《様式Ｂ》

研究テーマ	「クラウドサービスへのオフローディングを活用した機械学習に基づく軽量 NIDS の開発」		
研究責任者	所属機関名	豊橋技術科学大学	
	官職又は役職	准教授	
	氏　　名	中村 純哉	メールアドレス junya@imc.tut.ac.jp
共同研究者	所属機関名		
	官職又は役職		
	氏　　名		

（令和 4 年度募集）第 35 回 助成研究 完了報告書

上記様式記載後

１．実施内容および成果ならびに今後予想される効果の概要（１，０００字程度）

※産業技術として実用化の可能性や特許出願（予定も含む）の有無についてもご記載ください。

ネットワーク侵入検知システム（NIDS）は、組織のネットワークを流れるトラフィックを監視し，サイバー攻撃を早期に検知するための重要なセキュリティシステムである．従来の NIDS は既知の攻撃パターンを識別するためにシグネチャベースの手法を採用していたが，新たな攻撃や変異型には対応できないという問題があった．この問題に対処するため，機械学習を用いて未知の攻撃を検出する技術が開発されているが、これらのシステムは高い計算資源を必要とするため，導入が難しい環境も多い．

本研究では，機械学習に基づく NIDS の計算負荷を軽減し，より広範な環境での利用を可能にするために，クラウドサービスへのオフローディングを行うプロトタイプを開発した．プロトタイプは Amazon Web Services（AWS）上に構築され，分散ストリーミング処理システムとして Amazon MSK を採用する．分散処理基盤である Amazon EMR，機械学習モデルの訓練及び推論を行う Amazon Sagemaker，データの非同期処理を支える AWS Lambda と組み合わせることで，全体の処理負荷を抑えることを目指す．

提案システムは，ローカルネットワークから得られたセッション情報をクラウド上の Amazon MSK に送信する．Amazon EMR で特徴量を計算した後，Amazon MSK を通じて AWS Lambda にデータを送る．AWS Lambda は Sagemaker によってセッションが攻撃かどうか推論を行い，最終的な分類結果を AWS S3 に保存する．このプロセスは，高速かつス

ケーラブルであり，クラウドリソースの動的なスケーリングによって必要に応じてリソースを割り当てることができるという特徴を備える．

評価実験では，UNSW-NB15 データセットを使用し，システムの各部分でスループットとレイテンシを計測した．実験結果から，Amazon EMR から AWS Lambda への通信がボトルネックとなっていることが判明し，最大処理能力は約 5000 セッション/秒だった．この性能は，AWS Lambda のプロビジョニングを最適化することで改善する可能性がある．

本研究によって，クラウドオフローディングを活用した機械学習に基づく NIDS によって，リソースが限られた環境でも高度なセキュリティ対策を十分に実現できる可能性があることが示唆された．今後もシステムの検証と最適化を更に進めることで，機械学習に基づく NIDS の実用化を目指していきたい．

２．実施内容および成果の説明（Ａ４で、５ページ以内）

はじめに

ネットワーク型侵入検知システム（NIDS）は，ネットワークを流れるトラフィックを監視し，組織内の機器に向けられたサイバー攻撃を検知するセキュリティシステムである．このシステムを利用することによりサイバー攻撃を早期に気づくことができ，攻撃の被害を抑えることができるため，有効なセキュリティ対策として広く利用されている．従来の NIDS はシグネチャと呼ばれるデータベースに含まれた攻撃しか検知することができなかったが，近年では，サイバー攻撃の特徴を学習した機械学習モデルを利用することで未知の攻撃も検知できる，機械学習に基づく NIDS が注目を集めている．しかし，機械学習に基づく NIDS は動作のために高い計算能力を備えたコンピュータが必要であり，導入できる環境が限られている．例えば，家庭や工場などの環境で利用することは難しい．本研究では，機械学習に基づく NIDS の中で高い計算能力を必要とする処理をクラウドサービスへオフローディングする NIDS のプロトタイプを作成し，その性能を評価する．

システム構成

本研究では，多田らが提案した機械学習に基づく NIDS の分散処理フレームワーク[2]（図 1）をもとに，Amazon Web Services（AWS）上に機械学習に基づく NIDS を構築した．処理件数の変化に応じて適切な量のリソースを配分するために，それぞれのクラウドサービスにはストレージサイズやインスタンス数の動的スケーリングが可能であるマネージドサービスを用いた．図 2 に，本研究で構築した NIDS の構成を示す．

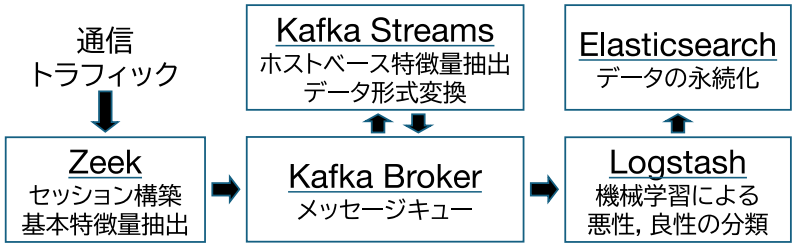

図 1 多田らのフレームワークの構成

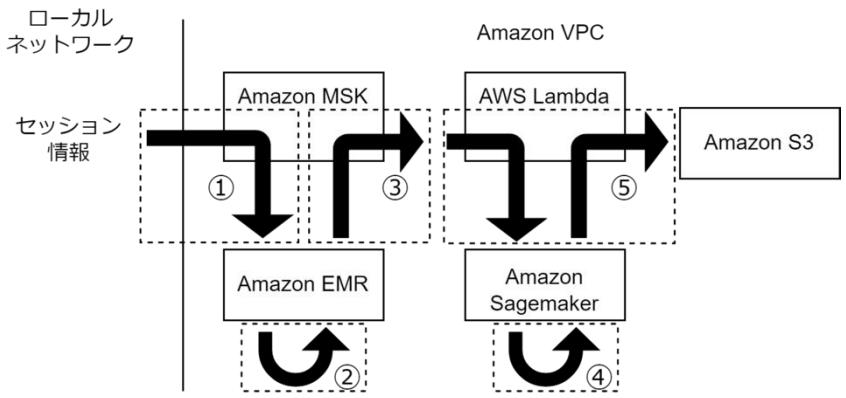

図 2 提案システムの構成

Amazon EMR は分散処理フレームワークを実行するためのマネージドサービスであり，インスタンス数やストレージのサイズのオートスケーリングが可能である．機械学習モデルによる推論のための特徴量を抽出するために利用した．Amazon Sagemaker は，AWS 上で機械学習モデルを利用するための一連の流れを実行できるフルマネージドサービスである．訓練済みの分類モデルをデプロイすることにより，推論のためのエンドポイントとして利用した．Amazon MSK は，ストレージのオートスケーリングが可能なメッセージキューのフルマネージドサービスである．オンプレミス環境からクラウド環境へのセッション情報の送信処理から，Amazon Sagemaker での推論処理までの間の非同期処理を実現するために利用した．AWS Lambda はイベントの発生に応じてプログラムを実行可能なサービスであり，イベントの発生件数に応じて同時実行数のオートスケーリングが可能である．Amazon MSK と Amazon Sagemaker の間のデータの中継に利用した．

本システムにおける処理の流れは，次のようになる．まず，ローカルネットワークで得られたセッション情報を基本特徴量として Amazon MSK の Kafka Broker に送信する．次に，Amazon EMR 上に構築した Apache Spark にて 機械学習モデルの推論に必要な特徴量を抽出し，完全な特徴量として Amazon MSK に書き戻す．完全な特徴量は AWS Lambda によって Amazon Sagemaker の推論エンドポイントに送られ，事前に訓練した機械学習モデルで攻撃通信かどうかを分類する．分類結果は，AWS Lambda を通じて Amazon S3 に保存される．

評価実験

本研究で作成した機械学習に基づく NIDS の処理性能を評価するために，図 2 に示す各区間及び全体でのスループットとレイテンシを計測する．本実験では，スクリプト作成のための時間的都合及び後述するデータセットにすでに特徴量抽出済みのデータが含まれていることから，ネットワークトラフィックからセッション情報を抽出する部分および Amazon EMR により特徴量抽出の部分を除いて実装した．計測のためのセッション情報には，Amazon VPC 上の Amazon EC2 インスタンスから送信した UNSW-NB15 データセット[1]を用いた．Amazon Sagemaker の推論エンドポイントには，scikit-learn と UNSW-NB15 データセットによって事前に訓練したランダムフォレストのモデルを用いた．

レイテンシの計測結果を表 1 に，スループットの計測結果を表 2 に示す．表 1 より，処理件数が増加すると，Amazon EMR から AWS Lambda への通信のレイテンシが非常に長くなっていることがわかる．区間 2 は，Amazon EMR での特徴長抽出処理を実装していないため，レイテンシが 0.00 ms となっている．それ以外の区間は，処理件数の増加に対するレイテンシの増加は区間 3 と比較して緩やかである．また，表 2 より，区間 3 以降のデータ流量が約 5300 セッション毎秒に抑えられていることがわかる．このことから，本研究で作成した NIDS では区間 3 での通信が全体の処理性能のボトルネックとなっていると判断できる．

表 1 各区間でのレイテンシ［ms］

送信速度 ［ｾｯｼｮﾝ毎秒］	区間 1	区間 2	区間 3	区間 4	区間 5	全区間
1	2264.42	0	1074.22	107.42	0.18	3446.24
10	2021.44	0	1062.33	115.85	0.23	3199.85
100	1798.38	0	1063.1	168.62	1.31	3031.41
1000	1674.28	0	1110.24	223.92	12.5	3020.93
10000	4595.6	0	52332.52	237.48	63.46	57229.06

表 2 各区間でのスループット［セッション毎秒］

送信速度［ｾｯｼｮﾝ毎秒］	区間 1	区間 2	区間 3	区間 4	区間 5	全区間
1	0.98	1.03	1.03	1.03	1.03	0.98
10	9.9	10.31	10.28	10.32	10.32	9.86
100	93.78	97.61	97.26	97.67	97.67	93.4
1000	961.7	1000.66	996.98	1001.04	1001.04	957.65
10000	9236.7	10367.73	5281.7	5313.61	5313.61	4965.88

区間 3 がボトルネックとなった原因としては，AWS Lambda においてコールドスタートが多く発生していることが挙げられる．AWS Lambda では，あらかじめ規定数のインスタンスを用意しており，AWS Lambda に処理が要求されると空いているインスタンスにその処理が割り振られる．大量の要求が短時間に届き，空いているインスタンスが不足してしまった場合には， AWS Lambda は新しくインスタンスを作成して要求の増加に対応しようとする．しかし，インスタンスの起動には数分の時間がかかるため，本実験では処理性能が改善しなかったと考えられる．そのため，想定される処理件数に応じて AWS Lambda の関数インスタンスをプロビジョニングすることによって，処理性能が改善する可能性がある．

おわりに

本研究では，クラウドサービスへのオフローディングを活用した機械学習に基づく NIDS のプロトタイプを開発し，その処理性能を評価した．Amazon Web Services 上に構築したプロトタイプは，クラウドサービスで提供されるマネージドサービスを多く採用しており，わずかな運用負荷で機械学習に基づく NIDS を実現できると期待される．性能評価では，Amazon EMR から AWS Lambda への通信がボトルネックとなり，最大スループットは約 5000 セッション毎秒であった．より多くのデータの処理を可能にするためには，Amazon EMR から AWS Lambda への通信のレイテンシを抑えることが必要である．

参考文献

1. N. Moustafa and J. Slay. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proc. MilCIS 2015, pp. 1–6, 2015.

2. 多田，中村，大村，小林. 機械学習ベース NIDS 構築のための分散処理フレームワーク. 情報処理学会論文誌, 60(9):1448‐1465, 2019.