European Journal of Business & Social Sciences
Available at https://ejbss.org/
ISSN: 2235-767X
Volume 07 Issue 05
May 2019

Machine Learning As Tool In Network Troubleshooting

ShantKaushik,, Ashok Kumar
D.A.V. College, Ambala City, India
shantdavc@gmail.comashokdavc@gmail.com

Abstract

In the era of machine learning, described as ability of computers to learn without actually being programmed for the fixture, there is one area where it can be used as troubleshooter. In this paper an effort has been made to enhance the capabilities of Machine learning as a tool to work in the process of network healing. Though the paper is more of survey study and proposed solutions yet it can serve as basic platform to carry out further research on this topic.

I. Introduction:

In 1959, Arthur Samuel coined the term "Machine Learning", as *"*the field of study that gives computers the ability to learn without being explicitly programmed*"*. The four broad categories of problems have been identified as; (a) *clustering*, (b) *classification*, (c) *regression* and (d) *rule extraction* . In clustering problems, the objective is to group similar data together, while increasing the gap between the groups. Whereas, in classification and regression problems, the goal is to map a set of new input data to a set of discrete or continuous valued output, respectively. Rule extraction problems are intrinsically different, as in this case the goal is to identify statistical relationships in data. ML techniques have been applied to various problem domains. A closely related domain consists of data analysis for large databases, called data mining. Though, ML techniques can be applied to aid in data mining, the goal of data mining problems is to critically and meticulously analyze data—its features, variables, invariants, temporal granularity, probability distributions and their transformations. However, ML goes beyond data mining to predict future events or sequence of events. Machine Learning (ML) has been enjoying an unprecedented surge in applications that solve problems and enable automation in diverse domains. Primarily, this is due to the explosion in the availability of data, significant improvements in ML techniques, and advancement in computing capabilities. Undoubtedly, ML has been applied to various mundane and complex problems arising in network operation and management. There are various surveys on ML for specific areas in networking or for specific network technologies. In this way, readers will benefit from a comprehensive discussion on the different learning paradigms and ML techniques applied to fundamental problems in networking as;

(i) Traffic prediction, routing and classification,
(ii) Congestion control,
(iii)Resource and fault management,

European Journal of Business & Social Sciences
Available at https://ejbss.org/
ISSN: 2235-767X
Volume 07 Issue 05
May 2019

(iv) QoS and QoE management, and network security.
(v) To remove data clogs in Transmission lines
(vi) Remove congestion to achieve Packet delivery
(vii) Carrying out self healing
(viii) Trying to achieve maximum throughput
(ix) Reducing delay if possible
(x) Achieving smooth and stable packet delivery

Therefore, this is a timely contribution of the implications of ML for networking, that is pushing the barriers of autonomic network operation and management.

II. Process Evolution :

Generally, ML is ideal for inferring solutions to problems that have a large representative dataset. In this way, as illustrated in Fig. 1, ML techniques are designed to identify and exploit hidden patterns in data for (i) describing the outcome as a grouping of data for clustering problems, (ii) predicting the outcome of future events for classification and regression problems, and (iii) evaluating the outcome of a sequence of data points for rule extraction problems. Though, the figure illustrates data and outcome in a two-dimensional plane, the discussion holds for multi-dimensional datasets and outcome functions. For instance, in the case of clustering, the outcome can be a non-linear function in a hyperplane that discriminates between groups of data. Networking problems can be formulated as one of these problems that can leverage ML. For example, a classification problem in networking can be formulated to predict the kind of security attack: Denial-of-Service (DoS), User-to-Root (U2R), Root-to-Local (R2L), or probing, given network conditions[9]. Whereas, a regression problem can be formulated to predict of when a future failure will transpire.

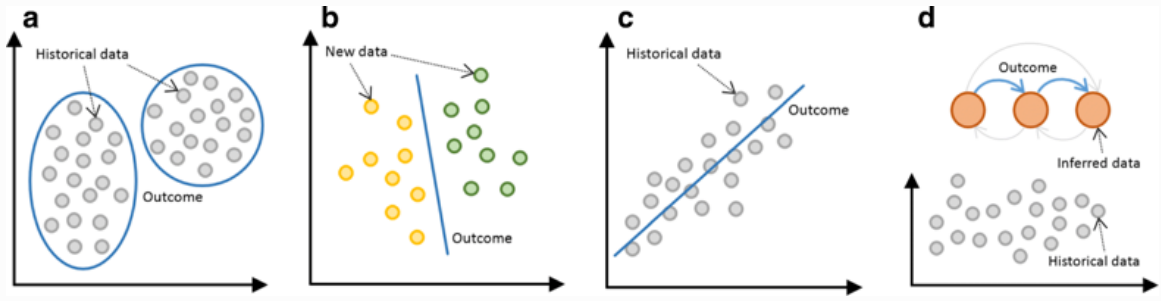

Fig 2: Problem categories **a** Clustering. **b** Classification. **c** Regression. **d** Rule extraction

Though there are different categories of problems that enjoy the benefits of ML, there is a generic approach to building ML-based solutions. Figure 3 illustrates the key constituents in

European Journal of Business & Social Sciences
Available at https://ejbss.org/
ISSN: 2235-767X
Volume 07 Issue 05
May 2019

designing ML-based solutions for networking. *Data collection* pertains to gathering, generating and, or defining the set of data and the set of classes of interest. *Feature engineering* is used to reduce dimensionality in data and identify discriminating features that reduce computational overhead and increase accuracy. Finally, ML techniques carefully analyze the complex inter- and intra-relationships in data and learn a model for the outcome.

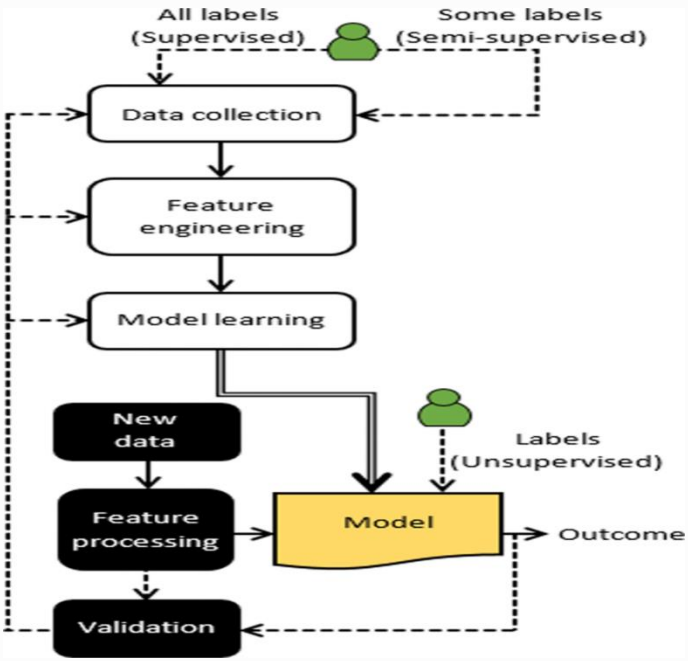

Fig. 3 :constituents of ML-based solutions

For instance, consider an example of Gold values over time, as illustrated in Fig. 2c. Naïvely, a linear regression model, shown as a best-fit line through the historical data, can facilitate in predicting future values of Gold. Therefore, once the ML model is built, it can be deployed to deduce outcomes from new data. However, the outcomes are periodically validated, since they can drift over time, known as concept drifting. This can be used as an indicator for incremental learning and re-training of the ML model. There are four learning paradigms in ML, *supervised*, *unsupervised*, *semi-supervised*and *reinforcement learning*. These paradigms influence data collection, feature engineering, and establishing ground truth. Recall, the objective is to infer an outcome, given some dataset. The dataset used in constructing the ML model is often denoted as training data and labels are associated with training data if the user is aware of the description of the data. The outcome is often perceived as the identification of membership to a class of interest.

III Motivation for the Goal: The process has been directed to achieve the target of networking troubleshooting using Machine learning. ML techniques require representative data, possibly without

bias, to build an effective ML model for a given networking problem. Data collection is an important step, since representative datasets vary not only from one problem to another but also from one time period to the next. In general, data collection can be achieved in two phases—offline and online. Offline data collection allows to gather a large amount of historical data that can be used for model training and testing. Whereas, real-time network data collected in the online phase can be used as feedback to the model, or as input for re-training the model. Offline data can also be obtained from various repositories, given it is relevant to the networking problem being studied.

Examples of these repositories include Waikato Internet Traffic Storage (WITS) [2] , UCI Knowledge Discovery in Databases (KDD) [1]Archive, Measurement and Analysis on the WIDE Internet (MAWI) [4], Working Group Traffic Archive, and Information Marketplace for Policy and Analysis of Cyber-risk & Trust (IMPACT) [3] Archive.

An effective way to collect both offline and online data is by using monitoring and measurement tools. These tools provide greater control in various aspects of data collection, such as data sampling rate, monitoring duration and location (e.g. network core *vs.* network edge). They often use network monitoring protocols, such as Simple Network Management Protocol (SNMP) [5], Cisco NetFlow, and IP Flow Information Export (IPFIX)[6]. However, monitoring can be active or passive. Active monitoring injects measurement traffic, such as probe packets in the network and collects relevant data from this traffic. In contrast, passive monitoring collects data by observing the actual network traffic. Evidently, active monitoring introduces additional overhead due to bandwidth consumption from injected traffic. Whereas, passive monitoring eliminates this overhead, at the expense of additional devices that analyze the network traffic to gather relevant information.

Once data is collected, it is decomposed into training, validation and test datasets. The training set is leveraged to find the ideal parameters (e.g. weights of connections between neurons in a Neural Network (NN)) of a ML model. Whereas, the validation set is used to choose the suitable architecture of a ML model, or choose a model from a pool of ML models. Note, if a ML model and its architecture are pre-selected, there is no need for a validation set. Finally, test set is used to assess the unbiased performance of the selected model. Note, validation and testing can be performed using one of two methods—holdout or *k*-fold cross-validation. In the holdout method, part of the available dataset is set aside and used as a validation (or testing) set. Temporal and spatial robustness of ML model can be evaluated by exposing the model to training and validation datasets that are temporally and spatially distant. For instance, a model that performs well when evaluated with datasets collected a year after being trained or from a different network, exhibits temporal and spatial stability, respectively.

IV. Prospective Solution As Feature Engineering

The collected raw data may be noisy or incomplete. Before using the data for learning, it must go through a pre-processing phase to clean the data. Another important step prior to learning, or training a model, is feature extraction. These features act as discriminators for learning and

inference. In networking, there are many choices of features to choose from. Broadly, they can be categorized based on the level of granularity. Feature engineering is a critical aspect in ML that includes feature selection and extraction. It is used to reduce dimensionality in voluminous data and to identify discriminating features that reduce computational overhead and increase accuracy of ML models. Feature selection is the removal of features that are irrelevant or redundant. Irrelevant features increase computational overhead with marginal to no gain in accuracy, while redundant features promote over-fitting. Feature extraction is often a computationally intensive process of deriving extended or new features from existing features, using techniques, such as entropy, Fourier transform and principal component analysis (PCA).

Features selection and extraction can be performed using tools, such as NetMate and WEKA [7][8].However, in this case, the extraction and selection techniques are limited by the capability of the tool employed. Therefore, often specialized filter, embedded, and wrapper-based methods are employed for feature selection. Filtering prunes out the training data after carefully analyzing the dataset for identifying the irrelevant and redundant features. In contrast, wrapper-based techniques take an iterative approach, using a different subset of features in every iteration to identify the optimal subset. Whereas, embedded methods combine the benefits of filter and wrapper-based methods, and perform feature selection during model creation. Furthermore, it is important to consider the characteristics of the task we are addressing while performing feature engineering. To better illustrate this, consider the following scenario from network traffic classification. One variant of the problem entails the identification of a streaming application (e.g. Netflix) from network traces. Intuitively, average packet-size and packet inter-arrival times are representative features, as they play a dominant role in traffic classification. Average packet size is fairly constant in nature and packet inter-arrival times are a good discriminator for bulk data transfer (e.g. FTP) and streaming applications.

V. Proposed Solution

A rich body of academic work backs the networks used today, certainly, but there is no unifying theory defining how a network, in an abstract sense, should behave, or how it needs to be structured. The networks that form the Internet certainly share some core principles, but they weren't built from a central theory. They emerged through trial-and-error, some good ideas and people telling each other how to do it, This is why machine learning is good at vision-based problems such as image processing or handwriting recognition. Vision can be studied in nature; there's an entire body of theory that can be applied to make a machine behave like an eye. Effort is to figure out if there's some general way to think of a network, Meyer says. If such a thing doesn't exist, then it's possible that every network is a kind of one-off. That would be bad, because it would mean each network has to be learned separately. It wouldn't play into one of the strengths of machine learning—namely, the ability to take that trained neural network and add your own things to it, custom things, Meyer says. The data set and the way you interpret them have to be somewhere similar for you to be able to do that. Lack of a theoretical model is only one obstacle that machine learning faces in networking. The other is people. Machine learning and networking are different skills, and the pipeline of people well versed in both is thin. Even though networking has just massively more compute and massively more data available, it's not yet clear how machine learning can be applied there, Meyer says. With security threats on the rise, many network administrators are exploring

how machine learning and artificial intelligence (AI) can automate threat assessment and response processes to mitigate the risk of data breach. Machine learning algorithms have been tracking credit card fraud for a few years now, so it's not new technology that we're discussing. In fact, the biggest change to network troubleshooting is that machine learning and AI are now getting baked into everything.

VI. Conclusion

While it's clear that machine learning holds the key to the more streamlined data center operations, it may also change the traditional network administrator role also. Monitoring isn't just about the servers and firewalls, of course, it's the cooling mechanisms as well that require 24/7 vigilance. According to Data Center Knowledge, data center infrastructure management (DCIM) software is the latest in machine learning for our increasingly large data centers. Companies like Vigilent offer data centers automated IoT sensors that use machine learning to evaluate the relationship between cooling, power use, and the risk of failure. The technology uses prescriptive analytics to monitor data centers and alert you when a critical cooling infrastructure is about to decline. The machine learning algorithms are so smart it can alert and correct hot spots rack by rack. It can also help data centers become a little greener by network troubleshooting and making minute output adjustments to improve electric usage.

Machine learning can help streamline network troubleshooting, allowing data center managers a more automated and intelligent process. But neither of these examples is effective without a human partnership. IoT sensors cannot hear the sound of a fan nearing the end of its life or the drip drop of a leaking pipe. That's why data center managers will continue to leverage machine learning as the first line of defense in network troubleshooting. The ability to capture, collate, and monitor the huge volume of security data will enable our network analysis of even the largest infrastructures. Whether it's monitoring and responding to a cyber-security breach or placing remote sensors to monitor hardware, machine learning algorithms and the tools they spawn are going to continue to improve DCIM in the future. Data center managers will no longer have to perform data crunching and analysis. The future state of DCIM will likely become a hybrid of AI and human interaction. This new model will not make the human data center manager obsolete, but will instead allow them to adapt to evolving cyber threats while automating and streamlining tasks. Keeping in mind that these advances aren't even in the toddler phase, it's clear that DCIM is changing as machine learning is evolving into a more generally accepted practice across all industries. **As the many changes and potential changes permeate the network troubleshooting paradigm, it becomes vital to ensure networks are kept running.**

References:

[1] R. Ramakrishan and S. Stolfo, editors. Proceedings of the Sixth ACM SIGKDD International Conference on Know ledge Discovery and Data Mining. The Association for Computing Machinery, 2000.

[2] WAND Network Research Group. WITS: Waikato Internet Traffic Storage. http://wand.cs.waikato.ac.nz/wits/ (accessed on 19.03.2009).

[3] Hemraj Saini, Yerra Shankar Rao, T.C.Panda,Cyber-Crimes and their Impacts: A Review / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2,Mar-Apr 2012, pp.202-209 202.

[4] http://www.wide.ad.jp/project/wg/mawi.html

[5] Neha, Mahendra Singh Meena, Rajbir, "Implementationof SNMP(Simple Network Management Protocol) on Sensor Network: A Review,International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 5, Issue 5, May 2016 ISSN:2278-1323, pp.1351-1354.

[6] Brian Trammell ; Elisa Boschi, An introduction to IP flowinformation export (IPFIX), IEEE Communications Magazine (Volume: 49 , Issue: 4 , April 2011) Page(s): 89 - 95.

[7] Rishabh Tulsyan ,Asha.S;"Network Traffic Analysis Using Weka Tool", International Journal of Pure and Applied Mathematics,Volume 118 No. 24 2018 ISSN: 1314-3395 (on-line version) url:http://www.acadpubl.eu/hub/

[8] Jamuna .A, VinodhEwards S.E / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 2, March -April 2013, pp.1324-1328.

[9] Swati Paliwal, Ravindra Gupta;"Denial-of-Service, Probing & Remote to User (R2L) Attack Detection using Genetic Algorithm";International Journal of Computer Applications (0975 – 8887) Volume 60– No.19. December 2012.