公開リポジトリにおける自動Exploit判定
－実装から得られた知見と課題－

株式会社ラック　サイバー・グリッド・ジャパン
赤木雅弥

研究背景　～増加するExploit～

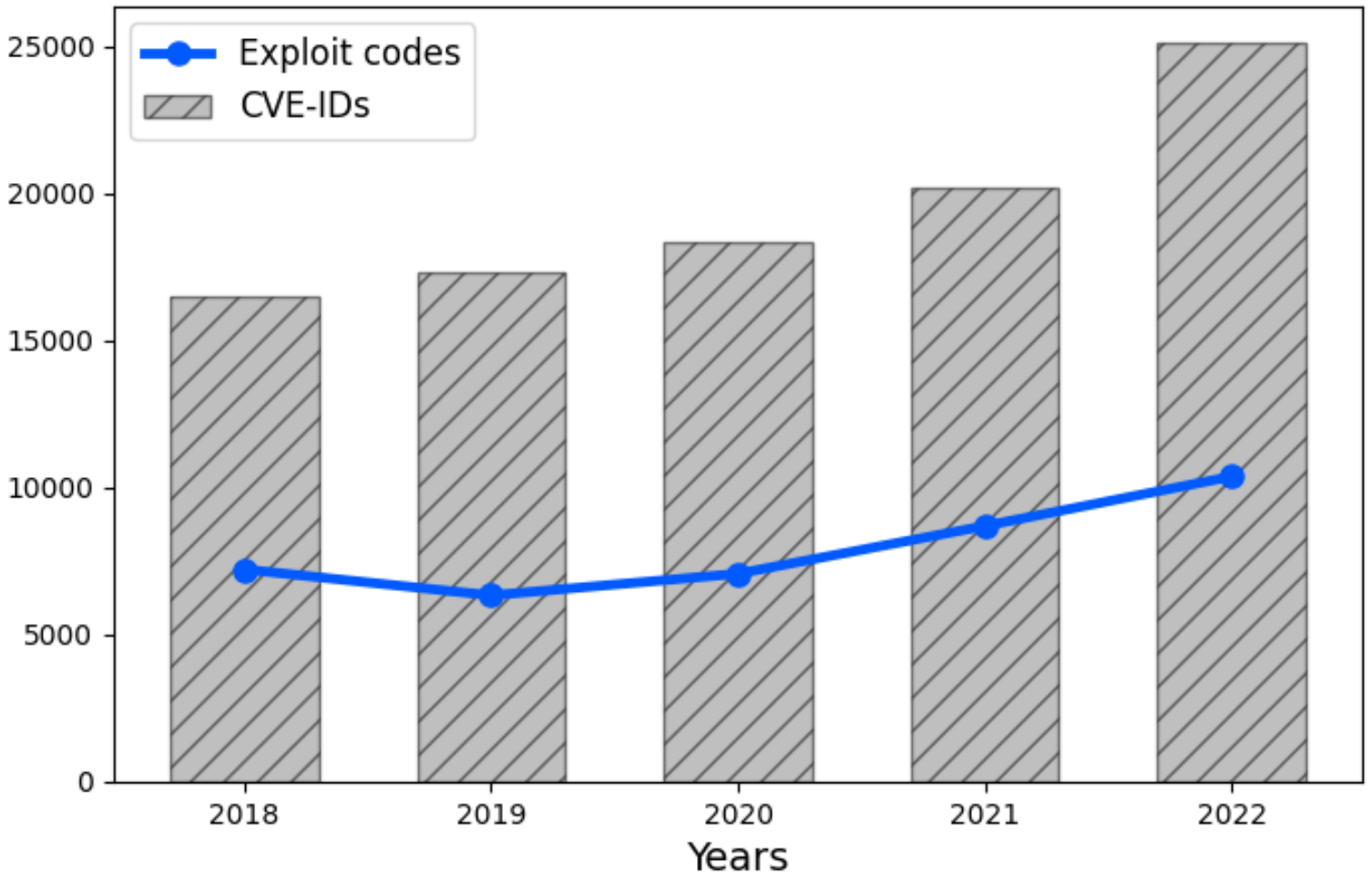
2021年

- CVE-ID発行件数：20,153
- Exploitコード件数：8,689

2022年

- CVE-ID発行件数：25,082
- Exploitコード件数：10,369

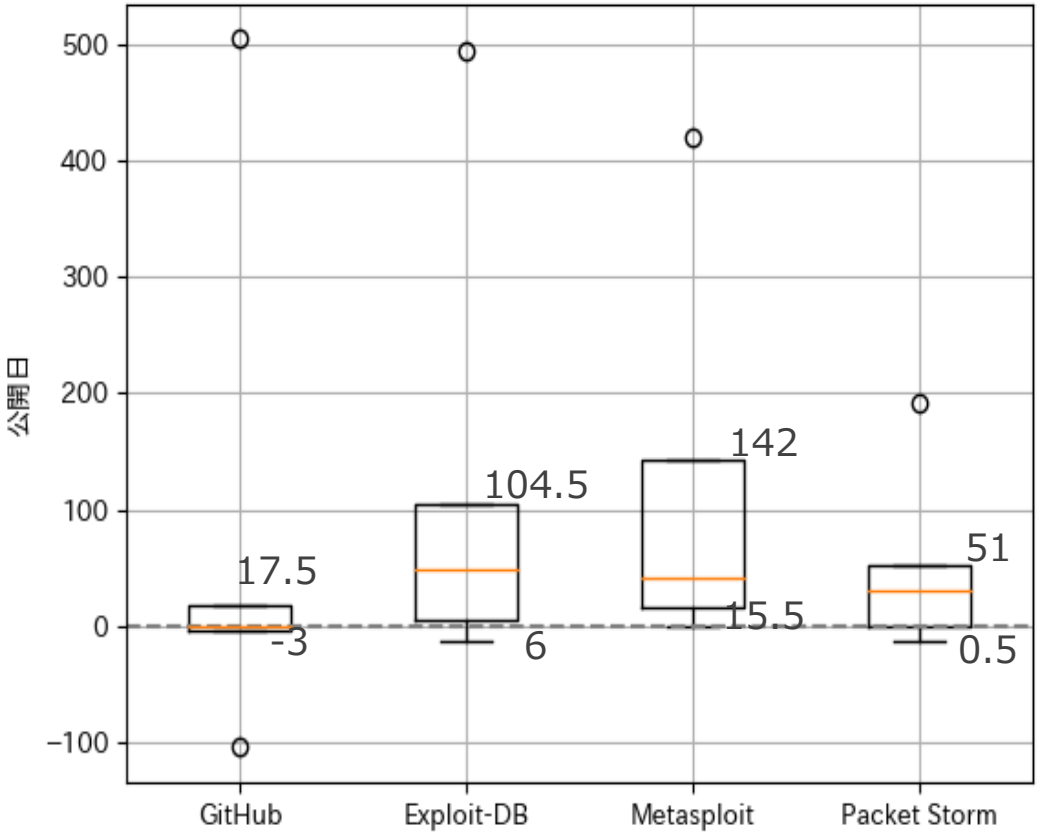
CVE-ID発生件数**25%**増加
Exploitコード件数**20%**増加



データ引用：Beyond the surface:Investigating Malicious CVE proof of Concept Exploits on GitHub. 2022

研究背景　～GitHubと他の主要アーカイブとの公開日比較～

GitHubと他の主要アーカイブを公開日で比較（当社独自に調査）



*) NVDの公開日を基準(0)

	GitHub	Exploit-DB	Metasploit	Packet Storm
平均値(日)	37	73	91	34
中央値(日)	0	48	41	30

算出方法

- 2022年から2023年に公開されたCVE-IDに紐づくExploitコードの公開日で比較
- NVDの公開日を基準(0)とした

結果

中央値の評価より *) 左図のオレンジ線
GitHubへのExploitコードの公開が一月程度速いことが判明

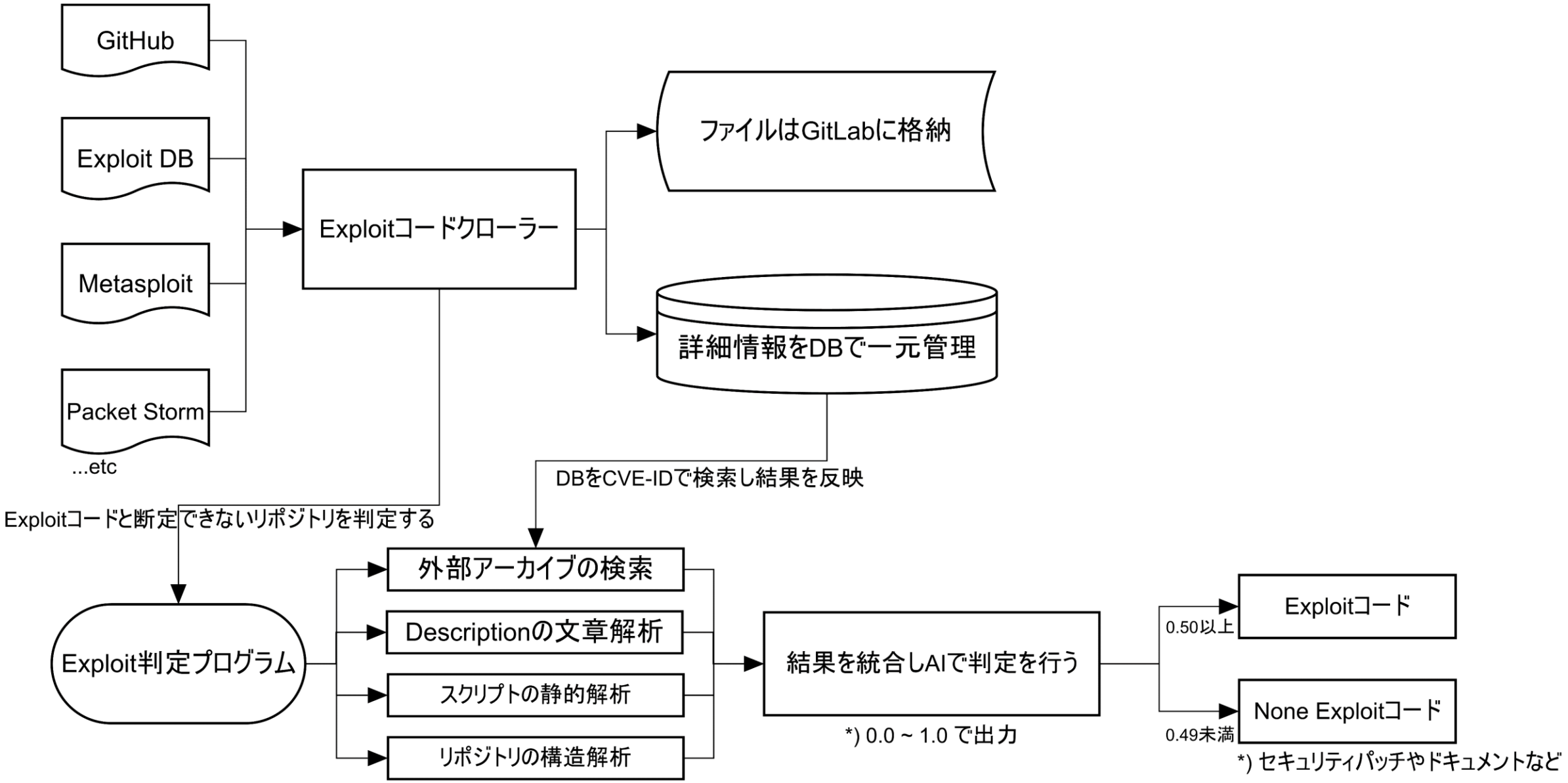
本研究の目的

AIを用いてGitHubからExploitコードを早期かつ正確に検出する仕組みの構築を目指す

＜現状の課題＞

- Exploitコードに関する公開リポジトリ数が膨大
- 自動でExploitコードを収集した際にノイズ（関連ツールやパッチ等）が入ってしまう
- Exploitコードの判定に高度なスキルが求められる

実装システムの概要



判定AIのモデル選定理由と判定精度

LightGBM

- 勾配ブースティングを用いている
- 数値データから他クラス分類が可能
- 少量のデータセットで学習が可能

Accuracy	0.83
Precision	0.83
Recall	0.96
F値	0.89

LightGBM以外で検討したモデル

- K-means法　　：教師なし学習のため本研究の趣旨と合わない
- LSTM　　：LSTMの学習に必要なデータセットを確保できない

GitHubリポジトリの判定結果

GitHubから過去に採取した326件のリポジトリを調査

AI判定	Exploitコード	268
	None Exploitコード	58

アナリストが326件を解析した結果との比較

		合計	アナリスト判定による内訳	
			Exploitコード	None Exploitコード
AI判定	Exploitコード	268	222	46
	None Exploitコード	58	12	46

結果

- 94%（222件 / 234件）の精度でExploitコードを検出した
- その一方で、46件のノイズをExploitコードと誤判定した

AIが誤ってExploitコードと判定した事例

CVE-2023-36884

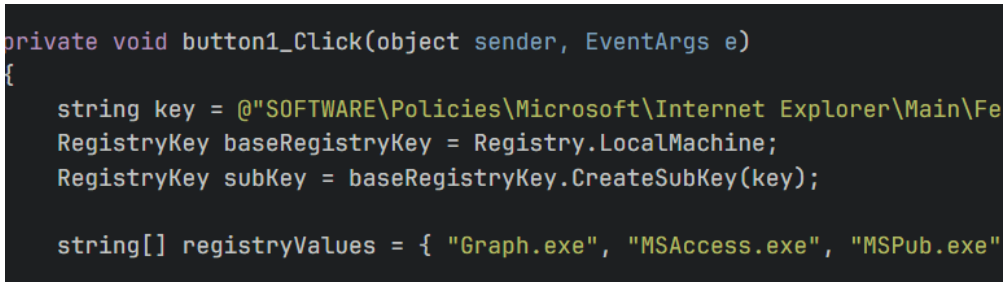
- 「Office」や「Windows HTML」のRCEの脆弱性
- AI判定結果は「0.80」と高いスコアを示したが、PoCではなくパッチであった

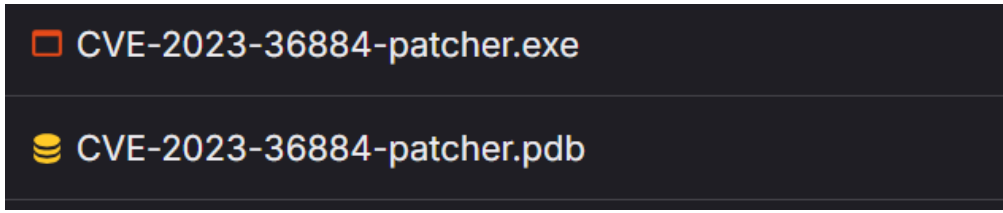
<原因>

1. READMEの文章が短く中国語で記述されており正常な判定ができなかった

2. 同様の脆弱性のExploitコードと酷似するスクリプトがあり見分けが困難だった

3. 複数のバイナリファイルが存在した





まとめ・今後の展望

まとめ

- GitHub上のExploit関連のリポジトリには、3割程度のノイズが存在する
- 複数の指標を用いてリポジトリをAIで判定した結果、高精度で判定できた
- スクリプト解析の改良などで精度改善の余地がある

今後の展望

- スクリプトファイルやバイナリファイルの動的解析の実装
- 誤って判定したリポジトリを再学習し、AI判定を継続的に強化
- 偽Exploitコードの特徴を分析し、検知を目指す
- GitHubリポジトリのクロール条件を工夫し、ゼロデイ攻撃に対応