Proyecto de investigación
Clasificación por Machine Learning de flujos de tráfico emulado

Responsable:
Dr. Jorge Ricardo Gómez Montalvo[a]
Estudiante:
Lic. Daniel Sánchez Ferriz[a]
Colaboradores:
Dr. Ernesto Exposito[b]
Dr. Jorge Alberto Ríos Martínez[a]

[a] Facultad de Matemáticas, Universidad Autónoma de Yucatán, México
[b] Université de Pau et des Pays de l'Adour, Francia

1. Introducción

El uso de Internet aumenta cada año, según las estadísticas de la Unión Internacional de las Telecomunicaciones, el ancho de banda internacional del Internet creció un 32 % entre el 2015 y el 2016 [1]. Este crecimiento facilita a que haya una mayor diversidad y demanda de servicios de transporte de información y comunicación. A su vez, tales servicios deben satisfacer las expectativas de calidad de funcionamiento del usuario, lo cual es estudiado por el área de Calidad de Servicio [2].

La Calidad de Servicio (conocida como QoS [3] por sus siglas en inglés *Quality of Service*[2]) se define en la recomendación *Terms and definitions related to the quality of telecommunication services* ITU-T E.800 [4] como "la totalidad de las características de un servicio de telecomunicaciones que determinan su capacidad para satisfacer las necesidades explícitas e implícitas del usuario del servicio". Al hablar de un entorno que ofrezca QoS, se tiene un servicio de transporte de información y comunicación cuyas características cumplan con un criterio de calidad [5].

Una de las técnicas utilizadas para cumplir con los requerimientos de QoS de las aplicaciones del usuario es la priorización del envío de paquetes de información [5]; es decir, a cada paquete de información se le asigna un nivel de prioridad y acorde a éste, es transmitido a través de la red hasta llegar al destino de manera que cumpla con los requerimientos establecidos para la QoS respectiva, en consecuencia, se proporciona al usuario final un servicio con calidad. La priorización de paquetes, con el fin de proveer QoS, requiere clasificación de tráfico de redes.

La clasificación de tráfico de redes es un proceso administrado en diferentes nodos, a través de la capa IP (por sus siglas en inglés *Internet Protocol*) de una red, donde se utilizan clasificadores de paquetes con el fin de diferenciar y hacer cumplir los requerimientos de la QoS. Los clasificadores de paquetes seleccionan paquetes en un flujo de tráfico, con base en el contenido del encabezado del paquete [6]. En la recomendación *Traffic Classification and Quality of Service (QoS) Attributes for Diameter* RFC.5777 [7] se plantean a los clasificadores de tráfico de redes como herramientas para la QoS: "si un paquete coincide con un patrón, dicho paquete será tratado con respecto a la especificación de QoS asociada".

En la revisión de [8] se mencionan 3 tipos de clasificación de tráfico de redes: por número de puerto de los protocolos TCP/UDP, por Análisis Profundo de Paquetes (DPI, por sus siglas en inglés) y por Machine Learning. De igual manera, otra técnica de clasificación es por análisis estadístico, como se usó en [9] y [10]. La clasificación por número de puerto de los protocolos TCP/UDP, a pesar de ser considerada como la clasificación más sencilla y rápida [11], es poco precisa [12], alternativamente, la clasificación por DPI ofrece mayor precisión a cambio de costo elevado en término de procesamiento [13], problemas al calisifcar tráfico de redes encriptado [14] y problemas legales en temas de privacidad [8]. Se ha demostrado que la clasificación por Machine Learning es viable en términos de costo computacional y precisión [15] [16] [17], comparable a los resultados de los clasificadores más precisos [16] sin necesidad de inspeccionar la carga útil de los paquetes [17] respetando así las leyes de privacidad [18].

El mayor problema que enfrenta la clasificación de tráfico de redes por Machine Learning es la falta de *ground truth* [8]. *Ground truth* es un conjunto de datos donde existe una relación entre atributos independientes y un atributo dependiente (datos etiquetados) [19] [20]. Una solución podría involucrar el manejo de herramientas de emulación de tráfico, pero herramientas como IP-TNE, Vint/NS Simulator y Swing emulan tráfico a partir de flujos dados [21] [22] [23]. Una alternativa que no requiere un flujo previamente generado es la utilización de una plataforma de *cloud computing* capaz de crear una red virtual, compuesta por máquinas virtuales que generen flujos reales.

En este proyecto de investigación se propone el desarrollo de una plataforma en la nube capaz de emular el tráfico de una red de Internet. Con base en esta plataforma, se emulan flujos de tráfico mediante los cuales se observan características significativas de las trazas para formar una base de datos relevante al tráfico. A partir de esta base de datos, se entrenan modelos preestablecidos de Machine Learning para clasificar el tráfico de flujo y se reportan los resultados.

2. Contexo y problemática

En la literatura de clasificación de tráfico de redes se manejan 4 tipos de clasificadores: por número de puerto de los protocolos TCP/UDP, por Análisis Profundo de Paquetes (DPI, por sus siglas en inglés), por análisis estadístico y por Machine Learning. La clasificación por puertos es la de menor precisión (<70 % [12]). La clasificación por DPI falla cuando la carga útil está encriptada [14], debido a esto, [11] cataloga a este tipo de clasificación como "poco realista" debido a la gran cantidad de información encriptada usada en aplicaciones como Skype y servicios P2P [24]. En relación al aspecto legal, la clasificación por *payload* puede ser problemática debido a las leyes de privacidad que pueden prevenir el acceso o el almacenamiento del contenido de los paquetes [8]. La clasificación con acercamientos estadísticos asume que la capa de red tiene propiedades estadísticas (como la distribución de la duración del flujo, el tiempo de inactividad del flujo, el tiempo de paquetes entre llegadas y las longitudes de los paquetes) que son exclusivas para ciertas clases de aplicaciones y permiten distinguir diferentes aplicaciones fuente entre sí [25]. La clasificación de tráfico de redes por Machine Learning no depende de los números de puertos y puede trabajar con datos encriptados [17].

La precisión de Machine Learning como clasificador de tráfico de redes se puede observar en el trabajo de [17], donde se manejó un algoritmo C4.5 y se clasificó correctamente el 99.8 % del tráfico. En la investigación de [15] hubo una precisión promedio de 99.3 % al 99.9 %. En [16] se compararon las herramientas de clasificación por DPI PACE, OpenDPI, NDPI, Libprotoident, NBAR, cuatro variantes de L7 Filter y un clasificador por *Machine Learning* basado en el algoritmo C5.0. El clasificador comparado no realiza DPI, pero fue entrenado con datos clasificados previamente con PACE. En dicho trabajo, se manejaron dos colecciones de datos donde cada aplicación estaba categorizada por una clase: en la primera colección había una gran variación entre el número de flujos por cada clase de aplicación; en la segunda colección el número de flujos por clase de aplicación estaba distribuido uniformemente. Durante la comparación se observó que, en la colección de datos donde había gran variación entre el número de flujos por clase de aplicación, el clasificador por *Machine Learning* catalogó correctamente un 98.45 % del tráfico, superando a la clasificación por PACE, la cual obtuvo una precisión de 93.5 % siendo la herramienta de clasificación por DPI más precisa en este caso. Cabe mencionar que cuando el número de flujos por clase de aplicación es uniforme, el algoritmo de *Machine Learning* clasificó correctamente sólo un 60.91 % del tráfico, el cual es inferior al 82.73 % de NDPI, el cual fue el mejor clasificador en esta clase.

Uno de los problemas más grandes que enfrenta la clasificación de tráfico de redes por *Machine Learning* es la creación de *ground truth* [8]. En [21] se discute cómo algunas herramientas de clasificación por DPI son comúnmente utilizadas en la literatura como generadoras de *ground truth* e incluso señala que la mejor de estas herramientas para este propósito son PACE, NDPI y Libprotoident, en ese orden, mientras que L7 filter es menos confiable y no debería usarse para este fin. Sin embargo, un *ground truth* obtenido por una clasificación de tráfico por DPI contiene los fallos de este tipo de clasificación, por lo que la comparación entre propuestas de investigación es difícil [26]. Para tratar el tema con base en el contexto y la problemática que esta tesis aborda, se proponen los siguientes objetivos.

3. Objetivos

3.1. Objetivo general

Identificar el nombre de una aplicación a través de un clasificador de Machine Learning entrenado con flujos reales emulados en una plataforma de *cloud computing*.

3.2. Objetivos específicos

- Desarrollar una plataforma que genere y monitoree tráfico de flujo. Se cuenta con una plataforma que consiste en 3 máquinas virtuales en una red, donde una es el *gateway* y captura el tráfico generado. La plataforma debe contar con una arquitectura que utilice los elementos de la red para generar los flujos requeridos para el entrenamiento.
- Emular comportamiento de usuario y aplicación. A partir de una arquitectura en la red de máquinas virtuales, se deben implementar herramientas que emulen actividades (escenarios) de usuarios para generar tráfico real.
- Monitorear todos los flujos y todas las características observables. Los datos capturados son almacenados en una base de datos para su monitoreo constante y análisis.
- Usar algoritmos de Machine Learning para el tráfico de análisis. A partir de la base de datos generada se seleccionan las características de los flujos para el entrenamiento de los algoritmos de Machine Learning, se seleccionan modelos de Machine Learning que han sido utilizados en la clasificación de tráfico de redes, se entrenan y se miden sus precisiones.

4. Metodología

Dados los objetivos específicos, se proponen los siguientes puntos.

- Desarrollar una plataforma que genere y monitoree tráfico de flujos.
  - Proponer una arquitectura para emular escenarios reales de tráfico de red. A partir de la plataforma donde existen 3 máquinas virtuales, se debe diseñar una arquitectura capaz de generar automáticamente flujos de tráfico en el *gateway* con base en escenarios propuestos.
- Emular comportamiento de usuario y aplicación.
  - Implementar escenarios en una plataforma de monitoreo. Diseñar escenarios comparables a los de los usuarios. Al implementarse cada escenario, su flujo respectivo es observado en el *gateway* de la red.
  - Adquirir y almacenar las características observables propuestas en los escenarios experimentales en una base de datos. Con base en los archivos de tráfico capturado en el *gateway* se administra una base de datos tomando en cuenta las características observables que podrán ser usadas en los algoritmos de Machine Learning.
- Monitorear todos los flujos y todas las características observables.
  - Manejar y analizar la base de datos. Se debe consolidar la información en la base de datos para administrar los datos obtenidos de los distintos flujos de tráfico para mejorar el entrenamiento de los algoritmos de Machine Learning.
- Usar algoritmos de Machine Learning para el análisis de tráfico de flujo.
  - Entrenar y probar modelos de Machine Learning para la clasificación de tráfico. Se usan diversos algoritmos de Machine Learning como C4.5, *Random Forest* y SVM para determinar cuál es el de mayor precisión.

4.1. Herramientas

Las herramientas a usar para llevar a cabo esta metodología son:

- Para la plataforma en la nube se usará *Proxmox*, una plataforma para virtualización con interfaz web incorporada que puede administrar máquinas virtuales y redes definidas por software [27]
- La emulación del comportamiento del usuario y la aplicación será realizada mediante la herramienta de automatización de aplicaciones web *Selenium*
- El monitoreo de tráfico será realizado con la *API* de *python*
- El tratamiento de datos y análisis con ML será manejado por *Pentaho*

5. Trabajos previos relacionados

Existen distintos tipos de algoritmos de Machine Learning, por lo que se ha investigado el uso de los algoritmos como clasificadores de tráfico de redes. [28] implementa el manejo de de herramientas de *clustering* para la clasificación de tráfico con datos de la capa de transporte, el tráfico usado como entrenamiento consistió en flujos generados en el enlace a Internet de la Universidad de Calgary durante una hora (60 GB) y una traza pública de la Universidad de Auckland. En [17] se usó un algoritmo C4.5 que clasificó con un 99.8 % de precisión, las trazas recolectadas para el entrenamiento fueron recolectadas durante dos días semanales consecutivos de tráfico de Internet con un intervalo de 8 meses en la Universidad de Cambridge. En [15] los datos de entrenamiento fueron obtenidos mediante la asociación de nombre de aplicación y flujos, la cual fue obtenida mediante un sistema de voluntarios que consistía en instalar clientes en sus computadoras y en un servidor responsable del almacenamiento de los datos recolectados, se usó un algoritmo C5.0 con una precisión promedio de 99.3 % al 99.9 %.

En [29] se usó *Support Vector Machine* (abreviado como SVM) para clasificar 7 diferentes clases de aplicaciones con una precisión de 96.9 % cuando los datos no están sesgados (es decir, existe el mismo número de flujos por cada clase de aplicación) y 99.4 % cuando hay sesgo, los datos usados fueron obtenidos de un enrutador de la universidad durante un total de 8 horas en un período de una semana. En [30] se reconoce el protocolo de aplicación responsable de enviar paquetes a través de un nodo de monitoreo, su análisis es por SVM obteniendo una precisión de 92.37 % en los datos recolectados del enrutador de la facultad.

En [31] se compararon 179 clasificadores de tráfico de redes de 17 familias diferentes (análisis discriminante, Bayes, redes neuronales, SVM, árboles de decisión, clasificadores basados en reglas, *boosting*, *stacking*, *bagging*, *random forests* y otros conjuntos, modelos lineales generalizados, vecinos más cercanos, mínimos cuadrados parciales, regresión de componentes principales, regresión logística y multinomial, *splines* de regresión adaptativa múltiple y otros métodos) en 121 conjuntos de datos. Se obtuvo que los clasificadores con mayor probabilidad de ser los mejores son *random forest* y SVM, donde el primero tuvo una precisión de 94.1 % y el segundo una de 92.3 %. De las 121 colecciones de datos, 117 fueron obtenidas del repositorio de Machine Learning de la UCI y 4 fueron colecciones de casos reales acerca de la estimación de la fecundidad para la pesca.

[32] propone una mejora a la clasificación por *random forest* de tráfico de redes que consiste en otorgar una probabilidad de selección a cada variable según su prioridad de clasificación. Se entrenó con trazas de la Universidad de Beihang, las cuales se recolectaron en su campus. Se comparó la clasificación de tráfico de redes por *random forest* estándar y con la versión propuesta y se obtuvo una mejora en precisión promedio de 96.846 % a 96.999 %.

En [33] se manejan algoritmos de Machine Learning no-supervisados para formar clústers de flujos sin etiquetas a tiempo real (clasfica acorde a los primeros paquetes TCP de una comunicación) con un 95 % de precisión, sus datos de entrenamiento los obtuvieron de otra universidad. En [34] se clasifica a tiempo real pero con algoritmos semi-supervisados obteniendo una precisión del 94 %.

La Tabla 1 resume trabajos de Machine Learning como clasificador, qué tipo de algoritmo fue usado, cuál fue su precisión y cómo se obtuvo el *ground truth*.

Tabla 1: Trabajos de clasificadores de Machine Learning

Trabajo	**Resumen**	**Precisión**	**Ground Truth**
Erman et al. [28]	*Clustering* en la capa de transporte	>95 % en promedio	Flujos generados en el enlace a Internet de la Universidad de Calgary
Li and Moore [17]	C4.5	99.8 %	Trazas recolectadas en la Universidad de Cambridge
Bujlow et al. [15]	C5.0	de 99.3 % al 99.9 %	Sistema de voluntarios
Bujlow et al. [16]	C5.0	98.45 % [1] y 60.91 % [2]	Sistema de voluntarios
Li et al. [29]	SVM	99.4 % [3] y 96.9 % [4]	Enrutador de la universidad
Este et al. [30]	SVM	92.37 %	Datos recolectados del enrutador de la facultad
Fernández-Delgado et al. [31]	Comparación de diversos clasificadores	ver nota[5]	117 casos de estudio tomados del repositorio de UCI y 4 casos de una investigación
Wang et al. [32]	*Random forest*	96.999 %	Trazas de la Universidad de Beihang
Erman et al. [33]	ML No-supervisado	95 %	Trazas de universidad
Erman et al. [34]	ML Semi-supervisado	94 %	Trazas de universidad

[1] Mismo número de flujos por clase de aplicación
[2] Diferente número de flujos por clase de aplicación
[3] Ver nota 1
[4] Ver nota 2
[5] Concluye que los mejores clasificadores pueden ser por *random forest* o SVM con precisiones de 94.1 % y 92.3 % respectivamente

6. Calendario de actividades

La siguiente tabla 2 define las actividades que se realizarán en este proyecto.

Tabla 2: Definición de actividades

Número de actividad	**Definición**
Act 1	Proponer una arquitectura para emular escenarios reales de tráfico de red
Act 2	Implementar escenarios en una plataforma de monitoreo
Act 3	Adquirir y almacenar las características observables propuestas en los escenarios experimentales en una base de datos
Act 4	Manejar y analizar la base de datos
Act 5	Entrenar y probar modelos de Machine Learning para la clasificación de tráfico
Act 6	Entregar resultados

La Tabla 3 muestra el calendario para las actividades anteriormente propuestas.

Tabla 3: Calendario de actividades

Actividad	Ago'18	Sep'18	Oct'18	Nov'18	Dic'18	Ene'19	Feb'19	Mar'19	Abr'19	May'19	Jun'19	Jul'19
Act 1	X	X	X	-	-	-	-	-	-	-	-	-
Act 2	-	-	X	X	X	X	X	-	-	-	-	-
Act 3	-	-	-	X	X	X	X	-	-	-	-	-
Act 4	-	-	-	-	-	-	-	X	-	-	-	-
Act 5	-	-	-	-	-	-	-	-	X	X	X	-
Act 6	-	-	-	-	-	-	-	-	-	-	-	X

Referencias

[1] Telecommunication Standardization Sector of ITU. Ict facts and figures, 2018. URL `https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2017.pdf`.

[2] Telecommunication Standardization Sector of ITU. Telephone network and isdn. quality of service, network management and traffic engineering. terms and definitions related to quality of service and network performance including dependability. Recommendation E.800, International Telecommunication Union, Geneva,Switzerland, October 1994.

[3] Telecommunication Standardization Sector of ITU. Series g: Transmission systems and media, digital systems and networks quality of service and performance communications quality of service: A framework and definitions. Recommendation G.1000, International Telecommunication Union, Geneva,Switzerland, November 2001.

[4] Telecommunication Standardization Sector of ITU. Series e: Overall network operation, telephone service, service operation and human factors quality of telecommunication services: concepts, models, objectives and dependability planning – terms and definitions related to the quality of telecommunication services. Recommendation E.800, International Telecommunication Union, Geneva,Switzerland, September 2008.

[5] Telecommunication Standardization Sector of ITU. Series e: Overall network operation, telephone service, service operation and human factors quality of telecommunication services: concepts, models, objectives and dependability planning – terms and definitions related to the quality of telecommunication services. definitions of terms related to quality of service. Recommendation E.800, International Telecommunication Union, Geneva,Switzerland, November 2008.

[6] Steven Blake, David L. Black, Mark A. Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss. An architecture for differentiated services. RFC 2475, RFC Editor, December 1998. URL `http://www.rfc-editor.org/rfc/rfc2475.txt`.

[7] J. Korhonen, H. Tschofenig, M. Arumaithurai, and A. Lior. Traffic classification and quality of service (qos) attributes for diameter. RFC 5777, RFC Editor, February 2010. URL `https://www.rfc-editor.org/rfc/rfc5777.txt`.

[8] Alberto Dainotti, Antonio Pescape, and Kimberly Claffy. Issues and future directions in traffic classification. *IEEE Network*, 26(1):35–40, January 2012. ISSN 0890-8044. doi: 10.1109/MNET.2012.6135854.

[9] Manuel Crotti, Maurizio Dusi, Francesco Gringoli, and Luca Salgarelli. Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Computer Communication Review*, 37(1):5, January 2007. doi: 10.1145/1198255.1198257. URL `https://doi.org/10.1145/1198255.1198257`.

[10] Matthew Roughan, Subhabrata Sen, Oliver Spatscheck, and Nick Duffield. Class-of-service mapping for qos: A statistical signature-based approach to ip traffic classification. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement - IMC '04*. ACM Press, 2004. doi: 10.1145/1028788.1028805. URL `https://doi.org/10.1145/1028788.1028805`.

[11] Laurent Bernaille, Renata Teixeira, Ismael Akodkenou, Augustin Soule, and Kave Salamatian. Traffic classification on the fly. *ACM SIGCOMM Computer Communication Review*, 36(2): 23, apr 2006. doi: 10.1145/1129582.1129589.

[12] Holger Dreger, Anja Feldmann, Michael Mai, Vern Paxson, and Robin Sommer. Dynamic Application-Layer Protocol Analysis for Network Intrusion Detection. In *15th USENIX Security Symposium*, pages 257–272. Usenix, 2006.

[13] Niccolò Cascarano, Luigi Ciminiera, and Fulvio Risso. Improving cost and accuracy of dpi traffic classifiers. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 641–646. ACM, 2010.

[14] Thomas Karagiannis, Andre Broido, Michalis Faloutsos, and Kc claffy. Transport layer identification of p2p traffic. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement - IMC '04*. ACM Press, 2004. doi: 10.1145/1028788.1028804.

[15] T. Bujlow, T. Riaz, and J. M. Pedersen. A method for classification of network traffic based on c5.0 machine learning algorithm. In *2012 International Conference on Computing, Networking and Communications (ICNC)*, pages 237–241, Jan 2012.

[16] Tomasz Bujlow, Valentin Carela-espa, and Pere Barlet-ros. Comparison of Deep Packet Inspection (DPI) Tools for Traffic Classification Technical Report Department of Computer Architecture (DAC). Technical report, Universitat Politècnica de Catalunya, Department of Computer Architecture (DAC), June 2013.

[17] W. Li and A. W. Moore. A machine learning approach for efficient traffic classification. In *2007 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pages 310–317, October 2007. doi: 10.1109/MASCOTS.2007.2.

[18] Douglas C. Sicker, Paul Ohm, and Dirk Grunwald. Legal issues surrounding monitoring during network research. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 141–148, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-908-1. doi: 10.1145/1298306.1298307. URL `http://doi.acm.org/10.1145/1298306.1298307`.

[19] Andrew Ng. Cs229: Machine learning - lecture notes, 2017. URL `http://cs229.stanford.edu/syllabus.html`.

[20] Scott Krig. *Computer Vision Metrics: Survey, Taxonomy, and Analysis*. Apress, 2014.

[21] Russell Bradford, Rob Simmonds, and Brian Unger. Packet reading for network emulation. In *MASCOTS 2001, Proceedings Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 150–157. IEEE Comput. Soc, 2001. ISBN 0-7695-1315-8. doi: 10.1109/MASCOT.2001.948864. URL `xxxhttp://ieeexplore.ieee.org/document/948864/`.

[22] Kevin Fall. Network emulation in the vint/ns simulator. *Computers and Communications, 1999. Proceedings. IEEE International Symposium on*, pages 244–250, 1999. ISSN 0-7695-0250-4. doi: 10.1109/ISCC.1999.780820.

[23] Kashi Venkatesh Vishwanath and Amin Vahdat. Swing: Realistic and responsive network traffic generation. *IEEE/ACM Transactions on Networking*, 17(3):712–725, 2009. ISSN 10636692. doi: 10.1109/TNET.2009.2020830.

[24] Petr Velan, Milan Čermák, Pavel Čeleda, and Martin Drašar. A survey of methods for encrypted traffic classification and analysis. *International Journal of Network Management*, 25(5): 355–374, sep 2015. ISSN 10557148. doi: 10.1002/nem.1901. URL `http://onlinelibrary.wiley.com/doi/10.1002/nem.604/abstracthttp://doi.wiley.com/10.1002/nem.1901`.

[25] Thuy T. T. Nguyen and Grenville Armitage. A survey of techniques for internet traffic classification using machine learning. *Communications Surveys & Tutorials, IEEE*, 10(4):56–76, 2008. ISSN 1553-877X. doi: 10.1109/SURV.2008.080406.

[26] Valentín Carela-Español, Tomasz Bujlow, and Pere Barlet-Ros. Is our ground-truth for traffic classification reliable. In Michalis Faloutsos and Aleksandar Kuzmanovic, editors, *Passive and Active Meassurement*, volume 8362 of *Lecture Notes in Computer Science*, pages 98–108, Cham, 2014. Springer International Publishing. ISBN 978-3-319-04917-5. doi: 10.1007/978-3-319-04918-2. URL `http://link.springer.com/10.1007/978-3-319-04918-2`.

[27] Proxmox. Proxmox Open-Source Virtualization Platform, 2018. URL `https://www.proxmox.com/en/proxmox-ve`.

[28] Jeffrey Erman, Martin Arlitt, and Anirban Mahanti. Traffic classification using clustering algorithms. In *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, pages 281–286. ACM, 2006.

[29] Zhu Li, Ruixi Yuan, and Xiaohong Guan. Accurate classification of the internet traffic based on the svm method. In *Communications, 2007. ICC'07. IEEE International Conference on*, pages 1373–1378. IEEE, 2007.

[30] Alice Este, Francesco Gringoli, and Luca Salgarelli. Support vector machines for tcp traffic classification. *Computer Networks*, 53(14):2476–2490, 2009.

[31] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15(1):3133–3181, 2014.

[32] C. Wang, T. Xu, and X. Qin. Network traffic classification with improved random forest. In *2015 11th International Conference on Computational Intelligence and Security (CIS)*, pages 78–81, December 2015. doi: 10.1109/CIS.2015.27.

[33] Jeffrey Erman, Anirban Mahanti, Martin Arlitt, Ira Cohen, and Carey Williamson. Offline/realtime traffic classification using semi-supervised learning. *Performance Evaluation*, 64 (9-12):1194–1213, 2007.

[34] Jeffrey Erman, Anirban Mahanti, Martin Arlitt, Ira Cohen, and Carey Williamson. Semi-supervised network traffic classification. In *ACM SIGMETRICS Performance Evaluation Review*, volume 35, pages 369–370. ACM, 2007.