SWIM研究会 2014年11月

機械学習を用いた侵入検知システムに関する学習手法の検討

沖縄高専
荻野 正

目次

1. はじめに
2. システム構成
3. 基礎測定
4. 学習手法の測定
5. まとめ

目次

1. はじめに
2. システム構成
3. 基礎測定
4. 学習手法の測定
5. まとめ

1. はじめに

サイバー攻撃

→ 手口の巧妙化

→ 標的型攻撃は年間1,000件以上

対応方法

1) Signature-based

2) Anomaly detection
外れ値検出

対応方法の比較

1) Signature-based

既存のデータベースと比較して検出

◯：DBにあれば、見逃すことはない

×：未知の攻撃は検出できない

2) Anomaly detection

正常状態から外れた状態を検出

×：検出エラーがある

◯：未知の攻撃を検出できる

標的型攻撃の検出には、未知の攻撃の検出が必要

目次

1. はじめに

2. システム構成

3. 基礎測定

4. 学習手法の測定

5. まとめ

2. システム概要

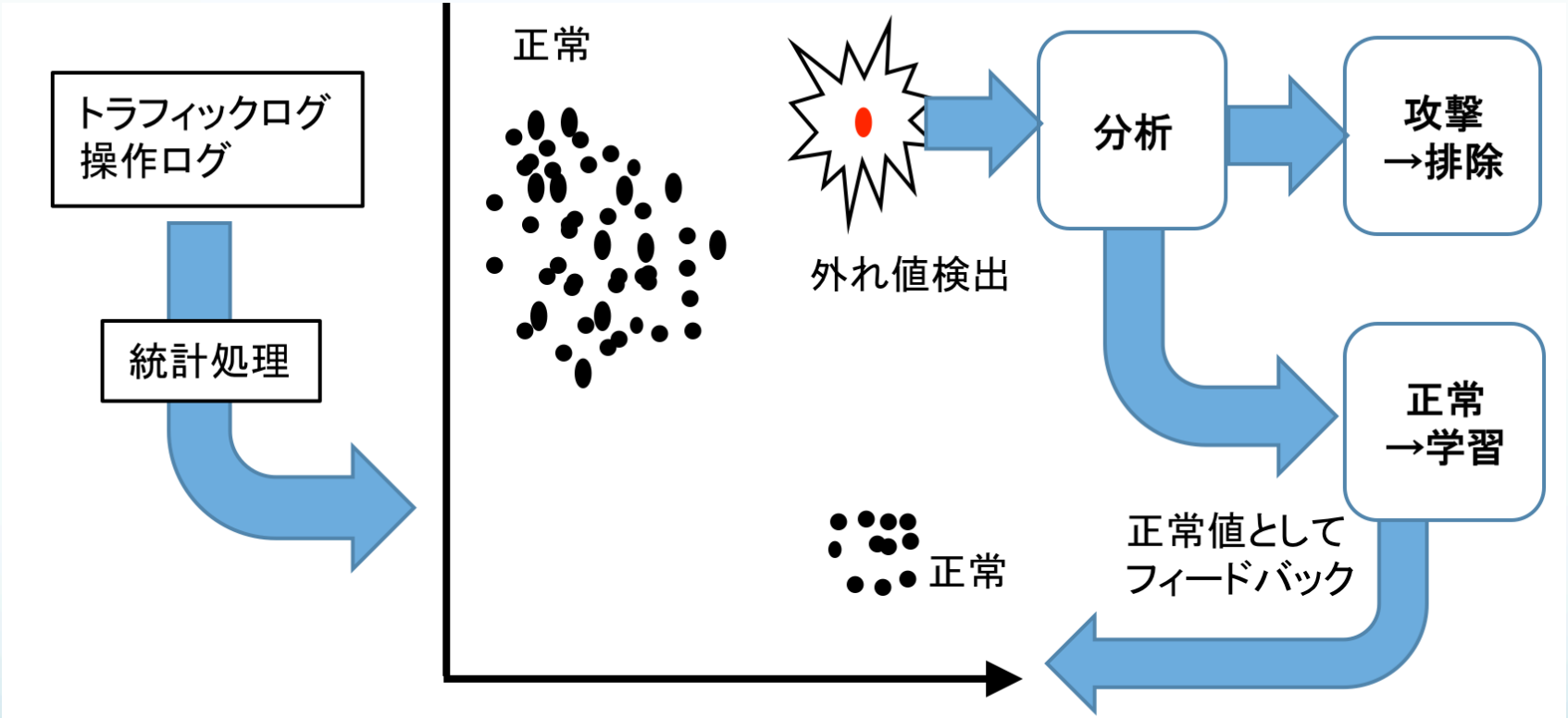
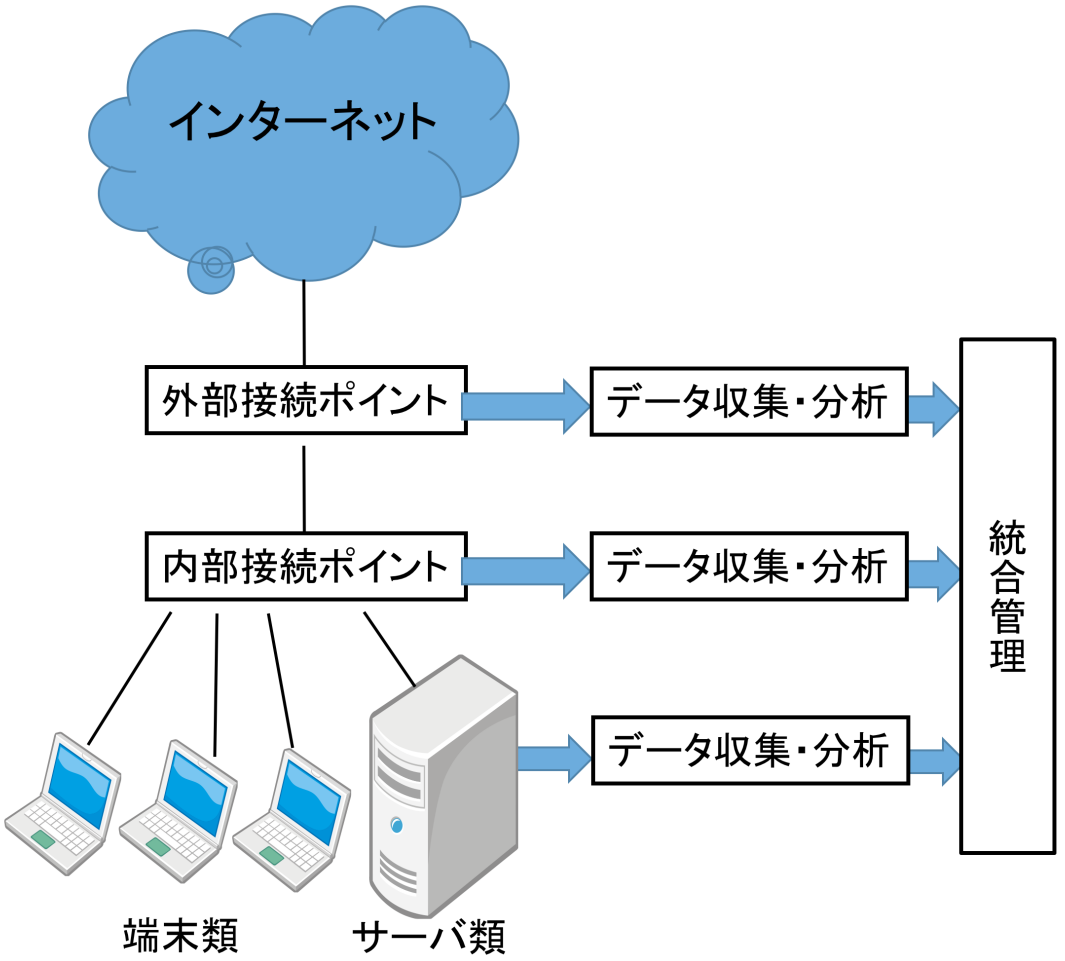


図1. システムの処理の概要

システム全体構成


インターネット
外部接続ポイント
データ収集・分析
内部接続ポイント
データ収集・分析
データ収集・分析
統合管理
端末類
サーバ類

処理プラットフォーム

- 大量のデータを処理する必要がある。
  → スケールアウト前提
- リアルタイムで検出したい。
- （LOFアルゴリズムを実装したい。）

→ 今回は Jubatus というプラットフォームを利用

検出手法：LOF 概要

- Markus M.Breunig, Hans-Peter Kriegel, Raymond T.Ng, Jörg Saner らによる(※)
- 最初から外れ値検出を目標としている
  （⟺ うまく分類できない例外）
- 1か0ではなく、外れ度を計算
- データ全体ではなく、近傍のデータから計算
- ネットワークの侵入検知アルゴリズムとして優れているという評価あり(*)

※ “LOF: Identifying Density-Based Local Outliers”, Proc. ACM SIGMOD 2000 Int. Conf. On Management of Data, 2000

* A.Lazarevie他, “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection,” 3rd SIAM Int. Conf. on Data Mining, 2003

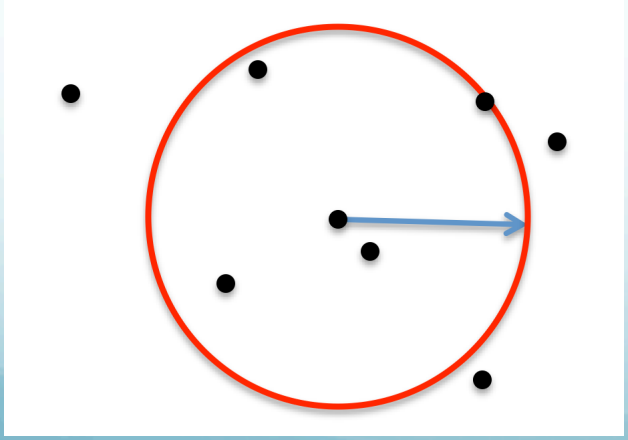
LOFの計算

（大雑把にまとめると）

lrd = 各点の近くのMinPts個の点との平均距離の逆数

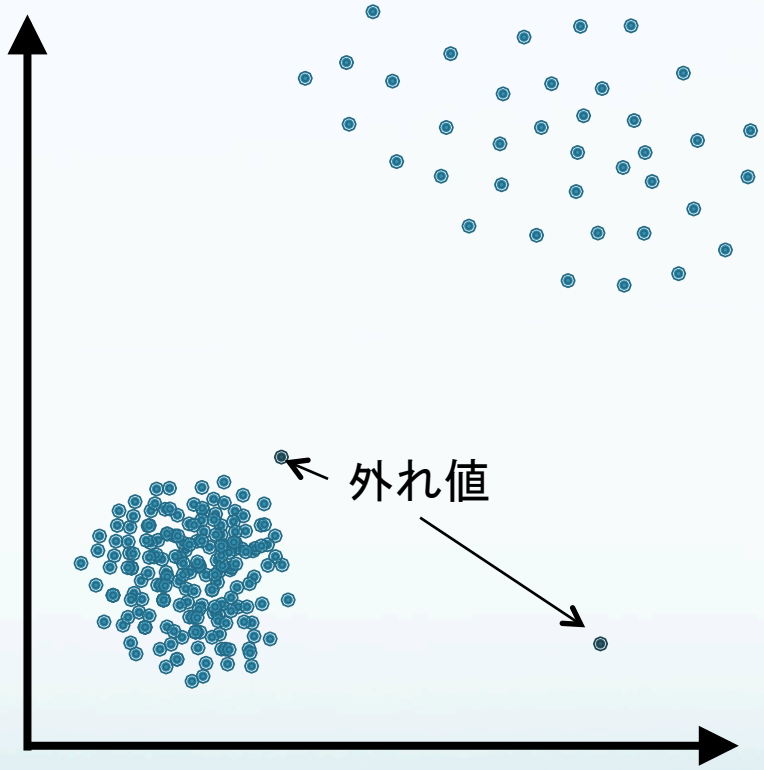
LOF = 近くのMinPts個の点のlrdとの比の平均

正常な点では、LOFはほぼ1、外れ値は1から外れる。



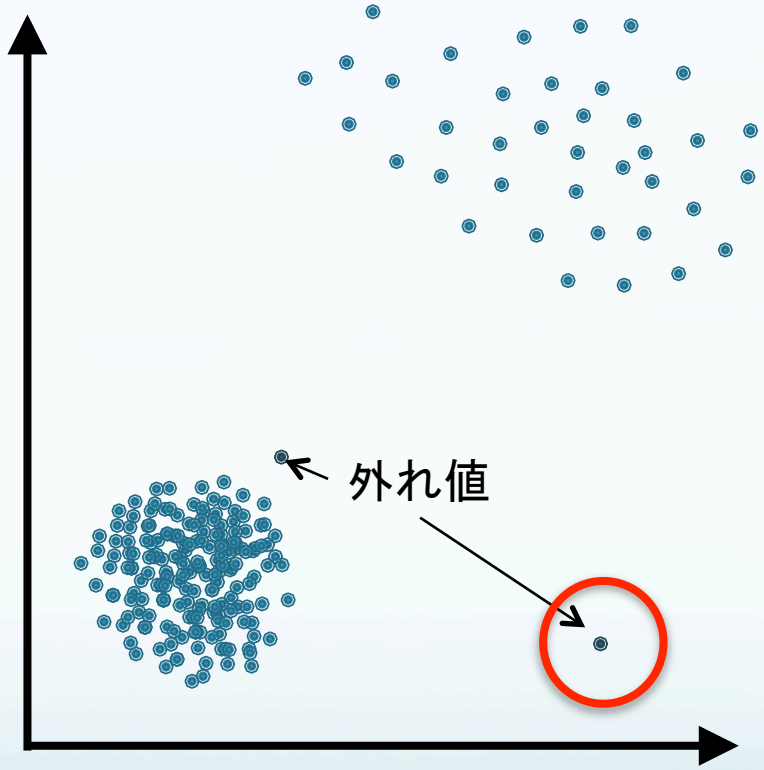
lrd : local reachability density
LOF : local outlier factor

LOFによる外れ値検出



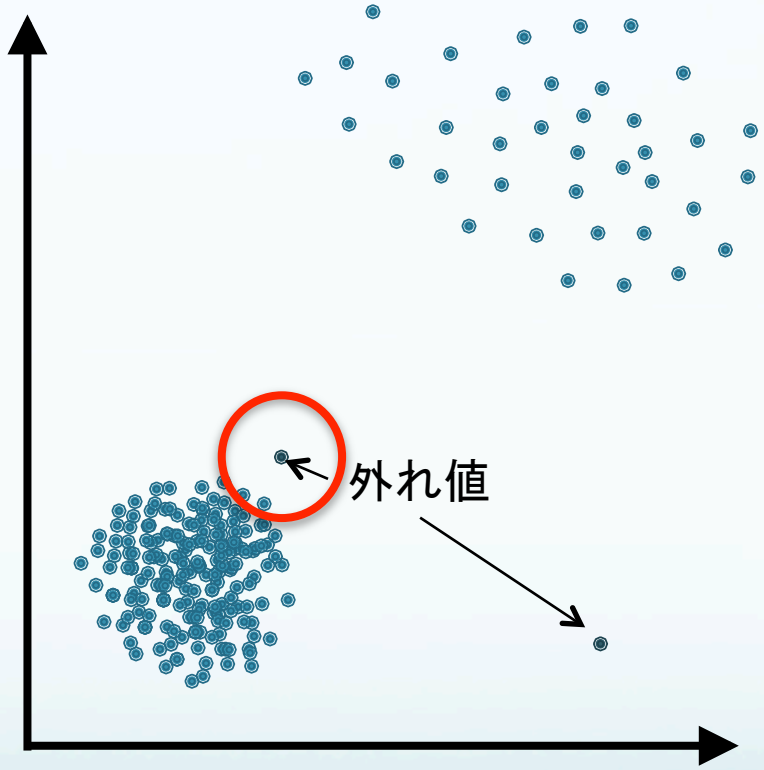
※ LOFのオリジナル論文より流用

LOFによる外れ値検出



※ LOFのオリジナル論文より流用

LOFによる外れ値検出



※ LOFのオリジナル論文より流用

Jubatus

- PFI社、NTT社が共同で開発した、オンライン機械学習向け分散処理フレームワーク
- オープンソース
- スケーラブル
- 様々な機械学習ライブラリを含む（LOF含む）
- リアルタイム

目次

1. はじめに

2. システム構成

3. 基礎測定

4. 学習手法の測定

5. まとめ

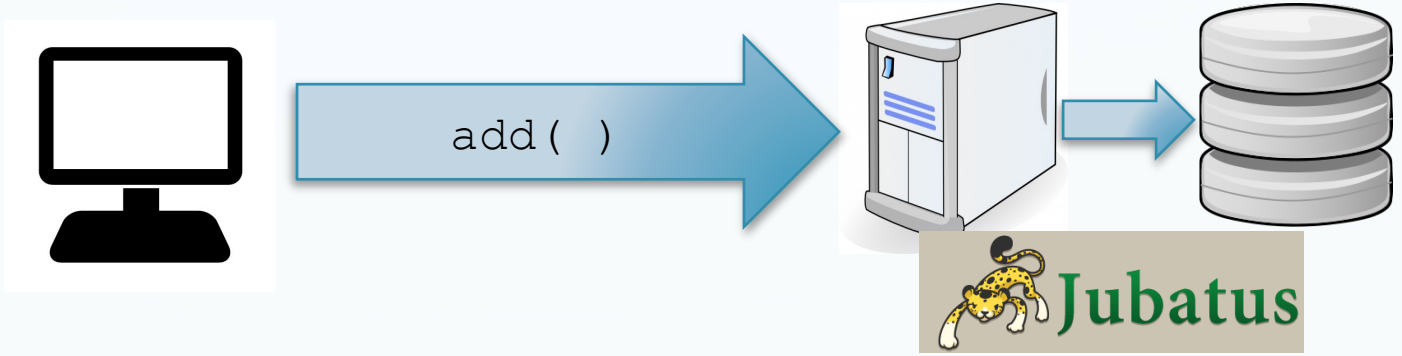
性能評価環境

- HW (Amazon Web Service)
  - CPU t2.micro (1CPU, 2.5GHz, Intel Xeon)
  - メモリ　1GiB
- SW
  - Ubuntu Server 14.04 LTS
  - M/W　Jubatus 0.6.4
- トラフィックデータ
  - KDD Cup 1999 データ

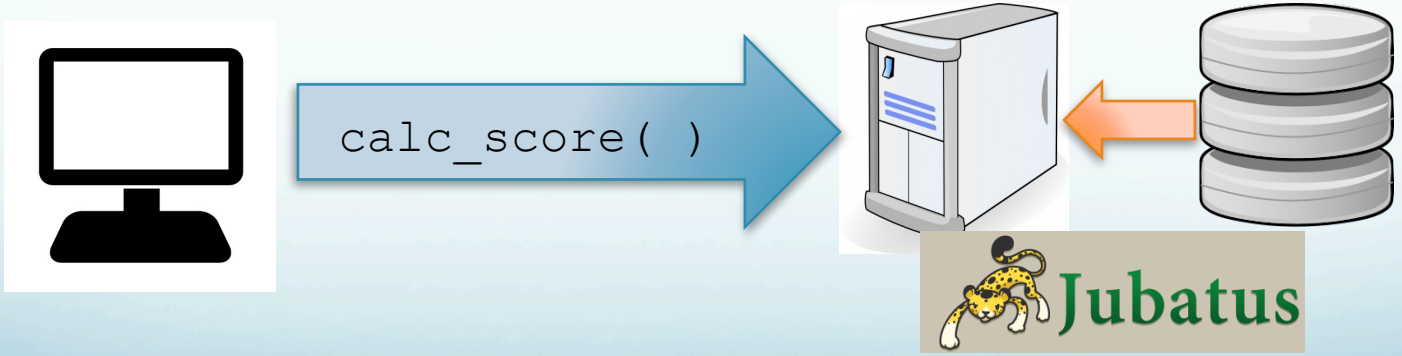
侵入検知評価用データ
軍の1週間のトラフィックを想定
ラベル付き
約500万件（評価は10%を使用）

性能評価内容(基礎評価)

1. 学習フェーズ



2. 評価フェーズ



学習したデータ数を変えて性能測定

学習方法(基本)

- データの先頭から正常データを学習。
- 正常データの数がある値を超えたら、それ以降は学習せずにスコア計算のみ

基礎評価結果(add)

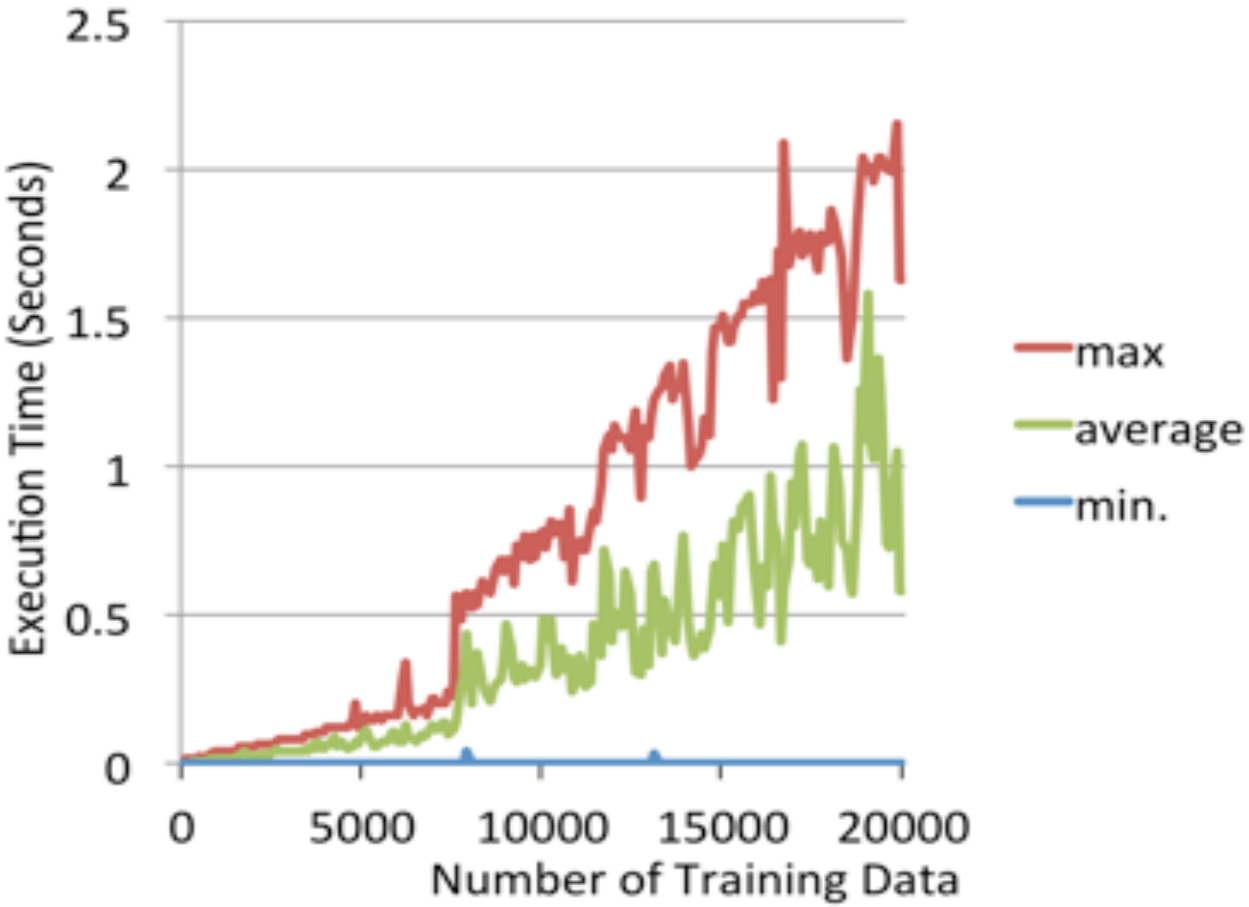


Fig. 5. Execution Time of add( ) API

基礎評価結果(calc_score)

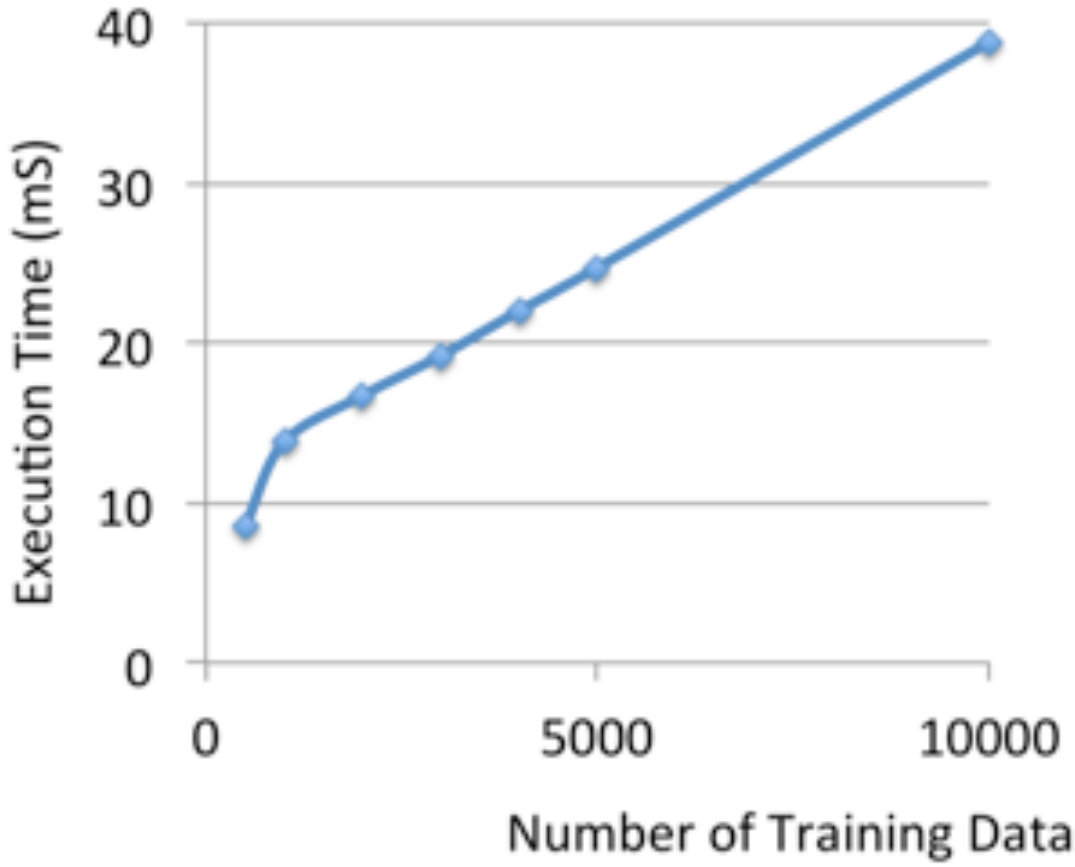


Fig.6. Execution Time of calc_score( ) API

全実行時間

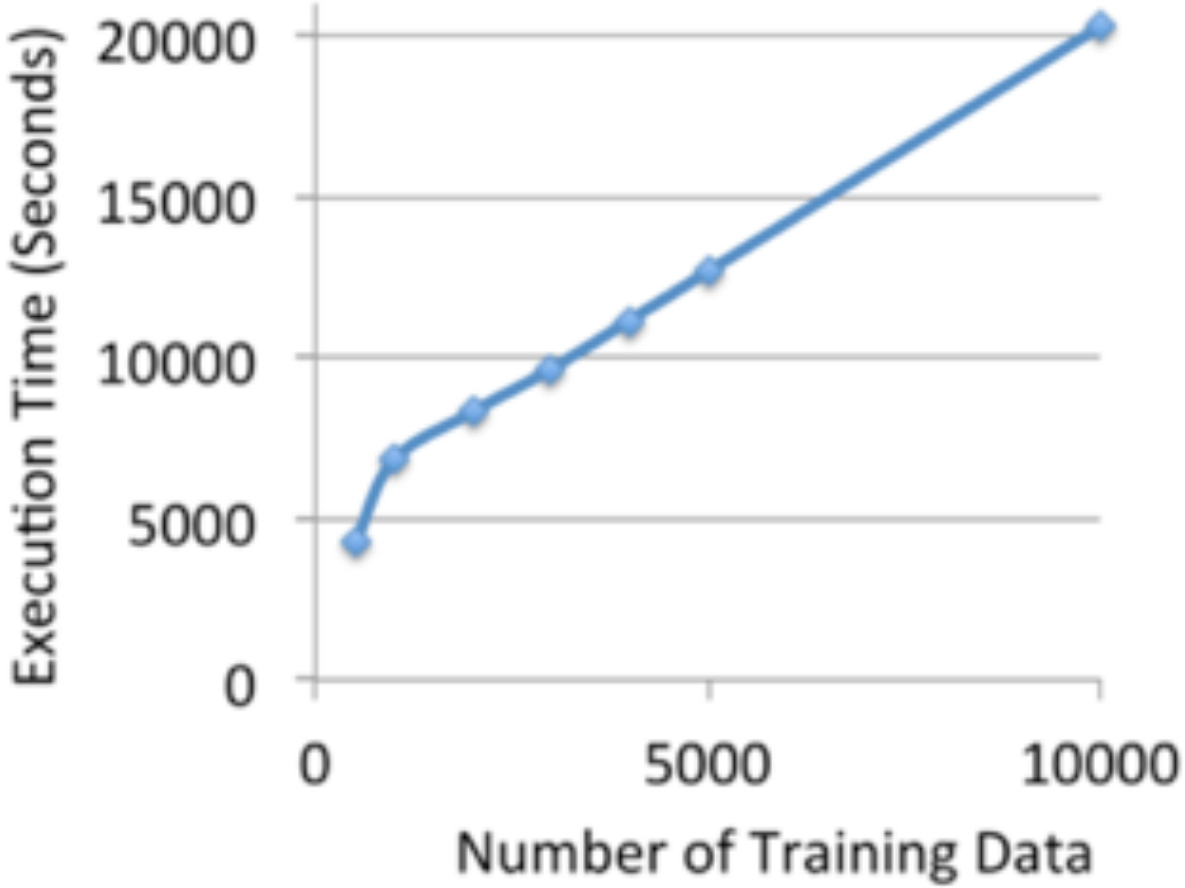
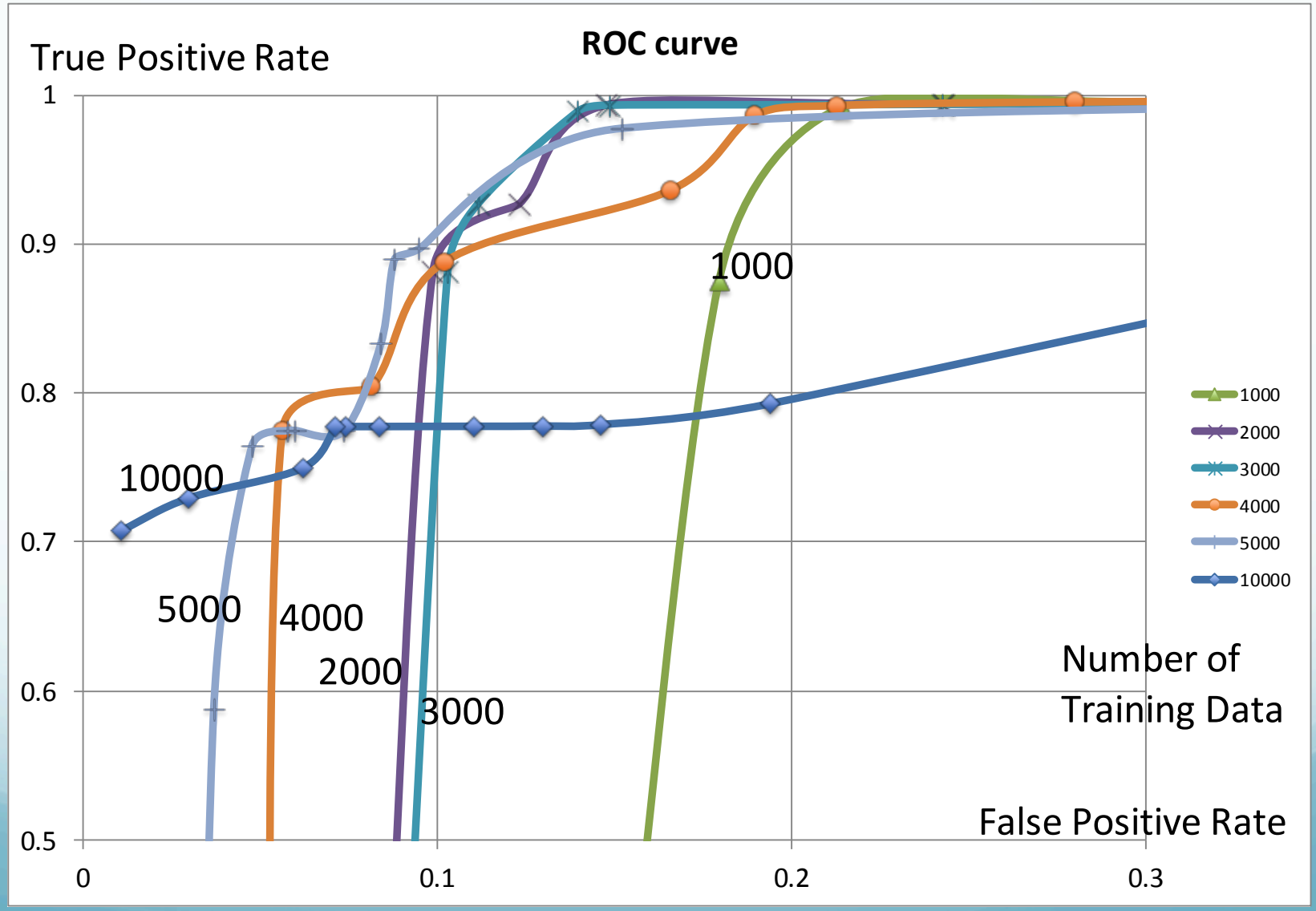


Fig.7 Total Execution time

精度評価


ROC curve
True Positive Rate
1
0.9
0.8
0.7
0.6
0.5
0
0.1
0.2
0.3
1000
10000
5000
4000
2000
3000
Number of Training Data
False Positive Rate
1000
2000
3000
4000
5000
10000

傾向

- 学習データ量が多いと遅くなる。
- 学習データ量が多いと精度が良くなる傾向がある。

- 学習データ量が少なくても精度が良くなる方法はないか？

目次

1. はじめに

2. システム構成

3. 基礎測定

4. 学習手法の測定

5. まとめ

新しい学習手法

1）Learn Gradually Method

- データの先頭から正常データを学習。
- 正常データの数がある値（今回は2000）を超えたら、それ以降は間引きをして学習する。

2）Learn Less Method

- スコア計算して、スレッショルドを超えたデータのみ学習（すでに学習してあるデータの近傍のデータは学習しない）

Learn Gradually Method

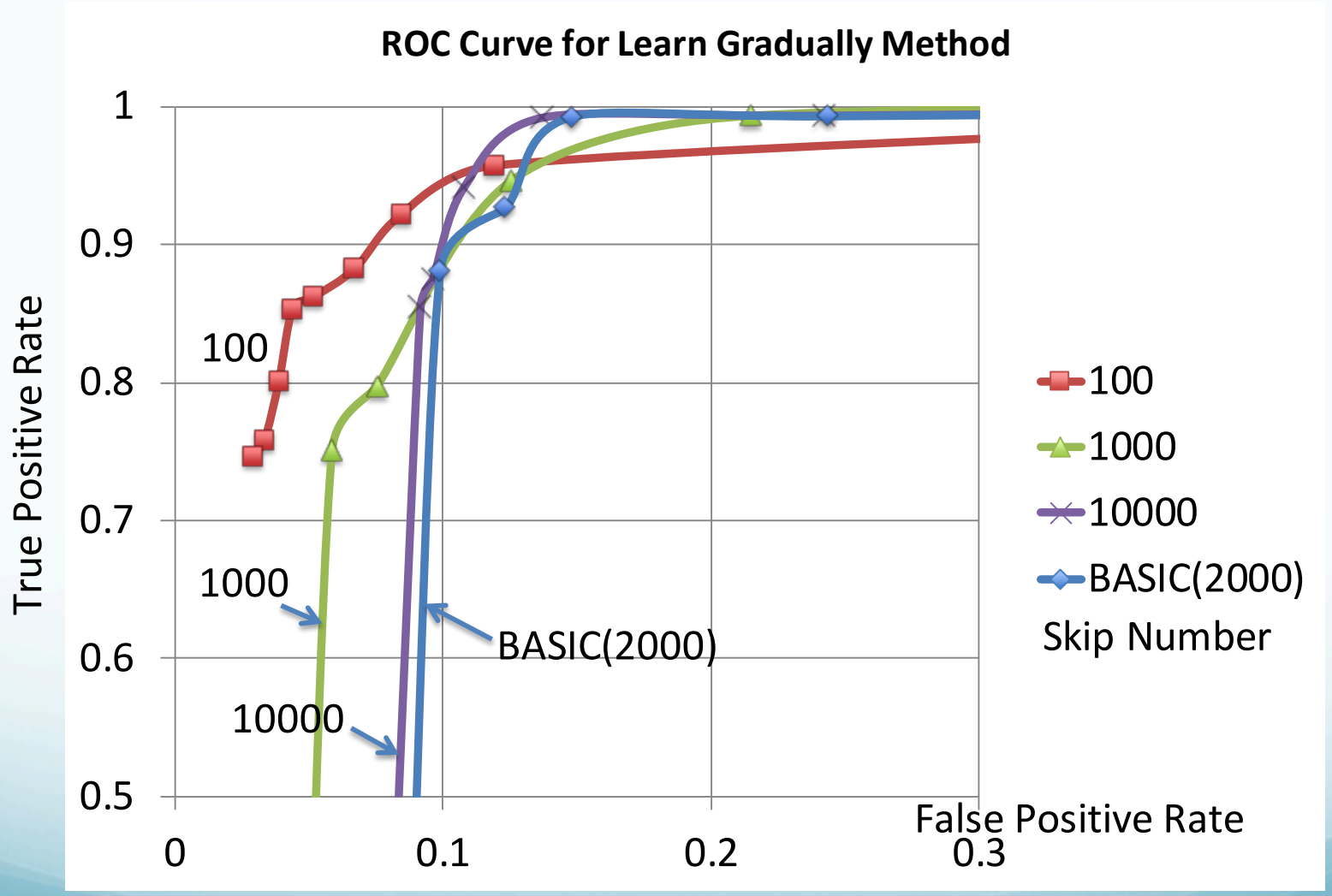
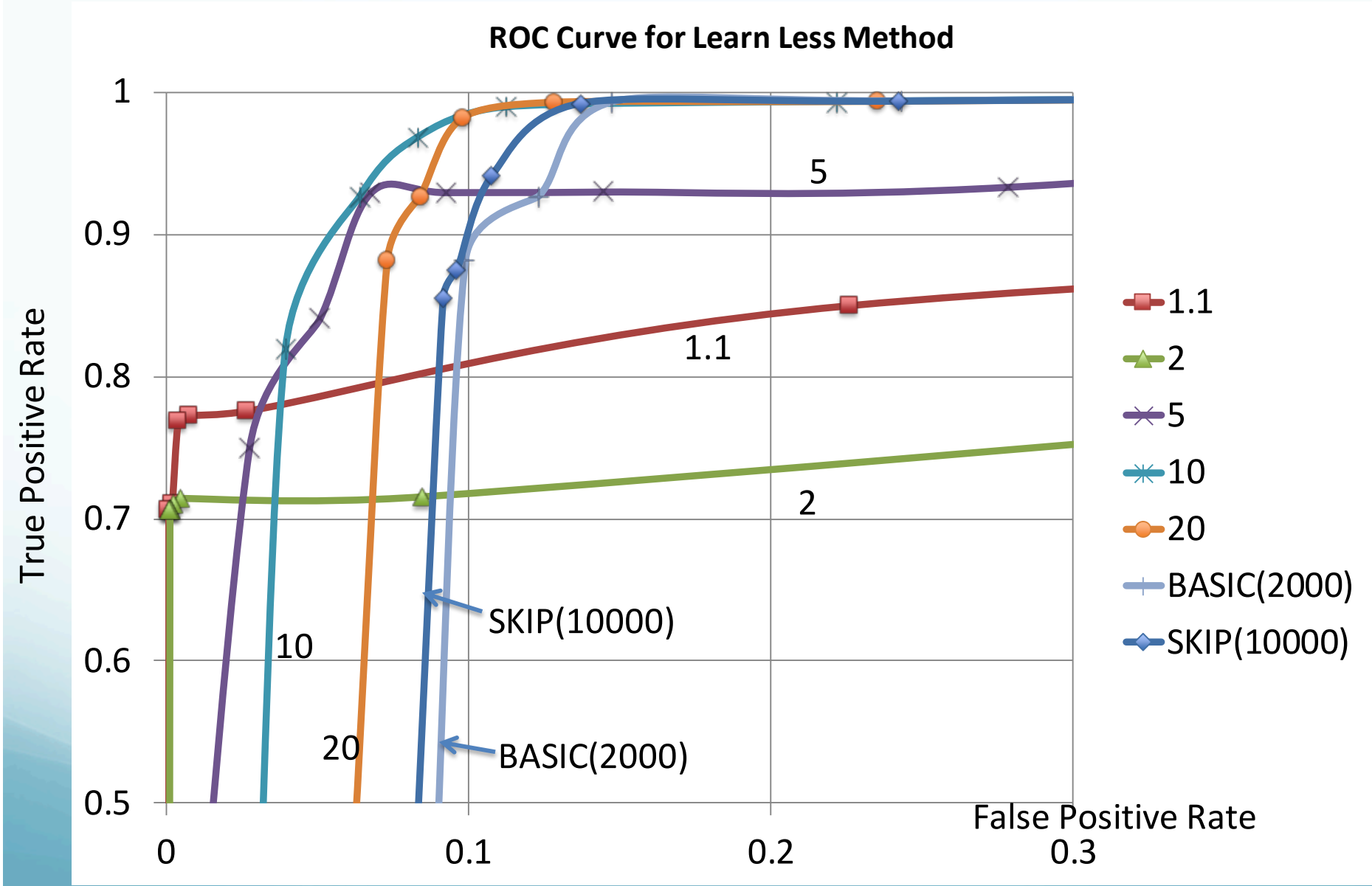


Fig.9 ROC Curve for Learn Gradually Method

Learn Less Method



結果

- Learn Less Methodにて、従来の手法よりも高い精度で攻撃データを検出できる。
- パラメータの微妙なチューニングが必要。

目次

1. はじめに

2. システム構成

3. 基礎測定

4. 学習手法の測定

5. まとめ

まとめ

- 今回の測定結果では、学習データが10000件の時、add()が数秒、calc_score()が数十m秒。

  → KDDCup1999のデータは 0.8秒/件
  add( )は性能不足だが、calc_score( )は性能十分

  → 15年前のデータなので、最新のデータでの評価が必要

  → スケールアウトの評価が必要

  → パラメータチューニングの自動化が必要

まとめ（続き）

- Learn Less Method で、精度向上を確認した。

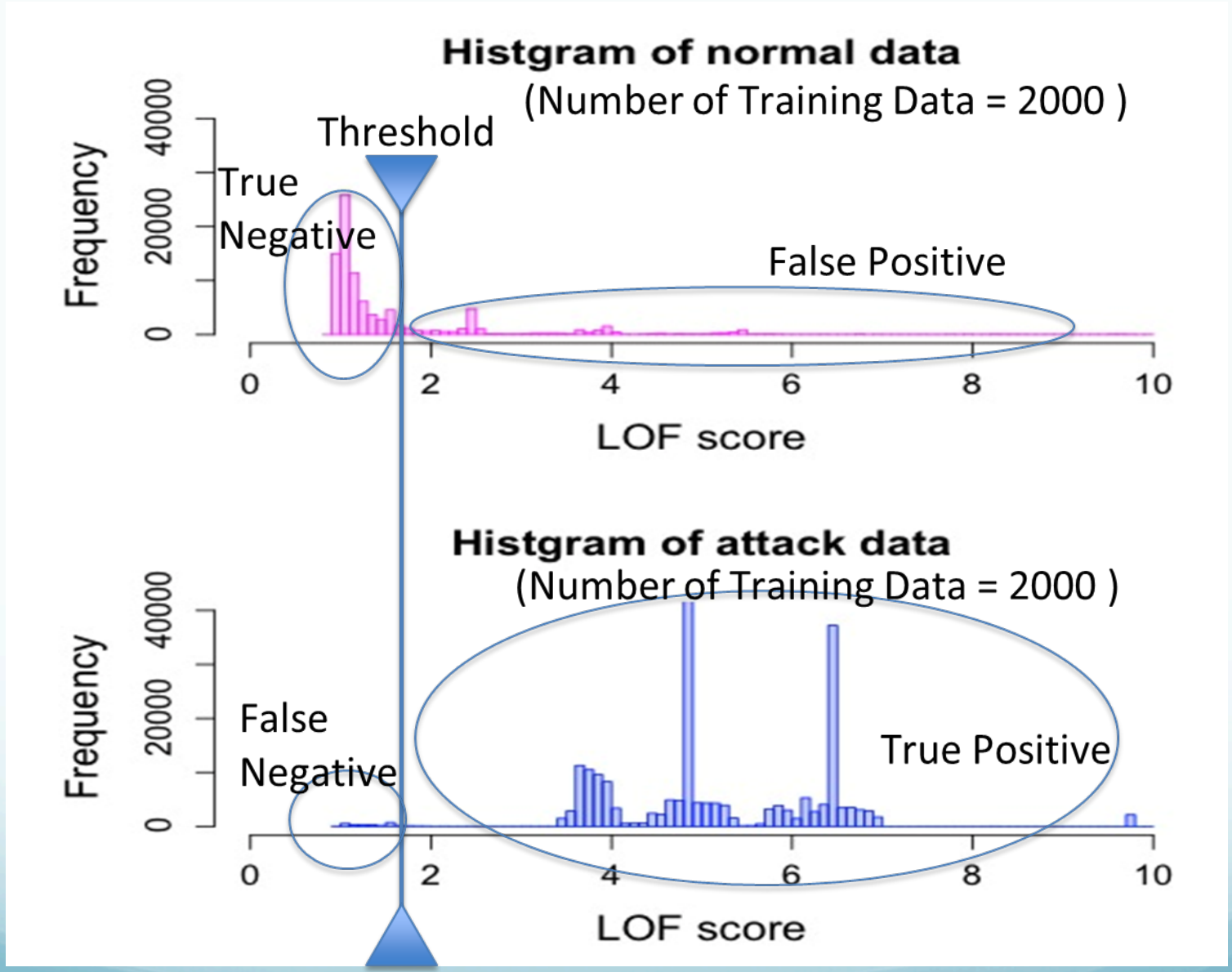
  → 学習データの総数を増やさない学習方法の検討が必要
  （＝ 過去に覚えたデータのうち、不要なデータを忘れる）

- 商用システムと比較すると、1桁くらい誤り検出率を下げる必要がある。

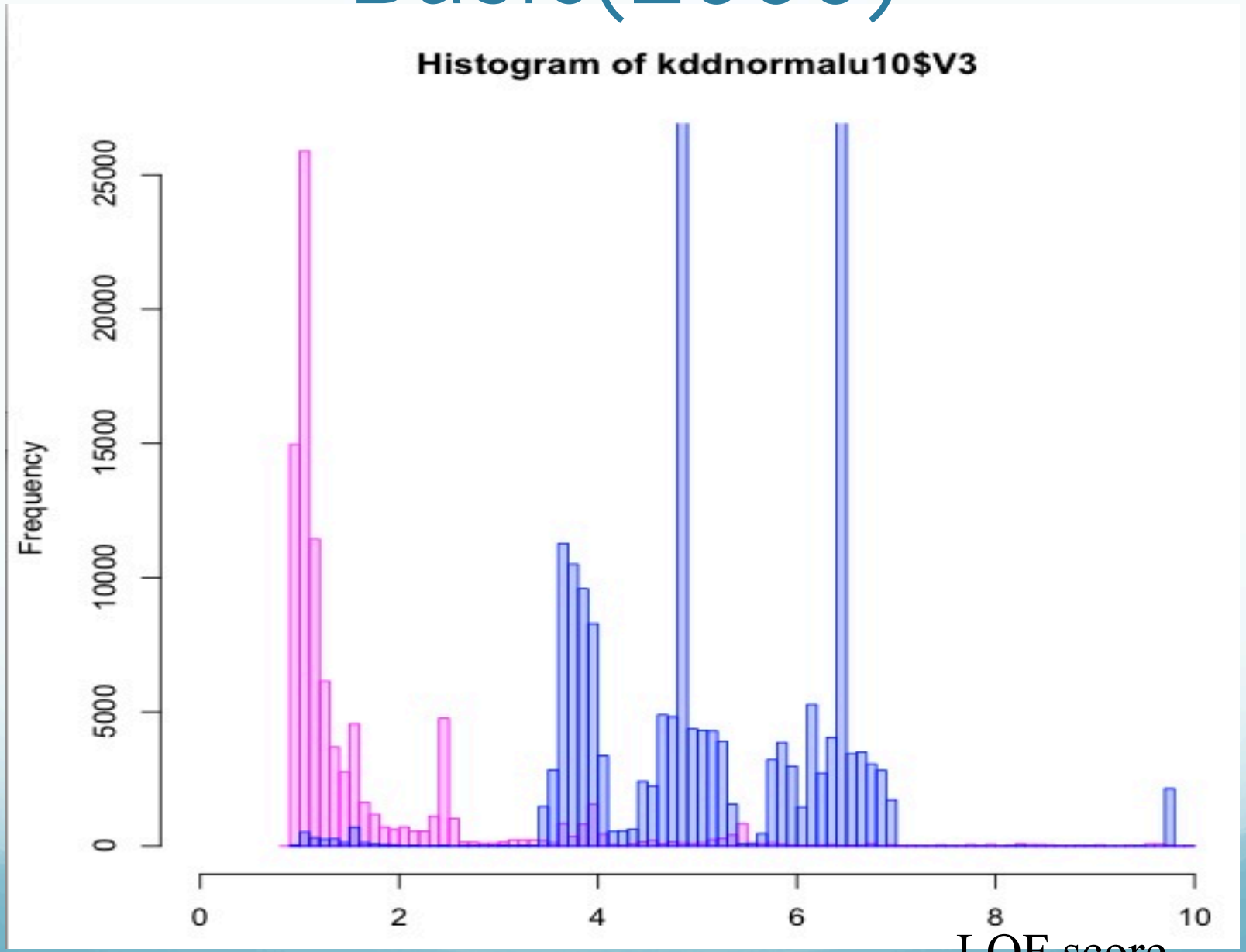
  → Intrusion Detection Systemとしてのチューニングが必要

ご静聴ありがとうございました

- 以下、補足資料


Histgram of normal data
(Number of Training Data = 2000)
Threshold
True
Negative
False Positive
Frequency
0
20000
40000
0
2
4
6
8
10
LOF score
Histgram of attack data
(Number of Training Data = 2000)
False
Negative
True Positive
Frequency
0
20000
40000
0
2
4
6
8
10
LOF score

Basic(2000)


Histogram of kddnormalu10$V3
Frequency
0
5000
10000
15000
20000
25000
0
2
4
6
8
10
LOF score