卒 業 研 究 報 告 書

題　　目

トラフィック特徴量の相関特性を用いた異常検出

Anomaly Detection by
Correlative Characteristic of Internet Traffic Features

研究グループ　　第 5 グループ

指導教員　　汐崎 陽 教授

平成 19 年 (2007 年) 度 卒業

(No. 1031060322)　　松本 亮介

大 阪 府 立 大 学 工 学 部 情 報 工 学 科

目次

図一覧

表一覧

1. はじめに

近年，コンピュータネットワークシステムが社会に普及していくにつれ，計算機やインターネットは社会に必要不可欠となってきた．しかし，それに伴って，ネットワークを経由した計算機に対する不正アクセスは年々増加しており，社会問題になっている．また，セキュリティの保護無しでは個人でインターネットを楽しむことでさえ難しくなってきている．そこで，不正アクセスを含む異常事象の検出のために，管理者がネットワークを常時監視し，分析する必要が生じているが，近年の計算機やインターネットの急速な発展によって，大規模なネットワークを人手で分析することは現実的ではなくなってきた．そこで，最近では自動的にネットワークセキュリティを保護するためのシステムが開発・研究されてきており，その 1 つに，ネットワーク上で発生する異常事象などを分析する IDS（Intrusion Detection System）と呼ばれるセキュリティシステムが注目されてきている．

現在，IDS で最も普及している手法として，不正検出がある．不正検出とは，既知の異常事象をデータベースとして保存しておき，それを用いてパターンマッチなどを行うことによって，異常事象を検出する手法である．一度データベースに保存した異常事象は，ほぼ完璧に防ぐことができるため，多くの企業やセキュリティソフトに用いられている．しかし，不正検出には未知の異常を検出できないという問題があるため，最近では異常検出と呼ばれる，対象となるデータに対して統計的手法などによって異常事象などの振る舞いを学習し，未知の異常を検出する手法が盛んに研究されてきている．

異常検出に関する重要な要件の 1 つに, 異常事象のリアルタイム検知がある．リアルタイム検知には，学習に必要な計算量を減らし，異常事象の特徴を素早く捉えるアルゴリズムが必要である．そこで，ネットワークトラフィック情報から通信量やトラフィックの特徴量などを時系列データとして抽出し，そのデータが急激に変化する時刻において異常である可能性が高いとするなど，時系列データから変化点を検出することにより異常とする手法が提案されている．時系列データからの変化点検出は，ログデータなどから異常行動や不正行為につながるデータや，新しいトレンドを示す重要なデータを発見することができるとされている．そのため，データマイニングに関する研究において，最も注目されている．文献[1]では，時系列データの標準偏差の 2 倍から 3 倍を示すような値である外れ値と変化点に明確な関係を与え，外れ値と変化点を区別して検出するアルゴリズムを提案している．この学習アルゴリズムは，時系列データを当てはめる確率モデルが過去の統計量を次第に忘れていくことによって，リアルタイムで時系列データの特徴をうまく抽出し追跡できる．ネットワーク

における異常事象は,トラフィック量やアクセス頻度を時系列データとして見た場合，大きな変化が現れた時刻に特徴として現れることが多い．これまでの研究では，そのような特徴量を単一で利用することにより異常を検出している．しかし，大量の通信量やアクセス頻度を伴うネットワークにおいては，単一の特徴量の変化だけで正常異常を区別することは困難である．また，全体の通信量に対して，異常な通信量が少ない場合においても，統計的手法では検出することができない．

そこで本稿では，通信量やトラフィック特徴量を単一で用いた場合に生じる問題点を改善し，リアルタイムで異常事象を検出するために，複数のトラフィック特徴量の相関関係を利用した手法を提案する．現在，インターネットやイントラネットで標準的に使われているプロトコルはTCP/IP（Transmission Control Protocol/Internet Protocol）であり，TCP/IP に基づいて通信を行っている．そのため，トラフィックの特徴を抽出した時系列データの中には，強い相関を持つデータが含まれている．このトラフィック特徴量間に生じる相関特性を利用して，通常状態において相関関係に大きな変化が現れないような特徴量を抽出し，特徴量間に生じる相関関係から相関係数を計算し，さらに統計的手法[1]による変化点検出手法を用いることで，相関係数が大きく変化する時刻を検出することにより異常事象を検出できる．

以降 2 章においてネットワークトラフィックからの異常検出ついて述べ，3 章でトラフィック特徴量の相関特性を用いた提案手法について説明する．4 章で実験と考察，5 章でむすびとする．

2. ネットワークトラフィックからの異常検出

2.1 ネットワークトラフィックの概要

インターネットは，TCP/IP プロトコルに基づいて，一定のルールの元で通信を行っている．そのため，大規模ネットワークにおいて，トラフィック情報から特徴を抽出した場合，通常の通信は類似した特徴が抽出できる場合が多い．そのため，通常の通信では発生しにくいトラフィックが現れた場合，その通信を通常の通信と区別することができると考えられる．以下では，TCP/IP で定められている通信方法と異常事象の特徴について説明する．

2.1.1 TCP ヘッダと IP ヘッダによる特徴

現在，インターネットにおいて標準的に使われているプロトコルは TCP/IP プロトコルである．TCP/IP プロトコルは、世界最大のネットワークであるインターネットを構成するための基盤技術として誕生し、インターネットとともに発展してきた。そのため、インターネットの成長に応じて発生したさまざまな問題を解決するための改良がなされている。つまり、TCP/IP プロトコルについてその詳細を見ることで、インターネットを構成するさまざまな技術を理解できるばかりか、ネットワークで発生する問題を知り、その解決方法を身につけることも可能である．さらに，国際標準化機構（ISO）に制定された，「OSI 基本参照モデル」（異機種間のデータ通信を実現するためのネットワーク構造の設計方針）には階層的に，物理層，データリンク層，ネットワーク層，トランスポート層，セッション層，プレゼンテーション層，アプリケーション層が存在する．OSI 基本参照モデルにおいてネットワーク層に属する IP は，ネットワークに接続している機器の住所付け(アドレッシング)や，相互に接続された複数のネットワーク間での通信経路の選択(ルーティング)を行っており，IP ヘッダの構造は図 2.1 のようになっている．IP ヘッダ部に含まれる情報のうち，送信先 IP，送信元 IP，パケットサイズはネットワークトラフィックを表現するために有効な特徴量である．IP アドレスは，ある特定のホストからアクセスが非常に多い場合，そのアクセス数をカウントし，時系列で統計処理して分析することで，通常と異なった状態を特定できる可能性がある．パケットサイズは，IP ヘッダ部とデータ部の合計，つまり，IP パケット全体のサイズをバイト単位で表現したものである．さらに，イーサネットを用いている場合，ネットワークの最大転送単位（MTU）は 1500 バイトであり，1500 バイト以上のパケットはフラグメント化される．そのため，

大量のデータを送りつける不正アクセスであれば, フラグメント化が多く行われ, その特徴が現れると考えられる．また，トランスポート層に属する TCP に関しては，コネクション型のエンドシステム間における通信の信頼性を高めており，TCP ヘッダの構造は図 2.2 のようになっている. TCP ヘッダ部に含まれる情報のうち，送信先 Port，送信元 Port，TCP フラグは，ネットワークトラフィックを表現する有効な特徴量として考えられる．一般的に，不正アクセスが行われる前段階において，サービスが提供されている主要ポートを調査し，そのポートにおける脆弱性をつく不正アクセスを行う場合が多い. そのため，ポートに対するアクセス数や接続ポートの種類などは, 異常状態を発見するのに有効だと考えられる．TCP フラグに関しては，2.2.1 で詳細に説明する．

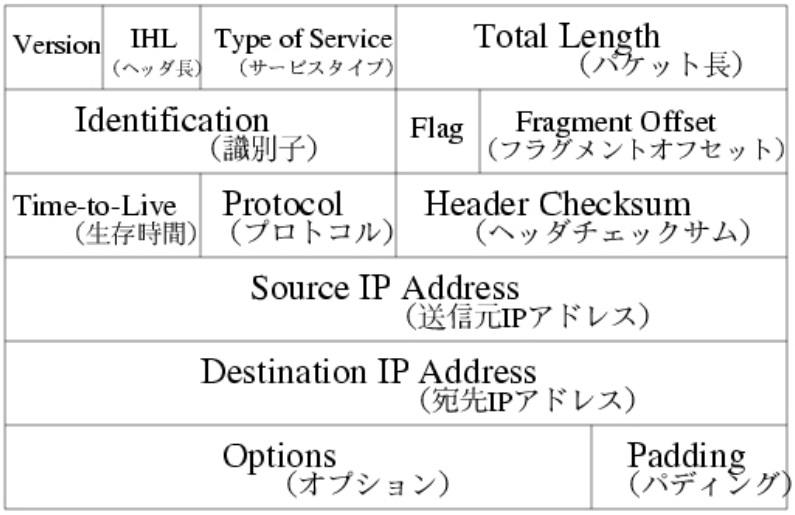

図 2.1　IP ヘッダ

Fig.2.1　IP header format.

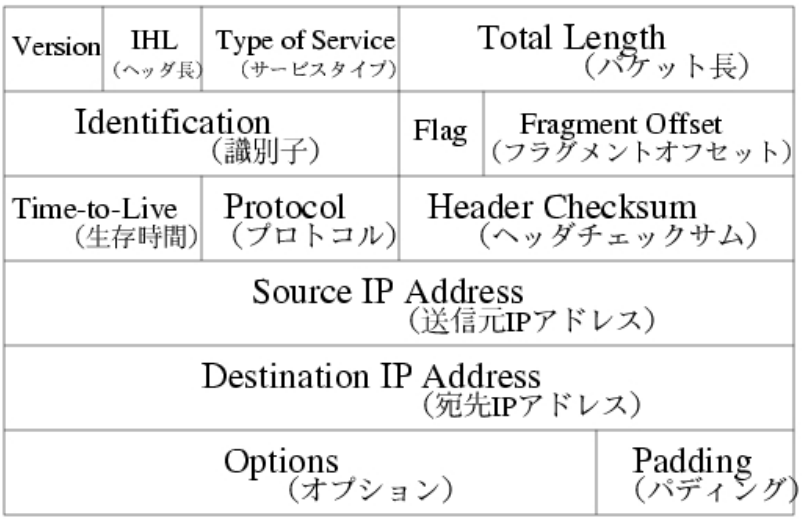

図 2.2　TCP ヘッダ

Fig.2.2　TCP header format.

2.1.2 3way-handshake

IP パケット（ネットワーク層やトランスポート層を流れる分割されたデータの単位）に付与されるヘッダのフィールドは数多くあるが，TCP ヘッダの「TCP フラグ」は，通信に信頼性を持たせるためのコネクション型の通信を行う際に重要な機能を担っている．TCP における通信の流れを図 2.3 に示す．まず，送信元からコネクション確立のために SYN フラグがセットされた IP パケットを送出し，送信先が SYN と ACK のフラグが立った IP パケットを返送してくると，最後にもう一度 ACK フラグがセットされた IP パケットを送出して通信を開始する．このコネクション確立のための手順を「3way handshake」と呼ぶ．また，データの通信を終える際には，送受信するホスト間で相互に FIN フラグのセットされた IP パケットを送り，片方ずつ通信を切断していく．このコネクション切断のための手順を「ハーフクローズ」と呼ぶ．図 2.3 にコネクションの開始から終了までの流れを示す．

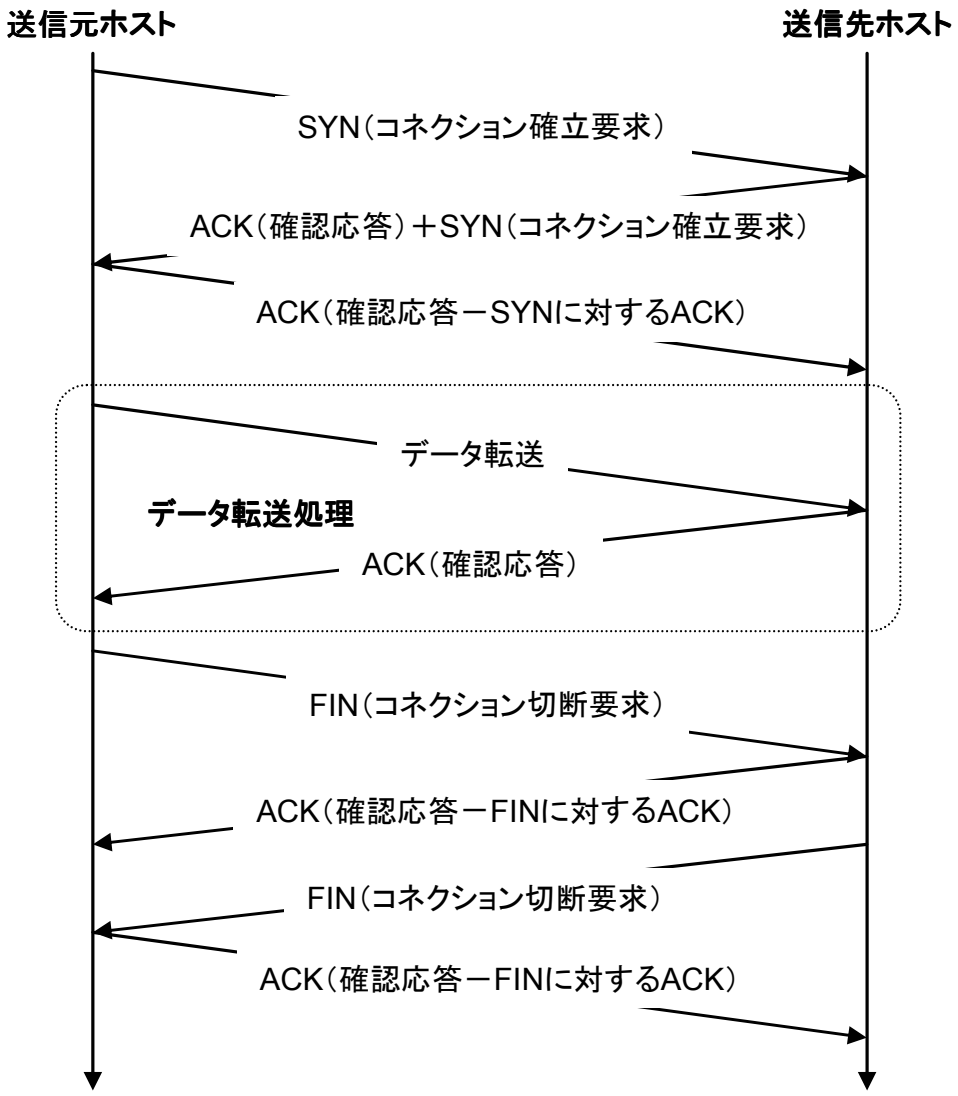

図 2.3　コネクションの開始から終了までの流れ

Fig.2.3　Correspondence of connection process.

2.2 異常事象の特徴

前述したように，侵入検出手法には不正検出と異常検出がある．異常検出では，正常な状態の学習データベースを作成し，通常と違ったデータを異常として検出する．そのため，不正検出で用いられるパターンマッチング手法において，パターンが登録されていないために検出できない不正アクセスを検出できる可能性がある．不正アクセスとして大きく取り上げられる問題のうち，不正アクセスの前段階で使用される「Prove 攻撃」と，サーバに高負荷をかけることを目的とした「DoS 攻撃」に注目し，それらの特徴を説明する．

2.2.1 Prove 攻撃

Prove 攻撃として、最も使用されるポートスキャンを例に挙げて説明する．ポートスキャンとは，サーバのサービスポートに順にアクセスし，動作しているサービスや OS の種類を調べ，侵入口となりうる脆弱なサービスポートがないかどうかを調べる行為のことである．もともとは，ネットワーク管理者がネットワークに接続されているホストの提供しているサービスを調査するための手法であったが，悪意のあるユーザがこのポートスキャンを，攻撃対象を調査するために使うことが多くなっている．一般にポートスキャンには短時間で自動的に多数のポートについて調査を行うプログラムが用いられる．さらに，アクセスに対するサーバの反応を調べるため，送信元 IP アドレスは偽装されないことが多い．そこで，ログにスキャンの痕跡を残すのを回避するために，通常の TCP 接続を確立するプロセスを途中まで行うことでネットワークサービスの存在を確認する．TCP 接続を確立するためには，3way handshake が用いられるため，正常なデータ転送を行うための通信の確立の流れは図 2.4 のようになる．しかし，接続先ホストが SYN と ACK の フラグが立ったパケットを返送してくると，接続の痕跡を残さないようにするために，接続の強制終了である RST のフラグを立ててパケットを送出する．このような手法を用いることにより，接続先に接続してきたことを知られることなく，接続先がどのようなサービスを提供しているか知ることができる（図 2.5）．また，接続先がサービスを提供していない場合は RST フラグが立ったパケットが接続先から送出され（図 2.6），スキャンを行ったホストが存在しなった場合は返答のパケットが返送されてこない（図 2.7）．このように TCP フラグに侵入の前段階であるポートスキャンの傾向が現れると考えられる．ポートスキャンは，途中までは通常の TCP コネクションを成立させる手順であるため，正常なアクセスと判別が難しく，防ぐことが困難であるとされている．

しかし，ポートスキャンでは RST フラグによる強制切断を行うため，通常の

アクセスにおいて FIN フラグをヘッダ情報の値として持つパケット数と，SYN フラグをヘッダ情報の値として持つパケット数は対応しており強い相関関係にある．そのため，この相関関係に大きな変化が現れると考えられる．この特徴を利用することで，相関が大きく変化した時刻が，異常状態である可能性が高くなると考えられる．このことから，TCP フラグに着目することで，通常と異なる異常状態の特徴として有効であると考えられる．

また，通常のアクセスでは，サーバで提供されているサービスはある程度決まっているため，短い時間間隔で，提供しているサービス以上のポートに特定の IP から複数回接続してきた場合も，異常状態である可能性が高くなると考えられる．

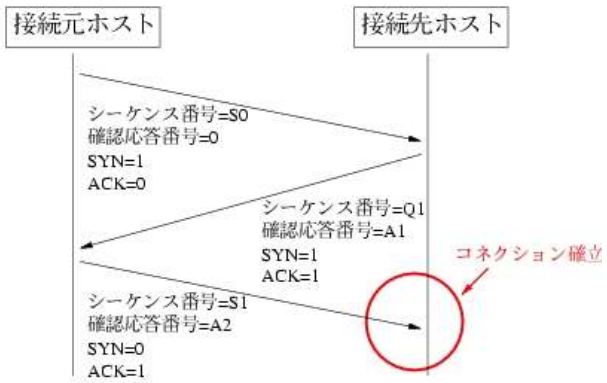

図 2.4　正常なコネクション確立

Fig.2.4　Establishment of normal connection.

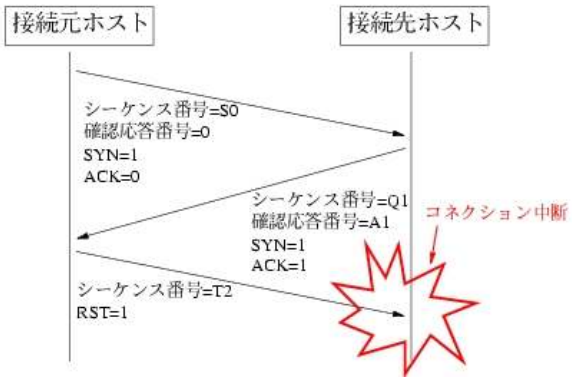

図 2.5　サービスが提供されている場合

Fig.2.5　Case of service existence.

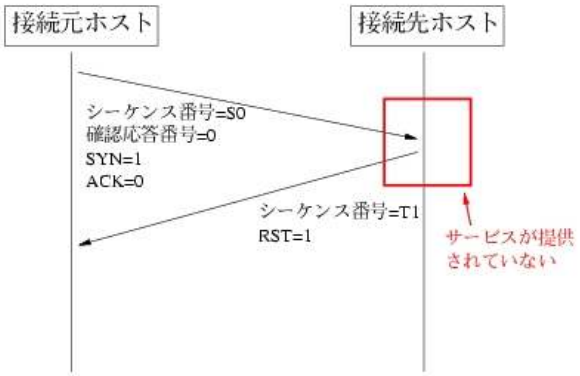

図 2.6　サービスが提供されていない場合

Fig.2.6　Case of non service.

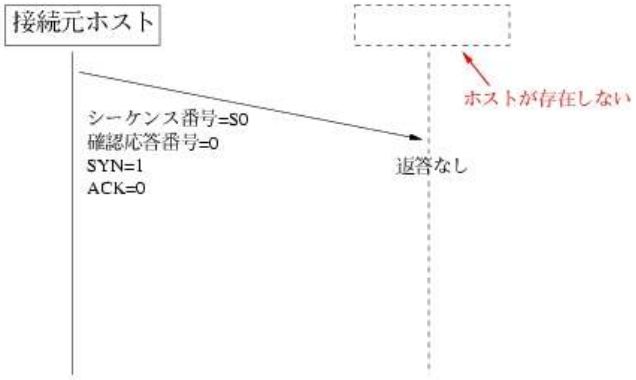

図 2.7　ホストが存在しない場合

Fig.2.7　Case of non host.

2.2.2 DoS 攻撃

DoS（Denial of Services）攻撃とは，サーバに大量の接続要求を送って回線速度を低下させたり，過負荷でサーバを停止させる攻撃のことである．送出するパケットサイズは非常に小さいものから，ネットワークが受け入れられる限界までの大きなサイズまで多岐に渡る．図 2.8 に一般的な DoS 攻撃の例を示す．一般的に，サーバに用いられているコンピュータは性能が高いため，サーバに過負荷を与えるために攻撃側は単独のホストからの攻撃ではなく，複数のホストからの攻撃である DoS 攻撃を同時に行う DDoS（Distributed DoS）攻撃による被害が最も多い．これは，単一のホストからの攻撃であればそのホストとの通信を遮断すればよいが，数千，数万台のホストからの攻撃の場合，サービスを保持したままでは単純に遮断することが難しい．通常の DoS 攻撃よりも防御が難しいとされ，さらに踏み台とされたホストが攻撃元として認識され，攻撃の意思のないホストがウィルスやツールなどによって踏み台とされる場合もある．そのため，ネットワークに接続している全てのコンピュータが DDoS 攻撃を行う踏み台となる可能性があることを意識すべきであるというのがコンピューターセキュリティにおける共通の見解である．標的にされてしまうと，遮断すべき対象の通信が広範囲に及ぶことから，正式な利用者までもが不利益を被ってしまうため，現時点でこれらの攻撃を完全に防御する方法は存在しないとされている．

DoS 攻撃では，サーバからの反応を逐一確認する必要がないため，送信元 IP アドレスは偽装されていることが多い．また，掲示板などによって人手を集め，単純に Web サイトを再読み込み（リロード）し続けることで DoS 攻撃を実行することも行われている．このため，一つ一つのアクセスは正常な通信であることが多く，DoS 攻撃は防ぐことが難しい攻撃といえる．しかし，特定の条件，例えば単純な DoS 攻撃では，ツールなどによって機械的に大量の通信を行うため，送信先 IP と送信先ポートは一定であり，さらにはパケットサイズが等しいことが多い．そこで，通常のアクセスと区別するために，このような特徴を抽出し時系列変化を分析することで，DoS 攻撃を検出できる可能性が高くなる．

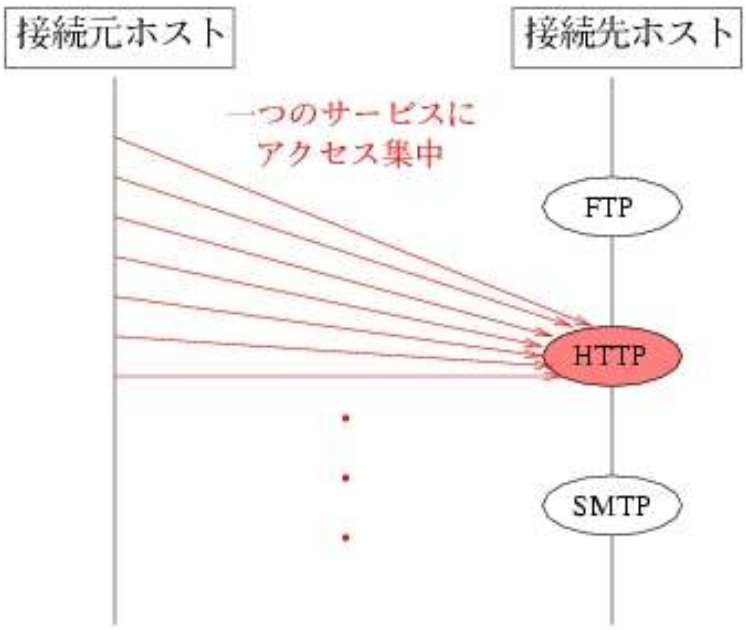

図 2.8　DoS 攻撃の例

Fig.2.8　Basic example of DoS attack.

2.3 異常検出における既存手法

ネットワークにおける異常検出手法では, まず, 様々な情報を含むネットワークトラフィックデータの中から異常検出に有効とされる情報を特徴量として抽出し, その情報を分析することにより異常を検出する. これまでの多くの手法は, 統計情報として異常データを収集し, その異常データを蓄積することによって学習データとし，学習データから正常状態と異常状態を分類している.

2.3.1 特徴量として用いられる情報

ネットワーク上において各種のサービスを提供しているサーバは, そのサービス以外はファイアーウォールなどで守られているのが一般的である．そのため, 主要なサービスを利用するトラフィックに対して異常検出を行えば, ほとんどの不正アクセスを検出できると考えられる. そこで, サービスに対するネットワークトラフィックから特徴を抽出するために, パケットのヘッダ情報を用いて特徴量を抽出するといった方法[2]がとられている.

2.3.2 従来手法とその問題点

ネットワークトラフィックデータから異常検出に有効とされる特徴量を抽出し, これらの特徴量を正常な分布と異常な分布に分類する必要がある. これまで, 分類手法として多くの提案がされているが, 分類対象に単一のトラフィック特徴量の時系列変化を使用し分類している場合が多い. しかし, 不正アクセスなどの異常事象であっても，1 つ 1 つの通信は正常である場合が多く，単一の特徴量を評価するだけでは正常状態か異常状態かを区別することが難しい.

そこで, 提案手法では, 複数のトラフィック特徴量における相関関係を利用した手法を提案する．分類対象として特徴量間における相関関係を用いることで, 単一の特徴量では検出することができない異常を検出することができる.

3. トラフィック特徴量の相関特性を用いた提案手法

3.1 概要

これまでの異常検出に関する手法の問題点として,ネットワークトラフィック情報から単一の特徴量の時系列変化を用いていることが挙げられる．そこで，提案手法では，複数のトラフィック特徴量の相関関係を利用した手法を提案する．通常状態では相関関係に大きな変化が現れない特徴量間の相関関係を抽出することで，相関関係が大きく変化すれば異常状態であるとする．3 章では，トラフィック特徴量の抽出方法，特徴量間の相関関係の抽出方法，相関関係が大きく変化する時刻を検出する方法について述べる．

3.2 トラフィック特徴量抽出処理

ここでは,ネットワークトラフィックからの特徴量の抽出方法について述べる．まず，前処理として，ネットワークの特徴を反映した数値による時系列データに変換する．抽出する方法は 2 種類ある．まず，トラフィック情報から，単一のヘッダ情報の値（TCP フラグ，パケットサイズの範囲，サービス）に着目して，パケットに，着目するヘッダ情報の値を含んでいれば，そのパケットを単位時間（1 分）でカウントする．これを特徴量（A）とする．例えば，サービスである HTTP に着目した場合，ヘッダ情報に HTTP を表すビット列を含んだパケットを単位時間でカウントする．同様に，特徴量（A）でカウントの対象となったパケットにおいて，複数のヘッダ情報の値（送信先 IP，送信先 Port，送信元 IP，送信元 Port，パケットサイズ）の組み合わせに着目し，パケットに，着目するヘッダ情報の値の組み合わせを含んでいれば,そのパケットを単位時間でカウントする．そのカウント対象となったパケットにおいて，ヘッダ情報の値の組み合わせの種類をカウントし，パケット数を種類数で除算する．このことにより，単位時間に複数のヘッダ情報の値の組み合わせを含むパケットに対して,パケットの種類が占める比率を計算することができる．これを特徴量（B）とする．ヘッダ情報を条件に特徴量を抽出した場合，特徴量（A）は条件に対する単位時間におけるパケット数を表しており，特徴量（B）は条件に一致したパケットにおいて，パケット数に対するパケットの種類数が占める比率を表している．図 3.1 に特徴量抽出の例を示す．例では，特徴量（A）に用いるヘッダ情報の条件は HTTP とし，特徴量（B）には，送信先 IP，送信先 Port，パケットサイズを用いた．単位時間は 1 分とした．HTTP に関する 00:37 のトラフィック情報については，総パ

ケット数は 7 パケットとなる．これが，00:37 における特徴量（A）となる．また，送信先 IP，送信先 Port，パケットサイズの組み合わせの種類数は，四角で囲んだ 4 種類が考えられる．ここで，00:37:07 と 00:37:17 のパケットは，パケットサイズの情報が無いため除外する．また，ヘッダ情報の値の組み合わせの種類が占めるパケット数は 5 パケットとなるため, パケット数を種類数で除算することにより，特徴量（B）が計算できる．

多種の通信が生じる通常のネットワークにおいて, パケット数とパケットの種類数は相関関係にあるため, これらの特徴量間にも相関関係があると考えられる．例えば DoS 攻撃などの異常が起きた場合は，機械的に短時間で大量の通信を行うため，プログラムやツールなどによる攻撃が多く，ヘッダ情報の送信先 IP と送信先 Port とパケットサイズが特定の値に偏ることが多い．そのため，特徴量（B）において，パケットの種類数が少なくなる一方，パケット数は大きく増加するため，特徴量（A）の変化よりも特徴量（B）の変化が急激になり，特徴量間の相関関係が大きく変化する．図 3.1 の通常状態の例の 1 分後に DoS 攻撃が生じた場合の特徴量抽出の例を図 3.2 に示す．特徴量（B）において，種類数が 1 種類であるため，特徴量（A）に対して特徴量（B）の値が急激に増加しているのが分かる. この場合, 00:38 の時間帯に異常事象が発生していると判断する．

[例]　特徴量(A):HTTP
特徴量(B):(送信先IP、送信先ポート、パケットサイズ)
単位時間:1分

時間　送信元IP.送信元ポート > 送信先IP.送信先ポート: (パケットサイズ)
00:37:07 IP 172.16.114.50.http > 206.48.44.50.2222: . ack 5841 win 32120
00:37:17 IP 172.16.114.50.http > 206.48.44.90.2313: . ack 2921 win 32120
00:37:25 IP 206.48.44.40.2222 > 172.16.114.30.http: . (256) ack 8192 win 31744
00:37:25 IP 206.48.44.50.2222 > 172.16.114.40.http: . (1320) ack 8192 win 31744
00:37:38 IP 206.48.44.60.2222 > 172.16.114.70.http: . (156) ack 8192 win 31744
00:37:49 IP 206.48.44.90.2313 > 172.16.114.50.http: . (1460) ack 8192 win 31744
00:37:58 IP 206.48.44.90.2313 > 172.16.114.50.http: . (1460) ack 8192 win 31744

7

$$[\text{総パケット数}] = 7 \quad \Rightarrow \text{特徴量(A)}$$

$$\frac{[\text{パケット数}]}{[\text{種類数}]} = \frac{5}{4} \quad \Rightarrow \text{特徴量(B)}$$

図 3.1 特徴量抽出の例（通常状態）

Fig.3.1　Extraction of internet traffic features(Normal).

時間　送信元IP.送信元ポート > 送信先IP.送信先ポート: (パケットサイズ)
00:38:07 IP 172.16.114.50.http > 206.48.44.50.2222: . ack 5841 win 32120
00:38:17 IP 172.16.114.50.http > 206.48.44.90.2313: . ack 2921 win 32120
00:38:25 IP 206.48.44.50.2222 > 172.16.114.50.http: . (320) ack 8192 win 31744
00:38:25 IP 206.48.44.50.2222 > 172.16.114.50.http: . (320) ack 8192 win 31744
00:38:38 IP 206.48.44.50.2222 > 172.16.114.50.http: . (320) ack 8192 win 31744
00:38:49 IP 206.48.44.90.2313 > 172.16.114.50.http: . (320) ack 8192 win 31744
00:38:58 IP 206.48.44.90.2313 > 172.16.114.50.http: . (320) ack 8192 win 31744

7

$$[\text{総パケット数}] = 7 \quad \Rightarrow \text{特徴量(A)0}$$

$$\frac{[\text{パケット数}]}{[\text{種類数}]} = 5 \quad \Rightarrow \text{特徴量(B)}$$

図 3.2 特徴量抽出の例（DoS 攻撃）

Fig.3.2　Extraction of internet traffic features(DoS Attack).

3.3 相関係数時系列データの計算

3.2 によって得られた 2 種類の特徴量の相関関係を表す時系列データを作成する．計算にはピアソンの積率相関係数を用いた．相関係数を求める際に，窓サイズ N を決定し，N を 1 ずつずらすことによって，N 間隔における相関係数の時系列データを得る．2 組の時系列データ $(x,y)=\{(x_i,y_i)\}(i=1,2,\cdots.N)$ が与えられた時，相関係数 c は，ピアソンの積率相関係数によって次式(3.1)で計算する．

$$c=\frac{\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i-\bar{y})^2}} \tag{3.1}$$

ただし，$\bar{x},\bar{y}$ はそれぞれの時系列データの窓サイズ N 内における相加平均である．ピアソンの積率相関係数は，線形相関にしか対応していないため，曲線相関の場合は正確に相関係数を求めることができないとされているが，N 区間を適宜変更し，最適な N を先験的に決める予備実験の結果，十分に異常事象の特徴が得られる値を計算できることが確認できた．

3.4 時系列データの変化点検出

3.3 で得られた相関係数の時系列データが急激に変化する時点を検出するために，ChangeFinder[1]と呼ばれる変化点検出エンジンを用いる．ChangeFinder は竹内，山西らによって提案され，計算量を減らすことによってオンラインで時系列データが急激に変化する時刻を検出できる手法である．ChangeFinder の特筆すべき点は，2 段階の学習過程を繰り返すところにある．最初に，第 1 段階で学習したモデルを利用して外れ値を検出する．次に，第 2 段階で学習したモデルを用いて変化点検出を行う．

3.4.1 ChangeFinder の概要

第一段階学習では，まず各時刻 t において，AR モデルを SDAR アルゴリズムによって学習する．そして，時系列データに対する外れ値らしさを示す，外れ値スコアを計算する．

第二段階学習では，外れ値スコアに対して，再度 AR モデルをあてはめ，これを学習して，変化点スコアを計算する．変化点スコアが大きいほど，t が変化点

である度合いが高い.

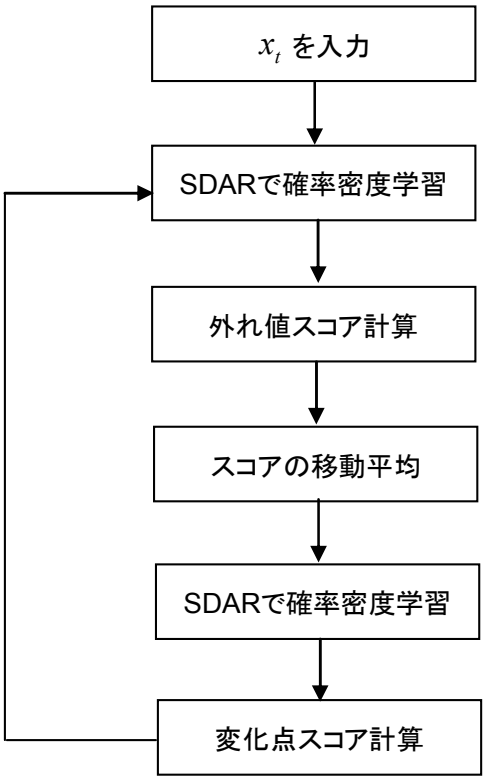

図 3.3　ChangeFinder の流れ
Fig.3.3　Flow chart of ChangeFinder.

ChangeFinder の特徴は, 第一段階学習では時系列中の外れ値しか検出できないが, 外れ値スコアの平滑化を行うことにより, 本質的なモデルの変動を検出しているところにある. 計算量に関しても, データ数 n に対して, 統計的検定に基づく方式が $O(n^2)$ であるのに対して, ChangeFinder の計算量は $O(n)$ で済むため, 明らかに効率がよい.

さらに, ChangeFinder は平均値の変化だけでなく, AR モデルのパラメータ（AR 係数や分散）の変化も原理的には検出できる. 実際に, 分散が突然変化する場合でも, 十分な効果が得られるという報告がある.

3.4.2 外れ値の検出

最初に, 時間 t に対して, $\{x_t : t = 1,2,\cdots\}$ で表すことができる時系列データを考える. tが変化するときの x_t が, 本稿における重要な意味を示す d 次元の実数値ベクトルである. ChangeFinder は, 第 1 段階で, $\{p_t : t = 1,2,\cdots\}$ として表される確率密度関数の時系列データを計算する. データ x_t が入力されると, この時系列データは $\{x_t\}$ から徐々に学習していく. 一般的に, それぞれの p_t が確率過程の密度を表すと考える. 確率過程 p に対して, $x^t = x_1 x_2 \cdots x_t$ を与える x_{t-1} の条件付き確率密度関数を考えるために, $p(x_{t-1} \mid x^t)$ のような表記法を用いる. 学習方法には, 確率過程 p を推定するために, 各入力 x_t に対して, 次式(3.2)を用いて, x_t の外れ値スコアを計算する.

$$\mathrm{Score}(x_t) = -\log p_{t-1}(x_t \mid x^{t-1}) \tag{3.2}$$

式(1)の左辺は, 確率密度関数 $p_{t-1}(\cdot \mid x^{t-1})$ に対する x_t の対数予測損失を表し, 対数損失スコアと呼ぶことにする.

また, 対数損失ではなく, 2 次的損失に基づく別のスコアを次式(3.3)のように定義する.

$$\mathrm{Score}(x_t) = (x_t - \hat{x}_t)^2 \tag{3.3}$$

また，次式(3.4)により，$\hat{x}_t$ は学習モデル p_{t-1} に基づいて，ある x に対する x^{t-1} からの予測を表している．

$$\hat{x}_t := E_{p_{t-1}}\left[x_t \mid x^{t-1}\right] \stackrel{\text{def}}{=} \int x p_{t-1}\left(x \mid x^{t-1}\right) dx \tag{3.4}$$

これを，二次損失スコアと呼ぶことにする．$\text{Score}(x_t)$ が高いほど，x_t が外れ値である可能性が高いことを示す．

3.4.3 AR モデル

例えば，一連の確率過程 $\{p_t\}$ の系列に，AR（自己回帰）モデルを用いるとする．初期値が 0 であるような，定常時系列 $\{z_t : t = 1,2,\ldots\}$ を考える．それぞれの z_t は d 次元の縦ベクトルを表す．このとき，次式(3.5)より，k 次の AR モデルを与えることができる．

$$z_t = \sum_{i=1}^{k} A_i z_{t-i} + \varepsilon \tag{3.5}$$

ただし，データ z_t は n 次元のベクトル，$A_i(i = 1,\ldots,k)$ は n 元正方行列，ε は期待値 0，共分散行列 Σ のガウス分布 $\mathrm{N}(0,\Sigma)$ に従うノイズ項であるとする．

実際に観測されるデータを

$$x_t = z_t + \mu \tag{3.6}$$

で表す．

これより，期待値が μ，$x_{t-k}^{t-1} = (x_{t-1}\cdots x_{t-k})$，であるとすると，$x_t$ の確率密度関数は，

$$p\left(x_t \mid x_{t-k}^{t-1} : \theta\right) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left(-\frac{\xi^T \Sigma^{-1} \xi}{2}\right) \tag{3.7}$$

で与えられる．ただし，

$$\xi = x_t - \left(\sum_{i=1}^{k} A_i z_{t-i} + \mu\right) \tag{3.8}$$

であり， $\theta=(A_1,\ldots,A_k,\mu,\Sigma)$ とする．

AR モデルに関する通常の推定アルゴリズムについて，次式(3.9)，(3.10)を定義する．

$$\hat{\mu}=\frac{1}{t-k}\sum_{i=k+1}^{t}x_i \tag{3.9}$$

$$C_j=\frac{1}{t-k}\sum_{i=k+1}^{t}(x_i-\hat{\mu})(x_{i-j}-\hat{\mu})^T \tag{3.10}$$

式(3.9)は μ の推定値，(3.10)は $x_1,\ldots,x_t$ の相関関数の推定値を表す．さらに A_i の推定値は，以下の $\overline{A_i}$ を未知数とする連立方程式を解くことで得られる．

$$C_j=\sum_{i=1}^{k}\overline{A_i}C_{j-i}\quad(j=1,\ldots,k) \tag{3.11}$$

式(3.11)の解 より， Σ の推定値は，

$$\hat{\Sigma}=C_0-\sum_{i=1}^{k}\overline{A_i}C_i \tag{3.12}$$

によって求めることができる．しかし，この手続きでは，情報源が定常であると仮定されており，いわゆるバッチ学習方式になっている．

3.4.4 SDAR アルゴリズム

ここで，AR モデルを以下のように改良する． θ_t は， x_t が与えられたときの θ の推定値を表し， $p_t=(\cdot\,|\,\cdot\theta_t)$ とする． θ の評価のために，以下の量を最大にする θ の値を計算するアルゴリズムを考える．

$$\sum_{i=1}^{t}(1-r)^{t-i}\log p(x_i\,|\,x^{i-1},\theta) \tag{3.13}$$

これは，オンラインで使用するための，最尤推定法の変形である．このとき，重さが時間 t で指数的に減少するところで，尤度が最大になる．これを，SDAR(sequentially discount-ing AR model estimating)アルゴリズムと呼ぶ．

SDAR アルゴリズムは，バッチ学習方式の AR モデルを改良した，逐次型学習方式である．SDAR アルゴリズムでは，逐次学習と忘却機能という 2 つのポイン

トがある．逐次学習とは，新たなデータを 1 つ読み込むごとにパラメータの推定値を更新する．忘却機能とは，i 時点前のデータの影響が $(1-r)^i$ 倍に減少するようにパラメータの推定値を更新する．これによって，非定常な情報源に対応できる．アルゴリズムのパラメータ r を忘却パラメータと呼び，$\frac{1}{r}$ 個程度の過去データの情報を蓄積するようにする．以下に SDAR アルゴリズムを示す．

SDAR アルゴリズム $(0<r<1$:所与$)$

STEP 1. 初期化

Set $\hat{\mu}, C_j, \hat{A}_j (j=1,\ldots,k), \hat{\Sigma}$.

STEP 2. パラメータ更新

For $t=1,2,\ldots,$

x_t を読み込む:

$$\hat{\mu} := (1-r)\hat{\mu} + r x_t \tag{3.14}$$

$$C_j := (1-r)C_j + r(x_t - \hat{\mu})(x_{t-j} - \hat{\mu})^T \tag{3.15}$$

以下の連立方程式を A_i について解く:

$$C_j = \sum_{i=1}^{k} A_i C_{j-i} \quad (j=1,\ldots,k). \tag{3.16}$$

方程式(3.16)の解を $\overline{A}_1,\ldots,\overline{A}_k$ とし，以下を計算

$$\hat{x}_t := \sum_{i=1}^{k} \overline{A}_i (x_{t-k} - \hat{\mu}) + \hat{\mu} \tag{3.17}$$

$$\hat{\Sigma} := (1-r)\hat{\Sigma} + r(x_t - \hat{x}_t)(x_t - \hat{x}_t)^T \tag{3.18}$$

このアルゴリズムにおいて t 番目のデータまで用いて得られる確率密度関数(3.7)を，p_t と書く．

3.4.5 変化点検出

次に変化点スコアを求める．T を正の整数とする．データ列$\{x_t\}$に対して，T 移動平均スコア y_t を次式(3.19)で定義する．

$$y_t = \frac{1}{T}\left(\sum_{i=t-T+1}^{t} \mathrm{Score}(x_i) \right) \tag{3.19}$$

ただし，$\mathrm{Score}(x_t)$ は式(3.2)より求める．この計算によって，新たな時系列 $\{y_t : t = 1,2,\cdots\}$を得る．

次に，$\{y_t\}$を入力データとして，再度 SDAR アルゴリズムを用いて AR モデルの学習を行う．q_t を y_t まで用いて得られる確率密度関数で表すと，AR モデルによる確率密度関数の列$\{q_t : t = 1,2,\cdots\}$が得られる．

さらに，対数損失式(3.2)と 2 次損失式(3.3)と同様に，T 移動平均スコアを次式(3.20)で定義する．

$$\mathrm{Score}(t) = \frac{1}{T} \sum_{i=t-T+1}^{t} \left(-\log q_{i-1}\left(y_i \mid y_{i-k'}^{i-1} \right) \right) \tag{3.20}$$

式(3.20)は，時間 t の変化点らしさを示す指標となる．すなわち，$\mathrm{Score}(t)$が大きければ，変化度合いが大きいと解釈することができる．これを，変化点スコアと呼ぶ．

3.5 相関係数時系列データの変化点検出による異常検出法

3.1 から 3.3 の一連の処理によって得られた相関係数時系列データを，3.4 による ChangeFinder によって変化点検出を行う．そして，変化点検出によって得られた変化点スコアの時系列データが閾値を越えていれば，その時間帯に異常事象が発生しているとする．

4. 実験と考察

4.1 実験データ

本論文では，検証実験を行うためのデータとして MIT の LINCOLN 研究所が作成した IDS 評価用のデータ[3]の一部を使用した. このデータは同研究所が DARPA(高等研究計画局)の支援によって 1998 年から作成されているもので，IDS の性能を比較するための一般的なデータとして広く利用されている．DARPA のデータには学習データと検証データが用意されている. 学習データと検証データともに 1 週間分のデータは Monday から Friday の 5 日間で構成されており，また，収集されているデータの種類は以下のようなものが含まれている．

・LAN の外部，内部のトラフィック
・カーネルのシステムコール
・Windows NT のイベントログ
・各種ログや設定ファイル
・全ファイルのリスト
・行った不正アクセスのデータ
・不正アクセスを識別するための情報
・ネットワークの構成図

学習データは week1，week2，week3 の 3 週間分あり week2 だけ不正アクセスも含まれたデータである. 検証データは, 不正アクセスを含んだ week4, week5 の 2 週間分が用意されており，含まれる不正アクセスについて詳しく示めされている．例えば，week4 の Monday の ID41.084031 の不正アクセスについてみた場合，表 4.1 のように詳細な情報が示されている．このように攻撃開始時間，攻撃継続時間，IP アドレス，ポート番号などの情報が与えられているため，検証データとしてどのパケットが異常パケットであるか定義することができる．しかし, 表 4.2 で示すように学習データに含まれる不正アクセスに関する情報は, 検証データで与えられている情報に比べると与えられている情報が少ないため, 異常なパケットを特定することはできない．そのため，異常を含んだ week2 の学習データにおいて正常データと異常データの比率を表すことはできなかった．

そこで，本論文では 1999 年 の検証用データとして用意されている week4（Tuesday を除く），week5 の LAN の内部のトラフィックデータを実験評価データに使用した．Week4，week5 の検証用データの総パケット数に対するサービ

スごとのパケット数を表にし，表 4.3，表 4.4 に示す．

表 4.1　week4 の検証データにおける不正アクセス情報
Table 4.1　Unauthorized access data in week4 test data.

ID: 41.084031
Date: 03/29/1999
Name: ps
Category: u2r
Start_Time: 08:18:35
Duration: 00:46:05
Attacker: 209.154.098.104
Victim: 172.016.112.050
Username: haraldl
Ports: At_Attacker: 80{1}, 6000{2} At_Victim: 23{3}

表 4.2　week2 の検証データにおける不正アクセス情報
Table 4.2　Unauthorized access data in week2 test data.

ID	Date	Start_Time	Destination	Score	Name
1	03/08/1999	08:01:01	hume.eyrie.af.mil	1	NTinfoscan
2	03/08/1999	08:50:15	zeno.eyrie.af.mil	1	pod
3	03/08/1999	09:39:16	marx.eyrie.af.mil	1	back
4	03/08/1999	12:09:18	pascal.eyrie.af.mil	1	httptunnel
:	:	:	:	:	:

表 4.3　検証データ（week4）.

Table 4.3　Test data of week4.

	week 4 Monday inside.tcpdump data
パケット数	597142パケット(TCP,Telnet のみ) / 1647573パケット(すべて) 39010パケット(TCP,SMTP のみ) / 1647573パケット(すべて) 10338パケット(TCP,HTTP のみ) / 1647573パケット(すべて)
	week 4 Wednesday inside.tcpdump data
パケット数	421183パケット(TCP,Telnet のみ) / 1766074パケット(すべて) 41708パケット(TCP,SMTP のみ) / 1766074パケット(すべて) 40738パケット(TCP,HTTP のみ) / 1766074パケット(すべて)
	week 4 Thursday inside.tcpdump data
パケット数	487626パケット(TCP,Telnet のみ) / 2356503パケット(すべて) 52359パケット(TCP,SMTP のみ) / 2356503パケット(すべて) 45453パケット(TCP,HTTP のみ) / 2356503パケット(すべて)
	week 4 Friday inside.tcpdump data
パケット数	524614パケット(TCP,Telnet のみ) / 1945538パケット(すべて) 48352パケット(TCP,SMTP のみ) / 1945538パケット(すべて) 26404パケット(TCP,HTTP のみ) / 1945538パケット(すべて)

表 4.4　検証データ（week5）.

Table 4.4　Test data of week5.

week 5 Monday inside.tcpdump data	
パケット数	424431パケット(TCP,Telnet のみ) / 2291319パケット(すべて) 35819パケット(TCP,SMTP のみ)/ 2291319パケット(すべて) 96431パケット(TCP,HTTP のみ)/ 2291319パケット(すべて)
week 5 Tuesday inside.tcpdump data	
パケット数	473277パケット(TCP,Telnet のみ) / 3404824パケット(すべて) 39861パケット(TCP,SMTP のみ) / 3404824パケット(すべて) 210695パケット(TCP,HTTP のみ) / 3404824パケット(すべて)
week 5 Wednesday inside.tcpdump data	
パケット数	429553パケット(TCP,Telnet のみ) / 2087942パケット(すべて) 44403パケット(TCP,SMTP のみ) / 2087942パケット(すべて) 72981パケット(TCP,HTTP のみ) / 2087942パケット(すべて)
week 5 Thursday inside.tcpdump data	
パケット数	482284パケット(TCP,Telnet のみ) / 3201381パケット(すべて) 56641パケット(TCP,SMTP のみ) / 3201381パケット(すべて) 127288パケット(TCP,HTTP のみ) / 3201381パケット(すべて)
week 5 Friday inside.tcpdump data	
パケット数	521429パケット(TCP,Telnet のみ) / 3393918パケット(すべて) 43358パケット(TCP,SMTP のみ) / 3393918パケット(すべて) 115742パケット(TCP,HTTP のみ) / 3393918パケット(すべて)

4.2 実験

本論文では，1 日分のデータを単位時間 1 分で区切り，主要サービス（HTTP，SMTP，TELNET）に対する DoS 攻撃，ランダムポートに対する Prove 攻撃と DoS 攻撃の検出を行った．

特徴量抽出には，表 4.5 に示す条件で行った．特徴量（A）は，主要サービスに対する DoS 攻撃の検出の場合，それぞれのサービスに対応するヘッダ情報（HTTP サービスの検出には HTTP ヘッダ）の値を用いて抽出した．この特徴量を，A1 とする．特徴量（B）は，（送信先 IP，送信先 Port，パケットサイズ），（送信元 IP，送信先 IP），（送信元 IP，パケットサイズ），（送信元 IP，送信先 Port）の 4 種類の組み合わせを用いて 4 種類抽出した．これらをそれぞれ，B1，B2，B3，B4 とする．ランダムポートに対する Prove 攻撃の検出は，ランダムポートに対する DoS 攻撃と同様の手法を用いた．

表 4.5　特徴量に用いる条件
Table.4.5　Conditions to extract internet traffic features.

DoS 攻撃の対象ポート	HTTP（80 番ポート）	SMTP（25 番ポート）	TELNET（23 番ポート）	ランダムポート
特徴量（A）のヘッダ情報	HTTP	SMTP	TELNET	全ポート
特徴量（B）のヘッダ情報	送信先 IP，送信先 Port，パケットサイズ			
	送信元 IP，送信先 IP			
	送信元 IP，パケットサイズ			
	送信元 IP，送信先 Port			

このようにして得られた特徴量（A）と特徴量（B）から相関係数時系列データを 4 種類抽出し，ChangeFinder によってそれぞれに対応する変化点スコアを計算し，その変化点スコアに対して閾値を設定することにより，閾値を超えたスコアの時刻を異常時刻とし検出する．実験 1 では，HTTP に対する DoS 攻撃に対して，単一の特徴量で異常検出を行った場合と，本手法の複数の特徴量の相関特性を用いた場合について，それぞれ異常検出を行い，その検出精度の比較を行った．データには，単一の特徴量を用いた場合において，最も検出精度の高かった week5 の Wednesday のデータを利用した．[手法 1]では，HTTP に対する DoS 攻撃を対象とした場合に得られる特徴量 A1 の変化点検出を行うことによって異常を検出し，[手法 2]では特徴量 B1 の変化点検出によって異常を検出した．また，[手法 3]では，特徴量 A1 と特徴長 B1 から得られる相関係数時系列データを用いて異常検出を行い，[手法 4]では，特徴量 A1 と B1，A1 と B2，A1 と B3，A1 と B4，から得られる 4 つの相関係数時系列データにおいて，いずれかで異常を検出すれば異常とすることにより異常検出を行った．

実験 2 では, 実験 1 の比較実験において最も精度が高かった[手法 4]を, week4, week5 の全ての日に適用した．また，week5 の HTTP に対する DoS 攻撃に関して，DoS 攻撃のカテゴリーに含まれた攻撃であるが，攻撃自体は 1 パケット程度の通信しか行われていないものが含まれていた．本手法では，統計的処理によって異常を検出するため，パケット数の極端に少ない異常を検出することは困難である．そのため，1 パケットの DoS 攻撃を除外した場合の異常検出の精度も確認した．

実験 1，実験 2 の評価方法には，最終的に得られる変化点スコアの時系列データの最小値から最大値を 100 区間に分割し，閾値を最大値から分割した区間ずつ下げていき，最小値まで閾値を下げることによって，閾値毎に得られる検出率と誤検出率を計算した．そして，検出の精度がどの程度であるかを示すために，ROC カーブを用いた．

ROC は，Receiver Operating Curve や Relative Operating Characteristic として知られている．ROC カーブの表現は，水平の軸に誤検出(false positive rate:正常状態を誤って異常状態と判別した数)の率を示し, 垂直の軸に検出(hit rate:異常状態を異常状態と判別した数)の率を示したものである．カーブ上におけるそれぞれの点は，ある特定の閾値を設定した時の検出率と誤検出率の点を示している．ROC カーブは，すべての閾値において点を求めそれを線で結んだものとして表現する．

4.2.1 実験結果 – 手法別比較実験 [実験 1]

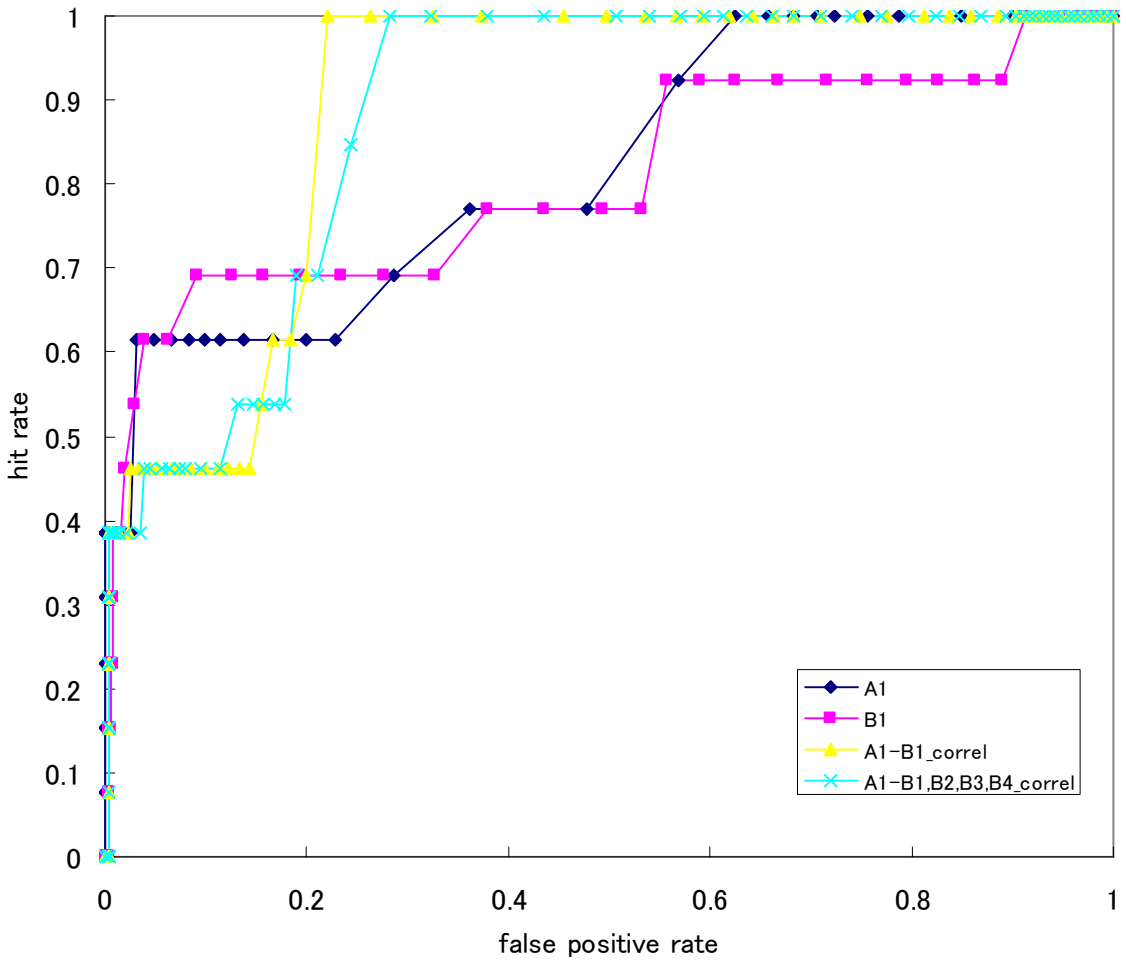

図 4.1 手法別比較実験結果

Fig.4.1 ROC curves for single feature and multiple features.

4.2.2　実験結果 – HTTP における DoS 攻撃検出 [実験 2]

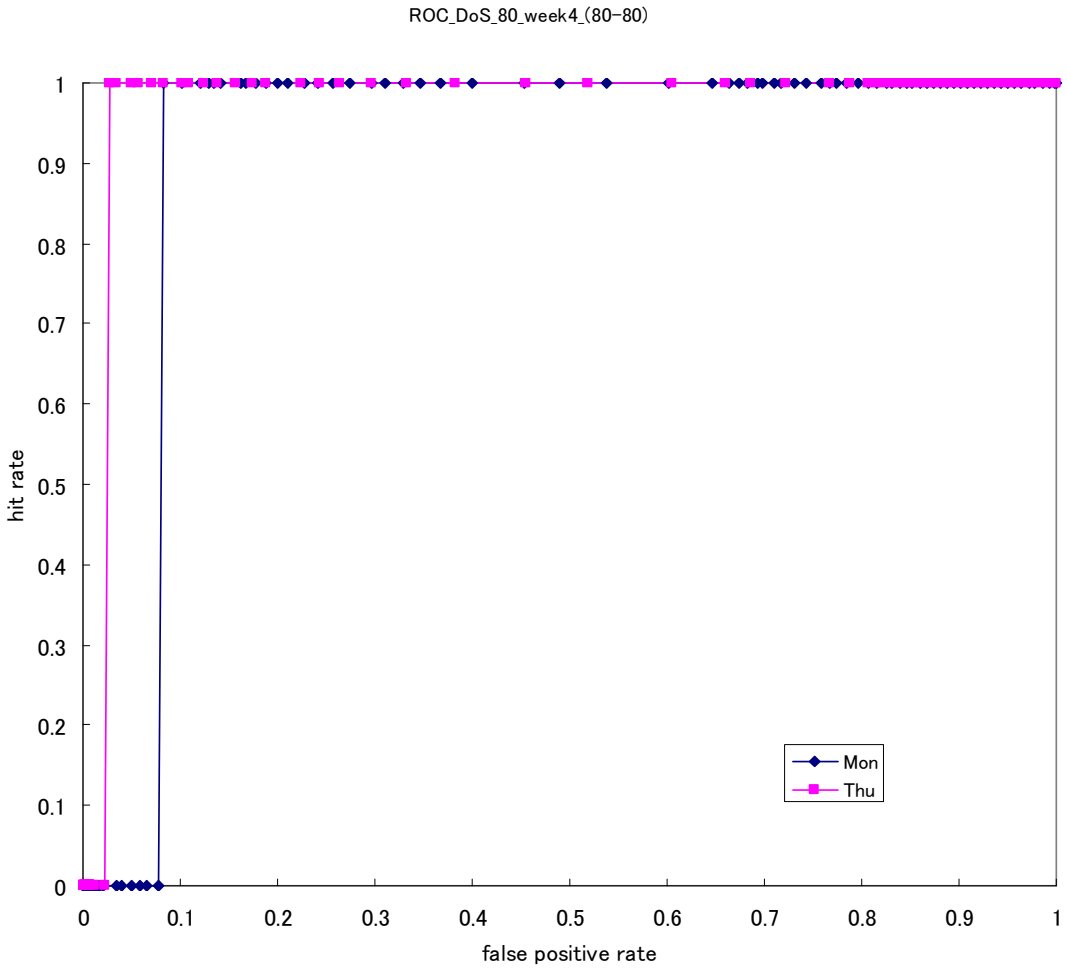

図 4.2　HTTP に対する DoS 攻撃検出結果（week4）

Fig.4.2　ROC curves of week4 under DoS attack in HTTP.

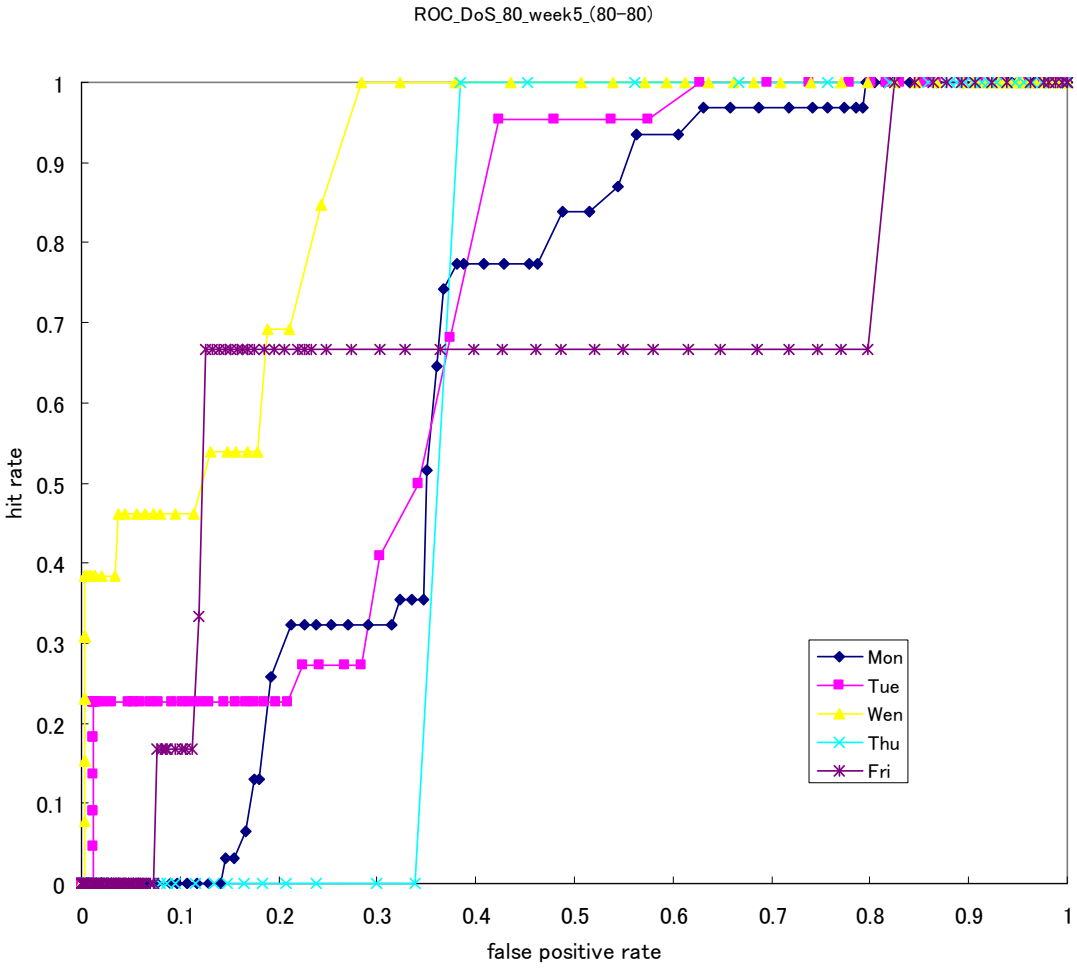

図 4.3　HTTP に対する DoS 攻撃検出結果（week5）

Fig.4.3　ROC curves of week5 under DoS attack in HTTP.

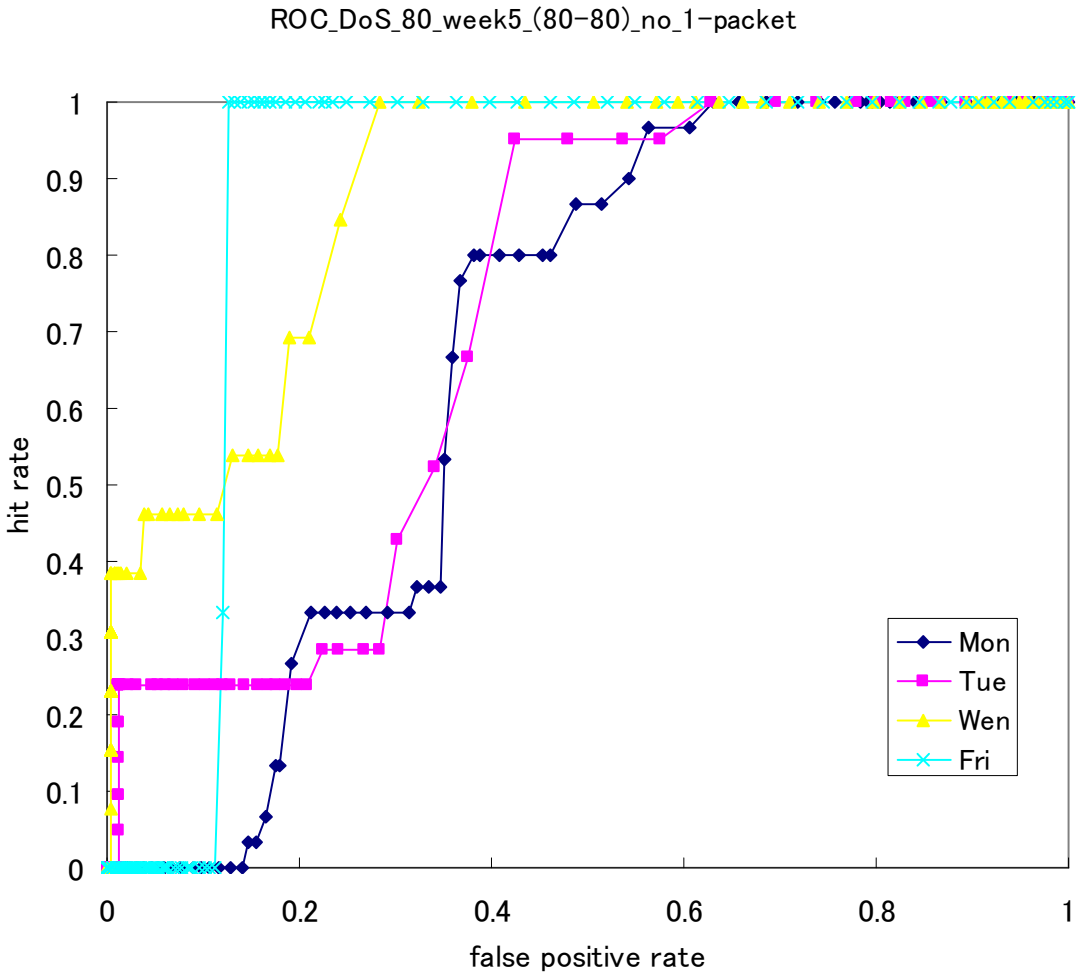

図 4.4　HTTP に対する DoS 攻撃検出結果（week5 - 1 パケットによる攻撃を除外）

Fig.4.4　ROC curves of week5 under DoS attack in HTTP except 1 packet attack.

4.2.3　実験結果 – SMTP における DoS 攻撃検出 [実験 2]

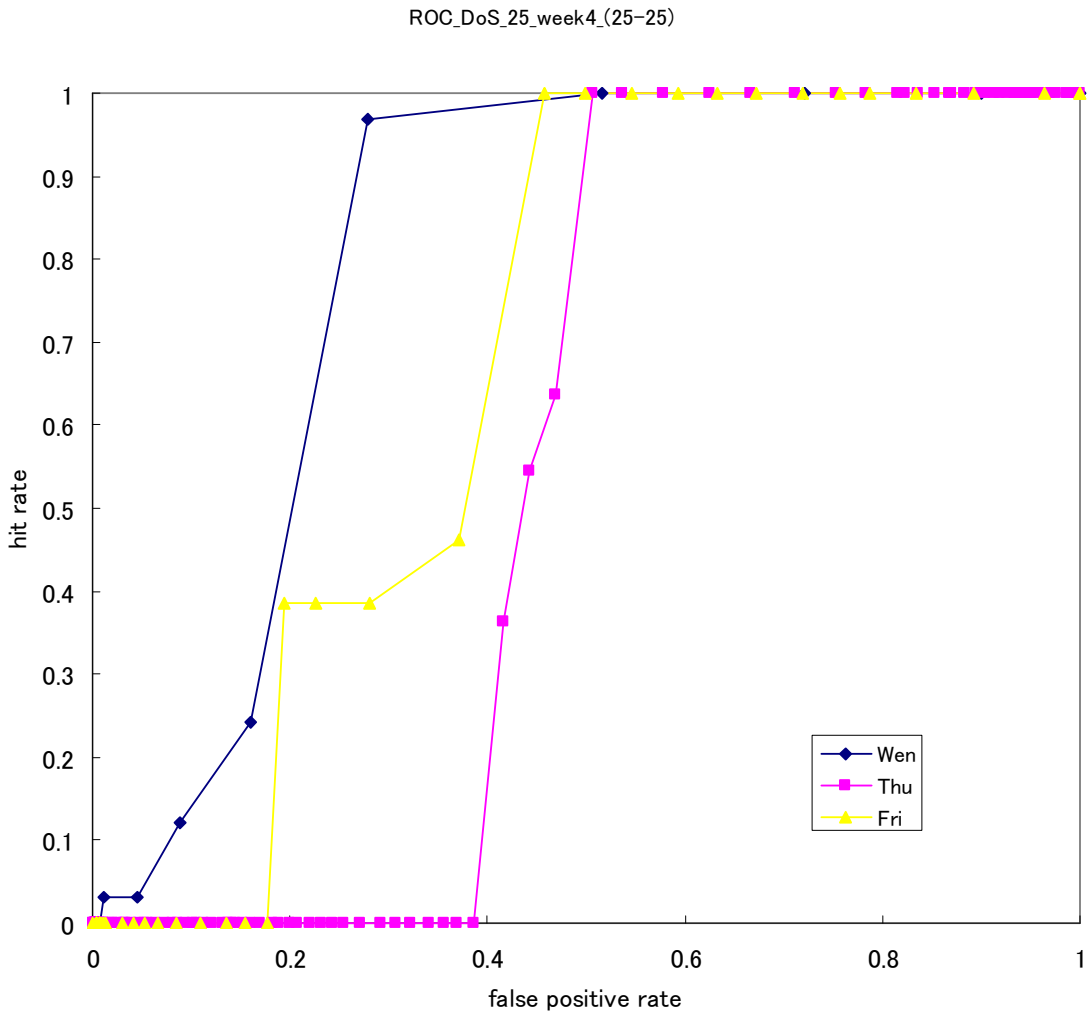

図 4.5　SMTP に対する DoS 攻撃検出結果（week4）

Fig.4.5　ROC curves of week4 under DoS attack in SMTP.

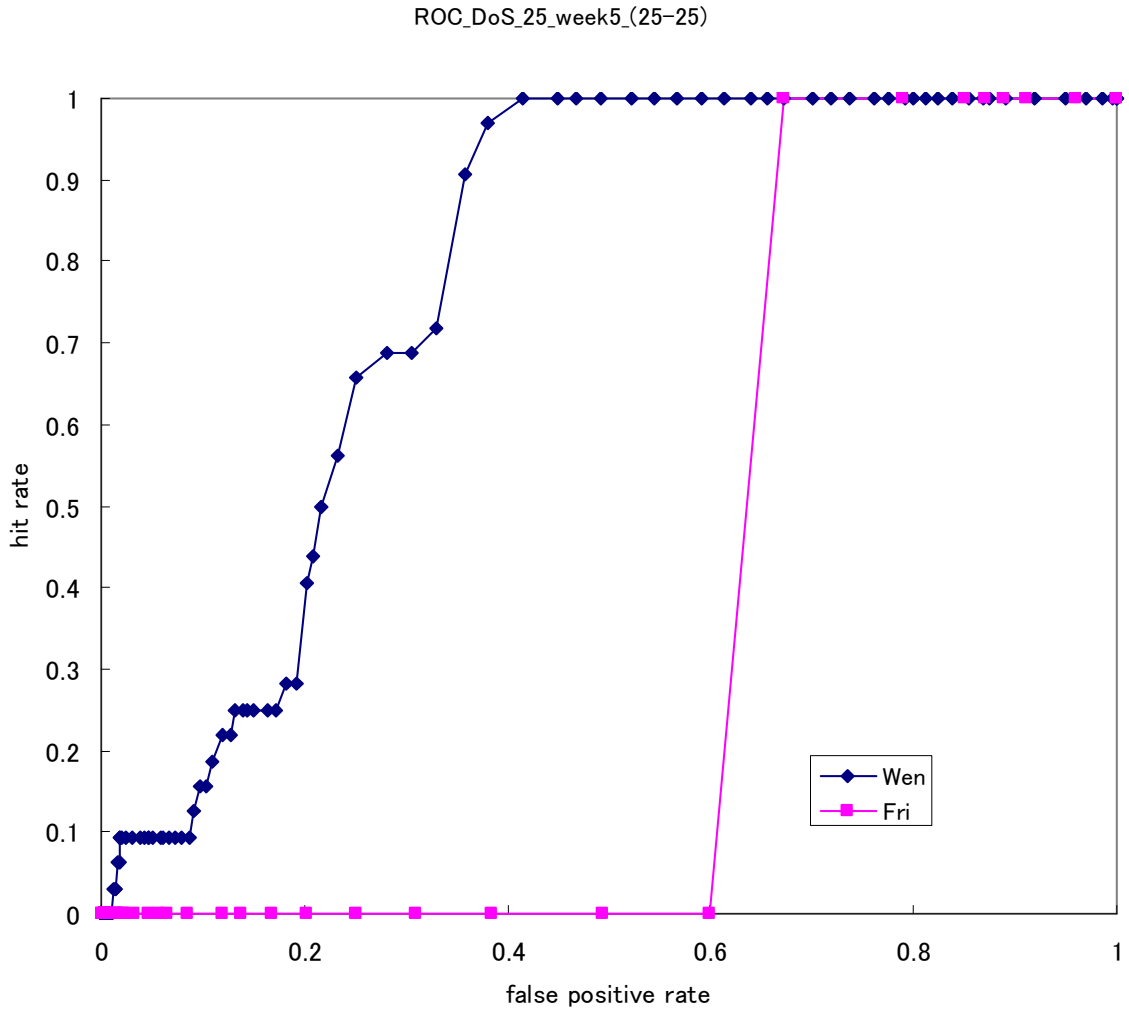

図 4.6　SMTP に対する DoS 攻撃検出結果（week5）

Fig.4.6　ROC curves of week5 under DoS attack in HTTP.

4.2.4　実験結果 – TELNET における DoS 攻撃検出 [実験 2]

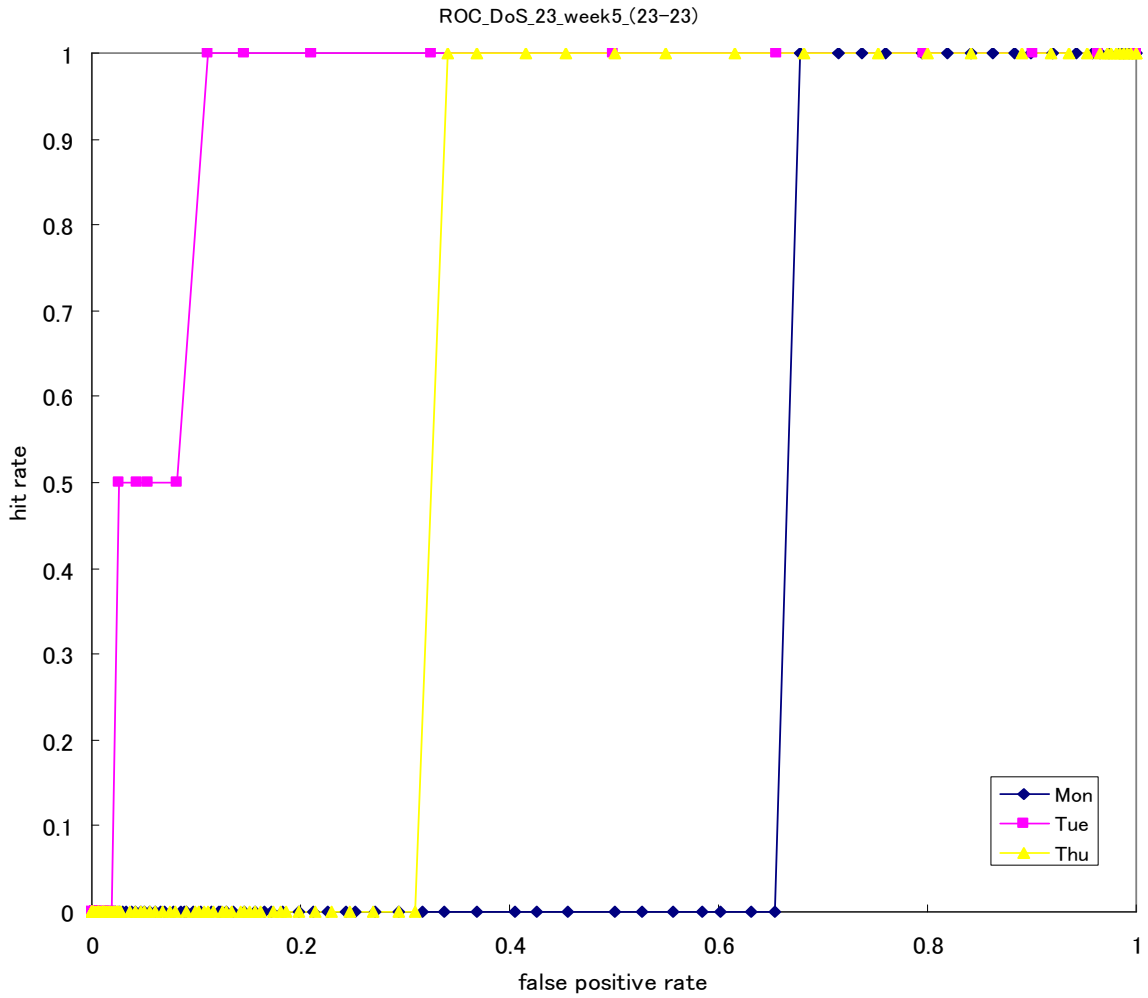

図 4.7　TELNET に対する DoS 攻撃検出結果（week5）
Fig.4.7　ROC curves of week5 under DoS attack in TELNET.

4.2.5　実験結果 – ランダムポートにおける DoS 攻撃検出 [実験 2]

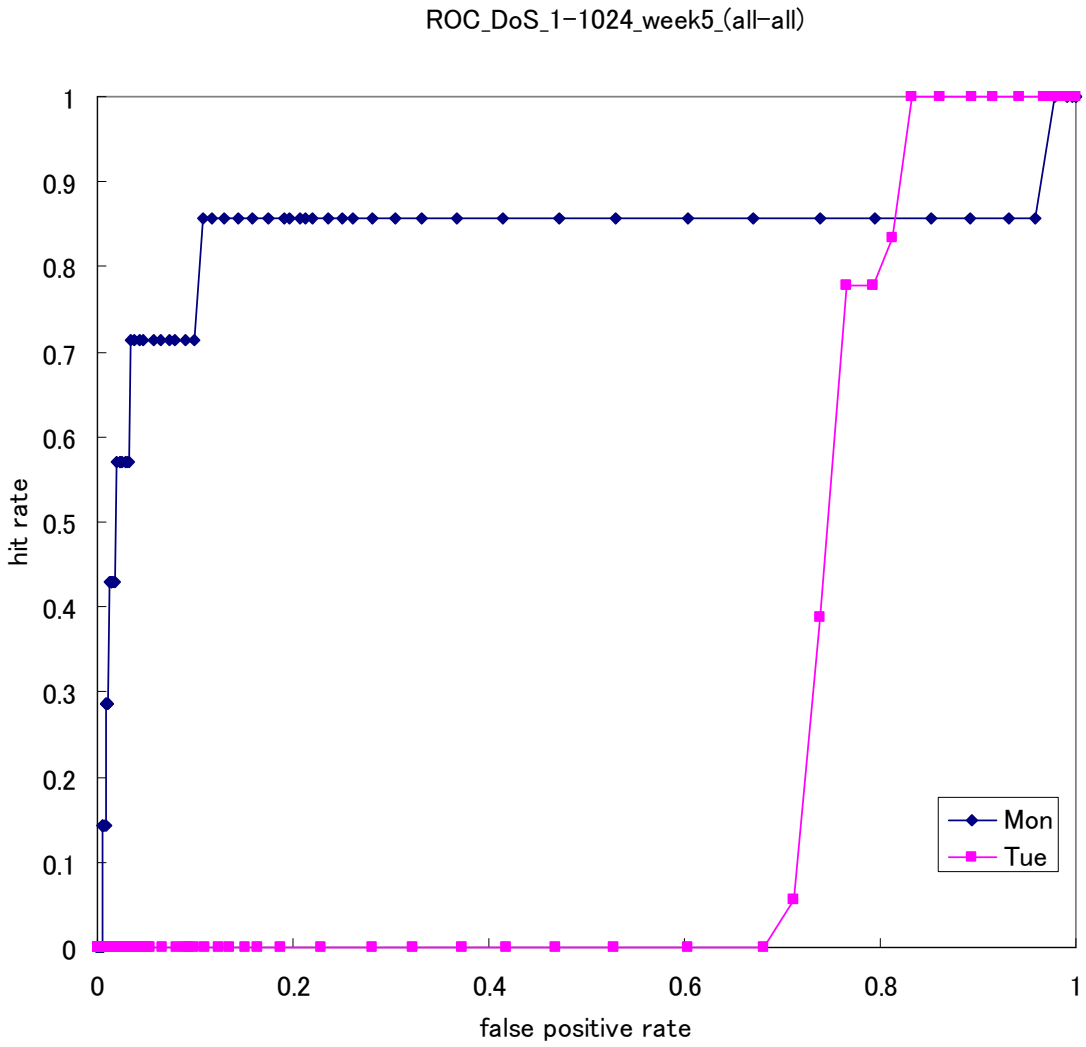

図 4.8　ランダムポートに対する DoS 攻撃検出結果（week5）

Fig.4.8　ROC curves of week5 under DoS attack by random port access.

4.2.6　実験結果 – ランダムポートにおける Prove 攻撃検出 [実験 2]

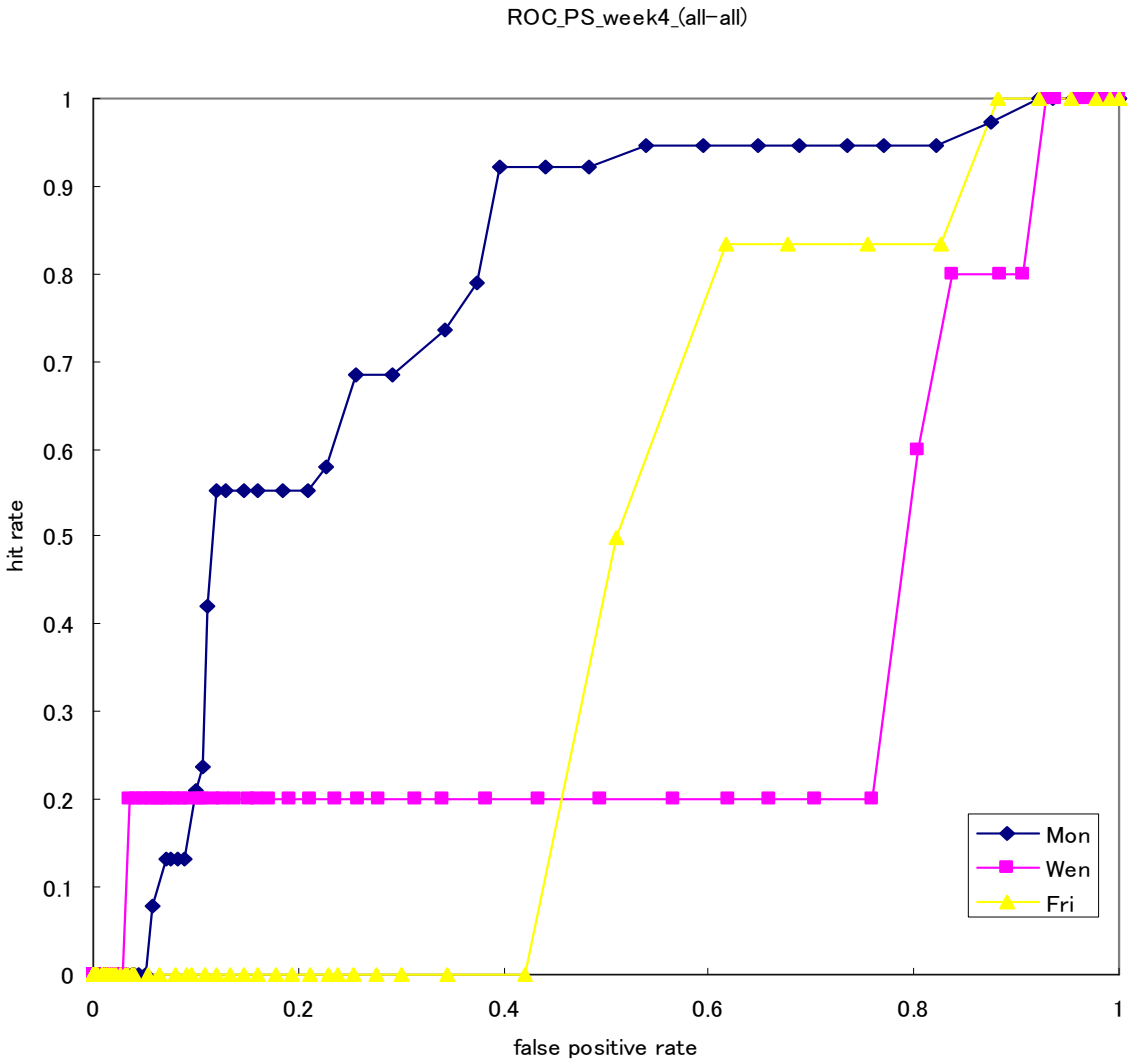

図 4.9　ランダムポートに対する Prove 攻撃検出結果（week4）

Fig.4.9　ROC curves of week4 under Prove attack by random port access.

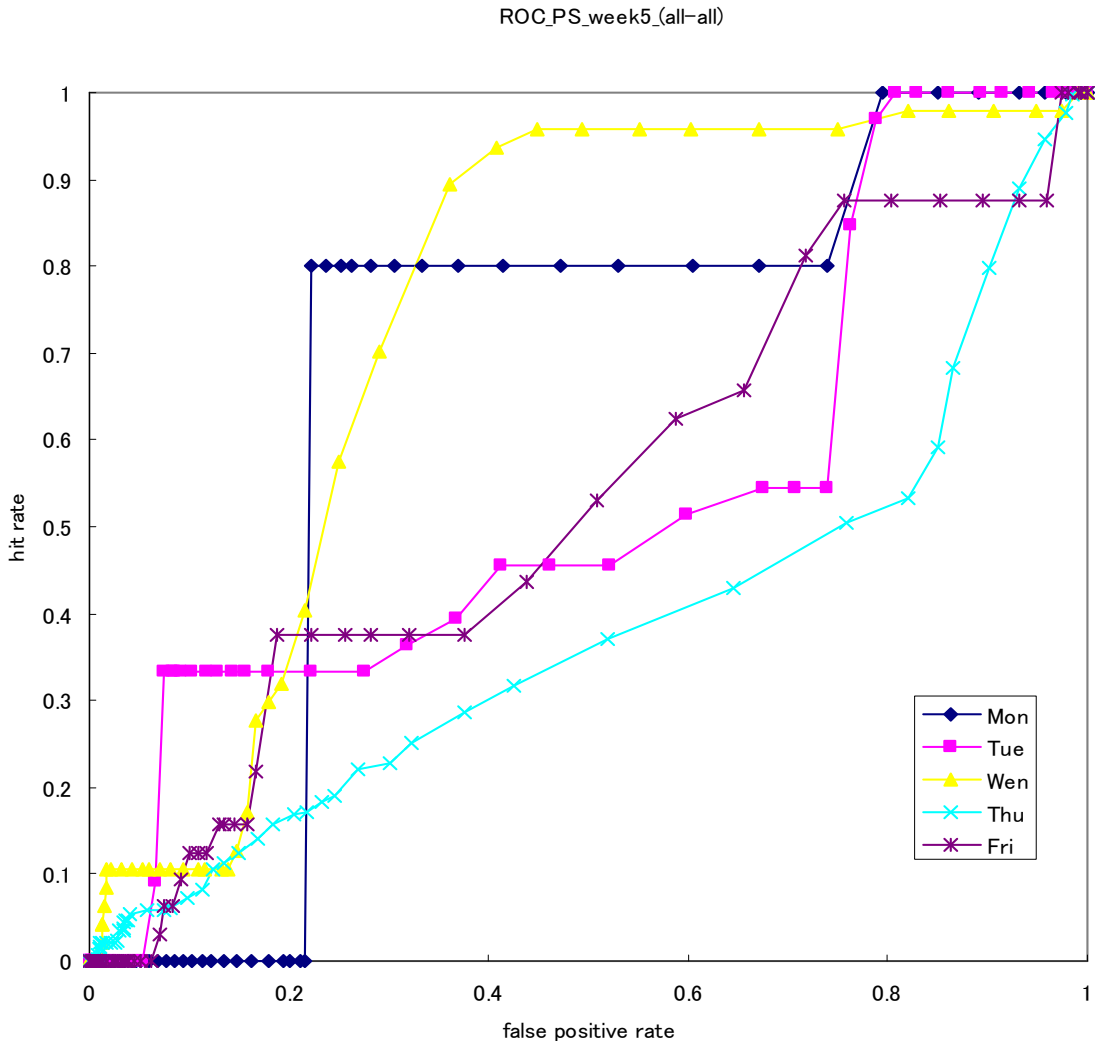

図 4.10　ランダムポートに対する Prove 攻撃検出結果（week5）

Fig.4.10　ROC curves of week5 under Prove attack by random port access.

4.3 考察

実験 1 による手法別の比較実験において，単一の特徴量を用いて異常検出を行う手法より，複数の特徴量の相関関係を用いた手法がよい結果を得られることが分かった．図 4.1 より，誤検出率が 20%程度までは，相関関係を用いた手法よりも単一の特徴量を用いた手法の方が, 検出率が高い場合がある. しかし，全ての DoS 攻撃を検出しようとすると，相関関係を用いた手法は誤検出率を 20%から 30%許す程度で全ての DoS 攻撃を検出できるが，単一の特徴量を用いた手法では 60%または 90%程度になっている．このことから，単一の特徴量を用いた場合では，ある特定の異常事象に対しては多くの特徴を抽出することができるが，全く特徴を抽出できないような異常事象もあり，できるだけ多くの種類の異常事象を検出する目的には適さないことが分かった．それに対して，相関関係を用いた手法では，単一の特徴量を用いた場合と比較して，特定の異常事象に対して多くの特徴を抽出することはできないが，より多くの種類の異常事象を検出することができている．また，複数の相関関係を用いて異常を検出する[手法 4]では，今回の全ての実験データにおいては，1 つの相関関係を用いた[手法 3]よりも良い結果が得られたが，より多くの異常事象の特徴を抽出できる半面，通常状態の特徴を誤って抽出してしまう可能性も高くなるため，誤検出率も増加していく危険性が考えられる.

実験 2 において，図 4.2 の HTTP に対する DoS 攻撃検出結果では，week4 では 10%誤検出率を許すと，すべての DoS 攻撃を検出することができている．しかし，図 4.4 の week5 での検出結果は，1 パケットによる DoS 攻撃を除外しても, 全ての DoS 攻撃を検出するのに 60%程度の誤検出率が発生する. また, HTTP, SMTP，TELNET，ランダムポートに対する Prove 攻撃，DoS 攻撃に共通した手法の問題点として，機械的に大量の通信を行うような DoS 攻撃の場合，送信先 IP やパケットサイズが同一になりやすく，本手法による特徴量（B）において特徴を抽出することができるが，パケットサイズにばらつきがある場合や，特定の送信先 IP ではなく，不特定の送信先 IP に対する DoS 攻撃であった場合は，抽出することができていない. そのため, すべての DoS 攻撃を検出するために，多くの誤検出をしている場合があった．さらに，1 パケットの DoS 攻撃も含んだ場合の HTTP に対する検出結果では，week5 の Thursday，Friday には DoS 攻撃の一種である crashiis と呼ばれる不正アクセスが多く含まれていたが，同じ DoS 攻撃ではあるが検出結果は良好ではなかった．crashis は，サーバの脆弱性をついた攻撃であり，80 番ポートに奇形の GET request を送信する．”GET ../..” のような GET request を送信することによってサーバをクラッシュさせる．この DoS 攻撃は，パケットを 1 つ送信するだけで，サーバをクラッシュさせること

ができるため，統計的手法で扱った場合，特徴を抽出することが困難である．そのため，このような DoS 攻撃を検出するためには，特徴量としてペイロードの一部をパラメータとして用いることを検討する必要がある．

5. むすび

本論文では，ネットワークトラフィック情報から複数の特徴量を時系列データとして抽出し，特徴量間における相関特性から相関係数時系列データを作成し，相関係数が急激に変化した場合に異常を検出する手法を提案した．従来手法において，単一の特徴量を用いた場合では検出困難であった異常事象の特徴を，2 種類以上の特徴量間の相関関係を用いることで抽出することができた．また，DoS 攻撃など，大量の通信が生じた場合の通常状態と異常状態の区別が困難である不正アクセスに対して，ヘッダ情報やパケット数の関係から異常である特徴を抽出することができた．しかし，誤検出率がまだ高いため，単体のセキュリティソフトとして使用するには不適当であり，セキュリティシステムの一部として，正常なパケットを判別するフィルタとしての使用などが考えられる．

今後の課題として，パケット数が少ない場合の例外処理や，変化点検出において，過去のデータから学習する際に行う時系列データの平滑化処理や，相関係数を求める際に設定する窓サイズによって生じる検出遅延を考慮に入れた閾値設定などが挙げられる．

謝辞

本研究の遂行にあたり,多大なる御指導を賜わりました大阪府立大学大学院工学究科電気・情報系専攻汐崎陽教授に心から感謝致します.

そして,本論文をまとめるにあたり,種々の御指導を賜わりました,荻原昭夫准教授,丸岡玄門講師,岩田基助教に深く感謝の意を表します.また,常日頃より有益な御指導を賜り,研究全般や本論文に関しまして直接御指導を賜りました宮本貴朗教授,青木茂樹講師に心より感謝致しますとともに,厚く御礼申し上げす.

さらに,常に激励と適切なるご教示を戴きました情報第 5 グループの大学院生,学部生の皆様に深く感謝致します.特に,常日頃より研究上有益な御指導,御意見を戴くとともに,お世話になりました,片岡祥啓氏,松本駿氏,武田新之助氏,に改めて感謝致します

平成 20 年 3 月 7 日

参考文献

[1] J. Takeuchi and K. Yamanishi, "A Unifying Framework for Detecting Outliers and Change Points from Time Series", IEEE transactions on Knowledge and Data Engineering, vol.18, no.4, pp. 482-492, 2006.

[2] 宮本雅人, “自己組織化マップを用いたネットワークトラフィックからの異常検出”, 大阪府立大学大学院工学研究科 電気・情報系専攻 情報工学分野 修士学位論文, 2006.

[3] MIT Lincoln Laboratory, “DARPA Intrusion Detection Evaluation”, http://www.ll.mit.edu/IST/ideval/