Timely Classification of Encrypted or Protocol-Obfuscated Internet Traffic Using Statistical Methods

Vanice Canuto Cunha

UNIVERSIDADE
BEIRA INTERIOR

Timely Classification of Encrypted or Protocol-Obfuscated Internet Traffic Using Statistical Methods

Vanice Canuto Cunha

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

Orientador: Prof. Doutor Mário Marques Freire, Universidade da Beira Interior
Co-orientador: Prof. Doutor Damien Magoni, Université de Bordeaux, France

Júri:
Prof. Doutor Hugo P. M. C. Proença, Universidade da Beira Interior (Presidente)
Prof. Doutor Damien Magoni, Université de Bordeaux, France
Prof. Doutor Rui Jorge Morais Tomaz Valadas, Universidade de Lisboa
Prof. Doutora Marília Pascoal Curado, Universidade de Coimbra
Prof. Doutor Alexandre Júlio Teixeira Santos, Universidade do Minho
Prof. Doutor Arturo Alejandro Z. Zavala, Universidade Federal de Mato Grosso
Prof. Doutor Pedro Ricardo Morais Inácio, Universidade da Beira Interior
Prof. Doutor Bruno Miguel Correia da Silva, Universidade da Beira Interior

abril de 2023

Thesis prepared at Instituto de Telecomunicações - Delegação da Covilhã and at the Department of Computer Science of the University of Beira Interior, and submitted to the University of Beira Interior for discussion in public session to obtain the Ph.D. Degree in Computer Science and Engineering.

This work has been funded by Portuguese FCT/MCTES through national funds and, when applicable, co-funded by EU funds under the project UIDB/50008/2020, and by operation Centro-01-0145-FEDER-000019 - C4 - Centro de Competências em Cloud Computing, co-funded by the European Regional Development Fund (ERDF/FEDER) through the Programa Operacional Regional do Centro (Centro 2020). This work has also been funded by CAPES (Brazilian Federal Agency for Support and Evaluation of Graduate Education) within the Ministry of Education of Brazil under a scholarship supported by the International Cooperation Program CAPES/COFECUB - Project 9090-13-4/2013 at the University of Beira Interior.

Cofinanciado por:

UNIVERSIDADE BEIRA INTERIOR

Declaração de Integridade

Eu, Vanice Canuto Cunha, que abaixo assino, estudante com número de inscrição D1360 do curso de Engenharia Informática da Faculdade de Engenharia, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridade da Universidade da Beira Interior**.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, e que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assim assumo na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 20/04/2022.

gov.br Documento assinado digitalmente
VANICE CANUTO CUNHA
Data: 05/04/2023 12:24:39-0300
Verifique em https://validar.iti.gov.br

Vanice Canuto Cunha.

Dedication

Eu dedico esta tese ao Senhor Deus, ao Senhor Jesus Cristo e ao meu esposo Alexandre Magno de Melo Faria.

Acknowledgements

First of all, I would like to honor and give grace to the Lord and to Jesus Christ, for giving me the strength, inner-peace, determination, wisdom, patience and the courage so that I could finish this doctoral program. Praise the Lord.

My most sincere and honest thanks to the love of my life, my husband, friend and professor Dr. Alexandre Magno de Melo Faria for all the support, for all the times he renounced his appointments to help me out, for helping me understand the results countless times, for giving me ideas, for the patience, for all the discussions about methodology, for explaining to me what a survey is, for reading my work... for believing more than I did that this project would work out and be completed. Without your support none of this would be possible. I love you!

The development and the conclusion of this thesis was only possible with the help, support and assistance of many important people that God placed in my life and I will carry them with me in my heart.

A huge thank you to professor Dr. Arturo Zavala Zavala for the statistical teachings, for being with me and teaching me countless times, for helping me with all the patience to interpret the equations and guiding me through the statistical methods, for the support and for being willing to help me all the times I had doubts about the theme. To you, all my gratitude and my admiration.

Gratitude to my friend and Information Technology Consultant Fábio Ricardo Araújo da Silva for all the help developing the prototypes, for introducing me to sckit-learn, for being willing to help me when the application would not work, for correcting bugs, for the patience on the infinity of tests and program versions we made. Words are not enough to say how thankful I am.

Thanks to professor Dr. Maurício Fernando Lima Pereira for helping me countless times to format this thesis on Latex, mainly when formatting the several tables, which used to distress me.Thank you for helping me so willingly.

Thanks to professor Msc. Fabrício César de Moraes for the classes and teachings about Kolmogorov-Smirnov and Chi-square tests.

Many thanks to my dear teacher Luana Bulgarelli Mendes for always being willing to translate and correct my texts to English.Thank you very much my friend!!!!

I would like to thank the professors from the Computing Institution (IC) from the Federal University of Mato Grosso (UFMT) for all the support, the incentive and the patience,

mainly professor Dr. Eunice Pereira dos Santos Nunes, the ex-directors professor Dr. Josiel Maimone de Figueiredo and professor Dr. Elmo Batista de Faria, the current directors professors Dr. Allan Gonçalves de Oliveira and professor Dr. Nielsen Cassiano Simões for understanding my situation and for many times allowing me to have face-to-face meetings with my advisor.

I would also like to thank CAPES (Brazilian Federal Agency for Support and Evaluation of Graduate Education) for funding this thesis through the International Cooperation Program CAPES/COFECUB - Project 9090-13-4/2013 at the University of Beira Interior.

My gratitude to my advisor, professor Dr. Mário Marques Freire for the teachings, for guiding me through this journey and for accepting me into this doctoral program, and to my co-advisor professor Dr. Damien Magoni for reading so carefully all my works.

I would like to thank my friends at Network and Multimedia Computing Group (NMCG), in particular my friend Musa Gwani Samaila for verifying whenever asked if my server was on so I could remotely.

I thank my family, my parents Teresinha de Jesus Canuto, Francisco Pereira da Cunha, my siblings Werthon Canuto Cunha e Vanise Canuto Cunha and my in-laws Julia Maria de Melo Faria and Lúcio Nunes de Faria for all the support, the patience, and the comprehension.

I thank everyone that directly or indirectly helped me somehow to conclude this journey.

Resumo

A classificação de tráfego Internet visa identificar o tipo de aplicação ou protocolo que gerou um determinado pacote ou fluxo de pacotes na rede. Através da classificação de tráfego, Fornecedores de Serviços de Internet (ISP), governos e administradores de rede podem ter acesso às funções básicas e várias soluções, incluindo gestão da rede, monitoramento avançado de rede, auditoria de rede e deteção de anomalias. Classificar o tráfego é essencial, pois assegura a Qualidade de Serviço (QoS) da rede, além de permitir planear com eficiência o uso de recursos.

Com o aumento de tráfego cifrado ou protocolo ofuscado na Internet e do encapsulamento de dados multicamadas, alguns métodos clássicos da classificação perderam interesse de investigação da comunidade científica. As limitações dos métodos tradicionais da classificação com base no número da porta e na inspeção de carga útil *payload* para classificar o tráfego de Internet cifrado ou ofuscado levaram a esforços significativos de investigação com foco em abordagens da classificação baseadas em técnicas de Aprendizagem Automática (ML) usando recursos estatísticos da camada de transporte. Na tentativa de aumentar o desempenho da classificação, as estratégias de Aprendizagem Automática ganharam o interesse da comunidade científica e se mostraram promissoras no futuro da classificação de tráfego, principalmente no reconhecimento de tráfego cifrado.

No entanto, a abordagem em ML também têm as suas próprias limi- tações, pois alguns desses métodos possuem um elevado consumo de recursos computacionais, o que limita a sua aplicação para classificação de grandes fluxos de tráfego ou em tempo real. As limitações no âmbito da aplicação de ML levaram à investigação de abordagens alternativas, incluindo procedimentos baseados em características e métodos estatísticos. Neste sentido, os métodos de análise estatística, tais como distâncias e divergências, têm sido utilizados para classificar tráfego em grandes fluxos e em tempo real.

A distância estatística possui como objetivo principal diferenciar os fluxos e permite encontrar um padrão nas características de tráfego através de propriedades estatísticas, que possibilitam a classificação. As divergências são expressões funcionais frequentemente relacionadas com a teoria da informação, que mede o grau de discrepância entre duas distribuições quaisquer.

Esta tese foca-se na proposta de uma nova abordagem metodológica para classificação de tráfego cifrado ou ofuscado da Internet com base em métodos estatísticos que possibilite avaliar o desempenho da classificação de tráfego de rede, incluindo a utilização de recursos computacionais, em termos de CPU e memória. Foi proposto um conjunto de classificadores de tráfego baseados nas Divergências de Kullback-Leibler e Jensen-Shannon e Distâncias Euclidiana, Hellinger, Bhattacharyya e Wootters. A seguir resumem-se os

quatro principais contributos para o avanço do conhecimento científico reportados nesta tese.

Primeiro, realizámos uma ampla revisão de literatura sobre classificação de tráfego cifrado e ofuscado de Internet. Os resultados sugerem que os métodos baseados em porta e baseados em carga útil estão se tornando obsoletos em função do crescimento da utilização de cifragem de tráfego e encapsulamento de dados multicamada. O tipo de métodos baseados em ML também está se tornando limitado em função da complexidade computacional. Como alternativa, pode-se utilizar a Máquina de Vetor de Suporte (SVM), que também é um método de ML, e os testes de Kolmogorov-Smirnov e Qui-quadrado como referência de comparação da classificação estatística. Em paralelo, surgiu na literatura a possibilidade de utilização de métodos estatísticos para classificação de tráfego de Internet, com potencial de bons resultados na classificação sem aporte de grandes recursos computacionais. Os métodos estatísticos potenciais são as Distâncias Euclidiana, Hellinger, Bhattacharyya e Wootters, além das Divergências de Kullback–Leibler (KL) e Jensen-Shannon.

Segundo, apresentamos uma proposta e implementação de um classificador baseado na Máquina de Vetor de Suporte (SVM) para o tráfego multimédia P2P (Peer-to-Peer), comparando os resultados com os testes de Kolmogorov-Smirnov (KS) e Qui-quadrado. Os resultados sugerem que a classificação da SVM com kernel Linear conduz a um melhor desempenho da classificação do que os testes KS e Qui-quadrado, dependente do valor atribuído ao parâmetro Self C. O método SVM com kernel Linear e com valores adequados para o parâmetro Self C pode ser uma boa escolha para identificar o tráfego Par a Par (P2P) multimédia cifrado na Internet.

Terceiro, apresentamos uma proposta e implementação de dois classificadores baseados na Divergência de KullbackLeibler (KL) e na Distância Euclidiana, sendo comparados com a SVM com kernel Linear, configurado para o parâmestro Self C padrão, apresenta reduzi- da capacidade de classificar fluxos com base apenas nos tamanhos dos pacotes em relação aos métodos KL e Distância Euclidiana. Os métodos KL e Euclidiano foram capazes de classificar todas as aplicações testadas, destacando-se streaming e P2P, onde para quase todos os casos foi eficiente identificá-las com alta precisão, com reduzido consumo de recursos computacionais.Com base nos resultados obtidos, pode-se concluir que os métodos KL e Distância Euclidiana são uma alternativa à SVM, porque essas abordagens estatísticas podem operar em tempo real e não precisam de retreinamento cada vez que surge um novo tipo de tráfego.

Quarto, apresentamos uma proposta e implementação de um conjunto de classificadores para o tráfego de Internet cifrado, baseados na Divergência de Jensen-Shannon e nas Distâncias de Hellinger, Bhattacharyya e Wootters, sendo os respetivos resultados comparados com os resultados obtidos com os métodos baseados na Distância Euclidiana, KL,

KS e Qui-quadrado. Além disso, apresentamos uma análise qualitativa comparativa dos métodos testados com base nos valores de Kappa e Curvas Característica de Operação do Receptor (ROC). Os resultados sugerem valores médios de precisão acima de 90% para todos os métodos estatísticos, classificados como "confiabilidade quase perfeita" em valores de Kappa, com exceçãode KS. Esse resultado indica que esses métodos são opções viáveis para a classificação de tráfego cifrado da Internet, em especial a Distância de Hellinger, que apresentou os melhores resultados do valor de Kappa em comparaçãocom os demais classificadores. Conclui-se que os métodos estatísticos considerados podem ser precisos e económicos em termos de consumo de recursos computacionais para classificar o tráfego da rede.

A nossa abordagem baseou-se na classificação de tráfego de rede Internet, focando em distâncias e divergências estatísticas. Nós mostramos que é possível classificar e obter bons resultados com métodos estatísticos, equilibrando desempenho de classificação e uso de recursos computacionais em termos de CPU e memória. A validação da proposta sustenta o argumento desta tese, que propõe a implementação de métodos estatísticos como alternativa viável à classificação de tráfego da Internet em relação aos métodos com base no número da porta, na inspeção de carga útil e de ML.

Palavras-chave

Classificação de tráfego, tráfego de Internet cifrado, Divergência de Kullback-Leibler, Distância Euclidiana, Máquina de Vetor de Suporte, métodos estatísticos, distribuição, Distância estatística, Divergência estatística, streaming de vídeo Par a Par, Divergência de Jensen-Shannon, Distância de Hellinger, Distância de Bhattacharyya, Distância de Wootters.

Resumo Alargado

Introdução

O presente resumo alargado, em língua Portuguesa, descreve a tese de doutoramento intitulada *"Timely Classification of Encrypted or Protocol-Obfuscated Internet Traffic Using Statistical Methods"*. Começa por apresentar uma breve explicação sobre a classificação de tráfego Internet cifrado ou com protocolo ofuscado centrando-se, em seguida, na apresentação do foco da tese no argumento da tese e nas temáticas mais relevantes a serem abordadas. Este resumo alargado termina com uma breve conclusão e a apresentação de futuras linhas de investigação.

Classificação de Tráfego Internet Cifrado e Protocolo Ofuscado

A classificação de tráfego pode determinar a classe de tráfego ou protocolo, agrupando-os e relacionando-os de acordo com a categoria, tornando-se essencial como técnica para controlar e proteger a rede, além de poder prever e identificar o comportamento do utilizador na rede. O objetivo da classificação de tráfego da Internet é facilitar a gestão de rede. Os mecanismos existentes para classificação de tráfego usam diferentes métodos para determinar o protocolo ou aplicação e correlacionar as propriedades de tráfego.

Os quatro principais métodos para classificação são [1–3]: abordagens baseadas em portas, abordagens baseadas em carga útil ou *payload* (DPI - Deep Packet Inspection), abordagens baseadas em Aprendizagem Automatizada (ML) e abordagens baseadas em estatísticas ou estatísticas comportamentais (conjuntos de heurísticas). Os métodos baseados em portas permitem classificar o protocolo ou aplicação com referência ao número da porta, que a aplicação ou protocolo usa com base na Autoridade de Números Atribuídos da Internet (IANA) [1]. Os métodos baseados em DPI permitem classificar examinando a carga útil [4]. Os métodos de ML baseiam-se na aprendizagem de padrões ou modelos das características de tráfego [5]. Os métodos baseados em estatísticas ou estatísticas comportamentais permitem classificar através de propriedades de fluxo de tráfego das camadas de rede e transporte, tais como tamanho do pacote, entropia, tempo de chegada entre pacotes *jitter*, duração dos fluxos, dentre outros [6].

Os métodos de classificação tradicionais como os métodos baseados em portas possuem limitações, pois não podem classificar protocolos ou aplicações que fazem uso de portas com números aleatórios ou desconhecidos. Outro método tradicional que também possui limitações é a abordagem baseada em DPI. No cenário atual dominado pelo tráfego de rede cifrado [7–9] ou pelo uso de protocolos ofuscados [10], os métodos baseados em carga útil perdem interesse e geralmente são ineficazes. A cifragem de tráfego pode ser definida como um conjunto de técnicas aplicadas para codificar o formato original dos da-

dos da Internet, garantindo privacidade e segurança [7]. A ofuscação de protocolo consiste na modificação de propriedades que podem ser medidas em nível de carga ou fluxo, impossibilitando a identificação do protocolo [10]. Para ofuscação de carga útil, a cifragem é normalmente usada para fazer com que os dados apareçam como aleatórios. A ofuscação em nível de fluxo ocorre quando propriedades estatísticas, como tamanhos de pacotes e tempos de chegada entre pacotes são modificadas [10–12].

Os métodos que utilizam a abordagem ML têm apresentado alta complexidade em sua implementação e muitas vezes operam offline, tendo limitações para classificar o tráfego Internet online. Além disso requerem retreino quando são usadas novas aplicações ou protocolos [13].

A classificação baseada em estatísticas ou comportamento estatístico usa parâmetros das camadas de rede e transporte e propriedades estatísticas de protocolos, fluxos e aplicações [2, 14]. Usando propriedades estatísticas, o método pode classificar o tráfego da Internet sem a necessidade de analisar a carga útil, mesmo que a carga útil do pacote seja cifrada, e sem aumentar os problemas de segurança ou privacidade.

Foco da Tese

O foco da investigação descrita ao longo desta tese consiste na abordagem de métodos estatísticos para classificação de tráfego na Internet. A fim de apresentar uma nova abordagem metodológica que explora métodos estatísticos para classificar o tráfego cifrado ou ofuscado, investigou-se o uso de análise estatística, nomeadamente distâncias e divergências estatísticas.

Argumento da Tese

Esta tese propõe uma nova abordagem metodológica para classificação de tráfego de Internet cifrado ou usando protocolo ofuscado com base em métodos estatísticos. Assim, é proposto um conjunto de classificadores baseados na Divergência de Kullback-Leibler, Divergência de Jensen-Shannon, Distância Euclidiana, Distância de Hellinger, Distância de Bhattacharyya e Distância de Wootters. O argumento da tese é o seguinte:

As propriedades estatísticas de tráfego Internet fornecem um aspecto característico relevante para identificar as aplicações e protocolos, formando uma identificação adequada, nomeadamente a frequência relativa do tamanho de pacotes. Essa identificação é apropriadamente explorada ao usar métodos estatísticos, como distâncias e divergências. Métodos de análise estatística, como Divergência de Kullback-Leibler, Divergência de Jensen-Shannon, Distância Euclidiana, Distância de Hellinger, Distância de Bhattacharyya e Distância de Wootters, podem ser utilizados como uma boa alternativa na classificação de tráfego cifrado, sem a necessidade de utilização da carga útil do pacote e

consumo adequado de recursos computacionais.

Processo da Classificação de Tráfego Baseado em Estatística

Como alternativa aos métodos tradicionais, abordamos métodos baseados em estatísticas. De acordo com [15] mesmo que algumas técnicas de ofuscação sejam empregadas, é possível classificar o tráfego com base em características estatísticas. Os métodos baseados em estatística podem ser categorizados em paramétricos e não paramétricos.

A categoria de métodos paramétricos inclui SVM com kernel Linear Distância Euclidiana, Correlação de Pearson e Divergência de Jensen-Shannon. A categoria de métodos não paramétricos inclui SVM não Linear, Distância de Bhattacharyya, Distância de Hellinger, Divergência de KL, Distância de Wootters, e testes de Kolmogorov-Smirnov (KS) e Qui-quadrado. Os classificadores baseados em métodos paramétricos possuem, para cada classe, uma distribuição estatística de probabilidade. A distribuição estatística de probabilidade descreve o comportamento aleatório de um fenómeno dependente do acaso.

Já os classificadores não paramétricos são usados para estimar a distribuição estatística de probabilidade, ou casos em que a função densidade é desconhecida. A função densidade é aquela que descreve o que aparenta ser ou é tido como verdadeiro de uma variável aleatória. Para dar suporte metodológico ao nosso mecanismo da classificação, nós propomos e usamos a taxonomia de métodos da classificação mostrada na Figura 1.

O processo da classificação de tráfego baseado em estatística consiste nas seguintes fases: categorização de tráfego de Internet (coleta dos dados), conjunto de dados, recursos (*features*), abordagem da classificação e validação. A coleta de dados em uma rede é um ponto crítico e serve como entrada para formar uma base de tráfego de rede. A extração e a seleção da *feature* é um processo vital, pois pode afetar a eficiência e a eficácia da classificação. A *feature* é o conjunto de características necessárias para dar início ao processo da classificação. A abordagem escolhida para classificar o tráfego é essencial para o sucesso da classificação, assim como os critérios de avaliação de desempenho do classificador.

O tráfego Internet é categorizado de acordo com as seguintes classes: Administração, Comunicações, Jogos, Partilha de Ficheiros Mercados, Redes Sociais, Entretenimento em Tempo Real, Armazenamento, Encapsulamento e Navegação na Web. Cada categoria tem uma descrição, que caracteriza o tráfego associado.

O conjunto de dados tem grande importância na avaliação e desempenho dos métodos. Um conjunto de dados deve conter muitas e diversificadas amostras de cada classe de tráfego de Internet. Se a amostra de dados é pequena, ou com poucas categorias de classes de tráfego o classificador pode se ajustar a um número muito restrito e específico de amostras, gerando um classificador com viés.

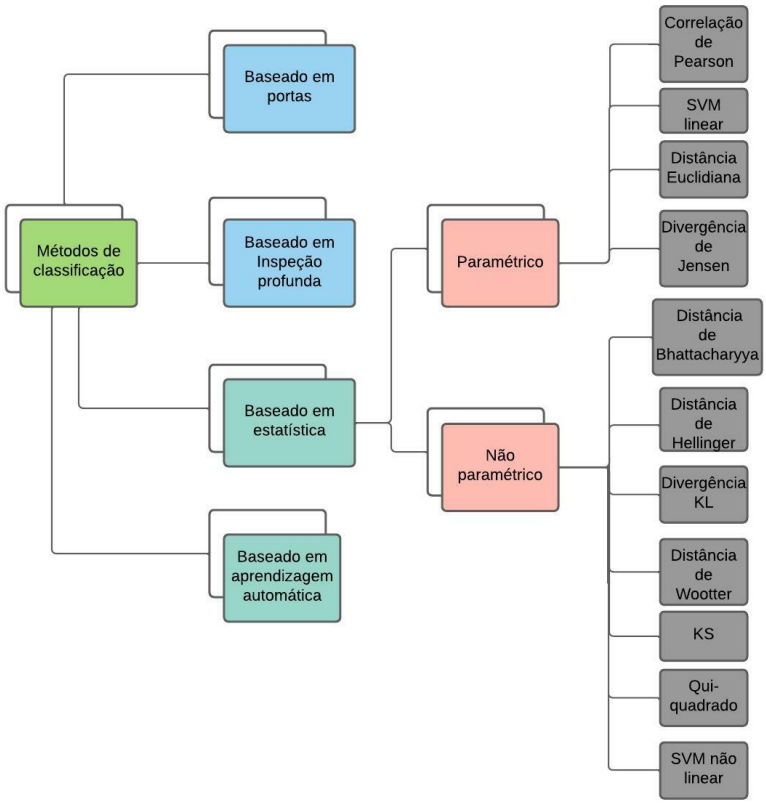


Figura 1: Visão geral de métodos estatísticos para classificação de tráfego da Internet.

Uma etapa importante na classificação de tráfego de Internet é a seleção da *feature*. A *feature* é definida como o processo de seleção do conjunto de características necessárias para alcançar uma classificação precisa. A classificação de diferentes categorias de classe de aplicações ocorre quando há algumas discrepâncias ou diferenças no comportamento de tráfego com base nas *features* selecionadas. Para a classificação, podem ser escolhidas uma ou mais *features*. Normalmente, a escolha da *features* muitas vezes ocorre apenas de forma qualitativa.

Nesta tese, para classificar o tráfego cifrado ou ofuscado foram utilizadas cinco tuplas: Protocolo de Datagrama de Utilizador (UDP)/ Protocolo de Controlo da Transmissão (TCP), Protocolo de Internet (IP) de origem/destino e endereços e números de portas. São *features* interessantes para abordagens estatísticas: i) tamanho dos pacotes, ii) número de pacotes, iii) estatística descritiva do tamanho do pacote (desvio padrão, variância etc.), iv) frequência acumulada do comprimento dos pacotes, v) frequência relativa do tamanho dos pacotes, vi) número de contagem de fluxos, vii) comprimento médio de fluxo, viii) comprimento máximo de fluxos, ix) tempo de chegada entre os pacotes e x) atraso entre os pacotes. Além destas *features*, podem ser identificadas outras passíveis de mensuração.

O processo de validação e avaliação do desempenho consiste em comparar os resultados

obtidos a partir da classificação com resultados previamente conhecidos, de modo a obter o desempenho da classificação.

Nesse sentido, os resultados da classificação obtidos são comparados com os resultados reais da classificação, obtidos previamente e de forma manual, sendo este processo conhecido como *ground truth*. O *ground truth* é a informação observada e medida, que permite calcular as taxas de verdadeiro positivo, falso positivo, verdadeiro negativo e falso negativo. Muitas medidas de desempenho são usadas para avaliar se um método de classificação pode alcançar o desempenho esperado como Accuracy, Precision, Recall e F-Measure, Curvas Característica de Operação do Receptor (ROC) e suas e respetivas Área Sob as Curvas (AUCs). A seguir serão apresentados os métodos utilizados para dar suporte a esta tese e as métricas de desempenho para validação e avaliação de desempenho do conjunto de classificadores.

Distância Euclidiana

A Distância Euclidiana, $D_E[x, y]$, entre dois pontos, x e y, em um, dois, três ou mais espaços dimensional é dada pela equação 1 [16], onde $D_E[x, y]$ representa a função de distância, N define o número de amostras, k define o número da amostra inicial, x_k representa o primeiro conjunto de amostras e y_k representa o segundo conjunto de amostras:

$$D_E[x, y] = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}. \quad (1)$$

Divergência de Jensen-Shannon

Para duas distribuições de probabilidade discretas $P = (p_1, p_2, ..., p_n)$ e $Q = (q_1, q_2, ..., q_n)$ com $pi \geq 0$, $qi \geq 0$, a Divergência de Jensen-Shannon (JSD) é representada pela equação 2 [17], onde N define o número de amostras, e i define o número da amostra inicial:

$$JSD\,(P, Q) = \quad \frac{1}{2}\left\{\sum_{i=1}^{N} pi \log\left(\frac{2pi}{pi + qi}\right) + \sum_{i=1}^{N} qi \log\left(\frac{2qi}{pi + qi}\right)\right\}. \quad (2)$$

Distância de Bhattacharyya

A Distância de Bhattacharyya é definida pela equação 3 [18], onde N é a quantidade de partições e pi e qi são a quantidade de membros da amostra e i representa o número inicial

da amostra:

$$Bc_D\left(P,Q\right) = \quad -\log\left(\sum_{i=1}^{N}\sqrt{pi \quad \times \quad qi}\right). \quad (3)$$

Distância de Hellinger

A função de cálculo é obtida a partir de duas distribuições de probabilidade p e q dado pela equação 4 [19], onde N define o número de amostras, e i define o número da amostra inicial.

$$H_D\left(P,Q\right) = \sqrt{\frac{1}{2}\sum_{i=1}^{N}\left(\sqrt{pi} - \quad \sqrt{qi}\right)^2}. \quad (4)$$

Divergência de Kullback-Leibler

Considerando que $D_{kl}[p||q]$ é uma função e p_i e q_i são duas distribuições de probabilidade, temos 5 [20]:

$$D_{kl}[p||q] \quad = \quad \sum pi\log\left(\frac{1}{pi}\right) - \sum pi\log\left(\frac{1}{qi}\right). \quad (5)$$

Então, podemos definir que a Divergência de Kullback-Leibler (KL) é representada pela equação 6:

$$D_{kL}[p||q] \quad = \quad \sum_{i=1}^{N} pi\log\left(\frac{qi}{pi}\right). \quad (6)$$

Onde $D_KL[p||q] >= 0$ e $D_KL[p||q] = 0$ se e somente se $p_i(x) = q_i(x)$, N define o número de amostras, i define o número da amostra inicial, pi define as frequências relativas da classe conhecida, qi define a frequência relativa da classe a ser comparada.

Distância de Wootters

A Distância de Wootters é definida pelas distribuições de probabilidade pi e qi, representada pela equação 7, onde o número de amostras é definido por N, e o número da amostra inicial é definido por i, [21]:

$$Wo_D\left(P,Q\right) = \arccos\left(\sum_{i=1}^{N}\sqrt{pi \quad \times \quad qi}\right). \quad (7)$$

Esta distância caracteriza-se por encontrar as diferenças das probabilidades inferiores aos valores das flutuações.

Máquina de Vetor de Suporte

A Máquina de Vetor de Suporte é um método popular de Aprendizagem Automática supervisionado, aplicado em classificação capaz de reconhecer padrões de amostras de classes predefinidas e suportar classificação multiclasse. Nós propomos, implementamos e avaliamos um classificador baseado na SVM para classificar o tráfego multimédia *Peer-to-Peer* (P2P). Para obter resultados relevantes, é necessário ajustar adequadamente o parâmetro Self C.

A SVM foi desenvolvida a partir da Teoria de Aprendizagem Estatística e tem como objetivo resolver problemas de classificação de padrões. A classificação baseada na SVM utiliza funções de kernel Linear, kernel *Radial Base Function* (RBF), kernel Polynomial e kernel Sigmoid. Para realizar a classificação de tráfego utilizando a SVM, é utilizada a arquitetura proposta na Figura 2. A classificação foi dividida em 3 etapas, da seguinte forma:

Etapa 1 - Tratamento dos dados - Geração da nova base de dados: para este passo, foi criado um script cujo objetivo foi converter a base de dados original em uma nova base de dados, que foi utilizada como entrada na SVM. Nós criamos *buckets* para calcular a distribuição da frequência relativa.

Etapa 2 - Fase de treino e teste: a base de dados gerada pelo script na etapa 1, é utilizada para gerar os modelos da SVM (fase de treino) e testar.

Etapa 3 - Validação e avaliação de desempenho.

Classificação Usando Distância Euclidiana e Divergência de Kullback-Leibler

As limitações dos métodos tradicionais da classificação com base no número da porta e na inspeção de carga útil para classificar o tráfego de Internet cifrado ou ofuscado levaram a esforços significativos de investigação com foco em abordagens de classificação baseadas em técnicas de Aprendizagem Automática usando recursos estatísticos da camada de transporte. No entanto, essas abordagens também têm as suas próprias limitações, levando à investigação de abordagens alternativas, tais como abordagens baseadas em estatísticas.

As abordagens estatísticas podem ser uma alternativa às de Aprendizagem Automática porque as abordagens estatísticas podem operar em tempo real e não precisam ser re-

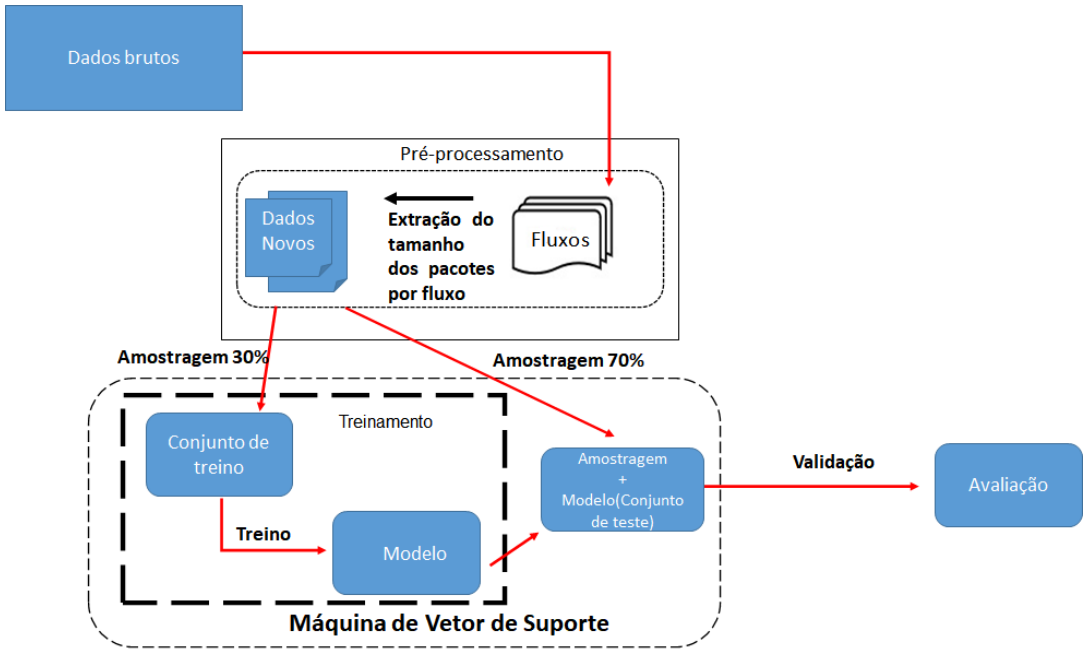


Figura 2: Arquitetura do classificador implementado com o método SVM.

treinadas cada vez que um novo tipo de tráfego aparece. Nesta etapa, propomos dois classificadores estatísticos para tráfego cifrado na Internet com base na Divergência de Kullback-Leibler e na Distância Euclidiana, que são calculados usando o fluxo e o tamanho do pacote obtidos de alguns dos protocolos utilizados pelas aplicações.

Nós propomos uma arquitetura para a classificação usando a Distância Euclidiana e a Divergência de KL. Esta arquitetura contém quatro módulos, conforme mostrado na Figura 3: captura e pré-processamento de pacotes, análise estatística e assinaturas armazenadas, classificação e validação.

Captura de pacotes e pré-processamento: primeiro, separamos o nosso conjunto de dados em ficheiros de fluxos individuais e ficheiros de fluxos coletivos. Classificamos como ficheiros de fluxos individuais aqueles em que sabemos quais as aplicações que foram usadas para gerá-los. Em seguida, consideramos o fluxo da aplicação, endereço IP de origem e endereço IP de destino, sequência de pacotes e número de pacotes pertencentes à mesma aplicação e endereços IP.

Geração de modelo empírico: nesta etapa, são gerados ficheiros de frequência relativa para alimentar a nova base de dados. Foi necessário criar um ficheiro com Frequências Relativas (FR) correspondentes aos fluxos com o tráfego conhecido.

Geração de amostras: as amostras foram geradas a partir de ficheiros de fluxos individuais e coletivos. Cada amostra representa um arquivo de frequência relativa gerado para cada protocolo. As frequências relativas dos fluxos individuais são armazenadas em um conjunto de dados diferente, pois servirão para fins de identificação de assinatura

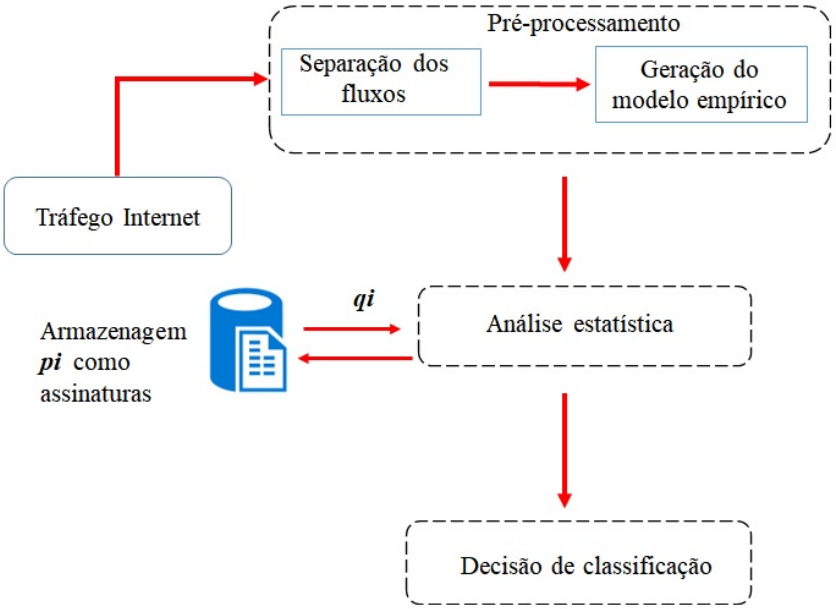


Figura 3: Proposta do classificador implementado usando a Distância Euclidiana e a Divergência de KL.

durante a fase de comparação de fluxos.

Análise Estatística e Assinaturas Armazenadas: nesta etapa são calculadas a distância e a divergência com base nas frequências relativas de cada fluxo. Para saber se dois fluxos de tráfego pertencem à mesma classe, usamos a distribuição empírica, no nosso caso a frequência relativa que cada fluxo possui, e comparamos as duas frequências aplicando o cálculo KL ou Euclidiano, e obtendo uma distância como saída.

Classificação e validação: a classificação dos fluxos com base em regras é feita nesta etapa. Observe que, nesta etapa, já temos todas as distâncias calculadas entre os fluxos.

Avaliação de desempenho: após o processo da classificação das amostras, verificamos e validamos os resultados da classificação utilizando *ground truth*.

Classificação usando métodos Jensen-Shannon, Bhattacharyya, Hellinger e Wootters

A classificação de tráfego Internet permite a identificação de protocolos utilizados em diferentes serviços da Internet, com base nas características apresentadas em pacotes ou fluxos gerados por esses serviços. Essa classificação e identificação de tráfego são realizadas através de diferentes técnicas, como Aprendizagem Automática (ML), Deep Packet Inspection (DPI), ou métodos estatísticos como distâncias e divergências, que têm sido usados para diferenciar objetos.

No entanto, os métodos de ML e DPI apresentam limitações nomeadamente para a clas-

sificação atempada de tráfego de Internet cifrado. Neste etapa, investigamos o uso de métodos estatísticos publicados na literatura que se mostraram bem-sucedidos para classificação em outras áreas, mas ainda não foram testados para classificação de tráfego de rede. Assim, propomos, implementamos e avaliamos um classificador baseado nos métodos de Jensen-Shannon, Hellinger, Bhattacharyya e Wootters para classificar o tráfego cifrado da Internet.

Para formar um classificador baseado no comportamento de tráfego, foi necessário uma amostra de distribuição de tráfego para que o algoritmo possa usar uma amostra de casos para os quais as classificações verdadeiras são conhecidas.

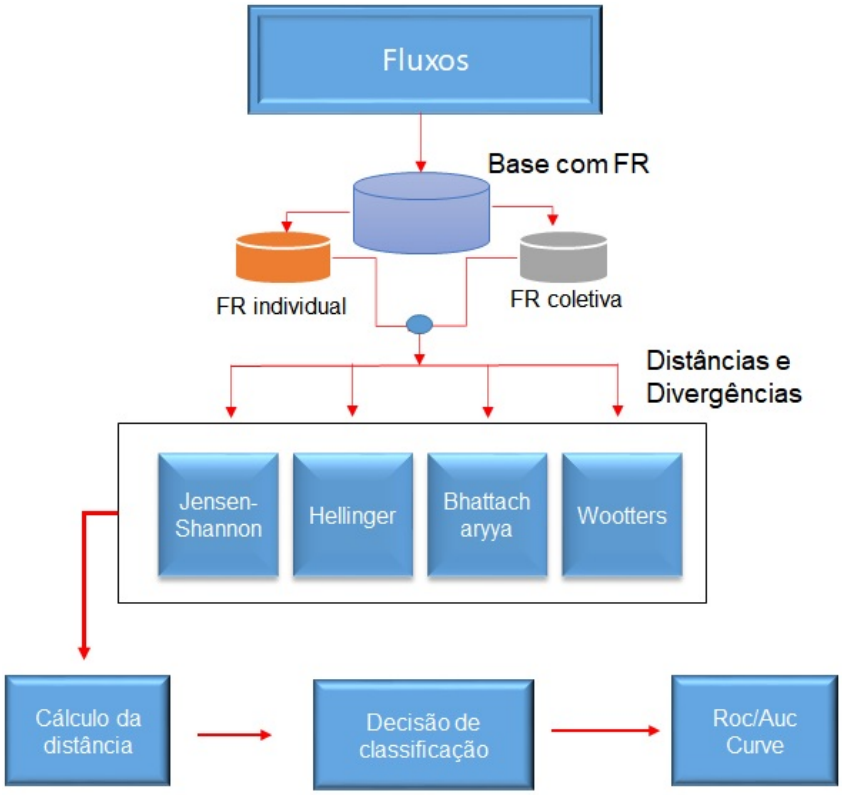


Figura 4: Arquitetura implementada para classificação de tráfego usando distâncias e divergências.

A Figura 4 mostra a arquitetura adotada para o processo da classificação, onde os fluxos foram pré-processados, e a distribuição do comprimento dos pacotes foi extraída de acordo com os fluxos, formando uma nova base de dados denominado ”Base com FR”. A arquitetura é composta por: ”Base com FR”, cálculo da distância, decisão da classificação e cálculo de ROC e suas AUCs.

Para a classificação, foi necessário definir a distribuição a ser utilizada (Jensen-Shannon, Hellinger, Bhattacharyya ou Wootters). As divergências e distâncias entre as distribuições de probabilidade foram calculadas por ”cálculos de distância” de acordo com o método utilizado. As saídas foram os valores das distâncias entre as distribuições. As regras fo-

ram aplicadas para a execução da ”decisão da classificação” após esses procedimentos, obtendo-se as saídas da classificação e as comparações entre os métodos (via curvas ROC e suas AUCs).

Desempenho dos Classificadores

Para avaliação de desempenho de cada um dos classificadores implementados nesta tese, usamos as métricas de desempenho clássicas definidas em livros didáticos de Aprendizagem Automática, sendo elas: Accuracy, Precision, Recall e F-Measure [22]. As métricas 8, 9, 10, 11 estão relacionadas com o desempenho dos métodos, sendo que, TP significa Verdadeiro Positivo, TN significa Verdadeiro Negativo, FP significa Falso Positivo e FN significa Falso Negativo.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \quad (8)$$

$$Recall = \frac{TP}{TP + FN}, \quad (9)$$

$$Precision = \frac{TP}{FP + TP}, \quad (10)$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall}. \quad (11)$$

Nesta tese também foi avaliado o desempenho computacional de todos os métodos implementados em termos do consumo de memória e de CPU. O desempenho de um método estatístico é considerado bom desde que o uso da CPU permaneça quase constante quando os pacotes estão sendo processados e analisados. Quanto ao uso de memória, ela flutua de acordo com as solicitações de cada método.

Os valores de Kappa foram usados para a avaliação qualitativa. O índice Kappa é um método estatístico para avaliar o nível de concordância ou reprodutibilidade entre os conjuntos de classificadores. Quanto maior a Accuracy, maior o índice Kappa. Os valores de Kappa foram calculados de acordo com as equações 12, e 13 conforme [23, 24]:

$$K = \frac{P_0 - P_e}{1 - P_e}. \quad (12)$$

Sendo que K = índice Kappa, P_0 é a taxa de aceitação relativa, P_e é a taxa hipotética de aceitação. Para estimar P_0, divide-se a soma das concordâncias (TP e TN) pela quan-

tidade total de indivíduos da amostra, que representa exatamente a Accuracy, definida pela equação 8.

Para estimar P_e é necessário calcular a probabilidade de ambos aleatoriamente aceitarem ou rejeitarem uma classificação de dados. Para isso, tem-se a seguinte equação 13:

$$P_e = \frac{\frac{(((TP+FP)*(TP+FN))+((FN+TN)*(FP+TN))}{(FP+TN+FN+TP)}}{(FP+TN+FN+TP)}. \quad (13)$$

Nós usamos duas interpretações diferentes: (1) [24] Landis e Koch e (2) McHugh [25]. As escalas não são totalmente sobrepostas, mas sugerem uma interpretação similar dos resultados. Para Landis e Koch, valores de Kappa acima de 0.41 já podem ser considerados moderados, enquanto para McHugh o valor de Kappa precisa alcançar no mínimo 0.60 para ser considerado moderado.

Na escala de Landis e Koch, um valor de Kappa acima de 0.80 já pode ser considerado com força de aceitação quase perfeita, enquanto para McHugh somente valores acima de 0.90 alcança nível de aceitação quase perfeita.

As escalas de Landis e Koch e McHugh são referências importantes para identificar a força e o nível de aceitação do conjunto de classificadores, sinalizando a qualidade das técnicas utilizadas na classificação de tráfego Internet cifrado ou ofuscado. A Tabela 1 apresenta a interpretação dos valores de Kappa sugeridos pelos autores Landis e Koch [24] e a Tabela 2 apresenta a interpretação dos valores de Kappa sugeridos pelo autor McHugh [25].

Tabela 1: Interpretação dos valores de Kappa de acordo com os autores Landis e Koch [24].

Estatística Kappa	Força de Aceitação
> 0.81	Quase perfeita
Entre $0.61 - 0.80$	Considerável
Entre $0.41 - 0.60$	Moderada
Entre $0.21 - 0.40$	Razoável
Entre $0.00 - 0.20$	Pouca
< 0.00	Ruim

Tabela 2: Interpretação dos valores de Kappa de acordo com o autor McHugh [25].

Valor de Kappa	Nível de Aceitação
> 0.90	Quase perfeita
Entre $0.80 - 0.90$	Forte
Entre $0.60 - 0.79$	Moderada
Entre $0.40 - 0.59$	Fraca
Entre $0.21 - 0.39$	Mínima
Entre $0.00 - 0.20$	Nenhuma

Principais Resultados

Após a obtenção dos resultados (*output*) fornecidos pelos classificadores, os resultados foram validados através de *ground truth* e avaliados usando a matriz de confusão, através das métricas Accuracy, Precision, Recall, e F-Measure. A Tabela 3 apresenta uma comparação entre os resultados da classificação obtidos com a SVM com kernels Linear e RBF, KS e testes Qui-quadrado para o tráfego P2P multimédia com alterações no parâmetro Self C.

Tabela 3: Resultados obtidos com a SVM Linear e RBF na melhor faixa do parâmetro Self C, KS e Qui-quadrado.

Desempenho	Método							
	Kernel Linear (C=[30 -70])		Kernel RBF (C=[50-70])		KS		Qui-quadrado	
	P2P file-sharing	P2P Video	P2P file-sharing	P2P Video	P2P file-sharing	P2P Video	P2P file-sharing	P2P Vídeo
Precision	97%	99%	91%	94%	84%	100%	91%	100%
Recall	99%	100%	94%	99%	56%	56%	80%	74%
F-Measure	98%	99%	92%	97%	67%	70%	85%	85%

A Tabela 4 apresenta os valores de Kappa alcançados e as avaliações qualitativas de acordo com Landis e Koch [24] e McHugh [25]. A Tabela 5 mostra a média e o desvio padrão da Accuracy, Precision, Recall e F-Measure obtidos pelos classificadores com base no teste KS, Distância Euclidiana, Divergência KL, Distância Wootters, Divergência de Jensen-Shannon, Teste do Qui-quadrado, Distância de Bhattacharyya e Distância de Hellinger.

Tabela 4: Qualidade da Classificação Associada aos Valores Estatísticos Kappa.

Classificador	Kappa	Avaliação qualitativa do classificador	
		Landis e Koch	**McHugh**
KS	0.72278	considerável	moderado
Euclidiana	0.82447	quase perfeita	forte
KL	0.83213	quase perfeita	forte
Jensen-Shannon	0.84363	quase perfeita	forte
Wootters	0.84371	quase perfeita	forte
Bhattacharyya	0.84540	quase perfeita	forte
Qui-quadrado	0.84910	quase perfeita	forte
Hellinger	0.85225	quase perfeita	forte

Os valores de Kappa foram usados para a avaliação qualitativa. Os valores obtidos através de Curvas ROC e suas AUCs foram ferramentas úteis e poderosas para a avaliação dos modelos de classificação.

Principais Conclusões e Linhas de Investigação Futura

Esta tese focou-se na proposta de uma nova abordagem metodológica para classificação de tráfego Internet cifrado ou ofuscado baseada em métodos estatísticos que visaram ser

Tabela 5: Resultados da Classificação Obtidos com Métodos Estatísticos.

	Accuracy		**Precision**		**Recall**		**F-Measure**	
Classificador	Média	Desvio padrão	Média	Desvio padrão	Média	Desvio padrão	Média	Desvio padrão
KS	0.99930	0.00062	0.80623	0.38551	0.48663	0.29667	0.58979	0.31884
Euclidiana	0.99967	0.00045	0.92188	0.26276	0.72309	0.27342	0.79668	0.26601
KL	0.99967	0.00049	0.92170	0.26284	0.73255	0.28943	0.79920	0.27535
Wootters	0.99967	0.00049	0.92222	0.26263	0.73646	0.28974	0.80191	0.27575
Jensen-Shannon	0.99969	0.00047	0.91442	0.26317	0.75649	0.28259	0.81441	0.27165
Qui-quadrado	0.99970	0.00046	0.92200	0.26271	0.75737	0.26271	0.81768	0.27121
Bhattacharyya	0.99971	0.00045	0.92160	0.26265	0.76178	0.28261	0.82043	0.27164
Hellinger	0.99971	0.00045	0.92280	0.26241	0.76293	0.28259	0.82175	0.27181

similar ou melhor que o desempenho da classificação usando a Máquina de Vetor de Suporte (SVM) com recursos computacionais adequados, em termos de CPU e memória. Foi proposto, implementado e avaliado um conjunto de classificadores estatísticos baseados em distâncias e divergências, em específico, Distância Euclidiana, Divergência de Kullback-Leibler (KL), Divergência de Jensen-Shannon, Distância de Wootters, Distância de Hellinger e Distância de Bhattacharyya. Para além disso, para fins de comparação, foram propostos, implementados e avaliados um classificador baseado na SVM, um classificador baseado no teste de Kolmogorov-Smirnov e um classificador baseado no teste do Qui-quadrado.

Verificámos que classificadores implementados através de métodos estatísticos são capazes de superar algumas limitações como complexidade computacional de recursos, quantidade das *features* utilizadas para classificação, operação em tempo real, grande quantidade de fluxos ou tráfego intenso.

Concluímos que os métodos estatísticos baseados em divergências ou distâncias podem ter uma boa precisão e uma utilização de recursos computacionais compatível com a classificação de tráfego Internet cifrado ou ofuscado, confirmando a hipótese inicial, validando o argumento apresentado nesta tese com expectativa que sejam reconhecidos como ferramentas úteis e confiáveis na classificação de tráfego Internet.

Alguns pontos em aberto identificados na revisão de literatura não foram solucionados, deixando espaço para avanços interessantes nesta área. Por exemplo, explorar a combinação do classificador SVM com divergências estatísticas. Na literatura encontramos trabalhos que combinam a Distância Euclidiana com o algoritmo K-means para classificadores, e a divergência de Kullback-Leibler combinada com a SVM. No entanto, não encontramos classificadores combinados com a Distância de Hellinger, com a Distância de Wootters e nem com a Divergência de Jensen-Shannon. Uma outra possibilidade é explorar a viabilidade do uso da similaridade de Manhattan, Mahalanobis e Minkowski para classificação de tráfego Internet cifrado. Outra importante possibilidade é usar classificadores baseados em distâncias e divergências para identificar *malware*, intrusões e

outros ataques.

Abstract

Internet traffic classification aims to identify the type of application or protocol that generated a particular packet or stream of packets on the network. Through traffic classification, Internet Service Providers (ISPs), governments, and network administrators can access basic functions and several solutions, including network management, advanced network monitoring, network auditing, and anomaly detection. Traffic classification is essential as it ensures the Quality of Service (QoS) of the network, as well as allowing efficient resource planning.

With the increase of encrypted or obfuscated protocol traffic on the Internet and multilayer data encapsulation, some classical classification methods have lost interest from the scientific community. The limitations of traditional classification methods based on port numbers and payload inspection to classify encrypted or obfuscated Internet traffic have led to significant research efforts focused on Machine Learning (ML) based classification approaches using statistical features from the transport layer. In an attempt to increase classification performance, Machine Learning strategies have gained interest from the scientific community and have shown promise in the future of traffic classification, specially to recognize encrypted traffic.

However, ML approach also has its own limitations, as some of these methods have a high computational resource consumption, which limits their application when classifying large traffic or real-time flows. Limitations of ML application have led to the investigation of alternative approaches, including feature-based procedures and statistical methods. In this sense, statistical analysis methods, such as distances and divergences, have been used to classify traffic in large flows and in real-time.

The main objective of statistical distance is to differentiate flows and find a pattern in traffic characteristics through statistical properties, which enable classification. Divergences are functional expressions often related to information theory, which measure the degree of discrepancy between any two distributions.

This thesis focuses on proposing a new methodological approach to classify encrypted or obfuscated Internet traffic based on statistical methods that enable the evaluation of network traffic classification performance, including the use of computational resources in terms of CPU and memory. A set of traffic classifiers based on Kullback-Leibler and Jensen-Shannon divergences, and Euclidean, Hellinger, Bhattacharyya, and Wootters distances were proposed. The following are the four main contributions to the advancement of scientific knowledge reported in this thesis.

First, an extensive literature review on the classification of encrypted and obfuscated

Internet traffic was conducted. The results suggest that port-based and payload-based methods are becoming obsolete due to the increasing use of traffic encryption and multi-layer data encapsulation. ML-based methods are also becoming limited due to their computational complexity. As an alternative, Support Vector Machine (SVM), which is also an ML method, and the Kolmogorov-Smirnov and Chi-squared tests can be used as reference for statistical classification. In parallel, the possibility of using statistical methods for Internet traffic classification has emerged in the literature, with the potential of good results in classification without the need of large computational resources. The potential statistical methods are Euclidean Distance, Hellinger Distance, Bhattacharyya Distance, Wootters Distance, as well as Kullback-Leibler (KL) and Jensen-Shannon divergences.

Second, we present a proposal and implementation of a classifier based on SVM for P2P multimedia traffic, comparing the results with Kolmogorov-Smirnov (KS) and Chi-square tests. The results suggest that SVM classification with Linear kernel leads to a better classification performance than KS and Chi-square tests, depending on the value assigned to the Self C parameter. The SVM method with Linear kernel and suitable values for the Self C parameter may be a good choice to identify encrypted P2P multimedia traffic on the Internet.

Third, we present a proposal and implementation of two classifiers based on KL Divergence and Euclidean Distance, which are compared to SVM with Linear kernel, configured with the standard Self C parameter, showing a reduced ability to classify flows based solely on packet sizes compared to KL and Euclidean Distance methods. KL and Euclidean methods were able to classify all tested applications, particularly streaming and P2P, where for almost all cases they efficiently identified them with high accuracy, with reduced consumption of computational resources. Based on the obtained results, it can be concluded that KL and Euclidean Distance methods are an alternative to SVM, as these statistical approaches can operate in real-time and do not require retraining every time a new type of traffic emerges.

Fourth, we present a proposal and implementation of a set of classifiers for encrypted Internet traffic, based on Jensen-Shannon Divergence and Hellinger, Bhattacharyya, and Wootters Distances, with their respective results compared to those obtained with methods based on Euclidean Distance, KL, KS, and Chi-Square. Additionally, we present a comparative qualitative analysis of the tested methods based on Kappa values and Receiver Operating Characteristic (ROC) curves. The results suggest average accuracy values above 90% for all statistical methods, classified as ”almost perfect reliability” in terms of Kappa values, with the exception of KS. This result indicates that these methods are viable options to classify encrypted Internet traffic, especially Hellinger Distance, which showed the best Kappa values compared to other classifiers. We conclude that the considered statistical methods can be accurate and cost-effective in terms of computational resource consumption to classify network traffic.

Our approach was based on the classification of Internet network traffic, focusing on statistical distances and divergences. We have shown that it is possible to classify and obtain good results with statistical methods, balancing classification performance and the use of computational resources in terms of CPU and memory. The validation of the proposal supports the argument of this thesis, which proposes the implementation of statistical methods as a viable alternative to Internet traffic classification compared to methods based on port numbers, payload inspection, and ML.

Keywords

Traffic classification, encrypted Internet traffic, Kullback-Leibler Divergence, Euclidean Distance, Support Vector Machine, statistical methods, distribution, statistical distance, statistical divergence, Peer-to-Peer video streaming, Jensen-Shannon Divergence, Hellinger Distance, Bhattacharyya Distance, Wootters Distance.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

A	Accuracy
AINA	Advanced Information Networking and Applications
AIDAE	Anti-Intrusion Detection Autoencoder
AUC	Area Under Curve
AIS	Artificial Immune Systems
AISVM	Attenuation factor Support Vector Machine
BoF	Bag of Flow
BIRCH	Balanced Iterative Reduction Clustering using Hierarchies
BGP	Border Gateway Protocol
BS	Behavioral Signatures
BW	ButterWorth
CMSVM	Called Cost Sensitive SVM
CFS	Correlation based Feature Selection
DDoS	Distributed Denial of Service
DL	Deep Learning
DM	Data Mining
DoS	Denial of Service
DPI	Deep Packet Inspection
DAGSVM	Directed Acyclic Graph Support Vector Machine
EC	Equal Coefficient
EEMD	Ensemble Empirical Mode Decomposition
ELM	Extreme Learning Machine
EKLM	Extreme Kernel Learning Machine
ENTMDL	Entropy based Minimum Description Length
ESSL	Enhanced SSL
FNR	False Negative Rate

FPR	False Positive Rate
FCFS	Fast Correlation based Feature Selection
FM	F-Measure
FN	False Negative
FP	False Positive
F1	Fscore
GMs	Gaussian Mixtures
GA	Genetic Algorithm
GA-WKELM	GA-Wavelet Kernel Extreme Learning Machine
G-mean	Geometric mean
DGC	Gravitation based Classification of Data
HD	Hellinger Distance
HNB	Hidden Naive Bayes
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
IANA	Internet Assigned Numbers Authority
IDGC	Imbalanced DGC
ISVM	Incremental SVM
IFA	Interest Flooding Attack
IoT	Internet of Things
IP	Internet Protocol
IDSs	Intrusion Detection Systems
ISP	Internet Service Provider
JSD	Jensen-Shannon Divergence
K	Kappa
KL	Kullback-Leibler
KNN	K-Nearest Neighbors

KS	Kolmogorov-Smirnov
LSSVM	Least Square SVM
MA	Moving Average
MAE	Mean Absolute Error
MF	Multifractal Formalism
ML	Machine Learning
MSD	Message Size Distribution
MSS	Maximum Sequence Size
MSE	Mean Square Error
MSSC	Message Size Sequence
MTU	Maximum Transmission Unit
NAPT	Network Address Port Translation
NN	Neural Network
N/A	Not Available
P	Precision
PCA	Principal Component Analysis
PCABFS	PCA Based Features Selection
PCIDSS	Payment Card Industry Data Security Standards
PDF	Probability Density Function
PPC	Packet Payload Content
PSO	Particle Swarm Optimization
P2P	Peer-to-Peer
QoS	Quality of Service
R	Recall
RBF	Radial Base Function
RTP	Real-Time Transport Protocol
ROI	Region Of Interest

RF	Relative Frequency
RSVMs	Relaxed Constraint Support Vector Machines
ROC	Receiver Operating Characteristic
RMSE	Root Mean Square Error
RTT	Round Trip Time
S	Sensitivity
SDN	SoftwareDefined Networks
SFS	Sequential Forward Selection
SIP	Session Initiation Protocol
SSL	Secure Sockets Layer
Sp	Specificity
SVs	Support Vectors
SVM	Support Vector Machine
TCP	Transmission Control Protocol
STIC	SVM for Traffic Identification and Classification
TLS	Transport Layer Security
TLS	Transport Layer Statistics
TN	True Negative
TP	True Positive
UDP	User Datagram Protocol
VM	Virtual Machine
WL	Wavelet
WLMF	Wavelet Leaders Multifractal Formalism
WK-ELM	Wavelet Kernel Extreme Learning Machine

Chapter 1

Introduction

1.1 Thesis Scope and Focus

Traffic classification has an important application on the current management of computer networks. Classifying Internet traffic means identifying the protocols and applications that generated such Internet traffic. Traditional methods of Internet traffic classification are based on techniques that have limitations in scenarios where the Internet traffic is encrypted [7–9] or the protocol is obfuscated [10]. Encrypted traffic has been getting more space on the Internet, because online privacy and security are highly dependent on encryption. Traffic encryption can be defined as a set of techniques applied to encode the original format of Internet data, ensuring privacy and security [7]. Nonetheless, encryption can also be applied to malicious codes, making it hard to identify them on networks and therefore allowing its traffic on the Internet if not classified and blocked. Protocol-obfuscation is the modification or removal of properties that can be measured at payload or flow level, which can even make impossible the identification of the protocol. For payload obfuscation, encryption is normally used to make data look random. Flow level obfuscation happens when statistical properties, such as packet lengths and time between arrivals, are hidden [10–12].

There are four main types of approaches to classify Internet traffic [1–3]: approaches based on ports, approaches based on payload, approaches based on Machine Learning (ML) and approaches based on statistics. Port based approaches identify applications and protocols through the ports mapped by the Internet Assigned Number Authority (IANA) [1]. Payload based approaches, also known as Deep Packet Inspection (DPI), identify applications and protocols through signatures or a set of strings in the packet payload [4]. ML based approaches identify applications and protocols through a learning model or patterns of learning differentiation [5], and statistical based approaches identify applications and protocols through statistical traffic parameters, statistical behavior, or heuristics and they are based on the monitoring of connection patterns of IP, port pairs [26] or statistics of packet lengths or at flow level [14].

Initially, Internet traffic identification was done through an easy way, using only port numbers of the applications or protocols. However, many applications started to use unknown or random ports, which is the case of Peer-to-Peer (P2P) protocols, or used well-known ports used by other protocols, e.g., port 80 for HTTP (Hypertext Transfer Protocol) traffic, making unfeasible the use of methods based on ports [26].

From that moment on, new strategies have been explored to identify Internet traffic. A proposed alternative were methods based on payload or DPI. DPI based methods have as characteristic the examination of packet content, independently of which port number the application uses to. However, DPI methods also have their own limitations, because they cannot recognize or identify encrypted content or protocol-obfuscated signatures in the payload [7]. Some studies and efforts were made to improve the efficiency of DPI based methods [8, 27–29]. In [30–33], the target of the research was methods and techniques that used payload or DPI to identify Internet traffic. Although many efforts have been made, DPI methods have difficulty overcoming limitations resulting from traffic being encrypted or obfuscated.

Aiming to overcome the limitations of DPI methods, studies and efforts were made based on ML methods applied to features at the packet or flow levels, as in [3,34–40]. However, methods using ML approach have presented high complexity in their implementation and many times they operate offline, being unable of classifying online Internet traffic.

Recently, attention has been paid to the development of approaches based on statistics that can be computationally efficient and that can work online or in a timely manner [2, 14, 41]. Studies made in [42, 43] had as focus the statistical behavior. The ones made in [26, 44–47] focused on heuristic approaches.

Classification based on statistics or statistical behavior uses parameters from the network and transport layers and statistical properties of protocols, flows and applications [2, 14]. Using statistical properties, the method can classify the Internet traffic without needing to analyze the payload, even if the packet payload is encrypted, and without rising security or privacy issues. Parameters of interest for this kind of classification include: i) packet length, ii) packet number, iii) descriptive statistics of the packet size (standard deviation, variance, etc), iv) accumulated frequency of packets length, v) relative frequency of packets size, vi) quantity of bursts, vii) average length of bursts, viii) maximum length of a burst, ix) inter-arrival times and x) delay between packets. Besides those parameters, others can be identified, also possible of being measured.

The focus of the research described along this thesis is to approach statistical methods for Internet traffic classification. In order to present a new methodological approach which explores statistical methods to classify encrypted or obfuscated traffic, the use of statistical analysis was investigated, namely statistical distances and divergences. A distance may allow the measurement of the difference between flows through statistical properties, differentiating them numerically. A divergence may allow the measurement of the discrepancy between probability distributions of the statistical properties of the flows. Therefore, this thesis is dedicated to encrypted or protocol-obfuscated Internet traffic classification using statistical methods.

1.2 Problem Statement

Data flow in the Internet has increased exponentially and with a multiplicity of formats, many of which makes the network control and security harder, such as protocols that use random or unknown ports and encrypted or obfuscated packets. Despite allowing content privacy, encrypted or obfuscated packets may contain malicious code, increasing the risks for users and stable functioning of the network and their interconnected systems. As a way to face this dilemma, it becomes necessary to develop methods for traffic classification that operate timely and are efficient without compromising security and privacy.

As reported in the previous section, there are four types of methods for traffic classification. Methods based on port numbers are nowadays obsolete since they are unable to correctly identify traffic that uses unknown or random ports or uses well-known ports assigned to other protocols. Besides, since the beginning of the Enhanced SSL (ESSL) service, which was originally created to be a separate network segment satisfying the requirements of the Payment Card Industry Data Security Standards (PCI-DSS), the use of encryption has become the standard for all kind of Internet traffic [48]. As a consequence, nowadays, 95% of Internet traffic uses the HTTP Secure (HTTPS) protocol as estimated by Google and around 80-90% of network traffic is encrypted, as reported by most industry analysts [49]. In this scenario dominated by encrypted network traffic, payload-based methods lose their strength and are often ineffective. This led to the search for new methods based on ML or on statistical approaches.

Methods based on ML are computationally complex and the supervised ones require a set of pre-classified data and a training phase, which requires retraining every time new applications or protocols are used [13]. This kind of classifiers are suitable for offline operation and may require additional mechanisms for timely or online classification, such as a sliding window.

Classifiers such as Bayesian estimation, C4.5, and nearest-neighbor estimation may be tied to local optimization and cannot work in real time due to their computational and storage requirements [50]. Support Vector Machines (SVMs), which may be seen as both an ML method and a statistical method, have also been widely investigated for classification of Internet traffic. SVMs have been used separately [37, 38, 51–54] or combined with other ML methods in order to improve performance or operation speed, such as Neural Network (NN) and Random Forest [55], Naive Bayes [56], k-Nearest Neighbors (KNN) [57], least square SVM with hybrid optimization algorithm [58], grid search and genetic algorithm [59] for SVM parameter optimization, nature inspired instance selection techniques [60] for SVM speed optimization, or multiclass SVM with active learning [61], or SVM used in a clustering based semi-supervised ML learning [62].

A few statistical methods have already been investigated for Internet traffic classifica-

tion, such as the Euclidean Distance, Pearson Correlation (Correlation Information), and Kolmogorov-Smirnov and Chi-square tests [63–69]. Despite being used separately, some of these methods are also used in conjunction with other statistical or ML methods in order to improve their performance.

By hypothesis, we believe that statistical models based on distances and divergences may be able to classify encrypted Internet traffic in an efficient and timely manner, with suitable computational resource consumption, showing potential for usage as new Internet traffic classification models. Therefore, the research question in the inception of the research work leading to this thesis is the following:

"How to obtain classification results for encrypted Internet traffic, similar or better than the ones obtained with ML-based models, namely with SVM, using statistical methods in a timely way?"

1.3 Research Objectives

The research work described in this thesis has as its main objective proposing, implementing, and evaluating a set of classifiers proposing, implementing and evaluating a set of classifiers for encrypted and protocol-obfuscated Internet traffic based on statistical methods, in specific Euclidean, Hellinger, Bhattacharyya and Wootters Distances, and Kullback-Leibler and Jensen-Shannon Divergences.

The methods explore statistical features such as the packet lengths of the Internet traffic flows and their performance can be evaluated in terms of Accuracy, Recall, Precision, F-Measure, Kappa index and the Curve of Receiver Operating Characteristic (ROC) curves with its Areas Under the Curves (AUCs), besides measuring the efficiency in computational terms, evaluating CPU elapsed time and necessary memory. To support the main purpose of this thesis, the following specific objectives were set:

- Presenting a literature and state of art review about classification of encrypted and protocol-obfuscated Internet traffic;
- Presenting a general view about statistical methods, in particular Euclidean Distance, Kullback-Leibler (KL) Divergence, Jensen-Shannon Divergence, Wootters Distance, Hellinger Distance and Bhattacharyya Distance;
- Implementing a lab testbed for classification of encrypted and protocol-obfuscated Internet traffic;
- Implementing a Support Vector Machine (SVM) based classifier;
- Implementing Chi-square and Kolmogorov-Smirnov test based classifiers;

- Implementing a set of classifiers based on distances and divergences, more specifically Euclidean Distance, Kullback-Leibler (KL) Divergence, Jensen-Shannon Divergence, Wootters Distance, Hellinger Distance and Bhattacharyya Distance;
- Analyzing the performance of the set of classifiers in terms of Precision, Recall, F-measure, Accuracy, Kappa index and ROC curves with its AUCs, and in terms of computational efficiency (CPU and memory usage);
- Comparing the performance of the implemented classifiers to other classifiers, such as SVM, Chi-square and Kolmogorov-Smirnov test through Kappa index and ROC curves with its AUCs.

1.4 Thesis Statement

This thesis proposes a new methodological approach for classification of encrypted or protocol-obfuscated Internet traffic based on statistical methods. Thus, it is proposed a set of classifiers based on the Kullback-Leibler Divergence, Jensen-Shannon Divergence, Euclidean Distance, Hellinger Distance, Bhattacharyya Distance, and Wootters Distance. The thesis statement is as follows:

Statistical properties of Internet traffic provide a characteristic aspect that has main relevance to identifying applications and protocols, the relative frequency of packet lengths, which forms an appropriate identification. This identification can be suitably explored by using statistical methods, such as distances and divergences. Methods of statistical analysis, like Kullback-Leibler Divergence, Jensen-Shannon Divergence, Euclidean Distance, Hellinger Distance, Bhattacharyya Distance, and Wootters Distance, can be used for encrypted traffic classification with suitable consumption of computing resources.

1.5 Adopted Approach for Solving the Problem

The construction process of the object of this Ph.D. thesis started with a wide research in several databases of academic articles, using as reference the key-words “encrypted and obfuscated Internet traffic classification”.

The limitations of methods based on ports, payload, and ML motivated us to search for a less complex approach that is able to provide good classification results, has a moderate resource consumption and is able to operate in real time.

Through the initial literature review, we verified that there were some open issues about traffic classification and that statistical methods, such as distances and divergences, showed themselves promising for classification in different areas, but have not been tested for encrypted or obfuscated Internet traffic classification yet. Like that, to each new reading

the research on the database started containing the key words: “statistical methods, divergences, distances, Kullback-Leibler, Jensen-Shannon, Euclidean, Hellinger, Wootters, Bhattacharyya, SVM, Chi-square, Kolmogorov-Smirnov, Kappa index, ROC curve and its AUCs, computational performance ”. Through the whole research, more than 700 materials were consulted from international databases, being 248 of them selected and cited in the bibliographic references.

With the purpose of this research well defined and aiming to solve the proposed question, the efforts were directed to implementing, evaluating and comparing a set of classifiers based on statistical methods with the potential to overcome dilemmas of random ports, encryption, obfuscation and computational complexity. Kullback-Leibler and Jensen-Shannon Divergences and Euclidean, Hellinger, Bhattacharyya and Wootters Distances were chosen trying to classify encrypted and obfuscated Internet traffic. Traffic classification by SVM, Kolmogorov-Smirnov (KS) and Chi-square tests were also chosen as reference to compare the many classifiers based on divergences and distances.

To test the statistical methodology, a published and available database was used with approximately 25GB of network traffic flow generated by 28 different Internet services and applications, captured using the tcpdump tool and stored in disk.

A script was built with all statistical models under study and implemented in Python, which estimated the capacity of classification of those models for the database. The estimates were generated and run successfully, looking towards refining the script from the statistical metrics suggested by literature, with interesting results when using accumulated and relative frequencies of packet length.

The results were organized considering the Accuracy, Recall, Precision and F-Measure for all 28 protocols and the eight statistical models that allowed the evaluation of the classification capacity of the different methods. An additional layer of evaluation was added when using the Kappa index and the ROC curve and its AUCs as a way to quantify the reliability and the strength of the classifiers.

With the results of the statistical methods capacity and the quality of classification, four articles were produced that contemplated since the proposal conception up to the evaluation of all methods, aiming to spread in the scientific community the potential of using statistical tools as a classifier for encrypted and obfuscated Internet traffic.

Finally, all material was gathered in this thesis format, organizing the articles as chapters in a coherent sequence, and having the methodology as the link among all the articles from the identification of the possibility of using statistical techniques to the evaluation of all the tested approaches.

1.6 List of Scientific Publications

This Ph.D. thesis includes the following research papers, as appeared in the conference proceedings or as recently submitted for review in international journals:

1. A Complete review on the application of statistical methods for evaluating Internet traffic Usage, Vanice Canuto Cunha, Arturo Zavala Zavala, Damien Magoni, Pedro R. M. Inácio, Mário M. Freire, IEEE Access, vol. 10, pp. 128433-128455, 2022. DOI:10.1109/ACCESS.2022.3227073 [70].
2. Impact of Self C Parameter on SVM-based Classification of Encrypted Multimedia Peer-to-Peer Traffic, Vanice Canuto Cunha, Damien Magoni, Pedro R. M. Inácio, Mario M. Freire. In: Barolli, L., Hussain, F., Enokido, T. (eds) Advanced Information Networking and Applications, AINA 2022, Lecture Notes in Networks and Systems, vol 449, Springer, Cham, pp. 180–193. DOI: https://doi.org/10.1007/978-3-030-99584-3_16 [71] .
3. Classification of Encrypted Internet Traffic Using Kullback-Leibler Divergence and Euclidean Distance, Vanice Canuto Cunha, Arturo A. Z. Zavala, Pedro R. M. Inácio, Damien Magoni, Mario M. Freire. In: Barolli, L., Amato, F., Moscato, F., Enokido, T., Takizawa, M. (eds) Advanced Information Networking and Applications, AINA 2020, Advances in Intelligent Systems and Computing, vol 1151, Springer, Cham, pp. 883–897. DOI: https://doi.org/10.1007/978-3-030-44041-1_77 [72].
4. Classification of Encrypted Internet Traffic Using Statistical Methods, Vanice Canuto Cunha, Arturo A. Z. Zavala, Pedro R. M. Inácio, Damien Magoni, Mário M. Freire, 2022, currently under submission.

Paper 1 [70] was published in a journal listed in Scimago Journal & Country Rank as Q1 and its original version, before revision, appears in Chapter 2 of this thesis.

Paper 2 [71] was presented in an international conference ranked as B in the CORE Conference Portal and appears with some updates in Chapter 3 of this thesis.

Paper 3 [72] was presented in an international conference ranked as B in the CORE Conference Portal and appears in Chapter 4 of this thesis.

Paper 4 is under submission in a journal listed in Scimago Journal & Country Rank as Q1 and appears in Chapter 5 of this thesis.

1.7 Main Scientific Contributions

This section briefly describes the main scientific contributions to the advancement of the state of the art resulting from the research work presented in this thesis.

The first contribution of this thesis consists of a wide literature review about the use of statistical methods for encrypted or obfuscated Internet traffic classification. The results suggest that methods based on ports and payload inspection have become obsolete because of the increase of P2P traffic using unknown or random ports, encrypted traffic and encapsulations of multilayer data. ML based methods have become limited because of their computational complexity. As an alternative, Support Vector Machine (SVM) and Kolmogorov-Smirnov and Chi-square tests can be used as reference to compare statistical classification. In parallel, it appeared in the literature the possibility of using statistical methods for Internet traffic classification, with potential for good classification results without the need of large computational resources. Potential statistical methods include Euclidean, Hellinger, Bhattacharyya and Wootters Distances, besides Kullback–Leibler (KL) and Jensen-Shannon Divergences. This study is described in chapter 2, which consists of the original version (before revision) of a paper published in IEEE Access (volume:10) [70].

The second contribution of this thesis consists of the proposal, implementation, and evaluation of a classifier based on a Support Vector Machine (SVM) for P2P multimedia traffic compared to the results of Kolmogorov-Smirnov (KS) and Chi-square tests. We proposed an SVM-based classifier that uses the relative frequency of the packet lengths of the protocols used by the application. The results suggest that the classification based on SVM with a Linear kernel is dependent on the value given to the Self C parameter and it presents a better classification performance than KS and Chi-square tests. Therefore, the SVM method with a Linear kernel and suitable values for the Self C parameter can be a good choice to identify encrypted P2P multimedia traffic on the Internet. This study is described in chapter 3, which is a paper included in the Proceedings of the 36th International Conference on Advanced Information Networking and Applications (AINA2022), published by Springer as part of the Lecture Notes in Networks and Systems book series [71].

The third contribution of this thesis consists of the proposal, implementation and evaluation of two classifiers based on Kullback-Leibler (KL) or Euclidean Distance compared to SVM. We proposed two statistical classifiers for encrypted Internet traffic based on Kullback-Leibler Divergence and Euclidean Distance, that are calculated using the flow and packet lengths obtained from some protocols used by applications. The results suggest that SVM method with Linear kernel set to default Self C parameter presents a reduced capacity of classifying flows based only on packet lengths when compared to KL and Euclidean methods. KL and Euclidean methods were capable of classifying all tested applications, P2P and streaming mostly, where in almost all cases they were efficient with a high precision level and a lower computational resource usage. It is shown that KL and Euclidean methods are an alternative to SVM because those statistical approaches can operate in real time and do not require to be retrained each time a new type of traffic surfaces. This study is described in chapter 4, which is a paper that appeared in the Proceedings of the 34th International Conference on Advanced Information Networking and

Applications (AINA 2020), published by Springer as part of the Advances in Intelligent Systems and Computing book series [72].

The fourth contribution of this thesis consists of the proposal, implementation and evaluation of a set of classifiers for encrypted Internet traffic based on Jensen-Shannon Divergence and Hellinger, Bhattacharyya and Wootters Distances compared to Euclidian, KL, KS and Chi-square tests. Besides that, a qualitative analysis comparing tested methods was presented based on Kappa index and its ROC curves. Results suggest an average Precision value over 90% for all statistical methods, classified as "almost perfect confiability" and "strong" in Kappa values, with the exception of KS (classified as "considerable" and "moderate"). This result indicates that those methods are viable options for encrypted Internet traffic classification, especially Hellinger, that presented the best results in Kappa values when compared to the other classifiers. This study is described in chapter 5, which consists of a paper under review in a scientific journal at the time of writing this Ph.D. thesis.

1.8 Thesis Organization

This thesis consists of a collection of papers and follows a similar organization to other theses based on papers, such as [73–78]. Therefore, except for the first and last chapters devoted to introduction and conclusions, respectively, each chapter of this thesis consists of a paper that has been published in conference proceedings or is under review in a peer-reviewed journal, with minor formatting adjustments to fit the layout of the dissertation. The remainder of this thesis is organized as follows:

- *Introduction*, the current chapter, introduces the scope and focus of the thesis, defines the problem to be addressed and the research objectives for the research work towards the PhD thesis, addresses the adopted approach for solving the problem, lists the Scientific Publications resulting from this research work, and highlights mains scientific contributions. This chapter ends with the organization of this document.

- *Chapter 2* provides an overview of the most relevant existing statistical methods for the classification of Internet traffic. An introduction to traffic classification and its main applications is provided, as well as the proposed taxonomy and classification methods. Section 2.2 presents an overview of the traffic classification process, as well as the data sets, resources and types of resources required for classification approaches. Section 2.3 presents an overview of parametric and non-parametric models, the concept and types of distances such as Euclidean, Hellinger, Bhattacharrya, Wootters and types of divergences such as Kullback-Leibler and Jensen-Shannon, as well as general properties for qualifying distances and divergences. The purpose of this section is to provide a brief description about the concepts of distance, divergence and SVM. Section 2.4 presents a detailed review and comparison of the

main studies found in the literature on traffic classification using statistical methods. Section 2.5 presents a brief discussion, and some important open issues for new approaches.

- *Chapter 3* addresses the classification of encrypted P2P multimedia traffic using Support Vector Machines. Section 3.1 presents an introduction to video streaming and the rationale for classifying encrypted multimedia P2P traffic. Section 3.2 presents recent studies to classify traffic based on SVMs. Section 3.3 provides background on SVM and presents the proposed methodology to the classifier based on SVM. Section 3.4 presents the results and the usage of the computational resources obtained for the traffic classification.

- *Chapter 4* presents the classification of traffic using Kullback-Leibler Divergence and Euclidean Distance. Section 4.1 presents an introduction to Internet traffic classification focused on statistical methods. Section 4.2 focuses on significant research found in the literature, which explored traffic flows or packet lengths or characteristics of packet lengths. Section 4.3 provides background on Kullback-Leibler Divergence and Euclidean Distance. The purpose of this section is to provide background on distance and divergence for classification traffic. Section 4.4 presents details on the approach to classifying Internet traffic, describing traffic features and classifier architecture used for classification. Section 4.5 addresses the performance of the classifier and the ground truth established.

- *Chapter 5* addresses the proposal, implementation, and evaluation of a set of classifiers based on Jensen-Shannon Divergence, Hellinger Distance, Bhattacharyya Distance, and Wootters Distance. Section 5.1 presents the importance of Internet traffic classification and the main activities and services used on the Internet. Section 5.2 summarizes the studies that use statistical information for classification. Section 5.3 presents an overview of statistical methods and flow statistical properties. Section 5.4 describes the proposal and implementation of a set of classifiers, as well as, the data sets and the used features. 5.5 discusses the main results of the set of classifiers and provides ROC curves and their AUCs analyses for the set of classifiers. This section provides classification results for the set of classifiers in terms of Kappa, Precision, Recall, Accuracy, and F-Measure for 28 tested applications.

- *Chapter 6* summarizes the main scientific contributions and presents the final conclusions of this thesis. In addiction, this chapter also presents research limitations and direction for future work.

Chapter 2

A Complete review on the application of statistical methods for evaluating Internet traffic Usage [1]

Internet traffic classification aims to identify the kind of Internet traffic. With the rise of traffic encryption and multi-layer data encapsulation, some classic classification methods have lost their strength. In an attempt to increase classification performance, Machine Learning (ML) strategies have gained the scientific community interest and have shown themselves promising in the future of traffic classification, mainly in the recognition of encrypted traffic. However, some of these methods have a high computational resource consumption, which make them unfeasible for classification of large traffic flows or in real-time. Methods using statistical analysis have been used to classify real-time traffic or large traffic flows, where the main objective is to find statistical differences among flows or find a pattern in traffic characteristics through statistical properties that allow traffic classification. The purpose of this chapter is to address statistical methods to classify Internet traffic that were little or unexplored in the literature. This chapter is not generally focused on discussing statistical methodology. It focuses on discussing statistical tools applied to Internet traffic classification.
Thus, we provide an overview on statistical distances and divergences previously used or with potential to be used in the classification of Internet traffic. Then, we review previous works about Internet traffic classification using statistical methods, namely Euclidean, Bhattacharyya, and Hellinger Distances, Jensen-Shannon and Kullback–Leibler (KL) Divergences, Support Vector Machines (SVM), Correlation Information (Pearson Correlation), Kolmogorov-Smirnov and Chi-square tests, and Entropy. We also discuss some open issues and future research directions on Internet traffic classification using statistical methods.

2.1 Introduction

Internet traffic classification may be used to solve several kinds of network issues. Through traffic classification, Internet Service Providers (ISP), governments, and network administrators can have access to network resource management, advanced network monitoring, network audit, anomaly detection, and device filtering [79].

[1] The content of this chapter consists of the original version, before revision, of the paper published in the following venue [70] Vanice Canuto Cunha, Arturo Zavala Zavala, Damien Magoni, Pedro R. M. Inácio, Mário M. Freire, "A Complete review on the application of statistical methods for evaluating Internet traffic Usage", IEEE Access, vol. 10, pp. 128433-128455, 2022. DOI:10.1109/ACCESS.2022.3227073.

Classifying traffic by categorizing network traffic according to its appropriate class is vital to many applications such as pricing, Quality of Service (QoS) control, malware/intrusion detection, and resource usage planning [80].

Due to the importance of classification, several approaches were thought with the development of different applications and scenarios. However, communication advances like encryption and port obfuscation added new challenges to network traffic classification [80].

According to Zhao *et al.* [81], to manage, detect intrusion, monitor the network security and classify the traffic in real time and in a precise way, the traffic classification is essential. Traffic classification determines the class of the data, grouping and relating them according to the category, making it essential as technique to control and secure the network, besides that, it can foresee and identify the user's behavior in the network [82]. The identification in the right way of traffic categories generated by different applications and protocols help the network operators and administrators, besides supplying a high QoS to the users [81].

Peng *et al.* [83] states that network traffic classification is a way to identify protocol and application type, besides classifying the traffic. It is the most vital step to manage modern networks and improve network services [84]. The increase of efforts is essential to improve the efficiency of classifiers based on applications and protocols when managing computer networks [83].

According to Valenti *et al.* [85], the identification of network applications and protocols is a process known as traffic classification. In the last two decades, this theme has gained space in research and several studies have proposed techniques and methods to classify traffic [82, 84, 86–88]. Among the more classical techniques, we can find payload-based techniques, ML-based techniques and port-based techniques.

By looking at the port number which an application or protocol uses to, port-based techniques enable us to classify those protocols and applications, based on the Internet Assigned Number Authority (IANA) [1]. There are many problems on port-based techniques, especially when dynamic port numbers are used on new applications to avoid detection [89]. This problem is widely known by researchers and has already been addressed on other researches [89]. A proposed alternative was to search within the packets for data sets that could be used as signature for a target application traffic [4, 86].

Payload-based techniques are also known as Deep Packet Inspection (DPI) [4]. These techniques are an alternative to the port-based techniques and are especially used in P2P applications that use random port numbers to stream applications over the network [4]. One of the characteristics of these techniques is the examination of the packets content,

regardless of the port number, to find attributes of network traffic protocols and applications [90, 91]. However, these techniques also have problems [42]. When faced with traffic from encrypted network applications, they are not efficient, having a high consumption of hardware resources to inspect the payload of each application and protocol [89]. Due to this disadvantage, methods that do not require DPI, such as "in the dark classification" have been developed [42].

In the dark classification sorts traffic by using behavioral and statistical patterns [42]. Gomez *et al.* [42] state that identifying the application without examining its packet is the major advantage of in the dark classification. The flow statistical behavior and transport layer information, such as packet length, packet inter-arrival time, Transmission Control Protocol (TCP)/Internet Protocol (IP) flags, and checksums are used for protocol identification. This approach can use a training set of sample traffic as a mechanism to identify and classify future traffic based on the application flow behavior [6]. Identification is done through traffic flow properties, such as packet size, entropy, and so on [6].

Different techniques are used to deduce the application protocol and correlate traffic properties, such as Machine Learning (ML) algorithms, sets of heuristics, or statistical measures [85]. For example, according to Liu [6], many researchers use ML to perform statistic-based classification. Statistical classification methods can be divided into two categories: parametric and non-parametric methods [92].

We propose and use the taxonomy of classification methods shown in Figure 2.1. We address statistical-based methods covering both parametric and non-parametric methods. The category of parametric methods includes Linear Support Vector Machines (SVM) [93], Euclidean Distance [94], Pearson correlation [95] and Jensen-Shannon Divergence [96]. The category of non-parametric methods includes non-linear SVM [93], Bhattacharyya Distance [97], Hellinger Distance [98], Kullback-Leibler (KL) Divergence [16], Wootters Distance [99], and Kolmogorov-Smirnov (KS) [100] and Chi-square [100] tests. Classifiers based in parametric methods have, for each class, a statistical probability distribution. As for the non parametric classifiers, they are used to estimate the statistical probability distribution, or in cases in which the density function is unknown [92].

Many surveys were written about traffic classification. Those surveys summarize the methods and have different focuses as presented on Table 2.1. The main difference between our research and other review works [2, 3, 14, 15, 40, 42, 80, 81, 84, 86, 88, 101–113] is that our proposal addresses solutions to solve traffic classification problems by using statistical methods, focusing on distances or divergences. Table 2.1 presents a comparison with other surveys published in the last ten years that were based on literature from previous decades. For this reason, we emphasized our work on the last ten years, since those papers already considered previous works. Details about ML and Deep Learning can be

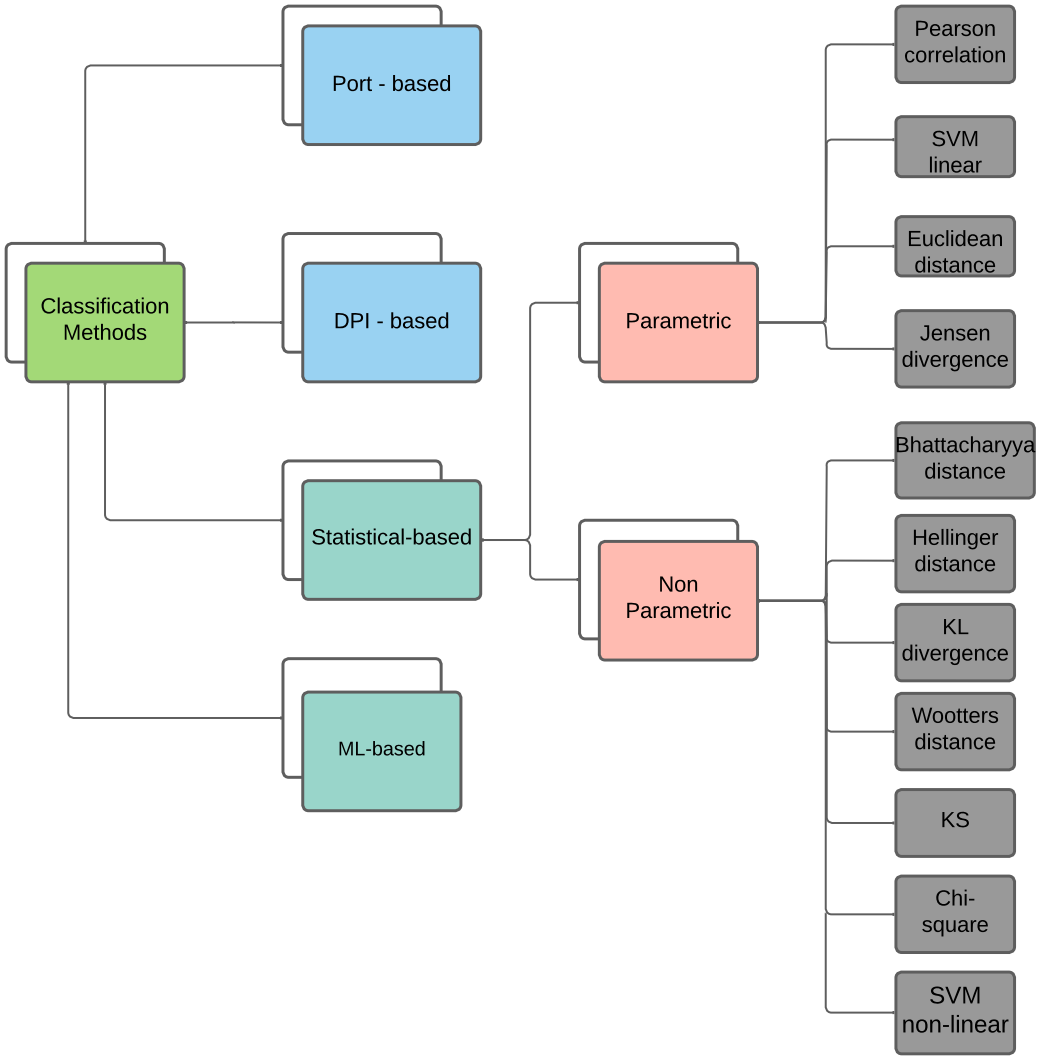


Figure 2.1: Overview of statistical methods for Internet traffic classification.

found at [80], [109], [3] and [84].

In this work, we review the classification of Internet traffic based on statistical methods, including classification methods applied ”in the dark”, observing the main objectives of each survey. It is important to emphasize that we also describe the statistical methods and distances proposed for classification in general, and specific traffic classification found in the literature. Specifying the research method is a crucial step in literature reviews [114]. Our study was guided by the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) methodology [115], [116].

According to [116], the PRISMA methodology offers us an evidence-based collection of items, which can be used as a basis for revision work. In addition, PRISMA provides us with a flowchart that allows us to visualize the search strategies and eligibility of the articles. The PRISMA flowchart describes the information cycle used in the different review phases. In order to present and detail our selection process, the flowchart was prepared

as shown in Figure 2.2. The flowchart has 3 phases: identification, screening and included. Through the flowchart, we mapped the number of articles identified, included and excluded, and the exclusions reasons.

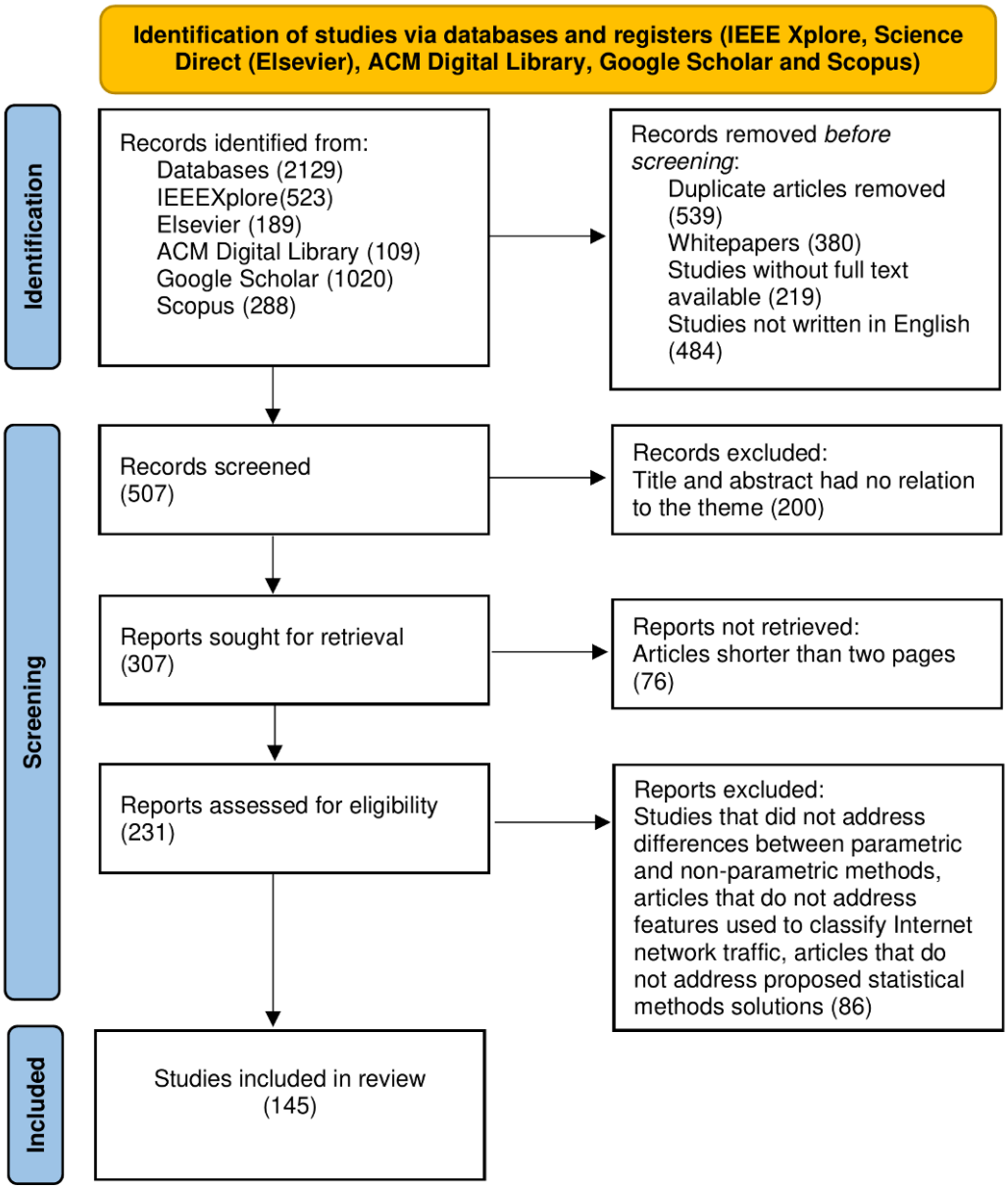


Figure 2.2: The PRISMA flowchart.

The reviewed articles in this paper were chosen from the almost 507 most relevant articles found on a search in IEEE Xplore, Elsevier, ACM Digital Library, Google Scholar and Scopus with the keywords: Internet Traffic Classification, Traffic Classification, Traffic Identification, Encrypted Network Traffic, Network Monitoring, and Statistical Distributions. We also searched for papers with the keywords: Statistical Methods, Statistical Distances, Statistical Analysis, Parametric, Non-Parametric, in the time period extending

Table 2.1: Comparison of recent surveys on classification.

Work	Year	Objective	Methods
Dunayts ev *et al.* [101]	2012	Presents the best practices and approaches developed to deal with P2P file sharing traffic, identifying those that may provide long-term benefits for both ISPs and users.	-
Pradhan [102]	2012	Presents a theoretical aspect of SVM, its concepts and applications overview.	SVM
Dainotti *et al.* [103]	2012	Presents reviews and discuss future directions in traffic classification, along with their applicability, reliability, and privacy.	Port/DPI/ML
Gomes *et al.* [42]	2013	Presents the studies on Peer-to-Peer traffic detection and port-based, DPI-based and ML-based classification approaches.	Port/DPI/ML
Se [104]	2013	Presents a survey on several ML techniques for IP traffic classification.	ML
Li *et al.* [105]	2013	Presents studies surveyed about advanced methodologies, such as machine learning datasets, and perspectives.	ML
Finsterbusch *et al.* [86]	2013	Presents a survey focused on performance analysis, technical requirements and accuracy in the DPI rating.	DPI
Dhote *et al. et al.* [14]	2016	Presents a research that addresses feature selection algorithms, focusing on: filter, wrapper, and embedded methods. It also provides an overview of some of the feature selection techniques presented in the literature.	Features - ML
Mehta and Shah [88]	2017	Presents a survey focusing on different types of network classification approaches.	Port/DPI/ML
Yan and Yuan [111]	2018	Examines emerging research on traffic classification techniques in Software-Defined Networks (SDN)	ML
Garrett *et al.* [107]	2018	Researches focused on finding tools and strategies to detect network traffic differentiation	Nearest Neighbor (NN)
Tavara [106]	2019	Presents a summary of parallel algorithmic approaches and parallel tools for SVM implementations focused on efficient approaches and large-scale problem solving.	SVM
Liu and Lang [40]	2019	Classifies and summarize Intrusion Detection Systems (IDSs) based on machine learning focused on solving network security issues.	ML
Rezaei and Liu [80]	2019	Presents a survey on the general structure to rank traffic based on Deep Learning, as well as the deep learning methods to rank traffic.	Deep Learning (DL)
Nalepa and Kawulok [112]	2019	Presents extensive research on existing methods to select SVM training data from large datasets.	Features - ML
Wang *et al.* [109]	2019	Presents a survey on the general Deep Learning-based mobile traffic classification framework, research approaches to traffic classification focused on mobile encrypted traffic classification in deep learning.	DL
Salman *et al.* [15]	2020	Presents a review of several data representation methods and the different goals of Internet traffic classification.	ML
Alqudah emphet al. [3]	2020	Presents a survey on different machine learning approaches for traffic analysis.	ML
Shen *et al.* [2]	2020	Presents a survey focusing on the systematic approach to optimize feature selection for an efficient classification of encrypted traffic.	Features - ML
Tahaei *et al.* [108]	2020	Provides a review of Internet of Things (IoT) problems and solutions for network traffic classification.	ML
Alam *et al.* [110]	2020	Provides a review focusing on issues related to one-class support vector classifiers.	SVM
Liu and Yu [84]	2021	Presents a survey about encrypted traffic identification focusing on ML.	ML
Zhao *et al.* [81]	2021	Provides a review of network traffic classification methods covering correlation-based, port-based, behavior-based, statistics-based, and payload-based classification.	correlation-based, statistics-based, behavior-based, payload-based, and port-based classification.
Our survey	2022	Presents a study on statistical methods, overviewing statistical distances and divergences for classification of Internet traffic.	Statistics-based, **distance-based**, and **divergence-based** classification.

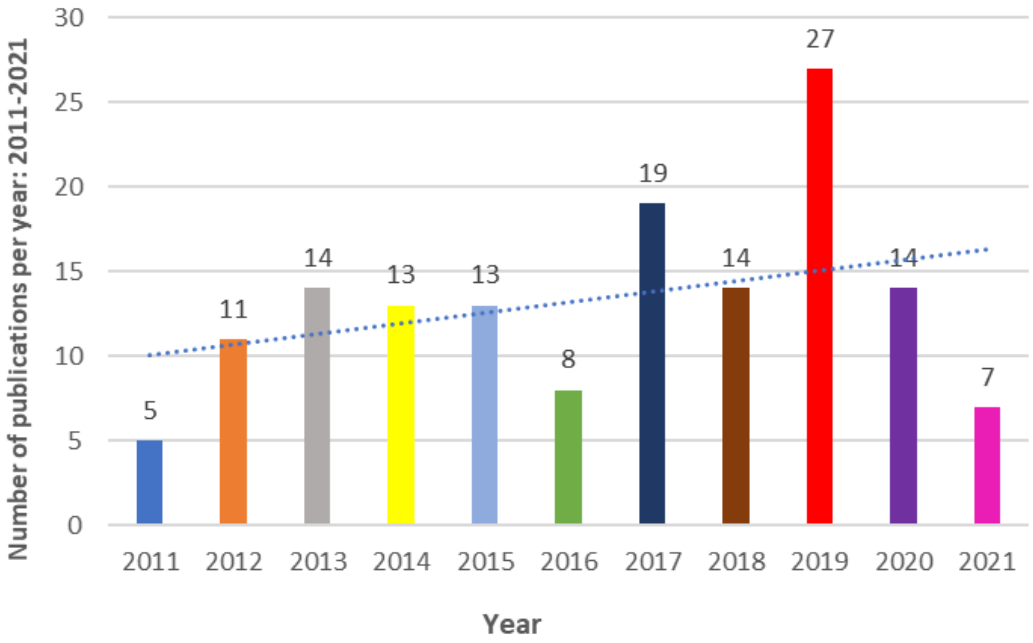


Figure 2.3: Selected studies on statistical methods for Internet traffic evaluation (2011 - 2021).

from 2011 to 2021. In total, 145 articles were reviewed for this work.

Our inclusion criteria were full papers published in journals, articles written in English, and articles that address features used to classify Internet network traffic. Our exclusion criteria were duplicate articles, whitepapers, articles shorter than two pages, studies that did not address differences between parametric and non-parametric methods, articles that do not address features used to classify Internet network traffic, articles that do not address proposed statistical methods solutions, articles written in languages other than English, and studies without full text available. After applying the keywords, articles that were not related to the topic in question were excluded by reading the abstract and title. We selected for full reading the articles that could be included after the exclusion and inclusion criteria.

In the initial phase, 2129 articles were identified; of which 1622 were excluded because they were duplicates, whitepapers, studies without full text available, and articles written in languages other than English; 507 were pre-selected. In the screening phase, the abstracts and titles of the articles were read and those unrelated to the topic were excluded, totaling 200 excluded and 307 eligible. Out of the 307 articles, 76 were eventually removed as they were too short, with only two pages. Finally, 231 articles were fully read, of which 86 were excluded for not addressing statistical methods solutions proposed or differences between parametric and non-parametric methods, totaling 145 that met our eligibility criteria and were included in our study. In order to present the results of our selection of articles, following our eligibility criteria, a statistical analytical visualization chart was generated, as shown in Figure 2.3 with the number of articles per year.

This survey was structured as: review of the classification processes on Section II. Overview of SVM and statistical methods focusing on distances and divergences on Section III. Several methods to classify Internet traffic by using statistical methods on Section IV. Discussion and list of open issues on Section V. Section VI concludes the survey.

2.2 Process of Traffic Classification: Overview

2.2.1 Classification Procedures

An overview of the traffic classification process was provided in this section as it follows: Internet traffic categorization, data-set, features, classification approach, and validation. Collecting data from a network is a critical point and serves as input to form a pool of network traffic. Extracting and selecting features is an vital process as it can impact the efficiency and effectiveness of classification. The approach chosen for traffic identification is essential to the classification success, as well as ranking performance evaluation criteria [81]. The Figure 2.4 shows the procedures for classification. As it follows, two topics will be approached: 1) Internet traffic applications, and 2) Dataset.

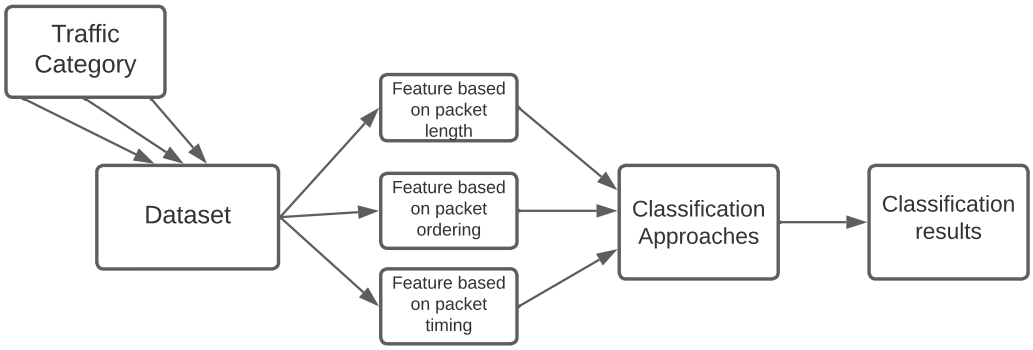


Figure 2.4: Classification procedure.

2.2.1.1 Internet Traffic Applications

Internet traffic describes the quantity of information or data presented throughout the Web and on different applications, it can be considered a data flow on the Internet [80]. According to [117] Internet traffic is categorized and described according to Table 2.2. Internet traffic is grouped forming a Dataset.

Internet traffic is divided into ten categories: Administration, Communications, Gaming, Filesharing, Marketplaces, Social Networking, Real-Time Entertainment, Storage, Tunneling, Web Browsing. Each category has a description, that characterizes the associated

Table 2.2: Description of traffic categories according to Global Internet Phenomena Report: 1H 2014 [117].

Category of Traffic	Explanation	Examples
Administration	Services and applications used for network administration.	SNMP, DNS, NTP, ICMP.
Communications	Protocols, applications and services that allow chat, video and voice communications; information sharing (photos, status, etc) between users.	FaceTime, WhatsApp, Skype, iMessage.
Filesharing	File distribution models or models that use Peer-to-Peer file sharing.	Newsgroups, , BitTorrent Ares, eDonkey, Gnutella.
Gaming	Application console, game updates and PC gaming download traffic of consoles.	PC games, Playstation 2, Xbox Live, Playstation 3, Nintendo Wii.
Marketplaces	Marketplace apps where purchases of media, apps, books, software, movie, music, and updates are performed by subscribers.	Windows Update, Apple iTunes, Google Android Marketplace.
Real-Time Entertainment	Protocols and applications that provide "on-demand" entertainment that is consumed (viewed or heard) as it arrives.	Buffered or streamed video and audio (RTP, RTSP, MPEG, RTMP, Flash), specific streaming sites, peercasting (Octoshape, PPStream), and services (Spotify,YouTube, Hulu, Netflix).
Social Web	Services and websites that allow interaction such as chats and other types of communication, as well as sharing information between customers and users.	Instagram, Twitter, Facebook, Linkedin.
Storage	It allows transferring through the File Transfer Protocol a massive volume of data, in addition to allowing file hosting, backup and download services.	Dropbox, FTP, zShare, Mozy, Rapidshare, Carbonite.
Tunneling	Services and protocols that mask application identity or allow remote access to network resources.	Remote Desktop, VNC, PC Anywhere, Secure Sockets Layer (SSL), SSH.
Web Browsing	Specific websites and web protocols.	WAP browsing, HTTP.

traffic. The Administration category can be described by services and applications used to administrate the network, such as SNMP, and ICMP protocols. Gaming includes traffic by PC gaming, console, download traffic of consoles, and game updates, like Xbox Live and Playstation traffic. File-sharing includes applications that use distribution protocol models or Peer-to-Peer, such as Gnutella, eDonkey, Bittorrent, Newsgroups, Ares. Web Browsing includes specific websites, and Web protocols, like WAP browsing, and HTTP.

2.2.1.2 Dataset

Dataset, in classification, has huge importance on evaluating and comparing the performance of different methods. A dataset must contain many diverse samples of each class. A model can fit itself to a specific dataset, doing so, the worry around the probability of the dataset having a deterministic behavior appears. That can happen when a model adjusts itself too much to a specific type of traffic or to a dataset, either because of a lack of interaction to group of users or even for interacting to a small group of users [80]. Usually, for being diversified, traffic is observed on ISPs core, which means that the farther away from the destiny the captured traffic set is, the smaller is the probability of having a deterministic behavior [80].

According to [118], and [3], a dataset is collected and used as an input for training and classification purposes on an ML classifier. On statistical-based methods, statistical resources are allowed to be extracted from flows. These resources are characteristics or properties of flows calculated over many packets [118]. Normally, different features and datasets are used classification-wise.

As stated by [15] a pre-processing phase happens after data collection to extract features that are going to be included into the model. For traffic classification, it is required to evaluate network flow with their main characteristics (packet inter-arrival time and size) with their various statistical values (standard deviation, quartile, min, and max). A set of packets that have the same connection parameters is defined as a flow. Those parameters include port numbers and transport protocol, destination and source IP addresses.

As said in [119], a different way of representing Internet traffic is through time series. They are network flows represented by generating time series of communicated packets/bytes. For each flow three-time series are generated: (1) for bytes channeled through input packets, (2) for bytes channeled through output packets, and (3) for bytes channeled through input and output packets. A short description about feature selection will be present as follows.

2.2.2 Feature Selection

Features are considered in the process of investigating methods and approaches to characterize and classify traffic. Feature selection is a important step in Internet traffic classification. The author [120] sets it as the process of selecting the smallest set of features needed to reach a precise classification. The classification of different application categories occurs when there are some discrepancies in traffic behavior based on the selected features. However, [14] claims that researchers have chosen one or some features from a set of characteristics to classify different traffic flows, basing it only on the qualitative analysis of different features. For analysis purposes, according to [2], it is needed to clas-

sify encrypted traffic into numerous flows based on five tuples: User Datagram Protocol (UDP)/TCP, source/destination IP addresses and port numbers. Hereafter the following features will be approached: 1) Packet-Length Based Features, 2) Packet-Ordering Based Features, and 3) Packet-Timing Based Features.

2.2.2.1 Packet-Length Based Features

As packet length is a feature related to network packets, according to [2], and [121], its information becomes a commonly used type of resource and it has demonstrated its effectiveness in analyzing traffic that has been encrypted. On packet-length based features, the packet length, the cumulative length sequence and the statistics that can be drawn out of the flow, such as minimum, maximum, average, median variance, standard deviation, relative frequency, kurtosis, skew, packet size and variance can be statistical values of packet length.

When obtaining the packet length, in each flow, the first length sequence of the X packets can be used as a key resource. Those X packets can vary a lot length-wise from one website to another, because of their different content and protocol parameters, like those in handshake process, more specifically in the the Transport Layer Security (TLS)/SSL. We can use lengths of distinctive packets to distinguish different traffic types.

In a flow, the packet lengths are distributed in intervals that depend on the transport layer and on the MTU (Maximum Transmission Unit). To obtain statistical characteristics of packet length in a flow, the packet length can be aggregated to a fixed number of buckets or bags. To obtain the cumulative length sequence, considering the flow direction, the length of up-link packets can be defined as negative, and as positive for down-link packets. The length of the packets sent are then accumulated to obtain a sequence of the first X cumulative packet lengths. Considering bidirectional flows, we can be define as positive when the packet length is up or down-link. A sequence of cumulative length of the first X packets on a flow seems to be a differentiating feature.

2.2.2.2 Packet-Ordering Based Features

In some cases, the lengths of packets are alike or even the same between different encrypted traffic flows. That makes so alternatives based only on packet length seen less efficient because of the information used. For that, a counter or techniques based on packet counting can be useful [2].

Some packet counts can be considered, such as counting the quantity of up and down-links for each X packets. We can also count the amount of packets before each up-link. Besides that, we can also extract a resource that indicates the number of down-links between two up-link packets.

According to the literature, burst counting can also be useful. An up-link packet burst can be used as example, since down-link packets are exposed to network delays. To do so, the quantity, maximum and average of bursts were considered for each flow.

2.2.2.3 Packet-Timing Based Features

Several information about timestamps of packets can be used to characterize and classify traffic [2], [14]. Inter-Packet Delay is one of the examples. When packets are sent through the network, they receive a timestamp of date and time. The difference between timestamps is defined by the Inter-Packet Delay.

To determine the period of time a transmission is concentrated, the quantity of packets in a time interval is calculated for every series of packets. Timing characteristics generally have limitations, as we most often consider time distributions to be equal, when in reality they are not. The timestamps of packets may experience network fluctuations. This feature can be combined with control packets such as ACKs, CTSs reference points. Table 2.3 presents a summary of packet-based features.

Table 2.3: Summary of packet-based features.

Type	Features
Packet-Length Based	Packet length in bytes; packet size; packets number; Statistical values of packet size; Sequences of cumulative length; Relative frequency; Statistical features (mean, maximum, minimum, variance, standard deviation, median absolute deviation, percentiles, kurtosis, skew).
Packet-Ordering Based	Count the number of bursts; In each flow, mean and maximum burst length.
Packet-Timing Based	Packet inter-arrival time; Inter-packet delays.

Even though some features were chosen to classify Internet traffic differently, they do not have the same level of importance. To better understand, each selected feature can receive a weight value that represents its importance. In order to select only important sets of resources, the author in [14] discusses three methods, Wrapper method, Filter method, and Embedded Method, which are briefly described below:

- Wrapper method - Makes use of machine algorithms to rate the performance of different subsets to aid learning. The results are not specific to the ML algorithms used, for this process Genetic Algorithm (GA), Sequential Forward Selection, Simulated Annealing, Sequential Backward Selection, Randomized Hill Climbing are used.

- Filter Method - Makes an independent evaluation based on data characteristics and depends on specific metrics to, before learning begins, rate and select the best subset. For that Correlation based Feature Selection (CFS) algorithm is normally used

with Fast Correlation based Feature Selection (FCFS), and Markov Blanket Filter method.

- Embedded Method - As part of the learning procedure, performs variable selection and it is usually specific to some learning machines. For this process decision tree, Naive Bayes, random forest, Support Vector Machine (SVM), and based methods are normally used in regularization techniques, etc.

2.2.3 Classification Approaches and Validation

In the literature, we find four kinds of approaches to Internet traffic classification: port-based approaches, payload-based approaches, ML-based and statistical approaches. We provide in Table 2.4 a comparison among these kinds of approaches, which we briefly describe in the following.

Table 2.4: Comparison of the main approaches for traffic classification.

Kind of Approaches	Short Explanation	Limitations
Port-based	Associates port numbers to match applications.	Does not solve random or unknown port numbers.
Payload-based	Searches for protocol/application signatures in the form of string(s) in packet payload.	Cannot scan encrypted packets, encrypted payload and encrypted connections.
Statistics-based	Uses statistical values from the network or transport layers.	Generally does not specify application/client type.
ML-based	Automated method that foresees and makes a decision based on data analysis.	Set of pre-classified (also called pre-labeled).

2.2.3.1 Port-Based Approaches

The oldest traffic classification method is the port-based approach. According to [1], this method uses the association of well-known TCP/UDP port numbers assigned by IANA with ports in the TCP/UDP header. It uses port numbers related to an application where the application is related to a specific port number [2], some examples are SSH traffic that relates to port 22, and SMTP to 25. Most applications use port numbers already "known" so other hosts can start communication.

During handshake, an identifier is placed in the communication channel, right in the middle of the network, awaiting for SYN packets. SYN packets have the destination port number and are used during the handshake on TCP. The application is recognized by the port number contained in the SYN packet. It becomes all possible because TCP is connection orientated. Traffic identification through port numbers is also used on UDP, even though this protocol does not have control packet in its connection.

Implementing this method is quite simple and quick, once it does not involve calculations and requires only the number of ports to identify the application. Although its easy implementation, this approach has limitations that have a huge negative impact on traffic classification. Protocols that use tunnels, random ports, and Network Address Port Translation (NAPT) cannot be identified by this approach [86]. One possibility to easily escape detection by this method is to use port 80, which is generally open for HTTP traffic.

Some other protocols that cannot be identified by port-based approaches are the telephony through Internet that uses encapsulated Session Initiation Protocol (SIP) on Real-Time Transport Protocol (RTP), which sometimes use random port numbers, and P2P protocols that use random ports or ports associated to other protocols aiming to mask the traffic [86].

2.2.3.2 Payload-Based Approaches

This kind of approaches recognizes applications by analyzing payload or packets. Aiming to find pre-defined byte sequences from the applications, payload is analyzed bit by bit. After those sequences, called signatures, are found, they are stored and compared to application packets for classification [2]. The great advantages of these methods are their capacity to generate low rates of false negatives and a highly accurate traffic classification.

The biggest limitations of these methods are: The development and maintenance of a database with application signatures. The high consumption of computational resources for the development requiring a longer processing time and storage space. It is an inefficient method to identify and classify traffic and packet payloads that are encrypted, unavailable payloads or on recognizing applications that have not been mapped yet. Besides that, it involves legitimacy and privacy issues of packets and traffic [122], [65].

Approaches based on statistical characteristics for traffic classification have been developed aiming to overcome limitations presented by traditional approaches, and they have caught the attention of researches. To identify and classify traffic, neural network and machine learning algorithms have been used.

2.2.3.3 ML-Based Approaches

Machine Learning is known by supplying computers with the capacity to learn through programming. It has been used to prepare machines to work with data in a more efficient way. Machine learning is divided into unsupervised and supervised. On the unsupervised learning, information is extracted through non labeled data. On the other hand, on supervised learning, the information depends necessarily on data lettering. Machine Learning uses data patterns to label things [3, 39, 40].

ML has the capacity to work and learn from big data volume by using specific algorithms. Tasks as prevision, regression and classification of massive quantities of data can be solved through it. Machine Learning also has the capacity to deal with long and wide data. Long data means that number of subjects exceeded the number of input variables. Wide data corresponds to the number of input variables exceeding the numbers of subjects [40, 123, 124].

As appointed by [39], ML has a different and specific algorithm to solve problems involving data. Choosing the best algorithm to be used depends on which modal will better suit the problem, what the problem is and the quantity of variables involved in it.

Besides Internet traffic classification, ML has also been used in network operations and management, aiming to optimize the resources and improve the system performance. In addition, ML can be applied to many different areas such as marketing, games, digital images, intruders and malware detection, information security and data privacy.

2.2.3.4 Statistical Approaches

Statistical-based classification uses statistics from the network and transport layers. By using parameters undependable of payload and payload analysis, statistical based classification methods go around payload, encrypted payloads and user privacy problems. They use statistical properties unique of protocols, flows and applications, which helps to differentiate the applications [2, 14].

Some examples of valid parameters of statistical-based network classification: packet inter-arrival time, flow duration, packet size, among others [2, 14]. Besides those parameters, statistical characteristics of packet tracking are captured and used, such as Border Gateway Protocol (BGP) updates and the unexpected rise of packet rate, which can also be an indicative of P2P applications in the network.

Commonly, Machine Learning uses statistical-based strategies to calculate resource parameters that will be used as data input in the supervised method classification, like SVM [2]. As stated by [41] techniques that are implemented based on statistical classification, are capable of perceiving flow behaviors expected through observations. Statistical methods combined with methods grounded on rules might offer scalability, adaptability, flexibility and robustness. Furthermore, to differentiate traffic that has any flaws from regular traffic, statistical measurements can be used. However, the manual selection of statistical resources can compromise the requirements of traffic classification, generating a lower accuracy.

2.2.4 Validation

The validating process consists on testing the obtained results from the classification, aiming to acquired its performance. In this sense, obtained classification results are compared to previously-known hand-based real data classification results, usually known as ground truth, which allows to compute true positive, false positive, true negative and false negative rates. Another challenge during validation is to collect original data in real time to obtain the ground truth [125].

Many performance measures are used to evaluate if a classification method could achieve the expected performance. Table 2.5 represents an overview of the metrics used to evaluate traffic classifiers. Metrics widely used are: F-measure [22], Precision, Recall, Specificity, Area Under Curve (AUC), Completeness [120], and F-1 Score [126].

2.3 Statistical Methods

In this section, we address the concept and properties of the statistical distances and divergence, as well as the SVM method based on statistics and widely used in traffic classification. Table 2.8 presents an overview of distances and divergences for quantitative (non-negative) data. We group the methods according to parametric and non-parametric approaches.

2.3.1 Overview of Parametric and Non-Parametric Models

On parametric models, datasets can be constructed by a probability distribution that has a number or a fixed set of parameters, which only the applied to variables. It is considered to be a parametric model some statistical and learning models that use a quantity of fixed parameters. For parametric ML, the quantity of parameters if fixed does not matter the amount of training data. Some examples of parametric models are Linear SVM, Pearson correlation, denominated correlation information and Euclidean Distance [59, 92, 127, 128].

Non-parametric modals represent data without a defined number of parameters, and when modeling this data, they do not make presumptions about the probability distribution. Models implemented with this approach do not accept a specific mapping function between input and output data as true. This kind of models assumes that parameters are not only adjustable, but can also be altered. Parametric model also assumes that the larger the quantity of training data is, the larger will be the number of parameters. The result of this is that the non parametric model can take longer to perform the training [92,128,129]. Table 2.6 presents a comparison between parametric and non-parametric models. Bhattacharyya Distance, Hellinger Distance, KL Divergence, Wootters Distance, KS and Chi-square tests, and non-linear SVM are examples of non-parametric models. Hereafter the

Table 2.5: Summary of the metrics often used to evaluate the traffic classifiers, where TP means True Positive, TN means True Negative, FP means False Positive, FN means False Negative, TPR means True Positive Rate, TNR means True Negative Rate.

Metrics	Description	Exemplification
Accuracy (A) [22]	It is the ratio between cases classified as truly positive and negative and the sum of all positive and negative cases predicted in the classification.	$\frac{TN+TP}{FP+TN+FN+TP}$
Precision (P) [22]	It correctly evaluates how many cases are identified as positive.	$\frac{TP}{FP+TP}$
Recall (R) [22]	It is known as true positive rate or hit rate, it presents the rate of positive cases that were correctly identified by the classifier in the dataset.	$\frac{TP}{FN+TP} (= TPR)$
Sensitivity [120]	It is also known as Recall metric.	$\frac{TP}{FN+TP}$
Specificity [126]	It calculates the number of correctly classified positive cases for the total positive cases found.	$TNR = \frac{TN}{TN+FP}$ or $\frac{TN}{TN+FP} = 1 - FPR$
Completeness [120]	For the total number of positive cases, the proportion of correctly or incorrectly classified positive cases is measured.	$\frac{FP+TP}{FN+TP}$
F-Measure [22]	It measures the effectiveness of debug testing, it is considered harmonic calculation between Precision and Recall.	$\frac{2*\text{Recall * Precision}}{\text{Recall + Precision}}$
F1-Score [126]	It is the harmonic mean between Precision and Sensitivity.	$\frac{2TP}{2TP+FP+FN}$
Area Under the Curve AUC [120]	It is known as Receiver Operating Characteristics (ROC)	$\frac{1+TPR-FPR}{2}$ or $\frac{\text{Sensitivity + Specificity}}{2}$
False Positive Rate (FPR) [126]	It is the calculation of the rate of negatives incorrectly classified as positives	$\frac{FP}{N} = \frac{FP}{FP+TN}$
False Negative Rate (FNR) [126]	It is the calculation of the rate of positives incorrectly classified as negatives	$\frac{FN}{P} = \frac{FN}{FN+TP}$
Geometric mean (G-mean) [126]	It is the calculation of the correlation between the rate of positives and the classified results	$\frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}}$

following subtopics will be approached: 1) Statistical Distances and 2) Statistical Divergences.

This section focus on Statistical Distances and Divergences. A brief description about

Table 2.6: Side-by-side comparison of parametric and non-parametric classifiers.

Parametric classifier	Non-Parametric classifier
The model is built using a fixed number of parameters.	The model is built using a flexible number of parameters.
It can only be applied to variables.	It can be applied to Attributes and Variables.
It makes strong data assumptions.	It normally does not make data assumptions.
It needs less data.	It needs much more data
It assumes a normal distribution.	No distribution is assumed.
Data manipulation – Ratio or interval data.	Original data is manipulated.
Outliers can seriously effect the results.	Outliers cannot seriously affect the results.
Performance peaks when the spread of each group is different.	Performance peaks when the spread of each group is the same.
It has more statistical power.	It has less statistical power.
It is faster, computationally speaking.	It is not so fast when compared to parametric models

these kinds of methods follows. Details about other methods herein mentioned that do not fall within those kind of statistical methods may be found elsewhere, namely details about Correlation Information (Pearson correlation) can be found in [130], [131], details about Kolmogorov-Smirnov and Chi-square tests can be found in [130], [132], and details about Shannon entropy can be found in [133], [134].

2.3.1.1 Statistical Distances

The concept of distance between objects or individuals allows us to interpret, geometrically-wise, many classical techniques of multivariate analysis, equivalent to representing these objects as points in a metric space. In classification [135] of network traffic, the main objective is to find statistical differences between flows or even a pattern in traffic characteristics through statistical properties. It is possible to interpret this way because the observed variables are considered of a more general category, and not only as quantitative variables or own variables. As it is, it makes sense to calculate the proximity between objects or individuals [135, 136].

As stated by [135] the distance calculation is vital to many statistical inferences being them theoretical or applied. Besides that, it has become essential to solve data processing problems, such as classification, estimation, detection, regression, selection models, diagnosis, identification, recognition, indexation and compression. Combining its properties to statistical distance concepts, we have an essential instrument for science and data analysis [137].

Through the distance computation, it is possible to create hypotheses tests, study the

Table 2.7: General properties of distances and divergences and qualification of a distance according to its properties, where δ_{ij} represents the distance between pairs.

Qualification of a distance according to its properties	Distance property
Dissimilarity: 1, 2, 3	1 – $\delta_{ij} \geq 0$
Metric distance: 1, 2, 3, 4, 5	2 – $\delta_{ij} = 0$
Ultrametric distance: 1, 2, 3, 6	3 – $\delta_{ij} = \delta_{ij}$
Euclidean distance: 1, 2, 3, 4, 8	4 – $\delta_{ij} \leq \delta_{ik} + \delta_{ik}$
Additive distance: 1, 2, 3, 7	5 – $\delta_{ij} = 0 \Leftrightarrow i = j$
Divergence: 1, 2, 10	6 – $\delta_{ij} \leq max\delta_{ik}, \delta_{ik}$ (ultrametric inequality)
	7 – $\delta_{ij} + \delta_{kl} \leq max\delta_{ik} + \delta_{jl}, \delta_{il} + \delta_{jk}$ (additive inequality)
	8 - δ_{ij} is euclidean
	9 - δ_{ij} is riemannian
	10 –δ_{ij} is a divergence

estimators properties, compare classes, objects and individuals. Furthermore, the distance offers the researcher an assistance to interpret the data, because it is a very intuitive concept, allowing an easy comprehension and a harmonious representation [137, 138].

In general, we consider two classes of statistical distances between individuals and populations. The individuals of each population are characterized by a random vector $X = (X_1, ..., X_P)$, which follows a probability distribution $f(x_i, ..., x_p; \theta)$. The distance between two individuals i, j, characterized by the points x_i, x_j, of R^p, is a non-negative symmetric measure, $\delta\ (x_i, x_j)$, which will depend on θ, where θ represents the parameters and R^p is the quantity of dimensions that the X variable may have. Therefore X has n observations and p variables.

Moreover, the distance between two populations will be measured by the divergence $\delta(\theta_1, \theta_2)$ between the parameters that characterize them. It may also be convenient to enter the distance $\delta(x_i, \theta)$ between an individual i and the θ parameters. Non-parametric distances can be defines by it functional divergence and the density functions. In some cases they are related to entropy measurements.

A δ distance over an Ω set is an application of Ω x Ω over R so that each pair (i, j) corresponds to a real number $\delta(i, j) = \delta ij$ fulfilling some of the following properties, according to the Table 2.7.
A distance must fulfill at least properties 1, 2, 3, presented in Table 2.7. When it fills these properties, it is called dissimilarity. In general, δ only meets approximately some of the stated properties. It is then a matter of representing (Ω, δ) through a model (V, d), approximating δ to d, where δ meets sufficient properties that are mandatory.

According to the representation technique, such as main component analysis, main coordinate analysis, proximity, correspondence analysis, cluster analysis, the distance d can

be Euclidean, ultrametric, additive, non-Euclidean, or Riemannian, among others.

2.3.1.2 Statistical Divergences

Non-parametric measures of divergence between probability distributions are defined as functional expressions often related to information theory, which measures the degree of discrepancy between any two distributions, not necessarily belonging to the same parametric family. Divergences have applications in statistical inference and in stochastic processes.

Let $p = (p_i, ..., p_n)$, $q = (q_i, ..., q_n)$ be two multinomial distributions. The divergence between q and p can be measured as the discrepancy between the quotient $x_i = q_i/p_i$ and 1. Based on the meaning of $Ho = (p) - entropy$, $\phi-$ Csiszar divergence is defined between p and q, where ϕ is a strictly convex function in which $\phi(1) = 0.H\phi$, and by Jensen inequality we have:

$$C_\phi[p, q] = \sum p_i \phi(x_i) \geq \phi(\sum p_i x_i) = \phi(1) = 0. \quad \textbf{(2.1)}$$

The equation 2.1 reaches the value 0 if and only if $p = q$. It can be taken as a measure of dissimilarity between p and q, but in general it is not a distance, as it is not always symmetrical, or if it is, it may not meet the triangular inequality. Shannon entropy and the $\phi-$ Csiszar divergence form the information measure known as the Kullback-Leibler (KL) [16].

2.3.2 Parametric Distances and Divergences

2.3.2.1 Euclidean Distance

The most familiar distance between two individuals i, j is the Euclidean Distance described by the equation [16]:

$$D_E[i, j] = \sqrt{\sum_{k=1}^{p} (x_{ik} - y_{jk})^2}. \quad \textbf{(2.2)}$$

Proposed by the Greek mathematician Euclid, it is based on calculating the distance between two points within the Euclidean space. where $D_E[i, j]$ represents the distance function, p defines the quantity of samples, k defines the initial value of the sample, x_{ik} represents the first point and y_{jk} represents the second point [17].

2.3.2.2 Jensen-Shannon Divergence

Jensen-Shannon Divergence (JSD) is the calculation of the difference between two series of probability distributions [139]. It is known for being the limited symmetrization of

KL [19].

JSD is a function that allows us to quantify the difference of two, maybe more, probability distributions [96]. JSD also has the additional advantage of not requiring absolute continuity of the distributions to compare them. Thereby, JSD can be used to compare the distribution of different packet sequences in a network flow, associating an appearing frequency to each flow with probability distribution.
For two discrete probability distributions $P = (p_1, p_2, ..., p_n)$ and $Q = (q_1, q_2, ..., q_n)$ with $pi \geq 0$, $qi \geq 0$, JSD Divergence is represented by [139]:

$$JSD\,(P,Q) = \quad \frac{1}{2}\left\{\sum_{i=1}^{N} pi \log\left(\frac{2pi}{pi+qi}\right) + \sum_{i=1}^{N} qi \log\left(\frac{2qi}{pi+qi}\right)\right\}. \quad (2.3)$$

JSD function equals 0, if and only if $(pi = qi)$. In this case, it means that they are the same distribution, in other words, the same application. It is a delimited and symmetric metric $(0 \leq JSD \leq log(2))$ for orthogonal distributions $(pi.qi = 0)$. As traffic classification was intended through the values of the distances between the application distributions, JSD determines the divergence between two probability distributions P and Q.

2.3.3 Non-Parametric Distances and Divergences

2.3.3.1 Bhattacharyya Distance

Bhattacharyya Distance, also known as divergence, was proposed by a statistician called Anil Kumar Bhattacharyya (1943 and 1946) working with Kailath (1967) [140]. This distance measures the dissimilarity between two probability distributions. It is very related to Bhattacharyya coefficient, that is the calculation of the quantity of overlap of two statistical population samples [141, 142]. In its first version, Bhattacharyya did not present the calculation, he used a logarithm scale.

Bhattacharyya Distance is independent of the distribution function and it can be applied to any data set or sample. This characteristic makes the distance appealing to be used in models in which the distribution is undetermined [141].

Bhattacharyya coefficient can be used in classification as a measure of the separability between classes [143], and to determine the relative proximity between samples that are being taken under consideration.

When two probability distributions have similar averages, Bhattacharyya Distance rises depending on the difference between standard deviations, in other words, the bigger the difference between standard deviations, the bigger the probability distribution. Bhat-

tacharyya statistical distribution is given by equation 2.4 [18]:

$$Bc_D\left(P,Q\right) = \quad -\log\left(\sum_{i=1}^{N}\sqrt{pi \quad \times \quad qi}\right), \quad (2.4)$$

where N is the quantity of partitions and pi and qi is the quantity of members from the sample in the I-th partition.

2.3.3.2 Hellinger Distance

Hellinger Distance was proposed by the German mathematician Ernst David Hellinger in 1909. It is a statistical divergence used to calculate the dissimilarity between two probability distributions. Hellinger Distance (HD) is related to Bhattacharyya Distance and it is part of the f-divergences family [19].

Studies presented in [144], [145] showed that Hellinger Distance can be used in classification. On the current scenario, this distance has been very used in machine learning, even as an alternative to methods such as entropy, aiming to detect failures in the classifiers [146] and breakpoints on the performance of those classifiers [147]. Furthermore, according to the literature, Hellinger Distance has been used in many parametric models being very successful on solving problems of statistical estimation [144], [145]. The calculation function is obtained from two probability distributions p and q as follows [145]:

$$H_D\left(P,Q\right) = \sqrt{\frac{1}{2}\sum_{i=1}^{N}\left(\sqrt{pi} - \quad \sqrt{qi}\right)^2}. \quad (2.5)$$

Hellinger Distance is non-negative and symmetric, and $H_D(P,Q)$ is in $\left[0, \quad \sqrt{2}\right]$. Note that the higher Hellinger Distance is, the better the differentiation between probabilities will be.

2.3.3.3 Kullback-Leibler Divergence

Kullback-Leibler (KL) Divergence, well known as relative entropy, was defined by the mathematicians Solomon Kullback and Richard A. Leibler in 1951. It represents the calculation between two probability distributions [148], [149], [150], [20]. Through statistical testing, those mathematicians started from the principle that two probability distributions are different, since there is a possibility of differentiation between them. KL measures the information gain and has been used in statistics, specially in Bayesian statistics.

KL is considered a special class of divergence, being an asymmetric measurement of difference or not dissimilarity. Therefore KL allows us to deduce both the difference and the

Table 2.8: Summary of distances and divergences for quantitative (non-negative) data.

Distance / Divergence	Explanation
Bhattacharyya Distance [18]	$Bc_D(P,Q) = -\log\left(\sum_{i=1}^{N}\sqrt{pi \times qi}\right)$
Euclidean Distance [16]	$D_E[i,j] = \sqrt{\sum_{k=1}^{p}(x_{ik} - y_{jk})^2}$
Hellinger Distance [18]	$H_D(P,Q) = \sqrt{\frac{1}{2}\sum_{i=1}^{N}\left(\sqrt{pi} - \sqrt{qi}\right)^2}$
Jensen-Shannon Divergence [139]	$JSD(P,Q) = \frac{1}{2}\left\{\sum_{i=1}^{N} pi \log\left(\frac{2pi}{pi+qi}\right) + \sum_{i=1}^{N} qi \log\left(\frac{2qi}{pi+qi}\right)\right\}$
Kullback–Leibler Divergence [148], [149], [150], [20]	$D_{kL}[p\|\|q] = \sum_{i=1}^{N} pi \log\left(\frac{qi}{pi}\right)$
Wootters Distance [21]	$(P,Q) = \arccos\left(\sum_{i=1}^{N}\sqrt{pi \times qi}\right)$

not dissimilarity between two distributions [20]. In KL, p_i e q_i are considered probability distributions, where the function is represented by $D_{kl}[p||q]$.

$$D_{kl}[p||q] = \sum pi \log\left(\frac{1}{pi}\right) - \sum pi \log\left(\frac{1}{qi}\right). \quad (2.6)$$

It can also be given by the equation:

$$D_{kl}[p||q] = \sum pi \log\left(\frac{pi}{qi}\right). \quad (2.7)$$

On problems of data processing or classification, the result of the function $D_{kl}[p||q]$ is the calculation of the expected p value, essential on samples based on q. Normally, the data is represented by p that assumes the real or current distribution of class, flow, application or model that are represented by the q variable [151, 152].

$$D_{kl}[p||q] = \sum_{i=1}^{N} p(x_i) log \frac{p(x_i)}{q(x_i)}, \quad (2.8)$$

where N defines the quantity of samples. See that the symmetric version of KL divergence is the Jensen-Shannon Divergence [19, 153].

2.3.3.4 Wootters Distance

Wootters Distance was proposed by the American physicist William Wootters in 1981, aiming to calculate the probability differences under the values of typical fluctuations. The main idea of this distance is to properly consider the statistical fluctuations inherent to any finite sample. It is purely and simply statistical and the concept can be used in any probabilistic area [21].

Considering two probability distributions p and q, the minimal distance between two points will be equivalent to the angle presented by them, represented by the equation 2.9 [18],

[21]. Wootters can also define the not dissimilarity between two samples [154]. Given two probability distributions $P_i = \{p_j^{(i)} , j = 1, ..., N\}$ with $i = 1, 2$.

$$Wo_D(P, Q) = \arccos\left(\sum_{i=1}^{N} \sqrt{pi \times qi}\right). \quad (2.9)$$

Note that $arccos()$ decreases in [0, 1], and that distances were used to discriminate traffic. Table 2.8 presents a summary of distances and divergences for quantitative (non-negative) data.

2.3.4 Support Vector Machines

Support Vector Machine (SVM) was developed by Vapnik, Guyon, and Hastie [155], based on the Statistical Learning Theory and aims to solve pattern classification problems. Statistical Learning Theory gives us mathematical conditions to choose an efficient classifier to train and test a specific set of data. SVM is a supervised method focused on classification and regression. To classify, initially, SVM was developed seeking binary classification capable of recognizing sample patterns in pre-defined classes [156].

Currently, SVM supports the task of multi-class learning and it is used to solve problems such as multi-classification. In addition, it has been widely used in the field of artificial intelligence. SVM is responsible for finding the best possible separation boundary between classes/labels for a given set of data that is linearly separable. For SVM, the many separation boundaries that are capable of completely separating classes are called hyperplanes. A decision plane that separates a set of objects with different class members is a hyperplane [157].

An important SVM aspect is the margin, which is seen as a breach between the two lines closest to the class points. The margin is calculated as the perpendicular distance of the support points closest to the vectors. A good margin is the one which has the greatest distance between classes, a lowers margin is a bad one [157].

SVM seeks to find the best hyperplane for a given data set whose classes are linearly separable. SVM builds a classifier according to a set of patterns identified by it in the training examples [158].

Classification problems tend to be more elaborated, requiring optimal separation through more complex structures. SVM proposes the classification of new objects (test) based on available data (training). For that, a set of mathematical functions is used to map the new objects, known as Kernels. SVM kernels are divided into two versions, linear and non-linear [151].

Kernel functions are intended to project vectors of input feature into a high-dimensional feature space to classify issues which lie in non-linearly separable spaces. This is done because as the problem of dimensional space increases, the probability of this problem becoming linearly separable around a low-dimensional space also increases. However, to obtain a good distribution of the complex problem, a training set with a high number of instances is necessary. SVM-based classification uses kernel functions Linear, Radial Base Function kernel(RBF), Polynomial, and Sigmoid [102, 159].

- Linear: it is the scalar product of observations. It is the sum of the multiplication of every pair of input vectors.

- RBF: it maps an input space in a finite dimensional space. It is the most used Kernel in SVM classification.

- Polynomial: This kernel distinguishes a non-linear input space from a curved one. It is known for being more generalized than Linear kernel.

- Sigmoid: Neural networks use the Sigmoid kernel as the activation function. This kernel is part of the class of differentiable, limited and crescent monotonically functions.

Note that SVM-based classification kernel function Linear is considered a parametric model, while the kernel functions RBF, Polynomial, and Sigmoid are considered a non-parametric models.

RBF and Polynomial are both suggestive kernels to separate non-linear application classes from curved ones. Through this choice, more precise classifiers can be obtained. RBF and Polynomial Kernels calculate the separation line in the higher dimension to classify some applications.

An important thing about SVM is the regulation parameters that can be used to configure the SVM [102]. One of them is the C parameter, which is the penalty parameter that represents the classification error or the error term, and it is used to maintain the regulation of the model. SVM optimization depends on controlling how much error can be handed. It is this way that trade-off is controlled between incorrect classification terms and the decision limit. See that the lower the value of C, the lower hyperplane margin and the greater the value of C, the greater the margin will be [160].

Another parameter that also deserves attention on SVM is the Gamma parameter. Low values of Gamma parameter makes so the data does not adapt much to the training data set. Now when the values are higher, the data adapts perfectly to the training set. See that there must have a balance on Gamma values, because values too high can cause an over adjustment and values too low may consider only points close to the margin.

2.4 Classification of Internet Traffic Using Statistical Methods

This section addresses the use of statistical methods for Internet traffic classification. Methods have been grouped by similarities. Tables 2.9 to 2.12 present an overview of previously used statistical methods for Internet traffic classification, as well as their main characteristics and performance.

In these tables, the performance values are given in %. The metrics are indicated as follows: Recall (R), Accuracy (A), Precision (P), F-Measure (FM), F-score (F1), Area Under the Curves (AUC), Receiver Operating Characteristic (ROC), False Positive Rate (FPR), True Positive Rate (TPR), Kappa (K), Geometric mean (G-mean), Specificity (Sp), Sensitivity (S), Not Available (N/A).

To the best of our knowledge, Wootters Distance, addressed in the previous section, has not yet been investigated for Internet traffic classification. Therefore it is not considered in this section, being left as a possible future research direction in the next section.
Some methods have few applications and were little explored for classification, such as Jansen-Shannon and KL. Others were quite explored, such as SVM, which has been extensively explored in this kind of classification, often presenting good accuracy values.

2.4.1 Distance-Based Methods

Table 2.9 details the papers describing distance-based methods for traffic statistical analysis.

2.4.1.1 Euclidean Distance

Euclidean Distance was addressed in several works found in the literature, including on the implementation of some famous machine learning algorithms, such as K-mean, and Nearest Neighbor (NN). In Table 2.9, there is a summary of works around this distance. Zhu *et al.* in [67] proposed a method for classifying an unknown protocol of the application layer based on the Euclidean Distance. In [65], Shi *et al.* discuss the method of extraction and selection of features for classification, where K-Means algorithm with Euclidean Distance were used to group the features. Pereira *et al.* in [63] developed a network traffic classification system based on real-time flow using NN technique and Euclidean Distance. The focus of [66] was to use statistical resources of network flows to identify the generated application, and the Euclidean Distance was used to test the classification algorithm. Singh in [64] used K-Means which calculates the distance between objects by using the Euclidean Distance to group the network traffic applications.

Table 2.9: Works related to distance-based statistical methods (Euclidean Distance, Bhattacharyya Distance, Hellinger Distance).

Method	Work	Year	Characteristics	Performance
Euclidean Distance	Pereira *et al.* [63]	2015	Features: number of packets, number of bytes, elapsed time between the first and last packets, the number of all packets with at least a byte of TCP data payload, the median and the variance of the number of bytes in IP packet, and the number of all packets seen with the PUSH bit set in the TCP header. Applications: HTTP and HTTPS, FTP, WWW, XVTTP, and ISAKMP.	A:87.40-89.86
	Singh [64]	2015	Features: packet length and inter-arrival time including (average, maximum, minimum, and standard deviation), number of bytes transferred, total number of packet in flow and Flow duration. Applications: HTTP, DHCP, ICMP, DNS, and SMTP. Technique: correlation-based feature selection-CFS.	A:55.00-88.00
	Shi *et al.* [65]	2017	Features: extract the multifractal features, multifractal spectrum, largest wavelet coefficient, variance ratio, and cumulative variance ratio. Applications: P2P, WWW, flash+HTTP, IM, SMTP, VoIP, IMAP, and POP. Techniques: method of linear regressions, and Wavelet Leaders Multifractal Formalism (WLMF).	A:55.70-99.80
	Schmidt *et al.* [66]	2017	Features: number of pushed data packets, Median of total bytes in IP packets, port number at server, bytes in the initial window, average segment size, bytes in the initial window, packets with at least a byte of TCP data payload, the total number of Round Trip Time (RTT) samples, variance of bytes in Ethernet packet, packets with the PUSH bit set in the TCP header, and the minimum segment size. Aplications: Postgres, FTP, Oracle, Sqlnet, IMAP, SSH, SMTP, POP2/3, X11, WWW, LDAP, DNS, KaZaA, NTP, BitTorrent, Games, Windows Media Player, and Worm and virus attacks. Techniques:Manhattan Distance, Euclidean, Chebyshev Distance, and Cosine Distance.	A:88.00-94.77, FM:86.90
	Zhu *et al.* [67]	2019	Features: number of labeled protocol in the dataset, protocol flow statistics, longest distance, and average distance. Applications: SMTP, HTTP, FTP, Bittorrent, and POP3. Techniques: clustering, and deep neural network.	A:96.00
Bhattacharyya Distance	Zanin [161]	2013	Features: analysis of the statistical properties, and Data Science analysis.	N/A
	Canali and Lancellotti [162]	2013	Features: Statistical properties of Virtual Machine.	N/A
	Dinani *et al.* [163]	2015	Features: overall mean, averaged in a given time duration of video, standard deviation, skew, R-inverse variance, uniformity, pixel length, and entropy.	N/A
	Sadreazami *et al.* [164]	2017	Features: signal statistics, mean, variance, and time.	ROC:93.15-99.75
	Sameen and Pradhan [165]	2017	Features: spectral, spatial, and texture properties. Technique: fuzzy logic for define rules.	N/A
	Baskoro *et al.* [166]	2017	Features: Probability Density Function (PDF), number of pixel, and color pdf's.	P:96.5, R:96.3
	Laz [167]	2017	Techniques: lagrange multipliers technique, and parallel computing.	N/A
	Shah and Dang [168]	2019/ 2020	Features: probability distribution, modulation pairs, and maximum distance.	N/A
Hellinger Distance	Liu *et al.* [169]	2014	Features: exponential distribution, Erlang distribution, small average distribution distance, and maximum entropy.	N/A
	Safarik *et al.* [170]	2014	Applications: SIP message, SIP attack classification, IP addresses, and specific SIP header values or ports.	N/A
	Luo *et al.* [171]	2015	Features: number of foreground pixels, number of background road pixels, and density ratio. Techniques: regression models, and Pearson correlation coefficient.	A:83
	Wang *et al.* [172]	2017	Features: number of hash functions, size of hash tables, and probability vector.	FPR:0-35, TPR:38-80
	Kumari and Thakar [173]	2017	Features: probability distribution, and synthetic sample value.	AUC: 69-94
	Liu *et al.* [174]	2019	Features: Packet size sequences, and inter-arrival time. Applications: Social, Streaming, Web, and Download.	A: 77.77, G-mean:0-90

2.4.1.2 Bhattacharyya Distance

Shah and Dang in [168] used Bhattacharyya Distance to select the the highest distance features from a test pool. In [164], the temporal analysis of the behavior of the network is established by calculating this same distance. Aiming to calculate the difference among solved and unsolved iEvents that correspond to the traffic density distributions, Zanin [161] also used this distance. In [165], Class separability was maximized using the Bhattacharyya Distance algorithm. In [162], the Bhattacharyya Distance is used to quantify the not dissimilarity of the probability distributions of Virtual Machine (VM) resources usage. In [163], the Bhattacharyya Distance is used to calculate changes of color histogram. In [166], Baskoro *et al.* proposed an algorithm for counting and tracking vehicles using the Bhattacharyya Distance. It is used by Laz in [167] to evaluate detection system performance. In Table 2.9, there is a summary of works around Bhattacharyya Distance.

2.4.1.3 Hellinger Distance

In Table 2.9, there is a summary of works about the use of Hellinger Distance. In [172] the Hellinger Distance was used by Wang *et al.* to find the deviations among sketches. A sketch is a collection of hash tables where Wang *et al.* propose the SkyShield method using the sketch technique aiming to detect anomalies. The Hellinger Distance was used in [175] to perform linear and non-linear transformations aiming the improvement of accuracy in dataset classification. Derivation of the Hellinger square distance was used by Liu *et al.* in [169]. In [173] Kumari and Thakar proposed an oversampling method based on the Hellinger Distance to identify the minority class in the classification. In [170] it is used to measure the not dissimilarity of two probability distributions to implement an attack classifier in a monitoring network. It was also used in [171] on the Linear SVM kernel implementation for the classifier training step. In [174] Hellinger Distance is used on feature value distribution.

2.4.1.4 Wootters Distance

In the research made throughout the databases referred to in this article during the period from 2011 to 2022, applications of Wootters Distance as a classification technique, feature selection and kernel increment in methods such as SVM, for example, were not found in the literature.

2.4.2 Divergence-Based Methods

Table 2.10 details the papers describing divergence-based methods for traffic statistical analysis.

2.4.2.1 Jensen-Shannon Divergence

In Table 2.10, there is a summary of works around the use of Jensen-Shannon Divergence (JSD). In [177] Zareapoor *et al.* applied JSD property to identify information deviation. In [178], Zhi *et al.* proposed an Interest Flooding Attack (IFA), that consists of a resistance mechanism based on JSD. This mechanism can help detect and mitigate Flooding Attack on the network. The obtained values from the JSD calculation were used on [179] to select the features. In [180] JSD was used to calculate the distribution not dissimilarity among original discrete attributes and the generated ones, aiming to evaluate the Anti-Intrusion Detection Autoencoder (AIDAE) performance. In [176], the difference between M1 and M2 (the histograms of two mixture distributions) is quantified using JSD of bin-placement approaches.

2.4.2.2 Kullback-Leibler Divergence

Some works were found in the literature using Kullback-Leibler Divergence (KL) for Internet traffic classification. In Table 2.10, there is a summary of works about the use of this divergence. Kim *et al.* in [181] proposed a network classification with a KL criterion. In [182], it was used to detect video clips. KL was also used in other fields of analysis, such as agriculture. In [183] KL is employed to validate the not dissimilarity of unknown pixels. In [72], KL is used to classification of encrypted internet traffic.

2.4.3 SVM

Several SVM applications for traffic classification were found in the literature. In Table 2.11, there is a summary of works around this statistical method. It was used in [38] with the Linear, Polynomial, Sigmoid and Radial kernels for traffic classification on a Software Defined Networking (SDN). Cao *et al.* in [52] proposed a real-time training model using SVM. It was also used in [54] with denoising schemes to improve prediction accuracy. In [55] Miao *et al.* used SVM to optimize feature selection. To distinguish data representing normal network traffic and Distributed Denial of Service (DDoS) flows, Aamir and Zaidi [62] tested different combinations of parameters on SVM. In [184], Sentas *et al.* developed a video data detection and classification system.

Luo *et al.* in [58] proposed the Least Square SVM (LSSVM) hybrid optimized, a model for short-term traffic flow forecasting. Suresh and Srijanee in [185] used SVM to analyze the traffic data pattern and detect anomalies in order to secure high-volume confidential data transmitted over wireless network. In [57], Xiao used this statistical method combined with KNN to detect traffic incidents. In [61] Dong proposed optimizing SVM method to improve training speed and classification, using this enhanced SVM called Cost-Sensitive SVM (CMSVM) to solve imbalance in network traffic identification. Cao and Fang [186] and Syarif *et al.* [59] optimized the SVM parameters based on the Genetic Algorithm (GA) for Internet traffic classification. Mostafa *et al.* in [187] proposed a new version of this

method named Relaxed Constraint Support Vector Machines (RSVMs) to optimize classification without needing source or destination IP addresses or port information. In [158] Liu *et al.* addressed SVM for Traffic Identification and Classification (STIC) aiming to identify applications, focusing on the duration and quality of YouTube streaming. Aggarwal and Singh in [56] made use of this method to categorize Internet traffic. In [188], a distributed SVM framework was implemented to classify network traffic using Hadoop.

In [51], Hao *et al.* improve a variation of it called Directed Acyclic Graph-Support Vector Machine (DAGSVM) to classify network traffic. In [37], SVM was used to sort network traffic by improving the algorithm to calculate its own resource weights and parameter values for every individual binary classifier. It was also used in [60] to classify large amounts of data. SVM was used in [53] as the basis to implement an optimized model in order to reduce memory and CPU cost in the training phase, called Incremental SVM (ISVM), and a modified version with Attenuation factor (AISVM).

2.4.4 Other Methods

Table 2.12 details the papers describing various other methods for traffic statistical analysis found in the literature.

2.4.4.1 Correlation Information

In Table 2.12, there is a summary of works around Correlation Information (Pearson Correlation). Correlation was used in [22] to boost network traffic ranking performance. In [56], Aggarwal and Singh used a Bag of Flow (BoF) to model correlation information in traffic flows and SVM to categorize traffic by application. The correlation was also object of research on [189], that presented a new traffic classification framework. For that, Zhang *et al.* used the BoF to model information of traffic flow correlation. Besides that they also used a model based on NN. A new classification method that took under consideration the network traffic flow correlation was also proposed by Zhang *et al.* in [189]. In [68], Zhang *et al.* considered real traffic and classified the correlated flows together. In Dong *et al.* [190] presented the disadvantages of using Pearson's Correlation Coefficient to measure the relationship between traffic flows. From the disadvantages, the authors presented a new proposal based on metric correlation quantitatively and accurately.

2.4.4.2 Statistical Kolmogorov-Smirnov and Chi-square Tests

Statistics such as Kolmogorov-Smirnov and Chi-square tests have also been used for traffic classification. In Table 2.12, there is a summary of works around those tests. Neto *et al.* [69] represented traffic classes by using empirical distributions that correspond to the traffic classes signatures, aiming to develop a classifier based in the dark mechanism that combined both Kolmogorov-Smirnov and Chi-square tests. Chi-square was also used in [191] to test if a set of data follows a specific distribution with a degree of confidence.

2.4.4.3 Entropy

Gomes *et al.* in [193] used entropy to emphasize and recognize VoIP P2P traffic flows that belonged to a VoIP session. The developed classifier aimed to identify the flow used in the conversation and focused on the specific characteristics of the voice codec instead of the application used in the VoIP session. In [192], Wang *et al.* used entropy to classify traffic more deeply. In [194], Zhou *et al.*, used entropy for evaluation of encrypted traffic classification. In Table 2.12, there is a summary of works about the use of entropy.

2.5 Discussion and Open Issues

2.5.1 Discussion

Distance and divergence computations are advanced methods of statistical analysis that can be used for classification and, in our context, were used for Internet traffic classification. Through the statistical properties, statistical traffic classification models may be created for a given application. For these methods, sometimes a learning phase is required to build a reference model that can be used to classify traffic.

Statistical classification, also known as logic based classification, allows traffic identification through statistical attributes of the flow. The packet length and duration, the traffic flow idle timing, and the time between packet arrivals are considered examples of statistical traffic attributes or measurements of flow level. On sight of traffic, statistical classification tends to assume and explore unique resources of each application, using data mining techniques to do so most of the time.

Statistical classifiers are light weight and do not require packet payload analysis. In addition, they can achieve the same precision as other methods found in the literature, even using fewer features. These advantages make them suitable candidates for the most restricted configurations. Also, given the current trend towards flow level monitors like NetFlow [195], the ability to operate on statistical characteristics only is an advantageous property for classifiers.

As for the computational complexity of statistical methods, Valenti *et al.* [85] show how tree-based statistical classification can sustain high rate of transference on off-the-shelf hardware.

Figure 2.5 shows the Network Visualization map created using the VosViewer tool. This map was created from the references cited in this article, and based on bibliographic data. The data was read from reference manager files .ris. We chose the co-authorship analysis with fractional counting method, that is the strength of the document is divided by the total number of authors. We do not ignore documents with a large number of authors.

For the generation of our map, we chose at least 1 author per document and found 462 different authors and co-authors. For each author, the total number of co-authors was calculated and the authors with the greatest total link strength will be selected were selected for the chart.

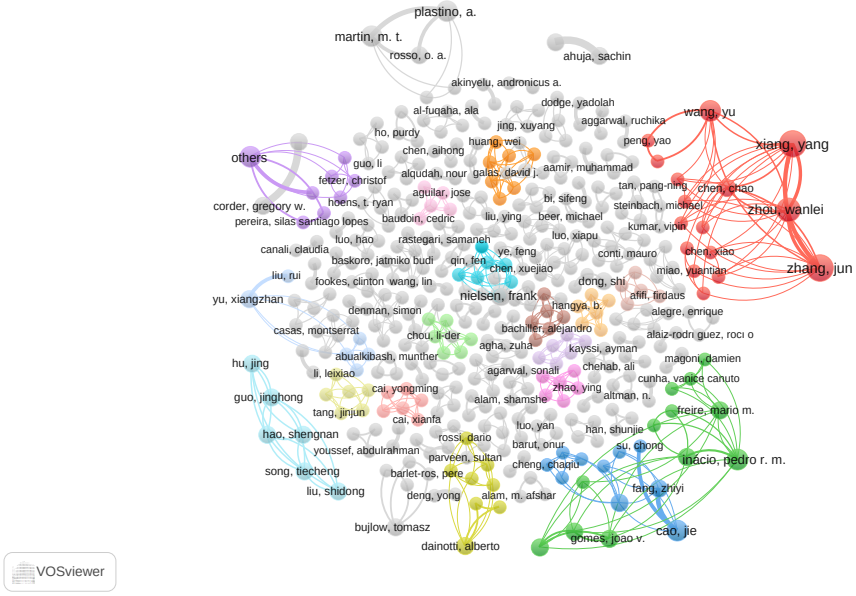


Figure 2.5: VosViewer Network Visualization Map.

2.5.2 Open Issues

In the literature, several significant types of research have been done on traffic classification and how to improve the performance of the classifiers, but there are still some challenges ahead. Considering the technologies and methods applied, most challenges still lie in classifying encrypted, unknown, and P2P traffic in real-time or timely with high precision and low processing power.

In this section, we outline some important open-ended research questions that need to be addressed in this field of research as follows:

- Although SVM has been widely used to classify traffic, traditional traffic classifiers based on SVM have their limitations, among them the high computational cost when it comes to memory, CPU, highly complex training and the difficulties to operate in real time, which makes the real time and timely classification unfeasible. Possible research directions may include the development of new SVM models to address the above issues, following the work in [53].

- SVM still faces resource selection imbalance issues in its training phase. For [196], solving the problems of imbalance in the SVM classification is kept an open issue.

- SVM performance does not absolutely depend on the size of the training data, but also on the quantity of Support Vectors (SV). An open question for research is balancing data volume and complexity because according to [53], with the increase in training data, computational complexity and the occupation of computational resources will also grow significantly.

- One of the issues to be worked on when implementing an Internet traffic classifier using SVM, is choosing correctly the self parameter C because, according to [196], the classification is sensitive to C, in which, if not chosen correctly, SVM, even optimized, produces worse classification results.

- Explore the feasibility of the use of the Wootters Distance for encrypted Internet traffic classification, which, to the best of our knowledge, has not yet been investigated.

- Investigate the use of less explored statistical distances and divergences for encrypted Internet traffic classification, namely Bhattacharyya and Hellinger Distances and Jensen-Shannon Divergence. Although these statistical methods have been investigated for network security and intrusion detection, among other, as reported in this work, we did not find specific applications of these methods for classification of encrypted Internet traffic.

- Explore the combination of the SVM classifier with statistical divergences. In the literature, we find works that combine Euclidean Distance with the K-means algorithm for classifiers, and Kullback-Leibler combined with SVM. However, we did not find classifiers combined with Hellinger Distance, Wootters Distance, Jensen-Shannon Divergence, for example.

2.6 Conclusion

The main purpose of this work was to explore statistical methods and techniques recently used or with the potential to be used in Internet network traffic classification. We provided an overview of the Internet traffic classification process as well as an insight into statistical methods with potential interest to be used as classifiers for encrypted Internet traffic, including those methods that have not yet been explored previously for Internet traffic classification. Then, we reviewed previously used statistical methods for Internet traffic classification, organized by distances, divergences, SVM, and other statistical methods.

Through the literature review, we identified that the most used statistical method for traffic classification is the SVM method. In addition, we also identified several open issues that could be the subject of further research on this topic. More specifically, we identified statistical distances and divergences that have not been much explored regarding to traffic

classification. Actually, they could be used separately or combined with the SVM classifier in order to address challenging problems such as real-time traffic classification and encrypted traffic classification.

Table 2.10: Works related to divergence-based statistical methods (Jensen-Shannon Divergence, Kullback–Leibler Divergence).

Method	Work	Year	Characteristics	Performance
Jensen-Shannon Divergence	Garcia and Korhonen [176]	2018	Features: total amount of Bytes in packets, number of packets in a flow, time between first and last packet sizes, min/max of packet sizes, number of downlink packets, skew/kurtosis of packet sizes, mean of packet sizes, standard deviation, and variance of packets. Technique: Random forest.	A:96, P:88, R:95, ROC:94
	Zareapoor *et al.* [177]	2018		N/A
	Zhi *et al.* [178]	2019	Features: probability distribution, high entropy values signify a more dispersed probability distribution, number of data packets, and sequential time interval.	ROC:98.88
	Barut *et al.* [179]	2020	Features: distribution of each feature, average, length numerical, and variable sizes. Techniques: correlation, random forest algorithm, Principal Component Analysis (PCA), the mean value of array, the length of the array, the maximum value in array, and the minimum value in array.	R:51-93, P:0-100, FM:69-90
	Chen *et al.* [180]	2020	Features: Mean Square Error (MSE), distribution of continuous features, and number of discrete features.	N/A
Kullback-Leibler Divergence	Kim *et al.* [181]	2016	Feature: Maximum Sequence Size (MSS) value. Applications: IMAP, and SMTP. Techniques: markov model, and concept of bag based on port number.	R:20-70, P:39.13-99.85
	Xu *et al.* [182]	2016	Features: stationary stochastic, probability distribution, discrete optical flow approach, number of frames in a video, and length sequence. Application: video. Techniques: bag-of-words paradigm, MPEG motion vectors, and Fourier coefficients.	AUC:59.43-78.80
	Zhang *et al.* [183]	2019	Features: reference time series data, and probability distribution of the NDVI. Application: video.	A:94.80, K:85
	Cunha *et al.* [72]	2020	Feature: Relative frequency. Applications: HTTP and Flash-based, RTSP, MMS, P2P streaming, PPStream, TVUPlayer and SopCast, P2P file-sharing: BitTorrent, e-Donkey, and Gnutella, VoIP: Skype, Google Talk, SIP traffic, FTP and SFTP transfers, Telnet, and SSH sessions. Techniques: KL Divergence calculation, and heuristics.	A:99-100, P:100, FM:85-100, R:74-100

Table 2.11: Works related to Support Vector Machines (SVM) statistical methods.

Method	Work	Year	Characteristics	Performance
SVM	Hao *et al.* [51]	2015	Features: selection algorithm, and Chi-square values. Applications: Mail, and WWW.	A:96.58
	Hao *et al.* [37]	2015	Features: total number of bytes sent by client to server, Fast Correlation-based Filter (FCBF), feature selection algorithm - server port, maximum of bytes in Ethernet packet, average window advertisement, total number of bytes sent by server to client, average segment size, minimum window advertisement, minimum segment size, maximum segment size, and maximum of total bytes in IP packet. Applications: WWW, Mail, FTP-Control, FTP-PASV, Attack, P2P, Database, FTP-Data, Multimedia, and Services.	A:57.38 -97.00
	Syarif *et al.* [59]	2016	Features: Particle Swarm Optimization (PSO), and Feature selection algorithm- Genetic Algorithm (GA). Dataset: Embryonal Tumours, Leukemia, Dexter, Madelon, Internet_ads, Spambase, Musk, Intrusion NSL KDD, and SPECTF Heart.	FM: 76.67-95.68
	Fan and Liu [38]	2017	Features: mean segment size, round trip time, and packet inter-arrival time. Applications: Web, SMTP, POP3, IMAP, FTP, DNS, X11, NTP, BitTorrent, eDonkey, Mysql, Oracle, Windows Media Player, Virus, Worm, Telnet, SSH, and Games.	A:80.59-97.96, P:12.5-99.24, R:2.43-99.93, FM:4.08-99.58
	Cao *et al.* [52]	2017	Features: Feature dimension by principal component analysis (PCA), and Number of folds. Applications: Mail, WWW, Attack, FTP, Database, P2P, Services, and Multimedia. Technique: Correlation□Based Feature Selection (CFS).	A:11.81-99.90
	Aggarwal and Singh [56]	2017	Features: Probability Density Function (PDF), size of the first packets of an SSL, and statistical features. Application: P2P-TV traffic.	A:88.87
	Miao *et al.* [55]	2018	Features: bytes volume, packets quantity, packet size statistic information (Min.,Max., Ave. and variance), duration, and inter-packet time statistic features. Application: EBUDDY, DNS, eDonkey, HTTP, FTP, MSN, IMAP, SMTP, POP3, RSP, RTSP, SMB, XMPP, SSL2, SSL3, YAHOOMSG, and SSH. Techniques: NN, and RandomForest.	A:25.01-92.92, FM:6.14-99.70
	Liu *et al.* [158]	2018	Features: sequence of packets from a source, unidirectional flow, and bidirectional flow, and packets in a specific transport. Applications: Google page, Yahoo page, YouTube, Facebook, Line, BitTorrent, eDonkey, Skype, League of Legends, Twitter, Twitch, Messager, Google Hangout, Instagram, Spotify, Dropbox, OneDrive, KKBOX, MoPTT, Sanguosha, PPS, WooTalk, IRC, Garena Messager, Foxy, Pokémon Go, and QQ,	A:92.54-99.00, R:92.73-98.89, P:92.21-99.00, FM:92.23-98.89
	Sun *et al.* [53]	2018	Features: attributes of the traffic flow, dimension of features, packet size, packet length, inter-packet timing, TCP window size, and information derived from traffic flows. Applications: WWW, P2P, and FTP.	A:82.90-95.40
	Akinyelu and Absalom [60]	2019	Features: Wrapper-based technique, and Filter-based technique.	A:55.11-99.86
	Tang *et al.* [54]	2019	Features: sampling interval, distribution of denoised traffic flow, and Number of forecasting. Techniques: Empirical Mode Decomposition, Wavelet (WL), Ensemble Empirical Mode Decomposition (EEMD), ButterWorth (BW) filter, and Moving Average (MA).	N/A
	Aamir and Zaidi [62]	2019	Features: Bwd Packet Length Std, cumulative entropies of clusters, flow duration, average packet size, and flow.	AUC: 95.04-96.75
	Luo *et al.* [58]	2019	Features: total sample size, true value at period, prediction value at period, particle swarm size, the maximum iteration number, cognitive factor, Social factor, and probability. Techniques: Root Mean Square Error (RMSE), the Equal Coefficient (EC), and Mean Absolute Error (MAE).	N/A
	Xiao [57]	2019	Technique: KNN	N/A
	Sentas *et al.* [184]	2020	Features: image size in pixels, block size, block stride, and block stride in pixel. Application: video, formats: by ImageNet. Technique: Region Of Interest (ROI).	R:78.81-98.55, P:87.73-98.55
	Dong [61]	2021	Features: high and low port number, flow transport protocol, and flow duration, TCP header flag including (TCPflags1, TCPflags2), bi-direction packets length ratio, bi-direction bytes, packets/duration (second), bytes/duration (second), mean packets arrived time (duration/packets), bi-direction packets ratio, bi-direction packets, mean packet length, and tos. Applications: FTP, HTTPS, HTTP, POP3, IMAP, SMTP, SQLnet, Oracle, DNS, NTP, LDAP, Kazaa, Bittorrent, Gnutella, eDonkey, Media Player, Real, SSH, klogin, Telnet, GAME Halflife, SIP, and Skype.	R:60-94, P:60-93, A:84-94, G-mean: 58.30-71.80

Table 2.12: Works related to other statistical methods (Correlation information, Kolmogorov-Smirnov, Chi-square, entropy).

Method	Work	Year	Characteristics	Performance
Correlation Information	Zhang *et al.* [189]	2012	Features: flows sharing, and period of time. Applications: DNS, P2P, SSH/SSL, and FTP. Technique: BoF model-based.	FM:20-99, A:90
	Dong *et al.* [190]	2012	Features: high port number, low port number, bytes of flow, packets of flow, the average packet payload length, the average packet length, the average packet header length, duration flow duration, the average packet arrival interval of flow, byte number per second, packets number ratio of bidirectional flow, packets number per second, packet length ratio of bidirectional flow and byte number ratio of bidirectional flow. Applications: unspecified.	N/A
	Zhang *et al.* [22]	2013	Features: volume of bytes, size and number of packets, inter-packet time, and number of flow statistical properties. Applications: SSL, SSH, and HTTP.	A:58-90, FM:60-95
	Zhang *et al.* [68]	2014	Features: client-to-server maximum packet bytes, number of packets, client-to-server average packet bytes, client-to-server minimum packet bytes, client-to-server minimum inter-packet time, the standard deviation of client-to-server packet bytes, server-to-client number of packets, server-to-client minimum packet bytes, and server-to-client maximum packet bytes.	A: 80-95, FM:88-95
	Aggarwal and Singh [56]	2017	Features: flow statistics technique: discrete statistics.	A:65-95
Chi-square test	Neto *et al.* [69]	2013	Features: length of the packets; Applications: HTTP, Skype, and P2P. Technique: sliding windows.	P:89.36-100, R:91.24-100
	Casino *et al.* [191]	2019	Feature: Chi-square Absolute value. Applications: Compression method ZIP, RAR, BZIP2, and GZIP.	A:68.68-94.72
Kolmogorov-Smirnov	Neto *et al.* [69]	2013	Features: length of the packets. Applications: HTTP, Skype, and P2P. Technique: sliding windows.	P:89.36-100, R:91.24-100
Entropy	Wang *et al.* [192]	2011	Features: frequencies of characters and entropy of consecutive bytes. Applications: encrypted files (AES, PGP and SSL) and compressed files (.gz, .rar, .zip), P2P torrent packets, torrent protocol, SMTP, and HTTP. Techniques: SVM and Sequential Forward Selection (SFS), KL and JSD.	A: 69-81
	Gomes *et al.* [193]	2012	Features: length of the packets. Applications: VoIP, SIP, and Skype.	S:78.57-100, Sp:99.51-100
	Zhou *et al.* [194]	2019	Features: packet's inter-arrival time, packet's sizes, and direction as the neural network's input. Applications: classes (VoIP, Audio, browsing, chat, email, FTP, P2P, video. Techniques: NN, SVM, Random Forest, Naive Bayes, and Logistical regression.	ROC:0.73-0.96, F1:33-95, P: 36-93, R:30-96.

Chapter 3

Impact of Self C Parameter on SVM-based Classification of Encrypted Multimedia Peer-to-Peer Traffic [1]

Home users are increasingly acquiring, at lower prices, electronic devices such as video cameras, portable audio players, smartphones, and video game devices, which are all interconnected through the Internet. This increase in digital equipment ownership induces a massive production and sharing of multimedia content between these users. The supervised learning machine method Support Vector Machine (SVM) is vastly used in classification. It is capable of recognizing patterns of samples of predefined classes and supports multi-class classification. The purpose of this chapter is to explore the classification of multimedia P2P traffic using SVMs. To obtain relevant results, it is necessary to properly adjust the so-called Self C parameter. Our results show that SVM with Linear kernel leads to the best classification results of P2P video with an F-Measure of 99% for C parameter ranging from 10 to 70 and to the best classification results of P2P file-sharing with an F-Measure of 98% for C parameter ranging from 30 to 70. We also compare these results with the ones obtained with Kolmogorov-Smirnov (KS) tests and Chi-square tests. It is shown that SVM with Linear kernel leads to a better classification performance than KS and chi-square tests, which reached an F-Measure of 67% and 70% for P2P file-sharing and P2P video, respectively, for KS test, and reached an F-Measure of 85% for both P2P file-sharing and P2P video for chi-square test. Therefore, SVM with Linear kernel and suitable values for the Self C parameter can be a good choice for identifying encrypted multimedia P2P traffic on the Internet.

3.1 Introduction

According to the 2020 report from Sandvine [117], 80% of the current Internet traffic is generated by three key application classes: video, gaming, and social sharing. Among these applications, video corresponds to the largest traffic volume. More specifically, video streaming grew its overall traffic share during lockdown, which included accelerated video releases to streaming, binge-watching multiple seasons of TV shows, search for entertainment and information on what is happening in the world and video traffic

[1]The content of this chapter was published in the following venue [71]: Vanice Canuto Cunha, Damien Magoni, Pedro R. M. Inácio, and Mário M. Freire, "Impact of Self C Parameter on SVM-based Classification of Encrypted Multimedia Peer-to-Peer Traffic", In: Barolli, L., Hussain, F., Enokido, T. (eds) Advanced Information Networking and Applications, AINA 2022, Lecture Notes in Networks and Systems, vol 449, Springer, Cham, pp. 180–193. DOI: https://doi.org/10.1007/978-3-030-99584-3_16.

from social networks like TikTok. Among video streaming applications, we pay a particular attention in this paper to Peer to Peer (P2P) video streaming. According to the global application total traffic share in 2020 reported by Sandvine [117], BitTorrent is the fourth most used application/platform after YouTube, NetFlix and HTTP-based streaming.

For P2P media streaming, users can take advantage of their aggregated upload bandwidth capacity for efficiently distributing video content among themselves. However, P2P traffic, including BitTorrent traffic, is difficult to detect, prioritize or mitigate, namely inside organizations, specially when protocol obfuscation techniques are used.

Streaming sessions among peers can last for long periods, which can interfere with the available network bandwidth in organizations required to perform critical network-based enterprise tasks. For this reason, Internet Service Providers (ISPs) and network administrators in organizations consider the identification and classification this type of traffic as an important matter, enabling to appropriately managing resource allocation and planning future network growth [197, 198].

On the other hand, nowadays P2P traffic is often encrypted and has varying packet lengths. It is important to classify encrypted multimedia P2P traffic to properly manage the network's resources. In that context, recognizing the different types of apps that use the network's resources and classify them is a pre-requirement that contributes for an advanced management of the network, such as providing Quality of Service (QoS) and price, besides identifying anomalies.

P2P multimedia applications can affect the performance of servers, services or critical applications of organizations or tasks dependent on the network. In this situation, a network administrator may need to impose limitations on P2P traffic, by limiting the transmission rate, differentiating services or even blocking those connections, to ensure a good performance of the internal applications, and / or to enforce rules to regulate the use of P2P systems. The purpose of this article is to investigate the impact of both adjusting the Self C parameter and selecting a particular SVM kernel for specifically classifying multimedia P2P traffic.

3.2 Related Work

Recently, many studies have been carried out to classify traffic with the help of the SVMs [54, 56–60, 62, 184, 196, 199–204]. Some of them have optimized the kernel settings and SVM parameters to improve the classification results, such as in [59, 204]. Self C is one of the parameters of SVM, also denominated as C Penalty, corresponding to the degree of punishment and causing implications on the experimental results. It is important to properly adjust this parameter, as it will directly affect the network traffic classification

Table 3.1: Summary of the main points on traffic classification using SVM addressed in articles found in the literature. In the Performance column: Precision - P, Recall – R, Accuracy – A, F-Measure – FM.

Work	Method	Real-time Operation	Detection of Encrypted Traffic	Performance(%)
Mavroforakis *et al.* (2006) [199]	SVM	No	No	A: -
Yuan *et al.* (2010) [200]	SVM	No	Yes	A: 81.75 and 95.98
Aggarwal *et al.* (2017) [56]	SVM and Naïve Bayes	Yes	Yes	A: 88.88
Aamir *et al.* (2019) [62]	SVM + KNN + RF	No	No	A: 95 – 96.66
Rezvani *et al.* (2019) [196]	Fuzzy + SVM	No	No	A: 99.44
Tang *et al.* (2019) [54]	SVM + Wavelet (WL)	No	No	A:-
Akinyelu *et al.*(2019) [60]	SVM	No	No	A: -
Sankaranarayanan *et al.* (2019) [201]	SVM	No	Yes	A: -
Han *et al.* (2019) [202]	Entropy + SVM	No	No	-
Luo *et al.* (2019) [58]	SVM and Genetic Algorithm	Yes	No	A: 100; FM: 61 – 66.67
Budiman *et al.* (2019) [204]	SVM	No	No	A:-
Şentaş *et al.* (2020) [184]	SVM	Yes	No	A:-
Raikar *et al.* (2020) [203]	SVM, NB, Nearest Centroid	Yes	No	A: 91 - 96

effectiveness. This parameter is responsible for the optimization of the SVM, avoiding an incorrect classification, being thus a regularization parameter [160].

Several works addressed the classification of Internet traffic using SVM, as we show concisely in Table 3.1. However, to our knowledge, the current literature is lacking a study presenting the impact of the adjustment of specific SVM parameters for the classification of multimedia P2P traffic. Therefore, this article addresses this issue.

3.3 Methodology

3.3.1 Classification Method

SVM takes ground on the static learning theory, which aims to provide requirements to pick a classifier that has a good performance. SVM is a supervised learning machine consisting of training and test phases for the available data groups.

It is capable of recognizing sample patterns of pre-defined classes, of supporting multi-class learning, and of implementing the *one-against-one* approach. In this approach, for k classes, $k(k-1)/2$ classifiers are built. Depending on the number of classes, each classifier is trained as if there were two classes only: the intended one and all others. For the implementation of this work, the *one-against-one* approach was used [156].

The choice of the Kernel function is vital in the learning process and classification with the SVM. This choice can have a meaningful role in the results. For Zhongsheng *et al.* in [205], when we use SVM, and properly choose the kernel functions, better results are reached.

As an example, in the training phase the SVM uses techniques to divide data that are not divided with the Linear kernel function use. To determine the separation hyperplane, the *smooth* margin technique allows an error margin of the classification. In SVM's training phase, there is a parameter set by the user that specifies the allowed *smoothness* of this margin.

Some parameters of the SVM method for classification are defined by the user, including the Self C parameter. The C parameter is responsible for the optimization of the SVM, preventing the classification from being done incorrectly. Self C is the main parameter in the SVM, this parameter is responsible for the tolerance and the level of acceptance of error in the classification [38].

The application of SVM for traffic identification requires fine-tuning the algorithm and the adjustment of its parameters for the classification of multiclass traffic. A trade-off must found between the efficiency and the Accuracy of the detection. The proposed method is also applicable to encrypted network traffic.
One of the problems encountered in configuring the classification with the SVM method was the selection of the kernel and its parameter values.

In this work, we explored the usage of four different kernels for SVM: the Linear kernel, the Sigmoid kernel, the Radial Basis Function (RBF) kernel, and the Polynomial (degree = 3) kernel. The higher the C value, the higher the probability to get all training points classified correctly [206].

The main settings for the SVM algorithm are the kernel employed and the error or cost penalty parameter C, which is beneficial in network traffic classification problems as shown in [207]. With respect to the cost variable, we tried several values in the interval $[0.1; \quad 70.0]$. In most implementations of an SVM technique (e.g., in Python), the Self C parameter comes with a default value of 1.0. Table 3.2 shows the parameters used for the classification.

Figure 3.1 shows the architecture of the classifier adopted to perform the classification. Raw data were pre-processed, extracting the distribution of the size of the packets by flows, forming a new database. This new base served as input for the SVM method, where 30% of the base sample was used for the training set, generating the training models and the other 70% of the sample was used for the test set.

Table 3.2: SVM parameters used for optimizing encrypted multimedia traffic detection.

Parameter	Value
Self C	[0.1; 70.0]
Kernel	'Linear', 'Sigmoid', 'RBF', 'Poly'
Degree	3
Gamma	auto deprecated
Coef 0	0.0
shrinking	True
probability	False
tol	0.001
cache size	200
class weight	None
verbose	False
max iter	-1
decision function shape	ovr
random state	None

For classification, SVM uses the models generated in the training set together with the test set. After these procedures, we obtained the exit from the classification. The experiments were executed on a desktop computer running Ubuntu 14.04.5 Operating System and equipped with a 64-bit Intel core i7, 2.93GHz, 6GB of system memory.

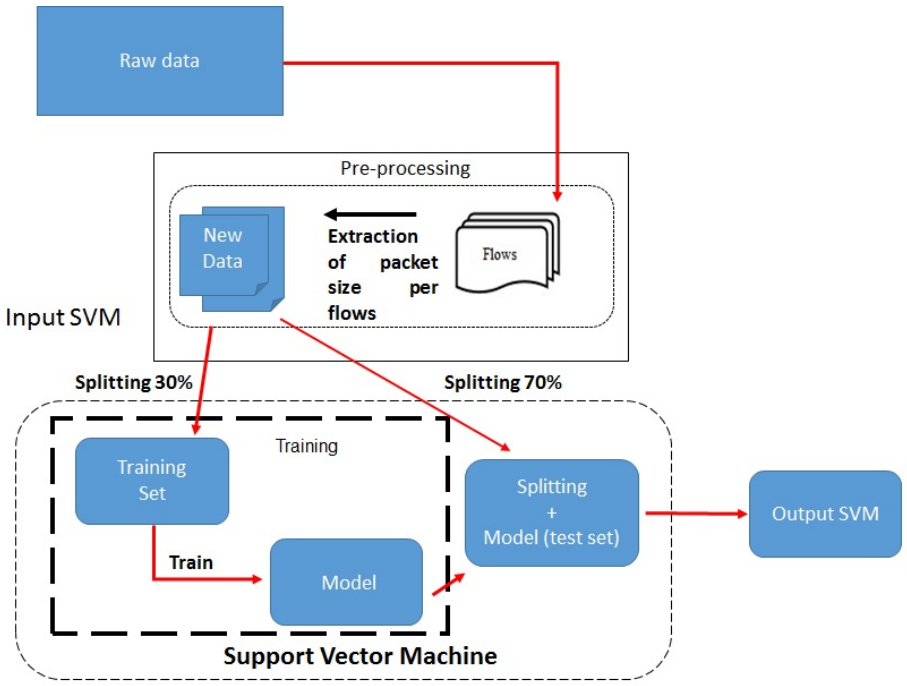


Figure 3.1: Architecture of the classifier.

For classification with SVM, the sklearn module[2] provided by the scikit-learn python library [208] was used and applied to our data set. The classification was divided into 3 steps, as follows:

[2] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

- Step 1 - Data treatment - Generation of the new database: For this step, a script $GeraBaseSVMNew.py$ was created using the python language, whose objective is to convert the raw database into a new database, which was used as input in SVM. First, we treat the flows using the tuple [source IP, destination IP, packet size] we extract from the streams the distribution of the relative frequencies of the size of the packets per stream, forming a new database.

  We create buckets to calculate the distribution of the relative frequency. The features, were the buckets, where each row, has 100 columns, considered a feature. The conversion of the raw database into a new database of relative frequency was necessary to improve computational performance.

  Mapping the classes - The classes were defined based on the IP of each application, for each collective file, formed a Target database with the protocols.

- Step 2 - Training and test phase - The database generated by the script in step 1, was used to generate the models (training phase) and test. To perform the tests, the models were created using the script $SVM\_Multiclass.py$ [209] also implemented in python, in addition, this script was used to classify and return the classification reports [210].

- Step 3 - Data validation and performance evaluation.

3.3.2 Dataset and Classification Features

In this research work, we use a dataset which was also described in a previously published work [72]. The data set contains approximately 25 GB of network traffic traces generated by different Internet applications and services, captured using the `tcpdump` tool and stored on disk. Since the flows were previously stored in a database, all tests carried out in this work have used offline classification only.

The data stored and generated by machines dedicated to a specific traffic, allowed us by construction to obtain the ground truth for the classes.

To accomplish step 1, it is necessary to calculate or update the cumulative probability distribution of the size of the packets per each type of flow, so that we can later obtain the values of the relative frequencies by type of flow, as shown in the Table 3.3.
With the amount of data obtained, the calculation of the distribution function was performed as follows:

- 100 buckets were defined for counting the occurrences of packet sizes.

- In each bucket, the number of observed packets having a size falling within the bounds of the bucket will be counted (Observed Frequency f_i).

- Once the observed frequencies are obtained, the relative frequencies are calculated by Eq. (3.1).

$$fr_i = \frac{f_i}{n}, \quad (3.1)$$

where n represents the total number of transmissions observed in each "Traffic Class" or "Application/Protocol"; Table 3.3 shows the distribution of flows. The classes considered for the traffic analysis are commonly used on the Internet, and are briefly presented in Table 3.4.

Table 3.3: Definition of the buckets for the distribution of packet sizes.

Bucket	Packet size bounds	Frequency	Relative Frequency
1	0 - 15	f_1	fr_1
2	16 - 31	f_2	fr_2
3	32 - 47	f_3	fr_3
.	.	.	.
.	.	.	.
.	.	.	.
100	1584 - 1600	$f100$	$fr100$

Table 3.4: Analyzed traffic flows.

Application / Protocol	Traffic Class	Number of flows
Bittorrent	P2P file-sharing	961
Edonkey	P2P file-sharing	961
Gaming Runscape	P2P Video	418
Gaming War of legends	P2P Video	418
Ppstream	P2P Video	419
Sopcast	P2P Video	419
Tvu	P2P Video	418
Http, web browsing, telnet	Others [a]	179

[a]The other classes are those that are not mapped.

3.4 Evaluation

3.4.1 Classification Results

After obtaining the results (output) provided by the classifier, the results were validated through the ground truth and evaluated using the confusion matrix, the Recall, Precision and F-measure metrics as defined in [211].

The features used as entrance to our classification were relative frequencies and accumulated frequencies. The results obtained through SVM were compared to the Kolmogorov-Smirnov(KS) and Chi-squared tests [69]. KS was used with the aim to select the dis-

tribution that best represents the applications (flows). On the other hand, Chi-squared test [132] was used to compare the relative frequency distribution to the relative frequency of a distribution previously selected that represents a traffic or application class. KS is defined by [69]:

$$D = MAX_x \mid F_{1,n}(x) - F_{2,n'}(x) \mid, \quad (3.2)$$

where $F_{1,n}$ and $F_{2,n'}$ are the accumulated distributions that were compared and for each variable n, n' were determined, that represents the observation numbers.
Chi-squared is defined by [132]:

$$X^2 = \sum_{i=1}^{k} \frac{(x_i - E_i)^2}{E_i}, \quad (3.3)$$

where x_i and E_i $(0 \leq i \leq k)$ are respectively the observed and expected frequencies, and $k \in \mathbb{N}$ represents the number of buckets.

The resulting classifications using the SVM classifier with the Linear, RBF, Sigmoid and Polynomial kernels are shown in Figures 3.2, 3.3 and 3.4. The amount of *support* was 2091 for the P2P video class and 1922 for the P2P file-sharing. The support is the number of occurrences of the class specified in the data set. In the case of our article, it corresponds to the number of items in the class (flows).

We observe that SVM can classify multimedia traffic and that we can optimize the results by adjusting the C parameter, specifically for P2P multimedia traffic. The results demonstrate that there is an impact of the parameter self C on the classification. The factor of that impact for the values self $C = \quad [0.1; \quad 70.0]$, are shown in Figures 3.2, 3.3 and 3.4 for each SVM kernel.

The results obtained in the Linear kernel with C = (0.1, 0.5) were below the values obtained with the default parameter, corresponding to 91% of Precision for the P2P video class with the self C = 0.1 and 89% of Precision for the P2P class file-sharing. For both the P2P video and P2P file-Sharing classes, we obtained the best results with the Linear kernel from self C = 30.0, when the classifier reached its highest classification level for both classes, reaching 99% for Precision, 100% for Recall, and 98% for F-Measure, as shown in 3.2, 3.3 and 3.4.

The results obtained with the RBF kernel showed a significant impact when compared to values of self C lower than the default and values greater than the default, mainly for the P2P file-sharing class. For this class, the impact was a 74% improvement in the performance of the F-measure with self C = 30.0, as shown in 3.4.

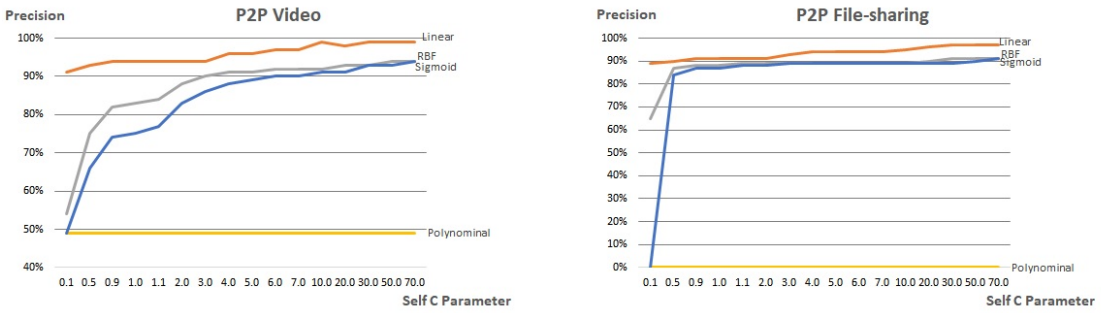


Figure 3.2: Precision, as a function of Self C parameter, of SVM-based classification for P2P video and P2P file sharing traffic.

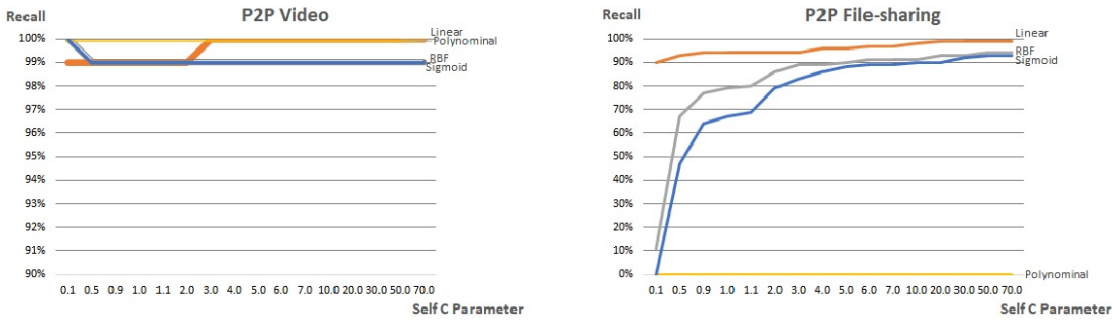


Figure 3.3: Recall, as a function of Self C parameter, of SVM-based classification for P2P video and P2P file sharing traffic.

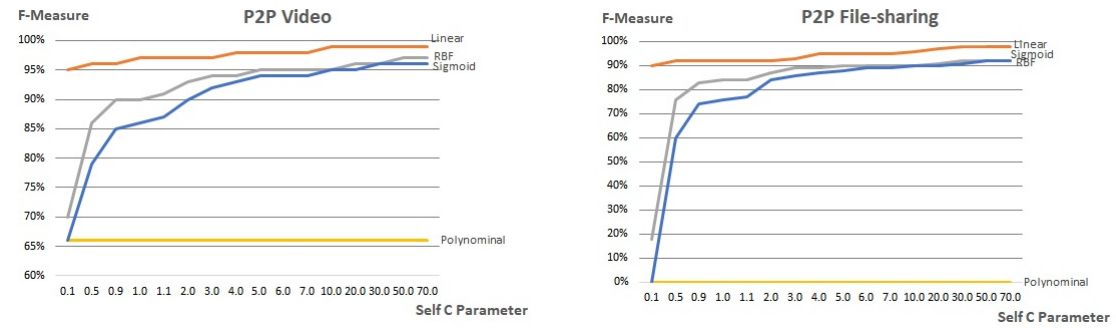


Figure 3.4: F-Measure, as a function of Self C parameter, of SVM-based classification for P2P video and P2P file sharing traffic.

The calculation of F-Measure was important to evaluate the efficiency of the classification, since it represents the value of the harmonic mean between the values found for Recall and Precision. For the P2P video and P2P file-sharing classes, Precision was higher than Recall, indicating that the methodology has greater ability to reduce false positive samples (type II error), than false negative samples (type I error).

Analyzing the impact of self C, in the classification with the Sigmoid kernel, we can see that the biggest impact was on the performance of the P2P file-sharing class. With the self C = 0.1 we have a performance so low that it reached 0% of Precision, Recall and F-measure. For self C = 70.0, we reached the highest performance point for the P2P file-sharing class

where we obtained 91% of Precision, 93% of Recall and 92% of F-measure.

For the classification with the Polynomial kernel, for both the P2P video class and the P2P file-sharing class, there was no impact. The performance for both classes remained the same for all tested self C values. We can conclude that given the analysis of Figures 3.2, 3.3 and 3.4 and for our test scenario, the self C in the Polynomial kernel did not have any impact on the classification performance.

For results with C = 50, it can be seen by the analysis that the P2P video and P2P file-sharing classes achieved 99% and 97% of Precision with the Linear kernel and 94% and 91% of Precision, respectively, with the RBF, showing an excellent performance to discriminate how many instances are correctly classified in these classes. However, the Linear kernel exhibited higher Precision results for P2P video and a slightly better one for P2P file-sharing.

The P2P video classes obtained 100% of Precision with the Linear and RBF kernels, and 99% of Recall, which means that both are able with high performance to identify how many of this class are encounters across the number of elements of that class. For the P2P file-sharing class, the Linear kernel presented a better result for the Recall, although the result obtained for the RBF kernel is also considered.

The method using the Polynomial kernel obtained 49% of Precision for the P2P video class. It could not classify the data set with the relative frequencies used in this article. The results achieved for the P2P file sharing class were very low or close to 0, for all values of C in [0.1; 70.0].

Table 3.5 presents a comparison among classification results obtained with SVM with Linear and RBF kernels, KS, and Chi-square tests. In the classification with the KS statistical method, we obtained a Precision of 84% for P2P file-sharing and 100% for P2P Video. For P2P file-sharing and P2P video, we obtained a Recall of 56%. The F-Measure values were 67% for P2P file-sharing and 70% for P2P video. This means that the classification with the statistical method KS had a lower average performance when compared to the classification with the Linear kernel associated with a C parameter in the range of 30 to 70, and with the RBF kernel associated with a C parameter in the range of 50 to 70.

In the classification with the Chi-square statistical method, we obtained a Precision of 91% and a Recall of 80% for P2P file-sharing, and a Precision of 100%, and a Recall of 74% for the P2P video. The F-Measure values achieved 85% for P2P file-sharing and P2P video. This means that the Chi-square achieved performance average better than KS, with 15% higher for P2P video and 18% higher for P2P file-sharing.

Although these values are better than compared to KS, the statistical method chi-square

Table 3.5: Comparative table of the results obtained with SVM-Linear and RBF in the best range of C parameter, KS and Chi-square.

Performance	Methods							
	Linear kernel (C=[30 -70])		RBF kernel (C=[50-70])		KS		Chi-square	
	P2P file-sharing	P2P Video	P2P file-sharing	P2P Video	P2P file-sharing	P2P Video	P2P file-sharing	P2P Video
Precision	97%	99%	91%	94%	84%	100%	91%	100%
Recall	99%	100%	94%	99%	56%	56%	80%	74%
F-Measure	98%	99%	92%	97%	67%	70%	85%	85%

was low to the mean performance when compared to Linear kernel and RBF kernel with the adjusted C parameter. In Linear kernel with the parameter C in the range of 30 to 70, we obtained 15% more than the performance average when compared to Chi-square. On RBF kernel with C parameter in the range of 50-70, we obtained 7% more than Chi-square for P2P file-sharing and 12% for P2P video.

Our results have shown that the Linear kernel leads to the best classification results of P2P video with an F-Measure of 99%, which is achieved for C parameter ranging from 10 to 70. The Linear kernel also leads to the best classification results of P2P file-sharing with an F-Measure of 98%, which is achieved for values of C parameter between 30 to 70.

3.4.2 Computational Performance

We evaluate the computational performance by measuring CPU consumption (in %) and memory consumption (in MB) during the execution time needed to classify the database using psrecord [3]. Figures 3.5 and 3.6 show the computational performance of the Linear, RBF, Sigmoid, and Polynomial kernels which presented the most significant results in the classification. During our tests, we have seen that the memory consumption was more significant when compared to the CPU consumption.

Analyzing the results, it can be seen that the memory is released by the process at the end of the execution of the Linear kernel. Note that the shortest execution time among the four kernels was obtained for SVM with the Linear kernel, with an execution time of 1.45 seconds.

This does not happen in the RBF kernel at the end of the execution, as we can see in the graph that the process does not release the memory. The execution time of the Linear kernel is relatively shorter when compared to the RBF kernel. The CPU usage (in %) is almost the same for both cases.

The computational performance of the Sigmoid kernel is lower when compared to the Polynomial kernel. The Sigmoid kernel has an execution time which is 2 seconds shorter

[3] `https://pypi.org/project/psrecord`

than the Polynomial kernel. However, it has a longer execution time when compared to the Linear and RBF kernels. As with the Linear kernel, the memory is released as soon as the process is released.

CPU consumption is about the same for both kernels. These results show that the Linear kernel, in addition to showing better classification results, correctly identifies flows that belong to the class and correctly identifies flows that do not belong to the class and has a lower computational cost.

The computational costs of the classification using the KS and Chi-square methods were higher compared to the Linear, RBF, Poly, and Sigmoid kernels. Memory consumption exceeded 600MB for both, and execution time achieved 3000 seconds for the KS method and almost 400 seconds for the Chi-square method. These execution times were considered high when compared to the ones of the SVM kernels.

3.5 Conclusion

SVM classification has shown significantly better results for the Linear kernel, RBF, and Sigmoid, when compared to the Polynomial kernel for the data set presented in this paper. These results can be attributed to the fact that SVM considers properties of the multimedia P2P traffic flow, such as the distribution of packets per flow, an important characteristic to differentiate it from the other protocols and classes found in internet traffic. With the adjustment of the self C parameter, SVM has demonstrated a high discrimination capacity for P2P protocols.

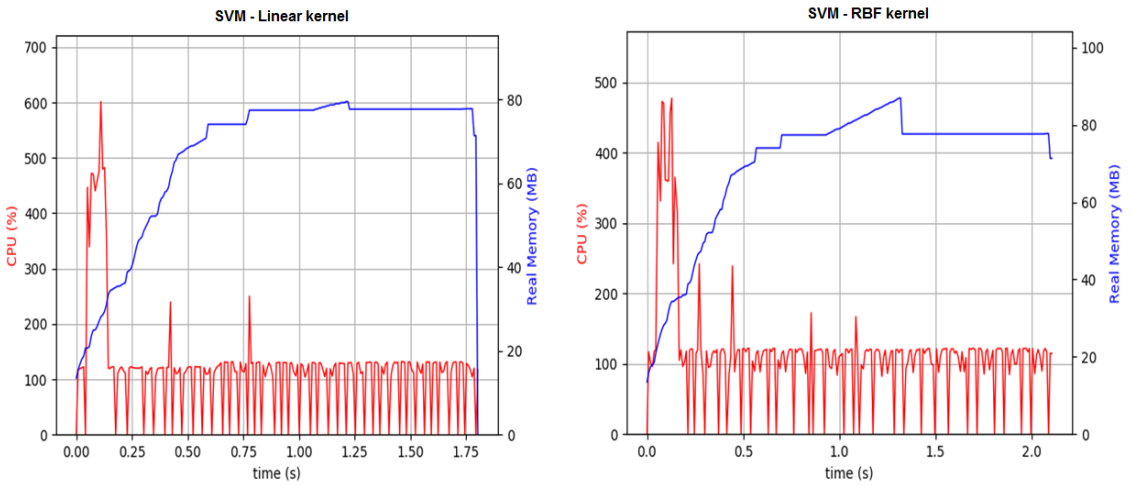


Figure 3.5: Computational resource usage in terms of CPU (%) and memory (MB) of Linear and RBF kernels.

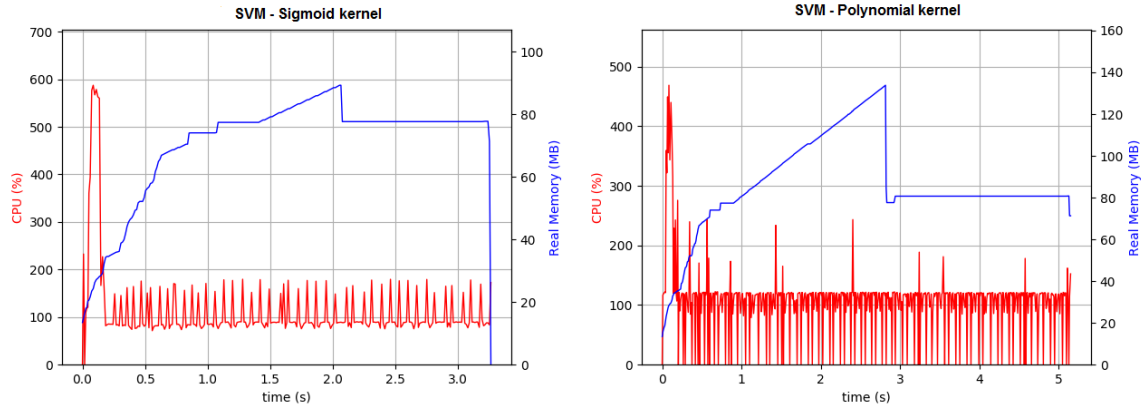


Figure 3.6: Computational resource usage in terms of CPU (%) and memory (MB) of Sigmoid and Polynomial kernels.

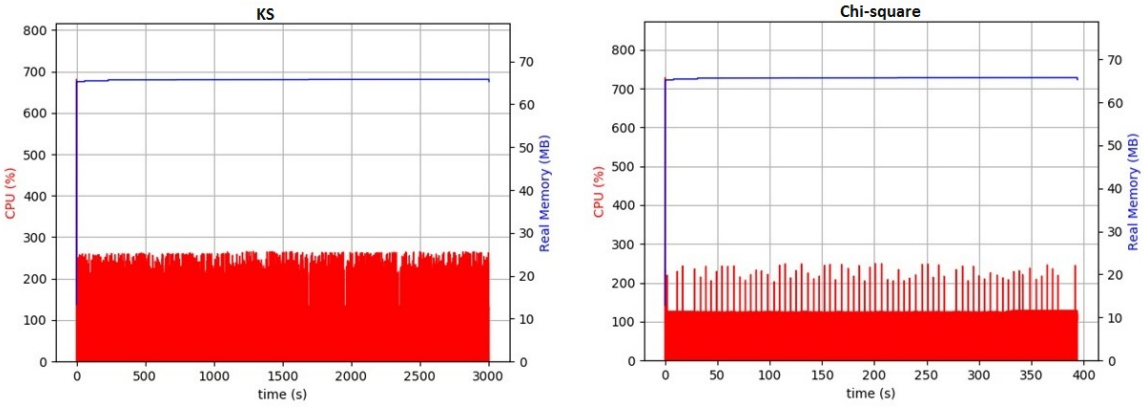


Figure 3.7: Computational resource usage in terms of CPU (%) and memory (MB) of KS and Chi-square.

The more data for the training are entered, the better the classification will be. When we increase the value of self C, we notice that the Precision and Recall values also increase. We can conclude that increasing the values in parameter C reduces type I and II errors and improves the ability to identify flows.

The computational cost for the execution of the SVM method was presented taking into account the use of both CPU and memory during the classification. We have observed that over time, the CPU usage remained the same, while the memory usage increased.

Our results show that SVM can indeed be a good choice for identifying multimedia P2P traffic on the internet. In comparison with the statistical methods KS and Chi-square, the Linear kernel has shown the best F-measure performance for both P2P file-sharing and P2P video results.

For future work, we intend to implement new classifiers for the Internet traffic based on statistical methods such as distances and divergences and compare them with the ones

investigated in this article.

Chapter 4

Classification of Encrypted Internet Traffic Using Kullback-Leibler Divergence and Euclidean Distance [1]

The limitations of traditional classification methods based on port number and payload inspection to classify encrypted or obfuscated Internet traffic have led to significant research efforts focusing on classification approaches based on Machine Learning techniques using Transport Layer Statistical features. However, these approaches also have their own limitations, leading to the investigation of alternative approaches, including statistics-based approaches. Statistical approaches can be an alternative to machine learning ones because statistical approaches can operate in real time and do not need to be retrained each time a new type of traffic appears. In this chapter, we propose two statistical classifiers for encrypted Internet traffic based on Kullback-Leibler Divergence and Euclidean Distance, which are computed using the flow and packet size obtained from some of the protocols used by applications. In our experiments, we evaluate the two proposed classifiers and compare them with a classifier based on Support Vector Machine (SVM). During our study, we were able to classify the traffic by using few features without compromising the performance of the classifier. The experimental results illustrate the effectiveness of our models used for traffic classification.

4.1 Introduction

Internet traffic classification has been the focus of significant research efforts in the past two decades due to its importance for network management and security defense, since it may provide valuable information about the traffic metadata and the eventual underlying motivations [53], [89], [212]. The study of statistical methods to classify network traffic is justified by the fact that many machine learning techniques are supervised and then have limited applicability for real time classification, as they require a new training model be created to a new classification each time new data is presented [53].

One of the major problems in classifying encrypted traffic is that the payload is encrypted, which makes difficult the analysis of the packet contents [29]. Through statistical

[1]The content of this chapter was published in the following venue [72]: Vanice Canuto Cunha, Arturo A. Z. Zavala, Pedro R. M. Inácio, Damien Magoni, Mário M. Freire, "Classification of Encrypted Internet Traffic Using Kullback-Leibler Divergence and Euclidean Distance", In: Barolli, L., Amato, F., Moscato, F., Enokido, T., Takizawa, M. (eds) Advanced Information Networking and Applications, AINA 2020, Advances in Intelligent Systems and Computing, vol 1151, Springer, Cham, pp. 883–897. DOI: https://doi.org/10.1007/978-3-030-44041-1_77.

methods [213] we can estimate, by means of empirical distributions, the behavior of the protocols, and with the help of divergences that show the similarities between two distributions, we can try to efficiently classify the applications that generate the Internet traffic under evaluation.

This article proposes and evaluates the performance of two classifiers for encrypted Internet traffic using statistical methods applied to traffic flows: the Kullback-Leibler (KL) Divergence and the Euclidian Distance. These two classifiers operate at a network flow level and make use of the relative frequency of the packet size to identify applications. KL Divergence has previously been used for speaker identification/verification and image classification [151] and for detection of low-rate Distributed Denial of Service (DDoS) attacks [214]. Euclidean Distance has previously been used for optimizing an artificial immune system algorithm used for flow-based Internet traffic classification [66]. Here, both KL Divergence and Euclidean Distance are used to build classifiers without the need to combine them with other methods.

4.2 Related Work

Significant research efforts have been carried out on the subjects of identification and classification of Internet traffic. Part of them rely on statistics, see, e.g, [69, 193]. Recent research still focuses on making improvements using machine learning, such as [53].
In [89], a classification module focusing on video streaming traffic, based on machine learning, is presented as a solution for networks that require real-time traffic analysis. In our work, we use real-world traces of encrypted traffic, including traffic generated by Peer-to-Peer (P2P) applications such as eDonkey, BitTorrent, and Gnutella.

Some research works propose classifiers based on learning methods and corresponding signatures [215], [216]. A large number of research works, regardless of the classification method, make use of traffic flows or packet sizes (e.g, [193], [217].

Statistical tests such as Chi-squared and Kolmogorov-Smirnov were used for traffic classification in [69], representing the classes through the corresponding signatures and the empirical distributions. The entropy was also used to measure and represent important differences regarding packet heterogeneity [218]. Exploring the heterogeneity of the packet sizes was accomplished by using samples obtained from a sliding window. For our work, we explored the characteristics of packet sizes, using all the relative frequencies of the size of packets of flows and not just samples of flows. Still in [218], Gomes *et al.* proposed an online classifier to separate traffic generated by P2P and non-P2P applications and a new method to identify VoIP sessions [193].

Peng *et al.* propose in [217] a statistical classification approach that uses the Message Size

Distribution (MSDC), which aims to identify the network flows precisely and the Message Size Sequence (MSSC) in real time. Such technique provided very good detection results, making a decision after inspecting less than 300 packets with a 99.98% precision, but do not display recall and F-Measure values.

In [219], Extreme Learning Machine (ELM) methods were used to classify Internet traffic. The Extreme Kernel Learning Machine (EKLM) approach was applied to the data. In particular, a Genetic Algorithm (GA)-based software was implemented for selecting the parameters used in the Extreme Kernel Learning Machine with Wavelet (WK-ELM) algorithm. This approach reached a precision rate over 95%.

4.3 Statistical Methods

4.3.1 Kullback-Leibler Divergence

The KL Divergence or relative entropy is a distance measured between two probability distributions and was introduced by the mathematicians S. Kullback and R.A. Leibler in 1951 [148], [149], [150], [20]. These researchers started with the assumption that two probability distributions differ more or less according to the possibility of discrimination between them by means of a statistical test.

The KL Divergence is a special case of a wider class of divergences. By using this method, we can infer a similar behavior, or divergence between two distributions [20]. Considering that $D_{kl}[p||q]$ is a function, p_i and q_i are two probability distributions, we have:

$$D_{kl}[p||q] \quad = \quad \sum pi \log\left(\frac{1}{pi}\right) - \sum pi \log\left(\frac{1}{qi}\right). \quad (4.1)$$

Then, it can be assumed that the Kullback-Leibler Divergence is represented by equation 4.2:

$$D_{kL}[p||q] \quad = \quad \sum_{i=1}^{N} pi \log\left(\frac{qi}{pi}\right), \quad (4.2)$$

where $D_KL[p||q] >= 0$ and $D_KL[p||q] = 0$ if and only if $p_i(x) = q_i(x)$, N defines the number of samples, i defines the number of the initial sample, pi defines the relative frequencies of the known class, qi defines the class relative frequency to be compared.

Note that, despite being also known as a distance, KL Divergence cannot be considered as a distance metric, since it does not meet the symmetry property, i.e., $D_KL[p||q] \neq D_KL[q||p]$. We mapped the behavior of some protocols through distributions (relative frequency) and used this mapping to classify traffic, assuming that each known distribu-

tion is pi and each unknown distribution is qi. KL Divergence was used to implement one of the classifiers described in section 4, more specifically the Statistical Analysis module.

4.3.2 Euclidean Distance

To calculate the distance between two traffic classes, one must consider the probabilities of each traffic class, in our case, p_i defines the relative frequencies for all the discrete values i (possible packet sizes) of the traffic class 1, representing the known traffic class, while q_i defines the relative frequencies for all the discrete values i (for possible packet sizes) of traffic class 2, representing the unknown traffic class. Therefore, the distance between both classes of traffic may be given by the following equation:

$$D_E[p,q] = \sqrt{\sum_{i=1}^{n} (p_i - q_k)^2}. \quad (4.3)$$

In this work, we use Euclidean Distance to compare its results with the ones obtained with KL and to test the efficiency of this method, when mapping the behavior of the relative frequency for each protocol. The Euclidean Distance is used to implement the second classifier described in section 4.

4.4 Traffic Classification Using Kullback-Leibler Divergence and Euclidean Distance

This section details our approach to classify Internet traffic. We describe the traffic features used for the classification, the classifier architecture incorporating the divergence or distance and system modules, as well as the rules implemented for traffic classification. For the purposes of this work, calculating the *distance* refers both to calculating the *divergence* or the *distance*.

4.4.1 Features

It is possible to create a new set of data from the original data, which contains important information obtained through the new features found [65]. This new set of data can present a smaller number of attributes than those found in the original set, bringing us as a benefit a lower dimensionality. In order to obtain this new set of data, we extract some features of the original database, such as packet sequence numbers, IP source and destination, packet size, and time [17], where we apply a different view of the original data to reveal its important characteristics.

We call the original base, the trace of traffic that has been stored after collection of network traffic. We consider each flow as a time series, which generates a standard characteristic:

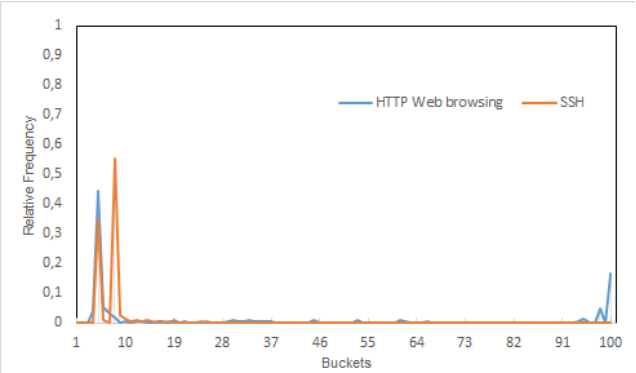


Figure 4.1: Relative frequency of packet size per application, extracted through the sampling process, for SSH and HTTP Web Browsing.

the relative frequency of packet size per application.

We observe that the flow (source IP and destination IP) of the application would give us a feature that could be extracted, the relative frequency of packet size in each flow. We believe that the distribution of sizes is of great importance to characterize the traffic, since it represents each protocol in a unique way, forming a signature for each one of them.

Figure 4.1 illustrates the relative frequency distribution of packets for some available applications in our new data set. As we can see, each protocol behaves in a unique way and exhibits a unique distribution, which we call a signature. We can use such behavior to measure the divergences to known distributions and to start the classification.

The X-axis of fig. 4.1 represents the number of buckets. Buckets were defined with the purpose of creating a histogram with intervals of 15 bytes. The maximum size of the histogram interval is 100. Buckets are required to calculate the relative frequencies of packets. The Y-axis represents the calculated relative frequency of HTTP Web browsing and SSH.

4.4.2 Classification Approaches

Divergence or distance is the measure of separation between two distributions, it indicates how similar or different two traces are. But for this we must insert some parameters so that, with that there is the comparison. The architecture for our classification system contains four modules, as shown in Fig. 4.2: packet capture and pre-processing, statistical analysis and stored signatures, classification and validation.

4.4.2.1 Packet Capture and Pre-Processing

This step deals with the traffic capture and storage. Traffic generation was done in the laboratory in a controlled environment, and the capture was performed with the *tcpdump* and *windump* tools. The files were stored in the .pcap file format. Then these file were

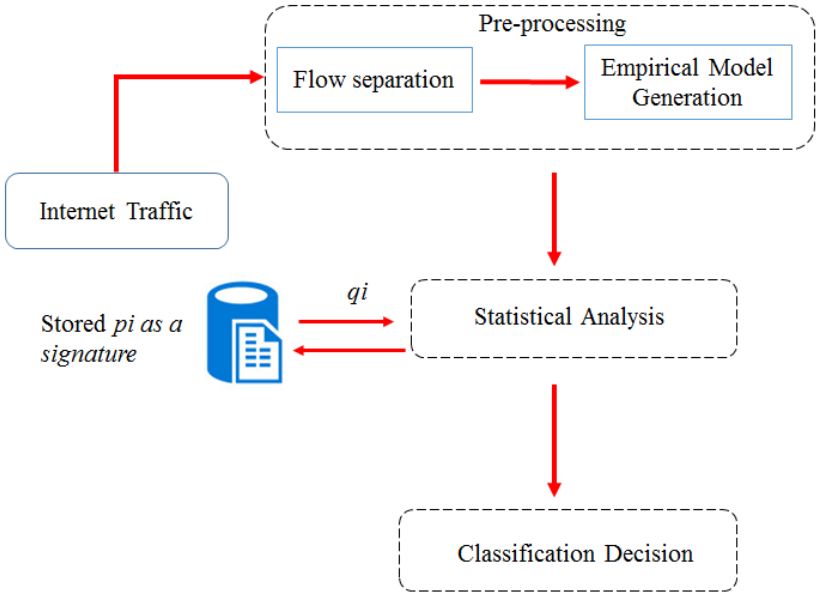


Figure 4.2: Proposed classifier: Architecture.

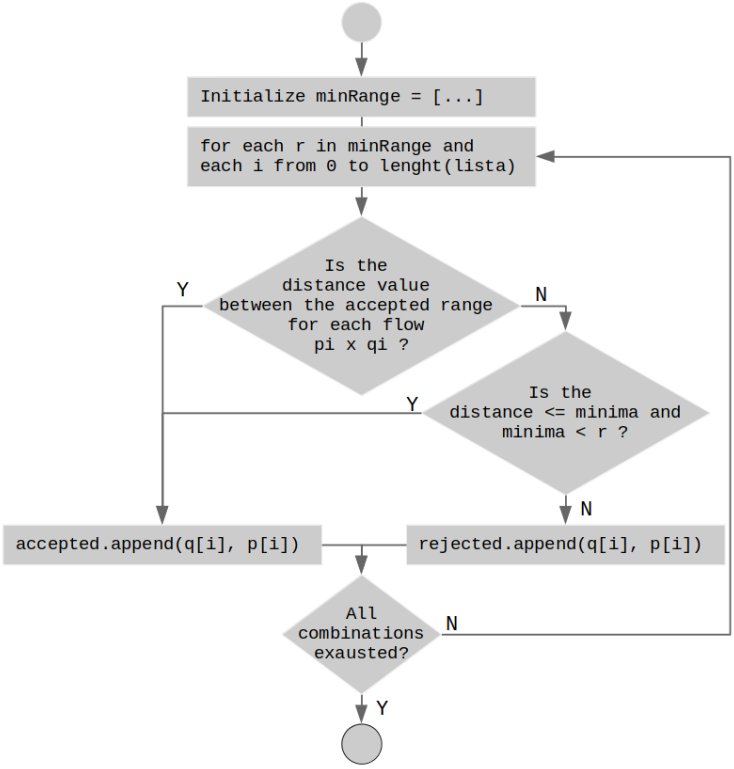


Figure 4.3: Flowchart of rules process.

converted to the .txt format. This conversion was done via command line with the command $tshark - rsrcfile.pcap - t$. Afterwards the files were separated into several files by application with the command $tcpdump - ettnnrfile.pcaphostIPorhostIP - wfile.pcap$.

First, we separated our data set into individual trace files and collective trace files. We

classify as individual trace files those in which we know what applications have been used to generate them. Then, we consider the application flow, source IP address and destination IP address, packet sequence and number of packets belonging to the same application and IP addresses.

Empirical Model Generation In this step, relative frequency files are generated in order to feed the new database. It was necessary to create a file with relative frequencies corresponding to the traces with known traffic. These files were created by means of samples. We developed a python script to create the files with relative frequencies. For the generation of relative frequencies, it was taken into account the Maximum Transmission Unit (MTU), equivalent to 1514 bytes. This reference was crucial for the construction of the packet size histogram.

Each byte packet was allocated in a sort of bucket. We take into account the size of 16 bytes to construct the histogram, so if we take the 1514 bytes and divide by 16 bytes, we will have an approximate value of 100 buckets, or intervals. Applying this logic to each flow, we are then able to calculate the empirical distributions - in our case, the relative frequencies.

Samples Generation In this step, the program reads files from collective and individual folders and separate flows. The samples were generated from both individual and collective trace files. Each sample represents a relative frequency file generated for each protocol. The relative frequencies of the individual traces are stored in a different data set, since they will serve for signature identification purposes during the trace comparison phase.

For flow grouping and separation, an application was built using the Python language, which allowed us to follow the entire flow and extract, with the construction of the histogram, the distributions of each flow. No feature was used to examine packets.

4.4.2.2 Statistical Analysis and Stored Signatures

In this step the distance and divergence based on the relative frequencies of each flow are calculated. In order to know if two traffic traces belong to the same class, we use the empirical distribution, in our case the relative frequency that each trace has, and compare the two frequencies by applying the KL or Euclidean calculation, and obtaining a distance as output.

We begin the comparison phase between the distributions, where in this phase, the divergences are used, as shown in Figure 4.2. In this step we consider the generated and separated files in the preprocessing module and rename these files in samples pi and qi.

Now, pi for individual flow files and qi for collective flow files. Relative frequencies individual are stored as a signature and compared to Relative Frequencies collective using the distances of KL and Euclidean.

For calculating distances, we choose which method we will apply, either Kullback-Leibler or Euclidean, and compare the relative frequency lists pi and qi generated. After this comparison, we have the values of the distances found between the two lists. This distance alone does not represent anything, therefore, in the classification step it is necessary to make use of heuristics. Note that, individual relative frequencies are stored as a signature and compared to collective relative frequencies using the KL and Euclidean distances.

4.4.2.3 Classification and Validation

The classification of flows based on rules is made in this step. Note that, at this stage, we already have all the distances calculated between the flows. For divergences, the closer to 0 the most similar the two protocols are, the closer to 1, the most different the two protocols are likely to be. After the calculation, and having the values of the obtained distances, classification rules have to be created so that the classifier can make a decision. These rules are based on statistical heuristics after several iterations of distance calculation.

Initially, the rules for classification were based on dissimilarity values, where dissimilarity can be defined as follows: 0 if the attribute values match and 1 if they do not match. Since we will not always have distances equal to 0, or equal to 1, we need to find a cut-off threshold for the classification to be done more efficiently. For the cut-off, after debiting all distances, we created two lists to insert accepted and rejected flows according to established rules.

As we mentioned earlier, we need a threshold to know when the flow could be accepted or rejected. At first, we used a heuristic where we applied a range of intervals so that flow distances could be accepted if they were within the interval or rejected otherwise. After several tests, we found that in the rejected list there were flows that could have been accepted, but given the cut-off threshold, had been rejected. So we found that this methodology to accept or reject the flow would not work in our case. Next, we defined another rule, this time making use of the average packet size found in each flow and also adding their standard deviation. Unfortunately, the values on the rejected list remained high. It is worth mentioning that when we checked the rejected lists, there were many flows that should have been accepted instead.

After several analyses, we chose to include in our list of accepted flows, the shortest distances found among all calculated distances. We also decided to use the standard deviation of the distances calculated by the samples as the cut-off threshold, inserting in our distance list all flows that were in the interval $[average, SD]$. Using this technique, we

were able to significantly reduce cases of false positives and false negatives, but it was not enough to deliver satisfactory results.

In order to create an efficient rule, we calculate the relative frequency of a given known protocol that is in our signature database, and compare it with all relative frequencies of unknown protocols. After this comparison, several distances will be generated. The first step is to select the minimum distance among the calculations to apply the classification rule. The second step, after several and continuous tests with each protocol, was defining acceptance baseline values that would define five different minimum ranges.

For each range value, the pre-calculated minimum distance of each protocol must necessarily be within 0 and the current range of the loop. Once the classification is over, the one yielding the best F-measure will be returned, as per auxiliary variables within the classification loop. The MaxR variable was created to store the minimum range of the best result, while MAxMatrix variable was created to store its confusion matrix. The maxFmeasure variable, initially set as -1, will then store the best f-measure among the results. Classification rules are illustrated in the flowchart shown in Figure 4.3.

4.5 Performance Evaluation

After the classification process of the samples, we checked and validated the results of the classification using the ground truth. For this, we created two new lists: in the first list we have the number of items classified as accepted and that were actually found in the mapping dictionary according to the source IP and the destination IP addresses corresponding to each protocol, and in the second list we have the number of items classified as rejected.

Note that even though a distance is in the accepted list, the flow may not actually belong to the corresponding protocol, and even though it is in the rejected list, the flow might actually belong to the corresponding protocol.

4.5.1 Data set and Ground Truth

For the study and analysis of the traffic, it was necessary to collect traffic traces and to build a database containing the traces to be analyzed. The data set is composed by traces, generated by different Internet applications and services. Network traffic used in this analysis was captured next to the source where it was generated, obtained in a controlled environment, where only one application was running in a certain computer. All traces were collected by using the TCPDUMP and WINDUMP tools, implemented by the capture libraries: libpcap in the Linux platform, and Winpcap in the Windows platform.

Table 4.1: Dataset Characteristics.

Data set	Volume (GB)	TCP (%)	UDP (%)
Data set 1	8.80	78.35	21.59
Data set 2	8.60	82.77	17.18
Data set 3	7.97	77.13	22.84

This way, we observed the properties both scenarios and established the ground truth of the traffic records. In order to analyze the empirical distribution of the traffic packet size, an application was built to create the relative frequency distribution for each collected traffic.

In this study, we chose services or applications that are widely used, heavy bandwidth consumers or raise more challenges from the perspective of traffic and network management, ending up with a set of services with varied characteristics. The classes considered for the traffic analysis were commonly used in the Internet, as listed below:

- Web browsing: browsing of general web pages, excluding media streaming.
- On-demand and live streaming: HTTP and Flash-based, RTSP, MMS, etc.
- P2P streaming: PPStream, TVUPlayer and SopCast.
- P2P file-sharing: BitTorrent, e-Donkey, and Gnutella.
- VoIP: Skype, Google Talk, Session Initiation Protocol (SIP) traffic.
- File transfer: FTP and SFTP transfers.
- Remote session: Telnet and SSH sessions.

In this work, “HTTP download” is used to refer to a long and continuous download of a large file using HTTP, while “Web navigation” is used to refer to the common activity of visiting Web pages through a Web Browser. As described in Table 4.1, we used three data sets, which include a total number of 92.317 traffic flows, corresponding to 35.317.091 packets and approximately 25.37 GB of traffic. The time to acquire all the data sets was approximately 61h.

Ground truth is widely used by researchers who collect their own traffic traces to test the accuracy of their solutions. In our work it is possible to establish ground truth because traffic was collected in a controlled environment where each computer was running only one application.

With this, we can map exactly the trace for a given application, calculate the relative frequency of the trace and compare it with the output of the classifier. The individual traces were used for the generation of our ground truth and the files with the collective traces were used for our tests.

4.5.2 Performance of the Classifier

To address these issues, we used the confusion matrix, which allows us to obtain the performance of the classifier, based on the values of TP (True Positive), TN (True Negative), FP (False Positive) and FN (False Negative). For the values of TP, the classifier added them to the list of accepted flows which, after the analysis and classification, are considered to belong to that flow.

For FP values, we have the protocols erroneously classified as belonging to the flow. For the values of TN, the classifier correctly understood that the protocol did not belong to the flow, inserting it in the accepted list. Finally, for FN, the classifier erroneously understood that the current protocol did not actually belong to the flow, inserting it in the rejected list. For performance evaluation, we use the classical performance metrics defined in machine learning textbooks: accuracy, precision, recall and F-measure.

Table 4.2 presents the accuracy, precision, recall and F-Measure for the Kullback-Leibler and Euclidean methods for all 8 classes tested. After evaluating the performance of the classification proposing the use of Kullback-Leibler Divergence, we compared the performance with methods already found and tested in the literature [66], [106], [156].

Table 4.2: Performance results for Kullback-Leibler Divergence and Euclidean Distance. Acc: Accuracy, Rec: Recall, Prec:Precision, F-M: F-Measure.

Traffic Category\Protocol	Methods							
	Kullback-Leibler				Euclidean			
	Acc.	Rec.	Prec.	F-M.	Acc.	Rec.	Prec.	F-M.
Web browsing	99%	74%	100%	85%	99%	67%	100%	80%
HTTP download	99%	83%	100%	90%	99%	66%	100%	80%
Streaming On-demand and live	100%	100%	100%	100%	100%	100%	100%	100%
P2P streaming	99%	86%	100%	92%	99%	85%	100%	91%
P2P file-sharing	99%	76%	100%	86%	99%	73%	100%	84%
VoIP	100%	100%	100%	100%	100%	100%	100%	100%
File Transfer	100%	100%	100%	100%	100%	100%	100%	100%
Remote session	100%	100%	100%	100%	100%	100%	100%	100%

For our comparison, we use of the Euclidian Distance and Support Vector Machine (SVM). For testing with SVM we use the $sklearn - import - svm$ [208] function provided by the Python library and applied it to our data set with the default parameters, except for the $Self$ parameter, which for our case we use C = 50. The largest problems encountered in setting up the SVM model were how to select the kernel function and its parameter values.

When there are large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. We use four different kernels, which are Linear, Polynomial, Sigmoid, and RBF kernels. For testing using SVM, we use the training module and change the Statistical Analysis KL module to Classification-SVM.

A summary of the results obtained from the classification using SVM is shown in Table 4.3. The method using the SVM RBF kernel and the Sigmoid kernel could not classify the data set relative frequencies used in this article, presenting results too low or close to 0 (not shown in the table). For the category of Web browsing traffic, the SVM method has a classification capability, the Euclidean and KL methods presented very similar results, having Kullback-Leibler higher Recall and F- Measure values.

Table 4.3: Performance results for SVM with Linear and Polynomial kernels. Acc: Accuracy, Rec: Recall, Prec:Precision, F-M: F-Measure.

Traffic Category\Protocol	Support Vector Machine - SVM							
	Linear				Polynomial			
	Acc.	Rec.	Prec.	F-M.	Acc.	Rec.	Prec.	F-M.
Web browsing	74%	74%	74%	74%	0%	0%	0%	0%
HTTP download	81%	82%	35%	4,9%	0%	25%	10%	15%
Streaming On-demand and live	57%	58%	57%	57%	0,90%	1%	9%	2%
P2P streaming	99%	99%	99%	99%	0,57%	10%	97¨%	10%
P2P file-sharing	98%	98%	97%	98%	0,0005%	0%	100%	0%
VoIP	100%	100%	100%	100%	0,22%	2%	100%	4%
File Transfer	0%	0%	0%	0%	0%	0%	0%	0%
Remote session	0%	0%	0%	0%	0%	0%	0%	0%

For the HTTP download category, the SVM method was unclear. Although the KL method had higher values than the SVM method, the values of Recall and F-Measure are still below acceptable values to state that for this type of application the method can classify this traffic category efficiently.

For on-demand and live streaming, the SVM method obtained results below the values found for the statistical methods. Analyzing the results, we realized that for this type of traffic, both KL and Euclidean methods, can classify efficiently and effectively. The results show that both methods are far superior than any of the SVM kernels.

For P2P streaming, Linear, Polynomial, and Sigmoid SVM kernels have at least one metric considered reasonable for classification, but overall they are insufficient. This leads us to conclude that for video and video streaming traffic, the KL and Euclidean methods are efficient for identification and classification of these applications.

For P2P file-sharing, the KL and Euclidean methods gave excellent results for the precision and F-Measure metrics and good results for the Recall metric. The SVM Linear kernel yielded results that are considered excellent.

For VoIP classification, the Linear kernel SVM method presents significant results. KL and Euclidean methods gave excellent results for this type of traffic, reaching 100 % precision, Recall and F-measure results, which means that for this application the KL and Euclidean methods were very promising.

For the File Transfer traffic category the KL and Euclidean methods can also, given the obtained values, present an excellent classification capacity. For Remote session applications, the values achieved by the KL and Euclidean methods were 100% for the Accuracy because, although we have it mapped in our individual database, there are no corresponding files in the collective test database. The classifier states that all tested files do not contain Remote sessions, which in our analysis is correct, since we do not have any.

By making a comparative analysis between the statistical methods, we conclude that KL and Euclidean methods obtain the same results for most cases. When comparing the F-Measure results obtained in both, we see that for the HTTP Download and Web Browsing traffic only, the values obtained by the KL method were higher, but we cannot say that this method is superior to the Euclidean one, given that for the other results, the values were the same or relatively similar.

It is interesting to note that SVM is considered an excellent classifier in the literature, but when we do not have many features, we notice from the results that SVM does provide satisfying results. Therefore, building network traffic classifiers using statistical methods can still be considered a viable alternative for encrypted traffic.

4.5.3 Resource Usage

The experiments were executed on a 64-bit Linux, desktop computer, equipped with an Intel (R) Core (TM) i7 CPU 2.93GHz, 6GiB system memory and a PCI Express Gigabit Ethernet Controller based on the RTL8111 chipset.

For computational analysis, the *psrecord* [220] tool is used. This tool records the core and memory activities of a process. In order to measure the computational performance of each method, we used the activity of the process that the method refers to.

We did not use packet number or host/port, as this data was used only as information to convert the raw database into relative frequencies. When we look at fig. 4.4 and fig. 4.5, we find that memory usage remains stable whatever the processing consumption of the chosen classifier.

When the classifier is started, the CPU and memory time start at 0. If we observe the CPU usage over time, we see that KL and Euclidean classifiers both use a similar CPU percentage to process the information and compare the relative frequencies. Although not shown, the SVM classifier requires more CPU especially at the beginning of the process. Regarding the memory usage over time for each statistical classifier, they are very close, being slightly larger for the KL classifier, because KL is more demanded.

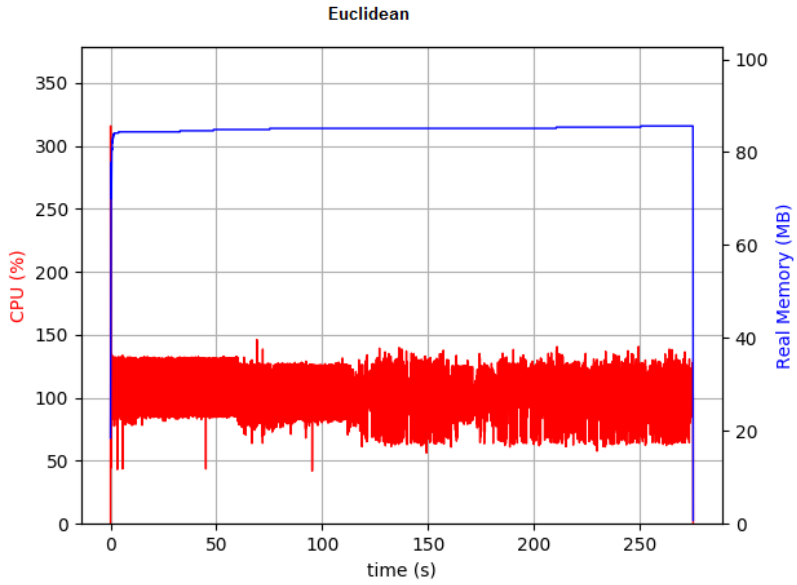


Figure 4.4: CPU and memory consumption from the beginning of the analysis of the trace to end of the classification (execution time) for the classifiers based on: Euclidean Distance.

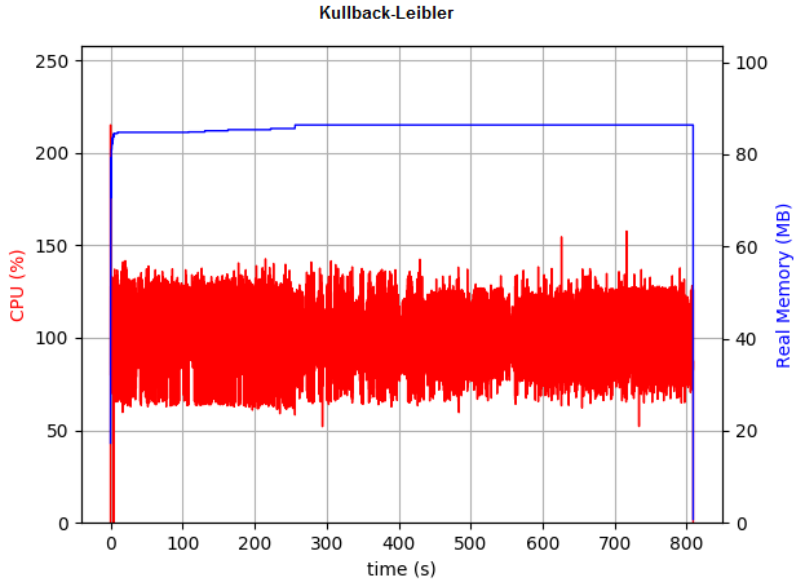


Figure 4.5: CPU and memory consumption from the beginning of the analysis of the trace to end of the classification (execution time) for the classifiers based on: Kullback-Leibler Divergence.

4.6 Conclusion

Our results for Internet traffic classification were presented using Kullback-Leibler (KL) Divergence, Euclidean Distance and SVM. According to the results obtained, we can conclude that the SVM method with kernel set to default parameters is not effective to classify flows based only on packet size: Linear SVM was only efficient for classifying P2P streaming and SVM Polynomial was efficient for classifying P2P streaming, P2P file sharing, and

VoIP.

In contrast, the KL and Euclidean methods were able to classify all tested applications, standing out in the streaming and P2P classification, where for almost all cases it was efficient to identify them with a high precision. We conclude that both the KL Divergence and the Euclidean Distance were efficient and that, in cases where the KL Divergence was superior, it was not significantly better than the Euclidean Distance. The performance of the statistical method is considered good as long as CPU usage remains almost constant when packets are being processed and analyzed. As for memory usage, it fluctuates according to the requests of each method. For future work, we intend to evaluate other statistical divergences indicators by performing new tests on encrypted traffic.

Chapter 5

Classification of Encrypted Internet Traffic Using Statistical Methods [1]

Internet traffic classification allows the identification of protocols used in different services on the Internet, based on features presented in packets or flows generated by those services. Such traffic classification and identification are performed through different techniques such as Machine Learning (ML), Deep Packet Inspection (DPI), or statistical methods like distances and divergences, which have been used to differentiate objects. However, ML and DPI methods present limitations namely for timely classification of encrypted Internet traffic. In this chapter, we investigate the use of statistical methods published in the literature that have proven successful for classification in other areas but have not yet been tested for network traffic classification. Thus, we propose, implement and evaluate a classifier based on Jensen-Shannon, Hellinger, Bhattacharyya, and Wootters methods to classify encrypted Internet traffic.
Furthermore, we present a qualitative comparative analysis of the tested methods based on their Kappa values (following the definition of Landis, Koch and McHugh) and their Receiver Operating Characteristic (ROC) curves (via the Areas Under the Curves (AUCs)). We present the Accuracy of each implemented classifier, obtaining average values above 90% for the Jensen-Shannon, Bhattacharyya, Hellinger and Wootters methods. Our study also includes a comparison of the computational costs of the evaluated methods. Thus, these methods may be used as alternatives for the classification of encrypted Internet traffic.

5.1 Introduction

In the information society, the movement of data packets in a network has become as relevant as the movement of people and tangible goods in the physical world. Therefore, the monitoring of network traffic and the understanding of traffic patterns have been attractive research areas for decades [1].

Nowadays, people use the Internet for supporting their daily activities, including communications, transactions and entertainment. Communications include email and instant messaging applications, transactions include e-commerce and online banking, and entertainment includes downloading movies and listening to Internet radio.

Through these kinds of operations, a user can provide some personal information, making

[1] This chapter consists of a paper under review at the time of writing this Ph.D. thesis.

it possible to know who he/she communicates with and what his/her interests are [221]. File sharing is one of the biggest sources of congestion on the Internet. Programs such as BitTorrent [222] transform user machines into nodes in an overlay network, using their connections to provide data to other users.

Access to sites such as YouTube and Spotify provides video and audio content to users on demand, generating streaming media traffic [223]. Videoconferencing can also cause congestion in end-user connections since several users connect simultaneously with video and audio, resulting in many streams in each connection. Therefore, it is important to learn how to identify [221] and classify traffic to maintain available bandwidth, prevent congestion and avert security issues [224].

Classifying traffic is a way of identifying the application or protocol used; this involves methods and techniques that go beyond simple traffic analyses for separating flows. The process of classifying traffic observes features and behaviors in traffic and checks for conformation to the assigned traffic type. Faced with a growing network traffic scenario [225, 226], many studies have been carried out within the scope of traffic classification [5, 122, 217, 227–230].

Classification can be performed by using the number of ports [231], payload [232], Machine Learning (ML) [5], Deep Packet Inspection (DPI) and Heuristic [229] in both supervised [50, 122, 228–230, 232], and unsupervised (also known as clustering [217, 228]) manners. These methods may operate online such as in [50, 225, 229] or offline such as in [233].

These methods involve the extraction of traffic features. For these features, it is quite common to use traffic patterns, jitter, latency, delays between packets, packet size, payload identification, and flow source/destination identification, in combination with packet size fragments and traffic flow resources such as packet sizes, the time between packets and information derived from traffic flows [233].

The method of associating transport port numbers with known application protocols is not efficient. Thus, several applications have started using port numbers employed by other known protocols [232] or random port numbers.

DPI is also not efficient, as it requires many computational resources due to the payload inspection process and the impossibility of analysis when payload encryption is utilized [50, 232].

Methods associated with ML have limitations and require an initial data training step. The Support Vector Machine (SVM), for example, is supervised and uses different kernels to comprehend the classification data and then finds an ideal limit among the possible

outputs. Algorithms such as Bayesian estimation, C4.5, and nearest-neighbor estimation may be tied to local optimization and cannot work in real time due to their computational and storage requirements [50].

Traffic classification methods based on flow statistics, on the other hand, offer varying degrees of success, including in real-time situations, and do not require high computational resource usage. Methods based on statistical analysis are also able to process encrypted traffic [122, 233].

Several such methods have been proposed, making statistical use of the packets' characteristics, including their flow signatures, through the size of the payload by referring to applications and application layer protocols.

Through this work, the results will be presented in the following sequence: Firstly, classifying traffic with statistical distances and divergences to obtain Precision, Recall and F-Measure values. Secondly, after getting the classification values, we calculate ROC values to obtain which is the best classification method for a specific kind of traffic.

This article's next sections are organized in: Section 5.2 presents the existing algorithms and approaches applied to network classification. Section 5.3 details the methodology of our approach and the design of our solution. We provide some experimental results in Section 5.5. Section 5.6 is the conclusion of the paper.

5.2 Related Work

In this section, we provide a review of traffic classification techniques and statistical methods, focusing on the methods that are explored in this work for classification purposes.

5.2.1 Traffic Classification

Lots of efforts have been made to research Internet traffic identification and classification. It was possible to see that some researches on the topic of Internet Traffic identification and classification were conducted by exploring network statistics such as the size of packages, flow quantity, jitter, latency or packet inter-arrival time. e.g., [69, 215, 234–236].

The works of [234–236] as an example, used traffic flows or size of packages to classify the traffic. The Table 5.1 represents a summary of related works, the types of traffic addressed, whether traffic classification occurs in real time or not, and the performance achieved.
It is important to mention that calculating traffic flow statistics is a strategy to identify the traffic in real time, because these statistics can be used in the classification model and by using such statistical information it is possible to determine the ideal model to be used for that classification [237].

Table 5.1: Summary of the Most Relevant Related Works. P:Precision, R:Recall, A:Accuracy, FM:F-Measure.

Work	Method	Type of Traffic	Real-time operation	Detection of encrypted traffic	Performance(%)
Holanda Filho *et al.* [235], 2008	Statistical discriminators, cluster analysis and k-means algorithm	P2P	No	Yes	A:16,87, 98,93 R:76.65, 100
Neto *et al.* [69], 2013	Kolmogorov-Smirnov test and Chi-square test	Web,HTTP download, live streaming, streaming on demand, P2P video streaming, P2P file sharing, VoIP, FTP, SFTP, SSH	Yes	Yes	P:77.78, 100 R:76.65, 100
Tongaonkar *et al.* [215], 2015	Automated signature generation based on the packet payload content	Web, SMTP, BitTorrent, DNS	Yes	Unable	P:97
Li *et al.* [216], 2015	Fast multitask sparse feature learning method using a non convex capped l1				
Shaikh *et al.* [236], 2015	Flow-level statistical properties, nearest clusters and k-means algorithm	HTTP	No	No	-
Raveendran and Menon [238], 2016	HNB and KStar (K*) lazy classifier	BitTorrent, DNS, FTP. HTTP, SMTP, Yahoomsg, SSH		Yes	P: 93.2, 100 R: 91.8, 100
Peng *et al.* [217], 2017	IDGC	FTP, POP3, SSH, Telnet, edonkey, Skype, Cloud Disk	Yes	-	P: 99.98
Ertam and Avci [219], 2017	WK-ELM and GA-WK-ELM	WWW, Email, bulk, attack, P2P, database, services			A:95
Schmidt *et al.* [66], 2017	Artificial immune system-inspired classification	FTP, Databases, LDAP, BitTorrent, rlogin, DNS POP2/3, SSH, telnet, SMTP, X11 klogin, NTP, WWW, KaZaA IMAP, attacks, games, WMP			A:95
Shi *et al.* [65], 2017	SVM using multifractal features extracted by wavelet leaders multifractal formalism and PCA	SMTP, IM, HTTP + Flash, WWW VoIP and IMAP, P2P, IM and POP	Yes		A:74.67, 100
Wang *et al.* [213], 2017	One-dimensional convolution neural network	VPN-Chat, Chat, VPN-file, file VPN-Email,Email, VPN-P2P, P2P VPN-streaming, streaming, VPN-VoIP, VoIP		Yes	P:78.2, 99.9 R: 81.3, 100
Sun *et al.* [5], 2018	ISVMs and an ISVM model with an attenuation factor	WWW, mail, FTP, attack, P2P, database, multimedia, services, interactive	Yes	Yes	A:91 FM: 97-98
Aceto *et al.* [29], 2019	Deep learning & Mobile			Yes	A: $\geq$ 90
Dias *et al.* [89], 2019	Gaussian Naive Bayes & Video streaming		Yes		A: 90 P:66.28
Tan*et al.* [239], 2019	Markov model, hidden Gaussian mixture model and deep neural network	HTTPS, social applications, multimedia and game clients	Yes	Yes	A:92, 96
Chari *et al.* [234], 2019	Packet length signatures/ decision tree	Audio and video streaming, browsing P2P, FTP and VOIP	Yes	Yes	A: 92, 96
Seddigh *et al.* [240], 2019	Logistic regression, SVMs, decision trees, Adaboost, neural networks and naive Bayes classifiers	YouTube, Netflix, Skype, Messenger, Spotify, SoundCloud, Dropbox, Google Drive, Gmail, Yahoo, Firefox, Chrome, BitTorrent, eDonkey, Facebook, Telegram, Video streaming, Web browsing	Yes	Yes	A: 93.14, 91.61 R:88 P:88
Labayen *et al.* [241], 2020	Balanced Iterative Reduction Clustering using Hierarchies (BIRCH), Gaussian Mixtures (GMs) and K-means SVMs, RFs and NNs	Remote shell session, SCP/FTP Web browsing, Twitch and YouTube	Yes	Unable	A: 97.37

In terms of protocol identification in application layer level, the author [65] approaches techniques that use statistical characteristics of Transport Layer Security (TLS) and ML techniques both for specific traffic and for conventional traffic.

The approaches used in most of the works presented here refer to methods that use ML. Through the studies it was possible to see that the statistical approaches are often used to create traffic signatures and foresee behaviors. From that information new techniques are proposed to classify traffic.

Neto*et al.* [69] described a traffic classification approach based on signature correspondence. Signatures are empirical distributions that represent the applications. To develop the traffic classifier the authors used Kolmogorov-Smirnov e Chi-square statistical tests. A network traffic classifier based on Packet Payload Content (PPC) was presented by Tongaonkar*et al.* [215]. The classifier is capable of learning new signatures from the applications being classified.

Chari*et al.* [234] also developed a classifier based on the extraction of package length signature to classify different classes of traffic such as audio streaming, video streaming, navigation, chad and Peer to Peer traffic (P2P).

Li*et al.* [216] affirmed that to extract statistical resources it is necessary to have previous knowledge about network traffic. Usually, resources are considered alike in ML algorithms, like SVMs or decision trees C4.5, and many use insufficient resources or methods to select/conduct primary resources. However, threshold-based algorithms (i.e., the ISTMTFL algorithm) and iterative shrinkage have been optimized by this perspective to solve multitask feature learning problems, and the model proposed was applied to learn common characteristics among many tasks of traffic classification.

Sun*et al.* [53] explored the behavior and statistical characteristics of traffic flows through ML methods to solve classification problems. The authors Implemented an Incremental SVMs model (ISVMs) with the purpose to minimize memory and CPU cost when classifying traffic in a rapid way. Besides that, with the ends to use the information from within the series of training data, a new model of the ISVM, called AISVM with an Attenuation factor was developed.

ML approaches were also used by Ertam*et al.* [219] who developed the Extreme Learning Machine method (ELM) to classify Internet traffic. On the input data, the ELM method based on Kernel (KELM) was applied. Besides that, the Generic Algorithm (GA) was used to select the parameters and later on, a software was developed based on Wavelet Kernel (GA-WK-ELM), in which the Wavelet function was used.

The Gravitation-based Classification of Data (DGC) is a classification model for the treatment of unbalanced data sets, and Imbalanced DGC (IDGC) was proposed in [217] to solve identification problems with unbalanced Internet traffic. For the implementation of the prototype, the authors also extracted the initial characteristics of Internet traffic according to the observed packet sizes. The authors stated that identifying traffic is a normal problem when classifying. Therefore, supervised learning techniques have been used to solve it.

The focus of the article in [66] was to use statistical resources of the given network flows on identifying the application generated. The authors reported that an SVM's performance is directly dependent to the kernel function used and its parameters; they also expanded the research field by introducing several optimizations to the Artificial Immune Systems (AIS) algorithm in its training and classification phases with respect to Internet traffic flows. They further applied the algorithm to a data set and found that the algorithm performed very well, making it valuable for embedded systems.

Other techniques have also been proposed for real-time traffic classification, such as that in [65], which analyzed the multifractal characteristic differences between different classes of traffic, provided the reasons for these differences and proposed a method of extracting multifractal characteristics that was based on wavelet leaders and Multifractal Formalism (WLMF). It is a robust method that can extract multifractal resources that are more efficient to classify traffic when compared to TLS. In addition, the authors developed a resource selection approach based on the analysis of main components (PCABFS) to select resources. Through the PCABFS method, it was possible to evaluate the impact and the validity of the explored resources.

The combination of Hidden Naive Bayes (HNB) and KStar (K*) classifiers were proposed by [238] with the purpose to explore traffic classification in real time. The authors used the Correlation-based Selection and discretization method (CFS) and Entropy-based Minimum Description Length (ENT-MDL) techniques in the data preprocessing stage. The author [193] also used the entropy technique to classify P2P traffic, extracting network resources like the package size and resources from the header in the transport layer, using the sliding window. The authors in [241] proposed a system to classify traffic by using a hybrid method with a Random Forest (RF) and K-means.

5.3 Statistical Methods

Statistical methods usually depend on priorities from network flow such as interval between packages, size and duration of a flow, size and length of the packages in the flow [242,243]. To classify using statistical methods, the flow properties can be grouped or used individu-

ally. When grouped, values are generated like average or variance or more complex measures such as calculating the function for probability density and package size frequencies in each flow [85].

Distance and divergence calculations are advanced statistical analysis methods that can be used for classification and, in our context, were used for Internet traffic classification. Through these properties, statistical traffic models can be created for a given application. For those techniques, it is necessary to develop a reference model of flow, protocol or application that can be used in a training phase, for example to identify unexplored traffic [244, 245].

The distances between populations can be interpreted as the distances between two probability distributions; therefore, they essentially measure the distances between measures of probability. A distance is the numerical calculation obtained between two classes or objects that presents the degree of difference between them [246].

In some situations, distances are considered a special class of divergence [19]. This situation occurs when the measure satisfies 3 properties, namely, positivity, symmetry and triangular difference [247]. Positivity occurs when the distance from a class to itself is equal to 0. Symmetry occurs when the distance between compared classes I and II is the same as that distance between II and I. However, some divergences do not satisfy this property [245].

Based on these properties, we selected the Jensen-Shannon Divergence (JSD) for our study, as as it is Kullback-Leibler (KL) Divergence based with some variances (such as symmetry and the use of values that are always finite), Hellinger Distance (which is also taken as a measure of statistical divergence), the Bhattacharyya Distance, and the Wootters Distance (which is characterized by finding the differences in the probabilities below the values of the typical fluctuations). For comparative study, we chose the Kolmogorov-Smirnov (KS), Chi-square, Euclidean and KL Distances.

The Chi-square difference [246] shows a relationship between two categorical variables and indicates the difference between the observed and expected counts; in other words, it can be used to compare the expected values with those actually collected.

The KL Divergence also known as relative entropy is the result of the calculation of the distance between two probability distributions [148–150, 150, 248].

Proposed by the Greek mathematician Euclid, the Euclidean Distance is the result of the calculation of the distance $D_E[x, y]$ between points x e y in an Euclidean plan. The traffic classification results using Euclidean Distances and KL were presented in [72, 249].

5.3.1 Jensen-Shannon Divergence

Jensen-Shannon Divergence (JSD) is known as limited symmetrization of the KL Divergence [19]. JSD is the divergence between groups of probability values P and Q [139]. For 2 discrete probability distributions P and Q, JSD divergence is given as follows: with $pi \geq 0$, $qi \geq 0$.

$$JSD\left(P,Q\right) = \frac{1}{2}\left\{\sum_{i=1}^{N} pi \log\left(\frac{2pi}{pi+qi}\right) + \sum_{i=1}^{N} qi \log\left(\frac{2qi}{pi+qi}\right)\right\} \quad (5.1)$$

Where N represents the number of samples, i represents the initial samples, pi and qi represent the relative frequency of known and unknown protocols respectively.

When the values of the calculation between distributions P and Q have 0 as a result or something close to it, it means that the distributions are the same, which also means that the applications are the same.

5.3.2 Hellinger Distance

Hellinger distance has the purpose to determine the similarity of the values of P and Q probability distributions.
Hellinger Distance has been applied to solve several problems of statistical estimation. In our it was used to calculate the distance between to relative frequencies of the protocols and applications [144, 145, 250]. Hellinger statistical distance is given by [18] eq. 5.2, defined by:

$$H_D\left(P,Q\right) = \sqrt{\frac{1}{2}\sum_{i=1}^{N}\left(\sqrt{pi} - \quad \sqrt{qi}\right)^2} \quad (5.2)$$

where for Hellinger N represents the number of samples, i represents the initial number of samples, pi and qi represent the relative frequency of the known and unknown protocols, respectively.

5.3.3 Bhattacharyya Distance

Bhattacharyya Distance uses Bhattacharyya coefficient that measures how alike to samples are. The name was given in honor of the statistician Anil Kumar Bhattacharyya [141,142]. Bhattacharyya Distance is independent of the distribution function and because of that, can be used in any sample group. It is a special distance, because it can be used to update models in which the distribution cannot be defined by exact numbers [141].

The Bhattacharyya statistical Distance is given by [18] eq. 5.3:

$$Bc_D(P,Q) = \quad -\log\left(\sum_{i=1}^{N}\sqrt{pi \quad \times \quad qi}\right) \quad (5.3)$$

where N is the quantity of samples, P, Q, pi, and qi represent the sample members in the i-th partition, respectively.

5.3.4 Wootters Distance

Presented by Wootters in 1981, this distance is characterized by finding the differences in probabilities that are below the values of typical fluctuations. Considering the probabilities of two classes of traffic (p and q), the minimum distance between two points is equivalent to the angle they present; this is represented by eq. 2.9 [18].

Wootters statistical Distance was used to detect data with duplicate characteristics [154]. The similarity between P and Q samples is measures Wootters Distance according to the distance calculation between them. Where pi and qi represent the relative frequencies of the applications. Wootters is given by [21] eq. 5.4:

$$Wo_D(P,Q) = \arccos\left(\sum_{i=1}^{N}\sqrt{pi \quad \times \quad qi}\right) \quad (5.4)$$

Note that $arccos()$ decreases in [0, 1]. Note also that distances are used to discriminate traffic; in addition to distances, to make a more accurate classification, it is necessary to use heuristics.

5.4 Proposed Approach and Implementation

To form a classifier based on traffic behavior, a traffic distribution sample is required so that the algorithm can use a sample of cases for which the true classifications are known. Each set of attributes (features) describes a case. To distinguish cases among possible classifications, each case is labeled with a special attribute, called a class, whose values refer to the true classifications of cases.

This section explains the characteristics used to design the traffic flow of each protocol/application and how the statistical distance methods were used for traffic classification.

5.4.1 Dataset

The data set used to perform the classification tests was the one used in [72], and available in [251]. With the help of the Wireshark tool, traffic traces were captured. Those traces

were stored in ".pcap" files. Each traffic trace contains a data flow. Each flow is formed by a bidirectional tuple composed by transport protocol and origin and destination IPs. In total, 28 types of applications were captured and included in the "file.pcap".

Several varieties of protocols and applications were selected to compose the database. In this selection, many applications widely used were included, such as HTTP for web applications, VoIP, SSH besides BitTorrent, Gnutella and eDonkey.

For our work, the captured traffic traces were denominated as the original base and the base containing Relative Frequencies (RF) was denominated as the new database. A dataset was formed by traffic collected through the TCPDUMP and WINDUMP tools. Part of the dataset was used to determine the ground truth.

5.4.2 Signature Traffic and Features

From the original trace database, it was possible to create a new data group. This new group contains important and relevant information, being able to present fewer attributes when compared to the original base, bringing us the benefit of a smaller dimensionality. They are features that can be added as input to the classification.

With the purpose to generate a new data set, we extracted from our original database some statistical properties from the flows (origin and destination IPs), being them the total length, the number of all packets, origin and destination IPs and inter packet arrival time.
Those properties that were extracted from the flows were vital for the relative frequency calculation of the flows. The absolute and relative frequencies calculation were performed according to the following equation:

$$fr_i = \frac{f_i}{n}, \quad (5.5)$$

where fr_i represents the relative frequency, f_i represents the absolute frequency (observed value and n represents the total number of elements in the sample). Each flow contains packages varying from 64 to 1514 in size. See that intending to create a histogram with intervals of 16 bytes, we defined 100 buckets.

We believe that the packet length distribution is of great importance for characterizing traffic because each protocol is presented in a singular way, forming a signature. To calculate the distributions, a Python script was created; this script took individual trace files as inputs and generated files with relative frequencies as outputs.

The relative frequency of the package length was generated by application. To generate the

relative frequency, we observed that each flow represented a temporal series. The Figures 5.1, 5.2, 5.3 and 5.4 illustrate the distribution of relative frequency of packages created from our new database. Those graphs show some available applications in our new data set. It was also possible to observe that each protocol presents a unique distribution and behaviour, which is called signature. We applied those unique distributions to calculate the divergences and start the process of classification. See that we used this behaviour to measure the known divergences of distribution and to start the classification process. According to the Figures 5.1, 5.2, 5.3 and 5.4 we have the number of buckets represented by the Y axis, and the relative frequency is represented by the X axis.

According to Figures 5.1, 5.2, 5.3 and 5.4 each application has a distribution. We can see that for applications such as P2P, which conduct transmission with varying packet lengths, the graph lines do not show such high frequency peaks. For the other applications, such as HTTP, which has larger packet lengths and a constant size transmission size, the graph line presents greater relative frequency peaks.

5.4.3 Prototype for Classification

The proposed classifier combines the features described in 5.4.1, and 5.4.2, and with the support of Figure 5.5, we describe the classification mechanism and its operation. For the implementation of the distances, an application was developed in Python [252]. The test mechanism was divided into 3 steps: separating flows and creating signatures, choosing the statistical distribution/methods to be used, and making the classification decision. The ROC curve presents the successes and refinement of the model. The ROC curve is described in the results analysis section.

Figure 5.5 shows the architecture adopted for the classification process, where the flows were preprocessed, and the distribution of the packet length was extracted according to flows, forming a new database called the ”Generate RF base”. This new base served as input for the statistical methods, where 30% of the base samples were used for individual RFs (generating the signatures to be used by the statistical methods) and the other 70% of the samples were used for the test set (formed by the database).

For classification, it was necessary to define the distribution to be used (Jensen-Shannon, Hellinger, Bhattacharyya or Wootters). The divergences/distances among the probability distributions were calculated by ”distance calculations” according to the utilized method. The outputs were the values of the distances between the distributions.
The rules were applied for executing classification decisions after these procedures, and the classification outputs and the comparisons between methods (via ROC curves) were obtained.

Separating flows and creating signatures: The Python application was developed for this

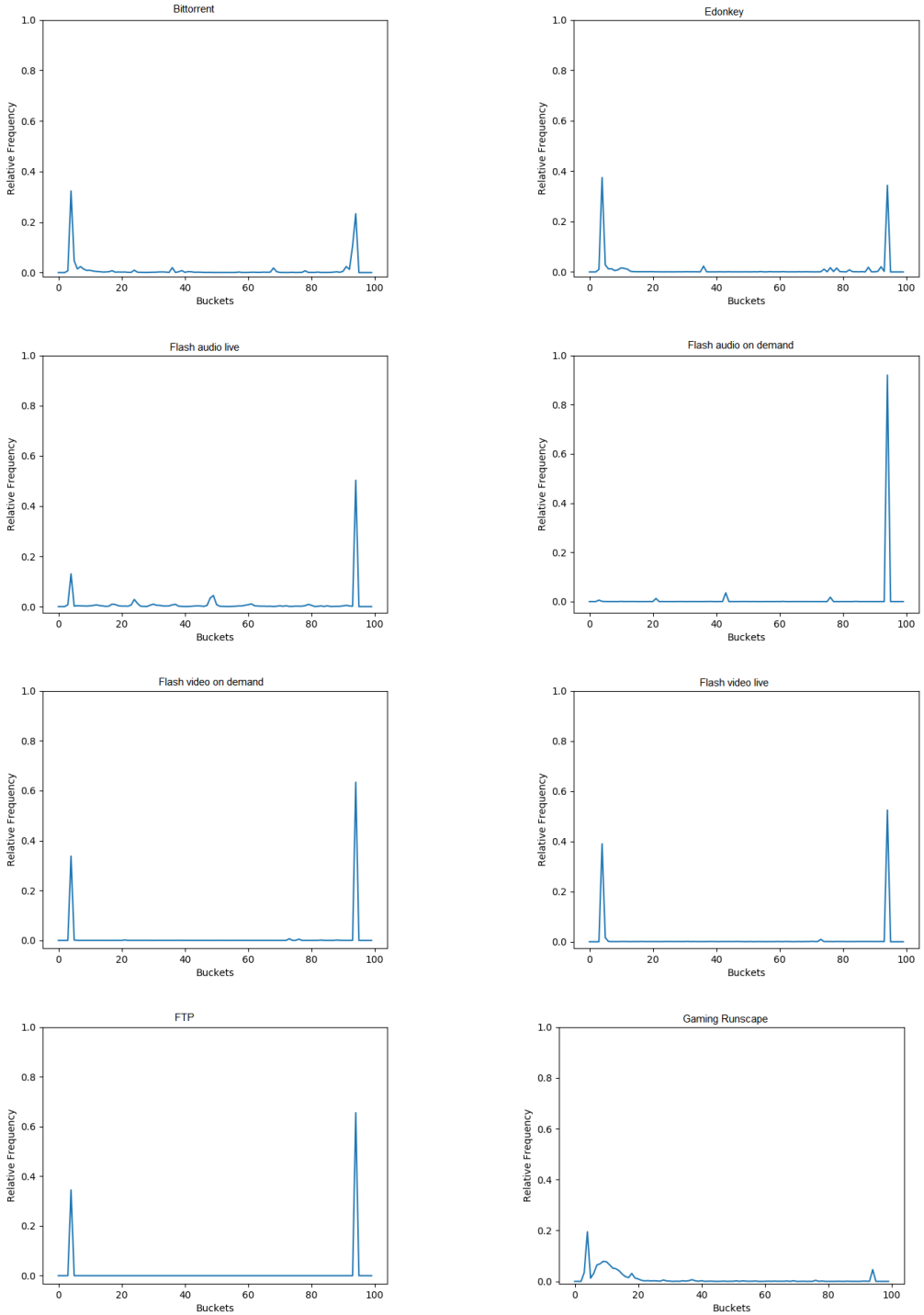


Figure 5.1: Relative frequency distributions of the packet lengths calculated for bittorrent, edonkey, flash audio live flash audio on demand, flash video on demand, gaming war of legends, ftp, gaming run scape.

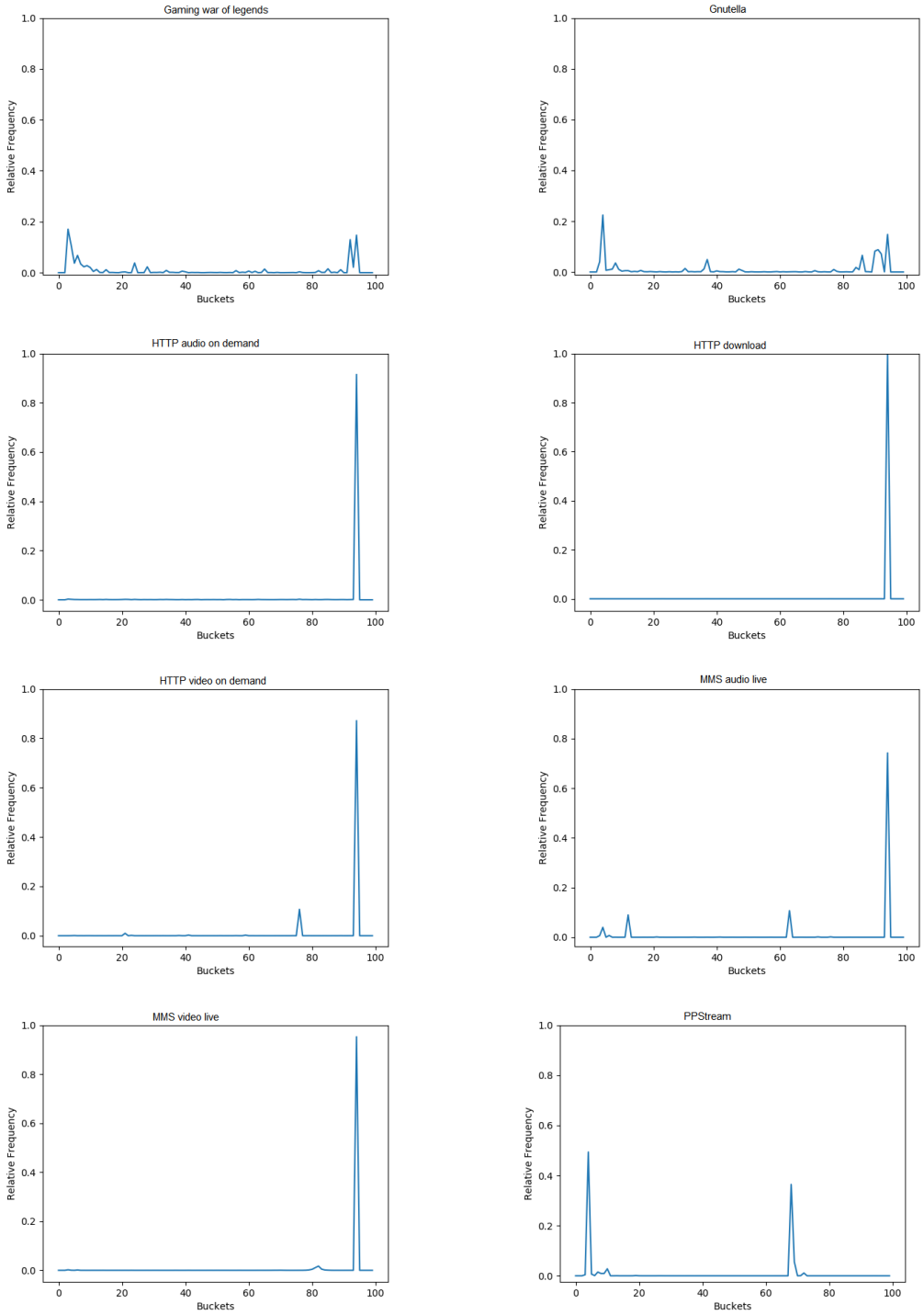


Figure 5.2: Relative frequency distributions of the packet lengths calculated for gaming war of legends, gnutella, http audio on demand, http download, http video on demand, mms audio live, mms video live, ppstream.

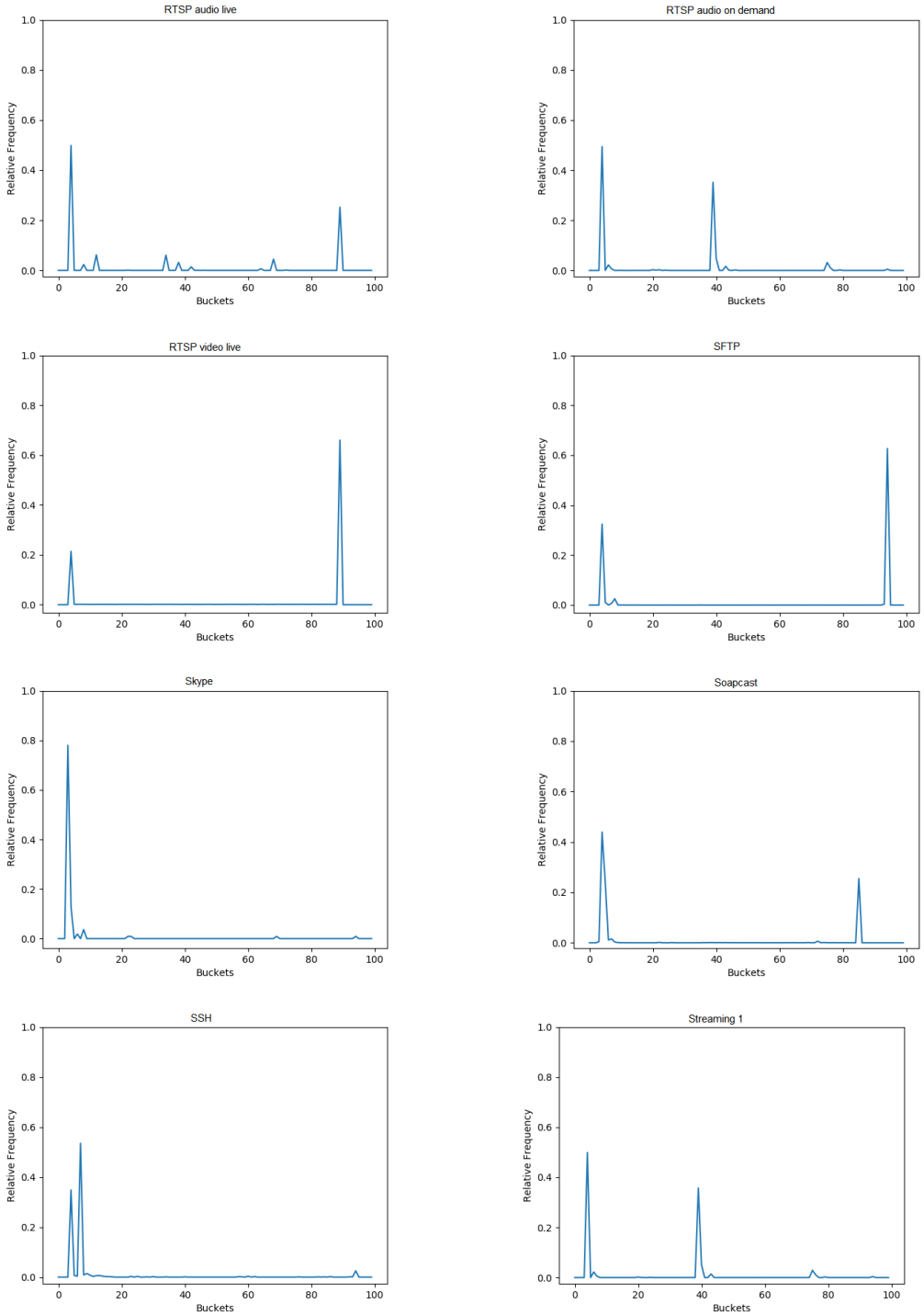


Figure 5.3: Relative frequency distributions of the packet lengths calculated for rtsp audio live, rtsp audio on demand, rtsp video live, stfp, skype, sopcast, ssh, streaming 1.

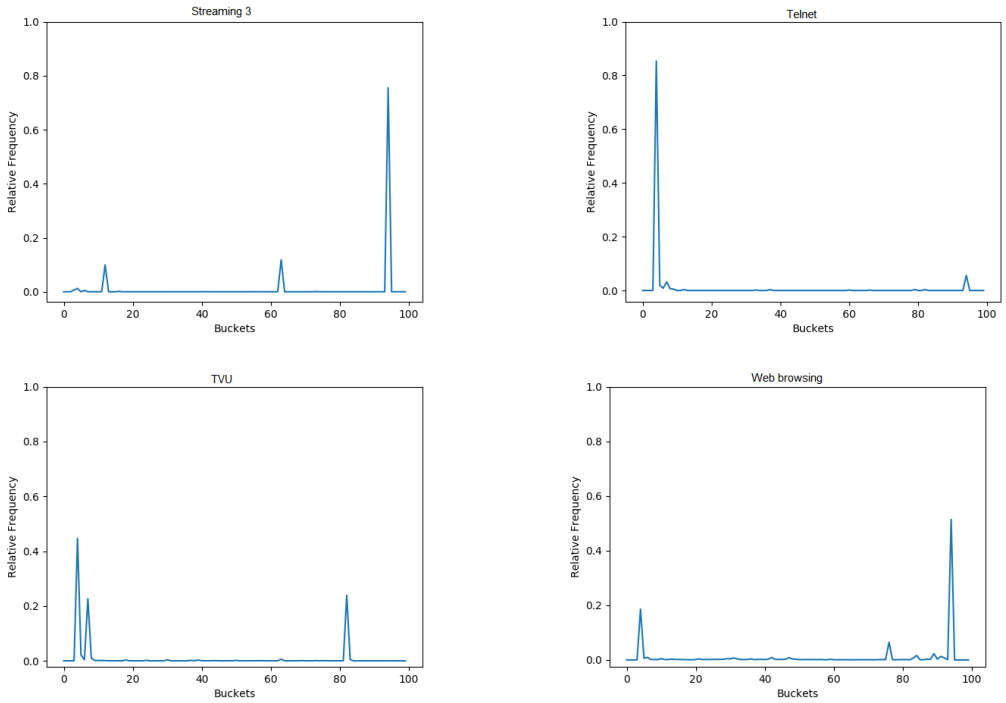


Figure 5.4: Relative frequency distributions of the packet lengths calculated for streaming 3, telnet, tvu, and webbrowsing.

step. In the application, the individual distributions were called the pi samples and the collective distributions of the qi samples. This step is best described in Section 5.4.2.

Choosing the statistical distribution/methods: For the calculation of the distances, we chose the Jensen-Shannon, Hellinger, Bhattacharyya and Wootters methods and compared the relative frequencies (pi and qi) generated.

The distance values were obtained after comparison between samples pi and qi. See that we considered sample pi as individual relative frequencies and sample qi as collective relative frequencies.

Classification decision: This part was the most complicated step since it was insufficient to obtain the values of the distances through the statistical methods for classification purposes. These distances alone do not represent anything; therefore, it was necessary to make use of heuristics in addition to the rules.

Classification rules are necessary so that the classifier can make decisions based on distance calculations. According to distance rules, the closer to 0 the distance values are, the more alike the protocols will be and the closer those values are from 1, the less similar they will be.

After several iterations and basing on statistical heuristic, classification rules we created.

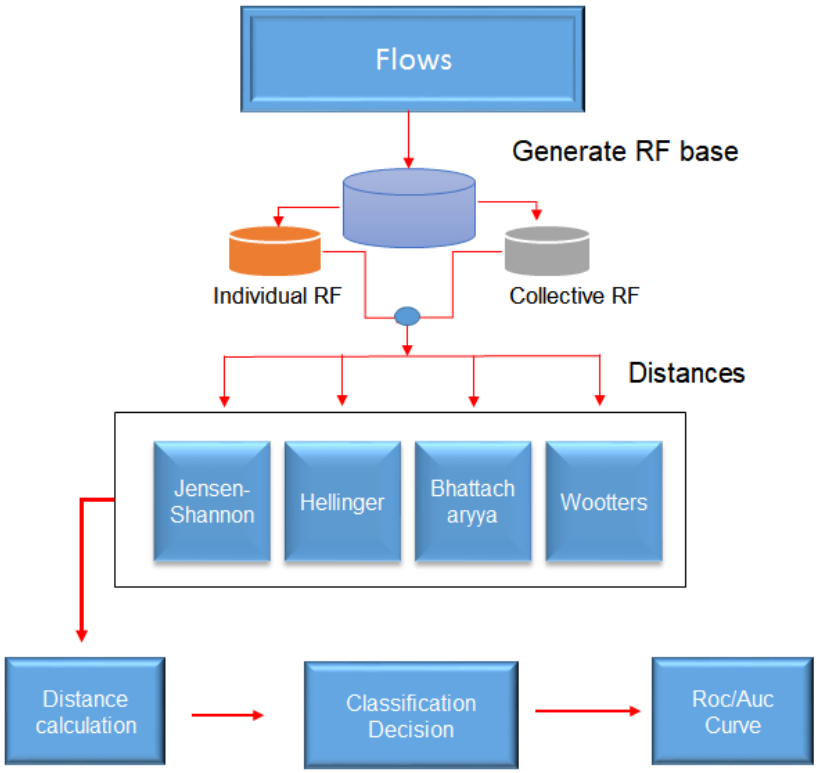


Figure 5.5: Architecture implemented for traffic classification using distances and divergences.

The rules consist on selecting the minimal distance and defining the base values for acceptance. In our case, we defined five different minimal ranges as base. The minimal distance is predicted to be between 0 and the current loop range for each value range.

Wrapping our classification process up, two variables were created MaxR, MaxMatrix. The first one stores the minimal range with the best result and the second stores its confusion matrix. Besides that, inside the classification process, it was defined maxFmeasure, a variable that stores the best F-Measure.

5.5 Results and Discussion

5.5.1 Performance Metrics

The results provided by the classifiers based on statistical methods were compared to the ground truth information to compute the number of True Negative (TN), True Positive (TP), False Negative (FN) and False Positive (FP) cases. Based on these metrics, we evaluate the performance of the classifiers using Accuracy, Recall, Precision and F-Measure defined by eqs. 5.6, 5.7, 5.8, and 5.9 [22]:

$$Accuracy = \frac{TN + TP}{FP + TN + FN + TP}, \quad (5.6)$$

$$Recall = \frac{TP}{FN + TP}, \quad (5.7)$$

$$Precision = \frac{TP}{FP + TP}, \quad (5.8)$$

$$F - Measure = \frac{Precision * Recall * 2}{Recall + Precision}. \quad (5.9)$$

Kappa values were used for qualitative valuation. Kappa index is a statistical method to evaluate the agreement or reproduction level among the set of classifiers. The higher the accuracy, the higher the kappa index. Kappa values were calculated according to the equation 5.10 as [23, 24]:

$$K = \frac{P_0 - P_e}{1 - P_e}. \quad (5.10)$$

Being K = Kappa index, P_0 relative acceptance rate, P_e hypothetical acceptance rate. To estimate P_0, the concordance sum is divided (TP e TN) by the total quantity of items in the sample, that represents exactly the accuracy, given by the equation 5.11:

$$P_0 = \frac{(TP + TN)}{(FP + TN + FN + TP)}. \quad (5.11)$$

To estimate P_e it is necessary to calculate the probability of both randomly accepting or rejecting a data classification. For that we have the following equation 5.12:

$$P_e = \frac{\frac{(((TP+FP)*(TP+FN))+((FN+TN)*(FP+TN))}{(FP+TN+FN+TP)}}{(FP + TN + FN + TP)}. \quad (5.12)$$

We used two different interpretations: (1) Landis and Koch [24], and (2) McHugh [25]. The scales are not all overlapped, but they suggested a similar interpretation of the results. For Landis and Koch [24], Kappa values above 0.41 are already considered moderated, while for McHugh [25] Kappa values need to reach at least 0.60 to be considered moderated.

For Landis and Koch [24] Kappa values over 0.80 can already be considered with acceptance strength almost perfect, while for McHugh [25] only values above 0.90 reach this

level of acceptance strength.

The scales of Landis and Koch [24], and McHugh [25] are important references to identify the strength and acceptance level of a set of classifiers, showing the quality of the techniques used to classify encrypted and obfuscated internet traffic. Table 5.2 presents Kappa values interpretation suggested by Landis and Koch [24] and Table 5.3 presents Kappa values interpretation suggested by McHugh [25].

Table 5.2: Kappa values interpretation according to Landis and Koch [24].

Kappa statistic	Strength of Agreement
> 0.81	Almost Perfect
Between $0.61 - 0.80$	Substantial
Between $0.41 - 0.60$	Moderate
Between $0.21 - 0.40$	Fair
Between $0.00 - 0.20$	Slight
< 0.00	Poor

Table 5.3: Kappa values interpretation according to author McHugh [25].

Value of Kappa	Level of Agreement
> 0.90	Almost Perfect
Between $0.80 - 0.90$	Strong
Between $0.60 - 0.79$	Moderate
Between $0.40 - 0.59$	Weak
Between $0.21 - 0.39$	Minimal
Between $0.00 - 0.20$	None

5.5.2 Classification Results

This section provides an evaluation of the classification methods for the data set under study, reported in section 5.4.1. Table 5.4 shows the average and standard deviation of Accuracy, Precision, Recall, and F-Measure obtained by the classifiers based on KS test, Euclidean Distance, KL Divergence, Wootters Distance, Jensen-Shannon Divergence (JSD), Chi-square test, Bhattacharyya Distance, and Hellinger Distance. The average corresponds to the linear average of the results obtained by each classifier for the 28 applications.

The standard deviation is necessary to show the variations in the average values across applications. Figures 5.6, 5.7, 5.8, 5.9 and 5.10 shows detailed classification results per application, obtained with the above distances/divergences for the set of 28 applications considered in the data set.

The lowest average values and the highest standard deviation values of the classification results were obtained by the Kolmogorov-Smirnov (KS) method. We observed that the highest average performance was obtained by the Hellinger method, which, in most cases,

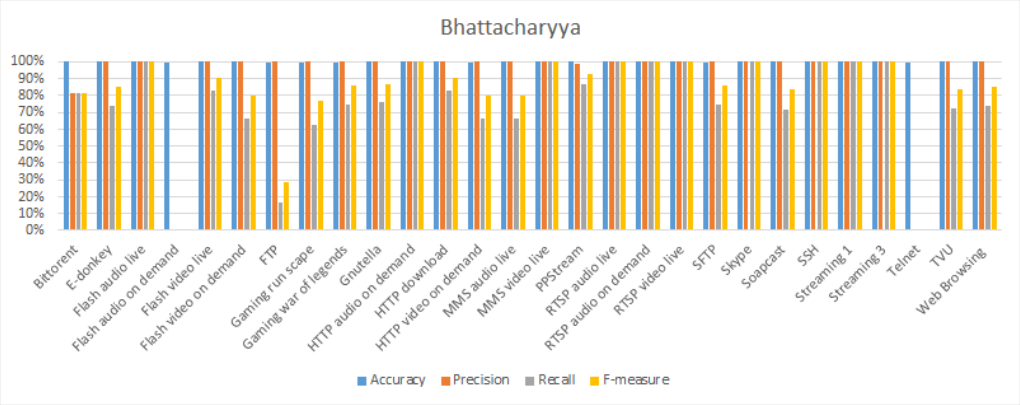


Figure 5.6: Classification results obtained using the Bhattacharyya Distance.

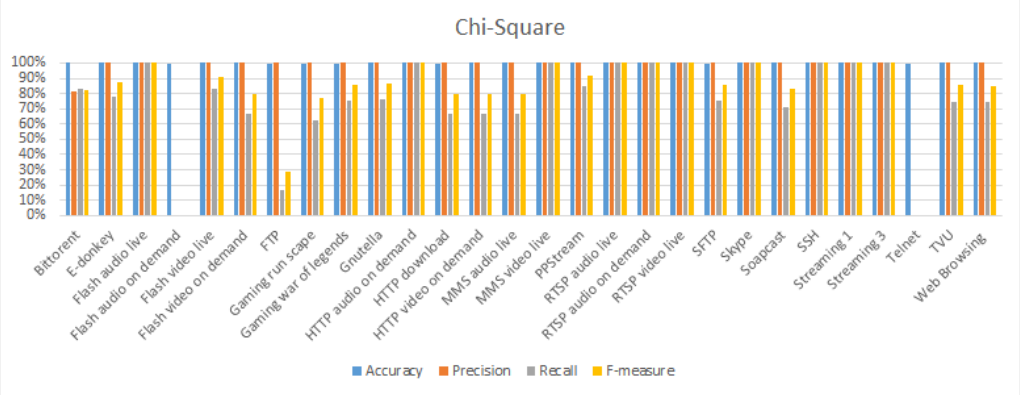


Figure 5.7: Classification results obtained using the Chi-square test.

Table 5.4: Classification Results Obtained with Statistical Methods.

	Accuracy		**Precision**		**Recall**		**F-Measure**	
Classifier	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation
KS	0.99930	0.00062	0.80623	0.38551	0.48663	0.29667	0.58979	0.31884
Euclidean	0.99967	0.00045	0.92188	0.26276	0.72309	0.27342	0.79668	0.26601
KL	0.99967	0.00049	0.92170	0.26284	0.73255	0.28943	0.79920	0.27535
Wootters	0.99967	0.00049	0.92222	0.26263	0.73646	0.28974	0.80191	0.27575
Jensen-Shannon	0.99969	0.00047	0.91442	0.26317	0.75649	0.28259	0.81441	0.27165
Chi-Square	0.99970	0.00046	0.92200	0.26271	0.75737	0.26271	0.81768	0.27121
Bhattacharyya	0.99971	0.00045	0.92160	0.26265	0.76178	0.28261	0.82043	0.27164
Hellinger	0.99971	0.00045	0.92280	0.26241	0.76293	0.28259	0.82175	0.27181

presented higher F-Measure and Precision values than those of the other approaches. The results presented make us conclude that statistical classifiers are considered good traffic discriminators for our scenario because with the exception of the KS method, all approaches obtained Precision averages above 90%.

This means that 90% of the time, these classifiers were able to predict true instances as actually being true and produce low FP values. We can conclude that, for our scenario, the statistical classifiers correctly rejected the classified samples and concluded that they

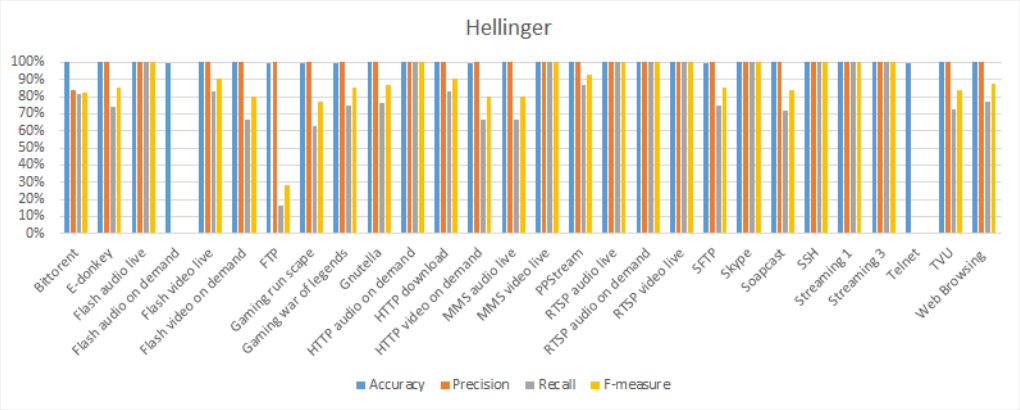


Figure 5.8: Classification results obtained using the Hellinger Distance.

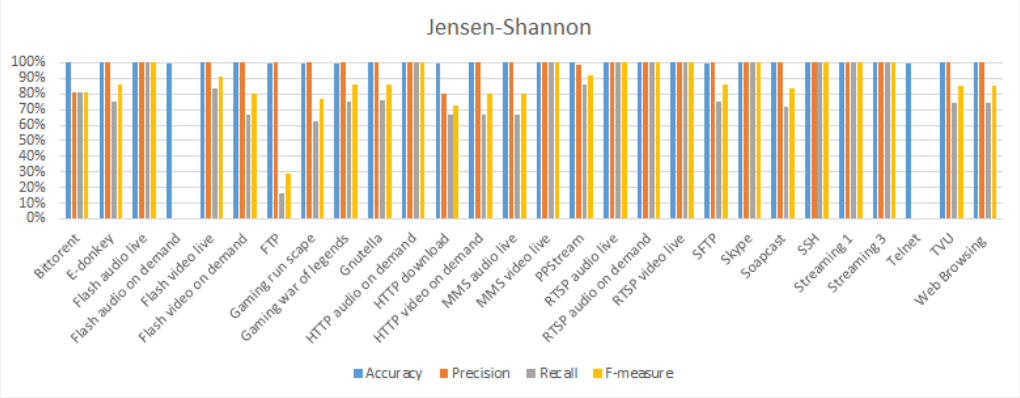


Figure 5.9: Classification results obtained using the JSD.

were indeed not part of the class.

Table 5.5: Classification Quality Associated with the Kappa Statistic Values.

Classifier	Kappa	Qualitative evaluation of the classifier	
		Landis and Koch	McHugh
KS	0.72278	substantial	moderate
Euclidian	0.82447	almost perfect reliability	strong
KL	0.83213	almost perfect reliability	strong
Jensen-Shannon	0.84363	almost perfect reliability	strong
Wootters	0.84371	almost perfect reliability	strong
Bhattacharyya	0.84540	almost perfect reliability	strong
Chi-square	0.84910	almost perfect reliability	strong
Hellinger	0.85225	almost perfect reliability	strong

Table 5.5 presents the achieved Kappa values and qualitative assessments according to Landis and Koch [24] and McHugh [25]. For this evaluation, we observed that the lowest Kappa value was obtained by the KS approach, which was assessed as substantial by Landis and Koch and moderate by McHugh, which means that even though it did not reach the best Kappa and performance results among the classifiers, the KS approach is still considered a good method for classifying traffic in view of our scenario.

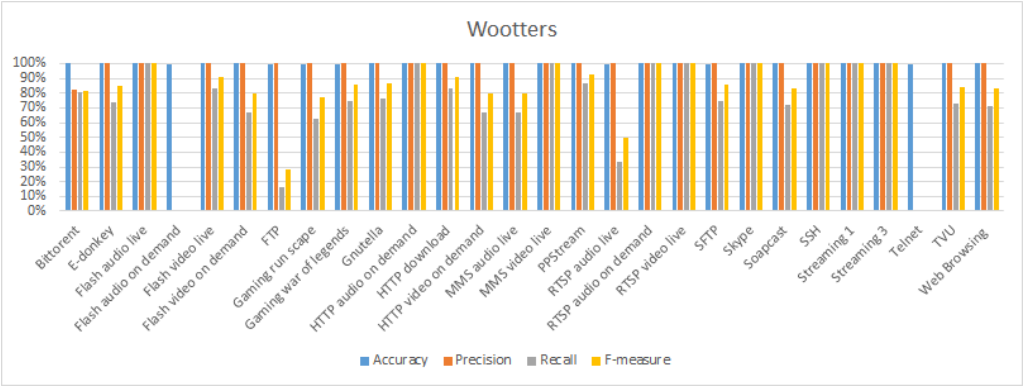


Figure 5.10: Classification results obtained using the Wootters Distance.

We observed that the Kappa value obtained by the Hellinger Distance was the highest when compared to those of the other classifiers, although the qualitative evaluation was the same for seven of the eight evaluated classifiers. Given this scenario, we verify that Hellinger qualitative analysis meets the performance values achieved by this classifier and that the qualitative analysis of KS meets the performance values achieved in terms of the F-Measure.

Table 5.6: Classification Results based on Bhattacharyya Distance in terms of Precision, Recall, Accuracy and F-Measure.

	Accuracy	Precision	Recall	F-Measure
Bittorent	0.999810	0.813725	0.815725	0.814724
Edonkey	0.999802	1.000000	0.741176	0.851351
Flash audio live	1.000000	1.000000	1.000000	1.000000
Flash audio on demand	0.999366	0.000000	0.000000	0.000000
Flash video live	0.999683	1.000000	0.833333	0.909091
Flash video on demand	0.999827	1.000000	0.666667	0.800000
FTP	0.998416	1.000000	0.166667	0.285714
Gaming runscape	0.999482	1.000000	0.625000	0.769231
Gaming war of legends	0.999525	1.000000	0.750000	0.857143
Gnutella	0.999907	1.000000	0.761905	0.864865
HTTP audio on demand	1.000000	1.000000	1.000000	1.000000
HTTP download	0.999683	1.000000	0.833333	0.909091
HTTP video on demand	0.999620	1.000000	0.666667	0.800000
MMS audio live	0.999728	1.000000	0.666667	0.800000
MMS video live	1.000000	1.000000	1.000000	1.000000
PPStream	0.999973	0.990991	0.866142	0.924370
RTSP audio live	1.000000	1.000000	1.000000	1.000000
RTSP audio on demand	1.000000	1.000000	1.000000	1.000000
RTSP video live	1.000000	1.000000	1.000000	1.000000
SFTP	0.999525	1.000000	0.750000	0.857143
Skype	1.000000	1.000000	1.000000	1.000000
Soapcast	0.999890	1.000000	0.719008	0.836538
SSH	1.000000	1.000000	1.000000	1.000000
Streaming 1	1.000000	1.000000	1.000000	1.000000
Streaming 3	1.000000	1.000000	1.000000	1.000000
Telnet	0.998099	0.000000	0.000000	0.000000
TVU	0.999895	1.000000	0.725490	0.840909
Web browsing	0.999696	1.000000	0.741935	0.851852

Table 5.7: Classification Results based on Chi-square in terms of Precision, Recall, Accuracy and F-Measure.

	Accuracy	Precision	Recall	F-Measure
Bittorent	0.999816	0.815981	0.828010	0.821951
Edonkey	0.999829	1.000000	0.776471	0.874172
Flash audio live	1.000000	1.000000	1.000000	1.000000
Flash audio on demand	0.999366	0.000000	0.000000	0.000000
Flash video live	0.999683	1.000000	0.833333	0.909091
Flash video on demand	0.999827	1.000000	0.666667	0.800000
FTP	0.998416	1.000000	0.166667	0.285714
Gaming runscape	0.999482	1.000000	0.625000	0.769231
Gaming war of legends	0.999525	1.000000	0.750000	0.857143
Gnutella	0.999907	1.000000	0.761905	0.864865
HTTP audio on demand	1.000000	1.000000	1.000000	1.000000
HTTP download	0.999366	1.000000	0.666667	0.800000
HTTP video on demand	0.999620	1.000000	0.666667	0.800000
MMS audio live	0.999728	1.000000	0.666667	0.800000
MMS video live	1.000000	1.000000	1.000000	1.000000
PPStream	0.999971	1.000000	0.850394	0.919149
RTSP audio live	1.000000	1.000000	1.000000	1.000000
RTSP audio on demand	1.000000	1.000000	1.000000	1.000000
RTSP video live	1.000000	1.000000	1.000000	1.000000
SFTP	0.999525	1.000000	0.750000	0.857143
Skype	1.000000	1.000000	1.000000	1.000000
Soapcast	0.999887	1.000000	0.710744	0.830918
SSH	1.000000	1.000000	1.000000	1.000000
Streaming 1	1.000000	1.000000	1.000000	1.000000
Streaming 3	1.000000	1.000000	1.000000	1.000000
Telnet	0.998099	0.000000	0.000000	0.000000
TVU	0.999902	1.000000	0.745098	0.853933
Web browsing	0.999696	1.000000	0.741935	0.851852

5.5.3 ROC Curves and their AUCs

ROC analysis was introduced in ML and Data Mining (DM) as a useful and powerful tool for the evaluation of classification models. It is a graphic way to evaluate, organize and select prevision systems. ROC analysis has also been used for building and refining models [253, 254]. It is particularly useful in areas where there are large disproportions between classes or when different costs/benefits yielded by the different classification errors/successes are taken into account for model refinement [255].

A graphic approach is showing the balance between false positive rate $FPR = TN$ represented by the X axis and the true positive rate $TPR = TP$ represented by the Y axis. Each model formed by the classifier corresponds to a point in the curve. Where TPR and FPR that are equal to 0 belong to the negative class. TPR and FPR that are equal 1 belong to the positive class and TPR=1 and FPR=0 denote the ideal model.

The perfect model is obtained when positive and negative examples are classified in the right way, represented by the point (0, 100 %). The models that make wrong previsions are represented by the point (100%, 0).

The ascending diagonal line (0,0) - (100% .100%) represents a model with stochastic behavior. If one point is above another an to its left in the ROC space, it means that the point is better than the other. The optimal model should be as close as possible to the point (0.100%).

The size of the ROC curve impacts directly on how good the model will be, so the bigger the curve the better the model will be, so the AUC value corresponds to the area delimited by ROC curve and X e Y axis [256]. Once the AUC is a fraction of the area of a square with side lengths of one, its value is always between 0 and 1. AUC =1 represents the perfect model, and AUC= 0.5 represent that the model simply provides random assumptions [257].

5.5.4 Discussions

In the classification process, the FPRs and TPRs of the applications were obtained, and the results were plotted in the ROC space. This is because in multiclassification, one of the

Table 5.8: Classification Results based on Hellinger Distance in terms of Precision, Recall, Accuracy and F-Measure.

	Accuracy	Precision	Recall	F-Measure
Bittorent	0.999825	0.838384	0.815725	0.826899
Edonkey	0.999802	1.000000	0.741176	0.851351
Flash audio live	1.000000	1.000000	1.000000	1.000000
Flash audio on demand	0.999366	0.000000	0.000000	0.000000
Flash video live	0.999683	1.000000	0.833333	0.909091
Flash video on demand	0.999827	1.000000	0.666667	0.800000
FTP	0.998416	1.000000	0.166667	0.285714
Gaming run scape	0.999482	1.000000	0.625000	0.769231
Gaming war of legends	0.999525	1.000000	0.750000	0.857143
Gnutella	0.999907	1.000000	0.761905	0.864865
HTTP audio on demand	1.000000	1.000000	1.000000	1.000000
HTTP download	0.999683	1.000000	0.833333	0.909091
HTTP video on demand	0.999620	1.000000	0.666667	0.800000
MMS audio live	0.999728	1.000000	0.666667	0.800000
MMS video live	1.000000	1.000000	1.000000	1.000000
PPStream	0.999974	1.000000	0.866142	0.928270
RTSP audio live	1.000000	1.000000	1.000000	1.000000
RTSP audio on demand	1.000000	1.000000	1.000000	1.000000
RTSP video live	1.000000	1.000000	1.000000	1.000000
SFTP	0.999525	1.000000	0.750000	0.857143
Skype	1.000000	1.000000	1.000000	1.000000
Soapcast	0.999890	1.000000	0.719008	0.836538
SSH	1.000000	1.000000	1.000000	1.000000
Streaming 1	1.000000	1.000000	1.000000	1.000000
Streaming 3	1.000000	1.000000	1.000000	1.000000
Telnet	0.998099	0.000000	0.000000	0.000000
TVU	0.999895	1.000000	0.725490	0.840909
Web browsing	0.999734	1.000000	0.774194	0.872727

classes can be marked as a positive class, and the other classes are all marked as negative classes. The classification effect can be better reflected by the AUC. The classification effect gets better, the higher the AUC value is. The maximum value of the AUC is 1.

The data were treated as described in Section 5.4.1 and classified according to the prototype developed in Section 5.4.3. For each method, we obtained classification and performance values. Analyzing the results of the ROC curves (AUCs), we can clearly verify which is the best statistical method for discriminating and classifying each application mapped in our data set.

Figures 5.11, 5.12, 5.13, and 5.14 show the ROC curve results (AUCs) obtained by the Jensen-Shannon, Hellinger, Bhattacharyya, and Wootters methods, and we compared them with the results achieved by the statistical methods implemented in some works found in the literature, such as the KS, Chi-square [69], Euclidean Distance and KL Distance [72] methods. Note that for comparison purposes, the methods found in the literature were also implemented in this work.

Table 5.9: Classification Results based on JSD in terms of Precision, Recall, Accuracy and F-Measure.

	Accuracy	Precision	Recall	F-Measure
Bittorent	0.999807	0.812808	0.810811	0.811808
Edonkey	0.999811	1.000000	0.752941	0.859060
Flash audio live	1.000000	1.000000	1.000000	1.000000
Flash audio on demand	0.999366	0.000000	0.000000	0.000000
Flash video live	0.999683	1.000000	0.833333	0.909091
Flash video on demand	0.999827	1.000000	0.666667	0.800000
FTP	0.998416	1.000000	0.166667	0.285714
Gaming run scape	0.999482	1.000000	0.625000	0.769231
Gaming war of legends	0.999525	1.000000	0.750000	0.857143
Gnutella	0.999907	1.000000	0.761905	0.864865
HTTP audio on demand	1.000000	1.000000	1.000000	1.000000
HTTP download	0.999049	0.800000	0.666667	0.727273
HTTP video on demand	0.999620	1.000000	0.666667	0.800000
MMS audio live	0.999728	1.000000	0.666667	0.800000
MMS video live	1.000000	1.000000	1.000000	1.000000
PPStream	0.999971	0.990909	0.858268	0.919831
RTSP audio live	1.000000	1.000000	1.000000	1.000000
RTSP audio on demand	1.000000	1.000000	1.000000	1.000000
RTSP video live	1.000000	1.000000	1.000000	1.000000
SFTP	0.999525	1.000000	0.750000	0.857143
Skype	1.000000	1.000000	1.000000	1.000000
Soapcast	0.999890	1.000000	0.719008	0.836538
SSH	1.000000	1.000000	1.000000	1.000000
Streaming 1	1.000000	1.000000	1.000000	1.000000
Streaming 3	1.000000	1.000000	1.000000	1.000000
Telnet	0.998099	0.000000	0.000000	0.000000
TVU	0.999902	1.000000	0.745098	0.853933
Web browsing	0.999696	1.000000	0.741935	0.851852

In the ROC space, the classification result gets better every time it gets closer to the upper left corner. As we can see in figures 5.11, 5.12, 5.13, and 5.14, the classifiers implemented in this work had good classification effects. The comprehensive analysis shows that the FPRs of the classification results were very low; the TPRs were close to 1. We can deduce that the probability of classifier judgment error was very low, and the existing samples could be classified accurately.

The AUCs of the Hellinger, Jensen-Shannon, Bhattacharyya and Chi-square methods were all 0.91, indicating that the discrimination accuracies of these methods for the bittorent application were relatively high (superior to those of methods such as the Euclidean, KL, Wootters and KS approaches). For the edonkey application, the method Chi-square was the one that had the best discrimination Accuracy, with an AUC of 0.89.

For the flash video application, the method that indicated the best discrimination Accuracy was the KL approach, presenting an AUC of 1.00. This indicates that the KL method achieved the best classification effect. For the flash audio on demand, FTP and telnet applications, all methods, despite having AUCs greater than or equal to 0.5, were very close

Table 5.10: Classification Results based on Wootters in terms of Precision, Recall, Accuracy and F-Measure.

	Accuracy	Precision	Recall	F-Measure
Bittorent	0.999811	0.822055	0.805897	0.813896
Edonkey	0.999802	1.000000	0.741176	0.851351
Flash audio live	1.000000	1.000000	1.000000	1.000000
Flash audio on demand	0.999366	0.000000	0.000000	0.000000
Flash video live	0.999683	1.000000	0.833333	0.909091
Flash video on demand	0.999827	1.000000	0.666667	0.800000
FTP	0.998416	1.000000	0.166667	0.285714
Gaming run scape	0.999482	1.000000	0.625000	0.769231
Gaming war of legends	0.999525	1.000000	0.750000	0.857143
Gnutella	0.999907	1.000000	0.761905	0.864865
HTTP audio on demand	1.000000	1.000000	1.000000	1.000000
HTTP download	0.999683	1.000000	0.833333	0.909091
HTTP video on demand	0.999620	1.000000	0.666667	0.800000
MMS audio live	0.999728	1.000000	0.666667	0.800000
MMS video live	1.000000	1.000000	1.000000	1.000000
PPStream	0.999974	1.000000	0.866142	0.928270
RTSP audio live	0.998733	1.000000	0.333333	0.500000
RTSP audio on demand	1.000000	1.000000	1.000000	1.000000
RTSP video live	1.000000	1.000000	1.000000	1.000000
SFTP	0.999525	1.000000	0.750000	0.857143
Skype	1.000000	1.000000	1.000000	1.000000
Soapcast	0.999890	1.000000	0.719008	0.836538
SSH	1.000000	1.000000	1.000000	1.000000
Streaming 1	1.000000	1.000000	1.000000	1.000000
Streaming 3	1.000000	1.000000	1.000000	1.000000
Telnet	0.998099	0.000000	0.000000	0.000000
TVU	0.999895	1.000000	0.725490	0.840909
Web browsing	0.999658	1.000000	0.709677	0.830189

to the minimum acceptable limit, corresponding to random assumption classifiers. On the other hand, for the SSH and http video on demand applications, all methods exhibited the same capacity to provide high-performance instances.

The KS method presented the worst performance among the eight classifiers analyzed, achieving the maximum performance in only 3 types of applications. Regarding the average AUC, this method predicted 74.29% of instances. Therefore, this method is not suitable for classifying the application flows presented in this work.

The Euclidean, KL and Wootters methods achieved similar performance. Among the 28 applications analyzed, the Euclidean method achieved high performance in 14 applications, with the capacity to forecast 86.18% of the instances. In turn, KL and Wootters achieved high performance in 19 applications and exhibited capacities to forecast 86.61% and 86.79% of the instances, respectively. Therefore, these methods can be used in specific flows, where they showed high performance.

The Chi-square method expressed high performance in 22 of the 28 applications, while the Hellinger, Jensen-Shannon, and Bhattacharyya methods expressed high performance in 21 applications. Despite this, the Hellinger method stood out with the highest instance prediction capacity (88.14%), while the Bhattacharyya method had the second-largest capacity at 88.07%, followed by Chi-square with 87.86% and Jensen-Shannon with 87.82%. Therefore, these 4 methods stood out as strictly better models.

Distances specify different ways of combining attributes. The results suggested that statistical methods are capable of predicting instances with high performance. For traffic classification, we have two statistical methods that stood out and solved the classification problem: the Hellinger and Chi-square methods. However, we still have room to improve the TPRs of these methods. The probabilistic problem found here is that the statistical classifiers had an excellent ability to identify false positives, reaching 100% FPRs; however, they were still unable to fully maximize the TPR. Among the methods tested in this work, Hellinger is the most evolved, as it a refinement of distances such as the Bhattacharyya and Jensen-Shannon Distances; this proves that improve upon a method can increase its probability of success.

It is possible to observe that for the flash audio on demand flow, the accuracy was 0.999366 and for the telnet flow the accuracy was 0.998099, that means that the set of classifiers implemented on this article rejects with high precision all the flow samples that are not really flash audio on demand or Telnet. For the flash audio on demand flows and telnet, the Recall, Precision and F-Measure presented values were 0.00, which means that the set of classifiers reject all possible flow samples of flash audio on demand flows and telnet, taking us to the conclusion that the rules to accept or reject the flow samples included on the implemented classifiers were not enough to correctly accept flow samples that really

belong to those two applications. Note that the rules inserted to the set of classifiers were enough to get 26 kinds of applications right from the 28 used, which means 92,85% of the selected applications in this work were correctly classified. Observe that it can be a positive thing when the classifier rejects all samples that it is not sure of the flow characteristic, because it prevents malicious code from entering the network. In addition, the possibility for new research of classifiers that are able to sufficiently identify flash audio on demand flows and telnet.

5.5.5 Computational Resource Usage

Every experiment described was performed on a machine with the following specifications: the Ubuntu 14.04.5 operating system; an Intel Core (TM) i7 870 CPU at 2.93 GHz; a 64-bit desktop computer a file size of 1200MHz; 6 GB of system memory; and an Ethernet interface gigabit Ethernet 100 Mbit/s with a capacity of 1 Gbit/s, a width of 64 bits, and a clock of 33 MHz.

The computational cost was obtained with the help of the $psrecord$ tool [220]. The computational performance of KS, Chi-square, Euclidean and KL methods were presented and described on the published work in [71, 72], shown in the chapters 3 and 4. Activity files referring to each statistical method were used to measure computational resources in terms of CPU and memory that were used through classification. Through the graphs represented by Figure 5.15, we could analyze the computational costs of the methods implemented in this work.

An important factor in decision taking and when choosing a classifier or algorithm is the computational cost. It should be taken under consideration because on package processing, accessing memory usually is the operation that has the highest computational cost.

Computational performance was presented according to the relationship between the CPU consumption (in %) and memory consumption (in MB) spent during the database classification period. On the conversion of our original database to the new database with relative frequency, we used the number of packages, hosts/ports only as information and not to classify or identify the applications.

Analyzing the results, we can see that the longest execution time was yielded by the JSD, which took more than twice the time of the other methods for execution. On the other hand, the memory cost of the JSD was one of the lowest (at less than 70 MB) in comparison with that of the Wootters, Bhattacharyya and Hellinger approaches. The memory cost for the Jensen-Shannon Distance, which was one of the lowest less than 70 MB, when compared to Wootters, Bhattacharyya and Hellinger.

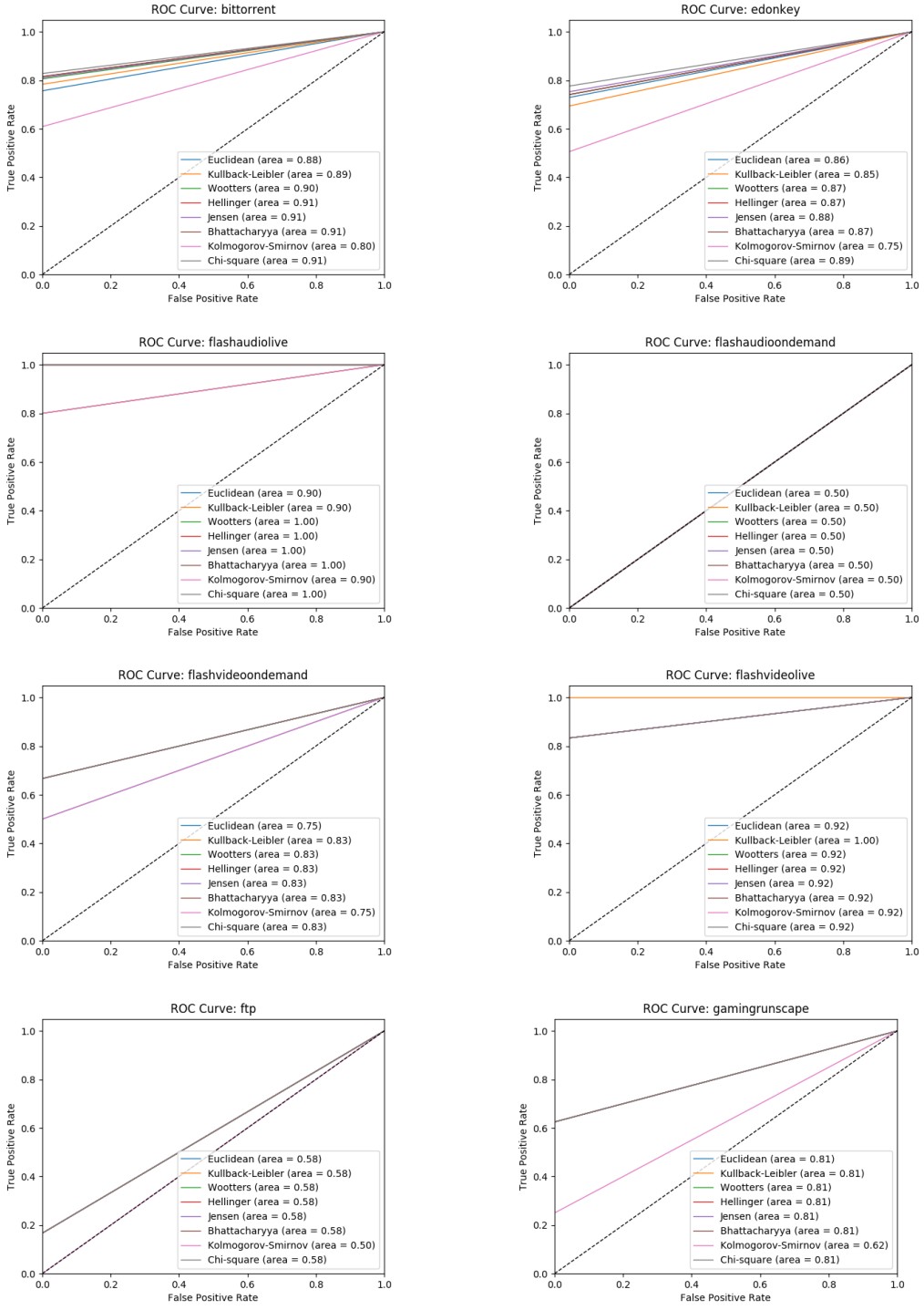


Figure 5.11: The ROC curves of the classifiers referring to applications bittorrent, edonkey, flash audio live flash audio on demand, flash video on demand, gaming war of legends, ftp, gaming run scape.

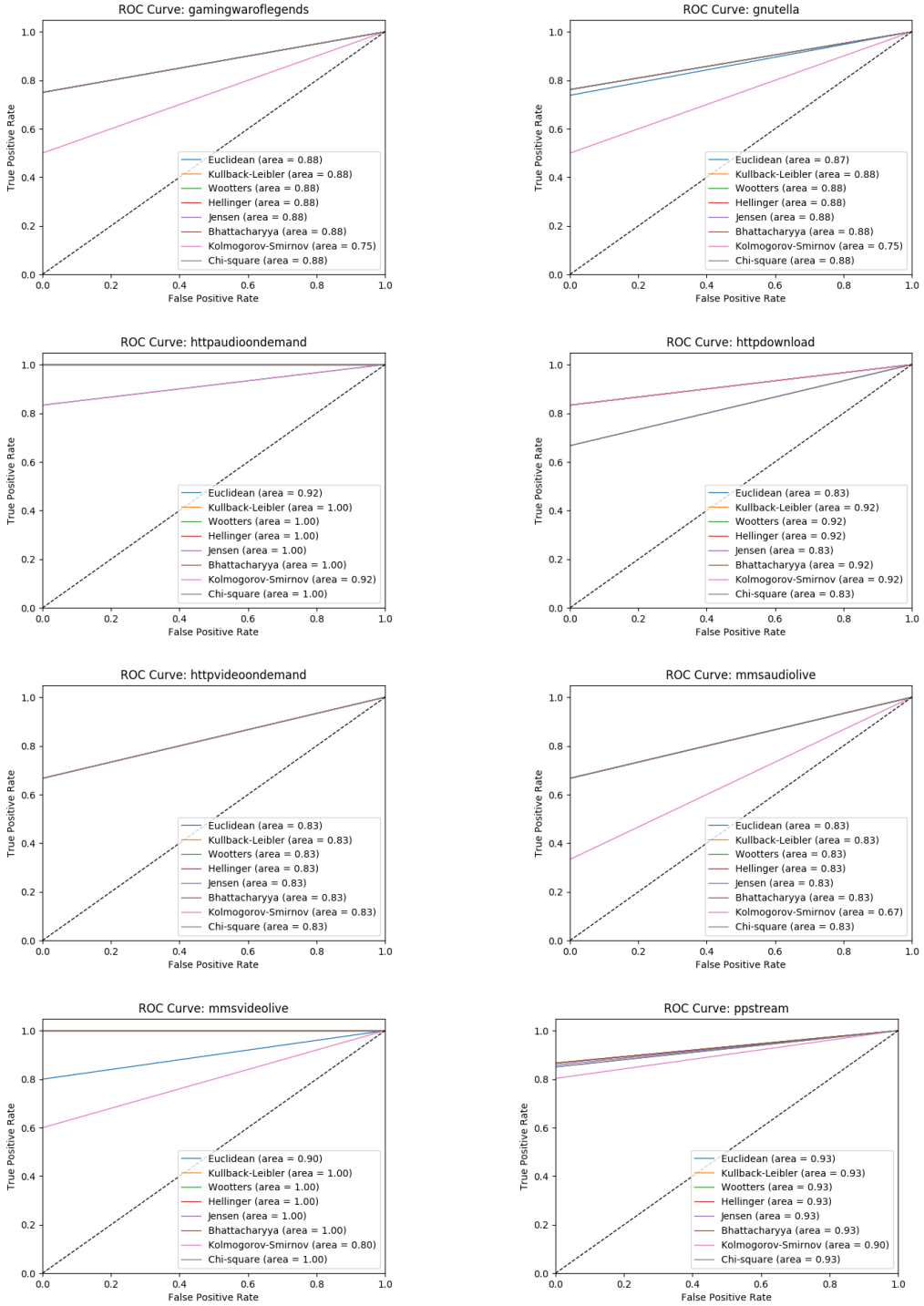


Figure 5.12: The ROC curves of classifiers referring to applications gaming war of legends, gnutella, http audio on demand, http download, http video on demand, mms audio live, mms video live, ppstream.

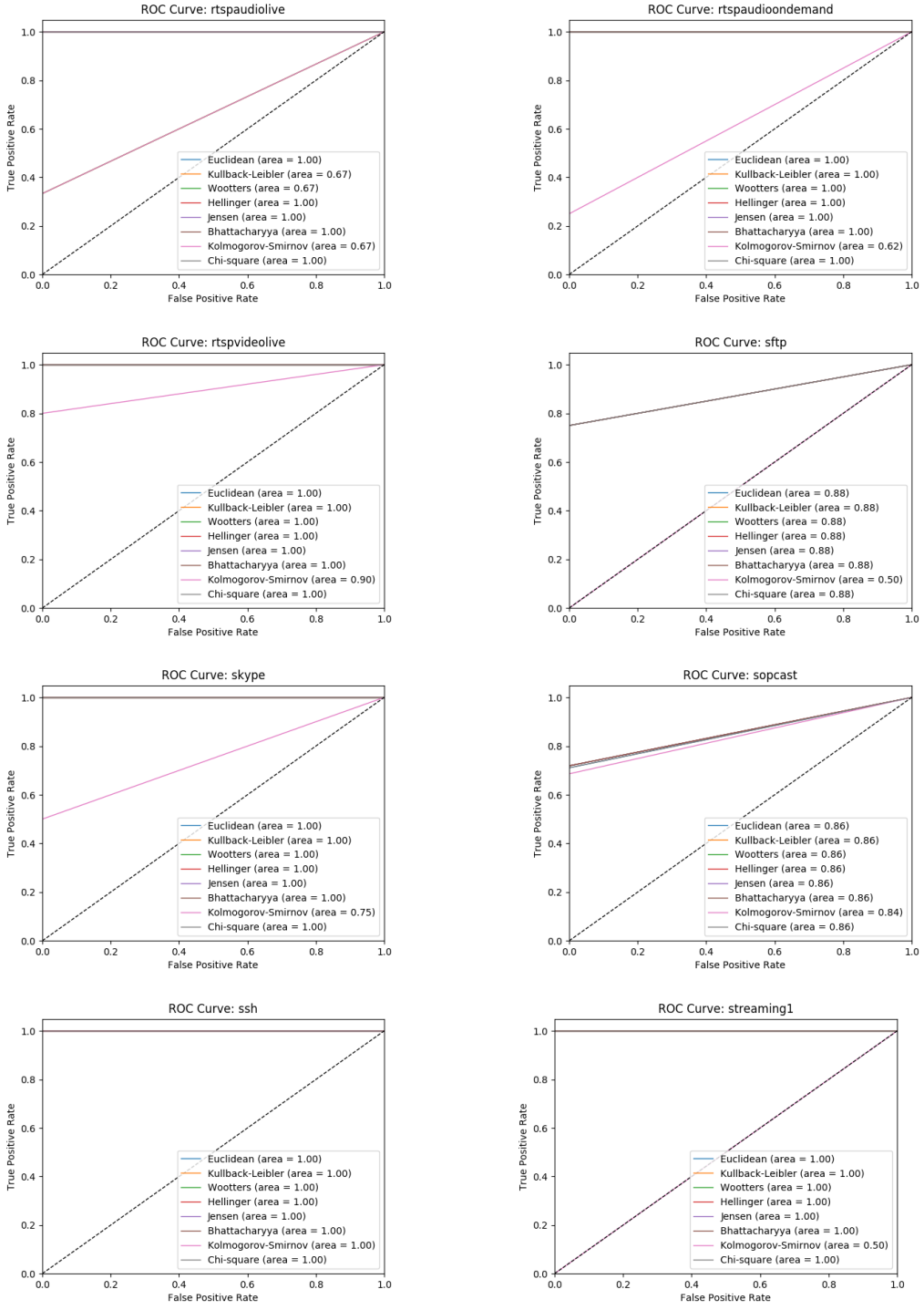


Figure 5.13: The ROC curves of the classifiers referring to applications rtsp audio live, rtsp audio on demand, rtsp video live, stfp, skype, sopcast, ssh, streaming 1.

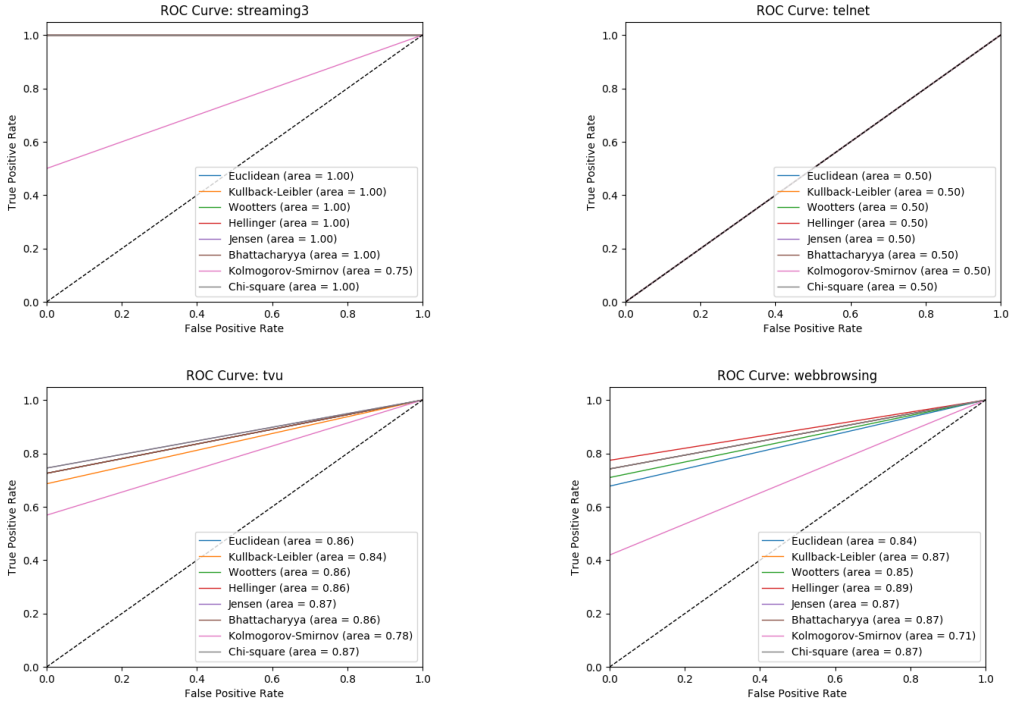


Figure 5.14: The ROC curves of the classifiers referring to applications streaming 3, telnet, tvu, and webbrowsing.

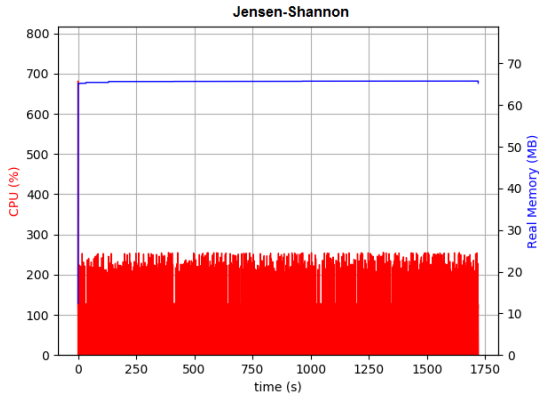
The computational costs of the methods were also calculated in terms of execution time, CPU costs and memory costs. The developed classifier classification effect was considered robust by our results at a relatively competitive computational cost when compared those of the other methods found.

5.6 Conclusion

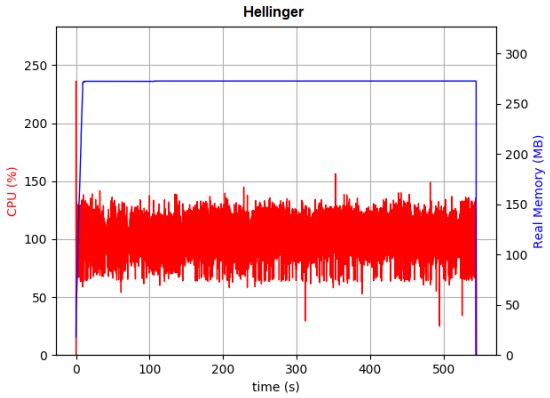
The purpose of this article was to study, implement and test various statistical methods for classifying encrypted network traffic. Traffic flows were mapped using relative frequencies to characterize each application flow. Through the flow properties, statistical traffic models were applied for the classification of traffic.

Our study was based on statistical JSD methods, Bhattacharyya Distance, Hellinger Distance, and Wootters Distance. We compared those implemented methods in means of network traffic classification to already implemented and tested statistical methods such as KS, Chi-square, KL approaches and Euclidean Distance.

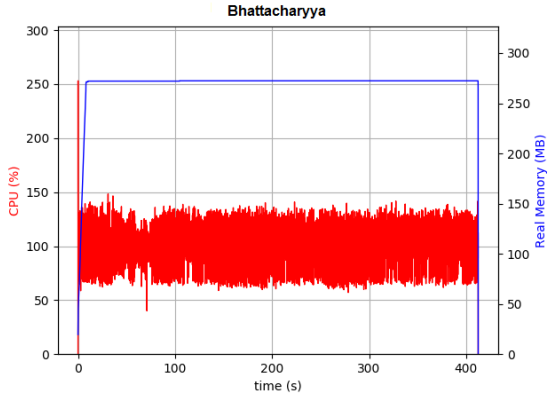
In addition to implementing these methods, we also evaluated their performance on the obtained Precision, Recall, Accuracy and F-Measure values. This analysis was important to verify the prediction efficiency of each method based on the calculated distances.



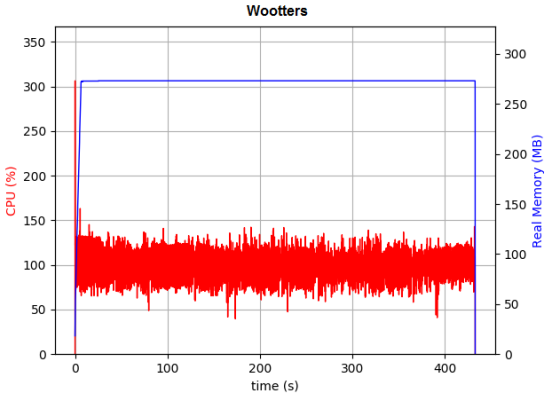
(a) Jensen-Shannon



(b) Hellinger



(c) Bhattacharyya



(d) Wootters

Figure 5.15: Representations of the CPU usage and memory consumption from the beginning of the analysis of the trace to the classification step required for the statistical methods (Jensen-Shannon, Hellinger, Bhattacharyya and Wootters).

We presented the Accuracy of each implemented classifier, obtaining average values above 90% for the Jensen-Shannon, Bhattacharyya, Hellinger and Wootters methods. These methods are viable options for the classification of encrypted Internet traffic.

The quantitative assessment performed by Kappa values showed that classifiers based on Jensen-Shannon, Bhattacharyya, Hellinger e Wootters methods were presented as "almost perfect reliability" according to the Kappa scale [24] and "Strong" [25], because they represent Kappa values higher than 0.8. Hellinger presented the best Kappa value results in comparison with those of the other classifiers.

The obtained Kappa values were very close to the obtained F-Measure values. Although they are computed differently, both yielded the same results. This led us to conclude that for the quantitative evaluation of a classifier, both tests produced the same results, helping to validate the implemented methods and pointing to Hellinger as a promising method for classifying encrypted traffic.

We present the results of the ROC curves (AUCs) for the Jensen-Shannon, Hellinger, Bhattacharyya, and Wootters methods. We compared them with the results achieved by statistical methods implemented in some works found, such as the KS, Chi-square , Euclidean and KL Distance approaches. We conclude that distances are valuable, accurate and economical methods of discrimination and that they have high predictive power for network traffic. For future work, we intend to apply distances and divergences to identify intrusions or attacks.

Chapter 6

Conclusions and Future Work

This chapter presents a summary of the main scientific contributions and conclusions. Furthermore, it discusses a few research topics related to the work developed in the doctoral program that may be addressed in the future.

6.1 Conclusions

This thesis focused on the proposal of a new methodological approach to classify encrypted and protocol-obfuscated Internet traffic based on statistical methods that aim to be similar or better performance than classification using Support Vector Machine (SVM) with the adequate computational resources in terms of CPU and memory. A set of statistical classifiers based on distances and divergences were proposed, implemented and evaluated. They are more specifically based on Euclidean Distance, Kullback-Leibler (KL) Divergence, Jensen-Shannon Divergence, Wootters Divergence, Hellinger Distance and Bhattacharyya Distance. Besides that, it was also proposed, implemented and evaluated a classifier based on SVM, a classifier based on Kolmogorov-Smirnov tests and a classifier based on Chi-square test for comparison means.

By hypothesis, we suggest that statistical models based on distances and divergences are capable of classifying in an efficient way encrypted and obfuscated Internet traffic and protocol that use random or unknown ports, with adequate computational resources, showing potential of usage for new Internet traffic classification models.

The first contribution of this thesis consisted of a wide literature review about encrypted and obfuscated Internet traffic classification. Through this review, it was possible to see that in the current literature the methods for Internet traffic classification based on ports and payload have become obsolete because of the increase of P2P traffic using unknown or random ports, encrypted traffic and encapsulations of multilayer data. It was also possible to see that ML based methods have become limited because of its computational complexity and high costs.

The second contribution of this thesis consisted of the proposal, implementation and evaluation of a classifier based on Support Vector Machine (SVM) for P2P multimedia traffic compared to the results of Kolmogorov-Smirnov (KS) and Chi-square tests. Internet traffic classification based on the SVM method with a Linear kernel with the right

parameters for the Self C parameter presents good results to classify encrypted P2P multimedia traffic on the Internet.

The third contribution of this thesis consisted of the proposal, implementation and evaluation of two classifiers based on Kullback-Leibler (KL) Distance or Euclidean Distance compared to SVM. KL and Euclidean methods are capable of working in real time and do not need to be retrained every time a new traffic type appears, being a good alternative to SVM classification method for almost all evaluated protocols, P2P and streaming mostly, with a high precision level and a lower computational resource usage.

The fourth contribution of this thesis consisted of the proposal, implementation and evaluation of a set of classifiers of encrypted Internet traffic based on Jensen-Shannon Divergence and Hellinger, Bhattacharyya and Wootters Distances compared to Euclidean, KL, KS and Chi-square results. All statistical methods used for classification of encrypted and obfuscated Internet traffic presented an average Precision value over 90% classified as "almost perfect confiability" and "strong" in Kappa values, with the exception of KS (classified as "considerable" and "moderate"). Hellinger Distance presented the best results in Kappa values when compared to the other classifiers, being highlighted as a more robust, strong and dependable method.

We identified that the classifiers implemented using statistical methods are capable of overcoming some limitations such as the computational complexity, quantity of features used during classification, real time operation, big quantity of flows and heavy traffic.

It is concluded that statistical methods based on divergences and distances can be valuable, precise and of low cost to implement to classify encrypted and obfuscated Internet traffic, confirming the initial hypothesis and validating the argument presented in this thesis with expectations that it can be recognized as a reliable and useful tool for Internet traffic classification.

6.2 Future Work

Although all specific purposes have been accomplished in this thesis, we realized that there are limitations when distances and divergences are used for classification. We realized that when selecting the classifier, after calculating the distances and divergences, the debited values on their own would not be enough to make decisions and later the classification. That is when we understood that it would be necessary to create rules so that the classifier could accept or reject flow samples. Those rules were based on distance and divergence properties. We realized that the decision factor can influence when choosing the classifier and on the classification results, making the classifier to reject everything or accept all flows that do not belong to the true classes. Note that this limitation did not invalid the evaluations and the results presented in this thesis. However, in order to in-

crease the sensitivity of the decision factor of the set of classifiers, other rules based on the properties of distances and divergences can be further investigated. The prototype of the set of classifiers implemented in this thesis was developed using Python language version 2.7 on a desktop computer. This set of classifiers was not tested on devices like smartphones, tablets or Arduino.

The work developed through this thesis allowed us to foresee future research lines. Some open issues identified during literature review were not solved, leaving an open space for interesting developments in the area. One possible research direction is to explore the optimization of SVM, namely the Self C parameter, or explore the combination of SVM with divergences and distances. In the literature we found works that combine Euclidean Distance with K-means algorithms for classifiers, and Kullback-Leibler combine with SVM. However, we did not find classifiers that were a combination of Hellinger Distance, Wootters Distance and Jensen-Shannon Divergence. A possibility is exploring the viability of using Manhattan, Mahalanobis and Minkowski similarities for encrypted Internet traffic classification. Another important possibility is using classifiers based on distances and divergences to detect malware, intrusions and other kinds of attack.

Bibliography

[1] M. Cotton, L. Eggert, J. Touch, M. Westerlund, and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry," *RFC*, vol. 6335, pp. 1–33, August 2011. xv, 1, 12, 23

[2] M. Shen, Y. Liu, L. Zhu, K. Xu, X. Du, and N. Guizani, "Optimizing feature selection for efficient encrypted traffic classification: A systematic approach," *IEEE Network*, vol. 34, no. 4, pp. 20–27, July 2020. xv, xvi, 1, 2, 13, 16, 20, 21, 22, 23, 24, 25

[3] N. Alqudah and Q. Yaseen, "Machine learning for traffic analysis: a review," *Procedia Computer Science*, vol. 170, pp. 911–916, April 2020. xv, 1, 2, 13, 14, 16, 20, 24

[4] A. Headquarters, "WAN and Application Optimization Solution Guide Cisco Validated Design," *Cisco Systems, Inc*, November 2008. xv, 1, 12

[5] G. Sun, L. Liang, T. Chen, F. Xiao, and F. Lang, "Network traffic classification based on transfer learning," *Computers & electrical engineering*, vol. 69, pp. 920–927, 2018. xv, 1, 80, 82

[6] Y. Liu, "A Survey of Machine Learning Based Packet Classification," in *Symposium on Computational Intelligence for Security and Defence Applications (CISDA)*, July 2009. xv, 13

[7] M. Uğurlu, İ. A. Doğru, and R. S. Arslan, "A new classification method for encrypted internet traffic using machine learning," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 29, no. 5, pp. 2450–2468, 2021. xv, xvi, 1, 2

[8] M. Li, S. Lim, and H. Feng, "A novel description of multifractal phenomenon of network traffic based on generalized cauchy process," in *Computational Science–ICCS 2007*. Springer, 2007, pp. 1–9. xv, 1, 2

[9] H. Shi, G. Liang, and H. Wang, "A novel traffic identification approach based on multifractal analysis and combined neural network," *annals of telecommunications-annales des télécommunications*, vol. 69, no. 3-4, pp. 155–169, 2014. xv, 1

[10] E. Hjelmvik and W. John, "Breaking and improving protocol obfuscation," *Chalmers University of Technology, Tech. Rep*, vol. 123751, 2010. xv, xvi, 1

[11] J. Duchêne, E. Alata, V. Nicomette, M. Kaâniche, and C. Le Guernic, "Specification-Based Protocol Obfuscation," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2018, pp. 478–489. xvi, 1

[12] J. Khan, *A Study in Protocol Obfuscation Techniques and Their Effectiveness*. GRIN Verlag, 2018. xvi, 1

[13] Y. Zhao, Y. Yang, B. Tian, J. Yang, T. Zhang, and N. Hu, "Edge Intelligence Based Identification and Classification of Encrypted Traffic of Internet of Things," *IEEE Access*, vol. 9, pp. 21 895–21 903, 2021. xvi, 3

[14] Y. Dhote, S. Agrawal, and A. J. Deen, "A survey on feature selection techniques for internet traffic classification," in *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE, August 2015, pp. 1375–1380. xvi, 1, 2, 13, 16, 20, 22, 25

[15] O. Salman, I. H. Elhajj, A. Kayssi, and A. Chehab, "A review on machine learning-based approaches for internet traffic classification," *Annals of Telecommunications*, vol. 75, no. 11, pp. 673–710, June 2020. xvii, 13, 16, 20

[16] Y. Dodge and D. Commenges, *The Oxford dictionary of statistical terms*. Oxford University Press on Demand, October 2006. xix, 13, 30, 33

[17] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*. Pearson Education India, March 2019. xix, 30, 66

[18] A. Majtey, P. Lamberti, M. Martin, and A. Plastino, "Wootters' distance revisited: a new distinguishability criterium," *The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics*, vol. 32, no. 3, pp. 413–419, January 2005. xix, 32, 33, 86, 87

[19] F. Nielsen, "On the Jensen–Shannon symmetrization of distances relying on abstract means," *Entropy*, vol. 21, no. 5, p. 485, May 2019. xx, 31, 32, 33, 85, 86

[20] S. Parveen, S. K. Singh, U. Singh, and D. Kumar, "A comparative study of traditional and kullback-leibler divergence of survival functions estimators for the parameter of lindley distribution," *Austrian Journal of Statistics*, vol. 48, no. 5, pp. 45–53, July 2019. xx, 32, 33, 65

[21] M. Martin, A. Plastino, and O. Rosso, "Generalized statistical complexity measures: Geometrical and analytical properties," *Physica A: Statistical Mechanics and its Applications*, vol. 369, no. 2, pp. 439–462, September 2006. xx, 33, 34, 87

[22] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and A. V. Vasilakos, "An effective network traffic classification method with unknown flow detection," *IEEE Transactions on Network and Service Management*, vol. 10, no. 2, pp. 133–147, March 2013. xxv, 26, 27, 40, 47, 94

[23] D. Rossiter, "Statistical methods for accuracy assesment of classified thematic maps," *Tech Note Enschede: Int Instit Geo-information Sci Earth Observation (ITC)*, 2004. xxv, 95

[24] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics*, pp. 159–174, 1977. xxv, xxvi, xxvii, 95, 96, 98, 111

[25] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia medica: Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012. xxvi, xxvii, 95, 96, 98, 111

[26] F. Constantinou and P. Mavrommatis, "Identifying known and unknown peer-to-peer traffic," in *Network Computing and Applications, 2006. NCA 2006. Fifth IEEE International Symposium on*. IEEE, 2006, pp. 93–102. 1, 2

[27] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Passive and Active Network Measurement*. Springer, 2005, pp. 41–54. 2

[28] H. Pereira, A. Ribeiro, and P. Carvalho, "Improving traffic classification and policing at application layer," in *Computers and Communications (ISCC), 2010 IEEE Symposium on*. IEEE, 2010, pp. 291–294. 2

[29] N. Cascarano, L. Ciminiera, and F. Risso, "Optimizing deep packet inspection for high-speed traffic analysis," *Journal of Network and Systems Management*, vol. 19, no. 1, pp. 7–31, 2011. 2, 63, 82

[30] P. Dorfinger, G. Panholzer, B. Trammell, and T. Pepe, "Entropy-based traffic filtering to support real-time Skype detection," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*. ACM, 2010, pp. 747–751. 2

[31] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing skype traffic: when randomness plays with you," in *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4. ACM, 2007, pp. 37–48. 2

[32] K. Xu, M. Zhang, M. Ye, D. M. Chiu, and J. Wu, "Identify P2P traffic by inspecting data transfer behavior," *Computer Communications*, vol. 33, no. 10, pp. 1141–1150, 2010. 2

[33] B.-C. Park, Y. J. Won, M.-S. Kim, and J. W. Hong, "Towards automated application signature generation for traffic identification," in *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*. IEEE, 2008, pp. 160–167. 2

[34] L. Bin and T. Hao, "P2P Traffic Classification Using Semi-Supervised Learning," in *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on*, vol. 1. IEEE, 2010, pp. 408–412. 2

[35] T. Seyed Tabatabaei, M. Adel, F. Karray, and M. Kamel, "Machine learning-based classification of encrypted internet traffic," in *International Workshop on Machine*

Learning and Data Mining in Pattern Recognition. Springer, 2012, pp. 578–592. 2

[36] A. Bianco, G. Mardente, M. Mellia, M. Munafò, and L. Muscariello, "Web user-session inference by means of clustering techniques," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 2, pp. 405–416, 2009. 2

[37] S. Hao, J. Hu, S. Liu, T. Song, J. Guo, and S. Liu, "Improved SVM method for internet traffic classification based on feature weight learning," in *2015 international conference on control, automation and information sciences (ICCAIS)*. IEEE, October 2015, pp. 102–106. 2, 3, 40, 46

[38] Z. Fan and R. Liu, "Investigation of machine learning based network traffic classification," in *2017 International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, August 2017, pp. 1–6. 2, 3, 39, 46, 52

[39] S. Angra and S. Ahuja, "Machine learning and its applications: A review," in *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*. IEEE, March 2017, pp. 57–60. 2, 24, 25

[40] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *applied sciences*, vol. 9, no. 20, p. 4396, October 2019. 2, 13, 16, 24, 25

[41] "Evolving statistical rulesets for network intrusion detection," *Applied soft computing*, vol. 33, pp. 348–359, August 2015. 2, 25

[42] J. V. Gomes, P. R. Inácio, M. Pereira, M. M. Freire, and P. P. Monteiro, "Detection and classification of peer-to-peer traffic: A survey," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, pp. 1–40, June 2013. 2, 13, 16

[43] J. V. Gomes, "Classification of Peer-to-Peer Traffic by Exploring the Heterogeneity of Traffic Features Through Entropy," *PhD Thesis, PhD in Computer Science and Engineering*, 2012. 2

[44] W. John and S. Tafvelin, "Heuristics to classify internet backbone traffic based on connection patterns," in *Information Networking, 2008. ICOIN 2008. International Conference on*. IEEE, 2008, pp. 1–5. 2

[45] F. Garcia-Palacios, "Host Based P2P Flow Identification and Use in Real-Time," 2010. 2

[46] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: multilevel traffic classification in the dark," in *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4. ACM, 2005, pp. 229–240. 2

[47] M. Iliofotou, H.-c. Kim, M. Faloutsos, M. Mitzenmacher, P. Pappu, and G. Varghese, "Graption: A graph-based P2P traffic classification framework for the internet backbone," *Computer Networks*, vol. 55, no. 8, pp. 1909–1920, 2011. 2

[48] Akamai, "State of the Internet / security Retail Attacks and API Traffic Reportht, volume 5, Issue 2," https://www.akamai.com/site/fr/documents/state-of-the-internet/state-of-the-internet-security-retail-attacks-and-api-traffic-report-2019.pdf, April 2019, april, 2022. 3

[49] J. Costlow, "How Decryption of Network Traffic Can Improve Security," https://threatpost.com/decryption-improve-security/176613/, November 2021, november, 2021. 3

[50] C. Gu, S. Zhang, and H. Huang, "Online internet traffic classification based on proximal SVM," *Journal of Computational Information Systems*, vol. 7, no. 6, pp. 2078–2086, 2011. 3, 80, 81

[51] S. Hao, J. Hu, S. Liu, T. Song, J. Guo, and S. Liu, "Network traffic classification based on improved DAG-SVM," in *2015 International Conference on Communications, Management and Telecommunications (ComManTel)*. IEEE, December 2015, pp. 256–261. 3, 40, 46

[52] J. Cao, Z. Fang, G. Qu, H. Sun, and D. Zhang, "An accurate traffic classification model based on support vector machines," *International Journal of Network Management*, vol. 27, no. 1, p. e1962, January 2017. 3, 39, 46

[53] G. Sun, T. Chen, Y. Su, and C. Li, "Internet traffic classification based on incremental support vector machines," *Mobile Networks and Applications*, vol. 23, no. 4, pp. 789–796, February 2018. 3, 40, 42, 43, 46, 63, 64, 83

[54] J. Tang, X. Chen, Z. Hu, F. Zong, C. Han, and L. Li, "Traffic flow prediction based on combination of support vector machine and data denoising schemes," *Physica A: Statistical Mechanics and its Applications*, vol. 534, p. 120642, November 2019. 3, 39, 46, 50, 51

[55] Y. Miao, Z. Ruan, L. Pan, J. Zhang, and Y. Xiang, "Comprehensive analysis of network traffic data," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 5, p. e4181, July 2018. 3, 39, 46

[56] R. Aggarwal and N. Singh, "A new hybrid approach for network traffic classification using SVM and Naïve Bayes algorithm," *Int. J. Comput. Sci. Mobile Comput*, vol. 6, pp. 168–174, August 2017. 3, 40, 46, 47, 50, 51

[57] J. Xiao, "SVM and KNN ensemble learning for traffic incident detection," *Physica A: Statistical Mechanics and its Applications*, vol. 517, pp. 29–35, March 2019. 3, 39, 46, 50

[58] C. Luo, C. Huang, J. Cao, J. Lu, W. Huang, J. Guo, and Y. Wei, "Short-term traffic flow prediction based on least square support vector machine with hybrid optimization algorithm," *Neural processing letters*, vol. 50, no. 3, pp. 2305–2322, March 2019. 3, 39, 46, 50, 51

[59] I. Syarif, A. Prugel-Bennett, and G. Wills, "Svm parameter optimization using grid search and genetic algorithm to improve classification performance," *Telkomnika*, vol. 14, no. 4, p. 1502, December 2016. 3, 26, 39, 46, 50

[60] A. A. Akinyelu and A. E. Ezugwu, "Nature Inspired Instance Selection Techniques for Support Vector Machine Speed Optimization," *IEEE Access*, vol. 7, pp. 154 581–154 599, October 2019. 3, 40, 46, 50, 51

[61] S. Dong, "Multi class SVM algorithm with active learning for network traffic classification," *Expert Systems with Applications*, vol. 176, p. 114885, August 2021. 3, 39, 46

[62] M. Aamir and S. M. A. Zaidi, "Clustering based semi-supervised machine learning for DDoS attack classification," *Journal of King Saud University-Computer and Information Sciences*, May 2019. 3, 39, 46, 50, 51

[63] S. S. L. Pereira, J. E. B. Maia *et al.*, "ITCM: A Real Time Internet Traffic Classifier Monitor," *arXiv preprint arXiv:1501.01321*, January 2015. 4, 36, 37

[64] H. Singh, "Performance analysis of unsupervised machine learning techniques for network traffic classification," in *2015 Fifth International Conference on Advanced Computing & Communication Technologies*. IEEE, February 2015, pp. 401–404. 4, 36, 37

[65] H. Shi, H. Li, D. Zhang, C. Cheng, and W. Wu, "Efficient and robust feature extraction and selection for traffic classification," *Computer Networks*, vol. 119, pp. 1–16, June 2017. 4, 24, 36, 37, 66, 82, 83, 84

[66] B. Schmidt, A. Al-Fuqaha, A. Gupta, and D. Kountanis, "Optimizing an artificial immune system algorithm in support of flow-Based internet traffic classification," *Applied Soft Computing*, vol. 54, pp. 1–22, May 2017. 4, 36, 37, 64, 73, 82, 84

[67] P. Zhu, S. Zhang, H. Luo, and Z. Wu, "A semi-supervised method for classifying unknown protocols," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. IEEE, March 2019, pp. 1246–1250. 4, 36, 37

[68] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, "Robust network traffic classification," *IEEE/ACM transactions on networking*, vol. 23, no. 4, pp. 1257–1270, May 2014. 4, 40, 47

[69] M. Neto, J. V. Gomes, M. M. Freire, and P. R. Inácio, "Real-time traffic classification based on statistical tests for matching signatures with packet length distributions," in *2013 19th IEEE Workshop on Local & Metropolitan Area Networks (LANMAN)*. IEEE, April 2013, pp. 1–6. 4, 40, 47, 55, 56, 64, 81, 82, 83, 102

[70] V. C. Cunha, A. Z. Zavala, D. Magoni, P. R. M. Inácio, and M. M. Freire, "A complete review on the application of statistical methods for evaluating internet traffic usage," *IEEE Access*, vol. 10, pp. 128 433–128 455, December 2022. 7, 8, 11

[71] V. C. Cunha, D. Magoni, P. R. M. Inácio, and M. M. Freire, "Impact of self c parameter on svm-based classification of encrypted multimedia peer-to-peer traffic," in *Advanced Information Networking and Applications*, L. Barolli, F. Hussain, and T. Enokido, Eds. Cham: Springer International Publishing, 2022, pp. 180–193. 7, 8, 49, 105

[72] V. C. Cunha, A. A. Zavala, P. R. Inácio, D. Magoni, and M. M. Freire, "Classification of encrypted internet traffic using kullback-leibler divergence and euclidean distance," in *International Conference on Advanced Information Networking and Applications*. Springer, March 2020, pp. 883–897. 7, 9, 39, 45, 54, 63, 85, 87, 102, 105

[73] T. Hruby, "On the design of reliable and scalable networked systems," *Vrije Universiteit Amsterdam. Available online: https://www.cs.vu.nl/ ast/Theses/hruby-thesis.pdf/(accessed on 4 April 2022)*, 2016. 9

[74] V. Subramaniam and E. van der Kouwe, "IMPROVING SOFTWARE FAULT INJECTION," *Vrije Universiteit Amsterdam. Available online: https://www.cs.vu.nl/ ast/Theses/kouwe-thesis.pdf/(accessed on 4 April 2022)*, 2016. 9

[75] S. Bhatt, "Attribute-Based Access and Communication Control Models for Cloud and Cloud-Enabled Internet of Things," Ph.D. dissertation, The University of Texas at San Antonio. Available online:https://www.profsandhu.com/ics/2018%20Smriti%20Bhatt.pdf, 2018. 9

[76] A. Alshehri, "Access Control Models for Cloud-Enabled Internet of Things," Ph.D. dissertation, The University of Texas at San Antonio. Available online:https://www.profsandhu.com/ics/2018%20Asma%20Alshehri.pdf, 2018. 9

[77] J. Charlton, "Inferring Malware Detector Metrics in the Absence of Ground-Truth," Ph.D. dissertation, The University of Texas at San Antonio. Available online: https://www.profsandhu.com/ics/2021%20John%20Charlton.pdf, 2021. 9

[78] D. M. B. Barradas, "Unobservable Multimedia-based Covert Channels for Internet Censorship Circumvention," Ph.D. dissertation, Instituto Superior Técnico (IST), Universidade de Lisboa, Available online: https://www.gsd.inesc-id.pt/ ler/students/diogobarradasphd.html, 2021. 9

[79] A. A. Mohamed, A. H. Osman, and A. Motwakel, "Classification of unknown Internet traffic applications using Multiple Neural Network algorithm," in *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*. IEEE, October 2020, pp. 1–6. 11

[80] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE communications magazine*, vol. 57, no. 5, pp. 76–81, May 2019. 12, 13, 14, 16, 18, 20

[81] J. Zhao, X. Jing, Z. Yan, and W. Pedrycz, "Network traffic classification for data fusion: A survey," *Information Fusion*, vol. 72, pp. 22–47, August 2021. 12, 13, 16, 18

[82] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: A systematic survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1988–2014, November 2018. 12

[83] Y. Peng, M. He, and Y. Wang, "A Federated Semi-Supervised Learning Approach for Network Traffic Classification," *arXiv preprint arXiv:2107.03933*, July 2021. 12

[84] R. Liu and X. Yu, "A Survey on Encrypted Traffic Identification," in *Proceedings of the 2020 International Conference on Cyberspace Innovation of Advanced Technologies*, December 2020, pp. 159–163. 12, 13, 14, 16

[85] S. Valenti, D. Rossi, A. Dainotti, A. Pescapè, A. Finamore, and M. Mellia, "Reviewing traffic classification," in *Data Traffic Monitoring and Analysis*. Springer, October 2013, pp. 123–147. 12, 13, 41, 85

[86] M. Finsterbusch, C. Richter, E. Rocha, J.-A. Muller, and K. Hanssgen, "A survey of payload-based traffic classification approaches," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1135–1156, October 2013. 12, 13, 16, 24

[87] X. Li, "Study on traffic flow base on RBF neural network," in *2014 Sixth International Conference on Measuring Technology and Mechatronics Automation*. IEEE, April 2014, pp. 645–647. 12

[88] P. Mehta and R. Shah, "A survey of network based traffic classification methods," *Database Systems Journal*, vol. 7, no. 4, pp. 24–31, January 2017. 12, 13, 16

[89] K. L. Dias, M. A. Pongelupe, W. M. Caminhas, and L. de Errico, "An innovative approach for real-time network traffic classification," *Computer Networks*, vol. 158, pp. 143–157, 2019. 12, 13, 63, 64, 82

[90] T. Bujlow, V. Carela-Español, and P. Barlet-Ros, "Independent comparison of popular DPI tools for traffic classification," *Computer Networks*, vol. 76, pp. 75–89, January 2015. 13

[91] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Computing*, vol. 24, no. 3, pp. 1999–2012, May 2020. 13

[92] G. Sahoo and Y. Kumar, "Analysis of parametric & non parametric classifiers for classification technique using WEKA," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 4, no. 7, p. 43, July 2012. 13, 26

[93] S. Han, C. Qubo, and H. Meng, "Parameter selection in svm with rbf kernel function," in *World Automation Congress 2012*. IEEE, June 2012, pp. 1–4. 13

[94] Z. Zhang, J. T. Kwok, and D.-Y. Yeung, "Parametric distance metric learning with label information," in *IJCAI*, vol. 1450, January 2003. 13

[95] H. Xu and Y. Deng, "Dependent evidence combination based on shearman coefficient and pearson coefficient," *IEEE Access*, vol. 6, pp. 11 634–11 640, December 2017. 13

[96] M. Menéndez, J. Pardo, L. Pardo, and M. Pardo, "The jensen-shannon divergence," *Journal of the Franklin Institute*, vol. 334, no. 2, pp. 307–318, March 1997. 13, 31

[97] T. Kailath, "The divergence and bhattacharyya distance measures in signal selection," *IEEE transactions on communication technology*, vol. 15, no. 1, pp. 52–60, February 1967. 13

[98] L. Su and H. White, "A nonparametric hellinger metric test for conditional independence," *Econometric Theory*, vol. 24, no. 4, pp. 829–864, April 2008. 13

[99] J. Poza, C. Gómez, M. García, A. Bachiller, A. Fernández, and R. Hornero, "Analysis of spontaneous meg activity in mild cognitive impairment and alzheimer's disease using jensen's divergence," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, August 2014, pp. 1501–1504. 13

[100] F. J. Massey Jr, "The Kolmogorov-Smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951. 13

[101] R. Dunaytsev, D. Moltchanov, Y. Koucheryavy, O. Strandberg, and H. Flinck, "A survey of p2p traffic management approaches: best practices and future directions," *Internet Engineering*, vol. 5, no. 1, pp. 318–330, June 2012. 13, 16

[102] A. Pradhan, "Support vector machine-a survey," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 8, pp. 82–85, August 2012. 13, 16, 35

[103] A. Dainotti, A. Pescape, and K. C. Claffy, "Issues and future directions in traffic classification," *IEEE network*, vol. 26, no. 1, pp. 35–40, January 2012. 13, 16

[104] V. E. SE, "Survey of traffic classification using machine learning," *International journal of advanced research in computer science*, vol. 4, no. 4, April 2013. 13, 16

[105] B. Li, J. Springer, G. Bebis, and M. H. Gunes, "A survey of network flow applications," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 567–581, March 2013. 13, 16

[106] S. Tavara, "Parallel computing of support vector machines: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–38, November 2019. 13, 16, 73

[107] T. Garrett, L. E. Setenareski, L. M. Peres, L. C. Bona, and E. P. Duarte, "Monitoring network neutrality: A survey on traffic differentiation detection," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2486–2517, March 2018. 13, 16

[108] H. Tahaei, F. Afifi, A. Asemi, F. Zaki, and N. B. Anuar, "The rise of traffic classification in IoT networks: A survey," *Journal of Network and Computer Applications*, vol. 154, p. 102538, March 2020. 13, 16

[109] P. Wang, X. Chen, F. Ye, and Z. Sun, "A survey of techniques for mobile service encrypted traffic classification using deep learning," *IEEE Access*, vol. 7, pp. 54 024–54 033, April 2019. 13, 14, 16

[110] S. Alam, S. K. Sonbhadra, S. Agarwal, and P. Nagabhushan, "One-class support vector classifiers: A survey," *Knowledge-Based Systems*, vol. 196, p. 105754, May 2020. 13, 16

[111] J. Yan and J. Yuan, "A survey of traffic classification in software defined networks," in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*. IEEE, August 2018, pp. 200–206. 13, 16

[112] J. Nalepa and M. Kawulok, "Selecting training sets for support vector machines: a review," *Artificial Intelligence Review*, vol. 52, no. 2, pp. 857–900, January 2019. 13, 16

[113] S. Hussain, M. Abualkibash, and S. Tout, "A survey of traffic sign recognition systems based on convolutional neural networks," in *2018 IEEE International Conference on Electro/Information Technology (EIT)*. IEEE, May 2018, pp. 0570–0573. 13

[114] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," 2007. 14

[115] J. Yepes-Nuñez, G. Urrutia, M. Romero-Garcia, and S. Alonso-Fernandez, "The prisma 2020 statement: an updated guideline for reporting systematic reviews." *Revista Espanola de Cardiologia (English ed.)*, vol. 74, no. 9, pp. 790–799, 2021. 14

[116] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan *et al.*, "The prisma 2020 statement: an updated guideline for reporting systematic reviews," *Systematic reviews*, vol. 10, no. 1, pp. 1–11, 2021. 14

[117] Sandvine, "The Global Internet Phenomena Report COVID-19 Spotlight," https://www.sandvine.com/covid-internet-spotlight-report?hsCtaTracking=69c3275d-0a47-4def-b46d-506266477a50%7Cac52173f-34c1-42df-8469-a091e7219e7a, May 2020, may, 2020. 18, 19, 49, 50

[118] M. Tamilkili, "A survey on recent traffic classification techniques using machine learning methods," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 12, pp. 368–373, June 2013. 20

[119] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde, "Analyzing android encrypted network traffic to identify user actions," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 114–125, September 2015. 20

[120] M. Shafiq, X. Yu, A. K. Bashir, H. N. Chaudhry, and D. Wang, "A machine learning approach for feature selection traffic classification using security analysis," *The Journal of Supercomputing*, vol. 74, no. 10, pp. 4867–4892, January 2018. 20, 26, 27

[121] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic classification through simple statistical fingerprinting," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 1, pp. 5–16, January 2007. 21

[122] S. E. Gómez, L. Hernández-Callejo, B. C. Martínez, and A. J. Sánchez-Esguevillas, "Exploratory study on class imbalance and solutions for network traffic classification," *Neurocomputing*, vol. 343, pp. 100–119, May 2019. 24, 80, 81

[123] D. Bzdok, N. Altman, and M. Krzywinski, "Points of significance: statistics versus machine learning," *Nature Methods 2018a*, pp. 1–7, April 2018. 25

[124] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The accuracy of machine learning (ml) forecasting methods versus statistical ones: extending the results of the m3-competition," in *Working Paper, University of Nicosia*. Institute for the Future, July 2017. 25

[125] Z. Chen, Z. Liu, L. Peng, L. Wang, and L. Zhang, "A novel semi-supervised learning method for internet application identification," *Soft Computing*, vol. 21, no. 8, pp. 1963–1975, November 2017. 26

[126] N. Antunes and M. Vieira, "On the metrics for benchmarking vulnerability detection tools," in *2015 45th Annual IEEE/IFIP international conference on dependable systems and networks*. IEEE, June 2015, pp. 505–516. 26, 27

[127] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE signal processing magazine*, vol. 13, no. 4, pp. 67–94, July 1996. 26

[128] G. W. Corder and D. I. Foreman, *Nonparametric statistics: A step-by-step approach*. John Wiley & Sons, April 2014. 26

[129] L. Kruglyak, M. J. Daly, M. P. Reeve-Daly, and E. S. Lander, "Parametric and nonparametric linkage analysis: a unified multipoint approach," *American journal of human genetics*, vol. 58, no. 6, p. 1347, June 1996. 26

[130] S. Boslaugh, *Statistics in a nutshell: A desktop quick reference*. " O'Reilly Media, Inc.", November 2012. 28

[131] S. Glen, "Correlation coefficient: Simple definition, formula, easy steps," *StatisticsHowTo. com. Available online: https://www. statisticshowto. com/probability-and-statistics/correlation-coefficient-formula/(accessed on 3 August 2020)*, August 2021. 28

[132] N. Pandis, "The chi-square test," *American journal of orthodontics and dentofacial orthopedics*, vol. 150, no. 5, pp. 898–899, November 2016. 28, 56

[133] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, July 1948. 28

[134] M. Borda, *Fundamentals in information theory and coding*. Springer Science & Business Media, May 2011. 28

[135] M. Basseville, "Divergence measures for statistical data processing—An annotated bibliography," *Signal Processing*, vol. 93, no. 4, pp. 621–633, April 2013. 28

[136] M. Martin, A. Plastino, and O. Rosso, "Statistical complexity and disequilibrium," *Physics Letters A*, vol. 311, no. 2-3, pp. 126–132, May 2003. 28

[137] C. A. Solà *et al.*, "Recent statistical methods based on distances," *Contributions to Science*, pp. 183–192, August 2002. 28, 29

[138] M. Markatou and E. M. Sofikitou, "Statistical distances and the construction of evidence functions for model adequacy," *Frontiers in Ecology and Evolution*, vol. 7, p. 447, November 2019. 29

[139] F. Nielsen, "On a generalization of the jensen-shannon divergence and the js-symmetrization of distances relying on abstract means," *arXiv preprint arXiv:1904.04017*, February 2019. 30, 31, 33, 86

[140] F. C. Schweppe, "On the Bhattacharyya distance and the divergence between Gaussian processes," *Information and Control*, vol. 11, no. 4, pp. 373–395, October 1967. 31

[141] S. Bi, M. Broggi, and M. Beer, "The role of the Bhattacharyya distance in stochastic model updating," *Mechanical Systems and Signal Processing*, vol. 117, pp. 437–452, February 2019. 31, 86

[142] D. J. Weller-Fahy, B. J. Borghetti, and A. A. Sodemann, "A survey of distance and similarity measures used within network intrusion anomaly detection," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 70–91, July 2014. 31, 86

[143] P. Vaidya and C. M. PVSSR, "Adaptive, robust and blind digital watermarking using bhattacharyya distance and bit manipulation," *Multimedia Tools and Applications*, vol. 77, no. 5, pp. 5609–5635, March 2018. 31

[144] J. Wu and R. J. Karunamuni, "Efficient Hellinger distance estimates for semiparametric models," *Journal of Multivariate Analysis*, vol. 107, pp. 1–23, May 2012. 32, 86

[145] C. Su and J. Cao, "Improving lazy decision tree for imbalanced classification by using skew-insensitive criteria," *Applied Intelligence*, vol. 49, no. 3, pp. 1127–1145, October 2019. 32, 86

[146] V. González-Castro, R. Alaiz-Rodríguez, and E. Alegre, "Class distribution estimation based on the Hellinger distance," *Information Sciences*, vol. 218, pp. 146–164, January 2013. 32

[147] D. A. Cieslak, T. R. Hoens, N. V. Chawla, and W. P. Kegelmeyer, "Hellinger distance decision trees are robust and skew-insensitive," *Data Mining and Knowledge Discovery*, vol. 24, no. 1, pp. 136–158, June 2012. 32

[148] T. M. Cover, *Elements of information theory*. John Wiley & Sons, July 2006. 32, 33, 65, 85

[149] C. Delpha, D. Diallo, and A. Youssef, "Kullback-Leibler Divergence for fault estimation and isolation: Application to Gamma distributed data," *Mechanical Systems and Signal Processing*, vol. 93, pp. 118–135, September 2017. 32, 33, 65, 85

[150] D. J. Galas, G. Dewey, J. Kunert-Graf, and N. A. Sakhanenko, "Expansion of the Kullback-Leibler divergence, and a new class of information metrics," *Axioms*, vol. 6, no. 2, p. 8, April 2017. 32, 33, 65, 85

[151] P. J. Moreno, P. Ho, and N. Vasconcelos, "A kullback-leibler divergence based kernel for svm classification in multimedia applications." in *NIPS*, December 2003, pp. 1385–1392. 33, 34, 64

[152] D. Kvitsiani, S. Ranade, B. Hangya, H. Taniguchi, J. Huang, and A. Kepecs, "Distinct behavioural and network correlates of two interneuron types in prefrontal cortex," *Nature*, vol. 498, no. 7454, pp. 363–366, May 2013. 33

[153] A. M. Kowalski, M. T. Martin, A. Plastino, O. A. Rosso, and M. Casas, "Distances in probability space and the statistical complexity setup," *Entropy*, vol. 13, no. 6, pp. 1055–1075, June 2011. 33

[154] W. Peng, A. Chen, and J. Chen, "Using general master equation for feature fusion," *Future Generation Computer Systems*, vol. 82, pp. 119–126, May 2018. 34, 87

[155] V. Vapnik, I. Guyon, and T. Hastie, "Support vector machines," *Mach. Learn*, vol. 20, no. 3, pp. 273–297, September 1995. 34

[156] T. Marwala, "Support vector machines," *Handbook of Machine Learning, Wold Scientific*, pp. 97–112, December 2018. 34, 51, 73

[157] E. J. C. Suárez, "Tutorial sobre máquinas de vectores soporte (svm)," *Tutorial sobre Máquinas de Vectores Soporte (SVM)*, pp. 1–12, November 2016. 34

[158] C.-C. Liu, Y. Chang, C.-W. Tseng, Y.-T. Yang, M.-S. Lai, and L.-D. Chou, "SVM-based classification mechanism and its application in SDN networks," in *2018 10th International Conference on Communication Software and Networks (ICCSN)*. IEEE, July 2018, pp. 45–49. 34, 40, 46

[159] S. E. N. Fernandes, A. L. Pilastri, L. A. M. Pereira, R. G. Pires, and J. P. Papa, "Learning kernels for support vector machines with polynomial powers of sigmoid," in *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE, August 2014, pp. 259–265. 35

[160] M. Singla and K. Shukla, "Robust statistics-based support vector machine and its variants: a survey," *Neural Computing and Applications*, pp. 1–22, December 2019. 35, 51

[161] M. Zanin, "The reasonable effectiveness of data in ATM," in *Proc. SESAR Innovation Days*, November 2013, pp. 1–5. 37, 38

[162] C. Canali and R. Lancellotti, "Automatic virtual machine clustering based on Bhattacharyya distance for multi-cloud systems," in *Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds*, April 2013, pp. 45–52. 37, 38

[163] M. A. Dinani, P. Ahmadi, and I. Gholampour, "Efficient feature extraction for highway traffic density classification," in *2015 9th Iranian Conference on Machine Vision and Image Processing (MVIP)*. IEEE, February 2015, pp. 14–19. 37, 38

[164] H. Sadreazami, A. Mohammadi, A. Asif, and K. N. Plataniotis, "Distributed-graph-based statistical approach for intrusion detection in cyber-physical systems," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 137–147, September 2017. 37, 38

[165] M. I. Sameen and B. Pradhan, "A two-stage optimization strategy for fuzzy object-based analysis using airborne LiDAR and high-resolution orthophotos for urban road extraction," *Journal of Sensors*, vol. 2017, February 2017. 37, 38

[166] J. B. Baskoro, A. Wibisono, and W. Jatmiko, "Bhattacharyya distance-based tracking: A vehicle counting application," in *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE, October 2017, pp. 439–444. 37, 38

[167] E. Laz, "Optimal cost allocation in centralized and decentralized detection systems using Bhattacharyya distance," in *2017 IEEE Radar Conference (RadarConf)*. IEEE, May 2017, pp. 1170–1173. 37, 38

[168] M. H. Shah and X. Dang, "Novel feature selection method using bhattacharyya distance for neural networks based automatic modulation classification," *IEEE Signal Processing Letters*, vol. 27, pp. 106–110, December 2019. 37, 38

[169] C.-H. Liu, P. Pawelczak, and D. Cabric, "Primary user traffic classification in dynamic spectrum access networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 11, pp. 2237–2251, November 2014. 37, 38

[170] J. Safarik, M. Voznak, F. Rezac, and J. Slachta, "Application of Artificial Intelligence on Classification of Attacks in IP Telephony," *Advances in Information Science and Applications*, vol. II, pp. 373–378, November 2014. 37, 38

[171] Z. Luo, P.-M. Jodoin, S.-Z. Li, and S.-Z. Su, "Traffic analysis without motion features," in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, September 2015, pp. 3290–3294. 37, 38

[172] C. Wang, T. T. Miu, X. Luo, and J. Wang, "SkyShield: A sketch-based defense system against application layer DDoS attacks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 559–573, October 2017. 37, 38

[173] A. Kumari and U. Thakar, "Hellinger distance based oversampling method to solve multi-class imbalance problem," in *2017 7th International Conference on Communication Systems and Network Technologies (CSNT)*. IEEE, November 2017, pp. 137–141. 37, 38

[174] Z. Liu, R. Wang, N. Japkowicz, Y. Cai, D. Tang, and X. Cai, "Mobile app traffic flow feature extraction and selection for improving classification robustness," *Journal of Network and Computer Applications*, vol. 125, pp. 190–208, January 2019. 37, 38

[175] S. Liang, “Feature Extraction of Broken Glass Cracks in Road Traffic Accident Site Based on Deep Learning,” *Complexity*, vol. 2021, May 2021. 38

[176] J. Garcia and T. Korhonen, “Efficient distribution-derived features for high-speed encrypted flow classification,” in *Proceedings of the 2018 Workshop on Network Meets AI & ML*, August 2018, pp. 21–27. 39, 45

[177] M. Zareapoor, P. Shamsolmoali, and M. A. Alam, “Advance DDOS detection and mitigation technique for securing cloud,” *International Journal of Computational Science and Engineering*, vol. 16, no. 3, pp. 303–310, May 2018. 39, 45

[178] T. Zhi, Y. Liu, J. Wang, and H. Zhang, “Resist Interest flooding attacks via entropy–SVM and Jensen–Shannon divergence in information-centric networking,” *IEEE Systems Journal*, vol. 14, no. 2, pp. 1776–1787, September 2019. 39, 45

[179] O. Barut, R. Zhu, Y. Luo, and T. Zhang, “TLS Encrypted Application Classification Using Machine Learning with Flow Feature Engineering,” in *2020 the 10th International Conference on Communication and Network Security*, November 2020, pp. 32–41. 39, 45

[180] J. Chen, D. Wu, Y. Zhao, N. Sharma, M. Blumenstein, and S. Yu, “Fooling intrusion detection systems using adversarially autoencoder,” *Digital Communications and Networks*, August 2020. 39, 45

[181] J. Kim, J. Hwang, and K. Kim, “High-performance internet traffic classification using a markov model and kullback-leibler divergence,” *Mobile Information Systems*, vol. 2016, January 2016. 39, 45

[182] J. Xu, S. Denman, C. Fookes, and S. Sridharan, “Detecting rare events using Kullback–Leibler divergence: A weakly supervised approach,” *Expert Systems with Applications*, vol. 54, pp. 13–28, July 2016. 39, 45

[183] X. Zhang, F. Qiu, and F. Qin, “Identification and mapping of winter wheat by integrating temporal change information and Kullback-Leibler divergence,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 76, pp. 26–39, April 2019. 39, 45

[184] A. Şentaş, İ. Tashiev, F. Küçükayvaz, S. Kul, S. Eken, A. Sayar, and Y. Becerikli, “Performance evaluation of support vector machine and convolutional neural network algorithms in real-time vehicle type and color classification,” *Evolutionary Intelligence*, vol. 13, no. 1, pp. 83–91, August 2020. 39, 46, 50, 51

[185] S. Sankaranarayanan and S. Mookherji, “SVM-based traffic data classification for secured IoT-based road signaling system,” in *Research Anthology on Artificial Intelligence Applications in Security*. IGI Global, November 2021, pp. 1003–1030. 39

[186] J. Cao and Z. Fang, "Network Traffic Classification using Genetic Algorithms based on Support Vector Machine," *International Journal of Security and Its Applications*, vol. 10, no. 2, pp. 237–246, April 2016. 39

[187] M. Sabzekar, M. H. Y. Moghaddam, and M. Naghibzadeh, "Tcp traffic classification using relaxed constraints support vector machines," in *Integration of practice-oriented knowledge technology: Trends and prospectives*. Springer, May 2013, pp. 129–139. 39

[188] V. D'Alessandro, B. Park, L. Romano, C. Fetzer *et al.*, "Scalable network traffic classification using distributed support vector machines," in *2015 IEEE 8th International Conference on Cloud Computing*. IEEE, June 2015, pp. 1008–1012. 40

[189] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network traffic classification using correlation information," *IEEE Transactions on Parallel and Distributed systems*, vol. 24, no. 1, pp. 104–117, March 2012. 40, 47

[190] S. Dong, W. Ding, and L. Chen, "Measure correlation analysis of network flow based on symmetric uncertainty," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 6, no. 6, pp. 1649–1667, June 2012. 40, 47

[191] F. Casino, K.-K. R. Choo, and C. Patsakis, "HEDGE: efficient traffic classification of encrypted and compressed packets," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 11, pp. 2916–2926, April 2019. 40, 47

[192] Y. Wang, Z. Zhang, L. Guo, and S. Li, "Using entropy to classify traffic more deeply," in *2011 IEEE Sixth International Conference on Networking, Architecture, and Storage*. IEEE, July 2011, pp. 45–52. 41, 47

[193] J. V. Gomes, P. R. Inacio, M. Pereira, M. M. Freire, and P. P. Monteiro, "Identification of peer-to-peer voip sessions using entropy and codec properties," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 10, pp. 2004–2014, December 2012. 41, 47, 64, 84

[194] K. Zhou, W. Wang, C. Wu, and T. Hu, "Practical evaluation of encrypted traffic classification based on a combined method of entropy estimation and neural networks," *ETRI Journal*, vol. 42, no. 3, pp. 311–323, January 2020. 41, 47

[195] A. Campazas-Vega, I. S. Crespo-Martínez, Á. M. Guerrero-Higueras, C. Álvarez-Aparicio, and V. Matellán, "Analysis of netflow features' importance in malicious network traffic detection," in *Computational Intelligence in Security for Information Systems Conference*. Springer, September 2021, pp. 52–61. 41

[196] S. Rezvani, X. Wang, and F. Pourpanah, "Intuitionistic fuzzy twin support vector machines," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 11, pp. 2140–2151, January 2019. 42, 43, 50, 51

[197] J. Yang, L. Yuan, C. Dong, G. Cheng, N. Ansari, and N. Kato, "On characterizing peer-to-peer streaming traffic," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 175–188, 2013. 50

[198] K. Pal, M. C. Govil, M. Ahmed, and T. Chawla, "A Survey on Adaptive Multimedia Streaming," in *Recent Trends in Communication Networks*. IntechOpen, 2019. 50

[199] M. E. Mavroforakis and S. Theodoridis, "A geometric approach to support vector machine (SVM) classification," *IEEE transactions on neural networks*, vol. 17, no. 3, pp. 671–682, 2006. 50, 51

[200] R. Yuan, Z. Li, X. Guan, and L. Xu, "An SVM-based machine learning method for accurate internet traffic classification," *Information Systems Frontiers*, vol. 12, no. 2, pp. 149–156, 2010. 50, 51

[201] S. Sankaranarayanan and S. Mookherji, "Svm-based traffic data classification for secured iot-based road signaling system," *International Journal of Intelligent Information Technologies (IJIIT)*, vol. 15, no. 1, pp. 22–50, 2019. 50, 51

[202] W. Han, J. Xue, and H. Yan, "Detecting anomalous traffic in the controlled network based on cross entropy and support vector machine," *IET Information Security*, vol. 13, no. 2, pp. 109–116, March 2019. 50, 51

[203] M. M. Raikar, S. Meena, M. M. Mulla, N. S. Shetti, and M. Karanandi, "Data Traffic Classification in Software Defined Networks (SDN) using supervised-learning," *Procedia Computer Science*, vol. 171, pp. 2750–2759, 2020. 50, 51

[204] F. Budiman, "Svm-rbf parameters testing optimization using cross validation and grid search to improve multiclass classification," *Scientific Visualization*, vol. 11, no. 1, pp. 80–90, 2019. 50, 51

[205] W. Zhongsheng, W. Jianguo, Y. Sen, and G. Jiaqiong, "Traffic identification and traffic analysis based on support vector machine," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 2, p. e5292, 2020. 52

[206] K.-B. Duan and S. S. Keerthi, "Which is the best multiclass SVM method? An empirical study," in *International workshop on multiple classifier systems*. Springer, 2005, pp. 278–285. 52

[207] J. Velasco-Mata, E. Fidalgo, V. González-Castro, E. Alegre, and P. Blanco-Medina, "Botnet Detection on TCP Traffic Using Supervised Machine Learning," in *International Conference on Hybrid Artificial Intelligence Systems*. Springer, 2019, pp. 444–455. 52

[208] S. L. user guide Release 0.21.2, https://scikit-learn.org/stable/_downloads/scikit-learn-docs.pdf, May 2020, may 09, 2020. 53, 73

[209] Vanice-ufmt, https://github.com/Vanice-ufmt/Codigo, october 2020, october 30, 2020. 54

[210] C. R. C. Reports, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html, april 2020, april 28, 2020. 54

[211] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011. 55

[212] M. Zhang, W. John, K. Claffy, and N. Brownlee, "State of the art in traffic classification: A research review," in *PAM Student Workshop*, 2009, pp. 3–4. 63

[213] J. Zhang, Z. Li, Z. Pu, and C. Xu, "Comparing prediction performance for crash injury severity among various machine learning and statistical methods," *IEEE Access*, vol. 6, pp. 60 079–60 087, 2018. 64, 82

[214] Y. Xiang, K. Li, and W. Zhou, "Low-rate DDoS attacks detection and traceback by using new information metrics," *IEEE transactions on information forensics and security*, vol. 6, no. 2, pp. 426–437, 2011. 64

[215] A. Tongaonkar, R. Torres, M. Iliofotou, R. Keralapura, and A. Nucci, "Towards self adaptive network traffic classification," *Computer Communications*, vol. 56, pp. 35–46, 2015. 64, 81, 82, 83

[216] D. Li, G. Hu, Y. Wang, and Z. Pan, "Network traffic classification via non-convex multi-task feature learning," *Neurocomputing*, vol. 152, pp. 322–332, 2015. 64, 82, 83

[217] L. Peng, H. Zhang, Y. Chen, and B. Yang, "Imbalanced traffic identification using an imbalanced data gravitation-based classification model," *Computer Communications*, vol. 102, pp. 177–189, 2017. 64, 80, 82, 84

[218] J. V. Gomes, P. R. Inácio, M. M. Freire, M. Pereira, and P. P. Monteiro, "Analysis of peer-to-peer traffic using a behavioural method based on entropy," in *Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International*. IEEE, 2008, pp. 201–208. 64

[219] F. Ertam and E. Avcı, "A new approach for internet traffic classification: GA-WK-ELM," *Measurement*, vol. 95, pp. 135–142, 2017. 65, 82, 83

[220] . L. PSrecord, https://pypi.org/project/psrecord/, April 2020, april 28, 2020. 75, 105

[221] X. Tan, Y. Xie, H. Ma, S. Yu, and J. Hu, "Recognizing the content types of network traffic based on a hybrid DNN-HMM model," *Journal of Network and Computer Applications*, vol. 142, pp. 51–62, 2019. 80

[222] A. Molavi Kakhki, A. Razaghpanah, R. Golani, D. Choffnes, P. Gill, and A. Mislove, "Identifying traffic differentiation on cellular data networks," in *Proceedings of the 2014 ACM conference on SIGCOMM*, 2014, pp. 119–120. 80

[223] M. B. Attia, K.-K. Nguyen, and M. Cheriet, "QoS-aware software-defined routing in smart community network," *Computer Networks*, vol. 147, pp. 221–235, 2018. 80

[224] S. Hosseini and B. M. H. Zade, "New hybrid method for attack detection using combination of evolutionary algorithms, SVM, and ANN," *Computer Networks*, p. 107168, 2020. 80

[225] T. I. T. Report, http://www.internettrafficreport.com/., Jul 2020, july 09, 2020. 80

[226] C. annual internet report (2018–2023), https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, Jul 2020, july 09, 2020. 80

[227] D. Wang, L. Zhang, Z. Yuan, Y. Xue, and Y. Dong, "Characterizing application behaviors for classifying p2p traffic," in *2014 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2014, pp. 21–25. 80

[228] Z. Nascimento and D. Sadok, "MODC: a pareto-optimal optimization approach for network traffic classification based on the divide and conquer strategy," *Information*, vol. 9, no. 9, p. 233, 2018. 80

[229] S. A. Abdullah and A. S. Al-Hashmi, "TiSEFE: Time Series Evolving Fuzzy Engine for Network Traffic Classification," *International Journal of Communication Networks and Information Security*, vol. 10, no. 1, pp. 116–124, 2018. 80

[230] J. Cao, D. Wang, Z. Qu, H. Sun, B. Li, and C.-L. Chen, "An Improved Network Traffic Classification Model Based on a Support Vector Machine," *Symmetry*, vol. 12, no. 2, p. 301, 2020. 80

[231] A. Madhukar and C. Williamson, "A longitudinal study of P2P traffic classification," in *14th IEEE International Symposium on Modeling, Analysis, and Simulation*. IEEE, 2006, pp. 179–188. 80

[232] A. Finamore, M. Mellia, M. Meo, and D. Rossi, "Kiss: Stochastic packet inspection classifier for udp traffic," *IEEE/ACM Transactions on Networking*, vol. 18, no. 5, pp. 1505–1515, 2010. 80

[233] G. Li and S. J. Qin, "Comparative study on monitoring schemes for non-Gaussian distributed processes," *Journal of Process Control*, vol. 67, pp. 69–82, 2018. 80, 81

[234] M. Chari, H. Srinidhi, and T. E. Somu, "Network Traffic Classification by Packet Length Signature Extraction," in *2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*. IEEE, 2019, pp. 1–4. 81, 82, 83

[235] R. Holanda Filho, M. F. F. do Carmo, J. E. B. Maia, and G. P. Siqueira, "An internet traffic classification methodology based on statistical discriminators," in *NOMS 2008-2008 IEEE Network Operations and Management Symposium*. IEEE, 2008, pp. 907–910. 81, 82

[236] Z. A. Shaikh and D. G. Harkut, "A Novel Framework for Network Traffic Classification Using Unknown Flow Detection," in *2015 Fifth International Conference on Communication Systems and Network Technologies*. IEEE, 2015, pp. 116–121. 81, 82

[237] E. Nazarenko, V. Varkentin, and T. Polyakova, "Features of Application of Machine Learning Methods for Classification of Network Traffic (Features, Advantages, Disadvantages)," in *2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*. IEEE, 2019, pp. 1–5. 81

[238] R. Raveendran and R. R. Menon, "A novel aggregated statistical feature based accurate classification for internet traffic," in *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*. IEEE, 2016, pp. 225–232. 82, 84

[239] X. Tan, W. Xu, K. Sun, Y. Xu, Y. Be'ery, X. You, and C. Zhang, "Improving massive MIMO message passing detectors with deep neural network," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1267–1280, 2019. 82

[240] N. Seddigh, B. Nandy, D. Bennett, Y. Ren, S. Dolgikh, C. Zeidler, J. Knoetze, and N. S. Muthyala, "A Framework & System for Classification of Encrypted Network Traffic using Machine Learning," in *2019 15th International Conference on Network and Service Management (CNSM)*. IEEE, 2019, pp. 1–5. 82

[241] V. Labayen, E. Magaña, D. Morató, and M. Izal, "Online classification of user activities using machine learning on network traffic," *Computer Networks*, vol. 181, p. 107557, 2020. 82, 84

[242] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying IoT devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, 2018. 84

[243] M. Bhatia and M. K. Rai, "Identifying P2P traffic: A survey," *Peer-to-Peer Networking and Applications*, vol. 10, no. 5, pp. 1182–1203, 2017. 84

[244] Q. Wang, J. Wan, and Y. Yuan, "Locality constraint distance metric learning for traffic congestion detection," *Pattern Recognition*, vol. 75, pp. 272–281, 2018. 85

[245] S. Venkatraman and M. Alazab, "Use of data visualisation for zero-day malware detection," *Security and Communication Networks*, vol. 2018, 2018. 85

[246] P. Vermeesch, "Dissimilarity measures in detrital geochronology," *Earth-Science Reviews*, vol. 178, pp. 310–321, 2018. 85

[247] L. Pardo, *Statistical inference based on divergence measures*. CRC press, 2018. 85

[248] J. Harmouche, C. Delpha, D. Diallo, and Y. Le Bihan, "Statistical approach for nondestructive incipient crack detection and characterization using Kullback-Leibler divergence," *IEEE Transactions on Reliability*, vol. 65, no. 3, pp. 1360–1368, 2016. 85

[249] M. Ring, D. Schlör, D. Landes, and A. Hotho, "Flow-based network traffic generation using generative adversarial networks," *Computers & Security*, vol. 82, pp. 156–172, 2019. 85

[250] A. Rahul, S. Prashanth *et al.*, "Detection of intruders and flooding in VoIP using ids, jacobson fast and hellinger distance algorithms," *American Journal of Advanced Computing*, vol. 1, no. 1, pp. 1–6, 2020. 86

[251] "Dataset-UBI", https://www.it.ubi.pt/~mario/IT_Database.zip, February 05, 2022, february 05. 87

[252] "Classifier distance Vanice", https://github.com/vanicecunha/classifier_distance_Vanice, February 08, 2022, february 08. 89

[253] I. A. Vergara, T. Norambuena, E. Ferrada, A. W. Slater, and F. Melo, "StAR: a simple tool for the statistical comparison of ROC curves," *BMC bioinformatics*, vol. 9, no. 1, pp. 1–5, 2008. 100

[254] C. Zhang, H. Wang, and R. Fu, "Automated detection of driver fatigue based on entropy and complexity measures," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 168–177, 2013. 100

[255] J. Yang, Y. Wang, C. Dong, and G. Cheng, "The evaluation measure study in network traffic multi-class classification based on AUC," in *2012 International Conference on ICT Convergence (ICTC)*. IEEE, 2012, pp. 362–367. 100

[256] H. Jiang and H. Deng, "Traffic Incident Detection Method Based on Factor Analysis and Weighted Random Forest," *IEEE Access*, vol. 8, pp. 168 394–168 404, 2020. 101

[257] B. T. Pham, I. Prakash, S. K. Singh, A. Shirzadi, H. Shahabi, D. T. Bui *et al.*, "Landslide susceptibility modeling using Reduced Error Pruning Trees and different ensemble techniques: Hybrid machine learning approaches," *Catena*, vol. 175, pp. 203–218, 2019. 101