Journal of Network and Systems Management (2020) 28:827–849
https://doi.org/10.1007/s10922-020-09512-5

A Frontier: Dependable, Reliable and Secure Machine Learning for Network/System Management

Duc C. Le[1] · **Nur Zincir-Heywood**[1]

Received: 14 November 2019 / Revised: 15 January 2020 / Accepted: 20 January 2020 /
Published online: 30 January 2020

Abstract
Modern networks and systems pose many challenges to traditional management approaches. Not only the number of devices and the volume of network traffic are increasing exponentially, but also new network protocols and technologies require new techniques and strategies for monitoring controlling and managing up and coming networks and systems. Moreover, machine learning has recently found its successful applications in many fields due to its capability to learn from data to automatically infer patterns for network analytics. Thus, the deployment of machine learning in network and system management has become imminent. This work provides a review of the applications of machine learning in network and system management. Based on this review, we aim to present the current opportunities and challenges in and highlight the need for dependable, reliable and secure machine learning for network and system management.

Keywords Network and system management · Reliable and dependable machine learning · Secure machine learning

1 Introduction

Networks are growing at exponential pace and becoming more and more diverse, not only connecting people but also machines and digital objects. The vast collections of network devices, end user devices and heterogeneous links are also growing, both in terms of numbers and types of devices. Naturally, this results in many opportunities as well as challenges in the process of managing such networks, services and systems. Furthermore, recent network developments, although

✉ Nur Zincir-Heywood
zincir@cs.dal.ca

Duc C. Le
lcd@dal.ca

[1] Dalhousie University, Halifax, Canada

creating tremendous potential applications and greatly enhancing network capabilities and user experiences, bring with them new challenges for network and system management (NSM). For example, the proliferation of 5G networks has been anticipated to open several new opportunities. This next generation mobile network technology greatly increases data transfer rates, while reducing latency and energy usage. Essentially 5G will enable Internet of Things (IoT) and many other use cases, such as smart transportation and high-performance edge analytics. Another example of network expansion and diversification is smart cities and homes. These in return create challenges in managing networks and services by introducing new heterogeneity and diversity, as well as cybersecurity concerns. Analyzing operational data and network traffic data generated by those networks for troubleshooting and detecting anomalies/faults/intrusions would be overwhelming to human analysts, given the sheer amounts of data they create. Similarly, Network Function Virtualization (NFV) and Software Defined Network (SDN) technologies bring many opportunities as well as challenges for network and system management by allowing centralized but potentially dynamic management functionalities via on the fly configuration, scheduling and analysis operations. Envisioning the scale and variability of networks and their potential growth in the near future, one can easily realize the need for more efficient and easily adaptable systems for NSM. Thus, machine learning based techniques and strategies may find applications which match well for their capabilities to learn from network/service data and provide support to analysts in monitoring, analyzing, and controlling networks and systems.

Machine learning (ML) is the computational process of automatically inferring and generalizing a model from sampled data [1]. In the last decade, ML has enjoyed an unprecedented surge in interest, thanks to the demonstrations of its usefulness (meet or exceed human level) in many tasks, such as computer vision, natural language processing, and computer gaming [2]. Hence, in recent years, with the surging popularity of ML, there has been a growing interest for its application to NSM as well. In this case, the goal is to leverage ML techniques and algorithms for analyzing huge data streams to support network and system management teams on daily operational tasks and/or to deliver self-driving networks [3]. In fact, both the current trends in networks and services, and the future outlook guarantee key requirements for application of ML and big data analytics: large data streams, increasing complexity and diversity of the services, technologies and protocols used, repetition of management tasks, and dynamics in networks/services and users/systems data.

This article aims to review the applications of ML in managing networks and systems in the literature. In doing so, we summarize the current state, highlight research opportunities for addressing the main challenges as well as preparing for future networks, systems and services. We highlight the need for robust and adaptable ML techniques for NSM applications, considering the dynamics in networks and systems of today as well as the trends indicating the future. The rest of the paper is organized as follows. Section 2 provides an overview of ML for NSM with an application workflow. Section 3 summarizes the current state of ML applications in NSM tasks, while Sect. 4 presents the main research challenges and opportunities in ML for network/system management. Finally, Sect. 5 draws conclusions and discusses the future directions.

2 ML for Network/System Management: An Overview

Figure 1 presents a workflow for ML adaptation in NSM applications, from data collection and processing steps to ML model construction, deployment, and inference steps. Although similar to a typical machine learning application workflow [4, 5], in this case, we emphasize the involvement of a network administrator/human analyst in the workflow, especially for the data processing as well as the results and analysis steps. The steps in the workflow are detailed as follows:

Data collection ML is a data-driven methodology for building analytical models automatically, so everything starts with data. A good monitoring and data collection procedure generates adequate data for employing ML techniques and supporting human analysts in making correct decisions [1, 6]. Data for network/service management may come from many sources, such as captured network traffic data (traffic traces or network flows), system and service event operation log data, and related information collected at different Internet protocol stack layers and devices. Depending on the application requirements and the ML algorithm used, the data collection step needs to be tailored to provide the suitable information. For example, in traffic prediction and classification tasks, the most important data source is network traffic captures, while for the fault prediction task, system event logs may be of more importance. Moreover, making decisions about what data to include and what not to include, may introduce biases into the type of ML solution one finds. Specific examples might include attempts to sample data to

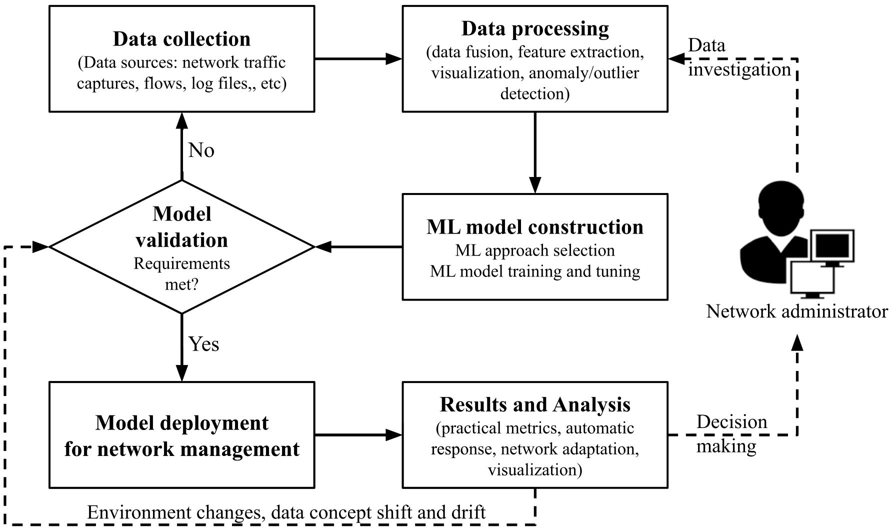

Fig. 1 A ML workflow in network/system management (adapted with changes from [5])

address class imbalance. This will 'prioritize' the detection or characterization of certain data properties. In traditional supervised learning, training data is usually collected and labelled to train the ML models in an offline fashion before the deployment of the ML based system.

Data processing This step essentially transforms the raw collected data into suitable data formats for training the ML algorithms. Usually, input data is represented as a finite set of fixed-length vectors, $X = \{x_1, x_2, ..., x_n\}$, where $x_i \in \mathbb{R}^m$. Therefore, different data processing phases need to be carried out. If data is collected from multiple sources, data fusion can be performed to unify the sources into a single data stream for further processing. Feature engineering or extraction can be performed to generate data features that are representative of the original data but in a more desired format. The process often includes data normalization, data imputation, feature selection and reduction. It should be noted here that too many features might result in ML solutions that do not generalize well. Last but not the least, the majority of ML techniques find correlations, not causations. Thus, the supervision of a network analyst is necessary in this step to provide any domain-specific knowledge that may be required for the subsequent ML application. This is often based on data analysis and exploration using visualization, alerts or unusual patterns in daily network/service operations and anomaly detection.

ML model construction This step involves selecting the ML algorithms to be used, and training them to address the needs of a specific analysis. There is a wide variety of ML techniques in the literature (Sect. 2.1). For example, for traffic classification, supervised learning techniques can be employed to learn from labelled data to predict future unknown data instances, while in network anomaly detection tasks, unsupervised learning techniques might be the primary option. Similarly, for applications involving temporal data properties, learning from data sequences using ML algorithms such as Hidden Markov Models (HMM) or Recurrent Neural Networks (RNN) can be adopted.

ML model validation ML models need to be validated in order to allow efficient and effective real-world deployment. Common metrics for validation include accuracy, precision, recall, F-score, detection and false alarm rates. Moreover, constraints related to the deployment environment, such as computational power and the response time are usually needed to be considered as well. Additionally, ML model validation should to be performed not only after the model construction step but also perpetually during the lifetime of the model deployment. We believe this is necessary in order to maintain a certain production level of performance that is acceptable to the organization. Many factors can negatively affect the ML model's performance, especially in dynamic network/service environments. Examples of such factors include changes in the networking environment (expansion of the network, topology changes, device replacements/upgrades) and behavioural changes such as concept shifts and drifts in the network, system, application and user activities.

ML model deployment Upon confirming that the trained ML solutions meet certain application requirements, the model is ready for deployment in network and systems. Specifically, this step involves preparing the necessary hardware, software, and manpower to ensure a smooth transition of the ML models to the production network/service environments.

Results and analysis (inference) This step presents the output of the ML model in a meaningful way in order to support network/service operations and management teams in making decisions regarding the related network/service behaviours and events. Traditional ML performance metrics include accuracy, precision, recall, detection rate, false positive/alarm rate, F-score, Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) [7]. However, there are specific needs of an output (reporting mechanism) of a network/service management system in order to allow a successful application of ML. Often, these may not be required in traditional data mining/machine learning applications. For example, in botnet detection using network traffic flows, a ML based botnet detector may simply output alarms based on individual suspicious flows. However, such a reporting scheme may overflow the analysts with alarms regarding a few hosts. This can also easily overlook the fact that other infected hosts may even be missed. Furthermore, if the botnet performs data exfiltration, all the important files may already be lost before a warning appears if the detection is taking too long. Hence, we take the view that for better understanding of the performance and for facilitating more meaningful responses (outputs), suitable metrics need to be reported by the solutions/models, such as host/user based results and detection delays. On the other hand, in this step, responses and adjustments in network/service systems can be made automatically by using the ML output.

2.1 ML Concepts

In this part, we present a brief overview to high level ML concepts used in the rest of the article. More detailed descriptions of these concepts can be found in [7]. This section is organized by the well-known ML tasks: classification, reinforcement learning, regression, clustering and anomaly/outlier detection. We note that by the use of labels (ground truth) in the learning process, ML methods can also be classified into: (i) supervised learning—where labels (ground truth information) are required for training the ML model; (ii) unsupervised learning—where labels are not required for training; and (iii) semi-supervised learning—where both labelled (typically a small amount) and unlabelled data are used during training.

Classification and regression Classification and regression are the most common supervised learning tasks in machine learning. In these tasks, the aim of a trained ML model is to identify the class/category (in classification) or output value (in regression) for a new data instance (observation) based on a set of training examples whose category membership/value (ground truth) are known. Some examples of the most popular classification algorithms are Artificial Neural Networks (ANN), Decision Trees, Random Forest, k-Nearest Neighbour (k-NN), Naive Bayes (NB), Logistic Regression (LR), and Support Vector Machines (SVM) [7]. In network/system management, classification is widely employed for many tasks, such as traffic/service classification, intrusion detection, or botnet detection (Sect. 3). Popular regression methods are Linear Regression, Polynomial Regression, Logistic Regression, and Lasso Regression [8]. In NSM, regression is mostly employed for time series prediction tasks, such as load and traffic prediction (Sect. 3.2).

Clustering and anomaly detection Cluster analysis and anomaly detection are typical unsupervised ML tasks. They are exploratory methods that are usually based on unlabelled training data. The aim of clustering is to group a set of data instances into sets/collections (clusters) so that instances in the same cluster are more similar to each other than to those in other clusters. Anomaly (outlier) detection aims to identify anomalous (rare, suspicious) events or observations (data instances) that deviate from a modelled norm. Clustering may act as a basis for anomaly detection in many cases [9]. In NSM, by modelling on the observed network data, which might not have the ground truth information, clustering and anomaly detection may allow the network analyst to discover anomalous and unusual patterns/clusters in network and systems operations. For example, anomalies in network traffic may indicate abnormal activities, such as botnets and intrusions, or performance and configuration problems. We note that as unsupervised methods, this category of ML usually generates a higher number of false alarms. This in return may require a higher level of attention from human analyst in order to identify the true network/system related issues or interesting patterns. Examples of their applications in NSM are network anomaly detection for intrusion/fault detection [10] (Sect. 3.3), and change detection [11] (Sect. 3.4).

In addition to supervised and unsupervised learning, another ML paradigm is *reinforcement learning*, which deals with how software agents interact with an environment in order to maximize cumulative reward. This differs from the supervised learning, where the labelled input/output pairs are used. In reinforcement learning, the agent is given an immediate reward (but not long-term reward) after choosing an action [12]. As such reinforcement learning is the ML equivalent of learning a policy for controlling a process by interacting directly with the environment. Naturally, significant effort might be necessary to ensure that the resulting policy satisfies all performance objectives.

Other important approaches in ML that have been employed or could potentially have applications in network/system management include online learning and transfer learning [13]. *Online learning* differs from traditional machine learning approaches in that data for training the ML models arrive in a sequential order and the models are updated at each step, as opposed to generating the best ML model by learning on the entire training data set at once [14]. Thus, online learning is commonly used in situations where it is computationally infeasible to train over the entire dataset, or when the algorithm needs to dynamically adapt to new patterns/trends in the data. On the other hand, *transfer learning* focuses on adapting knowledge learned by the ML solutions for solving a different but related problem [15]. In real-world situations, there are many cases where there is a classification task in one domain of interest, but sufficient training data is only available for a related domain. The differences between the domains may appear in feature space or data distribution. In such cases, knowledge transfer, if done successfully, would enable the ML application in the domains where training data is scarce. It will also improve the sample efficiency for the learning process by significantly lowering the data-labelling requirement. These methods may allow the ML models to actively adapt to dynamic and emerging patterns in the stream of data, which are commonly found in network environments [6]. It should also be noted here that the feature engineering [16], structured prediction (e.g Hidden Markov Model (HMM) [11]), and ensemble learning (bagging, boosting [17]) paradigms have also found several applications in NSM.

3 ML for Network Operations and Management

Machine learning has a long and vibrant history of applications in many network management tasks, which are related to the growth of networks and connected systems. There has been a considerable number of ML adaptations and developments for NSM by researchers. In this section, we summarize the ML based approaches for a wide range of NSM tasks [18, 19], including (but are not limited to) traffic and service classification, traffic prediction, performance management, security management, configuration management and fault management. Table 1 summarizes the literature review of ML approaches for NSM.

In summary, networks are diverse collections of devices and links, in which management tasks are complex and strongly correlated/connected. Furthermore, we emphasize the fact that almost all tasks in NSM are quintessentially related to the network monitoring and forensics, especially with the application of data mining and machine learning. Monitoring ensures that the network performance is recorded and allows systems and analysts to operate/control based on the events and activities observed on the various networks and services. Almost every task (functionality) in NSM starts with an adequate monitoring process to provide sufficient information for decision-making.

3.1 Traffic/Service Classification

The traffic/service classification task aims to identify the underlying traffic as well as the applications/services in the traffic. Accurate traffic classification provides critical information for network operators in order to manage the network bandwidth and to ensure the Quality of Service (QoS) and Quality of Experience (QoE) for their users. Furthermore, the understanding of traffic and services in a network enables the successful deployment of other management tasks, such as intrusion/anomaly detection, throughput modelling and prediction, and accounting management [20, 22].

Traditionally, network traffic and service classification has relied on the packet inspection and port number information. For example, traffic destined to port 80 can be categorized as web traffic. However, with the proliferation of traffic encryption and widespread usage of virtual private networks, anonymity networks such as Tor, and network tunnelling practices (e.g. SSH tunnelling), network traffic have become essentially indistinguishable under the traditional approach, making the traffic classification much more challenging. Another challenge comes from the fact of the growing number of Internet services, which are also dynamically changing based on user demands, network capacity, and trends. Hence the data driven approaches

Table 1 Summary of the literature survey on ML in NSM

Category	Surveys	NSM tasks and works
Traffic/service classification	Nguyen and Armitage [20], Velan et al. [21], Callado et al. [22]	*Unencrypted traffic classification* Kim et al. [23]$^{*,\beta,\alpha,\delta}$, Williams et al. [24]$^{*,\beta,\delta}$, Finamore et al. [25]*
		Encrypted traffic classification Alshammari and Zincir-Heywood [26]$^{*,\beta,\delta}$, Sun et al. [27]$^{*,\delta}$, Anderson and McGrew [28]$^{*,\beta,\alpha}$, Bar Yanai et al. [29]$^{*,\#,\P}$ Lotfollahi et al. [30]$^{*,\alpha}$
		Others host identification—Meidan et al. [31]$^{*,\beta}$, Khatouni et al. [32]$^{*,\beta,\delta,\epsilon}$, Anonimity traffic identification—Montieri et al. [33]$^{*,\beta,\delta}$, Shahbar and Zincir-Heywood [34]$^{*,\beta,\delta}$, and QoS class identification—Wang et al. [35]$^{\#,\P}$
Performance management	DrAlconzo et al. [36], Dalmazo et al. [37], Wang et al. [4], Boutaba et al. [19]	*Performance prediction* Cortez et al. [38]$^{*,\alpha}$, Oliveira et al. [39]$^{*,\alpha}$, Fadlullah et al. [40]$^{*,\alpha}$, Bantouna et al. [41]$^{\#,\P}$, Kim et al. [42]$^{*,\alpha}$, Moradi et al. [43]$^{*,\|,\alpha}$, Jeong et al. [44]*,§
		Others traffic management in cloud and mobile edge computing—Zhang and Zhou [45]$^{*,\gamma}$, network resource management and allocation—Yang et al. [46]$^{\ddagger,\alpha}$, Mao et al. [47]$^{\ddagger,\alpha}$, Tiwana and Tiwana [8]$^{*,\epsilon}$and congestion control [48]$^{\ddagger,\alpha}$, Li et al. [49]‡
Security management	Tsai et al. [50], Buczak and Guven [6], Bhuyan et al. [51], Dua and Du [1], Boutaba et al. [19]	*Anomaly detection* Sequeira and Zaki [52]†,¶,§, Jiang et al. [53]†,¶, Casas et al. [54]†,¶, Kayacık et al. [55]†,¶, Perdisci et al. [56]†, Veeramachaneni et al. [57]$^{\dagger\,\alpha}$
		Intrusion detection Zhao et al. [58]$^{*,\beta}$, Aburomman and Reaz [59]$^{*,\gamma}$, Shone et al. [60]$^{*,\alpha,\beta}$, Lopez-Martin et al. [61]$^{*,\ddagger,\alpha}$, Khanchi et al. [62]$^{*,\S,\gamma}$, Haddadi et al. [63]$^{*,\beta,\delta}$, Gu et al. [64]†,¶, Khan et al. [65]*
		Others moving target defence—Makanju et al. [66]$^{*,\gamma}$, Sengupta et al. [67]$^{*,\alpha}$, insider threat detection—Le et al. [68]$^{*,\alpha,\beta,\gamma,\S}$, Rashid et al. [69]†, and network content filtering – Chau and Chen [70]$^{*,\alpha}$

Table 1 (continued)

Category	Surveys	NSM tasks and works
Configuration management	Xie et al. [71], Zhang et al. [72]	*Self-organizing network* [73]‡, Moysen and Giupponi [74], Roy et al. [75]$^{*,\gamma}$
		Others: service configuration management—Wang et al. [76]‡, network routing—Valadarsky et al. [77]$^{*,\ddagger,\alpha}$, and network load balancing—Kim and Kim [78]$^{*,\ddagger,\alpha}$
Fault management	Xie et al. [71], Boutaba et al. [19]	*Fault detection*: Hajji [79]§, Yamanishi and Maruyama [80]†, Chen et al. [81]$^{*,\beta}$, Hashmi et al. [82]†,¶
		Others fault predicting—Zhang et al. [83]$^{*,\beta}$, fault management—Mismar and Evans [84]‡

ML approach: * Supervised ML, †unsupervised ML, ‡reinforcement learning, §online learning, $^{\|}$transfer learning, $^{\#}$semi-supervised learning, ¶clustering, $^{\alpha}$ANN—deep learning, $^{\beta}$decision tree, $^{\gamma}$evolutionary computation, $^{\delta}$Bayesian based, $^{\epsilon}$regression for pattern recognition and behaviour identification, become useful for analyzing the underlying traffic and services [21].

Early ML applications to traffic classification dates back to 2005, in which Zander et al. employed autoclass, an unsupervised Bayesian based algorithm, for categorizing network flows. Kim et al. and Williams et al. employed different ML approaches, including Decision Trees and Naive Bayes, for traffic classification based on network flows, and compared them to traditional approaches, such as port based, host based, and signature based [23, 24, 85] classification. Encrypted traffic analysis via the use of ML from detecting SSH to Skype to HTTPS traffic using network traffic flows (without using IP addresses and port numbers) also gained a lot of interest during the last decade [86]. Additionally, UDP flow extraction based on packet inspection has been applied in conjunction with a ML decision process in [25]. These works show that ML has the potential to support network/system management tasks under a high volume and dynamic network/system conditions [26, 27]. Recently, [28] demonstrated the use of ML for malware traffic classification in industrial environments with noise and non-stationarity traffic properties taken into account.

The variety of ML methods for traffic and service classification further demonstrates its potential in this application case. Along with popular supervised learning methods, such as the Decision tree, Random Forest, RIPPER, Logistic regression, C4.5 and C5.0, Neural networks, Genetic Programming [23–25, 28, 86, 87], semi-supervised and unsupervised learning methods, such as autoclass, k-means, Gaussian Mixture models have been employed as well [29, 35, 88–90]. Similarly, trending approaches, such as deep learning, has also found its application in this field [30].

On the other hand, many ML based solutions have been proposed to solve closely related problems to traffic classification, such as network host identification [31, 32], anonymity networks (e.g Tor) traffic identification [33, 34], QoS class identification [35], as well as network security (Sect. 3.3) and traffic prediction (Sect. 3.2).

3.2 Performance Management

Maintaining the network performance at production levels is the aim of performance management. The task requires monitoring and processing network data at different levels and devices for estimating the performance related key measures, such as throughput, delay, network utilization, and error rates. Network information for performance management is usually collected from the deployment of Simple Network Management Protocol (SNMP) agents, remote monitoring agents and/or active management agents, such as Nagios [10, 91]. The analysis of the monitored performance measures enables the identification of the health status of the network as well as the potential problems (i.e faults, Sect. 3.4). Additionally, trends in different performance measures can provide valuable information for long-term capacity planning and deployment. Given the current developments in networks and systems, machine learning has naturally found its place in performance management tasks for its ability to learn from large amounts of data to predict possible network conditions as well as to aggregate patterns automatically in order to identify suitable triggers for management actions.

In performance management, traffic prediction is a task that has seen multiple ML based proposals. The ML based traffic prediction methods, however, are mostly based on neural networks [36]. The advantages of neural network based methods over traditional time series forecasting methods (ARIMA, Holt-Winters [37]) in real-time, short-term, and mid-term forecasting of network traffic are demonstrated in several works in the literature [38–40, 92]. Other ML methods including Genetic Algorithms [45], SVR [93], Self Organizing Feature Maps [41], and HMMs [94] where they have been employed in different traffic prediction scenarios. In addition to traffic prediction, many other tasks in performance management have seen several proposed ML based solutions: Traffic management in cloud and mobile edge computing [45, 95], network resource management and allocation [8, 46, 47, 96], QoS assurance [97, 98], and congestion control [48, 49].

Given the latest developments in ML for other application areas, such as transfer learning, reinforcement learning, and online learning, one can easily imagine their uses in specific network performance management scenarios as well. These leverage the capabilities of certain ML techniques to learn from temporal and dynamic patterns of data. Current examples of such developments include deep neural networks such as Long Short Term Memory (LSTM) [42], transfer learning [43], deep reinforcement learning [47, 48], and online learning [44]. Recently, Fadlullah et al. summarizes the use of deep learning in network traffic control tasks and indicates the potential of intelligent network traffic control using the state-of-the-art ML techniques [40].

3.3 Security Management

Security management is one of the network and service management functionalities that has observed extensive and early endorsement of ML techniques. Network anomaly detection is a prime example, in which machine learning techniques are applied for their ability to automatically learn from the data and extract patterns that can be used for identifying network anomalies in a timely manner [51]. In anomaly detection, unsupervised learning is the most widely applied technique due to its ability to learn without *a priori* knowledge of anomalies, which is the defining characteristic of the task. Generally, a model generalizing normal network and user behaviours is constructed using unsupervised learning. Then, measures can be derived from the model to detect anomalies in network traffic/behaviours. Examples of works in this approach are [99], where temporal correlation, wavelet analysis and traditional change point detection approaches are applied to produce a network signal model and worm traffic model, [52] and [69], where the sequence of user actions in a time window is used to create user profiles using clustering techniques and Hidden Markov Models. A more straightforward approach is to apply clustering analysis and outlier detection methods directly to the collected data, with the assumption that normal behaviours account for the majority of the collected data [51]. For example, clustering algorithms are used to find significant clusters representing majority of normal network/system behaviours. Then, from clustering results, anomalies can be detected automatically using outlier detection methods to identify the data instances that do not fit into the constructed clusters [53, 54, 100, 101]. Unsupervised learning results can also be analyzed by human experts with or without the use of visualization as well as other supervised learning methods to give deeper insight [55, 56]. The Self Organizing Map (SOM) is a well-known unsupervised learning technique with the ability of summarizing and visualizing the data learned in a topologically preserved way for further inspection [55, 85]. Recently, Veeramachaneni et al. [57] applied the concept of big data for anomaly detection, where human experts, multiple outlier detection techniques, and supervised learning are combined to learn from large amounts of data to detect network anomalies and intrusions at the same time.

Similar to anomaly detection, ML has been extensively applied to network intrusion detection [6, 50]. ML methods for Intrusion Detection Systems (IDSs) include mostly supervised learning techniques, such as Neural Networks, Decision Trees, Evolutionary Computing, Bayesian Networks, SVMs, and Logistic Regression [58, 59, 68, 102, 103] and recently deep and reinforcement learning [60, 61, 103]. Unsupervised learning [64, 104] and stream online learning [62, 68] have been employed for security tasks as well. The diverse employment of ML in IDS can also be represented through different detection points of ML based IDSs: Network based IDS [105, 106], Host based IDS [107, 108], or hybrid systems [64, 65]. Furthermore, in many works, ML based methods are demonstratively superior to traditional approaches where the detection system uses handcrafted rules based on the expert knowledge [6, 63, 109]. Other notable examples of ML applications in security management include moving target defence [66, 67], insider threat detection [69, 102], and network content filtering [70, 110].

3.4 Configuration and Fault Management

The adoption of ML in configuration management has been slow in traditional wireline networks but quite widespread in wireless networks. However, with the advent of network function virtualization and software defined networking, this is changing as well. A particular example of network configuration management with ML approaches in recent years is self-organizing networks, which focus on the planning, configuration, management, optimization and healing of mobile radio access networks [72, 73]. With the development of 5G, several ML based solutions for self-organizing mobile networks are proposed based on techniques from deep reinforcement learning to bio-inspired algorithms [74, 75, 111]. Other example tasks in configuration management employing ML are service configuration management [76], network routing [77, 112], and network load balancing [78, 113].

In fault management, detection and prediction of network/system faults attracted the most ML applications. Fault detection is mostly formalized as anomaly/change detection, in which a normal baseline of a network/system operation and parameters are profiled using ML techniques. After which, any faults or abnormal activities observed on the network are detected as deviations from those models [79, 80, 82]. Notable ML methods employed in fault management include online learning for change point detection [79], fuzzy probabilistic neural networks [114], HMMs [80], decision trees [81], and several unsupervised learning algorithms [82, 115].

Additionally, some ML approaches have been introduced in fault prediction [83], fault [84], and automated fault mitigation [116]. With the proliferation of SDN and NFV approaches, which allow centralized configuration and management of networks, it is expected that ML will be adopted much more extensively in network configuration and fault management in the near future [71].

4 Challenges and Research Opportunities

Although machine learning has been extensively applied in many tasks in NSM, there are certain challenges that need to be overcome for a successful realization of the potentials in production network/service environments. Some challenges and research opportunities that come along with them are presented in this section.

4.1 Data Related Challenges

Network data is diverse and abundance, yet obtaining high quality data for designing and evaluating ML based systems for network /service operations and management poses many challenges. Firstly, most companies and organizations are prevented from sharing or even analyzing network data by agreements protecting users' identities and privacy related issues. Moreover, there are not many benchmarking data sets with ground truth to utilize in this area. The data sets that are publicly available for benchmarking purposes are old and out of date in terms of the behaviours they include (e.g. DARPA 1998, 1999 and 2000 IDS data sets) or they are more up to date but very small (e.g. Snort VRT Labs). Even when the data is shared, most of it comes heavily anonymized, encrypted, and without any forms of ground truth information for training and evaluating ML systems. Secondly, network data is usually highly imbalanced and impure. Most of the time only a small unidentified portion of the data is representing interesting patterns/events/behaviours, such as anomalous activities. For example, in the case of Advanced Persistent Threats (APTs), attackers can perform stealthy malicious actions over a very long duration to evade monitoring systems. Hence, the sign of network anomalies/events could be overlooked by ML based systems. Furthermore, impurity and noise in network data may cause ML models inadvertently to be built using abnormal/malicious data encoded as normal behaviour, e.g. advesarial training. This in return, makes them incapable of detecting the future anomalies/malicious activities of the targeted type. Finally, network, service and system data are presented at multiple levels of granularity and in wide variety of formats. The data can be acquired at host or network level, from end-user devices, network devices, security devices, and/or systems and servers, in many formats and structures. In organizations and networks of all sizes, the problem of data acquisition, data representation and data processing must be addressed systematically in order to provide high quality data for training ML systems efficiently. Furthermore, reducing computational complexity in pre-processing, training and deployment of ML based systems is also a priority for deployment of ML based

NSM solutions. Indeed, the computational overhead of deploying some ML paradigms might be prohibitive for some real-time network applications.

The presented challenges create research opportunities as well. In order to address data privacy requirements, new privacy aware machine learning approaches can be employed. Notable examples of such techniques are federated learning [117] approaches, which consist of training ML models collectively across multiple decentralized edge devices without exchanging local data samples. Moreover, privacy preserving practices, such as differentially private ML [118] and homomorphic encryption [119] have been attracting attention in terms of privacy aware ML based solutions. On the lack of data with established ground truth, unsupervised ML and anomaly detection techniques may find further application [120]. Finally, we note that many ML approaches for big data can be applied in network data to meet the requirements for processing and analyzing the huge amounts of data generated by networks, systems and services [36].

4.2 Towards Automatic Network Management

Networks and services are continuously evolving, with new protocols and technologies introduced in order to address current problems and improve the QoS/QoE of the services provided. Several recent examples include (but are not limited to) the development and adoption of SDNs, OpenFlow, and NFV systems. The ability to separate network control functions from network forwarding functions or to abstract network forwarding and other networking functions from the hardware brought by the SDN and NFV technologies create many opportunities for ML applications. For example, the centralized network intelligence in SDN controllers allow unified data storing and processing for ML based network analyzers. Furthermore, newly introduced network technologies, such as 5G, HTTP/2, HTTP/3, bring new challenges in collecting and analyzing the network/service data. Yet the ML approaches considering the new technologies and protocols are still lacking.

Similarly, with the development of automated systems, self-driving cars, vehicular networks, zero touch and self-driving networks have been proposed [3]. As a consequence, managing such systems and services, ML solutions, which are data driven, would be a very good match going forward. Self-driving networks will also require the ability of the ML based operation and management systems to be able to make decisions automatically and behave proactively based on the activities and events occurring on the networks and systems. Furthermore, comprehensive ML approaches for designing and actively monitoring the networks and services are also needed.

4.3 Human Involvement in ML Based Network/System Management

The feasibility for human network operators to understand and command ML based systems is of utmost importance for successful deployment in real-world network/service environments. Even self-driving networks still require transparency for human involvement and troubleshooting.

It is evident that ML applications for network management have to meet specialized requirements that do not necessarily exist in traditional ML applications. These are presented in ML workflow, such as data ingestion, specialized performance metrics, and/or analytic steps. Many of those need domain expert knowledge for suitable implementations. For example, network data comes from a wide variety of sources at arbitrary times, hence human experts' knowledge is required for designing solutions to aggregate data and extract features in a meaningful way for training the ML algorithms. Another example is in the case of network anomaly/intrusion detection systems, considering the dominance of normal data and the scale of networks, even a small false positive rate (in traditional ML standards) may result in a catastrophic amount of false alarms that require attention from cybersecurity analysts. Thus, the ML based anomaly/intrusion detection systems need to be able to correlate alerts and events (e.g. based on host, user, or subnet) to reduce the amount of alarms.

Human–machine interactions need to be addressed in designing ML based NSM solutions for future deployments. For these interactions, the ML models are needed to be transparent so that the automatic systems using these models could provide human understandable solutions for decision making and support purposes. Moreover, the ML models used need to allow/provide trace-back as a service to identify the source of problems/events for correct and timely human intervention. Furthermore, to incorporate the ability to learn from the human verdicts on ML output (e.g intrusion alarms) would help to continuously evolve the deployed ML based management systems and would greatly enhance the system capabilities. Finally, we note that the recent ML approaches for human–machine interactions, such as [121], may be advantageous in many NSM scenarios as well.

4.4 Robust, Adaptable, and Dependable ML for Networking

There are inherent dynamics in networks, systems and services. Network data are generated perpetually in streams of high volume and velocity. Remarkably, changes may happen in network devices and user/system behaviours, gradually as drifts in user behaviours, or suddenly (shifts) as in the cases of network failures or high volume Distributed Denial of Service (DDoS) attacks. Hence, ML models for network and service management needs to take the dynamics and changes into account in order to ensure successful deployment. In this respect, we emphasize the need for robust, reliable, and dependable ML based systems for NSM. Notable approaches for addressing the challenges include online learning, adversarial learning, and transfer learning.

The very nature of networks and services is data streams. In small scale networks and services, the gradual drifting and shifting of behaviours and concepts in data may be addressed by retraining the ML models periodically. However, in large scale networks and services, dynamic and adaptive learning algorithms and self-evolving architectures that are capable of working on high speed data streams are necessary. This is because predicting the intervals for retraining the ML models could become more and more challenging in large scale networks and services. There is a great potential for ML systems that have capabilities to revise or update themselves actively and timely according to the ever changing dynamics in networks and services, as well as the continuous feedback provided by human analysts (experts) without sacrificing their performances. *Stream online learning* may shine in this regard. There have been some applications of stream online learning, mostly in network security [44, 62, 68]. However, the technique has tremendous potential for further exploration in ML based NSM systems.

In many network management tasks, such as intrusion detection and fault detection, adversarial situations are inherent. In intrusion detection, the attackers are continuously evolving their attacks to evade detection systems. This creates an arm-race between attackers and defenders. In fact, with the recent waves of deep learning applications in networking, network adversaries may exploit different perturbations [122] in order to trick the defense system to make false decisions [123]. Similarly, in anomaly detection and fault detection, defining network anomalies/faults and distinguishing them from normal behaviours (i.e. learned patterns) present a challenge to the traditional ML models, such as classification and recommendation systems, which are designed to find similarities instead. Developing ML based NSM systems with adversaries in mind will definitely enhance their practicality, especially in terms of securing the ML model against (i) evasions and adversarial training, (ii) generalization to different deployment locations and (iii) robustness over time. An example is an artificial arm-race employing Evolutionary Algorithms or Generative Adversarial Neural Networks (GAN) for evolving attacks (or generating network anomalies) and defence mechanisms at the same time to prepare for future threats and anomalies [16, 124, 125]. Recent advances in adversary aware and resistant machine learning, such as [126], could provide resiliency to network threats as well. Other examples include ML applications in moving target defence [66] and traffic obfuscation [127].

Finally, there are significant challenges in generalizing ML based management systems, in order to become independent of the environment. Different network and service environments usually have different data, requirements, and conditions for defining the basic ML settings based on different user/system behaviours. *Transfer learning* techniques provide tools for addressing these scenarios, which have been applied in [43]. Adaptable ML methods aiming to provide dependable operations under changing and dynamic network/service conditions are explored in [128, 129] as well.

5 Conclusions and Future Work

Given the scale and dynamics of today's networks, systems and services, it is easy to envision that ML based network and service management solutions will become more and more prevalent and crucial for monitoring and securing the systems and devices of the future. Developing practical ML applications for the aforementioned management and operations tasks is an open field of research. This creates many opportunities for addressing the NSM challenges, while also bringing their own challenges such as secure and robust ML techniques. In this article, we briefly surveyed the current state of the ML applications in NSM. For the future research directions, we highlight the main challenges that relate to data for ML, new network technologies, human involvement, and specifically the need for robust and adaptable ML methods. We believe that creating reliable, dependable and secure ML models for network, system and service management will be the next frontier in this era.

Acknowledgements This research is supported by Natural Science and Engineering Research Council of Canada (NSERC). Duc C. Le gratefully acknowledges the supports of the Killam Trusts and the province of Nova Scotia. The research is conducted as part of the Dalhousie NIMS Lab at: https://projects.cs.dal.ca/projectx/. The authors would like to thank the anonymous reviewers.

References

1. Dua, S., Du, X.: Data Mining and Machine Learning in Cybersecurity. Auerbach Publications, Boca Raton (2016)
2. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
3. Kalmbach, P., Zerwas, J., Babarczi, P., Blenk, A., Kellerer, W., Schmid, S.: Empowering self-driving networks. In: Proceedings of the afternoon workshop on self-driving networks, pp. 8–14. ACM, New York (2018)
4. Shearer, C.: The CRISP-DM model: the new blueprint for data mining. J. Data Warehous. **5**(4), 13–22 (2000)
5. Wang, M., Cui, Y., Wang, X., Xiao, S., Jiang, J.: Machine learning for networking: workflow, advances and opportunities. IEEE Netw. **32**(2), 92–99 (2017)
6. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun. Surv. Tutor. **18**(2), 1153–1176 (2016). https://doi.org/10.1109/COMST.2015.2494502
7. Alpaydin, E.: Introduction to Machine Learning. The MIT Press, Cambridge (2014)
8. Tiwana, M.I., Tiwana, M.I.: A novel framework of automated RRM for LTE son using data mining: application to LTE mobility. J. Netw. Syst. Manag. **22**(2), 235–258 (2014)
9. Aggarwal, C.C.: Outlier Analysis, 2nd edn. Springer Publishing Company, Incorporated, New York (2016)
10. Calyam, P., Dhanapalan, M., Sridharan, M., Krishnamurthy, A., Ramnath, R.: Topology-aware correlated network anomaly event detection and diagnosis. J. Netw. Syst. Manag. **22**(2), 208–234 (2014)
11. Vaton, S., Brun, O., Mouchet, M., Belzarena, P., Amigo, I., Prabhu, B.J., Chonavel, T.: Joint minimization of monitoring cost and delay in overlay networks: optimal policies with a Markovian approach. J. Netw. Syst. Manag. **27**(1), 188–232 (2019)
12. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: a survey. J. Artif. Intell. Res. **4**, 237–285 (1996)
13. Nawrocki, P., Sniezynski, B.: Adaptive service management in mobile cloud computing by means of supervised and reinforcement learning. J. Netw. Syst. Manag. **26**(1), 1–22 (2018)
14. Heywood, M.I.: Evolutionary model building under streaming data for classification tasks: opportunities and challenges. Genet. Program. Evol. Mach. **16**(3), 283–326 (2015)
15. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2009)
16. Kayacık, H.G., Zincir-Heywood, A.N., Heywood, M.I.: Evolutionary computation as an artificial attacker: generating evasion attacks for detector vulnerability testing. Evolut. Intell. **4**(4), 243–266 (2011)
17. Breiman, L.: Random forests. Mach. Learn. (2001). https://doi.org/10.1023/A:1010933404324

18. ISO/IEC: Information Processing Systems—Open Systems Interconnection—Basic Reference Model—Part 4 Management Framework. Standard International Organization for Standardization, Geneva (1989)
19. Boutaba, R., Salahuddin, M.A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., Caicedo, O.M.: A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. J. Internet Serv. Appl. **9**(1), 16 (2018). https://doi.org/10.1186/s13174-018-0087-2
20. Nguyen, T.T.T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. IEEE Commun. Surv. Tutor. **10**(4), 56–76 (2008). https://doi.org/10.1109/SURV.2008.080406
21. Velan, P., Čermák, M., Čeleda, P., Drašar, M.: A survey of methods for encrypted traffic classification and analysis. Int. J. Netw. Manag. **25**(5), 355–374 (2015). https://doi.org/10.1002/nem.1901
22. Callado, A., Kamienski, C., Szabo, G., Gero, B.P., Kelner, J., Fernandes, S., Sadok, D.: A survey on internet traffic identification. IEEE Commun. Surv. Tutor. **11**(3), 37–52 (2009). https://doi.org/10.1109/SURV.2009.090304
23. Kim, H., Claffy, K.C., Fomenkov, M., Barman, D., Faloutsos, M., Lee, K.Y.: Internet traffic classification demystified: myths, caveats, and the best practices. In: Proceedings of 4th ACM international conference on emerging networking experiments and technologies, CoNEXT '08, https://doi.org/10.1145/1544012.1544023 (2008)
24. Williams, N., Zander, S., Armitage, G.: A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. Comput. Commun. Rev. **36**(5), 7–15 (2006). https://doi.org/10.1145/1163593.1163596
25. Finamore, A., Mellia, M., Meo, M., Rossi, D.: KISS: stochastic packet inspection classifier for udp traffic. IEEE/ACM Trans. Netw. **18**(5), 1505–1515 (2010). https://doi.org/10.1109/TNET.2010.2044046
26. Alshammari, R., Zincir-Heywood, A.N.: Machine learning based encrypted traffic classification: identifying ssh and skype. In: 2009 IEEE symposium on computational intelligence for security and defense applications, pp. 1–8 (2009) https://doi.org/10.1109/CISDA.2009.5356534
27. Sun, G., Xue, Y., Dong, Y., Wang, D., Li, C.: An novel hybrid method for effectively classifying encrypted traffic. In: 2010 IEEE global telecommunications conference GLOBECOM 2010, pp. 1–5 (2010). https://doi.org/10.1109/GLOCOM.2010.5683649
28. Anderson, B., McGrew, D.: Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, vol. Part F1296, pp. 1723–1732 (2017). https://doi.org/10.1145/3097983.3098163
29. Bar Yanai, R., Langberg, M., Peleg, D., Roditty, L.: Realtime classification for encrypted traffic. In: Festa, P. (ed.) Experimental Algorithms, pp. 373–385. Springer, Berlin (2010)
30. Lotfollahi, M., Jafari Siavoshani, M., Shirali Hossein Zade, R., Saberian, M.: Deep packet: a novel approach for encrypted traffic classification using deep learning. Soft Comput. (2019). https://doi.org/10.1007/s00500-019-04030-2
31. Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J.D., Ochoa, M., Tippenhauer, N.O., Elovici, Y.: Profiliot: a machine learning approach for iot device identification based on network traffic analysis. In: Proceedings of the symposium on applied computing, pp. 506–509. ACM, New York (2017). https://doi.org/10.1145/3019612.3019878
32. Khatouni, A.S., Zhang, L., Aziz, K., Zincir, I., Zincir-Heywood, N.: Exploring nat detection and host identification using machine learning. In: CNSM (2019)
33. Montieri, A., Ciuonzo, D., Aceto, G., Pescapé, A.: Anonymity services tor, i2p, jondonym: classifying in the dark. In: 2017 29th international teletraffic congress (ITC 29), vol. 1, pp. 81–89. IEEE, New York (2017)
34. Shahbar, K., Zincir-Heywood, A.N.: How far can we push flow analysis to identify encrypted anonymity network traffic? In: 2018 IEEE/IFIP network operations and management symposium, pp. 1–6 (2018). https://doi.org/10.1109/NOMS.2018.8406156
35. Wang, P., Lin, S.C., Luo, M.: A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs. In: 2016 IEEE international conference on services computing (SCC), pp. 760–765. IEEE, New York (2016)
36. DrAlconzo, A., Drago, I., Morichetta, A., Mellia, M., Casas, P.: A survey on big data for network traffic monitoring and analysis. IEEE Trans. Netw. Serv. Manag. (2019). https://doi.org/10.1109/tnsm.2019.2933358

37. Dalmazo, B.L., Vilela, J.P., Curado, M.: Performance analysis of network traffic predictors in the cloud. J. Netw. Syst. Manag. **25**(2), 290–320 (2017). https://doi.org/10.1007/s10922-016-9392-x
38. Cortez, P., Rio, M., Rocha, M., Sousa, P.: Internet traffic forecasting using neural networks. In: The 2006 IEEE international joint conference on neural network proceedings, pp. 2635–2642. IEEE, New York (2006)
39. Oliveira, T.P., Barbar, J.S., Soares, A.S.: Computer network traffic prediction: a comparison between traditional and deep learning neural networks. Int. J. Big Data Intell. **3**(1), 28–37 (2016)
40. Fadlullah, Z.M., Tang, F., Mao, B., Kato, N., Akashi, O., Inoue, T., Mizutani, K.: State-of-the-art deep learning: evolving machine intelligence toward tomorrow's intelligent network traffic control systems. IEEE Commun. Surv. Tutor. **19**(4), 2432–2455 (2017)
41. Bantouna, A., Poulios, G., Tsagkaris, K., Demestichas, P.: Network load predictions based on big data and the utilization of self-organizing maps. J. Netw. Syst. Manag. **22**(2), 150–173 (2014). https://doi.org/10.1007/s10922-013-9285-1
42. Kim, H.G., Lee, D.Y., Jeong, S.Y., Choi, H., Yoo, J.H., Hong, J.W.K.: Machine learning-based method for prediction of virtual network function resource demands. In: 2019 IEEE conference on network softwarization (NetSoft), pp. 405–413. IEEE, New York (2019)
43. Moradi, F., Stadler, R., Johnsson, A.: Performance prediction in dynamic clouds using transfer learning. In: 2019 IFIP/IEEE symposium on integrated network and service management (IM), pp. 242–250. IEEE, New York (2019)
44. Jeong, Y.S., Byon, Y.J., Castro-Neto, M.M., Easa, S.M.: Supervised weighting-online learning algorithm for short-term traffic flow prediction. IEEE Trans. Intell. Transp. Syst. **14**(4), 1700–1707 (2013)
45. Zhang, Y., Zhou, Y.: Distributed coordination control of traffic network flow using adaptive genetic algorithm based on cloud computing. J. Netw. Comput. Appl. **119**, 110–120 (2018)
46. Yang, T., Hu, Y., Gursoy, M.C., Schmeink, A., Mathar, R.: Deep reinforcement learning based resource allocation in low latency edge computing networks. In: 2018 15th international symposium on wireless communication systems (ISWCS), pp. 1–5. IEEE, New York (2018)
47. Mao, H., Alizadeh, M., Menache, I., Kandula, S.: Resource management with deep reinforcement learning. In: Proceedings of the 15th ACM workshop on hot topics in networks, pp. 50–56. ACM, New York (2016)
48. Bachl, M., Zseby, T., Fabini, J.: Rax: deep reinforcement learning for congestion control. In: ICC 2019-2019 IEEE international conference on communications (ICC), pp. 1–6. IEEE, New York (2019)
49. Li, W., Zhou, F., Chowdhury, K.R., Meleis, W.M.: Qtcp: adaptive congestion control with reinforcement learning. IEEE Trans. Netw. Sci. Eng. **6**(3), 445–458 (2018)
50. Tsai, C.F., Hsu, Y.F., Lin, C.Y., Lin, W.Y.: Intrusion detection by machine learning: a review. Expert Syst. Appl. **36**(10), 11994–12000 (2009)
51. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: Network anomaly detection: methods, systems and tools. IEEE Commun. Surv. Tutor. **16**(1), 303–336 (2014). https://doi.org/10.1109/SURV.2013.052213.00046
52. Sequeira, K., Zaki, M.: ADMIT: anomaly-based data mining for intrusions. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining, pp. 386–395 (2002). https://doi.org/10.1145/775047.775103
53. Jiang, S., Song, X., Wang, H., Han, J.J., Li, Q.H.: A clustering-based method for unsupervised intrusion detections. Pattern Recognit. Lett. **27**(7), 802–810 (2006). https://doi.org/10.1016/j.patrec.2005.11.007
54. Casas, P., Mazel, J., Owezarski, P.: Unsupervised network intrusion detection systems: detecting the unknown without knowledge. Comput. Commun. **35**(7), 772–783 (2012). https://doi.org/10.1016/j.comcom.2012.01.016
55. Kayacık, H.G., Zincir-Heywood, A.N., Heywood, M.I.: A hierarchical SOM-based intrusion detection system. Eng. Appl. Artif. Intell. **20**(4), 439–451 (2007)
56. Perdisci, R., Gu, G., Lee, W.: Using an ensemble of one-class svm classifiers to harden payload-based anomaly detection systems. In: Sixth international conference on data mining (ICDM'06), pp. 488–498 (2006). https://doi.org/10.1109/ICDM.2006.165
57. Veeramachaneni, K., Arnaldo, I., Korrapati, V., Bassias, C., Li, K.: AI^2: training a big data machine to defend. In: 2016 IEEE 2nd international conference on big data security on cloud (BigDataSecurity), pp. 49–54. IEEE, New York. (2016). https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2016.79

58. Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., Garant, D.: Botnet detection based on traffic behavior analysis and flow intervals. Comput. Secur. **39**, 2–16 (2013)
59. Aburomman, A.A., Reaz, M.B.I.: A novel SVM-kNN-PSO ensemble method for intrusion detection system. Appl. Soft Comput. **38**, 360–372 (2016)
60. Shone, N., Ngoc, T.N., Phai, V.D., Shi, Q.: A deep learning approach to network intrusion detection. IEEE Trans. Emerg. Top. Comput. Intell. **2**(1), 41–50 (2018)
61. Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A.: Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Syst. Appl. **141**, 112963 (2020)
62. Khanchi, S., Vahdat, A., Heywood, M.I., Zincir-Heywood, A.N.: On botnet detection with genetic programming under streaming data label budgets and class imbalance. Swarm Evolut. Comput. **39**, 123–140 (2018)
63. Haddadi, F., Le, D.C., Porter, L., Zincir-Heywood, A.N.: On the effectiveness of different botnet detection approaches. In: International conference on information security practice and experience, pp. 121–135. Springer, New York (2015)
64. Gu, G., Perdisci, R., Zhang, J., Lee, W.: Botminer: clustering analysis of network traffic for protocol- and structure-independent botnet detection. In: Proceedings of the 17th USENIX security symposium, pp. 139–154 (2008)
65. Khan, I.A., Pi, D., Khan, Z.U., Hussain, Y., Nawaz, A.: Hml-ids: a hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems. IEEE Access **7**, 89507–89521 (2019)
66. Makanju, A., Zincir-Heywood, A.N., Kiyomoto, S.: On evolutionary computation for moving target defense in software defined networks. In: Proceedings of the genetic and evolutionary computation conference companion, pp. 287–288. ACM, New York (2017)
67. Sengupta, S,, Chakraborti, T., Kambhampati, S.: Mtdeep: boosting the security of deep neural nets against adversarial attacks with moving target defense. In: Workshops at the thirty-second AAAI conference on artificial intelligence (2018)
68. Le, D.C., Khanchi, S., Zincir-Heywood, A.N., Heywood, M.I.: Benchmarking evolutionary computation approaches to insider threat detection. In: Genetic and evolutionary computation conference (GECCO '18), pp. 1286–1293 (2018). https://doi.org/10.1145/3205455.3205612
69. Rashid, T., Agrafiotis, I., Nurse, J.R.: A new take on detecting insider threats: exploring the use of hidden markov models. In: Proceedings of the 8th ACM CCS international workshop on managing insider security threats, pp. 47–56 (2016). https://doi.org/10.1145/2995959.2995964
70. Chau, M., Chen, H.: A machine learning approach to web page filtering using content and structure analysis. Decis. Support Syst. **44**(2), 482–494 (2008)
71. Xie, J., Yu, F.R., Huang, T., Xie, R., Liu, J., Wang, C., Liu, Y.: A survey of machine learning techniques applied to software defined networking (sdn): research issues and challenges. IEEE Commun. Surv. Tutor. **21**(1), 393–430 (2018)
72. Zhang, C., Patras, P., Haddadi, H.: Deep learning in mobile and wireless networking: a survey. IEEE Commun. Surv. Tutor. **21**(3), 2224–87 (2019)
73. Amiri, R., Almasi, M.A., Andrews, J.G., Mehrpouyan, H.: Reinforcement learning for self organization and power control of two-tier heterogeneous networks. IEEE Trans. Wirel. Commun. **18**(8), 3933–3947 (2019)
74. Moysen, J., Giupponi, L.: From 4G to 5G: self-organized network management meets machine learning. Comput. Commun. **129**, 248–268 (2018)
75. Roy, A., Saxena, N., Sahu, B.J., Singh, S.: Bison: a bioinspired self-organizing network for dynamic auto-configuration in 5g wireless. Wirel. Commun. Mobile Comput. (2018). https://doi.org/10.1155/2018/2632754
76. Wang, H., Wu, Q., Chen, X., Yu, Q., Zheng, Z., Bouguettaya, A.: Adaptive and dynamic service composition via multi-agent reinforcement learning. In: 2014 IEEE international conference on web services, pp. 447–454. IEEE, New York (2014)
77. Valadarsky, A., Schapira, M., Shahaf, D., Tamar, A.: Learning to route. In: Proceedings of the 16th ACM workshop on hot topics in networks, pp. 185–191. ACM, New York (2017)
78. Kim, H.Y., Kim, J.M.: A load balancing scheme based on deep-learning in iot. Clust. Comput. **20**(1), 873–878 (2017)
79. Hajji, H.: Statistical analysis of network traffic for adaptive faults detection. IEEE Trans. Neural Netw. **16**(5), 1053–1063 (2005)
80. Yamanishi, K., Maruyama, Y.: Dynamic syslog mining for network failure monitoring. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 499–508. ACM, New York (2005)

81. Chen, M., Zheng, A.X., Lloyd, J., Jordan, M.I., Brewer, E.: Failure diagnosis using decision trees. In: International conference on autonomic computing, 2004. Proceedings, pp. 36–43. IEEE, New York (2004)
82. Hashmi, U.S., Darbandi, A., Imran, A.: Enabling proactive self-healing by data mining network failure logs. In: 2017 international conference on computing, networking and communications (ICNC), pp. 511–517. IEEE, New York (2017)
83. Zhang, S., Liu, Y., Meng, W., Luo, Z., Bu, J., Yang, S., Liang, P., Pei, D., Xu, J., Zhang, Y., Chen, Y., Dong, H., Qu, X., Song, L.: Prefix: switch failure prediction in datacenter networks. Proc. ACM Meas. Anal. Comput. Syst. **2**(1), 2:1–2:29 (2018)
84. Mismar, F.B., Evans, B.L.: Deep Q-learning for self-organizing networks fault management and radio performance improvement. In: 2018 52nd asilomar conference on signals, systems, and computers, pp. 1457–1461. IEEE, New York (2018)
85. Alshammari, R., Zincir-Heywood, A.N.: Can encrypted traffic be identified without port numbers, IP addresses and payload inspection? Comput. Netw. **55**(6), 1326–1350 (2011). https://doi.org/10.1016/j.comnet.2010.12.002
86. Alshammari, R., Nur Zincir-Heywood, A.: A flow based approach for ssh traffic detection. In: 2007 IEEE international conference on systems, man and cybernetics, pp. 296–301 (2007). https://doi.org/10.1109/ICSMC.2007.4414006
87. Zander, S., Nguyen, T., Armitage, G.: Automated traffic classification and application identification using machine learning. In: Proceedings of the ieee conference on local computer networks 30th anniversary, LCN ’05, pp. 250–257. IEEE Computer Society, Washington, DC (2005). https://doi.org/10.1109/LCN.2005.35
88. Le, D.C., Zincir-Heywood, A.N., Heywood, M.I.: Data analytics on network traffic flows for botnet behaviour detection. In: IEEE symposium series on computational intelligence (SSCI ’16), pp. 1–7 (2016). https://doi.org/10.1109/SSCI.2016.7850078
89. Bernaille, L., Teixeira, R.: Early recognition of encrypted applications. In: Lecture notes in computer science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4427 LNCS, pp. 165–175 (2007). https://doi.org/10.1007/978-3-540-71617-4_17
90. Bacquet, C., Zincir-Heywood, A.N., Heywood, M.I.: Genetic optimization and hierarchical clustering applied to encrypted traffic identification. In: 2011 IEEE symposium on computational intelligence in cyber security (CICS), pp. 194–201 (2011). https://doi.org/10.1109/CICYBS.2011.5949391
91. Silva, J.M.C., Carvalho, P., Lima, S.R.: A modular traffic sampling architecture: bringing versatility and efficiency to massive traffic analysis. J. Netw. Syst. Manag. **25**(3), 643–668 (2017)
92. Hardegen, C., Pfülb, B., Rieger, S., Gepperth, A., Reissmann, S.: Flow-based throughput prediction using deep learning and real-world network traffic. In: International conference on network and service management. IEEE, New York (2019)
93. Mirza, M., Sommers, J., Barford, P., Zhu, X.: A machine learning approach to TCP throughput prediction. ACM SIGMETRICS Perform. Eval. Rev. **35**, 97–108 (2007)
94. Chen, Z., Wen, J., Geng, Y.: Predicting future traffic using hidden markov models. In: 2016 IEEE 24th international conference on network protocols (ICNP), pp. 1–6. IEEE, New York (2016)
95. Kim, S., Kim, D.Y., Park, J.H.: Traffic management in the mobile edge cloud to improve the quality of experience of mobile video. Comput. Commun. **118**, 40–49 (2018)
96. Mijumbi, R., Gorricho, J.L., Serrat, J., Claeys, M., De Turck, F., Latré, S.: Design and evaluation of learning algorithms for dynamic resource management in virtual networks. In: 2014 IEEE network operations and management symposium (NOMS), pp. 1–9. IEEE, New York (2014)
97. Yu, C., Lan, J., Xie, J., Hu, Y.: Qos-aware traffic classification architecture using machine learning and deep packet inspection in SDNS. Procedia Comput. Sci. **13**(1), 1209–1216 (2018)
98. Zhu, G., Zan, J., Yang, Y., Qi, X.: A supervised learning based QoS assurance architecture for 5G networks. IEEE Access **7**, 43598–43606 (2019)
99. Dainotti, A., Pescapé, A., Ventre, G.: A cascade architecture for DoS attacks detection based on the wavelet transform. J. Comput. Secur. **17**(6), 945–968 (2009)
100. Otey, M.E., Ghoting, A., Parthasarathy, S.: Fast distributed outlier detection in mixed-attribute data sets. Data Min. Knowl. Discov. **12**(2–3), 203–228 (2006). https://doi.org/10.1007/s10618-005-0014-6

101. Le, D.C., Zincir-Heywood, A.N.: Evaluating insider threat detection workflow using supervised and unsupervised learning. In: IEEE security and privacy workshops (SPW '18), San Francisco, CA, USA, pp. 270–275 (2018). https://doi.org/10.1109/SPW.2018.00043
102. Le, D.C., Zincir-Heywood, A.N.: Machine learning based insider threat modelling and detection. In: IFIP/IEEE international symposium on integrated network management, Washington DC, USA (2019)
103. Alrawashdeh, K., Purdy, C.: Toward an online anomaly intrusion detection system based on deep learning. In: 2016 15th IEEE international conference on machine learning and applications (ICMLA), pp. 195–200. IEEE, New York (2016)
104. Hofstede, R., Jonker, M., Sperotto, A., Pras, A.: Flow-based web application brute-force attack and compromise detection. J. Netw. Syst. Manag. **25**(4), 735–758 (2017)
105. Haddadi, F., Zincir-Heywood, A.N.: Benchmarking the effect of flow exporters and protocol filters on botnet traffic classification. IEEE Syst. J. **10**(4), 1390–1401 (2016)
106. Abubakar, A., Pranggono, B.: Machine learning based intrusion detection system for software defined networks. In: 2017 seventh international conference on emerging security technologies (EST), pp. 138–143. IEEE, New York (2017)
107. Deshpande, P., Sharma, S.C., Peddoju, S.K., Junaid, S.: Hids: a host based intrusion detection system for cloud computing environment. Int. J. Syst. Assur. Eng. Manag. **9**(3), 567–576 (2018)
108. Nobakht, M., Sivaraman, V., Boreli, R.: A host-based intrusion detection and mitigation framework for smart home iot using openflow. In: 2016 11th international conference on availability, reliability and security (ARES), pp. 147–156. IEEE, New York (2016)
109. Tegeler, F., Fu, X., Vigna, G., Kruegel, C.: Botfinder: Finding bots in network traffic without deep packet inspection. In: Proceedings of the 8th international conference on emerging networking experiments and technologies, pp. 349–360. ACM, New York (2012)
110. Guzella, T.S., Caminhas, W.M.: A review of machine learning approaches to spam filtering. Expert Syst. Appl. **36**(7), 10206–10222 (2009)
111. 5GPPP (2017) Cognitive network management for 5G. White paper, 5GPPP Working Group on Network Management and QoS
112. Boyan, J.A., Littman, M.L.: Packet routing in dynamically changing networks: a reinforcement learning approach. Advances in Neural Information Processing Systems, pp. 671–678. Morgan Kaufmann Publishers, San Mateo (1994)
113. Gomez, C., Shami, A., Wang, X.: Machine learning aided scheme for load balancing in dense iot networks. Sensors **18**(11), 3779 (2018)
114. Qader, K.: The computer network faults classification using a novel hybrid classifier. Ph.D. thesis, University of Portsmouth (2019)
115. Makanju, A., Zincir-Heywood, A.N., Milios, E.E.: Investigating event log analysis with minimum apriori information. In: Proceedings of the IFIP/IEEE international symposium on integrated network management (IM). IEEE, New York (2013)
116. Zakeri, H., Antsaklis, P.J.: A data-driven adaptive controller reconfiguration for fault mitigation: a passivity approach. arXiv preprint arXiv:190209671 (2019)
117. Konecný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: strategies for improving communication efficiency. CoRR abs/1610.05492, arxiv:1610.05492 (2016)
118. Jayaraman, B., Evans, D.: Evaluating differentially private machine learning in practice. In: 28th USENIX security symposium (USENIX Security 19), USENIX Association, Santa Clara, CA, pp. 1895–1912, https://www.usenix.org/conference/usenixsecurity19/presentation/jayaraman (2019)
119. Gentry, C., et al.: Fully homomorphic encryption using ideal lattices. Stoc **9**, 169–178 (2009)
120. Le, D.C., Zincir-Heywood, N.: Big data in network anomaly detection. In: Sakr, S., Zomaya, A. (eds.) Encyclopedia of Big Data Technologies, pp. 1–9. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-63962-8_161-1
121. Kim, B.: Interactive and interpretable machine learning models for human machine collaboration. Ph.D. thesis, Massachusetts Institute of Technology (2015)
122. Warde-Farley, D., Goodfellow, I.: Adversarial perturbations of deep neural networks. In: Hazan, T., Papandreou, G., Tarlow, D. (eds.) Perturbations, Optimization, and Statistics. The MIT Press (2016). https://doi.org/10.7551/mitpress/10761.003.0012
123. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: 2016 IEEE European symposium on security and privacy (EuroS&P), pp. 372–387. IEEE, New York (2016)

124. Rigaki, M., Garcia, S.: Bringing a gan to a knife-fight: Adapting malware communication to avoid detection. In: 2018 IEEE security and privacy workshops (SPW), pp. 70–75. IEEE, New York (2018)
125. Bronfman-Nadas, R., Zincir-Heywood, N., Jacobs, J.T.: An artificial arms race: could it improve mobile malware detectors? In: 2018 network traffic measurement and analysis conference (TMA), (2018). https://doi.org/10.23919/TMA.2018.8506545
126. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:170606083 (2017)
127. Verma, G., Ciftcioglu, E., Sheatsley, R., Chan, K., Scott, L.: Network traffic obfuscation: An adversarial machine learning approach. In: MILCOM 2018-2018 IEEE military communications conference (MILCOM), pp. 1–6. IEEE, New York (2018)
128. Guo, T., Xu, Z., Yao, X., Chen, H., Aberer, K., Funaya, K.: Robust online time series prediction with recurrent neural networks. In: 2016 IEEE international conference on data science and advanced analytics (DSAA), pp. 816–825. IEEE, New York (2016)
129. Le, D.C, Zincir-Heywood, N.: Learning from evolving network data for dependable botnet detection. In: International conference on network and service management (CNSM 2019), Halifax, Canada (2019)

Duc C. Le is a Ph.D. student at Dalhousie University, Halifax, Canada. He received the Master degree in computer science from the same university in 2017, and the B. Eng. degree in electronics and telecommunications engineering from Posts and Telecommunications Institute of Technology, Ha Noi, Vietnam, in 2015. His research focuses on machine learning and its applications in computer and network security and analysis.

Nur Zincir-Heywood is a Full Professor of Computer Science with Dalhousie University, Canada. Her research interests include machine learning and data mining techniques for networks, services and cybersecurity. She has published over 200 fully reviewed papers and has been a recipient of several best paper awards. She is an Associate Editor of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT and is the General Co-Chair of the International Conference on Network and Service Management 2020. She is a member of the IEEE and the ACM and a recipient of the 2017 DNS Women Leaders in the Digital Economy Award.