USING AI/ML TECHNIQUES IN TCP

Prepared for

Doç. Dr. Tolga Girici

TOBB University of Economics and Technology

Prepared by

Ayşe Nur Sadioğlu

191101077

April 2, 2023

Using AI/ML Techniques in TCP

Section I

Introduction to the Study

Introduction

The Transmission Control Protocol (TCP) is fundamental in the Internet protocol architecture. Due to practical reasons, a significant number of web applications, use the reliable end-to-end Transmission Control Protocol (TCP) as a transport protocol. TCP, a transport layer protocol that is the backbone of Internet connectivity, provides the reliable transmission of data over a network. It serves up to 85% of all Internet traffic worldwide. However, with the increasing demand for high-speed and low-latency network communication, the traditional TCP protocol has shown several limitations and weaknesses, such as inefficient congestion control, inability to adapt to dynamic network conditions, inefficient data transfer in high-speed networks, latency, and security vulnerabilities.

The performance of TCP protocols can be improved by applying machine learning (ML) and artificial intelligence (AI) techniques. By allowing TCP to adapt to changing network conditions, optimize its operations, and improve its security, AI/ML approaches can be applied to solve the flaws and limits of TCP.

Statement of the Problem

The purpose of this study is to discuss the flaws and restrictions of the TCP protocol currently in use, to address and explain academic studies about the AI/ML techniques used in TCP, to discuss the reason behind each technique's choice, and provide some brand-new ideas and directions for future research.

Limitations of currently deployed TCP protocols

The reliability and efficiency of a network can be affected by a variety of limitations and weaknesses in the TCP protocols currently in use. One of these limitations is the Slow Handshake. After a handshake is established, the TCP will perform a data transfer between the sender and the receiver. This causes slow connection establishment. Limited bandwidth utilization and sensitivity to DoS (Denial of Service) attacks in high-speed networks are other limitations of TCP. Also, Its nature is generic. This means it can only fairly reflect the TCP/IP family of protocols as a result. It is unable to define a Bluetooth connection, for instance.

Wide-area networks (WANs) were the initial scenario for the development and implementation of TCP. Although TCP is ideally suited for these kinds of networks, it might not be optimal for small networks like LANs.

Compared to WANs, LANs mostly have simpler networks with more predictable topologies, reduced latency, and lower packet loss rates. As a result, for these kinds of networks, the congestion control algorithms and other TCP characteristics could not be required or potentially be harmful.

TCP connections are vulnerable to spoofing attacks, which take place when an attacker pretends to be one of the endpoints or manipulates an active connection. Data can be stolen, altered, or deleted through spoofing, and other network attacks can also be carried out.

As mentioned in the paper "Security Problems in the TCP/IP Protocol Suite" by S.M Bellovin [2]. Morris described one interesting security vulnerability by creating a TCP packet sequence without ever receiving any responses from the server using TCP sequence number prediction. This allowed him to spoof a trusted host on a local network. [11]

Normal TCP connection includes a 3-way handshake. First, the client chooses an initial sequence number and sends it to the server. Then the server acknowledges the initial sequence number from the client and sends its own initial sequence number to the client.

If we take ISNC as the initial sequence number of the client and ISNS as the initial sequence number of the server we can express the exchange as follows:

C→S:SYN(ISNC)
S→C:SYN(ISNS) ,ACK(ISNC)
C→S:ACK(ISNS)
C→S:data
and/or
S→C:data [2]

For data transmission between the client and server, the client must first receive ISNS (Initial sequence number of Server).

Suppose X, an outsider, somehow guesses ISNS. In that situation, it may send the sequence outlined below to pretend to be a trusted host T:

$$X \rightarrow S: SYN(ISNX)\ ,SRC = T$$
$$S \rightarrow T: SYN(ISNS)\ ,ACK\ (ISNX)$$
$$X \rightarrow S: ACK(ISNS)\ ,SRC = T$$
$$X \rightarrow S: ACK(ISNS)\ ,SRC = T, nasty - data$$

Hence, even if the S→T communication does not reach X, X still has access to the message's contents.

Section II

Recent academic studies using AI/ML techniques

Transmission Control Protocol (TCP) is a reliable transport layer protocol, that ensures process-to-process communication. TCP performance optimization for wireless networks had typically been a difficult task for researchers. Several TCP variants have been proposed by researchers over the years to help TCP keep functioning well in wireless networks. Since its massive benefits, ML-Machine Learning techniques have started to be used in every conceivable field during the past ten years. To enhance TCP's performance for wired and wireless networks, numerous researchers have attempted to apply a variety of ML approaches. In this part, the foundations of ML approaches are covered along with how they can be used with TCP. [1]

TCP was initially designed for wired networks, where network congestion is the main reason for packet losses. TCP prioritizes congestion control and treats every packet loss as a potential source of congestion. Performance decreased noticeably when the same TCP was utilized with wireless networks, it was determined. The cause was TCP's inability to deal with different wireless network non-congestion problems. The congestion control-centric architecture of TCP required modifications to address the problems with wireless networks.

Through the past few years, ML has also been integrated into computer networks. For traffic analysis, network security, and protocol optimization, the idea of an intelligent network protocol or application design based on machine learning is being used.

Machine Learning (ML) refers to a set of statistical techniques that enable computers to learn from data without being explicitly programmed. The aim of ML is to develop a decision model that can improve its accuracy and effectiveness over time through experimentation. By exploring large datasets, ML algorithms identify underlying patterns that are used to make predictions about new, unseen data. This process of building decision models based on data is known as "training." Once a model is trained, it can be used to classify or predict the outcome of new data based on the patterns it has learned. Machine learning has become increasingly popular due to its flexibility, accuracy, and efficiency. One of the main advantages of machine learning is the ability to adapt and learn from new data, which allows it to make more accurate predictions over time. Two popular types of machine learning are supervised learning and reinforcement learning. Supervised learning involves processing training data to build a model, which is then used to make decisions about future data. On the other hand, reinforcement learning learns directly from interactions with the environment, without any prior model-building phase. This type of learning is particularly useful in scenarios where the optimal decision-making strategy is not known in advance, and the system must learn and adapt through trial and error.

Based on the paper “TCP with Machine Learning - Advances and Opportunities” by Hardik K. Molia and Dr. Amit D. Kothari, this subsection describes how ML techniques can be used to improve TCP performance in 2 parts.

We shall discuss the two different categories of learning which are offline and online learning. Online learning is an approach that receives data one observation at a time, whereas offline learning is an approach that ingests all the data at once to develop a model.

The variations that will be discussed at this point will attempt to overcome the following limitations:

1. The inability to adjust to new or unexpected network circumstances.
2. Failing to take lessons from past events.

1. TCP with Online Learning

1.1 Q-Learning TCP

If we consider Q-Learning TCP, Q-learning TCP is a variant of the TCP protocol that addresses the issue of unfairness in network traffic distribution. Q-learning is a reinforcement learning algorithm in which an agent learns through interaction with the environment. It is a cross-layer, and every TCP sender presents the network as an MDP-Markov Decision Process. TCP sender performs as the agent in this method, and the network performs as the environment.

Learning-based and data-driven TCP is a variant of the TCP protocol that uses Q-learning, a type of reinforcement learning, to determine the congestion window (*cwnd*) size.
The state space has four variables, each of which is discretized into 10 intervals. There are five possible actions to define changes in *cwnd* size in terms of the number of bytes.

A. States

1. Moving average of inter-arrival times between new ACKs.
2. Moving average of inter-arrival times between sent packets.
3. Ratio between current RTT and best RTT so far.
4. Slowstart Treshold.

B. Actions

1. Reduce by -1.
2. No change.
3. Increase by 5.
4. Increase by 10.
5. Increase by 15 [1].

To reduce the memory requirement for storing the Q-table in Q-learning-based TCP variants, which use continuous variables, they are discretized. Furthermore, function approximation techniques such as CMAC and Fuzzy are used to store an approximation of the Q-table, which requires less memory without affecting performance.

1.2 Neural Network-Based Reliable Transport Layer Protocol

A group of mobile devices or nodes create an adhoc wireless network known as a MANET, which stands for Mobile Adhoc Network. These devices can connect with one to another without the need for existing infrastructure, such as a router or an access point.
A neural network-based reliable transport layer protocol for MANET [3] is proposed to recognize and capture mobility patterns of nodes to differentiate packet losses. [1]
This solution is designed to prevent unnecessary delays and *Cwnd* reduction.

Each node holds a *M_Win*- MobilityWindow to capture the mobility pattern. *M_Win* works like a left shift register and is updated every Hello interval. If the neighbor table has the node's content, 1 is inserted; otherwise, 0 is inserted. Because a node with high mobility has more 1s in its *M_Win*, the *W_Win* - Average Weighted Mean of *M _Win* is closer to 1. W_Win of node n_i at time *t* is calculated as, [4]

$$W\_Win_{(n_i,t)} = \alpha * M\_Win_{(n_i)} + (1-\alpha) * W\_Win_{(n_i,t-1)}$$

Neural Network Design

To determine the *ECwnd* or Estimated Congestion Window, a neural network that implements biased and reinforcement learning and is built with a single layer and feed forward structure has been developed. When deciding whether to increase, decrease, or maintain the current size of the congestion window (*Cwnd*), one factor is the *ECwnd*. The output of the neural network, which is trained to make predictions based on the network's current state, defines the proper course of action in relation to the *Cwnd*.

2. TCP with Offline Learning

In this part, TCP variations with offline learning are covered.

2.1 TCP Ex Machina

TCP ex Machina [5] is a research paper that proposes a new approach to congestion control in computer networks. This approach involves the use of a program called Remy, which uses machine learning techniques to generate congestion control algorithms.

Remy, the proposed approach, is intended to find congestion control rules in a computer network based on previous hypotheses, traffic models, and objectives. For instance, the purpose can be to obtain high throughput and a short queuing delay. Using these models as inputs, Remy creates the Remy-CC congestion control algorithm for TCP senders, which maximizes the overall expected value of the objective function. In other words, Remy develops a congestion control algorithm that is precisely suited to the network circumstances and objectives at hand using machine learning techniques.

Machine Learning to Optimize TCP Communications Over High-Frequency Communications Links

In this section, using Machine Learning (ML) methods to predict two important parameters, Forced Retransmission Timeout (fRTO) and Mean Segment Size (MSS), in order to enhance the process of calculating fRTO for all communications instances proposed. By using ML techniques to dynamically predict these parameters, the fRTO calculation process can be improved.

With a prediction accuracy of 82%, the researchers found that Decision Tree Regression was the most accurate Machine Learning technique. A particular kind of machine learning method called Decision Tree Regression uses a tree-like model of decisions and possible consequences to produce predictions. In this research, it was used to predict the values of Forced Retransmission Timeout (fRTO) and Mean Segment Size (MSS) parameters based on input data. [6]

HF-TCP is an enhanced version of the TCP protocol that is fully compatible with the standard TCP [7]. It offers the ability to control the Mean Segment Size (MSS) and the Retransmission Timeout (RTO) timer. While the RTO in standard TCP is calculated dynamically using algorithms, the proposed HF-TCP implementation predicts the fRTO (composed of F1 and F2) and the MSS through a process that involves two models trained on simulated data. To achieve this, the authors developed Machine Learning models using Scikit-Learn libraries and Python. These ML models are used to predict the fRTO parameters to enable faster data transfer with HF-TCP [8].

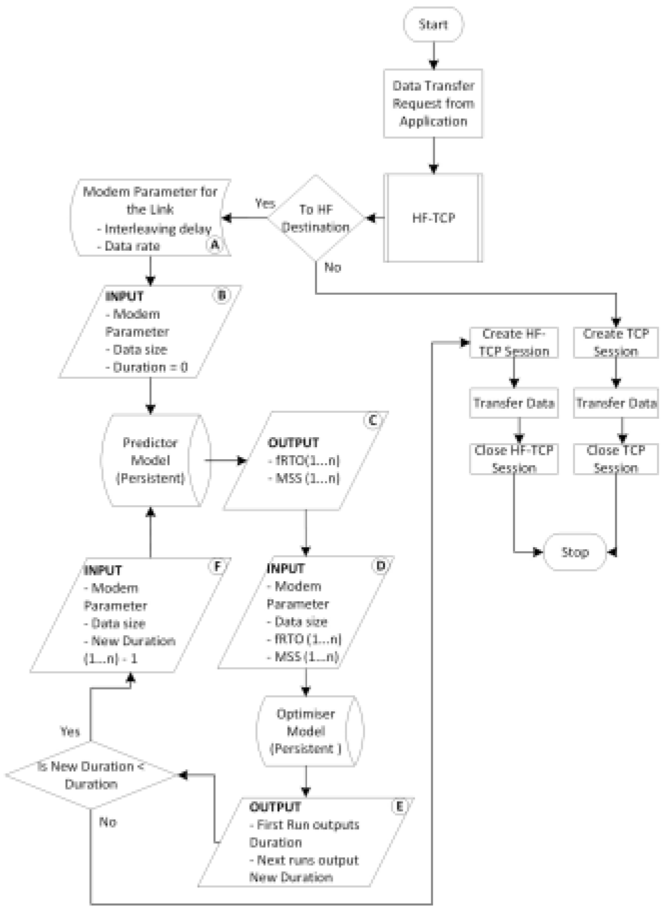


Figure 1. HF-TCP With ML Model Flowchart [6]

Machine Learning-Based TCP Security Action Prediction

As internet technology continues to grow, the frequency of network security incidents and cyber-attacks is also increasing. This has led to a growing need for computer network security. One way to achieve this is through the use of a TCP firewall, which allows or denies data transmission based on specific rules to provide security for the network. In traditional firewalls, network administrators manually set security rules, and often have to choose which packets to allow or deny in order to maintain network security. Due to the huge amount of data on the Internet, data administrator's job is getting harder.

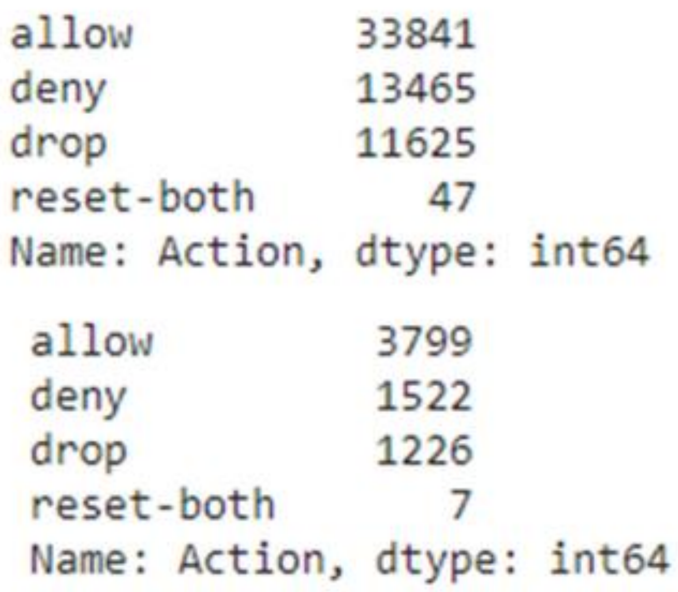
Given the growing importance of computer network security, it is crucial to use advanced techniques to address this issue. This research focuses on using machine learning methods to predict TCP security actions. To accomplish this, the researchers used the TCP transmission characteristics dataset provided by the UCI machine learning repository, and implemented various machine learning models such as neural networks, Support Vector Machines (SVM), AdaBoost, and logistic regression. By utilizing the idea of ensemble-learning, the final result has an accuracy score of over 98%. [9]

Method

The problem at hand is a multi-class classification problem, which means we have various options for machine learning algorithms to choose from. However, it is important to note that each algorithm has its strengths and weaknesses. Therefore, a sensible approach would be to implement multiple machine learning models to address the problem.

Training Set & Testing Set

The distribution of different classes in training set and testing set in shown below *Figure* 2.


```
allow          33841
deny           13465
drop           11625
reset-both        47
Name: Action, dtype: int64
```

```
allow          3799
deny           1522
drop           1226
reset-both        7
Name: Action, dtype: int64
```

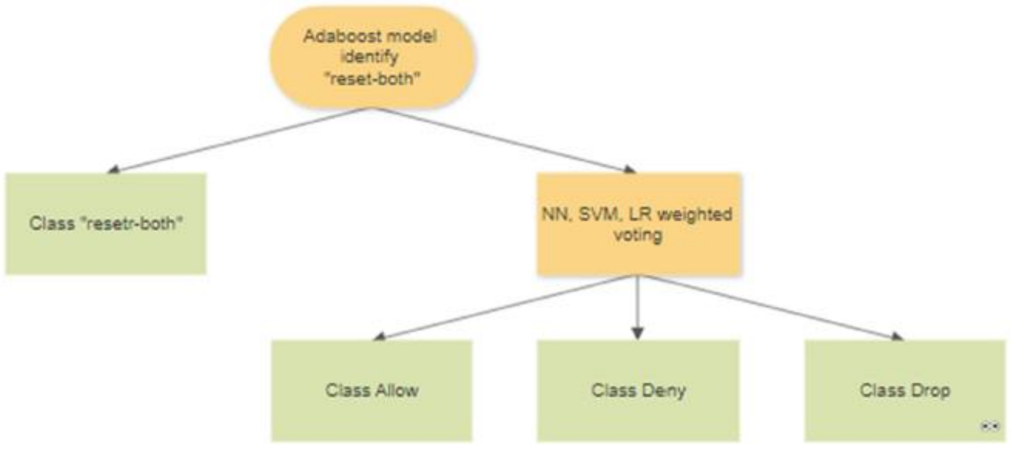
Figure 2. Class Distribution in training set and testing set [9]

Anomaly Detection

If we use multiple machine learning models together (an ensemble), some models may be affected by unusual data points. To make each model better, we need to find and remove those unusual points.

1) Isolation Forest

The chosen algorithm for detecting anomalies is Isolation Forest. This unsupervised algorithm calculates an anomaly score using a binary search tree (BST) to identify outlier data points [6]. It's particularly useful for processing large datasets quickly, which makes it a good fit for this task. However, since the anomaly detection step is only applied to the training set, it's only used in the machine learning process for the training phase.



Adaboost model identify "reset-both"
Class "resetr-both"
NN, SVM, LR weighted voting
Class Allow
Class Deny
Class Drop

Figure 3. Ensemble Flow Chart [9]

TCP Congestion Avoidance in Data Centres using Reinforcement Learning

TCP natively has mechanisms to handle congestion, but it can only control the traffic it sends and not the network conditions that cause congestion. The Transport layer cannot control the network conditions that lead to congestion. One of the methods used to handle congestion is called slow start, which sends data slowly to avoid overwhelming the network. However, this method does not utilize the full capacity of the network. [10]

TCP Congestion Avoidance / Reinforcement Learning

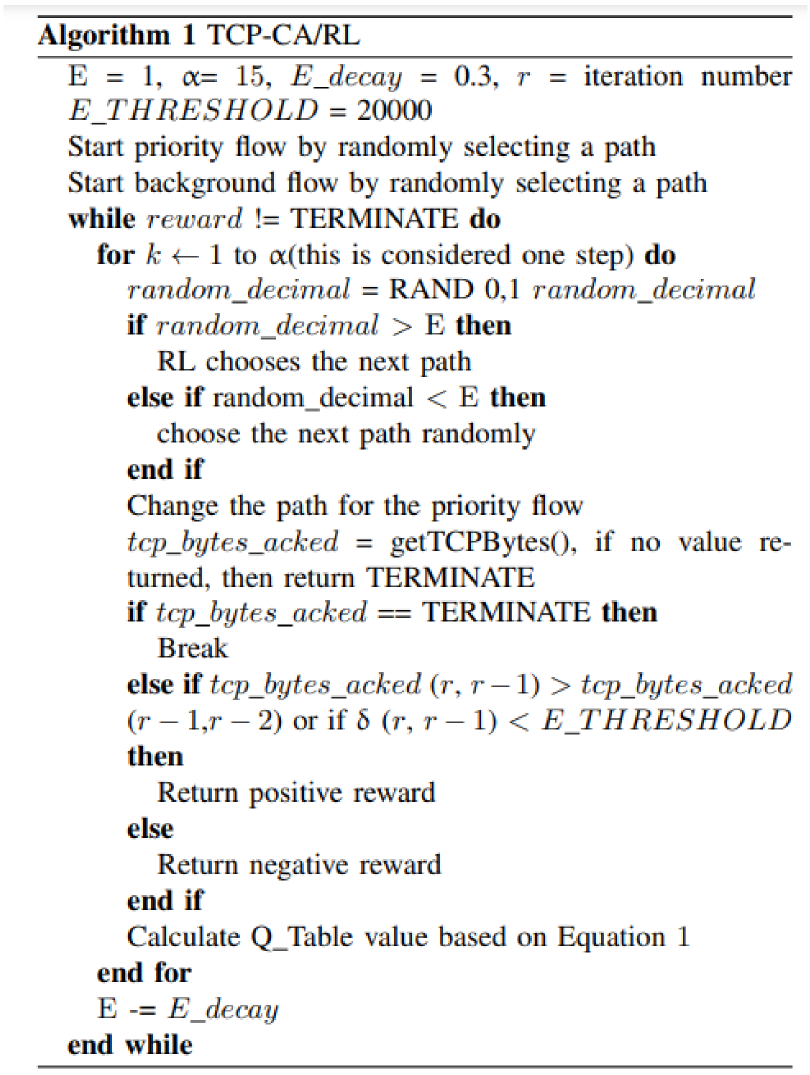
Reinforcement Learning (RL) is a machine learning technique that uses a reward-based system to learn from a continuously changing environment. In RL, an algorithm chooses an action based on a plan or policy that leads to a new scenario or state, and a reward is given based on the quality of the move. This reward is stored in a Q-table, which acts as a cheat-sheet of all possible options and the greatest Q-value for each state. [10] RL uses a Q-learning formula with parameters such as η and γ, which determine how much of the old value is retained and the importance given to future rewards, respectively. The main Q-learning formula for the RL algorithm is given in equation below.

$$\text{NewQ(state,action)}=(1 - \eta) * \text{Q(s,a)} + \eta * \text{ reward } + \gamma * \max Q'(s', a')$$

Datacenter traffic differs significantly from WAN traffic, and thus the methods used to prevent congestion cannot be applied in the same way. Datacenter networks have high bandwidth and low latency, whereas WAN traffic has low bandwidth and high latency. TCP operates at a higher throughput in datacenter networks, allowing for faster transmission of data, but also leading to faster congestion detection and a quicker decrease in throughput compared to WAN environments.

In a software-defined network, network functions can be controlled centrally from off the data path, enabling the deployment of cross-layer software modules with global network information to detect and reroute network traffic based on TCP parameters using techniques such as AI. The nature of network traffic makes it challenging to obtain realistic and precise datasets that can be effectively used for

training AI in all scenarios. This is because the AI may learn only from the training data's specific details, which may not translate well into other situations, particularly in real production networks.
Reinforcement Learning is a suitable AI technique for network control and management because of its operational characteristics and ability to learn from trial and error through continuous interactions with the environment. Reinforcement Learning is particularly useful for adapting to a changing network environment.[10]



```
Algorithm 1 TCP-CA/RL
  E = 1, α= 15, $E\_decay$ = 0.3, $r$ = iteration number
  $E\_THRESHOLD$ = 20000
  Start priority flow by randomly selecting a path
  Start background flow by randomly selecting a path
  while $reward$ != TERMINATE do
    for $k \leftarrow 1$ to α(this is considered one step) do
      $random\_decimal$ = RAND 0,1 $random\_decimal$
      if $random\_decimal > $ E then
        RL chooses the next path
      else if random_decimal $<$ E then
        choose the next path randomly
      end if
      Change the path for the priority flow
      $tcp\_bytes\_acked$ = getTCPBytes(), if no value re-
      turned, then return TERMINATE
      if $tcp\_bytes\_acked$ == TERMINATE then
        Break
      else if $tcp\_bytes\_acked$ $(r, r-1) > tcp\_bytes\_acked$
      $(r-1, r-2)$ or if δ $(r, r-1) < E\_THRESHOLD$
      then
        Return positive reward
      else
        Return negative reward
      end if
      Calculate Q_Table value based on Equation 1
    end for
    E -= $E\_decay$
  end while
```

Figure 4. The Algorithm [10]

Fig. 2. Reinforcement Learning - TCP-CA/RL

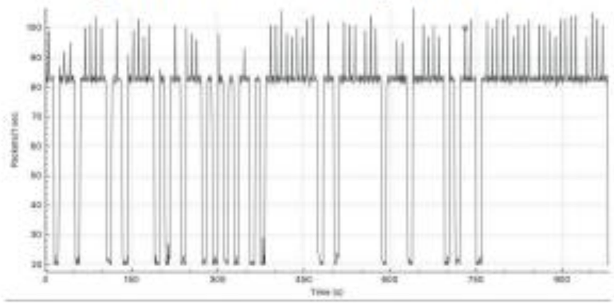


Fig. 3. Random Path Changes

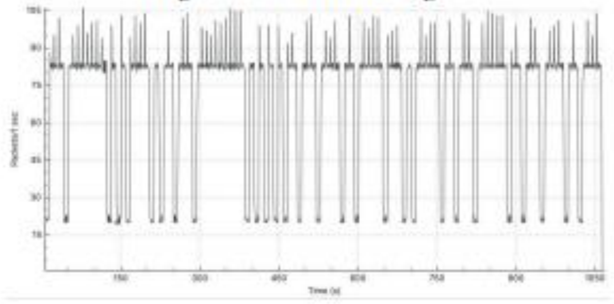


Fig. 4. TCP Congestion Control

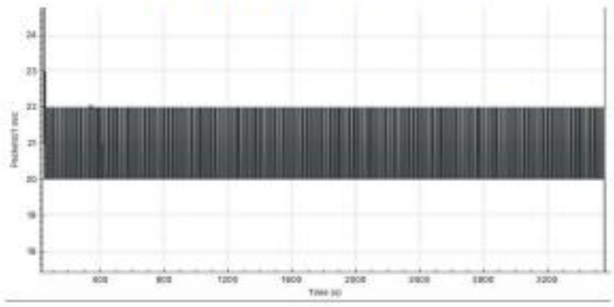


Figure 5. Reinforcement Learning (top), Random Path Changes (middle), TCP Congestion Control (bottom)

Section III

Novel Ideas and Directions for Future Research

Implementing ML-based network protocol design is still in the early stages, and most proposed solutions are not yet widely used in real-world networks. Most ML-based TCP variants focus on setting the transmission rate value (Cwnd) to address TCP's loss differentiation limitations. Additionally, ML-based ACK thinning schemes can help address intra-flow contention issues. However, the current challenge is selecting the most suitable ML algorithm for TCP. At the network protocol level, ML can be used to set various parameter values or even design a protocol itself. Further research is needed to develop a generalized and adaptable solution for dynamic networks through automatic protocol design. A systematic analysis of TCP performance with various ML algorithms can aid researchers in intelligent TCP design. [1]
ML-based network protocol is important for improved network performance, adatability to dynamic networks, reduced human intervention and future proofing.

Conclusion

In this report, using AI/ML techniques such as Reinforcement Learning, Q-learning, Decision Tree Regression and ensemble learning in TCP are mentioned. Firstly, limitations of TCP is given then AI/ML techniques explained. A literature search have been conducted and the studies explained thoroughly. Those researches used tools such as Python,Tensorflow,Scikit-learn and OpenAI Gym.

As a result of this research, theoretical knowledge about the following subjects was obtained:

- Limitations of currently deployed TCP protocols
- TCP with Online Learning & Offline Learning
- Machine Learning-Based TCP Security Action Prediction
- TCP Congestion Avoidance in Data Centres using Reinforcement Learning

In addition, research processes were learned by making literature research on these subjects.

References

[1] K. Molia, H. (2019). TCP with machine learning - advances and opportunities. International Journal of Advanced Trends in Computer Science and Engineering, 8(6), 3526–3534. https://doi.org/10.30534/ijatcse/2019/132862019

[2] Bellovin, S. M. (n.d.). A look back at "security problems in the TCP/IP protocol suite". 20th Annual Computer Security Applications Conference. https://doi.org/10.1109/csac.2004.3

[3] P. Kumar, S. Tripathi, and P. Pal. Neural network based reliable transport layer protocol for manet. 4th International Conference on Recent Advances in Information Technology IIT(ISM), Dhanbad-IEEE Explore, pages 1– 6, March 2018.

[4] Mo Dong, Qingxi Li, Doron Zarchy, Brighten Godfrey,and Michael Schapira. Pcc re-architecting congestion control for consistent high performance. Computing Research Repository Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation, abs/1409.7092, 2014.

[5] Keith Winstein and Hari Balakrishnan.Tcp ex machina: Computer-generated congestion control. Proceedings of the ACM SIGCOMM 2013 Conference, pages 123–134,2013.

[6] Dmello, A., Jayalath, D., Foo, E., &amp; Reid, J. (2022). Machine learning to optimize TCP Communications over high frequency communications links. IEEE Access, 10, 125526–125537. https://doi.org/10.1109/access.2022.3225401

[7] A. D. Mello, E. Foo, and J. Reid, ‘‘Characterizing TCP/IP for high frequency communication systems,’’ in Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS), Nov. 2018, pp. 1–7.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, ‘‘Scikit-learn: Machine learning in Python,’’ J. Mach. Learn. Res., vol. 12, pp. 2825–2830, Nov. 2011.

[9] Q. Zhao, J. Sun, H. Ren and G. Sun, "Machine-Learning Based TCP Security Action Prediction," 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Harbin, China, 2020, pp. 1329-1333, doi: 10.1109/ICMCCE51767.2020.00291.

[10] A. Hassan and S. S. Heydari, "TCP Congestion Avoidance in Data Centres using Reinforcement Learning," 2021 23rd International Conference on Advanced Communication Technology (ICACT), PyeongChang, Korea (South), 2021, pp. 306-311, doi: 10.23919/ICACT51234.2021.9370861.

[11] R. T. Morris. A weakness in the 4.2BSD unix TCP/IP software. Computing Science Technical Report 117, AT&T Bell Laboratories, Murray Hill, NJ, February 1985.