

修士論文

サポートベクターマシンを利用した
不正侵入検出について



平成 23 年度修了
三重大学大学院 工学研究科
博士前期課程 情報工学専攻

中野 孝彦

はじめに

近年、インターネットが LAN や社会的な基盤として重要なものとなっている。それに伴い、コンピュータシステムへの不正アクセスなど、コンピュータセキュリティに関する問題が発生している。その対策の一つとして侵入検知システム (IDS) が広く用いられている。現在使用される IDS の大半はシグネチャ型の IDS である。これは既知の攻撃パターンを IDS に登録し、パターンマッチングによって攻撃を検出する手法であるため、未知の攻撃を検出することができない。そこで現在注目されているのが機械学習型 IDS である。機械学習型 IDS は機械学習で通常パケットの特徴や攻撃パケットの特徴などを学習させて IDS に登録しておき、ネットワーク上のパケットと IDS 内の特徴の類似性から攻撃を検出する。このことから機械学習型 IDS は未知の攻撃に対処できる一方、誤検出が発生してしまう問題も発生する。

本研究では、機械学習アルゴリズムの一つのサポートベクターマシン (SVM) を利用して機械学習型 IDS を構成し、未知の攻撃に強く誤検出の少ない IDS を構成することを目的とする。IDS の機能として、通常のパケットを攻撃パケットと判断してしまうフォールスポジティブよりも、攻撃パケットを通常パケットと判断するフォールスネガティブを如何に減少させるかが重要である。本研究では SVM で通常パケットと攻撃パケットからそれらを分離する学習モデルを作成する際に、通常パケットの重みを軽くして学習させることでフォールスネガティブを減少させる手法を提案する。また、学習に使用する通常パケット数と攻撃パケット数が大きく異なる場合、SVM で過学習を引き起こす可能性がある。そこで上記で提案した手法に加え、学習用の通常パケット数と攻撃パケット数のバランスを調整して学習させる手法を提案する。以上の 2 点から、より判定精度の良い IDS の構成手法を提案する。

本手法の有効性を確かめるために KDDCup1999 Dataset を用いて実験し、既存研究との比較を行った。その結果、通常パケットの判定精度が 99.90%、攻撃パケットの判定精度が 99.75% と、ともに既存研究の結果よりも向上することができた。従って本研究の提案手法の有効性が確認された。

目次

はじめに	i
第 1 章 序論	1
1.1 背景	1
1.2 本論文の構成	2
第 2 章 不正アクセス	3
2.1 不正アクセスの種類	3
2.1.1 侵入 (Intrusion)	3
2.1.2 サービス妨害 (Denial of Service)	3
2.1.3 情報盗用 (Information Theft)	4
2.2 不正アクセスの手順	5
2.2.1 偵察段階	5
2.2.2 攻撃段階	7
2.2.3 占拠段階	8
2.3 ファイアウォール	9
第 3 章 侵入検知システム (Intrusion Detection System: IDS)	10
3.1 IDS	10
3.2 IDS の分類	10
3.2.1 入力データによる分類	10
3.2.2 検出方法による分類	12
3.3 IDS の評価方法	13
3.4 既存研究	14
第 4 章 提案手法	16
4.1 SVM	16
4.1.1 ソフト マージン法	17

4.1.2 非線形 SVM: カーネルトリック	18
4.2 提案手法	18
4.2.1 学習用データのバランスを調整し学習させる手法	19
4.2.2 通常パケットの重みを変更し学習させる手法	19
第5章 実験と考察	20
5.1 KDDCup1999 Dataset	20
5.2 実験	23
5.2.1 学習用データ数を変更した実験	24
5.2.2 重みパラメータを変更した実験	25
5.2.3 学習用データ数, 重みパラメータを変更した実験	26
5.2.4 既存研究の結果との比較	28
第6章 結論	29
謝辞	30
参考文献	31
付録	32
A 提案手法による全データ実験	32
A.1 normal - attack 分類実験	32
A.2 One-to-One を用いた分類実験	34
A.2.1 One-to-One	34
A.2.2 実験結果	35

第1章

序論

1.1 背景

近年、コミュニケーションや情報収集の手段としてインターネットが利用されるなど、インターネットは発展すると共に爆発的な普及を見せた。現在では、社会的な基盤としてインターネットは重要なものとなっており、インターネットの人口普及率は2010年末で78.2%と今もなお増加傾向を示している[1]。1997年末から2010年末までの国内のインターネット利用者数と人口普及率の推移を図1.1に示す。

しかしその一方でコンピュータシステムやネットワークへの不正アクセスやサイバーテロなどインターネットを利用した悪質な行為が増加している。2011年度7月から12月までのIPA(情報処理推進機構)[2]への不正アクセスに関する届出は326件にのぼり、そのうち108件は実害を伴ったものであった。IPAへの届出だけでもこれだけの数の不正アクセスが存在するが、実際にはユーザが気づかないだけでもっと多くの不正アクセスがあることが考えられる。そのためセキュリティ技術の研究が重要視されるようになり、現在様々なセキュリティ技術の研究が進められている。

不正アクセスの対策として、ファイアウォールと共に侵入検知システム (Intrusion Detection System: IDS) が用いられる。IDSには大別してシグネチャ型と機械学習型が存在するが、現在主に用いられているものがSnort[3]等のシグネチャ型のIDSである。シグネチャ型のIDSは既知の攻撃のパターンを登録し、パターンマッチングによって攻撃を検出する手法をとるため、未知の攻撃に対処できないといった問題点が存在する。そこで近年では機械学習型のIDSの研究が進められている。機械学習型のIDSは、通常パケットや攻撃パケットの特徴などをIDSに登録し、ネットワーク上のパケットとIDS内の特徴との類似性などによって攻撃を検出する。このことから未知の攻撃に対処することが可能となる一方、シグネチャ型のIDSに比べ検出率や誤検出の面が実用への障害となっている。

本研究では、未知の攻撃を検出できる機械学習型を用いる。機械学習アルゴリズムの一つであるサポートベクターマシン (Support Vector Machine: SVM) を用い、通常パケットと攻撃パケットを分離するモデルを作成することで機械学習型のIDSを構成し、評価実験を行う。

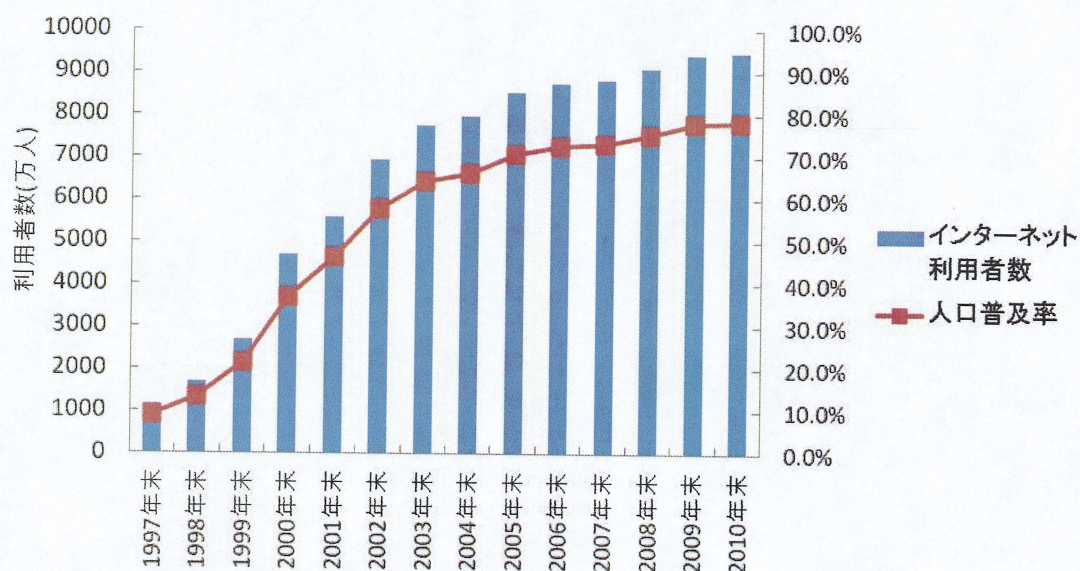


図 1.1 インターネット利用者数と人口普及率の推移

1.2 本論文の構成

本論文の構成は以下の通りである。

第2章では、不正アクセスの種類や手順など情報セキュリティの問題の一つである不正アクセスについて述べる。

第3章では、IDSについて解説し、関連研究を述べる。

第4章では、本研究でデータのモデル化に使用する SVM について解説し、提案手法となる SVM を用いた侵入検出手法について述べる。

第5章では、提案手法による実験とその評価について述べる。

第6章では、結論を述べる。

第2章

不正アクセス

本章ではコンピュータセキュリティにおける不正アクセスについて述べる。まず不正アクセスの種類を述べ、次に不正アクセスの手順について述べる。最後に最も一般的な不正アクセスへの防御策であるファイアウォールについて述べる。

2.1 不正アクセスの種類

一般に知られる不正アクセスの種類は侵入、サービス妨害、情報盗用の3つのタイプに分類することができる。以下でその3つについて述べる。

2.1.1 侵入 (Intrusion)

侵入とは、本来認められていないシステムに対するアクセス権限を不正な方法によって取得し、システムを操作することを指す。これは、外部から不正にシステムの利用権限(アカウント)あるいは管理権限を取得するだけでなく、内部ユーザが本来認められている以上の権限を不正に取得する場合も含む。不正な侵入がきっかけとなりリソースの不正な使用や情報盗用などの被害が発生する危険もある。

2.1.2 サービス妨害 (Denial of Service)

インターネットで用いられる技術の多くは幅広いサービスの要求に応えるように設計されているが、本来システムが想定していなかった使われ方がされた場合に、サービス不能や能力低下の状態に陥る場合が存在する。このような状態を意図的に引き起こす行為をサービス妨害攻撃 (Denial of Service attack: DoS 攻撃) と呼ぶ。代表的なサービス妨害攻撃の手法として ICMP フラッド攻撃と SYN フラッド攻撃がある。以下でその2つの攻撃手法を簡単に説明する。

ICMP フラッド攻撃は、IP レベルで通信相手の到達性を確認する ping コマンドで使用される ICMP Echo Request という制御用パケットを、攻撃対象に大量に送信する攻撃である。攻撃対象のシステムは ICMP リクエストへの応答のために処理能力を使いきってしまい、他のパケットを処理不能となる。

SYN フラッド攻撃は、攻撃者がサーバに対して大量の TCP 接続応答待ち状態を作り出させる攻撃である。TCP 接続では最初に 3WAY ハンドシェークを実行する。SYN フラッド攻撃は、攻

撃側がこの接続手順を途中でわざと止め、ハーフ・オープンという状態を作る行為を何度も繰り返す。サーバは接続途中の状態を保ったまま、次々と新しい接続要求を処理することになり、やがて応答待ちの接続数がサーバの限界を超えることで接続を受け付けられなくなる。

また、踏み台と呼ばれる複数のホストが、標的となるサーバなどに対していっせいに攻撃を行うタイプの攻撃を分散型サービス妨害攻撃 (Distributed Denial of Service attack: DDoS) と呼ぶ。DDoS 攻撃を図 2.1 に示す。踏み台とは、放置されたセキュリティホールを利用され、不正アクセスなどによって攻撃用プログラムをシステム内に組み込まれたホストのことである。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拒否すればよいが、数千、数万のホストからの攻撃は個々に対応することが難しい。そのため、通常の DoS 攻撃よりも防御が困難であり、攻撃による被害は DoS 攻撃よりも大きくなる。

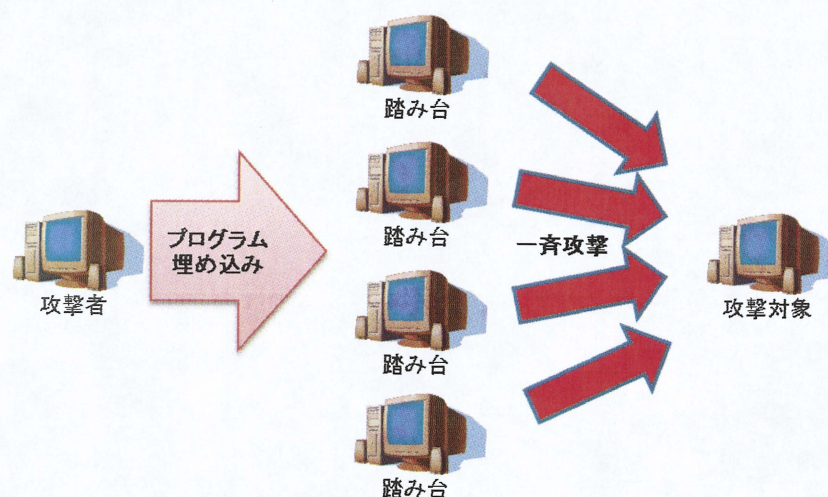


図 2.1 DDoS 攻撃

2.1.3 情報盗用 (Information Theft)

侵入者はシステムへのアクセス権限を取得したあと、ファイルシステム上のデータを不正に閲覧、転送したり、スニッファと呼ばれるネットワーク上に流れるパケットを全て読み取るプログラムを設置してネットワーク上の情報の盗聴を行ったりする。また、内部ユーザによる情報の漏洩の危険性も存在する。通信経路上のネットワークに多数のシステムが介在する場合には、各接続点や経路ネットワークのそれぞれから情報が盗まれる危険があるため、情報盗用の危険性が高くなる。情報盗用への対策として、電子メールの暗号化や VPN (Virtual Private Network) によるパケットレベルでの暗号化などが行われる。

2.2 不正アクセスの手順

不正アクセスの手順は様々なものが存在する．ここでは不正アクセスが偵察，攻撃，占拠の3つの段階を経て進められるとし，その各プロセスについて説明をする．

2.2.1 偵察段階

偵察段階において，侵入者は様々な情報源を駆使して目標となるホスト，ネットワーク，サービスあるいはユーザに関する情報を手に入れようとする．これにはネットワーク上で公開されている情報だけでなく，現実社会における情報まで幅広い情報源が利用される．そのため，システム管理において月に何千件もの攻撃を受けるという報告も見受けられるが，そのほとんどが偵察段階における調査活動のアクセスであるということも多い．

実際，攻撃者が意図的に侵入を試みる場合には，いきなり攻撃を開始するということではなく，事前に偵察行為が行われると考えられる．つまり，このような偵察行為はそれに続く攻撃の兆候として考えられる．偵察行為の主な手法を以下に示す．

公開情報の収集

最も簡単な偵察行為は，公開されている情報源から有益なじょうほうを手に入れることである．公開情報には，攻撃対象自身が Web サーバ上に公開している情報，ネームサーバやメールサーバなどが提供する情報，IP アドレスやドメインを管理する NIC(Network Information Center) が公開している情報などが存在する．また，個人的に使用しているメールアドレスや Web ページだけでなく，組織名や組織の活動を手がかりとして得られるあらゆる情報がセキュリティ侵害に利用される可能性がある．これらの情報は，一般に公開されることがサービスの本来の目的であるために，悪意のある一部のユーザだけに見えなくすることは難しい．

スイープ (ホスト スキャン)

スイープ (Sweep) あるいはホスト スキャン (Host Scan) とは，ネットワークに接続されて稼働状態にある機器の IP アドレスを調査する行為である．一定の範囲のアドレスに対して連続的にパケットを送信し，その反応を確認することでどのようなアドレスを持つホストが稼働中であることを確認する．最もシンプルな方法が ICMP プロトコルを用いた ping スイープ (ping Sweep) であるが，同じ目的のために上位層であるトランスポート層のプロトコルでスキャンが行われることもある．また，スイープの対象として公開情報によって特定されたアドレス帯が目標として使用されることもある．

ポートスキャン

ポートスキャンは、ネットワークに接続されたホスト上で、どのようなサービスが提供されているかを調査する行為である。すなわち、対象ホスト上の単一あるいは複数のポートに対してアクセスを試みる行為を指す。

TCP や UDP といった現在普遍的に用いられるプロトコルで提供されるサービスは、サーバ側とクライアント側にそれぞれ1つずつポートと呼ばれる仮想的な窓口を使用している。ポートは1から65535までの整数値の番号を持ち、1から1023までが Well-known port として IANA(Internet Assigned Number Authority) によって管理されている。1024 から 49151 までは登録済みポートとして、管理はされていないが登録リストの公開が行われている。49152 以降はダイナミックポートと呼ばれ、主にクライアントのための一時的なポート番号として使用されている。代表的なサービスに割り当てられたポート番号を表 2.1 で示す。

表 2.1 主要サービスに使用されるポート番号

ポート番号	プロトコル	サービス名	
20	TCP	FTP(Data)	File Transfer Protocol(Data)
21	TCP	FTP(Control)	File Transfer Protocol(Control)
22	TCP/UDP	SSH	Secure Shell
23	TCP	telnet	
25	TCP/UDP	SMTP	Simple Mail Transfer Protocol
43	TCP	WHOIS	
53	TCP/UDP	DNS	Domain Name System
67	UDP	DHCP(Server)	Dynamic Host Configuration Protocol
68	UDP	DHCP(Client)	Dynamic Host Configuration Protocol
80	TCP/UDP	HTTP	Hypertext Transfer Protocol
110	TCP	POP3	Post Office Protocol Version 3
119	TCP	NNTP	Network News Transfer Protocol
123	UDP	NTP	Network Time Protocol
143	TCP/UDP	IMAP	Internet Message Access Protocol
443	TCP/UDP	HTTPS	Hypertext Transfer Protocol Security

管理者がネットワーク上でどのようなサービスが稼動しているか実際の状況を確認するためにポートスキャンを実行する場合も存在するため、悪意のある者だけが実行すると限定できない。ポートスキャンの代表的なツールとしては Nmap が存在する。

通常、自分のネットワークに対してポートスキャンが行われているかの確認はアクセスログから可能であるが、ログが残らないように特殊なパケットを使用するステルススキャンや、低速にスキャンを行うスロースキャンなどが行われる場合が存在する。この場合には、通常のログには記録されないか、記録されたとしてもスキャンと認識することが困難である。

脆弱性スキャン

脆弱性スキャン (Vulnerability Scan) は、提供されるサービスのうちセキュリティホールが存在が知られているものについて、そのセキュリティホールへの攻撃が有効かどうかを調査するために行われる行為である。

脆弱性あるいはセキュリティホールとしては、リモートからの不正な操作が可能となる危険なサービスの存在や、バッファオーバーフローの存在などがある。このようなセキュリティホールに対するアクセスを順次繰り返すプログラム、スクリプトなどを用いるのが脆弱性スキャンの一般的な手口である。

2.2.2 攻撃段階

偵察段階において攻撃対象に関する情報を十分に入手した後に、侵入者は実際に不正を働くための攻撃段階へと移る。攻撃段階における手法を以下に述べる。

仕様上の弱点に対する攻撃

ソフトウェアやプロトコルなどのシステムの仕様上からは悪用が想定されていなかったような部分のセキュリティホールを狙った攻撃のことを指す。インターネットは信頼関係を前提としたオープンな技術として発展してきた。そのため、プロトコルに対する攻撃が想定されてなく、IPアドレスの詐称や、過剰な負荷によるサービス妨害などに対する基本的な防御方法が存在しない。

プログラムの弱点に対する攻撃

プログラムの実装やインストールなどの問題に対する攻撃のことを指す。プログラムの実装段階におけるバグや、プログラマがデバッグ用に残しておいたコードなどに対して攻撃が行われる。

特に、任意の機械語コードを実行することを可能とするバッファオーバーフローの問題は、C言語などの仕様起因する実装上の問題であり、問題点が存在することが発覚したならば、安全なプログラムにバージョンアップする他に効果的な防御手段がない。

設定の不備に対する攻撃

日常の業務で忙しいネットワーク管理者はネットワークに接続されたホストの一つ一つの設定状

態を確認する余裕がないことがある。実際の状況として、世界中にある多くのネットワークに接続されたマシンは、設定が適切でないまま放置されている。侵入者は、このような放置されたサービスのバグなどを利用したり、デフォルト状態のまま放置されているサービスなどからシステムに侵入したりする。

パスワードクラック

パスワードクラックとは、パスワードを用いた認証機器に対して、使用されるパスワードを予測、解読することで、認証メカニズムを攻撃することを指す。パスワードクラックに成功した侵入者は正当なユーザになりすまし、システムへのアクセスが可能となる。

2.2.3 占拠段階

システムへのアクセスが可能になった侵入者は、取得したアカウントの権限レベルに応じたシステム操作が可能となる。占拠段階で想定される活動を以下に述べる。

情報の盗用

侵入したシステム上のファイルから得られる情報を盗み見たり、ファイルの転送、スニッファープログラムによるトラフィックの盗聴などが行われることが想定される。また、盗聴された内容は、ローカルディスク内に格納されたり電子メールなどの手段で自動的に外部に転送されたりすることがある。

情報の改ざん

ホスト上のファイルの内容を改ざんしたり、Web ページの内容を書き換える愉快犯的なもの、重要な取引上のデータを改ざんすることで経営上の被害を発生させるものまで存在する。また、侵入の事実を隠蔽して追跡を回避するためにログの改ざんや消去なども行われる。

ホストの不正使用

システムを占拠した侵入者はそのホストをコントロールすることができる。そのため、占拠したシステムを別のシステムへの侵入のための踏み台として使用するなど多様な用途に用いることが考えられる。代表的な例を以下に述べる。

- スпамメールやメール爆弾の発信源としての利用
- パスワードの解析用端末としての利用
- DDoS 攻撃などの踏み台としての利用

2.3 ファイアウォール

ファイアウォールは外部ネットワークと内部ネットワークの間の通信を制限し、内部ネットワークの安全を維持することを目的としたシステムである。ファイアウォールの防御方法として、パケットフィルタリングとアプリケーションゲートウェイの2種類の仕組みが存在する。

パケットフィルタリング

パケットフィルタリングとは、ファイアウォールを通過する IP パケットの内容を調査し、パケットの通過を許可あるいは拒否する機能である。IP パケットの通過の判断は、IP ヘッダや TCP/UDP ヘッダといったネットワーク層、トランスポート層の情報に基づいて行われる。

パケットフィルタリングには静的なフィルタリングと動的なフィルタリングが存在する。以下でそれらについて簡単に述べる。

- 静的フィルタリング

静的フィルタリングは宛先や送信元の IP アドレス、ポート番号などを監視し、あらかじめ設定した条件によって、通信の許可、廃棄、拒否などの動作で通信の制御を行う。仕組みが単純なため高速に動作するが、設定に手間がかかることや、防ぎきれない攻撃が存在することなどの問題点がある。

- 動的フィルタリング

動的フィルタリングは宛先や送信元の IP アドレス、ポート番号などの条件を、IP パケットの内容に応じて動的に変化させて通信制御を行う方式である。動的フィルタリングの一種としてステートフルインスペクションという技術も存在し、セッション単位での通信管理が可能となっている。

アプリケーションゲートウェイ

アプリケーションゲートウェイとは、通信を中継するプロキシプログラムを使用し、アプリケーション層で通信の制御を行う方式である。具体的には、IP アドレス、ポート番号、個々のアプリケーションに依存した情報を用いるため、各アプリケーションプロトコルごとに個別のゲートウェイ・プログラムが必要となる。そのかわり、きめ細かなアクセス制御が可能となり、プロトコルに基づくログの取得もできるようになる。

また、パケットフィルタリングと比べると処理速度が遅いといわれるが、実用上はほとんど問題なく使用することが可能である。

第3章

侵入検知システム (Intrusion Detection System: IDS)

前章では、不正アクセスとその対策として使用されるファイアウォールについて述べた。しかし不正アクセスの対策としてファイアウォールだけでは十分とは言い切れず、多くの場合ファイアウォールと共に侵入検知システム (IDS) が使用されている。

本章では、まず本研究の研究対象である IDS について簡単に述べる。その後 IDS の種類について説明し、IDS の評価方法や現状について述べる。最後に本研究に関連のある既存手法について述べる。

3.1 IDS

IDS とは侵入検知システム、または不正アクセス監視システムなどと呼ばれ、ネットワークへの不正侵入を検知する技術であり、不正侵入が増加している昨今、ファイアウォールと共に重要性が増している技術である。

IDS はファイアウォールなどの他のセキュリティ技術で防げなかったような通信を監視し、異常を検出して管理者に通知したり記録したりする技術である。すなわち、ネットワークセキュリティに関してネットワーク管理者をサポートする技術となる。

3.2 IDS の分類

IDS は、入力として使用するデータによる分類と、検出方法による分類が存在する。

3.2.1 入力データによる分類

入力として使用するデータによってネットワーク型とホスト型に分類される。ネットワーク型とホスト型の IDS を図 3.1 で示し、以下でそれぞれの説明を述べる。

ネットワーク型

ネットワーク型侵入検知システム (Network Intrusion Detection System: NIDS) は、接続したネットワーク上のパケットデータを入力データとする IDS である。NIDS には、自ホスト宛の

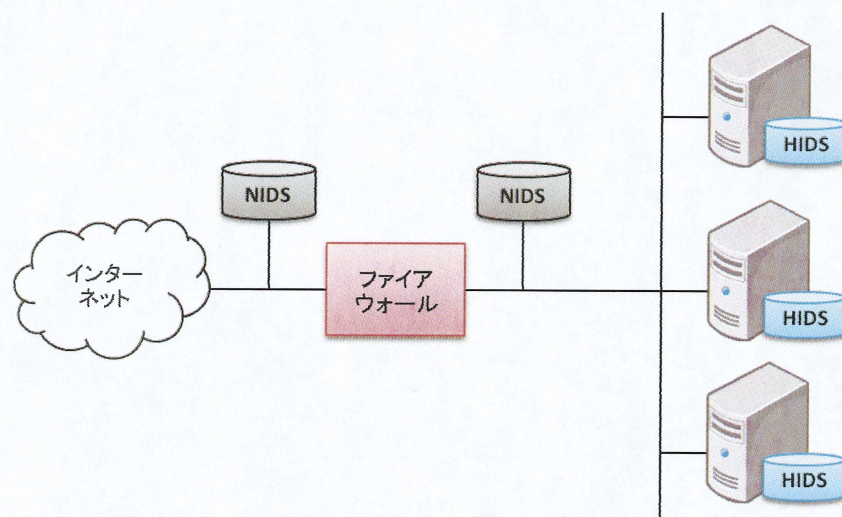


図 3.1 NIDS と HIDS

パケットのみを監視するものと、同一ネットワーク内の他ホスト宛のパケットを含む全トラフィックを監視するものの2つのタイプが存在し、現在では後者のタイプが主流となっている。また、NIDSは一般的にNICにIPアドレスを割り振らずに運用するステルスモードで運用されるため、外部からIDSを認識されることがない。しかし、NIDSはネットワーク上のパケットを監視しているため、そのパケットが暗号化されている場合検知することができない。

ホスト型

ホスト型侵入検知システム (Host Intrusion Detection System: HIDS) は、オペレーティングシステムおよびアプリケーションが生成するログデータや、コマンドヒストリなど単一ホストのシステム上で生成されるイベント情報を入力として侵入を検知するIDSである。HIDSの多くはログファイルを入力としている。NIDSが暗号化されたパケットを検知することができないのに対し、HIDSは暗号化パケットを複合したあとに検知することができるという利点が存在する。しかし、HIDSは監視したいホストにインストールする必要があるため、NIDSより運用・管理で手間がかかるといった面も存在する。

3.2.2 検出方法による分類

IDS は検出方法によってシグネチャ型と機械学習型に分類することができる。以下でそれぞれの説明を述べる。

シグネチャ型

シグネチャ型の IDS は、攻撃や不正侵入の際に見られるルール・パターンをあらかじめデータとして登録しておき、実際のデータとパターンマッチングすることで不正を検出する。このあらかじめ登録された情報をシグネチャと呼ぶ。現在の IDS の主流はシグネチャ型であり、製品として Snort などが存在する。

シグネチャ型の IDS は、シグネチャを適切に登録することで誤検出を減少させることができる。しかし、適切なシグネチャの設定にはネットワークやプログラムの高度な知識が必要となり、十分に時間をかけて調査をおこなう必要がある。また、シグネチャとデータをパターンマッチングによって不正を検出するため、登録されていない攻撃を検出することができない。そのため、新しい攻撃が発覚した場合にすぐに対応したシグネチャを追加する必要がある。現在も新しい攻撃は増え続けており、常にシグネチャを作成し続ける必要があるため、管理に多大な労力がかかるという問題点が存在する。

機械学習型

機械学習型の IDS は、統計的手法やデータマイニングの技術を利用し、データを判別するモデルを作成する。作成したモデルとデータの情報から判定を行い、攻撃のデータを検出する。そのため、モデルを作成するための学習用データがあらかじめ必要となる。

機械学習型では、既知の攻撃の情報を利用するため、未知の攻撃でも既知の攻撃と似た特徴を持つ攻撃の検出を行うことができる。新しい攻撃は増え続けるが、既存の攻撃を改変した亜種攻撃など攻撃の目的が同じものや、すでに判明しているセキュリティホールを利用した攻撃などがあるため、既知攻撃と似た特徴を持つことがたびたびある。そのため、未知の攻撃の検出手法として有用である。しかし、通常データの状態の特徴などをモデル化して判定を行うため、既存の攻撃であっても通常データの特徴と類似していれば通常データと判定してしまうなど、誤検出が多くなってしまふことがある。

3.3 IDS の評価方法

IDS の評価には検出精度とスループットが用いられることが多い。中でも検出精度は最も重要な評価項目である。しかし、検出精度を計算するためにはあらかじめデータ内のサンプルのクラスが判明していなければならない、そのクラスを正しく判定することが難しい。そのために KDDCup などのコンテストで使用されたデータを利用することが多いが、多種の環境に応用できる一般的な評価データが存在しないというのが現状である。

検出の際に間違った判定をしてしまうことが誤検出であるが、誤検出はその結果によってフォールスポジティブとフォールスネガティブの 2 つに分類される。

- フォールスポジティブ (False Positive)

フォールスポジティブとは、本来検出すべきではないものを検出してしまうことを指す。IDS などの検出システムにおいては、正常なデータを誤判定で弾いてしまうことである。

フォールスポジティブが多くてもシステムに影響はないが、管理者が調査を行うなど無駄な労力となってしまう。そのため、フォールスポジティブが多すぎると、管理者が報告を信じなくなってしまうため警報を無視する原因へとつながってしまう。

- フォールスネガティブ (False Negative)

フォールスネガティブとは、本来検出したいものが検出されないことを指す。IDS などの検出システムにおいては、不正なデータを誤判定によってそのまま通過させてしまうことである。フォールスネガティブが多いということは、攻撃の検出がほとんどできていないことにつながるため、検出器の意味がなくなってしまう。

一般的にフォールスポジティブとフォールスネガティブはトレードオフの関係にあると言われており、どちらもなくすということは難しい。そのため、どちらも減らすことは重要であるが、IDS ではフォールスポジティブが多いよりもフォールスネガティブが多いほうが問題となるため、フォールスネガティブを減らすことがより重要課題となる。

3.4 既存研究

本研究ではデータの学習・評価に KDDCup1999 Dataset[5] を用いる．そこでここでは KDDCup1999 Dataset を使用している研究を紹介する．

KDDCup1999 コンテストの優勝者である Austrian Research Institute for Artificial Intelligence の Bernhard Pfahringer は, “cost-sensitive bagged boosting algorithm” と呼ばれる手法を使って全体で 50×10 の C5 決定木を構成した [6][7]．この手法における全体のフォールスポジティブ率は 0.55%, フォールスネガティブ率は 8.19% であった．その詳細結果を表 3.1 に示す．

表 3.1 bagged boosting による判定結果

predicted actual						correct(%)
	Normal	Probe	DoS	U2R	R2L	
Normal	60262	243	78	4	6	99.45%
Probe	511	3471	184	0	0	83.32%
DoS	5299	1328	223226	0	0	97.12%
U2R	168	20	0	30	10	13.16%
R2L	14527	294	0	8	1360	8.40%
correct(%)	74.61%	64.81%	99.88%	71.43%	98.84%	

Normal Detection Accuracy = 99.45%

Attack Detection Accuracy = 91.81%

スロベニアの Carolina Fortuna らは線形 SVM を利用して One-to-All 方式, One-to-One 方式, One-to-All-3categ 方式の 3 つの方式でマルチクラス分類を行う IDS をそれぞれ構成した [8]．その結果, 少ない学習用データにおいては One-to-All-3categ 方式, 全てのデータを用いた場合には One-to-One 方式がよく攻撃を検出できているとの検証が行われた．One-to-One 方式を用いたときの全体のフォールスポジティブ率は 2.02%, フォールスネガティブ率は 9.09% であった．One-to-One 方式を用いたときの詳細結果を表 3.2 に示す．

表 3.2 One-to-One SVM による判定結果

actual \ predicted						
	Normal	Probe	DoS	U2R	R2L	correct(%)
Normal	59367	211	818	12	185	97.98%
Probe	901	3002	148	0	115	72.06%
DoS	7047	52	222754	0	0	96.91%
U2R	32	0	0	32	6	45.71%
R2L	14791	11	2	11	1532	9.37%
correct(%)	72.28%	91.64%	99.57%	58.18%	83.35%	

Normal Detection Accuracy = 97.98%

Attack Detection Accuracy = 90.91%

第 4 章

提案手法

本章では、提案手法に利用するサポートベクターマシン (SVM) について述べる。そして提案手法について述べる。

4.1 SVM

SVM は 1960 年代に Vapnik 等が考案した Optimal Separating Hyperplane を起源とし、1990 年代になってカーネル学習法と組み合わせた非線形の識別手法へと拡張された。このことによって、SVM は現在知られている手法の中で最もパターン認識性能の優秀な学習モデルの一つとなっている。SVM は、基本的に 2 クラスのパターン認識器を構成する手法である。このことはつまり、学習パターンに対して最適な識別境界を決定することと同義である。この識別境界を分離超平面と呼び、訓練サンプルから、各データ点との距離が最大となる分離超平面を求めるマージン最大化という基準でパラメータを学習する。

例として、図 4.1 に示すような 2 つのクラスの分類について考える。学習データの中で最も他クラスと近い位置にあるベクトルデータを特徴ベクトルとし、この特徴ベクトルを基準としてマージンが最大になるよう分離超平面を配置する。このときの識別関数を式 (1) で示す。

$$\begin{aligned} f(x) &= \langle w \cdot x \rangle + b \\ &= \sum_{i=1}^n w_i x_i + b \end{aligned} \quad (1)$$

ここで、 w はシナプス荷重に対応するパラメータで、 x は入力ベクトル、 x は入力ベクトル集合、 b はバイアスである。このときの $f(x)$ が分離境界、つまり分離超平面となる。線形分離可能な問題のとき、式 (1) を満たす超平面が複数存在する。そこで、式 (2) で表現される制約を加え、超平面を一意に決める。

$$\min_{i=1, \dots, r} |(w \cdot x_i) + b| = 1 \quad (2)$$

つまり、この制約によって w と b は距離 $1/\|w\|$ を持つ超平面に最も接近するデータ点を表現するパラメータとなり、式 (3) を識別関数とすることができる。

$$f(x) = \text{sgn}((w \cdot x) + b) \quad (3)$$

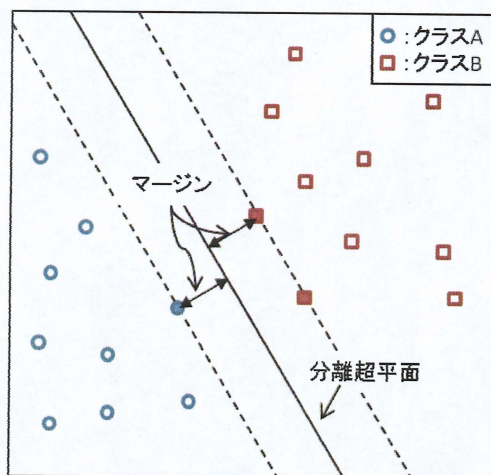


図 4.1 SVMによる2クラスパターン識別

4.1.1 ソフトマージン法

実世界の問題では完全線形分離問題が少ないため、訓練サンプルを完全に分離できる超平面はほとんど存在しない。そこでノイズが存在する実世界の問題に対応するため、ある程度の誤差を許して境界を設定するソフトマージンの概念が存在する。

ソフトマージンを利用した最適化では、マージン最大化を行いながらも、図4.2で示すように、幾つかの標本が境界を超えて反対側に存在することを許す。

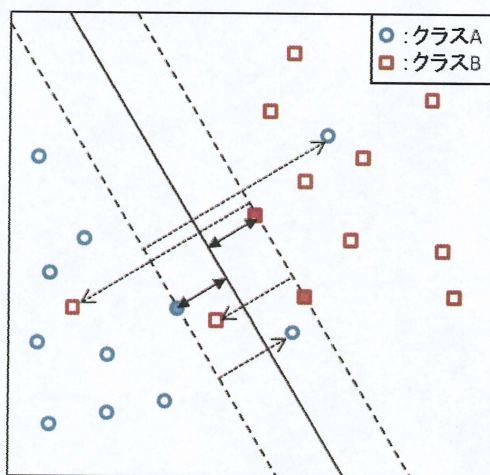


図 4.2 ソフトマージン

4.1.2 非線形 SVM: カーネルトリック

ソフトマージン法を用いることで、線形分離可能でない問題に対してもパラメータを求めることが可能となる。しかし、ソフトマージン法を用いた場合でも、本質的に非線形で複雑な識別問題に対しては、必ずしも良い性能の識別器を構成できるとは限らない。本質的に非線形な問題に対応するための方法として、高次元化が挙げられる。これは、図 4.3 に示すように、非線形写像 Φ によって元の入力データを高次元特徴空間に写像し、特徴空間において線形分離を行うという方法である。このことで、結果的に元の入力空間においては非線形な分類を行なっていることになる。しかし、高次元化を実装するにあたって、 Φ の計算は行わず、カーネル関数の計算に置き換える。これをカーネルトリックという。カーネルトリックによって Φ を直接計算することを避け、計算上の困難を克服し、非線形問題に対応することが可能となっている。

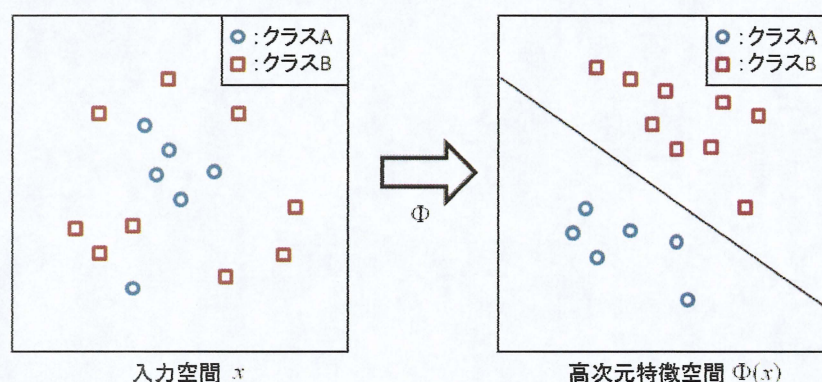


図 4.3 非線形写像による高次元化

4.2 提案手法

本研究では、SVM で通常パケットと攻撃パケットを効果的に分離するモデルを作成するための手法を提案する。なお、本研究では攻撃の分類は行わず、通常パケット以外のデータを全て攻撃とみなし、通常パケットのクラスと攻撃パケットのクラスの 2 種類に分類する。

1 つ目の手法は、学習用のデータとして使用する通常パケットと攻撃パケットの数のバランスをとってから学習させる手法である。

2 つ目の手法は、学習を行う際に通常パケットのクラスの重みを変更して学習させる手法である。以下でそれぞれの手法について述べる。

4.2.1 学習用データのバランスを調整し学習させる手法

学習用データの中には、片方のクラスに属するデータ数がもう片方の何十倍存在するようなデータセットが存在する。そのようなデータセットをそのまま用いて機械学習を行うと、過学習が起こってしまい適切な分類ができなくなる可能性が存在する。

そこで、学習前に学習用データ内のそれぞれのデータ数の差を少なくすることにより、データの過学習をできるかぎり抑える手法を提案する。このことで、過学習による偏った学習結果が抑えられ、検出精度の向上が期待できる。

4.2.2 通常パケットの重みを変更し学習させる手法

前章でも述べたが、IDS はフォールスポジティブを減らすことももちろん重要であるが、フォールスネガティブを減らすことの方がより重要な課題となる。SVM では前述のマージン最大化の考えに基づいているため、理論上では分離超平面が通常パケットと攻撃パケットの間に設置されることとなる。

そこで、本研究では通常パケットの重みを減少させて学習させる手法を提案する。ここで重みとはクラスの重要度を示し、重みが小さいほどデータのばらつきによる影響が軽減され、図 4.4 のように分離超平面の位置が移動すると考えられる。

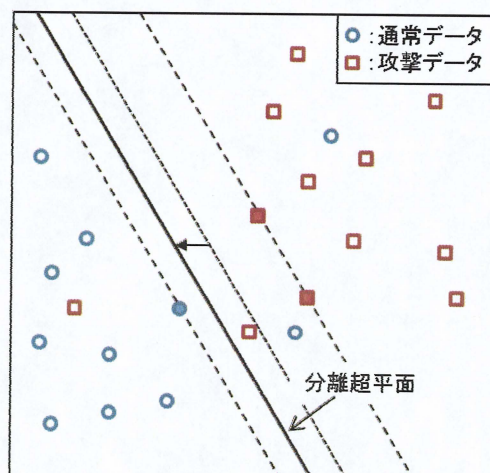


図 4.4 重み変更による分離平面の移動

第 5 章

実験と考察

本実験では、SVM の学習と実験の評価に KDDCup1999 Dataset を用いる。そこで本章では、まず KDDCup1999Dataset について説明し、提案手法を用いた実験と考察を述べる。

5.1 KDDCup1999 Dataset

KDDCup とは、データマイニングの分野で活発に研究活動を行っている ACM(the Association for Computing Machinery) という団体が主催で開催している知識発見とデータマイニングツールに関する国際競技会 (International Knowledge Discovery and Data Mining Tools Competition) のことである。1999 年の KDDCup のテーマはネットワークの侵入検出器の作成であり、与えられたコネクションのタスクの中から通常の接続と攻撃の接続を選別するというものであった。この大会で用意されたデータセットは、軍のネットワーク環境でシミュレートされた多種多様な攻撃を含むアクセスログデータのセットとなっている。不正アクセスの研究において評価用のデータは一般的なものがあまり存在しないため、KDDCup1999 のデータセットが研究用としてしばしば用いられる。

KDDCup1999 Dataset 内の各データは、それぞれ以下の 5 つのカテゴリに大別される。

- **Normal**
通常のコネクションを表すクラス。
- **Probe**
攻撃対象の監視や調査を行うアクセスに関するコネクションを表すクラス。
- **DoS**
サービス妨害攻撃のコネクションを表すクラス。
- **U2R**
スーパーユーザ (root) 権限への許可されていないアクセスに関するコネクションを表すクラス。
- **R2L**
リモートマシンからの無許可のアクセスに関するコネクションを表すクラス。

また、それぞれの攻撃はさらにサブクラスに分類することができ、データセット内のデータには属性とともにサブクラスが割り当てられる。それぞれの属性のサブクラスを表 5.1 に示す。

表 5.2 データの特徴

FeatureName	Description
duration	length (number of seconds) of the connection
protocol.type	type of the protocol, e.g. tcp, udp, etc.
service	network service on the destination, e.g. http, telnet, etc.
flag	normal or error status of the connection
src.bytes	number of data bytes from source to destination
dst.bytes	number of data bytes from destination to source
land	1 if connection is from/to the same host/port; 0 otherwise
wrong.fragment	number of "wrong" fragments
urgent	number of urgent packets
hot	number of "hot" indicators
num.failed.login	number of failed login attempts
logged.in	1 if successfully logged in; 0 otherwise
num.compromised	number of "compromised" conditions
root.shell	1 if root shell is obtained; 0 otherwise
su.attempted	1 if "su root" command attempted; 0 otherwise
num.root	number of "root" accesses
num.file creations	number of file creation operations
num.shells	number of shell prompts
num.access.files	number of operations on access control files
num.outbound.cmds	number of outbound commands in an ftp session
is.hot.login	1 if the login belongs to the "hot" list; 0 otherwise
is.guest.login	1 if the login is a "guest" login; 0 otherwise
count	number of connections to the same host as the current connection in the past two seconds
srv.count	number of connections to the same service as the current connection in the past two seconds
error.rate	% of connections that have "SYN" errors
srv.error.rate	% of connections that have "SYN" errors
error.rate	% of connections that have "REJ" errors
srv.error.rate	% of connections that have "REJ" errors
same.srv.rate	% of connections to the same service
diff.srv.rate	% of connections to different services
srv.diff.host.rate	% of connections to different hosts
dst.host.count	number of connections to the same host as the current connection in the past two seconds
dst.host.srv.count	number of connections to the same service as the current connection in the past two seconds
dst.host.same.srv.rate	% of connections to the same service
dst.host.diff.srv.rate	% of connections to the different service
dst.host.same.src.port.rate	% of connections to the same port
dst.host.srv.diff.host.rate	% of connections to different hosts
dst.host.error.rate	% of connections that have "SYN" errors
dst.host.srv.error.rate	% of connections that have "SYN" errors
dst.host.error.rate	% of connections that have "REJ" errors
dst.host.srv.error.rate	% of connections that have "REJ" errors

5.2 実験

本実験では、通常データを normal、それ以外を attack として、データが normal か attack か判定させる IDS を SVM で構成する。実験データとして KDDCup1999 Dataset 内のプロトコルタイプが TCP、サービスが HTTP のデータを抽出して使用する。実験に使用する学習用データとテスト用データの個数を表 5.3 に示す。

表 5.3 学習用データ数とテスト用データ数

学習用データ		テスト用データ	
normal	attack	normal	attack
61886	2506	39247	2050

今回 IDS を構成するにあたって、SVM ライブラリである LIBSVM[9] を利用して構成した。LIBSVM では、文字列のデータを扱うことができないため、KDDCup1999 Dataset 内の文字列部分であるプロトコル、サービス、フラグの符号化を行う必要がある。プロトコルは 3 種類、サービスは 66 種類、フラグは 11 種類存在するため、本実験ではそれぞれを次元とし、対応するものに 1、それ以外に 0 を当てはめることでデータの符号化を行っている。実験の基本的な流れを図 5.2 に示す。

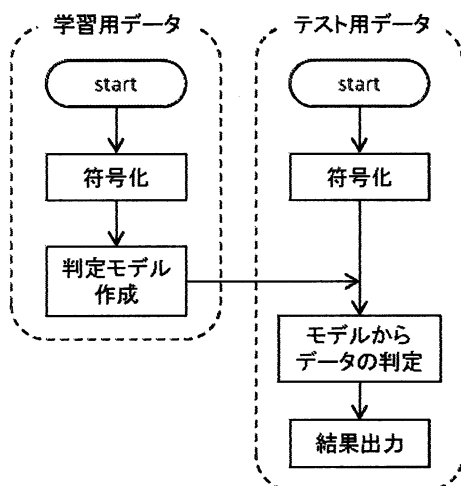


図 5.2 実験の基本的流れ

5.2.1 学習用データ数を変更した実験

本実験で使用する学習用データは、normal 数が 61886, attack 数が 2506 と圧倒的に normal のデータ数が多い。そこで、学習用データの attack 数はそのまま、normal 数のみを減少させてモデルを作成し、データを判定する実験を行った。学習用データの normal 数をランダムに 5000, 10000, 20000, 30000, 40000, 50000, 60000 個選択してモデルを作成し、テスト用データを判定させた結果を表 5.4 に、結果の推移を図 5.3 に示す。

表 5.4 学習用データの normal 数を変更したときの判定結果

学習用データ normal 数	normal 判定数 (判定率)	attack 判定数 (判定率)
5000	39216 (99.92%)	1905 (92.93%)
10000	39222 (99.94%)	1862 (90.83%)
20000	39224 (99.94%)	1829 (89.22%)
30000	39229 (99.95%)	1827 (89.12%)
40000	39229 (99.95%)	1826 (89.07%)
50000	39231 (99.96%)	1825 (89.02%)
60000	39235 (99.97%)	1822 (88.88%)

学習用データの normal 数を減らすにつれて attack の判定結果が向上していることが表 5.4 と図 5.3 からわかる。これは、学習用データの normal 数と attack 数のバランスをとることで、片方のクラスについて過学習が減ったこと、もしくはノイズとなるデータが減少したことに伴う SVM のソフトマージンによる修正が減少したことが原因であると考えられる。normal 数が 5000 の時でもフォールスネガティブ率が 7.07% とまだ高めの結果になっているため、もう少し精度の向上が必要である。

normal の判定結果は normal 数を減らすにつれて若干減少している。しかし、normal 数が 5000 のときでもフォールスポジティブ率が 0.08% と非常に低く、十分な精度を保っているといえる。

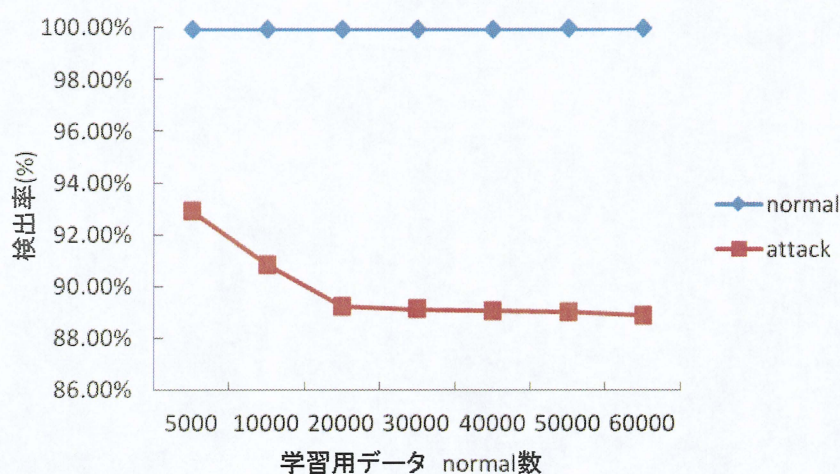


図 5.3 結果の推移

5.2.2 重みパラメータを変更した実験

学習用データの normal 数を 60000, attack を 2506 と固定し, normal クラスの重みパラメータを 1, 0.1, 0.01, 0.001 としてモデルを作成し, テスト用データを判定させた結果を表 5.5 に示す.

表 5.5 normal クラスの重みを変更したときの判定結果

normal 重み	normal 判定数 (判定率)		attack 判定数 (判定率)	
1	39235	(99.97%)	1822	(88.88%)
0.1	39218	(99.93%)	1867	(91.07%)
0.01	39211	(99.91%)	2010	(98.05%)
0.001	0	(0.00%)	2050	(100.00%)

表 5.5 から, normal クラスの重みを減らすほど attack の判定率が上がっていることがわかる. これは, normal クラスの重みを減らすことで, 学習データ内の normal クラスに存在する大きく特徴の異なるノイズデータの影響がなくなるためと考えられる. ただし, normal クラスの重みが 0.001 の場合のように, 重みを減らしすぎてしまうと normal クラス全体の特徴を無視してしまうため, 適切な値のを見つけ方が重要となる.

5.2.3 学習用データ数, 重みパラメータを変更した実験

学習用データの normal 数を変更した実験と, normal の重みパラメータを変更した実験のそれぞれで attack の検出精度が向上した. そこでこれらの手法を組み合わせることにより, より attack の検出精度を向上させることが可能か検証するために実験を行った.

学習用データの attack 数を 2506 と固定し, normal 数を 5000, 10000, 20000, 30000, 40000, 50000, 60000, normal クラスの重みを 1, 0.1, 0.01 と変更しモデルを作成する. 作成したモデルからテスト用データを判定させた結果を表 5.6 に, attack の判定結果の推移と normal の判定結果の推移を図 5.4, 5.5 にそれぞれ示す.

表 5.6 学習用データの normal 数と normal クラスの重みを変更した実験結果

normal 重み normal 数		1	0.1	0.01
5000	normal 判定数 (判定率)	39216 (99.92%)	39209 (99.90%)	39207 (99.90%)
	attack 判定数 (判定率)	1905 (92.93%)	2017 (98.39%)	2046 (99.80%)
10000	normal 判定数 (判定率)	39222 (99.94%)	39212 (99.91%)	39208 (99.90%)
	attack 判定数 (判定率)	1862 (90.83%)	2002 (97.66%)	2046 (99.80%)
20000	normal 判定数 (判定率)	39224 (99.94%)	39216 (99.92%)	39208 (99.90%)
	attack 判定数 (判定率)	1829 (89.22%)	1942 (94.73%)	2014 (98.24%)
30000	normal 判定数 (判定率)	39229 (99.95%)	39217 (99.92%)	39209 (99.90%)
	attack 判定数 (判定率)	1827 (89.12%)	1913 (93.32%)	2014 (98.24%)
40000	normal 判定数 (判定率)	39229 (99.95%)	39217 (99.92%)	39209 (99.90%)
	attack 判定数 (判定率)	1826 (89.07%)	1897 (92.54%)	2013 (98.20%)
50000	normal 判定数 (判定率)	39231 (99.96%)	39217 (99.92%)	39210 (99.91%)
	attack 判定数 (判定率)	1825 (89.02%)	1890 (92.20%)	2012 (98.15%)
60000	normal 判定数 (判定率)	39235 (99.97%)	39218 (99.93%)	39211 (99.91%)
	attack 判定数 (判定率)	1822 (88.88%)	1867 (91.07%)	2010 (98.05%)

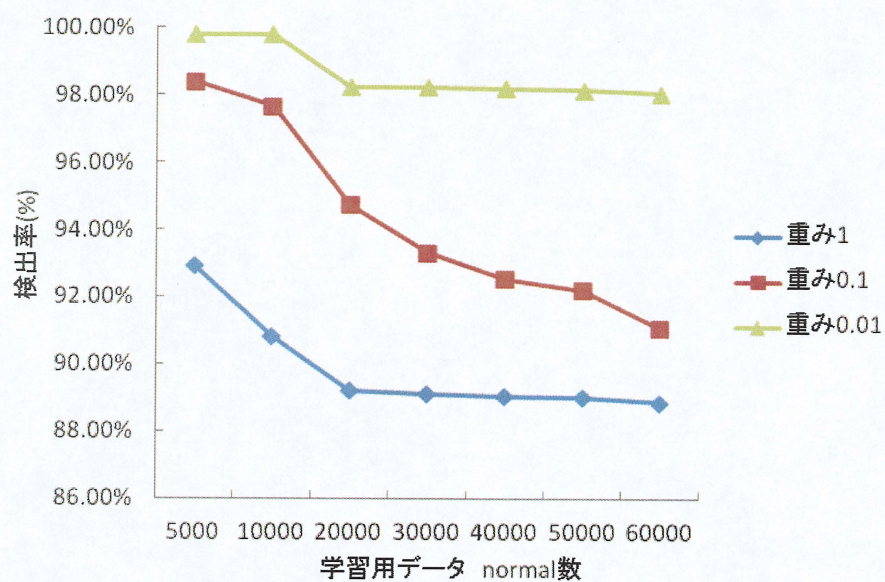


図 5.4 各条件下における attack 判定結果の推移

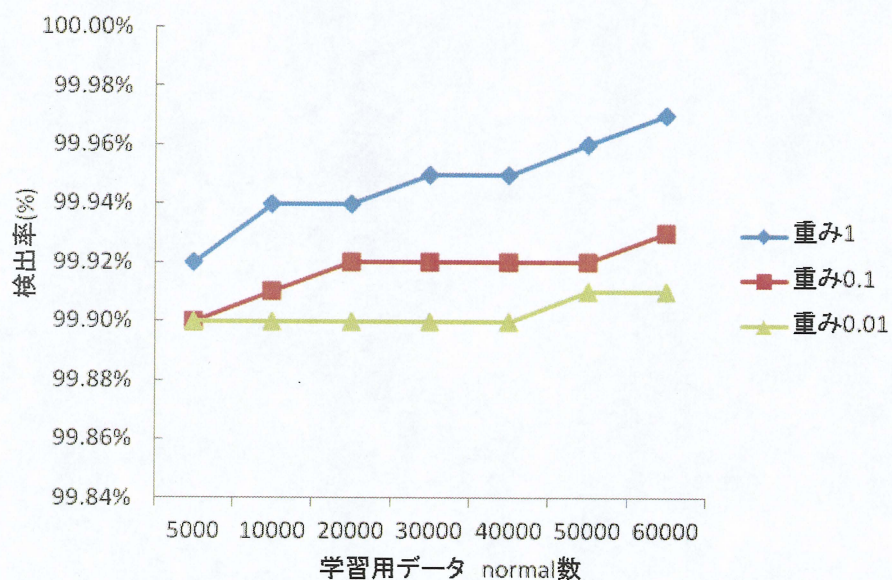


図 5.5 各条件下における normal 判定結果の推移

5.2.4 既存研究の結果との比較

本研究と同じく，プロトコルが TCP，サービスが HTTP のデータのみを用いて実験を行なっている既存研究との結果を比較する．提案手法における normal 数を 10000，normal クラスの重みを 0.01 としたときの結果，既存研究である遺伝的アルゴリズムを用いたときの結果 [10]，ニューラルネットワークを用いた時の結果 [11] をそれぞれ表 5.7 でまとめる．

表 5.7 各手法の評価結果

	normal 判定結果	attack 判定結果
本研究の提案手法	99.90%	99.80%
遺伝的アルゴリズム [10]	94.27%	99.73%
ニューラルネットワーク [11]	99.32%	99.74%

表 5.7 から，提案手法による判定精度が normal，attack とともに一番高い結果となった．提案手法で見逃した attack データ数，つまりフォールスネガティブは 4 つ存在し，そのすべてが apache2 のサブクラスに属するデータであった．この攻撃はテスト用データのみに含まれるデータで，テスト用データ内に計 794 個存在している．つまり，そのほとんどを検出することができており，既知攻撃はすべて検出できていることから，本手法は既知攻撃にとっても強く，未知攻撃にも対応できる IDS を構成可能であるといえる．

第6章

結論

本研究では、機械学習アルゴリズムの一つである SVM を利用して機械学習型の IDS を構成し、学習用データ数のバランスをとって学習させる手法と通常パケットの重みを変更して学習させる手法を提案した。さらに提案手法の有効性を調べるために、KDDCup1999 Dataset 内の TCP/HTTP データによる評価実験を行った。

その結果、学習用データ数のバランスをとる手法のみではフォールスネガティブ率を 4.5%、重みを変更する手法のみではフォールスネガティブ率を 9.17% 減少させることができた。また、2つの手法を組み合わせることで、フォールスポジティブ率を 0.1%、フォールスネガティブ率を 0.2% まで抑えることができた。この結果を、同じデータを用いた既存研究の結果と比較すると、どちらの誤検出も抑えることができていることから、既存研究よりも IDS の精度を向上することができたといえる。

今後の課題として、TCP/HTTP データのみではなく、全てのデータを使用した実験に適用できるよう改善する必要がある。そのために、適切なパラメータの調査や、データの変換方法を変化させる必要があると考えられる。

謝辞

日ごろから多くの御指導を頂きました太田義勝教授，鈴木秀智准教授に深く感謝いたします。そして，日頃何かとお世話になりました落合美子事務員に感謝いたします。また，本論文作成にあたって特にお世話になりました太田義勝教授に深く感謝いたします。最後に，日頃から熱心に討論して頂いた研究室の諸氏や，学生生活でお世話になりました全ての方々に深く感謝いたします。

参考文献

- [1] 総務省 平成 22 年度通信利用動向調査の結果
http://www.soumu.go.jp/menu_news/s-news/01tsushin02_01000014.html
- [2] 情報処理推進機構
<http://www.ipa.go.jp>
- [3] Brian Caswell, Jay Beale, James C. Foster, Jeffrey Posluns, Ryan Russell, “Snort 2.0 侵入検知”, 渡辺 勝弘, 鹿田 幸治 訳, ソフトバンクパブリッシング, 2004.
- [4] 宮地 充子, 菊池 浩明, “IT text 情報セキュリティ”, オーム社, 2003.
- [5] KDD Cup 1999 Data
<http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
- [6] Results of the KDD’99 Classifier Learning Contest
<http://cseweb.ucsd.edu/~elkan/clresults.html>
- [7] Bernhard Pfahringer, “Winning the KDD99 classification cup: bagged boosting”, ACM SIGKDD Explorations, Vol.1, No.2, pp.65-66, 2000.
- [8] Carolina Fortuna, et al., “ANOMALY DETECTION IN COMPUTER NETWORK USING LINEAR SVMs”, Conference on Data Mining and Data Warehouses, Oct 2007.
- [9] LIBSVM – A Library for Support Vector Machines
<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [10] 森 仁美, 中野 孝彦, 太田 義勝, 鈴木 秀智, “遺伝的アルゴリズムを利用した不正侵入検出について”, 2010 年度電気関係学会東海支部連合大会 (D2-5), 2010.8.30.
- [11] 胡 冰冰, “機械学習による不正アクセス検出について”, 卒業論文, 三重大学, 2008.

付録

A 提案手法による全データ実験

本提案手法を用いて, KDDCup1999 Dataset の全データを対象とした実験を行った. KDDCup1999 Dataset の学習用データ, テスト用データの内訳を表 A.1 に示す.

表 A.1 データ内訳

学習用データ					テスト用データ				
Normal	Probe	DoS	U2R	R2L	Normal	Probe	DoS	U2R	R2L
97278	4107	391458	52	1126	60593	4166	229853	228	16189

A.1 normal - attack 分類実験

まず, normal 以外のデータを attack とし, テスト用データを normal と attack の 2 つに分類する実験を行った. 学習用データの Normal を 60000, Attack を 60000 とし, Normal クラスの重みを 1, 0.1 に設定してテスト用データを判定した結果を表 A.2, 表 A.3 に示す. また, このときの学習用データの Attack の内訳は Probe が 4107, DoS が 54715, U2R が 52, R2L が 1126 である.

表 A.2 normal クラス重み 1

	normal 率 (検出数)		attack 率 (検出数)	
Normal	98.37%	(59606/60593)	1.63%	(987/60593)
Probe	14.81%	(617/4166)	85.19%	(3549/4166)
DoS	13.86%	(31853/229851)	86.14%	(197998/229851)
U2R	88.16%	(201/228)	11.84%	(27/228)
R2L	93.97%	(15214/16191)	6.03%	(977/16191)

normal 検出率 = 98.37%

attack 検出率 = 80.88%

表 A.3 normal クラス重み 0.1

	normal 率 (検出数)	attack 率 (検出数)
Normal	97.62% (59149/60593)	2.38% (1444/60593)
Probe	11.98% (499/4166)	88.02% (3667/4166)
DoS	2.81% (6452/229851)	97.19% (223399/229851)
U2R	80.26% (183/228)	19.74% (45/228)
R2L	90.76% (14695/16191)	9.24% (1496/16191)

normal 検出率 = 97.62%

attack 検出率 = 91.28%

[8] の結果と比較して、重みを 0.1 にした場合では attack の検出率は 0.37% 向上したが normal の検出率は 0.36% 低下した。また、表 A.2, 表 A.3, 既存研究 [7][8] のいずれの結果においても U2R, R2L の検出率が低いことが問題点として挙げられる。

そこで、U2R と R2L を攻撃としてより多く検出するために、normal と U2R・R2L を分類する学習モデル (n-ur モデル) を作成し、normal と判定されたデータに対してもう一度判定を行わせる方式で実験を行った。判定のフローチャートを図 A.1 に示す。normal と attack を分類する学習モデルを n-a モデルとし、n-a モデルと n-ur モデルの学習環境を表 A.4 に示し、その実験結果を表 A.5 に示す。

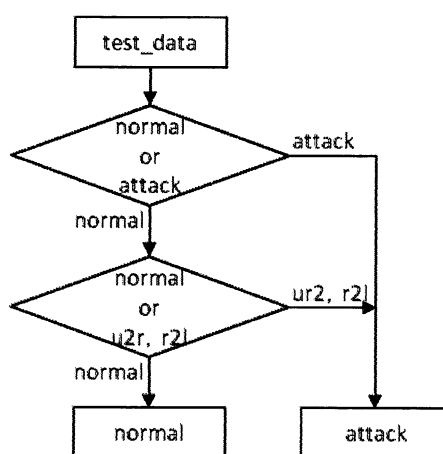


図 A.1 判定フローチャート

表 A.5 から、U2R についての検出精度が向上していることがわかる。しかし、R2L の検出精度はあまり向上していないため、attack 全体の検出精度は 0.59% の向上となった。同時に normal

表 A.4 各学習モデルの実験環境

	学習 normal 数	学習 attack 数	normal 重み
n-a モデル	60000	60000	0.1
n-ur モデル	10000	1178	0.01

表 A.5 n-ur モデルを導入した実験結果

	normal 率 (検出数)		attack 率 (検出数)	
normal	94.58%	(57309/60593)	5.42%	(3284/60593)
Probe	1.51%	(63/4166)	98.49%	(4103/4166)
DoS	2.69%	(6175/229853)	97.31%	(223678/229853)
U2R	0.00%	(0/228)	100.00%	(228/228)
R2L	87.17%	(14112/16119)	12.83%	(2077/16119)

normal 検出率 = 94.58%

attack 検出率 = 91.87%

の検出精度が低下しており, n-ur モデルを導入する前と比較して検出精度が 3.04% 低下した.

A.2 One-to-One を用いた分類実験

既存研究 [8] で使用されている One-to-One 方式を利用して実験を行った.

A.2.1 One-to-One

One-to-One は, まず n 個のクラスが存在する学習データの中から 2 つのクラスを抽出し, nC_2 通りの新しいデータを作成する. 次に, 作成した各データから 2 つのクラスを分類する学習モデルを SVM で構築する. KDDCup1999 Dataset におけるここまでの作業を図 A.2 に示す. 構築された全ての学習モデルによってデータを判定し, その結果の多数決をとることで多クラスの分類を行う.

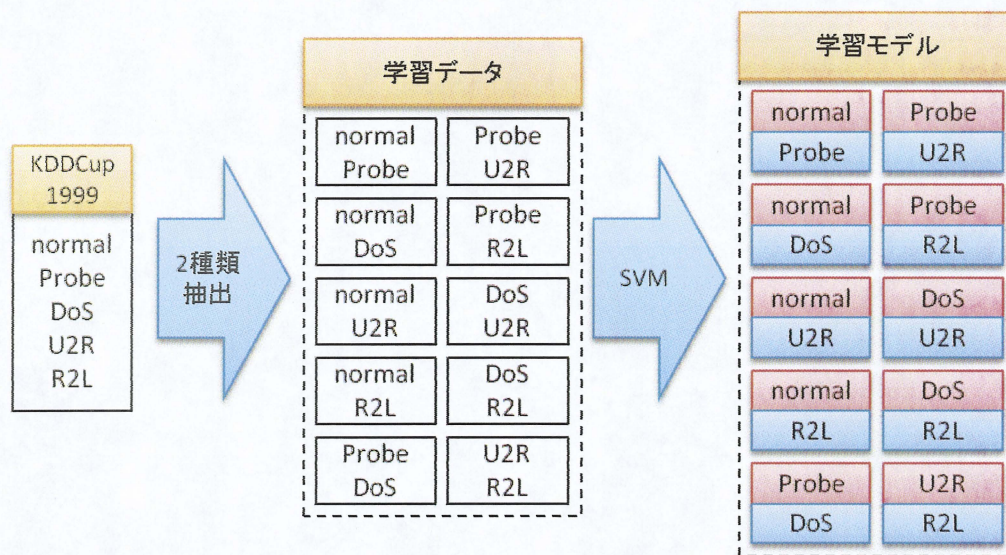


図 A.2 One-to-One 方式における学習モデル作成

A.2.2 実験結果

normal の重みを 1, 0.1 として実験を行った結果をそれぞれ表 A.6, A.7 に示す。

表 A.6 normal 重み 1

predicted actual						correct(%)
	Normal	Probe	DoS	U2R	R2L	
Normal	59584	911	42	6	50	98.33%
Probe	763	3072	189	0	142	73.74%
DoS	13683	19	215288	0	861	93.66%
U2R	193	0	0	24	11	10.53%
R2L	15235	1	0	7	948	5.86%
correct(%)	66.61%	76.74%	99.89%	64.86%	47.12%	

normal 検出率 = 98.33%

attack 検出率 = 88.07%

表 A.7 normal 重み 0.1

actual \ predicted						correct(%)
	Normal	Probe	DoS	U2R	R2L	
Normal	59133	1064	214	9	173	97.59%
Probe	498	3315	226	0	127	79.57%
DoS	6383	46	222958	0	464	97.00%
U2R	191	1	1	25	10	10.96%
R2L	14538	8	3	7	1635	10.10%
correct(%)	73.24%	74.76%	99.80%	60.98%	67.87%	

normal 検出率 = 97.59%

attack 検出率 = 91.37%