未知の攻撃の検知のための 増分学習と組合せたカーネル近似学習削減の性能への影響

山上　理久† 　Le Hieu Hanh† 　横田　治夫†

† 東京工業大学 情報工学系 横田治夫研究室　〒 152–8552 東京都目黒区大岡山 2–12–1
E-mail: †{yamagami,hanhlh}@de.cs.titech.ac.jp, ††yokota@cs.titech.ac.jp

あらまし　未知の攻撃の検知によく用いられる OneClassSVM は, 高精度で分類できる一方で, 増分学習非対応である. 増分学習対応の線形分離を行う SGDOneClassSVM は増分学習非対応のカーネル近似と組み合わせることで, OneClassSVM と同様の働きをする. 本研究では, カーネル近似の学習の頻度を落とした上で, SGDOneClassSVM で増分学習を行う手法を提案する. カーネル近似学習の削減と組み合わせた増分学習はカーネル近似学習の頻度低下による分類精度の低下を軽減し, 提案手法では毎回バッチ学習を行う OneClassSVM と比較して学習時間が約 5 倍高速化することを明らかにした.

キーワード　機械学習、時系列データ処理、増分学習

1 序　　論

1.1 背　　景

近年, インターネットトラフィックは増加しており [1], またサイバー攻撃も増加している [2] ため, ネットワークログから素早く攻撃を検知することが必要である. サイバー攻撃は進化し続けており [3], これに対応するためにさまざまな機械学習手法が提案されてきた. 進化し続けるサイバー攻撃は, 過去に類似する攻撃が存在する既知の攻撃と過去の攻撃とはかけ離れた未知の攻撃に分類できる. ネットワークログには攻撃か正常かのラベルが存在しない [4] ことから, 未知の攻撃の検知には教師なし学習が用いられている. 教師なし学習手法の一つである OneClassSVM は, 分類精度が高い一方でバッチ学習しかできないため学習時間が長いといった特徴を持つ.

1.2 目　　的

OneClassSVM の学習時間が長い原因として, OneClassSVM は基本的にパラメータの更新時に学習対象の訓練データを全て用いるバッチ学習で学習しており, 訓練データが増えるたびに増加データのみを用いて逐次的にモデルを改良して行く増分学習で学習することができないことがある. 線形分離しか行えず分類精度の悪い線形 OneClassSVM である SGDOneClassSVM は増分学習で学習できる一方で, 非線形分離を行うカーネルトリックを用いてカーネル化する OneClassSVM はバッチ学習でしか学習することができない. さらに, サンプル数に対する複雑さが SGDOneClassSVM は線形で増加するのに対し, カーネル化する OneClassSVM は良くて二次関数的に増加することもある. SGDOneClassSVM は増分学習非対応のカーネル近似と組み合わせることで, カーネル化する OneClassSVM と同様の働きをする.

本研究では, カーネル近似学習の頻度を落とした上で, SGDOneClassSVM で増分学習を行う手法を提案する. カーネル近似の学習を毎回行わないため, 分類精度の低下が予想されるが, OneClassSVM の学習をバッチ学習から増分学習にすることで学習時間の短縮が予想される. 増分学習と組み合せたカーネル近似学習の削減が性能に与える影響に関する実験結果を報告する.

本研究では, 増分学習と組合せたカーネル近似学習の削減が性能に与える影響を明らかにする. 具体的には以下の Research Question に回答する.

- RQ1 増分学習と組み合わせたカーネル近似学習の削減が分類精度にどのように影響しているか.
- RQ2 増分学習と組み合わせたカーネル近似学習の削減が学習時間にどのように影響しているか.

1.3 貢　　献

本論文の貢献は以下のとおりである.

- 増分学習と組み合わせたカーネル近似学習の削減は学習頻度低下に伴う分類精度の低下を軽減することを明らかにした.
- 増分学習と組み合わせたカーネル近似学習の削減は学習頻度に比例するように学習時間を短縮させ, 毎回バッチ学習を行う OneClassSVM と比較して, 提案手法は約 5 倍高速化することを明らかにした.

1.4 構　　成

2 節では, 本研究に関連する研究を紹介する. 3 節では, 本研究で提案する手法について説明する. 4 節では, 提案手法を評価する実験の設定と手順について説明する. 5 節では, 実験の結果とその考察について述べる. 6 節では, 本論文の結論を述べる.

2 関連研究

2.1 サイバーセキュリティへの機械学習の適応

サイバーセキュリティ分野では機械学習を用いた手法について, さまざまな研究がされている. 機械学習のサイバーセキュリティへの適用は, 手法の学習が教師あり学習か教師なし学習かにより異なる場面に適用される [5], [6].

教師あり学習は, ある入力の集合から到達すべき特定の目標が定義されている場合に行われ, 特定のセキュリティ問題に対して, 分類または将来を予測するために普及している. 教師あり学習手法は大きく分類法と回帰法に分けられる. 分類法は, サービス妨害攻撃の予測やネットワーク攻撃の異なるクラスの識別に用いられている. 手法としては, ZeroR 法, OneR 法, ナイーブベイズ法, 決定木, K 近傍法, サポートベクターマシン, AdaBoost, ロジスティック回帰などが知られている. 回帰法は連続値や数値の予測に用いられ, 手法としては, 線形回帰, サポートベクトル回帰が知られている [5].

教師なし学習は, ラベルのないデータからパターンや構造, 知識を見つけ出す場面で行われ, マルウェアのような検知されないように動的かつ自律的に振る舞いを変えるサイバー攻撃を対象とすることが多い. 一般的な教師なし学習手法であるクラスタリング手法は, 異常の特定, ポリシー違反, データ中のノイズの多いインスタンスの検出に用いられている. 手法としては, K-means 法や K-medoids 法が知られている [5].

2.2 侵入検知への機械学習の適用

ネットワーク侵入検知の分野では機械学習を用いた NIDS について, さまざまな研究がされている [6]. ネットワーク侵入検知は大きく誤用検知と異常検知の 2 つに分類される. 誤用検知の場合は, 訓練対象データを用いて各誤用クラスを学習し, 学習したモデルを用いて検査対象データが誤用クラスのいずれかに属するかどうかを分類する. どの誤用クラスにも属さない場合は, その検査対象データは正常であると分類する. 異常検知の場合は, 訓練対象データを用いて正常な状態を学習し, 学習したモデルを用いて検査対象データが正常な状態から離れているかどうか, 異常か正常かに分類する.

また, 取られるアプローチにより 2.1 節と同様に教師なし, 半教師付き, 教師ありの 3 種類のアプローチに分類される [6].

2.3 SVM による分類

サポートベクターマシン (SVM) は, 訓練データから各データポイントとの距離であるマージンが最大となる超平面を求め用いることで, テストデータの分類を可能とする教師あり学習の分類器である [7]. また, 線形分離不可能なデータセットに対して, 多少の誤分類を許すソフトマージンを導入することで, 多くのデータは正しく分類されるようにしている. 誤分類を許されるデータポイントの数や誤分類に対する罰則はユーザーが指定するハイパーパラメータによって制御され, 誤分類とマージンの大きさはトレードオフとなっている. ここでデータセットに対する誤分類の少なさは特化性能であり, マージンの大きさは未知のデータに対する分類性能である汎化性能である. さらに, 図 1 のようにカーネル関数を用いて各データポイントを高次元にマッピングすることで低次元では良い分類ができないデータセットに対する分類を可能としている. データポイントを写像した高次元空間での特徴の計算を避け, カーネルの計算のみで識別関数の構成をするカーネルトリックが用いられている [8]. カーネルトリックを用いることで, 識別関数が構造的には従来の 3 層ニューラルネットワークと等しくなっている. カーネルトリックを用いて非線形に拡張した SVM では, 中間層に非常に多くのユニット用いることで, 高次元空間への複雑な非線形写像を構成している.

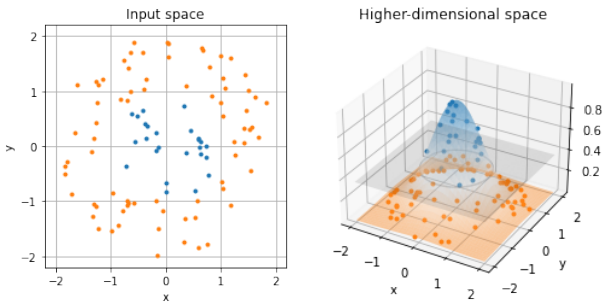

図 1: 高次元空間写像による非線形分類

2.4 偏りのあるデータセット

侵入検知の実際のアプリケーションでは, 多数派クラスと少数派クラスのサンプル数が 100 : 1, 1,000 : 1, 10,000 : 1 やそれ以上の偏りがあることがある [9]. このため, 偏りのあるクラス分類では, 多数派クラスに圧倒され少数派クラスが無視されてしまうことがある問題がある. さらに, 概念の複雑さの程度が増加すればするほど偏りに対する感度は増加する [10]. この問題に対して, サンプリングによりクラス間の偏りを解消する方法や少数派クラスのコストを調整する方法, 決定閾値を調整する方法や識別ベースの 2 クラス分類ではなく認識ベースの 1 クラス分類手法が研究されている [9], [11].

2.5 OneClassSVM

OneClassSVM は, 2 クラス分類である SVM を 1 クラス分類に適応した方法である. 訓練データ全てを正例と, 原点を負例とラベル付けすることで SVM を用いた 1 クラス分類を可能としている [12]. SVM と同様にカーネル関数を用いてデータポイントを高次元特徴空間にマッピングしており, このカーネルトリックにより非線形分類を実現している.

カーネル関数 K が RBF カーネル $\left(K(\mathbf{x},\mathbf{x}') = \mathrm{e}^{-\alpha(\mathbf{x}-\mathbf{x}')^2}\right)$ のようにある定数 C に対して $K(\mathbf{x},\mathbf{x}) = C$ を満たす場合, すべての点 $\mathbf{x} \in \mathbb{R}^n$ はヒルベルト空間 $\mathcal{H}$ においては, $K(\mathbf{x},\mathbf{x}) = ||\phi(\mathbf{x})||^2 = C$ なので, $\mathcal{H}$ において半径 $\sqrt{C}$ の超球面上に写像される. これより, 2 つのクラスが線形分離可能であれば異常のデータポイントも超球面上に写像する制約があるので, 図 2 のように正常のデータポイントと超球の中心との間に最大限のマージンを置く超平面で分離することができる [13].

侵入検知の実際のアプリケーションでは, 各データポイントが攻撃であるか正常であるかのラベル付けはされていない [4] ため, 教師なし学習が適している. 1 クラス分類を行う OneClassSVM は教師なし学習を行う.

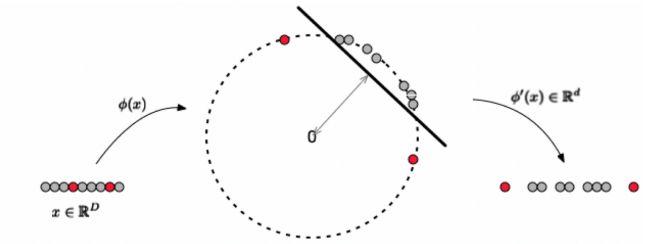

図 2: OneClassSVM [13, Figure 1]

2.6 確率的勾配降下法

勾配降下法はある位置ベクトル $\mathbf{w}$ から式 1 のように最急降下方向に進むことを繰り返し目的関数 E の最適解を求める手法である [14].

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} \tag{1}$$

しかし, 扱うデータの次元数が大きくなると $\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}$ の計算に時間がかかってしまう. これを解消するために, 確率的勾配降下法では $\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}$ の代わりに $\frac{\partial E_i(\mathbf{w})}{\partial \mathbf{w}}$ を用いる. このために目的関数 E は $E(\mathbf{w}) = \sum E_i(\mathbf{w})$ と微分可能な関数の和の形であることを必要とする. 確率的勾配降下法では各反復にてランダムに選ばれた i を用いて式 2 のように $\mathbf{w}$ を更新することで最適解を求める [14].

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \frac{\partial E_i(\mathbf{w})}{\partial \mathbf{w}} \tag{2}$$

確率的アルゴリズムは, これまでの反復の間にどのデータを用いたかを覚えておく必要がなく, 途中の状態から最適化を始めることができる.

この確率的勾配降下法を SVM の最適化に用いる SGD-SVM では増分学習が可能である. しかし, SGD-SVM は線形 SVM であり, 線形分離でしか分類ができない [14]. OneClassSVM の場合も同様に, OneClassSVM の最適化に確率的勾配降下法を用いる SGDOneClassSVM は増分学習が可能であるが, 線形分離でしか分類することができない [15].

3 提 案 手 法

3.1 概　　要

OneClassSVM はカーネルトリックを用いてデータポイントを高次元特徴空間にマッピングしながら最適化する. OneClassSVM は増分学習ができず, 学習の度に訓練データを全て用いて学習するバッチ学習しかできない. 訓練に用いるデータ数が多い場合, 毎回の学習に時間が多くかかってしまう問題がある. これを解消するために, OneClassSVM の学習頻度を下げることにより, 平均的に学習時間を短縮する方法が存在する. しかし, 訓練データの内容が大きく変化する時系列データを用いる場合, 学習頻度を下げることにより分類精度が低下することがある問題がある.

SGDOneClassSVM は確率的勾配降下法を用いる線形 OneClassSVM である. 線形 SVM のため, 線形分離不可能なデータセットには分類精度が大きく低下する. 確率的勾配降下法を用いているため, 増分学習が可能である. カーネル近似と組み合わせることにより, カーネル関数を用いる OneClassSVM の非線形分類と同様の働きをすることができる.

カーネル近似はカーネルトリックを用いずにて直接的に高次元特徴空間をデータの一部を用いて近似的に計算する. データの一部を用いて近似的に計算することにより, 全てのデータを計算するよりは学習時間が短縮されるという特徴がある.

本研究では OneClassSVM の問題を解消するために, OneClassSVM と同様の働きをする SGDOneClassSVM とカーネル近似の組み合わせを用いる. OneClassSVM では, カーネルトリックを用いて高次元空間に写像しながら最適化をしていたため, 学習全体が増分学習不可能である. しかし, カーネル近似の学習は増分学習不可能であるが SGDOneClassSVM 単体の学習は増分学習可能であるため, SGDOneClassSVM とカーネル近似の組み合わせは部分的に増分学習可能となる. 本研究では SGDOneClassSVM の増分学習とカーネル近似学習の削減を組み合わせた手法を提案する.

本手法では, カーネル近似学習の頻度を落とした上で, SGDOneClassSVM で増分学習を行う. カーネル近似の学習を毎回行わないため, 分類精度の低下が予想されるが, OneClassSVM の学習をバッチ学習から増分学習にすることで処理時間の短縮が予想される. 本手法は, 3.2 節のカーネル近似学習を伴う分類と 3.3 節のカーネル近似学習を伴わない分類の 2 種類を組み合わせて行う. 図 3 のように学習と標準化・カーネル化・分類を行う.

3.2 カーネル近似学習を伴う分類

本節では, 訓練対象データを用いて標準化器の学習とカーネル近似の学習, SGDOneClassSVM のバッチ学習を行った上で, 検査対象データの分類を行う. この際に, SGDOneClassSVM は増分学習ではなくバッチ学習を行うことで忘却を行う. 図 4 のように以下のステップで検査対象データの分類を行う.

a.1. 訓練対象データを用いて標準化器を学習

a.2. 標準化器を用いて訓練対象データを標準化

a.3. 標準化訓練対象データを用いてカーネル近似を学習

a.4. カーネル近似を用いて標準化訓練対象データをカーネル化

a.5. カーネル化標準化訓練対象データを用いて SGDOneClassSVM をバッチ学習

a.6. 標準化器を用いて検査対象データを標準化

a.7. カーネル近似を用いて標準化検査対象データをカーネル化

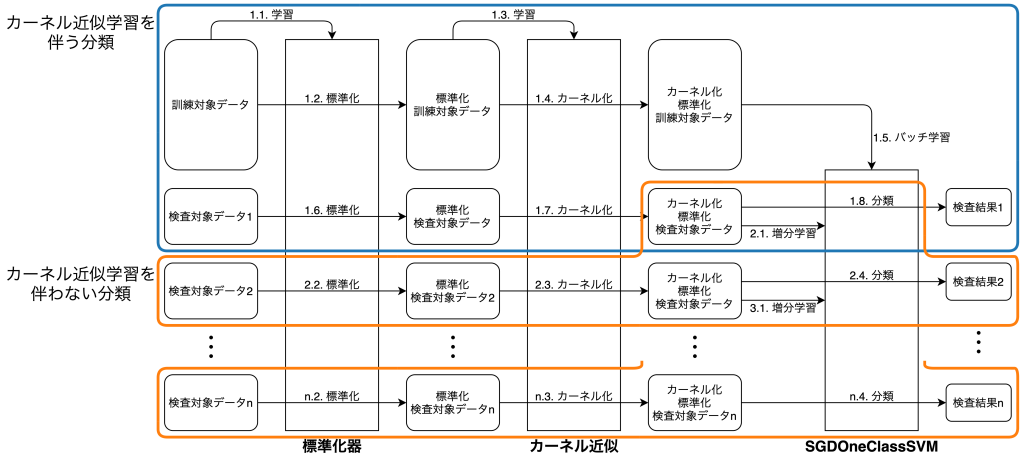

図 3: 全体像

a.8. SGDOneClassSVM を用いてカーネル化標準化検査対象データを分類

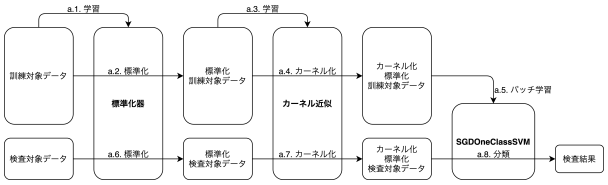

図 4: カーネル近似学習を伴う分類

3.3 カーネル近似学習を伴わない分類

本節では, 標準化器とカーネル近似の学習を行わずに直前のカーネル化標準化検査対象データを用いて SGDOneClassSVM の増分学習を行った上で, 新しい検査対象データの分類を行う. この際に, 標準化器とカーネル近似には 3.2 節で学習したモデルをそのまま使用する. 図 5 のように以下のステップで検査対象データの分類を行う.

b.1. カーネル化標準化検査対象データを用いて SGDOneClassSVM を増分学習

b.2. 標準化器を用いて新しい検査対象データを標準化

b.3. カーネル近似を用いて新しい標準化検査対象データをカーネル化

b.4. SGDOneClassSVM を用いて新しいカーネル化標準化検査対象データを分類

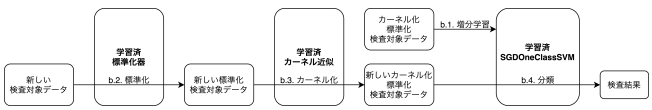

図 5: カーネル近似学習を伴わない分類

3.4 学習頻度

学習頻度は整数 n を用いて, $1/n$ で設定される. 図 6 のように 3.2 節のカーネル近似学習を伴う分類を 1 回行った後に, 3.3 節のカーネル近似学習を伴わない分類を $n-1$ 回連続で行う. $n-1$ 回カーネル近似学習を伴わない分類を行った後は再びカーネル近似学習を伴う分類を行う.

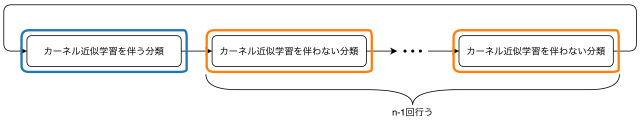

図 6: 2 種類の分類の手順

3.5 想定される構成

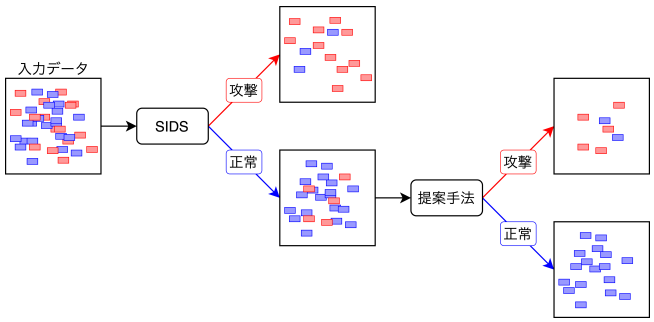

図 7: 想定される構成

実際のアプリケーションでは, 図 7 のように提案手法は従来のシグネチャベースの侵入検知器 (SIDS) と併用することを想定している. 入力データを SIDS を用いて分類し, 正常と分類されたデータポイントをさらに提案手法を用いて分類する. SIDS は過去に類似する攻撃が存在する既知の攻撃の分類は高精度でできる一方で, 過去の攻撃とはかけ離れた未知の攻撃の分類は難しい. このため, SIDS が正常と分類したデータポイントの中には未知の攻撃が混在していると考えられる. そこで, 入力データを SIDS を用いて分類し正常と分類されたデータポイントをさらに提案手法を用いて分類する. これにより, 既知の攻撃と未知の攻撃の両方を高精度で検知することが期待できる.

4 実　　験

4.1 実験設定

本節では, 増分学習と組合せたカーネル近似学習の削減が性能に与える影響を明らかにするための実験の設定について述べる. 具体的には, 以下の Research Question に回答するための実験の設定について述べる.

- RQ1 増分学習と組み合わせたカーネル近似学習の削減が分類精度にどのように影響しているか.
- RQ2 増分学習と組み合わせたカーネル近似学習の削減が学習時間にどのように影響しているか.

4.1.1 節では実験に用いたデータセットについて, 4.1.2 節では実験の手順について記述する. 4.1.3 節では RQ1 において議論する性能の評価尺度について説明する. また, 本実験は, Intel Xeon E-2278G, メモリ 64GB の環境上で行った. 実装には sklearn v1.0.2 を用いた. OneClassSVM には svm.OneClassSVM, SGDOneClassSVM には linear_model.SGDOneClassSVM, カーネル近似には kernel_approximation.Nystroem, 標準化には preprocessing.StandardScaler を用いた.

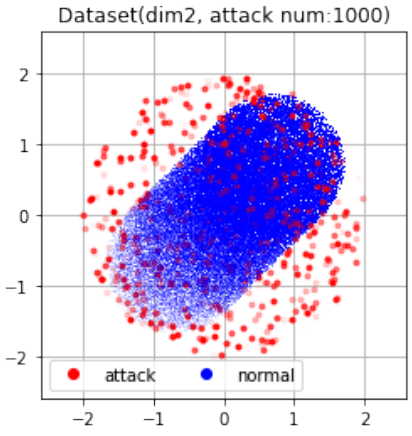

図 8: 人工データセット (2 次元)

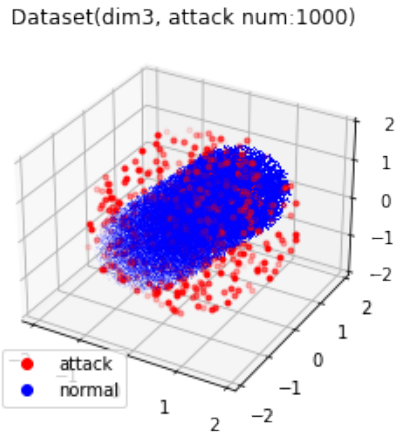

図 9: 人工データセット (3 次元)

4.1.1 人工データセット

実験対象として人工データセットも用いる. 人工データセットは, 攻撃は半径 2 の超球内の領域から, 正常は移動する半径 1 の超球内の領域から一様にランダムに生成する全 $1,000,000$ 件のデータポイントにより構成されている. 人工データセットは次元数が $2,3,4,5$, 攻撃の件数は $1,000$ 件 と $10,000$ 件 のデータセットを用いた. 図 8 と図 9 は, 次元数が $2,3$, 攻撃件数が $1,000$ 件のデータセットの $1 \sim 500,000$ 件を図示した図であり, より後のデータポイント程濃くプロットされている.

n 次元の人工データセットの全 M 件中の i 件目のデータポイントは, 半径 1 の超球内一様乱数のベクトルを $\mathbf{r}$ とすると, 正常のデータポイントをベクトル **normal** と攻撃のデータポイントを表すベクトル **attack** は以下のようになる.

$$\mathbf{normal} = \mathbf{r} + \frac{1}{\sqrt{n}} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \cos\left(2\pi \frac{i}{M}\right), \quad \mathbf{attack} = 2\mathbf{r}$$

4.1.2 実験方法

Research Question を評価するために OneClassSVM と提案手法で実験を行う. データセットの $1,001 \sim 1,000,000$ 件目のデータを検査対象データとして順に分類を行う. 提案手法では, 設定された学習頻度に応じて, 3.2 節のカーネル近似学習を伴う分類と 3.3 節のカーネル近似学習を伴わない分類を行う. カーネル近似学習を伴う分類では, 検査対象データから $1,000$ 件遡り訓練対象データとしてカーネル近似の学習・SGDOneClassSVM のバッチ学習を行い, 検査対象データの分類を行う. カーネル近似学習を伴わない分類では, 直前の検査対象データに用いたモデルに対し, 直前の検査対象データを訓練対象データの増分として SGDOneClassSVM の増分学習を行い, 新しい検査対象データの分類を行う. また, 提案手法と比較するために OneClassSVM では設定された学習頻度に応じてバッチ学習を行う.

4.1.3 評価尺度

本研究では図 10 に定義する *Precision* と *Recall* の 2 つの尺度を使用する. 本研究は 2 値決定問題であり, OneClassSVM や SGDOneClassSVM はデータを攻撃 (positive) または正常 (negative) のいずれかとして分類する. この分類結果は混同行列を用いて表現され, 混同行列の 4 つのカテゴリ, (1) TP: 正しい攻撃という分類; (2) FP: 誤った攻撃という分類 (正しくは正常である); (3) TN: 正しい正常という分類; (4) FN: 誤った正常という分類 (正しくは攻撃である) のいずれかに属すことになる [16]. また, 混同行列を用いて, 評価指標 *Precision* と *Recall* が図 10 のように定義される.

	actual positive	actual negative
predicted positive	TP	FP
predicted negative	FN	TN

(a) Confusion Matrix

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{True Positive Rate} = \frac{TP}{TP+FN}$$

$$\text{False Positive Rate} = \frac{FP}{FP+TN}$$

(b) Definitions of metrics

図 10: 混同行列と評価指標 [16, Figure 2]

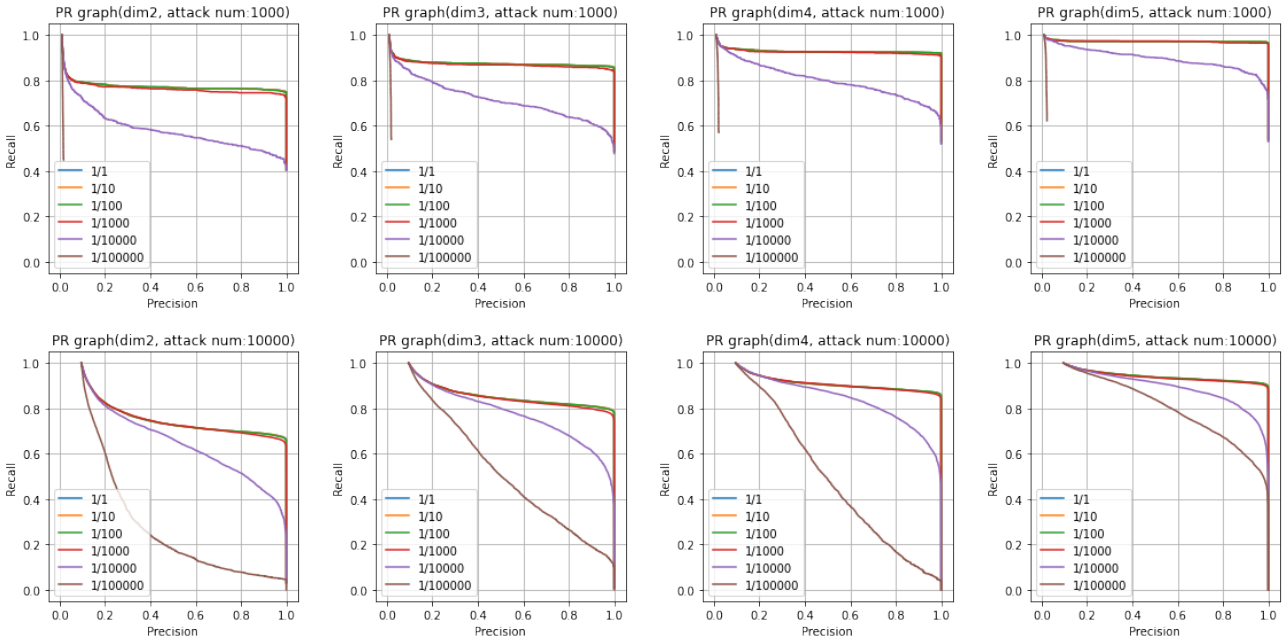

図 11: PR 曲線 (OneClassSVM)

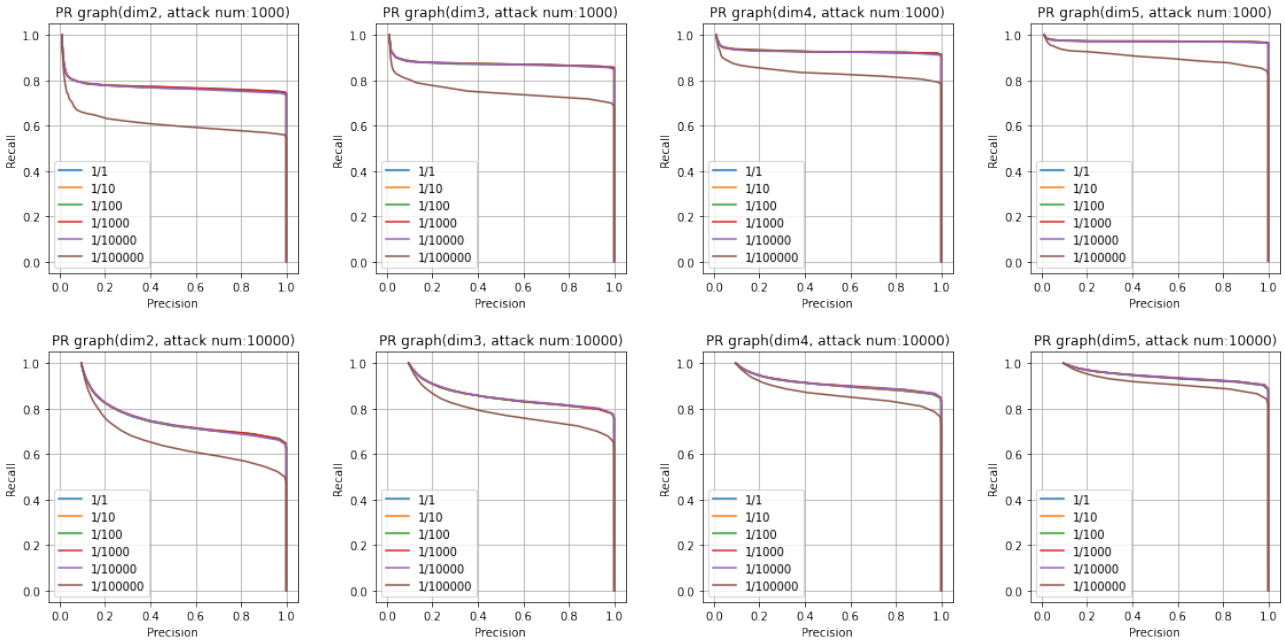

図 12: PR 曲線 (提案手法)

5 結　　果

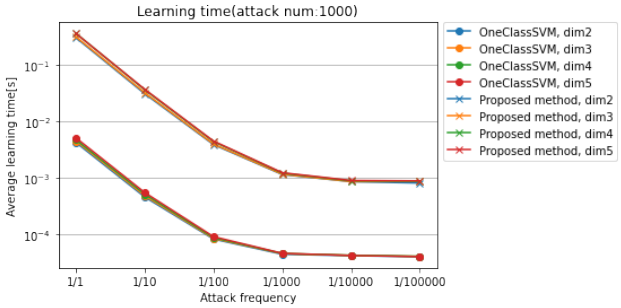

図 13: 学習時間 (攻撃件数 1,000)

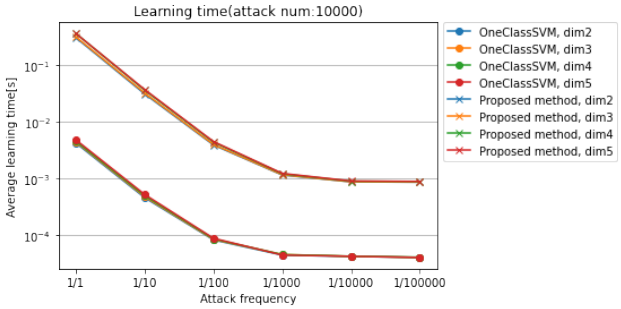

図 14: 学習時間 (攻撃件数 10,000)

4.1.1 節の人工データセットを用いて実験をした結果は図 11 と図 12、図 13 と図 14 の通りである。図 11 と図 12 は、$Precision - Recall$ 曲線となっており、SVM の超平面からの距離をもとに攻撃か否かを分類する閾値を変化させた際のさまざまな $Precision$ と $Recall$ をプロットしている。上段 4 つの図は攻撃件数 $1,000$ のデータセットを用いた場合の結果、下段 4 つの図は攻撃件数 $10,000$ のデータセットを用いた場合の結果であり, また, 左から順に次元数が $2, 3, 4, 5$ のデータセットを用いた場合の結果である. 図 13 と図 14 は, 学習頻度に対する平均学習時間を図示している.

5.1 カーネル近似と SGDOneClassSVM の分類精度への影響

図 11 と図 12 を比較すると, 学習頻度が $1/10,000$ より高い場合は, 提案手法は OneClassSVM とほぼ同じ精度である. このことから, カーネル近似と SGDOneClassSVM の組み合わせは OneClassSVM と同様の働きをしていることがわかる.

5.2 次元数の違いの分類精度への影響

図 15 のように図 11 と図 12 のどちらの上段 4 つや下段 4 つの同攻撃件数の結果の図を比較しても, 次元数が上がれば上がるほど精度が向上していることがわかる. これは, 攻撃が正常付近に生成される確率が下がるためと考えられる. 人工データセットの次元数が上がることで, 正常が生成される領域の体積に対する攻撃が生成される領域の体積の比が大きくなる. これにより, 攻撃が正常付近に分布される確率が下がり, 分類困難なデータポイントが減少したため, 分類が容易になり分類精度が向上していると考えられる.

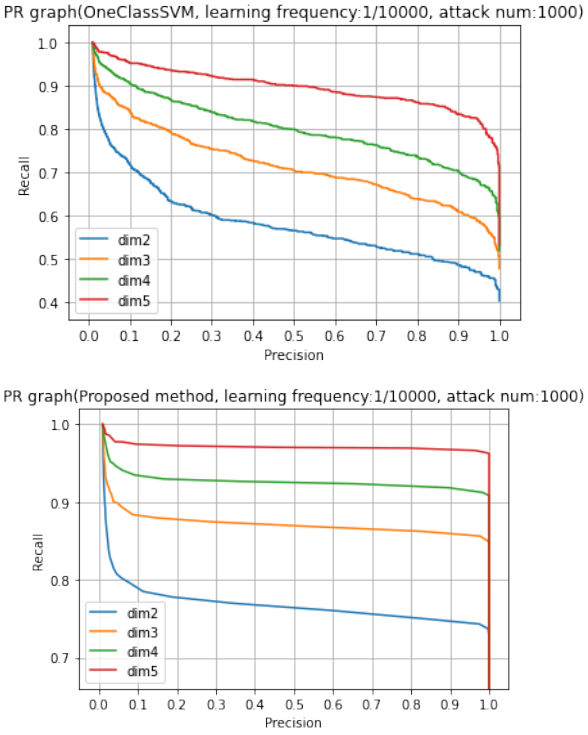

図 15: OneClassSVM と提案手法の異なる次元数のデータセットに対する分類精度の比較

5.3 攻撃の割合の違いの分類精度への影響

図 11 と図 12, どちらにおいても同次元数の結果において, $Precision$ が 0.8 以上の部分を比較すると, 攻撃件数が $10,000$ 件の方が $Recall$ が低いことがわかる. 攻撃件数が増えデータセットにおける攻撃の割合が増加したことにより, 攻撃が正常付近に生成される確率が上がった. これにより, 5.2 節と同様に分類困難なデータポイントが増加したため, 分類が困難になり $Recall$ が低下していると考えられる.

5.4 提案手法の分類精度への影響

図 16 の様に図 11 と図 12 の同攻撃件数同次元数のデータセットにおいて OneClassSVM と提案手法を比較すると, 学習頻度が $1/10,000 \sim 1/100,000$ と低い場合では提案手法の方が精度が高いことがわかる. 学習頻度が $1/10,000$ のとき, バッチ学習しか行えない OneClassSVM では最悪の場合, $10,000 \sim 10,999$ 件前のデータポイントを用いてバッチ学習を行ったモデルを用いて分類を行う. これにより, 今回のデータセットは, 正常が生成される領域が移動しているため, ある正常データポイントが生成される領域と $10,000 \sim 10,999$ 件前の正常データポイントが生成される領域があまり被らないため, OneClassSVM の分類精度が低下していると考えられる. 一方で, 提案手法では, バッチ学習は OneClassSVM と同様, 最悪の場合, $10,000 \sim 10,999$ 件前のデータポイントを用いた学習となることがあるが, $10,000$ 回に 1 度のバッチ学習に加えて増分学習を毎回行っているため, この増分学習が正常が移動する領域から生成されるデータセットによく働き, 提案手法のモデルが最新の正常の状態を OneClassSVM と比較してより反映しているためと考えられる.

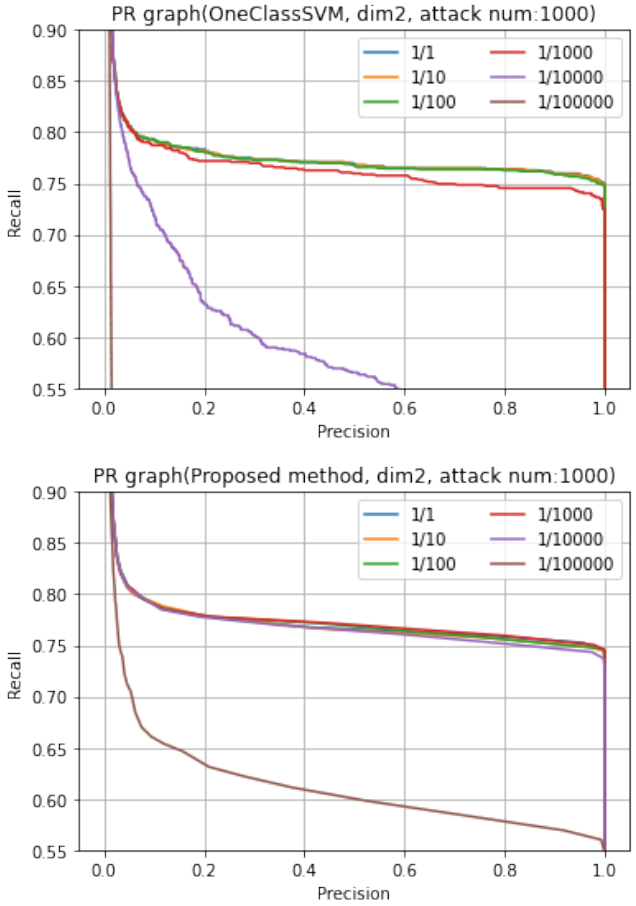

図 16: OneClassSVM と提案手法の比較

5.5 カーネル近似学習削減の分類精度への影響

図 12 より, 提案手法では学習頻度を低下させても精度が低下せず保たれていることがわかる. これは, カーネル近似学習の頻度を落としたことで分類精度の低下が予想されていたものの, カーネル近似学習の削減に増分学習を組み合わせたことにより, 毎回行う増分学習が分類精度の低下を軽減していると考えられる.

RQ1 への回答: 増分学習と組み合わせたカーネル近似学習の削減は学習頻度の低下に伴う分類精度の低下を軽減する.

5.6 提案手法の学習時間への影響

表 1: OneClassSVM の学習時間 [ms]

学習頻度	1/1	1/10	1/100	1/1,000	1/10,000
バッチ学習時	4.209	4.186	4.194	4.237	4.324
非バッチ学習時	–	0.039	0.040	0.039	0.041

図 13 と図 14, 加えて表 1 と表 2 を比較する. 表 1 と表 2 は, OneClassSVM と提案手法において, 次元数 2 攻撃件数 1,000 のデータセットに対するバッチ学習時と非バッチ学習時の平均学習時間を表にしたものである.

表 2: 提案手法の学習時間 [ms]

学習頻度	1/1	1/10	1/100	1/1,000	1/10,000
バッチ学習時	304.7	300.1	301.0	303.0	311.5
非バッチ学習時	–	0.965	0.859	0.834	0.819

図 13 と図 14, また表 1 と表 2 のどちらにおいても OneClassSVM と提案手法で比較すると, 学習頻度が 1/1 の場合では提案手法が OneClassSVM より約 100 倍学習に時間がかかっている. これは, カーネル近似の学習が OneClassSVM の学習と比べて時間がかかっているためと考えられる.

図 13 と図 14 の OneClassSVM と提案手法のどちらにおいても, 学習頻度が 1/1 ～ 1/100 と高い場合では, バッチ学習の頻度が 1/10 倍になるにつれて, 学習時間も約 1/10 倍していることがわかる. 一方で, 学習頻度が 1/1,000 ～ 1/100,000 と低い場合では, バッチ学習の頻度が 1/10 倍になるにつれて, 学習時間は 1/10 倍程は小さくなっていないこともわかる. これは, 表 1 と表 2 からわかるように, 非バッチ学習時の学習時間がバッチ学習時の学習時間に比べて小さいため, 学習頻度が高いときは非バッチ学習時の学習時間の影響が小さく, 学習頻度と学習時間がほぼ比例しており, 学習頻度が低いときは, 非バッチ学習時の学習時間の影響が比較的大きく, 学習時間は学習頻度と比例する程は小さくならないと考えられる.

また, 毎回バッチ学習を行う学習頻度 1/1 の OneClassSVM と比較すると, 提案手法は学習頻度 1/10,000 ～ 1/100,000 では, 学習時間が約 5 倍高速化していることがわかる. これは, カーネル近似の学習による学習時間の延長と学習頻度の低下による学習時間の短縮が合わさった結果, 約 5 倍高速化したと考えられる.

RQ2 への回答: 増分学習と組み合わせたカーネル近似学習の削減は学習頻度に比例するように学習時間を短縮させ, 毎回バッチ学習を行う学習頻度 1/1 の OneClassSVM と比較して, 学習頻度 1/100,000 の提案手法は約 5 倍高速化することを明らかにした.

5.7 カーネル近似学習削減の学習時間への影響

図 13 や図 14 の提案手法の学習時間より, カーネル近似学習を含んだ SGDOneClassSVM のバッチ学習を削減し, 代わりに SGDOneClassSVM の増分学習を行うことは学習時間を短縮させていることがわかる. また, 表 2 より, カーネル近似学習を含んだ SGDOneClassSVM のバッチ学習の学習時間は約 300ms であり, SGDOneClassSVM の増分学習の学習時間は約 0.9ms であるため, カーネル近似学習を含んだ SGDOneClassSVM のバッチ学習の学習時間が SGDOneClassSVM の増分学習の学習時間に比べて大きいことから, カーネル近似学習削減は学習時間を短縮させるとわかる.

6 結　　論

6.1 ま と め

近年, サイバー攻撃は増加し進化している. 進化し続けるサイバー攻撃のうち, 未知の攻撃は過去の攻撃とはかけ離れたものであり, ネットワークログには攻撃か正常かのラベルが存在しないことから未知の攻撃の検知には教師なし学習が用いられている. 教師なし学習手法の一つである OneClassSVM は, 精度は良い一方でバッチ学習しかできないため学習時間が長いという課題があった. そこで, 本研究ではカーネル近似学習の頻度の低下と SGDOneClassSVM の増分学習を組み合わせる手法を提案する.

本研究では, カーネル近似と SGDOneClassSVM を組み合わせて OneClassSVM と同等の働きをする様にした上で, カーネル近似学習の頻度を落とし, SGDOneClassSVM で増分学習する手法を実装した. 増分学習と組合せたカーネル近似学習の削減が性能に与える影響を明らかにするために, 評価用人工データセットを作成し用いて実験した.

実験結果より, 増分学習と組合せたカーネル近似学習の削減は学習頻度の低下に伴う分類精度の低下を軽減し, 学習頻度に比例するように学習時間を短縮させ, 毎回バッチ学習を行う OneClassSVM と比較して約 5 倍高速化させることを明らかにした.

6.2 今後の課題

本研究では, 次元数 $2, 3, 4, 5$ の人工データセットを作成し用いて実験を行った. この人工データセットは攻撃のデータポイントは半径 2 は超球内の領域から, 正常のデータポイントは移動する半径 1 の超球内の領域から生成しており, 正常のデータポイントは全ての要素の生成される領域が段々と変化するようになっている. しかし, サイバーセキュリティの実際のアプリケーションでは, 得られるデータの次元数はさらに高い. また, 全ての要素の分布が段々と変化するようなことはない. 次元数が高いことにより, 分類精度の低下や学習時間の延長が予想されるが, 分布の変化する要素が少ないことから分類精度が向上するとも予想される. 現実のデータセットを用いた際の増分学習と組み合せたカーネル近似学習の削減が性能に与える影響に関しては検討の余地がある.

また, カーネル近似に Nystroem 法を用いて RBF カーネルに近似したが, 他手法を用いて近似をしたり, 他カーネルに近似することで分類精度やカーネル近似の学習時間は変化することが予想され, 検討の余地がある.

本研究では OneClassSVM や SGDOneClassSVM のハイパーパラメータに対して最適化をしていない. ハイパーパラメータの最適化により分類精度や学習時間は変化することが予想され, 検討の余地がある.

文　　献

[1] Sumits, A.: The History and Future of Internet Traffic, `https://blogs.cisco.com/sp/the-history-and-future-of-internet-traffic`: (accessed: 2023-01-14).

[2] NICTER 観測レポート 2021, Technical report, 国立研究開発法人情報通信研究機構 サイバーセキュリティ研究所 サイバーセキュリティネクサス (2022).

[3] Bendovschi, A.: Cyber-Attacks – Trends, Patterns and Security Countermeasures, *Procedia Economics and Finance*, Vol. 28, pp. 24–31 (2015).

[4] Portnoy, L., Eskin, E. and Stolfo, S.: Intrusion Detection with Unlabeled Data Using Clustering (2001).

[5] Sarker, I. H., Kayes, A., Badsha, S., Alqahtani, H., Watters, P. and Ng, A.: Cybersecurity data science: an overview from machine learning perspective, *Journal of Big data*, Vol. 7, No. 1, pp. 1–29 (2020).

[6] Buczak, A. L. and Guven, E.: A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection, *IEEE Communications Surveys & Tutorials*, Vol. 18, No. 2, pp. 1153–1176 (2016).

[7] Noble, W. S.: What is a support vector machine?, *Nature biotechnology*, Vol. 24, No. 12, pp. 1565–1567 (2006).

[8] 栗田多喜夫：サポートベクターマシン入門, 産業技術総合研究所 脳神経情報研究部門, Vol. 2002, p. 1 (2002).

[9] Chawla, N. V., Japkowicz, N. and Kotcz, A.: Editorial: Special Issue on Learning from Imbalanced Data Sets, *SIGKDD Explor. Newsl.*, Vol. 6, No. 1, p. 1–6 (2004).

[10] Sun, Y., Wong, A. K. and Kamel, M. S.: Classification of imbalanced data: A review, *International journal of pattern recognition and artificial intelligence*, Vol. 23, No. 04, pp. 687–719 (2009).

[11] Krawczyk, B.: Learning from imbalanced data: open challenges and future directions, *Progress in Artificial Intelligence*, Vol. 5, No. 4, pp. 221–232 (2016).

[12] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J. and Williamson, R. C.: Estimating the Support of a High-Dimensional Distribution, *Neural Computation*, Vol. 13, No. 7, pp. 1443–1471 (2001).

[13] Yang, K., Kpotufe, S. and Feamster, N.: An Efficient One-Class SVM for Anomaly Detection in the Internet of Things, *arXiv preprint arXiv:2104.11146* (2021).

[14] Bottou, L.: Stochastic gradient descent tricks, in *Neural networks: Tricks of the trade*, pp. 421–436, Springer (2012).

[15] Lee, C.: Pegasos algorithm for one-class support vector machine, *IEICE TRANSACTIONS on Information and Systems*, Vol. 96, No. 5, pp. 1223–1226 (2013).

[16] Davis, J. and Goadrich, M.: The Relationship between Precision-Recall and ROC Curves, in *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, p. 233–240, New York, NY, USA (2006), Association for Computing Machinery.