Kitsune特徴量を用いた悪性通信のパケット分類

宮本 耕平[1,a)]　後藤 大輝[1]　石橋 亮典[1]　韓 燦洙[2]　班 涛[2]　高橋 健志[2]　竹内 純一[1]

概要：サイバー攻撃の件数は増加傾向にあり，悪性通信の検知やその攻撃手法の特定作業の負担およびその重要度は増している．それらを自動的に行うネットワーク侵入検知システム (NIDS) は多数存在しており，それぞれの検知手法や検知基準の違いによって異なる特性を持っている．異なる NIDS の特性を持ち合わせる新たな NIDS が構成できれば，それは単一のシステムでより良いセキュリティを提供できることが見込まれる．そのような NIDS を得る方法として考えられるのが，既存の NIDS の出力に基づいて生成したラベル付きデータを用いた教師付き学習を利用する方法である．本研究では，ニューラルネットワーク (NN) を用いた教師付き学習によって，入力された通信パケットを良性またはいずれかの攻撃手法に属する悪性パケットとして分類する NN ベースの NIDS の構成を目的とする．本稿では，NN の入力として用いる特徴量の設計として，オートエンコーダを用いた教師無し学習による異常検知に基づく軽量 NIDS である Kitsune に注目し，Kitsune が用いるものと同種の特徴量を入力とする NN を用いた多クラス分類を提案し，公開データセットを用いたパケット分類の実験を行うことで，データセットに含まれる各攻撃手法についての分類性能を評価した．一日分のデータだけを用いる実験においては，多くの攻撃種別に対して 90%台後半の適合率および再現率を達成できることが確認できた．

キーワード：ニューラルネットワーク，ネットワーク侵入検知

Malicious Packet Classification Using Kitsune Features

Kohei Miyamoto[1,a)]　Hiroki Goto[1]　Ryosuke Ishibashi[1]　Chansu Han[2]　Tao Ban[2]
Takeshi Takahashi[2]　Jun'ichi Takeuchi[1]

Abstract: Since cyberattacks are on the rise, detecting malicious communications and identifying attacks have become increasingly more challenging and critical. There are various network intrusion detection systems (NIDSes), which do that automatically. They employ different methods or criteria to detect attacks. This difference leads to their different characteristics. If we can construct a new NIDS which has the merits of different NIDSes, it will provide us with better security. In this study, we try to construct a neural network (NN)-based NIDS based on supervised learning, which classifies input packets into benign packets or malicious packets with identified their kinds of attack methods. By train a NN based on outputs of existing NIDSes, we will obtain a new NIDS which learned the characteristics of the NIDSes. In this paper, we employ features of the same kind used by Kitsune, a lightweight NIDS based on autoencoders, as inputs of a NN. We propose a method of multiclass classification of packets by using such a NN. We also provide an experimental result using an open dataset to evaluate the performance of the proposed method.

Keywords: neural network, network intrusion detection

1. はじめに

インターネットを介したサイバー攻撃の件数およびその多様性は年々増している．サイバー攻撃に対応するためには，通信ログの監視によるサイバー攻撃の疑いのある異常

[1] 九州大学
Kyushu University
[2] 国立研究開発法人情報通信研究機構
National Institute of Information and Communications Technology
[a)] miyamoto@me.inf.kyushu-u.ac.jp

な通信の検出およびそれらの通信の分析が重要となる．しかし，サイバー攻撃の件数やその種類が膨大となると，これらの作業を人力で行うことはセキュリティ担当者の負担が大きく，技術的にも難しい．そのため，企業等のネットワークではこれらの作業を自動的に行うネットワーク侵入検知システム (NIDS) が導入されることが多い．NIDS はネットワークのトラフィックを監視し，異常と思われるトラフィックを検知するとアラートを通知する．多くの場合，NIDS の出力したアラートとそれに付随する情報を元にしてセキュリティ担当者がより詳細な分析等を行う．様々な種類の NIDS が開発され，存在しており，それぞれが採用している異常検知の手法，あるいは同じ手法であっても用いている検知ルールなどのパラメータ等が異なっている．それゆえ，異なる NIDS は異常検知の性能や傾向に関して異なる特性を持つ．例えば，ある攻撃をある NIDS が検出する一方で別のある NIDS では検出できない，また別の攻撃ではその逆が起こるといった状況がありうる．

そのような事実を踏まえて，複数の NIDS をうまく組み合わせて利用することで，より良いセキュリティを実現できる可能性があるが，本研究ではそのようなセキュリティを単独で実現できるような NIDS の開発を目指して，教師付き学習に基いたニューラルネットワーク (NN) ベースの NIDS の構築を試みる．NN の学習を既存の複数の NIDS の出力を活用して得た教師データに基いて行うことで，それらの NIDS が持つ異なる特性を併せ持った新たな NIDS が得られることが期待される．構築する NIDS の形式として，通信パケットを入力とし，入力を良性通信の（正常）パケットもしくは特定の攻撃手法に属する悪性通信のパケットに分類する多値分類器を想定している．

NN による通信パケットの分類を行うにあたって，まず通信パケットから計算され NN の入力となる特徴量を設計する必要がある．本稿においては，NN を用いた軽量 NIDS である Kitsune [4] において提案されている特徴量を採用する．Kitsune は NN の中でも特にオートエンコーダ (AE) を用いた教師無し学習に基づく NN ベースの NIDS の一種である．リアルタイムで異常検知を行う NIDS として，Kitsune には特徴量の計算に要する計算時間やメモリ量が少ないという利点がある．

本稿においては，Kitsune の特徴量を入力とする NN を用いて，与えられた通信パケットに対して正常もしくはいくつかの攻撃手法をラベルとした多値分類を行う手法を提案する．また学習データとしてパケット毎にラベル付けが可能なデータセットとして，公開データセットである CSE-CIC-IDS2018 [1], [6] を用いた実験を行い，提案手法によるパケット分類性能を評価した結果を報告する．実験の結果として，データセットに含まれている攻撃種別の内，一部を除いた攻撃種別に関しては，適合率，再現率が共に 90%台後半となる程度の分類性能が提案手法によって達成可能であることが確かめられた．

2. 先行研究

本節ではまず，本稿の提案手法で用いる特徴量の背景となる先行研究として AE を用いた軽量 NIDS である Kitsune について述べる．また，本研究と同様に教師付き学習に基いた NIDS の構成に向けた先行研究として，既存 NIDS の出力を用いた教師付き学習用のデータセットの生成手法に関する研究も紹介する．

2.1 Kitsune

Kitsune [4] は Mirsky らが提案した NN ベースの軽量 NIDS の一種である．図 1 は [4] より引用した Kitsune のアーキテクチャ図である．Kitusne のシステム全体は，外部ライブラリを用いるパケットキャプチャおよびパケットのパースの段階を除くと，パケットをパースして得た情報から特徴抽出を行う Feature Extractor(FE)，抽出された特徴ベクトルを分割する Feature Mapper(FM)，そして複数の AE から成るアンサンブル構造と最終的な異常値スコアを出力する AE から成る Anomaly Detector(AD) の 3 要素から構成されている．

2.1.1 Feature Extractor(FE)

FE においては入力パケットに含まれる送信元および送信先の IP アドレス，MAC アドレス，ポート番号といった情報を元にしたいくつかの識別子毎に，入力されたパケットの個数と，パケットサイズおよびその二乗の時間減衰付きの累積和を管理する．これらをインクリメンタル統計量と呼ぶ．ここで時間減衰はパケットのタイムスタンプを利用して，同一識別子の最後の登場からの経過時間に対して指数的に減衰する係数を累積値に掛けることで行われる．各入力パケットに対して，関係するインクリメンタル統計量を更新した後，それらを用いて，関係する識別子毎のパケットサイズの平均や標準偏差，あるいは 2 つの識別子毎のパケットサイズの共分散，相関係数といった統計量を計算し，それらを並べたものを特徴ベクトルとして出力する．前者の 1 つの識別子に対して計算される統計量を 1D 統計量と呼び，後者の 2 つの識別子に対して計算される統計量を 2D 統計量と呼ぶ．

例として，同じ IP アドレス間での通信パケットが複数入力された時，それらのタイムスタンプが近い時刻だった場合には，時間減衰の影響は小さく，後から入力されたパケットに対しては以前の入力の影響が強く残った状態で特徴ベクトルが計算される．一方で，タイムスタンプが離れている場合には，時間減衰によって以前の入力の影響が大きく弱まった状態で特徴ベクトルが計算される．この仕組みによって入力パケットの時系列的な特徴が考慮され，

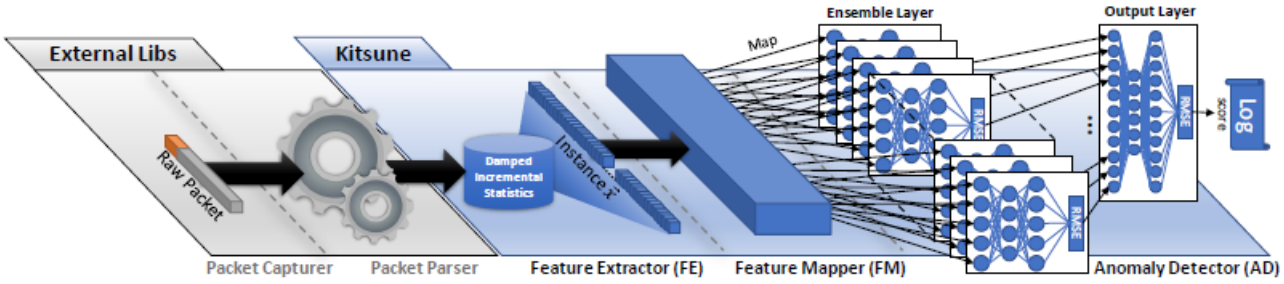


図 1: Kitsune のアーキテクチャ図 ([4] より引用)

DoS 攻撃の通信で期待されるような頻繁な通信といった特徴を捉えることができる．特徴抽出に関する詳細な計算式等は [4] の section Ⅳを参照されたい．

時間減衰を用いるために，時間減衰の定数 λ を設定する必要があり，本稿の実験で用いた [4] の著者が公開している参考実装 [8] においては $\lambda = 5, 3, 1, 0.1, 0.01$ の 5 通りを用いている．参考実装においては 1 つの λ と 1 つの入力パケットに対して出力されるベクトルは 20 次元であり，その内 12 次元が 1D 統計量，8 次元が 2D 統計量となっている．最終的な出力として，5 通りの λ 毎に抽出したベクトルを連結して得られる 100 次元ベクトルを入力パケットから抽出した特徴ベクトルとして出力している．

2D 特徴量を用いる場合，通信に加わるホストの種類が増えるにつれて各状態の更新に要する処理時間の増加率が大きいという問題があり，抽出部分の実装を高速な言語で置き換える，時間減衰を考慮した過去の影響が一定以下になった状態はハッシュテーブルから削除するなどの方法が [4] で既に提案されている．

2.1.2 Feature Mapper(FM)

FE の後に位置する FM の役割について述べる．FE が出力する特徴ベクトルに対して，FM はその分割を行う．分割はベクトルの要素間の相関に基いた階層的クラスタリングによって行われ，より相関の強い要素が分割後に同じベクトルに含まれるように分割される．適切な分割を実現するために，Kitsune の稼働の初期段階において一定数の入力パケットを用いて，分割に用いるクラスタリングを学習する．分割後の各ベクトルが AD のアンサンブル部分の AE への入力となる．

2.1.3 Anomaly Detector(AD)

最後に AE による異常検知を行う AD について述べる．Kitsune の AD は複数の AE から成るアンサンブル部分とそれに続く単一の AE である出力部から成る．アンサンブル部を構成する AE はそれぞれが，前述の FM による分割後の特徴ベクトルを入力とするように対応している．FM から分割された特徴ベクトルが与えられると，各 AE に入力としてそれらが渡され，それぞれの AE によって再構成したベクトルが出力される．ここで，入力された各ベクトルと各 AE の出力するそれらの再構成結果を用いて，再構成誤差が二乗平均平方根誤差 (RMSE) を損失関数として計算される．この再構成誤差を並べたベクトルがアンサンブル部の出力として，出力部の AE の入力として渡される．出力部の AE においても同様に，アンサンブル部の出力ベクトルの再構成を行い，RMSE を損失関数とする再構成誤差を出力する．この出力部の再構成誤差が Kitsune の算出する異常値スコアとなる．各 AE に再構成を学習させるために，Kitsune の稼働後しばらくは正常通信のパケットが入力されると仮定して，FM の学習後の一定数パケットに対しては正常通信のパケットを想定した学習モードとして動作を行い，各 AE がそれぞれの入力を用いて学習を行う．正常通信のパケットを入力として AE の再構成を学習しているため，異常通信に対しては各 AE が出力する RMSE が大きくなることが期待されることを利用して，最終的に出力された異常値スコアが適当に設定した閾値を超えるような場合には入力が異常通信であったと判断してアラートを出すというような利用ができる．

2.2 パケットデータへのラベリング

本研究の最終的な目標は，教師データを通じて既存の NIDS の特性を学習した教師付き学習ベースの新たな NIDS を構成することである．そのためには，パケットデータに対して既存の NIDS の出力に基いたラベリングを施した教師データを作成する必要がある．しかし，既存の NIDS の多くは検出した攻撃に関してアラートを出力するものの，アラートと検出された攻撃を構成する一連のパケットとの関連付けは行わない．そのため，単に既存 NIDS の出力を得るだけでは，入力したパケットデータに対するラベリングは行なえない．また，NIDS の出力と入力パケットを元に人力でその関連付けを行うことは作業量が膨大となり現実的ではないと考えられる．この問題に関して，NIDS の出力アラートと原因となった通信セッションの関連付けを行う手法が Ishibashi ら [2] によって提案されている．この手法は，本稿の提案手法に関して将来的に本稿における実験内容よりも実践的な評価および運用をするにあたって，前提となる学習データの作成方法として有用であると考え

られる．

2.3 関連研究

パケットデータへのラベリングに関して，パケットデータを再送することで既存 NIDS のアラートを得て，それを元にラベリングを行う手法が Masumi ら [3] によって提案されており，先行研究 [2] を通じて本研究とも親和性があると考えられる．

本研究の背景としてサイバー攻撃への対処の自動化に関する需要がある．背景を同じくする大観的な取り組みとして，サイバー攻撃の様々な分析手法を統合したプラットフォームの構築が Takahashi ら [7] によって試みられている．本研究の位置付けは，そのようなプラットフォームを構成するための要素技術の開発といえる．

3. 提案手法

本稿で提案する手法は，前節で述べた Kitsune の FE の出力を入力特徴量として多値分類を行うことである．図 2 に全体の概要を示す．前半部分は Kitsune の FE までと共通で，パケットのキャプチャおよびパースを行った後，FE にその結果を入力し，特徴抽出を行う．提案手法においては，得られた特徴ベクトルを教師付き学習に基く多値分類器に入力し，分類を行う．本稿においては分類器のモデルはシンプルに，適当な中間層とデータに含まれるラベルの種類と同数の出力ユニットを持つソフトマックス層から成る NN を用いる．

モデルの学習用のデータとしてキャプチャしたパケットとそのパケットに対するラベルの組の列が与えられているとし，それを用いた NN の学習を次のように行う．まず，パケット列を FE に時系列順に入力し，Kitsune と同様の特徴抽出を行う．これによって，各パケットと対応した特徴ベクトルの列が得られる．ただし，後述するデータセットを用いる場合，Kitsune の参考実装 [8] における 2D 特徴量を含む 100 次元の特徴ベクトルを用いた際に， [4] において提案されている高速化手法を採用した上でなお膨大な抽出時間が必要となったことから，本稿の実験においては 2D 特徴量を用いず 1D 特徴量のみを用いて実験を行っている．したがって以下で扱う特徴ベクトルは 1D 特徴量のみから成る 60 次元のベクトルである．NN の教師付き学習に用いるデータとして，この特徴ベクトルと与えられていたラベルの列の組を用いる．パケット列の時系列情報はこの特徴抽出の段階でのみ利用し，抽出された特徴ベクトルは標準的な教師付き学習手法と同様に独立なものとして扱い，シャッフルやテストデータ，検証データへの分割処理を行い，ミニバッチ学習によって NN を学習する．学習後の運用としては，実際にキャプチャしたパケットを逐次的に FE に入力し特徴抽出を行い，特徴ベクトルを逐次的に学習済み NN に入力することでその出力による分類結果をリアルタイムに得ることができる．

4. 実験とその評価

前節で述べた方針による学習を行った場合の NN の分類性能を評価した実験について報告する．

4.1 CSE-CIC-IDS2018 データセット

まず，今回の実験に用いた公開データセットである CSE-CIC-IDS2018 [1], [6] について述べる．このデータセットは標的ネットワークにおける統計的モデリングを利用した正常通信のシミュレーションと，標的ネットワークに対する様々な種類の攻撃ツールの実行によって作成されている．標的ネットワークは企業内部のネットワークを想定しており，400 台の Windows マシンと 20 台の Linux(Ubuntu) マシンおよび 30 台の Windows サーバーから構成されている．一方で攻撃ネットワークは，標的ネットワークの外部に存在する 50 台のマシン群（Windows および Ubuntu マシン）から成る．

このデータセットは 10 日間のデータを含んでおり，各日の標的ネットワーク内の IP アドレス毎にキャプチャされたパケットデータが pcap 形式で提供されている．また，各日毎に実行された攻撃の種別と，攻撃を実行した時間帯および標的となった IP アドレス，攻撃を実行したマシンの IP アドレスの情報が別途提供されており，今回の実験ではそれらの情報を元に該当するパケットを攻撃に関与するパケットとしてラベル付けを行ってラベル付きデータとして用いている．また，今回の実験では用いていないが，パケットデータとは別に CICFlowMeter-V3 [5] による特徴抽出の結果と標的ネットワーク内の各マシンのシステムログも提供されている．

4.2 特徴抽出とラベリング

前述したようにデータセットの各日毎に攻撃の種類とその標的となるマシンの IP アドレスと攻撃を行うマシンの IP アドレスおよび攻撃を行う時間帯が与えられている．例えば 2 月 14 日のデータにおいては，FTP-BruteForce と名付けられた攻撃があり，IP アドレス 172.31.69.25 のマシンを標的として，IP アドレス 18.221.219.4 のマシンから 10:32 から 12:09 の間に攻撃が実行されている．

したがってこの日付に関しては，攻撃に関するパケットは標的である IP アドレス 172.31.69.25 のマシンにおいてキャプチャされたパケット群に含まれており，該当する pcap ファイルを用いて特徴抽出を行った．特徴抽出の処理は参考実装 [8] に対して，抽出時間の高速化のための工夫として，抽出処理を Cython 化し，さらに時間減衰の係数が 0.1 以下になった状態を消去する処理を加え，1D 特

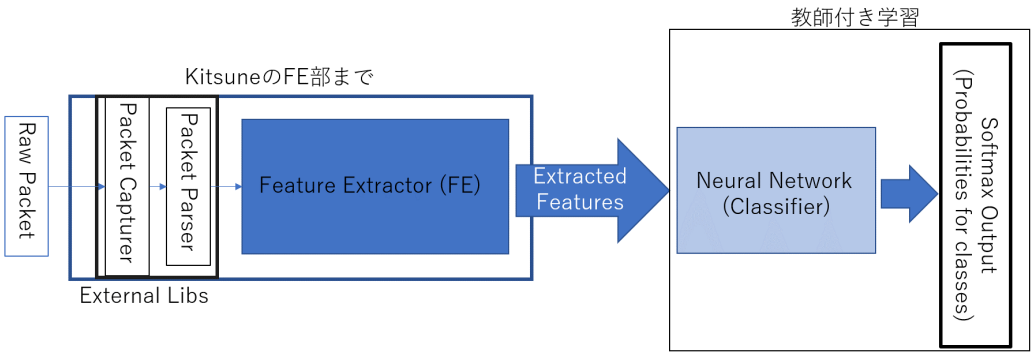


図 2: 提案手法の概要図

徴量だけを出力するように修正を施したものを用いた．ラベリングは前述の情報に基いて，10:32 から 12:09 の間に攻撃マシンの IP アドレス 18.221.219.4 から送信されたパケットを攻撃に関与しているものと考え，FTP-BruteForce のラベルを付与することで実施した．同様の特徴抽出およびラベリングを日付，攻撃種別毎に行い，攻撃のラベルが付けられなかったパケットに対しては，正常を意味する BENIGN ラベルを付与した．

4.3 実験内容

前述の方法で日付毎に抽出およびラベリングを行ったデータを用いて，分類器の学習および学習後の分類性能の評価を行った．各日付のデータに対して，その日のデータのみを用いた高々 4 値分類の実験と，全 9 日のデータを混ぜたデータ（統合データ）を用いた 15 値分類の実験の二通りを行った．後者の実験のほうがより実践的な設定と考えられる．

各日付のデータのみを用いる実験に関しては，その日付の標的 IP に関するデータを全て用いて実験を行った．一方で，統合データを用いる実験については，単に全日付の標的 IP に関するデータを混ぜてしまうと，各日付において含まれるパケット数に偏りがある事実と，さらに同日付内においてもクラス毎に該当するパケット数に偏りがあるという事実から，各クラスのデータ数の不均一がより強調されてしまう恐れがあるため，統合データを次のような手順で行った．各日付の標的 IP に関するデータからクラス毎に 20000 パケットずつランダムに抽出し，それを等分して学習データとテストデータのセットを作成する．ただし，元のデータに含まれているパケット数が 20000 パケットに満たないクラスに関しては，全て抽出し同じく等分する．

日付毎の実験においては，用いるデータを比率が 2:1:1 になるように学習データ，検証データ，テストデータにランダム分割し，学習を行った．なお，分割時に各ラベルの比率も均等になるような分割としている．統合データを用いる実験においては，前述の手順で作成したテストデータを再び等分して一方を検証データとして用いている．いずれの実験においても各特徴量に対する前処理として，学習データ中の平均値と標準偏差を用いた標準化を行っている．

NN の実装は Tensorflow を用いて，学習はマルチクラスクロスエントロピーを損失関数とし，最適化アルゴリズムには Adam を用い，学習率の初期値はデフォルト値である 0.001 として行った．また，検証データに関して損失関数が悪化した時点で学習を打ち切るようにした．また学習に関してミニバッチサイズは 32 とした．

以上の実験を各日付のデータと統合データに対して一回ずつ行い，学習済みのモデルの分類性能の評価を行った．

4.4 分類器の構成

今回の実験では簡易なモデルとして日付毎の実験においては 4 ユニットの中間層を 1 つだけ持ち，該当する日付のラベル種別数と同じユニット数を持つソフトマックス層を出力層とする 3 層 NN を用いて実験を行った．統合データを用いた実験においては同じく中間層のユニット数を 8 とした 3 層 NN を用いた．

4.5 評価指標

学習後のモデルの性能の評価指標としてクラス毎の accuracy(精度), precision(適合率), recall(再現率), F-measure の 4 通りを用いた．学習後のモデルが特徴ベクトル x に対して出力する予測クラスを $p(x)$，テストデータの特徴ベクトルとラベルの組の集合を $\mathcal{T}$，分類対象となるクラスの集合を $\mathcal{C}$，I を指示関数として，各クラス $c \in \mathcal{C}$ に対して

$$\mathrm{TP}(c) = \sum_{(x,y)\in\mathcal{T}} I(p(x) = c, y = c) \quad (1)$$

$$\mathrm{TN}(c) = \sum_{(x,y)\in\mathcal{T}} I(p(x) \neq c, y \neq c) \quad (2)$$

$$\mathrm{FP}(c) = \sum_{(x,y)\in\mathcal{T}} I(p(x) = c, y \neq c) \quad (3)$$

$$\mathrm{FN}(c) = \sum_{(x,y)\in\mathcal{T}} I(p(x) \neq c, y = c) \quad (4)$$

と定義するとき，それぞれ

$$\mathrm{accuracy}(c) = \frac{\mathrm{TP}(c) + \mathrm{TN}(c)}{\mathrm{TP}(c) + \mathrm{TN}(c) + \mathrm{FP}(c) + \mathrm{FN}(c)} \quad (5)$$

$$\mathrm{precision}(c) = \frac{\mathrm{TP}(c)}{\mathrm{TP}(c) + \mathrm{FP}(c)} \quad (6)$$

$$\mathrm{recall}(c) = \frac{\mathrm{TP}(c)}{\mathrm{TP}(c) + \mathrm{FN}(c)} \quad (7)$$

$$\mathrm{F}(c) = \frac{2\mathrm{precision}(c) \cdot \mathrm{recall}(c)}{\mathrm{precision}(c) + \mathrm{recall}(c)} \quad (8)$$

と定義される $[0, 1]$ 上に値を取る量である．

クラス $c \in \mathcal{C}$ だけに注目した場合に，$\mathrm{accuracy}(c)$ はテストデータ全体に対しての正答率，$\mathrm{precision}(c)$ はクラス c であるという予測に限定した正答率，$\mathrm{recall}(c)$ はクラス c が正答であるテストケースに限定した正答率を意味している．多くの場合，precision と recall はトレードオフの関係にあり，F-measure はそれらの調和平均として定義された総合的な指標とみなせる．

4.6 実験結果

図 3 に日付毎の実験結果を示す．縦軸はクラス名と実験データの日付であって，横軸は対応する評価指標の値である．同様に図 4 に統合データを用いた実験結果を示す．縦軸はクラス名であって，横軸は対応する評価指標の値である．

また，参考として，日付毎のデータおよび統合データのそれぞれについてテストデータ中に含まれている各クラスのデータ数をそれぞれ図 5 と図 6 に示す．データ数に関して，横軸は対数スケールであることに注意されたい．

5. 考察

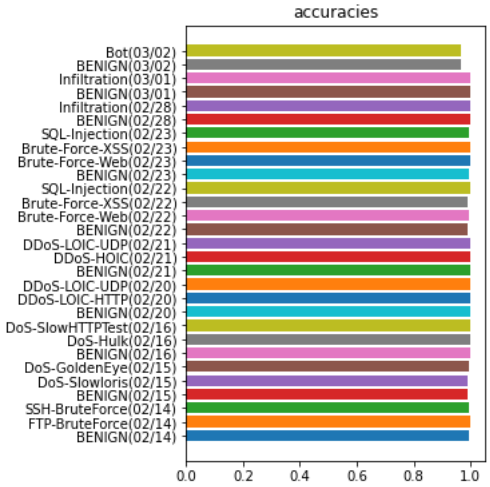
図 3 の日付毎の実験結果をみると SQL-Injection と Infiltration を除くクラスに関してはどの日付においても，1 に近い F-measure を実現できており，分類が上手く行えているといえる．一方で SQL-Injection と Infiltration に関しては分類が上手く行えていない．特に 02/22 のデータにおける SQL-Injection は precision, recall 共に 0 となっており，これは全く検知できていないことを意味する．accuracy に関しては SQL-Injection と Infiltration に関しても高い値となっているが，これは図 5 から分かるように，これらのクラスのデータがデータ全体に対してごく少数しか含まれていないというデータの不均衡の影響が大きいと考えられる．しかし，攻撃の種別によって検知のしやすさ等の性質が異なることも考えられるため，今後も調査が必要である．

図 4 の統合データを用いた実験の結果をみると，全体的に分類性能が悪化していることが確認できる．評価尺度を個別にみると，precision が全体的に低い値になっている一方で，recall は正常データのクラスである BENIGN が低めになっているが，他のクラスは FTP-BruteForce を除いて高い値である．これは正常通信の多くを異常通信として誤検知していることを意味しており，攻撃のクラスに関して recall が高いことから誤検知をしつつも実際の攻撃の検知自体は十分に行えているといえる．個別のクラスで注目すべきなのが，FTP-BruteForce，SQL-Injection および Infiltration である，FTP-BruteForce に関しては，前述の日付毎の実験においては十分な分類性能を達成していたクラスである一方で，統合データを用いた実験においては recall が 0 であり全く検知できていない．これは，日付毎の実験においては FTP-BruteForce と同時に扱っていなかった攻撃のクラスの中に，FTP-BruteForce を誤分類してしまう傾向の強いものが含まれているのではないかと考えられる．SQL-Injection および Infiltration に関しては，precision は低いものの recall は高くなっており，各日付毎の実験において recall が低かったこととは異なる結果となっている．これらのクラスに関して，図 6 から分かるように，データ数が全体に対して少ないという状況は統合データにおいても同様であるため興味深い挙動である．

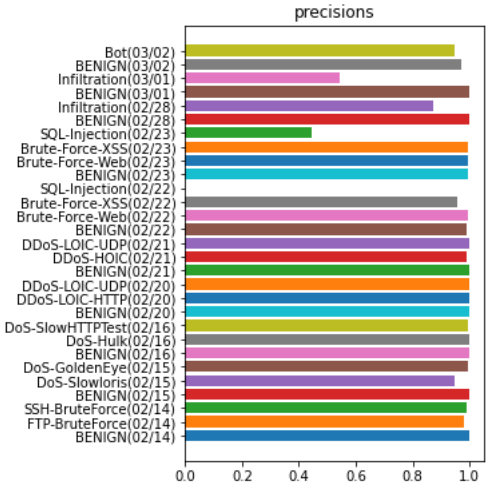
6. まとめ

本稿では，教師無し学習に基づいた異常検知を行う軽量 NIDS である Kitsune が用いる特徴量を利用して，教師付き学習に基いたパケット分類を行う NN ベースの手法を提案し，公開データセット CSE-CIC-IDS2018 を用いた分類実験について報告した．実験においては，分類器はユニット数の少ないシンプルなモデルとしたが，一日分のデータを用いて少数の攻撃種別に関する分類を行う場合は多くの攻撃種別に関しては，十分な分類性能が得られることが確認できた．一方で，数日分のデータを統合したデータを用いて多数の攻撃種別に関する分類行う場合は，前述の場合と比べて分類性能に関して，特徴的な劣化がみられることが確認できた．

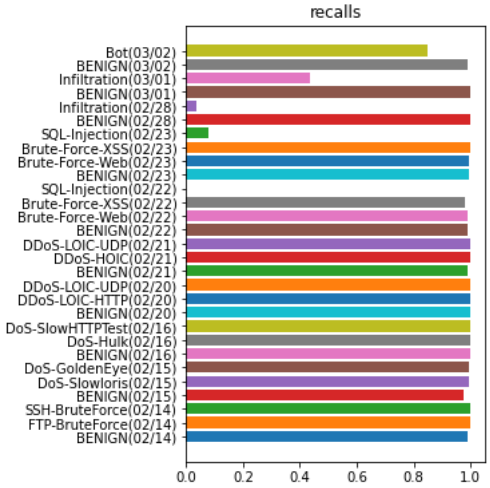
今後の課題としては，今回行った実験に関して，実験回数を増やし，分類性能に関してより統計的な評価を行うことが第一に上げられる．また，今回の実験で確認された，分類性能の問題点に関して，モデルの改良や分類手法の改良による改善についても検討する必要がある．例えば，NN の再帰構造を用いて NN 側でも時系列情報を活用することや，NN の出力を直接用いてパケット毎の分類を行うの



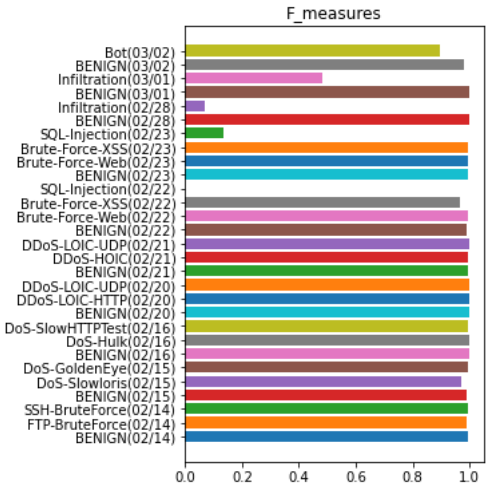
(a) 日付毎の accuracy



(b) 日付毎の precision



(c) 日付毎の recall



(d) 日付毎の F-measure

図 3: 日付毎の実験結果

ではなく，複数パケットに対する出力を総合し通信のセッション単位での分類を行うことで，より実践的な NIDS としての出力が得られる可能性がある．

謝辞　本研究は総務省の「電波資源拡大のための研究開発 (JPJ000254)」における委託研究「電波の有効利用のための IoT マルウェア無害化／無機能化技術等に関する研究開発」によって実施した成果を含む．

参考文献

[1] Iman, S., Arash, H. L. and Ali, A. G.: “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization”, *4th International Conference on Information Systems Security and Privacy (ICISSP)* (2018).

[2] Ishibashi, R., Goto, H., Han, C., Ban, T., Takahashi, T. and Takeuchi, J.: ”Which Packet Did They Catch? Associating NIDS Alerts with Their Communication Sessions”, *The 16th Asia Joint Conference on Information Security* (2021).

[3] Masumi, K., Han, C., Ban, T. and Takeshi, T.: Towards Efficient Labeling of Network Incident Datasets Using Tcpreplay and Snort, *the Eleventh ACM Conference on Data and Application Security and Privacy* (2021).

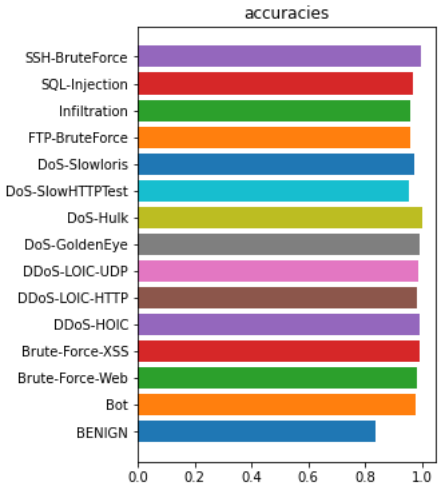
[4] Mirsky, Y., Doitshman, T., Elovici, Y. and Shabtai, A.: ”Kitsune: an ensemble of autoencoders for online network intrusion detection”, *Network and Distributed System Security Symposium 2018* (2018).

[5] Online: CICFlowMeter. `https://www.unb.ca/cic/research/applications.html`.

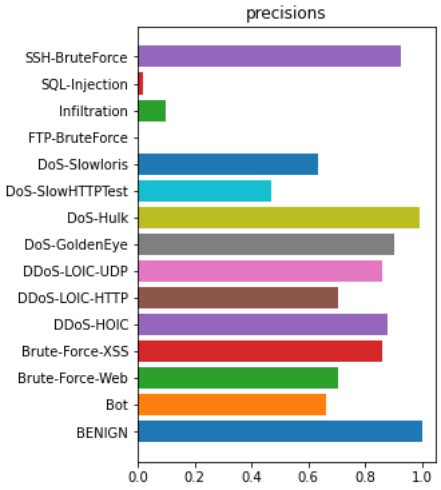
[6] Online: A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018). `https://registry.opendata.aws/cse-cic-ids2018/`.

[7] Takahashi, T., Umemura, Y., Han, C., Ban, T., Furumoto, K., Nakamura, O., Yoshioka, K., Takeuchi, J., Murata, N. and Shiraishi, Y.: Designing Comprehensive Cyber Threat Analysis Platform: Can We Orchestrate Analysis Engines?, *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)* (2021).

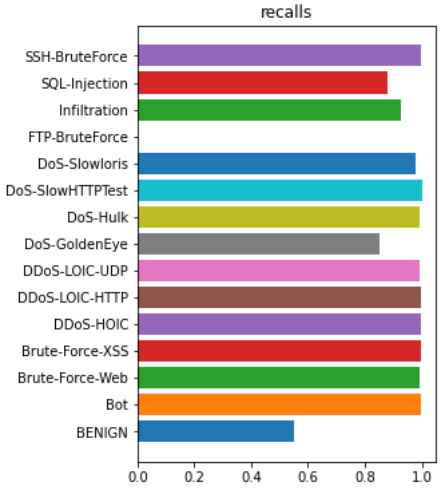
[8] ymirsky: Kitsune-py. `https://github.com/ymirsky/Kitsune-py`.



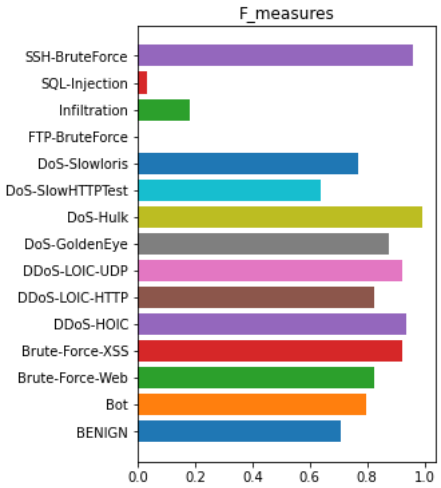
(a) 統合データに関する accuracy



(b) 統合データに関する precision



(c) 統合データに関する recall



(d) 統合データに関する F-measure

図 4: 統合データを用いた実験結果

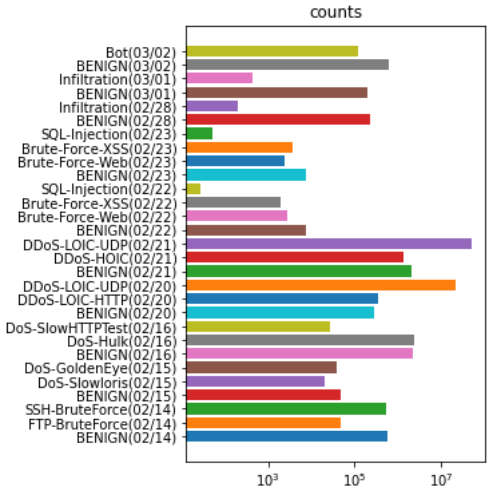


図 5: 日付毎のテストデータ数内訳

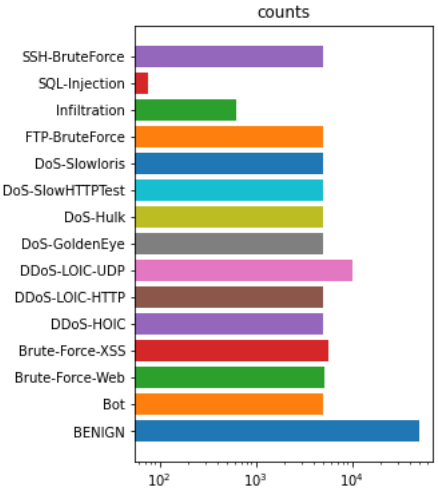


図 6: 統合データにおけるテストデータ数内訳