CHAPTER 1

INTRODUCTION

1.1 Background

In the modern era, the availability of good internet connectivity is of paramount importance [1]. As it enters into the digital era, the internet is becoming an indispensable tool for almost all activities, including buying and selling, company meetings, and the teaching and learning process. Furthermore, many applications and services rely on the quality of the network, such as video conferencing, media streaming, and online gaming [2, 3, 4]. A number of metrics can be employed to assess the quality of a network, whether it be deemed good or bad. One such metric is the Round-Trip Time (RTT) [5, 2, 6, 3]. Round-Trip Time (RTT) is the initial time interval for sending packets from sender to receiver until acknowledgement is received back by the sender, better known as the round-trip time of a package [4, 7]. Acknowledgement is a confirmation message that the data received has arrived and is intact [8, 9, 10, 2, 6]. RTT is one of the important metrics in measuring network performance that can be used to identify delays in data packets received in network communication [1, 5, 2]. In general, the performance of RTT is observed in terms of high and low RTT values. If the RTT value is high, it indicates that the network performance is below optimal levels and that network traffic is inefficient, resulting in high delays that affect the user experience [6, 3]. As a result, RTT is an important metric in many reliable transport protocols [8, 11]. Consequently, there is a demand to estimate RTT accurately. By estimating RTT, it is possible to determine the optimal Retransmission Time Out value in a network architecture. Retransmissions Timeout is a parameter used to determine the length of time that the sender must wait before considering that the data packet sent has been lost and must be resent. If the RTO value is too short, it will result in unnecessary retransmissions that affect network performance. Conversely, if the RTO value is too long, there will be a delay in detecting lost packets. In the absence of an estimated RTT, the quality of network services may be compromised, network performance may become less efficient, and a high enough delay may be introduced on the network.

Given the difficulties that arise when RTT is not estimated, numerous studies have been conducted on the estimation of RTT [6], with Chong et. al. [12] and Dong et. al. [9] developing neural network-based RTT estimation techniques. Dong et. al. achieved an accuracy value of 0.997. However, from these two studies, it can be seen that they used a peer-to-peer architecture, which is a relatively simple architecture. This allows the resulting RTT value to be smaller than if a more complex architecture were used. Jacobbson et. al. [8] managed to overcome the problem of overfitting and also keep the estimation smooth, but still able to capture sudden changes if there is a spike. However, the Kalman Filter parameters are set manually, which renders it less practical when used in congestion control. Jacobsson stated that the focus of his research is RTT estimation, not how network metrics can be used practically in congestion control algorithms. From the three aforementioned papers, it is evident that the overfitting problem persists, and that there is a need for practical algorithms to be developed for the estimation of RTT.

In light of the overfitting issue and the less practical algorithm employed in RTT estimation, it is deemed necessary to develop an improved RTT estimation model. Consequently, this research will concentrate on enhancing the RTT estimation model, with a model that can surmount overfitting and a practical algorithm in estimation.

1.2 Problem Identification

A neural network-based RTT estimation has been constructed by Chong et. al.[12] and Dong et. al.[9]. Chong et. al.[12] are the first researchers to estimate RTT using a neural network. This research has demonstrated that the neural network model has the capacity to predict RTT values. While Dong et. al.[9] employed an RNN with a Minimal Gated Unit. This research compared several baseline methods with RNN with MGU in predicting RTT. The R-square error accuracy value produced by this research was 0.997. However, both studies only used peer-to-peer architecture, which was the simplest and least complex architecture that allowd the resulting RTT value to be simpler than using a more complex architecture.

A filtering-based RTT estimation has developed by Guodong et. al.[13] and Jacobsson et. al.[8]. Guodong et. al.[13] employed adaptive filtering, which obviated the necessity to ascertain the precise statistical characteristics of the signal and noise. Furthermore, the algorithm was straightforward, rendering it conducive to implementation. Nevertheless, when there was a substantial alteration in RTT, Adaptive filtering was unable to track such changes because the estimate is based on the previous value, resulting in a similar estimate to the past. In contrast, Jacobsson et. al.[8] employed CUSUM Kalman filtering, which overcame the issue of overfitting and ensured the estimates remained smooth. However, this approach was still able to capture sudden changes in the event of a spike. However, the parameters of this Kalman Filter were set manually, which rendered it less practical when used in congestion control and did not align with the primary objective of this thesis, that is to develop an efficient algorithm for estimating RTT.

Dasgupta et. al.[14, 15] also built estimation functions using LSTM-RNN and LSTM-CNN to replace the Jacobbson estimation function in TCP.The RNN successfully reduced retransmission, although not significantly, by training the model for 10 epochs. CNN was quite significant in reducing retransmissions, but it took 100 epochs to create a model that was capable enough to estimate RTT. By training the model on multiple epochs, the computational time was so long that it took longer than the Jacobson algorithm.

Based on previous existing research, an algorithm is required that can accommodate outliers when estimating RTT, as well as an accessible and effective implementation of the algorithm. This thesis proposes the RELM method as a solution to address outliers and enhance the efficiency of the RELM by developing a regularization constant selection algorithm.

1.3 Objective and Hypotheses

The objective of this study is to address the issue of overfitting when estimating RTT and to develop a practical algorithm that can be used while maintaining a smooth estimation value. This thesis proposes the ELM method, given that ELM is known for its speed in training and its ability to handle problems with large features while providing satisfactory results [16, 17, 18, 19, 20, 21]. An important aspect of ELM is the random selection of input weights and the output weights of the Single Hidden Layer Feedforward Neural Network (SLFN) [22, 23, 24, 25, 21]. The performance of ELM in terms of speed is highly commendable. However, the method has several shortcomings, including issues with structure selection, overfitting and low generalist performance [26, 27, 28]. Furthermore, ELM does not consider heteroskedasticity in real applications, which can affect its performance when there are outliers in the dataset [16, 29]. Subsequently, the RELM method was proposed, which simultaneously minimizes training errors and least square weight norms by utilizing regularization constants [16, 29]. This enables RELM to effectively deal with outliers[29, 30]. RELM also retains the advantage of extremely fast training speed and the automatic selection of the number of hidden nodes, which results in its accuracy being almost the same as that of Standard ELM [29, 25]. The successful implementation of RELM addresses some of the limitations of ELM, while retaining the advantages of ELM itself. However, the selection of regularization constants in RELM is still based on trial and error [24, 31]. Consequently, this research will focus on the development of an efficient algorithm for the selection of regularization constants in RELM, to enhance the efficiency of RELM in RTT estimation.

The Regularization Extreme Learning Machine method employs data validation and regularization constants to guarantee that the model can learn from training data effectively, preventing overfitting and exhibiting the capacity to make accurate predictions on new data. The process of data validation helps to avoid overfitting and ensures that the model can generalize well on data that has never been seen before. The regularization constant then helps to prevent overfitting by preventing the model from becoming too complex and too tailored to the training data, thus improving the model's ability to generalize on new data. If the constant value is set at a high level, the model will adjust to the training data, allowing it to learn more and capture relevant patterns. However, if the constant value is set too high, it will lead to overfitting because the model is too adjusted to the training data and less able to generalize the model to new data. Conversely, if the constant value is set low, the model will be less adaptive to the training data. However, if the constant value is set too low, the model will be unable to capture the complexity of the training data, which will result in the inability to capture patterns that occur in the test data. Therefore, an optimal constant value is required for the model to be optimal and to generalize new data. Nevertheless, the selection of the regularization constant value remains a matter of trial and error. There is no specific algorithm that directly determines the optimal value for the regularization constant. With this research, which focuses on the regularization constant selection algorithm, it is anticipated that RELM will become more efficient in processing data.

1.4 Scope and Delimitation

- This research is limited to the selection of a model for the estimation of the RTT dataset, rather than the implementation of the model within TCP. Consequently, the RTT estimation experiment utilizes the RELM model, although the selection of constants within the RELM method is still a process of trial and error. The objective of this research is to create an algorithm that focuses on the selection of constants that produce the most optimal prediction value.

- This research employs a self-constructed dataset, wherein the author generates a virtual topology through the utilization of Mininet, programmed on the Ubuntu operating system within a VirtualBox environment. The topology encompasses four hosts and two principal routers, with a single backbone line connecting the two routers. Thereafter, these four hosts transmit messages to one another, with varying delays. The Round Trip Time (RTT) value, time taken, source address, destination address, and other pertinent data will be recorded.

- The research will be conducted using the MATLAB programming language and the MATLAB Integrated Development Environment (IDE) to construct the RELM and process the dataset generated in VirtualBox.

- It is anticipated that the model constructed will exhibit enhanced accuracy and execution time, thereby enabling optimal estimation of RTT.

- The efficacy of the method developed will be evaluated by means of a series of quantitative assessments, employing evaluation matrices such as MAE, MSE, MAPE, and R2.

- Some machine learning was also built to estimate RTT for comparison with Enhanced RELM.