# AIセキュリティに関する文献収集・分類プラットフォームの提案

長谷川 健人[1,a)]　披田野 清良[1]

**概要**：人工知能 (AI) 技術の発展に伴い，敵対的サンプルやデータポイズニングなどの攻撃や，フェイク動画による偽情報拡散などの負の影響が明らかになっている．こうした攻撃やそれに対する対策，負の影響に関する研究分野は AI セキュリティと呼ばれており，近年では AI セキュリティ分野の研究も進展している．AI セキュリティ分野は進展が早いことから，研究者やエンジニアがこれらの研究分野を追跡することが難しくなっている．本稿では AI セキュリティに関連する技術文書を収集・分析するプラットフォーム，AI Security Threat Intelligence Platform (AISTIP) を提案する．AISTIP は，インターネットから自動的に論文を収集するクローリング機能と，収集した論文にラベルを付与するラベリング機能を備える．ラベリング機能では，データベースから既知のラベルにもとづいてラベルを割り当てる方法と，大規模言語モデル (LLM) を用いて推測した新しいラベルにもとづく 2 つのアプローチを採用する．これにより，急速に進展する分野における正確なラベル付けと新しいラベルへの適応を可能とする．評価実験では，既存のラベリング方法と比較して新しいラベルを効果的に処理することを示す．

**キーワード**：AI セキュリティ，ラベル，ベクトルデータベース，大規模言語モデル

# Literature Collection and Labelling Platform on AI Security

Kento Hasegawa[1,a)]　Seira Hidano[1]

***Abstract:*** Due to the rapid advancements in the AI security field, it has become increasingly challenging for researchers and engineers to keep track of these areas of study. This paper proposes a platform called AI Security Threat Intelligence Platform (AISTIP) that collects and analyzes technical documents related to AI security. AISTIP features a crawling function that automatically gathers papers from the internet and a labeling function that assigns labels to the collected papers. The labeling function employs two approaches: one assigns labels based on known labels from a database, while the other uses large language models (LLMs) to infer new labels. This enables accurate labeling and adaptation to new labels in a rapidly evolving field. Evaluation experiments demonstrate that our method effectively handles new labels compared to existing labeling methods.

***Keywords:*** AI Security, labelling, vector database, large language model

## 1. はじめに

人工知能 (AI) の進歩は目覚ましく、多くの新技術や手法が急速に導入されている。急速な進展に伴い，新しい研究分野が立ち上げられることがある．“AI セキュリティ”分野はその代表的な例であり，AI の普及に伴って表出したリスクを主なテーマとして扱っている．しかし，このような急速に進展する分野では，進行中の研究を体系的に把握することは非常に難しい．こうした研究分野を俯瞰的に理解するためには，プレプリントサーバ等の Web サイトに日々投稿された文献を収集するだけでなく，体系的に整理し，ラベル付け等の手法を用いて整理することが重要である．

[1] 株式会社 KDDI 総合研究所 / KDDI Research, Inc.
[a)] kt-hasegawa@kddi.com

既存技術では，広範なデータセットを収集し，そのデータにもとづいて深層学習モデルを訓練するものであった．しかし，AI セキュリティのように進展中の分野においては，深層学習モデルの訓練に十分な量のサンプルデータを収集するのは難しい．また，短期間で新しい概念が提案されることがあるため，そのような概念にも追従していく必要がある．本研究では，これらの要件に応えたシステムを提案することを目的とする．

本稿では，AI セキュリティに関連する技術論文や情報を収集・分析するプラットフォーム，AI Security Threat Intelligence Platform (AISTIP) を提案する．このプラットフォームの特徴は，ラベル付けの操作に，ベクトルデータベースと大規模言語モデル (LLM) を組み合わせて使用することにある．ベクトルデータベースを用いたラベル付けでは，埋め込み表現を用いて文章の意味上の類似性を考慮することで，予め与えられた少量の教師データにもとづきラベルを付与する．LLM を用いたラベル付けでは，与えられた文章に対応するラベルを LLM により推測することで，教師データに含まれない未知のものを含むラベルを付与する．この 2 つのアプローチを併用し，採用するラベルを選択することで，文章にラベルを付与する．これにより，新しいトピックを扱う文書にラベルを付けることが可能となる．

**貢献**

本稿の貢献は，以下の通りである．

- 既知のラベルの集合にもとづくベクトル検索と，LLM を使用したラベル生成を利用することで，自動的なラベル付け手法を提案する．
- ラベル付け手法を用いて，AI セキュリティに関連する文書を収集・分析するプラットフォーム *AISTIP* を提案する．
- 提案プラットフォームは，ラベル付けの精度と分類の効率のバランスを取りながら，文書にラベルを割り当てる．評価実験の結果，提案プラットフォームではベースライン手法と比較してラベル付けの高い精度を得た．

## 2. 関連研究

関連研究として，文書のラベルを予測する手法をまとめる．テキストラベルを予測する既存手法は，大きく分けて 2 つのアプローチに分類できる．1 つ目は深層学習を利用するアプローチ，2 つ目は LLM を利用するアプローチである．2020 年前半頃は深層学習にもとづくアプローチが使用されていた．その後 2020 年代に LLM が広く普及したことで，文書のラベル予測においても LLM にもとづくアプローチが採られている．以下に，それぞれのアプローチの詳細を示す．

### 2.1 深層学習を利用するラベル分類

文章のラベル分類タスクにおいては，従来手法として深層学習モデルを利用するアプローチが利用されている．文献 [1] の調査によると，畳み込みニューラルネットワーク [2] や再帰型ニューラルネットワーク [3] など，いくつかの種類の深層学習モデルを利用したラベル分類手法が提案されている．これらのモデルを用いることで，文書の内容を考慮して，予め定められたラベル集合の中から文書に最適なラベルを付与することができる．より具体的には，教師用サンプルとそれに対応するラベルの情報を複数用意し，訓練データを構成する．訓練データを用いてモデルを訓練することで，分類モデルを作成する．深層学習モデルを利用するアプローチの課題としては，訓練データセットに含まれない未知の内容に対するラベル付けが難しい点が挙げられる．モデルを訓練するために十分な数の教師サンプルデータが必要であるため，新しいラベルを付与する場合にはそのためのデータセットを準備する必要がある．

ところが，本稿で対象とする AI セキュリティのような分野の場合，発表される文献の数が十分でないばかりでなく，新しい概念や内容を含む文献については非常に少ないと考えられる．こうした新しい概念や内容について，そのラベルを人間が把握し，そのラベルを付与するためのデータセットを収集するのは，現実的ではない．そのため，深層学習モデルを利用するアプローチを，本稿で提案するプラットフォームに適用するのは難しいと言える．

### 2.2 LLM を利用するラベル分類

TnT-LLM [4] は，LLM を利用したテキストマイニング手法である．TnT-LLM で提案される手法は，分類生成とテキスト分類の 2 つのフェーズで構成される．分類生成フェーズでは，与えられた文章をいくつかのチャンクに分割し，それぞれのチャンクに対して LLM を用いて要約する．これらの要約から，LLM を使用してラベルを作成する．テキスト分類フェーズでは，分類法生成フェーズで生成されたラベルにもとづいて，LLM を用いて疑似ラベルを生成する．ここで生成した疑似ラベルを利用して，大規模なデータセットを作成する．ここで作成したデータセットを用いて軽量なテキスト分類モデルを訓練することで，ラベル分類モデルを作成する．軽量なテキスト分類モデルは，疑似データを利用して生成された大規模なデータセットを利用してで訓練されるため，高品質なラベル付けを実現する．

しかし，TnT-LLM 手法においても，新しいラベルを追加する際に，テキスト分類モデルを再学習するための運用プロセスが必要になる問題が生じる．軽量なテキスト分類モデルとは言え，運用中のシステムで利用するモデルを高頻度に再学習して更新するのは現実的ではない．また，モデルの更新には一定の時間がかかる．したがって，モデル

表 1: 既存手法に対する提案プラットフォームの特徴.
**Table 1** Features of the Proposed Platform Compared to the Existing Methods.

| 設定 | 既知のラベル | 新しいトピック | モデル再学習なし |
|---|---|---|---|
| モデルベース [1] | ✓ | | |
| TnT-LLM [4] | ✓ | ✓ | |
| 提案プラットフォーム | ✓ | ✓ | ✓ |

を再学習することなく，新しい内容を含む文章に対して適したラベルを付与することができる仕組みが求められる．

### 2.3 本稿で提案するプラットフォームに向けて

既存の手法の問題点は，新しい話題を扱う文献に対するラベル付けを，大量の訓練データの作成や，ファインチューニング等を含むモデルの再学習に必要なコストが無いように実装することにある．表 1 に，提案プラットフォームの特徴を既存手法と比較してまとめる．提案するプラットフォームではこれらの問題に対処する．

## 3. 提案手法

本研究では，AI セキュリティに関する文書を収集・分析するためのプラットフォーム “AISTIP” を提案する．

### 3.1 動機

本稿で提案するプラットフォームに取り組む動機は，限られた数の既知のサンプルにもとづいてラベルを付与できるフレームワークを開発することにある．本稿で対象とする AI セキュリティの研究分野においては，代表的なトピックとして，2014 年に初めて指摘された “敵対的サンプル” がある [5]．それ以来，データポイズニングやメンバーシップ推論など，さまざまな他の AI セキュリティリスクが指摘されている．このように発展してきた AI セキュリティ分野における継続的なトレンドを追跡するためには，論文などの文献を整理し，さらなる分析を行うことができるプラットフォームが必要である．本稿では，既知のデータセットにもとづいてラベルを付与するだけでなく，新たなトピックに対応する新しいラベルを生成することを目指す．

### 3.2 概要

図 1 に，提案フレームワークの概要を示す．このフレームワークは，クローラー，ベクトル検索，ラベル生成，およびリランキングの 4 つのコンポーネントで構成される．

クローラーは，インターネットから技術文書（技術論文など）を収集する．

ベクトル検索では，事前にラベル付与に必要な少量の教師データをベクトルデータベースに登録しておく．ラベルを付与する文章に対して，埋め込みモデルを用いて埋め込み表現，すなわちベクトルを取得し，そのベクトルと類似度が高いものを教師データの中から検索する．検索された上位数件に対応づけられるラベルを，ラベル候補として抽出する．

ラベル生成では，文献のセクションごとに要約を作成し，その要約にもとづいて LLM を利用してラベルを生成し，ラベル候補とする．

最後にリランキングでは，ベクトル検索とラベル生成の両方から得られるラベル候補を収集し，どのラベルを文献に割り当てるかを決定する．以下では，4 つのコンポーネントの詳細を示す．

### 3.3 クローラー

クローラーでは，Web サイトから文献を収集する．文献を収集するにあたって，2 つ留意点がある．1 つ目はその Web サイトで公開される文献の内容の信頼性であり，2 つ目はその Web サイトにおけるクローリングや AI 利用に関する利用規約の問題である．両者の留意点に対応するため，クローラーで巡回する Web サイトの一覧をあらかじめ定義し，そこに含まれる Web サイトから利用規約に従って文献を収集するものとする．

### 3.4 ベクトル検索

ベクトル検索では，事前に与える少量の教師データに含まれる既知のサンプルにもとづいて，ラベル候補の集合を出力する．

事前準備として，少量のサンプルをベクトルデータベースに保存する．ここで，$\mathcal{D}$ をベクトルデータベースとする．サンプル $\mathbf{d} \in \mathcal{D}$ は，トリプレット $(\mathbf{t}, \boldsymbol{v}_\mathbf{t}, \mathbf{l}_\mathbf{t})$ として定義される．ここで，$\mathbf{t}$ は文献の内容を表す文章（例えば，論文のアブストラクト），$\boldsymbol{v}_\mathbf{t}$ は埋め込みモデル $\boldsymbol{v}_\mathbf{t} = \mathcal{M}_E(\mathbf{t})$ により得られる $\mathbf{t}$ の埋め込みベクトル，$\mathbf{l}_\mathbf{t}$ は $\mathbf{t}$ に割り当てられたラベルを示す．

ベクトル検索時には，まず与えられた文章 $\mathbf{t}'$（例えば，論文のアブストラクト）を，埋め込みモデル $\mathcal{M}_E$ を用いて $\boldsymbol{v}_{\mathbf{t}'} = \mathcal{M}_E(\mathbf{t}')$ を算出し，埋め込みベクトル $\boldsymbol{v}_{\mathbf{t}'}$ を得る．次に，与えられた埋め込みベクトル $\boldsymbol{v}_{\mathbf{t}'}$ にもとづき，ベクトルデータベース $\mathcal{D}$ に含まれる埋め込みベクトルと比較することで，関連するサンプルを抽出する．ここで，$\sigma$ をコサイン類似度の関数とするとき，与えられた文章とベクトル

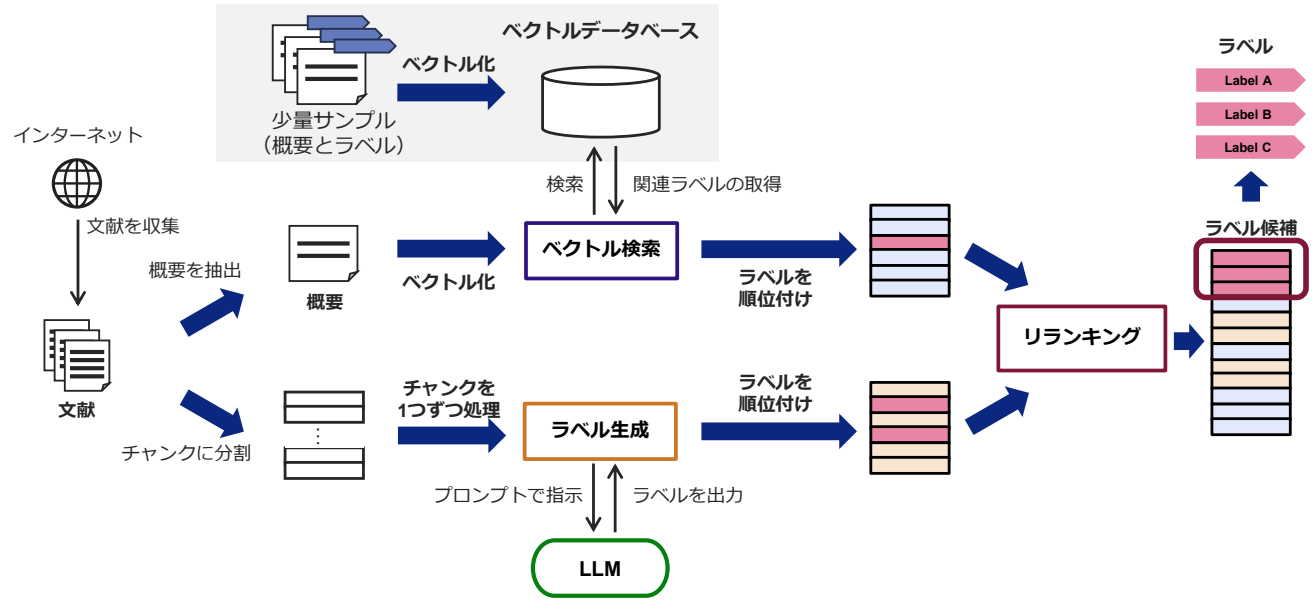


図 1: 提案フレームワークの概要.

**Fig. 1** Overview of the Proposed Platform.

**Algorithm 1** ラベル生成の流れ.

**Input:** ドキュメント Doc
**Output:** ラベルの集合 $L_l$

1: $\mathcal{C} \leftarrow \{c : \text{Doc の文章をチャンクに分割する }\}$.
2: $L \leftarrow \emptyset$
3: **for** $c$ in $\mathcal{C}$ **do**
4: $\quad \mathbf{l}_c \leftarrow$ LLM を用いて $c$ のラベルを予測する.
5: $\quad$ **if** キー $\mathbf{l}_c$ が $L$ に存在する場合 **then**
6: $\quad\quad L[\mathbf{l}_c] \leftarrow L[\mathbf{l}_c] + 1$
7: $\quad$ **else**
8: $\quad\quad L[\mathbf{l}_c] \leftarrow 1$
9: $\quad$ **end if**
10: **end for**
11: $L_l \leftarrow L$ のキーを割り当てられた値の降順でソートする.
12: **return** $L_l$

データベース内のサンプルとの関連性は，それぞれの埋め込みベクトルを用いて $\sigma(\boldsymbol{v}_{\mathbf{t}'}, \boldsymbol{v}_{\mathbf{t}})$ により導出される．この関連性の指標を用いて，ベクトルデータベース内の各サンプルを降順に並べ替えたランキングを作成する．ここで作成したランキングを，ラベル候補のリスト $L_v$ として，出力する．

### 3.5 ラベル生成

ラベル生成では，LLM を用いて新たに生成されたラベルの集合を出力する．

アルゴリズム 1 に，ラベル生成の処理の流れを示す．まず 1 行目において，与えられた文献をチャンクに分割する．このチャンクは，与えられた文献に含まれる章構成，または最大の文字列長で分割するものとする．次に各チャンクに対して 4 行目で LLM を用いることで，ラベルを生成する．ここでチャンクに割り当てられたラベルの数を，5 行目から 9 行目において数えあげる．最後に 11 行目において，これまで生成されたラベルを，割り当てられたラベルの数で降順に並べ替えることで，ラベル候補のリスト $L_l$ を出力する．

### 3.6 リランキング

リランキングでは，ベクトル検索およびラベル生成で出力されたラベル候補のリストに対して，再度順位付けをして並べ替えることで，文献に割り当てるべきラベルを決定する．

$\mathcal{M}_r$ をリランキングモデルとする．このリランキングモデルは，既に順位付けされた複数のラベル候補のリストを受け取り，文献に割り当てるものとしてふさわしいラベルを順に並べ替えたラベルリストを返す．すなわち，ラベルリスト $L$ は，リランキングモデルを用いて $L = \mathcal{M}_r(\{L_v, L_l\})$ により導出される．

リランキングモデルについては，具体的には Reciprocal Rank Fusion (RRF) モデル [6] や，エンコーダを利用したリランキングモデル [7]，および LLM を利用したリランキングモデル [8] など，いくつかの種類が存在する．このうち本稿では，RRF モデルと LLM モデルに着目する．

RRF モデルでは，複数のラベル候補のリストから得られた順位にもとづき，最終的な順位が再評価される．ここで，$RRF(\mathbf{l})$ をラベル $\mathbf{l}$ のスコアとする．このスコアは，$RRF(\mathbf{l}) = \sum_i 1/(k + rank_i(\mathbf{l}))$ によって算出される．なお，$k$ はパラメータであり，$rank_i$ は $i$ 番目のラベル候補のリストにおけるラベル $\mathbf{l}$ の順位を表す．RRF モデルのスコアでは，それぞれのラベル候補のリストにおいて順位が高い (すなわち，順位の数値が小さい) ラベルほど，ス

表 2: 実験で使用したモデル.
**Table 2** Models Used in the Experiments.

| モデル | 名称 |
|---|---|
| 埋め込みモデル | Multilingual-E5-Large [9] |
| LLM | GPT-4o mini [*1] |

表 3: 実験で適用した設定.
**Table 3** Settings Used in the Experiments.

| | ベースライン | | 提案手法 | |
|---|---|---|---|---|
| 設定 | $S_{\mathrm{VR}}$ | $S_{\mathrm{LG}}$ | $S_{\mathrm{RRF}}$ | $S_{\mathrm{LLM}}$ |
| ベクトル検索 | ✓ | | ✓ | ✓ |
| ラベル生成 | | ✓ | ✓ | ✓ |
| ハイブリッド | | | RRF | LLM |

コアが大きい値となる．各ラベル $l$ に対して算出されたスコア $RRF(\mathrm{l})$ にもとづいてラベルを降順に並べ替えることで，ラベルリストを出力する．

LLM モデルでは，プロンプトを用いてラベル候補のラベルの並べ替えを指示する．より具体的にはまず，文献の概要を表す文章と，複数のラベル候補のリストから得られたラベル候補とを含むプロンプトを作成する．このプロンプトでは，文献の概要にもとづいて，ラベル候補のリストの中から，文献に割り当てるのにふさわしい順にラベルを選ぶよう LLM に指示する．このプロンプトを LLM に与え，応答を受け取る．この応答の文章の中から，LLM により選択されたラベルを抽出し，それらをラベルリストにまとめて出力する．

最後に，ラベルリストの中から，文献に対して割り当てるラベルを決定する．ここで，割り当てるラベルの数をパラメータ $N$ として定める．ラベルリストは，文献に割り当てるのにふさわしい順に並べられている．そのため，上位 $N$ 件のラベルを選択して，割り当てるラベルとして決定する．

## 4. 評価実験

提案プラットフォームを対象に，実際にデータセットを用いた実験を通じて評価する．評価実験では，以下の観点 (Research Question) に着目して評価する．

**RQ1.** 提案プラットフォームは，どの程度正確にラベルを割り当てることができるか.

**RQ2.** 提案プラットフォームを使用して生成できる新しいラベルの数はどれほどか.

### 4.1 設定

提案プラットフォームは Python を使用して実装した．表 2 に，実装したプラットフォームで使用するモデルを示す．なお，本実験では少量の教師データに含まれない未知の内容に対しても柔軟にラベルを割り当てることを期待するため，温度パラメータを 0 に設定した．

ベクトルデータベースには，304 の教師データを格納した．この教師データには，AI セキュリティ分野における代表的な論文のアブストラクトと，それに割り当てたラベルの情報を含む．ラベルは，手動で割り当てることで作成した．

本実験では，オンラインプレプリントの Web サイトである arXiv [*2]から，AI セキュリティに関連する論文を収集した．収集には arXiv で用意された API を利用しており，論文を収集するための条件として “AI” と “セキュリティ” に関連するキーワードをタイトルまたはアブストラクトに含むものを指定した．論文の arXiv への登録日時が 2025 年 1 月 1 日から 2025 年 2 月 28 日のものを対象として，合計 109 件の論文を収集した．

表 3 に，実験で適用した設定の詳細を示す．設定として，ベースラインで 2 種類，提案プラットフォームで 2 種類を使用した．ベースライン手法の設定としては，設定 $S_{\mathrm{VR}}$ ではベクトル検索だけを使用し，設定 $S_{\mathrm{LG}}$ ではラベル生成のみを使用する．提案プラットフォームの設定としては，設定 $S_{\mathrm{RRF}}$ と $S_{\mathrm{LLM}}$ の両方について，ベクトル検索とラベル生成の双方を利用するものとしている．2 つの設定の異なる点としては，ラベル候補のセットを統合する方法にある．具体的には，設定 $S_{\mathrm{RRF}}$ では RRF [6] を使用し，設定 $S_{\mathrm{LLM}}$ では LLM を使用してラベルを決定する．各設定では，$N = 3$ のパラメータとして，最大で 3 つのラベルを割り当てることとした．

表 4: ラベル割当ての精度.
**Table 4** Accuracy of label assignment.

| | Baseline | | Proposed Method | |
|---|---|---|---|---|
| Setting | $S_{\mathrm{VR}}$ | $S_{\mathrm{LG}}$ | $S_{\mathrm{RRF}}$ | $S_{\mathrm{LLM}}$ |
| Accuracy | 0.327 | 0.844 | 0.603 | **0.866** |

### 4.2 ラベル割当ての精度

RQ1 に答えるため，ラベル割当ての精度を評価する．正しいラベルが割り当てられたかをある統一の判断基準で判定するため，LLM-as-a-Judge [10] の方法を利用した．すなわち，LLM に対してプロンプトで指示することで，正しいラベルが割り当てられたかを判定した．モデルとしては，Open AI 社が提供する “GPT-4o mini” [*1] モデルを使用した．評価には，Open AI 社の API ライブラリ [*3]においてデフォルトで提供されるパラメータを適用した．プロンプトは，Appendix A.1 章に示す．

表 4 に，ラベル割当ての精度の評価結果を示す．表に示

[*1] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[*2] https://arxiv.org/

[*3] https://pypi.org/project/openai/

表 5: 割り当てられたラベルの数と新たに生成されたラベルの数.

**Table 5** Number of Assigned Labels and Number of Newly Generated Labels.

| 設定 | ベースライン | | 提案手法 | |
|---|---|---|---|---|
| | $S_{\mathrm{VR}}$ | $S_{\mathrm{LG}}$ | $S_{\mathrm{RRF}}$ | $S_{\mathrm{LLM}}$ |
| # ラベル | 44 | 302 | 173 | 193 |
| # 新しいラベル | 0 | 298 | 137 | 168 |

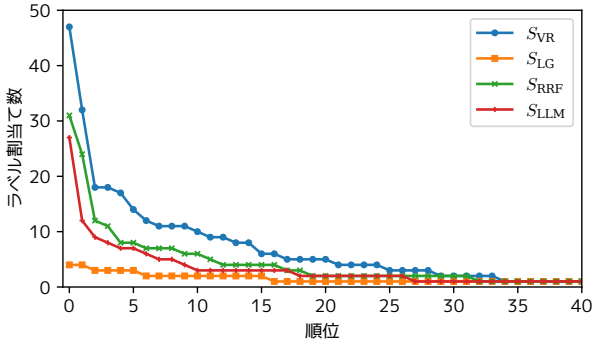


図 2: ラベル割り当ての数.

**Fig. 2** Number of Assigned Labels

される通り，$S_{\mathrm{LLM}}$ の精度が最も高い結果となった．一方，$S_{\mathrm{VR}}$ の精度は最も低い．これは，ベクトル検索がベクトルデータベースから最大 3 つの候補しか収集できないためである．類似性に対するしきい値を設定していないため，関連性のないラベルも取得された．また，本実験でラベル割当ての対象とした論文は 2025 年 1 月〜2 月に投稿された比較的新しいものであり，ベクトルデータベースの内容と合致しないものも一部含むため，ラベル割当ての精度としては低い結果となった．なお，ラベル候補を選択するため，類似性を表す指標にしきい値を設定することも可能であるが，実際にはしきい値にどのような値を設定するかを決定するのは難しい問題である．

$S_{\mathrm{LG}}$ を設定することで，よりベクトルデータベースよりも高い精度を達成した．この改善は，LLM が文献から適切なラベルを抽出する能力によるものである．しかし，後述の実験結果に示すようにほとんどのラベルは一度しか割り当てられなかった．そのため，割り当てられたラベルが，AI セキュリティ分野の体系的な整理に貢献するかは疑問の余地が残る．

### 4.3 ラベルのばらつき

研究課題 RQ2 に答えるため，生成されたラベルのばらつきを評価する．表 5 に，割り当てられたラベルの種類数を示す．この表における “新しいラベル” とは，ベクトルデータベースに存在せず，新たに生成されたラベルを指す．表に示されるように，設定 $S_{\mathrm{VR}}$ はあらかじめ定義されたラベルのみを割り当てるため，“新しいラベル” の数は 0 となる．この設定では事前に決められた種類のラベルを文献に付与するため，文献の体系的な分類に貢献すると考えられる．しかしながら，新しいラベルを導入することはないため，新しい概念に対応できないのが問題となる．


図 3: Web サイト上で公開された AISTIP [*4].

図 4: AISTIP Released on the Website.

対照的に，設定 $S_{\mathrm{LG}}$ で割り当てられたラベルのほとんどは新しいものであった．しかし，ほとんどのラベルは一度だけしか割り当てられない．この設定は，表 4 で示されるように高い精度を示すが，体系的にラベルを割り当てるという観点では，文献のグループ化に利用できない．

図 2 に，ラベルの割り当ての数を示す．横軸は各方法によって割り当てられたラベルの数にもとづくランキングを示し，縦軸は各ラベルの総割り当て数を示す．設定 $S_{\mathrm{VR}}$ では，ほとんどのラベルが複数回割り当てられており，文献を特定のラベルに分類できている．これに対し，設定 $S_{\mathrm{LG}}$ では，各ラベルは最大で 4 回までしか割り当てられなかった．提案手法の設定は他の中間に位置しており，このことより提案手法によって精度と分類のバランスが取れたと言える．

### 4.4 Web サイトへの実装

AISTIP を，“文献データベース” の名称を冠したコンテンツとして Web サイト上で公開している [*4]．Web サイト上では，収集した文献に対して，最大で 3 つまでのラベルを付与する．

文献を収集し，そのカテゴリ情報等を提供する類似の Web サービスとして，huggingface.co における “Trending Papers” [*5]や，alphaXiv[*6]が挙げられる．“文献データベース” がこれらのサービスと異なる点としては，AI セキュリティに特化した文献のみを収集している点と，割り当てられたラベルには AI セキュリティに関連するキーワードが多く含まれる点が挙げられる．

*4 https://aisecurity-portal.org/literature-database/
*5 https://huggingface.co/papers
*6 https://www.alphaxiv.org/

### 4.5 考察

#### 4.5.1 既存手法との比較

表 1 に，既存手法との比較を示す．提案プラットフォームは，ファインチューニングなどのモデルの再学習なしで新しい内容の文書に対してその内容に合わせたラベルを割り当てることができる．こうしたプラットフォームを安定的に運用する場合，モデルを更新するたびに精度を詳細に評価することが難しいため，頻繁にモデルを更新するのは現実的ではない．そのため，できるだけ再学習することなく運用できるプラットフォームが望ましい．その点で，提案プラットフォームはモデルを再学習することなく，ラベル付けの精度とばらつきのバランスを取ることができるのが特徴である．

#### 4.5.2 制約

LLM によって生成されたラベルを整合させることは，大きな課題である．LLM は確率的モデルであるため，その出力は大きく変動することがある．この問題は LLM に与えるプロンプトを洗練させることである程度解決できる可能性があるが，より根本的な解決策が必要である．

特に，表記ゆれへの対応が課題となる．表記ゆれには大きく分けて，英語から日本語への翻訳に起因する問題と，日本語としての表記ゆれの問題，そして英語での表記ゆれの問題がある．

英語から日本語への翻訳に起因する問題としては，文献に表記された英単語を日本語表現でのラベルに変換する際の表記ゆれがある．例えば，“Membership” という単語については，日本語に翻訳する場合，長音を入れるか否かで “メンバーシップ” と “メンバシップ” の表記が考えられる．どちらもそれ単体では日本語として問題ないものの，体系的整理を目的としたラベル割当てにおいては表記の統一が必要となる．

日本語としての表記ゆれは，例えば “機械学習” と “機械学習技術” のラベルのように，直後に “技術” などの単語が続くか否かのようなものを指す．一律に短い単語が好ましい訳ではなく，例えば “敵対的攻撃” と “敵対的攻撃検出” とでは，前者は攻撃手法，後者は防御 (検出) 手法を指すため，反対の手法を指す．そのため，これらを考慮する必要がある．

英語での表記ゆれは，例えば “Adversarial Training” と “Adversarial Learning” のようなものを指す．特に新しい内容を扱う論文では用語が明確に定まっていない可能性があり，英語の文章においても表記ゆれが含まれる場合がある．

以上の表記ゆれの問題への対応は，今後の課題である．

#### 4.5.3 将来の展開

上記の制約に対処することに加えて，ここでは他に考えられる将来の展開を概説する．

提案プラットフォームを通じて収集された文書とラベルは，AI セキュリティにおける研究を始めたばかりの研究者にとって有益なものであり，新しい研究トピックを探している研究者にも役立つ．我々はこのプラットフォームを Web サイトとして公開しているが，今後も更新を継続する予定である．

Web サイトに関連する将来の展開として挙げられるのは，多言語への対応である．分析された情報を様々な言語で提供することで，多くの人々が AI セキュリティ分野をよりよく理解できるようになる．現時点で Web サイトに公開したプラットフォームでは，英語の文献に対して日本語のラベルを割り当てることを想定しており，一部英語版のページを公開するにとどまる [*7]．特に文献については，多言語への対応を目指す．

もう一つの将来の展開として，AI セキュリティの分野を超えた拡張が挙げられる．提案プラットフォームは，ベクトルデータベースと LLM のハイブリッド利用によって特徴付けられる．この機能により，プラットフォームは AI セキュリティの領域だけでなく，他の新たなトピックにも幅広く適用できることが可能である．

## 5. おわりに

本稿では，AI セキュリティに関連する文書を収集し，文書にラベルを付けるためのプラットフォーム *AISTIP* を提案した．主な特徴は，ベクトル検索とラベル生成器を組み合わせることで，正確にラベルを付与し，新たな内容の文献に対して新しいラベルを付与できることにある．実験結果から，提案プラットフォームが設定 $S_{\mathrm{LLM}}$ を利用した際に，ラベル付けの精度が最も高い結果を得た．また，ラベルを活用した文献の分類においては，多数の種類のラベルを生成するよりも既存のラベルを活用することが求められるが，割り当てられたラベルのばらつきの観点でも提案手法が精度とのバランスに優れていることを示した．今後の取り組みとしては，Web サイトとして一般公開した提案プラットフォームについて，表記ゆれの統一や多言語への対応などの課題に取り組むとともに，提案プラットフォームを AI セキュリティ以外の領域への応用を拡大することが挙げられる．

**謝辞** 本研究は，JST 経済安全保障重要技術育成プログラム【JPMJKP24C4】の支援を受けたものです．

### 参考文献

[1] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M. and Gao, J.: Deep Learning–based Text Classification: A Comprehensive Review, *ACM Comput. Surv.*, Vol. 54, No. 3 (online), DOI: 10.1145/3439726 (2021).

[2] Kalchbrenner, N., Grefenstette, E. and Blunsom, P.: A

[*7] https://aisecurity-portal.org/en/literature-database/

Convolutional Neural Network for Modelling Sentences, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Toutanova, K. and Wu, H., eds.), Baltimore, Maryland, Association for Computational Linguistics, pp. 655–665 (online), DOI: 10.3115/v1/P14-1062 (2014).

[3] Tai, K. S., Socher, R. and Manning, C. D.: Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Zong, C. and Strube, M., eds.), Beijing, China, Association for Computational Linguistics, pp. 1556–1566 (online), DOI: 10.3115/v1/P15-1150 (2015).

[4] Wan, M., Safavi, T., Jauhar, S. K., Kim, Y., Counts, S., Neville, J., Suri, S., Shah, C., White, R. W., Yang, L., Andersen, R., Buscher, G., Joshi, D. and Rangan, N.: TnT-LLM: Text Mining at Scale with Large Language Models, *Proc. 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5836–5847 (online), DOI: 10.1145/3637528.3671647 (2024).

[5] Goodfellow, I. J., Shlens, J. and Szegedy, C.: Explaining and Harnessing Adversarial Examples, *arXiv preprint arXiv:1412.6572* (2015).

[6] Cormack, G. V., Clarke, C. L. A. and Buettcher, S.: Reciprocal rank fusion outperforms condorcet and individual rank learning methods, *Proc. 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759 (online), DOI: 10.1145/1571941.1572114 (2009).

[7] Xiao, S., Liu, Z., Zhang, P., Muennighoff, N., Lian, D. and Nie, J.-Y.: C-Pack: Packed Resources For General Chinese Embeddings, *Proc. 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 641–649 (online), DOI: 10.1145/3626772.3657878 (2024).

[8] Sharifymoghaddam, S., Pradeep, R., Slavescu, A., Nguyen, R., Xu, A., Chen, Z., Zhang, Y., Chen, Y., Xian, J. and Lin, J.: RankLLM: A Python Package for Reranking with LLMs, *Proc. 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3681–3690 (online), DOI: 10.1145/3726302.3730331 (2025).

[9] Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R. and Wei, F.: Multilingual E5 Text Embeddings: A Technical Report, *arXiv preprint arXiv:2402.05672* (2024).

[10] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E. and Stoica, I.: Judging LLM-as-a-judge with MT-bench and Chatbot Arena, *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23 (2024).

# 付　　録

## A.1 LLM-as-a-Judge で使用したプロンプト

Listing 1, 2 に, LLM-as-a-Judge で使用したプロンプトを示す．Listing 1 は，システムプロンプトである．Listing 2 は, 実際にラベルが適切かを判定する際に用いるプロンプトである．"abstract" は文献の概要に, "keyword_candidates" は割り当てられたラベルのリストを箇条書き形式で表記した文字列に置き換えられる．

Listing 1: LLM-as-a-Judge で使用したプロンプト（システムプロンプト）.

```
You are a keyword evaluation assistant. Your task
    →is to assess the relevance of a list of
    →keyword candidates to a given text. For each
    →keyword candidate, determine if it is
    →appropriate as a keyword for the text.

Please follow these instructions:

1. You will be provided with a paragraph of text.
2. You will also receive a list of keyword
    →candidates associated with that text.
3. For each keyword candidate, decide if it
    →accurately reflects the main topics or themes
    → of the text.

Respond in the following JSON format:

```json
[
    {
        "candidate": "Keyword candidate 1",
        "is_relevant": true/false,
        "reason": "Brief explanation for the
    →relevance judgment."
    },
    {
        "candidate": "Keyword candidate 2",
        "is_relevant": true/false,
        "reason": "Brief explanation for the
    →relevance judgment."
    },
    ...
]
```

For each keyword candidate, provide whether it is
    →relevant or not along with a reason for your
    →assessment. Keep your evaluation concise and
    →clear.
Note that the output must written in the JSON
    →format that is parsable.
```

Listing 2: LLM-as-a-Judge で使用したプロンプト（ユーザプロンプト）.

```
### Paragraph
{abstract}

### Keyword candidates
{keyword_candidates}
```