

Application of Machine Learning Technologies for Transport layer Congestion Control

¹Madeleine Wang Yue Dong and ²Yannis Yortsos

^{1,2}School of Design, University of Washington, Seattle, WA.

¹yuedongwang@hotmail.com

Article Info

Journal of Computing and Natural Science (<http://anapub.co.ke/journals/jcns/jcns.html>)

Doi: <https://doi.org/10.53759/181X/JCNS202202010>

Received 12 December 2021; Revised form 20 March 2022; Accepted 25 March 2022.

Available online 05 April 2022.

©2022 Published by AnaPub Publications.

Abstract – Due to the advent of technology, humans now live in the modern age of information and data. In this form of world, different objects are interlinked to data sources, and every aspect of human's lives are recorded in a digital form. For example, the present electronic globe has an abundance of distinct forms of data e.g., health data, social media data, smartphone data, business data, smart city data, cybersecurity data and Internet of Things (IoT) data, including Covid-19 data. Data can be unstructured, semi-structured and structured, and this is increasing on a daily basis. Machine Learning (ML) is significantly employed in different aspects of real-life e.g., Congestion Control (CC). This paper provides an evaluation of the aspect ML employed in CC. CC has emerged as a fundamental viewpoint in communications system infrastructure in the recent years, since network operations, and network capacity have enhanced at a rapid rate.

Keywords – Congestion Control (CC), Network Congestion (NC), Machine Learning (ML), Transmission Control Protocol (TCP), Congestion Window (CW).

I. INTRODUCTION

There are a variety of intelligent apps that may be developed by analysing data. This data may be utilised, for example, to construct a cyber security system that is both automated and intelligent, as well as other smart mobile apps that are tailored to a user's specific needs. As a result, real-world applications rely on data handling tools and processes that can quickly and intelligently extract insights or usable information from data.

Machine Learning (ML) is a subset of AI that has expanded quickly in recent years as a result of advances in data processing and computation that enable software programmes to behave intelligently. The types of ML have been illustrated in **Fig. 1**. The most popular technological advances in the 4th industrial revolution, ML is a system that can learn and improve from experience without having to be specially designed (4IR – 4th Industrial Revolution or Industry 4.0) [1]. Automation of traditional industrial and manufacturing operations, including experimental information processing, is generally referred to as "Industry 4.0." A ML technique is needed to assess these datasets and construct real-world applications that make use of them. Reinforcement learning, semi-supervised, unsupervised and supervised are the four main kinds of algorithms that may be found in this field.

Efficiency and effectiveness of a deep learning technology are often determined by data quality and algorithmic competence. Categorization study, extrapolation, data grouping, features design and dimensional decrease, affiliation rules training, or reinforced learning methods may be used to develop successful data-driven systems. There is also a larger family of deep learning algorithms called deep learning, which emerged from the deep neural network. As a result, it's difficult to pick a learning algorithm that's appropriate for a specific domain. For this reason, even within a single classification, the results of various learning methods may differ depending on the qualities of the information. There are several real-world implementations of deep learning, including IoT systems, cybersecurity applications, business and recommendations mechanisms, intelligent towns, medical and COVID-19, context-aware mechanisms, sustainable farming and many more.

New system topologies, e.g., millimeter-wave (mmWave) systems, ultra-dense broadband communication, and cognitive radio network, have emerged as a result of the fast growth of communication technology. Each system has its own unique characteristics and performance needs, which may vary over time. Additionally, new applications and services, such as Augmented Reality (AR), on-line gaming, cloud technologies, and automated driving, need increasingly demanding constraints on the communication system. End-to-end connectivity for upper-layer services are managed in large part by the transport layer. The performance of developing new applications is strongly dependent on the interconnections between the network and the transportation layer.

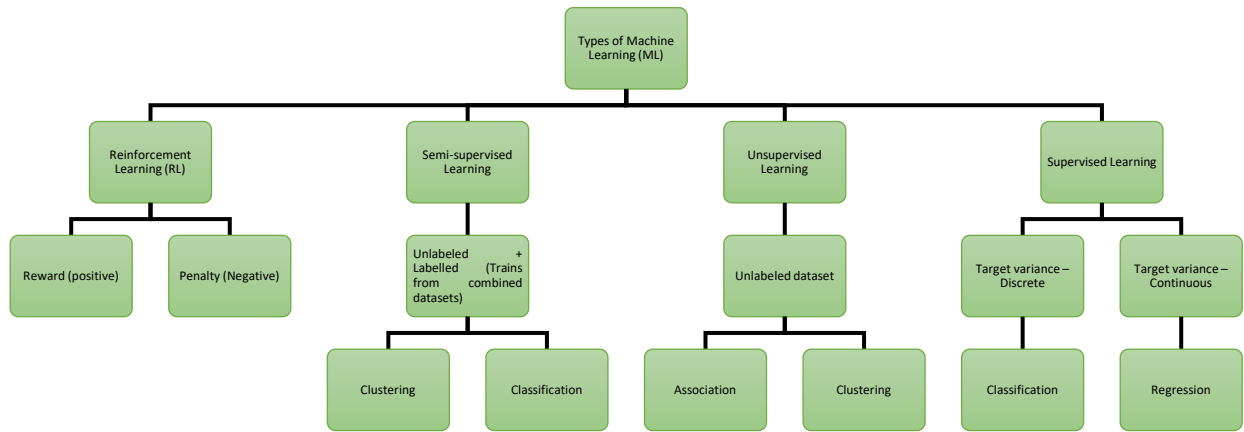


Fig 1. Types of ML

In this study, we concentrate on Machine Learning (ML) application to Congestion Control (CC). This paper has been organized as follows: Section II presents the background analysis of the paper. Section III discusses CC using ML. Section IV reviews the developments on ML-based CC. Section V presents a discussion of CC with Deep Reinforcement Learning (DRL). Section VI evaluates the challenges and outlook on ML’s application in CC. Lastly Section VII concludes the paper.

II. BACKGROUND ANALYSIS

A network's efficiency and reliability may be harmed if the buying power of its resources exceeds the maximum capacity, which may lead to congestion. The Transmission Control Protocol (TCP) [2] foundational congestion method guarantees system reliability and fairness in the utilization of resources. Congestion Control (CC) [3] mechanisms for wired networks were developed in the 1980s and are still used today. There are predefined rules that govern how the Congestion Window (CW) [4] is reduced when packet reduction is identified, and how it is adjusted based on Round-Trip Time (RTT) measurements. Since its inception, the TCP protocol and its variations have been widely used. CC systems like this one have had huge impact, but they may not be able to handle today or future complex and dynamic systems. Conventional rule-based approaches to CC are heuristics with no assurance of solving the complex problem of CC. In many cases, they lead to suboptimal solutions that may not perform as well as they should.

Many applications like speech identification, machine vision, and robot navigation have recently seen significant advancements in the field of ML. The obtained information or environment can be used to create a model by ML. The recent innovation of computing systems (e.g., GPU, ML, and TPU libraries) as well as distributed database frameworks has led to an increasing trend of leveraging ML to deal with difficult networking issues. This trend will only continue. ML performs fairly well in some areas, such as stagnation, categorization, and decision making. These tasks play a fundamental but critical role in networking problems; therefore, it is more imperative to employ ML approaches for end-to-end traffic management techniques.

Network Traffic Control (NTC) and artificial intelligence are linked, as is stated in. Deep Reinforcement Learning (DRL) is a viable paradigm for AI-NTC, according to the author's study. Network structure and optimizations are examined in the reference. The author outlines a process for implementing ML in a network setting. For efficient network traffic management systems, reference gives an outline of the current state in ML, a branch of ML. CC is included into TCP, which is meant to deliver packets reliably throughout an end-to-end connection. CC in TCP is based on the following principles: In order to maximize system resources, the endpoint should ramp up its transmission rate quickly when it's activated. Endpoints should decrease their sending rates when crowding is detected, but when traffic delays are gone, nodes should increase their sending rates for maximum network bandwidth usage. When the network is congested, the rate tends to rise or fall in tandem with the rate. Network Control (NC) is often detected by measuring loss or delay at the network edge, without coordinating or communicating with other users in the system.

CW or mailing rate is adjusted in accordance with predetermined rules based on signals used to detect congestion. CC methods can be classified into numerous classes based on impulse types employed as overcrowding indicators. Loss-centric approaches integrate New Reno, Reno, and TCP variations Tahoe. Packet loss is employed by these approaches for detection. The CW is adjusted using the AIMD algorithm, which increases the CW and decreases it. There are times when the bottleneck buffer is complete even before there is a loss in traffic. It is possible to avoid a significant decrease in throughput if congestion is detected early enough. As a result, congestion-indicating protocols like TCP Vegas and Verus, which use postponement as a metric, have been created.

In recent years, some hybrid techniques have attempted to merge the advantages of various existing remedies. " In the case of Compound TCP, the CW is the sum of the postponement and loss windows, which is accessible on all Windows machines. Veno, a mashup of Reno and Vegas, employs a more explicit framework of bottleneck buffer habitation in order

to better understand congestion. According to Ancillotti and Bruno [5], BBR anticipates both bandwidth and RTT in order to maintain CW at its optimal operating point, which is equivalent to the Bandwidth-Delay Product (BDP).

There are a number of drawbacks to rule-based methods: new platforms may not benefit from CC techniques that were developed for a single network type. On the wired lines where packet loss might be mistaken for congestion, TCP-NewReno was created. It is possible to lose packets in wireless networks due to either connection failures or congestion. Although link faults may be at blame, the transmitter will always half the CW regardless. As a consequence, the link's bandwidth is underutilized as a result. In addition, the CW update method does not have the ability to adapt to different network configurations." When the RTT is high over a satellite network connection, a more vigorous CW increase is needed. Whereas, in MANETs, when the BDP is low, the CW rise may be more conservative.

To use the rule-based strategy, you must follow a set of predetermined guidelines. It makes no use of previous experience and makes no assumptions about things like the bandwidth of a connection, the attributes of a channel, or the number of flows over a shared link. For better connection use, NewReno may be able to train the bandwidth data from past encounters and hence increase the network throughput more aggressively at sluggish start. It is difficult for end devices to take action in ahead since they lack the capacity to train from the past. The performance of today's system is being impacted by its increasing complexity and dynamic nature. Using rules that take into account how people think about networks, ruled-based CC is created. It is possible that the CC mechanism developed as a consequence of human understanding may not be successful or may only be able to attain a sub-optimal level of performance.

III. CONGESTION CONTROL USING MACHINE LEARNING

When the system (or, the agent) has been trained, it is able to do a task when presented with new knowledge or in a new environment. Most ML approaches fall into one of three classifications: controlled training, unsupervised classification, or Reinforcement Learning (RL).

Supervised ML

We train a framework in supervised methods using a labelled database, which implies that we have both the raw input data and the outcomes. Rather than using the same data for training and testing, we create two separate sets of data: one for training and the other for evaluating our model's correctness. Because the instructor has previously seen the outcomes, our model is able to learn from them in the same way that a student is able to learn from their teacher. It is via the use of supervised learning that we are able to obtain great levels of accuracy. Because we already know the intended outcomes in our dataset, the training period for the algorithm is less, resulting in faster performance. This model is capable of accurately predicting outcomes based on previously unobserved or brand-new data. The output of certain supervised learning models is reverted back to learn more so that the maximum accuracy may be achieved. An anticipated value might be discreetly showing a classification or it can be an actual or continuous value for each input instance {see Fig. 2}. Once trained, the algorithm can anticipate the proper output from an unknown input based on patterns learned from training data.

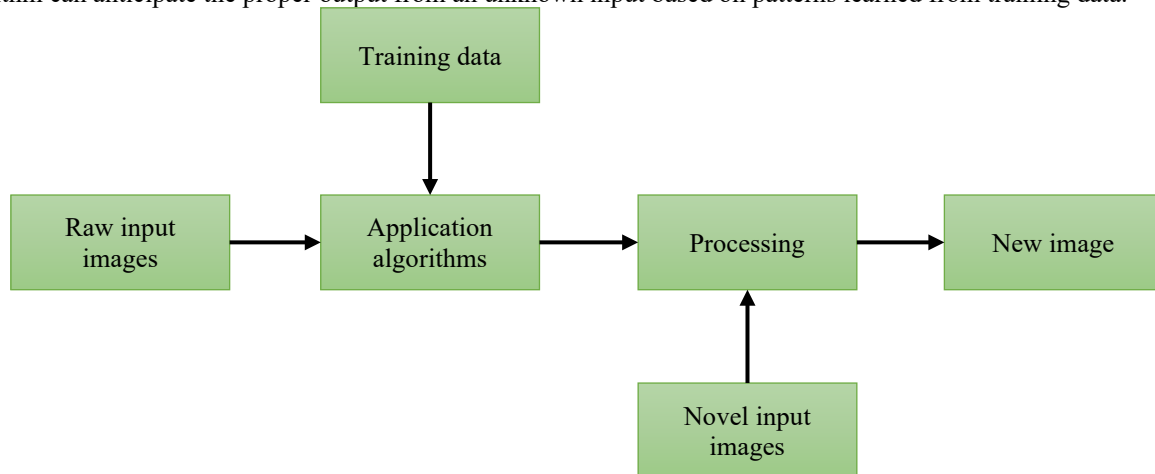


Fig 2. Analytical steps in supervised ML

Using the picture of an apple as an example, you could see why we are giving the algorithm raw inputs like this one in which we have a controller who corrects the machine, trains the machine, and tells it that yes, that is an apple. Once the algorithm has been trained, it can predict the proper output of a never-before-seen input with high accuracy. Some of the supervised learning techniques include Support Vector Machine (SVM), Random Forest (RF), and Regression Learning (RL).

Application

Sentiment Analysis

As part of natural language processing, this technique analyses and categorises the data supplied in the form of text in order to extract significance. If a post is classified as a question, grievance, suggestion, opinion, or piece of news, we may utilise sentiment categorization to make that determination.

Recommendations

Classification is a common strategy used by online retailers and media outlets to promote products and new launches based on customer and user activity. Companies such as Netflix, Flipkart, YouTube, and Amazon are making a lot of money because to their recommendation systems.

Spam Filtration

Any virus, spyware, or even harmful URLs may be detected in spam emails by this screening method. More over half of all mails sent over the web in March 2017 were junk, a marked decline from the 71% proportion of spam in April 2014, as per recent polls.

Unsupervised ML

For the purposes of training, no previous categorization or labelling of the data is performed. To understand how a system could infer a functionality from unmarked data to describe a hidden structure. Unsupervised categorization is primarily concerned with identifying patterns in the data. A system that has learnt to recognise patterns in prior datasets may forecast each new dataset via clustering. Inferences from unstructured datasets enable the system to identify hidden characteristics in unstructured data. Because it lacks a labelled dataset, your model won't be able to tell the difference between apples and cheese if you give it raw input data. It's possible, though, that new information will be able to fit right in with the existing groups. Principal Component Analysis (PCA), k-Means, and Singular Value Deposition (SVD) are examples of unsupervised learning techniques.

Application

Document Clustering

We use techniques like K-means to extract data from a piece of textual material, and to organise the content, determine subjects, and remove unnecessary data.

Data Reduction

Since ML approaches break down and the information no more connects with each other over a lengthy period of time, it is difficult to show and comprehend large datasets. In order to circumvent dimension-related difficulties, unsupervised techniques such as singular value decomposition and PCA are employed.

Anomaly Detection

When it comes to spotting any form of fraud, unsupervised learning is used since it may help by evaluating aberrant data items in the data. In addition, it may be used to identify all of the information that sticks out from the rest, known as outlier detection.

Reinforcement ML

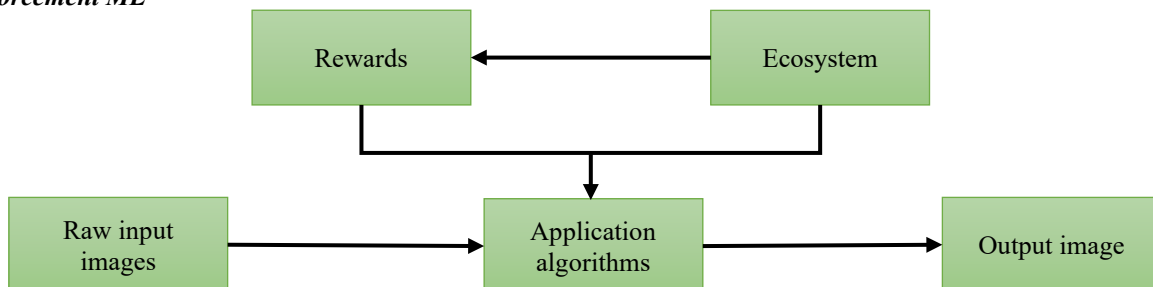


Fig 3. Representation of Reinforcement Learning (RL)

As a result of a ML algorithm, software agents and computers may autonomously choose the best course of action in a given situation. Learning from experience is the only method to complete a particular job in the absence of a dataset or outcomes connected with data. Positive reinforcement is given to an algorithm for every successful action or choice, whereas negative reinforcement is given to an algorithm for every unsuccessful action or decision. So, in this manner, the system has learned which activities are necessary and which are not. The gaming industry, as well as industrial automation, may both benefit from reinforcement learning.

In order to better understand Reinforcement Learning as shown in Fig. 3, it is fundamental to investigate the scenario in which the agent rewards itself for right actions or predictions. For an agent to be successful, he or she has to be able to earn more and more money. Finally, we'll choose the best feasible model based on the performance of the agents in the environment.

Application

Robotics

The growth of robotics relies on the application of reinforcement learning. In the assumption of reinforcement learning, these approaches are used to teach robots to learn from experience.

Traffic Control

The Reinforcement Learning approach used in the traffic control system offered better outcomes than the old way for the issue of congestion (see Fig. 4). When it comes to customized recommendations, Deep Reinforcement learning outperformed other ML models. It also performed well when it came to news suggestions, which included a variety of difficult problems such as dynamic news and a high click-through ratio. A wide range of complicated issues such as end-to-end CC may be addressed using the most recent achievements in deep learning, Generative Adversarial Networks (GAN), and transferred learning.

Only a limited number of metrics, such as packet loss and RTT, are considered in conventional CC. These evaluations and the pre-defined criteria related to human comprehension of the network are critical to the decision-making process. The rule-based approach suffers from poor performance since it is more vulnerable to a wide range of unknown events. When it comes to ML, on the other hand, the goal is to create programs or algorithms that can learn from their prior experiences or the networks around them and make judgments accordingly. A precise network model isn't required. As a result, it has the ability to surpass the rule-based approach.

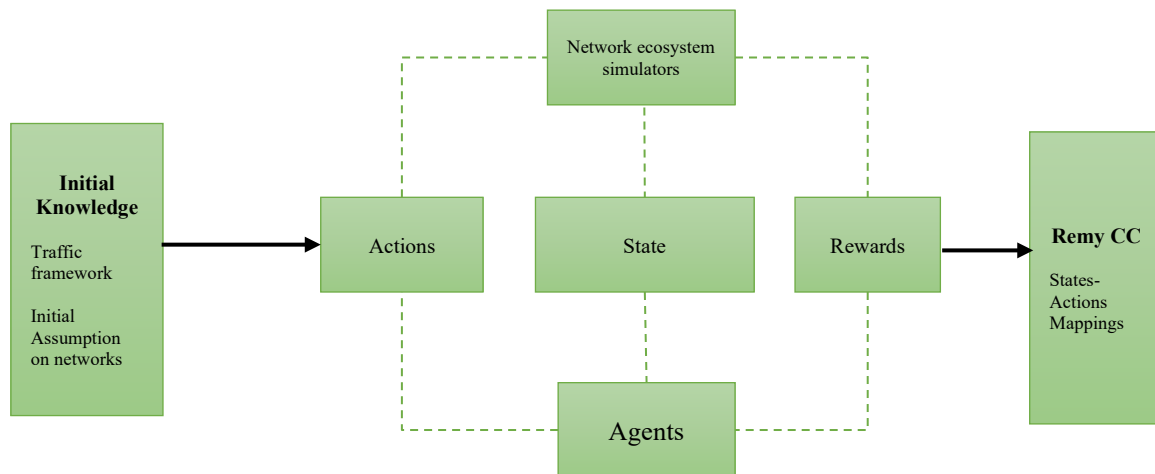


Fig 4. Depiction of Remy congestion management

The centralized controller and peripheral servers are now providing a wide range of processing power. Network slicing, software-defined network, and other cutting-edge technologies are reshaping how network traffic is handled. To make things even easier, libraries like TensorFlow, Caffe, and PyTorch have made it faster and easier to create ML models. It is now possible to use these computing resources to implement ML-based control systems for management purposes. A model may be trained over a lengthy period of time using global information and compute resources in practice. It is possible to change the parameters of the model at some point. The choice may be deduced with a few calculations using the learned ML model in the online installation step.

IV. DEVELOPMENTS ON ML-BASED CONGESTION CONTROL

Fida, Ocampo, and Elmokashfi [6] have worked hard to date to find a solution to the ICWN NC issue. Node buffers and encounter probabilities were combined in a novel way by researchers. A node may prevent node congestion by dynamically adjusting packet quotas based on the buffer condition of neighboring nodes. Congestion status may be avoided by using this strategy. When a node enters the congested state, this approach can only determine which packets to delete based on hop counts, which is unreasonable. The direct occurrence probability alone is not enough to accurately predict the node's capacity to forward packets.

Node [7] formulated an approach, which chooses a relay node based on the node connections in ICWN to maximize the social qualities of nodes. As a result, too many duplicate versions of a message are generated because of the multiple connections between nodes. As a result, NC is a simple and common problem. There were three fundamental parameters

that were estimated using the node's past encounter data: (i.e., probability of head of line deletion, reliability and blocking in [8]). These parameters are used by nodes to make choices.

When it comes to congestion, it's important to keep in mind that researchers are looking at both the local situation and the overall NC. Local buffer data dispersion determines the priority of forwarding requests. Active response may also be used to remove duplicate messages from a stream of transmission. A single node's congestion status does not provide information about the overall network condition. Nodes' congestion state is evaluated using "interest return" and "opportunity expenditure" by [9] to determine whether or not to accept messages. Some NC may be relieved, and performance can be improved, by using this approach. There is a problem, though, since it has a set congestion threshold.

A number of studies on ML-based CC have been published in the last several years. Using TCP Remy [10], the authors design CC as a decision-making issue under uncertainty, the first time this has been done. If a packet should be sent, it may be decided on by each endpoint. An EWMA of ACK intra-arrival time, an EWMA of ACK sending times, and a proportion of the most current RTT to the lowest observed RTT make up the network's state. Unicast connections are turned on and off randomly amongst sender-receiver pairs in the traffic paradigm. According to existing network conditions, the agent adjusts CW to achieve the right balance between output efficiency and decreased latency. Cubic, Compound, and Vegas were all built by humans, yet this ML-based method trumps them all. TCP Remy, on the other hand, works well, but only if the networking and traffic assumptions are rigorously assumed.

Performance-oriented Congestion Control (PCC) employs on-line training instead of off-line teaching and does not assume the same network model assumptions as TCP Remy. Sending rate and the Selected ACK (SACK) are employed in each micro-experiment by PCC to quantify the usefulness of an operation (delivery ACK, delay and loss). To maximise utility, PCC runs a series of micro-experiments and develops an experimentally optimal rate control approach while it is running. PCC outperforms TCP CUBIC by a factor of ten, with more fairness and stability. There have been no tests done in real-world network situations, therefore its performance across wireless networks is hampered by bufferbloat.

V. CONGESTION CONTROL WITH DRL

DRL Definition

It is fundamental to think about Deep Reinforcement Learning (DRL) as a way for learning to consistently make the right judgments over time. People use it from the minute they are born to go across the planet. Parents who praise their children for smiling teach them that it is an effective way to get attention. Crying also teaches children that they need a parent's attention, which may be used to alleviate discomfort or offer comfort. For example, a child's actions are rewarded or punished depending on whether or not their parents provide their approval or attention to them. The learning process of a youngster, however, does not come to an end there. A fresh set of activities is offered to the infant dependent on the response of its parents. During the process of reinforcement learning, the "policy" that guides the baby's choices is referred to. An infant's method of achieving his or her objectives. A "state" in reinforcement learning is an "activity" that the infant does as a result of their present circumstance.

This basic statement reveals numerous intricacies, including the setting in which the infant makes their decisions, the number of accessible possibilities (action sets), and the thoughts underlying the selections (policies). Many alternative representations and methods of instruction exist for each of these elements. Recently, one of the most widely used methods for determining the environment, rules, rewards or punishments and other elements in reinforcement learning has been using deep learning. However, we must first comprehend the subject's antecedents in order to fully describe its complexities.

Machine Learning (ML) refers to the use and development of computer systems that are able to learn and adapt without being explicitly programmed. The algorithms and statistical models used to evaluate and derive conclusions from patterns in data are instead based on algorithms and statistical models. Using methods known as artificial neural networks, a subset of machine learning is used to create artificial intelligence that works similarly to a human brain. These algorithms aid computers in discovering possibilities that humans would never be able to. Mathematical formulae play a crucial role in this process. If you want to use machine learning, you'll need to use a lot of these formulae. Deep reinforcement learning eliminates the need for many complicated formulae by merging reinforcement learning with deep learning. The requirement for deep reinforcement learning would be unnecessary if we could predict the result of every action. A computer programme that tells us how to make a certain choice in order to attain some goal may be developed under such circumstances. In the real world, however, we are unable to make precise predictions because to the large number of factors at play. Deep reinforcement ML may assist us with this.

DRL Background

This equation, titled after American scientist Richard Bellman, was devised in the mid-20th century and is essential for making the best possible decisions. As a key part of DRL algorithms, this equation tells users how much benefit they may anticipate in the long run based on their present condition if they choose the best action now and at each subsequent step in the future. This equation can't be used until you know what's going on around you and what the potential benefits are. Scientists historically employed the Bellman equation solely to handle more basic tasks, such as traversing a labyrinth, because of these criteria. Reinforcement algorithms, which aim to simulate human brains decision-making processes in a

computerised form, were not developed until decades later. Reinforcement learning's initial techniques were not very effective. Scientists at the time were unable to collect very complicated information.

In 2011, once deep learning debuted, DRL, which enables the system to learn from its errors via trial and error, became more practical. Ivakhnenko and Lapa, a Soviet mathematician and his collaborator, created modest but effective artificial neural networks in the mid-1960s. After John Hopfield hit the news in the early 1980s with his recurrent neural networks, it reignited interest in the field. His work, which was heavily influenced by neuroscience studies on memory and learning, laid the groundwork for all that followed later. Deep learning has had tremendous success, outperforming standard machine learning methods in a wide range of applications. Computer vision, voice control, translation software, art, medical data processing, robots and management, and cybercrime are just a few of the many fields in which these technologies are used. This newfound interest in ML may be traced back to three things. For the first time ever, scientists have unprecedented access to vast volumes of data. There will be twice as much information generated between 2025 and 2030 than there has been since the beginning of time, according to estimates.

Computer power has grown tremendously during the last several decades. Finally, algorithms' capacity to accomplish complicated jobs has significantly increased. Models of this kind may be used to tackle complicated issues in the fields of renewable energy and natural language interpretation, as well as other fields. Deep learning methods have made significant contributions to a variety of fields, including computer vision, computational linguistics, medical technology, and biomedical imaging, all of which rely on these techniques.

In the field of ML, reinforcement learning has always been a hotspot. To maximize the cumulative return in the present environment, reinforcement learning continuously enhances its theory by engaging with the real-world environment. A growing body of evidence suggests that deep learning is having an impact on a wide range of disciplines of study. As a result, there is a range of algorithms that may be used to learn via reinforcement learning using deep learning. Value-based reinforcement learning is a subset of the three forms of deep reinforcement learning that may be classified as previously mentioned.

While a neural network is used to calculate the value function, approaches such as the Deep Q Network (DQN) methodology utilize that value function to direct the agent in making policy decisions. Reinforcement learning approaches based on policy-based deep reinforcement teaching may parameterize policies and accomplish policy optimisation by learning variables, such as the Deterministic Policy Gradient (DPG) technique. It is simple to produce gradient extinction during policy update and subsequently fall into the local optimum solution issue using this sort of approach when dealing with high-dimensional continuous spatial issues. Actor-critic structure-based deep reinforcement learning approaches combine policy-based and value-based techniques to learn laws while fitting valued functionalities, such the Deep Deterministic Policy Gradient (DDPG) approach. A single step of the Time Difference (TD) approach is used to update the parameters of the critic network, which are used to train the actor system parameters. It's possible to combine the benefits of value-based techniques with the disadvantages of the policies gradient algorithm in actor-critic-based approaches; nevertheless, the policy update collapses into a local optimum solution owing to the absence of gradients.

DQN and DPG principles are used in the DDPG method to solve jobs with continuous action. By using an experienced playback pool, DDPG may be trained using past data and obtain superior performance in continuous action tasks, as an off-policy actor-critic technique. It was after this that, influenced by double DQN, two critical networks were used in tandem to suit the state implementation value function (TD3) based on the DDPG. The final estimate is based on the two specific network outputs' minimum values. When the DDPG median functionality is overestimated, TD3 corrects it and makes the agent more stable. When updating policy, both the policy-based methods, DDPG and TD3, utilise gradient information, which unquestionably suffer from the disappearing gradient issue when updating policy. Adding a little quantity of random variation to the neural network's policy output may reduce the impact of the gradient disappearing on policy updates. For instance, NoisyNets explicitly add randomized noises to the variables of the neural net in order to boost the exploration capabilities of the algorithm. Because random noise has a non-directional and unpredictable impact on policy, this method's effectiveness is restricted. The combination of gradient and ML policies may be used for complicated and demanding activities like game simulations, robot navigation, and conversation system. A fundamental issue persists when using policy gradient approaches in the context of continuous control, namely the local optimum problem induced by the removal of gradient during the updating process. Policy learning may be accomplished using the generation model, as suggested by Hinuma, Kohyama, and Tanaka [11]. To prevent a local optimum issue, the complexity of model training is raised.

The non-gradient optimisation technique of evolution policies has been used for years and works well in specific benchmark situations for reinforcement training. The evolution policy is quicker to construct, has less hyperparameters, does not need gradient knowledge, is easier to extend in a distributed context, and is less influenced by sparse incentives opposed to gradient optimization policies. When Gaudet et al. [12] presented Natural Evolution Policies (NEP) to optimize the policy, they used natural gradients to update it in a way that was more beneficial for their system. A non-gradient black box estimator based on the NEP was utilized by [13] to discover the best policy settings. By merging the population-based evolutionary algorithm with DDPG, Chen et al. [14] developed Evolutionary Reinforcement Learning (ERL). For reinforcement learning, Šljivić [15] merged the Cross-Entropy Method (CEM) with ERL and presented a CEM-RL technique, which increased the algorithm's efficiency.

The use of sophisticated learning techniques like RL and DRL in CC designs has increased in recent years. By making use of the network surroundings and its previous experiences, RL-enabled endpoints can figure out the best way to reduce congestion. These mechanisms are less reliant on pre-established norms and are more able to adapt to the constantly changing environment in which they operate. As opposed to RL, which is limited in terms of scalability, DRL utilizes the benefits of DNN to educate the training procedure and enhance performance. Depending on observed RTT, ACK, and other metrics, each endpoint modifies their transmitting rate or CW. Making a choice is difficult in this situation. Decision-making in unpredictable and stochastic contexts is often modelled as a Markov Decision Process (MDP). The aim is to design an MDP strategy that maximizes the predicted cumulative payoff. Network state (e.g., RTT and CW) is used to alter the DRL agent's activities (e.g., the communication rate or CW) in response to observed network conditions (e.g., RTT, CW). The reward degree (e.g., latency and output) might be improved by training a DNN to map state to action.

There have been significant developments in CC based on DRLs (see Fig. 5). This is the first time that CC methods have been developed using RL. CC policies may be learned on-the-fly without the need for previous knowledge of the network structure. Aurora is a DRL-based approach that use a densely integrated DNN to learn state-action combinations from previously recorded historical information. There is a surprising amount of versatility to this strategy. CC settings for short and long TCP traffic may be dynamically configured using RL as the environment changes. Real-world data is used in the assessment of how well the application performs.

The MPTCP CC issue is examined in [16]. CC for all operating flows is dynamically and cooperatively optimized by a single agent in [17]. To better represent network dynamics, a long short-term memory (LSTM) paradigm is included into the DRL structure in [18]. For the first moment, DRL integrates the actor-critic approach for constant CC.

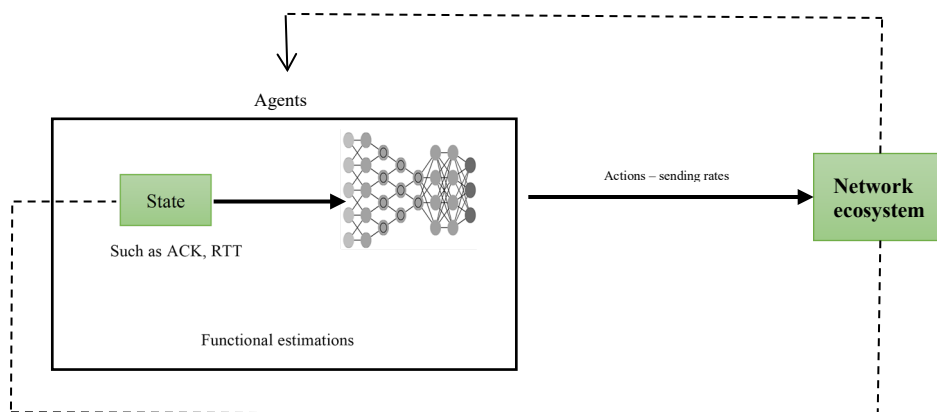


Fig 5. Model of Deep Reinforcement Learning (DRL) for congestion control

MPTCP's route diversity-induced decrease of Controllability is the topic of [19]. An RL-based architecture proposed by the researchers in [20] may be used to decouple model development and implementation. The CC framework is proposed offline and deployed in authentic for online display alteration decision-making in instantaneously. The initial CW (IW) choice issue in a mmWave network is addressed in [21]. Flow Completion Time (FCT) optimization may be achieved using a DRL-based online decision-making strategy.

Performance Enhancements

When it comes to traffic management, there are several variables that might influence the outcome. Predicting congestion-related characteristics may be accomplished with the use of supervised and unsupervised ML approaches. Poor throughput may be caused by a lack of capacity to discriminate between congestion and poor channel clarity in wireless networks. To enhance CC in next wave cellular networks when wireless connections are more readily obstructed, it is critical to precisely determine the source of packet loss. To determine the cause of transmission errors at the edge networks, several ML approaches were employed. Congestion may lead to packet loss in optical networks, as shown in [22]. TCP performance may be improved by using classifications. For wireline networks, a loss predictor-based CC strategy based on supervised learning is proposed in [23]. The balance between throughput and latency is better than with NewReno. Classifiers that use ML tend to be more accurate than those that don't. Variants of TCP that use ML to improve their classification accuracy have outperformed the usual rule-based alternatives.

Congestion Prediction

Constraint prediction plays a critical role when implementing dynamic routing and other network management techniques. In fact, when congestion occurs, it may be too late to take further steps to improve throughput performance. The sender may proactively react to congestion if congestion or congestion-related characteristics like TCP throughput and RTT can be reliably anticipated. There are now two basic approaches to estimating congestion-related factors, i.e., formula-based and history-based, which are currently being studied. Sender metrics, like RTT, packets loss rate, and CW, are integrated into a formula that generates predictions for the receiver. However, in today's extremely dynamic and complicated

networks, timely collection of such data is not simple at all. Furthermore, because of TCP's dynamic nature, it is difficult to establish a formula-based model that is correct. For example, TCP uses the EWMA method for RTT estimation, a time series assessment method that relies on historical data. The use of time series analysis may, on occasion, provide erroneous results.

Conventional methods of congestion prediction have been shown to be inadequate in the face of the challenges posed by ML-based techniques (mostly supervised and unsupervised learning). Gusti, I. Irhamah, and H. Kuswanto [24] uses Support Vector Regression (SVR) to forecast TCP throughput by using multivariate regression. Using supervised learning techniques such as random forest, Nascimento et al. [25] developed a loss prediction model to estimate the likelihood of packet loss due to congestion. Predicting and reducing packet loss incidents, reducing the frequency of transmitting rate reductions, and increasing throughput are all benefits of this technology. Congestion-related parameters may be predicted from passive data using ML techniques, which are promising for variable forecasting.

ML-based CC Workflow

RL, which may not need network information, will be the dominating model for the implementation of ML-based CC, since actual network data tends to be difficult to gather and categorize. As a result, it has the ability to let endpoints swiftly adapt to changes in their environment and to identify the optimum solution via trial and error. Both supervised and unsupervised learning methods may benefit from the use of prior performance data. DRL and reinforcement learning may be combined in [26]]. DRL is used for online decision making in the system. Data generated during online learning may be used to enhance performance by using a supervised learning strategy.

As seen in Fig. 6, CC may be implemented using ML. Decision-making issues are at the heart of the problem. The appropriate condition, action, and reward should be determined based on the detected signals and the objective. To assist the model to understand which control policy works best in its environment, a variety of training approaches may be used throughout the model's development phase. Using supervised and unsupervised learning methods, it is possible to extract useful characteristics from the training data. Finally, the model may be used in a real-world setting.

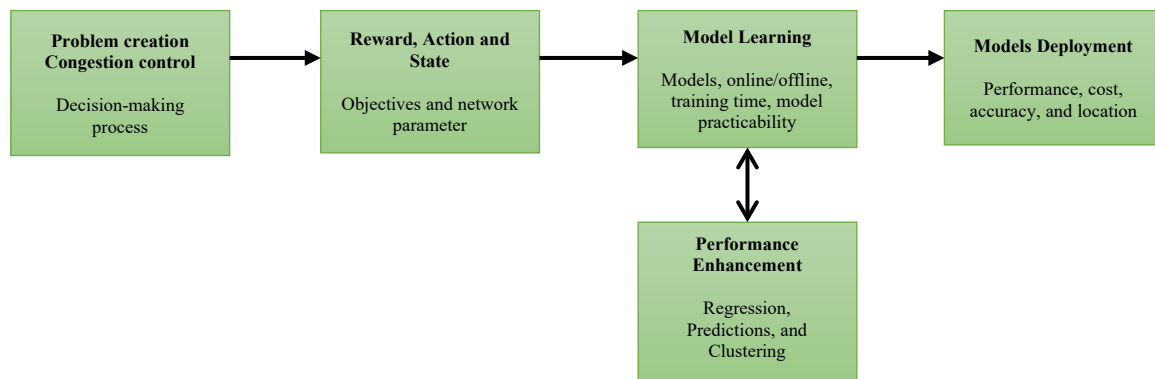


Fig 6. Workflow of ML-based CC.

VI. CHALLENGES AND OUTLOOK

End-to-end traffic control may be achieved with ML. However, the vast majority of ML algorithms have been developed to operate well in networks with a single node. ML-based CC strategies have not been completely deployed and thoroughly evaluated in the actual world. Here, we discuss some possible future study avenues.

Real Data Gathering

There has to be a considerable volume of high-quality network and data profiles obtained. Virtual datasets are often used in current research since they can be easily constructed using a particular network form. Thus, the subsequent ML prototype might be incompatible with a real-world setup. Labeled information may be time-consuming and expensive to obtain for supervised learning. A single open-source dataset and a shared infrastructure for scholars to review their results could save a deal of time and resources. Systems with a lot of power and authenticity are in great demand since testing large networks is costly and challenging to obtain.

Fairness, Generalization and Robustness

In certain cases, the use of CC based on ML may be unfair to current techniques. ML CC could be trained to occasionally cause packet loss in order to encourage other TCP protocols to backed off so that ML-centered CC may use more network capacity. In order to get along with other proto-cols, it's tough to find out how to do so. Because of the specific requirements of network services, most communication networks necessitate worst-case performance guarantee. ML-based CC must be able to withstand the frequent changes in the surrounding environment. ' As a final consideration, it is essential that the ML

algorithm be capable of adapting to dynamic network conditions without having to retrain every moment the network conditions change.

Cross-layer Optimization

The reward function is often used as an optimization target in ML. All typical incentive methods aim to increase throughput, minimise delay, decrease loss, and assure fairness. Users' Quality of Experience (QoE) may be included into the award design process since it is up to the designer to decide on the reward function. For instance, it is predicted that video traffic will account for more than 80% of all Internet traffic by 2020, with most of that data being carried via TCP. Users' QoE may greatly benefit from CC methods. Research into cross-layer optimization is a potential avenue.

Implementation in the Real World

Virtual environments are used to evaluate the vast majority of DRL-based CC algorithms. Real-world testing has not been done on them. System services like rule tables and ML models need to be included into operating system kernels for future implementations. CC techniques may be evaluated under dynamic network settings using a prototype that has been constructed.

VII. CONCLUSION

Machine Learning (ML) methods must be used to create efficient Congestion Control (CC) algorithms in today's increasingly complicated network. ML-based smart CC methods have the potential to significantly improve network performance despite recent gains. CC based on ML still has a long path to go before it can be put into effect for real-world problems. A selection of recent improvements in ML-based end-to-end CC was presented in this publication. Additionally, we spoke about unsolved issues that need additional networking and ML research. CC has emerged as a critical consideration in communications system architecture in recent decades, since network capacity and network operations have grown at such a fast rate. Congestion arises when the purchasing power for a resource surpasses the maximum capability of the resources, which may have a negative impact on network efficiency and dependability. Long data delivery delays, wastage of resources as a result of missing or lost packets, and even a hypothetical congestion collapse, when all network connections halt, are all consequences of severe congestion. No guarantee of delivery or timeliness is provided by Transmission Control Protocol (TCP), which can only give best effort service. As a result, the information source is unable to discern the network's current state. An increase in the queue size due to increased traffic will ultimately lead to congestion. The router's queue management strategy uses a queue to level out spikes in incoming packet rates. When a packet comes and the queue is full, it will be discarded according to the Drop Tail (DT) policy, which is the most comprehensive policy for packet dropping.

References

- [1] M. Molinaro and G. Orzes, "From forest to finished products: The contribution of Industry 4.0 technologies to the wood sector," *Comput. Ind.*, vol. 138, no. 103637, p. 103637, 2022.
- [2] B. Xiang, X. Liu, and Y. Chen, "Event-based networked predictive control systems with secure transmission protocol," *Int. J. Control Autom. Syst.*, vol. 20, no. 4, pp. 1076–1086, 2022.
- [3] L. Khoshnevisan, F. R. Salmasi, and X. Liu, "Integral sliding-mode robust observer-based congestion control for wireless access networks," *J. Contr. Decis.*, vol. 9, no. 2, pp. 152–164, 2022.
- [4] W. Pan, H. Tan, X. Li, and X. Li, "Improved RTT fairness of BBR congestion control algorithm based on adaptive congestion window," *Electronics (Basel)*, vol. 10, no. 5, p. 615, 2021.
- [5] E. Ancillotti and R. Bruno, "BDP-CoAP: Leveraging bandwidth-delay product for congestion control in CoAP," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019.
- [6] M.-R. Fida, A. F. Ocampo, and A. Elmokashfi, "Measuring and localising congestion in mobile broadband networks," *IEEE trans. netw. serv. manag.*, vol. 19, no. 1, pp. 366–380, 2022.
- [7] R. Node, "Relay node placement in hierarchical wireless sensor networks," *J. Adv. Comput. Netw.*, pp. 41–46, 2017.
- [8] G. K. Facenda et al., "Adaptive relaying for streaming erasure codes in a three node relay network," *arXiv [cs.IT]*, 2022.
- [9] S. Suma and B. Harsoor, "Detection of malicious activity for mobile nodes to avoid congestion in Wireless Sensor Network," in *2022 IEEE Fourth International Conference on Advances in Electronics, Computers and Communications (ICAEECC)*, 2022.
- [10] P. M. B. Melo, "An enhanced transmission control protocol initial sequence number steganographic method," *J. Adv. Res. Dyn. Control Syst.*, vol. 12, no. 01-Special, pp. 252–259, 2020.
- [11] Y. Hinuma, M. Kohyama, and S. Tanaka, "Boundary plane-oriented grain boundary model generation," *Model. Simul. Mat. Sci. Eng.*, vol. 30, no. 4, p. 045005, 2022.
- [12] G. Gaudet, "Forces underlying the evolution of natural resource policies in Quebec," in *Natural Resources in U.S.-Canadian Relations*, Routledge, 2019, pp. 247–265.
- [13] "The continuing evolution of immigration and naturalization issues and policies (Asians)," in *From All Points*, Indiana University Press, 2018, pp. 124–136.
- [14] S. Y.-C. Chen, C.-M. Huang, C.-W. Hsing, H.-S. Goan, and Y.-J. Kao, "Variational quantum reinforcement learning via evolutionary optimization," *Mach. Learn.: Sci. Technol.*, vol. 3, no. 1, p. 015025, 2022.
- [15] N. M. Šljivić, "Cross-entropy method for estimation of posterior expectation in Bayesian VAR models," *Commun. Stat. Theory Methods*, vol. 46, no. 23, pp. 11933–11947, 2017.
- [16] Q. Liu, L. Long, Q. Yang, H. Peng, J. Wang, and X. Luo, "LSTM-SNP: A long short-term memory model inspired from spiking neural P systems," *Knowl. Based Syst.*, vol. 235, no. 107656, p. 107656, 2022.
- [17] X. Long, J. Wang, S. Gong, G. Li, and H. Ju, "Reference evapotranspiration estimation using long short-term memory network and wavelet-coupled long short-term memory network," *Irrig. Drain.*, 2022.

- [18] Z. Wen, B. Long, K. Lin, and S. Wang, “Equivalent modeling based on long short-term memory neural network for virtual synchronous generator,” in 2021 33rd Chinese Control and Decision Conference (CCDC), 2021.
- [19] K. Fahmi, D. Leith, S. Kucera, and H. Claussen, “Understanding MPTCP in multi-WAN routers: Measurements and system design,” in 2021 IEEE 46th Conference on Local Computer Networks (LCN), 2021.
- [20] P. Ladosz et al., “Deep reinforcement learning with modulated Hebbian plus Q-network architecture,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. PP, pp. 1–12, 2021.
- [21] B. Zhang, F. Devoti, I. Filippini, and D. De Donno, “Resource allocation in mmWave 5G IAB networks: A reinforcement learning approach based on column generation,” *Comput. netw.*, vol. 196, no. 108248, p. 108248, 2021.
- [22] M. Asghari, J. Bagherzadeh, and S. Yousefi, “Loss estimation and control mechanism in bufferless optical packet-switched networks based on multilayer perceptron,” *Photonic netw. commun.*, vol. 35, no. 2, pp. 274–286, 2018.
- [23] S. Mohapatra, J. An, and R. Gómez-Bombarelli, “Chemistry-informed macromolecule graph representation for similarity computation, unsupervised and supervised learning,” *Mach. Learn.: Sci. Technol.*, vol. 3, no. 1, p. 015028, 2022.
- [24] K. H. Gusti, I. Irhamah, and H. Kuswanto, “Hybrid double seasonal ARIMA and support vector regression in short-term electricity load forecasting,” *Aptech Proceeding Int. Semin. Appl. Technol. Sci. Arts: Dev. Green Agro-Ind. Support Hum. Life Sustain.*, vol. 0, no. 6, p. 322, 2021.
- [25] C. A. Nascimento, A. M. Luiz, P. L. Barros, A. T. P. Neto, and J. J. N. Alves, “A CFD-based empirical model for hazardous area extent prediction including wind effects,” *J. Loss Prev. Process Ind.*, vol. 71, no. 104497, p. 104497, 2021.
- [26] M. Stephan, L. Servadei, J. Arjona-Medina, A. Santra, R. Wille, and G. Fischer, “Scene-adaptive radar tracking with deep reinforcement learning,” *Machine Learning with Applications*, vol. 8, no. 100284, p. 100284, 2022.