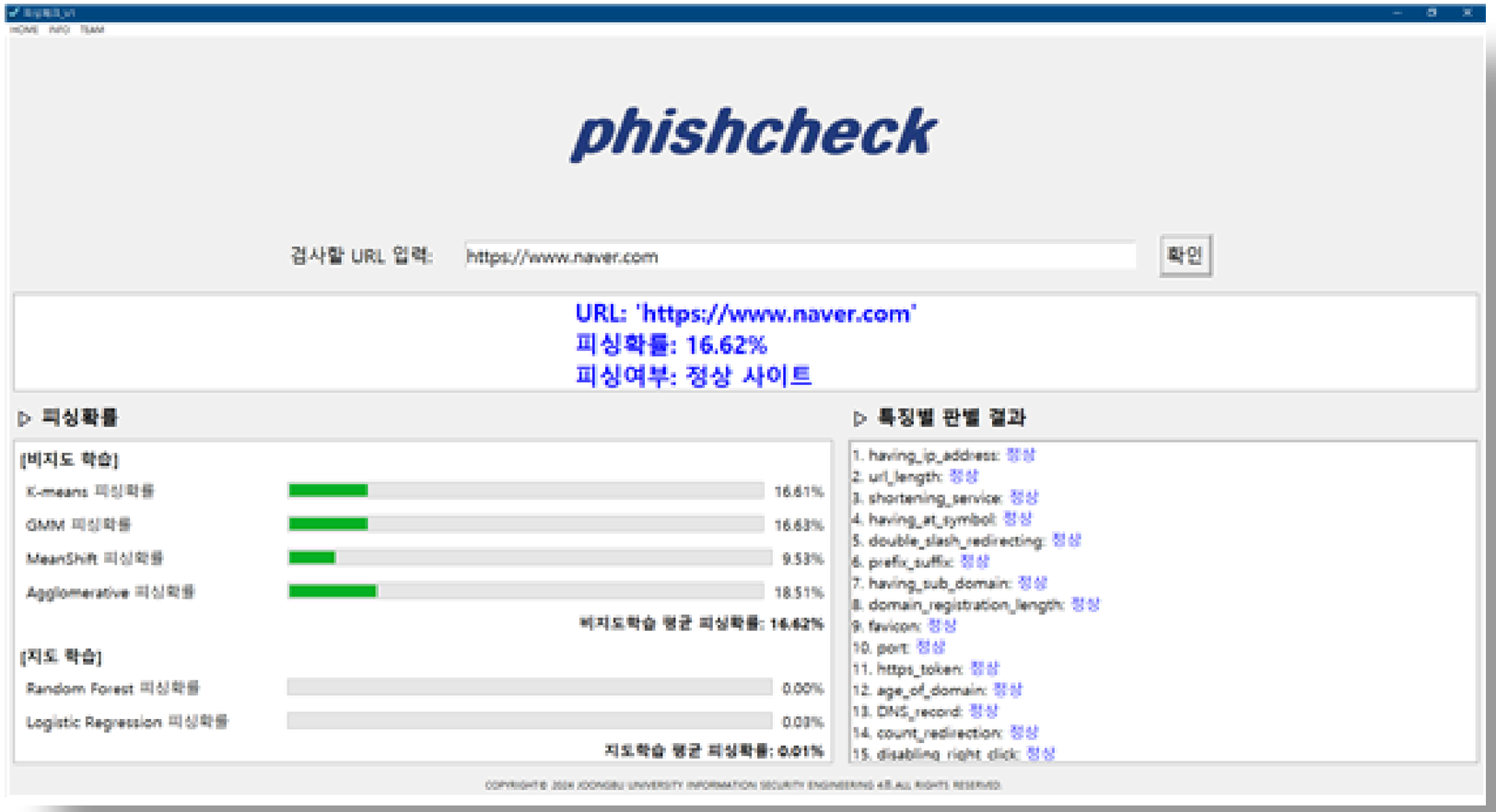
4. AgglomerativeCluster

▶ AgglomerativeCluster(병합군집)

```
hierarchical_cluster = AgglomerativeClustering(n_clusters=2)
cluster_labels = hierarchical_cluster.fit_predict(X)

print('군집 결과:', np.unique(cluster_labels, return_counts=True))

label_df = pd.DataFrame(y)
label_df['cluster_label'] = cluster_labels
print(label_df.groupby('label')['cluster_label'].value_counts())

# 정확도 확인
accuracy_score(y, cluster_labels)
```

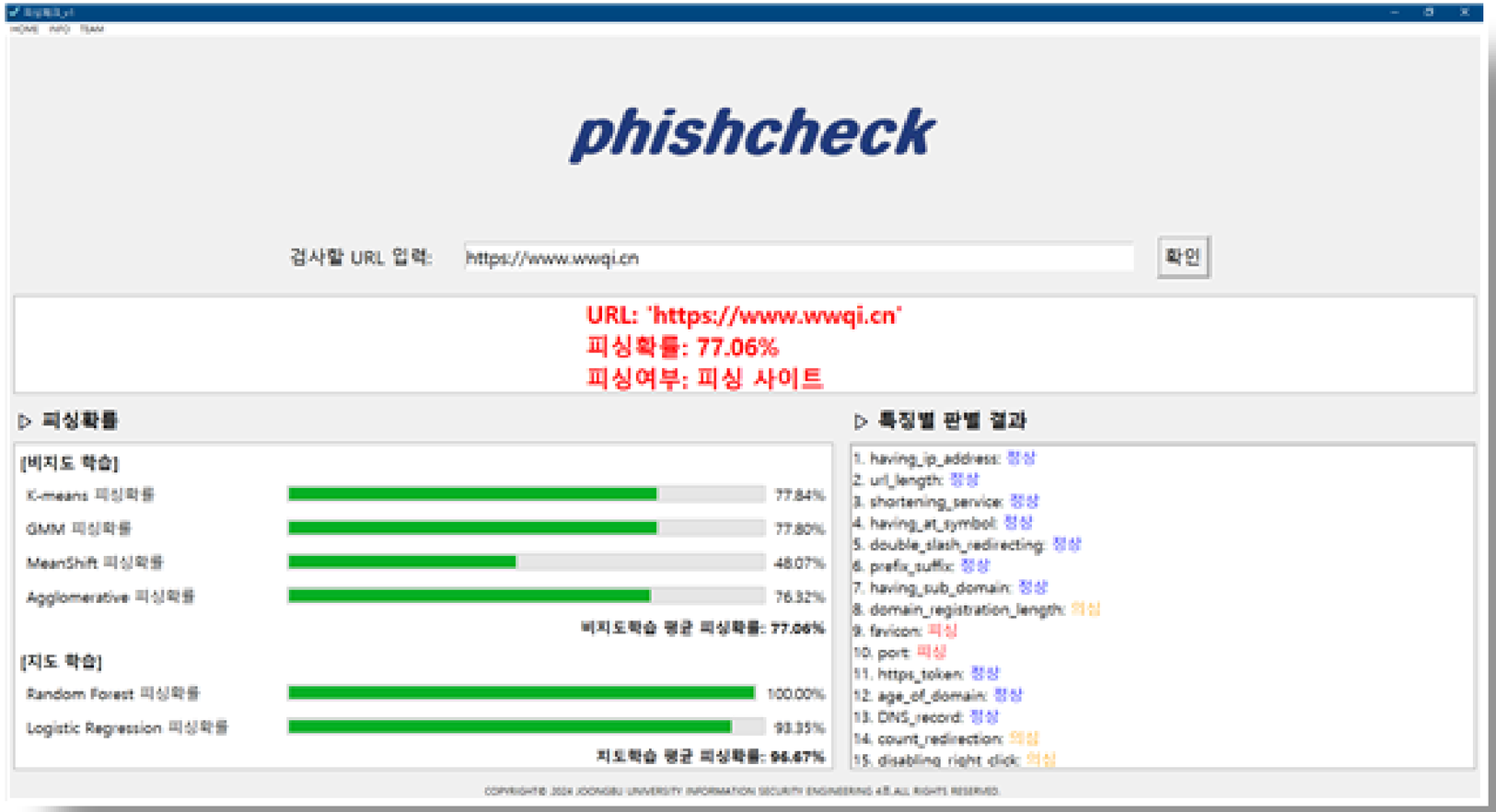
클러스터간의 거리 또는 유사성을 계산한 후
가장 가까운 두 클러스터를 찾아 병합

GUI 구현



GUI - 정상사이트 판별

GUI 구현



GUI - 피싱사이트 판별

chapter 3

프로젝트 결과

- 알고리즘 학습률 & 결론
- 프로젝트 보완사항
- 프로젝트 시연영상

알고리즘 학습률 & 결론

Algorithm	DataSet	learning rate
K-Means	정상:15,000&피싱:7,000	91.72%
Gmm	정상:15,000&피싱:7,000	91.67%
MeanShift	정상:15,000&피싱:7,000	88.26%
AgglomerativeCluster	정상:15,000&피싱:7,000	88.37%

비지도학습 모델은
대량의 URL 데이터를 분석하고
잠재적인 악성 패턴을 찾아내는 데에
탁월한 성능을 발휘

피싱과 정상사이트 간의
경계가 불분명하거나
새로운 형태의 공격이 등장했을 때
효과적으로 대응 할 수 있음

프로젝트 보완사항

1. 특징값들 추가로 추출
2. 더 다양한 데이터들 추가 학습
3. 웹에서의 실시간 탐지 기능

프로젝트 시연 영상

[시연 영상 QR 코드]

감사합니다