

モバイルアドホックネットワークにおける アンサンブル学習を用いた頑健な攻撃端末の検出

高 博奇^{1,a)} 前川 卓也^{1,b)} 天方 大地^{1,c)} 原 隆浩^{1,d)}

受付日 2018年4月24日, 採録日 2018年11月7日

概要: 近年, モバイルアドホックネットワーク (MANET) に関心が高まっている. MANET は無線移動端末のみで構成できるという利点から, 次世代のネットワークとなるポテンシャルがある. 一方, 自律分散的なネットワークであるという特徴ゆえ, 攻撃端末の参加も容易であるという側面もある. そのため, MANET におけるセキュリティは, 解決すべき重要な問題である. 本論文では, MANET において, 機械学習を用いて攻撃端末を特定する新たな方法を提案する. 一般的に, 機械学習を用いた方法では, テスト環境で動作する検出器を同様の環境で訓練することを想定している. しかし, テスト環境と訓練環境が異なる場合, その検出器は効果的に機能しない. MANET はモバイル環境であるため, テスト環境とまったく同様の訓練環境を用意して訓練を行うことは難しい. 提案手法は, アンサンブル学習を用いて, テスト環境に適応した検出器を作成する. まず, 条件の異なる様々な環境でシミュレーションを行い, それぞれの環境ごとの検出器を学習する. そして, テスト環境で効果的に動作すると推定される検出器をいくつか統合してアンサンブル検出器を作成する. 実験結果から, 提案手法の有効性を検証し, 既存手法を超える性能を持つことを示す.

キーワード: モバイルアドホックネットワーク, セキュリティ, 機械学習

Robust Malicious Node Detection with Ensemble Learning in MANETs

BOQI GAO^{1,a)} TAKUYA MAEKAWA^{1,b)} DAICHI AMAGATA^{1,c)} TAKAHIRO HARA^{1,d)}

Received: April 24, 2018, Accepted: November 7, 2018

Abstract: Mobile ad hoc networks (MANETs) have been recently receiving attention, and they have a potential to be a next generation network due to its advantage: a MANET consists only of wireless mobile nodes. Meanwhile, this feature may make malicious nodes join MANETs easily, thereby providing secure support to MANETs is an important problem. This paper presents a novel approach that builds a robust malicious node detector for detecting malicious nodes in MANETs based on machine learning. Existing machine learning-based malicious node detection assumes that a detector for a test environment is trained in the same network. However, when test environments are different from training environments, the detectors do not work well. As MANET environments are diverse, it is impractical to prepare training environments that are identical to the test environments. Our proposed method employs ensemble learning to construct an environment-adaptive malicious node detector. First, we prepare weak malicious node detectors trained in diverse training environments. Second, we design a method to find weak detectors which are estimated to work well in the test environment, and use them to construct an ensemble detector. Through comprehensive experiments, we demonstrate the effectiveness of our method, and our method significantly outperforms the state-of-the-art methods.

Keywords: mobile ad hoc networks, security, machine learning

¹ 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology,
Osaka University, Suita, Osaka 565-0871, Japan
^{a)} gao.boqi@ist.osaka-u.ac.jp

^{b)} maekawa@ist.osaka-u.ac.jp
^{c)} amagata.daichi@ist.osaka-u.ac.jp
^{d)} hara@ist.osaka-u.ac.jp

1. はじめに

スマートフォンやタブレットに代表される移動端末、および無線通信技術の普及により、モバイルアドホックネットワーク (MANET: Mobile Ad-hoc Network) の実現が進んでいる。MANET は、移動端末のみにより構成される無線ネットワークであり、ある 2 台の端末が通信範囲内に存在すれば、直接通信を行う。MANET は既存のインフラを必要としないという利点から、緊急災害現場等への応用が期待されている [14]。一方、MANET は端末が簡単にネットワークに参加できるため、攻撃端末の影響を受けやすく [9], [31], [36]、ネットワークに参加した攻撃端末による通信妨害が問題となる [11]。そのため、MANET におけるセキュリティは解決すべき重要な課題であり、本論文では、MANET における攻撃端末の検出問題に取り組む。この問題では、通常端末が観測した他の端末の振舞いから、その端末が攻撃端末かどうかを判定する。

1.1 動機

既存研究では、ルールに基いて攻撃を回避する手法 [22], [24] や攻撃端末を検出する手法 [5], [17], [18], [19], [32], [34] が提案されている。しかし、これらの研究では、攻撃端末は 1 種類の攻撃しか行わないという非現実的な想定を置いている。実際には、(i) これらの手法を回避するために攻撃端末は攻撃方法を変更したり、(ii) 異なる攻撃端末は異なる攻撃を行うといったことが考えられる。既存手法はこれらの場合に対応できず、実践性に乏しい。

また、ルールベースの手法は複数種類の攻撃が行われる環境に対応できないという欠点がある。そのため、機械学習によって複数種類の攻撃に対応するアプローチに注目が集まっている [2], [6], [21]。このようなアプローチでは、攻撃端末の特徴 (たとえば、送受信メッセージの数) に基づいて分類器を作成し、通常の端末はこの分類器によって攻撃端末を検出する。このアプローチの利点は、攻撃の種類が複数存在する場合でも、訓練データさえ提供すれば自動的に分類器を作成できることである。しかし、分類器は訓練データに基づくため、訓練環境と端末数や移動速度等が異なるネットワーク環境では性能が低下することが予想される。これは MANET における重要な課題であるが、既存研究 [2], [6], [21] ではこの問題に対応していない。ここで、実環境で大量の訓練データを収集することは非現実的であるため、最も実践的なアプローチの 1 つは、シミュレーションにより作成した分類器を実環境に適応させることであると考えられる。

1.2 貢献

本論文では、複数種類の攻撃が行われる MANET 環境を想定し、分類に用いるための特徴および環境適応型の攻撃

端末検出手法を提案する [10]。提案手法では、様々な環境で学習した弱検出器 (WDs: weak detectors) を用意し、与えられた実環境で効果的に動作すると推定される WDs を求める。さらに、それらの WDs から強検出器 (SD: strong detector) を作成し、攻撃端末の検出に用いる。

ここで、実 (テスト) 環境で効果的に動作する WD の選択方法について説明する。一般的に、テスト環境に似た環境で作成された分類器は、テスト環境においても効果的に動作する。たとえば、テスト環境における端末の平均速度が似ている環境で作成された分類器は、そのテスト環境においても高精度で分類することが期待できる。しかし、端末の平均速度といったパラメータは、実践的には可観測ではない。そのため、どの WD がテスト環境で効果的に動作するか推定するために、提案手法では可観測なパラメータを用いる。また、この推定にも機械学習を利用する。

さらに、複数種類の攻撃が行われる環境にも柔軟に対応するため、本研究では攻撃の目的に応じて 4 種類の攻撃のグループを定義し、グループごとに典型的な攻撃の振舞いをとらえることができる特徴を設計する。これは、通常の端末が受信・傍受したメッセージから得られる統計であり、特徴を得るために追加のメッセージ送信を必要としない。

本論文の主な貢献は以下のとおりである。

- 本論文では、複数種類の攻撃が存在する MANET における攻撃端末検出問題に取り組む。筆者らの知る限り、様々な環境で効果的に動作する方法はこれまでに考えられていない。
- 様々な環境で学習した弱検出器から、与えられた実環境で効果的に動作する (と推測された) 強検出器を構築する手法を提案する。
- 提案手法の性能をネットワークシミュレータを用いて実験的に評価する。実験の結果、提案手法は既存手法を超える性能を示すことを確認した。

本論文の構成は以下のとおりである。2 章で、本論文の想定を紹介し、3 章で提案手法を説明した後、4 章で実験結果を示す。最後に 5 章で関連研究を紹介し、6 章で本論文をまとめる。

2. 想定

2.1 ネットワークモデル

モバイルアドホックネットワーク (MANET) は、 n 台の無線移動端末により構成されており、各端末は一意の識別子を持つ。端末は自由に移動し、通信範囲内の端末とは直接通信が可能である。すべての端末は同じ通信範囲を持ち^{*1}、通信範囲内の端末を隣接端末と呼ぶ。

また、ルーティングプロトコルとして、MANET で最も基本的なものである AODV を用いる。端末 s がデータパ

^{*1} 例外として、攻撃端末は異なるチャネルを使って攻撃を行うことも想定する (詳細は 2.2 節)。

ケットを端末 d に送りたいとする。このとき、 s は経路要求メッセージ (RReq) をブロードキャストし、このメッセージは中継端末を経由して d まで転送される。端末 d がその RReq を受信すると、経路応答メッセージ (RRep) を s に向けて送信する。これにより構築された経路を用いて、 s はデータパケットを d に送信する。また、経路の管理には、経路エラーメッセージ (Rerr) およびハローメッセージが用いられる (詳細は文献 [23] を参照)。

2.2 攻撃モデル

本節では、攻撃端末が行う攻撃について説明する。下で紹介する攻撃は、MANET における攻撃としてこれまでに研究されてきたものである。

ブラックホール攻撃 [20], [24]. 攻撃端末はすべての RReq に対して偽の PRep を送信し、自身が経路上の端末であるかのように振る舞う。そして、受信したデータパケットをすべて破棄する。

グレイホール攻撃 [34]. ブラックホール攻撃と類似した攻撃方法であるが、違いは、データパケットの破棄を確率的に行うことである。

シビル攻撃 [1]. 攻撃端末は、通常の端末であるかのように振る舞うことで、隣接端末が自身を通常の端末と考えるように仕向ける。これにより、メッセージが自身に転送されるようになる。

経路メッセージ破棄攻撃 [19]. この攻撃では、RReq, RRep, および Rerr をランダムに破棄する。

ラッシュ攻撃 [12]. この攻撃では、攻撃端末が RReq を受信した際、迅速に (待ち時間を設定することなく) RReq を送信する。これにより、中継端末はその攻撃端末を経路上の端末と考えてしまう。

ワームホール攻撃 [8]. 攻撃端末は、受信したすべてのメッセージを自身のチャネル (有線や隠れチャネル) を用いて地理的に離れた端末に転送する。それらを受信した別の攻撃端末は、通常どおりに受信したメッセージをブロードキャストする。これにより、通常の端末は、遠くに存在する端末を近くに存在していると誤認してしまう。

ジェリーフィッシュ攻撃 [17]. これは、受信したメッセージをしばらく保持することにより、経路の有効性や経路表の正確性を混乱させる攻撃である。

フラディング攻撃 [35]. これは、(RReq やハローメッセージのような) メッセージをブロードキャストし続けることで、経路探索や経路修正がいくつも行われていると勘違いさせ、端末の消費電力を上げたり、ネットワーク帯域を無駄遣いする攻撃である。

2.3 攻撃端末の検出

MANET 内の通常の端末は、本論文の提案手法を用いて、自身の隣接端末が攻撃端末であるかどうかを判定す

る。その判定は、観測によって得られた情報が精度の良い判定に十分である限り、任意のタイミングで行われるものとする。ここで、情報量が少ない場合、機械学習によるアプローチは精度が低いことが一般的であるため、頻繁に判定を行うといった状況は想定しない*2。そのため、通常の端末にかかる計算負荷は小さい。

3. 提案手法

3.1 概要

図 1 および図 2 は提案手法の概要を示す。提案手法は教師あり学習の手順に基づき、訓練フェーズとテストフェーズで構成される。訓練フェーズでは、端末の移動速度や端末密度等を変化させた様々な環境をシミュレートする。具体的には、各訓練 (シミュレーション) 環境ごとに攻撃端末の弱検出器 (WD) を 1 つ訓練する (図 1 (1))。この WD は、該当する訓練環境において端末が収集したデータを用いて訓練される。通常の端末は、隣接端末の行動情報 (パケットの転送頻度等) を観測し、隣接端末が攻撃端末であ

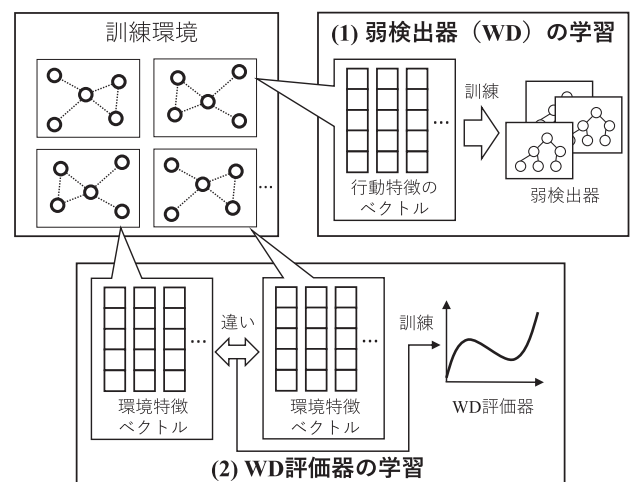


図 1 提案手法の訓練フェーズの概要。異なるネットワークパラメータの訓練環境を用意する。(1) 各訓練環境において、通常の端末が収集した行動特徴から弱検出器 (Weak Detector: WD) を訓練する。(2) 与えられた環境における WD の性能を推定する WD 評価器を訓練する (ある異なる 2 つの環境において環境特徴を計算し、その違いを使って訓練する)

Fig. 1 Overview of the training phase. We prepare training environments with different network parameters. (1) We train a weak detector in each training environment using behavioral feature vectors calculated by normal nodes in the environment. (2) We train a weak detector evaluator that estimates the performance of a weak detector trained in a training environment when the detector is run in a test environment (We compute environmental features in two different training environments and use the difference between the environmental features to train the weak detector estimator).

*2 本論文における実験では、観測を始めてから 50 秒以上経過した後隣接端末の判定を行う。

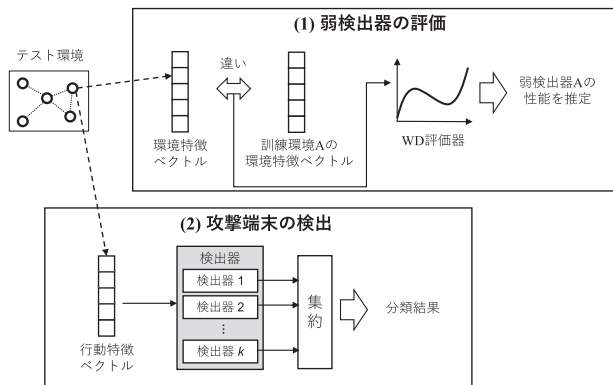


図 2 提案手法のテストフェーズの概要。(1) 通常の端末は、収集した情報から、環境特徴を計算する。その環境特徴から、現在の環境と訓練環境の違いを計算し、それを WD 評価器に入力する。(2) 各隣接端末の行動特徴を計算し、行動特徴ベクトルを作成する。WD 評価器によって、現在の環境で効果的に動作すると推定された WD を用いて隣接端末が攻撃端末であるかを判定する

Fig. 2 Overview of the test phase. (1) A normal node computes environmental features from its observable information. The node then calculates differences in the test environment and a training environment using the environmental features of the test environment and environmental features of the training environment calculated in advance. The computed differences are fed into the weak detector evaluator to estimate the performance of a weak detector trained in the training environment. (2) The normal node in a test environment computes a behavioral feature vector for each neighboring node using observed behavior of the neighboring node. The vector is fed into an ensemble of weak detectors whose performances are estimated to be high in the test environment.

るかどうかを判定する。そのため、隣接端末から観測した情報から特徴（行動特徴）を抽出し、WD に入力する。行動特徴は、隣接端末が送信したメッセージから抽出する。つまり、各端末は隣接端末が送信したメッセージを傍受することにより特徴を得るため、追加の通信量は発生しない。

ここで、テスト環境では訓練データを収集できない（テスト環境のネットワークは未知である）。そのため、実環境における WD の性能を推定する WD 評価器を訓練する（図 1(2)）。WD 評価器は、テスト環境と異なる訓練環境で訓練するものである。実環境ではグラウンドトゥールズが得られないため、WD の性能を実環境で訓練できない。そのため、実環境で観測したデータを用いて WD の性能を推定する方法が実践的である。これを実現するため、教師あり学習によるアプローチをとる。環境（ネットワーク）A と B が与えられたとする。環境 A における端末から得られたデータから、環境 A を表す特徴を計算する。環境 B に対しても同様の処理を行い、これらの特徴の違いを計算する。さらに、環境 A で訓練した WD を環境 B で実行し、

その性能を確認する。計算した特徴の違いと試行した性能を用いて、あるテスト環境における WD の性能を推定する評価器を訓練する。

環境の違いを WD の性能の推定に用いるというアイデアは、訓練した環境とまったく異なる環境でテストした際、その WD は効果的に動作しないという仮定に基づいている。たとえば、端末密度の大きい環境で訓練した WD は、端末密度が疎な環境では機能しない。なぜなら、メッセージの受信率が大きく異なるため、行動特徴に大きな違いが生じるからである。WD 評価器の入力はテスト環境と訓練環境の違いを表す特徴であり、これを環境特徴と呼ぶ。この特徴は、通常の端末が観測する情報から計算される。

テストフェーズでは、訓練環境とテスト環境の環境特徴を用いて、各 WD のテスト環境における性能を推定（図 2(1)）し、アンサンブル検出器を構築する。通常の端末は、この検出器を用いて隣接端末が攻撃端末であるか判定する（図 2(2)）。

3.2 提案手法で用いる特徴

本節では、攻撃端末の検出のために用いる行動特徴、および、WD の性能を推定するために用いる環境特徴について述べる。

3.2.1 行動特徴

既存研究では、MANET における攻撃モデルをルーティングプロトコルや層（レイヤ）に基づいて分類していた [3]。しかし、本論文では、効率的に攻撃端末を検出するために、攻撃の目的に基づいて 4 つのグループに分類し、そのグループに基づいて特徴を設計する。攻撃の目的は、以下の 4 つに分類できる：(i) リンク切断を引き起こすこと（CLF：Causing Link Failures）、(ii) パケットを破棄すること（DDP：Dropping Data Packets）、(iii) メッセージングの経路に自身（攻撃端末）がいるように仕向けること（DRA：Dragging Routes to Attackers）、および (iv) 無駄な通信量を発生させること（CTR：Causing Redundant Traffic）。

表 1 に、攻撃モデルをグループに分類した結果を示す。いくつかの攻撃モデルは、複数のグループに分類できる。たとえば、ブラックホール攻撃は RRep を送信してパケットを破棄するため、パケット破棄とメッセージングの経路に存在するという 2 つの目的がある。

通常の端末は、自身の隣接端末の情報を観測することにより攻撃端末であるかの判定を行う。このときに観測する情報を表 2 に示す（AODV はブロードキャストによりメッセージを送信するため、通信範囲内の端末は傍受が可能である）。次に、表 3 に提案手法で用いる行動特徴の一覧を示す。既存研究では「数」を特徴して用いていたが、提案手法では、ネットワーク環境の違いによる影響を小さくするために「率」を用いる。たとえば数を用いると、隣接端

表 1 攻撃の目的に基づくグループ化

Table 1 Grouping attacks by their objectives.

グループ	攻撃モデル
CLF	経路メッセージ破棄攻撃, シビル攻撃, ジェリーフィッシュ攻撃
DDP	ブラックホール攻撃, グレイホール攻撃
DRA	シビル攻撃, ラッシュ攻撃, ワームホール攻撃, ブラックホール攻撃
CRT	経路メッセージフラッディング攻撃, ハローメッセージフラッディング攻撃

表 2 各端末が観測する情報

Table 2 Information observed by nodes.

情報	定義
NReqRec	1つの隣接端末から傍受した RReq の数
NRepRec	1つの隣接端末から傍受した RRep の数
NRepSen	1つの隣接端末に送信された RRep の数
NRerRec	1つの隣接端末から傍受した Rerr の数
NRerSen	1つの隣接端末に送信された Rerr の数
NHelRec	1つの隣接端末から傍受したハローメッセージの数
NDatRec	1つの隣接端末から傍受したデータパケットの数
NDatSen	1つの隣接端末に送信されたデータパケットの数
TReqRec	傍受した RReq の総数
TReqSen	送信した RReq の総数
TRepRec	傍受した RRep の総数
TRepSen	送信した RRep の総数
TRerRec	傍受した Rerr の総数
TRerSen	送信した Rerr の総数
THelRec	傍受したハローメッセージの総数
THelSen	送信したハローメッセージの総数
TDatRec	受信したデータパケットの総数
TDatSen	送信したデータパケットの総数

表 3 行動特徴の一覧

Table 3 List of behavioral features.

行動特徴	定義
RepSenRatio	$NRepRec / TRepRec$
RepRecRatio	$NRepRec / TReqRec$
RepIgnRatio	$NRepRec / NRepSen$
ReqRecRatio	$NReqRec / TReqRec$
DatSenRatio	$NDatSen / TDatSen$
DatRecRatio	$NDatRec / TDatRec$
DatIgnRatio	$NDatRec / TDatSen$
RerRecRatio	$NRerRec / TRerRec$
HelRecRatio	$NHelRec / THelRec$
AllPckRatio	$(NDatSen + NRepSen) / (NDatRec + NRepRec)$
RepUsfRatio	$NDatRec / NRepSen$
RepReqRatio	$NRepRec / TReqSen$
RerSenRatio	$NRerSen / TRerSen$
HelCheckRatio	$NHelRec / THelSen$
ReqIgnRatio	$NReqRec / TReqSen$
RepUsfRatio	$NRepSen / TDatRec$
ReqUsfRatio	$NReqRec / TDatRec$
HelUsfRatio	$THelSen / TDatRec$

末の数に大きな影響を受けてしまう。以下では、各行動特徴に対して表 1 との関係を述べる。

RRep sent ratio (RepSenRatio), RRep ignored ratio (RepIgnRatio), Rerr sent ratio (RerSenRatio), および Rerr received ratio (RerRecRatio) は, CLF 用に設計した特徴である。リンク切断は, ネットワークトポロジの変化および経路メッセージを破棄することにより発生する。攻撃端末が経路メッセージを破棄する場合, 通常の端末は経路表を正確に更新できない。経路メッセージの送受信割合は, 隣接端末がメッセージを破棄しているかどうかを判断することに役立つ。

Data received ratio (DatRecRatio), Data ignored ratio (DatIgnRatio), RRep useless ratio (RepUsfRatio), RRep useful ratio (RepUsfRatio), および RReq useful ratio (ReqUsfRatio) は, DDP 用に設計した特徴である。傍受からこれらの特徴を計算することにより, 隣接端末が適切にメッセージを転送しているか分かる。たとえば, ある隣接端末の DatIgnRatio が他の隣接端末に比べて小さい場合, その隣接端末はメッセージを破棄していると考えられる。

Data packet sent ratio (DatSenRatio), RRep checked by RReq ratio (RepReqRatio), RRep received ratio (RepRecRatio), all routing packets ratio (AllPckRatio), および RReq ignore ratio (ReqIgnRatio) は, DRA 用に設計した特徴である。攻撃端末が自身を経路上に置く用に仕向ける場合, RRep の送信数およびデータパケットの受信数が多くなる。

RReq received ratio (ReqRecRatio), Hello check ratio (HelCheckRatio), Hello received ratio (HelRecRatio), および Hello useful ratio (HelUsfRatio) は, CRT 用に設計した特徴である。フラッディング攻撃を行う攻撃端末は, 無駄な RReq やハローメッセージを送信するため, これらの特徴はフラッディング攻撃を行う攻撃端末を検出するために有効である。

3.2.2 環境特徴

次に, ネットワーク環境を表す特徴を抽出する。この特徴を用いて, テスト環境で効果的に機能する WD を推定する。提案手法で用いる環境特徴を表 4 に示す。これらの特徴は, 一定期間で観測された送信および傍受メッセージの数から計算される。各端末は, ある時間区間ごとに, 表 4 に示されているそれぞれのメッセージの送信数および傍受したメッセージの数をカウントしている。ある時間区間における, ある端末がカウントしたある 1 種類のメッセージの数を x_i とする。このとき, ある環境特徴の値は, $\frac{\sum x_i}{w}$ であり, w は観測した時間区間の数である。

3.3 弱検出器の性能推定

各環境で WD を訓練 (作成) した後, WD 評価器を訓練する。これは, 訓練された環境と異なる環境における WD

表 4 環境特徴の一覧

Table 4 List of environmental features.

環境特徴	定義
AvgReqRec	傍受した RReq の平均の数
AvgReqSen	送信した RReq の平均の数
AvgRepRec	傍受した RRep の平均の数
AvgRepSen	送信した RRep の平均の数
AvgDatRec	傍受したデータパケットの平均の数
AvgDatSen	送信したデータパケットの平均の数
AvgRerRec	傍受した Rerr の平均の数
AvgRerSen	送信した Rerr の平均の数
AvgHelRec	傍受したハローメッセージの平均の数
AvgHelSen	送信したハローメッセージの平均の数
AvgNeiMet	平均の隣接端末数

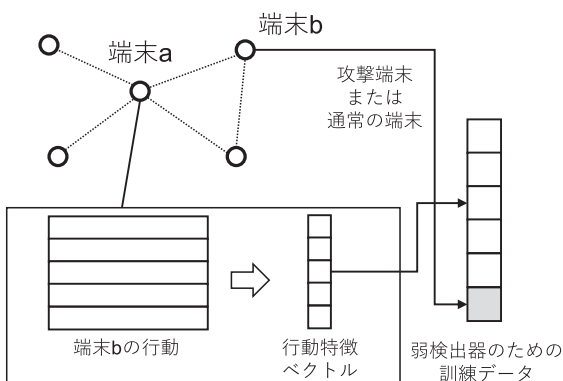


図 3 弱検出器の訓練

Fig. 3 Training weak detector.

の性能を推定するものである。

3.3.1 弱検出器の作成

まず、各シミュレーション環境においてラベル付きデータを用意し、その環境における WD を作成する。端末 a が端末 b を観測しているとする (図 3)。端末 a は、端末 b の行動特徴を計算し、それに基づいて行動特徴ベクトルを作成する。また、端末 b が攻撃端末であるかどうかを行動特徴ベクトルにラベル付ける (シミュレーションであるため、これは既知である)。隣接端末間のペアごとにそのベクトルを作成し、このシミュレーション環境における弱検出器を作成する。検出器には、トレーニング時間が短く、オーバーフィッティングを回避しやすい random forest^{*3}を用いる (つまり、あるシミュレーション環境に対して 1 つの random forest が作成される)。また、通常の端末を示すベクトルの数と攻撃端末を示すベクトルの数の比率が均等でない場合、多数派の方に対して、比率が同じとなるようにアンダーサンプリングを行う。

^{*3} 3.1 節で述べたとおり、提案手法は、単一の環境でしか動作しないと考えられる検出器を弱検出器とし、それらを組み合わせて様々な環境で動作する強検出器を構成している。そのため、本論文における提案手法では、random forest を弱検出器と呼ぶ。また、パラメータは Weka [21] におけるデフォルトを用いた。

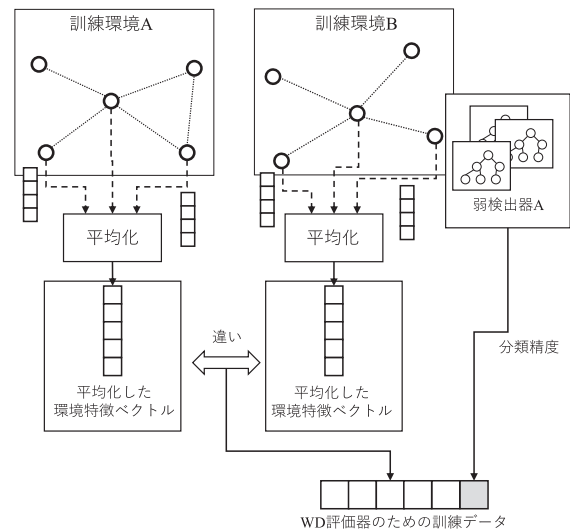


図 4 WD 評価器の訓練

Fig. 4 Training WD evaluator.

3.3.2 WD 評価器

次に、訓練環境で作成した WD と訓練環境で得られた環境特徴を用いて、WD 評価器を作成する。訓練環境 A で作成した WD の環境 B における性能について考える (図 4)。その性能を推定するため、環境 A および B で得られた環境特徴を利用する。まず、1 つの環境で通常の端末が観測したデータから、その端末における環境ベクトルを作成し、すべての通常の端末のものを平均化した平均環境特徴ベクトルを作成する。環境 A と環境 B の平均環境特徴ベクトルの違いを説明変数として利用し、環境 A で作成した WD の環境 B における性能を推定する。これにより、環境の違いによる性能を学習する。

より具体的な説明を行う。環境 A および B における平均環境特徴ベクトルをそれぞれ x_A および x_B とする。これらの違いは、以下のように計算される。

$$d(x_A, x_B) = \begin{pmatrix} |x_{A,1} - x_{B,1}| \\ |x_{A,2} - x_{B,2}| \\ \vdots \\ |x_{A,e} - x_{B,e}| \end{pmatrix} \quad (1)$$

ここで、 $x_{A,i}$ は、 x_A の i 番目の要素であり、 e は、環境特徴の数である。この違いを説明変数とする。性能の推定として、環境 A で作成した WD の環境 B における攻撃端末の分類精度 (平均 F 値) を計算し、これを独立変数とする。得られた変数 (説明変数と平均 F 値) を訓練データとし、すべての環境のペアに対して同様の処理を行う。そして、計算した訓練データから WD 評価器を作成する。環境 A で作成した WD の性能 w_A は、以下のように推定される。

$$w_A = f(d(x_A, x_B)) \quad (2)$$

ここで、 $f(\cdot)$ は、 w_A を計算する回帰分析である。提案手法では、 $f(\cdot)$ として事前実験により高い回帰精度を確認し

た SVM 回帰 [30] を用いる。また、カーネルは多項式カーネルを用いる。

3.4 アンサンブル攻撃端末検出器

ここから、通常の端末 $\mathbf{a}$ が、あるテスト環境において用いるアンサンブル検出器の構築方法について述べる。まず、 $\mathbf{a}$ が観測した環境特徴を用いて、各訓練環境の WD の性能を推定し、性能が高いと推定された上位 k 個を選択する (MANET における端末はメモリ量が限られるため、すべての WD ではなく k 個のみを用いる)。そして、選択した k 個の WD からアンサンブル検出器を構築する。この検出器は、それぞれの WD の検出結果を、推定性能を用いて以下のように集約する。

$$\text{Pr}_{\text{agg}}(y = \text{mal} | f_b) = \frac{1}{W} \sum_{i=1}^k w_i \text{Pr}_i(y = \text{mal} | f_b)$$

ここで、 $\text{Pr}_i(y = \text{mal} | f_b)$ は i 番目の WD における端末 $\mathbf{b}$ の判定結果 (攻撃端末である確率) であり、 f_b は端末 $\mathbf{b}$ に対する行動特徴ベクトルである。また、 w_i は、WD 評価器が推定した i 番目の WD の性能であり、 $W = \sum_{i=1}^k w_i$ である。この確率に基づいて、端末 $\mathbf{b}$ が攻撃端末であるかどうかを以下のように判定する。

$$\hat{y} = \begin{cases} \text{malicious} & (\text{Pr}_{\text{agg}}(y = \text{mal} | f_b) > 0.5) \\ \text{normal} & (\text{otherwise}) \end{cases}$$

要約. あるテスト環境で端末 $\mathbf{a}$ が攻撃端末を検出する場合、検出を行う前にアンサンブル検出器が必要である。端末 $\mathbf{a}$ は、ある一定期間で環境特徴を収集し、環境特徴ベクトルを計算した後にアンサンブル検出器を構築すると想定する。つまり、ここで計算した環境特徴ベクトルを用いて、上で説明した方法からアンサンブル検出器を構築する。その後、自身の隣接端末の行動を一定期間観測して行動ベクトルを計算し、その隣接端末が攻撃端末であるか判定する。

4. 評価実験

本章では、提案手法の性能評価のために行った実験の結果を紹介する。

4.1 セッティング

本実験では、ネットワークシミュレータ Qualnet 7.4 *4 を用いた。各端末は IEEE 802.11b を用いてメッセージおよびデータパケット (256 bytes) の送信を行う。通信範囲はおよそ 100 [m] であり、通信帯域は 11 Mbps である。既存研究 [1], [9], [14], [29] と同様に、端末の移動モデルにはランダムウェイポイントモデル [4] を用いた。端末の最大速度を v_{max} とし (端末の移動速度は $[0, v_{\text{max}}]$ から一様

表 5 パラメータ設定

Table 5 Parameter setting.

パラメータ	値
n	50, 60, 70, 80, 90, 100
m	0.1, 0.2, 0.3, 0.4
v_{max} [m/sec]	1.0, 2.0, 3.0, 4.0
ネットワークサイズ [m ²]	500 × 500, 600 × 600, 700 × 700, 800 × 800, 900 × 900, 1000 × 1000
f [sec]	1.0, 2.0, 3.0, 4.0
t [sec]	50, 100, 200, 300, 400, 600, 800, 1,000

ばれる), 停止時間は 0 [sec] とした。端末の数が n のとき、攻撃端末の数は $n \cdot m$ ($m \in [0.1, 0.4]$) である。AODV のソース端末と目的端末のペアは、 f [sec] ごとにランダムに選択した。もしソース端末が目的端末に対する経路をすでに把握している場合は経路探索を行わない。ネットワーク環境およびシミュレーション時間 t は、表 5 に示すとおりである (デフォルトとして、 $t = 300$ とした)。

ここで、攻撃モデルはプロアクティブ型とリアクティブ型に分類できる。プロアクティブ型には、RReq フラッディング攻撃およびハローメッセージフラッディング攻撃が含まれる。攻撃端末がメッセージを受信した際、プロアクティブ型からランダムに攻撃方法を選び、30 秒の間に 0.5 秒ごとに攻撃 (フラッディング) を行い、その後 30 秒間攻撃を停止する。リアクティブ型には、ブラックホール攻撃、グレイホール攻撃、シビル攻撃、経路メッセージ破棄攻撃、ラッシュ攻撃、ワームホール攻撃、ジェリーフィッシュ攻撃が含まれる。プロアクティブ攻撃と同様に、攻撃端末はリアクティブ型からランダムに選んだ攻撃を行う。

評価手法. 本実験では、表 5 に示すとおり、 n の値を 6 つ、 m の値を 4 つ、 v_{max} の値を 4 つ、ネットワークサイズの値を 6 つ、および f の値を 4 つ使い、2,304 ($= 6 \times 4 \times 4 \times 6 \times 4$) 種類の環境をシミュレートした。そのうちの 90% を訓練環境とし、残りをテスト環境とした。以下は、本実験で評価した手法である。

- **MLP** [21]: 多層パーセプトロンを利用した技術であり、TReqRec, TReqSen, TRepRec, TRepSen, TRerRec, TRerSen, TDatRec, TDatSen *5, 隣接端末数 (NeiNum), 経路表のエントリの更新割合 (PCR), および経路表におけるホップ数の更新割合 (PCH) といった特徴を用いている。オリジナルの方法と同様に、交差検証 [16] を用いて MLP のパラメータをチューニングした。

- **DC**: この方法は、discriminative classifier (たとえば、SVM や Naive bayes) に基づいて、すべての訓練環境で得られたすべての訓練データから作成された単一の

*4 <http://web.scalable-networks.com/qualnet-network-simulator-software>

*5 MLP は割合を用いない点で提案手法と異なる。

検出器を用いる。この方法においても、本論文で設計した行動特徴を用いた。

- *Proposed*: 本論文の提案手法。パラメータ k は 10 とした。
- *Proposed w/o EV**6: この手法では、アンサンブル学習の際、ランダムに選んだ k 個の WD を用いた。そのため、この手法では環境特徴を利用していない。

シミュレーションで得られたすべてのデータは、行動特徴および環境特徴を計算するために用いた (MLP, DC, および Proposed w/o EV は環境特徴を用いていない)。本実験では、各端末はシミュレーション終了後に攻撃端末を判定するものとし、通常の端末のみが判定を行うものとした。また、行動特徴を観測できたすべての端末に対して判定を行った。

評価値。 先述したように、提案手法は追加の通信が発生せず、計算時間も非常に短い*7そのため、上の手法を以下の評価値で比較した。

- 精度: これは、 $\frac{|T_{\text{nor} \rightarrow \text{nor}, \text{mal} \rightarrow \text{mal}}|}{|T|}$ で計算され、 $T_{\text{nor} \rightarrow \text{nor}, \text{mal} \rightarrow \text{mal}}$ および T は、それぞれ正確に分類されたインスタンスの集合およびすべてのインスタンスの集合である。
- 検出率: これは、 $\frac{|T_{\text{mal} \rightarrow \text{mal}}|}{|T_{\text{mal}}|}$ で計算され、 $T_{\text{mal} \rightarrow \text{mal}}$ および T_{mal} は、それぞれ正確に分類されたインスタンスの集合および攻撃端末を表すすべてのインスタンスの集合である。
- 誤検出率: これは、 $\frac{|T_{\text{nor} \rightarrow \text{mal}}|}{|T_{\text{nor}}|}$ で計算され、 $T_{\text{nor} \rightarrow \text{mal}}$ および T_{nor} は、それぞれ誤って分類されたインスタンスの集合および通常の端末を表すすべてのインスタンスの集合である。

4.2 実験結果

DC における分類器の選択。 表 6 は、DC に含まれる方法の性能を示している。本実験では、random forest, Naive bayes, decision tree, および SVM を用いた*8。実験結果から、random forest が最も性能が良いことが分かる。そのため、以降では DC を random forest とする。

表 6 DC の性能

Table 6 Performance of DC.

DC	精度	検出率	誤検出率
Random forest	0.768	0.745	0.209
Naive bayes	0.614	0.738	0.449
Decision tree	0.756	0.742	0.214
SVM	0.652	0.723	0.405

表 7 k を変えた場合の Proposed の性能

Table 7 Performance of Proposed vs. k .

k	精度	検出率	誤検出率
3	0.856	0.797	0.123
5	0.853	0.816	0.134
10	0.897	0.828	0.077
30	0.843	0.805	0.143
50	0.849	0.813	0.137
100	0.842	0.800	0.142
200	0.852	0.776	0.157
300	0.857	0.770	0.167

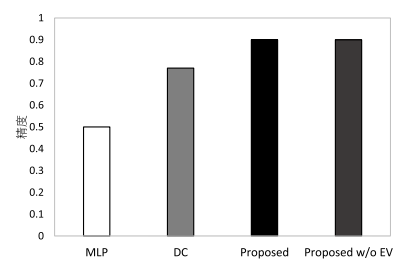


図 5 精度

Fig. 5 Accuracy.

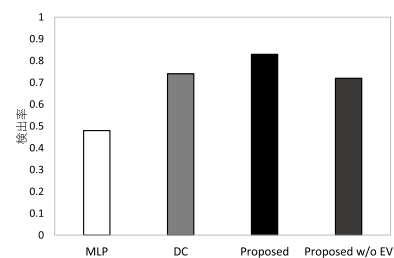


図 6 検出率

Fig. 6 Detection rate.

*6 Proposed without weak detector Evaluator

*7 2.3 GHz Core i3 で構成されたミニ PC で計算した結果、アンサンブル検出器の構築およびテストに要した平均計算時間はそれぞれ 2.47 [sec] および 46.66 [msec] であった。また、C# で実装した。

*8 本実験では、Weka によりこれらの分類器を使用し、パラメータはグリッドサーチによる最適値を用いた。Random forest では、maxDepth を 5, 10, 15, およびデフォルト、numIterations を 50, 200, 300, およびデフォルトと変えたときの性能で、検出率に重みをおいた際の最適値を選んだ (maxDepth を 10, numIterations を 200 とした)。同様に、Decision tree では、confidenceFactor を 0.1, 0.5, およびデフォルト、maxDepth を 4, 6, 8, およびデフォルト、numIterations を 2, 5, およびデフォルトと変え、confidenceFactor を 0.1, maxDepth をデフォルト、numIterations をデフォルトとした。また、SVM のカーネルはデフォルトの多項式カーネルを用いた。

k のチューニング。Proposed のパラメータである k を実験により決定する。表 7 は、 k を変えたときの Proposed の性能の結果である。この結果より、 k が大きくなると検出率が低下し、誤検出率が増加することが分かる。これは、アンサンブル検出器を構築する際に、 k が大きいとテスト環境において性能が高くない WD も考慮されてしまうからである。本実験の結果から、Proposed は $k = 10$ とした。また、Proposed w/o EV も同様に $k = 10$ とした。

MLP との比較。 図 5, 図 6, 図 7 から、既存手法である MLP は他の手法よりも性能が悪いことが分かる。特に、MLP は提案手法に対して 40% 程度性能が悪い。これは、

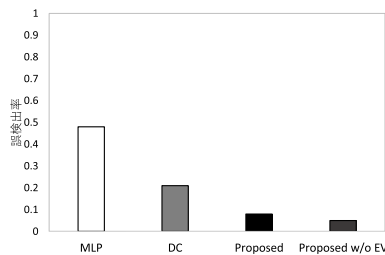


図 7 誤検出率

Fig. 7 Mis-detection rate.

表 8 行動特徴の情報ゲイン

Table 8 Information gain of behavioral features.

Behavioral features	Info. gain
RepSenRatio	0.021
RepRecRatio	0.021
RepIgnRatio	0.141
ReqRecRatio	0.092
DatSenRatio	0.003
DatRecRatio	0.023
DatIgnRatio	0.024
RerRecRatio	0.004
HelRecRatio	0.028
AllPckRatio	0.015
RepUsrRatio	0.023
RepReqRatio	0.018
RerSenRatio	0.021
HelCheckRatio	0.190
ReqIgnRatio	0.179
RepUsrRatio	0.014
ReqUsrRatio	0.103
HelUsrRatio	0.015

MLP で用いられている特徴は、訓練環境とテスト環境が異なることを考慮していないことが理由である。すべての訓練データを用いて、行動特徴および MLP で用いられている特徴の情報ゲイン [15] を計算した^{*9}。表 8 および表 9 にその結果を示す。この結果から、MLP で用いられている特徴の情報ゲインは非常に小さいことが分かる。一方、行動特徴の情報ゲインは高い。表 8 において、上位 5 個の情報ゲインを太字で表しているが、該当する特徴は表 1 で示しているすべての攻撃グループを検出することには貢献している。これらの結果より、本論文で設計した特徴は MANET における攻撃の検出に効果的であるといえる。

DC との比較。 図 5–図 7 から、提案手法は random forest を上回る性能であることが分かる。Random forest はすべての訓練環境で得られたデータを訓練データとして扱っているが、提案手法はテスト環境に適応する検出器を作成している。図 8 は、各 WD を各テスト環境に用いた際の平均の F 値のヒストグラムである。この図から分かるよう

^{*9} 情報ゲインは、有用な特徴を発見するために用いられる指標であり、情報ゲインの大きい特徴ほど、分類に有用な特徴といえる。

表 9 MLP で用いられている特徴の情報ゲイン

Table 9 Information gain of features of MLP.

行動特徴	情報ゲイン
TReqRec	6.4×10^{-5}
TReqSen	5.4×10^{-5}
TRepRec	3.7×10^{-5}
TRepSen	3.1×10^{-5}
TRerRec	3.3×10^{-5}
TRerSen	3.2×10^{-5}
TDatRec	5.4×10^{-5}
TDatSen	5.4×10^{-5}
NeiNum	3.0×10^{-4}
PCR	3.3×10^{-4}
PCH	0

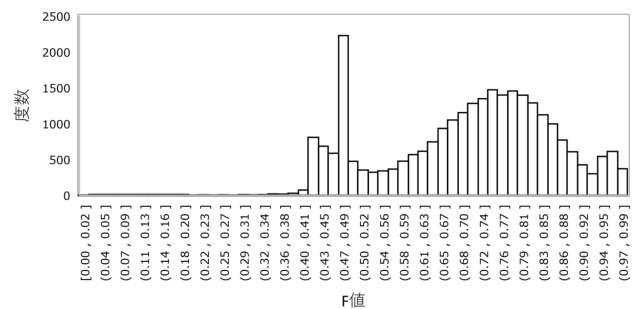


図 8 WD 評価器作成のために用いた WD の F 値の平均のヒストグラム

Fig. 8 Histogram of average F-measure of WD to train WD evaluator.

表 10 環境特徴の重要度

Table 10 Importance of environmental features.

環境特徴	重要度
AvgReqRec	0.002162
AvgReqSen	0.000172
AvgRepRec	-0.000208
AvgRepSen	-0.002034
AvgDatRec	-0.002018
AvgDatSen	-0.000882
AvgRerRec	0.000485
AvgRerSen	-0.000194
AvgHelRec	0.002325
AvgHelSen	0.000143
AvgNeiMet	-0.000394

に、多くの WD はテスト環境に適していない。提案手法では、テスト環境に適応すると推定された WD を利用する。WD 評価器は、与えられたテスト環境において、訓練環境で作成した WD の性能をそれぞれ推定しており、この推定の際に環境特徴を利用している（式 (1) および (2)）。表 10 に、ReliefF アルゴリズム [25] を用いて算出した環境特徴の重要度を示す。この結果から、AvgReqRec および AvgHelRec の重要度が高いことが分かる。これは、WD 評価器がフラッディングアタックによって発生する RReq

表 11 推定の平均絶対誤差
Table 11 MAE w.r.t. estimation.

環境特徴	平均絶対誤差
All	0.106
AvgReqRec	0.127
AvgReqSen	0.120
AvgRepRec	0.123
AvgRepSen	0.124
AvgDatRec	0.124
AvgDatSen	0.119
AvgHelRec	0.124
AvgHelSen	0.122
AvgRerRec	0.124
AvgRerSen	0.123
AvgNeiMet	0.124

とハローメッセージの数のネットワークごとのばらつきを主に学習していることを意味しており、環境の違いによる分類精度を上手く学習している理由となっている（フラディングアタックが行われない場合の解析については今後の課題とする）。これにより WD 評価器が機能し、提案手法によって選択された WD の平均の F 値は 0.89 であった。そのため、random forest よりも分類精度が高くなっている。

また、random forest に対して、行動特徴に加えて環境特徴を加えて訓練した場合の実験も行った。このとき、精度、検出率、および誤検出率はそれぞれ、0.758, 0.740, および 0.222 であり、環境特徴を加える前の性能と変わっていない。この結果から、random forest は環境特徴を活用できないこと、および提案手法の環境特徴を利用するアプローチが効果的であることが分かる。

Proposed w/o EV との比較。 図 6 から、Proposed w/o EV は DC よりも検出率がわずかに悪く、Proposed よりも 10% 程度悪いことが分かる。これは、Proposed w/o EV は、ランダムに WD を選択するため、テスト環境とまったく異なる環境で訓練された WD を用いてしまうことが多いためである。表 11 は、すべての環境特徴を使用した場合 (All) および単一の環境特徴を使用した場合の推定の平均絶対誤差を示している。この表から、All が平均絶対誤差を最も小さく抑えられていることが分かる一方、単一のものを用いた場合との差が小さいことも分かる。各特徴には相関があるため、それぞれの場合の平均絶対誤差は類似する。そのため、単一のもののみを用いた場合においても高い精度が得られる。

図 9 および図 10 はそれぞれ、各テスト環境における Proposed および Proposed w/o EV の検出率の分布を示している。これらの図から、Proposed w/o EV は、Proposed に比べて検出率が小さい場合が多いことが分かり、WD をランダムに選択することによる性能の低下が把握できる。

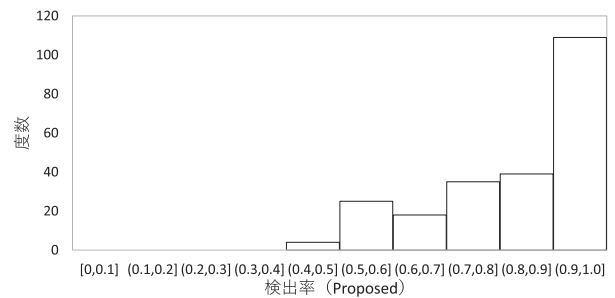


図 9 Proposed における検出率の分布

Fig. 9 Histogram of the distribution of detection rate of Proposed.

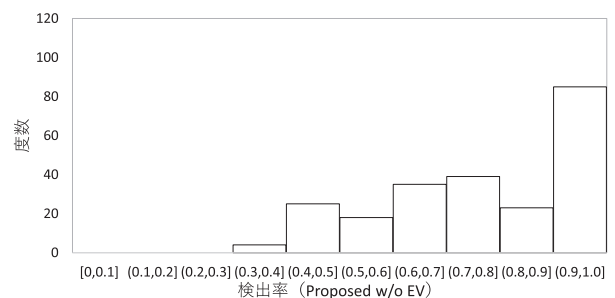


図 10 Proposed w/o EV における検出率の分布

Fig. 10 Histogram of the distribution of detection rate of Proposed w/o EV.

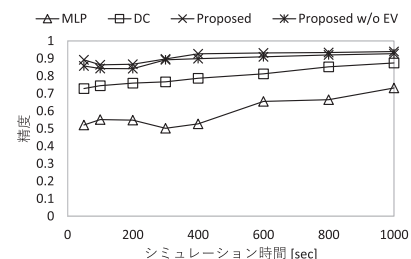


図 11 精度 vs. シミュレーション時間

Fig. 11 Accuracy with varying simulation time.

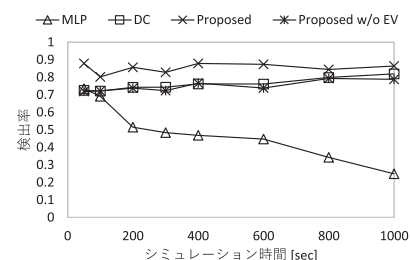


図 12 検出率 vs. シミュレーション時間

Fig. 12 Detection rate with varying simulation time.

また、Proposed はテスト環境において効果的に機能する WD を選択しているため、高い検出率である場合が多い。シミュレーション時間の影響。最後に、シミュレーション時間の影響を調べた。その結果を図 11, 図 12, 図 13 に示す。Proposed は、シミュレーション時間を変えた場合においても、最も高い精度、検出率、および最低の誤検出率を示している。この結果は、シミュレーション時間が短

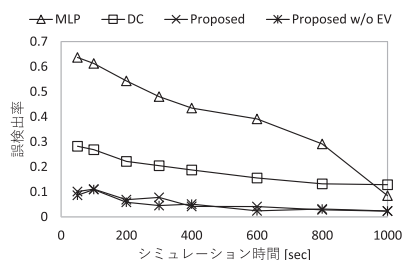


図 13 誤検出率 vs. シミュレーション時間

Fig. 13 Mis-detection rate with varying simulation time.

い（大量の情報量が得られない）場合においても、通常の端末と攻撃端末の違いを学習できること、および設計した特徴が効果的であることを示している。また、MLP はシミュレーション時間が長い場合に攻撃端末を検出できない。MLP は特徴量として、主に数を用いている。そのため、時間が経過すると通常の端末と攻撃端末の特徴に差がなくなり、性能が低下する。

5. 関連研究

本章では、MANET におけるセキュリティ問題の既存研究について紹介する。

MANET における有名な攻撃は、ブラックホール攻撃やグレイホール攻撃等のパケット破棄攻撃である。この攻撃の影響を最小化するために、複数経路を構築する方法 [33]、攻撃の影響を分析する方法 [31]、および攻撃を防止する方法 [26] が提案されている。シビル攻撃に対しては、電波強度 (RSS) を用いてその攻撃を検知する研究 [1] が行われており、文献 [7] では、ネットワークポロジ情報からワームホール攻撃を検知する手法が提案されている。フラッディング攻撃に対する研究も行われており、文献 [28] では、DSR (dynamic source routing protocol) に基づいて、他の端末の信頼度を推定する方法を提案している。文献 [13] では、フラッディング攻撃に対して、消費電力を削減する手法を提案している。これらの研究では、単一の攻撃に対する処理についてのみ考えており、複数種類の攻撃が行われる環境への適用は自明でない。

近年では、複雑なパラメータ設定（攻撃端末と判定するための閾値等）が必要ないという利点から、機械学習を用いたアプローチに注目が集まっている [2], [21], [27]。本論文では、機械学習は複数種類の攻撃モデルが存在する環境に適用できる点に着目した。これは既存研究 [2], [6], [27] では考慮されていなかった。4 章で紹介したように、文献 [21] は最新の研究であるが、訓練環境と実環境が異なる場合を想定していない。そのため、本論文における提案手法に対して、非常に性能が悪い。

6. まとめ

本論文では、モバイルアドホックネットワークにおける

攻撃端末検出問題に取り組んだ。本論文で提案した方法は、機械学習により、与えられた環境で効果的に機能する検出器を作成する。そのため、多様な環境に適応できるという利点がある。実験結果から、提案手法は既存手法を大きく上回る性能であることを確認した。今後の課題として、実環境に適用した際の提案手法の性能を評価することがあげられる。

謝辞 本研究の一部は、文部科学省科学研究費補助金・基盤研究 (A) (JP26240013) の研究助成によるものである。ここに記して謝意を表す。

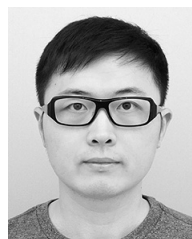
参考文献

- [1] Abbas, S., Merabti, M., Llewellyn-Jones, D. and Kifayat, K.: Lightweight sybil attack detection in manets, *IEEE Systems Journal*, Vol.7, No.2, pp.236–248 (2013).
- [2] Akbani, R., Korkmaz, T. and Raju, G.: EMLTrust: An enhanced machine learning based reputation system for MANETs, *Ad Hoc Networks*, Vol.10, No.3, pp.435–457 (2012).
- [3] Arora, J., Singh, P. and Rani, S.: Attacks in manets—a survey, *EXCEL International Journal of Multidisciplinary Management Studies*, Vol.3, No.10, pp.151–157 (2013).
- [4] Bettstetter, C., Hartenstein, H. and Pérez-Costa, X.: Stochastic properties of the random waypoint mobility model, *Wireless Networks*, Vol.10, No.5, pp.555–567 (2004).
- [5] Buchegger, S. and Le Boudec, J.-Y.: Performance analysis of the CONFIDANT protocol, *MobiHoc*, pp.226–236 (2002).
- [6] Deng, H., Zeng, Q.-A. and Agrawal, D.P.: SVM-based intrusion detection system for wireless ad hoc networks, *VTC*, Vol.3, pp.2147–2151 (2003).
- [7] Dong, D., Li, M., Liu, Y., Li, X.-Y. and Liao, X.: Topological detection on wormholes in wireless ad hoc and sensor networks, *TON*, Vol.19, No.6, pp.1787–1796 (2011).
- [8] Eriksson, J., Krishnamurthy, S.V. and Faloutsos, M.: Truelink: A practical countermeasure to the wormhole attack in wireless networks, *ICNP*, pp.75–84 (2006).
- [9] Galuba, W., Papadimitratos, P., Poturalski, M., Aberer, K., Despotovic, Z. and Kellerer, W.: Castor: Scalable secure routing for ad hoc networks, *INFOCOM*, pp.1–9 (2010).
- [10] Gao, B., Maekawa, T., Amagata, D. and Hara, T.: Environment-adaptive Malicious Node Detection in MANETs with Ensemble Learning, *ICDCS* (2018).
- [11] Hernandez-Orallo, E., Olmos, M.D.S., Cano, J.-C., Calafate, C.T. and Manzoni, P.: CoCoWa: A collaborative contact-based watchdog for detecting selfish nodes, *TMC*, Vol.14, No.6, pp.1162–1175 (2015).
- [12] Hu, Y.-C., Perrig, A. and Johnson, D.B.: Rushing attacks and defense in wireless ad hoc network routing protocols, *ACM Workshop on Wireless Security*, pp.30–40 (2003).
- [13] Jiang, F.-C., Lin, C.-H. and Wu, H.-W.: Lifetime elongation of ad hoc networks under flooding attack using power-saving technique, *Ad Hoc Networks*, Vol.21, pp.84–96 (2014).
- [14] Karaoglu, B. and Heinzelman, W.: Cooperative load balancing and dynamic channel allocation for cluster-based

- mobile ad hoc networks, *TMC*, Vol.14, No.5, pp.951–963 (2015).
- [15] Kent, J.T.: Information gain and a general measure of correlation, *Biometrika*, Vol.70, No.1, pp.163–173 (1983).
- [16] Kohavi, R. et al.: A study of cross-validation and bootstrap for accuracy estimation and model selection, *IJ-CAI*, Vol.14, No.2, pp.1137–1145 (1995).
- [17] Laxmi, V., Lal, C., Gaur, M.S. and Mehta, D.: Jelly-Fish attack: Analysis, detection and countermeasure in TCP-based MANET, *Journal of Information Security and Applications*, Vol.22, pp.99–112 (2015).
- [18] Li, W., Joshi, A. and Finin, T.: Coping with node misbehaviors in ad hoc networks: A multi-dimensional trust management approach, *MDM*, pp.85–94 (2010).
- [19] Li, X., Lu, R., Liang, X. and Shen, X.: Side channel monitoring: Packet drop attack detection in wireless ad hoc networks, *ICC*, pp.1–5 (2011).
- [20] Marti, S., Giuli, T.J., Lai, K. and Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks, *Mobi-Com*, pp.255–265 (2000).
- [21] Mitrokotsa, A. and Dimitrakakis, C.: Intrusion detection in MANET using classification algorithms: The effects of cost and model selection, *Ad Hoc Networks*, Vol.11, No.1, pp.226–237 (2013).
- [22] Patel, H.P. and Chaudhari, M.: A time space cryptography hashing solution for prevention Jellyfish reordering attack in wireless adhoc networks, *ICCCNT*, pp.1–6 (2013).
- [23] Perkins, C.E. and Royer, E.M.: Ad-hoc On-Demand Distance Vector Routing, *WMCSA*, pp.90–100 (1999).
- [24] Ramaswamy, S., Fu, H., Sreekantaradhya, M., Dixon, J. and Nygard, K.E.: Prevention of cooperative black hole attack in wireless ad hoc networks, *International Conference on Wireless Networks*, pp.570–575 (2003).
- [25] Robnik-Sikonja, M. and Kononenko, I.: An adaptation of Relief for attribute estimation in regression, *ICML*, pp.296–304 (1997).
- [26] Schweitzer, N., Stulman, A., Shabtai, A. and Margalit, R.D.: Contradiction Based Gray-Hole Attack Minimization for Ad-Hoc Networks, *TMC*, Vol.16, No.8, pp.2174–2183 (2017).
- [27] Shams, E.A. and Rizaner, A.: A novel support vector machine based intrusion detection system for mobile ad hoc networks, *Wireless Networks*, pp.1–9 (2017).
- [28] Shandilya, S.K. and Sahu, S.: A trust based security scheme for RREQ flooding attack in MANET, *International Journal of Computer Applications*, Vol.5, No.12, pp.4–8 (2010).
- [29] Shen, H. and Li, Z.: A hierarchical account-aided reputation management system for MANETs, *TON*, Vol.23, No.1, pp.70–84 (2015).
- [30] Shevade, S., Keerthi, S., Bhattacharyya, C. and Murthy, K.: Improvements to the SMO algorithm for SVM regression, *IEEE Trans. Neural Networks*, Vol.11, No.5, pp.1188–1193 (2002).
- [31] Shu, T. and Krunz, M.: Privacy-preserving and truthful detection of packet dropping attacks in wireless ad hoc networks, *TMC*, Vol.14, No.4, pp.813–828 (2015).
- [32] Subhadrabandhu, D., Sarkar, S. and Anjum, F.: A Statistical Framework for Intrusion Detection in Ad Hoc Networks, *INFOCOM* (2006).
- [33] Tseng, F.-H., Chou, L.-D. and Chao, H.-C.: A survey of black hole attacks in wireless mobile ad hoc networks, *Human-centric Computing and Information Sciences*,

Vol.1, No.1, p.4 (2011).

- [34] Vishnu, K. and Paul, A.J.: Detection and removal of cooperative black/gray hole attack in mobile ad hoc networks, *International Journal of Computer Applications*, Vol.1, No.22, pp.38–42 (2010).
- [35] Wu, B., Chen, J., Wu, J. and Cardei, M.: A survey of attacks and countermeasures in mobile ad hoc networks, *Wireless Network Security*, pp.103–135, Springer (2007).
- [36] Xing, K. and Cheng, X.: From time domain to space domain: Detecting replica attacks in mobile ad hoc networks, *INFOCOM*, pp.1–9 (2010).



高 博奇 (学生会員)

2014 年上海交通大学電子情報と電気工学学院学部卒業。2016 年大阪大学大学院情報科学研究科博士前期課程終了後、同大学院情報科学研究科博士後期課程在学中。深層強化学習、無線ネットワークにおける機械学習に関する研究に従事。



前川 卓也 (正会員)

2006 年大阪大学大学院情報科学研究科博士後期課程修了。2006 年 NTT コミュニケーション科学基礎研究所入所。2012 年より大阪大学大学院情報科学研究科マルチメディア工学専攻准教授となり、現在に至る。ユビキタスコンピューティング、ウェアラブルセンシングに関する研究に従事。2010 年度情報処理学会山下記念研究賞，2013 年度日本データベース学会上林奨励賞，2015 年電気学会優秀論文発表 A 賞，第 1 回大阪大学賞（若手教員部門）等受賞。



天方 大地 (正会員)

2012 年大阪大学工学部電子情報工学科卒業。2014 年同大学大学院情報科学研究科博士前期課程修了。2015 年同大学院情報科学研究科博士後期課程修了後、同年同大学院情報科学研究科マルチメディア工学専攻助教となり、現在に至る。情報科学博士。データベース，ネットワーク環境におけるデータ検索技術に関する研究に従事。IEEE，ACM，日本データベース学会の各会員。



原 隆浩 (正会員)

1995 年大阪大学工学部情報システム工学科卒業。1997 年同大学大学院工学研究科博士前期課程修了。同年同大学院工学研究科博士後期課程中退後、同大学院工学研究科助手、2002 年同大学院情報科学研究科助手、2004 年

同大学院情報科学研究科准教授。2015 年より同大学院情報科学研究科教授となり、現在に至る。工学博士。1996 年本学会山下記念研究賞受賞。2000 年電気通信普及財団テレコムシステム技術賞受賞。2003 年本学会研究開発奨励賞受賞。2008 年、2009 年本学会論文賞、2015 年日本学術振興会賞受賞。モバイルコンピューティング、ネットワーク環境におけるデータ管理技術に関する研究に従事。IEEE, ACM, 電子情報通信学会, 日本データベース学会の各会員。本会シニア会員。