Machine Learning Techniques in Network Optimization Problems

Bartosz Rodziewicz
nr indeksu: 226105
Politechnika Wrocławska

Adam Hyjek
nr indeksu: 234987
Politechnika Wrocławska

***Abstract*—The abstract goes here.**

I. Introduction

In recent years we can observe increasing amount of new devices connected to various networks. This trend is expected to increase further with 5G release and IoT concept gaining popularity. This causes increased network traffic leading to exhausting networks' capacity. To prevent that, there are numerous techniques of network optimization. Some of the most popular methods of network optimization discussed in the article are edge caching, real-time routing, load balancing, traffic shaping and data compression.

Edge caching is a single-sided storing of popular, recently viewed content allowing to reduce number of subsequent requests to external network. It allows not only to reduce external traffic, but also decreases response time significantly. Biggest problem in edge caching is predicting future demand for data and choosing what content already existing in cache should be replaced by incoming one. Most popular approaches are two simple algorithms: Least recently used (LRU) and Least frequently used (LFU)[1][2].

Real-time routing is a process where data is forwarded to it's destination based on current network condition, allowing route path to stay relevant even despite constant changes in network condition. There is a big variety in algorithms analyzing current network state such as RIP or EIGRP, but performing optimal routing based only on current network situation is a complex task. Routing results can be improved though by taking into account previous experiences to avoid some recurring problems[3][4].

Load balancing is a process of distributing network traffic across multiple servers to avoid overloading particular server leading to deterioration of service quality or even complete service degradation for some users. Typical algorithms used in load balancing problems are Round Robin algorithms and solutions based on current state's metrics, but just as in routing problem, making decision may be significantly improved by analyzing past circumstances and staying away from former issues[5][6].

Traffic shaping is an optimization technique which delays some of the packets to bring the overall traffic into the desired profile. It is used to optimize performance, increase usable bandwidth or improve latency for other applications and users. Since most of the network traffic is encrypted nowadays, the need to quickly and accurately classify Internet traffic for this purpose has been growing steadily[7]. This is a field where machine learning comes into play and provides a way to classify the encrypted network traffic.

Data compression is a subject that goes far beyond the network optimization problems. It is a process of encoding information using fewer bits than original representation. There are two types of data compression - lossy and lossless. The lossy compression reduce the amount of bits used to store information by removing less important information. The lossless compression reduces bits by identifying and eliminating statistical redundancy. There exists many different data compression algorithm, both lossy and lossless, and some of them use machine learning to increase their efficiency[8][9].

Machine learning is a subset of artificial intelligence. It is a study of algorithms that improve automatically over time. Machine learning algorithms use statistics to find patterns in massive amounts of data and make predictions based on that data. The more data the model has for training, the better the prediction are.

In this article we focus on machine learning usage in network optimization problems mentioned above. The article's goal is to point out increasing popularity of artificial intelligence methods applied to solving network optimization related issues and their success rate. The rest of this survey is organized into 5 sections dedicated to methods of network optimization and section containing our conclusion.

II. Edge Caching

Edge caching strategy depends on solving four fundamental issues: where, how, when and what to cache[10]. First issue, also called Cache placement, can be divided into infrastructure caching and infrastructureless caching. In infrastructureless caching content is stored at user equipment such as mobile devices and home routers. It lowers network load and latency, but drastically reduces amount of users using particular caching solution. On the other hand, infrastructure caching takes place in various parts of public network, for example in Base Transceiver Station (BTS), which causes higher latency when accessing cached data, but increases chances that content is already stored[10].

Caching policy determines how data is cached and is split into proactive and reactive approach. Reactive policy is a simpler one and means that uncached user request is immediately

saved. Most of the time this approach is applied in many layers of network, meaning that data is progressively requested from next network layer if not found locally. Proactive policy is based on predicting popularity and prefetching data probably needed in the future. This approach can potentially greatly increase network quality and decrease its load, but also can result in large decrease of cache hit ratio, depending on algorithm quality. Therefore it's great opportunity to use machine learning algorithms which are characterised by improving self over time[10].

There are many machine learning techniques used in edge caching, such as supervised learning, unsupervised learning, reinforcement learning, neural networks, similarity learning and transfer learning[11].

Supervised learning is used most commonly to predict traffic intensity and content demand and to decide which content should be cached. From supervised learning methods most used are classification and regression analysis[1]. Examples of usage are: predicting popularity of new videos and maximization of cache hit ratio with usage of learn-to-rank algorithm[10].

Most often exploited technique from unsupervised learning for caching solution is clustering, which is used to aggregate user equipments (UEs) into groups based on their previous requests. Thanks to this identification data demands can be predicted by comparing them with other users from the same group[1].

Transfer learning based methods for caching solutions are utilized for estimating content's future demand and appealing based on user-content correlations from another related domains[1]. Transfer learning can be also used for transferring knowledge from relative tasks to speed up learning process and avoid cold start[10].

Reinforcement learning is usually used to solve subproblem of other machine learning based techniques, because of its delayed reward and trial and error approach. Learning process can be modeled as Markov Decision Process and can be used to minimize cache replacement transmission cost or solve cache replacement problem[10].

Deep learning can be used for both edge caching and caching related content delivery. It can provide good results comparing to other machine learning approaches, but on the other hand it may require big amount of data to train the model[1].

Similarity learning is used to create similarity function based on two given pairs: one of similar objects and second consisting objects with lesser level of similarity. Such created function is then used to judge if incoming content is similar to recently popular ones among user equipments[1]. It can also be used to investigate how similar are preferences of users, what then leads to estimating content demand.

III. Real-Time Routing

While network traffic growths rapidly, most of the networks themselves use old routing frameworks designed for fixed networks, which are characterized by relatively stable state of links, which most of the times leads to easy to calculate and predict routes[3]. In wireless networks, in contrast to fixed ones, network link states become much more unstable and it becomes hard to calculate optimal routes quickly.

To evaluate possible routes there is a need to define some Quality of Service metrics that have to be satisfied to ensure proper quality of solution. There are two important routing measures that are used during designing routing protocols: packet delivery ratio (PDR) and bandwidth utilization. Packet delivery ratio is defined as a ratio of number of data packets received to data packets send. Reliable routing protocols should ensure PDR value of 100% or at least nearly 100%. Good bandwidth utilization is achieved by limiting transmission of not necessary data and and network control overhead[12].

The advantage of machine learning methods when applied to real-time routing problem is ability to predict and formulate future routing states, policies and protocols by observing previous ones[13]. Although potential of machine learning methods in routing is very high, there are several problems that need to be addressed when creating such solutions. First of all, it is not trivial to translate routing problems into machine learning solvable problems. Secondly, machine learning algorithms need to be trained, what creates computational delay, which is undesirable regarding short flow transmission time. Last, but not least, computational power of machine learning approach may not meet requirements of network routing optimization's needs such as accuracy, robustness and scalability[13].

Among machine learning methods that are being applied to real-time routing protocols we can mention supervised learning, unsupervised learning and reinforcement learning[12]. While in supervised learning samples carry label information, in unsupervised learning sample data is unlabeled. Supervised learning is used in classification, decision making and regression, while unsupervised learning is utilized in data clustering and data dimensionality reduction. In opposition to two already mentioned approaches, reinforcement learning determines what should be done based on obtained data. Taken actions are then judged by received reward, what allows to train the model. Trained model can then be used for prediction[13].

Supervised learning algorithms for real-time routing optimization problems uses network and traffic states as input of the training datasets, while routing solutions delivered by corresponding heuristic algorithm are the output. Such routing architecture consists of machine learning based meta-layer and heuristic algorithm layer, which output is used for training machine learning layer. After training phase, routing decision are taken by machine-learning based meta-layer independently and optimal heuristic-like solutions can be obtained then in real-time.[12].

Unsupervised learning is often organized as Self-Organizing Map (SOM) and is used to reduce dimensionality and data clustering. SOM consist of an input layer and map layer. Map layer include neurons with weight vectors, which are being compared with input data sample. Vector with highest similarity and its neighbourhood is then adjusted towards input data. Whole process is repeated until convergence condition is met[12].

Reinforcements learning based algorithms in general are used to solve decision making problems. In routing optimization problem for SDN networks controller takes agent role, while network is described as environment, state space is composed of network and traffic states and routing solution is described as action. Reward in such system is based on chosen optimization metric. Such constructed model can work efficiently even in highly chaotic environments[12].

Traffic prediction's goal is to predict trend of traffic by analysis past traffic information. It allows to establish proactive routing policies to foresee incoming traffic and avoid traffic congestion, which in result improves network's Quality of Service. Most commonly machine learning solution for traffic prediction are neural networks predicting network traffic load. This prediction is used to calculate optimal resource allocation using non-machine learning algorithms[12].

IV. Load Balancing

Load balancing is a technique with much broader scope than just its use by Internet services. However this paper focuses on that and how Internet services can benefit from its applications. The main ways of approaching it in this scenario are: the simple DNS based and the more complex server-side based.

In the DNS-based load balancing technique one domain is assigned to multiple IP addresses (representing multiple servers) and DNS server randomly chooses the server that will be provided to the client. In average usage this method offers similar split of traffic between the servers but there is no guarantee for that. Different factors may degrade the quality of this technique, such as DNS record caching, failure of one server or big differences in load caused by individual requests[14].

The server-side based technique is usually achieved by usage of specialized software created for load balancing which works as a kind of proxy for communication between a client and a chosen server. This method provides a way to have a full control over how load balancing is done and aspects such as scheduling algorithm or persistence. This is also a technique that could benefit from machine learning in some of its layers[6][15].

The classic approach to server-side load balancing is a software that uses a heuristic algorithm to decide which server to choose based on server stats. This data can consist of its capacity, current load, CPU utilization and many more[5]. The machine learning approaches often suggest changing the heuristic algorithm to the one based on machine learning which can make better prediction based on this data[15].

The advantage of machine learning techniques when applied to load balancers is ability to predict how the utilization of particular server will change in the future by analyzing the past. Some of the algorithms can also analyze how different types of request cause change in the server load. The data used for analysis are machine learning friendly which makes machine learning solutions suitable for this task[15]. There are few machine learning techniques used among load balancing solutions that are worth mentioning: supervised[15][16], unsupervised[16][17] and reinforced[18] learning. The supervised learning takes advantage of historical labeled data, while unsupervised can discover the hidden pattern behind the selected features.

The supervised learning can be used for prediction of amount of traffic coming to the servers and to predict which type of request will use up more load on the server[16]. The unsupervised and reinforced learning is used for analysis of server stats and making decision which server should get the request. The unsupervised one creates the patterns between server load utilization and ability to take the request while reinforced learning additional benefit from the system of rewards. The system of rewards in reinforced learning technique often is based on algorithm's ability to keep the load evenly split between the servers[16][18].

Other worth mentioning machine learning approach to the load balancing problem is ML algorithm that doesn't split load by itself but focuses on selecting the best heuristic load balancing algorithm for given circumstances and specific use. This method mostly uses supervised learning and few heuristic algorithms[19].

V. Traffic Shaping

Traffic shaping techniques require fast and accurate methods for classification of the traffic flow. Some time ago data traffic could be classified based on just port number or protocol. However since most of the traffic is encrypted nowadays, new traffic classification methods are required[7][20].

Machine learning based approach is independent from packet payload inspection and provides a way to analyze and classify encrypted data which can't be easily distinguished with typical properties. Because of that majority of modern traffic classification algorithms rely on machine learning. Most popular machine learning techniques used in traffic classification are supervised[7] and unsupervised[21] learning and both of them have its own pros and cons[20].

Supervised machine learning algorithms require a training phase on previously labeled traffic flows to create connection between traffic classes and applications. Because of that, supervised machine learning can be useful for the identification of specific set of application. However, it is worth mentioning that supervised learning algorithm works best when trained on data sets containing all the traffic classes the algorithm is expected to see in practise. Its performance may degrade when the algorithm is trained on data set vastly different than the traffic flows seen when operating. When evaluating supervised learning algorithm for potential use it is worth considering how

the classifier will be re-train if needed and how new type of applications will be detected[20].

One of potential benefit of unsupervised machine learning algorithm may be automatic discovery of traffic classes (clusters) in the dataset. However these automatically created classes still have to be labelled and mapped to application (mostly through inspection by human expert). Another benefit may be creation of a new traffic class when a new application starts to play significant role on the market yet before it spotted by human experts. On the other hand big issue when using unsupervised ML algorithms is that clusters often don't map 1:1 to application. It is common that one application may spread over few clusters and dominate them while another application spread over but doesn't dominate any cluster. When evaluating unsupervised learning algorithm for potential use it is worth considering how clusters will be mapped to specific application, how new applications will be detected and labels updated and what is the optimal number of clusters[20].

Another challenges that these algorithms have to overcome are timely and continuous classification, directional neutrality, efficient use of memory and processors, portability and robustness.

VI. Data Compression

Data compression is a topic that is only briefly related to a network optimization and a field that doesn't use many machine learning algorithms. However, it was chosen to be included in this paper because it is a field which may start to play bigger role as development and usage of IoT devices progresses.

VII. Conclusions

Acknowledgment

References

[1] Z. Chang, L. Lei, Z. Zhou, S. Mao and T. Ristaniemi, "Learn to Cache: Machine Learning for Network Edge Caching in the Big Data Era," in IEEE Wireless Communications, vol. 25, no. 3, pp. 28-35, JUNE 2018, doi: 10.1109/MWC.2018.1700317.

[2] H. Zhu, Y. Cao, W. Wang, T. Jiang and S. Jin, "Deep Reinforcement Learning for Mobile Edge Caching: Review, New Features, and Open Issues," in IEEE Network, vol. 32, no. 6, pp. 50-57, November/December 2018, doi: 10.1109/MNET.2018.1800109.

[3] F. Tang et al., "On Removing Routing Protocol from Future Wireless Networks: A Real-time Deep Learning Approach for Intelligent Traffic Control," in IEEE Wireless Communications, vol. 25, no. 1, pp. 154-160, February 2018, doi: 10.1109/MWC.2017.1700244.

[4] S. Troia et al., "Machine-Learning-Assisted Routing in SDN-Based Optical Networks," 2018 European Conference on Optical Communication (ECOC), Rome, 2018, pp. 1-3, doi: 10.1109/ECOC.2018.8535437.

[5] R. A. Haidri, C. P. Katti and P. C. Saxena, "A load balancing strategy for Cloud Computing environment," 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014), Ajmer, 2014, pp. 636-641, doi: 10.1109/ICSPCT.2014.6884914.

[6] Anna Victoria C R Oikawa, Vinicius Freitas, Márcio Castro, Laércio Lima Pilla. Adaptive Load Balancing based on Machine Learning for Iterative Parallel Applications. 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Mar 2020, Västerås, Sweden. ff10.1109/PDP50117.2020.00021ff. ffhal-02570549f

[7] T. S. Tabatabaei, F. Karray and M. Kamel, "Early internet traffic recognition based on machine learning methods," 2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Montreal, QC, 2012, pp. 1-5, doi: 10.1109/CCECE.2012.6335034.

[8] F. H. Kingma, P. Abbeel and J. Ho, "Bit-Swap: Recursive Bits-Back Coding for Lossless Compression with Hierarchical Latent Variables" 2019.

[9] J. Park, H. Park and Y. Choi, "Data compression and prediction using machine learning for industrial IoT," 2018 International Conference on Information Networking (ICOIN), Chiang Mai, 2018, pp. 818-820, doi: 10.1109/ICOIN.2018.8343232.

[10] Shuja, Junaid, K. Bilal, Eisa A. Alanazi, W. Alasmary and A. Alashaikh. "Applying Machine Learning Techniques for Caching in Edge Networks: A Comprehensive Survey." ArXiv abs/2006.16864 (2020): n. pag.

[11] Y. Sun, M. Peng, Y. Zhou, Y. Huang and S. Mao, "Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues," in IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3072-3108, Fourthquarter 2019, doi: 10.1109/COMST.2019.2924243.

[12] O. Ashour, M. St-Hilaire, T. Kunz and M. Wang, "A Survey of Applying Reinforcement Learning Techniques to Multicast Routing," 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York City, NY, USA, 2019, pp. 1145-1151, doi: 10.1109/UEMCON47517.2019.8993014.

[13] K. Yu, L. Tan, X. Wu and Z. Gai, "Machine Learning Driven Network Routing," 2019 6th International Conference on Systems and Informatics (ICSAI), Shanghai, China, 2019, pp. 705-712, doi: 10.1109/ICSAI48974.2019.9010507.

[14] Brisco, T., "DNS Support for Load Balancing", RFC 1794, doi: 10.17487/RFC1794, April 1995.

[15] S. Parida and B. Panchal, "An Efficient Dynamic Load Balancing Algorithm Using Machine Learning Technique in Cloud Environment", International journal of scientific research in science, engineering and technology, vol. 4, p. 1184-1186, 2018.

[16] C. Gomez, A. Shami and X. Wang, "Machine Learning Aided Scheme for Load Balancing in Dense IoT Networks", Sensors, vol. 18, p. 3779, 2018.

[17] A. Revar, M. Andhariya, D. Sutariya and M. Bhavsar. "Load Balancing in Grid Environment using Machine Learning - Innovative Approach." International Journal of Computer Applications, vol. 8, p. 24-28, 2010.

[18] A. Schaerf, Y. Shoham and M. Tennenholtz, "Adaptive Load Balancing: A Study in Multi-Agent Learning", Journal of Artificial Intelligence Research, vol. 2, doi: 10.1613/jair.121, 1995.

[19] C. R. Anna Victoria Oikawa, V. Freitas, M. Castro and L. L. Pilla, "Adaptive Load Balancing based on Machine Learning for Iterative Parallel Applications," 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Västerås, Sweden, 2020, pp. 94-101, doi: 10.1109/PDP50117.2020.00021.

[20] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," in IEEE Communications Surveys & Tutorials, vol. 10, no. 4, pp. 56-76, Fourth Quarter 2008, doi: 10.1109/SURV.2008.080406.

[21] S. Zander, T. Nguyen and G. Armitage, "Automated traffic classification and application identification using machine learning," The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)l, Sydney, NSW, 2005, pp. 250-257, doi: 10.1109/LCN.2005.35.