平成 30 年度

修士学位論文

機械学習手法を用いた
ネットワークトラフィックによる
マイニング検出方式の検討

A Network Traffic Based Mining Detection Method on a Machine Learning Approach

1215080　　合田 亮登

指導教員　　清水 明宏

高知工科大学大学院 工学研究科 基盤工学専攻

情報学コース

要 旨

機械学習手法を用いた
ネットワークトラフィックによるマイニング検出方式の検討

合田 亮登

Bitcoin に代表される暗号通貨の普及に伴い，マイニングとよばれる通貨の採掘行為が盛んに行われるようになっている．マイニングは，暗号通貨取引の正当性を検証する行為であり膨大な計算資源を必要とするが，その報酬として実行者は暗号通貨を得ることができる．そこで報酬目的に，不正な手段で計算資源を獲得しマイニングを行う攻撃が登場している．例としては，Web サイトにマイニングを実行するスクリプトを仕掛け閲覧者のリソースを利用するものや，VPS や公共施設等の資源を提供者の意図に反して利用するものが挙げられる．

従来の攻撃対策として，セキュリティソフトの導入や利用規約の策定が行われているが，そのような対策だけでは必ずしも十分ではない．

本研究ではマイニングで発生する通信に着目し，ネットワークトラフィック分析によりマイニングを検出する手法を検討する．

その結果，95%の精度で暗号通貨のマイニングトラフィックである Coinhive と CryptLoot の通信を識別できることが実現でき，実運用上十分な精度となった．

キーワード　暗号通貨, マイニング, ネットワークトラフィック検出

Abstract

A Network Traffic Based Mining Detection Method on a Machine Learning Approach

Ryoto GODA

With the spread of crypto currencies such as Bitcoin, mining for earning incentives are actively performed. Minings require an enormous amount of computing resources for verifying crypto currency transactions. For that purpose, there is an attack where attackers try to acquire computational resources with illegal means.

As an example, there are things that use a resource of browsers who set up a script to perform mining on a Web site, And those that use resources such as VPS and public facilities opposed to the intention of the provider.

Introduction of security software and formulation of terms of use are being carried out as measures against the previous attacks, Such measures alone are not always sufficient.

In this research, focusing on communication generated by mining, we examine a method to detect mining by network traffic analysis.

As a result, it was possible to identify Coinhive and Crypt-Loot communication which is mining traffic of encryption currency with an accuracy of 95 %, and it was sufficient accuracy for practical operation.

key words crypto currency, mining, network traffic detection

目次

図目次

表目次

第 1 章

序論

1.1 本研究の背景

Bitcoin に代表される暗号通貨は，トランザクションを記録するためにブロックチェーン技術を使用するデジタル通貨である．Bitcoin は国家による価値の保証が無いものの，国によっては法定通貨に交換可能であり，時価総額は 2017 年 4 月に 3 兆円を超え，2019 年 1 月時点では 3.7 兆円と非常に大きい [1][2].

ブロックチェーン技術により，トランザクションを参加者間で共有することで，取引記録の改ざん等の不正利用を防止することができる．そしてマイニングという，新しい取引を検証しブロックチェーンに記録する，ブロックチェーンを分散型のシステムで管理するための仕組みが導入されている [3][4]．検証には暗号化ハッシュアルゴリズムに基づいて計算量の大きい問題を解く必要があり，検証に成功した場合，2 つの報酬を受け取ることができる．一つ目は，ブロック生成報酬であり，検証する暗号通貨が発行される限り受け取ることができる．二つ目は，取引手数料であり，最低取引手数料は取引データのサイズによって決定される．

しかし今日では，マイニングの計算難易度は非常に高く，個人の力だけでマイニングの報酬を得ることは非常に困難である．そこで，複数人の計算リソースを組み合わせて集団でマイニングを行い，そのマイニング報酬をハッシュレートなどのマイニング貢献度に応じて分配する，マイニングプールを使用する手法が存在する [5]．よってマイニングプールを使うことで，マイニングの性能と時間に応じて報酬を受け取ることが可能となっている．

また，暗号通貨の一つに，発信者と送信者の身元，及び各トランザクションの金額など，

トランザクションの詳細を匿名になる Monero がある [6]．匿名性が高いことで利用者のプライバシーが保護される一方で，法執行を回避するためや違法行為に使用されおり，2018 年の違法にマイニングされた暗号通貨の中で最も使用されている [7]．

マイニングによって電力や計算資源などの価値を金銭に変換できることを悪用した例として，下記の場合がある．

1. マルウェアに感染したコンピュータがマイニングを行う場合
2. ウェブサイトや通信内容が改ざんされ，埋め込まれたマイニングスクリプトによって閲覧者のコンピュータでマイニングを行う場合
3. VPS(Virtual Private Server) や公共施設などの資源を利用し，資源提供者の意図に反して実行者が金銭目的でマイニングを実行する場合

1 と 2 については，アンチウイルスソフトウェアを端末にインストールすることで，マイニングマルウェアやマイニングの実行を CPU 使用率やプロセス情報などから検出する方法がある．しかし，マイニングマルウェアの中には，検出に耐性のある亜種が存在する [8]．マイニングプールの Proxy サーバを立てることが容易であり，単純なドメイン名を利用したブロックでは通信を防ぐことは困難である [9]．また，アンチウイルスソフトウェアのインストールが困難な IoT 機器が存在する．そこで，端末へのアンチウイルスソフトウェアのインストールを必要としない新しい検出手法を検討し，検出の手段を増やす．

3 については，VPS では 1 台のハードウェアを複数のユーザで共有して使用し，マイニングのために VPS を使用することを利用規約で禁止している場合がある [10]．VPS 契約者本人が意図してマイニングを行う場合や，契約者が意図せずともマルウェアによりマイニングを行う場合に，検出する方法が必要となる．また，公共施設やサービスとして無料で電気が使える場所において，実行者が自分の利益のためにマイニングする行為は，電力提供者の目的に反している可能性がある．これらの場合に，マイニングを検出する方法が必要となる．

尚，本論文中におけるマイニングは全て，仮想通貨におけるブロックチェーンの整合性を検証し，新たなブロックを生成することを指す．大規模データにデータ解析の技法を適応す

ることで有益なデータを取り出す，データマイニングを意味するものではない．

1.2　本研究の目的

前節の理由から，暗号通貨のマイニング行為を従来手法以外にも検討する必要性が高まっている．そこで本研究では，機械学習手法の一つであるディープラーニングを用いて，ネットワークトラフィックからマイニングトラフィックを識別を行う手法を検討することを目的とする．

尚，ネットワークの通信を解析することは法律や憲法の規定する「通信の秘密」と関連するため，通信当事者に個別かつ明示的な合意を得るか，違法阻却事由を満たす必要があるが，本論ではその是非は考慮しない [22][23]．しかし，ネットワークトラフィックから取得する情報は，可能な限り通信利用者のプライバシーを侵害しない必要がある．

1.3　本論文の構成

本論文は，以下の 5 章で構成される．第 1 章では，本研究内容を説明する導入として，研究の背景，目的と論文の構成について述べる．第 2 章では，トラフィック識別の関連研究について述べる．第 3 章では，第 2 章での別関連研究を踏まえ，ネットワークトラフィックからマイニングトラフィックを識別する方法を検討する．第 4 章では，実験結果とその評価を行い，検討手法の効果を考察する．第 5 章では，結論として，まとめと今後の展望について述べる．

第 2 章

関連研究と技術

2.1 関連研究

関連研究として，TCP ヘッダの TTL 初期値によって異常通信を検出する手法が提案されている [12]．この手法は，特別に設定を変更しない限り OS によって使用される TTL の値が固定であり，異常通信には一般的な OS では使用されない値を持つ場合がある特徴を利用している．TCP ヘッダのみの情報で通信を分類し，トランスポート層の情報を利用しないため，暗号化された通信に対しても利用できる．しかし，提案手法の異常通信が発生する環境は，独自のアプリケーションで通信を行う場合であり，一般的な OS のソフトウェアを使用しての通信が発生する場合にこの手法は適さない．TCP ヘッダの他の情報を併用することにより，より高い精度で通信を特定できる可能性がある．

また，TCP 初期シーケンス番号，IP ヘッダの ID 値，や DNS ヘッダの ID などに利用されている固定値を利用している通信パケットを抽出し，ネットワーク上で観測される通信を分類する手法がある [13].

2.2 Coinhive と Crypt-Loot

Coinhive は，任意の Web サイトに埋め込むことができる Monero ブロックチェーンのための JavaScript マイナーを提供するサービスである．多くのマイニングプールを利用している．Web ブラウザ上でネイティブに近いスピードで実行できる，コンパクトなバイナリ形式の低級言語である WebAssembly を使用している [14][15]．WebAssembly は，W3C

WebAssembly Working Group と Community Group によって,Web 標準として開発されており，現在では Mozilla Firefox, Google Chrome, Safari, Edge という主要なブラウザでサポートされている [18]．そのため，Coinhive は幅広い環境で高速に動作する．

Web サイトでは，Web 広告を表示することなく収入を得る手段として使用され，JavaScript マイナー実行者の環境の資源を消費して動作する．実行環境の性能と時間あたりで収入が変化し，Web サイト運営者本人が自分でマイニングすることが可能であることが，一般的な Web 広告と異なる．

誰でも簡単にマイニングを行うことができる反面，Web サーバなどが改ざんされ，攻撃者が利益を得るように Coinhive マイニングスクリプトが埋め込まれる被害が発生している [11].

類似するサービスとして，Crypt-Loot がある [16]．Coinhive の手数料は 30%であることに対し Crypto-Loot では 12%である特徴がある．

本論文では，データセットとして使用するマイニングトラフィックデータを取得するために Coinhive と Cryot-Loot を使用する．2017 年 12 月のマルウェアランキングの 1 位が Coinhive, 3 位 Crypt-Loot であった [17].

2.3　マイニングトラフィック

マイニングトラフィックは初回の認証を除き，下記の 2 つの通信に加えて Ack が送信される通信によって構成される．よって，送信間隔が Ack を除く短期間の 2 つのパケットが重要であると考える．

1. サーバが利用者に検証する値を送信
2. 利用者がサーバに検証結果を送信

マイニングトラフィックの例をとして，COinhive におけるマイニングトラフィックを図**??**に示す．Ack を除き，長さが 267 のサーバが利用者に検証する値を送信する通信と長さが 199 の利用者がサーバに検証結果を送信する通信，そして長さが 81 のサーバが利用者に

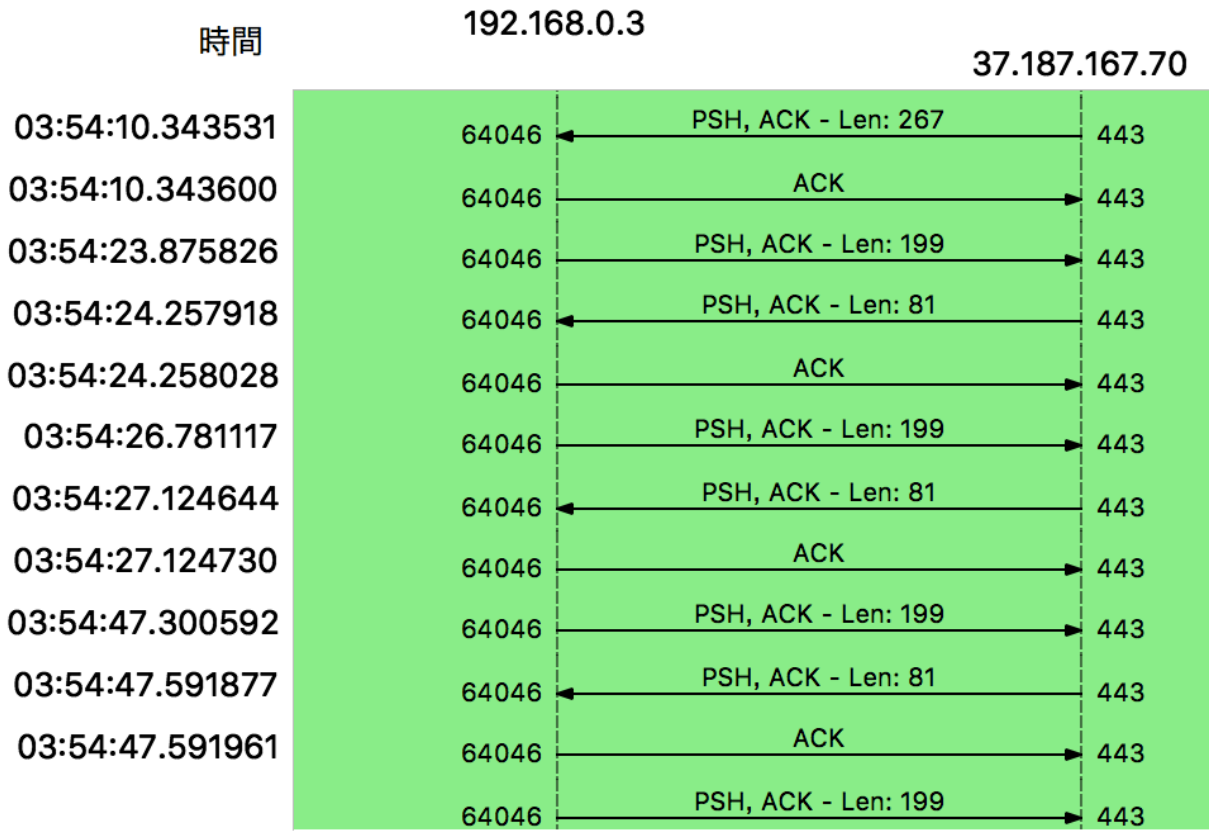

図 2.1　Coinhive におけるマイニングトラフィックの構成

受け取った値を検証した結果を送信する通信から成り立っている．ただし，常にこの固定の長さで通信されるわけではない．

第 3 章

検討する識別手法

本章では，トラフィックデータ識別にするために，DNN の一種であるマルチレイヤパーセプトロン (MLP) による 2 値分類モデルを使用する．

MLP には，tcpdump により取得したデータをを元に作成したデータセットから書きパラメータを入力として与え，トラフィックデータがマイニングトラフィックか否かの識別結果を出力として得る．

データセットには，IP ヘッダ情報から，時間，ttl，id，offset，flag，プロトコル，長さを用いる．さらに，同一の IP アドレスから連続する一つ前に受信した長さのデータを用いる．

マイニングを行うトラフィックとして，Coinhive と Crypto-Loot を使用する．マイニングを行わないトラフィックとして，Alexa による 2018 年 12 月の全世界におけるアクセスランキング上位 50 サイトを対象とする [19]．

MLP の構成パラメータの決定には，Hyper-Paramater 探索を使用し，最も精度が高いモデルと，最も精度が低いモデルを計測し，パラメータチューニングによる効果を確認する．

検討手法を元に実際に運用される環境では，ネットワーク管理者がインターネット利用者のトラフィックを取得できる環境である必要がある．尚，ネットワーク管理者はインターネット利用者にネットワークトラフィックの分析を行う旨を説明し，同意した上で使う必要がある．

キャプチャしたパケットを Wireshark で読み込むと「TCP ACKed unseen segment」と表示されるパケットが存在する場合，パケットキャプチャデバイスの性能が不足している可能性がある．この「TCP ACKed unseen segment」は，パケットの転送は正常に行われているにも拘らず，TCP ACK パケットがキャプチャできていないことを意味する．つまり，

送受信した実際のパケットと計測したパケットが異なっているということになり，データとして問題がある．そこで，トラフィック取得環境では,「TCP ACKed unseen segment」の存在が全体の 1 割以下となることを確認した上で測定を行う．

3.1　データセット構成

本節では，データセットとして用いるパラメータについて述べる．データセットとして用いる，tcpdump によって取得できる情報は図である．

IP ヘッダと TCP ヘッダである．

表 3.1　Hyper-Paramater 探索の結果

	パラメータ	値
1	隠れ層 1,2 のノード数	40
2	ミニバッチサイズ	128
3	optimizer	adam
4	activation	relu

3.2　Hyper-Paramater 探索

本研究では，DNN モデルの精度改善のために，Hyper-Parameter 探索を行う．Hyper-Parameter とは，ニューラルネットワークを構成する情報のことであり，活性化関数や隠れ層の次元数，最適化アルゴリズムなど，様々な選択肢から最適化組み合わせが存在する．本研究の実験では，自動できに最も精度の高い Hyper-Parameter の組み合わせを検証した．検査方法には，Grid-Search を使用し，全探索による最適な組み合わせを検証した．

Grid-Serch では，全データセットの 20% をテストデータして分離し，残りデータを使用してホールドアウト検証を行った．そして，その結果をパラメータセットの精度とした．

表 3.2　Hyper-Paramater 探索で使用したパラメータ

	パラメータ	探索範囲
1	隠れ層 1,2..,7 のノード数	[10,20,30,40]
2	ミニバッチサイズ	[32,128]
3	optimizer	[”RMSprop”,”Adagrad”,”adam”,”Nadam”]
4	activation	[”relu”,”tanh”,”softsign”]

3.3　ツール

DNN モデルの作成，学習及び判別に使用したツールとして，TenserFlow をバックエンドに使用した Keras ニューラルネットワークライブラリを使用した．TenseorFlow は機械学習の分野の中でもデイープラーニングに特化した学習と推論を行うことができるフレームワークである．そして Keras は，迅速な実験を可能とすることに重点を置いて開発された，高水準のニューラルネットワークライブラリである [20]．Keras を使用することで，ニューラルネットワークを用いたモデルの構成時間を高速に行うことが可能となる．

第 4 章

実験と評価

本章では，はじめに実験で使用する使用したデータセットを作成するための機材，環境とデータセットの作成結果について述べる．次に前章で述べた提案手法である，DNN のモデルを作成するための Hyper-Paramiter 探索の実験を行った機材，環境と実験結果について述べ，検出精度について評価する．

4.1 データセット作成環境と手順

本節では，データセットの作成に用いた機材の構成とソフトウェアについて述べ，その後作成手順について述べる．

4.1.1 使用するパラメータ

取得できるデータは，tcpdump によって取得できるパラメータを用いる**??**．tcpdump によって取得できる TCP パケットの各パラメータの検討内容を下記に示す．

```
15:30:12.864420 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), leng
    192.168.1.134.64711 > 192.168.1.228.443: Flags [P.], seq 1:518, ack 1, win 2059,
```

各種フィールド

TCPIP を利用して通信を行うとき，IP ヘッダの IP プロトコルが終点ノード間の通信を担い，TCP ヘッダの TCP プロトコルがその通信の信頼性を担保している．

そこで，IPv4，TCP パケットから IP ヘッダと TCP ヘッダ情報から通信の特徴量を抽出することを検討する．抽出した特徴量を元に作成した識別モデルに汎用性を持たせるため，クライアント環境に大きく依存しない特徴量を使用する必要がある．

表 4.1　データセットフィールド

1	一番目の通信方向
2	一番目のフラグの状態
3	一番目の TTL
4	一番目のフラグメントの状態
5	一番目のウィンドウサイズ
6	一番目の TCP パケットの長さ
7	二番目の通信方向
8	二番目のフラグの状態
9	二番目の TTL
10	二番目のフラグメント
11	第二のウィンドウサイズ
12	第二の TCP パケットの長さ

4.1.2　機材の構成

データセットの作成に用いた機材の構成は下記の通りである．

- MacBook Pro 13-inch Early 2013
  - CPU: Intel® Core™ i7-3540M CPU @ 3.00GHz
  - RAM: 8GB
  - OS: macOS Sierra 10.12.6
- Buffalo WZR-1750 DHP2

図 4.1　ネットワークトラフィック取得環境

- CPU: BCM4708 ARM® Cortex™-A9 dual-core @ 800MHz
- RAM: 512MB
- OS: Linux 4.4.159 #4001 SMP Wed Oct 10 09:28:16 CEST 2018 armv71
- FIRMWARE: DD-WRT v3.0-r37305 std (10/10/18)

図 4.1 の構成のように MacBook Pro (以下，PC) と WZR-1750 DHP2 (以下，ルータ) を Mini DisplayPort - Ethernet 変換アダプタを使用して LAN ケーブルで接続し，ルータからインターネット回線側の装置に接続した．

4.1.3　ソフトウェア環境

データセットの作成に用いた機器ごとのソフトウェア環境は下記の通りである．

- MacBook Pro 13-inch Early 2013
  - Python 3.6.1
  - Wireshark 64bit Version 2.4.0
  - tcpdump version tcpdump version 4.9.2 – Apple version 79.60.2
  - libpcap version 1.8.1 – Apple version 67.60.2
  - LibreSSL 2.2.7
- Buffalo WZR-1750 DHP2
  - tcpdump version 4.9.2

– libpcap version 1.7.2

4.1.4　作成手順

下記の手順で，マイニングを行うデータセットを作成した．

0. 予め，マイニングを行うために必要な情報を登録し，Web サイトを作成しておく．
1. ルータで tcpdump コマンドを実行してルータを経由する全パケットの取得を開始する．
2. PC のブラウザからマイニングを行う Web サイトにアクセスし，マイニングを一定時間行う．
3. ルータで実行中の tcpdump コマンドを停止する．
4. ルータで取得したパケットを PC に移し，パケットデータを修正する．
5. 修正したパケットデータをマイニングトラフィックデータとして扱い，データセットの形式に変換する．

同様にして，ブラウザからマイニングを行わない Web サイトにアクセスすることで，マイニングを行わないデータセットを作成した．

マイニングを行わないデータセットには，Alexa による 2018 年 12 月の全世界におけるアクセスランキング上位 50 サイトを対象として，あらかじめ各サイトのアクセス時にマイニングトラフィックが発生していないことを確認した上でアクセスを行なった．キャッシュデータを削除した上で各サイトに約 1 分間滞在し，可能であれば適当な速さで縦スクロールを行いつつコンテンツを確認した時のトラフィックデータを取得した．

尚，ルータで取得したパケットは，マイニング以外のアプリケーションによって発生した通信を含んでいる可能性がある．そこで，Wireshark を使用して該当するパケットを削除することで，マイニング以外のパケットがマイニングトラフィックとして扱われないようにした．

4.2　実験機材と環境

本節では，実験に用いた機材と環境について述べる．

4.2.1　ハードウェア機材とソフトウェア環境

実験に用いたハードウェア機材は前章と同じ PC である．

また，実験に用いたソフトウェア環境は以下の通りである．

- Python 3.6.1
- フレームワーク: TensorFlow 1.0 / Keras 2.1.4
- ライブラリ: numpy 1.12.0 / scikit-learn 0.18.1

4.3　識別結果

本節では，識別結果を評価する指標，パラメータ探索の結果決定したパラメータ，決定したパラメータによる識別結果について述べる．

4.3.1　指標

識別結果の評価を，予測結果の評価手法である ConfusionMatrix で行う．そして，分類器の精度，誤検出率，真陽性率，誤検出率を求める．

正確度は，全体としてどの程度の頻度で正しいかを示す指標であり，$(TP+TN)$/合計 で求める．1 に近いほど良く，精度が高いことを意味する．

誤分類率は，全体としてどの程度の頻度で間違っているかを示す指標であり，$(FP+FN)$/合計 で求める．0 に近いほど良く，誤り率が低いことを意味する．

適合率は，真のデータがどのぐらいの頻度で真と分類されるかを示す指標であり，$TP/(TP+FN)$ で求める．

誤検出率は，偽のデータがどのぐらいの頻度で偽と分類されるかを示す指標であり，

図 4.2　探索し結果を反映したモデル

$TN/(FP+TN)$ で求める．

F 値は，適合率と再現率の調和平均で定義される指標であり，1 に近いほど適合率と再現率のバランスが良いことを示す．指標であり，$2TN/(2*TN+FN+TN)$ で求める．

4.3.2　パラメータ探索結果

本研究で行なった，Hyper-Paramater 探索の結果，表 4.2 のパラメータの組み合わせのとき図 4.2 の構成となり，精度が 0.79 となり最も高かった．

表 4.2　Hyper-Paramater 探索の結果

	パラメータ	値
1	隠れ層 1,2 のノード数	40
2	ミニバッチサイズ	128
3	optimizer	adam
4	activation	relu

4.3.3　識別結果

前節のパラメータ検索結果から最も良い組み合わせでトレーニングしたモデルによる，マイニングトラフィックとそれ以外のトラフィックフローの評価結果は表 4.3 に示す．

表では，マイニング通信であることを Yes とし，それ以外の通信であることを No としている．

表 4.3 パラメータ探索結果を反映したモデルによるマイニングトラフィックの識別率

n=4157	Predicted: Yes	Predicted: No
Actual: Yes	TP=160	FN=3
Actual: No	FP=2	TN=3992

この結果から，パラメータ探索結果を反映したモデルによるマイニングトラフィック識別精度は 0.95, 誤検知率は，0.05 であることが分かる．

また，パラメータ探索結果が最も低い場合のモデルによるマイニングトラフィック識別精度 0.88 となった．

4.4 考察

パラメータ探索結果を反映したモデルによるマイニングトラフィック識別精度は 0.95 であり，パラメータ探索結果が最も低い場合のモデルによるマイニングトラフィック識別精度は 0.88 であったことから，パラメータのチューニングを行うことで，7%程度の改善が見られることが分かる．

検討手法はマイニングトラフィックの識別において，高い精度で識別できていることが確認できる．

よって検討手法を用いたネットワーク環境において少なくとも Coinhive と Crypt-Loot を使用した通信に関して，下記のトラフィックが検出できる．

1. マルウェアに感染したコンピュータが行うマイニング
2. Web サイト閲覧者のコンピュータで行うマイニング
3. VPS(Virtual Private Server) や公共施設などでのでマイニング

また，マイニング通信の通信先である，マイニングプールのドメインの変更や Proxy サ

イトを経由することによってアクセス先が変更された場合であっても，検知に使用するパラメータは変化しないので高精度で識別できると考えられる．

一方で，マイニングのトラフィックにダミーのデータや通信が挿入された場合は，新しいパターンのトラフィックを利用して再度モデルを構築する必要がある．

第 5 章

結論

本研究の検討手法により，IP ネットワーク上のをマイニングトラフィックを 95.7%の精度で識別することを実現できた．

マイニングトラフィックは，他のマルウェアのトラフィックと比較して，トラフィックが発生していても直ちに問題を引き起こす可能性は低く，緊急性が高くない．また，マイニングはネットワーク通信を必要とすることを考量すると，同一の IP アドレスから連続的にトラフィックが発生している可能性が高い．そのため，95.7%の識別率であっても実用に耐えうる数値であると考える．

今後は，トラフィックからマイニングトラフィックを識別するだけでなく，そのトラフィックが不正な通信で発生したものであるかを判別するできるようにする必要がある．

参考文献

[1] 一般社団法人日本仮想通貨交換業協会, “仮想通貨取引についての現状報告,” https://www.fsa.go.jp/news/30/singi/20180410-3.pdf, Jan. 2019 accessed.

[2] CoinMarketCap, “仮想通貨の時価総額 | CoinMerketCap,” https://coinmarketcap.com/ja/, Jan. 2019 accessed.

[3] Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,”

[4] N.Arvind, B.Joseph, F.Edward, “Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction,” Princeton University Press, Jul. 2016.

[5] G.Hileman, M.Rauchs, “GLOBAL CRYPTOCURRENCY BENCHMARKING STUDY,” https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2965436 , Jan. 2019 accessed.

[6] Monero Project, “Monero - secure, private, untraceable,” https://ww.getmonero.org/ , Jan. 2019 accessed.

[7] Evgeny Lopatin, “Kaspersky Security Bulletin 2018. Story of the year: miners,” https://securelist.com/kaspersky-security-bulletin-2018-story-of-the-year-miners/89096/ , Jan. 2019 accessed.

[8] Omri Segev Moyal, Christiaan Beek, “Rise of Coinminers,” https://bsidessf2018.sched.com/event/E6ir/rise-of-coinminers , Jan. 2019 accessed.

[9] x25, “CoinHive Stratum Mining Proxy,” https://github.com/x25/coinhive-stratum-mining-proxy/, Jan. 2019 accessed.

[10] RamNode LLC, “RamNode LLC - Acceptable Use Policy,” https://clientarea.ramnode.com/aup.php , Jan. 2019 accessed.

[11] Alexandre Mundo Alguacil et al., Christiaan Beek, “McAfee Labs Threats Report

December 2018," https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-dec-2018.pdf , Jan. 2019 accessed.

[12] Ryo Yamada, "Using abnormal TTL values to detect malicious IP packets," Yamada, R., & Goto, S. (2013). Using abnormal TTL values to detect malicious IP packets. Proceedings of the Asia-Pacific Advanced Network, 2012.

[13] 小出 駿 ほか, "通信プロトコルのヘッダの特徴に基づく不正通信の検知・分類手法," Computer Security Sympsium, October, 2014.

[14] Coinhive, "Coinhive Monero Mining Club," https://coinhive.com/ , Jan. 2019 accessed.

[15] Mozilla, "WebAssembly | MDN," https://developer.mozilla.org/en-US/docs/WebAssembly/ , Jan. 2019 accessed.

[16] Crypto-Loot, "CryptoLoot - Earn More From Your Traffic," https://crypto-loot.com/ , Jan. 2019 accessed.

[17] Check Point Software Technologies Ltd, "December's Most Wanted Malware: Crypto-Miners Affect 55% of Businesses Worldwide | Check Point Software Blog," https://blog.checkpoint.com/2018/01/15/decembers-wanted-malware-crypto-miners-affect-55-businesses-worldwide/ , Jan. 2019 accessed.

[18] Mozilla, "WebAssembly support now shipping in all major browsers The Mozilla Blog" https://blog.mozilla.org/blog/2017/11/13/webassembly-in-browsers/ , Jan. 2019 accessed.

[19] Alexa Internet, "The top 500 sites on the web," https://www.alexa.com/topsites, Dec. 2018 accessed and archived. (https://web.archive.org/web/20181231235959/https://www.alexa.com/topsites)

[20] Keras-team, "Keras Documentation," https://keras.io/ , Jan. 2019 accessed.

[21] The Tcpdump Group, "Manpage of TCPDUMP," https://www.tcpdump.org/manpages/tcpdump.1.html , Jan. 2019 accessed.

[22] 高橋郁夫, 吉田一雄, “－ネットワーク管理・調査等の活動と「通信の秘密」－,” https://www.jaipa.or.jp/info/2005/iw2005.pdf , Jan. 2019 accessed.

[23] 木村孝, “帯域制御ガイドラインから見る通信の秘密の考え方,” https://www.nic.ad.jp/ja/materials/iw/2009/proceedings/f1/iw2009-f1-07.pdf , Jan. 2019 accessed.