修士論文

センサデバイスを用いたネットワーク異常検知に関する研究

1710239　淺葉 祥吾

主指導教員　篠田 陽一 教授
審査委員主査　篠田 陽一 教授
審査委員　知念 賢一 特任准教授
丹 康雄 教授
Razvan Beuran 特任准教授

北陸先端科学技術大学院大学
先端科学技術研究科
(情報科学)

平成 31 年 2 月

概 要

インターネットは，人々が経済活動や社会生活を送る上で必要不可欠な社会基盤となっている．そのため，ネットワーク障害は，社会生活に対して深刻な影響を与える恐れがある．ネットワーク障害が長時間に及んだ場合，社会基盤に与える影響は大きい．よって，迅速および正確なネットワーク異常の検知は重要である．ネットワークの異常検知を行うことで，ネットワーク管理者が障害へ迅速に対応できる．

ネットワーク障害の原因は，大規模なネットワークになると多岐に渡り複雑化する．そのため，ネットワークを提供している組織では，ネットワークをモニタリングすることで，ネットワーク計測に基づく異常検知を行なっている．

ネットワークを構成する機器の不調や障害の予兆を捉えるためには，ネットワーク状態の変化を計測することが重要である．しかし，ネットワーク機器の計測サンプリング間隔を細かく設定したり，ネットワーク機器の計測項目を増加させてしまうと，機器の計算リソースを多く使い，負荷を増加させ，本来のネットワークシステムとして役割に支障をきたすことに繋がる．

そこで本研究では，センサデバイスからの計測結果からネットワーク異常検知を行なった．先行研究である SINDAN Project のネットワーク状態計測手法を用いて，センサデバイスから定期的にネットワーク状態計測を行い，計測結果にアルゴリズムを適応することで，発見が困難な異常を検知する手法を検証した．センサデバイスを用いることで，ネットワーク機器に大きな負荷をかけることなく計測できる．また，ネットワーク計測データは，そのデータの種類によって，異なる周期性や相関などの特性がある．それらの特性に着目し，発見が困難な異常の検知を実現した．

本研究で異常検知に使用するデータの収集の SINDAN Project のネットワーク状態計測手法では，ネットワーク状態をデータリンク層，インターフェース設定層，ローカルネットワーク層，グローバルネットワーク層，名前解決層，ウェブアプリケーション層のそれぞれの階層別に計測を行なっている．

本手法の評価のため，発見が困難な障害事例として，無線 LAN 環境の通信帯域が悪化することでスループットが低くなる障害を想定し，異常検知の実験を行なった．実験環境において 1 分間隔のネットワーク状態計測を行い，階層ごとのデータを取得する．計測結果に複数のアルゴリズムを適用することで異常検知を試み，その結果から考察を行なった．異常検知には，教師なし学習のアルゴリズムを採用し，相関するデータの集まりから外れ値を検出する Local Outlier Factor と変化点検出である Change Finder を使用した．教師なし学習は，正常データと異常データの学習を必要としないため，あらかじめネットワーク計測で得られるそれぞれのデータについて正常値と異常値を定義しなくても，異常検知を行える．

計測のメトリックとしては，ネットワーク状態計測手法のローカルネットワーク層からIPv4デフォルトルータまでのping 10回分の平均値の計測結果（以下では，v4rtt_router_aveと呼ぶ）とIPv4デフォルトルータまでのping 10回分の標準偏差の計測結果（以下では，v4rtt_router_devと呼ぶ）を選定した．

Local Outlier Factorによる異常検知では，v4rtt_router_aveとv4rtt_router_devに，相関関係が見られたが，ネットワーク異常を検知できる結果は得られなかった．Change Finderによる異常検知では，v4rtt_router_aveとv4rtt_router_devの双方において異常の検知に成功した．ただし，Change Finderでは揺らぎが大きいデータに適さないため，本研究では，前処理を行うことで異常検知の精度を向上させた．v4rtt_router_aveとv4rtt_router_devは，単調な増加や減少ではなく単発的な高い値をとりうるが，メディアンフィルタを用いて前処理を行うことで，Change Finderの精度の向上を実現した．これらの提案手法によって得られた結果について，F値を用いて検証を行い，提案手法の有効性を示した．

本研究の成果として，センサデバイスを用いたネットワーク異常検知の提案により，ユーザサイドにセンサデバイスを置くことで，発見が困難な無線LAN環境内の負荷変化による異常をv4rtt_router_aveとv4rtt_router_devの前処理とChange Finderから検知を実現した．

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　背景

総務省の情報通信機器の普及状況 [1] によると，2016 年の世帯における情報通信機器の世帯普及率は，モバイル端末全体が 94.7%，パソコンが 73.0%と高い値になっている．ICT (Information and Communication Technology) は，情報通信技術の略であり，コンピュータ関連の技術である．総務省は，ICT 利活用の推進 [2] を行なっている．また，将来のネットワークインフラに関する研究会の報告書 [3] によると，IoT やネットワークを介した高繊細による映像配信等により，今後もトラフィック量は増加することがわかり，ネットワークの安定運用は社会的に不可欠なものであるといえる．

サイバーセキュリティは，情報の機密性や完全性，可用性を維持することであり，日々深刻な問題になっている．サイバー攻撃は，世界中で攻撃が増加し，セキュリティの脅威を迅速に観測・分析し，有効な対策を導出することが重要である [4]．ネットワークの状態を，迅速に計測して分析することが重要である．

ネットワークを提供している大学やインターネットサービスプロバイダー (Internet Service Provider，以下は，ISP と呼ぶ) は，ネットワークのシステムを構成する機器に障害が起きても，ネットワークの全体の機能を維持できるように，単一障害点 (SPOF : Single Point Of Failure) を防ぐ冗長化をしている．そのため，モニタリング対象が多くなり，障害点の発見が困難になる．

無線ネットワーク環境は，様々な施設で提供されており，これからも提供範囲が拡大していくが，免許不要の周波数帯域を利用するため，周辺環境の依存や様々な電子機器との電波干渉などを起こすので，障害になりやすく，障害発見が困難になる．

また，AWS (Amazon Web Service) [5] などクラウドコンピューティング (以下は，クラウドと呼ぶ) のサービス利用が拡大している．クラウドは，クラウドサービスプラットフォームから，インターネット経由で，コンピューティング，データベース，ストレージ，アプリケーションをはじめとした，様々な IT リソースを随時に利用できるサービスである．クラウドは，それを使用する組織にハードウェアやネットワーク機器を導入するオンプレミスのサービスと異なり，必要な時に必要な量のリソースを簡単に利用できる．しかし，サーバやネットワーク機器の状態は，クラウドの利用者からは見えないので，障害が発生した時に，サーバやネットワーク機器の問題かクラウドのリソースの問題か判断が困難になる．

現在，主流であるインターネットプロトコルにはIPv4とIPv6がある．IPv6は，IPv4のアドレスが枯渇する問題を解消するためのインターネットプロトコルである．様々なネットワーク機器やクライアントとサーバのOSではIPv6に対応しており，IPv6に関するJPNICの記事 [6] からGoogleが公開しているIPv6採用に関する統計によると，全世界からGoogleに対してIPv6でアクセスするユーザの割合は，2018年5月6日では約22%であり増加していることがわかる．IPv6の普及により，IPv4とIPv6のデュアルスタックのネットワーク環境やIPv6のみの環境へ移行した環境になると，新たにIPv6の固有の問題による障害が起こり，障害の発見が困難になる．

また，ネットワーク管理者は，ユーザから「ネットワークにつながらない」や「ネットワークが遅い」などの障害報告を受ける．このユーザからの障害報告は，ユーザ端末のDHCPでのアドレス問題や無線ネットワーク環境の障害，DNSにおける名前解決の問題など様々な原因があるが，ユーザが障害情報を的確に捉えることは困難であり，ネットワーク管理者にネットワーク状態を的確に報告することが難しい．

ネットワーク管理者は，大規模のネットワークでも，品質を維持し，ネットワークのシステムを提供しなければならない．ネットワーク管理者は，ネットワーク機器の障害防止や迅速な障害復旧が求められており，ネットワーク障害を発見するため，それらのネットワーク異常の検知が重要である．

1.2 目的

ネットワーク障害は，ネットワークを用いて提供されているシステム全体の停止に繋がるので，迅速に対応することが求められる．ネットワークに障害がおきた場合，経済活動や社会活動が止まる恐れがある．そのため，ネットワーク管理者が，ネットワーク障害時に迅速に対応できるように，障害点の情報を瞬時に把握でき，復旧作業の手続きが的確なければならない．

本研究はセンサデバイスを用いたネットワークの異常検知を検証する．センサデバイスを使用することで，ネットワーク機器に高負荷を掛けることを減らせ，ネットワーク環境の測定をおこない，また，各サブネットワークにセンサデバイスを置くことで，細かなネットワークセグメントにおいてもモニタリングすることができる．

センサデバイスを用いたネットワーク異常検知の手法を導入することで，ネットワーク管理者は，ネットワーク障害を迅速に発見することができる．本研究では，センサデバイスからのネットワーク計測のメトリックから，異常検知に繋がるメトリックを選定し，発見が困難な障害を，検知するアルゴリズムを検討し，異常検知について評価を行う．

1.3　本論文の構成

第 2 章は，ネットワーク計測手法と課題を示す．ネットワークに関する技術と計測手法とモニタリングを述べる．また，ネットワーク計測で現れる特徴とネットワーク計測における課題を述べる．

第 3 章は，センサデバイスを用いたネットワークの障害に関する計測手法の研究，異常検知の研究，障害復旧の手法の研究に加え，関連研究と本研究の差分について述べる．

第 4 章は，先行研究である SINDAN Project のネットワーク状態計測手法がどのような計測しているのか述べる．

第 5 章は，本研究のセンサデバイスを用いたネットワーク異常検知の手法の設計を述べる．また，異常検知に使用したアルゴリズムとアルゴリズムの特性について述べる．

第 6 章は，実験ネットワーク環境の検証と計測結果について述べる．また，ネットワーク異常検知アルゴリズムの結果と評価を述べる．

第 7 章は，本研究の考察を述べ，第 8 章は，本研究のまとめと今後の展望を述べる．

第2章　ネットワーク計測手法と課題

本章では，ネットワークの障害に関する問題を整理し，一般的なネットワークに関する計測手法についてまとめる．また，ネットワークの計測とモニタリングについて述べる．異常検知のためのネットワーク計測手法は様々な提案がされており，ネットワーク計測における特徴がある．そして，ネットワーク計測手法が抱えている課題について述べる．

2.1　ネットワーク障害

ネットワーク障害には，様々な原因がある．プロトコルは，通信の約束事である．インターネットの通信では，ノートパソコンやスマートフォンなどの多種多様なデバイス間でも通信を行えるように TCP/IP プロトコルスイートに準拠している．TCP/IP プロトコルスイートは，IETF (Internet Engineering Task Force) [7] で議論を通して標準化している．以下に，一般的な TCP/IP プロトコルスイートによる，クライアントからサーバまでのデータの流れについて述べる．図 2.1 に，TCP/IP プロトコルスイートにより複数のルータを通したクライアントからサーバへデータ送信の流れを示す．クライアントはからサーバへデータ送信を行う時，クライアントは，アプリケーション層，トランスポート層，インターネット層，ネットワークインターフェイス層の順でルータに送信する．ルータは，ネットワークインターフェイス層，インターネット層，ネットワークインターフェイス層の順で次のルータやサーバに送信する．サーバは，ネットワークインターフェイス層，インターネット層，トランスポート層，アプリケーション層の順で受信する．TCP/IP プロトコルスイートの階層モデルは，ネットワークインターフェイス層，インターネット層，トランスポート層，アプリケーション層である．ネットワークインターフェイス層は，同一のセグメントに対して通信するためのインターフェイスとなる階層である．有線 LAN や無線 LAN などのデータリンクを利用する．有線 LAN の代表的な規格は，イーサネット (Ethernet) であり，イーサネットは，IEEE802.3 [8] 諸規格で整備されている．また，無線 LAN は，IEEE802.11 [9] 諸規格で整備されている．インターネット層は，異なるセグメントに存在する機器間のデータ伝送を行う階層である．IP アドレスに基づきパケットを送信する．インターネット層で使用してるプロトコルは，IPv4 [10] や IPv6 [11]，ICMPv4 [12]，ICMPv6 [13] などある．トランスポート層は，データの通信制御する階層である．

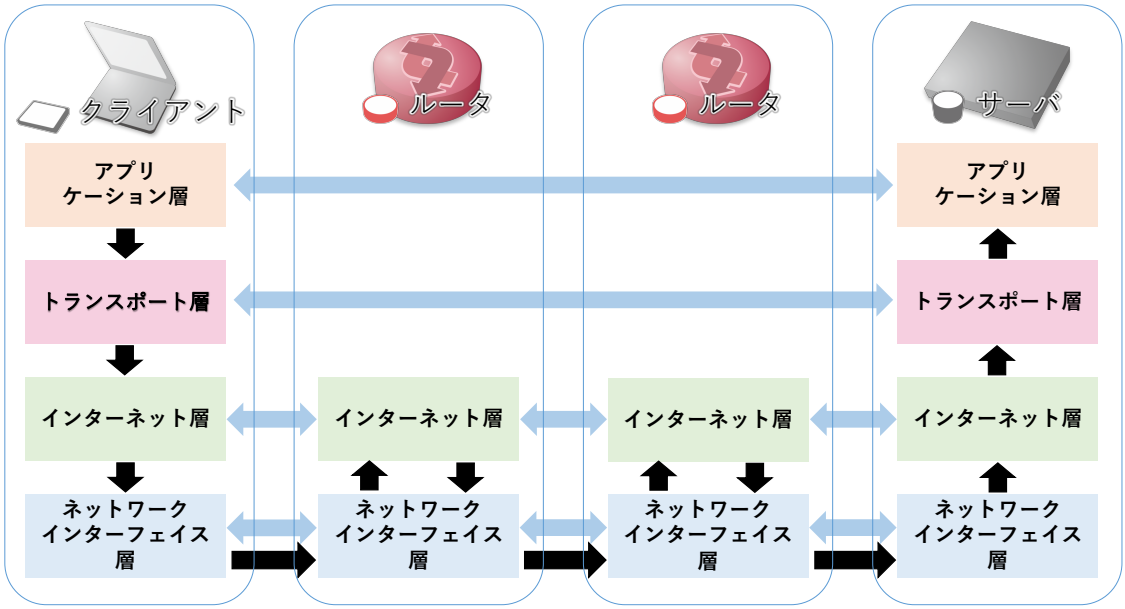

図 2.1: 複数のルータを経由した送信の流れ

トランスポート層で使用してるプロトコルは，Transmission Control Protocol (以下では，TCP と呼ぶ) [14] や User Datagram Protocol (以下では，UDP と呼ぶ) [15] などである．TCP は，信頼性の高い通信を実現するために，確認応答，順序制御，ウィンドウ制御，フロー制御などの機能がある．UDP は，信頼性を確保しないが，リアルタイム性を求められる通信に使用される．アプリケーション層は，アプリケーション間でのデータ伝送やデータの形式，データ送受の手順を定める層である．アプリケーション層で使用しているプロトコルは，HTTP [16] や HTTPS [17]，DNS [18]，DHCP [19]，DHCPv6 [20] などである．Domain Name System (以下では，DNS と呼ぶ) は，インターネット上でドメイン名を管理・運用するためのシステムである．Dynamic Host Configuration Protocol (以下では，DHCP と呼ぶ)/DHCPv6 は，IP アドレスの自動設定を行うのシステムである．

2.2　発見が困難なネットワーク障害例

このようなネットワークにおいて通信している場合，起こりえる障害は多岐にわたる．アプリケーション層で障害があるときは，実際にアプリケーション層を使用する通信を使わないと発見できない時もある．以下に，ネットワークインターフェイス層とインターネット層・トラスポート層，アプリケーション層の発見が困難な障害の例をあげる．

2.2.1　ネットワークインターフェイス層の障害事例

無線 LAN 環境の障害事例は以下がある．2.4GHz 帯においては，Bluetootch や電子レンジが発生する周波数が干渉する障害がある．5GHz 帯においては，気象レーダやドローンが通信を行う時に DFS (Dynamic Frequency Selection) によって動的に使用する電波帯域の変更をおこなうが，無線 LAN の AP における MP レーダの受信レベルが低くなると，DFS が動作しなく，気象レーダの周波数と干渉する障害がある．また，無線 LAN 環境を提供する AP(Accsess Point) と，ユーザが各自に使用しているスマートフォンのデザリングの周波数の衝突する障害がある．有線 LAN では，一部のサーバやネットワーク機器の不調による特定のネットワーク区間のみ通信が困難になる障害がある．

2.2.2　インターネット層・トランスポート層の障害事例

インターネット層では，IPv4 と IPv6 のデュアルスタックのネットワーク環境は，Happy Eyeballs [21] [22] による DNS の問い合わせから，実際に IPv4 と IPv6 を通信し，通信状態の良い方を優先して使用する仕組みがあるが，複雑化し正しく行えているか問題がある．また，トランスポート層では，コネクションに関する障害などある．

2.2.3　アプリケーション層の障害事例

DHCP/DHCPv6 サーバによるアドレス自動設定ができないやサーバやネットワーク機器の設定ミス，HTTP/HTTPS で通信できない，DNS リゾルバの設定ミスによる名前解決ができない，セキュリティアプライアンスの設定ミスなどによる障害がある．

2.3　ネットワーク計測

ネットワーク障害を阻止や発見するために，ネットワーク計測やモニタリングが重要である．ネットワーク計測とは，サーバやネットワーク機器から，ネットワーク状態を様々な角度から測ることである．

ネットワークの障害を発見するネットワーク測定の手法として，アクティブ測定とパッシブ測定がある．表 2.1 に，アクティブ測定とパッシブ測定の概要を示す．アクティブ測定は，計測のために実際の通信を行い，ネットワーク状態を測定する．アクティブ測定により，ネットワークのスループットや往復遅延時間やパケットロス率，ジッタを計測できる．パッシブ測定は，計測用のパケットを送らずに，ネットワークに流れるパケットを測定する．実際のトラフィック量やパケットの種類，サービスの状態が計測できる．

表 2.1: アクティブ測定とパッシブ測定の測定項目

測定手法	計測手法の概要
アクティブ測定	スループットや往復遅延時間 パケットロス率 パケット到達時間のゆらぎ
パッシブ測定	トラフィック量 パケットの種類 サービスの状態

表 2.2 に，ネットワーク管理に用いる計測技術と概要を示す．ネットワーク管理に用いる計測は，IP アドレスを持つネットワーク機器に対して到達性を調べる Ping [12] や IP アドレスを持つネットワーク機器に到達するまでにどの経路を使用したのかを測定する Traceroute [23]，ネットワークの End-to-end のスループットを測定する iperf [24,25]，ネットワーク管理プロトコルの SNMP (Simple Network Management Protocol) [26]，リアルタイムにネットワークに流れているトラフィックをモニタできる xFlow，リアルタイムにネットワーク機器の状態をモニタできる Telemetry [27]，ネットワークを通してネットワーク機器などのシステムログを伝送するプロトコル Syslog [28] がある．

2.3.1 ping

ping は，IP アドレスを持つに対してサーバやネットワーク機器に対して到達性を調べるアクティブ測定のコマンドである．ping は，もともと管理用のツールであり，計測用のツールではない [29]．したがって，計測用に十分な精度を持っていないが，測定結果に統計処理を行うことで，計測ツールとして使用できる．ping は，IP アドレスで相手のノードを指定してメッセージを送信し，相手のノードからエコーされる応答メッセージを受信することで，到達性を確認している．メッセージには，ICMP エコーリクエスト/リプライを使用している．ICMP エコーリクエストのメッセージを受信した相手のノードが，送信元のノードに ICMP エコーリプライのメッセージを返す決まりである．送信元のノードが，受信した相手ノードから，リプライが帰ってくるまでの時間を使って，往復遅延時間 (Round-Trip Time または，Round-Trip delay Time 以下では，RTT と呼ぶ) を計算できる．また，パケットロス率も測定ができる．

ファイアウォールなどネットワークアプライアンスにより，ICMP をフィルタしている時もある．また，ping によって相手のノードとの到達性が確認できるが，ネットワークに異常があるのかわからない．ICMP が到達性があったとしても，TCP による通信が到達性がない時や，ping のパケットサイズは小さいので到着性があ

表 2.2: ネットワーク計測技術の概要

計測技術	計測技術の概要
ping	アクティブ測定 IP アドレスを持つサーバや ネットワーク機器に対して到達性
traceroute	アクティブ測定 IP アドレスを持つネットワーク機器に到達するまでに どの経路を使用したのかを測定
iperf	アクティブ測定 ネットワークの End-to-end のスループットを測定
SNMP	パッシブ測定 ネットワーク管理プロトコル
xFlow	パッシブ測定 リアルタイムにネットワークに 流れているトラフィックをモニタ
Telemetry	パッシブ測定 リアルタイムにネットワーク機器を 細くネットワーク機器の状態をモニタ
Syslog	パッシブ測定 ネットワークを通してサーバや ネットワーク機器などのシステムログを伝送

るが最大パケットサイズ (Maximun Transmisson Unit 以下では，MTU と呼ぶ) ではフラグメント処理によって到着性がなくなる時もある．

2.3.2 traceroute

traceroute は，ネットワーク機器に対してどのネットワークの経路の通過したのか調べるアクティブ測定のコマンドである．OS が，Windows の場合は tracert となる．IP アドレスを持つネットワーク機器に到達するまでに，通過したゲートウェイのルートとゲートウェイ間の応答遅延時間を計測する．ゲートウェイ間のネットワークに異常がないか確認できる．

traceroute は，一般的に UDP を使用してどの経路を通過するか確認するが，オプションで，TCP や ICMP を使用できる．traceroute の応用として，リアルタイムに traceroute を行う mtr [30] というコマンドがある．

2.3.3 iperf

iperf は，ネットワークのスループット計測を行うコマンドである．iperf は，ネットワーク機器から他のネットワーク機器へと，実際にテストデータを流して計測を行うアクティブ測定を行う．オプションにより TCP と UDP，インターバル，送信帯域などを指定できる．iperf で実際に計測を行うと，計測しているネットワークにも負荷をかけ，同じネットワークを使用している他の通信にも影響を与える．

2.3.4 SNMP

SNMP は，業界標準のネットワーク管理プロトコルで，パッシブ測定である．複数のベンダーによって構築されたネットワークでも，SNMP をサポートしているネットワーク機器を利用してれば，ネットワークを経由することで管理できる．SNMP は，様々なことをモニタリングや管理できるように設計されている．SNMP の通信は，情報を取得したいデバイスをエージェントと情報を受け取るデバイスをマネージャがある．SNMP は，マネージャからエージェントにリクエストを送信するインバンドのポーリングとエージェントからマネージャにエラーに関する情報を送信するアウトバンドのトラップがある．エージェントは，オブジェクト ID(OID) で構成されているツリー状で表記されて情報を管理している．SNMP は，ネットワーク機器の CPU を使用するため，頻繁に問い合わせするとネットワーク機器に負荷をかける．

2.3.5 xFlow

xFlow は，NetFlow [31] や sFlow [32]，J-Flow [33]，IPFIX [34–37] などのリアルタイムにネットワークに流れるトラフィックのフローをもとにモニタするパッシブ測定である．入力インターフェイスや，送信元 IP アドレス，宛先 IP アドレス，L3 プロトコル，TCP/UDP の送信元ポート，TCP/UDP の宛先ポート，IP ToS (Type of Service) をモニタする．フローモニタリングは，帯域幅を多く利用している通信やノードを突き止めたり，IP やプロトコル，アプリケーション，サービスごとの単位で分析できる．

NetFlow は，Cisco のフローモニタリング技術であり，実際にネットワークで流れているトラフィックフローを受動的にモニタできる機能である．NetFlow が有効なインターフェイスに流れているパケットをフロー集計する．sFlow は，受信または送信のパケットに対してサンプリングを行う．J-Flow は，Juniper のフローモニタリング技術であり，機能は，sFlow と同じである．IPFIX は，netflow v9 がベースとなっており，ペイロードの情報も取得できる．

2.3.6 Telemetry

Telemetry は，遠隔測定法を指しており，ネットワーク機器の様々な状態を効率よく別のノードへ定期的に送りモニタリングできる技術である [38]．Telemetry は，パッシブ測定である．プッシュ型にすることで，リアルタイム性を高め，より多くの情報を送れて，取得した情報から分析を行えるようにする．

2.3.7 Syslog

サーバやネットワーク機器，アプリケーションの動作記録をシステムログと呼ぶ．システムログは，フォーマットは自由記述であり，人が読めるように記録されて管理することが多い．システムログを，管理するサーバに転送することで，ネットワークにどのような状態にあるのか把握しやすくなる．Syslog は，ネットワークを通してシステムログを転送する標準規格のプロトコルである．Syslog では，ログの種別するファシリティと優先度を示すプライオリティがあり，危険な状態などを指定してログを送ることができる．

2.4 ネットワークモニタリングツール

ネットワーク計測は，ネットワーク環境から様々なメトリックを測定することであり，ネットワークモニタリングは，ネットワーク計測で観測したメトリックの結果を監視して，対象のネットワークが正常な動作をしているか観察し続けることである．ネットワークモニタリングツールは，高信頼なネットワークを提供する場合に必要不可欠である．

ネットワーク管理者は，アクティブ測定とパッシブ測定を組み合わせた様々なネットワークモニタリングツールを利用してネットワークを監視している．モニタリングシステムからネットワーク状態に異常を発見した場合に，管理者に対してその障害を通知することにより障害を迅速に対応できる．表 2.3 に，ネットワーク管理に用いるモニタリング技術と概要を示す．ネットワークトラフィックやリソースなどをグラフ化できる Cacti [39] やホストシステムやサービス，リソースなどの状態モニタリングできる Nagios [40]，ネットワークの状態やサーバの稼働状態と整合性をモニタリングできる Zabbix [41]，オープンソースプロジェクトのシステムモニタリングおよび警告ツールキットの Prometheus [42]，SNMP によるネットワーク機器とグローバルネットワーク，Web アプリもモニタリングする ThousandEyes [43]，IP アドレス管理をする IPAM，IPAM およびデータセンターインフラストラクチャ管理（DCIM）ツールである Netbox [44]，リアルタイムでホストのパフォーマンスを可視化しモニタリングできるツールである Netdata [45] などをある．また，モニタリングツールを運用者が作成したり，企業が提供して

表 2.3: モニタリングツールと概要

モニタリングツール	モニタリングツールの概要
Cacti	ネットワークトラフィックや リソースなどをグラフ化
Nagios	ホストシステムやサービス, リソースなどの状態モニタリング
Zabbix	ネットワークの状態やサーバの稼働状態と 整合性をモニタリング
Prometheus	オープンソースプロジェクトの システムモニタリングおよび警告ツールキット
ThousandEyes	SNMP によるネットワーク機器とグローバルネットワーク, Web アプリもモニタリング
IPAM	IP アドレス管理をする IPAM
Netbox	IPAM および データセンターインフラストラクチャ管理 (DCIM) ツール
Netdata	リアルタイムでホストのパフォーマンスを 可視化しモニタリングできるツール

る有償のツールを使用することがある．リソースは，CPU 利用率やメモリ，ディスクなどである．

2.5 ネットワーク計測データの特性と異常検知

ネットワーク計測データには，様々な特徴があることが知られている．計測データの特徴は，ネットワークを提供している組織やネットワークを構成している機器，ネットワークの利用者の量などが関連している．ネットワーク計測データでは，周期性や相関がある場合や，非常に不安定な値が計測される場合がある．図 2.4 に，ネットワーク計測データの特性と異常検知アルゴリズムの関係について示す．また，ネットワーク計測データの特性と異常検知アルゴリズムの概要について述べる．

2.5.1 周期性

ネットワークのトラフィック量は，時系列データで表すと一般的に周期性がある．大学や企業では，日中の人が多くなるつれてネットワークの利用者も増加することでネットワークのトラフィック量も増加し，夜中に人が少なくなるにつれてネットワークの利用者も減少することでネットワークのトラフィック量も減少す

表 2.4: ネットワーク計測データの特性と適した異常検知アルゴリズム

ネットワーク計測データの特性	異常検知アルゴリズムの関係
周期性	特異スペクトル変換法，自己回帰モデル， 移動平均，自己回帰移動平均モデル， 自己回帰和分移動平均モデル， 季節自己回帰和分移動平均モデル， 外生変数付き自己回帰和分移動平均モデル， 状態空間モデル
相関	カーネル密度推定， ガウス混合分布モデル， Local Outlier Factor， Isolation Forest， One-Class SVM

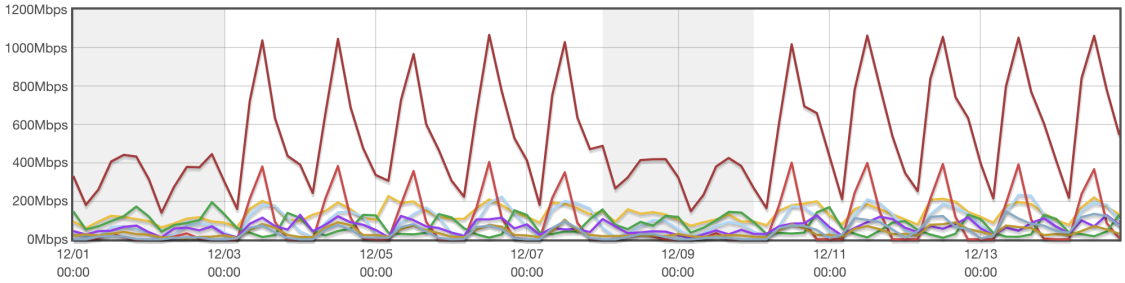

図 2.2: 周期性のあるトラフィックデータの例

る．また，平日は講義や業務があるのでネットワークの利用者が多いが，休日では休みの日なのでネットワークの利用者が少ない．したがって，ネットワークのトラフィックでは日単位や週単位で周期性が表れる．しかし，大学でイベントなどが開催されると，普段とは異なり周期性が現れないこともある．

図 2.2 に，ネットワーク計測から周期性がある例を示す．この図は，WIDE Project の MAWI Working Group [46] から，WIDE backbone で流れている一部のトラフィック量を可視化した [47]．横軸に，2018 年 12 月 1 日から 2018 年 12 月 14 日までの時刻，縦軸に，ネットワークのトラフィック量を示す．この図の各色の折れ線から，ネットワークのトラフィックでは日単位や週単位で周期性があることがわかる．

2.5.2 相関

ネットワーク計測では，相関が見られることがある．様々なネットワーク計測から相関関係を定義し，相関から外れる値は，異常値とみなすことがある．

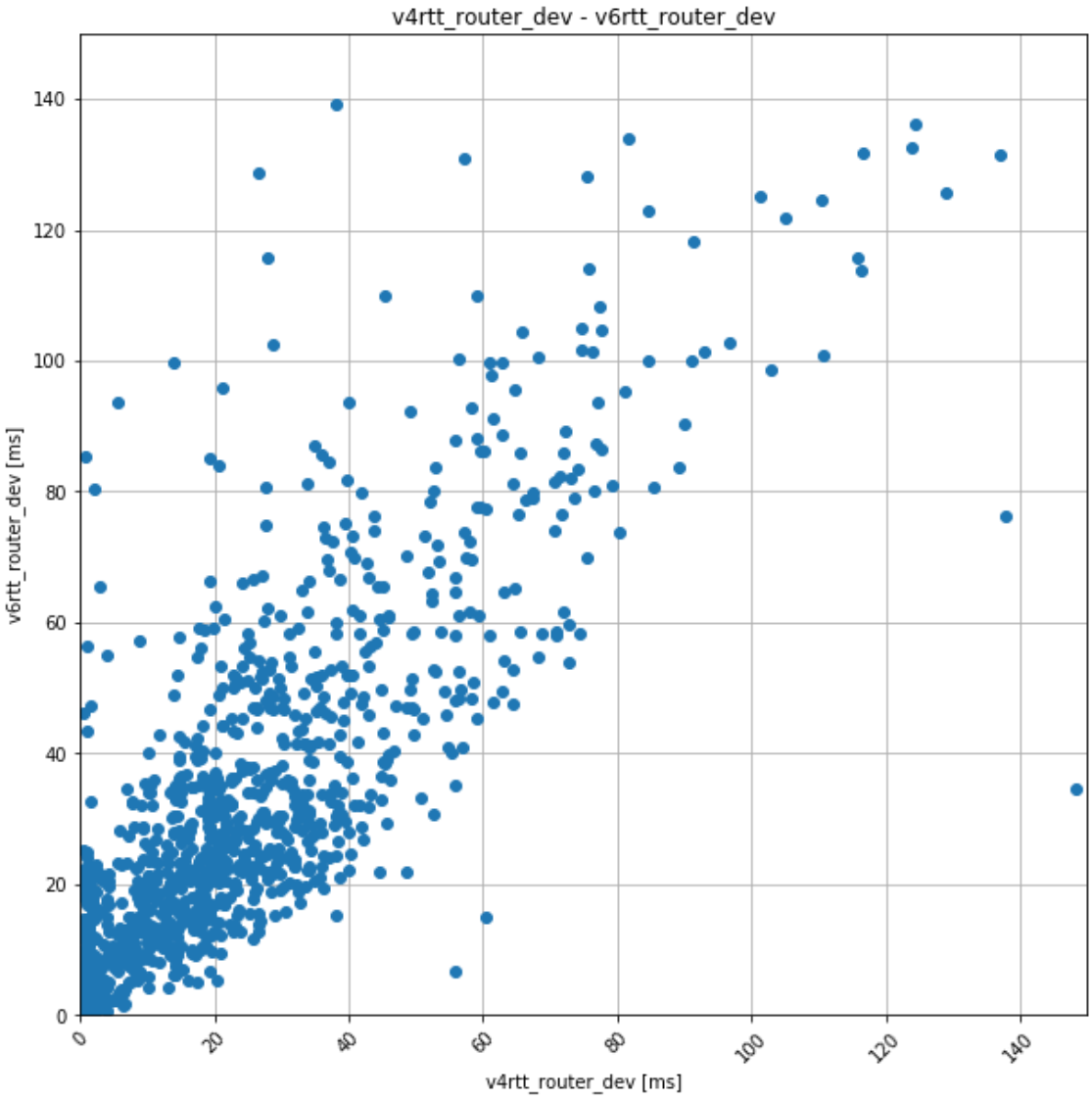

図 2.3: 計測データ同士の相関がある例

図 2.3 に，ネットワーク計測から相関がある例を示す．横軸に，ローカルネットワークのデフォルトルータに向けて，IPv4 による 10 回の ping による標準偏差の値，縦軸に，ローカルネットワークのデフォルトルータに向けて，IPv6 による 10 回の ping による標準偏差の値を示す．1 日間の 1 分おきに各 IPv4，IPv6 による ping の標準偏差の値である．この図から，各 IPv4，IPv6 による ping の標準偏差は比例関係にあることがわかる．

2.5.3 外れ値と異常検知

異常検知と変化点検知の典型的な外れ値の例を示す [48]．図 2.4 に，時系列データに対して外れ値の例を示す．各図は，横軸に時刻，縦軸に計測値である．正常値は青点，異常値は赤点で表している．(a) は，計測データ全ての仲間から値が外れている外れ値，(b) は，計測データの周期性から値が外れている外れ値，(c) は，計測データの周期性から周期がずれている変化点，(d) は，計測データの周期性から変化点または異常部位である．時系列データは，周期性などを含んだデータなどがある．閾値で異常値検出を行うと，周期の最大値と最小値の間に隠れることもあり，周期特性から外れ値を定義する必要がある．

周期性のあるデータに対して異常検知に利用するアルゴリズムは，特徴ベクトルから変化点検出の特異スペクトル変換法 (SIngular Spectrum Transformation) [49,50] や，自己回帰 (AR : Autoregressive) モデル，移動平均 (MA : Moving Average)，自己回帰移動平均 (ARMA : Autoregressive and Moving Average) モデル，自己回帰和分移動平均 (ARIMA : Autoregressive, Integrated and Moving Average) モデル，季節自己回帰和分移動平均 (SARIMA : Seasonal ARIMA) モデル，外生変数付き自己回帰和分移動平均モデル，状態空間モデルなどある．

図 2.5 に，相関のあるデータに対して外れ値の例を示す．相関のあるデータに対して外れ値の図は，横軸と縦軸に 2，-2 とに比重を置いたデータの値にである．$(x, y) = (-2, -2), (2, 2)$ を中心に島ができ，この 2 つ島から離れた値のデータは外れ値とみなすことがある．

相関のあるデータに対して異常検知に利用するアルゴリズムは，各データに対してカーネルを重ねてデータの分布を表現することで，データの存在確率が低い箇所を異常値とみなすカーネル密度推定 (KDE : Kernel Density Estimation) やデータの分布を数個の Gaussian の線形和の近似で表現しパラメータの最尤推定値を EM アルゴリズムで求めるガウス混合分布モデル (GMM : Gaussian Mixture Model)，Nearest Neighbor 法を改良した手法であり，周辺のデータとの局所的な関係から異常検知する Local Outlier Factor，決定木 (DT : Decision Tree) を改良した手法であり，木を構成する際に，分割する特徴量はランダム決め，深さをスコアとして異常検知する Isolation Forest，SVM (Support Vector Machine) を改良した手法であり，教師なし学習で 1 クラス分類に応用した One-Class SVM などがある [51]．

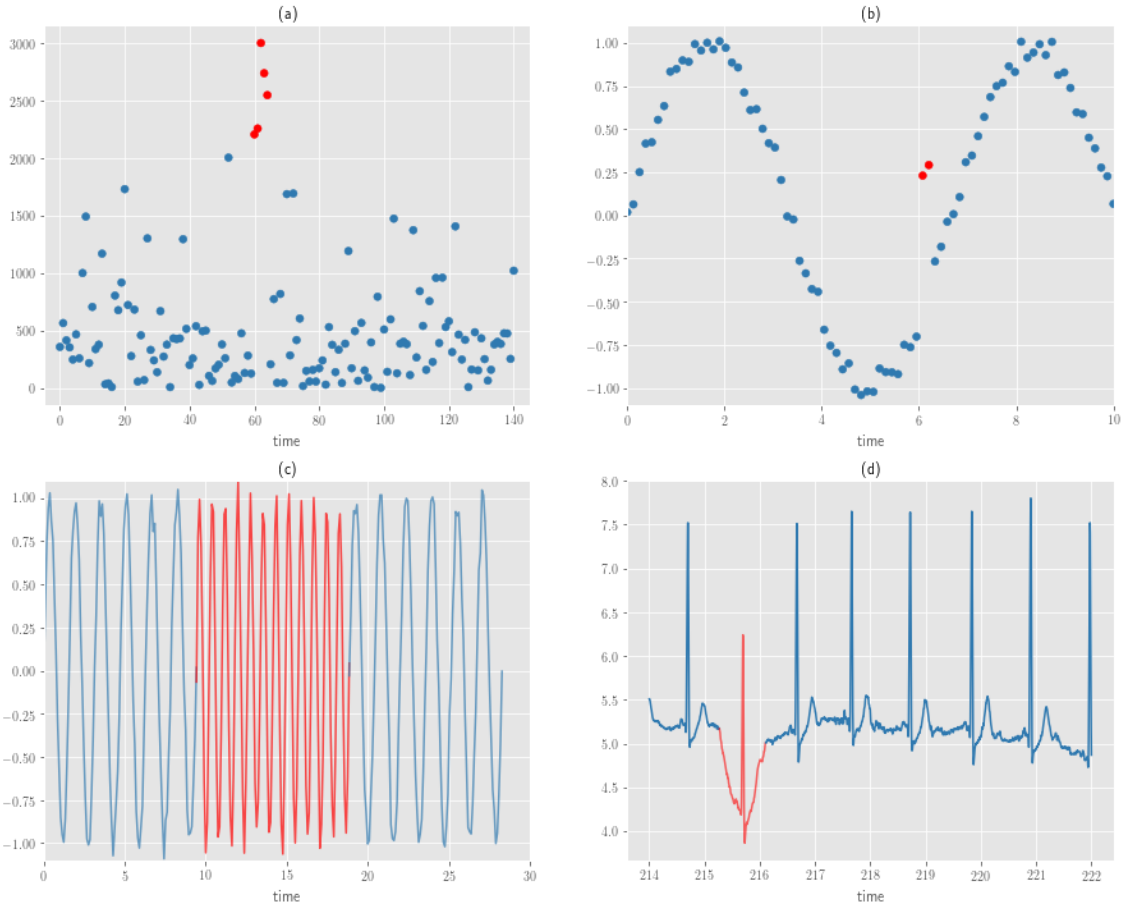

図 2.4: 時系列データにおける外れ値の例

2.6 ネットワーク計測における課題

ネットワーク計測の異常値は，計測ミスと外れ値がある．計測ミスは，ネットワーク計測時にプロセスが止まることでタイムアウトとなり，計測が終わらないことによる計測値が取れない時の値である．異常値は，統計的に離れてる値でも，ネットワーク計測では異常でない時や測定値から周期性や相関からの傾向から外れている値を定義する必要がある．

ネットワークに負荷が高い場合など，1 つのネットワーク計測の結果が外れ値であったとしても，ネットワークの構成上の異常ではない時もある．また，ネットワーク機器が故障しているにも関わらず，計測値は正常である場合である時もある．したがって，ネットワーク計測から異常検知を行うときは，ネットワークを構成するシステム全体として正常か異常かを考慮する必要がある．すなわち，ネットワーク計測は，ネットワーク環境に大きく依存する．複雑なネットワークになるにつれて，異常検知が困難になる．

既存のモニタリングツールでは，閾値による異常検知がベースであり，ネット

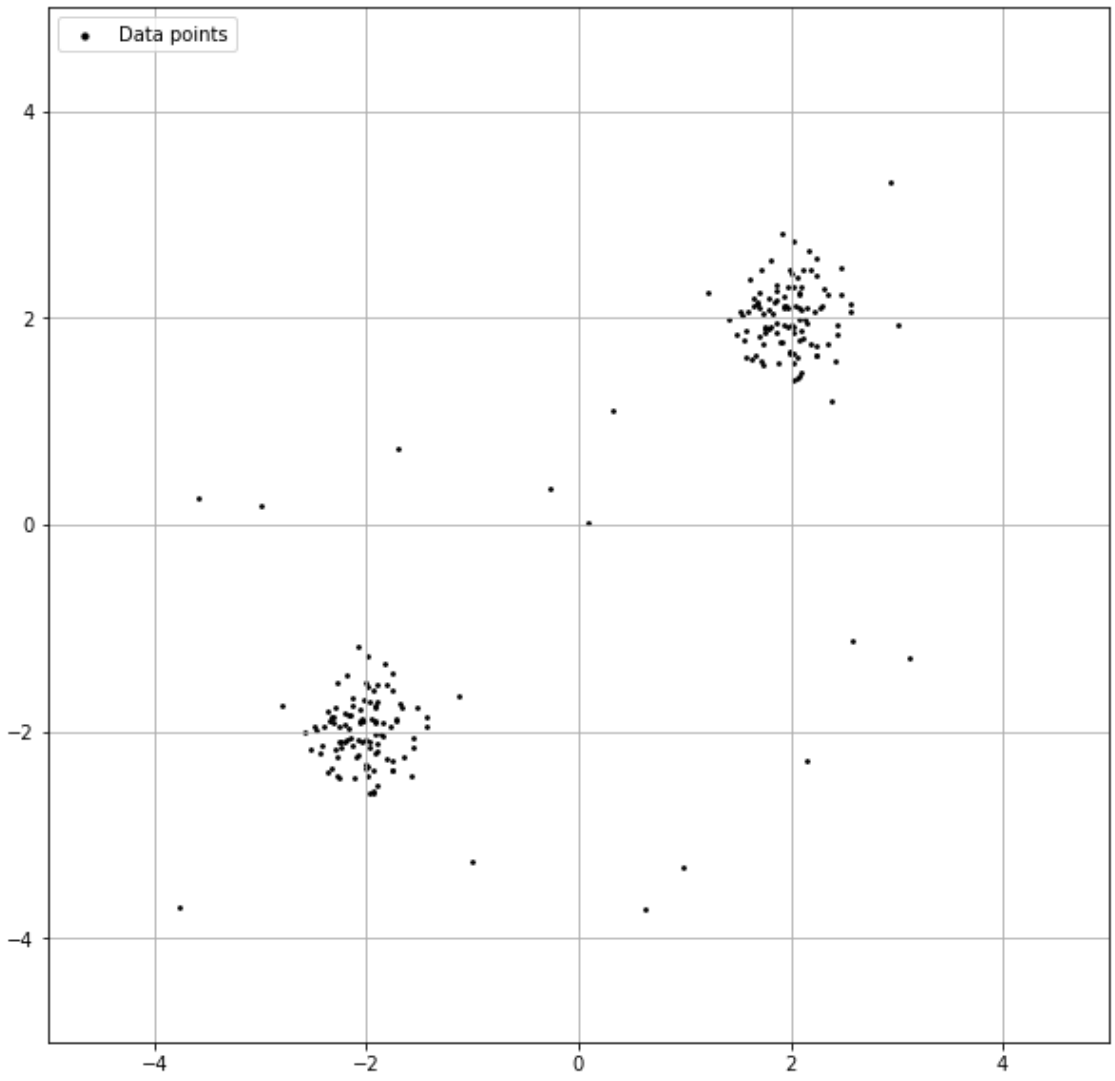

図 2.5: 相関があるデータにおける外れ値の例

ワーク計測データの特性を考慮した異常検知はできない.

第3章　関連技術と関連研究

本章では，センサデバイスを用いたネットワーク計測手法の研究，異常検知の研究，障害復旧の手法の研究について述べる．また，関連研究と本研究の差分について述べる．

3.1　センサデバイスを用いたネットワーク計測手法

本項では，様々な組織が行っているネットワーク計測についてまとめた．

ヨーロッパや北アメリカ，中東地域のローカルインターネットレジストリへのアドレスの配布配布や管理を行ってる RIPE NCC [52] は，RIPE Atlas [53] と呼ばれるプローブを世界中に配布し，インターネットの接続性と到達性に関する計測を行っている．RIPE Atlas は，グローバルネットワークに対してリアルタイムでネットワーク状態を計測し，すべての計測結果をもとに，インターネットマップやデータツール，可視化情報を提供している．RIPE Atlas は，ping や traceroute，SSL/TLS，DNS，NTP，HTTP の測定を行っている．一般的に，迅速で柔軟な接続性の確認によるネットワークの問題調査によるトラブルシューティングや RIPE Atlas のステータスチェックによるアラート機能，DNS 応答の確認，IPv6 による接続性などの用途で利用されている．

ixia [54] には，ネットワークとアプリケーションのパフォーマンス評価と監視するために XRPi Active Monitoring Probe [55] というセンサデバイスのプロダクトがある．XRPi Active Monitoring Probe は，VoIP (Voice over IP) や UC (Unified Communication)，ストリーミング，リアルタイム性を必要とするアプリケーションなどの優れたネットワークの性能を必要とするネットワークが，どれほどの性能を持っているのかを測定して報告するプローブである．実際に，ネットワークのトラフィックや応答時間をもとに，ネットワークの状態と QoE (Quality of Exprience) を測定する．XRPi Active Monitoring Probe は，様々なデバイスと連携や実データや VoIP トラフィックを生成，ウェブサービスとストリーミングの UX (User Exprience) の指数，Wi-Fi 環境の調査を行っている．

3.2　ネットワーク異常検知

今までに，ネットワーク計測の様々なメトリックを用いた異常検知の研究をが行われている．

Romain らは，RIPE Atlas から 30 分ごとの DNS ルートサーバと 15 分ごとのコラボレーションサーバへの traceroute RTT 計測の結果から，異なるリターンパスをもつプローブを利用して，ISP などのグローバルネットワークにおいての遅延検知を行った [56]．福田は，インターネットバックボーントラフィックに対して，プロトコル番号や送受信 IP アドレス，送受信 TCP/UDP ポート，TTL，TCP フラグ，パケットサイズ等の特徴量から，時系列処理・信号処理や近傍法・クラスタリング，主成分分析，統計的モデルに基づく手法，学習に基づく手法，パケットの空間パターンに基づく手法など組み合わせながら異常検知を行っている [57]．村井らは，データセンタでの SNMP を利用したバーストトラフィック検知方式を提案している [58]．100 ms 以上 10 000 ms 以下のバーストトラフィックを，SNMP のトラップで 1 秒平均の通信利用率が 80%以上の時に監視サーバに送り，SNMP のポーリングで 5 分間隔でパケットロス数を収集するで検知する方式である．鈴木らは，NetFlow によるトラフィックデータの取得と解析することで，DoS (Denial of Service attack)/DDoS (Distributed DoS) の攻撃の検出を行った [59]．NetFlow のパラメータやサンプルレートの調整を行い，UTM (Unified Threat Management) 装置等のセキュリティアプライアンスを導入しなくても，攻撃トラフィックを検知できるか検討している．UTM などのセキュリティアプライアンスは，DDoS 攻撃などが行われてしまうと，ネットワークにボトルネックを作ってしまう．阿部らは，大規模なイベントネットワークにおける Syslog からポリンシャーバンドを用いて異常検知手法を提案している [60]．Syslog のメッセージに含まれるキーワード検知や閾値による異常検知は，Syslog のフォーマットなどが決まっていないため，マルチベンダで構成されるネットワーク環境だと，ログの意味解析やキーワードによる異常検知が行えないことが多い．そのため，ポリンシャーバンドを用いて Syslog の総量による分析を行い，異常を検知行う手法を提案した．石川らは，大規模な Wi-Fi ネットワークにおけるパッシブな通信品質測定手法を提案している [61]．IEEE 802.11 フレームから NFDF (Null Function Data Frame) の再送率がフレーム誤り率との相関を調査した．Fok らは，ルータの障害やケーブルの故障，ネットワークを構成しているシステムなどのネットワーク障害に対して，様々な地点から End-to-End でアクティブ測定し，RTT からネットワーク障害の診断の自動化を行っている [62]．Guo らは，無線センサネットワークにおけるデータフュージョンアルゴリズムに基づきリアルタイム性の高い異常検知を提案した [63]．PAA (Piecewise Aggregate Approximation) によりデータを圧縮し，K-means と AIS (Artificial Immune System) により，正常と異常の分類により異常検知を行った．山村らは，ハニーポットのログデータから Change Finder を用いて新種スキャンの早期発見手法を検討している [64]．IoT 機器などの有する脆弱性の早期発見のた

め，ハニーポットの TCP/UDP のポートにおけるアクセス数から Change Finder を用いて変化点検知を行い，早期検知の検討をしている．

3.3 ネットワーク障害復旧の手法や応用

ネットワークの障害の対応として規則や知識ベース，オントロジーからトラブルシューティングする研究がある．

Espinet らは，計測プローブとネットワークのトラブルシューティングのためのフレームワークを定義することで，ネットワークのトラブルシューティングを自動化する方法を研究している [65]．Bocchi らは，CGN (Carrier-Grade NAT) に，インターネットの測定や展開，収集，分析するためのスケーラブルなアーキテクチャの mPlane [66] と，高性能パッシブ測定できる tstat [67] を使用して，ISP 間を PoPs (Point of Prsence) を測定することで計測レイヤを形成し，エンドユーザのパケットを観察することで TCP と UDP のフローに関するログをリアルタイムで計測する手法を研究している [68]．Baer らは，大規模のネットワークモニタリングおよび分析アプリケーションで，IPS の CEL (Continuous Execution Language) を使用して，運用ネットワークに対して，データ処理し分析を自動化する方法を研究している [69]．木村らは，大規模なネットワークに対し，Statistical Template Extraction (以下では，STE と呼ぶ) と Log Tensor Factorization (以下では, LTF と呼ぶ) を使用して，障害に対応する研究をしている [70]．STE では，統計的なクラスタリング手法を用いて，非構造のログメッセージからプライマリとなるテンプレートを自動で抽出し，LTF では，ログメッセージの時空間的パターンを捉える統計モデルを構築し，隠れたネットワークイベントの影響と根本的な原因を把握する．

NTT や富士通，AlaxalA は，サービス品質の維持や，サイバー攻撃への対応，サイレント障害の回避，運用の効率化などを目的に，ネットワークを構成しているシステム情報の分析から，機械学習や AI などを用いて，リアルタイム分析や可視化，異常検知ソリューション，異常の予兆検知を提案している [71–73]．

3.4 関連研究と本研究の差分

3.1 章のセンサデバイスを用いたネットワーク計測手法は，グローバルネットワークのネットワーク計測であるが，本研究では，ユーザサイドにおけるネットワーク状態計測に特化しており，ユーザサイドにおけるネットワークの異常検知を行なった．

3.2 章のネットワーク異常検知はパッシブ測定の計測結果を中心に異常検知を行っているが，本研究では，アクティブ測定の計測結果に基づいた計測から，変化点検出である Change Finder アルゴリズムを用いることで異常検知を行なった．

本研究では，3.3 章のネットワーク障害復旧の手法や応用より，ネットワーク管理者に，迅速にネットワーク状態を報告する手法の設計を検討する．ネットワーク管理者が，異常検知した際に，迅速にネットワーク状態を把握することを目標とする．

第4章　SINDANにおけるネットワーク状態計測手法

本章では，本研究で採用した先行研究の SINDAN Project のネットワーク状態計測手法について述べる．ネットワーク状態計測手法は，ユーザサイドにセンサデバイスを設置し，実際にインターネットに通信が可能であるか計測する手法である．

4.1　SINDAN Probeの計測手法

本研究は，SINDAN Project [74] の計測方式を利用した．SINDAN(Simple Integrated Network Diagnosis And Notification) Project は，ユーザサイドやエンドポイントにおけるネットワーク状態を評価し，ネットワーク運用者が，迅速に問題を把握できる手法の確立を目的としている．ネットワーク障害点を発見するため，ユーザサイドやエンドポイントからの観測を階層毎に整理している．

SINDAN Probe のネットワーク状態計測は，ユーザサイドにセンサデバイスを設置し，実際に様々な設定確認や外部のサーバと接続確認をするアクティブ計測を行い，ネットワーク状態を階層レイヤごとに整理を行い，迅速にネットワーク状態を把握する方法である．ネットワークを構成している機器ではないセンサデバイスは，SNMP や NetFlow，sFlow などの管理技術より，実際のネットワークに通信を行う計測のため，ネットワークの機器に計測のための負荷はかかるが，高負荷をかけることない．

また，管理下のネットワークにセンサデバイスを置くだけなので，導入コストが少ない．そして，センサデバイスを用いていることにより，ネットワーク機器の一部に故障してネットワーク機器の設定からは見えない障害にも対応できる．

図 4.1 に．SINDAN Probe の計測方法とネットワーク管理者への通知のイメージ図を示す．データリンク層 (datalink)，インターフェース設定層 (interface)，ローカルネットワーク層 (localnet)，グローバルネットワーク層 (globalnet)，名前解決層 (dns)，ウェブアプリケーション層 (web) の計測レイヤ設計がある [75]．センサーデバイスからの計測結果は，独自に用意した計測結果収集サーバに送りデータベースに保存される．

実際に，外部ネットワークのサーバに，センサーデバイスからの計測結果が送

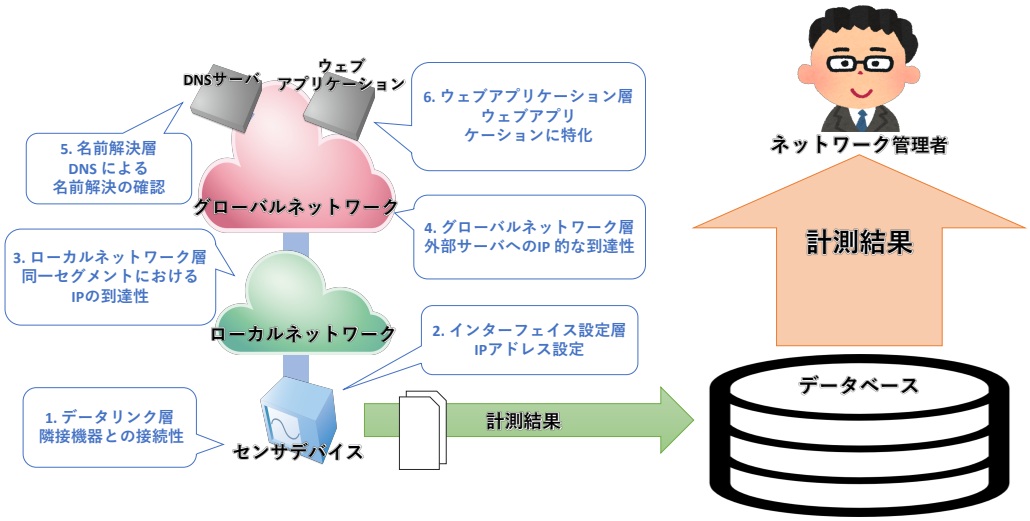

図 4.1: SINDAN Probe の計測方法とネットワーク管理者への通知

れない障害が発生しても，センサーデバイス内に計測結果が残るので，障害と独立したネットワークにサーバを立てば障害時の計測結果を分析できる．

データリンク層からウェブアプリケーション層の 1 サイクルの SINDAN Probe の計測データ量は少ないため，ネットワーク計測に影響は微小である．ネットワーク管理者は，すべての SINDAN Probe の計測結果が膨大なので，計測結果が成功しているか失敗しているかの 2 つの値の計測は確認できるが，すべての計測結果を確認するのは困難である．そこで，SINDAN Probe の計測結果から時系列のグラフで可視化を行い，ネットワーク状態の変化を詳細に把握できる．

4.2 SINDAN Probe の計測項目の各階層構造

SINDAN Probe の計測項目の各階層の計測について示す．図 4.2 に，SINDAN Probe の計測項目の各階層のイメージと，表 4.1 に，SINDAN Probe の計測項目の各階層で確認している概要を示す．階層は，データリンク層 (datalink)，インターフェース設定層 (interface)，ローカルネットワーク層 (localnet)，グローバルネットワーク層 (globalnet)，名前解決層 (dns)，ウェブアプリケーション層 (web) であり，各階層の計測結果によりどこに障害があるのか判断できる．

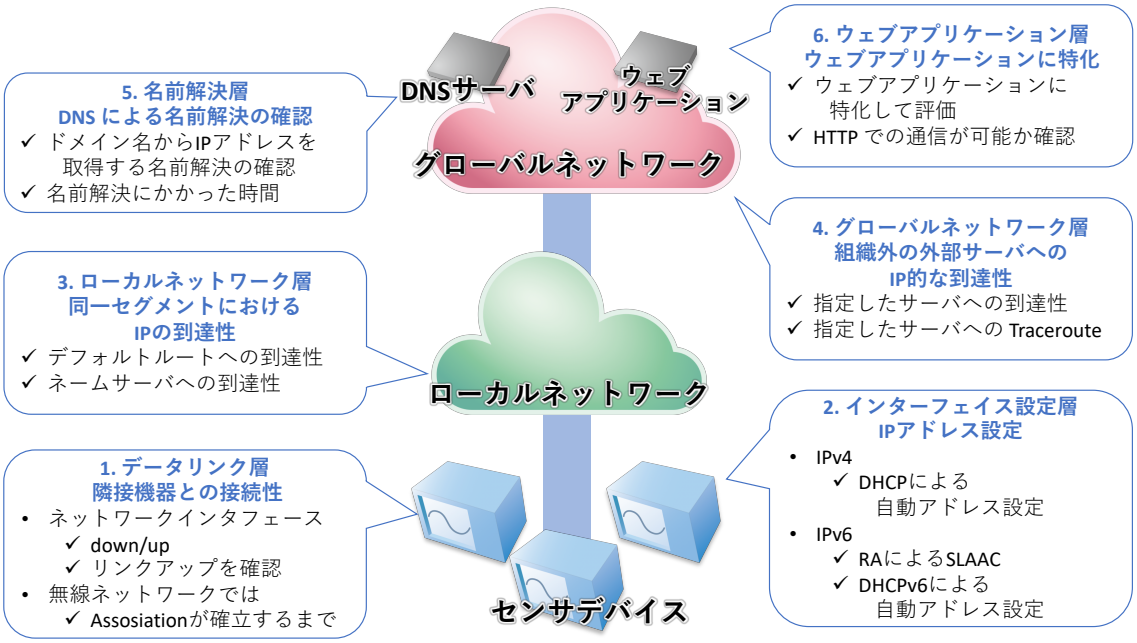

図 4.2: SINDAN Probe の計測項目の各階層

4.2.1 データリンク層

データリンク層は，TCP/IP の階層におけるネットワークインターフェイス層にあたり，隣接機器との接続性を確認するための階層である．ネットワークインターフェイスの Down/Up からリンクアップできるまでを計測する．無線ネットワーク環境では，どの無線基地局に繋がっているのかと Assosiation が確立するまで計測する．

4.2.2 インターフェース設定層

インターフェース設定層は，TCP/IP 階層モデルにおけるインターネット層にあたり，IP アドレス設定を計測する階層である．IPv4 では，DHCP (Dynamic Host Configuration Protocol) による自動アドレス設定を計測する．IPv6 では，RA (Router Advertisement) [76] による SLAAC (Stateles Address Auto Configuration) [77] の確認や DHCPv6 による自動アドレス設定を計測する．

4.2.3 ローカルネットワーク層

ローカルネットワーク層は，TCP/IP 階層モデルにおけるインターネット層のローカルネットワークへの IP 的な到達性を計測する階層である．ローカルネットワーク内にあるデフォルトルートやネームサーバへの到達性と ping コマンドにより RTT(Round Trip Time) とパケットロスト率，パス MTU を計測する．

表 4.1: SINDAN Probe の計測項目の各階層で確認している概要

階層	階層の計測概要
データリンク層 datalink	隣接機器との接続性 ネットワークインタフェースの Down/Up リンクアップを確認 無線ネットワークでは Assosiation が確立するまで
インターフェース設定層 interface	IP アドレス設定 IPv4 は DHCP による自動アドレス設定 IPv6 は RA による SLAAC と DHCPv6 による自動アドレス設定
ローカルネットワーク層 localnet	同一セグメントにおける IP の到達性 デフォルトルートへの到達性 ローカルネットワークにあるネームサーバへの到達性
グローバルネットワーク層 globalnet	組織外の外部サーバへの IP 的な到達性 指定したサーバへの到達性 指定したサーバへの Traceroute
名前解決層 dns	DNS による名前解決の確認 ドメイン名から IP アドレスを 取得する名前解決の確認 名前解決にかかった時間
ウェブアプリケーション層 web	ウェブアプリケーションに特化 ウェブアプリケーションに特化して評価 HTTP での通信が可能か確認

4.2.4 グローバルネットワーク層

グローバルネットワーク層は，TCP/IP 階層モデルにおけるインターネット層のグローバルネットワークへの IP 的な到達性を計測する階層である．ping コマンドにより RTT (Round Trip Time) とパケットロスト率，traceroute コマンドによるパス計測の到着性の確認，パス MTU を計測する．

4.2.5 名前解決層

名前解決層は，TCP/IP 階層モデルにおけるアプリケーション層の DNS (Domain Name System) よる名前解決の確認を行う階層である．アプリケーションを利用する際に必須となる機能として，ドメイン名から IP アドレスを取得する名前解決があり，名前解決できるかと名前解決するまでの時間を計測する．OS の resolver API 毎に挙動が異なることが想定されるので，DHCP/DHCPv6 等で得られた自動

アドレス設定で配布されたネームサーバとパブリック DNS サーバとの挙動に変化があるのか確認する．パブリック DNS サーバは，どの様な DNS サーバも設定可能である．今回の計測では，Google のパブリック DNS サーバ [78] と Cloudflare と APNIC が提供している DNS サーバ [79]，を利用している．また，A レコードのみや AAAA レコードのみ，双方をもつサーバドメイン名の名前解決ができるか計測する．

4.2.6 ウェブアプリケーション層

ウェブアプリケーション層は，TCP/IP 階層モデルにおけるアプリケーション層にあたり，ウェブアプリケーションに特化して計測する階層である．この層では、組織外の外部サーバに対して HTTP での通信が可能か計測する．

第5章　センサデバイスを用いたネットワーク異常検知の手法の設計

本章では，先行研究であるセンサデバイスを用いたネットワーク状態計測の結果から，複雑なネットワークの異常の検知を行う．センサデバイスを用いたネットワーク異常検知の手法について述べる．また，異常検知を行う際に，採用したアルゴリズムについて述べる．

5.1　センサデバイスを用いたネットワーク異常検知の手法

先行研究である SINDAN Probe の計測結果から，先行研究では発見が困難な障害を，検知できる手法を設計した．

センサデバイスを用いた SINDAN Probe のネットワーク状態計測の結果から，複雑なネットワークの異常の検知を行った．センサデバイスで，定位置から定期的にネットワーク計測を行うことで，計測結果を時系列に分析ができる．図 5.1 に，SINDAN Probe の計測結果をもとに異常の検知を行いネットワーク管理者へ通知するイメージ図を示す．SINDAN Probe の計測結果を時系列でグラフに表示するとともに，時系列の計測結果からネットワークの異常をアルゴリズムで検知させ，ネットワーク管理者にアラートを送ることで，発見が困難な障害に迅速に対応することができる．

ネットワーク異常の検知では，ネットワーク状態の変化を捉えることが重要である．ネットワーク状態の変化は，ネットワークの構成している機器の不調や障害を示している可能性がある．しかし，ネットワーク機器の計測のサンプリング間隔を細かく設定したり，ネットワーク機器の計測項目を増加させてしまうと，ネットワーク機器の計算リソースを多く使い，ネットワーク機器の負担が増大し，本来のネットワークシステムとしての役割に支障をきたすことに繋がる．そこで，ネットワークを構成している機器ではないセンサデバイスの計測結果からネットワーク異常の検知を行うことで，サンプリング間隔を短く設定しても，ネットワーク機器に負担が少なく，ネットワーク状態の変化を捉えることができる．

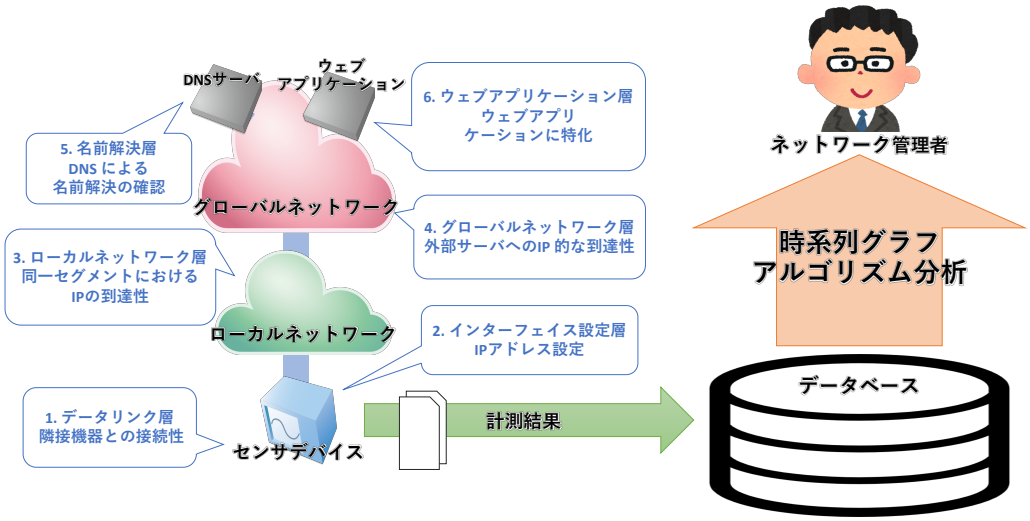

図 5.1: SINDAN Probe の計測結果に基づく異常検知とネットワーク管理者への通知

5.2 本研究において使用した異常検知アルゴリズム

本研究は，正常と異常にあらかじめクラス分けされた学習データを必要なく実装できる教師なしアルゴリズムである相関分析を行うための Local Outier Factor と，ネットワーク状態の変化を捉えることができる変化点検知のアルゴリズムである Change Finder を使用した．教師あり学習をするためには，ネットワーク計測の結果をぞれぞれ正常と異常にクラス分けする必要があるが，ネットワーク計測の結果を正常と異常のクラスに定義することは難しい．また，既知のネットワーク状態の異常にしか検知できなくなる可能性があるので，適切ではない．

5.3 Local Outlier Factor

Local Outlier Factor (以下では，LOF と呼ぶ) [80] は，相関するデータの集まりから外れ値を検出するアルゴリズムである．隣接するデータに関して所与のデータポイントの局所的な密度偏差を計算する教師なし異常検出方法である．図 5.2 に，LOF アルゴリズムのイメージ図を示す．LOF は，あるデータの一点から近傍 k 個の点といかに密接であるのかを表す局所密度距離 (Local reachability density) と言われる指標に注目することで，着目点を中心に，近傍に加えて，比較される相手方の局所密度距離をも考慮して外れ値を検出する [81]．

観測点 x' の $k > 1$ である k 近傍を $N_k(x')$ と表すと，$N_k(x')$ の要素をすべて含み

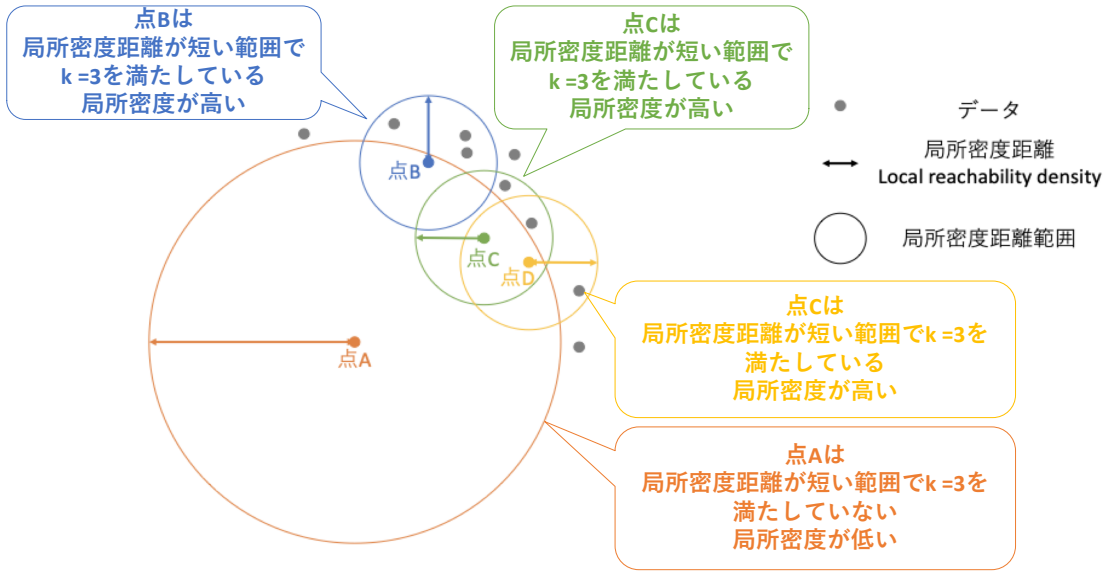

図 5.2: LOF アルゴリズムの動作

x' を中心とする最小の球の半径を $\epsilon_k(x')$ と表すことができる．このとき，局所密度距離 d が定義された M 次元空間において，u から u' への近傍有効距離 $\ell_k(u \to u')$ は，

$$\ell_k(u \to u') \equiv \begin{cases} \epsilon_k(u') & if(u \in N_k(u' \text{かつ } u' \in N_k(u))), \\ d_k(u, u') & otherwise. \end{cases}$$

と定義できる．

局所密度距離を，使用すると，局所外れ値の考えに基づく異常度 α_{LOF} は，

$$\alpha_{LOF} = \frac{1}{k} \sum_{x \in N_k(x')} \frac{d_k(x')}{d_k(x)}$$

と定義できる．

ただし，一般的に，$d_k(u)$ は，近傍有効距離を u の周りの k 近傍にわたり平均したもので，

$$d_k(u) = \frac{1}{k} \sum_{x \in N_k(x)} \ell_k(u \to u')$$

と定義できる．

5.3.1 LOF の利用に適したデータ

図 5.2 に，LOF アルゴリズムのテストデータの結果を示す．LOF アルゴリズムのテストデータの結果の横軸と縦軸に 2，-2 とに比重を置いたデータの値にであり，LOF の n_neighbor は 20 とし，n_neighbor は，考慮される近傍の数である．

$(x, y) = (-2, -2), (2, 2)$ を中心に島ができ，この 2 つ島から離れた値のデータは外れ値とみなすと，各島の近傍の点では，近傍に密接しているので，Outlier Scores は小さく，各島から離れる点につれて，Outlier Scores は大きくなっている．

5.4 Change Finder

Change Finder [82] は，変化点検出 (Change point detection) であり，スコアが大きいほど変化点である可能性を示すアルゴリズムである．図 5.4 に，Change Finder の処理の流れを示す．統計に基づく方式よりも迅速に異常を検知できるので，時系列データに対してリアルタイム性が良い．Change Finder は，時系列モデルの 2 段階学習 (Two-stage learning of time series models) に基づく方式を用いている [83]. AR モデルという自己回帰モデル (Auto Regression model) を利用して計算すると計算量が多くなるが，SDAR モデルというオンライン忘却学習モデル (Sequentinally Discounting AR model) の忘却パラメータと最尤推定値を用いることで，AR モデルを推定することができるので計算量が削減する．

Change Finder のメトリックとして，忘却パラメータである $r(0 < r < 1)$，AR モデルの次数である Order，外れ値計算スコアの移動平均平滑化する範囲である Smooth の T がある．

初期値の平均値が 0 であるような，連続値をとる時系列変数を $\{z_t : t = 1, 2, ...\}$ で表すと，z_t は，d 次元のベクトルになり，k 次の AR モデルは，

$$z_t = \sum_{i=0}^{k} \omega_i z_{t-i} - \varepsilon$$

となる．$\omega_i \in \mathcal{R}^{d \times d} (i = 1, ..., k)$ は，d 次パラメータ行列であり，

$$z_{t-k}^{t-1} = (z_{t-1}, z_{t-2}, ..., z_{t-k})^T \in \mathcal{R}^{d \times k}$$

と記す．ε は平均，共分散行列 Σ のガウス分布 $\mathcal{N}(0, \Sigma)$ に従うガウス変数である．実際に観測される時系列を $\{x_t : t = 1, 2, ...\}$ とすると，

$$x_t = z_t + \mu$$

である．また，

$$x_{t-k}^{t-1} = (x_{t-1}, x_{t-2}, ..., x_{t-k})^T$$

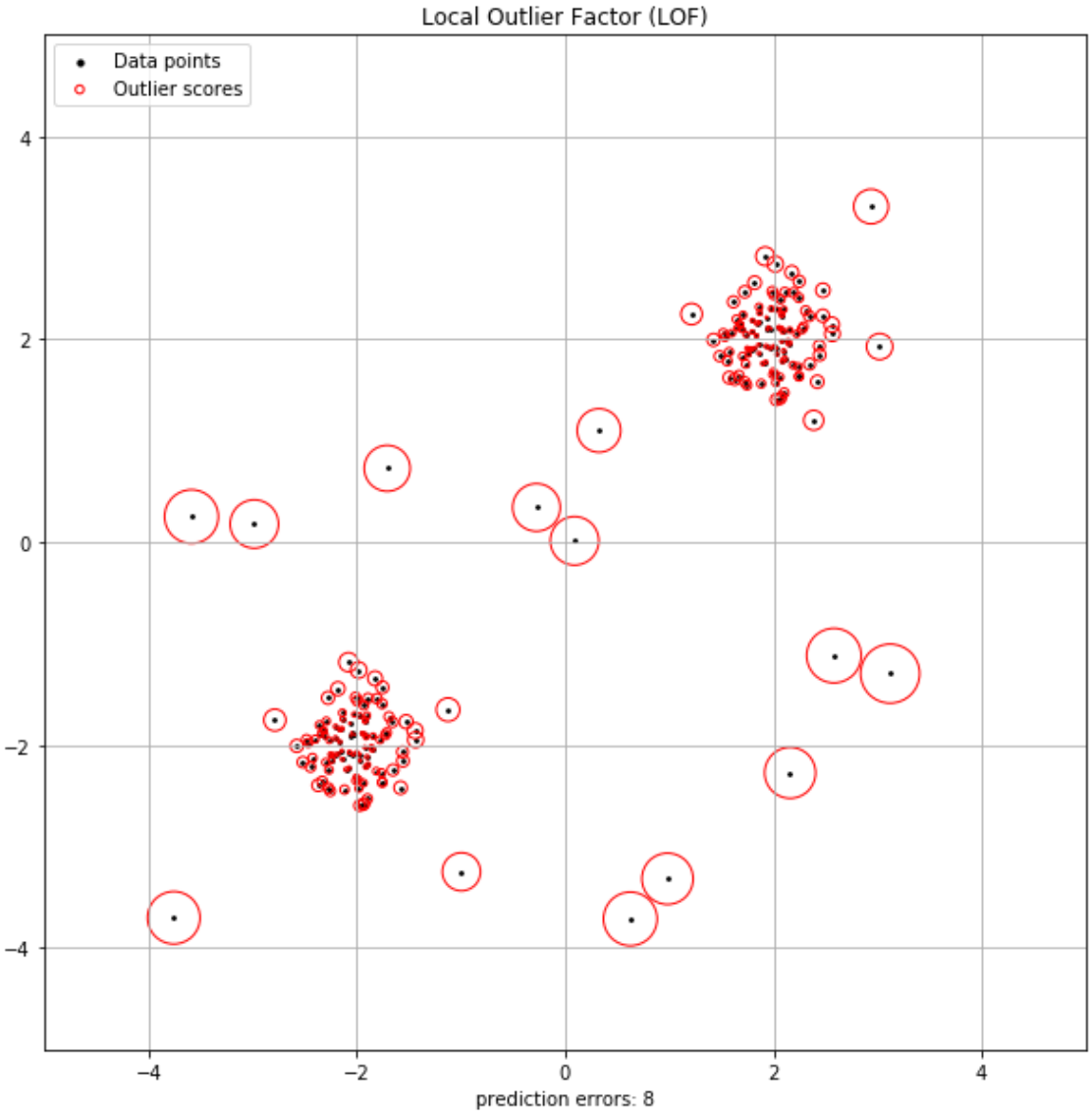

図 5.3: LOF アルゴリズムの検出結果

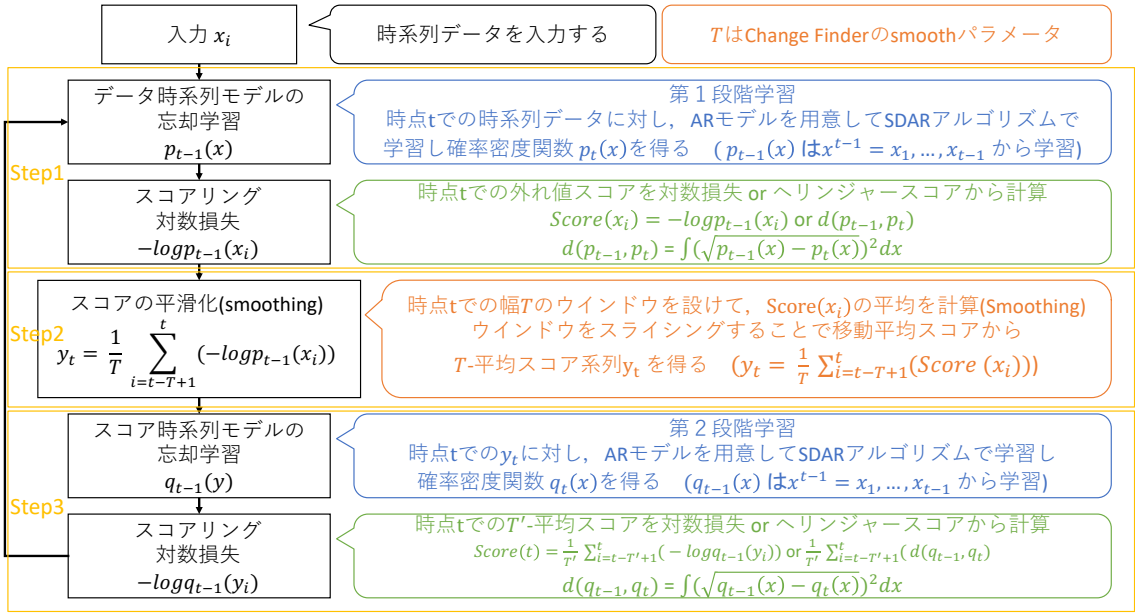

図 5.4: Change Finder の処理の流れ

とおくことで，AR モデルによって表される x_t の確率密度関数は，

$$p(x_t|x_{t-k}^{t-1} : \theta) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x_t - w)^T\Sigma^{-1}(x_t - w)\right) \tag{5.1}$$

になる．
ここに，$w = \sum_{i=k}^{k} \omega_i(x_{t-i} - \mu) + \mu$ であり，パラメータをまとめて，
$\theta = (\omega_1, \ldots, \omega_k, \mu, \Sigma)$ とする．
2 段階学習に基づく変化点スコアリングの基本のステップは，x_i を入力して，

1. データ時系列モデルの忘却学習を行い，データの忘却学習をもとにデータをスコアリングをする
2. スコアの平滑化を行う
3. スコア時系列モデルの忘却学習を行い，スコアの忘却学習をもとにのスコアをスコアリングをする
4. $i = i + 1$ を行い，x_i を入力して 1 に戻る

である．
第 1 ステップは，第一段階学習であり，時系列データの確率モデルとして AR モデルを用意して，これを SDAR アルゴリズムを用いて学習する．学習で得られた確率密度関数の列を $\{p_t(x) : t = 1, 2, \ldots\}$ とすると，p_{t-1} は，$x^{t-1} = x_1, \ldots, x_t - 1$

から学習された確率密度関数である．各地点 t ののデータ x_t の外れ値スコアを対数損失の

$$Score(x_t) = -\log p_{i-1}(q_t)$$

または，ヘリンジャースコアの

$$\begin{aligned} Score(x_t) &= d(p_{t-1}, p_t) \\ &= \int (\sqrt{p_{t-1}(x)} - \sqrt{p_t(x)})^2 dx \end{aligned}$$

で計算する．

第 2 ステップは，平滑化 (Smoothing) であり，T を与えられた整数として，幅が T のウインドウを設け，ウインドウ内のデータに関して第 1 ステップで求めた外れ値スコアの平均を計算する．さらに，スライディングウインドウを行うことで，移動平均スコアの時系列データ $y_t : t = 1, 2, ...$ を作成する．これは，スコア系列 $\{Score(x_i) : i = t - T + 1, ..., t\}$ に対して T-平均スコア系列 y_t をスコア移動平均として

$$y_t = \frac{1}{T} \sum_{i=t-T+1}^{t} Score(x_i)$$

と定義する．

第 3 ステップは，第二段階学習であり，第 2 ステップで求めた新しい時系列データ $y_t : t = 1, 2, ...$ に対して AR モデルを用いてモデル化し，SDAR アルゴリズムを用いて学習する．得られる確率モデルの時系列を $\{q_t : t = 1, 2, ...\}$ とする．さらに，T'-平均スコアを対数損失の

$$Score(t) = -\log p_{q-1}(y_t)$$

または，ヘリンジャー距離の

$$\begin{aligned} Score(t) &= d(q_{i-1}, q_i) \\ &= \int (\sqrt{q_{i-1}(y)} - \sqrt{q_i(y)})^2 dx \end{aligned}$$

で計算する．$Score(t)$ の値が高いほど時刻 t が変化点の度合いが高いと見なすことができる．

Change Finder は，第 1 段階学習では，時系列中の外れ値しか検知できないが，外れ値スコアの平滑化を通じて，ノイズに反応した外れ値を除去し，第 2 段階学習によって本質的な変動のみを検出できる．

SDAR アルゴリズムは，オンライン忘却学習アルゴリズムであり．AR モデルのパラメータ推定アルゴリズムである．総数が n の時系列データ $x_1, ..., x_n$ が与えられたとする．$x_t = z_t + \mu$ となる変換を行うことにより，$z_1, ..., z_n$ に関する尤度は，

$$\prod_{t=1}^{k} p(z_t|\theta) \cdot \prod_{t=k+1}^{n} p(z_t|z_{t-k}^{t-1} : \theta)$$

である．よって，対数尤度をとると，

$$\sum_{t=1}^{k} \log p(z_t|\theta) + \log p(z_t|z_{t-k}^{t-1} : \theta)$$

である．n は，k より十分大きいと仮定し，第 2 項と 5.1 から，

$$-(n-k)\log((2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}) - \frac{1}{2}\sum_{t=k+1}^{n}(z_t - \sum_{i=1}^{k}\omega_i z_{t-1})^T\Sigma^{-1}(z_t - \sum_{i=1}^{k}\omega_i z_{t-i})$$

を近似する．ω について偏微分して 0 とおくことにより，対数尤度を最大する $\omega_i(i = 1, ..., k)$ は，

$$\sum_{i=1}^{k}\omega_i C_{j-1} = C_j(j = 1, ..., k) \tag{5.2}$$

を満たさなければいけない．C_j は，自己共分散関数 (Autocovariance Function) であり，

$$C_j = \frac{1}{n-k}\sum_{t=k+1}^{n} z_t z_{t-j}^T = \frac{1}{n-k}\sum_{t=k+1}^{n}(x_t - \mu)(x_{t-j} - \mu)^T$$

である．ここで，すべての整数 s に対して，

$$C_s = C_{-s}^T$$

を満たす．

5.2 は，Yule-Walker の方程式 (Yule-Walker Equattion) から，μ と Σ の最尤推定値 $\hat{\mu}$ と $\hat{\Sigma}$ は，

$$\hat{\mu} = \frac{1}{n-k}\sum_{t=k+1}^{n} x_t$$

と求めることができる．

C_j の μ に $\hat{\mu}$ を代入したときの Yule-Walker の方程式の解を $\hat{\omega_1}, ..., \hat{\omega_k}$ とすると，

$$\begin{aligned}\hat{\Sigma} &= \frac{1}{n-k}\sum_{t=k+1}^{n}(z_t - \sum_{i=1}^{k}\hat{\omega_i} z_{t-i})(z_t - \sum_{i=1}^{k}\hat{\omega_i} z_{t-i})^T \\ &= \frac{1}{n-k}\sum_{t=k+1}^{n}(x_t - \hat{\mu} - \sum_{i=1}^{k}\hat{\omega_i}(x_{t-i} - \hat{\mu}))(x_t - \hat{\mu} - \sum_{i=1}^{k}\hat{\omega_i}(x_{t-i} - \hat{\mu}))^T\end{aligned} \tag{5.3}$$

と求めることができる．

SDAR アルゴリズムは，5.3 のような計算を逐次的するために，パラメータや計算に必要な統計量を，現在の値と新しい新しい値の $(1-r) : r$ の比の重みつき平均の形で更新する．$r(0 < r < 1)$ は，忘却パラメータ (Discounting Parameter)

であり，r が小さいほど SDAR アルゴリズムは過去のデータの影響を大きく引きずる．

SDAR アルゴリズムの流れは，忘却パラメータ r を与え，パラメータ $\hat{\mu}, C_j, \hat{\omega_j}(j = 1, ..., k), \hat{\Sigma}$ を初期化して，時系列データ x_t をから

$$\begin{aligned} \hat{\mu} &= (1-r)\hat{\mu} + rx_t \\ C_j &= (1-r)C_j + r(x_t - \hat{\mu})(x_{t-j} - \hat{\mu})^T \end{aligned}$$

を計算し，Yule-Walker の方程式から，

$$\sum_{i=1}^{k} \omega_i C_{j-i} = C_j (j+1, ..., k)$$

の解を，$\hat{\omega_1}, ..., \hat{\omega_k}$ とおき，

$$\begin{aligned} \hat{x_t} &:= \sum_{i=1}^{k} \hat{\omega_i}(x_{t-i} - \hat{\mu}) \\ \hat{\Sigma} &:= (1-r)\hat{\Sigma} + r(x_t - \hat{x_t})(x_t - \hat{x_t})^T \end{aligned}$$

となる．

AR モデルは，一般的に時系列の定常性を仮定したモデルであるが，実際の変化点検知に適用するときは，非定常性を仮定しなければならない．定常性とは，初期値に依存しないことである．SDAR アルゴリズムでは，忘却パラメータを取り入れることで，形式的に，AR モデルを用いて非定常なモデルの学習を実現している．統計的検定に基づく方式は，データ数の二乗のオーダであるのに対し，SDAR アルゴリズムの計算量は，線形のオーダであるので，Change Finder の計算量は，データ数に関して線形のオーダに抑えられる．

5.4.1 Change Finder の利用に適したデータ

Change Finder は，変化点検出であり，緩やかな変化ではなく，データの属性が急激な変化のデータに適している．また，誤差の平方和を最小化して変化点検出する Guralnik and Srivasava に基づく手法（GS）[84] や GS に AR モデルを導入した確率的コンプレキシティ（Stochastic Complexity）の最小化して変化点検出する手法より変化点検出に遅れが出にくい．表 5.1 に，Change Finder に入力するテストデータを示す．Change Finder に，平均 1000 から，10，9，8，7，6，5，4，3 と加算を行い，平均が 1052 までの標準偏差 1 の正規分布に従う乱数を 1000 件ずつの出力を入力した．

図 5.5 に，上に Change Finder が得意とする入力データ，下に Change Finder のスコアの結果を示す．上段の図は，横軸にデータの個数，縦軸にデータの値である．下段の図は，横軸にデータの個数，縦軸に Change Finder のスコアである．

データ通し番号の先頭	平均	標準偏差	出力数
1	1000	1	1000
1001	1010	1	1000
2001	1019	1	1000
3001	1027	1	1000
4001	1034	1	1000
5001	1040	1	1000
6001	1045	1	1000
7001	1049	1	1000
8001	1052	1	1000

表 5.1: Change Finder に入力するテストデータ

Change Finder のメトリックは，忘却パラメータ r は 0.03，AR モデルの次数 $Order$ は 1，外れ値計算スコアの移動平均平滑化する範囲である Smooth の T は 10 とした．

Change Finder のスコアを見ると，入力データが 7000 までの変化点の前後の差が大きい箇所は異常検知できるが，8000 の箇所などの変化点の前後の差が小さい箇所では，外れ値により異常検知ができないことがある．このことから，大きく変化点はスコアが大きくなるが，小さい変化点はスコアが小さくなるので，異常検知できない．

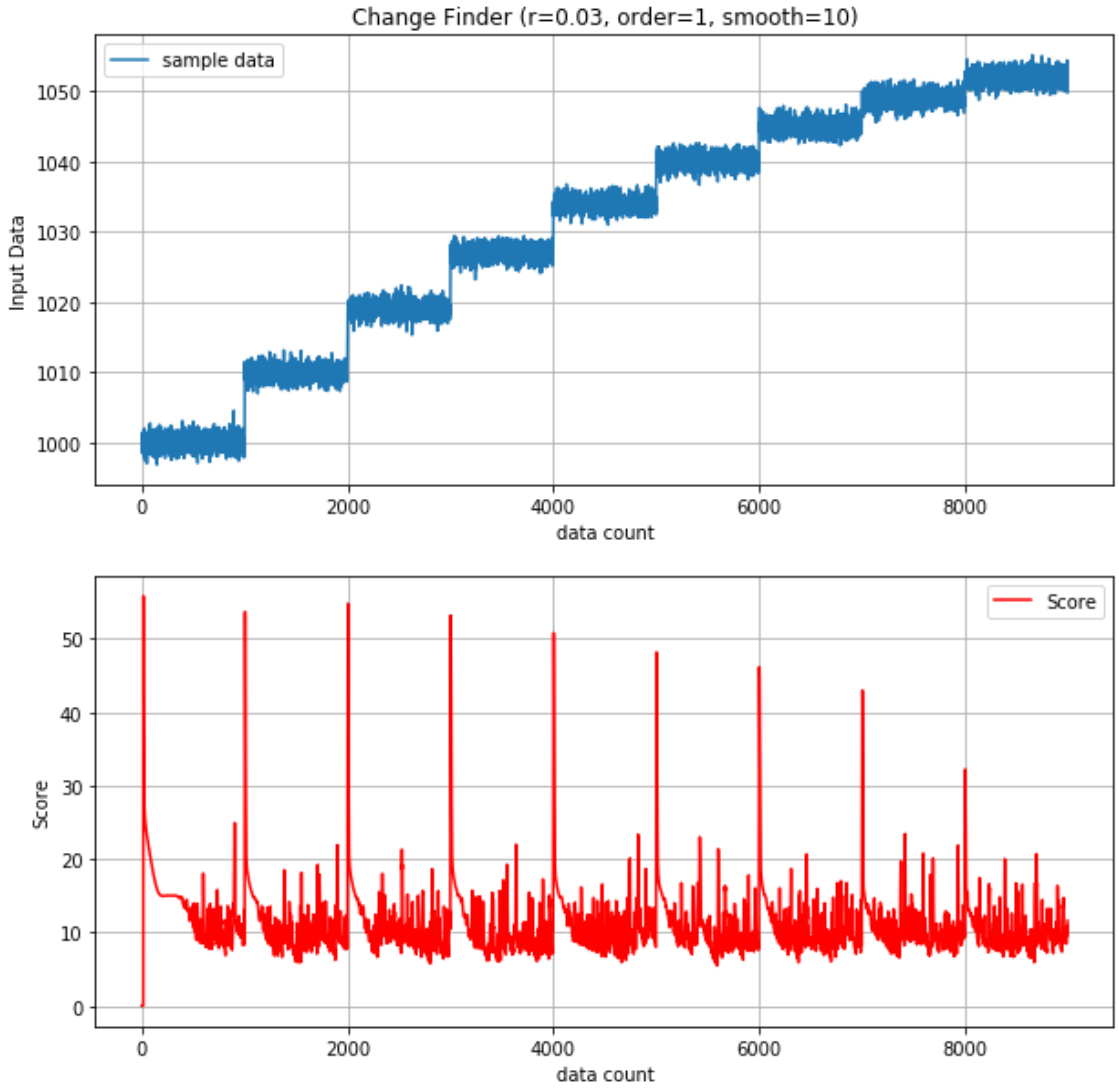

図 5.5: [上] 入力データ [下]Change Finder のスコア

第6章　センサデバイスを用いたネットワーク異常検知の手法の評価

本章では，実験ネットワーク環境の概要とネットワーク負荷実験の結果と異常検知の評価を述べる．

6.1　実験ネットワーク環境

本研究は，実際のネットワーク環境に近い環境を再現し，外的因子を少なく制御できる実験ネットワーク環境を作成した．図 6.1 に，実験ネットワーク環境と表 6.1 に，実験ネットワーク環境を構成している機器を示す．実験を行なった期間は，24 時間である．

アクセスポイント (AP : Acsess Point) に Cisco Aironet 3602i(c3602i) を使用し，ユーザサイドとして無線 LAN 環境を PHY mode 802.11n で提供した．ローカルネットワークは，ルータ (Router) の Juniper SRX650(SRX650)，スイッチ (Switch) の Arista 7050(Arista 7050) と DELL Networking S4180on(4180on)，Juniper EX4200(EX4200)，Cisco Catayst 3650(cat3650)，アクセスポイントの c3602i で構築した．アクセスポイントの c3602i とスイッチの EX4200 は 1 Gbps 回線で繋がっており，ローカルホストのスイッチ間は，10 Gbps から 80 Gbps 回線で繋がっている．グローバルネットワークへは，ルータの SRX650 を介して接続されている．KVM で構成した Cloud 上の VM のサーバでは，DHCP サーバに kea [85]，DNS サーバに bind9 [86] を使用した．本研究は，センサーデバイスからの計測結果を，SINDAN Project の管理下のグローバルネットワークからアクセスできるサーバに送られてデータベースに保存されるようにした．また，大規模なネットワーク環境を再現するために，複数台のベンダーが異なるスイッチを利用した．

SINDAN Probe は，センサデバイスとして計測を行うために，Raspberry Pi [87] に USB の無線 LAN アダプタのドングル GW-450D2 [88] をつけて 2.4 GHz 帯と 5 GHz 帯で通信できるようにし，SINDAN Client [89] のスクリプトを動かした．また，負荷実験用も Raspberry Pi に USB の無線 LAN アダプタのドングル GW-450D2 をつけ，2.4 GHz 帯と 5 GHz 帯の両方で通信が行えるように設定した．

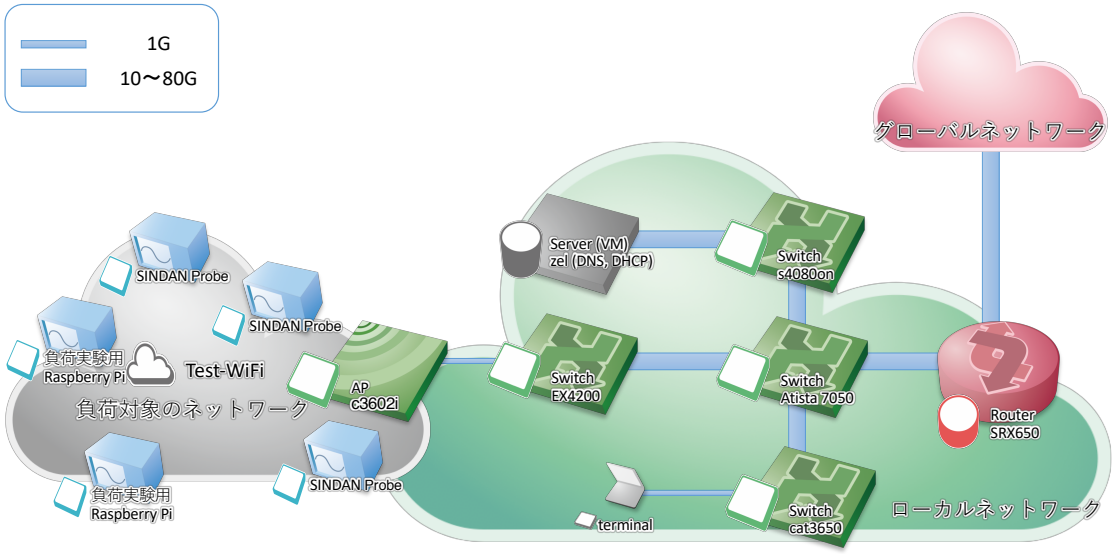

図 6.1: 実験ネットワーク環境

種類	名称	ネットワーク機器
Router	Juniper SRX650	Juniper SRX650
Switch	Arista 7050	Arista 7050
	4180on	DELL Networking S4810on(cumulus)
	EX4200	Juniper EX4200
	cat3650	Cisco Catayst 3650
Acsess Point	c3602i	Cisco Aironet 3602i
Server	DNS, DHCP	KVM で構成させた Cloud 上の VM

表 6.1: 実験ネットワーク環境を構成機器

SINDAN Probe の計測項目は，MTU(Maximun Transmion Unit) などの計測に時間がかかるものがあるので，SINDAN Probe を安定して 1 分おきに計測できるように，3 台の SINDAN Probe を用いた．実験期間が 24 時間なので，1440 回の計測結果がある．

スイッチに繋がってる実験環境以外のセグメントの影響を減少させるために，実験ネットワーク環境の専用セグメントを VLAN で作成した．VLAN (Virtual Local Area Network) は，1 台の L2 スイッチを複数の仮想 L2 セグメントに分割して利用できる技術である．また，複数の物理スイッチを組み合わせて 1 つの物理 Switch とし，これを複数の仮想 L2 セグメントに分割もできる．VLAN を使用することで，実験と関係のないトラフィックを分割し，ブロードキャストドメインを限定できる．

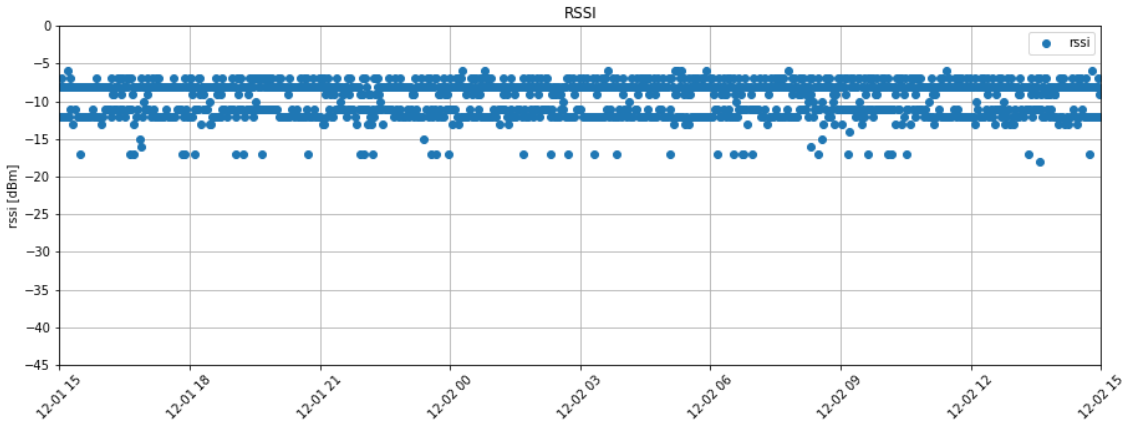

図 6.2: SINDAN Probe による AP の RSSI 計測結果

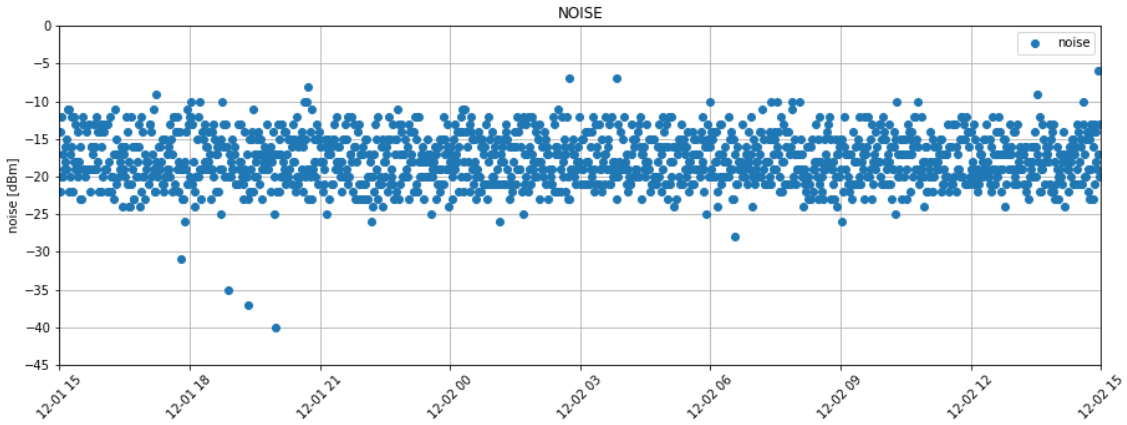

図 6.3: SINDAN Probe による AP の NOISE 計測結果

無線 LAN 環境の外乱を減らすために，AP と 3 台の SINDAN Probe，2 台の負荷実験用 Raspberry Pi に対する 20 dB 程度の電磁波シールドを作成した．

RSSI(Received Signal Strength Indicator) は，受信信号強度インジケータであり，デバイスがアクセスポイントからどの程度の信号を受信している指標である．NOISE は，デバイスが受信してるノイズを示す指標である．dBm は，電力レベルを対数で表しており，RSSI と NOISE は，チップセットから計測できる相対的な指標である．SNR(Signal-to-Noise Ratio) とは，SN 比であり，受信信号強度インジケータと背景ノイズの差である．SNR の比が高いと電波品質が良く，低いと電波品質が悪いことを示す．

図 6.2 と 6.3，6.4 に，実験ネットワーク環境から SINDAN Probe による RSSI の計測結果と NOISE の計測結果，SNR の計算結果を示す．横軸は時刻，縦軸はそれぞれ RSSI の値，NOISE の値，SNR の値である．

本研究は，AP と SINDAN Probe，負荷実験用の Raspberry Pi 以外の外来を少なくするために，それぞれの機器が近接している環境なので，RSSI と NOISE が

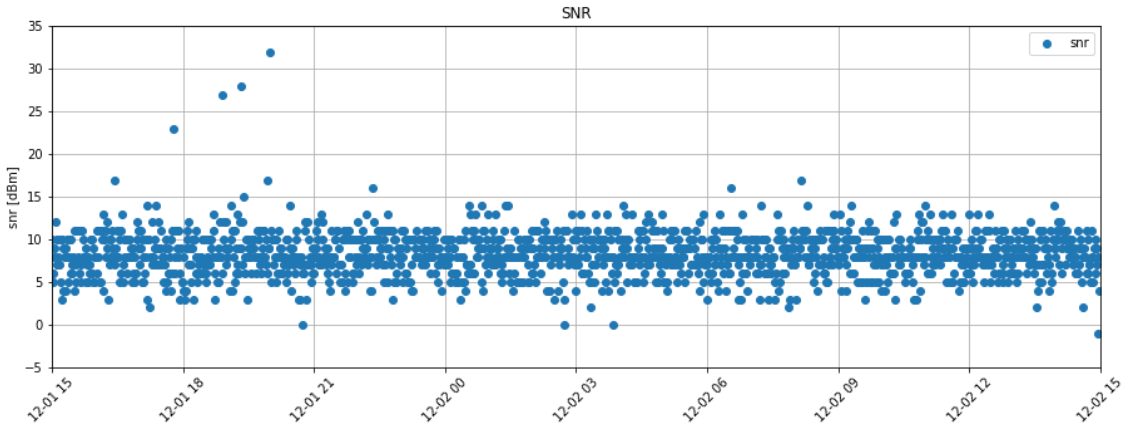

図 6.4: SINDAN Probe による AP の SNR 計算結果

非常に高い値なっており SNR の差が低い結果になっている．また，測定値の欠損により SNR が 0 以下の低い結果がある．実験ネットワーク環境からの SINDAN Probe による RSSI は，−20 dBm から −5 dBm の間で安定しているので，AP と 3 台の SINDAN Probe，2 台の負荷実験用も Raspberry Pi が受ける外乱は少ないと考えられる．

6.1.1 ネットワーク負荷実験

見つけにくいネットワーク障害の事例として，ネットワークの一部区間のみに高負荷がかかり，インターネットにつながりにくいというものがある．この障害の原因は主に，帯域がボトルネックになってスループットが低くなることで起きる．この障害は，ネットワークの一部区間のジッタが不安定になったり，パケットロス率が高くなることで発見できる．本実験は，無線 LAN 環境の悪化によるインターネットにつながりにくい障害を再現した．

無線 LAN 環境の悪化を検知するため，負荷実験用の Raspberry Pi 間で，高負荷な通信を行い，SINDAN Probe の計測手法を用いて検知する実験を行った．高負荷をかけるために，iperf3 コマンドを利用した．iperf3 は，ネットワークのスループット計測を行うコマンドであるが，実際に帯域に負荷をかけて計測を行うので，同じ帯域を使用してるデバイスにも影響する．ゆえに，無線 LAN 環境区間で iperf3 をすると，無線 LAN 環境に負荷をかけることができる．負荷実験用の Raspberry Pi の 1 つを iperf3 のクライアント，もう 1 つを iperf3 のサーバに設定した．表 6.2 に，負荷実験用の Raspberry Pi 同士で実行した iperf3 のタイムスケジュールを示す．iperf3 コマンドのオプションにより TCP で通信し，負荷の度合いを変化させながら行なった．負荷の度合いは，小負荷，中負荷，最大負荷を想定し，計測によって得られた最大のスループットが，約 15 Mbps であることから，小程度の負荷を 3 Mbps のスループット，中程度の負荷を 5 Mbps のスループットと設定した．

また，負荷のタイミングをずらし，アルゴリズムが周期的なパターンを学習をしないようにした．iperf3 の負荷の度合いを，IDLE は帯域に負荷をかけてない状態，3M はスループットを 3 Mbps の状態，5M はスループットを 5 Mbps の状態，MAX はスループットを指定せずに最大限の負荷をかけた最大の負荷状態である．最大の負荷のレートは，11 Mbps から 20 Mbps であった．

開始時間 [s]	継続時間 [s]	iperf3 の負荷の度合い
1	60	IDLE
61	60	MAX
121	60	IDLE
181	60	3M
241	60	5M
301	60	IDLE
361	60	MAX
421	60	5M
481	60	MAX
541	60	IDLE
601	60	3M
661	60	5M
721	60	IDLE
781	60	3M
841	60	IDLE
901	60	MAX
961	60	IDLE
1021	60	MAX
1081	60	5M
1141	60	3M
1201	60	IDLE
1261	60	MAX
1321	60	5M
1381	60	IDLE

表 6.2: iperf3 による負荷実験のタイムスケジュール

6.2 ネットワーク状態計測のメトリック選定

SINDAN Probe のネットワーク状態計測結果から，ネットワーク異常検知に使用する計測のメトリックを選定した．SINDAN Probe のネットワーク状態計測は，データリンク層，インターフェース設定層，ローカルネットワーク層，グローバルネットワーク層，名前解決層，ウェブアプリケーション層がある．どの階層の計測メトリックを用いれば良いのか選定した．

ローカルネットワーク層は，ローカルネットワークの状態を計測する層であるので，ローカルネットワーク層の計測項目からメトリック選定を行なった．データリンク層の計測結果は，無線 LAN 環境のアソシエーションについては意図的に障害を起こしてないので本実験では選ばなかった．インターフェイス設定層は，主にアドレス設定を計測するので本実験では選ばなかった．名前解決層とウェブアプリケーション層は，本実験では名前解決やウェブアプリケーションに意図的に障害を起こしてないため本実験では選ばなかった．グローバルネットワーク層は，本実験ではグローバルネットワークの異常を考慮した実験ではないので選ばなかった．

ローカルネットワーク層の計測項目は，IPv4 と IPv6 によるローカルネットワークルータとローカルネットワーク DNS サーバに対する，到達性を計測している．本実験では，ローカルネットワークルータとローカルネットワーク DNS サーバへの計測結果が変化する実験を行なっていたので，ローカルネットワークルータに対する計測のメトリックを選択した．また，IPv4 と IPv6 による計測結果に対しても，計測結果が変化する実験を行なっていたので，現在のインターネットで IPv4 が主流なので，IPv4 による計測のメトリックを選択した．IPv6 による計測も行なっているので，IPv6 による計測のメトリックを選択もできる．ping による到達性と 10 回の計測結果から最小値と最大値，平均，標準偏差を求めている．本研究では，平均と標準偏差のメトリックを選択した．

v4rtt_router_ave は，IPv4 デフォルトルータまでの ping 10 回分の平均値の計測結果である．v4rtt_router_dev は，IPv4 デフォルトルータまでの ping 10 回分の標準偏差の計測結果である．

図 6.5 と 6.6 に，実験ネットワーク環境で計測した v4rtt_router_ave の結果と v4rtt_router_dev の結果を示す．横軸は時刻，縦軸は v4rtt_router_ave の計測結果，v4rtt_router_dev の計測結果である．

v4rtt_router_ave は，ネットワーク状態の RTT，v4rtt_router_dev は，ネットワーク状態のジッタを表している．

v4rtt_router_ave は，平均を取っているが，稀に 2500 ms や 4500 ms 付近の非常に高い値を取り，異常検知において支配的になる．v4rtt_router_dev は，v4rtt_router_ave より安定している値である．

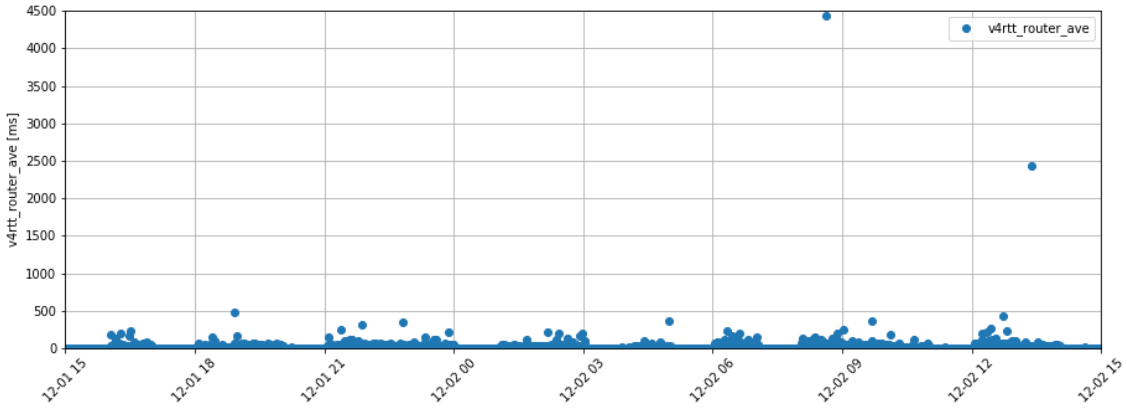

図 6.5: v4rtt_router_ave

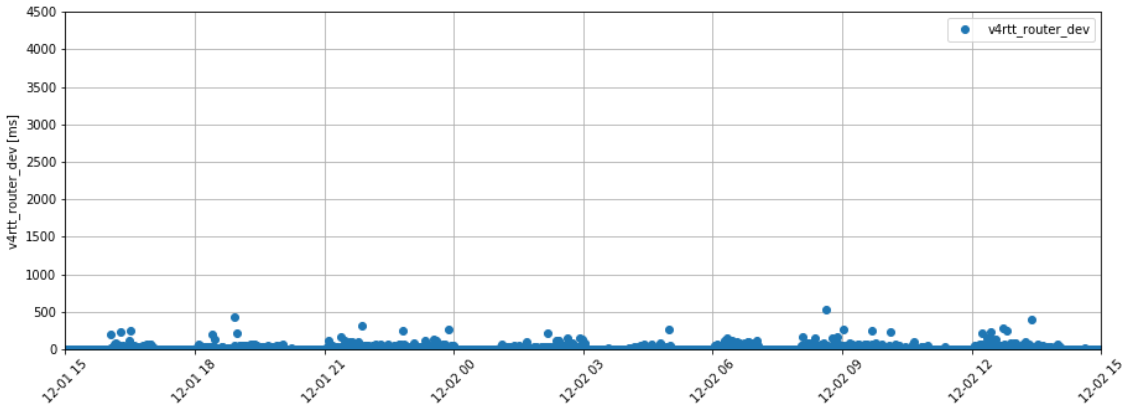

図 6.6: v4rtt_router_dev

6.3 LOF を用いた異常検知

本実験の SINDAN Probe の計測結果から LOF を用いることで異常検知できるか検証した．LOF は，互いに相関があるメトリックに対して，外れ値を検出するアルゴリズムである．

6.3.1 LOF を用いた異常検知の結果と評価

図 6.7 に，実験ネットワーク環境で計測した v4rtt_router_ave と v4rtt_router_dev の LOF の結果を示す．横軸は v4rtt_router_ave の結果，縦軸は v4rtt_router_dev の LOF の結果である．

今回，メトリック選定した v4rtt_router_ave と v4rtt_router_dev では，相関関係ではあるが，ネットワーク異常を検知できる結果が得られなかった．センサデバイスを用いたネットワーク状態計測の v4rtt_router_ave と v4rtt_router_dev の計測

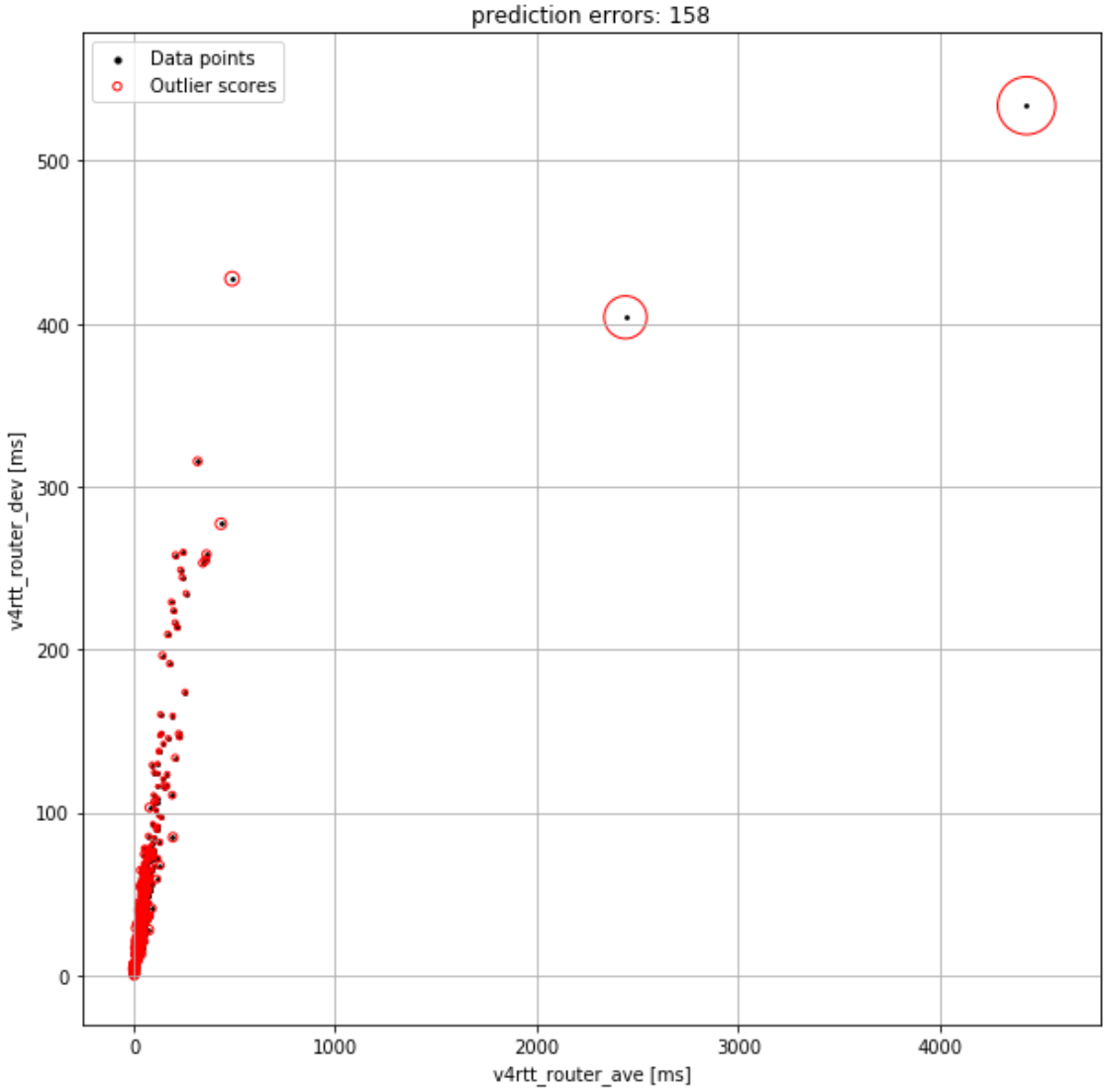

図 6.7: v4rtt_router_ave と v4rtt_router_dev の LOF

結果は，連続性が高く，LOF の異常検知に適する複数の島ができるような相関ではないと考えられる．

6.4 Change Finder を用いた異常検知

本実験の SINDAN Probe の計測結果から Change Finder を用いることで異常検知できるか検証した．Change Finder は，変化点検出するアルゴリズムであり，計測データから揺らぎとなる値を減らすために前処理を行い検証をした．

6.4.1 前処理

前処理は，フィルタによるデータの整形のことであり，アルゴリズムの精度を向上させたり，誤検知を減らすことができる．

単発に現れる高い値は，外れ値検出を行えば検知できる．変化点検知では，単発に現れる高い値はノイズとなるので移動平均で前処理とメディアンフィルタの前処理を比較した．

移動平均は，ウインドウサイズ分の過去の入力した値から平均を求めるフィルタであり，メディアンフィルタは，ウインドウサイズ分の過去の入力した値から中央値を求めるフィルタである．

移動平均とメディアンフィルタのウインドウサイズを，共に 5 サンプルにした．移動平均とメディアンフィルタで前処理を行い，Change Finder で変化点検知を行うと検知に 3 分遅れが生じるが，SNMP などのポーリングの間隔は 5 分で設定することが多く，パッシブ計測などのポーディングより早く検知ができる．SNMP などのポーリングの間隔は 5 分より短い間隔にすると，ネットワーク機器の CPU に高負荷を与える．

図 6.8 と 6.9 に，実験ネットワーク環境で計測した v4rtt_router_ave の結果を移動平均で前処理した値，メディアンフィルタで前処理した値を示す．縦軸は時刻，縦軸は v4rtt_router_ave の結果を移動平均で前処理した値，メディアンフィルタで前処理した値である．v4rtt_router_ave の前処理は，移動平均による前処理を行なっても，単発的に現れる高い値に影響を受ける．また，変化点の検知において，単発的な高い値が与える影響が支配的になるため，そのような値がウインドウに含まる場合には，変化点の前後における差が見つけにくくなる．しかし，メディアンフィルタを用いることで，単発的に現れる高い値の影響を受けずに，変化点付近の前後の変化の差も移動平均より緩やかにはならない．

図 6.10 と 6.11 に，実験ネットワーク環境で計測した v4rtt_router_dev を移動平均で前処理した値，メディアンフィルタで前処理した値を示す．縦軸は時刻，縦軸は v4rtt_router_dev の結果を移動平均で前処理した値，メディアンフィルタで前処理した値である．

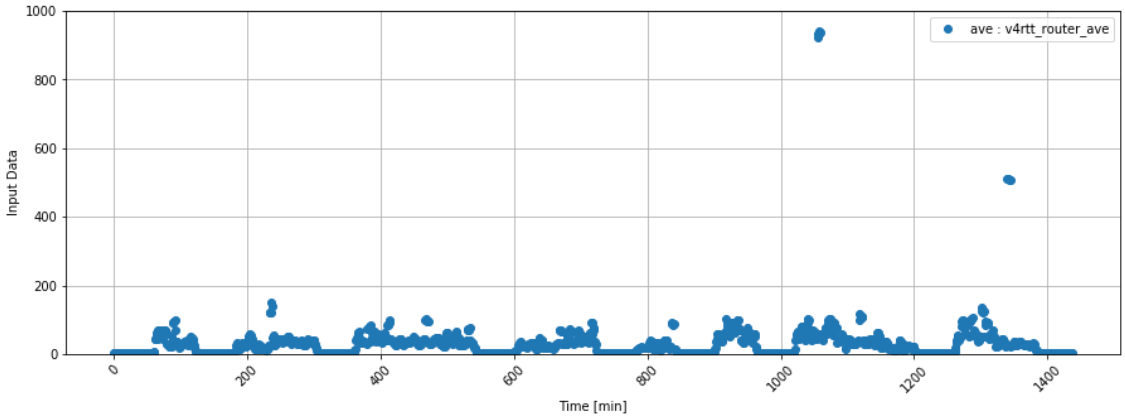

図 6.8: v4rtt_router_ave を移動平均で前処理した値

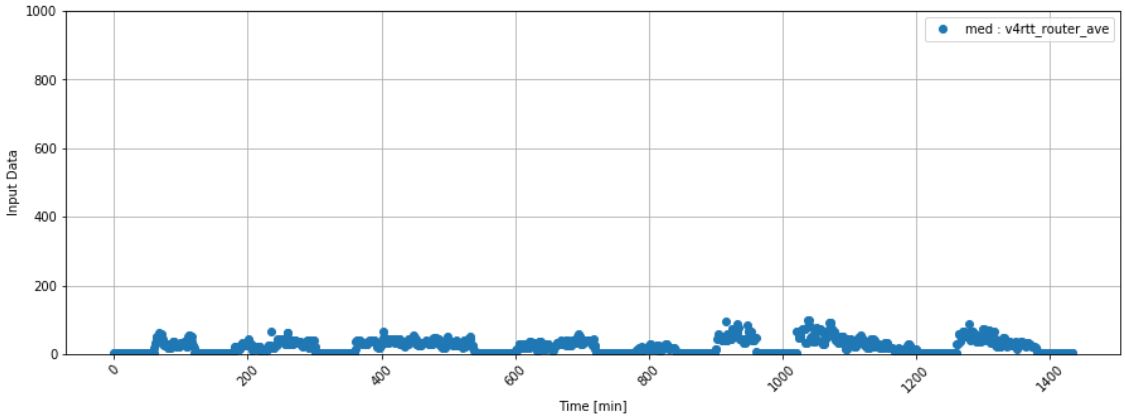

図 6.9: v4rtt_router_ave をメディアンフィルタで前処理した値

v4rtt_router_dev の前処理は，v4rtt_router_ave の前処理同様に，移動平均による前処理を行なっても，単発的に現れる高い値に影響を受ける．また，変化点付近では，前後の値がウインドウサイズに含まることにより，前後の変化の差が緩やかになる．しかし，メディアンフィルタを用いることで，単発的に現れる高い値の影響を受けずに，変化点付近の前後の変化の差も移動平均より緩やかにはならない．

センサデバイスを用いたネットワーク状態計測の v4rtt_router_ave と v4rtt_router_dev の計測結果は，計測結果のばらつきはガウス分布に従うよりも，インパルス的なノイズが多い．

以上の理由から，本実験の Change Finder に用いる前処理は，メディアンフィルタを採用した．

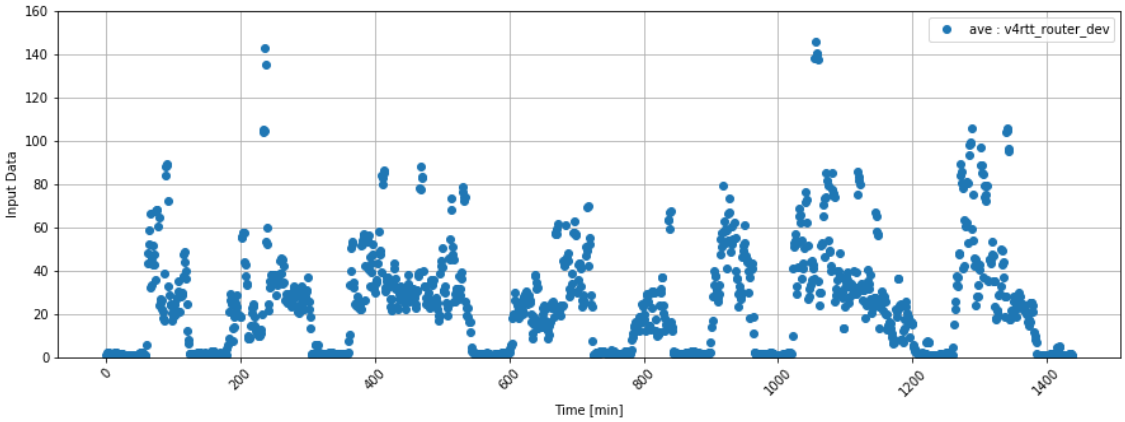

図 6.10: v4rtt_router_dev を移動平均で前処理した値

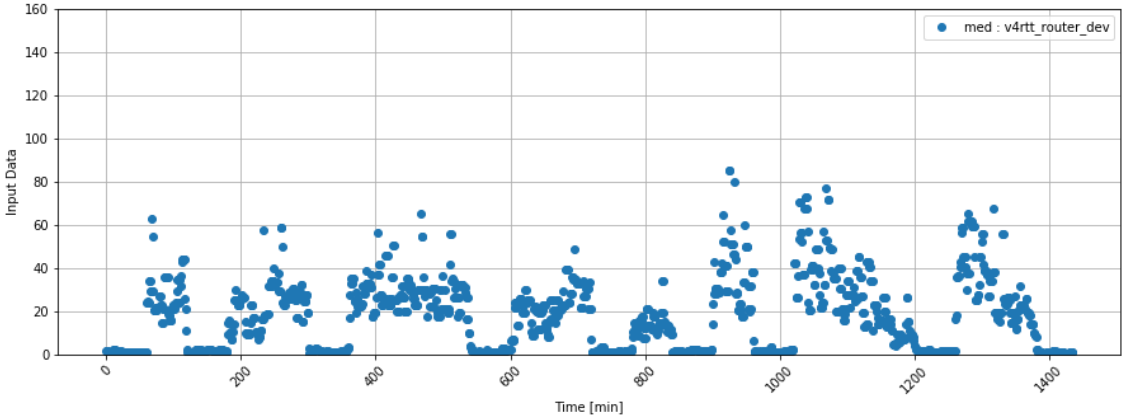

図 6.11: v4rtt_router_dev をメディアンフィルタで前処理した値

6.4.2 Change Finder を用いた異常検知の結果

無線 LAN 環境の負荷は，v4rtt_router_ave と v4rtt_router_dev から読み取ることができるので，メディアンフィルタによる前処理を行なった v4rtt_router_ave と v4rtt_router_dev を入力データとして，Change Finder を適用した．図 6.12 と 6.13 に，上に変化点を検知するために Change Finder の入力にメディアンフィルタで前処理を行なった v4rtt_router_ave の結果，下に Change Finder のスコアの結果と，メディアンフィルタで前処理を行なった v4rtt_router_dev の結果，下に Change Finder のスコアの結果を示す．横軸は時刻，上の縦軸は v4rtt_router_ave の結果をメディアンフィルタで前処理した値で前処理した値，下の縦軸は Change Finder のスコアの結果，上の縦軸は v4rtt_router_dev の結果をメディアンフィルタで前処理した値で前処理した値，下の縦軸は Change Finder のスコアの結果，また，iperf3 の負荷の段階を，I を IDLE は帯域に負荷をかけてない状態，3 を 3M はスループットを 3 Mbps の状態，5 を 5M はスループットを 5 Mbps の状態，M を MAX はスループットを指定せずに最大限の負荷をかけた最大の負荷状態を表した．

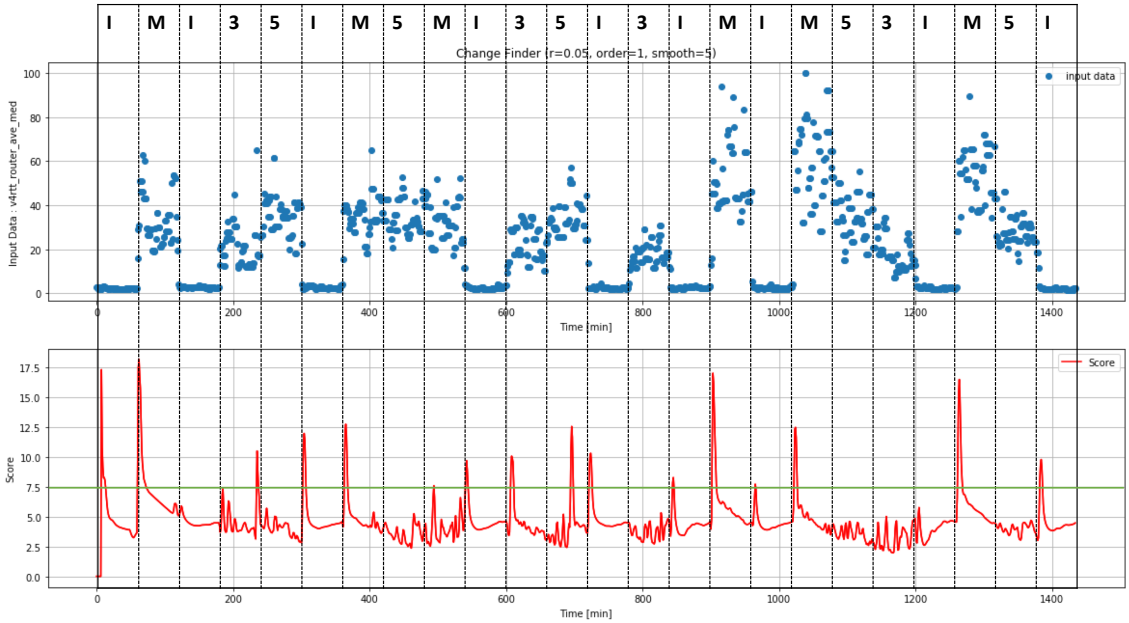

図 6.12: [上] メディアンフィルタで前処理を行なった v4rtt_router_ave[下]Change Finder のスコア

忘却パラメータ r は，値が小さいほど SDAR アルゴリズムにおいて過去のデータの影響を大きく受ける．変化が激しいネットワーク状態なら，低い値にする必要がある．AR モデルの次数 $Order$ は，次数を高めると計算コストが高くなるので，検出程度に問題がない範囲で低い値が望ましい．Smooth の T は，平滑化の範囲が長いと変化を捉えやすくなるが大きな変化には捉えにくくなる．本研究では，v4rtt_router_ave と v4rtt_router_dev の Change Finder のメトリックは，忘却パラメータ r は 0.05，AR モデルの次数 $Order$ は 1，外れ値計算スコアの移動平均平滑化する範囲である Smooth の T は 5 とした．

6.4.3 Change Finder を用いた異常検知の評価

Change Finder を用いた異常検知は，いくつか単発に現れる高い値を検出するのではなく，計測している無線 LAN 環境の負荷の変化を検知が目的である．したがって，Change Finder のスコアは，負荷がない IDLE 状態から高負荷の MAX の状態の変化点や高負荷の MAX の状態から負荷がない IDLE 状態の変化点のときは高くなることが期待される．また，状態が変化していない場合には，スコアは低くなっている必要がある．

v4rtt_router_ave と v4rtt_router_dev に対して Change Finder を適用した結果は，負荷がない IDLE 状態から高負荷の MAX の状態への変化点は検知はできるが，高負荷の MAX の状態から負荷がない IDLE 状態への変化点は検知できていない．ま

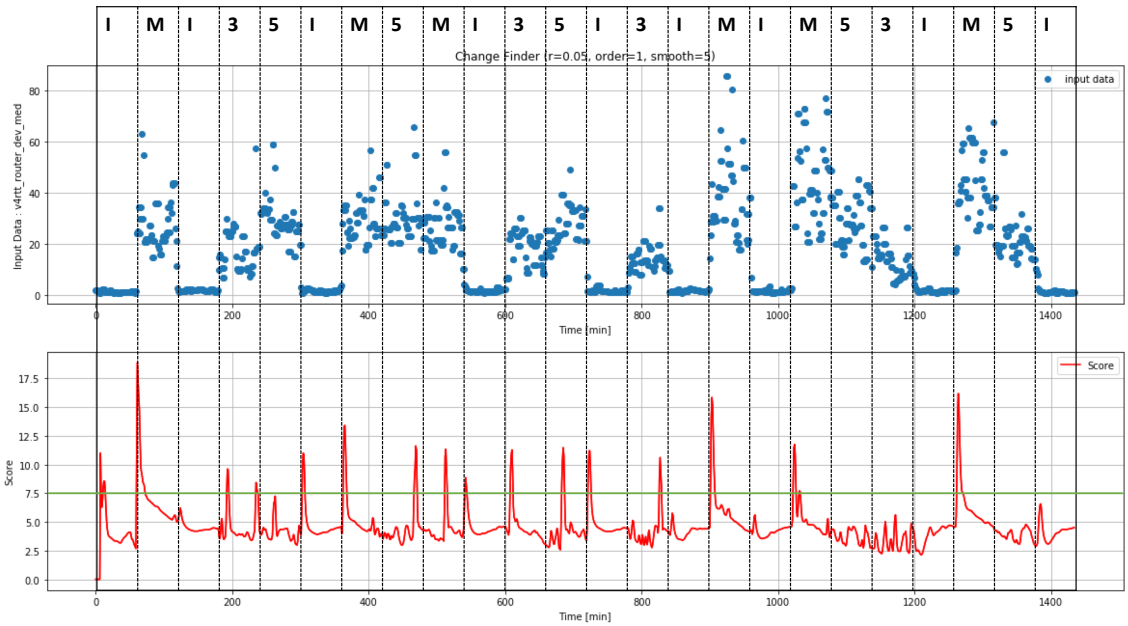

図 6.13: [上] メディアンフィルタで前処理を行なった v4rtt_router_dev[下]Change Finder のスコア

		Predictel	
		Positive	Negative
Actual	Positive	TP True Positive	FN False Negative
	Negative	FP False Positive	TN True Negative

表 6.3: 2 値の混合行列の内容

た，徐々に負荷変化する変化点は，あまり検知できていない．

Change Finder を用いた異常検知の結果を 2 x 2 混同行列を用いて評価した．混同行列 (Confusion matrix) とは，分類の性能を様々な観点から評価する際に，機械学習アルゴリズムで構成したモデルの評価する際に用いる指標である．図 6.3 に，2 値の混合行列を示す．Change Finder は，正常クラスと異常クラスの 2 クラスの分類により，2 値の混同行列で評価できる．

Predictel は予測の識別クラスであり，Predictel の Positive は予測の識別の正クラス，Predictel の Negative は予測の識別の負クラスである．Actual は正解の真値クラスであり，Actual の Positive は正解の真値の正クラス，Actual の Negative は正解の真値の負クラスである．TP (True Positive) は，真陽性であり，正解の真値の正かつ予測の識別の正であるクラスである．FN (False Negative) は，偽陰性であり，正解の真値の正かつ予測の識別の負であるクラスである．FP (False

Positive) は，偽陽性であり，正解の真値の負かつ予測の識別の正であるクラスである．TN (True Negative) は，陽陰性であり，正解の真値の負かつ予測の識別の負であるクラスである．TP と TN は，正しく Positive と Negative に分類できたことを示す．FP は，正解の真値の負であるのに関わらず，予測の識別の正であると誤検知したことを示し，FN は，正解の真値の正であるのに関わらず，予測の識別の負であると誤検知したことを示す．

正解率 (Accuracy) は，本来 Positive に分類すべきデータを Positive に分類し，本来 Negative に分類すべきデータを Nagative に分類した割合である．

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}$$

と表す．

精度 (Precision) は，予測の識別が Positive に分類されたデータのうち，本来 Positive であったデータの割合である．

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

と表す．

検出率 (Recall) は，真陽性率 (TPR : True Positive Rate) や感度 (Sensitivity) とも呼ばれ，本来 Positive に分類すべきデータを正しく Positive に分類できたデータの割合である．

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

と表す．

F 値 (F-measure) は，精度と検出率を持ち合わせた指標である．精度が高くても検出率が低いモデルでないか，検出率が高くても精度が低いモデルでないかを評価する．

$$\begin{aligned} \text{F} - \text{measure} &= \frac{2(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \\ &= \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \end{aligned}$$

と表す．

v4rtt_router_ave と v4rtt_router_dev の Change Finder の結果を F 値で評価した．表 6.4 に，v4rtt_router_ave の Change Finder を用いた異常検知の 2 値の混合行列を用いて評価を示し，表 6.5 に，v4rtt_router_dev の Change Finder を用いた異常検知の 2 値の混合行列を用いて評価を示す．各 v4rtt_router_ave と v4rtt_router_dev の Change Finder のスコアを，7.5 以上のときを異常とした．この値の時は，v4rtt_router_ave と v4rtt_router_dev ともに検知率が良く，誤検知も少ない．

初期の学習直後のスコアが高い箇所は，学習の揺らぎなので評価には除いた．

		Predictel	
		Positive	Negative
Actual	Positive	TP 13	FN 10
	Negative	FP 3	TN 1414

表 6.4: v4rtt_router_ave の Change Finder を用いた異常検知の混合行列

		Predictel	
		Positive	Negative
Actual	Positive	TP 9	FN 14
	Negative	FP 6	TN 1411

表 6.5: v4rtt_router_dev の Change Finder を用いた異常検知の混合行列

v4rtt_router_ave の Change Finder を用いた異常検知の F 値は，

$$\begin{aligned} \mathrm{F-measure} &= \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} \\ &= \frac{2\times 13}{2\times 13+3+10} \\ &= \frac{2}{3} \end{aligned}$$

である．

v4rtt_router_dev の Change Finder を用いた異常検知の F 値は，

$$\begin{aligned} \mathrm{F-measure} &= \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} \\ &= \frac{2\times 9}{2\times 9+6+14} \\ &= \frac{9}{19} \end{aligned}$$

である．

F 値から，v4rtt_router_dev の Change Finder を用いた異常検知より v4rtt_router_ave の Change Finder を用いた異常検知が精度が良いことがわかる．

第7章　考察

本章では，異常検知に用いたアルゴリズムの考察とセンサデバイスを用いたネットワーク異常検知の考察について述べる．

7.1　異常検知に用いたアルゴリズムの考察

異常検知に使用したLOFとChnage Finderについて考察を述べる．

本研究で行なった実験では，センサデバイスからの計測結果から相関による異常検知はできなかった．センサデバイスからの計測結果は，揺らぎが大きいので，相関する計測メトリックに対して，単発的にずれていれも，異常であるのか判断するのは困難であると考えられる．

7.1.1　Change Finderの考察

センサデバイスからのネットワーク計測の結果の v4rtt_router_ave と v4rtt_router_dev は，単調な増加や減少ではなく単発的な高い値をとるが，メディアンフィルタを用いて前処理を行うことで，Change Finderで変化点検知の精度の向上が期待できる．

センサデバイスからのネットワーク計測の結果の v4rtt_router_ave と v4rtt_router_dev から，無線 LAN 区間などのトラフィックの正確な値ではないが，トラフィックの推移が読み取ることができ，Change Finderでトラフィックの推移の変化を検知できる．

本研究の実験では，負荷に度合いを付けて実験し異常検知を行なったが，実際のネットワーク環境での異常検知の場合，ワームなどの狭いネットワークセグメントに拡散するようなマルウェアの検知に利用できる可能性がある．

ネットワーク機器は，Queueを用いてデータの転送を行なっている．センサデバイスを用いたネットワーク計測の結果では，転送量が急激に上がる場合は，Queueが詰まったり溢れなどが起きるので，変化の差が生じたのに対し，転送量が急激に下がる場合は，Queueが徐々に空きができるので，徐々に計測結果の値も改善したため，緩やかな変化になったと考えられる．

実ネットワーク環境では，センサデバイスを用いたネットワーク計測の結果から異常検知の際は，前処理やChange Finderアルゴリズムはネットワーク環境ご

とに，センサデバイスにおけるネットワーク計測の結果の特性が違うので，パラメータチューニングが必要である．また，Change Finder は，非定常性なので，初期値によって変動するので，学習時に注意が必要である．

7.2 センサデバイスを用いたネットワーク異常検知の考察

SINDAN におけるネットワーク状態計測手法では，ネットワーク管理者は，センサデバイスのネットワーク状態計測の結果から各階層が成功か失敗しか把握しにくく，詳細なネットワーク状態を把握するためには，各階層の計測項目を精査しなければいけない．

本研究で提案したセンサデバイスを用いた異常検知の設計では，センサデバイスのネットワーク状態計測の結果からアルゴリズム用いて時系列分析をすることで，各階層の計測項目を精査しなくても，ネットワーク異常があればネットワーク管理者へ報告できる設計にした．

様々なネットワーク異常に対応するには，様々な異常検知するアルゴリズムを利用しなければならなく，検知してアラートを送る際は，ネットワーク障害につながる異常であるのか，異常であるが早急に対象が必要なのかなどのアラートに対してトリアージが必要になる．

SNMP などの計測は，SNMP のポーリングの粒度を細かく設定すると，ネットワーク機器に高負荷を掛ける．また，NetFlow や sFlow などの計測は，導入コストが大きく，小さなネットワークセグメントまでモニタリングするのは難しい．小さなネットワークセグメントにセンサデバイスを設置し，前処理を行い Change Finder からネットワークの異常検知の可能性を示した．

第8章　おわりに

本章では，まとめと実ネットワーク環境における異常検知についての展望を述べる．

8.1　まとめ

本研究は，発見が困難なネットワーク障害とネットワーク計測について整理を行い，センサデバイスからの計測をもとに発見が困難な障害であるローカルネットワークのAP間に負荷をかけて，異常を検知できるか検証をした．センサデバイスでは，ネットワーク状態計測を行い，ネットワーク状態計測の計測結果から異常検知できるメトリックを選定し，Change Finderを用いて異常検知を行なった．負荷がないIDLE状態から高負荷のMAXの状態の変化点は検知はできるが，高負荷のMAXの状態から負荷がないIDLE状態の変化点は検知できていない．また，徐々に負荷変化する変化点は，あまり検知できていない結果であった．

8.2　今後の展望

センサデバイスのネットワーク状態計測から異常検知は，様々なネットワーク障害に対応できる．また，センサデバイスのネットワーク状態計測から異常検知の結果を，ネットワーク管理者に報告するためには，異常検知の結果の精度が問われる．そして，センサデバイスのネットワーク状態計測の計測からネットワーク異常検知を示すメトリックを作成することで，検知の精度を向上できる．

謝辞

本研究を進めるにあたり，本学情報社会基盤研究センターの篠田陽一教授には指導教員として終始ご指導を頂き，深く感謝致します．また，本研究室の知念賢一特任准教授，宇田仁助教には，研究に関する活発な議論を頂き，深く感謝致します．

WIDE ワーキンググループの SINDAN Project の東京工業大学 学術国際情報センター 北口善明准教授，東京大学大学院 総合文化研究科 石原知洋助教，フリーランス 高嶋健人氏，北陸先端科学技術大学院大学と株式会社レピダム 阿部博氏には，研究方針や実験に関するご指導頂き，深く感謝致します．

情報通信医学研究所 中川晋一氏には，研究に関する助言を頂き，深く感謝致します．また，金沢大学 大野浩之教授には，NT 金沢 2018 にて無線計測について貴重な機会と指導を頂き，深く感謝致します．

本研究の実験するのにあたり，様々なネットワーク環境を提供してくださった，第 17 回 日本ボーイスカウトジャンボリーの皆様，並びに会場でネットワーク計測する機会を提供してくださった CISCO 様と JAIST 様に深く感謝致します．

WIDE Project 2018 年 9 月 合宿にて，ネットワークの構築とネットワーク計測する機会をくださり，深く感謝致します．

本研究室の博士前期課程の橋本光世氏，阿波史和氏，砂川真範氏，広瀬太志氏，三島航氏，宮崎駿氏，山口礼央氏，小松源氏，菅野洋信氏，北沢尭宏氏，廣中颯氏，渡邊司揮氏には活発な議論や研究生活を送る上で様々なご助力を頂き，心から感謝致します．

最後に，生活を支えてくださった家族に深く感謝します．

参考文献

[1] 総務省. 情報通信機器の普及状況 第 2 部 基本データと政策動向 29 年度版. http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/html/nc262110.html, 2017.

[2] 総務省. 総務省 | ict 利活用の促進. http://www.soumu.go.jp/menu_seisaku/ictseisaku/ictriyou/index.html.

[3] 総務省. 将来のネットワークインフラに関する研究会 報告書 平成 29 年 7 月. http://www.soumu.go.jp/main_content/000496762.pdf.

[4] サイバー攻撃の観測情報を web で公開 - nicter が収集した情報の利活用を促進 -. https://www.nict.go.jp/publication/NICT-News/1205/06.html.

[5] Amazon web services offers reliable, scalable, and inexpensive cloud computing services. free to join, pay only for what you use. https://aws.amazon.com/.

[6] APNIC. 6th anniversary of world ipv6 launch 〜 日本の ipv6 普及状況 〜. https://blog.nic.ad.jp/blog/june6_ipv6.

[7] Ietf — internet engineering task force. https://www.ietf.org/.

[8] Ieee 802.3 ethernet. http://www.ieee802.org/3/.

[9] Ieee 802.11, the working group setting the standards for wireless lans. http://www.ieee802.org/11/.

[10] J. Postel. Internet Protocol, RFC791. September 1981.

[11] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification, RFC2460. December 1998.

[12] J. Postel. Internet Control Message Protocol, RFC792. September 1981.

[13] A. Conta, S. Deering, M. Gupta, and Ed. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, RFC4443. March 2006.

[14] J. Postel. Transmission Control Protocol, RFC793. September 1981.

[15] J. Postel. User Datagram Protocol, RFC768. August 1980.

[16] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1, RFC2616. June 1999.

[17] E. Rescorla. HTTP Over TLS, RFC2818. May 2000.

[18] P.V. Mockapetris. Domain names - concepts and facilities, RFC1034. November 1987.

[19] R. Droms. Dynamic Host Configuration Protocol, RFC2131. March 1997.

[20] R. Droms, Ed., J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney. Dynamic Host Configuration Protocol for IPv6 (DHCPv6), RFC3315. July 2003.

[21] D. Wing and A. Yourtchenko. Happy Eyeballs: Success with Dual-Stack Hosts, RFC6555. April 2012.

[22] D. Schinazi and T. Pauly. Happy Eyeballs Version 2: Better Connectivity Using Concurrency, RFC8305. December 2017.

[23] G. Malkin. Traceroute Using an IP Option, RFC1393. January 1993.

[24] A. Bryan, N. McNab, T. Tsujikawa, P. Poeml, and H. Nordstrom. Metalink/HTTP: Mirrors and Hashes, RFC6249. June 2011.

[25] iperf - iperf3 and iperf2 user documentation. `https://iperf.fr/iperf-doc.php`.

[26] R. Frye, D. Levi, S. Routhier, and B. Wijnen. Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework, RFC3584. August 2003.

[27] Cisco. Network telemetry. `https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Security/Baseline_Security/securebasebook/sec_chap5.pdf`.

[28] R. Gerhards. The Syslog Protocol, RFC5424. March 2009.

[29] 長健二郎. インターネットと計測技術 (第 2 回). `https://www.iijlab.net/~kjc/papers/iijnews-200611.pdf`.

[30] mtr - a network diagnostic tool - linux man pages (8). https://www.systutorials.com/docs/linux/man/8-mtr/.

[31] Cisco. Cisco ios netflow. https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html.

[32] P. Phaal, S. Panchen, and N. McKee. InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks, RFC3176. September 2001.

[33] Juniper. Juniper flow monitoring. https://www.juniper.net/us/en/local/pdf/app-notes/3500204-en.pdf.

[34] B. Claise and Ed. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information, RFC5101. January 2008.

[35] B. Trammell, E. Boschi, L. Mark, T. Zseby, and A. Wagner. Specification of the IP Flow Information Export (IPFIX) File Format, RFC5655. October 2009.

[36] B. Claise, Ed., B. Trammell, Ed., and P. Aitken. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, RFC7011. September 2013.

[37] B. Claise, A. Kobayashi, and B. Trammell. Operation of the IP Flow Information Export (IPFIX) Protocol on IPFIX Mediators, RFC7119. February 2014.

[38] Shelly Cadora. Ten lessons from telemetry. https://www.nanog.org/sites/default/files/Cadora_Ten_Lessons.pdf.

[39] Cacti® - the complete rrdtool-based graphing solution. https://www.cacti.net.

[40] Nagios - the industry standard in it infrastructure monitoring. https://www.nagios.org/.

[41] Zabbix :: The enterprise-class open source network monitoring solution. https://www.zabbix.com/.

[42] Prometheus - monitoring system & time series database. https://prometheus.io/.

[43] Network intelligence software — thousandeyes. https://www.thousandeyes.com/.

[44] digitalocean/netbox: Ip address management (ipam) and data center infrastructure management (dcim) tool. https://github.com/digitalocean/netbox.

[45] netdata/netdata: Real-time performance monitoring, done right! https://my-netdata.io/. https://github.com/netdata/netdata.

[46] Mawi working group traffic archive. http://mawi.wide.ad.jp/mawi/.

[47] 長健二郎. トラフィック計測のための多次元フロー集約アルゴリズム. https://www.iij.ad.jp/dev/tech/techweek/pdf/171110_02.pdf.

[48] 異常検知と変化検知. 機械学習プロフェッショナルシリーズ. 講談社, 2015.

[49] Valentina Moskvina and Anatoly Zhigljavsky. An algorithm based on singular spectrum analysis for change-point detection. *Communications in Statistics - Simulation and Computation*, Vol. 32, No. 2, pp. 319–352, 2003.

[50] T. IDE. Knowledge discovery from heterogeneous dynamic systems using change-point correlations. *Proc. 2005 SIAM Int. Conf. on Data Mining (SDM 05), Newport Beach, CA, USA, April 21-23*, pp. 571–576, 2005.

[51] Dllab 異常検知ナイト 資料 2018021. https://www.slideshare.net/KosukeNakago/dllab-20180214-88470902.

[52] Ripe atlas - ripe network coordination centre. https://atlas.ripe.net/.

[53] What is ripe atlas? - ripe atlas - ripe network coordination centre. https://atlas.ripe.net/about/.

[54] イクシアでネットワークをより強力に. https://www.ixiacom.com/ja.

[55] Xrpi active monitoring probe — ixia. https://www.ixiacom.com/products/xrpi-active-monitoring-probe.

[56] Romain Fontugne, Emile Aben, Cristel Pelsser, and Randy Bush. Pinpointing delay and forwarding anomalies using large-scale traceroute measurements. *CoRR*, Vol. abs/1605.04784, , 2016.

[57] 健介福田. インターネットバックボーントラフィックにおける異常検出. コンピュータソフトウェア, Vol. 30, No. 2, pp. 23–32, apr 2013.

[58] 村井秀聡, 砂田英之, 牧和宏. Snmp を利用したバーストトラフィック検知方式の検知精度評価の報告. 研究報告インターネットと運用技術（IOT）, Vol. 2015, No. 28, pp. 1–6, feb 2015.

[59] 鈴木彦文, 浅川圭史, 永井一弥, 林裕平, 工藤伊知郎. Netflow トラフィックデータの取得と解析に関する共同研究. 学術情報処理研究, Vol. 22, No. 1, pp. 3–11, 2018.

[60] 阿部博, 敷田幹文, 篠田陽一. イベントネットワークにおける syslog を用いた異常検知手法の提案と実データを用いた評価. 情報処理学会論文誌, Vol. 59, No. 3, pp. 1006–1015, mar 2018.

[61] 石川直樹, 大石恭弘, 中山奨, 前田香織. 大規模 wi-fi ネットワークにおけるパッシブ通信品質推定手法の実験的評価. 電子情報通信学会技術研究報告. IA, インターネットアーキテクチャIA2018(17), pp. 9–14, 2018.

[62] Waiting Fok, Xiapu Luo, R Mok, Weichao Li, Yujing Liu, Edmond Chan, and Rocky Chang. Monoscope: Automating network faults diagnosis based on active measurements. pp. 210–217, 01 2013.

[63] Xingfeng Guo, Dianhong Wang, and Fenxiong Chen. An anomaly detection based on data fusion algorithm in wireless sensor networks. *Int. J. Distrib. Sen. Netw.*, Vol. 2015, pp. 78:78–78:78, January 2015.

[64] 山村翔, 熊谷充敏, 神谷和憲, 倉上弘. 変化点検知を用いた新種スキャンの早期発見手法の検討. コンピュータセキュリティシンポジウム 2017 論文集, 第 2017 巻, oct 2017.

[65] Framework, models and controlled experiments for network troubleshooting. *Comput. Netw.*, Vol. 107, No. P1, pp. 36–54, October 2016.

[66] B. Trammell, P. Casas, D. Rossi, A. Br, Z. B. Houidi, I. Leontiadis, T. Szemethy, and M. Mellia. mplane: an intelligent measurement plane for the internet. *IEEE Communications Magazine*, Vol. 52, No. 5, pp. 148–156, May 2014.

[67] Alessandro Finamore, Marco Mellia, Michela Meo, Maurizio M. Munafò, P. D. Torino, and Dario Rossi. Experiences of internet traffic monitoring with tstat. *IEEE Network*, Vol. 25, , 2011.

[68] Enrico Bocchi, Ali Safari Khatouni, Stefano Traverso, Alessandro Finamore, Maurizio Munaf, Marco Mellia, and Dario Rossi. Statistical network monitoring: Methodology and application to carrier-grade nat. *Computer Networks*,

Vol. 107, pp. 20 – 35, 2016. Machine learning, data mining and Big Data frameworks for network monitoring and troubleshooting.

[69] Arian Baer, Pedro Casas, Alessandro DAlconzo, Pierdomenico Fiadino, Lukasz Golab, Marco Mellia, and Erich Schikuta. Dbstream: A holistic approach to large-scale network traffic monitoring and analysis. *Computer Networks*, Vol. 107, pp. 5 – 19, 2016. Machine learning, data mining and Big Data frameworks for network monitoring and troubleshooting.

[70] T. Kimura, K. Ishibashi, T. Mori, H. Sawada, T. Toyono, K. Nishimatsu, A. Watanabe, A. Shimoda, and K. Shiomoto. Spatio-temporal factorization of log data for understanding network events. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 610–618, April 2014.

[71] Network-ai 技術の活用による新たな価値の創出. url-http://www.ntt.co.jp/journal/1803/files/JN20180313.pdf.

[72] 花森利弥, 西村利浩. システムの異常予兆を検知するリアルタイム監視ソリューション (特集ヒューマンセントリック iot), mar 2016.

[73] 機械学習を活用して周期性の異常も捉える「ネットワークの可視化・異常検知ソリューション」. `https://www.alaxala.com/jp/solution/security/kashika/kashika/index.html`.

[74] Sindan project simple integrated network diagnosis and notification. `https://www.sindan-net.com`.

[75] 北口善明, 石原知洋, 高嶋健人. センサデバイスを利用したネットワーク状態計測手法の評価. 情報処理学会 マルチメディア・分散・協調とモバイル (DICOMO) シンポジウム 2017 論文集.

[76] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6), RFC2461. December 1998.

[77] E. Nechamkin and J-F. Mule. Multimedia Terminal Adapter (MTA) Management Information Base for PacketCable- and IPCablecom-Compliant Devices, RFC4682. December 2006.

[78] Google public dns. `https://developers.google.com/speed/public-dns/`.

[79] 1.1.1.1 — the internet's fastest, privacy-first dns resolver. `https://1.1.1.1/`.

[80] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, Vol. 29, No. 2, pp. 93–104, May 2000.

[81] 入門機械学習による異常検知: R による実践ガイド. コロナ社, 2015.

[82] Kenji Yamanishi and Jun ichi Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. *KDD '02 Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002.

[83] 山西健司著. データマイニングによる異常検知 Anomaly Detection with Data Mining. 2009.

[84] Valery Guralnik and Jaideep Srivastava. Event detection from time series data. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pp. 33–42, New York, NY, USA, 1999. ACM.

[85] Kea dhcp server — internet systems consortium.

[86] Bind 9 open source dns server — internet systems consortium.

[87] The raspberry pi is a tiny and affordable computer that you can use to learn programming through fun, practical projects. join the global raspberry pi community. `https://www.raspberrypi.org/`.

[88] 11ac,gw-450d2,wi-fi, 無線, 無線 lan, 子機,5ghz,wps, 接続, 設定,pci, プラネックス. `http://www.planex.co.jp/products/gw-450d2/`.

[89] Simple integrated network diagnosis and notification - sindan project. `https://github.com/SINDAN`.