



UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Performance Evaluation of Network Anomaly Detection Systems

Gilberto Fernandes Junior

Tese para obtenção do Grau de Doutor em
Engenharia Informática
(3º ciclo de estudos)

Orientador: Prof. Doutor Joel José Puga Coelho Rodrigues

Covilhã, Abril de 2019

Dedication

To God.

*My child, if you aspire to serve the Lord,
prepare yourself for an ordeal.
Be sincere of heart, be steadfast,
and do not be alarmed when disaster comes.
Cling to him and do not leave him,
so that you may be honoured at the end of your days.
Whatever happens to you, accept it,
and in the uncertainties of your humble state, be patient,
since gold is tested in the fire,
and the chosen in the furnace of humiliation.
Trust him and he will uphold you,
follow a straight path and hope in him.
Ecclesiasticus 1:6*

Acknowledgments

The accomplishment of this doctoral thesis counted on essential support and incentive of the individuals and organizations mentioned here, without which it would not have become a reality and to which I will be eternally grateful.

To my supervisor and kind friend Professor Joel José Puga Coelho Rodrigues, for accepting me in his laboratory and research group NetGNA, giving all the support and advices to make me feel at home since the first contact. Also thanks to his orientation, total support, availability, for the transmitted values, for the opinions and criticism, full collaboration in solving doubts and problems that have arisen throughout the accomplishment of this research and for all incentive words. Without his effort, encouragement, and understanding especially in the most challenging moments, none of this would have been possible.

To my friend Professor Luiz Fernando Carvalho, for his friendship and support while working together and whenever I needed his help during this Thesis development.

I would also like to thank all my colleagues of the Next Generation Networks and Applications (NetGNA) research group, especially to Professor Mario Moreira, Ms. Germanno Teles, Professor Simone Ferreira, Professor José Victor Sobral and Professor Bruno Silva for all support, advices, guidance, and friendship. They have been my family.

For all support that was given to me, I am most grateful to the University of Beira Interior, and the Instituto de Telecomunicações, Covilhã delegation.

To the Brazilian National Council for Scientific and Technological Development (CNPq) and the Science without Borders program, through the grant contract number 249794/2013-6, for all the support and trust that was given to me.

Also, heartfelt thanks to my wife Amanda França for her constant support, love, and understanding. She has supported me both mentally and emotionally during this doctorate.

To my sisters Bianca Fernandes and Giovana Fernandes for all their support and kindness during the most complicated periods I had in which I needed their help and they never hesitated.

My last and most profound gratitude goes to my parents Gilberto Fernandes and Tania Silene Alves Queiroz Fernandes for their constant love and support. If I am the man I am now, it is because of them. No words are possible to describe all the gratitude I have deep in my heart. I owe to them not just this thesis, but everything.

Foreword

This thesis describes the research work performed in the scope of the 4-year doctoral research programme and presents its main contributions and achievements. This doctoral programme and inherent research activities were carried out at the Next Generation Networks and Applications Group (NetGNA) research group of the Departamento de Informática, Universidade da Beira Interior, Covilhã, Portugal and Instituto de Telecomunicações, Delegação da Covilhã, Portugal. The research work was supervised by Prof. Dr. Joel José Puga Coelho Rodrigues, and financially supported by the National Council for Scientific and Technological Development (CNPq) through the grant contract 249794/2013-6.

List of Publications

Papers resulting from this doctoral research programme

1. A Comprehensive Survey on Anomaly Detection

Gilberto Fernandes Jr., Joel J. P. C. Rodrigues, Luiz Fernando Carvalho, Jalal Al-Muhtadi, and Mario L. Proença Jr.

Telecommunication Systems, Springer US, pp 1-43, 2018, ISSN 1018-4864 (Print) 1572-9451 (Online)

DOI: <https://doi.org/10.1007/s11235-018-0475-8>

2. Autonomous profile-based anomaly detection system using principal component analysis and flow analysis

Gilberto Fernandes Jr., Joel J. P. C. Rodrigues, and Mario L. Proença Jr.

Applied Soft Computing, Elsevier Science BV, Vol. 34, pp. 513-525, 2015, ISSN 1568-4946

DOI: <https://doi.org/10.1016/j.asoc.2015.05.019>

3. Network Anomaly Detection using IP Flows with Principal Component Analysis and Ant Colony Optimization

Gilberto Fernandes Jr., Luiz Fernando Carvalho, Joel J. P. C. Rodrigues, and Mario L. Proença Jr.

Journal of Network and Computer Applications, Elsevier Science BV, Vol. 64, pp 1-11, 2016, ISSN 1084-8045

DOI: <https://doi.org/10.1016/j.jnca.2015.11.024>

4. Statistical, Forecasting and Metaheuristic Techniques For Network Anomaly Detection

Gilberto Fernandes Jr., Eduardo M. H. Pena, Luiz Fernando Carvalho, Joel J. P. C. Rodrigues, and Mario L. Proença Jr.

The 30th ACM/SIGAPP Symposium on Applied Computing (SAC 2015), Salamanca, SPAIN, April, 2015, pp701-707.

DOI: <https://doi.org/10.1145/2695664.2695852>

5. A novel anomaly detection system to assist network management in SDN environment

Luiz Fernando Carvalho, Gilberto Fernandes Jr., Joel J. P. C. Rodrigues, Leonardo S. Mendes, and Mario L. Proença Jr.

2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-6.

DOI: doi.org/10.1109/ICC.2017.7997214

6. Digital signature to help network management using flow analysis: Network Management Using Flow Analysis

Mario L. Proença Jr., Gilberto Fernandes Jr., Luiz Fernando Carvalho, Marcos V. O. de Assis, and Joel J. P. C. Rodrigues

International Journal of Network Management, Wiley, Vol. 26, Issue 2, pp. 76-94, 2015.

DOI: <https://doi.org/10.1002/nem.1892>

7. Digital signature of network segment for healthcare environments support

Luiz Fernando Carvalho, Gilberto Fernandes Jr., Marcos V. O. de Assis, Joel J. P. C. Rodrigues, and Mario L. Proença Jr.

Innovation and Research in BioMedical engineering (IRBM), Elsevier Science BV, Vol. 35, Issue 6, pp. 299-309, December 2014

DOI: <https://doi.org/10.1016/j.irbm.2014.09.001>

Resumo

Atualmente, existe uma enorme e crescente preocupação com segurança em tecnologia da informação e comunicação (TIC) entre a comunidade científica. Isto porque qualquer ataque ou anomalia na rede pode afetar a qualidade, interoperabilidade, disponibilidade, e integridade em muitos domínios, como segurança nacional, armazenamento de dados privados, bem-estar social, questões econômicas, e assim por diante. Portanto, a detecção de anomalias é uma ampla área de pesquisa, e muitas técnicas e abordagens diferentes para esse propósito surgiram ao longo dos anos.

Ataques, problemas e falhas internas quando não detetados precocemente podem prejudicar gravemente todo um sistema de rede. Assim, esta Tese apresenta um sistema autônomo de detecção de anomalias baseado em perfil utilizando o método estatístico Análise de Componentes Principais (PCADS-AD). Essa abordagem cria um perfil de rede chamado Assinatura Digital do Segmento de Rede usando Análise de Fluxos (DSNSF) que denota o comportamento normal previsto de uma atividade de tráfego de rede por meio da análise de dados históricos. Essa assinatura digital é utilizada como um limiar para detecção de anomalia de volume e identificar disparidades na tendência de tráfego normal. O sistema proposto utiliza sete atributos de fluxo de tráfego: bits, pacotes e número de fluxos para detetar problemas, além de endereços IP e portas de origem e destino para fornecer ao administrador de rede as informações necessárias para resolvê-los.

Por meio da utilização de métricas de avaliação, do crescimento de uma abordagem de detecção distinta da proposta principal e comparações com outros métodos realizados nesta tese usando dados reais de tráfego de rede, os resultados mostraram boas previsões de tráfego pelo DSNSF e resultados encorajadores quanto a geração de alarmes falsos e precisão de detecção.

Com os resultados observados nesta tese, este trabalho de doutoramento busca contribuir para o avanço do estado da arte em métodos e estratégias de detecção de anomalias, visando superar alguns desafios que emergem do constante crescimento em complexidade, velocidade e tamanho das redes de grande porte da atualidade, proporcionando também alta performance. Ainda, a baixa complexidade e agilidade do sistema proposto contribuem para que possa ser aplicado a detecção em tempo real.

Palavras-chave

detecção de Anomalias, Sistemas de detecção de Intrusão, Caracterização de Tráfego, Segurança de Redes, Gestão de Redes, Análise de Componentes Principais, Redes de Computadores, Assinatura Digital do Segmento de Rede usando Análise de Fluxos.

Extended Abstract in Portuguese

Introdução

Esta seção resume, de forma alargada, os 4 anos de trabalho de investigação no âmbito da tese de doutoramento intitulada “Performance Evaluation of Anomaly Detection Systems”. Esta tese foca-se no estudo e na proposta de estratégias e metodologias de análise de dados de tráfego para o monitoramento e deteção de anomalias de rede. A primeira etapa descreve a estrutura da tese, bem como define o problema abordado e os principais objetivos do estudo. As principais contribuições deste trabalho para o avanço do estado da arte também são apresentadas.

Enquadramento

Atualmente, a comunidade científica tem uma preocupação constante com segurança de alta eficiência e qualidade de serviço em redes de larga escala. A expansão de novas tecnologias e serviços de comunicação, juntamente com um número crescente de dispositivos de rede interconectados, usuários da Web, serviços e aplicativos, contribui para tornar as redes de computadores cada vez maiores e mais complexas como sistemas. Além disso, há o chamado paradigma de comunicação ilimitada, para redes de próxima geração, que prevê oferecer comunicações a qualquer hora, em qualquer lugar e de qualquer forma aos seus usuários e requer a integração completa e interoperabilidade de tecnologias emergentes [1]. Estes problemas tornam ainda mais complexo e desafiador manter uma gestão de redes precisa, além de conduzir a sérias vulnerabilidades de rede, pois incidentes de segurança podem ocorrer com mais frequência [2, 3].

Estas instâncias de segurança podem ser causadas por indivíduos externos à rede, como ataques maliciosos com o objetivo de desligar serviços ou roubar informações particulares, ou por fatores internos (problemas operacionais), como erros de configuração, falhas de servidor, falta de energia, congestionamento ou grandes transferências de arquivos não mal-intencionadas [4]. Independentemente da origem, estas ameaças, comumente chamadas de anomalias, podem ter um impacto significativo no serviço de rede e nos usuários finais e prejudicar as operações e a disponibilidade das redes de computadores.

O termo anomalia tem várias definições dentro da literatura. Barnett e Lewis definem uma anomalia no conjunto de dados como “observação (ou um subconjunto de observações) que parece ser inconsistente com o restante daquele conjunto de dados” [5]. Chandola *et al.* expressa este termo como “padrões em dados que não estão em conformidade com uma noção bem definida de comportamento normal” [6]. De acordo com Lakhina *et al.*, “Anomalias são mudanças incomuns e significativas nos níveis de tráfego de uma rede, que muitas vezes podem abranger vários links” [7]. Hoque *et al.* define como “padrões interessantes não-conformes comparados à uma noção bem definida de comportamento normal” [8]. Por estas definições, é claro que o conceito de normalidade é um dos principais passos para o desenvolvimento de uma solução para detetar anomalias de rede.

Embora aparentemente desprezioso, o problema de definir uma região denotando comportamento normal e distinguindo como uma anomalia qualquer ocasião contrastando esse padrão normal, é desafiador. Diagnóstico mais rápido, menor complexidade e correções adequadas das causas são os principais objetivos do campo. Todos os fatores são vitais para desenvolver uma abordagem de detecção de anomalias mais eficiente. Os fatores de precisão e velocidade, juntamente com a identificação correta de tais eventos anormais em tempo hábil, são essenciais para reduzir a degradação significativa do serviço, danos maliciosos e custos computacionais. Por esta razão, a comunidade de pesquisa vem desenvolvendo muitos modelos, algoritmos e mecanismos ao longo dos anos, para desenvolver melhores soluções e abordagens para garantir a robustez de sistemas de rede cada vez maiores e complexos.

Na literatura, os métodos de detecção de anomalias podem ser classificados de duas formas: baseado em assinatura e baseado em perfil. Os sistemas baseados em assinaturas usam um conhecimento prévio sobre as características de cada tipo de anomalia para identificar possíveis incidentes já conhecidos anteriormente. Além disso, as abordagens baseadas em perfil criam um perfil de rede que representa o comportamento normal do tráfego, e as anomalias de tráfego são detetadas a partir de desvios em relação a esse perfil [9, 10]. Embora os métodos baseados em assinatura tenham sido amplamente investigados na literatura, eles têm uma clara desvantagem. É pré-requisito que as assinaturas de anomalia sejam conhecidas antecipadamente, dificultando o reconhecimento de novas anomalias. Além disso, os métodos baseados em assinatura podem ser evitados por fontes mal-intencionadas falsificando assinaturas de anomalias. Por outro lado, um sistema baseado em perfis cria um perfil de comportamento normal da atividade de rede, eliminando a necessidade de conhecimento prévio sobre a natureza e as propriedades das anomalias. Essa característica leva a algumas vantagens: a possibilidade de descobrir novos tipos de anomalias; a detecção de ataques internos; e também torna difícil para um atacante saber com convicção que ação maliciosa ele pode realizar sem ser detetado pelo sistema [9, 11]. Assim, a proposta desta tese é criar um sistema autônomo de monitoramento de rede baseado em perfil capaz de identificar o comportamento normal da rede, adotando um método eficiente de caracterização do tráfego para criar um perfil de comportamento normal do tráfego para identificar possíveis anomalias no tráfego.

Definição do Problema

Realizar uma análise e monitoramento de tráfego completos em sistemas de rede de larga escala é uma tarefa quase impossível de ser executada manualmente por um administrador de rede. As altas velocidades de conexão combinadas com o grande e crescente número de links e segmentos tornam essa tarefa ainda mais complexa [12]. Além disso, existe a necessidade de agilidade na detecção e prevenção de problemas, pois o administrador da rede deve trabalhar de forma proativa para evitar interrupções na operação da rede, já que uma das premissas de governança da TIC é que os serviços de comunicação nunca devem ser interrompidos.

Se faz necessário adotar um modelo eficiente para monitorar autonomamente um segmento de rede, identificar padrões de tráfego e, assim, criar um perfil de rede que represente o comportamento regular do tráfego [13]. Esta tese aborda a criação deste perfil, utilizado para a caracterização de tráfego e detecção de anomalias, que é denominado Assinatura Digital do Segmento de Rede utilizando Análise de Fluxo (DSNSF). Quanto à detecção de anomalias e problemas, assim que o modelo para estabelecer um perfil de rede que caracterize o compor-

tamento esperado do tráfego é gerado, qualquer atividade que não esteja em conformidade à este padrão (limiar) pode ser considerada como uma possível anomalia. Essa abordagem é comumente conhecida como um sistema de detecção de anomalias baseado em perfis (baseados em perfil) [9]

Objetivos de Investigação

O objetivo principal desta tese é a construção e avaliação de um sistema de detecção de anomalias baseado na técnica estatística de Análise de Componentes Principais (PCA) para auxiliar a gestão da rede. Esta proposta monitora automaticamente o estado do tráfego por meio de uma abordagem baseada em perfil e assinatura para prever suas tendências normais. Assim, essa previsão pode impedir automaticamente quatro classes de anomalias - DDoS (Distributed Denial-of-Service), DoS (Denial-of-Service), portscan e flash crowds - de prejudicar a disponibilidade e a interoperabilidade da rede.

Para alcançar este objetivo, foram definidos os seguintes objectivos parciais:

- Uma revisão abrangente sobre o tema de detecção de anomalia, cobrindo uma visão geral de vários aspetos pertinentes ao tema, bem como um estudo central sobre as técnicas, métodos e sistemas mais relevantes dentro da área. A revisão foi realizada em cinco dimensões: (i) anomalias de tráfego de rede, (ii) tipos de dados de rede, (iii) categorias de sistemas de detecção de intrusão, (iv) métodos e sistemas de detecção e (v) questões abertas.
- Proposta e implementação de um sistema robusto de detecção de anomalias baseado em perfis que deteta e avisa automaticamente sobre anormalidades de rede através da Análise de Componentes Principais.
- Avaliação de desempenho do sistema de detecção de anomalias usando métricas de avaliação sobre um banco de testes de rede real envolvendo usuários e componentes reais, e sobre um ambiente simulado criado por um software simulador de anomalias de rede.
- Comparação do modelo proposto com outros modelos distintos para fins de validação e determinação de novos meios eficazes de auxiliar na gestão de redes no quesito segurança.

Principais Contribuições

A primeira contribuição desta tese é uma revisão abrangente do estado da arte do domínio de detecção de anomalia sob cinco direções de pesquisa; estudo detalhado das técnicas, métodos e sistemas mais relevantes dentro da área; abordagem das principais desvantagens encontradas nos inquéritos analisados extraídos da literatura; análise dos quatro tipos de anomalia de tráfego categorizados pelo aspeto causal; discussão prospectiva e análise comparativa de outras pesquisas sobre questões abertas e tendências futuras. Este estudo foi publicado na revista *Telecommunication Systems*, da Springer [14].

A segunda contribuição consiste na proposta e avaliação com dados reais do PCADS-AD, um sistema autônomo de detecção de anomalias baseado em perfis. Ele gera uma assinatura digital usando a Análise de Componentes Principais de uma maneira diferente da PCA da literatura, a fim de descrever o comportamento normal de um segmento de rede e usá-lo como base

para a detecção de anomalias. Ainda, o Módulo de Relatório do PCADS-AD resume as informações qualitativas sobre os intervalos anômalos para ajudar o administrador da rede a tomar medidas rápidas para solucionar o problema. Este estudo foi publicado na revista *Applied Soft Computing*, da Elsevier [15].

A terceira contribuição é uma análise comparativa entre o PCADS e o ACODS (um ADS baseado em meta heurística) usando os métodos DTW Adaptativo (ADTW) para detecção de anomalias. Este estudo foi publicado na revista *Journal of Network and Computer Applications*, da Elsevier [16].

Finalmente, a última contribuição desta tese é uma análise comparativa direta de três sistemas de detecção de anomalia de três classes distintas de algoritmos: estatístico (PCADS-AD), previsão (ARIMADS) e clusterização (ACODS), indicando suas semelhanças e divergências quando aplicados a um ambiente real. Este estudo foi publicado na conferência 30th ACM/SIGAPP Symposium On Applied Computing, da ACM [17].

Principais Conclusões

Ao longo desta Tese, foi apresentado e avaliado um novo sistema autônomo de detecção de anomalias baseado em perfil para auxiliar o gerenciamento de redes usando Assinatura Digital de Segmento de Rede usando Análise de Fluxo (DSNSF) gerada via Análise de Componentes Principais. A principal contribuição consiste na aplicação e contextualização do PCA para um ambiente de detecção de anomalias usando atributos de fluxo IP. O sistema cria uma assinatura digital (DSNSF) com base no método estatístico PCA, explorando seu recurso de redução de dimensionalidade, aplicando-o sobre o tráfego da semana anterior, garantindo que tais assinaturas sejam capazes de representar as principais características e padrões do tráfego de rede. Outra contribuição é a criação de limiares de confiança utilizando os autovalores obtidos na fase de caracterização do tráfego, que estabelece um intervalo para o DSNSF onde as variações de tráfego são consideradas normais. Por fim, o Módulo de Relatório do PCADS-AD pode fornecer aos administradores de rede informações úteis sobre anormalidades encontradas.

No Cenário 1, referente à caracterização do tráfego para criação do DSNSF, o sistema proposto obteve bons resultados, apresentando pequenos erros (abaixo de 0,1) e bons índices de correlação (média de 0,8) quando o DSNSF foi comparado com o tráfego real, mostrando que pode ser eficiente na previsão do comportamento esperado de um segmento de rede. Agora, em relação à detecção de anomalias, os resultados referentes a taxas de alarmes falsos e taxa de precisão são encorajadores e, além de alertar o administrador da rede sobre o problema, o sistema proposto também pode fornecer as informações necessárias para resolvê-lo.

Em relação ao módulo de caracterização de tráfego, foram comparados dois métodos diferentes, PCADS e ACODS. De acordo com os resultados do NMSE e do Coeficiente de Correlação, ambos alcançaram resultados semelhantes, produzindo boas previsões de tráfego, podendo verificar apenas pequenos erros entre o DSNSF e o tráfego real.

No módulo de detecção e identificação, o algoritmo DTW Adaptativo (ADTW) investigado

nesta Tese apresentou desempenho satisfatório em relação às taxas de alarmes falsos. Ambos os sistemas produziram melhores resultados ao ajustar o valor de ϕ do ADTW para 20%. Além disso, analisando os gráficos ROC e as taxas de precisão, o PCADS teve um desempenho melhor do que o ACODS. Além disso, a correspondência entre taxas positivas e falsos positivos demonstra que os sistemas são capazes de uma detecção eficaz de comportamento anômalo, mantendo uma taxa satisfatória de alarmes falsos. Além disso, a metodologia de detecção de anomalias pode também fornecer ao administrador de rede importantes estatísticas de tráfego para ajudar na solução de problemas, visando à detecção precisa e rápida de anomalias. Portanto, as metodologias propostas, utilizando PCADS, ACODS e ADTW, são adequadas para auxiliar o gerenciamento da rede, detectando anomalias de tráfego e, conseqüentemente, fornecendo disponibilidade e confiabilidade às redes e seus serviços prestados.

Por fim, no Cenário 3, foi discutido e avaliado o reconhecimento de eventos anormais originado por três sistemas de detecção de anomalias. Embora cada um deles pertença a classes distintas de algoritmos, eles tiveram como objetivo caracterizar o comportamento normal do tráfego de rede criando o DSNSF.

Todos os sistemas produziram DSNSFs semelhantes, igualmente inteligentes na descrição do comportamento normal do tráfego de rede analisado. Consequentemente, as variações encontradas na eficácia da detecção de anomalias estão ligadas ao mecanismo usado para verificar as diferenças entre as assinaturas digitais e o tráfego observado. ARIMADS provou ser mais promissor no reconhecimento de anormalidades do que os outros métodos, uma vez que usa o DSNSF combinado com Lógica Paraconsistente para lidar com o conceito de incerteza. ACODS teve um desempenho inferior em comparação com ARIMADS por conta de falsos positivos relatados durante a análise. Finalmente, o PCADS-AD alcançou a menor taxa de detecção, o que é justificado pela adoção de limites menos flexíveis para a identificação de atividades normais na rede.

Alguns tipos de ataques e anomalias, como DoS, DDoS e Flash Crowds, causam variações de tráfego em atributos de tráfego distintos. O DDoS, por exemplo, afeta apenas o tráfego de pacotes e o número de fluxos. Este trabalho também contribui com a detecção de anomalias de volume de tráfego através da análise de três atributos quantitativos de fluxos IP (bits/s, pacotes/s e fluxos/s), visando a detecção efetiva de diferentes comportamentos anômalos.

A baixa complexidade computacional do processo de caracterização e do método de detecção de anomalias, e os resultados obtidos nos testes apresentados, utilizando dados reais, implicam que a abordagem proposta usando a Análise de Componentes Principais apresenta alta aplicabilidade para identificação automática de anomalias. Além disso, ainda se mostrou um passo promissor para um sistema mais amplo de diagnóstico on-line de anomalias em redes de larga escala.

Perspectivas de Trabalho Futuro

Para concluir este trabalho de investigação, resta sugerir futuros tópicos de estudo resultantes do trabalho de investigação desenvolvido:

- Melhorar o sistema PCADS-AD, minimizar a geração de alarmes falsos e usar outros atributos

de fluxo do tráfego agregado, em um esforço para detectar e identificar outros tipos de ataques e anomalias, como portscans, probing, U2R (User-to-Root) ou R2L (Remote-to-Local).

- Aplicar e avaliar as propostas desta Tese em um ambiente de Rede em tempo real, para sua validação e comparação com os resultados obtidos por outros métodos similares.
- Combinar o modelo proposto com técnicas de *Machine Learning*, melhorando o processo e diminuindo custos.

Abstract

Nowadays, there is a huge and growing concern about security in information and communication technology (ICT) among the scientific community because any attack or anomaly in the network can greatly affect many domains such as national security, private data storage, social welfare, economic issues, and so on. Therefore, the anomaly detection domain is a broad research area, and many different techniques and approaches for this purpose have emerged through the years.

Attacks, problems, and internal failures when not detected early may badly harm an entire Network system. Thus, this thesis presents an autonomous profile-based anomaly detection system based on the statistical method Principal Component Analysis (PCADS-AD). This approach creates a network profile called Digital Signature of Network Segment using Flow Analysis (DSNSF) that denotes the predicted normal behavior of a network traffic activity through historical data analysis. That digital signature is used as a threshold for volume anomaly detection to detect disparities in the normal traffic trend. The proposed system uses seven traffic flow attributes: Bits, Packets and Number of Flows to detect problems, and Source and Destination IP addresses and Ports, to provides the network administrator necessary information to solve them.

Via evaluation techniques, addition of a different anomaly detection approach, and comparisons to other methods performed in this thesis using real network traffic data, results showed good traffic prediction by the DSNSF and encouraging false alarm generation and detection accuracy on the detection schema.

The observed results seek to contribute to the advance of the state of the art in methods and strategies for anomaly detection that aim to surpass some challenges that emerge from the constant growth in complexity, speed and size of today's large scale networks, also providing high-value results for a better detection in real time.

Keywords

Anomaly Detection, Intrusion Detection System, Traffic Characterization, Network Security, Network Management, Principal Component Analysis, Computer Networks, Digital Signature of Network Segment using Flow analysis.

Contents

Dedication	iii
Acknowledgments	v
Foreword	vii
List of Publications	ix
Resumo	xi
Extended Abstract in Portuguese	xiii
Abstract	xix
Keywords	xix
Contents	xxi
List of Figures	xxiii
List of Tables	xxv
Acronyms	xxvii
1 Introduction	1
1.1 Focus and Scope	1
1.2 Problem Definition	2
1.3 Main Objectives	3
1.4 Main Contributions	3
1.5 Thesis Statement	4
1.6 Document Organization	4
2 Related Work	5
2.1 Anomaly Detection	6
2.1.1 Anomaly categorization based on its nature	7
2.1.2 Anomaly categorization based on its causal aspect	8
2.2 Network data types	10
2.2.1 TCP dump	10
2.2.2 SNMP	10
2.2.3 IP flow	12
2.3 Intrusion detection systems	13
2.3.1 IDS Types by monitored platform (data source)	15

2.3.2	IDS types by detection technique	17
2.4	Anomaly detection techniques, methods and systems	20
2.4.1	Statistical methods	20
2.4.2	Clustering methods	26
2.4.3	Finite state machine methods	30
2.4.4	Classification-based methods	31
2.4.5	Information theory	35
2.4.6	Evolutionary computation	38
2.4.7	Hybrid/others	41
2.5	Open Issues	46
3	The Proposed Anomaly Detection System using Principal Component Analysis	49
3.1	Traffic Characterization	49
3.2	Anomaly Detection	53
4	Performance Evaluation	55
4.1	Scenario 1	55
4.1.1	Data Set	55
4.1.2	Evaluation Metrics	55
4.1.3	DSNSF Creation	56
4.1.4	Traffic Characterization Evaluation	56
4.1.5	Anomaly Detection Evaluation	58
4.2	Scenario 2	64
4.2.1	Adaptive Dynamic Time Warping	64
4.2.2	Performance evaluation of PCADS x ACODS	66
4.3	Scenario 3	71
4.3.1	Data Preparation	72
4.3.2	Evaluation	73
4.4	Computational Complexity Analysis	74
5	Conclusion	77
5.1	Final Conclusions	77
5.2	Future Work	79
	References	80

List of Figures

2.1	Chapter Summary	7
2.2	Traffic anomalies categorization	8
2.3	Network data types categorization	10
2.4	Intrusion detection systems categorization	15
2.5	Network-based IDS example	16
2.6	Host-based IDS example	17
2.7	Misuse detection (signature-based) techniques general scheme.	18
2.8	General scheme of anomaly detection (anomaly-based) techniques	19
2.9	Anomaly detection methods, techniques and systems analyzed in this research .	20
2.10	Occurrences (%) of discussed open issues in the analyzed surveys	47
3.1	PCADS-AD System Description	50
3.2	Comparison between DSNSFs generated using eigenvectors of intermediate (a), minimum (b) and maximum (c) significance	52
4.1	NMSE indices over the generated DSNSFs and the real movement of analyzed days using from 1 to 10 weeks	56
4.2	Traffic Characterization example comparing the DSNSFs of bits, packets and num- ber of flows transmitted per second generated for four days in November 2012 .	57
4.3	NMSE tests between the generated DSNSFs and the real traffic from October 15th to November 09th	58
4.4	Correlation tests between the generated DSNSFs and the real traffic from October 15th to November 09th	58
4.5	Alarm generation example, depicting alarm time-frame and attack time-frame .	60
4.6	ROC graph showing TPR and FPR trade-offs of four weeks of tests	62
4.7	Accuracy Rate of four weeks of tests	62
4.8	PCADS-AD Reporting Stage for Flash Crowd simulation	63
4.9	PCADS-AD Reporting Stage for DDoS simulation	63
4.10	Comparison schemes of two time series: (a) by using ADTW and (b) by comparing time series using Euclidian distance.	66
4.11	Traffic Characterization example comparing the DSNSFs of bits, packets and num- ber of flows transmitted per second with the real traffic movement observed at November 08th for both PCADS (a) and ACODS (b) methods	67
4.12	NMSE indices between the generated DSNSFs and the real traffic movement of analyzed days	68
4.13	Correlation Coefficients between the generated DSNSFs and the real traffic move- ment of analyzed days	69
4.14	ROC curve of workdays from October 15th to November 9th for both PCADS and ACODS using different ϕ values	69

4.15 Accuracy Rate of four weeks of tests for both PCADS and ACODS using different ϕ values	70
4.16 Network traffic statistics from two kinds of anomalies	70
4.17 Traffic characterization using the proposed methods. The graphs show the prediction calculated for each analyzed attribute and the observed traffic behavior on November 8th, 2012	75
4.18 General alarm comparison	76
4.19 ROC curves comparing the trade-off between TPR and FPR rates of the proposed methods	76

List of Tables

2.1	A comparison between anomaly detection surveys	6
2.2	Detailed description of most common network abuse anomalies	11
2.3	Comparison between commonly used data sources for network anomaly detection	14
2.4	IDS type classification and organization summary	15
2.5	Comparison of statistical anomaly detection approaches	27
2.6	Comparison of clustering anomaly detection approaches	29
2.7	Comparison of finite state machine anomaly detection approaches.	31
2.8	Comparison of classification-based anomaly detection approaches	36
2.9	Comparison of Information Theory anomaly detection approaches	39
2.10	Comparison of evolutionary computation anomaly detection approaches	42
2.11	Comparison of hybrid/unclassified anomaly detection approaches	45
4.1	Anomaly simulation parameters using Scorpis tool	61
4.2	Anomaly simulation	63
4.3	Artificial Anomaly Simulation on 11/08	73

Acronyms

5G	Fifth-generation
ACO	Ant Colony Optimization
ACODS	Ant Colony Optimization for Digital Signature
ADTW	Adaptative Dynamic Time Warping
AI	Artificial Intelligence
AIS	Artificial Immunology System
ANN	Artificial Neural Network
ARIMA	Auto-Regressive Integrated Moving Average
ARIMADS	Auto-Regressive Integrated Moving Average for Digital Signature
ASCII	American Standard Code for Information Interchange
ASIC	Application Specific Integrated Circuit
ASR	Average Saving Rate
AUC	Area Under the Curve
BLGBA	Baseline for Automated Backbone Management
BP	Back Propagation
CC	Correlation Coefficient
CF	Collaborative Filtering
CNPq	<i>Conselho Nacional de Desenvolvimento Científico e Tecnológico</i>
CPM	Correlational Paraconsistent Machine
CRR	<i>Centro de Referência em Radiocomunicações</i>
CUSUM	Cumulative Sum
DBN	Deep Belief Network
DDoS	Distributed Denial of Service
DM	Data Mining
DoS	Denial of Service
DSNSF	Digital Signature of Network Segment using Flow Analysis
DT	Decision Tree
DTW	Dynamic Time Warping
ECG	Electrocardiogram
e-GRNN	Evolutionary General Regression Neural Network
EL	Eigenvalue Limit
EML	Extreme Machine Learning
ES	Evolution Strategies
FCT	<i>Fundação para a Ciência e a Tecnologia</i>
FN	False Negative
FP	False Positive
FPR	False Positive Rate
FSM	Finite State Machine
FTP	File Transfer Protocol
GA	Genetic Algorithm

GPEN	Genetic Programming-Based Ensembling
GT	Game Theory
HIDS	Host-based Untrusion Detection System
HMM	Hidden Markov Model
HsMM	Hidden semi-Markov Model
HTTP	Hypertext Transfer Protocol
HW	Holt Winters
HWDS	Holt-Winters for Digital Signature
IANA	Internet Assigned Numbers Authority
IBRL	Integrated Bioprocessing Research Lab
ICT	Information and Communication Technology
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engeneering Task Force
IMA	Illegal Memory Access
INATEL	<i>Instituto Nacional de Telecomunicações</i>
IP	Internet Protocol
IPFIX	IP Flow Information Export
IT	Information Technology
KDD	Knowledge Discovery in Databases
KDDCUP	Knowledge Discovery in Databases Cup
KHM	K-Harmonic Means
K-L	Kullback-Leibler
KM	K-Means
KMDS	K-Means for Digital Signature
KPCA	Kernel Principal Component Analysis
LAMS	Local Adaptive Multivariate Smoothing
LTE	Long Term Evolution
MIB	Management information base
MIC	Maximal Information Coefficient
MLP	Multilayer Perceptron
MSPC	multivariate statistical process control
MVE-PCA	Minimum Volume Elliptical PCA
NB	Naive Bayes
NetGNA	Next Generation Networks and Applications Group
NIDS	Network-based Intrusion Detection System
NIPS	Network Intrusion Prevention System
NMS	Network Management System
NMSE	Normalised Mean Square Error
NN	Nearest Neighbor
NNWRw	Neural Network With Random Weights
NP	Neyman-Pearson
NSA	Negative Selection Algorithm
P2P	Peer-to-peer
PBIL	Population-Based Incremental Learning
PC	Principal Component
PCA	Principal Component Analysis

Acronyms

PCADS	Principal Component Analysis for Digital Signature
PL	Paraconsistent Logic
PRC	Precision-recall Curve
PSO	Particle Swarm Optimization
QoS	Quality of Service
RBF	Radial Basis Function
RFC	Request for Comments
RNN	Random Neural Network
ROC	Receiver Operating Characteristic
SCTP	Stream Control Transmission Protocol
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSO	Simplified Swarm Optimization
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
UDP	User Datagram Protocol
UEL	Universidade Estadual de Londrina
URL	Uniform Resource Locator
VLAN	Virtual Local Area Network
WBAN	Wireless Body Area Networks
WMA	Weighted-Majority Algorithm
WSN	Wireless Sensor Network

Chapter 1

Introduction

This section summarizes, in a comprehensive way, the 4 years of research work under the Ph.D. thesis titled "Performance Evaluation of Anomaly Detection Systems." This thesis focuses on the study and proposal of data analysis strategies and methodologies for the monitoring and anomaly detection in networks. The first stage describes the structure of the thesis as well as it defines the problem addressed and the primary objectives of the study. The main contributions of this work for the advance of the state of the art are also presented.

1.1 Focus and Scope

Nowadays, the scientific community has a constant worry about high-efficiency security and quality of service in large-scale networks. The expansion of new communication technologies and services, along with an increasing number of interconnected network devices, web users, services, and applications, contributes to making computer networks ever larger and more complex as systems. Moreover, there is the so called boundless communication paradigm, for next generation networks, which envisages offering anytime, anywhere, anyhow communications to its users and requires the full integration and interoperability of emergent technologies [1]. These issues make it even more complex and challenging to maintain precise network management and lead to serious network vulnerabilities, as security incidents may occur more frequently [2, 3].

Such security instances can be caused either by outsiders, as malicious attacks aiming to shut down services or steal private information, or by inside factors (operational problems), such as configuration errors, server crashes, power outages, traffic congestion, or non-malicious large file transfers [4]. Regardless of the source, such threats, which are commonly called anomalies, can have a significant impact on the network service and end-users and harm computer network operations and availability.

The term anomaly has several definitions. Barnett and Lewis define a data set anomaly as "observation (or a subset of observations) which appears to be inconsistent with the remainder of that set of data" [5]. Chandola et al. express this term as "patterns in data not conforming to a well-defined notion of normal behavior" [6]. According to Lakhina et al., "anomalies are unusual and significant changes in a network's traffic levels, which can often span multiple links" [7]. Hoque et al. define it as "non-conforming interesting patterns compared to the well-defined notion of normal behavior" [8]. By these definitions, it is clear that the concept of normality is one of the main steps toward developing a solution to detect network anomalies.

Although apparently unpretentious, the problem of defining a region denoting normal behavior and marking as an anomaly any occasion contrasting this normal pattern, is defiant. Faster diagnosis, lower complexity and suitable corrections of the causes are the main objec-

tives of the field. Every factor is vital to developing a better anomaly detection approach. The precision and speed factors, alongside with the correct identification of such abnormal events in a timely fashion are critical to reducing significant service degradation, malicious damage, and cost. For this reason, the research community has been developing a lot of models, algorithms, and mechanisms, over the years, to develop better solutions and approaches to guaranteeing the health of ever larger and complex network systems.

In the literature, anomaly detection methods can be classified into two ways: Signature-based and profile-based. Signature-based systems use a prior knowledge about the characteristics of each kind of anomaly to identify potential incidents previously known. Moreover, profile-based approaches create a network profile representing the traffic normal behavior, and traffic anomalies are detected from deviations with respect to this profile [9, 10]. Although signature-based methods have been widely investigated in the literature, they have a clear drawback. It is prerequisite that anomaly signatures are known in advance, hampering the recognition of new anomalies. Also, signature-based methods can be avoided by malicious sources by tampering anomaly signatures. In contrast, a profile-based system creates a baseline profile of the normal network activity, eliminating the need of prior knowledge about the nature and properties of anomalies. This trait leads to some advantages: The possibility of discovering new and unforeseen types of anomalies; the detection of insider attacks; and also makes it difficult for an attacker to know with conviction what malicious action it can carry out without being detected by the system [9, 11]. Thus, this thesis proposal is to create an autonomous profile-based monitoring system capable of identifying the normal network behavior by adopting an efficient method for traffic characterization in order to create a baseline profile of normal traffic to discover possible anomalies in the traffic.

1.2 Problem Definition

To hold a complete traffic analysis and monitoring in large-scale network systems is almost an impossible task to be manually performed by a network administrator. The high connection speeds combined with the large and growing number of links and segments make this task even more complex [12]. In addition, there exists a need for agility in detecting and preventing problems, as the network administrator must work proactively to avoid interruptions in the operation of the network, since one of the ICT (Information and Communication Technology) governance premises is that communication services should never be interrupted.

It is necessary to adopt an efficient model to autonomously monitor a network segment, identify traffic patterns and thus create a network profile that represents normal traffic behavior [13]. This thesis addresses the creation of this profile, used for the traffic characterization and anomaly detection, which is called Digital Signature of Network Segment using Flow Analysis (DSNSF). Regarding the detection of anomalies and problems, as soon as that there is a model to establish a network profile that characterizes the expected traffic behavior, any activity that differs from this standard (threshold) may be considered as a possible anomaly. This approach is commonly known as an anomaly detection system based on profiles (profile-based) [9]

1.3 Objectives

The main objective of this thesis is the construction and evaluation of an anomaly detection system based on the statistical technique Principal Component Analysis to assist network management. This proposal automatically monitors the state of traffic through a hybrid approach using profile-based and signature-based procedures to predict its normal tendencies. Thus, this prediction can automatically prevent three classes of anomalies- DDoS, DoS, and flash crowds -from harming network availability and interoperability.

To achieve this main objective, the following partial objectives have been defined:

- A comprehensive review on the anomaly detection subject, covering an overview of a background analysis as well as a core study on the most relevant techniques, methods, and systems within the area. The review was performed under five dimensions: (i) network traffic anomalies, (ii) network data types, (iii) intrusion detection systems categories, (iv) detection methods and systems, and (v) open issues.
- Proposal and implementation of a robust autonomous profile-based anomaly detection system that automatically detect and warn about network abnormalities through the Principal Component Analysis method.
- Performance evaluation of the anomaly detection system using robust metrics over a real network testbed involving real users and components, and over a simulated environment created by a network anomaly simulation software.
- Comparison of the proposed model with other distinct models aiming to validate the system and understand different ways to assist network management.

1.4 Main Contributions

The first contribution of this thesis is a comprehensive survey review of state-of-the-art anomaly detection domain: review the anomaly detection subject under five research directions; detailed study of the most relevant techniques, methods, and systems within the area; address the main drawbacks found in the analyzed surveys extracted from the literature; analysis of the four traffic anomaly types categorized by the causal aspect; forward-looking discussion and comparative analysis of other surveys regarding open issues and future trends. This study was accepted for publication in the Telecommunication Systems journal of Springer [14].

The second contributions consist in the proposal and performance evaluation of PCADS-AD, an autonomous profile-based anomaly detection system. It generates a digital signature by using Principal Component Analysis in an unusual way than the PCA from the literature, in order to describe the normal behavior of a network segment, and then using it as the basis for anomaly detection. Furthermore, the PCADS-AD Reporting Stage summarizes the qualitative information about the anomalous intervals in order to assist the network administrator to take quick measures to solve the problem. This study was accepted for publication in the Applied Soft Computing journal of Elsevier [15].

The third contribution is a comparative analysis between PCADS and ACODS (a metaheuristic-based ADS) using the Adaptive DTW methods (ADTW) for anomaly detection. This study was accepted for publication in the Journal of Network and Computer Applications of Elsevier [16].

Finally, the last contribution of this thesis is an straightforward comparative analysis of three anomaly detection systems of three distinct algorithm classes: statistical (PCADS-AD), forecasting (ARIMADS) and clustering (ACODS), stating their similarities and divergences when applied to a real network analysis. This study was accepted for publication in the 30th ACM/SIGAPP Symposium On Applied Computing of ACM [17].

1.5 Thesis Statement

Traffic prevention from computer networks should be automatically monitored in order to detect problems, attacks, and abnormalities that may harm the whole system, by guaranteeing availability, operability and security. Despite the complexity and size of today's network systems, this study demonstrates that the traffic prediction and anomaly detection can be precise using the statistical learning method Principal Component Analysis in an alternative mode. Furthermore, this study claims that this model leads to competitive outcomes regarding false positive and available additional data to ease a deep analysis about the problem found.

1.6 Document Organization

This thesis consists of 5 Chapters, which are organized as follows. The first chapter presents the scope of the thesis, focusing the topics under study, the definition of the problem and primary objectives. The research hypothesis, the main contributions, and the document's organization are also included in this chapter.

Chapter 2 presents a comprehensive survey focusing on the main aspects of anomaly detection domain. It is divided into two main parts: The anomaly detection background, which discuss anomalies and attacks, data types and Intrusion Detection systems (IDS); and the anomaly detection core study, which surveys many techniques, methods and systems developed for anomaly and intrusion detection using many distinct algorithm classes.

In Chapter 3, the proposed hybrid anomaly detection system PCADS-AD (Principal Component Analysis for Digital Signature and Anomaly Detection) is presented.

Chapter 4 presents the performance evaluation of the proposed ADS compared with other anomaly detection systems (ADS) based on different algorithm classes. The evaluation is presented in three test scenarios.

Finally, Chapter 5 concludes the Thesis, summarizing all the main conclusions of the thesis drawn throughout the document and proposes several insights and suggestions for future work.

Chapter 2

Related Work

Nowadays, there is a huge and growing concern about security in information and communication technology (ICT) among the scientific community because any attack or anomaly in the network can greatly affect many domains such as national security, private data storage, social welfare, economic issues, and so on. Therefore, the anomaly detection domain is a broad research area, and many different techniques and approaches for this purpose have emerged through the years.

Researchers have been studying the anomaly detection subject since the early 19th century, and so far, they have produced a multitude of papers, each using a variety of techniques, from statistical models, up to evolutionary computation approaches. Nevertheless, it is not a straightforward task to identify and categorize all existing anomaly detection techniques. Plenty of topics must be considered, such as anomaly types, system types, techniques and algorithms used, as well as technical dilemmas such as processing costs and network complexity. Therefore, this leads to the fragmented literature available today, in which many works try to summarize everything but are unable to show the bigger picture of the anomaly detection spectrum.

As in [18] and [9], the focus is just on the most popular techniques and methods, such as machine learning, clustering and statistical approaches. Still, surveys such as [19] and [20] briefly discuss the whole problem statement, setting aside relevant topics such as data set, challenges, and recommendations. Marnerides *et al.* [21] have a reviewed anomaly detection over backbone networks. Although each of those inspected surveys summarizes many important topics pertaining to anomaly detection, they are not entirely complete. For instance, some of them emphasize anomaly types but do not cover all kinds of methods while others research upon vast approaches but forget about the basis of intrusion detection systems and data input, and so on. For this reason, the main objective is to review the most important aspects pertaining to anomaly detection, covering an overview of a background analysis as well as a core study on the most relevant techniques, methods, and systems within the area. Therefore, in order to ease the understanding of this chapter's structure, the anomaly detection domain was reviewed under five dimensions: (i) network traffic anomalies, (ii) network data types, (iii) intrusion detection systems categories, (iv) detection methods and systems, and (v) open issues. Table 2.1 provides a comparison between some anomaly detection surveys with regard to the variety of techniques they address.

This chapter is organized as follows. Section 2.1 defines, categorizes, explains, and provides examples of most common types of network anomalies. Section 2.2 gives a brief explanation of network data types used as input in anomaly detection systems. Section 2.3 gives a complete overview of intrusion detection systems and the differences between each approach. Section 2.4 is the core section, which lists many anomaly detection methods and systems using a variety of techniques and algorithms of different nature and purpose. Finally, section 2.5

Table 2.1: A comparison between anomaly detection surveys

Surveys		Patcha and Park [9]	Chandola <i>et al.</i> [6]	Weiyu <i>et al.</i> [20]	Thottan <i>et al.</i> [18]	Yu <i>et al.</i> [19]	Bhuyan <i>et al.</i> [10]	Marnerides <i>et al.</i> [21]	Ahmed <i>et al.</i> [22]	This survey
Year	Content	2007	2009	2009	2010	2012	2014	2014	2016	2018
Traffic anomalies by nature	Point		✓				✓		✓	✓
	Collective		✓				✓		✓	✓
	Contextual		✓				✓		✓	✓
Traffic anomalies by causal aspect	Operational									✓
	Flash Crowd									✓
	Measurement									✓
	Network attack						✓	✓	✓	✓
Network data types	TCP dump									✓
	SNMP				✓					✓
	IP flows				✓					✓
IDS overview	-		✓				✓			✓
Detection techniques, methods and systems	Statistical	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Clustering		✓			✓	✓		✓	✓
	Classification		✓	✓			✓		✓	✓
	Finite State Machines			✓			✓			✓
	Information Theory		✓				✓	✓		✓
	Evolutionary Computation						✓			✓
	Hybrid/Others						✓			✓

summarizes everything discussed in previous sections into some topics considered as open challenges in the anomaly detection domain. Figure 2.1 shows all contents presented and discussed within the survey.

2.1 Anomaly Detection

One of the first tasks envisioned by researchers in creating an anomaly detection model is the correct identification and definition of the problem statement. It means that there must be prior knowledge about what type of anomaly researchers would deal with. There are several types of network traffic anomalies, and each author surveying this topic addresses them differently. For the sake of simplicity, and after analyzing and studying the anomaly context, Figure 2.2 illustrates its categorization and all points that are covered in this section. Network anomalies can be categorized giving two relevant properties: according to their nature (grouped by how they are characterized, regardless of whether they are malicious or not); and according to their causal aspect (distinguished depending on their cause, regarding either their malicious or non-malicious aspect).

Chapter 2. Related Work

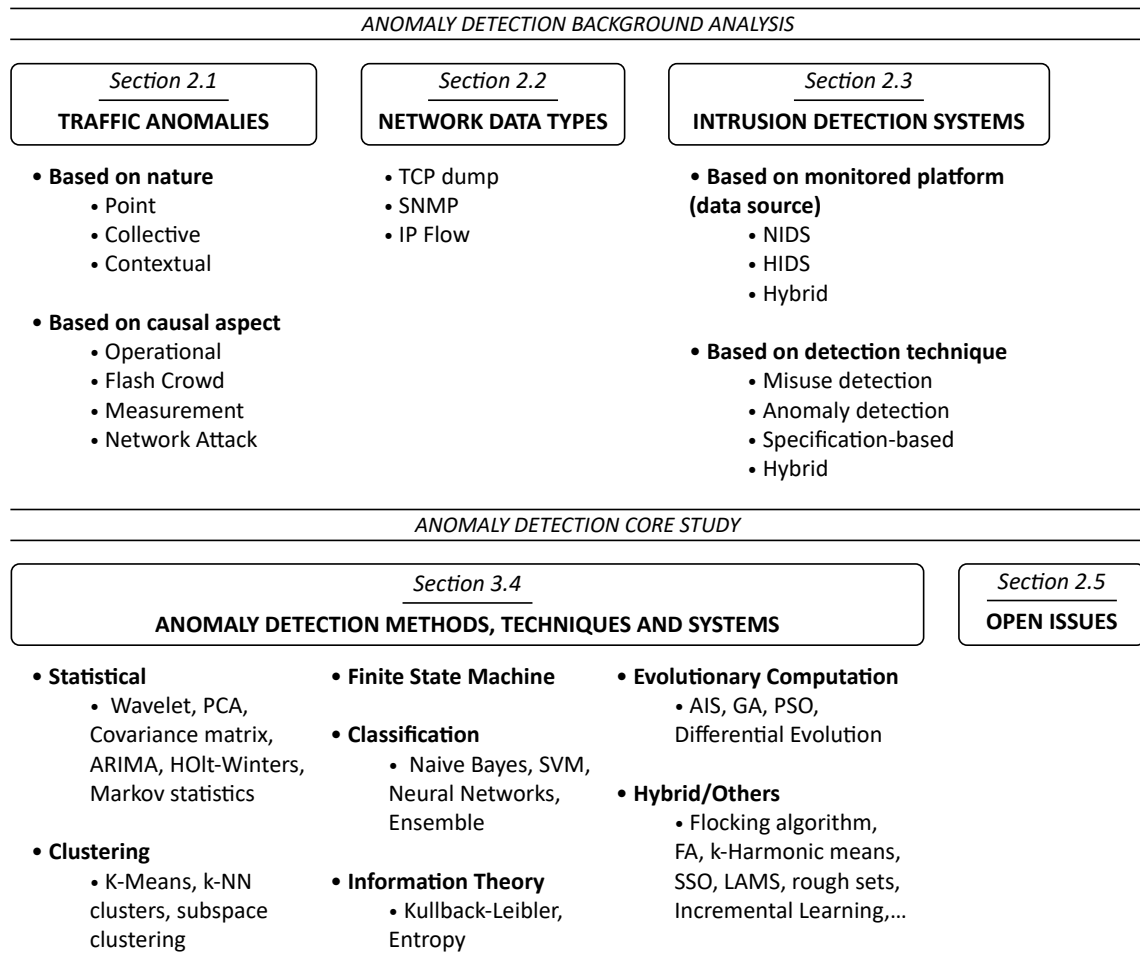


Figure 2.1: Chapter Summary

2.1.1 Anomaly categorization based on its nature

The nature of an anomaly is an important aspect of an anomaly detection technique. Depending on the context within which an abnormality is found, or on how it occurred, it can be or not be an abnormality. This aspect can direct how the system will handle and understand mined and detected anomalies. Based on their nature, there are three categories of anomalies: point anomalies, collective anomalies, and contextual anomalies [6, 10, 22].

A point anomaly is the deviation of an individual data instance from the usual pattern/behavior. These anomalies are the simplest ones, and because of that, they are the focus of most researchers. For better understanding, suppose that the daily spending of a person is one hundred dollars; then, on a specific day, he spends three hundred dollars. This situation characterizes a point anomaly [6, 22].

A collective anomaly occurs when only a collection of similar data instances behaves anomalously with reference to the whole dataset. In a collective anomaly, individual anomalous behaviors themselves are not considered anomalies; however, their collective occurrence is considered an anomaly. A point anomaly occurring continuously for an extended period or in a cluster amid background data is a collective anomaly. Consider this example: in a sequence

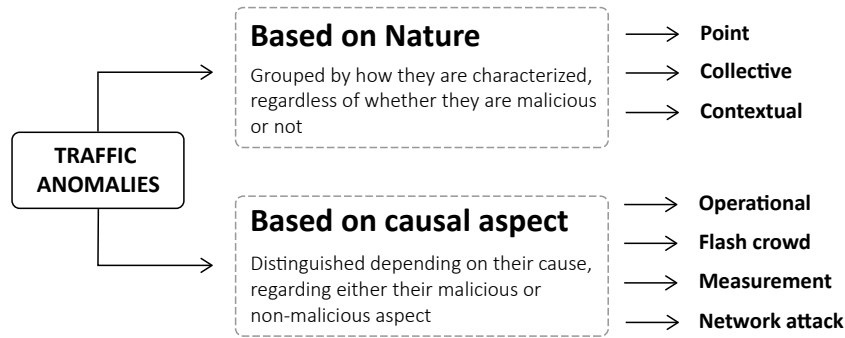


Figure 2.2: Traffic anomalies categorization

of actions in a computer like “...HTTP-web, buffer-overflow, HTTP-web, HTTP-web, FTP, HTTP-web, SSH, HTTP-web, SSH, buffer-overflow, FTP...”, the sequence is a collective anomaly. The individual events occurring in other positions in the sequence are not anomalies; however, the sequence matches a web-based attack by a remote machine followed by the copying of data from the host computer to a remote destination via FTP. Another common example is the ECG exam output, in which low values observed over a long period indicate an anomaly, while one unique low value is not considered abnormal [10, 22].

Contextual Anomalies, also called conditional anomalies, are events considered as anomalous depending on the context in which they are found. Two sets of attributes define a context (or the condition) for being an anomaly, both of which must be specified during problem formulation. Contextual attributes define the context (or environment); for instance, geographic coordinates in spatial data or time in time-series data specifies the location or position of an instance, respectively. On the other hand, behavioral attributes denote the non-contextual features of an instance, i.e., indicators determining whether or not an instance is anomalous in the context [6, 22, 23]. Consider a time-series data set describing the average bits/s of network traffic in a set of days (contextual attribute), in which every day, at 0 h, the server does a regular backup (behavioral attribute). Although the backup generates an outlier in the traffic series, it may not be anomalous since it is normal behavior due to a regular backup. However, a similar traffic outlier at 12 h could be considered a contextual anomaly.

2.1.2 Anomaly categorization based on its causal aspect

The causal aspect distinguishes anomalies depending on their cause, regarding either their malicious or non-malicious aspect. Anomalies are not always related to attacks intended to harm computer systems or steal information. They can be both events caused by human/hardware failure, bugs or private users when demanding heavy traffic usage, for instance. Thus, as found in Barford *et al.* [24] and Marnerides *et al.* [16], anomalies are grouped into four categories: operational/ misconfiguration/ failure events; flash crowd/ legitimate but abnormal use; measurement anomalies; and network abuse anomalies/ malicious attacks (or simply, network attacks) [25, 24].

Operational events (also called Misconfiguration events or Failures) are non-malicious issues, which may occur in a network system mostly by hardware failures, software bugs or

Chapter 2. Related Work

human mistakes. Server crashes, power outages, configurations errors, traffic congestion, non-malicious large file transfers, inadequate resource configuration, or significant changes in network behavior caused by imposing rate limits or adding new equipment, are all examples of this category of anomaly [4]. Such problems can be perceived visually by nearly abrupt changes in bit rate, which appear steady but occur at a different level over a time period [25].

Legitimate but not abnormal use is commonly referred to as flash crowds. Flash crowds are large floods in traffic, which occur when rapid growth of users attempts to access a specific network resource, causing a dramatic surge in server load. Anomalies in this category consist of legitimate requests, which are usually an aftermath of mutual reaction to hot events but far bigger than the load which the system can handle. Flash crowds may occur when a contest result is published on a URL, or when an e-commerce website announces a big sale, or even due to software release. Although it is not malicious, if there is not enough time to react and provide the necessary resources to handle overload demand, these flash events can seriously flood or lead to complete web service failure [26, 27]. Flash crowd behavior is related to the rapid growth of particular traffic flow types, such as FTP flows, or the gradual fall of a well-known destination over time.

Measurement anomalies are other issues, which are not network infrastructure problems, abnormal usage, or malicious attacks. These anomalies are related to collection infrastructure problems and problems during data collection. Examples are the loss of flow data caused by router overload, or when there is a collection of infrastructure problems and the UDP NetFlow transport to the collector becomes unreadable.

Network abuse anomalies (or network attacks) are a set of malicious actions aiming to disrupt, deny, degrade or destroy information and services from computer network systems, compromising their integrity, confidentiality or availability. Numerous types and classes of attacks currently existing may vary from simple email spam to intrusion attacks on critical network infrastructures. Worms, malicious resource abuse, bug exploits and unauthorized access are some examples of common computer attacks. According to Ghorbani *et al.* [28], attackers gain access to a system, or limit the availability of that system through some general approaches. These are:

- **Social Engineering:** when an attacker manipulates people to obtain confidential information, making use of hostile persuasion or other interpersonal tactics [29]. Examples are email phishing and email Trojan horses;
- **Masquerading:** this is a type of attack in which the attacker uses a fake identity to gain unauthorized access or greater privileges in a system through official access identification. The attacker illegitimately poses or assumes the identity of another legitimate user, generally by using stolen IDs and passwords [30].
- **Implementation Vulnerabilities:** these are cases in which the attacker exploits software bugs in their targets, such as software, services or applications, in order to gain unauthorized access. Examples are the buffer overflow vulnerability or the mishandling of temporary files.
- **Abuse of Functionality:** malicious activities performed by attackers excessively performing a legal action in order to congest a link or cause a system to fail. A denial-of-service

performed on a web-login system by flooding it with valid usernames and arbitrary passwords in order to lock out authentic users, when the allowed login retry limit is exceeded, constitutes an abuse of functionality.

Based on those general approaches of network abuse anomalies (network attacks), there are various classes of attacks. Table 2.2 shows the main attack, which commonly harms computer networks and is the major target of anomaly detection mechanisms.

2.2 Network data types

Another essential step required for building an anomaly detection system is choosing the network data source. The nature of the selected data set may dictate which types of anomalies the system can detect. One needs to choose a data source correctly depending on what kind of anomalies and IDS approaches are intended as the focus of the research. Because of that, accurate data characterization results in the better performance of the anomaly detection system. This section presents some of the most popular sources used in the anomaly detection subject and Figure 2.3 summarizes them.

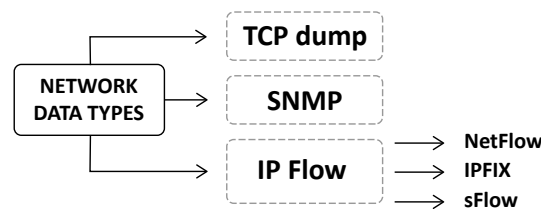


Figure 2.3: Network data types categorization

2.2.1 TCP dump

Tcpdump is a packet analyzer tool used to monitor packets on a computer network. It shows the headers of TCP/IP packets passing through the network interface. It is a tool for network packet capturing and analysis and is recommended to professionals who need to perform monitoring and maintenance on computer networks, as well as to students who want to understand the operation of the TCP/IP protocol stack. Nevertheless, this type of data is not used as much nowadays due to its limited information.

2.2.2 SNMP

The Simple Network Management Protocol (SNMP) [34] is one of the widely used standards for managing IP network components. This protocol has a client-server structure (SNMP managers and SNMP agents) which runs throughout the UDP protocol [31]. SNMP data has been used on intrusion detection systems, since it is useful when it comes to collecting accurate network activity data at a single host level. All collected data are stored, as SNMP objects, in a hierarchical database called MIB (Management Information Base). SNMP objects are summary

Chapter 2. Related Work

Table 2.2: Detailed description of most common network abuse anomalies

Attack	Definition	Examples	T ^a
Virus	<ul style="list-style-type: none"> • Piece of code inserted into a file or program which replicates itself without the user's permission. • Harmful activities: theft of private information, data corruption, spam messages. • Needs human intervention to abet its propagation. 	Rootkit.Sirefef.Gen, Trivial.88.D	IV
Worm	<ul style="list-style-type: none"> • Self-replicating software designed to spread through the network. • Exploit security or policy flaws in widely used services. 	Morris, CodeRed, Nimda	IV
Trojan	<ul style="list-style-type: none"> • A piece of program masquerading as a benign application, when in fact it secretly performs malicious activities. • They do not replicate as viruses and worms do but can be just as destructive. 	ZeroAccess Rootkit, Beast, Zeus	SE
Buffer Overflow	<ul style="list-style-type: none"> • Takes over programs through buffer vulnerabilities to execute arbitrary code by storing more data in a buffer than the buffer can hold. • Can corrupt or overwrite valid data held in a buffer 	-	IV
Denial of Service (DoS)	<ul style="list-style-type: none"> • Malicious attempts to deny access to shared network resources or services. • Generally, it uses significant packet volume containing useless traffic to congest and waste resources serving legitimate traffic. It can be a single or multi-source attack. 	SYN flood, HTTP flood, ping of death (PoD), RUDY, teardrop, Slowloris	AF
Distributed DoS (DDoS)	<ul style="list-style-type: none"> • DDoS are DoS attacks; they are easy to launch and difficult to locate their source since they are implemented by a group of computers (botnet). • Defeat the target server while keeping their identity unknown by using compromised computers. 	UDP flood, TCP flood, Slowloris, Zero-day DDoS, NTP amplification	AF
Distributed Reflective DoS (DRDoS)	<ul style="list-style-type: none"> • Attacks that just cannot be addressed by traditional on-premise solutions. These use legitimate hosts (reflectors) to flood a large number of response packets to the target system by using spoofed IP addresses. • The attacker sends many requests with a spoofed source IP address (the target server address) to legitimate nodes (reflectors), which reply with several voluminous responses to the spoofed IP (target server), thus flooding the victim. 	Smurf attack, Fraggle attack	AF
Stealthy attack	<ul style="list-style-type: none"> • Quietly introduced and remain undetected by hiding the evidence of the attacker's actions. 	Stealthy packet dropping	IV
Physical attack	<ul style="list-style-type: none"> • An endeavor to harm physical components of a computer or network. • Attackers with physical access to a computer can retrieve encryption keys from a running operating system, for instance. • As soon as a computer is physically controlled, it can be destructive. 	Cold Boot attack, Stoned Boot, Evil Maid	AF
Password attack	<ul style="list-style-type: none"> • Attempts to gain passwords. • They are specified by a series of unsuccessful logins (brute force) in a short period of time. 	Dictionary attack, phishing attack	IV
Cyber Reconnaissance	<ul style="list-style-type: none"> • Information gathering attack. • Gathers information on network systems and services. • Exploits vulnerabilities or weaknesses by scanning or probing devices or systems. 	Ping sweeps, Port scans, packet sniffers	IV
Probe	<ul style="list-style-type: none"> • It is accomplished before an attacker launches an attack on a given target. • Scans or probes the target's network or host by searching for vulnerabilities, open ports, valid IP addresses, services offered, operating system used, etc. 	IPsweep, portsweep	IV
User to Root (U2R)	<ul style="list-style-type: none"> • Consists of unauthorized access to local superuser privileges by starting as a regular unprivileged user. • U2R attacks may end in substantial loss of time and money. 	Loadmore, perl, Xterm	M
Remote-to-Local (R2L)	<ul style="list-style-type: none"> • Unauthorized access via a remote machine • Remote to local attack detection using a supervised neural network 	FTP write, Warezmaster	M

^aType: Network attack type; SE=Social Engineering, M=Masquerading, IV=Implementation Vulnerabilities, AF=Abuse of Functionalities.

traffic data constructed by the aggregation of raw data (pcap records) collected mostly by TCP dump tools [21].

Although efficient in their proposals, the works by Cabrera *et al.* [32] and Yu *et al.* [33] are limited to detecting only DoS/DDoS attacks, since these are volume anomalies and SNMP objects rely on volume attributes (bits and packet counts). As presented in Moises *et al.* [34] and Zarpelao *et al.* [35], the proposed alarm systems developed over SNMP data have shown high anomaly detection rates by combining clustering and parameterizing techniques. However, none of them had any other information about unknown anomalies, despite the alarms being triggered.

A significant advantage is that SNMP is still a widely deployed protocol with available fine-grained data. It is used in traditional network management tools for measuring performance parameters such as error counter interfaces and traffic volume. Packet and bit interface counters are useful; however, nowadays, understanding which IP addresses are the source and destination of traffic and which TCP/UDP ports are generating traffic is vital.

2.2.3 IP flow

IP flow analysis is a complete management technology that has been used as an alternative to the SNMP protocol. The development of new services and the increasing complexity of networks led to a need for more detailed information on transmitted data, which is essential in understanding application behavior, users, business departments and other structures relying on the network for their operation.

Accordingly, using flow management tools and protocols allows the construction of a detailed database composed of essential traffic information, enabling the better understanding of more subjective aspects of network operation [36, 37]. Thus, it was necessary to go beyond the limited bit and packet counters provided by SNMP in order to characterize more specific traits in the traffic, showing network trends and behavior. Moreover, although packet and byte interface counters are useful, knowing the source and destination IP addresses of the traffic, and which applications are producing it, is invaluable [38].

As a result of these constraints, Cisco Systems presented the NetFlow protocol in 1996 [37, 39] and pioneered the introduction of flow structure. A flow [40] is defined as a set of IP packets passing through an observation point over a pre-defined time interval. All packets constituting a flow have a set of common properties including source/destination IP addresses and TCP/UDP ports, VLAN, application protocol type (layer 3 from the OSI model) and TOS (Type of Service). Moreover, a flow also has some other important attributes, such as byte and packet counts, timestamps, class of service (CoS) and router/switch interface. NetFlow introduced a new practice to assist network management. This was the NetFlow probe, embedded into the network devices (switches), which captures all packets coming through the switch and aggregates them into IP flows according to their common properties. Then, after the timeout of the previously established maximum flow duration, flows were exported out to a collector responsible for analyzing the flow data [41]. NfSen [42] and nTop [43] are the most common graphical applications enabling the analysis of exported flow data.

Chapter 2. Related Work

Besides NetFlow, there are other protocols that have emerged for the same purpose. sFlow was introduced by the InMon Corp. in 2001 [44, 45]. Its major difference to other protocols is the usage of random sampling mechanisms during traffic flow aggregation. This feature is appropriate for high-speed networks (gigabit or more). By the year 2008, the Internet Engineering Task Force (IETF) standardized the export of IP flow information from routers, probes, and switches by introducing the IPFIX (IP Flow Information Export) protocol [40]. IPFIX was based on NetFlow version 9; it was developed with more flexible data handling and is able to operate regardless of which transport protocol or message formats are used. Recently, two NetFlow enhancements appeared. Flexible NetFlow uses an extensible format and can export other features apart from the traditional ones. It also has the immediate cache concept, which lets the direct export of flow information without hosting a local cache. NetFlow-lite [46, 47] comes at a lower price tier, compared to standard NetFlow, due to not using expensive customer application specific integrated circuits (ASIC). It offers flexibility, similar network visibility and maintains the same packet forwarding performance.

There are several advantages of using flow traffic to detect anomalies [48, 49, 50]:

- Lower processing cost. Since flow-based IDSs are based only on packet headers, they only process a small number of flows compared to the big amount of packets processed in packet-based approaches;
- Reduced privacy issues, such as the packet's payload, are not considered in the analysis;
- Detailed traffic data, mainly regarding NetFlow v9 and IPFIX.

Regarding the disadvantages of developing anomaly detection methods under IP flow data, most of them rely on the following:

- Untrustworthy state of UDP protocol and drawbacks of SCTP (Stream Control Transmission Protocol) in confronting scenarios, where multiple network interfaces (routers and switches) need to interact with multiple NetFlow data collectors.
- There is also difficulty in understanding end-to-end traffic, since it may be passing through many hops and routing paths and changing dynamically.
- Although sampling techniques for both flow and packets are efficient in reducing the load of exported and aggregated traffic, respectively, they offer a non-reliable view of the entire network operation. Many researchers have discussed the problems and proposed solutions to optimizing sampling mechanisms; namely, Bartos *et al.* [51], Zhang *et al.* [52] and Silva *et al.* [53].

Table 2.3 compares the two data sources discussed in this section.

2.3 Intrusion detection systems

Intrusion Detection Systems (IDS) are automated defense and security systems for monitoring, detecting and analyzing hostile activities within a network or a host. Although the name "Intrusion detection" suggests that these systems actually detect "intrusions", it is not that simple. Kemmerer and Vigna [54] say that, in fact, IDSs do not detect intrusions at all, but they

Table 2.3: Comparison between commonly used data sources for network anomaly detection

Source	Advantages	Disadvantages
TCP dump	- Provides comprehensive information about the operation of the TCP/IP protocol stack	- Limited information
SNMP	- Widely deployed protocol - Available fine-grained data	- Limited information. Only packet and bit interface counters and IP/ports.
IP Flow	- Lower processing cost - Based only on packet headers - Reduced privacy issues - Detailed traffic data	- Untrustworthy state of UDP protocol - Drawbacks of SCTP in confronting scenarios where multiple network interfaces need to interact with multiple flow data collectors - Difficult to understand end-to-end traffic - Sampling techniques offer a non-reliable view of the entire network

are only able to recognize evidence of intrusions, either during or after the circumstance.

Additionally, Lee and Stolfo [55] state that there are four essential elements to be considered when creating an IDS: resources to protect (accounts or file systems, for instance); models to identify the typical behavior of these resources; techniques that compare the actual activities of these resources with their normal behaviors; and finally, identify what is considered abnormal or intrusive. Apart from these basic IDS functions, they may also be able to provide reports for network administrators and track user policy violations as well as to take self-measures to stop threats or correct problems [9, 10, 56].

An IDS detects hostile activities by either monitoring network traffic, gathering packets (mostly as a kind of sniffer) to analyze possible incidents, or by analyzing computational system events (such as log files, for instance), in search of security policy violations, unusual use, etc. These incidents may occur due to various reasons, from malware (worms, spyware, etc.) to unauthorized access attacks. The goal of any IDS is to guarantee the security of a network or computer system with regard to confidentiality, integrity, and availability. A firewall is commonly the first defensive line in a network and an IDS is used when there is evidence of an intrusion/attack, which the firewall was unable to stop or mitigate. The IDS then works as the second line of defense. Furthermore, the task is difficult, and in fact intrusion detection systems do not detect intrusions at all, they only identify evidence of intrusions, either while they are in progress or after the fact.

IDSs can be categorized in many ways [57]. Depending on the monitored platform (data source), IDSs are divided into three types: network-based IDS (NIDS), host-based IDS (HIDS), and hybrid. Furthermore, regarding the technique of detecting unusual activity, IDSs can be categorized into four types: anomaly-based IDS, signature-based IDS, specification-based IDS, and hybrid. Figure 2.4 and Table 2.4 condense the seven aforementioned IDS types, which are presented and discussed thoroughly in subsequent sections.

Chapter 2. Related Work

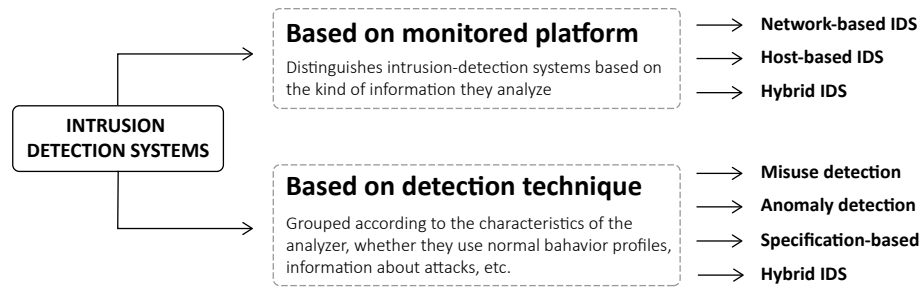


Figure 2.4: Intrusion detection systems categorization

Table 2.4: IDS type classification and organization summary

Classification	IDS Type / Description	Advantages	Disadvantages
DATA SOURCE / MONITORED PLATFORM	Network-based (NIDS)	<ul style="list-style-type: none"> - Monitor both inbound and outbound network traffic - Detect network-specific attacks, such as denial-of-service - Detect known worms and viruses, flash crowds, port scan 	<ul style="list-style-type: none"> - Difficulty in processing all packets from a large and overloaded network - Failure to recognize attacks launched during periods of intense traffic - Unable to analyze encrypted packets - Demand for more sensors in today's large networks is costly
	Host-based (HIDS)	<ul style="list-style-type: none"> - Detect Local suspicious activities - Detect attacks based on encrypted data, since they are located on the destination - Privilege abuse, buffer overflows 	<ul style="list-style-type: none"> - Incomplete network picture - Since they are agent-based, support for different operating systems is required.
	Hybrid	<ul style="list-style-type: none"> - Aggregate benefits of both approaches - Overcome many drawbacks 	<ul style="list-style-type: none"> - Get distinct approaches to interoperate and coexist in a single system
DETECTION TECHNIQUE	Misuse Detection - Use of prior-knowledge attack database (signatures)	<ul style="list-style-type: none"> - High detection accuracy - Low false alarm rate 	<ul style="list-style-type: none"> - Unable to detect unknown anomalies - Difficult and time-consuming task to build and update signatures
	Anomaly Detection - Profile representing normal network behavior	<ul style="list-style-type: none"> - Detect both known and unknown anomalies - Discover new attacks (and use on signature-based IDSs) - No demand for prior knowledge 	<ul style="list-style-type: none"> - High false positives and false negatives - Less efficient in dynamic network environments - Demand time and resources to construct the profile
	Specification-based - Set of constraints to describe and monitor the operation of a program or protocol	<ul style="list-style-type: none"> - Unknown attacks discovery - Low false positive rates - Resistant to subtle attack changes 	<ul style="list-style-type: none"> - Complexity - Elaboration of detailed specifications and constraints is costly and time consuming - Restricted to the proper operation of a program or protocol
	Hybrid	<ul style="list-style-type: none"> - Aggregate benefits of the three approaches - Overcome many drawbacks 	<ul style="list-style-type: none"> - Get distinct approaches to interoperate and coexist in a single system

2.3.1 IDS Types by monitored platform (data source)

2.3.1.1 Network-based IDS (NIDS)

A network-based IDS is deployed in order to detect intrusions in network data over network connections and to protect all network nodes. Since intrusions usually occur as irregular

patterns, this kind of IDS analyzes and models traffic to identify the occurrence of regular traffic and suspicious activities. They are composed of a set of sensors placed at many network points in order to monitor traffic. Each sensor performs a local analysis and reports suspicious activity to a central management console. A network-based IDS is capable of gathering and analyzing entire transmitted packets as well as their payloads, IP addresses, and ports.

NIDS are effective for monitoring both inbound and outbound network traffic. This type of IDS ensures that a large network can be monitored with only a few installed IDSs, as long as they are well positioned. It is usually simple to add this type of IDS to a network and they are considered well secured against attacks. However, they have some disadvantages, such as the difficulty in processing all packets from a large and overloaded network. Thus, they may fail to recognize an attack launched during periods of intense traffic. Moreover, many of the advantages of network-based IDSs do not apply to more modern networks based on switches since they segment the network and require enabling monitoring ports for the sensors to function properly. Port mirroring or spanning is used to enable a complete view in a switched network; however, this causes overhead.

Another disadvantage of network-based IDSs is that they are unable to analyze encrypted network packets, since those appear only on the target machine. Finally, since NIDSs can detect the presence of suspicious activities, there is no reassurance for their success or failure [10, 56, 58]. Figure 2.5 illustrates a conventional network-based IDS.

2.3.1.2 Host-based IDS (HIDS)

A Host-based IDS is set to operate on specific hosts (single PCs). Its focus is to monitor events on the host and detect local suspicious activities, i.e., attacks performed by users of the monitored machine or attacks occurring against the host where it operates.

Since this type of IDS is designed to operate with only a host, it is capable of specific tasks, which are not possible with an NIDS, such as integrating code analysis, detecting buffer overflows, monitoring system calls, privilege misuse, privilege abuse, system log analysis, and

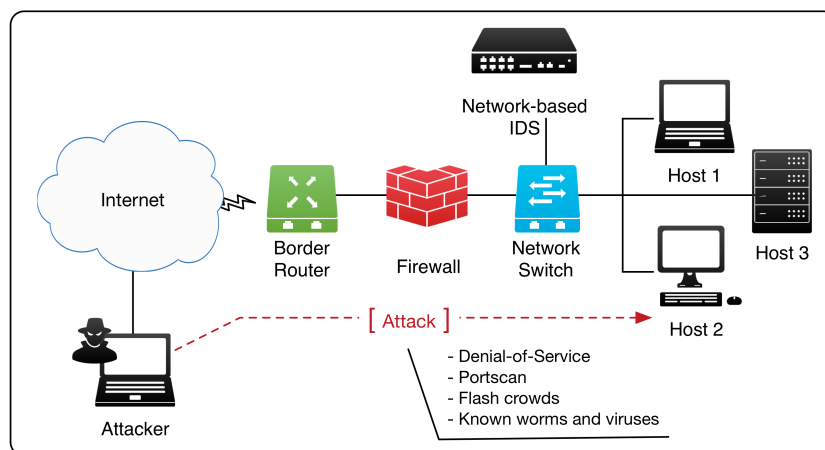


Figure 2.5: Network-based IDS example

Chapter 2. Related Work

others. These systems are classified as agent-based, since they require the installation of software on the host. This IDS evaluates the safety of the host based on operating system log files, access log, and application log, for instance. It is vital because it provides security against the types of attack that the firewall and NIDS do not detect, such as those based on encrypted protocols, since they are located at the destination. Another benefit of HIDS over NIDS is that the success or failure of an attack can be promptly determined [10, 56, 58]. Figure 2.6 illustrates a general host-based IDS.

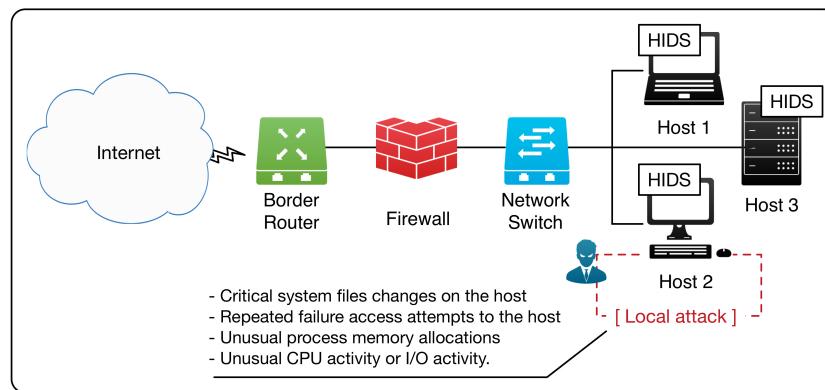


Figure 2.6: Host-based IDS example

2.3.1.3 Hybrid IDS

Hybrid IDSs are developed considering data provided by the host events and the network segments and by combining the functionalities of both network and host-based IDSs [57]. These systems aggregate the benefits of both approaches while overcoming many of the drawbacks. However, hybrid systems may not always mean better systems. Since different IDS technologies analyze traffic and look for intrusive activity in various ways, getting these different technologies to interoperate and coexist in a single system successfully and efficiently is a challenging task.

2.3.2 IDS types by detection technique

2.3.2.1 Signature-based (misuse detection)

Signature-based techniques, also known as knowledge-based or misuse detection, evaluate network activities by using a set of well-known signatures or patterns of attack stored in the IDS database. Whenever an attempt matches a signature, the IDS triggers an alarm. This operation ensures an efficient detection with minimal false alarms, and a good level of accuracy with regard to the identification and classification of abnormalities, making it easier for network administrators to take preventive or corrective measures.

However, as any other action not recognized by the IDS knowledge database is considered normal, unknown anomalies, or little variations in known attacks, cannot be detected. For this reason, signature-based IDSs require constant updating of their knowledge database. Signatures must be defined in order to ensure that all probable variations of an attack are

covered. Additionally, they do not match non-malicious activities, which can be a hard task [21, 28, 59, 60]. Generally, misuse detection techniques work as shown in Figure 2.7.

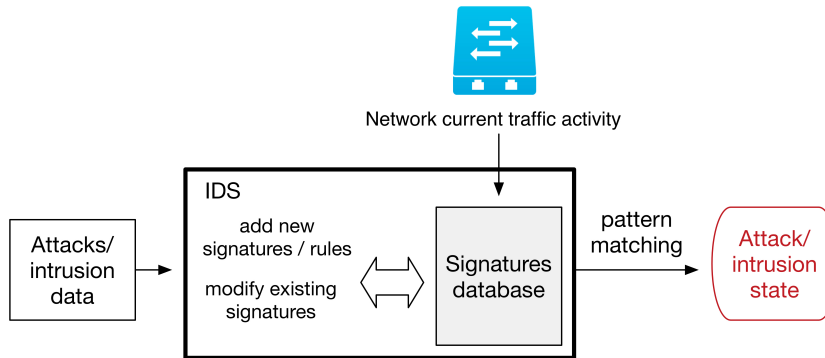


Figure 2.7: Misuse detection (signature-based) techniques general scheme.

2.3.2.2 Anomaly-based (anomaly detection)

Anomaly-based techniques, also known as profile-based or anomaly detection, are founded on the creation of a baseline profile representing normal/expected network behavior, and on that any observed deviation of current activity compared to this profile is considered anomalous. This profile is generated mostly through statistical and historical network traffic data.

A classic example of this type of detection is when a specific user always uses the Internet for a certain period of the day, during business hours. Imagine that this user is a manager at a company being monitored by an anomaly-based IDS. This IDS has spent a whole week creating this user's normal profile, and from the last day of that week, it employed this profile as mandatory for the time allowed to use the Internet. While detection is active, the manager wants to use the Internet during night-time in order to submit a last-minute report, which is something unusual to regular usage. The response of the anomaly-based IDS to this unusual behavior is to restrict Internet access to that user, which would be valid if this was not an exception; however, this would actually be treated as a false positive.

Therefore, the main drawback of profile-based techniques is the possibility of increased false alarm (false positive) rates, because users and system behavior may widely vary. Additionally, attacks may be launched during the learning period and result in a profile containing intrusion behavior, which may not be able to detect some anomalous behaviors. These are false negatives, which is even more serious than false positives. Therefore, constant retraining of the profile is required; however, this may cause the unavailability of the detection system or an increase of false alarms [61]. Finally, depending on the approach, profile creation may demand an extended monitoring period or high computational resource usage.

Anomaly detection techniques are the most commonly used IDS detection type. This is due to their ability of detecting both known and unknown attacks and anomalies, since the detection is performed under the discovery of unusual patterns, which makes this technique more dynamic than the static signature-based technique. It is also helpful in discovering new types of

Chapter 2. Related Work

attack and behavior, and as a knowledge builder for new signatures in misuse detection systems.

Anomaly-based detection is the most popular and well-investigated topic among researchers. There are many different techniques and algorithms, described in the literature, used to build this normal profile and find unusual patterns, such as statistical procedures, machine learning, clustering, fuzzy logic, and heuristics. This has been studied for over 20 years, and there is still a wide investigation panel to be discovered, as well as critical challenges and open issues to overcome, as will be presented later in this survey. Figure 2.8 shows the general structure of an anomaly detection approach.

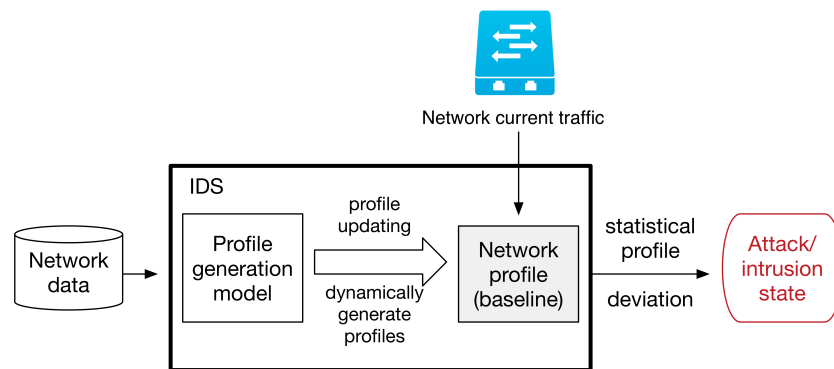


Figure 2.8: General scheme of anomaly detection (anomaly-based) techniques

2.3.2.3 Specification-based

As described in [59] and [62], anomaly detection systems detect the effect of abnormal behavior, while misuse detection systems recognize already known abnormal behavior. Accordingly, specification-based methodologies were created in order to utilize the benefits of both techniques. Therefore, these IDSs manually develop specifications and constraints to characterize normal network behavior. This methodology is accomplished by obtaining the correct operation of a program or protocol and monitoring its execution through the definition of set constraints. Accordingly, this methodology can be more resistant to suitable changes in attacks and allows the discovery of previously unknown attacks while having a very low false positive rate.

On the other hand, specification-based techniques are much more complex since their analysis can be performed in the layers existing below the application layer of the Internet protocol stack, or at the operating system control level. These techniques are restricted to the proper operation of a program, or protocol, and can be excessively tedious and susceptible to errors since they rely on user knowledge. Furthermore, the elaboration of detailed specifications and constraints is costly and time consuming.

This detection model is not as widely distributed as others cited in this thesis, especially because of its greater development complexity and restricting the intended application, since it is aimed, for example, to be a single application.

2.3.2.4 Hybrid techniques

Hybrid IDSs, or Compound detection, implement combinations of misuse, anomaly and specification detection techniques. These systems can be based on the normal network profile and also attack behavior, for instance.

An example of a hybrid IDS has been proposed by Assis *et al.* [63], in which a network profile called digital signature of network segment using flow analysis (DSNSF) was created to detect unknown anomalies within network traffic. Then, pre-loaded signatures classified the discovered anomalous behavior as a DoS, DDoS, flash crowd or port scan attack. Another example has been presented by Stakhanova *et al.* [68], who combined specification-based techniques with anomaly-based ones in an effort to mitigate the limitations of the former. The need of user expertise is overcome by an approach for the automatic generation of normal and abnormal behavioral specifications as variable-length patterns, which are classified via anomaly-based machine learning techniques.

2.4 Anomaly detection techniques, methods and systems

In this survey chapter, we focus on anomaly detection (anomaly-based IDS); the following chapters contain a review of its most current techniques, methods, and systems. Figure 2.9 illustrates the topics. However, since there are many emerging types of research proposing hybrid approaches, the combination of both misuse and anomaly detections, for instance, may be addressed as well.

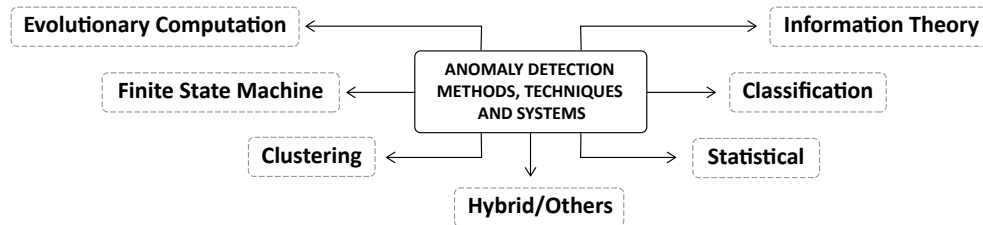


Figure 2.9: Anomaly detection methods, techniques and systems analyzed in this research

2.4.1 Statistical methods

Statistical methods for anomaly detection are widely used and are commonly based on probabilistic models associated with training data for the purpose of tracking network behavior. Anomalies are related to sudden changes in network data. Mostly, these abrupt changes are detected by modeling hard thresholds. The primary challenge for statistical techniques is to find methods reducing false alarm generation caused by hard thresholds [18]. For instance, statistical signal processing procedures may be used to increase the detection rate while decreasing false alarms, as Lakhina *et al.* did in their work with principal component analysis [64, 7, 65].

Chapter 2. Related Work

2.4.1.1 Wavelet analysis

Wavelet analysis focuses on modeling non-stationary data series'. Such data series may contain signals that can vary in both amplitude and frequency over extended periods of time. Unlike Fourier analysis, which uses trigonometric polynomials, data series are modeled using wavelets, which are powerful basis functions localized in time and frequency, allowing a close connection between the series being represented and their coefficients. In this manner, wavelet analysis is fundamentally a way to describe levels of detail with regard to particular data, which can be images, curves, surfaces, and so on.

Callegari *et al.* [66] propose a real-time anomaly detection method using wavelets combined with sketches. It is a router level analysis performed by extracting NetFlow traces and transforming them into ASCII data files. After formatting, sketches are used to aggregate different traffic flows in sketch tables through hash functions. Next, the time series are submitted to a wavelet transform for the purpose of discovering discontinuities.

Another study using wavelets was produced by Hamdi and Boudriga [67]. It relied on identifying attack-related anomalies by differentiating between dangerous and non-threatening anomalies. This task was achieved based on the concept of period observation, where wavelet theory was used to decompose one-dimensional signals in order to analyze both their special frequencies and time localization.

2.4.1.2 Principal component analysis

Principal component analysis (PCA) is a widely used statistical technique for anomaly detection in computer networks. It is defined as a dimensionality reduction approach, in which a data set consisting of n correlated variables can be mapped onto a new and reduced set of k variables, the principal components (PCs), where $k \ll n$. These PCs are a set of orthonormal vectors, which define a k -subspace, and are uncorrelated and arranged so that the first components retain most of the variation present in all original variables [68, 69].

Lakhina *et al.* [64], who pioneered this field, addressed the anomaly diagnosis problem in network wide-traffic by using PCA to efficiently separate traffic measurements into normal and anomalous subspaces. The main idea was that PCA results in a reduced set of k variables (principal components or k -subspace) which corresponds to normal network traffic behavior, while the remaining subspace of m components ($m = n - k$) consist of anomalies or noise. Then, every new traffic measurement is projected onto both subspaces so that different thresholds can be set to classify these measurements as normal or anomalous. Their work was responsible for the massive attention on PCA-based approaches for anomaly detection received in the last decade. However, although it was a notorious work with good results and advances in the area, it received some criticism from various authors, mainly related to the calibration sensitivity of PCA, as reported in Ringberg *et al.* [70].

Ringberg *et al.* [70] and others [71, 72] have criticized the studies of Lakhina *et al.* [64, 65] on PCA by outlining four main challenges regarding its sensitivity: (i) false positive rates are affected by small noises in the normal subspace; (ii) the level of traffic aggregation can mitigate the value of PCA; (iii) large anomalies can infect the normal subspace; (iv) no mapping amongst the reduced subspace PCA produced and the original spatial source of the anomaly.

In this manner, the anomaly detection method proposed by Pascoal *et al.* [73] used a robust PCA detector merged with a robust feature selection algorithm in order to obtain adaptability to distinct network contexts and circumstances. Additionally, this robust PCA approach does not require perfect ground-truth for training, which is one of the limitations of standard PCA discussed in [70]. In [74], the authors propose ADMIRE, which is a combination of three-step sketches and entropy-based PCA, and results in better true and false positive rates while being capable of capturing distinct kinds of anomalies due to the different entropy time series for PCA. Furthermore, O'Reilly *et al.* [75] surpassed those limitations in finding the PCs from a dataset with anomalies by proposing a Minimum Volume Elliptical PCA (MVE-PCA) method, consisting of the solution to a convex optimization problem by creating a soft-margin minimum volume ellipse around the training dataset, which decreases the effect of anomalies existing in the data.

Nevertheless, Camacho *et al.* [76] actively maintain that neither the original PCA proposal nor critical researchers could effectively surpass the disadvantages of using PCA for anomaly detection. To overcome these drawbacks, the authors used a PCA-based multivariate statistical process control (MSPC) approach, which monitors both the Q-statistic and D-statistic. Thereby, it was possible to establish control limits in order to detect anomalies, when they became consistently exceeded. Additionally, the MSPC approach has contribution plots used for finding the root cause of the anomaly. Data pre-processing relies on the feature-as-a-counter approach in which variables are counters for the number of times some event is logged throughout a given time interval. This is in contrast to the idea of Lakhina *et al.* [64], which considers counters as simple quantitative variables.

Fernandes *et al.* [15, 17] proposed PCADS-AD, an autonomous profile-based anomaly detection system based on a dimensionality reduction procedure and principal component analysis (PCA). It is an enhanced version of their initial work presented in [78]. The system was divided into two main stages. First, the authors used a different interpretation of PCA to generate a network profile called Digital Signature of Network Segment using Flow analysis (DSNSF). The system analyzed historical network traffic data over a period of days, identifying among them the most significant traffic time intervals while reducing the data set so that the new reduced set could efficiently characterize normal network behavior. Then, the DSNSF was used as a threshold to detect volume anomalies by restricting an interval, where deviations were considered normal, through some PCA parameters. This system used three IP flow features (bits/s, packets/s and flows/s) to predict network normal behavior and generate the DSNSF. Another four flow attributes (origin and destination IP addresses and TCP/UDP Ports) were used to produce a report containing useful information concerning the abnormal traffic interval; thus, the network administrator was assisted in taking fast measures to resolve the identified problem. The drawback of this approach is the usage of only volume attributes for anomaly detection, which only considers the detection of volume-based attacks. In this manner, the system is unable to detect attacks which do not impact on bits, packets, and flows.

2.4.1.3 Covariance matrix

Covariance matrices are second-order statistics and have been proven to be a powerful anomaly detection method. An interesting direction in this area is finding which variables best

Chapter 2. Related Work

label network anomalies and improve detection performance.

The work presented in [79] employs covariance matrix analysis to detect flooding attacks. This approach models network traffic as covariance-matrix samples in order to make use of statistical assets contained in the temporally sequential samples for the purpose of detecting flooding attacks. Then, it directly uses changes of covariance matrices and differences of correlation features to reveal the alterations between normal traffic and various types of flooding attacks.

Miao Xie [80] performed anomaly detection in a segment-based manner by handling a collection of neighboring data segments, with the aid of random variables, and exploiting their spatial predictabilities to determine which ones behaved abnormally. This approach used a sample covariance matrix approximated per the concepts of Spearman's rank correlation coefficient and differential compression in order to substantially reduce the computational cost.

Huang *et al.* [81] supported the use of covariance matrix for dimensionality reduction instead of traditional PCA discussed in the previous section. They pointed out that a static choice of k principal components is poor at capturing real-time changes, in addition to only allowing weak heuristics due to sensitivity to small variations in the dimensions representing the normal subspace. Therefore, to overcome the limitations of variance-based approaches, the authors came up with a distance-based dimensionality reduction approach for anomaly detection. Depending on their types, anomalies manage to cause distinct types of deviances in the covariance matrix of observed traffic. These deviances allow the categorization of detected anomalies and immediate decision-making with regard to mitigation actions. Their proposal was also able to adapt to changing patterns in the test data such that the model would only use a few important dimensions at any time.

2.4.1.4 Others

This section presents other noteworthy statistical methods, which do not fit into the previous ones since they combine different statistical techniques.

The study by Kalkan and Alagöz [82] used traffic filtering as a way to prevent network attacks and especially DDoS attacks. ScoreForCore was classified as a statistical filtering model based on reaction time and collaboration, which selects the most suitable features from the attack related traffic. The model calculates a score for each packet using the nominal and current profile; then, it compares them in order to find the two features deviating the most from the nominal profile by using collaboration between routers and thresholds. Ozkan *et al.* [83] studied the anomaly detection problem for fast streaming temporal data, in an online setting, and proposed an efficient statistical online algorithm fusing Markov statistics with Neyman-Pearson (NP) characterization. Their proposal successively learns the feasible varying nominal Markov statistics in a time series and detects anomalous subsequences by first assigning scores to each fixed length subsequence using pair-wise distances and then considering the magnitude of the anomaly score and providing Neyman-Pearson characterization.

Network traffic is currently composed of cycles consisting of bursts with specific characteristics directly affected by working days and user access periods. Under this assumption,

Proença *et al.* [84] introduced the Digital Signature of Network Segment (DSNS), which is a set of information capable of defining the traffic profile. It automates the task of monitoring network segments by statistically estimating the traffic behavior based on historical traffic data. The algorithm is called BLGBA and is based on a variation of the statistical measure mode. After extracting SNMP traffic samples from the MIB, the DSNS is built second by second through the analysis of a prior period. The calculation distributes the elements in frequencies according to differences between the size of each sample. Then, the authors validated the DSNS through visual analysis, Bland-Altman plots, residual analysis, linear regression, and the Hurst parameter.

A correlational paraconsistent machine (CPM) has been proposed by Pena *et al.* [85] and relies on two unsupervised traffic characterization methods and non-classical paraconsistent logic (PL). The authors used both ant colony optimization for digital signature (ACODS) and auto regressive integrated moving average (ARIMA) [86] methods in order to analyze historical network traffic data and generate two distinct network profiles able to describe normal traffic behavior. These profiles are called digital signature of network segment using flow analysis (DSNSF) and is derived from the work proposed in [84]. The existence of anomalies is related to degrees of certainties and contradictions produced by paraconsistent logic over a correlation between two prediction profiles and associated real traffic measurements. From the Euclidian distance calculation between the two DSNSFs and the evaluation of paraconsistent logic signals, the model obtains real evidence for the proposition P ($P \rightarrow$ “interval contains an anomaly”) to be true.

Another statistical traffic characterization approach for anomaly detection by creating the DSNSF network profile is proposed by Assis *et al.* [63]. It is a seven-dimensional profile-based anomaly detection system based on the Holt-Winters forecasting technique. The IP flow traces bits/s, packets/s, flows/s, origin and destination IP addresses, and Ports, are simultaneously analyzed in every one-minute time window; therefore, the system can identify different anomalies and generate alarms. The normal network profile of how the network should behave in the next day is predicted dynamically by using the current traffic of the day and the previous day's generated profile. Authors use thresholds to indicate the interval between real traffic and the profile considered as normal. These thresholds are calculated in an asymmetric way, using the profile, a scaling factor for its width, and a deviation measure. The intervals with mostly greater errors are updated with the absolute deviation of the interval while the opposite confidence band is updated with the standard deviation of the profile. Finally, the alarm system is capable of detecting anomalies in two ways: (i) by alerts, which are related to anomalous behaviors not existing in the system anomaly database; and (ii) by alarms, generated when the system knows the anomalous behavior signature.

Bang *et al.* [87] propose an IDS using a hidden semi-Markov model (HsMM) aimed specifically at the detection of advanced LTE signaling attacks on WSNs. According to the authors, traditional hidden Markov Models (HMM) cannot represent many possible transition behaviors; therefore, HsMM overcomes this limitation since it has arbitral state sojourn time and is more suitable to time-series behavior analysis. They used the HsMM to effectively model the spatial-temporal characteristic of the wake-up packet generation process, taking the process log-likelihood as the test basis of normality. Then, their detector compared observed spatiotemporal features of a server's wake-up packet generation, with the normal criteria established by the HsMM. Therefore, an alarm is set off whenever significant divergence occurs.

Chapter 2. Related Work

Although classical Markov chain techniques are widely accepted in anomaly detection applications, their short memory property may ignore interactions among the data. On the contrary, the long memory property of a higher order Markov model clouds the relationship between previous and current test data and, thus, it reduces reliability. In light of this, Ren *et al.* [88] defended that once Markov models are established in the training phase, their order is fixed to detect anomalies in the testing phase. However, the fixed Markov models (n-order) force each state of a sequence to be conditioned on previous n states and may not be enough to provide an accurate estimate of the detecting state. Thus, the authors proposed a dynamic Markov model to balance the length of the memory property of Markov models and keep the strong correlation between memory (or the Markov model) and current test data. To achieve this, the proposed approaches repeatedly calculate the Pearson correlation in order to find the proper order of the Markov model in a sliding window, where the sequential data is segmented. To keep detection continuous, a substitution strategy of anomalies was reported to protect the building of models from the infection of detected anomalies.

Jazi *et al.* [89] explored several types of application-layer DoS attacks and proposed a detection approach based on a nonparametric CUSUM algorithm. The proposed approach relies on a selected combination of application and network-level attributes for anomaly detection. According to the authors, the resulting method was evaluated on various types of attacks on modern web servers since they represent the most common target for DoS attacks. In addition, the study investigated the performance fluctuation in the presence of thirteen different sampling methods and explored the impact of sampling on the detection of application level DoS attacks. The results confirmed that even specialized sampling techniques could introduce some distortion in detection quality. In this manner, detection should be tied to the sampling technique in order to compensate for distortions provided by sampling and to ensure the improved assessment of traffic characteristics.

2.4.1.5 Summary

In summary, statistical approaches include the following advantages.

- Intrinsic capability to detect network anomalies than any other method,
- Ability to learn the expected behavior of the traffic (network system),
- Traffic analysis is based on the theory of sudden changes, which sets an alarm whenever a significant deviation happens.
- The methods do not require any kind of prior knowledge about the system as an input.

However, there are some relevant drawbacks that must be considered.

- Some kinds of attacks may be a regular part of the training dataset and may be incorporated in the normal behavior, causing it to be considered as normal.
- It requires some relevant time to train the models in order to be able to set the first alarm.
- The use of thresholds may not be reliable in some real-world cases due to its limited and static nature.

Table 2.5 summarizes the characteristics of discussed statistical approaches, regarding techniques, data precedence, investigated anomalies, and validation metrics used to test

detection performance.

2.4.2 Clustering methods

Clustering analysis aims to group a set of objects into classes of similar objects. These classes, or groups, called a cluster, and its objects, are similar (in one way or another) to each other and dissimilar to those in other clusters. Clustering-based processes are adaptable to changes and help single out useful features distinguishing different groups. Clustering techniques can be used for outlier detection, identifying values, which are too “far away” from any cluster, or as a preprocessing step for other algorithms/approaches. Additionally, classification is an effective resource for distinguishing groups or classes of objects; however, it requires the often costly collection and labeling of a large set of training tuples or patterns, which the classifier uses to model each group [90].

Rajasegarar [91] presented a distributed hyperspherical cluster based algorithm for anomaly detection in wireless sensor networks. Clustering was used to model the traffic data at each node by classifying data vectors as either normal or anomalous. Anomalous clusters were identified by using the average inter-cluster distance of the k nearest neighbor (KNN) clusters. This works under a distributed scheme, where sensor nodes report on cluster summaries, which are merged by intermediate nodes before communicating with other nodes and, thus, minimize communication overhead.

Mazel *et al.* [92] introduce a non-supervised approach to detecting and characterizing network anomalies. This approach initially works by using a clustering technique, combining sub-space clustering with evidence accumulation clustering and inter-clustering results association in order to blindly identify anomalies in traffic flows.

K-means is a popular clustering technique in the anomaly detection field and is able to classify data into distinct categories. However, it has drawbacks such as local convergence and sensitivity to the selection of cluster centroids. Therefore, many researchers try to combine k-means with other techniques in order to overcome these shortcomings. Karami and Guerrero-Zapata [93] introduced a fuzzy anomaly detection system based on the hybridization of particle swarm optimization (PSO) and k-means with local optimization in order to determine the optimal number of clusters. It is divided into two phases: the training phase aims to find the near optimal solution by combining a novel boundary handling approach of PSO’s global search with the fast convergence of k-means; thus, it avoids being trapped in a locally optimal solution. The fuzzy approach is used in the detection phase, in which false positive rates are reduced with a reliable detection of intrusive activities. This is due to any data (normal or attack), which may be at close distance to some clusters.

Carvalho *et al.* [94] developed a proactive network monitoring system that can detect unusual events and reduce manual intervention and error probability in decision-making. Their proposal consists of creating a network profile called DSNSF (digital signature of network segment using flow analysis), which describes normal network usage using a clustering approach through the modification of the ant colony optimization (ACO) metaheuristic, called ACODS. ACODS characterizes network traffic discovery in the large volume of high-dimensional input

Chapter 2. Related Work

Table 2.5: Comparison of statistical anomaly detection approaches

Paper	Year	Tech. ^a	Anomaly Type ^b	Dataset ^b	Source ^c	Validation Metrics
Hamdi and Boudriga [67]	2007	Wavelet	DoS/DDoS (S)	Real network from MIT (R)	IP flow	Packet count
Callegari <i>et al.</i> [66]	2011	Wavelet	Generic synthetically added anomalies in the data (S)	Abilene/Internet2 Network (R)	NetFlow	Detection Rate
Lakhina <i>et al.</i> [64]	2004	PCA	Synthetic injection of large and small anomalies (S)	Sprint-1 and Abilene backbones (R)	NetFlow	Detection Rate, FPR, Identification Rate and mean absolute relative error
Pascoal <i>et al.</i> [73]	2012	PCA	Portscans and snapshots (R)	Small private laboratory network scenario (R)	-	Recall, FPR and Precision
Kanda <i>et al.</i> [74]	2013	PCA	22 attack categories (TCP SYN flood, port scan, etc.) (R)	Backbone link from the MAWI traffic repository (R)	IP flow	TPR, FPR, Accuracy, F-measure, ROC and Euclidean distance
Fernandes <i>et al.</i> [15]	2015	PCA	DoS, DDoS and Flash Crowd (S)	University network (R) and simulated anomalies (S)	NetFlow	NMSE, Correlation Coefficient, TPR, FPR, ROC
Camacho <i>et al.</i> [76]	2016	PCA	DoS and other general network faults/anomalies (R)	VAST 2012 2nd mini challenge (R) and a controlled scenario (R)	Firewall and IDS logs, NetFlow	TPR, TNR, FPR, FNR, Recall, Specificity, Accuracy
O'Reilly <i>et al.</i> [75]	2016	PCA	Generic (S) (R)	2-dimensional synthetic Gaussian data (S), UCI Machine Learning Repository (R)	-	Area Under the ROC Curve (AUC), FPR, TPR, ROC,
Yeung <i>et al.</i> [79]	2007	Cov. Matrix	flooding attacks (DDoS) (R)	KDDCUP 99 (R)	TCP dump	Detection Rate, FPR
Xie <i>et al.</i> [80]	2015	Cov. Matrix	Generic (constant, burst, small noise and large noise anomalies) (R) and artificially injected (S)	IBRL network (R)	-	ROC, average saving rate (ASR)
Huang <i>et al.</i> [81]	2016	Cov. Matrix	Generic labeled anomalies (R)	Kyoto2006+ dataset (R)	-	FPR, ROC
Proença <i>et al.</i> [84]	2004	Statistical mode	Generic outliers (R)	University network (R)	SNMP	Hurst parameter, residual analysis, linear regression, bland-altman plot
Assis <i>et al.</i> [63]	2014	Holt-Winters	DoS, DDoS, Flash Crowd, portscan (S)	University network (R) and simulated anomalies (S)	NetFlow	Accuracy, TPR, FPR, ROC
Pena <i>et al.</i> [85]	2014	ARIMA, Paraconsistent logic	Generic (R)	University network (R) and simulated anomalies (S)	NetFlow	Real evidence level, TPR, FPR, ROC
Kalkan and Alagöz [82]	2016	Filtering model, reaction time	Different types of DDoS (S)	MAWI Working Group Traffic Archive (R) and simulated environment (S)	-	Precision, recall, TNR, Negative predictive value (NPV), f-measure, f-measure complement, accuracy and attack prevention efficiency (APE)
Ozkan <i>et al.</i> [89]	2016	Markov statistics, Neyman-Pearson	Sudden change in the source statistics (S)	Monte Carlo simulations (S)	-	ROC, FPR
Bang <i>et al.</i> [87]	2017	Hidden semi-Markov model	Advanced LTE signaling attack types (S)	Simulated environment (S)	-	FPR, FNR, TNR
Ren <i>et al.</i> [88]	2017	Dynamic Markov model	Generic outliers (R) (S)	Synthetic dataset (S) and Shanghai airport traffic, UCR archives	-	TPR, FPR
Jazi <i>et al.</i> [89]	2017	CUSUM	Application layer DoS (S)	ISCX dataset, academic network traces (R)	-	Detection rate, FPR

^aStatistical Techniques/Methods used.

^bData precedence; R=Real, S=Simulated.

^cSource types in blank are either not clearly specified by the authors or not relevant in their research.

data in a cluster set, and by optimizing the extraction of behavioral patterns through an unsupervised learning mechanism. Then, to detect anomalous behavior, authors use the pattern matching technique called dynamic time warping (DTW). They first compute the similarity between real traffic and normal profile in each time interval; then, compute the distance between the series and provide a measure based on both form and distance. The proposed alarm system works with seven flow attributes, using entropy to summarize information regarding IP addresses and Port features. When an anomaly is detected, ACODS provides a full report containing IP flow information indicating the impact of each attribute on the detected anomalous time interval. ACODS has a square complexity, resulting in a solution convergence by many iterations, in which authors try to mitigate by using local search and pheromone updating.

In, Dromard *et al.* [95] proposed ORUNADA, an unsupervised anomaly detector based on the incremental grid clustering algorithm called IDGCA and a discrete time sliding window. Incremental grid clustering is more efficient than usual clustering algorithms since they later only update the previous feature space partition, instead of repartitioning the whole space whenever few points are added or removed. Then, the system merges these updated partitions in an effort to recognize the most dissimilar outliers. Incremental grid clustering usage contributes to lowering system complexity, which makes it more feasible for real-time detection.

Regarding SDNs and their challenges, like high density and variety of hosts, He *et al.* [96] recently developed a two-stage unsupervised clustering algorithm for anomaly detection. The first stage is a feature selection procedure used to remove unnecessary features in the dataset. Its basis is the calculation of a maximal information coefficient (MIC), which describes the relationship between two continuous features, and relevancy, which is a symmetric uncertainty estimator for discrete features. After selecting relevant features, a density peak-based clustering algorithm classifies the reduced dataset into normal and misbehaved patterns. Their experimental results proved that when a typical SDN hierarchy of controllers is used, the traffic data can be locally analyzed in each controller. This lessens the volume of traffic shuffled across the network.

Some of the main limitations of anomaly detection methods are basically: the absence of labeled data; finding of new unknown anomaly patterns; noisy data; and high false alarm rates. As an effort to overcome these problems, Bigdeli *et al.* [97] proposed an incremental two-layer cluster based structure for anomaly detection. The core idea is to cluster network data and represent these clusters as a Gaussian Mixture Model, so the model can categorize new instances and also detect and ignore redundant ones. Moreover, the high false alarm rate issue was addressed by a collective labeling method, which labels new inward instances in both collective and incremental ways.

2.4.2.1 Summary

With regard to clustering-based approaches, their advantages are listed as the following.

- Incremental clustering has a fast response generation.
- Stable performance when comparing to statistical methods or classifiers.
- Reduce computational complexity due to the ability to group large datasets into small ones.

Chapter 2. Related Work

However, limitations of these techniques can be seen below.

- They are highly dependent on proximity measures, and each one can affect the detection rate in a positive or negative way.
- Time consuming.
- They are not optimized for anomaly detection.
- Sometimes, the algorithms can be trapped in the local minima.

At last, Table 2.6 summarizes some characteristics of the discussed clustering approaches with regard to data precedence, investigated anomalies, and validation metrics used to test detection performance.

Table 2.6: Comparison of clustering anomaly detection approaches

Paper	Year	Tech. ^a	Anomaly Type ^b	Dataset ^b	Source ^c	Validation Metrics
Rajasegarar <i>et al.</i> [91]	2014	k-NN clusters	Randomly generated set of anomalous data (S)	IBRL and GDI (R) and Banana and Gaussmix datasets (S)	-	ROC, Detection rate (DR) and False positive rate (FPR)
Mazel <i>et al.</i> [92]	2011	Sub-Space clustering, Evidence Accumulation and Inter-Clustering	Few ICMP pkts, network scan (R)	real traffic trace from the public MAWI repository of the WIDE project (R)	IP flow	Cluster similarity graph and outlier similarity graph for destination aggregated data
Karami and Guerrero-Zapata [93]	2015	K-Means, PSO	Abnormal Source Behavior, flooding attack (R)	UCI machine learning repository / CCNx data repository of Univ. of Politecnica Catalunya (R)	IP flow	Detection Rate (DR - Recall), FPR, Precision, F-measure
Carvalho <i>et al.</i> [94]	2016	ACO (modified for clustering), DTW	DoS, DDoS, port scan, flash crowd (S)	University network (R) and simulated anomalies (S)	NetFlow	NMSE, accuracy, TPR, FPR, ROC curve
Dromard <i>et al.</i> [95]	2017	Incremental grid clustering algorithm (IGDCA)	RST and SYN attacks, and generic outliers (R)	ONTS dataset and MAWILab network traces (R)	IP flow	TPR, FPR
He <i>et al.</i> [96]	2017	Density peak based clustering algorithm	DoS, Probe, R2L, U2R (R)	KDDcup99 (R)	TCP dump	Classification accuracy
Bigdeli <i>et al.</i> [97]	2018	Spectral-based and density-based clustering	DoS, Probe, R2L, U2R (R/S)	KDDCUP99, Darpa98, NSLKDD, DataSetMe, and IUSTSip (R/S)	TCP dump, IP flow	ROC curve

^aClustering Techniques/Methods used.

^bData precedence; R=Real, S=Simulated.

^cSource types in blank are either not clearly specified by the authors or not relevant in their research.

2.4.3 Finite state machine methods

A Finite State Machine (FSM), also called finite automata, is a mathematical behavioral model composed of states, transitions, and actions, used to represent computer problems or logical circuits. Each state stores information about the past, which are changes that have occurred since the entry into a state from the start of the system to the present time. This type of machine can only be in one state at a time. A transition indicates a state change and is disclosed by a condition, which must be achieved for the transition to occur. An action is a description of an activity, which must be carried out at a particular time. Moreover, these machines have strong analytical techniques, given that one can explore every possible sequence of states, since their alphabet of input and output allows representing a wide variety of situations.

Estevez-Tapiador *et al.* [98] presented a protocol anomaly detector using a finite state machine (FSM) approach, where network protocols were modeled from state sequences and transitions through a Markov chain. Its main idea was to monitor a given protocol in order to find deviations from “normal” usage. If the conditions are complete enough, the model can detect illegitimate behavioral patterns successfully.

Su [99] employed finite state machines to implement a framework applying frequent episode rules for a network intrusion prevention system (NIPS). The presented NIPS was developed to explore Probe attacks and anomalies that are difficult to be effectively detected by firewalls and anti-virus software. At first, it works by mining log files, which are posteriorly refined, resulting in episode rules that are converted to build an FSM. Via the FSM, every connection on a particular port is monitored and mapped out. Once a default alarm condition is achieved, the integrated real-time firewall update tool disconnects the malicious connection.

In [100], the authors produced an engineering method of gathering only a small volume of relevant IP flow records and aggregated them into a state space representation. This aggregation served as input to a finite state machine scheme. They developed an FSM with a stream learning component, such that it would be feasible to start modeling and learning a fine-grained communication profile in real-time. Their system produced promising detection rates over botnet malware detection. Additionally, they concluded that it is worthwhile to use limited IP flow data rather than large datasets for training.

2.4.3.1 Summary

Finite state machine techniques are not as popular as statistical or classification techniques, however, they have some good points to consider.

- Robustness and flexibility
- Strong analytical techniques, since their alphabet of input and output, allows representing a wide variety of situations
- High detection rate whether there is a considerable knowledge base regarding attacks and normal cases.

Some disadvantages of these techniques are listed below.

- Time-consuming.
- Inability to detect rare or indefinite attacks.

Chapter 2. Related Work

- Dynamic updating of rules/conditions are costly.

Table 2.7 summarizes some characteristics of the discussed clustering approaches, regarding data precedence, investigated anomalies, and validation metrics used to test detection performance.

Table 2.7: Comparison of finite state machine anomaly detection approaches.

Paper	Year	Anomaly Type ^a	Dataset ^a	Source ^c	Validation Metrics
Estevez-Tapiador <i>et al.</i> [98]	2003	Protocol misusages (R)	TCP traffic filtered by destination port (R)	-	-
Su <i>et al.</i> [99]	2010	DoS, worm (R)	SMB with NetBIOS Session service (R)	-	-
Hammerschmidt <i>et al.</i> [100]	2016	Botnet malware (R)	Publicly available dataset of manually labeled IP flow traces (R)	-	TP, FP, Precision

^aData precedence; R=Real, S=Simulated.

^cSource types in blank are either not clearly specified by the authors or not relevant in their research.

2.4.4 Classification-based methods

Classification [101] is widely used in the anomaly detection field. The main idea of such techniques applied to this area can be summarized as two steps. First, during the training phase, a classifier is built (learned) using labeled training data. Then, this classifier is used to classify an instance as normal or anomalous (testing phase). According to each available labeled data for training, classification-based anomaly detection techniques can be either multi-class or one-class. The latter occurs when all training data have only one normal class label. The first assumes that training instances have multiple normal class labels. In this case, a classifier is built to be able to distinguish instances among normal classes and those who do not belong to any class (anomaly).

2.4.4.1 Naïve Bayesian

Naïve Bayesian is a simple probabilistic classifier commonly used for network intrusion detection problems. It combines prior information with sample information and implements it in statistical deduction, which uses probability to show all forms of uncertainty. Its principles are founded on the assumption that all input attributes are conditionally independent to each other. Thus, it calculates the probability of a certain instance belonging to a singular class.

Klassen and Ning [102] proposed a Naïve Bayesian approach to detect black holes, selective forwarding and DDoS attacks, in real time. The system monitored packets sent from nodes; therefore, their behavior is checked in order to detect any abnormality. The classifier assumes that data are normally distributed; then, the probability of a sample belonging to a class is calculated by a normal distribution probability procedure. Tao *et al.* [103] also used a Naïve Bayesian approach; however, they combined it with a time slicing function and, thus, they exploited the relationship between time and network traffic, since network traffic changes at distinct times and some traffic does not occur at a particular time. The work of Swarnkar

and Hubballi [104] accurately detected suspicious payload content in network packets through the use of the one class Naive Bayes classifier for payload based anomaly detection (OCPAD), a combination of frequency information of short sequences with a one class multinomial naïve Bayes classifier.

2.4.4.2 Support vector machines

Another classification method is Support Vector Machine (SVM) [111], which is also used in pattern recognition. SVMs are a supervised learning concept characterized by the use of feature vectors/kernels (such as radial basis function - RBF), the nonexistence of local minima, sparseness of the solution, and capacity check achieved by operating on the border (the distance of the solution hyperplane to its closest point). Classifiers are obtained with good generalization, which is defined as its ability to correctly predict the class of new data from the same domain in which learning occurs.

Catania *et al.* [105] proposed a novel approach to providing autonomous labeling to normal traffic, in order to overcome imbalanced class distribution situations and reduce the presence of attacks in the traffic data used for training an SVM classifier. Amer *et al.* [106] applied two modifications of the unsupervised one-class SVM: Robust one-class SVMs and Eta one-class SVMs. Their goal was to make the decision boundary less sensitive to outliers in the data.

Erfani *et al.* [107] stated that problem domains with a high number of dimensions are an obstacle to anomaly detection since irrelevant features can cover the presence of anomalies. Additionally, although the use of SVMs in detecting anomalies is effective on small datasets with many features, in complex high-dimensional data, the method is likely to take a long time for training. To overcome this limitation, the authors combined an unsupervised deep belief network (DBN) with one-class SVMs. The unsupervised DBN is trained to extract the features that are less sensitive to irrelevant deviations in the input data, producing a new data set suitable for being used to train a one-class SVM.

Additionally, Wang *et al.* [108] created an effective IDS based on a SVM with augmented features. Their framework integrates the SVM with the logarithm marginal density ratios transformation (LMDRT), a feature transduction technique that transforms the dataset into a new one. The new and concise dataset is used to train the SVM classifier, improving its detection. By evaluating the framework using the mostly used NSL-KDD dataset, the authors could achieve a fast training speed, high accuracy and detection rates, as well as low false alarm presences.

Kabir *et al.* [109] proposed an IDS based on a modification of the standard SVM classifier, known as the least square support vector machine (LS-SVM). This alteration is sensitive to outliers and noise in the training dataset when compared to a regular SVM. Their decision-making process is divided into two stages. The first stage is responsible for reducing the dataset dimension by selecting samples depending on the variability of data by using an optimum allocation scheme. Then, the next stage uses these representative samples as the input of the LS-SVM. The algorithm was optimized to work on both static and incremental data and produced effective results.

Chapter 2. Related Work

2.4.4.3 Artificial neural networks

Artificial Neural Networks (ANNs) are computational techniques that present a mathematical model inspired by the neural structure of intelligent organisms, which acquire knowledge through experience. They are self-adapting, self-organizing and able to learn according to inputs and feedback from the ecosystem within which they operate. Although neural networks are considered a bio-inspired model, they are used in the anomaly detection domain mostly as classifiers. Multi-layer Perceptron (MLP) and Back Propagation (BP) algorithms are the most common ANN techniques.

Subba *et al.* [110] employed an ANN model in order to introduce an intelligent agent for classifying whether the underlying patterns of audit records are normal or abnormal while classifying them into new and unseen records. This goal is accomplished through feed forward and back propagation (BP) algorithms. They are responsible for feeding the neural network with inputs processed to become vectors, comparing the calculated and expected output generated by the ANN, and at finally, altering the weights of the ANN connections in order to approximate the output. After some experiments, this approach proved to be high in performance and low in terms of computational overhead.

Saeed *et al.* [111] proposed a two-level anomaly-based IDS using a Random Neural Network (RNN) model in an IoT environment. The RNN model was employed in order to build a behavior profile based on both valid and invalid system input parameters to distinguish normal and abnormal patterns. The system learns the relationship between input and output by adjusting the interconnection weights of the RNN. The second level of the IDS is responsible for detecting a broad range of Illegal Memory Access (IMA) bugs and data integrity attacks.

Brown *et al.* [112] proposed a two-class classifier using an evolutionary general regression neural network (E-GRNN) for intrusion detection based on the features of application layer protocols such as HTTP, FTP, and SMTP. Authors used evolutionary computation to evolve parameters and salient features (feature mask) from the general regression neural network and to find its optimal configuration. This method reduces computational complexity by eliminating unnecessary features and increases classification accuracy.

Supervised learning models can train a classifier by only using labeled samples, which are difficult to obtain due to requiring expert knowledge. On the other hand, unsupervised approaches consider only unlabeled samples, which are easily available in real-world situations. Ashfaq *et al.* [113] proposed a fuzziness-based semi-supervised learning approach, merging both unlabeled and labeled data to build a better classifier. The base classifier was the neural network with random weights (NNRW) due to its excellent learning feature. For all unlabeled samples produced by the NNRW, their model computes fuzziness as an effort to discover relationships between the output fuzzy membership vectors and misclassification rates. Subsequently, the unlabeled samples receive a predicted label according to fuzziness groups (high, mid and low), and the classifier is retrained with them. The authors found out that samples within low and high fuzziness groups are vital in improving the performance of the NNRW classifier and result in high accuracy rates. Additionally, samples belonging to mid-fuzziness groups showed increased uncertainty of misclassification.

2.4.4.4 Ensemble approach

An ensemble approach means combining responses of multiple classifiers into a single one, thus yielding better performance compared to using individual classifiers. The weighted-majority algorithm (WMA) is a well-known ensemble technique responsible for combining and selecting the best response among all classifiers [114].

Aburomman and Reaz [115] cite that an ensemble classifier achieves success conditional to the diversity in the outcomes of its component classifiers and the method chosen to combine these outcomes into a single one. In this manner, they first trained six SVM experts (in which an expert consists of five binary classifiers producing a binary vector of outcomes) and six other experts using the k-nearest neighbor (k-NN). Then, they used particle swarm optimization (PSO), meta-optimized PSO and weighted majority algorithm (WMA) techniques to combine the experts' opinions and accurately create three new ensembles. After testing and comparing the three new techniques over some KDD99 datasets, the PSO ensemble approach achieved better results, improving accuracy by 0.756%, in a short runtime. The authors explained that the sets of generated weights, which were also optimized to produce results with the best possible accuracy, were responsible for the success of the PSO-based ensemble. Despite the fact that the meta-optimized PSO approach accomplished a better accuracy gain, it took 500 times more time to achieve it. On the other hand, the WMA approach had the worst results since it had a reasonably low accuracy of base classifiers for the occurrences of Normal and R2L classes.

Sornsuwit and Jaiyen [116] proposed a novel ensemble approach for intrusion detection using the AdaBoost algorithm, which combines the solution of the following classifiers: naïve Bayes, decision tree, multilayer perceptron (MLP), k-NN and SVM. The AdaBoost algorithm initializes the distribution of data, trains the classifiers, evaluates errors and assigns weights to each of them. Then, the combination of classifiers is linear and based on a weighted voting approach.

Bukhtoyarov and Zhuckov [117] developed an ensemble-distributed classifier for network IDS based on a new tree-level approach for combining the individual classifiers' decisions. The approach relies on using ensembles of neural networks designed through genetic programming-based ensembling (GPEN). GPEN automatically builds a program using genetic programming operators to indicate how to combine the component networks' predictions in order to get a reliable ensemble prediction. This study differs from others dealing with traditional ensemble since it provides the partial obtaining of adaptive outcomes by distinct classifiers deprived of an ensemble classifier.

2.4.4.5 Summary

To sum up, classification-based methods are prevalent due to its simplicity and effectiveness. Here are some additional advantages.

- Flexibility for testing and training by incorporating new information into the execution strategies.
- High detection rates for acknowledged attacks.
- Artificial Neural Networks have an adaptive nature, being possible to train and test cases incrementally.

Chapter 2. Related Work

- Regarding efficiency, multi-level neural network techniques are better than a single-level neural network.
- Ensemble methods perform well by combining multiple classifiers, even if they are weak ones.

However, despite being popular among researchers, there are some disadvantages, as follows.

- High resource consumption.
- Inability to detect unknown anomalies without some relevant training information.
- Neural network usage may cause over-fitting.
- The selection of sample datasets is slow for big datasets.
- In some cases, real-time performance is hard to acquire.

Table 2.8 summarizes some characteristics of discussed clustering approaches, regarding data precedence, investigated anomalies, and validation metrics used to test detection performance.

2.4.5 Information theory

Information Theory is a mathematical subject centered on the quantification of information and redundancy analysis. It was formerly envisioned by Claude E. Shannon, in 1948, while seeking data compression, transmission, and storage for signal processing and communication operations [118]. However, its application extended to many other purposes such as telecommunications, estimation, decision support systems, pattern recognition and so on [119]. There are several information-theoretic measures, such as Shannon entropy, generalized entropy, conditional entropy, relative entropy, information gain and information cost.

Its use for anomaly detection purposes relies mainly on the calculus of mutual information or entropy values for designated traffic features in order to identify anomalous distributions on them. Since it adopts statistical properties for the time series of a traffic-related features (e.g. Gaussian), this methodology may result in inaccuracies.

2.4.5.1 Entropy

Entropy is the most well-known information theoretical measure, defined as the equivalent probabilities, or the uncertainty, involved in the value of a stochastic variable or the occurrence of a random process. Considering the use of entropy in the anomaly detection field, it is efficient in describing traffic features, such as source/destination ports or IP addresses, as distributions, since there are certain types of anomalies causing significant disturbances on these distributions. In this manner, it is possible to detect, for instance, a port scan attack, indicated by a change in the entropy of destination ports, or even the occurrence of a DDoS attack, denoted by changes in the entropy of source/destination IP addresses [120].

David *et al.* [121] proposed an enhanced detection of DDoS attacks through a fast entropy method and the use of flow-based analysis. Authors aggregate the observed flows into a single one with consideration to the flow count of each connection at a certain time interval

Table 2.8: Comparison of classification-based anomaly detection approaches

Paper	Year	Tech. ^a	Anomaly Type ^b	Dataset ^b	Source ^c	Validation Metrics
Swarnkar and Hubballi [104]	2016	Naïve Bayesian	buffer overflow, shell-code attacks	Network of IIT Indore (R) and HTTP attack dataset (R)	-	Detection rate, FPR
Klassen and Ning [102]	2012	Naïve Bayesian	Black Holes, selective forwarding, DDoS (S)	NS2 simulated network traffic data (S)	-	Confusion Matrix (TP FP Precision Recall F-measure)
Tao <i>et al.</i> [103]	2008	Naïve Bayesian	Scan attack, DoS, ARP attack, Fragment attack, and comprehensive attack (R)	DARPA1999 (S)	TCP dump	average detection rate
Kabir <i>et al.</i> [109]	2017	LS-SVM	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	Precision, recall, F-value, probability of detection, probability of correct detection, FPR, accuracy
Wang <i>et al.</i> [108]	2017	SVM	DoS, Probe, R2L, U2R (R)	NSL-KDD (R)	TCP dump	Accuracy, DR, False alarm rate
Erfani <i>et al.</i> [107]	2016	SVM	Outliers (R)	UCI Machine Learning Repository (R) and two synthetic datasets (S)	-	ROC and Area Under the Curve (AUC)
Catania <i>et al.</i> [112]	2012	SVM	Generic attack distributions (R)	1998 DARPA (S)	TCP dump	Attack Detection rate (DR) and False Alarm rate (FA).
Amer <i>et al.</i> [106]	2013	SVM	Outliers (R)	UCI machine learning repository (R)	-	area under the ROC curve (AUC) and ROC curves
Subba <i>et al.</i> [110]	2016	Artificial Neural Networks	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	Accuracy, DR
Saeed <i>et al.</i> [111]	2016	Random Neural Networks	Data integrity attacks and ilegal memory access (S)	wireless sensor nodes-based IoT system (S)	-	Accuracy, FPR, FNR, TPR, TNR
Brown <i>et al.</i> [112]	2016	General Regression Neural Network	Generic anomalous instances (S)	UNB ISCX dataset (S)	features of application layer protocols	Accuracy, DR, FNR, TNR, FPR
Ashfaq <i>et al.</i> [113]	2017	Neural Network with random weights	DoS, Probe, R2L, U2R (R)	NSL-KDD (R)	TCP dump	Accuracy
Aburomman and Reaz [115]	2016	Ensemble (SVM, k-NN, PSO, WMA)	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	Accuracy
Sornsuwit and Jaiyen [116]	2015	Ensemble (Adaboost, Naïve Bayes, MLP, SVM, DT)	R2L, U2R (R)	KDD99 (R)	TCP dump	Sensitivity, Specificity
Bukhtoyarov and Zhuckov [117]	2014	Ensemble (GPEN, neural networks)	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	DR, FPR

^aClassification Techniques/Methods used.^bData precedence; R=Real, S=Simulated.^cSource types in blank are either not clearly specified by the authors or not relevant in their research.

Chapter 2. Related Work

instead of taking the packet count of every connection. The second step is basically the calculation of the fast entropy of the flow count for each connection. Finally, an adaptive threshold is generated based on the fast entropy and the mean and standard deviations of flow counts. The constant update of the threshold with regard to the traffic pattern condition improves detection accuracy, while fast entropy use reduces computational processing time.

Amaral *et al.* [122] proposed a feature-based anomaly detection system using both IP Flow properties and a graph representation in order to carry out a deep inspection of network traffic. The detection is based on the Tasallis entropy, a generalization of Shannon entropy. The major divergence is that it has a parameter to define which probabilities will contribute to the entropy result. It adjusts the sensibility of the anomaly detector, allowing it to adapt to different types of networks and detect more inexpressive attacks than those detected by methods based on volume analysis.

The work presented by Bhuyan *et al.* [123] brings an outlier-based anomaly detection approach using generalized entropy and mutual information for creating a feature selection technique capable of choosing a relevant, non-redundant subset of features. According to the authors, since mutual information reduces the uncertainty about one random variable and generalized entropy measures the amount of uncertainty in the data, they make detection faster and more accurate.

Moreover, Berezinski *et al.* [124] introduced a network anomaly detector, based on Shannon entropy, in order to detect modern botnet malware. Their approach created a network profile, which stores min and max entropy values in a sliding time window of 5 minutes. These values were used for comparison with the observed entropy. This defines a threshold, thus, abnormal dispersion or concentration for different feature distributions can be identified. Finally, the authors used popular classifiers, such as decision trees and Bayesian networks, in order to classify the anomalies.

Behal and Kumar [125] stated that since DDoS attacks and flash events cause substantial alterations in network traffic patterns, information theory-based entropy or divergence can rapidly capture such disparities in network traffic behavior. Therefore, they proposed a generalized anomaly detection algorithm, which exploits the entropy difference between traffic flows. They employed a set of generalized ϕ -Entropy and ϕ -Divergence metrics, in which the detection efficiency was directly connected to the information distance between legitimate and attack traffic. The proposed algorithm resulted in high detection accuracy with regard to flash events and High-Rate DDoS, overcoming the results of other information theory approaches in the literature.

2.4.5.2 Kullback-Leibler distance

The Kullback-Leibler Distance or Divergence (KLD) measures the difference between the true probability distribution P and an arbitrary probability distribution Q (an approximation of P).

The work of Xie *et al.* [126] consisted of an algorithm to track long-term anomalies in WSNs by using the Kullback-Leibler divergence to measure the differences between global

Probability Density Functions (PDF) for each of two consecutive periods of time. This function produces a time series to be analyzed and make decisions based on the adaptive threshold, identifying any unusual changes. The approximate Kullback-Leibler divergences, obtained from distributed computing with no significant accuracy degradation, is used to reduce the communication cost since it can reflect the variation among PDFs in a sensitive manner. Li and Wang [127] proposed a differential Kullback-Leibler divergence based anomaly detection scheme for wireless sensor networks. The authors used a clustering approach to separate the sensor nodes into clusters. All the nodes composing a cluster had related sensed value and were physically close to each other. Then, the Kullback-Leibler divergence was used within each cluster in order to detect abnormal values by statistically measuring the disparity between two data sets. Their work achieved a good detection rate and low false alarm rate while consuming less energy than other similar studies in the literature.

2.4.5.3 Summary

In conclusion, information theoretic-based approaches have been emerging increasingly in the network anomaly detection field. Their main benefits are that they can be highly scalable, very sensitive and low to false positives. Other advantages are stated below.

- Operating in an unsupervised mode is possible.
- There are no assumptions about the primary statistical distribution for the data.
- Since information theory-based methods only use header information for calculation, the complexity of time and space is a minor problem.

Besides, they are susceptible to these limitations.

- The adoption of statistical properties for the time series of traffic-related features (e.g., Gaussian) may cause inaccuracies.
- The detection of anomalies may be possible only if there is a significant presence of them in the data set. This way, these approaches need a highly sensitive information theoretic measure to detect irregularities made by very few anomalous patterns.
- Difficulty in associating an anomaly score with a trial case.

Table 2.9 summarizes some characteristics of discussed clustering approaches, regarding data precedence, investigated anomalies, and validation metrics used to test detection performance.

2.4.6 Evolutionary computation

The field of evolutionary computation, also named bio-inspired computing [128], is a set of intelligent algorithms and methods inspired by natural evolution and able to learn and adapt like biological organisms [129]. It encompasses genetic algorithms (GA), genetic programming (GP), evolution strategies (ES), particle swarm optimization (PSO), and artificial immune systems (AIS) [130, 131].

2.4.6.1 Artificial Immune Systems (AIS)

Artificial Immune Systems (AIS) are adaptive systems, enhanced by theoretical immunology and biological immune system functions, principles and models, and are applied to

Chapter 2. Related Work

Table 2.9: Comparison of Information Theory anomaly detection approaches

Paper	Year	Tech. ^a	Anomaly Type ^b	Dataset ^b	Source ^c	Validation Metrics
David <i>et al.</i> [121]	2015	Entropy	DDoS (R)	CAIDA dataset (R)	IP flow	Empiric entropy variation analysis
Amaral <i>et al.</i> [122]	2017	Entropy	DDoS, alpha flow, portscan, network scan (R/S)	Universities traffic (R)	IP Flow	TPR, FPR
Bhuyan <i>et al.</i> [123]	2016	Entropy	DoS, probe, R2L, U2R (S)	Testbed dataset (S), KDDcup99, NSL-KDD, UCI ML repository datasets (R)	Packet/flow records	Detection rate, precision, recall, f-measure
Berezinski <i>et al.</i> [124]	2015	Entropy	Port scan, DDoS (S)	Legitimate traffic from medium size network (R)	IP flow	Correlation, Accuracy, FPR, TPR, ROC
Behal and Kumar [125]	2017	Entropy	DDoS, flash events (S)	MIT Lincoln dataset, FIFA, DDoSTB, and CAIDA datasets (R) and D-ITG traffic generator, Bonesi (S)	IP flow	DR, precision, FPR, TNR, NPV, F-measure, F-measure complement, Classification rate
Xie <i>et al.</i> [126]	2017	Kullback-Leibler	Long-term anomalies	RSS measurement from UMICH network (R)	-	FPR, Accuracy, ROC
Li and Wang [127]	2012	Kullback-Leibler	Generic anomaly data values (R)	real sensed data (R)	-	Detection rate, FPR

^aInformation Theory Techniques/Methods used.

^bData precedence; R=Real, S=Simulated.

^cSource types in blank are either not clearly specified by the authors or not relevant in their research.

problem-solving, as defined by de Castro and Timmis [132].

The authors of [133] presented EPAADPS, a proactive anomaly detection and prevention system based on an Artificial Immune System (AIS) aiming to identify and prevent new and undetected anomalies. Their motivation lied on gaps found in related previous studies, such as the lack of co-operation between detectors in order to classify any pattern as anomalous, the identification and inhibition of novel and zero-day attacks and lacks in self-configuration, learning, adaptability and preventive abilities. The whole system consists of three modules: the repertoire training module (RTM), responsible for selecting efficient detectors to generate a detector set (DS); the vulnerability assessment module (VAM), which creates collaborative detector agents (DA) able to correctly identify and flag any test set instance happening to be an anomaly; and response module (RM), which takes appropriate preventive actions in the cases where VAM found an anomalous instance. Saurabh and Verma also combined PCA and min-max normalization as a pre-processing feature in order to make the dataset both substantial and stable, respectively. In this manner, the number of features is reduced, helping the choice of better-trained detectors.

Moreover, Igbe *et al.* [134] proposed a distributed NIDS against cyber-attacks using the Negative Selection Algorithm (NSA), which exist in the AIS field. The entire system has autonomous agents communicating with each other while running the NSA to create classification rules. These rules and identified threat vectors are shared among all agents and enhance the fast detection of more problems.

Shahaboddin *et al.* [135] introduced Co-FAIS, a cooperative-based fuzzy artificial immune system for detecting malicious activities in a WSN. The adopted defense strategy is modular and derived from the danger theory of the human immune system as an AIS. The agents work in a mutual way in order to identify attackers or any abnormalities in sensor behavior regarding the context antigen value (CAV). Then, agents inform the Fuzzy Q-learning algorithm initiation threshold, which examines the attack behavior and checks if the system can respond and defend itself. That response was designed to act similar to the ability of rapid response to recurring attacks present in a natural immune system. The response module elaborates an attack signature and eliminates it from the safe list; therefore, if repeated, the reaction to the same attack will be quicker.

2.4.6.2 Genetic algorithms (GA)

Genetic algorithms (GA) are commonly used as part of a whole intrusion detection system together with other techniques. As in [136], the authors use a genetic algorithm to transform the data set such that an SVM classifier can better process it. In [124], for instance, the authors combine a genetic algorithm (GA) with kernel principal component analysis (KPCA). The genetic algorithm creates a new optimal set of features and assigns a separate group with a certain priority to each obtained feature.

In their research [137], Singh and Kushwah employed genetic algorithms to build an optimized cluster-based intrusion detection system in wireless sensor networks. The entire system was divided into four modules: the data collection module, which makes the head node observe the movement of the member sensor node; the intrusion information module, which gathers intrusion information for explanation; the intrusion detection module, responsible for setting a device activity as the misbehavior or legitimate behavior prior to a threshold; and the alert module, in which the cluster-head node alerts nearby nodes about the existence of intrusion. A genetic algorithm is used to mutate the nodes presenting less energy by a mutation parameter with a mutation probability in order to flip the node energies; thus, power consumption and network efficiency are improved.

Another approach using Genetic Algorithms is presented by Hamamoto *et al.* [138]. The GA is used to deal with uncertainties in network traffic, and through natural selection, learn the normal characteristics of the traffic flows. As all traffic attributes used in the research are numeric, the authors applied a numeric chromosome encoding to optimize each time interval separately and each attribute in parallel. The result is the Digital Signature of Network Segment using Flow Analysis (DSNSF), a prediction of the network traffic behavior in each time interval. Moreover, the authors also added a Fuzzy logic approach to assess whether a time interval in the IP flows data has an anomaly or not. The evaluation was conducted by using a real network traffic from an university with simulated anomalies injected in some flow entries.

2.4.6.3 Differential evolution

Differential evolution is a global search evolutionary algorithm also used for detecting anomalies. Although it is not widely used yet, it has great potential in order of being worthy to mention in this section. It encompasses two concepts within the area: the idea of using larger

Chapter 2. Related Work

population from genetic algorithms and self-adapting mutation from evolutionary strategies. Elsayed *et al.* [139] applied a feature reduction mechanism using a flexible neural tree to select significant traffic features and then adopted a differential evolution algorithm to evolve individual (rules) for anomaly detection. A fitness function calculates the quality of every rule or individual.

2.4.6.4 Particle swarm optimization

Particle Swarm Optimization is a common evolutionary computation technique used for anomaly detection. Its main purpose is to perform an optimum search, and that is why this algorithm is mainly combined with clustering techniques and classifiers, such as k-means [34, 93] and SVM [115, 140, 141], for instance (please refer to works discussed in sections 6.2 and 6.4).

Bamakan *et al.* [142] proposed a novel intrusion detection framework by using a modification of the PSO, called time-varying chaos particle swarm optimization (TVCP SO). It is a new adaptive, robust, precise optimization method, aimed at doing parameter setting and feature selection for multiple criteria linear programming (MCLP) and SVM simultaneously. The authors introduced time varying inertia weight and a time varying acceleration coefficient, along with the adoption of the chaotic concept in the PSO. In this manner, the PSO algorithm searches the optimum faster than normal, while avoiding the search being stuck to a local optimum.

2.4.6.5 Summary

Evolutionary computation methods are increasingly obtaining distinction due to their intelligent algorithms that can learn and adapt like real living organisms, therefore being able to produce latent solutions to many of the complex network problems that have been intensified recently. Other advantages of these methods are the following.

- They add to intrusion detection systems capabilities for parallel processing.
- Prior knowledge of the problem space is not required.
- The natural retraining ability makes the entire system more adaptable.
- Noise and discontinuities existing in the dataset do not cause a considerable impact on solutions.

Although their efficiency, evolutionary methods also have some limitations.

- The fitness function may not be trivial to find.
- Choosing the optimal parameters is hard.
- Sometimes, it can be a complicated task to map the problem into a biological approach.

Table 2.10 summarizes some characteristics of the discussed evolutionary computation approaches, with regard to data precedence, aimed network paradigm, techniques, anomalies, and validation metrics used to test detection performance.

2.4.7 Hybrid/others

This section presents hybrid approaches to anomaly detection, which are a combination of various classes of algorithms, techniques, and methods. Additionally, unclassified

Table 2.10: Comparison of evolutionary computation anomaly detection approaches

Paper	Year	Tech ^a	Anomaly Type ^b	Dataset ^b	Source ^c	Validation Metrics
Saurabh and Verma [133]	2016	AIS	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	FPR, DR
Igbe <i>et al.</i> [134]	2016	AIS	DoS, Probe, R2L, U2R (R)	KDDTrain+20% (R)	TCP dump	DR, FPR
Shahaboddin <i>et al.</i> [135]	2014	AIS, FQL	DDoS (S)	NS-2 simulation (S)	UDP traffic	Accuracy, FNR, FPR
Singh and Kushwah [137]	2016	GA	WSN node issues	Sensor Field of Area 100×100 m (R)	-	Packet delivery rate, end-to-end delay, distance vector, system lifetime and throughput
Hamamoto <i>et al.</i> [138]	2017	GA	DoS, DDoS, Flash crowd (S)	University dataset (R)	NetFlow	Accuracy, precision, recall, F-measure, FPR, ROC AUC, Mis. Rate
Elsayed <i>et al.</i> [139]	2015	DE	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	DR, FPR, FNR
Bamakan <i>et al.</i> [142]	2016	PSO	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	Accuracy, DR, FPR

^aEvolutionary Computation Techniques/Methods used.

^bData precedence; R=Real, S=Simulated.

^cSource types in blank are either not clearly specified by the authors or not relevant in their research.

techniques, which are not listed in previous sections but are still interesting and promising, are also listed here.

Grill and Pevný [143] state that successive alarm analysis is costly and cannot cover all alarms, only a small portion, as well as the noise in training data is always an important feature to consider. Moreover, combining anomaly detectors, although simple, may become a significant challenge when it attempts to combine the output of individual detectors. Therefore, the authors propose a novel approach to finding a convex combination of various anomaly detector outputs and carried out a study on the effects of label noise in the training dataset over the accuracy of combinations achieved by different detectors. They compare their approach to two existing ensemble methods, one using NetFlow and the other using HTTP server logs.

Another interesting hybrid intrusion detection system is proposed by Al-Yaseen *et al.* [144], in which the authors combine the SVM and extreme machine learning (EML) classifiers and the k-means clustering technique. The classifiers are responsible for reducing false positives as well as improving detection accuracy. The categories of attacks are divided into three groups; four SVMs classify instances as DoS, U2R, R2L, or Normal while an ELM classifier detects probe attacks, since they are better for them than an SVM. On the other hand, k-means is modified to build a suitable training dataset, which can meaningfully contribute to improving the classifiers' training time and overall performance. The modification consists of selecting the initial centroids of clusters conditional to the maximum distance between them and dataset instances. Five datasets are produced to each one of the five classification categories and serve as the basis of creating accurate SVM and ELM classifiers.

Forestiero [145] used a swarm intelligence technique to build a bio-inspired clustering algorithm in order to identify anomalies in distributed data streams. Bio-inspired agents follow the principles of the flocking-based examination approach, which states that agents will

Chapter 2. Related Work

interact autonomously with immediate neighbors and form flocks (clusters) of similar agents. The similarity between agents depends on the carried data items and can be calculated using various techniques such as measuring the Euclidian distance of associated data items.

Salem *et al.* [146] developed a framework for anomaly detection that operates in wireless body area networks (WBAN). They combined the SVM classification algorithm with the statistical linear regressive model. The SVM part classifies incoming sensor data as normal or abnormal. Then, whenever an abnormality is found, a linear regressive prediction model analyzes it and decides whether the patient is entering a dangerous state or a sensor is reporting incoherent readings. This decision is accomplished by building a decision tree and searching for linear coefficients from normal vital signs falling inside a given threshold.

Wang *et al.* [147] combined three classes of algorithms for the purpose of introducing a data abstraction phase situated between the well-known attribute construction and detection of model building phases that most IDSs have. Their idea was applied to process big data by reducing the amount of data while keeping the valuable information they carry. For that purpose, three strategies were proposed and evaluated. The attribute abstraction strategy was based on applying PCA for reducing the data to a low dimensional subspace and then projecting the testing data onto it in order to detect anomalies. The attribute selection strategy consists of calculating the information gain (IG) to rank the correlations of each attribute to the class - whether it is normal or attack - and select key attributes based on this ranking. After the selection, the authors combined this with a k-nearest neighbor and a PCA or SVM-based detection approach. Finally, the exemplary extraction strategy uses either k-means or affinity propagation clustering techniques to extract exemplars from the large audit data. After the extraction, authors also combined this with a k-nearest neighbor, a PCA or an SVM based detection approach.

Adaniya *et al.* [148] created a hybrid anomaly-based clustering approach for anomaly detection, combining the k-harmonic means (KHM) clustering method with the bio-inspired heuristic firefly algorithm (FA). The traffic profile is created through the GBA tool by using the historical traffic data, proposed by Proença *et al.* [149]. KHM solves the initialization sensitivity of k-means and the FA helps it converge to local optima. This approach groups data points in order to separate normal from abnormal ones. They achieved good outcomes, with true-positive rates above 80% and false-positive rates below 20%.

Chen *et al.* [150] managed to build a novel classifier through an evolutionary computation basis for intrusion detection. The central segment of this IDS is an artificial immune system (AIS), which is enhanced by a population-based incremental learning (PBIL) procedure. The PBIL enhancement in the AIS consists of evolving new antibodies with higher affinities than older ones, which are not capable of properly recognizing the class (removal of weak antibodies). Then, the authors combined the AIS-PBIL with collaborative filtering (CF) in order to cluster all antibodies related to a target-occurrence and categorize target intrusions.

Bostani and Sheikhan [57] proposed a novel hybrid IDS framework consisting of anomaly-based and specification-based modules. Their goal was the detection of two routing attacks that cause significant problems in IoT: sinkhole and selective-forwarding attacks. The framework is divided into three stages. In the first stage, the specification-based agents in the router nodes identify suspicious nodes by analyzing the behavior of their host nodes and sending it to the root

node. In the second stage, an anomaly-based agent located in the root node uses that information to extract traffic features and create samples for each source node. This is accomplished by using an unsupervised optimum-path forest algorithm and a MapReduce architecture for projecting clustering models. Finally, the last stage uses the first stage results to make decisions about mistrustful behavior detected in the second stage through a voting mechanism.

Grill *et al.* [151] propose a local adaptive multivariate smoothing (LAMS) method to effectively smooth an anomaly detector output in order to reduce the rate of unstructured false positives by the Nadaraya-Watson estimator. It replaces the output of a networking event with an aggregate of its output on similar network events observed previously.

Guo *et al.* [152] combine both misuse detection and anomaly detection to build an IDS. The development is divided into 2 phases. The first phase is the elaboration of a lightweight misuse detector based on the change of location of cluster centers. In phase 2, two anomaly detectors are built using the k-nearest neighbor (k-NN) algorithm. By this combination, authors were capable of detecting both known and unknown anomalies with a low false positive rate (FPR).

As an emergent network paradigm, Software-Defined Networks (SDN) also face the problem of DoS attacks, since massive malicious requests can truly harm their centralized control characteristic. Thus, although many researchers propose detection mechanisms, most of them only focus on detection itself. In this manner, Assis *et al.* [150] proposed GT-HWDS, a hybrid autonomic defensive approach for SDNs against DoS/DDoS attacks by applying a game theory (GT) decision-making model together with their Holt-Winters-based anomaly detection system (HWDS [153]). The GT-HWDS system is fully able to detect, identify and mitigate events of DoS/DDoS in SDN traffic. Their core contribution is the mitigation module performed by a GT-based method. GT consists of changing a problem with opposing interests into a game, where many “players” take actions to optimize the results of trying to achieve their objectives. Therefore, the system analyzes a set of probable actions for both attacker (malicious nodes) and defense systems, estimates rewards and costs for all measures, and finally, performs an optimal countermeasure. This blocks (mitigates) any traffic originating from the attackers’ IP and port.

2.4.7.1 Summary

In summary, hybrid techniques are an excellent choice for situations in which the same system might solve many distinct problems. Also, one technique may overcome the limitations of others, leading to a more reliable system. These are the main advantages of hybrid anomaly detection methods.

- Hybrid methods can benefit from the main features of both anomaly and signature-based approaches.
- They can detect both known and unknown anomalies.

However, when developing hybrid methods, there some limitations to consider.

- As more techniques are used together, there is a high demand for computational resources, increasing its cost.
- Dynamism is still an unsolved problem.

Chapter 2. Related Work

Table 2.11: Comparison of hybrid/unclassified anomaly detection approaches

Paper	Year	Tech ^a	Anomaly Type ^b	Dataset ^b	Source ^c	Validation Metrics
Al-Yaseen <i>et al.</i> [144]	2017	SVM, EML, k-Means	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	FPR, detection rate, accuracy
Bostani and Sheikhan [57]	2017	optimum-path forest algorithm	Sinkhole attack, selective-forwarding attack	WSN simulator (S)	-	Accuracy, TPR, FPR
Grill and Pevný [143]	2016	Classification methods	Portscans, SSH brute force (R) / ZeroAccess and other malwares (R)	Czech Technical University network (R) / 30 arbitrary companies traffic data (R)	NetFlow / HTTP logs	Precision and recall
Forestiero [145]	2016	Flocking algorithm	Generic anomalies	Gauss and STREAM (S); UCI Machine Learning Repository and 1998DARPA (S)	TCP dump	Normalized Mutual Information (NMI), Precision and recall
Salem <i>et al.</i> [146]	2014	SVM and Linear Regression	Outlier detection (R)	Physionet database (R)	-	FPR, TPR, ROC
Wang <i>et al.</i> [147]	2016	PCA, IG, AP, k-means, SVM, k-NN	http common attacks, DoS, Probe, R2L, U2R (R)	Real http traffic dataset and KDD99 (R)	-	Detection rate, and execution time
Adaniya <i>et al.</i> [148]	2013	K-Harmonic Means, Firefly Algorithm	DoS, Flash Crowds (R)	University network (R) and simulated anomalies (S)	SNMP	
Chen <i>et al.</i> [150]	2016	AIS, population-based incremental learning, collaborative filtering	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	Accuracy,
Grill <i>et al.</i> [151]	2017	local adaptive multivariate smoothing (LAMS)	Generic artificial attacks (S) and malwares (R)	Czech Technical University (CTU) network (R) and varied HTTP database (R)	NetFlow, HTTP proxy logs	AUC,
Guo <i>et al.</i> [152]	2016	location of cluster centers, k-NN	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	DR, TNR, FPR, Accuracy
Assis <i>et al.</i> [150]	2017	Game Theory (GT), Holt-Winters	DoS, DDoS (S)	University network traffic (R)	NetFlow	Precision, accuracy, drop rate

^aTechniques/Methods used.

^bData precedence; R=Real, S=Simulated.

^cSource types in blank are either not clearly specified by the authors or not relevant in their research.

Table 2.11 summarizes some characteristics of the approaches discussed in this section, with regard to data precedence, aimed network paradigm, techniques, investigated anomalies, and validation metrics used to test detection performance.

2.5 Open Issues

There are a significant number of challenges within the anomaly detection field. This section aims to summarize the most relevant open issues found during the development of this thesis and also to consider those most discussed in the literature. All of them were identified by analyzing and comparing all surveys [9,12-18] listed in table 1 and every research addressed in this survey. The list and a brief discussion upon each topic can be seen below:

- *The concept of normality*: It is one of the main steps to build a solution to detect network anomalies. The question “how to create a precise idea of normality?” is what has driven most researchers into creating different solutions through the years. This can be considered as the main challenge related to anomaly detection and has not been entirely solved yet. Many of the works discussed in this survey tried to achieve this goal.
- *Adaptability*: Anomalies keep changing every time new ones are introduced or old ones are improved to overcome current detection solutions. Therefore, IDSs need to be constantly updated in order to adapt to those changes, and this is not an easy task.
- *Dynamic profile update*: Whenever an unknown attack is detected and addressed by anomaly-based IDSs, the profile database needs to be updated with these new data. Nevertheless, it is a challenge to carry out such updates dynamically, without compromising performance and generating conflicts.
- *Standard datasets*: There are only a few openly available intrusion datasets with enough information about attacks; however, none of them is a standard evaluation dataset for anomaly detection. The lack of reliable public standard datasets, which can simulate accurate network environments, is still a problem.
- *Noisy data*: Normal variations in datasets are also a problem when creating a profile since they can be misunderstood as abnormalities if they are not well defined. Moreover, this information is neither always clear in public datasets nor private ones.
- *False alarm rates*: Another problem is to keep false alarms as minimal as possible; although it is still not possible to completely avoid them and build a one hundred percent reliable IDS. That still remains a challenge.
- *Real-time monitoring*: The amount of traffic generated by computer networks today is constantly increasing as Internet traffic doubles every year. Therefore, it has been difficult to produce a reliable monitoring process on a network, in real time.
- *Complexity*: As researchers try to cover all the challenges mentioned above, the complex nature of the systems increases by adding and mixing different techniques and approaches. Additionally, regarding data collection and preprocessing, the complexity of today’s network architectures also contributes to the persistence of this issue.

The graph in Figure 2.10 shows the relevance of each open issue discussed in this section amongst the other analyzed surveys [6, 18, 9, 19, 20, 21, 10, 22] presented in Table 2.1, showing the most concerning issue in the anomaly detection field based on the study in this survey. For instance, the complexity issue appeared in 43% of the other surveys, while the standard dataset problem was considered in 60%. So, it can be observed that although

Chapter 2. Related Work

many of the other discussed topics had a significant rate of discussion, the problem of not having a standard and updated dataset that simulates a real environment and contains labels for anomalies is a major concern among practitioners in the literature.

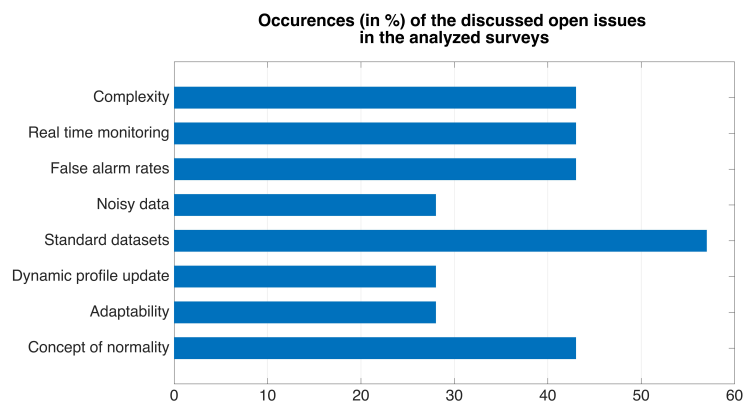


Figure 2.10: Occurrences (%) of discussed open issues in the analyzed surveys

Chapter 3

The Proposed Anomaly Detection System using Principal Component Analysis

In this chapter, the hybrid anomaly detection system using principal component analysis is presented. However, before explaining its full process, Figure 3.1 summarizes the overall operation of it. The system is divided in two parts: Traffic Characterization and Anomaly Detection. The traffic characterization is responsible for extracting quantitative attributes (bits/s, packets/s and number of flows/s) from a flow database containing historical data about the network segment activity, and generate the corresponding DSNSFs.

In the anomaly detection phase, the first step is the creation of confidence bands based on the DSNSFs. The Detection Stage analyses those confidence bands along with the real time network traffic observed from a network device (switch). This analysis is performed using only the three quantitative attributes, and it identifies time intervals where an anomalous event occurred. Then, the Reporting Stage is informed about those time intervals, which uses the qualitative attributes observed from the network device in order to generate a top-N ranking list. It contains useful and detailed information about the abnormal interval identified, such as source and destination IP addresses and ports with higher occurrence frequency. At last, the network administrator is notified about the abnormality and its qualitative information, so he can direct its efforts in solving that issue.

3.1 Traffic Characterization

Principal Component Analysis (PCA) was first introduced in 1901 by Karl Pearson [154], and it is a statistical technique used for data compression and classification. The main idea of PCA is to reduce the dimensionality of a data set comprised of a large number of correlated variables, retaining as much as possible of the variation of the data set. This is achieved through transformation into a new set of variables, the principal components (PCs), which are uncorrelated and arranged so that the first components retain most of the variation present in all original variables, i.e., the input data can be represented by a reduced set of dimensions without much loss of information [68, 69].

In the standard PCA algorithm, the input is an $n \times p$ matrix, composed by p columns representing the dimensions (variables) and n lines, as the n samples of each variable.

Data collected from flow records are arranged in such a way that the traffic movement of each day is denoted by three vectors containing a total of bits, packets and flows corresponding to the 24 hours of each day. Thus, the input matrix of PCADS-AD algorithm is constructed for each attribute separately. The p dimensions will be the p traffic movements (days) chosen as a basis to generate the DSNSF, and the n lines will be the n measurements of bits, packets or number of flows transmitted per second extracted from the flow records.

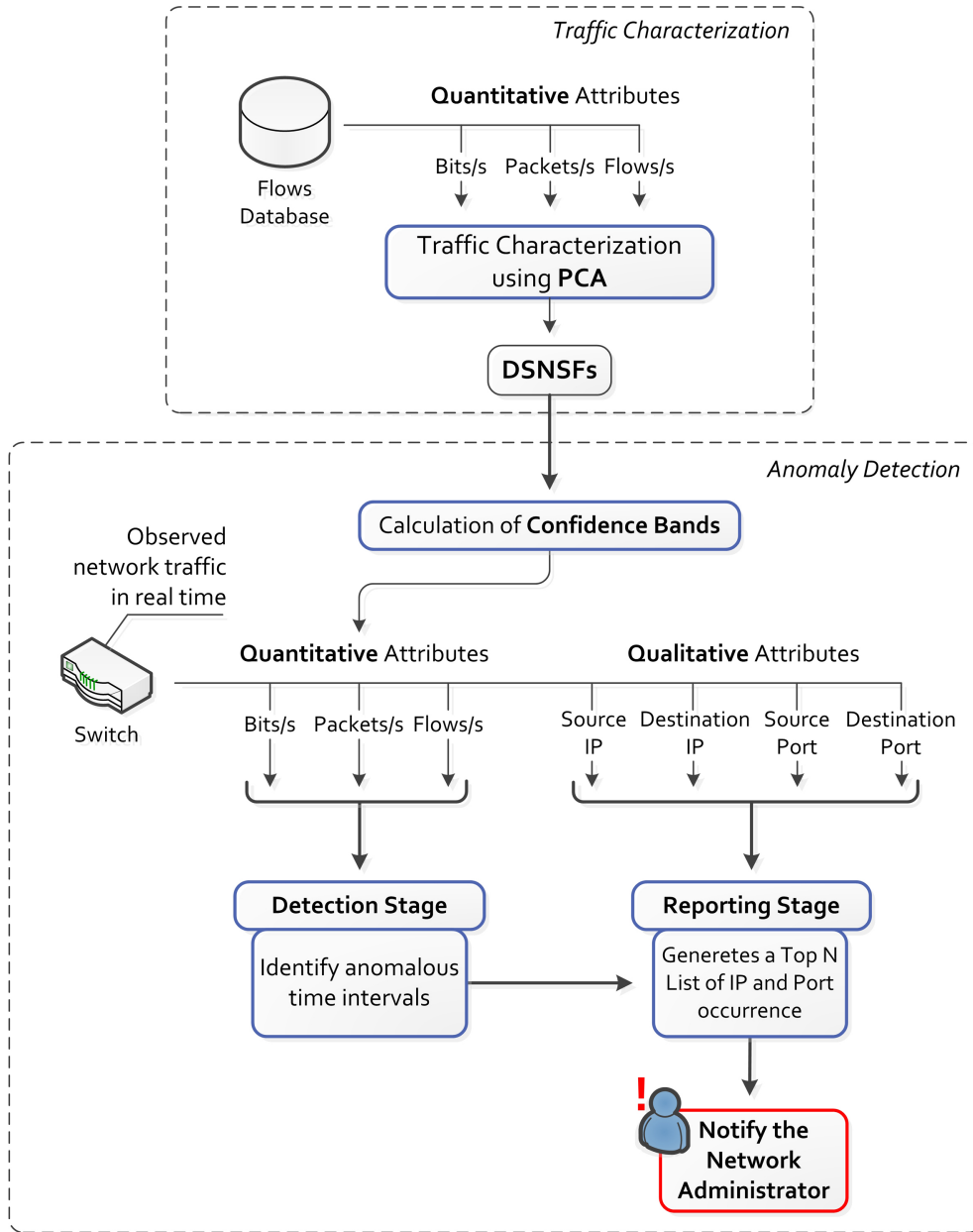


Figure 3.1: PCADS-AD System Description

The PCA method used in the PCADS-AD characterization process is presented in Algorithm 1. Since each traffic period has its own characteristics and seeking to prevent a period from interfering in another one, the algorithm is performed in a one-minute time window ($t = 24 \text{ hours} / 1 \text{ minute} = 1440 \text{ time intervals}$).

First, it is required to move the origin to the mean of the data set, by subtracting the mean from each column in the input matrix, so that its columns have zero mean. This is called a mean-centered matrix, and it is important because it ensures that PCA dimensions capture the true variance, and hence avoid distorted results due to differences in mean link utilization [64].

Algorithm 1 - Algorithm used for DSNSF creation

Require: Set of bits/s, packets/s or flows/s collected from historic database arranged in a $n \times p$ matrix.

Ensure: μ : a vector representing the bits/s, packets/s or flows/s sets of a day, arranged in 1440 intervals of 1 minute, i.e. the DSNSF.

```

1: for t=1 to 1440 do.
2:   Normalize the input data (mean deviation form).
3:   Calculate the covariance matrix.
4:   Calculate the eigenvectors and eigenvalues.
5:   Select the eigenvector associated to an eigenvalue of intermediate value amongst the others
6:   Multiplies the selected eigenvector by the input matrix in mean-centered form
7: end for
8: return  $\mu$ 

```

Then, the covariance matrix is calculated by the algorithm, using the mean-centered matrix, as can be seen in Equation 3.1:

$$C_X = \frac{1}{n-1} X^T X, \quad (3.1)$$

where C_X is the covariance matrix with p rows and p columns, X is the input $n \times p$ matrix in mean-centered form, and n is the number of bits, packets or flow samples.

The covariance matrix measures in what way the variables of X change together and provides a main diagonal of variances, specifying the direction and strength of the linear correlation amongst two variables [155]. It is used to compute two important structures: the eigenvectors and eigenvalues. Each dimension has an associated eigenvector, which points toward the variance of data, and an eigenvalue, a numerical value which indicates the significance of its associated dimension among the others. In other words, it shows the amount of information of the data set that a dimension can represent. These structures are obtained by decomposing CX as shown in Equation 3.2):

$$C_X = QDQ^T, \quad \text{with } Q^T Q = I, \quad (3.2)$$

where C_X is the covariance matrix, the columns of the matrix Q are the orthonormal eigenvectors of C_X , and $D = (\delta_1, \delta_2, \delta_3, \dots, \delta_n)$ is the diagonal matrix of eigenvalues. This decomposition is computed by solving the symmetric eigenvalue problem, as presented in [156, 157].

After computing all eigenvectors and eigenvalues, the eigenvectors with the highest eigenvalues are called principal components, and they are used by standard PCA to compose a new reduced dataset. In the proposed algorithm, a digital signature using only one principal component (eigenvector) is created. However, instead of selecting the eigenvector with the highest eigenvalue, an eigenvector with a corresponding eigenvalue of intermediate value is chosen. This is because the most significant eigenvector represents the dimension with the

largest variance among all components of the dataset, and creating a digital signature based on that premise will cause an impact on the normal traffic pattern of the network segment, such as the incorporation of irregular outliers present in the traffic of the day. And similarly, the use of an eigenvector of low significance may involve the incorporation of situations where there has been server crash or power outage.

After numerical experiments, it was observed that a component, whose variance corresponds to an average (intermediate) value between the components of maximum and minimum variances, produces a more uniform digital signature, befitting with the normal behavior. So, it prevents possible disparities (anomalies) in the training dataset to generate noise in the DSNSF. Figure 3.2 shows the difference between digital signatures (DSNSFs) generated using eigenvectors of maximum, intermediate and minimum significance. In green, there is the real traffic in bits/s for the 24 hours of November 08th, 2012, and in blue, the DSNSFs generated for that day. Note that in Figure 3.22 (a), where an intermediate eigenvector is used, the DSNSFs are closer to the real traffic behavior. In contrast, the signatures from Figure 3.2(b) and Figure 3.2(c) do not show good traffic estimations, as they notably deviate from observed real traffic behavior. Furthermore, note that in Figure 3.2(c), there are some spikes in the DSNSF, which is a result of outliers or anomalies in the historical traffic data used to create the DSNSF. Thus, a median eigenvalue is chosen among all p eigenvectors.

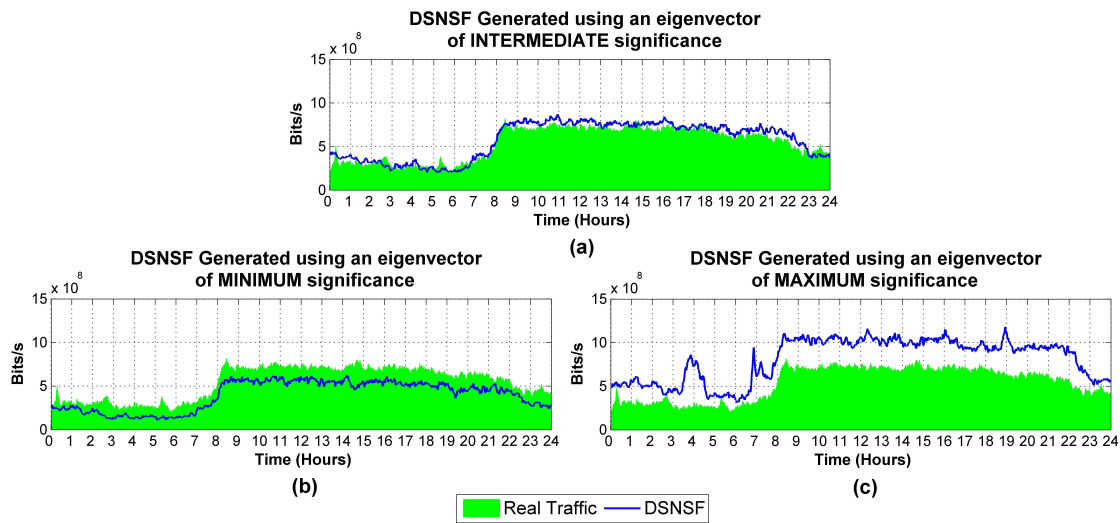


Figure 3.2: Comparison between DSNSFs generated using eigenvectors of intermediate (a), minimum (b) and maximum (c) significance

In step six of Algorithm 1, for each time interval t , the selected eigenvector is multiplied by the mean-centered matrix, as show in Equation 3.3:

$$\mu_t = \tilde{q}_t^T \times X^T, \quad \text{with } t = 1, \dots, 1440, \quad (3.3)$$

where μ is the DSNSF, \tilde{q} is the selected eigenvector that has an eigenvalue with an median value, and X is the input matrix in mean-centered form.

Finally, after computing it for each time interval t , the output is the DSNSF.

3.2 Anomaly Detection

The DSNSF is a network profile which estimates the traffic behavior of a network segment, and Figure 3.2 (a) illustrates an example. The blue line is the DSNSF, which describes the expected network behavior for the traffic of bits transmitted per second, while the green area is the real network activity. Based on this, wherever the real traffic exceeds or goes below the DSNSF, that particular time interval will be classified as an anomalous event, which is the central idea of a profile-based anomaly detection approach. The detection system will then trigger an alarm so that the network administrator can direct its efforts to that problem. However, there are some time intervals which deviate slightly from the DSNSF and cannot possibly be a real anomaly. If all deviations were considered an anomaly, hundreds of alarms would be generated for each day, contrasting with the idea of self-management, as the network administrator would be notified numerous times to observe and analyze an excessive number of alarms, where most of them might not be relevant.

Then, to minimize the excessive appearance of alarms and to classify as anomaly only critical and important volume deviations, it is proposed the creation of confidence bands or thresholds, based on the DSNSF. According to Assis *et al.* [63], confidence bands are an effective approach for anomaly detection, in which it is possible to specify an interval where deviations are considered normal. In this manner, a higher and a lower threshold to the DSNSF are created, so that the real traffic which slightly varies from the DSNSF will not set off an alarm. These percentages are based on the eigenvalue of the eigenvector chosen by the algorithm to create the DSNSF in the traffic characterization phase. The sum of all the eigenvalues of a data set results in the total variance of that dataset, and since the DSNSF is generated from an eigenvector, explicit in Section 3.1, the eigenvalue that represents the eigenvector variance (significance) may thus represent the variance of DSNSF itself, leading to a percentage, α . From α , the thresholds, called *Eigenvalue Limit (EL)*, are calculated as described in Equation 3.4 and Equation 3.5:

$$EL_{up} = \mu + \mu \frac{\alpha}{100}, \quad (3.4)$$

$$EL_{down} = \mu - \mu \frac{\alpha}{100}, \quad (3.5)$$

where EL_{up} is the upper threshold, EL_{down} is the bottom threshold, μ is the DSNSF, and α is the percentage obtained from the eigenvalue associated with the eigenvector used in the DSNSF composition.

In addition, as the system analyses the traffic of bits, packets and flows, it was developed to trigger an alarm only when an abnormal event is detected in two or more attributes in the same time interval.

Hereafter, PCADS-AD uses the qualitative attributes extracted from traffic flows, as shown in Figure 3.1. After collecting this information, they are processed in order to find the total occurrence for each IP address and Port number. When an alarm is triggered, the system starts the Reporting Stage, which generates a top-N list for the detected anomalous time interval

Chapter 3. The Proposed Anomaly Detection System using Principal Component Analysis

containing the N source and destination IP addresses and Ports, and their occurrences. These qualitative statistics help the network administrator to find the source and target of the attack, while Port numbers can identify the application that was targeted by the attacker or had a problem.

Chapter 4

Performance Evaluation

In this chapter, all evaluations and experiments to consolidate the proposed anomaly detection system is divided into evaluation scenarios. In each scenario, the performance of PCADS-AD is evaluated by using real and simulated data, statistical measures and comparisons with other methods.

4.1 Scenario 1

In this scenario, the performance of PCADS-AD is evaluated regarding the traffic characterization phase used to create the DSNSF, and the anomaly detection accuracy. Also, the Reporting Stage is demonstrated.

4.1.1 Data Set

All experiments were performed using real flow data in order to test whether the proposed system can operate in a real environment. Traffic flows were collected from a core switch of the State University of Londrina (UEL) from September 10th to November 9th 2012. The university network consists of 5000 equipment and 3000 wireless devices with a core type Extreme BD 8801. It is used NFDUMP tools to collect and export traffic flows from the core switch. Flows are exported using a 1:256 sampling rate in the version 5 of sFlow format. This protocol is widely used in high-speed networks (gigabit or above) due to its sampling mechanism [34]. The Core Switch is the University main router, where the entire aggregate traffic from border segments is concentrated.

Due to sampling in the sFlow flow extraction process, the general traffic flows extracted for both packets/s and flows/s of the university presents a total correlation that will be observed in the experiments performed. This is because each flow is an aggregation of a number of packets, and according to the use of a 1:256 sampling rate, each flow aggregates 256 packets, ensuing a similar behavior between these two attributes.

4.1.2 Evaluation Metrics

The effectiveness of the digital signatures (DSNSF) were measured by submitting it to two experiments: Normalized Mean Square Error (NMSE) and Correlation Coefficient (CC). The NMSE test measures the differences between time series predicted by a model and what was really observed. This metric has 0 (zero) value as its optimal measure [158]. The Correlation Coefficient indicates how much two objects are associated. In this metric, the value 1 indicates total correlation, 0 indicates that two objects are not correlated, and -1 specifies an inverse correlation [159].

The detection rate was evaluated by using Accuracy Rates and the Receiver Operating Characteristics (ROC) graph. A ROC graph is defined as a technique to measure the performance of classifiers and is widely used in signal detection theory to describe the trade-off between hit rates (true-positive rates) and false alarm rates (false-positive rates). TPR (true-positive rate) describes correctly detected signals, while FPR (false-positive rate) describes how often a signal was wrongly detected. Then, the Accuracy measure describes the overall hit rate, i.e., the hit rate of both classes (positive and negative) [160].

4.1.3 DSNSF Creation

The methodology of the proposed system assumes that a DSNSF is generated for each workday, based on the network activity history of the previous weeks [38] [39]. However, before evaluating the characterization, we aimed to find out the number of weeks in which PCADS-AD would achieve better results in generating the DSNSF. Figure 4.1 shows the NMSE of the DSNSFs created for bits, packets and flows using from one to ten weeks prior to the second week of November (from 11/05 to 11/09). As can be seen in Figure 4.1 (a), for the traffic of bits/s, PCADS-AD started to produce better NMSE indices using five weeks. Above five weeks, the results are almost the same. For the traffic of packets/s and flows/s presented in Figure 4.1 (b), there was a divergence in the results by using one to four weeks, but the NMSE indices of the five analyzed days started to steady by using five weeks.

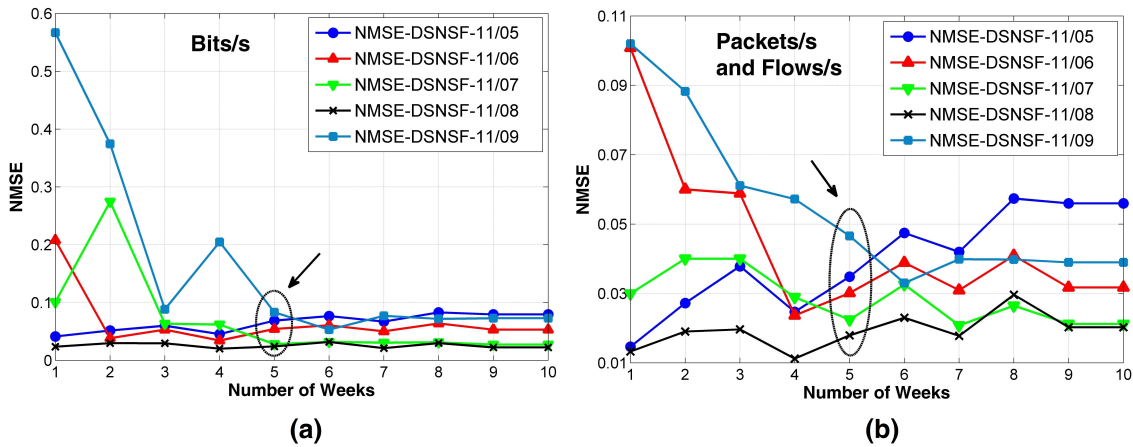


Figure 4.1: NMSE indices over the generated DSNSFs and the real movement of analyzed days using from 1 to 10 weeks

Hence, through this analysis, the data set was divided into two groups: The workdays of the first five weeks are used by PACDS-AD as historical information for DSNSF creating, and the last four weeks for system evaluation.

4.1.4 Traffic Characterization Evaluation

Figure 4.2 illustrates the DSNSFs for the traffic of bits, packets and number of flows transmitted per second generated by the traffic characterization phase. They were compared with the real traffic of the university (in green) observed during the 24 hours of a day. By analyzing the figure, the digital signature curves of the three attributes could efficiently estimate the normal behavior of the network, as a great adjustment between the DSNSF and the real traffic

Chapter 4. Performance Evaluation

can be observed. During the period from 3:30 to 5:30 a.m. there was a disparity between the DSNSF and the real traffic due to a particularity of the university network behavior. Every day a backup is made at the university during that period, resulting in an excessive amount of traffic. Since the backup was performed during the past weeks but not during that period presented in the figure, the traffic characterization algorithm learned that behavior. By the time the backup was no longer made, the DSNSF rapidly adapted to the new network behavior without the backup.

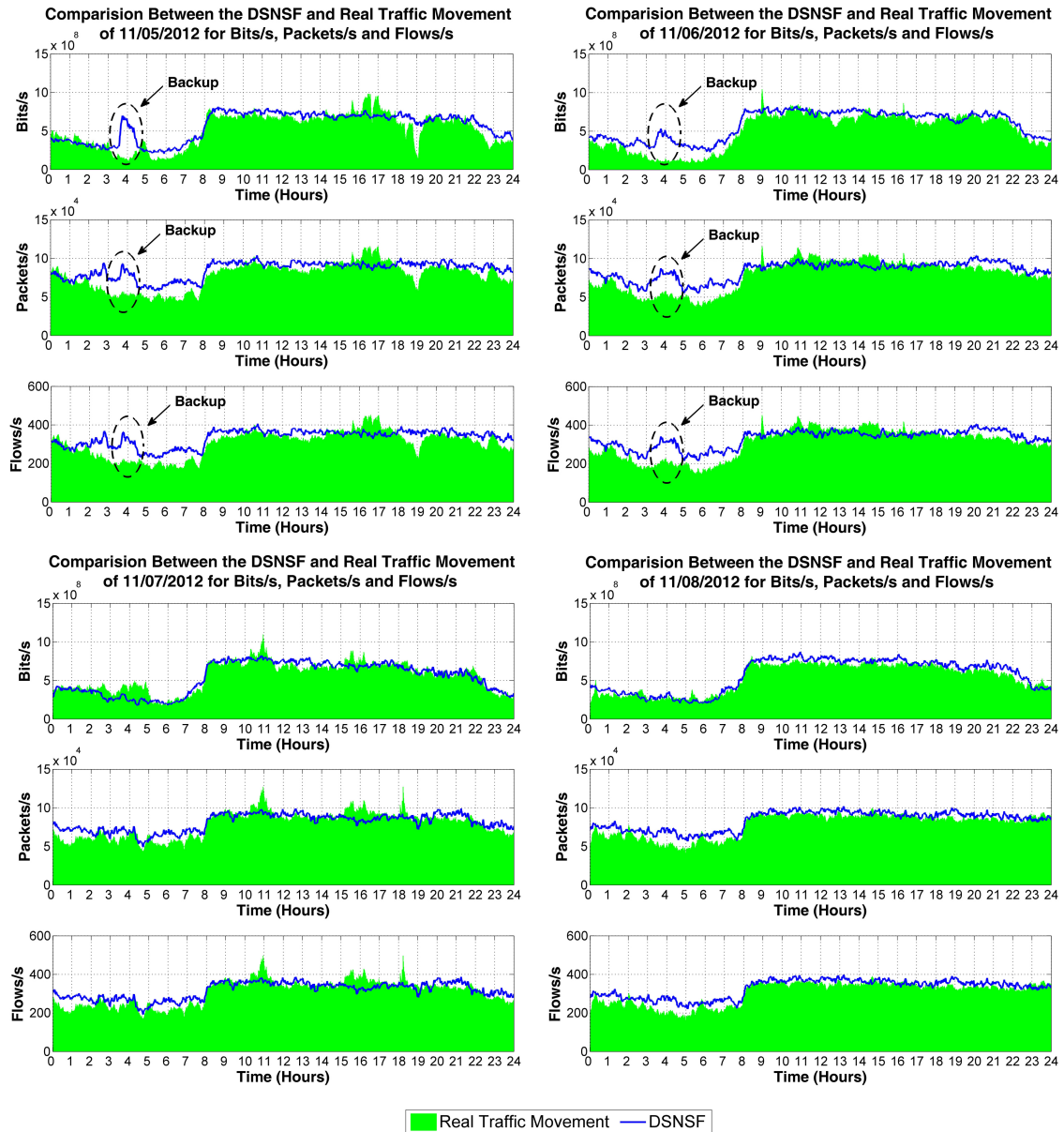


Figure 4.2: Traffic Characterization example comparing the DSNSFs of bits, packets and number of flows transmitted per second generated for four days in November 2012

The first evaluation metric used was the Normalized Mean Square Error, and results are presented in Figure 4.3. The presented system showed good results for all the three attributes, obtaining error indices closer to zero (bellow 0.1). In some cases, the error is higher because October 15th and November 2nd are national holidays, resulting in a network activity different from the usual behavior.

Figure 4.4 presents the results when calculating the Correlation Coefficient (CC) over the DSNSFs and the Real traffic. Figure shows that the correlation test for the DSNSF of Bits/s was better than for Packets/s and Flows/s, but all had good results, with an average of 0.9 for bits/s and an average of 0.7 for Packets/s and Flows/s.

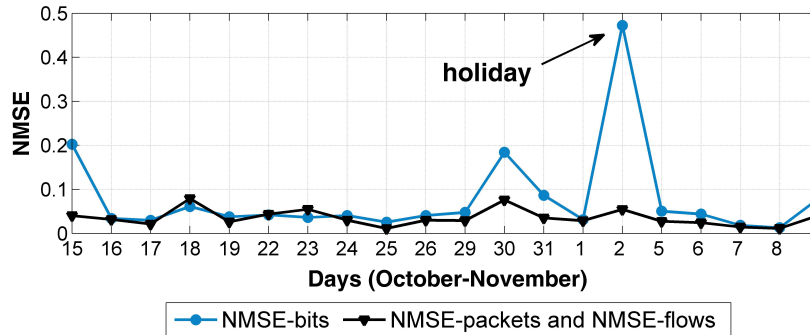


Figure 4.3: NMSE tests between the generated DSNSFs and the real traffic from October 15th to November 09th

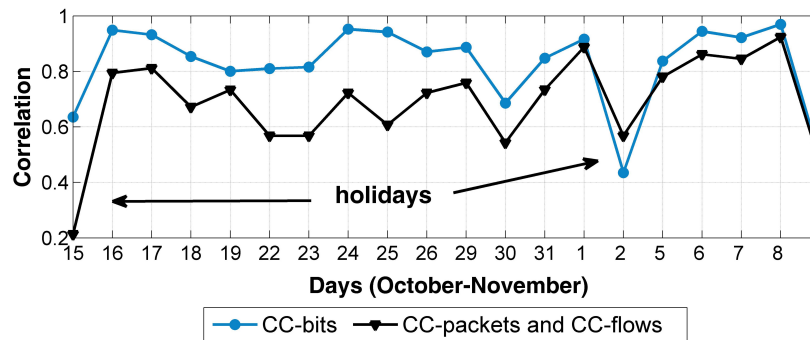


Figure 4.4: Correlation tests between the generated DSNSFs and the real traffic from October 15th to November 09th

4.1.5 Anomaly Detection Evaluation

To properly evaluate the anomaly detection system, it was used a tool to artificially inject anomalous events in the real traffic. Anomalous situations simulated in the data set by using a tool named Scorpius [161], developed by our network research group, which simulates network anomalies in real traffic flow data, such as DoS, DDoS, and Flash Crowd, used by the network group to help test anomaly detection methods. The anomalies are injected in the collected flow data without real and direct intervention in the network, preserving it from impacts caused by anomalies.

A DoS is a single-source attack that attempts to deny access to shared network services or resources. Generally, it uses a great packet volume containing useless traffic to congest and wastes network resources that could service legitimate traffic, for example, TCP SYN flooding, UDP flooding. When that denial is composed of multi sources, it is called DDoS. In DDoS, populations of network nodes are corrupted with malwares, so that the attacker can manipulate them

Chapter 4. Performance Evaluation

in order to set up his attack against the targeted service, strengthening the attack. Therefore, as it comes from diverse and common network nodes, it makes defense more complicated [162]. According to [163, 164], the affected attributes for DoS/DDoS are packets and number of flows.

A Flash Crowd is defined as large floods in traffic. It occurs when a rapid growth of users attend to access a specific network resource, causing a dramatic surge in server load. Unlike DoS/DDoS attacks, this anomaly consists of legitimate requests, usually an aftermath of mutual reaction to hot events. For example, when a contest result is published on a URL, when there are multiple access on an on-line play-along web site for a popular television program, or an e-commerce web site carries a big sale. Although it is not malicious, if there is not enough time to react and to provide necessary resources to handle the overload demand, those flash events can seriously flood or lead to a complete web service failure [27, 26]. Flash Crowd anomaly affects all three volume attributes studied in this thesis (bits, packets and number of flows) as presented in [164].

Thus, DoS, DDoS and Flash Crowds were simulated in the data set described in section 4.1.1, in order to create a template containing all infected time intervals, aiming to compare it with the alarms generated by the proposed system. Therefore, in Table 4.1 all information and parameters regarding the anomaly simulation is presented. All IP and port numbers are fictitious, created for testing purposes.

Figure 4.5 illustrates the alarm generation for the traffic of bits, packets and number of flows transmitted per second for two days with artificially injected anomalies. The thick blue line is the DSNF, and the thin lines are the lower and higher thresholds, called EL_{down} and EL_{up} , which are calculated using the eigenvalue limit. Any time interval where the real traffic remains inside the area between the thresholds is considered normal, triggering an alarm (red) wherever it deviates from those boundaries. Also, at figure bottom, the time-frames when the corresponding simulated attacks took place are depicted for comparison. The alarms in red shown in the figure are the possible anomalies detected for each attribute distinctly, but according to the proposed approach, the system will only notify the network administrator when an alarm sounds in two or more attributes. It is observed that the system correctly identified the occurrences of anomalous traffic caused by the artificially injected anomalies.

Figure 4.6 shows the ROC graph for the four weeks selected for study. The ROC graph was constructed by calculating the true-positive and false-positive rates for each day based on the labeled anomaly data set that defines what is an anomaly or not in the traffic. Also, the detection using other threshold values to compare with the Eigenvalue Limit was tested. These other values were produced varying the threshold calculated via PCA by 2% and 4% to more and less. Analysis of the curves showed there were few great improvements or losses in the TPR and FPR by varying the eigenvalue limit. PCADS-AD performed well, with 94% true-positive rates in 94% and 23% false-positive rates.

Also, Figure 4.7 presents the calculated accuracy for the same studied period of four weeks and the same threshold comparison performed in Figure 4.8. This measure is the proportion of true results (true-positives and true-negatives) obtained, thereby achieving an accuracy average of 85%. Again, it can be noted that there was little improvement or loss in accuracy results by varying the eigenvalue limit. Thus, the Eigenvalue Limit approach proved to be ef-

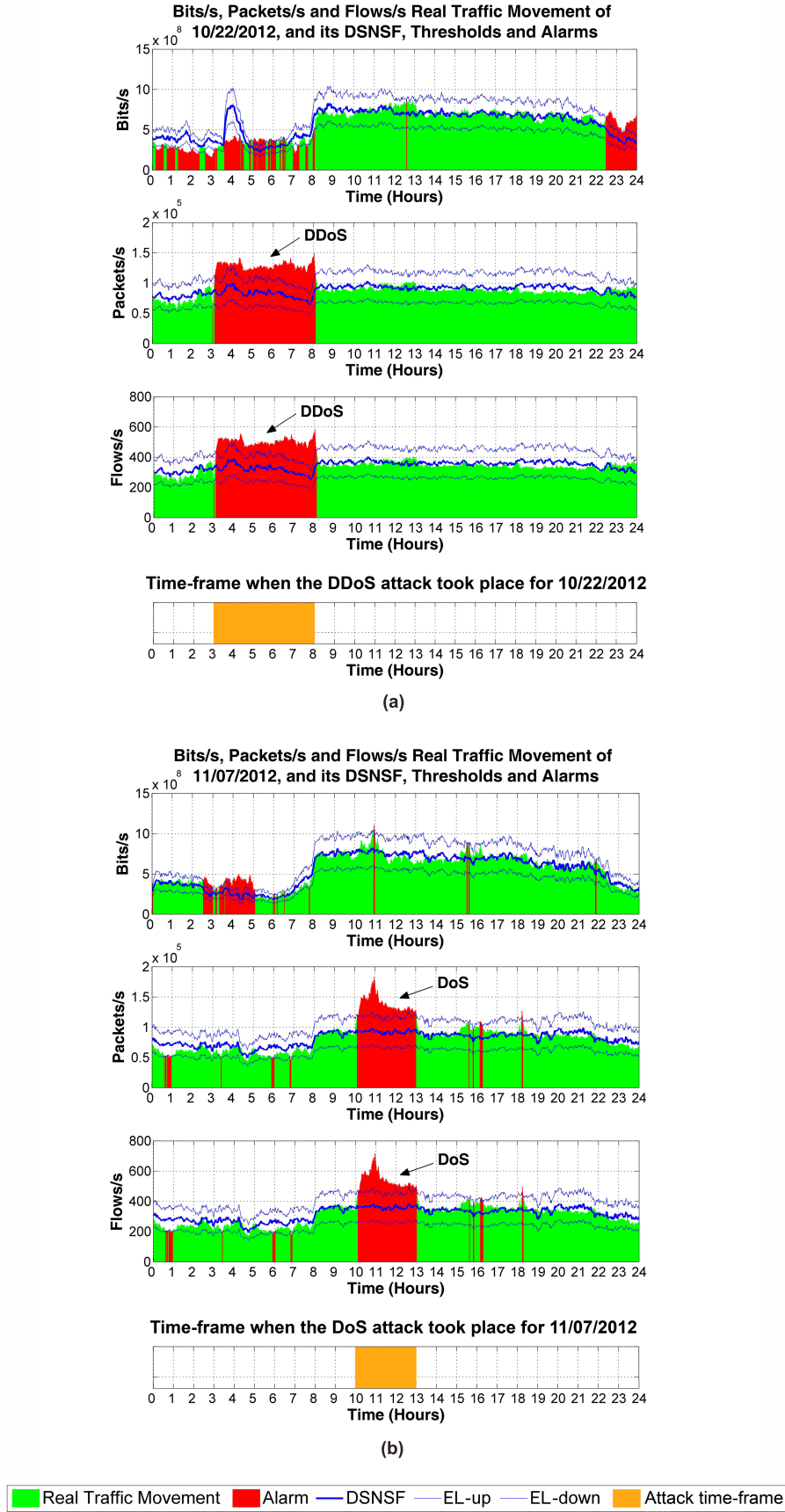


Figure 4.5: Alarm generation example, depicting alarm time-frame and attack time-frame

Chapter 4. Performance Evaluation

Table 4.1: Anomaly simulation parameters using Scorpius tool

Day	Anomaly	Time interval	Source IP	Source port	Destination IP	Destination port	Protocol
10/16/12	DoS	9h - 11h	28.235.160.128	617	83.94.15.23	6008	TCP
10/17/12	DDoS	12h - 15h	10.90.123.456	-	60.89.255.157	6724	ICMP
10/19/12	DDoS	5h - 7h	5.90.123.456	-	180.122.5.131	8188	UDP
10/22/12	DDoS	3h - 8h	5.90.123.456	-	220.151.209.46	3486	TCP
10/23/12	Flash Crowd	0h - 8h	15.90.123.456	-	108.57.76.10	7213	TCP
10/25/12	Flash Crowd	17h - 19h	5.90.123.456	-	12.12.12.12	5055	TCP
10/26/12	DDoS	6h - 10h	7.90.123.456	-	125.127.254.176	2222	TCP
10/29/12	DDoS	16h - 19h	5.90.123.456	-	3.40.112.112	9031	TCP
10/31/12	Flash Crowd	4h - 8h	7.90.123.456	-	148.236.85.173	3446	TCP
11/01/12	DDoS	16h - 18h	10.90.123.456	-	218.209.223.219	2717	TCP
11/05/12	Flash Crowd	15h - 17h	10.90.123.456	-	254.234.112.61	2365	TCP
11/06/12	DoS	5h - 8h	130.232.209.93	7530	67.17.25.44	673	TCP
11/07/12	DoS	10h - 13h	74.110.154.14	8006	122.191.184.218	2346	ICMP
11/08/12	DDoS	9h - 10h	5	-	10.10.10.10	4041	TCP

fective.

After an anomalous situation is detected, the PCADS-AD Reporting Stage can display the qualitative information about the anomalous interval in order to assist the network administrator to take measures to solve the problem. IP addresses and Port numbers are very useful and significant information for accurate and fast anomaly detection. These attributes may reveal where the problem occurred, or who caused it, and also what kind of application was targeted.

To exemplify this module, two days with artificial anomalies was selected from the data set, both using fictitious IP addresses and Port numbers. Table 4.2 summarizes the information about the attack simulation in this testbed.

Figure 4.8 and Figure 4.9 display the PCADS-AD Reporting Stage. Both Figures show the top-3 source and destination IP addresses and Ports that occurred for a time interval when the alarm was triggered, and also a graph showing the traffic anomalous behavior. The Reporting Stage can exhibit the top-N statistics, but it is only used the top-3 because it was enough for these examples.

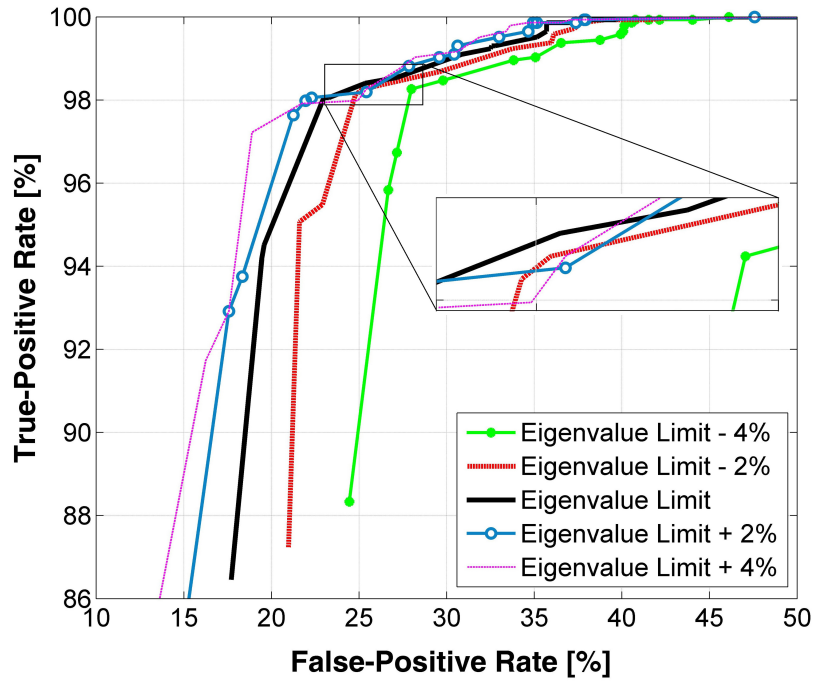


Figure 4.6: ROC graph showing TPR and FPR trade-offs of four weeks of experiments

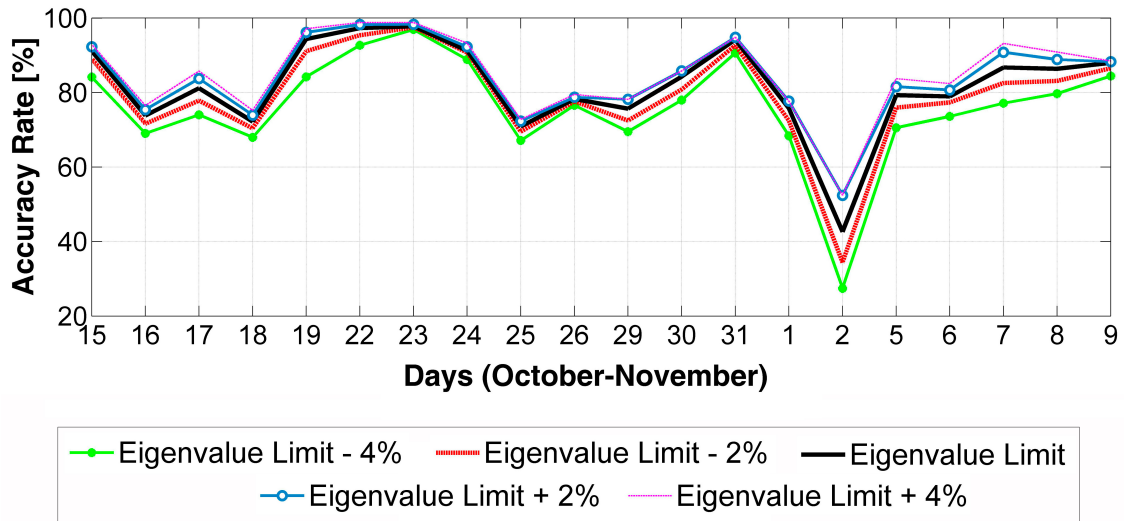


Figure 4.7: Accuracy Rate of four weeks of experiments

By analyzing the results in Figure 4.8 and Figure 4.9, the top-1 destination IP addresses and Ports identified by PCADS-AD were actually the fictitious attributes from Table 4.2 used in the anomaly simulation. It can be verified that an anomalous situation affects a large traffic flow proportion when compared to normal activity not just related to volume, but likewise to qualitative attributes.

Chapter 4. Performance Evaluation

Table 4.2: Anomaly simulation

Day	Anomaly	Time Interval	N° of SrcIP	DstIP	DstPort
10/25/2012	Flash Crowd	17:00pm to 19:00pm	5	12.12.12.12	5055
11/08/2012	DDoS	9:00am to 10:00am	5	10.10.10.10	4041

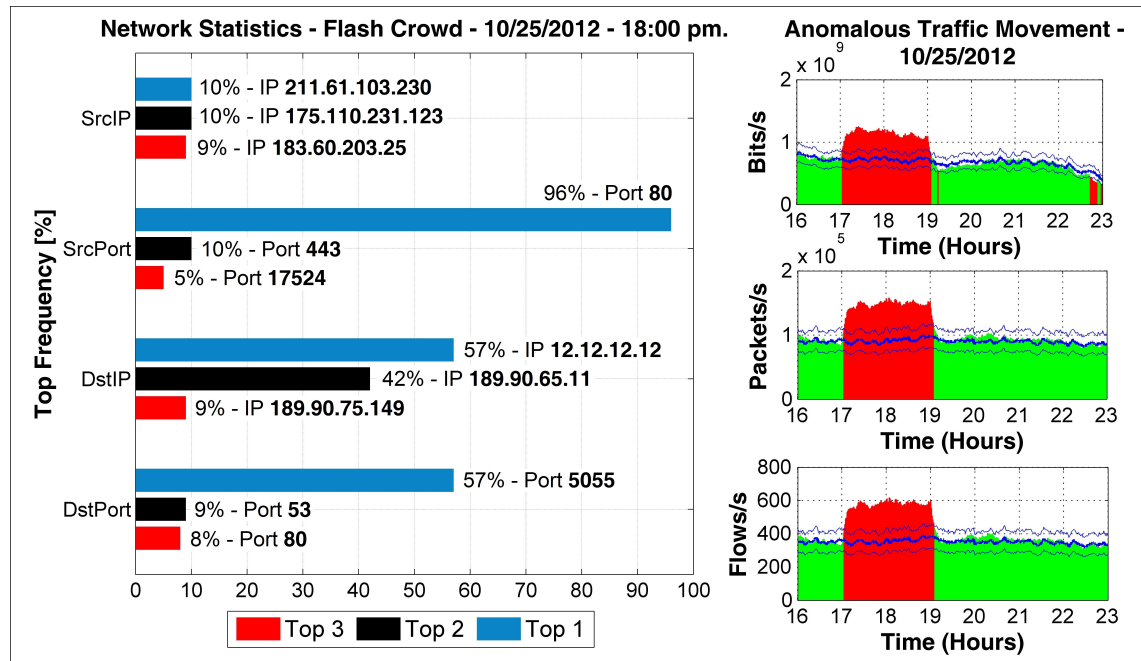


Figure 4.8: PCADS-AD Reporting Stage for Flash Crowd simulation

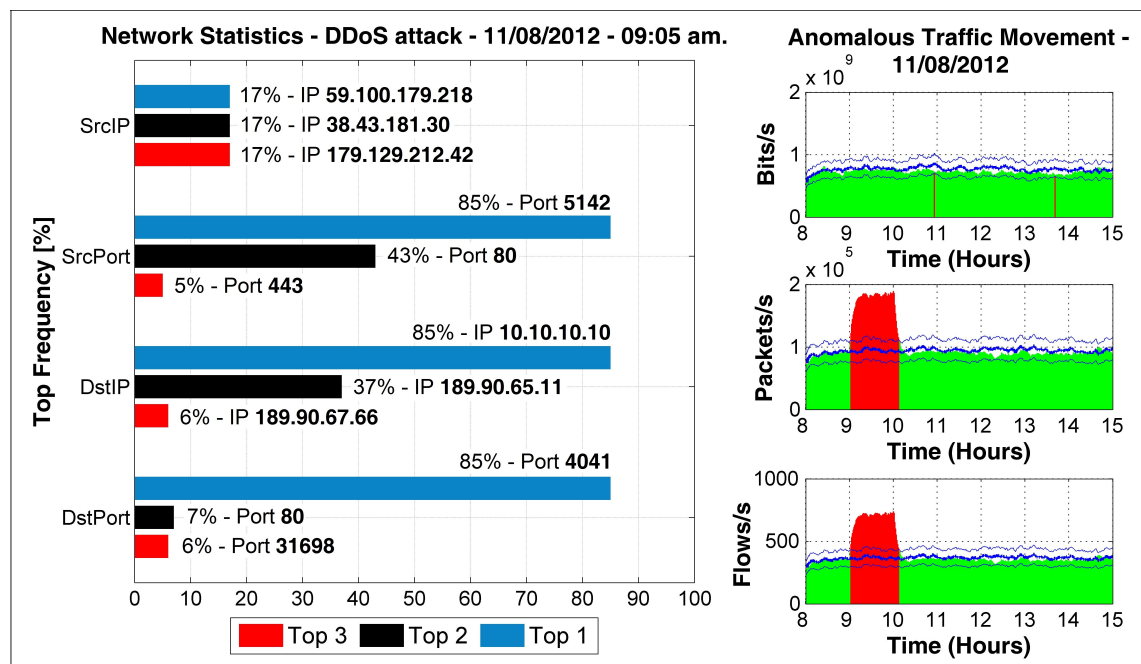


Figure 4.9: PCADS-AD Reporting Stage for DDoS simulation

4.2 Scenario 2

In this Scenario, the proposed PCADS-AD Traffic Characterization Module is compared with ACODS (Ant Colony Optimization for Digital Signature), another traffic characterization method based on clustering.

ACODS [165, 94] is a modification of the Ant Colony Optimization (ACO) meta-heuristic for DSNSF creation using a clustering approach, which is capable of characterizing network traffic through the discovery of a cluster set from the large volume of high dimensional input data. It seeks solutions for grouping data, minimizing the variance between the elements of a given set and maximizing with respect to the other groups, so that it is possible to extract patterns of network measurements. For more details about ACODS, please refer to [165, 94]

However, The detection mechanism is constructed over an adaptation of the pattern matching technique Dynamic Time Warping (DTW) [166], called ADTW (Adaptative DTW). This technique is used to recognize shifted behavior between the DSNSF and real traffic series through time alignment, enabling improved analysis of sudden events and those that occur along the time. This aims to improve the accuracy and reduce false alarm rates in anomaly detection. The ADTW is introduced below.

4.2.1 Adaptive Dynamic Time Warping

In order to find traffic behaviors which are different from DSNSF, a similarity measure should be adopted. The Euclidean distance between each point of the same index has been widely used in time series for this purpose [167]. However, this metric is not suitable for identifying shifts in data sequence. Thus, given two time series, one of them shifted on the time axis, it is possible for the calculation of the Euclidean distance to consider totally different series. Believing that normal traffic behavior can suffer such displacements due to the changes in the schedule of users' activities, an adaptive similarity measure to fit these situations was developed.

Dynamic Time Warping (DTW) is a pattern matching technique widely used in speech recognition utilized to find an optimal alignment between two series, where one may present alterations being partially elongated or shortened relative to other, along the time axis [166]. Assuming the analysis of DSNSF series denoted by $X = x_1, x_2, \dots, x_n$ and real traffic series $Y = y_1, y_2, \dots, y_m$, the DTW result can be given by a correlation factor between the two series, calculated after alignment. For this measure, when result is closer to zero, the input sequences are more similar. Another way to obtain the DTW result is by a graphical representation, provided using a matrix of size $n \times m$ where the axes denote the analyzed series. Using this approach, the algorithm creates an optimal path alignment, ω , between the input sequences, minimizing the distortion D expressed by Equation 4.1.

$$D = \sum_{n=1}^{nm} d(X(n), Y(m)), \quad (4.1)$$

where $m = \omega(n)$ and each element (i, j) contains the distance d between the points (x_i, y_j) , calculated as observed in [166]. The DTW calculation of the optimal alignment to com-

Chapter 4. Performance Evaluation

pare the time series using is given by four basic steps:

Step 1: Create the solution matrix S , which must consists of n rows and m columns, wherein each element in row i and column j represents the modulus of difference between each interval of comparative series, since n corresponds to the DSNSF length and m corresponds to the length of the time series that describes the real traffic.

Step 2: Establish the Accumulated Distance matrix (DA), composed of n rows and m columns. This matrix is given by the sum of its own values with the upper element of the solution matrix, as shown in equation Equation 4.2.

$$AD_{i,j} = AD_{i-1,j} + S_{i,j}, \quad \text{for } i > 1, j > 1 \quad (4.2)$$

Step 3: Create of the dimensions movement matrix, composed of n rows and m columns. This matrix must be initiated by assigning the value zero to the last element of the first column. Therefore, an iteration towards bottom-up should be performed in AD matrix, to know which is the lowest value. If the lowest value is below the current element of the iteration, the movement matrix must be filled with value 1; if the lowest value in AD is on the left, the matrix motion should receive value 3. Finally, if the smallest value is in the left inferior diagonal or the values are equal, the attributed value to the movement matrix current element is 2.

Step 4: Create the best path matrix w . For this purpose, the movement matrix is analyzed from the last element of the first row. Therefore, it is selected element with a smaller distance, d , between other elements, as suggested by Equation 4.3.

$$d = \min(|w_{i,j} - w_{i-1,j}|, |w_{i,j} - w_{i,j-1}|, |w_{i,j} - w_{i-1,j-1}|), \quad \text{for } i > 1, j > 1 \quad (4.3)$$

The Adaptive Dynamic Time Warping (ADTW) approach for anomaly detection is performed at preset time intervals of one minute and consists of two steps. The first one comprises the similarity calculation, S_t , between real traffic and DSNSF at time interval t , using the conventional DTW algorithm. Even small shifts in a series are verified, the results indicate a good match between them because of the time alignment. Until then, only the correspondence found between the shapes of analyzed time series were verified.

In the second step, the distance between the series is calculated, Δ_t , considering their amplitudes. Thus, a subtraction between the average values of both time series is made at the same interval t , as shown in Equation 4.4. The result used in the detection of significant changes in network traffic with respect to normal model is calculated normalizing the multiplication between vectors S and Δ , as shown in equation Equation 4.5.

$$\Delta_t = \bar{Y}_t - \bar{X}_t, \quad (4.4)$$

$$R = \frac{S \times \Delta}{\max(S \times \Delta)}, \quad (4.5)$$

in which $\Delta = \Delta_1, \Delta_2, \dots, \Delta_t, \dots$ and $S = S_1, S_2, \dots, S_t, \dots$. The goal is to provide a measure based in both form and distance of the series in which they are complementary, e.g., it may be that the result of S_t is close to zero, but the distance between the series at interval t is accentuated. It could be a consequence of a failure or misconfiguration, affecting the normal use of the network, since traffic presents normal behavior but a different intensity. Figure 4.11 (a) exemplifies analysis by ADTW for comparing the series, in contrast to the approach of the Euclidean distance shown in Figure 4.10 (b).

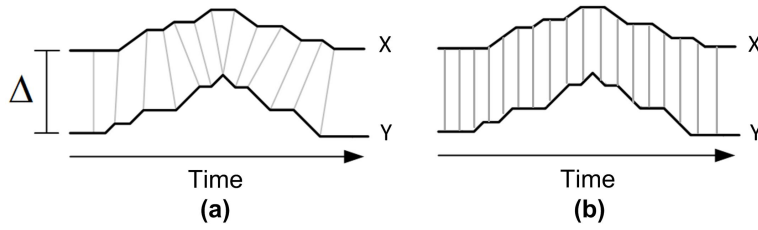


Figure 4.10: Comparison schemes of two time series: (a) by using ADTW and (b) by comparing time series using Euclidian distance.

To improve the detection system efficiency, real traffic movement and DSNSF are evaluated in the same time window t , which comprises a one-minute interval. This approach allows recognition of both punctual anomalies as those which occur over time. Additionally, only an alarm is generated in a time window, ensuring that the administrator is alerted only in event of situations which actually deserve attention.

The flow attributes are analyzed separately, checking the correspondence with the DSNSF created for each of them. A significance coefficient $\phi = 20\%$ is used as threshold for error between the real traffic and DSNSF at interval t , i.e., R_t . This value is set to compensate for possible inaccuracies occurred during the calculation of r , as well as the small variations of the legitimate use of the network. Moreover, the choice of this value occurred by checking several other thresholds, and this proved to be the most suitable for the proposed application, as can be seen in the session results.

4.2.2 Performance evaluation of PCADS x ACODS

Aiming to validate whether the proposed methods can operate in a real network environment, we collected IP flows from a core switch at State University of Londrina (Brazil) network, which is composed of about seven thousand interconnected devices. Due to the large traffic volume, a sampling rate 1:256 was used, implemented by the collection protocol sFlow [168].

Chapter 4. Performance Evaluation

The collection period comprises two months, starting on September 10th and ending on November 09th, 2012. To ease the evaluation, the data set was separated in two groups: The traffic data of first weeks were used by ACODS and PCADS as historical information for DSNSF creating and the workdays of following period - from October 15th to November 09th - was used for traffic characterization and anomaly detection evaluation.

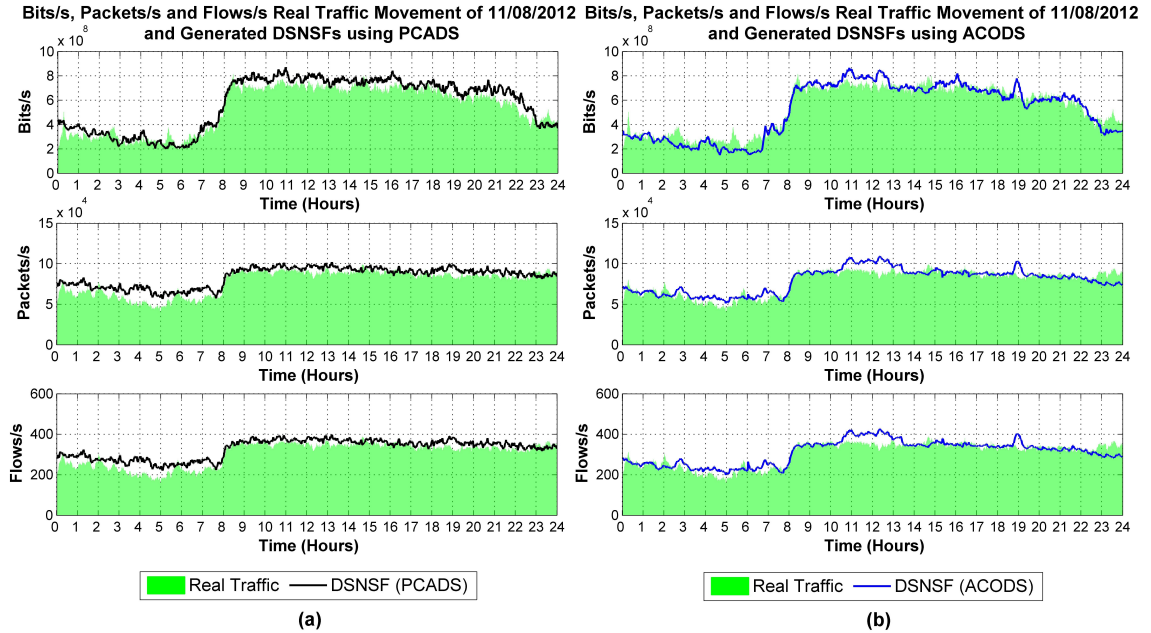


Figure 4.11: Traffic Characterization example comparing the DSNSFs of bits, packets and number of flows transmitted per second with the real traffic movement observed at November 08th for both PCADS (a) and ACODS (b) methods

Figure 4.11 illustrates the DSNSFs of November 8th for the three traffic attributes studied in this thesis compared with the real traffic observed, each of them describing the 24 hours of the day. As we can observe, the digital signature curves generated by both PCADS (Figure 4.12 (a)) and ACODS (Figure 4.12 (b)) could estimate efficiently the normal behavior of that network segment, as there is a great adjustment between the DSNSF and the real traffic.

To measure the accuracy of each method on DSNSFs generation, we adopted two different evaluation metrics: Normalized Mean Square Error (NMSE) and normalized Correlation Coefficient (CC). The Normalized Mean Square Error (NMSE) [158] evaluates the difference between the expected and what was actually verified. This measures' limit is the value zero, which indicates the situation where the expected value is exactly equal to the verified. Thus, higher values of this metric indicate more distant results from the expected.

The normalized Correlation Coefficient [159] indicates the degree of correlation between two variables, as well as the direction of this correlation (positive or negative). The values obtained are within the range of -1 to +1. Value 1 indicates total correlation, score 0 (zero) shows that the two variables are not correlated, and -1 specifies a full inverse correlation, that is, where a variable increases, the other decreases and vice versa.

Figure 4.12 (a) depicts results obtained using the NMSE metric for the traffic of bits/s.

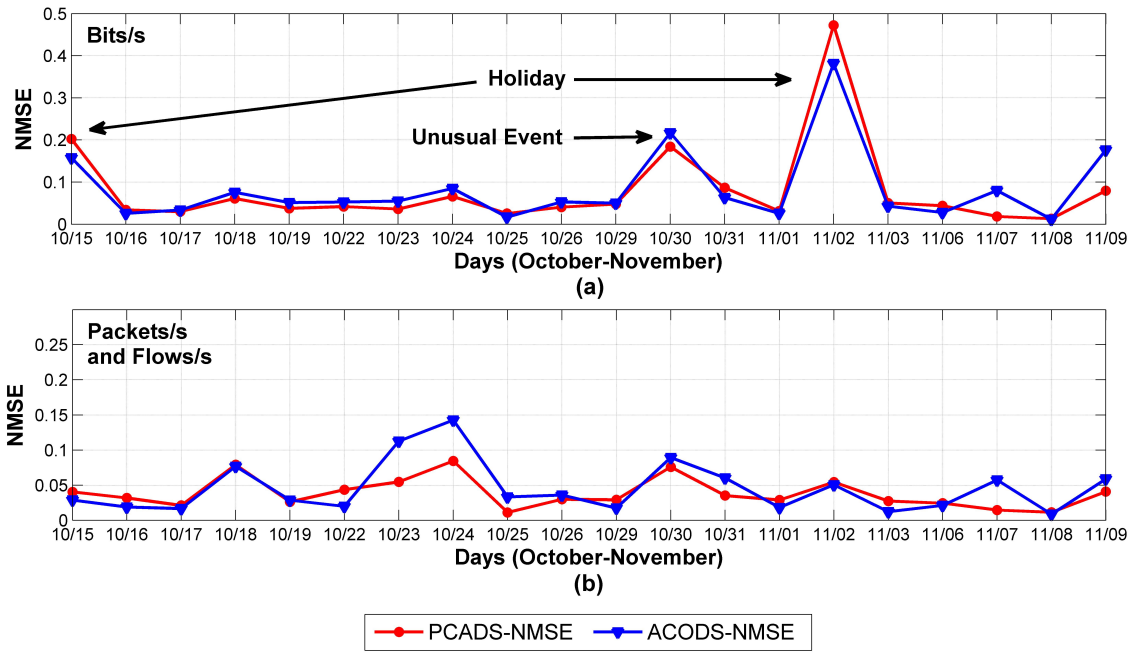


Figure 4.12: NMSE indices between the generated DSNSFs and the real traffic movement of analyzed days

As can be seen, PCADS and ACODS achieved similar results, obtaining small errors. October 15th and November 2nd presented accentuated errors since they are national holidays, where the network traffic behavior differs from its normal pattern. Another anomalous behavior can be noted in October 30th, in which a large traffic volume is observed in all flow attributes analyzed, with peaks up to 56% in excess of the traffic forecasted by DSNSF. It is due to the result of a public tender of the university, which was released on this day, causing a large number of accesses to the university server. Likewise, Figure 4.12 (b) exhibit NMSE outcomes for the traffic of packets/s and flows/s, and we can observe that both methods reached low errors and resembling results again.

Now, Figure 4.13 shows the results relating to bits/s pointed by normalized Correlation Coefficient. The results for bits/s (Figure 4.13 (a)) ranged between 0.8 and 1 for the two methods. Furthermore the results relating to packets/s and flows/s (Figure 4.13 (b)) have lower correlation values, achieving a mean of 0.7. Both PCADS and ACODS produced similar results, which can be classified as strong correlation, as it can be observed in [159], where authors points out that it occurs in cases where correlation coefficient values are above 0.7.

To properly evaluate the proposed anomaly detection system, it was used a tool to artificially inject anomalous events in the real traffic. We simulated anomalous situations in the data set by using a tool named Scorpius, which was developed by a network research group [161]. Scorpius is a tool which simulates network anomalies in real traffic flow data, like DoS, DDoS, Port Scan and Flash Crowd, used by the group to help in testing anomaly detection systems. The anomalies are injected in the collected flow data without real and direct intervention in the network, preserving it from impacts caused by anomalies. Thus, it was simulated DoS, DDoS and Flash Crowds in the real data set in order to create a template containing all infected time intervals, aiming to compare it with the alarms generated by the proposed system.

Chapter 4. Performance Evaluation

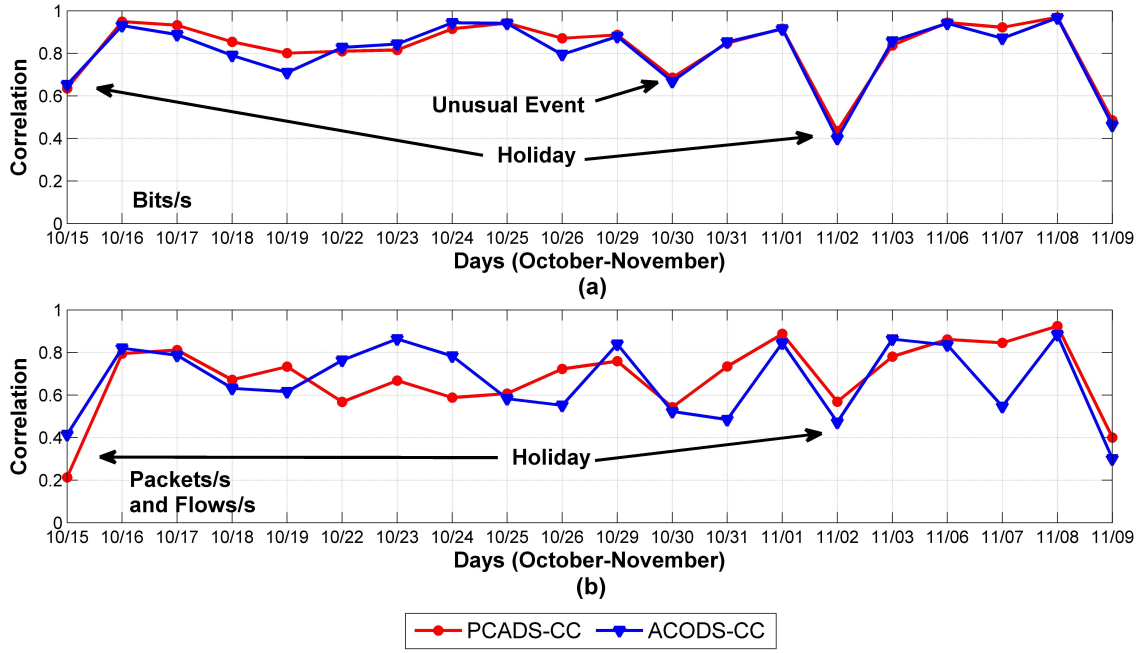


Figure 4.13: Correlation Coefficients between the generated DSNSFs and the real traffic movement of analyzed days

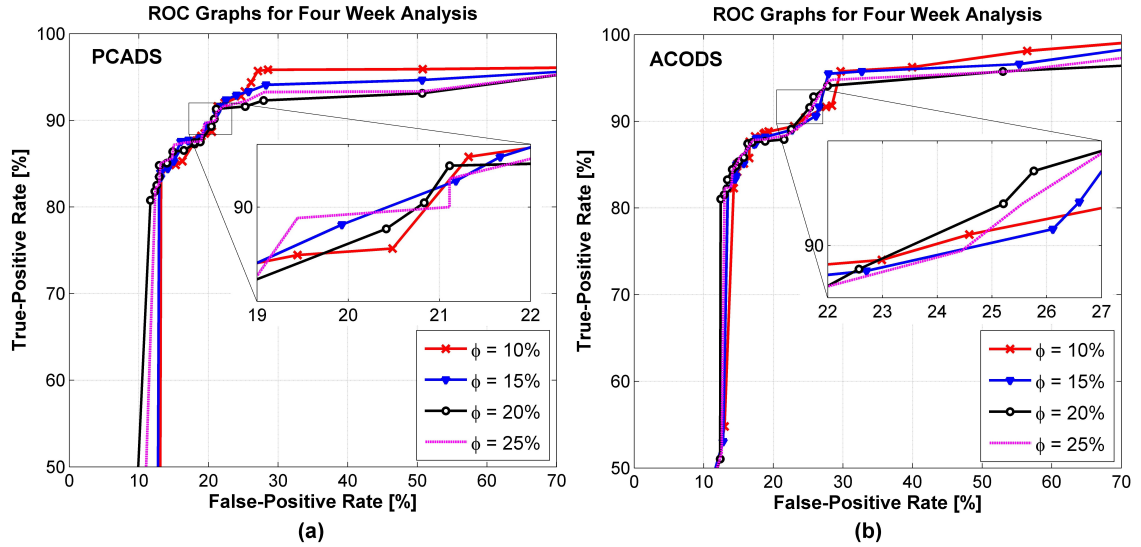


Figure 4.14: ROC curve of workdays from October 15th to November 9th for both PCADS and ACODS using different ϕ values

In order to measure the overall efficiency of proposed detection system, it was used the Receiver Operating Characteristics (ROC) graph and the Accuracy measure. A ROC graph is defined as a technique to measure the performance of classifiers, being widely used in signal detection theory to describe the trade-off between hit rates (true-positive rates) and false alarm rates (false-positive rates). TPR (true-positive rate) describes correctly detected signals, while FPR (false-positive rate) describes how often a signal was detected wrongly [160]. Then, the Accuracy measure describes the overall hit rate, i.e., the hit rate of both classes (positive and negative).

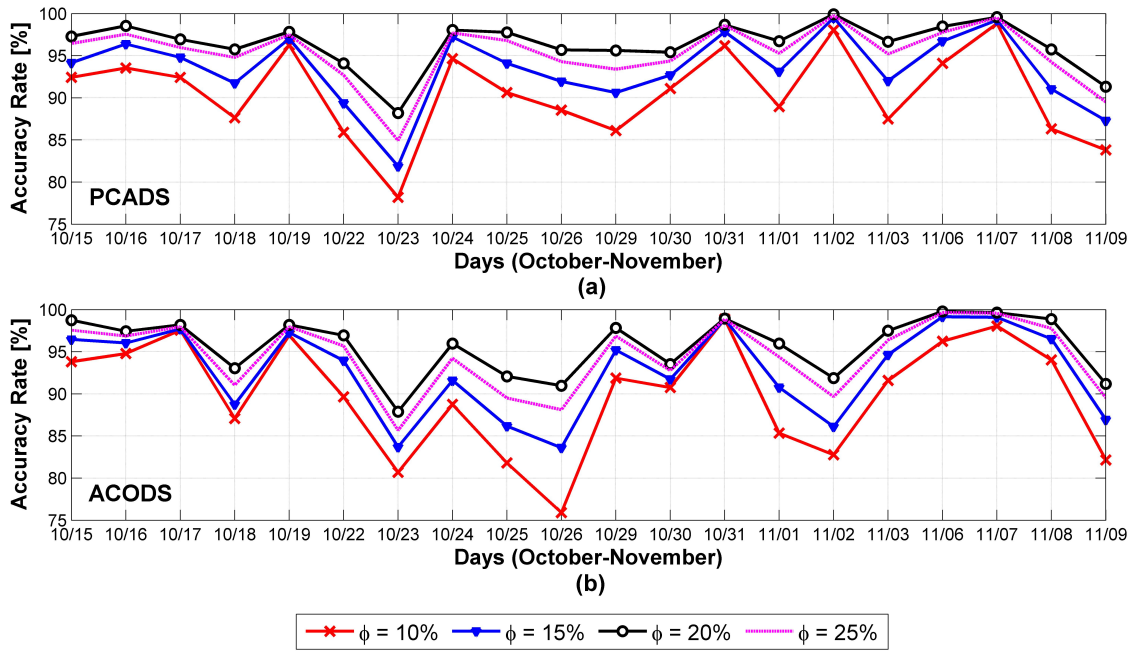


Figure 4.15: Accuracy Rate of four weeks of experiments for both PCADS and ACODS using different ϕ values

Figure 4.14 shows the ROC graphs calculated for both PCADS (Figure 4.14 (a)) and ACODS (Figure 4.14 (b)) for the four weeks selected to study (from October 15th to November 9th) with the artificial anomalous behaviors. Also, the detection was tested using different ϕ values, aiming to verify which is the best threshold for detecting events that differentiate from the DSNSF. As seen, the proposed approach was able to recognize a higher percentage of intervals containing anomalous traffic behavior using smaller values of coefficient significance. It is important since ϕ cannot be large enough so that anomalous behaviors are classified as normal, because only anomalies which cause great impact on the flow attributes behavior would be rec-

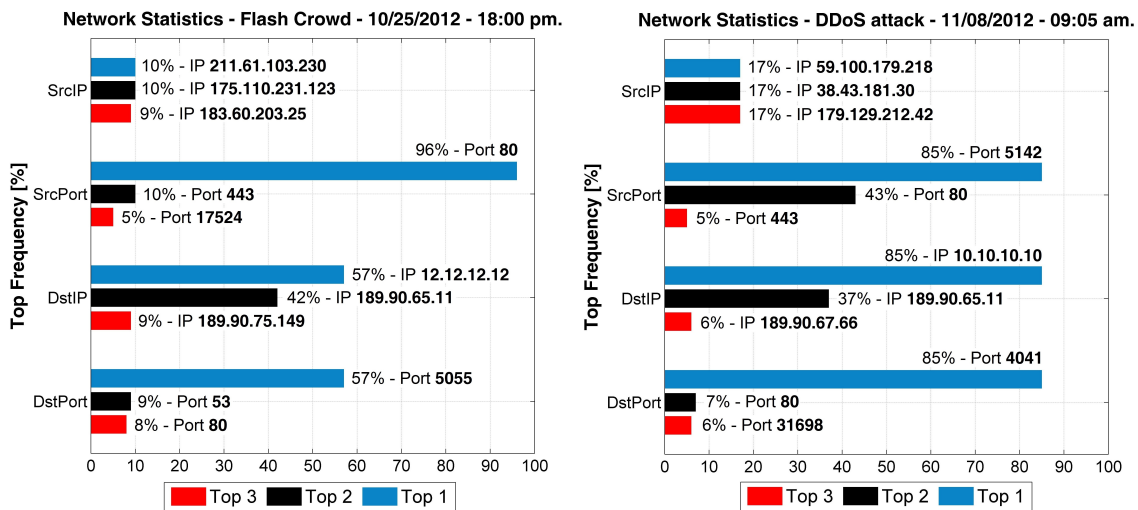


Figure 4.16: Network traffic statistics from two kinds of anomalies

ognized.

Several other values were examined for ϕ ; however, it is clear that values lower than 15% have worse results for TPR and FPR. This occurs due to the reduction of threshold, allowing minimal traffic variation in relation to DSNSF, including legitimate behaviors, to be wrongly characterized as anomalies. On the other hand, when this value is higher than 20%, anomalies are detected only when their behavior deviates greatly from the traffic pattern established by DSNSF. Through the ROC curve analysis, it can be inferred that such a situation causes lower rates of true-positive, and this deficiency is enhanced while the value of ϕ increases. In general, as can be seen in the zooms in Figure 4.14, the one based on PCADS performed better than ACODS, with reduced false-positive rates, reaching a trade-off of 92% TPR with 21% FPR, as ACODS reaches 92% TPR with 24% FPR.

In Figure 4.15, the Accuracy measure for the same study period of four weeks and the same comparison using different ϕ values performed in Figure 4.14 is shown. This measure is the proportion of true results (true-positive and true-negative). Both systems produced better accuracy rates when $\phi = 20\%$, perceiving that for ϕ values above or below this threshold, the accuracy rate begins to decrease. So, $\omega = 20\%$, both systems achieved worthy results, obtaining an average accuracy rate of 96%.

After an anomalous event is detected, the detection system can provide the network manager a detailed report about that time interval, containing information like IP source and destination addresses, and origin and destination ports. These descriptive attributes may unveil where the problem ensued, or who caused it, as well as what kind of application was targeted.

To demonstrate this service, we selected two days with artificial anomalies from the actual data set, both using fictitious IP addresses and Port numbers. Figure 4.16 presents a top 3 list of traffic statistics of a Flash Crowd (a) and a DDoS attack (b). The top 1 destination IP addresses and Ports identified by the proposed system are actually the fictitious attributes used in the anomaly simulation. It can be observed that an anomalous situation affects a large traffic flows proportion when compared to normal activity not just related to volume, but likewise in descriptive attributes.

4.3 Scenario 3

This Scenario focuses on analyzing and comparing three anomaly detection systems founded on distinct algorithm classes, and they are: the statistical procedure Principal Component Analysis for Digital Signature (PCADS-AD) proposed in this Thesis; the Ant Colony Optimization for Digital Signature (ACODS) metaheuristic [165, 94]; and the forecasting method AutoRegressive Integrated Moving Average for Digital Signature (ARIMADS) [85, 86]. These methods potentially exhibit extremely diverse behaviors, therefore, by exploring its own characteristics, seeking to determine which yields better results regarding detection rate, runtime and complexity.

PCADS-AD [15, 78] is developed under a different interpretation of Principal Component Analysis (PCA) multivariate statistical procedure. Its pattern recognition and dimensional-

ity reduction features enables the reduction of an initial large traffic dataset to only network traffic intervals which can be used to efficiently represent the normal behaviour of a network segment and create the DSNSF. Also, PCADS analyses eigenvectors and eigenvalues to create thresholds for anomaly detection and alarm generation.

ACODS [165, 94] is a modification of the Ant Colony Optimization (ACO) metaheuristic for DSNSF creation using a clustering approach, which is capable of characterizing network traffic through the discovery of a cluster set from the large volume of high dimensional input data. It seeks solutions for grouping data, minimizing the variance between the elements of a given set and maximizing with respect to the other groups, so that it is possible to extract patterns of network measurements.

ARIMADS [85, 86] is a training-based forecasting model focused on temporal processes. It uses the ARIMA model for investigating normality and linear trends in network traffic and then builds a Digital Signature of Network Segment related to series of traffic features. By using ARIMADS prediction capabilities, it is possible to accurately establish normal patterns for high-speed network traffic.

These models perform a six-dimensional flow analysis by extracting normal patterns of traffic features from a real data set. They are able to combine information related to the volume of traffic (bits and packets) along with the dispersion of the IP addresses and ports used during the communication process (entropy).

Although they share the same methodology of creating a digital signature and dividing tasks into two well-defined categories, characterization and detection of anomalous events, each one implements different routines to fulfill their role. These approaches are established by the class of the algorithm which the method belongs to.

The accurate notion of an anomaly may be different for different application domains. When observing simultaneously multiple characteristics of traffic data, a simple euclidean distance may be unable to perceive similar behavior, even though they are not mathematically alike. For this reason, each system has employed a different approach to evaluate dispersion between real measurements and normal traffic profile.

4.3.1 Data Preparation

Aiming to test whether the systems can operate in a real environment, all experiments were performed using real flow data. Flows were extracted from a core switch of the State University of Londrina (UEL), comprising workdays from October and November, 2012, using the sFlow format. The first month is used to train the systems on generating the DSNSFs of the six analyzed attributes, while a day from November is selected as a specific case study, in order to accomplish a punctual and detailed performance evaluation.

The majority of the approaches discussed in the literature employs five-minute window analysis interval for detecting anomalies. This thesis concerns in characterizing the traffic behavior and detect anomalies throughout the day by considering a time window of one-minute.

Chapter 4. Performance Evaluation

In this manner, there are a total of $T = 1440$ intervals ($24\text{hours} \times 60\text{minutes} = 1440\text{intervals}$) to be investigated for each day monitored. This approach reduces the time required for anomaly notification, which contributes to availability and control of the network.

To properly evaluate anomaly detection rates, it was used Scorpius [161], a tool developed by a network research group, to artificially inject anomalous events in the real traffic, like DDoS and Port Scan. The anomalies are injected in the collected flow data without real and direct intervention in the network, preserving it from impacts caused by anomalies.

Table 4.3: Artificial Anomaly Simulation on 11/08

DDoS			
Start Time	End Time	SrcIP number	DstIP
10:00	13:00	5	145.52.155.26
Port Scan			
Start Time	End Time	DstIP	Port Range
03:00	04:00	70.203.136.78	50 to 40000

DDoS and a Port Scan attacks were simulated in a particular day from the data set, and detailed information about them are shown in Table 4.3. DDoS attack is composed of multi-source generating requests to a single destination. In this experiment, the fictitious IP address 145.52.155.26 received in its port 1170 numerous requests via TCP coming from five different sources. According to Chang *et al.* [169], this anomaly directly affects the behavior of attributes source and destination IP addresses entropy, destination port entropy and transmitted packets.

In a port scan attack, multiple sources send a packet with the SYN flag enabled to different ports of destination, aiming to receive the confirmation whether they are operative. The proposed model scanned the port range 50-40000 of the fictitious IP address 70.203.136.78. The attributes most affected by this activity were IP addresses entropy and destination port entropy.

4.3.2 Evaluation

Figure 4.17 shows the DSNSFs generated for the six analyzed traffic attributes on November 8th, 2012. It is possible to notice that the digital signature generated by the systems are similar, except for certain behaviors. Furthermore, the graphs depict the behavior of traffic during Portscan and DDoS attacks. As can be seen, each of these anomalies affects differently the attributes by changing their distributions, which means that they detract from the normal pattern described by the DSNSF.

Figure 4.18 illustrates a labeling process where hits and false alarms are seen on time. For this study, a group of network specialists have deeply evaluated the traffic data available and properly set an anomaly template for the entire day analyzed. Then, each approach for anomaly detection provides its classification for comparison of results.

The Receiver Operating Characteristics (ROC) graph and the Accuracy measure, trustful evaluation metrics most commonly used in the area, are used in this thesis. A ROC graph is defined as a technique to measure the performance of classifiers, being widely used in signal detection theory to describe the trade-off between hit rates (true-positive rates) and false alarm rates (false-positive rates). TPR (true-positive rate) describes correctly detected signals, while FPR (false-positive rate) describes how often a signal was detected wrongly. Then, the Accuracy measure describes the overall hit rate, i.e., the hit rate of both classes (positive and negative) [160].

Figure 4.19 shows the ROC curve for the selected case study with artificial anomalies. The ROC plot was generated by calculating the TPR and FPR for it, based on the labeled template. By analyzing the zoom part of the figure, it is observed that ARIMADS and ACODS performed better, both achieving similar results, obtaining trade-offs with nearly null FPR. This indicates a lower number of intervals wrongly detected, avoiding the analysis of false alarms by the network administrator. ARIMADS had a trade-off of 98% TPR with 0% FPR, as ACODS reaches 98% TPR with 1% FPR. In contrast, PCADS results diverges from the others. Although its TPR is high, it flashes a high number of false alarms, resulting in a trade-off of 98% TPR with 8% FPR. Concerning the accuracy measure, ACODS and ARIMADS preserved its similarity in results, with 96% and 97% of accuracy, respectively. PACDS had an accuracy of 90%.

The approach used by each system for detection of anomalous intervals is the differential in the results. PCADS creates thresholds for the DSNSF, which although effective, has a static nature, hindering the detection of anomalies, many of which may vary and present dynamic behaviors. ACODS fared better due to ADTW dynamicity, which takes into account the duration and amplitude of the anomalous event. ARIMADS, in turn, uses the training data set as a second verification which based on paraconsistent metrics, showed better accuracy.

Regarding execution time, although there is a meaningful disparity between the systems, all three approaches proved to perform in a timely fashion. In more detail, the required time for traffic characterization and anomaly detection using ACODS is higher than the other systems. It is explained by having a metaheuristic with a robust character intrinsic to its deployment. The two other deterministic systems presented similar runtime; however, PCADS spends less time for both characterization and detection of abnormal events.

4.4 Computational Complexity Analysis

The computational complexity of the whole anomaly detection system proposed in this Thesis (PCADS-AD) is based on the complexity of the traffic characterization phase, since the system is totally based on the DSNSF and the thresholds that are also calculated during the characterization phase. Therefore, for PCADS, computing all the principal components of a given $n \times p$ matrix X , according to Lakhina *et al.* [64], is equivalent to solving the symmetric eigenvalue problem for a covariance matrix X^T . To solve this problem, it is necessary to compute the Singular Value Decomposition (SVD), a method used to obtain the eigenvectors and eigenvalues of a matrix X [170]. Thus the computational complexity of a complete SVD of a $n \times p$ matrix is limited by $O(np^2)$.

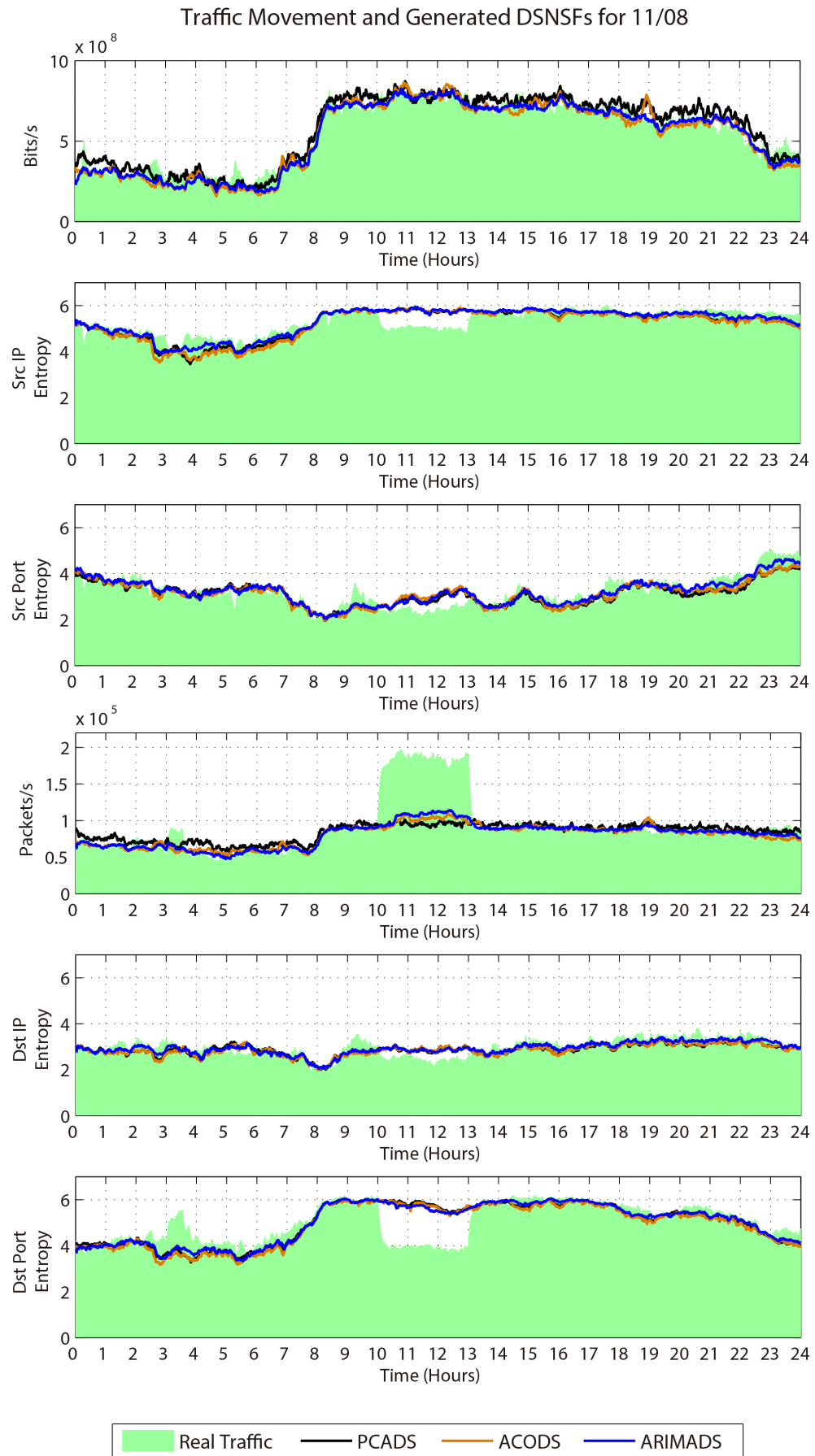


Figure 4.17: Traffic characterization using the proposed methods. The graphs show the prediction calculated for each analyzed attribute and the observed traffic behavior on November 8th, 2012

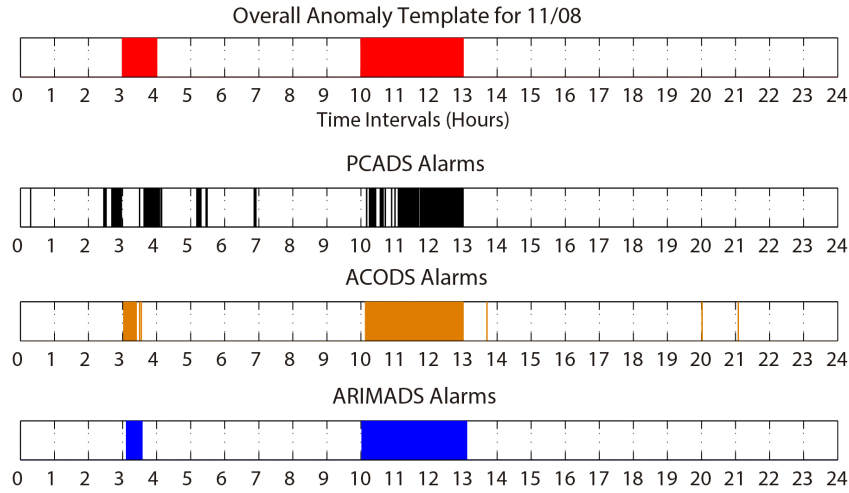


Figure 4.18: General alarm comparison

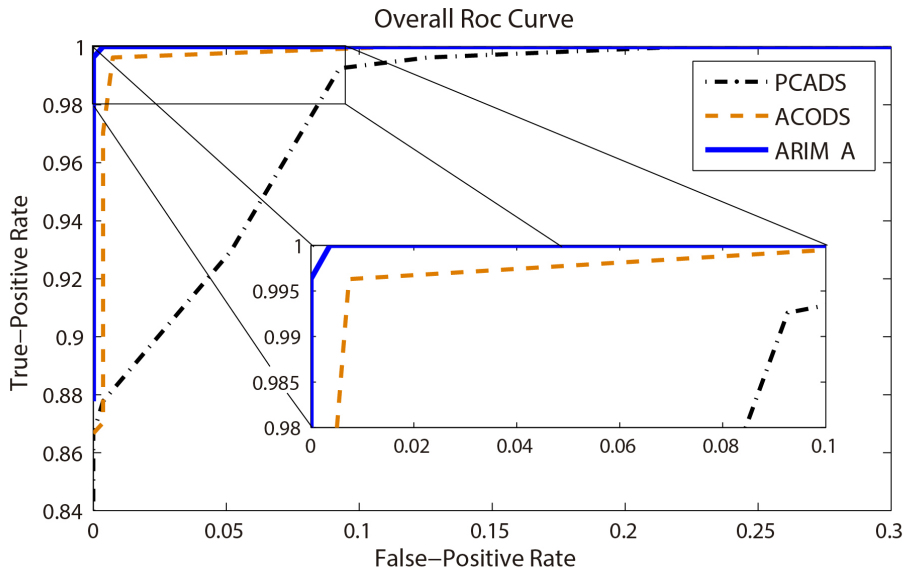


Figure 4.19: ROC curves comparing the trade-off between TPR and FPR rates of the proposed methods

Under a 3.0 GHz Intel-based processor, PCADS-AD performed well, taking less than three seconds to create a DSNSF for bits, packets and flows. This performance indicates the feasibility of the proposed system for real-time anomaly detection with reduced use of computational resources.

Chapter 5

Conclusion

This chapter presents the main conclusions that result from the research work described in this thesis. Furthermore, it discusses several research topics related to the work developed along the doctoral programme that can be addressed in further research works.

5.1 Final Remarks

In this Thesis, a novel autonomous profile-based anomaly detection system to help network management using Digital Signature of Network Segment using Flow analysis (DSNSF) generated via Principal Component Analysis was presented and evaluated. The main contribution consists in the application and contextualization of PCA to an anomaly detection environment using IP flow attributes. The system creates a digital signature (DSNSF) based on the PCA statistical method, exploring its dimensionality reduction feature by applying it over past week traffic, ensuring that such signatures are able to represent the main characteristics and patterns of network traffic. Another contribution is the creation of confidence bands using the eigenvalues obtained in the traffic characterization phase, which states an interval for the DSNSF where traffic variations are considered normal. At last, PCADS-AD Reporting Stage can provide to network administrators useful information about abnormalities found.

In Scenario 1, regarding the traffic characterization for DSNSF creation, the proposed system achieved good results, showing small errors (below 0.1) and good correlation indices (0.8 average) when the DSNSF was compared with the real traffic, showing that it can be a good choice for predicting the expected behavior of a network segment. Now, on the subject of anomaly detection, results pertaining to false alarm and accuracy rate are encouraging, and also, in addition to warning the network administrator about the problem, the proposed system can also provide the necessary information to solve it through the PCADS-AD Reporting Stage.

In Scenario 2, it is presented and evaluated two profile-based anomaly detection systems to help network management. The major contribution consists in the application and contextualization of Principal Component Analysis, Ant Colony Optimization and Dynamic Time Warping methods to an environment of pattern recognition and anomaly detection. It also stands out as a contribution, the analysis of seven IP flow attributes, where: i) three quantitative attributes - bits, number of packets and flows - are used in order to characterize the network traffic through DSNSF generation, a key to effectively identify anomalous behaviors and ii) four descriptive attributes - source IP, destination IP, TCP/UDP source port and TCP/UDP destination port - which are used by the Information Module to provide the network manager information needed to identify the problem and take specific measures against it.

Regarding Traffic Characterization module, iw was compared two different methods, PCADS and ACODS. According to NMSE and Correlation Coefficient results, both accomplished

similar results, leading to good traffic predictions, since small errors can be verified between the DSNSF and the real traffic in Figure 4.11.

In the Detection and Identification module, the Adaptive DTW (ADTW) algorithm investigated in this work had satisfactory performance pertaining to false alarm rates. Concerning that subject, both systems produced better results when adjusting the ADTW ϕ value to 20%. Moreover, by analyzing the ROC graphs and Accuracy rates, PCADS-AD performed better than ACODS. Moreover, the correspondence between true-positive and false-positive rates demonstrates that the systems are able to enhance the detection of anomalous behavior by maintaining a satisfactory false-alarm rate. In addition, as presented in Figure 4.16, the proposed anomaly detection methodology can supply the network administrator with important traffic statistics in order to help in problems solution, aiming for accurate and fast anomaly detection. Therefore, the proposed methodologies, by using PCADS, ACODS and ADTW, is suitable to help network management, detecting traffic anomalies and consequently, supplying availability and reliability to networks and their provided services.

Finally in Scenario 3, the recognition of abnormal events held by three anomaly detection systems was discussed and evaluated. Although each of them belongs to distinct algorithm classes, they aimed to characterize network traffic normal behavior by creating the DSNSF (Digital Signature of Network Segment using Flow analysis).

All the systems produced similar DSNSFs, equally clever in describing the normal behavior of the analyzed network traffic. Accordingly, variations found in the effectiveness of anomaly detection are linked with the engine used for scanning differences between digital signatures and the observed traffic. ARIMADS proved more promising in recognizing abnormalities than the other methods since it uses the DSNSF combined with Paraconsistent Logic for handling the concept of uncertainty. ACODS had an inferior performance compared to ARIMADS on account of false positives reported during analysis. Finally, PCADS-AD achieved the lowest detection rate, which is justified by the adoption of less flexible thresholds for normal network activity provision.

The low computational complexity of the characterization process and the anomaly detection method and the results obtained in the presented tests using real data implies that the proposed approach using Principal Component Analysis shows high applicability for automatic anomaly identification and is a promising step towards a broader system for online diagnosis of anomalies in large scale networks.

Some kinds of attacks and anomalies such as DoS, DDoS and Flash Crowds cause traffic variations in distinct traffic attributes. DDoS, for example, affects only the traffic of packets and number of flows. This thesis also contributes by detecting traffic volume anomalies through the analysis of three IP flow quantitative attributes (bits/s, packets/s and flows/s), aiming for effective detection of different anomalous behaviors.

5.2 Future Work

To conclude this research work, it remains to suggest future study topics resulting from the developed research work:

- To improve PCADS-AD system by minimizing false alarm generation and use other flow attributes from the aggregate traffic, in an endeavor to detect and identify other kinds of attacks and anomalies, like port scans, probing, U2R or R2L.
- To apply and evaluate the proposals of this Thesis on a real-time environment, for their validation and comparison with the results obtained by other similar methods.
- Combine the proposed method with Machine Learning techniques, improving the detection and reducing costs.

References

- [1] F. Hashim, K. S. Munasinghe, and A. Jamalipour, "Biologically Inspired Anomaly Detection and Security Control Frameworks for Complex Heterogeneous Networks," *IEEE Transactions on Network and Service Management*, vol. 7, no. 4, pp. 268-281, dec 2010. [Online]. Available: <http://ieeexplore.ieee.org/document/5668982/>
- [2] L. Song-song, W. Xiao-feng, and M. Li, "Network security situation awareness based on network simulation," in *Electronics, Computer and Applications, 2014 IEEE Workshop on*, 2014, pp. 512-517.
- [3] S. M. Hosseini Bamakan, H. Wang, and Y. Shi, "Ramp loss K-Support Vector Classification-Regression; a robust and sparse multi-class approach to the intrusion detection problem," *Knowledge-Based Systems*, vol. 126, pp. 113-126, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950705117301314>
- [4] A. Lof and R. Nelson, "Annotating network trace data for anomaly detection research," in *Local Computer Networks Workshops (LCN Workshops), 2014 IEEE 39th Conference on*, 2014, pp. 679-684. [Online]. Available: <https://ieeexplore.ieee.org/document/6927720>
- [5] V. Barnett and T. Lewis, "Outliers in Statistical Data," (3rd edition) John Wiley and Sons, New York, 1994. [Online]. Available: <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471930946.html>
- [6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1-58, 2009. [Online]. Available: <https://dl.acm.org/citation.cfm?id=1541882>
- [7] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," *ACM SIGMETRICS Performance Evaluation Review*, vol. 32, no. 1, p. 61, jun 2004. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1012888.1005697>
- [8] N. Hoque, M. H. Bhuyan, R. Baishya, D. Bhattacharyya, and J. Kalita, "Network attacks: Taxonomy, tools and systems," *Journal of Network and Computer Applications*, vol. 40, pp. 307-324, apr 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804513001756>
- [9] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448-3470, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S138912860700062X><http://ac.els-cdn.com/S138912860700062X/1-s2.0-S138912860700062X-main.pdf?tid=655bec9a-69bd-11e4-8a10-00000aab0f26&jacdnat=141572246972a76c30be9750161688672b9aae69fc>
- [10] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 303-336, 2014. [Online]. Available: <http://ieeexplore.ieee.org/ielx7/9739/6734841/06524462.pdf?tp=&arnumber=6524462&isnumber=6734841>
- [11] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based change detection: methods, evaluation, and applications," *Proceedings of the 3rd ACM SIGCOMM conference*

References

- on Internet measurement*, pp. 234-247, 2003. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=948205.948236><http://doi.acm.org/10.1145/948205.948236>
- [12] K. Eunju and H. Sumi, "A Practical Activity Recognition Approach Based on the Generic Activity Framework," *International Journal of E-Health and Medical Communications (IJEHMC)*, vol. 3, no. 3, pp. 54-71, 2012. [Online]. Available: <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/jehmc.2012070105>
- [13] G. Casale, "Combining queueing networks and web usage mining techniques for web performance analysis," Santa Fe, New Mexico, pp. 1699-1703, 2005.
- [14] G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, 2018. [Online]. Available: <http://link.springer.com/10.1007/s11235-018-0475-8>
- [15] G. Fernandes, J. J. Rodrigues, and M. L. Proença, "Autonomous profile-based anomaly detection system using principal component analysis and flow analysis," *Applied Soft Computing*, vol. 34, pp. 513-525, sep 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494615003191>
- [16] G. Fernandes, L. F. Carvalho, J. J. Rodrigues, and M. L. Proença, "Network anomaly detection using IP flows with Principal Component Analysis and Ant Colony Optimization," *Journal of Network and Computer Applications*, vol. 64, pp. 1-11, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804516000618>
- [17] G. Fernandes, E. H. M. Pena, L. F. Carvalho, J. J. P. C. Rodrigues, and M. L. Proença, "Statistical, forecasting and metaheuristic techniques for network anomaly detection," in *Proceedings of the 30th Annual ACM Symposium on Applied Computing - SAC '15*. New York, New York, USA: ACM Press, apr 2015, pp. 701-707. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2695664.2695852>
- [18] M. Thottan, G. Liu, and C. Ji, "Anomaly Detection Approaches for Communication Networks," in *Algorithms for Next Generation Networks*, ser. Computer Communications and Networks, G. Cormode and M. Thottan, Eds. London: Springer London, 2010, vol. 2, ch. 11, pp. 239-261. [Online]. Available: http://dx.doi.org/10.1007/978-1-84882-765-3_11http://link.springer.com/chapter/10.1007/978-1-84882-765-3_11http://link.springer.com/10.1007/978-1-84882-765-3_11
- [19] Y. Yu, "A survey of anomaly intrusion detection techniques," *J. Comput. Sci. Coll.*, vol. 28, no. 1, pp. 9-17, 2012.
- [20] Z. Weiyu, Y. Qingbo, and G. Yushui, "A Survey of Anomaly Detection Methods in Networks," in *Computer Network and Multimedia Technology, 2009. CNMT 2009. International Symposium on*, 2009, pp. 1-3. [Online]. Available: http://ieeexplore.ieee.org/ielx5/5374431/5374489/05374676.pdf?tp=*&arnumber=5374676&isnumber=5374489
- [21] A. K. Marnerides, A. Schaeffer-Filho, and A. Mauthe, "Traffic anomaly diagnosis in Internet backbone networks: A survey," *Computer Networks*, vol. 73, pp. 224-243, nov 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128614002850>

- [22] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19-31, jan 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804515002891><http://linkinghub.elsevier.com/retrieve/pii/S1084804515002891>
- [23] S. Xiuyao, W. Mingxi, C. Jermaine, and S. Ranka, "Conditional anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 5, pp. 631-644, may 2007. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4138201>
- [24] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proceedings of the second ACM SIGCOMM Workshop on Internet measurment - IMW '02*. New York, New York, USA: ACM Press, 2002, p. 71. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=637201.637210>
- [25] P. Barford and D. Plonka, "Characteristics of network traffic flow anomalies," San Francisco, California, USA, pp. 69-73, 2001. [Online]. Available: http://delivery.acm.org/10.1145/510000/505211/p69-barford.pdf?ip=193.137.97.130&id=505211&acc=ACTIVESERVICE&key=2E5699D25B4FE09E.1FCDFBD0D1B4F091.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=598776887&CFTOKEN=76542358&__acm__=1415908245__6dff31f45715d169c
- [26] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash crowds and denial of service attacks," Honolulu, p. 293, 2002. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=511446.511485>
- [27] J. Pan, H. Hu, and Y. Liu, "Human behavior during Flash Crowd in web surfing," *Physica A: Statistical Mechanics and its Applications*, vol. 413, no. 0, pp. 212-219, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378437114005731>
- [28] A. A. Ghorbani, W. Lu, and M. Tavallaee, "Network attacks," *Advances in Information Security*, vol. 47, pp. 1-25, 2010. [Online]. Available: http://link.springer.com/10.1007/978-0-387-88771-5__1
- [29] F. Mouton, M. M. Malan, and H. S. Venter, "Social engineering from a normative ethics perspective," in *Information Security for South Africa, 2013*, 2013, pp. 1-8.
- [30] R. A. Maxion and T. N. Townsend, "Masquerade detection using truncated command lines," in *Dependable Systems and Networks, 2002. DSN 2002. Proceedings. International Conference on*, 2002, pp. 219-228.
- [31] M. Thottan and J. Chuanyi, "Anomaly detection in IP networks," *Signal Processing, IEEE Transactions on*, vol. 51, no. 8, pp. 2191-2204, 2003.
- [32] J. Cabrera, L. Lewis, Xinzhou Qin, Wenke Lee, R. Prasanth, B. Ravichandran, and R. Mehra, "Proactive detection of distributed denial of service attacks using MIB traffic variables-a feasibility study," in *2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium (Cat. No.01EX470)*. IEEE, 2001, pp. 609-622. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=918069>

References

- [33] J. Yu, H. Lee, M.-S. Kim, and D. Park, "Traffic flooding attack detection with SNMP MIB using SVM," *Computer Communications*, vol. 31, no. 17, pp. 4212-4219, nov 2008. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0140366408005094>
- [34] M. F. Lima, L. D. H. Sampaio, B. B. Zarpelao, J. J. P. C. Rodrigues, T. Abrao, and M. L. Proenca Jr., "Networking Anomaly Detection Using DSNs and Particle Swarm Optimization with Re-Clustering," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. IEEE, dec 2010, pp. 1-6. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5683910>
- [35] B. B. Zarpelao, L. S. Mendes, M. L. Proenca Jr., and J. J. P. C. Rodrigues, "Parameterized Anomaly Detection System with Automatic Configuration," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*. IEEE, nov 2009, pp. 1-6. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5426189>
- [36] N. Duffield, P. Haffner, B. Krishnamurthy, and H. Ringberg, "Rule-Based Anomaly Detection on IP Flows," in *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*. IEEE, apr 2009, pp. 424-432. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5061947>
- [37] R. Fontugne and K. Fukuda, "A Hough-transform-based anomaly detector with an adaptive time interval," *ACM SIGAPP Applied Computing Review*, vol. 11, no. 3, pp. 41-51, aug 2011. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2034594.2034598>
- [38] Cisco Systems, "Introduction to Cisco IOS ® NetFlow," *White Paper*, no. May, pp. 1-16, 2012. [Online]. Available: http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod{_}white{_}paper0900aecd80406232.htmlhttp://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod{_}white{_}paper0900aecd80406232.pdf
- [39] B. Claise, "RFC 3954: Cisco Systems NetFlow Services Export Version 9," pp. 1-33, 2004. [Online]. Available: <https://tools.ietf.org/html/rfc3954>
- [40] B. Trammell and B. Claise, "RFC 7011: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information," pp. 1-53, 2013. [Online]. Available: <https://tools.ietf.org/html/rfc7011>
- [41] C. Chapman, "Chapter 10 - Traffic performance testing in the network," in *Network Performance and Security*, 2016, pp. 295-317.
- [42] "NfSen - NetFlow Sensor," 2011. [Online]. Available: <http://nfsen.sourceforge.net>
- [43] "nTop," 2016. [Online]. Available: <http://www.ntop.org/>
- [44] S. Panchen, P. Phaal, and N. McKee, "RFC 3176: InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks," pp. 1-31, 2001. [Online]. Available: <https://tools.ietf.org/html/rfc3176>
- [45] N. Duffield, "Sampling for Passive Internet Measurement: A Review," *Statistical Science*, vol. 19, no. 3, pp. 472-498, aug 2004. [Online]. Available: <http://projecteuclid.org/Dienst/getRecord?id=euclid.ss/1110999311/>

- [46] "Cisco NetFlow-Lite Solution Overview," 2016. [Online]. Available: http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-2960-x-series-switches/solution_overview_c22-728776.html
- [47] L. Deri, E. Chou, Z. Cherian, K. Karmarkar, and M. Patterson, "Increasing data center network visibility with cisco netflow-lite," in *2011 7th International Conference on Network and Service Management*, Oct 2011, pp. 1-6.
- [48] Z. Jadidi, V. Muthukkumarasamy, E. Sithirasenan, and K. Singh, "Flow-Based Anomaly Detection in Big Data," in *Networking for Big Data*, ser. Chapman & Hall/CRC Big Data Series. Chapman and Hall/CRC, jul 2015, pp. 257-279. [Online]. Available: <http://dx.doi.org/10.1201/b18772-17>
- [49] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An Overview of IP Flow-Based Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 3, pp. 343-356, 2010. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5455789><http://ieeexplore.ieee.org/document/5455789/>
- [50] P. Winter, E. Hermann, and M. Zeilinger, "Inductive Intrusion Detection in Flow-Based Network Data Using One-Class Support Vector Machines," in *2011 4th IFIP International Conference on New Technologies, Mobility and Security*. IEEE, feb 2011, pp. 1-5. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5720582>
- [51] K. Bartos, M. Rehak, and V. Krmicek, "Optimizing flow sampling for network anomaly detection," in *2011 7th International Wireless Communications and Mobile Computing Conference*. IEEE, jul 2011, pp. 1304-1309. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5982728>
- [52] Y. ZHANG, B. FANG, and H. LUO, "Identifying High-Rate Flows Based on Sequential Sampling," *IEICE Transactions on Information and Systems*, vol. E93-D, no. 5, pp. 1162-1174, 2010. [Online]. Available: <http://joi.jlc.jst.go.jp/JST.JSTAGE/transinf/E93.D.1162?from=CrossRef>
- [53] J. M. C. Silva, P. Carvalho, and S. R. Lima, "Analysing traffic flows through sampling: A comparative study," pp. 341-346, 2015.
- [54] R. Kemmerer and G. Vigna, "Intrusion detection: a brief history and overview," *Computer*, vol. 35, no. 4, pp. 27-30, apr 2002. [Online]. Available: <http://ieeexplore.ieee.org/document/1012428/>
- [55] W. Lee and S. Stolfo, "Data mining approaches for intrusion detection," in *Proceedings of the 7th {USENIX} Security Symp.* USENIX Association, 1998, pp. 6-6. [Online]. Available: <https://dl.acm.org/citation.cfm?id=1267555>
- [56] W. Bul'ajoul, A. James, and M. Pannu, "Improving network intrusion detection system performance through quality of service configuration and parallel technology," *Journal of Computer and System Sciences*, vol. 81, no. 6, pp. 981-999, sep 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0022000014001767>
- [57] H. Bostani and M. Sheikhan, "Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach," *Computer Communications*, vol. 98, no. Supplement C, pp. 52-71, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0140366416306387>

References

- [58] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, "Evaluating Computer Intrusion Detection Systems: A Survey of Common Practices," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1-41, sep 2015. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2808687.2808691>
- [59] I. Butun, S. D. Morgera, and R. Sankar, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 266-282, 2014.
- [60] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Computer Networks*, vol. 31, no. 8, pp. 805-822, apr 1999. [Online]. Available: <http://dl.acm.org/citation.cfm?id=324119.324126http://linkinghub.elsevier.com/retrieve/pii/S1389128698000176>
- [61] L. Meng Hui and A. Jones, "Network Anomaly Detection System: The State of Art of Network Behaviour Analysis," in *Convergence and Hybrid Information Technology, 2008. ICHIT '08. International Conference on*, 2008, pp. 459-465.
- [62] T. S. Sobh, "Wired and wireless intrusion detection system: Classifications, good characteristics and state-of-the-art," *Computer Standards and Interfaces*, vol. 28, no. 6, pp. 670-694, sep 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S092054890500098X>
- [63] M. V. O. de Assis, A. H. Hamamoto, T. Abrao, and M. L. Proenca, "A Game Theoretical Based System Using Holt-Winters and Genetic Algorithm with Fuzzy Logic for DoS/DDoS Mitigation on SDN Networks," p. 1, 2017.
- [64] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, p. 219, oct 2004. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1030194.1015492http://portal.acm.org/citation.cfm?doid=1030194.1015492>
- [65] —, "Mining anomalies using traffic feature distributions," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, p. 217, oct 2005. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1090191.1080118>
- [66] C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "Combining sketches and wavelet analysis for multi time-scale network anomaly detection," *Computers & Security*, vol. 30, no. 8, pp. 692-704, nov 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404811001064http://ac.els-cdn.com/S0167404811001064/1-s2.0-S0167404811001064-main.pdf?{_}tid=8340683e-7274-11e4-9532-00000aacb35d{\&}acdnat=1416680727{_}1b3455fb49aa070126a519d2dec286e7http://linkinghub.els
- [67] M. Hamdi and N. Boudriga, "Detecting Denial-of-Service attacks using the wavelet transform," *Computer Communications*, vol. 30, no. 16, pp. 3203-3213, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366407002678http://ac.els-cdn.com/S0140366407002678/1-s2.0-S0140366407002678-main.pdf?{_}tid=801aa0c0-7274-11e4-935d-00000aacb35e{\&}acdnat=1416680722{_}5a5b98e4ce0c6976469dc322e1822f09

- [68] I. T. Jolliffe, *Principal Component Analysis*. Springer, 2002. [Online]. Available: http://books.google.com.br/books?id={_}olByCrhJwIC
- [69] J. E. Jackson, *A User's Guide to Principal Components*. Wiley, 2005. [Online]. Available: <https://books.google.com.br/books?id=f9s6g6cmUTUC>
- [70] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for traffic anomaly detection," *SIGMETRICS Perform. Eval. Rev.*, vol. 35, no. 1, pp. 109-120, 2007.
- [71] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Matrices via Convex Optimization," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 2080-2088.
- [72] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust Principal Component Analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1--11:37, jun 2011. [Online]. Available: <http://doi.acm.org/10.1145/1970392.1970395>
- [73] C. Pascoal, M. Rosario de Oliveira, R. Valadas, P. Filzmoser, P. Salvador, and A. Pacheco, "Robust feature selection and robust PCA for internet traffic anomaly detection," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 1755-1763. [Online]. Available: <http://ieeexplore.ieee.org/ielx5/6189419/6195452/06195548.pdf?tp={\&}arnumber=6195548{\&}isnumber=6195452>
- [74] Y. Kanda, R. Fontugne, K. Fukuda, and T. Sugawara, "ADMIRE: Anomaly detection method using entropy-based PCA with three-step sketches," *Computer Communications*, vol. 36, no. 5, pp. 575-588, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366412003994http://ac.els-cdn.com/S0140366412003994/1-s2.0-S0140366412003994-main.pdf?{_}tid=a08503e6-9cc2-11e4-a24e-00000aab0f26{\&}acdnat=1421332225{_}7db084292d3c403936aabd8034e2bef4
- [75] C. O'Reilly, A. Gluhak, and M. A. Imran, "Distributed Anomaly Detection Using Minimum Volume Elliptical Principal Component Analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2320-2333, sep 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7456266/>
- [76] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, and G. Maciá-Fernández, "PCA-based multivariate statistical network monitoring for anomaly detection," *Computers & Security*, vol. 59, pp. 118-137, jun 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167404816300116http://linkinghub.elsevier.com/retrieve/pii/S0167404816300116>
- [77] G. Fernandes, J. J. Rodrigues, and M. L. Proença, "Autonomous profile-based anomaly detection system using principal component analysis and flow analysis," *Applied Soft Computing*, vol. 34, pp. 513-525, sep 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494615003191>
- [78] G. Fernandes, A. M. Zacaron, J. J. P. C. Rodrigues, and M. L. Proença, "Digital signature to help network management using principal component analysis and K-means clustering," in *2013 IEEE International Conference on Communications (ICC)*. IEEE, jun 2013, pp. 2519-2523. [Online]. Available: <http://ieeexplore.ieee.org/document/6654912/>

References

- [79] D. S. Yeung, J. Shuyuan, and W. Xizhao, "Covariance-Matrix Modeling and Detecting Various Flooding Attacks," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 37, no. 2, pp. 157-169, 2007. [Online]. Available: <http://ieeexplore.ieee.org/ielx5/3468/4100767/04100784.pdf?tp={\&}arnumber=4100784{\&}isnumber=4100767>
- [80] M. Xie, J. Hu, and S. Guo, "Segment-Based Anomaly Detection with Approximated Sample Covariance Matrix in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 574-583, feb 2015. [Online]. Available: <http://ieeexplore.ieee.org/ielx7/71/4359390/06748064.pdf?tp={\&}arnumber=6748064{\&}isnumber=4359390http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6748064>
- [81] T. Huang, H. Sethu, and N. Kandasamy, "A New Approach to Dimensionality Reduction for Anomaly Detection in Data Traffic," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 651-665, sep 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7580700/>
- [82] K. Kalkan and F. Alagöz, "A distributed filtering mechanism against DDoS attacks: ScoreForCore," *Computer Networks*, vol. 108, pp. 199-209, oct 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1389128616302754>
- [83] H. Ozkan, F. Ozkan, and S. S. Kozat, "Online Anomaly Detection Under Markov Statistics With Controllable Type-I Error," *IEEE Transactions on Signal Processing*, vol. 64, no. 6, pp. 1435-1445, mar 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7339701/>
- [84] M. L. Proença, C. Coppelmans, M. Bottoli, A. Alberti, and L. S. Mendes, "The Hurst Parameter for Digital Signature of Network Segment," in *Telecommunications and Networking - ICT 2004: 11th International Conference on Telecommunications, Fortaleza, Brazil, August 1-6, 2004. Proceedings*, J. N. de Souza, P. Dini, and P. Lorenz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 772-781. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-27824-5{_}103http://link.springer.com/10.1007/978-3-540-27824-5{_}103
- [85] E. H. M. Pena, L. F. Carvalho, S. Barbon Jr., J. J. P. C. Rodrigues, and M. L. Proença Jr., "Anomaly detection using the correlational paraconsistent machine with digital signatures of network segment," *Information Sciences*, vol. 420, no. Supplement C, pp. 313-328, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020025517309131>
- [86] E. H. M. Pena, L. F. Carvalho, S. Barbon, J. J. P. C. Rodrigues, and M. L. Proença, "Correlational paraconsistent machine for anomaly detection," in *2014 IEEE Global Communications Conference*. IEEE, dec 2014, pp. 551-556. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7036865>
- [87] J.-h. Bang, Y.-J. Cho, and K. Kang, "Anomaly detection of network-initiated LTE signaling traffic in wireless sensor and actuator networks based on a Hidden semi-Markov Model," *Computers & Security*, vol. 65, pp. 108-120, mar 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167404816301614http://linkinghub.elsevier.com/retrieve/pii/S0167404816301614>

- [88] H. Ren, Z. Ye, and Z. Li, "Anomaly detection based on a dynamic Markov model," *Information Sciences*, may 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020025517307302><http://linkinghub.elsevier.com/retrieve/pii/S0020025517307302>
- [89] H. H. Jazi, H. Gonzalez, N. Stakhanova, and A. A. Ghorbani, "Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling," *Computer Networks*, vol. 121, pp. 25-36, jul 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128617301172><http://linkinghub.elsevier.com/retrieve/pii/S1389128617301172>
- [90] J. Han, M. Kamber, and J. Pei, "10 - Cluster Analysis: Basic Concepts and Methods," in *Data Mining (Third Edition)*, J. H. Kamber and J. Pei, Eds. Boston: Morgan Kaufmann, 2012, pp. 443-495. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/B9780123814791000101>
- [91] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Hyperspherical cluster based distributed anomaly detection in wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 74, no. 1, pp. 1833-1847, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0743731513002013>http://ac.els-cdn.com/S0743731513002013/1-s2.0-S0743731513002013-main.pdf?{_}tid=d7a0abe8-9ca7-11e4-8d59-00000aab0f6b{\&}{acdnat=1421320721{_}e6ef259e9b92d93a66ab81d9c0aa3f98
- [92] J. Mazel, P. Casas, Y. Labit, and P. Owezarski, "Sub-space clustering, inter-clustering results association & anomaly correlation for unsupervised network anomaly detection," in *CNSM '11 Proceedings of the 7th International Conference on Network and Services Management*, 2011, pp. 73-80. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2147683>
- [93] A. Karami and M. Guerrero-Zapata, "A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks," *Neurocomputing*, vol. 149, Part, no. 0, pp. 1253-1269, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925231214011588>http://ac.els-cdn.com/S0925231214011588/1-s2.0-S0925231214011588-main.pdf?{_}tid=20b9b8ee-9a49-11e4-bcab-00000aab0f27{\&}{acdnat=1421060140{_}feb30cf4305eddfb9d7023d8b74d686e
- [94] L. F. Carvalho, S. Barbon, L. d. S. Mendes, and M. L. Proença, "Unsupervised learning clustering and self-organized agents applied to help network management," *Expert Systems with Applications*, vol. 54, pp. 29-47, jul 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417416000555>
- [95] J. Dromard, G. Roudiere, and P. Owezarski, "Online and Scalable Unsupervised Network Anomaly Detection Method," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 34-47, mar 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7740019/>
- [96] D. He, S. Chan, X. Ni, and M. Guizani, "Software-Defined-Networking-Enabled Traffic Anomaly Detection and Mitigation," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1-1, 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7902104/>

References

- [97] E. Bigdeli, M. Mohammadi, B. Raahemi, and S. Matwin, "Incremental anomaly detection using two-layer cluster-based structure," *Information Sciences*, vol. 429, pp. 315-331, mar 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0020025517310939>
- [98] J. M. Estevez-Tapiador, P. Garcia-Teodoro, and J. E. Diaz-Verdejo, "Stochastic protocol modeling for anomaly based network intrusion detection," in *Information Assurance, 2003. IWIAS 2003. Proceedings. First IEEE International Workshop on*, 2003, pp. 3-12.
- [99] M.-Y. Su, "Discovery and prevention of attack episodes by frequent episodes mining and finite state machines," *Journal of Network and Computer Applications*, vol. 33, no. 2, pp. 156-167, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804509001337>http://ac.els-cdn.com/S1084804509001337/1-s2.0-S1084804509001337-main.pdf?{_}tid=5bbe5a1e-9cc7-11e4-9e5e-00000aab0f02{\&}acdnat=1421334258{_}228f93e964acbb42b1ffca110b0ad1e6
- [100] C. Hammerschmidt, S. Marchal, R. State, G. Pellegrino, and S. Verwer, "Efficient Learning of Communication Profiles from IP Flow Records," in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. IEEE, nov 2016, pp. 559-562. [Online]. Available: <http://ieeexplore.ieee.org/document/7796840/>
- [101] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley, 2012. [Online]. Available: <http://books.google.pt/books?id=Br33IRC3PkQC>
- [102] M. Klassen and Y. Ning, "Anomaly based intrusion detection in wireless networks using Bayesian classifier," in *Advanced Computational Intelligence (ICACI), 2012 IEEE Fifth International Conference on*, 2012, pp. 257-264. [Online]. Available: <http://ieeexplore.ieee.org/ielx7/6449391/6462631/06463163.pdf?tp={\&}arnumber=6463163{\&}isnumber=6462631>
- [103] L. Tao, Q. Ailing, H. Yuanbin, and C. Xintan, "Method for network anomaly detection based on Bayesian statistical model with time slicing," in *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, 2008, pp. 3359-3362. [Online]. Available: <http://ieeexplore.ieee.org/ielx5/4577718/4592780/04593458.pdf?tp={\&}arnumber=4593458{\&}isnumber=4592780>
- [104] M. Swarnkar and N. Hubballi, "OCPAD: One class Naive Bayes classifier for payload based anomaly detection," *Expert Systems with Applications*, vol. 64, pp. 330-339, dec 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0957417416303839>
- [105] C. A. Catania, F. Bromberg, and C. G. Garino, "An autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection," *Expert Systems with Applications*, vol. 39, no. 2, pp. 1822-1829, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417411011808>http://ac.els-cdn.com/S0957417411011808/1-s2.0-S0957417411011808-main.pdf?{_}tid=c74d040e-9d91-11e4-9ed3-00000aacb362{\&}acdnat=1421421196{_}e2813b05c7bb6e5b92998b07f49c6096
- [106] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," Chicago, Illinois, pp. 8-15, 2013. [Online]. Available: <http://delivery.acm.org/10.1145/2510000/2500857/p8-amer.pdf>

- ip=193.136.67.236&id=2500857&acc=ACTIVESERVICE&key=2E5699D25B4FE09E.1FCDFBD0D1B4F091.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=472463909&CFTOKEN=33700232&acm=1421425889403ef1b41ae42174d5
- [107] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 121-134, oct 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0031320316300267>
 - [108] H. Wang, J. Gu, and S. Wang, "An effective intrusion detection framework based on SVM with feature augmentation," *Knowledge-Based Systems*, vol. 136, no. Supplement C, pp. 130-139, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S095070511730415X>
 - [109] E. Kabir, J. Hu, H. Wang, and G. Zhuo, "A novel statistical technique for intrusion detection systems," *Future Generation Computer Systems*, jan 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167739X17301371>
<http://linkinghub.elsevier.com/retrieve/pii/S0167739X17301371>
 - [110] B. Subba, S. Biswas, and S. Karmakar, "A Neural Network based system for Intrusion Detection and attack classification," in *2016 Twenty Second National Conference on Communication (NCC)*. IEEE, mar 2016, pp. 1-6. [Online]. Available: <http://ieeexplore.ieee.org/document/7561088/>
 - [111] A. Saeed, A. Ahmadiania, A. Javed, and H. Larijani, "Intelligent Intrusion Detection in Low-Power IoTs," *ACM Transactions on Internet Technology*, vol. 16, no. 4, pp. 1-25, dec 2016. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3023158.2990499>
 - [112] J. Brown, M. Anwar, and G. Dozier, "An Evolutionary General Regression Neural Network Classifier for Intrusion Detection," in *2016 25th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, aug 2016, pp. 1-5. [Online]. Available: <http://ieeexplore.ieee.org/document/7568493/>
 - [113] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484-497, feb 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020025516302547>
<http://linkinghub.elsevier.com/retrieve/pii/S0020025516302547>
 - [114] S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems," *Journal of Network and Computer Applications*, vol. 30, no. 1, pp. 114-132, jan 2007. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1084804505000445>
 - [115] A. A. Aburomman and M. B. Ibne Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Applied Soft Computing*, vol. 38, pp. 360-372, jan 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1568494615006328>
 - [116] P. Sornsuwit and S. Jaiyen, "Intrusion detection model based on ensemble learning for U2R and R2L attacks," in *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*. IEEE, oct 2015, pp. 354-359. [Online]. Available: <http://ieeexplore.ieee.org/document/7408971/>

References

- [117] V. Bukhtoyarov and V. Zhukov, "Ensemble-Distributed Approach in Classification Problem Solution for Intrusion Detection Systems," in *Intelligent Data Engineering and Automated Learning - IDEAL 2014: 15th International Conference, Salamanca, Spain, September 10-12, 2014. Proceedings*, E. Corchado, J. A. Lozano, H. Quintián, and H. Yin, Eds. Cham: Springer International Publishing, 2014, pp. 255-265. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10840-7_{32}
- [118] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379-423, jul 1948. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6773024>
- [119] T. M. Cover and J. A. Thomas, *Elements of information theory*, 2nd ed. Wiley-Interscience, 2006. [Online]. Available: <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471241954.html>
- [120] Wenke Lee and Dong Xiang, "Information-theoretic measures for anomaly detection," in *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*. IEEE Comput. Soc, pp. 130-143. [Online]. Available: <http://ieeexplore.ieee.org/document/924294/>
- [121] J. David and C. Thomas, "DDoS Attack Detection Using Fast Entropy Approach on Flow-Based Network Traffic," *Procedia Computer Science*, vol. 50, pp. 30-36, 2015. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1877050915005086>
- [122] A. A. Amaral, L. d. S. Mendes, B. B. Zarpelão, and M. L. P. Junior, "Deep IP flow inspection to detect beyond network anomalies," *Computer Communications*, vol. 98, pp. 80-96, jan 2017. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0140366416306612>
- [123] M. H. Bhuyan, D. Bhattacharyya, and J. Kalita, "A multi-step outlier-based anomaly detection approach to network-wide traffic," *Information Sciences*, vol. 348, pp. 243-271, jun 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0020025516300779>
- [124] P. Berezinski, B. Jasiul, and M. Szpyrka, "An Entropy-Based Network Anomaly Detection Method," *Entropy*, vol. 17, no. 4, pp. 2367-2408, apr 2015. [Online]. Available: <http://www.mdpi.com/1099-4300/17/4/2367/>
- [125] S. Behal and K. Kumar, "Detection of DDoS attacks and flash events using novel information theory metrics," *Computer Networks*, vol. 116, pp. 96-110, apr 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128617300592http://linkinghub.elsevier.com/retrieve/pii/S1389128617300592>
- [126] M. Xie, J. Hu, S. Guo, and A. Y. Zomaya, "Distributed Segment-Based Anomaly Detection With Kullback-Leibler Divergence in Wireless Sensor Networks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 101-110, jan 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7555360/>
- [127] G. Li and Y. Wang, "Differential Kullback-Leibler Divergence Based Anomaly Detection Scheme in Sensor Networks," in *2012 IEEE 12th International Conference on Computer and Information Technology*. IEEE, oct 2012, pp. 966-970. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6392034>

- [128] A. K. Kar, "Bio inspired computing - A review of algorithms and scope of applications," *Expert Systems with Applications*, vol. 59, pp. 20-32, oct 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S095741741630183X>
- [129] A. Firdaus, N. B. Anuar, M. F. A. Razak, and A. K. Sangaiah, "Bio-inspired computational paradigm for feature investigation and malware detection: interactive analytics," pp. 1-37, mar 2017. [Online]. Available: <http://link.springer.com/10.1007/s11042-017-4586-0>
- [130] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153-1176, jan 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7307098/>
- [131] S. Sen, "A Survey of Intrusion Detection Systems Using Evolutionary Computation," in *Bio-Inspired Computation in Telecommunications*. Elsevier, 2015, pp. 73-94. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/B9780128015384000045>
- [132] L. N. de Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. London. UK.: Springer-Verlag, sep 2002. [Online]. Available: <http://www.cs.kent.ac.uk/pubs/2002/1507>
- [133] P. Saurabh and B. Verma, "An efficient proactive artificial immune system based anomaly detection and prevention system," *Expert Systems with Applications*, vol. 60, pp. 311-320, oct 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0957417416301403>
- [134] O. Igbe, I. Darwish, and T. Saadawi, "Distributed Network Intrusion Detection Systems: An Artificial Immune System Approach," in *2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, jun 2016, pp. 101-106. [Online]. Available: <http://ieeexplore.ieee.org/document/7545821/>
- [135] S. Shamshirband, N. B. Anuar, M. L. M. Kiah, V. A. Rohani, D. Petković, S. Misra, and A. N. Khan, "Co-FAIS: Cooperative fuzzy artificial immune system for detecting intrusion in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 42, pp. 102-117, jun 2014. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1084804514000666>
- [136] B. M. Aslahi-Shahri, R. Rahmani, M. Chizari, A. Maralani, M. Eslami, M. J. Golkar, and A. Ebrahimi, "A hybrid method consisting of GA and SVM for intrusion detection system," *Neural Computing and Applications*, vol. 27, no. 6, pp. 1669-1676, 2016. [Online]. Available: <http://dx.doi.org/10.1007/s00521-015-1964-2>
- [137] S. Singh and R. S. Kushwah, "Energy Efficient Approach for Intrusion Detection System for WSN by applying Optimal Clustering and Genetic Algorithm," in *Proceedings of the International Conference on Advances in Information Communication Technology & Computing - AICTC '16*. New York, New York, USA: ACM Press, 2016, pp. 1-6. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2979779.2979840>
- [138] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão, and M. L. Proença, "Network Anomaly Detection System using Genetic Algorithm and Fuzzy Logic," *Expert Systems*

References

- with Applications*, vol. 92, no. Supplement C, pp. 390-402, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S095741741730619X>
- [139] S. Elsayed, R. Sarker, and J. Slay, "Evaluating the performance of a differential evolution algorithm in anomaly detection," in *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, may 2015, pp. 2490-2497. [Online]. Available: <http://ieeexplore.ieee.org/document/7257194/>
- [140] C.-L. Huang and J.-F. Dun, "A distributed PSO-SVM hybrid system with feature selection and parameter optimization," *Applied Soft Computing*, vol. 8, no. 4, pp. 1381-1391, sep 2008. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1568494607001299>
- [141] S.-W. Lin, K.-C. Ying, S.-C. Chen, and Z.-J. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," *Expert Systems with Applications*, vol. 35, no. 4, pp. 1817-1824, nov 2008. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0957417407003752>
- [142] S. M. Hosseini Bamakan, H. Wang, T. Yingjie, and Y. Shi, "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization," *Neurocomputing*, vol. 199, pp. 90-102, jul 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0925231216300510>
- [143] M. Grill and T. Pevný, "Learning combination of anomaly detectors for security domain," *Computer Networks*, vol. 107, pp. 55-63, oct 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1389128616301669>
- [144] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Systems with Applications*, vol. 67, pp. 296-303, jan 2017. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0957417416305310>
- [145] A. Forestiero, "Self-organizing anomaly detection in data streams," *Information Sciences*, vol. 373, pp. 321-336, dec 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S002002551630737X>
- [146] O. Salem, A. Guerassimov, A. Mehaoua, A. Marcus, and B. Furht, "Anomaly Detection in Medical Wireless Sensor Networks using SVM and Linear Regression Models," *International Journal of E-Health and Medical Communications (IJEHMC)*, vol. 5, no. 1, pp. 20-45, 2014. [Online]. Available: <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/ijehmc.2014010102>
- [147] W. Wang, J. Liu, G. Pitsilis, and X. Zhang, "Abstracting massive data for lightweight intrusion detection in computer networks," *Information Sciences*, oct 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0020025516312385>
- [148] M. H. A. C. Adaniya, T. Abrão, and M. L. Proença Jr., "Anomaly Detection Using Metaheuristic Firefly Harmonic Clustering," *Journal of Networks*, vol. 8, no. 1, pp. 82-91, jan 2013. [Online]. Available: <http://ojs.academpublisher.com/index.php/jnw/article/view/9061>

- [149] M. Proenca, B. Zarpelao, and L. Mendes, "Anomaly detection for network servers using digital signature of network segment," in *Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop (AICT/SAPIR/ELETE'05)*. IEEE, 2005, pp. 290-295. [Online]. Available: <http://ieeexplore.ieee.org/document/1517644/>
- [150] M.-H. Chen, P.-C. Chang, and J.-L. Wu, "A population-based incremental learning approach with artificial immune system for network intrusion detection," *Engineering Applications of Artificial Intelligence*, vol. 51, pp. 171-181, may 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0952197616000245>
- [151] M. Grill, T. Pevný, and M. Rehak, "Reducing false positives of network anomaly detection by local adaptive multivariate smoothing," *Journal of Computer and System Sciences*, vol. 83, no. 1, pp. 43-57, feb 2017. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0022000016300022>
- [152] C. Guo, Y. Ping, N. Liu, and S.-S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, pp. 391-400, nov 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0925231216306592>
- [153] M. V. de Assis, J. J. Rodrigues, and M. L. Proença, "A seven-dimensional flow analysis to help autonomous network management," *Information Sciences*, vol. 278, pp. 900-913, sep 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020025514003995>
- [154] K. P. F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559-572, 1901. [Online]. Available: <https://doi.org/10.1080/14786440109462720>
- [155] M. Navas and C. Ordonez, "Efficient computation of PCA with SVD in SQL," in *Proceedings of the 2nd Workshop on Data Mining using Matrices and Tensors - DMMT '09*. New York, New York, USA: ACM Press, 2009, pp. 1-10. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1581114.1581119>
- [156] J. J. Dongarra and D. C. Sorensen, "A fast algorithm for the symmetric eigenvalue problem," *Proceedings of the 7th Symposium on Computer Arithmetic*, pp. 337-342, jun 1987. [Online]. Available: <http://ieeexplore.ieee.org/document/6158944/>
- [157] B. Parlett, *The Symmetric Eigenvalue Problem*. Society for Industrial and Applied Mathematics, 1998. [Online]. Available: <https://epubs.siam.org/doi/abs/10.1137/1.9781611971163>
- [158] A. A. Poli and M. C. Cirillo, "On the use of the normalized mean square error in evaluating dispersion model performance," *Atmospheric Environment. Part A. General Topics*, vol. 27, no. 15, pp. 2427 - 2434, 1993. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/096016869390410Z>
- [159] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise Reduction in Speech Processing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1-4. [Online]. Available: https://doi.org/10.1007/978-3-642-00296-0_5

References

- [160] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861-874, jun 2006. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S016786550500303X>
- [161] M. V. De Assis and M. L. Proença, "Scorpius: sFlow network anomaly simulator," in *Journal of Computer Science*, jul 2015, vol. 11, no. 4, pp. 662-674.
- [162] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Comput. Surv.*, vol. 39, no. 1, p. 3, 2007.
- [163] D. Wang, L. He, Y. Xue, and Y. Dong, "Exploiting Artificial Immune systems to detect unknown DoS attacks in real-time," in *Proceedings - 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, IEEE CCIS 2012*, vol. 2. IEEE, oct 2013, pp. 646-650. [Online]. Available: <http://ieeexplore.ieee.org/document/6664254/>
- [164] S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, and F. Tang, "Discriminating DDoS attacks from flash crowds using flow correlation coefficient," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, pp. 1073-1080, jun 2012. [Online]. Available: <http://ieeexplore.ieee.org/document/6060809/>
- [165] L. F. Carvalho, T. Abrão, L. d. S. Mendes, and M. L. Proença, "An ecosystem for anomaly detection and mitigation in software-defined networking," *Expert Systems with Applications*, vol. 104, pp. 121-133, aug 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417418301726?via=ihub>
- [166] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43-49, feb 1978. [Online]. Available: <http://ieeexplore.ieee.org/document/1163055/>
- [167] P. Esling and C. Agon, "Time-series data mining," *ACM Computing Surveys*, vol. 45, no. 1, pp. 1-34, nov 2012. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2379776.2379788>
- [168] P. Phaal, S. Panchen, and N. McKee, "InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks," Tech. Rep., sep 2001. [Online]. Available: <https://www.rfc-editor.org/info/rfc3176>
- [169] S. Chang, X. Qiu, Z. Gao, K. Liu, and F. Qi, "A flow-based anomaly detection method using sketch and combinations of traffic features," in *Proceedings of the 2010 International Conference on Network and Service Management, CNSM 2010*. IEEE, oct 2010, pp. 302-305. [Online]. Available: <http://ieeexplore.ieee.org/document/5691206/>
- [170] E. Biglieri and K. Yao, "Some properties of singular value decomposition and their applications to digital signal processing," *Signal Processing*, vol. 18, no. 3, pp. 277-289, nov 1989. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/016516848990039X>

