一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

プロンプト整合済みLLMによる Telegramからの STIX脅威インテリジェンス抽出

金子　翔威† 　青砥　　陸† 　グエンティヴァンアン†† 　インミンパパ†† 　田辺　瑠偉††

吉岡　克成†††,††

† 横浜国立大学大学院環境情報学府
†† 横浜国立大学先端科学高等研究院
††† 横浜国立大学環境情報研究院

E-mail: †{kaneko-shoi-ps,aoto-riku-hx}@ynu.jp, ††{nguyen-anh-xd,yinminn-papa-jp,yoshioka}@ynu.ac.jp

あらまし　近年，サイバー犯罪者やハクティビストは，攻撃サービスの広告，作戦状況の共有，キャンペーンの調整を目的として Telegram を積極的に活用している．これらの投稿には脅威インテリジェンスとして有用な情報が含まれているが，その非構造的かつ多様な性質により，大規模な分析は困難である．本研究では，Telegram 上の攻撃者による投稿を STIX 2.1 形式の構造化脅威インテリジェンスに自動変換する手法を提案する．具体的には，人間アノテータと大規模言語モデル（LLM）が協調してプロンプトを段階的に改善し，整合したアノテーションを可能とする「プロンプト整合フレームワーク（prompt-alignment framework）」を提案する．本フレームワークでは，フィードバックループを通じてプロンプトを改善することで，人間と LLM が生成する STIX オブジェクトの整合性と品質を向上させる．ハクティビスト活動に関与する 5 つの Telegram チャンネルを入力として提案フレームワークを適用し，10 回を超える反復によるプロンプト改善を実施したところ，人間と LLM が出力する STIX オブジェクトの F1 スコアは調整前の 0.49 から 0.80 まで向上した．最終的に得られたプロンプトが他の人間や LLM に対しても有効であるかを検証するため，第二の人間アノテータおよび異なる LLM に適用した結果，それぞれ第一人間アノテータの結果との F1 スコアは 0.66 および 0.62 となった．さらに，最終プロンプトを 37 個の Telegram チャンネル（総メッセージ数 74,679 件）に適用し，得られた 1051 個の STIX オブジェクトを分析することで，同一標的への攻撃，共通の攻撃手法の使用，攻撃者間の連携パターンなどが明らかになった．

キーワード　STIX，脅威インテリジェンス，テレグラム

STIX Threat Intelligence Extraction from Telegram via Prompt-Aligned LLM

Shoi KANEKO†, Riku AOTO†, Nguyen THI VAN ANH††, Yin Minn Pa Pa††, Rui TANABE††, and Katsunari YOSHIOKA†††,††

† Graduate School of Environment and Information Sciences, Yokohama National University
†† Institute of Advanced Sciences, Yokohama National University
††† Faculty of Environment and Information Sciences, Yokohama National University
E-mail: †{kaneko-shoi-ps,aoto-riku-hx}@ynu.jp, ††{nguyen-anh-xd,yinminn-papa-jp,yoshioka}@ynu.ac.jp

Abstract　Cybercriminals and hacktivists increasingly use Telegram to promote attack services, share operational updates, and coordinate campaigns. While their posts contain valuable threat intelligence, large-scale analysis is difficult due to their unstructured nature. This study proposes a method to convert such posts into STIX 2.1 format using a prompt-alignment framework, where human annotators and LLMs collaboratively refine prompts through iterative feedback. Applied to five Telegram channels, the framework improved the F1 score of human − LLM alignment to 0.80. Validation with another annotator and LLM yielded F1 scores of 0.66 and 0.62. Applying the method to 37 channels (74,679 messages) generated 1,051 STIX objects, uncovering shared targets, common techniques, and actor collaboration patterns.

Key words　STIX, Threat Intelligence, Telegram

1. はじめに

近年，サイバー犯罪者は Telegram をはじめとするソーシャルプラットフォームを積極的に活用し，攻撃サービスの宣伝，協力者の募集，攻撃活動の調整等を行っている．特に，サイバー攻撃のサービス化（Cybercrime-as-a-Service: CaaS）や政治的・思想的動機に基づくハクティビスト集団によるキャンペーンが注目されている．これらの攻撃者は Telegram 上で，サービスの紹介，攻撃成功事例の共有，キャンペーンへの参加呼びかけ等の投稿を積極的に行っている．これらの投稿には，価格情報，被害者 IP アドレス，使用ツール，攻撃手法といったサイバー脅威インテリジェンスとして有用な情報が含まれている [1]〜[3].

しかしながら，これらの投稿は膨大かつ非構造的で，内容も口語的・断片的であるため，手動での分析は現実的ではない．従来研究では，Web サイトや脅威レポートといった構造的情報源 [4] や，Twitter のような半構造的プラットフォーム [5] が主な対象とされてきた．しかし，攻撃者による Telegram 投稿を STIX 2.1 形式（機械可読な脅威インテリジェンス記述フォーマット）へと変換する研究は存在しない．

この課題に対し，大規模言語モデル（LLM）は有望な解決策を提供する．LLM は非構造的かつ口語的なテキストを処理し，構造化された JSON 形式で出力する能力を有しており，ノイズを含むサイバー犯罪の記述を STIX 形式に変換できる．しかしながら，このタスクには複数の課題が存在する．STIX 2.1 は柔軟性と表現力に富むため，同一の内容に対して複数の妥当な表現が存在する．また，人間の専門家同士であっても，アクター情報・攻撃手法・インフラ情報のどこを強調するかによってアノテーションが異なる場合がある．さらに，LLM はプロンプト設計に対して感度が高く，誤生成や汎化性のばらつきも生じるため，「正しい構造化出力」の定義自体が曖昧になるという根本的な問題が存在する．

本研究では，この問題に対処するため，LLM を協調的なアノテーションパートナーとみなす「プロンプト整合フレームワーク」を提案する．5 つの Telegram チャンネル（DDoS-for-hire サービスおよび DoS 中心のハクティビストを含む）から収集した未処理データを基に，人間アノテータと LLM が 10 回以上の反復的フィードバックループを通じて，STIX 2.1 形式のゴールドスタンダードデータセットを共同構築した．各イテレーションでは，アノテーションとプロンプトを同時に洗練させ，表現の自由度が高いフィールド（例：`description`）は整合性維持のため除外した．最終的に，（1）高品質な STIX 出力を安定して生成する最終プロンプト，（2）人間と LLM の合意を反映した正解データセット (ゴールドスタンダードデータセット) という 2 つの成果物を得た．また，このとき，人間と LLM が出力する STIX オブジェクトの F1 スコアは調整前の 0.49 から 0.80 に向上した．

最終プロンプトの汎用性を検証するため，さらに 2 つの実験を実施した．1 つ目は，最終プロンプトを第 2 の人間アノテータに提示し，これに基づきアノテーションを実施したところ，ゴールドスタンダードデータセットに対する F1 スコア 0.66 を達成した．2 つ目は，異なる LLM（GPT-4o）に最終プロンプトを適用し，ゴールドスタンダードデータセットに対する F1 スコア 0.62 の一致率を得た．これは，最終プロンプトが他の人間・モデルに対しても第 1 アノテータと整合した出力を促すアノテーション指針として機能することを示唆している．

最後に，この最終プロンプトを 37 個のチャンネル・74,679 件の Telegram 投稿に適用し，構造化された脅威インテリジェンスを生成した．その結果，攻撃者の共通標的，再利用されるインフラ，頻繁に使用される攻撃ツールなどのパターンが明らかとなった．

本論文の貢献は以下の通りである：

- 人間と LLM の協働によって STIX 2.1 出力を逐次洗練する「プロンプト整合フレームワーク」を提案した．
- プロンプトの汎化性を検証し，他の人間・モデルへの適用可能性を実証した．
- Telegram 上の大規模な投稿データを構造化し，攻撃者・手口・標的に関する洞察を得た．

2. 関連研究

STIX を用いた CTI 構造化における課題について，Jin ら [6] は (1) 手動作成への依存によるデータ拡張の遅延，(2) 脅威出現から構造化までの時間的遅延，(3) 既存データソースがマルウェアシグネチャや URL 等の低レベル情報に偏重し，攻撃者情報等の高度な情報が不足している点を指摘している．

これらの課題に対し，本研究では，LLM を用いた STIX 構造化の自動化により，攻撃者の投稿に含まれる CTI をリアルタイムかつ効率的に構造化することを目指す．さらに，LLM によるアノテーション精度の向上により，攻撃者や標的等の複雑な情報を STIX 形式で自動構造化し，攻撃者の投稿に含まれる情報を網羅的に構造化することを目指す．

また，機械学習を用いて CTI を含むメディアを STIX 形式で構造化する研究には複数の先行事例が存在する．藤井ら [7] の研究では，CTI を発信している複数のウェブサイトの情報から機械学習などを用いて情報を抽出し，STIX 形式への整理を行っている．また，Marchiori ら [4] は，CTI レポートから自然言語処理などを用いて STIX 形式への構造化を行っている．さらに，Siracusano ら [8] は，CTI レポートに対して手動で生成した STIX オブジェクトのデータセットを作成し，2 段階に分けて LLM を用いた STIX 形式への構造化を行っている．

これらの先行研究における構造化対象は主に CTI レポートであり，記載内容は報告すべき CTI に合わせて十分に整理されているものであった. 本研究では，体系化されていない Telegram チャンネルの投稿に対し，LLM を用いた STIX 形式への自動構造化を行う．さらに，人間による構造化結果との比較を通じて，LLM による STIX 構造化の信頼性を検証する．

また，防御側が Facebook をはじめとするソーシャルネットワークプラットフォーム上で脅威情報を共有するサービスが

存在し，これらのデータソースを利用してサイバー攻撃の実態把握を行った研究が報告されている [9]．さらに，Twitter からセキュリティ関連イベントを自動収集・分析する手法が提案されている [5], [10], [11].

3. 用語説明

3.1 Telegram

Telegram とは，インターネットを用いてメッセージの送受信を行うアプリケーションであり，2025 年現在で約 9.5 億人のユーザが利用している [12]．Telegram は大規模なユーザ基盤と高い匿名性により注目される一方，サイバー攻撃者やサイバー攻撃サービス提供者が活動の宣伝・発信手段として利用する事例が報告されている [1]〜[3].

Telegram の機能は多岐に渡るが，ユーザ間のメッセージやファイルの送受信の他に，チャンネルと呼ばれる機能がある．チャンネルでは，管理者となったユーザが参加しているユーザに対して一方向的に発信を行う形式をとる．攻撃者が管理するチャンネルは，攻撃活動に関する重要な情報源となるため，継続的な監視が必要である．特に，攻撃者が活動発信に利用するチャンネルには，攻撃結果，提供サービス，標的情報等が投稿される．これらの情報の迅速な分析により，攻撃の予防および実態解明が可能となる．

無数に存在する攻撃者の Telegram チャンネルから発信される情報は膨大かつリアルタイムであるため，情報収集・分析の自動化が不可欠である．しかし，他の情報源における脅威情報と比較して，Telegram 上の情報は形式が定まっておらず，構造化が困難である．

本研究では，この課題を解決するため，LLM を用いて Telegram チャンネル上の脅威情報を標準化データフォーマットである STIX 2.1 形式 [13] へ自動変換する手法を提案する．

3.2 STIX(Structured Threat Information eXpression)

STIX は，OASIS CTI 技術委員会によって策定された，サイバー脅威情報の構造化および共有を目的とした標準データフォーマットである．

STIX 2.1 では，JSON 形式でサイバー脅威情報を STIX Object として記述し，オブジェクトの内容とオブジェクト間の関係により全体を体系的に表現する．具体的には，攻撃者，標的，攻撃手段，脆弱性等の情報を STIX Domain Object（SDO）として表現し，オブジェクト間の関係を STIX Relationship Object（SRO）として表現する．また，各 STIX Object には種類ごとに定義された JSON キーが存在し，これらをプロパティと呼ぶ．

STIX 2.1 形式のドキュメント利用者は，情報源に含まれる情報とその関係性を STIX オブジェクトから読み取ることができる．”description”プロパティ等の自由記述項目を除き，多くのプロパティは STIX 2.1 仕様により値が制約されているため，解釈の一貫性が保たれる．

SDO および SRO の具体例を以下に示す．SDO を作成する際は，対象の種類に応じてタイプを決定する必要がある．SDO には 18 種類のタイプが定義されており，選択したタイプは”type”プロパティの値として記録される．SDO の”type”以外のプロパティは，”type”の値に応じて必須・任意が決定される．SRO を作成する際は，関係を定義する 2 つのオブジェクトの ID を”source_ref”および”target_ref”プロパティに記述する．さらに，”relationship_type”プロパティにより，2 つのオブジェクト間の関係性を明示する．なお，”relationship_type”プロパティの取りうる値は，STIX 2.1 仕様 [14] において，”source_ref”および”target_ref”に指定された STIX オブジェクトのタイプごとに定義されている．

```
{
    "type" = "threat-actor",
    "name" = "ExampleStresser",
    "id" = "threat-actor--00...",
    "description" = "Example of SDO",
    "roles" = ["director"],
    "resource_level" = "club",
    "primary_motivation" = "organizational-gain"
},
{
    "type" = "relationship",
    "relationship_type" = "uses",
    "id" = "relationship--00...",
    "description" = "Example of SRO",
    "source_ref" = "threat-actor--00...",
    "target_ref" = "attack-pattern--02..."
}
```

4. 手　　法

4.1 概　　要

本研究の目的は，サイバー犯罪関連の Telegram チャンネルから収集した非構造的メッセージを，構造化された STIX 2.1 オブジェクトに変換することである．しかし，STIX の柔軟性や Telegram 投稿の非公式かつ曖昧な表現の性質から，信頼性と一貫性のあるアノテーションパイプラインの構築は容易ではない. そこで我々は，LLM（大規模言語モデル）を協調的アノテータとして位置づける「プロンプト整合フレームワーク」を提案する．反復的な改善を通じて，汎用的な最終プロンプトとゴールドスタンダードデータセットの両方を構築する．全体のワークフローを図 1 に示す.

4.2 データ収集と初期アノテーション

まず，DDoS-for-hire サービスおよび DoS を中心としたハクティビスト活動に関与する 5 つの Telegram チャンネル Ch1，Ch2，Ch3，Ch4，Ch5 から初期アノテーションのための代表的なサンプルとして，各チャンネルごとに最新 10% のメッセージを最低 10 件抽出し，合計 84 件のメッセージを収集した．これらのチャンネルでは，主に攻撃者の設備やサービスのアップデートの情報，成功させた攻撃の報告，攻撃サービスの割引に関する宣伝が多く発信されており，高い活動頻度と攻撃サービスの継続的な宣伝を基準として選定した.

人間アノテータ（アノテータ A）は，このサンプルに対して STIX 2.1 形式で手動アノテーション G_0 を作成した．同時

に，OpenAI の LLM モデル（gpt-o1）[15] に初期プロンプト P_0 を与え，LLM によるアノテーション L_0 を生成させた．

4.3 プロンプトおよびアノテーションの反復的改善

我々は，いずれか一方のアノテータを絶対的な「正解」とはせず，アノテータ A と LLM の間で協調的にアノテーションとプロンプトを改善する戦略を採用した．以降では i 回目の改善により生成されたプロンプトを P_i，改善されたアノテーションをそれぞれ G_i, L_i と書く．G_i と L_i の差異に基づいてプロンプトを修正し，アノテーションも更新する．

人間のアノテータが文脈的に必要であると判断しているにもかかわらず，LLM が特定の STIX オブジェクトを生成しなかった場合には，そのオブジェクトを作成するよう明示的にプロンプトを修正した．この際，プロンプトの汎化性能を高めるために，初期の 5 チャンネルに特化しすぎないよう留意した．逆に，LLM が生成したオブジェクトの中で人間のアノテータが見逃していたものがある場合にはアノテーションセットに当該オブジェクトを追加した．また，`description` や `labels` などの表現の自由度が高いフィールドは，主観的なばらつきの原因となるため除外した．f 回の繰り返しの結果，アノテーション基準として明確となった最終プロンプト P_f と，LLM と人間の双方の強みを反映したゴールドスタンダードデータセット G_f を得た．この反復作業は 10 回以上繰り返した．

4.4 最終プロンプトの汎用性評価

最終プロンプト P_f の汎用性を評価するため，以下の 2 つの独立した検証を行った：

人間アノテータによる評価： 第二の人間アノテータ（アノテータ B）に P_f を提供し，同じメッセージセットのアノテーションを実施した．得られた結果は G_B とし，G_f と比較した．

LLM による汎化評価： 異なるモデル（LLM-2）に P_f を与え，アノテーションを生成し G_f と比較した．ここでは OpenAI の LLM モデル gpt-4o を使用 [16] した．なお，ここで得られた結果は L_{4o} とした．

4.5 スケーリング：大規模データへの応用

最後に，P_f を用いて，DDoS 関連の 37 個の Telegram チャンネルから収集した計 74,679 件のメッセージに対してアノテーションを実施した．得られた STIX 2.1 オブジェクトにより，攻撃者のインフラ構成，攻撃手法との関連，チャンネル間の協調パターンなどを明らかにした．

4.6 評 価 方 法

本研究では，最終的に得られたゴールドスタンダードデータセット（G_f）と，他のアノテータ（別 LLM または別の人間）による STIX オブジェクト出力の一致度を定量的に評価するために，次のような手法を用いた．

まず，すべての STIX オブジェクトをタイプと属性値に基づいてペアリングし，一致度を計算する．`type`，`created`，`id` 等の属性は一致度計算から除外し，意味的な一致に焦点を当てる．特に `name` フィールドに関しては，Python の difflib ライブラリ [17] を用いて，名前間の字句的な類似度をベースラインとして評価した．さらに，`campaign`，`location` オブジェクトに関しては，類似した文脈や目的を持つもの同士をより正確に照合するため，gpt-40 [16] による補助評価を行った．具体的には 2 つのオブジェクトを提示し，プロンプトで `campaign` に関しては Can these name relate to same STIX campaign? Answer 'yes' or 'no' only. `location` に関しては Are these name relate to same location? Answer 'yes' or 'no' only. と指示した．これにより，`Legend500Coupon` と「$500 discount with coupon」など，異なる表記でも同一の意味を持つものは一致と判定した．

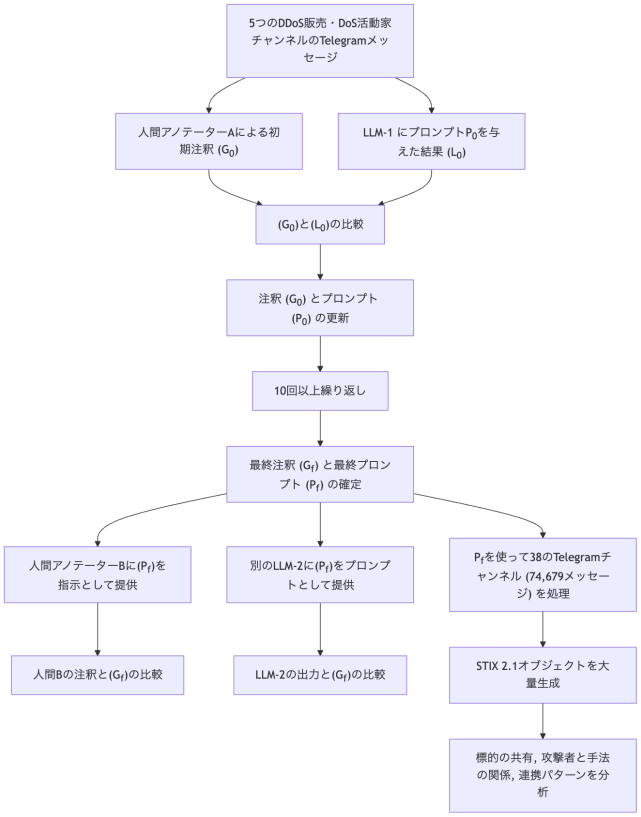

図 1: 手法概要．アノテータ A と LLM のアノテーションが整合するようにプロンプトを改善し，最終的なゴールドスタンダード G_f とプロンプト P_f を構築．P_f の汎用性を第二の人間アノテータおよび第二の LLM で評価すると共に，37 の Telegram チャンネルに適用し，脅威情報を分析した．

これらの手法により得られた一致ペア数および未一致数に基づいて，各アノテータの出力に対して **Precision（適合率）**，**Recall（再現率）**，および **F1-score** を算出した．具体的には，G_f と他のアノテータの両方に存在し一致と判定されたオブジェクトを **True Positives (TP)**，他のアノテータにのみ存在し G_f と一致しなかったオブジェクトを **False Positives (FP)**，G_f のみに存在し他のアノテータと一致しなかったオブジェクトを **False Negatives (FN)** と定義する．

5. 評 価 結 果

5.1 作成された STIX オブジェクト

はじめに，本研究での実験のうち，4.2 節および 4.3 節にお

表 1: 各アノテーションの獲得 STIX オブジェクト数

		Telegram チャンネル				
		Ch1	Ch2	Ch3	Ch4	Ch5
メッセージ数		10	21	10	12	31
アノテーション (オブジェクト数)	G_0	43	88	31	37	112
	G_f	45	88	39	37	112
	L_0	25	29	11	9	66
	L_f	36	76	32	35	65

表 2: G_f の STIX オブジェクト数のタイプ内訳

オブジェクトのタイプ	Telegram チャンネル				
	Ch1	Ch2	Ch3	Ch4	Ch5
attack-pattern	8	4	7	2	2
campaign	3	0	1	2	2
domain-name	1	7	3	4	1
grouping	0	0	0	0	0
identity	5	13	1	4	13
indicator	1	1	2	1	2
infrastructure	2	1	4	2	0
observed-data	2	16	1	4	14
relationship	21	35	16	15	42
threat-actor	1	1	1	1	3
tool	0	0	0	0	0
location	0	0	0	0	2
url	0	8	0	1	31
user-account	1	2	3	1	0
合計	45	88	39	37	112

いて生成された STIX オブジェクトの内訳を示す．4.2 節において，アノテーション G_0, G_f, L_0, L_f に含まれる STIX オブジェクトの数と内訳を表 1 に示す．チャネル Ch1 とチャネル Ch3 においては，G_f において生成されたオブジェクトの数が G_0 に比べて増加している．このことから，本手法により人間のアノテータだけが作成を行った場合よりも多くの情報をオブジェクト化できているといえる．また，L_0 と L_f を比べるとそれぞれのチャネルにおいてオブジェクト数が増加しており，プロンプトの改良により人間のアノテーションに近い数のオブジェクトが得られるようになっていることがわかる．

次に，G_f に含まれるオブジェクトタイプの内訳を表 2 に示す．表 2 から，最終的に生成されたゴールドスタンダードには，`url` などの単純なデータを示すオブジェクトと同様に，`attack-pattern` や `identity` などのより複雑なオブジェクトが生成されていることが確認できる．

5.2 提案フレームワークによるアノテーションの評価

本節では，人間と LLM によるアノテーションの整合性について述べる．表 3 は各アノテーションの一致度をまとめたものである．まずプロンプトの改良を行う前の人間アノテータ A による初期アノテーション G_0 と LLM (gpt-o1) による初期アノテーション L_0 の一致度は適合率 0.38，再現率 0.72, F1 スコア 0.49 であり，大きな乖離がある．一方，提案手法による最終プロンプト P_f に基づく人間アノテータ A のアノテーション G_f と LLM(gpt-01) によるアノテーション L_f の一致度は，適

表 3: アノテーションの一致度 (右側のアノテーションを基準とした左側のアノテーションの一致度)．G_B, L_{4o} はそれぞれ人間アノテータ B，LLM(gpt-4o) のプロンプト P_f に基づくアノテーション．

アノテーション	適合率（Precision）	再現率（Recall）	F1 スコア
L_0 vs G_0	0.38	0.72	0.49
L_f vs G_f	**0.75**	**0.86**	**0.80**
G_B vs G_f	0.82	0.56	0.66
L_{4o} vs G_f	0.49	0.83	0.62

表 4: G_f を基準とした L_f のタイプ別一致度 (L_f vs G_f)

タイプ	適合率（Precision）	再現率（Recall）	F1 スコア
attack-pattern	0.9565	0.9565	0.9565
campaign	0.75	0.8571	0.8
domain-name	1	0.9333	0.9655
identity	0.7778	0.7778	0.7778
infrastructure	0.2222	1	0.3636
observed-data	0.4054	0.6522	0.5
threat-actor	1	1	1
tool	null	null	null
location	0	0	0
url	0.9744	0.95	0.962
Average	0.75	0.86	0.8

合率 0.75, 再現率 0.86, F1 スコア 0.80 となっており大きく向上しており，プロンプトの改良により，整合性が向上していることが確認できる．さらに，最終プロンプト G_f を別の人間アノテータ B に示し，アノテーションを実施した結果 G_B においても，適合率 0.82, 再現率 0.56, F1 スコア 0.66 となっており，人間アノテータ A と LLM (gpt-o1) が協同で作成したプロンプト G_f は別のアノテータ B に対しても一定の効果がみられる．同様にプロンプト G_f を用いて別の LLM (gpt-4o) が行ったアノテーション L_{4o} においても適合率 0.49, 再現率 0.83, F1 スコア 0.62 となっており，一定の整合性の改善が確認できる．

また，各アノテータについて，STIX オブジェクトのタイプごとの一致度を，それぞれ表 4，表 5，表 6 に示す. この結果から，`url` や `domain-name` などの STIX Cyber-observable Object（SCO）に分類されるタイプを持つオブジェクトや，`threat-actor` や `identity`，`attack-pattern` などのタイプを持つオブジェクトにおいて，特に高い一致率となったことが確認された．

5.3 大規模データへの応用

本研究では，最終的なプロンプト（P_f）を用いて，37 個の DDoS 関連 Telegram チャンネルから得られた 74,679 件の投稿を構造化し，STIX 2.1 形式に変換した．その結果として生成された STIX オブジェクトの内訳を表 7 に示す．また，得られた全体構造を図 2 に示す．

図 2 のネットワークグラフは pyvis [18] によって可視化されており，ノードは SDO または SCO を表し，有向エッジは SRO（関係性）を示している．赤いノードは ThreatActor や AttackPattern などの攻撃者に関する情報，青いノードは Identity や Location などの標的や関連団体に関する情報，黄色のノー

表 5: G_f を基準とした G_B のタイプ別一致度 (G_B vs G_f)

タイプ	適合率（Precision）	再現率（Recall）	F1 スコア
attack-pattern	0.8261	0.8636	0.8444
campaign	0.5	0.6667	0.5714
domain-name	0.8571	0.8571	0.8571
identity	0.7778	0.7778	0.7778
infrastructure	0.7778	0.28	0.4118
observed-data	0.8108	0.5085	0.625
threat-actor	0.7143	1	0.8333
tool	0	0	0
location	1	0.2857	0.4444
url	0.9487	0.9024	0.925
Average	0.82	0.66	0.73

表 6: G_f を基準とした L_{4o} のタイプ別一致度 (L_{4o} vs G_f)

タイプ	適合率（Precision）	再現率（Recall）	F1 スコア
attack-pattern	0.6957	0.9412	0.8
campaign	0.625	0.7143	0.6667
domain-name	0.3571	0.8333	0.5
identity	0.5	0.72	0.5902
infrastructure	0	0	0
observed-data	0.1892	0.7	0.2979
threat-actor	0.5714	1	0.7273
tool	null	null	null
location	0	0	0
url	0.7949	0.9118	0.8493
Average	0.49	0.83	0.62

表 7: 37 チャンネルから生成された STIX オブジェクトの内訳

タイプ	オブジェクトの数	タイプ	オブジェクトの数
attack-pattern	69	observed-data	77
campaign	17	relationship	421
domain-name	88	threat-actor	71
grouping	3	tool	1
identity	145	location	29
indicator	1	url	53
infrastructure	15	user-account	57
ip-address	4		
合計	1051		

ドは SCO および ObservedData に関する情報を示す．なお，ここで述べた本論文でのネットワークグラフのノードの凡例は，図 3 にて表される．

図 4 は，37 の Telegram チャンネルにおける STIX オブジェクト種別の出現数を示している．特定のチャンネル（例：Channel 35，Channel 15，Channel 3）は，構造化可能な脅威情報を多く含んでおり，一方で出現頻度の低いチャンネルも存在する．中には，AttackPattern，Infrastructure，Observable など複数のタイプがバランスよく出現するチャンネルもあれば，Infrastructure に偏った構成のチャンネルも見られた．このような分布の違いは，攻撃者グループの投稿スタイルや提供するサービス内容の差異を反映していると考えられる．

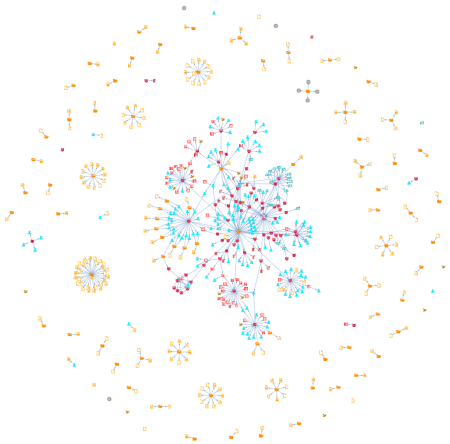

図 2: DDoS チャンネル群の全体像

図 3: ネットワークグラフのノードの凡例

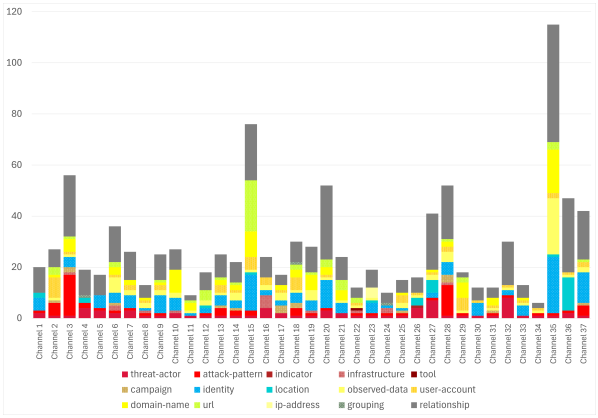

図 4: チャンネルごとの STIX オブジェクト種別出現数

5.4 攻撃者と標的の関係性分析

次に，構造化されたデータから攻撃者と標的の関係に焦点を当てて抽出したネットワークグラフを図 5 に示す．

図 5 の中で，図 6 に示すように，複数の攻撃者が同一の標的を攻撃対象としている例が見られた．このことから，攻撃者間での連携や協調攻撃が行われている可能性が示唆される．このような関係性を考慮することで，将来の攻撃シナリオの予測や攻撃規模の推定に役立つと考えられる．

5.5 攻撃者と攻撃手法の分析

さらに，攻撃者と使用された攻撃手法の関係性に焦点を当てた分析を行い，その全体像を図 7 に示す．

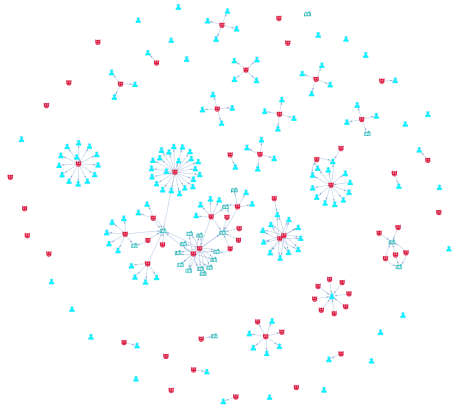

図 5: 攻撃者と標的の全体像．青いノードが特定の組織や地域およびシステムを，赤いノードが攻撃者を表す．

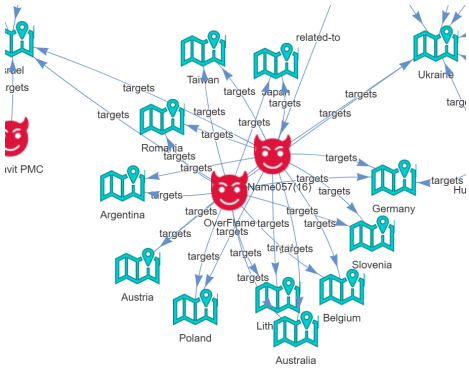

図 6: 同一の標的を持つ攻撃者の例．青緑のノードが標的となった地域を，赤いノードが攻撃者を表す．

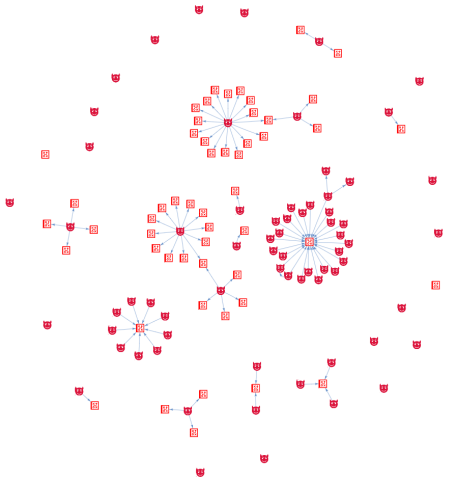

図 7: 攻撃者と攻撃の手口の全体像．四角形の図で表されるノードが攻撃の手口を，顔のノードが攻撃者を表す．

図 8 に示すように，多くの攻撃者が同一の攻撃手段（例：Layer7 DDoS，Amplification など）を利用していることが明らかになった．この結果は，主要な攻撃手法の特定により，効率的な対策立案が可能であることを示唆している．

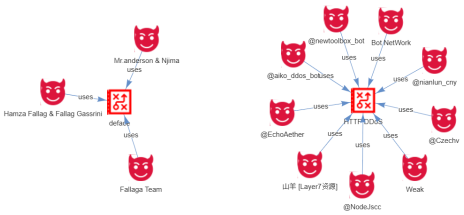

図 8: 多くの攻撃者に利用される攻撃手段の例．四角形の図で表されるノードが攻撃の手口を，顔のノードが攻撃者を表す．

6. 考　　察

6.1 構造化インテリジェンスから得られた洞察

本研究では，Telegram 上の DDoS チャンネルから収集した 74,679 件のメッセージを構造化し，STIX 2.1 形式の脅威インテリジェンスとして記述した．その結果，以下のような重要な洞察が得られた．

LLM による拡張可能なインテリジェンス抽出： 最終的なプロンプト P_f を使用することで人手により生成した STIX オブジェクトと LLM が生成したオブジェクトの F1 スコアは 0.8 となり，人間によるアノテーション作業を LLM が代行した場合にも一定の精度で構造化が可能であることがわかった．すなわち，提案したフレームワークを用いることで，大量の非構造化データを構造化インテリジェンスとして一括処理できることを実証した．本手法は，他の SNS プラットフォームやサイバー犯罪カテゴリにも応用可能である．

最終プロンプトの汎用性： 表 3 に示すように，人間アノテータ (Human A) と LLM(GPT-o1) アノテータにより生成された最終プロンプト P_f に基づいて，別の人間アノテータ（Human B）や別の LLM（GPT-4o）がアノテーション作業を行った場合でも F1 スコアでそれぞれ 0.66，0.62 であり，最終プロンプトがある程度汎用的に機能することが示唆された．

攻撃対象の重複と攻撃者の連携可能性： 図 5 および図 6 に示されるように，複数の攻撃者が同一の標的に対して攻撃を仕掛けている事例が確認された．これは攻撃者間での連携や情報共有の存在を示唆しており，今後の攻撃を予測・防御する上で重要な知見となる．

共通の攻撃手法の利用： 図 7 および図 8 では，多数の攻撃者が同一の攻撃手法（例：UDP Flood，HTTP GET Flood）を採用している様子が可視化された．これは，主要な攻撃手法の傾向を把握することで，効率的な対策の設計が可能であることを意味する．

攻撃インフラの再利用： STIX オブジェクトからは，複数の攻撃者が同一または類似のインフラ（例：ホスティングサービス，スクリプト名）を利用しているケースも明らかになった．これは，被害の拡大を防ぐ上で重要な対処点となる．

本手法は，攻撃者の行動・手口・インフラを構造化データとして定量的かつ可視的に把握できる点で，CTI（Cyber Threat Intelligence）に有用である．たとえば，攻撃者間の連携関係を可視化し，頻出する攻撃手法を抽出して対策の優先度を判断

したり，攻撃インフラの再利用パターンから拡散経路を推定したりすることが可能となる．これにより，インシデント対応や予兆検知，戦略的な防御策の策定に資する知見が得られる．

6.2 限界と今後の課題

本研究には以下のような限界がある：

- 分析対象が DDoS 関連チャンネルに限定されており，他の攻撃カテゴリ（例：フィッシング，情報窃取）への汎用性は今後の検証が必要である．
- STIX 2.1 は表現力が高いが，主観的な判断を伴う項目も多く，アノテーションの一貫性維持が難しい．本研究ではこれを軽減するため，`description` や `labels` などの自由度の高いフィールドは評価対象から除外した．
- LLM はあくまで言語モデルであり，過学習や文脈の誤解，事実の捏造（hallucination）が生じる可能性がある．今後は，ドメイン知識を付与する手法や，マルチモーダル分析との統合も検討すべきである．

今後は，他の犯罪カテゴリや多言語データへの適用，構造化インテリジェンスの自動可視化・分析ツールとの連携も視野に入れて拡張を行っていく予定である．

7. おわりに

本研究では，Telegram 上の DDoS チャンネルにおけるサイバー攻撃関連の投稿を対象として，STIX 2.1 形式への自動構造化手法を提案した．特に，大規模言語モデル（LLM）を単なる生成ツールとしてではなく，人間アノテータとの協調的なアノテーションパートナーとして活用し，プロンプトの反復的改良とデータセットの共構築を通じて，高品質なアノテーション指針とゴールドスタンダードデータ G_f を構築した．

その結果，LLM と人間アノテータ間で最大 F1 スコア 0.8 の一致を達成し，得られたプロンプトは他の人間や別モデルにも適用可能であることを示した．最終的に，37 チャンネル・74,679 件の投稿に本プロンプトを適用し，攻撃者の連携・共通標的・攻撃手法の共有などの重要な洞察を構造化データから得ることができた．

本研究は，非構造的なサイバー犯罪関連情報を機械可読なインテリジェンスとして活用可能にする一つの実践的アプローチであり，今後の CTI 業務への応用が期待される．今後は，対象範囲の拡張（他の攻撃カテゴリや言語）や，自動可視化・分析ツールとの連携により，さらなる実用性向上を目指す．

謝辞　本研究の一部は国立研究開発法人新エネルギー・産業技術総合開発機構（ＮＥＤＯ）の委託事業「経済安全保障重要技術育成プログラム／先進的サイバー防御機能・分析能力強化」（JPNP24003）によるものである．

文　献

[1] S.S. Roy, E.P. Vafa, K. Khanmohammadi, and S. Nilizadeh, “Darkgram: A large-scale analysis of cybercriminal activity channels on telegram,” arXiv preprint arXiv:2409.14596, pp.••–••, 2024. To appear in USENIX Security 2025. https://arxiv.org/abs/2409.14596

[2] B. News, “How cybercriminals are using telegram to sell hacking services,” May 2025. Accessed: 2025-05-10. https://www.bbc.com/news/articles/cdey4prn3e1o

[3] A. Waldman, “Infosec experts detail widespread telegram abuse,” Sept. 2024. Accessed: 2025-05-10. https://www.techtarget.com/searchsecurity/feature/Inf experts-detail-widespread-Telegram-abuse

[4] F. Marchiori, M. Conti, and N.V. Verde, “STIXnet: A Novel and Modular Solution for Extracting All STIX Objects in CTI Reports,” Proceedings of the 18th International Conference on Availability, Reliability and Security, pp.1–11, ARES ’23, Association for Computing Machinery, New York, NY, USA, Aug. 2023.

[5] J. Cui, H. Kim, E. Jang, D. Yim, K. Kim, Y. Lee, J.-W. Chung, S. Shin, and X. Liao, “Tweezers: A framework for security event detection via event attribution-centric tweet embedding,” arXiv preprint arXiv:2409.08221, pp.••–••, 2024.

[6] B. Jin, E. Kim, H. Lee, E. Bertino, D. Kim, and H. Kim, “Sharing cyber threat intelligence: Does it really help?,” Proceedings of the 31st Annual Network and Distributed System Security Symposium (NDSS), pp.••–••, 2024.

[7] 藤井翔太，川口信隆，重本倫宏，山内利宏，“Cyber Threat Intelligence の構造化による分析支援手法の提案,” 研究報告コンピュータセキュリティ（CSEC），vol.2021-CSEC-92, no.47, pp.1–8, March 2021．

[8] G. Siracusano, D. Sanvito, R. Gonzalez, M. Srinivasan, S. Kamatchi, W. Takahashi, M. Kawakita, T. Kakumaru, and R. Bifulco, “Time for aCTIon: Automated Analysis of Cyber Threat Intelligence in the Wild,” ••, vol.••, no.arXiv:2307.10214, pp.••–••, July 2023. http://arxiv.org/abs/2307.10214

[9] V.G. Li, M. Dunn, P. Pearce, D. McCoy, G.M. Voelker, and S. Savage, “Reading the tea leaves: A comparative analysis of threat intelligence,” 28th USENIX security symposium (USENIX Security 19), pp.851–867, 2019.

[10] Q. Le Sceller, E.B. Karbab, M. Debbabi, and F. Iqbal, “Sonar: Automatic detection of cyber security events over the twitter stream,” Proceedings of the 12th International Conference on Availability, Reliability and Security, pp.1–11, 2017.

[11] H. Shin, W. Shim, J. Moon, J.W. Seo, S. Lee, and Y.H. Hwang, “Cybersecurity event detection with new and re-emerging words,” Proceedings of the 15th ACM asia conference on computer and communications security, pp.665–678, 2020.

[12] “Telegram FAQ,” https://telegram.org/faq.

[13] “Introduction to STIX,” https://oasis-open.github.io/cti-documentation/stix/intro.

[14] “STIX Version 2.1,” https://docs.oasis-open.org/cti/stix/v2.1/os/stix-v2.1-os.html.

[15] OpenAI, “Gpt-4 technical report,” https://openai.com/research/gpt-4, 2023. Accessed June 2025.

[16] OpenAI, “Gpt-4o: Openai’ s new multimodal flagship model,” https://openai.com/index/gpt-4o, 2024. Accessed June 2025.

[17] Python Software Foundation, “difflib — helpers for computing deltas,” 2025. Python 3.13.4 documentation. https://docs.python.org/3/library/difflib.html

[18] J. Unpingco and W.H. Institute, “pyvis: A python library for interactive network visualizations,” 2023. Version 0.3.2, BSD license. https://pypi.org/project/pyvis